

UMAP

Uniform Manifold
Approximation and Projection for
Dimension Reduction

Бекназаров Назар

Мотивация

- Необходимо иметь возможность смотреть на данные

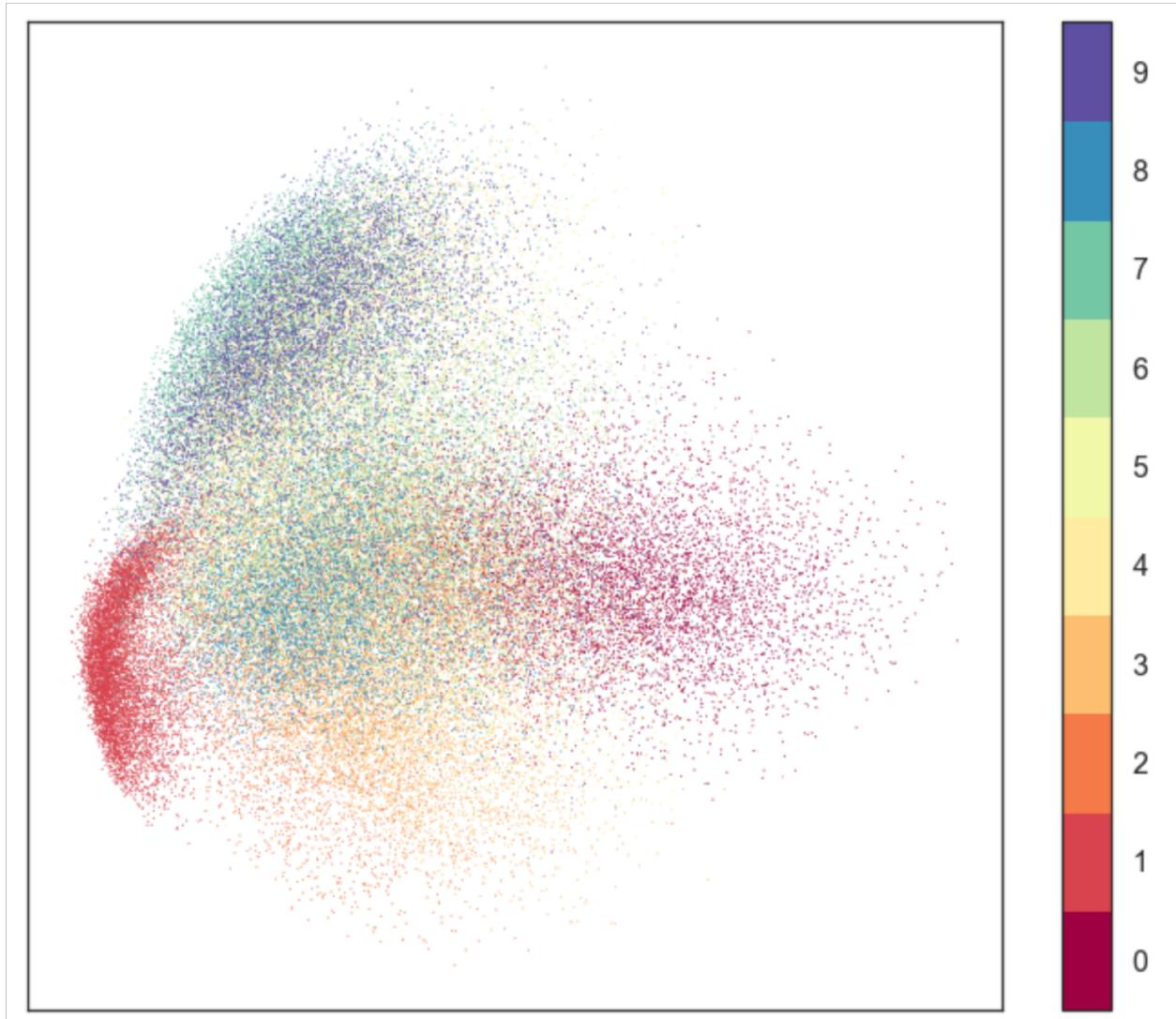
Методы на основе ближайших соседей.

- t-SNE
- Locally linear embedding
- UMAP
- Laplacian/Hessian eigenmaps
- Local tangent space alignment
- JSE
- Iso-map

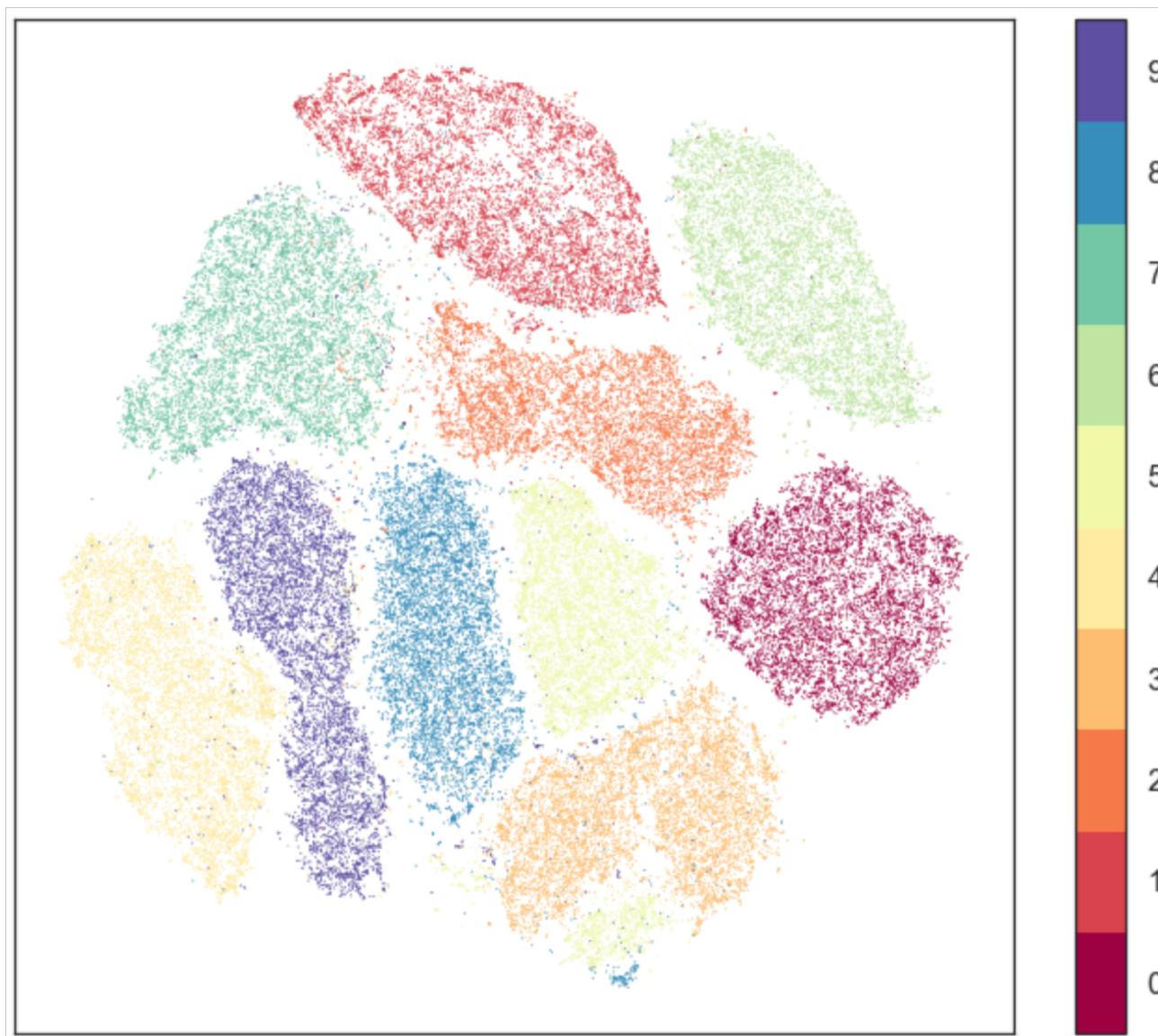
Методы на основе факторизации матриц

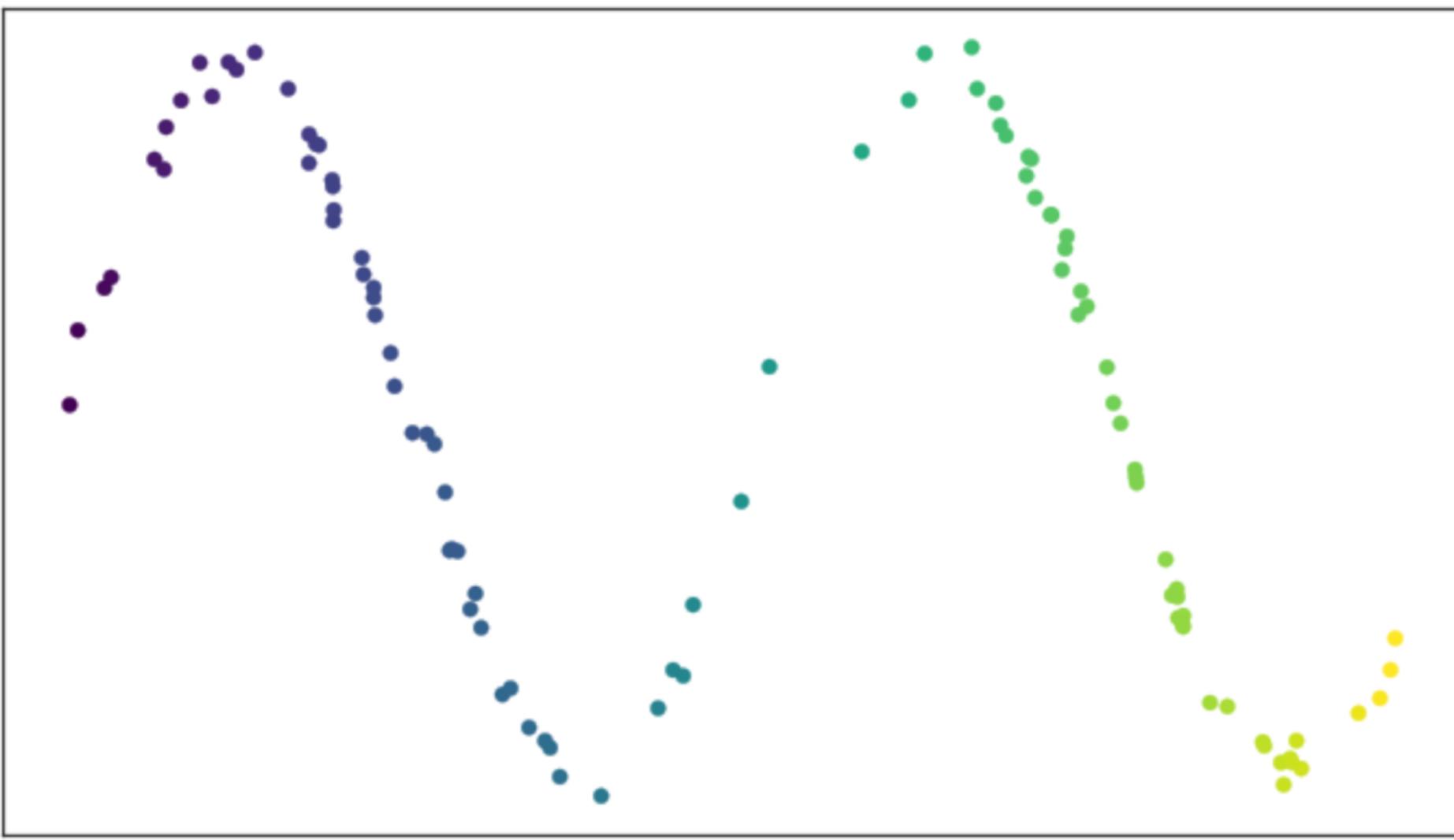
- PCA
- Linear Autoencoders
- LDA
- Non-negative matrix factorization
- Generalised low rank models
- Word2Vec
- Glove

