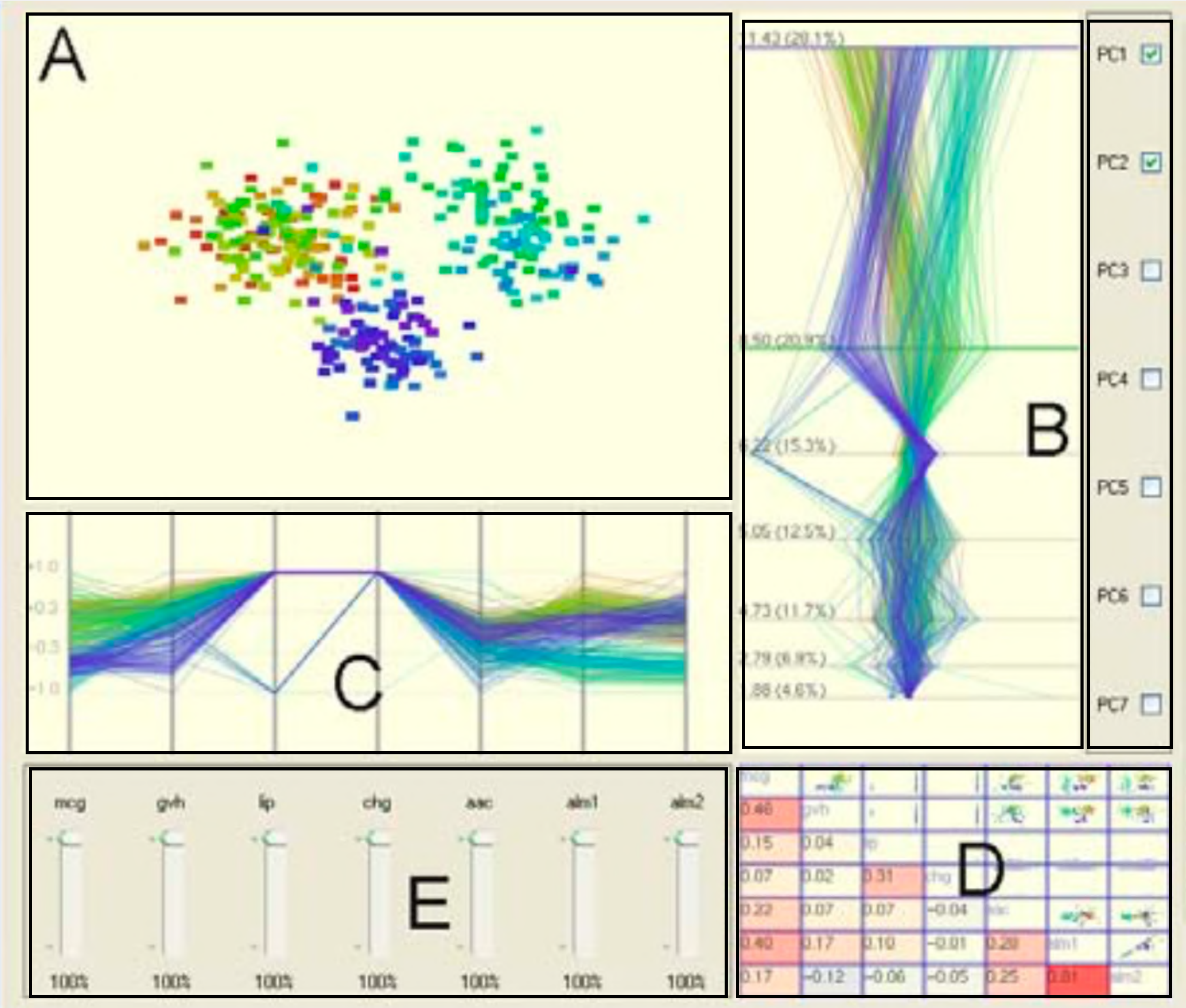
Interactive Dimensionality Reduction

Interactivity + DR: Tasks

- As we discussed: many design decisions regarding visual encodings and interactions. What to choose?
- Design should be driven by **user tasks**.
    - Understand **clusters**, **trends**, and **outliers** in data.
    - Model **fit**, and model **deviation**.
    - Model **selection**: visualize data under different views of the model.

iPCA

[Jeong et al. 2009]



Types of Tasks in iPCA

- Find an outlier. (what *is* an outlier?)
- Find two dimensions with high correlation, along with a point that does not fit such correlation.
- What dimension least impacts PCA?
- Include/exclude a specific dimension, how does the projection change?

