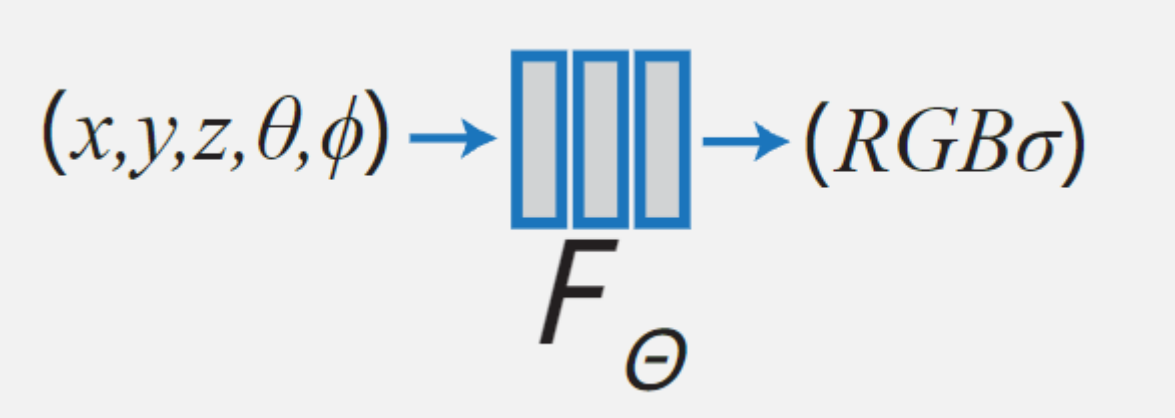
小组汇报NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Abstract

我们提出了一种方法，通过优化基于一组稀疏的输入视角的连续体积场函数，实现了合成复杂场景的新视角的最新成果。我们的算法使用一个**全连接（非卷积）深度网络**来表示场景，其**输入是**一个**连续的5D坐标（空间位置（x；y；z）和视角方向（θ；φ））**，**输出是该空间位置处的体积密度和视角相关的辐射强度。**我们通过**查询相机光线上的5D坐标来合成视角，并使用经典的体积渲染技术将输出的颜色和密度投影到图像中**。由于体积渲染在自然上是可微分的，**优化我们的表示所需的唯一输入是一组具有已知相机姿态的图像**。我们描述了如何有效地优化神经辐射场以呈现具有复杂几何和外观的照片真实的新视角，并展示了在神经渲染和视角合成方面超越以往工作的结果。视角合成的结果最好作为视频来观看，因此我们建议读者观看我们的附加视频以进行令人信服的比较。



Introduction

我们将静态场景表示为一个连续的 5D 函数，它在空间中的每个点 (x, y, z) 输出每个方向（θ, φ）发出的辐射强度。在每个点上具有一个密度，起到差分不透明度的作用，控制通过 (x, y, z) 的光线积累了多少辐射强度。

我们的方法通过优化一个没有卷积层的深度全连接神经网络 (MLP) 来表示这个函数。从单个5D坐标（x；y；z；θ；φ）回归到单个体积密度和视角相关的RGB颜色。为了

从特定视点渲染这个神经辐射场（NeRF），我们采取以下步骤：

1. 沿着相机光线穿过场景生成一组采样的 3D 点
2. 将这些点及其对应的 2D 视角方向作为输入传递给神经网络，生成一组输出的颜色和密度
3. 使用经典的体积渲染技术将这些颜色和密度积累成一张2D图像

由于这个过程在自然上是可微分的，我们可以使用梯度下降来通过最小化每个观察到的图像与从我们的表示渲染的相应视角之间的误差来优化这个模型。通过在多个视角上最小化这个误差，鼓励网络对场景进行一致的建模，将高体积密度和准确的颜色分配给包含真实场景内容的位置。

