# COARSE-TO-FINE: HIERARCHICAL INCREMENTAL ACCELERATION FOR SOLVING EXACT SEMI-DISCRETE OPTIMAL TRANSPORT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Monte Carlo (MC)-based geometric variational methods address the intractability of analytical Power-Voronoi cell volume computation for semi-discrete optimal transport. However, their global cold-start synchronous iteration incurs drastically rising partitioning cost as target points scale up, severely slowing total solving efficiency. To mitigate this limitation, this paper proposes a hierarchical fast solving framework. First, we preprocess target points and partition them layer by layer in a nearest-to-farthest order. Second, we combine the head and tail layers to build global skeleton that locks the overall Power diagram topology in advance. In incremental updates, each point in every newly added layer is initialized with the Brenier height of its nearest solved neighbor. Using converged layer outputs for warm-start refinement, we sequentially solve optimal Brenier heights for all targets layer by layer. Experiments on standard and complex real-world distributions show that our approach achieves an average 53.88% acceleration versus the original MC solver, with stable permille-scale error between solved Brenier heights and ground truth. Supplementary tests on generative modeling, adversarial attacks and out-of-distribution overconfidence further confirm that our method causes only negligible percentage-to-permille scale performance deviations in downstream tasks.

## 1 INTRODUCTION

As a pivotal branch of Optimal Transport (OT) theory, Semi-Discrete OT (SDOT) boasts distinctive modeling advantages. It is formulated with a continuous source probability measure on one side and a discrete target measure composed of finitely points on the other, which naturally aligns with data distribution patterns in numerous real-world scenarios. Therefore, it has attracted sustained attention and wide practical deployment, covering computer vision (Li et al., 2025b; Tang et al., 2026), fluid simulation (Lavier, 2025; Lavier & Pelloni, 2026), materials science (Buze et al., 2024; Bourne et al., 2025b), topology optimization (Lévy, 2022; Dapogny et al., 2024), and other research domains.

Conventional numerical solvers for SDOT suffer severely from the curse of dimensionality when tackling high-dimensional problems, resulting in a drastic drop in computational efficiency and rendering them inadequate for distribution modeling in real-world tasks. Against this backdrop, Monte Carlo (MC)-based geometric variational solvers (Gu et al., 2013; An et al., 2020a) approximate integrals of continuous measures via random sampling, and constructs a convex variational optimization framework based on Power Diagram tessellation. Endowed with inherent high-dimensional adaptability, it has been deeply integrated into deep learning research pipelines, spanning multiple cutting-edge directions: images (Li et al., 2023; 2025a) and 3D point clouds (Zhang et al., 2022) generation, adversarial attack (Li et al., 2025b), and Out-of-Distribution (OOD) overconfidence evaluation (Tang et al., 2026), thus exhibiting prominent practical value for engineering applications.

Despite its high-dimensional adaptability and broad application prospects, the MC-based geometric variational solver has non-negligible performance bottlenecks that slow down its computational efficiency: (1) when dealing with large-scale target discrete points, the algorithm must complete cost comparison and cell assignment between all sampling points and all discrete target points within each iteration; (2) the full scale cold-start initialization strategy requires a large number of iterations for convergence. Recent works also explore Neural Network (NN)-based SDOT solvers (Li et al.,



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2024a;b). While they achieve fast inference relative to numerical solvers, such methods suffer from poor generalization and lack universal applicability: trained NNs only fit specific data distributions and show limited flexibility for diverse distributions, with no quantitative control over solving errors.

To overcome the limitations of the original MC-based geometric variational solver, we propose hierarchical incremental acceleration strategy that significantly improves computational efficiency without sacrificing precision. Distinct from NNs fitting, our purely geometric solver exhibits outstanding adaptability across diverse data distributions. Therefore, our contributions are summarized as:

• Motivated by the insight that neighboring target points tend to yield similar Brenier heights, we design a nearest-neighbor hierarchical preprocessing strategy for the target point set. We first randomly sample a subset of target points as the initial layer. Iteratively, the samples closest to the current processed set are selected from the remaining points as subsequent layers, which are then merged into the processed set until hierarchical partitioning terminates.

• We build global skeleton by merging the first and last layers, and then use the original MC-based geometric variational solver to lock the coarse-grained topological structure of the Power diagram. During incremental updates, target points of newly added layers are initialized with Brenier heights from their solved nearest-neighbor points. Relying on converged outputs of previous layers for warm-start refinement, we solve the optimal Brenier heights for all target points in a layer-wise manner.

• Extensive comparative experiments are conducted over standard and complex real-world distributions. The results show that the proposed hierarchical acceleration strategy strictly preserves the accuracy of the resulting SDOT mapping, introducing only per-mille-level errors against the original MC baseline and achieving an average speed-up of 53.88%. Further experiments on downstream tasks including generative modeling, adversarial attacks and OOD overconfidence verify that our method incurs negligible percentage-to-permille level performance deviation.

## 2 Preliminaries and Related Works

**SDOT Mapping.** Let $\mu$ denote an absolutely continuous Borel measure with compact support defined on the source domain $\Omega$, and $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$ represent a finite collection of target points associated with a discrete probability measure $\boldsymbol{\nu} = (\nu_i)_{i\in\mathcal{I}}$, where each weight satisfies $\nu_i > 0$ and the weights sum to unity: $sum(\boldsymbol{\nu}) = \sum_{i\in\mathcal{I}} \nu_i = 1$. The quadratic SDOT problem seeks a measurable transport mapping $T : \Omega \to \mathcal{Y}$ that pushes the source measure $\mu$ forward to the target measure $\boldsymbol{\nu}$ (denoted $T_{\#}\mu = \boldsymbol{\nu}$), while minimizing the squared Wasserstein distance between the two measures:

$$T_{ot} = \arg\inf_{T_{\#}\mu=\boldsymbol{\nu}} \int_{\Omega} \|\boldsymbol{x} - T(\boldsymbol{x})\|_2^2 d\mu(\boldsymbol{x}). \tag{1}$$

By Brenier's polar factorization theorem (Brenier, 1991), the SDOT mapping $T_{ot}$ admits a unique characterization via the Brenier height vector $\boldsymbol{h} = (h_i)_{i\in\mathcal{I}}$. Specifically, $T_{ot}$ corresponds to the gradient of a piece-wise convex potential function $u(\boldsymbol{x})$, defined as the upper envelope of a family of affine functions: $u(\boldsymbol{x}) = \max_{i\in\mathcal{I}} (\langle \boldsymbol{x}, \boldsymbol{y}_i\rangle + h_i)$, such that $T_{ot} = \nabla u$. For each target point $\boldsymbol{y}_i$, the region of the source domain mapped to $\boldsymbol{y}_i$ under $T_{ot}$ forms a Power-Voronoi cell, defined as:

$$W_i(\boldsymbol{h}) = \{\boldsymbol{x} \in \Omega \mid \langle \boldsymbol{x}, \boldsymbol{y}_i\rangle + h_i \geq \langle \boldsymbol{x}, \boldsymbol{y}_j\rangle + h_j, \forall j \neq i, j \in \mathcal{I}\}, i \in \mathcal{I}, \tag{2}$$

where $\langle\cdot,\cdot\rangle$ denotes the standard Euclidean inner product. The collection of cells $\{W_i(\boldsymbol{h})\}_{i\in\mathcal{I}}$ constitutes a disjoint partition of the entire source domain $\Omega$.

The Brenier height vector $\boldsymbol{h}$ must satisfy the fundamental volume-matching condition: the measure of each Power-Voronoi cell under the source measure $\mu$ must exactly equal the corresponding target weight, i.e., $\mu(W_i(\boldsymbol{h})) = \nu_i$ for all $i \in \mathcal{I}$. Under this optimal configuration, the SDOT mapping takes a simple piecewise-constant form: $T_{ot}(\boldsymbol{x}) = \boldsymbol{y}_i$ if and only if $\boldsymbol{x} \in W_i(\boldsymbol{h})$.

**Monte Carlo-Based Geometric Variational Solver (Full-Scale Cold-Start).** A standard approach to solving the SDOT problem leverages its equivalent variational formulation (Gu et al., 2013), which recasts the volume-matching condition as a convex optimization problem defined over the Brenier height vector $\boldsymbol{h}$. This framework enables the use of efficient gradient-based optimization algorithms. The corresponding strictly convex energy functional is given by:

$$E(\boldsymbol{h}) = \int_{\Omega} \max_{i\in\mathcal{I}} (\langle \boldsymbol{x}, \boldsymbol{y}_i\rangle + h_i)\, d\mu(\boldsymbol{x}) - \sum_{i\in\mathcal{I}} h_i\nu_i. \tag{3}$$

This functional is both strictly convex and continuously differentiable. Its partial derivative with respect to each height variable $h_i$ has an intuitive geometric interpretation, and is given by:

$$\nabla_{h_i} E(\boldsymbol{h}) = \mu\left(W_i(\boldsymbol{h})\right) - \nu_i, \forall i \in \mathcal{I}. \tag{4}$$

Due to strict convexity, the functional $E(\boldsymbol{h})$ admits a unique global minimizer $\boldsymbol{h}^*$, which corresponds exactly to the optimal Brenier height satisfying the volume-matching condition.

For large-scale or high-dimensional SDOT problems, exact evaluation of the integral in equation 3 is computationally intractable. To address this, An et al. (2020a) propose a MC sampling strategy to approximate the integral and its gradient stochastically. Let $\{\boldsymbol{x}_j\}_{j=1}^{N_{mc}}$ denote a set of independent and identically distributed ($i.i.d.$) samples drawn from the source measure $\mu$. The resulting MC estimator for the gradient of the energy functional $E(\boldsymbol{h})$ in equation 4 takes the form:

$$\nabla_{h_i} \hat{E}(\boldsymbol{h}) = \frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} \mathbb{I}\left\{\boldsymbol{x}_j \in W_i(\boldsymbol{h})\right\} - \nu_i, i \in \mathcal{I}, \tag{5}$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function, which equals 1 when its argument is true and 0 otherwise. Using this stochastic gradient estimator, the Brenier height vector is updated iteratively via $\boldsymbol{h}^{(t+1)} = \boldsymbol{h}^{(t)} - lr^{(t)} \cdot \nabla \hat{E}(\boldsymbol{h}^{(t)})$, where $lr^{(t)}$ denotes the learning rate at the $t$-th iteration.

Algorithm 3 outlines the complete pipeline of the MC-based geometric variational solver. While effective for SDOT, this method adopts a full-scale cold-start strategy with zero-initialized Brenier potentials, incurring prohibitive computational overhead in large-scale and high-dimensional settings. Capitalizing on the prior that neighboring target points share similar Brenier heights, this work proposes a layer-wise warm-start framework that substantially accelerates SDOT computation.

**Theoretical Foundations of SDOT.** The variational formulation of SDOT was established by Gu et al. (2013), who recast the problem as convex energy minimization over power weights via a Minkowski-type principle. Subsequent regularity analysis by (De Gournay et al., 2018) showed that the Brenier potential has Lipschitz-continuous derivatives with respect to discrete point positions and weights under mild assumptions. Bansil & Kitagawa (2022) further derived quantitative stability bounds and invertibility conditions linking Laguerre-cell measures to dual variables. On the statistical side, central limit theorems and asymptotic inference for Wasserstein distances and SDOT maps were developed in (Del Barrio et al., 2024; Sadhu et al., 2024), while Paindaveine & Passeggeri (2024) characterized solver robustness under source-distribution perturbations. Most recently, Genans et al. (2026) established minimax convergence rates for stochastic SDOT, proving that empirical maps converge to the population map at $\mathcal{O}(n^{-1/2})$ under i.i.d. source sampling. *Additional references related to numerical solver of SDOT and its application are placed in Appendix B.*

## 3 Hierarchical Fast Solver for Semi-Discrete Optimal Transport

Our critical insight, summarized in Lemma 1, is that spatially adjacent target points possess similar Brenier heights. Accordingly, full-scale cold-start optimization over all samples is unnecessary. Instead, we only perform cold-start optimization on a subset of key points, while the remaining points inherit the Brenier heights from their nearest neighbors for warm-start initialization (Fig. 1). Sufficient with minor fine-tuning, this strategy substantially reduces the overall computational cost.

**Lemma 1** (Proof in Appendix C.2)**.** *Suppose that $\boldsymbol{h}$ is the Brenier height vector associated with the SDOT mapping from the continuous source domain $\Omega$ to the discrete target point set $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$. Then we have $|h_i - h_j| \leq \mathrm{norm}(\Omega) \cdot \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$ for $\forall i, j \in \mathcal{I}$, where $\mathrm{norm}(\Omega) = \sup_{\boldsymbol{x}\in\Omega} \|\boldsymbol{x}\|_2$.*

### 3.1 Hierarchical Preprocessing Strategy

**Initial Core Layer Sampling via Uniform Random Sampling.** The initial core layer $\mathcal{Y}_1$ establishes the global skeleton topology of the Power diagram. We sample $\mathcal{Y}_1$ from the full target set $\mathcal{Y}$ via uniform random sampling without replacement, controlled by a hierarchical ratio $r \in (0, 1]$ such that $|\mathcal{Y}_1| = \lfloor |\mathcal{Y}| \cdot r \rfloor$. This yields an unbiased initial skeleton with full spatial coverage across the support domain. The remaining points form the residual candidate set $\widehat{\mathcal{Y}}_1 = \mathcal{Y} \setminus \mathcal{Y}_1$, which is further partitioned into successive incremental layers.