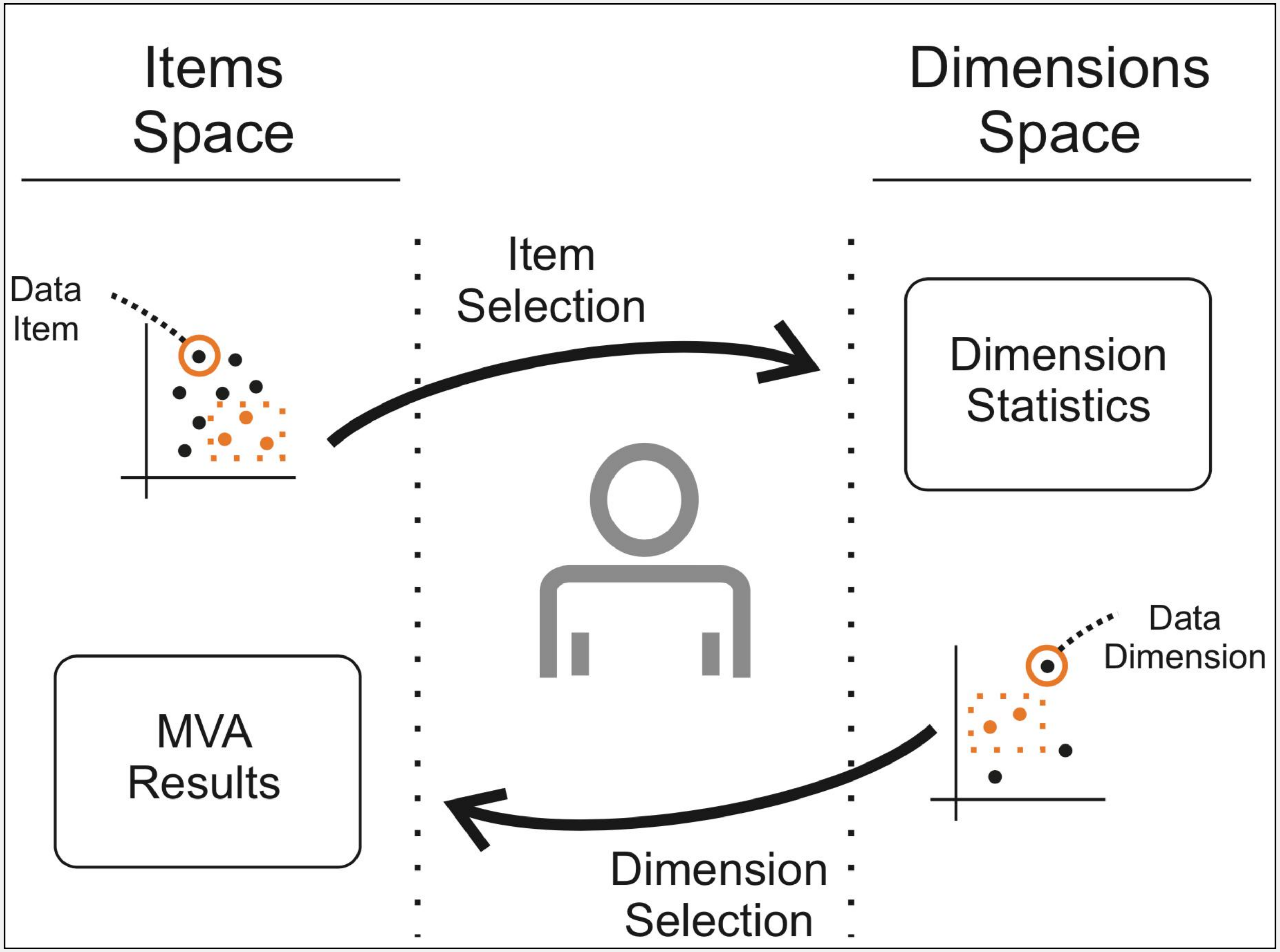
iPCA Notebook

Steering PCA

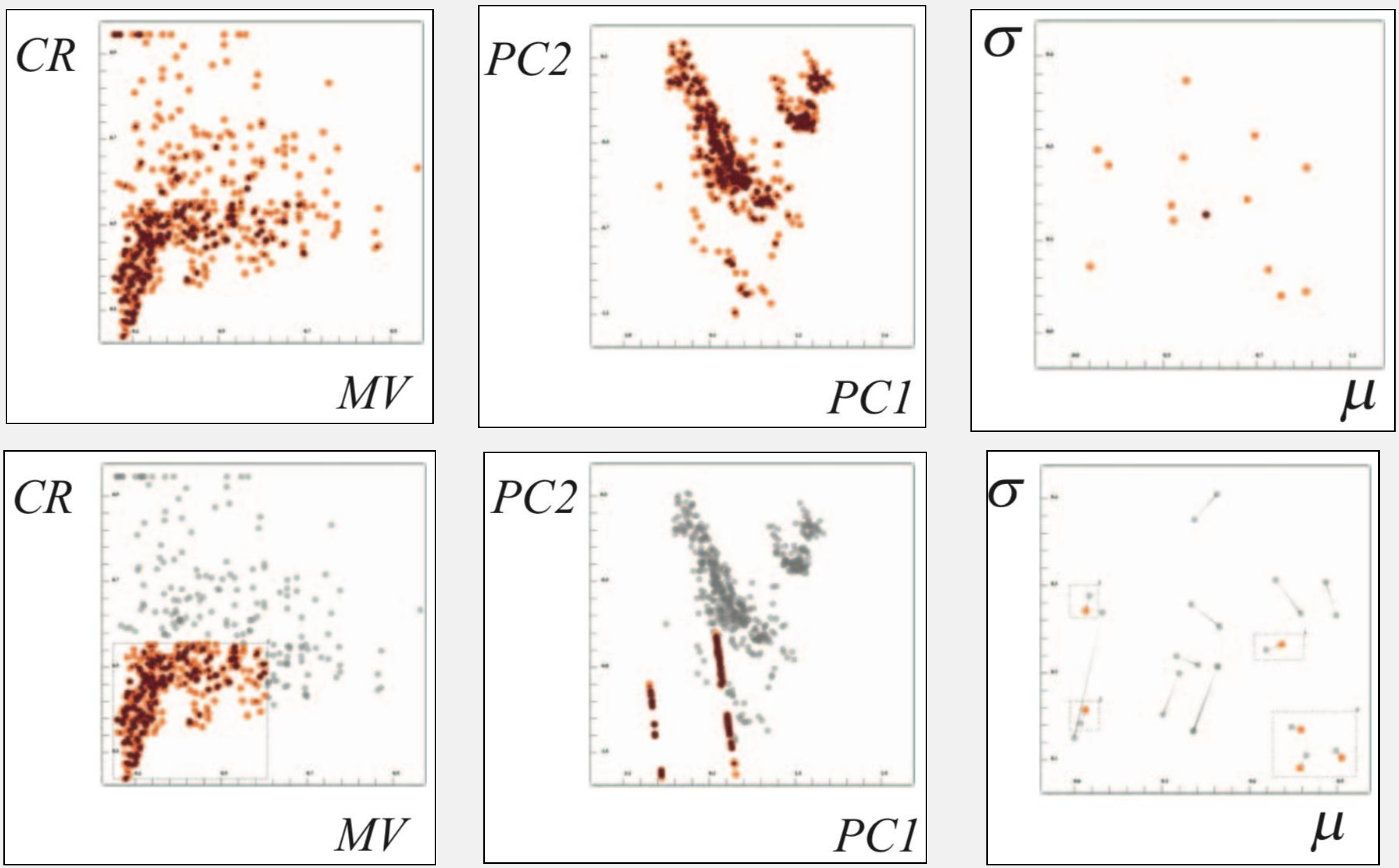
- iPCA allows us to weight individual dimensions.
- But what *guidance* are we given on adjusting dimensions?
- How can we provide the user some insight on what would be of interest to change in PCA? Subsets of data? Dimensions?

Dual Space

[Turkay et al. 2011]

