

分类号: \_\_\_\_\_

单位代码: \_\_\_\_\_

密 级: \_\_\_\_\_ 无

学 号: 21621049

# 浙江大学

## 硕士学位论文



中文论文题目: 基于 GAN 的浅水方程模拟的  
超分辨率

英文论文题目: Shallow Water Equation Simulation's  
Super-resolution Using GAN

申请人姓名: 王天宇

指导教师: 黄劲 教授

专业学位类别: 计算机技术

专业学位领域: 计算机技术

所在学院: 计算机科学与技术学院

论文提交日期\_\_\_\_\_

# 基于 GAN 的浅水方程模拟的

## 超分辨率



论文作者签名: \_\_\_\_\_

指导教师签名: \_\_\_\_\_

论文评阅人 1: \_\_\_\_\_

评阅人 2: \_\_\_\_\_

评阅人 3: \_\_\_\_\_

评阅人 4: \_\_\_\_\_

评阅人 5: \_\_\_\_\_

答辩委员会主席: \_\_\_\_\_

委员 1: \_\_\_\_\_

委员 2: \_\_\_\_\_

委员 3: \_\_\_\_\_

委员 4: \_\_\_\_\_

委员 5: \_\_\_\_\_

答辩日期: \_\_\_\_\_

**Shallow Water Equation Simulation's  
Super-resolution Using GAN**

---



**Author's signature:** \_\_\_\_\_

**Supervisor's signature:** \_\_\_\_\_

Thesis Reviewer 1: \_\_\_\_\_

Thesis Reviewer 2: \_\_\_\_\_

Thesis Reviewer 3: \_\_\_\_\_

Thesis Reviewer 4: \_\_\_\_\_

Thesis Reviewer 5: \_\_\_\_\_

Examining Committee Chair: \_\_\_\_\_

Committeeman 1: \_\_\_\_\_

Committeeman 2: \_\_\_\_\_

Committeeman 3: \_\_\_\_\_

Committeeman 4: \_\_\_\_\_

Committeeman 5: \_\_\_\_\_

Date of oral defence: \_\_\_\_\_

## 浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 浙江大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名： 签字日期： 年 月 日

## 学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 浙江大学 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名： 导师签名：  
签字日期： 年 月 日 签字日期： 年 月 日

## 摘要

超分辨率任务一般是指从低分辨率的图像或者视频估计其对应的高分辨率版本，在安防监控、医学影像、视频图像压缩传输等方面具有广泛应用，因此超分辨率是一项非常重要的技术。因为其不适定性，当超分辨率的上采样因子较大时，该任务变得十分困难。传统的超分辨率技术一般基于在空域或频率域中构造插值或者投影算子来实现锐化；在如今深度学习的浪潮中，人们试图通过数据驱动的方式来完成该任务，并取得了一系列成果。数据驱动的方法相比较传统方法具有自动提取特征、更好的可拓展性、结果更接近高分辨率真实值等优势，因此目前被广泛研究。而当前的基于深度学习的超分辨率技术主要应用于视频图像数据，而对于图形学中存在的具有和图像或者视频性质不同的数据来源如何进行超分辨率，尚未获得充分研究。这其中，浅水方程（SWE）模拟生成的数据是一种图形学中很常见的数据，常用于实时游戏场景中的流体模拟，在低码率的传输环境需要对原始场景进行降采样压缩传输，因此会损失大量高频细节，有效的浅水方程数据的超分辨率技术能够在减少带宽使用量的同时提高用户体验感，而如何有效地恢复原始数据的高频细节尚待探索。

考虑到深度学习在处理特定数据超分辨率任务上具有的优势，本文提出基于生成对抗网络（GAN）的生成模型进行浅水方程模拟的数据的超分辨率算法。通过对浅水方程数据超分辨率任务过程中独特性质的思考，考虑到了浅水方程模拟超分辨率重建在时间维度上的光滑性需求，超分辨率任务的旋转等价性需求，模拟数据可能为负值的性质，物理过程中速度信息对神经网络训练有影响等关键因素，本文提出了 1. 引入度量浅水方程模拟时间维度光滑性的时间鉴别器，2. 使用浅水方程速度场信息作为生成器的输入的一部分，3. 使用异步的基于全训练集梯度的训练方法，4. 使用旋转对称的数据扩充方法扩充训练集，5. 使用 Leaky ReLU 激活函数作为生成器的激活函数，代替一般采用的 ReLU 激活函数，从而在结果上能通过低分辨降采样数据推测出较多的高频细节。

本文进行了详细的实验，验证每一步技术选取的必要性和可行性，并讨论了对于浅水方程模拟数据进行超分辨率这一任务的困难所在和该方法的局限性。

**关键词：** 基于物理的深度学习，生成模型，计算机动画，流体模拟

## Abstract

Estimating the high resolution counterparts from the low resolution images or videos is mainly called as the super-resolution task. It has extensive applications in security monitoring, medical imaging, video and image compression transmission and so on, so super-resolution is a very important technology. Because its ill-posed nature, when the super-resolution's upscaling factors are high, this task becomes very challenging. The classical methods of super-resolution are mainly based on the construction of interpolation or projection operator to sharpen the data; with the rising of the deep learning algorithms, people try to achieve this task in a data driven manner and make some advances. Data-driven method has the advantages of automatic feature extraction, better scalability and the result closer to the high-resolution ground truth, etc. Therefore, the current super-resolution technology based on deep learning is widely studied. However, the current super-resolution technology based on deep learning is mainly applied to video or image data, but how to conduct super-resolution for data sources with different properties from images or videos in computer graphics has not been fully studied. The shallow water equation(SWE) simulation data is one of the most common data in computer graphics and it is often used for fluid simulation of real-time game scenes and in a low bit rate transmission environment, we need to downsample the original data to make the compression transmission, so it will lose a lot of details. Effective super-resolution technology for shallow water equation simulation data can improve user experience while reducing bandwidth usage. But how to recover the high frequency details from the the downsampled data has not been explored effectively.

Considering the advantages of deep learning in processing specific data super-resolution tasks, this thesis proposes to achieve the shallow water equation simulation data super-resolution using a kind of generative model named as generative adversarial networks(GAN). After considering the shallow water equation super-resolution task's uniqueness, thinking about that the smoothness demand on time dimension in the shallow water equation sim-

ulation data super-resolution reconstruction, rotation equivalence requirement for super-resolution tasks, the value of the simulation data may be negative, velocity information impact on the neural network training in the physics process are key factors, this thesis proposes 1.introducing the time discriminator to measure the smoothness of time dimension of the reconstructed high resolution results, 2.using shallow water equation velocity information as a part of the generator's input, 3. using the asynchronous training method based on full training set's gradient, 4.using rotation symmetric data augmentation method for training set, 5.using Leaky ReLU activation function as the activation function of the generator, instead of the commonly used ReLU activation function, so that more high-frequency details can be inferred from the low-resolution down sampled data in the results.

This thesis made a series experiments to verify the necessity and feasibility of technology selection at each step and discuss the challenge of the super-resolution task for shallow water equation simulation and our method's limitation.

**Keywords:** physics-based deep learning, generative models, computer animation, fluid simulation

# 目 录

摘要 .....	I
Abstract .....	III
图目录 .....	VIII
表目录 .....	X
第1章 绪论 .....	1
1.1 研究背景 .....	1
1.2 本文工作 .....	2
1.3 本文结构 .....	3
第2章 相关技术综述 .....	5
2.1 频率域方法 .....	5
2.2 空间域方法 .....	5
2.2.1 基于插值的方法 .....	6
2.2.2 基于重建的方法 .....	6
2.2.3 基于机器学习的方法 .....	8
2.3 图形学中的超分辨率任务的解决方案 .....	12
2.4 本章小结 .....	14
第3章 卷积神经网络、生成对抗网络和浅水方程 .....	15
3.1 卷积神经网络 .....	15
3.1.1 历史发展 .....	15
3.1.2 理论背景 .....	17
3.1.3 常见网络功能层 .....	17
3.1.4 经典模型 .....	21
3.2 生成对抗网络 .....	22
3.2.1 产生背景 .....	22
3.2.2 理论基础 .....	22

3.2.3 存在的问题.....	25
3.2.4 GAN 变种 .....	25
3.2.5 具体应用场景 .....	27
3.3 浅水方程 .....	28
3.3.1 物理背景 .....	29
3.3.2 数值解法 .....	29
3.4 本章小结 .....	32
第 4 章 算法设计和分析 .....	33
4.1 目标问题的数学描述.....	33
4.2 损失函数设计 .....	35
4.2.1 生成器的损失函数.....	35
4.2.2 空间鉴别器的损失函数 .....	35
4.2.3 时间鉴别器的损失函数 .....	36
4.3 GAN 结构 .....	37
4.3.1 生成器 .....	37
4.3.2 时间鉴别器和空间鉴别器.....	38
4.4 数据生成方法.....	39
4.4.1 浅水方程模拟的数据生成.....	39
4.4.2 旋转对称的数据扩充方法.....	39
4.5 GAN 训练方法.....	40
4.5.1 基于 mini-batch 梯度的训练 .....	41
4.5.2 异步的基于全训练集梯度的训练.....	42
4.5.3 训练和测试时使用的低分辨率数据 .....	42
4.6 本章小结 .....	43
第 5 章 实验结果展示和分析 .....	45
5.1 实验环境 .....	45
5.2 结果分析和对比 .....	45
5.2.1 使用基于 mini-batch 梯度的训练方法的问题 .....	47
5.2.2 是否使用低分辨率速度场.....	48
5.2.3 是否旋转对称的数据扩充.....	48
5.2.4 是否使用时间鉴别器 .....	49

---

5.2.5 生成器的激活函数的选取.....	50
5.2.6 训练方法的改进 .....	51
5.3 本文方法的局限性 .....	52
5.3.1 初值和边界条件的局限 .....	52
5.3.2 训练时间 .....	53
5.4 本章小结 .....	54
第6章 总结和展望.....	57
6.1 全文总结 .....	57
6.2 展望 .....	58
参考文献.....	59
攻读硕士学位期间主要的研究成果 .....	64
致谢 .....	65

## 图 目录

2.1	Tsai 和 Huang 算法 <sup>[1]</sup> 中低分辨率图像和高分辨率图像的关系 .....	6
2.2	FSRCNN 和 SRCNN 的结构对比 <sup>[2,3]</sup> .....	10
2.3	ESPCN 结构 <sup>[4]</sup> .....	10
2.4	DRCN 结构 <sup>[5]</sup> .....	11
2.5	LapSRN 结构 <sup>[6]</sup> .....	12
2.6	SRGAN 结构 <sup>[7]</sup> .....	13
3.1	哺乳动物视觉局部感受外部环境的特点 .....	16
3.2	神经认知器结构 <sup>[8]</sup> .....	16
3.3	LeNet-5 结构 <sup>[9]</sup> .....	17
3.4	ResNet 中的残差块结构 <sup>[10]</sup> .....	21
3.5	使用 GAN 生成手写数字 .....	22
3.6	随着训练，原始的 GAN 的生成器的输出的变化 <sup>[11]</sup> .....	23
3.7	原始的 GAN 训练过程 <sup>[11]</sup> .....	24
3.8	极大极小值问题求解对应寻找一个高维空间中的鞍点 .....	24
3.9	条件 GAN .....	26
3.10	通过 GAN 进行人脸去遮挡 <sup>[12]</sup> .....	29
3.11	错列网格上需要记录的变量值 .....	30
3.12	基于浅水方程的单相流体的模拟结果 .....	31
3.13	基于浅水方程模拟的流体和固体耦合的模拟结果 .....	32
4.1	典型的浅水方程的高分辨率数据和其降采样后的低分辨率数据 .....	34
4.2	本文的 GAN 结构 .....	37
4.3	高分辨率模拟结果 .....	40
5.1	使用基于 mini-batch 梯度的训练方法只能获得模糊的高分辨率恢复结果 .....	47

---

5.2 基于 mini-batch 梯度的训练方法在某些给定超参情况下会容易发散 .....	48
5.3 是否使用低分辨率速度场作为输入的对比结果 .....	49
5.4 测试集的特征和训练集的特征的相似性 .....	50
5.5 CNN 学习的特征不满足旋转等价性 .....	51
5.6 使用旋转对称的数据扩充后可以获得更好的泛化性 .....	52
5.7 使用时间鉴别器可以提高结果质量 .....	53
5.8 相比较 ReLU 作为生成器激活函数，使用 Leaky ReLU 可以获得更好的结果 ..	55
5.9 异步的基于全训练集梯度的训练方法可以恢复出更多的高频信息 .....	56

## 表 目录

5.1 实验结果的缩写对应 .....	46
---------------------	----

## 第1章 绪论

### 1.1 研究背景

图像或者视频的超分辨率任务是计算机视觉领域的一个重要任务，它一般指如何从低分辨率的图像或者视频中估计其对应的高分辨率数据。超分辨率具有不适应性，即数学上低分辨率到高分辨率是一个一对多映射，换句话说，会有众多的高分辨率数据通过下采样后会对应于一个相同的低分辨率数据。如何从一对多的映射中获取高分辨率近似性最高的映射不具有唯一性，因此有效地处理该问题事实上非常困难。尤其当超分辨率的上采样因子增大时，比如从低分辨率的1个像素到高分辨率的4个像素的超分辨率任务变成从低分辨率的1个像素到高分辨率的16个像素的超分辨率任务，其一对多的解空间随着上采样因子指数增大，使得该任务变得愈发棘手。但是另一方面，它却有巨大的现实意义，吸引着众多的科研人员寻求其在精度、效率和鲁棒性上的突破。对于占网络流量比例超过80%的图片或者视频数据进行超分辨率能够有效地减少流量传输和存储消耗，并且现实中通过摄像机、照相机或者其他仪器获得的图片或者视频数据可能本身就受限于环境、设备等因素分辨率不高，对其进行超分辨率处理可以有效地帮助后续分析，可以有效地提高后续高层次视觉任务比如图像识别、说明、分割等的准确率，在自动驾驶、医学影像分析等应用中有重要意义。

计算机视觉领域，对图片和视频数据进行超分辨率方法大致可以分为：1. 频率域方法，2. 空间域方法。其中，空间域方法又可以分为以下几类：2.1. 基于插值的方法，2.2. 基于重建的方法，2.3. 基于机器学习的方法<sup>[13]</sup>。而其中基于深度神经网络的数据驱动方法在图像视频的超分辨率上取得了重大发展，出现了众多不同的模型，如 SRCNN<sup>[2]</sup>, FSRCNN<sup>[3]</sup>, DRCN<sup>[5]</sup>, LapSRN<sup>[6]</sup>, SRGAN<sup>[7]</sup>。这是有赖于GPU等硬件革新和神经网络的训练方法的进步，使得神经网络的层数可以变得很深并且相对容易训练，让其在机器视觉和自然语言处理等领域大放异彩。但是另一方面，深度学习对图形学的影响则出现的较晚。原因之一在于，相比于图像或视频数据的规整矩阵表示且归一化到一定范围[0, 1](或者[0, 255])的数据格式，图形学中生成和处理的数据结构则更为复杂，除了绘制生成的图像或者视频数

据，还有如网格、点云、代表密度场的体素数据、代表高度场的类像素数据；且这些数据因为其生成方法的原理不同（如点云可能由扫描仪获得，网格、密度场、高度场可能由求解不同的偏微分方程获得），其多样性和复杂性远超过图像或者视频。

最近五年，深度神经网络在图形学中的研究热点主要集中于：1. 图像空间的草稿简化<sup>[14]</sup>，上色<sup>[15]</sup>，去噪<sup>[16]</sup>等；2. 探索让三维模型的各种表示（基于多角度图像、体素、网格、点云）如何更好地在神经网络的框架中发挥作用，以完成特定的任务，如模型分割<sup>[17]</sup>，模型生成<sup>[18]</sup>等；3. 使用神经网络对物理模拟数据的处理，如基于神经网络的角色控制<sup>[19]</sup>，利用神经网络数值求解物理方程<sup>[20]</sup>，物理数据的超分辨率<sup>[21]</sup>等。

在超分辨率问题上，数据驱动的方法相比较传统方法具有自动提取特征、更好的可拓展性、结果更接近高分辨率真实值等优势，因此目前被广泛研究。而当前的基于深度学习的超分辨率技术主要应用于视频图像数据，而对于图形学中存在的具有和图像或者视频性质不同的数据来源如何进行超分辨率，尚未获得充分研究。如上文所述，对图形学中的非图像或者视频数据进行超分辨率更为困难，需要考虑到其数据的独特性，才能更好地对其进行超分辨率重建。这其中，浅水方程（SWE）模拟生成的数据是一种图形学中很常见的数据，常用于实时游戏场景中的流体模拟，在低码率的传输环境需要对原始场景进行降采样压缩传输，因此会损失大量高频细节，有效的浅水方程数据的超分辨率技术能够在减少带宽使用量的同时提高用户体验感，而如何有效地恢复原始数据的高频细节尚待探索。

## 1.2 本文工作

考虑到深度学习在特定数据的超分辨率这一任务上具有一定的优势，如上文所述有自动提取特征、更好的可拓展性、结果更接近高分辨率真实值等优势，本文提出基于 GAN 的生成模型进行浅水方程模拟的数据的超分辨率，实现了从浅水方程数据从降采样低分辨率的 1 个单位到高分辨率的 16 个单位的超分辨率任务。基于深度学习的超分辨率方法可以从压缩后的低分辨率的浅水方程模拟的数据中恢复出较多的高分辨率数据的细节。

本文提出利用生成对抗模型（GAN）对浅水方程模拟的数据进行超分辨率。GAN 的结构包括一个基于残差块（Residual Block）的生成器和一个基于深度卷积神经网络（CNN）的度量单帧高分辨率高度场恢复质量的空间鉴别器和一个基于卷积神经网络的度量恢复高度场的时间连续性的时间鉴别器。通过训练 GAN，最终达到在一定的初值条件下，输入下采样的低分辨率数据给生成器，能够恢复出对应的高分辨率高度场，并恢复出许多细小的高频细节。

经过对浅水方程数据超分辨率任务过程中独特性质的思考，考虑到了浅水方程模拟超分辨率重建在时间维度上的光滑性需求，超分辨率任务的旋转等价性需求，模拟数据可能为负值的性质，物理过程中速度信息对神经网络训练有影响等关键因素，本文提出了 1. 引入度量浅水方程模拟时间维度光滑性的时间鉴别器，2. 使用浅水方程速度场信息作为生成器的输入的一部分，3. 使用异步的基于全训练集梯度的训练方法，4. 使用旋转对称的数据扩充方法扩充训练集，5. 使用 Leaky ReLU 激活函数作为生成器的激活函数，代替一般采用的 ReLU 激活函数，从而在结果上能通过低分辨降采样数据中推测出较多的高频细节。

本文对这一任务进行了一系列实验，从数据生成方式，数据扩充方式，神经网络结构和损失函数构造以及训练方式，激活函数的选取等多方面进行了探究，结合浅水方程的特性，获得了比已有方法更好的训练和测试结果，验证了每一步技术选取的必要性和可行性。

现有的超分辨率技术主要处理的是图像或者视频数据，但是对物理数据进行超分辨率尚未得到充分探究，不同的物理过程有其不同的性质，如何有效地利用这些性质对有效的神经网络训练十分重要，这不仅能缩短训练迭代次数，也是为了获得更好的泛化效果。

针对数值解浅水方程获得的数据进行超分辨率，本文相比较传统的图像或视频数据，有效地利用了速度信息，并构造了度量序列在时间维度上的光滑性的鉴别器有效地提高了训练和测试结果的质量。实验结果表明在使用神经网络处理物理数据的过程中，有效地利用这些物理数据非常关键，神经网络能够学到速度场和高度场之间的潜在的耦合关系，在相同的迭代次数下，获得更为光滑和精确的结果。

针对浅水方程这一特定物理模拟生成的数据，在一定的初边值条件下，本文提出的旋转对称的数据扩充方法，和异步地使用全局梯度进行训练的方法可以获得更好的泛化效果，恢复出更多的高频细节。

### 1.3 本文结构

本文一共分为六章，组织结构如下：

第一章为绪论部分，首先介绍了超分辨率技术在计算机视觉领域中的重要意义并简述了超分辨率技术的分类，着重介绍了基于神经网络的数据驱动的方法在该任务上的应用，并接着介绍了图形学领域的数据和视觉领域通常处理的图像和视频数据的区别，以及深度学习算法对图形学领域发展的影响和本文工作的意义所在，然后介绍了本文工作，最后是本文结构。

第二章对当前的超分辨率算法进行了文献综述，并对于最近几年快速发展的基于数据

驱动的超分辨率算法展开进行了更为详细的介绍。

第三章着眼于本文的研究主题：如何通过 GAN 模型对浅水方程模拟的数据进行超分辨率，介绍一些紧密相关的必要知识，包括深度卷积神经网络，常见的神经网络功能层，GAN 的理论背景、变种和应用，以及如何对浅水方程进行数值求解获得高分辨率数据等方面，为第四章介绍本文算法进行铺垫。

第四章详细介绍了通过 GAN 进行浅水方程模拟超分辨率的算法设计细节：包括神经网络结构的设计，损失函数的设计，数据生成方式的设计，数据扩充方式的设计和网络训练方法的设计。

第五章展示了第四章所述算法的实验结果，针对本文提出的改进验证了每一步技术选取的必要性和可行性，并最后讨论了本文方法的局限性和该研究问题的困难所在。

第六章是全文总结和展望。

## 第2章 相关技术综述

传统的超分辨率 (super resolution) 是一项增强图像分辨率的技术，通过一张或者多张低分辨率图像试图恢复其对应的高分辨率图像，是为了更好地抽取图像中的信息。在计算机视觉领域，对超分辨率方法的研究主要可以分为两类：1. 频率域方法，2. 空间域方法。本章将对这些方法中的代表性工作进行综述，最后介绍图形学中存在的超分辨率任务的解决方案。

### 2.1 频率域方法

图像或视频数据的超分辨率的主要目的是为了增强低分辨率数据的高频信息，以便后续的数据分析。自然而然，在频率空间处理低分辨率数据是一个合理选择。早期的研究集中于如何通过多张连续的图像恢复高分辨率图像，更类似于如今的防抖动应用的解决方案。在 1984 年，Tsai 和 Huang<sup>[1]</sup> 就提出了一个将多张低分辨图像通过离散傅里叶分解 (DFT) 转换到频率空间，恢复单张清晰高分辨率图像的算法，这可以认为是超分辨率任务的开山之作。其主要思路是通过将多张低分辨率采样图像的离散傅里叶分解系数和原始图像域的连续傅里叶分解系数联立起来构建方程组，方程组的解对应高分辨率图像应当的频率域系数，在通过反向傅里叶分解重构出高分辨率图像的空间域值，其核心思想如图2.1所示。他们的方法计算效率高，但是有非常强的假设：只能局限于多张低分辨率图像是在全局平移运动下生成的模糊图像，忽略了图像成像中的生成噪声，旋转，抖动等各种影响，很大程度上限制了其应用范围，因为现实应用大多不满足该假设。之后的基于频率域空间的重构方法也都存在着和 Tsai 和 Huang<sup>[1]</sup> 一样的整体平移的假设，因此这类频率域方法研究到目前为止尚没有很大的进展。

