Loss Distillation via Gradient Matching for Point Cloud Completion with Weighted Chamfer Distance

Fangzhou Lin*[1,3], Haotian Liu*[1], Haoying Zhou*[1], Songlin Hou*[2],
Kazunori D Yamada[3], Gregory S. Fischer[1], Yanhua Li[4], Haichong K. Zhang[1], and Ziming Zhang†[,5]

***Abstract*— 3D reconstruction enables robots to perceive the three-dimensional structure of the environments, making it possible for many downstream tasks such as object detection and scene understanding. The performance of these tasks, though, heavily relies on the quality of data input, as incomplete or missing geometry information can lead to poor results. Recent training loss functions designed for deep learning-based point cloud completion, such as Chamfer distance (CD) and its variants (such as HyperCD [1]), imply a good gradient weighting scheme can significantly boost performance. However, these CD-based loss functions usually require data-related parameter tuning, which can be time-consuming for data-extensive tasks. To address this issue, we aim to find a family of weighted training losses (*weighted CD*) that requires no parameter tuning. To this end, we propose a search scheme, *Loss Distillation via Gradient Matching*, to find good candidate loss functions by mimicking the learning behavior in backpropagation between HyperCD and weighted CD. Once this is done, we propose a novel bilevel optimization formula to train the backbone network based on the weighted CD loss. We observe that: (1) with proper weighted functions, the weighted CD can always achieve similar performance to HyperCD, and (2) the Landau weighted CD, namely *Landau CD*, can outperform HyperCD for point cloud completion and lead to new state-of-the-art results on several benchmark datasets.** *Our demo code is available at* `https://github.com/Zhang-VISLab/IROS2024-LossDistillation`.

I. Introduction

The applications of 3D point clouds widely expand to every corner of industrial and civilian areas like object recognition [2], mapping [3], robotic grasping [4], and pose estimation [5]. However, because of occlusions, transparency, light reflections, or the limitation of the equipment's position and precision, the point clouds are usually sparse and incomplete [6]. To mitigate this issue, many learning-based point cloud completion methods [7] have been introduced, where supervised learning featured with a standard encoder-decoder architecture has emerged as the predominant choice for many researchers. These methods have achieved state-of-the-art performance on many benchmark datasets for point cloud completion [8], [9], [10], [11].

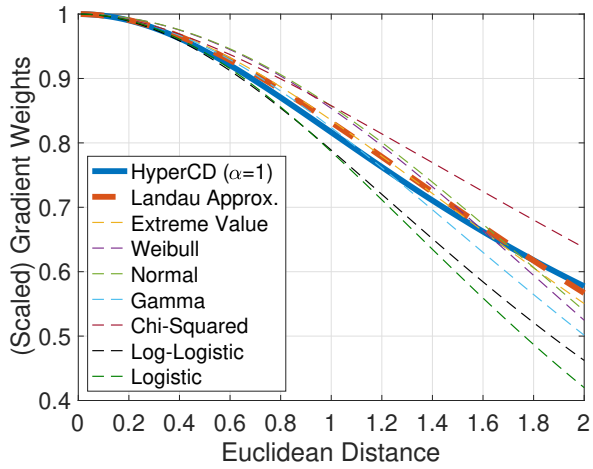


Fig. 1. Illustration of distributions (with scaling and proper hyperparameters) that are similar to the gradient weighting distribution from HyperCD in backpropagation and can be taken as candidate weighting functions in weighted CD.

A. Training Loss

Chamfer distance (CD) serves as a popular training loss in point cloud completion for training neural networks such as SnowflakeNet [9] and PointAttN [11]. It evaluates the dissimilarity between any two point clouds by calculating the average distances of each point in one set to its nearest matching point in the other set. CD can faithfully reflect the global dissimilarity by treating the distances of all nearest-neighbor pairs between both sets with equal importance. However, it is not an ideal loss function solution for network training. The formation of CD works as the uniform distribution weight operation for paired distance, and thus, it is likely to be negatively affected by some outlier points. As the consequence, the sensitivity to outliers often results in a phenomenon where a considerable number of points from one set correspond to a single point in the other set, leading to the visual formation of small and dense clusters. This behavior can readily disrupt the commonly used assumption of uniform sampling from the underlying geometric surfaces, which is often used in the generation of point clouds. To mitigate these aforementioned problems in point cloud completion, several CD variant loss functions have been proposed: *Density-aware CD (DCD)* [12], *InfoCD*

This work was supported by part of NSF CCF-2006738.
*These authors contributed equally to this work.
†Corresponding author.
[1]Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA
[2]Dell Technologies, Hopkinton, MA 01748, USA
[3]Graduate School of Information Sciences, Tohoku University, Sendai, 980-8579, Japan
[4]Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA
[5]Department of Electrical & Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA (zzhang15@wpi.edu)

[13], and *HyperCD* [1].

