A Computation-Aware Shape Loss Function for Point Cloud Completion

Shunran Zhang[1, 2]*, Xiubo Zhang[1]*, Tsz Nam Chan[3], Shenghui Zhang[1], Leong Hou U[1]†

[1]University of Macau
[2]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[3]Shenzhen University
{yc27968, mc25209}@um.edu.mo, edisonchan@szu.edu.cn, {yc07428, ryanlhu}@um.edu.mo

Abstract

Learning-based point cloud completion tasks have shown potential in various critical tasks, such as object detection, classification, and registration. However, accurately and efficiently quantifying the shape error between the predicted point clouds generated by networks and the ground truth remains challenging. While EMD-based loss functions excel in shape detail and perceived density distribution, their approach can only yield results with significant discrepancies from the actual EMD within a tolerable training time. To address these challenges, we first propose an *initial price* based on an auction algorithm, reducing the number of iterations required for the algorithm while ensuring the correctness of the assignment results. We then introduce a method to compute the initial price through a successive shortest path and the Euclidean information between its nodes. Finally, we adopt a series of optimization strategies to speed up the algorithm and offer an EMD approximation scheme for point cloud problems that balances time loss and computational accuracy based on point cloud data characteristics. Our experimental results confirm that our algorithm achieves the smallest gap with the real EMD within an acceptable time range and yields the best results in end-to-end training.

Introduction

Understanding the environment is a fundamental necessity across various domains, ranging from autonomous driving and intelligent transportation systems to robotics and mixed reality applications. The success of these cutting-edge technologies heavily depends on the capacity of a system to interact with and accurately perceive its surroundings. In this regard, point clouds have emerged as a pivotal element in environmental sensing, and their generation can be facilitated by diverse technologies like LiDAR (Light Detection and Ranging) (Reutebuch, Andersen, and McGaughey 2005). Point clouds provide a rich representation of the environment and can capture detailed geometrical and spatial information, making them invaluable in various tasks related to perception, localization, mapping, and object recognition. As a result, point cloud data has become a fundamental input for many state-of-the-art algorithms and systems in modern sensing and navigation applications.

However, one challenge of using LiDAR-generated spatial point clouds is that they may fail to fully depict the surface of the observed object due to occlusions or the inability to arrange sensors at sufficient angles. As a result, the captured point cloud data may only provide a partial representation of the object. This limitation can have a significant impact on subsequent point cloud processing tasks, such as point cloud registration, object detection, and classification.

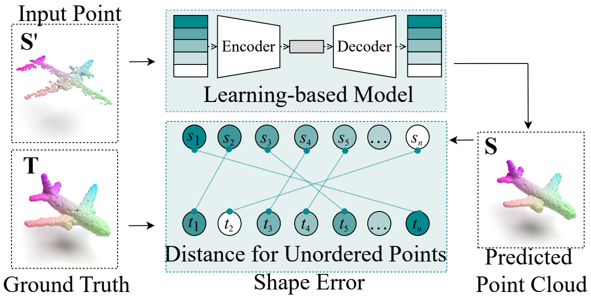
Input Point
S'
Encoder
Decoder
Learning-based Model
s_1 s_2 s_3 s_4 s_5 … s_n
t_1 t_2 t_3 t_4 t_5 … t_n
S
T
Distance for Unordered Points
Shape Error
Ground Truth
Predicted Point Cloud

Figure 1: The illustration of point cloud completion task.

Numerous studies (Achlioptas et al. 2018; Yuan et al. 2018; Sarmad, Lee, and Kim 2019) have successfully showcased the efficacy of employing learning-based techniques for 3D shape completion, offering significant advantages for subsequent processing tasks. As depicted in Figure 1, dealing with irregular and unordered point clouds introduces substantial challenges in this context. Accurately quantifying the distance between point cloud pairs, accounting for shape discrepancies in the output loss of a deep learning model concerning the ground truth, and facilitating the update of the model parameters pose significant hurdles due to the inherent complexities of point cloud data.

Fan et al. (Fan, Su, and Guibas 2017) introduced two permutation invariant functions, namely Chamfer Distance (CD) and Earth Mover's Distance (EMD). While CD is efficient, it does not penalize uneven distributions, leading to shape detail fluctuations (Achlioptas et al. 2018). EMD is more sensitive to shape detail and density distribution differences (Liu et al. 2020), which can achieve more accurate comparison between a pair of point clouds. However,

*These authors contributed equally.

†Corresponding author.

EMD is computationally expensive, which is not scalable to be used in deep learning models (with large number of iterations) for supporting large data. As such, researchers often opt for Chamfer Distance (CD) over Earth Mover's Distance (EMD), which can significantly degrade the quality of point cloud completion.

To address the computational challenges of EMD, Fan et al. (Fan, Su, and Guibas 2017) develop an iterative $(1+\epsilon)$-approximation method. However, this method still cannot be scalable to large point cloud sizes. Liu et al. (Liu et al. 2020) propose an alternative approach, which can handle larger point clouds and only takes $O(n)$ memory.

However, our experiments (cf. Table 1, where two existing methods are denoted as emd_1 and emd_2.) show both methods significantly deviate from the actual EMD, leading to estimation errors and increased gradient update noise in the training process.