### 2.2 空间域方法

如上所述，频率域方法有着明显的不足，相比而言，空间域方法可以更好地使用先验知识来克服超分辨率问题本身的不适定性 (ill-posed)，可以通过引入先验知识约束低分辨

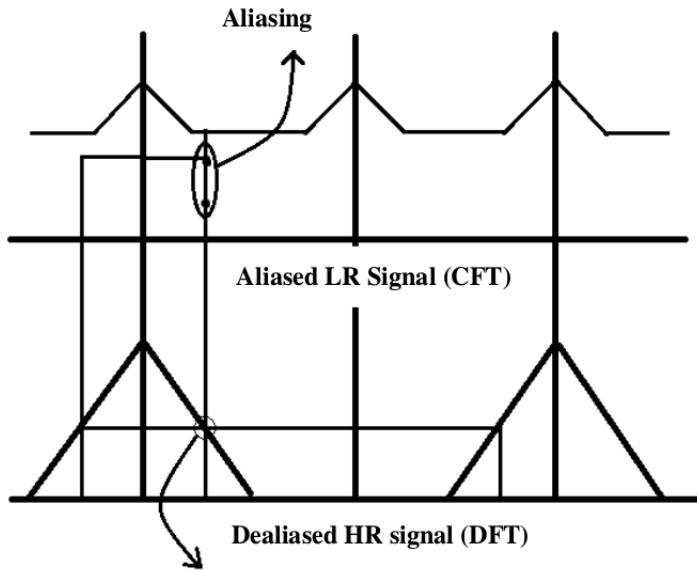


图 2.1 Tsai 和 Huang 算法<sup>[1]</sup> 中低分辨率图像和高分辨率图像的关系。图来自<sup>[1]</sup>率图像和高分辨率图像之间的对应关系。空间域方法主要可以分为：1. 基于插值的方法，2. 基于重建的方法，3. 基于机器学习的方法。

### 2.2.1 基于插值的方法

传统的插值方法假设图像中像素强度的变化连续且遵循一定规则，对此自然可以设计特定的内插公式将低分辨率图像转化到更好的高分辨率定义域上，常见的有最近邻插值，双线性插值，双三次插值等方法。这些方法可以实现一定程度的锐化，但是因为并没有引入新的高频信息，因此超分辨率的效果有限，如今经常作为深度学习一类方法的前处理阶段，将低分辨率数据先通过插值扩充到高分辨率定义域，再进行映射训练。

而在此之前，比较著名的纯基于插值进行锐化方法有 H.Aftab<sup>[22]</sup> 在 2008 年提出的混合插值方法：在边缘使用基于协方差的方法进行插值，而在光滑过渡的区域使用基于迭代曲率的方法进行插值。因此首先需要通过一定方法发现低分辨率图像的边缘和光滑区域，接着使用低分辨率图像的协方差系数估计高分辨率图像的对应系数进行插值。该方法相较于双线性插值、双三次插值、最近邻插值有更快的处理时间，更高的信噪比和更好的视觉质量。

### 2.2.2 基于重建的方法

除了上述的通过插值进行锐化的方法以外，通过多张低分辨率图像作为一组约束，结合一定的假设比如光滑性，进行低分辨率图像之间的像素之间的配准，再通过数值优化求

解高分辨率图像。

Irani<sup>[23,24]</sup> 提出的迭代式的反向投影方法是重建类的代表之一。通常首先给定一个低分辨率图像的插值作为高分辨率解的初值  $x^0$ , 然后根据公式2.1

$$y^0 = Hx^0 + n \quad (2.1)$$

估计其模拟的低分辨率图像, 其中  $n$  代表噪声,  $H$  是由下采样矩阵  $D$ , 模糊矩阵  $B$ , 运动矩阵  $M$  连乘获得, 即  $H = DBM$ , 如果  $x^0$  符合高分辨率参照, 且  $y^0$  符合参照  $y$ , 则算法停止, 否则将误差  $y - y^0$  投影到  $x^0$  上, 按照公式2.2进行修正。

$$x^1 = x^0 + H^{BP}(y - y^0) \quad (2.2)$$

但是由于需要事先已知  $H^{BP}$ , 而其计算无一定可靠的标准, 其结果并不可靠。

Schultz<sup>[25,26]</sup> 提出的最大后验概率方法则是典型的基于统计的数学模型, 他们假设低分辨率图像和高分辨率图像是两个不同的随机过程, 然后通过已知的最大后验概率准则, 根据公式2.3进行优化。公式2.3可以进一步变形为可以数值优化的形式2.4, 式中的  $Pr(x)$  有对于图像的先验假设获得。在具有有效的先验假设的情况下, 最大后验概率方法可以获得很好的锐化效果, 但是因为其优化中的计算量相对而言较大, 因此不能快速的进行高频信息推测。

$$X_{map} = argmax_x[Pr(x|y)] \quad (2.3)$$

$$X_{map} = argmin_x[-logPr(y|x) - logPr(x)] \quad (2.4)$$

Stark 和 Oskoui<sup>[27]</sup> 提出的基于凸集合投影的方法也可以完成一组低分辨率图像推测高分辨率的任务。它将高分辨率图像的解空间通过一系列约束划分成更紧的解空间, 从全空间某一点出发不断进行投影, 找到满足所有约束的解。显而易见, 这是基于投影的外点法的优化策略在超分辨率上的应用。其优点在于可以通过各种先验知识构造约束集合, 但是可能因为约束集合的构造不同以及初始点不同, 获得不一致解。

Nguyen 等<sup>[28]</sup> 提出的正则参数化的超分辨率重建算法将  $H$  矩阵中的模糊矩阵  $M$  参数化为  $B(\delta)$ , 因此  $H$  矩阵可以表示成  $H(\delta)$ , 通过构造优化泛函能量如下2.5, 优化对应的参数  $\delta$ ,

$$x, \delta, \phi = argmin(||y - H(\delta)x||_2^2 + \phi||x||_2^2) \quad (2.5)$$

其待优化的参数化的模糊矩阵需要满足一些要求，方能参数化，另一方面，正则项会在磨平噪声的同时，导致优化能量的第一项未能充分下降，导致模糊的现象。

基于重建的方法有着共同的特点在于，通过构造一些先验假设，通过数值优化的方法完成锐化过程，相比较插值方法而言，该框架更方便地使用图像或者视频的先验知识，构造优化目标，但是受限于要求多张低分辨率且需要进行低分辨率像素间的配准，其重建速度慢，结果不稳定，因此结果通常也不能满足实用需求，并且当超分辨比率较大时（如从1个像素到16个像素的映射），进行多张低分辨率像素配准本身就非常困难，因此会引入更多不确定性因素，结果更加不理想。

### 2.2.3 基于机器学习的方法

基于机器学习的方法逐渐成了超分辨率算法研究的热点，其希望通过构建训练集，通过训练集构造最佳的映射关系，以获得正确的先验知识来描述低分辨率到高分辨率的映射。当人工构造的先验假设不能满足高分辨率的精度需求时，机器学习的方法能够自发地获取更多的高层次信息，因此如果能构造恰当的训练集并且选取的学习模型合适时，该方法有更高的潜力获得更高的精度。一般而言，训练集同时包含低分辨率图像和对应的高分辨率图像，待训练的目标映射模型形式固定，通过“喂”给模型训练集，该模型可以学习到一组从低分辨率到高分辨率的映射关系，例如一个可学习的插值核函数，这相比较固定的插值方法（最近邻，双线性插值等）有更强的灵活性，并有生成更多高频细节的潜力。

Freeman 等<sup>[29,30]</sup>第一次提出了基于 Markov 网络进行超分辨率的模型。他将图像分成若干小块，通过训练集训练，获得低分辨率的小块到高分辨率小块之间的转移概率矩阵。这种方法为后来一系列的基于神经网络进行超分辨率的工作奠定了基础，相比较基于插值和基于重建的方法，这种方法能够生成更多的高频细节，但是这种基于数据驱动的方法，毫无例外都存在数据依赖的问题，需要手工选择恰当的训练集。

Chang 等<sup>[31]</sup>则是第一次应用流形学习的方法解决超分辨率。它假定低分辨率图像对应的解形状和对应的高分辨率的解形状在特征空间是具有相同结构的流形。通过训练可以获得不同分辨率的流形结构，那么自然而然此时可以对测试集合的每一个待估计的小块，寻找其在低分辨率的流形空间的 K 近邻表示，通过系数加权，再去高分辨率的流形空间寻找其近似表示作为最终结果。相比较 Freeman 等<sup>[29,30]</sup>的方法，该方法需要更小的训练集。

Karl 和 Nguyen<sup>[32]</sup>则在 2006 年提出了使用支持向量回归的方法解决超分辨率。他们构造了一些额外的约束条件，将传统的半正定优化问题转换成二次规划问题。测试时，首先对小块进行分类操作，再在类别里进行支持向量的回归，获得高分辨率图像。相比较其

他方法，该方法训练集可以很小，并且训练时间代价较小。

Yang 等<sup>[33]</sup>则提出使用构造完备字典进行稀疏学习的方法解决超分辨率。基于压缩感知理论，图像可以被一组稀疏系数向量和完备字典的内积重建，因此基于此假设，可以通过训练高分辨率和低分辨率图像小块字典，强化高分辨率图像和低分辨率图像的对应性和其字典稀疏系数之间的对应性之间的相似程度，从而可以用低分辨率的稀疏系数当做高分辨率对应块的稀疏系数的近似，再和高分辨率字典内积重建出高分辨率小块，从而完成超分辨率。该方法可以自动地确定字典大小和系数的维度，但由于压缩感知的假设，目前该方法只能使用于特定类别图像的超分辨率，通用性能不强。

机器学习方法处理超分辨率任务除了上述方法之外，还有诸如使用主成分分析等方法来解决该问题。另一方面，除了这些经典的学习方法可以建模超分辨率问题之外，神经网络也越来越体现出其在非线性拟合方法的巨大优势，而恰好超分辨率问题本身就是一个非线性过程，适合通过神经网络拟合潜在的映射关系。所以除了早期 Freeman 等<sup>[29,30]</sup>的工作之外，最近几年，络绎不绝地出现了多种基于神经网络的超分辨率模型，并且这些网络的层数都较深，训练方法也有所发展，这里都被归类于深度学习下的超分辨率方法。这些工作大多使用了卷积神经网络作为网络的核心结构，这是有赖于卷积层的权值共享和自动提取特征的优势，不仅能够减少神经网络训练变量的个数而且能够有效地避免人工选取特征的不当干扰。

Dong 等<sup>[2]</sup>在 2014 年提出了 SRCNN，这是使用 CNN 作为主体结构处理超分辨率的开山之作，实现单幅图像的超分辨率。其主体结构是先将低分辨率图像小块进行双三次插值到高分辨率尺寸，然后接着是三层卷积层：卷积核大小分别  $9*9$ ,  $1*1$ ,  $5*5$ ，通道数目分别是从图像通道数到 64, 64 到 32, 32 到图像通道数，填充参数为 0，而三层的作用分别是特征提取，特征进行重整，重整后的特征重建高分辨率图像。而目标函数则使用高分辨率参照和网络输出的欧氏距离的平方 (MSE) 作为目标损失函数进行训练。训练集是可以选取 ImageNet。该网络恢复的质量相比较传统方法在峰值信噪 (PSNR) 比和结构相似性指数 (SSIM) 两个指标上都表现更好。

之后在 2016 年，Dong 等<sup>[3]</sup>接着提出了 FSRCNN，这是对 SRCNN 提出的改进算法，层数上相比较 SRCNN 更深，并将原先的通过双三次插值将低分辨率图像映射到高分辨率空间的前置阶段改为使用同样可学习的反卷积作为最后一层来代替，同样实现了上采样的功能，综合看 FSRCNN 的层次共有 8 层，分别是：第 1 层是卷积核大小为  $5*5$  的卷积层，第 2 层是卷积核大小为  $1*1$  的卷积层减少特征通道数，第 3-6 层使用  $3*3$  的卷积核继续堆叠，第 7 层用  $1*1$  的卷积层增加特征通道数，最后一层用  $9*9$  的反卷积实现上采样。他们

还对数据通过旋转缩小进行数据扩充，最后仍然沿用均方误差作为损失函数，相比 SRCNN 其效果更好。二者的结构对比如下图2.2所示。

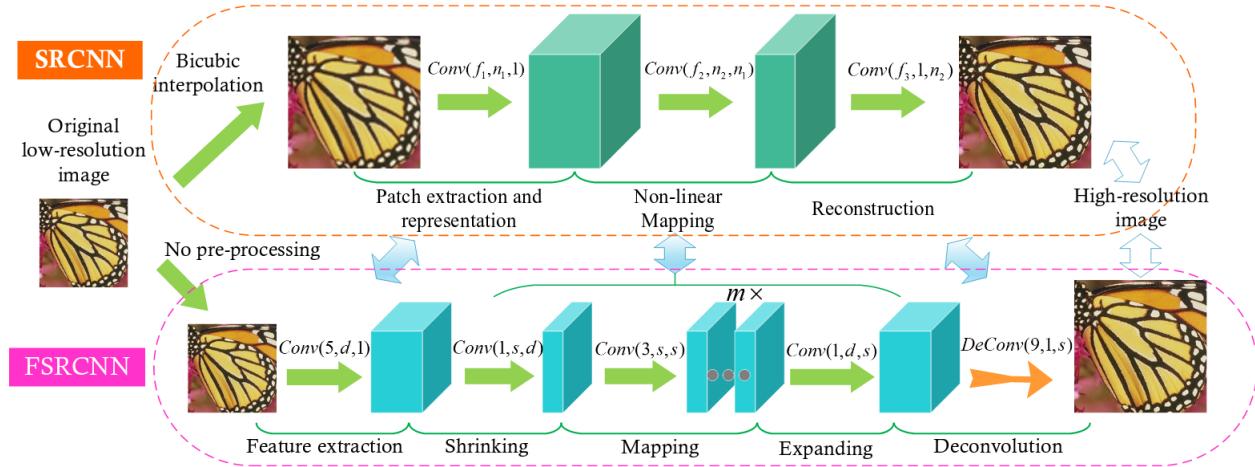


图 2.2 FSRCNN 和 SRCNN 的结构对比。图来自 [2,3]。

Shi 等<sup>[4]</sup>提出了 ESPCN 完成超分辨率，它首次提出了亚像素卷积层（sub-pixel convolutional layer）的概念，其动机可以认为是：在先前的工作如 SRCNN 中，低分辨率数据首先是通过一个固定的插值算子（如双三次插值）扩充到高分辨率空间，然后在高分辨率空间做卷积提取特征，重构操作，这意味着卷积核要覆盖的范围变大，在计算时会有更大的时间空间消耗，为此它使用了亚像素卷积层：首先在输入的低分辨率图像上做两次卷积提取特征且不改变特征图尺寸，但是通道数扩充为  $r^2$ ， $r$  是图像的放大倍数（如 2 倍，4 倍），然后将  $r^2$  个通道平铺成  $r * r$  的平面，这个平面就对应高分辨率的一个放大单位（4 个像素，16 个像素），这样就可以同理实现超分辨率。该想法利用了特征通道数作为放大倍数的等价替代，在峰值信噪比（PSNR）上比 SRCNN 有所提高，并且速度更快，其结构如图2.3所示。

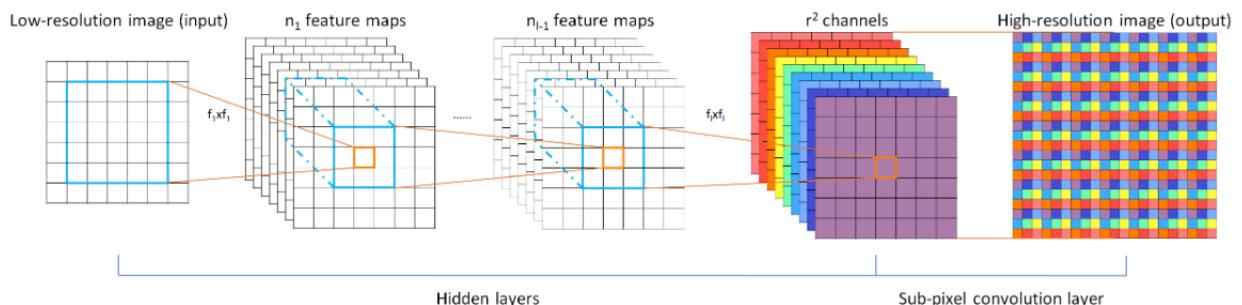


图 2.3 ESPCN 结构。图来自 [4]。

Tai 等<sup>[34]</sup>提出了 VDSR，VDSR 是将残差网络（ResNet）应用到超分辨率任务上的先锋工作，通过残差学习和跳层相加的思路搭建网络。残差网络的优势在于其克服了网络深度过深时，损失函数反向传播到末端的梯度消失，导致训练变慢乃至不可训练的严重问题。另一方面，残差学习也符合超分辨率任务的性质，输入的低分辨率图像和高分辨率参照本身就是相似的，他们之间在低频部分相似，只不过高分辨率携带了更多的高频信息，在训练时残差网络的跳层可以直接拷贝已有的低频信息，因此网络只需要学习缺失的高频分量即可。它继承了残差网络的优势，深度更深（将网络搭建到了 20 层），在训练时间上能更快收敛。

Kim 等<sup>[5]</sup>提出了使用深度递归卷积神经网络 (DRCN) 完成超分辨率，是第一次在超分辨率任务上使用深度递归卷积神经网络。相比较 SRCNN 的三层网络，Kim 等<sup>[5]</sup>提高了网络层数，同时为了避免层数增加导致的训练参数过多的问题，其使用了递归神经网络结构缓解。和 SRCNN 类似，其可以分成三个部分，第一个嵌入网络，由递归网络构成，它使得多个级连的卷积层可以共享一组参数，减少了训练变量个数，为完成特征提取的功能；第二个是推测网络，它是由 D 个共享参数的串联卷积层组成；第三个网络是重建网络，二中的 D 个卷积层会直连到三中相加取平均输出作为重建的高分辨率图像，其结构如图2.4所示。目标函数是 D 个卷积层输出的 MSE 误差和最终均值输出的 MSE 误差加权平均获得。

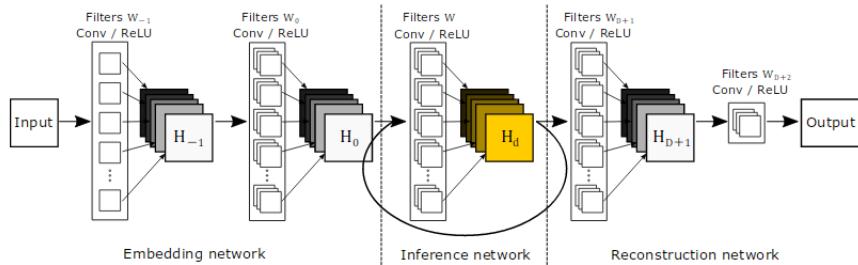


图 2.4 DRCN 结构。图来自<sup>[5]</sup>。

Lai 等<sup>[6]</sup>提出了 LapSRN，该方法基于拉普拉斯金字塔概念。它以低分辨率作为输入，而非直接将低分辨率直接插值到目标高分辨率尺寸，它逐渐地增加图像尺寸（例如要实现八倍超分辨率，则构造  $\log_2 8$  层“金字塔”网络结构），每层“金字塔”包括若干层卷积层实现特征提取和一层转置卷积层实现一倍的分辨率的增加，其结构如图2.5所示。每一层“金字塔”的转置卷积的输出作为更高一层“金字塔”的输入，该输入分别通过两个并联卷积层，一个继续抽取特征，一个完成残差学习，最终相加重构出目标分辨率的高分辨率图像。在每一个卷积操作后接着的激活函数这里选取了 Leaky rectified linear units(Leaky ReLU,LReLU)。此外 Lai 等<sup>[6]</sup>指出原先通常使用的 MSE 误差度量会导致重构结果过于光滑，因此它设计

新的损失函数缓解该问题。相比较直接插值再做卷积的网络搭建思路，由于其逐渐放大的特性，因此速度更快。

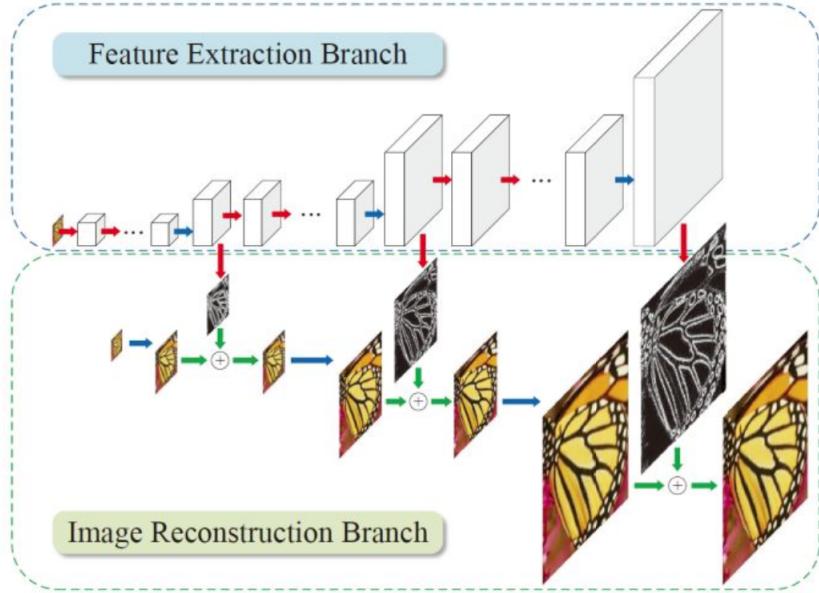


图 2.5 LapSRN 结构。图来自<sup>[6]</sup>。

Ledig 等<sup>[7]</sup>提出了 SRGAN 来完成单幅图像的超分辨率任务，这是第一次将生成对抗网络（Generative Adversarial Network, GAN）应用在超分辨率任务上。对于存在的 MSE 作为损失函数进行训练，恢复图像会过于光滑，缺乏高频细节的问题，SRGAN 应用了 GAN 中生成对抗的搭建网络和训练方式。网络结构包括生成器和鉴别器两个部分：生成器结构和传统的超分辨率中的网络结构类似，会完成包括插值（或者类似的扩充像素的方法），提取特征，特征的非线性变化，高分辨率图像重构等部分，鉴别器则将高分辨率参照物和高分辨率生成图像作为输入，判断其是真是假，其结构如图2.6所示。生成器则通过不断训练来欺骗鉴别器，鉴别器则通过不断训练来提高鉴别能力。因此可以把生成器当做是一个利用神经网络作为回归器的例子，而将鉴别器当做是一个利用神经网络作为 01 二分类器的例子。SRGAN 也使用了残差网络作为生成器的主体，相比较传统的 MSE 损失误差的神经网络，该方法能够获得更好的视觉效果。

### 2.3 图形学中的超分辨率任务的解决方案

历史上超分辨率任务多集中于图像和视频数据，并且多在图像处理和机器视觉领域被深入研究，而如今随着技术发展，人们认识到传统图形学的数据处理中也存在类似问题待解决。因此，最近两年存在着一些这样的工作，他们或是借鉴传统分析式的正向建模的方