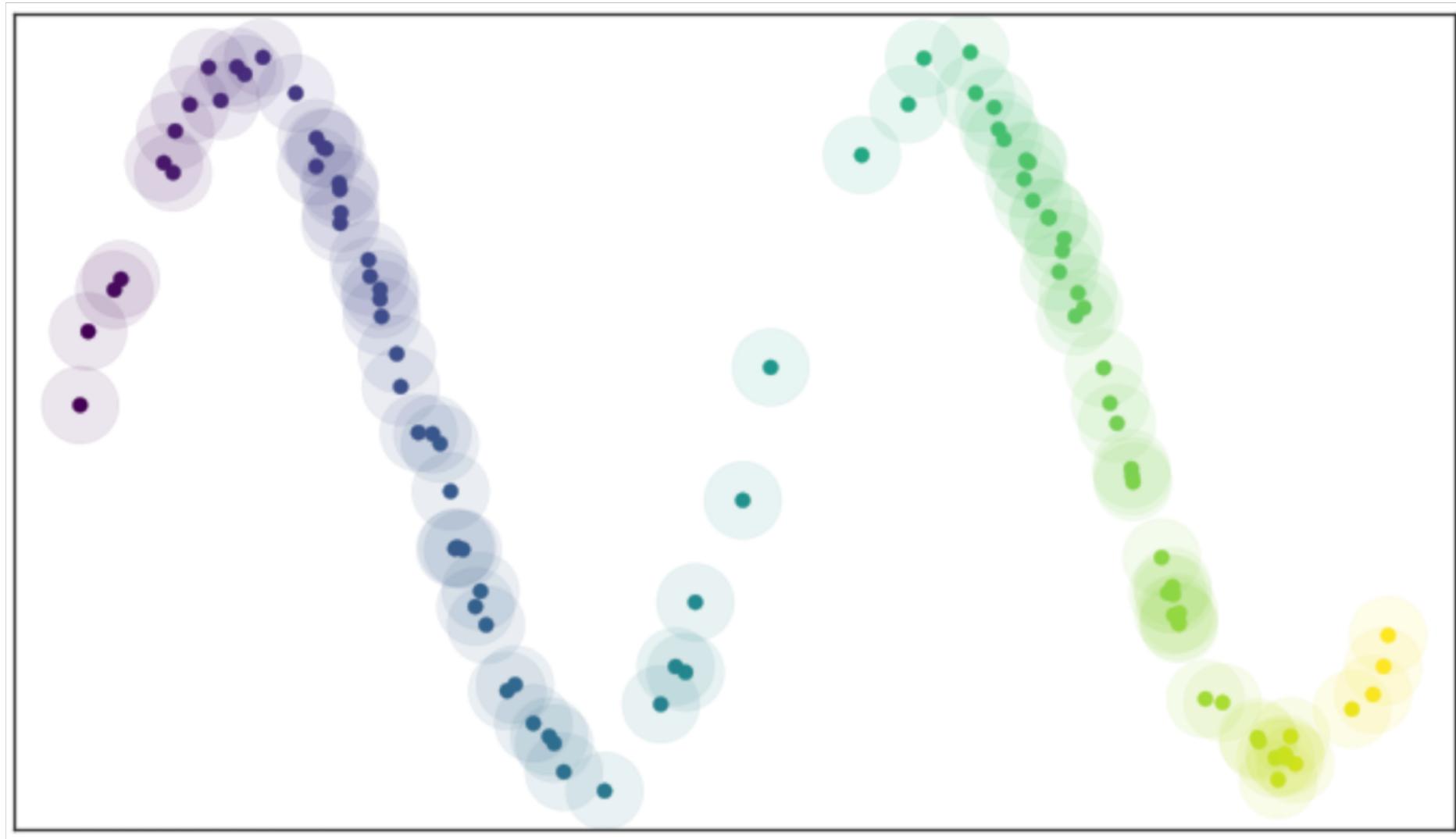
PCA на MNIST

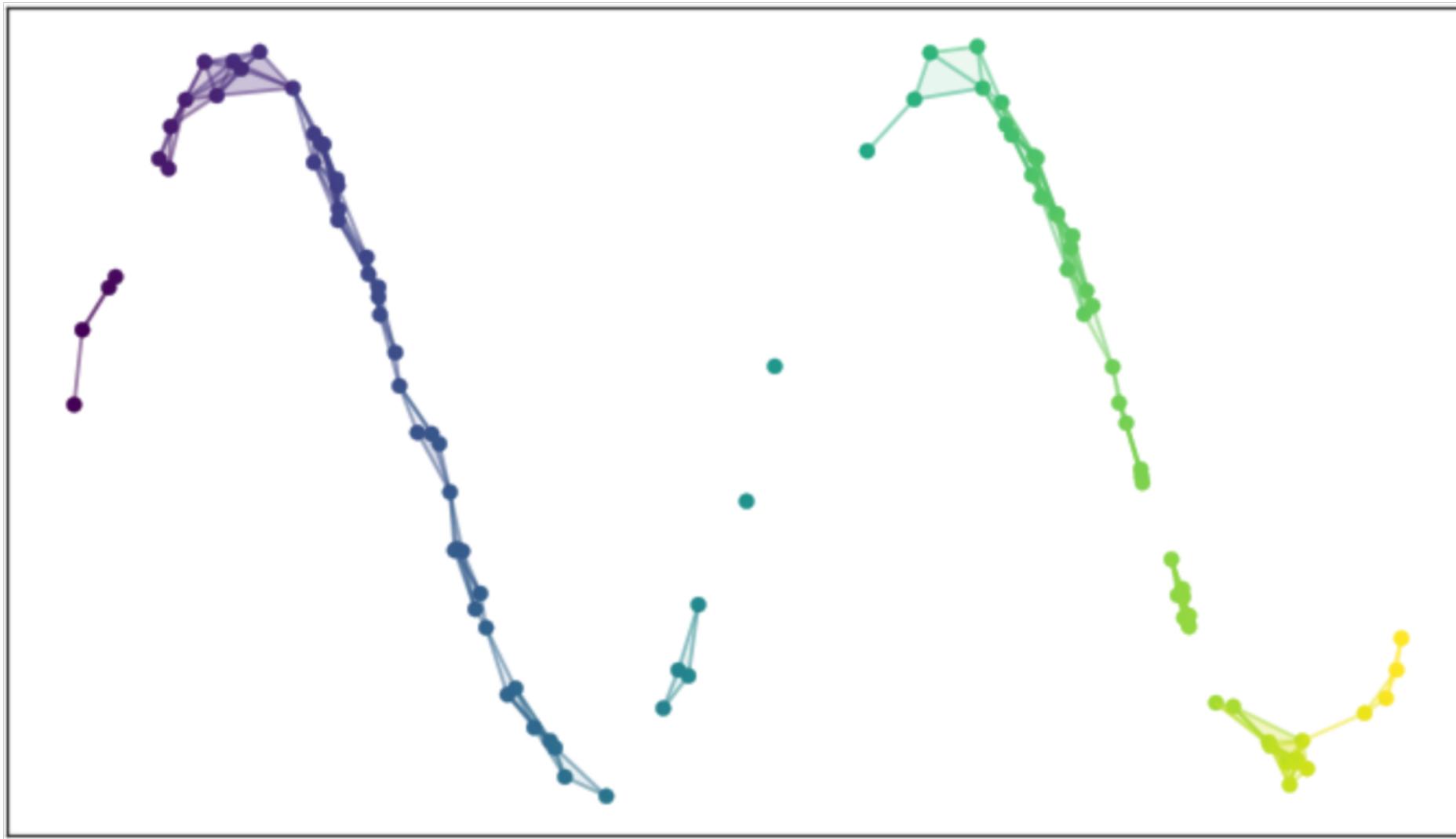


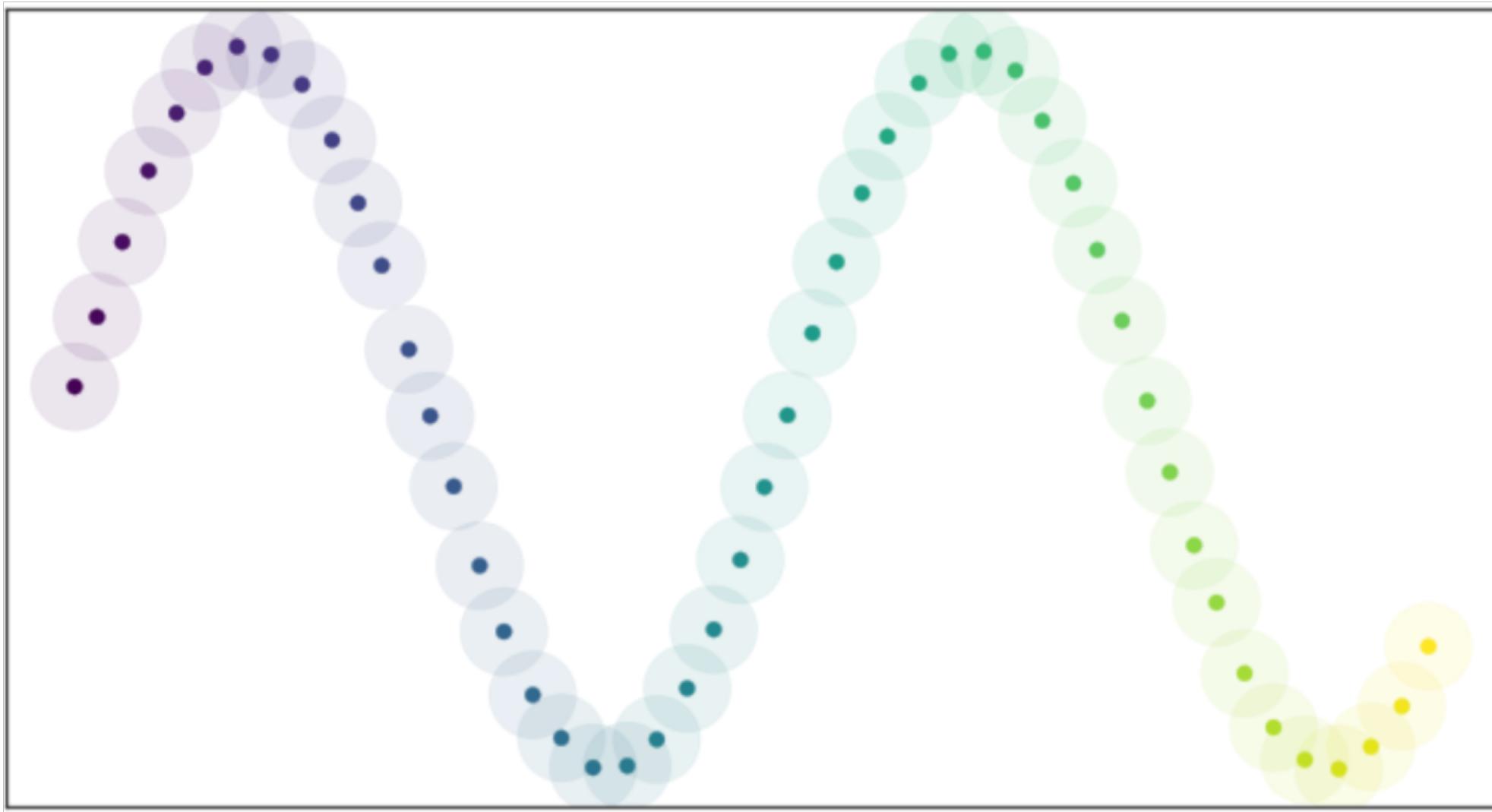
