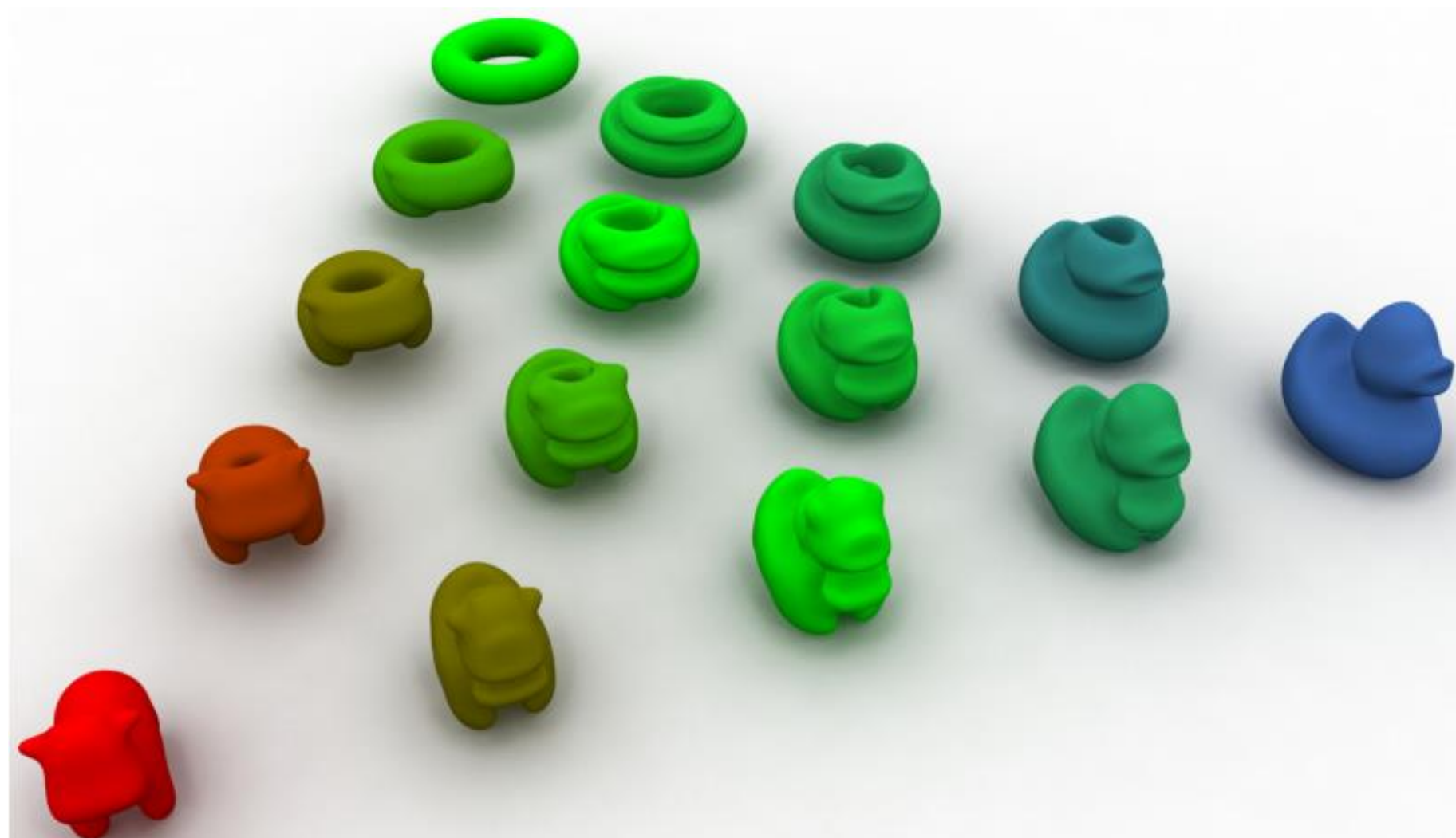


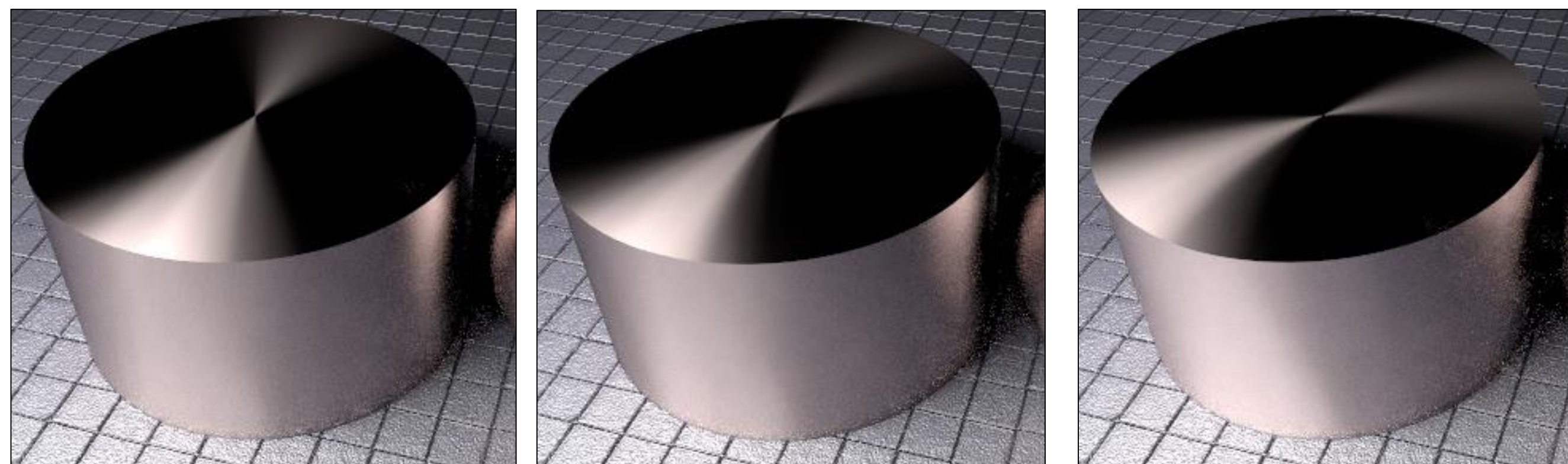
[Levy 2015]



[de Goes et al. 2012]



[Solomon et al. 2015]



[Bonneel et al. 2011]

# Matching points with optimal transport

- Monge (Linear Assignment Problem)

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

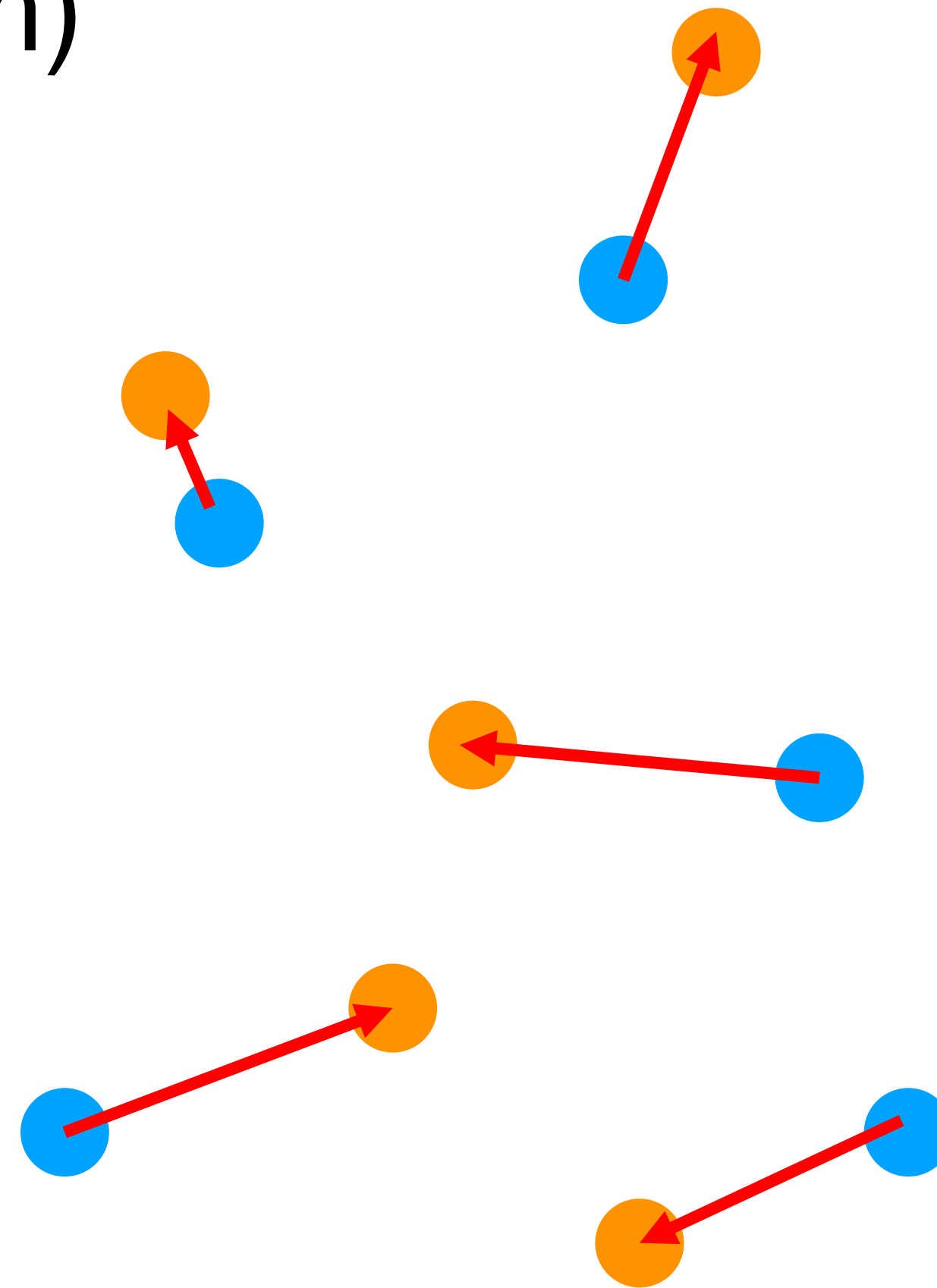
- Kantorovich

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

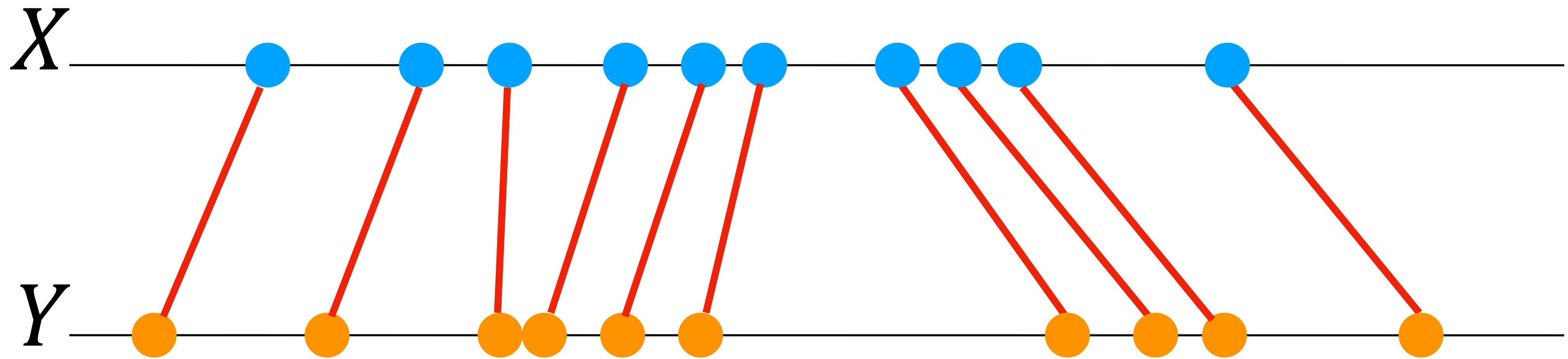
$$\text{s.t.} \quad \sum_j \pi_{i,j} = 1$$

$$\sum_i \pi_{i,j} = 1$$

$$\pi_{i,j} \geq 0$$

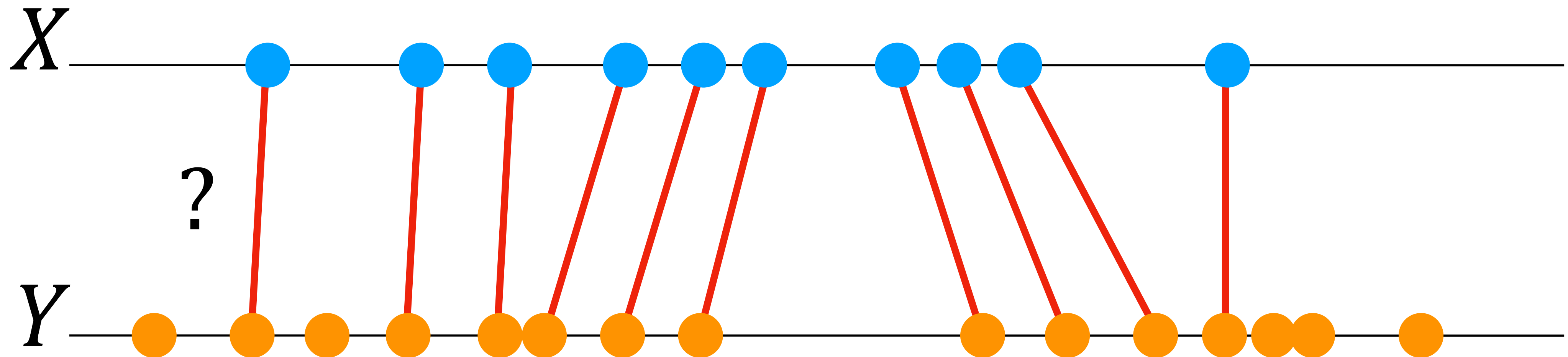


# 1-d Linear Assignment Problem is trivial\*



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

# Partial optimal assignment ?



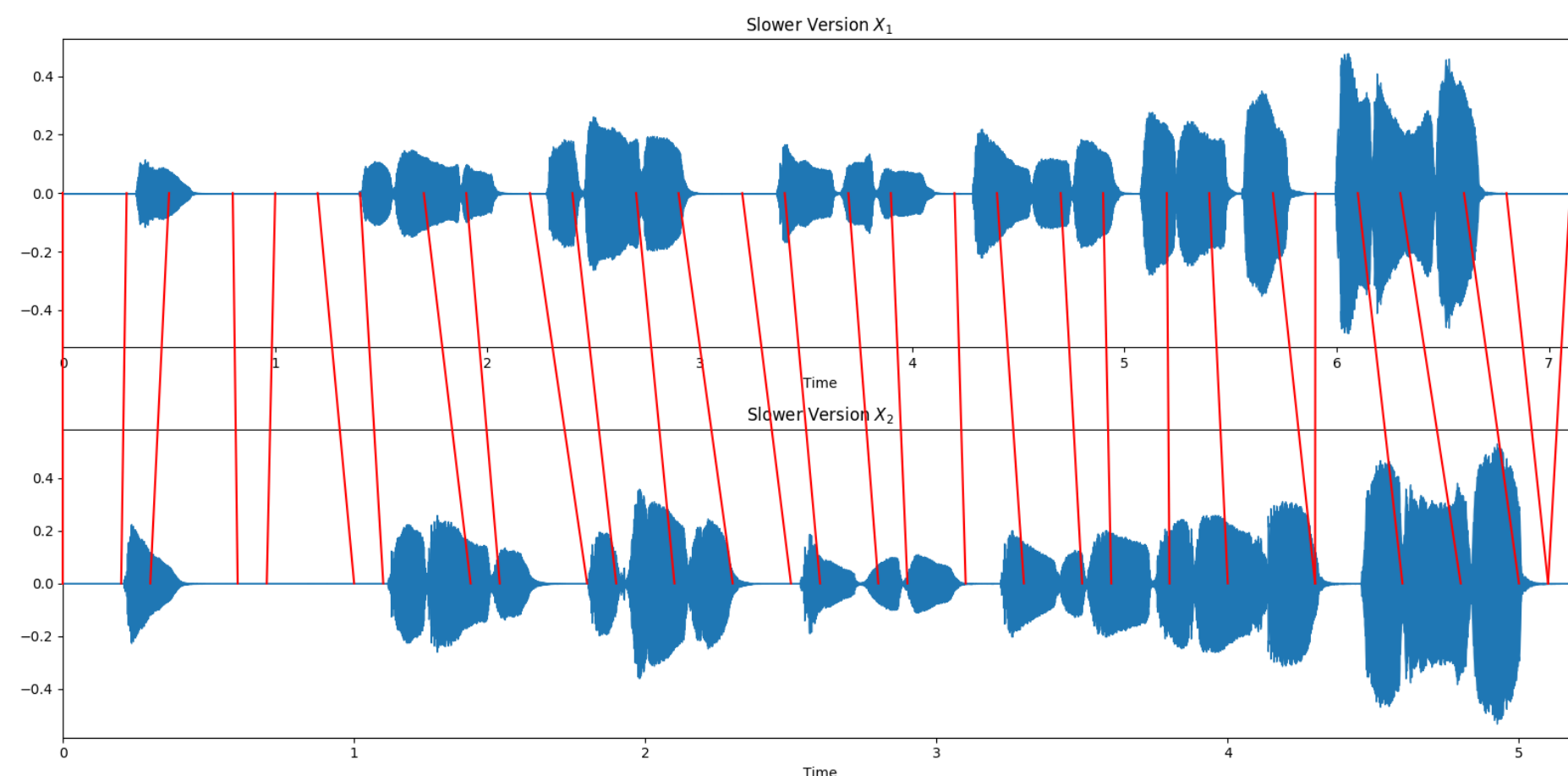
$$W(f, g) = \min \sum_{i,j} c_{i,j} \pi_{i,j} \quad \text{s.t.} \quad \begin{cases} \sum_j \pi_{i,j} = 1 \\ \sum_i \pi_{i,j} \leq 1 \\ \pi_{i,j} \geq 0 \end{cases}$$

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

# Similar problems

- DNA sequence alignment
- Text alignment
- Music synchronization
- ...

