Optimal Transport Problem

D.Kuznedelev, D.Chaplygina, D.Biba, L.Kulikov, G.Mqawass

Skoltech

December 16, 2020

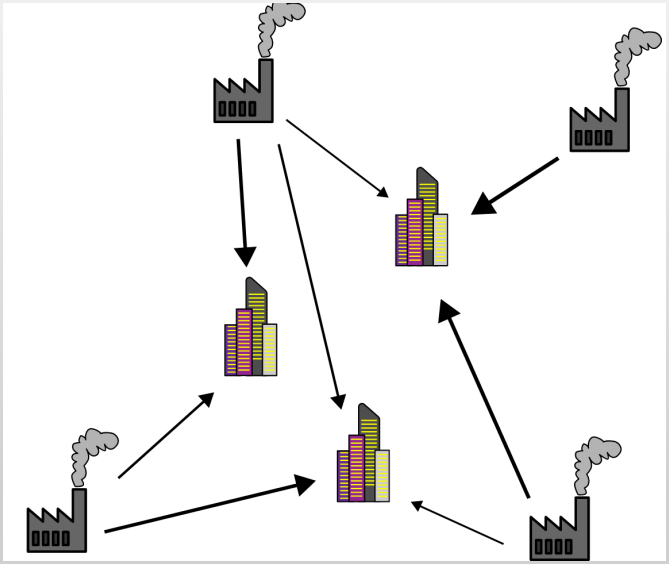
Outline

- Problem statement
- Theoretical foundations
- Numerical approaches
- Results
  - Benchmarking on empirical distributions
  - Investigation of impact of various hyperparameters on the result
- Applications
- Conclusions and further improvements

Problem formulation

Given the locations of the producers and consumers with the known cost of transportation of one unit of mass, make the optimal arrangement between the sources and the targets, that the total cost is as small as possible.

The problem was originally formulated by G.Monge in 1789. The reformulation, allowing the splitting of the masses into pieces was proposed by L.Kantorovich in 1942.



Mathematical setup

Kantorovich formulation

Given:

- source distribution a
- target distribution b
- cost matrix C_{ij} of transportation one unit of mass from i to j

find the optimal assignment, minimizing the total cost

Admissible couplings

$$U(a,b) \overbrace{=}^{\text{def}} \{\mathbf{P} \in \mathbb{R}_+^{n \times m} : \mathbf{P}1_m = a \quad \text{and} \quad \mathbf{P}^T 1_n = b\}$$

Kantorovich optimal transport problem reads:

$$L_C(a,b) \overbrace{=}^{\text{def}} \min_{\mathbf{P} \in U(a,b)} \langle C, P \rangle \overbrace{=}^{\text{def}} \sum_{i,j} P_{i,j} C_{i,j}$$

It is in fact a linear programming problem

Optimal transport before Sinkhorn

Before the invention of Sinkhorn algorithm the problem was solved via **network simplex method**.

Complexity $\mathcal{O}(n^3 \log n)$ arithmetic operations

Need for a faster approach!

Regularization

Add a regularization term to make a problem strictly convex:

$$L_C(a, b) \overbrace{=}^{\text{def}} \min_{\mathbf{P} \in U(a,b)} \langle C, P \rangle - \varepsilon H(P)$$

Where the entropy of the coupling matrix is defined by:

$$H(P) \overbrace{=}^{\text{def}} -\sum_{i,j} P_{i,j}(\log(P_{i,j}) - 1)$$

Introducing the Lagrange multipliers f and g, one finally gets a following optimization problem:

$$\mathcal{E}(P, f, g) = \langle C, P \rangle - \varepsilon H(P) - \langle f, \mathbf{P} 1_m - a \rangle - \langle g, \mathbf{P}^T 1_n - b \rangle$$

And the stationary point of the functional is:

$$\frac{\partial \mathcal{E}(P, f, g)}{\partial P_{i,j}} = C_{i,j} + \varepsilon \log(P_{i,j}) - f_i - g_j \Rightarrow P_{i,j} = e^{f_i/\varepsilon} e^{-C_{i,j}/\varepsilon} e^{g_j/\varepsilon}$$

Sinkhorn iterations

Introduce a following notation:

$$u_i = e^{f_i/\varepsilon} \quad v = e^{g_j/\varepsilon} \quad K_{i,j} = e^{-c_{i,j}/\varepsilon}$$

Constraints on the u, v, arising from the form of marginal distributions have the form:

$$u \odot (Kv) = a \qquad v \odot (K^T u) = b$$

These equations are handled by the iterative approach

$$u^{(l+1)} = \frac{a}{Kv^{(l)}} \qquad v^{(l+1)} = \frac{b}{K^T u^{(l+1)}}$$

Complexity of the algorithm [D., Gasnikov, Kroshnin, 2018]

Algorithm outputs optimal $P_{i,j}$ in $\mathcal{O}\left(\frac{n^2 \|C\|_\infty \log n}{\varepsilon^2}\right)$ arithmetic operations

Numerical considerations

- Can we do even better?
- The most computationally demanding operation is matvec $\mathcal{O}(n^2)$.
- For small enough ε couplings for separated points would be exponentially suppressed.
- In the best case we can reduce matvecs complexity to $\mathcal{O}(n)$.

Sparsification

Eliminate all elements $K_{i,j}$ below a certain threshold θ or keep in each column and row n-largest elements.