MSE	emd_1	emd_2	Ours
Sparse	1388.26	912.78	**121.79**
Dense	1014.70	1131.23	**408.13**

Table 1: The mean squared error (MSE) of different methods for estimating EMD. Experiments were conducted on "Sparse" point clouds (1024 points) and "Dense" point clouds (8192 points), calculating the MSE between the computed results and the actual EMD. Here, two widely used EMD loss functions (Fan, Su, and Guibas 2017; Liu et al. 2020) are denoted as emd_1 and emd_2.

To tackle the challenge of accurate EMD approximation within a reasonable time frame, we propose the ***A**daptive **A**uction with **I**ntial **P**rice Algorithm* (AAIP)[1]. This approach is built on the auction algorithm, using successive shortest path principles to redefine distance relationships between specific spatial points. We have proven that initializing the auction algorithm with these initial prices converges to the assignment outcomes of the original algorithm and reduces the number of iterations in the training process.

Our principal contributions can be succinctly encapsulated as follows.

- We propose a novel concept, the Initial Prices of auction algorithm, to accelerate the convergence of the algorithm while ensuring the correctness of the results. We theoretically demonstrate its correctness and effectiveness.
- We propose an efficient algorithm for computing the Initial Prices. By utilizing initial prices and the data features, we introduce a novel adaptive EMD approximation scheme for shape loss function.
- We conduct sufficient experiments on point cloud data of various categories and sizes. The experimental results show that the proposed approach can effectively reduce the error in the shape loss function, which can further achieve the best training results compared with the existing methods.

Related Work

Point cloud completion via learning-based methods. Over recent years, many deep learning methods have been developed for the point cloud completion problem, which can be further categorized into two camps, namely (1) voxel-based methods and (2) point-based methods.

Voxel-based methods (Dai, Qi, and Nießner 2017; Han et al. 2017; Sharma, Grau, and Fritz 2016; Stutz and Geiger 2018; Liu et al. 2019a,b) first represent each point cloud as a set of voxels and then train the learning-based models for these point cloud data. However, it is hard to accurately tune the voxel size, which can either result in huge computational costs (and huge memory space consumption) or low accuracy during the learning process. Unlike the voxel-based methods, point-based methods (Qi et al. 2017; Li et al. 2018; Yuan et al. 2018; Sarmad, Lee, and Kim 2019; Tchapmi et al. 2019; Chen, Chen, and Mitra 2020) adopt different loss functions for directly measuring the distance between data points in a pair of point clouds in order to train the learning-based models, which, to the best of our knowledge, can achieve better accuracy for this point cloud completion problem compared with the voxel-based methods.

Loss function for point cloud completion. There are two types of commonly used loss functions (Fan, Su, and Guibas 2017) for point-based methods, which are Chamfer Distance (CD) and Earth Mover's Distance (EMD). Compared with EMD, CD is unable to penalize errors in shape details and is insensitive to differences in density distribution (Yuan et al. 2018; Achlioptas et al. 2018; Liu et al. 2020). Therefore, using EMD as the loss function in a learning-based model can provide more accurate results for the point cloud completion problem. However, EMD suffers from huge computational cost, which is not scalable to large-scale point cloud data.

Although numerous approximation methods have been proposed to boost the efficiency of computing EMD for the image retrieval tasks (Jang et al. 2011; Cuturi 2013; Solomon et al. 2015; Altschuler, Weed, and Rigollet 2017; Chan, Yiu et al. 2019), using these methods for the point cloud completion problem needs to construct the cost matrix (with quadratic computational cost) between each pair of point clouds. Hence, these methods still suffer from huge computational burden (especially for the point cloud with large number of data points).

Recently, Fan et al. (Fan, Su, and Guibas 2017) have directly proposed the EMD approximation scheme for point clouds, which has been widely adopted in many learning-based point cloud completion networks (Yuan et al. 2018; Chen, Chen, and Mitra 2019; Wu, Miao, and Fu 2021; Chang, Jung, and Xu 2021). However, this approach still takes $O(n^2)$ memory space, limiting its applicability to large point cloud datasets. Later, Liu et al. (Liu et al. 2020) further propose the auction-based (Bertsekas 1979) loss function estimation method (with $O(n)$ memory space), which is scalable to large-scale point cloud datasets. However, compared with our methods, all these approximation methods are still not accurate for computing EMD (cf. Table 1).

[1] The code is available at https://github.com/coldbubbletea/AAIP-Point-Cloud-Completion.

Preliminary

Point cloud completion problem statement. As shown in Figure 1, consider $\mathbf{S}^{'}$ as an ensemble of spatial points, situated on the visibly observed facets of an object, acquired through singular or sequential observations via LiDAR. Concurrently, envisage $\mathbf{T}$ as a densely populated collection of spatial points, uniformly distributed across both the observed and unseen facets of the said entity. In this context, the shape completion problem is framed as the prediction of $\mathbf{T}$ using learning based methods, given $\mathbf{S}^{'}$ as input.

Loss function of shape error. It is key to note that $\mathbf{S}^{'}$ is not necessarily contained within $\mathbf{T}$ due to independent sampling. This absence of direct point-to-point correspondence between $\mathbf{S}^{'}$ and $\mathbf{T}$ means that the chosen metric in the loss function computation of shape error, measuring the distance between $\mathbf{S}$ and $\mathbf{T}$, should exhibit robust permutation invariance and accurately reflect shape error. This choice directly impacts evaluative efficacy and, consequently, the quality of point cloud prediction.

Fan et al. (Fan, Su, and Guibas 2017) utilized Earth Mover's Distance (EMD) as a loss function, which demonstrates greater robustness compared to Chamfer Distance (CD), for point cloud completion problems.

