Diffusion Probabilistic Models for 3D Point Cloud Generation

用於 3D 點雲生成的擴散概率模型

Shitong Luo, Wei Hu *
Wangxuan Institute of Computer Technology
Peking University
{luost, forhuwei}@pku.edu.cn

CVPR 2021 Best Paper Candidate
https://arxiv.org/abs/2103.01458

Abstract 摘要

We present a probabilistic model for point cloud generation, which is fundamental for various 3D vision tasks such as shape completion, upsampling, synthesis and data augmentation.
我們提出了一個用於點雲生成的概率模型，它是各種 3D 視覺任務（例如形狀完成、上採樣、合成和數據增強）的基礎。

Inspired by the diffusion process in nonequilibrium thermodynamics, we view points in point clouds as particles in a thermodynamic system in contact with a heat bath, which diffuse from the original distribution to a noise distribution.
受非平衡熱力學中擴散過程的啟發，我們將點雲中的點視為與熱浴接觸的熱力學系統中的粒子，它們從原始分佈擴散到噪聲分佈。

Point cloud generation thus amounts to learning the reverse diffusion process that transforms the noise distribution to the distribution of a desired shape.
因此，點雲生成相當於學習將噪聲分佈轉換為所需形狀分佈的反向擴散過程。

Specifically, we propose to model the reverse diffusion process for point clouds as a Markov chain conditioned on certain shape latent.
具體來說，我們建議將點雲的反向擴散過程建模為以特定形狀為條件的馬爾可夫鏈。

We derive the variational bound in closed form for training and provide implementations of the model.
我們推導出封閉形式的變分界用於訓練並提供模型的實現。

Experimental results demonstrate that our model achieves competitive performance in point cloud generation and auto-encoding.
實驗結果表明，我們的模型在點雲生成和自動編碼方面取得了有競爭力的性能。

The code is available at https://github.com/luost26/diffusionpoint-cloud.
該代碼可在

1. Introduction 前言

With recent advances in depth sensing and laser scanning, point clouds have attracted increasing attention as a popular representation for modeling 3D shapes.
隨著深度傳感和激光掃描的最新進展，點云作為 3D 形狀建模的流行表示已經引起越來越多的關注。

Significant progress has been made in developing methods for point cloud analysis, such as classification and segmentation [16, 17, 23].
在開發點雲分析方法方面取得了重大進展，例如分類和分割 [16、17、23]。

On the other hand, learning generative models for point clouds is powerful in unsupervised representation learning to characterize the data distribution, which lays the foundation for various tasks such as shape completion, upsampling, synthesis, etc.
另一方面，學習點雲的生成模型在無監督表示學習中功能強大，可以表徵數據分佈，為形狀補全、上採樣、合成等各種任務奠定基礎。

Generative models such as variational auto-encoders (VAEs), generative adversarial networks (GANs), normalizing flows, etc., have shown great success in image generation [13, 8, 5, 6].
變分自動編碼器 (VAE)、生成對抗網絡 (GAN)、標準化流等生成模型在圖像生成方面取得了巨大成功 [13, 8, 5, 6]。

However, these powerful tools cannot be directly generalized to point clouds, due to the irregular sampling patterns of points in the 3D space in contrast to regular grid structures underlying images.
然而，這些強大的工具不能直接推廣到點雲，因為與圖像底層的規則網格結構相比，3D 空間中點的不規則採樣模式。

Hence, learning generative models for point clouds is quite challenging.
因此，學習點雲的生成模型非常具有挑戰性。

Prior research has explored point cloud generation via GANs [1, 22, 19], auto-regressive models [21], flowbased models [25] and so on.
先前的研究探索了通過 GAN [1, 22, 19]、自回歸模型 [21]、基於流的模型 [25] 等生成點雲。

While remarkable progress has been made, these methods have some inherent limitations for modeling point clouds. For instance, the training procedure could be unstable for GANs due to the adversarial losses.
雖然已經取得了顯著的進展，但這些方法對於點雲建模存在一些固有的局限性。 例如，由於對抗性損失，GAN 的訓練過程可能不穩定。

Auto-regressive models assume a generation ordering which is unnatural and might restrict the model's flexibility.
自回歸模型假設生成順序不自然，可能會限制模型的靈活性。

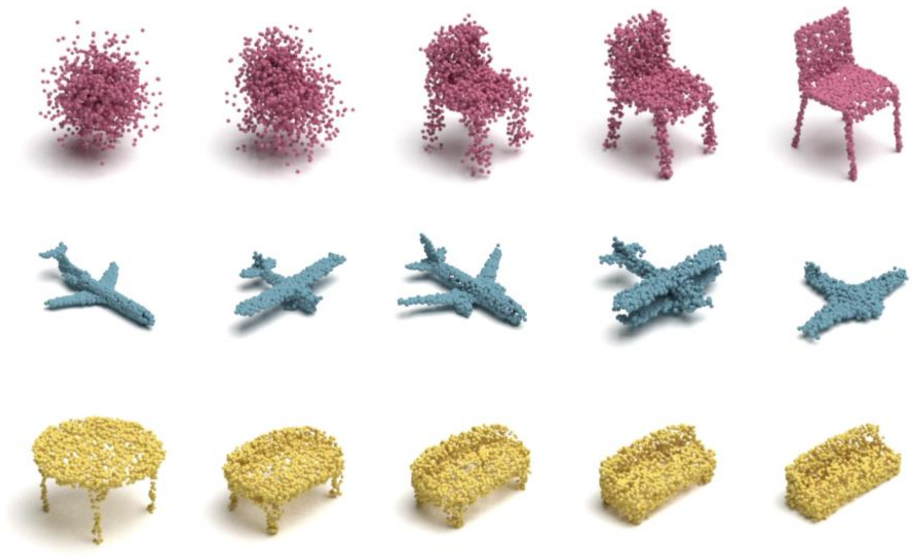

Figure 1. **Top**: The diffusion process that converts noise to some shape (left to right). **Middle**: Generated point clouds from the proposed model. **Bottom**: Latent space interpolation between the two point clouds at both ends.

Figure 1. Top: The diffusion process that converts noise to some shape (left to right). Middle: Generated point clouds from the proposed model.
圖 1. 頂部：將噪聲轉換為某種形狀的擴散過程（從左到右）。 中間：從建議的模型生成的點雲。

Bottom: Latent space interpolation between the two point clouds at both ends.
底部：兩端點雲之間的潛在空間插值。

In this paper, we propose a probabilistic generative model for point clouds inspired by non-equilibrium thermodynamics, exploiting the reverse diffusion process to learn the point distribution.
在本文中，我們提出了一種受非平衡熱力學啟發的點雲概率生成模型，利用反向擴散過程來學習點分佈。

As a point cloud is composed of discrete points in the 3D space, we regard these points as particles in a non-equilibrium thermodynamic system in contact with a heat bath.
由於點雲由 3D 空間中的離散點組成，我們將這些點視為與熱浴接觸的非平衡熱力學系統中的粒子。

Under the effect of the heat bath, the position of particles evolves stochastically in the way that they diffuse and eventually spread over the space.
在熱浴的作用下，粒子的位置以它們擴散並最終散佈在空間中的方式隨機演變。

This process is termed the diffusion process that converts the initial distribution of the particles to a simple noise distribution by adding noise at each time step [12, 20].
這個過程被稱為擴散過程，通過在每個時間步[12, 20]添加噪聲，將粒子的初始分佈轉換為簡單的噪聲分佈。

Analogously, we connect the point distribution of point clouds to a noise distribution via the diffusion process.
類似地，我們通過擴散過程將點雲的點分佈連接到噪聲分佈。

Naturally, in order to model the point distribution for point cloud generation, we consider the reverse diffusion process, which recovers the target point distribution from the noise distribution.
自然地，為了對點雲生成的點分佈進行建模，我們考慮了反向擴散過程，該過程從噪聲分佈中恢復了目標點分佈。