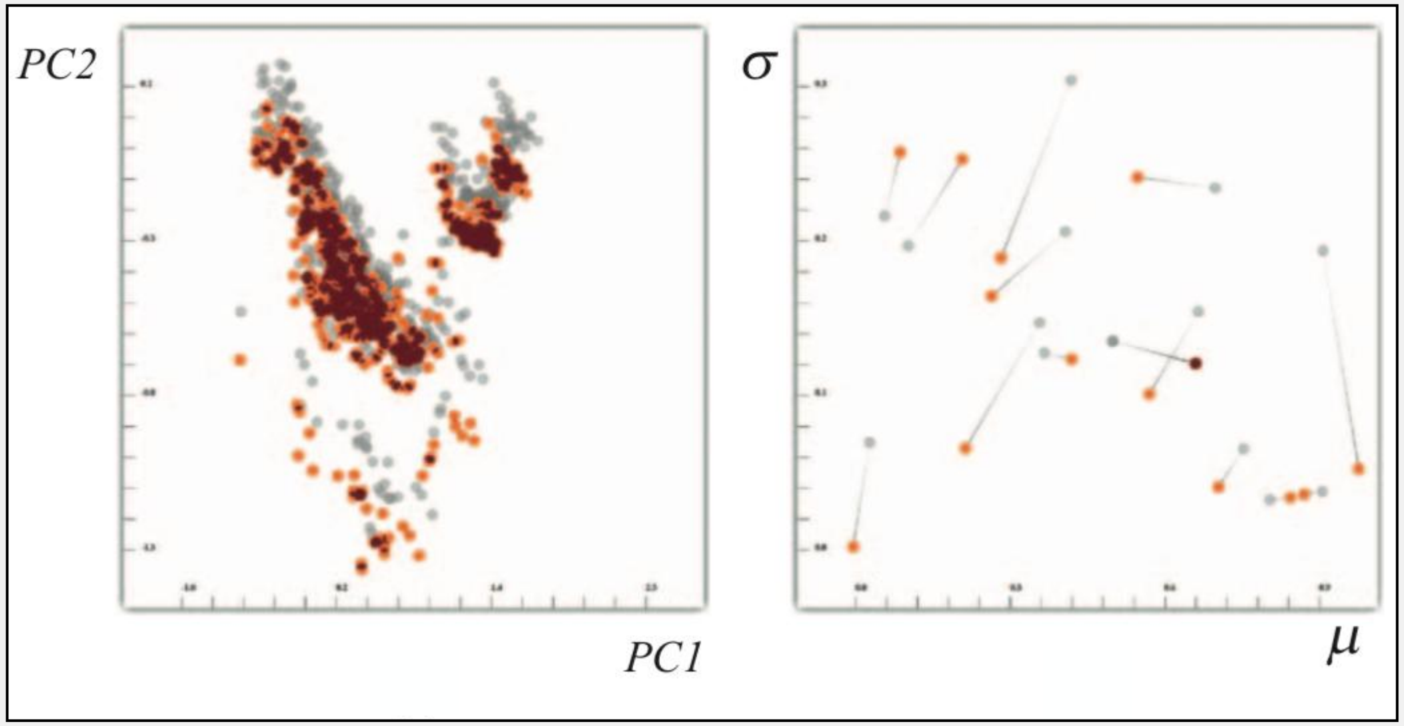


Brushing Dimensions



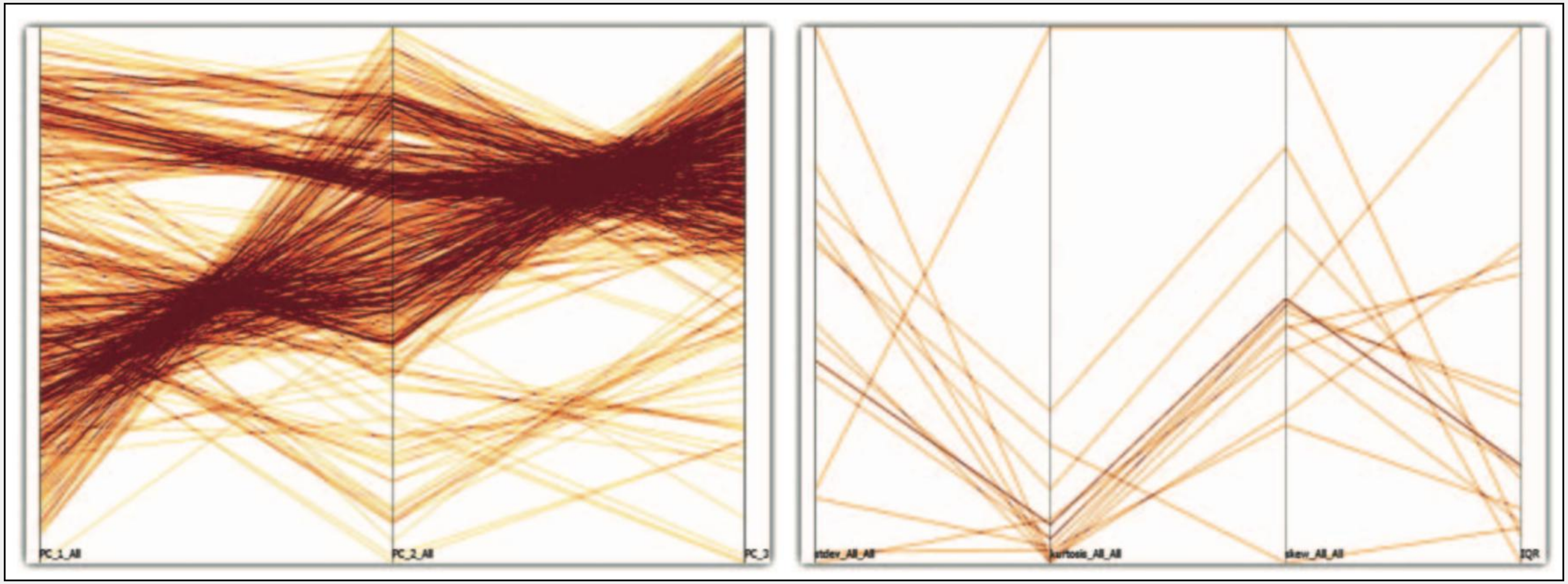
Focus+Context

- Recomputed values: in **focus**
- Prior values: remain in **context**

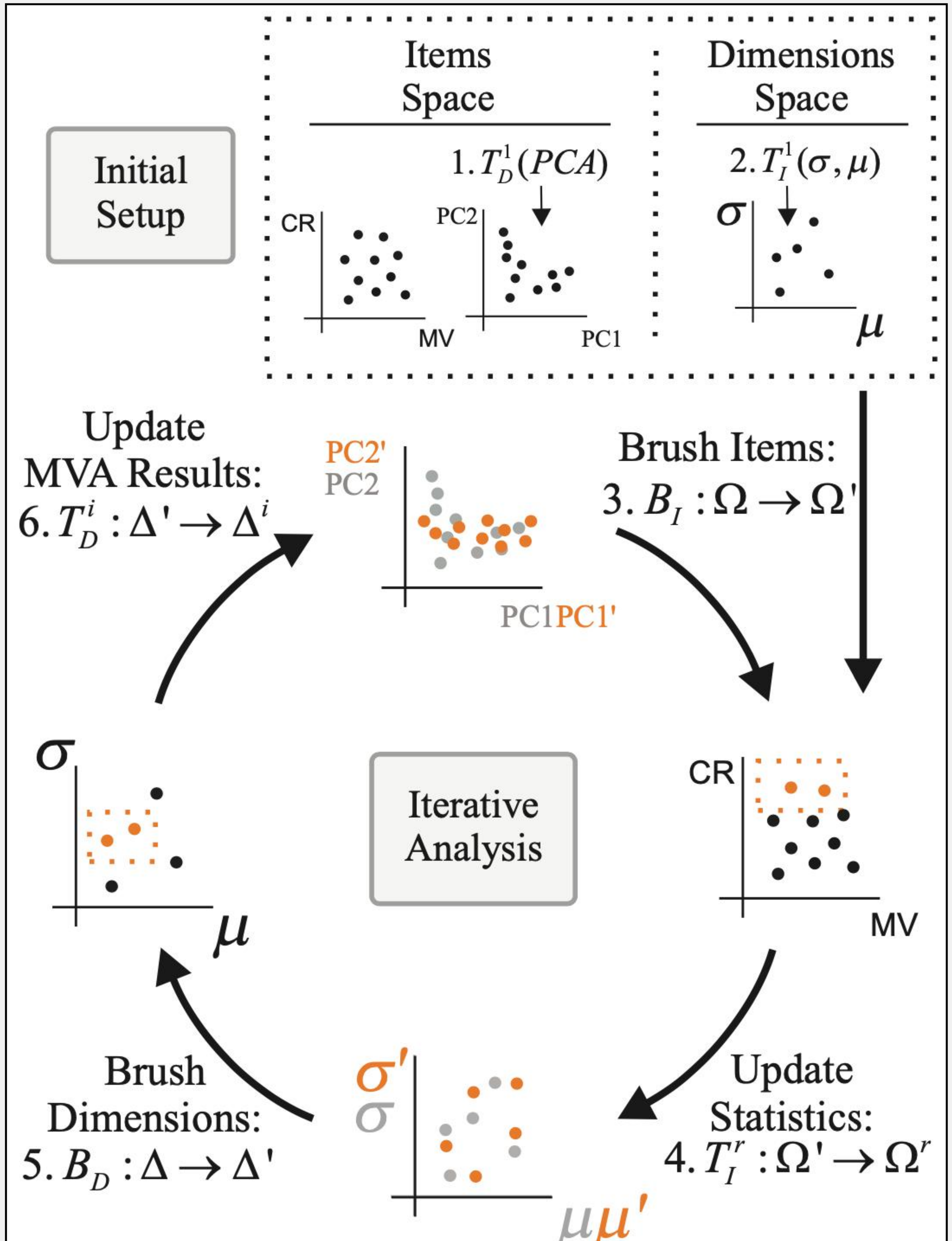


- Other options:
  - instantaneous change
  - animation
  - What are trade-offs?

