基于 Transformer 的 NeRF：面向前向场景表示的探索

摘要

Neural Radiance Fields（NeRF）通过对三维空间中的辐射场进行隐式建模，实现了高质量新视角合成。然而，经典 NeRF 的训练属于“场景专属的逆向优化”：针对单一场景迭代优化一个 MLP，并在体渲染积分中进行大量采样与反向传播，导致训练代价高、泛化能力弱。本文探索一种“前向（forward）预测”的替代范式：利用视觉 Transformer（以 VGGT 为代表的多视角聚合架构）从多视角图像直接预测可用于体渲染的场景表示，从而减少/避免对每个场景的长时间优化。我们提出两条实现路径：（1）预测 NeRF 的 MLP 参数（超网络/权重生成）；（2）预测结构化的低维网格表示（Tri-plane/Hybrid Grid）并配合全局共享的轻量解码器（Tiny MLP）。此外，针对动态场景，我们提出一种在特征空间进行“语义一致”的时间插值方法：不显式对时间维度做回归，而是对跨时刻 token 进行匹配与插值，从而生成中间时刻的稳定表示。本文给出方法设计、训练目标与工程实现中的关键 trick，并讨论该前向范式在泛化、语义一致性与动态生成方面的意义与局限。

1. 引言

NeRF 作为三维重建与新视角合成的重要里程碑方法，通过学习连续函数 $f_\theta : (\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma)$, 将空间坐标 $\mathbf{x} \in \mathbb{R}^3$ 与视角方向 $\mathbf{d} \in \mathbb{R}^3$ 映射为颜色 $\mathbf{c} \in \mathbb{R}^3$ 与体密度 $\sigma \in \mathbb{R}$，再通过体渲染积分生成像素颜色。该范式的核心问题在于：经典训练流程对每个场景都需进行长时间的逆向优化（per-scene optimization），并伴随高昂的射线采样与体渲染计算。

近年来，Instant-NGP 等方法在表示与加速上取得进展，但多数仍依赖场景级训练。与之相对，视觉 Transformer 在多视角几何推理方面展现出强大的“前向理解”能力：模型可以从多视角图像直接推理深度、点云、相机参数等三维要素。受此启发，本文探索：**能否让多视角 Transformer 直接输出可渲染的辐射场表示，从而将 NeRF 从“逆向优化”转向“前向预测”？**

我们关注以下目标：

- **泛化性**：用单一模型处理不同场景，减少每个场景的专属优化。
- **结构化表示**：将场景压缩为低维结构（如 Tri-plane / 网格 latent），提高可控性与可编辑性。
- **动态一致性**：在时间维度上避免逐点回归导致的抖动/模糊，获得语义一致的动态生成。

2. 背景与相关工作（简述）

2.1 NeRF 与体渲染

对相机射线 $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ 采样 N 个点 $\{t_i\}$，得到 $(\mathbf{c}_i, \sigma_i)$，像素颜色通过离散体渲染近似：$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i, \quad \alpha_i = 1 - \exp(-\sigma_i \Delta_i), \quad T_i = \prod_{j<i}(1 - \alpha_j)$. 训练通常最小化渲染颜色与真实像素的误差，并可加入 density/regularization 等项。

2.2 VGGT：（CVPR 2025 Best Paper）

VGGT（Visual Geometry Grounded Transformer）是一类**前向（feed-forward）多视角网络**：给定一张、少量或大量视图，能够在**不依赖后处理几何优化**的前提下，直接推理场景的关键三维要素，包括**相机内外参、深度图、点图（point maps）、稠密点云**以及**3D 点轨迹（point tracks）**等[2]。其特点是结构相对简洁、推理高效（可在秒级生成重建结果），并在多项 3D 任务上取得 SOTA 表现。

关于具体的实现方案，VGGT 可以分为两个主要部分：aggerator与head. aggerator首先使用DINOv2/CNN将输入图像 patchify 为 tokens，并拼接表示3D位置的camera tokens，用于相机参数预测。然后在层级结构中交替使用frame-wise self-attention）与global self-attention，从而同时保留单帧细节与多视角一致性，学习到3D特征。Heads：VGGT设置了多个DPT head类型的head进行输出。如camera head 输出相机内外参；。

