ParaDime

A FRAMEWORK FOR PARAMETRIC DIMENSIONALITY REDUCTION

Andreas Hinterreiter, Christina Humer, Bernhard Kainz, Marc Streit

EuroVis 2023

JOHANNES KEPLER UNIVERSITÄT LINZ

Imperial College London

TEAM

JKU

Andreas
Hinterreiter

Christina
Humer

Bernhard
Kainz

Marc
Streit

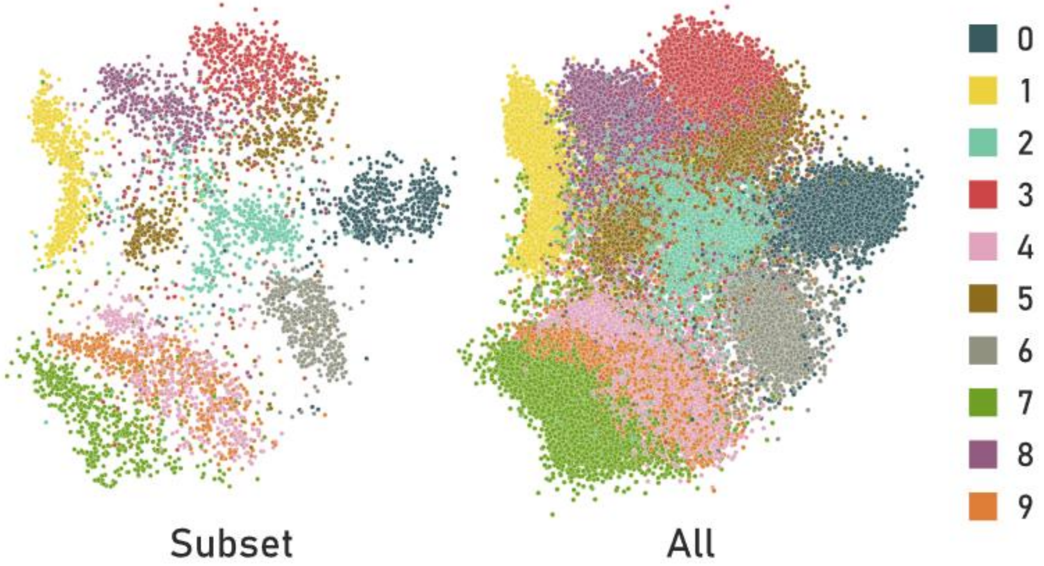
JKU
JOHANNES KEPLER
UNIVERSITY LINZ

JKU
Visual Data
Science Lab

Imperial College
London

FAU
Friedrich-Alexander-Universität
Erlangen-Nürnberg

paraDime




```yaml
relations:
  - name: dists hd
    level: global
    type: pairwise
    options:
      metric: euclidean
  - name: dists ld
    level: batch
    type: pairwise
    options:
      metric: euclidean
```

```yaml
losses:
  - name: mds
    type: relation
    func: mse
    keys:
      rels:
        - dists hd
        - dists ld
training phases:
  - loss:
      components: mds
```

MOTIVATION & GOALS

Unify

Parameterize

Share & Reuse

Customize

MOTIVATION & GOALS

Unify

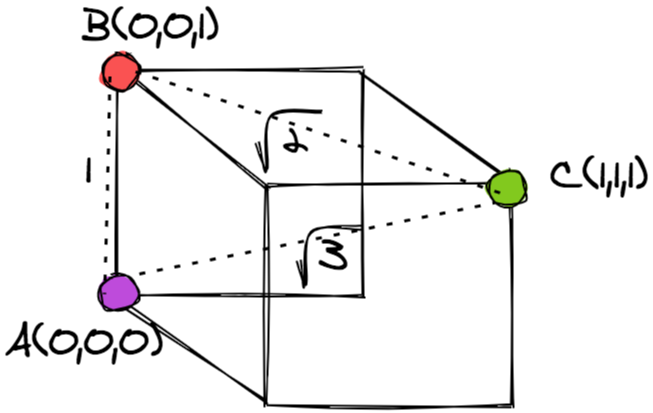
Parameterize

Share & Reuse

Customize

SIMILARITIES BETWEEN METHODS

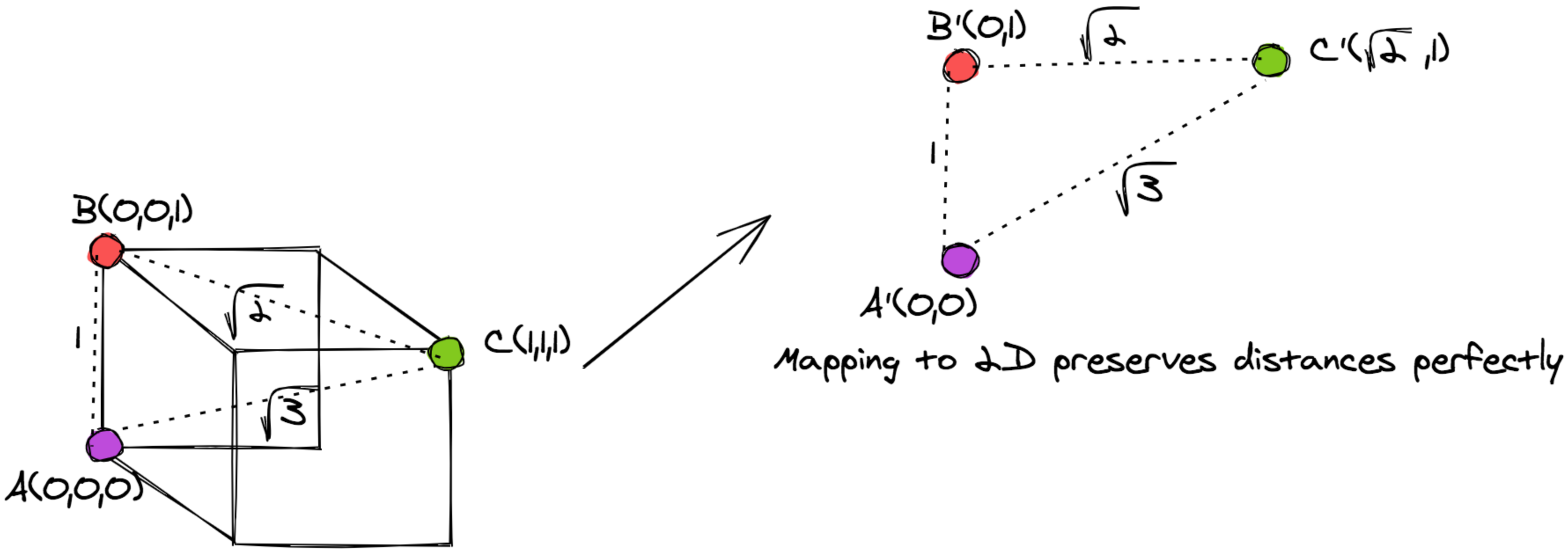
- MDS
- Isomap
- t-SNE
- LargeVis
- UMAP



