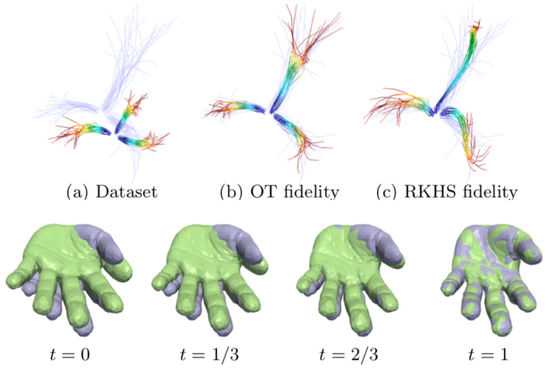
Massively scalable Sinkhorn distances via the Nyström method

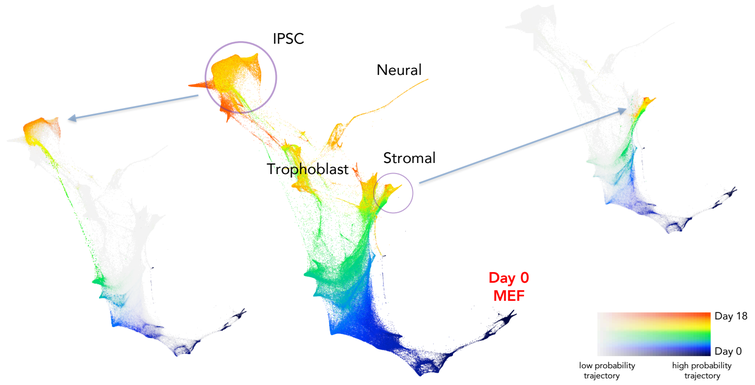
J. Altschuler, F. Bach, J. Niles-Weed, A. Rudi

Optimal Transport

Kusner et al. (2015)



Feydy et al. (2017)

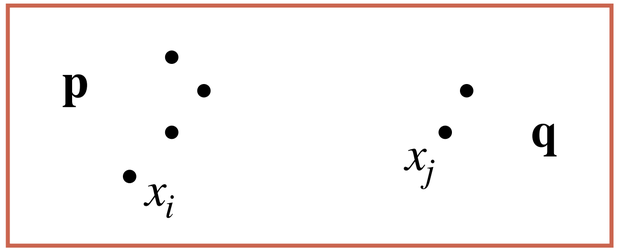


Schiebinger et al. (2018)

Gulrajani et al. (2017)

Increasingly popular tool in machine learning

Definitions



$x_1, \ldots, x_n \in \mathbb{R}^d$

$\|x_i\| \leq R \quad \forall i$

$\mathbf{p}, \mathbf{q} \in \Delta_n$

two probability distributions on n points in $\mathbb{R}^d$.

$$\mathscr{M}(\mathbf{p}, \mathbf{q}) := \left\{ P : \begin{array}{c} P \in \mathbb{R}_+^{n \times n} \\ P\mathbf{1} = \mathbf{p} \\ P^\top \mathbf{1} = \mathbf{q} \end{array} \right\}$$

set of *couplings* between $\mathbf{p}$ and $\mathbf{q}$

$$W(\mathbf{p}, \mathbf{q}) := \min_{P \in \mathscr{M}(\mathbf{p}, \mathbf{q})} \sum_{i,j} P_{ij} \|x_i - x_j\|^2$$

Wasserstein distance between $\mathbf{p}$ and $\mathbf{q}$

Slow to compute

Sinkhorn distance

$$W(\mathbf{p}, \mathbf{q}) := \min_{P \in \mathscr{M}(\mathbf{p},\mathbf{q})} \sum_{i,j} P_{ij} \|x_i - x_j\|^2$$

Wasserstein distance between $\mathbf{p}$ and $\mathbf{q}$

Slow to compute — $\tilde{O}(n^3)$ time

$$W_\eta(\mathbf{p}, \mathbf{q}) := \min_{P \in \mathscr{M}(\mathbf{p},\mathbf{q})} \sum_{i,j} P_{ij} \|x_i - x_j\|^2 - \eta^{-1} H(P)$$

$$H(P) = \sum_{ij} P_{ij} \log \frac{1}{P_{ij}}$$

[Cuturi '13] : **Sinkhorn "distance"**

Faster to compute — $\tilde{O}(n^2)$ time

Can we do better?

