

# Interactive Composition Operators

## An Alternative Approach for Selecting Linear Embedding Parameters

Dirk J. Lehmann, Kai M. Blum, Manuel Rubio-Sánchez, Konrad Simon

**Abstract**—Linear embeddings support interactive visual exploration by mapping high-dimensional ( $nD$ ) data into a two-dimensional space. Despite their popularity, selecting meaningful projection parameters remains a key challenge due to the infinite  $2n$ -dimensional parameter space. Once an informative projection is found, users often seek similar ones that emphasize specific items differently while preserving global structure. For instance: Do clusters become outliers under slight changes? Can grouped items separate—or merge—through parameter adjustments? Which changes to the embedding parameters lead to such projections — and do they exist at all? Answering these questions efficiently is critical for effective visual search. Yet, current methods—such as projection tours or manual parameter tuning—are time-consuming and risk overlooking important views, including those of specific interest. We propose Composition Operators, a mathematical foundation for a novel set-of-point manipulation concept for linear embeddings—such as Star Coordinates—as an alternative approach to selecting informative embedding parameters in a more controllable manner with respect to the desired outcome. Users specify item-based constraints on the projection result; the corresponding  $2n$  parameters are then derived automatically, eliminating the need to exhaustively search the entire parameter space to get a similar outcome. Neither the embedding space nor the set of parameters is altered — only the mechanism for navigating and selecting parameters is redefined. We provide closed-form solutions for this and demonstrate our interactive prototype on  $nD$  datasets from the UCI repository.

**Index Terms**—Star Coordinates, Multivariate Projections, Composition Operators, Multidimensional Data

### 1 INTRODUCTION

A problem in the visual analysis of projections is that the outcome of a projection method is rarely controllable in a targeted way. Instead, one can only influence the parameter space that generates a projection. This process often produces projections that are informative to the user only by chance — making the analysis somewhat of a guessing game. Some works identify this problem similarly [12, 31, 39, 52]. The central question then becomes: How can the outcome of projection methods be better controlled?

For instance, Brown et al. [12] and Kim et al. [39] investigate how parameters can be chosen to steer the outcome of bivariate scatterplots, which lacks multivariate analysis. Hu et al. [31] examine how parameters can be selected to control the outcome of nonlinear embeddings, which considerably complicates the interpretation of the data measures. In contrast, Molchanov and Linsen [52] studies how parameters for linear, multivariate projections can be chosen to ensure good interpretability — even though often only one form of interaction is offered and the axes of the settings may degenerate. Moreover, many studies lack a solid theoretical foundation.

This motivates us to address the topic by providing a generalization of existing approaches and a comprehensive introduction to the mathematical methodology for achieving a controlled outcome in projections — particularly for multivariate linear projections. In doing so, we extend the concepts from previous work regarding a controllable projection outcome with respect to the use of additional dimensions, the implementation of further interaction tools for the user, the prevention of degeneracies while maintaining good interpretability.

These advances enable practical gains. We target linear techniques, such as Star Coordinates, prized for their interpretability and broad use in interactive analysis. These methods map high-dimensional data into two-dimensional spaces, yet selecting suitable embedding parameters remains challenging due to the complex parameter space. Users often wish to explore alternative projections that emphasize certain data relationships — such as separating or regrouping points [65, 70, 74] — without altering the overall structure.

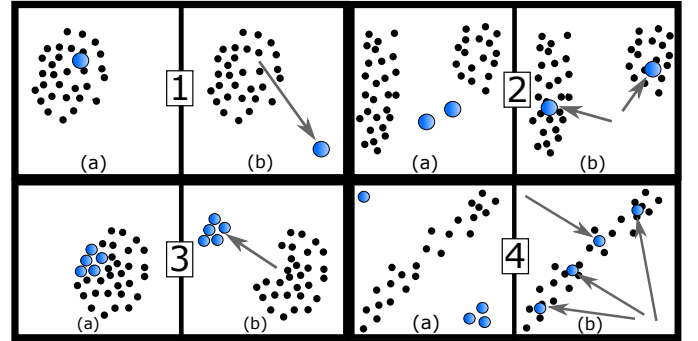


Fig. 1: Examples for different interesting relations of points of interest (blue) in the projection space w.r.t. interesting patterns.

Those tasks are shown in Figure 1: Each pair of subfigures (a) and (b) represents a user-driven question: Can selected points be repositioned in a way that changes their relation to the surrounding structure (e.g., moving a point out of a cluster, integrating an outlier into a group, or separating a subset) while preserving the global layout? These examples further motivate the need for a projection parameter selection method that enables targeted manipulations without manual tuning or exhaustive search.

Traditional techniques navigate the  $2n$ -dimensional parameter space via projection tours [8, 15, 22, 23, 44] or manual adjustments [25, 37, 38, 43, 45, 58], which may overlook important views. Recent interactive systems infer embedding parameters from direct manipulations in projection space, but rely on iterative refinement, heuristic key-point choice, or nonlinear tuning — incurring cost and obscuring the mapping [24, 61, 62]. We replace these ad-hoc steps with a closed-form solution: when users shift or anchor points, the corresponding Star-Coordinates matrix is computed directly — no iterations or heuristic reference points. The projection thus remains a valid Star-Coordinates embedding, without new spaces or extra distortions.

Our mathematical method lets users specify the behavior of selected points and automatically derives the corresponding parameters, preserving the embedding space's integrity while making point-based analysis more accessible. By manipulating shift-able and sticky points in a loop, the desired projection is approached. If a solution exists, it is computed in a least-squares sense; otherwise, the projection is infeasible for the given data. We demonstrate our approach using an interactive prototype

• Corresponding author: Dirk J. Lehmann (di.lehmann@ostfalia.de)

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxxx

and benchmark datasets, showing that *Composition Operators* enable targeted exploration of meaningful projections. Importantly, the resulting outcome remain standard Star Coordinates — no new embedding spaces are introduced. Our contributions in a nutshell are:

1. Introduction of Composition Operators as a toolbox for interactive manipulation of points in linear projections to uncover more informative views
2. A precise mathematical and technical formulation
3. An analysis of how data-vector norms affect manipulation quality
4. An interactive prototype implementation, and
5. Evaluation on benchmark  $n$ D datasets.

We begin by reviewing related work.

## 2 RELATED WORK

Work that is related to ours covers the field of *projection techniques* and the topic of *projection parameter selection*.

**Projection techniques:** Projection methods embed  $n$ D data in 2D or 3D, yet 3D adds little benefit [60]; we therefore focus on 2D. They fall into *local* and *global* approaches.

*Local* techniques reconstruct distance relationships (e.g., via neighborhood graphs [1,71]). Key examples are Stochastic Proximity Embedding [16], Local Linear Embedding [57], Laplacian Eigenmaps [9], and Stochastic Neighbor Embedding [29], used for visualization [13,55,72] and in LAMP [35]. Surveys include [7,54,73].

*Global* methods comprise (i) *bivariate* and (ii) *multivariate* projections. (i) Bivariate plots - scatterplots or Parallel Coordinates [33,34] - depict two dimensions at a time; complete SPLOMs [14] grow quadratically. (ii) Multivariate projections map the full  $n$ D space to 2D, yielding a  $2n$ -dimensional parameter space. Examples include Radial Visualizations, Star Coordinates [37,38], Orthographic Star Coordinates [43,53], linear Biplots [25,58], General Projective Maps [45], and iLamp [5]. Such complexity necessitates tailored selection strategies.

**Projection Parameter Selection:** A common approach for bivariate projections is to pre-compute projections of the parameter space and rank them with quality metrics to capture diverse patterns of interest [3,4,10,40–42,47,63,67]. Alternatively, sketch-based methods train models on user sketches [59,64], and Fink et al. [21] discuss optimal 2D axis scaling. However, finding good parameter configurations is challenging [6] because the space grows non-linearly with  $n$ , especially for multivariate projections like Star Coordinates. Manual exploration or sampling [32,36] often finds interesting views only by chance. Projection tours [8,15,22,23,51] offer an alternative, but they grow exponentially in  $n$  and become time-consuming. Shorter, interactive tours [19] or methods that avoid transformable projections [44] can be more efficient. Other techniques reduce the  $2n$ -parameter space to lower-dimensional subspaces (with  $n_0 < n$ ) [50,68,69] — for example, using subspace clustering [2,48] — since data structures typically do not span the full  $n$ D space [11,28]. These approaches rely on predefining parameters before generating the embedding, which means relevant projections are either found by chance or completely overlooked.

To overcome this, Fujiwara et al. [24] invert dimensionality-reduction (DR) mappings: users specify desired point relations, and the system infers parameters accordingly. This idea matches our goal, but our Composition Operators supply a *closed-form* solution for linear embeddings such as Star Coordinates, yielding instant, interpretable results without iterative refinement.

Dis-function [12] is an interactive system that lets users steer a projection by manipulating points in a scatterplot. It iteratively learns an improved distance function (feature weighting) based on these movements, quickly yielding a 2D view that reflects the user’s domain knowledge. This approach aligns with the idea behind Composition Operators — both use point-wise interactions to influence embedding parameters — but Composition Operators provide a closed-form solution for instantly deriving linear multivariate projection weights from constraints instead of relying on an iterative loop for only bivariate views.

InterAxis [39] offers a similar interactive technique, allowing users to redefine scatterplot axes on the fly by dragging data points to desired

extremes and computing new axes as linear combinations of attributes. The key difference is that InterAxis focuses on axis formation by example, while Composition Operators allow users to impose positional constraints on any subset of points.

Moreover, Hu et al. [31] investigate how user drag-and-drop interactions in spatial (non-linear) MDS-projection can be translated into model updates. The paper introduces interaction semantics and weighting their influence so that the underlying projection algorithm can incorporate the user’s intent. Our approach stands out in three ways: (i) *Direct derivation instead of interpretative ambiguity* - Unlike the weighted MDS [31], which may yield ambiguous interpretations of moved versus unmoved points, Composition Operators directly derive parameter changes from user-specified instructions (shift-able and sticky points), eliminating guesswork. (ii) *Preservation of Embedding* - Composition Operators preserve Star Coordinates as a linear embedding, ensuring that projections remain interpretable in terms of their data dimensions and mathematically consistent. In contrast, the MDS-based method [31] relies on iterative optimization, yielding results that are less direct and less interpretable. (iii) *Efficiency* - Closed-form expressions and least-squares optimization compute precise changes to the projection matrix, enabling efficient navigation of the  $2n$ -dimensional parameter space without exhaustive search, unlike the iterative approach.

Another way goes Piecewise Laplacian-based Projection (PLP) [56], which uses a local, piecewise linear method based on Laplacian eigenmaps to make the projection adaptable to user interventions. Users can adjust parts of the layout and the algorithm updates the 2D projection dynamically. While PLP introduces a new embedding algorithm that can be hard to interpret, Composition Operators navigate the parameter space of an existing linear embedding (like Star Coordinates) by deriving parameters that yield a more interpretable, user-desired view.

A more conceptional approach is given in SIRIUS [17], which is a framework that treats user-driven interactions and machine-driven projections as equal, interlinked parts of the dimensionality reduction process. This philosophy is very much in line with the spirit of Composition Operators, which likewise closes the loop by computing new embedding parameters from user interactions. In that sense, SIRIUS provides a general conceptual framework for such two-way interaction in dimensionality reduction, while our Composition Operators exemplify these abstract and general ideas in a specific setting by offering a concrete mathematical mechanism for user-steered updates in linear embeddings.

Explainers [27] are user-crafted projection functions that capture key concepts by replacing unintuitive knobs with goal-driven parameter selection—via annotations or interactive adjustments—for more insightful visualizations. While Explainers let users define semantic dimensions upfront, Composition Operators let users manipulate points in an existing projection to achieve a desired configuration.

Observation-level interaction lets users steer statistical models, as showed by Endert et al. [20]. Self et al. [62] generalised the idea, mapping imperative commands on point relations back to model parameters. They later realised it in *Andromeda*, a weighted-MDS tool where each drag or slider move triggers an inverse optimisation of the weight vector [61]. By contrast, our Composition Operators recover the full  $2n$  Star-Coordinates parameters in a single closed-form step, yielding immediate, interpretable updates with no iterative tuning.

Mamani et al. [49] let users reshape the feature space locally: semantic constraints on selected points trigger piecewise adjustments in their neighbourhoods. Although this empowers direct, point-level manipulation, it does not ensure a globally consistent linear embedding.

More recently, Wei et al. [75] introduced *SpaceEditing*, enabling interactive editing of deep latent spaces for (nonlinear) deep neural networks. Their work underscores a shared trend: users specify the desired layout and the system adapts the projection accordingly. Our contribution complements this direction by offering the same control in interpretable *linear* spaces - with closed-form parameter updates and guaranteed projection consistency.

Li and Zhou [46] allow the user to label certain samples and then adding a corresponding classification objective to the embedding al-

gorithm. While Li and Zhou’s technique does so by integrating labels into a deep learning model (altering the embedding with an additional loss term), Composition Operators let the user directly manipulate the projected positions of points to impose their domain knowledge.

Ghosh et al. [26] focus on how well a low-dimensional embedding preserves the key structures of the original high-dimensional data. Their approach could be used alongside Composition Operators to ensure that user-driven adjustments do not inadvertently distort the important data relationships, thereby combining controllable projection manipulation with rigorous structural interpretability.

Among related work, Molchanov and Linsen [52] offers interactive parameter selection; our approach complements this with formally defined, chainable Composition Operators that broaden the range of task-specific transformations and can reveal clearer views of the data.

In difference to most related work — which first fixes parameters, uses hard-to-interpret embeddings, or limits itself to bivariate views — we start from the desired outcome and derive the best-fitting parameters. This yields a closed-form, multivariate linear embedding that stays interpretable and supports diverse interaction schemes for greater user flexibility. The next section details the operators’ technical background.

### 3 NOTATIONS & BACKGROUND

The subsequent mathematical notations are used throughout this work.

| Symbol                       | Size / Type            | Meaning                                                                                                        |
|------------------------------|------------------------|----------------------------------------------------------------------------------------------------------------|
| $a$                          | scalar                 | real number                                                                                                    |
| $\mathbf{a}$                 | $n \times 1$ vector    | column vector $(a_1, \dots, a_n)^\top$                                                                         |
| $\ \mathbf{a}\ _2$           | $\mathbb{R}_{\geq 0}$  | Euclidean (2-)norm of vector $\mathbf{a}$                                                                      |
| $\ \mathbf{A}\ _F$           | $\mathbb{R}_{\geq 0}$  | Frobenius norm of matrix $\mathbf{A}$                                                                          |
| $\mathbf{0}_n, \mathbf{1}_n$ | $n \times 1$           | all-zero / all-one column vector                                                                               |
| $\mathbf{I}_n$               | $n \times n$           | identity matrix                                                                                                |
| $n, m$                       | scalars                | $n$ : data dimensionality; $m$ : number of data records                                                        |
| $\mathbf{d}$                 | $n \times 1$           | single data record $(d_1, \dots, d_n)^\top$                                                                    |
| $\mathbf{D}$                 | $n \times m$           | full data matrix $[\mathbf{d}_1 \dots \mathbf{d}_m]$                                                           |
| $\mathbf{M}$                 | $n \times q$           | shift-able data matrix $[\mathbf{m}_1 \dots \mathbf{m}_q]$                                                     |
| $\mathbf{Q}$                 | $n \times k$           | sticky data matrix $[\mathbf{q}_1 \dots \mathbf{q}_k]$                                                         |
| $q, k$                       | scalars                | $q$ : #shift-able points; $k$ : #sticky points                                                                 |
| $\mathbf{R}$                 | $n \times (m - q - k)$ | remaining data $[\mathbf{r}_1 \dots \mathbf{r}_{m-q-k}]$ ; $\mathbf{D} \setminus (\mathbf{M} \cup \mathbf{Q})$ |
| $\mathbf{A}$                 | $2 \times n$           | Projection matrix with anchors $\mathbf{a}_i = (x_i, y_i)^\top$                                                |
| $\mathbf{p}$                 | $2 \times 1$           | projection of one record: $\mathbf{p} = \mathbf{A}\mathbf{d}$                                                  |
| $\mathbf{P}$                 | $2 \times m$           | all projected points $[\mathbf{p}_1 \dots \mathbf{p}_m]$                                                       |
| $\mathbf{p}_m$               | $2 \times 1$           | projection of shift-able record: $\mathbf{p}_m = \mathbf{A}\mathbf{m}$                                         |
| $\mathbf{P}_m$               | $2 \times m$           | all projected shift-able points $[\mathbf{p}_{m_1} \dots \mathbf{p}_{m_q}]$                                    |
| $\mathbf{p}_q$               | $2 \times 1$           | projection of sticky record: $\mathbf{p}_q = \mathbf{A}\mathbf{q}$                                             |
| $\mathbf{P}_q$               | $2 \times m$           | all projected sticky points $[\mathbf{p}_{q_1} \dots \mathbf{p}_{q_k}]$                                        |
| $\mathbf{p}_r$               | $2 \times 1$           | projection of remaining record: $\mathbf{p}_r = \mathbf{A}\mathbf{r}$                                          |
| $\mathbf{P}_r$               | $2 \times m$           | all projected remaining points $[\mathbf{p}_{r_1} \dots \mathbf{p}_{r_{m-q-k}}]$                               |
| $\Delta\mathbf{A}$           | $2 \times n$           | perturbation of projection matrix                                                                              |
| $\Delta\mathbf{p}$           | $2 \times 1$           | user-specified shift in projection                                                                             |
| $f_i(a)$                     | function               | interpolation function, appearing as $f_0, f_1, f_2$                                                           |
| $\nabla e$                   | vector                 | Gradient of $e$                                                                                                |

Table 1: Mathematical Notation Summary

We note a lowercase letter  $a$  as a scalar, a bold lowercase letter  $\mathbf{a} = (a_1 \dots a_n)^\top$  an  $n \times 1$  column vector, and a bold uppercase letter  $\mathbf{A}$  an  $n \times m$  real-valued matrix.  $\|\mathbf{a}\|_2$  denotes the Euclidean norm of  $\mathbf{a}$  and  $\|\mathbf{A}\|_F$  denotes the Frobenius norm of a matrix  $\mathbf{A}$  given by

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^n a_i^2} \quad \text{and} \quad \|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m a_{ij}^2}.$$

The term *norm* is used synonymously in the text and is meant to be the Euclidean norm in case of vectors and the Frobenius norm in case of matrices. The term *data norm* is the Euclidean norm of a data record.  $\mathbf{0}_n$  is the zero column vector with  $\mathbf{0}_n = (0 \ 0 \dots 0)^\top$ , while  $\mathbf{1}_n$  is the one column vector  $\mathbf{1}_n = (1 \ 1 \dots 1)^\top$ , and  $\mathbf{I}_n$  is the  $n \times n$  identity matrix. We avoid to mention the indices if they are clear from the context. Be  $n$  the number of data dimensionality and be  $m$  the number of data records in the data. Be  $\mathbf{d} = (d_1 \dots d_n)^\top$  a data point in vector form in the  $nD$  data set. Then, the  $n \times m$  data matrix  $\mathbf{D}$  is set to be  $\mathbf{D} = (\mathbf{d}_1 \mathbf{d}_2 \dots \mathbf{d}_m)$ .

