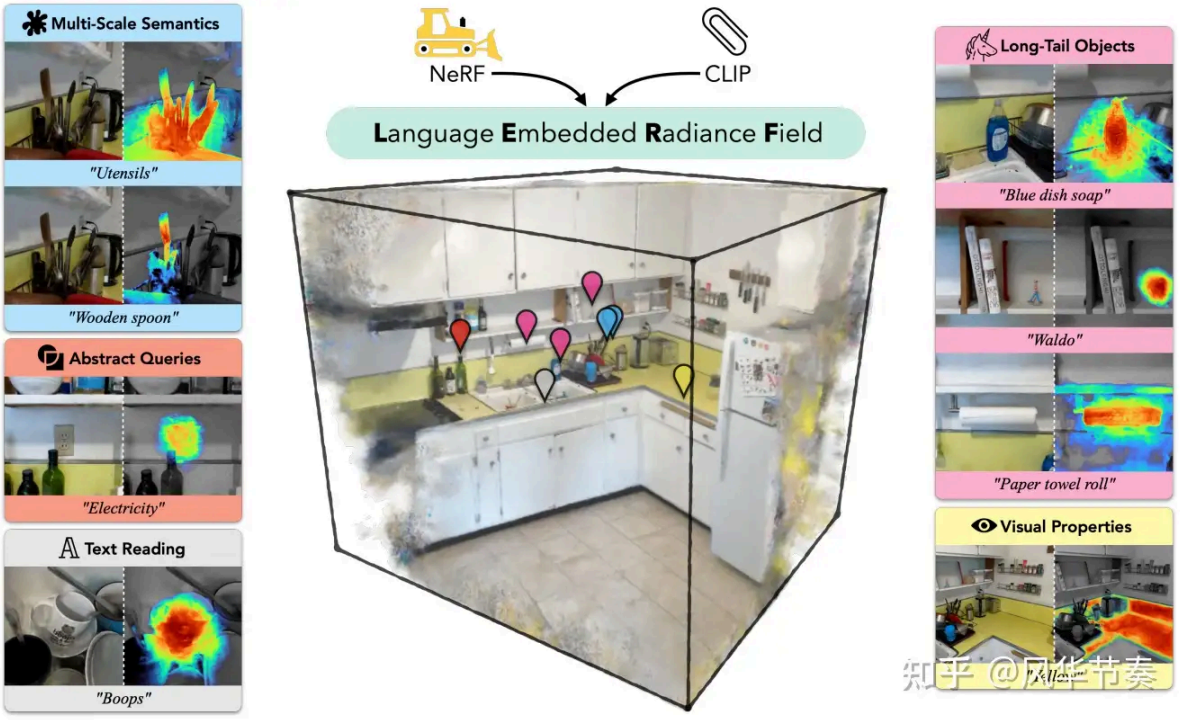
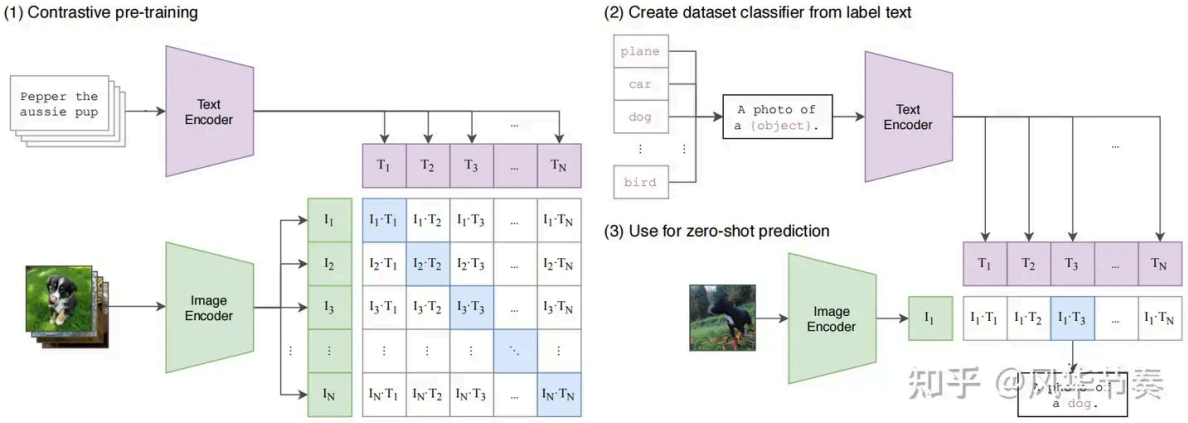
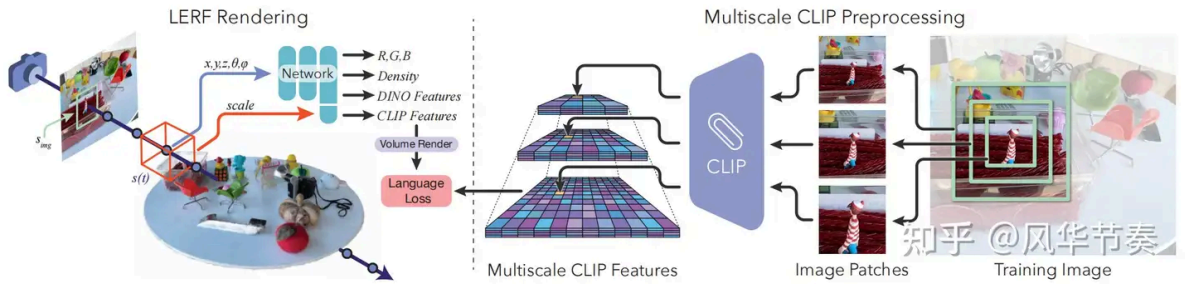
LERF:Language Embedded Radiance Fields