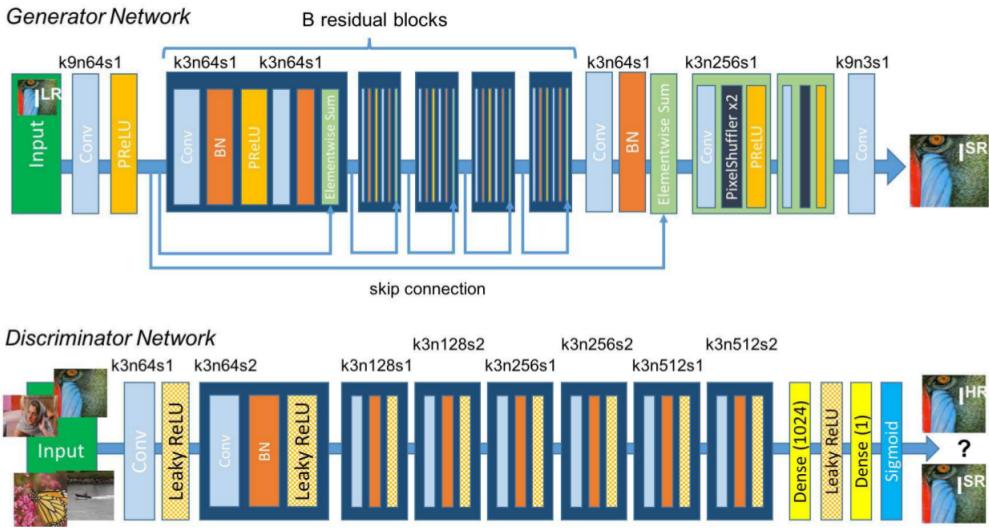


图 2.6 SRGAN 结构。图来自<sup>[7]</sup>。

法, 如<sup>[35]</sup>, 或者使用数据驱动下的深度学习的框架, 如<sup>[21,36]</sup>, 对图形学中常见的数据如何进行超分辨率展开研究。

Hamdi 等<sup>[35]</sup> 提出使用局部相似性进行点云的超分辨率。通常普通的 3D 扫描仪可以快速扫描获得物体的几何信息, 但是分辨率较低, 如何在不利用专业的多角度扫描仪的情况下, 如何提高获取点云细节质量十分关键。Hamdi 等<sup>[35]</sup> 基于自相似假设: 假定该点云的表面几何文理是重复的, 另外在低分辨率下的相似细节在高分辨率下依然保持。由此该文分析低分辨率下的相似细节并通过插值来增强它, 以实现超分辨率。但是, 如果扫描点云本身不符合该假设, 那么增强的结果就会和真实情况出入很大。

Argudo 等<sup>[36]</sup> 等提出使用航空图像和全卷积神经网络进行地形数据的超分辨率。通常的地形海拔高度模型只能获得较低分辨率版本, 只有对于个别地方具有精确的地形海拔模型, 因此如何丰富这些低分辨率海拔模型的地貌细节成为了一个待解决的问题。另一方面, 这些地方通过卫星获得的正射投影是容易的, 因此本文设计了一个监督学习下的全卷积网络通过训练 (低分辨率地形模型加其高分辨率正射投影) 到高分辨率地形模型的映射完成超分辨率任务。

Xie 等<sup>[21]</sup> 提出使用对抗生成网络的结构完成 Navier-Stokes 方程模拟下的烟雾数据的超分辨率任务。它是第一个使用生成对抗网络完成 4D 数据 (3D 空间 + 时间) 的超分辨率工作。它也是本文工作的基础, 在它的基础之上, 本文提出了其一些不足之处加以改进并应用到浅水方程数据的超分辨率任务上。它使用生成对抗网络进行烟雾序列的超分辨率, 通过设计一个空间鉴别器判断推测数据空间上和真实值的相似程度, 以及一个新的时间鉴

别器判断推测序列的时间连续程度，并且通过数据扩充的方法减小了训练集的大小。

## 2.4 本章小结

本章介绍了对图像或者视频进行超分辨率任务的研究现状，对其方法进行了综述，主要可以频率域方法和空间域方法，空间域方法中又可以分为基于插值，基于重建，基于学习三类方法，并对如今快速发展的基于数据驱动的学习算法进行了较为详细的介绍，包括最近五年成功使用神经网络进行超分辨率的典型工作，最后介绍了图形学领域最近对其他非图像数据类型（点云，海拔高度模型，烟雾密度场序列）进行超分辨率相关的研究工作。下一章，将围绕本文主题：通过 GAN 对浅水方程数据进行超分辨率，介绍一些紧密相关的必要知识作为具体的浅水方程数据超分辨率算法构建的基础，为第四章介绍本文算法进行铺垫。因此在第 4 章叙述具体浅水方程数据的超分辨率算法时将假设读者对于这些第 3 章的必要知识已经足够了解。

## 第3章 卷积神经网络、生成对抗网络和浅水方程

本章将着眼于本文的研究主题：如何通过 GAN 对浅水方程模拟的数据进行超分辨率，介绍一些紧密相关的必要知识，包括深度卷积神经网络（CNN），生成对抗网络（GAN）相关的理论背景、变种和应用，以及浅水方程（SWE）的数据的获取方法。

### 3.1 卷积神经网络

人工神经网络的前身可以追溯到计算机尚未发明之前。早在 1890 年，在心理学和神经科学领域就提出了神经细胞的激励相互叠加的理论，这可以认为是现代人工神经网络的启蒙，它为构造人工神经网络处理外部数据提供了思路。之后数学家和神经生物学家提出了神经网络的数学概念，和神经细胞之间的连接模型。最初的神经网络被称为“感知器”，在当时它只能被用来处理简单的线性问题，并因为著名计算机科学家 Minsky 的消极观点而发展停止。而 1974 年，反向传播算法诞生并在之后由于众多电子科学家、物理学家、生物学家的促进，在此基础上神经网络得以复兴。虽然随后十多年机器学习领域一直被传统的基于统计和优化的学习算法占据，但随着深度信念网络（DBN）和卷积神经网络（CNN）在大型数据集上的结果远远超过支持向量机等传统学习算法，由此在二十一世纪至今被大力的发展，在计算机视觉，自然语言处理，机器人，语音识别等领域大放异彩。其中卷积神经网络由于其在完成视觉任务方面的出色表现，成为了现在深度学习的主流模型之一。

#### 3.1.1 历史发展

卷积神经网络（Convolutional Neural Network,CNN）是因其特殊的自动学习数据特征的卷积层结构而得名，其具有局部感知，权值共享，自动提取特征等优点，相比较全连接神经网络在视觉任务中有巨大优势。其起源可以追溯到生物学家对于人体视觉生理的研究，比如神经元的级连构造，并且更重要的是哺乳动物单个视神经中的神经元仅仅会它所负责的感受区域做出响应，然后传递给其他神经元，而不是对于整个图像区域做出响应，如图3.1所示。这项发现的科学家 Hubel 和 Wiesel 因此获得了 1981 年的诺贝尔生物学奖，这

也为卷积神经网络的仿真提供了思路，可以说卷积神经网络从数学上建模了生物的视神经通过级连传播信号传播和局部感受信号的性质。

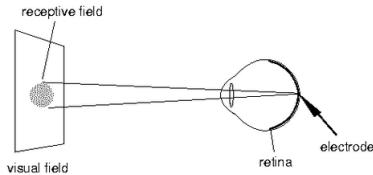


图 3.1 哺乳动物视觉局部感受外部环境的特点。

1988 年，日本研究员福岛邦彦提出了“神经认知”的概念，将 Hubel 和 Wiesel 的关于生物视觉原理的工作进行了数学建模，其原理图如图3.2所示。它是现代卷积神经网络的雏形，第一次定义了如今的级连式的卷积形式，第一次提出了 ReLU 的激活函数为整个过程提供非线性能力，第一次提出了平均池化操作来进行信号的整合，第一次提出了局部感受的信号处理操作方式，基本上奠定了卷积神经网络的基础，但是由于当时计算机计算力所限和非监督式的学习方法的制约，其实用性较差。

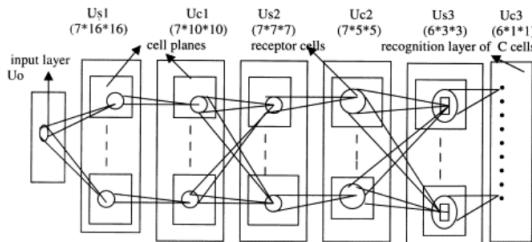


图 3.2 神经认知器结构

直到 1990 年前后，LeCun 在其博士论文中提出了利用梯度反向传播训练福岛邦彦的网络模型的方法，这奠定了卷积神经网络实用性的基础，是现在卷积神经网络训练方法的基础，并且随后将这种监督式的训练方法训练的网络成功地应用在了手写数字识别的任务中，就是著名的 LeNet-5，其结构如图3.3所示。这基本上已经是现在卷积神经网络算法的大体形式了。接着，美国科学家翁巨杨在他的论文中提出了数据增强的提高泛化性避免过拟合的方法和最大池化的方法，这两个技巧也在现代卷积神经网络设计和训练中被广泛使用。

但由于当时计算力欠缺和支持向量机算法的盛行，卷积神经网络在 21 世纪初并没有受到太多的关注。但是随着 GPU 的出现和快速迭代，卷积神经网络焕发生机，以“深度学习”的名称重出江湖，标示性的工作是 2012 年 AlexNet<sup>[37]</sup> 在图片分类任务比赛上登顶，并且高出了第二名十几个百分点，体现了其巨大的实用性，由此推动了深度学习在视觉领

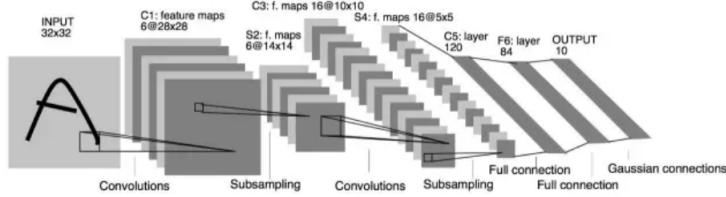


图 3.3 LeNet-5 结构。图来自<sup>[9]</sup>。

域的研究和应用落地，随后的基于卷积神经网络作为网络主体的研究工作络绎不绝。2012年之后，著名的卷积神经网络架构有：VGG<sup>[38]</sup>、GoogleNet<sup>[39]</sup>、ResNet<sup>[10]</sup>、ENet<sup>[40]</sup>等。

### 3.1.2 理论背景

卷积神经网络结构中最重要的结构就是局部感受的卷积层，它是由一组可以学习的卷积核和一组偏移量构成。卷积数学上是一种积分相关的操作，连续情况定义为：

$$f * g(n) = \int_{-\infty}^{\infty} f(r)g(n-r)dr \quad (3.1)$$

离散情况下定义为：

$$(f * g)(n) = \sum_{r=-\infty}^{\infty} f(r)g(n-r) \quad (3.2)$$

其中  $n$  表示卷积操作的定义域， $f$  表示原始信号， $g$  表示特定卷积函数。在图像处理领域，传统的卷积操作可以表示为有一组预先定义好的卷积核矩阵  $g$ （也被成为滤波器）在图像空间  $f$ （即原始信号）上滑动，所到之处利用卷积公式3.3进行处理，根据事先定义的卷积核的功能不同，可以实现包括光滑（去噪），边缘检测，锐化等操作，因此狭义的图像视觉领域的卷积操作可以认为是一组数值固定，全局共享的矩阵，通过和原始图像矩阵加权相加，可以提取出用户需要的某些特征（如边缘）。但是因为手动构造卷积核形式以满足各种各样的实际需求非常困难，而卷积神经网络通过数据驱动的方式让网络自动学习适合的卷积核可以极大程度避免手动构造卷积核的困难。

$$(f * g)(a, b) = \sum_h \sum_k f(h, k)g(a-h, b-k) \quad (3.3)$$

其中  $h$  遍历卷积核当前覆盖的图像的长度索引， $k$  遍历卷积核当前覆盖的图像的宽度索引。

### 3.1.3 常见网络功能层

卷积网络中常见的功能层包括卷积层，激活函数层，池化层，全连接层，损失函数层等。

**卷积层。**如上所述，卷积操作根据定义的卷积核的值不同，可以完成不同的特征提取功能，并且每一个卷积核是全局共享使用的，即从该卷积核获得的新映射在滑动过程中始终保持卷积核相同，只提取特定的某个特征。这符合生物视觉的局部感知原理，并且实际过程中局部性也减少了待训练变量的个数和中间变量的个数，缩短了前向传播函数值和反向传播函数误差的时间。一层卷积层则可以认为是若干个卷积核分别和原始数据进行卷积操作的输出的深度叠加形成的多维矩阵，每个卷积核学习到特定特征，输出被称为特征图（feature map）。假设共有  $j$  个卷积核，输入为  $x^{l-1}$ ,  $l-1$  表示输入位于  $l-1$  层,  $x_j^l$  则为卷积层输出， $k_j^l$  表示当前  $l$  层使用的一个卷积核变量，卷积核共有  $j$  个，则卷积层的数学操作如下，其中  $b$  表示可学习的偏移向量， $b_j$  表示其对应于卷积核  $k_j$  的偏移分量：

$$x_j^l = f(\sum x_i^{l-1} * k_{ij}^l + b_j^l) \quad (3.4)$$

其中  $f$  表示可能的激活函数，因为这里将卷积层和激活函数层分开讨论，因此这里的  $f$  为简单的恒等映射。

**池化层。**池化层事实上是实现一组固定形式的下采样操作，即将上一层输入进行降维处理，池化操作行为固定，因此不存在可学习变量。但是事实上池化层可以有一层卷积层代替，实现同样的下采样功能，并且实现特征提取。和其相关的操作是，反池化操作用来实现上采样，即将输入升维，而转置卷积操作同样可以实现上采样，但是有一组转置卷积构成，需要学习获得。池化层一定程度上可以提供给卷积神经网络一定的旋转等价性，而卷积操作本身只满足平移不变性，所以池化层可以提高卷积神经网络的泛化性和缓解过拟合。常见的池化操作有平均池化和最大池化。

平均池化 (Average Pooling)，其数学上如下所示：

$$x^l = \frac{1}{N * N} \sum_{n*n \in N * N} x_{n*n}^{l-1} \quad (3.5)$$

最大池化 (Max Pooling)，其数学上如下所示：

$$x^l = \max(x_{n*n}), n * n \in N * N \quad (3.6)$$

可以看出平均池化是在一个  $N * N$  的范围里做平均作为对应范围下采样的值，而最大池化是在一个  $N * N$  的范围里取最大值作为下采样的值。在基于 LeNet 的卷积神经网络架构中，多使用平均池化完成下采样功能，而现代卷积神经网络发展中发现最大池化对于保持特征更有利，因此目前实现中更经常被使用。

**全连接层。**全连接层相比较卷积的局部操作不同，该层的每一个顶点都和上一层的所有顶点相互连接，连接的边权为待学习变量。因此假设该层有  $n$  个节点，上一层有  $m$  个节点，每一条节点和节点连接的边都是一个待学习的变量，则通过上一层更新该层的公式如下，对应一次矩阵乘法操作和一次向量加法操作， $b$  是一个可学习的偏移变量：

$$x_n = W_{n*m} * y_m + b_n \quad (3.7)$$

通常在卷积神经网络中，全连接层获取最后要提取的特征向量，比如分类器会在倒数第二层使用一个  $n$  维向量表示分类结果，该向量会和上一层进行全连接操作，事实上这个分类结果也应当和上一层每个节点值相关，因此使用全连接合适，可以看出全连接层需要训练的变量数目会很大，因此卷积神经网络中多在网络末端使用全连接层进行最终特征的抽取。

**激活函数层。**多层神经网络通过若干功能层进行级连，这些网络层都是一些线性操作，包括卷积层进行卷积操作，全连接进行矩阵乘法，池化层进行都是线性操作，如果这些层之间直接级连，那么因为线性操作的可加性，它们的结果只能是原始输入的线性组合，因此势必要通过嵌入非线性函数提供给神经网络非线性拟合的能力。常见的激活函数有 sigmoid 函数，tanh 函数，ReLU 函数等。

sigmoid 函数。Sigmoid 函数数学形式如下：

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.8)$$

sigmoid 函数可以将结果规范到  $[0,1]$  之间，这种形式和神经生理学上的神经元受“抑制”或者“兴奋”对应，这很适合作为分类问题中倒数第二层的全连接层之后添加，可以将结果规范到  $[0,1]$  之间，表示结果属于哪一类的概率，但是另一方面可以看出来 sigmoid 函数在输入值过大时梯度值都非常小，因此多层使用 sigmoid 函数作为激活函数会导致“梯度饱和”（“梯度消失”）的问题，即误差反向传播到网络初始层时会非常小，导致无法有效地更新网络中的待学习变量，从而训练收敛非常慢甚至不收敛，并且其函数值不具有零值中心化的特点。

tanh 函数。tanh 函数数学形式如下：

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.9)$$

相比较 sigmoid，tanh 具有零值中心化的特点，但是其依然存在多层使用时“梯度饱和”的问题。

ReLU(Rectified Linear Unit) 函数。ReLU 函数数学形式如下：

$$Relu(x) = \max(0, x) \quad (3.10)$$

ReLU 函数通过当输入中出现负值进行截断处理提供非线性，相比较 sigmoid 函数和 tanh 函数，其求导更为方便，也克服了“梯度饱和”的问题，但是当输入多为负值时，它会导致这部分信息无法有效向前传播（因为负值直接被截断成 0），因此其也存在名为“dead ReLU”的副作用。对此，存在着一些 ReLU 的变种激活函数，包括 Leaky ReLU、PReLU、RReLU 等，改进 ReLU 激活函数的这一问题。

Leaky ReLU。Leaky ReLU 函数数学形式如下：

$$Leaky\_ReLU(x) = \begin{cases} x, & (x > 0) \\ ax, & (x \leq 0) \end{cases} \quad (3.11)$$

它不会直接输出零而忽略输入为负值时的情况，而是调整斜率为  $a$ ，这样就可以克服 ReLU 无法有效处理负值输入的情况。

**损失函数层。**网络最终需要定义一个损失值函数，该值函数是描述该网络的需要优化的目标，通过设计不同的损失函数表示不同的目标，通过目前基于梯度的优化方法（梯度下降法，随机梯度下降法，ADAM 方法等）最小化该目标函数。常见的损失函数会通过监督学习的方式，给定某些神经网络的输出结果（上一层网络）和真实参考数据之间差距的度量，通过最小化该度量，使得在这些观测样本空间中，神经网络输出的分布和真实参考数据分布越来越一致。常见的损失函数有均方误差 (mean square error, MSE)，交叉熵误差函数等。

均方误差。均方误差通常用于回归器，公式如下所示：

$$Loss = \frac{1}{2n} \sum_{x=0}^n (a_x - y_x)^2 \quad (3.12)$$

该式表示  $n$  个样本中，每个样本的输出  $a_x$  和真实参考值  $y_x$  之间的欧氏距离取平均，传统的超分辨率损失函数通常使用该误差度量。

交叉熵误差。交叉熵误差可以用于分类器和回归器设计，公式如下所示：

$$Loss = -\frac{1}{n} \sum_{x=0}^n (y_x \ln a_x + (1 - y_x) \ln (1 - a_x)) \quad (3.13)$$

该式表示当样本的估计值  $a_x$  和真实值  $y_x$  之间的差距如果二者相近，那么可以看出该式趋于 0，如果二者相差很大，极端情况下，则该式趋于正无穷大，因此最小化该式，也可以使得网络的输出分布尽可能和真实观测数据一致。更进一步，交叉熵误差的收敛性优于均方误差。

### 3.1.4 经典模型

LeNet-5<sup>[9]</sup>。LeNet 是现代卷积神经网络的雏形，包含了卷积神经网络的主体结构，其诞生在 1994 年，由纽约大学的 Yann Lecun 发明，其主体有一层卷积层，一层平均池化层，一层卷积层，一层平均池化层，一层全连接层级连形成，如图3.3，它被成功应用于手写数字识别的任务之中，虽然受限于层数较少，也没有 GPU 加速，但是其结构中所使用的卷积层，平均池化层，sigmoid 激活函数，通过全连接层的独热编码 (one-hot encoding) 进行分类等开创性工作都被沿用至今。

AlexNet<sup>[37]</sup>。AlexNet 主体沿用了 LeNet-5 的结构，它于 2012 年在图像分类竞赛 ImageNet 中技压群雄夺冠，它相比较 LeNet 的 5 层隐藏层结构更深，使用了 ReLU 函数提供网络非线性拟合能力，使用了 Dropout 技巧训练网络，提高了网络的泛化性，使用了最大池化层代替平均池化层，提高了特征保持的同时实现下采样，通过 GPU 加速了训练。AleNet 通过其显著的实用性和可接受程度的训练时间使得卷积神经网络重新焕发生机。

VGG<sup>[38]</sup>。VGG 由牛津大学在 2014 年提出，其相比较 AlexNet 更深，采用了  $3 \times 3$  大小的卷积核进行卷积操作，网络层数达到了 19 层，这在当时是非常深的神经网络架构了。它在 ImageNet 的图像分类获得了第二，图像定位中获得了第一。其加深网络提高泛化性的思路也被后来的网络所采用。

ResNet<sup>[10]</sup>。ResNet (Residual Network, 残差网络) 由 Kaiming He 等<sup>[10]</sup>于 2015 年提出，其在 ImageNet 中获得了检测，分类，定位三个任务的冠军，其深度达到了惊人的 152 层，它通过残差学习的方式克服了神经网络随着深度加深而难以学习的问题，通过卷积层学习残差和直接的原始信号相加的办法形成所谓的残差块，残差块之间串联可以进行非常深的网络的搭建和训练，如图3.4，该方法有效地缓解了梯度爆炸、梯度消失等问题，在如今的卷积神经网络设计中被广泛使用。

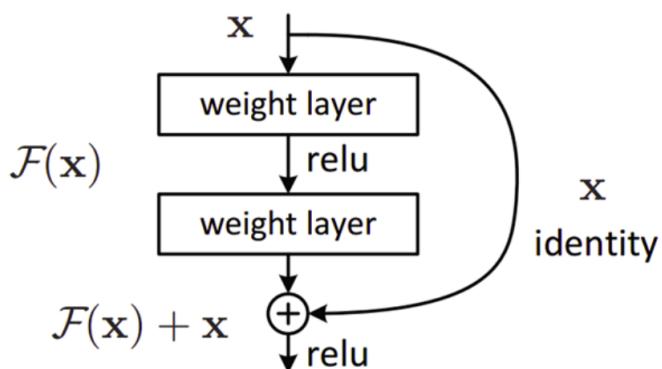


图 3.4 ResNet 中的残差块结构。图来自<sup>[10]</sup>。

## 3.2 生成对抗网络

### 3.2.1 产生背景

生成对抗网络 (Generative Adversarial Network, GAN) 由 Goodfellow 等<sup>[11]</sup>于 2014 年提出，原始的 GAN 试图学习一个已知数学分布（高斯分布或者均匀分布）的随机变量到真实数据分布的映射关系，并且抛弃了传统定义固定损失函数（MSE 等）进行神经网络训练的方式，引入了生成器和鉴别器两个概念，通过对抗学习的方式使得生成器的输出的概率分布尽可能和真实数据分布接近。最早 GAN 被应用于图片生成任务中，其提出之后出现了一系列工作，或是提高其训练速度、解决其训练困难、模型崩溃等问题，或是修改结构以适用于更多的应用场景。