The formulation of EMD in the context of point cloud distance measurement is shown as follows.

$$\mathrm{EMD}\left(\mathbf{S},\mathbf{T}\right)=\min_{\phi:\mathbf{S}\rightarrow\mathbf{T}}\frac{1}{|\mathbf{S}|}\sum_{x\in\mathbf{S}}\|x-\phi(x)\|_2 \tag{1}$$

where $\phi(x)$ denotes a bijection that establishes a one-to-one mapping relationship between point clouds $\mathbf{S}$ and $\mathbf{T}$, minimizing the average distance of corresponding points.

Auction algorithm on point cloud. The auction algorithm (Bertsekas 1985) is contemplated as an elegant solution for computing the bijection amidst points within dual point clouds because parallel computation is straightforwardly achievable, rendering the procedure apt for calculating EMD as loss function in deep learning model.

The auction algorithm treats the two point clouds as *sources* $\mathbf{S}$ and *sinks* $\mathbf{T}$. We will illustrate the algorithmic process using the example in Figure 2 and display it in Table 2. For each independent source $s_i \in \mathbf{S}$ and sink $t_j \in \mathbf{T}$ pairing, there exists a g_{ij}, which represents the source's evaluation of the sink's attractiveness, expressed as a formula as

$$g_{ij}=C-d_{ij}, \tag{2}$$

where C is a constant here to ensure positive value, and d_{ij} represents the Euclidean Distance between s_i and t_j. Assume $C=20$ in the example. In the auction algorithm, each sink in the graph is given a numerical quantity, often denoted as μ_j, known as the price, which represents the *cost* of assigning that particular sink. Initially, the price for every sink is usually set to 0.

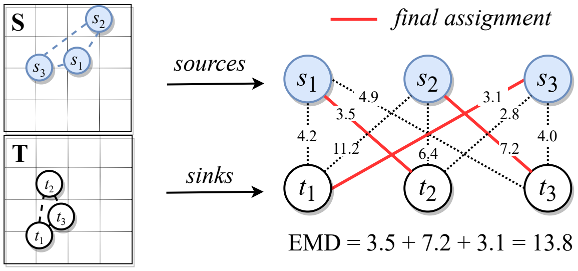

Figure 2: An example illustrates the efficacy of the method. Two point clouds are regarded as sources and sinks, with the objective of finding an one-to-one mapping of three-dimensional points that satisfy optimization conditions. Finally, the global minimum distance EMD is determined.

During each round, as illustrated in Table 2, each source s_i identifies a sink t_j that maximizes $(g_{ij}-\mu_j)$, to choose which sink is the optimal choice. As an example, s_1, s_2, s_3 place their bids for t_2 in the first round. If the sink t_j is presently unassigned or if s_i offers a higher bidding price than the other sources bidding on t_j, then s_i is assigned to t_j, and the price μ_j is updated to the bidding price of s_i accordingly. The bidding price denoted as β for a source is established by appending a *bid increment* to the existing price μ_j of a sink t_j. This bid increment lies within the range of $[\epsilon, \pi+\epsilon]$, where ϵ is a pre-established relaxation parameter, and π represents the profit discrepancy between the *optimal* and *suboptimal* choices for s_i in its present state. However, if s_i does not offer the highest bidding price, it must seek another sink in the following round. In the example, the bid increment attains its maximum value $(\pi+\epsilon)$, with $\epsilon=0.01$. As s_2 presents the highest bidding price 0.81 ($\beta_{12}=0.71, \beta_{22}=0.81, \beta_{32}=0.31$), t_2 is ultimately assigned to s_2, with its price updated to 0.81. This process is repeated until each source is assigned a sink. As shown in Table 2, the algorithm proceeds for three more rounds as outlined in the table, culminating in a one-to-one mapping, denoted by the red lines in Figure 2. The pseudocode for the specific algorithm is provided in the Appendix.

Round	Bid			Price Update		
	s_1	s_2	s_3	t_1	t_2	t_3
Init.	-	-	-	0	0	0
1	t_2: 0.71	t_2: 0.81	t_2: 0.31	0	0.81	0
2	t_1: 0.71	-	t_1: 0.91	0.91	0.81	0
3	t_2: 1.41	-	-	0.91	1.41	0
4	-	t_3: 0.62	-	0.91	1.41	0.62

Table 2: The bidding and price update process of the auction algorithm across different rounds.

Although the auction algorithm has been widely used as the fundamental algorithm for solving the EMD in point cloud problems due to its outstanding parallel capabilities, its application still presents challenges. Firstly, its heuristic nature results in significantly varying time costs for different data characteristics, which are difficult to predict. Moreover, for some point cloud data, the algorithm requires a large number of iterations before it can stop. These factors are detrimental to the iterative computation process in deep learning with large dataset.

Methodology

Motivating Example

In the example provided in Figure 2, the auction algorithm requires at least *four rounds* to complete the calculation for optimal assignment. However, if we set the initial prices as $\mu_1 = 0.80, \mu_2 = 1.00, \mu_3 = 0$, it becomes apparent in Table 3 that merely *a single auction round* is necessary to finalize the assignment.

Round	Bid			Price Update		
	s_1	s_2	s_3	t_1	t_2	t_3
Init.	-	-	-	0.80	1.00	0
1	t_2: 1.41	t_3: 0.21	t_1: 0.91	0.91	1.41	0.21

Table 3: The bidding and price update process of the auction algorithm with initializing the price.