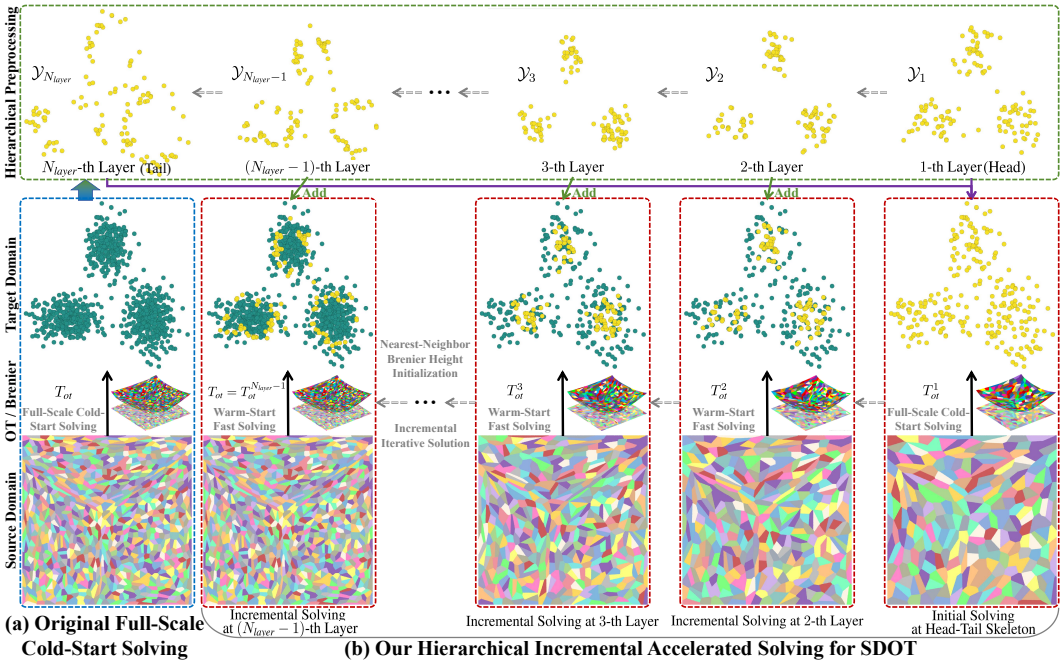



Fig 1: To address the high per-iteration cell-assignment cost and slow convergence caused by global random initialization in the **(a) Original Full-Scale Cold-Start SDOT Solver** for large-scale target sets, we propose **(b) Hierarchical Incremental Accelerated SDOT Solver**. It maintains solution accuracy comparable to direct solving over the full target set while substantially reducing the overall computational overhead. The proposed method consists of three core stages: **(1) Hierarchical Preprocessing**. A subset of target points is randomly sampled to form the initial layer $\mathcal{Y}_1$. Points closest to the current processed set are iteratively selected from the remaining samples and incorporated, yielding the layered target sets $\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_{N_{layer}}$. **(2) Head-Tail Skeleton Initialization**. The initial core layer $\mathcal{Y}_1$ and the farthest edge layer $\mathcal{Y}_{N_{layer}}$ are combined as the initial solving set. This small scale set rapidly produces the initial Brenier potential, which locks the global support range and topological structure of the Power Diagram. **(3) Layer-Wise Incremental Filling**. Incremental layers are incorporated sequentially from near to far. New points are initialized using the Brenier heights of their nearest solved neighbors. Warm-started from solutions of previous layers, only a few iterations are required to refine the current-layer solution until all target points are processed.

**Nearest-Neighbor Priority Incremental Layer Partitioning.** Following the sampling of the initial core layer $\mathcal{Y}_1$ and the derivation of the remaining candidate set $\widehat{\mathcal{Y}}_1 = \mathcal{Y} \setminus \mathcal{Y}_1$, a nearest-Neighbor Priority partitioning strategy with fixed per-layer sample size is adopted to construct incremental subsets iteratively. The number of samples selected for each layer is uniformly set to the cardinality of the initial core layer $|\mathcal{Y}_1|$. The detailed procedure is described as follows:

(1) For each target point in $\widehat{\mathcal{Y}}_1$, compute its minimum Euclidean distance to the set $\mathcal{Y}_1$:

$$d(\boldsymbol{y}) = \min_{\boldsymbol{y}' \in \mathcal{Y}_1} \|\boldsymbol{y} - \boldsymbol{y}'\|_2, \boldsymbol{y} \in \widehat{\mathcal{Y}}_1. \tag{6}$$

Sort all distances in ascending order, select the $|\mathcal{Y}_1|$ samples with the smallest distances to form incremental layer $\mathcal{Y}_2$, and update the remaining candidate set to $\widehat{\mathcal{Y}}_2$ by removing the selected samples.

(2) Take the union of all selected layers $\bigcup_{e=1}^{k-1} \mathcal{Y}_e$ as the new reference set. Compute the minimum distance from each point in $\widehat{\mathcal{Y}}_{k-1}$ to this reference set, select the $|\mathcal{Y}_1|$ nearest samples to form $\mathcal{Y}_k$, and denote the remaining samples as $\widehat{\mathcal{Y}}_k$.

(3) Repeat the above procedure iteratively. In each layer, the union of all currently selected layers serves as the reference set for distance calculation, and $|\mathcal{Y}_1|$ nearest samples are selected to generate the next layer $\mathcal{Y}_k$. The loop terminates when the size of the remaining candidate set is smaller than $|\mathcal{Y}_1|$, and all remaining samples are merged into the final layer $\mathcal{Y}_{N_{layer}}$, where $N_{layer} = \lceil \frac{|\mathcal{Y}|}{|\mathcal{Y}_1|} \rceil$.

(4) For each layer $\mathcal{Y}_k, k \in [1, N_{layer}]$, we extract the corresponding index set $\mathcal{I}_k$ and mass assignment vector $\boldsymbol{\nu}_k$ from the global index set $\mathcal{I}$ and discrete target measure $\boldsymbol{\nu}$, which facilitates the subsequent incremental computation.

The resulting partition yields the sequence of disjoint layers $\mathcal{H} = \{(\mathcal{Y}_1, \boldsymbol{\nu}_1, \mathcal{I}_1), \cdots, (\mathcal{Y}_{N_{layer}}, \boldsymbol{\nu}_{N_{layer}}, \mathcal{I}_{N_{layer}})\}$ satisfying $\bigcup_{k=1}^{N_{layer}} \mathcal{Y}_k = \mathcal{Y}$ and $\mathcal{Y}_k \cap \mathcal{Y}_{k'} = \emptyset$ for $\forall 1 \le k < k' \le N_{layer}$. This strategy ensures that the target points added in each layer are spatially adjacent to the existing solved cell structure, which justifies the nearest-neighbor Brenier height initialization for new points. The overall hierarchical preprocessing pipeline is summarized in Algorithm 1.

**Algorithm 1** Hierarchical Preprocessing of Target Point Sets

**Input**: Target Dataset $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$ with Measure $\boldsymbol{\nu} = (\nu_i)_{i\in\mathcal{I}}$; Hierarchical Ratio $r \in (0, 1]$.

$m \leftarrow \lfloor |\mathcal{Y}| \cdot r \rfloor$; $N_{layer} \leftarrow \lceil \frac{|\mathcal{Y}|}{m} \rceil$.
**for** $k = 1 : N_{layer}$ **do**
  **if** $k = 1$ **then**
    $\mathcal{I}_k \overset{u.a.r.}{\sim} \mathcal{I}$, $|\mathcal{I}_k| = m$. ▷{Initial Core Layer Sampling via Uniform Random Sampling}
  **else**
    $d_i \leftarrow \min_{j \in \bigcup_{e=1}^{k-1} \mathcal{I}_e} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2,\ i \in \mathcal{I}$.
    $\mathcal{I}_k \leftarrow \arg\min_{\mathcal{I}' \subseteq \mathcal{I}, |\mathcal{I}'| = \min\{m, |\mathcal{I}|\}} \sum_{i \in \mathcal{I}'} d_i$. ▷{Nearest-Neighbor Priority Incremental Layer Partitioning}
  **end if**
  $\mathcal{Y}_k \leftarrow \{\boldsymbol{y}_i\}_{i\in\mathcal{I}_k}$; $\boldsymbol{\nu}_k \leftarrow (\nu_i)_{i\in\mathcal{I}_k}$; $\mathcal{I} \leftarrow \mathcal{I} \setminus \mathcal{I}_k$.
**end for**
$\mathcal{H} \leftarrow \{(\mathcal{Y}_1, \boldsymbol{\nu}_1, \mathcal{I}_1), \cdots, (\mathcal{Y}_{N_{layer}}, \boldsymbol{\nu}_{N_{layer}}, \mathcal{I}_{N_{layer}})\}$.

**Return**: Hierarchically Preprocessed $\mathcal{H}$.

### 3.2 Joint Initial Solution with Head-Tail Skeleton

This stage solves for the union of the head and tail layers to quickly obtain a globally consistent initial Brenier potential, providing high-quality warm-start for subsequent incremental steps.

**Initial Target Set Construction.** The initial target set is defined as $\mathcal{Y}_{init} = \mathcal{Y}_1 \cup \mathcal{Y}_{N_{layer}}$. $\mathcal{Y}_1$ covers core sparse control points, while $\mathcal{Y}_{N_{layer}}$ contains the farthest edge points. Their combination locks the global support of SDOT and the overall Power Diagram topology, preventing large-scale cell reconstruction and Brenier height oscillation during later incremental filling.

**Monte Carlo Solution for Initial Target Set.** The MC-Based Geometric variational Algorithm 3 is applied to solve the SDOT mapping from $\Omega$ to $\mathcal{Y}_{init}$, and the current Brenier height $\boldsymbol{h}_1^*$ is obtained.

### 3.3 Layer-Wise Incremental Update Mechanism

Starting from the initial skeleton solution, incremental layers are added sequentially. Nearest-neighbor Brenier height initialization and minimal iterative refinement enable faster incremental solving.

**Nearest-Neighbor Brenier Height Initialization for New Points.** For the $k$-th incremental layer $\mathcal{Y}_k, k \in [2, N_{layer} - 1]$, the Brenier height of each new point $\boldsymbol{y}_j^k \in \mathcal{Y}_k$ is initialized as follows:

(1) Find its spatial nearest neighbor in the current solved set $\mathcal{Y}_{current} = \mathcal{Y}_{init} \cup (\bigcup_2^{k-1} \mathcal{Y}_e)$:

$$\lambda = \arg\min_{i \in \mathcal{I}_{current}} \|\boldsymbol{y}_j^k - \boldsymbol{y}_i\|_2, j \in \mathcal{I}_k, \boldsymbol{y}_i \in \mathcal{Y}_{current}. \tag{7}$$

(2) Assign the optimal Brenier height of $\boldsymbol{y}_\lambda$ directly to $\boldsymbol{y}_j^k$: $h_j^{k,init} = h_\lambda$.

This initialization is theoretically justified. As proven in Lemmas 2-4 (Appendix C.1), the Brenier potential under the quadratic cost is convex and Lipschitz continuous on the support set. Combined with the strong correlation of optimal heights among spatially adjacent points (Lemma 1), nearest-neighbor assignment can produce initial values close to the global optimum, and the corresponding initialization error bound is rigorously characterized in Theorem 1. This strategy significantly reduces the overall computational cost caused by conventional full-scale cold-start optimization.

**Theorem 1** (Proof in Appendix C.4). *Let $\boldsymbol{h}^{k,*} = (h_i^{k,*})_{i\in\mathcal{I}_{current}\cup\mathcal{I}_k}$ be the Brenier height vector of SDOT from $\Omega$ with measure $\mu$ to the discrete target set $\mathcal{Y}_{current} \cup \mathcal{Y}_k$ with measure $(\boldsymbol{\nu}_{current}, \boldsymbol{\nu}_k)/(1 + sum(\boldsymbol{\nu}_k))$. The resulting initialization error satisfies relation*

$$|h_j^{k,*} - h_j^{k,init}| \le \mathrm{norm}(\Omega) \cdot \|\boldsymbol{y}_j^k - \boldsymbol{y}_\lambda\|_2 + \frac{\sqrt{2} \cdot \nu_j^k}{1 + \nu_j^k}, j \in \mathcal{I}_k. \tag{8}$$

**Incremental Iterative Solution.** After initialization, $\mathcal{Y}_k$ and its associated Brenier heights are appended to the current target set $\mathcal{Y}_{current}$ and optimal Brenier height vector $\boldsymbol{h}^{k-1,*}$, respectively. The learning rate is reset, and the number of MC samples at the current layer is moderately reduced

to accelerate convergence. We iteratively perform cell assignment and Brenier height updates until the volume deviation falls below the layer-specific convergence threshold, at which point the optimal weight $\boldsymbol{h}^{k,*}$ of the current layer is obtained. Notably, a stricter, smaller threshold $\tau^{fine}$ is only adopted for the final layer to guarantee overall solution accuracy, while looser threshold $\tau^{coarse}$ is used for all preceding layers to speed up hierarchical computation.

This incremental addition and refinement process repeats for all layers $\{\mathcal{Y}_k\}_{k=2}^{N_{layer}-1}$. The final output is the optimal Brenier height vector and corresponding Power Diagram for the full target set $\mathcal{Y}$, from which the SDOT mapping is derived. The complete solving procedure is outlined in Algorithm 2.

### 3.4 Computational Complexity Comparative Analysis

The computational cost incurred by the hierarchical preprocessing is negligible relative to the overall complexity of the proposed Algorithm 2. The per-iteration overhead is dominated by Power-Voronoi cell assignment, while cell sample statistics and Brenier height updates are negligible. Complexity analysis thus focuses exclusively on the core cell assignment step.

**Original Full-Scale Cold-Start Algorithm 3.** The original cold-start algorithm involves all $|\mathcal{Y}|$ target points simultaneously, with zero-initialized Brenier heights for cold-start iteration until convergence. (1) Per-Iteration Complexity. With KD-tree acceleration for optimal Power-Voronoi cell assignment of each source sample, the complexity of a single iteration is $\mathcal{O}(2^i \cdot N_{init} \log |\mathcal{Y}|)$, where $i \in \mathbb{N}$ and $2^i \cdot N_{init}$ denotes the MC sample size in the current iteration. (2) Overall Complexity. For $2^i \cdot N_{init}$-size MC samples, $t^i_{full}$ iterations are required to reach full convergence, so the total complexity of the original Algorithm 3 is $C_{base} = \mathcal{O}(\sum_{i=0}^{N} t^i_{full} \cdot 2^i \cdot N_{init} \log |\mathcal{Y}|)$.

