*计算机图形学大作业报告*

基于深度学习的实时全局光照渲染

何浩然, 张子殷, 卿云帆

摘 要: 基于光学模型的光线追踪渲染方法被广泛应用于生成照片级的真实图像，但由于其速度较慢，往往需要离线长时间的渲染，很难实时得到渲染结果。一些现有方法通过减少光路采样以得到实时渲染，但往往会较大地损害图像质量。为解决此问题，本小组在现有研究的基础上，对当前的预测全局光照神经网络模型进行了改进，在对抗式生成网络框架下使用双向卷积神经网络作为基本结构，引入多尺度结构相似度损失函数，获得了能够在合理时间预算内取得更高质量图像效果的新模型。相比同类神经网络方法，本小组避免了过深的网络结构，在网络的复杂度和图像质量之间取得了较好的平衡性，并保证了模型对连续图像生成的鲁棒性，提高了模型精度和运行速率，实现了从直接照明图像到全局光照图像的快速转变。分析测试显示，与相近训练时间内获得的双向卷积神经网络及对抗式生成网络相比，本组模型输出的图像具有明显视觉优越性。而在相同场景上与现有离线及实时渲染引擎进行的渲染时间比较也显示，本组模型可以达到相当理想的实时全局光照预测效果。

Approximating Dynamic Global Illumination with Deep Learning Network

Haoran He, Ziyin Zhang, Yunfan Qing

**Abstract**: Physically based ray-tracing rendering technique has been extensively applied in both industry and academia to calculate the effect of global illumination, as an integral step to generate photo-realistic images. This method, however, can hardly achieve an interactive rendering rate due to large computational overhead. Most of the current trend in the literature is to achieve real-time rendering through sparse sampling, but this often results in reduced image quality. Other global illuminating methods, on the other hand, have their respective drawbacks as well: screen-space directional occlusion cannot render global effects, voxel tracing requires a huge amount of storage, while deep neural network cannot achieve real-time rendering either. To tackle this problem, our group modified existing models for predicting global illumination, and propose a novel approach to generate high-quality global illumination by predicting global illumination from direct lighting and scene-specific parameters (G-buffers) at an interactive rate. We integrate bilateral Convolutional Neural Network (BCNN) into the framework of Conditional Generative Adversarial Networks (cGAN), and design a tailored loss function consisting of L1 loss, SSIM loss and MS-SSIM loss to achieve higher performance. Compared with similar networks, our model circumvents deep network structures, achieving an ideal tradeoff between the quality of the output and the rendering speed, producing much more visually pleasing results of global illumination compared with BCNN models and cGAN models trained within roughly the same amount of time. Tests of rendering time also show that our model can achieve an ideal interactive rate no slower than the traditional real-time rendering engines in the global illumination prediction task.

**Key word：Real-Time Rendering; Neural networks; Global illumination; cGAN**

# **简介与意义**/Introduction

## 项目意义和依据/Significance

基于物理模型进行的渲染的方法，广泛地被应用于生成高质量图片的场景中，对工业界和学术界都有着不可估量的影响。目前基于物理模型的渲染非常依赖于光线追踪，但一个场景中可能包含上千个以上的光源，想要对它们投射出的所有光影进行精确的模拟，需要极高程度的计算量。这不仅提高了硬件门槛，还几乎不可能实现实时交互速率的渲染。因此，如何以实时的交互速率生成高质量的图片一直是学术和工业界十分关注的热门话题。

传统的实时渲染方法通过减少光线追踪过程中的采样率，以及通过对全局光照中部分要素的选择性忽略，以求在可被接受的时间内得到渲染结果，特别是对于较为复杂的模型和场景，图像质量无法得到较好的保证。一些最新的研究也提出了更巧妙的采样方法，通过使用少量精心挑选的动态光源[1]，忽视对全局效果贡献较小的光线，在图像质量上取得了令人满意的渲染结果，但在优化光线追踪的同时，又引入了对光线贡献的实时计算过程，使得整体计算耗时减少并不理想；近年来，随着机器学习和深度学习越来越成为计算机视觉等研究领域的热门课题，在光线追踪上也出现了很多基于神经网络得到的渲染方法。但往往无法在网络结构的复杂性带来的渲染耗时和图像质量之间取得较好的平衡，同时，在泛化性和鲁棒性上也面临很多的挑战。

本小组在辛杭高等研究的基础上[2]，进行了进一步的优化，主要对现有的网络模型进行了改进，在对抗式生成网络框架下，本小组使用双向卷积神经网络作为基本结构，减小了生成器模型复杂度，同时引入了新的损失函数MS-SSIM，极大地提高了生成器的性能。在训练过程中，创新性地引入了Patch分割技术进行数据增强，能够在合理时间预算内取得更高质量的图像效果。

为了提升预测模型的训练效果，本小组利用Blender采集了多个不同特征场景的大量渲染数据，在单场景下，通过不断改变光线、视角和模型位置来生成不同的数据，实现了较好的泛化性能。最后，本小组将训练好的网络与其他离线和实时方法进行比较。相比其它基于神经网络的方法，本方案避免了过深的网络结构，在相同的训练条件下，取得了更好的效果，在网络的复杂度和图像质量之间取得了较好的平衡性，提高了模型精度和运行速率，实现了从直接照明图像到全局光照图像的快速转变。

