PointINet：一种新型点云插帧网络

卢凡，陈广*，瞿三清，李智军，刘银龙，Alois Knoll
同济大学 中国科学技术大学 慕尼黑工业大学

编者按：激光雷达和摄像头是智能驾驶车辆的主要感知器，针对于不同的任务，研究人员已经提出了许多表现优异的解决方案。但相较于图像，点云的研究起步和进展显得较为缓慢。许多图像中已经得到很好处理和应用的领域，在点云中仍然少有人涉足。同济大学陈广老师课题组为大家带来的一篇 AAAI2021 的最新论文。该工作很好地借鉴图像领域，提出了点云的一个全新方向：点云插帧。该工作的出发点在于克服雷达的低帧率，作者通过使用场景流、自适应架构和注意力模块成功地将低帧率的激光雷达点云流(10～20 Hz)上采样为高帧率的点云流(50～100 Hz)。随后作者通过详细的实验证实了其有效性，并且给出了实际应用实验展示此方向良好的使用前景。

摘要： 受硬件性能的限制，激光雷达点云流在时间维度上通常是稀疏的。通常，机械激光雷达传感器的帧率为 10 到 20 Hz，这远低于其他常用的传感器（如相机）。为了克服激光雷达传感器的时间限制，本文研究了一种名为点云插帧的新的任务。给定两个连续的点云帧，点云插帧的目的是在它们之间生成中间帧。为此，我们提出了一种新颖的框架，即点云插帧网络（PointINet）。基于所提出的方法，我们可以将低帧率点云流上采样到更高的帧率。我们首先估算两个点云之间的双向 3D 场景流，然后根据 3D 场景流将它们转换到给定的时间步长。为了融合两个变换的帧并生成中间点云，我们提出了一种新颖的基于学习的点融合模块，该模块同时考虑了两个变换的点云。我们设计了定量和定性实验来评估点云插帧方法的性能，并且在两个大型室外 LiDAR 数据集上进行的广泛实验证明了所提出的 PointINet 的有效性。我们的代码发布在了 https://github.com/ispc-lab/PointINet.git 上。

关键词：自动驾驶汽车，深度学习，激光雷达，点云插值

英文标题：《PointINet: Point Cloud Frame Interpolation Network》
文章来源： AAAI 2021
作者：Fan Lu，Guang Chen*，Sanqing Qu，Zhijun Li，Yinlong Liu，Alois Knoll
原文链接：https://arxiv.org/abs/2012.10066
项目网址：https://ispc-group.github.io/pages/pointinet.html
***通讯联系：guangchen@tongji.edu.cn**

I. 引言

激光雷达是众多应用(例如、自主车辆和智能机器人)中最重要的传感器之一。然而，典型的机械式激光雷达传感器(例如、Velodyne HDL-64E、Hesai Pandar64

等)的帧率受到硬件性能的极大限制。激光雷达的帧率一般为 10～20 Hz，这会造成点云流的时空不连续。与激光雷达的低帧率相比，智能车辆和机器人上其他常用传感器的帧率通常要高很多。例如，摄像头和惯性测量单元（IMU）的帧率可以达到 100Hz 以上。帧率的巨大差异会给激光雷达与其他传感器的同步带来困难。将低帧率的激光雷达点云流上采样到更高的帧率，可以有效地解决这一问题。此外，更高的帧率可能会提升一些应用的性能，比如物体跟踪。值得注意的是，视频插帧通常被用来从低帧率的视频生成高帧率的视频（例如，从 30Hz 到 240Hz）。与目前得到广泛注意的视频插帧相比，3D 点云的插帧还没有得到很好的探索。因此，我们需要探索 3D 点云的插帧算法，以克服激光雷达传感器的时间限制。

