# Machine learning for visualization

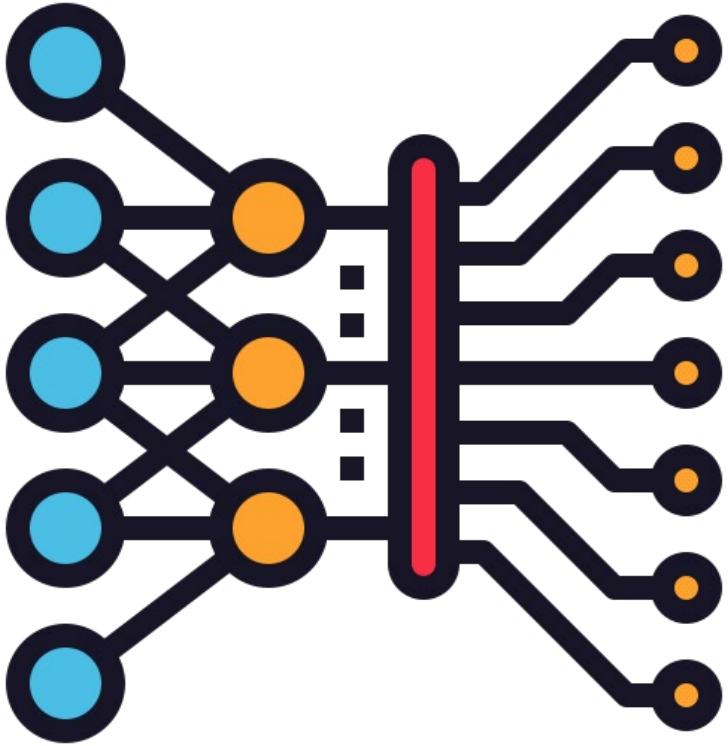
## CS524: Big Data Visualization & Visual Analytics

**Fabio Miranda**

**https://fmiranda.me**

Slides based on Claudio Silva's ml+vis course

# Machine learning is pervasive.



# Machine learning in many domains

Banks are using tabular data to identify fraud or calculate loan risk.

Machine translation has improved remarkably over the past few years.

Healthcare providers look to interpretable models to guide decisions.

# Understanding the gap

- Model understanding has yet to catch up to model performance improvements.
- Visualization is key to improve model understanding to a wide range of audiences.
- There are many rich domains to apply VisML techniques, including industry and research.

# Where does visualization fit in?

Many tasks in machine learning inevitably involve some form of visualization.

In other cases, visualization is a byproduct of the process.

We have also seen an increase in the number of interactive visualization systems, designed with machine learning as a critical component.

# Visual Analytics in Deep Learning | Interrogative Survey Overview

## §4 WHY

***Why would one want to use visualization in deep learning?***

Interpretability & Explainability
Debugging & Improving Models
Comparing & Selecting Models
Teaching Deep Learning Concepts

## §6 WHAT

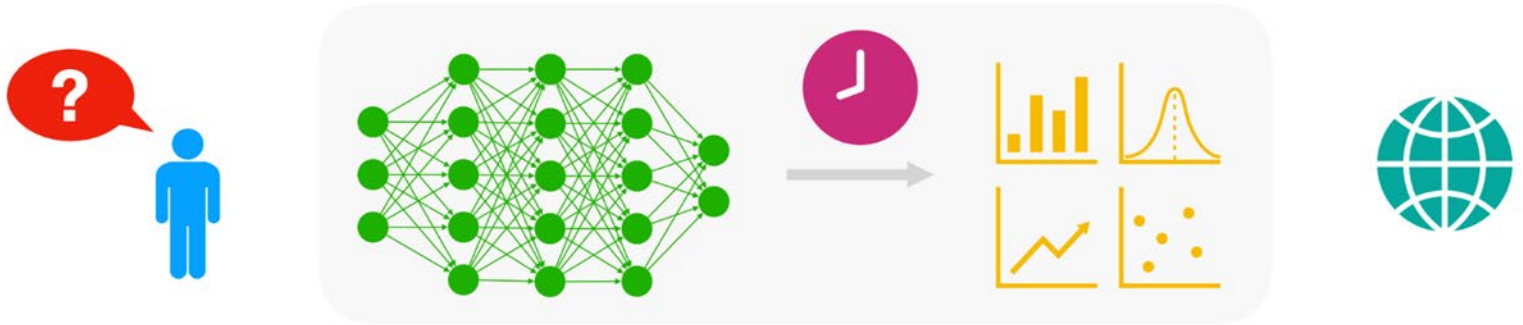
***What data, features, and relationships in deep learning can be visualized?***

Computational Graph & Network Architecture
Learned Model Parameters
Individual Computational Units
Neurons In High-dimensional Space
Aggregated Information

## §8 WHEN

***When in the deep learning process is visualization used?***

During Training
After Training



## §5 WHO

***Who would use and benefit from visualizing deep learning?***

Model Developers & Builders
Model Users
Non-experts

## §7 HOW

***How can we visualize deep learning data, features, and relationships?***

Node-link Diagrams for Network Architecture
Dimensionality Reduction & Scatter Plots
Line Charts for Temporal Metrics
Instance-based Analysis & Exploration
Interactive Experimentation
Algorithms for Attribution & Feature Visualization

## §9 WHERE

***Where has deep learning visualization been used?***

Application Domains & Models
A Vibrant Research Community

[Hohman et al., 2018]

# Dimensionality reduction

- Input data may have thousands or millions of dimensions
  - E.g., text, images, videos, …
- **Dimensionality reduction** aims to represent data with fewer dimensions.
  - Easier learning: fewer parameters.
  - Discover “intrinsic dimensionality” of data.
    - High dimensional data that is truly lower dimensional.
    - Noise reduction.
  - Visualization: show high-dimensional data in a visual space (2D or 3D).
    - How are the points spread?
    - Are there well defined groups of similar instances?

| Technique | Taxonomy | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data Types | | | | | Linearity | Supervision | Multi-level | Locality | Steerability | Stability | OOC | Comp. |
| | Di | Or | Ca | Ne | Ct | | | | | | | | |
| PCA [65] | | | ✔ | | | ✔ | | | | | • | ✔ | $O(p^3)$ |
| LDA [50] | | | ✔ | | | ✔ | ✔ | | | | | ✔ | $O(np^2+p^3)$ |
| Classical MDS [155] | ✔ | | ✔ | • | | ✔ | • | | | • | • | | $O(n^3)$ |
| Kruskal [79] | ✔ | ✔ | ✔ | • | | | • | | | • | | • | $O(in^2)$ |
| NLM [132] | ✔ | | ✔ | • | | | • | | | • | | • | $O(in^2)$ |
| MCA [17] | | | • | | ✔ | | | | | | | | $O(n^3)$ |
| Smacof [42] | ✔ | ✔ | ✔ | • | | | • | | | | | • | $O(in^2)$ |
| SOM [126] | | | ✔ | | | | | | | | | ✔ | $O(l^2np+l^2)$ |
| FastMap [48] | ✔ | | ✔ | • | | | • | | | | | • | $O(n)$ |
| Chalmers [32] | ✔ | | ✔ | • | | | • | • | | • | ○ | ✔ | $O(in)$ |
| GTM [22] | | | ✔ | | | | | | | | | • | $O((lnp)^3+m^3)$ |
| Pekalska [120] | ✔ | | ✔ | • | | ✔ | • | | | | | | $O(r^3+rn)$ |
| CCA [43] | ✔ | | ✔ | • | | | • | | | | | ✔ | $O(l^2)$ |
| LLE [129] | ✔ | | ✔ | • | | | • | | ✔ | | | • | $O(n^3)$ |
| Isomap [153] | ✔ | | ✔ | ✔ | | | • | | | | | • | $O(n^3)$ |
| Lapl. Eigenmaps [15] | ✔ | | ✔ | ✔ | | | • | | ✔ | | | • | $O(n^3)$ |
| Force-Directed [152] | ✔ | | ✔ | | | | • | • | | | | ✔ | $O(in^2)$ |
| LTSA [180] | | | ✔ | | | • | | | ✔ | | | | $O(n^3)$ |
| MVU [169] | ✔ | | ✔ | • | | | • | | ✔ | | | • | $O(n^3)$ |
| LSP [117] | ✔ | | ✔ | ✔ | | | • | • | ✔ | ✔ | ○ | | $O(n^3)$ |
| SNE [64] | ✔ | | ✔ | • | | | • | • | | ○ | | • | $O(in^2)$ |
| PLMP [118] | | | ✔ | | | • | • | • | ✔ | ✔ | ○ | | $O(n^3)$ |
| LAMP [73] | | | ✔ | | | | • | • | ✔ | ✔ | ✔ | ✔ | $O(pn)$ |
| RBF-MP [2] | ✔ | | ✔ | • | | | • | | | ✔ | ○ | • | $O(r^3+n)$ |
| LoCH [47] | ✔ | | ✔ | • | | | • | | ✔ | ✔ | | ✔ | $O(n\sqrt{(n)})$ |
| ClassiMap [90] | ✔ | | ✔ | • | | | ✔ | | | | | • | $O(in^2)$ |
| Kelp [13] | ✔ | | ✔ | • | | • | • | | ✔ | ✔ | ○ | • | $O(r^3)$ |

Core Techniques

| Technique | Taxonomy | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data Types | | | | | Linearity | Supervision | Multi-level | Locality | Steerability | Stability | OOC | Comp. |
| | Di | Or | Ca | Ne | Ct | | | | | | | | |
| PCA [65] | | | | | | ✔ | | | | | • | ✔ | $O(p^3)$ |
| LDA [50] | | | | | | ✔ | ✔ | | | | | ✔ | $O(np^2+p^3)$ |
| Classical MDS [155] | ✔ | | ✔ | • | | ✔ | • | | | • | • | | $O(n^3)$ |
| Kruskal [79] | ✔ | ✔ | ✔ | • | | | • | | | • | | • | $O(in^2)$ |
| NLM [132] | ✔ | | ✔ | • | | | • | | | • | | • | $O(in^2)$ |
| MCA [17] | | | • | | ✔ | | | | | | | | $O(n^3)$ |
| Smacof [42] | ✔ | ✔ | ✔ | • | | | • | | | | | • | $O(in^2)$ |
| SOM [126] | | | ✔ | | | | | | | | | ✔ | $O(l^2np+l^2)$ |
| FastMap [48] | ✔ | | ✔ | • | | | • | | | | | • | $O(n)$ |
| Chalmers [32] | ✔ | | ✔ | • | | | • | • | | • | ○ | ✔ | $O(in)$ |
| GTM [22] | | | ✔ | | | | | | | | | • | $O((lnp)^3+m^3)$ |
| Pekalska [120] | ✔ | | ✔ | • | | ✔ | • | | | | | | $O(r^3+rn)$ |
| CCA [43] | ✔ | | ✔ | • | | | • | | | | | ✔ | $O(l^2)$ |
| LLE [129] | ✔ | | ✔ | • | | | • | | ✔ | | | • | $O(n^3)$ |
| Isomap [153] | ✔ | | ✔ | ✔ | | | • | | | | | • | $O(n^3)$ |
| Lapl. Eigenmaps [15] | ✔ | | ✔ | ✔ | | | • | | ✔ | | | • | $O(n^3)$ |
| Force-Directed [152] | ✔ | | ✔ | | | | • | • | | | | ✔ | $O(in^2)$ |
| LTSA [180] | | | ✔ | | | • | | | ✔ | | | | $O(n^3)$ |
| MVU [169] | ✔ | | ✔ | • | | | • | | ✔ | | | • | $O(n^3)$ |
| LSP [117] | ✔ | | ✔ | ✔ | | | • | • | ✔ | ✔ | ○ | | $O(n^3)$ |
| SNE [64] | ✔ | | ✔ | • | | | • | • | | ○ | | • | $O(in^2)$ |
| PLMP [118] | | | ✔ | | | • | • | • | ✔ | ✔ | ○ | | $O(n^3)$ |
| LAMP [73] | | | ✔ | | | | • | • | ✔ | ✔ | ✔ | ✔ | $O(pn)$ |
| RBF-MP [2] | ✔ | | ✔ | • | | | • | | | ✔ | ○ | • | $O(r^3+n)$ |
| LoCH [47] | ✔ | | ✔ | • | | | • | | ✔ | ✔ | | ✔ | $O(n\sqrt{(n)})$ |
| ClassiMap [90] | ✔ | | ✔ | • | | | ✔ | | | | | • | $O(in^2)$ |
| Kelp [13] | ✔ | | ✔ | • | | • | • | | ✔ | ✔ | ○ | • | $O(r^3)$ |

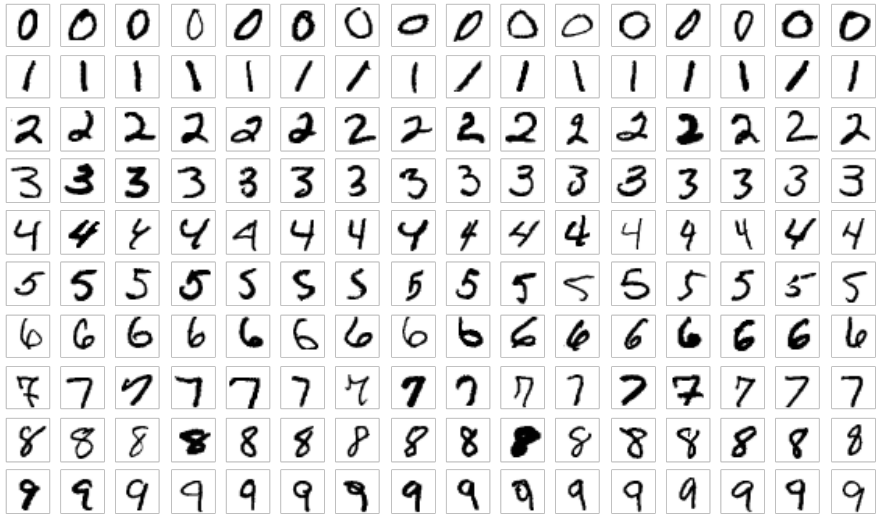
| Technique | Taxonomy | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data Types | | | | | Linearity | Supervision | Multi-level | Locality | Steerability | Stability | OOC | Comp. |
| | Di | Or | Ca | Ne | Ct | | | | | | | | |
| PCA [65] | | | | | | ✔ | | | | | • | ✔ | $O(p^3)$ |
| LDA [50] | | | | | | ✔ | ✔ | | | | | ✔ | $O(np^2+p^3)$ |
| Classical MDS [155] | ✔ | | ✔ | • | | ✔ | • | | | • | • | | $O(n^3)$ |
| Kruskal [79] | ✔ | ✔ | ✔ | • | | | • | | | • | | • | $O(in^2)$ |
| NLM [132] | ✔ | | ✔ | • | | | • | | | • | | • | $O(in^2)$ |
| MCA [17] | | | • | | ✔ | | | | | | | | $O(n^3)$ |
| Smacof [42] | ✔ | ✔ | ✔ | • | | | • | | | | | • | $O(in^2)$ |
| SOM [126] | | | ✔ | | | | | | | | | ✔ | $O(l^2np+l^2)$ |
| FastMap [48] | ✔ | | ✔ | • | | | • | | | | | • | $O(n)$ |
| Chalmers [32] | ✔ | | ✔ | • | | | | | | | | | $O(in)$ |
| GTM [22] | | | ✔ | | | | | | | | | | $O((lnp)^3+m^3)$ |
| Pekalska [120] | ✔ | | ✔ | • | | | | | | | | | $O(r^3+rn)$ |
| CCA [43] | ✔ | | ✔ | • | | | | | | | | ✔ | $O(l^2)$ |
| LLE [129] | ✔ | | ✔ | • | | | | | | | | • | $O(n^3)$ |
| Isomap [153] | ✔ | | ✔ | ✔ | | | | | | | | • | $O(n^3)$ |
| Lapl. Eigenmaps [15] | ✔ | | ✔ | ✔ | | | | | | | | • | $O(n^3)$ |
| Force-Directed [152] | ✔ | | ✔ | | | | | | | | | ✔ | $O(in^2)$ |
| LTSA [180] | | | ✔ | | | | | | | | | | $O(n^3)$ |
| MVU [169] | ✔ | | ✔ | • | | | | | | | | • | $O(n^3)$ |
| LSP [117] | ✔ | | ✔ | ✔ | | | | | | ✔ | ○ | | $O(n^3)$ |
| SNE [64] | ✔ | | ✔ | • | | | | | | ○ | | • | $O(in^2)$ |
| PLMP [118] | | | ✔ | | | | | | | ✔ | ○ | | $O(n^3)$ |
| LAMP [73] | | | ✔ | | | | | | | ✔ | ✔ | ✔ | $O(pn)$ |
| RBF-MP [2] | ✔ | | ✔ | • | | | • | | | ✔ | ○ | • | $O(r^3+n)$ |
| LoCH [47] | ✔ | | ✔ | • | | | • | | ✔ | ✔ | | ✔ | $O(n\sqrt{(n)})$ |
| ClassiMap [90] | ✔ | | ✔ | • | | | ✔ | | | | | • | $O(in^2)$ |
| Kelp [13] | ✔ | | ✔ | • | | • | • | | ✔ | ✔ | ○ | • | $O(r^3)$ |