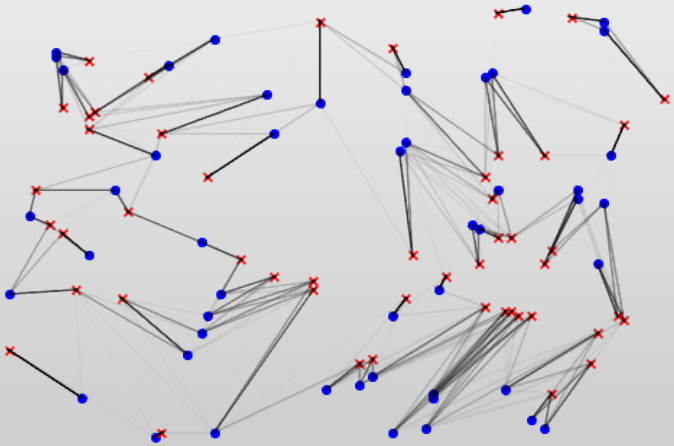
Multigrid approach

Clusterize adjacent points, and solve the problem on the coarse grid. Then perform a refinement on the cluster matching on previous step.

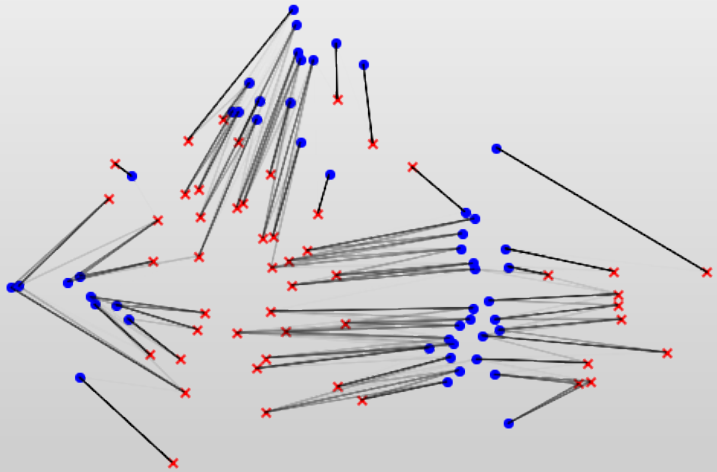
However, the obtained solution is not an exact solution of the initial problem and the sparsification strategy needs to be chosen carefully in order to avoid numerical instabilities.

