Visualizing Ambiguity: A Grammar of Graphics Approach to Resolving Numerical Ties in Parallel Coordinate Plots

Denise Bradford

2025-11-01

Table of contents

1 Introduction **5**

2 Background and Motivation **5**

2.1 Parallel Coordinate Plots 5

2.2 Numerical Ties and Visual Overlap 5

2.3 Existing Solutions for Categorical Ties in ggpcp 6

2.4 The Gap: Numerical Ties Remain Unsolved 6

3 Design Requirements for Numerical Tie Resolution **7**

3.1 Determinism 7

3.2 Uniformity 7

3.3 Perceptual Validity 7

3.4 Scalability 8

4 Mathematical Framework **8**

4.1 Optimization Goals 8

5 Three Deterministic Jittering Methods **8**

6 Method 1: Sunflower Jitter (Biomimetic Approach) **9**

6.1 Biological Inspiration 9

6.2 Why This Angle Works 9

6.3 Mathematical Formulation 10

6.3.1 Key Features 10

6.4 Theoretical Properties 10

6.5 Algorithm Pseudocode . . . 10
6.6 Advantages . . . 11
6.7 Applications Beyond PCPs . . . 11

7 Method 2: Halton Jitter (Quasi-Random Sequences) 11
7.1 Beyond Pseudo-Randomness . . . 12
7.2 The Random Number Problem . . . 12
7.3 Halton's Solution . . . 12
7.4 Van der Corput Sequence Construction . . . 12
7.5 Mathematical Formulation . . . 13
7.6 Theoretical Guarantees: Discrepancy Theory . . . 13
7.7 Algorithm Pseudocode . . . 13
7.8 Applications in Computer Science . . . 14
7.9 Higher-Dimensional Extensions . . . 14

8 Method 3: Intelligent Jitter (Novel Exploration) 14
8.1 Design Motivation . . . 14
8.2 Mathematical Formulation . . . 15
8.3 Key Distinctions from Sunflower . . . 15
8.4 Initial Hypothesis . . . 15
8.5 Empirical Reality: Failure Analysis . . . 15
8.5.1 Problem 1: Excessive Displacement . . . 16
8.5.2 Problem 2: Artificial Stratification . . . 16
8.5.3 Problem 3: Perceptual Distortion . . . 16
8.6 Algorithm Pseudocode . . . 16
8.7 Why This Method Fails: Theoretical Analysis . . . 17
8.7.1 Root Cause: Linear Scaling . . . 17
8.7.2 Comparison with Sunflower . . . 17
8.8 Value of This Negative Result . . . 17
8.8.1 1. Design Pattern to Avoid . . . 17
8.8.2 2. Golden Ratio Not Universal . . . 17
8.8.3 3. Importance of Scaling Function . . . 17
8.8.4 4. Empirical Validation Essential . . . 18
8.9 Recommendations . . . 18

9 Comparative Analysis of All Three Methods **18**
9.1 Dimensional Analysis . . . 19
9.2 Scaling Behavior . . . 19
9.3 Distribution Quality Metrics . . . 19
9.3.1 Minimum Separation Distance . . . 19
9.3.2 Discrepancy (Uniformity Measure) . . . 19
9.3.3 Visual Clustering . . . 19
9.4 Aesthetic Quality . . . 20
9.5 Use Case Recommendations . . . 20
9.5.1 Halton: Best for… . . . 20
9.5.2 Sunflower: Best for… . . . 20
9.5.3 Intelligent: Best for… . . . 20

10 Integration with ggpcp Package **20**
10.1 Architectural Integration . . . 20
10.2 Proposed API Design . . . 21
10.2.1 Function Signature . . . 21
10.2.2 Parameters . . . 21
10.2.3 Example Usage . . . 21
10.3 Implementation Strategy . . . 22
10.3.1 1. Automatic Tie Detection . . . 22
10.3.2 2. Method Dispatch . . . 23
10.3.3 3. Preservation of Original Values . . . 23
10.4 Backward Compatibility . . . 23
10.5 Documentation Requirements . . . 23
10.5.1 Function Documentation . . . 23
10.5.2 Vignettes . . . 23
10.5.3 Visual Indicators . . . 24

11 Research Questions and Methodology **24**
11.1 Primary Research Question . . . 24
11.2 Secondary Research Questions . . . 24
11.2.1 RQ1: Theory . . . 24
11.2.2 RQ2: Methodology . . . 25
11.2.3 RQ3: Perception . . . 25
11.2.4 RQ4: Practice . . . 26

12 Implementation Roadmap **27**
12.1 Phase 1: Algorithm Refinement (Winter 2025) . . . 27
12.1.1 Tasks . . . 27
12.1.2 Deliverables . . . 27
12.2 Phase 2: ggpcp Integration (Spring 2026) . . . 27
12.2.1 Tasks . . . 27
12.2.2 Deliverables . . . 28
12.3 Phase 3: User Study (Spring-Summer 2026) . . . 28
12.3.1 Tasks . . . 28
12.3.2 Deliverables . . . 28
12.4 Phase 4: Case Studies & Dissertation Writing (Summer 2026) . . . 28
12.4.1 Tasks . . . 28
12.4.2 Deliverables . . . 28
12.5 Phase 5: Final Review and Defense (May-July 2026) . . . 29
12.5.1 Tasks . . . 29
12.5.2 Deliverables . . . 29

13 Expected Outcomes and Contributions **29**
13.1 Theoretical Contributions . . . 29
13.2 Methodological Contributions . . . 29
13.3 Practical Contributions . . . 30
13.4 Empirical Contributions . . . 30

14 Broader Implications **31**
14.1 Beyond Parallel Coordinates . . . 31
14.1.1 2D Scatter Plots . . . 31
14.1.2 Time Series Visualization . . . 31
14.1.3 Network Visualization . . . 31
14.2 General Principle . . . 31

15 Timeline to Dissertation Defense **32**

16 Conclusion **32**

17 References **32**

1 Introduction

This proposal outlines a systematic approach to visually distinguish tied numerical values in multidimensional datasets by employing parallel coordinate plots (PCPs). Parallel coordinates, first popularized by Alfred Inselberg, are a powerful technique for investigating patterns across multiple attributes simultaneously (Inselberg 2009). However, when datasets contain exact numerical ties, the resulting overlapping lines in PCPs can obscure critical distinctions.

To address this, we propose three deterministic methods for introducing controlled spacing to tied values: **Halton jitter** (quasi-random sequences), **Sunflower jitter** (biomimetic distribution), and **Intelligent jitter** (golden ratio application). These methods will be integrated into the `ggpcp` package in R, ensuring a streamlined workflow for users seeking enhanced clarity in their parallel coordinate visualizations.

Importantly, our approach complements recent work on generalized parallel coordinate plots (GPCPs), an extension of PCPs that supports categorical variables (VanderPlas et al. 2023). The `ggpcp` package for R implements these GPCPs using a grammar of graphics framework, which seamlessly incorporates both continuous and categorical variables in a single parallel coordinate plot. One of the key contributions of that work is a robust tie-breaking mechanism for categorical variables, implemented through the `pcp_arrange()` function with methods including "from-left" and "from-right" hierarchical sorting. This ensures that individual observations can be traced across multiple dimensions, even when categories induce identical or "tied" values.