The number  $q$  of data points  $\mathbf{d}_i$  that are allowed to be interactively *shifted* by the user in the projection are relabeled as  $\mathbf{m}_i$  and collected within an  $n \times q$  matrix  $\mathbf{M} = (\mathbf{m}_1 \dots \mathbf{m}_q)$ . The number  $k$  of such data points  $\mathbf{d}_j$  that are not allowed to move around in the projection space but have to *stick* to the same position (even if the projection parameters change) are relabeled as  $\mathbf{q}_j$  and collected within an  $n \times k$  matrix  $\mathbf{Q}$  as  $\mathbf{Q} = (\mathbf{q}_1 \dots \mathbf{q}_k)$ , with  $\mathbf{M} \cap \mathbf{Q} = \emptyset$ , meaning that a data point is either supposed to be *shift-able* or to be *sticky* in the projection space (but not both). The remaining data set  $\mathbf{R} = (\mathbf{r}_1 \dots \mathbf{r}_{m-q-k})$  with those data points that shall be none of the two are given by  $\mathbf{R} = \mathbf{D} \setminus \{\mathbf{M} \cup \mathbf{Q}\}$ , which results in a set of disjoint sets, as a decomposition of the data

$$\underbrace{\mathbf{D}}_{n \times m} = \underbrace{\mathbf{R}}_{n \times (m-q-k)} \cup \underbrace{\mathbf{M}}_{n \times q} \cup \underbrace{\mathbf{Q}}_{n \times k}.$$

In the following, we describe the Star Coordinate projection process. Be the  $2n$ -projection parameter space  $\mathbf{A}$  given as a  $2 \times n$  real-valued projection matrix

$$\mathbf{A} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{pmatrix} = (\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_n).$$

A column vector  $\mathbf{a}_i = (x_i \ y_i)^\top$  of  $\mathbf{A}$  is denoted as *anchor point* and it represents the influence of a single dimension  $i$  to the resulting projection. An intuitive visualization of the projection matrix  $\mathbf{A}$  is as a set of 2D points representing the anchor points  $\mathbf{a}_i$  and the axes  $\mathbf{a}_i - \mathbf{0}_2$  (around the origin  $\mathbf{0}_2$ ) as Figure 2 (left) illustrates. Interacting with the anchor points (which updates the projection matrix  $\mathbf{A}$ ) gives an interactive approach to explore the  $2n$ -projection parameter space. In order to get the Star Coordinates, an  $nD$  data record  $\mathbf{d}$  is mapped onto the 2D projection space by the operation  $\mathbf{p} = \mathbf{A} \cdot \mathbf{d}$ , where  $\mathbf{p}$  is a 2D point, giving the  $2 \times m$  matrix  $\mathbf{P}$  of projected points as

$$\mathbf{P} = \mathbf{A} \cdot \mathbf{D} = (\mathbf{p}_1 = \mathbf{A}\mathbf{d}_1 \ \mathbf{p}_2 = \mathbf{A}\mathbf{d}_2 \ \dots \ \mathbf{p}_m = \mathbf{A}\mathbf{d}_m).$$

This gives us a useful interpretation for a projected point in Star Coordinates as a linear combination of the data dimensions, enabling a dimension-wise data interpretation to the user (Figure 2 (left)). Note that for non-linear local projections such an interpretation is not feasible, due to the non-linearly embedded dimensions (Figure 2 (right)). This motivates us to rely on Star Coordinates.

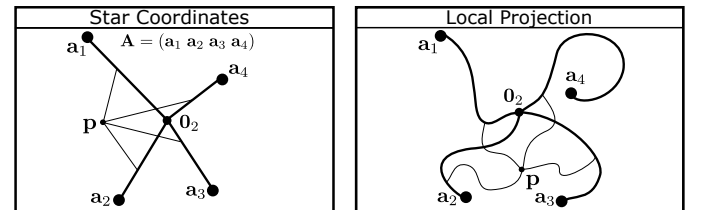


Fig. 2: Projection matrix visualization via  $n = 4$  anchor points in Star Coordinates (left) and for any local projection technique (right).

Its projection yields three domains: the  $nD$  **data space**, the  $2nD$  **projection matrix space**, and the 2D **projection space**. Building on these concepts, we subsequently introduce Composition Operators.

#### 4 CONCEPT OF COMPOSITION OPERATORS

To set the stage, we first outline the core principles that underpin our Composition Operator framework: Adding a perturbation  $\Delta\mathbf{A}$  to the projection matrix  $\mathbf{A}$  displaces a projected point  $\mathbf{p}$  by  $\Delta\mathbf{p}$ :

$$\mathbf{p} + \Delta\mathbf{p} = (\mathbf{A} + \Delta\mathbf{A}) \cdot \mathbf{d}, \quad \mathbf{p} = \mathbf{A} \cdot \mathbf{d}, \quad \Delta\mathbf{p} = \Delta\mathbf{A} \cdot \mathbf{d}, \quad \Delta\mathbf{p} = (\Delta p_1, \Delta p_2)^T.$$

Given a user-specified shift  $\Delta\mathbf{p}$  for  $\mathbf{p}$ , we seek a matching  $\Delta\mathbf{A}$  that tailors the new projection  $\mathbf{A} + \Delta\mathbf{A}$  for better insight. Let  $\mathbf{P}_m$  be projected from *shift-able* data  $\mathbf{M}$ ,  $\mathbf{P}_q$  projected from *sticky* data  $\mathbf{Q}$ , and  $\mathbf{P}_r = (\mathbf{p}_r \dots \mathbf{p}_{m-q-k})$  projected from remaining data  $\mathbf{R}$ ; giving

$$\underbrace{\mathbf{P}}_{\text{Projected Data}} = \underbrace{\mathbf{P}_r}_{\text{Projected Remaining Data}} \cup \underbrace{\mathbf{P}_m}_{\text{Projected Shift-able Data}} \cup \underbrace{\mathbf{P}_q}_{\text{Projected Sticky Data}}.$$

We start by the initial projection  $\mathbf{P} = \mathbf{A} \cdot \mathbf{D}$ : then the user marks a number  $k$  of points as *sticky*, forming the  $2 \times k$  matrix  $\mathbf{P}_q$ . The corresponding data records build matrix  $\mathbf{Q}$  in a way that the point  $\mathbf{p}_{q_j} \in \mathbf{P}_q$  is the projection point  $\mathbf{p}_{q_j} = \mathbf{A} \cdot \mathbf{q}_j$  of the data point  $\mathbf{q}_j \in \mathbf{Q}$ .

$\mathbf{P}_q \dots \text{sticky projection points} \rightarrow \mathbf{Q} \dots \text{sticky related data points}.$

These *sticky* projection points are supposed to preserve their position as good as possible under varying the initial projection matrix  $\mathbf{A}$  with a slight change  $\Delta\mathbf{A}$  to a new projection  $\bar{\mathbf{A}} = \mathbf{A} + \Delta\mathbf{A}$ .

Moreover, the user tags a second set (being disjoint to  $\mathbf{Q}$ ) of a number of  $q$  points as *shift-able*, defining the  $2 \times q$  matrix of points  $\mathbf{P}_m$ . The related data records  $\mathbf{m}_i \in \mathbf{M}$  define the matrix  $\mathbf{M}$  as described above:

$\mathbf{P}_m \dots \text{shift-able projection points} \rightarrow \mathbf{M} \dots \text{shift-able related data points}.$

The *shift-able* projection points are supposed to change their position about a certain shift vector  $\Delta\mathbf{p}$ , as  $\bar{\mathbf{p}}_{m_i} = \mathbf{p}_{m_i} + \Delta\mathbf{p}$  under varying the initial projection  $\mathbf{A}$  with a slight change  $\Delta\mathbf{A}$  to a new projection  $\bar{\mathbf{A}} = \mathbf{A} + \Delta\mathbf{A}$ . Based on a user-given shift vector  $\Delta\mathbf{p} = (\Delta p_1 \Delta p_2)^T$ , we are now interested in finding a change of the projection  $\Delta\mathbf{A}$  to a novel projection  $\bar{\mathbf{A}} = \Delta\mathbf{A} + \mathbf{A}$  in a way that:

- **Low Impact**: the global projection does not change too much,
- **Best Shift**: the *shift-able* points are placed (closely) to  $\mathbf{p}_{m_i} + \Delta\mathbf{p}$ ,
- **Best Stickiness**: the *sticky* points change as little as possible.

A) **Low Impact**: when changing from projection  $\mathbf{A}$  to  $\bar{\mathbf{A}}$ , the total impact for the projection is given by the term (cf. Figure 3(a))

$$\sum_{i=1}^{m-q-k} \|\bar{\mathbf{A}} \cdot \mathbf{r}_i - \mathbf{A} \cdot \mathbf{r}_i\|_2^2 = \sum_{i=1}^{m-q-k} \|(\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{r}_i\|_2^2 \quad (1)$$

$$\leq \|(\bar{\mathbf{A}} - \mathbf{A})\|_F^2 \cdot \sum_{i=1}^{m-q-k} \|\mathbf{r}_i\|_2^2 = \underbrace{\|(\bar{\mathbf{A}} - \mathbf{A})\|_F^2}_{\text{Projection Amount}} \cdot \underbrace{\left( \text{Tr}(\mathbf{R}^T \mathbf{R}) \right)}_{\text{Data Amount}}. \quad (2)$$

Since the data only enter through the constant positive trace value, we drop it - leaving the topology unchanged - so the bound simplifies to  $e_A$

$$e_A = \|(\bar{\mathbf{A}} - \mathbf{A})\|_F^2 \dots \text{Projection Impact Quality}. \quad (3)$$

Note that (3) is a *weaker* condition than (1); if (1) vanishes, then (3) also vanishes, but not vice versa. We accept this trade-off because omitting the data here is more efficient: the scheme depends only on the size  $n$  of the projection matrix, and typically  $n \ll m - q - k$  applies.

B) **Best Shift**: when changing from projection  $\mathbf{A}$  to  $\bar{\mathbf{A}}$ , the overall shift quality  $e_M$  is the difference between the desired position  $\mathbf{p}_{m_i} + \Delta\mathbf{p} = (\mathbf{A} \cdot \mathbf{m}_i) + \Delta\mathbf{p}$  and the reached position  $\bar{\mathbf{p}}_{m_i} = \bar{\mathbf{A}} \cdot \mathbf{m}_i$  (cf. Figure 3(b)) given by the term

$$e_M = \sum_{i=1}^q \|(\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{m}_i - \Delta\mathbf{p}\|_2^2 \dots \text{Shift Quality}. \quad (4)$$

C) **Best Stickiness**: when changing from projection  $\mathbf{A}$  to  $\bar{\mathbf{A}}$ , the overall stickiness quality  $e_Q^-$  *without memory* is given as the difference of the position of point  $\mathbf{p}_{q_j} = \mathbf{A} \cdot \mathbf{q}_j$  for projection  $\mathbf{A}$  and the position  $\bar{\mathbf{p}}_{q_j} = \bar{\mathbf{A}} \cdot \mathbf{q}_j$  for the successor projection  $\bar{\mathbf{A}}$  (Figure 3(c1)), which gives

$$e_Q^- = \sum_{j=1}^k \|(\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{q}_j\|_2^2 \dots \text{Stickiness Quality.} \quad (5)$$

--- without memory ---

The overall stickiness quality  $e_Q^+$  *with memory* equals the difference of the current position of point  $\bar{\mathbf{A}} \cdot \mathbf{q}_j$  and the original position  $\mathbf{p}_{q_j} \in \mathbf{P}_q$ , where it was when the user tagged it as *sticky* (cf. Figure 3(c2)), giving

$$e_Q^+ = \sum_{j=1}^k \|(\bar{\mathbf{A}} \cdot \mathbf{q}_j) - \mathbf{p}_{q_j}\|_2^2 \dots \text{Stickiness Quality.} \quad (6)$$

++ with memory ++

The term *without memory*  $e_Q^-$  measures the deviation of each sticky point with respect to its *immediately preceding* position in the current interaction step; it thus penalises only the *incremental drift* that occurs from one projection matrix to the next. In contrast, the term *with memory*  $e_Q^+$  measures the deviation with respect to the *initial anchor position* at the moment the point was declared sticky, thereby accumulating the *total drift* over the entire interaction history. Put differently,  $e_Q^-$  acts like a short-range spring that smooths local jitter, whereas  $e_Q^+$  behaves like a long-range spring that continually pulls the point back to its original location, ensuring long-term positional stability.

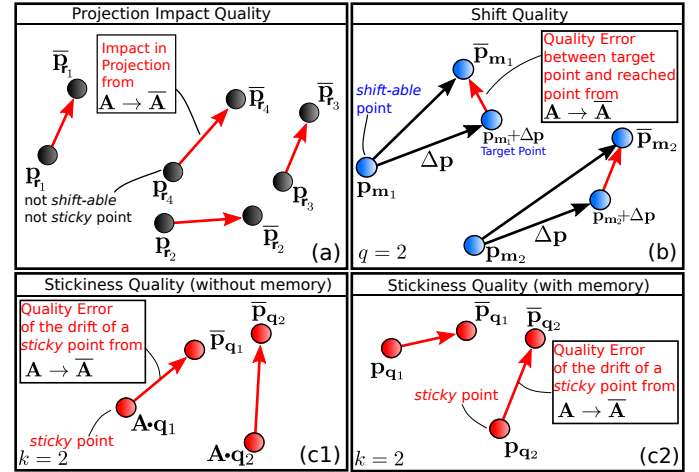


Fig. 3: Quality metrics for impact (a), shift quality (b), and stickiness (c).

Now, we define the total manipulation quality energy  $e(\bar{\mathbf{A}})^*$  as:

$$e(\bar{\mathbf{A}})^* = \left[ (\alpha \cdot e_A) + (\beta \cdot e_M) + (\gamma \cdot e_Q^*) \right] \dots \text{Manipulation Energy} \quad (7)$$

where  $*$  in  $e(\bar{\mathbf{A}})^*$  is a placeholder denoting  $e(\bar{\mathbf{A}})^-$  is the total manipulation energy *without memory* and  $e(\bar{\mathbf{A}})^+$  *with memory*, and  $\alpha, \beta, \gamma \in [0, 1]$  are weightings for the three quality terms. Note that while the terms  $e_M$  and  $e_Q^*$  have their global minimum at 0, the term  $e_A$  gets 0 for  $\Delta\mathbf{p} = 0$ , i.e., when no shift operation is applied. Thus, the minimum of  $e_A$  is given in practise by a small but positive value, and thus the manipulation energy  $e(\bar{\mathbf{A}})^*$  does.

The gradient  $\nabla e(\bar{\mathbf{A}})^-$  *without memory* is given as:

$$\begin{aligned} & \text{Manipulation Energy Gradient } \nabla e(\bar{\mathbf{A}})^- \\ & \dots \dots \text{without memory} \dots \dots \\ & = 2 \left( (\bar{\mathbf{A}} - \mathbf{A}) \cdot \left[ \beta \cdot \mathbf{M} \mathbf{M}^T + \gamma \cdot \mathbf{Q} \mathbf{Q}^T + \alpha \cdot \mathbf{I}_n \right] - \left[ \beta \cdot \Delta\mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T \right] \right) \end{aligned} \quad (8)$$

and the gradient  $\nabla e(\bar{\mathbf{A}})^+$  *with memory* is given as:

$$\begin{aligned} & \text{Manipulation Energy Gradient } \nabla e(\bar{\mathbf{A}})^+ \\ & \dots \dots \text{with memory} \dots \dots \\ & = 2 \left( (\bar{\mathbf{A}} - \mathbf{A}) \left[ \beta \mathbf{M} \mathbf{M}^T + \alpha \mathbf{I}_n \right] + \left[ \gamma \bar{\mathbf{A}} \mathbf{Q} \mathbf{Q}^T \right] - \left[ \beta \Delta\mathbf{p} (\mathbf{M} \cdot \mathbf{1}_q)^T - \gamma \mathbf{P}_q \mathbf{Q}^T \right] \right) \end{aligned} \quad (9)$$

Please find the proof for (8) and (9) in the *additional material*.

A solution for the projection matrix  $\bar{\mathbf{A}}^* = \mathbf{A} + \Delta\mathbf{A}$  leading from  $\mathbf{A}$  to  $\bar{\mathbf{A}}$  minimizing the manipulation energy  $e(\bar{\mathbf{A}})^*$ , i.e., having vanishing gradient  $\nabla e(\bar{\mathbf{A}})^*$ , is given *without memory* in its general form as

Composition Operator  
----- without memory -----

$$\bar{\mathbf{A}}^- = \left[ \beta \cdot \Delta\mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T \cdot \mathbf{H}^{-1} \right] + \mathbf{A} \quad (10)$$

and *with memory* in its general form as

Composition Operator  
+++++ with memory +++++

$$\bar{\mathbf{A}}^+ = \left[ \beta \cdot \Delta\mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T + (\gamma \cdot \mathbf{P}_q \mathbf{Q}^T) + \mathbf{A}(\alpha \cdot \mathbf{I}_n + \beta \cdot \mathbf{M} \mathbf{M}^T) \right] \cdot \mathbf{H}^{-1}$$

with the  $n \times n$  matrix  $\mathbf{H} = (\beta \cdot \mathbf{M} \mathbf{M}^T + \gamma \cdot \mathbf{Q} \mathbf{Q}^T + \alpha \cdot \mathbf{I}_n)$ .

We set  $\alpha = 1$  as further discussed in the *additional material*. Moreover, please find the proof of the Composition Operators in (10) and (11) and a discussion for the existence of  $\mathbf{H}^{-1}$  in the *additional material*, too. Note that  $\mathbf{H}^{-1}$  needs to be computed once for a selection of points but allows then an infinite number of interactive shifting operations to explore projections of interest. Moreover, inverting a matrix might be expensive. However, in this case,  $\mathbf{H}$  is growing with the number  $n$  of dimensions. Since  $n \ll m$  applies for most cases, inverting  $\mathbf{H}$  is neither a critical nor an expensive operation. In the following, we derive and discuss various special cases and scenarios from the general forms of the composition operators.