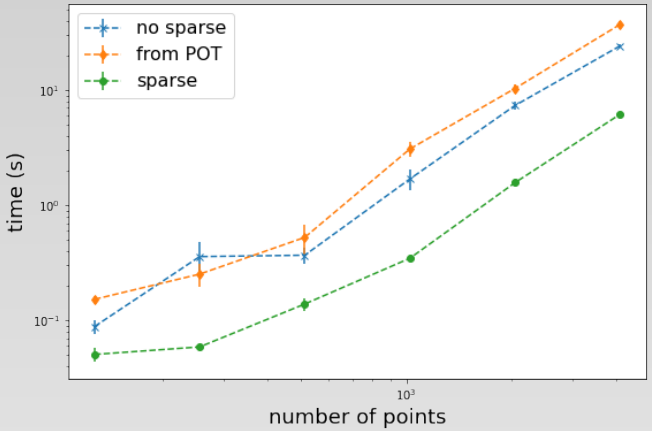
Experiments on generated distributions

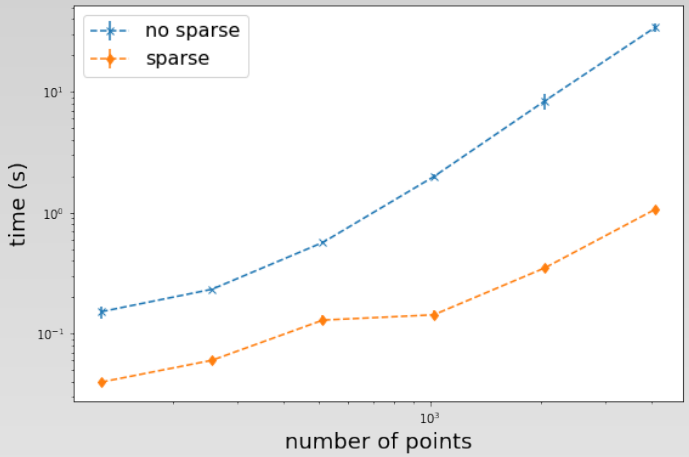
Uniform sampling



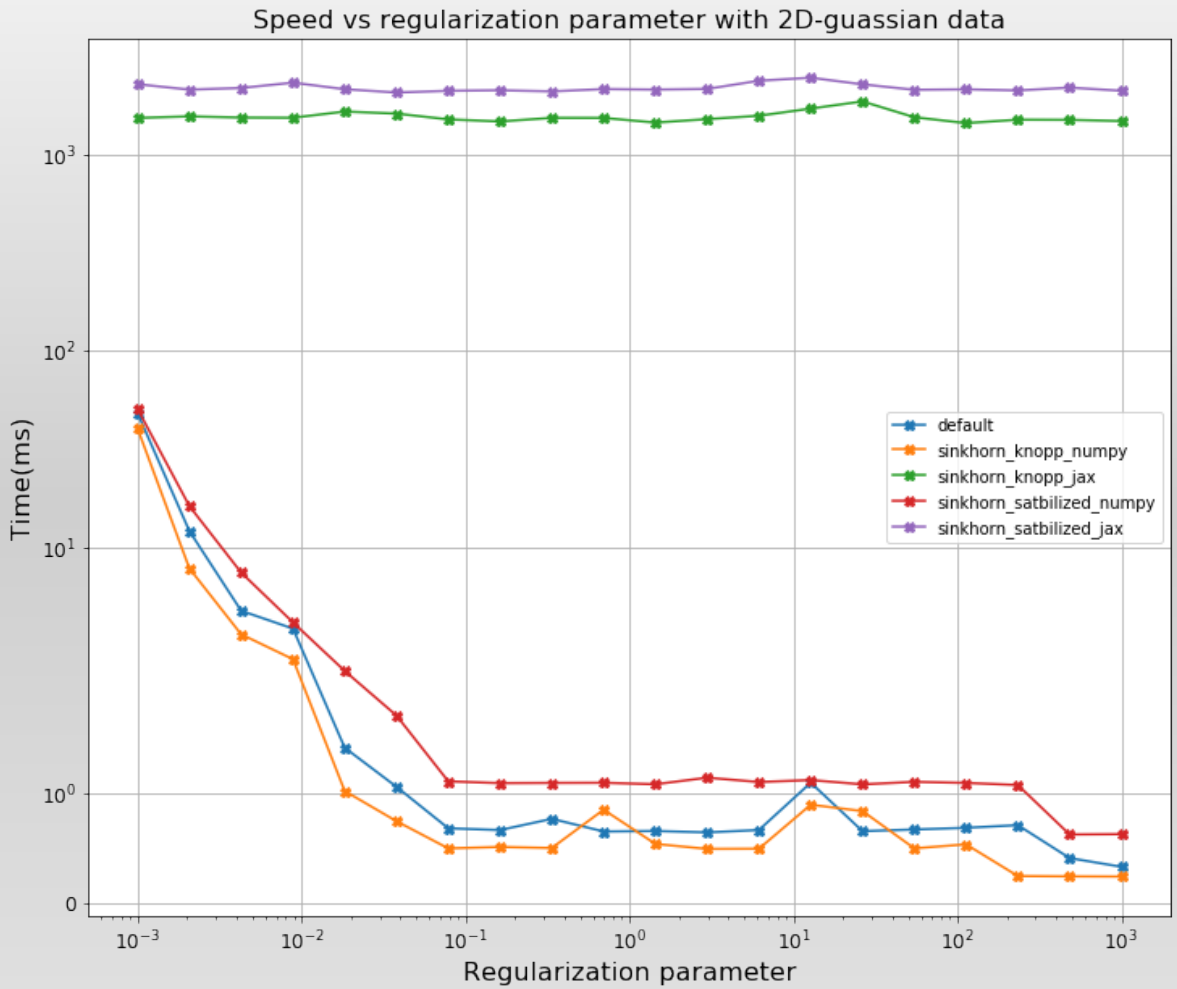
Mixture of gaussians





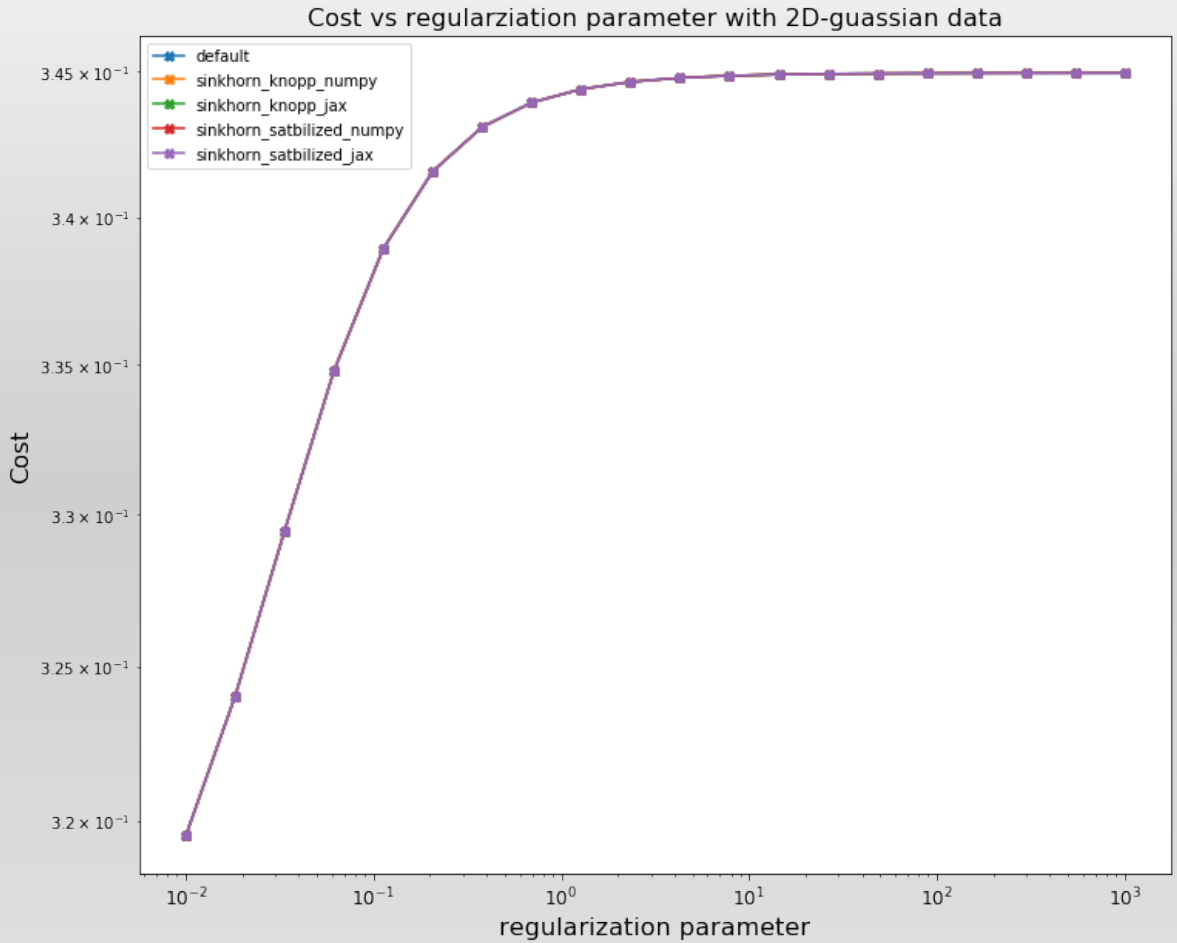


Two gaussians



Dependence of the time elapsed on the regularization parameter

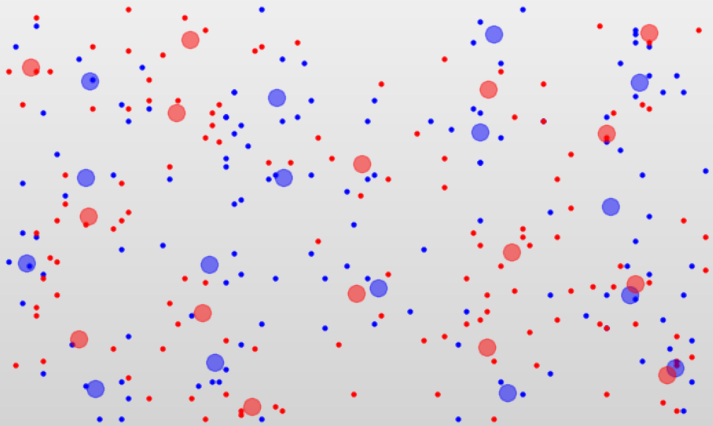
Two gaussians



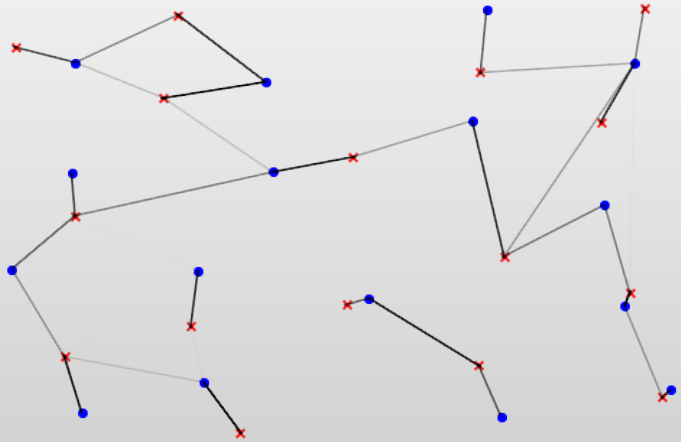