我们发现，对于复杂场景，基本的神经辐射场表示的优化实现不能收敛到足够高分辨率的表示，并且在每个相机光线所需的采样数方面效率低下。为了解决这些问题，我们通过使用位置编码对输入的5D坐标进行转换，使MLP能够表示更高频率的函数，并提出了一种分层采样过程来减少对高频率场景表示进行充分采样所需的查询数量。我们的方法继承了体积表示的优点：两者都能表示复杂的真实世界几何和外观，并且非常适合使用投影图像进行基于梯度的优化。重要的是，我们的方法克服了在高分辨率下对离散化体素网格进行建模时的存储成本问题。

Contribution:

- 使用基本的MLP网络将具有复杂几何和材质的连续场景表示为5D神经辐射场的方法。
- 基于经典的体积渲染技术的可微分渲染过程，我们使用该过程从标准RGB图像中优化这些表示。这包括一种分层采样策略，将MLP的容量分配给具有可见场景内容的空间。
- 一种位置编码，将每个输入的5D坐标映射到更高维度的空间，这使我们能够成功地优化神经辐射场以表示高频率的场景内容。

Neural Radiance Field Scene Representation - 如何用 NeRF 来表示 3D 场景？

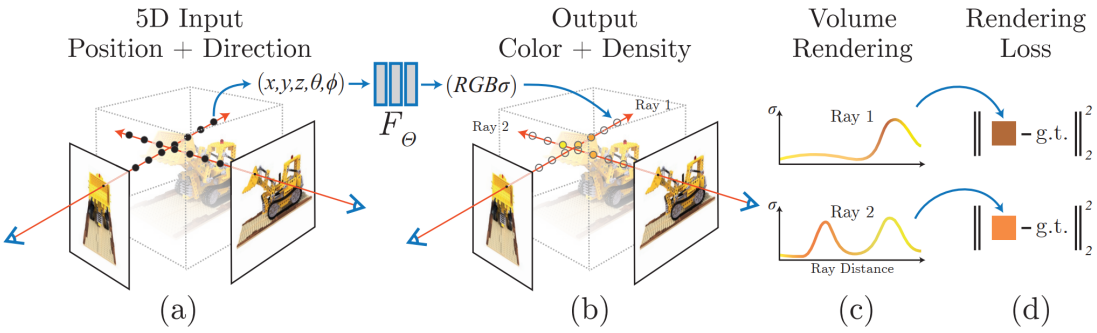
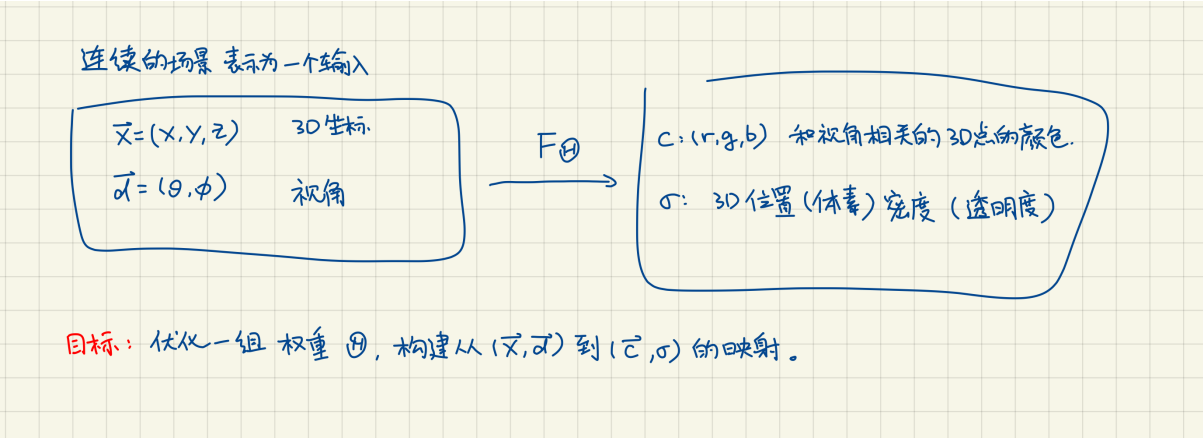


