



时序图神经网络改进的拉格朗日流体模拟方法

摘要

在计算能力不断提高的当下，计算机图形学中的物理模拟也越来越重要。流体模拟在电影、游戏和工业界都有广泛的应用。其中基于拉格朗日视角的流体模拟方法特别流行。本文提出了一种使用图形神经网络来描述拉格朗日流体模拟的方法，并在其中引入了时序网络以使得网络能够自适应地构建图结构，避免了图卷积网络对图结构先验知识的需求和对图结构变化的敏感性。本文中的网络可以处理多变量时间序列的粒子特征数据。通过引入门控循环单元和自注意力机制来捕获粒子时序变化特征，并在此基础上，通过图卷积网络捕获多粒子之间的交互行为和潜在的关系。

本文所使用的训练数据集均为光滑粒子动力学方法生成。本文模型在使用有限的情况下证明了相比于原始工作的有效性。

关键词：SPH，图卷积神经网络，时序神经网络



IMPROVED LAGRANGIAN FLUID SIMULATION WITH TEMPORAL NEURAL NETWORK

ABSTRACT

With the continuous improvement of computing power, physical simulation is becoming a more and more important topic in computer graphics. Fluid simulation has a wide range of applications in film, games and industry. Fluid simulation methods based on the Lagrangian perspective are particularly popular. This paper proposes a method for describing Lagrangian fluid simulations using graph neural networks, and introduces a time-series network to construct graph structures adaptively. This network avoids the need of prior knowledge of graph structures in graph convolutional networks and reduces the model sensitivity to graph structure changes. The network in this paper handles multivariate time-series particle features. By introducing a gated recurrent unit and a self-attention mechanism the model can capture the temporal features of particles. On this basis, the interaction behavior and potential relationships between multiple particles are captured through a graph convolutional network.

The training datasets used in this paper are generated by the smooth particle dynamics method. The model in this paper demonstrates its effectiveness compared to the original work using limited data.

Key words: SPH, graph convolutional neural network, temporal neural network



目 录

第一章 绪论.....	1
1.1 研究现状.....	1
1.1.1 SPH 流体模拟模型	1
1.1.2 SPH 机器学习研究	2
第二章 基础知识.....	4
2.1 SPH 算法	4
2.1.1 光滑粒子动力学 (Smoothed Particle Hydrodynamics)	4
2.1.2 对流体用粒子视角进行建模.....	5
2.1.3 压力计算 (Pressure)	6
2.1.4 黏度计算 (Viscosity)	6
2.1.5 外力 (External Forces)	6
2.1.6 光滑核 (Smoothing Kernels)	6
2.2 机器学习方法.....	8
2.2.1 神经网络.....	8
2.2.2 图神经网络 (GNN)	8
2.2.3 时序神经网络 (GRU)	9
第三章 模型构建.....	11
3.1 SPH 算法实现	11
3.2 神经网络结构.....	11
第四章 实验设计.....	15
4.1 数据集收集.....	15
4.2 训练过程.....	15
4.3 训练结果.....	15
第五章 结论.....	18
5.1 实验结果.....	18
5.2 实验缺陷.....	19
致谢.....	23



第一章 绪论

对于物理的理解和模拟对现实环境的探索和交互至关重要。随着计算机科学的发展，数值模拟方法已经和理论分析、物理实验一同成为了主要物理研究方式。相较于昂贵且困难的实验，数值模拟方法可以提供一种廉价、高效、安全的实验方法。这使得数值模拟方法已经成为工业界广泛应用的科学工具之一。数值模拟方法主要包括网格法（如有限元法）和无网格法（如光滑粒子动力学）。

在如何处理例如数值模拟这类复杂问题上，采用数据驱动的方式，使用神经网络辅助已经成为了一种越发重要并有广大前景的尝试。借助神经网络的物理表达方式的其中一种重要方向是基于拉格朗日的无网络表达方式（Lagrangian presentation）^[1]。拉格朗日无网络方式基于粒子进行表达，在数值计算中占有举足轻重的地位，并且在刚体，可变形固体和流体等方面的数据模拟都有了出色的表现和充分的探索^[2]。

在将机器学习方法引入物理模拟的过程中，现在主流的方法使用图结构来定义粒子之间的交互——两个粒子之间是否存在一条有向/无向的边，来确定两个粒子之间是否发生交互。因此主流的机器学习方式需要在每次更新粒子状态的同时，重新构建粒子之间的图结构，并将重新构建的图结构重新传输进入神经网络中。

但是这种做法会带来两个主要的问题。首先在粒子数量增多时，如何从大量的粒子中正确提取粒子之间的关系是一个具有挑战的问题。其次在图神经网络的构建中，粒子之间是否发生交互只有两个状态（发生交互—图节点之间有边相连，不发生交互—图节点之间无边相连）。事实上粒子之间的交互程度不应该是一个离散的值而应当是一个连续的数值。

因此本文在使用图表示粒子之间关系的基础上，建立了一个自适应的图结构学习模块，在神经网络中计算图结构。由于粒子本身带有速度和动量，粒子之间的关系并不是一个独立时间片上的状态，本文因此在网络结构中加入了时间序列编码。

1.1 研究现状

数值模拟主要包括网格法和无网格法。网格法在理论上非常完善，在实践中也有广泛的应用。相较于无网格法，网格法在计算精度和复杂情况下都有优势。但是在一些特殊情况下例如大变形、爆炸冲击、自由表面等问题上，网格法在处理上就会遇到问题，在此情况下，无网格方法的优势就体现了出来^[4]。

1.1.1 SPH 流体模拟模型

流体模拟无论在理解现实世界的物理现象还是在材料科学、工程中都有重要的地位和意义。其中一种重要的方法是光滑动力粒子Smoothed Particle Hydrodynamics（以下简称SPH）。SPH方法基于拉格朗日视角模拟流体粒子。SPH方法起源于基于粒子的天体物理学模型^[3]，最早应用于模拟行星爆炸等天体物理问题。作为一种采用解偏微分方程的数值计算方法，SPH方法可以在不用绘制特殊网格的方法下更好地处理介质的扭曲和拉伸，包括一些真空的空洞；此外SPH方法在经过调整粒子和粒子之间的作用力方程后还可以更好地模拟更为复杂的物理现象。此后SPH算法在流体等现象中也有广泛的应用。然而天体物理



学中所采用的可压缩性流体处理方法并不能直接应用于类似于水的非压缩流体。此后Muller对SPH方法进行了较大的改善，使其能够适用于流体^[5]。SPH方法基于粒子在某一特定时刻的分布信息场的近似，使得算法具有较好的自适应性且粒子能够同时表示空间点和物质点两种属性。因此SPH算法在提出后就一直经历了快速的发展，并相继提出了多种衍生算法。SPH和其各种变种已经被广泛地应用在对现实复杂现象的物理建模和视觉展示上。通过采用不同的函数来构造近似函数后使用不同方法对连续场进行离散，SPH算法可以有非常丰富的变种类型。目前SPH算法仍然在多个计算力学领域有着广泛的应用。