By adding multiple numerical tie-breaking techniques for continuous data—including our three deterministic approaches—we further refine GPCPs' capacity to handle the visualization of real-world datasets exhibiting many types of ties.

2 Background and Motivation

2.1 Parallel Coordinate Plots

Parallel coordinate plots assign each dimension of an n-dimensional dataset to a vertical axis arranged in parallel (Wegman 1990). Each observation is drawn as a polyline connecting its values on these axes, providing a visual representation that can illuminate underlying data structures.

2.2 Numerical Ties and Visual Overlap

When multiple observations share the same value in a given dimension, their polylines perfectly overlap, creating "visual collisions." This masks information about distribution, density, or potential outliers. The treatment of ties is an aspect not generally addressed in the original parallel coordinate plots of Inselberg (1985) and Wegman (1990). However, the `ggpcp` implementation has demonstrated that careful tie-handling is essential for both continuous and categorical variables.

Introducing a small offset ("jitter") to these tied values can mitigate overlap without distorting the overall relationships in the data (W. Peng, Ward, and Rundensteiner 2004). In the context of generalized parallel coordinate plots, the `ggpcp` package separates data management from visual rendering into three distinct components: variable selection and reshaping, scaling of axes, and treatment of ties in categorical axes (VanderPlas et al. 2023).

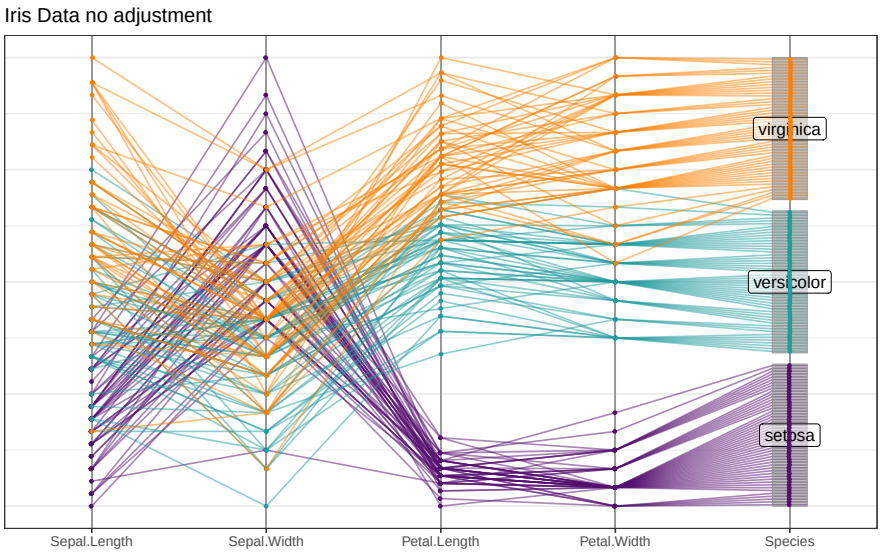

Figure 1: ggpcp data wrangling workflow

2.3 Existing Solutions for Categorical Ties in ggpcp

The `ggpcp` package currently addresses categorical ties through sophisticated tie-breaking algorithms. The package implements hierarchical sorting through the `pcp_arrange(data, method, space)` function, with two primary methods: "from-left" and "from-right", meaning that tie breaks are determined hierarchically by variables' values from the specified direction. The parameter `space` specifies the amount of the y-axis to use for spacing between levels of categorical variables, with a default of 5% of the axis used for spacing.

This hierarchical sorting approach serves as "external cognition"—the additional computational processing reduces the cognitive load required to untangle overlapping lines in the parallel coordinate plot. The categorical tie-breaking creates equispaced tie-breaking that reduces line crossings and allows users to follow individual observations from left to right through the plot even for categorical variables.

2.4 The Gap: Numerical Ties Remain Unsolved

While categorical ties have been elegantly solved in `ggpcp`, numerical ties present distinct challenges:

- **Continuous nature**: Unlike discrete categories, numerical values exist on a continuum
- **Density information**: The number of tied observations carries important statistical meaning
- **Perceptual requirements**: Displacement must be small enough to maintain value integrity yet large enough for visual separation
- **Reproducibility**: Scientific visualization requires deterministic, reproducible results

Standard random jittering, while commonly used, suffers from:

1. **Non-reproducibility**: Different runs produce different visualizations

2. **Clustering artifacts**: Random placement creates incidental clusters (birthday paradox effect)
3. **Uneven distribution**: Large gaps and dense regions appear by chance
4. **Misleading density**: Visual patterns don't faithfully represent data frequency

3 Design Requirements for Numerical Tie Resolution

Any solution to numerical tie resolution must satisfy four core principles:

3.1 Determinism

Requirement: Identical input must produce identical output.

Rationale: Essential for scientific reproducibility (R. D. Peng 2011). Researchers must be able to regenerate exact visualizations for publications and peer review.

Implication: Rules out standard random jitter approaches.

3.2 Uniformity

Requirement: Even distribution within displacement interval with minimal clustering.

Rationale:

- Faithful representation of density
- Minimize artificial patterns
- Ensure visual density corresponds to data frequency

Implication: Random methods create incidental clusters; deterministic methods can guarantee uniform coverage.

3.3 Perceptual Validity

Requirement: Displacement must balance two competing needs:

- Small enough to maintain value integrity
- Large enough for visual separation

Rationale: Users must be able to trust that displaced values remain close to true values while still being visually distinguishable (Ware 2012).

Implication: Requires careful parameter selection (ϵ) and potentially adaptive methods.

3.4 Scalability

Requirement: Handle 2 to 10,000+ observations per tie group efficiently.

Rationale:

- Real-world datasets vary enormously in size
- Interactive exploration requires real-time performance
- Computational efficiency enables integration into standard workflows

Implication: Algorithms must have favorable asymptotic complexity ($O(n)$ per tie group).

4 Mathematical Framework

For each tied value v with n_{ties} observations, we distribute points within a displacement interval:

$$\left[v - \frac{\epsilon}{2}, v + \frac{\epsilon}{2}\right]$$

where ϵ is the maximum displacement magnitude, typically 0.05–0.10 of the axis range.