Our contribution

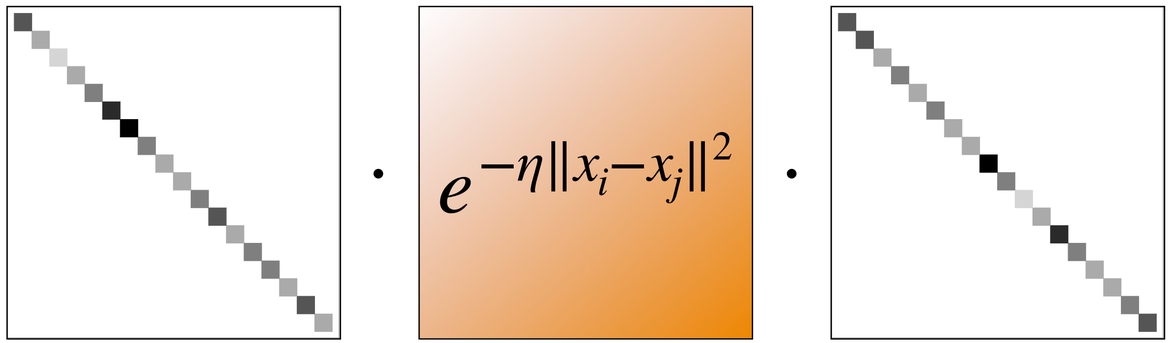
New algorithm (NYS-SINK) approximates Sinkhorn distance in $\tilde{O}(n)$ **time and space**

Guarantees automatically **adaptive** to low-dimensional structure

Outperforms standard approaches by **orders of magnitude**

Why Sinkhorn distance?

Minimizer: **unique** matrix in $\mathscr{M}_{\mathbf{p},\mathbf{q}}$ of form



$\in \mathscr{M}_{\mathbf{p},\mathbf{q}}$

entrywise exponential

positive diagonal matrices

Sinkhorn scaling

Easy iterative algorithm to find minimizer

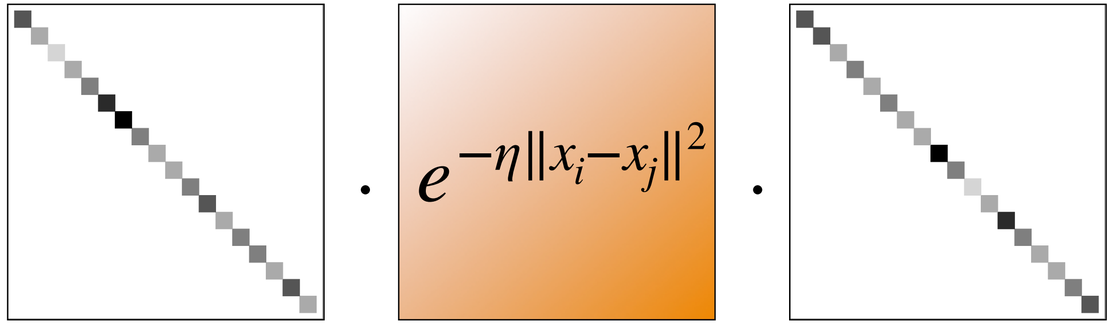
Goal: find (unique) $P^\eta = D_1 e^{-\eta \|x_i - x_j\|^2} D_2 \in \mathscr{M}_{\mathbf{p},\mathbf{q}}$

1. Rescale rows to match $\mathbf{p}$
2. Rescale columns to match $\mathbf{q}$
3. Repeat

Converges! [Sinkhorn ’67]

