arXiv:2503.11651v1 [cs. CV] 2025年3月14日

VGGT :视觉几何基础变换器

王建元[1,2]陈明浩[1,2]尼基塔·卡拉耶夫[1,2]安德烈亚·韦达利[1,2]
Christian Rupprecht[1] David Novotny[2]

[1]牛津大学视觉几何学小组 [2]Meta AI

图1。**VGGT**是一个大型前馈变换器，采用最小化三维感应偏置的架构，基于海量三维标注数据训练而成。该系统可同时处理数百张图像，仅需不到一秒即可预测所有图像的相机、点云图、深度图及点轨迹，其性能通常优于无需额外处理的优化方案。

摘要

我们推出了VGGT，这是一种前馈神经网络，能够直接从单个、少量或数百个场景视角中推断出所有关键3D属性，包括相机参数、点云图、深度图和3D点轨迹。该方法在3D计算机视觉领域实现了重大突破——传统模型通常仅限于单一任务的约束与专业化。该网络不仅简单高效，能在1秒内完成图像重建，其性能仍优于需要通过视觉几何优化技术进行后处理的替代方案。在相机参数估计、多视角深度估计、密集点云重建和3D点追踪等多任务中，该网络均取得业界领先成果。我们还证明，使用预训练VGGT作为特征骨干可显著提升下游任务表现，例如非刚性点追踪和前馈新视角合成。相关代码与模型已公开发布于https://github.com/facebookresearch/vggt。

1. 介绍

我们研究如何利用前馈神经网络从一组图像中估计场景的三维属性。传统三维重建主要采用视觉几何方法，通过束调整（BA）[45]等迭代优化技术实现。机器学习常作为重要补充手段，解决仅靠几何方法无法完成的任务，例如特征匹配和单目深度预测。如今，机器学习与视觉几何的融合日益紧密，像VGGSfM[125]这类最先进的结构从运动（SfM）方法，已通过可微分束调整技术实现了端到端的整合。尽管如此，视觉几何在三维重建中仍起着关键作用，这导致了复杂度和计算成本的增加。

随着网络变得越来越强大，我们提出一个问题：最终，三维任务是否可以*直接*通过神经网络解决，几乎完全避免几何后处理。最近的贡献如DUSt3R[129]及其演化

MASt3R[62]在该方向上已显示出令人鼓舞的结果，但这些网络仅能同时处理两张图像，并依赖后处理来重建更多图像，通过融合成对重建结果实现。

在本文中，我们进一步实现了后处理阶段无需优化三维几何结构的目标。为此，我们引入了*视觉几何基础变换器*（VGGT），这是一种前馈神经网络，能够从单个、少量甚至数百个场景输入视图中进行三维重建。 VGGT 仅需一次前向运算，数秒内即可预测出完整的三维属性集，包括相机参数、深度图、点云图和三维点轨迹。值得注意的是，即使不进行后续处理，其性能也常优于基于优化的替代方案。这与DUSt3R、MASt3R或VGGSfM等方法形成鲜明对比——这些方法仍需耗费大量资源进行迭代后优化才能获得可用结果。

我们还证明，无需为3D重建设计专用网络。相反，VGGT 基于一个相当标准的大变压器[119]，没有特殊的3D或其他归纳偏置（除了在帧级和全局注意力之间交替），但通过大量带有3D标注的公开数据集进行训练。因此， VGGT 与自然语言处理和计算机视觉领域的大型模型（如GPTs[1、29、148]、CLIP[86]、DINO[10、78]和Stable Diffusion[34]）采用相同的架构。这些模型已成为可微调以解决新特定任务的通用骨干网络。类似地，我们证明 VGGT 计算的特征能显著增强动态视频中的点追踪和新型视角合成等下游任务。

近期出现了多个大型3D神经网络的实例，包括DepthAnything[142]、MoGe[128]和 LRM [49]。然而这些模型仅专注于单一3D任务，例如单目深度估计或新视角合成。相比之下， VGGT 采用共享主干网络同时预测所有感兴趣的3D属性。我们证明，*学习*预测这些相互关联的3D属性能提升整体精度，尽管可能存在冗余。同时我们发现，在*推理*过程中，通过分别从深度和相机参数预测中提取点云图，相比直接使用专用点云图头能获得更优精度。

综上所述，我们的研究贡献如下：(1)我们提出 VGGT ——一种大型前馈变换器，能够基于单张、少量甚至数百张场景图像，在数秒内预测所有关键三维属性，包括相机本征参数、外在参数、点云图、深度图及三维点轨迹。(2)实验证明 VGGT 的预测结果具有直接应用价值，其性能不仅极具竞争力，通常优于采用缓慢后处理优化技术的现有最先进方法。(3)我们还证实，当与BA后处理技术结合使用时，VGGT 在所有方面均取得顶尖水平，即便与专注于特定3D任务子集的方法相比，其质量也往往显著提升。

我们将代码和模型公开在https://github.com/facebookresearch/vggt上。我们相信，这将促进这一方向的进一步研究，并为快速、可靠和通用的3D重建提供新的基础，从而造福计算机视觉社区。

2. 相关工作

结构从运动是计算机视觉领域的经典问题[45，77，80]，其核心在于通过不同视角拍摄的静态场景图像，估计相机参数并重建稀疏点云。传统的SfM流程[2，36，70，94，103，134]包含图像匹配、三角测量和束调整等多个阶段。 COLMAP [94]是基于传统流程的主流框架。近年来，深度学习显著提升了SfM流程的多个组件，其中关键点检测[21，31，116，149]与图像匹配[11，67，92，99]是两大核心研究方向。最近的方法[5，102，109，112，113，118，122，125，131，160]探索了端到端可微分的SfM，其中VGGSfM[125]开始在具有挑战性的摄影旅游场景中超越传统算法。

多视点立体旨在通过多个重叠图像密集重建场景的几何结构，通常假设已知相机参数（这些参数常通过SfM进行估计）。MVS方法可分为三类：传统手工方法[38，39，96，130]、全局优化方法[37，74，133，147]以及基于学习的方法[42，72，84，145，157]。与SfM类似，基于学习的MVS方法近期取得了显著进展。其中DUSt3R[129]和MASt3R[62]直接从一对视图中估计对齐的密集点云，类似于MVS但无需相机参数。部分并行研究[111，127，141，156]尝试用神经网络替代DUSt3R的测试时优化，但这些尝试仅达到次优或与DUSt3R相当的性能。相比之下， VGGT 在性能上大幅超越DUSt3R和MASt3R。

任意点追踪最初由Particle Video[91]提出，并在深度学习时代被PIP[44]重新提出，旨在跨视频序列（包括动态运动）追踪兴趣点。给定一段视频和若干二维查询点，任务是预测这些点在所有其他帧中的二维对应关系。TAP-Vid[23]为此任务提出了三个基准，并在tapir[24]中改进了简单的基线方法。CoTracker[55，56]利用不同点之间的相关性实现遮挡追踪，而DOT[60]则实现了密集遮挡追踪。最近， TAPTR [63]提出了