图2：我们的神经辐射场场景表示和可微分渲染过程的概述。我们通过沿相机光线采样 5D 坐标（位置和视角方向）来合成图像（a），将这些位置输入MLP以生成颜色和体积密度（b），并使用体积渲染技术将这些值组合成一幅图像（c）。这个渲染函数是可微分的，因此我们可以通过最小化合成图像和实际观察到的图像之间的残差来优化我们的场景表示（d）。



所以这中间的函数 F，也就是图当中的神经网络，就是用来表示 3D 场景的。 这种表示方法是**隐式的**

> **显式**：直接描述三维空间的几何结构和表面属性。比如，可以用一个三维网格来描述一个物体的形状，网格的顶点表示形状的特征点，顶点之间的连线构成了物体的表面。常见的显式表示方式还包括点云（point cloud）、体素（voxel）等。
>
> **隐式**：间接的表示方法。它通过描述一个函数，来间接地表示出三维空间的物体或者场景。例如，我们可以定义一个距离函数，它的值表示空间中任意一点到物体表面的距离，那么物体的表面就可以表示为所有使得这个距离函数等于 0 的点的集合。

Volume Rendering with Radiance Fields - 如何基于NeRF渲染出2D图像？

我们的 5D 神经辐射场将场景表示为空间中任意点的 **体积密度** 和 **方向性辐射强度**。

经典的体积渲染方式

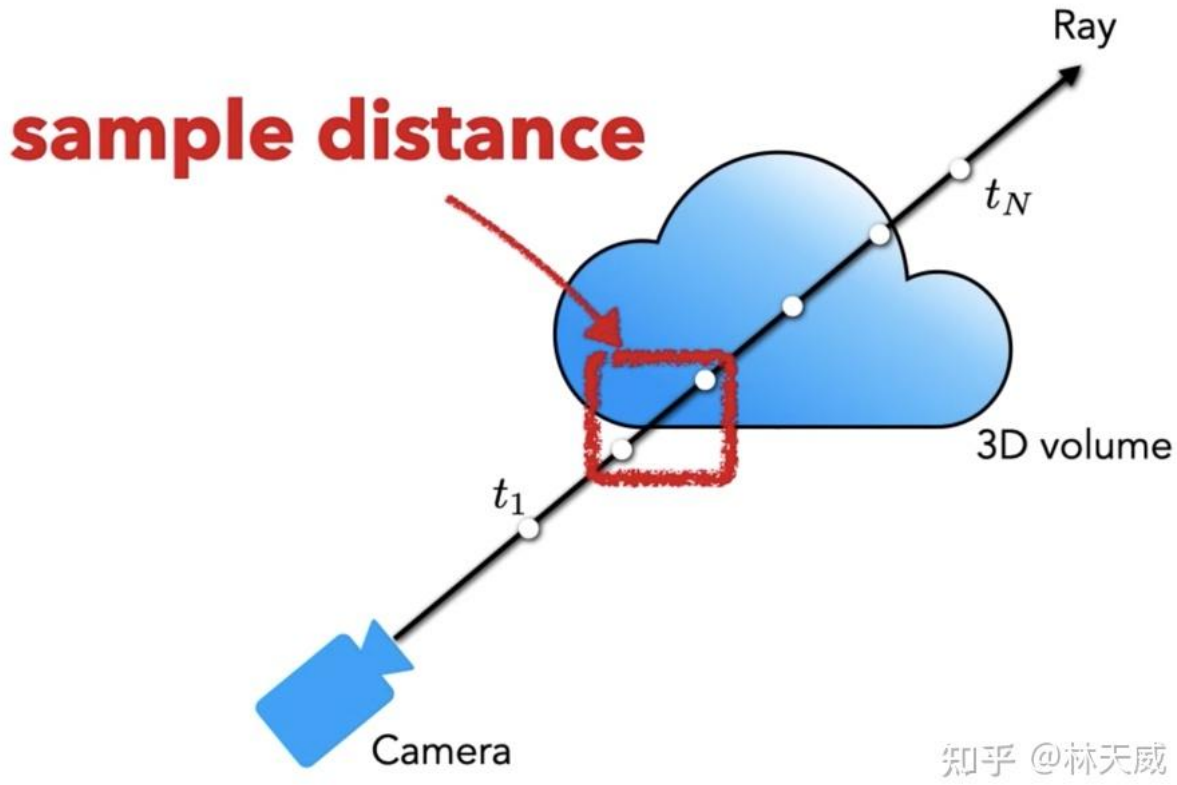
$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right).$$