In particular, we model this reverse diffusion process as a Markov chain that converts the noise distribution into the target distribution.
特別是，我們將此反向擴散過程建模為將噪聲分佈轉換為目標分佈的馬爾可夫鏈。

Our goal is to learn its transition kernel such that the Markov chain can reconstruct the desired shape.
我們的目標是學習其過渡核，以便馬爾可夫鏈可以重建所需的形狀。

Further, as the purpose of the Markov chain is modeling the point distribution, the Markov chain alone is incapable to generate point clouds of various shapes.
此外，由於馬爾可夫鏈的目的是對點分佈進行建模，因此僅靠馬爾可夫鏈無法生成各種形狀的點雲。

To this end, we introduce a shape latent as the condition for the transition kernel.
為此，我們引入了一個潛在的形狀作為過渡核的條件。

In the setting of generation, the shape latent follows a prior distribution which we parameterize via normalizing flows [5, 6] for strong model expressiveness.
在生成設置中，潛在形狀遵循先驗分佈，我們通過標準化流 [5, 6] 對其進行參數化，以實現強大的模型表達能力。

In the setting of auto-encoding, the shape latent is learned end-to-end.
在自動編碼的設置中，潛在的形狀是端到端學習的。

Finally, we formulate the training objective as maximizing the variational lower bound of the likelihood of the point cloud conditional on the shape latent, which is further formulated into tractable expressions in closed form.
最後，我們將訓練目標制定為最大化以潛在形狀為條件的點雲似然的變分下界，並進一步將其表述為封閉形式的易處理表達式。

We apply our model to point cloud generation, autoencoding and unsupervised representation learning, and results demonstrate that our model achieves competitive performance on point cloud generation and auto-encoding and comparable results on unsupervised representation learning.
我們將我們的模型應用於點雲生成、自動編碼和無監督表示學習，結果表明我們的模型在點雲生成和自動編碼方面取得了有競爭力的性能，並在無監督表示學習上取得了可比的結果。

Our main contributions include:
我們的主要貢獻包括：

・ We propose a novel probabilistic generative model for point clouds, inspired by the diffusion process in nonequilibrium thermodynamics.
・ 我們提出了一種新的點雲概率生成模型，其靈感來自非平衡熱力學中的擴散過程。

・ We derive a tractable training objective from the variational lower bound of the likelihood of point clouds conditioned on some shape latent.
・ 我們從以某種潛在形狀為條件的點雲可能性的變分下界得出一個易於處理的訓練目標。

・ Extensive experiments show that our model achieves competitive performance in point cloud generation and auto-encoding.
・ 大量實驗表明，我們的模型在點雲生成和自動編碼方面取得了有競爭力的性能。

2. Related Works 相關工作

Point Cloud Generation Early point cloud generation methods [1, 7] treat point clouds as N X 3 matrices, where N is the fixed number of points, converting the point cloud generation problem to a matrix generation problem, so that existing generative models are readily applicable.
點雲生成早期的點雲生成方法 [1, 7] 將點雲視為 NX 3 矩陣，其中 N 是固定點數，將點雲生成問題轉換為矩陣生成問題，以便現有的生成模型易於應用。

For example,[7] apply variational auto-encoders [13] to point cloud generation.
例如，[7] 將變分自動編碼器 [13] 應用於點雲生成。

[1] employ generative adversarial networks [8] based on a pre-trained auto-encoder.
[1] 採用基於預訓練自動編碼器的生成對抗網絡 [8]。

The main defect of these methods is that they are restricted to generating point clouds with a fixed number of points, and lack the property of permutation invariance.
這些方法的主要缺陷是它們僅限於生成具有固定點數的點雲，並且缺乏排列不變性的特性。

FoldingNet and AtlasNet [26, 10] mitigate this issue by learning a mapping from 2D patches to the 3D space,

which deforms the 2D patches into the shape of point clouds.
FoldingNet 和 AtlasNet [26, 10] 通過學習從 2D 塊到 3D 空間的映射來緩解這個問題，這將 2D 塊變形為點雲的形狀。

These two methods allow generating arbitrary number of points by first sampling some points on the patches and then applying the mapping on them.
這兩種方法允許通過首先在補丁上採樣一些點然後在它們上應用映射來生成任意數量的點。

In addition, the points on the patches are inherently invariant to permutation.
此外，補丁上的點對排列具有固有的不變性。

The above methods rely on heuristic set distances such as the Chamfer distance (CD) and the Earth Mover's distance (EMD).
上述方法依賴於啟發式設置距離，例如倒角距離（CD）和地球移動距離（EMD）。

As pointed out in [25], CD has been shown to incorrectly favor point clouds that are overly concentrated in the mode of the marginal point distribution, and EMD is slow to compute while approximations could lead to biased gradients.
正如 [25] 中所指出的，CD 已被證明錯誤地偏向於過度集中在邊緣點分佈模式中的點雲，並且 EMD 計算速度緩慢，而近似值可能會導致有偏差的梯度。

Alternatively, point clouds can be regarded as samples from a point distribution.
或者，點雲可以被視為來自點分佈的樣本。

This viewpoint inspires exploration on applying likelihood-based methods to point cloud modeling and generation.
這一觀點激發了將基於似然的方法應用於點雲建模和生成的探索。

PointFlow [25] employs continous normalizing flows [4, 9] to model the distribution of points.
PointFlow [25] 採用連續歸一化流 [4, 9] 來模擬點的分佈。

DPF-Net [14] uses affine coupling layers as the normalizing flow to model the distribution.
DPF-Net [14] 使用仿射耦合層作為歸一化流來對分佈進行建模。

PointGrow [21] is an auto-regressive model with exact likelihoods.
PointGrow [21] 是一個具有精確似然的自回歸模型。

More recently, [2] proposed a score-matching energy-based model ShapeGF to model the distribution of points.
最近，[2] 提出了一個基於分數匹配能量的模型 ShapeGF 來模擬點的分佈。

Our method also regards point clouds as samples from a distribution, but differs in the probabilistic model compared to prior works.
我們的方法還將點雲視為來自分佈的樣本，但與之前的工作相比，概率模型有所不同。

We leverage the reverse diffusion Markov chain to model the distribution of points, achieving both simplicity and flexibility.
我們利用反向擴散馬爾可夫鏈對點的分佈進行建模，實現簡單性和靈活性。

Specifically, the training process of our model involves learning the Markov transition kernel, whose training objective has a simple function form.
具體來說，我們模型的訓練過程涉及學習馬爾可夫轉移核，其訓練目標具有簡單的函數形式。

By contrast, GAN-based models involve complex adversarial losses, continuous-flow-based methods involve expensive ODE integration.
相比之下，基於 GAN 的模型涉及復雜的對抗性損失，基於連續流的方法涉及昂貴的 ODE 集成。

In addition, our model is flexible, because it does not require invertibility in contrast to flowbased models, and does not assume ordering compared to auto-regressive models.
此外，我們的模型是靈活的，因為與基於流的模型相比，它不需要可逆性，並且與自回歸模型相比，它不假設排序。

Diffusion Probabilistic Models 擴散概率模型

The diffusion process considered in this work is related to the diffusion probabilistic model [20, 11].
在這項工作中考慮的擴散過程與擴散概率模型有關 [20, 11]。

Diffusion probabilistic models are a class of latent variable models, which also use a Markov chain to convert the noise distribution to the data distribution.
擴散概率模型是一類潛變量模型，它也使用馬爾可夫鏈將噪聲分佈轉換為數據分佈。

Prior research on diffusion probabilistic models focuses on the unconditional generation problem for toy data and images.
先前對擴散概率模型的研究側重於玩具數據和圖像的無條件生成問題。

In this work, we focus on point cloud generation, which is a conditional generation problem, because the Markov chain considered in our work generates points of a point cloud conditioned on some shape latent.
在這項工作中，我們專注於點雲生成，這是一個條件生成問題，因為我們工作中考慮的馬爾可夫鏈生成點雲的點以某種潛在形狀為條件。

This conditional adaptation leads to significantly different training and sampling schemes compared to prior research on diffusion probabilistic models.