As shown in this example, the auction iteration is due to different source points selecting the same sink during the process, concurrently boosting the price of the sink points. The essence of the auction algorithm is to iteratively heighten the cost of selecting a sink, thereby reducing the number of sources interested in that sink. This continues until all unsuitable sources (those with better options) are eliminated, achieving an one-to-one assignment. *Thus, can we adopt certain strategies to initialize prices in the auction algorithm, ensuring the same results but with a quicker elimination of unsuitable sources*?

Auction Algorithm with Initial Prices

In the widely used auction algorithm, the price of each sink is often initialized at 0. Setting prices incorrectly could potentially deteriorate the assignment, thus we propose a guaranteed initial price that ensures the final optimized assignment $\mathcal{A}'$ remains unaffected and accelerates the convergence of the algorithm.

Correctness of initial prices. Yet, we discern that given a local assignment $\mathcal{A}^-$ of $\mathbf{S}^- \subseteq \mathbf{S}$ and $\mathbf{T}^- \subseteq \mathbf{T}$, denoting the final *selling price* of the assigned $t_j \in \mathbf{T}^-$ in $\mathcal{A}^-$ as μ_j^-. We propose the *initial price*, which is described in Definition 1.

Definition 1 (Initial prices, Δ)**.** *We say the set of initial prices Δ is valid if* $0 \leq \delta_j \leq \mu_j^-, \forall t_j \in \mathbf{T}^-$.

When each initial price $\delta_j \in \Delta$ is employed to initialize the corresponding sink price μ_j for each sink t_j in the auction algorithm, we demonstrate that the final assignment remains correct. This is encapsulated in our proposed Theorem 1.

Theorem 1 (Correctness of initial prices)**.** *By initializing the price of the corresponding sink with the initial prices Δ, the final assignment $\mathcal{A}'$ is equivalent to the result $\mathcal{A}^*$ of the auction algorithm without using initial prices for price initialization, that is, $\mathcal{A}' \equiv \mathcal{A}^*$.*

The proof of Theorem 1 is provided in Appendix. In simple terms, the auction algorithm ensures that the selling price of a sink in a local optimal assignment does not exceed its selling price in a globally solved problem. This implies that even if we initialize the initial price δ_j for t_j at a value less than or equal to μ_j^-, it will eventually be procured by a suitable source at a higher (correct) selling price by the auction process. This maintains the consistency of the assignment.

The introduction of Theorem 1 provides a solvable upper bound when we need to initialize the price of the sink. As long as we do not exceed this upper bound when assigning the price, we can ensure that the results remain unchanged.

Effectiveness of initial prices. Furthermore, the initial price strategy can expedite the convergence speed of the auction algorithm towards its final result by causing unsuitable sources to cease their bidding for a specific $sink$ prematurely. This is stated in our proposed Lemma 1.

Lemma 1. *The initial price can reduce the upper bound of the number of iterations required by the Auction Algorithm.*

For proof details, please refer to the Appendix. As demonstrated in Table 3, the initialization of prices leads to a scenario where the unsuitable source s_2 cannot bid for t_2 in the first round of auction, and instead directly selects the optimal choice t_3. This results in the algorithm completing the assignment within a single iteration. Clearly, the core of the initial prices is to reconstruct the relationship between sources and sinks in the initial state, which reduces the number of unsuitable sources, thereby accelerating the convergence speed. Thus, based on Lemma 1, we prove that the proposed initial price, when used to initialize prices, indeed accelerates the convergence of the auction algorithm.

Configuration of Initial Prices, Δ

However, this approach presents practical challenges. Since the upper bound, μ_j^-, for the initial price δ_j for t_j must be obtained through the auction algorithm to find a local assignment, considering the unpredictability of the computational cost associated with solving such problem using the auction algorithm (as per Definition 1), it is possible that the overall computational cost of (1) determining the initial price Δ and (2) solving the final assignment $\mathcal{A}^{'}$ using the initial price Δ could surpass that of the original Auction Algorithm.

In this endeavor, our aim is to configure the initial prices in a manner that fulfills the following requirements: (1) ensuring the fulfillment of Theorem 1, which states that for every task t_j in the set $\mathbf{T}$, δ_j must be less than or equal to μ_j^-, and (2) providing a solution cost that can be controlled, thereby reducing the overall computational burden.

Initial prices derived from a local assignment. To begin, let us delve into an initial price that not only adheres to the conditions of Theorem 1, but also can be computed using the result of a local assignment. Let $\mathbf{A}^-$ be the local assignment and d_{ij} denote the Euclidean distance between point $s_i \in \mathbf{S}$ and point $t_j \in \mathbf{T}$. The specific content is presented in Lemma 2.

Lemma 2. *Assuming that (s_i, t_j) is an assigned pair in a local assignment $\mathbf{A}^-$. Furthermore, within this assignment, $\forall s_p \in (\mathbf{S}^- - s_i)$, the sink assigned to s_p is denoted as $t_q \in \mathbf{T}^-$. Considering the value α_j obtained by evaluating* $\max_{s_p \in \mathbf{S}^-} \{d_{pq} - d_{pj} + \epsilon\}$*, we can set* $\delta_j = \max\{0, \alpha_j\}$.

This choice ensures that δ_j satisfies Theorem 1, specifically $\delta_j \leq \mu_j^-$.