#### 4.1 Composition Operator for Shift-Able Sets

In this section, we begin with the scenario in which the user interactively selects and moves only a set  $\mathbf{M}$  of shift-able points - no sticky points are selected or considered yet. Please note, the distance of a data point to its origin in the data space relate to its manipulation ability in projection space. We are going to discuss this effect deeply in Sec. 5.

For now, we deal with this by introducing a process named *Norm Equalization*, in the following manner: we introduce a user parameter  $\varepsilon \in [0, 1]$  and normalize the matrix  $\mathbf{M}$  of data points to  $\mathbf{M}_\varepsilon$  with

$$\mathbf{m}_\varepsilon = (\mathbf{m} / \|\mathbf{m}\|_2) \cdot f_0(\varepsilon),$$

where  $f_0(\varepsilon)$  is a polynomial of degree 2 with  $f_0(0) = \|\mathbf{m}_{\min}\|_2$ ,  $f_0(0.5) = \|\mathbf{m}_{\max}\|_2$ ,  $f_0(1) = \|\mathbf{m}\|_2$ , giving  $f_0$  as

$$f_0(\varepsilon) = \|\mathbf{m}_{\min}\|_2 + (-3\|\mathbf{m}_{\min}\|_2 - \|\mathbf{m}\|_2 + 4\|\mathbf{m}_{\max}\|_2)\varepsilon + (2\|\mathbf{m}_{\min}\|_2 + 2\|\mathbf{m}\|_2 - 4\|\mathbf{m}_{\max}\|_2)\varepsilon^2,$$

with  $\mathbf{m}_{\min}$  and  $\mathbf{m}_{\max}$  have minimal or maximal norm ( $\|\mathbf{m}_{\min}\|_2$  or  $\|\mathbf{m}_{\max}\|_2$ ) in  $\mathbf{M}$ . This gives an intuitive interpretation of the parameter  $\varepsilon$ : For  $\varepsilon = 1$ , no *Norm Equalization* in  $\mathbf{M}$  is applied, i.e.,  $\mathbf{M}_\varepsilon = \mathbf{M}$  (see Fig. 4(a)); for  $\varepsilon = 0.5$ , the data points in  $\mathbf{M}_\varepsilon$  are mapped to the hypersphere of the most distant position of  $\mathbf{M}$  and thus all members have the same distance now (see Fig. 4(b) green circle); and for  $\varepsilon = 0$  the data points in  $\mathbf{M}_\varepsilon$  are mapped to the hypersphere of the smallest distant position of  $\mathbf{M}$  (see Fig. 4(c) green circle).

With the *Norm Equalization*, and by putting  $\beta = 1$ ,  $\gamma = 0$ ,  $q > 1$  in (8), the related gradient of the manipulation energy  $e$  is

Manipulation Energy Gradient  $\nabla e(\bar{\mathbf{A}})_\varepsilon$   
..... for shift of a point set .....

$$= 2 \left( (\bar{\mathbf{A}} - \mathbf{A}) \cdot [\mathbf{M}_\varepsilon \mathbf{M}_\varepsilon^T + \mathbf{I}_n] - [\Delta\mathbf{p} \cdot (\mathbf{M}_\varepsilon \cdot \mathbf{1}_q)^T] \right) \quad (12)$$

With this, and (10) the Composition Operator is given as

Optimal Projection Composition Operator  
..... for shift of a point set .....

$$\bar{\mathbf{A}}_\varepsilon = \left[ \Delta\mathbf{p} \cdot (\mathbf{M}_\varepsilon \cdot \mathbf{1}_q)^T \cdot \mathbf{H}_\varepsilon^{-1} \right] + \mathbf{A} \quad (13)$$

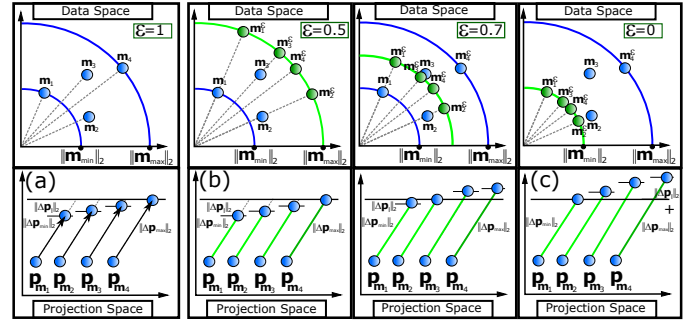


Fig. 4: Norm Equalization of matrix  $\mathbf{M}$ .

$$\text{with } \mathbf{H}_\varepsilon = (\mathbf{M}_\varepsilon \mathbf{M}_\varepsilon^T + \mathbf{I}_n). \quad (14)$$

Please note, if only one single point shall be shifted, (13) can be used as well. However, for that case, a simpler version of (13) can be derived as shown in the additional material under *One Shift-Able Point*.

#### 4.2 Composition Operator for Shift-Able vs. Sticky Sets

In this section, we consider now the scenario in which the user interactively selects and moves shift-able points and selects sticky points as well. Given a set  $\mathbf{M}$  of *shift-able* points and a set  $\mathbf{Q}$  of *sticky* points, the general manipulation gradients from (8) and (9), along with the Composition Operators from (10) and (11) - with or without memory - become relevant. Our aim is to keep a set  $\mathbf{Q}$  (with  $k > 1$ ) as close as possible to its initial position while shifting another set  $\mathbf{M}$  to a desired location using the same projection matrix. The shift and stick constraints are not inherently conflicting: if the shift distances and the number of elements are small, the constraints may be self-consistent. However, with an increased number of shifts or larger sets, the constraints can become contradictory by over-constraining the projection matrix.

For compensation, we introduce a second user parameter  $\delta \in [0, 1]$  to provide users with a streamlined means to blend between the constraints on demand in the following way: Usually, one does not want to have two parameters to choose the importance of constraints in an optimization, as is the case here with  $\beta$  and  $\gamma$ . Instead, one would like to have a single user parameter - e.g., a parameter  $\delta \in [0, 1]$  - to control all sensible configurations of the weighting of the conditions. In doing so, the following should apply:

- $\delta = 1.0 \Rightarrow \beta = \gamma = 1$ : *shift-able* & *sticky* points equally important.
- $\delta = 0.5 \Rightarrow \beta = 1, \gamma = 0$ : *shift-able* important, *sticky* not.
- $\delta = 0.0 \Rightarrow \beta = 0, \gamma = 1$ : *shift-able* not important, *sticky* important.

This gives rise to the question of how  $\beta$  and  $\gamma$  can be chosen as a function of  $\delta$  given the three support points:

|                 |   |     |   |
|-----------------|---|-----|---|
| $\delta$        | 1 | 0.5 | 0 |
| $\beta(\delta)$ | 1 | 1   | 0 |

|                  |   |     |   |
|------------------|---|-----|---|
| $\delta$         | 1 | 0.5 | 0 |
| $\gamma(\delta)$ | 1 | 0   | 1 |

Such a problem is a simple interpolation through three points, which requires at least a degree-2 polynomial; by parsimony, we choose this minimal degree, yielding a quadratic interpolation:

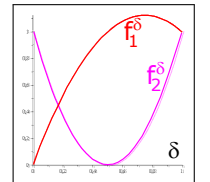
$$\beta(\delta) = -2\delta^2 + 3\delta, \quad \gamma(\delta) = 4\delta^2 - 4\delta + 1. \quad (15)$$

Hence the user adjusts solely parameter  $\delta$ , while the algorithm derives consistent trade-offs - purely algebraic, with no hidden heuristics.

Then, we set the energy weightings in (7)

to  $\beta = \beta(\delta) = f_1^\delta$  and  $\gamma = \gamma(\delta) = f_2^\delta$ , giving the blending parameter  $\delta$  a clear meaning: for  $\delta = 1$  shift and stickiness are equally weighted; at  $\delta = 0.5$  stickiness is reduced to favor shifting; at  $\delta = 0$  shift vanishes, maximizing stickiness.

With the impact control parameter  $\varepsilon$  and the blending parameter  $\delta$  (and  $\alpha = 1$ ;  $q, k > 1$ ) introduced before, from (10) and (11) it follows the Composing Operator *with/without memory* as<sup>1</sup>:



<sup>1</sup> Note that from (8) and (9) the gradient follows similarly.

Optimal Projection Composition Operator  
: : without memory and with control and blending : :

$$\bar{\mathbf{A}}_{\varepsilon\delta} = \left[ f_1^\delta \cdot \Delta \mathbf{p} \cdot (\mathbf{M}_\varepsilon \cdot \mathbf{1}_q)^T \cdot \mathbf{H}_{\varepsilon\delta}^{-1} \right] + \mathbf{A} \quad (16)$$

and with memory as

Optimal Projection Composition Operator  
 $\approx \approx \approx$  with memory, control and blending  $\approx \approx \approx$

$$\bar{\mathbf{A}}_{\varepsilon\delta}^+ = \left[ f_1^\delta \cdot \Delta \mathbf{p} \cdot (\mathbf{M}_\varepsilon \cdot \mathbf{1}_q)^T + (f_2^\delta \cdot \mathbf{P}_q \mathbf{Q}^T) + \mathbf{A} (\mathbf{I}_n + f_1^\delta \cdot \mathbf{M}_\varepsilon \mathbf{M}_\varepsilon^T) \right] \cdot \mathbf{H}_{\varepsilon\delta}^{-1} \quad (17)$$

with the  $n \times n$  matrix  $\mathbf{H}_{\varepsilon\delta} = (f_1^\delta \cdot \mathbf{M}_\varepsilon \mathbf{M}_\varepsilon^T + f_2^\delta \cdot \mathbf{Q} \mathbf{Q}^T + \mathbf{I}_n)$ . (18)

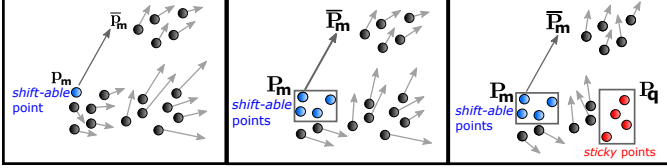


Fig. 5: Composition Operators: (left) one *shift-able* point  $\mathbf{p}_m$ , (middle) a set  $\mathbf{P}_m$  of *shift-able* points (right) with an additional set  $\mathbf{P}_q$  of *sticky* points.

Our Composition Operators enable various interactive projection point manipulation strategies to form a parameter set of interest in  $\bar{\mathbf{A}}$ . Figure 5 illustrates these manipulations. With this, we conclude our discussion of operator classifications and now address practical aspects of using these operators.

### 4.3 Practical Aspects When Using Composition Operators

**Solution Space:** Due to the  $2n$  degrees of freedom, a good fitting solution in  $\bar{\mathbf{A}}$  exists if  $q + k \leq n$ . When  $q + k > n$ , we can still approximate the constraints well — either by adjusting the points or by keeping the sticky points fixed. To balance these approaches, we introduced the blending parameter  $\delta$ . More details are found in Section 6.

However, note that the solution space in  $\bar{\mathbf{A}}$  for the *sticky* points is larger than for the *shift-able* points. For sticky points, only the distance around the original position matters for a good fit, whereas for shift-able points both the distance to the target and the angular deviation are important - making shifting more restrictive than keeping points fixed. The optimal  $\delta$  depends on whether accurate shifting or precise attachment is more important to the user, and including  $\delta$  in the optimization leads to a non-linear problem without a closed-form solution. Nevertheless, the parameter remains a valuable tool for supporting the projection composition process.

**Attached/Fixed Anchor Points:** In order to support the visual analysis between the projected points and the data dimensions even further, an interesting operation is to forbid an anchor point  $\mathbf{a}_i \in \mathbf{A}$  to change its position. Clearly, keeping a constant position for the anchor point  $\mathbf{a}_i$  when the Composition Operators are used, i.e.,  $\mathbf{a}_i \in \mathbf{A} = \mathbf{a}_i \in \bar{\mathbf{A}}$ . The naive solution is either to set the  $i^{\text{th}}$  column vector in  $\mathbf{A}$  to  $\mathbf{0}_2$ , or to replace the  $i^{\text{th}}$  column vector in  $\bar{\mathbf{A}}$  by  $\mathbf{a}_i$  in the aftermath. Both approaches would negatively influence the manipulation energy and thus worsen the results. A reasonable solution is to set the  $i^{\text{th}}$  row vector in  $\mathbf{M}_\varepsilon$  and  $\mathbf{Q}$  to  $\mathbf{0}^T$ , and to subtract from the projected points  $\mathbf{P}_q$  the linear influence of the fixed axes as  $\mathbf{P}_q := \mathbf{P}_q - (\mathbf{A}_{\text{fix}} \cdot \mathbf{Q})$ , which finally preserves the position of  $\mathbf{a}_i$ . Here,  $\mathbf{A}_{\text{fix}}$  is a  $2 \times n$  zero matrix with the anchor point  $\mathbf{a}_i$  at the  $i^{\text{th}}$  position, e.g., to fix the third dimension gives  $\mathbf{A}_{\text{fix}} = (\mathbf{0}_2 \ \mathbf{0}_2 \ \mathbf{a}_3 \ \mathbf{0}_2 \ \dots \ \mathbf{0}_2)$ , or to fix the fourth dimension gives  $\mathbf{A}_{\text{fix}} = (\mathbf{0}_2 \ \mathbf{0}_2 \ \mathbf{0}_2 \ \mathbf{a}_4 \ \dots \ \mathbf{0}_2)$  and so on. This procedure can be applied to fix an arbitrary number of anchor points. Note, to cancel rows in the matrices  $\mathbf{M}$  and  $\mathbf{Q}$  is equivalent to removing two degrees of freedom each time for  $\bar{\mathbf{A}}$ , while the system gets over-constrained with a fewer number of points as discussed before.

**Distortions Minimizing Operators:** Given a Composition Operator  $\bar{\mathbf{A}}$ , an orthographic version  $\bar{\mathbf{A}}^\pi := \text{orth}(\bar{\mathbf{A}})$  can be generated [43] minimizing projection distortions and axis degeneration. A cost-effective reconditioning for *orth* can be applied with complexity  $O(n)$  (see [43]). The additional impact on projection is negligible for small  $\Delta \mathbf{p}$ , since  $\lim_{\Delta \mathbf{p} \rightarrow 0} \|\bar{\mathbf{A}}^\pi - \bar{\mathbf{A}}\|_F^2 = 0$ .

**Gradient Descent:** By having a closed form solution for the operators  $\bar{\mathbf{A}}$ , a one step solution is feasible. However, if the *shift-able* and/or the *sticky* point sets are really large, and the sets do not fit into the storage anymore, or for other reasons, a gradient descent could be used alternatively to find a minimum for  $\bar{\mathbf{A}}$ . By using the manipulation energy gradients of (8), (9), and (12), the gradient descent is:

$$\bar{\mathbf{A}}_{i+1} = \bar{\mathbf{A}}_i - h \cdot \nabla e(\bar{\mathbf{A}}_i)^*, \quad \bar{\mathbf{A}}_0 = \mathbf{A}, \quad (19)$$

The descent stops at  $\|\nabla e(\bar{\mathbf{A}})^*\|_F^2 = 0$  or close to it:  $\|\nabla e(\bar{\mathbf{A}})^*\|_F^2 \leq \tau$ ,  $\tau \rightarrow 0$ . To avoid oscillation effects, the step size  $h$  needs to be small, ideally depending on the norm of the gradient  $\nabla e(\bar{\mathbf{A}})^*$ , i.e., the smaller the norm the smaller  $h$  should be. Since the descent is more expensive, we use the closed form solutions for our proof of concept.

In that regard, please note, it is somewhat surprising that not every component of an  $nD$  data point can be manipulated in the same way. In fact, an inherent characteristic - related to the data's **norm** - determines whether data points can be mutually manipulated within the projection space. The norm properties of the data and those of the items in the three domains are interrelated and play a crucial role in how effectively the data can be manipulated, as discussed subsequently.

## 5 DATA NORM-BASED CONSTRAINTS ON MANIPULABILITY: LIMITATIONS AS AN INHERENT PART OF THE DATA MODEL

Let us begin by considering a small shift  $\Delta \mathbf{p}$  of a projection point  $\mathbf{p}$ . Such a shift must be caused by a corresponding small change  $\Delta \mathbf{A}$  in the projection matrix  $\mathbf{A}$  associated with the data point  $\mathbf{d}$  represented by  $\mathbf{p}$ . This relationship is given by  $\Delta \mathbf{p} = \Delta \mathbf{A} \cdot \mathbf{d}$ . Next, consider the norm  $\|\Delta \mathbf{p}\|_2$ , which can be interpreted as the radius within  $\mathbf{p}$  is allowed to move in the projection space. From these considerations, the relation among the norms in the three domains follows as:  $\|\Delta \mathbf{p}\|_2 \approx \|\Delta \mathbf{A}\|_F \cdot \|\mathbf{d}\|_2$  and thus as:

$$\underbrace{\frac{\|\Delta \mathbf{p}\|_2}{\|\Delta \mathbf{A}\|_F}}_{\text{Projections Matrix Space}} \approx \underbrace{\|\mathbf{d}\|_2}_{\text{Data Space}}. \quad (20)$$

Interestingly, the norms in the three domains are mutually coupled. While the data space norm is fixed, the norms in the projection space and the projection matrix space can vary and must satisfy the equation of state in (20). Thus, the data norm sets the limit for the ratio of the other two norms. Technically,  $\|\Delta \mathbf{p}\|_2$  represents the radius over which the projection point  $\mathbf{p}$  shifts in the projection space when the projection matrix  $\mathbf{A}$  contributes a norm  $\|\Delta \mathbf{A}\|_F$ . Following this reasoning, we can assess how changes in the projection matrix affect the shift of a projection point based on the data norm. In short, data points with smaller norms cannot be manipulated as strongly as those with larger norms (since they provide a smaller equilibrium in (20)). We call this the *shift impact* of a data point.

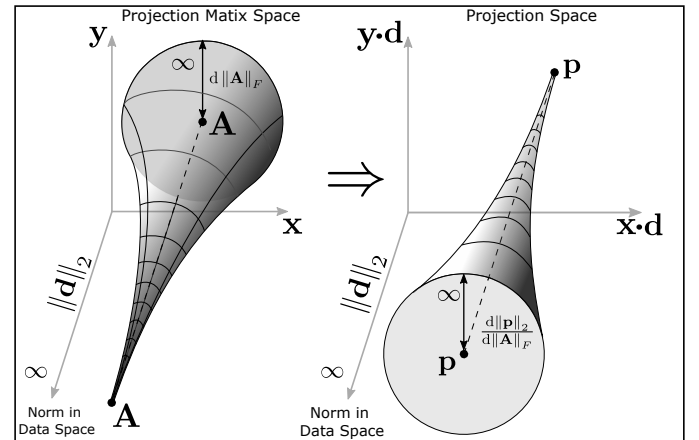


Fig. 6: Shift Impact:  $\|\Delta \mathbf{p}\|_2 / \|\Delta \mathbf{A}\|_F$  for given decreasing  $\|\Delta \mathbf{A}\|_F$  in comparison to an increasing data norm  $\|\mathbf{d}\|_2$ .