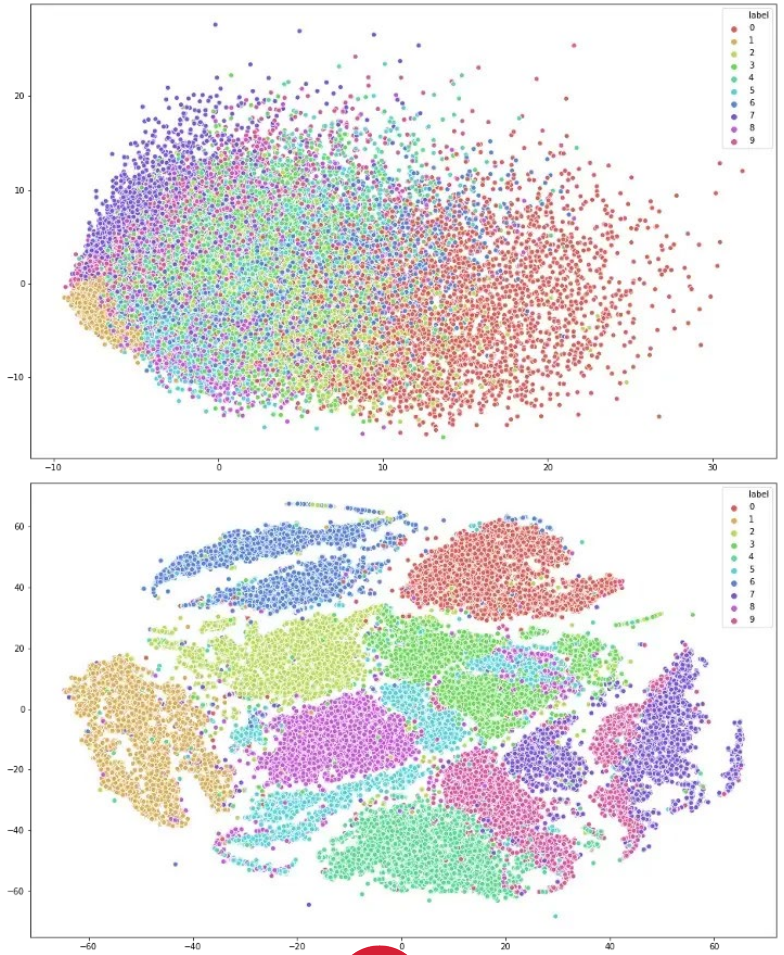
Core Techniques

Many Variants

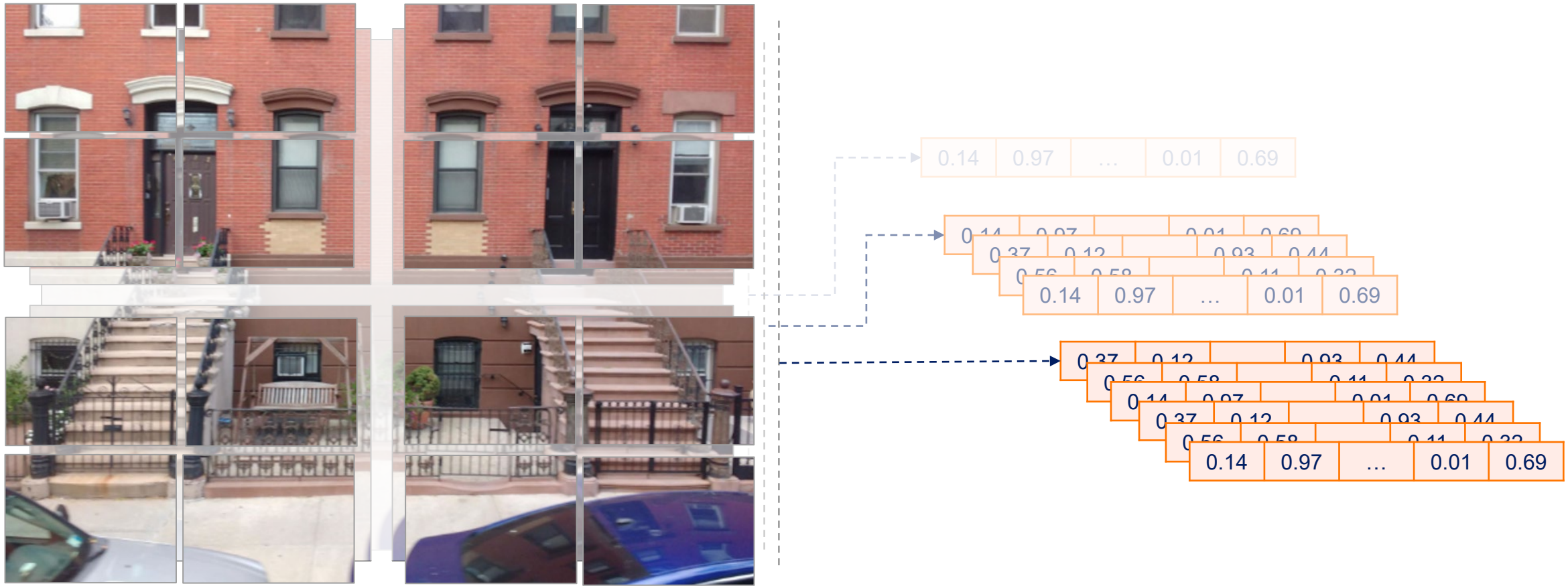
| Core Method | Variants |
|---|---|
| **PCA** [65] | see [181] for variants; iPCA [71], PCP [182] |
| **LDA** [50] | see [61] for variants |
| **Classical MDS** [155] | L-MDS [144], Pivot-MDS [24], CFMDS [113] |
| **MCA** [17] | see [154] for variants |
| **SOM** [126] | see [150] for variants |
| **Chalmers** [32] | Glimmer [70] |
| **LLE** [129] | SLLE [178] |
| **Isomap** [153] | S-Isomap [53] |
| **LTSA** [180] | LLTSA [179] |
| **MVU** [169] | MUHSIC [147] |
| **LSP** [117] | PLP [115], E-LSP [33], Hipp [116] |
| **SNE** [64] | NeRV [162], t-SNE [158], DS t-SNE [77], Q-SNE [69] A-tSNE [122], H-SNE [121], BH-tSNE [157] |

# Dimensionality reduction

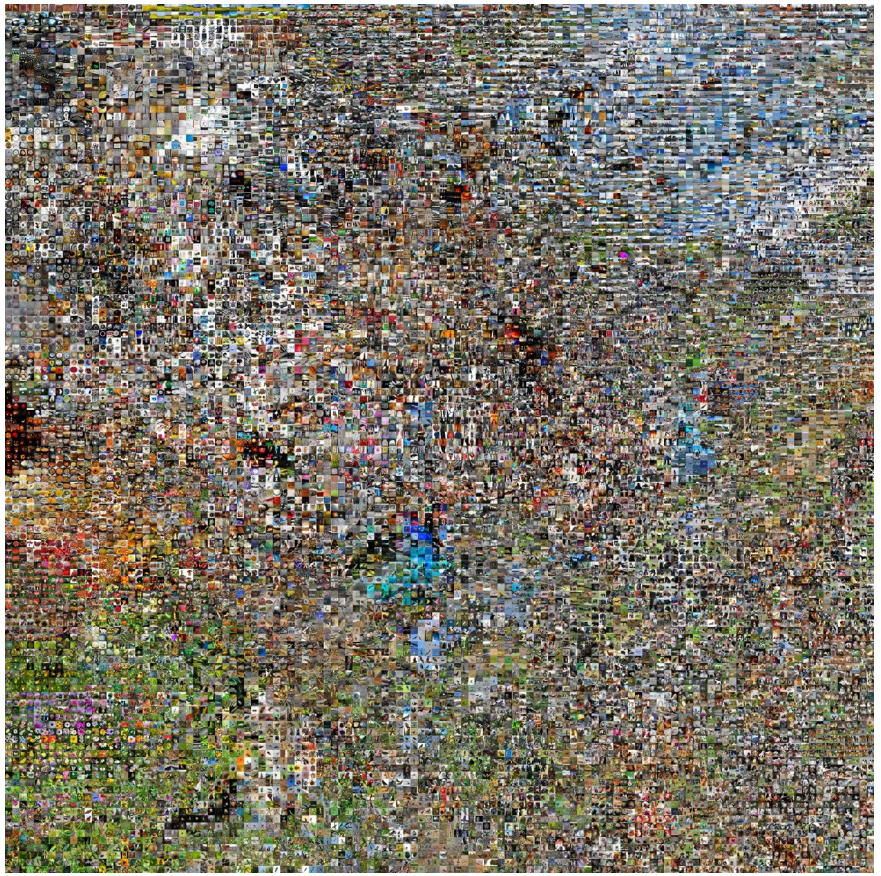




# Image embeddings



# Dimensionality reduction

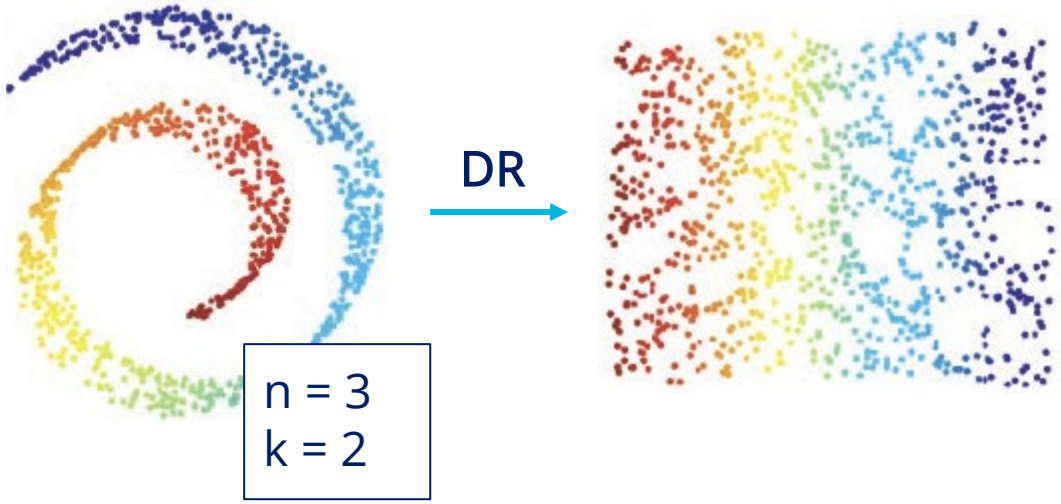


# Dimensionality reduction

# Dimensionality reduction

- High-dimensional data typically has an intrinsic dimension smaller than the space in which it is embedded.

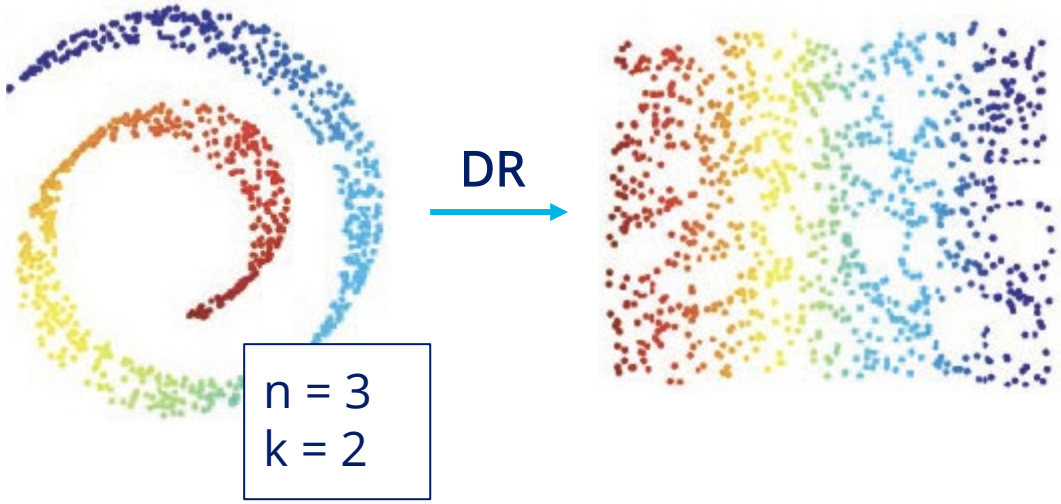
DR comprises a category of unsupervised methods that aim to capture, to a certain extent, the intrinsic dimension of the data.



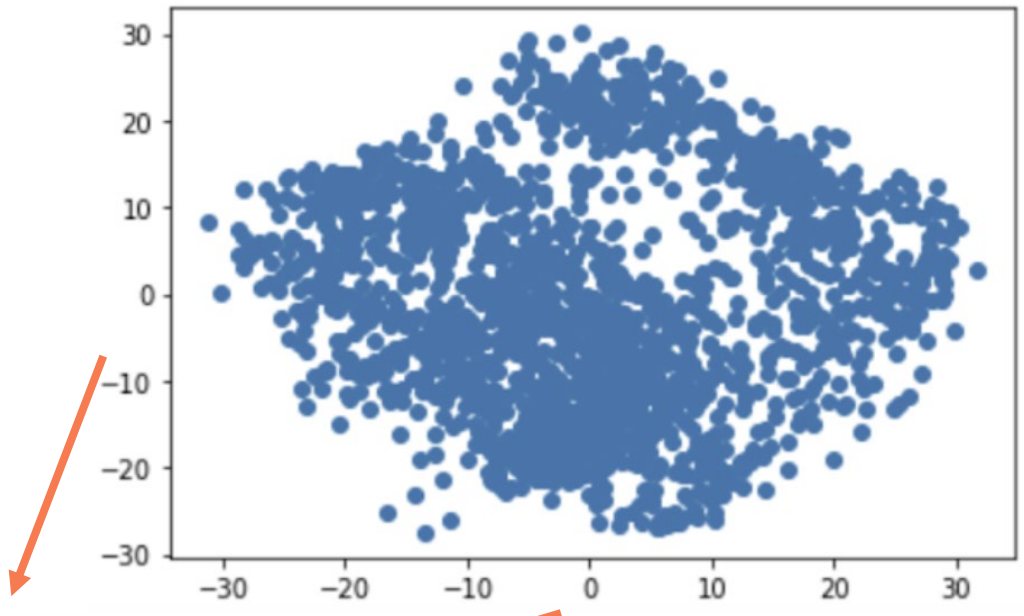
# Dimensionality reduction

- High-dimensional data typically has an intrinsic dimension smaller than the space in which it is embedded.

The goal is to perform the dimensionality reduction preserving some similarity information among data instances.



# Dimensionality reduction




It is not possible to interpret the meaning of the axes.

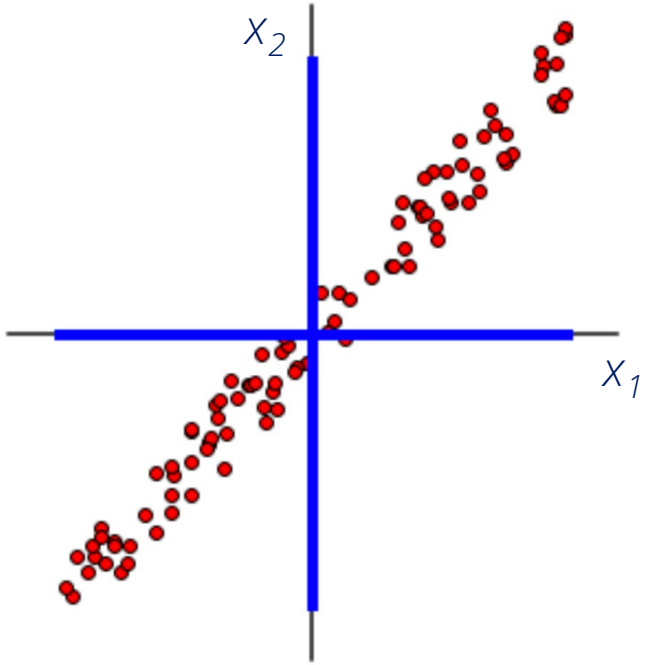
Dimensionality reduction techniques embed data in a latent space where the point coordinates (axes) do not have a semantic meaning.

# $\Omega \rightarrow \mathbb{R}^p$

- Dimensionality reduction is a mapping between two spaces.
- For visualization purposes, usually $p = 2$ or $p = 3$ .
- Principal Component Analysis (PCA).
- Multidimensional Scaling (MDS).
- t-distributed Stochastic Neighbor Embedding (t-SNE).
- Uniform Manifold Approximation and Projection (UMAP).
- Local Affine Multidimensional Projection (LAMP).

# Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.



# Principal Component Analysis (PCA)

Given a dataset of $n$ points $a_1, \ldots, a_n \in \mathbb{R}^d$:

- $d = 1$
  - Mean
    - $\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}$
  - Variance
    - $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu) \in \mathbb{R}$

# Principal Component Analysis (PCA)

Given a dataset of $n$ points $a_1, \ldots, a_n \in \mathbb{R}^d$:

- $d = 1$
  - Mean
$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}$$
  - Variance
$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu) \in \mathbb{R}$$

- $d \geq 2$
  - Mean
$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}^d$$
  - Covariance matrix
$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu)\,(a_i - \mu)^T \in \mathbb{R}^{dxd}$$

# Principal Component Analysis (PCA)

Given a dataset of $n$ points $a_1, \ldots, a_n \in \mathbb{R}^d$:

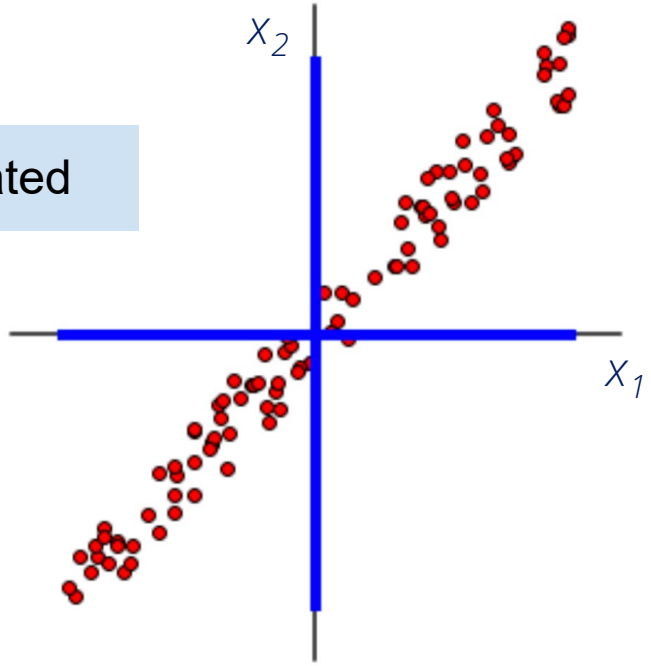
- $d = 1$
  - Mean
$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}$$
  - Variance
$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu) \in \mathbb{R}$$

- $d \geq 2$
  - Mean
$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}^d$$
  - Covariance matrix
$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu)\,(a_i - \mu)^T \in \mathbb{R}^{dxd}$$
$$= \frac{1}{n}\sum_{i=1}^{n} a_i a_i^T \quad if \quad \mu = 0$$

# Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.