https://stackabuse.com/guide-to-multidimensional-scaling-in-python-with-scikit-learn/

SIMILARITIES BETWEEN METHODS

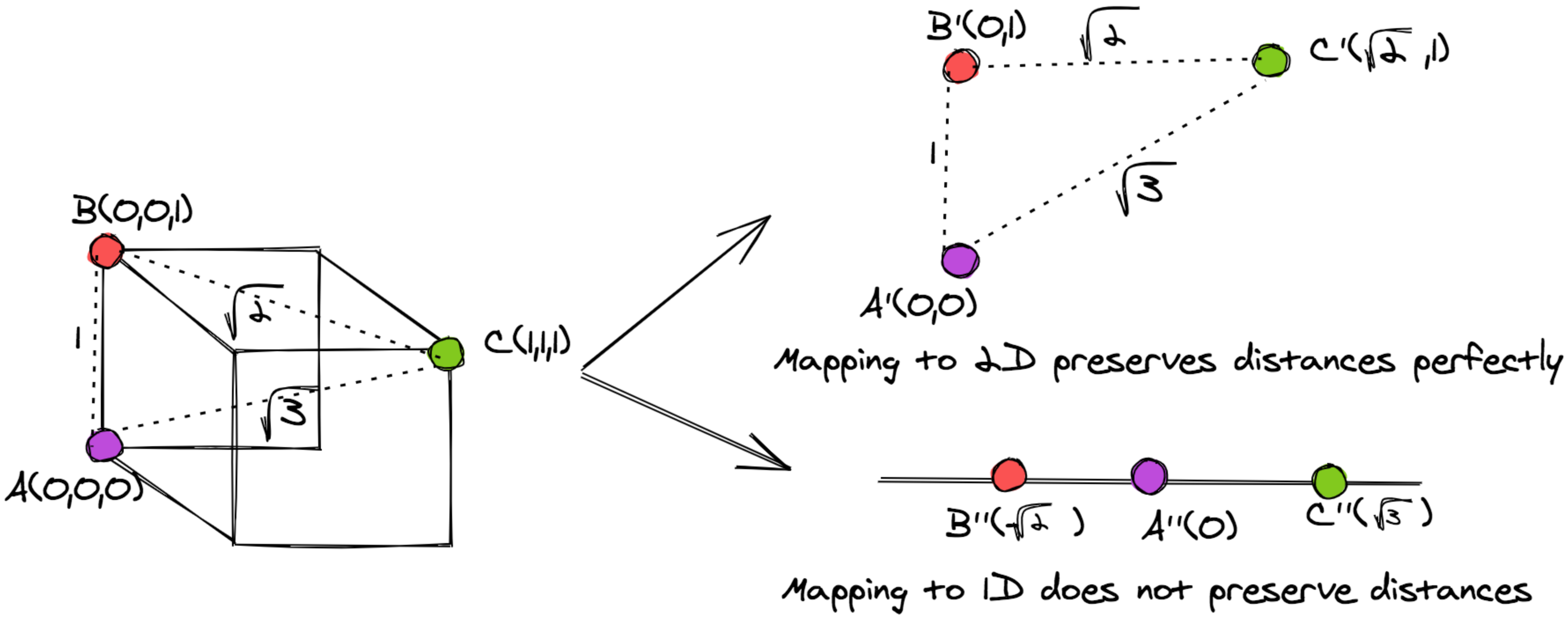
- MDS
- Isomap
- t-SNE
- LargeVis
- UMAP



https://stackabuse.com/guide-to-multidimensional-scaling-in-python-with-scikit-learn/

