arXiv:2611.28793v1 [cs.GR] 21 Nov 2026

STAC++：面向流式三维重建的即插即用自适应时空感知缓存压缩

Boyi Zhu[1] Youcheng Cai[1] Dong Wang[2]1中国科学技术大学 2安徽建筑大学

摘要

流式三维重建必须在有限内存下保持几何结构，但 *STAC* 的固定分辨率空间键值对（*KV*）缓存会随着场景扩展而增长。我们提出***STAC++***，一种即插即用、免训练的框架，用于压缩和复用空间记忆，从而在无界场景中进行长期重建。自适应空间 ***KV*** 缓存（***ASKC***）将 *KV* 状态组织成八叉树，并在不同区域和层之间采用自适应分辨率。响应感知缓存压缩（***RACC***）通过测量注意力输出变化来指导 *KV* 合并，在存储节省与几何保持之间取得平衡。时序上下文细化（***TCR***）复用缓存的中间特征，仅重新计算骨干网络后缀的一小部分，从而利用后续观测细化历史预测结果。将 *STAC++*集成到四种预训练流式骨干网络中后，在长序列和大型场景上，*STAC++*可在保持具有竞争力的重建质量的同时降低空间内存消耗；在有限的计算开销下，通过细化还能进一步提升质量。代码将在论文接收后公开。

1. 引言

三维重建是计算机视觉和计算机图形学中的一项基础问题，应用范围广泛，包括机器人建图 [44], 增强现实 [28], 和新视角合成 [16, 26]。传统的基于图像的三维重建结合了运动恢复结构（SfM） [30], 用于相机估计和稀疏结构恢复，以及多视图立体（MVS） [31], 用于稠密几何重建。这些流程依赖显式的对应匹配和迭代几何优化。相比之下，近期的前馈几何模型利用预训练网络直接从图像推断场景几何，减少了对特定场景迭代优化的依赖，并提升了重建效率。

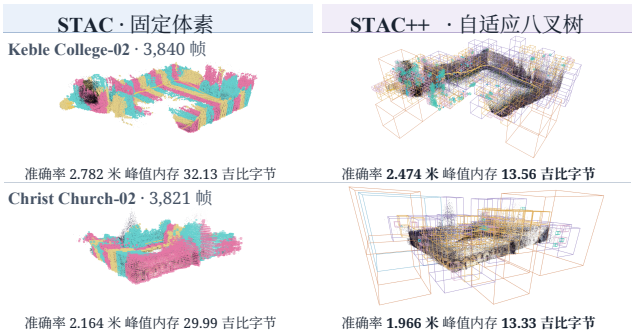

图 1. **STAC 与 STAC++ 在两个 Oxford序列上的对比。**STAC++在两个序列上均实现了更低的重建误差（准确率），且峰值 GPU内存占用更少。

前馈式重建方法大致可分为离线方法（处理固定的图像集合）和在线方法（按顺序处理输入图像）。离线方法包括 DUSt3R [42],其先预测成对点图，再进行全局对齐；VGGT [38], 则从多视图中联合预测相机和稠密几何。然而，它们的计算和内存开销限制了对长序列的扩展能力。在线方法 [5, 17, 49, 54] 则通过复用历史表示，逐步重建输入帧。StreamVGGT [54] 通过因果注意力和缓存的历史键值对（KV）状态，将VGGT适配到流式推理中，但其缓存会随着序列长度增长。InfiniteVGGT [49] 通过有界滚动缓存和多样性感知的词元剪枝控制这一增长，使长序列推理能够在固定内存预算内进行。然而，这些方法并未在KV缓存管理中显式利用三维空间信息。STAC [41]引入了空间KV缓存，将历史词元合并为体素对齐代表，在保留几何证据的同时减少空间冗余。然而，其固定分辨率网格会使占用体素的数量随场景范围增长，因此，探索无限环境的长序列流仍可能耗尽GPU内存。统一的体素粗化可以节省内存，但在复杂区域可能会丢失精细几何细节。