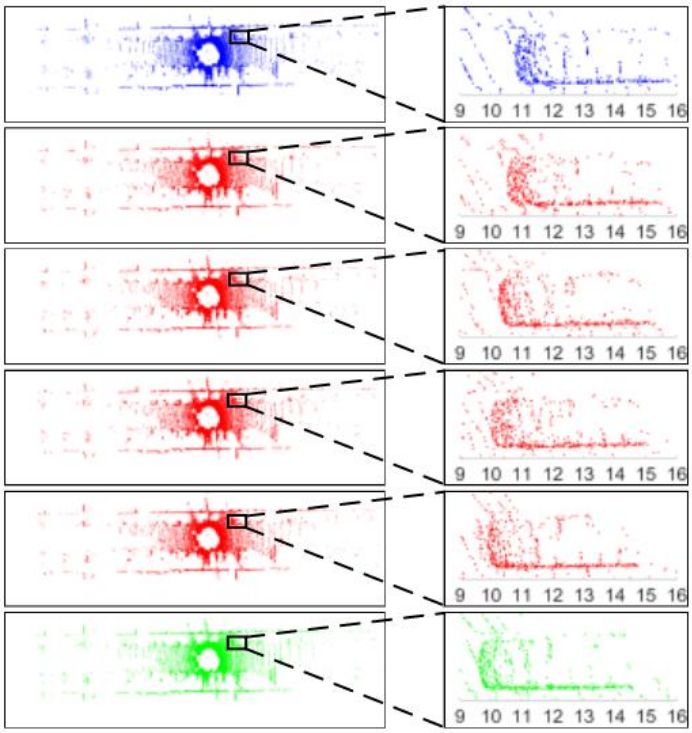

图 1 点云插帧的示意图。 蓝色和绿色的点云是两个输入帧，红色的点云是四个插值帧。 我们放大区域以显示详细信息，以实现更好的可视化。

基于以上考虑，本文研究了一个名为 "点云插帧 "的新型任务。给定两个连续的点云，点云插帧的目的是根据给定的时间步长预测中间点云帧，形成空间和时间上相干的点云流(见图 1)。因此，基于点云插帧，可以将低帧率的激光雷达点云流(10～20 Hz)上采样为高帧率的点云流(50～100 Hz)。

具体来说，为了实现点云流的时空插值，我们提出了一种新型的基于学习的框架，称为 PointINet（点云插帧网络）。我们提出的 PointINet 主要由两部分组成：点云变换模块和点融合模块。首先将两个连续的点云输入点云变换模块，将两个点云变换到给定的时间步长。为此，我们首先估计两个连续点云之间的双向三维场景流以进行运动估计。3D 场景流代表了从一个点云到另一个点云的运动场。这里我们采用名为 FlowNet3D 的基于学习的场景流估计网络来预测 3D 场景流。然后根据线性插值的 3D 场景流，将两个点云变换到给定的时间步长。此后，关

键问题是如何将两帧融合，形成新的中间点云。3D 点云是非结构化且无序化的。因此，两个点云中的点之间并不像两幅图像中的像素那样存在直接的对应关系。因而，要对两个点云进行融合是非易事。为了解决这个问题，我们提出了一种新的点融合模块。该点融合模块自适应地对两个变换点云中的点进行采样，并根据时间步长为每个采样点构建 k 最近邻(kNN)聚类，以调整两个点云的贡献。之后，我们提出的注意点融合采用注意机制将每个聚类中的点聚集起来以生成中间点云。所提出的 PointINet 的整体架构如图 2 所示。

为了评估该方法，我们设计了定性和定量实验。 此外，还进行了应用实验，以评估生成的插值点云的质量。 在两个大型室外激光雷达数据集上进行的大量实验证明了所提出的 PointINet 的有效性。

总而言之，我们的主要贡献如下：

- 为了克服激光雷达传感器的时间限制，研究了一种新颖的任务——点云插帧。
- 提出了一种名为 PointINet 的基于学习的新框架，以有效地生成两个连续点云之间的中间帧。
- 通过定性和定量实验，以验证所提出方法的有效性。

II. 相关工作

在本节中，我们简要回顾与点云插帧有关的文献。 我们从描述视频插帧的常用方法开始，然后回顾点云的 3D 场景流估计方法。

视频插帧

当前，大量视频插帧方法基于光流估计。基于光流的方法最具代表性的工作之一是 Super SloMo，它利用基于学习的方法来预测双向光流来估计连续帧之间的运动。然后，将两个输入帧进一步变换并与遮挡推理结合以生成最终的中间帧。Reda et al. 利用循环一致性来支持视频插帧的无监督学习。 Xu et al.提出了一种二次视频插值方法来利用视频中的加速度信息。视频插帧方法的另一部分是基于内核的(Niklaus, Mai, and Liu 2017a,b)。(Niklaus, Mai, andLiu 2017a)估计每个位置的内核，并通过对补丁进行卷积来预测输出像素的位置。Niklaus, Mai, and Liu 2017b 通过使用一对一维内核将插帧公式化为输入帧上的局部可分卷积来进一步改进了该方法。最近，(Bao et al. 2019)结合了基于核和光流的方法。他们利用光流来预测像素的大致位置，然后使用估计的核来完善位置。

3D 场景流估计

点云的 3D 场景流可以看作是 3D 场景中 2D 光流的推广，它表示点的 3D 运动场。与研究者对 2D 光流估计的高度研究兴趣相比，关于 3D 场景流估计的工作相对较少。 FlowNet3D 是基于深度学习的 3D 场景流估计的开创性工作。(Liu,Qi, and Guibas 2019)提出了一个流嵌入层，以对不同点云中点的运动进行建模。在 FlowNet3D 之后，FlowNet3D ++提出了几何约束来进一步提高精度。HPLFlowNet 在场景流估计中引入了双边卷积层（BCL）。 PointPWC-Net 提出了一种新颖的成本方法，并以从细到精的方式估算了 3D 场景的流量。最近，(Mittal, Okorn, and Held 2020)提供了几个不受监督的损失函数，以支持在更真实的数据集上推广预训练的场景流估计模型。在我们的实现中，由于简单性和有效性，我们选择 FlowNet3D 在两个点云之间执行 3D 场景流估计。