Scarites	C	T	T	A	G	A	T	C	G	T	A	C	C	A	A	-	-	-	A	A	T	A	T	T	A	C
Carenum	C	T	T	A	G	A	T	C	G	T	A	C	C	A	C	A	-	T	A	C	-	T	T	T	A	C
Pasimachus	A	T	T	A	G	A	T	C	G	T	A	C	C	A	C	T	A	T	A	A	G	T	T	T	A	C
Pheropsophus	C	T	T	A	G	A	T	C	G	T	T	C	C	A	C	-	-	-	A	C	A	T	A	T	A	C
Brachinus armiger	A	T	T	A	G	A	T	C	G	T	A	C	C	A	C	-	-	-	A	T	A	T	A	T	T	C
Brachinus hirsutus	A	T	T	A	G	A	T	C	G	T	A	C	C	A	C	-	-	-	A	T	A	T	A	T	A	C
Aptinus	C	T	T	A	G	A	T	C	G	T	A	C	C	A	C	-	-	-	A	C	A	A	T	T	A	C
Pseudomorpha	C	T	T	A	G	A	T	C	G	T	A	C	C	-	-	-	-	-	A	C	A	A	A	T	A	C



```
File Edit Changes View Tabs Help
[tecmint] functio...d] functions.php x
/TecMint-WpUseOf-Site-Backups/tecmint Browse...
/* -----
// Content width
if ( !isset( $content_width ) ) { $content_width = 7

/* Theme setup
/* ----- */
if ( ! function_exists( 'alx_setup' ) ) {

function alx_setup() {
// Enable title tag
add_theme_support( 'title-tag' );

// Enable automatic feed links
add_theme_support( 'automatic-feed-links' );

// Enable featured image
add_theme_support( 'post-thumbnails' );

// Enable post format support
add_theme_support( 'post-formats', array( 'audio

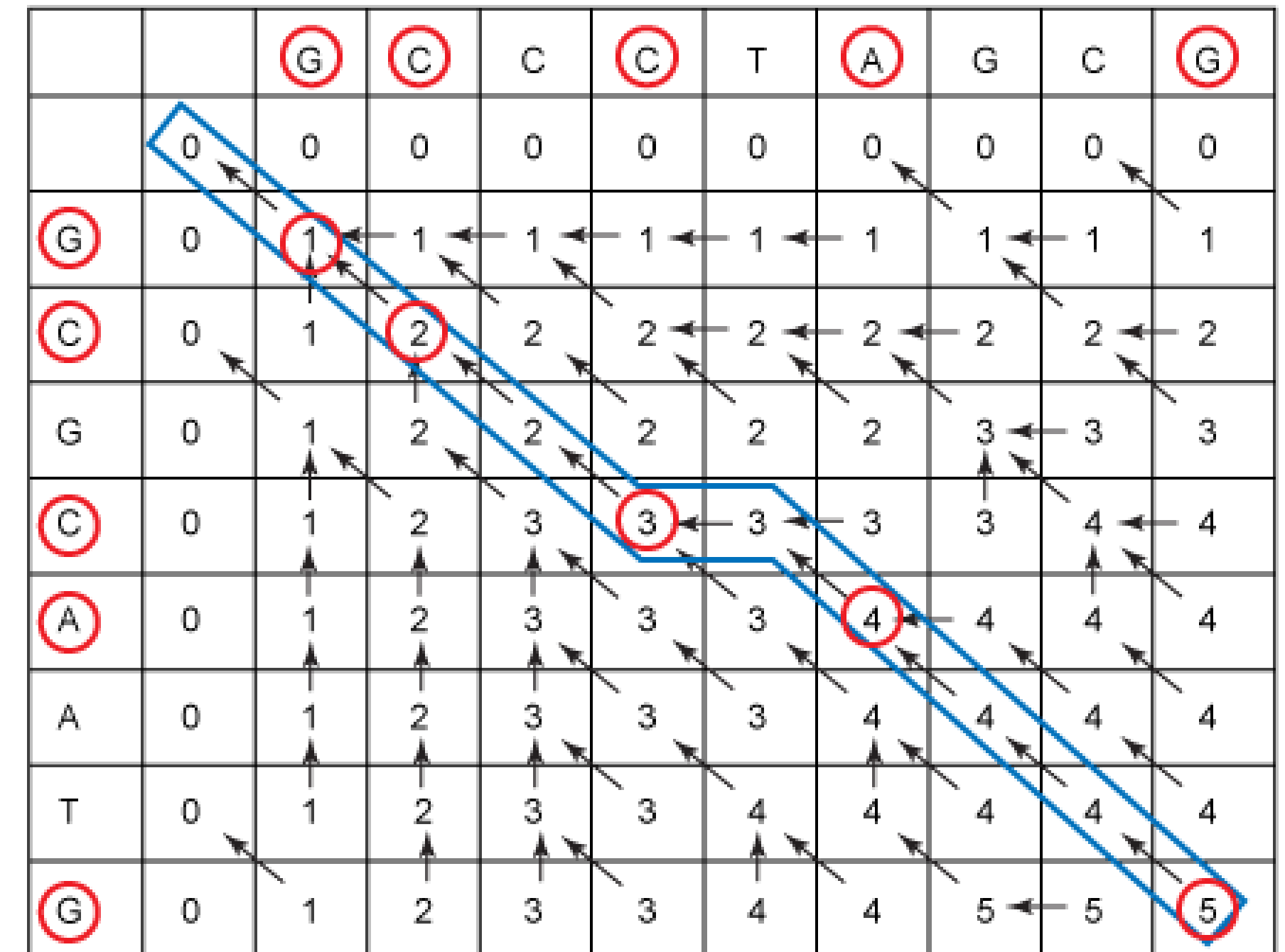
// Declare WooCommerce support
add_theme_support( 'woocommerce' );

// Thumbnail sizes
add_image_size( 'thumb-small', 160, 160, true );
add_image_size( 'thumb-medium', 520, 245, true );
add_image_size( 'thumb-large', 720, 340, true );

// Custom menu areas
```

# Existing solutions

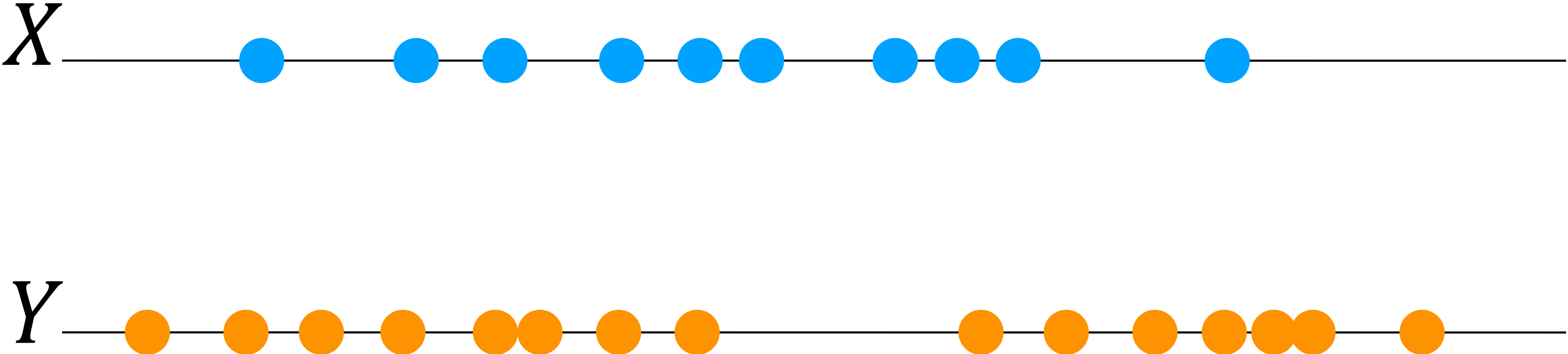
- Dynamic Time Warping
  - Solves a dynamic programming problem
  - Smith–Waterman algorithm, Needleman–Wunsch algorithm  $O(N^2)$  space and time
  - Hirschberg's algorithm  $O(N^2)$  time,  $O(N)$  space
- All end up doing variants of
  - $A_{i,j} = \min(A_{i-1,j-1} + cost, A_{i-1,j} + cost', A_{i,j-1}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



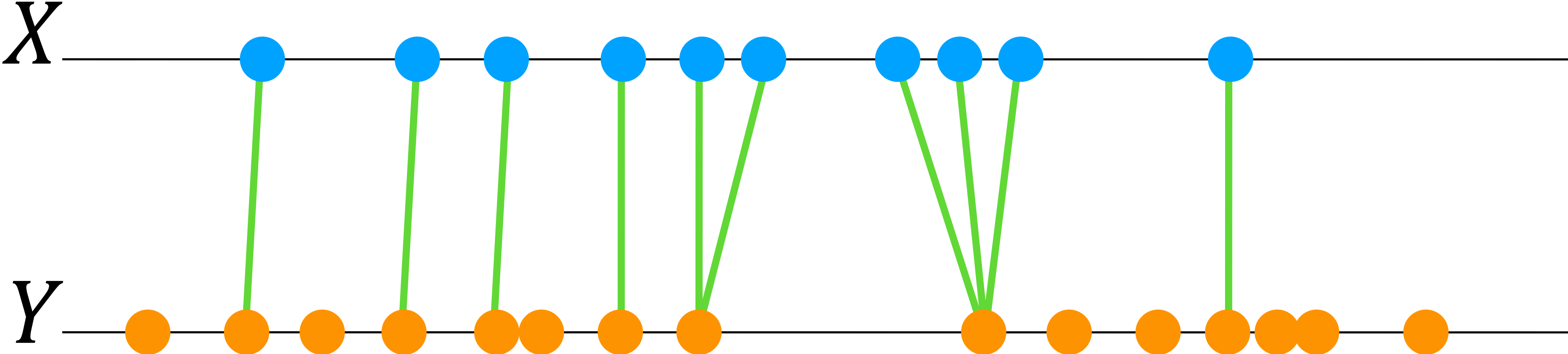
Quadratic time complexity  
algorithm (linear space)



# Quadratic time complexity algorithm (linear space)

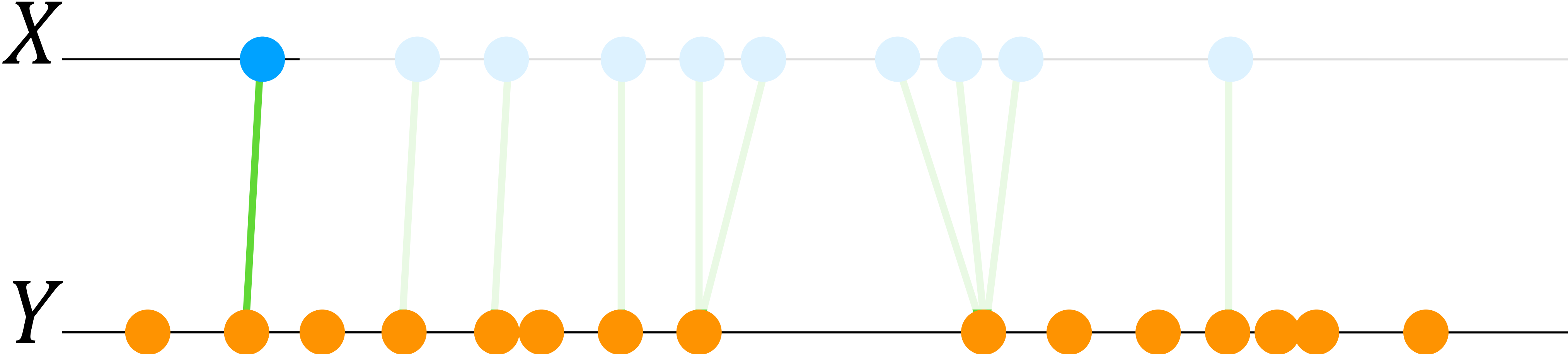


# Quadratic time complexity algorithm (linear space)



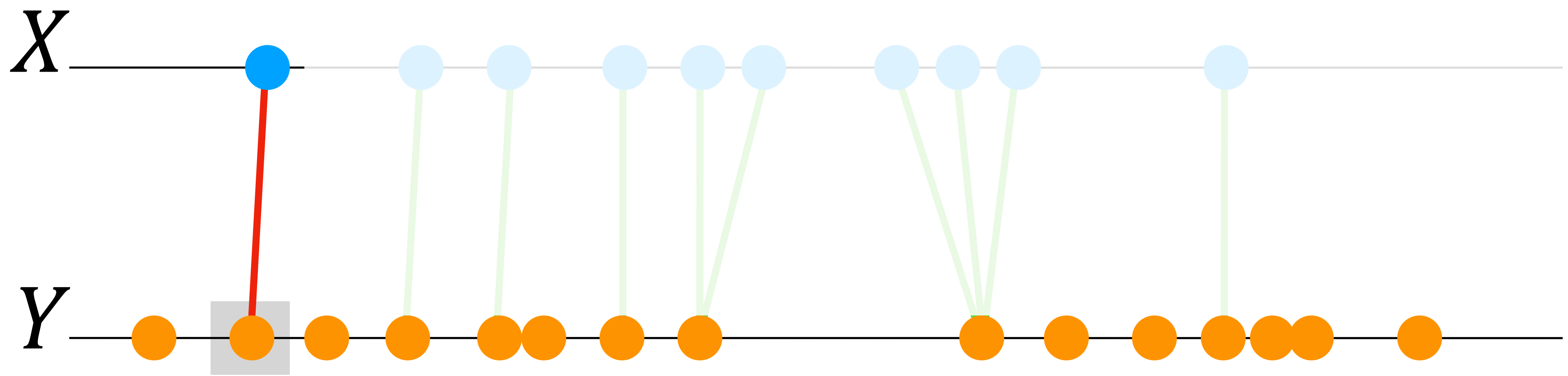
— Euclidean Nearest Neighbor assignment

# Quadratic time complexity algorithm (linear space)



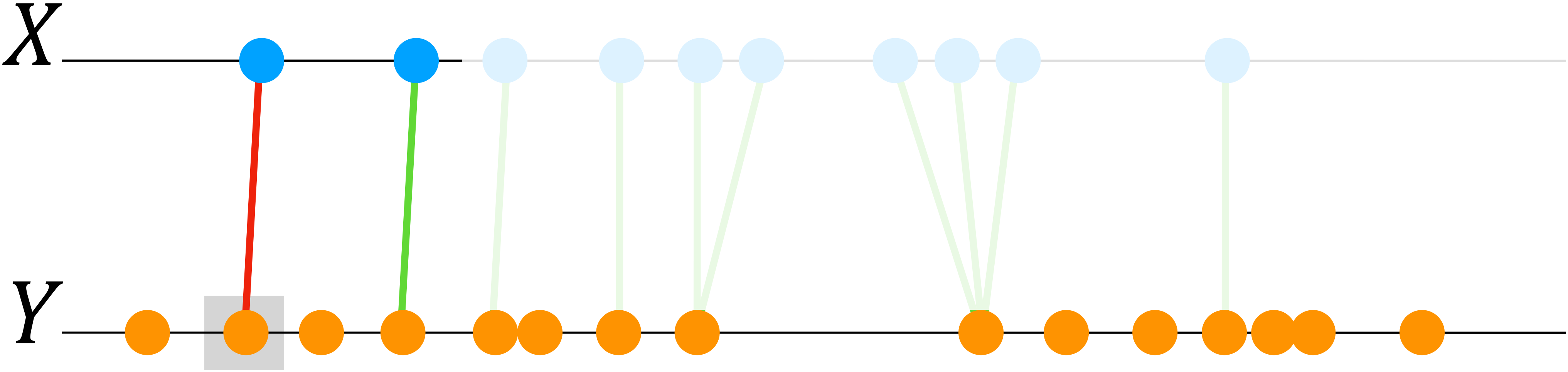
**—** Euclidean Nearest Neighbor assignment

# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)