與先前對擴散概率模型的研究相比，這種有條件的適應導致了顯著不同的訓練和採樣方案。

3. Diffusion Probabilistic Models for Point Clouds 點雲的擴散概率模型

In this section, we first formulate the probabilistic model of both the forward and the reverse diffusion processes for point clouds.
在本節中，我們首先制定點雲正向和反向擴散過程的概率模型。

Then, we formulate the objective for training the model. The implementation of the model will be provided in the next section.
然後，我們制定訓練模型的目標。 該模型的實現將在下一節中提供。

3.1. Formulation 公式

We regard a point cloud X(0) = {x(0)i}Ni=1 consisting of N points as a set of particles in an evolving thermodynamic system.
我們將一個由 N 個點組成的點雲 X(0) = {x(0)i}Ni=1 視為演化熱力學系統中的一組粒子。

As discussed in Section 1, each point xi in the point cloud can be treated as being sampled independently from a point distribution, which we denote as q(x(0)i|z).
如第 1 節所述，點雲中的每個點 xi 都可以被視為獨立於點分佈進行採樣，我們將其表示為 q(x(0)i|z)。

Here, z is the shape latent that determines the distribution of points.
這裡，z 是決定點分佈的潛在形狀。

Physically, as time goes by, the points gradually diffuse into a chaotic set of points.
在物理上，隨著時間的推移，這些點逐漸擴散成一組混亂的點。

This process is termed the diffusion process, which converts the original meaningful point distribution into a noise distribution.
這個過程稱為擴散過程，它將原始有意義的點分佈轉化為噪聲分佈。

The forward diffusion process is modeled as a Markov chain [12]:
前向擴散過程被建模為馬爾可夫鏈 [12]：

$$q(\boldsymbol{x}_i^{(1:T)}|\boldsymbol{x}_i^{(0)}) = \prod_{t=1}^{T} q(\boldsymbol{x}_i^{(t)}|\boldsymbol{x}_i^{(t-1)}), \qquad (1)$$

where q(x(t) i |x(t-1) i) is the Markov diffusion kernel.
其中 q(x(t) i |x(t-1) i) 是馬爾可夫擴散核。

$$q(\boldsymbol{x}_i^{(t)}|\boldsymbol{x}_i^{(t-1)})$$

The kernel adds noise to points at the previous time step and models the distribution of points at the next time step.
內核在上一個時間步向點添加噪聲，並在下一個時間步對點的分佈進行建模。

Following the convention of [20], we define the diffusion kernel as:
遵循[20]的約定，我們將擴散核定義為：

$$q(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(t-1)}) = \mathcal{N}(\boldsymbol{x}^{(t)}|\sqrt{1-\beta_t}\boldsymbol{x}^{(t-1)}, \beta_t \boldsymbol{I}), t = 1, ..., T, \quad (2)$$

where β1, ..., βT are variance schedule hyper-parameters that control the diffusion rate of the process.
其中 β1, …, βT 是控製過程擴散速率的方差調度超參數。

Our goal is to generate point clouds with a meaningful shape, encoded by the latent representation z.
我們的目標是生成具有有意義形狀的點雲，由潛在表示 z 編碼。

We treat the generation process as the reverse of the diffusion process, where points sampled from a simple noise distribution p(x(T)i) that approximates q(x(T)i) are given as the input.
我們將生成過程視為擴散過程的逆過程，其中從近似 q(x(T)i) 的簡單噪聲分佈 p(x(T)i) 採樣的點作為輸入給出。

Then, the points are passed through the reverse Markov chain and finally form the desired shape.
然後，點通過逆馬爾可夫鏈，最終形成所需的形狀。

Unlike the forward diffusion process that simply adds noise to the points, the reverse process aims to recover the desired shape from the input noise, which requires training from data.
與簡單地向點添加噪聲的前向擴散過程不同，反向過程旨在從輸入噪聲中恢復所需的形狀，這需要從數據中進行訓練。

We formulate the reverse diffusion process for generation as:
我們將生成的反向擴散過程公式化為：

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(0:T)}|\boldsymbol{z}) = p(\boldsymbol{x}^{(T)}) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)}, \boldsymbol{z}), \qquad (3)$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)}, \boldsymbol{z}) = \mathcal{N}\big(\boldsymbol{x}^{(t-1)}\big|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t)}, t, \boldsymbol{z}), \beta_t \boldsymbol{I}\big), \quad (4)$$

where μθ is the estimated mean implemented by a neural network parameterized by θ.
其中 μθ 是由 θ 參數化的神經網絡實現的估計均值。

z is the latent encoding the target shape of the point cloud.
z 是點雲目標形狀的潛在編碼。

The starting distribution p(x(T)i) is set to a standard normal distribution N(0, I).
起始分佈 p(x(T)i) 設置為標準正態分佈 N(0, I)。

Given a shape latent z, we obtain the point cloud with the target shape by passing a set of points sampled from p(x(T) i) through the reverse Markov chain.
給定一個形狀潛在 z，我們通過將一組從 p(x(T) i) 採樣的點通過反向馬爾可夫鏈來獲得具有目標形狀的點雲。

For the sake of brevity, in the following sections, we use the distribution with respect to the entire point cloud X(0).
為簡潔起見，在以下部分中，我們使用關於整個點雲 X(0) 的分佈。

Since the points in a point cloud are independently sampled from a distribution, the probability of the whole point cloud is simply the product of the probability of each point:
由於點雲中的點是從分佈中獨立採樣的，因此整個點雲的概率只是每個點的概率的乘積：

$$q(\boldsymbol{X}^{(1:T)}|\boldsymbol{X}^{0}) = \prod_{i=1}^{N} q(\boldsymbol{x}_i^{(1:T)}|\boldsymbol{x}_i^{(0)}), \qquad (5)$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0:T)}|\boldsymbol{z}) = \prod_{i=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(0:T)}|\boldsymbol{z}). \qquad (6)$$

Having formulated the forward and reverse diffusion processes for point clouds, we will formalize the training objective of the reverse diffusion process for point cloud generation as follows.
制定了點雲的正向和反向擴散過程後，我們將點雲生成的反向擴散過程的訓練目標形式化如下。

3.2. Training Objective 培訓目標

The goal of training the reverse diffusion process is to maximize the log-likelihood of the point cloud: E[log pθ(X(0))].
訓練反向擴散過程的目標是最大化點雲的對數似然：E[log p θ (X(0))]。

$$\mathbb{E}[\log p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0)})].$$

However, since directly optimizing the exact log-likelihood is intractable, we instead maximize its variational lower bound:
然而，由於直接優化精確對數似然是棘手的，我們改為最大化其變分下界：

$$\begin{aligned}
\mathbb{E}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0)})\right] &\geq \mathbb{E}_q\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0:T)}, \boldsymbol{z})}{q(\boldsymbol{X}^{(1:T)}, \boldsymbol{z}|\boldsymbol{X}^{(0)})}\right] \\
&= \mathbb{E}_q\Big[\log p(\boldsymbol{X}^{(T)}) \\
&\qquad + \sum_{t=1}^{T} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(t-1)}|\boldsymbol{X}^{(t)}, \boldsymbol{z})}{q(\boldsymbol{X}^{(t)}|\boldsymbol{X}^{(t-1)})} \\
&\qquad - \log \frac{q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)})}{p(\boldsymbol{z})}\Big].
\end{aligned} \tag{7}$$

The above variational bound can be adapted into the training objective L to be minimized (the detailed derivation is provided in the supplementary material):
上面的變分界可以適配到要最小化的訓練目標 L 中（詳細推導在補充材料中提供）：

$$\begin{aligned}
L(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_q\Big[&\sum_{t=2}^{T} D_{\mathrm{KL}}\big(q(\boldsymbol{X}^{(t-1)}|\boldsymbol{X}^{(t)}, \boldsymbol{X}^{(0)}) \| \\
&\qquad\qquad p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(t-1)}|\boldsymbol{X}^{(t)}, \boldsymbol{z})\big) \\
&- \log p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0)}|\boldsymbol{X}^{(1)}, \boldsymbol{z}) \\
&+ D_{\mathrm{KL}}\big(q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)}) \| p(\boldsymbol{z})\big)\Big].
\end{aligned} \tag{8}$$