… in $\tilde{O}(n^2 \epsilon^{-2})$ time! [Altschuler, Weed, Rigollet ’17]

Prior analysis [AWR '17]



$\tilde{O}(1)$ iterations

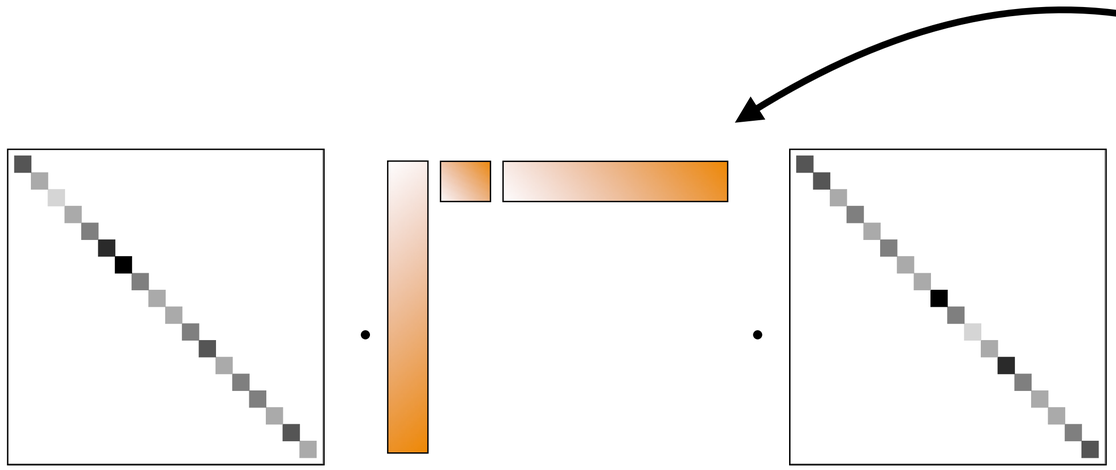
x

$O(n^2)$ per iteration (matrix-vector product)

=

$\boxed{\tilde{O}(n^2)}$ time

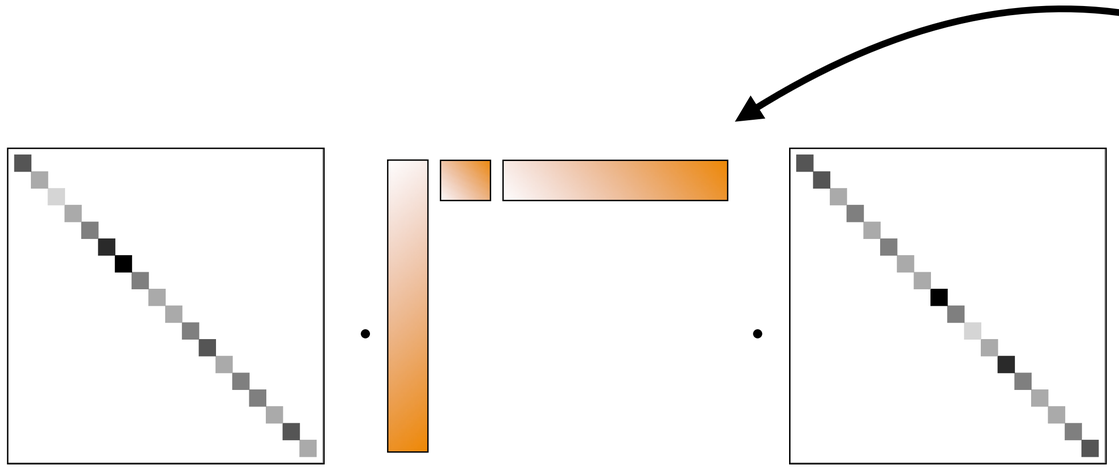
Idea #1



low rank approximation
(via Nyström method)

$\tilde{O}(1)$ iterations **x** $O(n)$ per iteration (matrix-vector product) **=** $\boxed{\tilde{O}(n)}$ time