B. Motivation

From a high-level understanding, we would like to propose a general method to efficiently learn loss functions for downstream tasks, as this is crucial for deep learning nowadays. Different from conventional hyperparameter tuning with scalars, we aim to explore functional spaces to search for good functions. Considering the specifications of point cloud completion, the sensitivity of CD to the outliers inspires us that the *distances in the metric as a training loss should be weighted in some form rather than uniform*. This provides us a good testbed to evaluate our high-level idea because currently, weighted CD is highly underexplored in this field. To address this issue, we borrow the idea from network distillation, where a simpler student network is trained to mimic the behavior of a more complicated teacher network. Rather than networks, we aim to learn weighted CD losses instead.

C. Approach: Loss Distillation via Gradient Matching

Loss functions guide the networks during training based on gradients through backpropagation, while gradients contain all the knowledge from the loss for training the networks. If we can reproduce the exact gradients in training, we can then reproduce the performance of a certain loss. Based on such considerations, we propose a family of training losses for point cloud completion using weighted CD to mimic the learning behavior of HyperCD by approximately matching the gradients. As illustrated in Fig. 1 (see more details in Sec. III-C), by taking the gradient weighting function in HyperCD as a reference, we can easily find some distributions as candidate weighting functions for weighted CD to approximate the reference curve, especially when the distance is small.

D. Contributions

We list our main contributions as follows:

- We propose an efficient gradient-matching method for loss distillation to select candidate weighting functions for weighted CD from a pool of potential distributions.
- We demonstrate strong performance for point cloud completion based on weighted CD that can always be similar to our reference loss, even leading to state-of-the-art results on several benchmark datasets.
- *Our loss distillation method is so naive, yet effective and efficient, to determine good loss functions that all the calculations can be done using* **simple simulated data with mathematical derivations**. It does provide us the solutions for our problems and potentially for other downstream tasks as well by matching reference gradients.

II. Related Work

A. Distance Metrics for Point Cloud Completion

Distance in point clouds refers to a non-negative function that measures the dissimilarity among them [14]. Considering the keen demand for high-density point cloud, the structures of point cloud completion networks have become increasingly complicated [15]. CD and its variants are extensively used in almost all recent learning-based methods for point cloud completion [16], [17], [18], [19].

B. Knowledge Distillation (KD)

Generally, KD [20] refers to a model compression method in machine learning, where a smaller, more compact neural network (*i.e*. student model) is trained to replicate the behavior of a larger, more complex network (*i.e*. teacher model) [21]. The teacher model is used to produce the outputs of knowledge, while the student model tries to learn such knowledge by mimicking the outputs. Some nice survey papers on this topic can be found in [22], [23], [24].

C. Weighted Chamfer Distance

In 2D image processing, weighted distances have become notable in generating distance maps from point lattice [25] and image segmentation [26]. In the distance map generation task, this methodology facilitates the computation of rotation-invariant distances through optimal weighting, particularly in face-centered cubic [27] and body-centered cubic lattice structures [25], [28]. In 3D point cloud applications, the weighted CD emerges as a pivotal loss function and metric [29], [30], [31]. However, to the best of our knowledge, in point cloud completion we do not find any reference based on weighted CD.

III. Our Approach

A. Chamfer Distance (CD)

We denote (x_i, y_i) as the i-th point cloud pair, $x_i = \{x_{ij}\}$ and $y_i = \{y_{ik}\}$ as two sets of 3D points, and $d(\cdot,\cdot)$ as a certain distance metric. Then the CD loss for point clouds can be defined as follows:

$$D_{CD}(x_i, y_i) = \frac{1}{|x_i|}\sum_{j=1}^{|x_i|}\min_k d(x_{ij}, y_{ik}) + \frac{1}{|y_i|}\sum_{k=1}^{|y_i|}\min_j d(x_{ij}, y_{ik}), \quad (1)$$

where $|\cdot|$ denotes the cardinality of a set. Note that for point cloud completion, function d is usually defined in Euclidean space, referring to

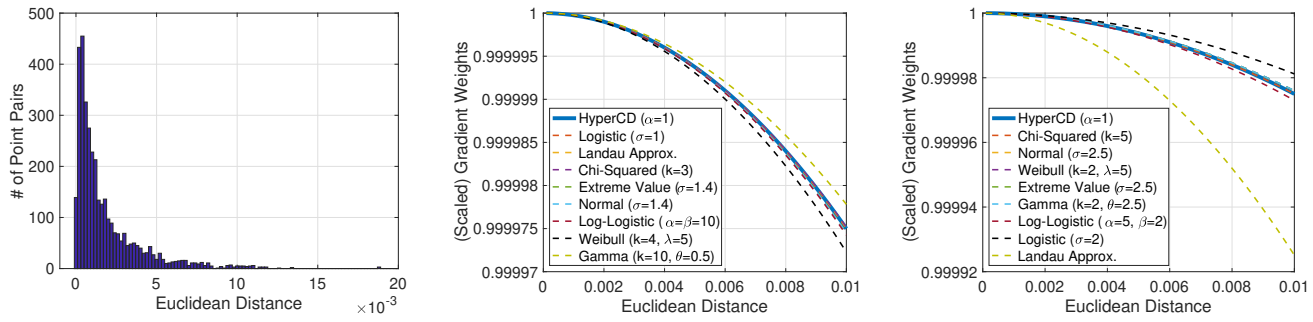
$$d(x_{ij}, y_{ik}) = \begin{cases} \|x_{ij} - y_{ik}\| & \text{as } \textit{L1-distance} \\ \|x_{ij} - y_{ik}\|^2 & \text{as } \textit{L2-distance} \end{cases} \quad (2)$$

where $\|\cdot\|$ denotes the ℓ_2 norm of a vector.