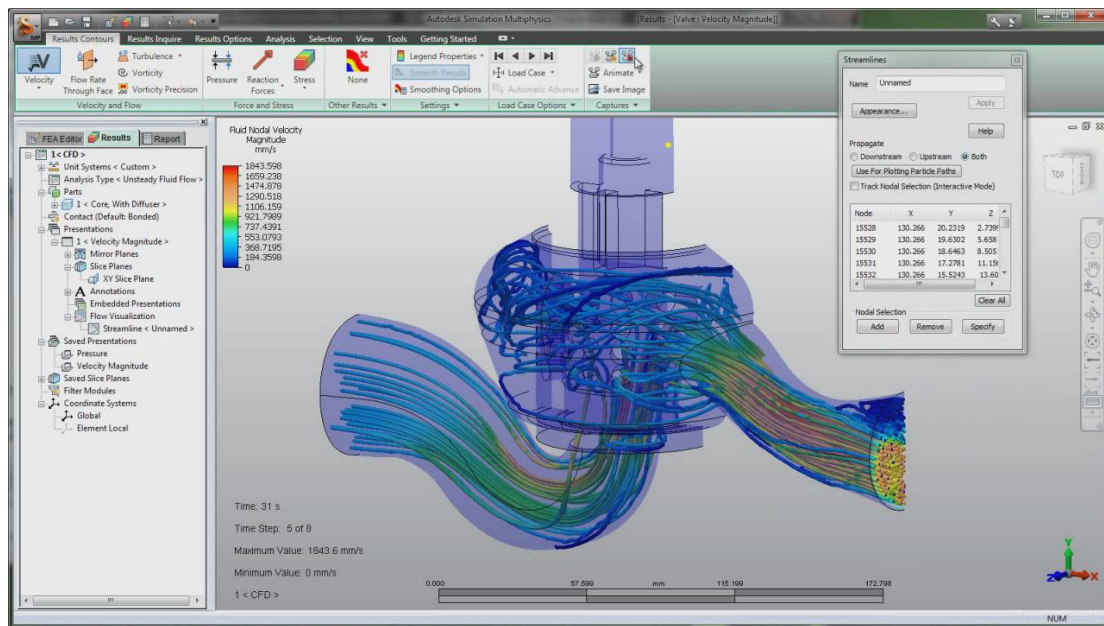


图 1-1 Autodesk 仿真软件的流体模拟^[20]

1.1.2 SPH 机器学习研究

随着机器学习方法的兴起，越来越多的机器学习方法和流体模拟结合在一起。以拉格朗日为视角的粒子流已经被证实适合和机器学习结合在一起。Ladicky 首先证明流体计算可以使用回归森林 (Regression Forests) 进行学习^[6]。卷积神经网络 (CNN) 也曾经被添加到计算中用于加速压力部分的计算^[7]。此后在对拉格朗日视角的图表示的通用化物理模拟 (包括刚体和可变性物体) 也有了一系列的成果。Battaglia 采用了一种交互网络，将图作为模型的输入，基于对象和关系，使用深度学习网络来预测复杂系统中对象的交互，在处理几十个对象在数千个时间步长上取得了较好的结果^[8]。Alvaro 实现了一个基于图神经网络的模型，通过归纳偏置，可以对处于以对象和关系为中心的体系学习到一个准确的前向模型并且有着良好的泛化性^[9]。此外也有大量直接将图表示的神经网络直接应用于流体模拟的例子^{[10][11]}。本文与以上的工作都有着共同的出发点的思想——通过使用可微神经网络对拉格朗日流体进行建模，并使用图的方式对粒子关系进行建模。

传统的深度学习本身可以有效地捕捉到欧几里得数据潜在的特征，但是越来越多的数据不能通过欧几里得数据表示，特别是 SPH 方法中粒子的表示。因此使用非欧的数据结构，图，会是一个更好的选择。然而图可能是有可变大小的无序节点，也可能有复杂的邻居关系，因此需要提出一种特殊的图神经网络来处理图数据。



受到卷积神经网络成功的启发和鼓舞，有大量的研究讨论如何定义图数据的卷积概念。本文也将基于图卷积网络实现，图卷积网络的具体定义将在下文讨论。

需要额外指出的是从物理的角度考虑，流体力学和其他广泛的物理过程均可以通过连续的方法来描述。使用连续的偏微分方程，而不是离散后抽象而成的图结构。现在也有一些工作不使用图作为粒子的底层表示方法，而使用连续的视角，使用连续卷积的方法来学习流体力学^[21]。



第二章 基础知识

2.1 SPH 算法

本文所采用的 SPH 算法基于 2003 年 Matthias Muller 提出的针对流体的 SPH 算法^[5]。

2.1.1 光滑粒子动力学 (Smoothed Particle Hydrodynamics)

光滑粒子动力学 (Smoothed Particle Hydrodynamics) 首先由 Lucy 和 Gingold 提出用于解决天体物理的模拟问题。但是这一方法也可以应用于任意形式的流体模拟。

SPH 方法本质上是一种粒子系统的插值算法，通过插值来处理离散化的粒子场量。在计算某个物理量时，SPH 方法通过对一个小的邻近区域的粒子使用辐射对称的光滑核进行采样插值。例如某个物理量 A 在局部 r 范围内的粒子插值得到：

$$f(x) = \int_{x'} f(x') \delta(|x - x'|) dV \quad (2-1)$$

其中 x' 表示所有的粒子（空间内所有正在模拟的粒子或一个小的邻近区域的粒子）； $f(x)$ 表示我们需要插值计算的物理量； V 表示粒子的体积； δ 表示此处的插值方程。

在 SPH 方法下，我们使用的是光滑核 (Smoothing Kernel) 进行插值。因此上述方程也可以写为：

$$f(x) = \int_{x'} f(x') W(|x - x'|, h) dV \quad (2-2)$$

在此基础上再对积分进行离散化，就可以得到 SPH 方法的基本公式：

$$f(x) = \sum_i f_i W(|x - x'|, h) V_i \quad (2-3)$$

在插值中使用到的方程 $W(r, h)$ 即为半径距离为 h 的光滑核， r 为粒子之间的距离。在 SPH 方法中光滑核的选取可以是任意的，但是应该满足以下条件。

1. 光滑核是对称的。 $W(r, h) = W(-r, h)$ 。
2. 光滑核是归一化的。 $\int W(r) dr = 1$ 。

在满足以上两个条件的前提下，SPH 算法的插值具有二阶精度。

注意该公式在原文中被写作^[5]：

$$A_s(r) = \sum_j m_j \frac{A_j}{\rho_j} W(r - r_j, h) \quad (2-4)$$

这是因为在流体中粒子的体积被写作质量 m 比作密度 ρ 。



即为 $V_i = \frac{m_i}{\rho_i}$ 。

以下讨论均以公式 2-4 为基础进行讨论。

例如我们在推导密度 ρ 时，可以利用公式 2-4：

$$\rho_i = \sum_j m_j \frac{\rho_j}{\rho_j} W(r - r_j, h) = \sum_j m_j W(r - r_j, h) \quad (2-5)$$