Dependence of the resulting cost on the regularization parameter

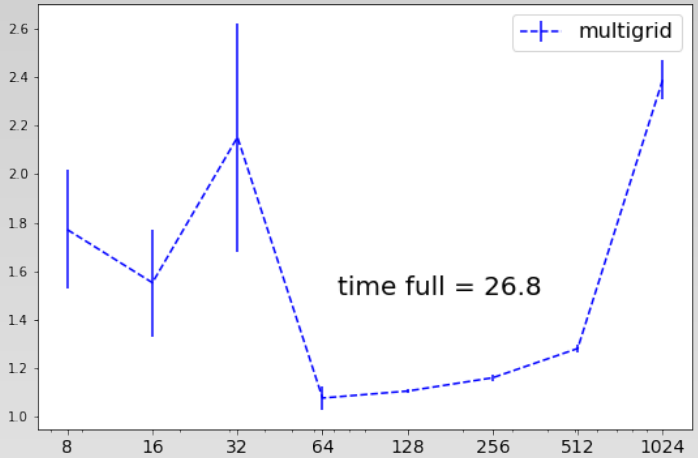
Multigrid

Clustering of points



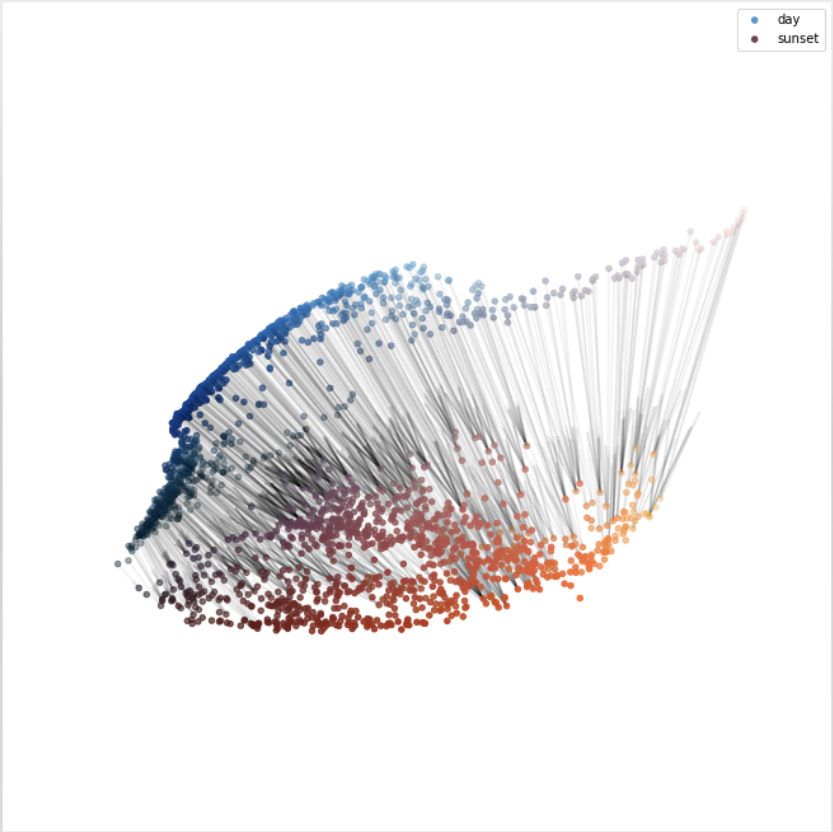
Matching on the coarse grid





Time for various number of clusters on 4096 points

Pixel matching

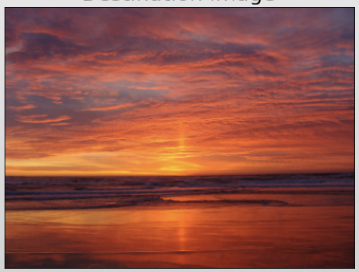


Randomly sampled points from the daylight and sunset images and the optimal matching between the pixels