**Shift Impact:** A data point  $\mathbf{d}$  with a large norm ( $\|\mathbf{d}\|_2 \rightarrow \infty$ ) produces a maximal shift of the projection  $\mathbf{p}$  for a minimal change

in the projection matrix  $\Delta\mathbf{A}$ , whereas a data point with a small norm ( $\|\mathbf{d}\|_2 \rightarrow 0$ ) yields only a minimal shift of  $\mathbf{p}$  even if  $\Delta\mathbf{A}$  changes considerably. Figure 6 illustrates this effect. In summary, shifting a projection point corresponding to data with a small norm requires a larger adjustment of the projection matrix compared to shifting a projection point associated with data of a larger norm. In other words, projection points for data farther from the origin (i.e., with larger norms) lead to less pronounced changes in the projection matrix than those closer to the origin.

**Recommendation 1 for the work with Composition Operators:**

From the shift impact, it follows that if the user wishes to change the projection as little as possible (in order to best preserve the overall structure), they should manipulate projection points corresponding to data points with larger norms when using Composition Operators.

When manipulating multiple projection points, the shift impact implies that different points may move by varying amounts for the same change in the projection matrix. We refer to this phenomenon as the *shift lack* for multiple points.

**Shift Lack:** We see from (20) that a data point  $\mathbf{d}_{\min}$  with a smaller norm cannot be shifted over the same range  $\Delta\mathbf{p}$  as a data point  $\mathbf{d}_{\max}$  with larger norm point could - by using the same  $\Delta\mathbf{A}$ . with  $\|\mathbf{d}_{\min}\|_2 < \|\mathbf{d}_{\max}\|_2$  which gives

$$\frac{\|\Delta\mathbf{p}_{\max}\|_2}{\|\mathbf{d}_{\max}\|_2} \approx \frac{\|\Delta\mathbf{p}_{\min}\|_2}{\|\mathbf{d}_{\min}\|_2} \rightarrow \|\Delta\mathbf{p}_{\min}\|_2 \approx \|\Delta\mathbf{p}_{\max}\|_2 \cdot \frac{\|\mathbf{d}_{\min}\|_2}{\|\mathbf{d}_{\max}\|_2}.$$

From this follows the difference between the minimum and maximum shift in projection space for two points, denoted as shift lack, as

$$\left( \|\Delta\mathbf{p}_{\max}\|_2 - \|\Delta\mathbf{p}_{\min}\|_2 \right) = \|\Delta\mathbf{p}_{\max}\|_2 \cdot \left( 1 - \frac{\|\mathbf{d}_{\min}\|_2}{\|\mathbf{d}_{\max}\|_2} \right). \quad (21)$$

Informally speaking: some projected points move faster than others in the projection space, when changing the projection matrix.

**Shift Impact & Shift Lack vs. the Norm Equalization of Sec. 4.1:**

We can now come back and explain why norm equalization is helpful. Shifting a set  $\mathbf{M}$  of points does not allow every point  $\mathbf{m}_i$  to be shifted exactly to  $\mathbf{p}_{\mathbf{m}_i} + \Delta\mathbf{p}$  in projection space due to the shift lack (21). This effect increases with the range between the data points  $\mathbf{m}_{\max}$  with the largest norm  $\|\mathbf{m}_{\max}\|_2$  and  $\mathbf{m}_{\min}$  with the smallest norm  $\|\mathbf{m}_{\min}\|_2$ . To control the impact of a shift on the projection/projection matrix, the *Norm Equalization* was introduced to blend the most and least distant points. However, note that only the shift impact can be controlled; the shift lack is inherent to the dataset and cannot be prevented. Therefore, the data norm ultimately determines and limits the quality of the interactions available to the user.

## 6 EVALUATION

In the following section, we examine the key properties of our Composition Operators, which enable an interactive, intuitive, and rapid visual search for optimal projection parameters in the linear mapping of high-dimensional data. For further details, please refer to the accompanying video and additional material.

### 6.1 Evaluation of Quality Terms versus Limitations Based on Data Norms

**Shifting a set of points:** We consider a set of  $q$  points  $\mathbf{M}$  while simultaneously trying to keep another set of  $k$  points  $\mathbf{Q}$  as fixed as possible. A good fit is given if the shift and the stickiness quality minimizes (cf. Sec. 4). We subsequently analyze the properties of the shift and the stickiness quality. We start with the shift quality and get the estimate

$$e_{\mathbf{M}}(\bar{\mathbf{A}}) \leq q^2 \beta^2 \|\Delta\mathbf{p}\|_2^2 \left( \frac{1}{\beta^2} + \frac{(\sigma_{\min}(\mathbf{M}) + \delta_{\mathbf{M}})^4}{(\alpha + \beta \sigma_{\min}^2(\mathbf{M}) + \gamma \sigma_{\min}^2(\mathbf{Q}))^2} - \frac{2}{q\beta} \frac{\sigma_{\min}^2(\mathbf{M})}{\alpha + \beta(\sigma_{\min}(\mathbf{M}) + \delta_{\mathbf{M}})^2 + \gamma(\sigma_{\min}(\mathbf{Q}) + \delta_{\mathbf{Q}})^2} \right) \quad (22)$$

where  $\sigma_{\min}(\mathbf{M})$  denotes the minimum singular value of  $\mathbf{M}$  and  $\delta_{\mathbf{M}} = \sigma_{\max}(\mathbf{M}) - \sigma_{\min}(\mathbf{M})$  denotes the difference between the maximum and minimum singular value.  $\delta_{\mathbf{Q}}$  is given analogously. The proof of (22)

can be found in the additional material. Note that (22) relates to three quantities that characterize the point cloud  $\mathbf{M}$  in data space: the number of points  $q$ , the minimum singular value  $\sigma_{\min}(\mathbf{M})$  and the diameter of the spectrum  $\delta_{\mathbf{M}}$ . Using this, one can analyze several scenarios:

1. (a) Suppose  $\delta_{\mathbf{M}}$  is small compared to  $\sigma_{\min}(\mathbf{M})$ . Then, the expected shift error is dominated by  $\sigma_{\min}(\mathbf{M})$ . For small  $\sigma_{\min}(\mathbf{M})$  we have

$$e_{\mathbf{M}}(\bar{\mathbf{A}}) \leq q^2 \|\Delta\mathbf{p}\|_2^2. \quad (23)$$

A large minimum singular value of  $\mathbf{M}$  will cause the shift error to

$$e_{\mathbf{M}}(\bar{\mathbf{A}}) \leq 2q^2 \left(1 - \frac{1}{q}\right) \|\Delta\mathbf{p}\|_2^2. \quad (24)$$

2. (a) If the gap  $\delta_{\mathbf{M}}$  is large compared to  $\sigma_{\min}(\mathbf{M})$  the expected shift error is expected to be large since this means that the points of  $\mathbf{M}$  are distributed in an ellipsoid that is very long in one direction and very slim in another direction in data space. Specifically, we have for small  $\sigma_{\min}(\mathbf{M})$  the estimate

$$e_{\mathbf{M}}(\bar{\mathbf{A}}) \leq q^2 \left( 1 + \frac{\delta_{\mathbf{M}}^4}{(\alpha + \gamma \sigma_{\min}^2(\mathbf{Q}))^2} \beta^2 \right) \|\Delta\mathbf{p}\|_2^2. \quad (25)$$

For large  $\sigma_{\min}(\mathbf{M})$  estimate (24) holds.

3. (a) Estimate (22) shows that the error for the shift is expected to be smaller if the data points in  $\mathbf{Q}$  are uniformly distributed, i.e., if  $\delta_{\mathbf{Q}}$  is small and if they have a small minimum singular value.

With the same methods a similar estimate for the stickiness error can be shown to take the form

$$e_{\mathbf{Q}}(\bar{\mathbf{A}}) \leq q^2 \beta^2 \|\Delta\mathbf{p}\|_2^2 \left( \frac{(\sigma_{\min}(\mathbf{M}) + \delta_{\mathbf{M}})^2 (\sigma_{\min}(\mathbf{Q}) + \delta_{\mathbf{Q}})^2}{(\alpha + \beta \sigma_{\min}^2(\mathbf{M}) + \gamma \sigma_{\min}^2(\mathbf{Q}))^2} \right) \quad (26)$$

This implies the following for the user:

1. (b) If  $\delta_{\mathbf{M}}$  or  $\delta_{\mathbf{Q}}$  are large compared to either  $\sigma_{\min}(\mathbf{M})$  or  $\sigma_{\min}(\mathbf{Q})$  the stickiness error is expected to be large.
2. (b) If  $\sigma_{\min}(\mathbf{M})$  becomes large the stickiness error is essentially bounded by  $e_{\mathbf{Q}}(\bar{\mathbf{A}}) \leq q^2 \|\Delta\mathbf{p}\|_2^2 (\sigma_{\min}^2(\mathbf{Q}) + \delta_{\mathbf{Q}}^2)$ . In this case, a large  $\sigma_{\min}(\mathbf{Q})$  or a large  $\delta_{\mathbf{Q}}$  can lead to a large stickiness error.
3. (b) If  $\sigma_{\min}(\mathbf{Q})$  becomes large, we essentially have  $e_{\mathbf{Q}}(\bar{\mathbf{A}}) \leq q^2 \beta^2 / \gamma^2 \|\Delta\mathbf{p}\|_2^2 (\sigma_{\min}^2(\mathbf{M}) + \delta_{\mathbf{M}}^2)$ . If both  $\sigma_{\min}(\mathbf{M})$  and  $\sigma_{\min}(\mathbf{Q})$  are large and of the same order of magnitude then we have  $e_{\mathbf{Q}}(\bar{\mathbf{A}}) \leq q^2 \frac{\beta^2}{\beta^2 + \gamma^2} \|\Delta\mathbf{p}\|_2^2$ .

To sum up, with 2(a) and 3(a), the shift quality is small if the data norm differences of the data in  $\mathbf{M}$  and  $\mathbf{Q}$  are small, uniformly distributed, and the minimal norm in  $\mathbf{Q}$  is small, too. With 2(b) and 3(b), the stickiness quality is small if the minimal norm in  $\mathbf{M}$  is large. Thus, the minimal norm of  $\mathbf{M}$  should be larger than the minimal norm of  $\mathbf{Q}$ . Based on our evaluation of the quality terms with respect to the data norms, the following recommendation for the optimal use of the Composition Operators can be directly derived:

**Recommendation 2 for the work with Composition Operators:** In order to minimize manipulation energy and achieve optimal shift results, the user should select data points that exhibit comparably large data norms for shift-able points in  $\mathbf{M}$  and comparably small data norms for sticky points in  $\mathbf{Q}$ .

Note that as the dimensionality  $n$  increases, the data distributions tend to become more uniform [11, 28], implying that the effectiveness of Recommendation 2 gradually diminishes for large  $n$ . This observation is consistent with the notion that the degrees of freedom for Composition Operators increase as  $n$  grows.

### 6.2 Evaluation for Visual Support to Perceive Data Norms

As seen, data norms are essential for effective use of Composition Operators, especially when selecting shift-able and sticky points (cf. Recommendations 1 and 2). To support norm-aware selections, we provide visualizations that encode data norms directly in the projection.

Figure 7 illustrates this for the WDBC dataset. The *Traditional View* (top-left) shows a standard projection of all data points. The

*Data Norm View* (top-center) maps each point’s norm to its circle size: larger norms result in larger circles. The *Color-Coded View* (top-right) additionally encodes norms by color (e.g., yellow for low norms, purple for high norms), which further reduces clutter. The *Relevance View* (bottom-right) focuses on the selected shift-able (blue) and sticky (red) points, hiding all others. The *Data Norm View + Shift-Able/Sticky Points* (bottom-left) shows the same selection but also includes the unselected points in black.

These views enable users to quickly assess data norms and apply selection strategies effectively. The system also provides direct feedback on composition feasibility based on the data norms.

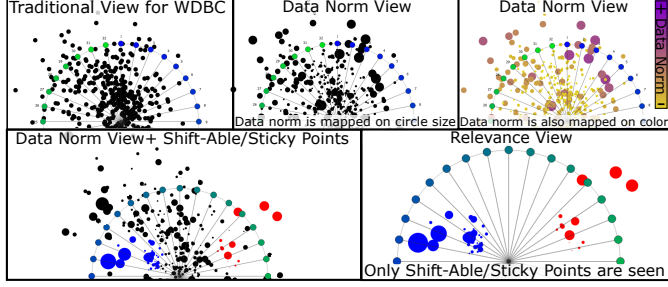


Fig. 7: Norm-related views to the WDBC data.

### 6.3 Qualitative Evaluation

**Properties of Composition Operators:** In Figure 8, we examine the properties of the Composition Operators using the Iris (see Fig. 8 (1 - 12)) and Yeast datasets (see Fig. 8 (13 - 24)) from the UCI Repository [18]. Orange points indicate *sticky* points and their current positions under interaction, while blue points represent *shift-able* points.

*With/Without sticky points:* Since shift operations with  $\Delta p$  affect only *shift-able* points, they are central to our technique. When applied without *sticky* points (see Fig. 8 (1 - 6) and (13 - 17)), these operations replace points (while the projection parameters are updated), similar to a conventional ArcBall interaction, where a user intuitively rotates a 3D object by dragging a virtual trackball with the mouse, mapping 2D input to 3D orientation changes [66]. This enables the formulation of hypotheses about cluster structures in the data and facilitates insights into the sub-space in which the selected points are embedded.

*With/Without Memory:* Without the memory effect, the sticky points move with the direction of the shift-able points, but at a slower pace. As a result, the separation between sticky and shift-able points increases over time. In contrast, using the memory effect (see Sec. 4) induces a continuous restoring motion of the sticky points back toward their memory positions, similar to a spring-force model. This minimizes the sticky bias and results in an adaptation of the projection parameters. To verify this, compare Fig. 8 (7) with (11) and Fig. 8 (19) with (23).

*Blending Parameter  $\delta$ :* It controls the stiffness, emphasizing either the memory effect (by penalizing the shift) or vice versa. The user can adjust this parameter to focus on the desired aspect. In our experience,  $\delta = 1$  works best for achieving common goals; however, tweaking  $\delta$  may enhance the projection composition quality in certain cases (see Fig. 8 (7 - 9) and Fig. 8 (19 - 21)).

*Fixing the Axes:* Fixing axes is a useful tool for feature-related analysis. For the Iris dataset, axes 3 and 4 are fixed, and for the Yeast dataset, axes 4 through 8 are fixed. Compare Fig. 8 (7) with (10) and Fig. 8 (19) with (22). While this reduces the degrees of freedom — resulting in stronger adaptation of the remaining axes and occasionally preventing the desired shift — the axes do not degenerate over time due to error accumulation. In fact, returning shift points to their original positions yields axis configurations that closely resemble the initial ones. Further details to this can be found in the additional material.

**Composing Class Separation:** By toggling *shift-able* and *sticky* items and shifting them appropriately, our approach supports complex analytical operations. A key example is the separation of classes in the projection space — a common scenario in visual analytics. Using the composition tools described above, our method can quickly yield well-separated classes. Figure 9 demonstrates this with the Census, Wine, and Sponge datasets (classes are color-coded). At the top, different

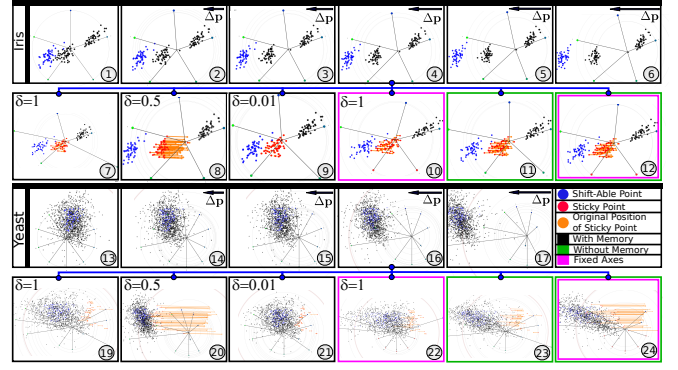


Fig. 8: Composition Operator properties w.r.t. memory effect, use of the blending parameter, axis fixation, and the interplay between *shift-able* and *sticky* points.

classes overlap; the goal is to separate one or more classes as effectively as possible. At the bottom, our operators resolve the overlap on demand. For further details, please refer to the additional video.

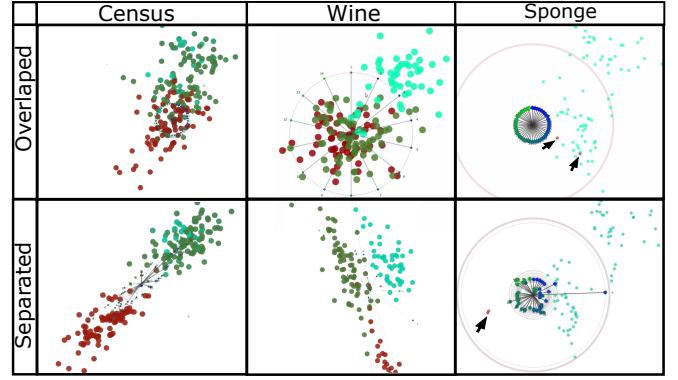


Fig. 9: Composition Operators for (color-coded) class separation.

**Composing Minimum Separation:** Composition operators, used directly in projection space, make formerly complex visual tasks intuitive. In Fig. 10, a user analysis process is emulated: finding a projection that minimizes separation among items of interest (orange circles) to cluster them. Traditionally difficult due to the large parameter space, this task becomes effortless with *shift-able* and *sticky* points.

Let us focus on the top row of Fig. 10 from left to right: The following sequence shows the user’s steps with Composition Operators to find a projection where the four points of interest (orange circles) have minimal separation.