在大多数流体方程中，我们不仅需要考虑对场量进行插值计算，还需要对场量的导数进行计算。类似地，基于公式 2-4，场量 A 的导数可以如下计算：

$$\nabla A_s(r) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(r - r_j, h) \quad (2-6)$$

二阶导数如下：

$$\nabla^2 A_s(r) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(r - r_j, h) \quad (2-7)$$

SPH 方法可以保有基本的物理条件（如能量守恒），但是 SPH 算法也有自身的问题无法保证所有的条件（如力的对称性）。

2.1.2 对流体用粒子视角进行建模

在对等温流体（isothermal fluids）的建模中，网格方法使用欧拉视角，用速度场 v 、密度场 ρ 和压力场 p 描述流体。物理量的更新基于以下两个公式：

$$\frac{\partial m}{\partial t} + \nabla \cdot m = 0 \quad (2-8)$$

将质量展开 $m = \rho v$ ，公式 2-8 可以改写为

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0 \quad (2-9)$$

公式 2-9 可以保证质量守恒。

另一个公式为 Navier-Stokes 公式^[12]，用于保证动量守恒：

$$\rho \left(\frac{\partial v}{\partial t} + v \cdot \nabla v \right) = -\nabla p + \rho g + \mu \nabla^2 v \quad (2-10)$$

公式 2-10 为一个针对不可压缩流体的简单的 Navier-Stokes 公式形式。其中 g 为外力场，在大多数情况（包括本文），其表示重力场； μ 表示流体的黏度，为一个常量系数。



无网格方法（粒子方法）使用拉格朗日视角。以上两个公式（公式 2-9、公式 2-10）可以被大幅度简化。某个粒子 i 的加速度可以如下表示：

$$a_i = \frac{dv_i}{dt} = \frac{f_i}{\rho_i} = \frac{-\nabla p_i + \rho_i g + \mu \nabla^2 v_i}{\rho_i} \quad (2-11)$$

其中 v 表示粒子的速度， f 表示粒子收到的力。公式 2-11 已经基本阐明了 SPH 计算的方法，接下来我们只需要解释该公式的计算细节。

2.1.3 压力计算（Pressure）

基于公式 2-4，压力场 $-\nabla p_i$ 可以表示为：

$$pressure_i = -\nabla p_i = -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(r_i - r_j, h) \quad (2-12)$$

公式 2-12 是由公式 2-4 直接推导而来，但是由于 SPH 算法本身并不能保证力的对称性，公式 2-12 亦不能保证压力的对称性。因此在本文实现的 SPH 算法中对压力的计算进行了调整在插值中引入算数平均：

$$pressure_i = -\nabla p_i = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(r_i - r_j, h) \quad (2-13)$$

对于以拉格朗视角的 SPH 方法而言，粒子本身所保有的性质只有自身的位置（position），压力经由周边的粒子插值得到。如果所有的粒子压力都要通过周边的粒子压力插值得到，这将无法计算。因此在实际计算某个粒子处的压力值时，SPH 算法先对该粒子的数值进行估计，之后再行插值计算。

压力的计算基于理想气体方程：

$$p = k(\rho - \rho_0) \quad (2-14)$$

其中 ρ_0 为流体的静止密度，在 SPH 算法中为需要手动调整的超参数。其本身仅仅为了是的 SPH 算法的数值计算更加稳定，对压力梯度的计算没有影响^[13]。

2.1.4 黏度计算（Viscosity）

SPH 方法中黏度的计算同样基于公式 2-4 得到：

$$viscosity_i = \mu \nabla^2 v_i = \mu \sum_j m_j \frac{v_j - v_i}{\rho_j} \nabla^2 W(r_i - r_j, h) \quad (2-15)$$

类似于公式 2-13，公式 2-15 也经过调整以保证力的对称性。

2.1.5 外力（External Forces）

SPH 算法中的外力包括重力和其他外部施加的力。这些力都会被直接施加到粒子上。在本文中所使用的力包括重力和在粒子接触到边界时受到的反弹力。

2.1.6 光滑核（Smoothing Kernels）

在 2.1.2、2.1.3、2.1.4 中均使用了光滑核 W 来进行插值。 W 的选择在满足 2.1.1 中的要求时可以是任意的（甚至不需要完全满足），在 SPH 方法中很多时候使用高斯方程。光滑核的选择对于 SPH 方法是至关重要的，直接关系到 SPH 方法是否稳定、准确但在本文中，为了使计算更加稳定，使用三种不同的光滑核来应对不同的情况。

$$W_{\text{poly6}}(r, h) = \frac{15}{64\pi h^9} \begin{cases} (h^2 - r^2)^3 & 0 \leq r \leq h \\ 0 & \text{otherwise} \end{cases} \quad (2-16)$$

$$W_{\text{spiky}}(r, h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3 & 0 \leq r \leq h \\ 0 & \text{otherwise} \end{cases} \quad (2-17)$$

$$W_{\text{viscosity}}(r, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1, & 0 \leq r \leq h \\ 0, & \text{otherwise} \end{cases} \quad (2-18)$$

其中 W_{poly6} 用于计算流体的密度。 W_{spiky} 用于计算流体的压力。 $W_{\text{viscosity}}$ 用于计算流体的黏度。

需要特别提到的是，在公式 2-13 中实际使用到的为 W_{spiky} 的一阶导数：

$$\nabla W_{\text{spiky}}(r, h) = \frac{15}{\pi h^6} \nabla(h - r)^3 = -\frac{45}{\pi h^6} (h - r)^2 \quad (2-19)$$

在公式 2-15 中实际使用到的为 $W_{\text{viscosity}}$ 的二阶导数：

$$\nabla^2 W_{\text{viscosity}}(r, h) = \frac{45}{\pi h^6} (h - r) \quad (2-20)$$

SPH 方法的光滑核选择主要是为了能够适应流体的物理特性。如若使用 W_{poly6} 计算流体的压力，则会出现 W_{poly6} 在 $r \rightarrow 0$ 时梯度趋近于 0，然而在流体力学中某一点的压力应当主要取决于该点附近小范围内的粒子，因此光滑核需要经过特殊设计以解决这一问题。

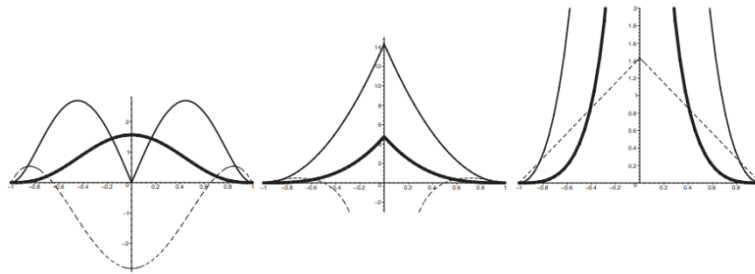


图 2-1 三个光滑核^[6]