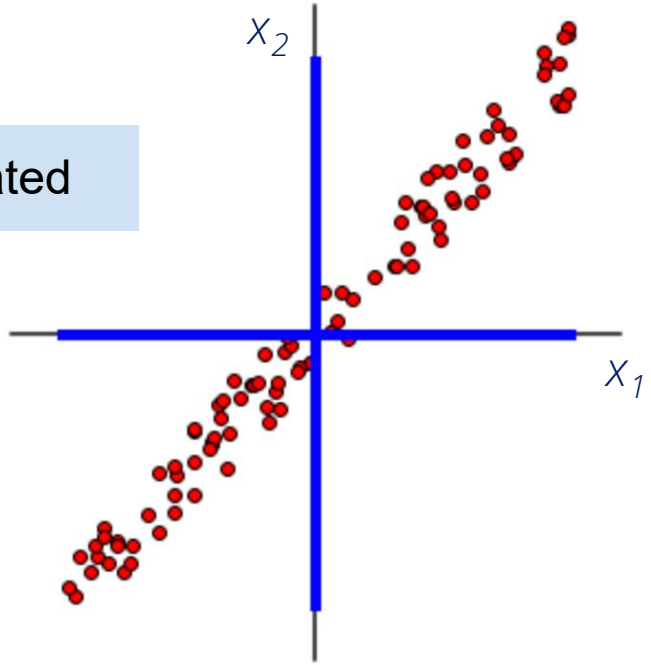
Highly correlated

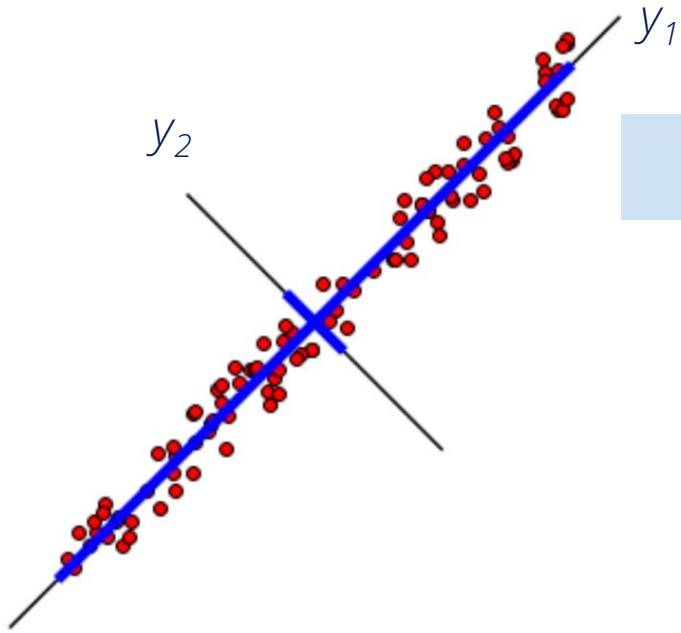


# Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.

Highly correlated




Uncorrelated

# Principal Component Analysis (PCA)

- Given a dataset of $n$ points $a_1, \ldots, a_n \in \mathbb{R}^d$, where $d$ is large.
- Goal: represent this dataset in lower dimension,
  i.e., find $a_1, \ldots, a_n \in \mathbb{R}^k$ where $k \ll d$.
- Assume that the dataset is centered:

# Principal Component Analysis (PCA)

- Given a dataset of $n$ points $a_1, \ldots, a_n \in \mathbb{R}^d$, where $d$ is large.
- Goal: represent this dataset in lower dimension, i.e., find $a_1, \ldots, a_n \in \mathbb{R}^k$ where $k \ll d$.
- Assume that the dataset is centered:

$$\sum_{i=1}^{n} a_i = 0$$

# Principal Component Analysis (PCA)

- Given a dataset of $n$ points $a_1, \ldots, a_n \in \mathbb{R}^d$, where $d$ is large.
- Goal: represent this dataset in lower dimension, i.e., find $a_1, \ldots, a_n \in \mathbb{R}^k$ where $k \ll d$.
- Assume that the dataset is centered:

$$\sum_{i=1}^{n} a_i = 0$$

$$\mathbf{S} = \sum_{i=1}^{n} a_i a_i^T = \mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$$

# Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.
- Therefore, the goal is to find a change of basis matrix $\mathbf{P}$ such that:

$$\mathbf{Y} = \mathbf{PX} \Rightarrow \mathbf{YY}^{\mathbf{T}} = \mathbf{D}$$

Coordinate system in which the variables are uncorrelated.

# Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.
- Therefore, the goal is to find a change of basis matrix $\mathbf{P}$ such that:

$$\mathbf{Y} = \mathbf{PX} \Rightarrow \mathbf{YY}^{\mathbf{T}} = \mathbf{D}$$

Covariance matrix (new coordinates)

# Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.
- Therefore, the goal is to find a change of basis matrix $\mathbf{P}$ such that:

$$\mathbf{Y} = \mathbf{PX} \Rightarrow \mathbf{YY}^{\mathbf{T}} = \mathbf{D}$$

$$\mathbf{YY}^{\mathbf{T}} = (\mathbf{PX})(\mathbf{PX})^{\mathbf{T}} = \mathbf{PXX}^{\mathbf{T}}\mathbf{P}^{\mathbf{T}}$$

Covariance matrix (old coordinates)

# Spectral theorem

Theorem: Let $\mathbf{A} \in \mathbb{R}^{nxn}$ be a symmetric matrix. Then there exists an orthogonal matrix $\mathbf{P}$ and a diagonal matrix $\mathbf{D}$ of sizes $n\ x\ n$ such that

$$\mathbf{A} = \mathbf{PDP^T}$$

Entries of the diagonal of $\mathbf{D}$ are the eigenvalues of $\mathbf{A}$.

Column vectors of $\mathbf{P}$ are the eigenvectors of $\mathbf{A}$.

# Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.
- Therefore, the goal is to find a change of basis matrix $\mathbf{P}$ such that:

$$\mathbf{Y} = \mathbf{PX} \Rightarrow \mathbf{YY}^{\mathbf{T}} = \mathbf{D}$$
$$\mathbf{YY}^{\mathbf{T}} = (\mathbf{PX})(\mathbf{PX})^{\mathbf{T}} = \mathbf{PXX}^{\mathbf{T}}\mathbf{P}^{\mathbf{T}}$$

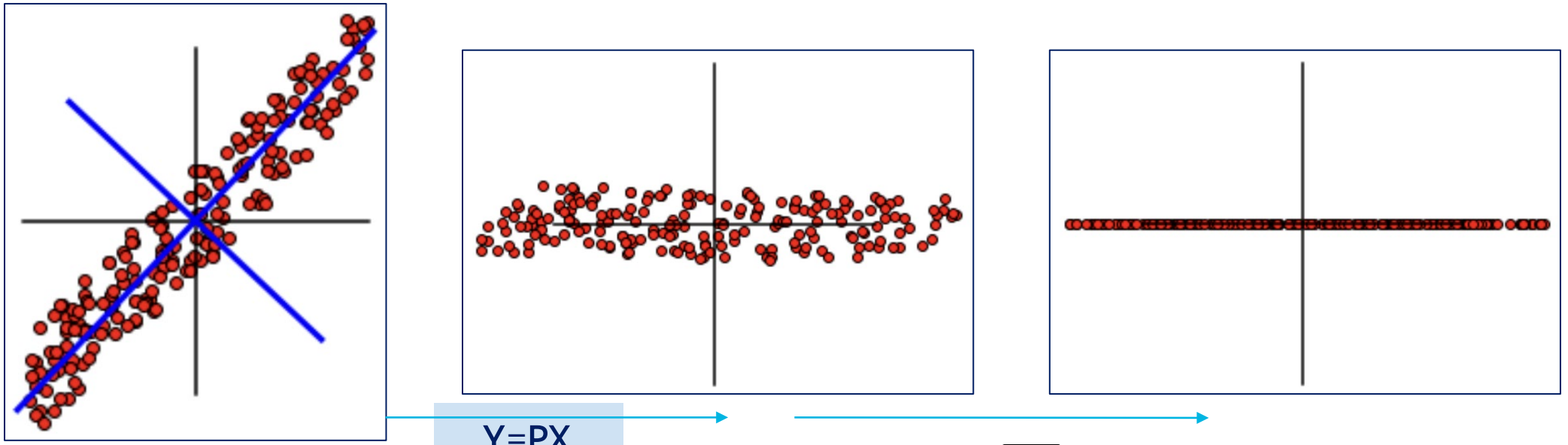
From the Spectral Theorem we conclude that rows of $\mathbf{P}$ must be the eigenvectors of $\mathbf{XX}^{\mathbf{T}}$ . Moreover, the diagonal elements in $\mathbf{D}$ are the eigenvalues of $\mathbf{XX}^{\mathbf{T}}$ .In other words, the eigenvalues of $\mathbf{XX}^{\mathbf{T}}$ are the variances of each coordinate.

# Principal Component Analysis (PCA)

- For covariance matrix:
  - Eigenvectors give the PCA components and eigenvalues give the explained variances of the components.
  - Eigenvectors can be sorted by eigenvalues in descending order to provide a ranking of the components.
  - Eigenvalues close to zero: components that can be discarded.

# Principal Component Analysis (PCA)

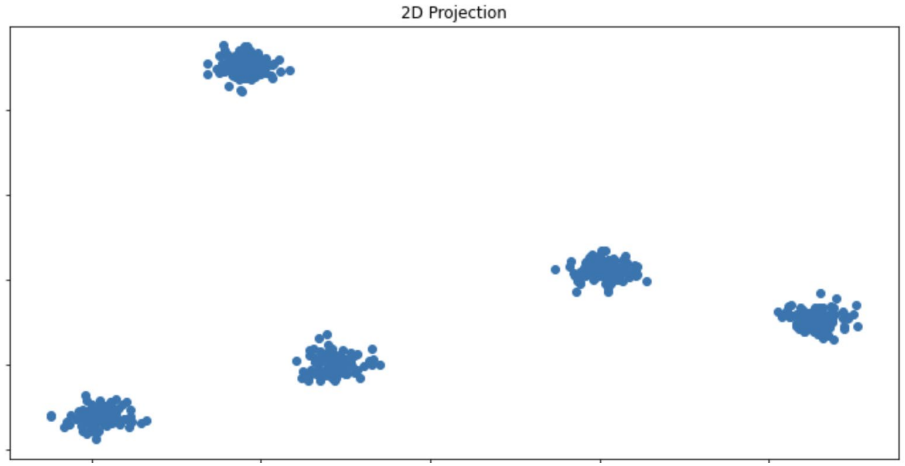
- Typically, the variance associated to certain principal components are close to zero, meaning that the coordinate associated to those directions corresponds to noise and can be disregarded.




PCA minimize: $\sum \|X_i - Y_i\|^2$

# Principal Component Analysis (PCA)

```
from sklearn.decomposition import PCA

data_pca=PCA(svd_solver='full')    # svd_solver='full' computes all the principal directions
data_transformed = data_pca.fit(X) # Computes the PC basis

var = data_transformed.explained_variance_
expl_var = data_transformed.explained_variance_ratio_

Y = np.dot(data_transformed.components_[:2,:],X.T) # projects original data onto the 2 main PC
```

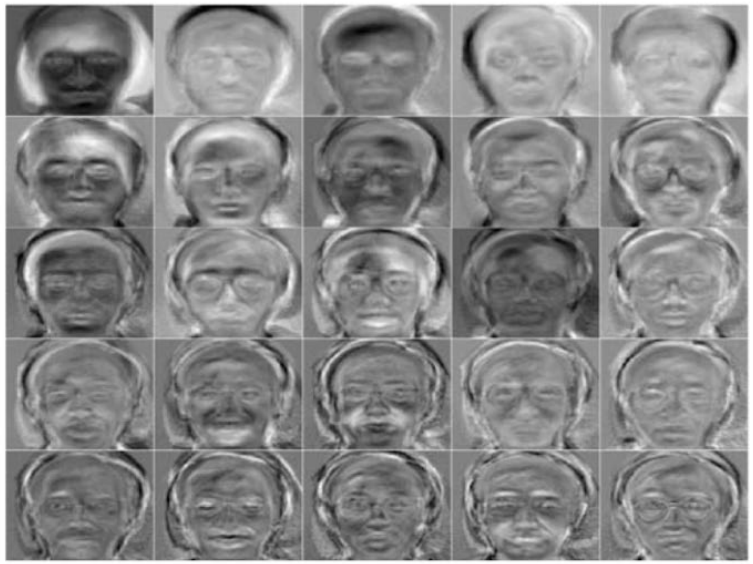


# Principal Component Analysis (PCA)

## Eigenfaces [Turk, Pentland '91]

- Input images:

- Principal components:



# Multidimensional Scaling (MDS)

- Given the pairwise distance between data points

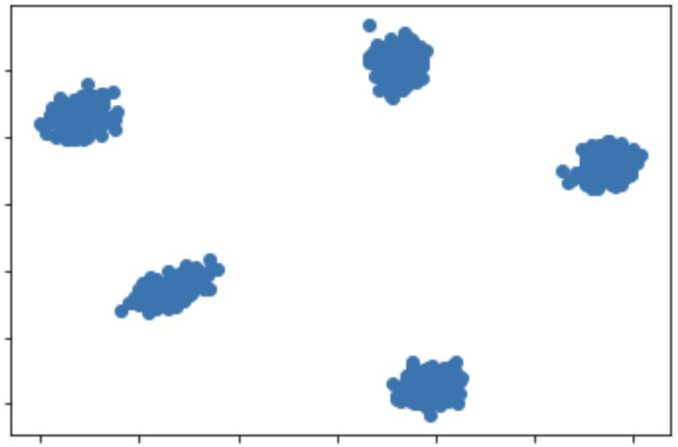
$$\begin{bmatrix} d_{12} & d_{13} & d_{14} & \cdots & d_{1n} \\ & d_{23} & d_{24} & \cdots & d_{2n} \\ & & & \ddots & \\ & & & & d_{(n-1)n} \end{bmatrix}$$

- Find an Euclidian embedding with total minimal distortion:

$$\min_{Y_i \in \mathbb{R}^k} \sum_{i,j} (\|Y_i - Y_j\|^2 - d_{ij}^2)^2$$

# Multidimensional Scaling (MDS)

```
from sklearn.manifold import MDS
from sklearn.metrics.pairwise import euclidean_distances

D = euclidean_distances(X)  # pairwise distance matrix

embedding = MDS(n_components=2,dissimilarity='precomputed') # 2D embedding from D
Y = embedding.fit_transform(D) # computes the embedding
```

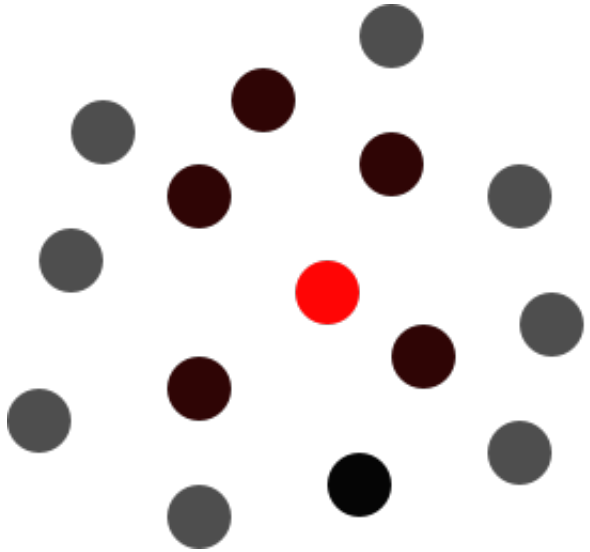


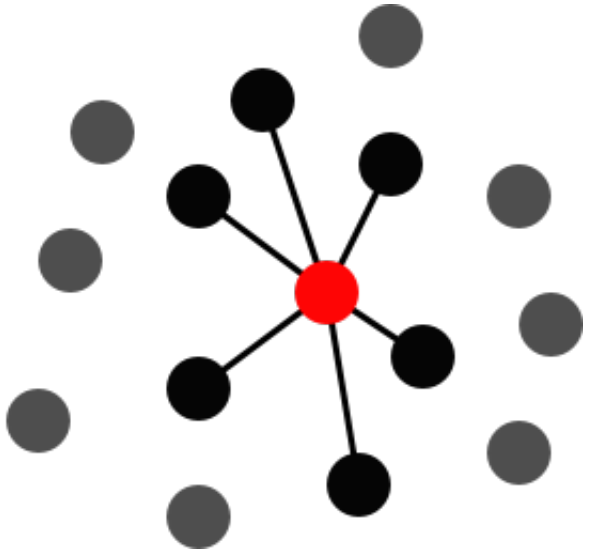
# tSNE

- tSNE: tool for dimensionality reduction of high-dimensional data.
- Main idea:
  - Similarity between data points in the high-dimensional space treated as distribution P.
  - Similarity between data points in the low-dimensional space Q.
  - Achieve a representation (embedding) in the low dimension where Q faithfully represents P.
- Steps:
  1. Computation of similarities ($O(n^2)$).
  2. Minimization of cost function (divergence between P and Q).