图 3.5 使用 GAN 生成手写数字。最右列为真实数字，其余为生成器生成数字，可以看到真实样本和生成样本并不完全一致，因此 GAN 具有泛化性，而非简单的编解码的记忆。图来自<sup>[11]</sup>。

### 3.2.2 理论基础

生成对抗网络和传统的神经网络不同，它是有至少两个网络构成，分别充当生成器 (generator, G) 和鉴别器 (discriminator, D)，生成器完成回归任务；生成器的输出和真实参考值  $y$  会被当做输入传给鉴别器，鉴别器进行二分类任务，判断当前输出是真实参考值的概率。原始的生成对抗网络的生成器接受一个已知数学分布（高斯分布等）的随机变量  $z$ ，它通过生成器内部的待学习的非线性映射，输出一个“伪造的真实数据”，如手写数字生成任务，如图3.5，而生成器的输出和真实数据都会被鉴别器判断真假，其示意图如下3.6。因此生成器期望尽可能欺骗鉴别器，而鉴别器期望尽可能判断正确，二者进行零和博弈，所以生成对抗网络的目标可以归结为如下的极大极小问题来求解：

$$\min_G \max_D V(D, G) = E_{y \sim p_{data}(y)}[\log D(y)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (3.14)$$

其中  $y \sim p_{data}(y)$  表示真实数据样本  $y$  服从概率分布  $p_{data}$ , 同理  $z \sim p_z(z)$  表示噪音数据样本  $z$  服从概率分布  $p_z$ 。

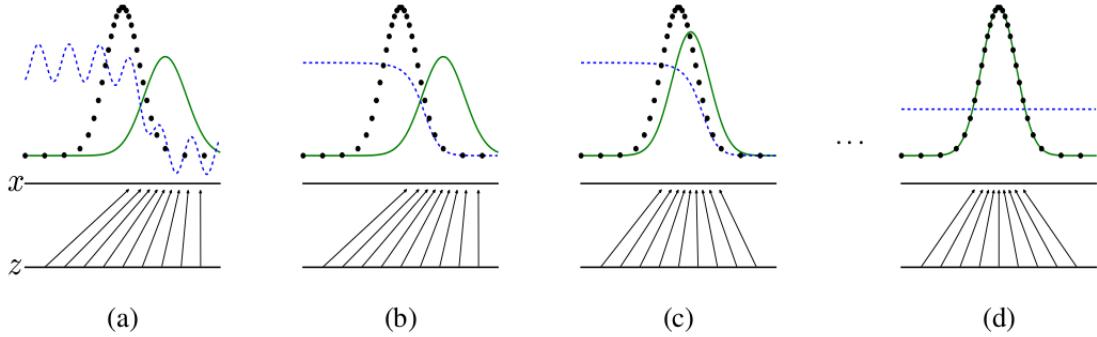


图 3.6 随着训练, 原始的 GAN 的生成器的输出的变化, GAN 试图学习一个和真实样本分布一致的生成器输出样本分布。图来自<sup>[11]</sup>。

而在实际过程中, 借鉴 ADMM 算法的思路, 会将此过程分开进行, 控制一个网络不变情况下, 优化另外一个网络的可学习变量, 通过分别最小化生长器的交叉熵损失函数3.15和最小化鉴别器的交叉熵损失函数3.16进行生成器和鉴别器的训练。

$$\min_G V(D, G) = E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (3.15)$$

$$\min_D V(D, G) = -E_{y \sim p_{data}(y)}[\log D(y)] - E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (3.16)$$

因此可以看到生成对抗网络有两个网络需要训练, 一般而言二者交替训练: 训练若干次鉴别器, 提高其判断能力, 再训练若干次生成器, 提高其欺骗鉴别器能力,<sup>[11]</sup> 一文中的训练方法如下图3.7所示。在<sup>[11]</sup> 中 Goodfellow 证明了 GAN 存在唯一全局最优解, 这一性质也是大多数非线性优化无法满足的: 当且仅当鉴别器最终判断无法判断一个输入是真是假时, 即其真实的概率为 0.5, 达到所谓的”纳什均衡“状态, 类似高维空间中的鞍点处, 如图3.8, 该算法达到全局最优解, GAN 训练收敛。另一方面, 虽然理论上存在这样一个收敛状态, 但是实际过程中, 生成器和鉴别器被分开训练, 存在着很大程度的训练不稳定的情况, 如倘若鉴别器在早期被训练的过强, 导致无论如何生成器的输出都被视为赝品, 那么会使得过程难以收敛, 因此随后也出现了一系列生成对抗网络的变种, 试图提高其训练稳定性。

生成器和鉴别器本身可以由传统的神经网络来搭建, 如图像任务经常使用卷积神经网络来搭建生成器, 如 DCGAN<sup>[41]</sup> 就将卷积神经网络作为子网络用于 GAN 的搭建中。

**Algorithm 1** Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator,  $k$ , is a hyperparameter. We used  $k = 1$ , the least expensive option, in our experiments.

---

**for** number of training iterations **do**

**for**  $k$  steps **do**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .

- Sample minibatch of  $m$  examples  $\{x^{(1)}, \dots, x^{(m)}\}$  from data generating distribution  $p_{\text{data}}(x)$ .

- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[ \log D(x^{(i)}) + \log (1 - D(G(z^{(i)}))) \right].$$

**end for**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .

- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(z^{(i)}))).$$

**end for**

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

---

图 3.7 原始的 GAN 训练过程。图来自<sup>[11]</sup>。

相比较传统的定义特定损失函数来进行训练的方法，GAN 并不需要显式地提供损失函数的设计，因此避免了手动设计损失函数不满足需求的障碍，在风格迁移，超分辨率，图像或者三维模型补全，图像去噪等方面都相比较传统的损失函数定义下的训练出来的神经网络具有更好的质量。

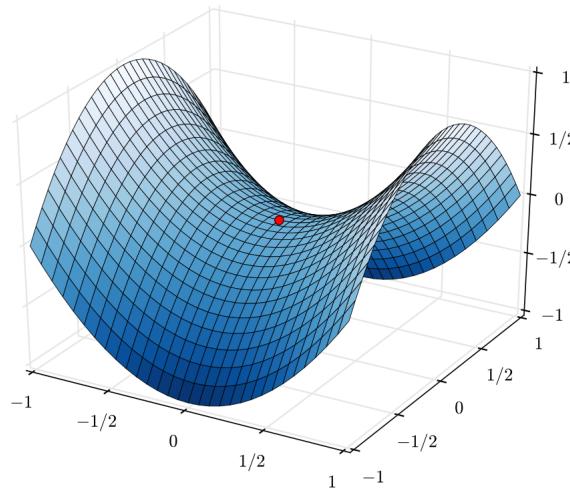


图 3.8 极大极小问题求解对应寻找一个高维空间中的鞍点。

### 3.2.3 存在的问题

虽然 GAN 相比较传统的神经网络具有更大的灵活性和优势，能够胜任多种多样的任务，并且具有很好的拓展性，但是它自身也存在着一些理论问题需要去解决。

训练不稳定，容易发散不收敛。虽然理论上存在着理想的纳什均衡状态，但是目前的训练方法依然依赖于生成器和鉴别器双方分别根据估计的梯度进行优化，且内部均为非凸的神经网络搭建，因此很难达到该平衡状态。

模型崩溃。在生成器生成样本过程中，可能存在无论输入的噪声如何变化，生成器始终生成相同的样本，不具有结果的多样性，这被称为生成对抗网络的“模型崩溃”问题。

### 3.2.4 GAN 变种

自 GAN 模型提出之后，存在着一系列工作，试图或是提高其训练速度、解决其训练困难、模型崩溃等问题，或是修改结构以适用于更多的应用场景。以下本文将列举其中的代表性工作。

条件 GAN。原始的 GAN<sup>[11]</sup> 将一组固定分布的噪声  $z$  映射到真实数据分布，但是对于真实数据分布内部的标签信息没有有效的利用，比如在手写数字生成的任务之中，<sup>[11]</sup> 中的 GAN 模型对于输入噪声通过生成器可以获得一个手写数字，但是该数字的标签却不能事先可知，完全依赖于生成器内部的分布行为和输入的噪声，因此很难控制，无法完成类似希望输出某个数字（标签），那么就能输出某个数字的任务。因此<sup>[42]</sup> 提出了在原始 GAN 的基础上给生成器的输入加上约束进行训练，该变种被称为“条件生成对抗网络”（Conditional Generative Adversarial Network，Conditional GAN）。相比较原始的 GAN，条件 GAN 在生成器和鉴别器中都引入了条件变量  $x$  作为约束，从而可以完成类似指定类别的图像的生成之类任务，其结构示意图如下图3.9所示。条件生成对抗网络从数学上就是将上述的极大极小值问题变为条件概率下的极大极小值问题来求解：

$$\min_G \max_D V(D, G) = E_{y \sim p_{data}(y)}[\log D(y|x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z|x)|x))] \quad (3.17)$$

DCGAN<sup>[41]</sup>。DCGAN 则将卷积神经网络作为 GAN 的子网络来搭建，用它来完成图像生成任务同样可以通过条件 GAN 的方式引入标签，完成特定标签下的图像的生成。另一方面它抛弃了通过插值进行图片维度提高的方法，而是通过转置卷积来实现更高分辨率的图像生层任务。其使用了批归一化（batch normalization）的方式提高网络的泛化性和缩短时间，也避免了需要人工对数据进行手工归一化的不便。

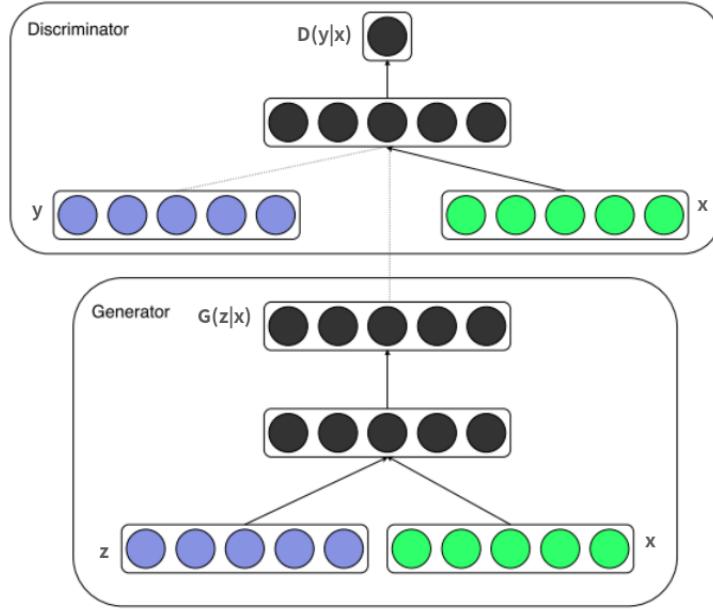


图 3.9 条件 GAN 示意图。

除了增加条件约束提供 GAN 可控性方面的工作以外，对于 GAN 训练不稳定的问题，有以下这些工作进行了改进。

W-GAN<sup>[43]</sup>。无论是 DCGAN, conditional GAN 还是最原始的 GAN，它们都基于交叉熵函数对网络进行训练，但是 W-GAN<sup>[43]</sup> 指出这种损失函数会带来训练不稳定的问题，而且生成器和鉴别器之间的交叉训练很难做到平衡，即无法显式地知晓训练生成器多少次和训练鉴别器多少次分别合适。W-GAN 从该角度进行了改进：将判别器全连接层之后紧跟的 sigmoid 激活函数层去掉，不使用交叉熵函数，而是改为根据3.18进行训练，并且将更新后的网络中的变量截断到一定范围，以满足论文中的数学要求。其从理论上分析了 GAN 训练不稳定的问题所在，并且使用了 Wasserstein 距离去估计生成分布和真实分布之间的距离度量，代替了交叉熵函数使用的 JS 散度度量。并且，W-GAN 从一定程度对模型崩溃问题也有所缓解。

$$\min_G V(D, G) = -E_{z \sim p_z(z)}[D(G(z))] \quad (3.18)$$

$$\min_D V(D, G) = -E_{y \sim p_{data}(y)}[D(y)] + E_{z \sim p_z(z)}[D(G(z))]$$

LS-GAN<sup>[44]</sup>。和 W-GAN 出发点一样，同样是通过修改损失函数缓解生成对抗网络训练不稳定的情况，LS-GAN (least square GAN, 最小二乘 GAN) 则是通过最小二乘损失函

数3.19代替了原始的交叉熵函数进行训练。

$$\begin{aligned} \min_G V(D, G) &= E_{z \sim p_z(z)}[0.5(D(G(z)) - 1)^2] \\ \min_D V(D, G) &= E_{y \sim p_{data}(y)}[0.5(D(y) - 1)^2] + E_{z \sim p_z(z)}[0.5(D(G(z)))^2] \end{aligned} \quad (3.19)$$

### 3.2.5 具体应用场景

上述的 GAN 变种模型从基础的数学原理层面对生成对抗网络进行了修改，缓解其训练不稳定性、缺乏控制性和模型崩溃的问题。另一方面，GAN 具有巨大的灵活性，不需要去手动设计损失函数，鉴别器本身就充当了一个自学习的损失函数的角色，因此有大量工作将 GAN 拓展到各种应用中去，如下本章根据应用的不同做简略介绍。

模型生成<sup>[18]</sup>。原始的 GAN<sup>[41]</sup>就可以完成图像的自动生成，条件 GAN<sup>[42]</sup>更将其推广到可以给定标签生成符合该标签的图像。因此将 GAN 应用于三维模型生成是一个很自然的推广，但是从二维图像生成推广到三维模型却有一些不可绕过的困难，目前的思路有如下几种：直接将模型体素化，生成器生成对应的体素，然后将体素再转成网格；将一类网格参数化二维空间，然后 GAN 生成器生成多样的参数化以后的结果，因为参数化之后在二维空间，因此标准的卷积神经网络非常适合，再根据该类网格对应的模板将参数化之后的二维网格重构到三维空间，实现模型生成<sup>[18]</sup>；直接在模型网格表面重新定义卷积操作，实现模型生成，但因如何从二维的图像欧式空间推广卷积操作到黎曼空间尚未取得突破进展，因此该方案目前并没有非常好的基于卷积神经网络和生成对抗网络的结果。

风格迁移<sup>[45]</sup>。风格迁移任务是试图将某一种风格的图片转化成另外一种，并保持内容不变，最早的风格迁移学习通过均方误差定义风格和内容度量，希望内容尽可能保持为原始图像内容，风格尽可能转变为目标图像风格<sup>[46]</sup>。随着条件 GAN 提出，可以看出将条件 GAN 的目标函数3.17中的  $y$  视作目标风格下原内容的目标数据而  $z, x$  均视作原始风格下原内容的数据，那么通过这些配对模型训练 GAN，就可以完成风格迁移的任务，但是麻烦之处在于需要这些配对好的数据，但是<sup>[45]</sup>通过双向映射的方法，即训练  $z -> y$  的无条件 GAN 和  $y -> z$  的无条件 GAN，并且引入双向映射的误差度量，从而可以避免数据需要配对才能实现风格迁移的困难。

去噪<sup>[47]</sup>。图像去噪的目的是将图像中的噪声  $n$  去除，同包含噪声  $n$  和无噪声图像  $p$  的和之中抽取  $p$ ，显而易见可以套用条件 GAN 的框架，将带噪声的图像作为  $z, x$  而将无噪声图像作为  $y$ ，使用目标函数3.17进行训练，实现去噪任务。

超分辨率<sup>[7]</sup>。<sup>[7]</sup>最早将GAN应用到超分辨率任务上，和基于条件GAN进行风格迁移和去噪等任务类似，将条件GAN的目标函数3.17中的 $y$ 视为高分辨率目标数据，而将 $z, x$ 视作低分辨率数据，就可以通过条件GAN的框架进行图像超分辨率任务的学习。<sup>[7]</sup>的生成器使用残差块作为主体，鉴别器使用卷积神经网络作为主体，其结果相比原先的使用均方误差的超分辨具有更好的恢复高频细节的能力。

模型补全，去遮挡<sup>[12]</sup>。<sup>[12]</sup>也是基于条件GAN的框架实现了人脸数据去遮挡的任务，将带遮挡的人脸数据作为 $z, x$ ，不带遮挡的对应人脸数据作为 $y$ 进行训练。该文中的生成器由自编码器作为主体，输入带遮挡人脸，输出去遮挡之后的人脸；文中使用了两个鉴别器，一个是鉴别被遮挡部分的恢复情况，一个是鉴别整个图像的恢复情况。文中的结果展示了GAN非常好的实用性和灵活性，如图3.10。

可以看出，在风格迁移、去噪、超分辨率、模型补全这些任务中，均可以使用条件生成对抗网络的框架，并且都使得 $x == z$ ，因此在这些应用中需要求解的条件概率下的极大极小问题由3.17简化为如下：

$$\min_G \max_D V(D, G) = E_{y \sim p_y(y)}[\log D(y|x)] + E_{x \sim p_x(x)}[\log(1 - D(G(x)|x))] \quad (3.20)$$

同样和公式3.15和公式3.16相似，这些任务中也通过分别最小化生长器的简化后的交叉熵损失函数3.21和最小化鉴别器的简化后的交叉熵损失函数3.22进行生成器和鉴别器的训练。

$$\min_G V(D, G) = E_{x \sim p_x(x)}[\log(1 - D(G(x)|x))] \quad (3.21)$$

$$\min_D V(D, G) = -E_{y \sim p_{data}(y)}[\log D(y|x)] - E_{x \sim p_x(x)}[\log(1 - D(G(x)|x))] \quad (3.22)$$

### 3.3 浅水方程

在介绍完了卷积神经网络和生成对抗网络之后，本节将介绍浅水方程模拟的数据生成方式，这里和公开的图像数据集不同，由于其特殊性，需要通过数值求解一组偏微分方程获得高分辨率下的浅水方程模拟的解来构造，获取之后训练GAN完成超分辨率学习中需要的训练和测试数据集。



图 3.10 通过 GAN 进行人脸去遮挡。从左到右分别是无遮挡真实图片，带遮挡图片和 GAN 的生成器去遮挡结果。图来自<sup>[12]</sup>。

### 3.3.1 物理背景

浅水方程 (shallow water equation, SWE) 是一组偏微分方程，其可由 Navier-Stokes 方程简化获得。而 Navier-Stokes 方程是一组描述三维流体宏观运动的偏微分方程。而基于水平速度在纵向上的梯度为零的假设，浅水方程将其从三维简化至二维高度场来描述。因此其可以表示一个 Navier-Stokes 子集内的流体运动。将设流体高度域为  $h$ ，沿着  $z$  轴，其运动平面  $xy$  区域水平， $u$  表示流体在该水平面上的速度向量， $u_x, u_y$  分别是沿着  $x$  轴速度分量和沿着  $y$  轴速度分量。则浅水方程是如下这样一组偏微分方程：

$$\begin{aligned} \frac{\partial h}{\partial t} + (\nabla h)u &= -h\nabla.u \\ \frac{\partial u}{\partial t} + (\nabla u)u &= a_n\nabla h \end{aligned} \quad (3.23)$$

其中  $a_n$  表示沿着  $z$  轴的加速度，例如重力加速度， $\nabla.$  表示散度算子，而  $\nabla$  表示梯度算子。

### 3.3.2 数值解法

图形学中物理模拟占据了很大一部分，其中包括弹性体、流体、刚体等，这些物理过程在宏观物理中可以通过一些偏微分方程和一组给定的初边值条件进行数值求解，而数值方法可以包括有限元方法 (FEM)、有限差分方法 (FDM)、有限体积方法 (FVM)、光滑粒子流体动力学 (SPH)、材料点 (MPM) 等。其中流体模拟是一个被广泛研究的问题，基于 Naiver-Stokes 的流体模拟可以模拟更为复杂的流体运动，而基于浅水方程的流体模拟更为高效，更适合实时模拟。这里的浅水方程的数值求解需要可以计算高度场随着时间变化的

过程。将方程3.23展开可以写成：

$$\begin{aligned}\frac{\partial h}{\partial t} + (\nabla h)u &= -h\nabla \cdot u \\ \frac{\partial u_x}{\partial t} + (\nabla u_x)u &= a_n \nabla h \\ \frac{\partial u_y}{\partial t} + (\nabla u_y)u &= a_n \nabla h\end{aligned}\quad (3.24)$$

这组方程的等号左边部分可以认为是分别  $h, u_x, u_y$  分别和  $u$  发生平流 (advection)，而等号右边部分表示额外的加速度项。因此图形学经常使用的数值求解策略是通过时间显式积分的方式并将左右两部分分成两步求解，简化该问题的数值求解的复杂度。

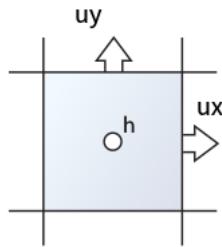


图 3.11 错列网格上需要记录的变量值。

数值求解该微分方程，需要首先将定义域从连续区域离散化，这里使用有限差分的方式，因此需要将流体的二维区域离散化成沿着  $x$  轴有  $n_x$  格子，沿着  $y$  轴有  $n_y$  格子的  $n_x * n_y$  区域。为了方便，这里假设这些格子是方格，边长为  $\Delta x$ 。加速度系数  $a_n$  沿着  $z$  轴，假设为重力加速度。此时完成了将空间离散化，还需要对时间维度离散化，方能数值解出若干时间刻度上的空间物理量的值，这里假设时间维度离散化后的每两个时间刻度之间的间隔为  $\Delta t$ 。数据中需要保存三个未知场  $h, u_x, u_y$  在格子内部并更新，这里的数值模拟采用常用的错列网格，记录格子中心的  $h$  值，而记录格子边上的对应速度分量，其示意图如图3.11所示。该种错列网格的表示方法对于流体十分适用，可以提高数值求解的稳定性。而这三个未知场在给定第一帧的初值条件之后，就可以通过时间积分的方法进行更新，其更新步骤为：

1. 先以当前的速度场根据3.24分别和  $h, u_x, u_y$  进行平流运动；
2. 根据 1 更新完成之后的  $h, u_x, u_y$  计算3.24的等号右边的加速度项，更新高度  $h$ ；
3. 根据 2 更新完成之后的  $h, u_x, u_y$  计算3.24的等号右边的加速度项，更新高度  $u_x, u_y$ 。

步骤 1 中，本文使用半拉格朗日方法<sup>[48]</sup> 计算，该算法进行平流计算通过对每个格子中

的一个虚拟的粒子进行时间向后查找。假设当前和速度平流的量为  $s$ , 首先需要更新在二维坐标  $x$  处的  $s$  的值, 通过假设此时在  $x$  的虚拟粒子是在上一帧通过当前速度  $u(x)$  获得移动获得, 那么可以知道, 上一帧时该粒子位置为  $x^{t-1} = x - \Delta t u(x)$ 。此时再通过在  $x^{t-1}$  处的  $s$  值更新  $x$  处的  $s$  值, 即  $s_{new}(x) = s(x^{t-1})$ 。可以看到虽然  $x$  只能取 3.11 上的边中点和格子中心, 因此该数据始终可以有记录值, 但是  $x^{t-1}$  可以是该二维空间中的任意位置, 因此需要内插该点的  $s$  值, 通常可以通过双线性插值或者三次样条插值来获得。