图 2-1 中三个光滑核分别为 W_{poly6} 、 W_{spiky} 和 $W_{\text{viscosity}}$ 。其中粗线表示光滑核，细线表示光滑核的在朝向中心方向的梯度，虚线为表示拉普拉斯算子。



2.2 机器学习方法

2.2.1 神经网络

深度学习是机器学习的重要分支之一，是一种基于神经网络架构，数据驱动进行特征学习的算法。深度学习主要使用人工神经网络（Artificial Neural Networks），最早的神经网络由 McCulloch 和 Pitts 提出用模仿人类的大脑行为方式来解决复杂问题^[16]。

一个神经网络由众多人工神经元组成，人工神经元是采用了生物神经元的概念的可以执行计算任务的电路。数据通过输入层进入神经网络然后进入人工神经元（hidden layer）中，在此中进行计算并得到结果。总体上神经网络可以处理结构化的数据，产出特定的结果。

神经网络分为有监督学习，无监督学习和强化学习，在本节中只考虑本文使用到的有监督学习网络。

经由神经网络，计算机可以直接学习数据特征，并进行预测而不用进行知道数据的统计结果^[17]。神经网络使用表示学习（又名，特征学习），将输入特征映射到输出。映射过程中使用的是多层连接神经元。每个神经元都是一个数学计算单元，可以以特定的方式处理输入数据。神经网络所学习到的特征被保存在神经元的权重中。当输入信息在神经网络各层中传递过后，基于样本的标签和使用者定义的损失函数，我们可以计算神经网络对于此样本的损失。损失函数一般基于具体的模型，是一种重要的方法来评估模型的效果并引导训练的目标。

2.2.2 图神经网络（GNN）

数据驱动的神经网络模型在流体模拟中已经取得了较好的成果，其可以对一个复杂动态系统进行降阶的方式得到显著的效果提升^[15]。神经网络非常善于处理适合被离散化的、规则的网格数据，但是在拉格朗日视角下的 SPH 方法中，粒子并不能被表示为结构化的基于网格的数据，粒子之间的关系是动态变化的。对于这不规则的数据，一般使用图结构表达，图结构的表示更能够粒子的特征和粒子之间的潜在的互动。图数据的复杂性在传统的机器学习算法中是特别困难的。图结构本身就是不规则的，图的节点可能是无序的、大小可变的，图中的阶段的邻居可能是不确定的。因此很多特殊操作（例如卷积）在欧式空间能容易实现，却在非欧空间的图域上难以实现。此外机器学习中一个重要的核心假设，实例（节点）之间相互独立，在图数据中不能使用（图各个节点都会通过各种方式连接）^[18]。

在图神经网络之中，图卷积网络（GCN）是其中重要的一个分支，同时也是本文网络的基石。

在数学上，卷积的定义如下。

$$(f * g)(t) = \int_{\mathbb{R}} f(x)g(t-x)dx \quad (2-21)$$

其中 g 可以被称为 filter。

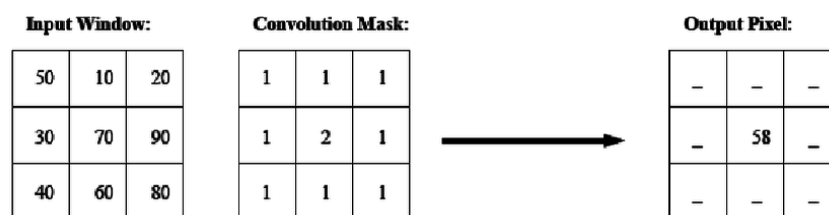


图 2-2 二维平面卷积示意图^[22]



在欧几里得空间内，如图像，卷积的操作非常清晰，我们可以直接将公式 2-21 离散后计算。但是在非欧几里得空间中，如图结构中，节点之间没有明确的空间关系，因此无法通过上面的传统卷积公式计算。在空域内难以计算往往会迁移到频域进行计算。因此我们使用傅里叶变换 \mathcal{F} 将卷积改为定义在图上。

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\} \quad (2-22)$$

我们可以定义图上的二阶导数拉普拉斯矩阵为 $L = D - A$ ，其中 D 为度矩阵， A 为邻接矩阵。标准化后的拉普拉斯矩阵为：

$$L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \quad (2-23)$$

此时拉普拉斯矩阵为半正定矩阵， L 自然可以分解为：

$$L = U\Lambda U^T \quad (2-24)$$

其中 U 为拉普拉斯矩阵的特征向量矩阵， Λ 为由对应特征值组成的对角线矩阵。在此基础上可以定义图上的卷积：

$$g_\theta * x = Ug_\theta U^T x = Ug_{\theta'}(\Lambda)U^T x \quad (2-25)$$

其中 $g_{\theta'}(\Lambda)$ 的计算是非常复杂的，因此在图卷积网络和本文中均使用 Chebyshev 多项式进行近似（不同的是本文所使用的多项式更高阶，但是对结果影响很小）。若使用一阶（ $K=1$ ）的 Chebyshev 多项式，图卷积可以写作：

$$g_{\theta'} * x = \theta \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \right) x \quad (2-26)$$

其中 $\tilde{A} = I + A$ ， I 为单位矩阵， A 为邻接矩阵。

本文中由于引入了时序网络，所以需要些许修改，但是总体上与上述相同。

2.2.3 时序神经网络（GRU）



门控循环单元（GRU）为循环神经网络（RNN）中的一种，和 LSTM 结构类似，由于本文的核心之一便在于使用 GRU 对输入数据进行预处理，提取粒子之间的关系，因此需要先在此对 GRU 的一些特性进行说明。

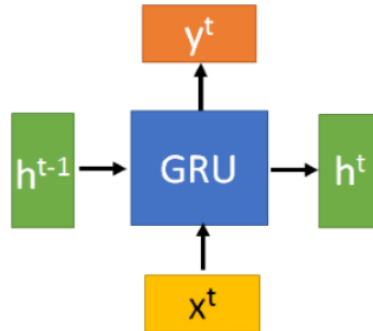


图 2-3 GRU 节点示意图^[23]

标准 GRU 在处理单个时序序列信息时，输入将会在时间维度被分开输入到每一个 GRU cell 中，因此最终的隐藏状态输出可以视为在时间维度对时序序列信息的特征提取。下文中将使用 GRU 类似的特征方法进行特征提取。



第三章 模型构建

3.1 SPH 算法实现

在本文中 SPH 方法的实现基本与在 2.1 章节中实现的方法相同。本文中的 SPH 方法在 Unity2022 平台中实现。Unity 使用 C#脚本控制逻辑，并可以实时地渲染计算结果。基于 Unity 自身的设计特性，本文使用 Unity 中 ComputeShader 进行 SPH 方法中粒子状态的更新。在将深度学习方法嵌入 SPH 中后，将会在 python 端建立 socket server，unity client 作为客户端发送和接收数据，从而实现和 python 的交互。