在本项目中，我们将VGGT视为一个有充足能力的backbone，利用其预训练好的参数，进行一些调整与改进，从而应用到NeRF当中。采用 VGGT 的“**Aggregator + Head**”范式，核心改动在于：将 head 的预测目标从几何要素扩展为**可用于 NeRF 体渲染查询的场景表示**（例如 MLP 参数或 Tri-plane latent），从而探索“由多视角图像直接生成辐射场表示”的前向路线。

3. Novel NeRF：从“逆向场景优化”到“once-for-all 前向场景表征”

传统 NeRF 的主流范式是**逆向**的：对每一个新场景都要执行长时间的 per-scene 优化，才能得到一个“能渲染该场景”的辐射场参数 θ。这带来两个根本性瓶颈：
(1) **效率瓶颈**：同一套训练流程需要对每个场景重复数千/数万步迭代，推理无法“一次完成”；
(2) **部署与泛化瓶颈**：模型更像“场景记忆器”，而非可泛化的场景编码器（scene encoder）。

本文所称的 Novel NeRF，核心是在范式上转向 **once-for-all 的前向（forward）生成**：给定多视角图像，模型直接输出一个可渲染的场景表示 $\mathcal{S}$，从而在生成“场景表示”这一步上做到**一次前向推理**（而非多次优化）。这条路线的意义不仅在于加速，更在于它有潜力发展为一种**新型 3D 场景表示**：

- $\mathcal{S}$ 可被视为对场景的**压缩表示**（compact scene code），存储成本低；
- 给定 $\mathcal{S}$ 可通过体渲染得到**任意视角投影**，满足下游使用；

- 若该表示能跨场景泛化，则“从图像到 3D 表示”的链路将更接近通用视觉系统的需求。

3.1 Feed-Forward的实现路径

围绕“前向生成可渲染表示”，我们规划了两条实现路线,出于时间和资源的限制，我们主要围绕第二条路线进行模型训练：

3.2 路线一：T-NeRF

3.2.1 模型构建

T-NeRF 继承 VGGT 的“Aggregator + Head”范式：

- 输入：K 张多视角图像 $\{I_k\}_{k=1}^{K}$。
- Aggregator：提取并跨视角聚合 token，得到全局场景表征 $\mathbf{Z}$。
- Radiance Head：将 $\mathbf{Z}$ 映射为可渲染表示 $\mathcal{S}$，使得对任意查询 $(\mathbf{x}, \mathbf{d})$ 都能输出 $(\mathbf{c}, \sigma)$，并可接入标准体渲染器生成像素。

核心思想：把“复杂几何/外观”主要存入结构化网格 latent，由 Transformer 前向预测；把“解码成 RGB/σ 的映射”交给一个全局共享的轻量 MLP。整体可视为：$\mathcal{S} = \{\mathbf{F}_{xy}, \mathbf{F}_{xz}, \mathbf{F}_{yz}\}$，$(\mathbf{c}, \sigma) = h_\psi\Big([\mathbf{f}_{xy}(\mathbf{x}), \mathbf{f}_{xz}(\mathbf{x}), \mathbf{f}_{yz}(\mathbf{x}), \gamma(\mathbf{x}), \gamma(\mathbf{d})]\Big)$.

Tri-plane 预测：

- head 输出三个二维特征平面（示例：$64 \times 64 \times 32$），分别对应 xy, xz, yz 投影；
- 对任意 $\mathbf{x} = (x, y, z)$，在三个平面上做双线性插值：
  - $\mathbf{f}_{xy} = \text{bilinear}(\mathbf{F}_{xy}, (x, y))$
  - $\mathbf{f}_{xz} = \text{bilinear}(\mathbf{F}_{xz}, (x, z))$
  - $\mathbf{f}_{yz} = \text{bilinear}(\mathbf{F}_{yz}, (y, z))$
- 拼接后输入 Tiny MLP 输出 $(\mathbf{c}, \sigma)$。

3.2.2 损失构建与训练

本文使用 NeRF-MAE 提供的预训练数据[1]（https://huggingface.co/datasets/mirshad7/NeRF-MAE），该数据包含多个3D场景的场数据——在三维网格上提供每个 3D 点的 $\mathbf{c}^*(\mathbf{x})$ 与 $\sigma^*(\mathbf{x})$。

因此训练时可采用**点监督（field supervision）**的训练方式：从场数据中采样一批 3D 点 $\{\mathbf{x}_i\}$，由 T-NeRF 预测 $(\mathbf{c}_i, \sigma_i)$，与 GT 直接回归对齐。

