

DistillNeRF

DistillNeRF: Perceiving 3D Scenes from Single-Glance Images by Distilling Neural Fields and Foundation Model Features

<https://arxiv.org/abs/2406.12095>

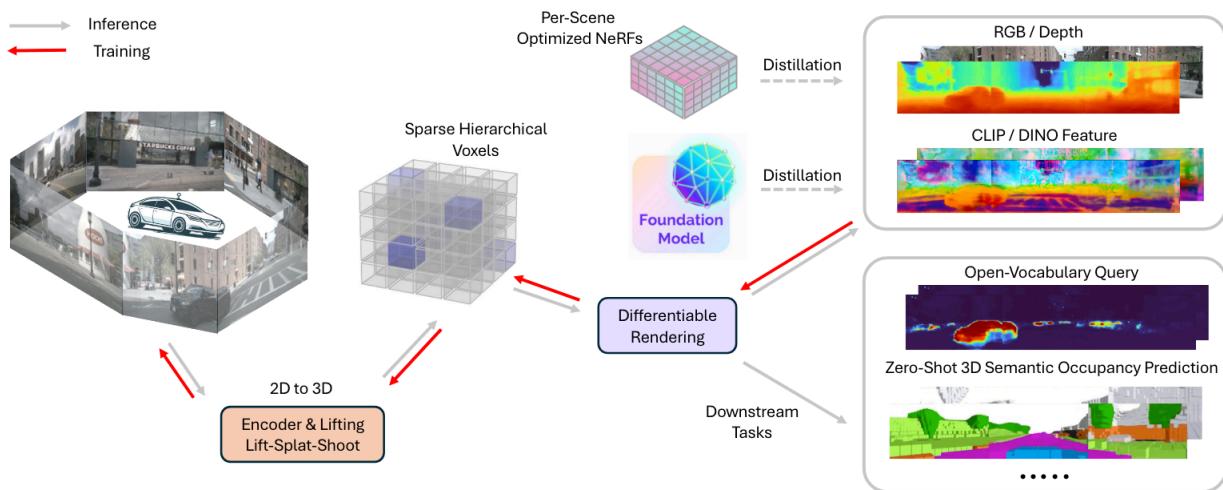


Figure 1: DistillNeRF is a generalizable model for 3D scene representation, self-supervised by natural sensor streams along with distillation from offline NeRFs and vision foundation models. It supports rendering RGB, depth, and foundation feature images, without test-time per-scene optimization, and enables zero-shot 3D semantic occupancy prediction and open-vocabulary text queries.

又一个把CLIP/DINO和自动驾驶结合起来的工作，通过蒸馏来解决Few Shot问题

DistillNeRF comprises two stages: offline per-scene NeRF training, and distillation into a generalizable model. The first stage trains a NeRF for each scene individually from each driving log, exploiting all available multi-view, multi-timestep information. Specifically, we use EmerNerf [1], a recent NeRF approach with decomposed static and dynamic fields. The second stage trains a generalizable encoder to directly lift multi-camera 2D images captured at a single timestep to a 3D continuous feature field, from which we render images, and supervise with dense depth and novel-view RGB targets generated from the per-scene optimized NeRFs, and foundation model features. Specifically, we propose a novel model architecture with 1) a two-stage Lift-Splat-Shoot encoder [24] to lift 2D observations into 3D; 2) a sparse hierarchical 3D voxel for efficient runtime and memory, parameterized to account for unbounded driving scenes; 3) feature image generation via differentiable volumetric rendering, decoded into appearance, depth, and optionally, foundation model features.