## 本方法/系统框架/Article Structure



图 1‑1 系统流程图

图 1‑1展示了本小组在进行本项目过程中的工作流程。在学习相关基础知识并且阅读与理解指定参考文献的基础上，本组先使用Blender收集数据。将数据分为测试集、训练集和交叉验证集，作为输入训练参考论文给定的模型，对其进行复现。根据复现结果，本小组整合其他相关的研究，对模型做出改进与创新，并将实验结果与其它相关研究进行对比评价。

# **相关工作**/Related Works

**实时渲染。**实时渲染一直都是计算机图形学关注的重点。Jesu´ s Gumbau等提出了基于随机修剪的LoD模型[3]，在对复杂情形尤其是森林场景的渲染中进行了高效的剪枝；Mark J. Harris和Anselmo Lastra证明了使用简单模型替代复杂模型加以处理[4]，能够得到高质量实时速率的渲染方法，但方法应用较为局限只能对云或类似状态模型进行使用；Peter-Pike Sloan等提出了基于预计算辐射率的方法[5],通过对低频照明环境的低频入射光的预处理，实时的实现了捕捉软阴影等全局光照特点。

**使用神经网络进行渲染**。神经网络在今年被广泛的使用在3D模型的渲染之中。O. Nalbach等在2017年利用CNN网络实现了以实施速率交互的渲染方法[6],表明了深度学习可以在数据中完整的学习图像渲染与合成，但对图像的全局特性并没有做到很好的处理；M. M. Thomas等提出了基于GAN的网络结构[7]，给出了VXGI-质量的渲染效果并实现了比VXGI更快的速率，但模型泛化能力较差，必须一场景对应一个模型，无法将一个模型广泛的应用于多个场景；Peng Dai等提出了PBR网络[8]，加速了通过网络的渲染性能，运用到图像和视频的生成中能够较快得到具有真实感的图像，但其效果较为局限，过多的关注了室内照明，对其它场景下的渲染问题不够关注；2019年Nvidia使用了深度学习超采样模型[9]，通过神经网络对低分辨率的渲染内容进行实时采样，但实现必须依赖于专有硬件。

**轻量级网络**。使用更加简洁高效的层和网络结构，能够帮助模型在有限时间内取得更好的时间效果。CHEMS-EDDINE HIMEUR等在CNN的基础上引入了新的参数化方法，基于尺度空间矩阵，构建了比CNN更轻量级更高性能的3D点云边缘检测的PCEDNet网络[10]；国防科技大学郭裕兰老师等提出了LightNet[11],利用轻量级网络实时的进行图像识别。尽管上述工作无法直接使用在本项目的渲染中，但利用轻量级网络取得实时化效果的想法让本小组获益颇深。

**时序连续性**。时序连续性在非静态图像渲染中是必须要关注的问题。Daniel Scherzer等研究者提出了基于时间相干性的方法[12]，给出了一种通用方法，使用图像空间重投影，在相邻帧之间重用了信息，在确保时序上的连续的同时，也极大的减少了渲染流程需要的计算资源。

# **研究内容与方法**/Contnts and Methods

## 前期准备

在课程学习的图形学知识基础上，对现有工业渲染引擎做了广泛的了解，并最终在老师和助教建议下选择使用Blender进行数据集的采集和制作。Blender作为成熟的开源渲染软件，同时提供了友好的GUI界面和专业的API接口，可以支持用户编写Python脚本实现各种目的图像渲染，为本小组数据集的制作提供了极大的便利，同时极其丰富的社区支持也对初学者更加友好。

由于本小组成员之前并未接触过深度学习相关工作，因此在学习相关基础知识和理解各类网络模型上投入了大量时间和精力。本小组对老师推荐的两篇相关论文进行了详细的学习和解读。

M. M. Thomas在研究中创造性地引入对抗式神经网络cGAN作为预测模型的主要框架[7]，使用含8层卷积层的UNet网络作为生成器，并使用4层卷积层作为判别器来解码以输出二进制值0或1，以判别输入是生成器生成的还是真实图像。

而辛杭高等将计算机视觉研究领域中的双向方法加入到卷积网络，提出了BCNN这样的轻量级网络来减少计算量，具体使用UNet网络，具有3层编码器和3层解码器。他们使用深度、法向量和直接光照作为模型输入，输出间接光照，最后进行上采样并做反走样处理，结合漫反射率信息后最终得到全局光照。具体流程框架可见图 3‑3[2]。该篇论文还引入了时序损失函数，来保持相邻帧之间的连续性。总的来说，使用BCNN的模型框架，在训练条件足够的情况下，可以实现图片和视频渲染的良好效果。