- Fig. 10 (1): The user identifies the goal of clustering the four marked points of interest (orange circles).
- Fig. 10 (2): The user marks two points of interest as *shift-able* (blue) and one as *sticky* (red).
- Fig. 10 (3): The user moves the shift-able points along the blue arrow  $\Delta p$ ; the shown underlying anchor points adjust to reflect the updated projection.
- Fig. 10 (4): The user continues moving the shift-able points until those two and the sticky point of interest form a cluster; the anchor points update accordingly.
- Fig. 10 (5): The user sets the three clustered points of interest to *sticky* (red) and the remaining point of interest to *shift-able* (blue).
- Fig. 10 (6): The user moves the last shift-able point of interest along the blue arrow  $\Delta p$  until it joins the sticky points of interest; anchor points adapt as before.
- Fig. 10 (7): The process completes; all *shift-able* and *sticky* properties are removed, leaving the four points of interest clustered, with minimal separation. The user can now read the projection parameters from the anchor configuration.

Let us now focus on the bottom row of Fig. 10 from left to right: A plain visualization of projection parameters via anchor points is shown

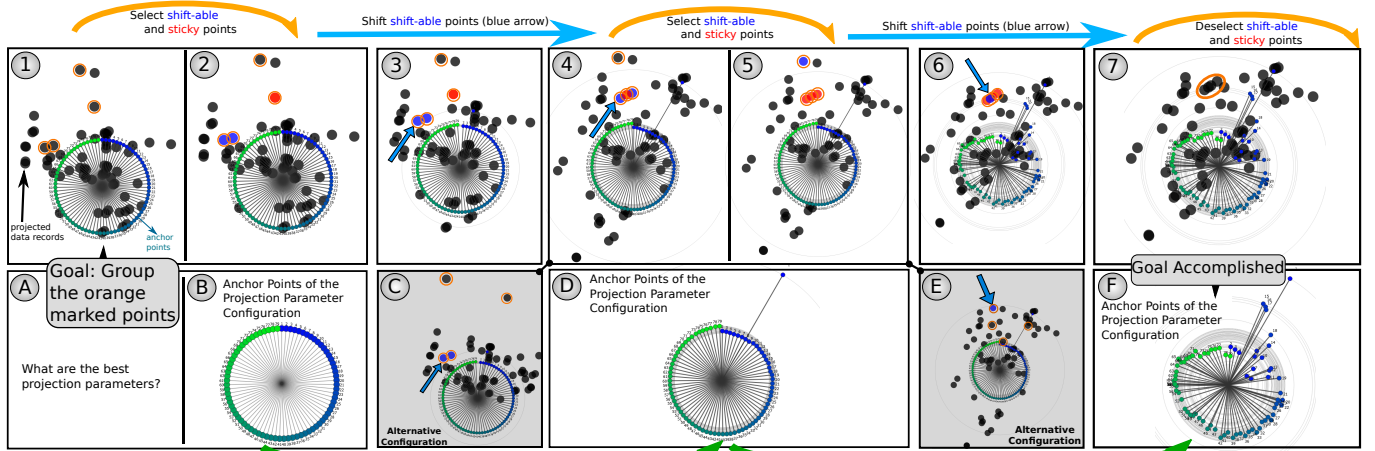


Fig. 10: User Analysis Process for composing a minimal separation between four points of interest (orange-colored circles) by Composition Operators.

in Fig. 10 (B), (D), and (F) (related to steps of Fig. 10 (2), (4), and (7)), illustrating how parameters adapt and confirm the final clustering of points of interest. Additionally, Fig. 10 (A) restates the analyst’s goal.

Fig. 10 (C) offers a “what-if” on step (3): without marking a point of interest as *sticky*, the *shift-able* point of interest would not move closer to them in step (4). Fig. 10 (E) shows a “what-if” on step (6): without the marking from step (5), the points of interest would drift apart when moving the shift-able point of interest. These “what-if” cases highlight that while many Operator sequences can achieve the goal, some fail; iterative, successive use reliably guided the analyst to the solution in all our experiments.

The entire process took less than 20 seconds. Compared to traditional approaches (cf. Sec. 2), composing the projection in this manner dramatically speeds up the visual analysis process. Further examples of the Composition Operator’s application in various use-case scenarios can be found in the additional material.

## 7 DISCUSSION & LIMITATIONS

**Performance:** Since computing the Composition Operator  $\bar{\mathbf{A}}$  scales only with the sizes of the sets  $q$  and  $k$ , and since the inversion of  $\mathbf{H}$  depends solely on the dimensionality  $n$  (cf. Sec. 4), which is fixed for a given  $n$ D dataset, we achieved interactive rates (over 100 fps) on our 2.7 GHz machine for all our experiments and benchmark datasets.

**Shift Path Dependency:** Let  $\bar{\mathbf{A}}_{start}$  denote the initial projection matrix, and let  $\mathbf{p}(t)$  be a 2D curve running from  $\mathbf{p}(0) = \mathbf{p}_{start}$  to  $\mathbf{p}(1) = \mathbf{p}_{end}$ , with  $\bar{\mathbf{A}}_{start}$  associated with  $\mathbf{p}_{start}$ . Our operation steps,  $\Delta\mathbf{p}$ , are obtained by sampling this curve for  $t \in [0, 1]$ . Assuming sufficiently small steps, the resulting projection matrix  $\bar{\mathbf{A}}_{end}$  at  $\mathbf{p}_{end}$  depends on the chosen path  $\mathbf{p}(t)$  — indicating that many projection matrices can optimally reconstruct the set of point manipulations. Moreover, if the curve is closed (i.e.,  $\mathbf{p}_{start} = \mathbf{p}_{end}$ ), then in general  $\bar{\mathbf{A}}_{start} \neq \bar{\mathbf{A}}_{end}$ . However, our experience shows that these projections are quite similar, ensuring a stable composition process (see the degeneration discussion in the *additional material*).

**Bivariate Scatterplots vs. Multivariate Star Coordinates:** Since scatterplots offer only two degrees of freedom [21], only the one-point shift operation of Sec. 4 can be applied. In contrast, all additional Operators require multivariate Star Coordinates [37] with  $n > 2$ .

**Differences to Related Interactive Projection Scheme.** Molchanov and Linsen [52] let users tune projection parameters by dragging control points (often class centroids) in the projection space. Our Composition Operator framework differs in four main aspects:

- Well-posed for high dimensions.* When the data dimensionality exceeds the number of classes, the equations in [52] become under-determined; they resolve this by adding random auxiliary points, which may alter the semantics. Our formulation stays well defined without such points.
- Item-level versus class-level control.* We assign the “shift-able” or “sticky” status independently to every point, enabling both individual and group manipulations, with or without class labels.
- Sticky points.* The framework supports points that remain fixed while others move, so users can iteratively test hypotheses without

losing established structures.

- Distortion-aware axes.* Axes are constrained by Orthographic Star Coordinates [43], preventing axis degeneration and mitigating projection distortions observed when axis length is unconstrained.

**Limitations in Linear vs. Projective Projections:** Our operators cannot be directly applied to projective projections (e.g., Rad-Vis [30, 45]) because these techniques introduce non-linear distortions. In such cases, shifting a point  $\mathbf{p}$  to  $\mathbf{p} + \Delta\mathbf{p}$  depends non-linearly on both the data norm  $\|\mathbf{d}\|_2$  and the current position  $\mathbf{p}$  in the projection space, posing a higher-order issue. Consequently, defining a single linear operation  $\mathbf{A} + \Delta\mathbf{A}$  that can shift various data records is challenging.

**Limitations in Linear vs. Non-Linear Projections:** Our method does not readily extend to nonlinear embeddings like SNE [29], whose iterative, non-analytic nature precludes closed-form solutions. Moreover, we favor linear embeddings, as they enable a clearer analysis of dimensional properties.

**Limitations in Data Dimensions:** If the number  $n$  of dimensions is small, the system becomes over-constrained, and shift operations may not be as accurate as desired.

## 8 CONCLUSION

In summary, we addressed the challenge of uncontrollable projection outcomes by developing a mathematically grounded framework that enables targeted, interactive adjustments in multivariate linear projections for both classified and unclassified data — eliminating the trial-and-error nature of previous approaches. To this end, we introduce interactive Composition Operators that empower users to find informative projections. Our Composition Operator Toolbox provides, to the best of our knowledge, the most comprehensive set of interactive tools available for supporting the user’s search for informative projections, including functionalities such as

- (1) *sticky points*,
  - (2) *shift-able points*,
  - (3) *restrict anchor points to be movable or not*, enabling both an
  - (4) *item-based visual analysis* and a
  - (5) *group-based visual analysis*,
  - (6) *with and without a memory function*, a support for
  - (7) *distortion minimizing projections* (avoiding axes degeneration) &
  - (8) *class separating projections*.
- In addition, we derived
- (9) *recommendations* for the user on how to work with
- Composition Operators in practice, and we provided
- (10) *closed form solutions* and
  - (11) *numerical solutions*

in order to support the use of Composition Operators across various computing environments. Although our approach is limited to linear embeddings and cannot be trivially extended to nonlinear methods, it facilitates the discovery of informative projections for both classified and unclassified  $n$ D data. While this work primarily focuses on linear shift operations, we plan to extend our approach to include shifting curves in the future.

## REFERENCES

- [1] P. K. Agarwal. Relative neighborhood graphs in three dimensions. *Computational Geometry*, 1992. 2
- [2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 1998. 2
- [3] G. Albuquerque, M. Eisemann, D. J. Lehmann, H. Theisel, and M. Magnor. Quality-based visualization matrices. In *Proceedings of Vision, Modeling, and Visualization*, 2009. 2
- [4] G. Albuquerque, M. Eisemann, D. J. Lehmann, H. Theisel, and M. Magnor. Improving the visual analysis of high-dimensional datasets using quality measures. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, 2010. 2
- [5] E. Amorim, E. Brazil, J. Daniels, P. Joia, L. Nonato, and M. Sousa. ilamp: Exploring high-dimensional spacing through backward multidimensional projection. *IEEE Conference on Visual Analytics Science and Technology*, 2012. 2
- [6] G. Appleby, M. Espadoto, R. Chen, S. Goree, A. Telea, E. Anderson, and R. Chang. Hypermp: Interactive visual exploration of multidimensional projection hyperparameters. *Computer Graphics Forum*, 2022. 2
- [7] A. Artero, M. de Oliveira, and H. Levkowitz. Enhanced high dimensional data visualization through dimension reduction and attribute arrangement. *International Conference on Information Visualisation*, 2006. 2
- [8] D. Asimov. The grand tour: a tool for viewing multidimensional data. *Journal on Scientific and Statistical Computing*, 1985. 1, 2
- [9] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, 2001. 2
- [10] E. Bertini. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 2011. 2
- [11] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is “nearest neighbor” meaningful? In *Proceedings of the 7th International Conference on Database Theory*, 1999. 2, 7
- [12] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2012. 1, 2
- [13] A. Chatzimpampas, R. Martins, and A. Kerren. t-visne: Interactive assessment and interpretation of t-sne projections. *IEEE Transactions on Visualization and Computer Graphics*, 04 2020. 2
- [14] W. S. Cleveland. Visualizing data. *Hobart Press, Summit, New Jersey*, 1993. 2
- [15] D. Cook, A. Buja, J. Cabreta, and C. Hurley. Grand tour and projection pursuit. *Journal of Computational and Statistical Computing*, 1995. 1, 2
- [16] D. B. Dimitris K. Agrifiotis and E. Yang. Stochastic proximity embedding: A simple, fast and scalable algorithm for solving the distance geometry problem. *Distance Geometry*, 2013. 2
- [17] M. Dowling, J. Wenskovitch, J. Fry, S. Leman, L. House, and C. North. Sirius: Dual, symmetric, interactive dimension reductions. *IEEE Transactions on Visualization and Computer Graphics*, 2018. 2
- [18] D. Dua and C. Graff. UCI machine learning repository, 2017. 8, 16
- [19] N. Elmquist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 2008. 2
- [20] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. *VAST 2011 - IEEE Conference on Visual Analytics Science and Technology 2011, Proceedings*, pages 121 – 130, 11 2011. 2
- [21] M. Fink, J.-H. Haurert, J. Spoerhase, and A. Wolff. Selecting the aspect ratio of a scatter plot based on its delaunay triangulation. *IEEE Transactions on Visualization and Computer Graphics*, 2013. 2, 9
- [22] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 1987. 1, 2
- [23] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, 1974. 1, 2
- [24] T. Fujiwara, X. Wei, J. Zhao, and K.-L. Ma. Interactive dimensionality reduction for comparative analysis. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):758–768, Jan. 2022. 1, 2
- [25] K. R. Gabriel. The biplot graphic display of matrices with application to principal component analysis 1. *Biometrika*, 1971. 1, 2
- [26] A. Ghosh, M. Nashaat, J. Miller, and S. Quader. Interpretation of structural preservation in low-dimensional embeddings. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 3
- [27] M. Gleicher. Explainers: Expert explorations with crafted projections. *IEEE Transactions on Visualization and Computer Graphics*, 2013. 2
- [28] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases*, 2000. 2, 7
- [29] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, 2002. 2, 9
- [30] P. Hoffman, G. Grinstein, and D. Pinkney. Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In *NPIVM '99*, pages 9–16, New York, NY, USA, 1999. ACM. 9
- [31] X. Hu, L. Bradel, D. Maiti, L. House, C. North, and S. Leman. Semantics of directly manipulating spatializations. *IEEE Transactions on Visualization and Computer Graphics*, 2013. 1, 2
- [32] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel mds on the gpu. *IEEE Transactions on Visualization and Computer Graphics*, 2009. 2
- [33] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1985. 2
- [34] A. Inselberg. Parallel coordinates. *Springer Berlin*, 2009. 2
- [35] P. Joia, F. V. Paulovich, J. A. Cuminato, D. Coimbra, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 2011. 2
- [36] B. M. A. T. Julian Stahnke, Marian Dörk. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Transactions on Visualization & Computer Graphics*, 2015. 2
- [37] E. Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium*, 2000. 1, 2, 9
- [38] E. Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. *Proc. of the seventh ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, 2001. 1, 2
- [39] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE Transactions on Visualization and Computer Graphics*, 2016. 1, 2
- [40] D. J. Lehmann, G. Albuquerque, M. Eisemann, M. Magnor, and H. Theisel. Selecting coherent and relevant plots in large scatterplot matrices. *Computer Graphics Forum*, 2012. 2
- [41] D. J. Lehmann, S. Hundt, and H. Theisel. A study on quality metrics vs. human perception: Can visual measures help us to filter visualizations of interest? *Journal of Information Technology: Special Issue Visual Analytics*, 2015. 2
- [42] D. J. Lehmann, F. Kemmler, T. Zhyhalava, M. Kirschke, and H. Theisel. Visualnostics: Visual guidance pictograms for analyzing projections of high-dimensional data. *Computer Graphics Forum*, 2015. 2
- [43] D. J. Lehmann and H. Theisel. Orthographic star coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 2013. 1, 2, 6, 9, 16
- [44] D. J. Lehmann and H. Theisel. Optimal sets of projections of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 2015. 1, 2
- [45] D. J. Lehmann and H. Theisel. General projective maps for multidimensional data projection. *Computer Graphics Forum*, 2016. 1, 2, 9
- [46] J. Li and C.-q. Zhou. Incorporation of human knowledge into data embeddings to improve pattern significance and interpretability. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2
- [47] S. Liu, P.-T. Bremer, J. J. Jayaraman, B. Wang, B. Summa, and V. Pascucci. The grassmannian atlas: A general framework for exploring linear projections of high-dimensional data. *Computer Graphics Forum*, 2016. 2
- [48] S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci. Visual Exploration of High-Dimensional Data through Subspace Analysis and Dynamic Projections. *Computer Graphics Forum*, 2015. 2
- [49] G. Mamani, F. Fatore, L. Nonato, and F. Paulovich. User-driven feature space transformation. *Computer Graphics Forum*, 32(3):291–299, 2013. 2
- [50] J. Matute and L. Linsen. Hinted star coordinates for mixed data. *Computer Graphics Forum*, 39(1):117–133, 2020. 2
- [51] V. Molchanov, S. Hamid, and L. Linsen. Efficient morphing of shape-preserving star coordinates. In *IEEE Pacific Visualization Symposium*, 2020. 2
- [52] V. Molchanov and L. Linsen. Interactive Design of Multidimensional Data Projection Layout. In *EuroVis - Short Papers*, 2014. 1, 3, 9, 16, 17
- [53] V. Molchanov and L. Linsen. Shape-preserving star coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 2019. 2

- [54] L. Nonato and M. Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1, 06 2018. 2
- [55] A. C. T. Paulo E. Rauber, Alexandre X. Falcão. Visualizing time-dependent data using dynamic t-sne. *Computer Graphics Forum*, 2016. 2
- [56] F. V. Pavlovich, D. M. Eler, J. Poco, C. P. Botha, R. Minghim, and L. G. Nonato. Piecewise Laplacian-based Projection for Interactive Data Exploration and Organization. *Computer Graphics Forum*, 2011. 2
- [57] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 2000. 2
- [58] M. Rubio-Sánchez, A. Sanchez, and D. J. Lehmann. Adaptable radial axes plots for improved multivariate data visualization. *Computer Graphics Forum*, 2017. 1, 2
- [59] M. Scherer, J. Bernard, and T. Schreck. Retrieval and exploratory search in multivariate research data repositories using regression features. In *Proc. International ACM/IEEE Joint Conference on Digital Libraries*, 2011. 2
- [60] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatter-plot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics*, 2013. 2
- [61] J. Self, M. Dowling, J. Wenskovitch, I. Crandell, M. Wang, L. House, S. Leman, and C. North. Observation-level and parametric interaction for high-dimensional data analysis. *ACM Transactions on Interactive Intelligent Systems*, 8:1–36, 06 2018. 1, 2
- [62] J. Self, R. Vinayagam, J. Fry, and C. North. Bridging the gap between user intention and model parameters for human-in-the-loop data analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 2016. 1, 2
- [63] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 2005. 2
- [64] L. Shao, M. Behrisch, T. Schreck, T. von Landesberger, M. Scherer, S. Bremm, and D. A. Keim. Guided sketching for visual search and exploration in large scatter plot spaces. 2014. 2
- [65] L. Shao, A. Mahajan, T. Schreck, and D. J. Lehmann. Interactive regression lens for exploring scatter plots. *Computer Graphics Forum (EuroVis)*, 2017. 1
- [66] K. Shoemake. ARCBALL: A user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of Graphics Interface*, pages 151–156. Canadian Human-Computer Communications Society, 1992. 8
- [67] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 2009. 2, 16
- [68] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès. Robust subspace clustering. *CoRR*, 2013. 2
- [69] A. Tatu, F. Maaß, I. Färber, E. Bertini, T. Schreck, T. Seidl, and D. A. Keim. Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, 2012. 2
- [70] S. T. Teoh and K.-L. Ma. Starclass: Interactive visual classification using star coordinates. In *SDM*, 2003. 1
- [71] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 1980. 2
- [72] L. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 2008. 2
- [73] L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative review. *J Mach Learn Res*, 2009. 2
- [74] Y. Wang, J. Li, F. Nie, H. Theisel, M. Gong, and D. J. Lehmann. Linear discriminative star coordinates for exploring class and cluster separation of high dimensional data. *Computer Graphics Forum*, 2017. 1
- [75] J. Wei, D. Xia, H. Xie, C.-M. Chang, C. Li, and X. Yang. Spaceediting: A latent space editing interface for integrating human knowledge into deep neural networks. In *Proceedings of the 29th International Conference on Intelligent User Interfaces, IUI '24*, page 489–503, New York, NY, USA, 2024. Association for Computing Machinery. 2