SIMILARITIES BETWEEN METHODS

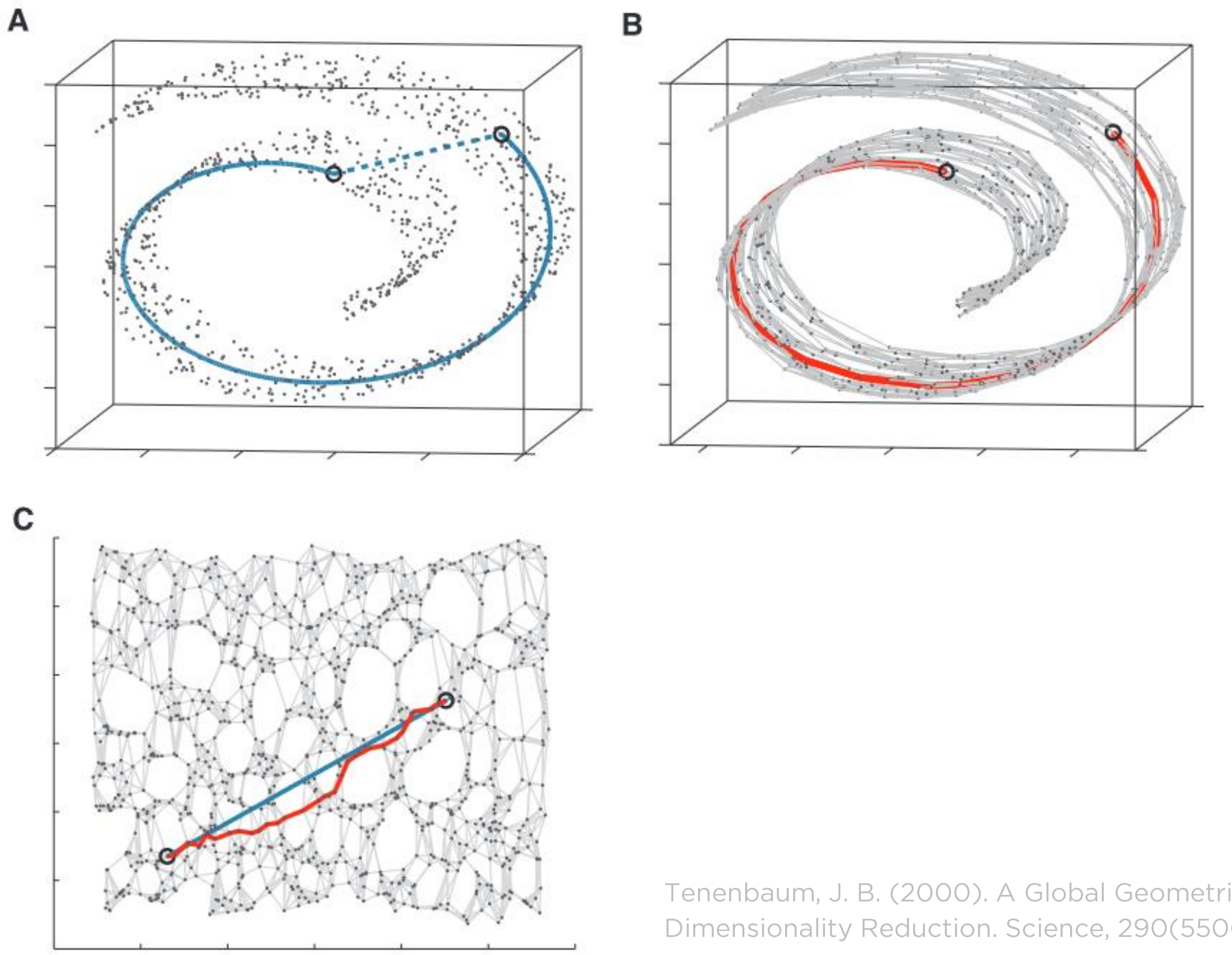
- MDS
- Isomap
- t-SNE
- LargeVis
- UMAP



https://stackabuse.com/guide-to-multidimensional-scaling-in-python-with-scikit-learn/

SIMILARITIES BETWEEN METHODS

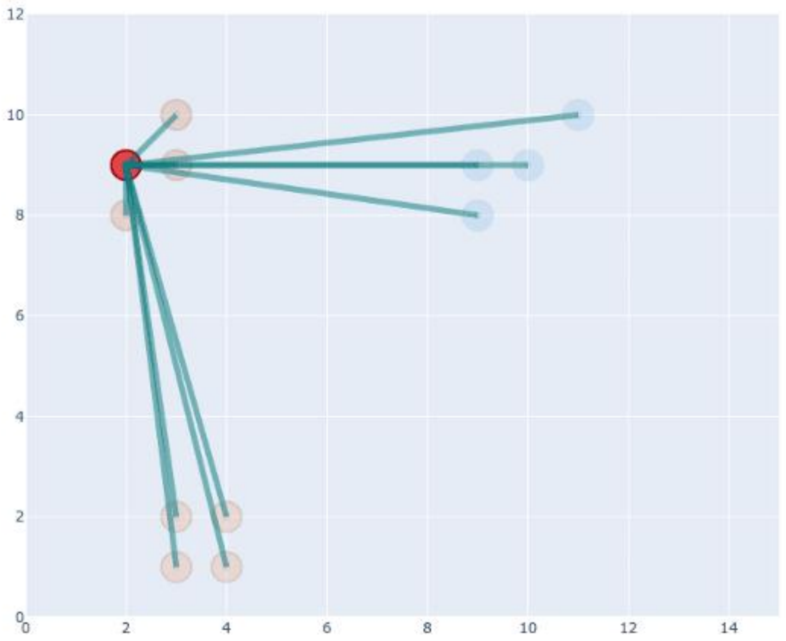
- MDS
- **Isomap**
- t-SNE
- LargeVis
- UMAP

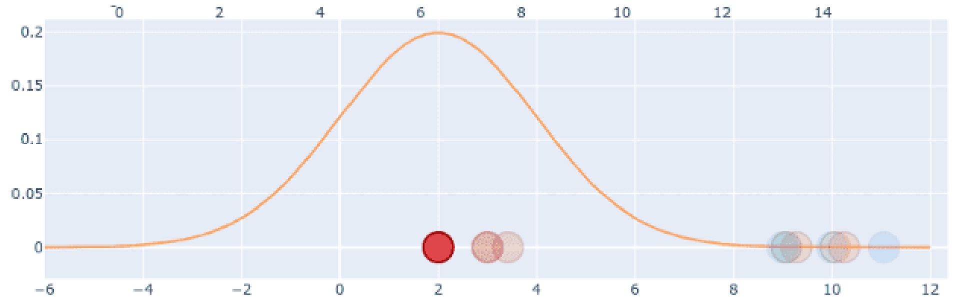


Tenenbaum, J. B. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, 290(5500), 2319–2323.