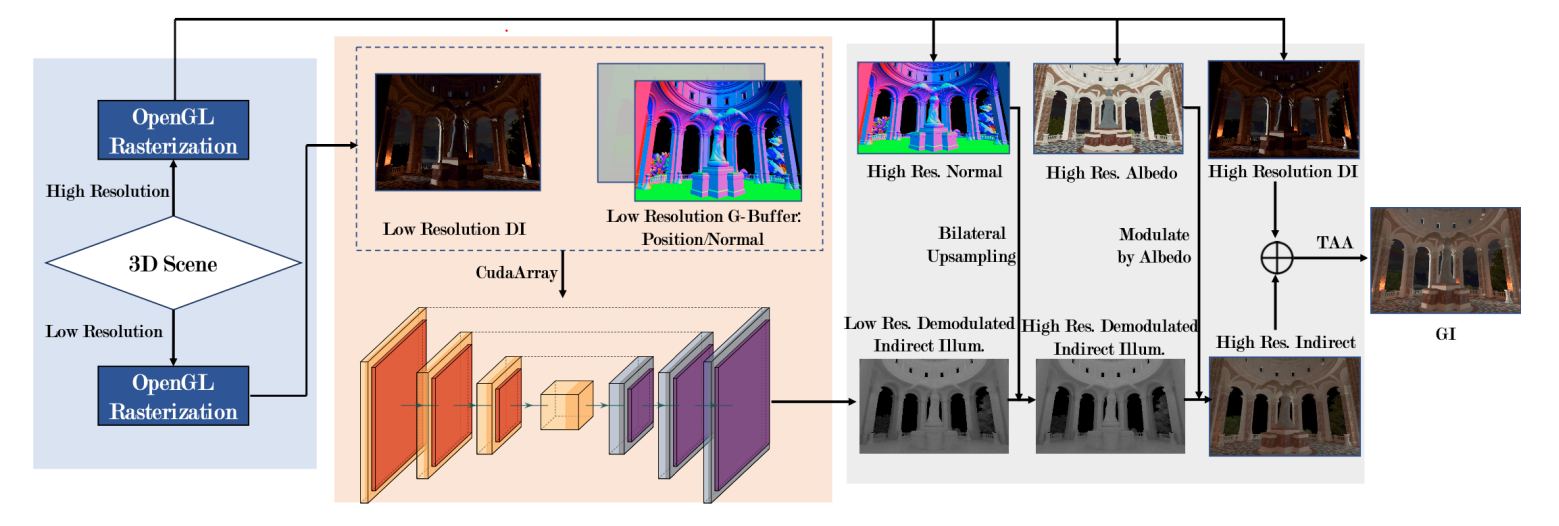


图 3‑3 BCNN模型渲染流程图

在此基础上，本组广泛地调研了相关领域的最新研究成果，形成了对实时渲染较为深入的理解。

## 数据采集

由于多数全局光照预测参考文献提出的模型，以及本组所提出的全局光照预测模型均以场景G-Buffer作为输入，本组在如何获取此数据上做了较多工作。G-Buffer包含3D场景的位置、法线、材质等几何光照信息。在综合比较了底层OpenGL、Unity、UE4等备选方案后，本小组决定使用Blender，从一些开源的3D模型[13]中导出法向量、3D位置信息、光照以及材质信息作为数据来源，如图 3‑4所示。

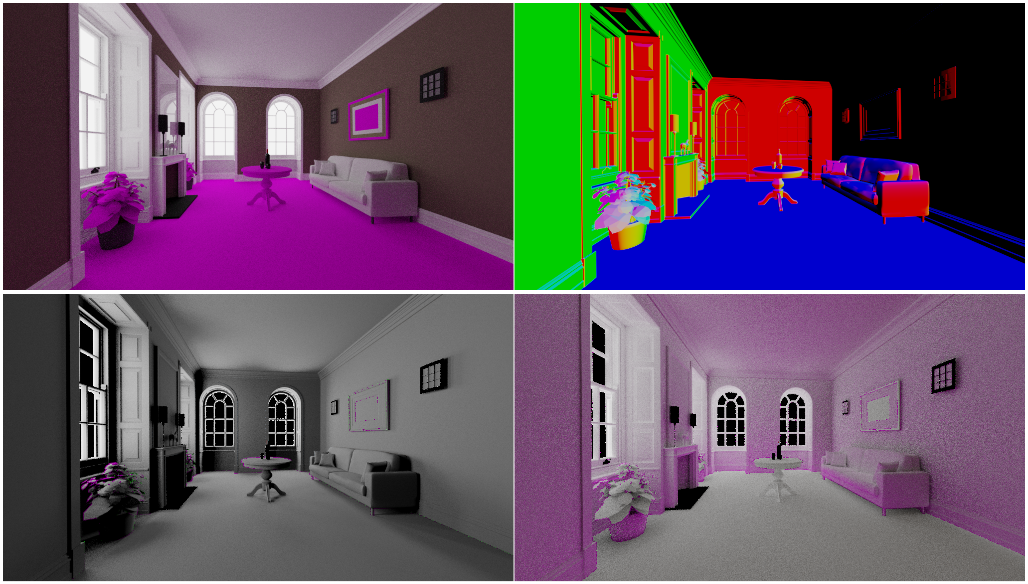


图 3‑4 同一画面（左上）中的法向量（右上）、直接光照（左下）、间接光照（右下）

本组首先对用于制作数据集的模型进行了初步筛选，从经典开源图形学模型中选出了相对较为封闭且没有大面积光泽表面的场景。在同一场景中，本小组通过随机移动或旋转相机、光源，在不同的相机视角、光源位置下进行采样，以保证数据的多样性，确保模型泛化能力。在三周时间内，本组在两台电脑上用Blender的经典光追引擎渲染了超过八千幅、共约120GB的EXR格式原始图像。其中，由于论文复现及模型训练过程中除G-buffer外，还需用到场景的漫反射率信息（albedo），而Blender又无法在同一文件中同时导出G-buffer与漫反射率，故本组制作的数据集中每个样本由两个文件组成。完成数据集的制作后，本小组编写脚本对原始数据进行了人工筛选，去除部分“噪点”数据，主要包括过于空旷或灰暗场景，以及部分相机随机移动到物体内部导致视角被遮挡的情况，最终实际获得有效样本约三千个，如图 3-5所示。同时，由于时间及算力有限，本小组采用了较低分辨率的图像以加快数据集的制作，所有数据样本的分辨率均为1024576，采样频率均为512。