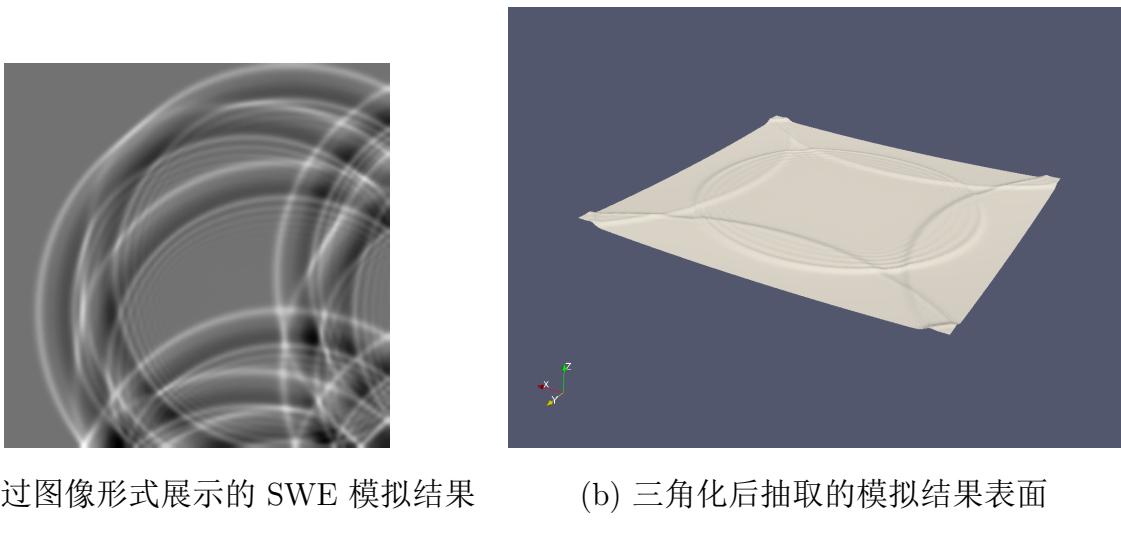


图 3.12 基于浅水方程的单相流体的模拟结果。

步骤 2, 3 中根据加速度更新  $h, u_x, u_y$  可以非常容易地通过差分代替微分的方式进行更新。但是需要注意所有的步骤 1,2,3 需要在区域的边界进行特殊处理, 因为边界处无法进行导数计算, 因此需要在每一帧更新时使用不同于上述的方法处理边界, 通常的浅水方程的边界条件可以设置为反射边界条件, 即当波传至边缘时它会被镜面反射回去。反射边界条件可以通过强制垂直边界的速度分量为 0, 并且边界高度值镜像复制其内部一层的高度, 这样模拟依然允许流体做切向运动而不穿过边界。通过以上的算法就可以完成单相浅水方程的流体模拟, 并且经过后处理, 可以将高度场  $h$  三角化抽取其流体表面网格, 其结果如图 3.12 所示。因为本文进行超分辨率的数据仅为单相浅水方程流体模拟数据的超分辨率, 即只有一种流体本身的模拟数据, 而不包含其他如刚体、弹性体的其他物理介质, 因此不详细讨论如何实现更为复杂的流体模拟, 例如其如何和刚体进行双向耦合的模拟数据的生成细节, 具体的算法可以参考<sup>[49,50]</sup>, 大致思路就是将流体给刚体的浮力计算得到并传给刚体更新刚体位置速度信息, 再将刚体对流体的作用力通过排开高度场的体积进行估计进行高度场的后处理更新, 其结果如图 3.13 所示。

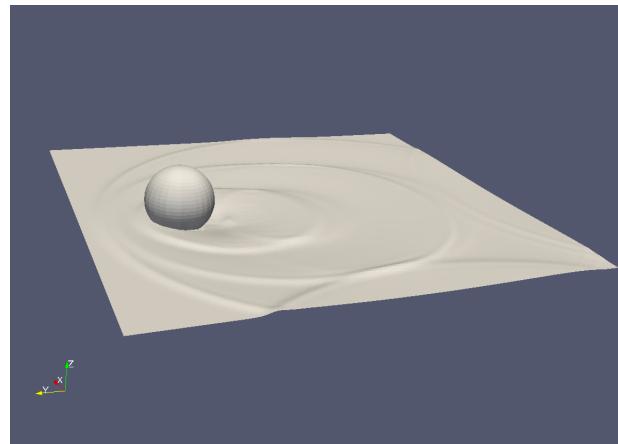


图 3.13 基于浅水方程模拟的流体和固体耦合的模拟结果。

### 3.4 本章小结

本章将着眼于本文的研究主题：如何通过 GAN 对浅水方程模拟的数据进行超分辨率，介绍一些紧密相关的必要知识，包括深度卷积神经网络 (CNN) 的历史发展、理论背景、常见的网络功能层和著名的基于卷积神经网络的模型，生成对抗网络 (GAN) 相关的理论背景、变种和应用，以及浅水方程 (SWE) 的数据的物理背景和数值求解方法。这为下一章介绍具体的浅水方程数据的超分辨率算法过程提供了铺垫。

## 第4章 算法设计和分析

本章将介绍通过生成对抗网络（GAN）进行浅水方程（SWE）模拟数据超分辨率的算法的设计细节，通过对浅水方程数据超分辨率任务过程中独特性质的思考，考虑到了浅水方程模拟超分辨率重建在时间维度上的光滑性需求，超分辨率任务的旋转等价性需求，模拟数据可能为负值的性质，物理过程中速度信息对神经网络训练有影响等关键因素，进行了神经网络结构的设计，损失函数的设计，数据的生成和扩充方法的设计和网络的训练方法的设计。本章中提出的算法设计细节的必要性和可行性将会在下一章中的实验中加以验证。最后是本章小结。

### 4.1 目标问题的数学描述

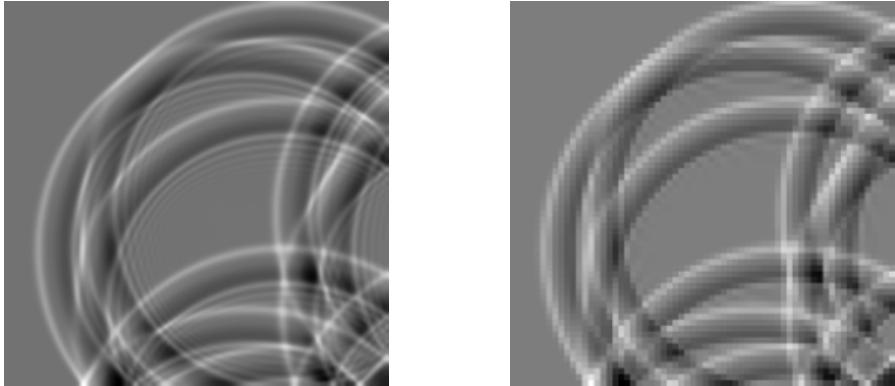
根据第三章的介绍，条件 GAN 的结构适合进行超分辨率任务的数学建模，因此本文这里将  $y$  认为是浅水方程模拟的高分辨率数据，而  $x$  则是降采样后的低分辨率对应，通过数据驱动的方式，试图让生成器学习到低分辨率到高分辨率的映射关系，可以对降采样后的低分辨率数据进行有效地超分辨率重建获得其高分辨率对应真实值的较好近似。典型的  $y$  和  $x$  如下图4.1所示。为了展示上的公平性，避免因为高分辨率数据和低分辨数据因为分辨率不一致导致的认为低分辨率非常模糊的感受，本文在展示低分辨率数据时，均将其通过 4 倍的最近邻插值到高分辨率维度。

在上一章中可以看到，针对超分辨率任务可以将原始的条件 GAN 的目标极大极小问题由公式3.17简化为如下：

$$\min_G \max_D V(D, G) = E_{y \sim p_y(y)}[\log D(y|x)] + E_{x \sim p_x(x)}[\log(1 - D(G(x)|x))] \quad (4.1)$$

其中  $y \sim p_y(y)$  表示高分辨率数据样本  $y$  服从概率分布  $p_y$ ，同理  $x \sim p_x(x)$  表示低分辨率数据样本  $x$  服从概率分布  $p_x$ ，而在实际计算中将被简单的平均代替，因此这里简化符号  $E_{y \sim p_y(y)}$  为  $E_m$ ，而  $E_{x \sim p_x(x)}$  为  $E_n$ ，均表示简单平均获得的期望。

同样，在上一章中可以看到浅水方程是一组生成在一个二维区域的高度场来描述特定流体行为的偏微分方程。在浅水方程中，基于二维区域的水平速度在垂直方向上的梯度为零



(a) SWE 数据高分辨率的一帧 (b) 降采样 SWE 数据低分辨率的一帧

图 4.1 典型的浅水方程的高分辨率数据和其降采样后的低分辨率数据。

的假设，流体被定义为一个变化的高度场  $h(t) : \Omega \rightarrow R$ , 和对应的水平速度场  $u(t) : \Omega \rightarrow R^2$  来描述。因此特定边界情况  $\mathcal{B}$  下的浅水方程的模拟结果一个三通道（一个通道描述  $h$ , 两个通道描述  $u$ ）的序列  $Y$ 。这里用  $Y_*^k, * \in \{h, u\}$  表示第  $k$  帧的第  $*$  通道。对应的， $Y_0$  表示高度  $h$  通道，而  $Y_1$  表示  $u_x$  速度分量通道， $Y_2$  表示  $u_y$  速度分量通道。

给定高分辨率结果  $Y$ , 低分辨率下采样结果为  $X$ 。问题归结为通过 GAN 结构，使得生成器学习到一个  $\mathcal{F}$  通过  $X$  估计  $Y_0$ 。因为本文只关心高分辨率的高度场恢复情况，因此这里忽略  $Y_1$  和  $Y_2$ ，并且重新用  $Y$  直接表示高分辨率高度场  $Y_0$ ，而  $X$  则可以是只使用下采样的高度场通道  $X_0$  或者使用了下采样高度场通道和两个速度分量通道  $X_{0,1,2}$ 。总结为，本文希望通过神经网络学习到如下映射：

$$\mathcal{F}(X) \approx Y, \quad (4.2)$$

其中  $\approx$  需要通过恰当的损失函数设计和恰当的网络结构设计来实现。

与大多数生成对抗网络的任务只需要鉴别器鉴别生成器  $G$  的一个独立输出的质量不一样，这里因为需要学习的是从一个下采样的低分辨率序列  $X$  到一个对应高分辨率序列  $Y$  的映射，单纯的一个鉴别器鉴别生成器  $G$  的一个独立输出的质量欠缺对浅水方程序列时间维度光滑性的考量，因此需要引入新的机制来判断输出质量的时间维度光滑性这一指标。为此，本文需要引入一个新的时间鉴别器  $D_t$ ，使得在整个序列上的恢复质量尽可能好。如果缺少了该时间鉴别器  $D_t$ ，从后面的实验结果中可以看出在时间维度上远离训练集的数据上进行测试其结果质量明显不如使用了时间鉴别器  $D_t$  进行训练的结果质量。这里，本文将原先的鉴别器称作空间鉴别器  $D_s$ 。因此这里相当于首先定义了一个空间鉴别器  $D_s$ ，作为一个可学习的损失函数估计生成器的超分辨率的质量，而用一个时间鉴别器  $D_t$ ，作为一个可学习的损失函数估计生成器生成的序列时间维度上的连续性。

## 4.2 损失函数设计

这里， $x, y, G(x)$  分别表明低分辨率数据  $X$  的单独一帧、高分辨率高度场  $Y$  的单独一帧和 GAN 生成器  $G$  的“伪造的”高分辨率高度场。而  $\tilde{X}$  和  $\tilde{Y}$  和  $\tilde{G}(\tilde{X})$  分别表示为低分辨率数据  $X$  连续的三帧、高分辨率高度场  $Y$  的连续的三帧、生成器根据连续三帧低分辨率输入“伪造的”连续三帧高分辨率高度场。在上一章的 GAN 的基础之上，这里对生成器和鉴别器分别设计恰当的损失函数来进行网络训练。

### 4.2.1 生成器的损失函数

首先，依照上一章超分辨率任务下的生成器的交叉熵损失函数3.21，这里同样构造相似的能量项  $-E_n[\log(D_s(G(x)|x))]$  作为生成器训练的损失函数的一项。极小化该项能量的过程中，其实是在试图提高生成器伪造数据的能力，希望生成器  $G$  尽可能地欺骗空间鉴别器  $D_s$ 。除此之外，在<sup>[51]</sup> 表明额外的 L1 范数损失函数项可以添加到生成器的损失函数中，保证生成器的输出  $G(x)$  尽可能和  $y$  相似，因此考虑加入该项  $\lambda_{L_1}E_n||G(x) - y||_1$  以提高最终生成结果的可靠性，其中  $\lambda_{L_1}$  是该项的权重。

另外，在基于卷积神经网络框架的生成对抗网络中，约束真实值和生成器伪造值到各层特征图 (feature map) 的输出尽可能相似也可以提高最终生成的质量并且可以缓解模型坍塌的问题：即加强了约束的限制，不仅仅通过第二项  $\lambda_{L_1}E_n||G(x) - y||_1$  约束生成器的输出  $G(x)$  本身尽可能和  $y$  相似，还通过添加该项约束要求  $G(x)$  和  $y$  对空间鉴别器  $D_s$  的各层特征图的输出也尽可能相似。为此，这里将空间鉴别器  $D_s$  的各层特征图  $F^j$  的真实值  $y$  的输出  $F^j(y)$  和生成器的输出  $G(x)$  的输出  $F^j(G(x))$  考虑进来，设计如下的均方误差项  $E_{n,j}\lambda_f^j||F^j(G(x)) - F^j(y)||_2^2$ ，其中  $\lambda_f^j$  是各层的单独误差的权重。

### 4.2.2 空间鉴别器的损失函数

对于空间鉴别器  $D_s$  则需要设计一个可学习的神经网络去度量生成器的输出  $G(x)$  的生成质量和真实值  $y$  的差异，这里依照上一章的超分辨率的鉴别器的交叉熵损失函数3.22，这里构造同样的能量项  $-E_m[\log D_s(y|x)] - E_n[\log(1 - D_s(G(x)|x))]$  提高鉴别器鉴别高分辨率真实数据和生成器伪造的数据的能力。

### 4.2.3 时间鉴别器的损失函数

如上的 GAN 框架仅仅对单独一帧的生成质量进行度量，但是序列之间的连续性并没有得到充分的考虑，因为生成器  $G$  和  $D_s$  对于每一帧数据独立处理，这对于浅水方程数据这一带时间维度信息的数据类型的超分辨率任务并不足够，因为这样很难保证在训练过程中帧与帧之间能够尽可能的光滑过渡，而这在动态物理模拟中十分重要，因为最终超分辨率数据进行时间上逐帧展示的时候，时间上如果有明显的不连续性会给用户带来不适的体验感。为此本文设计了这样一个时间鉴别器  $D_t$ ，它分别将  $\tilde{Y}$  和  $\tilde{G}(\tilde{X})$  作为输入，通过度量连续三帧伪造的数据的质量或者连续三帧真实数据的质量，以提高时间连续性约束，因此对应的时间鉴别器的训练损失函数项为  $-E_m[\log D_t(\tilde{Y})] - E_n[\log(1 - D_t(\tilde{G}(\tilde{X})))]$ ，而生成器  $G$  需要通过时间鉴别器  $D_t$  的反馈梯度进行生成器的变量校正，因此其对应的能量项为  $-\lambda_{D_t} E_n[\log D_t(\tilde{G}(\tilde{X}))]$ ，其中  $\lambda_{D_t}$  表示该项对于生成器  $G$  的反馈影响，因此当  $\lambda_{D_t} = 0$  时其实就表示不使用时间鉴别器  $D_t$ ，因为此时时间鉴别器的对生成器的反向传播的梯度值为 0。

综上，本文定义的 GAN 有三个子网络组成：生成器  $G$ 、空间鉴别器  $D_s$ 、时间鉴别器  $D_t$ ，训练过程通过分别根据其对应的损失函数  $Loss_G$ 、 $Loss_{D_s}$ 、 $Loss_{D_t}$  训练，其各能量项的设计意义如上描述，最终损失函数可以总结如下：

$$Loss_{D_s}(D_s, G) = -E_m[\log D_s(y|x)] - E_n[\log(1 - D_s(G(x)|x))] \quad (4.3)$$

$$Loss_{D_t}(D_t, G) = -E_m[\log D_t(\tilde{Y})] - E_n[\log(1 - D_t(\tilde{G}(\tilde{X})))] \quad (4.4)$$

$$\begin{aligned} Loss_G(D_s, D_t, G) = & -E_n[\log D_s(G(x)|x)] - \lambda_{D_t} E_n[\log D_t(\tilde{G}(\tilde{X}))] \\ & + E_{n,j} \lambda_f^j \|F^j(G(x)) - F^j(y)\|_2^2 + \lambda_{L_1} E_n\|G(x) - y\|_1. \end{aligned} \quad (4.5)$$

为了下一章的实验描述的方便，这里将通过上述这组损失函数在  $\lambda_{D_t} > 0$  情况下训练的 GAN 通过  $\tilde{L}$  表示，代表同时度量了超分辨率任务的空间相似性质量和时间连续性质量，而将这组损失函数在  $\lambda_{D_t} = 0$  情况下训练的 GAN 通过  $L$  表示，代表仅仅度量了超分辨率任务的空间相似性质量，因此对应的生成器  $G$  的结果分别可以表示为  $\tilde{L}(x)$  和  $L(x)$ 。

### 4.3 GAN 结构

由于浅水方程的模拟数据具有明显的时序性质，因此在进行该种类型的数据的超分辨率时保持高分辨率输出的时间连续性很关键，最终超分辨率数据进行时间上逐帧展示的时候，时间上如果有明显的不连续性会给用户带来不适的体验感。因此如上所述，本文除了设计了一个判断独立一帧输出的超分辨率质量的空间鉴别器  $D_s$ ，还设计了一个判断连续帧之间的时间维度上光滑性的时间鉴别器  $D_t$ 。实验结果表明，除了空间鉴别器  $D_s$  之外引入时间鉴别器  $D_t$  在同等训练情况提高了生成器的输出质量：在时间维度上远离训练集的数据上进行测试，不使用时间鉴别器  $D_t$  的测试数据的结果质量明显不如使用了时间鉴别器  $D_t$  进行训练的测试数据的结果质量。在训练完毕之后，两个鉴别器都不再需要，测试样本的低分率数据通过生成器即可完成通过单独一帧数据进行超分辨率的任务。

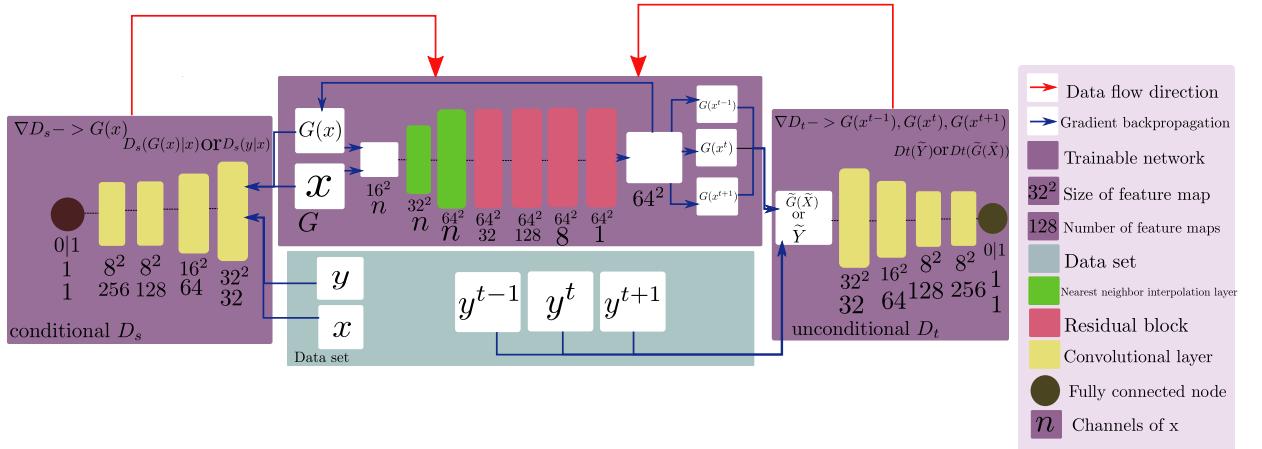


图 4.2 本文的 GAN 结构。

#### 4.3.1 生成器

虽然本质上生成器完成非线性回归任务，因此可以通过多种多样的存在的神经网络去搭建，但是实际上，其行为和生成器的输出质量和其搭建的组件密切相关，目前已有<sup>[11,52]</sup>等通过不同的结构去搭建 GAN 的生成器。本文将使用全卷积神经网络作为主体搭建生成器。所谓全卷积神经网络即放弃池化层、全连接层，只使用卷积层，其优势在于训练的网络可以适用于各样大小的输入，因为卷积的局部感受的性质，这样就可以在空间较小尺度的数据上进行训练，然后应用到更大尺度的数据上进行测试。为了加速训练，避免梯度消失和梯度爆炸的问题，本文将使用第三章中的残差块进行搭建；每个残差块有两层卷积层进行残差学习，其结果和输入并联相加构成整个残差块的输出，其形式如图4.2所示。

首先将低分辨率的小块训练数据或者整块测试数据  $x$  作为生成器的输入，先经过两层

最近邻插值层，将低分辨率数据插值到高分辨率空间（本文进行固定比例为 4 的超分辨率），然后进入 4 层残差网络（其中包含 8 层卷积层）输出超分辨率的高分辨率数据，每层残差网络的特征图数目如图所示，最后一层特征图数目为 1，即超分辨率后的高分辨率的高度场数据  $G(x)$ 。

残差块之间通过激活函数层给予非线性拟合能力，由于本文中浅水方程数据可能有负值（高度场本身起伏变化，可能在波传播过程中出现低于原始零值平面的负值情况），因此恰当的激活函数选取很关键：根据第三章的介绍可以知道原始的 ReLU 激活函数，它会导致网络传播过来的输入的负值部分无法有效向前传播（因为负值直接被截断成 0），出现名为“dead ReLU”的副作用，不适合作为该文中浅水方程数据超分辨率任务的激活函数，因此本文选取 Leaky ReLU 激活函数作为该任务的激活函数。下一章实验结果表明使用 Leaky ReLU 激活函数层相比较 ReLU 激活函数层更有利于结果质量。

### 4.3.2 时间鉴别器和空间鉴别器

鉴别器由于不参与最后的超分辨率任务，因此只要引入足够的非线性就可以实现判断，使用复杂的结构对最终的生成器的质量影响不显著。考虑到卷积神经网络的自动提取特征，局部视野，待优化网络参数较少等优势，很适合作为鉴别器子网络的组成基础，因此本文使用如图4.2所示的结构构成本文的鉴别器。