B. Hyperbolic Chamfer Distance (HyperCD)

Based on Eq. 1, HyperCD defines the function d in a hyperbolic space as follows:

$$d(x_{ij}, y_{ik}) = arccosh\left(1 + \alpha\|x_{ij} - y_{ik}\|^2\right), \alpha > 0. \quad (3)$$



(a) (b) (c)

Fig. 2. Illustration of **(a)** reference distance distribution from HyperCD, and **(b-c)** curve fitting using different approximations of $z^{(W)}$.

C. Loss Distillation with Weighted CD

Weighted Chamfer Distance. In this paper, we propose the following formula for our weighted CD:

$$D_W(x_i, y_i) = \frac{1}{|x_i|}\sum_{j=1}^{|x_i|} f(\tilde{d}_{ijk})\tilde{d}_{ijk} + \frac{1}{|y_i|}\sum_{k=1}^{|y_i|} f(\tilde{d}_{ikj})\tilde{d}_{ikj}$$
$$\text{s.t. } \tilde{d}_{ijk} = \min_k d(x_{ij}, y_{ik}), \tilde{d}_{ikj} = \min_j d(x_{ij}, y_{ik}). \quad (4)$$

Clearly, the vanilla CD is a special case of our new formula with $f(\tilde{d}_{ijk}) = f(\tilde{d}_{ikj}) = 1, \forall \tilde{d}_{ijk}, \tilde{d}_{ikj} \geq 0$.

Loss Distillation via Gradient Matching. For point cloud completion, let us denote (x_i, y_i) as the output from the network and the ground-truth point cloud, respectively. Precisely, we denote $x_i = h(\tilde{x}_i; \omega)$ where function h is presented by the network with parameters ω and $\tilde{x}_i$ is an incomplete point cloud as the input. Therefore, each point x_{ij} is also a function of ω, and so are $\tilde{d}_{ijk}$ and $\tilde{d}_{ikj}$.

Recall that the goal of gradient matching is to develop effective losses based on weighted CD by mimicking the learning behavior of HyperCD. To simplify our explanation of gradient matching for loss distillation, we denote $g^{(H)}_{ijk} = arccosh\left(1 + \alpha\tilde{d}^2_{ijk}\right), g^{(W)}_{ijk} = f(\tilde{d}_{ijk})\tilde{d}_{ijk}$. To match a pair of gradients from both losses, we propose minimizing their difference as follows:

$$\left\|\frac{\partial D_H}{\partial \omega} - \frac{\partial D_W}{\partial \omega}\right\|$$
$$\leq \frac{1}{|x_i|}\left\|\sum_j\left(\frac{\partial g^{(H)}_{ijk}}{\partial \omega} - \frac{\partial g^{(W)}_{ijk}}{\partial \omega}\right)\right\| + \frac{1}{|y_i|}\left\|\sum_k\left(\frac{\partial g^{(H)}_{ikj}}{\partial \omega} - \frac{\partial g^{(W)}_{ikj}}{\partial \omega}\right)\right\|$$
$$\leq \frac{1}{|x_i|}\sum_j\left\|z^{(H)}_{ijk} - z^{(W)}_{ijk}\right\|\left\|\frac{\partial \tilde{d}_{ijk}}{\partial \omega}\right\| + \frac{1}{|y_i|}\sum_k\left\|z^{(H)}_{ikj} - z^{(W)}_{ikj}\right\|\left\|\frac{\partial \tilde{d}_{ikj}}{\partial \omega}\right\|$$

where $z^{(H)}_{ijk} = \frac{2\alpha\tilde{d}_{ijk}}{\sqrt{\left(1+\alpha\tilde{d}^2_{ijk}\right)^2 - 1}}, z^{(W)}_{ijk} = f'(\tilde{d}_{ijk})\tilde{d}_{ijk} + f(\tilde{d}_{ijk})$ are *gradient weights* for the HyperCD loss, D_H, and the weighted CD loss, respectively (*resp.* $z^{(H)}_{ikj}, z^{(W)}_{ikj}$), and f' denotes the derivative of function f.

Minimizing the LHS of Eq. 5 is very challenging, because we do not have prior knowledge about the network and data. To get rid of the effects of such unknown information in learning, we instead try to minimize the RHS of Eq. 5 with the following assumptions on

- *Network:* All the gradients can be upper-bounded.
- *Data:* $|x_i|, |y_i|$ are sufficiently large so that the distributions of $\tilde{d}_{ijk}, \tilde{d}_{ikj}$ follow the reference distance distribution from HyperCD.