SIMILARITIES BETWEEN METHODS

- MDS
- Isomap
- t-SNE
- LargeVis
- UMAP

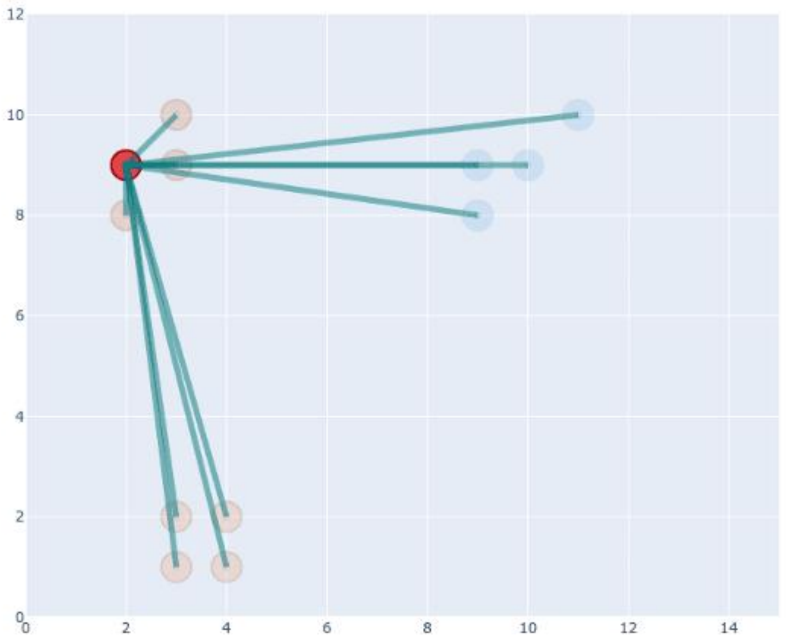


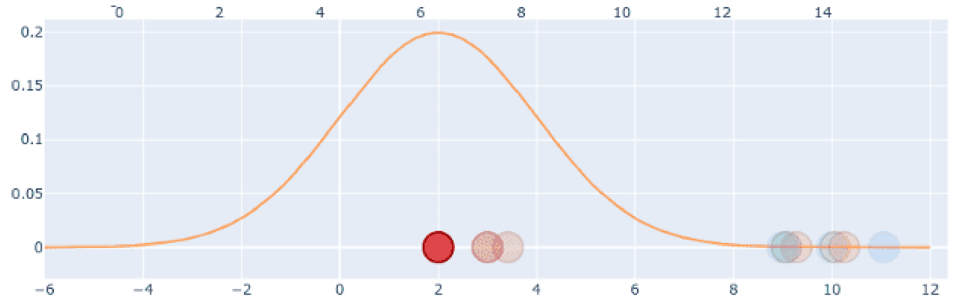


https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

SIMILARITIES BETWEEN METHODS

- MDS
- Isomap
- t-SNE
- LargeVis
- UMAP





https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

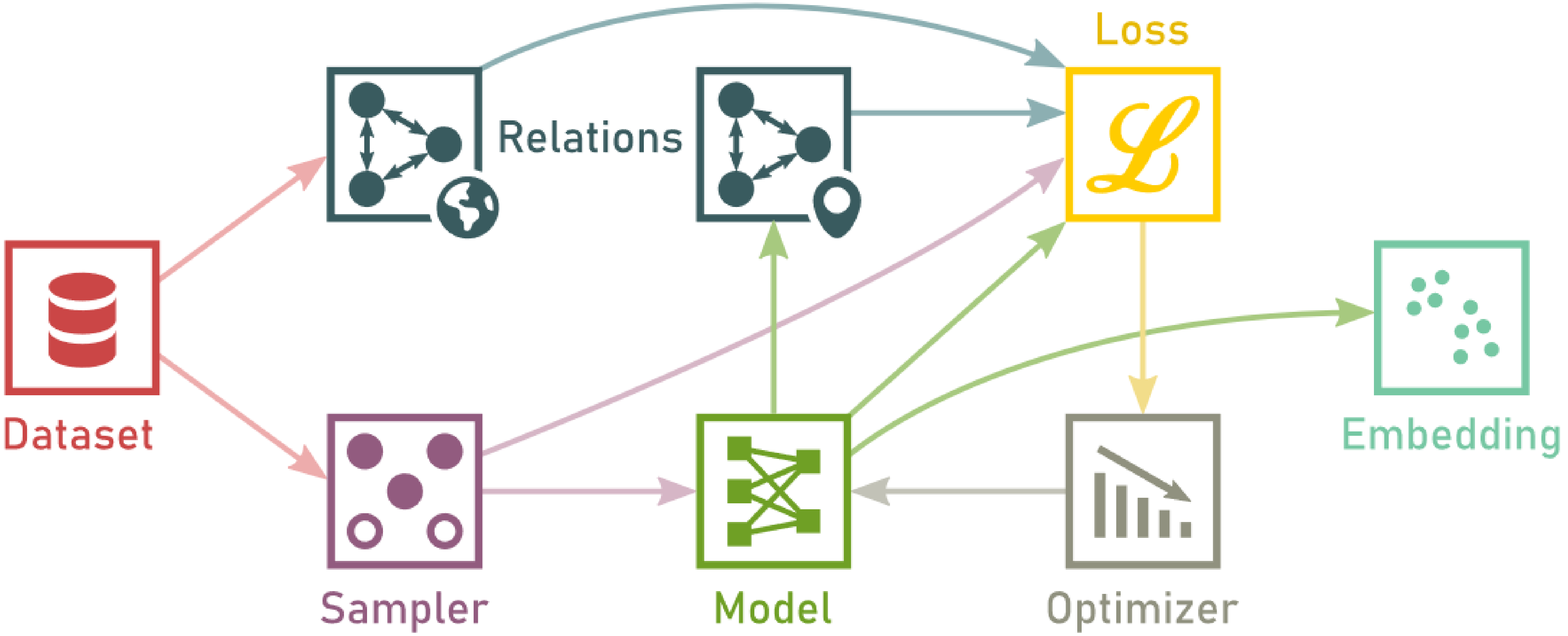
SIMILARITIES BETWEEN METHODS

- MDS
- Isomap
- t-SNE
- LargeVis
- UMAP

Summary

1. Compute distances in high-dimensional space
2. Optionally transform HD distances
3. Arrange points in low-dimensional space and compute distances
4. Optionally transform LD distances
5. Make (transformed) distances match