We leave the proof in Appendix. As per the insights provided by Lemma 2, we have the flexibility to relax the value of δ_j by reducing it from the price upper bound μ_j^- (from the Auction algorithm) to $\max\{0, \alpha_j\}$ (from the local assignment). Remarkably, this adjustment preserves the assurance that the final assignment remains in alignment with the original auction algorithm. This implies that we merely necessitate the acquisition of a locally optimal assignment, and through the relationship of t_j, utilizing the local assignment will still yield a δ_j that satisfies Lemma 2.

Computing a local assignment. According to Lemma 2, our subsequent challenge is to efficiently determine the local assignments required by the lemma. The local assignment can be transformed into a minimum cost flow problem (MCF) in directed bipartite graphs (Waissi 1994). In the context of spatial point cloud issues, each edge e satisfies: a capacity of 1, a cost equivalent to the distance between points for $e(v_i, v_j)$. Several algorithms exist to solve the MCF problem, with the Hungarian algorithm (Kuhn 1955) and the Successive Shortest Path Algorithm (SSPA) (Derigs 1981) being the most commonly used due to their low complexity. However, the Hungarian algorithm requires to construct a cost matrix makes it unsuitable for high-density point cloud problems. SSPA, on the other hand, is more suited for this task, as it operates directly on the flow graph and iteratively computes the assignment via shortest path searches. Specifically, we utilize SSPA to obtain the local assignment $\mathbf{A}^-$ on the bipartite graph $(\mathbf{S}^-, \mathbf{T})$. The outcome $\mathbf{A}^-$ produced by SSPA successfully fulfills the prerequisites outlined in Lemma 2 for determining the initial prices. We name this calculation process as **In**itial **Pr**ices **A**lgorithm (InPrA).

In the next section, we shall delve into an even more streamlined approach to compute the local assignment, leveraging a simplified graph strategy.

Optimization Strategies

In this section, we will explore advanced optimization techniques for algorithms, enhancing computational speed, strengthening the adaptability of algorithms as loss function solvers, and finalizing the algorithm design.

Simplified graph strategy. We employ SSPA for local optimal assignment and determine the initial prices. Like many others, the time complexity grows with spatial points, complicating the shortest path identification in a directed bipartite graph and increasing computation time. The Simplified Graph Incremental Algorithm (SIA) (U et al. 2010) mitigates this complexity by preserving a novel subgraph in each loop, where edges are added until a specific condition is met. This ensures the shortest path found on the subgraph also represents a shortest path in the original graph. SIA involves a k-nearest neighbour search to establish a simplified graph, for which we use an R*-Tree (Beckmann et al. 1990) to pre-process spatial point cloud data. We believe any data structure that facilitates a k-nearest neighbour search will be appropriate for this algorithm. Upon applying SIA, the resolution of local optimization assignment $\mathbf{A}^-$ is achieved, followed by the search of $\max\{0, \alpha_j\}$ of t_j through the established shortest path tree, expediting the computation of the initial prices.

Algorithm 1: Source Sorted Algorithm (SSA)

```
Input: Spatial point cloud $\mathbf{S}$ and $\mathbf{T}$ (with size $n$)
Output: Selected point list $\mathbf{S}^-$
Initialize a $(s, t)$-pair list $H$, a point set $\mathcal{C}_D$, a point list $\mathbf{S}^-$
foreach $s_i \in \mathbf{S}$ do
    $t_{j^*} \leftarrow \arg\max_{t_j \in \mathbf{T}}\{g_{ij}\}$, push pair $(s_i, t_{j^*})$ to $H$
Sort $H$ based on $t$ in $(s, t)$-pair
for $1 \leq i \leq n$ do
    if $H(i).t = H(i-1).t$ then
        insert point $s_i$ to $\mathcal{C}_D$
$\mathbf{S}^* \leftarrow$ sort $\mathbf{S}$ with priority $s_1 > s_2$, where $s_1 \in \mathcal{C}_D$ and
    $s_2 \in (\mathbf{S} - \mathcal{C}_D)$
```

Computation order in SSPA/SIA. To meet training cost requirements, we have imposed a time constraint that may restrict the involvement of only a subset of source points $\mathbf{S}^-$. Instead of randomly selecting source points, we prioritize the inclusion of those source points that bid for the same sink points in the auction algorithm. This approach (cf. Algorithm 1) has proven beneficial, as it increases the number of sink points that receive initial prices. Our experiments have further validated this observation, reinforcing the effectiveness of this selective approach in the SSPA/SIA algorithm.

Algorithm 2: Auction with Initial Price Algorithm

```
$\mathbf{S}^* \leftarrow$ SSA($\mathbf{S}, \mathbf{T}$)
$\mathbf{S}^- \leftarrow$ getTopK($\mathbf{S}^*$)
$\mathbf{A}^- \leftarrow$ SIA($\mathbf{S}^-, \mathbf{T}$)
$\Delta \leftarrow$ InPrA($\mathbf{A}^-$)
Initialize $\mu_j$ to be $\delta_j \in \Delta$ for all $t_j \in \mathbf{T}$
$\mathcal{A}' \leftarrow$ Auction Algorithm with $\delta_j$ for all $t_j$
```

Main algorithm. Thus, we have finalized the workflow for the Auction with Initial Price algorithm as described in Algorithm 2. It sorts the sources by SSA, selects k points as $\mathbf{S}^-$, calculates local assignment $\mathbf{A}^-$ using SIA, computes the initial price through InPrA, and finally initializes the auction algorithm's price to the computed initial price. This process culminates in the final assignment.