空间鉴别器  $D_s$  是一个条件鉴别器，即它需要低分辨率的结果作为  $x$ ,  $y$  和  $G(x)$  分别是高分辨率真实数据和生成器伪造数据，由此本文将  $G(x) + x$  或者  $y + x$  作为输入送给空间鉴别器，此时经过 4 层卷积神经网络不断减少数据的分辨率（倒数最后两层的分辨率相同，不再对半减少分辨率，如图4.2所示），一方面为了抽取足够的主要特征，一方面保证最后一层全连接的可学习变量尽可能的少。最后全连接的输出是一个单独值进行二分类（不需要独热编码，因为是二分类问题），进行判断是真是假，然后通过空间鉴别器的损失函数层进行训练。

时间鉴别器  $D_t$  是一个无条件鉴别器，即它不需要低分辨率的结果作为条件约束，只接受  $y$  和  $G(x)$  分别是高分辨率真实数据和生成器伪造数据， $x$  同样表示低分辨率数据。但是和空间鉴别器不同之处在于，其同时接受连续三帧数据即  $y^{t-1}, y^t, y^{t+1}$  或者  $G(x^{t-1}), G(x^t), G(x^{t+1})$  作为输入，经过 4 层卷积神经网络不断减少数据的分辨率（倒数最后两层的分辨率相同，不再对半减少分辨率，如图4.2所示），一方面为了抽取主要特征，一方面保证最后一层全连接的可学习变量尽可能的少。最后全连接的输出是一个值进行二分类，进行判断是真是假，然后通过时间鉴别器的损失函数层进行训练。因为其只是度量三帧之间的连续性，因此这里

不需要引入低分辨率数据  $x$  作为条件约束。

## 4.4 数据生成方法

### 4.4.1 浅水方程模拟的数据生成

训练数据的选取对于数据驱动的方法的结果至关重要，对于浅水方程模拟这一复杂的物理过程而言，想要构造恰当的数据集相比较图像或者视频这种归一化之后的数据更为困难，而且模拟过程本身中随着时间，其复杂的边界条件，相互干涉叠加的物理性质等因素会导致该问题无法很好的被处理。从实验中发现，即使是单相浅水方程，由于其干涉叠加，反射边界条件，初始高度无限制，物理参数可变等问题，无法构造有限的训练集进行泛化性非常高的超分辨率。因此本文在一定的约束下进行超分辨率实验，通过第三章的方法在高分辨率 (256\*256) 下求解浅水方程，固定时间间隔，重力加速度，初始流体表面高度固定(高度为 5)，初始时速度为 0，只改变如下的初始条件去生成数据集。对于每个序列，本文从高分辨率区域 (256\*256) 中随机选择三个位置将表面高度从固定值 (高度为 5) 提升至高度为 15，如此生成 5 个这样的序列，每个序列 90 帧。因为高度场数据本身范围无限制，这点和图像非常不一样 (图样的值范围始终处于 [0,1] or [0,255])，导致浅水方程模拟的数据的超分辨率更为困难：我们很难构造有限的数据集去使得其在各种各样的测试集下表现良好，因此本文构造相似性较高但不一样的数据集进行实验，在这样一个给定的初边值条件下， $h$  的变化范围有一定的上下界 [4.5,15]，速度  $u$  的分量绝对值有一定的上下界 [0.15,3.2]。获得高分辨率数据之后本文对数据进行固定  $\sigma$  参数的高斯模糊并且通过 4 倍下采样因子的最近邻插值获得低分辨率数据。下采样算子记作  $\mathcal{D}$ 。

### 4.4.2 旋转对称的数据扩充方法

在该种初值条件下生成的浅水方程数据呈现出环形向四周传播，在遇到边界后进行镜面反射，镜面反射回来的波纹会和原始的波纹发生干涉叠加的现象，如图4.3。因此在这种情况下需要考虑训练集中采样的数据在该种边界条件下呈现出的不对称性，即特征在各个朝向上并不相同，如图4.3中朝向左上的特征就和朝向右下的特征就很不相同（朝向左上的半圆部分形成的干涉叠加明显少于其朝向右下部分形成的干涉叠加），这是由于波在传播过程中遇到障碍物的时间有先后，先后发生干涉，导致呈现的该帧的特征不相同。但是这种特征很有可能在测试集中发生旋转出现（考虑将初始值为 15 的高度位置进行旋转，那么该帧的特征也会发生类似的旋转，或者移动初始值为 15 的高度位置，也会导致发生干涉叠加的

先后顺序发生变化)。而又因为本文使用了卷积神经网络作为主要子网络的组成部分，但是通过第三章可以知道卷积层学习的特征不满足旋转等价性，即当训练集中的样本被旋转之后或者使用和训练样本方向朝向相似度低的样本进行测试，将不会和原始训练样本一样被恢复的同样好，即如果是相同的样本进入网络，但是即使只是朝向发生整体旋转，那么超分辨率恢复的质量也会差很多。但是，注意到，浅水方程数据的超分辨率过程是一个有物理意义的重建过程，需要遵守潜在的物理过程的不变性性质，如伽利略不变性；并且神经网络训练也需要尽可能地提高训练获得的网络的泛化性，尽量避免过拟合现象的发生，将网络应用于和训练集特征不同的测试集时结果也应当尽可能保真，因此旋转等价性需要尽可能地被满足。而另一方面，如果是通过大量采样生成训练集，以规避采样获得的训练集的有偏性，会导致训练时间过长，因此通过合理的数据扩充和数据生成策略在较小的训练集上获得上述不变性较好的满足是更为合适的选择。综上所述，本文考虑将原始训练数据进行  $90 * i (i = 1, 2, 3)$  度的整体旋转扩充原始数据集，进行训练。这样数据扩充之后的训练集在特征提取过程中有偏性会降低，在将网络运用到和原始训练集中的样本的特征朝向性差异较大的数据上进行测试时，结果会比不做数据扩充的结果更好，如下一章实验所示。

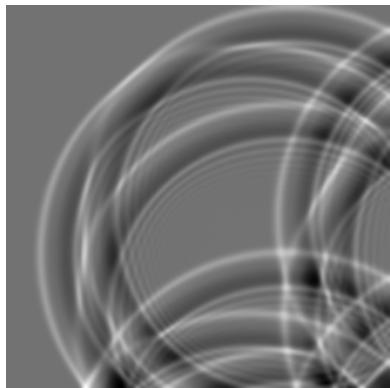


图 4.3 如图为高分辨率浅水方程模拟的结果，可以看出水波呈现出以圆环的形式向四周扩散，碰到障碍物后镜像反射，和原始的波纹产生叠加的现象。

## 4.5 GAN 训练方法

除了损失函数和网络结构的设计，数据生成和扩充方式的选取之外，训练方法对生成对抗网络的最终结果的好坏也至关重要。对于不同的问题，不同的训练方法可能优劣不一，因此需要根据问题本身的性质和实验结果选择恰当的训练方法。

网络变量的初始化。本文的神经网络中待学习的变量包括卷积层的卷积核值  $k$  和偏移量  $b_c$ ，全连接层的连接权重  $W$  和偏移量  $b_f$ ，对于  $k, W$  本文使用随机正态分布初始化这些

变量，对于  $b_c, b_f$  本文使用常数 0.1 初始化这些变量。

神经网络的训练中通常根据梯度信息优化目标函数更新网络变量，每次的梯度信息来源对整个训练过程影响很大。常用的包括：

全局梯度下降，即使用整个训练集的梯度作为变量的更新方向，并指定特定步长更新变量，该步长在深度学习中常被称为学习率。全局梯度下降的方式对于较小的训练集合适，可以获取非常准确的梯度信息，但是训练集较大时，常因为 GPU 内存不够无法有效的计算。

随机梯度下降，和全局梯度下降不同，其梯度信息只来自于训练集中的一个样本，因此可以快速估计梯度，但是因为该梯度并不准确，可能相互制约，导致训练无法充分收敛，往往需要非常小的学习率否则会很难收敛，并且没有充分利用 GPU 内存资源和并行能力，造成计算资源的浪费。

一个中间的选择则是通过训练集中的一一个真子集（深度学习中称之为小批次，mini-batch）的梯度来进行变量更新，一方面有助于充分利用 GPU 的内存资源和并行能力，又能够提供一个较好的梯度估计。现在的框架一般基于该方案进行梯度估计，此时需要确定该真子集的大小（batch size），深度学习中称此变量为批量大小。

如前面所示，确定梯度方向之后，需要给定一个步长（学习率），即沿着该方向需要一次走多远，学习率对于训练有很大的影响，通常会在刚开始时给定一个较大的学习率，加速训练，再在训练后期逐渐减小学习率，进行精细化的微调，ADAM 优化方法可以动态的调整学习率，因此在 GAN 训练中被主要使用。这里本文使用 ADAM 优化器初始时的学习率为 0.0002，并在训练迭代次数的后一半将其减小为原始学习率的 1/20，实验的迭代次数在 40k-50k 左右，并且训练过程不需要 drop out 的技巧。

#### 4.5.1 基于 mini-batch 梯度的训练

如上所述，训练方法对生成对抗网络的最终结果的好坏也至关重要。对于不同的问题，不同的训练方法可能优劣不一，因此需要根据问题本身的性质和实验结果选择恰当的训练方法。在下一章的实验中本文首先使用基于 mini-batch 的梯度进行生成对抗网络的训练，训练的算法框架如算法1所示，但是实验结果表明一方面它可能在一些情况造成 GAN 的训练发散，另一方面，即使不发散，也很难恢复出足够多的高频细节，并不能满足本文的目标。为此本文发现如果能够有效的使用全局梯度会缓解该问题，不仅能够有效的避免网络的训练发散的产生，而且能够在测试时恢复出更多的高频细节，但是由于 GPU 内存限制实际中不能使用使用过大的训练集，因此本文提出使用异步的训练策略解决该问题。

**Algorithm 1** 基于 mini-batch 梯度的训练算法

---

```

1: for 训练的迭代次数 do
2:   for  $D_s$  的梯度更新次数 do
3:     在数据扩充后的训练集中获取 mini batch  $x, y$ 
4:     通过 mini batch 的梯度, 使用 ADAM 优化器更新  $D_s$  的待学习变量
5:   for  $D_t$  的梯度更新次数 do
6:     在数据扩充后的训练集中获取 mini batch  $\tilde{X}$  和  $\tilde{Y}$  和  $\tilde{G}(\tilde{X})$ 
7:     通过 mini batch 的梯度, 使用 ADAM 优化器更新  $D_t$  的待学习变量
8:   for  $G$  的梯度更新次数 do
9:     在数据扩充后的训练集中获取 mini batch  $x, y, \tilde{X}$ 
10:    通过 mini batch 的梯度, 使用 ADAM 优化器更新  $G$  的待学习变量

```

---

**4.5.2 异步的基于全训练集梯度的训练**

所谓异步的基于全训练集梯度的训练, 一方面针对 GPU 内存限制不能一次性装入大量样本的制约, 首先确定最大可装入批大小, 通过将训练集切分成若干互补包含的最大可装入批样本, 每次计算装入样本的梯度, 再在所有的批样本被处理完毕之后, 将所有批样本给出的更新梯度相加, 获得全局梯度, 然后再进行变量更新, 训练的算法框架如算法2所示。另一方面本文发现, 相近的序列之间具有较强的相似性, 因此本文将一次模拟的 90 帧数据, 每隔 10 帧取中间一帧数据作为训练集, 再通过旋转对称的数据扩充增加训练集样本个数。这种方式, 能够一方面减少训练样本的个数需求, 另一方面尽可能提高该算法的泛化性。下一章的实验表明, 相比较基于 mini-batch 梯度的训练, 该方法能够恢复出更多的高频信息。

**4.5.3 训练和测试时使用的低分辨率数据**

通常的超分辨率问题中, 一般仅仅是将单张低分辨率图像当做输入  $x$  即可, 毕竟, 本文最终只关心输出的高分辨率高度场  $G(x)$  和真实的高分辨率高度场  $y$  之间尽可能相似。但是, 考虑在浅水方程模拟数据的超分辨率中, 其实是在进行一个降采样的物理过程的超分辨率重建, 这和传统的图像视频超分辨率都十分不同, 因此如果能恰当地使用更多的物理信息(比如速度信息), 会对最终结果的收敛性带来好的影响。下一章实验结果也表明如果提供给神经网络额外的物理信息, 将显著地提高收敛性和输出的高分辨率高度场的质量。这里的实验不需要改变网络结构, 唯一的差别是 GAN 网络的生成器输入从将低分辨率高

**Algorithm 2** 异步的基于全训练集梯度的训练算法

---

```

1: for 训练的迭代次数 do
2:   for  $D_s$  的梯度更新次数 do
3:     for 整个训练集被分成的 mini batch 的个数 do
4:       获取一个独立不重复的 mini batch  $x, y$ 
5:       获取该 mini batch 的梯度并保存
6:       将所有独立不重复的 mini batch 的梯度加和
7:       使用加和的全局梯度和 ADAM 优化器更新  $D_s$  的待学习变量
8:   for  $D_t$  的梯度更新次数 do
9:     for 整个训练集被分成的 mini batch 的个数 do
10:    获取一个独立不重复的 mini batch  $\tilde{X}$  和  $\tilde{Y}$  和  $\tilde{G}(\tilde{X})$ 
11:    获取该 mini batch 的梯度并保存
12:    将所有独立不重复的 mini batch 的梯度加和
13:    使用加和的全局梯度和 ADAM 优化器更新  $D_t$  的待学习变量
14:   for  $G$  的梯度更新次数 do
15:     for 整个训练集被分成的 mini batch 的个数 do
16:       获取一个独立不重复的 mini batch  $x, y, \tilde{X}$ 
17:       获取该 mini batch 的梯度并保存
18:       将所有独立不重复的 mini batch 的梯度加和
19:       使用加和的全局梯度和 ADAM 优化器更新  $G$  的待学习变量

```

---

度场  $h$  当做  $x$ , 改为将低分辨率的高度场  $h$  和速度场  $u$  当做  $x$ 。下一章的实验结果表明在相同的迭代次数之后, 使用  $h$  作为输入的网络的输出仍然具有明显的分块和模糊的问题, 但是使用  $h + u$  作为输入的网络的输出明显具有更高的质量, 这也进一步验证了猜测, 降采样的物理过程的超分辨率重建需要更多的物理信息加以辅助神经网络进行拟合。直觉上, 辅助的速度场信息  $u$  能够自动地提供给网络更多的拟合约束信息, 帮助网络更快地拟合出和潜在的物理过程相关的超分辨率映射。

## 4.6 本章小结

本章将介绍通过 GAN 进行浅水方程模拟超分辨率的算法设计细节, 经过对浅水方程数据超分辨率任务过程中的独特性质的思考, 考虑到了其时间维度上的光滑性需求, 超分

分辨率任务的旋转等价性需求，数据的可能为负值的性质，物理过程中速度信息的影响等关键因素，进行了神经网络结构的设计，损失函数的设计，数据的生成扩充方法和网络的训练方法的设计。总结如下：1. 引入度量浅水方程模拟时间维度光滑性的时间鉴别器，2. 使用浅水方程速度场信息作为生成器的输入的一部分，3. 使用异步的基于全训练集梯度的训练方法2，4. 使用旋转对称的数据扩充方法扩充训练集，5. 使用 Leaky ReLU 激活函数作为生成器的激活函数，代替一般采用的 ReLU 激活函数，从而在结果上能通过低分辨降采样数据中推测出较多的高频细节。下一章将展示本章所述算法的实验结果，验证每一步技术选取的必要性和可行性，最后还会讨论了本文方法的局限性和该研究课题的困难所在。

## 第5章 实验结果展示和分析

本章将展示第四章所述算法的实验结果，首先本章将介绍下算法实现的实验环境和数据集，并针对本文提出的改进验证了每一步技术选取的必要性和可行性，并讨论了本文方法的局限性和该研究问题的困难所在。最后是本章小结。

### 5.1 实验环境

本文的实验环境如下：

内存：32G

操作系统：Ubuntu 16.04

GPU：NVIDIA GeForce GTX 1080 Ti

CPU：Intel(R) Core(TM) i7-8700k CPU@3.70GHz

编程语言：1.C++ 编写浅水方程求解器生成数据；2.python 训练神经网络

本文通过 Tensorflow 编写深度学习算法。Tensorflow 是一个开源的深度学习框架，其基于静态图设计，内部集成了包括神经网络搭建，自动微分求导，基于梯度的数值优化等模块，能够帮助快速实验和开发深度学习算法，其张量式的变量表示可以和开源库 numpy 无缝结合，提高数据的处理效率。Tensorflow 中大量矩阵操作内部实现了基于 GPU 的并行，因此可以高效地进行训练。本文实验通过 Tensorflow 的低级接口搭建上一章算法中的 GAN 模型，然后根据异步的基于全训练集梯度的训练算法2或者基于 mini-batch 梯度的训练算法1进行训练。

### 5.2 结果分析和对比

根据上一章的数据生成方法，生成 5 组序列，每个序列各有 90 帧，每间隔 10 帧，取其中间一帧加为训练集，并做可能的旋转对称的数据扩充，剩余的帧作为测试集。本章延续上一章算法中的符号，这里将使用如下表5.1中的表示代表各种情况下的结果。为了展示

abb.	含义
Hi	高分辨率真实高度场结果
Lo	低分辨率输入
$\tilde{L}$	同时使用空间鉴别器和时间鉴别器的结果
$L$	只使用空间鉴别器的结果

表 5.1 实验结果的缩写对应

上的公平性，避免因为高分辨率数据和低分辨数据因为分辨率不一致导致的认为低分辨率非常模糊的感受，本文在展示低分辨率数据时，均将其通过 4 倍的最近邻插值到高分辨率维度。

本文的实验结果如下：

### 1. 生成器使用的低分辨率数据相关

- 是否使用浅水方程降采样之后的低分辨率速度场作为生成器输入  $x$  的一部分

### 2. 损失函数相关：

- 调节损失函数中的超参，学习率或者对于生成器和鉴别器的迭代次数
- 是否使用度量浅水方程模拟时间维度光滑性的时间鉴别器并且通过其损失函数进行训练

### 3. 网络结构相关

- 对于生成器使用激活函数 ReLU 或 Leaky ReLU 的影响 (ReLU vs Leaky ReLU)

### 4. 数据集相关

- 是否使用旋转对称的数据扩充后的数据集训练

### 5. 训练方法相关

- 使用基于 mini-batch 梯度的训练方法1还是使用异步的基于全训练集梯度的训练方法2

### 5.2.1 使用基于 mini-batch 梯度的训练方法的问题

训练方法对生成对抗网络的最终结果的好坏也至关重要。对于不同的问题，不同的训练方法可能优劣不一，因此需要根据问题本身的性质和实验结果选择恰当的训练方法。首先，本文在构造的训练集上通过基于 mini-batch 梯度的训练方法1训练，本文实验发现无论是否使用时间鉴别器、如何调整学习率、训练子网络使用的损失函数中的超参（人工设置的权重）或者训练方法1中生成器和鉴别器分别迭代的次数（平衡它们的训练程度，避免其中一个过强），无论在训练集或者测试集上测试时都无法恢复出足够的高频信息，总是出现模糊的问题，如图5.1所示。

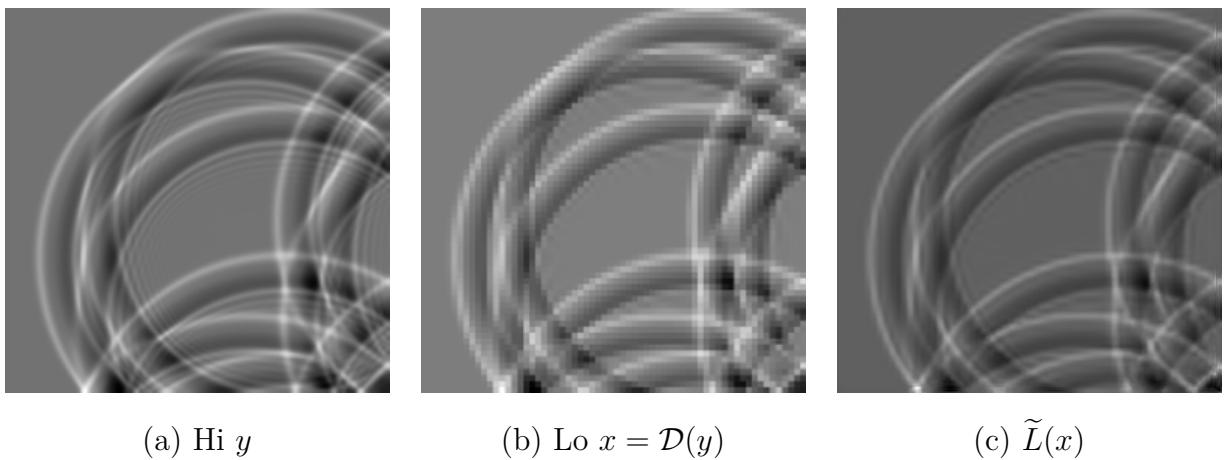


图 5.1 使用基于 mini-batch 梯度的训练方法始终只能获得模糊的高分辨率恢复结果，如 (c)。

另外，实验发现基于 mini-batch 梯度的训练方法，在某些给定的超参情况下，训练会很容易地发散，例如当实验中将4.5中的  $\lambda_f^j$  设置成非常小的负值时 (-1e-5)，经过约 40000 次迭代后，GAN 的结果会非常奇怪，如图5.2所示。

在这种情况下，为了消除训练方法对该问题的影响，这里暂时简化训练数据为单帧数据以公平地展示其他技术选择的必要性和可行性，此时训练集只有一帧浅水方程数据展示出来的特征，通过观察该样本的训练结果以查看其他因素（如是否使用时间鉴别器，是否使用浅水方程速度场信息作为生成器的输入的一部分，是否使用旋转对称的数据扩充方式，是否使用 Leaky ReLU 代替一般的 ReLU 作为生成器的激活函数）对结果的影响，因为此时无论是基于 mini-batch 梯度的训练方法1还是使用异步的基于全训练集梯度的训练方法2都是一样的，mini-batch 等价于全训练集。