III. 点云插帧

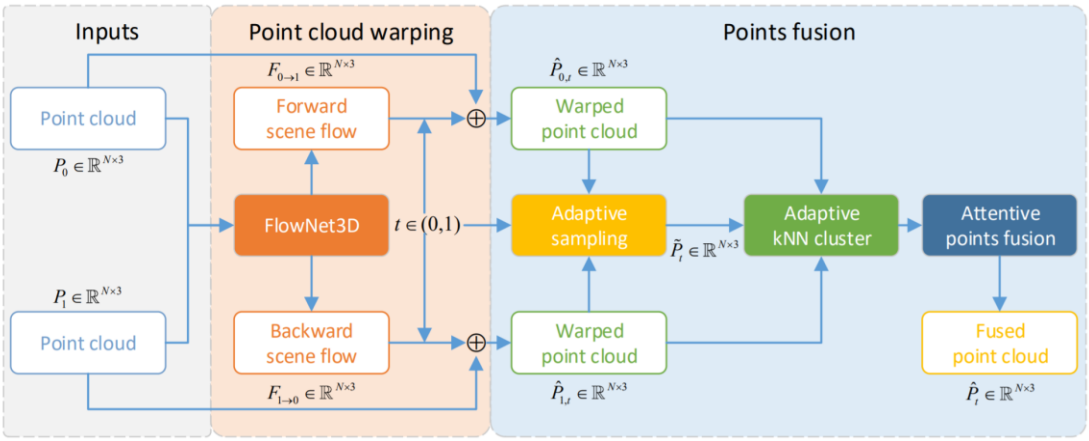

图 2 PointINet 的总体结构。 给定输入两个连续的点云，PointINet 遵循由点云变换模块和点融合模块组成的通道。

在本节中，我们首先介绍了所提出的点云插帧网络(PointINet)的整体架构，然后详细介绍了 PointINet 的两个关键部分，即点云变换模块和点融合模块。

架构总览

PointINet 的整体架构如图 2 所示。给定两个连续的点云 P0 和 P1，时间步长 t∈ (0,1)，PointINet 的目标是预测时间步长 t 中的中间点云$\hat{P}_t$。PointINet 由两个关键模块组成：点云变换模块将两个输入的点云变换到给定的时间步 t，点融合模块将两个变换的点云进行融合。下面我们将详细介绍这两个模块。

点云变换

给定两个点云 P0 和 P1，点云变换模块旨在预测 P0 中每个点在$\hat{P}_{0,t}$的位置，其中$\hat{P}_{0,t}$是 P0 的在时间步长 t 对应的点云（对于点云 P1 我们也预测$\hat{P}_{1,t}$）。这里的

关键是估计每个点从 P0 到$\hat{P}_{0,t}$的运动。我们首先预测两个点云 P0 和 P1 之间的双向 3D 场景流$F_{0\to1}\in\mathbb{R}^{N\times3}$和$F_{1\to0}\in\mathbb{R}^{N\times3}$来估计点的运动。3D 场景流是指点的 3D 运动场，它可以看作是 3D 点云中光流的推广。这里我们利用现有的基于学习的框架 FlowNet3D 来估计双向的 3D 场景流。假设连续两帧点云之间的点的运动是线性的，那么场景流$F_{0\to t}$和$F_{1\to t}$可以通过线性插值$F_{0\to1}$和$F_{1\to0}$来逼近，可以表示为：

$$\begin{aligned} F_{0\to t} &= t\times F_{0\to1} \\ F_{1\to t} &= (1-t)\times F_{1\to0} \end{aligned} \tag{1}$$

然后 P0 和 P1 可以根据插值 3D 场景流$F_{0\to t}$和$F_{1\to t}$被变换为给定的时间步长，

$$\begin{aligned} \hat{P}_{0,t} &= P_0 + F_{0\to t} \\ \hat{P}_{1,t} &= P_1 + F_{1\to t} \end{aligned} \tag{2}$$

点云融合

点融合模块的目标是将两个变换的点云进行融合，生成中间点云。点融合模块的架构显示在图 2 的右栏中。该模块的输入是两个变换的点云$\hat{P}_{0,t}\in\mathbb{R}^{N\times3}$和$\hat{P}_{1,t}\in\mathbb{R}^{N\times3}$，输出是融合后的中间点云$\hat{P}_t\in\mathbb{R}^{N\times3}$。在视频插帧中，由于图像可以表示为结构化的 2D 网格，融合步骤大多集中在遮挡和缺失区域预测上。然而，由于点云是非结构化和无序的，因此两个点云的融合是不易的。在所提出的 PointINet 中，我们首先基于时间步长 t 从两个变换的点云中自适应地采样，然后以采样点为中心构建 k 最近邻(kNN)聚类来进行融合。之后，注意点融合模块采用注意力机制，生成最终的中间点云。下面将详细介绍点融合模块的主要组成部分。

自适应采样 点融合模块的第一步是将两个变换的点云合并成一个新的点云。直观上，两个点云对中间点云的贡献并不总是相同的。例如，在 t=0.2 时的中间帧$\hat{P}_t$与第一帧 P0 应该比第二帧 P1 更相似。基于上述观察，我们从$\hat{P}_{0,t}$和$\hat{P}_{1,t}$中随机抽取 N0 和 N1 个点，分别生成两个采样点云$\hat{P}_{0,t}\in\mathbb{R}^{N\times3}$和$\hat{P}_{1,t}\in\mathbb{R}^{N\times3}$。其中 N0=（1-t）×N 和 N1=t×N。这种操作使得网络能够根据目标时间步长 t 自适应地调整两个变换点云的贡献。接近时间步长 t 的点云对中间帧$\hat{P}_t$的贡献更大。之