Adaptive iteration strategy. The existing methods frequently encounter the need to abruptly terminate Earth Mover's Distance (EMD) calculations due to the heavy computational burden and extensive iterations involved in deep learning. Contrary to these methods lacking robustness, we propose an approach that adaptively adjusts the algorithm's time cost based on data characteristics.

Despite the ability to arbitrarily select $\mathbf{S}^-$ in local assignments for SIA resolution, fixing the number of assignment points is not robust in practical point cloud problems with diverse data features, potentially leading to insufficient solution points or prolonged solving time. Conversely, by restraining the SIA algorithm under certain conditions, the number of completed local assignments, or loops, can better reflect data characteristics. In our method, we set a fixed total search edge limit for the construction of the SIA subgraph. This limit is determined by the size of the point cloud data, and when reached, it signals the end of local assignment computation. As the SIA carries out more loops, it needs fewer edges on average to construct the subgraph used for the shortest path search. This efficiency translates to fewer suitable match candidates in one point cloud for a point in another, which in turn reduces the chance of encountering competitors. The end result is a decrease in the iterations required for the auction algorithm to conclude. Conversely, fewer loops necessitate increased iterations. Based on this, an *Adaptive Iteration Strategy* is proposed as follows: If the spatial point cloud terminates at the k^{th} loop when using SIA to calculate the local assignment, then the auction algorithm with initial price iteration number i satisfies

$$i = \omega|\mathbf{S}|k^{-1} + \lambda, \tag{3}$$

where i is informed by the provided scale factor ω and the ratio of the number of rounds k that the SIA can complete within the given computational limit to the size of $|\mathbf{S}|$. The symbol λ signifies the minimum number of rounds required to complete the auction. Thus, if a point cloud is readily solvable by the auction algorithm, the algorithm will correspondingly decrease the number of iterations. Conversely, if the problem is more complex, the algorithm will augment the number of iterations as much as possible. Finally, should the auction algorithm terminate due to adaptive iteration while there remain unassigned points, in accordance with MSN (Liu et al. 2020) proposed, we assign these points to the nearest counterparts in the alternate point cloud, which is deemed as their respective distances.

Integrated with the above optimization strategy, this section completes the core design and elucidation of the proposed EMD approximation algorithm for spatial point clouds. With this optimization strategy, we have expanded our Initial Price Algorithm into an Adaptive Auction Initial Price Algorithm (AAIP).

Experiments

Experimental Setup

Dataset. In this study, the ShapeNet CAD dataset (Chang et al. 2015) was used for spatial point cloud data. This dataset was chosen to ensure the fairness of our experiments, as it is used for training and testing by the models under our investigation. We conducted experiments with point clouds uniformly selected from eight categories, which include watercraft, cabinet, table, airplane, car, chair, sofa, and lamp. The complete point clouds, serving as the ground truth, were produced through uniform sampling from the model's mesh surfaces, while partial spatial point clouds were simulated using back-projected depth images (Yuan et al. 2018). Notably, these partial point clouds were collected from eight random viewpoints of each model to more closely reflect real-world conditions. Ultimately, we generated a total of 64000 pairs of point clouds for training and 9600 pairs of point clouds for testing.

Compared methods. Two predominant EMD loss function algorithms are employed in the realm of point cloud completion problems: initially, Fan et al. introduces an approximation scheme, employed as an estimation of EMD amid point cloud pairings, consequently facilitating shape loss computations (Fan, Su, and Guibas 2017). In the experimental section of this paper, such an algorithm is distinguished as "emd$_1$". Subsequently, MSN (Liu et al. 2020) presents an enhanced approximation approach predicated on the auction algorithm, necessitating solely $O(n)$ memory, and this methodology is referred to as "emd$_2$". Our elucidated EMD estimation algorithm is denoted as "AAIP".

Experimental Results from Point Cloud Completion Networks

We scrutinize the empirical outcomes of the deep learning model for point cloud completion, utilizing diverse EMD approximation strategies in an end-to-end fashion.

Backbone models. Our proposed algorithm AAIP is independently applied to two deep learning point cloud completion models, PCN (Yuan et al. 2018) and MSN (Liu et al. 2020). PCN is a two-stage point cloud generation model and its superior performance has instigated a cascade of subsequent methodologies. Many point cloud completion models have adopted the coarse-dense network architecture proposed by PCN. Due to the limitations of the EMD approximation scheme it employs, PCN uses CD/EMD as the loss function for its coarse output (1024), but only uses CD for the loss function of its dense output (16384). MSN directly generates a dense point cloud as coarse output, which is further optimized to produce the final point cloud output. Different from PCN, MSN exclusively uses EMD as the loss function and maintains 8192 points for both the coarse output and the final output. To ensure a fair comparison of model functionalities, we replaced their respective EMD approximation schemes with our proposed AAIP as the loss function, to present the experimental results.

Comparison results. The experimental results conducted on PCN and MSN are presented respectively in the Table 4. In line with the standards of other works, we employ the precise EMD as the criterion for evaluating the quality of model output. A smaller EMD signifies less shape discrepancy between the generated point cloud and the ground truth, implying a higher quality of the generated point cloud.

(a) PCN. When acting as the loss function for coarse output, the AAIP outperforms the training results using CD or emd$_1$, in any categories of training outcomes. However, for point clouds of complex categories like "lamp", due to their intricate topological structures, the CD fails to adequately penalize the differences in detail shape, resulting in a larger EMD. Moreover, adopting the AAIP as the loss function on large-scale dense output has also achieved superior experimental results within the framework of a PCN-based model.