# tSNE

- The core idea of Stochastic Neighbor Embedding (SNE) is to convert the pairwise Euclidean distances between data points into conditional probabilities that represent similarities:





# tSNE

- The core idea of Stochastic Neighbor Embedding (SNE) is to convert the pairwise Euclidean distances between data points into conditional probabilities that represent similarities:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \qquad q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

# tSNE

- The core idea of Stochastic Neighbor Embedding (SNE) is to convert the pairwise Euclidean distances between data points into conditional probabilities that represent similarities:

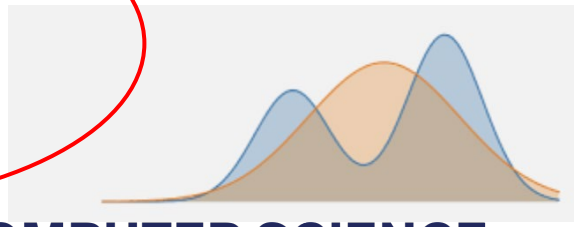
$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

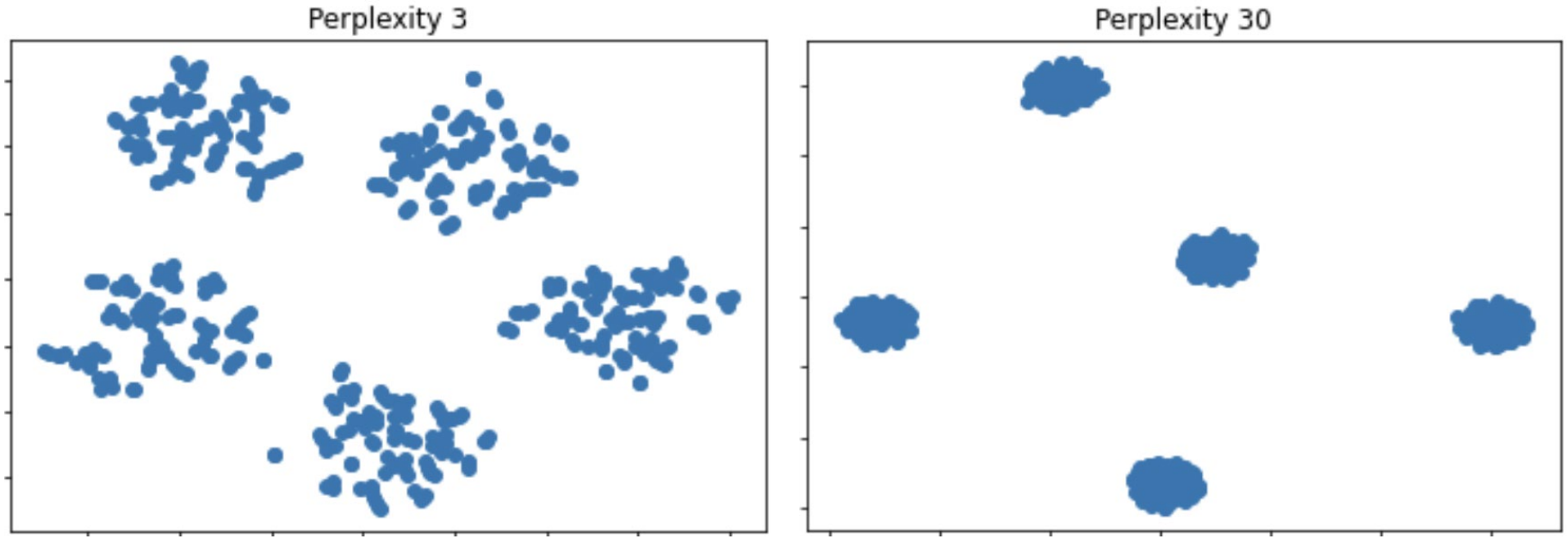
KL-divergence

$$\arg\min_{y_i \in \mathbb{R}^k} \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$



# tSNE

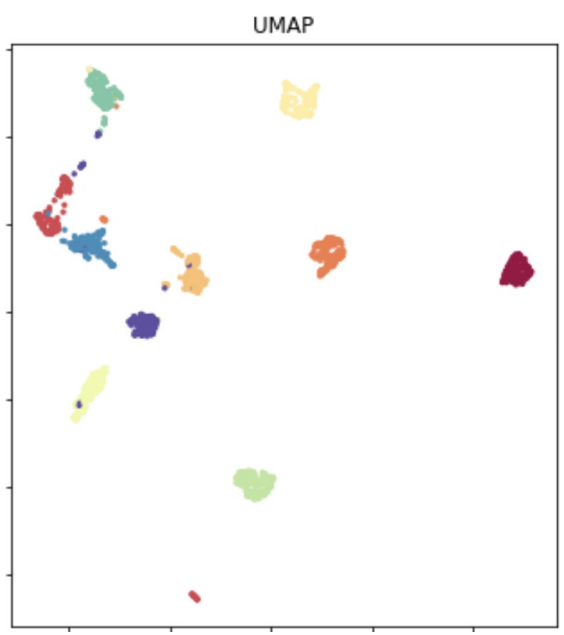
```
from sklearn.manifold import TSNE

t_sne = TSNE(n_components=2, perplexity=3)
Y = t_sne.fit_transform(X)
```

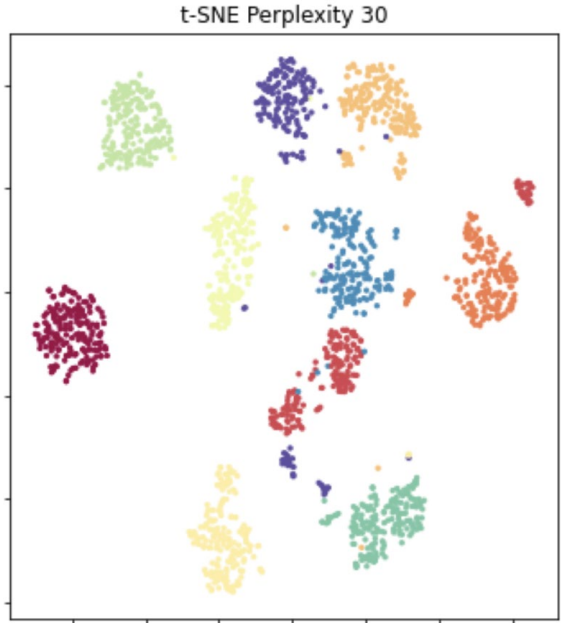


# UMAP

```
conda install -c conda-forge umap-learn
```

```
import umap

umap_r = umap.UMAP()
Y = umap_r.fit_transform(X)
```



# What about interactivity?

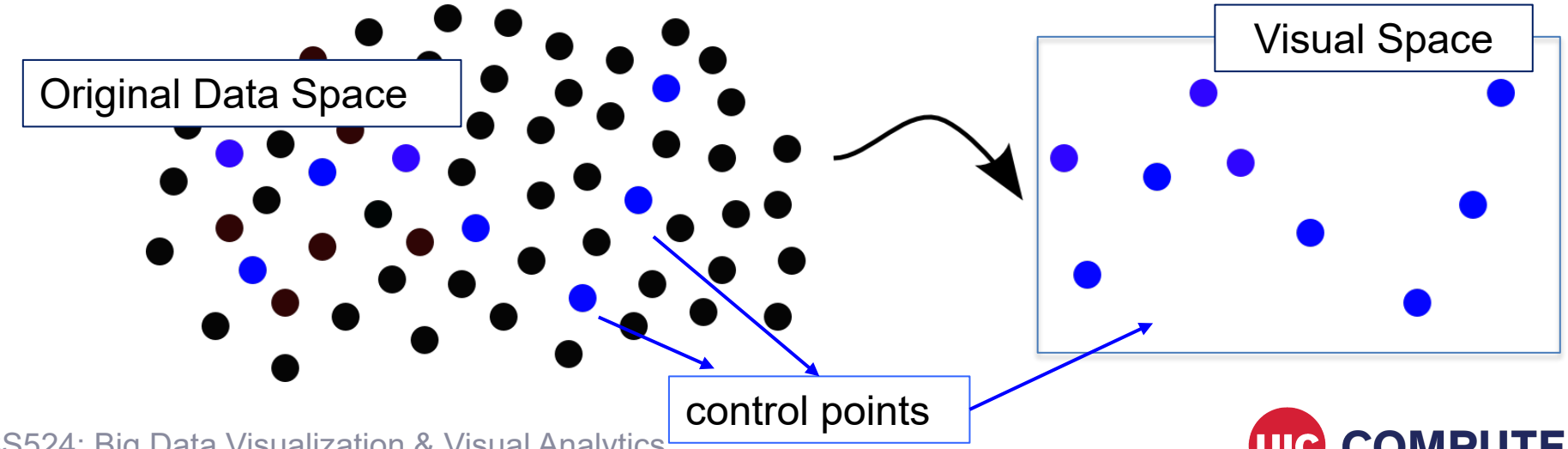
- Dimensionality reduction methods discussed previously are unsupervised, so users can only tune hyperparameters.
- **What if users want to have some control where certain points must be in the visual space while preserving their neighborhood?**

# What about interactivity?

- Dimensionality reduction methods discussed previously are unsupervised, so users can only tune hyperparameters.
- **What if users want to have some control where certain points must be in the visual space while preserving their neighborhood?**
- Interactive dimensionality reduction methods are designed to enable users with some control over the mapping.

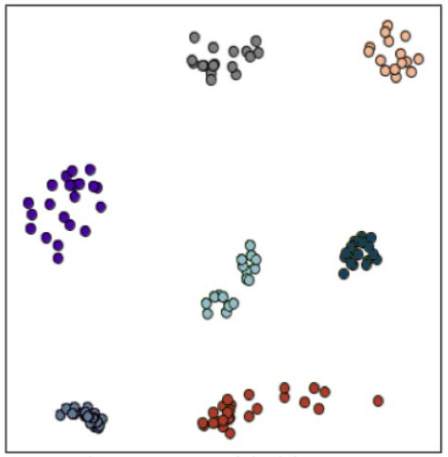
# Interactive dimensionality reduction

- There are several interactive dimensionality reduction methods:
  - Least Squares Projection (LSP).
  - Piecewise Laplacian Projection (PLP)
  - Local Affine Multidimensional Projection (LAMP)

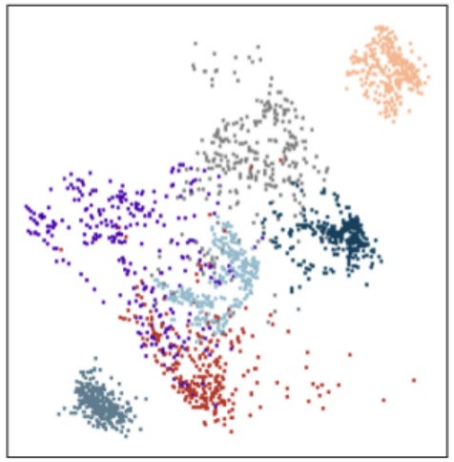


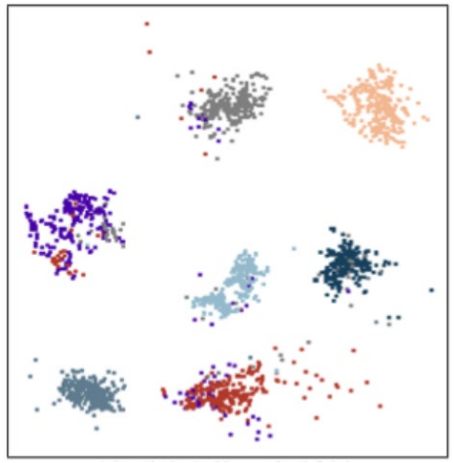
# LAMP

Control points



Tuning the hyperparameter $\sigma_i$



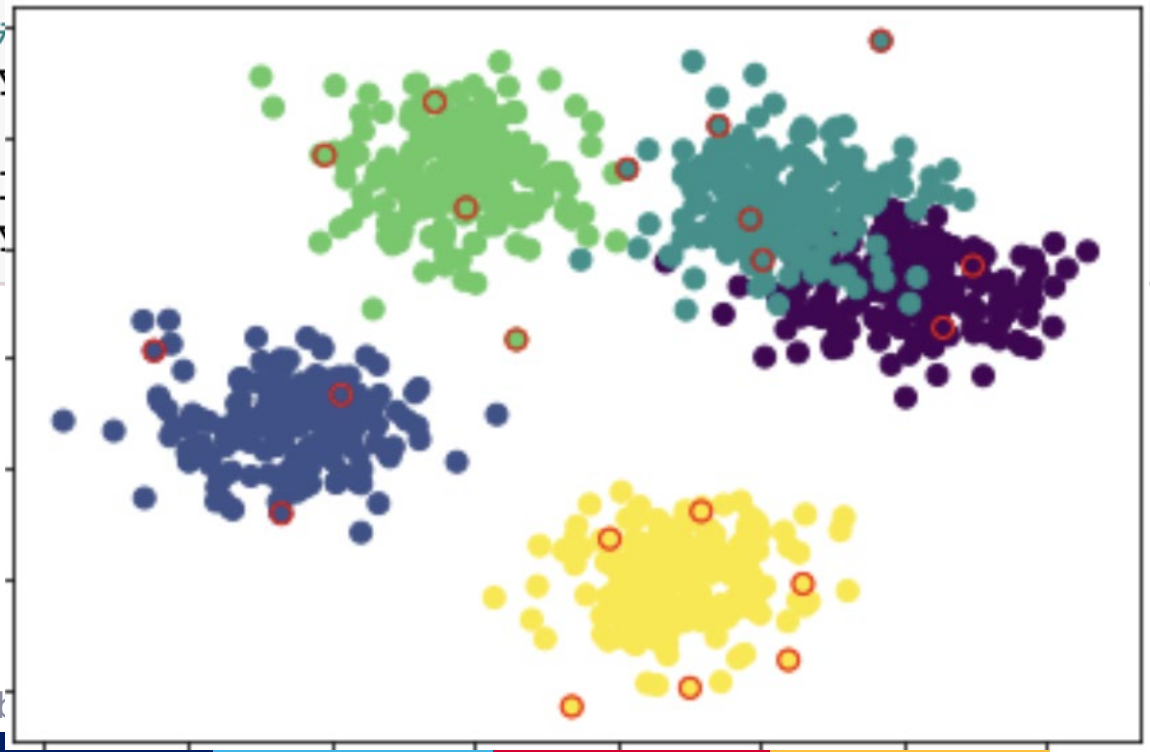


# LAMP

```
from lamp import Lamp # https://github.com/lgnonato/LAMP

# choose the control points from X
control_points_id = np.random.randint(0, high=X.shape[0], size=20)

# set the coordinates of the control points in 2D
Yp = mds.fit_transform(X[control_points_id,0:-1]) # here we are using mds for placing
                                                  # control points in 2D
# including control points ids into Yp
Yp = np.hstack((Yp,control_points_id.reshape(20,1)))

lamp_proj = Lamp(Xdata = X, control_points = Yp, label=True, scale=True)
Y = lamp_proj.fit()
```

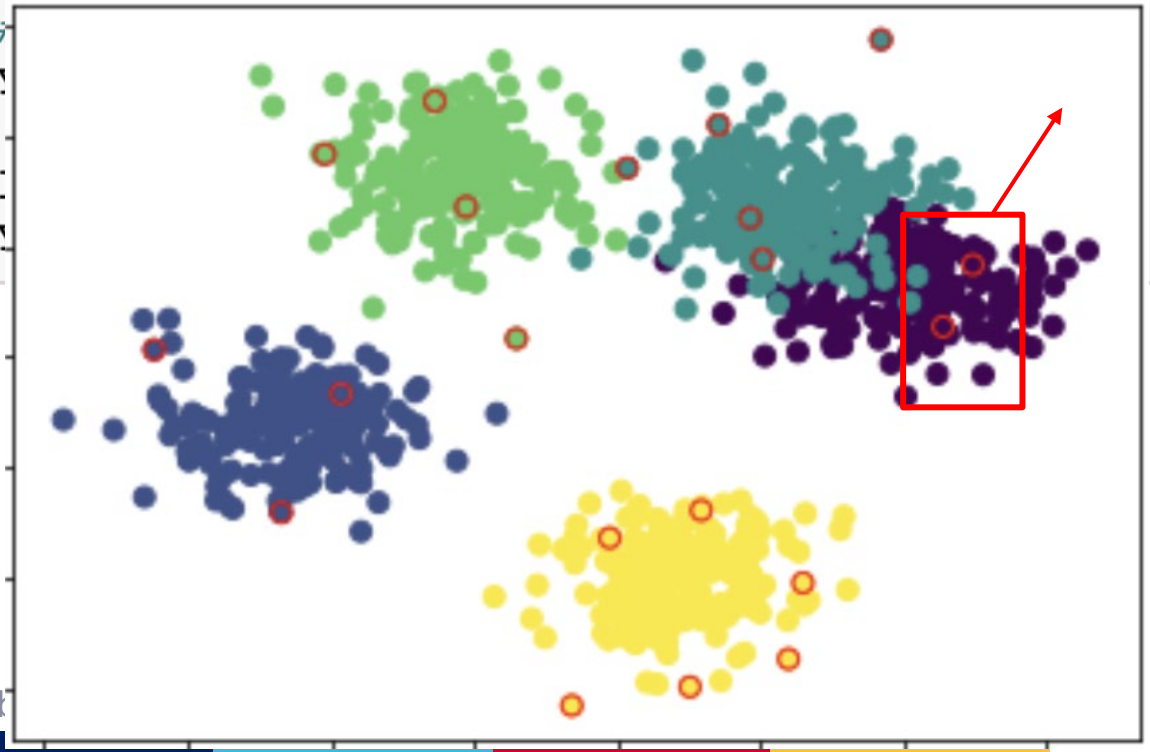
# LAMP

```
from lamp import Lamp # https://github.com/lgnonato/LAMP

# choose the control points from X
control_points_id = np.random.randint(0, high=X.shape[0], size=20)

# set the coordinates of the control points in 2D
Yp = mds.fit_transform(X[control_points_id,0:-1]) # here we are using mds for placing
                                                  # control points in 2D

20,1)))

Yp, label=True, scale=True)
```

