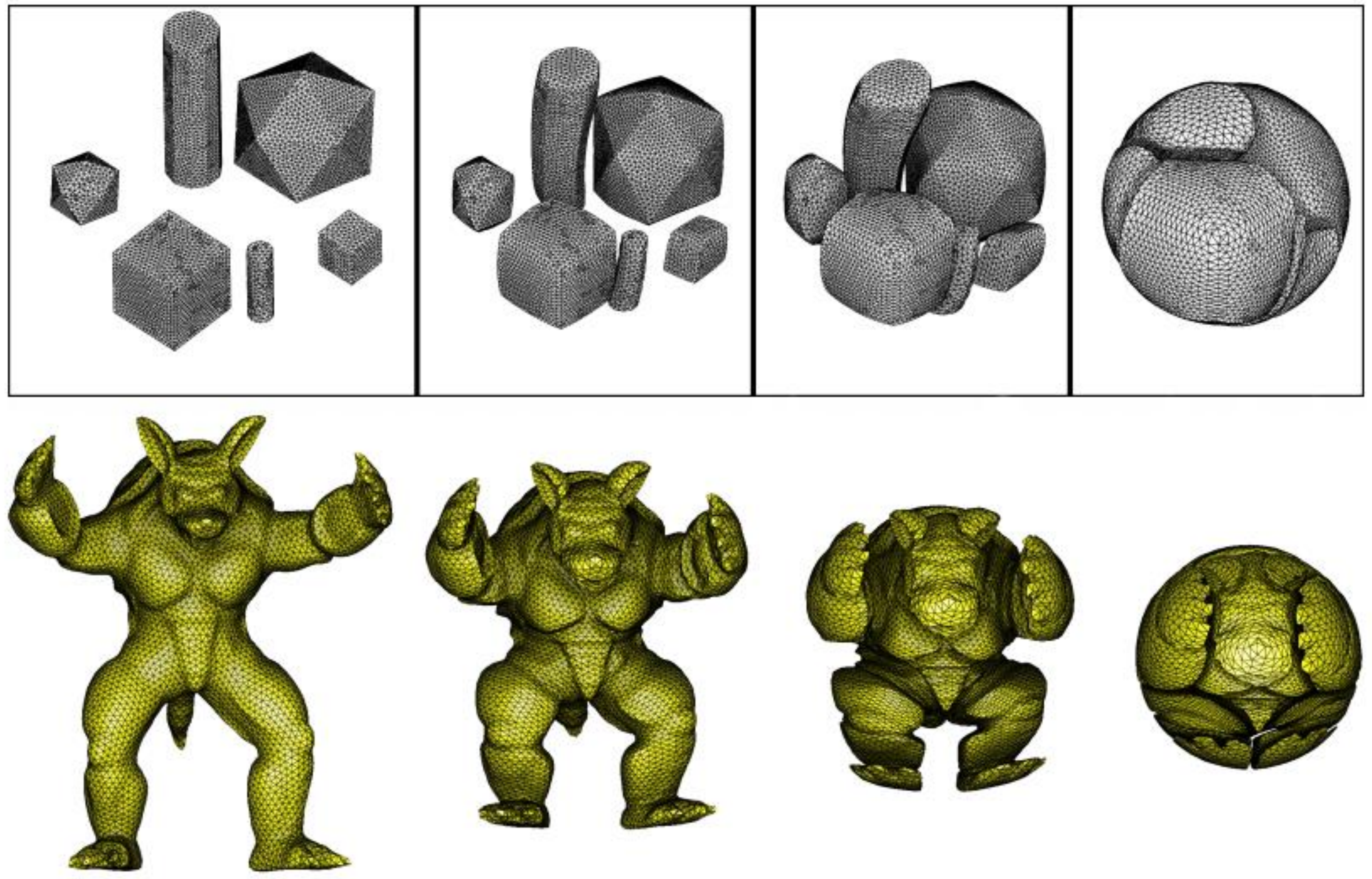
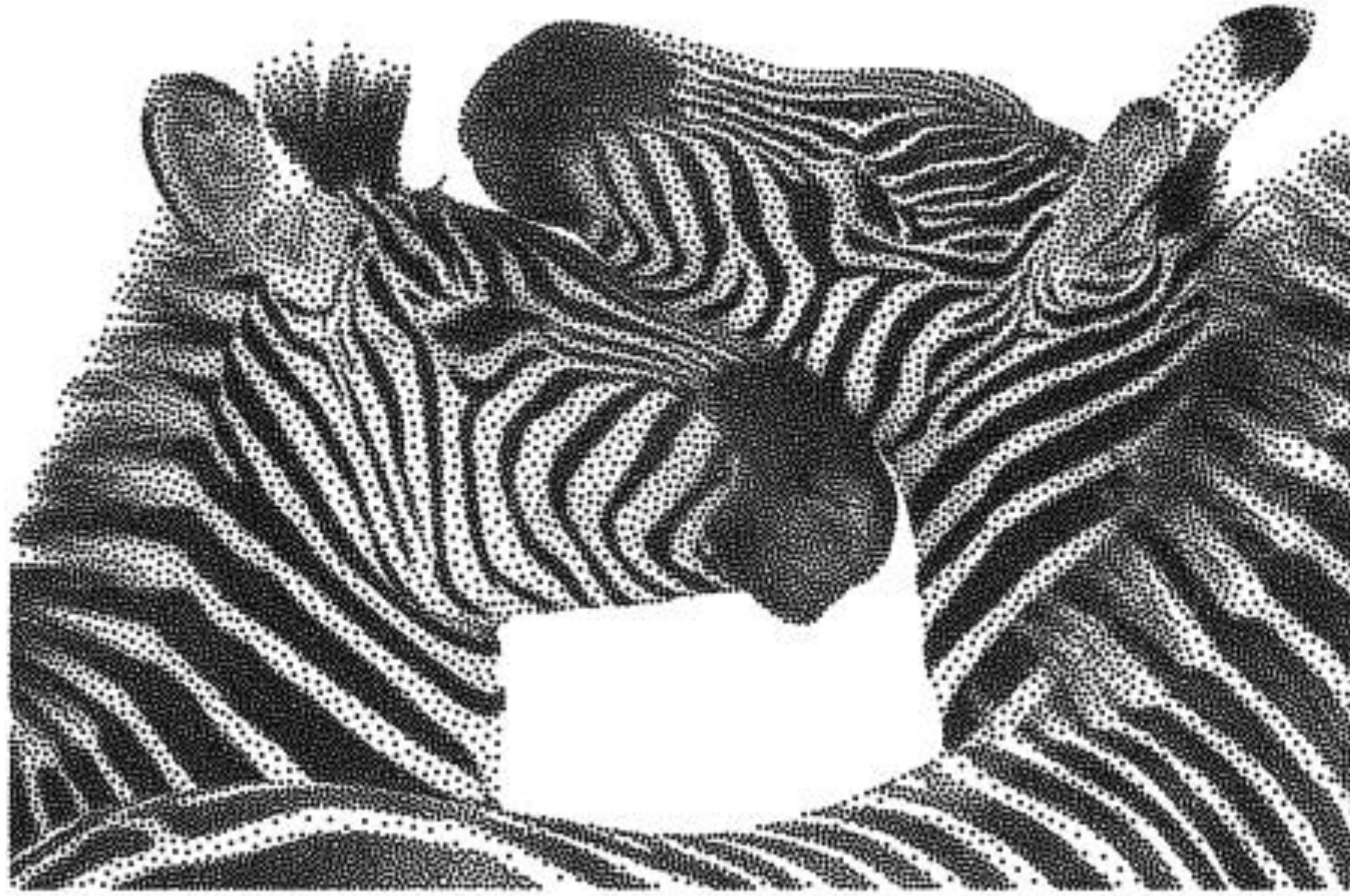
SLICED PARTIAL OPTIMAL TRANSPORT

Nicolas Bonneel*, David Coeurjolly*

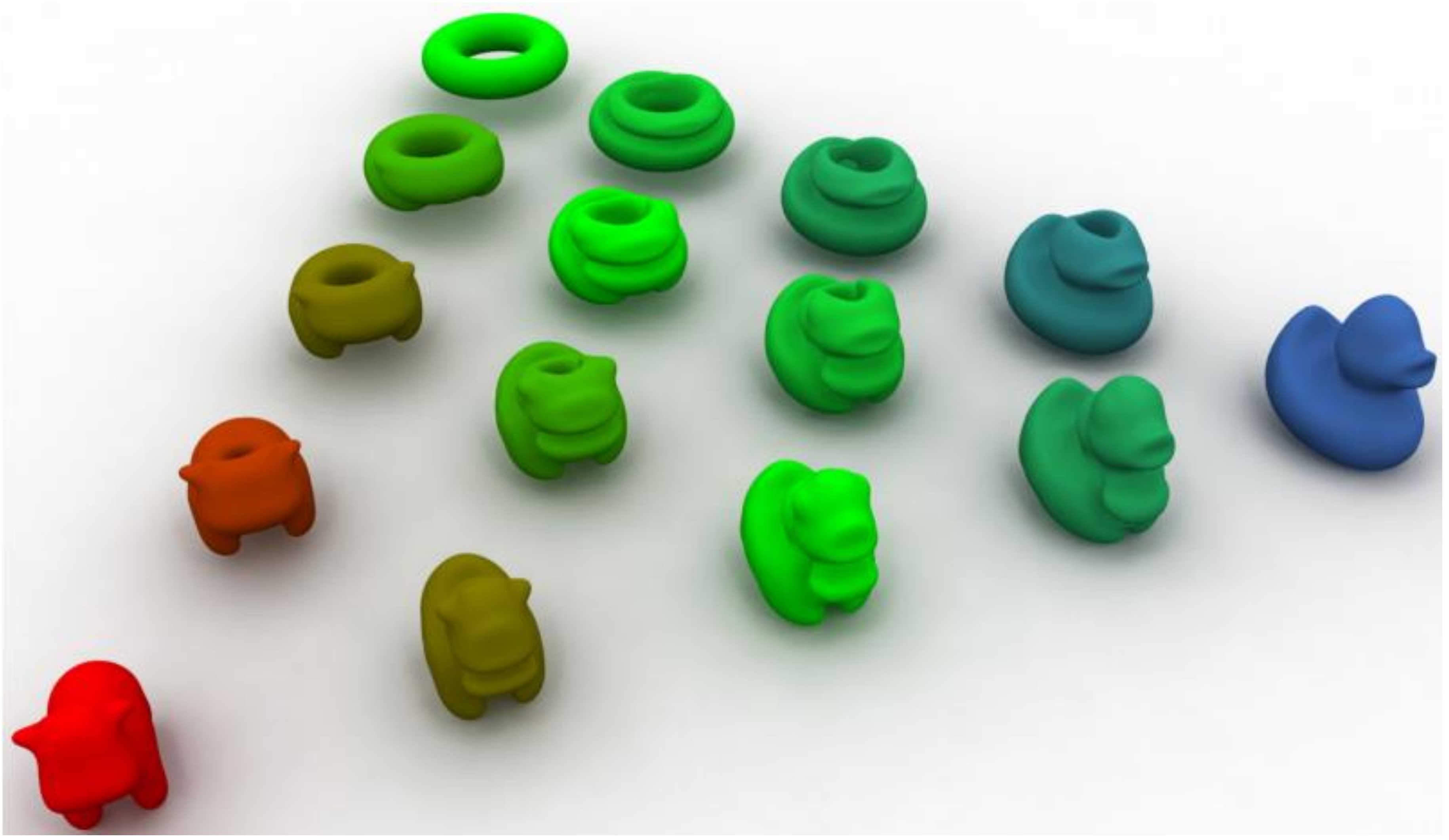
*CNRS, Univ. Lyon.



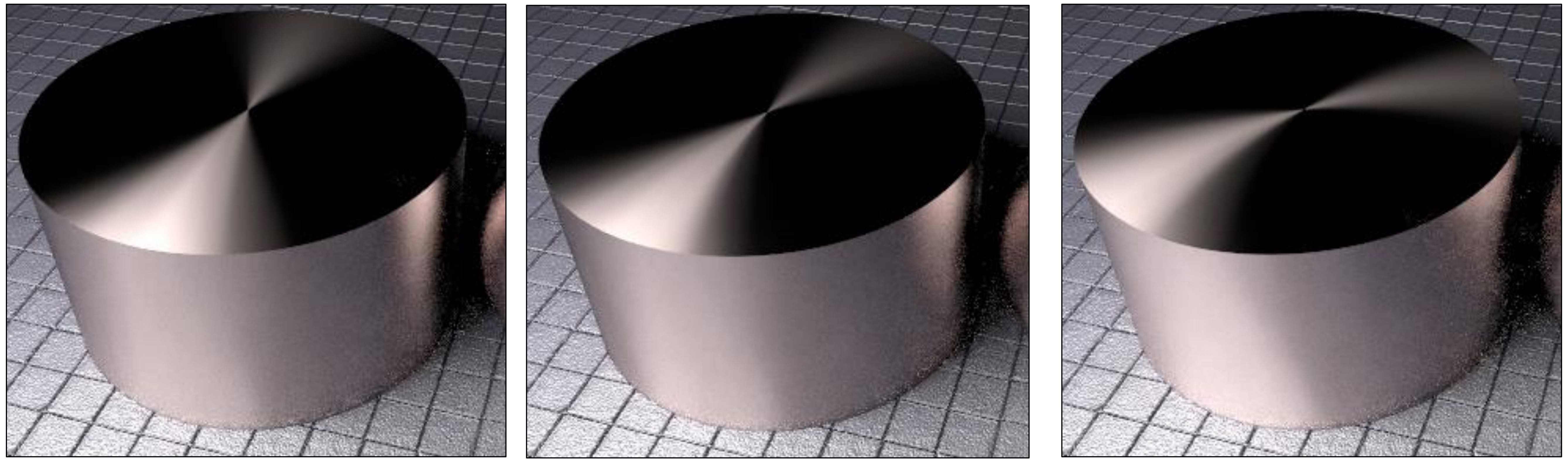
[Levy 2015]



[de Goes et al. 2012]



[Solomon et al. 2015]



[Bonneel et al. 2011]

Matching points with optimal transport

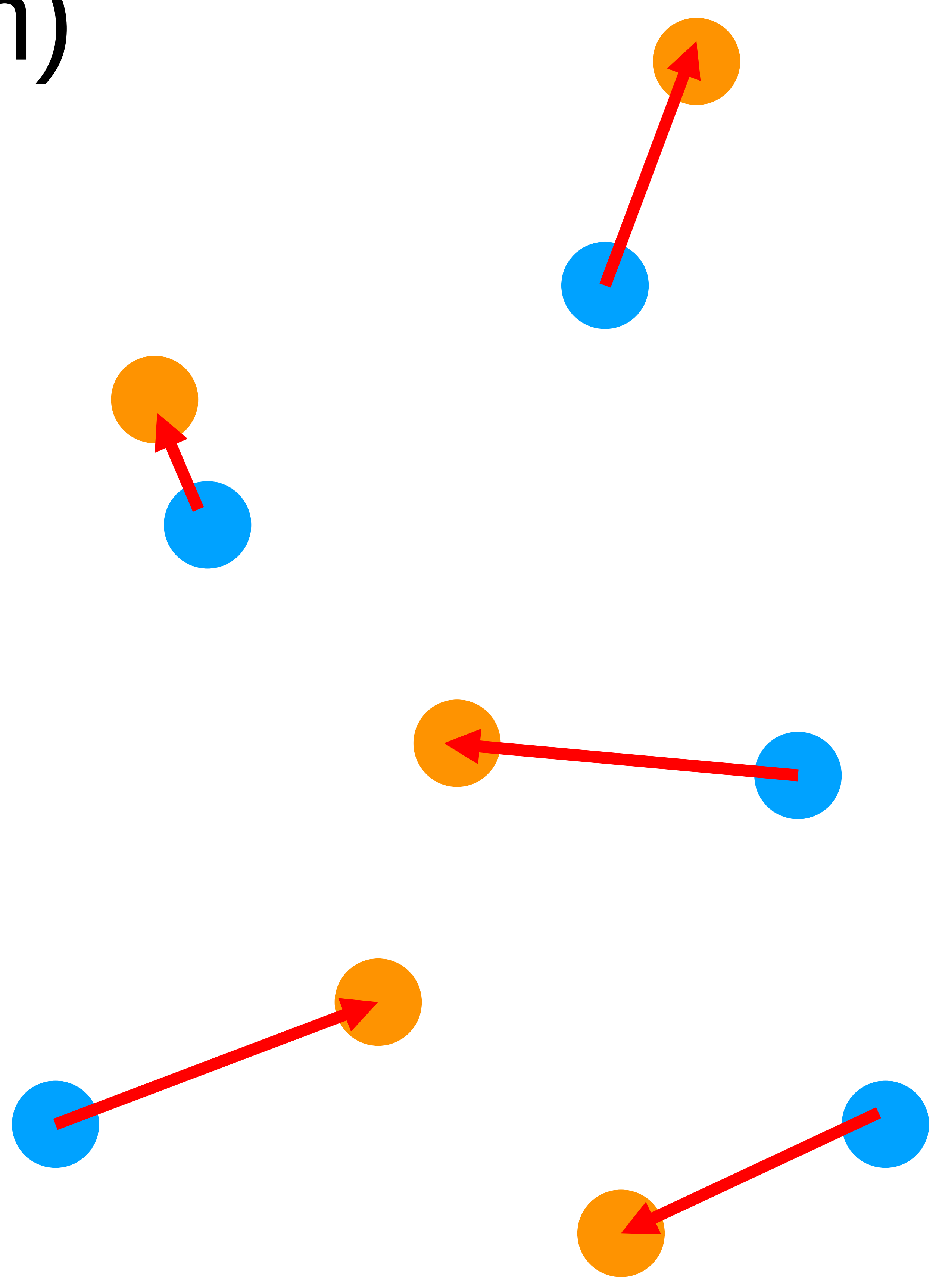
- Monge (Linear Assignment Problem)

$$\min_{T \text{ bijective}} \sum_i c(x_i, y_{T(i)})$$

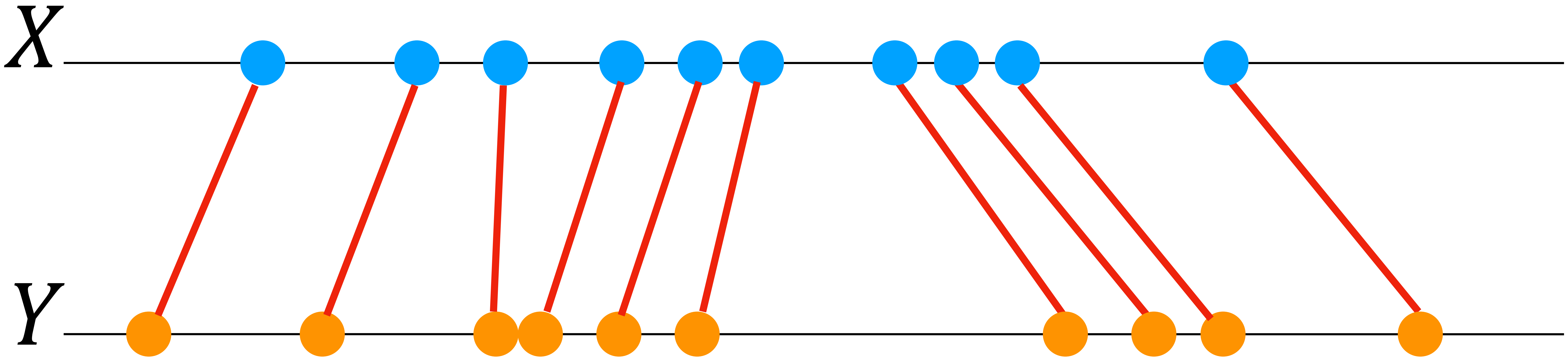
- Kantorovich

$$W(f,g) = \min \sum_{i,j} c_{i,j} \pi_{i,j}$$

s.t.
$$\sum_j \pi_{i,j} = 1$$
$$\sum_i \pi_{i,j} = 1$$
$$\pi_{i,j} \geq 0$$

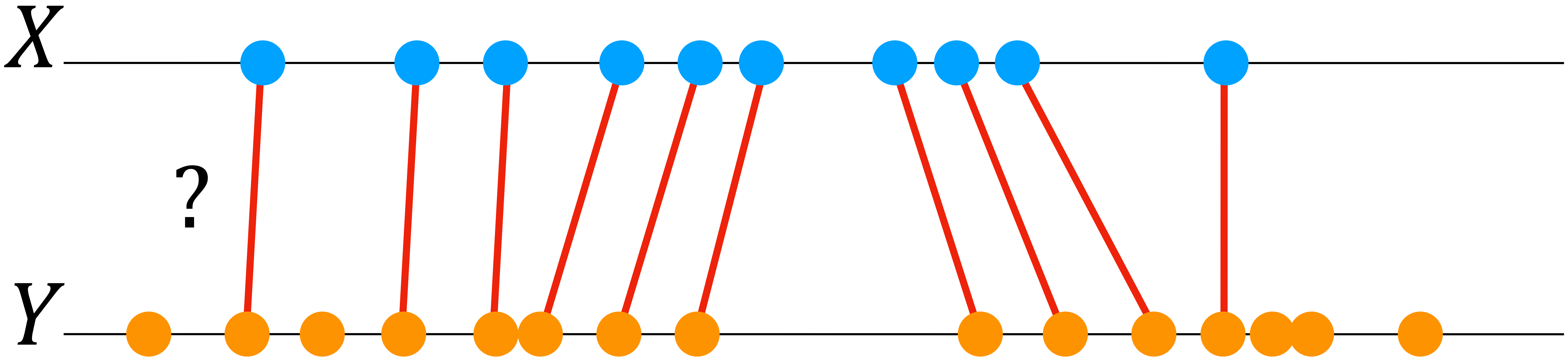


1-d Linear Assignment Problem is trivial*



*assuming the cost c is a convex function of |x-y|

Partial optimal assignment ?