(b) MSN. Given that the shape error calculation of MSN is entirely anchored on EMD, the overall results it produces

Methods	chair	table	sofa	cabinet	lamp	car	airplane	watercraft	average
PCN (CD+CD)	62.46	66.88	52.85	61.07	102.88	50.86	38.17	52.22	60.93
PCN (emd_1+CD)	62.83	59.94	52.53	54.91	69.50	54.12	33.22	55.30	55.29
PCN (AAIP+CD)	52.73	49.77	48.21	49.66	**62.95**	38.15	27.22	44.13	46.60
PCN (CD+AAIP)	**43.23**	**43.54**	**34.58**	**35.56**	63.55	**31.13**	**25.79**	**35.96**	**39.17**
MSN (emd_2)	33.12	31.12	31.11	36.13	36.66	32.90	18.70	25.66	30.68
MSN (AAIP)	**28.99**	**28.25**	**28.48**	**34.18**	**31.53**	**31.45**	**16.58**	**22.58**	**27.71**

*Referring to PCN, the former term inside parentheses indicates the loss function type utilized for the coarse output, while the latter term denotes the loss function type employed for the dense output.

Table 4: The training results (EMD×10^3) of point cloud completion network on the ShapeNet dataset.

are superior to PCN, with the output of completed point clouds being closer to the ground truth. This is particularly noticeable in complex point cloud types such as the lamp class, where due to the EMD's precise reaction to local shape variations, it aids in the completion of complex shapes beyond the intrinsic differences of the model itself. Similarly, results of AAIP outperform of emd_2 across all categories. Thus, the two sets of experimental results demonstrate the superior performance of AAIP in deep learning models for point cloud completion, where it serves as an approximation scheme to solve the loss function in response to EMD.

Performance of EMD Approximation

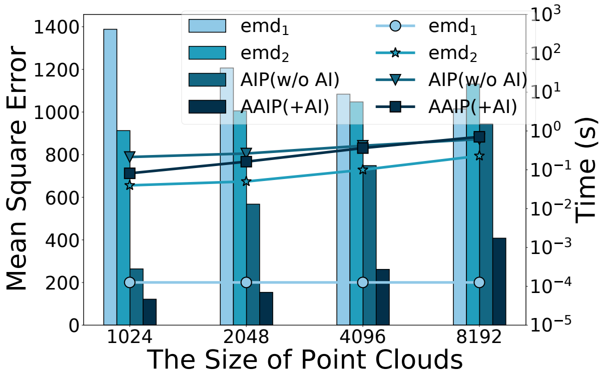

Figure 3: The accuracy and efficiency results (in terms of mean square error and time, respectively) of different EMD estimation methods for the point cloud completion problem, where we compare our proposed methods, AIP (w/o AI) and AAIP (+ AI), with two most widely used approaches (Fan, Su, and Guibas 2017; Liu et al. 2020), emd_1 and emd_2.

We conduct experiments to analyze the accuracy and efficiency of various EMD approximation schemes across point cloud data of different sizes. All experimental data are derived from the training of actual point cloud completion models, with specific details presented in the Appendix.

Within the bar chart section, it can be observed that across different point cloud size datasets, the EMD estimates of emd_1 and emd_2 diverge significantly from the actual EMD, whereas the MSE between the EMD estimated by AAIP (for both settings) and the actual EMD is substantially smaller than the other two. Conversely, AAIP maintains a low MSE in EMD calculations across various point cloud sizes. This suggests that AAIP provides superior accuracy in EMD estimation between point cloud pairs and offers robustness across different point cloud sizes. Within the line chart section, emd_1 exhibits the swiftest computational speed, with a marginal difference between our methods and emd_2. In order to complete the estimation quickly, emd_1 sacrifices the ability to correctly assign most points, leading to a significant discrepancy from the true value. These results illustrate that, compared to emd_2, AAIP guarantees the performance of the EMD approximation algorithm at the expense of minimal time cost, establishing a significant precision gap with other algorithms. Given that emd_2 is a widely adopted approximation scheme, such a time cost is acceptable.

Ablation Study on Optimization Strategy

Adaptive iteration strategy. We conducted experiments comparing the use of an Adaptive iteration strategy (denoted as "AAIP (+AI)") and its absence (designated as "AIP (w/o AI)"). After computing the initial price, AIP (w/o AI) terminated the auction algorithm through a fixed number of iterations. On point cloud data of varying sizes, the EMD estimation accuracy of AIP (w/o AI), due to the initial prices, has already surpassed that of emd_1 and emd_2. However, the efficiency and accuracy of the method have further improved after the use of the adaptive iteration strategy. This improvement is particularly evident in larger point cloud datasets, which validates that our proposed optimization strategy can effectively enhance the method's effectiveness based on the characteristics of the data.

Conclusion

In this paper, we introduce the Adaptive Auction with Initial Price Algorithm, designed to efficiently and accurately estimate the shape loss function in point cloud completion problems. Initially, we propose the use of initial prices to expedite the convergence of the auction algorithm, coupled with a theoretical proof. Then we propose an efficient calculation method for initial prices, based on the successive shortest path. In accordance with the practical scenario of loss function in deep learning models, we propose optimization strategies that adaptively aligns with data characteristics. Through experiments conducted on eight different categories of point cloud data and two distinct deep learning models, we demonstrate the effectiveness and superiority of the proposed method.