For large-scale $|\mathcal{Y}|$, $\log |\mathcal{Y}|$ increases slowly, but random cold-start causes a drastic rise in $T_{full}$. Meanwhile, the spatial index must be constructed for all $|\mathcal{Y}|$ target points in each iteration, keeping the baseline overhead of a single iteration persistently high.

**Proposed Hierarchical Incremental Algorithm 2.** The proposed Algorithm 2 consists of two core stages: initial skeleton solving and layer-wise incremental warm-start iteration. Throughout the process, new points are initialized via nearest-neighbor Brenier potential, and the converged solution from the previous layer serves as the warm-start initial value, so only a small number of fine-tuning iterations are needed per layer. (1) Stage 1: Initial Skeleton Solving. Only $|\mathcal{Y}_1| + |\mathcal{Y}_{N_{layer}}|$ target points are involved in this stage, and $\sum_{i=0}^{N} t_1^i$ iterations are performed until coarse-level topological convergence. The complexity of this stage is $C_1 = \mathcal{O}(\sum_{i=0}^{N} t_1^i \cdot 2^i \cdot N_{init} \log(|\mathcal{Y}_1| + |\mathcal{Y}_{N_{layer}}|))$. (2) Stage 2: Layer-Wise Incremental Update. At the $k$-th layer, $|\mathcal{Y}_1|$ new target points are added. With the converged Brenier heights as warm-start values, and new points initialized via nearest-neighbor potentials, only $\sum_{i=0}^{N} t_k^i$ iterations are required to fine-tune the Power-Voronoi cells and Brenier heights. For the $k$-th layer, the number of target points is $k \cdot |\mathcal{Y}_1| + |\mathcal{Y}_{N_{layer}}|$. The accumulated complexity over $N_{layer}$ layers is $C_2 = \sum_{k=2}^{N_{layer}-1} \mathcal{O}(\sum_{i=0}^{N} t_k^i \cdot 2^i \cdot N_{init} \log(k \cdot |\mathcal{Y}_1| + |\mathcal{Y}_{N_{layer}}|))$. (3) Overall Complexity. By aggregating the computational overhead of the two consecutive solving stages, the total complexity of the proposed hierarchical incremental solver is derived as $C_{prop} = C_1 + C_2 = \sum_{k=1}^{N_{layer}-1} \mathcal{O}(\sum_{i=0}^{N} t_k^i \cdot 2^i \cdot N_{init} \log(k \cdot |\mathcal{Y}_1| + |\mathcal{Y}_{N_{layer}}|))$.

Table 1: Comparison of Iteration Counts and RunTime (second) on Single RTX 4090 GPU across Different MC Sample Sizes. We sample 50000 points from 20-dimensional Exponential distribution as the discrete target set. Hyperparameter settings are provided in Section 4.1. Essentially, our hierarchical incremental Algorithm 2 distributes the time-consuming iterations induced by the large MC sample size in the original full-scale cold-start Algorithm 3 into iterations with smaller MC sample sizes. Experimental results demonstrate that the total runtime of our solver is only 8407.2294s, compared with 56329.52s for the full-scale cold-start solver, yielding a Speed-Up Ratio of 85.307491%.

| Methods | Count | $2^0 \cdot N_{init}$ | $2^1 \cdot N_{init}$ | $2^2 \cdot N_{init}$ | $2^3 \cdot N_{init}$ | $2^4 \cdot N_{init}$ | $2^5 \cdot N_{init}$ | $2^6 \cdot N_{init}$ | $2^7 \cdot N_{init}$ | $2^8 \cdot N_{init}$ | $2^9 \cdot N_{init}$ | RunTime |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Original | 50000 | 13700 | 14300 | 15500 | 3500 | 7000 | 4500 | 10500 | 6900 | 5600 | 5200 | 56329.52 |
| Ours ($r = 0.2$) | 20000 (1) | 25300 | 6200 | 2500 | 5700 | 2400 | 9700 | 0 | 0 | 0 | 0 | 1837.373 |
| | 30000 (2) | 0 | 0 | 0 | 10200 | 400 | 0 | 0 | 0 | 0 | 0 | 528.8834 |
| | 40000 (3) | 0 | 0 | 10900 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 334.0912 |
| | 50000 (4) | 0 | 0 | 10600 | 2500 | 2300 | 2000 | 1900 | 1600 | 400 | 0 | 5706.882 |

The core acceleration mechanism of our method 2 lies in decomposing and distributing the expensive iterations induced by large MC samples in the original full-scale cold-start algorithm 3 into multi-

layer small-sample iterations. The latter adopts a cold-start scheme without prior height information. Each iteration requires matching MC samples to all target points to update the cell partition in the source domain. Therefore, increasing the MC sample size to improve convergence and solution accuracy leads to an exponential surge in per-iteration computational cost. In contrast, our hierarchical algorithm mitigates the high overhead of large-MC iterations via two core designs: (1) the head-tail skeleton initialization stage adopts a coarse convergence criterion to lock the global topology of the Power diagram. With a small set of target points at this stage, only small MC samples are required to rapidly converge to a high-precision solution. (2) newly added points in each coarse intermediate incremental layer are initialized using nearest-neighbor Brenier potentials. Their initial heights lie close to the optimum and only require minor fine-tuning, without heavy optimization relying on large MC samples. With these two strategies, we obtain a high-quality initial solution before entering full-scale solving at the final fine layer. Even for full-scale computation in the final layer, only a small number of iterations with large MC samples are needed. A clear example is shown in Tab. 1.

# 4 EXPERIMENTS

## 4.1 COMPUTATIONAL EFFICIENCY AND ERROR EVALUATION

**Target Distributions.** To cover comprehensive test scenarios, we construct two categories of target distributions $p$: classical and real-world latent distributions. For the classical scenario, we adopt the standard Gaussian distribution, von-Mises distribution with mean of 0.0 and concentration parameter of 1.0, Exponential distributions with scale parameter of 1.0, symmetric Triangular distribution defined on $[0, 1]$ with its peak located at 0.5, and Weibull distribution with shape parameter of 1.5. For the real data scenario, we train feature encoders on Cifar10 (Krizhevsky et al., 2009), CelebA (Liu et al., 2015) and ImageNet (Deng et al., 2009) respectively, and take the output latent codes as $p$.

**Experimental Setup.** The source domain is uniformly set as $\Omega = [-0.5, 0.5]^d$ with uniform measure $\mu$, where $d$ denotes the dimension of the target distribution. The number of initial MC samples is set to $N_{init} = 10000$, with a maximum iteration number $T = 1000000$ and $s = 100$. To accelerate convergence, a larger learning rate $lr$ is adopted for higher-dimensional cases, and the detailed configurations are presented in Tab. 2. First, we sample $N$ points $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i\in\mathcal{I}} \overset{i.i.d}{\sim} p$ with $|\mathcal{Y}| = N$ to construct the target points. We set the termination thresholds as $\{\tau^{grad} = 1\times10^{-4}, \tau^{lr} = 1\times10^{-12}\}$ and employ the full-scale cold-start Algorithm 3 to solve the SDOT mapping from $\Omega$ to $\mathcal{Y}$. The corresponding Brenier height $\boldsymbol{h}_{GT}$ is treated as the ground truth. Subsequently, we preprocess the target set $\mathcal{Y}$ via hierarchical Algorithm 1 with a hierarchical ratio $r = 0.2$. The multi-scale termination thresholds are set to $\{\tau^{coarse} = 1\times10^{-3}, \tau^{fine} = 1\times10^{-4}, \tau^{lr} = 1\times10^{-12}\}$. Based on the proposed hierarchical incremental warm-start Algorithm 2, we solve the SDOT mapping from $\Omega$ to $\mathcal{Y}$ and obtain the Brenier height $\boldsymbol{h}_{HF}$.

Table 2: Learning Rate Setting for Dimensions.

| $d$ | 2 | 3 | 5 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|---|---|
| $lr$ | $1\times10^{-5}$ | $1\times10^{-5}$ | $5\times10^{-5}$ | $1\times10^{-4}$ | $1\times10^{-4}$ | $5\times10^{-4}$ | $5\times10^{-4}$ |

**Evaluation Metrics.** We conduct quantitative evaluation from two dimensions: (1) Solution Accuracy. Four metrics are utilized to quantify the numerical difference between the optimized Brenier height $\boldsymbol{h}_{HF}$ from the proposed hierarchical fast Algorithm 2 and the ground-truth height $\boldsymbol{h}_{GT}$ from the full-scale cold-start Algorithm 3: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Absolute Relative Error (ARE), and Cosine Similarity (CS). (2) Computational Efficiency. The indicator Speed-Up Ratio $= 1 - \frac{\text{RunTime}(\boldsymbol{h}_{HF})}{\text{RunTime}(\boldsymbol{h}_{GT})}$ is used to measure the GPU time benefit of our hierarchical incremental solver, where $\text{RunTime}(\cdot)$ represents the GPU runtime for solving Brenier heights.

**Results and Analysis.** Figs. 2(a)-(f), 6 illustrate the trends of solving efficiency and numerical error of our method against sample size $N$ under various distributions, with additional quantitative results summarized in Tab. 3. The experimental results demonstrate that the proposed hierarchical acceleration method achieves a prominent acceleration performance: the Speed-Up Ratio is mostly maintained above 40% as $N$ increases, with only a few individual cases falling below this threshold, while the numerical error is predominantly kept at the permille level.

To investigate the influence of $r$, we fix $N = 50000$ and tune $r$ from $0.05$ to $0.5$ with step size $0.05$. The corresponding acceleration efficiency and error variations are reported in Figs. 2(g)-(l). Empirically, desirable efficiency gains are obtained within $r \in [0.1, 0.45]$, whereas excessively small or large $r$ degrades performance. Specifically, $r \geq 0.5$ yields $N_{layer} = 2$, and our algorithm degenerates

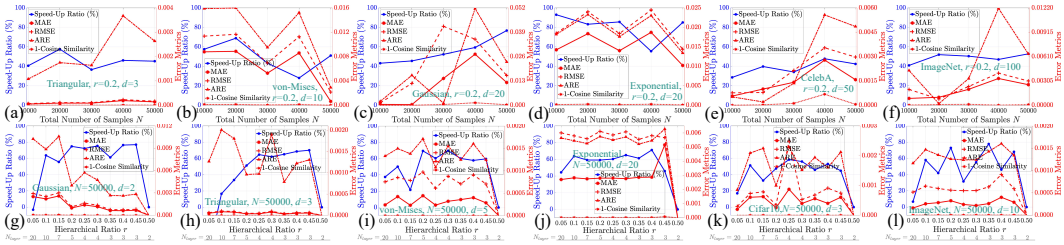


Fig 2: Computational Efficiency and Error Evaluation of Our Method over Various $N$ and $r$.

Table 3: Computational Efficiency and Error Evaluation of Our Method over Various Distributions on Single RTX 4090 GPU. $r$ is fixed at 0.2. ↓/↑ mark the desired value direction.

| $p$ | $d$ | $N$ | RunTime(second)↓ Baseline $\boldsymbol{h}_{GT}$ | RunTime(second)↓ Ours $\boldsymbol{h}_{HF}$ | Speed-Up Ratio↑ | Metrics MAE↓ | Metrics RMSE↓ | Metrics ARE↓ | Metrics CS↑ |
|---|---|---|---|---|---|---|---|---|---|
| Gaussian | 2 | 50000 | 44190.78785 | 24541.57589 | 44.464498% | 0.000693 | 0.000890 | 0.003126 | 0.999998 |
| | 5 | 50000 | 27590.89047 | 12522.79223 | 54.612584% | 0.000302 | 0.000577 | 0.001297 | 0.999999 |
| | 50 | 50000 | 41014.64147 | 22632.08273 | 44.819504% | 0.001729 | 0.004667 | 0.006098 | 0.999981 |
| | 100 | 50000 | 76190.18382 | 35447.18510 | 53.475391% | 0.001943 | 0.004877 | 0.006112 | 0.999979 |
| Von-Mises | 2 | 40000 | 32308.83481 | 13936.96914 | 56.863288% | 0.000555 | 0.000709 | 0.001944 | 0.999999 |
| | 20 | 50000 | 21088.78387 | 9923.386282 | 52.944720% | 0.001758 | 0.005506 | 0.006296 | 0.999980 |
| | 50 | 40000 | 6389.375845 | 2771.108218 | 56.629438% | 0.001265 | 0.002419 | 0.002548 | 0.999997 |
| | 100 | 50000 | 63153.76149 | 39103.16045 | 38.082611% | 0.002576 | 0.010087 | 0.010325 | 0.999947 |
| Exponential | 2 | 10000 | 23885.40922 | 11846.86582 | 50.401244% | 0.000273 | 0.000542 | 0.001321 | 0.999999 |
| | 3 | 50000 | 188632.4905 | 66064.20117 | 64.977295% | 0.000001 | 0.000002 | 0.000004 | 1.000000 |
| | 5 | 10000 | 2731.084435 | 692.8908070 | 74.629462% | 0.000554 | 0.000940 | 0.001479 | 0.999999 |
| | 10 | 50000 | 28600.81935 | 3930.404657 | 86.257720% | 0.001004 | 0.002663 | 0.003102 | 0.999995 |
| | 20 | 50000 | 56329.51593 | 8407.229379 | 85.074913% | 0.010219 | 0.014386 | 0.013408 | 0.999910 |
| | 100 | 50000 | 8489.400468 | 6607.630974 | 22.166106% | 0.003629 | 0.009644 | 0.006857 | 0.999977 |
| Triangular | 2 | 50000 | 43203.97573 | 7929.604832 | 81.646122% | 0.000284 | 0.000335 | 0.006539 | 0.999983 |
| | 3 | 30000 | 1721.111855 | 1091.008549 | 36.610247% | 0.000086 | 0.000103 | 0.001643 | 0.999999 |
| | 5 | 40000 | 939.7378380 | 609.2023340 | 35.173161% | 0.000024 | 0.000064 | 0.000793 | 1.000000 |
| | 20 | 50000 | 7636.216427 | 3554.268723 | 53.455108% | 0.000288 | 0.000660 | 0.005311 | 0.999986 |
| | 100 | 40000 | 2067.834412 | 1026.412986 | 50.362902% | 0.000792 | 0.001379 | 0.009923 | 0.999951 |
| Weibull | 2 | 20000 | 33296.86238 | 20271.61232 | 39.118551% | 0.000421 | 0.001182 | 0.002877 | 0.999997 |
| | 3 | 50000 | 29533.83604 | 8980.889039 | 69.591187% | 0.000919 | 0.000572 | 0.004847 | 0.999992 |
| | 5 | 50000 | 5578.780666 | 3194.148749 | 42.744679% | 0.000154 | 0.000332 | 0.001075 | 0.999999 |
| | 20 | 50000 | 22324.08464 | 4818.220727 | 78.416939% | 0.006743 | 0.008832 | 0.017834 | 0.997910 |
| | 50 | 50000 | 8364.493056 | 3315.657345 | 60.360331% | 0.001518 | 0.004192 | 0.007371 | 0.999841 |
| Cifar10 | 2 | 50000 | 28723.04411 | 15414.98554 | 46.332340% | 0.006003 | 0.005419 | 0.004218 | 0.999988 |
| | 5 | 50000 | 21422.01154 | 10084.68921 | 52.923706% | 0.009157 | 0.009475 | 0.007922 | 0.998261 |
| | 20 | 50000 | 12720.44125 | 6854.332415 | 46.115608% | 0.016557 | 0.003571 | 0.007491 | 0.997933 |
| | 100 | 50000 | 23574.01486 | 11879.41025 | 49.608031% | 0.001566 | 0.002448 | 0.011287 | 0.999721 |
| CelebA | 3 | 50000 | 20417.01456 | 9911.014578 | 51.457082% | 0.002147 | 0.002270 | 0.007134 | 0.999957 |
| | 5 | 50000 | 12408.02145 | 6721.014566 | 45.833310% | 0.006324 | 0.007752 | 0.002785 | 0.999992 |
| | 20 | 50000 | 15781.10455 | 7076.541233 | 55.158137% | 0.003469 | 0.005575 | 0.006964 | 0.999914 |
| | 50 | 50000 | 15142.65456 | 8716.210445 | 42.439349% | 0.001576 | 0.002971 | 0.004854 | 0.999944 |
| ImageNet | 2 | 40000 | 53575.99321 | 13694.98169 | 74.438212% | 0.001093 | 0.001307 | 0.004607 | 0.999991 |
| | 10 | 50000 | 18799.54459 | 10256.02445 | 45.445357% | 0.003554 | 0.002981 | 0.003118 | 0.999968 |
| | 50 | 50000 | 15741.02556 | 8798.874552 | 44.102279% | 0.001652 | 0.001974 | 0.001834 | 0.999978 |
| | 100 | 50000 | 26147.95243 | 12337.01455 | 52.818430% | 0.002578 | 0.003042 | 0.006441 | 0.999921 |