t-SNE на MNIST







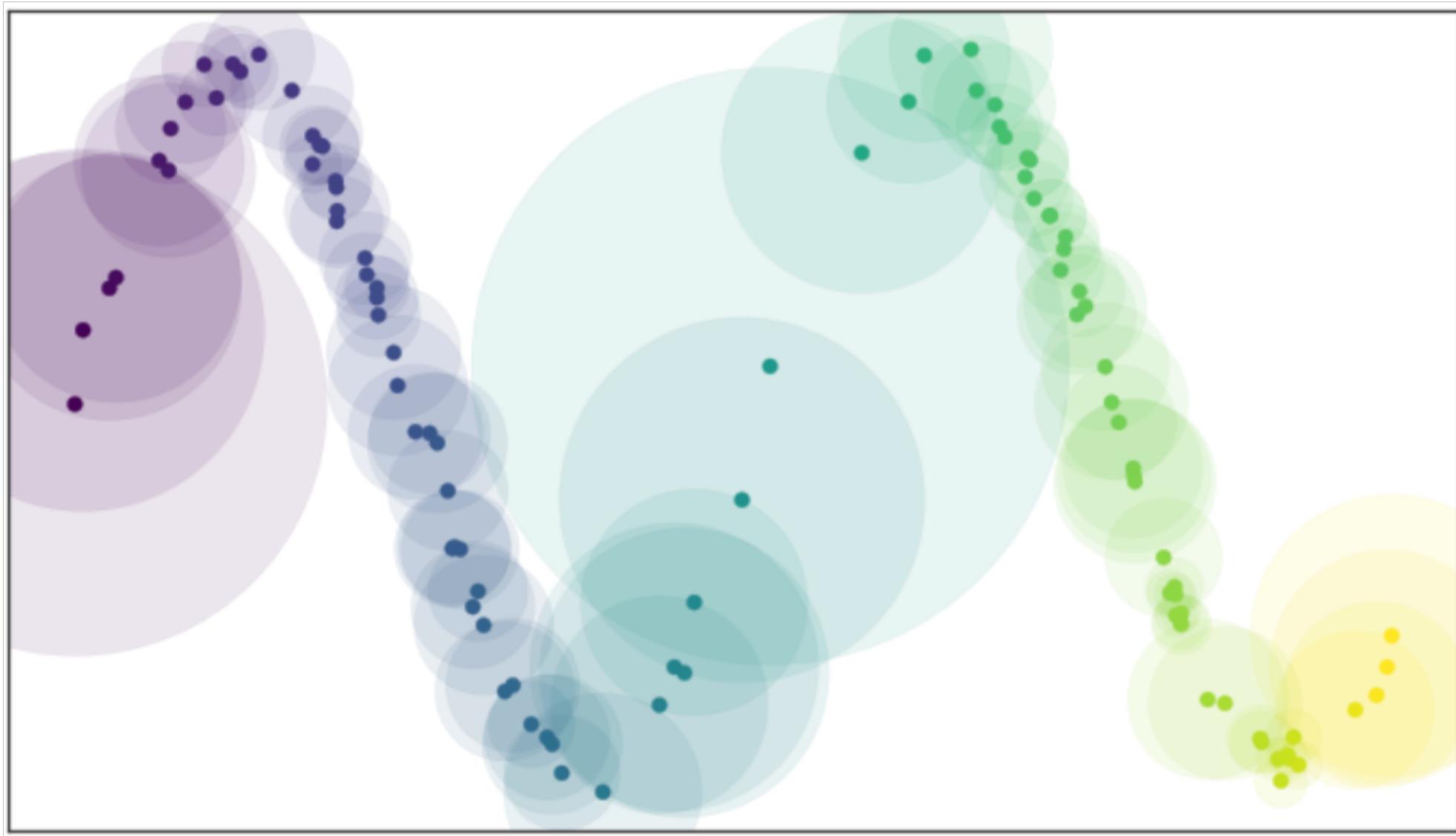


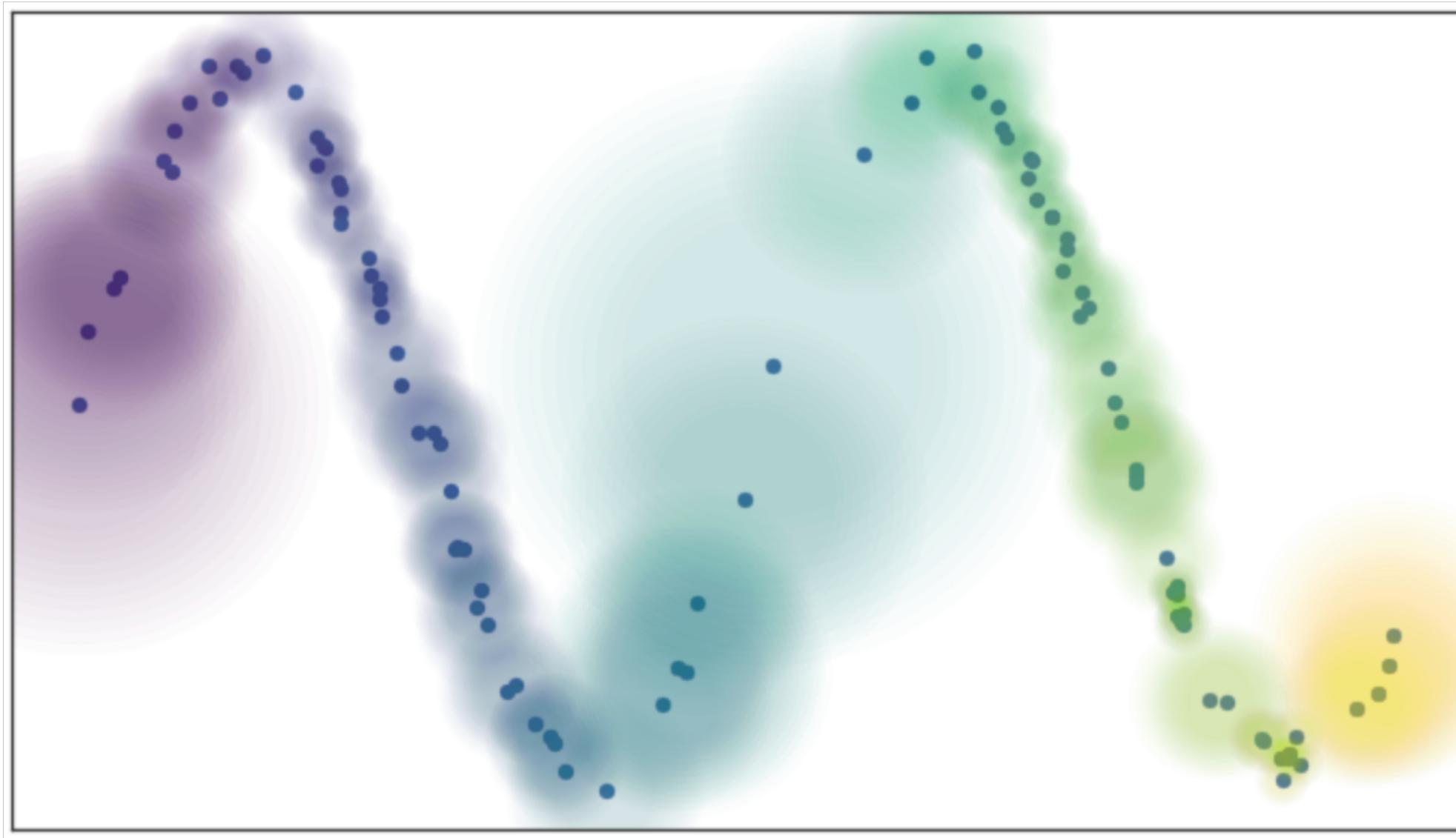


Часто ли данные равномерно распределены?