SPH 方法在处理容器的边界问题和与刚体的碰撞问题有多种解决方法^[14]，其中常用的方法有镜像粒子法（在边界处镜像布置粒子）和虚拟粒子法（在边界布置多层虚拟粒子，实际中使用较多），容易造成穿透，如果想要保证不会发生穿透需要布置的粒子足够密集，同时保证计算的步长足够小，这对于图神经网络而言计算过于复杂，故本文中 SPH 方法中使用硬边界处理，粒子在接触到容器边缘后会直接反弹。

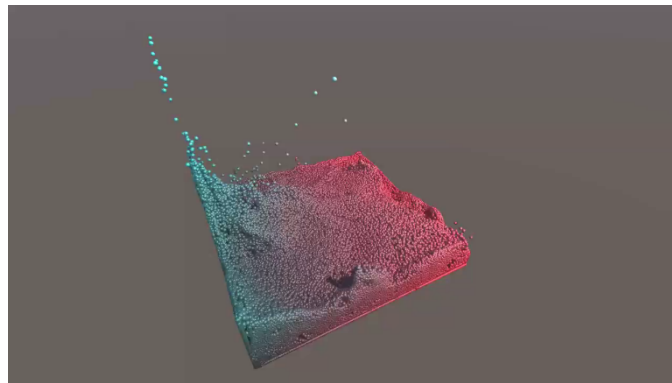


图 3-1 SPH 渲染结果

如上图所示，本文使用的 SPH 方法在容器内部表现稳定，但是在容器边界处，模型表现有所欠缺。

3.2 神经网络结构

本文中的神经网络输入为时序的多粒子特征数据，输出为下一次更新后的粒子位置相对于当前粒子位置的差值。

下图为神经网络结构图。

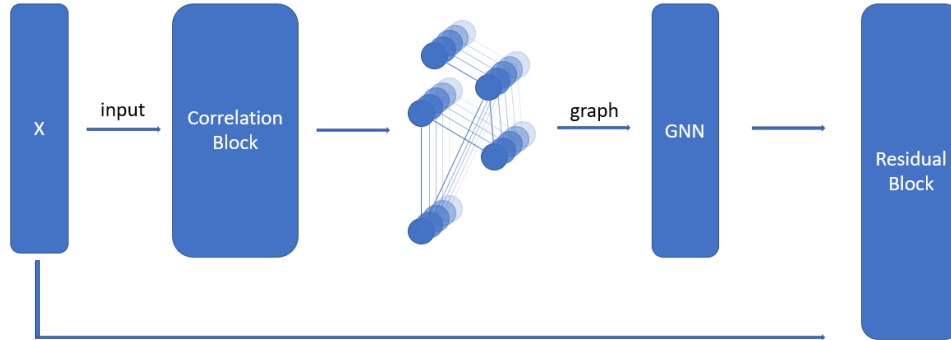


图 3-2 神经网络结构图

最初的输入维度为 $X \in R^{S,N,P}$ ， S 为时间序列长度， N 为粒子数量， P 为每个粒子的特征数量。

在本文中，考虑了多元时间序列的预测问题。我们的网络可以同时接受多个粒子时序特征的输入，并将它们视为多个并行的互相影响的事件。我们的网络目标是为了能够在特定的时序窗口期内学习该系统的时间之间和事件之间的相关性。

本文所采用的神经网络，其同时考虑了多个时间序列，并对其建模形成图节点，并尝试在其中对每个事件之间的交互进行建模形成图边。在大多数 SPH 算法中使用的图神经网络都忽略了时间序列之间的相关性，同时需要假设图关系的先验知识，而获得图关系先验的方法往往是独立于神经网络之外的，因此神经网络损失了大量重要的信息。

在频域实现的卷积神经网络往往需要先验的图邻接矩阵。在本文中尝试将获得图邻接矩阵的方法纳入神经网络之中，采取一种类似自我注意的机制来捕捉节点特征之间的关系。此图邻接矩阵将直接被纳入图神经网络中，在一定程度上，它表示节点之间的特征关系。

模型主要分为 3 个模块：相关矩阵模块（Correlation Block），图模块（GNN Block），残差模块（Residual Block）。

相关矩阵模块中时间序列会被输入到门控循环单元（GRU）中来学习不同序列之间的时间表示，因此此 GRU 的使用和传统的 GRU 使用略有不同，GRU 单元的每个隐藏状态（hidden state）其实是每个时间序列在时间上的聚合信息。我们仅仅使用最后一个隐藏状态来进行后续计算。类似于自注意力机制，我们基于 GRU 最后一个隐藏状态计算可学习的 Q、K 注意力矩阵。注意力矩阵本身计算了不同序列之间隐藏状态的相关性，注意力分



数可以在后续构建的图中作为节点之间的边。注意力矩阵在经过对称化后即成为了邻接矩阵 A 。

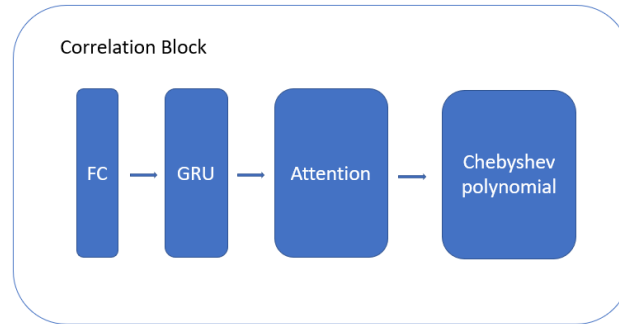
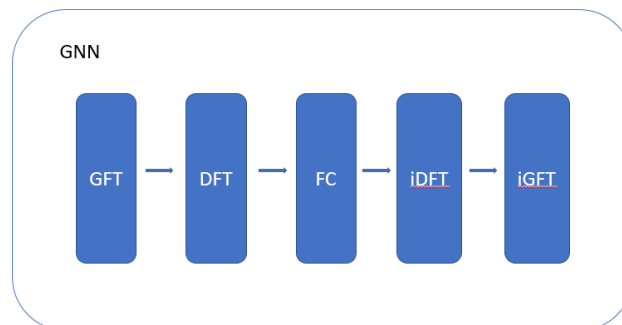


图 3-3 相关矩阵模块结构图

在相关矩阵模块中可以在进入 GRU 之前加入一层线性连接层，具有类似 **particle embedding** 的效果，使得模型特征提取效果有细微提升。在 GRU 中本文将特征维度和时间序列维度调换，在输入网络后得到 $R^{N,N}$ 的特征矩阵，实际上根据 2.2.3 章节中的理解，此处提取的特征是在时间序列和粒子特征空间同时两个空间下提取特征。

此后，我们计算归一化后的拉普拉斯距离： $D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}$ ，其中 D 为邻接矩阵行 A 求和得到的对角矩阵。之后基于拉普拉斯矩阵，我们可以用切比雪夫多项式用以计算近似的正交图谱表示。