后，$\hat{P}_{0,t}$和$\hat{P}_{1,t}$合并成一个新的点云$\tilde{P}_t \in \mathbb{R}^{N\times 3}$。

自适应 kNN 聚类 我们将$\tilde{P}_t$输入到自适应的 kNN 聚类模块中，生成 k 个最近邻聚类，作为后续注意点融合模块的输入。对于$\tilde{P}_t$中的每一个点，我们在两个变换点云$\hat{P}_{0,t}$和$\hat{P}_{1,t}$中搜索 K 个邻近点。与自适应采样类似，$\hat{P}_{0,t}$和$\hat{P}_{1,t}$中的邻近点数也根据 t 进行自适应调整，以平衡两个点云的贡献。因此，我们在$\hat{P}_{0,t}$中查询 K0 个邻近点，在$\hat{P}_{1,t}$中查询 K1 个邻近点，其中 K0=(1-t) ×K，K1=t×K。因此，我们得到 N 个聚类，每个聚类由 K 个邻近点组成。将聚类的中心点表示为 xi，邻点表示为$\{x_1^i,\cdots,x_k^i,\cdots,x_K^i\} \in \mathbb{R}^{K\times 3}$。然后将每个邻点以$(x_k^i - x^i)$的形式减去中心点，得到邻点在聚类中的相对位置。此外，邻点与中心点之间的欧氏距离$\|x_k^i - x^i\|_2$作为聚类的附加通道被计算出来。因此，单个聚类的最终特征可以表示为$F^i = \{f_1^i,\cdots,f_k^i,\cdots,f_K^i\} \in \mathbb{R}^{K\times 4}$。

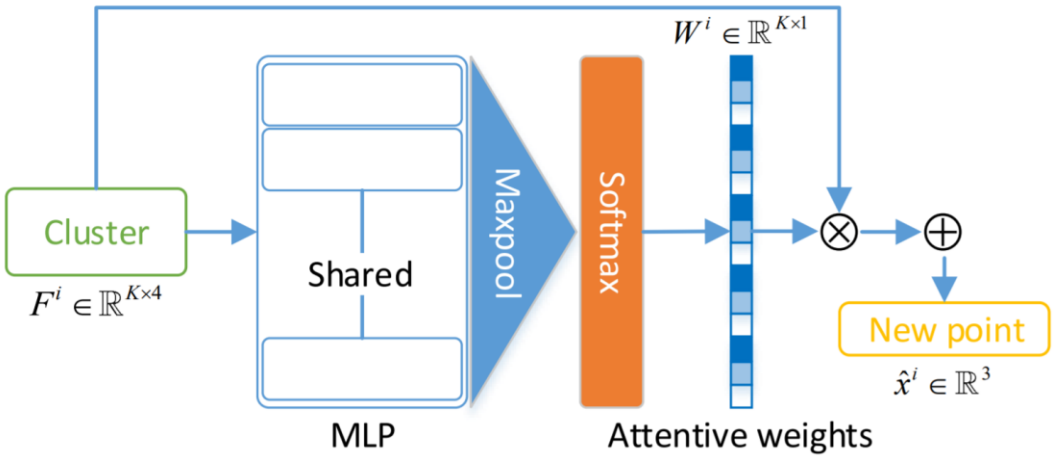

图 3 注意点融合模块的网络架构

注意点融合 注意力机制在 3D 点云学习中已经得到了广泛的应用。在这里，我们采用注意机制来聚合相邻点的特征来生成中间点云的新的点。注意力点融合模块的网络架构可参见图 3。受 PointNet 和 PointNet++的启发，我们将单个聚类的特征 Fi 输入共享多层感知器（Shared-MLP）以生成特征图。然后应用后续的 maxpool 层和 Softmax 函数对聚类中的所有邻点预测一维注意权重 Wi=$\{w_1^i,\cdots,w_k^i,\cdots,w_K^i\} \in \mathbb{R}^{K\times 1}$。之后，新点$\hat{x}^i$可以表示为邻点的加权和。

$$\hat{x}^i = \sum_{k=1}^{K} x_k^i \cdot w_k^i, \quad i = 1, \cdots, N \tag{3}$$

最后，生成的中间点云$\tilde{P}_t$可以表示为$\hat{P}_t = \{\hat{x}^1, \cdots, \hat{x}^N\} \in \mathbb{R}^{N\times 3}$。直观地讲，所提出的注意点融合模块可以给点集中与目标点云更一致的点分配更高的权重。在点融合模块之后，新的中间点云中的每一个生成点都是由其接受场中的两个点云中的邻点聚合而成。此外，借助自适应采样和自适应 kNN 集群模块，可以根据时间步长 t 动态调整两个点云的贡献。因此，生成的中间点云是两个输入点云的有效融合。