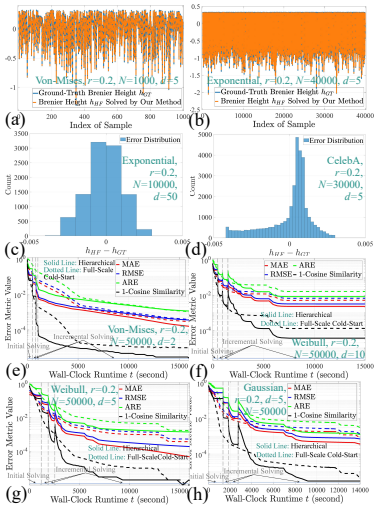


Fig 3: (a)-(b): Comparison of Brenier Heights. (c)-(d): Error Distribution of Brenier Heights. (e)-(h): Convergence Curves Versus RunTime Under Different Error Metrics.

to the original full-scale cold-start solver 3 with Speed-Up Ratio of 1. By contrast, too small $r$ produces excessive layers with sharply reduced per-layer point counts. The constructed head-tail skeleton fails to adequately cover the whole target set, such that nearest-neighbor-based Brenier height initialization and incremental updates cannot fully exploit the benefits of hierarchical processing.

Moreover, Figs. 3(a)-(d), 4, 5 show component-wise comparisons between the Brenier heights $\boldsymbol{h}_{HF}$ solved by our hierarchical algorithm 2 and ground-truth values $\boldsymbol{h}_{GT}$, together with their error distributions. All component-wise errors concentrate at the permille level. Figs. 3(e)-(h), 7 further depict the linear convergence curves of two methods under different error metrics. We use runtime instead of iteration count as the horizontal axis of convergence, since the per-iteration cost differs greatly between algorithms. Evaluations based on wall-clock time reflect the true convergence performance more objectively. Compared with the full-scale cold-start method, our approach exhibits sharp drops in convergence curves upon the completion of each layer, and this phenomenon is independent of error metrics, substantially reducing the time to reach a given accuracy.

We further conduct comparative experiments across multiple distribution scenarios between our approach and three methods, namely the NeuralOT solver (Li et al., 2024a), Stochastic Multi-Layer Algorithm (SMLA) (Leclaire & Rabin, 2021), and SDOT solver embedded in the POT library (POT-SDOT) (Flamary et al., 2021), with quantitative results listed in Tab. 4. NeuralOT achieves an extremely low inference time but relies on massive data for pre-training, yielding numerical errors two orders of magnitude higher than our method. As an approximate solver, SMLA (4 layers) suffers from inherent theoretical biases for $d \geq 2$. Although its computational time is comparable to that of

Table 4: Computational Efficiency and Error Evaluation of Our Method and Other SDOT Solvers.

| $p$ | Gaussian ($N = 50000, d = 10$) | | | Exponential ($N = 50000, d = 20$) | | | Weibull ($N = 50000, d = 5$) | | | CelebA ($N = 50000, d = 50$) | | | ImageNet ($N = 50000, d = 50$) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | Training↓ | Solving↓ | ARE↓ | Training↓ | Solving↓ | ARE↓ | Training↓ | Solving↓ | ARE↓ | Training↓ | Solving↓ | ARE↓ | Training↓ | Solving↓ | ARE↓ |
| NeuralOT | 25578.4s | 3.02441s | 0.178221 | 27956.5s | 3.35514s | 0.253562 | 14687.4s | 2.74551s | 0.158774 | 35510.4s | 3.67422s | 0.234102 | 32542.8s | 3.58716s | 0.257413 |
| SMLA | — | 19448.7s | 0.101563 | — | 7978.55s | 0.033451 | — | 3398.64s | 0.023055 | — | 8076.15s | 0.059742 | — | 6849.08s | 0.046855 |
| POT-SDOT | — | 66874.4s | 0.045122 | — | 60086.7s | 0.018661 | — | 13680.2s | 0.016884 | — | 18845.7s | 0.034015 | — | 22475.9s | 0.038415 |
| Ours ($r = 0.2$) | — | 24257.8s | 0.003778 | — | 8407.23s | 0.013408 | — | 3194.15s | 0.001075 | — | 8716.21s | 0.004854 | — | 8798.88s | 0.001834 |

ours, its average ARE is 17.74 times larger. POT-SDOT only supports CPU computation, leading to substantial runtime overhead for large-scale target points, with an average ARE 11.40 times that of ours. In contrast, the proposed algorithm requires no pre-training and enables unbiased solutions in arbitrary $d$, outperforming all baselines in both computational efficiency and numerical accuracy.

### 4.2 Generalization Validation on Downstream Tasks

To verify the practical value and scenario adaptability of the proposed method, we conduct validation on downstream scenarios. Existing studies have demonstrated that SDOT mappings are widely applicable to image (Li et al., 2025a) and 3D point cloud (Zhang et al., 2022) generation, adversarial attack (Li et al., 2025b), and OOD overconfidence calibration (Tang et al., 2026). These methods follow a unified paradigm: (1) raw data are mapped into the latent space via a pre-trained encoder to obtain the latent code set $\mathcal{Z}$; (2) the SDOT mapping $T_{ot}$ from the regular source domain $\Omega$ to $\mathcal{Z}$ is solved.

For generative tasks, noises are randomly sampled from $\Omega$, and corresponding latent codes are obtained via extended $T_{ot}$, which are then decoded to the original data space (Li et al., 2025a; Zhang et al., 2022). For adversarial and OOD sample generation, methods (Li et al., 2025b; Tang et al., 2026) identify the singular set $\mathcal{S}$ of $T_{ot}$ via dihedral angles between adjacent hyperplanes of Power-Voronoi cells, construct latent codes along $\mathcal{S}$, and decode them to generate anomalous samples.

**Experimental Design.** We replace the globally solved $T_{ot}$ from the full-scale cold-start Algorithm 3 in original methods with the solution of our hierarchical fast Algorithm 2, to evaluate its impact on downstream task performance. Only the SDOT solving module is replaced during experiments; the rest of the model architecture and training configurations remain fixed. All benchmark settings are consistent with original studies to ensure fair and comparable evaluation.

**Datasets, Baselines and Metrics.** We conduct experiments across four tasks: image generation, point cloud generation, adversarial attack, and OOD overconfidence evaluation. The dataset configurations, state-of-the-art (SOTA) baselines adopting the full-scale cold-start SDOT solver 3, and evaluation metrics for each task are detailed in Appendix D.

**Results and Analysis.** Quantitative results in Tab. 5 demonstrate that models embedded with our hierarchical accelerated SDOT solving module achieve highly consistent metrics with the original baselines, with ARE at the percentage-to-permille level. This verifies the favorable adaptability and generalization of our hierarchical incremental fast algorithm across diverse scenarios: it significantly improves solving efficiency while fully preserving downstream task performance.

Table 5: Impact of Our Method on Model Performance Across Four Scenarios.

| Tasks | Image Generation (FID↓) | | | | Point Cloud Generation (CD×100 ↓/EMD↓) | | Adversarial Attack (ASR%↑/CD×100 ↓) | | | OOD Overconfidence (MMC↑/FPR95↓) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | Cifar10 | CelebA | CelebA-HQ | LSUN | ShapeNet (4) | ShapeNet (8) | PointNet | PointConv | DGCNN | iNaturalist | Places | Textures |
| Baseline | 2.904 | 1.871 | 4.763 | 5.512 | 2.808/6.383 | 2.798/5.931 | 100/0.224 | 82.8/0.243 | 88.7/0.251 | 31.01/49.16 | 45.78/68.08 | 38.81/66.50 |
| Ours | 2.911 | 1.856 | 4.773 | 5.497 | 2.835/6.301 | 2.787/5.955 | 100/0.229 | 83.4/0.232 | 89.9/0.260 | 32.28/51.34 | 44.38/69.41 | 37.77/67.07 |
| ARE↓ | 0.002 | 0.008 | 0.002 | 0.003 | 0.010/0.013 | 0.004/0.004 | 0.000/0.022 | 0.007/0.045 | 0.014/0.036 | 0.041/0.044 | 0.031/0.020 | 0.027/0.009 |

## 5 Conclusion

MC-based geometric variational solver circumvents the intractable analytical computation of Power-Voronoi cell volumes in SDOT. Nevertheless, their global cold-start synchronous iteration leads to a sharp surge in partitioning overhead with the growth of target point scale, substantially degrading the overall solving efficiency. To address this computational limitation, this work proposes a novel hierarchical fast solving framework. We first partition target points layer by layer in a nearest-to-farthest order and constructs a global skeleton via head and tail layers to pre-lock the global topology of the Power diagram. During incremental optimization, newly added points are warm-started using the Brenier heights of their nearest solved neighbors and fine-tuned layer-wisely to obtain the optimal Brenier heights for all target points. Extensive experiments on both standard and complex real-world distributions demonstrate that the proposed method achieves a stable average speed-up of 53.88% over the original MC solver, with numerical errors consistently maintained at the permille level. Further evaluations on downstream tasks including generative modeling, adversarial attacks, and OOD overconfidence verification validate that our method only introduces negligible percentage-to-permille performance fluctuations, exhibiting superior practicality and robustness.

### AI USE STATEMENT

In this work, we used generative AI tools for polishing manuscript text and minor auxiliary code formatting. We have not used generative AI tools for formulating research ideas, designing experiments, analyzing experimental results, drawing conclusions, or writing the core content of the manuscript, and AI-assisted generation of research hypotheses, dataset construction, and full manuscript drafting are not applicable to this work. Additionally, we used generative AI tools for grammar refinement and minor code snippet adjustment. We have reviewed all AI-assisted work. All AI-polished text was manually revised and checked by the authors; AI-assisted code was fully examined, tested and modified by the authors to ensure correctness. No research ideas, interpretations or conclusions were generated by AI. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

All theoretical assumptions, derivations, and complete proofs are elaborated in the main text and appendix to ensure theoretical reproducibility. For empirical experiments, we adopt publicly available datasets, and all implementation details including data preprocessing pipelines, hyperparameters, and evaluation settings are fully documented in the appendix. The complete source code will be publicly released upon paper acceptance.

## REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pp. 40–49. PMLR, 2018.

Pankaj Agarwal, Sharath Raghvendra, Pouyan Shirzadian, and Keegan Yao. Efficient algorithms for robust and partial semi-discrete optimal transport. *Advances in Neural Information Processing Systems*, 38:44053–44108, 2026.

Pankaj K Agarwal, Sharath Raghvendra, Pouyan Shirzadian, and Keegan Yao. A combinatorial algorithm for the semi-discrete optimal transport problem. *Advances in Neural Information Processing Systems*, 37:25857–25887, 2024a.

Pankaj K Agarwal, Sharath Raghvendra, Pouyan Shirzadian, and Keegan Yao. Fast and accurate approximations of the optimal transport in semi-discrete and discrete settings. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 4514–4529. SIAM, 2024b.

Jason M Altschuler, Jonathan Niles-Weed, and Austin J Stromme. Asymptotics for semidiscrete entropic optimal transport. *SIAM Journal on Mathematical Analysis*, 54(2):1718–1741, 2022.

Luca Ambrogioni, Umut Güçlü, and Marcel van Gerven. k-gans: Ensemble of generative models with semi-discrete optimal transport. *arXiv preprint arXiv:1907.04050*, 2019.