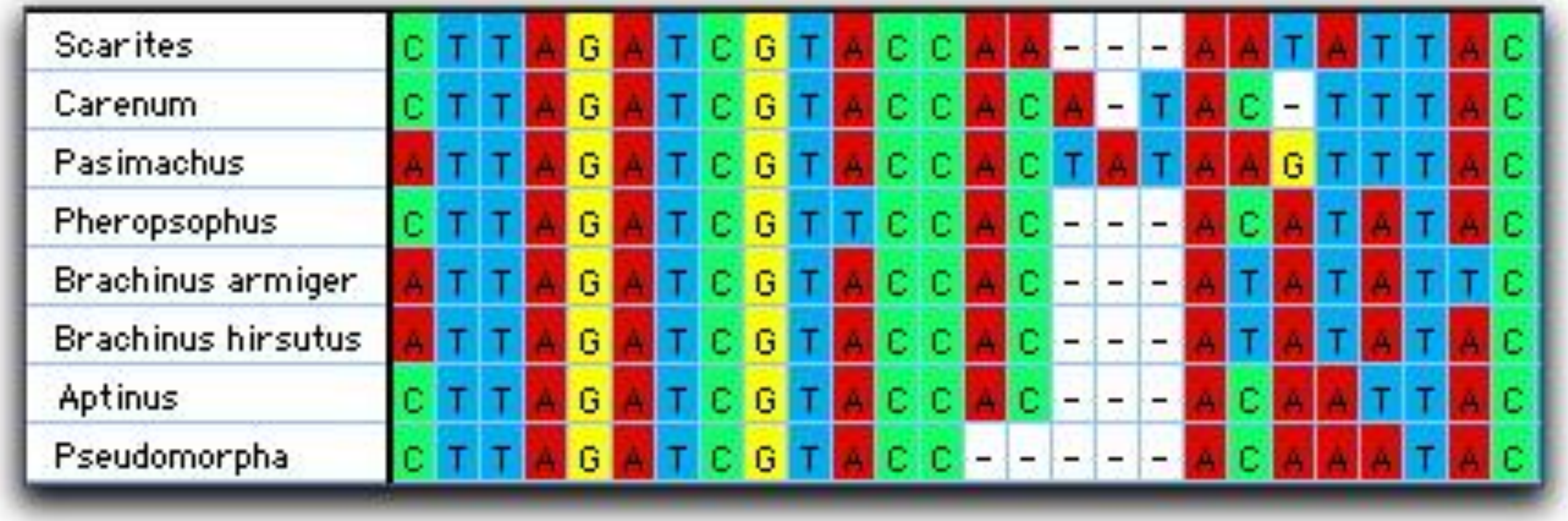


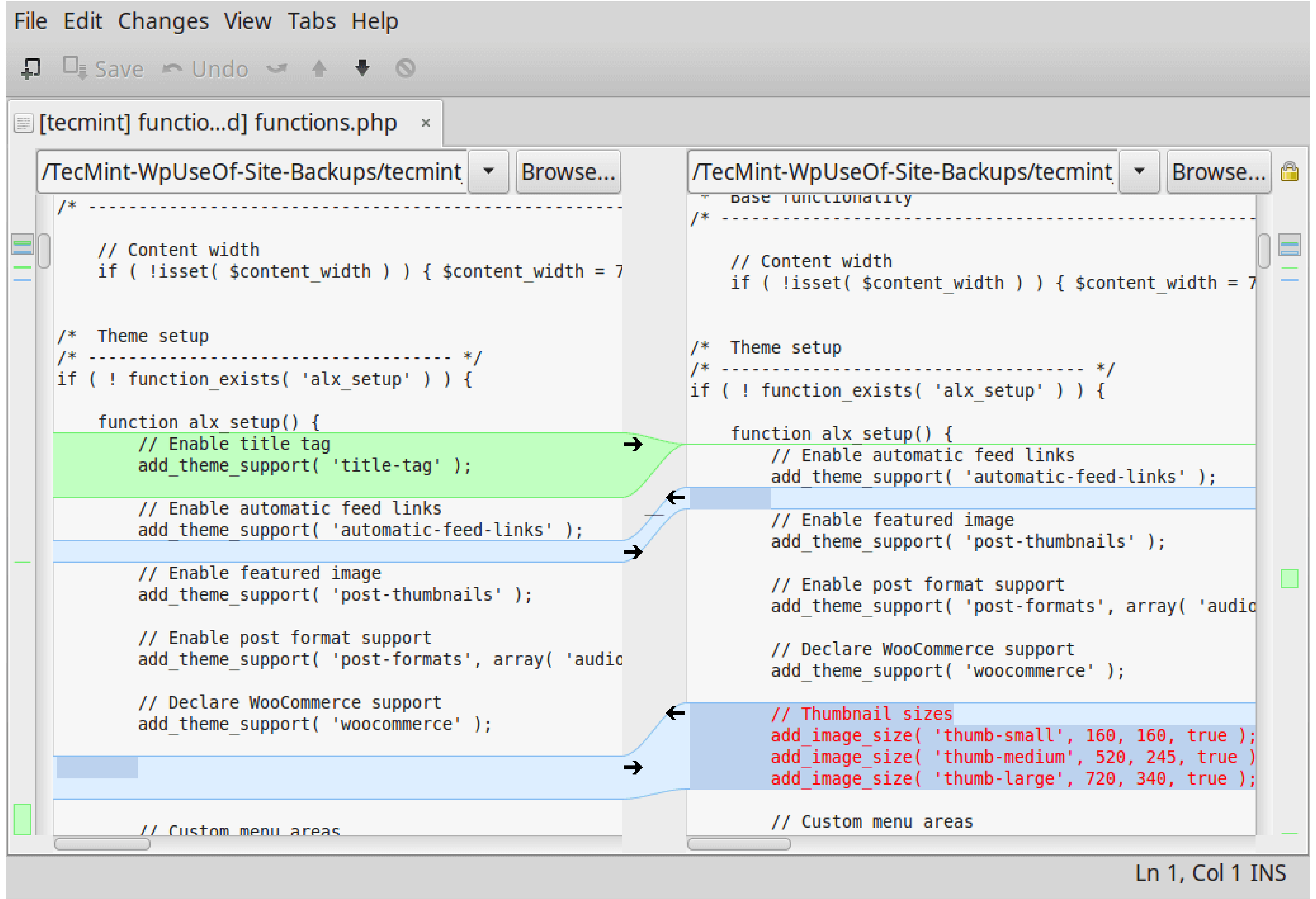
$$W(f,g) = \min \sum_{i,j} c_{i,j}\pi_{i,j} \quad \textbf{s.t.} \quad \sum_j \pi_{i,j} = 1, \quad \sum_i \pi_{i,j} \leq 1, \quad \pi_{i,j} \geq 0$$

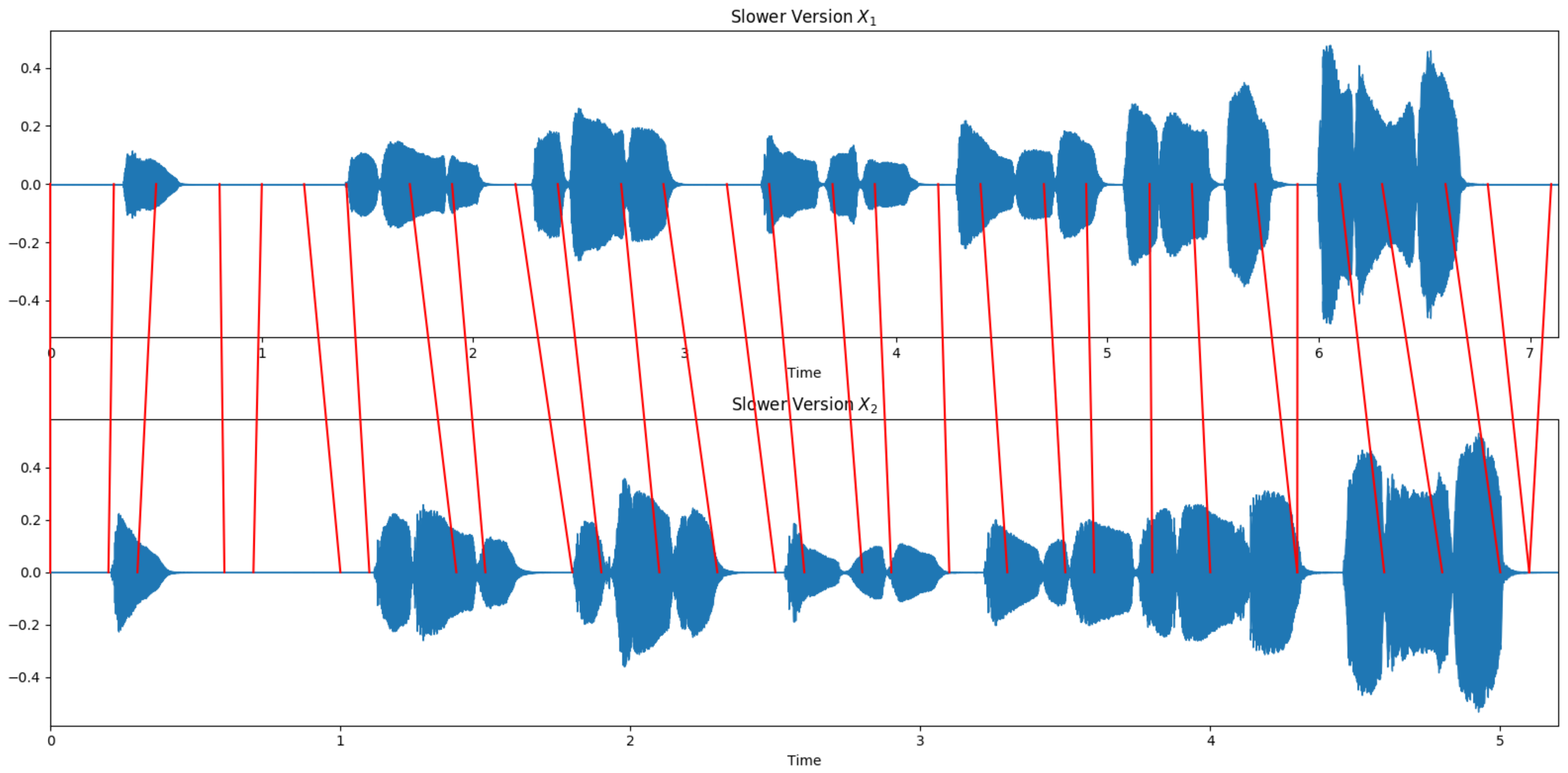
$$\min_{T \text{ injective}} \sum_i c(x_i, y_{T(i)})$$

Similar problems

- DNA sequence alignment
- Text alignment
- Music synchronization
- …

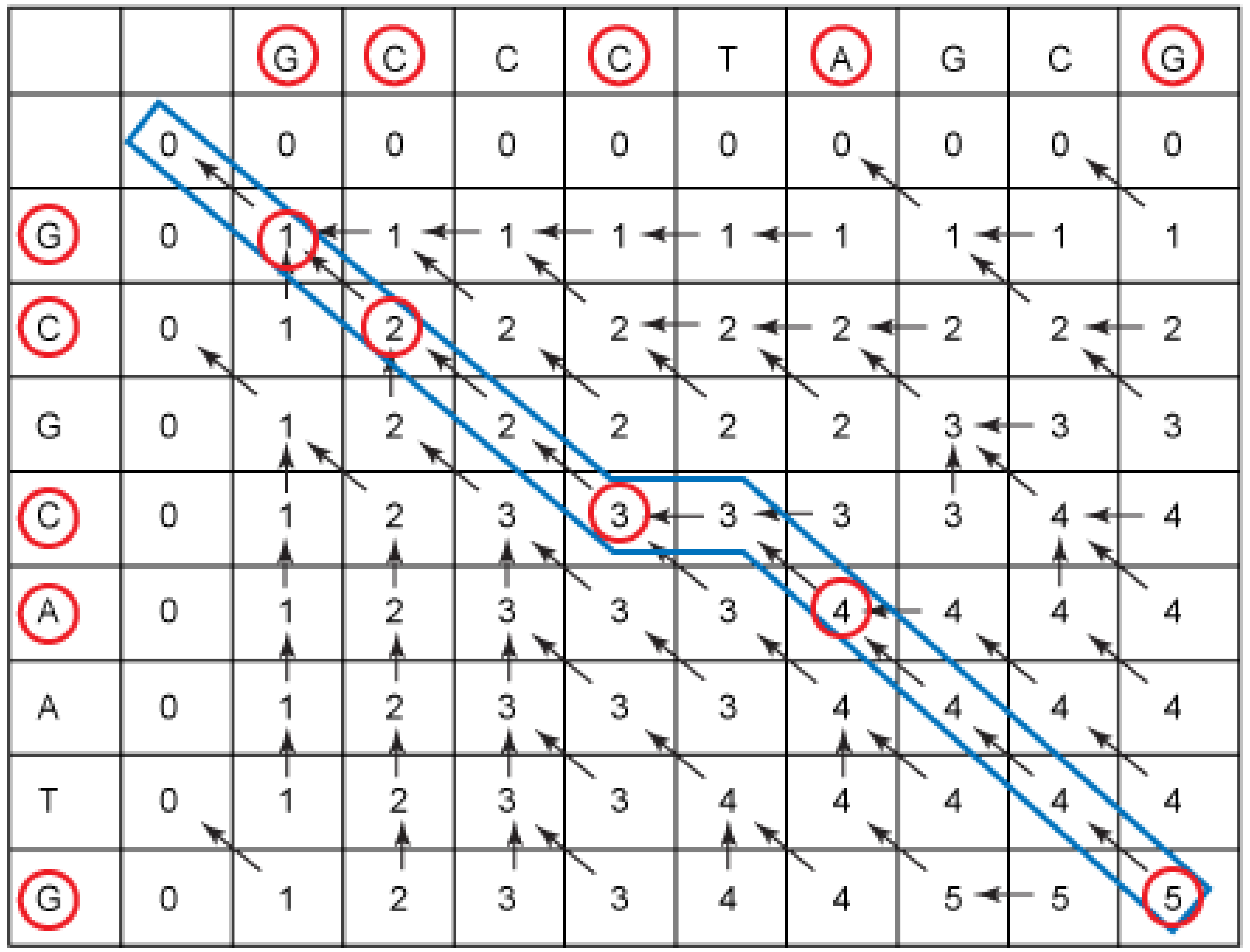






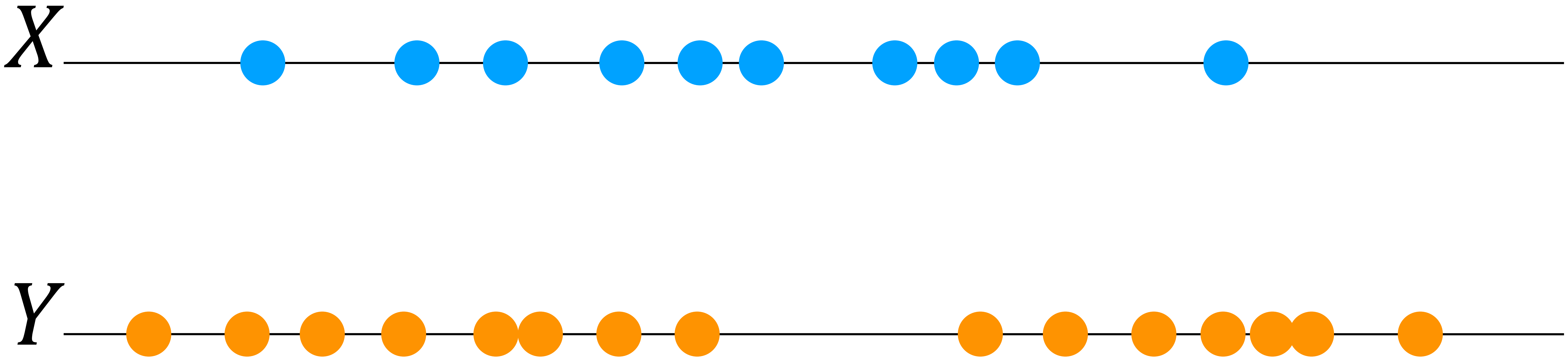
Existing solutions

- Dynamic Time Warping
  - Solves a dynamic programming problem
  - Smith–Waterman algorithm, Needleman–Wunsch algorithm $O(N^2)$ space and time
  - Hirschberg's algorithm $O(N^2)$ time, $O(N)$ space
- All end up doing variants of
  - $A_{i,j} = \min(A_{i-1,j-1} + cost, A_{i-1,j} + cost', A_{i,j-1} + cost'')$

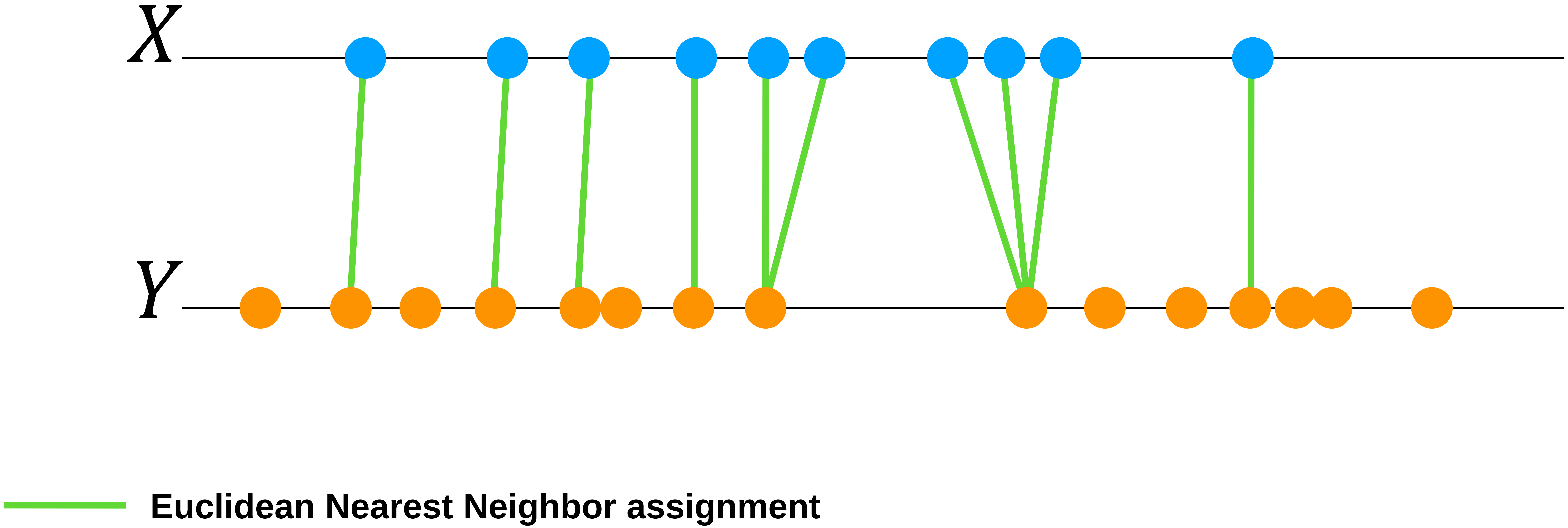


Quadratic time complexity algorithm (linear space)

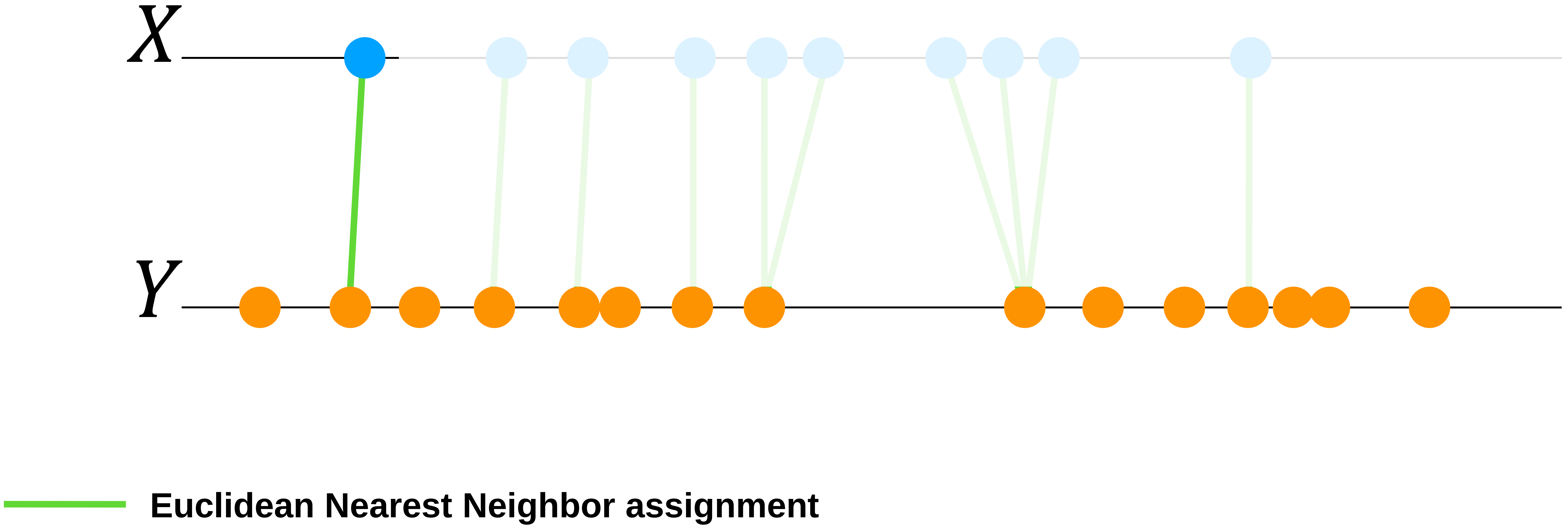
Quadratic time complexity algorithm (linear space)



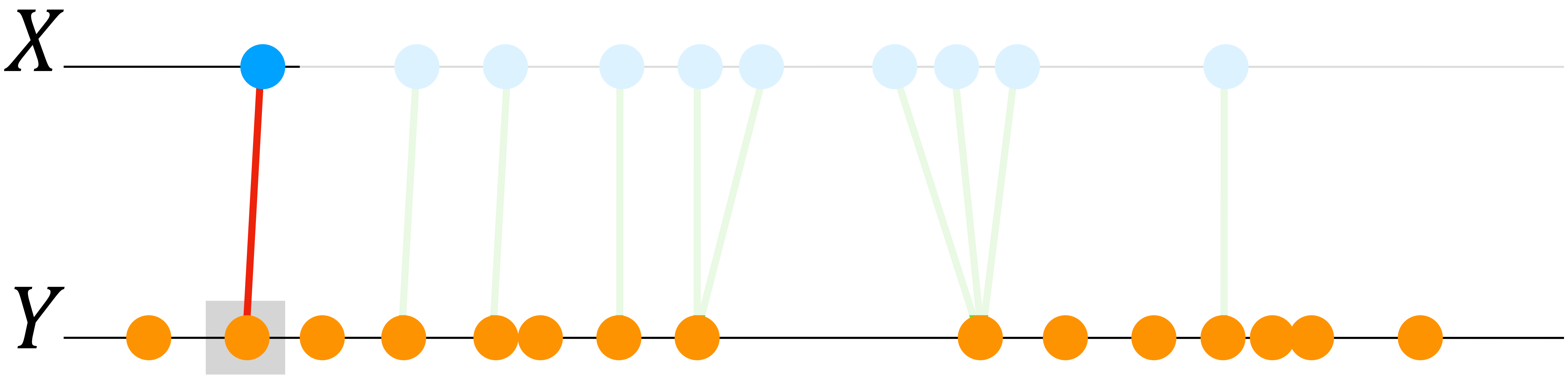
Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)