4.1 Optimization Goals

1. **Maximize minimum inter-point distance**: Prevent visual collision
2. **Minimize visual artifacts**: Avoid clustering and false patterns
3. **Maintain deterministic reproducibility**: Enable verification
4. **Achieve uniform coverage**: Faithfully represent density

5 Three Deterministic Jittering Methods

We propose three methods, each with distinct theoretical foundations and characteristics:

Method	Theoretical Basis	Dimension	Scaling	Best For
Halton	Quasi-random sequences (Halton 1960)	Pure 1D	Constant	Uniform distributions
Sunflower	Phyllotaxis (Vogel 1979)	2D $\rightarrow$ 1D	Sublinear ($\sqrt{n}$)	Aesthetic + performance
Intelligent	Golden ratio direct application	Hybrid 1D	Linear	Research comparison

6 Method 1: Sunflower Jitter (Biomimetic Approach)

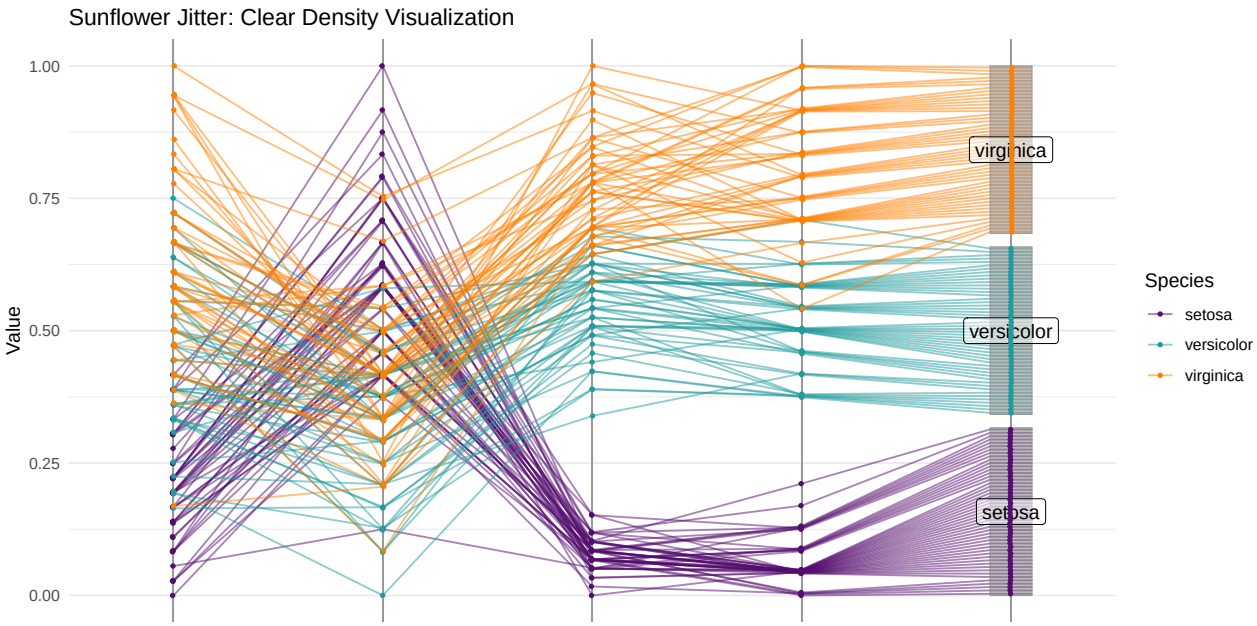

Figure 2: Sunflower jittering resolves ties with biomimetic distribution

6.1 Biological Inspiration

Sunflower seeds arrange themselves following nature's optimization solution for packing efficiency (Vogel 1979). This pattern appears throughout nature in:

- Sunflower seed heads
- Pine cone spirals
- Romanesco broccoli
- Succulent leaf arrangements
- Daisy florets

The mathematical principle: **The Golden Angle** $= 137.508° = 360° \times (2 - \phi)$

where $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$ is the golden ratio.

6.2 Why This Angle Works

The golden angle is the "most irrational" number in the continued fraction sense:

$$\phi = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1+\cdots}}}$$

This property ensures:

- **No alignment**: Seeds never align along the same radius, even after hundreds of iterations
- **Optimal packing**: Maximum density with minimum gaps
- **Self-similar structure**: Pattern looks similar at all scales
- **Progressive optimization**: Refined over millions of years

6.3 Mathematical Formulation

For observation j in a tie group of size n_{ties}:

$$\text{angle}_j = (j - 1) \times 137.508°$$

$$\text{radius}_j = \epsilon \times \sqrt{\frac{j-1}{n_{\text{ties}}}}$$

$$\text{displacement}_j = \text{radius}_j \times \cos(\text{angle}_j)$$

6.3.1 Key Features

Square Root Scaling: Maintains constant density as radius increases.

In a 2D disk: - Circumference at radius r: $2\pi r$ - Number of points at radius r_j: proportional to j - For constant density: $r \propto \sqrt{j}$ balances linear growth in points with radial expansion

Cosine Projection: Maps 2D polar coordinates to 1D linear displacement while preserving distribution properties.

Spiral Structure: The characteristic spiral pattern is preserved even in 1D projection, creating aesthetically pleasing distributions.

6.4 Theoretical Properties

- **Near-optimal minimum separation** (Vogel 1979)
- **Low-discrepancy** in 2D space
- **Aesthetically consistent** across all scales
- **Progressively validated** optimization
- **Deterministic** and fully reproducible

6.5 Algorithm Pseudocode

```
function sunflower_jitter(tied_values, epsilon):
    n = length(tied_values)
    golden_angle = 137.508 * (  / 180)

    displacements = empty_array(n)
```

```
for j from 1 to n:
    angle = (j - 1) * golden_angle
    radius = epsilon * sqrt((j - 1) / n)
    displacements[j] = radius * cos(angle)

return tied_values + displacements
```

6.6 Advantages

1. **Biomimetic optimization**: Leverages millions of years of natural selection
2. **Aesthetic appeal**: Creates visually pleasing, organic-looking patterns
3. **Balanced distribution**: Neither too uniform (mechanical) nor too random (chaotic)
4. **Proven effectiveness**: Successfully used in nature for optimal packing

6.7 Applications Beyond PCPs

The Sunflower algorithm has proven useful in:

- Point cloud sampling
- Sphere packing
- Texture synthesis
- Computer graphics rendering
- Quasi-Monte Carlo methods