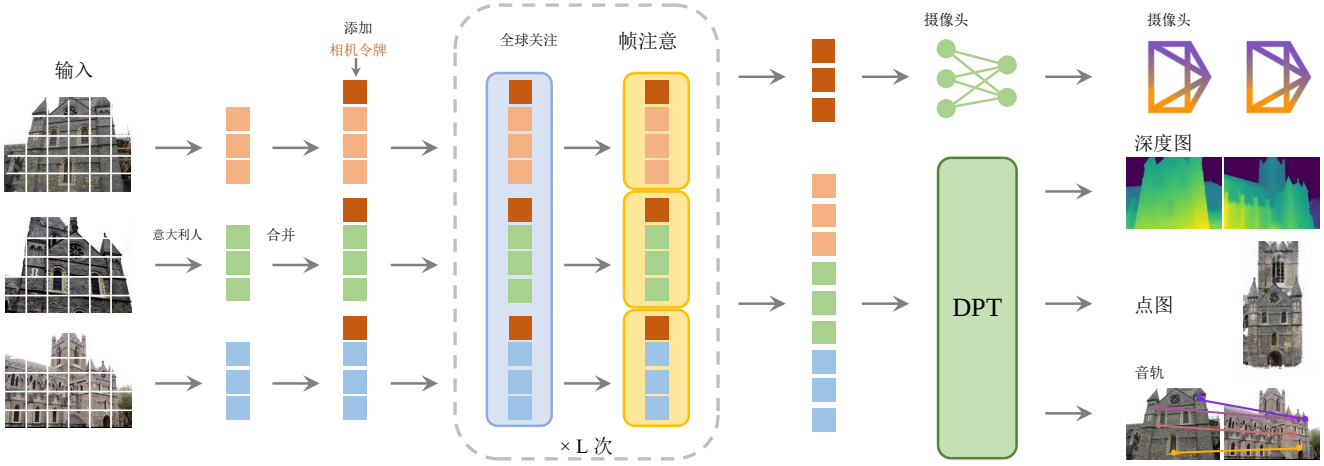

图2.**架构概述。**我们的模型首先通过DINO将输入图像补丁化为标记，并附加相机标记用于相机预测。随后在帧级和全局自注意力层之间交替操作。相机头对相机的外在和内在参数进行最终预测，而DPT[87]头则负责任何密集输出。

针对该任务的端到端变换器，以及LocoTrack[13]将常用点特征扩展至邻近区域。这些方法均为专用点追踪器。本文证明， VGGT 的特征与现有点追踪器结合时，可实现最先进的追踪性能。

3. 方法

我们介绍 VGGT ，一种大型变压器，它以一组图像作为输入，并生成多种3D量作为输出。我们首先在第3.1节介绍问题，接着在第3.2节介绍我们的架构及其预测头在第3.3节，最后在第3.4节介绍训练设置。

3.1. 问题定义与符号

输入为序列$(I_i)_{i=1}^N$ N个 RGB 图像$I_i \in \mathrm{R}^{3\times H\times W}$，观察相同的 3D 场景。 VGGT 的变换器是一个函数，将该序列映射到一组相应的 3D 注释，每帧一个：

$$f\left((I_i)_{i=1}^N\right) = (\mathbf{g}_i,\ D_i, P_i,\ Ti)_{i=1}^N . \qquad (1)$$

因此，该变换将每张图像I_i映射到其相机参数$\mathbf{g}_i \in \mathrm{R}^9$（本征参数与外在参数）、深度图$D_i \in \mathrm{R}^{H\times W}$、点图$P_i \in \mathrm{R}^{3\times H\times W}$，以及一个$C$维特征的网格$T_i \in \mathrm{R}^{C\times H\times W}$用于点追踪。接下来我们将解释这些参数的定义方式。

对于**相机参数**$\mathbf{g}_i$，我们采用[125]中的参数化方法，并设定$\mathbf{g}=[\mathbf{q},\ \mathbf{t},\ \mathbf{f}]$，该参数由旋转四元数$\mathbf{q} \in \mathrm{R}^4$、平移向量$\mathbf{t} \in \mathrm{R}^3$以及视场$\mathbf{f} \in \mathrm{R}^2$的拼接构成。我们假设相机主点位于图像中心，这在SfM框架中是常见的做法[95，125]。

我们将图像I_i的定义域记为I（I_i）={1，...，H} × {1，...，W}，*即*像素位置集合。**深度图**D_i将每个像素位置$\mathbf{y} \in I$（Ii）与其对应的深度值D_i（$\mathbf{y}$）$\in \mathrm{R}^+$相关联，该值来自第i个相机的观测。**点图**P_i将每个像素与其对应的三维场景点P_i（$\mathbf{y}$）$\in \mathrm{R}^3$相关联。重要的是，与DUSt3R [129]类似，点图具有*视点不变性*，即三维点P_i（$\mathbf{y}$）在第一相机$\mathbf{g}_1$的坐标系中定义，我们将该坐标系作为世界参考系。

最后，对于**关键点跟踪**，我们采用类似[25，57]的轨迹-任意点方法。具体而言，给定查询图像I_q中的固定查询图像点$\mathbf{y}_q$，网络会输出一条轨迹$T^*(\mathbf{y}_q) = (\mathbf{y}_i)_{i=1}^N$由所有图像$I_i$中对应的二维点$\mathbf{y}_i \in \mathrm{R}^2$构成。

注意，上述变压器f并不直接输出轨迹，而是输出$T_i \in \mathrm{R}^{C\times H\times W}$，这些用于跟踪。跟踪任务被委托给一个单独的模块，该模块在第3.3节中描述，其实现了一个function $\mathcal{T}((\mathbf{y}_j)_{j=1}^M, (T_i)_{i=1}^N) = ((\hat{\mathbf{y}}_{j,i})_{i=1}^N)_{j=1}^M$. 它接收查询点$\mathbf{y}_q$和由变换器$f$输出的密集跟踪特征$T_i$，然后计算轨迹。两个网络$f$和$T$采用端到端联合训练。

预测顺序。输入序列中图像的顺序是任意的，但第一个图像被选为参考帧。该网络架构设计为对除第一个帧外的所有帧具有置换等变性。

过度完备的预测。值得注意的是，并非所有由 VGGT 预测的量都是独立的。例如，如DUSt3R[129]所示，相机参数$\mathbf{g}$可以从不变点图P中推断出来，例如通过求解透视n点（PnP）问题[35，61]。

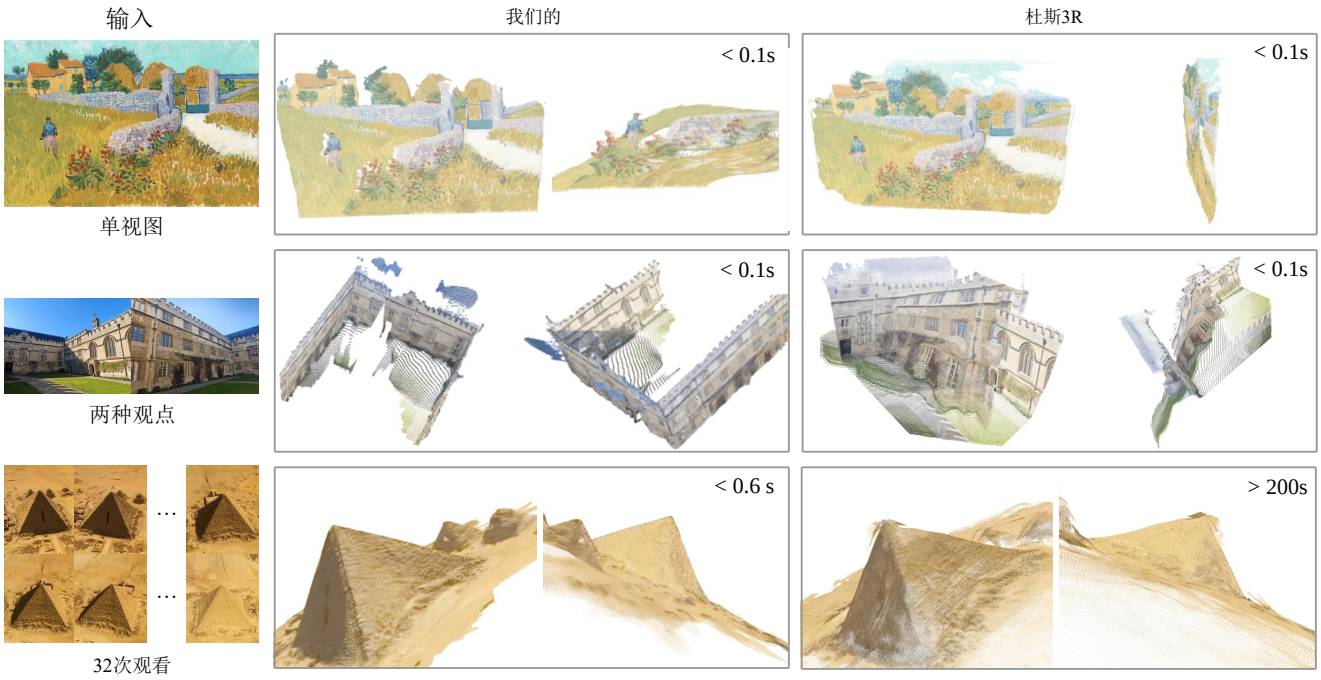

图3.**我们预测的三维点与DUSt3R在真实场景图像中的定性对比。**如上图所示，我们的方法成功预测了油画的几何结构，而DUSt3R仅能预测出略微扭曲的平面。第二行展示了我们的方法如何从两张无重叠图像中准确还原三维场景，而DUSt3R则未能成功。第三行呈现了一个具有重复纹理的高难度案例，但我们的预测仍保持高质量。由于超过32帧的样本会导致DUSt3R内存不足，我们未包含超过该数量的示例。