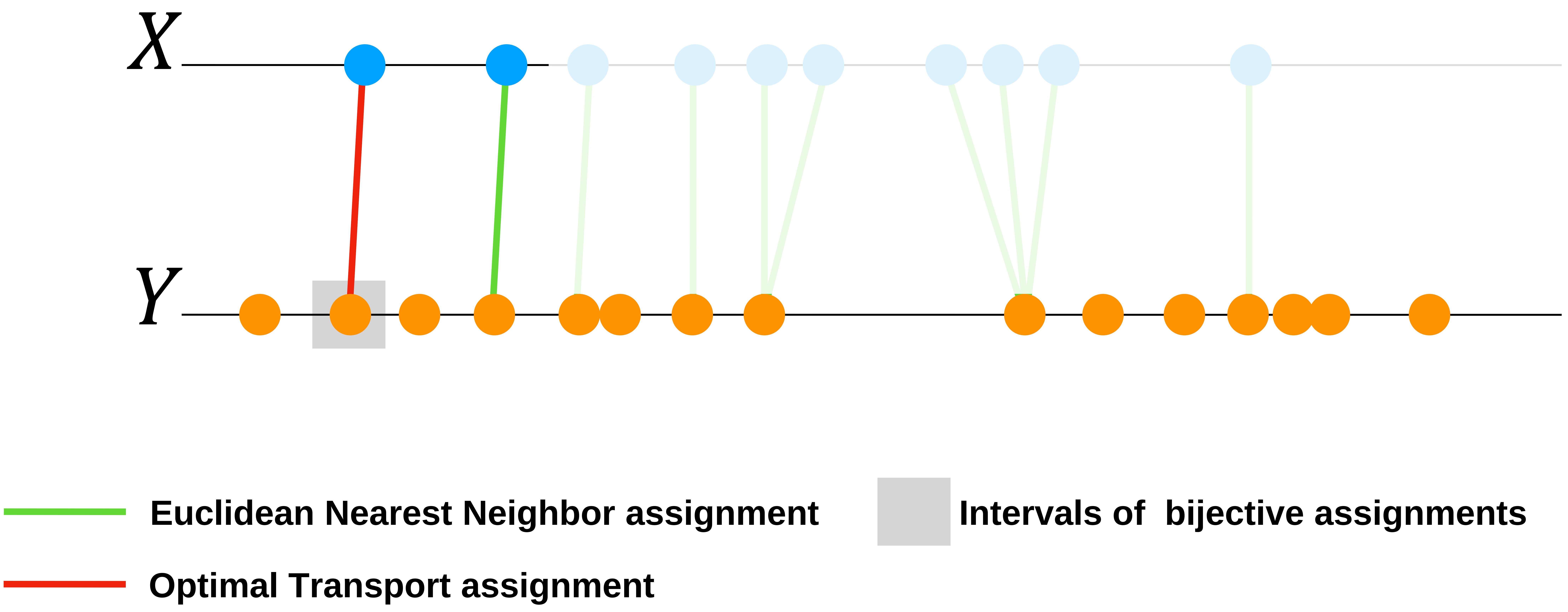


Euclidean Nearest Neighbor assignment

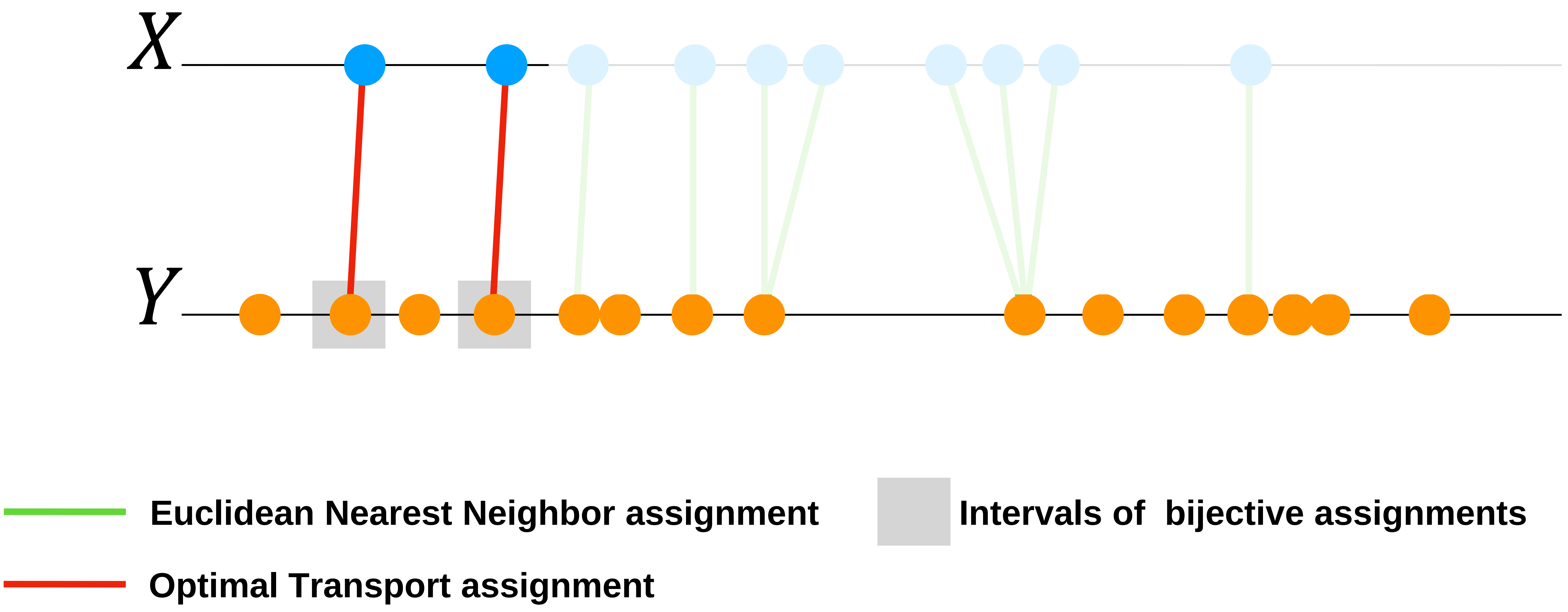
Intervals of bijective assignments

Optimal Transport assignment

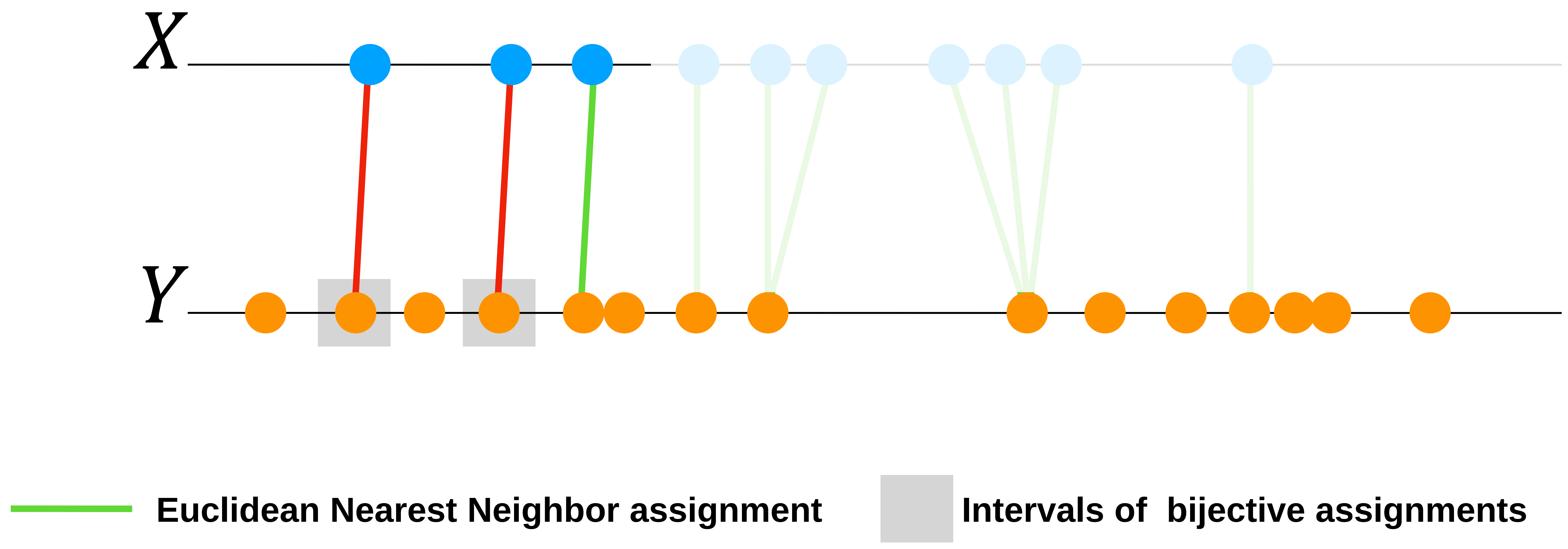
Quadratic time complexity algorithm (linear space)



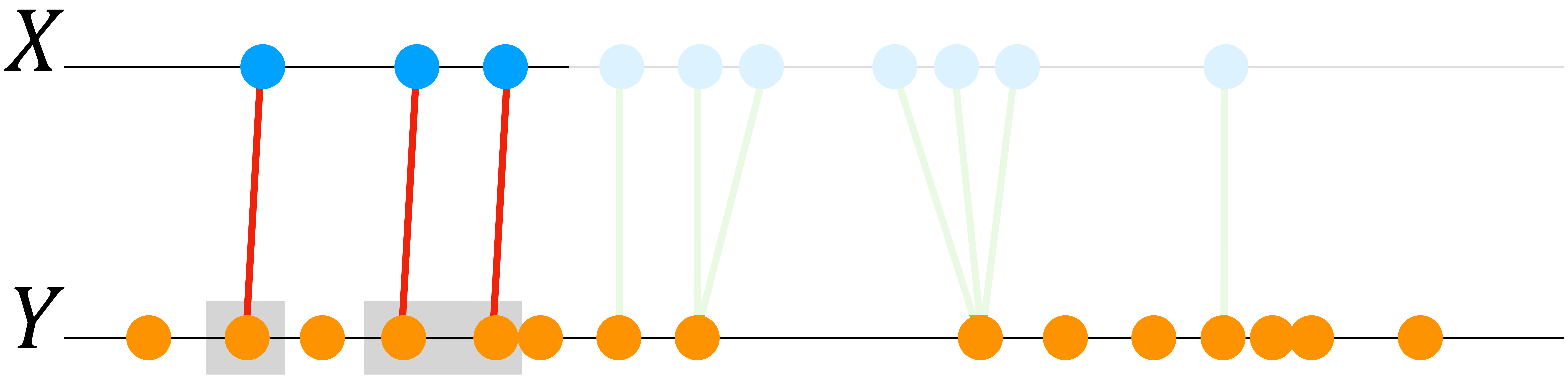
Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)

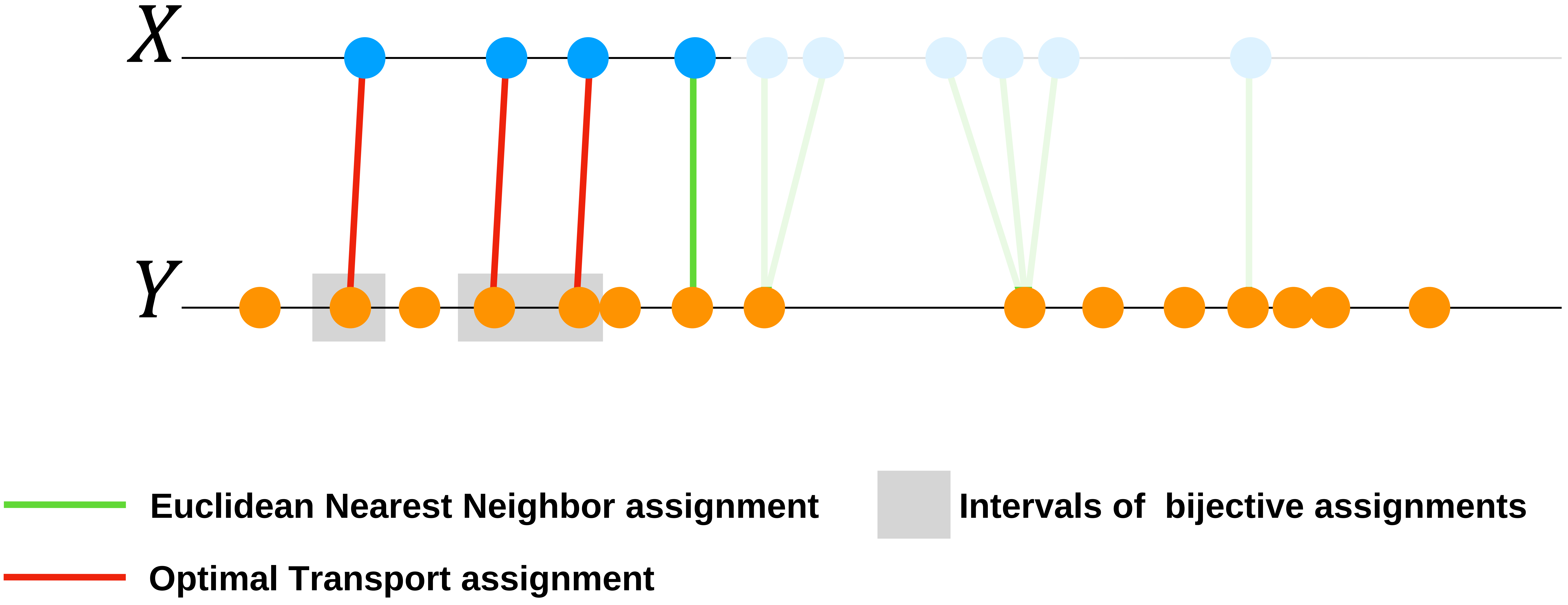


Euclidean Nearest Neighbor assignment

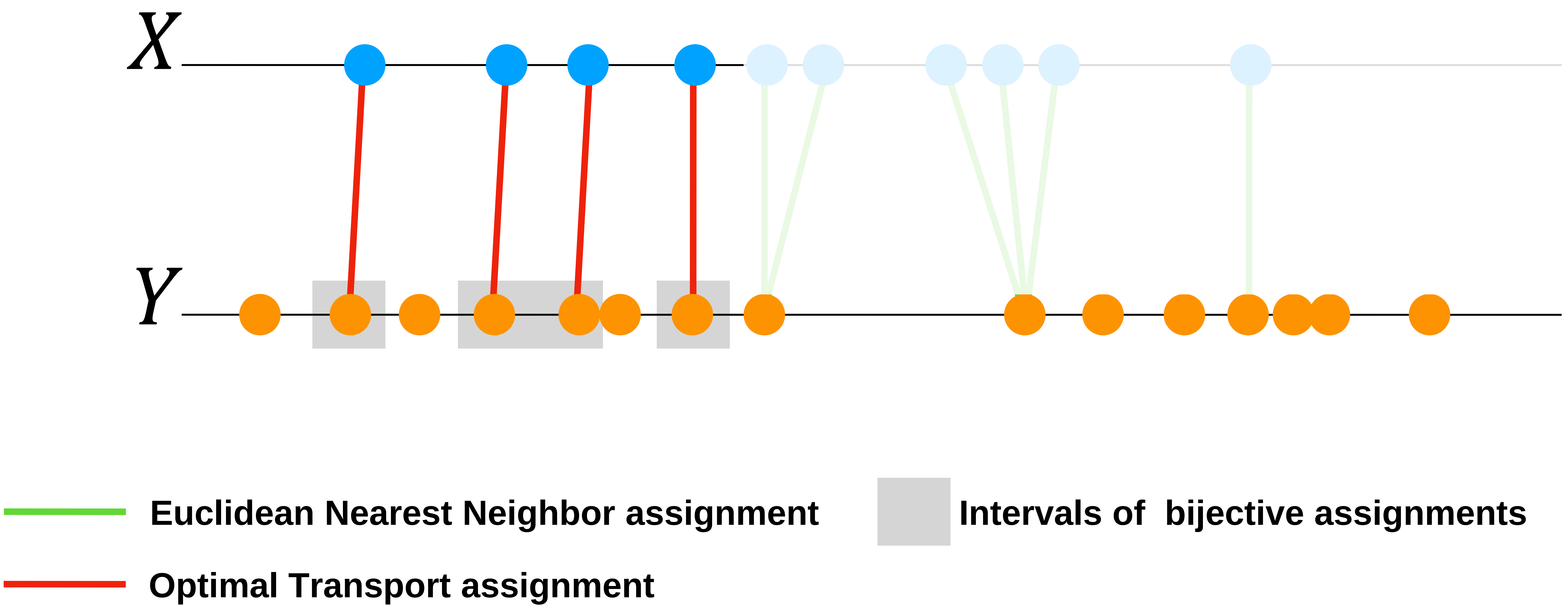
Intervals of bijective assignments

Optimal Transport assignment

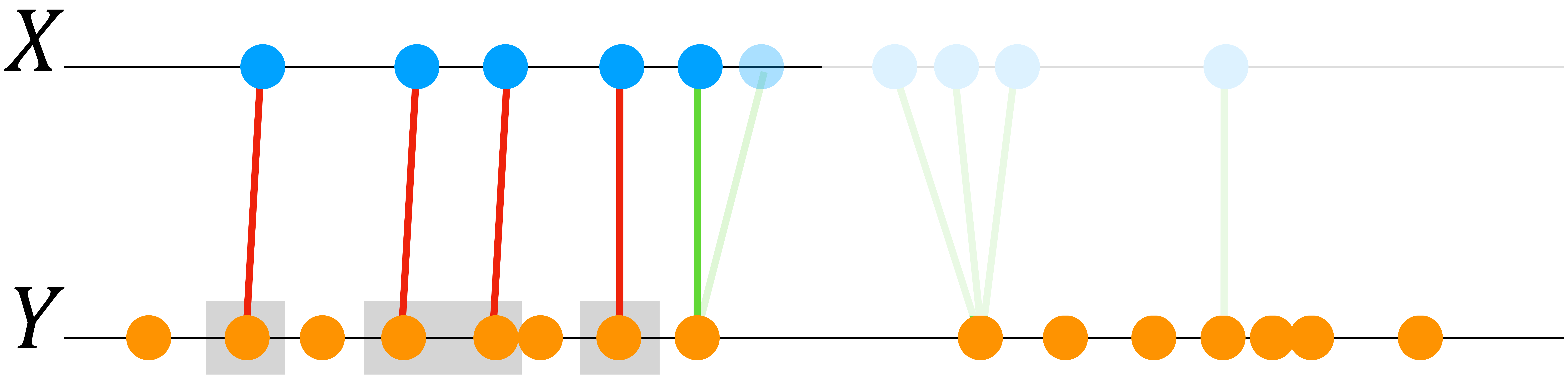
Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)