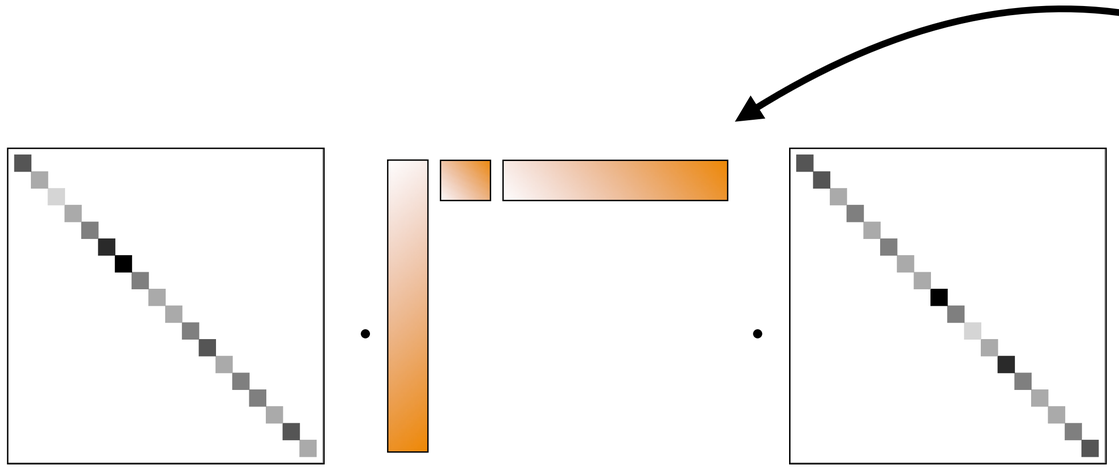
Idea #1



low rank approximation
(via Nyström method)

Rigorous analysis needs **new stability guarantees**
for Sinkhorn scaling

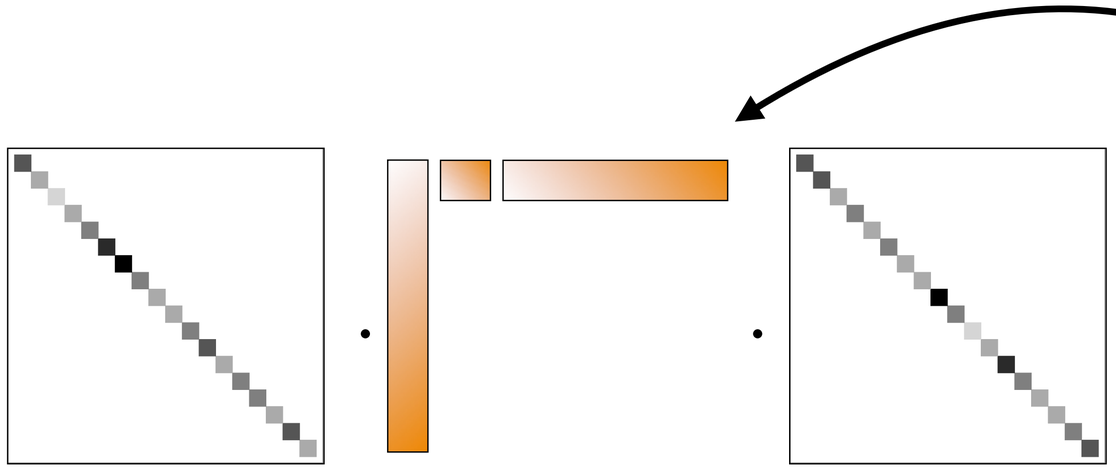
Idea #1



low rank approximation
(via Nyström method)

Works well, but error guarantee depends on **ambient** dimension (could be large)

