Steering Dimensionality Reduction

Semantics of DR Steering

- Thus far, DR steering has been rather explicit; largely from *user interface*, to *model update*

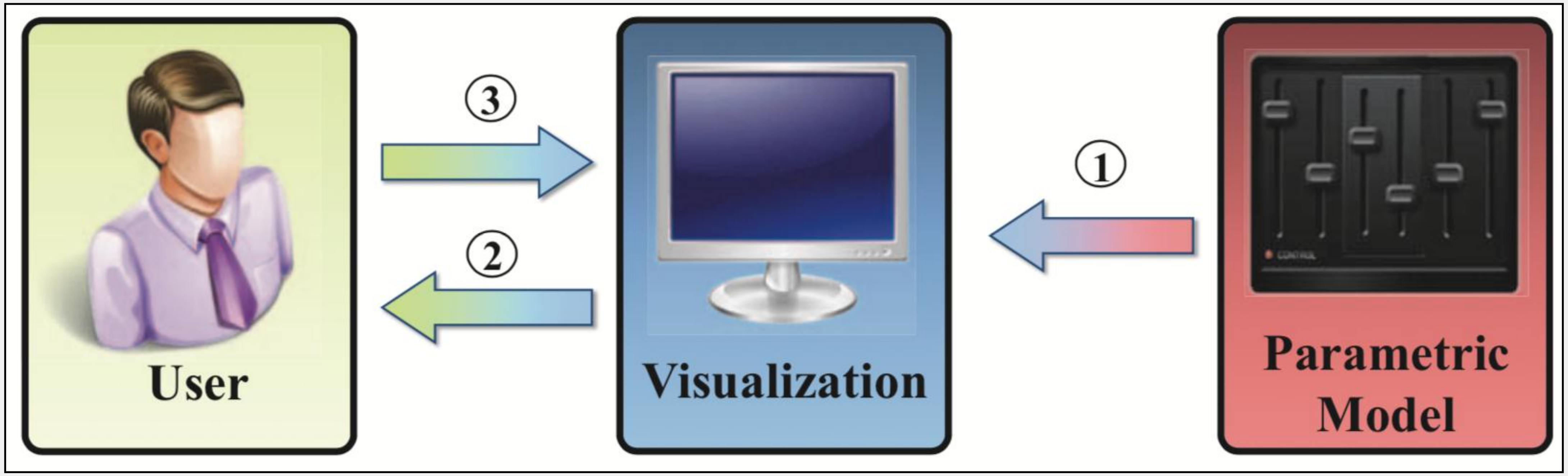
- iPCA: dimension importance

- Dual Brushing: dimension selection

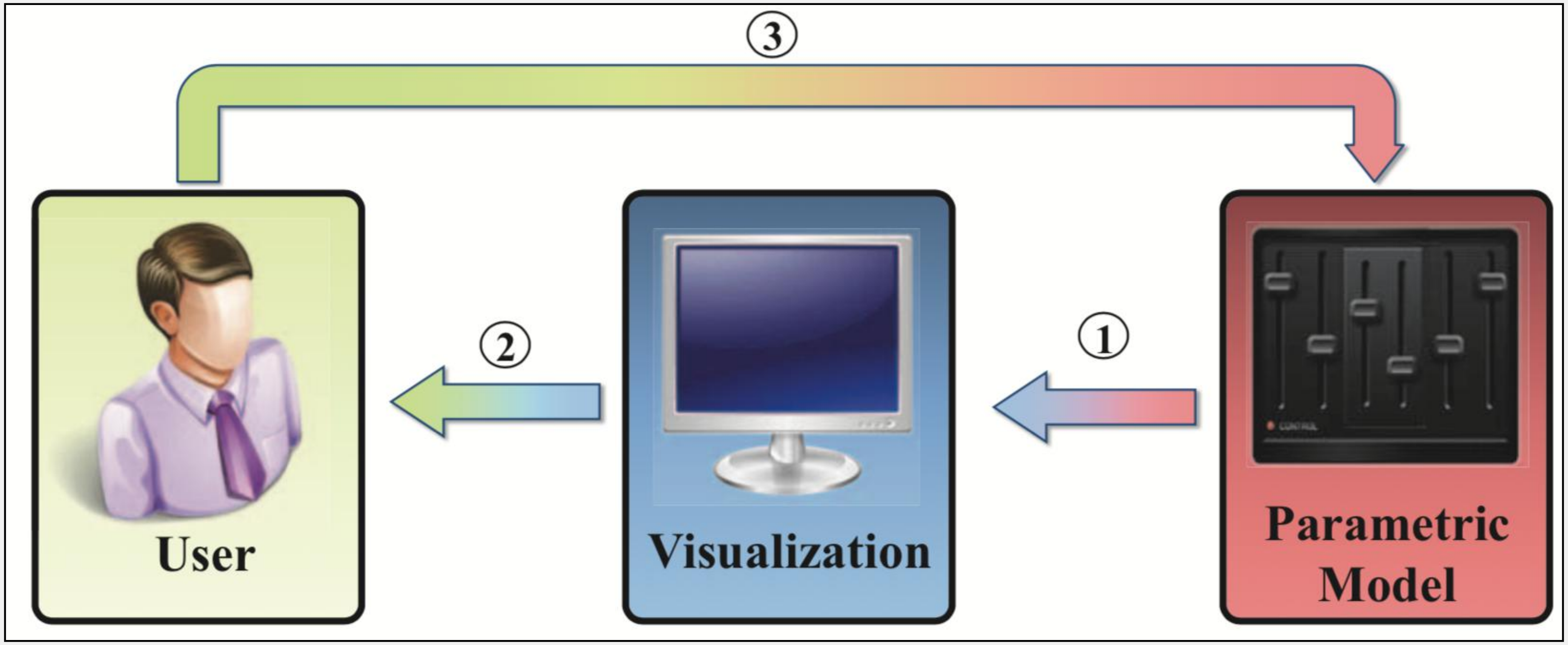
- Let us dig a bit deeper into the ways in which users can interact with DR…

[Hu et al. 2013]