In point cloud completion, both assumptions can hold easily. Finally, due to the symmetry of Eq. 5, we propose the following minimization problem for loss distillation:

$$\min_{f\in\mathcal{F}} \mathbb{E}_{\tilde{d}\sim\tilde{\mathcal{D}}}\left\|z^{(H)}(\tilde{d}) - z^{(W)}(\tilde{d})\right\|$$
$$\approx \min_{f\in\mathcal{F}}\sum_{\tilde{d}} p(\tilde{d})\left\|z^{(H)}(\tilde{d}) - z^{(W)}(\tilde{d})\right\|, \quad (5)$$

where $z^{(H)}(\tilde{d}) = \frac{2\alpha\tilde{d}}{\sqrt{\left(1+\alpha\tilde{d}^2\right)^2 - 1}}, z^{(W)}(\tilde{d}) = f'(\tilde{d})\tilde{d} + f(\tilde{d})$, $\mathcal{F}$ denotes the feasible space for f, $\tilde{D}$ denotes the reference distance distribution, and $\mathbb{E}$ denotes the expectation operator. Note that when Eq. 5 reaches 0 , it will guarantee to recover the performance of HyperCD using weighted CD.

TABLE I
DISTRIBUTIONS AS WEIGHTING FUNCTIONS IN WEIGHTED CD.

Distribution	**Params**	**Mode** m	**PDF**
Chi-Squared	k	$\max(k-2, 0)$	$\frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2}$
Extreme Value	β	0	$\frac{1}{\beta} e^{-(z+e^{-z})}, z = \frac{x}{\beta}$
Weibull	k, λ	$\begin{cases}\lambda\left(\frac{k-1}{k}\right)^{1/k}, & k > 1,\\ 0, & k \leq 1.\end{cases}$	$\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$
Log-Logistic	α, β	$\begin{cases}\alpha\left(\frac{\beta-1}{\beta+1}\right)^{1/\beta}, & \text{if } \beta > 1,\\ 0, & \text{otherwise}\end{cases}$	$\frac{\beta}{x}\left(1 + \left(\frac{x}{\alpha}\right)^{-\beta}\right)^{-1-\beta}$
Gamma	α, β	$\begin{cases}\frac{\alpha-1}{\beta}, & \text{for } \alpha \geq 1,\\ 0, & \text{for } \alpha < 1\end{cases}$	$\frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$
Logistic	σ	0	$\frac{e^{-x/\sigma}}{\sigma(1+e^{-x/\sigma})^2}$
Normal	σ	0	$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma^2}\right)$
Landau Approx.	-	0	$\frac{1}{\sqrt{2\pi}}\exp\left\{\left(-\frac{x+e^{-x}}{2}\right)\right\}$

D. Optimization

To solve Eq. 5, we first specify the notations in our implementation as follows:

- *Feasible space $\mathcal{F}$:* Table I lists some well-known distributions that we tested as the weighting functions for weighted CD. Each distribution defines an $\mathcal{F}$, and we try to learn its parameters, if exist, to determine f.
- *Distance $\tilde{d}$:* Recall that $z^{(H)}$ in HyperCD is monotonically decreasing and $\tilde{d} = 0$ reaches the maximum. To mimic this behavior, we only consider the partial distributions beyond their modes. Correspondingly, the input data for each distribution is its mode, m, plus distance $\tilde{d}$.
- *Reference distance distribution $\tilde{\mathcal{D}}$ and samples $\tilde{d}$:* Fig. 2 (a) illustrates the distance distribution from HyperCD, which is used as $\tilde{\mathcal{D}}$ in our implementation. As we see, about 99% of point pairs, *i.e.* $\tilde{d}$, fall into $[0, 0.01]$. Therefore, to optimize Eq. 5 efficiently, we uniformly sample $\tilde{d}$ from $[0, 0.01]$ with step 2e-4, and the corresponding probabilities are sampled from Fig. 2 (a).
- *Approximation of $z^{(W)}$:* The exact computation of f' in $z^{(W)}$ causes trouble, even if we may know its function (*e.g.* for some functions we may not have their analytical solutions of their gradients). To address this issue, we propose the following two ways: (1) $z^{(W)}(\tilde{d}) \approx f(m+\tilde{d})$, because $\tilde{d} \in [0, 0.01]$ is very small and thus the calculation of $z^{(W)}$ may be dominated by the second term; or (2) substituting $f'(\tilde{d}) \approx \frac{1}{\Delta\tilde{d}}(f(m+\tilde{d}+\Delta\tilde{d}) - f(m+\tilde{d}))$ into $z^{(W)}$ with small value $\Delta\tilde{d} \geq 0$. Based on these two strategies, we plot the curves in Fig. 2 (b-c), respectively, where all the curves are rescaled by the maximum values and ordered by the minimum of Eq. 5. The optimal parameters are determined using a grid search for simplicity and efficiency. As we see, all eight distributions can well fit the reference curve from HyperCD, and the parameters listed in the figure are used to report the performance of weighted CD in our experiments. Finally, we list our gradient matching algorithm in Alg. 1.