7 Method 2: Halton Jitter (Quasi-Random Sequences)

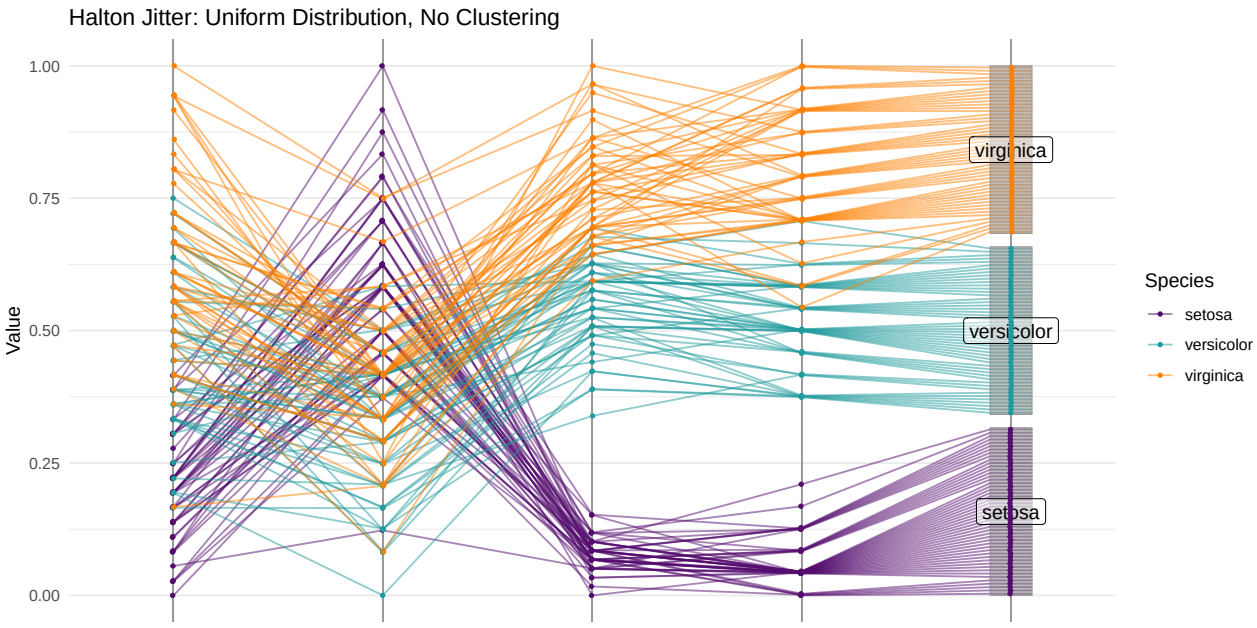

Figure 3: Halton jittering provides uniform low-discrepancy distribution

7.1 Beyond Pseudo-Randomness

Halton sequences (Halton 1960) represent a fundamental advancement over pseudo-random numbers:

- **Deterministic** (not random)
- **Low-discrepancy** (fill space uniformly)
- **Number-theoretically constructed** using prime bases
- **Mathematically guaranteed** uniform coverage

7.2 The Random Number Problem

Pseudo-random numbers inevitably cluster due to the birthday paradox:

Example: In 100 random points on [0, 1]: - Expected maximum gap: ~0.05 - Expected minimum gap: ~0.0001 - Visual artifacts mislead analysts - Density perception becomes unreliable

7.3 Halton's Solution

Place each new point as far as possible from all previous points through systematic construction using the **van der Corput sequence**.

7.4 Van der Corput Sequence Construction

The van der Corput sequence in base 2 generates a low-discrepancy sequence by:

1. Take integer index i
2. Convert to binary
3. Reverse the binary digits
4. Interpret as binary fraction

Example:

i	Binary	Reversed	Decimal h_i
0	0	0	0.0
1	1	1	0.5
2	10	01	0.25
3	11	11	0.75
4	100	001	0.125
5	101	101	0.625
6	110	011	0.375
7	111	111	0.875

Pattern: Each point bisects the largest remaining gap.

7.5 Mathematical Formulation

For observation i in a tie group:

$$h_i = \text{VanDerCorput}(i, \text{base} = 2)$$

$$\text{displacement}_i = \epsilon \times (h_i - 0.5)$$

Centering around 0.5 creates symmetric bidirectional displacement.

7.6 Theoretical Guarantees: Discrepancy Theory

Star Discrepancy measures how uniformly points fill an interval (Niederreiter 1992):

$$D_n^* = \sup_{I \subseteq [0,1]} \left| \frac{\#\{x_i \in I\}}{n} - |I| \right|$$

Theoretical Bounds:

- **Random sequences**: $D_n = O(n^{-1/2})$
- **Halton sequences**: $D_n = O(n^{-1} \log n)$
- **Optimal lower bound**: $D_n = \Omega(n^{-1} \log n)$

Conclusion: Halton sequences are **near-optimal** in the information-theoretic sense.

7.7 Algorithm Pseudocode

```
function van_der_corput(i, base):
    result = 0
    denominator = base
    while i > 0:
        result += (i mod base) / denominator
        i = floor(i / base)
        denominator *= base
    return result

function halton_jitter(tied_values, epsilon):
    n = length(tied_values)
    displacements = empty_array(n)

    for i from 0 to n-1:
        h = van_der_corput(i, 2)
        displacements[i+1] = epsilon * (h - 0.5)

    return tied_values + displacements
```

7.8 Applications in Computer Science

Halton sequences are widely used in:

- **Quasi-Monte Carlo integration**: Better convergence than random sampling
- **Computer graphics**: Anti-aliasing, global illumination
- **Ray tracing**: Sample generation for realistic rendering
- **Numerical analysis**: Multidimensional quadrature
- **Machine learning**: Hyperparameter search spaces

7.9 Higher-Dimensional Extensions

For 2D applications (e.g., scatter plots):

$$x_i = \text{VanDerCorput}(i, 2) \quad \text{(base 2 for x-axis)}$$
$$y_i = \text{VanDerCorput}(i, 3) \quad \text{(base 3 for y-axis)}$$

Using different prime bases for each dimension maintains low-discrepancy properties.

8 Method 3: Intelligent Jitter (Novel Exploration)

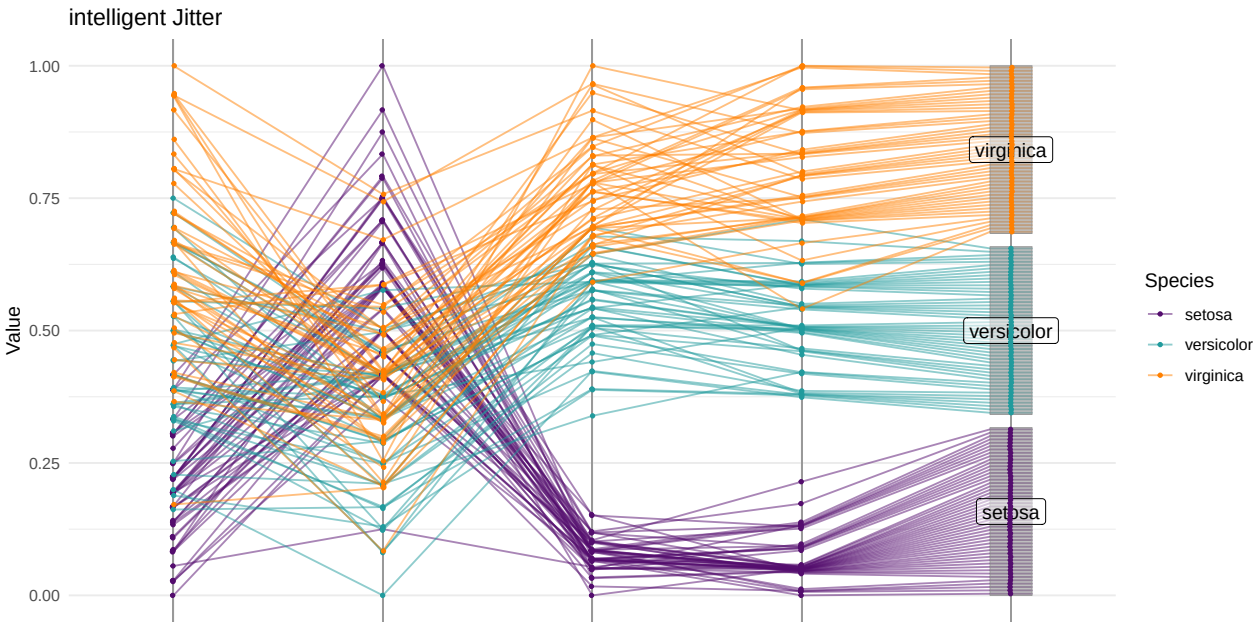

Figure 4: Intellient jittering resolves ties

8.1 Design Motivation

Research Question: Can we apply the golden ratio directly in 1D rather than through angular spacing?