Dongsheng An, Yang Guo, Na Lei, Zhongxuan Luo, Shing-Tung Yau, and Xianfeng Gu. Ae-ot: A new generative model based on extended semi-discrete optimal transport. In *International Conference on Learning Representations*, 2020a.

Dongsheng An, Yang Guo, Min Zhang, Xin Qi, Na Lei, and Xianfang Gu. Ae-ot-gan: Training gans from data specific latent distribution. In *European Conference on Computer Vision*, pp. 548–564. Springer, 2020b.

JR Ashworth and EL McLellan. Textures. In *Migmatites*, pp. 180–203. Springer, 1985.

Mohit Bansil and Jun Kitagawa. A newton algorithm for semidiscrete optimal transport with storage fees. *SIAM Journal on Optimization*, 31(4):2586–2613, 2021.

Mohit Bansil and Jun Kitagawa. Quantitative stability in the geometry of semi-discrete optimal transport. *International Mathematics Research Notices*, 2022(10):7354–7389, 2022.

David P Bourne, Bernhard Schmitzer, and Benedikt Wirth. Semi-discrete unbalanced optimal transport and quantization. *European Journal of Applied Mathematics*, pp. 1–43, 2018.

David P Bourne, Piet JJ Kok, Steven M Roper, and Wil DT Spanjer. Laguerre tessellations and polycrystalline microstructures: a fast algorithm for generating grains of given volumes. *Philosophical Magazine*, 100(21):2677–2707, 2020.

David P Bourne, Charlie P Egan, Beatrice Pelloni, and Mark Wilkinson. Semi-discrete optimal transport methods for the semi-geostrophic equations. *Calculus of Variations and Partial Differential Equations*, 61(1):39, 2022.

David P Bourne, Charlie P Egan, Théo Lavier, and Beatrice Pelloni. Semi-discrete optimal transport techniques for the compressible semi-geostrophic equations. *arXiv preprint arXiv:2504.20807*, 2025a.

David P Bourne, Mason Pearce, and Steven M Roper. Inverting laguerre tessellations: recovering tessellations from the volumes and centroids of their cells using optimal transport. *ESAIM: Mathematical Modelling and Numerical Analysis*, 59(2):841–871, 2025b.

DP Bourne, M Pearce, and SM Roper. Geometric modelling of polycrystalline materials: Laguerre tessellations and periodic semi-discrete optimal transport. *Mechanics Research Communications*, 127:104023, 2023.

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

Maciej Buze, Jean Feydy, Steven M Roper, Karo Sedighiani, and David P Bourne. Anisotropic power diagrams for polycrystal modelling: Efficient generation of curved grains via optimal transport. *Computational Materials Science*, 245:113317, 2024.

Philip Claude Caplan. Higher-dimensional power diagrams for semi-discrete optimal transport. *arXiv preprint arXiv:2106.14730*, 2021.

Yucheng Chen, Matus Telgarsky, Chao Zhang, Bolton Bailey, Daniel Hsu, and Jian Peng. A gradual, semi-discrete approach to generative network training via explicit wasserstein minimization. In *International Conference on Machine Learning*, pp. 1071–1080. PMLR, 2019.

Charles Dapogny, Bruno Lévy, and Edouard Oudet. A lagrangian shape and topology optimization framework based on semi-discrete optimal transport. *arXiv preprint arXiv:2409.07873*, 2024.

Pedro Machado Manhaes De Castro, Quentin Mérigot, and Boris Thibert. Far-field reflector problem and intersection of paraboloids. *Numerische Mathematik*, 134(2):389–411, 2016.

Frédéric De Gournay, Jonas Kahn, and Léo Lebrat. Differentiation and regularity of semi-discrete optimal transport with respect to the parameters of the discrete measure. *arXiv preprint arXiv:1803.00827*, 2018.

Eustasio Del Barrio, Alberto González Sanz, and Jean-Michel Loubes. Central limit theorems for semi-discrete wasserstein distances. *Bernoulli*, 30(1):554–580, 2024.

Alex Delalande. Nearly tight convergence bounds for semi-discrete entropic optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pp. 1619–1642. PMLR, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Luca Dieci and Daniyar Omarov. Solving semi-discrete optimal transport problems: star shapedeness and newton's method. *Numerical Algorithms*, 99(2):949–1004, 2025.

Luca Dieci and Joseph D Walsh III. The boundary method for semi-discrete optimal transport partitions and wasserstein distance computation. *Journal of Computational and Applied Mathematics*, 353:318–344, 2019.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Nicolás García Trillos and Javier Morales. Semi-discrete optimization through semi-discrete optimal transport: a framework for neural architecture search. *Journal of Nonlinear Science*, 32(3):27, 2022.

Ferdinand Genans, Antoine Godichon-Baggioni, François-Xavier Vialard, and Olivier Wintenberger. Semi-discrete optimal transport: Nearly minimax estimation with stochastic gradient descent and adaptive entropic regularization. 2024.

Ferdinand Genans, Antoine Godichon-Baggioni, François-Xavier Vialard, and Olivier Wintenberger. Stochastic optimization in semi-discrete optimal transport: Convergence analysis and minimax rate. *Advances in Neural Information Processing Systems*, 38:51490–51527, 2026.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016.

Xianfeng Gu, Feng Luo, Jian Sun, and S-T Yau. Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations. *arXiv preprint arXiv:1302.5472*, 2013.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational visual media*, 7(2):187–199, 2021.

Valentin Hartmann and Dominic Schuhmacher. Semi-discrete optimal transport-the case p= 1. *arXiv preprint arXiv:1706.07650*, 2017.

Valentin Hartmann and Dominic Schuhmacher. Semi-discrete optimal transport: a solution procedure for the unsquared euclidean distance case. *Mathematical Methods of Operations Research*, 92(1):133–163, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6629–6640, 2017.

Antoine Houdard, Arthur Leclaire, Nicolas Papadakis, and Julien Rabin. On the gradient formula for learning generative models with regularized optimal transport costs. 2023.

Lian-Bao Jin, Na Lei, Zhong-Xuan Luo, Jin Wu, Chao Ai, and Xianfeng Gu. Semi-discrete optimal transport for long-tailed classification. *Journal of Computer Science and Technology*, 40(1):252–266, 2025.

Lianbao Jin, Dayu Lang, and Na Lei. An optimal transport view of class-imbalanced visual recognition. *International Journal of Computer Vision*, 131(11):2845–2863, 2023.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Jun Kitagawa, Quentin Mérigot, and Boris Thibert. Convergence of a newton algorithm for semi-discrete optimal transport. *Journal of the European Mathematical Society*, 21(9), 2019.

Lingkai Kong, Molei Tao, Yang Liu, Bryan Wang, Jinmiao Fu, Chien-Chih Wang, and Huidong Liu. Alignflow: Improving flow-based generative models with semi-discrete optimal transport. *arXiv preprint arXiv:2510.15038*, 2025.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Théo Lavier. A semi-discrete optimal transport scheme for the 3d incompressible semi-geostrophic equations. *IMA Journal of Applied Mathematics*, pp. hxaf023, 2025.

Théo Lavier and Beatrice Pelloni. A semi-discrete optimal transport scheme for the semi-geostrophic slice compressible model. *arXiv preprint arXiv:2603.08105*, 2026.

Arthur Leclaire and Julien Rabin. A fast multi-layer approximation to semi-discrete optimal transport. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 341–353. Springer, 2019.

Arthur Leclaire and Julien Rabin. A stochastic multi-layer algorithm for semi-discrete optimal transport with applications to texture synthesis and style transfer. *Journal of Mathematical Imaging and Vision*, 63(2):282–308, 2021.

Bruno Lévy. A numerical algorithm for $l_\{2\}$ semi-discrete optimal transport in 3d. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1693–1715, 2015.

Bruno Lévy. Partial optimal transport for a constant-volume lagrangian mesh with free boundaries. *Journal of Computational Physics*, 451:110838, 2022.

Bruno Lévy and Erica L Schwindt. Notions of optimal transport theory and how to implement them on a computer. *Computers & Graphics*, 72:135–148, 2018.

Bruno Lévy, Roya Mohayaee, and Sebastian von Hausegger. A fast semidiscrete optimal transport algorithm for a unique reconstruction of the early universe. *Monthly Notices of the Royal Astronomical Society*, 506(1):1165–1185, 2021.

Bruno Lévy, Nicolas Ray, Quentin Mérigot, and Hugo Leclerc. Large-scale semi-discrete optimal transport with distributed voronoi diagrams. *Journal of Computational Physics*, pp. 114374, 2025.

Shenghao Li, Zezeng Li, Zhanpeng Wang, Zebin Xu, Na Lei, and Zhongxuan Luo. Measure-driven neural solver for optimal transport mapping. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(10):10437–10448, 2024a.

Shenghao Li, Zhanpeng Wang, Zhongxuan Luo, and Na Lei. An optimal transport-guided diffusion framework with mitigating mode mixture. *Neurocomputing*, 616:128910, 2025a.

Zezeng Li, Shenghao Li, Zhanpeng Wang, Na Lei, Zhongxuan Luo, and David Xianfeng Gu. Dpm-ot: a new diffusion probabilistic model based on optimal transport. In *Proceedings of the ieee/cvf international conference on computer vision*, pp. 22624–22633, 2023.

Zezeng Li, Shenghao Li, Lianbao Jin, Na Lei, and Zhongxuan Luo. Ot-net: a reusable neural optimal transport solver. *Machine Learning*, 113(3):1243–1268, 2024b.

Zezeng Li, Xiaoyu Du, Na Lei, Liming Chen, and Weimin Wang. Nopain: No-box point cloud attack via optimal transport singular boundary. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3492–3502, 2025b.

Judy Yangjun Lin, Shaoyan Guo, Longhan Xie, Ruxu Du, and Gu Xu. A discrete method for the initialization of semi-discrete optimal transport problem. *Knowledge-Based Systems*, 212:106608, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. *arXiv preprint arXiv:1909.12180*, 2019.

Quentin Mérigot and Jean-Marie Mirebeau. Minimal geodesics along volume-preserving maps, through semidiscrete optimal transport. *SIAM Journal on Numerical Analysis*, 54(6):3465–3492, 2016.

Quentin Mérigot, Jocelyn Meyron, and Boris Thibert. An algorithm for optimal transport between a simplex soup and a point cloud. *SIAM Journal on Imaging Sciences*, 11(2):1363–1389, 2018.

Jocelyn Meyron. Initialization procedures for discrete and semi-discrete optimal transport. *Computer-Aided Design*, 115:13–22, 2019.

Alireza Mousavi-Hosseini, Stephen Y Zhang, Michal Klein, and Marco Cuturi. Flow matching with semidiscrete couplings. *arXiv preprint arXiv:2509.25519*, 2025.

Luca Nenna, Daniyar Omarov, and Brendan Pass. Characterizing and computing solutions to regularized semi-discrete optimal transport via an ordinary differential equation. *arXiv preprint arXiv:2504.03030*, 2025.

Farnik Nikakhtar, Ravi K Sheth, Bruno Lévy, and Roya Mohayaee. Optimal transport reconstruction of baryon acoustic oscillations. *Physical Review Letters*, 129(25):251101, 2022.

Davy Paindaveine and Riccardo Passeggeri. On the robustness of semi-discrete optimal transport. *arXiv preprint arXiv:2410.19596*, 2024.

Aram-Alexandre Pooladian, Vincent Divol, and Jonathan Niles-Weed. Minimax estimation of discontinuous optimal transport maps: The semi-discrete case. In *International Conference on Machine Learning*, pp. 28128–28150. PMLR, 2023.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Ritwik Sadhu, Ziv Goldfeld, and Kengo Kato. Stability and statistical inference for semidiscrete optimal transport maps. *The Annals of Applied Probability*, 34(6):5694–5736, 2024.

Nathan A Seifert, Kirill Prozument, and Michael J Davis. Computational optimal transport for molecular spectra: The semi-discrete case. *The Journal of Chemical Physics*, 156(13), 2022.

Keke Tang, Ziyong Du, Xiaofei Wang, Weilong Peng, Peican Zhu, and Zhihong Tian. Optimal transport-induced samples against out-of-distribution overconfidence. *arXiv preprint arXiv:2601.21320*, 2026.

Bahar Taşkesen, Soroosh Shafieezadeh-Abadeh, and Daniel Kuhn. Semi-discrete optimal transport: Hardness, regularization and numerical solution. *Mathematical Programming*, 199(1): 1033–1106, 2023.

Evgeny Tenetov, Gershon Wolansky, and Ron Kimmel. Fast entropic regularized optimal transport using semidiscrete cost approximation. *SIAM Journal on Scientific Computing*, 40(5):A3400–A3422, 2018.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Sebastian von Hausegger, Bruno Lévy, and Roya Mohayaee. Accurate baryon acoustic oscillations reconstruction via semidiscrete optimal transport. *Physical Review Letters*, 128(20):201302, 2022.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5): 1–12, 2019.

Zhanpeng Wang, Shuting Cao, Yuhang Lu, Na Lei, Zhongxuan Luo, et al. Ot-ald: Aligning latent distributions with optimal transport for accelerated image-to-image translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26760–26768, 2026.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 9621–9630, 2019.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Peng Xu, Changbo Zhu, and Xiaohui Chen. A unified approach for computing wasserstein barycenters of discrete and continuous measures. *arXiv preprint arXiv:2605.11270*, 2026.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Jinlai Zhang, Yinpeng Dong, Jun Zhu, Jihong Zhu, Minchi Kuang, and Xiaming Yuan. Improving transferability of 3d adversarial attacks with scale and shear transformations. *Information Sciences*, 662:120245, 2024.

Ruonan Zhang, Jingyi Chen, Wei Gao, Ge Li, and Thomas H Li. Pointot: interpretable geometry-inspired point cloud generative model via optimal transport. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10):6792–6806, 2022.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Antonio Torralba, and Aude Oliva. Places: An image database for deep scene understanding. *ArXiv*, abs/1610.02055, 2016. URL `https://api.semanticscholar.org/CorpusID:14078472`.