此外，深度图可通过点云数据和相机参数推导得出。但正如我们在第4.5节所述，训练时 VGGT 显式预测*所有*上述参数能显著提升性能，即便这些参数间存在闭式关系。而在推理阶段，研究发现：相较于直接使用专用点云分支，将独立估算的深度图与相机参数相结合，能生成更精确的三维点云数据。

3.2. 特征脊柱

基于近期3D深度学习的研究[53，129，132]，我们设计了一个具有最小化3D归纳偏置的简单架构，使模型能够从大量3D标注数据中学习。具体而言，我们将模型f实现为一个大型Transformer[119]。为此，每个输入图像I首先通过DINO[78]被分割为一组K个标记[1] $\mathrm{t}^{I} \in \mathrm{R}^{K\times C}$。所有帧的图像标记组合集，*即*$\mathrm{t}^{I}=\cup_{i=1}^{N}\{\mathrm{t}_{Ii}\}$随后通过主网络结构进行处理，该结构交替包含帧级和全局自注意力层。

交替注意力机制。我们通过引入交替注意力机制对标准Transformer架构进行了微调（AA）使变换器在每帧内及全局范围内交替聚焦。具体而言，*帧内自注意力*分别关注每帧内的标记tIk，而*全局自注意力*则联合关注所有帧内的标记t^{I}。这种设计在整合不同图像信息与对每帧内标记激活值进行归一化之间取得了平衡。默认情况下，我们采用L=24层的全局和帧内注意力机制。在第4节中，我们将展示AA架构带来的显著性能提升。需注意的是，我们的架构未采用任何交叉注意力层，仅使用自注意力层。

[1]标记数量取决于图像分辨率。

3.3. 预测头

在此，我们描述了f如何预测相机参数、深度图、点图和点轨迹。首先，对于每张输入图像I_i，我们通过添加额外的相机标记tgi $\in \mathrm{R}^{1\times C'}$和四个配准标记[19]tRi $\in \mathrm{R}^{4\times C'}$来增强相应的图像标记tIi。将$(\mathrm{tIi}, \mathrm{tgi}, \mathrm{tRi}j)_{i=1}^{N}$随后传递至AA变压器，生成输出标记$(\hat{}_{It\,i}, \hat{}_{\mathbf{gt\,i}}, \hat{}_{Rt\,i})_{i=1}^{N}$。在此，第一帧的相机标记和寄存器标记（tg1：$=\bar{}\mathrm{t}^{\mathbf{g}}$，tR1 := -tR）被设置为一组不同的可学习标记$\mathrm{t}^{\mathbf{g}}$、t^{R}，而非那些在所有其他帧中（$\mathrm{t}_{i}^{\mathbf{g}}:=\mathrm{t}^{-\mathbf{g}}$，$\mathrm{t}_{i}^{R}:=\mathrm{t}^{-R}$，$i \in [2, \ldots, N]$），

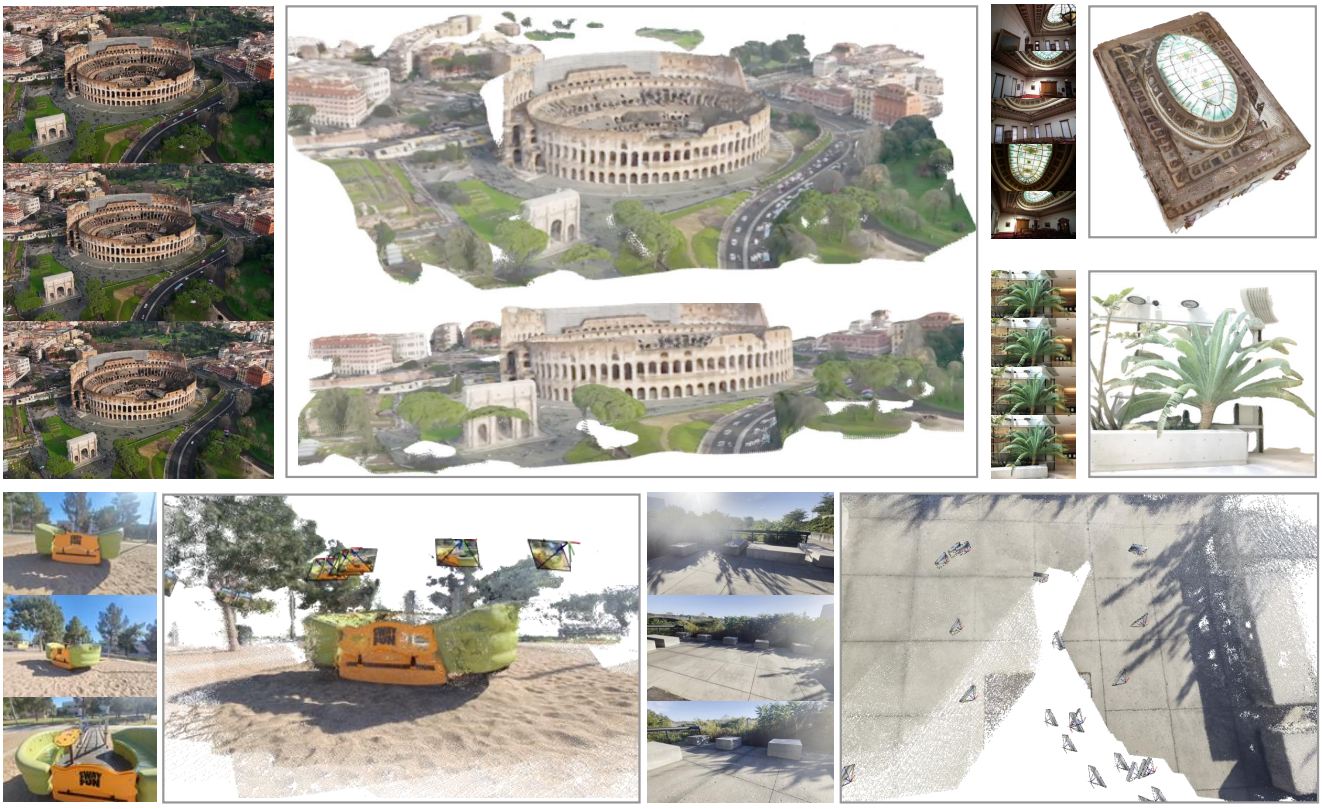

图4.**点图估计的附加可视化效果。**摄像机截头体展示了估计的相机姿态。通过我们的交互式演示，可获得更优质的可视化效果。

这些特征同样具有可学习性。这使得模型能够区分首帧与其他帧，并在首台相机的坐标系中呈现三维预测结果。需要说明的是，优化后的相机与配准标记现在具有帧特异性——这是因为我们的AA转换器包含逐帧自注意力层，使转换器能将相机与配准标记与同一图像中的对应标记进行匹配。遵循常规做法，输出配准标记 $\hat{}_{Rt_i}$被舍弃，而 $\hat{}_{It\_i}$、 $\hat{}_{\mathbf{gt_i}}$则用于预测。

坐标系。如上所述，我们预测相机、点云图和深度图均在第一台相机的坐标系$\mathbf{g}_1$中。因此，第一台相机的相机外参输出被设为单位矩阵，*即*第一旋转四元数为$\mathbf{q}_1$= [0,0,0,1]，第一平移向量为$\mathbf{t}_1$= [0,0,0]。需要指出的是，特殊相机和寄存器标记 tg1 :=$\bar{\mathbf{t}}^{g}$、tR1 :=$\bar{\mathbf{t}}^{R}$可使Transformer识别第一台相机。