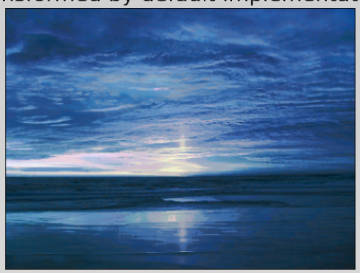
Pixel matching

Sourse image

Destination image



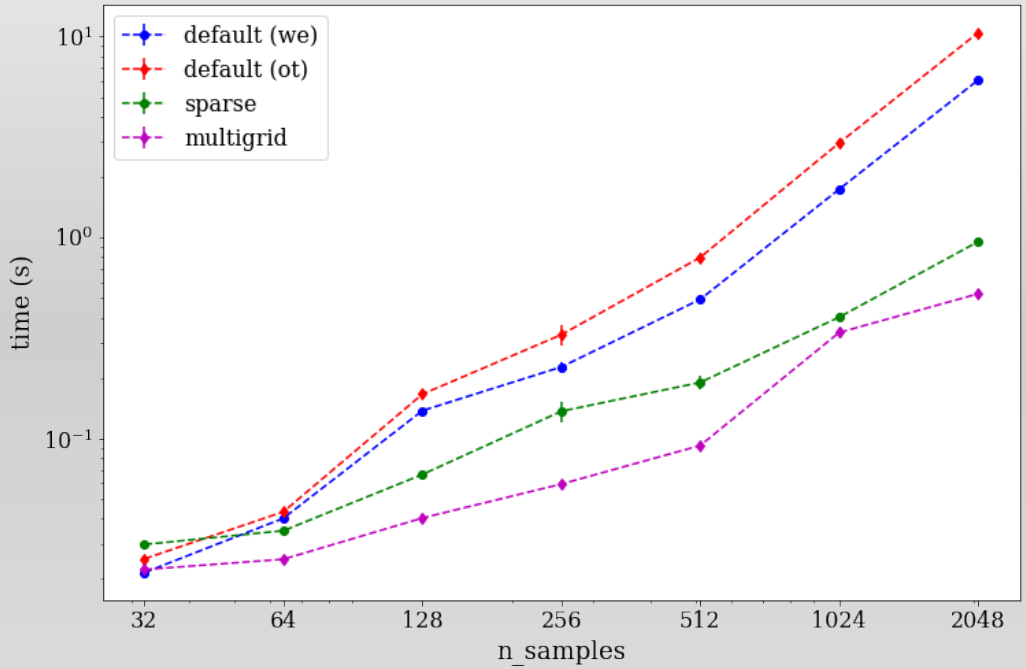
Transformed by default implementation



Transformed by custom implementation

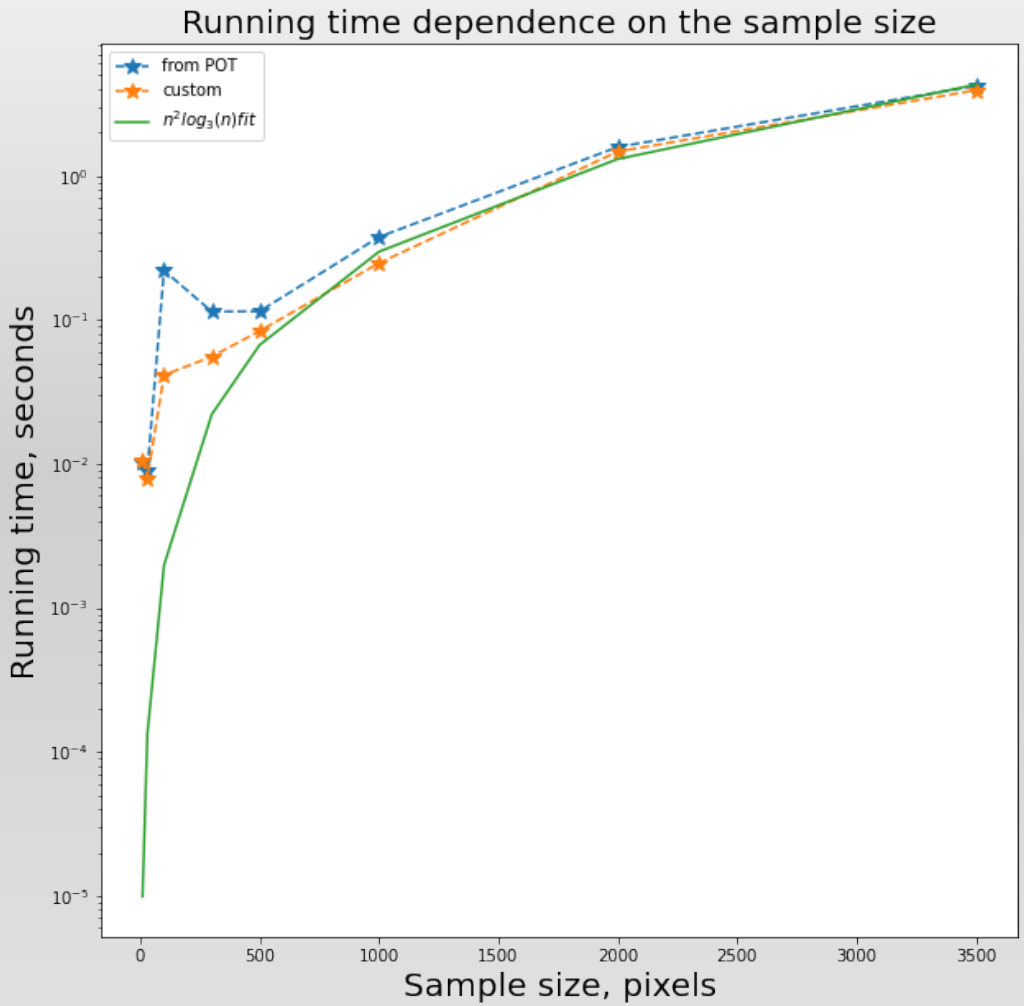
The quality of color transfer by Sinkhorn algorithm implementation from POT (left) is the same as by our implementation (right)