Yinchu Zhu and Ilya O Ryzhov. Semidiscrete optimal transport with unknown costs. *arXiv preprint arXiv:2310.00786*, 2023.

Yinchu Zhu and Ilya O Ryzhov. Quantile optimization in semidiscrete optimal transport. *arXiv preprint arXiv:2602.10515*, 2026.

# Appendix

## A Main Algorithms

**Algorithm 2** Hierarchical Fast Solving for SDOT

**Input**: Hierarchically Preprocessed $\mathcal{H} = \{(\mathcal{Y}_1, \boldsymbol{\nu}_1, \mathcal{I}_1), \cdots, (\mathcal{Y}_{N_{layer}}, \boldsymbol{\nu}_{N_{layer}}, \mathcal{I}_{N_{layer}})\}$; Source Domain $\Omega$ with Measure $\mu$; Initial MC Sample Size $N_{init}$; Learning Rate $lr$; Maximum Iterations $T$; Integer $s > 0$ and Multi-Scale Termination Thresholds $\{\tau^{coarse}, \tau^{fine}, \tau^{lr}\}$.

**for** $k = 1 : N_{layer} - 1$ **do**
  **if** $k = 1$ **then**
    $\boldsymbol{\nu} \leftarrow (\boldsymbol{\nu}_1, \boldsymbol{\nu}_{N_{layer}})/(sum(\boldsymbol{\nu}_1) + sum(\boldsymbol{\nu}_{N_{layer}}))$.
    $\mathcal{Y} \leftarrow \mathcal{Y}_1 \cup \mathcal{Y}_{N_{layer}}$; $\mathcal{I} \leftarrow \mathcal{I}_1 \cup \mathcal{I}_{N_{layer}}$; $\boldsymbol{h}^{(0)} \leftarrow \mathbf{0}_{1\times|\mathcal{Y}|}$. ▷{Initial Target Set Construction}
  **else**
    $h_j^k \leftarrow h_\lambda^{current}, \lambda = \arg\min_{i\in\mathcal{I}} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2, j \in \mathcal{I}_k$. ▷{Nearest-Neighbor Brenier Height Initialization}
    $\boldsymbol{h}^k \leftarrow (h_j^k)_{j\in\mathcal{I}_k}$; $\boldsymbol{h}^{(0)} \leftarrow (\boldsymbol{h}^{current}, \boldsymbol{h}^k)$; $\boldsymbol{\nu} \leftarrow (\boldsymbol{\nu}, \boldsymbol{\nu}_k)/(1 + sum(\boldsymbol{\nu}_k))$; $\mathcal{Y} \leftarrow \mathcal{Y} \cup \mathcal{Y}_k$; $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{I}_k$.
  **end if**
  $N_{mc} \leftarrow 2^0 \cdot N_{init}$ if $k = 1$, $\max\{\frac{1}{2^2} N_{mc}, 2^0 \cdot N_{init}\}$ if $1 < k < N_{layer} - 1$ otherwise $2^2 \cdot N_{init}$; $lr^k \leftarrow lr$.
  $\tau \leftarrow \tau^{coarse}$ if $k < N_{layer} - 1$ otherwise $\tau^{fine}$.
  **for** $t = 0 : T - 1$ **do**
    $W_i(\boldsymbol{h}^{(t)}) \leftarrow \{\boldsymbol{x} \in \Omega \mid \langle \boldsymbol{x}, \boldsymbol{y}_i\rangle + h_i^{(t)} \geq \langle \boldsymbol{x}, \boldsymbol{y}_j\rangle + h_j^{(t)}, \forall j \neq i, j \in \mathcal{I}\}, i \in \mathcal{I}$.
    $\nabla_{h_i} \hat{E}(\boldsymbol{h}^{(t)}) \leftarrow \frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} \mathbb{I}\{\boldsymbol{x}_j \in W_i(\boldsymbol{h}^{(t)})\} - \nu_i, i \in \mathcal{I}$, where $\{\boldsymbol{x}_j\}_{j=1}^{N_{mc}} \overset{i.i.d}{\sim} (\Omega, \mu)$.
    $\nabla \hat{E}(\boldsymbol{h}^{(t)}) \leftarrow \nabla \hat{E}(\boldsymbol{h}^{(t)}) - mean(\nabla \hat{E}(\boldsymbol{h}^{(t)}))$.
    $\boldsymbol{h}^{(t+1)} \leftarrow \boldsymbol{h}^{(t)} - lr^k \cdot \nabla \hat{E}(\boldsymbol{h}^{(t)})$; $\boldsymbol{h}^{current} \leftarrow \boldsymbol{h}^{(t+1)}$.
    $N_{mc} \leftarrow 2 \cdot N_{mc}$, $lr^k \leftarrow 0.9 \cdot lr^k$ if $\|\nabla \hat{E}(\boldsymbol{h}^{(t)})\|_2$ has not decreased for $s$ steps.
    break if $\|\nabla \hat{E}(\boldsymbol{h}^{(t)})\|_2 < \tau$ or $lr^k < \tau^{lr}$.
  **end for** ▷{Incremental Iterative Solution}
**end for**
$T_{ot}(\cdot) \leftarrow \nabla \max_{i\in\mathcal{I}}(\langle \cdot, \boldsymbol{y}_i\rangle + h_i^{current})$.

**Return**: SDOT Mapping $T_{ot}(\cdot)$.

**Algorithm 3** MC-Based Geometric Variational Solving for SDOT

**Input**: Source domain $\Omega$ with measure $\mu$, target dataset $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$ with measure $\boldsymbol{\nu} = (\nu_i)_{i\in\mathcal{I}}$, initial MC sample size $N_{init}$, learning rate $lr$, maximum iterations $T$, integer $s > 0$ and termination thresholds $\{\tau^{grad}, \tau^{lr}\}$.

$\boldsymbol{h}^{(0)} \leftarrow \mathbf{0}_{1\times|\mathcal{I}|}$; $N_{mc} \leftarrow 2^0 \cdot N_{init}$.
**for** $t = 0 : T - 1$ **do**
  $W_i(\boldsymbol{h}^{(t)}) \leftarrow \{\boldsymbol{x} \in \Omega \mid \langle \boldsymbol{x}, \boldsymbol{y}_i\rangle + h_i^{(t)} \geq \langle \boldsymbol{x}, \boldsymbol{y}_j\rangle + h_j^{(t)}, \forall j \neq i, j \in \mathcal{I}\}, i \in \mathcal{I}$.
  $\nabla_{h_i} \hat{E}(\boldsymbol{h}^{(t)}) \leftarrow \frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} \mathbb{I}\{\boldsymbol{x}_j \in W_i(\boldsymbol{h}^{(t)})\} - \nu_i, i \in \mathcal{I}$, where $\{\boldsymbol{x}_j\}_{j=1}^{N_{mc}} \overset{i.i.d}{\sim} (\Omega, \mu)$.
  $\nabla \hat{E}(\boldsymbol{h}^{(t)}) \leftarrow \nabla \hat{E}(\boldsymbol{h}^{(t)}) - mean(\nabla \hat{E}(\boldsymbol{h}^{(t)}))$.
  $\boldsymbol{h}^{(t+1)} \leftarrow \boldsymbol{h}^{(t)} - lr \cdot \nabla \hat{E}(\boldsymbol{h}^{(t)})$; $\boldsymbol{h}^{current} \leftarrow \boldsymbol{h}^{(t+1)}$.
  $N_{mc} \leftarrow 2 \cdot N_{mc}$, $lr \leftarrow 0.9 \cdot lr$ if $\|\nabla \hat{E}(\boldsymbol{h}^{(t)})\|_2$ has not decreased for $s$ steps.
  break if $\|\nabla \hat{E}(\boldsymbol{h}^{(t)})\|_2 < \tau^{grad}$ or $lr < \tau^{lr}$.
**end for**
$T_{ot}(\cdot) \leftarrow \nabla \max_{i\in\mathcal{I}}(\langle \cdot, \boldsymbol{y}_i\rangle + h_i^{current})$.

**Return**: SDOT mapping $T_{ot}(\cdot)$.

## B Additional Related Works

**Numerical Solvers for SDOT.** Four main SDOT solver paradigms have been developed. *(1) Exact Geometric Solvers.* Lévy et al. (Lévy, 2015; Lévy & Schwindt, 2018), and Kitagawa et al. (2019) combined Newton methods with exact power-diagram integration to build efficient 2D/3D solvers. Extensions include the $p = 1$ and unsquared Euclidean settings (Hartmann & Schuhmacher,

2017; 2020), high-dimensional power-diagram computation (Caplan, 2021), boundary-based formulations (Dieci & Walsh III, 2019), combinatorial near-linear-time approximations (Agarwal et al., 2024a;b), star-shaped convergence guarantees (Dieci & Omarov, 2025), and robust/partial SDOT variants (Agarwal et al., 2026). Unified frameworks for unbalanced transport (Bourne et al., 2018), quantile objectives (Zhu & Ryzhov, 2026), point-cloud transport (Mérigot et al., 2018), and Wasserstein barycenters (Xu et al., 2026) further enrich the exact-solver family. *(2) Large-Scale Engineering Solvers.* Warm-start initialization schemes (Meyron, 2019; Lin et al., 2021) reduce the Newton iteration burnin phase; distributed Voronoi diagrams scale SDOT beyond $10^8$ points (Lévy et al., 2025); and storage-efficient extensions (Bansil & Kitagawa, 2021) and semi-supervised cost-learning frameworks (Zhu & Ryzhov, 2023) address practical deployment constraints. *(3) Entropic-Regularized Solvers.* Stochastic optimization was first introduced by (Genevay et al., 2016) for scalable computation. Follow-up work includes second-order asymptotics (Altschuler et al., 2022), nearly tight Sinkhorn bounds (Delalande, 2022), minimax estimation for discontinuous maps (Pooladian et al., 2023), computational hardness results and regularized schemes (Taşkesen et al., 2023), adaptive SGD with near-minimax rates (Genans et al., 2024), ODE-characterized regularized power diagrams (Nenna et al., 2025), and cost-approximation acceleration (Tenetov et al., 2018). *(4) Neural Network-Based Solvers.* Most recently, OT-Net and Neural-OT (Li et al., 2024b;a) fine-tune pre-trained Brenier potentials to accommodate target-point variations.

**Cross-Domain Applications of SDOT.** *(1) Generative Modeling and Translation.* AE-OT (An et al., 2020a;b) maps continuous sources to discrete latents to alleviate GAN mode collapse; gradual training (Chen et al., 2019) and ensemble methods (Ambrogioni et al., 2019) improve diversity. SDOT couplings also enhance diffusion models (Li et al., 2023), image-to-image translation (Wang et al., 2026), flow-based frameworks (Kong et al., 2025), flow matching (Mousavi-Hosseini et al., 2025), 3D shape (Zhang et al., 2022) and mode-separated generation (Li et al., 2025a), with gradient formulas for regularized costs (Houdard et al., 2023) supporting end-to-end backpropagation. *(2) Robustness and Discriminative Learning.* Singular set of SDOT enable adversarial/OOD detection (Li et al., 2025b; Tang et al., 2026), while SDOT geometric priors improve class-imbalanced (Jin et al., 2023) and long-tailed (Jin et al., 2025) recognition by reweighting discrete class centers. *(3) Science and Engineering.* SDOT has been applied to baryon acoustic oscillation reconstruction in cosmology (Lévy et al., 2021; von Hausegger et al., 2022; Nikakhtar et al., 2022), semi-geostrophic atmospheric/ocean simulations (Bourne et al., 2022; Lavier, 2025; Bourne et al., 2025a), polycrystalline Laguerre microstructure recovery (Bourne et al., 2025b; 2020; 2023; Buze et al., 2024), geometric-optics reflector design and molecular spectra comparison (De Castro et al., 2016; Seifert et al., 2022), geodesics and meshes (Mérigot & Mirebeau, 2016; Lévy, 2022), topology optimization (Dapogny et al., 2024), and neural architecture search (García Trillos & Morales, 2022).

**Multi-Scale and Hierarchical Acceleration for SDOT.** Closest to our work are strategies that exploit multi-scale structure to accelerate SDOT convergence. For texture synthesis and style transfer, Leclaire & Rabin (2019; 2021) proposed multi-layer and stochastic multi-layer algorithms, which hierarchically cluster the target measure and perform layer-wise weighted nearest-neighbor search. However, these methods induce an inherent structural bias from fixed clustering topology that cannot be eliminated by weight optimization, and their joint all-layer optimization is heuristic without rigorous convergence guarantees. Our hierarchical incremental warm-start framework differs fundamentally from prior multi-scale methods: we progressively augment the target point set and solve a full SDOT subproblem at each layer, inheriting the converged Brenier potential from the coarser layer as initialization. This preserves exact optimality of the final solution while substantially reducing total iterations, eliminating the structural bias intrinsic to clustering-based multi-layer approaches.

# C The Main Theoretical Basis of This Paper

## C.1 Preliminary Lemmas

**Lemma 2** (Differentiability and Convexity of Energy Functional (Gu et al., 2013))**.** *Fix the target point set $\mathcal{Y}$. The functional $E(\boldsymbol{h}, \boldsymbol{\nu})$ is differentiable and strictly strongly convex on $\{\boldsymbol{h} \in \mathbb{R}^{|\mathcal{I}|} \mid \mu(W_i(\boldsymbol{h})) > 0, \forall i \in \mathcal{I}, sum(\boldsymbol{h}) = 0\}$.*

**Lemma 3** (Local Lipschitz Continuity of the Cell Measure Mapping (Kitagawa et al., 2019))**.** *Fix the target point set $\mathcal{Y}$ and define the cell measure mapping $\boldsymbol{\Phi}(\boldsymbol{h}) = (\mu(W_i(\boldsymbol{h})))_{i\in\mathcal{I}}$. Then $\boldsymbol{\Phi}$ :*

$\mathbb{R}^{|\mathcal{I}|} \to \mathbb{R}^{|\mathcal{I}|}$ *is locally Lipschitz continuous, and its Jacobian matrix is positive definite everywhere* $\mathbf{J}_{\boldsymbol{\Phi}}(\boldsymbol{h}) = \nabla \boldsymbol{\Phi}(\boldsymbol{h}) \succeq \lambda \boldsymbol{I}, \lambda > 0$.