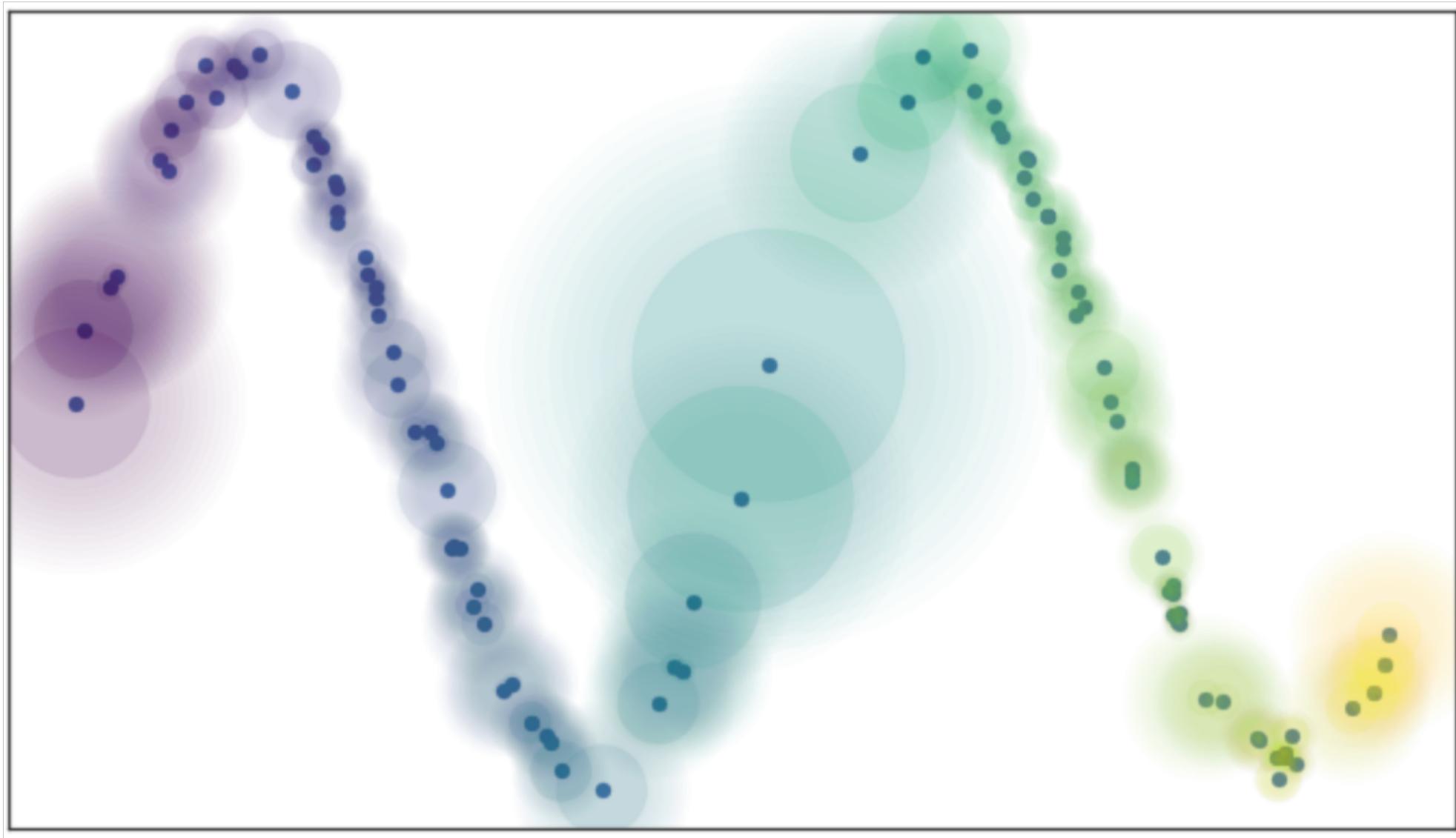
Да, если взять нужную метрику.

Для каждой точки в окрестности определяем
Риманову метрику на этой окрестности.