Since the distributions of points are independent of each other as described in Eq. (5), we further expand the training objective:
由於點的分佈彼此獨立，如 Eq. (5) 中所述，我們進一步擴展了訓練目標：

$$L(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_q\Big[\sum_{t=2}^{T}\sum_{i=1}^{N} D_{\mathrm{KL}}\big(\underbrace{q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)})}_{①} \,\|\, \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z})}_{②}\big) - \sum_{i=1}^{N} \underbrace{\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(0)}|\boldsymbol{x}_i^{(1)}, \boldsymbol{z})}_{③} + D_{\mathrm{KL}}\big(\underbrace{q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)})}_{④} \,\|\, \underbrace{p(\boldsymbol{z})}_{⑤}\big)\Big]. \tag{9}$$

The training objective can be optimized efficiently since each of the terms on the right hand side is tractable and q is easy to sample from the forward diffusion process.
訓練目標可以有效地優化，因為右側的每個項都很容易處理，而且 q 很容易從前向擴散過程中採樣。

Next, we elaborate on the terms to reveal how to compute the objective.
接下來，我們詳細說明術語以揭示如何計算目標。

① q(x(t-1)i |x(t)i ; x(0)i) can be computed with the following closed-form Gaussian according to [11]:
① q(x(t-1)i |x(t)i ; x(0)i) 可以根據[11]用以下閉合形式高斯計算：

$$q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)}) = \mathcal{N}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{\mu}_t(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)}), \gamma_t \boldsymbol{I}), \tag{10}$$

where, using the notation αt = 1 −βt and α-t … :
其中，使用符號 αt 和 α-t … ：其中，使用符號

$$\alpha_t = 1 - \beta_t$$

$$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

②,③ pθ (x(t-1)i |x(t)i ,z)(t = 1, … ,T) are trainable Gaussians as described in Eq. (4).
②,③ pθ(x(t-1)i |x(t)i ,z)(t = 1, … , T) 是可訓練的高斯分佈，如等式。 (4).

④qϕ(z|X(0)) is the approximate posterior distribution.
④qφ(z|X(0)) 是近似後驗分佈。

Using the language of variational auto-encoders, qφ (z|X(0)) is an encoder that encodes the input point cloud X(0) into the distribution of the latent code z.
使用變分自動編碼器的語言，qφ (z|X(0)) 是一個編碼器，它將輸入點雲 X(0) 編碼為潛在代碼 z 的分佈。

We assume it as a Gaussian following the convention:
我們假設它是遵循約定的高斯函數：

$$q(\boldsymbol{z}|\boldsymbol{X}^{(0)}) = \mathcal{N}\big(\boldsymbol{z}|\boldsymbol{\mu}_{\varphi}(\boldsymbol{X}^{(0)}), \boldsymbol{\Sigma}_{\varphi}(\boldsymbol{X}^{(0)})\big). \qquad (12)$$

⑤ The last term p(z) is the prior distribution.
⑤ 最後一項 p(z) 是先驗分佈。

The most common choice of p(z) is the isotropic Gaussian N(0, I).
p(z) 的最常見選擇是各向同性高斯 N(0, I)。

In addition to a fixed distribution, the prior can be a trainable parametric distribution, which is more flexible.
除了固定分佈之外，先驗還可以是可訓練的參數分佈，更加靈活。

For example, normalizing flows [5, 6] can be employed to parameterize the prior distribution.
例如，可以使用歸一化流 [5, 6] 來參數化先驗分佈。

In the following section, we show how to optimize the objective in Eq. (9) in order to train the model.
在下一節中，我們將展示如何優化 Eq. (9) 中的目標以訓練模型。

3.3. Training Algorithm 訓練演算法

In principle, training the model amounts to minimizing the objective in Eq. (9).
原則上，訓練模型相當於最小化 Eq. (9) 中的目標。

However, evaluating Eq. (9) requires summing the expectation of the KL terms over all the time steps, which involves sampling a full trajectory x(1)I, …, x(T)i from the forward diffusion process in order to compute the expectation.
然而，評估 Eq. (9) 需要在所有時間步長上對 KL 項的期望求和，這涉及從前向擴散過程中採樣完整的軌跡 x(1)I, …, x(T)i 以計算期望。

To make the training simpler and more efficient, following [11], instead of evaluating the expectation of the whole summation over all the time steps in Eq. (9), we randomly choose one term from the summation to optimize at each training step.

為了使訓練更簡單、更高效，按照 [11]，我們不是在 Eq. (9) 中的所有時間步長上評估整個總和的期望，而是從總和中隨機選擇一項以在每個訓練步進行優化。

Specifically, this simplified training algorithm is as follows:
具體來說，這個簡化的訓練演算法如下：

Algorithm 1 Training (Simplified)

repeat
  Sample $\boldsymbol{X}^{(0)} \sim q_{\text{data}}(\boldsymbol{X}^{(0)})$
  Sample $\boldsymbol{z} \sim q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)})$
  Sample $t \sim \text{Uniform}(\{1, \ldots, T\})$
  Sample $\boldsymbol{x}_1^{(t)}, \ldots, \boldsymbol{x}_N^{(t)} \sim q(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(0)})$
  $L_t \leftarrow \sum_{i=1}^{N} D_{\text{KL}}\left(q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)}) \middle\| p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z})\right)$
  $L_{\boldsymbol{z}} \leftarrow D_{\text{KL}}(q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)}) \| p(\boldsymbol{z}))$
  Compute $\nabla_{\boldsymbol{\theta}}(L_t + \frac{1}{T} L_{\boldsymbol{z}})$. Then perform gradient descent.
until converged

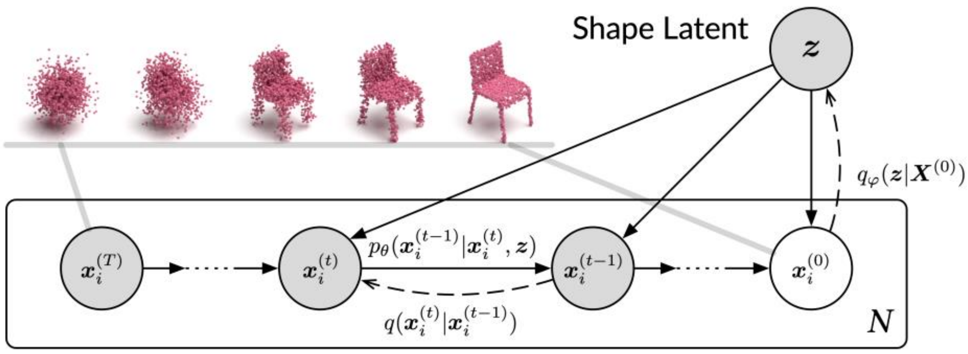

Figure 2. The directed graphical model of the diffusion process for point clouds. N is the number of points in the point cloud $\boldsymbol{X}^{(0)}$.