使用点监督的训练方法，对采样点 $\{\mathbf{x}_i\}$ 的损失可写为：$\mathcal{L}_{field} = \lambda_c \cdot \frac{1}{N}\sum_i \|\mathbf{c}_i - \mathbf{c}_i^*\|_1 + \lambda_\sigma \cdot \frac{1}{N}\sum_i \|\sigma_i - \sigma_i^*\|_1$. 在实际训练过程中，冻结 Aggregator，仅训练新加 head，同时为提升训练稳定性与泛化，需要加入若干正则项：

- **参数/特征正则**（抑制投机解，提升平滑性）
  Tri-plane/grid latent 加 L_2 正则： $\mathcal{L}_{grid} = \|\mathbf{F}_{xy}\|_2^2 + \|\mathbf{F}_{xz}\|_2^2 + \|\mathbf{F}_{yz}\|_2^2$.

  最终目标： $\mathcal{L} = \mathcal{L}_{field} + \lambda_{grid}\mathcal{L}_{grid}$ (+ 可选的渲染重建项).
  训练流程可概括为：

1) 多视角输入经 VGGT Aggregator 得到 $\mathbf{Z}$；
2) Radiance Head 输出 $\mathcal{S}$（MLP 参数或 Tri-plane）；
3) 对采样点/射线进行查询（以及可选体渲染）得到预测；
4) 计算损失并反向传播，更新 head（及可选更新 aggregator）。

3.3 路线二：ToDO

4. dynamic NeRF

4.1 朴素 Dynamic NeRF

为建立对照，本项目实现并测试了两类常见的动态 NeRF（可对应 vanilla NeRF 风格代码的最小改动实现），它们的共同目标是学习带时间的辐射场： $f_\theta : (\mathbf{x}, \mathbf{d}, t) \mapsto (\mathbf{c}, \sigma)$.

实现方案：

方案 A：时间嵌入直接注入 MLP（Time-as-Input）
做法是对时间标量 t 进行嵌入/位置编码（例如 $\gamma(t)$），并与空间位置编码、方向编码拼接后送入同一个 NeRF MLP： $(\mathbf{c}, \sigma) = f_\theta([\gamma(\mathbf{x}), \gamma(\mathbf{d}), \gamma(t)])$. 这一实现简单，但本质是对“时空变化”做连续回归：当动态较复杂或监督稀疏时，容易出现**边界抖动、局部模糊**，表现为不同时间的高频细节不稳定。

方案 B：静态场 + 动态残差（Two-Network / Residual Dynamics）
将场分解为静态部分与动态偏差，两网络共同决定输出： $(\mathbf{c}_s, \sigma_s) = f_{\theta_s}(\mathbf{x}, \mathbf{d})$, $(\Delta\mathbf{c}, \Delta\sigma) = f_{\theta_d}(\mathbf{x}, \mathbf{d}, t)$, $(\mathbf{c}, \sigma) = (\mathbf{c}_s + \Delta\mathbf{c},\ \sigma_s + \Delta\sigma)$.

实现上，为了提升训练稳定性并加速收敛，我们在 deformation 的动态分支（或残差 MLP）中额外加入了 **LayerNorm**。该做法能有效缓解梯度爆炸/数值不稳定，使模型不必依赖更长的训练轮次来稳定优化过程。

效果分析

我们在 D-NeRF 数据集的 8 个动态场景上对两种朴素实现方案进行了量化评估。表格展示了 PSNR（Peak Signal-to-Noise Ratio，越高越好）和 SSIM（Structural Similarity Index，越高越好）两个主要指标及其标准差。

表 1：Deformation

场景 (Scene)	PSNR (dB)	SSIM	PSNR Std	SSIM Std
bouncingballs	28.70	0.9602	3.31	0.0261
hellwarrior	24.82	0.9487	2.15	0.0145
hook	29.17	0.9634	3.22	0.0228
jumpingjacks	**32.39**	**0.9742**	4.09	0.0327
lego	21.83	0.8481	1.37	0.0422
mutant	31.25	0.9736	3.23	0.0147
standup	**33.31**	**0.9804**	3.61	0.0114
trex	31.38	0.9726	2.85	0.0247
平均	**29.11**	**0.9527**	**2.98**	**0.0236**