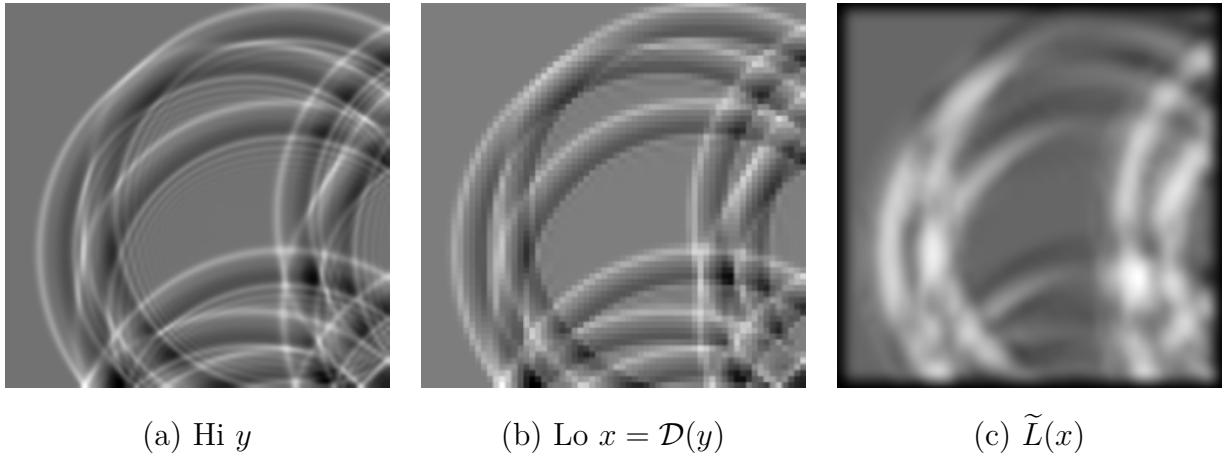


图 5.2 基于 mini-batch 梯度的训练方法，在某些给定的超参情况下，会容易发散，输出有问题的奇怪结果，如 (c)。

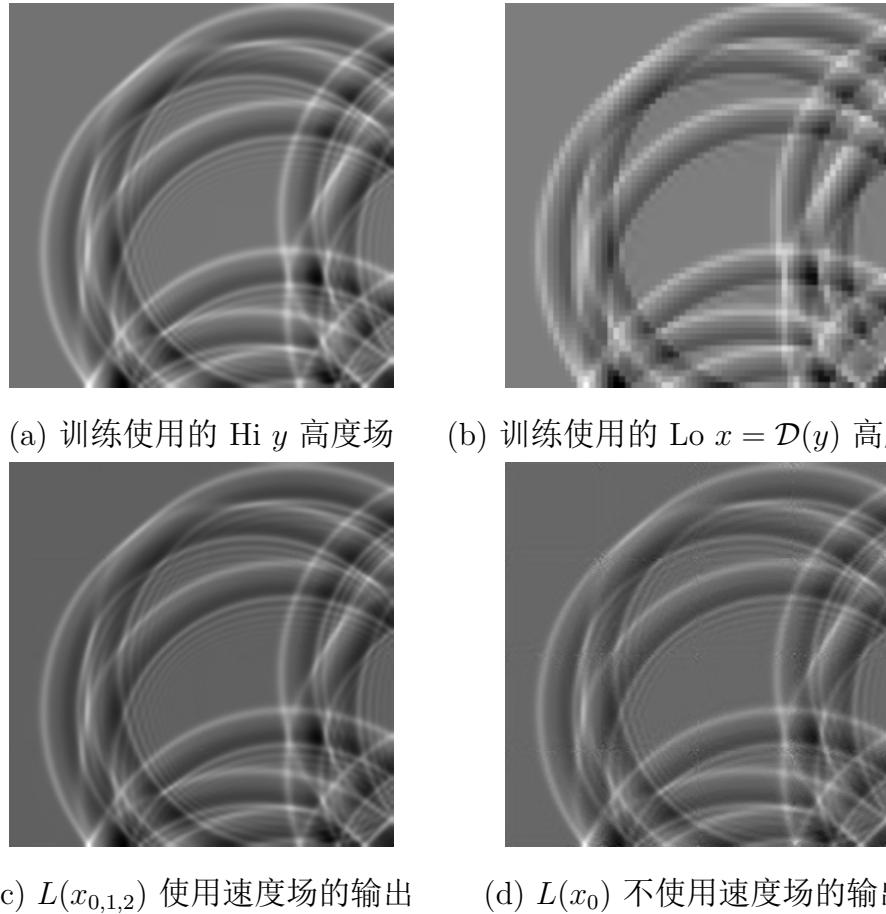
### 5.2.2 是否使用低分辨率速度场

考虑在浅水方程模拟数据的超分辨率中，其实是在进行一个降采样的物理过程的超分辨率重建，这和传统的图像视频超分辨率都十分不同，因此如果能恰当地使用更多的物理信息（比如速度信息），会对最终结果的收敛性带来好的影响。在上述的基础之上，实验结果确认了加入浅水方程降采样低分辨率速度场数据会更好地帮助拟合。本文的实验结果表明如果输入的低分辨率数据包括速度，那么生成器输出的伪造数据将具有更好的连续性，这里实验只使用  $L$  即只使用空间鉴别器而仅仅度量空间相似性质量。实验结果表明，在单帧训练的情况下，GAN 的生成器具有足够的非线性能力拟合从低分辨率到高分辨率数据的映射，并且加入低分辨率速度场作为生成器输入  $x$  的一部分会帮助提高结果质量。其结果如图5.3所示。

### 5.2.3 是否旋转对称的数据扩充

训练集中的数据高度决定了 GAN 学习的特征，由上述的一组实验可以看出，在单帧情况下 GAN 事实上有能力恢复出高频信息，但是测试集是否和训练集包含相似的特征非常重要。如果二者的特征相似，如图5.4所示，或者极端情况下，测试集属于训练集，那么结果如上图5.3所示，单帧情况确实可以恢复出高频信息，但是当二者的特征不相似时，则不适用。

由于这里使用了卷积神经网络作为主要子网络的组成部分，但是通过第三章可以知道卷积层学习的特征不满足旋转等价性，即当训练集中的样本被旋转之后或者使用和训练样本方向朝向相似度低的样本进行测试，将不会和原始训练样本一样被恢复的同样好，如



**图 5.3** 实验结果表明在相同的训练迭代次数下，使用低分辨率速度场数据作为生成器的输入的一部分可以获得更连续光滑的输出。

图5.5所示。

但是注意到，浅水方程数据的超分辨率过程是一个有物理意义的重建过程，需要遵守潜在的物理过程的不变性性质，如伽利略不变性；并且神经网络训练也需要尽可能地提高训练获得的网络的泛化性，尽量避免过拟合现象的发生，将网络应用于和训练集特征不同的测试集时结果也应当尽可能保真，因此旋转等价性需要尽可能地被满足。因此，本文考虑将原始训练数据进行  $90 * i (i = 1, 2, 3)$  度的整体旋转扩充原始数据集，进行训练。这样数据扩充之后的训练集在特征提取和非线性回归过程中有偏性会降低，在将网络运用到和原始训练集中的样本的特征朝向性差异较大的数据上进行测试时，结果会比不做数据扩充的结果更好，如图5.6所示。

#### 5.2.4 是否使用时间鉴别器

由于浅水方程的模拟数据具有明显的时序性质，因此在进行该种类型的数据的超分辨率时保持高分辨率输出的时间连续性很关键，最终超分辨率数据进行时间上逐帧展示的时

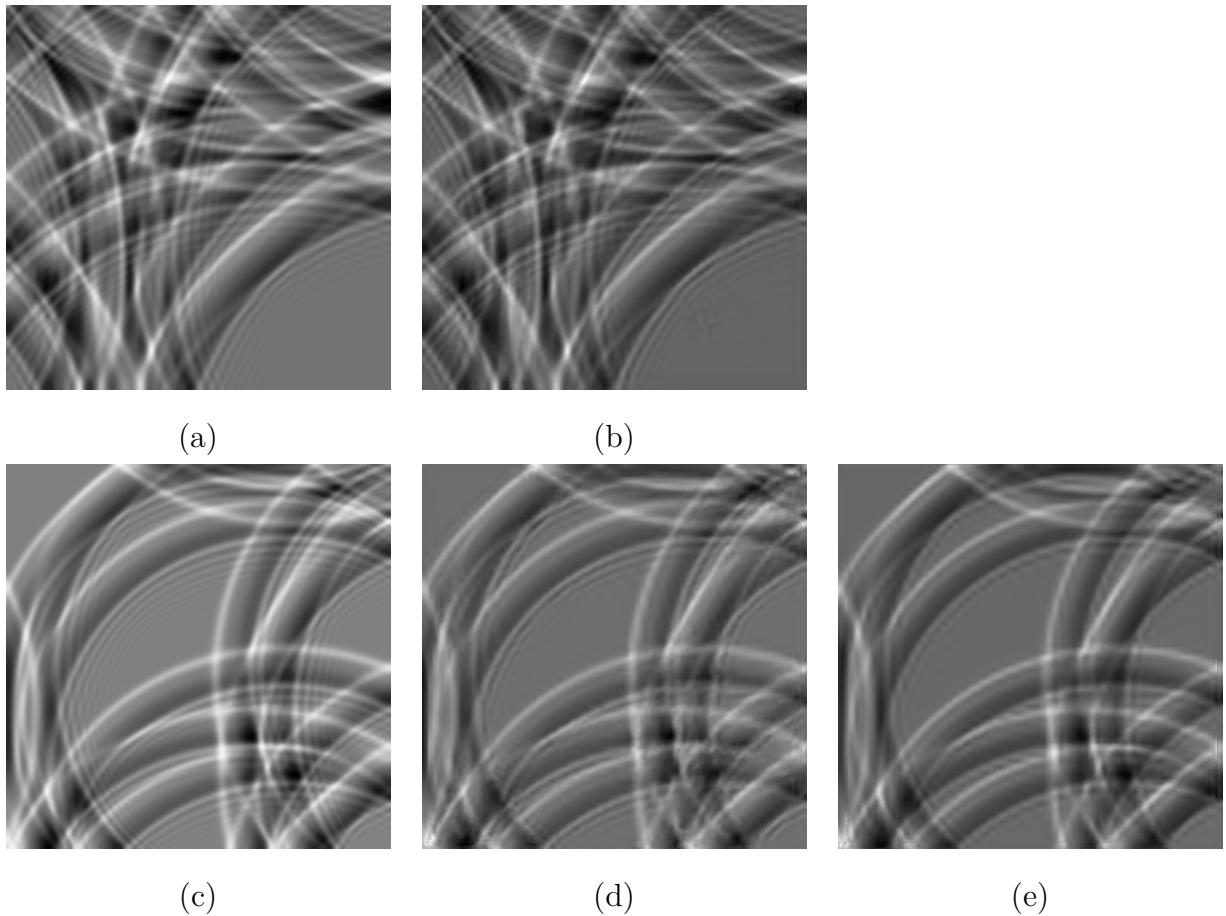


图 5.4 (a)  $\text{Hi } y_{train}$  (b)  $L(x_{train})$  (c)  $\text{Hi } y_{test}$  (d)  $L(x_{test})$  表明仅仅使用单帧训练的情况下，实验表明如果测试样本具有和训练样本相似的特征，高频信息可以被恢复出来。(e)  $L(x_{test})$  和 (d) 相同的测试样本，但是 (e) 是基于 mini-batch 梯度的训练方法，这些高频信息就会丢失，在相同的迭代次数下，恢复的结果就会模糊，失去细节。

候，时间上如果有明显的不连续性会给用户带来不适的体验感。因此如上所述，本文除了设计了一个判断独立一帧输出的超分辨率质量的空间鉴别器  $D_s$ ，还设计了一个判断连续帧之间的时间维度上光滑性的时间鉴别器  $D_t$ 。本文实验发现在不使用时间鉴别器的情况下，当测试集和训练集的特征非常不相似时，测试集上的结果非常不好，如图5.7(d) 所示，但是如果实验引入时间鉴别器并且仅仅将训练集中的数据的相邻前后两帧加入训练集，测试集的结果可以被明显地提高，如图5.7(e) 所示。

### 5.2.5 生成器的激活函数的选取

本文的生成器采用基于全卷积神经网络的残差块作为基础模块。残差块之间通过激活函数层给予非线性拟合能力，由于本文中浅水方程数据可能有负值（高度场本身起伏变化，可能在波传播过程中出现低于原始零值平面的负值情况），因此恰当的激活函数选取很关

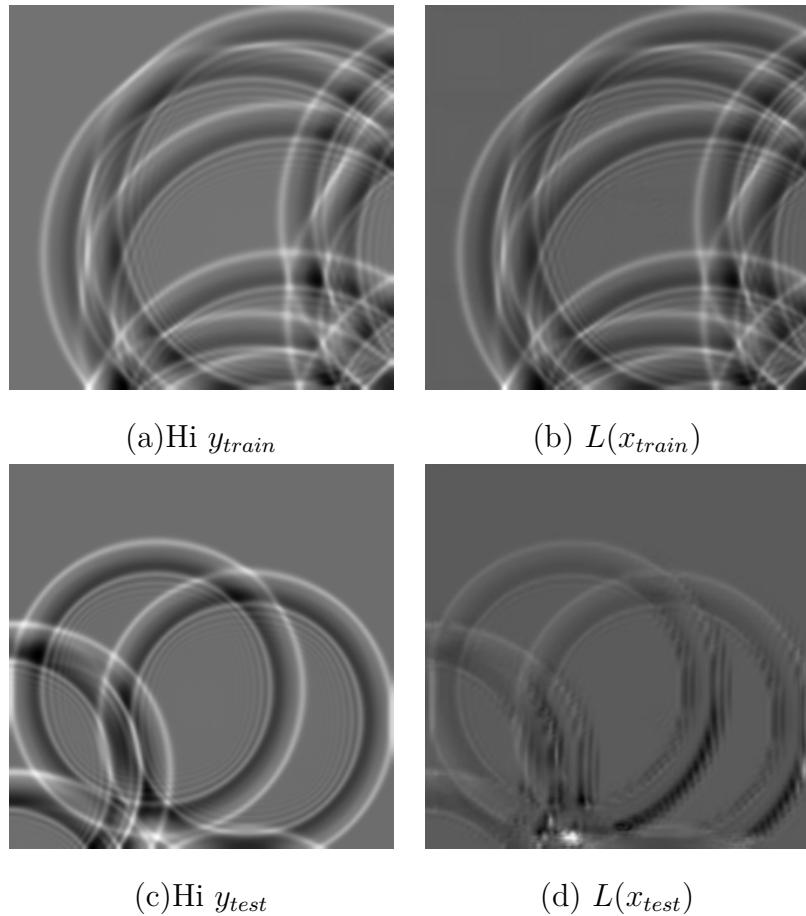


图 5.5 CNN 学习的特征不满足旋转等价性。如 (b) 和 (d) 中所示, (b) 作为训练样本, 其通过 GAN 恢复了足够的高频信息, 但是这些特征的朝向主要指向左上方, 而 (d) 作为测试样本, 则明显地显示其指向左上的特征恢复的质量明显好于指向右下的特征。

键: 根据第三章的介绍可以知道原始的 ReLU 激活函数, 它会导致网络传播过来的输入的负值部分无法有效向前传播 (因为负值直接被截断成 0), 出现名为 “dead ReLU” 的副作用, 不适合作为该文中浅水方程数据超分辨率任务的激活函数, 因此本文选取 Leaky ReLU 激活函数作为该任务的激活函数。实验结果表明相比较使用 ReLU 函数作为生成器的激活函数, 使用 Leaky ReLU 函数作为生成器的激活函数可以避免 ReLU 带来的稀疏选择的负面影响, 这里使用  $L$  仅仅度量空间相似性的质量训练本文的网络。其结果如图5.8所示。

### 5.2.6 训练方法的改进

虽然 GAN 在单帧训练上可以恢复出足够的高频信息, 如图5.3, 但是当本文的实验使用 mini-batch 梯度的训练方法1时会生成模糊的结果, 如图5.1。虽然本文的实验结果表明将生成器的激活函数改为 Leaky ReLU, 使用速度场作为生成器的输入的一部分, 使用时间鉴别器都有助于提高结果质量, 但是依然无法有效地克服模糊的问题。因此本文认为是在

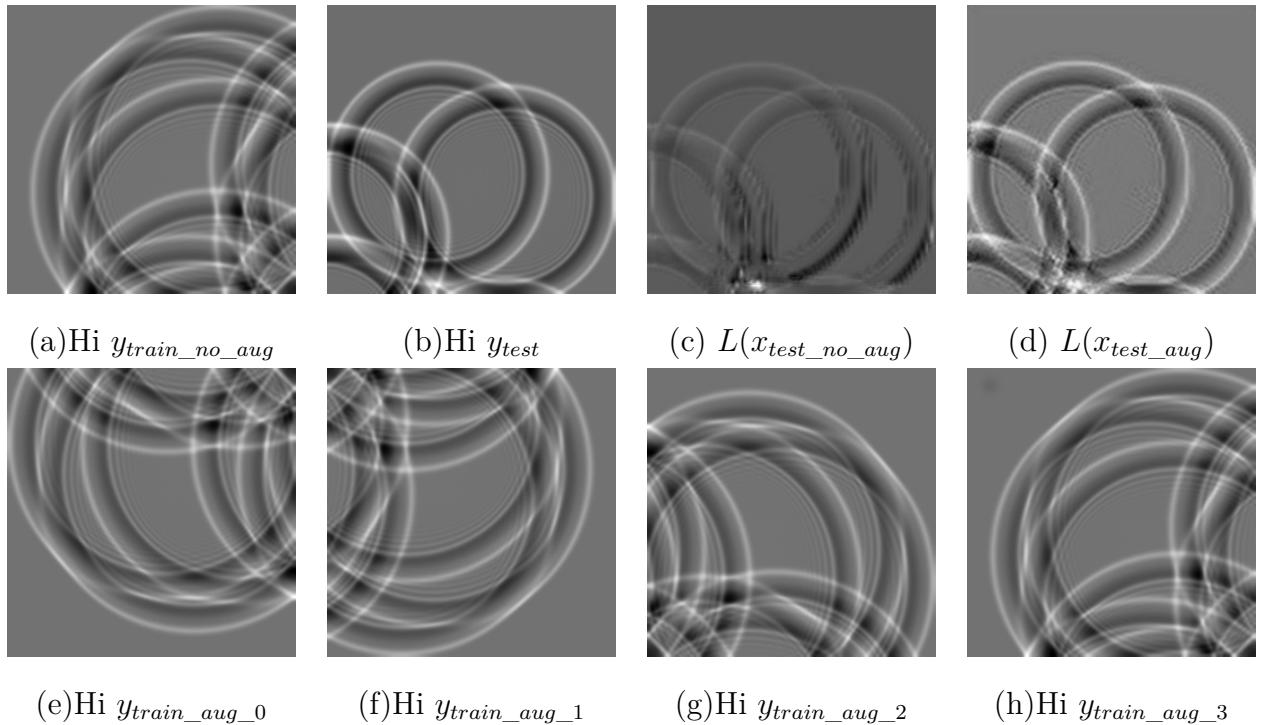


图 5.6 使用旋转对称的数据扩充后可以获得更好的泛化性。图 (a,b,c) 分别是 (a). 原始的不用数据扩充进行训练的数据的高分辨率参考值, (b). 进行测试的数据的高分辨率参考值和 (c). 神经网络对测试数据 (b) 进行超分辨率获得的结果, 可以看出由于 CNN 学习的特征不满足旋转等价性, 因此 (c) 的结果中指向右下的特征相比指向左上的特征的恢复质量较差。本文考虑将原始训练数据进行  $90 * i (i = 1, 2, 3)$  度的整体旋转扩充原始数据集, 如图 (e,f,g,h), 进行训练。这样数据扩充之后的训练集在特征提取和非线性回归过程中有偏性会降低, 在将网络运用到和原始训练集中的样本的特征朝向性差异较大的数据上进行测试时, 结果会比不做数据扩充的结果更好, 如图 (d) 所示, 图 (d) 的结果明显比图 (c) 更好。

浅水方程的超分辨率的训练中, 局部梯度无法提供一个非常好的全训练集梯度的近似, 为此本文试图使用全训练集梯度优化变量。由于 GPU 内存限制, 为此本文提出异步的基于全训练集梯度的训练方法2。实验结果表明为了恢复出更多的高频信息, 使用异步的基于全训练集梯度的训练方法2比基于 mini-batch 梯度的训练方法1更有效果。如图5.9(a),(b) 和 (c) 所示。

### 5.3 本文方法的局限性

#### 5.3.1 初值和边界条件的局限

对于该方法的泛化性问题, 浅水方程的高度场无限制这一特性严重地制约了其泛化性。当前的改进的方法也只能在相似的测试集上获得不错的成绩, 但是训练的 GAN 用于其他

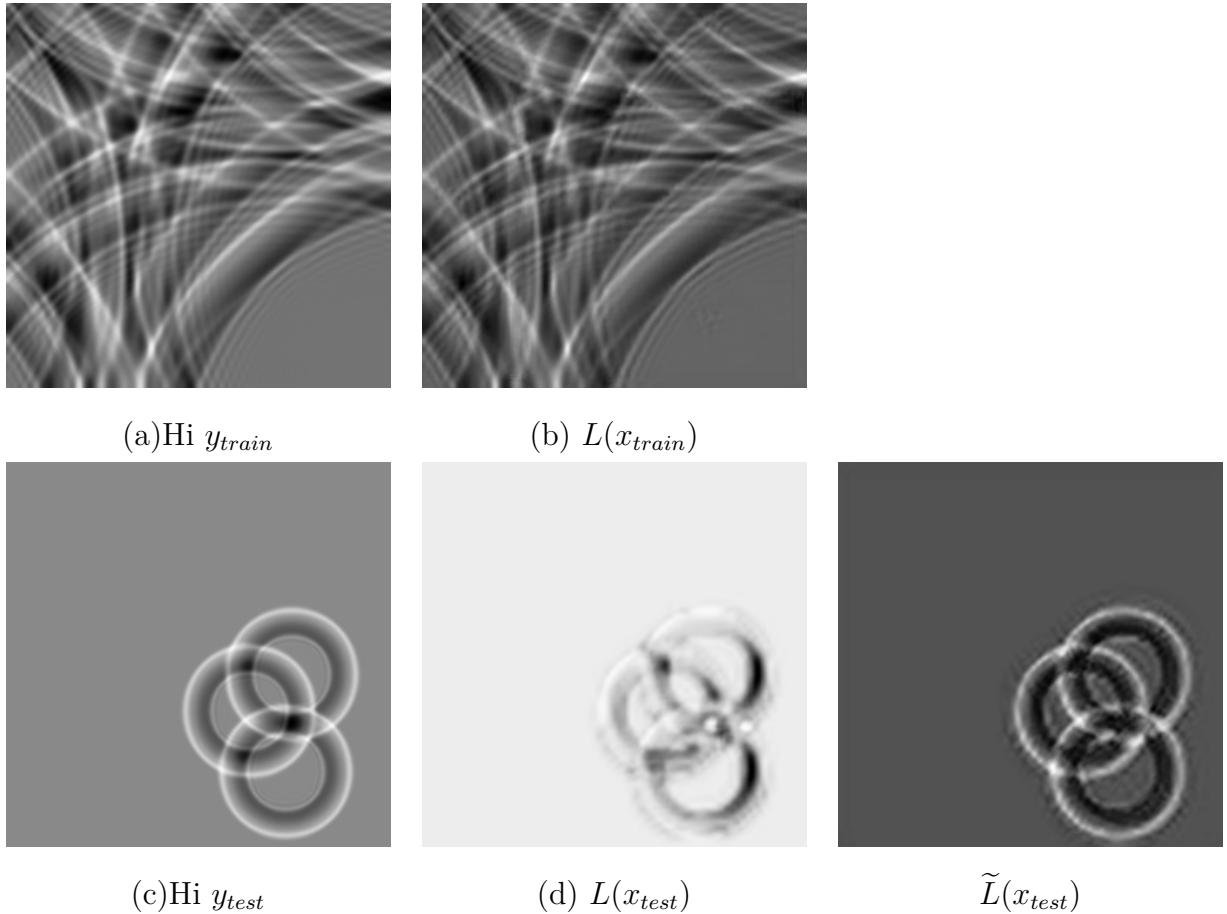


图 5.7 如果测试的样本不具有和训练样本相似的特征，那么恢复结果十分糟糕，如 (d) 所示，但是如果实验在训练时使用时间鉴别器，那么测试集的结果会被明显的提高，如 (e) 所示。