损失

倒角距离通常用于测量两个点云的相似度。 在这里，我们利用倒角距离来监督所提出的 PointINet 的训练。 给定生成的中间点云$\hat{P}_t \in \mathbb{R}^{N\times 3}$和真值点云$P_t \in \mathbb{R}^{N\times 3}$ ，倒角距离损失可以表示为

$$\mathcal{L} = \frac{1}{N} \sum_{\hat{x}^i \in \hat{P}_t} \min_{x^j \in P_t} \left\| \hat{x}^i - x^j \right\|_2 + \frac{1}{N} \sum_{x^j \in P_t} \min_{\hat{x}^i \in \hat{P}_t} \left\| \hat{x}^i - x^j \right\|_2 \tag{4}$$

其中‖ • ‖代表 L2 归一化。

IV. 实验

我们进行定性和定量实验，以证明所提出的方法的性能。 此外，我们还在两个应用上（例如，关键点检测和多帧迭代最近点（ICP））进行了实验，以更好地评估生成的中间点云的质量。

数据集

我们在两个大型室外 LiDAR 数据集（即 KITTI 里程表数据集和 nuScenes 数据集）上评估了该方法。 KITTI 里程表数据集提供了 11 个具有地面真实性的序列（00-10），我们使用序列 00 训练网络，使用 01 进行验证，而使用其他序列进行评估。 NuScenes 数据集包含 850 个训练场景，我们使用前 100 个场景进行训练，其余 750 个场景进行评估。 由于缺乏高帧速 LiDAR 传感器，我们简单的将 KITTI 里程表数据集中的 10 Hz 点云下采样为 2 Hz，将 nuScenes 数据集中的 20 Hz 点云下采样为 4 Hz，用于训练和定量实验。 因此，在降采样点云流的两个连续帧之间有 4 个中间点云。

实施细节

我们首先在 Flythings3D 数据集上训练 FlowNet3D，然后在 KITTI 场景流数据集上优化网络。我们直接使用(Liu, Qi, and Guibas 2019)预处理的数据来训练 FlowNet3D。然后，我们分别在 KITTI 里程表数据集和 nuScenes 数据集上分别完善预训练的 FlowNet3D。在此过程中，将当前帧与前后 Ns 帧内随机选择的帧作为训练对。然后，以预测的场景流将第一帧变换到第二帧，并使用变换点云和第二点云之间的倒角距离（请参见公式 4）损失函数进行监督 FlowNet3D 的完善。之后，在训练随后的点融合模块时，FlowNet3D 的权重是固定的。在训练点融合模块的过程中，将两个连续的帧和来自 4 个中间点云的随机采样帧以及相应的时间步长用作训练样本。我们在训练期间将点云随机降采样为 16384 点，并且在我们的实现中将相邻点的数量 K 设置为 32。注意点融合模块中 Shared-MLP 的层的通道设置为[64,64,128]。所有网络均使用 PyTorch 实现，而 Adam 被用作优化器。此外，点融合模块仅在 KITTI 里程表数据集上进行训练，我们仅将训练后的模型推广到 nuScenes 数据集进行评估。

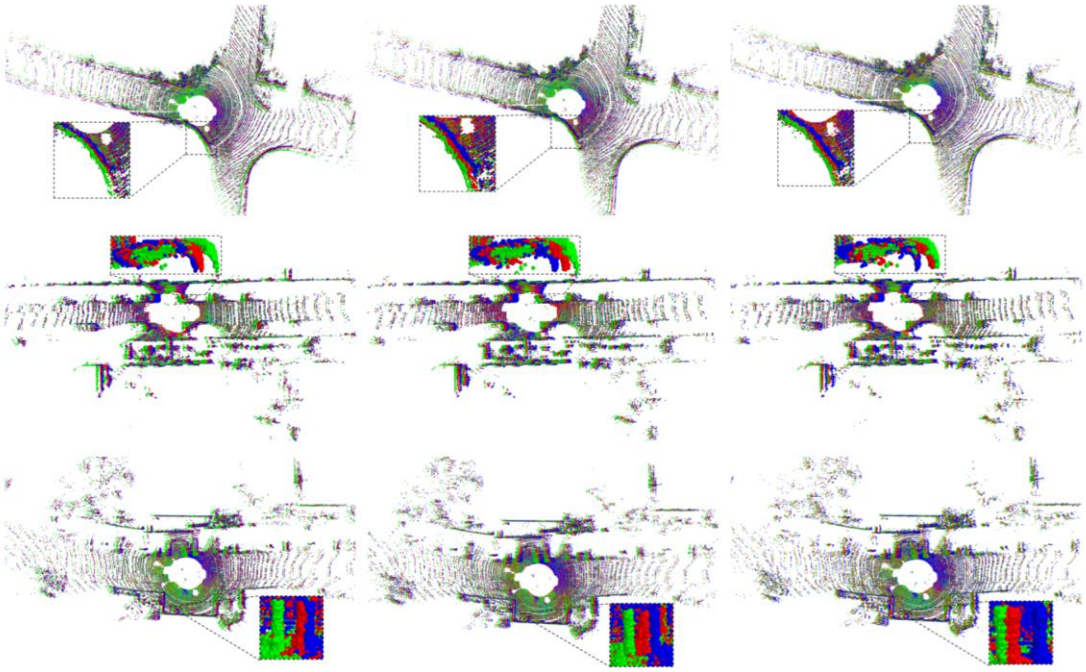

图 4 PointINet 的定性结果。 从上至下是三对连续帧的插值结果。 列从左到右的时间步长 t 分别为 0.25、0.50 和 0.75。 蓝色，绿色和红色点云分别代表第一帧，第二帧和预测的中间帧。 此外，我们放大点云的某个区域，然后将其旋转到适当的角度，以更好地可视化插值点云的细节。