DATA FLOW


Loss
Relations
Dataset
Embedding
Sampler
Model
Optimizer

MOTIVATION & GOALS

Unify

Parameterize

Share & Reuse

Customize

PARAMETRIC EMBEDDINGS

- Non-parametric embeddings:

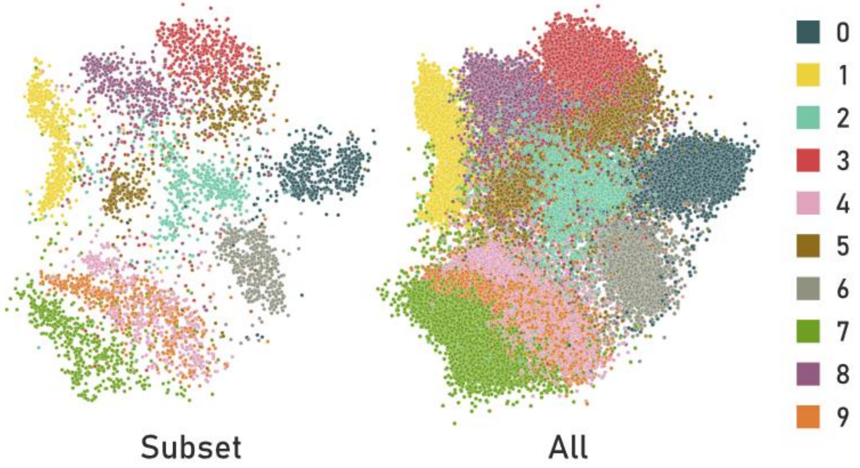
$$\mathrm{DR}(\{x_1, \ldots, x_n\}) = \arg\min_{y_i \in \mathbb{R}^k} \mathcal{L}\big(\{x_1, \ldots, x_n\}, \{y_1, \ldots, y_n\}\big)$$

- **Parametric** embeddings:

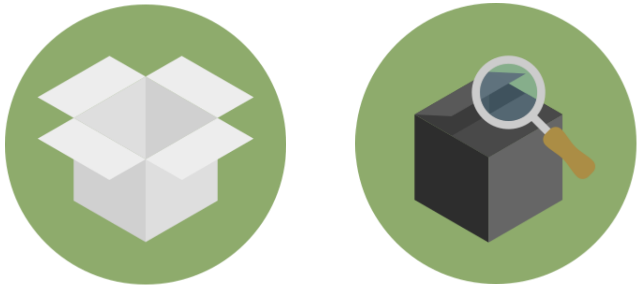
$$\mathrm{DR}_{\text{param}}(\{x_1, \ldots, x_n\}) = \arg\min_{f \in F} \mathcal{L}\big(\{x_1, \ldots, x_n\}, \{f(x_1), \ldots, f(x_n)\}\big)$$