的边界条件 (如改变重力加速度，流体密度，或者初始的高度场的绝对高度等) 下生成的数据的超分辨率任务时，就无法适用。如图5.9(d) 和 (e) 所示。

### 5.3.2 训练时间

异步的基于全训练集梯度的训练方法可以恢复出更多的高频细节，但是同时它需要更长的训练时间因为其一次更新变量的时间，本文需要计算所有的不相交的 mini-batch 的梯度并且加和，所以在非并行获得 mini-batch 梯度的情况下，训练时间的花费大约是基于 mini-batch 梯度的训练方法的 [训练集大小/mini-batch 大小] 倍。未来可以通过增加 GPU 内存或者并行计算每个 mini-batch 的梯度来缓解该问题。

## 5.4 本章小结

本章详细地展示了第四章所述算法的实验结果，首先本章介绍了算法实现的实验环境和数据集，针对本文提出的改进验证了每一步技术选取的必要性和可行性：实验结果表明，对于使用生成对抗网络进行浅水方程的模拟数据的超分辨率，可以通过 1. 引入度量浅水方程模拟时间维度光滑性的时间鉴别器，2. 使用浅水方程速度场信息作为生成器的输入的一部分，3. 使用异步的基于全训练集梯度的训练方法2，4. 使用旋转对称的数据扩充方法扩充训练集，5. 使用 Leaky ReLU 激活函数作为生成器的激活函数，代替一般采用的 ReLU 激活函数，这 5 点可以帮助提高结果质量，恢复出更多的高频信息。最后本章讨论了本文方法的局限性和该研究问题的困难所在。

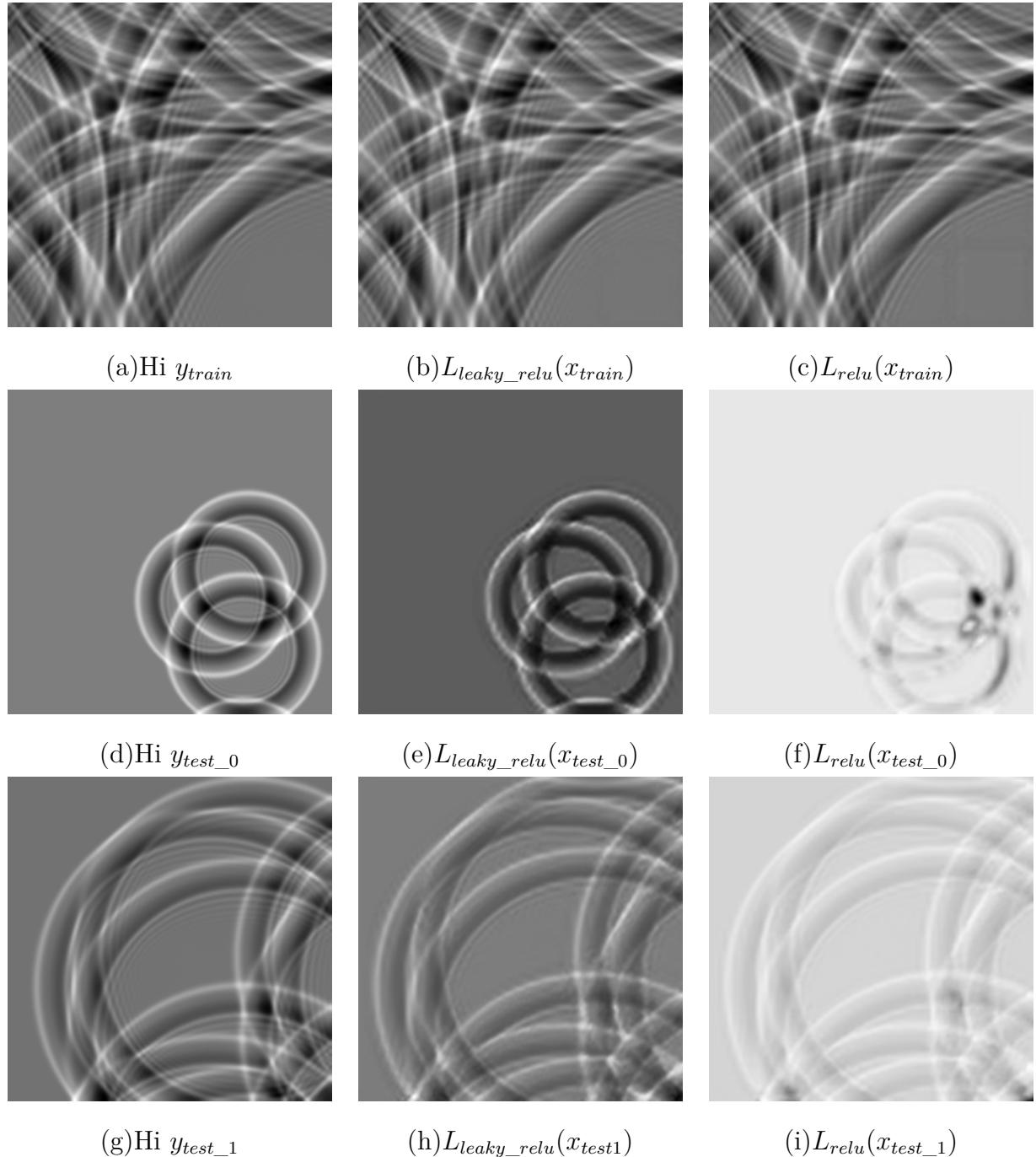


图 5.8 相比较 ReLU 作为生成器的激活函数，使用 Leaky ReLU 可以获得更好的结果。比较 (e) 和 (f) 或者比较 (h) 和 (i)，实验结果表明，使用 Leaky ReLU 生成的结果质量好于使用 ReLU 生成结果的质量。

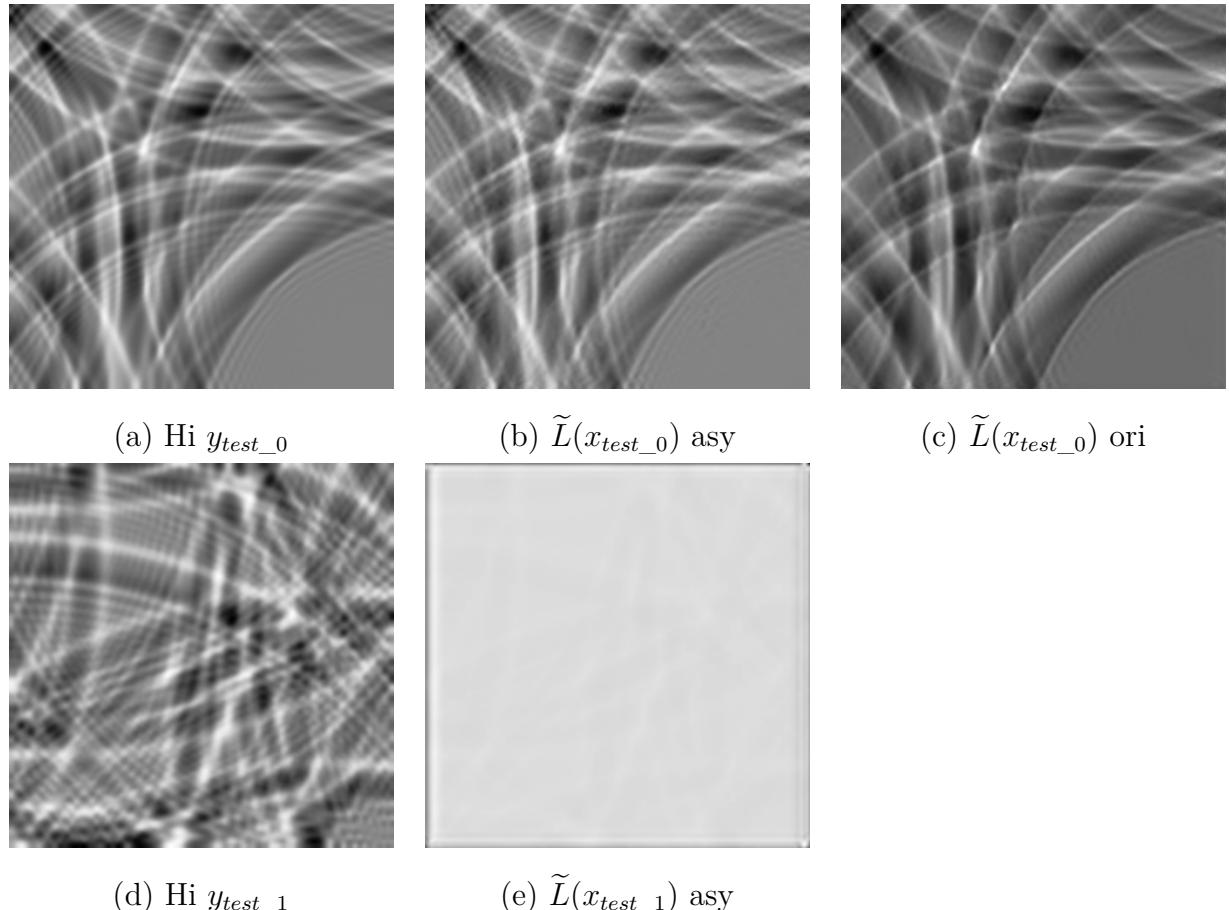


图 5.9 异步的基于全训练集梯度的训练方法可以恢复出更多的高频信息。比较 (b) 和 (c)，可以看出异步的基于全训练集梯度的训练方法可以恢复出更多的高频细节，缓解模糊的问题。但是训练的 GAN 仍然只适用于相似的初始高度的数据的超分辨率，如果明显的提升初始高度，如 (d) 和 (e)，则生成器无法适用。

## 第6章 总结和展望

### 6.1 全文总结

图像或者视频的超分辨率任务是计算机视觉领域的一个重要任务，目前已经存在大量的算法在解决该问题获得了丰硕成果，其中基于深度学习的方法在该问题体现出了明显的优势。但是图形学中的数据的超分辨率问题却未被充分研究。这其中，浅水方程模拟生成的数据是一种很常见的数据，它常用于实时游戏场景中的流体模拟，在低码率的传输环境下需要对原始场景进行降采样压缩传输，因此会损失大量高频细节，有效的浅水方程数据的超分辨率技术能够在减少带宽使用量的同时提高用户体验感，而如何有效地恢复原始数据的高频细节尚待探索。考虑到深度学习在处理超分辨率问题上的优势，本文基于目前的深度学习的算法基础，提出了通过生成对抗网络进行浅水方程模拟数据的超分辨率的算法。

本文提出利用生成对抗网络对浅水方程模拟的数据进行超分辨率。生成对抗网络的结构包括一个基于残差块的生成器和一个基于深度卷积神经网络（CNN）的度量单帧高分辨率高度场恢复质量的空间鉴别器，和一个基于卷积神经网络的度量恢复高度场的时间连续性的时间鉴别器。通过训练生成对抗网络，最终达到在一定的初值条件下，输入低分辨率的下采样的低分辨率数据给生成器，能够恢复出对应的高分辨率高度场，并恢复出许多细小的高频细节。

经过对浅水方程数据超分辨率任务过程中独特性质的思考，考虑到了浅水方程模拟超分辨率重建在时间维度上的光滑性需求，超分辨率任务的旋转等价性需求，模拟数据可能为负值的性质，物理过程中速度信息对神经网络训练有影响等关键因素，本文提出了 1. 引入度量浅水方程模拟时间维度光滑性的时间鉴别器，2. 使用浅水方程速度场信息作为生成器的输入的一部分，3. 使用异步的基于全训练集梯度的训练方法2, 4. 使用旋转对称的数据扩充方法扩充训练集，5. 使用 Leaky ReLU 激活函数作为生成器的激活函数，代替一般采用的 ReLU 激活函数，从而在结果上能通过低分辨降采样数据中推测出较多的高频细节。

## 6.2 展望

本文算法使用的异步的全训练集的训练方法可以恢复出更多的高频细节，但是同时它需要更长的训练时间。针对该问题，未来可以通过增加 GPU 内存或者并行计算每个 mini-batch 的梯度来缓解该问题。针对现在的深度学习框架，未来可以引入分布式的梯度更新的框架，在多张显卡上并行获得 mini-batch 的梯度，再最后加和更新。

针对浅水方程的高度场没有限制导致的泛化性受限的问题，需要寻找更为巧妙的方法，通过设计一些有效的相对量的映射关系处理，避免直接映射绝对高度的超分辨率映射，这样才能使得算法具有良好的泛化性。

另一方面，除了浅水方程的模拟数据之外，图形学中仍然存在着大量的特殊数据的超分辨率问题（或者如第三章介绍的相似问题）未被有效的处理，如点云、三角网格等。通过神经网络构造这些数据的低分辨率到高分辨率的映射是一个非常值得探索的领域，其中对于无时序关系的数据，其关键在于如何拓展图像的卷积操作到这些不规整的定义域上；而对于有时序关系的数据，则更为复杂，还需要考虑时间维度上的连续性的保持。希望本文的方法、实验结果和局限性能够为后来的研究人员提供一些经验和启发。

## 参考文献

- [1] RY Tsai. Multiframe image restoration and registration[J]. Advance Computer Visual and Image Processing, 1984. 1:317–339.
- [2] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang. Learning a deep convolutional network for image super-resolution[C]//European conference on computer vision. Springer, 2014:184–199.
- [3] Chao Dong, Chen Change Loy, Xiaoou Tang. Accelerating the super-resolution convolutional neural network[C]//European Conference on Computer Vision. Springer, 2016:391–407.
- [4] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:1874–1883.
- [5] Jiwon Kim, Jung Kwon Lee, Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:1637–1645.
- [6] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate superresolution[C]//IEEE Conference on Computer Vision and Pattern Recognition. volume 2. 2017:5.
- [7] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017:4681–4690.
- [8] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition[J]. Neural Networks, 1988. 1(2):119–130.
- [9] Yann Lecun, Leon Bottou, Yoshua Bengio, Patrick Haffner. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998. 86(11):2278–2324.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:770–778.

- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative adversarial nets[C]//Advances in neural information processing systems. 2014:2672–2680.
- [12] Yijun Li, Sifei Liu, Jimei Yang, Ming-Hsuan Yang. Generative face completion[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:3911–3919.
- [13] Robinson Macwan, Nehal Patel, Priteshkumar Prajapati, Jaimin Chavda. A survey on various techniques of super resolution imaging[J]. International Journal of Computer Applications, 2014. 90(1).
- [14] Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, Hiroshi Ishikawa. Learning to simplify: fully convolutional networks for rough sketch cleanup[J]. ACM Transactions on Graphics (TOG), 2016. 35(4):121.
- [15] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, Alexei A Efros. Real-time user-guided image colorization with learned deep priors[J]. arXiv preprint arXiv:1705.02999, 2017.
- [16] Steve Bakó, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony DeRose, Fabrice Rousselle. Kernel-predicting convolutional networks for denoising monte carlo renderings[J]. ACM Trans. Graph, 2017. 36(4):97.
- [17] Kan Guo, Dongqing Zou, Xiaowu Chen. 3d mesh labeling via deep convolutional neural networks[J]. ACM Transactions on Graphics (TOG), 2015. 35(1):3.
- [18] Heli Ben-Hamu, Haggai Maron, Itay Kezurer, Gal Avineri, Yaron Lipman. Multi-chart generative surface modeling[C]//SIGGRAPH Asia 2018 Technical Papers. ACM, 2018:215.
- [19] Xue Bin Peng, Glen Berseth, KangKang Yin, Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning[J]. ACM Transactions on Graphics (TOG), 2017. 36(4):41.
- [20] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, Ken Perlin. Accelerating eulerian fluid simulation with convolutional networks[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017:3424–3433.
- [21] You Xie, Erik Franz, Mengyu Chu, Nils Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow[J]. ACM Transactions on Graphics (TOG), 2018. 37(4):95.
- [22] Hassan Aftab, Atif Bin Mansoor, Muhammad Asim. A new single image interpolation technique for super resolution[C]//Multitopic Conference, 2008. INMIC 2008. IEEE International. IEEE, 2008:592–596.
- [23] Michal Irani, Shmuel Peleg. Super resolution from image sequences[C]//[1990] Proceedings. 10th International Conference on Pattern Recognition. volume 2. IEEE, 1990:115–120.

- [24] Michal Irani, Shmuel Peleg. Improving resolution by image registration[J]. CVGIP: Graphical models and image processing, 1991. 53(3):231–239.
- [25] Richard R Schultz, Robert L Stevenson. Improved definition video frame enhancement[C]//1995 International Conference on Acoustics, Speech, and Signal Processing. volume 4. IEEE, 1995:2169–2172.
- [26] Richard R Schultz, Robert L Stevenson. Extraction of high-resolution frames from video sequences[J]. IEEE transactions on image processing, 1996. 5(6):996–1011.
- [27] Henry Stark, Peyma Oskoui. High-resolution image recovery from image-plane arrays, using convex projections[J]. JOSA A, 1989. 6(11):1715–1726.
- [28] Nhat Nguyen, Peyman Milanfar, Gene Golub. Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement[J]. IEEE Transactions on image processing, 2001. 10(9):1299–1308.
- [29] William T Freeman, Thouis R Jones, Egon C Pasztor. Example-based super-resolution[J]. IEEE Computer graphics and Applications, 2002. 22(2):56–65.
- [30] William T Freeman, Egon C Pasztor, Owen T Carmichael. Learning low-level vision[J]. International journal of computer vision, 2000. 40(1):25–47.
- [31] Hong Chang, Dit-Yan Yeung, Yimin Xiong. Super-resolution through neighbor embedding[C]//Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. volume 1. IEEE, 2004:I–I.
- [32] Karl S Ni, Sanjeev Kumar, Nuno Vasconcelos, Truong Q Nguyen. Single image superresolution based on support vector regression[C]//2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings. volume 2. IEEE, 2006:II–II.
- [33] Jianchao Yang, John Wright, Thomas S Huang, Yi Ma. Image super-resolution via sparse representation[J]. IEEE transactions on image processing, 2010. 19(11):2861–2873.
- [34] Ying Tai, Jian Yang, Xiaoming Liu. Image super-resolution via deep recursive residual network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. volume 1. 2017:5.
- [35] Azzouz Hamdi-Cherif, Julie Digne, Raphaëlle Chaine. Super-resolution of point set surfaces using local similarities[J]. Computer Graphics Forum, 2018. 37(1):60–70.
- [36] Oscar Argudo, Antoni Chica, Carlos Andujar. Terrain super-resolution through aerial imagery and fully convolutional networks[J]. Computer Graphics Forum, 2018. 37(2):101–110.
- [37] Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012:1097–1105.
- [38] Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image

- recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015:1–9.
- [40] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation[J]. arXiv preprint arXiv:1606.02147, 2016.
- [41] Alec Radford, Luke Metz, Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. international conference on learning representations, 2016.
- [42] Mehdi Mirza, Simon Osindero. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.
- [43] Martin Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein generative adversarial networks[C]//International Conference on Machine Learning. 2017:214–223.
- [44] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, Stephen Paul Smolley. Least squares generative adversarial networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017:2794–2802.
- [45] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017:2223–2232.
- [46] Leon A Gatys, Alexander S Ecker, Matthias Bethge. A neural algorithm of artistic style[J]. Nature Communications, 2015.
- [47] Jingwen Chen, Jiawei Chen, Hongyang Chao, Ming Yang. Image blind denoising with generative adversarial network based noise modeling[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:3155–3164.
- [48] Jos Stam. Stable fluids.[C]//Siggraph. volume 99. 1999:121–128.
- [49] Nils Thürey, Ulrich Rüde, Marc Stamminger. Animation of open water phenomena with coupled shallow water and free surface simulations[C]//Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2006:157–164.
- [50] Matthias Müller, Jos Stam, Doug James, Nils Thürey. Real time physics: class notes[C]//ACM SIGGRAPH 2008 classes. ACM, 2008:88.
- [51] Hang Zhao, Orazio Gallo, Iuri Frosio, Jan Kautz. Loss functions for neural networks for image processing[J]. arXiv preprint arXiv:1511.08861, 2015.

- [52] David Berthelot, Thomas Schumm, Luke Metz. Began: Boundary equilibrium generative adversarial networks[J]. arXiv preprint arXiv:1703.10717, 2017.

## 攻读硕士学位期间主要的研究成果

1. Jiong Chen, Hujun Bao, **Tianyu Wang**, Mathieu Desbrun, Jin Huang. Numerical Coarsening using Discontinuous Shape Functions[J]. ACM Trans. Graph. (Proc. SIGGRAPH 2018), July 2018. 37(4), 92:1-92:15

## 致 谢

时间过得很快，转眼间我在浙大的硕士研究生学习历程也快告一段落。回想从本科毕业到如今已经过去了三个春秋，对图形学这个领域的理解也增进了一些，对未来的职业规划也清晰了一些。感觉这段旅程中自己不仅学到了丰富的知识，还锻炼了自己多方面的能力，从数学到 coding，从发现问题到试图解决问题，从实验到写作，从和老师同学的合作到日常的社交。真心觉得，这段旅程重新塑造了我，感谢其间遇到的所有人和事。

首先我要感谢我的导师黄劲教授。犹记得当时大四暑假，黄老师让我去实现两篇 SIGGRAPH 的论文作为初步训练，他的细致灵活且有条理的做事风格就给我留下了深刻印象。他对科学的严谨和执着，对这个领域未来发展的见解，对学生的严格要求和悉心指导，对待人生的真知灼见和达观豁达，这些无不一点一滴地影响着我，对我职业生涯的规划乃至对待自己人生的态度都有着莫大的影响。非常有幸能在浙大遇到黄老师，非常感谢您的帮助和教诲。

同时我要感谢实验室的小伙伴们。和你们在一起的这段时光很开心，无论是工作上还是生活上，你们都给予了我莫大的帮助和鼓励。感谢贾颜铭、钟俊成、王煦涛、王恺怿、石泽云、赵崇遥、蒋静、高宏玉这些实验室师兄（弟）师姐的帮助。特别感谢方贤忠师兄在科研上的帮助，陈炯师兄在科研上的悉心指导和讨论，王诗怡、陈俊熠、郑一村、林敏良、郑驭聪这些实验室小伙伴在生活和工作上对我的帮助。感谢你们。

我还要感谢在杭州或者远方的朋友们。特别感谢我的朋友汪令野，从成都到杭州，你我认识也快七年了，感谢你在生活和工作上曾今以及往后的支持、开导和帮助。感谢芳姐、伍源培、李旻辰、杨子辰等小伙伴们给过的建议和帮助。感谢张婷。

最后，我要感谢我的父母和至亲，感谢你们对我数十年如一日的支持和无微不至的照料。感恩。

王天宇

二零一九年六月于求是园