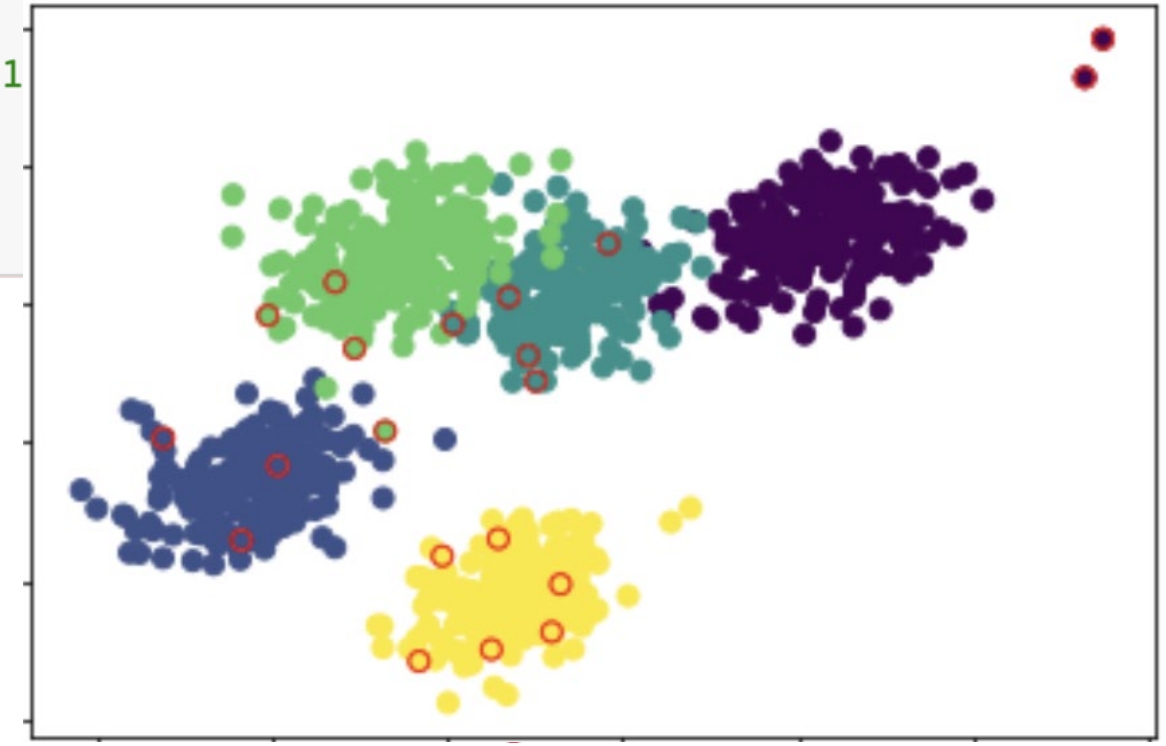


# LAMP

```
from lamp import Lamp # https://github.com/lgnonato/LAMP

# choose the control points from X
control_points_id = np.random.randint(0, high=X.shape[0], size=20)

# set the coordinates of the control points in 2D
Yp = mds.fit_transform(X[control_points_id,0:-1]) # here we are using mds for placing
                                                  # control points in 2D
```

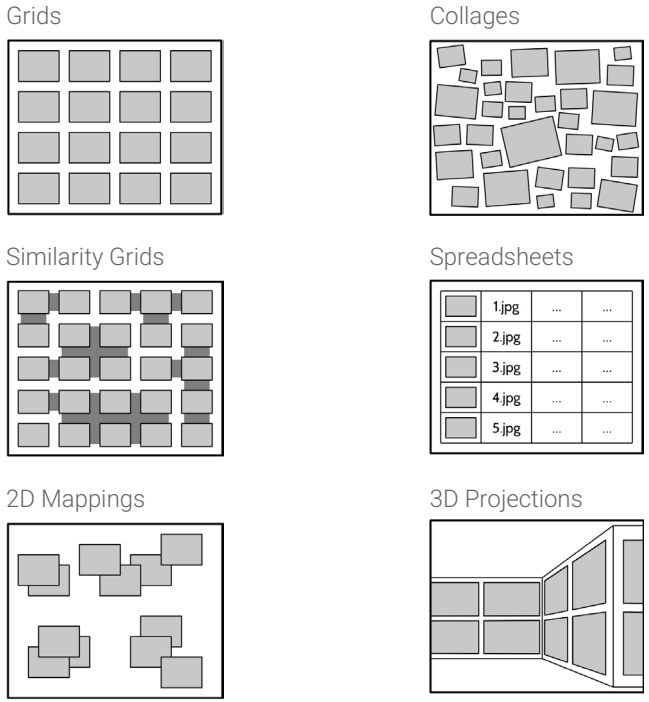


```
20,1

Yp,
```



# Urban data exploration

Image data



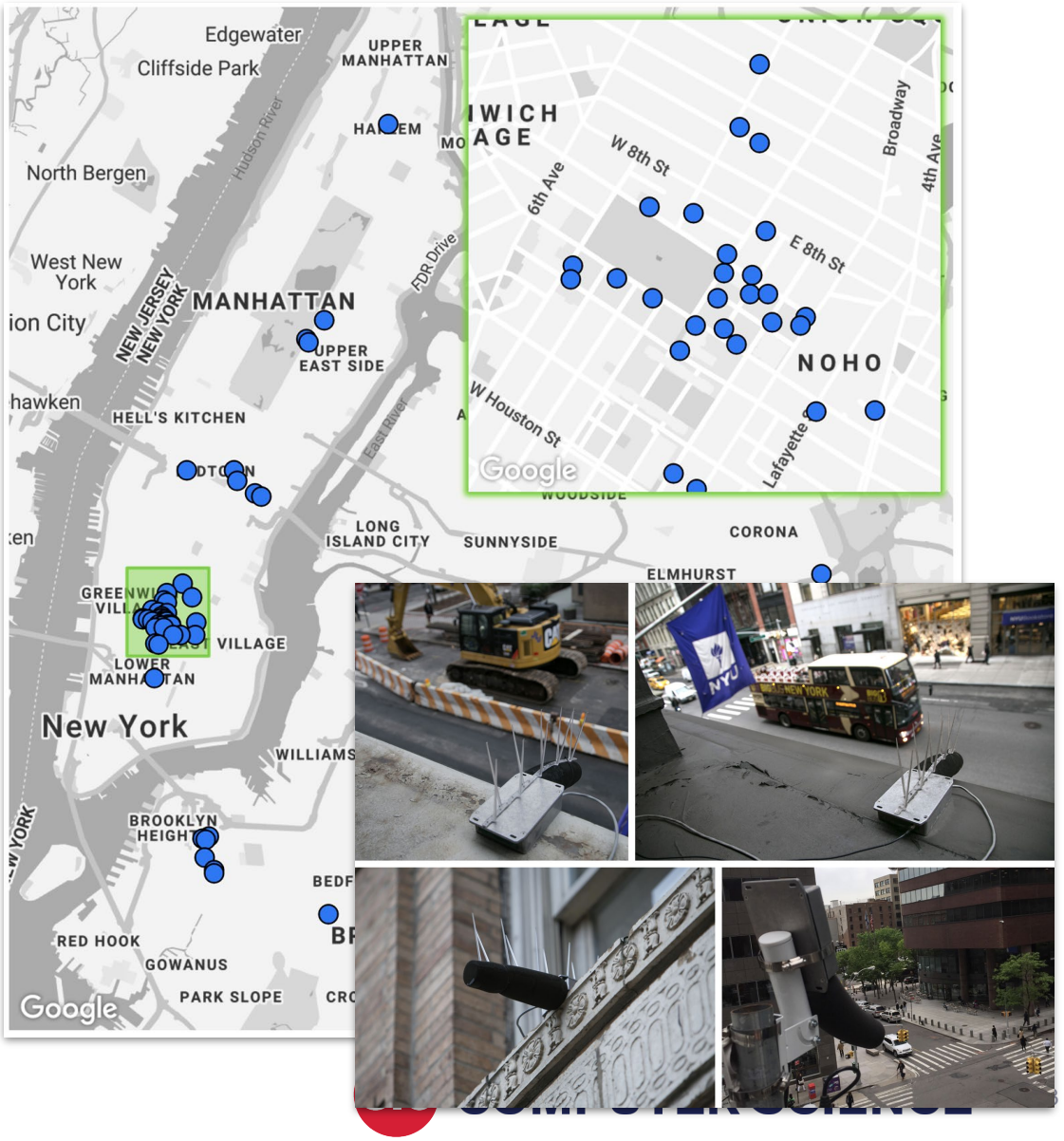

Natural pictorial representation of image data.

Humans are able to visualize and understand sets of images in a ***parallel approach.***

Organization approaches for images were proposed in the past. These approaches try to optimize the observation of specific **patterns present in collections of images.**

# Deployment

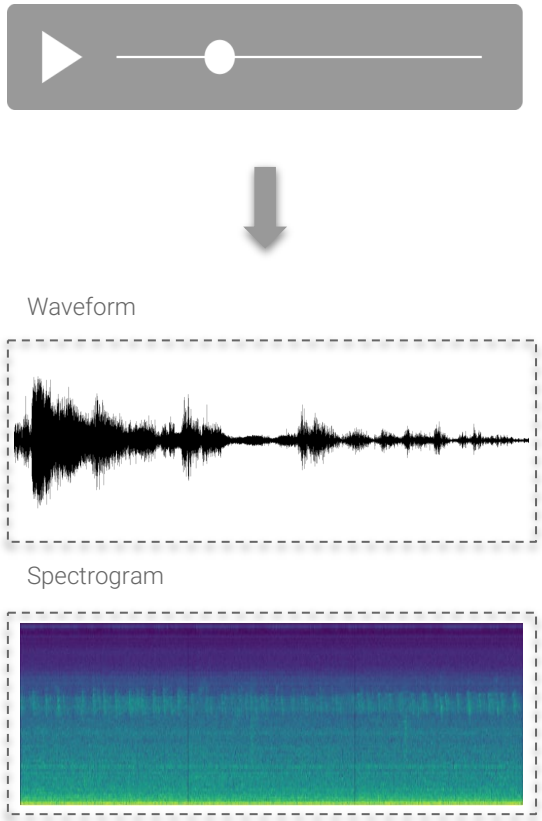
- **55 sensors** deployed over 5 years.
- **150 sensor years** of decibel data.
- **75 years** of audio data.
- **200M audio recordings** which accounts for **75 Terabytes of data.**



# Urban audio data exploration

Audio data

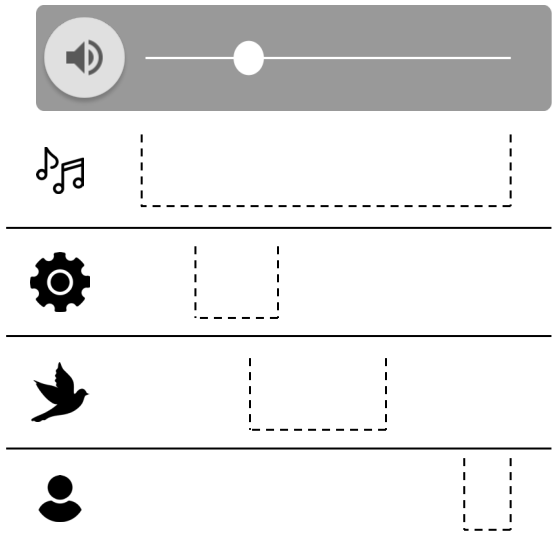
- Audio recordings are consumed in a **serial** way by us.
- To understand events happening in a 10-second audio snippet, users **must listen through the entire recording.**
- Although the **visualization of specific frequencies or loudness** can help identify interest periods of the recording, it is still difficult to build a **semantic understanding** of the recording.



# Urban audio data exploration

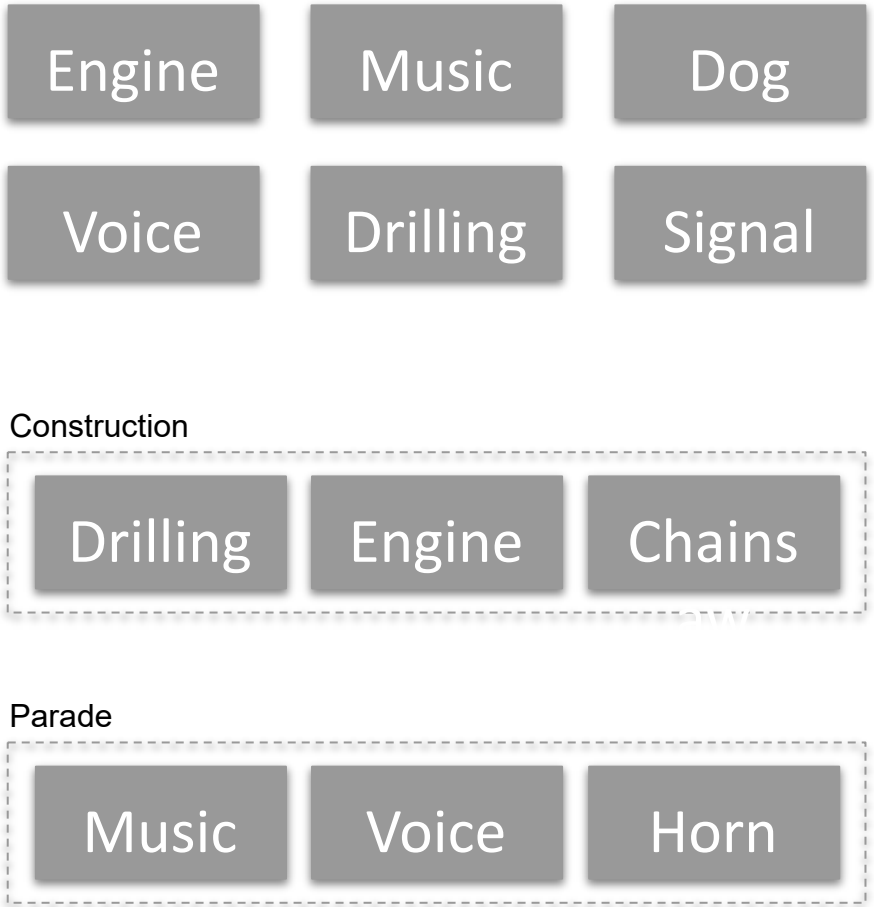
Audio data

- Unlike images, where visual objects are opaque, sound objects are conceptually ***transparent***, meaning that multiple objects (sound sources) can have energy at the same frequency.
- At any given instant in time, a sound recording might have a **mixture of background** (birds, dog barks) and **foreground sounds** (party, sirens).



# Sound labeling and classification

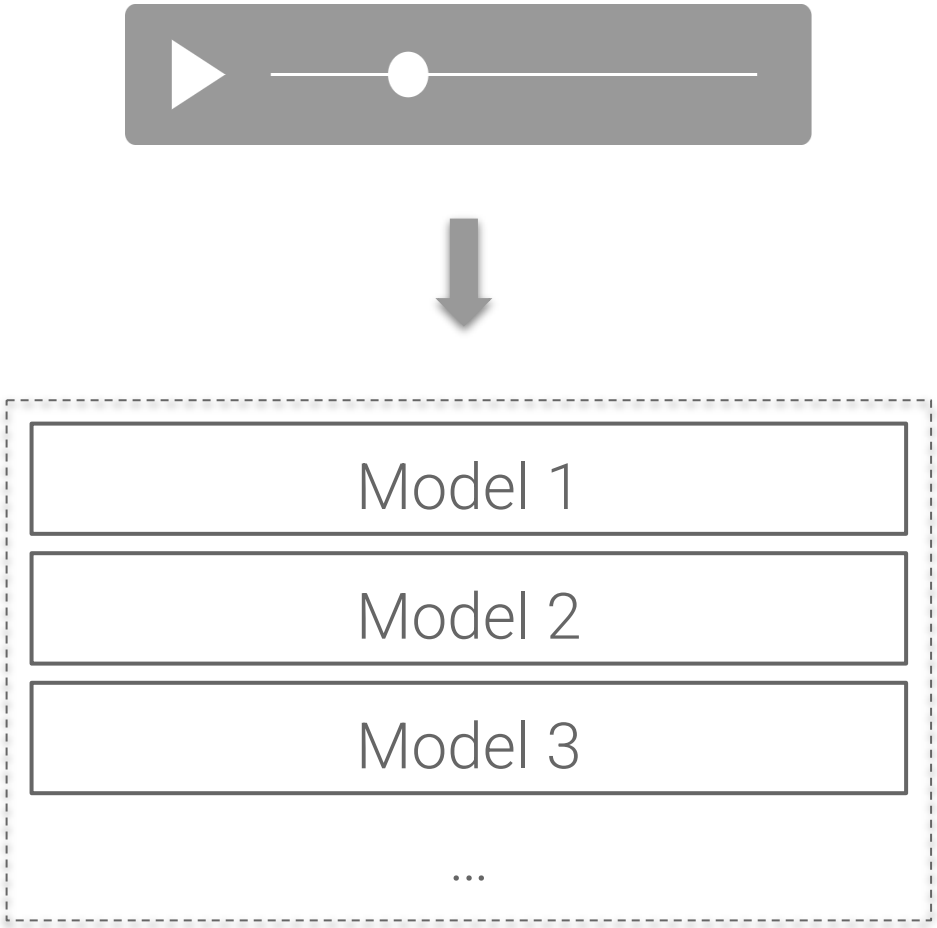
- Previously proposed classifiers provide a reasonable link between embeddings and human-understandable vocabulary.
- However, their **class vocabularies are limited**, providing a **narrow view of the rich and varied soundscape** of the city.
- Also, they do not account for classes that are a composition of previously created classes.



# Sound representation

The **scarcity of labeled urban audio data** makes it hard to generate models capable of transforming audio into representations that can represent different audio classes.

Also, the **complexity of the urban soundscape** makes it even harder for models to be representative of such a dynamic environment.



# Urban Rhapsody: Requirements

[R1] **Interactive Identification and Labeling of Similar Sounds**: Given the highly complex acoustic environment we observe in cities, audio representations cannot clearly encode specific audio events that users might be interested in.

[R2] **Projection Steering Based on User Perception**: When exploring audio embeddings extracted from urban recordings through multidimensional projections, we often recognize clusters that do not represent the user's perception of the soundscape.

[R3] **Iterative Creation of Classification Models**: Considering that current machine listening models present certain limitations, the system should provide the capability to iteratively create new classification models based on the data points labeled by the user (and, therefore, the user's perception of the soundscape).

*Urban Rhapsody: Large-scale Visual Exploration of Urban Soundscapes, EuroVis 2022*

# Urban Rhapsody: Requirements

[R4] **Local and Global Sound Perspectives**: Audio embeddings might possess certain characteristics that only become clear when they are analyzed locally or globally.

[R5] **Match between Audio and Visual Representations**: Visualizing audio files in the frequency domain is important for the user when assessing the accuracy of both the embeddings and classifications.

[R6] **Support Interactive Query Times**: The system should support interactive queries to enable the easy and quick labeling of data points and the creation of classification models.

*Urban Rhapsody: Large-scale Visual Exploration of Urban Soundscapes, EuroVis 2022*

# Data size

- SONYC has captured more than **70TB** of audio recordings throughout **5 years.**
- As past work studied, interactive systems must provide reasonably **low response times** for querying datasets.
- One limiting factor for domain experts, usually professionals with no computer science background, is to generate useful insights from a large and complex dataset such as the one produced by SONYC.