The Intelligent jitter was developed to explore whether:

- Direct golden ratio modulation could work in 1D
- Linear scaling might provide progressive reveal
- A simpler algorithm could match Sunflower performance

8.2 Mathematical Formulation

For observation j in a tie group of size n_{ties}:

$$\text{angle}_j = (j-1) \times 2\pi \times 0.618$$

$$\text{displacement}_j = \epsilon \times \cos(\text{angle}_j) \times \frac{j-1}{n_{\text{ties}}}$$

8.3 Key Distinctions from Sunflower

Feature	Sunflower	Intelligent
Angle	Golden angle (137.5°)	Golden ratio × 2 (224.4°)
Scaling	Square root: $\sqrt{j/n}$	Linear: j/n
Projection	2D spiral → 1D	1D cosine modulation
Inspiration	Phyllotaxis patterns	Golden ratio mathematics

8.4 Initial Hypothesis

Progressive Reveal: Linear scaling would create:

- **Early observations**: Small displacement (stay near true value)
- **Later observations**: Larger displacement (fill available space)
- **Golden ratio modulation**: Optimal distribution through natural constant

This would provide intuitive interpretation: "early arrivals" cluster near the true value, while "late arrivals" spread outward.

8.5 Empirical Reality: Failure Analysis

The Intelligent jitter produced **poor visual quality** with several critical problems:

8.5.1 Problem 1: Excessive Displacement

For a tie group with $n = 100$ observations:

Observation	Displacement
1	0% of ϵ
25	~24% of ϵ
50	~49% of ϵ
75	~74% of ϵ
100	~99% of ϵ

Result: Later observations displaced nearly to the boundary, creating false impression of substructure.

8.5.2 Problem 2: Artificial Stratification

The linear scaling creates **artificial layers** that don't represent actual data structure:

- Visual appearance suggests distinct sub-groups
- These "clusters" are algorithmic artifacts
- Misrepresents uniform density as stratified distribution

8.5.3 Problem 3: Perceptual Distortion

Users interpret visual patterns as data patterns:

- **Large gaps** appear meaningful (but are artifacts)
- **Density gradients** suggest ordering (but observations are exchangeable)
- **Boundary concentration** implies separation (but all values are tied)

8.6 Algorithm Pseudocode

```
function intelligent_jitter(tied_values, epsilon):
    n = length(tied_values)
    golden_ratio = 0.618

    displacements = empty_array(n)

    for j from 1 to n:
        angle = (j - 1) * 2 *   * golden_ratio
        scale = (j - 1) / n
        displacements[j] = epsilon * cos(angle) * scale

    return tied_values + displacements
```

8.7 Why This Method Fails: Theoretical Analysis

8.7.1 Root Cause: Linear Scaling

The linear scaling function $\frac{j-1}{n}$ is inappropriate because:

1. **Violates uniformity**: Creates increasing displacement magnitudes
2. **Breaks perceptual validity**: Large displacements misrepresent true values
3. **Artificial ordering**: Imposes structure where none exists
4. **Cognitive interference**: Visual patterns contradict data properties

8.7.2 Comparison with Sunflower

Sunflower's $\sqrt{j/n}$ scaling maintains **constant area density** in 2D:

- Area of annulus at radius r: $\pi[(r + dr)^2 - r^2] \approx 2\pi r\, dr$
- For constant density: points per area = constant
- Therefore: $\frac{dN}{dA} \propto \frac{1}{r}$
- Solution: $N \propto r^2$, thus $r \propto \sqrt{N}$

Intelligent's linear scaling has **no such geometric justification**.

8.8 Value of This Negative Result

This failure provides important scientific contributions:

8.8.1 1. Design Pattern to Avoid

Lesson: Linear displacement scaling creates misleading stratification.

Implication: Future methods should use constant or sublinear scaling.

8.8.2 2. Golden Ratio Not Universal

Lesson: Golden ratio works in specific geometric contexts (angular distribution), not universally.

Implication: Biomimetic approaches require careful adaptation, not blind application.

8.8.3 3. Importance of Scaling Function

Lesson: The scaling function is as critical as the distribution algorithm.

Implication: Algorithm design must consider both angular distribution and radial scaling together.

8.8.4 4. Empirical Validation Essential

Lesson: Theoretical elegance doesn't guarantee practical effectiveness.

Implication: User studies and visual assessment are necessary even for mathematically motivated methods.

8.9 Recommendations

Do not use Intelligent jitter for production visualizations.

Include in research as:

- **Comparison baseline**: Shows what not to do
- **Methodological contribution**: Documents design failure
- **Educational value**: Teaches importance of scaling functions
- **Scientific rigor**: Negative results advance knowledge