表 2：Straightforward

场景 (Scene)	PSNR (dB)	SSIM	PSNR Std	SSIM Std
bouncingballs	31.64	0.9702	1.44	0.0065
hellwarrior	21.77	0.9153	0.91	0.0115
hook	24.29	0.9068	1.06	0.0145
jumpingjacks	**27.58**	**0.9467**	1.90	0.0173
lego	23.54	0.8581	0.74	0.0198
mutant	25.96	0.9283	1.16	0.0128
standup	26.20	0.9452	1.72	0.0157
trex	27.12	0.9338	1.15	0.0143
平均	**26.01**	**0.9255**	**1.26**	**0.0141**

结果分析与讨论

1. **Deformation 方案的平均效果显著优于 Straightforward 方案：**
   - 平均 PSNR：29.11 dB vs 26.01 dB（提升约 3.09 dB）
   - 平均 SSIM：0.9527 vs 0.9255（提升约 0.027，约 2.9%）
   - 说明将场景分解为静态与动态两部分的建模策略更有效，静态背景提供稳定基础，动态残差仅需学习变化部分，降低了优化难度。
2. 训练稳定性差异：
   - Deformation 的 PSNR 标准差（2.98）显著高于 Straightforward（1.26），说明其在不同时刻/视角的表现波动较大，但峰值质量更高。
   - Straightforward 方案整体更稳定（PSNR/SSIM 的 std 更小），但在大多数场景上仍低于 Deformation，说明仅将时间作为连续变量直接回归往往难以充分建模复杂动态。

3. **朴素方法的共同局限：**

   - 对时间的连续回归（无论直接注入还是残差预测）缺乏语义级约束，导致**时序一致性差、容易产生闪烁/模糊**（尤其在遮挡/快速形变区域）。
   - 训练对超参数（如 weight decay、learning rate）极其敏感，容易退化为"发白/输出均值场"的塌缩解（如前文 5.1 所述）。

   基于上述局限，本文提出在特征空间进行语义一致插值的 Novel Dynamic NeRF 方案（见下节 5.2），从根本上避免逐点时间回归，进行实验验证方案的可行性。

4.2 Novel Dynamic NeRF based on vggt:

动机：

传统动态 NeRF 将时间 t 作为额外输入回归场函数，或用静态/动态双网络预测偏移。由于每个 3D 点（或采样点）被独立回归，缺乏“语义级约束”，容易出现局部不一致：同一物体不同部位预测位移不协调，导致边界模糊与闪烁。受到动画原理中的关键帧插值技术的启发，我们提出了一种基于VGGT模型的动态场景生成技术方案，经过测试取得了良好的效果，具有可推广性。

实现方法：跨时刻 token 匹配 + 语义一致插值

设有两组时刻 t_0, t_1 的多视角照片，且相机参数与视角一一对应。我们在 **Aggregator 的 token 特征空间**进行插值，得到中间时刻 $t_i \in [t_0, t_1]$ 的 token，再送入后续注意力层与 head 输出三维结果（点云/网格/辐射场参数）。

步骤：

1. **局部匹配（语义对齐）：**对 t_0 中某 patch token，在 t_1 同位置附近（如 11×11 邻域）用余弦相似度搜索最相近 token，得到对应关系，避免仅按网格坐标硬匹配导致漂移。

2. **线性插值：**对匹配到的两端 token 特征做 $\mathbf{z}(t_i) = (1-\alpha)\mathbf{z}(t_0) + \alpha\mathbf{z}(t_1), \quad \alpha = \frac{t_i - t_0}{t_1 - t_0}$. 同时对 2D token 位置也插值，得到连续位置以提升运动的平滑性。

3. **双线性泼溅（splatting）+ 权重归一化：**将连续位置的插值 token 分配到邻近四个网格点，维护 Weight Grid 并做归一化，避免因覆盖不均导致能量漂移。

demo与效果分析

dynamic的模型代码在tnerf/model/dynamic_tnerf.py程序中实现，我们在VGGT的demo代码的基础上进行一定的ui修改。运行时将VGGT模型的预训练参数放置在vggt/model_weights目录中，运行vggt/demo_gradio.py，在页面中点击加载历史记录，就可以看到下面在两个时刻仅仅分别传入一张图片（无相机内外参）的预测示例：