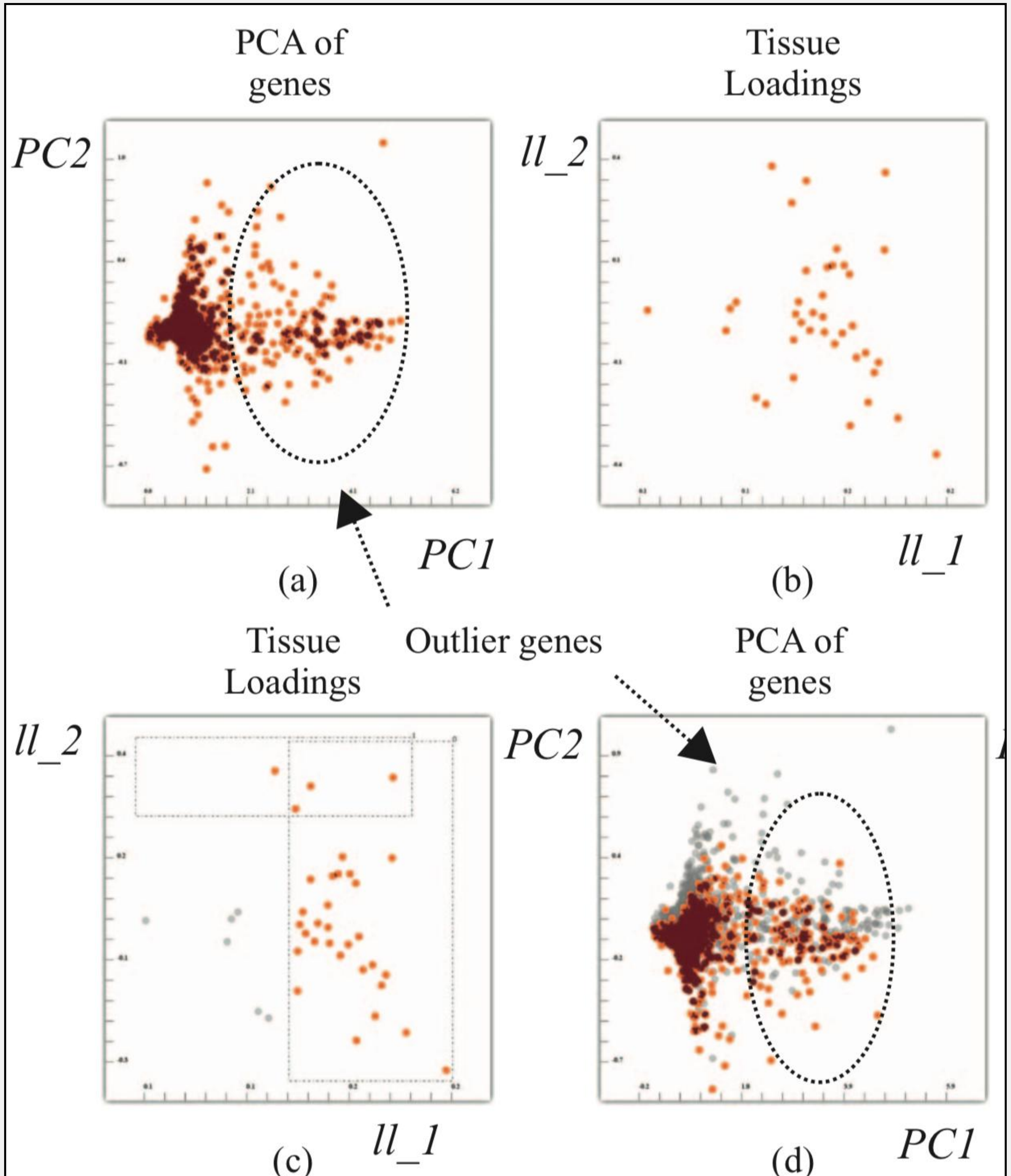
Dimension Space



Full Workflow



Example: Microarray Data



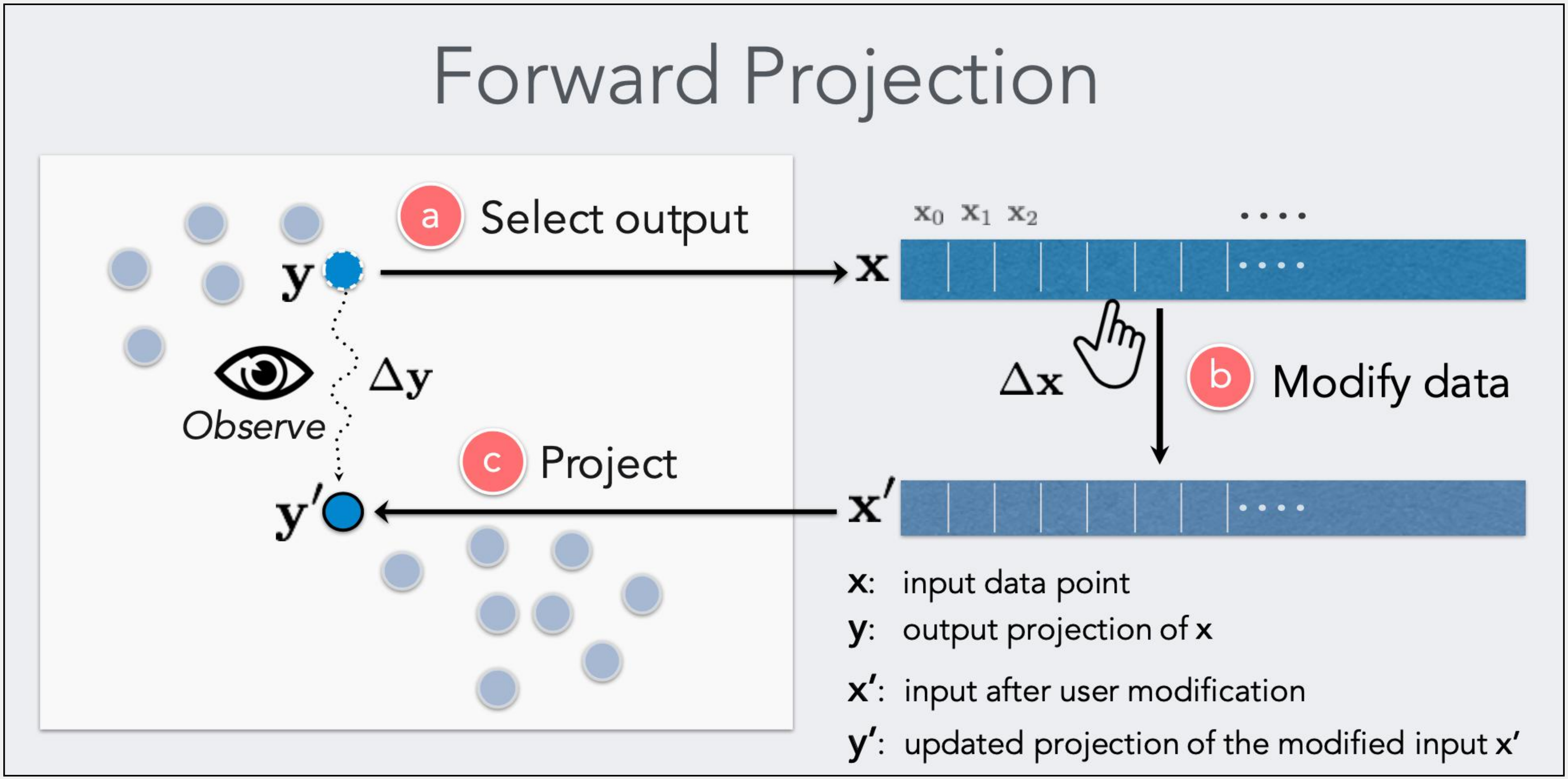
- Data:
    - item: gene
    - dimension: tissue sample
- Statistics: PCA loadings (eigenvectors weighted by eigenvalues)
- High loadings: indicate tissues that are important, *leads to reduction in outliers*.

Dimensionality Reduction: What-ifs

- What if I perturb an input point?
  - What happens to the projection?
- What if I perturb a projected point?
  - What does this correspond to in the input?
- Forward and Inverse Projections

[Cavallo & Demiralp 2018]