图 3‑5 本组用Blender生成的部分数据

## 相关工作复现

本组首先对M. M. Thomas提出的cGAN预测模型及辛杭高等人提出的BCNN预测模型进行了复现。其中，cGAN模型使用二分类交叉熵（公式1，其中为当前样本在该样本批次中的索引，为该样本的权重）及L1两种损失函数，而BCNN则使用了拉普拉斯损失、L1损失、结构相似度（SSIM）损失及时序损失四种损失函数。其中，拉普拉斯损失是对图像进行了拉普拉斯锐化处理后获得的L1损失；SSIM损失由公式2给出，公式中代表图像或的颜色均值，表示标准差，而是与图像动态范围（即像素可能取值范围）有关的常数，用于维持结果稳定性。在此公式中，第一项是对图像亮度均值的估计，而第二项则可被看作是用方差与协方差对图像对比度、结构相似度的估计。相比L1损失，SSIM更适合于图像处理及图形学领域的任务，在应用于相关神经网络时能获得视觉效果更佳的输出效果。而时序损失则由光流计算，用于保证动态场景（如视频、动画等）中相邻两帧间的画面连续性，与本组工作无关，就不做详细介绍。

在将cGAN模型及BCNN模型进行复现后，本组对它们进行了分析比较。其中cGAN模型在训练时收敛较快，但结果较不理想。而BCNN相比cGAN收敛速度较慢，由于本组所掌握的算力有限，未能获得收敛后的模型。但是，BCNN最大的优势之一是灵活轻量：cGAN包括八层网络，模型大小超过600MB，虽训练较快，但使用时加载较慢，而BCNN的网络只有三层，模型大小不超过500KB。

两篇论文的复现效果不尽如人意，可能是由于本组使用的数据分辨率相比原文较小（1024576），且数据集相对单一，同一场景产生的不同样本以变换相机视角为主。另外，由于本组在处理数据的过程中借鉴了计算机视觉领域的图像分割技术做了数据增强，但并没有控制变量研究其对模型训练的影响，可能也会在一定程度上影响最终效果。具体细节分析将在**实验结果与分析模块**给出。

## 预测模型设计

本组在设计模型时，综合了两篇参考文献所用模型各自的优势，考虑到生成对抗式神经网络在预测间接光照上有良好的表现效果，仍然使用对抗式方法作为总体框架。同时，为了实现网络的轻量性，本小组又沿用了双向卷积神经网络结构作为生成器的基本单元，并在网络层上作了进一步优化和调整。

虽然cGAN模型的网络结构对于损失函数具有较好的容错性，但其深达八层的网络结构不仅使网络显得颇为臃肿笨拙，同时不能高效地利用全局光照预测任务中直接光照信息与全局光照信息间的逻辑相关性，并不能很好的完成对全局光照的预测。

因此，综合考虑复现结果并对此分析讨论后，本小组创新性地提出了Bi-cGAN模型，将原有cGAN模型中生成器部分的深层网络结构替换为了轻量级双向卷积网络，明显缩减了模型规模，提高了全局光照预测性能,同时极大的加速了模型的收敛。图 3‑6展示了本模型的整体网络架构。

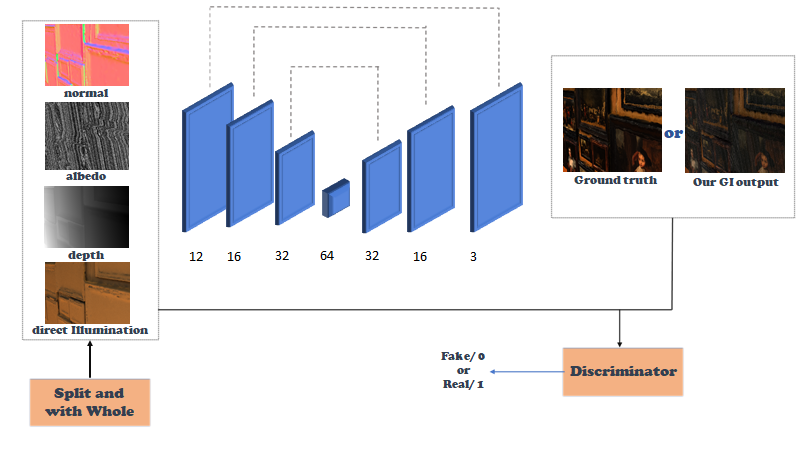


图 3‑6 Bi-cGAN的网络结构

Bi-cGAN以法向量、深度、漫反射率和直接光照作为生成器的输入，分别经过3层双向卷积网络的编码器和解码器后输出预测的全局光照结果。在判别器部分，本小组仍然使用了四层卷积网络作为decoder进行二分类0/1输出，以加速生成器的收敛。在具体实现模型的代码中，本组参考了提出BCNN与cGAN模型的两篇论文的源代码[2][7]。