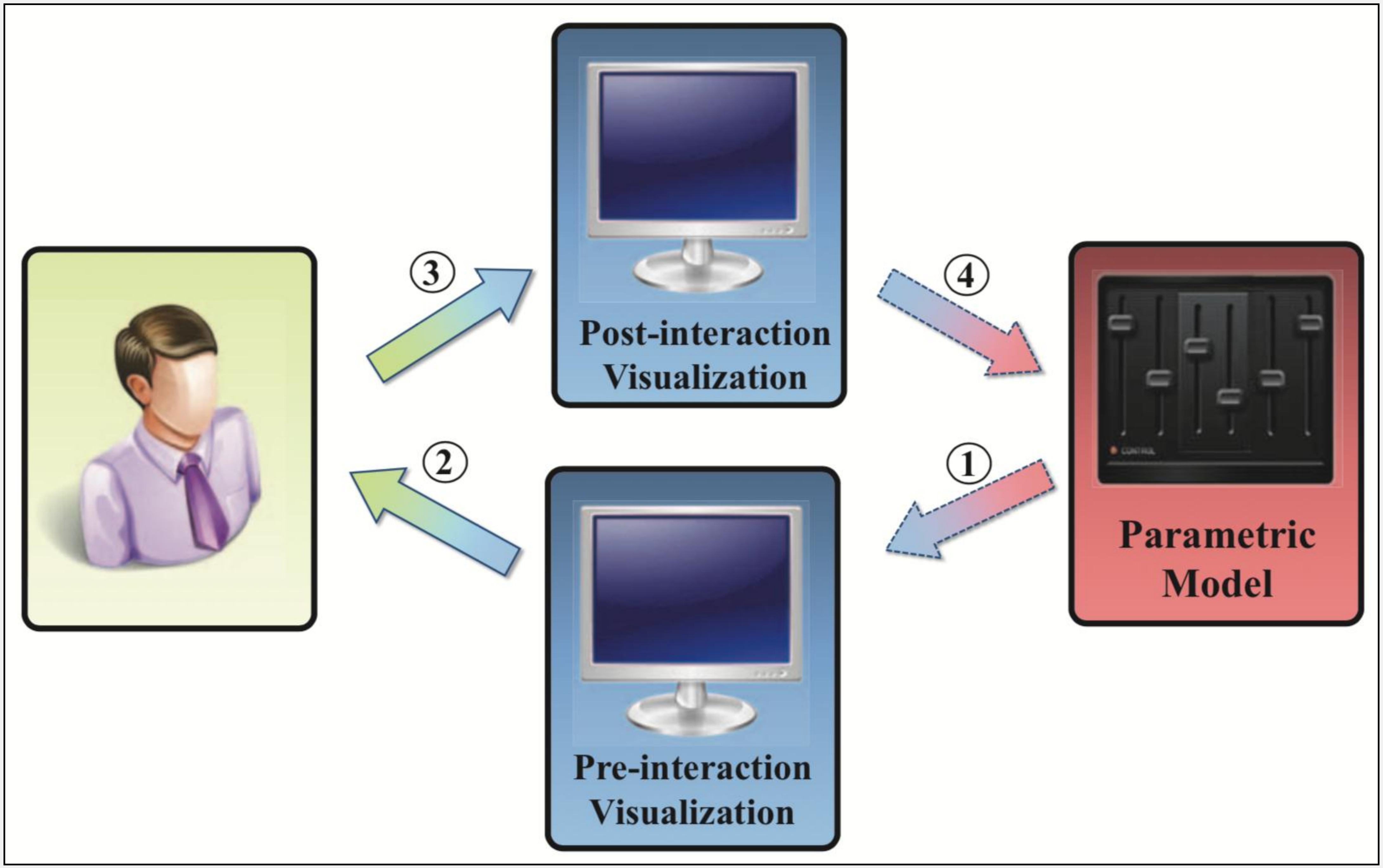
Surface Level Interaction



Parametric Level Interaction



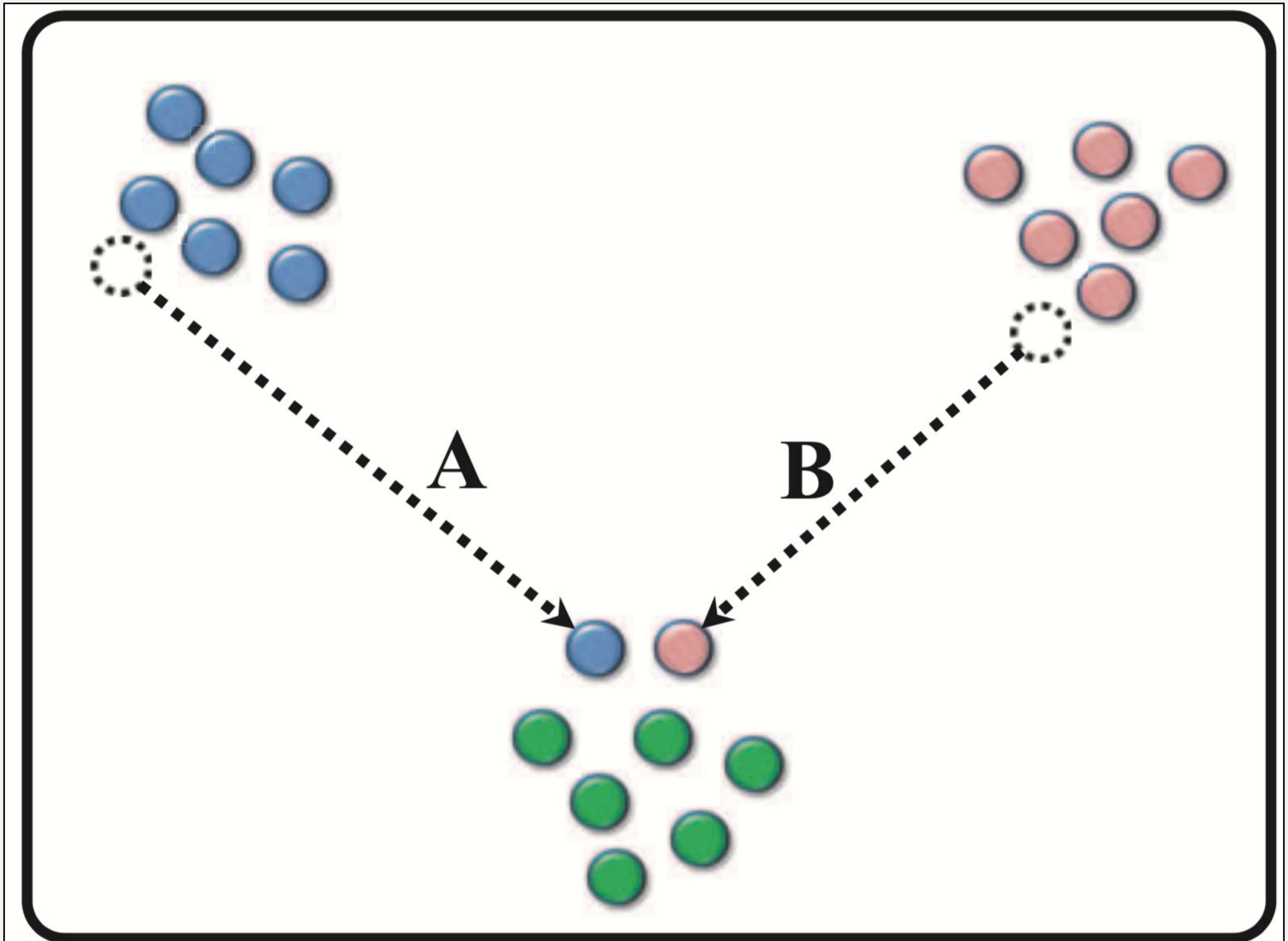
Visual to Parametric Interaction



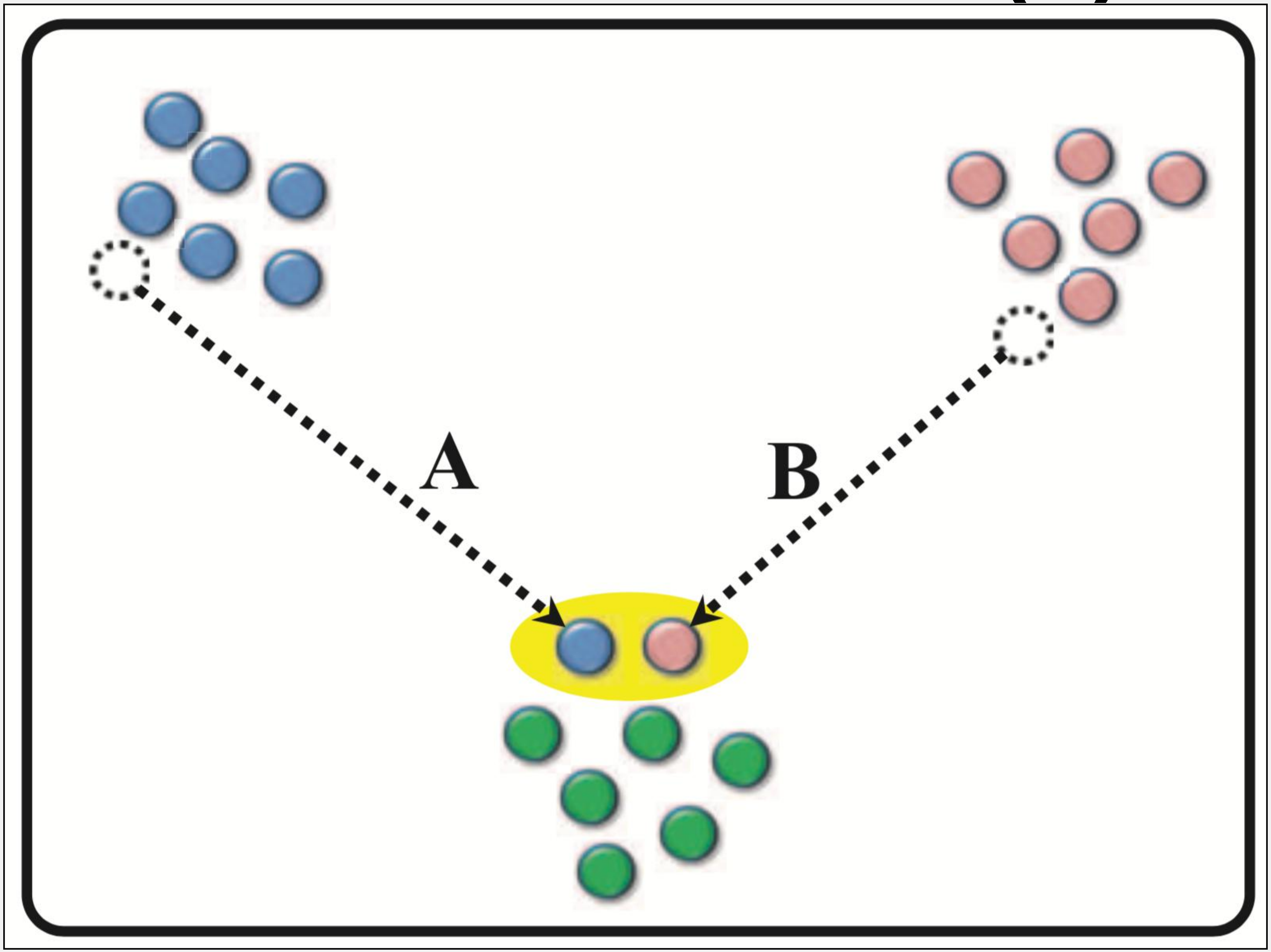
Direct Manipulation for DR

- User *directly* interacts with the visualization to invoke changes to the model
- Note distinction from, e.g., iPCA: there, user tunes sliders (*not* the data vis!)
- Direct manipulation: user explicitly modifies visual channels, and this should imply *something* about their intent for the underlying model
- But what is that *something*? Invert the visual channel mapping (e.g. back to data)?