- 弄了个2D特征转3D的encoder
- 八叉树表示的3D体素结构来储存

- 通过奇奇怪怪的方式反复pass，生成coarse、fine类似的深度图，特征图

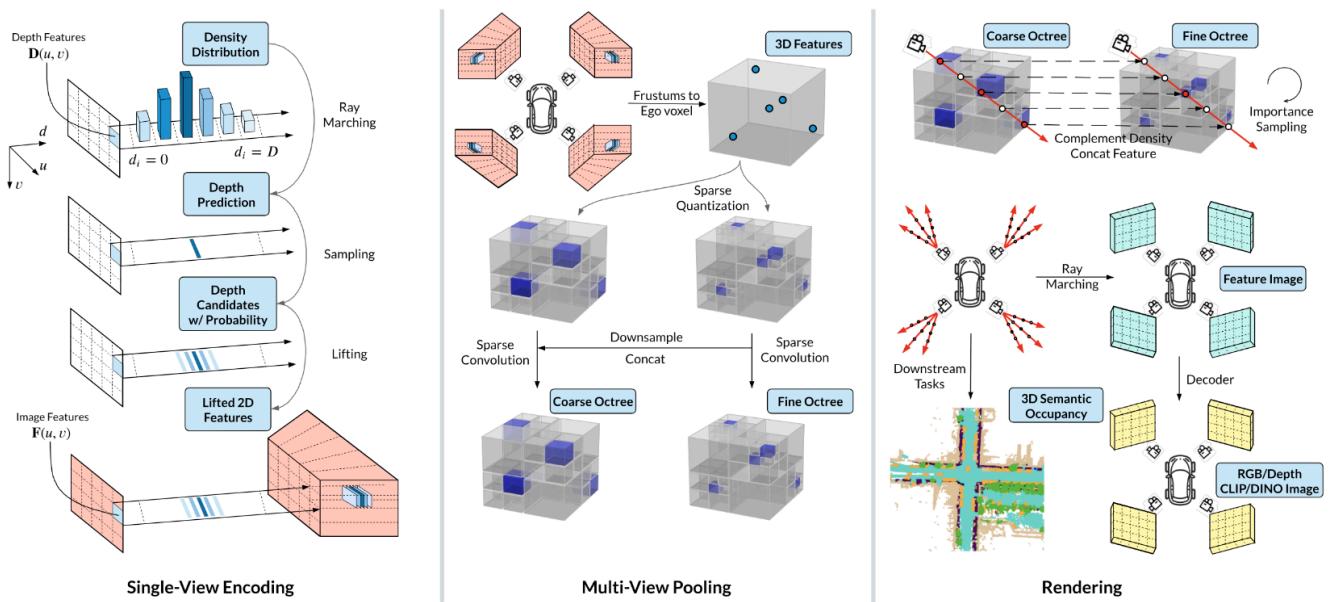


Figure 2: DistillNeRF model architecture. (left) single-view encoding with two-stage probabilistic depth prediction; (center) multi-view pooling into a sparse hierarchical voxel representation using sparse quantization and convolution; (right) volumetric rendering from sparse hierarchical voxels.

Method

3 Method

DistillNeRF predicts a generalizable scene representation in the form of sparse hierarchical voxels from single-timestep multi-view RGB image inputs, and is trained through volumetric rendering to output RGB, depth, and feature images.

The method is depicted in Fig. 1, the detailed architecture in Fig. 2, and key capabilities in Fig. 3. Inputs are N posed RGB camera images $\{I_i\}_{i=1}^N$. We use a 2D backbone to extract N feature images $\{X_i\}_{i=1}^N$. We then lift the 2D features to a 3D voxel-based neural field $\mathcal{V} \in \mathbb{R}^{H \times W \times D \times C}$ using the corresponding camera matrix, and apply sparse quantization and convolution to fuse features from multiple views. To account for unbounded scenes we use a parameterized neural field with fixed-scale inner voxels, and varying-scale outer voxels contracting the infinite range. Volumetric rendering is performed to supervise the reconstruction of the scene. For better guidance on scene geometry, we “distill” knowledge from offline optimized NeRFs, using rendered dense depth images from original camera views and virtual camera views. Foundation model features, from CLIP or DINOv2, are set as additional reconstruction objectives and thus are also “distilled” into our model to enrich scene

缝了Mip-NeRF/NeRF++

3.1 Sparse Hierarchical Voxel Model

Single-View Lifting. For each of the N camera image inputs, we follow a similar procedure as Lift-Splat-Shoot (LSS) [24] to lift the 2D image features to the 3D neural field. Unlike typical LSS and variants [24, 39, 43] that predict depth in one shot, we propose a two-stage, coarse-to-fine strategy with two jointly trained predictors to capture more nuanced depth. Following prior works, the first stage predicts categorical depth and aggregates them into a single prediction with ray marching. The second stage then predicts a distribution over a fine-grained set of categorical depth values, chosen around the coarse depth prediction.