由于对抗式方法的高性能表现主要来自于生成器的损失函数设计，本小组进一步对模型中的损失函数作了改进以更好地适应于该间接光照预测场景。在阅读2016年Hang Zhao 等关于多种损失函数的比较结果[14]分析的论文后，本小组在使用L1损失与SSIM损失的基础上，还新加入了多尺度结构相似度（MS-SSIM）损失。这里本组使用了MS-SSIM较为成熟的第三方开源库pytorch-msssim[15]。

MS-SSIM定义如公式3所示。相比SSIM，MS-SSIM引入了由原始图像的不同分辨率版本构成的图像金字塔（公式中就代表金字塔层数），并使用，两个参数对图像的亮度均值、方差及结构相似度进行调制，使神经网络对计算过程中使用的高斯滤波器的方差敏感度降低，不再会在滤波器方差过小时在图像中引入噪音而破坏局部结构。

根据由阅读相关文献获得的先验结论以及多次测试结果，本小组最终将三种损失函数的权重定为60%的L1，20%的SSIM，以及20%的MS-SSIM，以获得最优网络预测效果。

## 改进与优化

参考文献提出的对抗生成网络模型，以及前文提及的本小组设计的模型所使用的对抗生成网络框架都属于条件式对抗生成网络。在尝试进一步优化模型的过程中，本组也尝试了对抗生成网络的另一变种——信息式对抗生成网络。二者的区别在于，在训练过程中，条件式对抗生成网络使用传统的监督学习方法，显式地将图片真伪作为标签输入判别器进行学习，使生成器学习如何生成真实数据。而信息式对抗生成网络的判别器使用无监督学习方法，同时在生成器中引入隐含变量，引导其生成指定的目标类型数据。

但是，一方面由于原模型cGAN本身就学习到了数据的标签信息，使用infoGAN在本预测全局光照任务上显得不恰当，另一方面也由于本组的算力、时间有限，不支持进行更多的实验和测试，所以本小组最终放弃使用信息式对抗生成网络。