GFT--Graph Fourier Transform
DFT--Discrete Fourier Transform

图 3-4 图神经网络模块结构图



图模块中将输入为图结构，其中图的节点为网络的输入，图的边为相关矩阵模块中经过切比雪夫多项式近似得到的正交图谱矩阵，图的节点关系都将被投影到正交图谱上。在图卷积网络中到此为止，我们就可以在此频谱上实现图卷积滤波器。到此为止，图模块的内容和基本的图卷积网络完全一致。但是由于我们的网络使用了时序的输入，图卷积的傅立叶变换仅仅能捕获序列之间的关系，序列内时序的关系并没有被考虑在内^[19]。我们使用一种简单的方式来解决。通过使用离散傅立叶变换将单个序列的时域表示转化为频域表示。在频域中我们使用线性变换用于聚合时序的信息。此后变换后的特征会经过逆变换再次从频域表示映射回时域表示。在此基础上即可以类似传统图卷积网络一样，继续进行图卷积。



第四章 实验设计

4.1 数据集收集

数据集基于 SPH 方法中计算的结果。在此前的 SPH 和图神经网络结合的研究中，输入通常为粒子当前的位置、粒子当前的速度、边界的位置。在本文中由于采用了时序的网络，因此粒子的速度可以通过时序的粒子位置得到，因此不会作为输入。同时，由于本文没有实现虚拟粒子法处理 SPH 方法的边界问题，因此也不需要输入边界的位置。由 SPH 理论上，粒子密度可以通过根据粒子的位置和周围粒子的密度插值得到，但是由于粒子密度收到初始值的影响，因此神经网络不能清晰地得到粒子的密度。因此粒子密度也被作为输入进行神经网络中。

在 Unity 中，在 SPH 算法每次更新前，所有粒子位置、所有粒子的当前密度被记录；在 SPH 算法更新后，再次收集所有粒子位置、所有粒子密度。所有的数据紧密排列，以二进制格式、float32 精度保存。

由于收集得到的数据本身就有较大的干扰和错误，因此在收集数据时特意避免了初始状态后一段时间内的数据，待数值稳定后再收集数据以保证训练结果受到初始化参数数值影响小。

4.2 训练过程

数据集被划分为 50%、30%、20% 作为训练集、验证集和测试集。在训练中模型结构进行了多次微调，特别是对残差模块的调整，并尽量尝试在残差模块和图神经模块中得到一个平衡。神经网络的训练过程以均方根误差（RMSE）作为评价标准。然而在 SPH 方法中，RMSE 并不是一个好的评价标准，因此模型结果将会嵌入 SPH 方法，SPH 方法实际渲染出的结构也是重要的评价标准。

图神经网络需要把所有的粒子数据全部纳入训练，并由于增加了时间序列模块，模型的容量大为提升。相较之下数据量相对较小。原始 SPH 数据收集时，由于边界状态的不稳定，再边界状态收集到的数据可能带有大量干扰。因此模型在训练时，很有可能造成数据过拟合。更具体地说是模型对于边界状态可能会出现过拟合（在实际训练中此问题更为突出，本文通过修改模型内部实现细节和增加残差模块使得此现象有一定程度缓解）。

因此本文在训练时训练 epoch 较少，避免出现出现过拟合。

4.3 训练结果

以下内容展示了如何具体选择训练超参数并分析结果。

由于在图神经网络中引入了时序结构，神经网络的开销大为增加，因此在实际的训练中，batch size 的选择都较小，事实上，如果采取更大的 batch size，可能会在一定程度上避免模型对数据的过拟合，提升训练结果。



表1 不同batch size的训练结果

Batch size	RMSE (sum)
1	13.895962
2	13.844831
4	13.917532

在上述的训练结果中，batch size 为 2 的训练结果更好，但是以上训练的参数中 batch size 都较小，且差距不大，训练结果的差异受到数据扰动影响的可能性较大。因此如能适当增大 batch size，很有可能取得更好的训练效果，但同时也会大大增加训练内存的需求。

表2 不同epoch的训练结果

Epoch	RMSE (sum)
2	18.987665
4	13.844831
6	13.631237
8	12.881877
10	12.998324

在 batch size 为 2 的前提下，尝试了不同 epoch 的训练结果。在 epoch 大于 4 时，模型的收敛速度迅速下降。进一步增加 epoch 大小对训练结果影响微弱且模型有过拟合的风险。

表3 不同learning rate的训练结果

learning rate	RMSE (sum)
1e-3	13.928437
1e-4	13.844831
1e-5	365.975452

训练参数细节：

表4 训练参数

Hyper Parameter	Value
Learning rate	1e-4
Num epochs	4
Batch size	2
Loss function	MSE loss
Optimizer	RMSprop

在上述训练条件下得到：



表5 加入残差结构的训练结果

Residual block	RMSE (sum)
Yes	13.844831
No	9.631548

由于本文模型本身模型容量较大，因此在实际的训练中加入了残差模块避免模型对数据过拟合。残差模块的加入使得训练结果有明显提升。在 **SPH** 方法得到的粒子运动效果中也能看到粒子更加稳定。



第五章 结论

5.1 实验结果

本文在传统光滑粒子动力学（SPH）方法的基础上，加入了时序图神经网络对算法进行改进。神经网络采取时序数据作为输入，可以有效捕获粒子特征。在将神经网络嵌入 SPH 算法中后，粒子运动的结果渲染如下。