- *C*(*r*) 代表的是光线 *r* 穿过一个三维空间时的颜色。这条射线 *r* 可以理解为摄像机视点通过图像的某一像素与三维场景的交点形成的光线。换句话说，每一条射线 *r* 对应于渲染图像的一个像素点。
- *r*(*t*) = *o* + *td* 是表示从摄像头出发的光线路径的参数化函数，其中 *t* 是从摄像头出发的距离。*d* 是相机射线角度
- *σ*(*r*(*t*)) 是在光线路径上位置 *r*(*t*) 的体积密度。某个点的体积密度（或者说密度）在NeRF中描述的是空间中某一点的“物质存在量”，你可以将它类比为三维环境中的不透明度。体积密度越大，说明该点处存在更多的物质，光线更可能在这里被吸收或散射，所以体积密度与点的不透明度正相关。
- *c*(*r*(*t*),*d*) 是位置 *r*(*t*) 在光线方向 *d* 的辐射强度。
- *T*(*t*) 是从摄像头到 *t* 位置的透射函数，它表示的是光线从摄像头到该位置的路径上受到的衰减。

基于分段随机采样的离散近似 volume rendering 方式

动机：

1. **离散近似：**上面的公式是连续的，实际计算当中无法进行真正的连续积分，因此需要一种离散的近似方法。虽然采样是离散的，但是 MLP 的 evaluation 在整个优化过程当中，是位置上连续的。
2. **分段随机采样：**如果我们选择在光线路径上均匀地进行采样，虽然可以在一定程度上近似这个积分，但这种方法在处理密度较高的区域或者快速变化的区域时效果并不好。因为这些区域可能需要更高的采样密度来准确地估计积分。



首先将射线需要积分的区域分为N份，然后在每一个小区域中进行均匀随机采样。这样的方式能够在只采样离散点的前提下，保证采样位置的连续性。

在这种策略下，第 i 个采样点可以表示为：

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right].$$

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,,\ \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right),$$