# ADDITIONAL MATERIAL FOR INTERACTIVE COMPOSITION OPERATORS

## Proof and Derivations

### Proof of Gradient in Eq. (8) and Eq. (9):

With  $e(\bar{\mathbf{A}})^* = \alpha e_{\mathbf{A}} + \beta e_{\mathbf{M}} + \gamma e_{\mathbf{Q}}^*$  of Eq. (7) gives  $\nabla e(\bar{\mathbf{A}})^*$  as

$$\nabla e(\bar{\mathbf{A}})^* = \alpha \nabla e_{\mathbf{A}} + \beta \nabla e_{\mathbf{M}} + \gamma \nabla e_{\mathbf{Q}}^* \quad (27)$$

giving the independent terms of  $\nabla e(\bar{\mathbf{A}})^*$  as

$$\alpha \nabla e_{\mathbf{A}} = \frac{d(\alpha(\|\bar{\mathbf{A}} - \mathbf{A}\|_F^2))}{d\bar{\mathbf{A}}} \quad (28)$$

$$= \alpha \cdot 2 \cdot (\bar{\mathbf{A}} - \mathbf{A}) \quad (29)$$

and

$$\beta \nabla e_{\mathbf{M}} = \frac{d(\beta \cdot (\sum_{i=1}^q \|(\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{m}_i - \Delta \mathbf{p}\|_2^2))}{d\bar{\mathbf{A}}} \quad (30)$$

$$= \beta \cdot \sum 2 \cdot ((\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{m}_i - \Delta \mathbf{p}) \cdot \mathbf{m}_i^T \quad (31)$$

$$= \beta \cdot 2 \cdot \sum ((\bar{\mathbf{A}} \cdot \mathbf{m}_i) \cdot \mathbf{m}_i^T - (\mathbf{A} \cdot \mathbf{m}_i) \cdot \mathbf{m}_i^T - \Delta \mathbf{p} \cdot \mathbf{m}_i^T) \quad (32)$$

$$= 2\beta \cdot \sum (\bar{\mathbf{A}} - \mathbf{A}) \cdot (\mathbf{m}_i \cdot \mathbf{m}_i^T) - 2\beta \sum \Delta \mathbf{p} \cdot \mathbf{m}_i^T \quad (33)$$

$$= 2\beta \cdot (\bar{\mathbf{A}} - \mathbf{A}) \cdot \underbrace{\sum (\mathbf{m}_i \cdot \mathbf{m}_i^T)}_{\mathbf{M} \cdot \mathbf{M}^T} - 2\beta \cdot \Delta \mathbf{p} \cdot \underbrace{\sum \mathbf{m}_i^T}_{(\mathbf{M} \cdot \mathbf{1})^T} \quad (34)$$

$$= 2\beta \cdot (\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{M} \cdot \mathbf{M}^T - 2\beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T \quad (35)$$

and

$$\gamma \nabla e_{\mathbf{Q}}^- = \frac{d(\gamma \cdot (\sum_{j=1}^k \|(\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{q}_j\|_2^2))}{d\bar{\mathbf{A}}} \quad (36)$$

$$= \gamma \cdot \sum 2((\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{q}_j) \cdot \mathbf{q}_j^T \quad (37)$$

$$= \gamma \cdot 2 \cdot \sum ((\bar{\mathbf{A}} \cdot \mathbf{q}_j) \cdot \mathbf{q}_j^T - (\mathbf{A} \cdot \mathbf{q}_j) \cdot \mathbf{q}_j^T) \quad (38)$$

$$= \gamma \cdot 2 \cdot \sum ((\bar{\mathbf{A}} - \mathbf{A}) \cdot (\mathbf{q}_j \cdot \mathbf{q}_j^T)) \quad (39)$$

$$= \gamma \cdot 2 \cdot (\bar{\mathbf{A}} - \mathbf{A}) \cdot \underbrace{\sum (\mathbf{q}_j \cdot \mathbf{q}_j^T)}_{\mathbf{Q} \mathbf{Q}^T} \quad (40)$$

$$= \gamma \cdot 2 \cdot (\bar{\mathbf{A}} - \mathbf{A}) \cdot \mathbf{Q} \mathbf{Q}^T. \quad (41)$$

and

$$\gamma \nabla e_{\mathbf{Q}}^+ = \frac{d(\gamma \cdot (\sum_{j=1}^k \|(\bar{\mathbf{A}} \cdot \mathbf{q}_j - \mathbf{p}_{\mathbf{q}_j})\|_2^2))}{d\bar{\mathbf{A}}} \quad (42)$$

$$= \gamma \cdot \sum 2(\bar{\mathbf{A}} \cdot \mathbf{q}_j - \mathbf{p}_{\mathbf{q}_j}) \cdot \mathbf{q}_j^T \quad (43)$$

$$= \gamma \cdot 2 \sum (\bar{\mathbf{A}} \cdot \mathbf{q}_j \cdot \mathbf{q}_j^T - \mathbf{p}_{\mathbf{q}_j} \cdot \mathbf{q}_j^T) \quad (44)$$

$$= (\gamma \cdot 2 \cdot \sum \bar{\mathbf{A}} \cdot \mathbf{q}_j \cdot \mathbf{q}_j^T) - (\gamma \cdot 2 \cdot \sum \underbrace{\mathbf{p}_{\mathbf{q}_j} \cdot \mathbf{q}_j^T}_{\mathbf{P}_{\mathbf{q}} \mathbf{Q}^T}) \quad (45)$$

$$= (\gamma \cdot 2 \cdot \bar{\mathbf{A}} \cdot \underbrace{\sum \mathbf{q}_j \cdot \mathbf{q}_j^T}_{\mathbf{Q} \mathbf{Q}^T}) - (\gamma \cdot 2 \cdot \mathbf{P}_{\mathbf{q}} \mathbf{Q}^T) \quad (46)$$

$$= \gamma \cdot 2 \cdot (\bar{\mathbf{A}} \cdot \mathbf{Q} \mathbf{Q}^T - \mathbf{P}_{\mathbf{q}} \mathbf{Q}^T). \quad (47)$$

Putting (29), (35), and (41) in (27) gives (8); putting (29), (35), and (47) in (27) gives (9) ■

### Proof of Projection Composition Operator in Eq. (10):

The optimal projection  $\bar{\mathbf{A}}^-$  is given when the gradient of (8) vanishes with

$$\nabla e(\bar{\mathbf{A}})^- = 0 \quad (48)$$

$$\begin{bmatrix} \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T \\ 2 \times 1 \end{bmatrix} = \underbrace{(\bar{\mathbf{A}} - \mathbf{A})}_{2 \times n} \cdot \underbrace{\left( \underbrace{\beta \cdot \mathbf{M} \mathbf{M}^T}_{n \times n} + \underbrace{\gamma \cdot \mathbf{Q} \mathbf{Q}^T}_{n \times n} + \underbrace{\alpha \cdot \mathbf{I}_n}_{n \times n} \right)}_{\mathbf{H}} \quad (49)$$

$$\begin{bmatrix} \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T \cdot \mathbf{H}^{-1} \end{bmatrix} = (\bar{\mathbf{A}} - \mathbf{A}) \cdot \underbrace{\mathbf{H} \mathbf{H}^{-1}}_{\mathbf{I}_{n \times n}} \quad (50)$$

$$\begin{bmatrix} \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T \cdot \mathbf{H}^{-1} \end{bmatrix} = \bar{\mathbf{A}} - \mathbf{A}. \quad (51)$$

(51) and some exercise in algebra gives (10) with (33) ■

### Proof of Projection Composition Operator in Eq. (11):

The optimal projection  $\bar{\mathbf{A}}^+$  is given when the gradient of (9) vanishes with

$$\nabla e(\bar{\mathbf{A}})^+ = 0 \quad (52)$$

$$\begin{bmatrix} \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T - \gamma \cdot \mathbf{P}_{\mathbf{q}} \mathbf{Q}^T \end{bmatrix} = (\bar{\mathbf{A}} - \mathbf{A}) \cdot \begin{bmatrix} \beta \cdot \mathbf{M} \mathbf{M}^T + \alpha \cdot \mathbf{I}_n \\ + [\gamma \cdot \bar{\mathbf{A}} \cdot \mathbf{Q} \mathbf{Q}^T] \end{bmatrix} \quad (53)$$

$$\begin{bmatrix} \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T - \gamma \cdot \mathbf{P}_{\mathbf{q}} \mathbf{Q}^T \end{bmatrix} = \bar{\mathbf{A}} \cdot \begin{bmatrix} \beta \cdot \mathbf{M} \mathbf{M}^T + \alpha \cdot \mathbf{I}_n \\ - \mathbf{A} [\beta \cdot \mathbf{M} \mathbf{M}^T + \alpha \cdot \mathbf{I}_n] \\ + \bar{\mathbf{A}} \cdot [\gamma \cdot \mathbf{Q} \mathbf{Q}^T] \end{bmatrix} \quad (54)$$

$$\begin{bmatrix} \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T - \gamma \cdot \mathbf{P}_{\mathbf{q}} \mathbf{Q}^T \end{bmatrix} = \bar{\mathbf{A}} \cdot \begin{bmatrix} \beta \cdot \mathbf{M} \mathbf{M}^T + \alpha \cdot \mathbf{I}_n \\ + \mathbf{A} [\beta \cdot \mathbf{M} \mathbf{M}^T + \alpha \cdot \mathbf{I}_n] \\ + \bar{\mathbf{A}} \cdot [\gamma \cdot \mathbf{Q} \mathbf{Q}^T] \end{bmatrix} \quad (55)$$

$$\begin{bmatrix} \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T - \gamma \cdot \mathbf{P}_{\mathbf{q}} \mathbf{Q}^T \end{bmatrix} = \bar{\mathbf{A}} \cdot \underbrace{\left[ \beta \cdot \mathbf{M} \mathbf{M}^T + \gamma \cdot \mathbf{Q} \mathbf{Q}^T + \alpha \cdot \mathbf{I}_n \right]}_{\mathbf{H}} + \mathbf{A} [\beta \cdot \mathbf{M} \mathbf{M}^T + \alpha \cdot \mathbf{I}_n] \quad (56)$$

(56) with  $\mathbf{H}^{-1}$  gives (11) with (33) ■

**Proof of Optimal Projection for case  $\alpha = \beta = 1, \gamma = 0$ , and  $\mathbf{M} = \mathbf{m}$  in Eq. (80):**

The optimal projection  $\bar{\mathbf{A}}$  is given if the gradient of (79) vanishes with

$$\nabla e(\bar{\mathbf{A}})_{(1,1,0)} = 0 \quad (57)$$

$$\Delta \mathbf{p} \mathbf{m}^T = (\bar{\mathbf{A}} - \mathbf{A}) \cdot (\mathbf{m} \mathbf{m}^T + \mathbf{I}_n) \quad (58)$$

$$(\Delta \mathbf{p} \mathbf{m}^T) \cdot (\mathbf{m} \mathbf{m}^T + \mathbf{I}_n)^{-1} = (\bar{\mathbf{A}} - \mathbf{A}) \cdot \underbrace{(\mathbf{m} \mathbf{m}^T + \mathbf{I}_n)(\mathbf{m} \mathbf{m}^T + \mathbf{I}_n)^{-1}}_{\mathbf{I}_n} \quad (59)$$

$$(\Delta \mathbf{p} \mathbf{m}^T) \cdot (\mathbf{m} \mathbf{m}^T + \mathbf{I}_n)^{-1} = \bar{\mathbf{A}} - \mathbf{A} \quad (60)$$

With (67) to (71) for the inverse  $(\mathbf{m} \mathbf{m}^T + \mathbf{I}_n)^{-1}$  follows for the left side in (60):

$$= (\Delta \mathbf{p} \cdot \mathbf{m}^T) \cdot \left( \left[ \frac{1}{1 + \|\mathbf{m}\|_2^2} \right] \cdot [\mathbf{I}_n + (\|\mathbf{m}\|_2^2 \cdot \mathbf{I}_n) - \mathbf{m} \mathbf{m}^T] \right) \quad (61)$$

$$= \left[ \frac{1}{1 + \|\mathbf{m}\|_2^2} \right] \cdot (\Delta \mathbf{p} \cdot \mathbf{m}^T) \cdot [\mathbf{I}_n + (\|\mathbf{m}\|_2^2 \cdot \mathbf{I}_n) - \mathbf{m} \mathbf{m}^T] \quad (62)$$

$$= \left( \frac{1}{1 + \|\mathbf{m}\|_2^2} \cdot \Delta \mathbf{p} \right) \cdot (\mathbf{m}^T \cdot [\mathbf{I}_n + (\|\mathbf{m}\|_2^2 \cdot \mathbf{I}_n) - \mathbf{m} \mathbf{m}^T]) \quad (63)$$

$$= \left( \frac{1}{1 + \|\mathbf{m}\|_2^2} \cdot \Delta \mathbf{p} \right) \cdot (\mathbf{m}^T \cdot \mathbf{I}_n + (\mathbf{m}^T \|\mathbf{m}\|_2^2 \cdot \mathbf{I}_n) - (\mathbf{m}^T \cdot \mathbf{m} \mathbf{m}^T)) \quad (64)$$

$$= \left( \frac{1}{1 + \|\mathbf{m}\|_2^2} \cdot \Delta \mathbf{p} \right) \cdot \left[ (\mathbf{m}^T \cdot \mathbf{I}_n) + \underbrace{(\mathbf{m}^T \|\mathbf{m}\|_2^2 \cdot \mathbf{I}_n)}_0 - \underbrace{(\mathbf{m}^T \cdot \mathbf{m}) \mathbf{m}^T}_{\|\mathbf{m}\|_2^2} \right] \quad (65)$$

$$= \left( \frac{1}{1 + \|\mathbf{m}\|_2^2} \cdot \Delta \mathbf{p} \right) \cdot (\mathbf{m}^T \cdot \mathbf{I}_n) \quad (66)$$

By exchanging the left side in (60) with (66), (80) follows directly ■

**Derivation of Identity for  $(\mathbf{m} \mathbf{m}^T + \mathbf{I}_n)^{-1}$ :**

$$(\mathbf{m} \mathbf{m}^T + \mathbf{I}_n)^{-1} \quad (67)$$

$$= \begin{pmatrix} \frac{1 + \|\mathbf{m}\|_2^2 - m_1^2}{1 + \|\mathbf{m}\|_2^2} & \frac{-m_1 \cdot m_2}{1 + \|\mathbf{m}\|_2^2} & \dots & \frac{-m_1 \cdot m_n}{1 + \|\mathbf{m}\|_2^2} \\ \frac{-m_2 \cdot m_1}{1 + \|\mathbf{m}\|_2^2} & \frac{1 + \|\mathbf{m}\|_2^2 - m_2^2}{1 + \|\mathbf{m}\|_2^2} & \dots & \frac{-m_2 \cdot m_n}{1 + \|\mathbf{m}\|_2^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{-m_n \cdot m_1}{1 + \|\mathbf{m}\|_2^2} & \frac{-m_n \cdot m_2}{1 + \|\mathbf{m}\|_2^2} & \dots & \frac{1 + \|\mathbf{m}\|_2^2 - m_n^2}{1 + \|\mathbf{m}\|_2^2} \end{pmatrix} \quad (68)$$

$$= \left[ \frac{1}{1 + \|\mathbf{m}\|_2^2} \right] \cdot \begin{pmatrix} \frac{1 + \|\mathbf{m}\|_2^2 - m_1^2}{1 + \|\mathbf{m}\|_2^2} & \frac{-m_1 \cdot m_2}{1 + \|\mathbf{m}\|_2^2} & \dots & \frac{-m_1 \cdot m_n}{1 + \|\mathbf{m}\|_2^2} \\ \frac{-m_2 \cdot m_1}{1 + \|\mathbf{m}\|_2^2} & \frac{1 + \|\mathbf{m}\|_2^2 - m_2^2}{1 + \|\mathbf{m}\|_2^2} & \dots & \frac{-m_2 \cdot m_n}{1 + \|\mathbf{m}\|_2^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{-m_n \cdot m_1}{1 + \|\mathbf{m}\|_2^2} & \frac{-m_n \cdot m_2}{1 + \|\mathbf{m}\|_2^2} & \dots & \frac{1 + \|\mathbf{m}\|_2^2 - m_n^2}{1 + \|\mathbf{m}\|_2^2} \end{pmatrix} \quad (69)$$

$$= \left[ \frac{1}{1 + \|\mathbf{m}\|_2^2} \right] \cdot \left[ \underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}}_{\mathbf{I}_n} + \underbrace{\begin{pmatrix} \|\mathbf{m}\|_2^2 & 0 & \dots & 0 \\ 0 & \|\mathbf{m}\|_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \|\mathbf{m}\|_2^2 \end{pmatrix}}_{\|\mathbf{m}\|_2^2 \cdot \mathbf{I}_n} - \underbrace{\begin{pmatrix} m_1^2 & m_1 \cdot m_2 & \dots & m_1 \cdot m_n \\ m_2 \cdot m_1 & m_2^2 & \dots & m_2 \cdot m_n \\ \vdots & \vdots & \ddots & \vdots \\ m_n \cdot m_1 & m_n \cdot m_2 & \dots & m_n^2 \end{pmatrix}}_{\mathbf{m} \mathbf{m}^T} \right] \quad (70)$$

$$= \left[ \frac{1}{1 + \|\mathbf{m}\|_2^2} \right] \cdot [\mathbf{I}_n + (\|\mathbf{m}\|_2^2 \cdot \mathbf{I}_n) - \mathbf{m} \mathbf{m}^T] \quad (71)$$