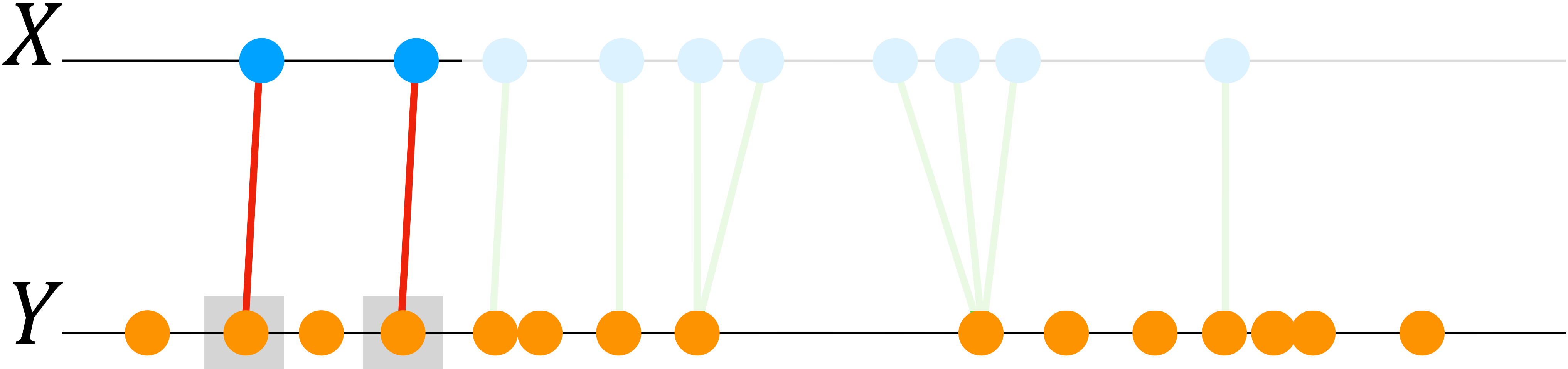


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

— Optimal Transport assignment

# Quadratic time complexity algorithm (linear space)

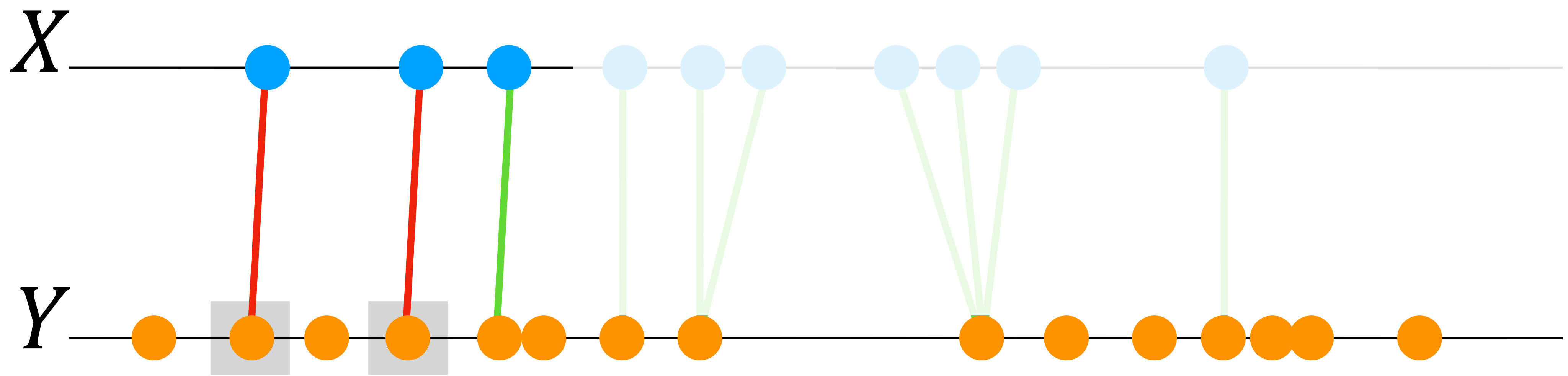


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

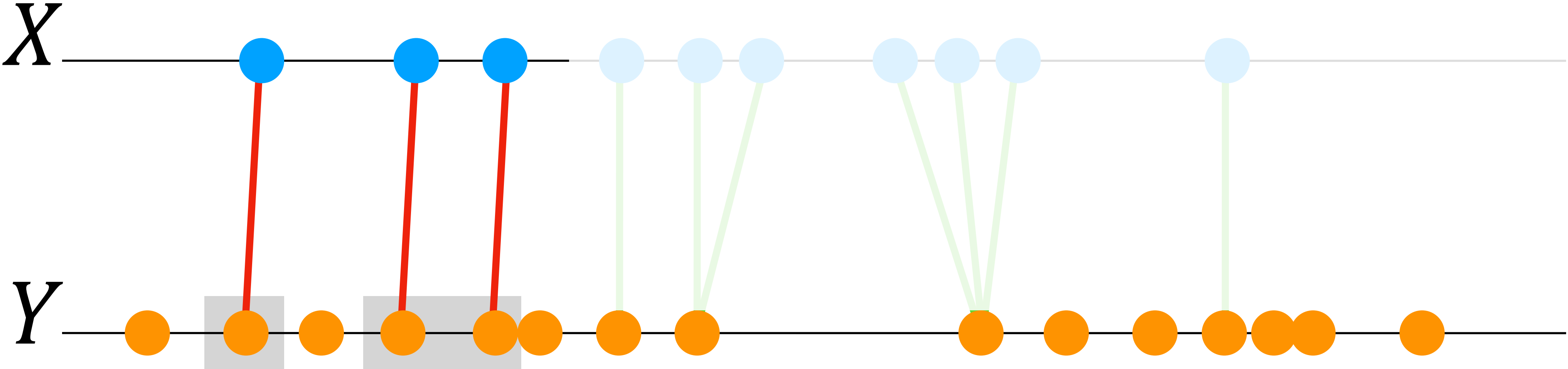
— Optimal Transport assignment

# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)



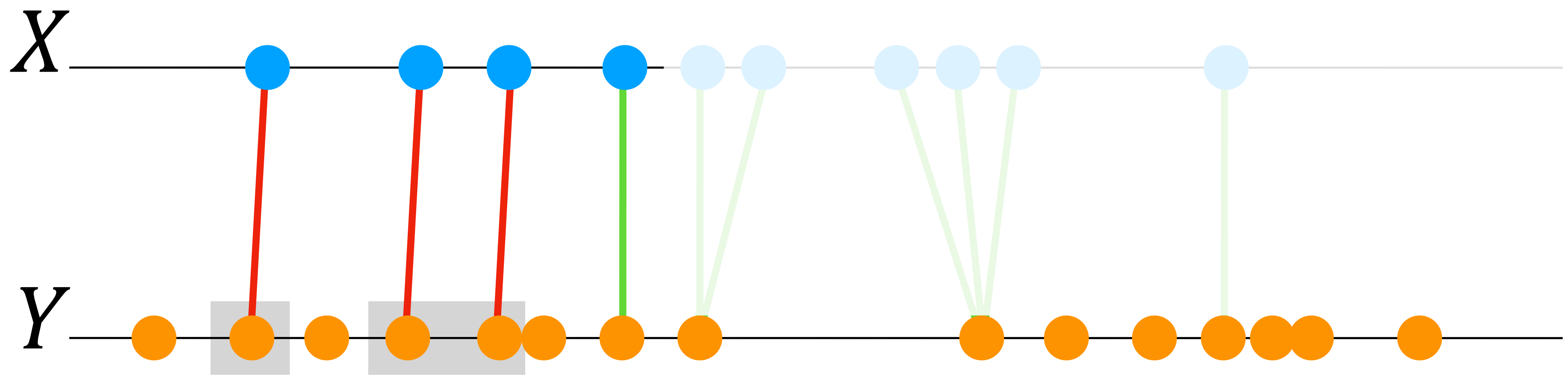
— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

— Optimal Transport assignment

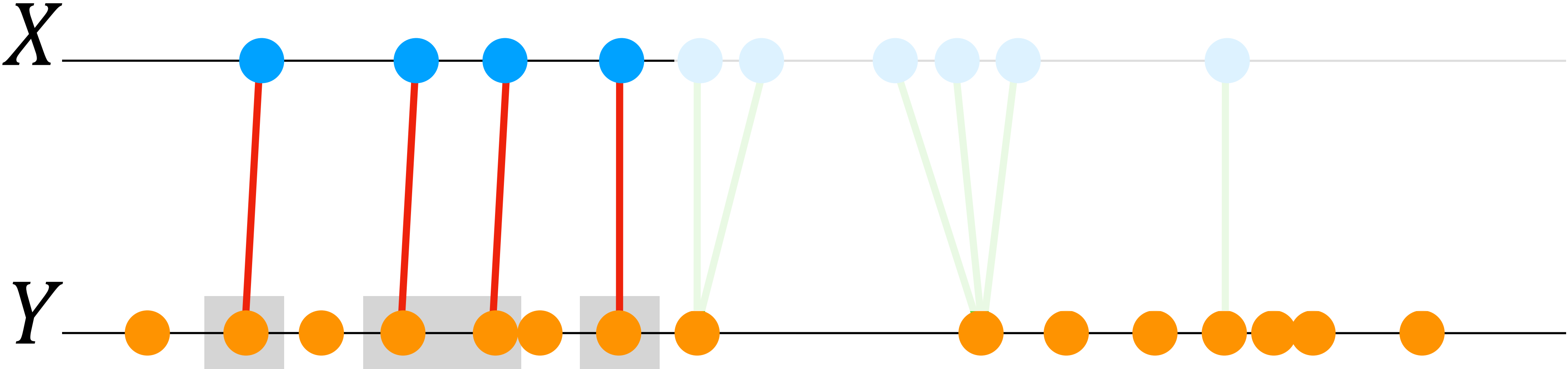


# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)

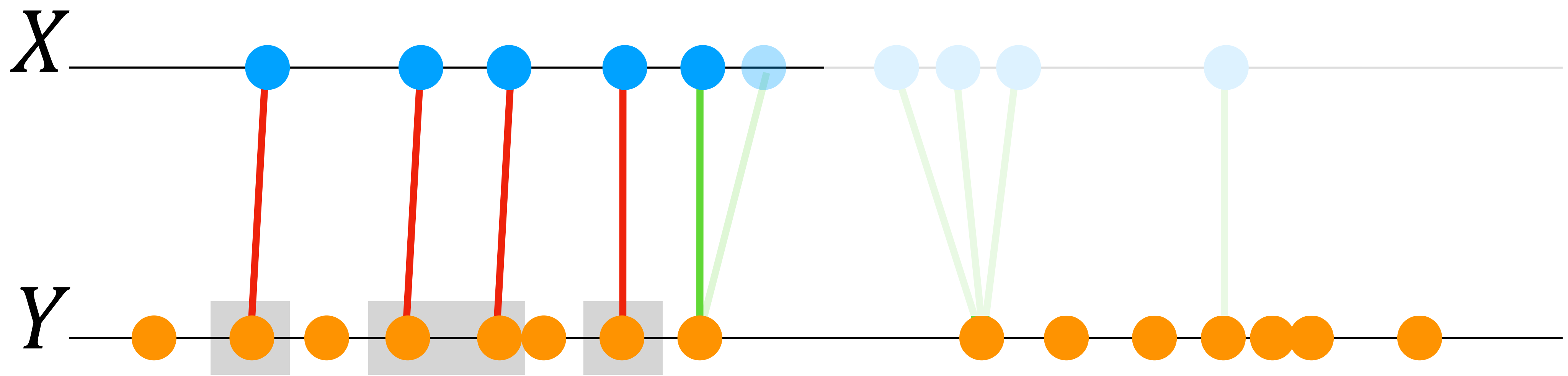


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

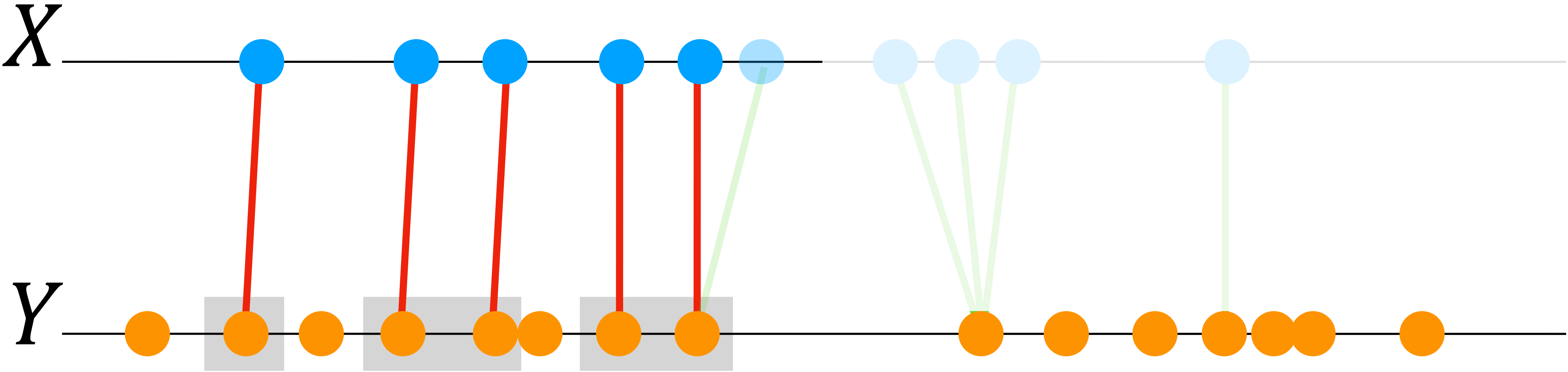
— Optimal Transport assignment

# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)

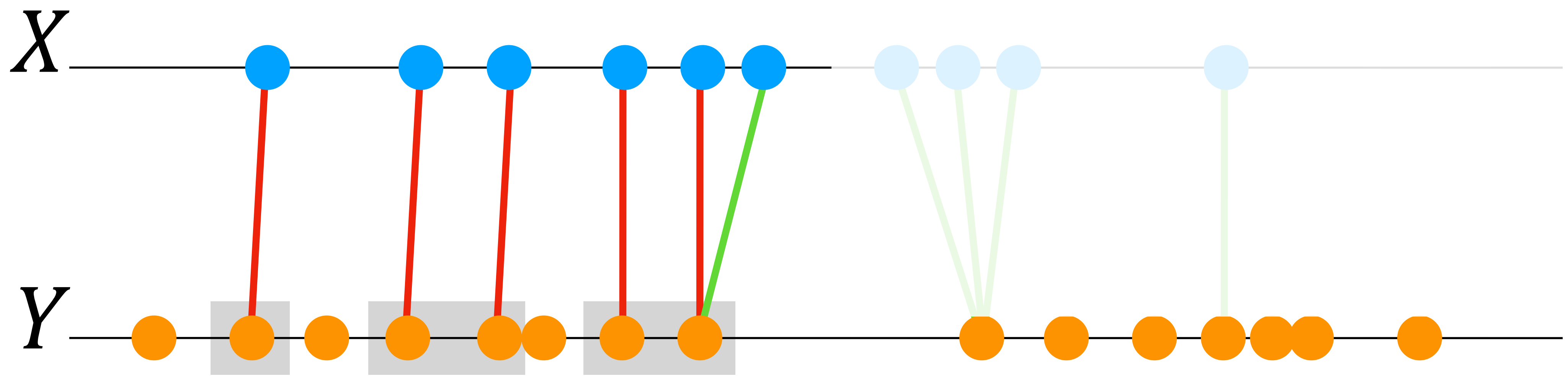


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

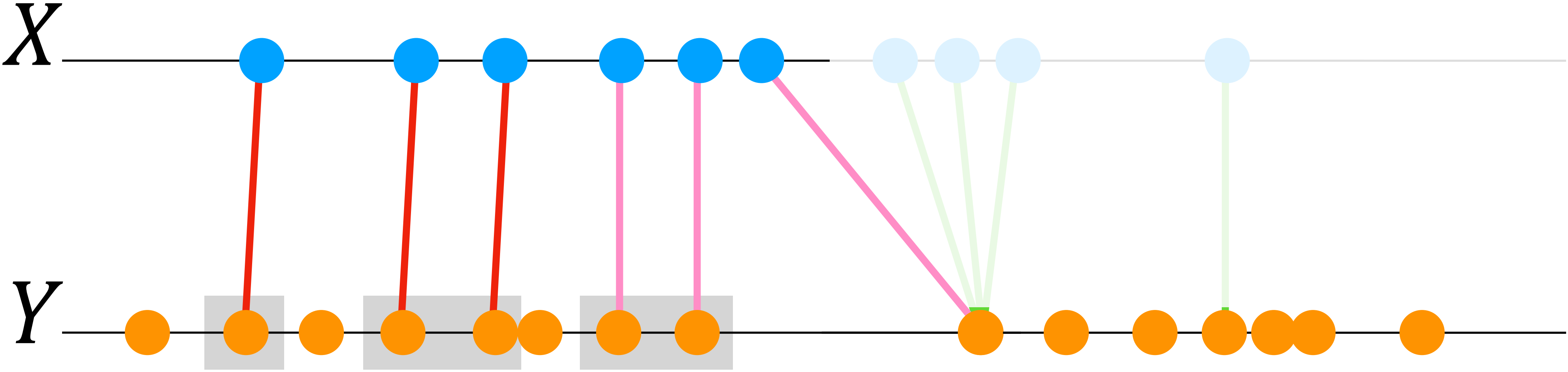
— Optimal Transport assignment

# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)