NeRF对场景进行了高效渲染和3D重建，但是NeRF的输出仅为由color和density组成的神经辐射场。现实世界中，人们通过描述物体的语义特征对世界进行描述，由此，本文将物体的语义嵌入到3D神经辐射场中，进而实现了3D场景中对物体进行语义查询的功能。



CLIP，全称为Contrastive Language-Image Pre-training，其将NLP和CV相结合，能够对图像表示和文本描述进行联合学习，理解和生成相关信息。CLIP对大量(image,text)对进行学习，掌握其中的映射关系。首先，CLIP对了Text encoder和Image encoder进行了联合训练，将N个文本特征和N个图像特征进行组合，其中，矩阵对角线上的N个组合为正样本，其余个为负样本，这里对文本特征和图像特征的余弦相似度进行计算，训练的目标即为最大化正样本相似度，最小化负样本相似度。由于CLIP可以实现zero-shot图像分类，故可以不需微调直接在下游任务上进行使用。如图根据下游任务的分类标签构建每个类别的描述文本，将这些文本送入Text encoder得到N个对应的文本特征。接着，将要预测的图像样本输入Image encoder得到图像特征，将其与N个文本特征组合并计算各自的余弦相似度，取最大值的组合，其文本即为输入图像对应的类别。CLIP样例中为单张图片多个类别，LERF中为多个3D辐射场单个语义输入，故含义相同，与此例对调即可。



NeRF的输出仅为RGB和体密度σ，在此基础上，我们将引入了语义场，物体位置和尺度作为输入，d维语义向量作为输出。这个嵌入是关于视角独立的，因为3D场景中某个位置的语义应与视角无关。由于LERF是关于物体的而非关于某个点的方法，故对于光线上的每个点我们都引入一个尺度参数。我们在图像平面上固定一个初始尺度，并定义s(t)随焦距f和该点到光源的距离t成比例增长。

$$s(t) = s_{\text{img}} * f_{xy}/t$$

然后对语义嵌入进行渲染，得到原始语义向量

$$\hat{\phi}_{\text{lang}} = \int_t w(t) F_{\text{lang}}\left(r(t), s(t)\right) dt$$