**Proof of Eq. (82):**

Putting  $\bar{\mathbf{A}}$  of (80) in  $e_{\mathbf{M}}$  of (4) gives:

$$\left\| \left[ \left( \frac{1}{1 + \|\mathbf{m}\|_2^2} \right) \Delta \mathbf{p} \cdot \mathbf{m}^T \right] \cdot \mathbf{m} - \Delta \mathbf{p} \right\|_2^2 \quad (72)$$

$$= \left\| \left[ \left( \frac{1}{1 + \|\mathbf{m}\|_2^2} \right) \Delta \mathbf{p} \cdot \underbrace{(\mathbf{m}^T \mathbf{m})}_{\|\mathbf{m}\|_2^2} \right] - \Delta \mathbf{p} \right\|_2^2 \quad (73)$$

$$= \left\| \left( \frac{\|\mathbf{m}\|_2^2}{1 + \|\mathbf{m}\|_2^2} \right) \Delta \mathbf{p} - \Delta \mathbf{p} \right\|_2^2 = \left\| \left( \frac{\|\mathbf{m}\|_2^2}{1 + \|\mathbf{m}\|_2^2} - 1 \right) \Delta \mathbf{p} \right\|_2^2 \quad (74)$$

$$= \left\| \left( 1 - \frac{1}{1 + \|\mathbf{m}\|_2^2} - 1 \right) \Delta \mathbf{p} \right\|_2^2 = \left\| \left( -\frac{1}{1 + \|\mathbf{m}\|_2^2} \right) \Delta \mathbf{p} \right\|_2^2 \quad (75)$$

$$= \left( -\frac{1}{1 + \|\mathbf{m}\|_2^2} \right)^2 \cdot (\Delta p_1^2 + \Delta p_2^2) \quad (76)$$

Considering  $(\Delta p_1^2 + \Delta p_2^2)$  as  $\|\Delta \mathbf{p}\|_2^2$  in (76) and dissolving the binomial gives (82) ■

**Proof of Eq. (22):**

First, we see using expression (18) for  $\mathbf{H}$  and (10) for  $\mathbf{A} - \bar{\mathbf{A}}$  that

$$e(\bar{\mathbf{A}})_{\mathbf{M}} = \sum_{i=1}^q \|(\mathbf{A} - \bar{\mathbf{A}}) \mathbf{m}_i - \Delta \mathbf{p}\|_2^2 \quad (77)$$

$$= \left\| \beta \Delta \mathbf{p} \mathbf{1}^T \mathbf{M}^T \mathbf{H}^{-1} \mathbf{M} - \Delta \mathbf{p} \mathbf{1}^T \right\|_F^2$$

$$\leq q \beta^2 \|\Delta \mathbf{p}\|_2^2 \left\| \mathbf{M}^T \mathbf{H}^{-1} \mathbf{M} - \frac{1}{\beta} \mathbf{I}_q \right\|_F^2.$$

Using the identity  $\|\mathbf{M}\|_F^2 = \text{tr}(\mathbf{M}^T \mathbf{M})$  for the Frobenius norm and the fact that  $\mathbf{H}$  is symmetric positive definite (for  $\alpha > 0$ ) we get

$$\left\| \mathbf{M}^T \mathbf{H}^{-1} \mathbf{M} - \frac{1}{\beta} \mathbf{I}_q \right\|_F^2 = \left\| \mathbf{M}^T \mathbf{H}^{-1} \mathbf{M} \right\|_F^2 - \frac{2}{\beta} \left\| \mathbf{H}^{-1/2} \mathbf{M} \right\|_F^2 + \frac{q}{\beta^2}$$

$$\leq \left\| \mathbf{H}^{-1} \right\|_F^2 \|\mathbf{M}\|_F^4 - \frac{2}{\beta} \frac{\sigma_{\min}^2(\mathbf{M})}{\sigma_{\max}^2(\mathbf{H}^{1/2})} + \frac{q}{\beta^2}$$

$$\leq q \frac{\sigma_{\max}^4(\mathbf{M})}{(\alpha + \beta \sigma_{\min}^2(\mathbf{M}) + \gamma \sigma_{\min}^2(\mathbf{Q}))^2}$$

$$- \frac{2}{\beta} \frac{\sigma_{\min}^2(\mathbf{M})}{\alpha + \beta \sigma_{\max}^2(\mathbf{M}) + \gamma \sigma_{\max}^2(\mathbf{Q})} + \frac{q}{\beta^2} \quad (78)$$

Using (78) together with (77), the estimates  $\|\mathbf{M}\|_F^2 \leq q \sigma_{\max}^2(\mathbf{M})$  and  $\|\mathbf{M}\|_F^2 \geq \sigma_{\min}^2(\mathbf{M})$  as well as  $\sigma_{\min}(\mathbf{M}) + \delta_{\mathbf{M}} = \sigma_{\max}(\mathbf{M})$  we arrive at (22).

### Considerations to the existence of the inverse of $\mathbf{H}$

The existence of a solution in (10) and (11) depends on the existence of  $\mathbf{H}^{-1}$ . Thus, we going to discuss this aspect. If  $\alpha > 0$ , the term  $\alpha \cdot \mathbf{I}_n$  acts as a regularization term for  $\mathbf{M}\mathbf{M}^T$  and  $\mathbf{Q}\mathbf{Q}^T$  making  $\mathbf{H}$  a positive definite and thus invertible matrix, independent of the choice for  $\beta$  and  $\gamma$  and even if  $\mathbf{M}$  and  $\mathbf{Q}$  are empty. If  $\alpha$  is zero, the existence of the inverse depends on several factors: If  $\beta \neq 0, \gamma = 0, q > n$  and the data points  $\mathbf{m}_i \in \mathbf{M}$  are linearly independent,  $\mathbf{M}$  becomes positive definite. If  $\beta = 0, \gamma \neq 0, k > n$ , and the data points  $\mathbf{q}_j \in \mathbf{Q}$  are linearly independent,  $\mathbf{Q}$  becomes positive definite. If  $\beta > 0, \gamma > 0$  and either  $\mathbf{M}$  or  $\mathbf{Q}$  (or both) are positive definite, the existence of  $\mathbf{H}^{-1}$  can be guaranteed. Note that the data points become linearly independent (if they are not) by adding some tiny noise to them. Figure 11 illustrates the related decision tree for the existence of the inverse of  $\mathbf{H}$ .

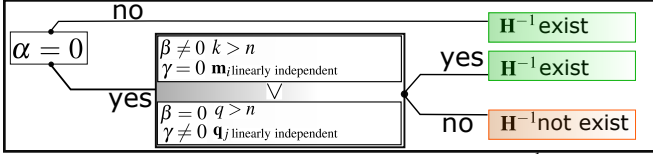


Fig. 11: Decision tree for the existence of  $\mathbf{H}^{-1}$ .

Besides the algebraic properties of parameter  $\alpha$  – making the inverse in  $\mathbf{H}$  existent or not – the parameter  $\alpha$  has an additional deeper semantic meaning: It fades the global change impact term in, in order to get ready to operate. Since we are generally interested in an as-small-as-possible change of projections under using our Composition Operators we chose  $\alpha = 1$ , which makes the inverse existent, and enables a minimization of global impact under interaction.

### Composition Operator for One Shift-able Point

Manipulation Energy Gradient  $\nabla e(\bar{\mathbf{A}})$   
\*\*\* one point shift \*\*\*

$$= 2(\bar{\mathbf{A}} - \mathbf{A}) \cdot (\mathbf{m}\mathbf{m}^T + \mathbf{I}_n) - 2\Delta\mathbf{p} \cdot \mathbf{m}^T \quad (79)$$

A closed form solution for the projection matrix  $\bar{\mathbf{A}}$  that minimizes energy  $e$  when shifting one point  $\mathbf{A}\mathbf{m}$  to its desired new position  $(\mathbf{A}\mathbf{m}) + \Delta\mathbf{p}$  is given as:

Optimal Projection Composition Operator  
\*\*\* one point shift \*\*\*

$$\bar{\mathbf{A}} = \left( \frac{1}{1 + \|\mathbf{m}\|_2^2} \right) \Delta\mathbf{p} \cdot \mathbf{m}^T + \mathbf{A} \quad (80)$$

(80) does not trivially follow from gradient (79). Thus, please find a proof of (80) in this additional material. This solution grows linearly in  $n$ , can be computed quickly, and thus allows an interactive scheme.

### Evaluation of the Quality Terms v.s. the Data Norms for One Point Shift

In order to evaluate the influence of the shift operations to the quality terms (cf. Section 4), we consider a certain configuration: Usually, data are normed into a  $[0, 1]$  interval, e.g., when they are loaded. Following the central limit theorem, the mean component value is then likely  $\bar{d} = 0.5$  ( $(\max - \min)/2$ ) when the number  $m$  of records is large. Moreover, usually, it cannot be expected that a shift  $\Delta\mathbf{p}$  is more distant than 1, describing a unit circle in the projection space. Hence, we assume the following *standard configuration* as lower and upper estimations for a component  $d_i$  of the data records  $d_i \in \mathbf{d}$  and the norm of any shift  $\Delta\mathbf{p}$ :

$$[d_{\min} = 0, \bar{d} = 0.5, d_{\max} = 1, \|\Delta\mathbf{p}\|_{2, \max} = 1]. \quad (81)$$

Note that our standard configuration represents the space of configurations, meaning that the discussion for other min-max configurations go analogously.

The general shift quality of (4) gives, with the operator of (80), the shift quality for the one point case as

$$e(\bar{\mathbf{A}})_{\mathbf{M}} = \frac{\|\Delta\mathbf{p}\|_2^2}{\|\mathbf{m}\|_2^4 + 2\|\mathbf{m}\|_2^2 + 1}, \quad (82)$$

describing the quality of how good the point  $\mathbf{p}_m$  is placed to point  $\bar{\mathbf{p}}_m$  by applying the Composition Operator of (80). Find the proof of (82) in the additional material, too. Ideally, it vanishes under shift operations.

First, this energy usually seems not to be zero, but growing with the shift vector  $\Delta\mathbf{p}$ , reflecting the required regularization in  $\mathbf{H}$  discussed above, since the term  $\mathbf{m}\mathbf{m}^T$  is singular and thus  $(\mathbf{m}\mathbf{m}^T)^{-1}$  does not exist (cf. Figure 11). Second, from Sec. 5, we know that a data record with vanishing norm requires a larger change  $\Delta\mathbf{A}$  in  $\mathbf{A}$  than a record with larger norm to cross  $\Delta\mathbf{p}$ . Thus, the norm  $\|\mathbf{m}\|_2$  of the record is also important for the error, which shift quality (82) reflects, too.

In order to analyzing this aspect, let  $\mathbf{m}_s = (m_s, \dots, m_s)^T$  be a *shift-able* data point, with constant component  $s$ , giving its norm as  $\|\mathbf{m}_s\|_2 = \sqrt{n} \cdot s$ , where  $s$  defines the radius of a hypersphere. Since each point on a hypersphere has the same norm,  $\mathbf{m}_s$  allows us to generally investigate norm properties. Then, the behavior of the shift quality for the standard configuration for a data dimensionality  $n$  from 3 to 10 is given in Figure 12. It can be seen that the quality error shrinks to 0 if the norm of the shift point in data space gets large and grows to infinity. This is an expected behavior, from Section 5. The opposite statement applies as well, meaning that small values increase the error. This is plausible, since the smallest normed value 0 cannot be shifted at all. Hence, the shift quality blends from  $\|\Delta\mathbf{p}\|_{2, \max} = 1$  to zero, considered over the data norm spectrum from zero to infinity. Moreover, the mean shift quality for the mean norm  $\|\mathbf{m}_{0.5}\|_2$  is 0.32 to 0.08 for dims 3 to 10, when the user wants to cross the length 1 by a shift. This shift error is small, even negligible; and in practice the shift is usually smaller than one, decreasing this negligible quality error further. In conclusion, we learn that the quality improves when  $n$  grows (or for small  $n$ , when the data norm  $\|\mathbf{m}\|_2$  grows) and that the mean shift quality is small and negligible.

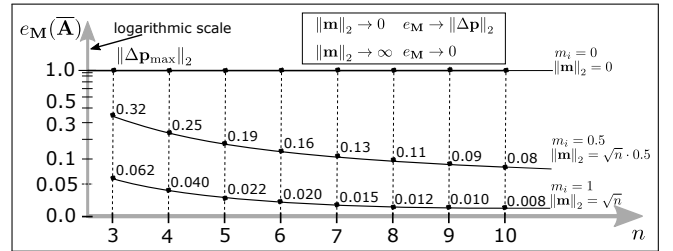


Fig. 12: Shift quality  $e_{\mathbf{M}}$  for standard configuration vs. dimensionality  $n$ .

### Axes Configuration: Degeneration vs. Stability

One aspect that needs to be considered is the degeneration of the axes configuration under interaction. Fig. 13 illustrates this effect: It means that the difference between the norm of the smallest and largest anchor point may get large and even grow to infinity (reflecting the degeneration of the two singular values in  $\mathbf{A}$ ). Though these projections exist, they are not appropriate for a visual inspection anymore, because of the following required excessive use of zooming and panning operations. Figure 14 shows the Iris axes during a closed shift tour, without (top) and with (bottom) sticky points. No axis degeneration occurs when points are moved away and back - confirmed in all our experiments - because the manipulation energy balances all axes during Composition. Thus, axis stability behaves as desired.

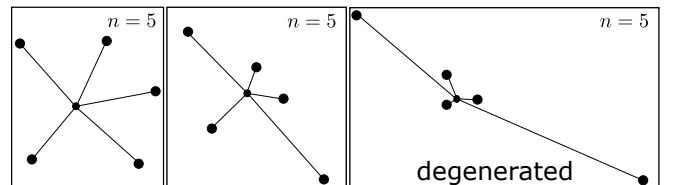


Fig. 13: Illustration for a degeneration of projection matrix  $\mathbf{A}$  for  $n = 5$  dimensions: (left to right) evaluation from the well-defined to the degenerated case.

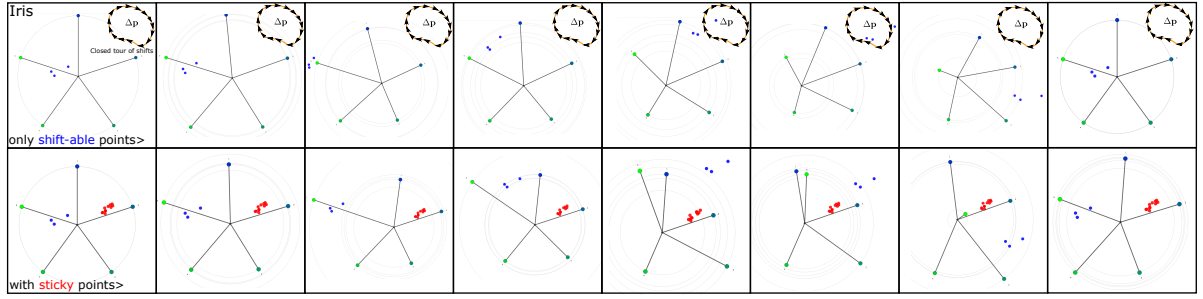


Fig. 14: Axes Configuration: Degeneration vs. Stability.

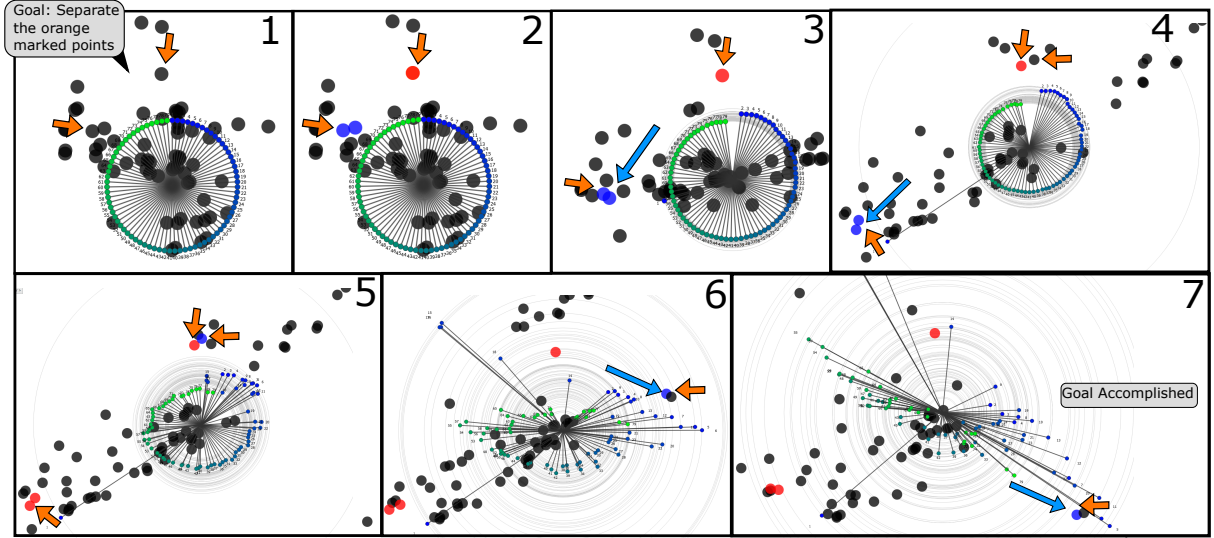


Fig. 15: Case Study - Maximum Separation for Points of Interest in the US Money Data Set.

### Case Study - Maximum Separation for Points of Interest in the US Money Data Set

In Fig. 15, a case study for a visual task is shown that can be solved with the aid of our Composition Operators. In detail, a user may be interested in maximizing the separation of a set of points of interest (orange arrows). For doing so, using a simple strategy of shifting points and selecting sticky points appropriately does the job. See the axes configuration outcome in sub-figure 7. It would be challenging (and likely time-consuming) to get such a configuration, which fulfills the desired visual task, by a manual exploration of the parameter space or by using projection tours.

### Case Study - Cluster Emphasis in the Iris Dataset

Fig. 16, illustrates our cluster emphasis workflow. In Fig. 16a we can see our initial Star Coordinate projection. Next in Fig. 16b, we anchor the Setosa class (pink) as sticky and select one representative Versicolor point (brown) and one Virginica point (yellow) as shiftable. To reduce overlap, we first drag the Virginica point leftward, increasing its distance from the Versicolor cluster (Fig. 16c). Then in Fig. 16d, we move the Versicolor point towards the top-right, pushing it further away from Virginica. By applying further sequential shifts along different axes (Fig. 16e+f), the projection achieves greater class separation and markedly reduces overlap. Although Versicolor and Virginica remain inherently intermingled — complete separation is not possible given their feature similarity — the adjusted layout brings the overlapping region into clearer focus, making it easier to inspect and interpret borderline cases as we can see in Fig. 16g.