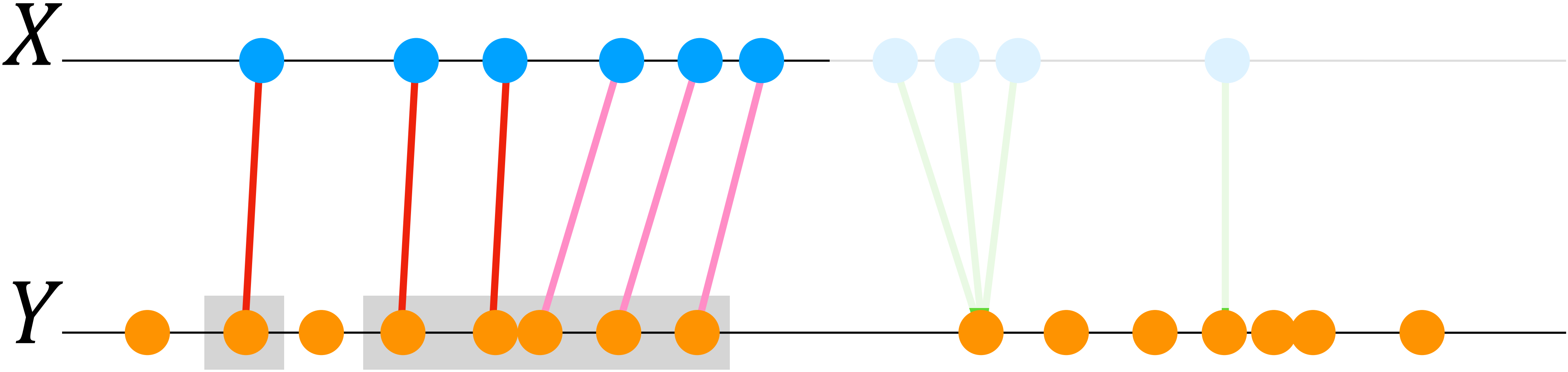


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

— Optimal Transport assignment

# Quadratic time complexity algorithm (linear space)

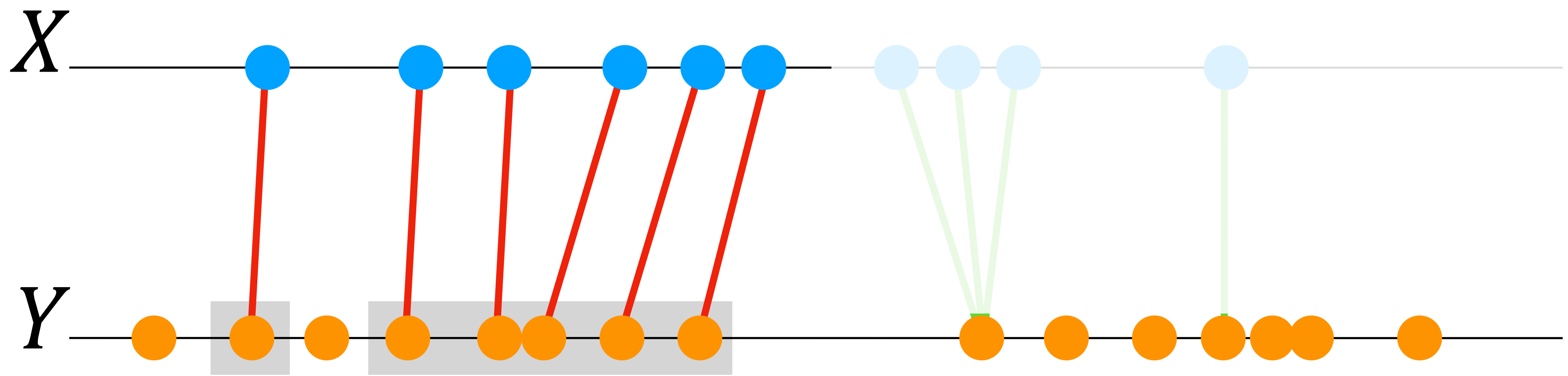


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

— Optimal Transport assignment

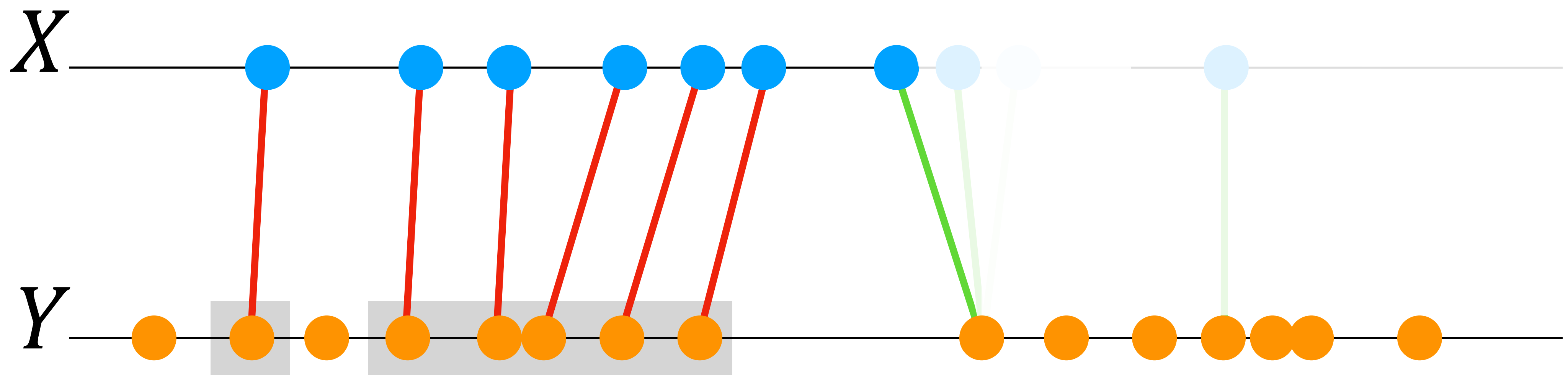
# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

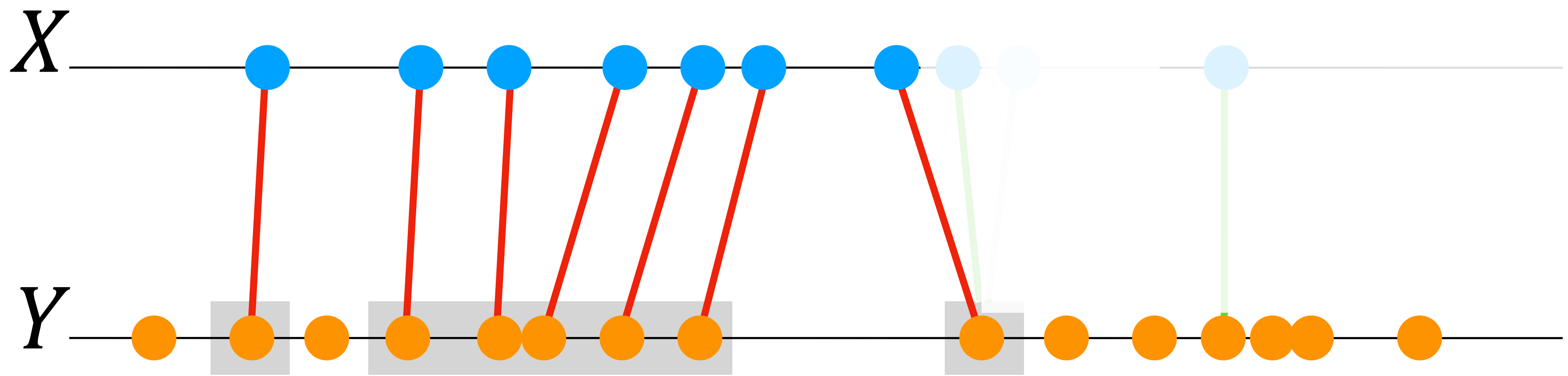


# Quadratic time complexity algorithm (linear space)



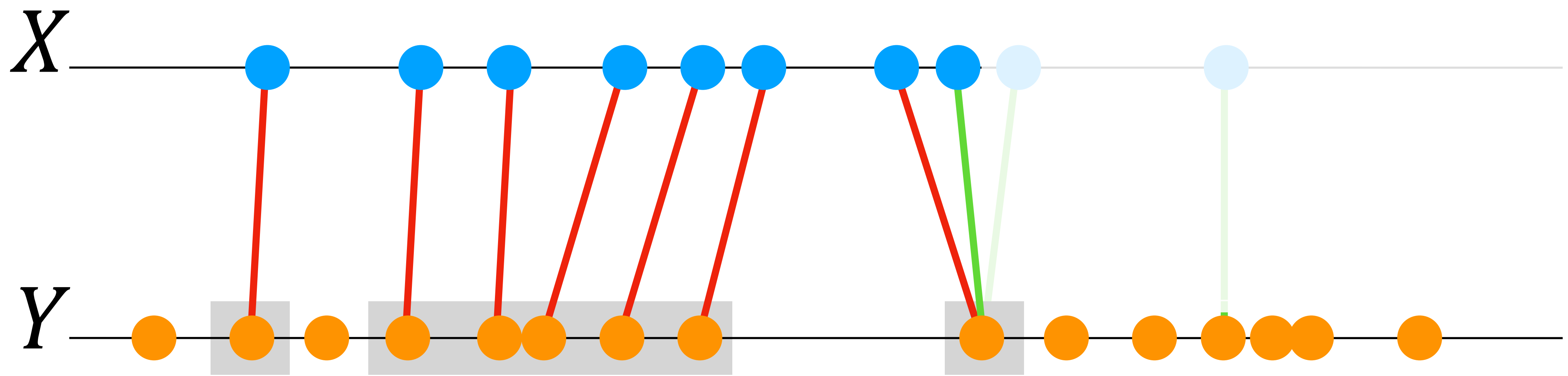
-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)



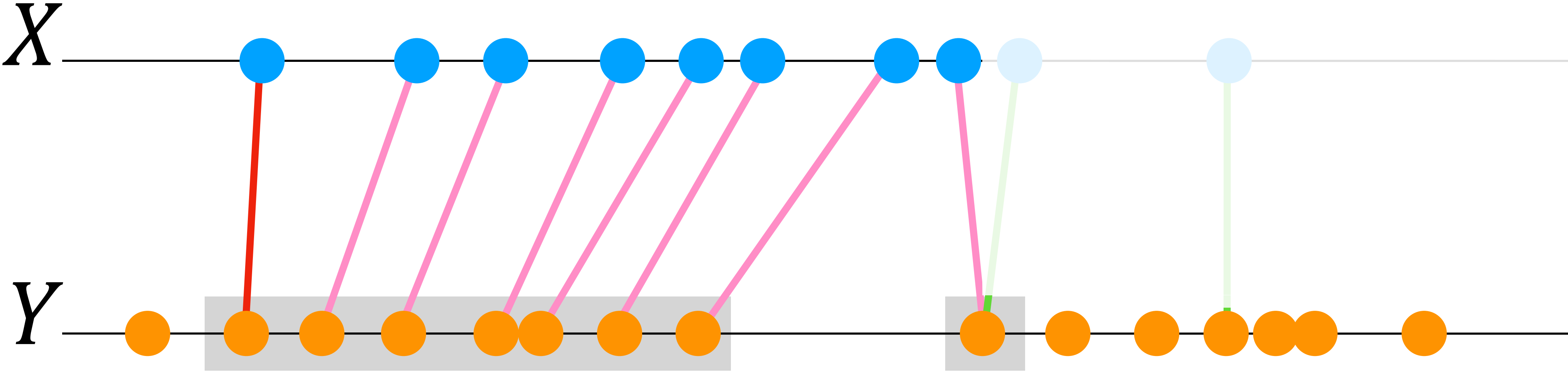
-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)

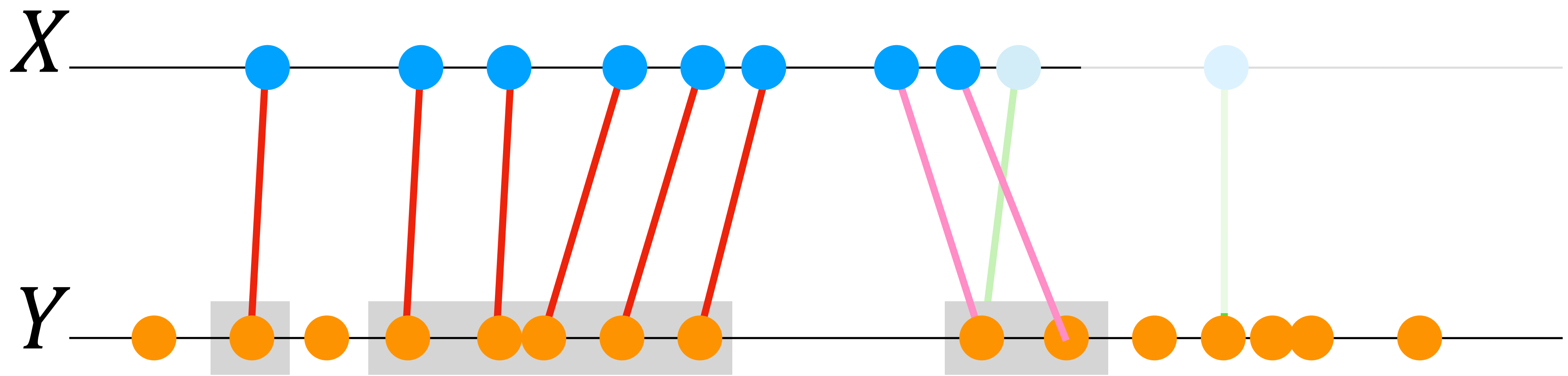


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

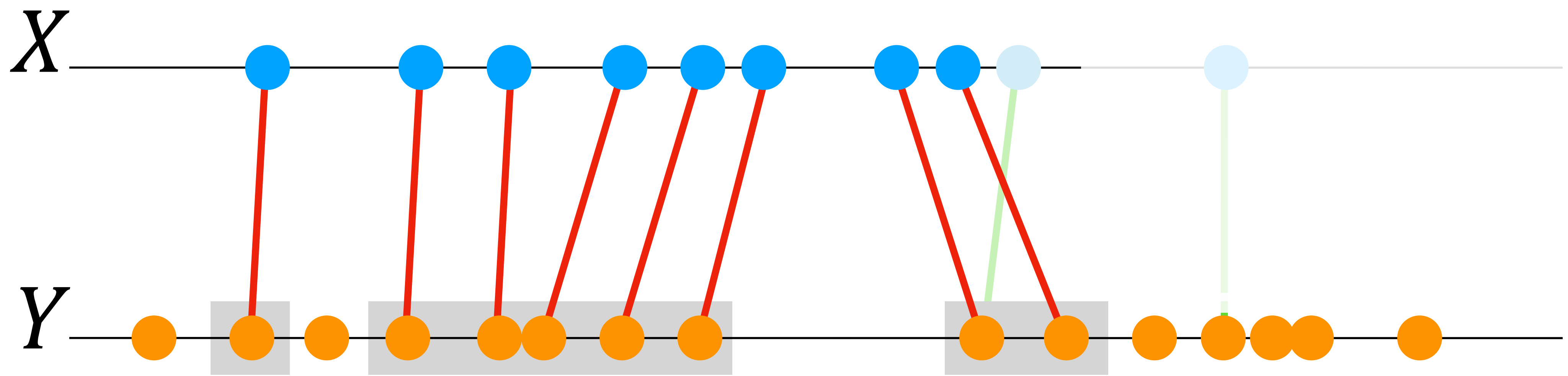
— Optimal Transport assignment



# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

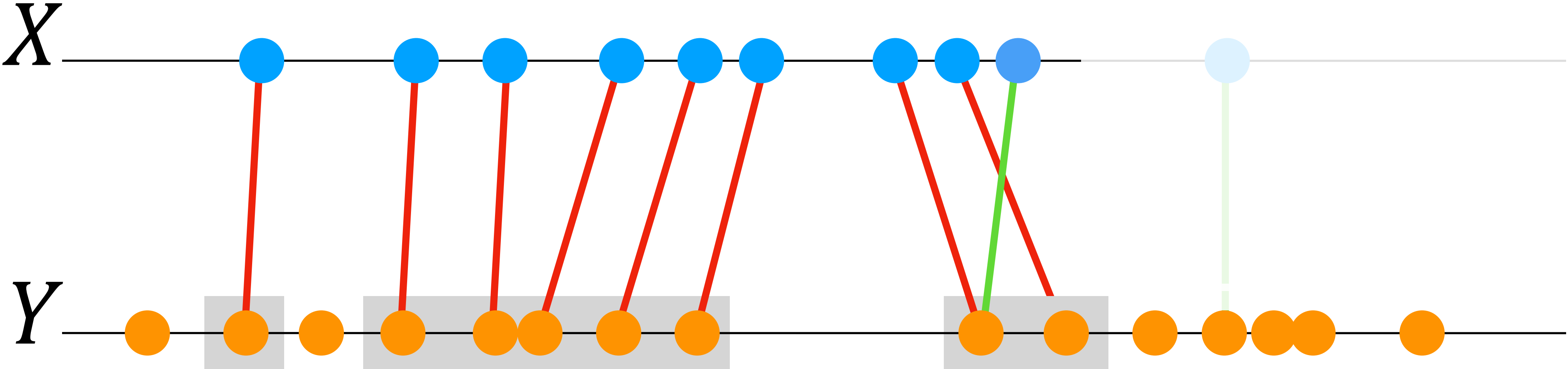
# Quadratic time complexity algorithm (linear space)