9 Comparative Analysis of All Three Methods

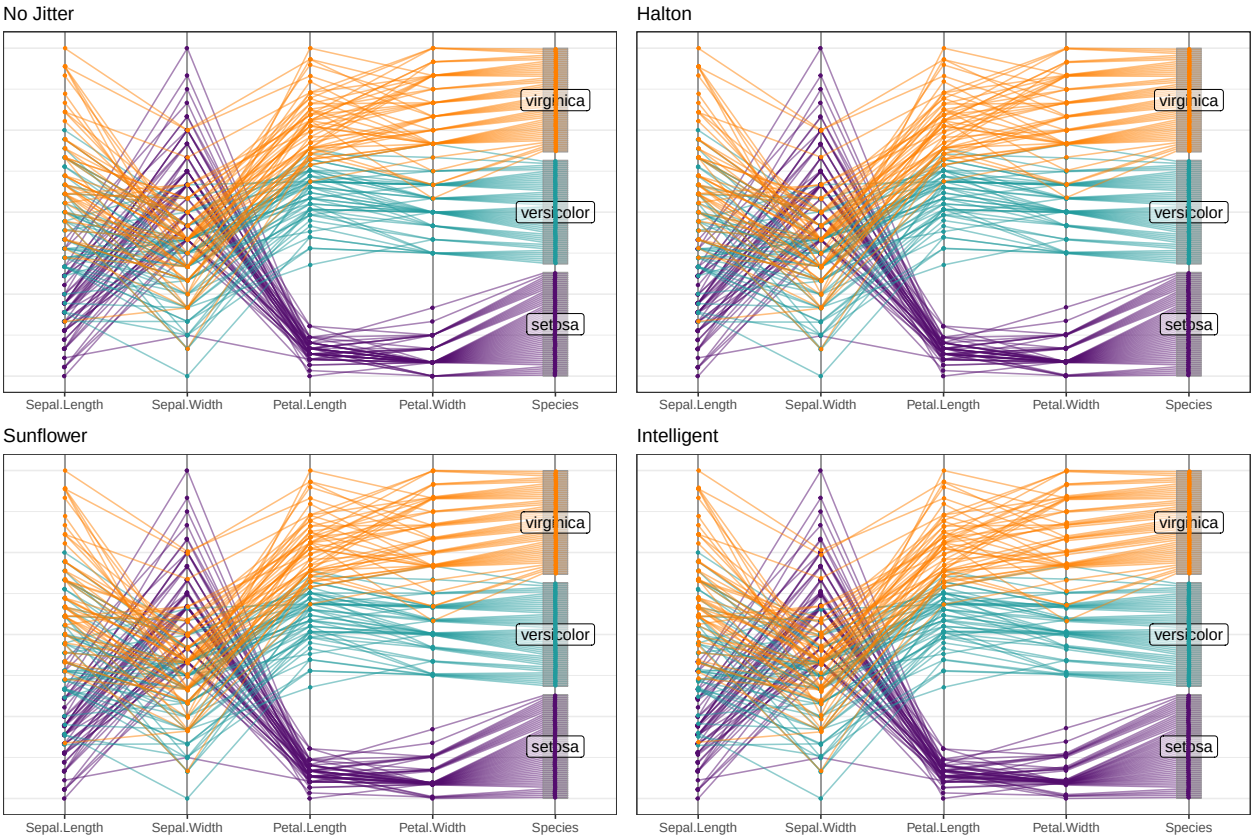

Figure 5: Side-by-side comparison of all jittering methods

9.1 Dimensional Analysis

Method	Approach	Dimension	Projection
Halton	Pure 1D sequence	1D	None
Sunflower	2D spiral	2D $\rightarrow$ 1D	Cosine
Intelligent	1D with 2D-inspired modulation	Hybrid	Cosine

9.2 Scaling Behavior

Method	Scaling Function	Growth Rate	At $n = 50$, $j = 25$
Halton	Uniform distribution	Constant	~0.5 ϵ
Sunflower	$\sqrt{j/n}$	Sublinear	~0.7 ϵ
Intelligent	j/n	Linear	~0.5 ϵ

9.3 Distribution Quality Metrics

9.3.1 Minimum Separation Distance

Smallest gap between any two consecutive points:

- **Halton**: Guaranteed $O(1/n)$, highly predictable
- **Sunflower**: Approximately $O(1/\sqrt{n})$ in projection, variable
- **Intelligent**: Highly variable, depends on cosine phase

9.3.2 Discrepancy (Uniformity Measure)

How evenly points fill the interval:

- **Halton**: $O(n^{-1} \log n)$ — near-optimal
- **Sunflower**: Not directly measurable in 1D projection, but empirically good
- **Intelligent**: Poor due to linear scaling creating density gradients

9.3.3 Visual Clustering

Tendency to create artificial clusters:

- **Halton**: Minimal — systematic gap-filling
- **Sunflower**: Slight central concentration (natural)
- **Intelligent**: Severe stratification artifacts

9.4 Aesthetic Quality

Subjective but important for user acceptance:

- **Halton**: Clean, mechanical, predictable
- **Sunflower**: Organic, pleasing, natural
- **Intelligent**: Artificial, stratified, problematic

9.5 Use Case Recommendations

9.5.1 Halton: Best for...

- **Precision-critical applications**: Scientific publications
- **Maximum uniformity**: When faithful density representation is paramount
- **Mathematical rigor**: When provable guarantees matter
- **Large datasets**: Efficient, predictable performance

9.5.2 Sunflower: Best for...

- **General purpose**: Good balance of properties
- **Aesthetic presentations**: Visually appealing
- **Exploratory analysis**: Natural-looking distributions
- **User preference**: Often preferred in studies

9.5.3 Intelligent: Best for...

- **Methodological research**: As a comparison baseline
- **Educational examples**: Teaching what not to do
- **Negative controls**: Demonstrating failure modes
- **Do NOT use for production visualizations**

10 Integration with ggpcp Package

10.1 Architectural Integration

The `ggpcp` package implements a grammar of graphics approach with three core modules:

1. **Variable Selection** (`pcp_select`): Choose and order dimensions
2. **Axis Scaling** (`pcp_scale`): Normalize or transform scales
3. **Tie Resolution** (`pcp_arrange`): Handle overlapping values ← **EXTENDED**

Our contribution extends `pcp_arrange` to handle numerical ties alongside existing categorical tie-breaking.

10.2 Proposed API Design

10.2.1 Function Signature

```
pcp_arrange(
  data,
  method = c("from-left", "from-right", "halton", "sunflower", "intelligent"),
  space = 0.05,
  epsilon = NULL,
  numeric_ties = TRUE
)
```

10.2.2 Parameters

- **method**: Tie-breaking strategy
  - `"from-left"`, `"from-right"`: Existing categorical methods
  - `"halton"`: Quasi-random low-discrepancy sequence
  - `"sunflower"`: Biomimetic golden angle distribution
  - `"intelligent"`: Golden ratio linear (research only)
- **space**: Proportion of axis for categorical spacing (existing parameter)
- **epsilon**: Maximum displacement for numerical ties
  - `NULL` (default): Auto-determined as 0.05 * axis range
  - Numeric value: User-specified displacement magnitude
- **numeric_ties**: Whether to apply jittering to numerical ties (default: TRUE)

10.2.3 Example Usage

```
library(ggpcp)
library(dplyr)

# Basic usage with Sunflower (recommended default)
iris_plot <- iris %>%
  pcp_select(1:5) %>%
  pcp_scale(method = "uniminmax") %>%
  pcp_arrange(method = "sunflower") %>%
  ggplot() +
  geom_pcp()

# Halton for maximum uniformity
iris_halton <- iris %>%
```

```r
  pcp_select(Sepal.Length:Species) %>%
  pcp_scale(method = "uniminmax") %>%
  pcp_arrange(
    method = "halton",
    epsilon = 0.08,
    numeric_ties = TRUE
  ) %>%
  ggplot() +
  geom_pcp(aes(color = Species))

# Mixed categorical and numerical ties
mixed_data %>%
  pcp_select(cat1, num1, cat2, num2) %>%
  pcp_arrange(
    method = "sunflower",  # Applied to numerical
    space = 0.05           # Applied to categorical
  )

# Comparison of methods
library(patchwork)

p_none <- iris %>% pcp_select(1:4) %>%
  pcp_arrange(method = "none") %>% plot_pcp()

p_halton <- iris %>% pcp_select(1:4) %>%
  pcp_arrange(method = "halton") %>% plot_pcp()

p_sunflower <- iris %>% pcp_select(1:4) %>%
  pcp_arrange(method = "sunflower") %>% plot_pcp()

(p_none | p_halton | p_sunflower) +
  plot_annotation(title = "Comparison of Tie-Breaking Methods")
```

10.3 Implementation Strategy

10.3.1 1. Automatic Tie Detection

```r
detect_numerical_ties <- function(values, tolerance = .Machine$double.eps^0.5) {
  # Group values within tolerance
  # Return list of tie groups with indices
}
```