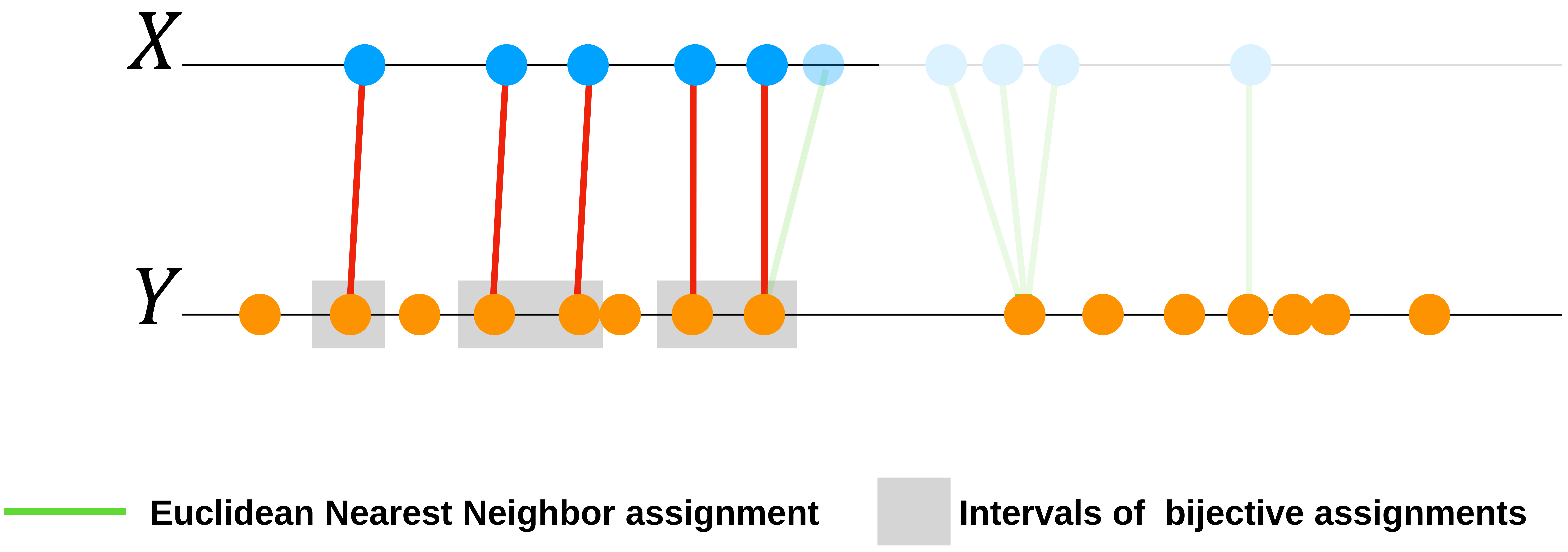


Euclidean Nearest Neighbor assignment

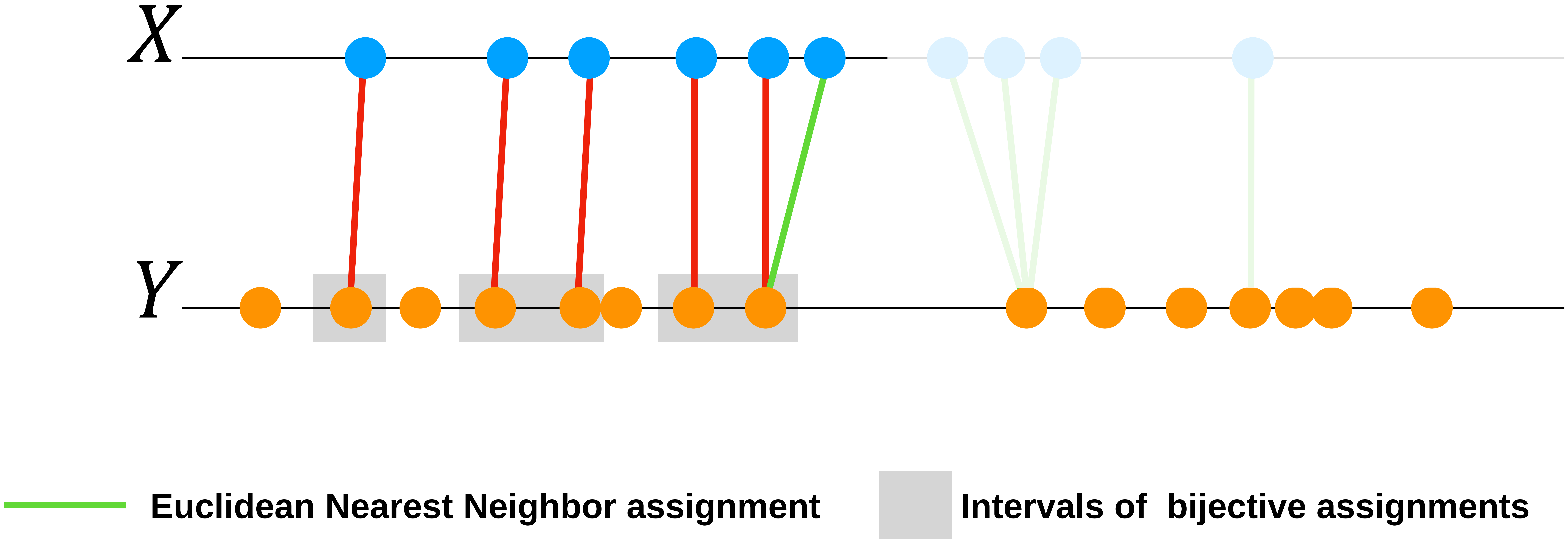
Intervals of bijective assignments

Optimal Transport assignment

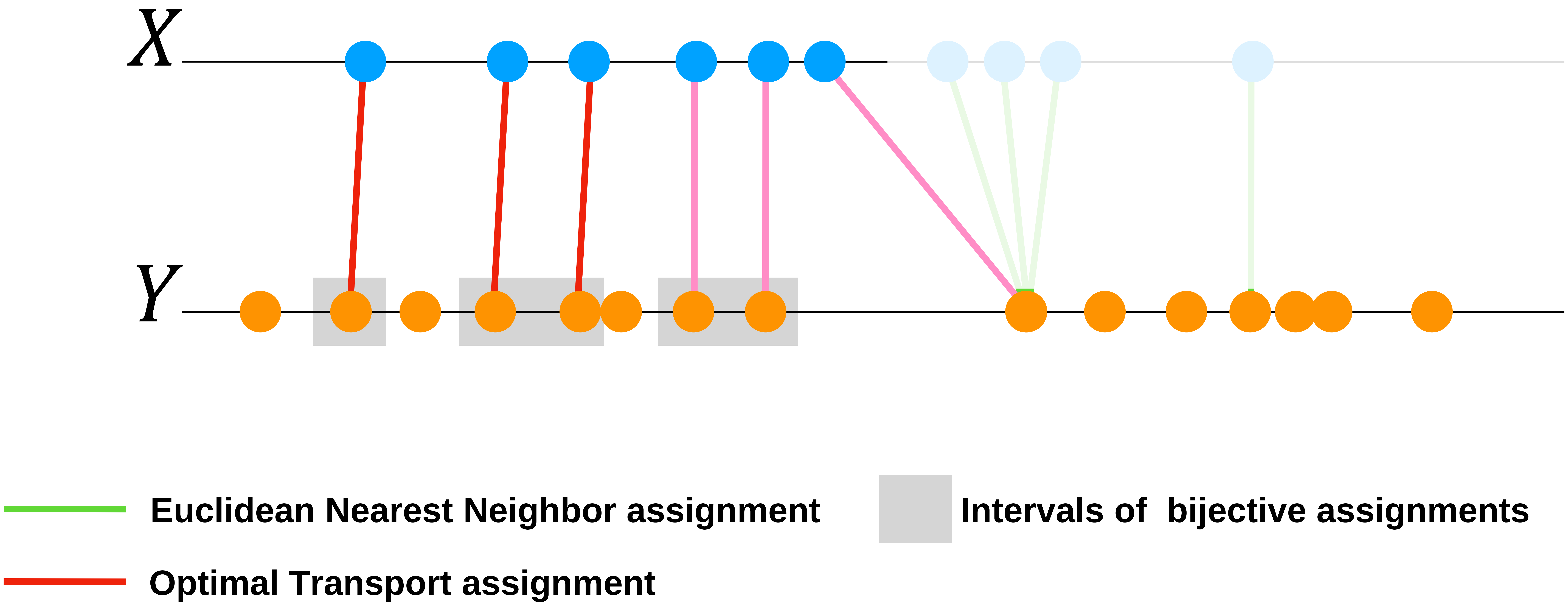
Quadratic time complexity algorithm (linear space)



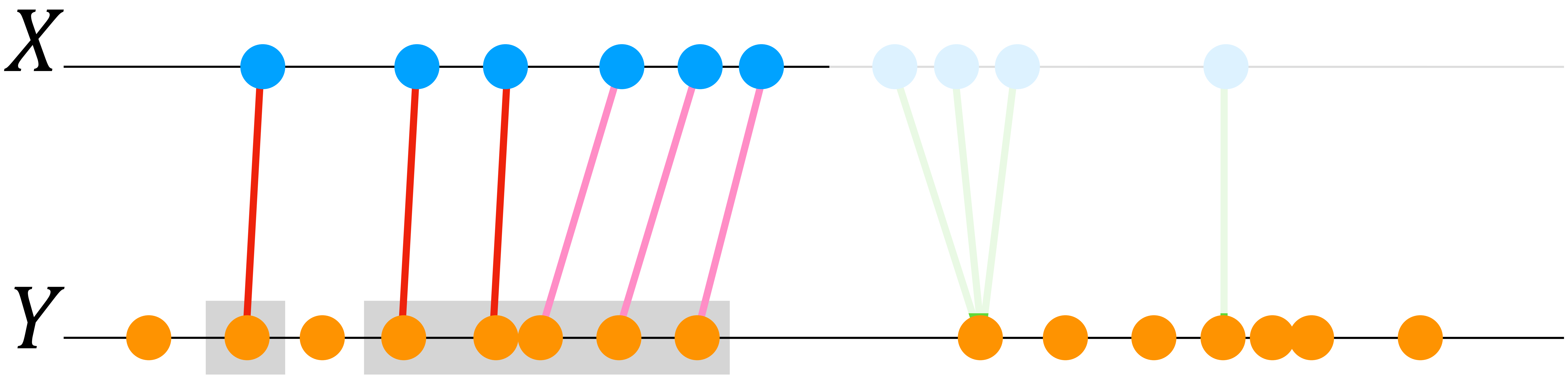
Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)

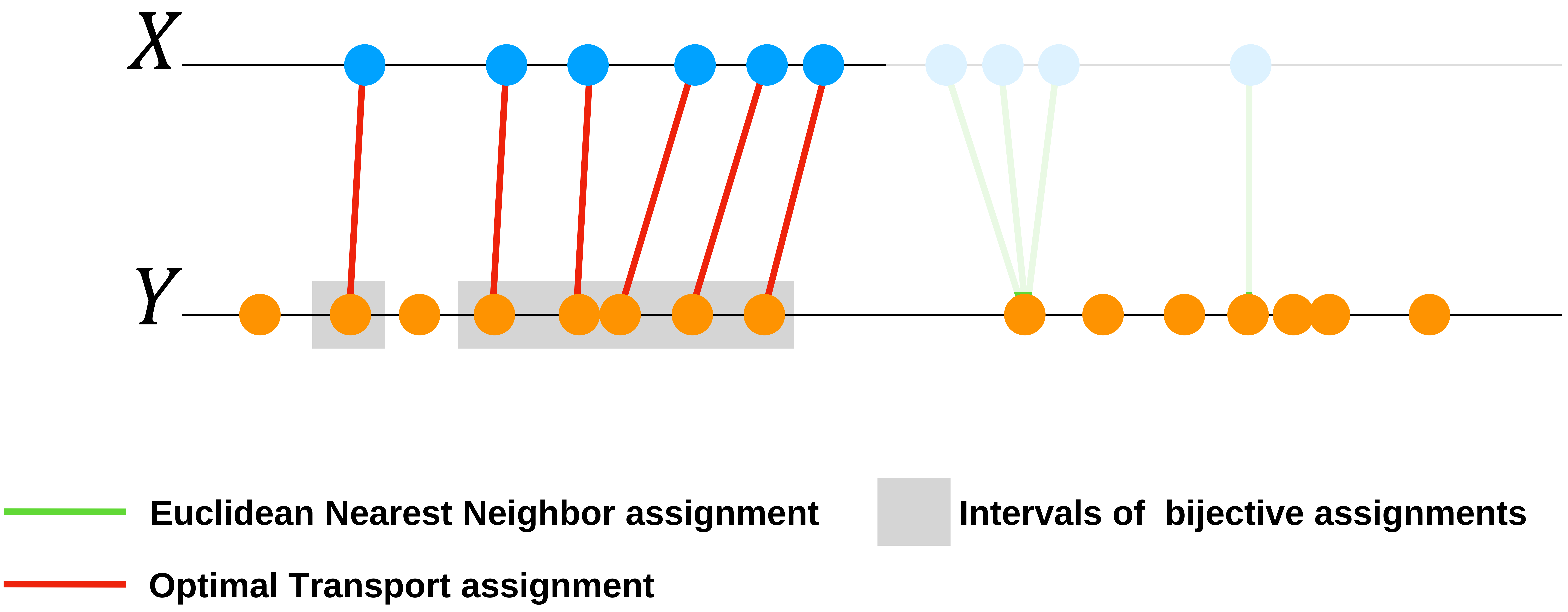


Euclidean Nearest Neighbor assignment

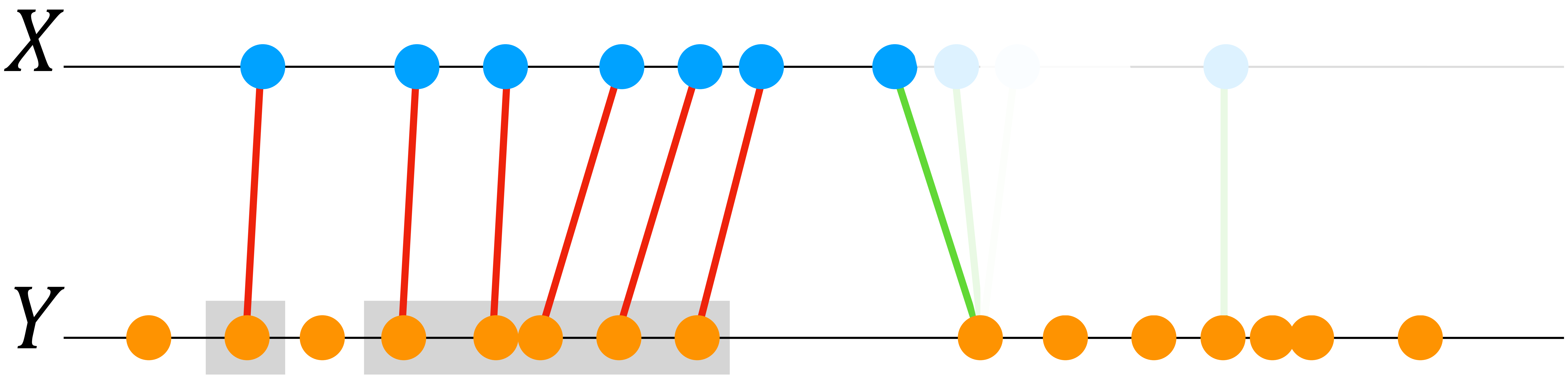
Intervals of bijective assignments

Optimal Transport assignment

Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)

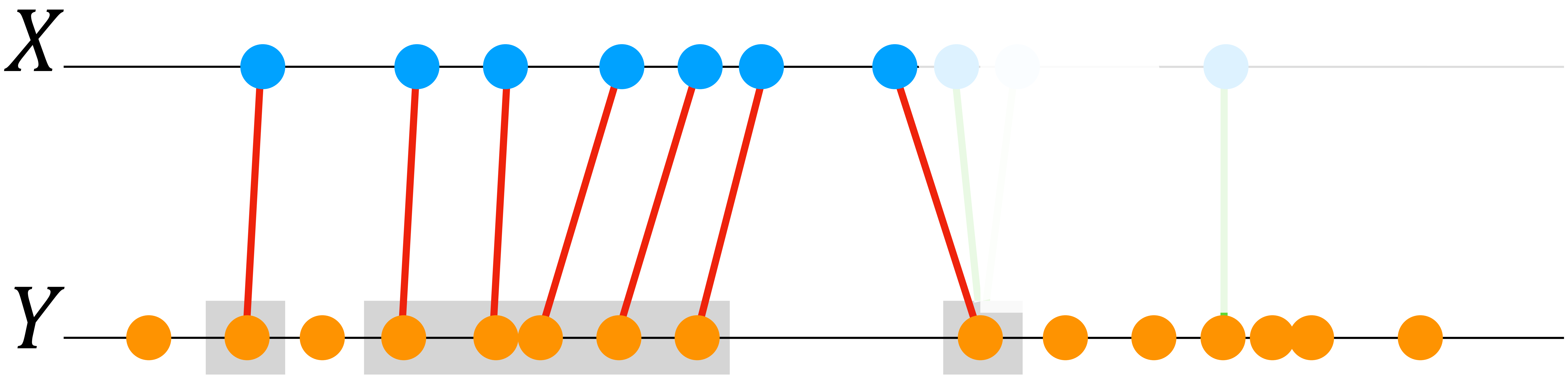


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

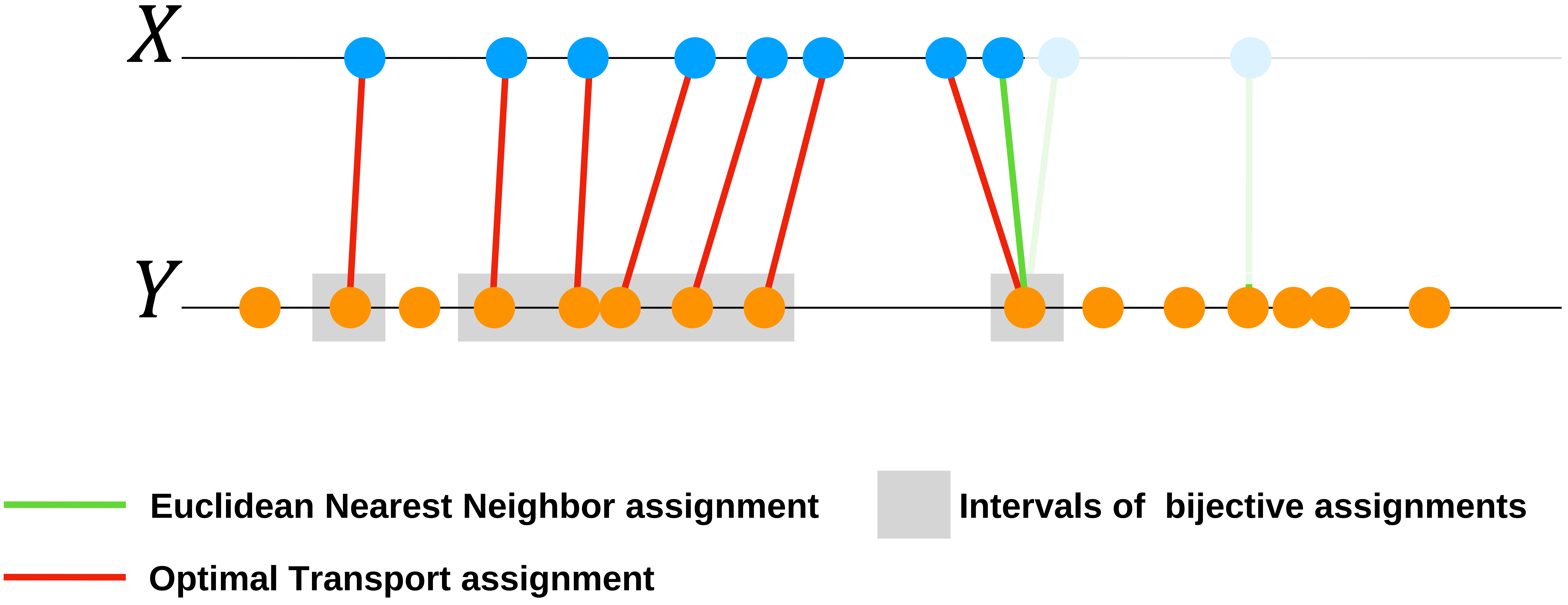


Euclidean Nearest Neighbor assignment

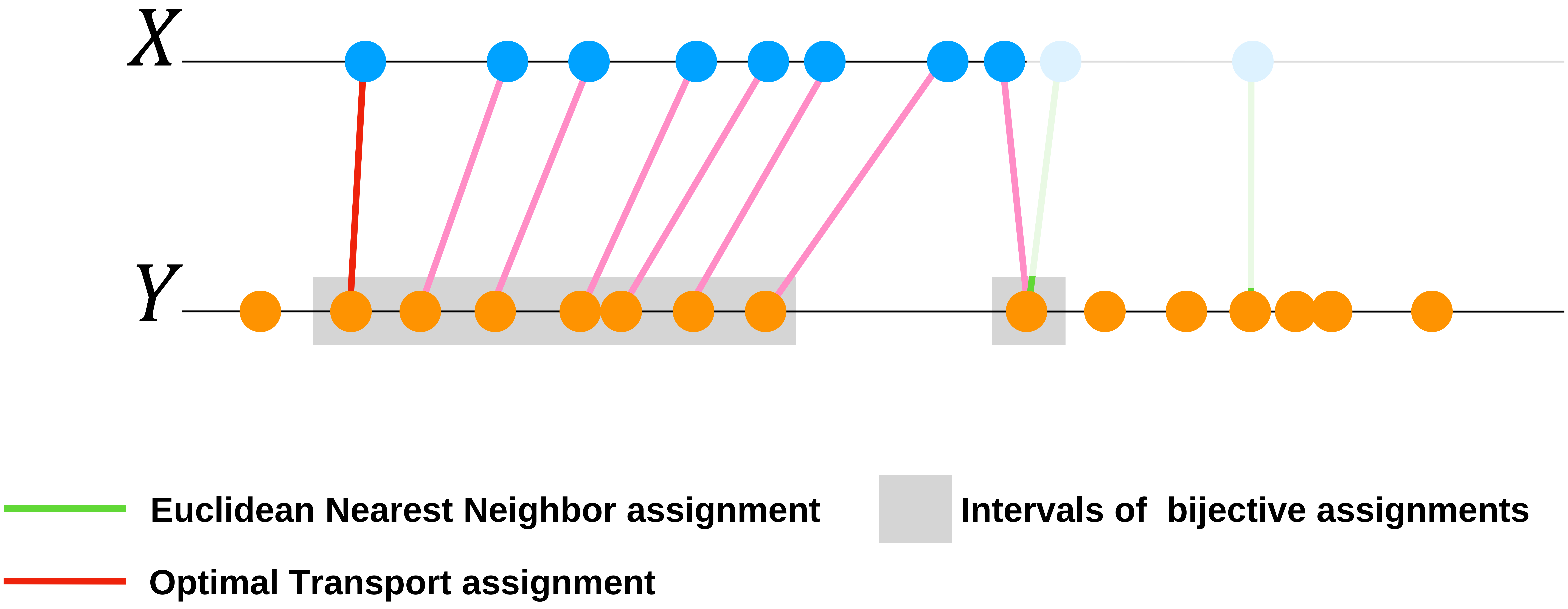
Intervals of bijective assignments

Optimal Transport assignment

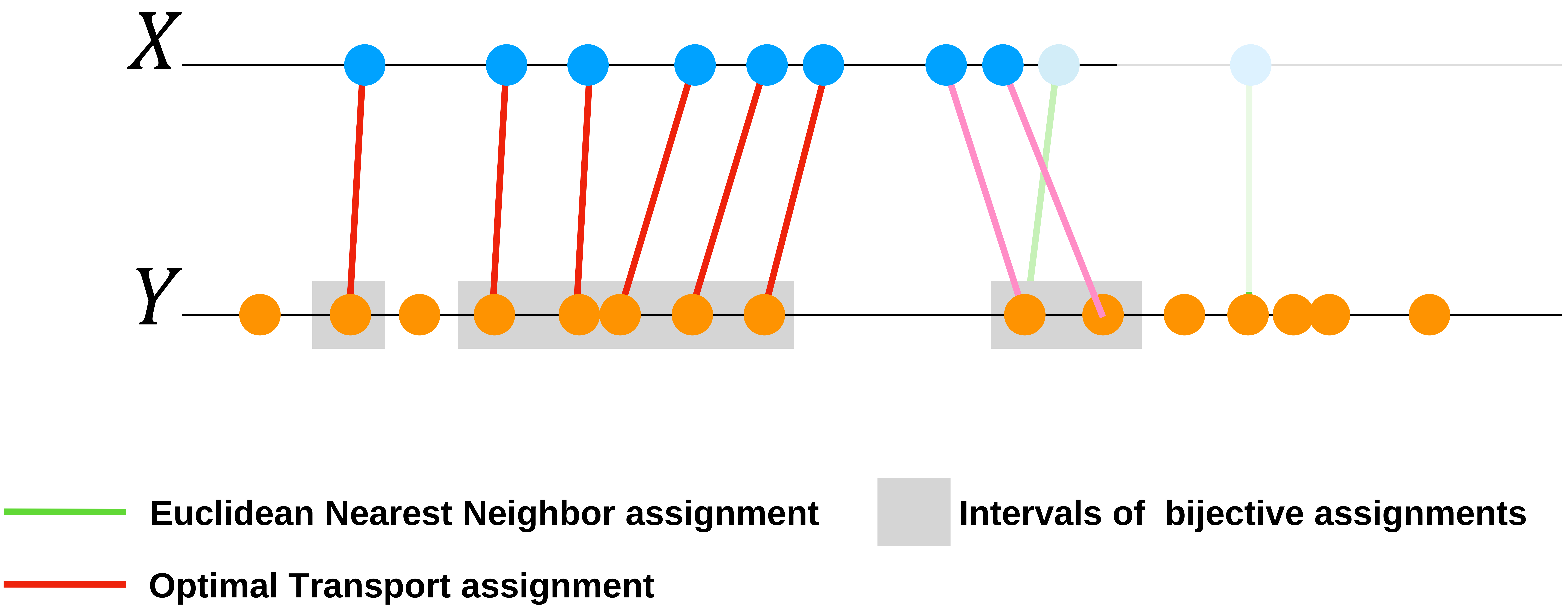
Quadratic time complexity algorithm (linear space)