 **Euclidean Nearest Neighbor assignment**  
 **Optimal Transport assignment**

 **Intervals of bijective assignments**

# Quadratic time complexity algorithm (linear space)

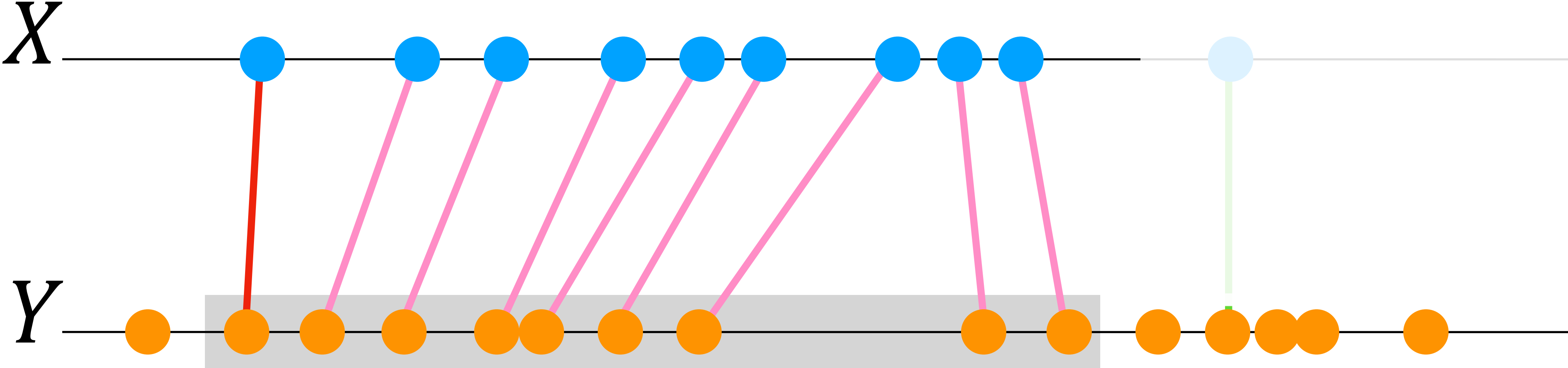


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

— Optimal Transport assignment

# Quadratic time complexity algorithm (linear space)



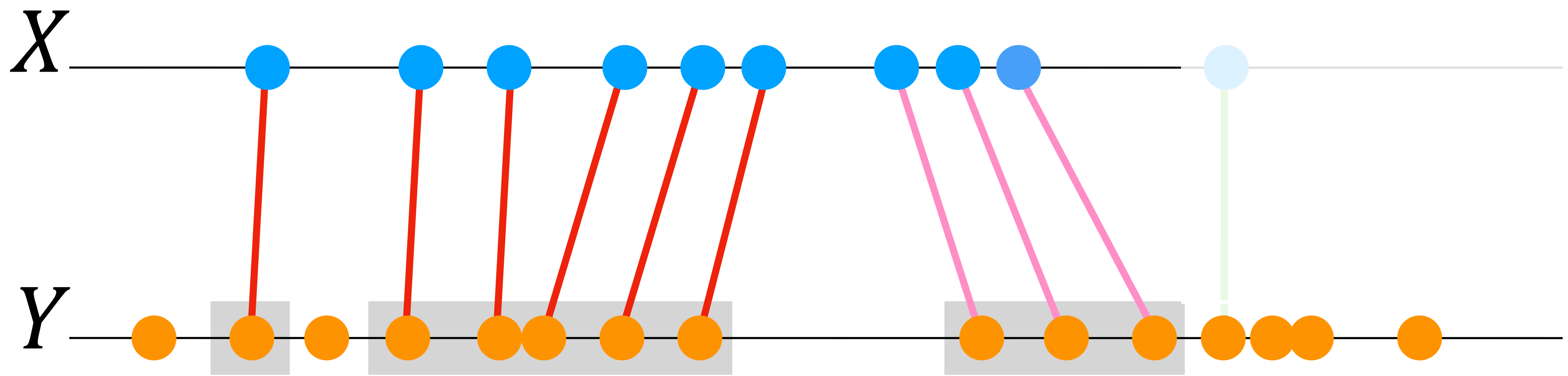
— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

— Optimal Transport assignment

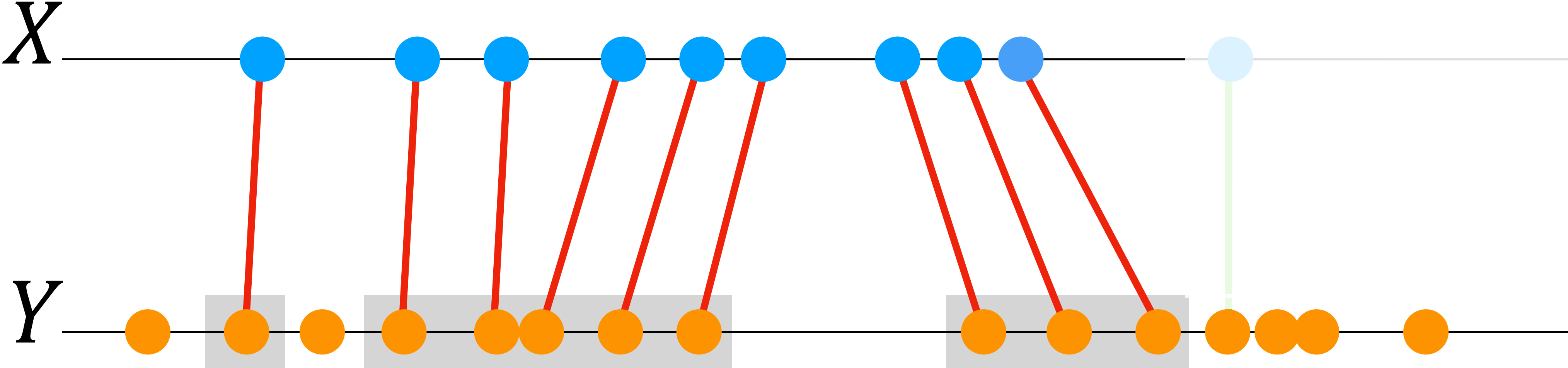


# Quadratic time complexity algorithm (linear space)



-  Euclidean Nearest Neighbor assignment
-  Optimal Transport assignment
-  Intervals of bijective assignments