Optimizing a Neural Radiance Field: 如何训练 NeRF

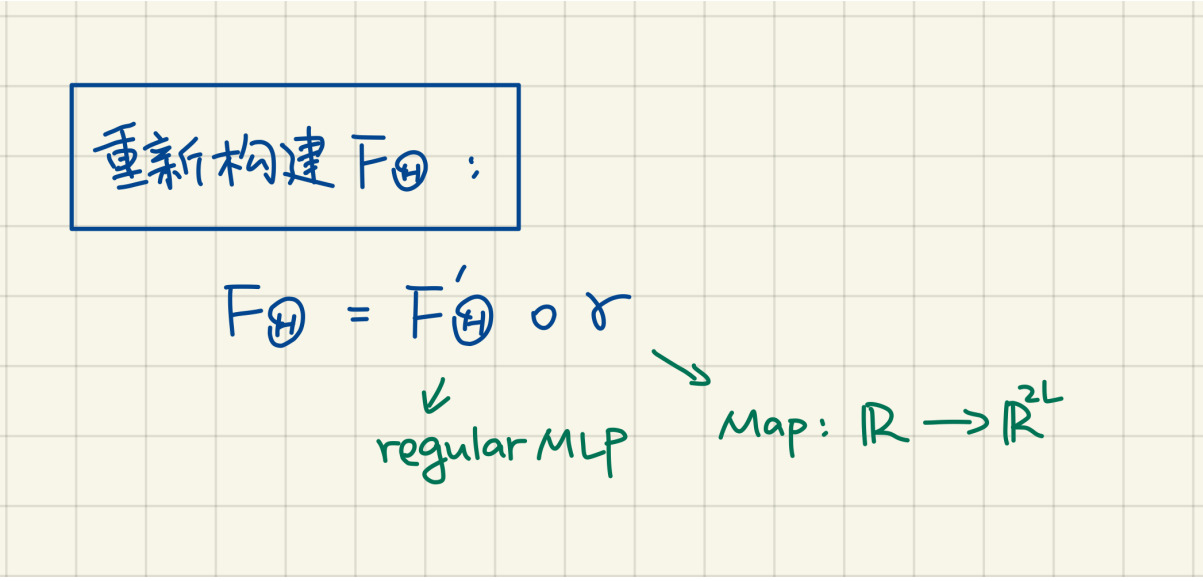
1. Positional Encoding

动机：

对神经网络 F 直接输入 [x, y, z, θ, φ] 会导致渲染效果较差，无法很好地表示颜色和几何中的高频变化。

> **这与Rahaman等人最近的研究结果一致：**
> 1. 深度网络对学习较低频率函数有偏向性
> 2. 将输入通过高频函数映射到更高维度空间，然后再输入神经网络，可以更好地拟合包含高频变化的数据。

方法：



$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right).$$

这个函数 γ(·) 分别应用于x的三个坐标值（这些值被归一化到 [-1, 1] 范围内）和笛卡尔视角方向单位向量 d 的三个分量（根据构造，这些分量位于 [-1, 1] 范围内）

使用这种 Position Encoding 的理由

"Positional Encoding"是一种广泛应用在各类深度学习模型中的技术，尤其在处理序列数据（例如文本和时间序列）或者空间数据（例如图像和3D模型）时非常有用。其目的是为了给模型提供关于输入数据中的顺序或者位置信息。

这个方法的基本思想是将原始的输入通过一组不同频率的正弦和余弦函数进行编码，得到一组新的值。这些函数可以**将原始输入的不同尺度的信息映射到不同的维度**。

💬 举个例子，如果我们使用频率为1和2的正弦函数对一个周期性的信号进行编码，那么频率为1的正弦函数能够捕捉到信号的大尺度（或者说低频）的特性，而频率为2的正弦函数则能够捕捉到信号的小尺度（或者说高频）的特性。这样，通过将编码后的值作为神经网络的输入，神经网络就可以同时理解和利用信号的不同尺度的特性了。

2. Hierarchical Volume Sampling

动机：

NeRF 的渲染过程需要在每条光线上采样一系列的3D点，对于这些点，NeRF 会使用神经网络预测其体积密度和颜色。

问题在于NeRF的渲染过程计算量很大，每条射线都要采样很多点。但实际上，一条射线上的大部分区域都是空区域，或者是被遮挡的区域，对最终的颜色没有啥贡献。

因此，作者采用了一种“coarse to fine" 的形式，同时优化 coarse 网络和 fine 网络。先在光线上均匀采样一些点，并计算其体积密度，然后基于这些密度值进行重采样，得到更倾向于物体内部和表面的点，以提高渲染效果。