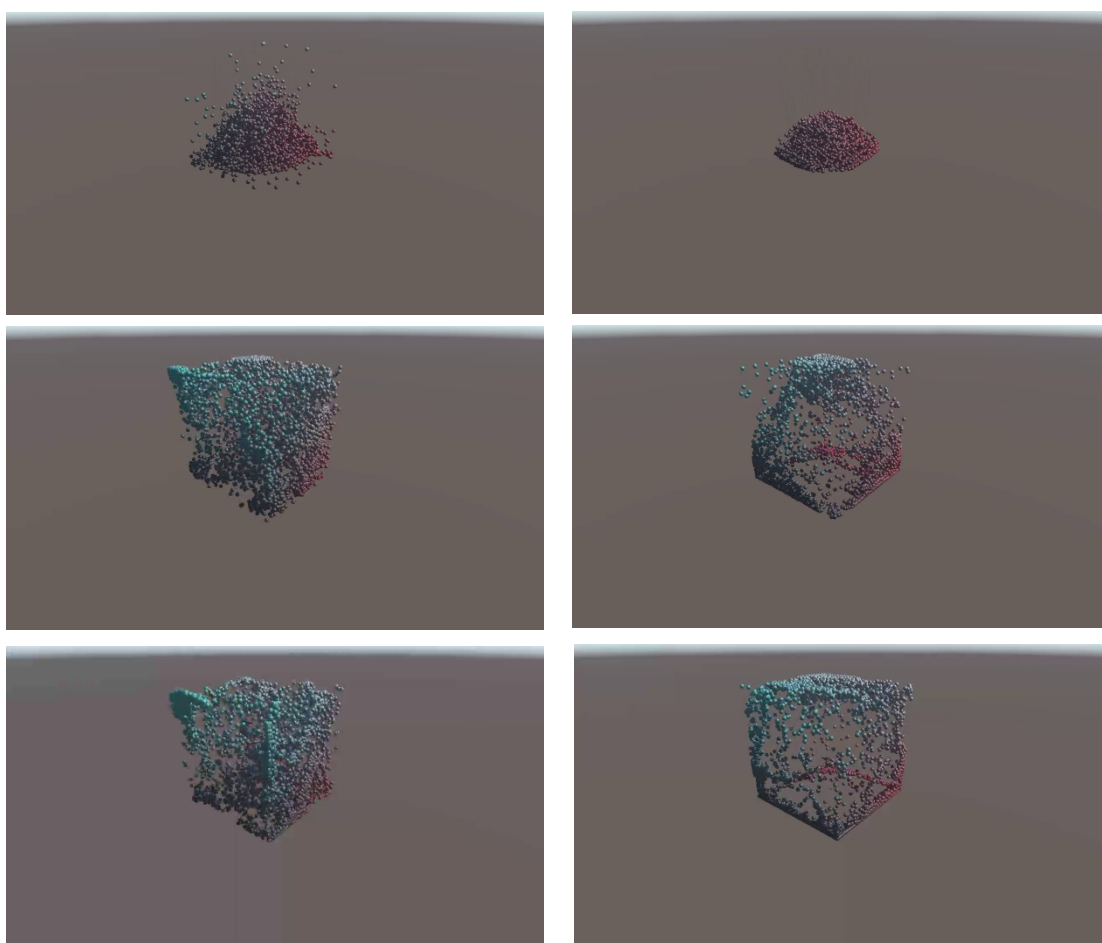


图 5-1 SPH 渲染结果 1

在图 5-1 中左侧为未添加残差模块的渲染效果，右侧为添加残差模块的渲染效果。两侧图片均在相同时间得到。

在渲染的前半部分出现了上文中提到的边界问题上的模型过拟合问题。在未添加残差模块的结果中，粒子长时间保持混乱的运动状态，短时间内无法收敛。在加入残差模块会这一现象大为缓解。粒子仍会出现非预期的运动，但是总体上相对稳定。