### Case Study - Feature-Selective Separation

The user first anchors one or more exemplar points for a target cluster, then applies targeted shifts to pull other points with similar characteristics toward those exemplars. This operation consolidates the cluster

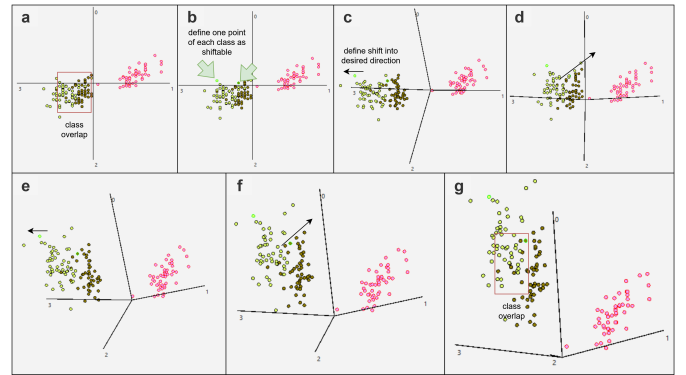


Fig. 16: Case Study - Cluster Emphasis in the Iris Dataset

into a tighter, more coherent grouping, enhancing intra-cluster cohesion and overall class separability. Figure 17 demonstrates this on the WDBC dataset. By pulling benign instances toward a chosen exemplar, the benign cluster becomes markedly more compact and more clearly separated from malignant samples.

### Case Study - Shape-Driven Exploration

Figure 18 illustrates shape-driven exploration on a small, synthetically generated five-dimensional dataset. In the left panel, all axes in the Star Coordinate View are equally weighted and the red points appear as an unstructured cloud with no discernible pattern. In the center panel, a handful of points have been marked as shiftable and dragged onto a circular guide; using the Composition Operator then computes a new projection matrix in closed form so that those points lie exactly

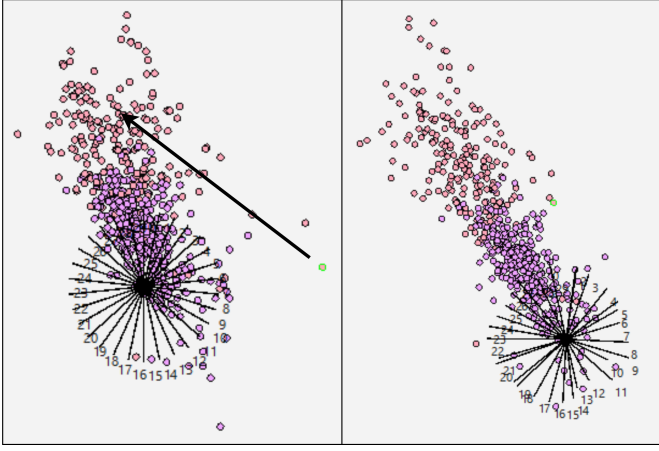


Fig. 17: Feature Selective Separation creating more compact clusters for WDBC dataset

on the circle, yielding a clean, ring-shaped arrangement of the entire dataset. In the right panel, a different subset of points is aligned along a straight-line band and similarly shifted; the operator recomputes the projection matrix to project them precisely onto that line, producing a narrow filament of points. These examples demonstrate that, by simply specifying a geometric form and shifting a few points, one can sculpt arbitrary shapes out of high-dimensional data in a single, deterministic solve without any iterative parameter tuning or exhaustive search if this structure lies in the given data.

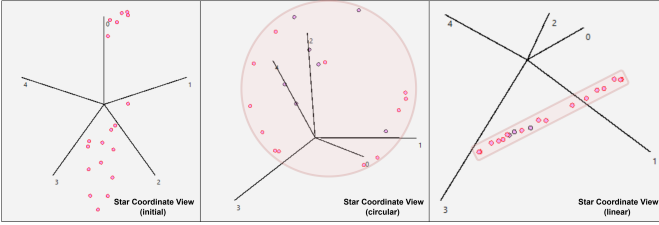


Fig. 18: Shape-Driven Exploration of a synthetically generated sample dataset

### Case Study - Automatically Class Separating Projections

A common visual task is to select projections that exhibit class separation [52, 67]. Here, class separation means that classes appear as distinct clusters in the projection space if and only if they are also separated in the original  $nD$  dataset (i.e., one that is free of mutual overlap whenever possible). Given the infinite possibilities offered by the projection space — and considering that this problem is NP-hard — we demonstrate the potential of Composition Operators by employing them to automatically identify class separated projections.

Fig. 19 presents a class separated projection of the Statlog data [18, 52], achieved using the greedy scheme described below as the **Algorithm for Greedy Class Separation with Composition Operators**. As the scheme progresses from the initial state on the left to convergence on the right, the separation between classes improves markedly, demonstrating the effectiveness of composition operators. Note that the DSC values are normalized by the number of data points.

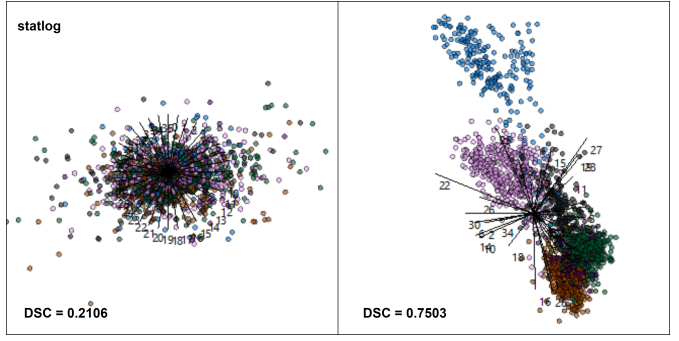


Fig. 19: Greedy separation of Statlog data with Composition Operators

#### Algorithm: Greedy Class Separation with Composition Operators

1) Our scheme quantifies the current quality of class separation in the projection  $\mathbf{P}$  using the Distance Consistency (DSC) metric from Sips et al. [67]. The DSC is defined as  $DSC(\mathbf{P}) = |\{p_{k,i} : \|p_{l,i} - c_k\| \leq \|p_{k,i} - c_l\| \ \forall l \in |C|\}|$ , where  $p_{b,c} \in \mathbf{p}$  denotes the  $b^{th}$  2D projection point (with  $b \in [1, t]$ ) of class  $c$  (with  $t = |C|$ ) with respect to the projection  $\mathbf{P}$ . Larger DSC values indicate better class separation.

2) The 2D centroid  $\mathbf{c}_c$  of class  $c$  in the projection  $\mathbf{p}$  is defined as the mean of all 2D points that belong to that class.

3) Within our greedy scheme, we alternate by allowing the points of a single class  $c$  to be freely shifted using a specific shift vector  $\Delta \mathbf{p}$ , while keeping the points of all other classes fixed. This yields an updated projection matrix  $\mathbf{A}_{k+1}$  derived from the current matrix  $\mathbf{A}_k$  and the shift  $\Delta \mathbf{p}$  according to Eq. 16, with the goal of progressively maximizing class separation.

In other words, one class is designated as *shift-able* while the others remain *sticky*. This alternating one-vs.-rest strategy is repeated until convergence — that is, until the DSC ceases to improve (evaluated over 50 iterations to mitigate numerical noise in the convergence behavior). For the currently movable class  $c$ , the shift vector  $\Delta \mathbf{p}$  is defined as:

$$\Delta \mathbf{p} = \frac{\|\mathbf{c}\|_2^2}{(d_{\min} + \|\mathbf{c}\|_2^2)} \cdot \mathbf{c}_{ci} + f(k) \cdot \frac{d_{\min}}{(d_{\min} + \|\mathbf{c}\|_2^2)} \cdot \mathbf{c} \quad (83)$$

with  $\mathbf{c}$  is the average of the points in the projection  $\mathbf{p}$ ,  $d_{\min} = \min(\|\mathbf{c}_{ci}\|_2^2 = \|\mathbf{c}_c - \mathbf{c}_i\|_2^2)$  with  $i$  represents a loop over all classes, except for class  $c$ ,  $k$  is the iteration number of the greedy approach, and  $f(k)$  is  $f(k) = 100/(100 + k)$ , supposed to fade out the impact of the overall average  $\mathbf{c}$  in the later iterations. The shift vector  $\Delta \mathbf{p}$  is designed to penalize classes that exhibit substantial overlap according to the DSC metric, thereby driving them apart. Our approach initializes the greedy scheme with the projection matrix  $\mathbf{A}_0$  corresponding to the orthographic standard configuration [43].

### Comparison between Composition Operators and Molchanov [52]

Molchanov et al. [52] proposed a comparable method for class separation in linear projections. Their approach shifts an entire class by representing its centroid, while all points in the projection space remain shiftable. The actual shift is performed through manual user input (drag & drop) of the class centroid in the desired direction. However, automatic optimization with Composition Operators exceeds this approach in several respects:

1. The use of sticky points preserves the existing structure, enabling increased compactness within classes and greater separation between them.
2. Degenerated axes are prevented through orthographic representation.

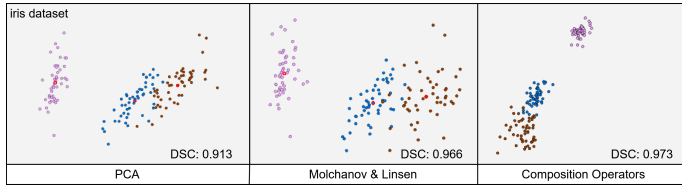


Fig. 20: Comparison between PCA initialized star coordinate widget (left), the technique by Molchanov et al. [52] (middle) and automatically computed class separating projections with Composition Operators (right) for iris dataset. The level of separation is indicated by the DSC value in the figure.

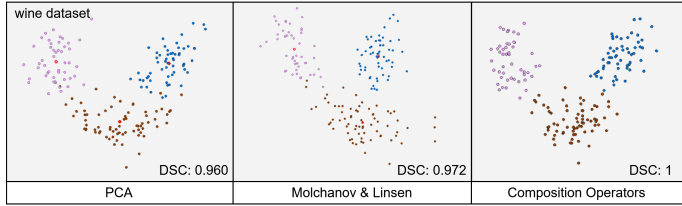


Fig. 21: Comparison between PCA initialized star coordinate widget (left), the technique by Molchanov et al. [52] (middle) and automatically computed class separating projections with Composition Operators (right) for wine dataset. The level of separation is indicated by the DSC value in the figure.

tion.

3. Automatic optimization allows reproducibility of the manipulation process, resulting in consistent outcomes across multiple runs.
4. Both single- and multi-point shifts offer enhanced flexibility in manipulation tasks.

Figure 20 and Figure 21 show the results of class separation on various benchmark datasets using automatic optimization with Composition Operators compared to the manual approach by Molchanov et al. [52]. To ensure a fair comparison, we selected the best result out of 5 attempts from Molchanov et al.’s approach, as this method requires manual intervention and thus exhibits variability in separation quality. We used well-known benchmark datasets — iris, wine, statlog, and ecoli — from the UCI repository. Using DSC as the reference metric (cf. Sips et al.), both approaches perform differently compared to the initial configuration using PCA (see Table 2), where Composition Operators outperform. Furthermore, with extensive user interactions, the approach of Molchanov et al. may lead to degenerated axes, whereas optimization with Composition Operators uses orthogonally scaled axes to prevent such degeneration.

|         | DSC <sub>PCA</sub> | DSC <sub>Mol</sub> | DSC <sub>CO</sub> |
|---------|--------------------|--------------------|-------------------|
| iris    | 0.913              | 0.966              | 0.973             |
| wine    | 0.960              | 0.972              | 1.000             |
| statlog | 0.752              | 0.793              | 0.818             |
| ecoli   | 0.678              | 0.749              | 0.761             |

Table 2: Benchmark for class separation: DSC<sub>PCA</sub> is the DSC after embedding the data using PCA (which also corresponds to Star Coordinates), DSC<sub>Mol</sub> is the best DSC value after 5 attempts at class separation using the approach by Molchanov et al., and DSC<sub>CO</sub> is the DSC value after automatic class separation with Composition Operators.

## Operators at a Glance

A compact overview of the introduced Composition Operators can be found here.

### General Form of Composition Operators (no memory)

$$\bar{\mathbf{A}}^- = \left[ \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T \cdot \mathbf{H}^{-1} \right] + \mathbf{A}$$

with

$$\mathbf{H} = (\beta \cdot \mathbf{M} \mathbf{M}^T + \gamma \cdot \mathbf{Q} \mathbf{Q}^T + \alpha \cdot \mathbf{I}_n).$$

### General Form of Composition Operators (memory)

$$\bar{\mathbf{A}}^+ = \left[ \beta \cdot \Delta \mathbf{p} \cdot (\mathbf{M} \cdot \mathbf{1}_q)^T + (\gamma \cdot \mathbf{P}_q \mathbf{Q}^T) + \mathbf{A}(\alpha \cdot \mathbf{I}_n + \beta \cdot \mathbf{M} \mathbf{M}^T) \right] \cdot \mathbf{H}^{-1}$$

with

$$\mathbf{H} = (\beta \cdot \mathbf{M} \mathbf{M}^T + \gamma \cdot \mathbf{Q} \mathbf{Q}^T + \alpha \cdot \mathbf{I}_n).$$

### Special Case of Composition Operators:

Only Set-Shift, No Sticky ( $\alpha = 1, q > 1, k = 0$ )

$$\bar{\mathbf{A}}_\epsilon = \left[ \Delta \mathbf{p} \cdot (\mathbf{M}_\epsilon \cdot \mathbf{1}_q)^T \cdot \mathbf{H}_\epsilon^{-1} \right] + \mathbf{A}$$

with

$$\mathbf{H}_\epsilon = (\mathbf{M}_\epsilon \mathbf{M}_\epsilon^T + \mathbf{I}_n).$$

with  $\mathbf{m}_\epsilon \in \mathbf{M}_\epsilon$  as

$$\mathbf{m}_\epsilon = (\mathbf{m} / \|\mathbf{m}\|_2) \cdot f_0(\epsilon),$$

and

$$f_0(\epsilon) = \|\mathbf{m}_{\min}\|_2 + (-3\|\mathbf{m}_{\min}\|_2 - \|\mathbf{m}\|_2 + 4\|\mathbf{m}_{\max}\|_2)\epsilon + (2\|\mathbf{m}_{\min}\|_2 + 2\|\mathbf{m}\|_2 - 4\|\mathbf{m}_{\max}\|_2)\epsilon^2.$$

Use when only a set of shift-able points are selected.

### Special Case of Composition Operators:

Single-Point Shift, No Sticky ( $\alpha = 1, q = 1, k = 0$ )

$$\bar{\mathbf{A}} = \left( \frac{1}{1 + \|\mathbf{m}\|_2^2} \right) \Delta \mathbf{p} \cdot \mathbf{m}^T + \mathbf{A}$$

Drag exactly one shift-able point.

### Special Case of Composition Operators:

Set-Shift, Set-Sticky, no memory ( $\alpha = 1, q > 0, k > 0$ )

$$\bar{\mathbf{A}}_{\epsilon\delta}^- = \left[ f_1^\delta \cdot \Delta \mathbf{p} \cdot (\mathbf{M}_\epsilon \cdot \mathbf{1}_q)^T \cdot \mathbf{H}_{\epsilon\delta}^{-1} \right] + \mathbf{A}$$

with

$$\mathbf{H}_{\epsilon\delta} = (f_1^\delta \cdot \mathbf{M}_\epsilon \mathbf{M}_\epsilon^T + f_2^\delta \cdot \mathbf{Q} \mathbf{Q}^T + \mathbf{I}_n)$$

with  $\mathbf{m}_\epsilon \in \mathbf{M}_\epsilon$  as

$$\mathbf{m}_\epsilon = (\mathbf{m} / \|\mathbf{m}\|_2) \cdot f_0(\epsilon),$$

and

$$f_0(\epsilon) = \|\mathbf{m}_{\min}\|_2 + (-3\|\mathbf{m}_{\min}\|_2 - \|\mathbf{m}\|_2 + 4\|\mathbf{m}_{\max}\|_2)\epsilon + (2\|\mathbf{m}_{\min}\|_2 + 2\|\mathbf{m}\|_2 - 4\|\mathbf{m}_{\max}\|_2)\epsilon^2,$$

and

$$f_1^\delta = -2\delta^2 + 3\delta, f_2^\delta = 4\delta^2 - 4\delta + 1.$$

Allows sticky points to drift slightly.

### Special Case of Composition Operators:

Set-Shift, Set-Sticky, memory ( $\alpha = 1, q > 0, k > 0$ )

$$\bar{\mathbf{A}}_{\epsilon\delta}^+ = \left[ f_1^\delta \cdot \Delta \mathbf{p} (\mathbf{M}_\epsilon \cdot \mathbf{1}_q)^T + (f_2^\delta \cdot \mathbf{P}_q \mathbf{Q}^T) + \mathbf{A}(\mathbf{I}_n + f_1^\delta \cdot \mathbf{M}_\epsilon \mathbf{M}_\epsilon^T) \right] \cdot \mathbf{H}_{\epsilon\delta}^{-1}$$

with

$$\mathbf{H}_{\epsilon\delta} = (f_1^\delta \cdot \mathbf{M}_\epsilon \mathbf{M}_\epsilon^T + f_2^\delta \cdot \mathbf{Q} \mathbf{Q}^T + \mathbf{I}_n)$$

with  $\mathbf{m}_\epsilon \in \mathbf{M}_\epsilon$  as

$$\mathbf{m}_\epsilon = (\mathbf{m} / \|\mathbf{m}\|_2) \cdot f_0(\epsilon),$$

and

$$f_0(\epsilon) = \|\mathbf{m}_{\min}\|_2 + (-3\|\mathbf{m}_{\min}\|_2 - \|\mathbf{m}\|_2 + 4\|\mathbf{m}_{\max}\|_2)\epsilon + (2\|\mathbf{m}_{\min}\|_2 + 2\|\mathbf{m}\|_2 - 4\|\mathbf{m}_{\max}\|_2)\epsilon^2,$$

and

$$f_1^\delta = -2\delta^2 + 3\delta, f_2^\delta = 4\delta^2 - 4\delta + 1.$$

Sticky points are pulled back to their *original* locations (memory).

All inverses are well-defined because  $\mathbf{I}_n$  acts as fixed Tikhonov term ( $\alpha \equiv 1$ ). Only one  $n \times n$  inverse per selection required.

### Practical Recommendations

1. **Select high-norm records as shift-able.**  
Requires smaller changes in  $\mathbf{A}$ ; preserves global context.
2. **Combine high-norm shift-able with low-norm sticky points.**  
Minimises manipulation energy and produces crisp moves.

### Further Tools

- **Axis locking.** Zero the corresponding rows in  $\mathbf{M}$  and  $\mathbf{Q}$  and subtract the linear influence of fixed axes from the projected points; the operator keeps those axes fixed in the application of the Composition Operators.