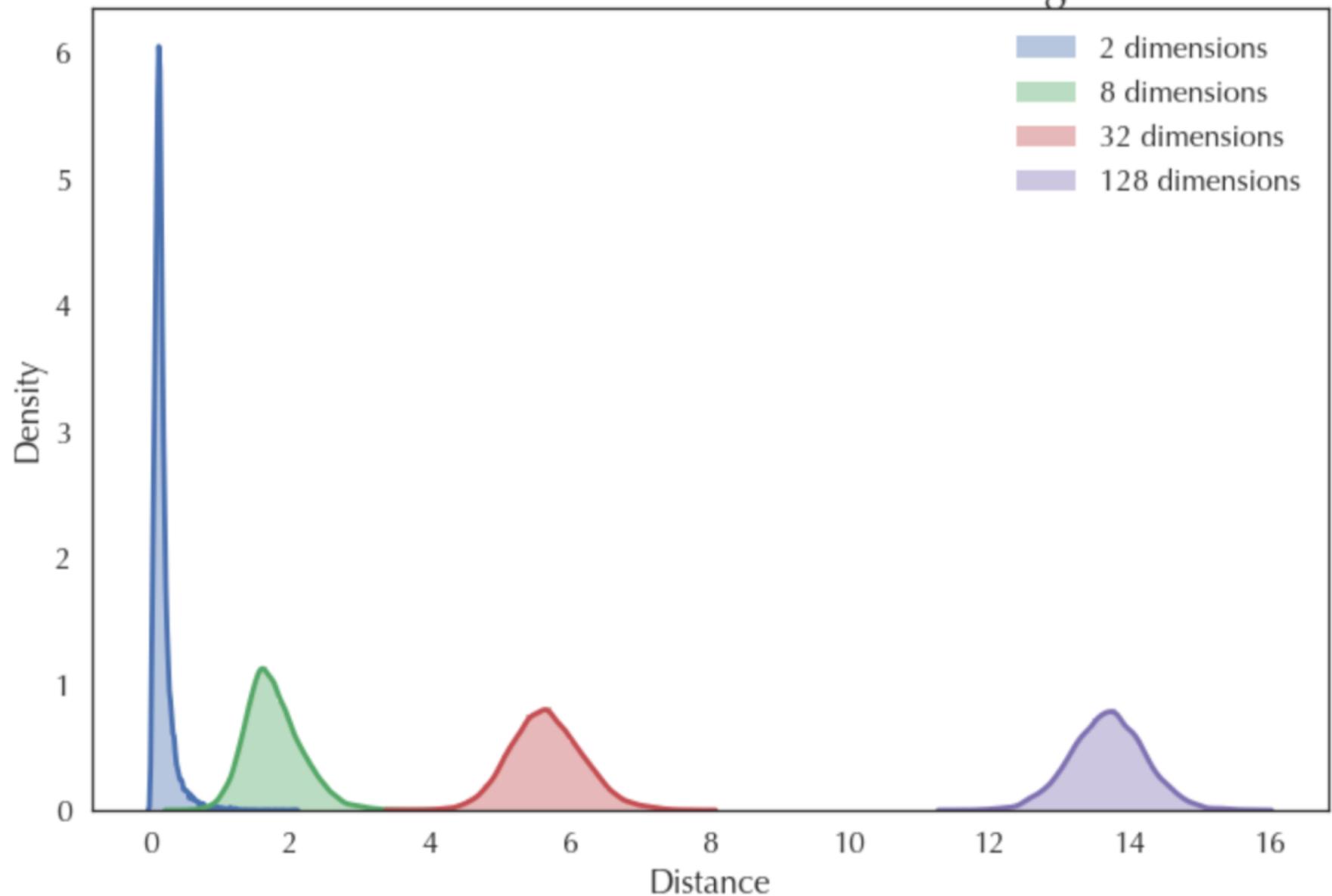




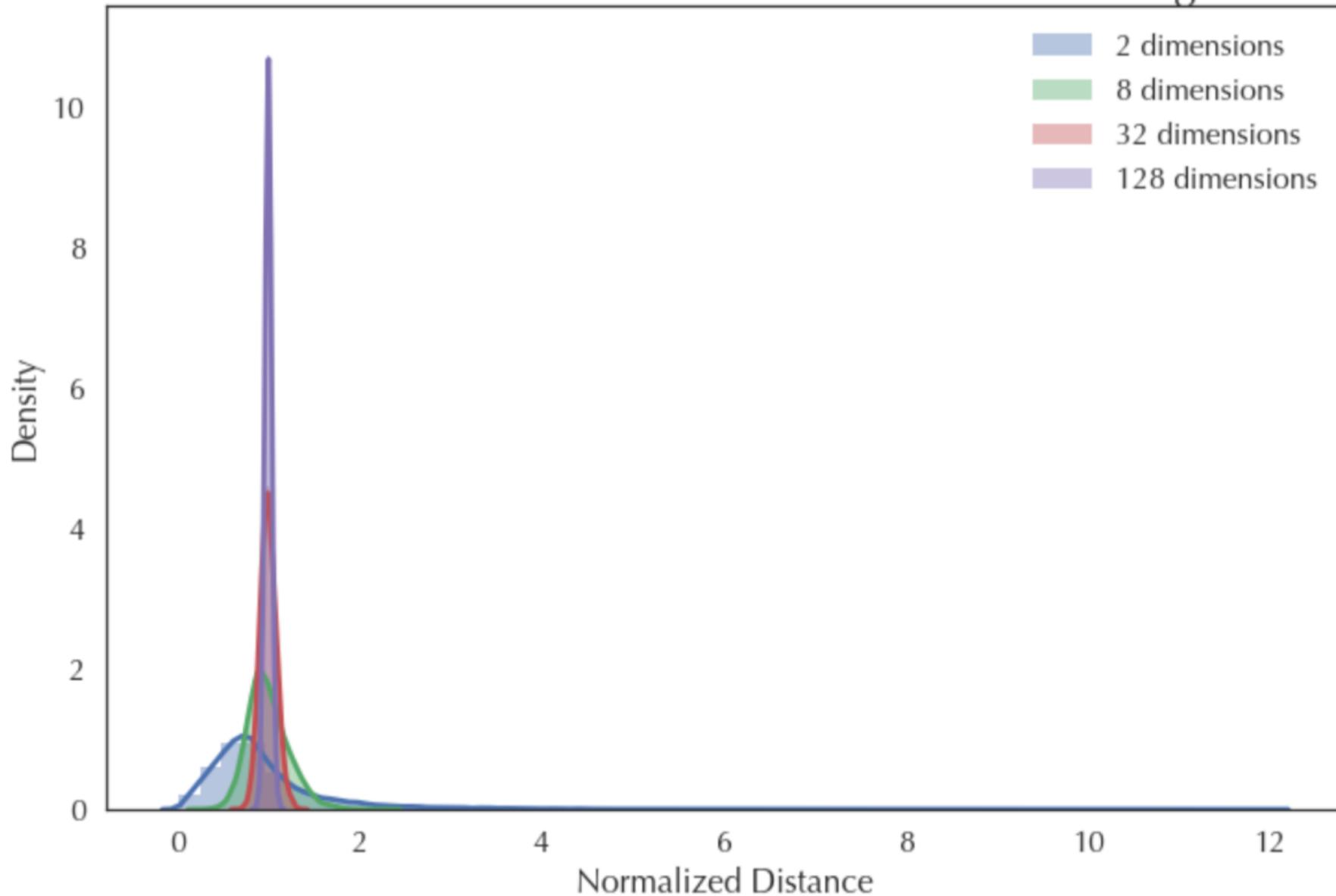
Предположим, что нет изолированных точек.



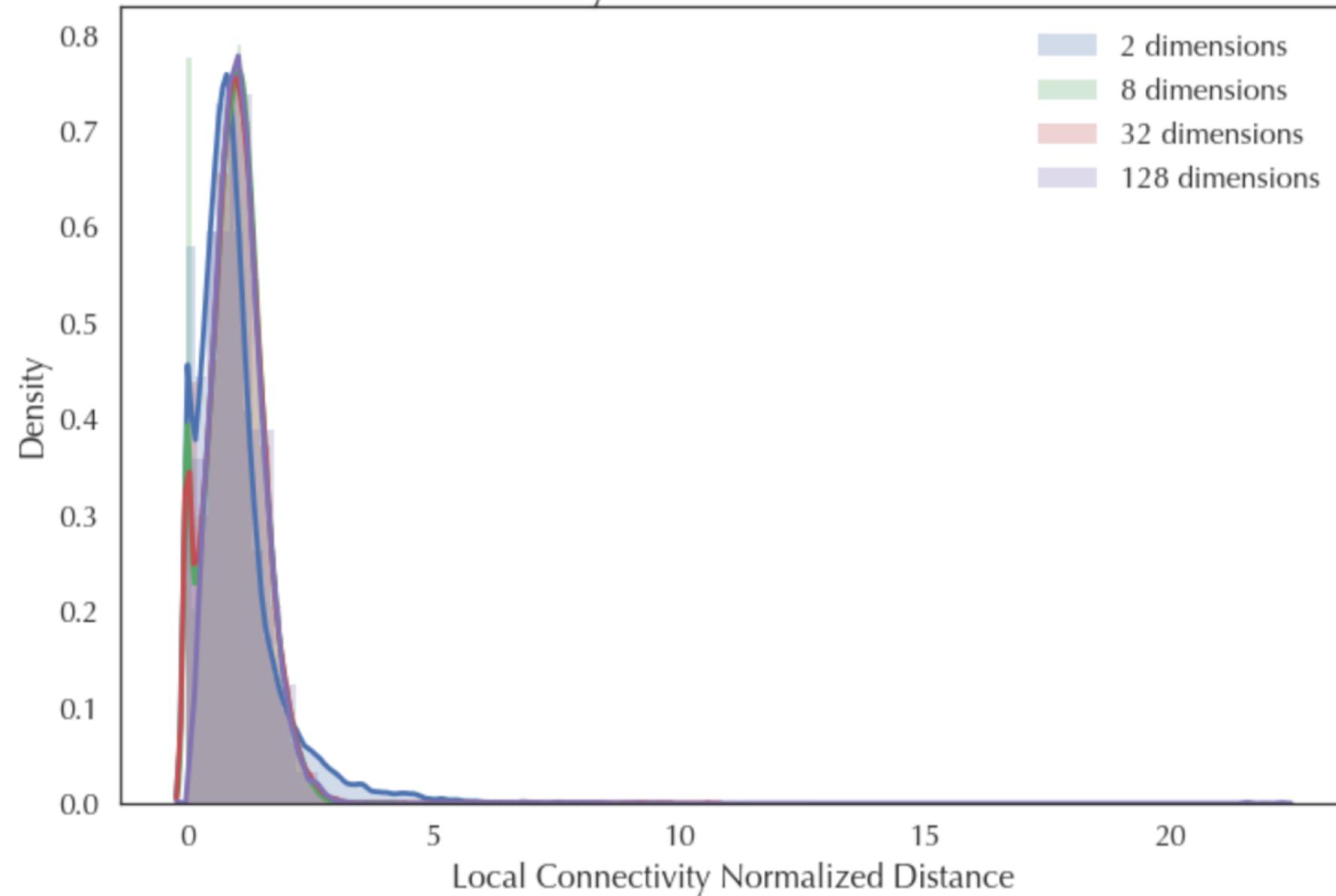
Distribution of distances to 20 nearest neighbors



Distribution of normalized distances to 20 nearest neighbors



Distribution of local connectivity normalized distances to 20 nearest neighbors



Используя теорию нечетких множеств
получим итоговую метрику

$$f(\alpha, \beta) = \alpha + \beta - \alpha \cdot \beta$$

Формальная формула

For each x_i we will define ρ_i and σ_i . Let

$$\rho_i = \min\{d(x_i, x_{i_j}) \mid 1 \leq j \leq k, d(x_i, x_{i_j}) > 0\},$$

and set σ_i to be the value such that

$$\sum_{j=1}^k \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right) = \log_2(k).$$

Построение весов графа

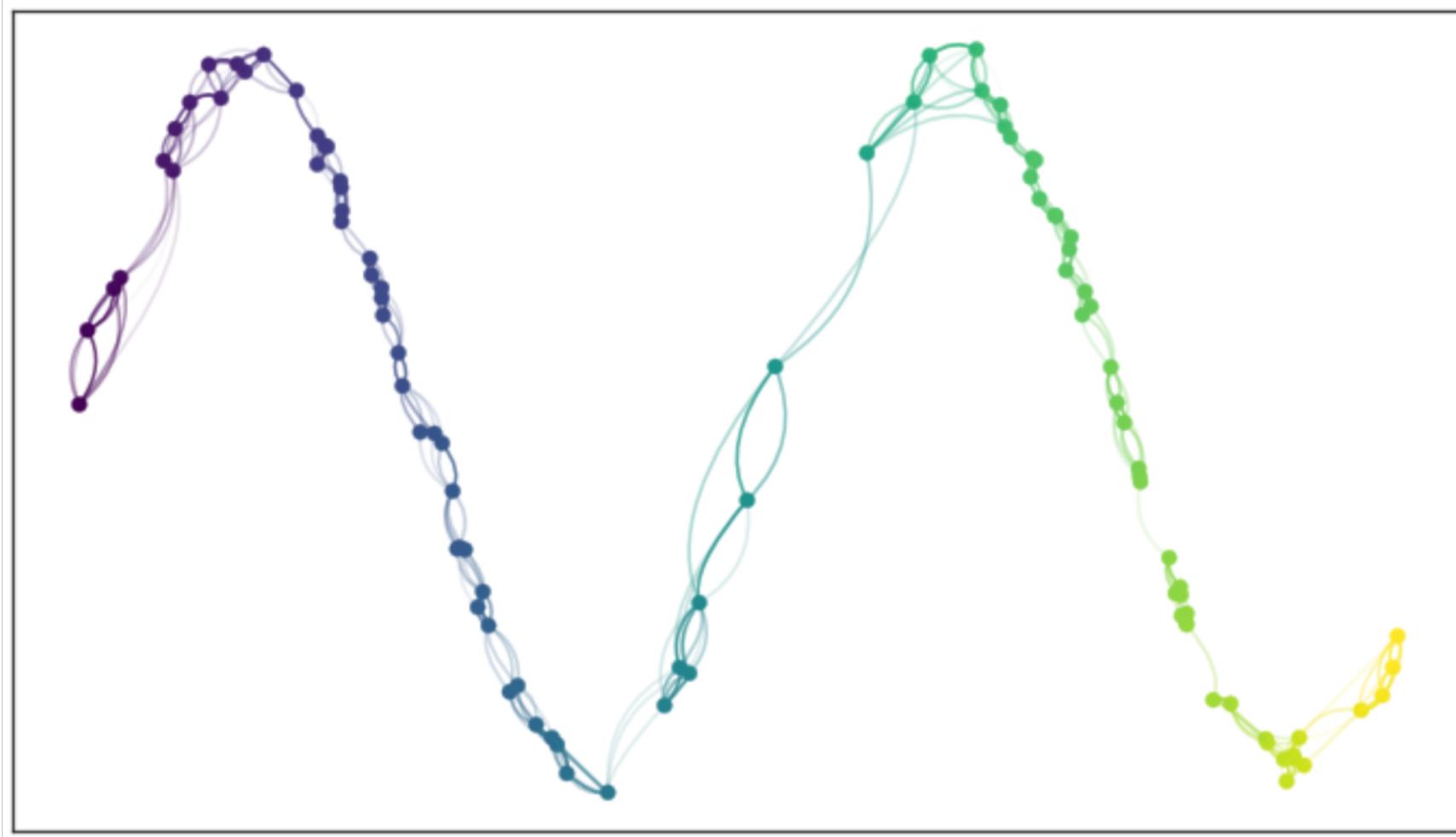
We can now define a weighted directed graph $\bar{G} = (V, E, w)$. The vertices V of \bar{G} are simply the set X . We can then form the set of directed edges $E = \{(x_i, x_{i_j}) \mid 1 \leq j \leq k, 1 \leq i \leq N\}$, and define the weight function w by setting

$$w((x_i, x_{i_j})) = \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right).$$