The Effects of Interactive Latency on Exploratory Visual Analysis

Zhicheng Liu and Jeffrey Heer

**Abstract**—To support effective exploration, it is often stated that interactive visualizations should provide rapid response times. However, the effects of interactive latency on the process and outcomes of exploratory visual analysis have not been systematically studied. We present an experiment measuring user behavior and knowledge discovery with interactive visualizations under varying latency conditions. We observe that an additional delay of 500ms incurs significant costs, decreasing user activity and data set coverage. Analyzing verbal data from think-aloud protocols, we find that increased latency reduces the rate at which users make observations, draw generalizations and generate hypotheses. Moreover, we note interaction effects in which initial exposure to higher latencies leads to subsequently reduced performance in a low-latency setting. Overall, increased latency causes users to shift exploration strategy, in turn affecting performance. We discuss how these results can inform the design of interactive analysis tools.

**Index Terms**—Interaction, latency, exploratory analysis, interactive visualization, scalability, user performance, verbal analysis

## 1 Introduction

One stated goal of interactive visualization is to enable data analysis at "rates resonant with the pace of human thought" [19, 20]. This goal entails two research directions: understanding the rate of cognitive activities in the context of visualization, and supporting these cognitive processes through appropriately designed and performant systems.

Latency is a central issue underlying these research problems. Due to the time required for query processing, data transfer, and rendering, data-intensive visualization systems incur delay. It is generally held that low latency leads to improved usability and better user experience. Unsurprisingly, multiple research efforts focus on reducing query and rendering latency for large datasets, which may include billions or more data points. Latencies in state-of-the-art systems can range from 20 milliseconds up to multiple seconds for a unit task [2, 28, 29].

Despite the shared goal of minimizing latency, the effects of interaction delays on user behavior and knowledge discovery with visualizations remain largely unevaluated. While previous research on the effects of interactive latency in puzzle solving [4, 17, 35, 36] and search [8] has shown that user behavior changes in response to millisecond-scale differences in latency, studies in other domains such as computer games report no significant effects [23, 39].

It is unclear to what degree these findings apply to exploratory visual analysis. Unlike problem-solving tasks or most computer games, exploratory visual analysis is open-ended and does not have a clear goal state. User interaction may be triggered by salient visual cues in the display, driven by *a priori* hypotheses, or carried out through exploratory browsing. The process is more spontaneous and is unconstrained by factors such as game rules.

How does latency affect user behavior and knowledge discovery in exploratory visual analysis? To answer this question, we conduct controlled experiments comparing two latency conditions, differing by 500ms per operation. We analyze data collected from both system logs and think-aloud protocols to test if (a) delay impacts interaction strategies and (b) lower latency leads to better analysis performance.

Our work makes the following contributions. First, we present the design and the results of a controlled study confirming that a 500ms difference can have significant impacts on visual analysis. Specifically, we find that (1) the additional delay results in reduced interaction and reduced dataset coverage during analysis; (2) the rate at which users make observations, draw generalizations and generate hypotheses (as determined using a think-aloud protocol) also declines due to the delay; and (3) initial exposure to delays can negatively impact overall performance even when the delay is removed in a later session. Second, we extend the insight-based evaluation methodology [37, 38] for comparative analysis of qualitative data regarding visualization use. We introduce a procedure for segmenting, coding and analyzing think-aloud protocols for visualization research. Our analysis contributes coding categories that are potentially applicable for future protocol analysis. Finally, our results show that the same delay has varying influences on different interactive operations. We discuss some implications of these findings for system design.

## 2 Related Work

Our research draws on related work in scalable visualization systems, cognitive science and domain-specific investigations on the effects of interactive latency. We review relevant literature below.

### 2.1 Scalable Data Analysis Systems

Building low latency analysis systems has been a focus for many research projects and commercial systems, spanning both back-end and front-end engineering efforts. Spark [44, 45] supports fast in-memory cluster computing through read-only distributed datasets for machine learning tasks and interactive ad-hoc queries. Nanocubes [28] contribute a method to store and query multi-dimensional aggregated data at multiple levels of resolution in memory for visualization. Profiler [26] builds in-memory data cubes for query processing. Tableau's data engine [1] optimizes both in-memory stores and live connections to databases on disk. imMens [29] decomposes multi-dimensional data cubes into binned data tiles of reduced dimensionality and performs accelerated query processing and rendering on the GPU.

In cases where long-running queries are unavoidable, sampling and online aggregation [22] are often used to improve user experience. BlinkDB [2] builds multi-dimensional, multi-resolution samples and dynamically estimates a query's response time and error. With online aggregation [22], visualizations of estimated results are incrementally updated as a query progresses. Studies suggest that data analysts can interpret approximate results visualized as bar charts with error bars to make confident decisions [16].

### 2.2 Time Scales of Human Cognition

Decades of psychology research have produced evidence that different thought processes operate at varying speeds [25]. Newell [33] provides a framework outlining proposed time scales of human cognition. Relevant to studies of human-computer interaction are the cognitive (100 milliseconds to 10 seconds) and rational (minutes to hours) time bands. Within the cognitive band, Newell identifies three types of time constants: deliberate act, operation, and unit task. Card et al. [11] make similar distinctions using a different terminology. Table 1 summarizes these scales, exemplary actions, and the time ranges during which these actions occur.

• *Zhicheng Liu is with Adobe Research. E-mail: leoli@adobe.com.*
• *Jeffrey Heer is with the University of Washington. E-mail: jheer@uw.edu.*

*Manuscript received 31 March 2014; accepted 1 August 2014; posted online 13 October 2014; mailed on 4 October 2014.*
*For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.*

# Similarity search

- Urban Rhapsody allows for two different query approaches: **by example** and **by concept** .
- When querying by example, the user can pick an audio frame from predefined list of audio recordings in the interface.
- The user also must define the number of similar points to be retrieved.

**[R1] Interactive identification** and labeling **of similar concepts**

# Similarity search

- Urban Rhapsody allows users to search for similar audio events captured by a sensor in the space of one year.
- Our definition of similarity is based on the euclidean distance between two vectors, where each vector represents 1 second of audio.
- Not feasible to calculate this for 1 year of recordings for a given sensor in interactive time.
- To allow for interactive querying time we use an ANN approach.

ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms*

Martin Aumüller[1], Erik Bernhardsson[2], and Alexander Faithfull[1]

1 IT University of Copenhagen, Denmark, {maau,alef}@itu.dk
2 Better Inc., mail@erikbern.com



Abstract

This paper describes ANN-Benchmarks, a tool for evaluating the performance of in-memory approximate nearest neighbor algorithms. It provides a standard interface for measuring the performance and quality achieved by nearest neighbor algorithms on different standard data sets. It supports several different ways of integrating $k$-NN algorithms, and its configuration system automatically tests a range of parameter settings for each algorithm. Algorithms are compared with respect to many different (approximate) quality measures, and adding more is easy and fast; the included plotting front-ends can visualise these as images, LaTeX plots, and websites with interactive plots. ANN-Benchmarks aims to provide a constantly updated overview of the current state of the art of $k$-NN algorithms. In the short term, this overview allows users to choose the correct $k$-NN algorithm and parameters for their similarity search task; in the longer term, algorithm designers will be able to use this overview to test and refine automatic parameter tuning. The paper gives an overview of the system, evaluates the results of the benchmark, and points out directions for future work. Interestingly, very different approaches to $k$-NN search yield comparable quality-performance trade-offs. The system is available at http://ann-benchmarks.com.

1998 ACM Subject Classification H.3.3 Information Search and Retrieval

Keywords and phrases benchmarking, nearest neighbor search, evaluation

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Nearest neighbor search is one of the most fundamental tools in many areas of computer science, such as image recognition, machine learning, and computational linguistics. For example, one can use nearest neighbor search on image descriptors such as MNIST [25] to recognize handwritten digits, or one can find semantically similar phrases to a given phrase by applying the word2vec embedding [31] and finding nearest neighbors. The latter can, for example, be used to tag articles on a news website and recommend new articles to readers that have shown an interest in a certain topic. In some cases, a generic nearest neighbor search under a suitable distance or measure of similarity offers surprising quality improvements [9].

In many applications, the data points are described by high-dimensional vectors, usually ranging from 100 to 1000 dimensions. A phenomenon called the *curse of dimensionality*, a consequence of several popular algorithmic hardness conjectures (see [4, 38]), tells us that, to obtain the true nearest neighbors, we have to use either linear time (in the size of the dataset) or time/space that is

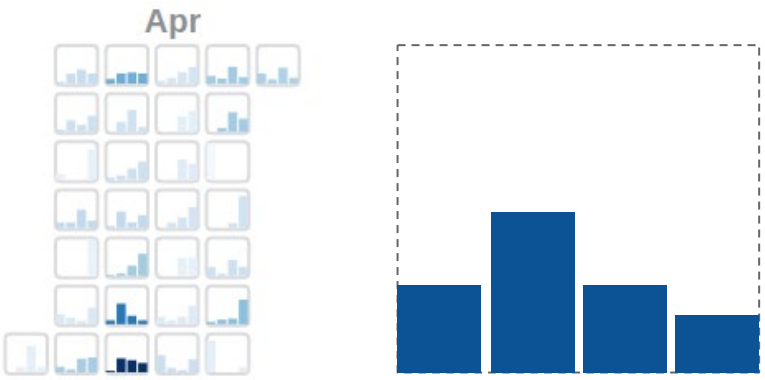
* The research of the first and third authors has received funding from the European Research Council under the European Union's 7th Framework Programme (FP7/2007-2013) / ERC grant agreement no. 614331. A conference version of this work was published at SISAP'17 and is available at http://dx.doi.org/10.1007/978-3-319-68474-1_3.



**[R1] Interactive identification** and labeling **of similar concepts**

# Event distribution

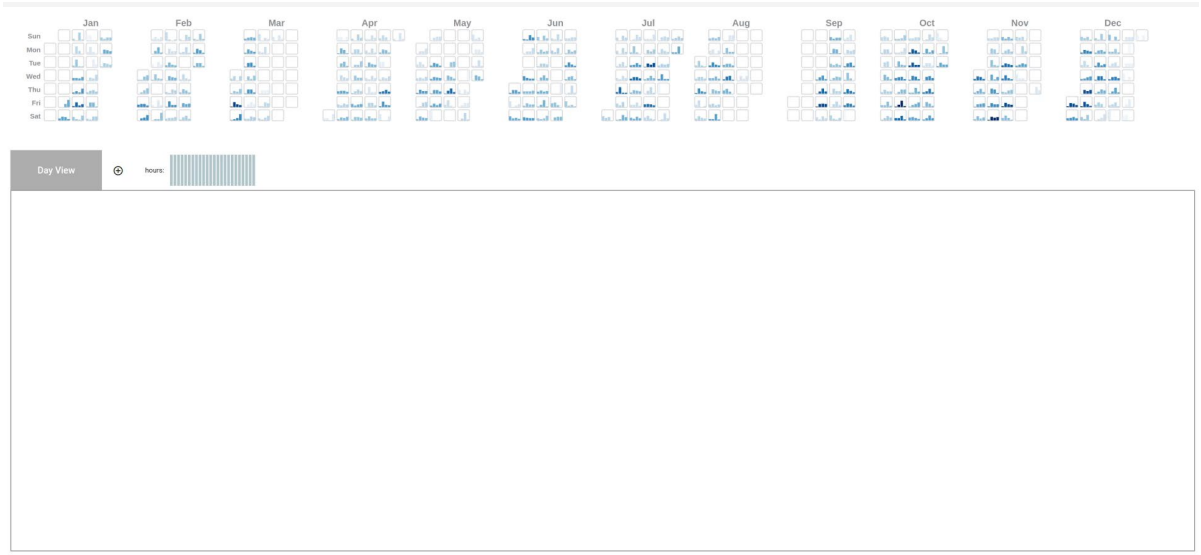

- Once the user runs a similarity query on Urban Rhapsody database using our query system, Urban Rhapsody provides feedback on the distribution of that specific event throughout a given year.The color scale encodes the density of the events across the year.
- The distribution plot shows the distribution of the event within days.
- The distribution plot shows the distribution of the event within days.




**[R4]** Local and **Global Audio Perspectives**

# Day view

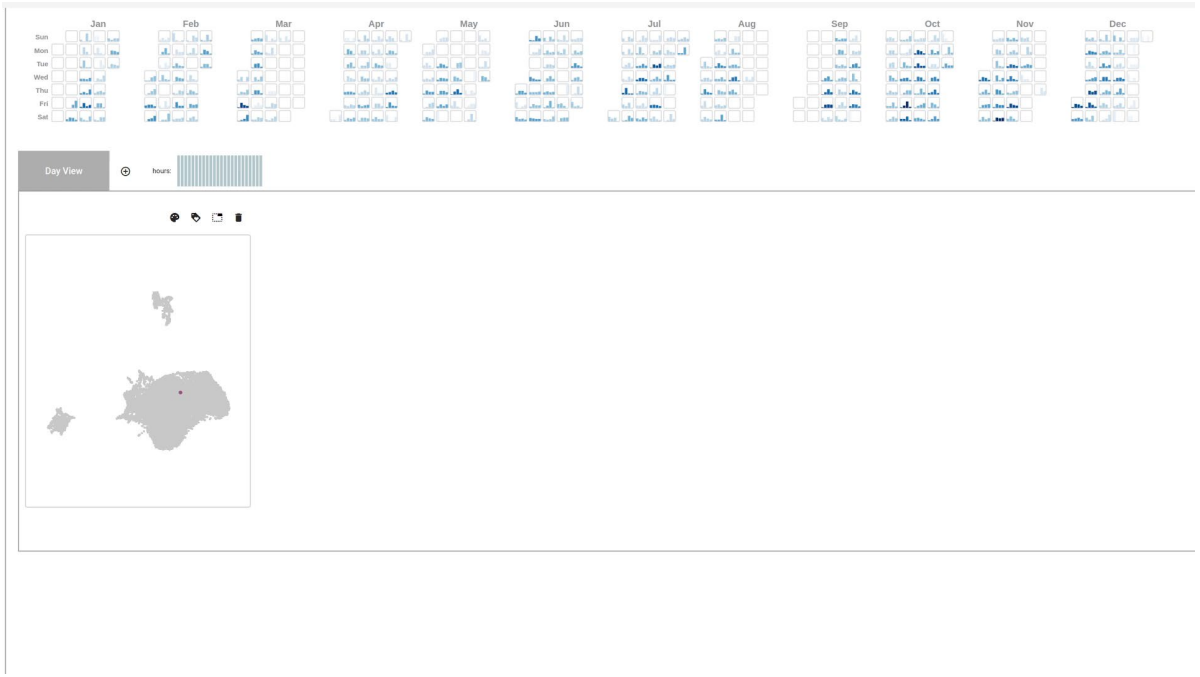
- Following Schneiderman's visualization-seek mantra: **Overview first. Zoom and Filter. Details on Demand.** Urban Rhapsody allows users to analyze specific days by loading the whole data available for a given day.
- To allow users to visualize and explore all the feature vectors for a given day, we use well-known projection techniques such as UMAP.




**[R4] Local** and Global **Audio Perspectives**

# Labeling

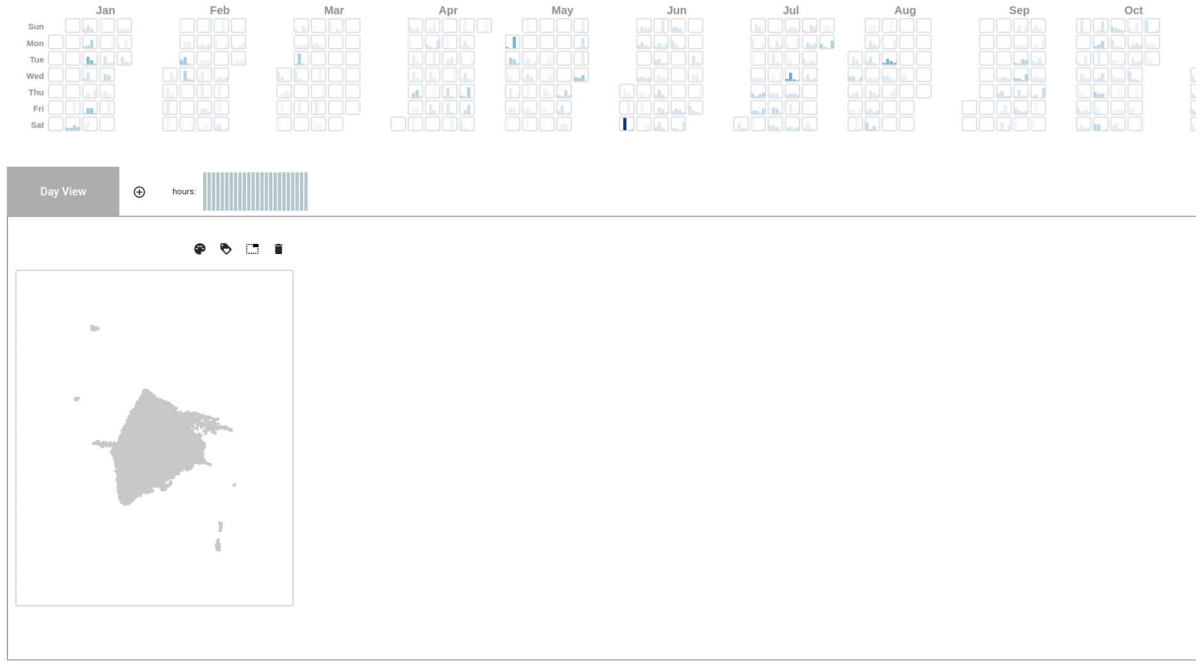
- Once the user loads a specific day of data, it is possible to select specific audio frames to listen to by brushing the scatter plot presented on the day view.
- At this point, the user is able to label the audio frame as part of a specific classes.
- The popup window allows users to write the class name they want to label the specific frame. Urban Rhapsody allows the user to label frames with **positive or negative classes.**