相机预测。相机参数$(\hat{\mathbf{g}}^i)_{i=1}^N$根据输出的相机标记（$\hat{}_{\mathbf{gti}}$）进行预测$)_{i=1}^N$通过增加四个自注意力层并接续一个线性层，最终形成*摄像头头部*，用于预测相机的内在特性。

sics和外在因素。

密集预测。输出图像标记 $\hat{}_{It_i}$用于预测密集输出，*即*深度图D_i、点图P_i和跟踪特征T_i。具体而言， $\hat{}_{It_i}$首先通过DPT层[87]转换为密集特征图$F_i \in \mathrm{R}^{C'' \times H \times W}$。每个$F_i$随后通过3 × 3卷积层映射到对应的深度图和点图$D_i$与$P_i$。此外，DPT头还输出密集特征$T_i \in \mathrm{R}^{C \times H \times W}$，作为跟踪头的输入。我们还预测随机不确定性[58，76] $\Sigma_{Di} \in \mathrm{R}_{+}^{H \times W}$与 $\Sigma_{Pi} \in \mathrm{R}_{+}^{H \times W}$分别对应每个深度和点云图。如第3.4节所述，不确定性图被用于损失函数中，训练后其数值与模型对预测结果的置信度成正比。

跟踪。为实现跟踪模块T，我们采用CoTracker2架构[57]，该架构以密集跟踪特征T_i作为输入。具体而言，给定查询图像I_q中的查询点$\mathbf{y}_j$（训练时始终设q= 1，但其他图像均可作为查询点），跟踪头T会预测一组二维点$\mathcal{T}((\mathbf{y}_j)_{j=1}^M, (T_i)_{i=1}^N) = ((\hat{\mathbf{y}}_{j,i})_{i=1}^N)_{j=1}^M$ 在所有图像I_i中，这些图像对应于与$\mathbf{y}$相同的3D点。为此，首先对查询图像的特征图T_q进行双线性采样

查询点$\mathbf{y}_j$以获取其特征。该特征随后与所有其他特征图T_i、i进行相关性分析$\neq q$以获取一组相关性图。这些图随后通过自注意力层进行处理，以预测最终的二维点 $\hat{}$ **yi**，它们均与$\mathbf{y}_j$相对应。需要注意的是，与VGGSfM[125]类似，我们的跟踪器不假设输入帧的任何时间顺序，因此可应用于任何一组输入图像，而不仅限于视频。

3.4. 训练

训练损失。我们采用多任务损失对 VGGT 模型f进行端到端训练：

$$\mathcal{L} = \mathcal{L}_{\text{camera}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pmap}} + \lambda \mathcal{L}_{\text{track}}. \qquad (2)$$

我们发现相机损失（$\mathcal{L}_{\text{camera}}$）、深度损失（$\mathcal{L}_{\text{depth}}$）和点云损失（$\mathcal{L}_{\text{pmap}}$）的范围相近，无需相互加权。跟踪损失$\mathcal{L}_{\text{track}}$的权重系数$\lambda$=0.05。我们逐一描述每个损失项。

相机损失$\mathcal{L}_{\text{camera}}$监督相机 $\hat{}$ **g**：$\mathcal{L}_{\text{camera}}=\sum_{i=1}^{N}$ // $\hat{}$ $\mathbf{g}_i$ − $\mathbf{g}_i$ //$_\epsilon$，通过Huber损失$|\cdot|_\epsilon$将预测摄像头 $\hat{}$ **gi**与真实值$\mathbf{g}_i$进行比较。

深度损失$\mathcal{L}_{\text{depth}}$遵循DUSt3R[129]，并实现了随机不确定性损失[59，75]，通过预测深度 $\hat{}$ D_i与真实深度D_i之间的差异，以及预测不确定性图 $\hat{}$**Σ** $_{Di}$进行加权。与DUSt3R不同，我们还应用了基于梯度的项，该方法在单目深度估计中被广泛使用。因此，深度损失为$\mathcal{L}_{\text{depth}}=\sum_{i=1}^{N}$ // **Σ** $_{Di}$ ⊙（$\hat{}$ D_i − D_i）//+// **Σ** $_{Di}$ ⊙（∇ $\hat{}$ D_i − ∇ D_i）// −/ log$_{\square}$ $_{Di}$，其中⊙表示通道广播的元素乘积。点图损失的定义类似，但使用点图不确定性$_{\square}$ $^{P}_{i}$：$\mathcal{L}_{\text{pmap}}=\sum_{i=1}^{N}$ //$_{\square}$ $_{Pi}$ ⊙（$\hat{}$ P_i − P_i）//+//$_{\square}$ $_{Pi}$ ⊙（∇ $\hat{}$ P_i − ∇ P_i）// −/ log$_{\square}$ $_{Pi}$。

最后，跟踪损失由L*track*=给出$\sum_{j=1}^{M}\sum_{i=1}^{N}$ // $\mathbf{y}_{j,i}$ − $\hat{}$ $\mathbf{y}_{j,i}$ //。此处外和遍历查询图像I_q，$\mathbf{y}_j$，$_i$为$\mathbf{y}_j$在图像I_i中的真实对应点，而 $\hat{}$ **yj**，$_i$是通过应用获得的相应预测结果$\mathcal{T}((\mathbf{y}_j)_{j=1}^{M},(T_i)_{i=1}^{N})$ 关于跟踪模块。此外，根据CoTracker2[57]的方法，我们应用可见性损失（二元交叉熵）来估计某点在给定帧中是否可见。

真实坐标归一化。当我们缩放场景或改变其全局参考系时，场景的图像完全不受影响，这意味着任何此类变体都是3D重建的合法结果。我们通过数据归一化消除这种歧义，从而做出规范选择，并让Transformer输出该特定变体。我们遵循[129]的方法，首先将所有量值表示为第一摄像机$\mathbf{g}_1$的坐标系。然后，我们计算所有3D点在将点云P映射到原点，并使用此比例尺对相机平移**t**、点云P和深度图D进行归一化。重要的是，与[129]不同，我们*不*对Transformer输出的预测结果进行此类归一化；相反，我们强制其从训练数据中学习我们选择的归一化方法。

实现细节。默认情况下，我们分别采用L= 24层全局注意力和帧级注意力。该模型总参数量约为12亿。我们通过AdamW优化器对训练损失（2）进行16万次迭代训练。采用余弦学习率调度器，峰值学习率为0.0002，预热8000次迭代。每个批次从随机训练场景中随机抽取2-24帧。输入帧、深度图和点图均调整至最大518像素尺寸，宽高比随机分布在0.33到1.0之间。同时对帧进行随机颜色抖动、高斯模糊和灰度增强处理。训练在64个A100 GPU上持续九天，采用梯度范数剪切阈值1.0以确保训练稳定性，并通过bfloat16浮点精度和梯度检查点技术提升GPU内存和计算效率。

训练数据。该模型使用大量多样化的数据集进行训练，包括：Co3Dv2[88]、BlendMVS[146]、DL3DV[69]、MegaDepth[64]、Kubric[41]、WildRGB[135]、ScanNet[18]、HyperSim[89]、Mapillary[71]、Habitat[107]、Replica[104]、MVS-Synth[50]、PointOdyssey[159]、Virtual KITTI [7]、Aria Synthetic Environments[82]、Aria Digital Twin[82]，以及一个类似于Objaverse[20]的艺术家创作资产合成数据集。这些数据集涵盖多种领域，包括室内和室外环境，并包含合成和真实场景。这些数据集的3D标注来源于多种渠道，如直接传感器捕获、合成引擎或SfM技术[95]。我们的数据集组合在规模和多样性上与MASt3R[30]大致相当。