**Lemma 4.** *Fix the target point set* $\mathcal{Y}$*. Let* $\boldsymbol{\nu}'$ *and* $\boldsymbol{\nu}''$ *be two sets of target measures, with corresponding optimal Brenier heights* $\boldsymbol{h}'$ *and* $\boldsymbol{h}''$*. Then there exists a constant* $C > 0$ *(depending only on* $\mathcal{Y}$*) such that* $\|\boldsymbol{h}' - \boldsymbol{h}''\|_2 \leq C \cdot \|\boldsymbol{\nu}' - \boldsymbol{\nu}''\|_2$.

### C.2 The Proof of Lemma 1

*Proof.* Choose arbitrary two points $\boldsymbol{x}_i \in W_i(\boldsymbol{h})$ and $\boldsymbol{x}_j \in W_j(\boldsymbol{h})$. From the definition of Power-Voronoi cells, we have

$$\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle - h_i \geq \langle \boldsymbol{x}_i, \boldsymbol{y}_j \rangle - h_j \implies h_i - h_j \leq \langle \boldsymbol{x}_i, \boldsymbol{y}_i - \boldsymbol{y}_j \rangle. \tag{9}$$

Similarly,

$$\langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle - h_j \geq \langle \boldsymbol{x}_j, \boldsymbol{y}_i \rangle - h_i \implies h_j - h_i \leq \langle \boldsymbol{x}_j, \boldsymbol{y}_j - \boldsymbol{y}_i \rangle. \tag{10}$$

By the Cauchy–Schwarz inequality, we obtain

$$|h_i - h_j| \leq \max\{\|\boldsymbol{x}_i\|_2, \|\boldsymbol{x}_j\|_2\} \cdot \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2. \tag{11}$$

Since $\boldsymbol{x}_i, \boldsymbol{x}_j \in \Omega$, we have $\max\{\|\boldsymbol{x}_i\|, \|\boldsymbol{x}_j\|\} \leq \mathrm{norm}(\Omega)$. The desired inequality is thus proved. □

### C.3 The Proof of Lemma 4

*Proof.* By the differentiability and convexity of $E(\boldsymbol{h}, \boldsymbol{\nu})$ in Lemma 2, the optimal solution is unique and satisfies the first-order critical point condition:

$$\nabla_{\boldsymbol{h}} E(\boldsymbol{h}, \boldsymbol{\nu}) = \boldsymbol{0}. \tag{12}$$

Directly computing the gradient of the energy functional:

$$\frac{\partial E}{\partial h_i} = \frac{\partial}{\partial h_i} \int_{\Omega} \max_{j \in \mathcal{I}} \left( \langle \boldsymbol{x}, \boldsymbol{y}_j \rangle + h_j \right) d\mu(\boldsymbol{x}) - \nu_i. \tag{13}$$

By the maximum property of Power-Voronoi cells:

$$\frac{\partial}{\partial h_i} \max_{j \in \mathcal{I}} \left( \langle \boldsymbol{x}, \boldsymbol{y}_j \rangle + h_j \right) = \mathbb{I}_{W_i(\boldsymbol{h})}(\boldsymbol{x}), \tag{14}$$

where $\mathbb{I}_{W_i}$ denotes the indicator function of the set $W_i$. Thus:

$$\frac{E}{\partial h_i} = \int_{\Omega} \mathbb{I}_{W_i(\boldsymbol{h})}(\boldsymbol{x}) d\mu(\boldsymbol{x}) - \nu_i = \mu\left(W_i(\boldsymbol{h})\right) - \nu_i. \tag{15}$$

In matrix-vector form, this becomes:

$$\boldsymbol{\Phi}(\boldsymbol{h}) - \boldsymbol{\nu} = \boldsymbol{0}. \tag{16}$$

equation 16 is the core equality for the optimality of SDOT. We define the implicit function:

$$\boldsymbol{F}(\boldsymbol{h}, \boldsymbol{\nu}) = \boldsymbol{\Phi}(\boldsymbol{h}) - \boldsymbol{\nu}. \tag{17}$$

The optimal solution satisfies $\boldsymbol{F}(\boldsymbol{h}, \boldsymbol{\nu}) = \boldsymbol{0}$. For the two sets of measures, we have:

$$\begin{cases} \boldsymbol{\Phi}(\boldsymbol{h}') - \boldsymbol{\nu}' = \boldsymbol{0}, \\ \boldsymbol{\Phi}(\boldsymbol{h}'') - \boldsymbol{\nu}'' = \boldsymbol{0}. \end{cases} \tag{18}$$

Subtracting these two equations gives:

$$\boldsymbol{\Phi}(\boldsymbol{h}') - \boldsymbol{\Phi}(\boldsymbol{h}'') = \boldsymbol{\nu}' - \boldsymbol{\nu}''. \tag{19}$$

By Lemma 3, $\boldsymbol{\Phi}$ is differentiable. Using the mean value theorem for multivariate functions:

$$\boldsymbol{\Phi}(\boldsymbol{h}') - \boldsymbol{\Phi}(\boldsymbol{h}'') = \mathbf{J}_{\boldsymbol{\Phi}}(\bar{\boldsymbol{h}}) \cdot (\boldsymbol{h}' - \boldsymbol{h}''), \tag{20}$$

where $\bar{\boldsymbol{h}}$ is some intermediate point on the line segment connecting $\boldsymbol{h}'$ and $\boldsymbol{h}''$. Substituting into equation 19:

$$\mathbf{J}_{\boldsymbol{\Phi}}(\bar{\boldsymbol{h}}) \cdot (\boldsymbol{h}' - \boldsymbol{h}'') = \boldsymbol{\nu}' - \boldsymbol{\nu}''. \tag{21}$$

By Lemma 3, there exists a uniform positive lower bound $\lambda > 0$ such that $\mathbf{J}_{\mathbf{\Phi}} \succeq \lambda \mathbf{I}$. Thus, the Jacobian matrix $\mathbf{J}_{\mathbf{\Phi}}$ is invertible, and its operator norm satisfies:

$$\|\mathbf{J}_{\mathbf{\Phi}}^{-1}\|_2 \leq \frac{1}{\lambda}. \tag{22}$$

Left-multiplying both sides of equation 21 by the inverse Jacobian matrix:

$$\boldsymbol{h}' - \boldsymbol{h}'' = \mathbf{J}_{\mathbf{\Phi}}(\bar{\boldsymbol{h}})^{-1}(\boldsymbol{\nu}' - \boldsymbol{\nu}''). \tag{23}$$

Taking the $\ell_2$-norm on both sides:

$$\|\boldsymbol{h}' - \boldsymbol{h}''\|_2 \leq \|\mathbf{J}_{\mathbf{\Phi}}(\bar{\boldsymbol{h}})^{-1}\|_2 \cdot \|\boldsymbol{\nu}' - \boldsymbol{\nu}''\|_2. \tag{24}$$

Substituting the positive definite constant estimate:

$$\|\boldsymbol{h}' - \boldsymbol{h}''\|_2 \leq \frac{1}{\lambda} \cdot \|\boldsymbol{\nu}' - \boldsymbol{\nu}''\|_2. \tag{25}$$

Let the global constant be $C = \frac{1}{\lambda}$, then we obtain:

$$\|\boldsymbol{h}' - \boldsymbol{h}''\|_2 \leq C \cdot \|\boldsymbol{\nu}' - \boldsymbol{\nu}''\|_2. \tag{26}$$

□

### C.4 The Proof of Theorem 1

*Proof.* We present the proof for the case of single-point addition. The addition of the entire point set $\mathcal{Y}_k$ can be regarded as a composite operation consisting of successive single-point additions.

We add a single point $\boldsymbol{y}_j^k$ with measure $\nu_j^k$ to the target set $\mathcal{Y}_{current}$. Since point addition acts as the inverse operation of point removal, this process is equivalent to gradually increasing the measure of $\boldsymbol{y}_j^k$ from zero to $\nu_j^k$. Let

$$\boldsymbol{\nu}_{before} = [\nu_1, \nu_2, \cdots, \nu_{|\mathcal{I}_{current}|}, 0]^\top, sum(\boldsymbol{\nu}_{before}) = 1 \tag{27}$$

and

$$\boldsymbol{\nu}_{after} = \frac{1}{1+\nu_j^k}[\nu_1, \nu_2, \cdots, \nu_{|\mathcal{I}_{current}|}, \nu_j^k]^\top \tag{28}$$

represent the measure distributions before and after the addition operation respectively, we have

$$\begin{aligned}
\boldsymbol{\nu}_{after} - \boldsymbol{\nu}_{before} &= \left[\frac{1}{1+\nu_j^k}\nu_1 - \nu_1, \cdots, \frac{1}{1+\nu_j^k}\nu_{|\mathcal{I}_{current}|} - \nu_{|\mathcal{I}_{current}|}, \frac{\nu_j^k}{1+\nu_j^k}\right]^\top \\
&= -\frac{\nu_j^k}{1+\nu_j^k}[\nu_1, \cdots, \nu_{|\mathcal{I}_{current}|}, -1]^\top \\
&= -\frac{\nu_j^k}{1+\nu_j^k}(\boldsymbol{\nu}_{current}, -1)^\top.
\end{aligned} \tag{29}$$

By combining Lemma 4, we obtain

$$|h_\lambda^{k,*} - h_\lambda| \leq \|\boldsymbol{h}^{k,*} - \boldsymbol{h}^{k-1,*}\|_2 \leq \|\boldsymbol{\nu}_{after} - \boldsymbol{\nu}_{before}\|_2 \leq C_1 \cdot \frac{\nu_j^k}{1+\nu_j^k}, \tag{30}$$

where $C_1 = \|(\boldsymbol{\nu}_{current}, -1)\|_2 \leq \sqrt{sum(\boldsymbol{\nu}_{current})^2 + 1} = \sqrt{2}$. Finally, we apply the triangle inequality and substitute the results from Lemma 1 and equation 30

$$\begin{aligned}
|h_j^{k,*} - h^{k,init}| =& |h_j^{k,*} - h_\lambda| \\
\leq& |h_j^{k,*} - h_\lambda^{k,*}| + |h_\lambda^{k,*} - h_\lambda| \\
\leq& \operatorname{norm}(\Omega) \cdot \|\boldsymbol{y}_j^k - \boldsymbol{y}_\lambda\|_2 + C_1 \cdot \frac{\nu_j^k}{1+\nu_j^k} \\
\leq& \operatorname{norm}(\Omega) \cdot \|\boldsymbol{y}_j^k - \boldsymbol{y}_\lambda\|_2 + \sqrt{2} \cdot \frac{\nu_j^k}{1+\nu_j^k},
\end{aligned} \tag{31}$$

which completes the proof. □

## D Datasets, Baselines and Metrics for Downstream Tasks

**Datasets, Baselines and Metrics.** (1) Image Generation. We take the method in (Li et al., 2025a) as the baseline, and conduct experiments on Cifar10 (Krizhevsky et al., 2009), CelebA (Liu et al., 2015), CelebA-HQ (Karras et al., 2017) and LSUN-bedroom (Yu et al., 2015). Fréchet Inception Distance (FID) (Heusel et al., 2017) is used to measure generation quality. (2) 3D Point Cloud Generation. We take PointOT (Zhang et al., 2022) as the baseline, and test on 4- and 8-class targets from ShapeNet (Wu et al., 2015). Chamfer Distance (CD) (Fan et al., 2017) and Earth Mover's Distance (EMD) (Achlioptas et al., 2018) are adopted as evaluation metrics. (3) Adversarial Attack. We choose classifiers from the state-of-the-art (SOTA) NoPain (Li et al., 2025b), including PointNet (Qi et al., 2017), PointConv (Wu et al., 2019), DGCNN (Wang et al., 2019), and PCT (Guo et al., 2021). Based on pre-trained models from (Zhang et al., 2024), we evaluate on ModelNet40 (Wu et al., 2015). Attack Success Rate (ASR) measures attack performance, and CD quantifies perturbation magnitude of adversarial samples. (4) OOD Overconfidence. We take the SOTA method OTIS (Tang et al., 2026) as the baseline. ImageNet serves as the in-distribution (ID) dataset, ResNet-50 as the evaluated model, and Places (Zhou et al., 2016), iNaturalist (Van Horn et al., 2018), Textu-res (Ashworth & McLellan, 1985) as OOD bench-mark datasets. Following the protocol in (Meinke & Hein, 2019), we calculate Mean Maximum Confidence (MMC) on ID and OOD datasets respectively. We also report False Positive Rate at 95% True Positive Rate (FPR95) to evaluate overconfidence suppression for unseen inputs, where confidence serves as the scoring function for ID/OOD discrimination.

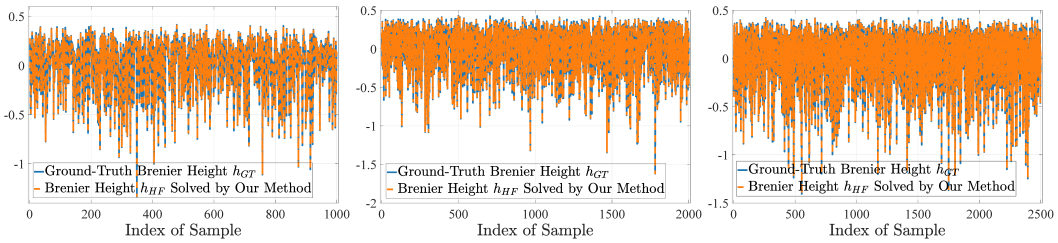
## E More Experimental Results

We conduct comprehensive supplementary experiments under the exact same experimental settings and hyperparameters as those in the main text to further validate the effectiveness, stability, and generality of the proposed hierarchical algorithm. Specifically, Fig. 4 compares the Brenier heights solved by our hierarchical method with the ground truth under various data distribution scenarios, which intuitively verifies the accuracy of our solution. Fig. 5 further characterizes the error distribution of the estimated Brenier heights relative to the ground truth under different target distributions, clearly revealing the error convergence pattern and quantitative accuracy range of the proposed method.