Algorithm 1 Loss Distillation via Gradient Matching

Input : a PDF f with parameters $\mathcal{A}$, $z^{(H)}$, $\{(\tilde{d}, p(\tilde{d}))\}$
Output: $\mathcal{A}$

Discretize the parameter space into $\{\mathcal{A}_i\}$ for grid search;
Compute the mode m_i for each $\mathcal{A}_i$ used in $z^{(W)}$;
$\mathcal{A}^* = argmin_{\{\mathcal{A}_i\}} \sum_{\tilde{d}} p(\tilde{d}) \left\| z^{(H)}(\tilde{d}) - z^{(W)}(\tilde{d}) \right\|$;
return $\mathcal{A} \leftarrow \mathcal{A}^*$

E. Bilevel Optimization with Weighted CD

Once we choose the weighting function in weighted CD, we propose optimizing the following optimization problem for point cloud completion, given training samples $\{(\tilde{x}_i, y_i)\}$:

$$\min_{\omega} \sum_i \left[\frac{1}{|x_i|} \sum_{j=1}^{|x_i|} f(\tilde{d}_{ijk}) \tilde{d}_{ijk} + \frac{1}{|y_i|} \sum_{k=1}^{|y_i|} f(\tilde{d}_{ikj}) \tilde{d}_{ikj} \right]$$
$$\text{s.t. } x_i = h(\tilde{x}_i; \omega) = \{x_{ij}\}, \forall i,$$
$$\tilde{d}_{ijk} = \min_k d(x_{ij}, y_{ik}), \tilde{d}_{ikj} = \min_j d(x_{ij}, y_{ik}). \quad (6)$$

Essentially, this defines a bilevel optimization problem that can be solvable using the iterative differentiation algorithm [32], leading to our algorithm in Alg. 2. In fact, the learning algorithms for both HyperCD follow the same bilevel optimization strategy as ours.

Algorithm 2 Point Cloud Completion with Weighted CD

Input : a weighting function f, a network h with learnable parameters ω, training data $\{(\tilde{x}_i, y_i)\}$
Output: ω

repeat
  Pick a sample $(\tilde{x}_i, y_i)$ uniformly at random;
  Compute $\tilde{d}_{ijk}, \forall j$ in $\tilde{x}_i$ and $\tilde{d}_{ikj}, \forall k$ in y_i;
  Compute the weighted CD loss based on Eq. 4;
  Update the parameters ω using the gradient of the loss;
until *converges*;
return ω

IV. Experiments

A. Datasets

In our experiments we use PCN [33], ShapeNet-55 [8], ShapeNet-Part [34], and KITTI [35]. For the dataset details of PCN, ShapeNet-55, and ShapeNet-Part, you may refer to the HyperCD paper [1]. KITTI is composed of a sequence of real-world Velodyne LiDAR scans, also derived from the PCN dataset [33]. For each frame, the car objects are extracted according to the 3D bounding boxes, which results in 2,401 partial point clouds. The partial point clouds in KITTI are highly sparse and do not have complete point clouds as ground truth. Datasets are split into training, testing, and validation sets by the ratios of 70%, 20%, and 10%, respectively.

B. Network Backbones

We compare our method using 7 different backbone networks, *i.e.* FoldingNet [36], PMP-Net [37], PoinTr [8], SnowflakeNet [9], CP-Net [38], PointAttN [11] and SeedFormer [10], by replacing the CD loss with our weighted CD losses wherever it occurs.

C. Hyperparameters

The hyperparameters in the weighting functions are selected from the candidate functions shown in Fig. 2 (b-c) that achieve better performance. Except that the learning rates are tuned slightly, the training hyperparameters such as batch sizes and balance factors in the original losses are kept the same as HyperCD.

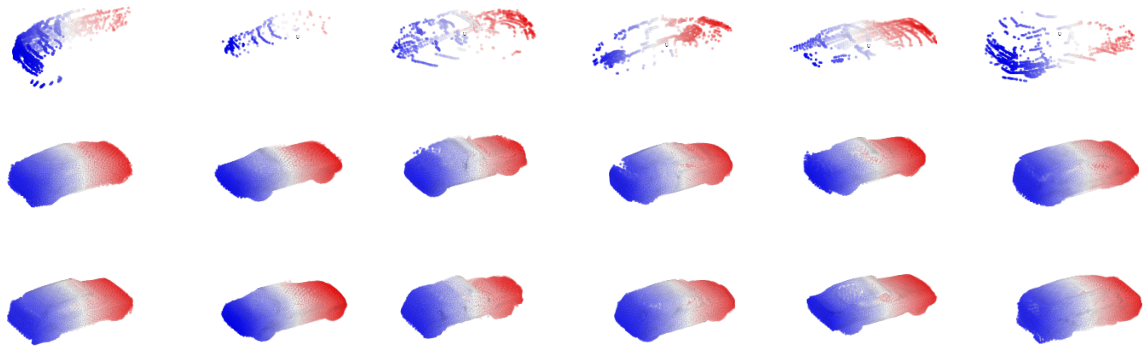


Fig. 3. Visualization of the real-world(KITTI) benchmark (**Row 1:** sparse input, **Row 2:** HyperCD, **Row 3:** LandauCD).