方法：

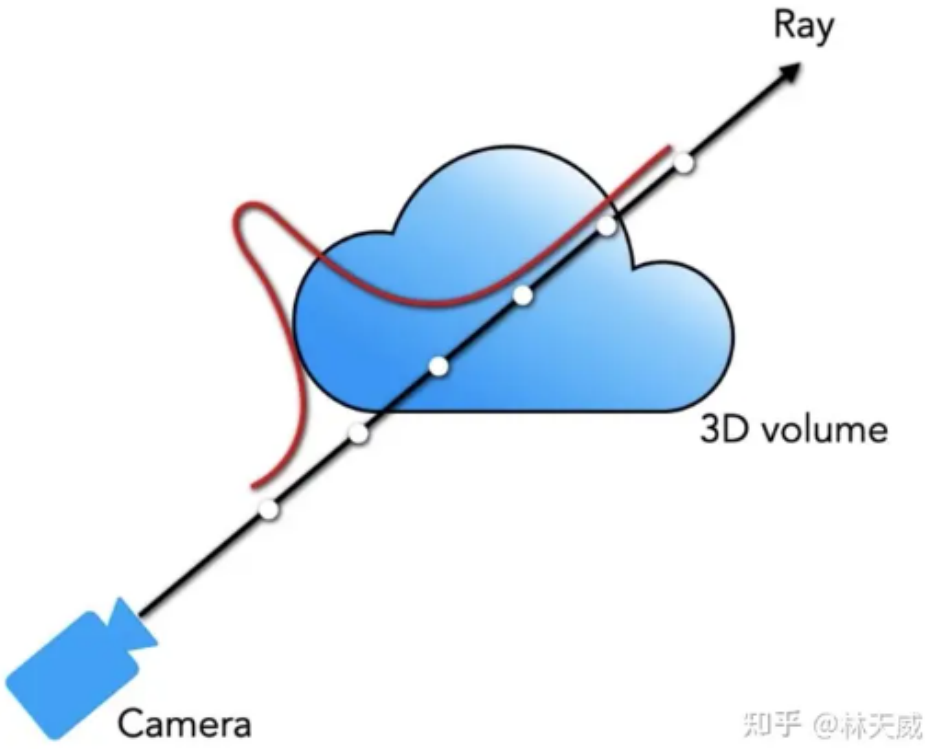
- 首先，我们使用分层采样对一组 Nc 个较为稀疏的位置进行采样
- 根据前面离散近似的公式在这 Nc 个位置上评估 “coarse” 网络
- 给定这个“粗略”网络的输出，我们然后在每个光线上产生更有信息的采样点，其中样本**更偏向于**体积的相关部分

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} w_i c_i \,, \qquad w_i = T_i(1 - \exp(-\sigma_i \delta_i)) \,.$$

得到的 w 要进行归一化

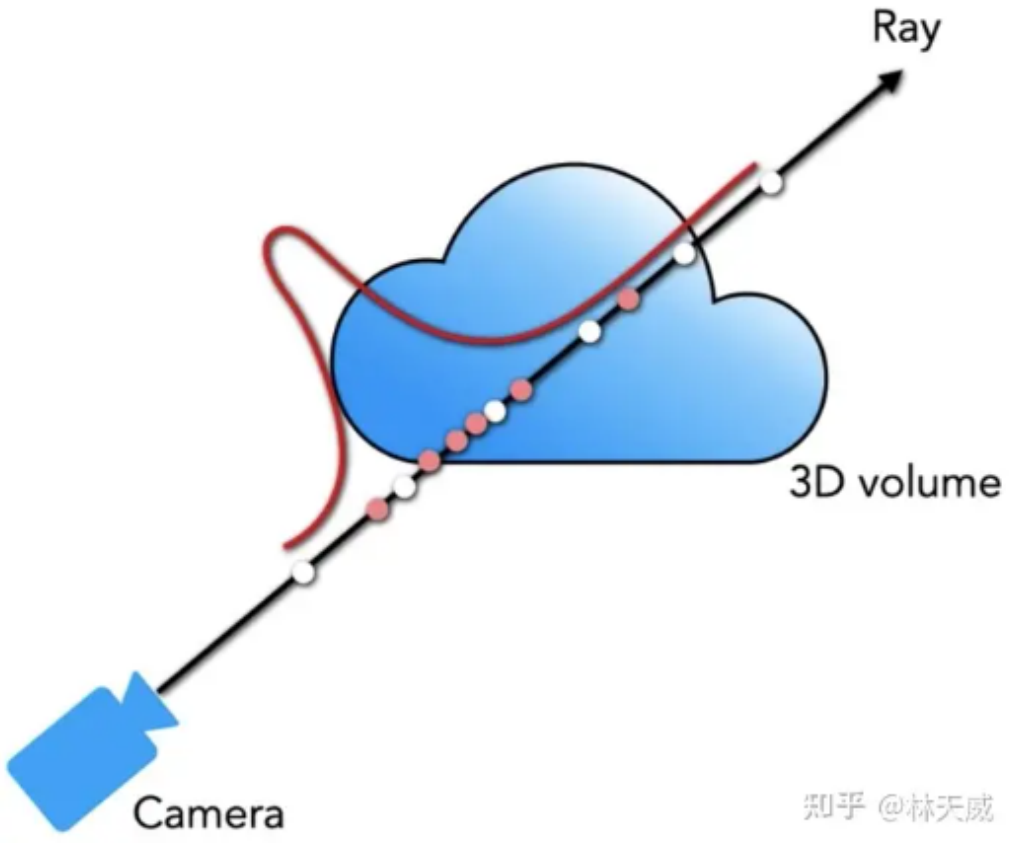
$$\hat{w}_i = \frac{w_i}{\sum_{j=1}^{N_c} w_j}$$