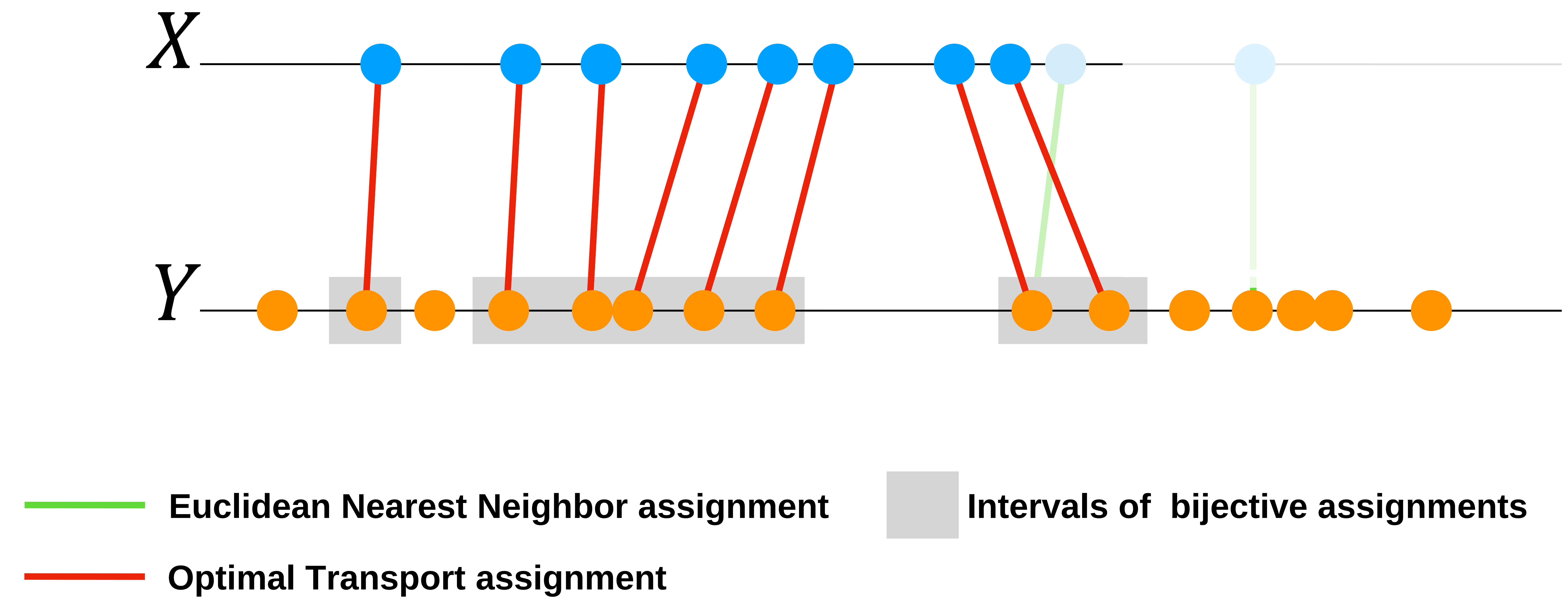
Quadratic time complexity algorithm (linear space)



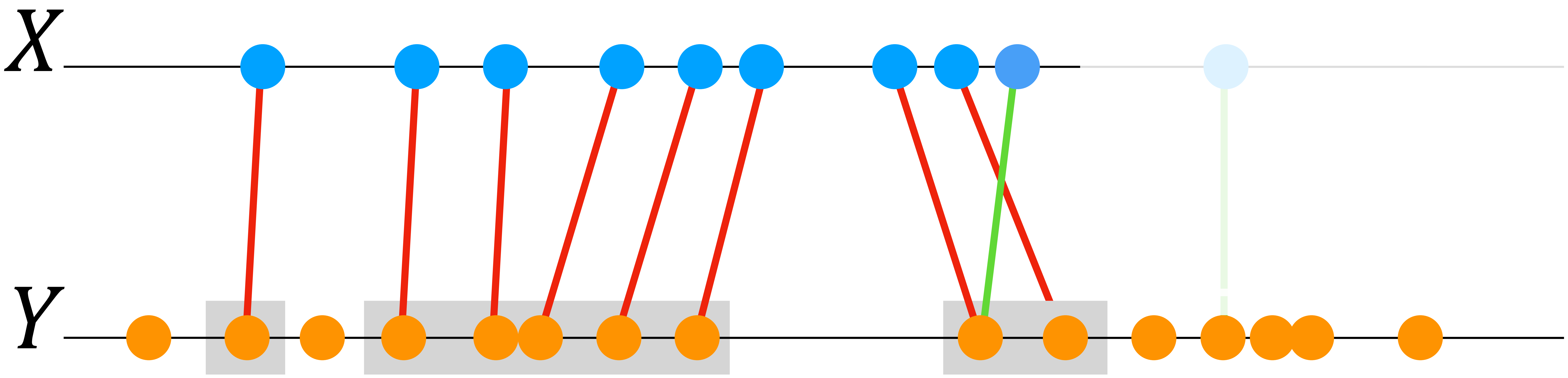
Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)

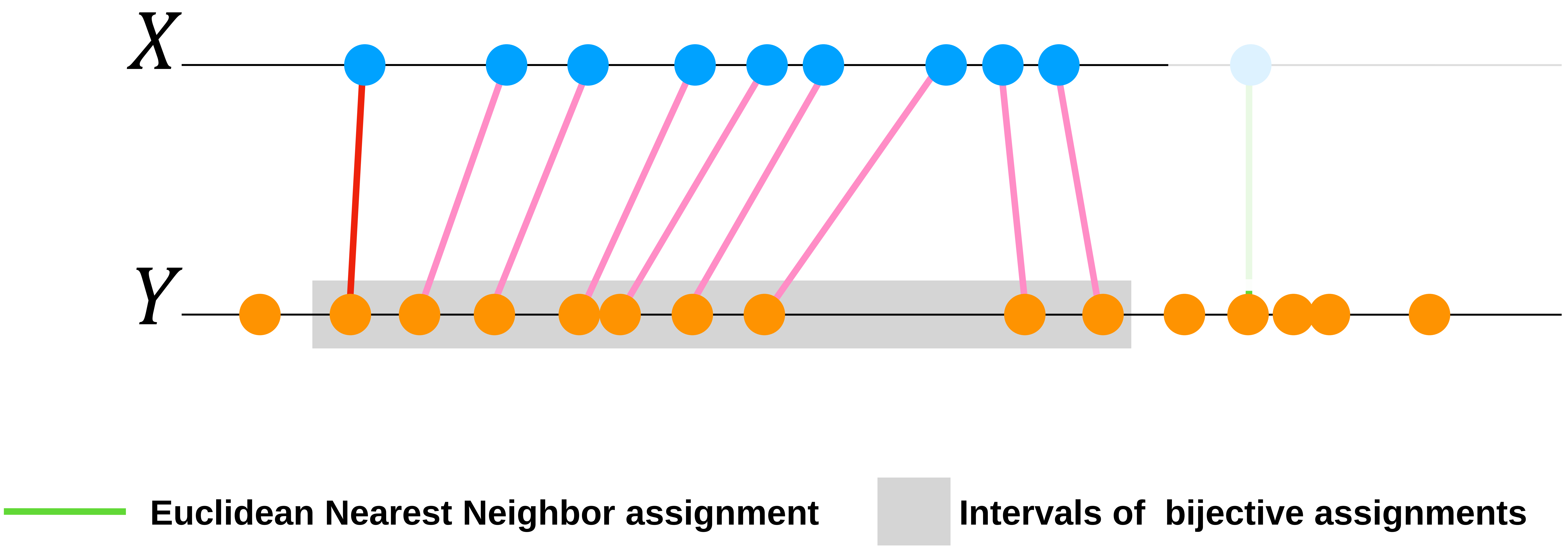


Euclidean Nearest Neighbor assignment

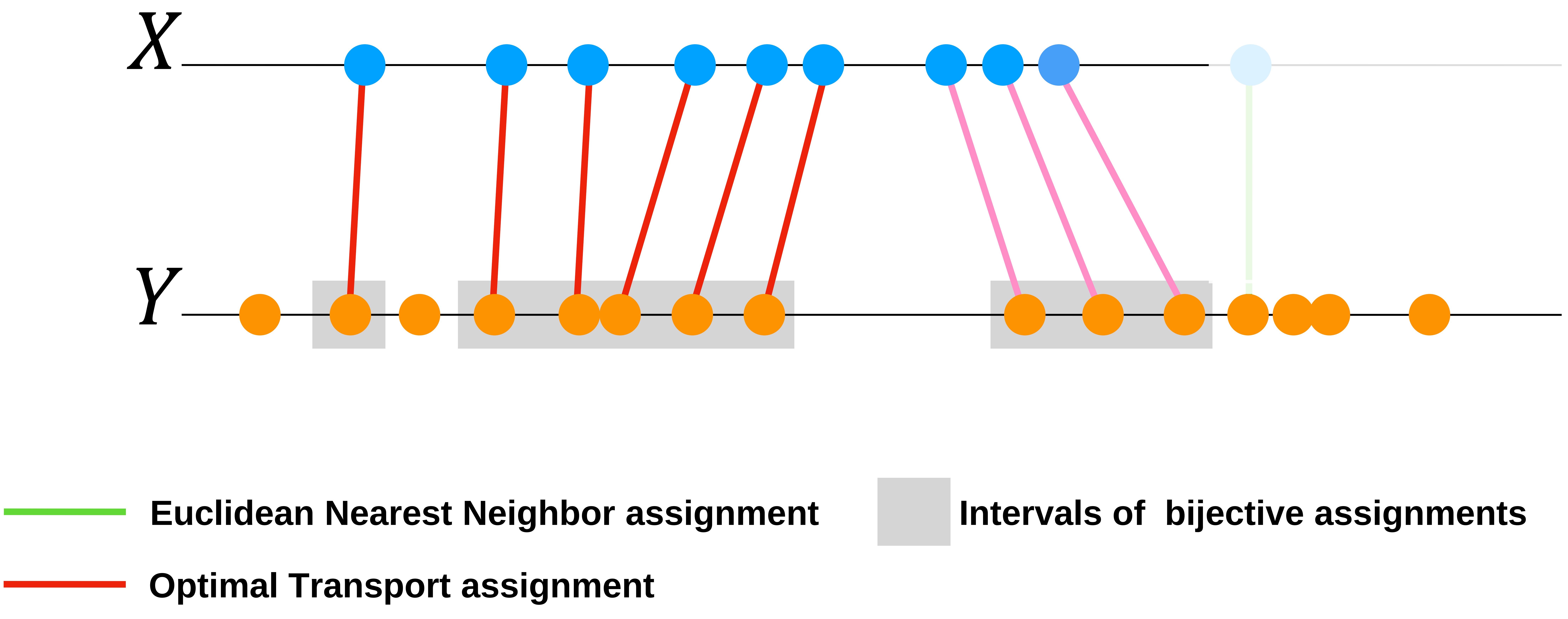
Intervals of bijective assignments

Optimal Transport assignment

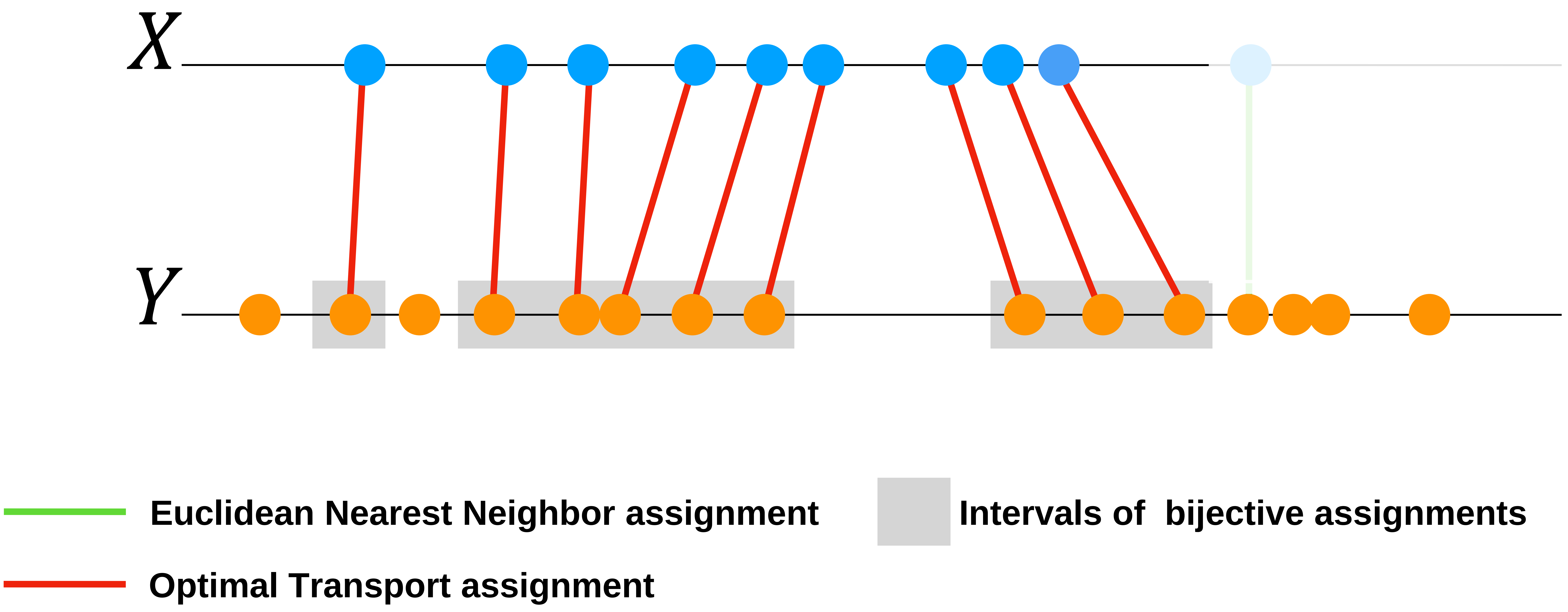
Quadratic time complexity algorithm (linear space)



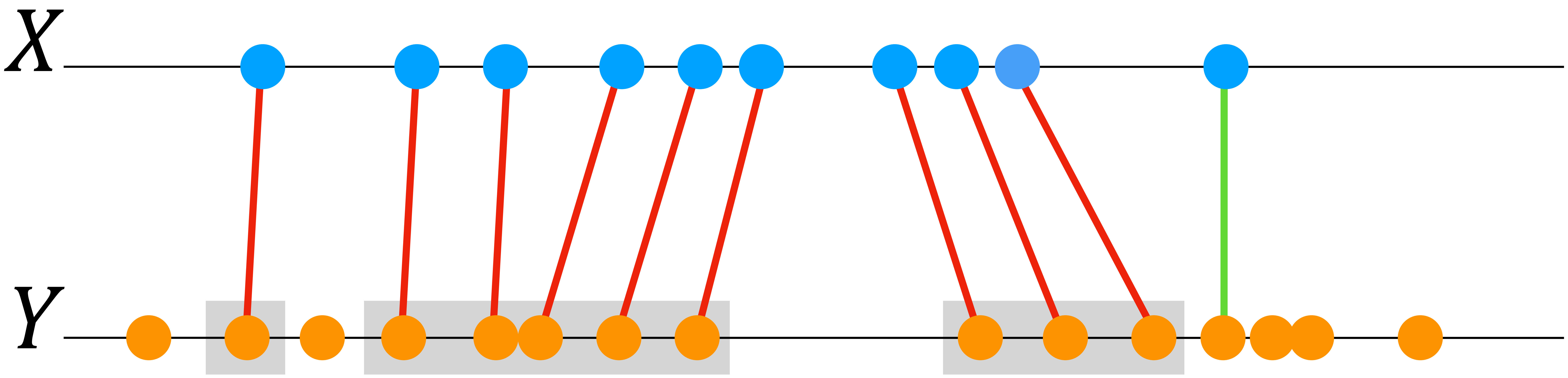
Quadratic time complexity algorithm (linear space)



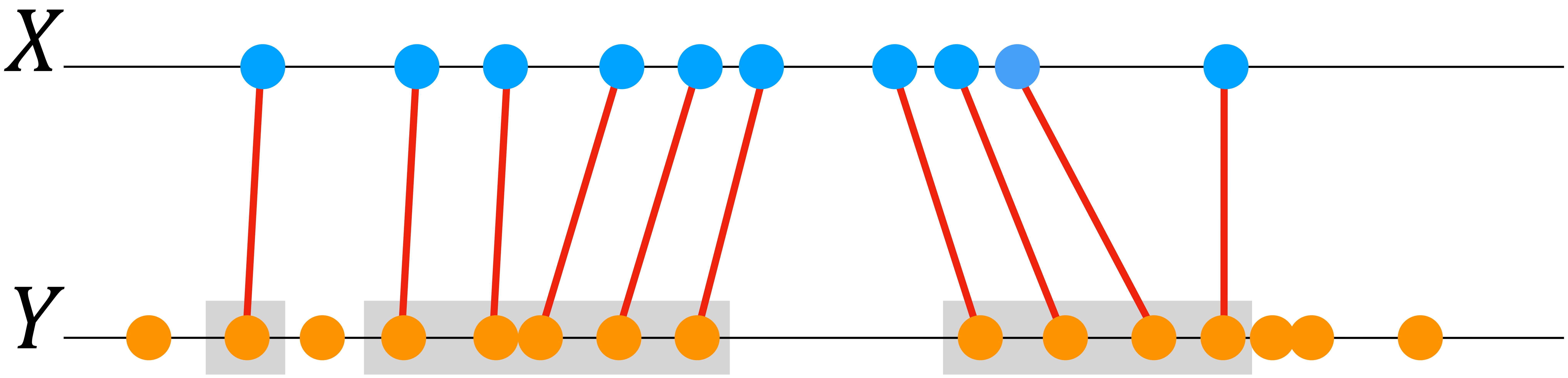
Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)



Quadratic time complexity algorithm (linear space)