Specifically, in the first stage, we feed each image to a 2D backbone to generate a depth feature map of size $H \times W \times D$. Inspired by the volume rendering equation [4], the depth feature map is regarded as a discrete frustum where D denotes the number of pre-defined categorical depths. Each entry in the frustum is a density value. That is, the d 'th channel of the frustum at pixel (h, w) represents the density value $\sigma_{h,w,d}$ of the frustum entry at (h, w, d) . The occupancy weight of entry (h, w, d) is then

$$\mathbb{O}(h, w, d) = \exp\left(-\sum_{j=1}^{d-1} \delta_j \sigma_{h,w,j}\right) (1 - \exp(-\delta_d \sigma_{h,w,d})), \quad (1)$$

where $\delta_d = t_{d+1} - t_d$ is the distance between each pre-defined depth t in the frustum. Coarse depth for pixel (h, w) is obtained by aggregating with ray marching:

$$\mathbb{D}(h, w) = \sum_{d=1}^D \mathbb{O}(h, w, d) t_d. \quad (2)$$

In the second stage, we dynamically generate a fine-grained set of D' depth candidates centered around the coarse depth prediction. We then combine learned embeddings for each candidate with the depth features from the first stage, and feed them to another network to generate the density of each depth candidate. The occupancy weights \mathbb{O}' of the fine-grained depth candidates are predicted similarly by Eq 1, which can also be regarded as probabilities of each depth candidate.

With the candidate depths associated with probabilities, we then lift 2D image features to 3D. Specifically, we use the feature pyramid network (FPN) [46] to get 2D image features ϕ , and assign the 2D image features to the 3D frustum. That is, for pixel (h, w) , its image feature $\phi_{h,w}$ is distributed to each depth candidates t'_d by $[\mathbb{O}'_{h,w,d} \phi_{h,w}, \sigma'_{h,w,d}]$, where we scale the pixel image feature $\phi_{h,w}$ with occupancy $\mathbb{O}'_{h,w,d}$ and concatenate it with density $\sigma'_{h,w,d}$.

两阶段深度预测策略：

- 与传统的“一步到位”深度预测方法不同，DistillNeRF 采用了两阶段的策略：
 - 第一阶段：**预测分类深度，即对不同深度的类别进行估计，并通过射线行进 (ray marching) 聚合成单一的深度预测。它生成了一个深度特征图，大小为 $H \times W \times D$ ，其中 D 表示预定义的深度类别数。
 - 第二阶段：**在第一阶段的基础上，根据粗略的深度预测结果，选择一个更精细的深度集合，并预测这些候选深度的概率分布。

密度和占据权重计算：

- 第一阶段生成的深度特征图可以视作一个离散的截锥体，每个条目表示某一位置的密度值，即 $\sigma_{h,w,d}$ 代表在像素 (h, w) 处第 d 个深度条目的密度。
- 使用体渲染方程，对每个深度位置的占据权重进行计算，公式如下：

$$O(h, w, d) = \exp \left(- \sum_{j=1}^{d-1} \delta_j \sigma_{h,w,j} \right) (1 - \exp(-\delta_d \sigma_{h,w,d})), \quad (1)$$

其中， δ_d 表示相邻深度间的距离。该公式描述了在给定像素 (h, w) 处深度位置 d 的占据概率。

深度聚合：

- 通过射线行进，将不同深度位置的占据权重与对应深度结合，聚合出每个像素 (h, w) 的粗略深度：

$$D(h, w) = \sum_{d=1}^D O(h, w, d) t_d, \quad (2)$$

其中， t_d 是预定义的深度位置。这个聚合过程能够得到每个像素的整体深度估计。

通过这种两阶段的深度预测策略，DistillNeRF 能够更精细地捕捉场景中的深度变化，并在稀疏视角的情况下提升 3D 重建的精度。第一阶段提供了粗略的全局深度信息，第二阶段则进一步细化这些估计，从而提升整体的深度预测效果。