**[R1]** Interactive identification and **labeling of similar concepts**

# Projection steering

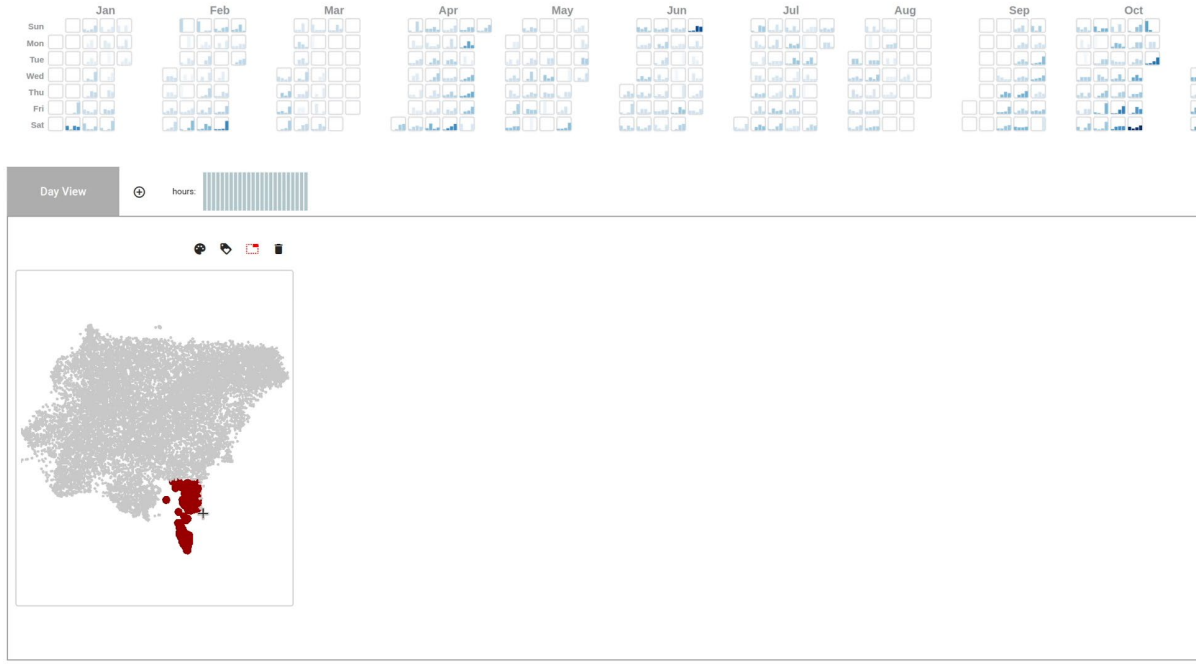
- Although the day view helps users zoom in on specific days by filtering points from the rest of the year, it is still hard to explore a whole day of audio recordings.
- While users select and listen to specific audio frames, they can generate new projections that take the acquired knowledge as input.
- Users can perform three operations: **steer, focus and remove.**




**[R2] Projection Steering based on user perspective**

# Projection steering

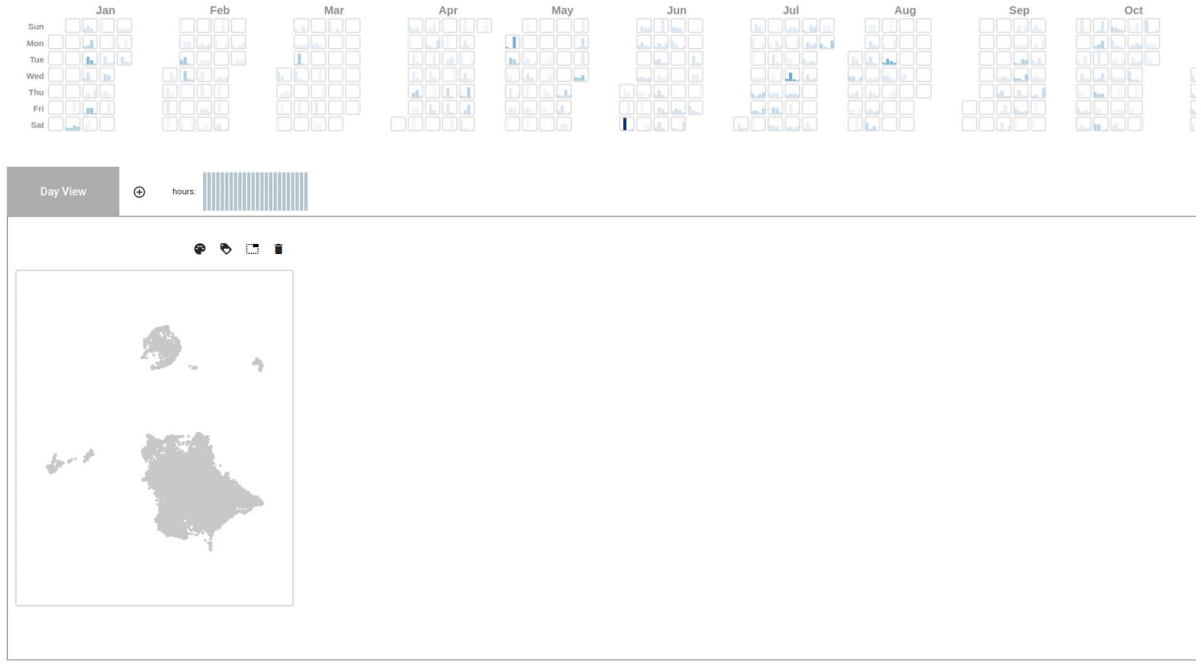
- Although the day view helps users zoom in on specific days by filtering points from the rest of the year, it is still hard to explore a whole day of audio recordings.
- While users select and listen to specific audio frames, they can generate new projections that take the acquired knowledge as input.
- Users can perform three operations: **steer, focus and remove.**




**[R2] Projection Steering based on user perspective**

# Projection steering

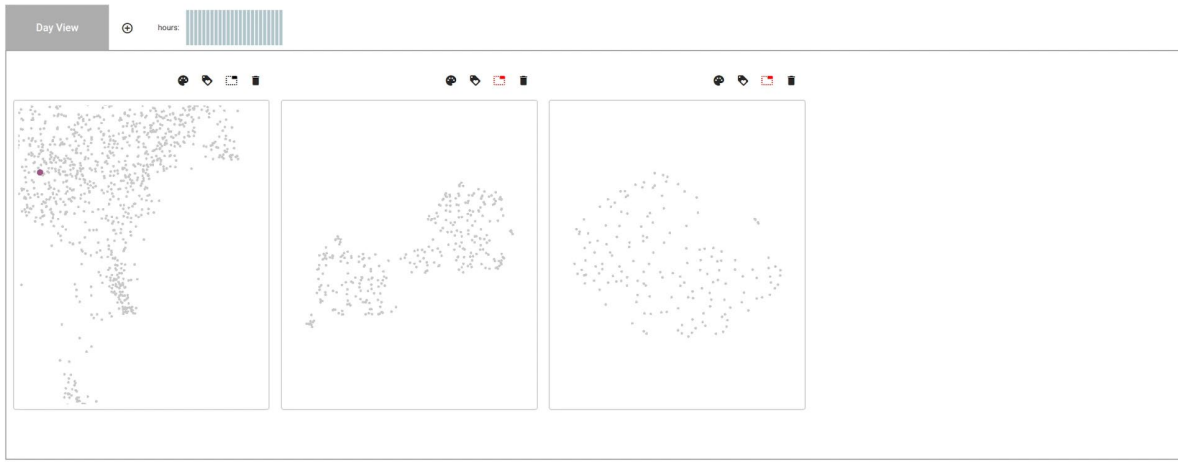
- Although the day view helps users zoom in on specific days by filtering points from the rest of the year, it is still hard to explore a whole day of audio recordings.
- While users select and listen to specific audio frames, they can generate new projections that take the acquired knowledge as input.
- Users can perform three operations: **steer, focus and remove.**




**[R2] Projection Steering based on user perspective**

# Focused view

- When a specific day is loaded, the user can interact with the projected scatterplots by selecting either individual points or subsets of the day through bounding boxes
- Once a selection is made, the user is able to see the spectrograms relative to the selection.
- When hovering the mouse over the spectrogram the user is able to listen to that specific second of audio.




**[R4] Local** and Global **Audio Perspectives**

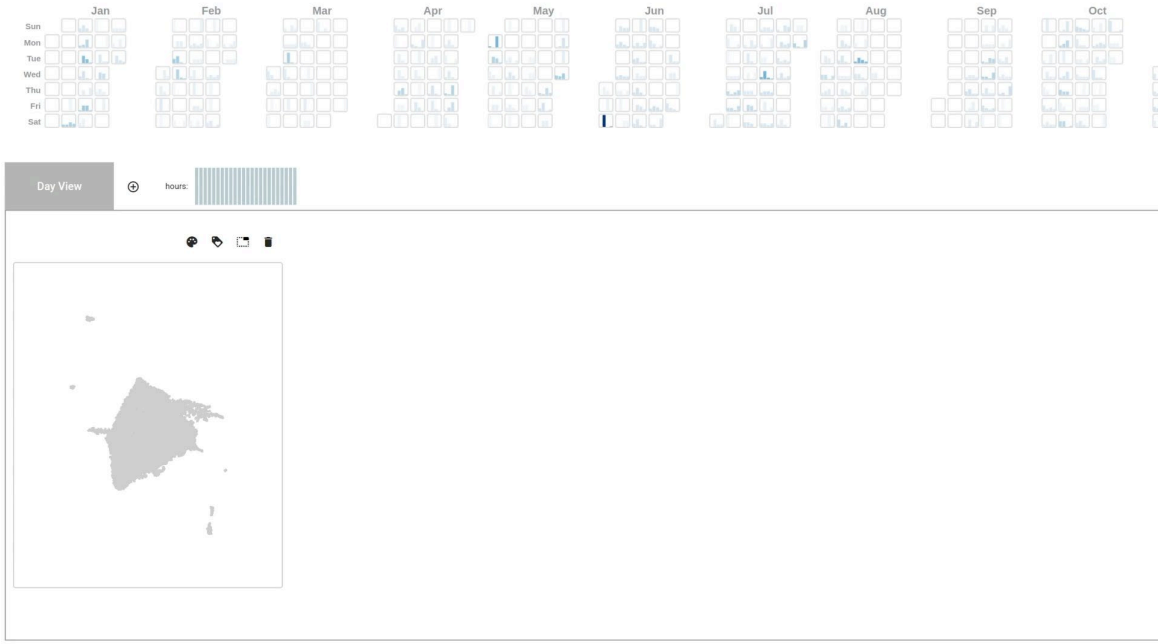
**[R2] Match between Audio and Visual Representations**

# Urban Rhapsody: Creating prototypes

Binary Classification Model

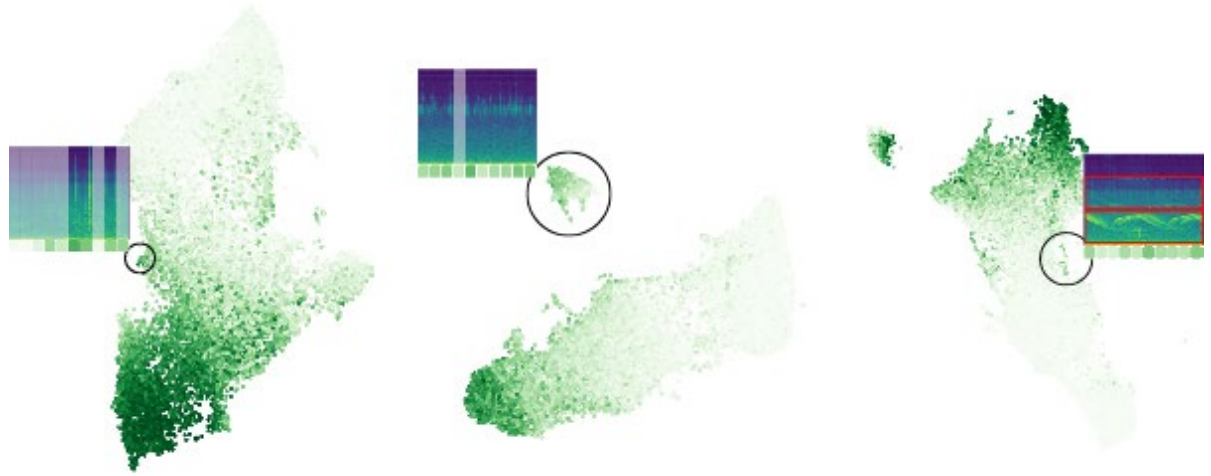
+

Representatives



# Model-aided exploration

- Once an initial model is created, the users are able to navigate through different days and using the model to speed their search for similar audio concepts.
- During the exploration the users can explore specific prediction ranges of the model and annotate more data points as either positive or negative labels.
- At any point in time, the users can refine their models.

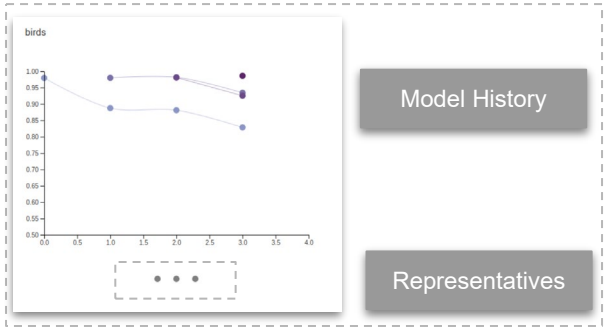


**[R3] Iterative Creation of Classification Models**

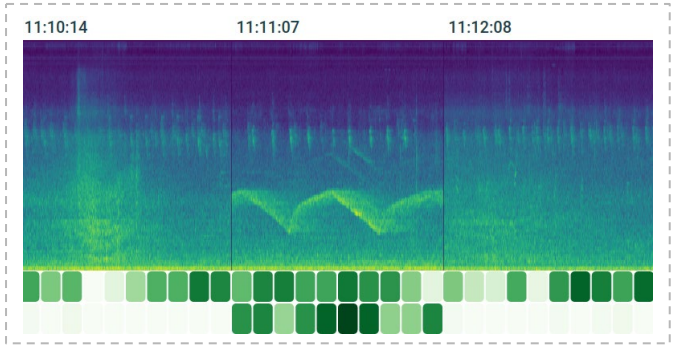
# Model assessment

- We can assess the model performance historically by comparing it with older models. At each new model refinement, we evaluate older models on the newest set of annotated data to assess model convergence.
- Also, a local assessment of how the model behaves for specific audio frames is provided in the Urban Rhapsody interface.
- Underneath each frame we can see the output of each created model represented by the color scale.

Model Convergence



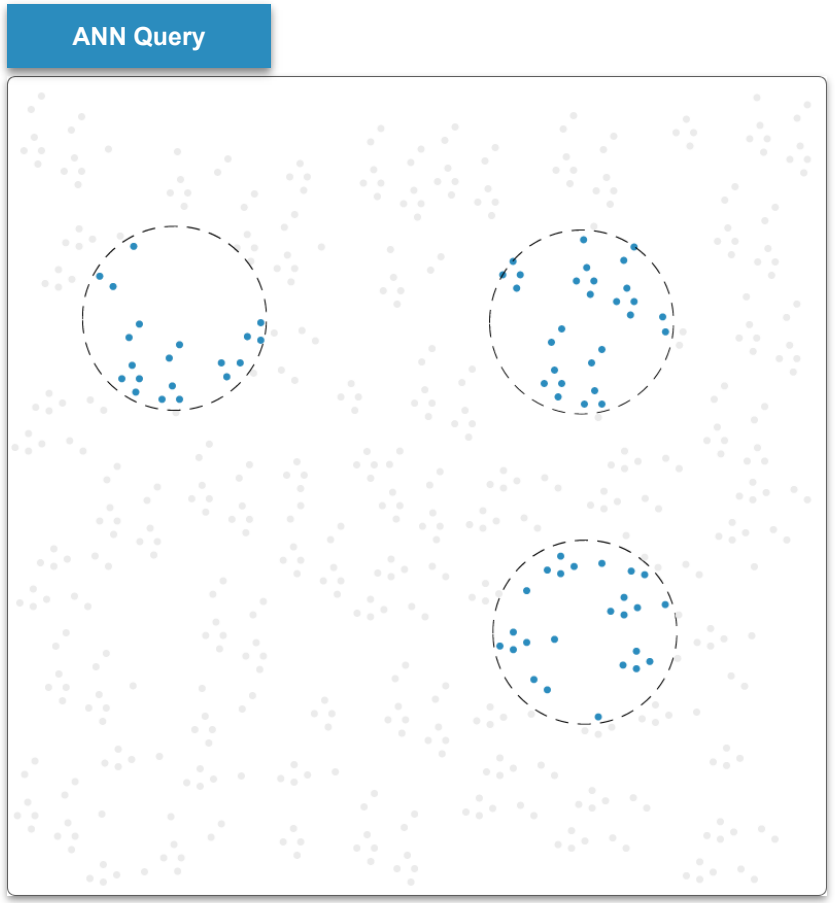

Frame Assessment




**[R3] Iterative Creation of Classification Models**

# Concept query

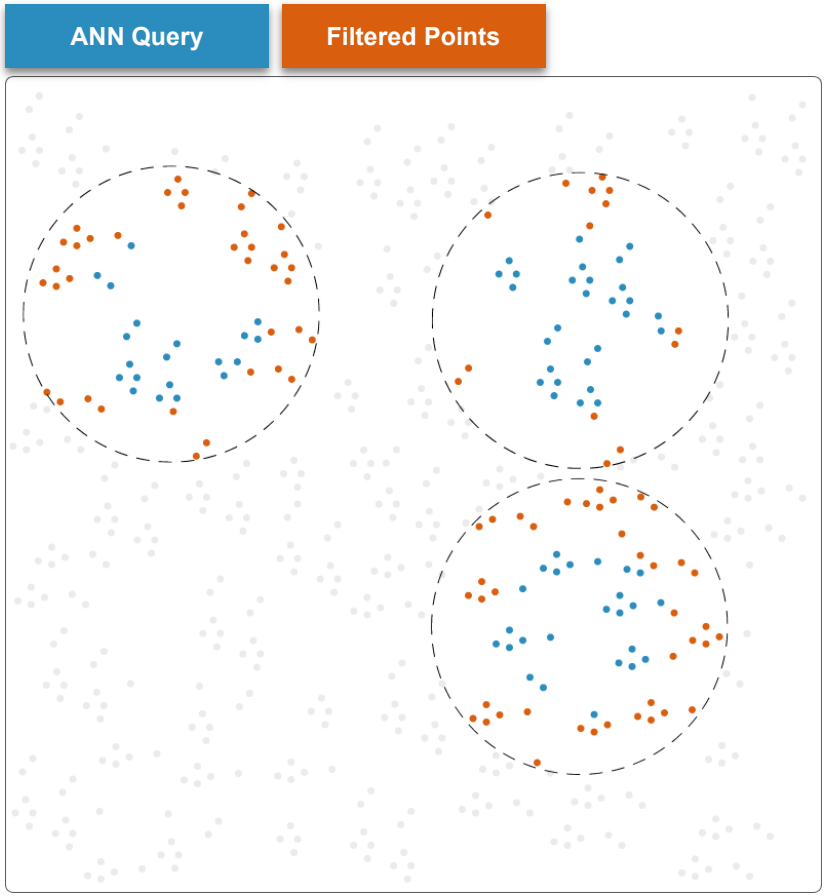
- Construction noise is usually a combination of many different audio sources.
- Powered Saw, Large Engines, and Drilling Machines are examples of sounds that you can find on a construction site.
- Urban Rhapsody allows users to run similarity queries based on these complex concepts
- Using the representative points as input for the similarity search and filtering the points based on the classification model.