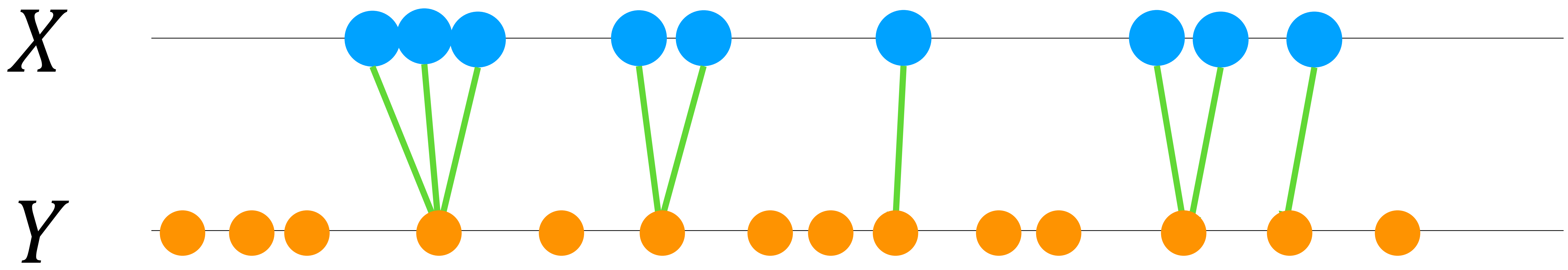
Euclidean Nearest Neighbor assignment

Intervals of bijective assignments

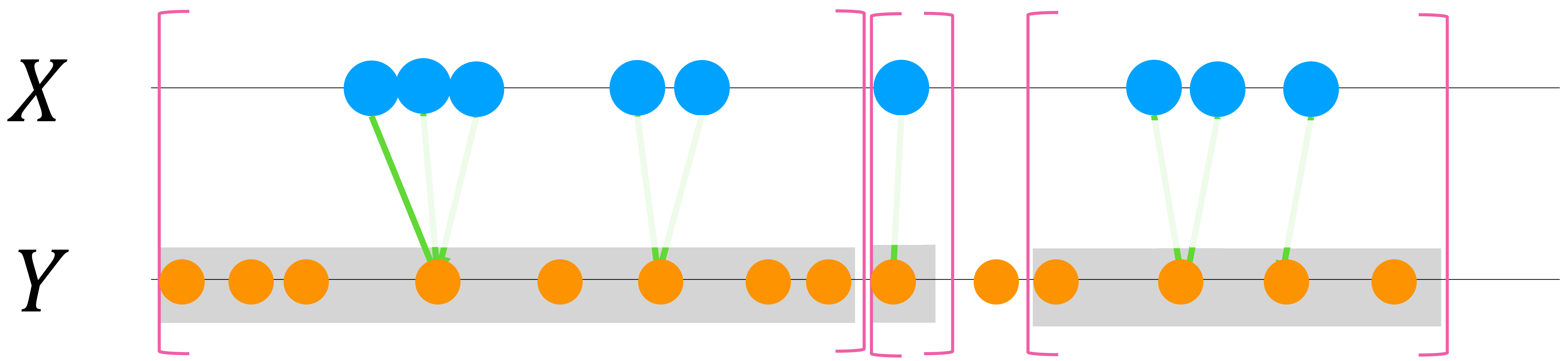
Optimal Transport assignment

Linear time problem decomposition

Problem decomposition



Problem decomposition



Problem decomposition

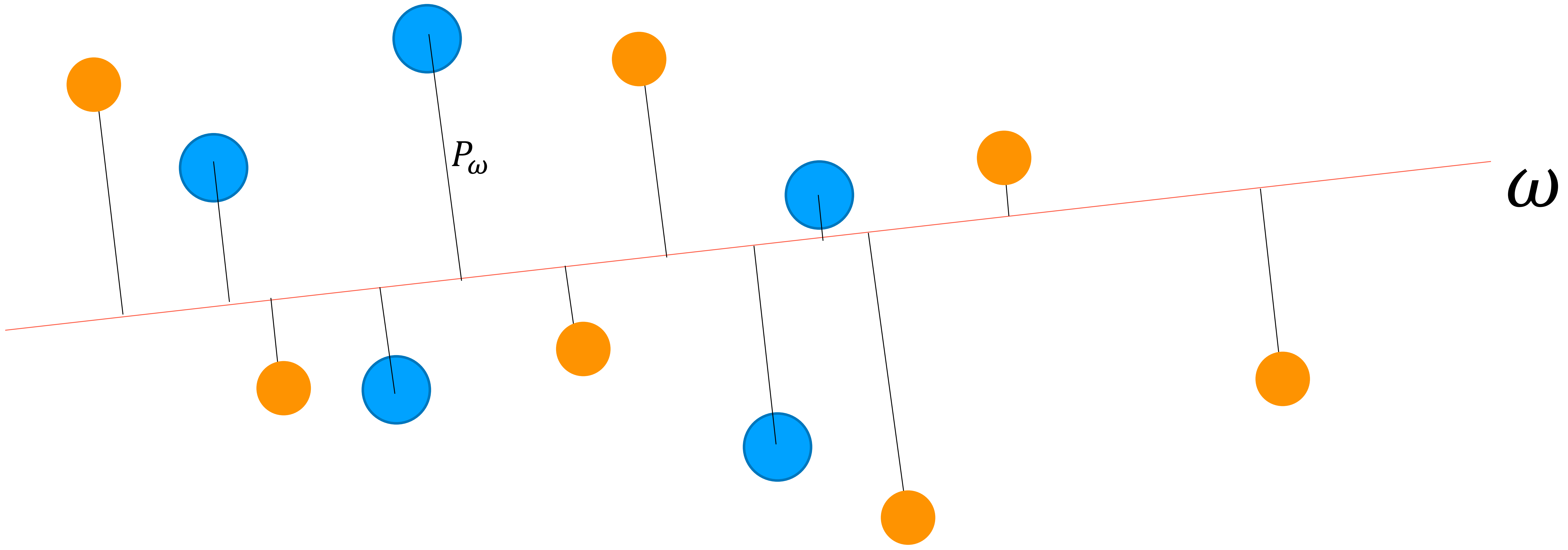
- Computed in quasi-linear time via Union-Find
  - 1 M points in a fraction of a second
- Yields independent subproblems
  - Solvable in parallel
  - That can be further simplified (see paper)

Sliced Partial Optimal Transport (SPOT)

Extension to d dimensions

- Sliced optimal transport

$$E = \int_{\mathbb{S}^{d-1}} W(P_\omega X, P_\omega Y)\mathrm{d}\omega = \int_{\mathbb{S}^{d-1}} \min_T \sum_i \left(P_\omega x_i - P_\omega y_{T(i)}\right)^2 \mathrm{d}\omega$$

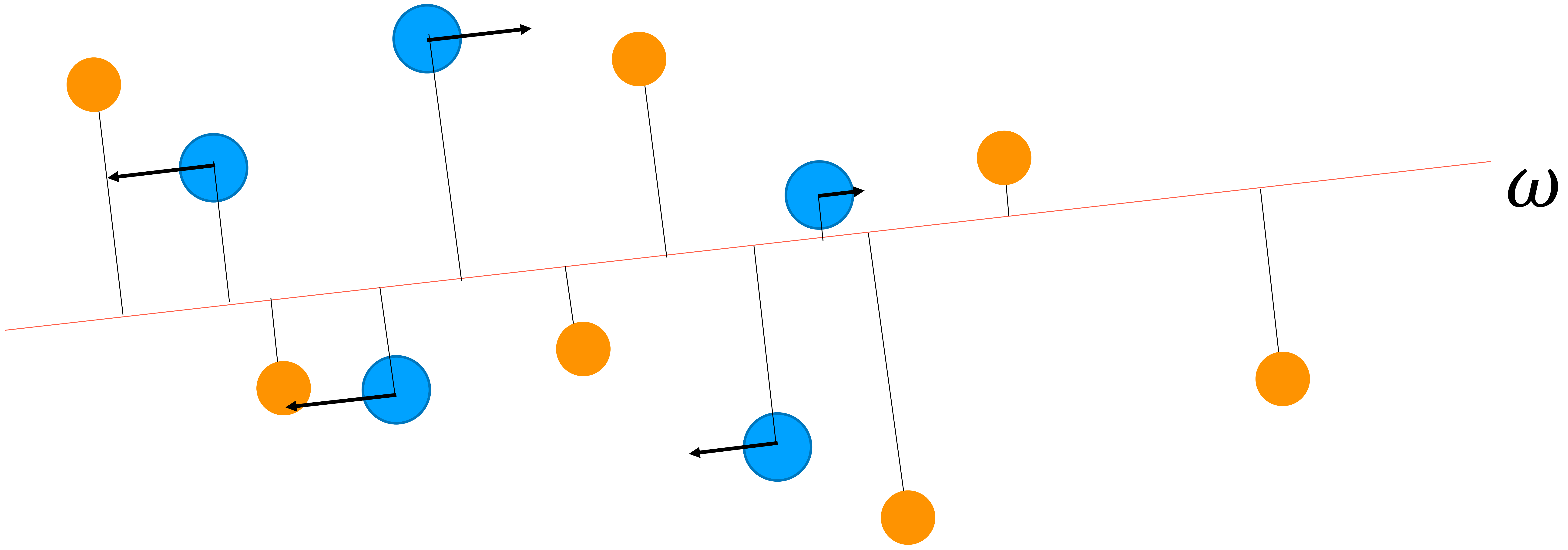


Gradient flow

- Sliced optimal transport

$$X^{n+1} = X^n - \nabla E$$

Stochastic descent: $X^{n+1} = X^n - \nabla \mathrm{W}(P_{\omega^n}X, P_{\omega^n}Y).\omega^n$

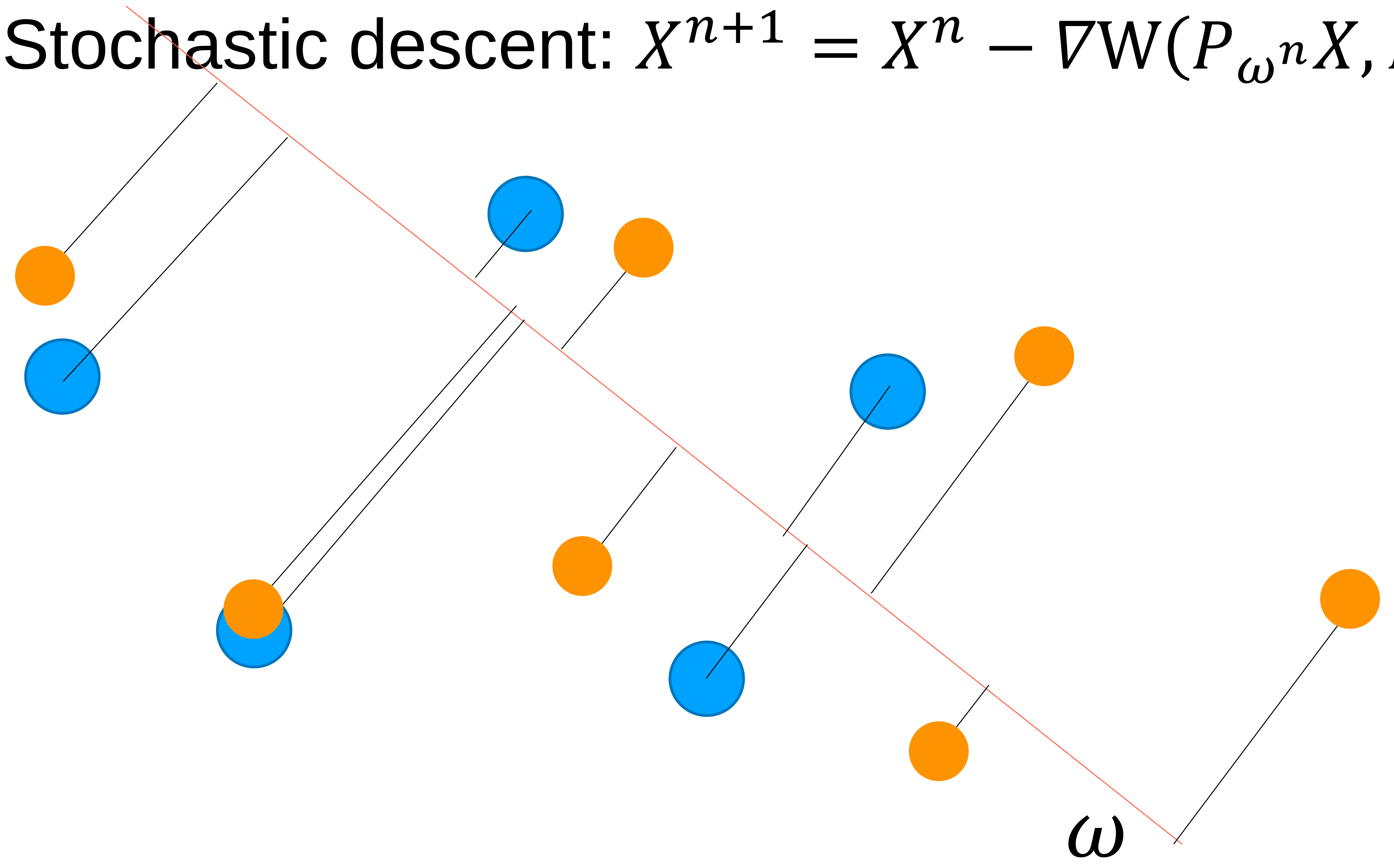


Gradient flow

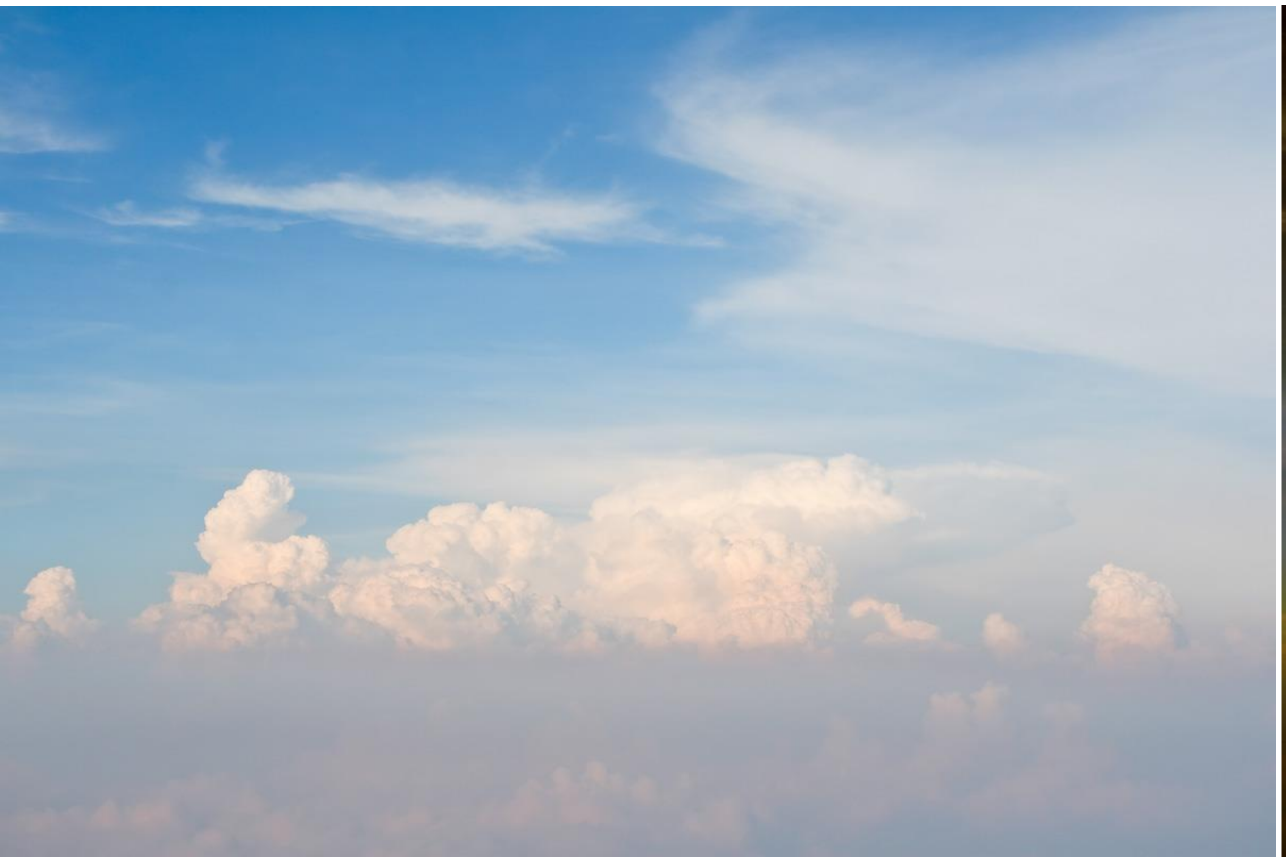
- Sliced optimal transport

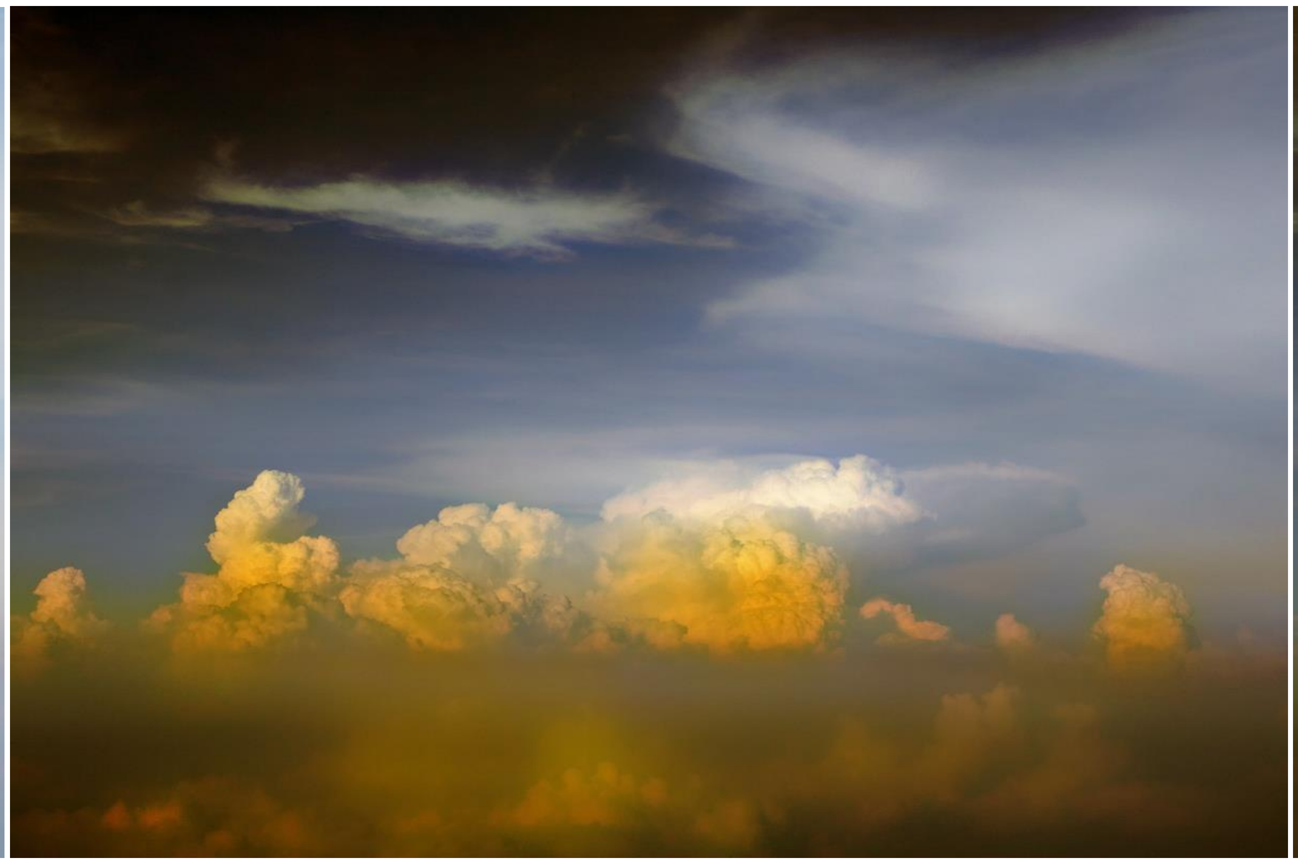
$$X^{n+1} = X^n - \nabla E$$

Stochastic descent: $X^{n+1} = X^n - \nabla \mathrm{W}(P_{\omega^n}X, P_{\omega^n}Y).\omega^n$