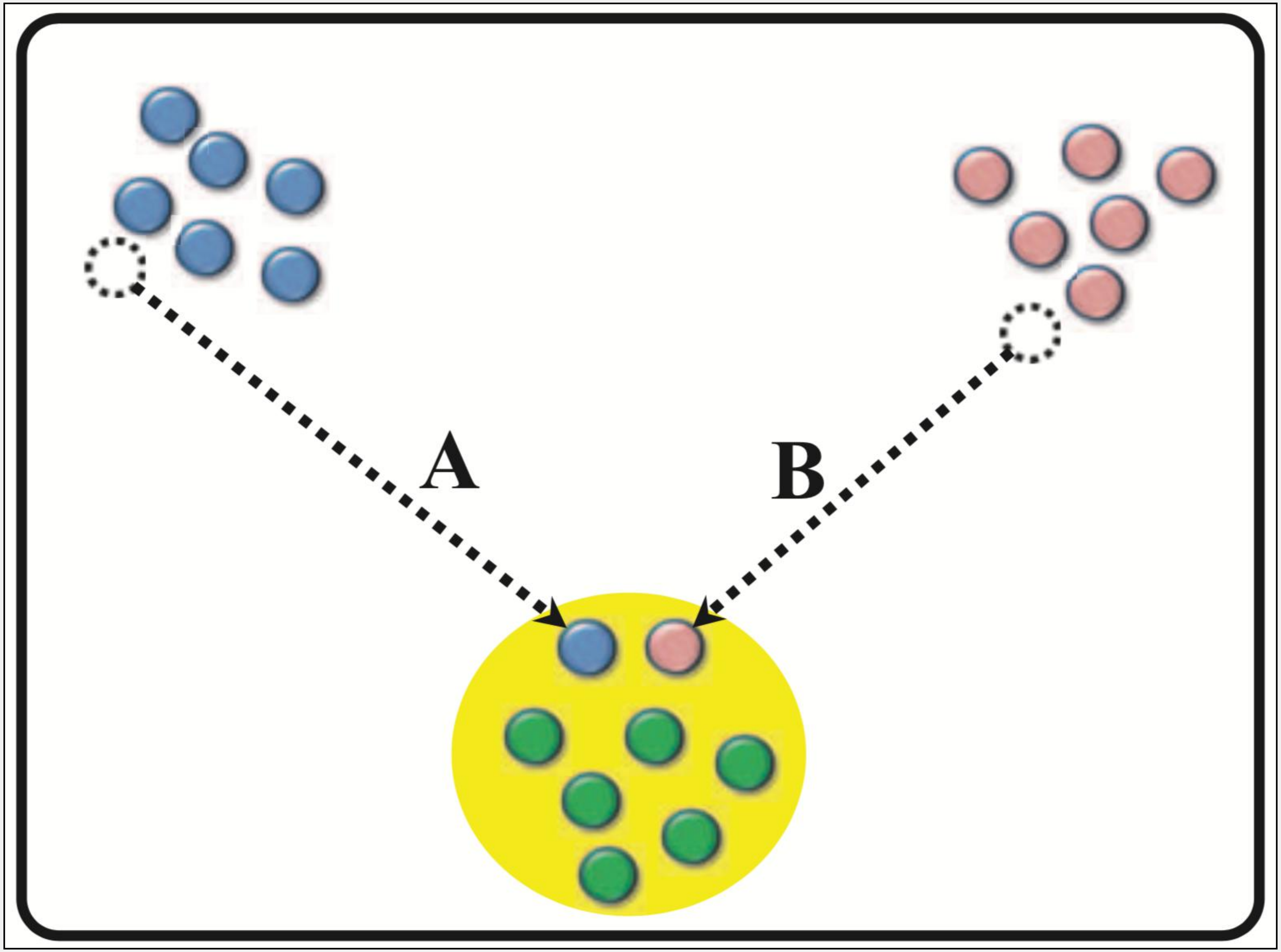
Direct Manipulation



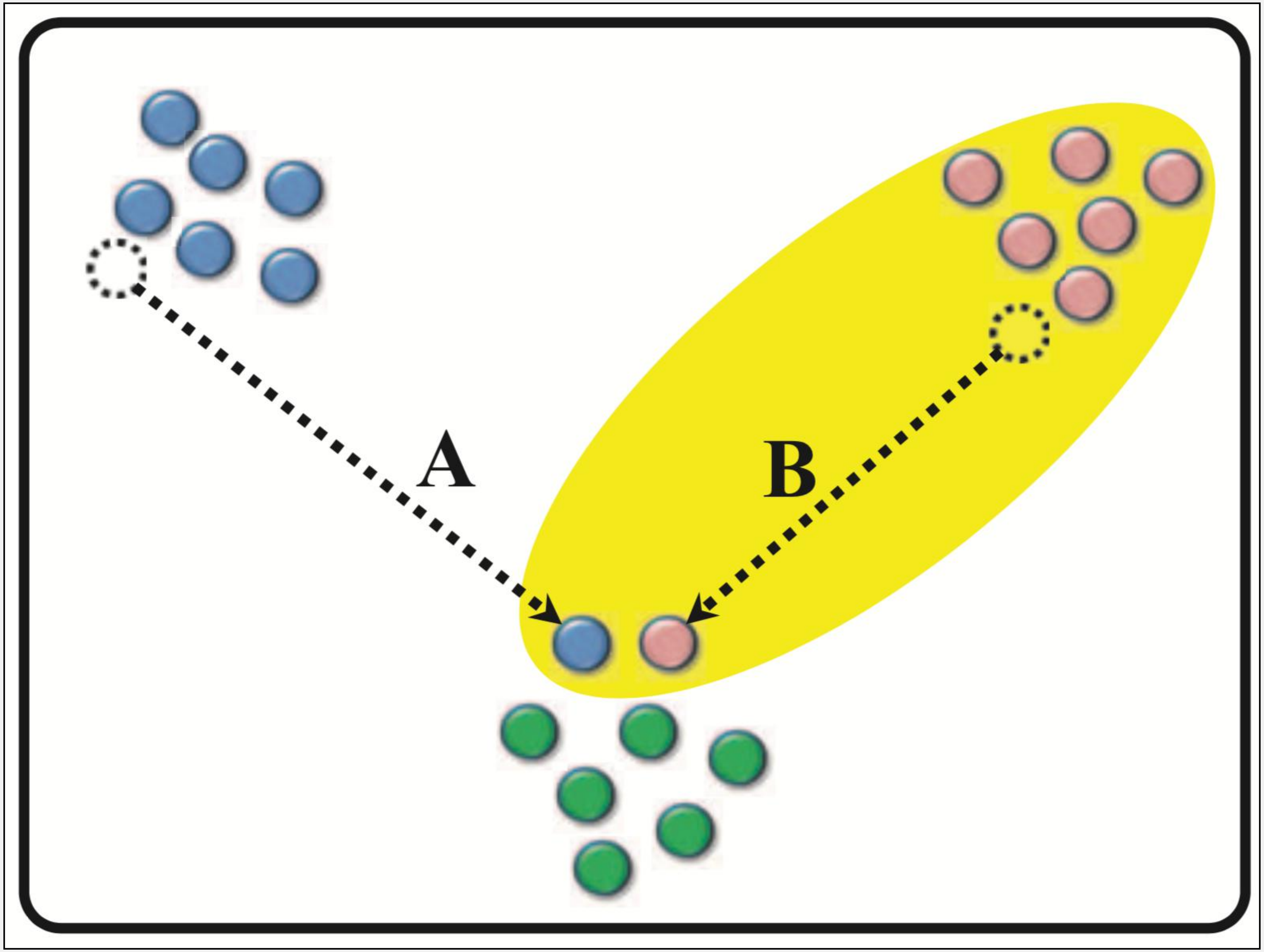
User Intention (1)



User Intention (2)



User Intention (3)



Semantics of Spatialization

- Two key characteristics we want to preserve in visual analytics:
  - Visualization's exposed interactions should support user's natural mode of communication
  - Result of interaction should be consistent with user expectations
- For now: assume we have a semantically meaningful interaction; can we translate this to just *any* model?