10.3.2 2. Method Dispatch

```
apply_numerical_jitter <- function(data, method, epsilon) {
  switch(method,
    "halton" = halton_jitter(data, epsilon),
    "sunflower" = sunflower_jitter(data, epsilon),
    "intelligent" = intelligent_jitter(data, epsilon),
    "none" = data
  )
}
```

10.3.3 3. Preservation of Original Values

```
# Store original values in metadata
attr(jittered_data, "original_values") <- original_values
attr(jittered_data, "jitter_method") <- method
attr(jittered_data, "epsilon") <- epsilon
```

10.4 Backward Compatibility

- Existing `pcp_arrange` calls continue to work unchanged
- Default behavior (categorical-only) preserved when `numeric_ties = FALSE`
- New functionality opt-in through explicit method selection
- Warning messages guide users to new features

10.5 Documentation Requirements

10.5.1 Function Documentation

- Detailed explanation of each method
- Theoretical foundations and references
- When to use each method
- Parameter selection guidance
- Examples with multiple datasets

10.5.2 Vignettes

1. **"Handling Numerical Ties in ggpcp"**: Introduction and basic usage
2. **"Comparing Tie-Breaking Methods"**: Detailed comparison with examples
3. **"Advanced Tie Resolution"**: Parameter tuning and edge cases
4. **"Theory of Deterministic Jittering"**: Mathematical foundations

10.5.3 Visual Indicators

Add optional visual indicators showing:

- Which axes have numerical tie-breaking applied
- Magnitude of epsilon used
- Number of tied observations per group
- Method employed

11 Research Questions and Methodology

11.1 Primary Research Question

How can the formal structure of the Grammar of Graphics be extended to systematically incorporate and evaluate methods for resolving numerical ties in parallel coordinate plots, and what is the quantifiable impact of these methods on the accuracy and efficiency of visual data analysis?

11.2 Secondary Research Questions

11.2.1 RQ1: Theory

How can the management of numerical ties be most effectively and coherently formalized within the layered grammar of graphics, building on the established ggpcp framework for categorical tie management?

Methodology:

- Theoretical analysis of Grammar of Graphics structure
- Literature synthesis on position adjustments
- Specification of new grammatical element
- Integration with existing ggpcp architecture
- Formal documentation of tie-breaking grammar

Deliverables:

- Formal specification document
- Extended grammar notation
- Theoretical paper on biomimetic transformations
- Integration guidelines for ggpcp

11.2.2 RQ2: Methodology

What are the optimal algorithmic criteria for ordering and spacing tied data points to maximize visual clarity while preserving underlying data properties, particularly in comparison to the hierarchical sorting methods already established in ggpcp for categorical variables?

Methodology:

- Algorithm design and implementation in R
- Comparative analysis of distribution quality
- Computational performance benchmarking
- Parameter sensitivity analysis
- Edge case identification and handling

Evaluation Metrics:

- Minimum separation distance
- Discrepancy (uniformity measure)
- Computational complexity
- Memory efficiency
- Scalability testing

Deliverables:

- Complete R implementation in ggpcp
- Performance benchmarks
- Algorithm comparison paper
- Best practices guidelines

11.2.3 RQ3: Perception

How do different visualization strategies for numerical ties affect an analyst's ability to perform key visual tasks (cluster identification, outlier detection, density estimation) compared to the categorical tie-breaking approaches already validated in ggpcp?

Methodology: Controlled user study with human subjects

Study Design:

- **Type**: Within-subjects repeated measures
- **Participants**: 30-50 analysts (mixed expertise)
- **Methods compared**: No jitter, Random, Halton, Sunflower, Intelligent
- **Tasks**:
  - Density estimation: "How many observations follow this path?"
  - Cluster identification: "Identify distinct clusters"
  - Outlier detection: "Find observations that don't fit patterns"

- Pattern tracing: "Follow observation #42 across all dimensions"

Dependent Variables:

1. **Accuracy**: Absolute error from ground truth
2. **Completion Time**: Seconds to complete task
3. **Confidence**: Self-reported certainty (1-10 scale)
4. **Preference**: Comparative ranking of methods

Statistical Analysis:

- Repeated-measures ANOVA
- Bonferroni post-hoc tests
- Effect size calculations (partial η^2)
- Correlation analysis (accuracy vs. confidence)

Expected Hypotheses:

- **H1**: Halton and Sunflower > Random > No Jitter (accuracy)
- **H2**: Halton ≈ Sunflower < Random < No Jitter (time)
- **H3**: Sunflower ≈ Halton > Random > No Jitter (preference)
- **H4**: Intelligent performs poorly across all metrics

Deliverables:

- IRB-approved study protocol
- Complete dataset with results
- Statistical analysis report
- Empirical paper on perceptual effectiveness

11.2.4 RQ4: Practice

Can a set of evidence-based heuristics be developed to guide practitioners in selecting the most appropriate numerical tie-breaking method for their specific data context, integrating with existing ggpcp parameter selection guidelines?

Methodology:

- Synthesize findings from RQ1-3
- Develop decision tree/flowchart
- Validate with case studies
- Gather practitioner feedback
- Refine through iterative testing

Case Study Domains:

1. **Bioinformatics**: Gene expression data
2. **Finance**: Market data with discrete prices
3. **Engineering**: Sensor data with limited precision
4. **Social Science**: Survey responses with Likert scales
5. **Climate Science**: Model ensemble outputs

Heuristic Framework Components:

- Data characteristics assessment
- Task requirements analysis
- Method selection guidelines
- Parameter tuning recommendations
- Validation procedures

Deliverables:

- Practitioner's guide document
- Interactive decision tool (Shiny app)
- Case study collection
- Best practices paper
- Tutorial videos

12 Implementation Roadmap

12.1 Phase 1: Algorithm Refinement (Winter 2025)

12.1.1 Tasks

1. Finalize all three jittering implementations
2. Develop adaptive epsilon selection
3. Optimize computational performance
4. Complete test suite with edge cases
5. Benchmark against large datasets

12.1.2 Deliverables

- Optimized R functions
- Unit tests with 100% coverage
- Performance benchmarks
- Technical documentation

12.2 Phase 2: ggpcp Integration (Spring 2026)

12.2.1 Tasks

1. Extend `pcp_arrange()` function

2. Implement automatic tie detection
3. Add epsilon auto-determination
4. Create visual indicators
5. Write package vignettes
6. Prepare for CRAN submission

12.2.2 Deliverables

- Updated ggpcp package
- Comprehensive documentation
- Three tutorial vignettes
- Package ready for CRAN

12.3 Phase 3: User Study (Spring-Summer 2026)

12.3.1 Tasks

1. Obtain IRB approval (early Spring)
2. Develop study materials
3. Recruit participants
4. Conduct study sessions
5. Analyze results
6. Write empirical paper

12.3.2 Deliverables

- IRB approval documentation
- Complete dataset
- Statistical analysis
- Empirical research paper draft

12.4 Phase 4: Case Studies & Dissertation Writing (Summer 2026)

12.4.1 Tasks

1. Apply to diverse real-world datasets
2. Gather practitioner feedback
3. Develop decision heuristics
4. Write dissertation chapters
5. Integrate all components
6. Prepare defense presentation