PARAMETRIC EMBEDDINGS

- Out-of-sample extension



- Potentially explainable



- Modern infrastructure

PyTorch

TensorFlow

- Differentiable

$$\frac{\partial f}{\partial x}$$

- Customizable

$$\mathcal{L} = \sum_{i=1}^{n} w_i \times \mathcal{L}_i$$

MOTIVATION & GOALS

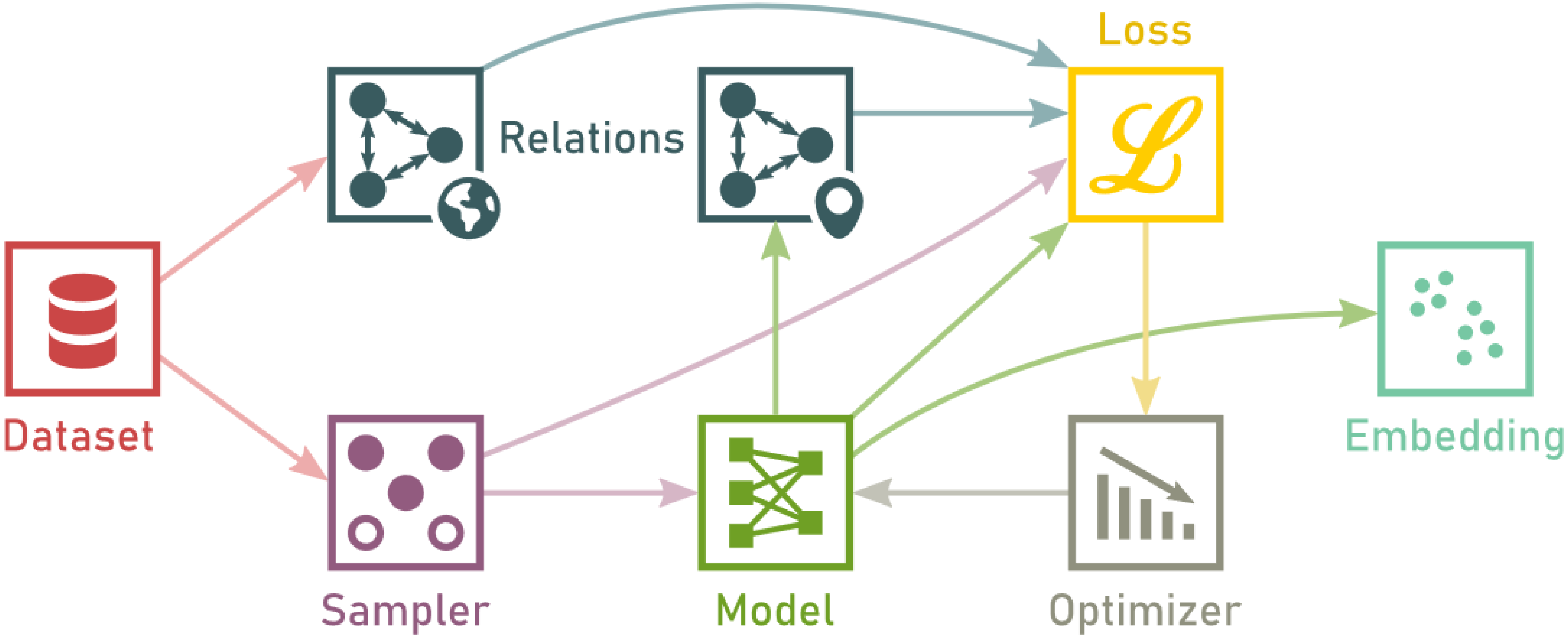
Unify

Parameterize

Share & Reuse

Customize

DATA FLOW



Loss
Relations
Dataset
Embedding
Sampler
Model
Optimizer

GRAMMAR & SPECIFICATIONS

```
relations:
```

```
losses:


training phases:
```

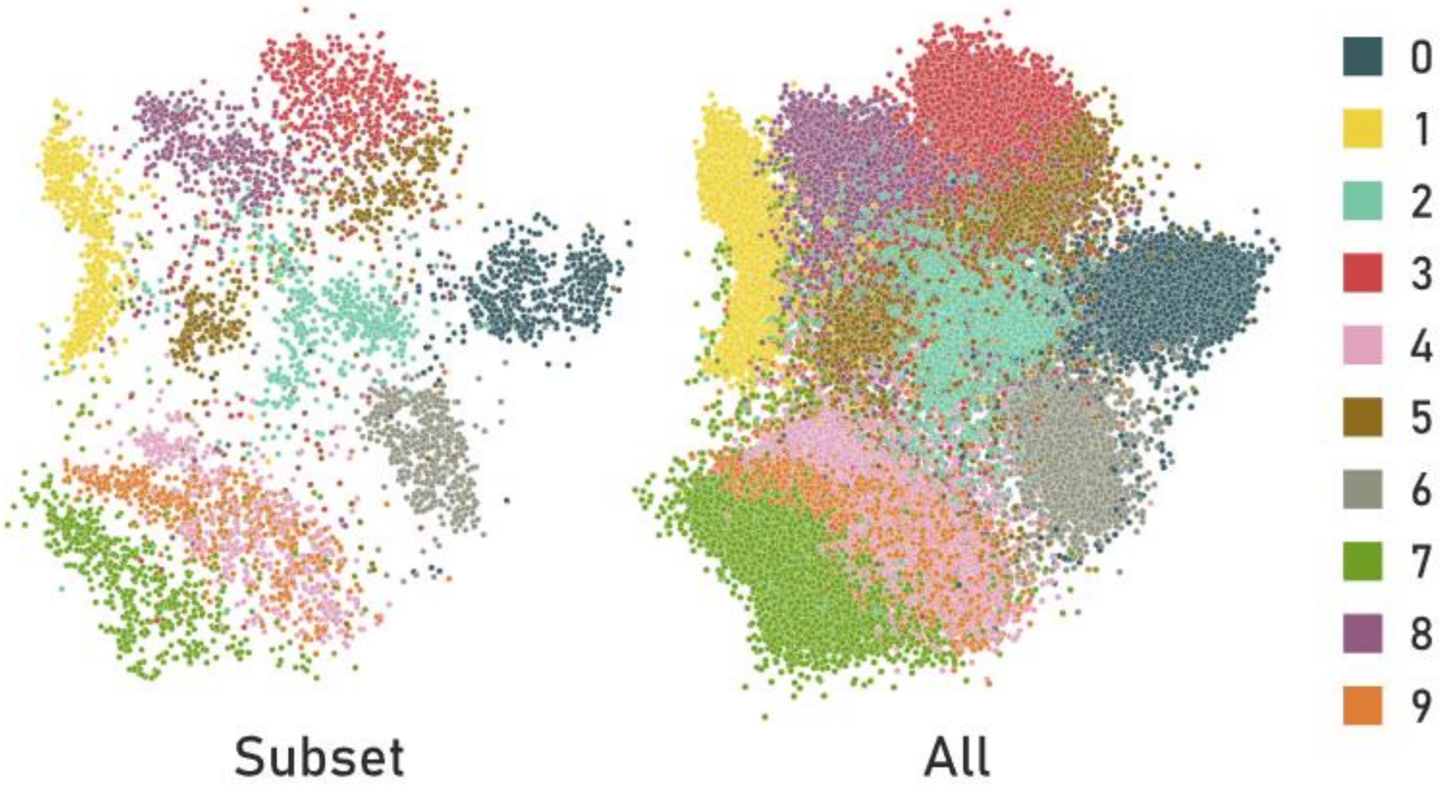
GRAMMAR & SPECIFICATIONS

```yaml
relations:
  - name: dists hd
    level: global
    type: pairwise
    options:
      metric: euclidean
  - name: dists ld
    level: batch
    type: pairwise
    options:
      metric: euclidean
```

```yaml
losses:
  - name: mds
    type: relation
    func: mse
    keys:
      rels:
        - dists hd
        - dists ld
training phases:
  - loss:
      components: mds
```