# Quadratic time complexity algorithm (linear space)

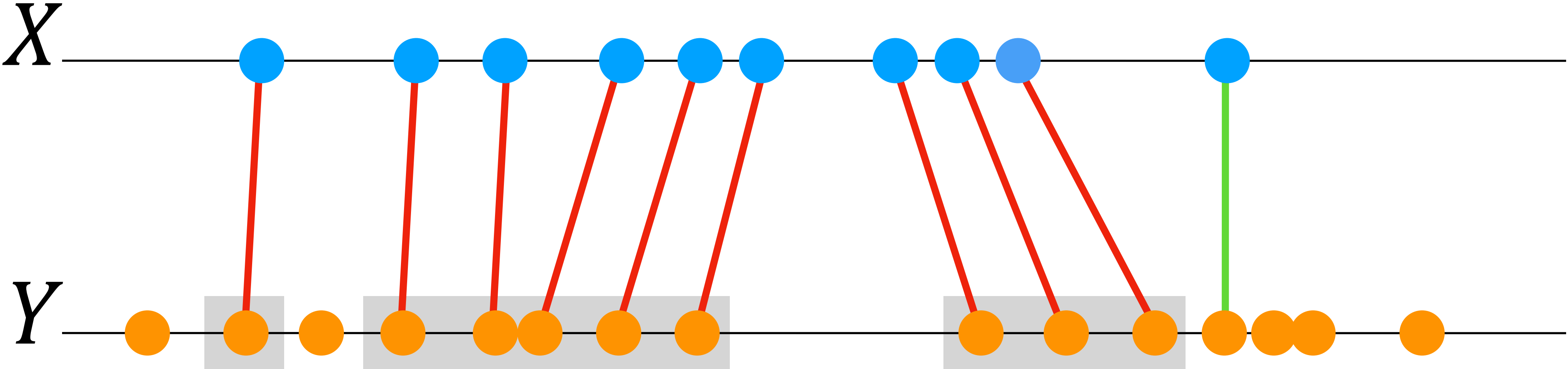


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

— Optimal Transport assignment

# Quadratic time complexity algorithm (linear space)

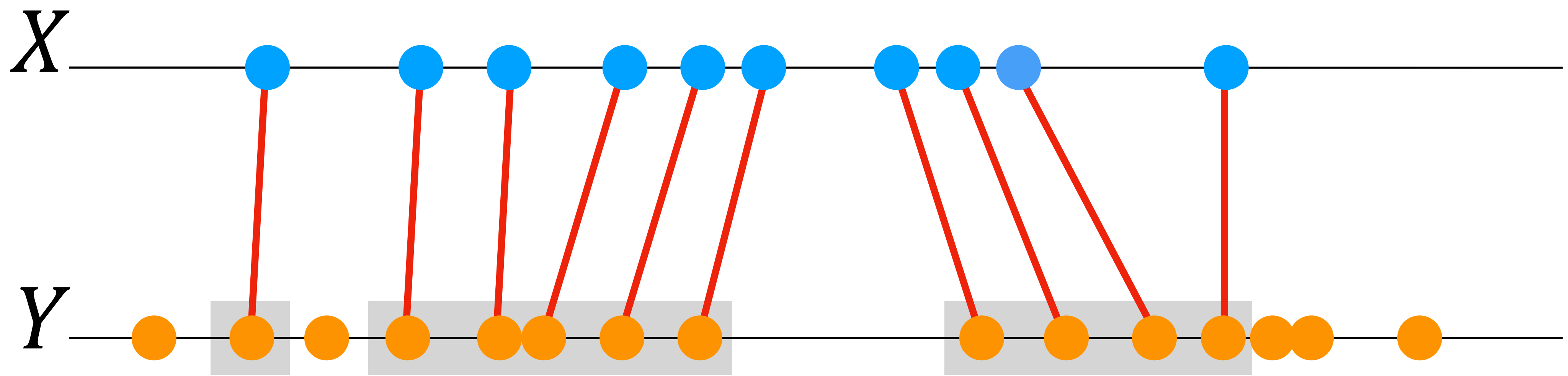


— Euclidean Nearest Neighbor assignment

■ Intervals of bijective assignments

— Optimal Transport assignment

# Quadratic time complexity algorithm (linear space)



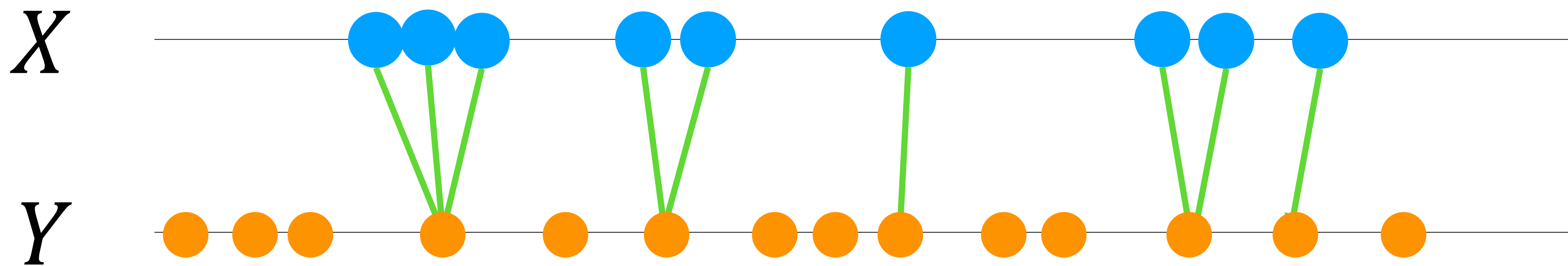
 Euclidean Nearest Neighbor assignment

 Intervals of bijective assignments

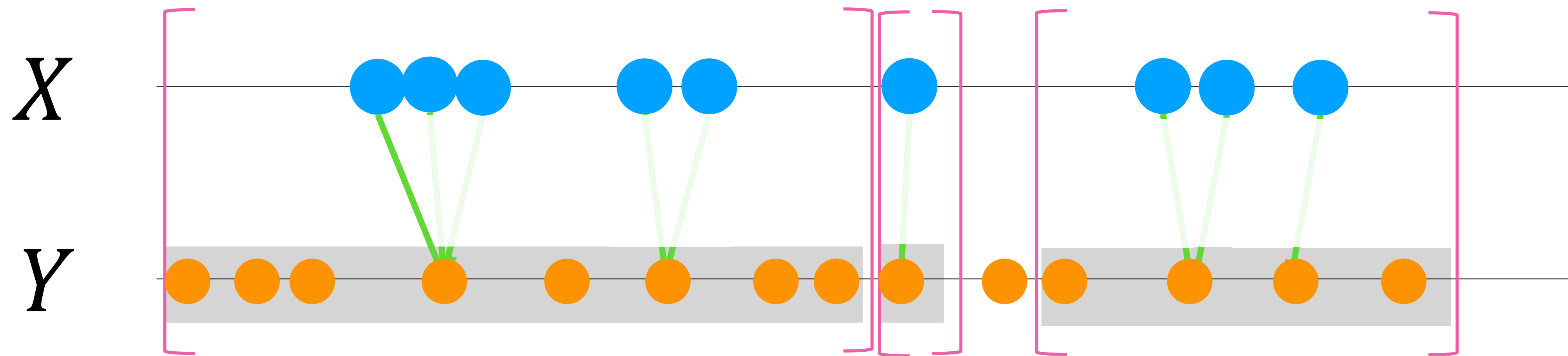
 Optimal Transport assignment

# 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)

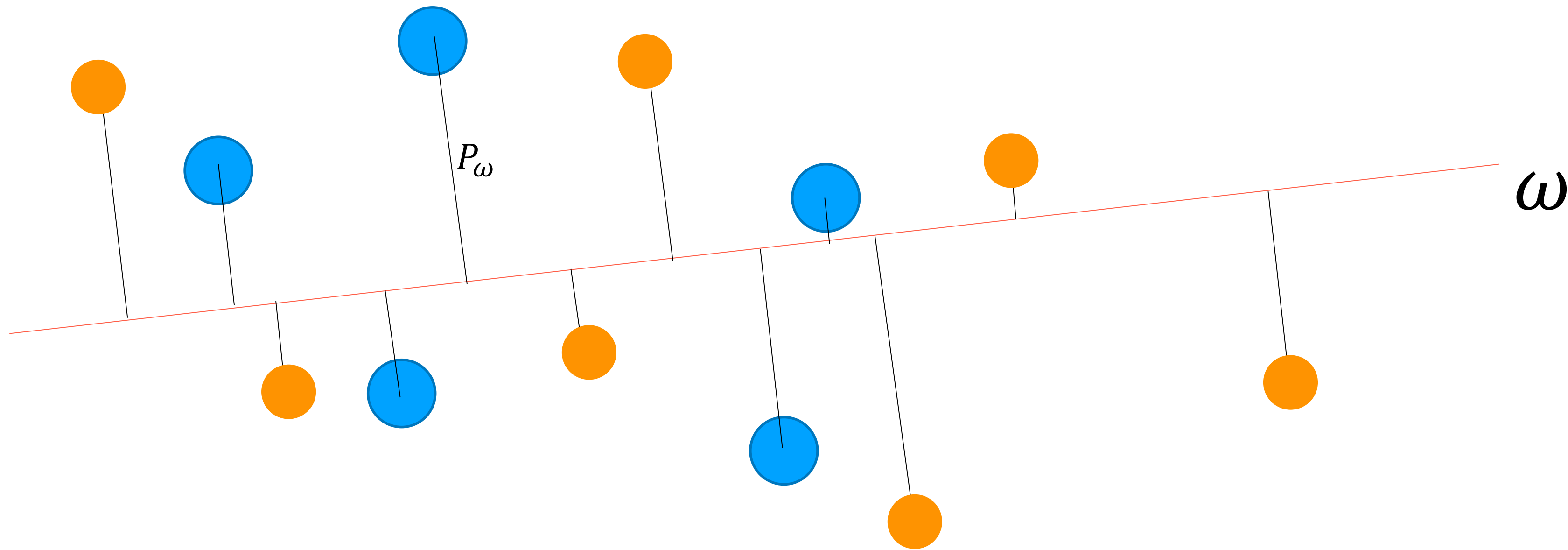


# Sliced Partial Optimal Transport (SPOT)

# Extension to d dimensions

- Sliced optimal transport

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

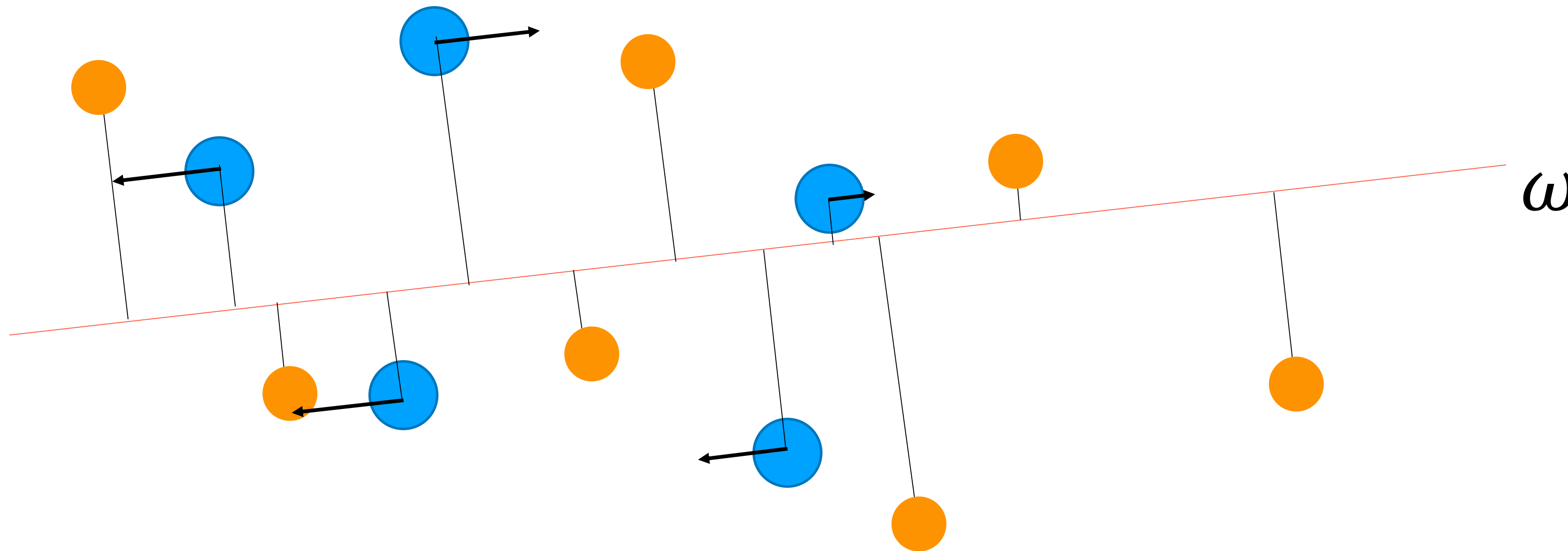


# Gradient flow

- Sliced optimal transport

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

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

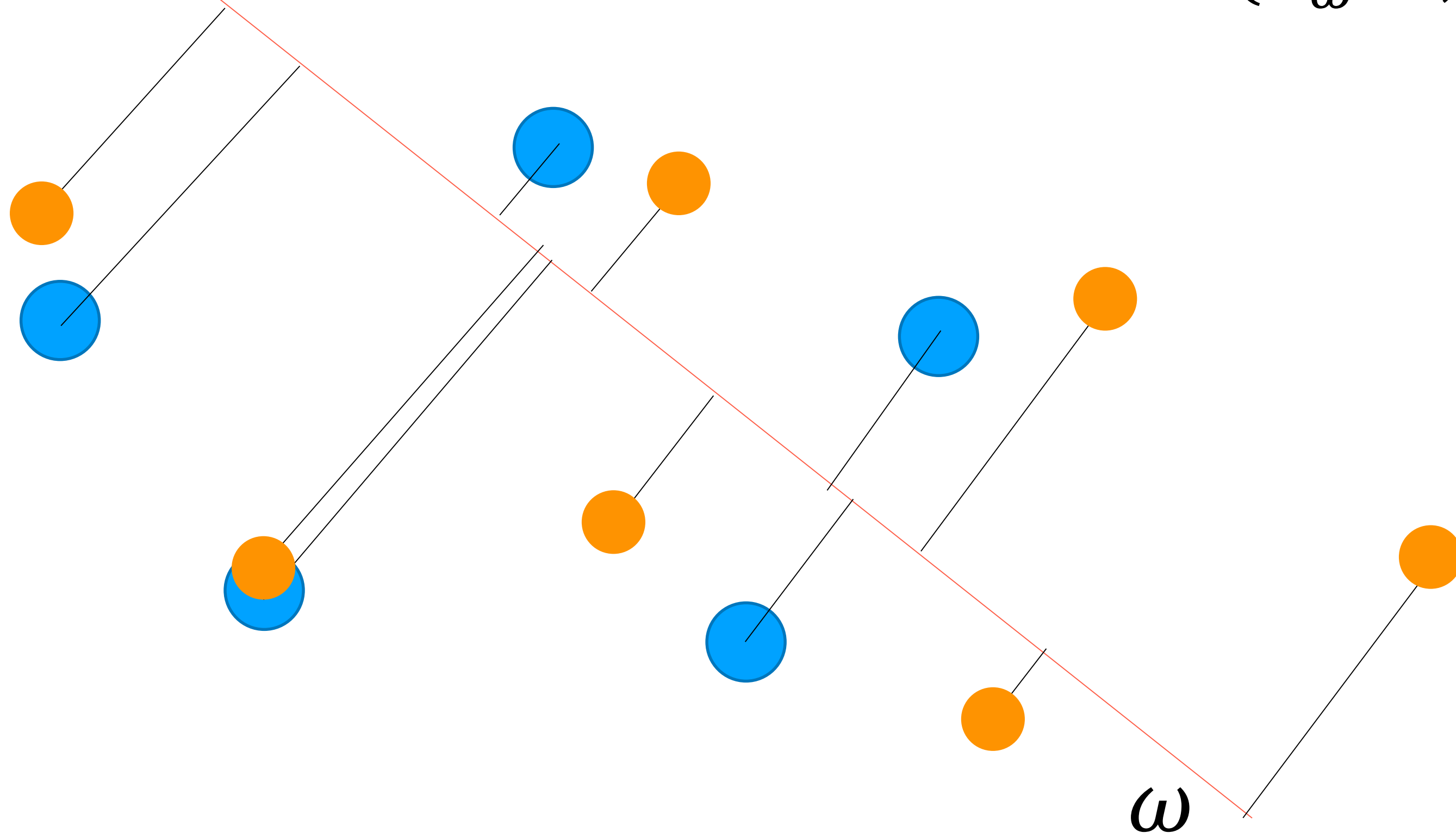


# Gradient flow

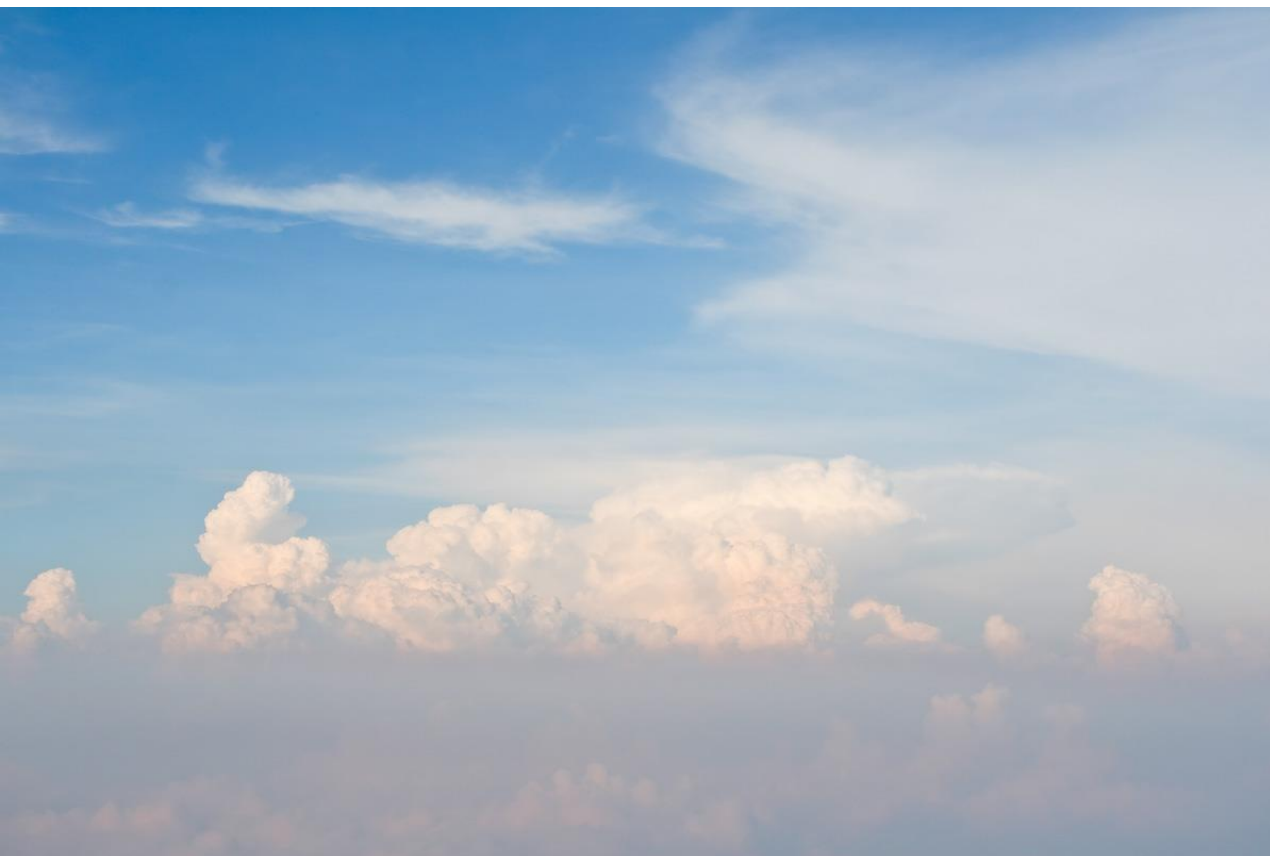
- Sliced optimal transport

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

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



# Color transfer application



**Full Transfer**

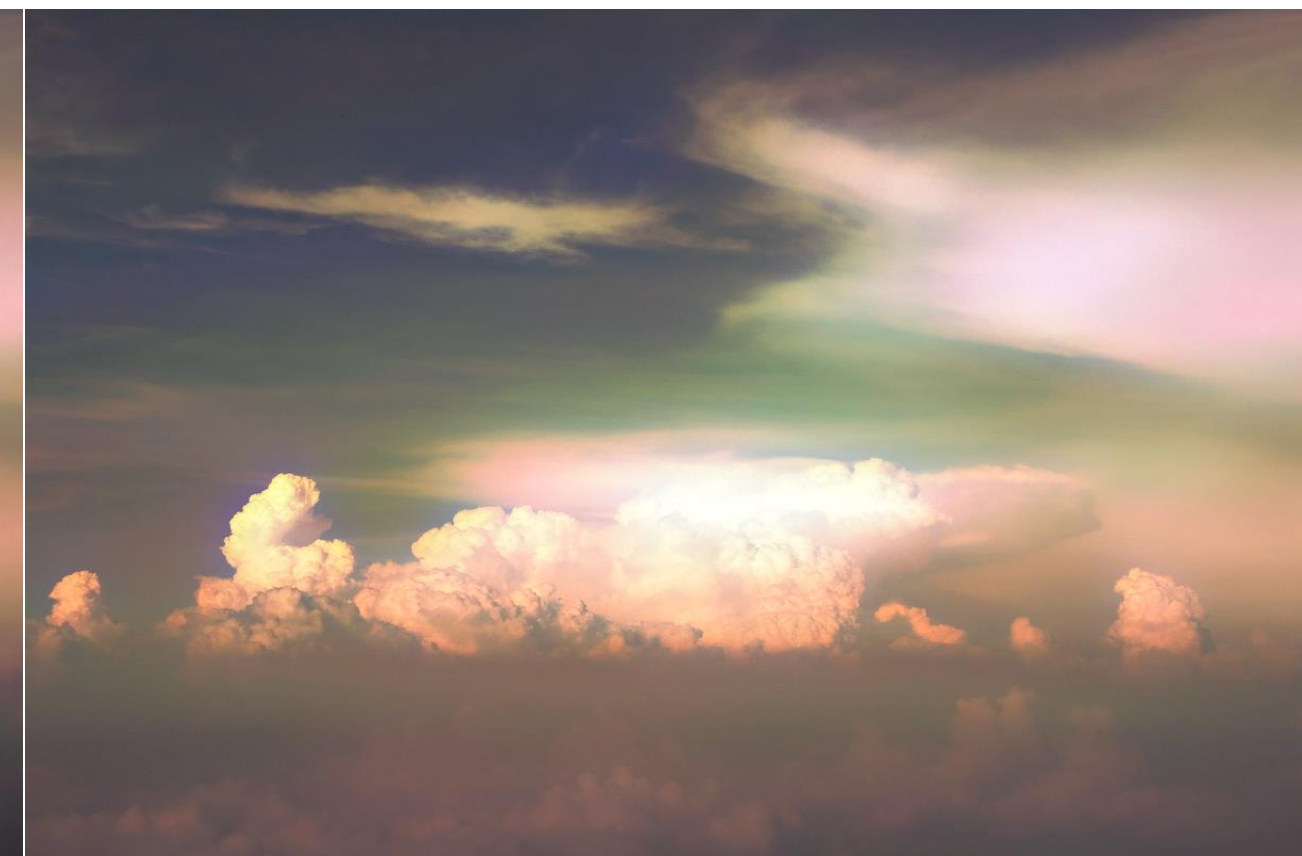
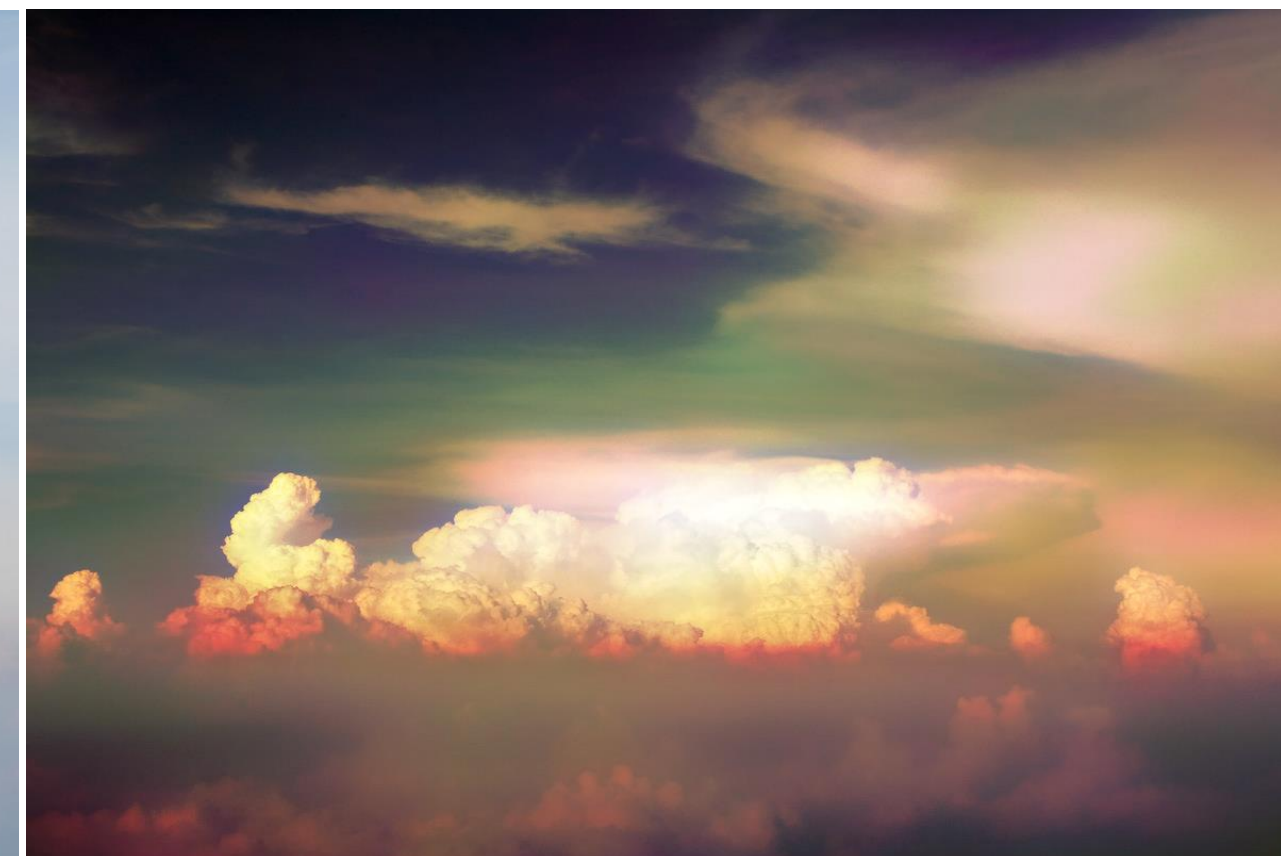
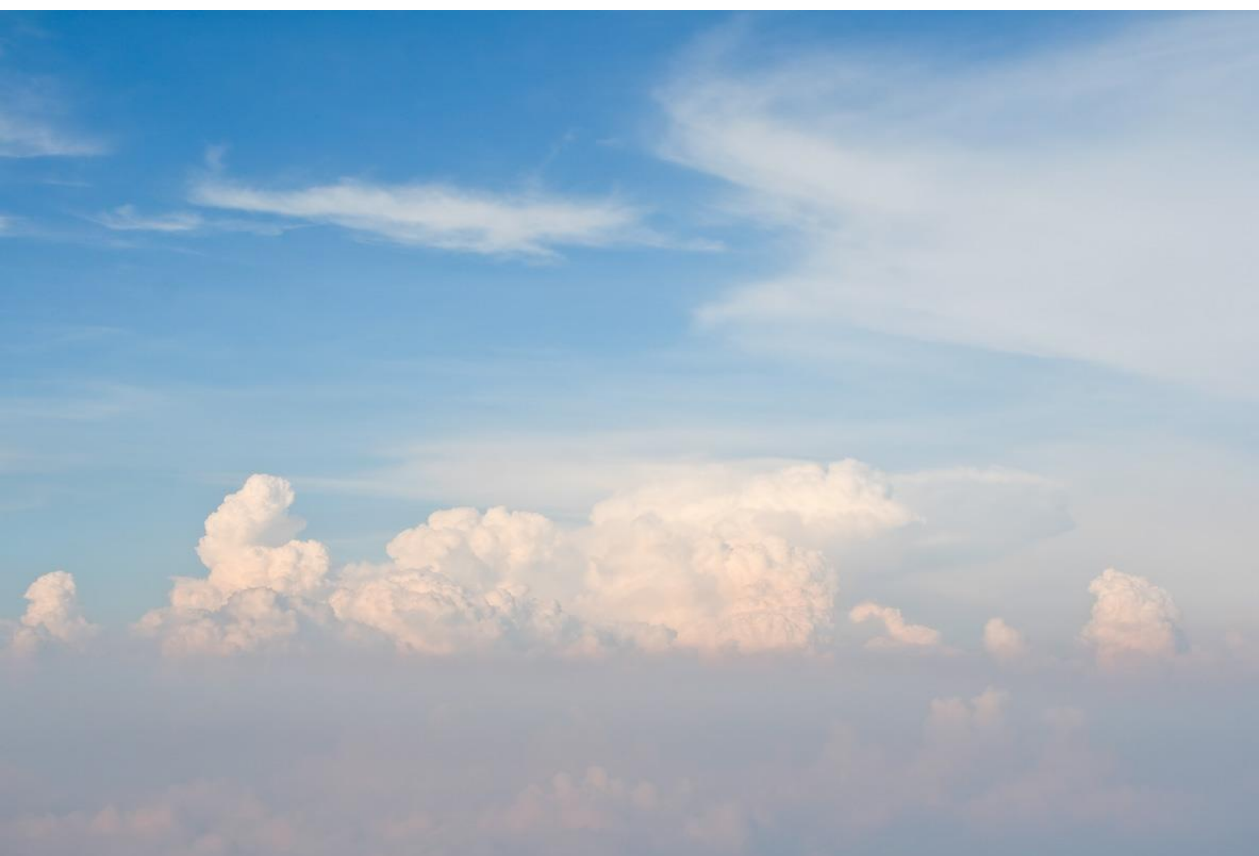