D. Evaluation

Following the literature, we evaluate the best performance of all the methods using vanilla CD (lower is better). We also use F1-Score@1% [39] (higher is better) to evaluate the performance on ShapeNet-55. For KITTI, we use Fidelity and MMD metrics [8]. For better comparison, we cite the original results of some other methods on PCN, ShapeNet-55, and KITTI. In each table of the results, the top-performing results are highlighted in red, while the second-highest ones are marked in blue.

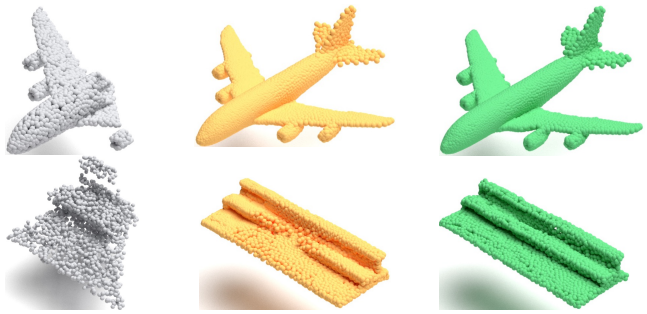


Fig. 4. Visualization of ShapeNet-55 benchmark. Gray represents the partial input. Yellow represents HyperCD. Green represents Landau CD.

TABLE II

CP-NET COMPARISON RESULTS ON SHAPENET-PART WITH DIFFERENT LOSSES. IN THE SEQUEL, WE COLOR THE BEST PERFORMANCE WITH *red*, AND SECOND BEST WITH *blue*.

Loss Function	Loss Params.	L2-CD$\times 10^3$
CD	\	4.16
EMD	\	15.38
Truncated CD	thd=0.2	4.72
DCD [12]	α = 40, γ=0.5	5.74
HyperCD [1]	α=1	4.03
Weibull CD	k=2, λ=5	4.19
Normal CD	σ=1.4	4.17
Logistic CD	σ=1	4.14
Log-Logistic CD	α=5, β=2	4.12
Extreme-Value CD	σ=1.4	4.08
Chi-Squared CD	k=3	4.07
Gamma CD	k=2, θ=2.5	4.03
Landau CD	\	4.00$\pm$0.005

E. Ablation Study

For this purpose, we use CP-Net as the backbone network and train it on ShapeNet-Part with different losses. We refer to our different weighted CDs based on the names of the distributions as weighting functions. For instance, we call a weighted CD **Landau CD** if the weighting function follows the Landau approximation distribution.

F. Performance

Table II summarizes our comparison results where we report the best performance for all the methods (our loss parameters are chosen from Fig. 2 (b-c) through gradient matching). As we see here, 6 out of 8 weighted CD losses perform better than CD, and 4 of them perform similarly to HyperCD (within the difference of ± 0.05). Surprisingly our Landau CD even beats the state-of-the-art. Notice that the performance ranking of weighted CD losses is not consistent with the function matching ranking in Fig. 2 (b-c), indicating that the selected weighting functions have to be tested with the weighted CD losses. Overall, such results demonstrate that our weighted CD can mimic the learning behavior of HyperCD, with proper weighting functions and parameters, and have great potential for boosting CD performance significantly.

G. Convergence

Fig. 5 provides a direct visual juxtaposition of the convergence trends, revealing that our weighted CD losses exhibit a more rapid convergence compared to HyperCD. Notably, during the initial 50 epochs, the curves of weighted CD loss functions consistently remain lower than the ones of HyperCD, and eventually, all the curves converge to a similar loss, leading to similar performance as well. Such convergence behavior of our weighted CD also demonstrates the success of our loss distillation method.

H. State-of-the-art Comparison

PCN. In accordance with the literature, we report performances in terms of vanilla CD with L1-distance in Table III. As we can see, most of weighted CD losses achieve above-average results compared with the baseline networks used in training. In particular, we obtain some new state-of-the-art results using Landau CD. Meanwhile, Extreme-Value

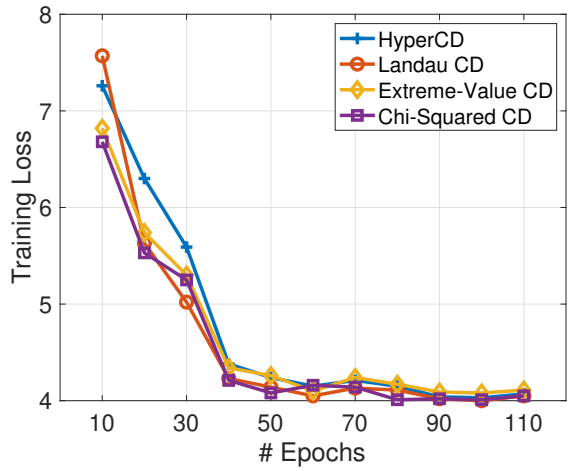


Fig. 5. Loss convergence of CP-Net on ShapeNet-Part.

TABLE III
COMPARISON ON PCN IN TERMS OF PER-POINT L1-CD ×1000.