接着如CLIP中所示，将每个语义嵌入规范化到单位球

$$\phi_{\text{lang}} = \hat{\phi}_{\text{lang}} / ||\hat{\phi}_{\text{lang}}||$$

在对输出语义场Flang进行监督时，我们只能查询到语义嵌入所对应的图像补丁，而无法精确到像素。因此，我们采用以射线起源像素点为中心的，大小为Simg的图像补丁作为监督来优化每个渲染部分(frustum)。但是，在LERF优化过程中计算语义嵌入的代价十分昂贵，所以我们预先计算了多个图像裁剪尺度上的图像金字塔，并存储每个裁剪尺度上的语义嵌入。图像金字塔从最小尺度Smin到最大尺度Smax之间共有n层，图像裁剪之间要有50%的重叠部分。在训练过程中，我们从输入视图中随机采样射线源点，并为其均匀随机的选择Simg。由于射线源点并不一定在图像金字塔image crop的中心，所以我们对其上下范围内4个最近的crop产生的语义嵌入进行三线性插值，得到ground truth embedding 。然后最小化rendered embedding 和ground truth embedding之间的损失，即最大化它们之间的余弦相似度，缩放常数为λ

$$L_{\text{lang}} = -\lambda_{\text{lang}} \phi_{\text{lang}} \cdot \phi_{\text{lang}}^{\text{gt}}$$

此时得到的渲染结果，有时是不完整的并且在区域中包含异常值，甚至导致相关图的平滑度和边界质量恶化。因此，为缓解这种情况，我们训练了另一个场Fdino，其在每一个点上都输出一个DINO特性，DINO是一种自监督的学习方式，其利用对比学习框架高效有力地对无标签图像数据进行表示。其核心为在自监督学习过程中发现并捕捉图像的涌现性质，主要作用是提高神经网络的表示学习能力，使其能够更好地理解和提取输入数据的特征。我们采用与语义嵌入Φlang相同的方式渲染DINO嵌入Φdino,但不将其规范化到单位球。DINO在推理过程中被显式地应用，因为DINO与CLIP的输出头共享同一个架构主干，所以仅为一个额外的正则化器。

以上为LERF的各个组成部分，在进行整合时，直观上讲，3D语义嵌入不应该影响底层场景表示中的密度分布。为此，我们训练了两个相互独立的网络，一个输出特征向量,另一个输出标准NeRF输出(color,density)。

在开放式场景中，不存在非常详尽的类别列表供我们进行语义查询，我们认为自然语言的开放性和模糊性是一个好处，并提出了一种给定任意文本，从LERF中查询三维相关性映射的方法--Querying LERF。Querying LERF包含两个部分：(1)获得渲染嵌入的相关性分数；(2)自动选择给定提示词对应的物理尺度s。

在计算相关性分数之前，我们定义了一系列的规范短语，包括：“object”, “things”, “stuff”, and “texture”，我们选择这些词作为用户可能进行的查询的定性“平均”词，并且发现不论是具体还是抽象的查询词，它们具有非常好的鲁棒性。

为了给每个语义嵌入分配一个分数，我们为文本查询计算了一个语义嵌入Φquer,以及一组规范短语语义嵌入Φcanon。我们计算了渲染语义嵌入和规范语义嵌入之间的余弦相似度，之后计算了与提示文本嵌入之间的成对的softmax函数。相关性分数为

$$\min_i \frac{\exp(\phi_{\text{lang}} \cdot \phi_{\text{quer}})}{\exp(\phi_{\text{lang}} \cdot \phi^i_{\text{canon}}) + \exp(\phi_{\text{lang}} \cdot \phi_{\text{quer}}))}$$

直观地说，这个分数表示与规范嵌入相比，渲染嵌入更接近查询嵌入的程度。

对于每一次查询，我们都为语义场计算一个物理尺度s,在计算这个尺度s时，我们生成了一个尺度范围为0到2米的相关度地图，并选择了产生最高相关度得分的尺度。这个尺度被用于输出映射中的所有像素。