Acknowledgments

This work was supported by the Science and Technology Development Fund Macau SAR (0052/2023/RIA1, 0031/2022/A, 0015/2019/AKP, SKL-IOTSC-2021-2023), the Research Grant of University of Macau (MYRG2022-00252-FST), the National Natural Science Foundation of China under Grant No. 62202401, and Wuyi University Hong Kong and Macau joint Research Fund (2021WGALH14). This work was performed in part at SICC which is supported by SKL-IOTSC, University of Macau.

References

Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. 2018. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, 40–49. PMLR.

Altschuler, J. M.; Weed, J.; and Rigollet, P. 2017. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 1964–1974.

Beckmann, N.; Kriegel, H.; Schneider, R.; and Seeger, B. 1990. The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. In Garcia-Molina, H.; and Jagadish, H. V., eds., *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, USA, May 23-25, 1990*, 322–331. ACM Press.

Bertsekas, D. P. 1979. A distributed algorithm for the assignment problem. *Lab. for Information and Decision Systems Working Paper, MIT*.

Bertsekas, D. P. 1985. A distributed asynchronous relaxation algorithm for the assignment problem. In *1985 24th IEEE Conference on Decision and Control*, 1703–1704. IEEE.

Chan, T. N.; Yiu, M. L.; et al. 2019. The power of bounds: Answering approximate earth mover's distance with parametric bounds. *IEEE Transactions on Knowledge and Data Engineering*, 33(2): 768–781.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chang, Y.; Jung, C.; and Xu, Y. 2021. FinerPCN: High fidelity point cloud completion network using pointwise convolution. *Neurocomputing*, 460: 266–276.

Chen, X.; Chen, B.; and Mitra, N. J. 2019. Unpaired point cloud completion on real scans using adversarial training. *arXiv preprint arXiv:1904.00069*.

Chen, X.; Chen, B.; and Mitra, N. J. 2020. Unpaired Point Cloud Completion on Real Scans using Adversarial Training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In Burges, C. J. C.; Bottou, L.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2292–2300.

Dai, A.; Qi, C. R.; and Nießner, M. 2017. Shape Completion Using 3D-Encoder-Predictor CNNs and Shape Synthesis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 6545–6554. IEEE Computer Society.

Derigs, U. 1981. A shortest augmenting path method for solving minimal perfect matching problems. *Networks*, 11(4): 379–390.

Fan, H.; Su, H.; and Guibas, L. J. 2017. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2463–2471. IEEE Computer Society.

Han, X.; Li, Z.; Huang, H.; Kalogerakis, E.; and Yu, Y. 2017. High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 85–93. IEEE Computer Society.

Jang, M.-H.; Kim, S.-W.; Faloutsos, C.; and Park, S. 2011. A linear-time approximation of the earth mover's distance. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, 505–514.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Li, C.-L.; Zaheer, M.; Zhang, Y.; Poczos, B.; and Salakhutdinov, R. 2018. Point cloud gan. *arXiv preprint arXiv:1810.05795*.

Liu, M.; Sheng, L.; Yang, S.; Shao, J.; and Hu, S.-M. 2020. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11596–11603.

Liu, Y.; Fan, B.; Xiang, S.; and Pan, C. 2019a. Relation-Shape Convolutional Neural Network for Point Cloud Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 8895–8904. Computer Vision Foundation / IEEE.

Liu, Z.; Tang, H.; Lin, Y.; and Han, S. 2019b. Point-Voxel CNN for Efficient 3D Deep Learning. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 963–973.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 77–85. IEEE Computer Society.

Reutebuch, S. E.; Andersen, H.-E.; and McGaughey, R. J. 2005. Light detection and ranging (LIDAR): an emerging tool for multiple resource inventory. *Journal of forestry*, 103(6): 286–292.

Sarmad, M.; Lee, H. J.; and Kim, Y. M. 2019. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5898–5907.

Sharma, A.; Grau, O.; and Fritz, M. 2016. VConv-DAE: Deep Volumetric Shape Learning Without Object Labels. In Hua, G.; and Jégou, H., eds., *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III*, volume 9915 of *Lecture Notes in Computer Science*, 236–250.

Solomon, J.; de Goes, F.; Peyré, G.; Cuturi, M.; Butscher, A.; Nguyen, A.; Du, T.; and Guibas, L. J. 2015. Convolutional wasserstein distances: efficient optimal transportation on geometric domains. *ACM Trans. Graph.*, 34(4): 66:1–66:11.

Stutz, D.; and Geiger, A. 2018. Learning 3D Shape Completion From Laser Scan Data With Weak Supervision. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 1955–1964. Computer Vision Foundation / IEEE Computer Society.

Tchapmi, L. P.; Kosaraju, V.; Rezatofighi, H.; Reid, I. D.; and Savarese, S. 2019. TopNet: Structural Point Cloud Decoder. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 383–392. Computer Vision Foundation / IEEE.

U, L. H.; Mouratidis, K.; Yiu, M. L.; and Mamoulis, N. 2010. Optimal matching between spatial datasets under capacity constraints. *ACM Trans. Database Syst.*, 35(2): 9:1–9:44.

Waissi, G. R. 1994. Network flows: Theory, algorithms, and applications.

Wu, H.; Miao, Y.; and Fu, R. 2021. Point cloud completion using multiscale feature fusion and cross-regional attention. *Image and Vision Computing*, 111: 104193.

Yuan, W.; Khot, T.; Held, D.; Mertz, C.; and Hebert, M. 2018. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, 728–737. IEEE.