4. 实验

本节通过多任务对比，将我们的方法与现有最先进方法进行比较，以验证其有效性。

4.1. 相机姿态估计

我们首先在CO3Dv2[88]和RealEstate10K[161]数据集上评估相机姿态估计方法，如表1所示。参照[124]，我们随机选取每个场景的10张图像，采用结合 RRA 和RTA 的标准指标AUC@30进行评估。其中 RRA（相对旋转精度）和RTA（相对平移精度）分别计算每对图像在旋转和平移方向上的相对角度误差。这些角度-

方法	Re10K（*未见*）AUC@30 ↑	CO3Dv2 AUC@30 ↑	时间
Colmap+SPSG [92]	45.2	25.3	∼ 15s
PixSfM [66] PoseDiff [124] DUSt3R [129] MASt3R [62] VGGSfM v2 [125]	49.4 48.0 67.7 76.4 78.9	30.1 66.5 76.7 81.8 83.4	> 20s ∼ 7s ∼ 7s ∼ 9s ∼ 10s
MV-DUSt3R [111] ‡	71.3	69.5	∼ 0.6s
CUT3R [127] ‡	75.3	82.8	∼ 0.6s
[156] ‡	78.8	83.3	∼ 0.5s
Fast3R [141] ‡	72.7	82.5	∼ **0.2s**
我们的（前馈）	85.3	88.2	∼ **0.2s**
我们的（含BA）	**93.5**	**91.8**	∼ 1.8s

表1.**Camera Pose Estimation on RealEstate10K [161]and CO3Dv2 [88]** with 10 random frames. All metrics the higher the better. None of the methods were trained on the Re10K dataset. Runtime were measured using one H100 GPU. Methods marked with ‡ represent concurrent work.

已知GT照相机	方法	Acc.↓	比较↓	总体↓
✓	Gipuma [40]	**0.283**	0.873	0.578
✓	MVSNet [144]	0.396	0.527	0.462
✓	苹果酒[139]	0.417	0.437	0.427
✓	PatchmatchNet [121]	0.427	0.377	0.417
✓	MASt3R [62]	0.403	0.344	0.374
✓	GeoMVSNet [157]	0.331	**0.259**	**0.295**
✗	DUSt3R [129]	2.677	0.805	1.741
✗	我们的	**0.389**	**0.374**	**0.382**

表2.**DTU 上密集MVS估计[51]数据集。**表中上半部分为已知真实相机的方法，下半部分为未知真实相机的方法。

方法	Acc.↓	比较↓	总体↓	时间
杜斯3R	1.167	0.842	1.005	∼ 7s
MASt3R	0.968	0.684	0.826	∼ 9s
我们的（点）	0.901	0.518	0.709	∼ 0.2s
我们的（深度+相机）	**0.873**	**0.482**	**0.677**	∼ 0.2s

表3.**ETH3D上的点云估计[97]**。DUSt3R和MASt3R采用全局对齐，而我们的方法是前馈的，因此更快。行*我们（点云）*表示直接使用点云头部的结果，而*我们（深度+相机）*表示结合深度地图头部和相机头部构建点云。

随后对分类错误进行阈值处理以确定准确率评分。AUC是 RRA 与RTA在不同阈值下最小值的准确率-阈值曲线下面积。表1中的（可学习）方法已在Co3Dv2数据集上训练，**未**在RealEstate10K数据集上训练。我们的前馈模型始终优于竞争方法-

方法	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑
SuperGlue [92]	16.2	33.8	51.8
LoFTR [105]	22.1	40.8	57.6
DKM [32]	29.4	50.7	68.3
CasMTR [9]	27.1	47.0	64.4
罗马[33]	31.8	53.4	70.9
我们的	**33.9**	**55.2**	**73.4**

表4.**ScanNet-1500上的双视图匹配比较[18，92]**。尽管我们的跟踪头并非专为双视图设置设计，但其性能优于当前最先进的双视图匹配方法Roma。以AUC衡量（数值越高越好）。

在两个数据集上所有指标的ODS（优化后数据集），包括那些采用计算成本高昂的优化后步骤的指标，例如DUSt3R/MASt3R的全局对齐和VGGSfM的束调整，通常需要超过
10秒。相比之下， VGGT*仅以前馈方式运行即可实现更优性能*，在相同硬件上仅需0.2秒。与并发研究[111、127、141、156]（以‡标示）相比，我们的方法展现出显著的性能优势，其速度与最快变体Fast3R [141]相当。此外，我们的模型在RealEstate10K数据集上的性能优势更为突出，而表1中展示的任何方法均未在此数据集上训练。这验证了 VGGT 的优越泛化能力。

我们的研究结果还表明，通过将 VGGT 与视觉几何优化方法（如BA）相结合，可以进一步提升其性能。具体而言，使用BA对预测的相机位姿和轨迹进行精细化处理，能显著提高精度。值得注意的是，我们的方法直接生成接近精确的点云/深度图，这可作为BA的良好初始化数据。这消除了[125]中BA需要进行三角测量和迭代优化的步骤，使我们的方法运行速度大幅提升（即使结合BA也仅需约2秒）。因此，尽管 VGGT 的前馈模式优于所有先前方案（无论是否采用前馈机制），但仍有改进空间，因为后优化阶段仍能带来额外优势。

4.2. 多视点深度估计

在完成MASt3R[62]后，我们进一步在 DTU [51]数据集上评估多视图深度估计结果。我们报告了标准 DTU 指标，包括准确率（预测值与真实值的最小欧氏距离）、完整性（真实值与预测值的最小欧氏距离）及其平均总体指标（*即*Chamfer距离）。在表2中，DUSt3R和我们的 VGGT 是仅有的两种无需知晓真实相机信息的方法。MASt3R通过使用真实相机对匹配点进行三角测量来生成深度图。与此同时，诸如GeoMVS-等深度多视图立体方法

图5.**刚性与动态点追踪的可视化。**上图： VGGT 的追踪模块T输出关键点轨迹，用于描述静态场景的无序输入图像集。下图：我们对 VGGT 的主干网络进行微调，以增强动态点追踪器CoTracker[56]，该追踪器处理顺序输入。

Net使用地面实况相机来构建成本量。

我们的方法在性能上大幅超越DUSt3R，整体评分从1.741降至0.382。更重要的是，其结果已达到与测试时已知真实相机数据的方法相当的水平。这种显著的性能提升很可能归功于我们模型的多图像训练方案——该方案使模型能够原生地进行多视角三角测量推理，而非像DUSt3R那样依赖临时对齐流程（后者仅对多对相机三角测量结果进行平均）。

4.3. 点图估计

我们还在ETH3D[97]数据集上比较了预测点云与DUSt3R和MASt3R的准确性。针对每个场景，我们随机抽取10帧图像。使用Umeyama[117]算法将预测点云与真实数据对齐。结果在使用官方掩码过滤无效点后呈现。我们报告了点云估计的准确率、完整率和总体（切线距离）指标。如表3所示，尽管DUSt3R和MASt3R需要进行耗时的全局对齐优化（每场景约10秒），但我们的方法在简单前馈模式下仅需0.2秒即可显著超越它们。

与此同时，与直接使用我们估算的点云图相比，我们发现深度图和相机头的预测结果（*即*通过预测的相机参数将深度图反投影为三维空间）具有更高的准确性。我们认为这是由于将复杂任务（点云图估计）分解为更简单的子问题（深度图和相机预测）所带来的优势，尽管相机、深度图和点云图在训练过程中是联合监督的。

我们在图3中展示了与DUSt3R在野外场景中的定性对比，并在图4中提供了更多示例。 VGGT 输出高质量预测结果，并具备泛化能力

ETH3D数据集	Acc.↓	比较↓	总体↓
交叉注意	1.287	0.835	1.061
仅全局自注意力	1.032	0.621	0.827
交替注意	**0.901**	**0.518**	**0.709**

表5.**ETH3D数据集上的Transformer Backbone消融研究**。我们将交替注意力架构与两种变体进行对比：一种仅采用全局自注意力机制，另一种则采用交叉注意力机制。