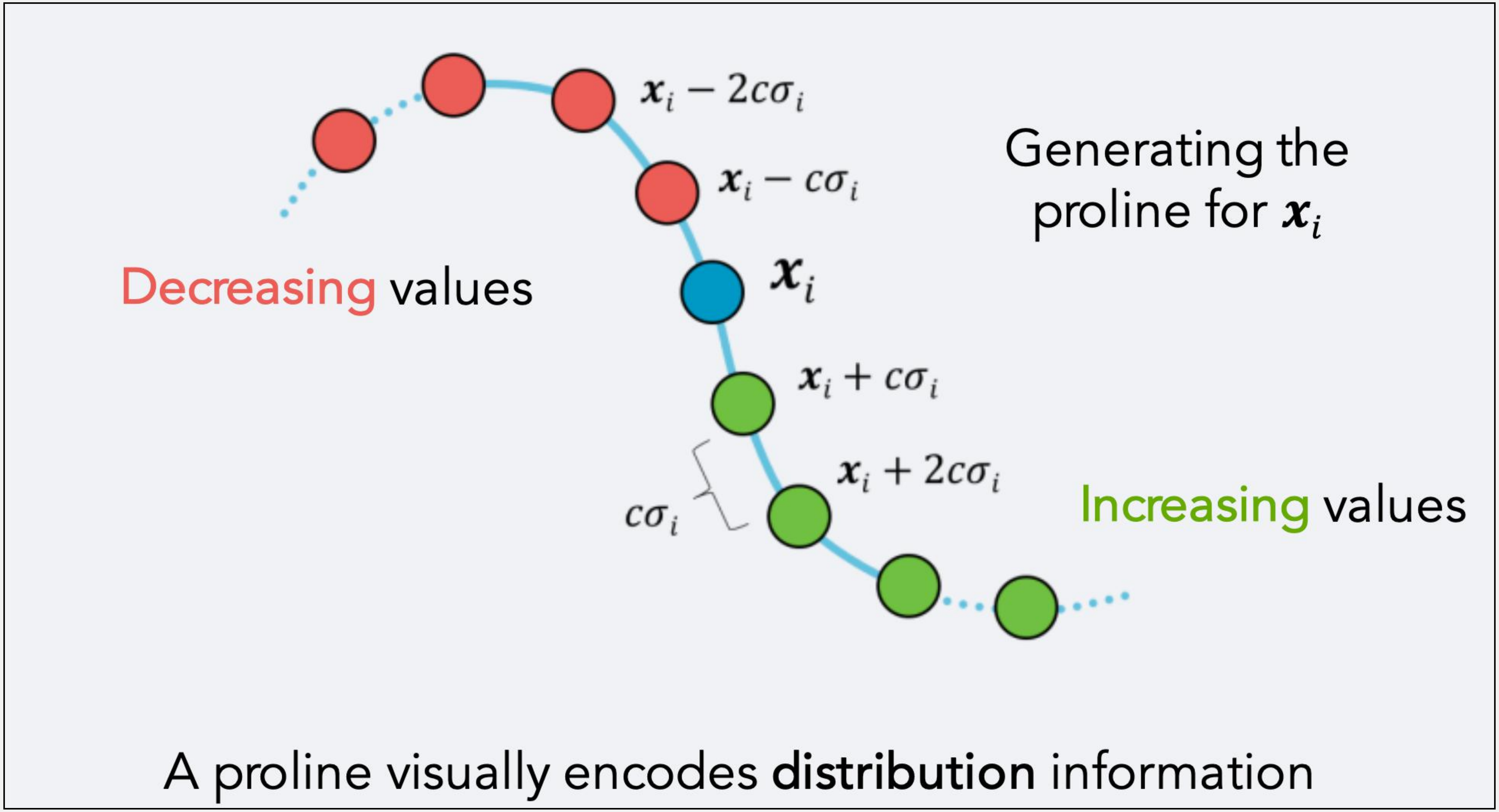
Forward Projection



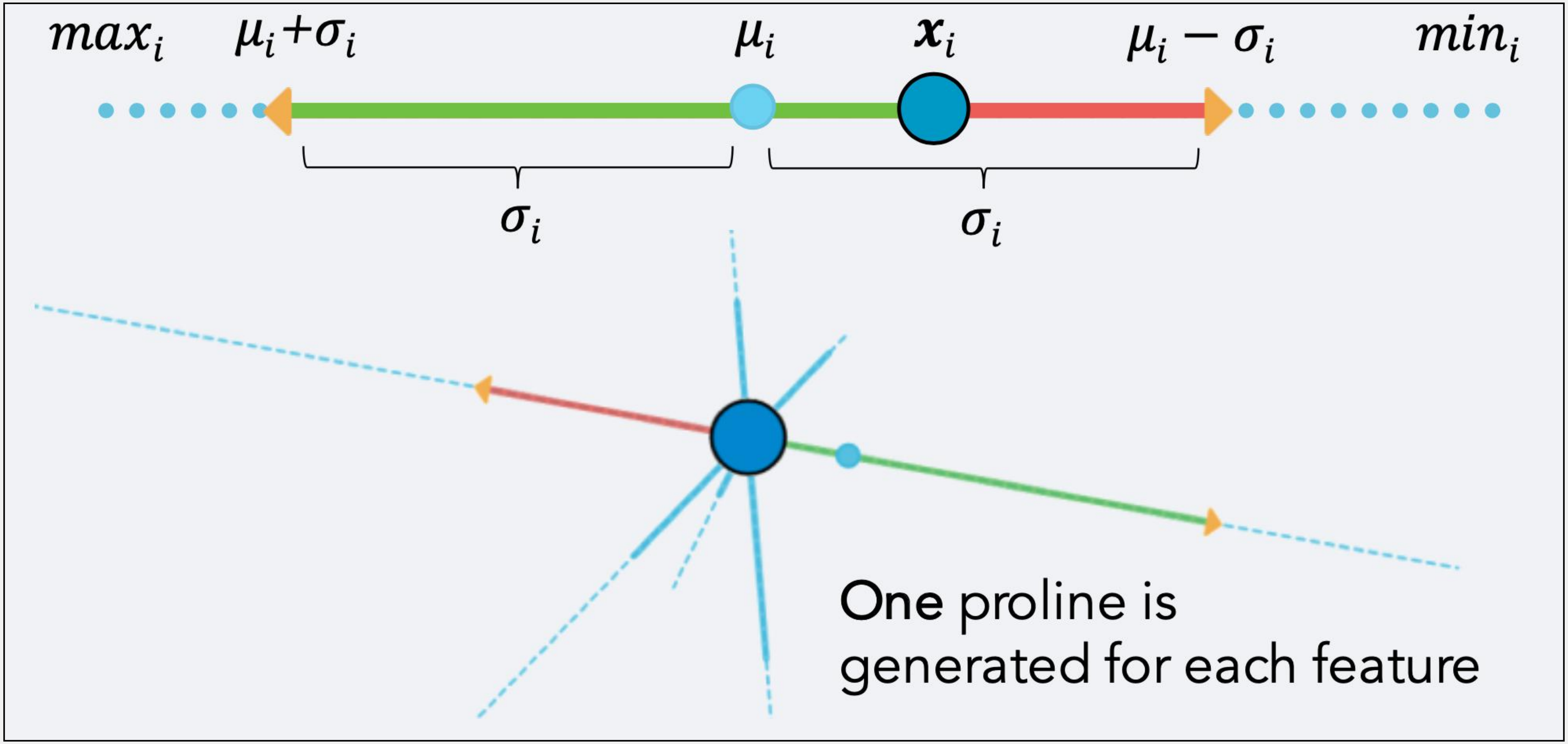
Forward What-ifs

- Don't want the user to manually specify perturbations!
- Instead: sample each attribute, associate each attribute with a curve (PCA: line) that corresponds to attribute range.
    - Need: min, max, mean, standard deviation
    - Uniformly sample between min & max using standard deviation

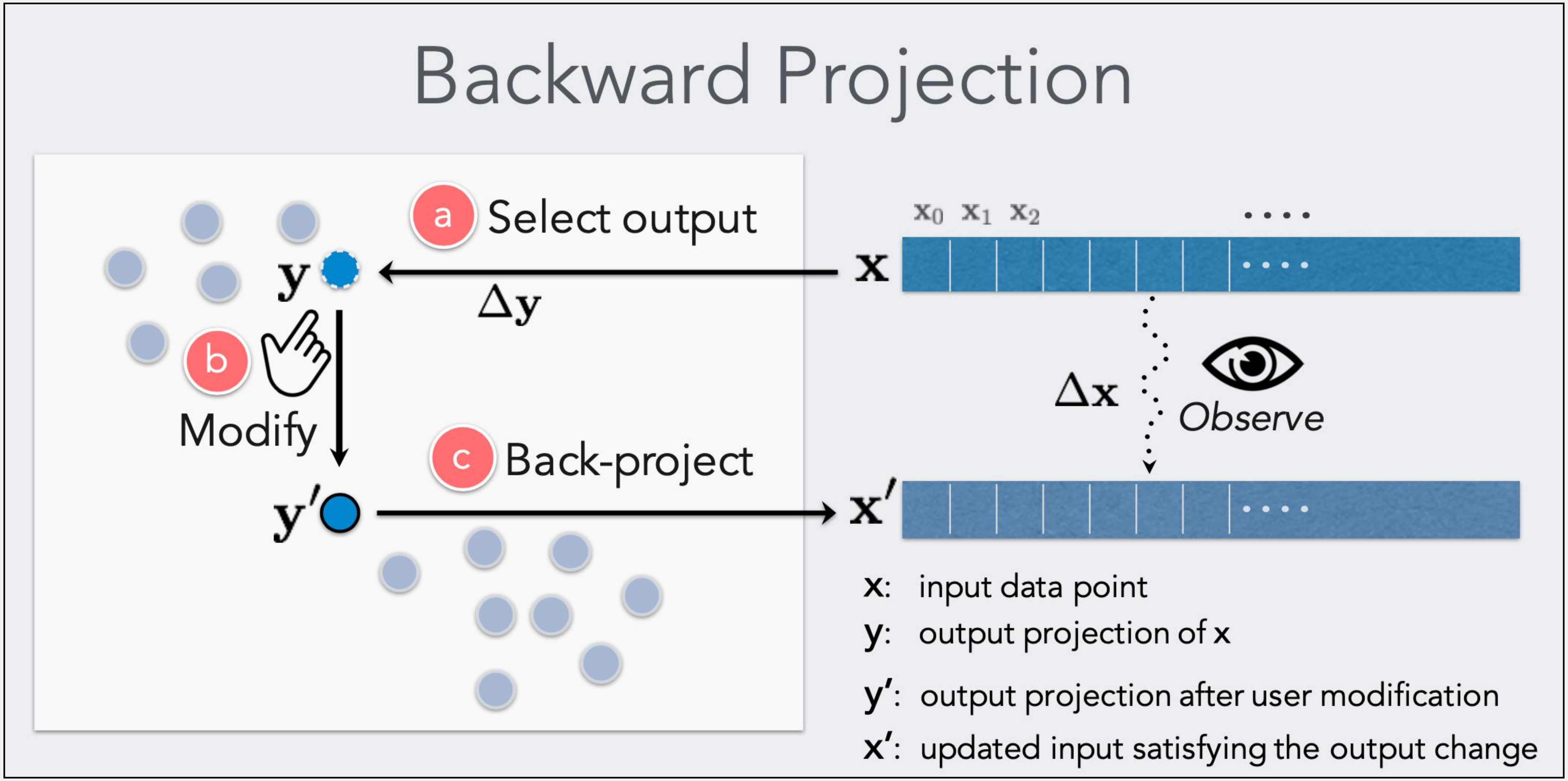
Forward Projection: Sampling



Forward Projection: Prolines



Backward Projection



Unconstrained Projection

- Problem: find a high-dimensional point that, upon projection, maps to a given low-dimensional point.
- Ill-posed! Many such points could satisfy this.
- Least-norm: $\min \|\mathbf{x}\|$ **s.t.** $P\mathbf{x} = \mathbf{y}$ $\quad P \in \mathbb{R}^{2xd}$
- The solution: $\mathbf{x}^* = P^T(PP^T)^{-1}\mathbf{y}$
- Why? $P(\mathbf{x} - \mathbf{x}^*) = \mathbf{y} - (PP^T)(PP^T)^{-1}\mathbf{y} = 0$

$$
\begin{aligned}
(\mathbf{x} - \mathbf{x}^*)^T\mathbf{x}^* &= (\mathbf{x} - \mathbf{x}^*)^T P^T(PP^T)^{-1}\mathbf{y} \\
&= (P(\mathbf{x} - \mathbf{x}^*))^T(PP^T)^{-1}\mathbf{y} \\
&= 0
\end{aligned}
$$

$$\|\mathbf{x}\|^2 = \|\mathbf{x}^* + \mathbf{x} - \mathbf{x}^*\|^2 = \|\mathbf{x}^*\|^2 + \|\mathbf{x} - \mathbf{x}^*\|^2 \geq \|\mathbf{x}^*\|^2$$