Figure 2. The directed graphical model of the diffusion process for point clouds. N is the number of points in the point cloud X(0).
圖 2. 點雲擴散過程的有向圖模型。 N 是點雲 X(0) 中的點數。

Algorithm 1 Training (Simplified)
算法 1 訓練（簡化版）

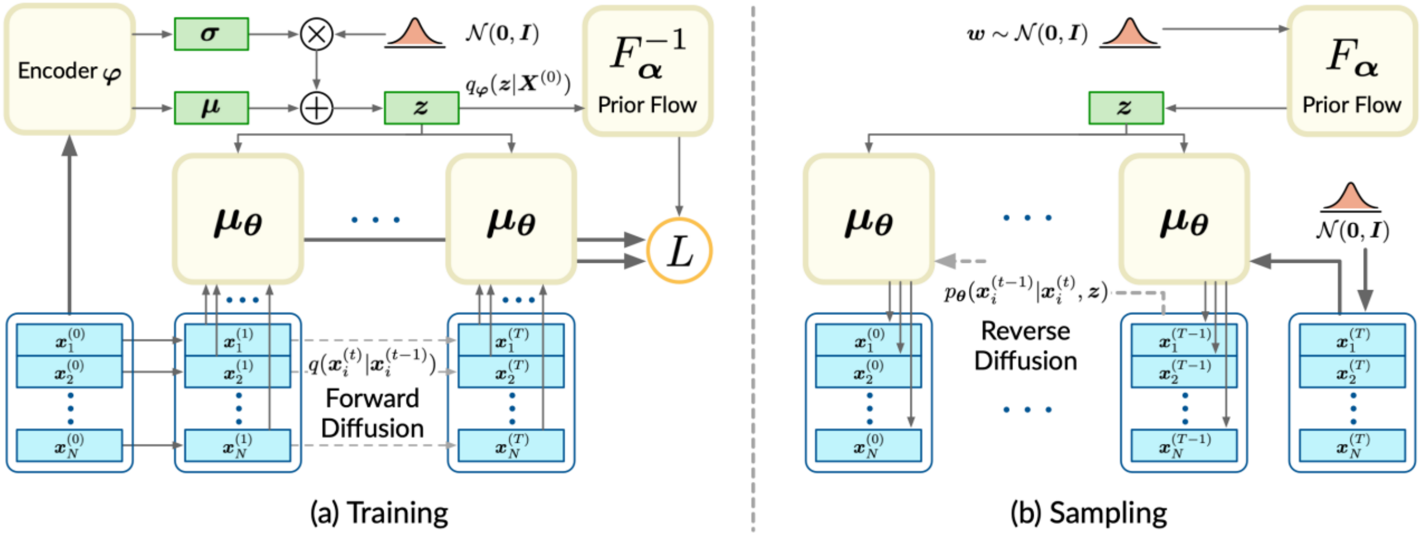

Figure 3. The illustration of the proposed model.
圖 3. 建議模型的說明。

(a) illustrates how the objective is computed during the training process.
(a) 說明了在訓練過程中如何計算目標。

(b) illustrates the generation process.
(b) 說明了生成過程。

To efficiently sample from q(x(t)|x(0)) (5th statement) and avoid iterative sampling starting from t = 1, we leverage on the result in [11], which shows that q(x(t)|x(0)) is a Gaussian:
為了有效地從 q(x(t)|x(0))（第 5 條語句）採樣並避免從 t = 1 開始迭代採樣，我們利用 [11] 中的結果，這表明 q(x(t)| x(0)) 是高斯分佈：

$$q(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(0)}) = \mathcal{N}(\boldsymbol{x}^{(t)}|\sqrt{\bar{\alpha}_t}\boldsymbol{x}^{(0)}, (1-\bar{\alpha}_t)\boldsymbol{I}). \qquad (13)$$

The gaussianity of q(x(t)|x(0)) makes further simplification on Lt (6th statement) possible by using the reparameterization trick [13].
q(x(t)|x(0)) 的高斯性通過使用重新參數化技巧 [13] 可以進一步簡化 Lt（第 6 條語句）。

We put the detail of this simplification to the supplementary material.
我們將這種簡化的細節放在補充材料中。

Last, note that the KL divergence in Lz is evaluated stochastically by -Ez~qφ(z|X(0))[p(z)]- H[q'(z|X(0))].
最後，請注意 Lz 中的 KL 散度由 -Ez~qφ(z|X(0))[p(z)]-H[q'(z|X(0))] 隨機評估。

4. Model Implementations 模型實現

The general training objective and algorithm in the previous section lay the foundation for the formulation of specific point cloud tasks.
上一節的一般訓練目標和演算法為具體點雲任務的製定奠定了基礎。

Next, we adapt the training objective to point cloud generation and point cloud auto-encoding respectively.
接下來，我們分別將訓練目標調整為點雲生成和點云自動編碼。

4.1. Point Cloud Generator 點雲生成器

Leveraging on the model in Section 3, we propose a probabilistic model for point cloud generation by employing normalizing flows to parameterize the prior distribution p(z), which makes the model more flexible [18, 5].
利用第 3 節中的模型，我們通過使用歸一化流來參數化先驗分佈 p(z)，提出了一種用於點雲生成的概率模型，這使得模型更加靈活 [18, 5]。

Specifically, we use a stack of affine coupling layers [6] as the normalizing flow.
具體來說，我們使用一堆仿射耦合層 [6] 作為歸一化流程。

In essence, the affine coupling layers provide a trainable bijector F that maps an isotropic Gaussian to a complex distribution.
本質上，仿射耦合層提供了一個可訓練的雙射器 F，它將各向同性高斯分佈映射到復雜分佈。

Since the mapping is bijective, the exact probability of the target distribution can be computed by the change-of-variable formula:
由於映射是雙射的，目標分佈的準確概率可以通過變量變化公式計算：

$$p(\boldsymbol{z}) = p_{\boldsymbol{w}}(\boldsymbol{w}) \cdot \left| \det \frac{\partial F_{\boldsymbol{\alpha}}}{\partial \boldsymbol{w}} \right|^{-1} \quad \text{where} \quad \boldsymbol{w} = F_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{z}). \tag{14}$$

Here, p(z) is the prior distribution in the model, F α is the trainable bijector implemented by the affine coupling layers, and pw(w) is the isotropic Gaussian N(0, I).
這裡，p(z) 是模型中的先驗分佈，F α 是仿射耦合層實現的可訓練雙射器，pw(w) 是各向同性高斯 N(0, I)。

As for the encoder q ϕ (z|X(0)), we adopt PointNet [16] as the architecture for μϕ and Σϕ of the encoder qϕ(z|X(0)).
對於編碼器 q φ (z|X(0))，我們採用 PointNet [16] 作為編碼器 qφ(z|X(0)) 的 μφ 和 Σφ 的架構。

Substituting Eq. (14) into Eq. (9), the training objective for the generative model is:
將 Eq. (14) 代入 Eq. (9) ，生成模型的訓練目標為：

$$L_G(\boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{\alpha}) = \mathbb{E}_q\Big[\sum_{t=2}^{T}\sum_{i=1}^{N} D_{\mathrm{KL}}\big(q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)}) \| p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z})\big) - \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(0)}|\boldsymbol{x}_i^{(1)}, \boldsymbol{z}) + D_{\mathrm{KL}}\Big(q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)}) \Big\| p_{\boldsymbol{w}}(\boldsymbol{w}) \cdot \Big|\det \frac{\partial F_{\boldsymbol{\alpha}}}{\partial \boldsymbol{w}}\Big|^{-1}\Big)\Big]. \tag{15}$$

The algorithm for optimizing the above objective can be naturally derived from Algorithm 1.
優化上述目標的算法自然可以從演算法 1 推導出來。

To sample a point cloud, we first draw w ~N(0, I) and pass it through F α to acquire the shape latent z.
為了對點雲進行採樣，我們首先繪製 w ~N(0, I) 並將其通過 F α 以獲取形狀潛在 z。

Then, with the shape latent z, we sample some points {x(T) i} from the noise distribution p(x(T)) and pass the points through the reverse Markov chain pθ(x(0:T)i|z) defined in Eq. (3) to generate the point cloud X(0) = {x(0)i}Ni=1.
然後，使用形狀潛在 z，我們從噪聲分佈 p(x(T)) 中採樣一些點 {x(T) i} 並將這些點通過逆馬爾可夫鏈 p θ (x(0:T)i|z) 等式中定義。 (3) 生成點雲 X(0) = {x(0)i}Ni=1。

4.2. Point Cloud Auto-Encoder 點云自動編碼器

We implement a point cloud auto-encoder based on the probabilistic model in Section 3.
我們基於第 3 節中的概率模型實現了一個點云自動編碼器。

We employ the PointNet as the representation encoder, denoted as E φ (X(0)) with parameters ϕ, and leverage the reverse diffusion process presented in Section 3.1 for decoding, conditioned on the latent code produced by the encoder.
我們使用 PointNet 作為表示編碼器，用參數 φ 表示為 E φ (X(0))，並利用第 3.1 節中介紹的反向擴散過程進行解碼，條件是編碼器產生的潛在代碼。

Table 1. Comparison of point cloud generation performance. CD is multiplied by 10^3, EMD is multiplied by 10^1, and JSD is multiplied by 10^3.