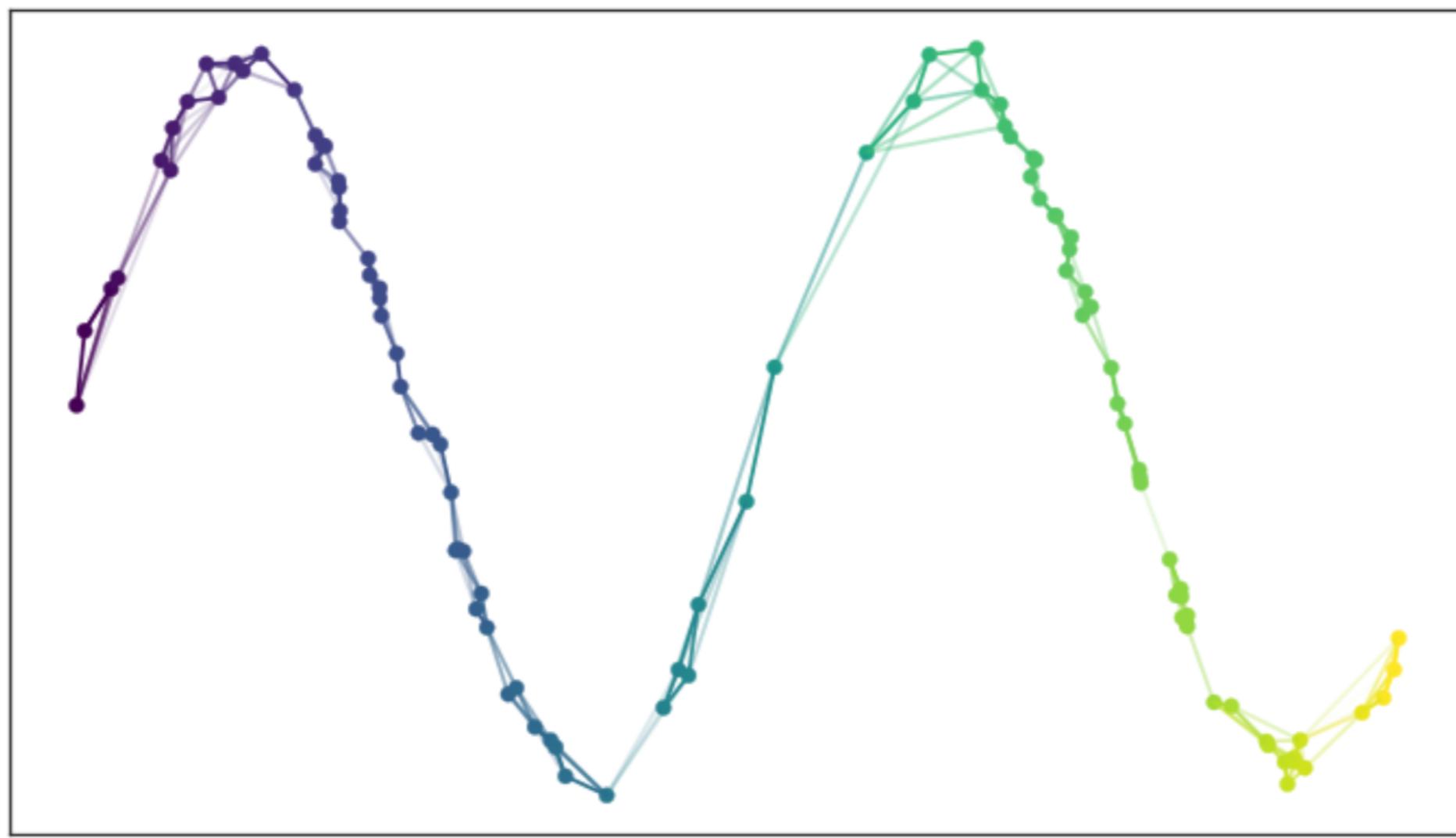
Let A be the weighted adjacency matrix of \bar{G} , and consider the symmetric matrix

$$B = A + A^\top - A \circ A^\top,$$

Для каждой пары точек получаем 2
расстояния



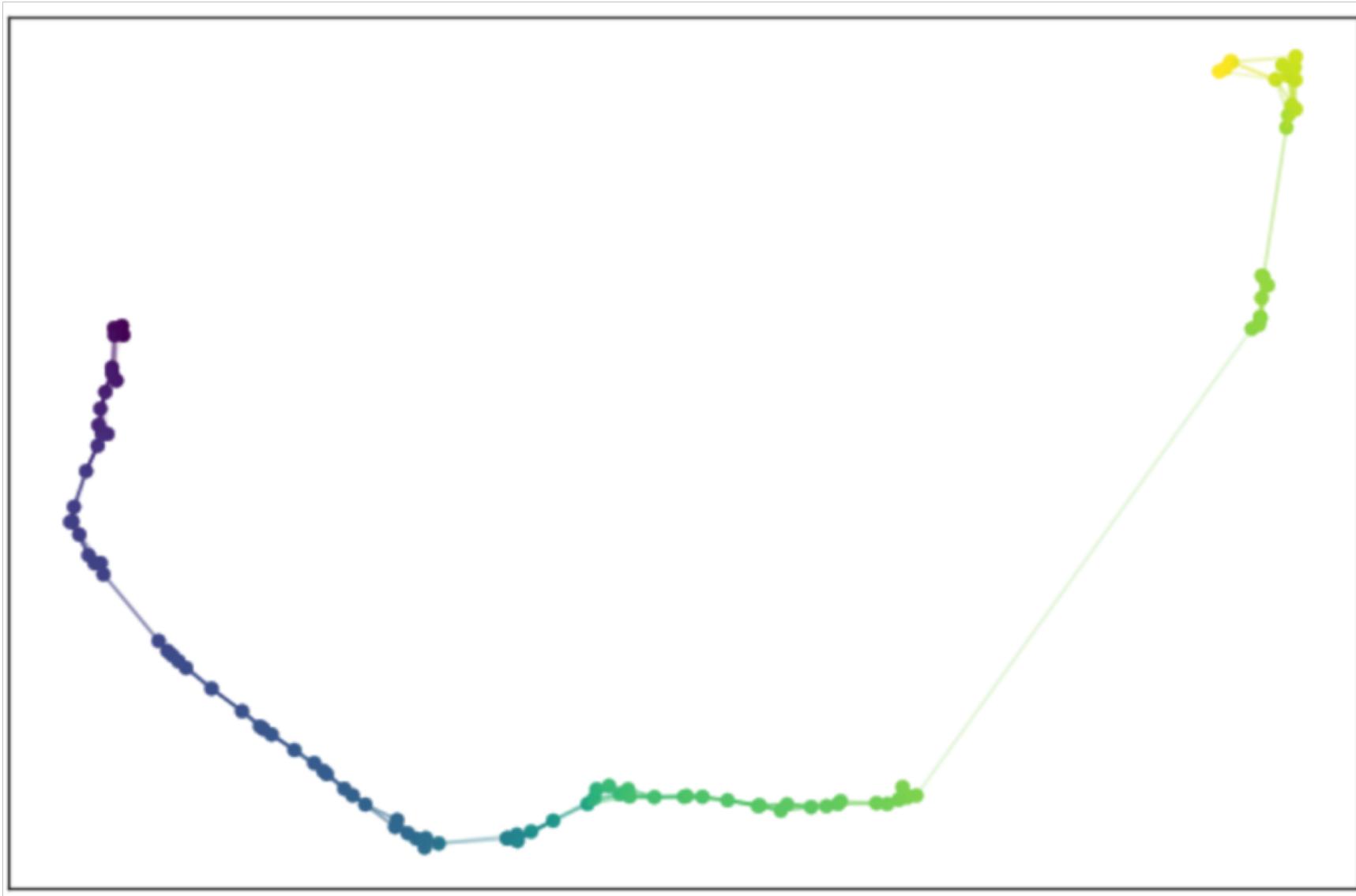
Полученный граф ближайших соседей.