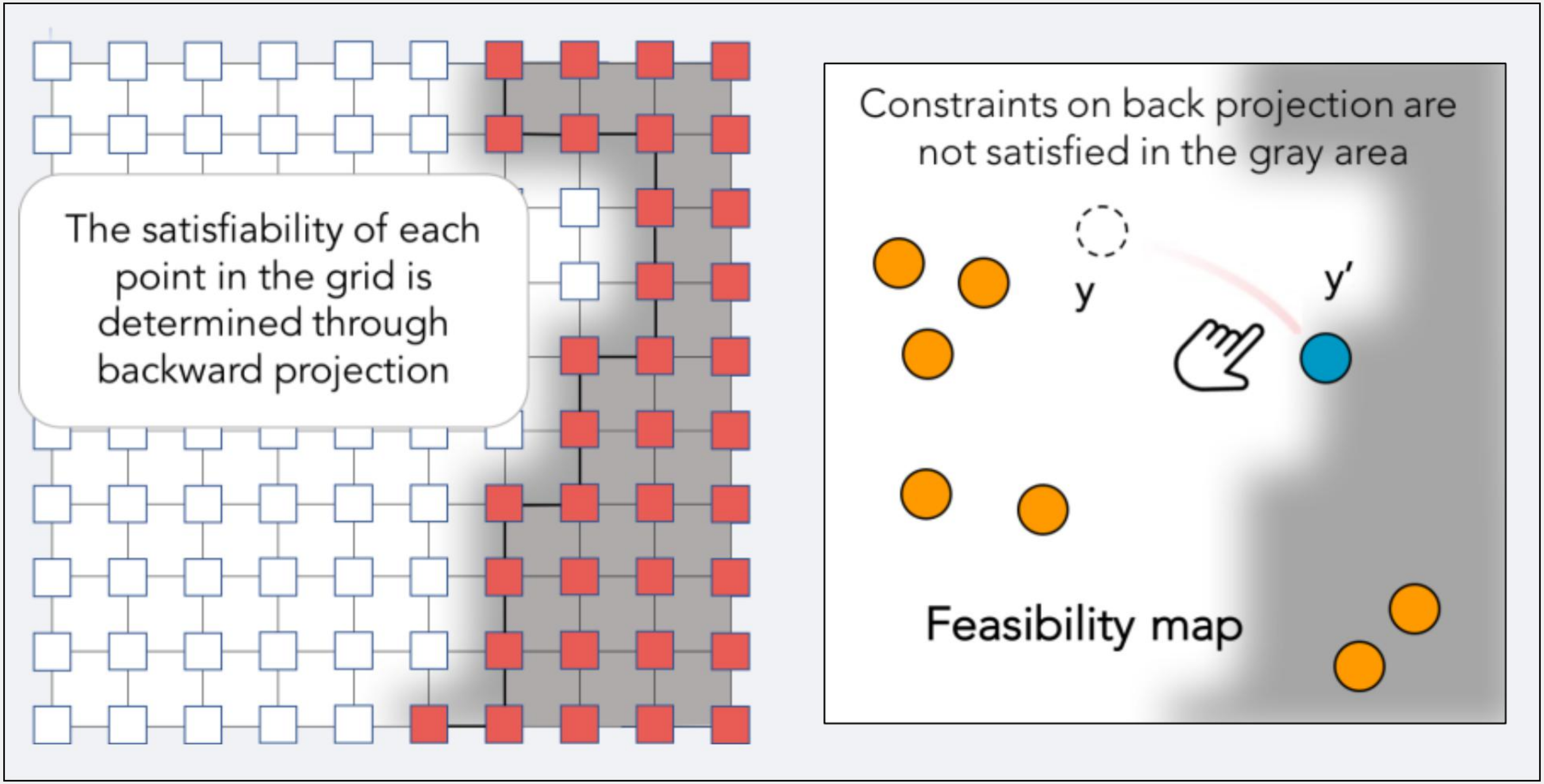
$$(\|\mathbf{a} + \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\mathbf{a}^T\mathbf{b})$$

PCA Pseudoinverse

- Eigenvectors are orthogonal! No need to explicitly perform inversion:

$$\mathbf{x}^* = P^T(PP^T)^{-1}\mathbf{y} = P^T\mathbf{y}$$

Backward Projection: Feasibility



Feasibility: Constrained Projection

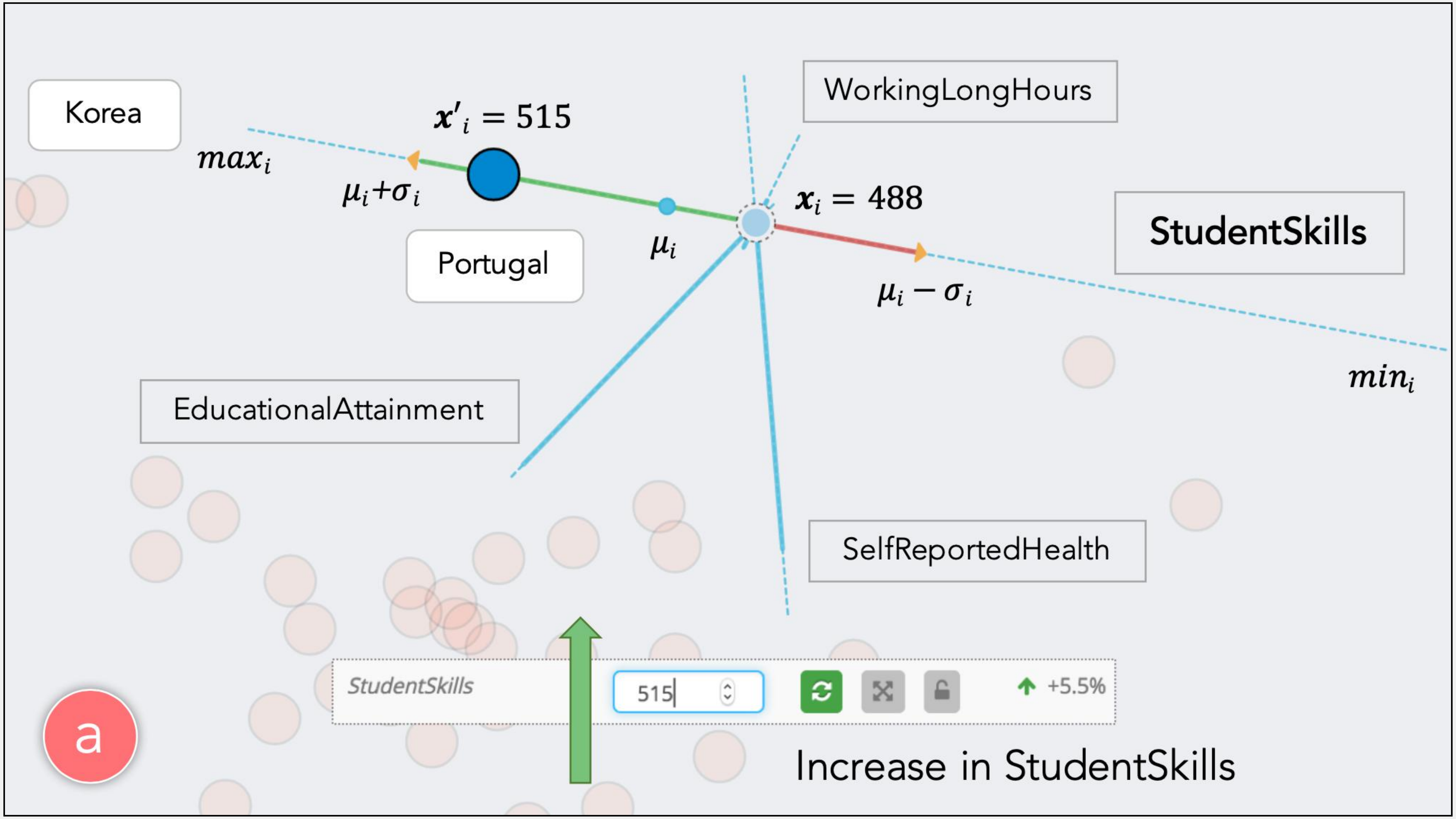
- Linear subspace spanned by orthogonal eigenvectors does not help us out in this case.