Shape	Model	MMD (↓)		COV (%, ↑)		1-NNA (%, ↓)		JSD (↓)
		CD	EMD	CD	EMD	CD	EMD	-
Airplane	PC-GAN [1]	3.819	1.810	42.17	13.84	77.59	98.52	6.188
	GCN-GAN [22]	4.713	1.650	39.04	18.62	89.13	98.60	6.669
	TreeGAN [19]	4.323	1.953	39.37	8.40	83.86	99.67	15.646
	PointFlow [25]	3.688	1.090	44.98	44.65	66.39	**69.36**	1.536
	ShapeGF [2]	**3.306**	**1.027**	**50.41**	**47.12**	**61.94**	**70.51**	**1.059**
	Ours	**3.276**	**1.061**	**48.71**	**45.47**	**64.83**	75.12	**1.067**
	Train	3.917	1.003	51.73	54.04	48.85	50.82	0.809
Chair	PC-GAN [1]	13.436	3.104	46.23	22.14	69.67	100.00	6.649
	GCN-GAN [22]	15.354	2.213	39.84	35.09	77.86	95.80	21.708
	TreeGAN [19]	14.936	3.613	38.02	6.77	74.92	100.00	13.282
	PointFlow [25]	13.631	1.856	41.86	43.38	66.13	**68.40**	12.474
	ShapeGF [2]	**13.175**	**1.785**	**48.53**	**46.71**	**56.17**	**62.69**	**5.996**
	Ours	**12.276**	**1.784**	**48.94**	**47.52**	**60.11**	69.06	**7.797**
	Train	13.954	1.756	53.29	54.90	49.14	48.28	3.602

Table 1. Comparison of point cloud generation performance.
表 1. 點雲生成性能比較。

CD is multiplied by 10^3, EMD is multiplied by 10^1, and JSD is multiplied by 10^3.
CD 乘以 10^3，EMD 乘以 10^1，JSD 乘以 10^3。

Leveraging on Eq. (9), we train the auto-encoder by minimizing the following adapted objective:
利用 Eq. (9)，我們通過最小化以下自適應目標來訓練自動編碼器：

$$L(\boldsymbol{\theta},\boldsymbol{\varphi}) = \mathbb{E}_q\Big[\sum_{t=2}^{T}\sum_{i=1}^{N} D_{\mathrm{KL}}\big(q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)},\boldsymbol{x}_i^{(0)})\| \; p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)},E_{\boldsymbol{\varphi}}(\boldsymbol{X}^{(0)}))\big) - \sum_{i=1}^{N}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(0)}|\boldsymbol{x}_i^{(1)},E_{\boldsymbol{\varphi}}(\boldsymbol{X}^{(0)}))\Big]. \tag{16}$$

To decode a point cloud that is encoded as the latent code z, we sample some points {x(T) i} from the noise distribution p(x(T) i) and pass the points through the reverse Markov chain p(x(0:T) i |z) defined in Eq. (3) to acquire the reconstructed point cloud X(0) = {x(0) i }Ni =1.
為了解碼編碼為潛在代碼 z 的點雲，我們從噪聲分佈 p(x(T) i) 中採樣一些點 {x(T) i} 並將這些點通過反向馬爾可夫鏈 p(x(0:T) i |z) 在 Eq. (3) 中定義以獲取重建的點雲 X(0) = {x(0) i }Ni =1。

5. Experiments 實驗

In this section, we evaluate our model's performance on three tasks: point cloud generation, auto-encoding, and unsupervised representation learning.
在本節中，我們評估模型在三個任務上的性能：點雲生成、自動編碼和無監督表示學習。

5.1. Experimental Setup 實驗裝置

Datasets 數據集

For generation and auto-encoding tasks, we employ the ShapeNet dataset [3] containing 51,127 shapes from 55 categories.
對於生成和自動編碼任務，我們採用了包含來自 55 個類別的 51,127 個形狀的 ShapeNet 數據集 [3]。

The dataset is randomly split into training, testing and validation sets by the ratio 80%, 15%, 5% respectively.
數據集被隨機分成訓練集、測試集和驗證集，比例分別為 80%、15%、5%。

For the representation learning task, we use the training split of ShapeNet to train an encoder.
對於表示學習任務，我們使用 ShapeNet 的訓練分割來訓練編碼器。

Then we adopt ModelNet10 and ModelNet40 [24] to evaluate the representations learned by the encoder.
然後我們採用 ModelNet10 和 ModelNet40 [24] 來評估編碼器學習的表示。

We sample 2048 points from each of the shape to acquire the point clouds and normalize each of them to zero mean and unit variance.
我們從每個形狀中採樣 2048 個點以獲取點雲並將它們中的每一個歸一化為零均值和單位方差。

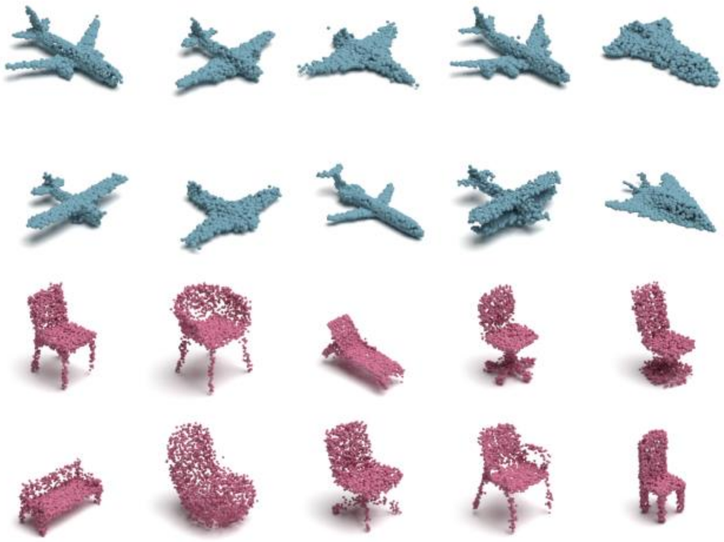

Figure 4. Examples of point clouds generated by our model.

Figure 4. Examples of point clouds generated by our model.
圖 4. 我們的模型生成的點雲示例。

Table 2. Comparison of point cloud auto-encoding performance. Atlas (S1) and Atlas (P25) denote 1-sphere and 25-square variants of AtlasNet respectively. CD is multiplied by 10^3 and EMD is multiplied by 10^2.