定性实验

提出的 PointINet 的目标是从低帧率的流生成高帧率的激光雷达流。但是，没有现有的高帧率 LiDAR 传感器。因此，我们用 Ns= 1 训练 FlowNet3D，以便为更近的点云提供适当的场景流估计，然后将在降采样点云流上训练的点融合模块直接应用于 KITTI 里程表数据集的 10 Hz 点云流上以生成高帧速率点云流。在这里，我们在图 4 中提供定性可视化，其中此处的点数设置为 32768。10Hz 点云流被上采样到 40 Hz，中间帧的时间步长设置为 0.25、0.50 和 0.75。根据图 4，

提出的 PointINet 很好地估计了两个云之间的点的运动，并且融合算法可以保留点云的细节。除此之外，我们还在补充材料中提供了一些演示视频，以比较高帧率点云流和低帧率点云流。根据演示视频，高帧率点云流在时间和空间上明显比低帧率点云平滑。

定量实验

评价指标 我们使用两个评估指标：倒角距离（CD）和推土机距离（EMD）来评估降采样点云流上生成的点云与地面真实云之间的相似性和一致性。CD 先前在等式 4 中进行了描述。 EMD 也是比较两个点云(Weng et al. 2020)的常用度量，其是通过解决线性分配问题来实现的。 给定两点云$\hat{P}_t \in \mathbb{R}^{N\times 3}$和$P_t \in \mathbb{R}^{N\times 3}$，EMD 可以被描述为：

$$EMD = \min_{\phi:\hat{P}_t \to P_t} \frac{1}{N} \sum_{\hat{x}\in\hat{P}_t} \|\hat{x} - \phi(\hat{x})\|_2 \qquad (5)$$

其中，$\phi : \hat{P}_t \to P_t$是一个双映射。

基线 为了演示建议的 PointINet 的性能，我们定义了 3 个基线与我们的方法进行比较：（1）一致性方法 我们仅将第一点云框架复制为中间点云。 （2）对齐 ICP 方法 我们首先使用迭代最近点（ICP）算法估计点云的两个连续帧之间的刚性变换，然后对其进行线性插值以获得第一帧和中间帧之间的变换。之后，基于该变换将第一点云变换为中间帧。 （3）场景流方法 我们使用 FlowNet3D 估计连续两个帧之间的 3D 场景流，并通过线性插值计算从第一帧到中间帧的场景流，然后根据 3D 场景流对第一点云进行变换得到中间点云。在定量实验中，通过随机采样将所有点云降采样为 16384 点。

表 1 在 KITTI 里程表数据集上对 PointINet 和其他基线进行定量评估的结果

Metric	Identity	Align-ICP	Scene flow	Ours
CD↓	1.398	0.752	0.687	**0.457**
EMD↓	68.93	83.79	57.13	**39.46**

表 2 在 nuScenes 数据集上对 PointINet 和其他基线进行定量评估的结果

Metric	Identity	Align-ICP	Scene flow	Ours
CD↓	0.617	0.555	0.511	**0.487**
EMD↓	54.24	51.12	50.97	**47.98**

结果 提议的 PointINet 的 CD 和 EMD 以及 KITTI 里程表数据集和 nuScenes 数据集上的其他基线分别显示在表 1 和表 2 中。 根据结果，我们的方法的性能明显优于其他基准。 例如，建议的 PointINet 的倒角距离约为 KITTI 里程表上一致性方法，对齐 ICP 方法和场景流方法的 1/3 ，3/5 和 2/3。 值得注意的是，我们的方法明显优于场景流方法，这也反映了点融合模块的有效性。 注意，我们仅在 KITTI 里程表数据集上训练点融合模块，而在 nuScenes 数据集上的结果也证明了网络的泛化能力。

应用

为了更好地评估生成的中间点云的质量以及与原始点云的相似性，我们在插值点云流和原始点云流上测试了两个应用的性能，即关键点检测和多帧 ICP。 我们首先分别将 KITTI 里程表数据集中的 10 Hz 点云和 nuScenes 数据集中的 20 Hz 点云分别下采样到 5 Hz 和 10 Hz，然后将它们作为内插点云流内插到原始的帧率。 比较了两个不同点云流上两个应用程序的结果，以验证所提出的 PointINet 的有效性。

表 3 KITTI 里程表数据集上原始和插值点云的 3 个不同关键点的可重复性

Keypoints	Harris-3D	SIFT-3D	ISS
Original	0.155	0.174	0.163
Interpolated	0.138	0.151	0.133