- Need to solve the following:

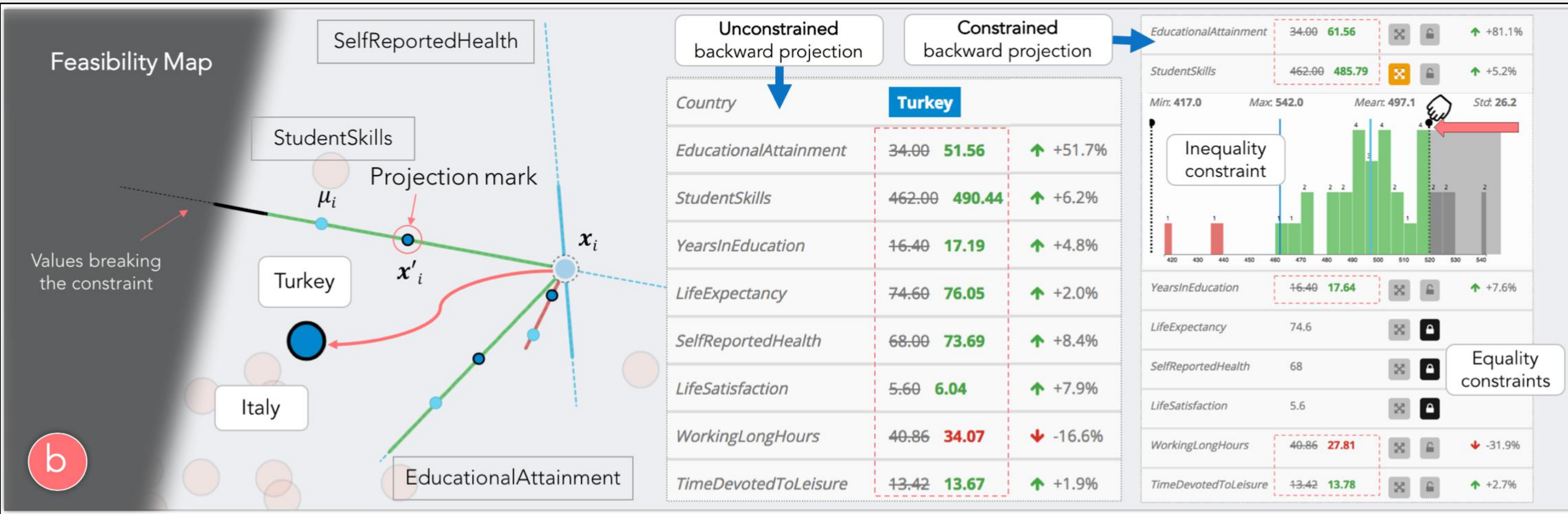
$$\min_{\mathbf{x}} \|P\mathbf{x} - \mathbf{y}\|^2 \quad \textbf{s.t.} \quad C\mathbf{x} = \mathbf{d} \text{ , } \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$$

- Further implications: user needs to set constraints, so we need a way to visualize the original, high-dimensional data.

Forward Projection Example

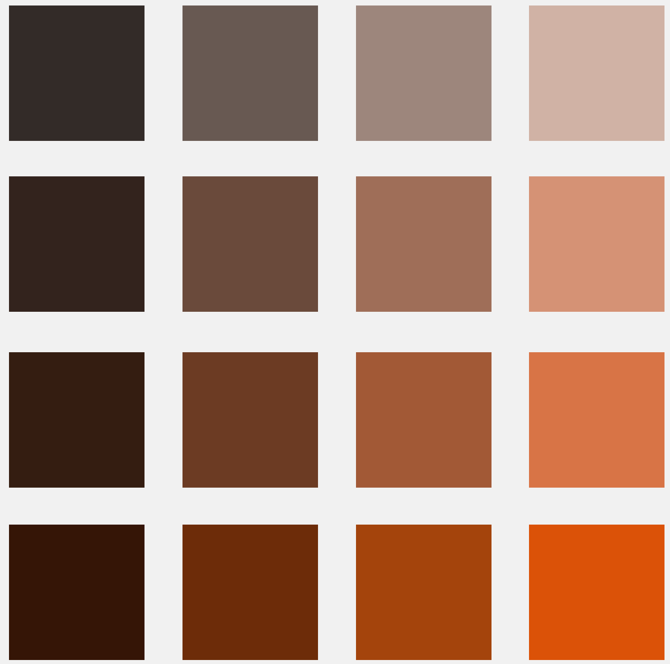


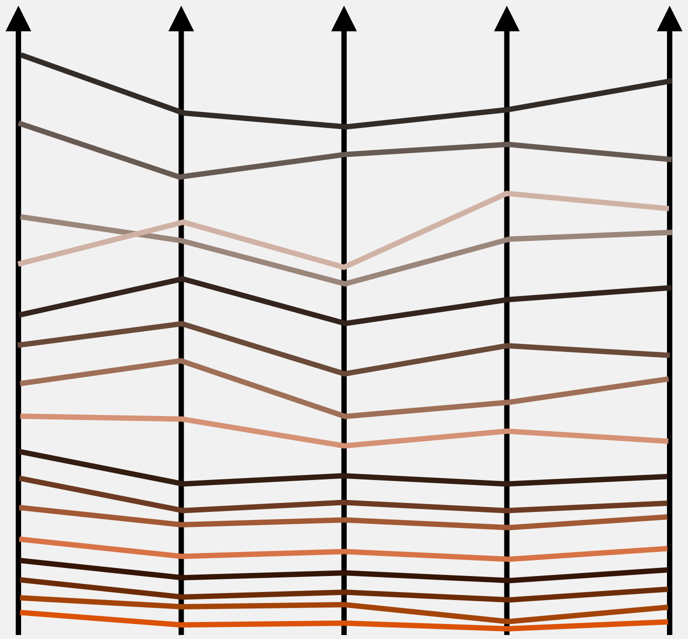
Backward Projection: Example



Alternative Design: Unconstrained Projection

- Showing tabular data directly is ok. Alternatives?





Steering Computational Resources

- Multidimensional Scaling
- Input: set of pairwise distances between points
- Output: a projection whose Euclidean distance best matches the input pairwise distance
- Typical objective: stress

$$Stress = \frac{\sum_{i<j}(d_{ij} - p_{ij})^2}{\sum_{i<j} p_{ij}^2}$$

MDS: Types

- Classic Multidimensional Scaling: Euclidean distances
  - Can be solved as an eigenvalue problem
- Metric Multidimensional Scaling
  - Function of Euclidean distances: often solved via stress majorization, or just gradient descent
- Nonmetric Multidimensional Scaling
  - We are not given distances, but just general dissimilarities

MDS Challenges

- Classic Multidimensional Scaling: Euclidean distances
  - Can be solved as an eigenvalue problem
- Metric Multidimensional Scaling
  - Function of Euclidean distances: often solved via stress majorization, or just gradient descent
- Nonmetric Multidimensional Scaling
  - We are not given distances, but just general dissimilarities