Dataset	Metric	Atlas (S1)	Altas (P25)	PointFlow	ShapeGF	Ours	Oracle
Airplane	CD	2.000	**1.795**	2.420	2.102	2.118	1.016
	EMD	4.311	4.366	3.311	3.508	**2.876**	2.141
Car	CD	6.906	6.503	5.828	**5.468**	5.493	3.917
	EMD	5.617	5.408	4.390	4.489	**3.937**	3.246
Chair	CD	5.479	**4.980**	6.795	5.146	5.677	3.221
	EMD	5.550	5.282	5.008	4.784	**4.153**	3.281
ShapeNet	CD	5.873	5.420	7.550	5.725	**5.252**	3.074
	EMD	5.457	5.599	5.172	5.049	**3.783**	3.112

Table 2. Comparison of point cloud auto-encoding performance. Atlas (S1) and Atlas (P25) denote 1-sphere and 25-square variants of AtlasNet respectively.
表 2. 點云自動編碼性能比較。 Atlas (S1) 和 Atlas (P25) 分別表示 AtlasNet 的 1-sphere 和 25-square 變體。

CD is multiplied by 10^3 and EMD is multiplied by 10^2.
CD 乘以 103，EMD 乘以 102。

Evaluation Metrics 評估指標

Following prior works, we use the Chamfer Distance (CD) and the Earth Mover's Distance (EMD) to evaluate the reconstruction quality of the point clouds [1].
根據先前的工作，我們使用倒角距離 (CD) 和地球移動距離 (EMD) 來評估點雲的重建質量 [1]。

To evaluate the generation quality, we employ the minimum matching distance (MMD), the coverage score (COV), 1-NN classifier accuracy (1-NNA) and the Jenson-Shannon divergence (JSD) [25].
為了評估生成質量，我們採用最小匹配距離 (MMD)、覆蓋率 (COV)、1-NN 分類器準確度 (1-NNA) 和 Jenson-Shannon 散度 (JSD) [25]。

The MMD score measures the fidelity of the generated samples and the COV score detects mode-collapse.
MMD 分數衡量生成樣本的保真度，COV 分數檢測模式崩潰。

The 1-NNA score is computed by testing the generated samples and the reference samples by a 1-NN classifier.
1-NNA 分數是通過 1-NN 分類器測試生成的樣本和參考樣本來計算的。

If the performance of the classifier is close to random guess, i.e., the accuracy is close to 50%, the quality of generated samples can be considered better.
如果分類器的性能接近隨機猜測，即準確率接近 50%，則可以認為生成樣本的質量更好。

The JSD score measures the similarity between the point distributions of the generated set and the reference set.
JSD 分數衡量生成集和參考集的點分佈之間的相似性。

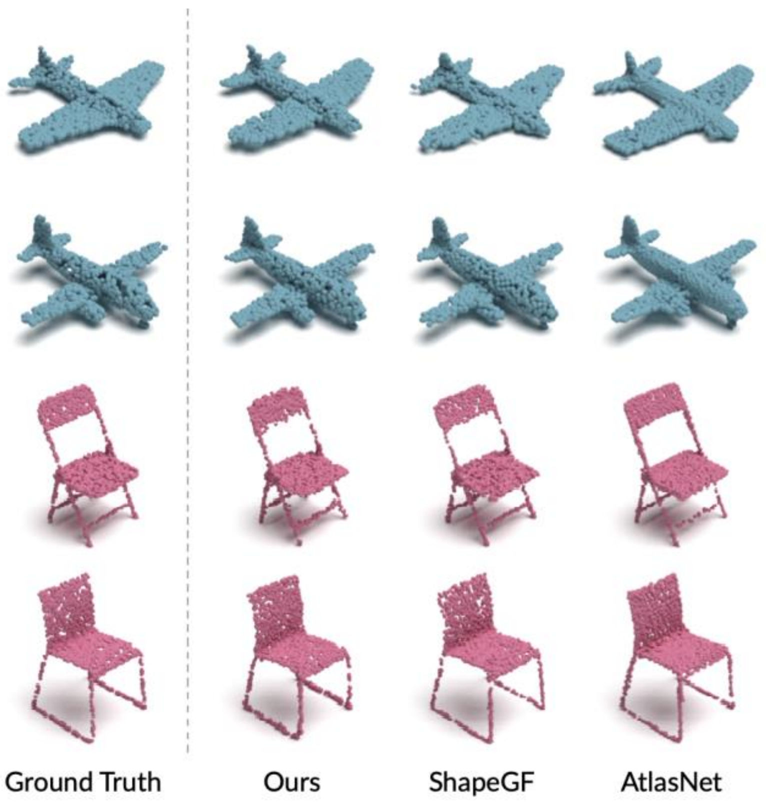

Figure 5. Examples of reconstructed point clouds from different auto-encoders.

Figure 5. Examples of reconstructed point clouds from different auto-encoders.

圖 5. 來自不同自動編碼器的重建點雲示例。

Table 3. Comparison of representation learning in linear SVM classification accuracy.
表 3. 線性 SVM 分類精度中表徵學習的比較。

Table 3. Comparison of representation learning in linear SVM classification accuracy.

Model	ModelNet10	ModelNet40
AtlasNet [10]	91.9	86.6
PC-GAN (CD) [1]	**95.4**	84.5
PC-GAN (EMD) [1]	**95.4**	84.0
PointFlow [25]	93.7	86.8
ShapeGF [2]	90.2	84.6
Ours	94.2	**87.6**

5.2. Point Cloud Generation 點雲生成

We quantitatively compare our method with the following state-of-the-art generative models: PC-GAN [1], GCNGAN [22], TreeGAN [19], PointFlow [25] and ShapeGF [2] using point clouds from two categories in ShapeNet:
我們將我們的方法與以下最先進的生成模型進行定量比較：PC-GAN [1]、GCNGAN [22]、TreeGAN [19]、PointFlow [25] 和 ShapeGF [2] 使用來自兩個類別的點雲 在 ShapeNet 中：

airplane and chair. 飛機和椅子。

Following ShapeGF [2], when evaluating each of the model, we normalize both generated point clouds and reference point clouds into a bounding box of [-1; 1]^3, so that the metrics focus on the shape of point clouds but not the scale.
遵循 ShapeGF [2]，在評估每個模型時，我們將生成的點雲和參考點雲都歸一化為一個邊界框 [?1; 1]3，因此指標關注點雲的形狀而不是規模。

We evaluate the point clouds generated by the models using the metrics in Section 5.1 and summarize the results in Table 1.
我們使用第 5.1 節中的指標評估模型生成的點雲，並將結果總結在表 1 中。

We also visualize some generated point cloud samples from our method in Figure 4.
我們還從圖 4 中的方法中可視化了一些生成的點雲樣本。

5.3. Point Cloud Auto-Encoding 點雲自動編碼

We evaluate the reconstruction quality of the proposed auto-encoder, with comparisons against state-of-theart point cloud auto-encoders:
我們評估了所提出的自動編碼器的重建質量，並與最先進的點云自動編碼器進行了比較：

AtlasNet [10], PointFlow [25] and ShapeGF [2].
AtlasNet [10]、PointFlow [25] 和 ShapeGF [2]。

Four datasets are used in the evaluation, which include three categories in ShapeNet:
評估中使用了四個數據集，其中包括 ShapeNet 中的三個類別：

airplane, car, chair and the whole ShapeNet.
飛機、汽車、椅子和整個 ShapeNet。

We also report the lower bound “oracle” of the reconstruction errors.
我們還報告了重建錯誤的下限“oracle”。

This bound is obtained by computing the distance between two different point clouds with the same number of points and the identical shape.
這個界限是通過計算具有相同點數和相同形狀的兩個不同點雲之間的距離來獲得的。

As shown in Table 2, our method outperforms other methods when measured by EMD, and pushes closer towards the lower bounded “oracle” performance.
如表 2 所示，當用 EMD 衡量時，我們的方法優於其他方法，並且更接近下限的“oracle”性能。

The CD score of our method is comparable to others.
我們方法的 CD 分數與其他方法相當。

Notably, when trained and tested on the whole ShapeNet dataset, our model outperforms others in both CD and EMD, which suggests that our model has higher capacity to encode different shapes.
值得注意的是，當在整個 ShapeNet 數據集上進行訓練和測試時，我們的模型在 CD 和 EMD 上都優於其他模型，這表明我們的模型具有更高的編碼不同形狀的能力。

Also, the visualization of reconstructed point clouds in Figure 5 validates the effectiveness of our model.
此外，圖 5 中重建點雲的可視化驗證了我們模型的有效性。