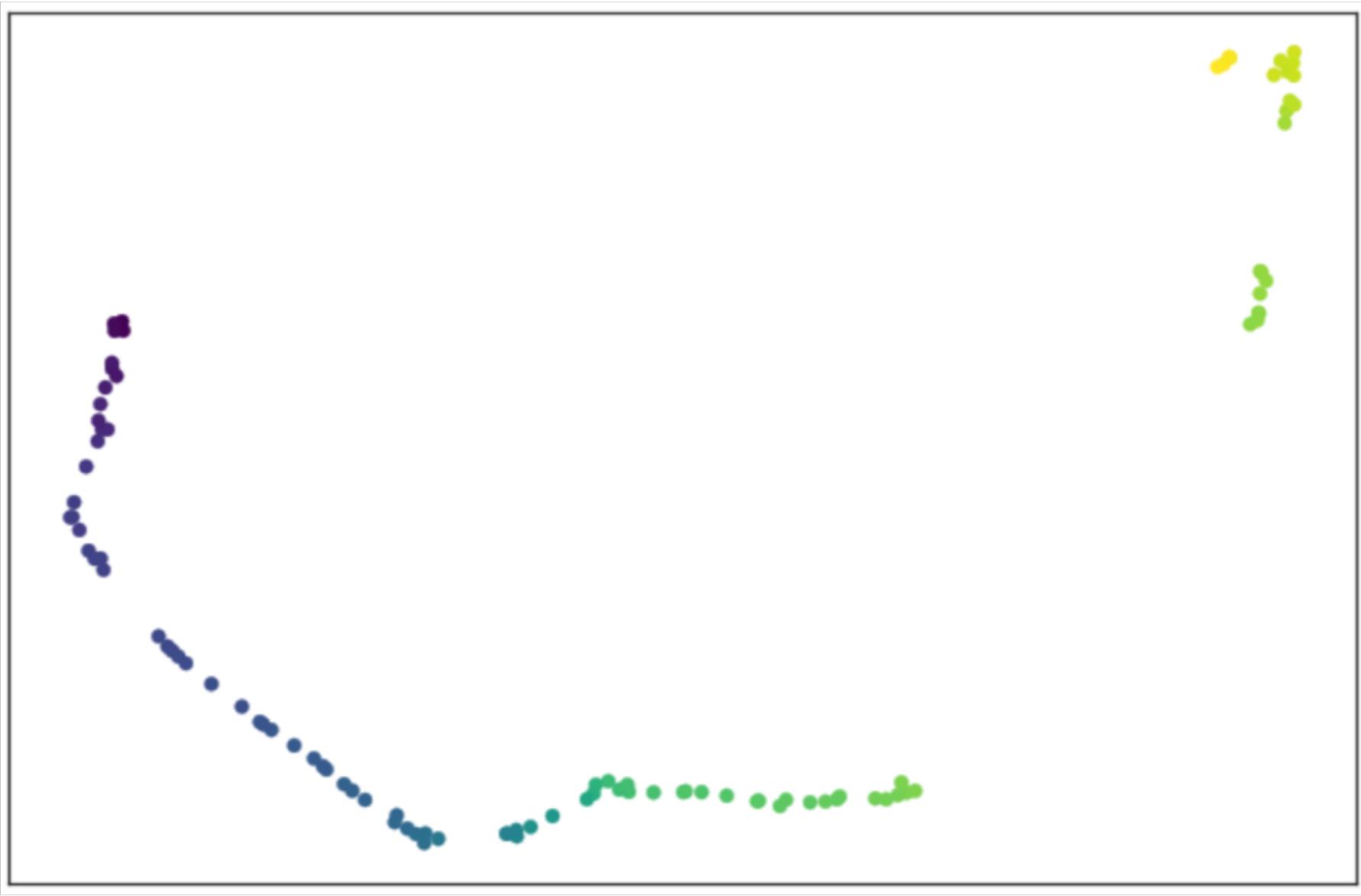


Получив граф – далее просто отображаем его на n -мерную плоскость.

Для эмбеддинга графа используем кросс-энтропию для двух пространств $(A, \mu), (A, \nu)$

$$\sum_{a \in A} \mu(a) \log \left(\frac{\mu(a)}{\nu(a)} \right) + (1 - \mu(a)) \log \left(\frac{1 - \mu(a)}{1 - \nu(a)} \right)$$





Пример работы

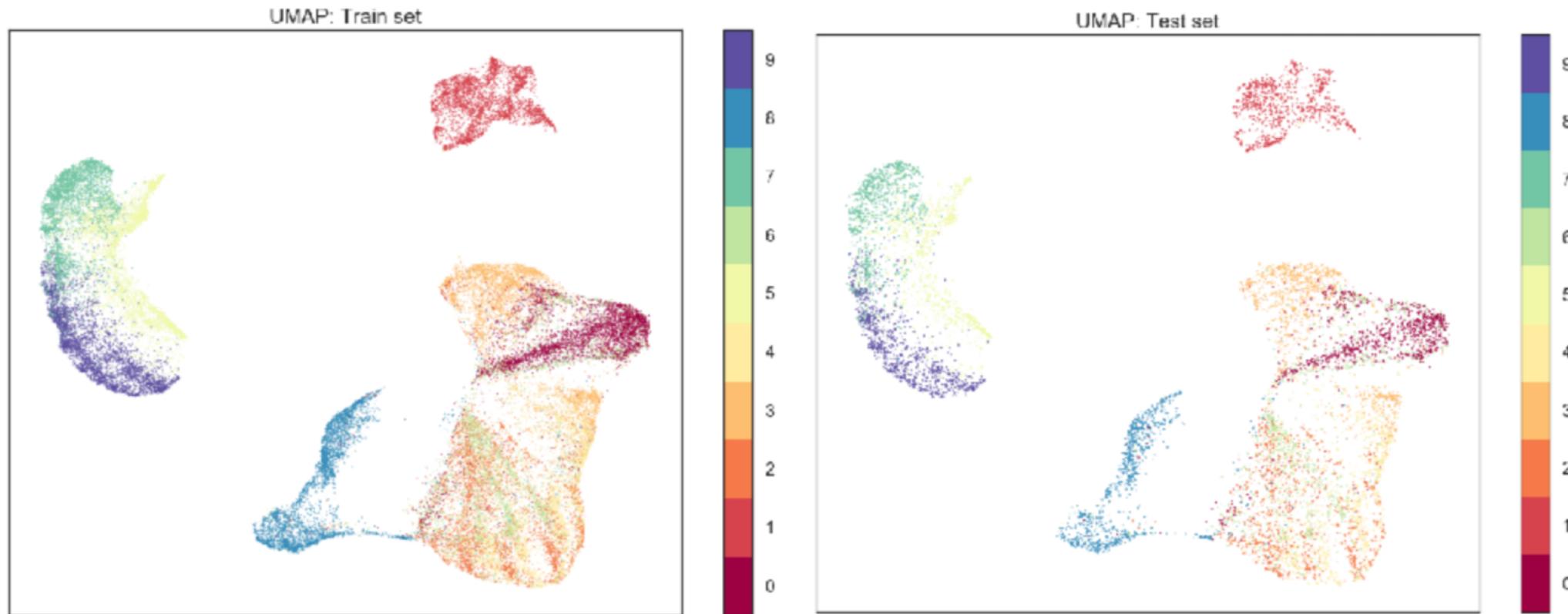
<https://www.youtube.com/watch?v=nCk8dyU7zUM>

Реализация

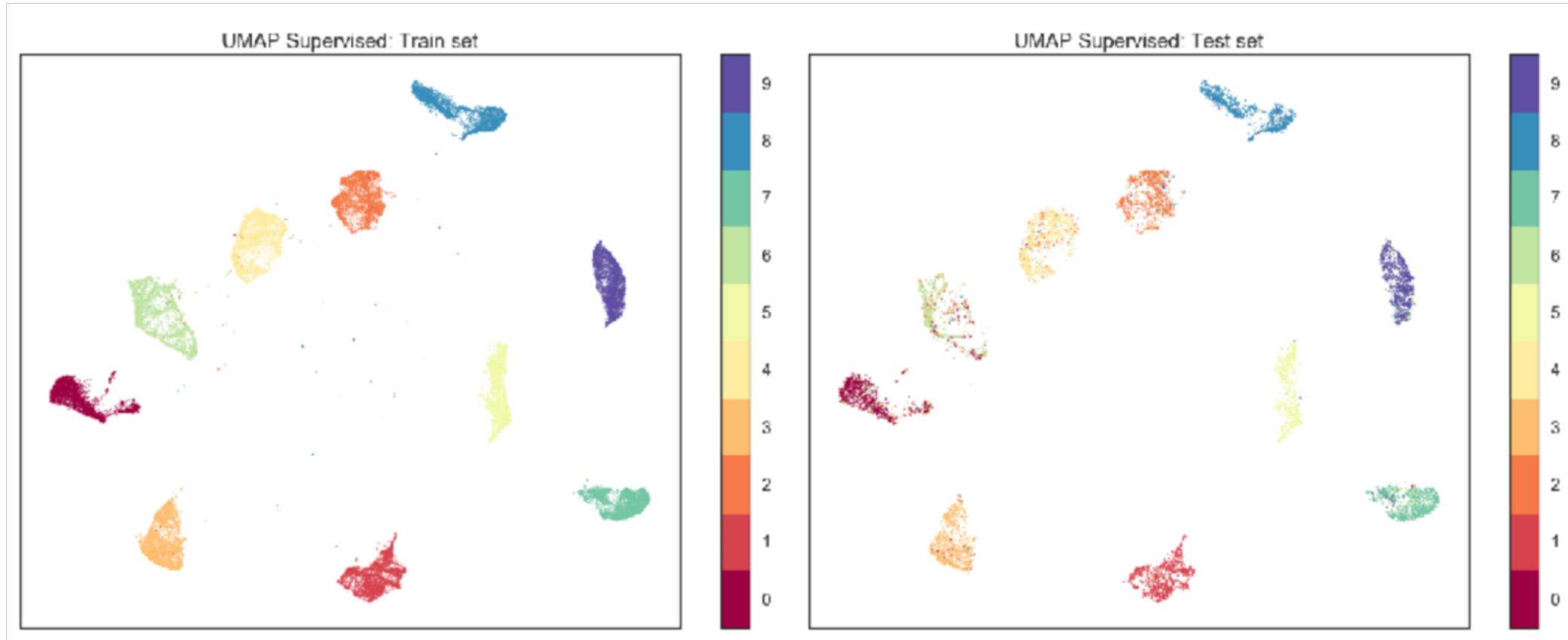
- Требуется быстро считать дерево ближайших соседей
- RP-trees + NN-decent
- Построение эмбеддинга за суб-квадратную сложность
- SGD + negative sampling
- Реализовать
- Текущая реализация на Python + numba