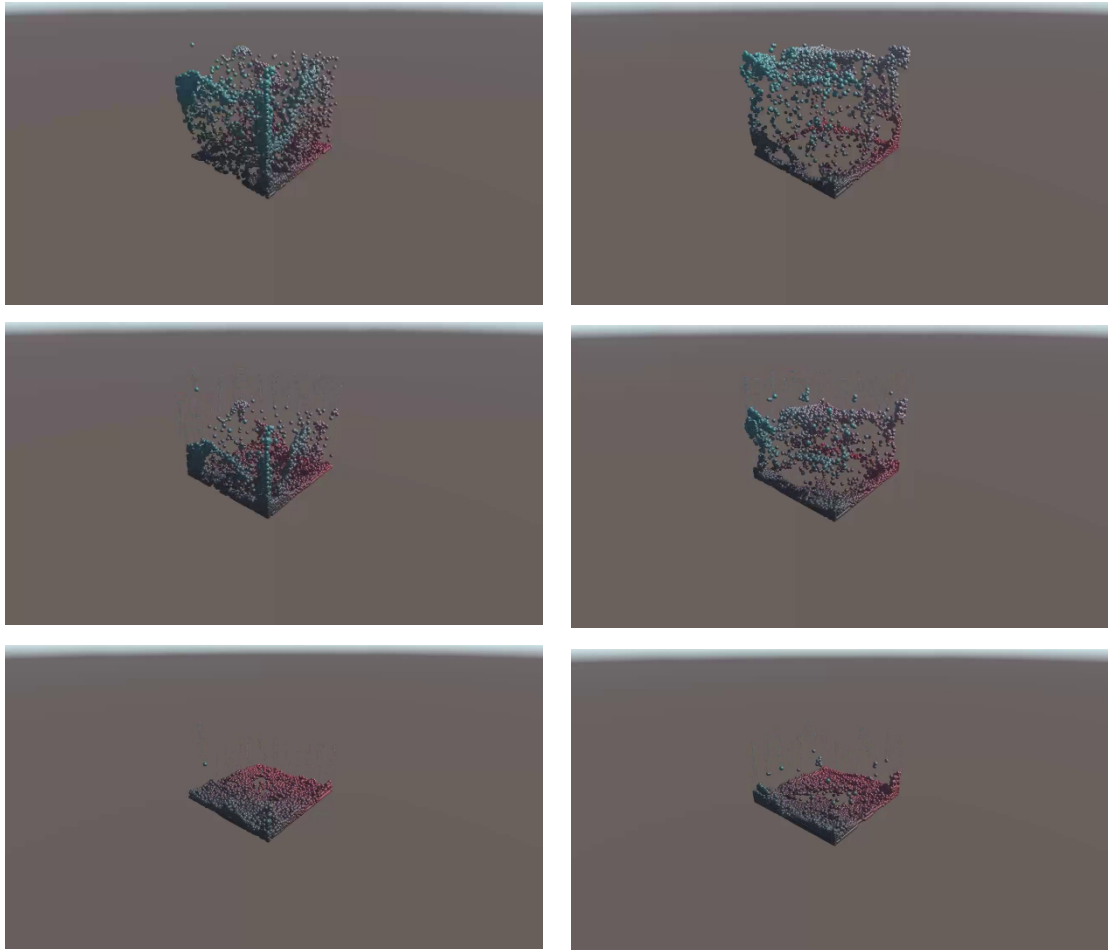


图 5-2 SPH 渲染结果 2

图 5-2 为图 5-1 的后续部分。

可以看到尽管不添加残差结构在相对稳定的流体状态下依然能够保持稳定。在此阶段两方面无明显可见差异。

5.2 实验缺陷

本文的模型设计仍有不少缺陷，预想的想法也有部分未能实现，故在此列出。

SPH 方法受到参数影响较大。细微调整参数即可对效果有较大影响。

本文所使用的模型内部细节仍有大量可以调整的空间。

使用图神经网络难以扩大粒子数量规模，粒子规模扩大对模型容量和模型训练难度都具有较大影响。是否有方法可以使用小模型适用于大规模粒子群

本文尝试采用小规模图卷积网络，针对每个粒子都独立进行推理。

SPH 方法本身会在粒子周围寻找邻居，利用邻居数值对自身进行插值，此尝试即类似用神经网络模拟此插值过程。

事实上模型无法达到稳定，但在粒子运动微弱时可以保持一定程度的稳定。

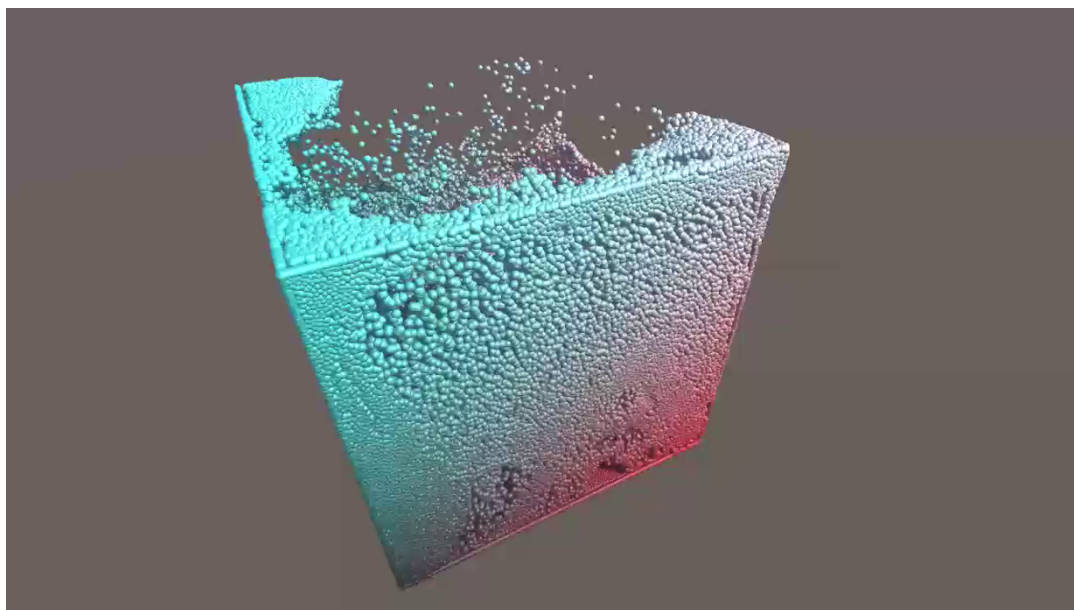


图 5-3 大规模粒子 SPH 渲染结果 1

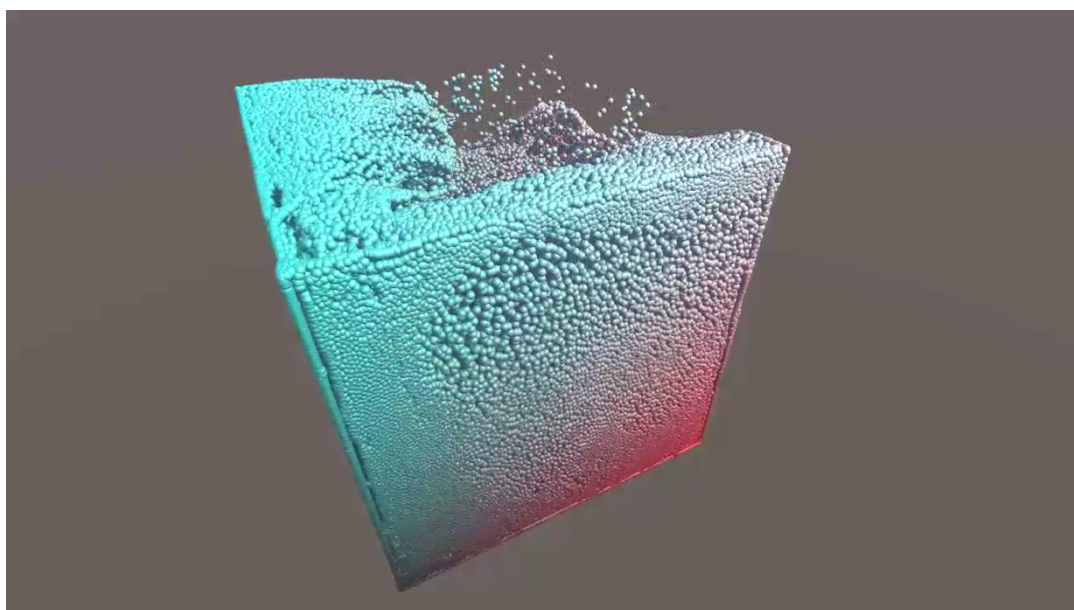


图 5-3 大规模粒子 SPH 渲染结果 2



参考文献

- [1] Jeremy Morton, Antony Jameson, Mykel J Kochenderfer, and Freddie Witherden. Deep dynamical modeling and control of unsteady fluid flows. In *Advances in Neural Information Processing Systems*, 2018.
- [2] J Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *ICLR*, 2019.
- [3] Gingold R A, Monaghan J J. Smoothed particle hydrodynamics: theory and application to non-spherical stars[J]. *Monthly notices of the royal astronomical society*, 1977, 181(3): 375-389.
- [4] 朱曼琳. 三种无网格法的对比研究[D]. 吉林大学, 2020.
- [5] Müller M, Charypar D, Gross M H. Particle-based fluid simulation for interactive applications[C]//*Symposium on Computer animation*. 2003, 2.
- [6] Ladický L, Jeong S H, Solenthaler B, et al. Data-driven fluid simulations using regression forests[J]. *ACM Transactions on Graphics (TOG)*, 2015, 34(6): 1-9.
- [7] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating Eulerian fluid simulation with convolutional networks. In *ICML*, 2017.
- [8] Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, 2016.
- [9] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin A. Riedmiller, Raia Hadsell, and Peter W. Battaglia. Graph networks as learnable physics engines for inference and control. In *ICML*, 2018.
- [10] Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li F Fei-Fei, Josh Tenenbaum, and Daniel L Yamins. Flexible neural representation for physics prediction. In *Advances in Neural Information Processing Systems*, 2018.
- [11] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *ICLR*, 2019.
- [12] D. Pnueli and C. Gutfinger. *Fluid Mechanics*. Cambridge Univ. Press, NY, 1992.
- [13] M. Desbrun and M. P. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation '96 (Proceedings of EG Workshop on Animation and Simulation)*, pages 61–76. Springer-Verlag, Aug 1996.
- [14] 郑坤,孙昭晨,张志明,周丰,赵西增.SPH 数值模拟中边界处理方法及对比分析[J].*水运工程*,2009,(08):23-27.
- [15] Wiewel Steffen, Becher Moritz, Thuerey Nils. Latent-space physics: Towards learning the temporal evolution of fluid flow. 2018, CoRR. arXiv:1802.10123.
- [16] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. *The bulletin of mathematical biophysics*, 1943, 5(4): 115-133.
- [17] De Veaux R D, Ungar L H. A brief introduction to neural networks[J]. Unpublished: <http://www.cis.upenn.edu/~ungar/papers/nnet-intro.ps>, 1997.



- [18] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. IEEE transactions on neural networks and learning systems, 2020, 32(1): 4-24.
- [19] Yuen K C, Chung S, Balsebre P, et al. Stock Price Prediction with StemGNN and Dynamic Time Warping[J]. 2021.
- [20] Autodesk 博客.(2017-05-23)[2022-05-31].https://blogs.autodesk.com/simulation/2017/05/23/where_we_stand_in_the_world_of_simulations/
- [21] Ummenhofer B, Prantl L, Thuerey N, et al. Lagrangian fluid simulation with continuous convolutions[C]//International Conference on Learning Representations. 2019.
- [22] Tech, M & Arya, Kavi. (2022). Low Cost – FPGA based system for pothole detection on Indian Roads.
- [23] 李宏毅.机器学习. (2017)[2022-05-31].<https://www.youtube.com/watch?v=T8mGfIy9dWM>



致谢

(正文内容)