**Target 20% larger**



**Target 40% larger**

# Color transfer application



**Full Transfer**

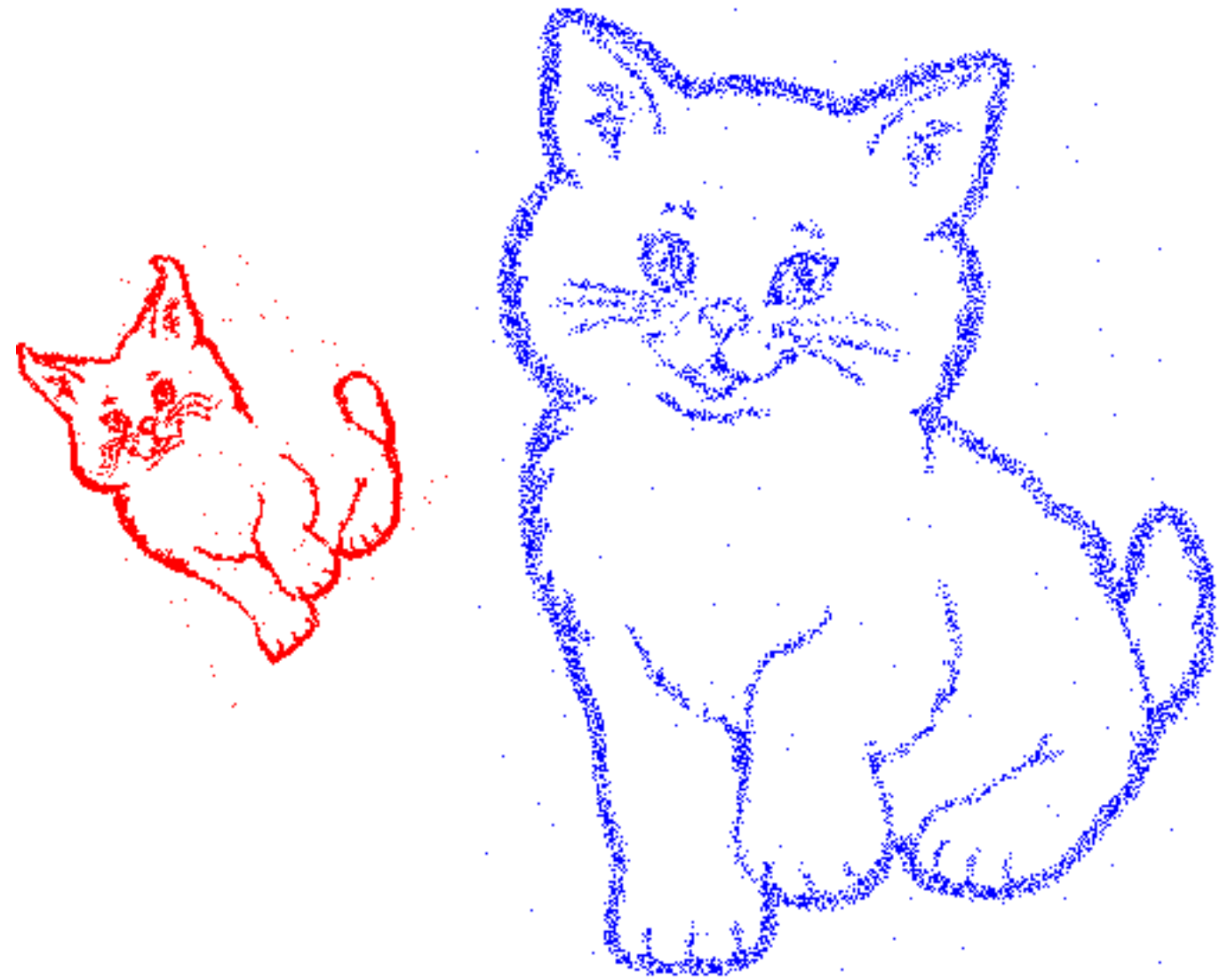
**Target 20% larger**

**Target 40% larger**

# Fast Iterative Sliced Transport (FIST)

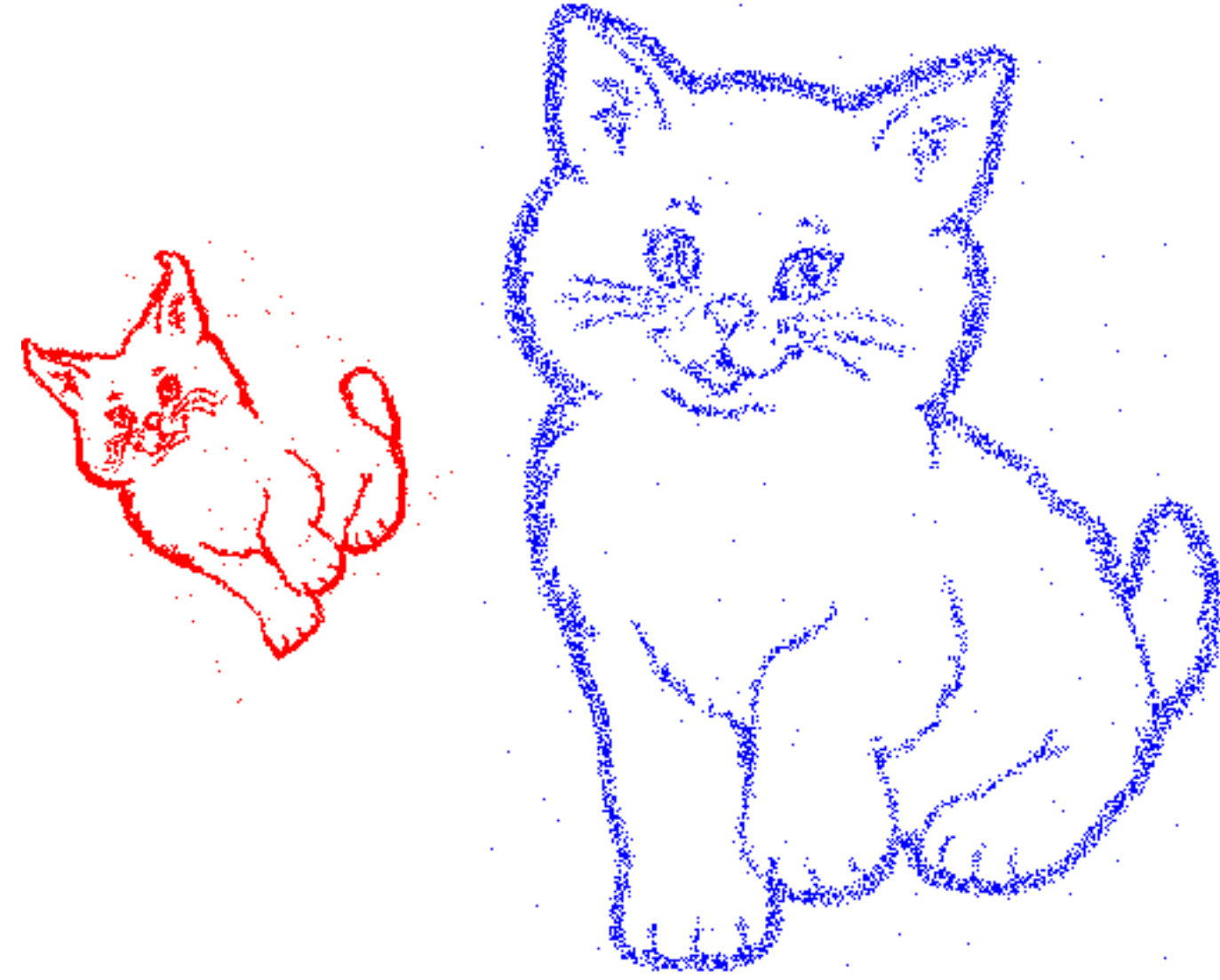
**Source: 8k samples**  
**Target: 10k samples**