Pixel matching



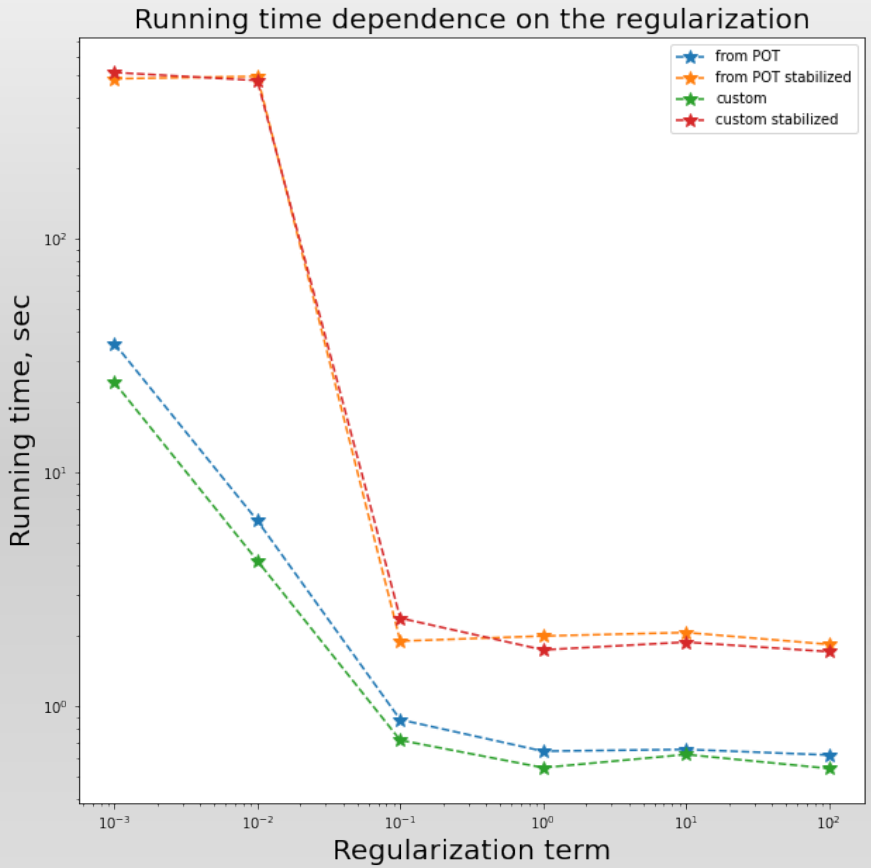
Time for pixel transfer from one image to another

Pixel matching



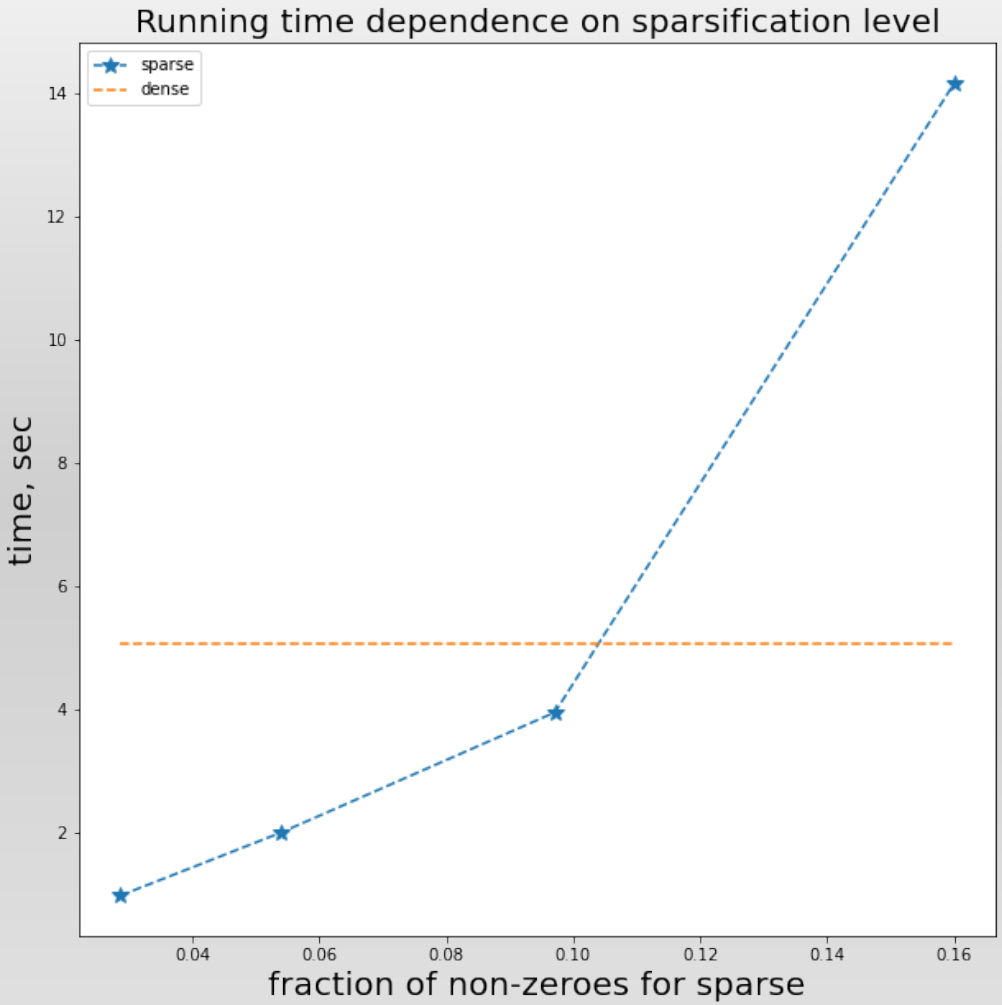
Running time is in agreement with $\mathcal{O}(n^2 \log n)$