GRAMMAR & SPECIFICATIONS

```yaml
derived data:
  - name: pca
    data func: pca
    keys: [[data, main]]
relations:
  - name: p
    level: global
    type: neighbor
    data: main
    options:
      metric: euclidean
    transforms:
      - type: perplexity
        options:
          perplexity: <p>
      - type: symmetrize
      - type: normalize
  - name: q
    level: batch
    type: pairwise
    data: main
    options:
      metric: euclidean
    transforms:
      - type: t-dist
        options:
          alpha: 1.
      - type: normalize
losses:
  - name: init
    type: position
    func: mse
    keys:
      data: [main, pca]
  - name: emb
    type: relation
    func: kl div
    keys:
      rels: [p, q]
training phases:
  # pca initialization
  - loss:
      components: [init]
      sampling:
        type: item
  # main embedding
  - loss:
      components: [emb]
      sampling:
        type: item
```



MOTIVATION & GOALS

Unify

Parameterize

Share & Reuse

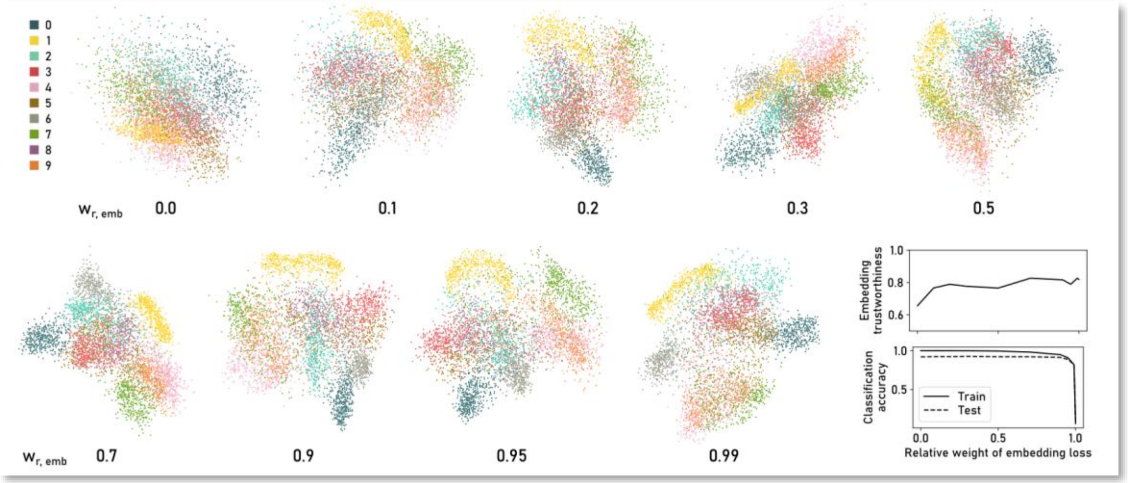
Customize

CUSTOMIZATION EXAMPLES

- Hybrid/multitask routines
  - Classification
  - Reconstruction

```
relations: <UMAP relation specs>
losses: [<UMAP loss>, <classification loss>]
training phases:
  - loss:
      components: [umap, class]
      weights: <w>
```

- Multi-phase routines
- Supervised routines



Forest covertype Dataset



SUPERVISED DR – TRIPLETS

- Aspen
- Cottonwood/willow
- Douglas-fir
- Krummholz
- Lodgepole pine
- Ponderosa pine
- Spruce/fir

R = w(t-SNE) / w(Triplet)

Parametric t-SNE (ParaDime)

t-SNE (scikit-learn)