0



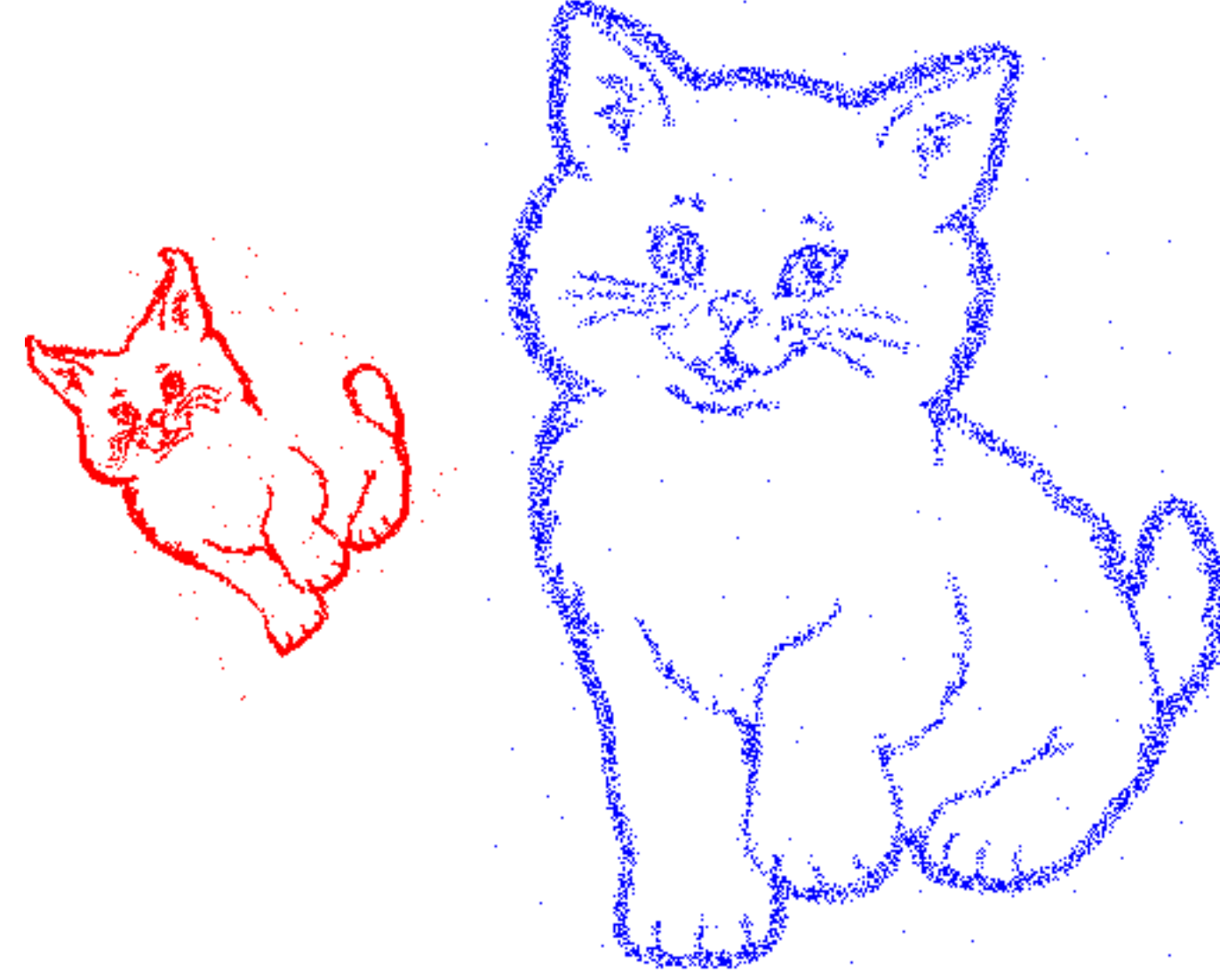
**ICP**  
**(0.005 s / iteration)**

0



**Iterative Transport**  
**with network simplex**  
**(40 s / iteration)**

0

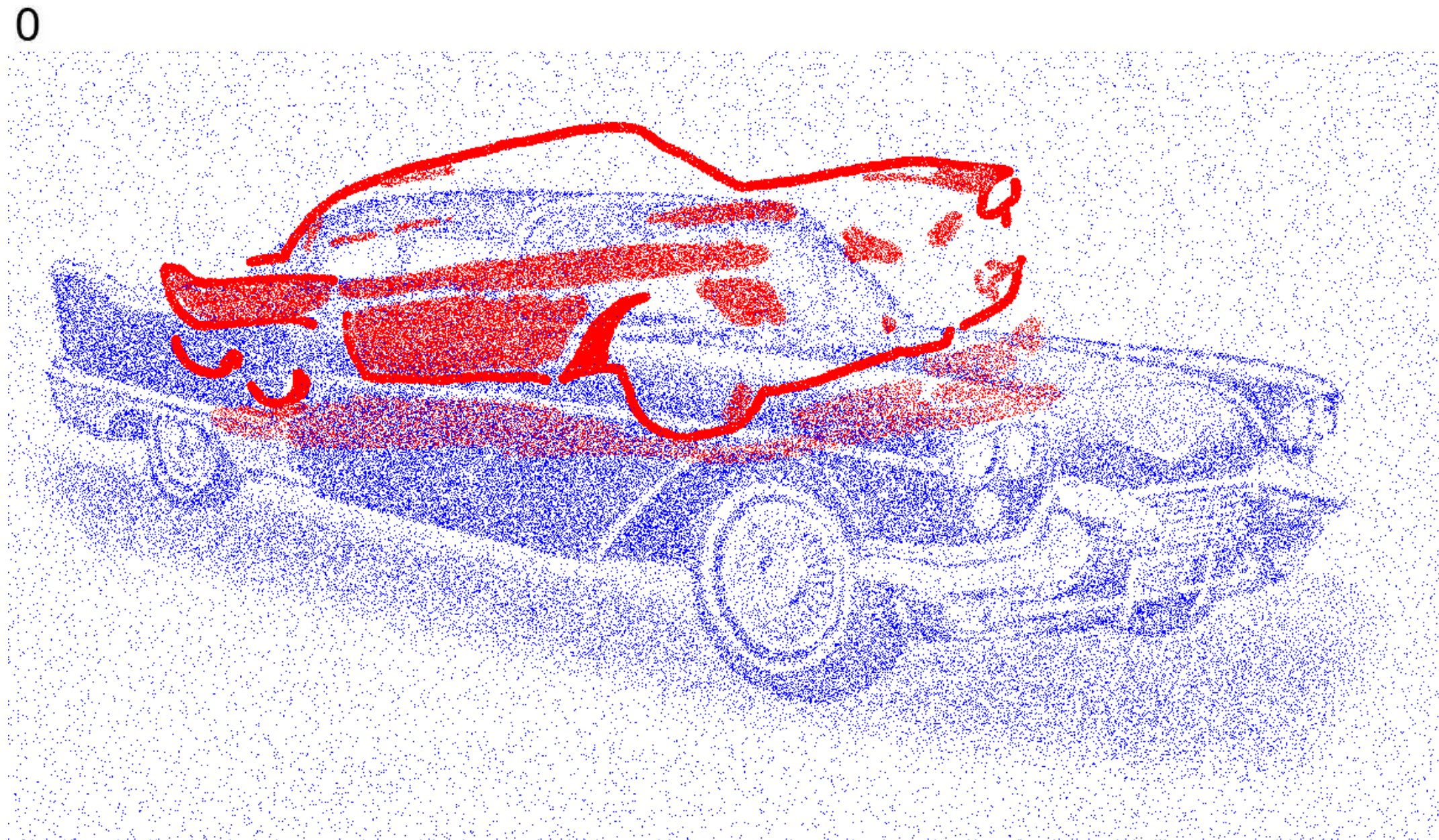


**Our FIST algorithm**  
**(0.04 s / iteration)**

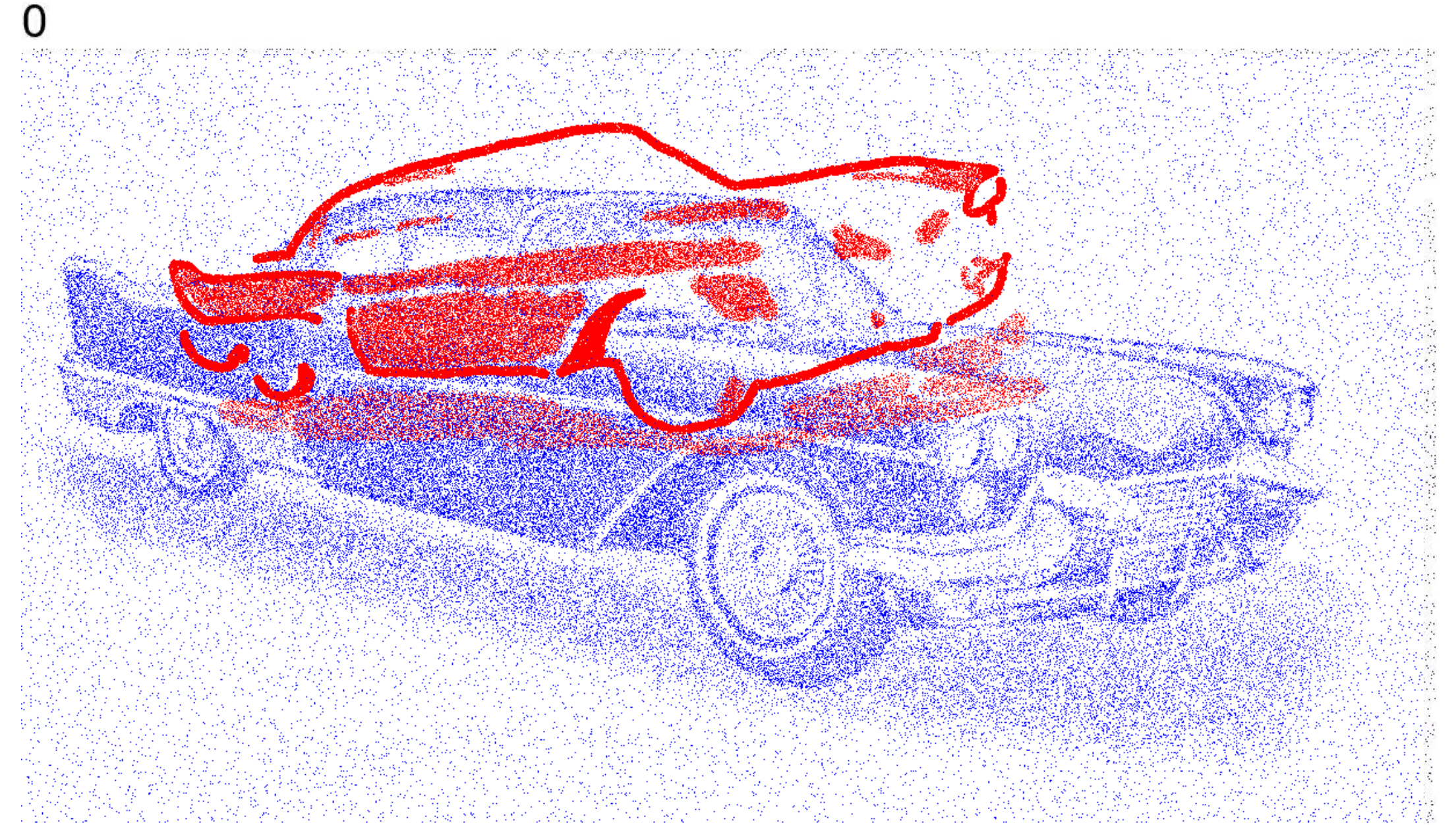


**Source: 90k samples**  
**Target: 100k samples**

*(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)*

0



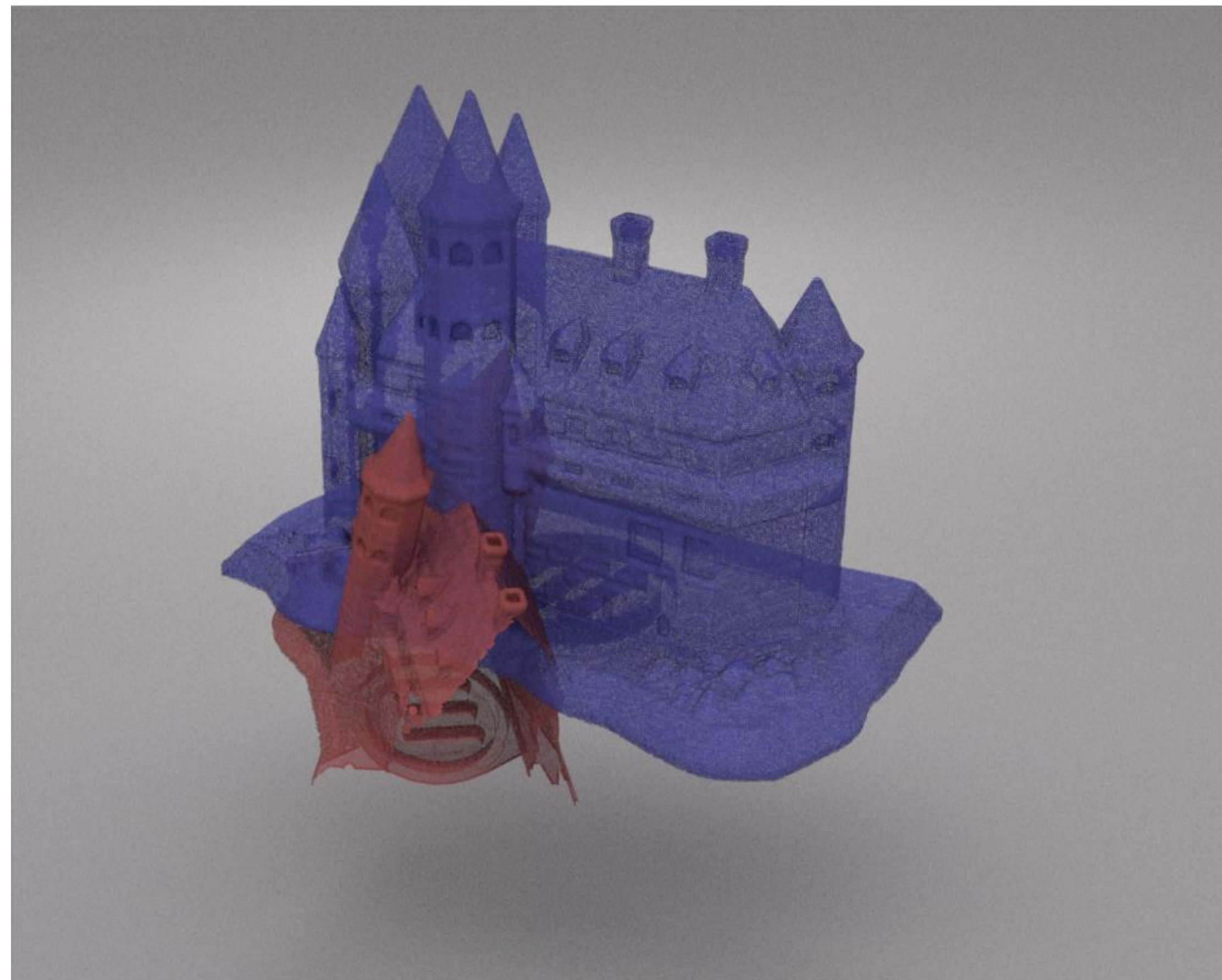
**ICP**  
**(0.05 s / iteration)**

0



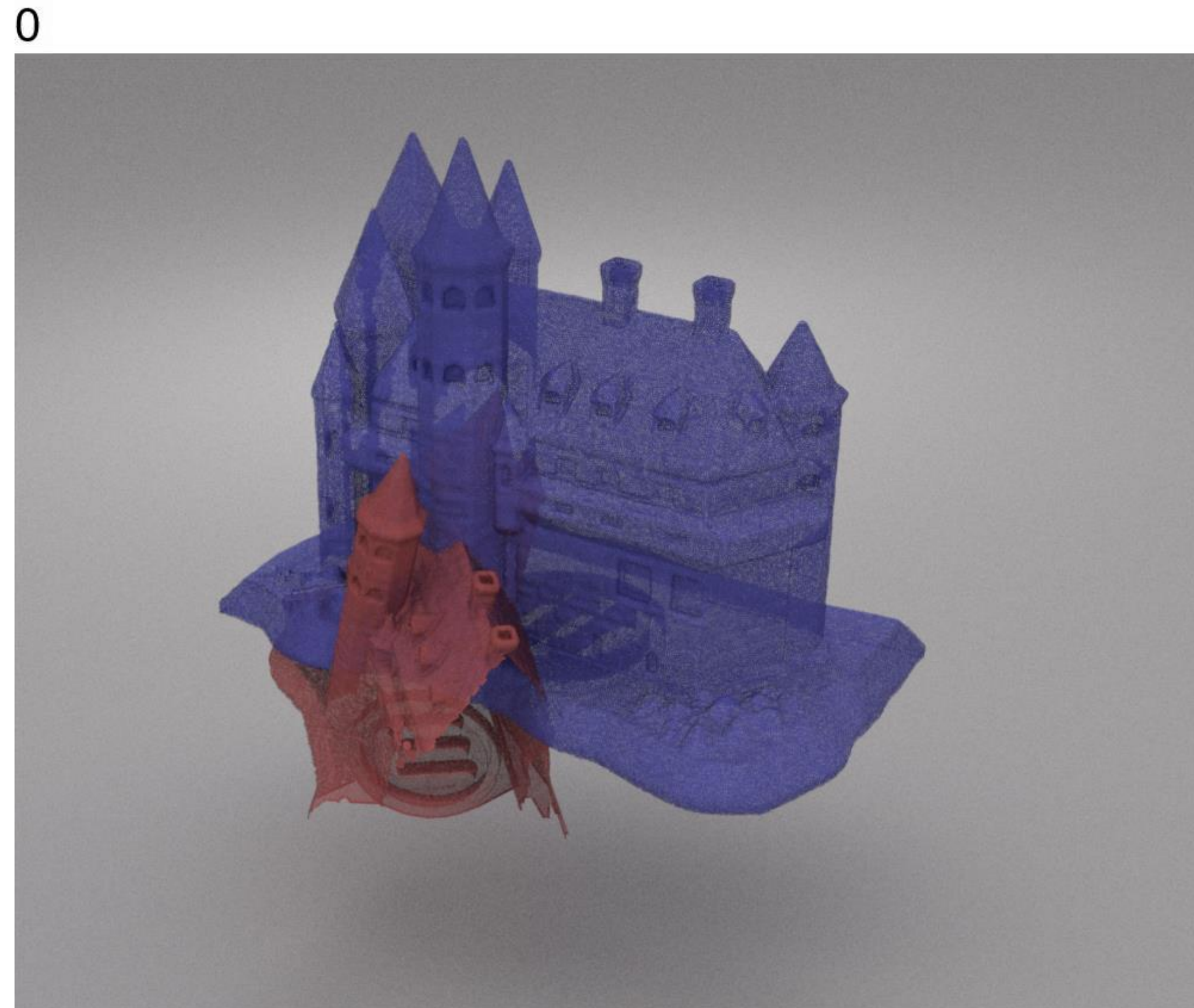
**Our FIST algorithm**  
**(0.69 s / iteration)**

**Source: 150k samples**  
**Target: 200k samples**



**ICP**  
**(0.09 s / iteration)**

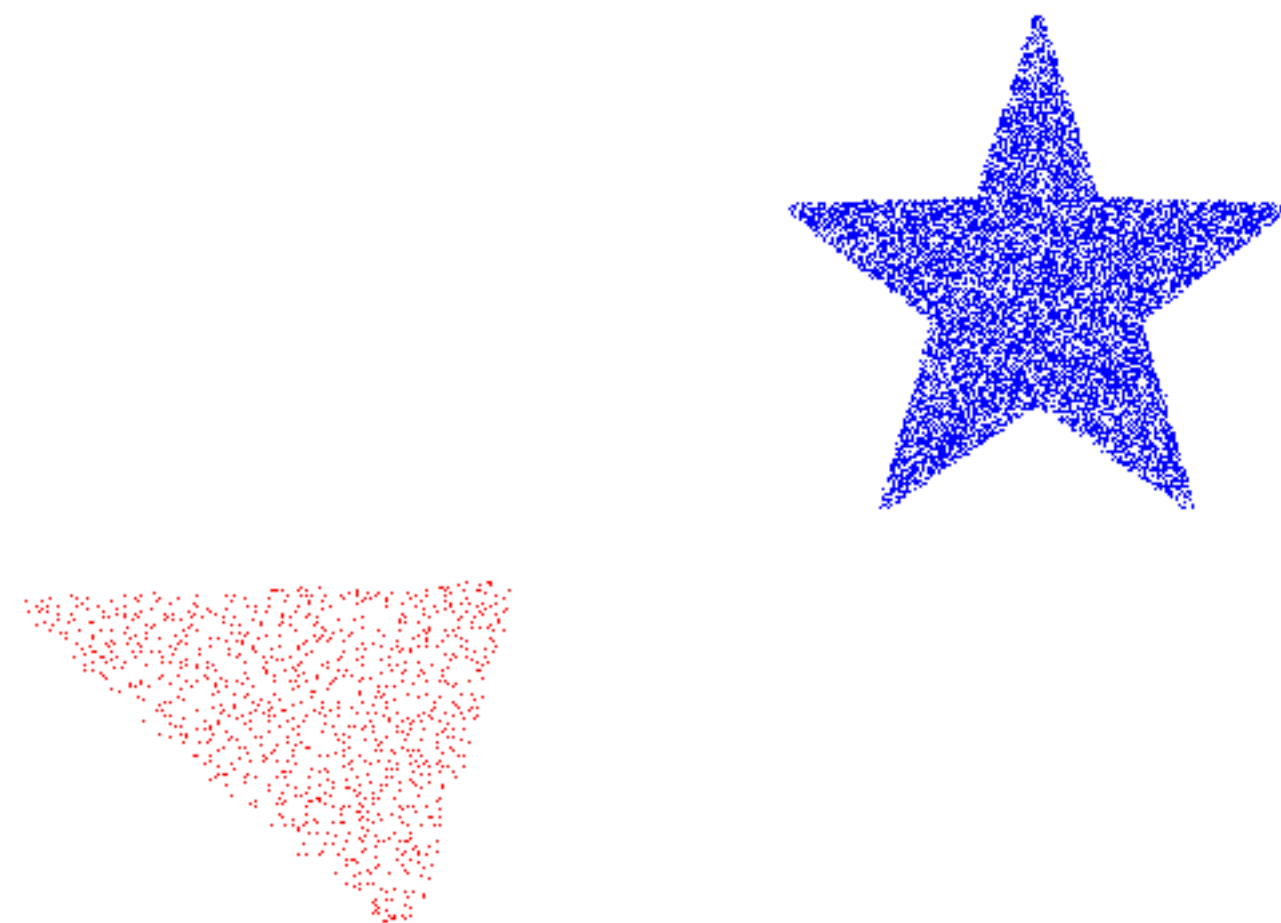
*(input too large for iterative  
transport with network simplex)*



**Our FIST algorithm**  
**(2.18 s / iteration)**

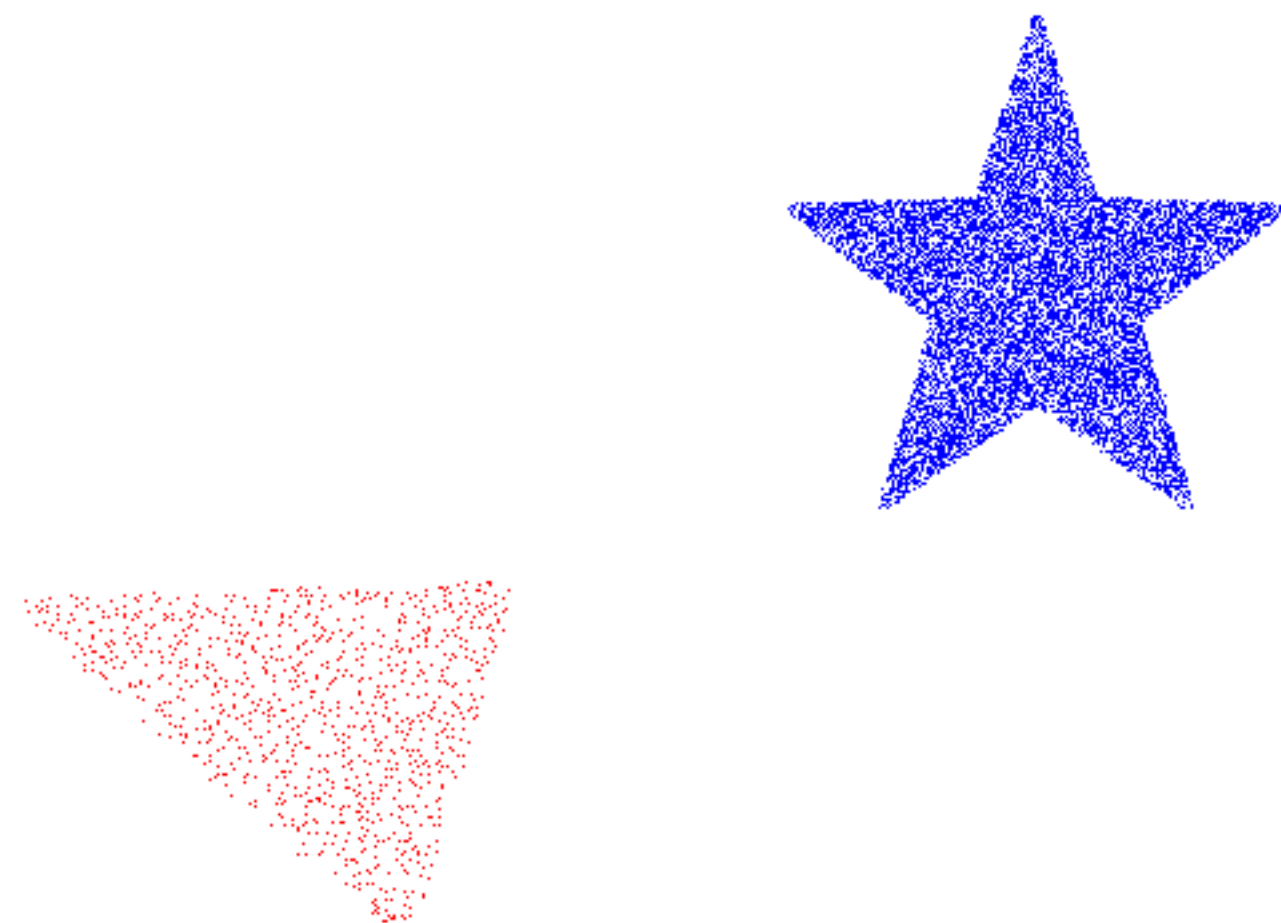
# Failure case: the transport is optimal only on projections

0



**Iterative Transport  
with Network Simplex**

0



**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/>