在处理具有挑战性的非典型场景时表现尤为出色，例如油画作品、非重叠画面，以及沙漠等具有重复或均匀纹理的场景。

4.4. 图像匹配

双视图图像匹配是计算机视觉中被广泛研究的主题[68，93，105]。它代表了刚性点跟踪的一个特例，仅限于两个视图，因此即使我们的模型并非专门针对此任务，它仍是一个合适的评估基准来衡量我们的跟踪精度。我们遵循标准协议[33，93]在ScanNet数据集[18]上进行实验，并将结果报告于表4。对于每对图像，我们提取匹配项并利用它们估计本质矩阵，随后将其分解为相对相机姿态。最终指标是通过AUC衡量的相对姿态精度。评估时，我们使用 ALIKED [158]检测关键点，并将其视为查询点$\mathbf{y}_q$。这些点随后被传递至我们的跟踪分支T以在第二帧中寻找对应点。我们采用Roma[33]的评估超参数（*例如*匹配数量、 RANSAC阈值）。尽管未明确针对双视图匹配进行训练，表4显示， VGGT 在所有基准模型中达到了最高的准确率。

4.5. 消融研究

特征主干网络。我们首先通过对比验证所提出的交替注意力机制的有效性

w. L相机	w. *L深度*	w. *L型轨道*	Acc.↓	比较↓	总体↓
✗	✓	✓	1.042	0.627	0.834
✓	✗	✓	0.920	0.534	0.727
✓	✓	✗	0.976	0.603	0.790
✓	✓	✓	**0.901**	**0.518**	**0.709**

表6.**多任务学习消融研究**显示，在ETH3D数据集上，同时进行相机、深度和轨迹估计的训练可获得点云图估计的最高准确率。

针对两种替代注意力架构进行对比：(a)*仅全局自注意力*，(b)*交叉注意力*。为确保公平比较，所有模型变体均保持相同参数数量，共使用2L个注意力层。在交叉注意力变体中，每个帧独立关注其他所有帧的标记，虽然显著增加运行时间（尤其当输入帧数量增加时），但能最大化跨帧信息融合。隐藏维度和头数量等超参数保持一致。选择点图估计准确率作为消融研究的评估指标，因其反映了模型对场景几何和相机参数的联合理解。表5结果表明，我们的交替注意力架构明显优于两种基线变体。此外，其他初步探索性实验一致显示，采用交叉注意力的架构通常表现逊于仅使用自注意力的架构。

多任务学习。我们还验证了训练单一网络同时学习多个三维量的优势，即使这些输出可能存在重叠（*例如*深度图与相机参数结合可生成点云图）。如表6所示，当不包含相机、深度或轨迹估计时，点云图估计的准确率会显著下降。值得注意的是，引入相机参数估计能明显提升点云图精度，而深度估计仅带来有限的改进。

4.6. 针对下游任务的微调

我们证明 VGGT 预训练特征提取器可复用于下游任务，具体展示了其在前馈式新视角合成和动态点追踪中的应用。

前馈新颖视角合成正快速发展[8，43，49，53，108，126，140，155]。现有方法大多以已知相机参数的图像作为输入，预测对应新视角的目标图像。我们不依赖显式3D表示，而是遵循 LVSM [53]，将 VGGT 修改为*直接*输出目标图像。但我们的输入帧*并不*假设已知相机参数。

我们遵循 LVSM 的培训与评估方案

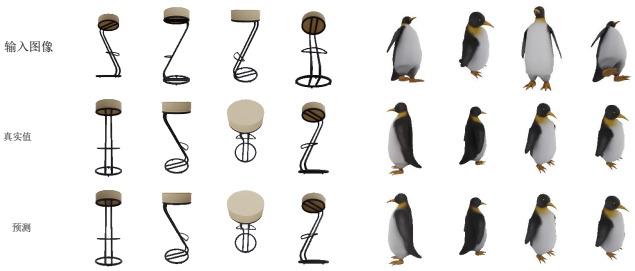

图6.**新颖视角合成的定性示例。**上排展示输入图像，中排呈现目标视角的真实图像，下排展示我们的合成图像。

方法	已知输入摄像机	尺寸	PSNR ↑	SSIM ↑	LPIPS ↓
LGM [110]	✓	256	21.44	0.832	0.122
GS-LRM [154]	✓	256	29.59	0.944	0.051
LVSM [53]	✓	256	31.71	0.957	0.027
欧氏 NVS *	✗	224	30.41	0.949	0.033

表7.**GSO[28]数据集视图合成的定量比较。**前馈式新视图合成的微调 VGGT 表明，即使不知道输入图像的相机外在和内在参数，其性能仍具有竞争力。注意*表示使用了较小的训练集（仅20%）。

紧密地，*例如*，使用4个输入视图并采用Plü使用cker射线表示目标视点。我们对 VGGT 进行了简单修改。与之前相同，输入图像通过DINO转换为标记。随后，针对目标视点，我们采用卷积层对其进行位置编码。ü将卷积层的图像映射为标记。这些标记同时包含输入图像和目标视图，经过拼接后由AA转换器处理。随后，DPT头用于回归目标视图的RGB颜色。需要特别说明的是，我们*并未*输入Plü对源图像进行克尔射线处理。因此，该模型未获得这些输入帧的相机参数。

LVSM 在Objaverse数据集[20]上进行训练。我们使用了一个规模约为Objaverse 20%的类似内部数据集。关于训练和评估的更多细节可参见[53]。如表7所示，尽管不需要输入相机参数且使用的训练数据量少于LVSM，我们的模型在GSO数据集[28]上仍取得了具有竞争力的结果。我们预计使用更大的训练数据集将获得更优结果。定性示例展示于图6。

动态点追踪近年来已成为极具竞争力的任务[25、44、57、136]，并作为我们学习特征的另一种下游应用。遵循标准做法，我们报告以下点追踪指标：遮挡准确率（OA），其包含遮挡预测的二元准确率；δ va$\sqrt{}$isg，包括

方法	动力学			RGB-S			戴维斯		
	AJ	δ^{x}_{avg}	OA	AJ	δ^{x}_{avg}	OA	AJ	δ^{x}_{avg}	OA
TAPTR [63]	49.0	64.4	85.2	60.8	76.2	87.0	63.0	76.1	91.1
LocoTrack [13]	52.9	66.8	85.3	69.7	83.2	89.5	62.9	75.3	87.2
BootsTAPIR [26]	54.6	68.4	86.5	70.8	83.0	89.9	61.4	73.6	88.7
CoTracker [56]	49.6	64.3	83.3	67.4	78.9	85.2	61.8	76.1	88.3
CoTracker + Ours	**57.2**	**69.0**	**88.9**	**72.1**	**84.0**	**91.6**	**64.7**	**77.5**	**91.4**

表8.**TAP-Vid基准测试中的动态点追踪结果。**虽然我们的模型并非专为动态场景设计，但通过使用预训练权重对CoTracker进行微调即可显著提升性能，这充分证明了我们所学特征的稳健性和有效性。

可见点在特定像素阈值内被准确追踪的平均比例；以及平均杰卡德系数（AJ），用于同时衡量追踪与遮挡预测的准确性。

我们通过用预训练的特征骨干替换其骨干来适配最先进的CoTracker2模型[57]。这是必要的，因为 VGGT 是在无序图像集合而非连续视频上训练的。我们的骨干预测跟踪特征T_i，这些特征替代了特征提取器的输出，并随后进入CoTracker2架构的其余部分，最终预测轨迹。我们在Kubric[41]上对整个修改后的跟踪器进行微调。如表8所示，预训练 VGGT 的集成显著提升了CoTracker在TAPVid基准测试[23]上的性能。例如，VGGT 的跟踪特征改善了δ^{i}_{vis}指标在TAPVid RGB-S数据集上的表现从78.9提升至84.0。尽管TAP-Vid基准测试包含来自不同数据源的快速动态运动视频，但我们的模型表现出色，充分证明了其特征的泛化能力，即使在未明确设计的场景中也是如此。

5. 讨论

局限性。尽管我们的方法在多种真实场景中展现出强大的泛化能力，但仍存在若干局限。首先，当前模型不支持鱼眼镜头或全景图像。其次，在输入图像存在极端旋转的情况下，重建性能会显著下降。此外，虽然模型能处理轻微非刚性运动的场景，但在涉及显著非刚性变形的场景中则表现欠佳。