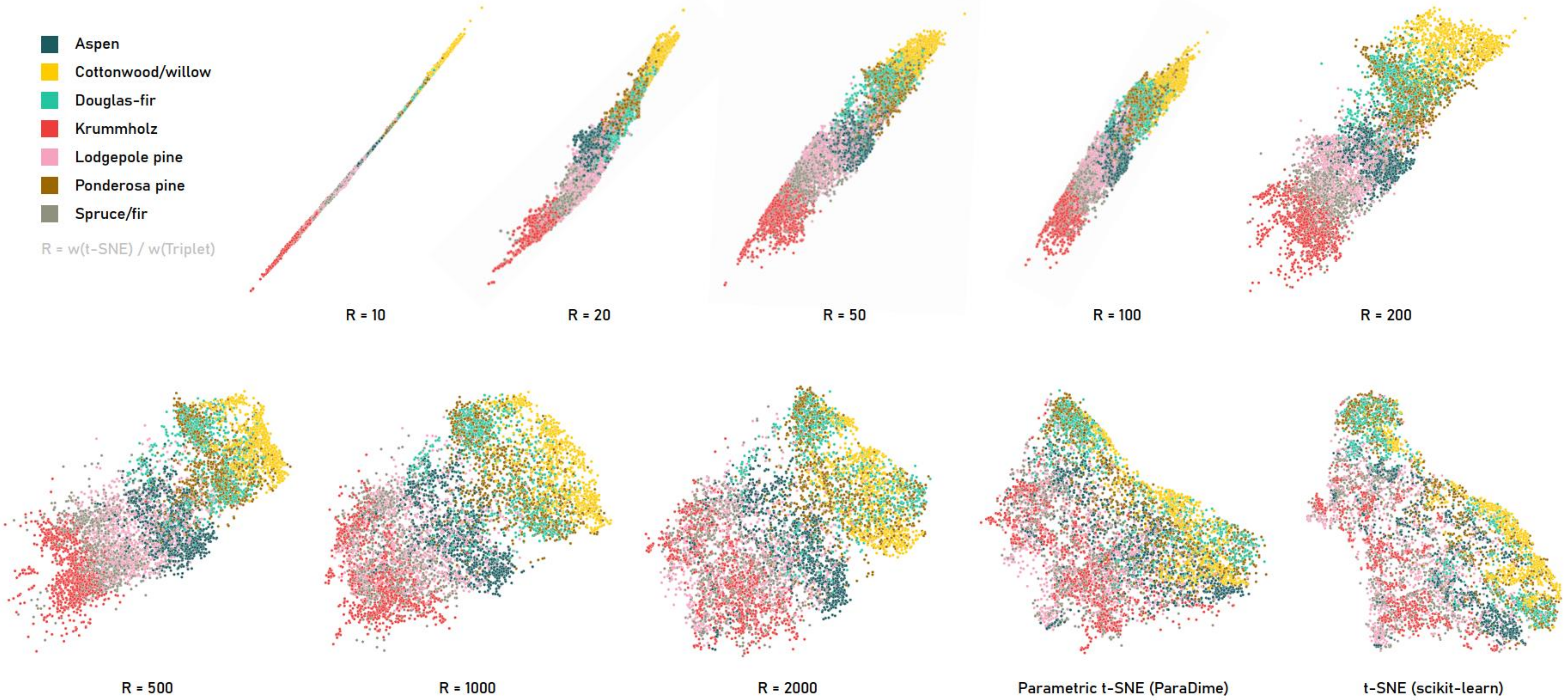
SUPERVISED DR – TRIPLETS

R = 10

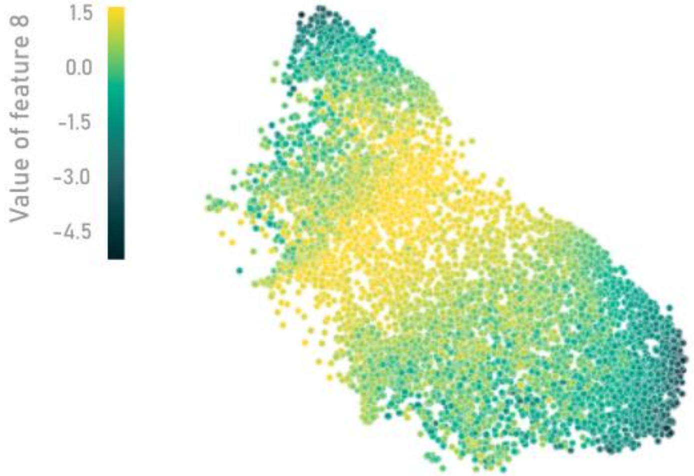
Parametric t-SNE (ParaDime)

t-SNE (scikit-learn)

SUPERVISED DR – TRIPLETS



SUPERVISED DR – CORRELATION



Unguided → Guided

SUPERVISED DR – CORRELATION

SUPERVISED DR – CORRELATION

IMPLEMENTATION & DOCUMENTATION

JKU

PyTorch

python™ Package Index

Read the Docs

Docs » paraDime: A Framework for Parametric Dimensionality Reduction   Edit on GitHub

paraDime

latest

Search docs

USER GUIDE

- Installation
- Simple Usage
- Building Blocks of a paraDime Routine
- Examples

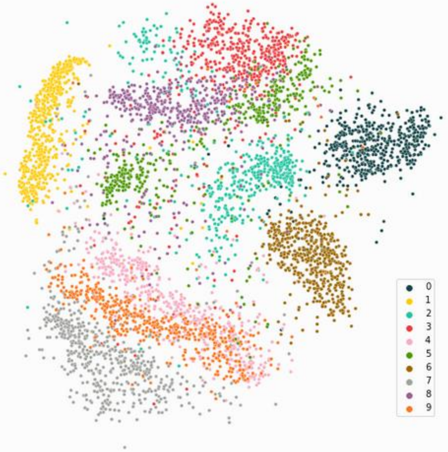
API REFERENCE

- API Reference

paraDime: A Framework for Parametric Dimensionality Reduction

paraDime is a modular framework for specifying and training parametric dimensionality reduction (DR) models. These models allow you to add new data points to existing low-dimensional representations of high-dimensional data. ParaDime DR models are constructed from simple building blocks (such as Relations and Relation Transforms), so that experimentation with novel DR techniques becomes easy.

Here you can see a parametric version of t-SNE [1] trained on a subset of 5000 images of handwritten digits from the MNIST dataset [2]:



The rest of the 60,000 images can then be easily embedded into the same space without retraining the t-SNE:

Read the Docs   v: latest

https://paradime.readthedocs.io/

CONCLUSION

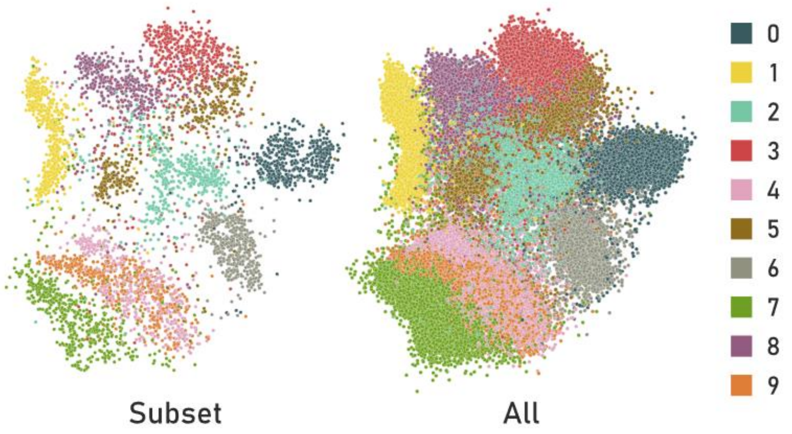
paraDime

Unify

Parameterize

Customize

Share



JKU

JOHANNES KEPLER
UNIVERSITÄT LINZ