Idea #2

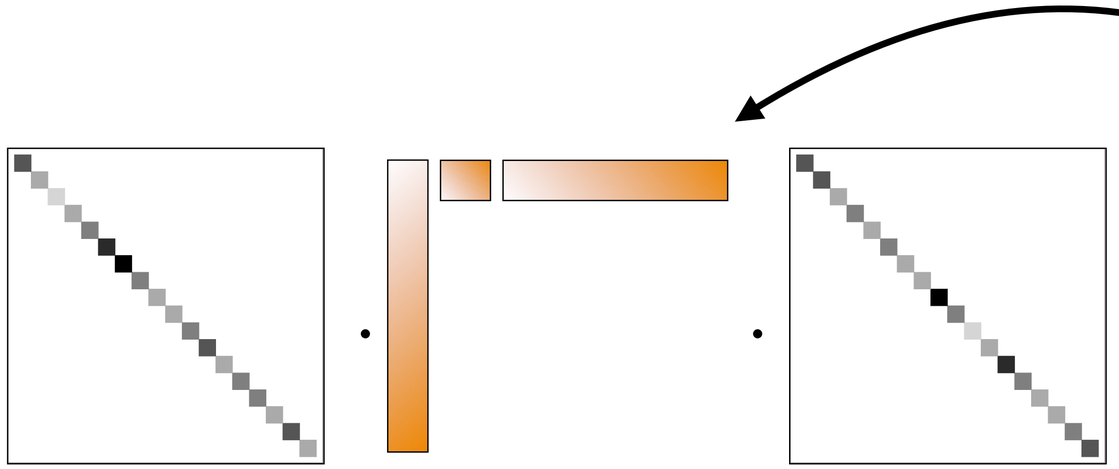


low rank approximation
(via Nyström method)

Works well, but error guarantee depends on **ambient** dimension (could be large)

Choose rank **adaptively** and pay only for intrinsic dimension

Idea #2



low rank approximation
(via Nyström method)

Rigorous analysis needs **new interpolation guarantees** for Gaussian kernel

Experiments

Experiment 1: $n \approx 3 \times 10^5$	W_η	time (s)
Nys-Sink ($r = 2000, T = 20$)	0.087 ± 0.008	0.4 ± 0.1
Dual-Sink + Annealing ($\alpha = 0.95$)	0.087	35.4
Dual-Sink Multiscale + Annealing ($\alpha = 0.95$)	0.090	3.4

Experiment 2: $n \approx 3.8 \times 10^6$	W_η	time (s)
Nys-Sink ($r = 2000, T = 20$)	0.11 ± 0.01	6.3 ± 0.8
Dual-Sink + Annealing ($\alpha = 0.95$)	0.10	1168
Dual-Sink Multiscale + Annealing ($\alpha = 0.95$)	0.11	103.6

Comparison with existing approaches

(r : approximation rank, T : iterations)

Experiments

Experiment 1: $n \approx 3 \times 10^5$	W_η	time (s)
Nys-Sink $(r = 2000, T = 20)$	0.087 ± 0.008	0.4 ± 0.1
Dual-Sink + Annealing $(\alpha = 0.95)$	0.087	35.4
Dual-Sink Multiscale + Annealing $(\alpha = 0.95)$	0.090	3.4

Experiment 2: $n \approx 3.8 \times 10^6$	W_η	time (s)
Nys-Sink $(r = 2000, T = 20)$	0.11 ± 0.01	6.3 ± 0.8
Dual-Sink + Annealing $(\alpha = 0.95)$	0.10	1168
Dual-Sink Multiscale + Annealing $(\alpha = 0.95)$	0.11	103.6