Pixel matching



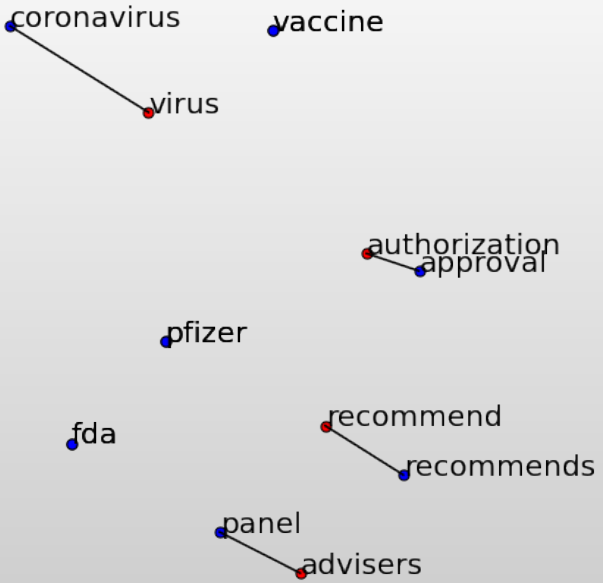
Both stabilized versions run longer than not stabilized, but inside categories (stab/not stab) our implementation performs slightly better

Pixel matching

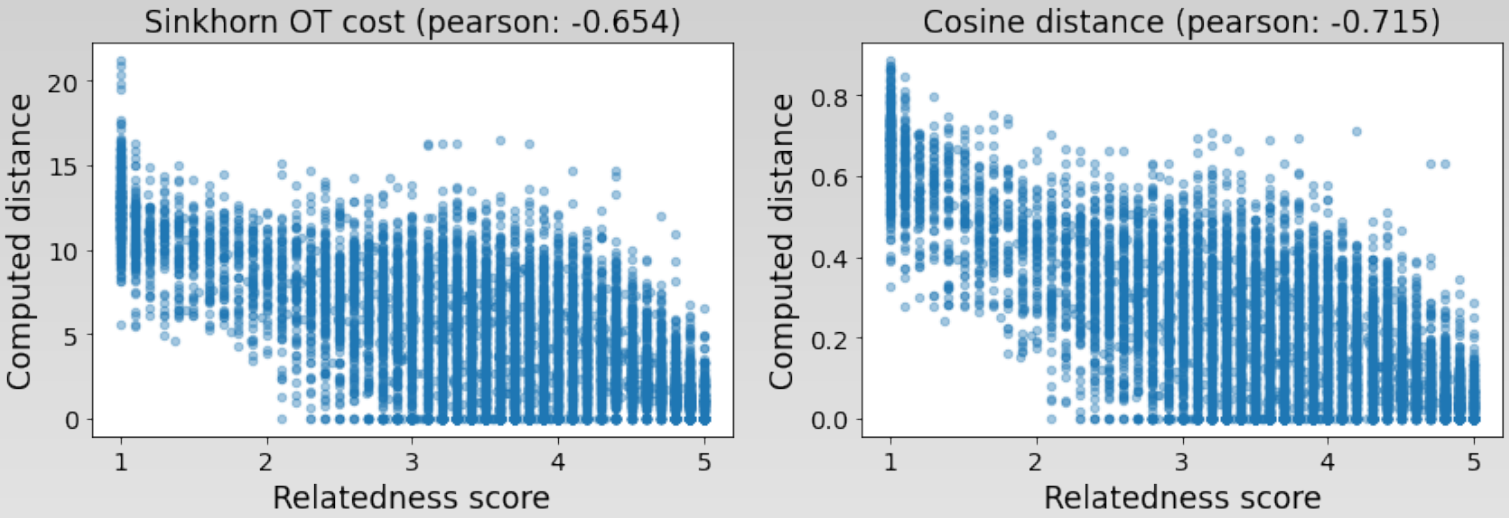


Sparsification gives advantage in running time for high sparsification levels, but not for low.