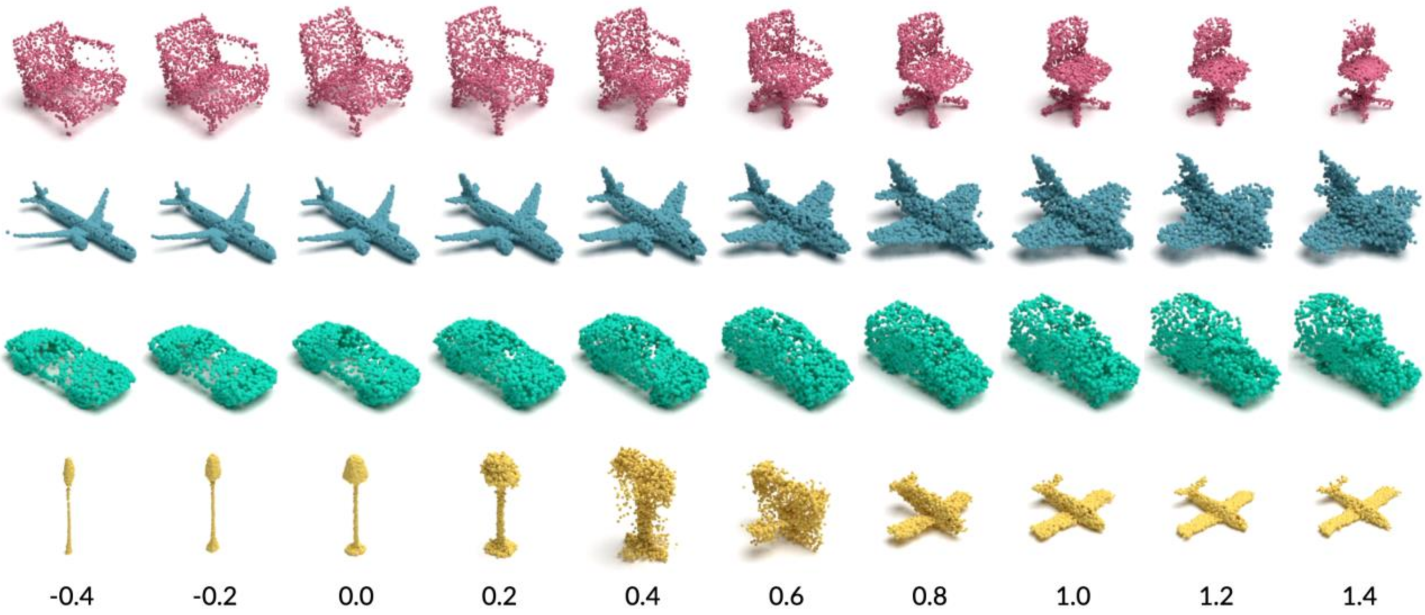

Figure 6. Latent space interpolation and extrapolation.

Figure 6. Latent space interpolation and extrapolation.

圖 6. 潛在空間內插和外推。

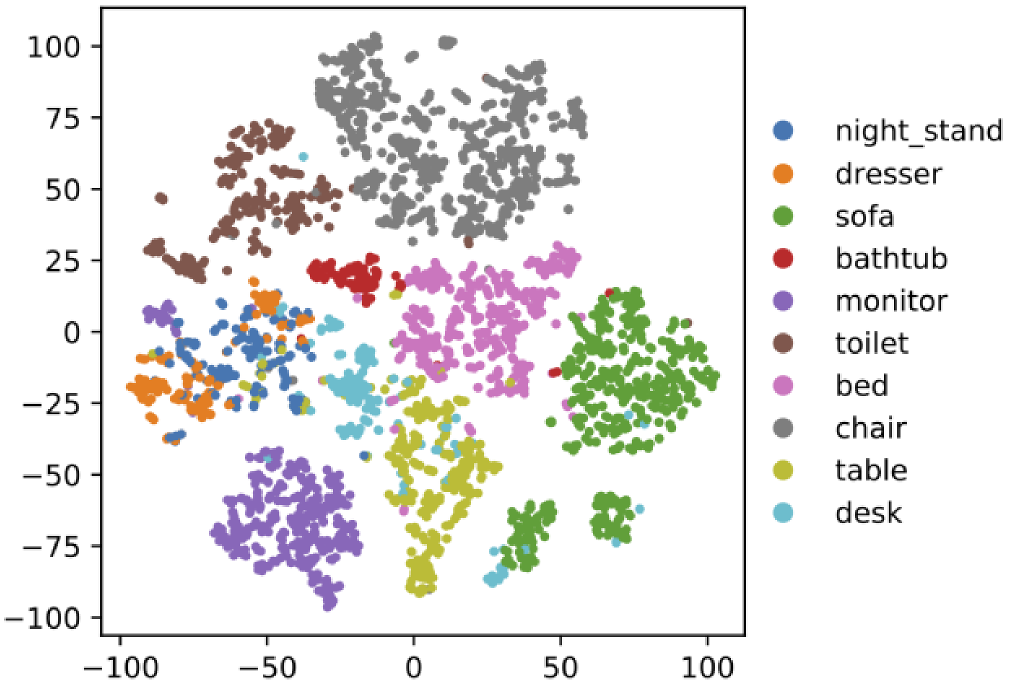

Figure 7. The t-SNE clustering visualization of latent codes obtained from the encoder.

Figure 7. The t-SNE clustering visualization of latent codes obtained from the encoder.

圖 7. 從編碼器獲得的潛在程式碼的 t-SNE 集群視覺化。

5.4. Unsupervised Representation Learning 無監督表示學習

Further, we evaluate the representation learned by our auto-encoder.
此外，我們評估我們的自動編碼器學習的表示。

Firstly, we train an auto-encoder with the whole ShapeNet dataset.
首先，我們用整個 ShapeNet 數據集訓練一個自動編碼器。

During the training, we augment point clouds by applying random rotations along the gravity axis, which follows previous works.
在訓練期間，我們通過沿重力軸應用隨機旋轉來增強點雲，這遵循了之前的工作。

Then, we learn the feature representations of point clouds in ModelNet-10 and ModelNet-40 using the trained encoder, and train a linear SVM using the codes of point clouds in the training split and their categories.
然後，我們使用訓練好的編碼器學習 ModelNet-10 和 ModelNet-40 中點雲的特徵表示，並使用訓練分割中的點雲代碼及其類別訓練線性 SVM。

Finally, we test the SVM using the testing split and report the accuracy in Table 3.
最後，我們使用測試分割測試 SVM，並在表 3 中報告準確性。

We run the official code of AtlasNet and ShapeGF to obtain the results, since the results are not provided in their papers.
我們運行 AtlasNet 和 ShapeGF 的官方代碼來獲得結果，因為他們的論文中沒有提供結果。

For PC-GAN and PointFlow, we use the results reported in the papers.
對於 PC-GAN 和 PointFlow，我們使用論文中報告的結果。

The performance of our encoder is comparable to related state-of-the-art generative models.
我們的編碼器的性能可與相關的最先進的生成模型相媲美。

In addition, we project the latent codes of ModelNet-10 point clouds produced by the encoder into the 2D plane using t-SNE [15], and present it in Figure 7.
此外，我們使用 t-SNE [15] 將編碼器產生的 ModelNet-10 點雲的潛在代碼投影到 2D 平面，並在圖 7 中呈現。

It can be observed that there are significant margins between most categories, which indicates that our model manages to learn informative representations.
可以觀察到，大多數類別之間存在顯著的差距，這表明我們的模型設法學習了信息表示。

Further, we visualize the interpolation and extrapolation between latent codes in Figure 6.
此外，我們在圖 6 中可視化了潛在代碼之間的內插和外插。

6. Conclusions 結論

We propose a novel probabilistic generative model for point clouds, taking inspiration from the diffusion process in non-equilibrium thermodynamics.
我們從非平衡熱力學中的擴散過程中汲取靈感，提出了一種新的點雲概率生成模型。

We model the reverse diffusion process for point cloud generation as a Markov chain conditioned on certain shape latent, and derive a tractable training objective from the variational bound of the likelihood of point clouds.
我們將點雲生成的反向擴散過程建模為以某種潛在形狀為條件的馬爾可夫鏈，並從點雲似然的變分界推導出一個易於處理的訓練目標。

Experimental results demonstrate that the proposed model achieves the state-of-the-art performance in point cloud generation and auto-encoding.
實驗結果表明，所提出的模型在點雲生成和自動編碼方面達到了最先進的性能。