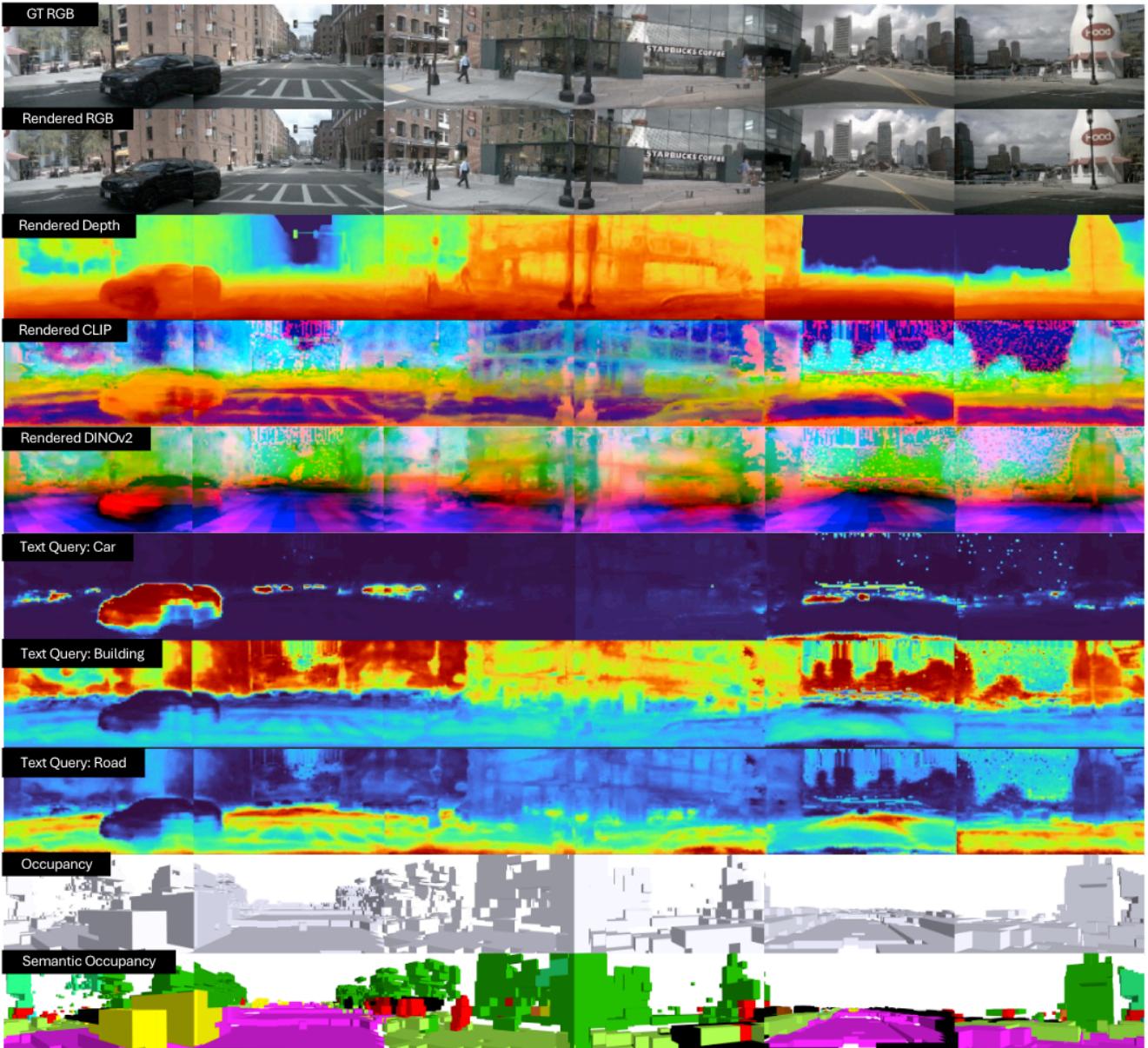


Figure 3: Given single-frame multi-view cameras as input and without test-time per-scene optimization, DistillNeRF can reconstruct RGB images (row 2), estimate depth (row 3), render foundation model features (rows 4, 5) which enables open-vocabulary text queries (rows 6, 7, 8), and predict binary and semantic occupancy in zero shot (rows 9, 10).