Applications in text transfer

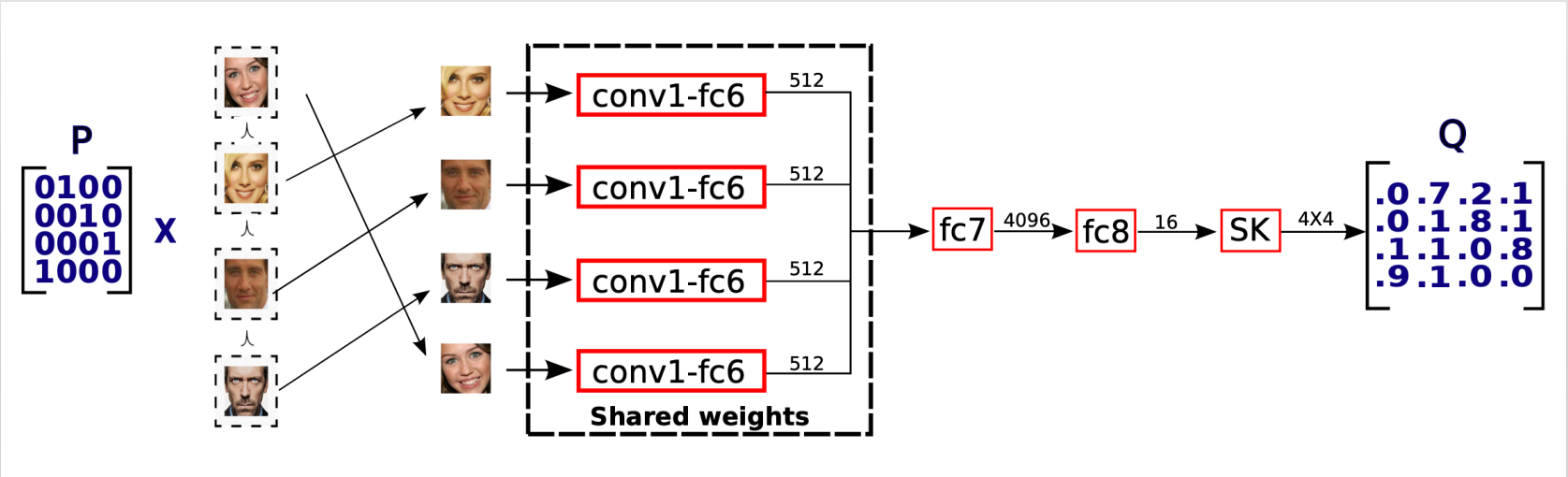


”FDA advisers recommend authorization of Pfizer virus vaccine”
”FDA panel recommends approval of Pfizer’s coronavirus vaccine”



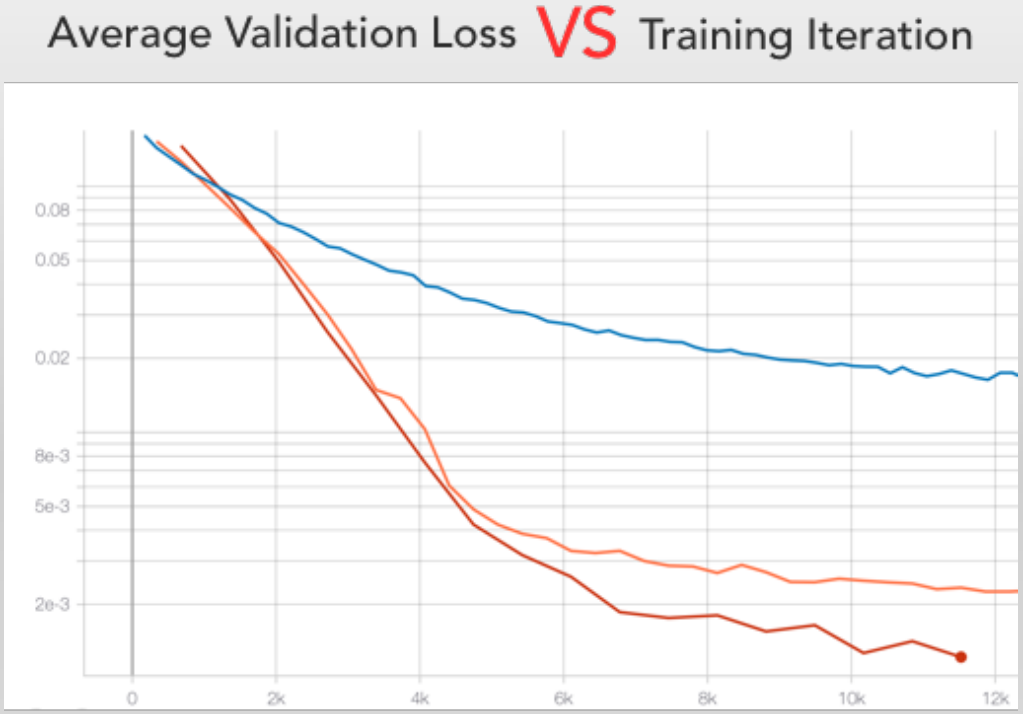
OT cost compared to cosine distance

Visual Permutation Learning: Improvement of DeepPermNet



Architecture of PermNet on high level

Visual Permutation Learning: Improvement of DeepPermNet



	Name	Smoothed	Value	Step
●	v1_original	0.01728	0.01728	11.56k
●	v1_stable	2.3393e-3	2.3393e-3	11.53k
●	v5_stable	1.2221e-3	1.2221e-3	11.53k

Conclusions and the further improvements

Achieved results

- Noticeable speedup with the sparsification strategies
- Successful performance on emprical distributions and pixel data
- Decent quality on transfer problems without long training and large datasets

Cons of the algorithm and issues

- Regularization parameter ε needs to be carefully tuned
- Pruning of the elements is successful provided there are distances of different magnitude
- Stabilized version allows for smaller ε but for the cost of the computation time increase
- Single precision computations (`jnp.float32`) more vulnerable to failure

Conclusions and the further improvements

Further improvements

- Algorithm can be implemented on highly parallel architectures (GPU's)
- Educated adjustment of the parameters
- Multilevel multigrid approach for large scale problems