根据上面的归一化之后的 w_i，可以粗略地得到射线上物体分部的情况



然后基于得到的概率密度函数来采样 Nf 个点，并用这 Nf 个点，结合前面 Nc 个点，一同计算 fine 网络的渲染结果 C_f

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$



3. 损失函数和训练细节

对于每个场景，我们针对单独的神经连续体积表示网络进行优化。这只需要场景的捕获的RGB图像数据集、相应的相机姿态和内参参数以及场景边界（对于合成数据，我们使用地面真实的相机姿态、内参和边界；对于真实数据，我们使用COLMAP结构运动包[39]估计这些参数）。

在每次优化迭代中，我们从数据集的所有像素中随机采样一个批次的相机光线，然后按照5.2节中描述的分层采样方法从粗略网络查询Nc个样本和从精细网络查询Nc + Nf个样本。然后，我们使用4节中描述的体积渲染过程从两组样本中渲染每条光线的颜色。

我们的损失函数仅仅是粗略和精细渲染的渲染像素颜色与真实颜色之间的总平方误差

最后，训练损失的定义倒是非常简单，直接定义在渲染结果上的L2损失(同时优化coarse 和 fine）：

$$\mathcal{L} = \sum_{r \in R} \left[\left\| \hat{C}_c(r) - C(r) \right\|_2^2 - \left\| \hat{C}_f(r) - C(r) \right\|_2^2 \right]$$