然而，本方法的重要优势在于其灵活性与易适应性。通过在目标数据集上进行微调，仅需最小架构改动即可轻松克服这些局限。这种适应性显著区别于现有方法——后者通常需要在测试时进行大量重构以适配此类特定场景。

输入帧	1	2	4	8	10	20	50	100	200
时间（秒）	0.04	0.05	0.07	0.11	0.14	0.31	1.04	3.12	8.75
Mem. (GB)	1.88	2.07	2.45	3.23	3.63	5.58	11.41	21.15	40.63

表9.**不同输入帧数下的运行时与GPU内存峰值使用情况。**运行时以秒为单位，GPU内存使用量以GB为单位。

运行时间与内存。如表9所示，我们评估了特征骨干在处理不同数量输入帧时的推理运行时间及GPU内存峰值使用量。测量使用单个 NVIDIA H100 GPU配合flash attention v3[98]进行。图像分辨率为336 × 518。

我们重点分析特征骨干网络的成本，因为用户会根据具体需求和可用资源选择不同的分支组合。摄像头头部体积轻巧，通常仅占特征骨干网络运行时间的5%和GPU内存的2%。DPT头部每帧平均耗时0.03秒，占用0.2GB GPU内存。

当GPU内存充足时，单次前向传播即可高效处理多帧数据。但本模型中，帧间关联仅在特征骨干网络内处理，而DPT头则对每帧进行独立预测。因此，受限于GPU资源的用户可选择逐帧预测。我们保留这一权衡的自主权。

我们意识到，若对全局自注意力机制进行简单粗暴的实现，当处理大量标记时可能会造成严重的内存占用。通过借鉴大型语言模型（LLM）部署中的技术方案，可以有效节省资源或提升运算效率。例如，Fast3R[141]采用张量并行技术通过多GPU加速推理，该方案可直接套用于我们的模型。

图像拼接。如第3.2节所述，我们尝试了两种图像拼接方法：使用14 × 14卷积层或预训练的DINOv2模型。实验结果表明，DINOv2模型不仅性能更优，还能显著提升训练稳定性，尤其在模型初期表现突出。该模型对学习率、动量等超参数的敏感度也更低。因此，我们最终选定DINOv2作为模型中的默认拼接方案。

可微分BA。我们还探索了采用VGGSfM[125]中可微分束调整的方法。在小规模初步实验中，可微分BA展现出良好的性能。然而，其训练过程中的计算成本成为瓶颈。在PyTorch中使用Theseus[85]启用可微分BA时，通常会使每个训练步骤的运行速度降低约4倍，这

大规模训练成本高昂。虽然定制框架以加速训练可能是一种潜在解决方案，但这超出了本研究的范畴。因此，我们选择不在本工作中包含可微分BA，但我们将其视为大规模无监督训练的一个有前景的方向，因为它可以在缺乏明确3D标注的场景中作为有效的监督信号。

单视图重建。与DUSt3R、MASt3R这类需要复制图像生成配对的系统不同，我们的模型架构天生支持单图像输入。此时全局注意力机制直接转换为逐帧注意力机制。虽然模型并非专门针对单视图重建进行训练，但其表现却出人意料地出色。具体案例可参考图3和图7。我们强烈推荐您通过我们的演示版进行更直观的可视化体验。

预测归一化。如第3.4节所述，我们的方法通过计算三维点的平均欧氏距离对真实值进行归一化处理。虽然DUSt3R等方法也会对网络预测进行此类归一化，但研究发现这既不利于收敛，也无助于提升最终模型性能。更值得注意的是，这种处理方式在训练阶段往往会引入额外的不稳定性。

6. 结论

我们提出视觉几何基础Transformer（VGGT），这是一种前馈神经网络，能够直接估算数百个输入视角下所有关键的3D场景属性。该模型在相机参数估计、多视角深度估计、密集点云重建及3D点追踪等多项3D任务中均取得业界领先成果。我们采用的简洁神经网络优先方法，突破了传统视觉几何方法依赖优化与后处理才能获得精确任务特定结果的局限。相较于基于优化的方法，本方案的简洁高效特性使其特别适合实时应用场景，这正是其相较于传统优化方法的另一优势。

附录

在附录中，我们提供以下内容：

- 附录A中关键术语的正式定义。
- 全面的实施细节，包括附录B中的架构和训练超参数。
- 附录C中的补充实验与讨论
- 单视角重建的定性示例见附录D。
- 附录E中对相关文献的扩展性综述。

A. 形式定义

本节中，我们提供了进一步奠定方法部分基础的附加正式定义。

相机外参量是相对于*世界参考坐标系*定义的，我们将其视为第一台相机的坐标系。因此我们引入两个函数。第一个函数$\gamma(\mathbf{g}, \mathbf{p})=\mathbf{p}'$将$\mathbf{g}$编码的刚性变换应用于世界参考坐标系中的点$\mathbf{p}$，以获得相机参考坐标系中的对应点$\mathbf{p}'$。第二个函数$\pi(\mathbf{g}, \mathbf{p})=\mathbf{y}$进一步应用透视投影，将三维点$\mathbf{p}$映射为二维图像点$\mathbf{y}$。我们还用$\pi^{D}(\mathbf{g}, \mathbf{p})=d \in \mathrm{R}^{+}$表示从相机$\mathbf{g}$观测到的点的深度。

我们将场景建模为规则曲面集合$S_i \pitchfork \mathrm{R}^3$。由于场景会随时间变化[151]，我们将其作为第$i$个输入图像的函数。像素位置$\mathbf{y} \in I_i$处的深度定义为场景中投影到$\mathbf{y}$的任意3D点$\mathbf{p}$的最小深度，*即*，$D_i(\mathbf{y}) = \min\{\pi^{D}(\mathbf{g}_i, \mathbf{p}) : \mathbf{p} \in S_i \wedge \pi(\mathbf{g}_i, \mathbf{p})=\mathbf{y}\}$。像素位置$\mathbf{y}$处的点由$P_i(\mathbf{y})=\gamma(\mathbf{g}, \mathbf{p})$给出，其中$\mathbf{p} \in S_i$是使上述表达式最小化的3D点，*即*。$,\mathbf{p} \in S_i \wedge \pi(\mathbf{g}_i,\mathbf{p}) =\mathbf{y} \wedge \pi^{D}(\mathbf{g}_i,\mathbf{p}) =D_i(\mathbf{y})$.

B. 实施细节

架构。如主论文所述， VGGT 由24个注意力模块组成，每个模块配备一个帧级自注意力层和一个全局自注意力层。沿用DINOv2[78]中使用的ViT-L模型，每个注意力层配置为1024特征维度并采用16个头。我们使用PyTorch官方实现的注意力层*即*，*torch.nn.MultiheadAttention*，并启用闪速注意力。为稳定训练，每个注意力层还使用QKNorm[48]和LayerScale[115]，其中LayerScale初始值设为0.01。图像分词采用DINOv2[78]并添加位置嵌入。如[143]所述，将第4、11、17和23模块的分词输入DPT[87]进行上采样。

训练。为构建训练批次，我们首先随机选取训练数据集（每个数据集具有不同但近似相似的权重，如[129]所述），并从

在数据集上，我们随后随机均匀采样场景。在训练阶段，我们为每个场景选择2到24帧，同时保持每批次总帧数恒定为48帧。训练时使用各数据集对应的训练集，排除帧数少于24帧的训练序列。RGB帧、深度图和点图首先进行各向同性缩放，使较长边保持518像素。随后沿主点方向裁剪较短边至168至518像素之间，同时保持14像素块大小的整数倍。值得注意的是，我们对同一场景内每帧独立应用激进的色彩增强，以提升模型对光照变化的鲁棒性。我们按照[33，105，125]的方法构建真实轨迹：将深度图反投影为三维空间，将点位重新投影至目标帧，并保留重投影深度与目标深度图匹配的对应关系。与查询帧相似度较低的帧在批次采样时被排除。在极少数无有效对应关系的情况下，会省略跟踪损失。