Methods	Plane	Car	Chair	Lamp	Couch	Avg.
FoldingNet [36]	9.49	12.61	15.55	16.41	15.97	14.01
HyperCD + F.	7.89	10.67	14.55	13.87	14.09	12.21
Landau CD + F.	7.30	10.46	13.00	11.92	13.39	11.21
PMP [37]	5.65	9.64	9.51	6.95	10.83	8.55
HyperCD + PMP	5.06	9.30	9.11	6.83	11.01	8.26
Landau CD + PMP	4.59	8.90	8.57	6.38	10.47	7.78
PoinTr [8]	4.75	8.68	9.39	7.75	10.93	8.33
HyperCD + P.	4.42	8.22	8.22	6.62	9.62	7.42
Landau CD + P.	4.12	8.07	7.82	6.30	9.28	7.12
SnowflakeNet [9]	4.29	8.08	7.89	6.07	9.23	7.11
HyperCD + S.	3.95	7.88	7.37	5.75	8.94	6.78
Landau CD + S.	3.98	7.78	7.40	5.76	8.86	6.76
PointAttN [11]	3.87	7.63	7.43	5.90	8.68	6.70
DCD + PA.	4.47	8.14	8.12	6.75	9.60	7.41
HyperCD + PA.	3.76	7.49	7.06	5.61	8.48	6.48
Gamma CD + PA.	3.83	7.58	7.15	5.69	8.56	6.56
Chi-Sq. CD + PA.	3.77	7.49	7.08	5.64	8.50	6.49
Log-Logis. CD + PA.	3.78	7.47	7.10	5.63	8.51	6.50
Ex.-Va. CD + PA.	3.73	7.46	7.03	5.61	8.46	6.46
Landau CD + PA.	3.72	7.46	7.04	5.60	8.47	6.46
SeedFormer [10]	3.85	8.06	7.06	5.21	8.85	6.61
DCD + SF.	16.42	21.08	20.06	18.30	26.51	20.47
HyperCD + SF.	3.72	7.79	6.83	5.11	8.61	6.41
Log-Logis. CD + SF.	3.86	7.79	6.89	5.15	8.64	6.47
Gamma CD + SF.	3.84	7.82	6.89	5.13	8.63	6.46
Chi-Sq. CD + SF.	3.75	7.71	6.80	5.11	8.48	6.45
Ex.-Va. CD + SF.	3.73	7.70	6.80	5.08	8.48	6.36
Landau CD + SF.	3.65	7.64	6.80	5.04	8.57	6.34

and Chi-Squared CD losses also achieve better performance than HyperCD in more complicated backbone networks PointAttN and SeedFormer. In the sequel, by default we will only report the results using Landau CD, due to its great performance on PCN.

KITTI. In order to validate the effectiveness of weighted CD loss functions in real-world scenarios, we follow the method used in [40] to fine tune two baseline models with Landau CD on ShapeNetCars [33] and evaluate the performance on KITTI. We report the Fidelity and MMD metrics in Table IV. We observe that Landau CD can improve the baselines consistently with even better results compared with HyperCD. Furthermore, we present comprehensive visualization results in Fig. 3. Note both HyperCD and Landau CD are able to recover the general geometrical structure from partial sparse input, Landau CD, however, perform better with less noise and outliers, especially on small details on corners and edges.

TABLE IV
RESULTS ON LIDAR SCANS FROM KITTI DATASET UNDER THE FIDELITY AND MMD METRICS.

	FoldingNet	HyperCD+F.	**Landau CD+F.**
Fidelity ↓	7.467	2.214	1.956
MMD ↓	0.537	0.386	0.342
	PoinTr	HyperCD+P.	**Landau CD+P.**
Fidelity ↓	0.000	0.000	0.000
MMD ↓	0.526	0.507	0.503

TABLE V
COMPARISON ON SHAPENET-55 IN TERMS OF L2-CD×1000 AND F1 SCORE (HIGHER THE BETTER).

Methods	CD-S	CD-M	CD-H	Avg.	F1
FoldingNet	2.67	2.66	4.05	3.12	0.082
HyperCD + F.	2.43	2.45	3.88	2.92	0.109
Landau CD + F.	2.15	2.46	3.39	2.66	0.141
PoinTr	0.58	0.88	1.79	1.09	0.464
HyperCD + P.	0.54	0.85	1.73	1.04	0.499
Landau CD + P.	0.43	0.70	1.47	0.88	0.527
SeedFormer	0.50	0.77	1.49	0.92	0.472
HyperCD + S.	0.47	0.72	1.40	0.86	0.482
Landau CD + S.	0.45	0.73	1.39	0.86	0.489

ShapeNet-55. Table V enumerates the performance across three levels of difficulty with the average. Qualitative evaluation results are shown in Fig. 4 as well from Seedformer trained with HyperCD and Landau CD as a supplement to numerical values. We can see Landau CD can successfully learn the general geometrical structure like HyperCD, which matches our intuition. When dealing with corners and edges, Landau CD even outperforms HyperCD with reduced noise and outliers.