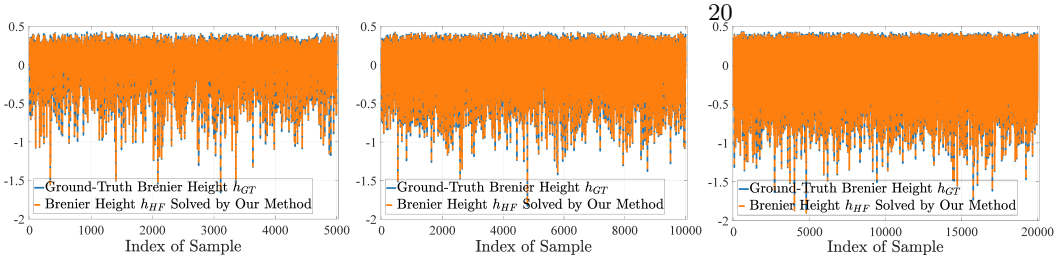
On this basis, two groups of comparative experiments are performed to investigate the influence of core hyperparameters on the acceleration performance. With the hierarchical ratio fixed as $r = 0.2$, Fig. 6 explores the effect of the target sample number N on the speed-up ratio relative to the baseline under various distribution scenarios and provides quantitative evaluations over multiple error metrics , which demonstrates the superior efficiency and stable accuracy of our algorithm with varying sample scales. Corresponding quantitative evaluations on diverse error metrics further clarify the optimal parameter range of the hierarchical ratio.

Fig. 7 further analyzes the convergence performance of the proposed method under different error metrics and application scenarios, verifying its stable and fast convergence property. In addition, the subcaption of each subgraph explicitly presents the key experimental configurations, including the distribution type, sample size, data dimension, and hierarchical ratio, ensuring the reproducibility and rigor of the experimental results.

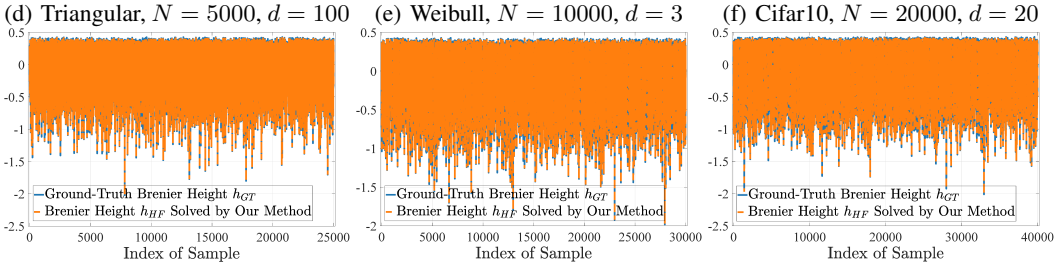
Overall, the supplementary experimental results demonstrate that the proposed hierarchical acceleration strategy achieves prominent performance improvements. The estimation error is stably controlled at the permille level, which guarantees high solution accuracy. Meanwhile, the proposed method significantly reduces the computational time and improves the solving efficiency. Extensive experiments validate the excellent robustness and universal applicability of our algorithm across diverse distributions, sample scales, and parameter settings.



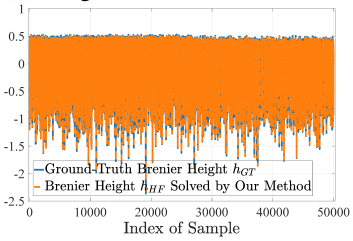

(a) Gaussian, $N = 1000$, $d = 5$ (b) Von-Mises, $N = 2000$, $d = 10$ (c) Exponential, $N = 2500$, $d = 20$




(d) Triangular, $N = 5000$, $d = 100$ (e) Weibull, $N = 10000$, $d = 3$ (f) Cifar10, $N = 20000$, $d = 20$

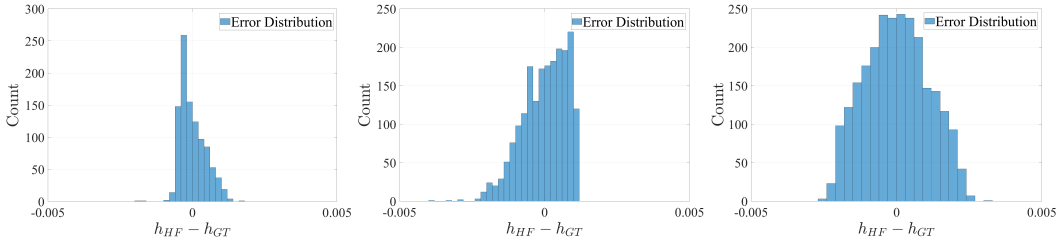



(g) CelebA, $N = 25000$, $d = 50$ (h) ImageNet, $N = 30000$, $d = 50$ (i) Gaussian, $N = 40000$, $d = 20$

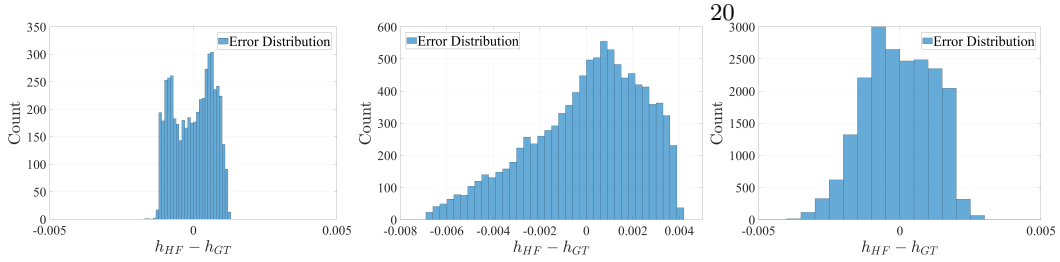



(j) Von-Mises, $N = 50000$, $d = 20$

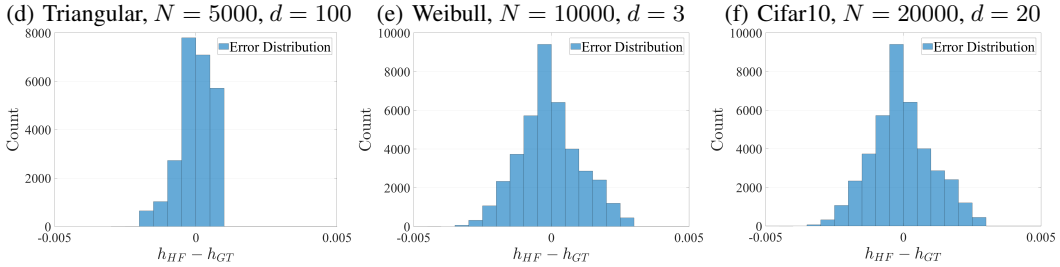
Fig. 4: Comparison of Brenier Heights over Various $N$. $r$ is fixed at 0.2.



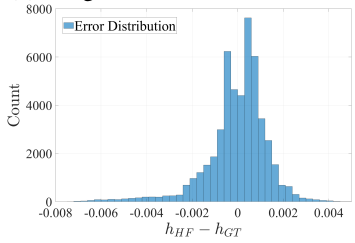

(a) Gaussian, $N = 1000$, $d = 5$ (b) Von-Mises, $N = 2000$, $d = 10$ (c) Exponential, $N = 2500$, $d = 20$




(d) Triangular, $N = 5000$, $d = 100$ (e) Weibull, $N = 10000$, $d = 3$ (f) Cifar10, $N = 20000$, $d = 20$

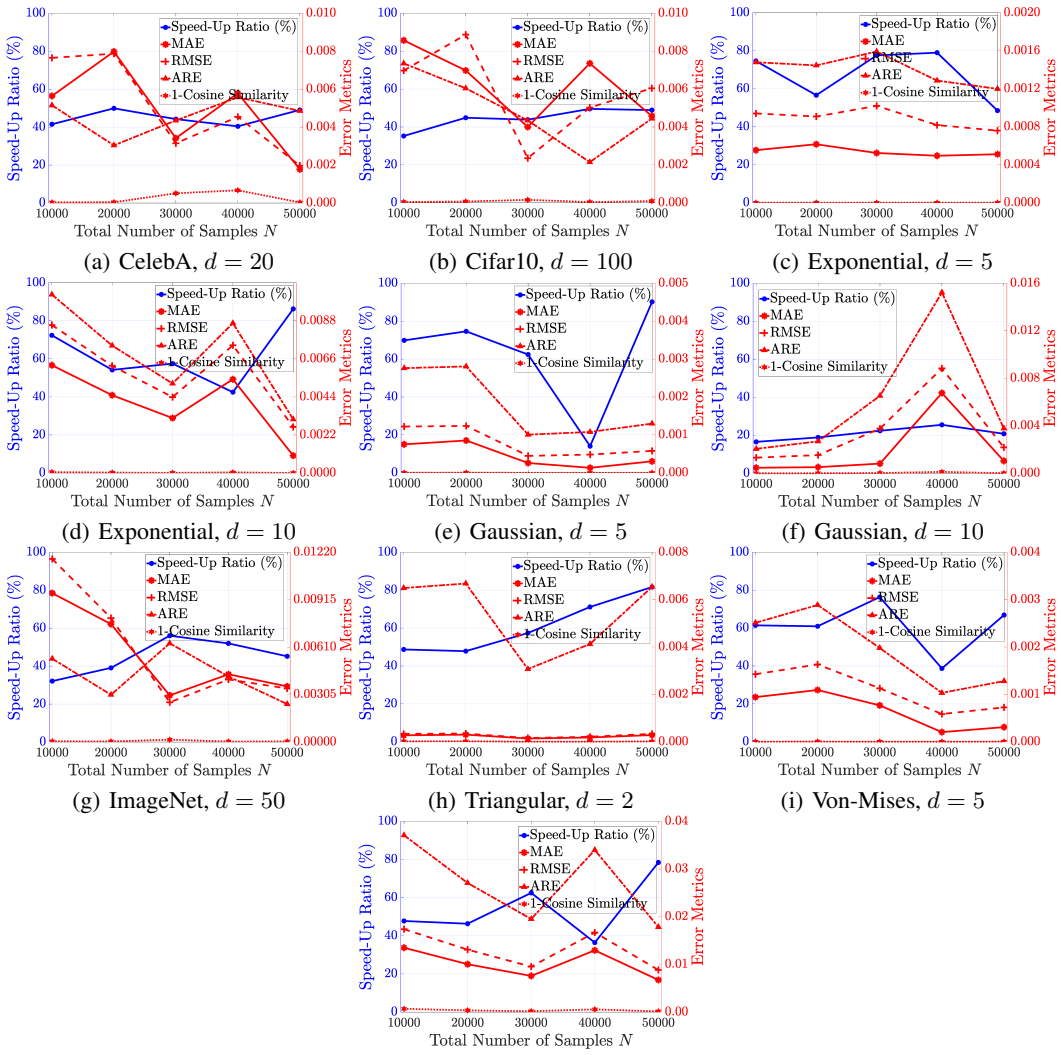



(g) CelebA, $N = 25000$, $d = 50$ (h) ImageNet, $N = 30000$, $d = 50$ (i) Gaussian, $N = 40000$, $d = 20$




(j) Von-Mises, $N = 50000$, $d = 20$

Fig. 5: Error Distribution of Brenier Heights over Various $N$. $r$ is fixed at 0.2.



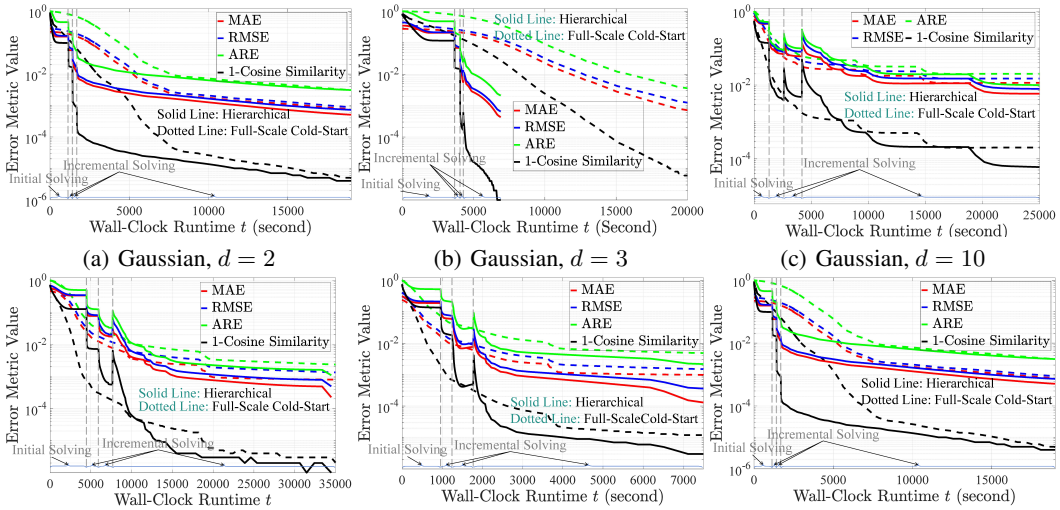

(a) CelebA, $d = 20$ (b) Cifar10, $d = 100$ (c) Exponential, $d = 5$

(d) Exponential, $d = 10$ (e) Gaussian, $d = 5$ (f) Gaussian, $d = 10$

(g) ImageNet, $d = 50$ (h) Triangular, $d = 2$ (i) Von-Mises, $d = 5$

(j) Weibull, $d = 20$

Fig. 6: Computational Efficiency and Error Evaluation of Our Method over Various $N$. $r$ is fixed at 0.2.




(a) Gaussian, $d = 2$ (b) Gaussian, $d = 3$ (c) Gaussian, $d = 10$

(d) Von-Mises, $d = 5$ (e) Weibull, $d = 10$ (f) Weibull, $d = 2$

Fig. 7: Convergence Curves Versus RunTime Under Different Error Metrics. $N = 50000$, $r = 0.2$.