From User Interactions to Model Updates

- Dimensionality reduction technique: **multidimensional scaling** (MDS)

- Inputs: $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| \quad , \quad \mathbf{x} \in \mathbb{R}^d$ (could also be geodesics, which leads to **Isomap** [Tenenbaum et al. 2000])

- To solve: $\delta_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\| \longrightarrow$ **stress** $= \min_{\mathbf{Y}} \sum_{i<j} (D_{ij} - \delta_{ij})^2 \quad , \quad \mathbf{Y} \in \mathbb{R}^{n \times 2}$

- Traditional MDS: solved via matrix factorization (eigenvalue problem) by performing a “double centering” of distance matrix - convert distances to inner products

$$\mathbf{y}_i^T \mathbf{y}_j = \frac{1}{2}\left(\bar{d}_i + \bar{d}_j + \frac{1}{n}\sum_{k=1}^{n} \bar{d}_k - D_{ij}^2\right) = \hat{D}_{ij} \quad , \quad \bar{d}_i = \frac{1}{n}\sum_{k=1}^{n} D_{ik}$$

Weighted MDS

- Assign weights on dimensions, yielding a modified distance:

$$D_{ij}^{M} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T M (\mathbf{x}_i - \mathbf{x}_j)}$$

- Matrix assumed diagonal, interpretation: importance of dimensions in computing distance.
- Modified stress: $\textbf{weighted-stress} = \min_{\mathbf{Y}} \sum_{i<j} (D_{ij}^{M} - \delta_{ij})^2$
- Where do the weights come from? In iPCA, these would be directly specified.
- Here, we solve for them *given the current scatterplot, and user's interactions*.

Workflow

- Compute MDS to obtain a scatterplot.
- User interacts with scatterplot, prescribing weights between pairs of points (to be determined).
- We then solve for a new distance function:

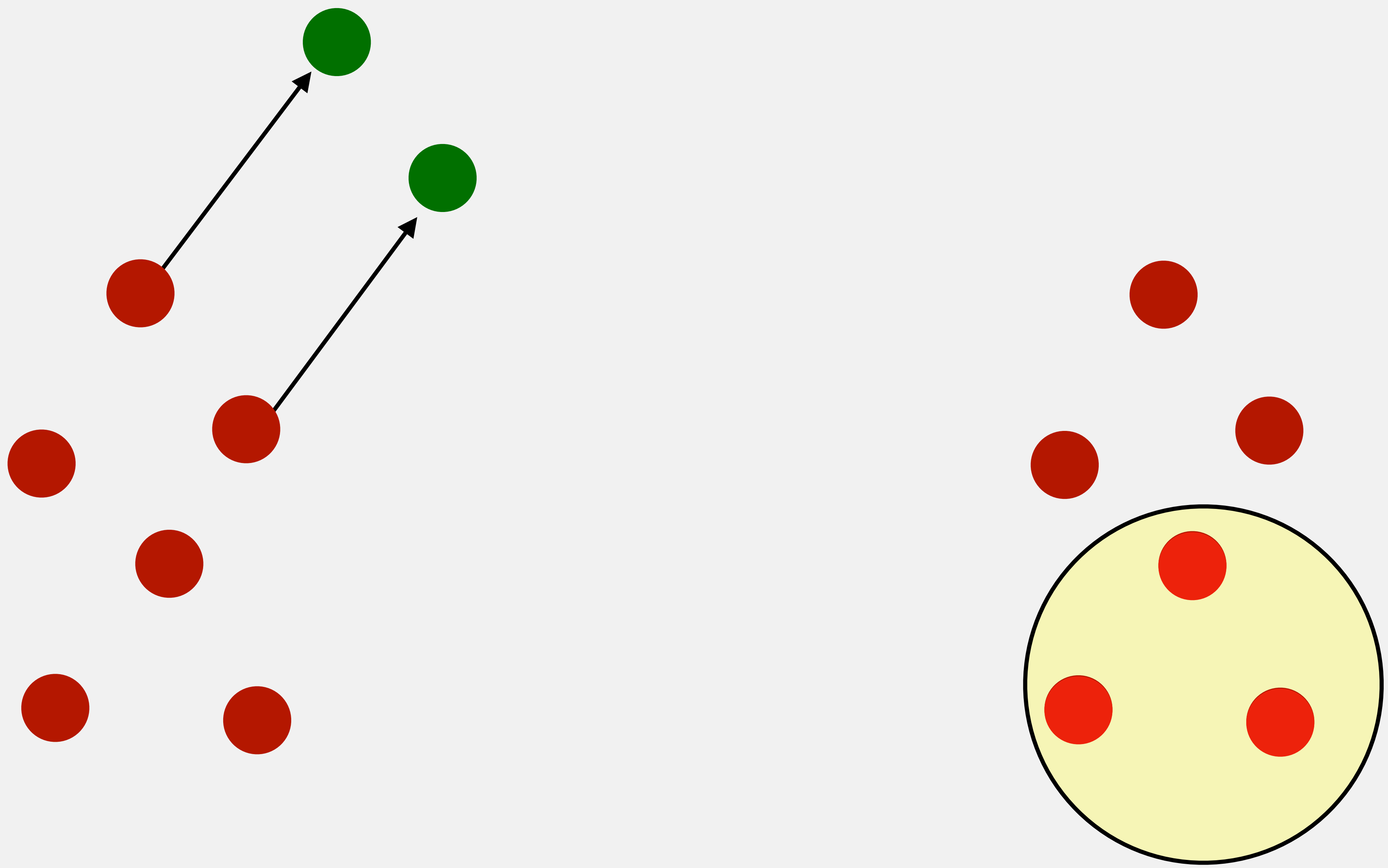
$$\min_{M \geq 0} \sum_{i<j} \alpha_{ij}(D_{ij}^M - \tilde{\delta}_{ij})^2 \quad \text{(may solve via projected gradient descent)}$$

- Reproject the points based on the modified distances:

$$\min_{\mathbf{Y}} \sum_{i<j} (D_{ij}^M - \delta_{ij})^2$$

Linking to User Interactions

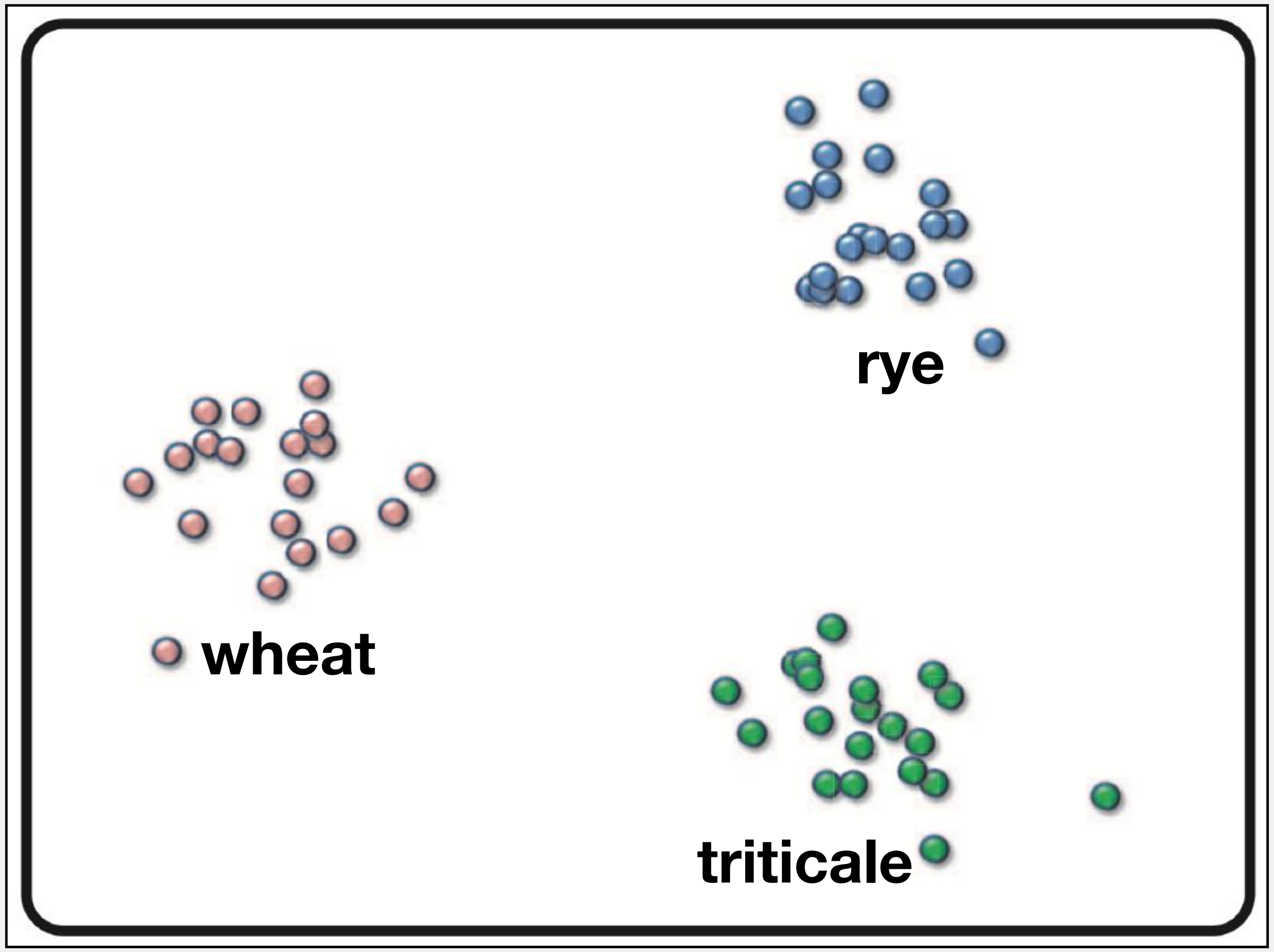
- Two types of interactions: <u>moved points</u>, <u>highlighted points</u>



Deriving Weights

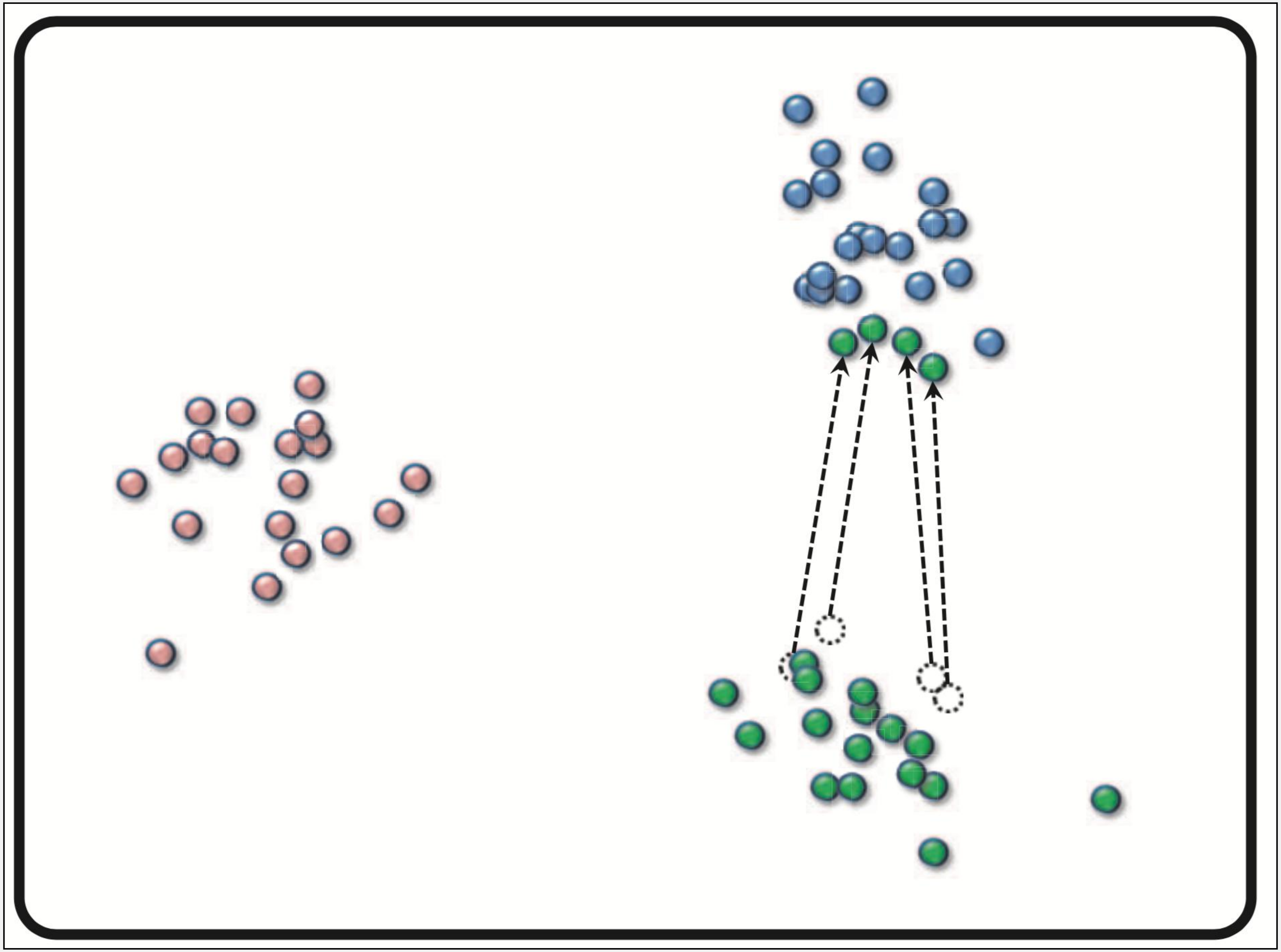
moved	**highlighted**	**untouched**	
$\frac{2}{n_m(n_m-1)}$ **(i)**	$\frac{1}{n_m n_h}$ **(ii)**	0	**moved**
	$\frac{2}{n_h(n_h-1)}$	0	**highlighted**
		0	**untouched**