关键点检测 我们在两个点云流中执行 3D 关键点检测，并评估检测到的关键点的可重复性。我们选择了 3 个手工制作的 3D 关键点，即 SIFT-3D，Harris-3D 和 ISS。使用 PCL 中的实现提取所有关键点。如果点云中的一个关键点到另一个点云中最近的关键点的距离（在基于地面真相姿势进行刚性变换之后）在阈值 δr（δr 设置为 0.5 m）之内，则该点被视为可重复。在此，可重复性是可重复关键点的比率。我们计算当前点云中关键点在前后 5 帧中的平均可重复性，并将关键点的数量设置为 256。由于 nuScenes 数据集中缺少每帧地面真相姿势，因此关键点检测实验为仅在 KITTI 里程表数据集上执行，其结果显示在表 3 中。根据结果，与原始点云流相比，插值点云流的可重复性仅稍有降低。例如，插值点云的 Harris-3D 的可重复性仅比原始点云的可重复性低 0.017。结果从侧面反映了生成的中间点云与地面真点云的高度一致性。

表 4 KITTI 里程表数据集上原始点和内插点云流的多帧 ICP 性能。

Metric	Original	Interpolated	Difference
RTE (m)	4.31	4.57	0.26
RRE (deg)	2.70	2.95	0.25

表 5 nuScenes 数据集上原始和内插点云流的多帧 ICP 性能。

Metric	Original	Interpolated	Difference
RTE (m)	1.65	1.72	0.07
RRE (deg)	0.91	0.92	0.01

多帧 ICP 我们对 Nm 个连续帧执行迭代最近点（ICP）算法，以估计第一帧与最后一帧之间的刚性变换。在 KITTI 里程表数据集上将 Nm 设置为 10。对于 nuScenes 数据集，仅为关键帧（大约 2 Hz）提供真实姿态。因此，将 Nm 设置为与 nuScenes 数据集上两个关键帧之间的帧数相同。我们利用 PCL 中的实现来执行 ICP 算法。累加一次转换以获得第一帧和最后一帧之间的转换。计算相对平移误差（RTE）和相对旋转误差（RRE），以评估多帧 ICP 估计变换的误差。有关 KITTI 里程表数据集和 nuScenes 数据集的结果分别显示在表 4 和表 5 中。我们还计算原始点和插值点云流的误差之间的差异，并将结果显示在表 4 和表 5 的右栏中，以便进行更好的比较。根据结果，插值点云流上的多帧 ICP 算法的 RTE 和 RRE 非常接近原始点云流。例如，根据表 5，两个结果在 nuScenes 数据集上的 RTE 仅相差 0.07 m。紧密的性能表明生成的中间点云与地面真实云之间的相似性。

根据这两个应用的实验，由于所提出的插值方法可能存在误差，因此在插值点云上算法的性能略逊于原始点云流。尽管如此，这两个应用程序的紧密性能证明了生成的点云与原始应用程序具有高度的相似性和一致性。

表 6 PointINet 及其组件针对不同数量的点的运行时间（ms）。

Number of points	16384	32768	65536
Point cloud warping	167.3	291.1	529.3
Points fusion	36.4	81.3	196.6
PointINet	203.7	372.4	725.9

表 7 在 KITTI 里程表数据集上对消融研究的定量评估结果。

Methods	CD↓	EMD↓
full PointINet	**0.457**	**39.46**
w/o adaptive sampling	0.580	48.00
w/o adaptive kNN cluster	0.534	41.66
w/o attentive points fusion	0.555	40.67

效率

在配备 NVIDIA Geforce RTX 2060 的 PC 上评估了拟议的 PointINet 的效率，并在表 6 中显示了为包含 16384、32768 和 65536 点的点云生成一个中间帧的平均运行时间。 根据结果，大多数运行时都用于变换点云，并且所提出的点融合模块需要相对较少的时间进行计算。 然而，由于用于融合的每点计算，用于点融合模块的计算时间随着点的数量而增加。 总体而言，建议的 PointINet 可以有效地生成中间帧。

消融实验

我们进行了多次消融研究，以分析拟议 PointINet 的不同组成部分（例如，自适应采样，自适应 kNN 聚类和注意点融合）对最终结果的影响。 实验设置与定量实验一致，我们还使用倒角距离（CD）和推土机距离（EMD）来评估性能。所有消融研究均在 KITTI 里程表数据集上进行。

自适应采样 我们通过简单地随机采样两个变换点云中一半的点以形成一个新点云作为自适应 kNN 集群模块的输入来代替自适应采样策略。结果显示在表 7 的第二行中。根据结果，在不进行自适应采样的情况下，CD 和 EMD 分别增加了 0.123 和 8.54，这表明自适应采样策略显着提高了性能。

自适应 kNN 聚类 我们从两个变换的点云中固定地查询 $K/2$ 个邻居点，而不是根据时间步长 t 来查询点。根据表 7 第三行中显示的结果，没有自适应 kNN 聚类的 CD 和 EMD 分别从 0.457 增加到 0.534，从 39.46 增加到 41.66。结果证明了自适应 kNN 集群模块的有效性。

注意点融合 为了演示注意点融合模块的效果，我们直接使用自适应采样中的点云$\tilde{P}_t$作为中间点云，并将结果显示在表格 7 的底行中。根据结果，注意点融合模块明显提高了最终性能。