训练时长方面，论文中提及的速度是一个场景要用单卡V100 训练1–2天左右。

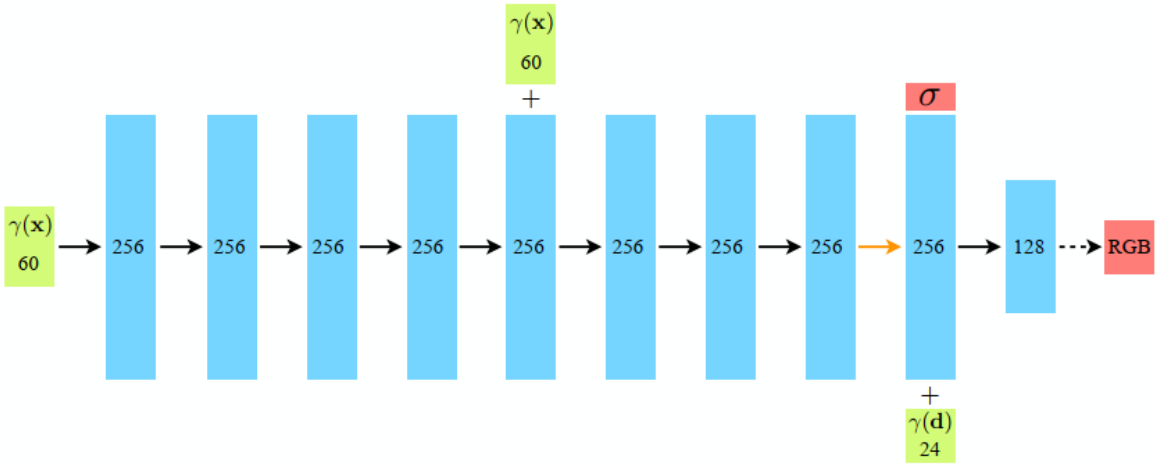


图7：我们全连接网络结构的可视化。绿色代表输入向量，蓝色代表中间隐藏层，红色代表输出向量，每个块中的数字表示向量的维度。所有层都是标准的全连接层，黑色箭头表示有ReLU激活的层，橙色箭头表示没有激活函数的层，黑色虚线箭头表示有sigmoid激活的层，+ 表示向量串联。输入位置的位置编码（γ(x)）通过8个具有256个通道的全连接ReLU层进行处理。我们遵循 DeepSDF [32]的架构，并在第五层的激活函数中包含一个跳跃连接以串联此输入。额外的一层输出体积密度σ（经ReLU激活以确保输出的体积密度非负）和一个256维的特征向量。该特征向量与输入视线方向的位置编码（γ(d)）进行串联，并由具有128个通道的额外全连接ReLU层进行处理。最后一层（带有sigmoid激活）输出在位置x处由方向为d的射线观察到的发射RGB辐射。

Conclusion

我们的工作直接解决了先前使用MLP来表示对象和场景为连续函数的工作的不足之处。我们证明了将场景表示为 5D 神经辐射场（一个MLP，其输出体积密度和视角相关的辐射作为3D位置和2D视角方向的函数）比以往主导的方法——训练深度卷积网络输出离散体素表示——产生更好的渲染效果。

虽然我们提出了一种分层采样策略来提高渲染的采样效率（包括训练和测试），但在有效优化和渲染神经辐射场方面仍有许多进展空间。**未来工作的另一个方向是可解释**

性：例如体素网格和网格等采样表示允许对渲染视图的期望质量和失败模式进行推理，但当我们将场景编码到深度神经网络的权重中时，如何分析这些问题尚不清楚。我们相信这项工作在基于真实世界图像的图形管线方面取得了进展，复杂的场景可以由从实际物体和场景图像中优化得到的神经辐射场组成。

Thinking

NeRF 的最大的优点和价值：

1. **连续性和完整性**：NeRF 使用一个神经网络隐式地表示3D场景。这个表示不仅是连续的，但也包含了空间中未采样区域的信息。相比于传统的离散表示（如三维体素网格或点云），NeRF 可以生成无间隙、无需填充的渲染图像。
2. **详细的渲染**：NeRF 能够生成高质量的、细节丰富的渲染图像，甚至包括复杂的视差和阴影效果。NeRF 还可以模拟不同的光照条件和相机视角，从而得到更加丰富和多样的渲染结果。
3. **高效的存储和计算**：NeRF 的存储效率比显式的三维数据结构（如三维网格或点云）要高得多。这是因为它是由一个固定大小的神经网络表示的，而不是直接存储每一个空间点的信息。此外，NeRF 的计算效率也比许多传统的渲染技术要高。
4. **泛化能力**：NeRF 的神经网络可以学习到场景的一般特性，使得它能够在少量的训练数据下，预测出未观察到的视角或光照条件下的场景外观 (虽然不一定准)。
5. **合成和编辑**：由于 NeRF 是使用神经网络表示的，因此可以很方便地进行场景合成和编辑。例如，可以通过改变网络的输入或参数，来改变场景的形状、颜色或光照。