Example: 3-Class Dataset

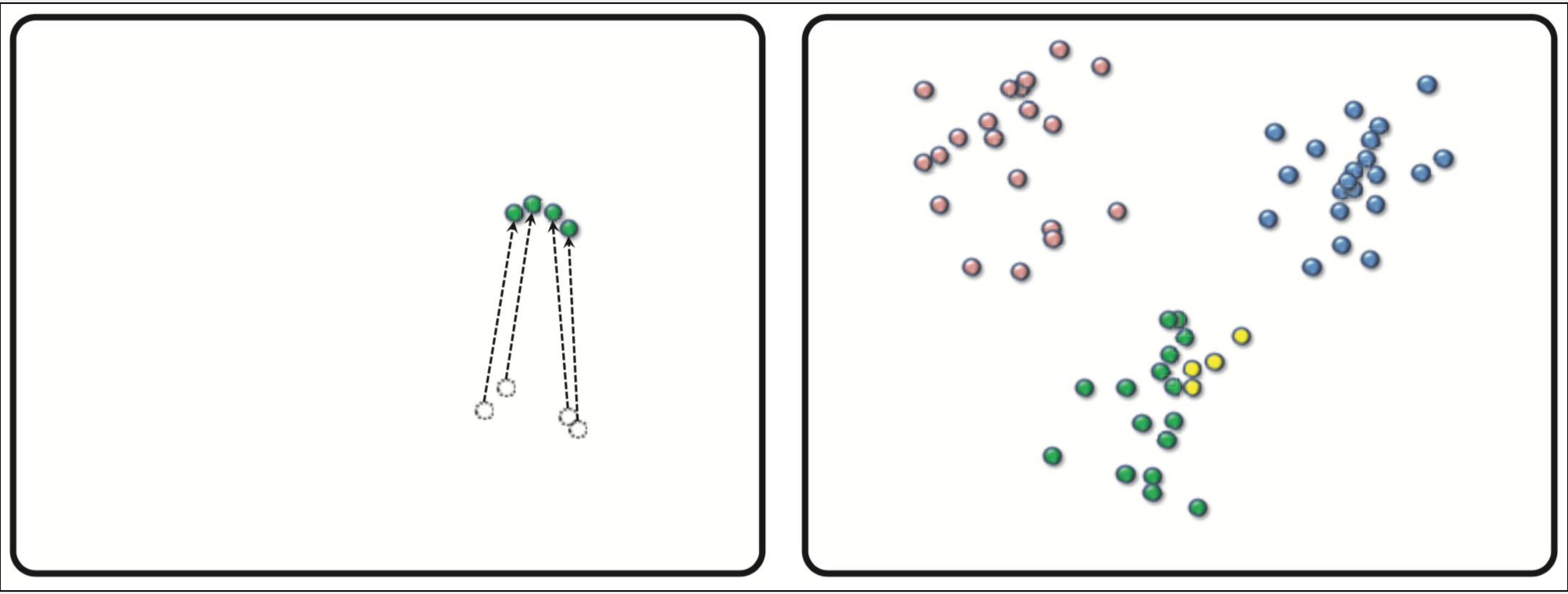


- Dimensions: 14 nutrients.
- Triticale: crossbreed of wheat and rye.
- Evaluation: through user interactions, can we identify what nutrients rye and triticale have in common?

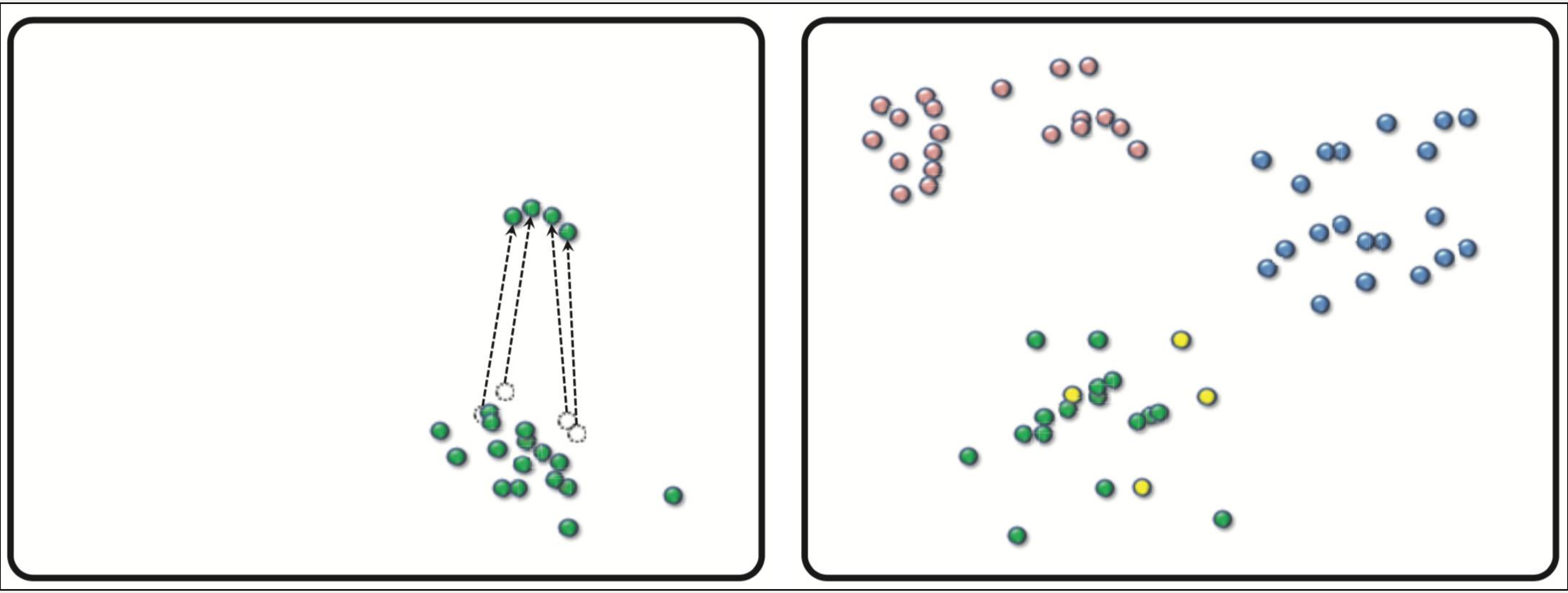
Move some points?