Color transfer application



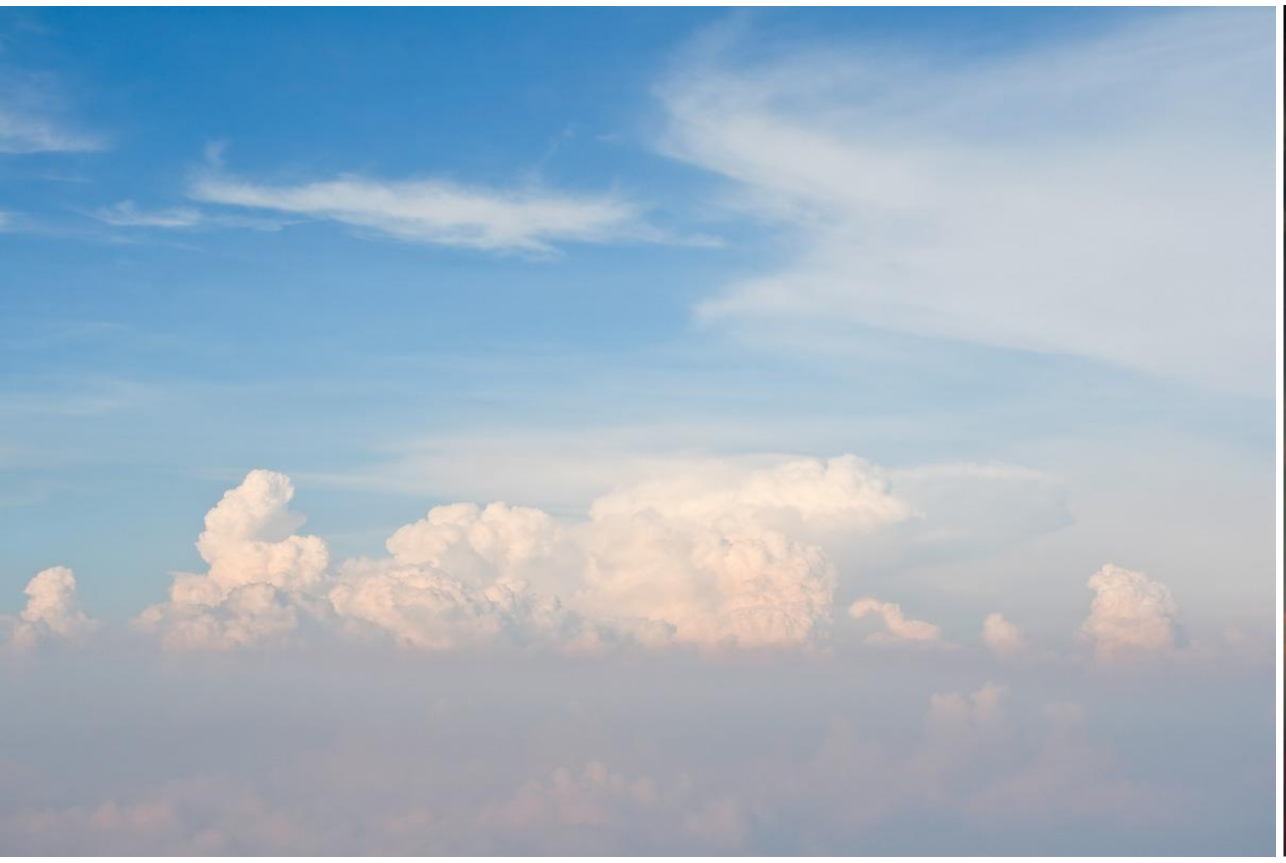


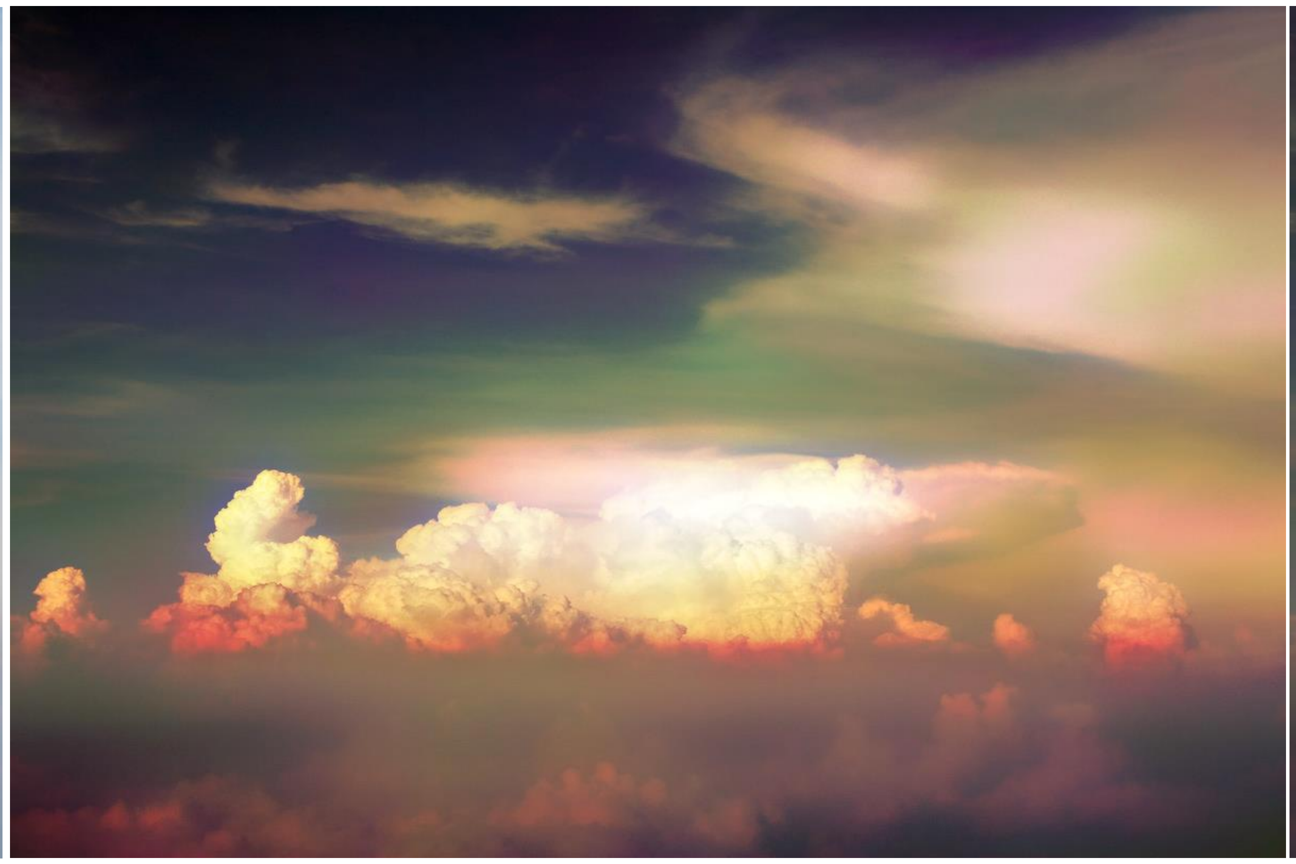
Full Transfer

Target 20% larger

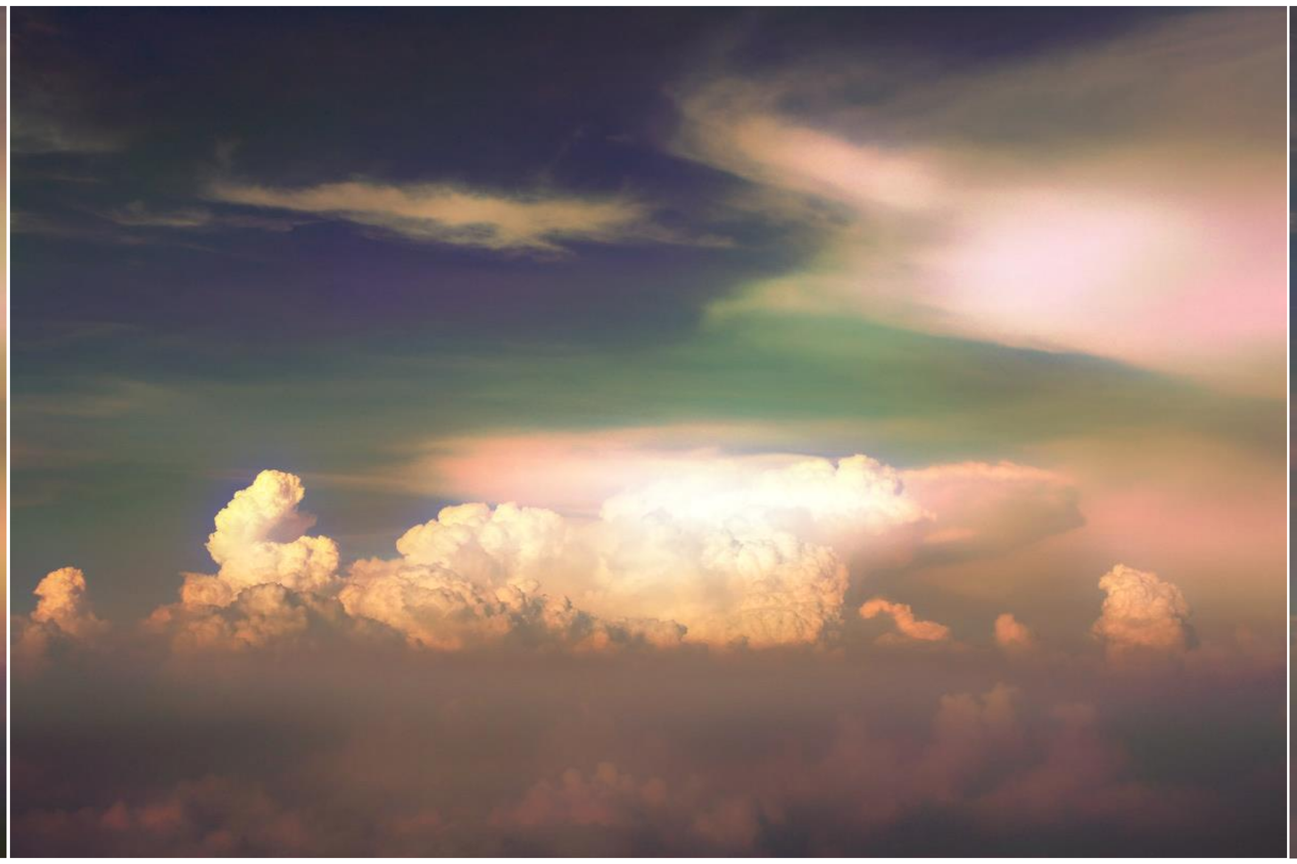
Target 40% larger

Color transfer application

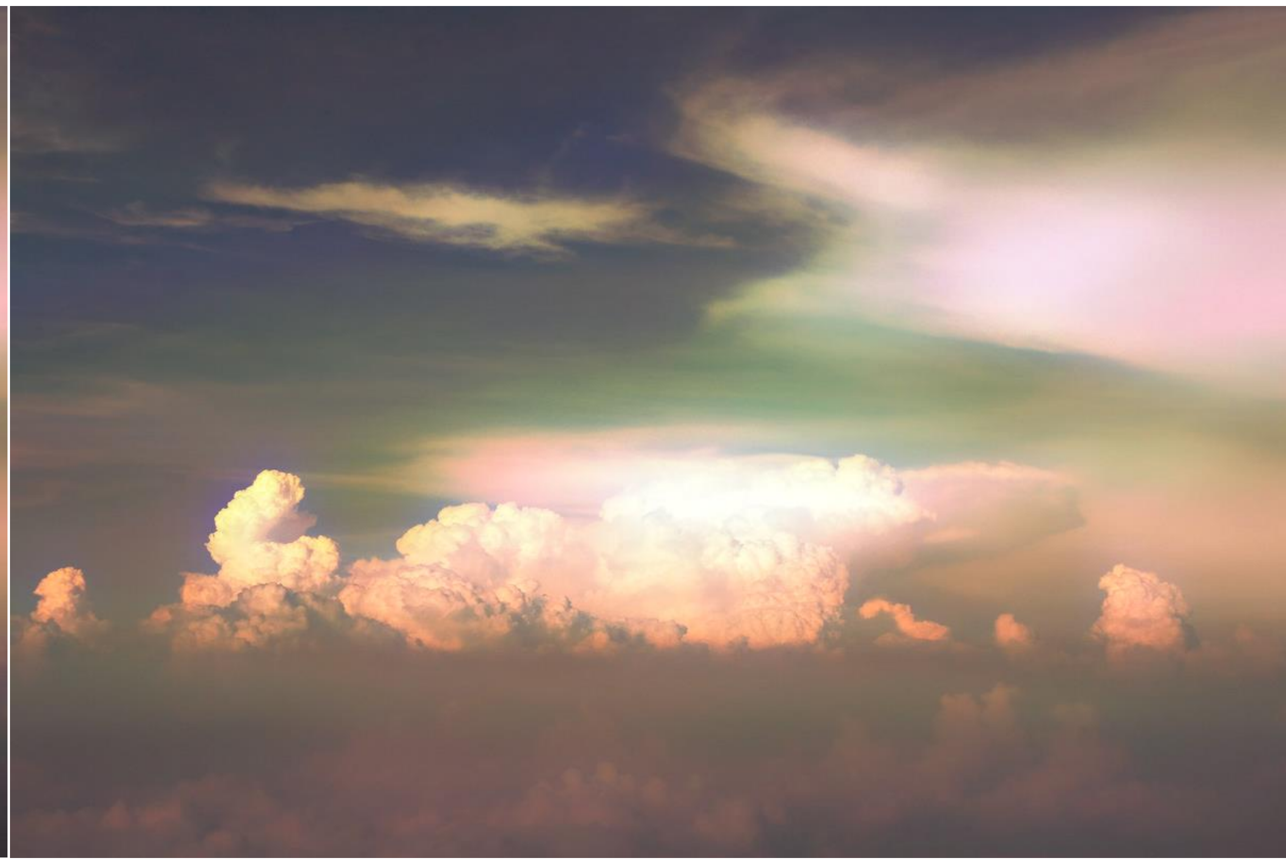




Full Transfer



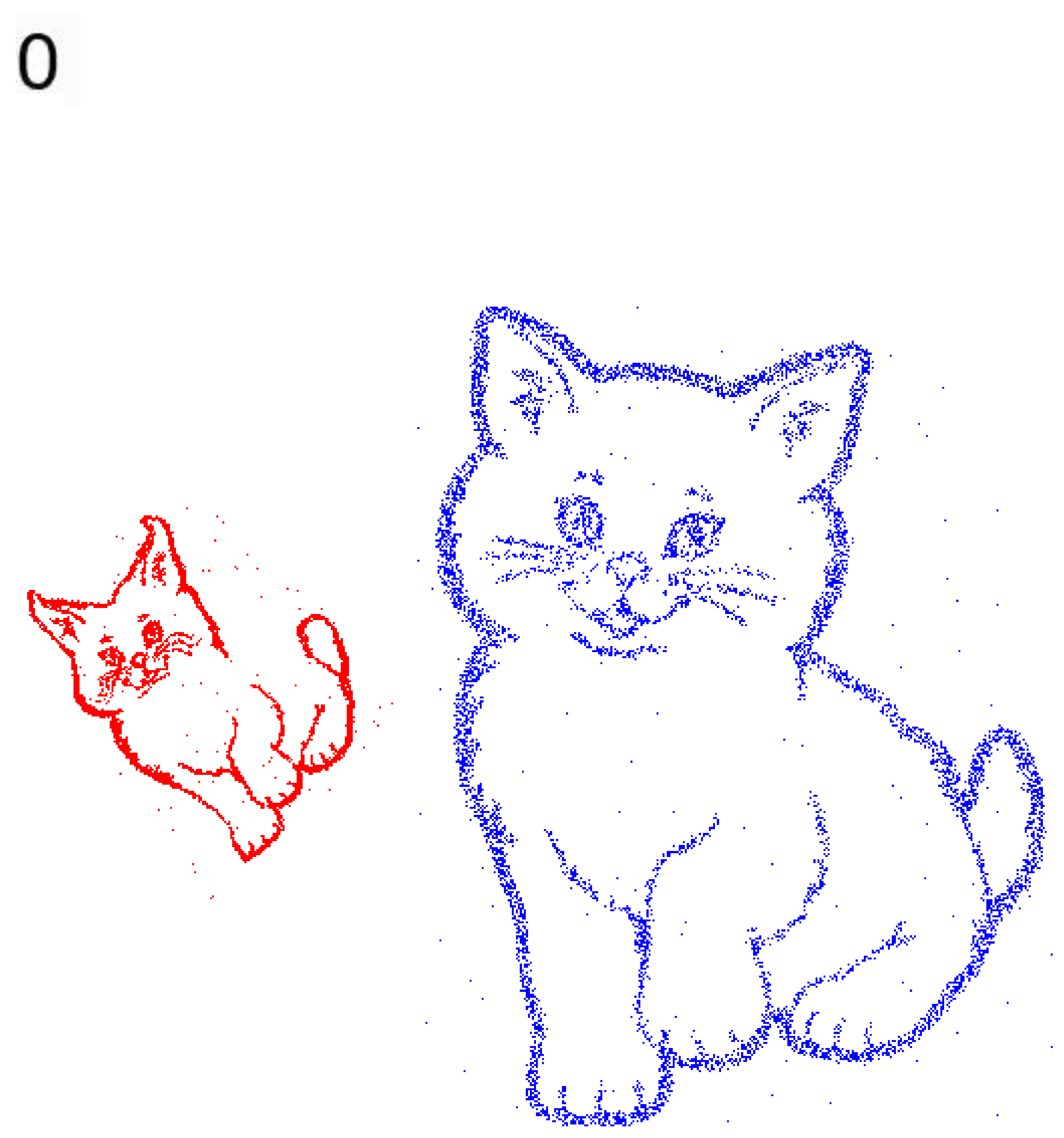
Target 20% larger



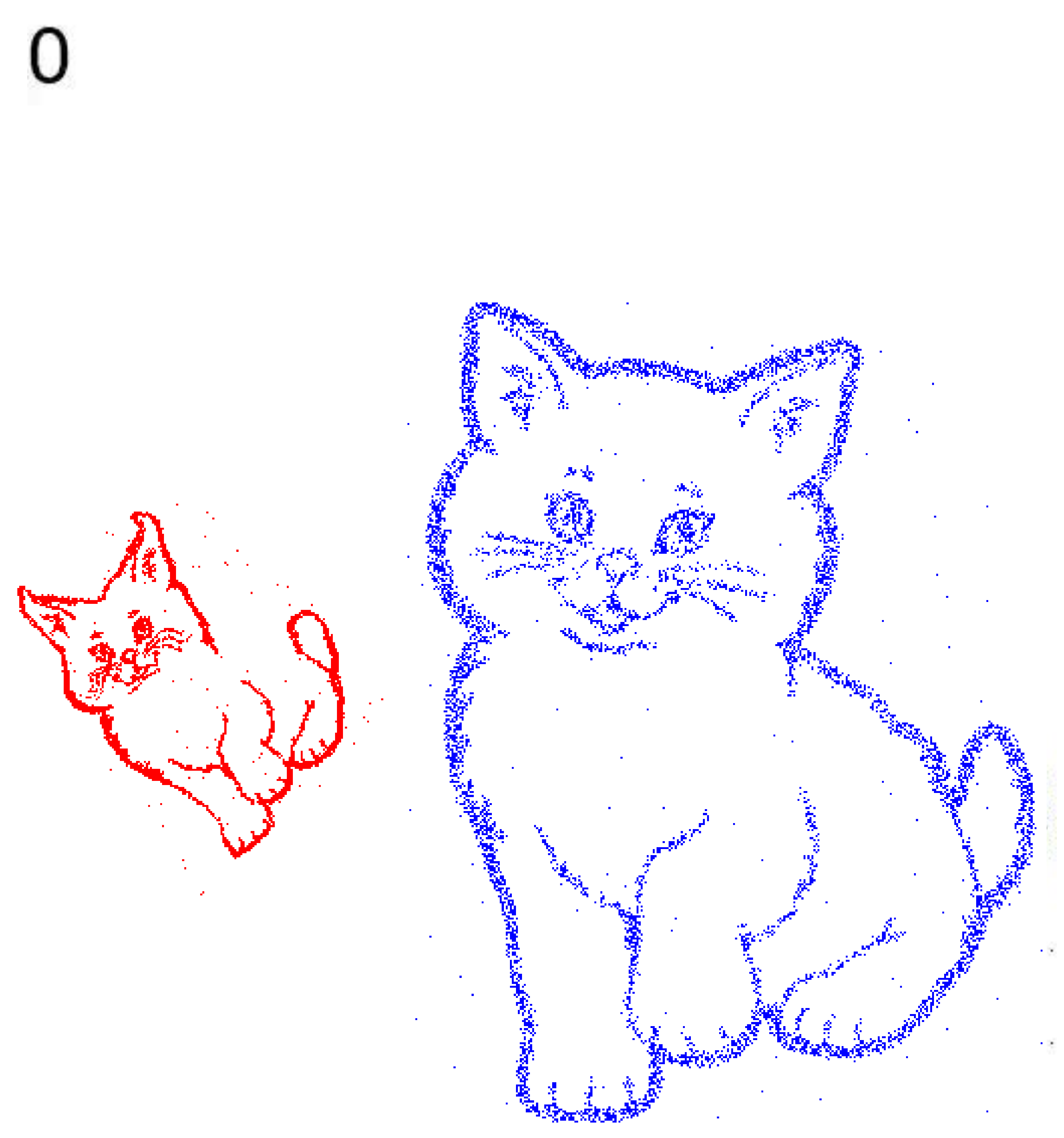
Target 40% larger

Fast Iterative Sliced Transport (FIST)

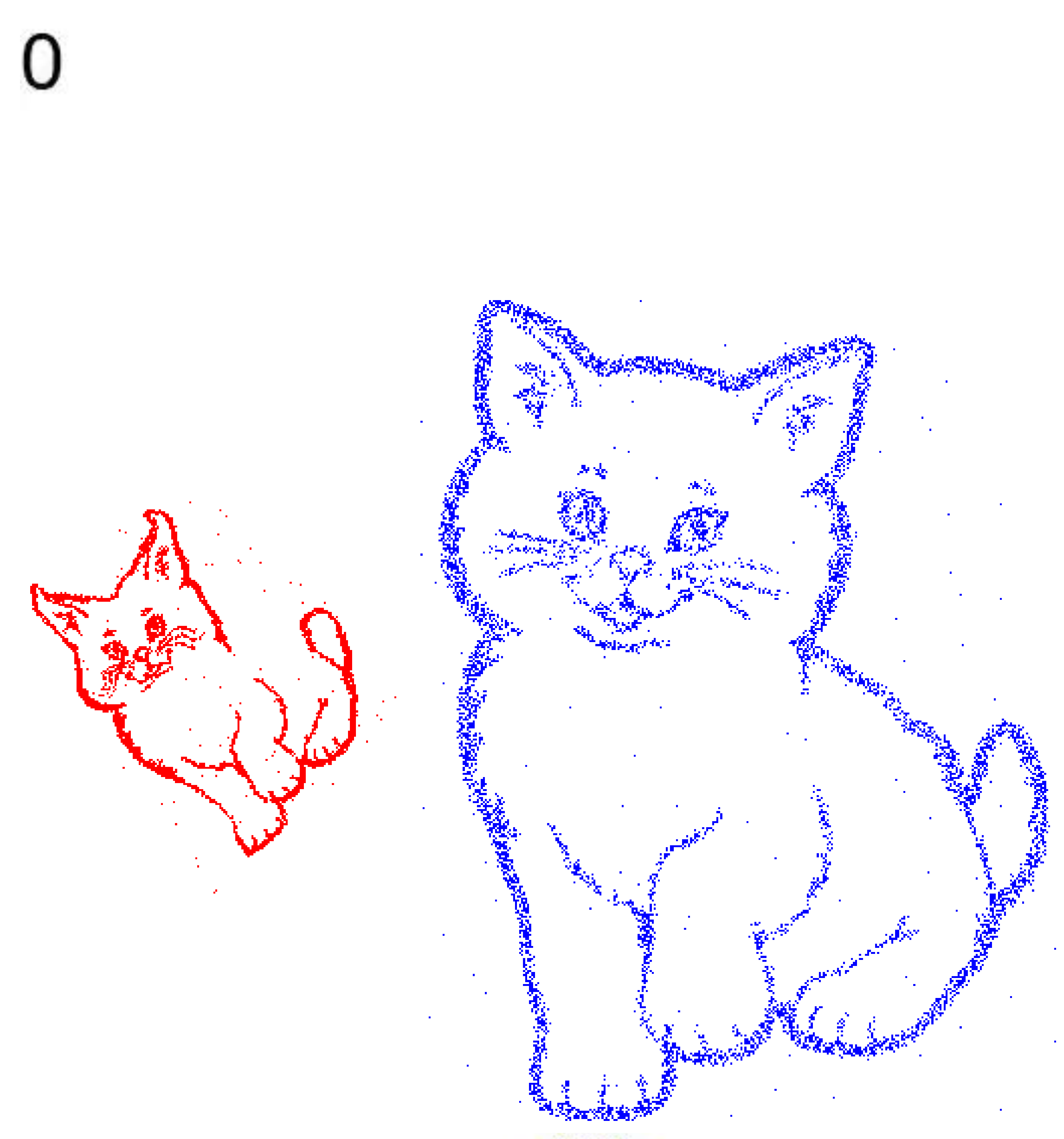
Source: 8k samples
Target: 10k samples



ICP
(0.005 s / iteration)