V. CONCLUSION

In this paper, we propose a novel loss distillation method for point cloud completion by mimicking the learning behavior of HyperCD based on weighted CD. To this end, we propose an efficient and effective gradient matching algorithm to search for potential weighting functions from a pool of distributions for weighted CD by comparing them with the reference curve from HyperCD. This eventually converts to a bilevel optimization problem in training backbone networks, with empirical convergence based on our iterative differentiation algorithm. We conduct comprehensive experiments using real-world datasets such as KITTI[35] to demonstrate the effectiveness of weighted CD losses, particularly Landau CD which achieves new state-of-the-art results on several benchmark datasets.

REFERENCES

[1] F. Lin, Y. Yue, S. Hou, X. Yu, Y. Xu, K. D. Yamada, and Z. Zhang, "Hyperbolic chamfer distance for point cloud completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 14 595–14 606.

[2] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional shapecontextnet for point cloud recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4606–4615.

[3] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "Tumindoor: An extensive image and point cloud dataset for visual indoor localization and mapping," in *2012 19th IEEE International Conference on Image Processing*. IEEE, 2012, pp. 1773–1776.

[4] P. Ni, W. Zhang, X. Zhu, and Q. Cao, "Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3619–3625.

[5] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[6] F. Leberl, A. Irschara, T. Pock, P. Meixner, M. Gruber, S. Scholz, and A. Wiechert, "Point clouds," *Photogrammetric Engineering & Remote Sensing*, vol. 76, no. 10, pp. 1123–1134, 2010.

[7] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 12, pp. 4338–4364, 2020.

[8] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *ICCV*, 2021.

[9] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. Han, "Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *ICCV*, 2021.

[10] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, and C. Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer," *arXiv preprint arXiv:2207.10315*, 2022.

[11] J. Wang, Y. Cui, D. Guo, J. Li, Q. Liu, and C. Shen, "Pointattn: You only need attention for point cloud completion," *arXiv preprint arXiv:2203.08485*, 2022.

[12] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 29 088–29 100.

[13] F. Lin, Y. Yue, Z. Zhang, S. Hou, K. Yamada, V. B. Kolachalama, and V. Saligrama, "InfoCD: A contrastive chamfer distance loss for point cloud completion," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[14] F. Mémoli and G. Sapiro, "Comparing point clouds," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp. 32–40.

[15] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *arXiv:2007.08501*, 2020.

[16] H. Deng, T. Birdal, and S. Ilic, "3d local features for direct pairwise registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3244–3253.

[17] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3d point cloud completion," *arXiv preprint arXiv:2112.03530*, 2021.

[18] K. Zhang, X. Yang, Y. Wu, and C. Jin, "Attention-based transformation from latent features to point clouds," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 3291–3299.

[19] J. Tang, Z. Gong, R. Yi, Y. Xie, and L. Ma, "Lake-net: topology-aware point cloud completion by localizing aligned keypoints," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1726–1735.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[21] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 6, pp. 3048–3068, 2021.

[22] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[23] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 6, pp. 3048–3068, 2021.

[24] A. Alkhulaifi, F. Alsahli, and I. Ahmad, "Knowledge distillation in deep learning and its applications," *PeerJ Computer Science*, vol. 7, p. e474, 2021.

[25] C. Fouard, R. Strand, and G. Borgefors, "Weighted distance transforms generalized to modules and their computation on point lattices," *Pattern Recognition*, vol. 40, no. 9, pp. 2453–2474, 2007.

[26] A. Protiere and G. Sapiro, "Interactive image segmentation via adaptive weighted distances," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1046–1057, 2007.

[27] K. Petkov, F. Qiu, Z. Fan, A. E. Kaufman, and K. Mueller, "Efficient lbm visual simulation on face-centered cubic lattices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 802–814, 2009.

[28] A. Entezari, D. Van De Ville, and T. Moller, "Practical box splines for reconstruction on the body centered cubic lattice," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 313–328, 2008.

[29] M. A. Gennert and A. L. Yuille, "Determining the optimal weights in multiple objective function optimization," in *ICCV*, 1988, pp. 87–89.

[30] A. F. Guarda, N. M. Rodrigues, and F. Pereira, "Neighborhood adaptive loss function for deep learning-based point cloud coding with implicit and explicit quantization," *IEEE MultiMedia*, vol. 28, no. 3, pp. 107–116, 2020.

[31] D. Urbach, Y. Ben-Shabat, and M. Lindenbaum, "Dpdist: Comparing point clouds using deep point cloud distance," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 545–560.

[32] K. Ji, J. Yang, and Y. Liang, "Bilevel optimization: Convergence analysis and enhanced design," in *International conference on machine learning*. PMLR, 2021, pp. 4882–4892.

[33] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: point completion network," in *3DV*, 2018.

[34] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3d shape collections," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.

[35] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[36] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 206–215.

[37] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Pmp-net: Point cloud completion by learning multi-step point moving paths," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7443–7452.

[38] F. Lin, Y. Xu, Z. Zhang, C. Gao, and K. D. Yamada, "Cosmos propagation network: Deep learning model for point cloud completion," *Neurocomputing*, vol. 507, pp. 221–234, 2022.

[39] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3d reconstruction networks learn?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3405–3414.

[40] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "Grnet: Gridding residual network for dense point cloud completion," in *ECCV*, 2020.