comparable results

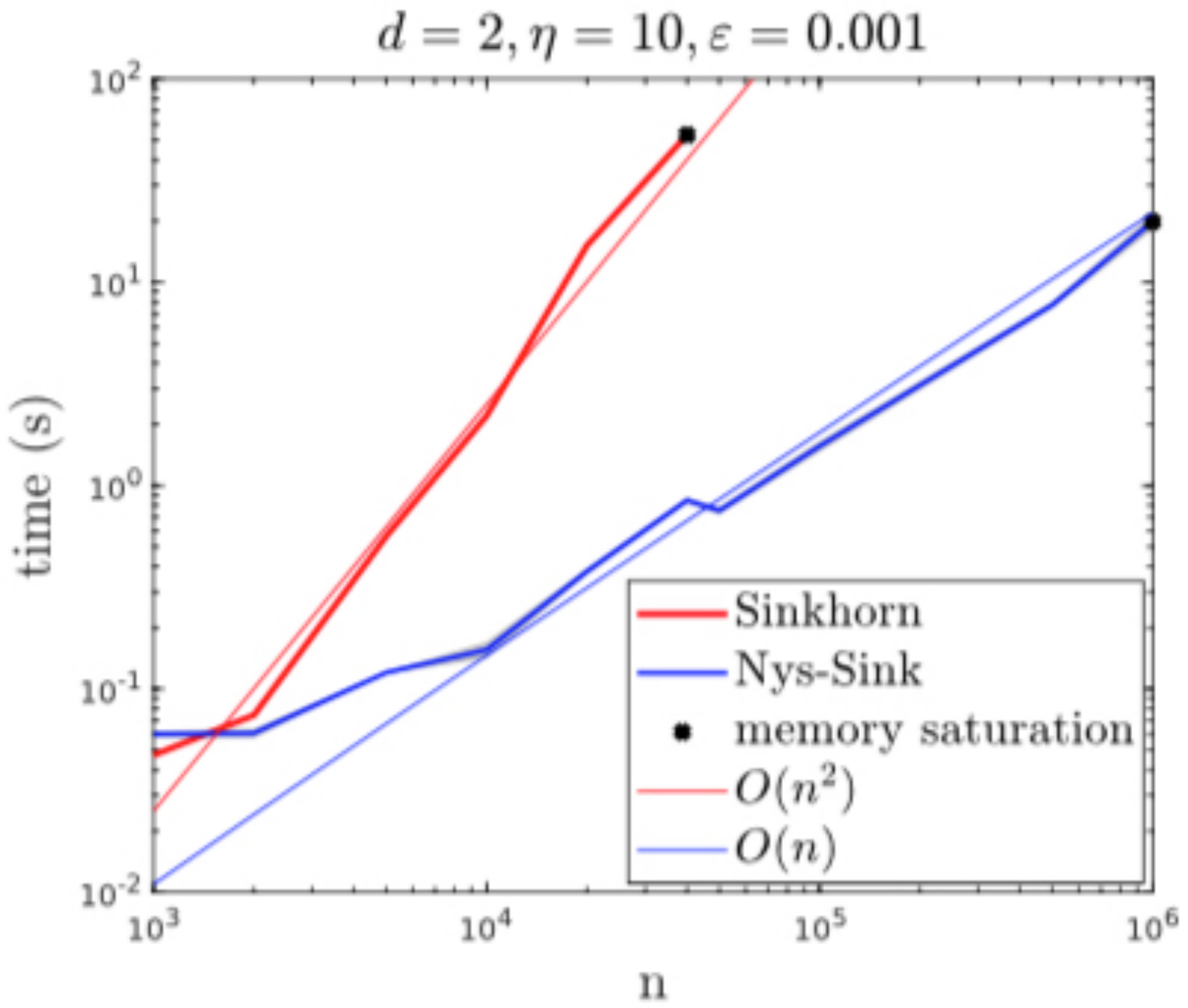
Experiments

Experiment 1: $n \approx 3 \times 10^5$	W_η	time (s)
Nys-Sink $(r = 2000, T = 20)$	0.087 ± 0.008	0.4 ± 0.1
Dual-Sink + Annealing $(\alpha = 0.95)$	0.087	35.4
Dual-Sink Multiscale + Annealing $(\alpha = 0.95)$	0.090	3.4