# **实验结果与分析**/Experiment Results and Analysis

## 模型效果评估

图 4‑1给出了在复现参考文献中的cGAN模型、BCNN模型，以及训练本小组设计的模型共三个神经网络过程中，平均损失函数值随训练代数的变化。

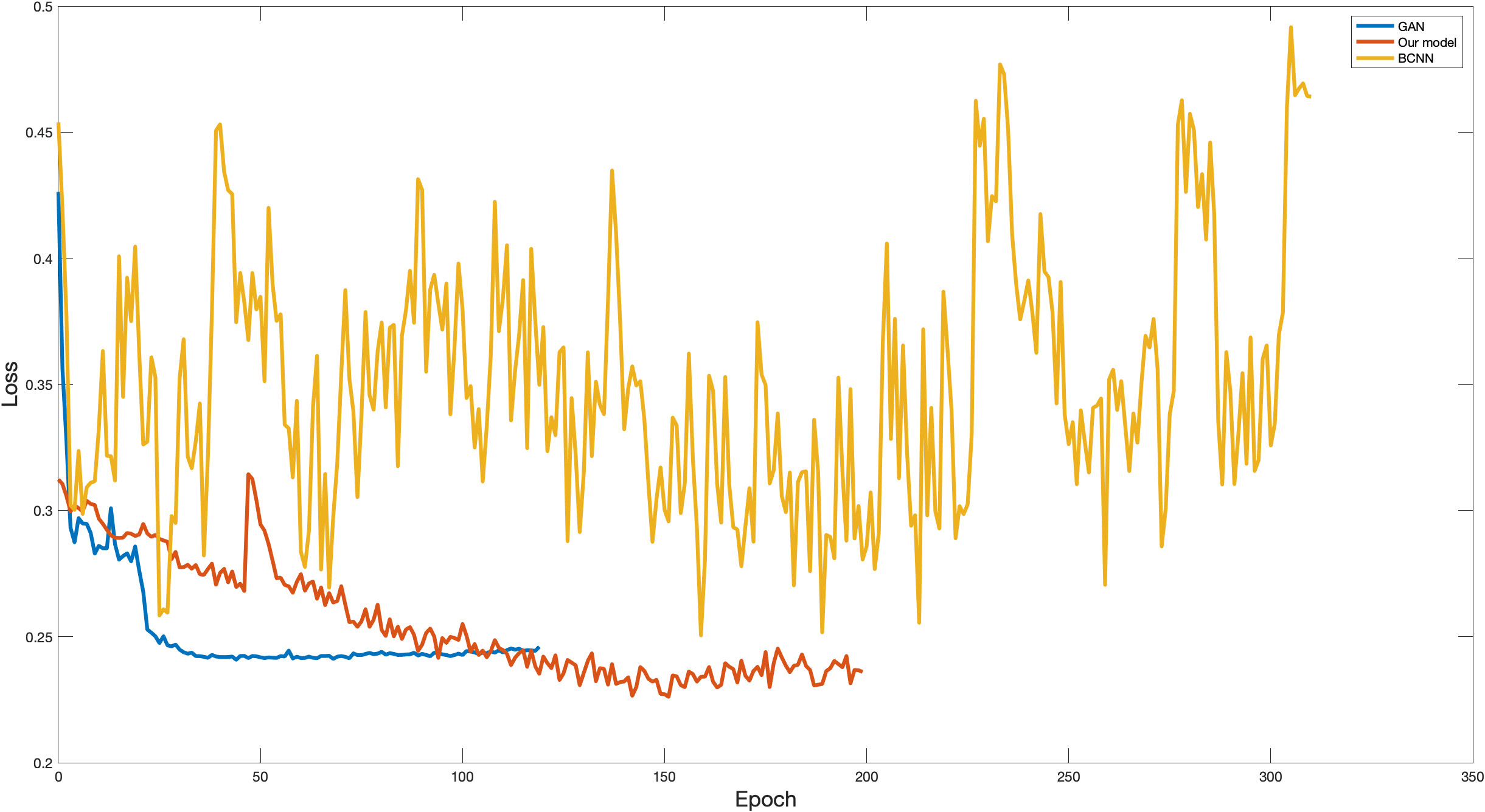


图 4‑1 三个模型的训练损失

通过观察蓝色折线表示的cGAN损失函数随迭代次数的走势图，本小组发现其在第30代左右就能达到收敛，因此在120代时提前终止了训练。本小组认为这一方面是因为GAN模型本身具有加速收敛的作用，另一方面也是因为本小组的训练集较少，不能达到论文中展示的数据集的大小，也会导致模型本身较快的收敛。

而通过观察黄色折线表示的BCNN损失函数随迭代次数的走势图，本小组发现其在整个训练过程中都出现较大的波动，无法在310代之内达到收敛。本小组认为是因为BCNN模型本身对数据和算力的要求比较高，而本小组目前难以实现论文中提到的训练条件：原论文提到，训练过程使用近五万个样本，在4个NVIDIA 1080TI上用超过120小时共训练了一千代，并且在每一次迭代中，每张图片训练30000次；本小组因源数据量，硬件条件以及时间上均较为有限，用近两天半的时间训练了310代，仍然未能使其收敛。

而红色曲线代表本组模型的训练损失。可以看到，虽然本组设计的模型收敛速度比GAN模型略慢，但最终获得了更小的损失，而在测试集的图像输出结果上也有更好的视觉效果。而相比于BCNN，由于本组设计的Bi-cGAN使用了对抗生成式框架，不仅训练收敛所用代数大幅减少，每一代训练所需的时间也明显缩短。

图 4‑2展示了cGAN、BCNN、Bi-cGAN在测试集上做出的预测示例。



图 4‑2 cGAN（右上）、BCNN（左下）、本组模型（右下）预测性能比较

其中左上角为原始图像真值，通过使用Blender提供的Cycles光线追踪渲染引擎获得。对于cGAN模型生成的结果，正如前文所述，由于其深达八层的网络结构过于复杂，不能高效地捕获直接光照信息与全局光照信息间的逻辑相关性，因此在结果上并不能达到一个可以接受的视觉效果，也存在较为明显的摩尔纹。对此，本组尝试了与该篇论文的作者直接进行邮件联系，但截止到本篇报告提交前仍未能得到回复。

而对于擅长于预测光照信息的BCNN模型来说，由于本组在训练该模型过程中使用的数据量与训练时间与原论文相比都较小，且所用的数据分辨率较低，在降采样过程中光照信息有部分损失，此场景又本身较暗，未能获得理想效果。至于本组模型Bi-cGAN的输出，虽然与离线光追算法获得的真值存在一定差距，但图像质量较另两个模型有肉眼可见的明显提升。而由图 4‑3给出的三个模型与真值的损失差距比较也可以看出，本组模型预测结果的SIMM、L1损失都小于BCNN与GAN。

可见，虽然本组模型的性能与辛杭高等人用大量算力得到的完全收敛后的BCNN模型无法抗衡，但对于算力及时间都有限的应用场景，Bi-cGAN明显更适宜使用。因此可以得出结论，本组的设计在性能与代价间做出了良好的平衡。

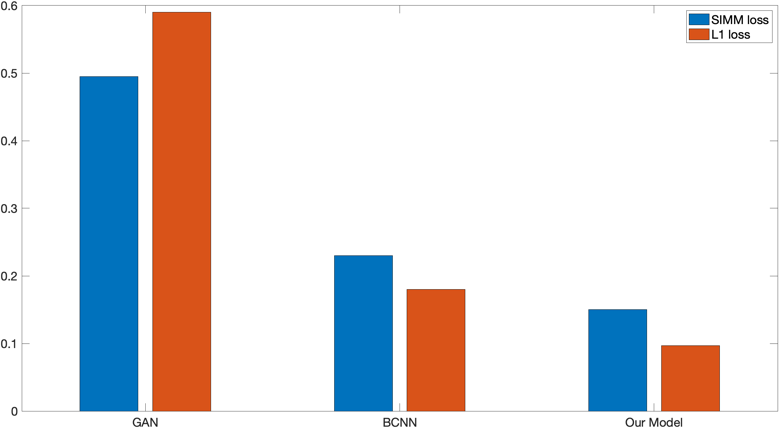


图 4‑3 GAN、BCNN与本组模型在测试中与图像真值的差距

图 4‑4展示了三个模型在一个训练集中未出现过的新场景中进行全局光照预测的表现。由于cGAN模型中一个网络只适用于一个场景，不具泛化能力，在新场景中未能做出正确的光照预测，产生大量网格状的摩尔纹，并且丢失了大量全局光照信息。而BCNN模型即使远没有收敛，仍然成功地在部分细节处（如沙发底下、门框和天花板处）捕捉到了间接光照与阴影信息。相比之下，本组模型Bi-cGAN的泛化能力就远强于cGAN与BCNN。虽然在经过TAA反走样处理与图像动态范围的调整后，画面中存在部分颜色失真的问题（如本应为紫色的地板与左侧的深色墙壁），但整体仍成功实现了全局光照的预测并且获得了可接受的视觉效果。