**[R1] Interactive identification** and labeling **of similar concepts**

# Concept query

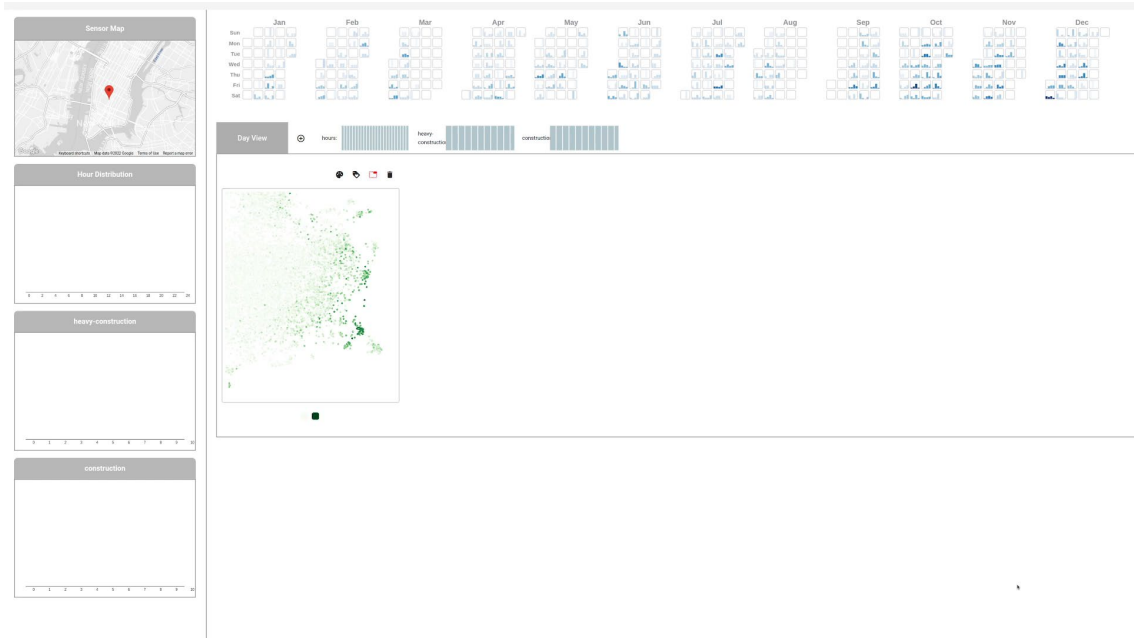
- Construction noise is usually a combination of many different audio sources.
- Powered Saw, Large Engines, and Drilling Machines are examples of sounds that you can find on a construction site.
- Urban Rhapsody allows users to run similarity queries based on these complex concepts
- Using the representative points as input for the similarity search and filtering the points based on the classification model.




**[R1] Interactive identification** and labeling **of similar concepts**

# Distribution view

- When a specific day is loaded, the user can interact with the projected scatterplots by selecting either individual points or subsets of the day through bounding boxes.
- When selecting points the user can see how those audio events are distributed over a day.
- The users are also able to see the distribution of the probabilities generated by a prototype for the selection made.



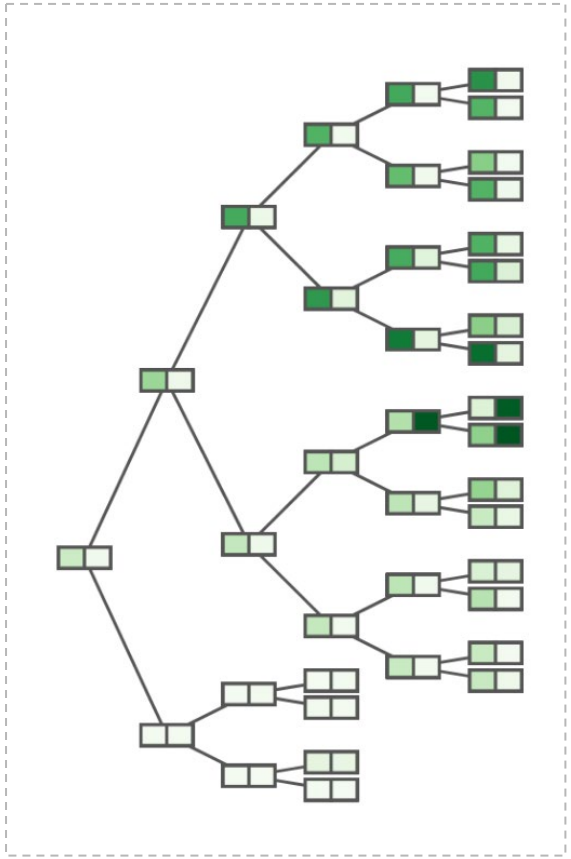

**[R4] Local** and Global **Audio Perspectives**

**[R2] Match between Audio and Visual Representations**

# Mixture explorer

- Although the day view of Urban Rhapsody provides an easier way to inspect instances from a given day, going through hours of recordings is still laborious
- Moreover, finding regions of the embeddings space that contains mixtures of sounds is a difficult task
- We took a visual approach to facilitate the selection and identification of regions that potentially contain mixture of sounds based on the trained models

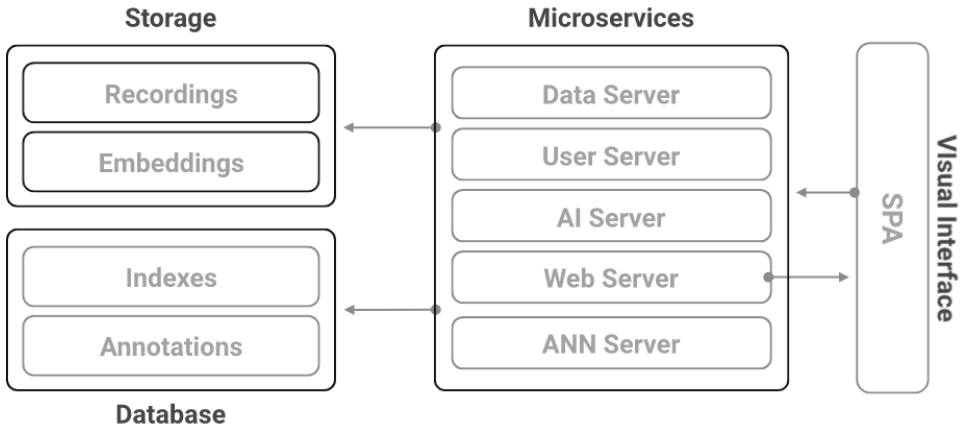
Mixture explorer



**[R1] Interactive identification** and labeling **of similar concepts**

# Architecture

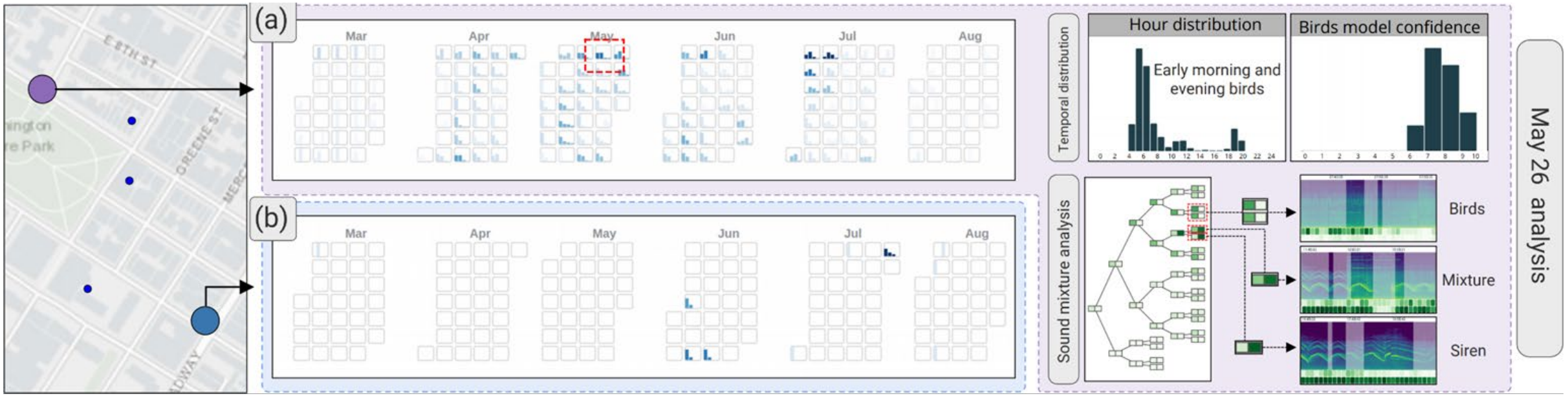
- We designed Urban Rhapsody following the microservices architecture, where each module should be responsible for very specific tasks
- This approach also helps us to extend the system and easily replace modules to test different implementations that can lead to performance improvements
- The interface was built using Angular framework to facilitate the reuse of modules, WebGL for data-intensive visualizations, and D3 for lightweight visualizations



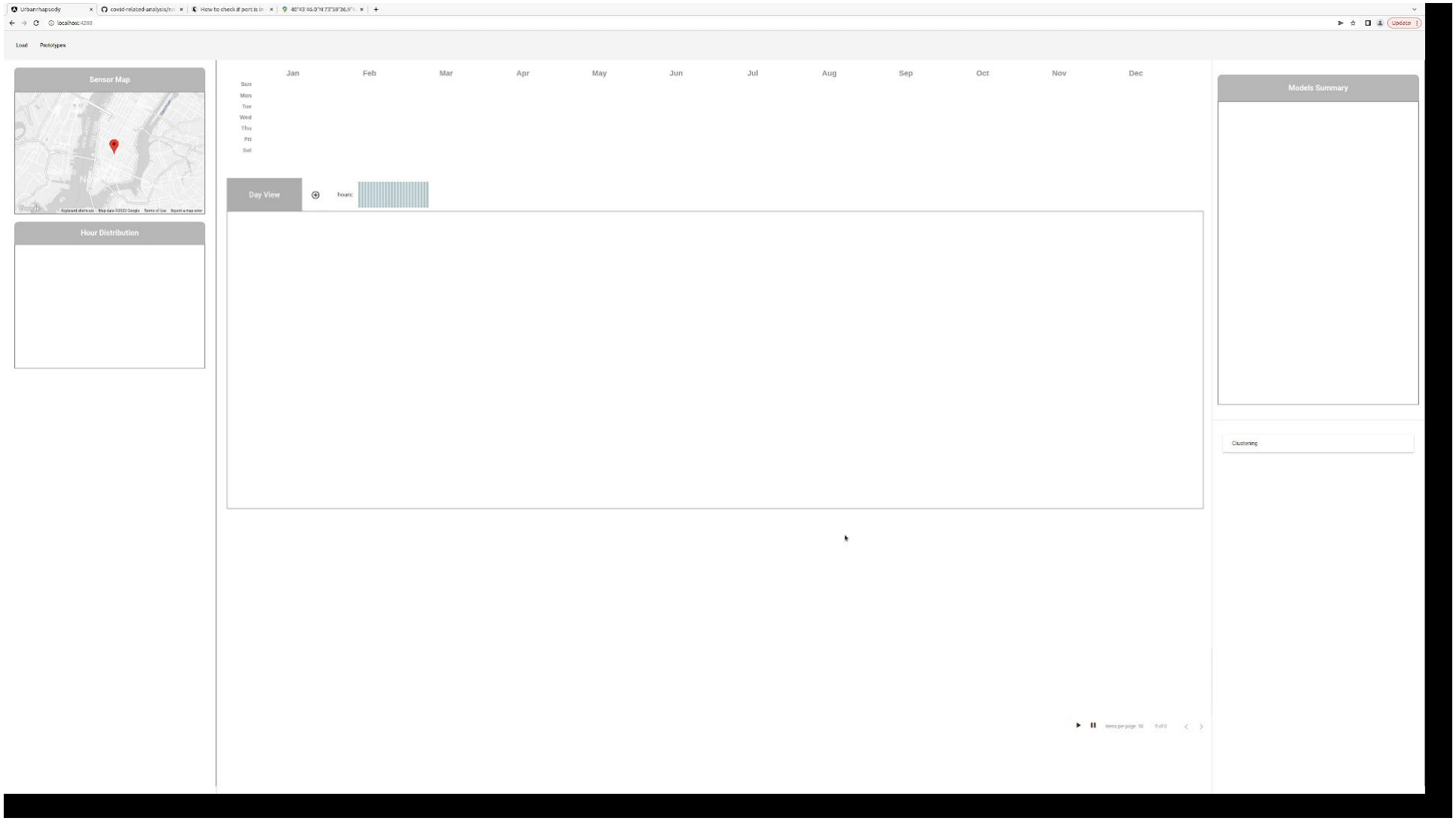
# Bioacoustics monitoring

- Does exposure to loud urban noise lead to significant changes in bird’s song traits and the time and frequency of their chorus?
- Do loud siren noises halt birds' dawn chorus?
- Are the birds nesting in noisy urban areas like Manhattan local parks adapted to the level of urban noise?

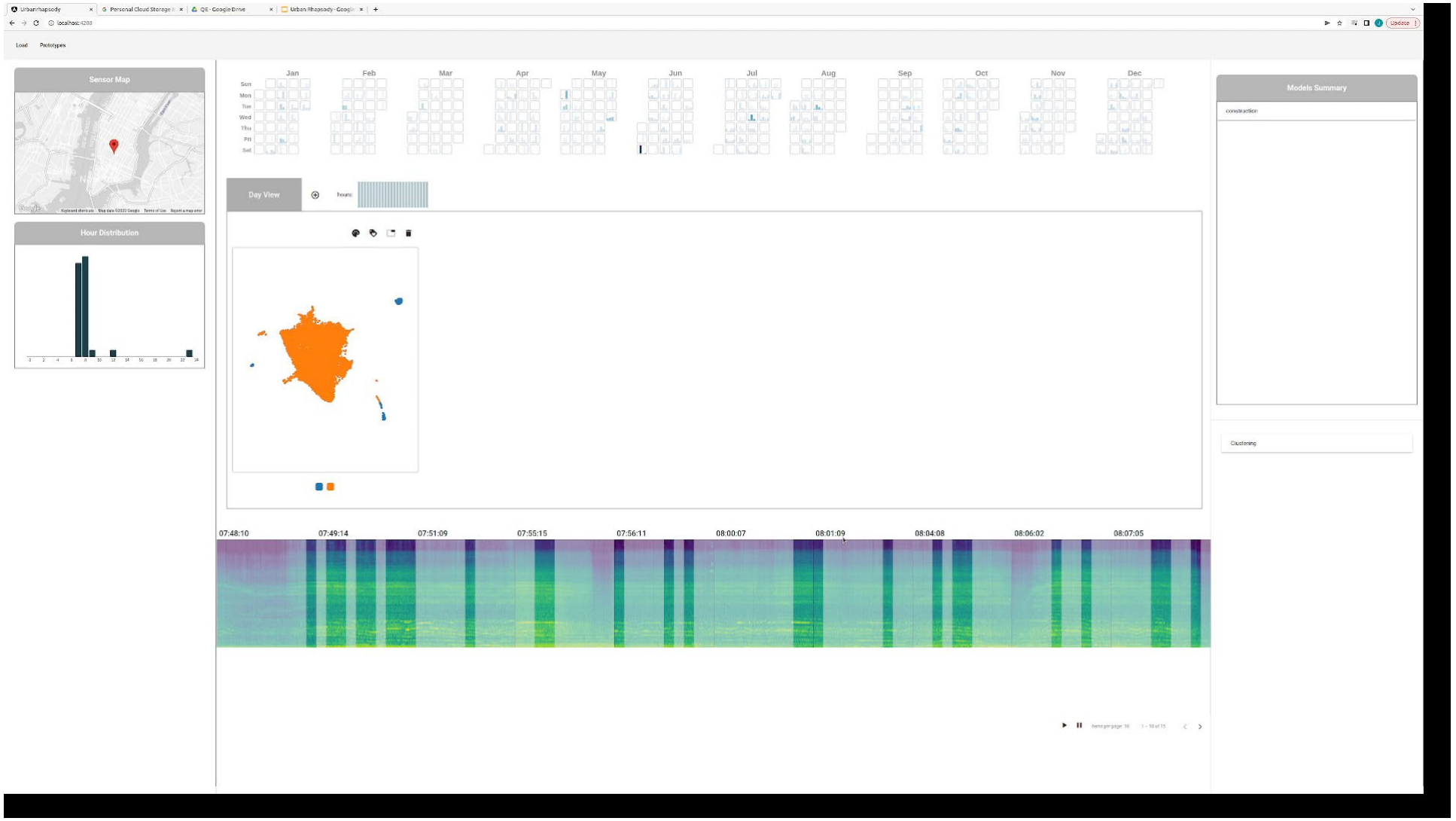
# Bioacoustic monitoring



# Construction in NYC: Querying



# Construction in NYC: Exploring



# Construction in NYC: Analyzing

# Construction in NYC


Construction


Heavy-construction