Преимущества УМАР

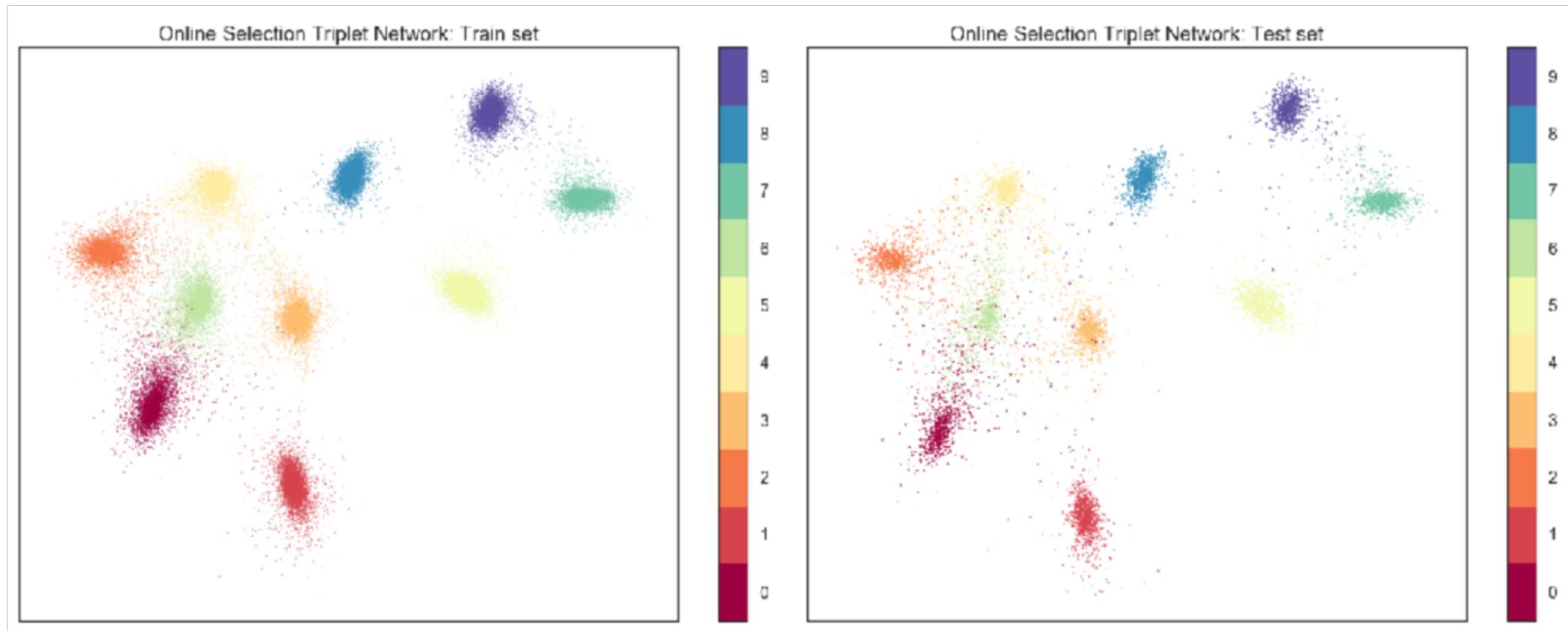
Доступна операция transform, эмбеддинг
новых точек уже после обучения



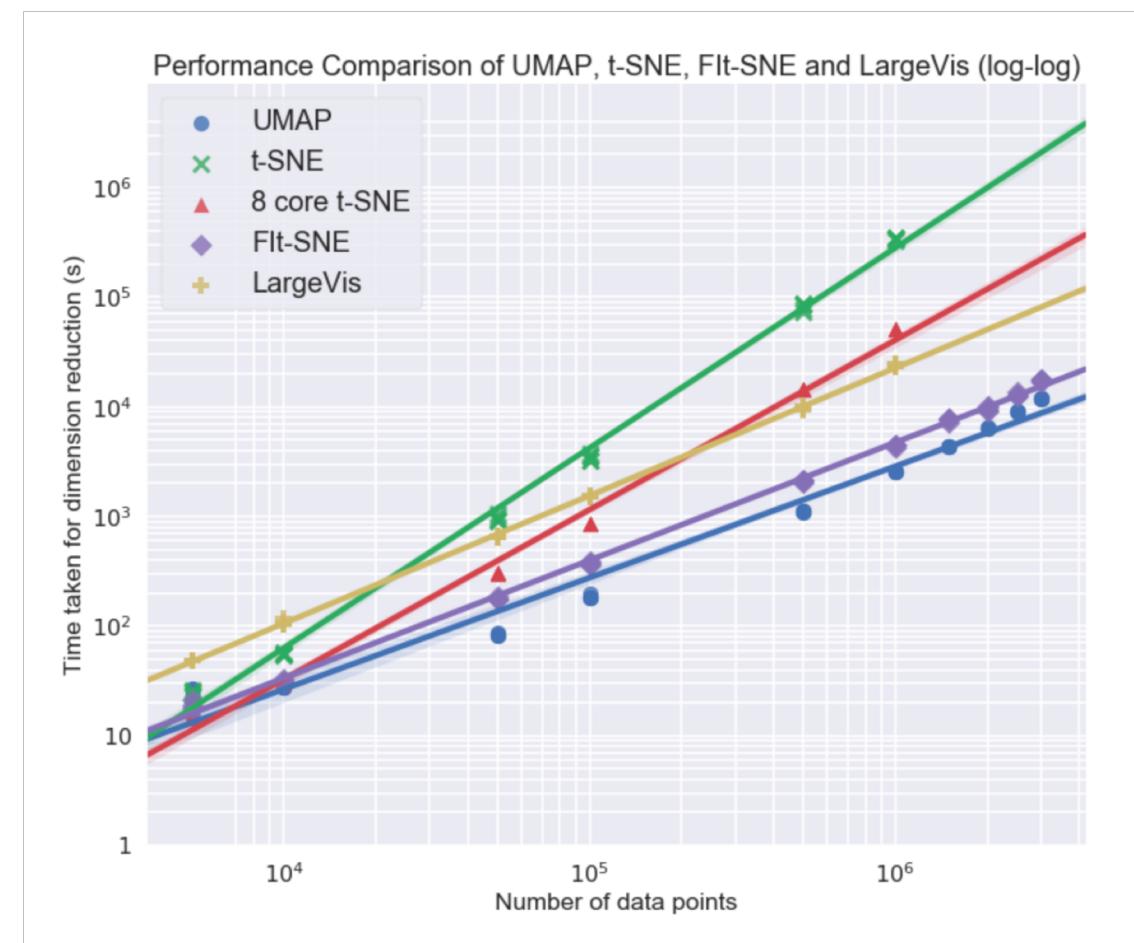
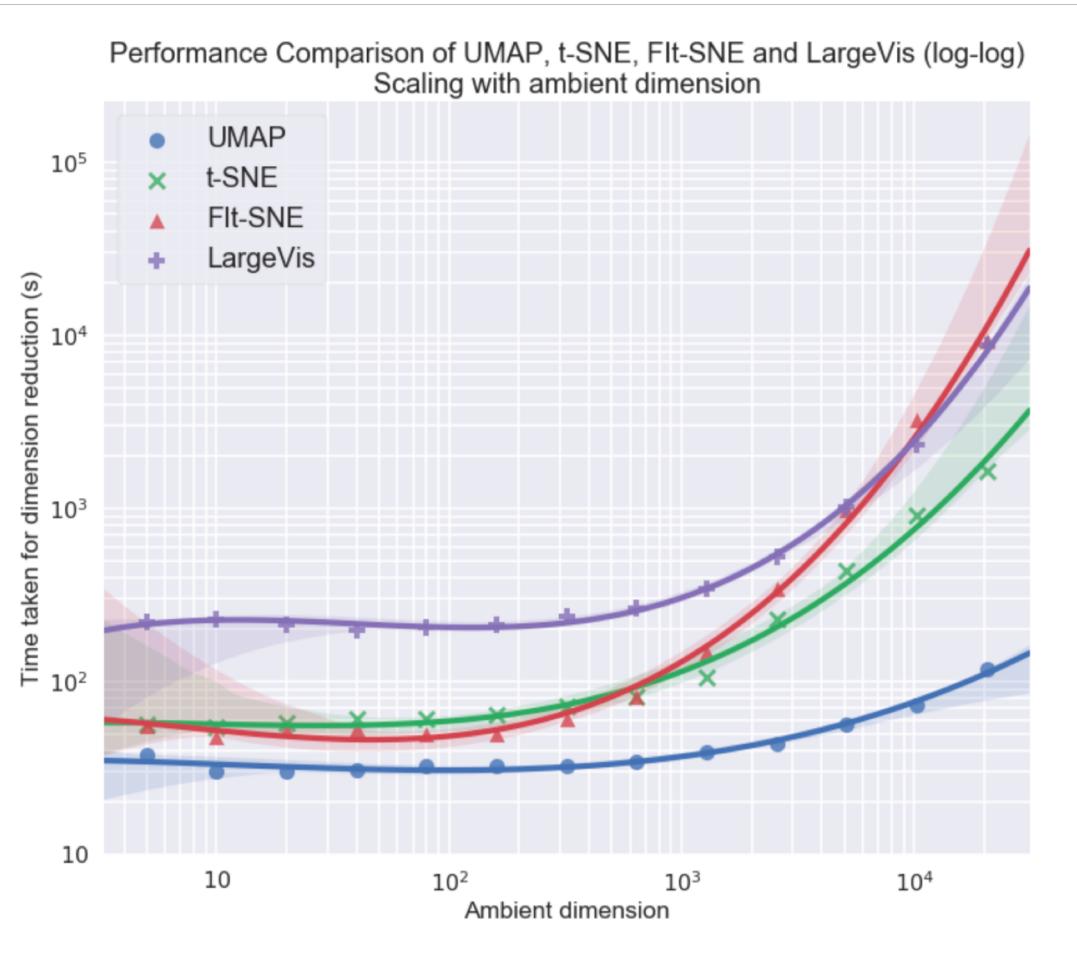
Можно использовать для обучения с учителем



Сравнение с триплет-нейросетью



Время работы



Сравнение проводилось с tsne реализованным в Sklearn.
UMAP реализован на Numba в 1-поточном варианте.

5 lines (4 sloc) | 55 Bytes

```
1 numpy>=1.13
2 scipy>=0.19
3 scikit-learn>=0.18
4 numba>=0.37
```

Ссылка на статью

- <https://arxiv.org/abs/1802.03426>