如图 1 所示，绿色为我们传入的t0与t1时刻的图像，选择alpha分别为0,0.4,0.7,1.0预测出了中间的四个点云，可以发现，成功且合理准确地预测出中间状态。

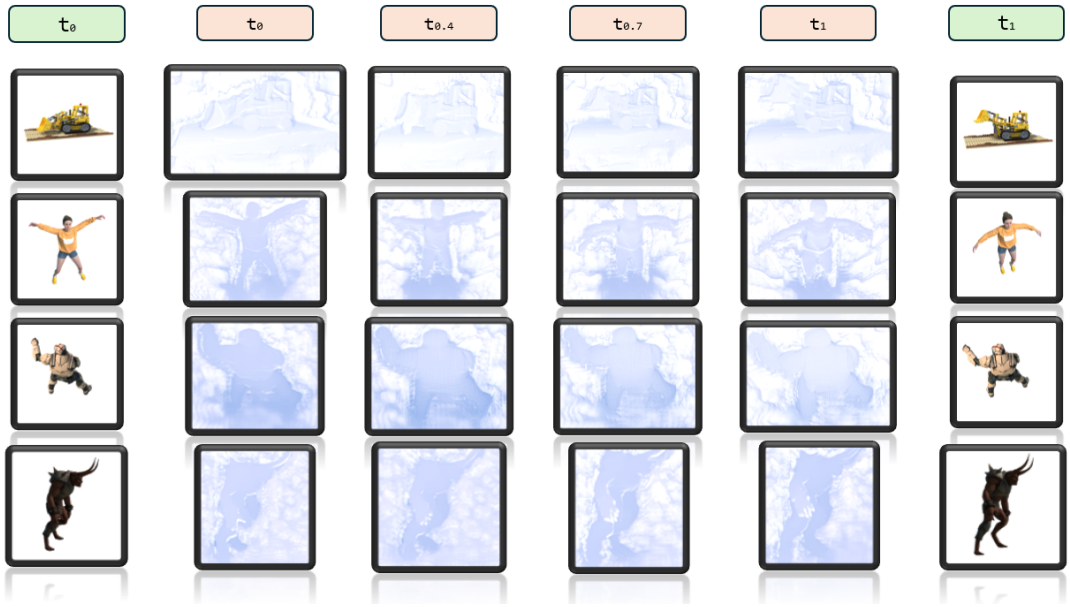

图 1: Dynamic NeRF 的中间时刻点云插值示例（$\alpha = 0, 0.4, 0.7, 1.0$）

意义：把动态从“点级回归”提升为“语义级一致生成”

1. 该方案的关键优势在于：插值发生在 ViT 语义特征空间，token 表示的是”物体部件/语义片段”，因此运动一致性更强。

2. 由于VGGT本身的优势，此方法能够很适应不同数目的图像传入，即使只有一张图片也能获得很好的处理效果，天然更适配稀疏视角。

3. 在处理时间上，由于方法路线的差别，此方法也只需要一个推理的时间，实测传入两个时刻各一张图片，从插值到生成[0,0.1,0.2,...,0.9,1]时刻的点云，所需时间为5分钟左右。

相较逐点回归（容易让相邻点的位移互相矛盾），语义 token 插值在生成中间时刻时更稳定、更清晰。

—

5. 局限与未来工作

1. **训练数据与监督形式受限：**本文主要依赖 NeRF-MAE 的 RGB/Sigma 场数据进行点监督训练。该监督对”场函数拟合”较直接，但与真实应用中的图像监督仍存在 domain gap；同时数据覆盖的场景类型有限，可能影响跨域泛化能力。

2. **训练规模与算力预算不足：**出于时间和资源的限制，我们没有充分训练并验证T-NeRF的实现思路，仅完成代码框架与小规模训练验证(尽管看起来我觉得很promising www)

3. **创新方案的dynamic nerf：**我们已经验证了方法的可行性，由于时间的限制，目前仅实际实现了动态点云的生成，还可以将此方法迁移到训练好的模型上，实现任意时刻的体渲染。

参考文献

[1] Muhammad Zubair Irshad, Sergey Zakharov, Vitor Guizilini, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. Nerf-mae: Masked autoencoders for self-supervised 3d representation learning for neural radiance fields. In *European Conference on Computer Vision (ECCV)*, 2024.

[2] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer, 2025.