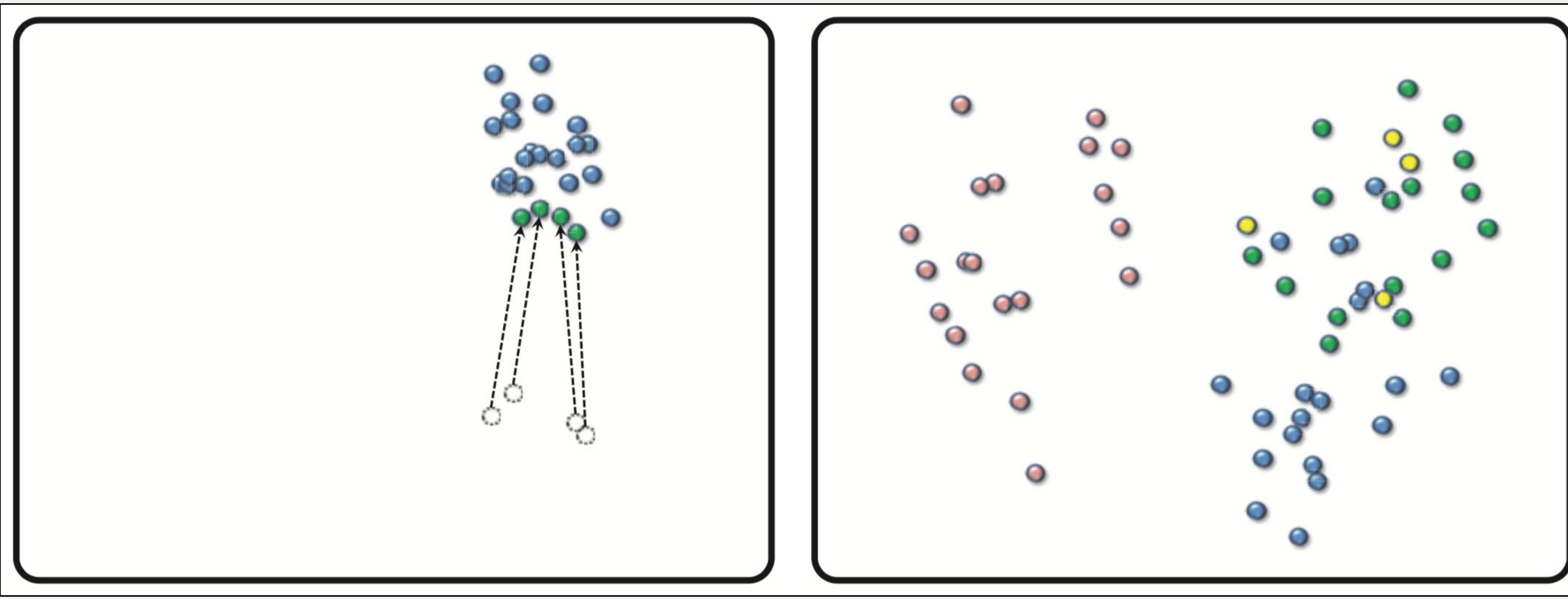
Just movement



Movement+selection

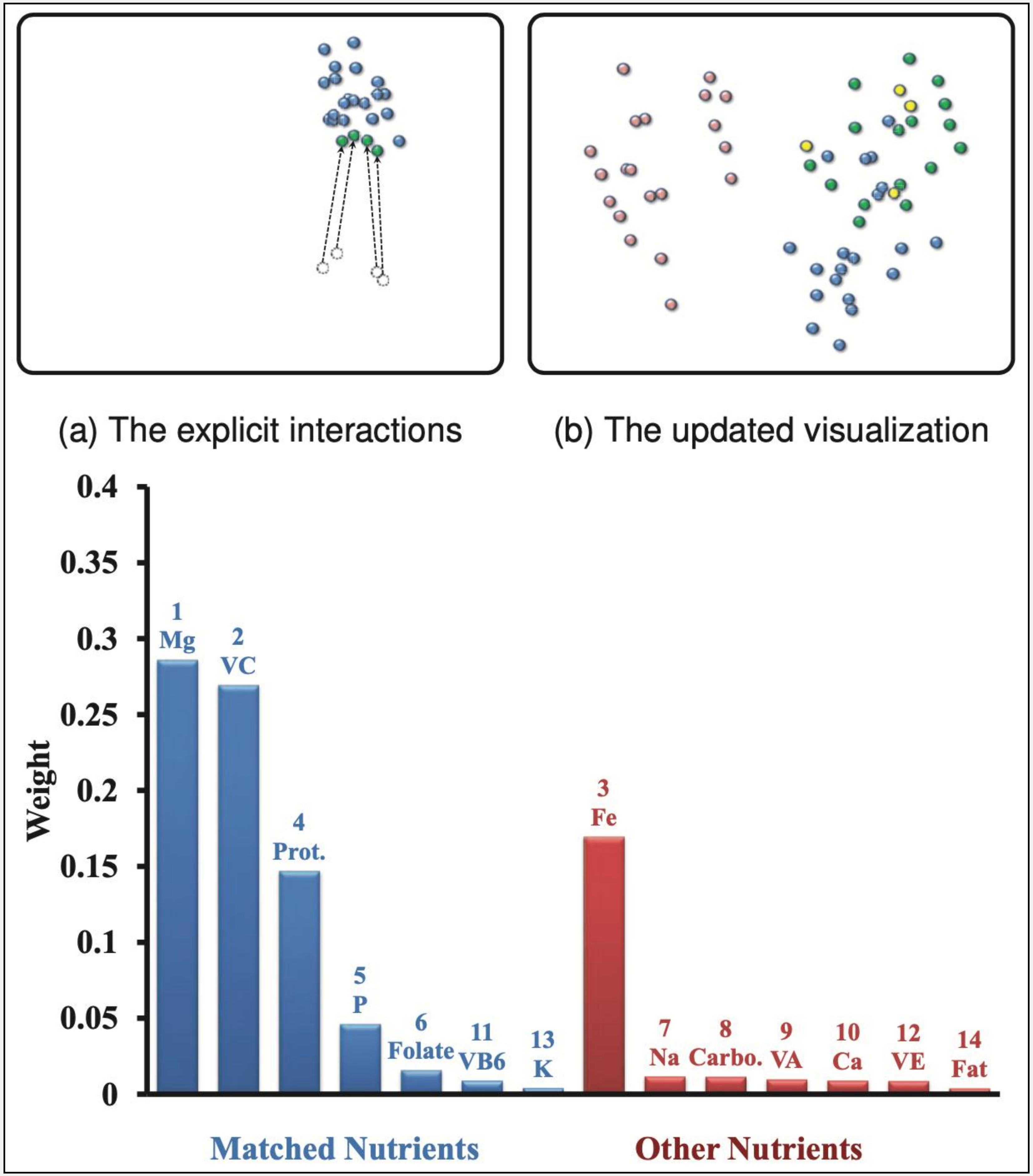


Movement+selection?



Recall: we are updating the distance metric, which can have a global impact even if the user makes local edits.

The distance?



Weighted MDS

- One way to realize user intent.
- But, what do the axes represent in the resulting 2D projection?
- All we can really say about the visualization:
  - Points that are closer are similar
  - Points that are further away are different