图 4‑4 三个模型在新场景上的测试（右上：cGAN，左下：BCNN，右下：本组模型）

## 模型性能比较

为测试本组设计的模型性能，本组采用五种不同渲染方式，对同一测试数据样本进行了渲染并比较了它们所用的时间：Blender中的Cycles与Eevee两种引擎、cGAN模型、BCNN模型及本组模型。其中，Cycles引擎使用经典光追算法计算全局光照，Eevee引擎通过光栅化算法近似估计光线与物体材质的相互作用，而后三者都是在Python环境下执行。其结果展现于表 4‑1中。

图 4‑4给出了五种渲染方式的直观性能比较。由于cGAN、BCNN与本组模型都直接通过Python环境读取输入，将图像作为矩阵进行变换、运算，直接获取输出结果，它们的渲染速度都远快于Blender中的两个引擎。在三个神经网络模型内部，本组模型Bi-cGAN与BCNN模型的时间性能几乎相同，而GAN模型由于较为庞大，速度略慢于两者。可见本组模型Bi-cGAN较好地实现了前一节中提出的项目动机：在保留cGAN训练时快速收敛优点的同时，逼近BCNN轻量级网络的优秀性能。相比之下，Blender自带实时渲染引擎的耗时是三个神经网络的百倍之多，而光追算法作为离线渲染算法，渲染一张图像更是需要四十秒之久。

表 4‑1 不同渲染方式耗时比较

|  |  |
| --- | --- |
| 渲染方式 | 耗时（s） |
| Cycles | 40.1 |
| Eevee | 0.9 |
| cGAN | 0.0064 |
| BCNN | 0.0051 |
| Our Model | 0.0052 |

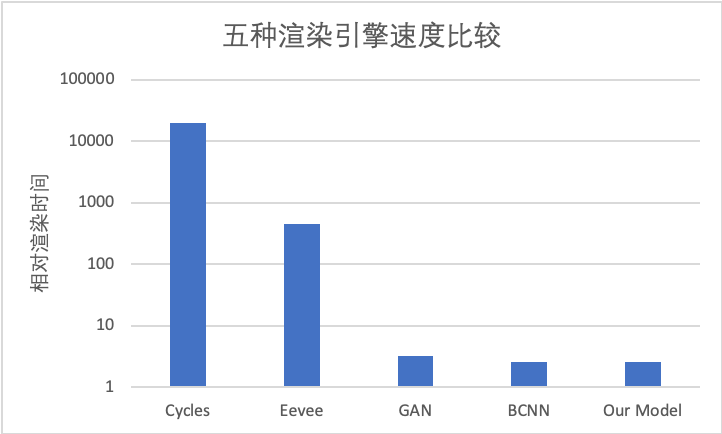


图 4‑5不同渲染方式性能比较

# **特色与创新**/ Distinctive or Innovation Points

本组提出的模型将双向卷积神经网络整合进对抗式生成网络框架中，结合了前者的轻便性与后者的鲁棒性，同时又弥补了前者训练速度慢、后者网络结构过于复杂无法精确捕获间接光照与全局光照间相互关系的缺点。

在此基础上，本组在模型训练过程中又借鉴了计算机视觉领域的图像分割技术[16]，将每一张训练样本的G-buffer、漫反射率及原始图像同步分为16小块，然后将来自不同训练样本的小块进行组合构成新样本，并且保证输入样本中Albedo、Normal、Depth、Direct Illumination和Ground Truth的相对应图像位置来自于同一张数据图，最终将组合得到的新样本与原始样本数据作为输入进行训练，不仅能提高模型对于不同场景的全局光照的学习能力，还在一定程度上缓解了训练数据不足的问题。

在经过大量实验后，本小组还在神经网络的损失函数中引入了多尺度结构相似度损失函数MS-SSIM，以更好地保存输出图像中的细节。使生成器与Ground Truth之间的损失最小化，能够在最大程度上逼近全局光照。具体细节可见3.4节预测模型与设计中。

# **总结**/ Conclusion

本小组在复现cGAN和BCNN的基础上，综合两种基于神经网络的渲染方式的优势，同时引入了其它相关研究，提出了Bi-cGAN的模型，能在相同的硬件条件和训练条件下，得到更快的收敛速度以及更好地图片渲染效果，目前来看取得了训练成本（数据量、训练时间）和模型性能的良好的平衡。同时，本小组也认为在有更充足的数据和硬件条件下，Bi-cGAN的性能能够得到进一步提升，并且其由于引入了对抗式训练加速收敛，能够更快的取得BCNN论文中达到的模型效果