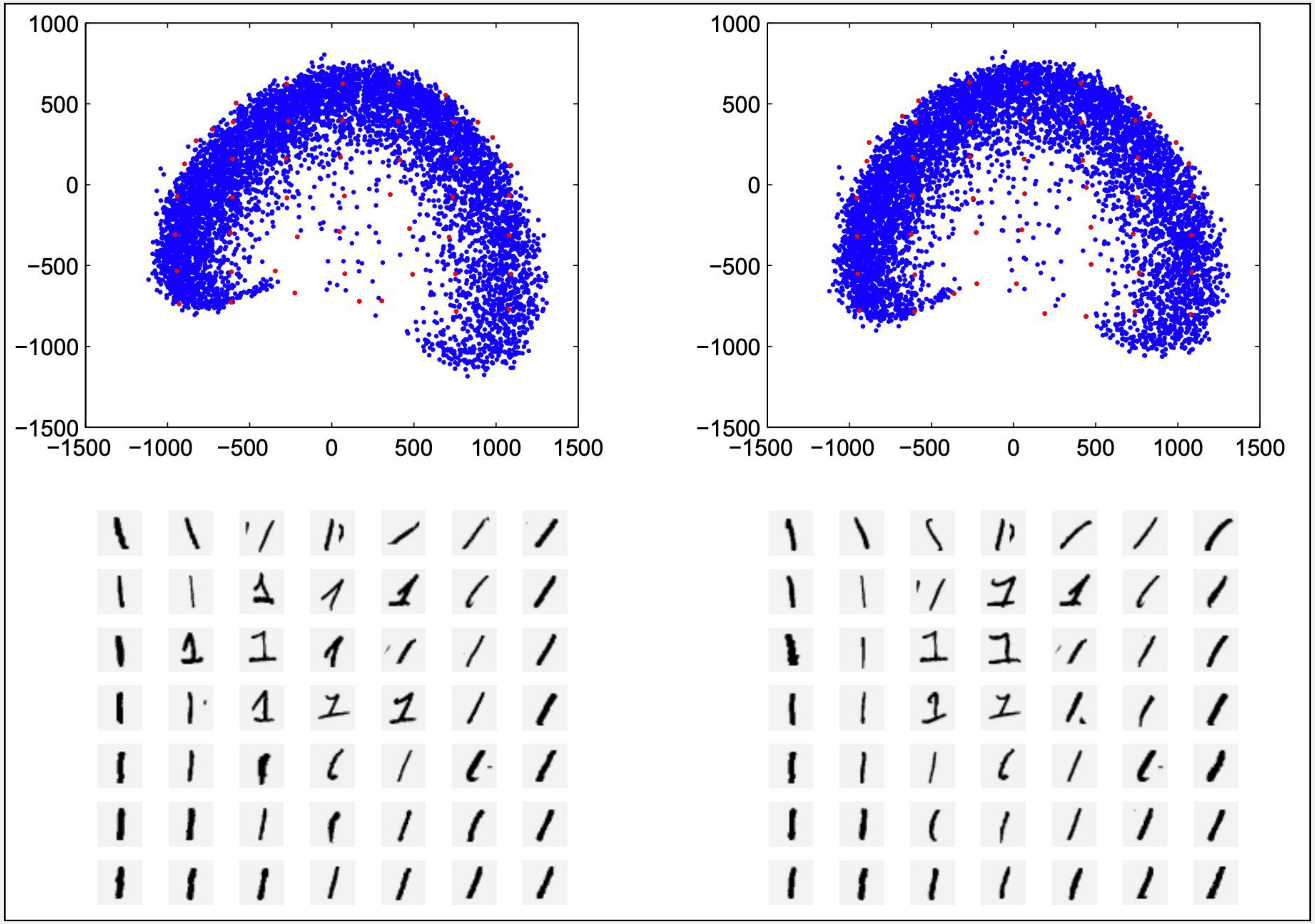
Issues with scalability

Landmark MDS

[de Silva & Tenenbaum 2004]

- Algorithm:
  - Select a small set of landmark points
  - Compute MDS on these landmarks
  - Perform scattered data interpolation to recover the rest of the points

Landmark MDS: Example



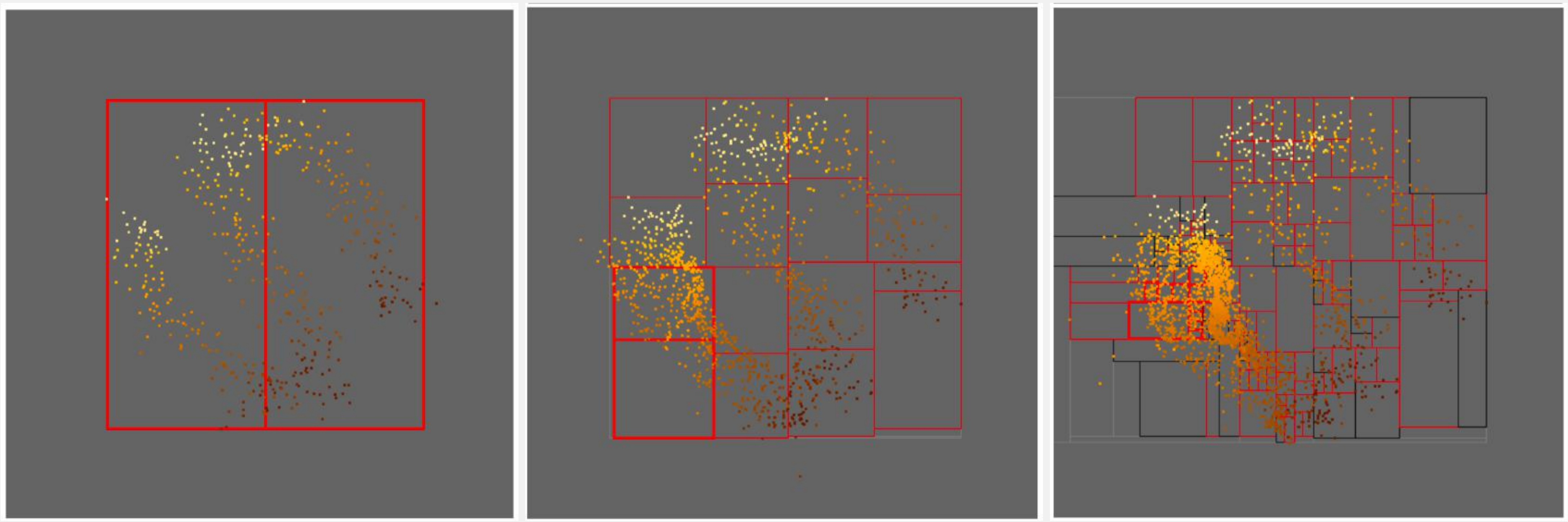
Landmark MDS: Limitations

- How do we select the landmarks?
- In [de Silva & Tenenbaum 2004], this is done automatically via farthest point sampling (or random sampling…).
- How do we have the human involved?

MDSteer

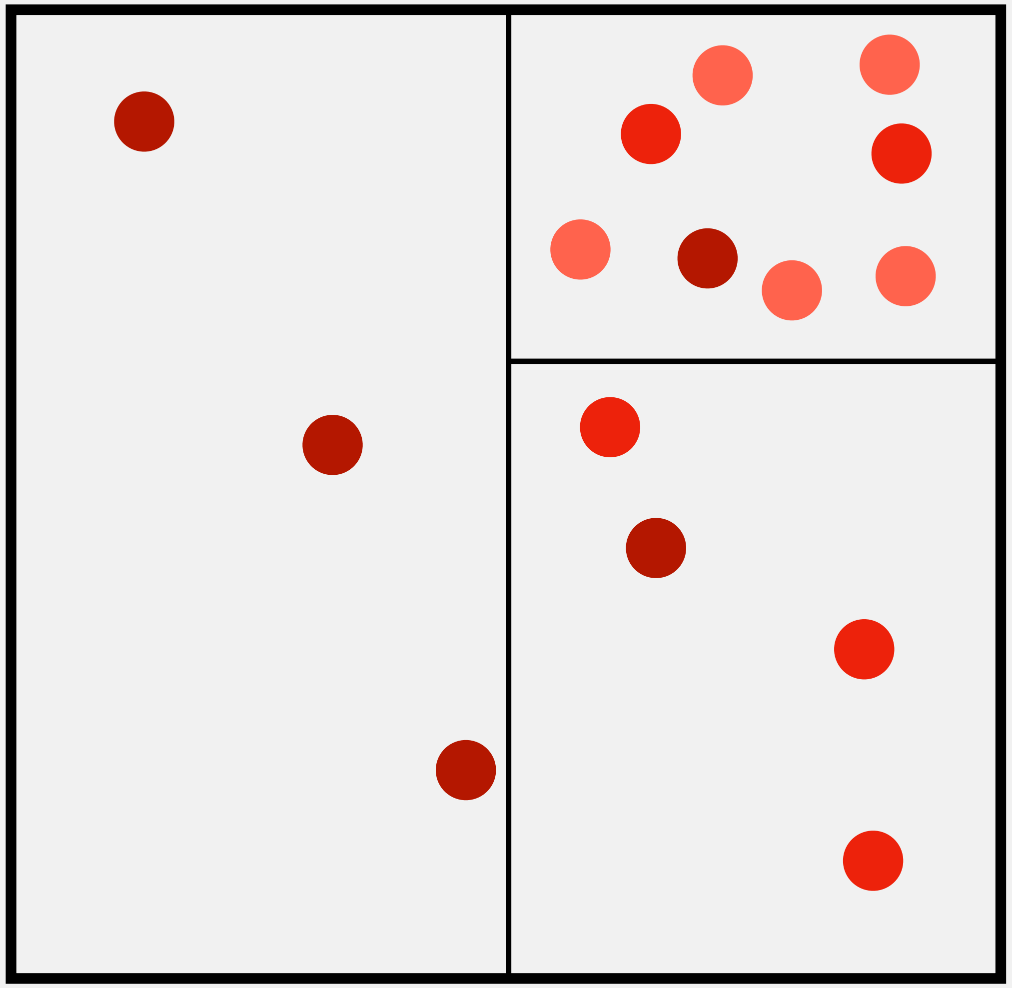
[Williams & Munzner 2004]

- Progressively perform MDS, having the user *steer the points with which to perform optimization*.



MDSteer: Bins

- Progressively subdivide the 2D plane on which the points are to be placed



- User controls what bins to subdivide
- If they do nothing: full MDS
- But, as computation progresses, they can observe, and adjust which bins to subdivide

MDSteer: Example