Experiment 2: $n \approx 3.8 \times 10^6$	W_η	time (s)
Nys-Sink $(r = 2000, T = 20)$	0.11 ± 0.01	6.3 ± 0.8
Dual-Sink + Annealing $(\alpha = 0.95)$	0.10	1168
Dual-Sink Multiscale + Annealing $(\alpha = 0.95)$	0.11	103.6

orders-of-magnitude speedup

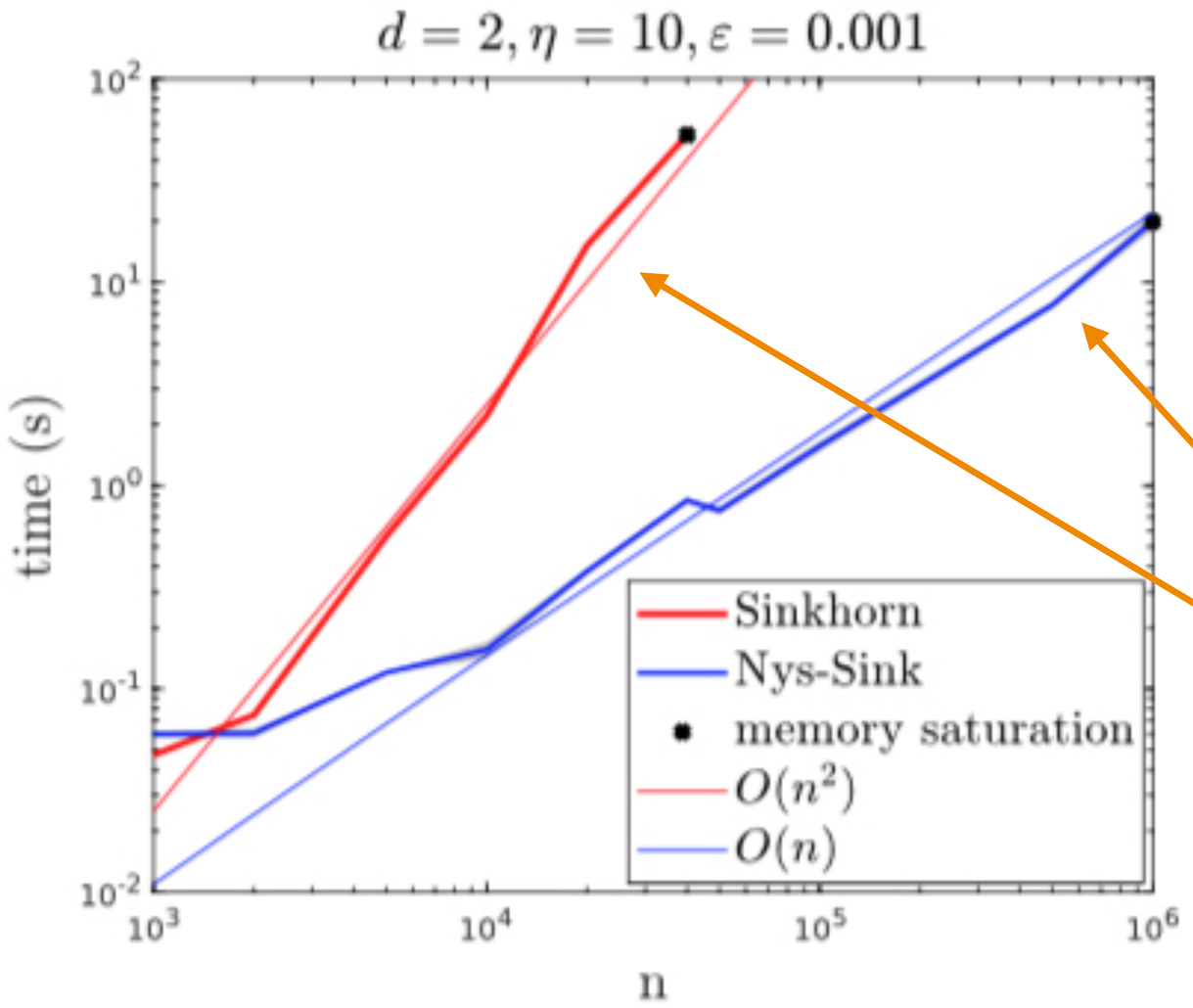
Experiments



Comparison with Sinkhorn scaling

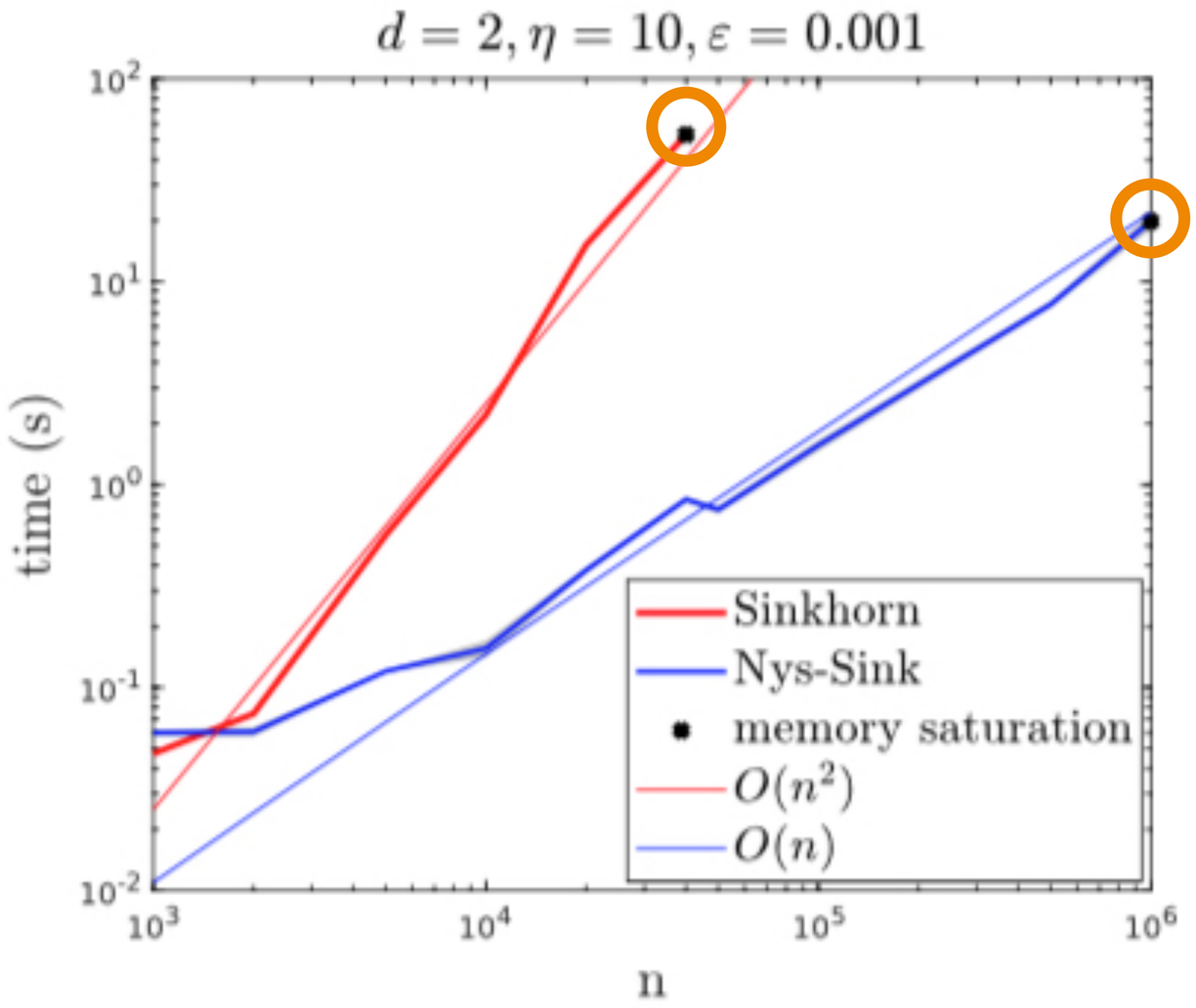
(ε : desired accuracy)

Experiments



linear vs. quadratic runtime

Experiments



much larger problems solvable in memory