# **致谢**/Acknowledgment

感谢盛斌老师，易冉老师对项目的指导和支持！

感谢舒子曦助教，陈昂助教对项目的帮助和答疑！

感谢辛杭高学长对源码理解提供的帮助！

8、补充说明

8.1、术语：

|  |
| --- |
| cGAN（conditional Generative Adversarial Network，对抗式生成网络）：一种通过让两个神经网络以相互博弈的方式进行学习的非监督式学习方法。  BCNN（Bilateral Convolutional Neural Networks，双向卷积神经网络），一种被新提出的双层卷积神经网络。  VXGI（Voxel Global Illumintaion，立体像素全局光照），一种NVIDIA提出的实时全局光照渲染技术。  GUI（Graphical User Interface，图形用户界面），一种交互式的用户友好的操作界面。  API（Application Programming Interface，应用程序接口），预先约定的软件系统不同组成部分衔接接口。  损失函数：机器学习中用以代替期望风险的经验风险值。  L1损失：机器学习中目标值与估计值的绝对差值之和，一种常用的损失函数。  SSIM（structural similarity index，结构相似度损失）：一种应用于图像处理领域的损失函数，综合考虑图像亮度、对比度、结构相似度等信息。  MS-SSIM（multi-scale structural similarity index，多尺度结构相似度损失）：SSIM的改进版本，弥补了SSIM在其所用的高斯滤波器方差较小时可能给图像引入噪音，破坏局部结构的问题。 |

References:

[1] MOREAU P, CLARBERG P. Importance sampling of many lights on the GPU [M]. Ray tracing gems. Springer. 2019: 255-83.

[2] XIN H, ZHENG S, XU K, et al. Lightweight Bilateral Convolutional Neural Networks for Interactive Single-bounce Diffuse Indirect Illumination [J]. IEEE Transactions on Visualization and Computer Graphics, 2020: 1-.

[3] GUMBAU J, CHOVER M, REMOLAR I, et al. View-dependent pruning for real-time rendering of trees [J]. Computers & Graphics, 2011, 35(2): 364-74.

[4] HARRIS M J, LASTRA A. Real‐time cloud rendering; proceedings of the Computer graphics forum, F, 2001 [C]. Wiley Online Library.

[5] SLOAN P-P, KAUTZ J, SNYDER J. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments; proceedings of the Proceedings of the 29th annual conference on Computer graphics and interactive techniques, F, 2002 [C].

[6] NALBACH O, ARABADZHIYSKA E, MEHTA D, et al. Deep shading: convolutional neural networks for screen space shading; proceedings of the Computer graphics forum, F, 2017 [C]. Wiley Online Library.

[7] THOMAS M M, FORBES A G. Deep illumination: Approximating dynamic global illumination with generative adversarial network [J]. arXiv preprint arXiv:171009834, 2017.

[8] DAI P, LI Z, ZHANG Y, et al. PBR-Net: Imitating Physically Based Rendering Using Deep Neural Network [J]. IEEE Transactions on Image Processing, 2020, 29: 5980-92.

[9] EDELSTEN A, JUKARAINEN P, PATNEY A. Truly next-gen: Adding deep learning to games and graphics; proceedings of the In NVIDIA Sponsored Sessions (Game Developers Conference), F, 2019 [C].

[10] HIMEUR C-E, LEJEMBLE T, PELLEGRINI T, et al. PCEDNet: A Lightweight Neural Network for Fast and Interactive Edge Detection in 3D Point Clouds [J]. ACM Transactions on Graphics (TOG), 2021, 41(1): 1-21.

[11] ZHI S, LIU Y, LI X, et al. LightNet: A Lightweight 3D Convolutional Neural Network for Real-Time 3D Object Recognition; proceedings of the 3DOR@ Eurographics, F, 2017 [C].

[12] SCHERZER D, YANG L, MATTAUSCH O, et al. Temporal coherence methods in real‐time rendering; proceedings of the Computer Graphics Forum, F, 2012 [C]. Wiley Online Library.

[13] MCGUIRE M. Computer Graphics Archive [Z]. 2017

[14] ZHAO H, GALLO O, FROSIO I, et al. Loss functions for image restoration with neural networks [J]. IEEE Transactions on computational imaging, 2016, 3(1): 47-57.

[15] VAINF. pytorch-msssim [Z]. 2021

[16] RAN W, YANG B, MA P, et al. TRNR: Task-Driven Image Rain and Noise Removal with a Few Images Based on Patch Analysis [J]. arXiv preprint arXiv:211201924, 2021.

时间安排与分工统计表

|  |  |  |  |
| --- | --- | --- | --- |
| **组员信息**（含组长） | | | |
| 学生姓名 | 何浩然 | 学 号 | 519020910061 |
| 项目分工 | 设计与训练模型 | | |
| 学生姓名 | 张子殷 | 学 号 | 519021910348 |
| 项目分工 | 制作数据集，撰写项目报告 | | |
| 学生姓名 | 卿云帆 | 学 号 | 516020910073 |
| 项目分工 | 制作数据集，撰写项目报告 | | |
| **时间安排/**  **Schedule** | （如选题、方案制定、试验研究、数据处理、研制开发、撰写总结报告等）(Such as topic selection, program formulation, experimental research, data processing, research and development, writing summary reports, etc.)  第七周：确定选题  第八周：准备期末考  第九周：阅读参考文献  第十周：学习OpenGL等图形学底层接口和模型相关基础知识  第十一周：学习Blender，复现参考论文提供的模型  第十二周：撰写中期报告，学习Blender的Python接口，制作数据集  第十三周：制作数据集，训练复现出的参考文献模型，设计本组模型  第十四周：继续制作数据集，调试、训练本组模型  第十五周：测试比较模型性能，撰写项目报告，制作视频与ppt准备答辩  第十六周：整理项目材料，完善展示视频与工作报告，提交最终成果 | | |