V. 结论

本文研究了一种名为点云插帧的新颖任务，并为此任务设计了基于学习的框架 PointINet。 给定两个连续的点云，该任务旨在预测它们之间在时间和空间上一致的中间帧。因此，使用所提出的方法可以将低帧率点云流上采样到高帧率。为此，我们利用现有的场景流估计网络进行运动估计，然后将两个点云变换到给定的时间步长。然后提出了一种新颖的基于学习的点融合模块，以有效地融合两个点云。 我们为此任务设计了定性和定量实验。 在 KITTI 里程表数据集和 nuScenes 数据集上进行的大量实验证明了所提出的 PointINet 的性能和有效性。

VI. 道德声明

提出的点云插帧方法可能对无人驾驶和智能机器人的发展产生积极影响，可

以减轻驾驶员和工人的工作量，并减少交通事故的发生。 但是，这种发展也可能导致驾驶员和工人失业。 此外，所提出的方法可能具有潜在的军事应用，例如军用无人机，可能会威胁到人类的安全。 我们应该探索更多可以改善人类生活而不是有害生活的应用。

VI. 致谢

这项工作由中国国家自然科学基金（No. 61906138），欧盟根据特定拨款协议第 945539 号达成的 Horizon 2020 研究与创新框架计划（人脑计划 SGA3）和 2018 上海 AI 创新发展计划资助。

参考文献

Bao, W.; Lai, W.-S.; Zhang, X.; Gao, Z.; and Yang, M.-H. 2019. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE transactions on pattern analysis and machine intelligence* .

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11621–11631.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 2758–2766.

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

Flint, A.; Dick, A.; and Van Den Hengel, A. 2007. Thrift: Local 3d structure recognition. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, 182–188. IEEE.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361. IEEE.

Gu, X.; Wang, Y.; Wu, C.; Lee, Y. J.; and Wang, P. 2019. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3254–3263.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2462–2470.

Jiang, H.; Sun, D.; Jampani, V.; Yang, M.-H.; Learned-Miller, E.; and Kautz, J. 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9000–9008.

Kiani Galoogahi, H.; Fagg, A.; Huang, C.; Ramanan, D.; and Lucey, S. 2017. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 1125–1134.

Liu, H.; Liao, K.; Lin, C.; Zhao, Y.; and Guo, Y. 2020. Pseudo-LiDAR Point Cloud Interpolation Based on 3D Motion Representation and Spatial Supervision. *arXiv preprint arXiv:2006.11481* .

Liu, X.; Qi, C. R.; and Guibas, L. J. 2019. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 529–537.

Liu, Y.-L.; Liao, Y.-T.; Lin, Y.-Y.; and Chuang, Y.-Y. 2019. Deep video frame interpolation using cyclic frame generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8794–8802.

Liu, Z.; Yeh, R. A.; Tang, X.; Liu, Y.; and Agarwala, A. 2017. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, 4463–4471.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4040–4048.

Menze, M.; and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3061–3070.

Mittal, H.; Okorn, B.; and Held, D. 2020. Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11177–11185.

Niklaus, S.; Mai, L.; and Liu, F. 2017a. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 670–679.

Niklaus, S.; Mai, L.; and Liu, F. 2017b. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, 261–270.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, 8026–8037.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, 5099–5108.

Reda, F. A.; Sun, D.; Dundar, A.; Shoeybi, M.; Liu, G.; Shih, K. J.; Tao, A.; Kautz, J.; and Catanzaro, B. 2019. Unsupervised video interpolation using cycle consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, 892–900.

Rusu, R. B.; and Cousins, S. 2011. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, 1–4. IEEE.

Sipiran, I.; and Bustos, B. 2011. Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes. *The Visual Computer* 27(11): 963.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8934–8943.

Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; and Shan, J. 2019. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10296–10305.

Wang, X.; He, J.; and Ma, L. 2019. Exploiting Local and Global Structure for Point Cloud Semantic Segmentation with Contextual Point Representations. In *Advances in Neural Information Processing Systems*, 4571–4581.

Wang, Z.; Li, S.; Howard-Jenkins, H.; Prisacariu, V.; and Chen, M. 2020. FlowNet3D++: Geometric losses for deep scene flow estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, 91–98.

Weng, X.; Wang, J.; Levine, S.; Kitani, K.; and Rhinehart, N. 2020. Inverting the Pose Forecasting Pipeline with SPF2: Sequential Pointcloud Forecasting for Sequential Pose Forecasting. *CoRL* .

Wu, W.; Wang, Z.; Li, Z.; Liu, W.; and Fuxin, L. 2019. PointPWC-Net: A Coarse-to-Fine Network for Supervised and Self-Supervised Scene Flow Estimation on 3D Point Clouds. *arXiv preprint arXiv:1911.12408* .

Xu, X.; Siyao, L.; Sun, W.; Yin, Q.; and Yang, M.-H. 2019. Quadratic video interpolation. In *Advances in Neural Information Processing Systems*, 1647–1656.

Yang, J.; Zhang, Q.; Ni, B.; Li, L.; Liu, J.; Zhou, M.; and Tian, Q. 2019. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3323–3332.

Zhong, Y. 2009. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 689–696. IEEE.