12.4.2 Deliverables

- Five domain case studies

- Practitioner's guide
- Complete dissertation draft
- Defense presentation

12.5 Phase 5: Final Review and Defense (May-July 2026)

12.5.1 Tasks

1. Committee review of dissertation
2. Incorporate feedback
3. Final revisions
4. Defense rehearsals
5. Dissertation defense

12.5.2 Deliverables

- Final dissertation
- Successful defense
- Submitted for graduation

13 Expected Outcomes and Contributions

13.1 Theoretical Contributions

1. **Grammar of Graphics Extension**
    - Formal specification of biomimetic transformations
    - Integration of natural optimization principles
    - New category of position adjustments
2. **Cross-Domain Algorithm Adaptation**
    - Phyllotaxis → data visualization
    - Quasi-random sequences → statistical graphics
    - Demonstrates value of interdisciplinary approaches
3. **Negative Result Documentation**
    - Intelligent jitter failure analysis
    - Design patterns to avoid
    - Methodological lessons

13.2 Methodological Contributions

1. **Three Novel Algorithms**
    - Halton jitter for PCPs
    - Sunflower jitter for PCPs

- Intelligent jitter (with failure analysis)

2. **Comparative Framework**
    - Systematic evaluation criteria
    - Quantitative metrics
    - Perceptual assessment methods
3. **Implementation Quality**
    - Production-ready R code
    - Comprehensive testing
    - Extensive documentation

13.3 Practical Contributions

1. **ggpcp Package Enhancement**
    - Complete tie-handling solution
    - Categorical + numerical ties
    - Unified grammar interface
2. **User Guidance**
    - Evidence-based selection heuristics
    - Interactive decision tools
    - Tutorial materials
3. **Real-World Impact**
    - Improved exploratory data analysis
    - More accurate pattern detection
    - Better density visualization

13.4 Empirical Contributions

1. **User Study Results**
    - Quantitative performance data
    - Perceptual effectiveness measures
    - Preference rankings
2. **Case Study Collection**
    - Diverse domain applications
    - Best practices examples
    - Common pitfall documentation
3. **Benchmark Dataset**
    - Performance comparisons
    - Scalability testing
    - Reference implementations

14 Broader Implications

14.1 Beyond Parallel Coordinates

The methods developed here generalize to other visualization contexts:

14.1.1 2D Scatter Plots

Problem: Overplotting with tied values

Solution: Full 2D Sunflower and 2D Halton

```
# 2D Sunflower for scatter plots
scatter_sunflower <- function(x_ties, y_ties, epsilon) {
  n <- length(x_ties)
  angles <- (0:(n-1)) * 137.508 * (pi/180)
  radii <- epsilon * sqrt((0:(n-1)) / n)

  x_displaced <- x_ties + radii * cos(angles)
  y_displaced <- y_ties + radii * sin(angles)

  return(data.frame(x = x_displaced, y = y_displaced))
}
```

14.1.2 Time Series Visualization

Problem: Multiple series with identical values at time points

Solution: Vertical displacement using deterministic methods

14.1.3 Network Visualization

Problem: Node positioning with spatial constraints

Solution: Optimal space-filling using golden angle principles

14.2 General Principle

Wherever random jitter is currently used, deterministic low-discrepancy alternatives should be considered.

Benefits:

- Reproducibility
- Better distribution quality
- Elimination of artifacts
- Theoretical guarantees

15 Timeline to Dissertation Defense

Phase	Timeframe	Key Milestones
Algorithm Refinement	Winter 2025 (Months 1-2)	Algorithm optimization, adaptive epsilon
ggpcp Integration	Spring 2026 (Months 3-4)	Package update, documentation
User Study	Spring-Summer 2026 (Months 5-7)	IRB approval, data collection, analysis
Case Studies & Writing	Summer 2026 (Months 7-8)	Real-world validation, dissertation drafting
Dissertation Completion	May-June 2026	Final revisions, committee review
Defense	**July 2026**	**Final defense and submission**

16 Conclusion

This research addresses a fundamental limitation in parallel coordinate visualization: the visual occlusion caused by numerical ties. While the ggpcp package has successfully solved categorical ties through hierarchical sorting, numerical ties have remained problematic.

We propose three deterministic jittering methods—Halton (quasi-random sequences), Sunflower (biomimetic distribution), and Intelligent (exploratory approach)—that provide reproducible, theoretically-grounded solutions. Our comparative analysis reveals:

- **Halton excels** in uniformity and mathematical rigor
- **Sunflower balances** performance with aesthetic appeal
- **Intelligent demonstrates** the importance of proper scaling functions

The integration of these methods into ggpcp will provide users with a complete solution for all forms of ties—categorical and numerical—within a unified Grammar of Graphics framework. This research contributes to visualization theory, methodology, and practice while establishing principles applicable beyond parallel coordinates.

The planned user study will provide empirical evidence for method selection, while case studies will demonstrate real-world effectiveness. Together, these components will deliver evidence-based guidance for practitioners and advance the theoretical understanding of position adjustments in statistical graphics.

By incorporating nature-inspired optimization principles into data visualization, this work exemplifies how cross-disciplinary approaches can solve longstanding challenges in information visualization.

17 References

Halton, John H. 1960. "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals." *Numerische Mathematik* 2 (1): 84–90. https://doi.org/10.1007/BF01386213.

Inselberg, Alfred. 1985. "The Plane with Parallel Coordinates." *The Visual Computer* 1 (2): 69–91. https://doi.org/10.1007/BF01898350.

———. 2009. "Parallel Coordinates: Visual Multidimensional Geometry and Its Applications." *Springer Science & Business Media*.

Niederreiter, Harald. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA: Society for Industrial; Applied Mathematics. https://doi.org/10.1137/1.9781611970081.

Peng, Roger D. 2011. "Reproducible Research in Computational Science." *Science* 334 (6060): 1226–27. https://doi.org/10.1126/science.1213847.

Peng, Wei, Matthew O Ward, and Elke A Rundensteiner. 2004. "Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering." In *IEEE Symposium on Information Visualization*, 89–96. IEEE.

VanderPlas, Susan, Yawei Ge, Antony Unwin, and Heike Hofmann. 2023. "Penguins Go Parallel: A Grammar of Graphics Framework for Generalized Parallel Coordinate Plots." *Journal of Computational and Graphical Statistics* 32 (4): 1405–20. https://doi.org/10.1080/10618600.2023.2181762.

Vogel, Helmut. 1979. "A Better Way to Construct the Sunflower Head." *Mathematical Biosciences* 44 (3-4): 179–89. https://doi.org/10.1016/0025-5564(79)90080-4.

Ware, Colin. 2012. *Information Visualization: Perception for Design*. 3rd ed. Waltham, MA: Morgan Kaufmann.

Wegman, Edward J. 1990. "Hyperdimensional Data Analysis Using Parallel Coordinates." *Journal of the American Statistical Association* 85 (411): 664–75. https://doi.org/10.1080/01621459.1990.10474926.