C. 其他实验

IMC上的相机姿态估计我们还使用图像匹配挑战（IMC）[54]进行评估，该基准测试专注于摄影旅游数据。直到最近，该基准测试仍由经典的增量SfM方法[94]主导。

基线模型。我们评估了两种模型变体：VGGT和VGGT +BA。VGGT直接输出相机姿态估计，而VGGT +BA通过额外的束调整阶段对估计值进行优化。我们将其与经典增量SfM方法（如[66、94]）以及近期提出的深度方法进行对比。具体而言，近期提出的VGGSfM[125]首次提供了端到端训练的深度方法，在具有挑战性的摄影旅游数据集上超越了增量SfM。

除了VGGSfM，我们还对比了近期流行的DUSt3R[129]和MASt3R[62]。需要特别说明的是，DUSt3R和MASt3R在训练时使用了MegaDepth数据集的大部分场景，仅排除了0015和0022这两个场景。虽然MegaDepth数据集与IMC基准测试存在部分场景重叠（例如MegaDepth的0024场景对应大英博物馆，而大英博物馆也是IMC基准测试的场景），但两套数据集的图像并非完全相同。为确保公平比较，我们采用了与DUSt3R和MASt3R相同的训练数据划分方案。在主论文中，为保证ScanNet-1500数据集的公平性，我们从训练集中排除了对应的ScanNet场景。

结果。表10包含我们评估的结果。尽管摄影旅游数据是SfM的传统关注点

方法	测试时间优化	AUC@3°	AUC@5°	AUC@10°	运行时
COLMAP（SIFT+NN）[94]	✓	23.58	32.66	44.79	>10s
PixSfM（SIFT + NN）[66]	✓	25.54	34.80	46.73	>20s
PixSfM (LoFTR) [66]	✓	44.06	56.16	69.61	>20s
PixSfM (SP + SG) [66]	✓	45.19	57.22	70.47	>20s
DFSfM (LoFTR) [47]	✓	46.55	58.74	72.19	>10s
DUSt3R [129]	✓	13.46	21.24	35.62	~ 7s
MASt3R [62]	✓	30.25	46.79	57.42	~ 9s
VGGSfM [125]	✓	45.23	58.89	73.92	~ 6s
VGGSfMv2 [125]	✓	59.32	67.78	76.82	~ 10s
VGGT（我们的）	✗	39.23	52.74	71.26	**0.2s**
VGGT + BA（本院）	✓	**66.37**	**75.16**	**84.91**	1.8s

表10.**IMC上的相机姿态估计[54]**。我们的方法在具有挑战性的向光性数据上达到了最先进的性能，超越了VGGSfMv2[125]，后者在最新的CVPR ‘24 IMC挑战赛中相机姿态（旋转和平移）估计中排名第一。

在方法方面，本 VGGT 的前馈性能与最先进的VGGSfMv2相当，其AUC@10为71.26对比76.82，同时显著更快（每场景0.2秒对比10秒）。值得注意的是， VGGT 在所有准确率阈值下均显著优于MASt3R[62]和DUSt3R[129]，且速度更快。这是因为MASt3R和DUSt3R的前馈预测仅能处理帧对，因此需要昂贵的全局对齐步骤。此外，通过束调整， VGGT +BA进一步大幅提升，在IMC上达到最先进水平，将AUC@10从71.26提升至84.91，将AUC@3从39.23提升至66.37. 需要注意的是，我们的模型直接预测三维点，这些点可以作为BA的初始化。这消除了对三角化和BA迭代精炼的需求，如[125]所述。因此， VGGT + BA比[125]快得多。

D. 定性示例

我们在图7中进一步展示了单视角重建的定性示例。

E. 相关工作

本节将讨论其他相关研究。

视觉Transformer。Transformer架构最初是为语言处理任务提出的[6、22、120]。后来ViT将其引入计算机视觉领域[27]，并引发广泛采用。得益于其简洁性、高容量、灵活性以及捕捉长距离依赖关系的能力，视觉Transformer及其变体已成为各类计算机视觉任务架构设计的主流[4、12、83、137]。

DeiT[114]证明，通过采用强大的数据增强策略，Vision Transformers可以在ImageNet等数据集上进行有效训练。DINO[10]揭示

图7.**基于点图估计的单视图重建。**与需要将图像复制为一对的DUSt3R不同，我们的模型能够从单个输入图像预测点图。该模型对未见过的真实世界图像展现出强大的泛化能力。

视觉Transformer通过自监督方式学习特征的有趣特性。CaiT[115]引入了层缩放技术来解决训练深度视觉Transformer的挑战，有效缓解了梯度相关问题。此外，诸如QKNorm[48，150]等技术已被提出以稳定训练过程。另外，[138] 同时探讨了目标跟踪中帧级注意力模块与全局注意力模块之间的动态关系，尽管采用了交叉注意力机制。

相机姿态估计。从多视角图像中估计相机姿态是三维计算机视觉中的关键问题。过去几十年间，结构从运动（SfM）已成为主流方法[46]，无论是增量式[2，36，94，103，134]还是全局式[3，14–17，52，73，79，81，90，106]。近期，一系列方法将相机姿态估计视为回归问题[65，100，109，112，113，118，122，123，131，152，153，160]，这些方法在稀疏视角场景下展现出良好效果。AceZero[5]进一步提出通过回归三维场景坐标，而FlowMap[101]则专注于深度图作为相机预测的中间变量。相比之下，VGGSfM[125]将经典SfM流程简化为可微分框架，尤其在摄影旅游数据集上展现出卓越性能。与此同时，DUSt3R[62，129]提出了一种学习像素对齐点云的方法，通过简单对齐即可恢复相机姿态。这种范式转变引发广泛关注，因为点云作为过度参数化的表征形式，能与三维高斯点云渲染等下游应用实现无缝集成。

参考文献

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, 等. GPT-4技术报告.*arXiv预印本 arXiv:2303.08774*,2023.2

[2] 萨米尔·阿加瓦尔、古川康隆、诺亚·斯纳维利、伊恩·西蒙、布莱恩·柯莱斯、史蒂文·M·塞茨和理查德·谢利斯基。《一日建成罗马》。*ACM通讯*，54（10）:105–112,2011。2，13

[3] Mica Arie-Nachimson 、Shahar Z Kovalsky、 Ira Kemelmacher-Shlizerman 、 Amit Singer 和 Ronen Basri .基于点匹配的全局运动估计.在*2012第二届国际3D成像、建模、处理、可视化与传输会议*上，第81-88页。IEEE，2012。13

[4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, ChenSun, Mario Lučić，以及Cordelia Schmid。Vivit：一种视频视觉变换器。收录于*IEEE/ CVF 国际计算机视觉会议论文集*，第6836–6846页，2021年。12

[5] Eric Brachmann、Jamie Wynn、Shuai Chen、Tommaso Cavallari、Aron Monszpart、Daniyar Turmukhambetov 和 Victor Adrian Prisacariu。场景坐标重建：通过重定位器的增量学习进行图像集合的摆拍。发表于*ECCV*，2024。2，13

[6] 汤姆·B·布朗。语言模型是少样本学习器。*arXiv预印本 arXiv:2005.14165*，2020。12

[7] 约翰·卡本、奈拉·默里和马丁·胡门伯格。虚拟基蒂2。*arXiv预印本 arXiv:2001.10773*，2020。6

[8] 曹、贾斯汀·约翰逊、安德烈亚·韦达利和大卫·诺沃特尼。Lightplane：神经3D场的高度可扩展组件。发表于*2025年国际3D视觉会议（3DV）论文集*。9

[9] 曹晨杰与傅彦伟。通过级联捕获空间信息关键点提升基于变换器的图像匹配。发表于*IEEE/ CVF 国际计算机视觉会议（ICCV）论文集*，第12129–12139页，2023年。7

[10] Mathilde Caron、Hugo Touvron、Ishan Misra、Hervé Jé Gou、Julien Mairal、Piotr Bojanowski 和 Armand Joulin。