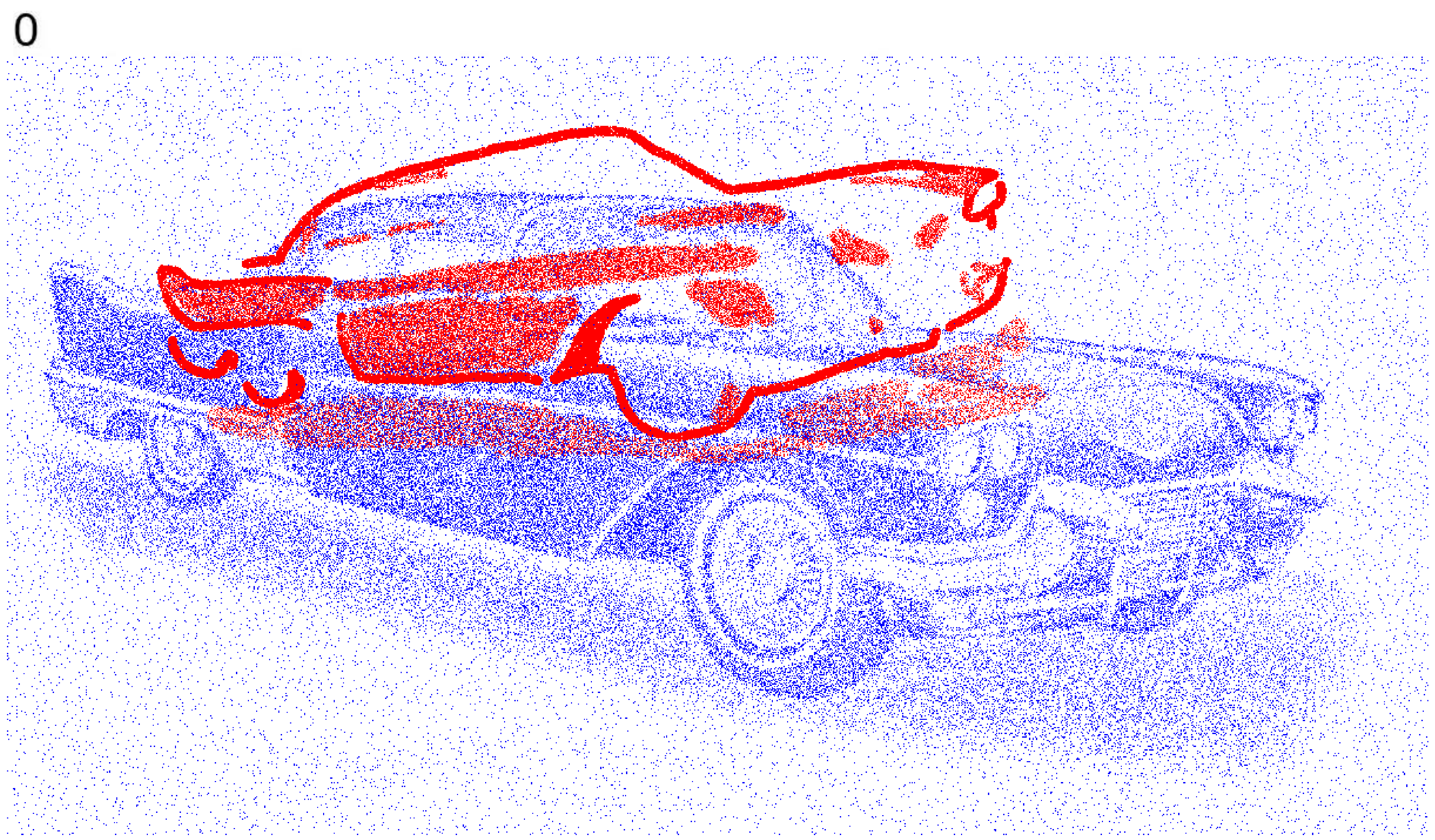
Iterative Transport
with network simplex
(40 s / iteration)



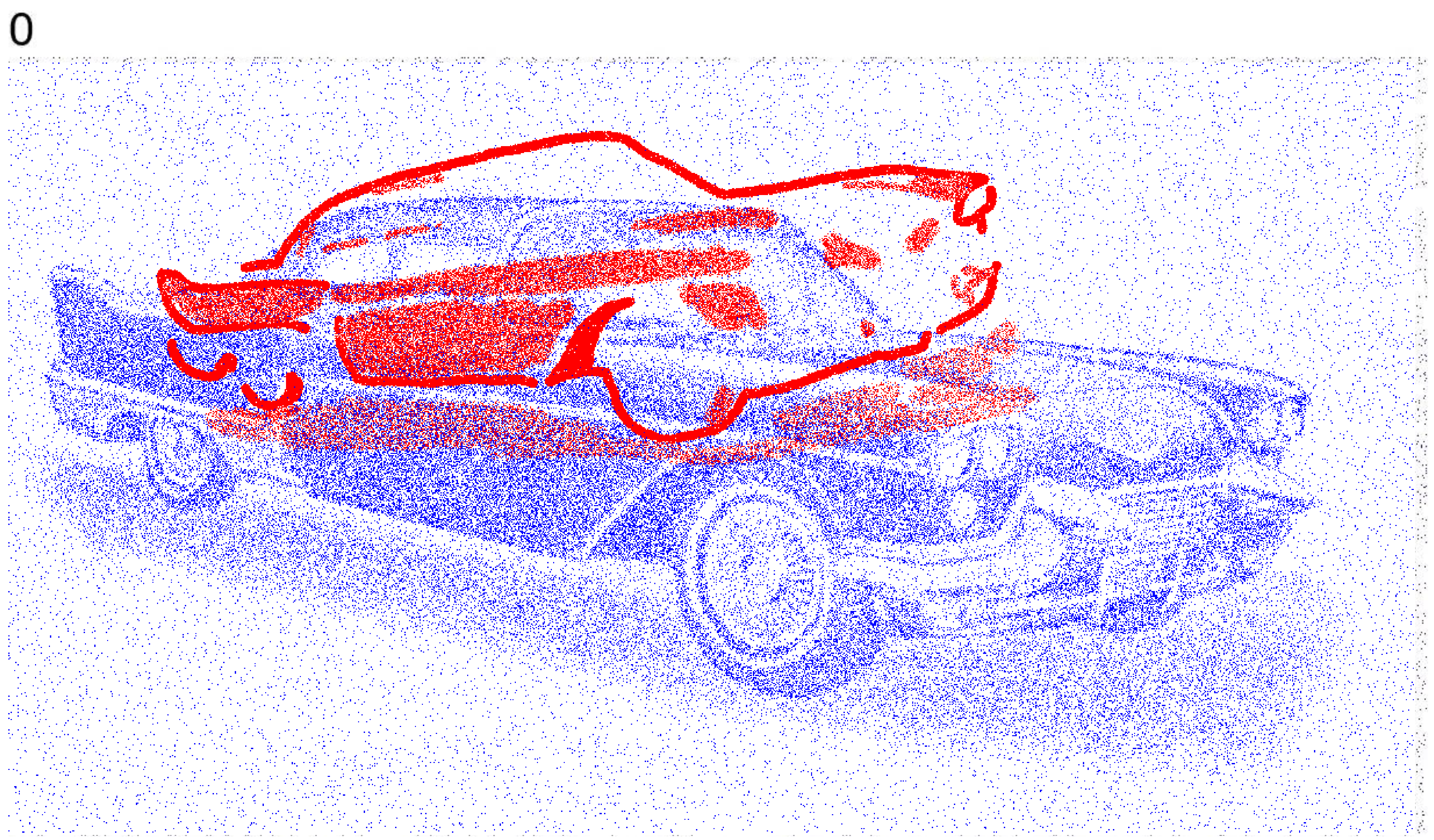
Our FIST algorithm
(0.04 s / iteration)

Source: 90k samples
Target: 100k samples

(input too large for iterative transport with network simplex)



ICP
(0.05 s / iteration)



Our FIST algorithm
(0.66 s / iteration)

Source: 90k samples
Target: 100k samples

(input too large for iterative transport with network simplex)

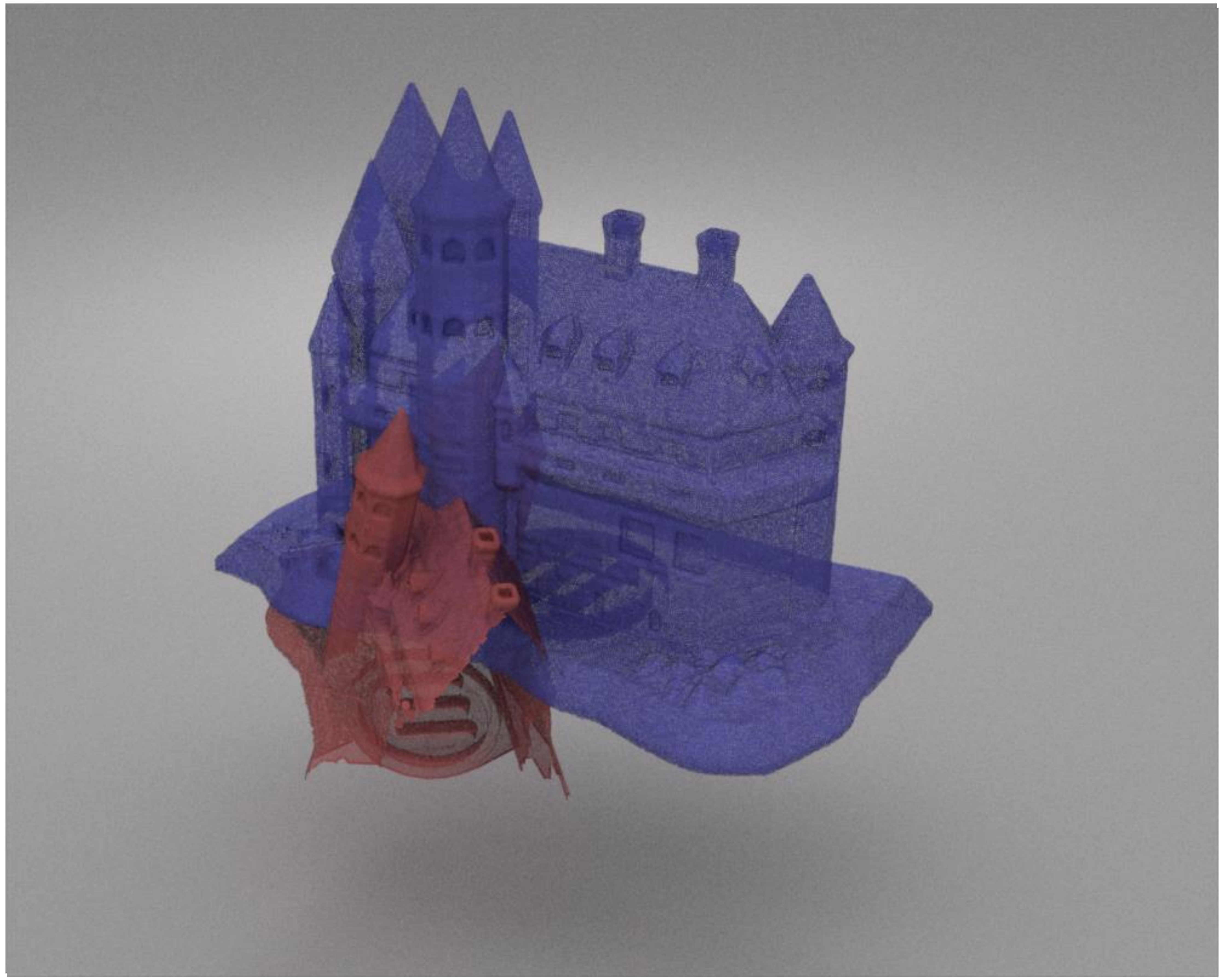
ICP
(0.05 s / iteration)

Our FIST algorithm
(0.69 s / iteration)

Source: 150k samples
Target: 200k samples

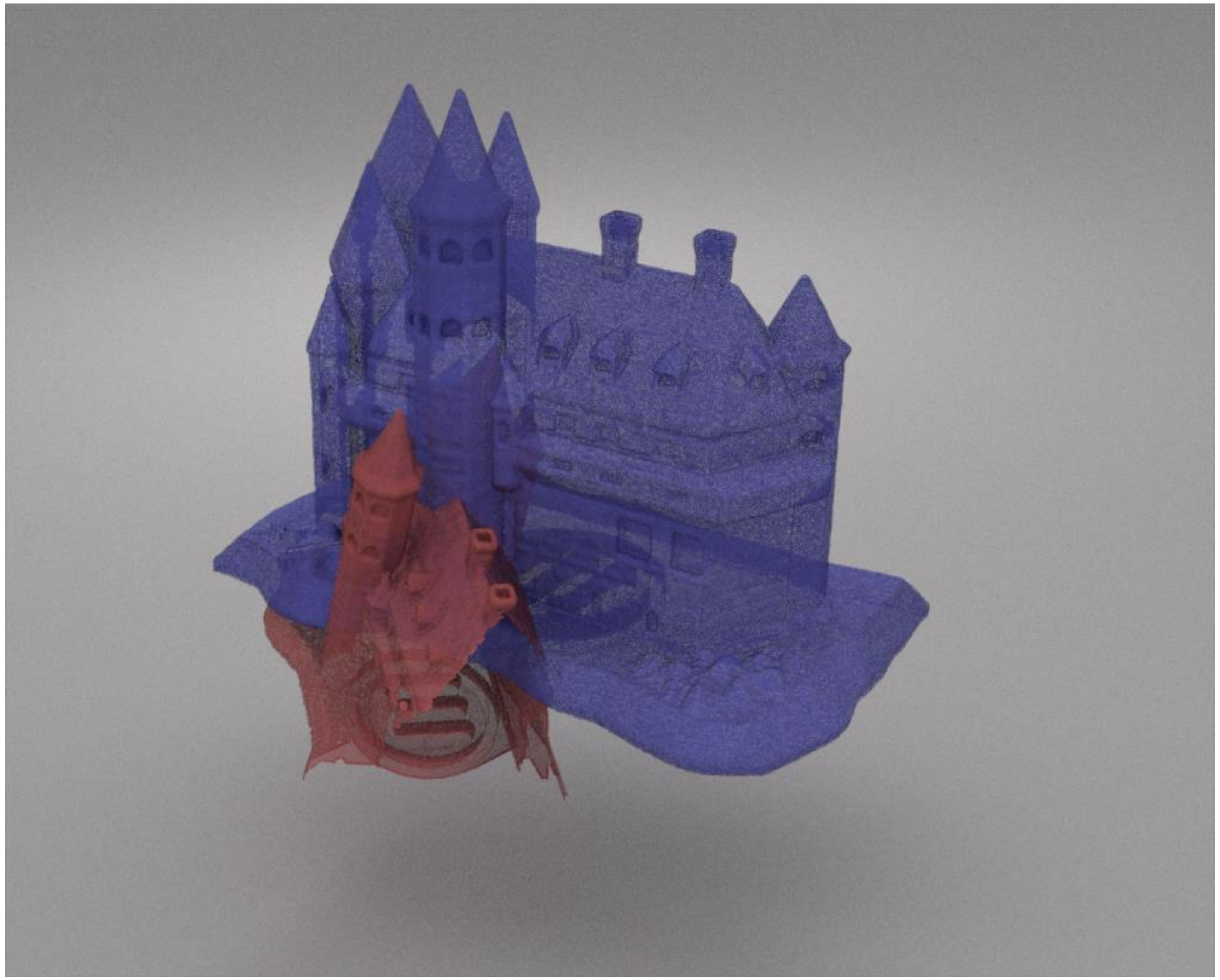
(input too large for iterative transport with network simplex)

0



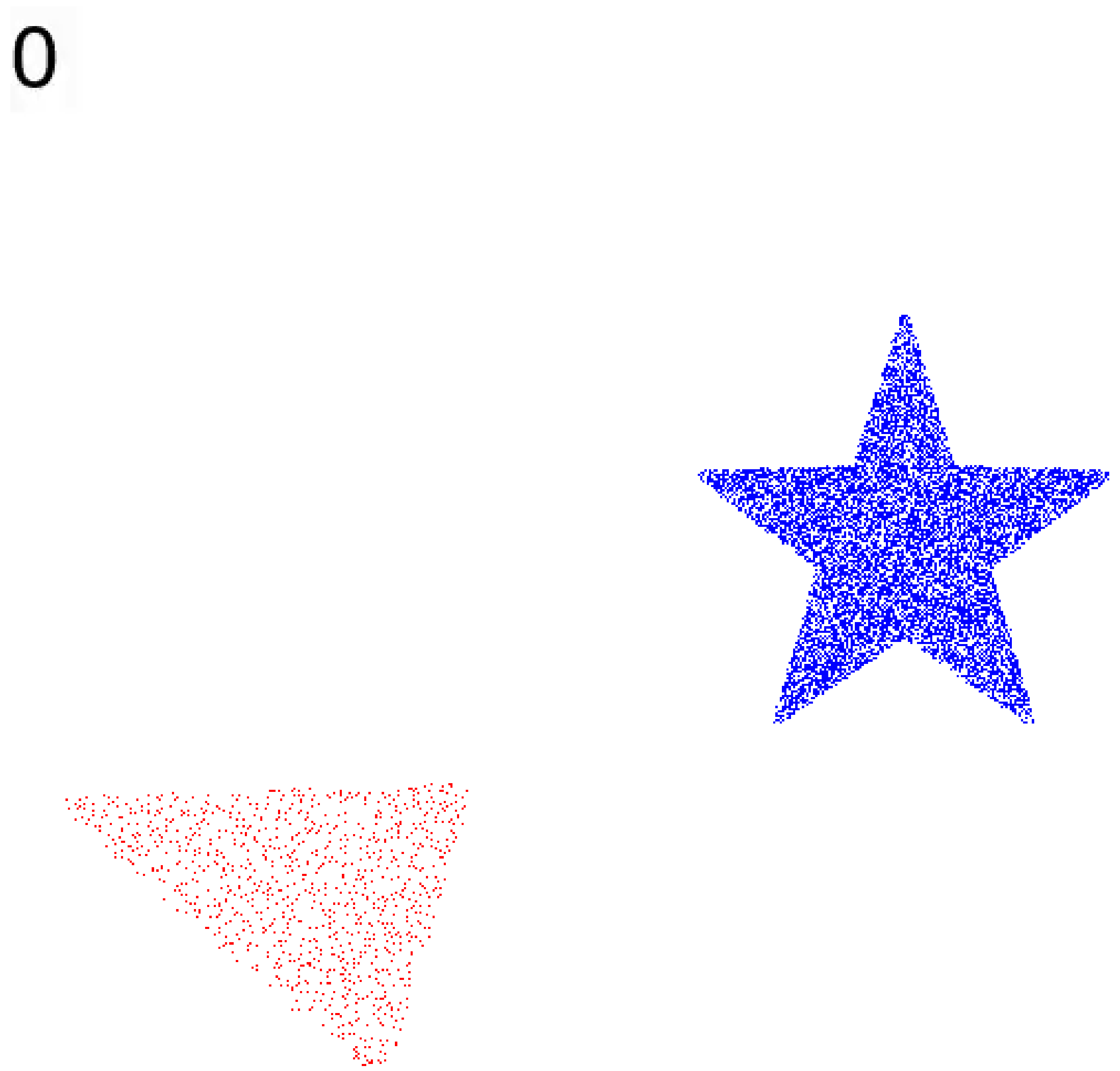
ICP
(0.09 s / iteration)

0

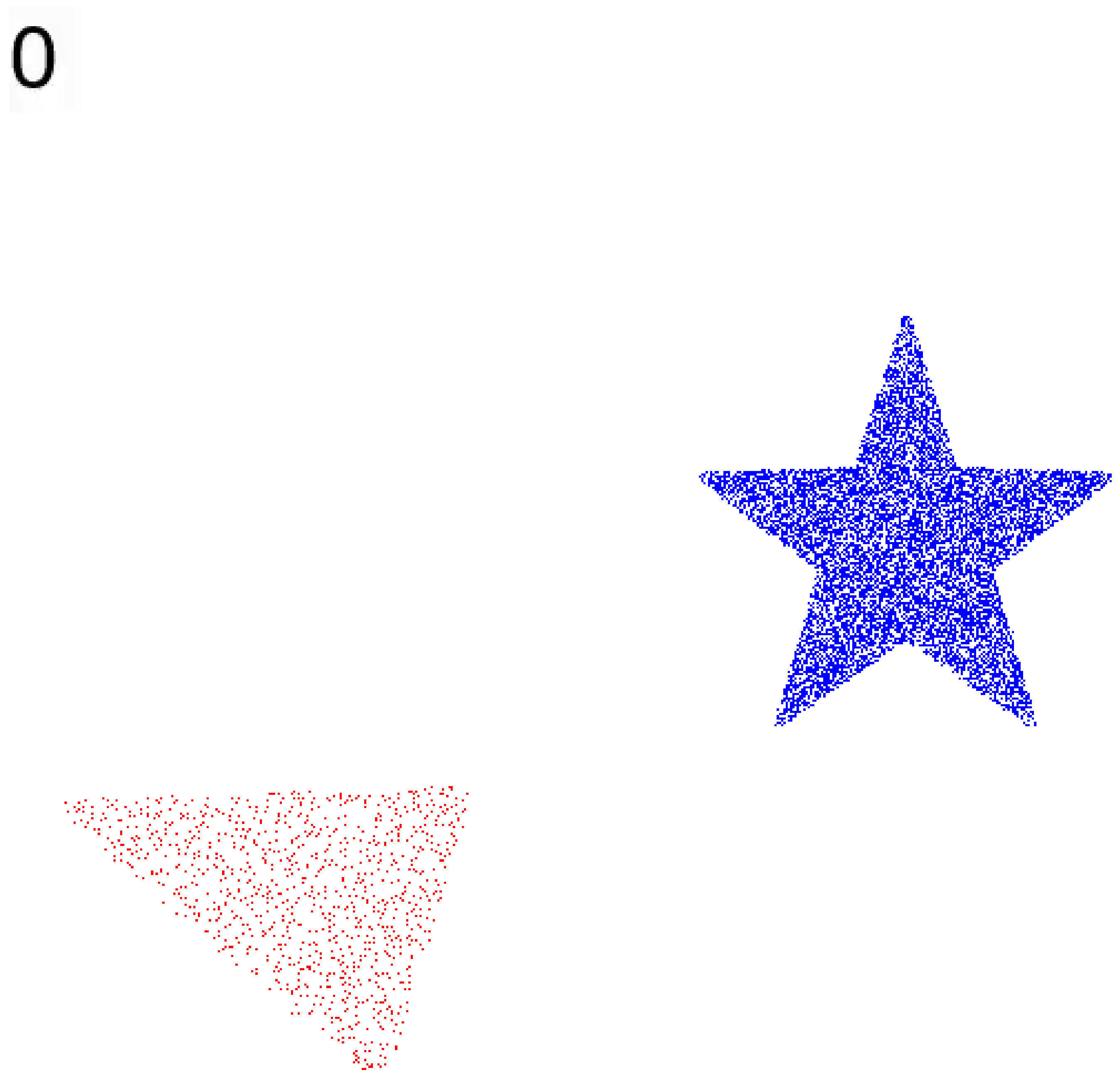


Our FIST algorithm
(2.18 s / iteration)

Failure case: the transport is optimal only on projections



Iterative Transport with Network Simplex



Our FIST algorithm

Conclusions

- Fast partial optimal transport in 1d
  - Quadratic-time algorithm (worst case)
  - Quasi-linear time decomposition
- Sliced Partial Optimal Transport
- Fast Iterative Sliced Transport
- Applications: point cloud registration, color matching
- Code available: https://perso.liris.cnrs.fr/nicolas.bonneel/spot/