

Generalized OT Geodesics: Short Mathematical Summary

Problem Setup

Let $X_0, X_1 \in \mathbb{R}^{d \times n}$ be point clouds with n particles in dimension d . We optimize a discrete path

$$X = (X_t)_{t=0}^{T-1}, \quad X_t \in \mathbb{R}^{d \times n},$$

where $X_t(:, i)$ denotes particle i at time t .

Energy

We use the objective

$$\mathcal{E}(X) = \frac{1}{n} W_2^2(X_{T-1}^{\text{path}}, X_1) + \frac{\gamma}{n(T-1)} \sum_{t=1}^{T-1} \sum_{i=1}^n \phi(X_t, X_t(:, i)) \|X_t(:, i) - X_{t-1}(:, i)\|_2^2.$$

Here $\gamma > 0$ controls regularity and trajectory smoothness. The initial state is imposed as a hard constraint:

$$X_0^{\text{path}} = X_0.$$

Hence only $\{X_t\}_{t=1}^{T-1}$ are optimized.

Discrete Wasserstein Term

For clouds $Y, Z \in \mathbb{R}^{d \times n}$ with uniform weights, the squared 2-Wasserstein cost is

$$W_2^2(Y, Z) = \min_{\Pi \in \mathcal{U}(a, b)} \sum_{i,j} \Pi_{ij} \|y_i - z_j\|_2^2,$$

with $a = b = \frac{1}{n} \mathbf{1}$ and $\mathcal{U}(a, b)$ the transport polytope. In the implementation, this is computed with POT.

Choices of ϕ

- Classical OT-like kinetic term: $\phi \equiv 1$.
- Transformer-OT inspired variant:

$$\phi(X_t, \cdot) = \sum_{i,j} \exp(-\|X_t(:, i) - X_t(:, j)\|_2^2),$$

which is independent of the query position.

Optimization

Any feasible initialization can be used for $t \geq 1$ while keeping $X_0^{\text{path}} = X_0$. In the notebook example we use the trivial feasible path

$$X_t^{(0)} = X_0 \quad \text{for all } t = 0, \dots, T - 1.$$

Then we minimize \mathcal{E} with L-BFGS in PyTorch. The constrained structure is enforced by parameterizing and optimizing only the tail slices X_1, \dots, X_{T-1} , while concatenating X_0 as a fixed first slice at every iteration.

For reference, an alternative unconstrained warm-start is linear interpolation:

$$X_t^{(0)} = (1 - \alpha_t)X_0 + \alpha_t X_1, \quad \alpha_t = \frac{t}{T - 1}.$$

At each closure call, POT provides an OT plan for endpoint terms, and PyTorch evaluates the plan-weighted cost to backpropagate through point locations.