多视角融合 (Multi-View Fusion):

- **视锥体转换:** 在处理每个视角的图像时，模型首先为每个视角构建一个视锥体 (Frustum)。视锥体代表了相机视野内从相机到物体的 3D 空间范围。
- **转换为世界坐标系:** 然后，将这些视锥体转换到世界坐标系中，这样可以统一各个视角的空间表示。通过使用相机的外参矩阵，模型能够将这些视角信息对齐到同一个 3D 空间中。

- **融合到共享的 3D 体素场**: 模型将多个视角的信息融合到一个共享的 3D 体素表示中。每个体素 (Voxel) 代表世界坐标系中的一个区域，并且可以存储密度和特征信息。
- **特征融合**: 当来自不同视角的提升后的视锥体条目落在同一个体素中时，模型使用平均池化 (average pooling) 的方法，将这些不同视角的信息进行融合，得到该体素的最终特征。

稀疏分层体素 (Sparse Hierarchical Voxels):

- **稀疏量化的优势**: 与使用密集体素的方法不同，稀疏量化可以避免在大部分为空的区域进行不必要的计算，从而节省计算资源和内存。传统密集体素方法会对整个神经场进行均匀量化，即使在大多数为空的区域也进行计算，而稀疏体素则更高效。
- **八叉树表示 (Octree Representation)**: 模型使用八叉树的方式对神经场进行递归划分。根据提升的 2D 特征在 3D 空间中的位置，将神经场分割成不同层次的体素。这种表示方法可以更精确地捕捉提升特征的 3D 位置。
- **细致与粗略的八叉树**: 为了解决细致体素在渲染时可能遇到的查询困难（例如，由于采样点之间的间隔较大，可能会丢失远处的特征），模型生成了两种八叉树：
 - **细致八叉树**: 有更多的量化层级，可以捕捉到更精细的 3D 特征细节。
 - **粗略八叉树**: 覆盖更大范围的场景信息，适合表示大范围的粗略信息。
- **稀疏卷积 (Sparse Convolutions)**: 在这两种八叉树上应用稀疏卷积，用于编码体素之间的关系和交互。细致八叉树中的特征还会被下采样并与粗略八叉树的特征进行拼接，以进一步增强细节的表现。

与传统方法的对比:

- 传统的神经场方法通常在一个固定范围内定义场景，这意味着它们只能对预设的场景范围进行建模 [39, 21, 22]。

- DistillNeRF 的目标是处理自动驾驶场景中的无界场景 (unbounded-scene) , 例如远处的天空或建筑物等。

参数化神经场:

- 为了处理无界场景, DistillNeRF 提出了一种参数化的神经场方法, 它在不同的距离范围内采用不同的分辨率。
- **内部范围 (Inner Range)** : 保持近距离区域 (如 50 米内) 的体素 (Voxel) 在真实比例和高分辨率下, 因为这些区域对于占据预测等任务至关重要。
- **外部范围 (Outer Range)** : 对远处的场景进行“收缩” (contracting) , 将其映射到较低分辨率的体素表示上, 从而减少对远距离场景 (如天空、远处的建筑物) 进行渲染时的内存和计算成本。

转换函数:

- 该方法使用了一个转换函数, 将世界坐标系中的 3D 点 (\mathbf{p}) 映射到参数化的神经场坐标系中:

$$f(\mathbf{p}) = \begin{cases} \alpha \frac{\mathbf{p}}{p_{\text{inner}}}, & \text{if } |\mathbf{p}| \leq p_{\text{inner}}, \\ \left(1 - \frac{p_{\text{inner}}}{|\mathbf{p}|}(1 - \alpha)\right) \frac{\mathbf{p}}{|\mathbf{p}|}, & \text{if } |\mathbf{p}| > p_{\text{inner}}. \end{cases} \quad (3)$$

- 这里, $|\mathbf{p}|$ 表示 3D 点 (\mathbf{p}) 的距离, p_{inner} 表示内部体素的范围, α 是内部范围在参数化神经场中所占的比例, 取值范围在 $([0, 1])$ 之间。
- 当点 \mathbf{p} 位于内部范围时, 使用 $\alpha \frac{\mathbf{p}}{p_{\text{inner}}}$ 进行线性缩放; 当点超出内部范围时, 使用一种非线性变换, 将其映射到远处的范围内。

效果与一致性:

- 这种转换使得 3D 点在参数化后的神经场中始终位于 $[0, 1]$ 范围内，同时确保在内部范围内的体素具有高分辨率、真实比例，而在外部范围则以较低分辨率进行表示。
- 该参数化方法在单视角提升过程（针对深度空间）和多视角融合过程（针对 3D 坐标空间）中保持一致性，从而提高整体建模的精度和效率。

这段文字介绍了 DistillNeRF 模型中自监督蒸馏训练（Self-supervised Training with Distillation）的策略，主要包括两部分：从离线 NeRF 模型蒸馏以及从基础模型蒸馏。具体解释如下：

从离线 NeRF 模型蒸馏（Distillation from Offline NeRFs）：

- **问题背景**：虽然可以通过重建 RGB 图像来训练模型，但单纯依赖单时间步的图像输入，学习几何结构信息仍然具有挑战性，特别是在自动驾驶应用中，车载相机的视角通常是面向外部且视角重叠少，这使得多视角重建变得更加困难。
- **提出的解决方案**：使用高质量的每视角优化 NeRF（如 EmerNeRF）来提供几何信息。这些离线优化的 NeRF 模型能够提供关于场景的更精确的深度信息，可以用来对当前模型进行监督。主要有两种蒸馏方式：
 - **稠密 2D 深度（Dense 2D Depth）**：通过从离线 NeRF 生成的稠密深度图进行监督。传统的 LiDAR 点云通常较稀疏，提供的深度标签也有限。通过将离线优化的 NeRF 作为深度自动标注工具，可以为训练目标图像生成稠密的深度图，并将其作为额外的深度监督项。
 - **虚拟相机（Virtual Cameras）**：除了使用原始视角的深度信息外，还利用虚拟相机生成的新视角数据。通过从离线 NeRF 模型生成虚拟相机视角下的深度图和 RGB 图像，将其作为额外的重建目标。这样可以人为地增加目标图像的数量，提高视角重叠，从而促进深度预测的一致性，并提升新视角合成的效果。

从基础模型蒸馏（Distillation from Foundation Models）：

- 引入基础模型特征：除了 RGB 图像和深度图的重建，作者还希望通过基础模型（如 CLIP 和 DINOv2）的特征蒸馏，学习更广泛的 3D 表示能力。
- 方法：在渲染的 2D 特征图上引入一个多层感知机（MLP），使模型能够重建基础模型的特征图像。通过最小化 L1 损失 $\mathcal{L}_{\text{found}}$ ，训练模型对基础模型的特征进行重建，从而引入更丰富的语义信息。

训练目标（Training Objective）：

- 训练过程的损失函数由多项损失组合而成：

$$L = L_{\text{rgb}} + L_{\text{depth}} + L_{\text{density}} + L_{\text{NeRF}} + L_{\text{found}}, \quad (4)$$

- L_{rgb} 和 L_{depth} 是针对 RGB 和深度图的渲染损失。
- L_{density} 是一个密度熵损失项，用于优化体积密度。
- L_{NeRF} 和 L_{found} 则是从离线 NeRF 模型和基础模型蒸馏的损失，用于引导模型学习几何结构和语义信息。