STAC 使用空间记忆重建输入

帧，但即使后续视角更清晰地呈现了相同表面，先前的预测仍然固定不变。这些局限促使我们提出核心问题：如何自适应地压缩并主动复用空间 *KV* 记忆，以支持无界场景的长期重建，同时在内存和计算预算有限的情况下保留几何细节并修正先前的预测？回答这一问题面临两项挑战。(1) 处理无界场景需要空间表示能够适应持续扩大的场景覆盖范围，同时控制缓存增长并保留几何细节。(2) 在有限的内存预算内运行，需要保留有价值的历史证据，并高效复用这些证据，利用后续观测修正先前的预测，而无需重新执行整个骨干网络。

为应对这些挑战，我们提出**STAC++**，这是一种即插即用、免训练的框架，可控制空间记忆的增长，并高效复用历史证据，以在无界场景中进行长期流式重建（见图1）。首先，**自适应空间 KV 缓存（ASKC）**将空间 KV 状态组织为八叉树层级结构，并针对不同场景区域和 Transformer 层采用不同分辨率。随着场景扩展，该层级结构支持将细粒度单元格合并为更粗粒度的表示。其次，**响应感知缓存压缩（RACC）**通过评估候选合并对骨干网络注意力输出（下文称为**响应**）的影响来指导键值合并。它识别出可在响应变化有限的情况下合并的空间表示，从而在节省内存与保留几何信息之间取得平衡。最后，**时序上下文细化（TCR）**会在固定延迟后重新处理历史帧，利用后续观测和累积的空间记忆细化较早的预测结果。TCR 复用缓存的中间特征，仅重新计算骨干网络的短后缀，从而在限制细化所需计算量的同时，进一步创造利用空间记忆的机会。我们将 STAC++ 实现为 STream3R 的推理时插件 [17], 、StreamVGGT [54], 、InfiniteVGGT [49],和 LingBot-Map [5], ，保留其预训练权重且无需额外训练。在这四种骨干网络上的实验表明，STAC++能够降低空间记忆消耗，同时保持有竞争力的重建质量；可扩展至长序列和大型场景，并能通过延迟后缀细化进一步提升重建质量，且计算开销有限。

我们的主要贡献总结如下：

•我们提出了 **STAC++**，这是一种**即插即用**的免训练框架，将自适应空间记忆压缩与历史证据复用相结合，可在不修改预训练主干网络权重的情况下，对无界场景进行长期流式重建。

• 我们提出**ASKC**、**RACC**和**TCR**，用于分层空间键值对缓存、响应引导合并以及历史预测的后缀细化，在保留几何信息的同时控制内存和计算开销。

2. 相关工作

2.1. 离线前馈式三维重建

DUSt3R [42]通过全局对齐预测成对点图；MASt3R [18]则加入了学习得到的匹配特征。VGGT、Fast3R 和 MUSt3R [3, 38, 48] 将预测扩展到多视图，而 FLARE [51] 则联合恢复几何、外观和相机。 π^3 [43] 通过置换等变性消除了对参考视图的依赖，VGGT-Ω [39] 则利用高效的寄存器注意力扩展几何学习。尽管如此，随着图像集合不断增大，密集的跨视图交互仍然代价高昂。

为提升可扩展性，MERG3R [8]对图像子集进行划分和对齐，而 LiteVGGT [34]则合并几何上冗余的词元，并在各层复用合并决策。VGG-T^3, 、ZipMap 和 Scal3 R [11, 14, 47] 将上下文压缩为测试时训练得到的状态，并利用快速权重或同步的自适应记忆，在图像块之间保留信息；Test3R [50] 则在测试时调整重建模型。这些方法针对的是离线可扩展性或适应，而非自适应因果 KV 存储。Octree-GS [29] 为高斯渲染提供分层空间细节；而我们的分层结构则压缩 Transformer KV 内存，以供后续流式推理使用。

2.2. 在线前馈式三维重建

Spann3R [37] 通过查询外部空间记忆来预测全局对齐的点图，CUT3R [40] 维护循环场景状态，Point3R [45] 则引入了显式的空间指针记忆。TTT3R [6] 采用置信度引导更新，而 FILT3R [15]则在不重新训练的情况下应用不确定性自适应卡尔曼式状态更新。SLA M3R [22]学习局部重建与配准；MASt3R-SLAM 和多智能体 MAG-iSt3R [12, 27] 还采用了几何优化。LASER [10] 通过在不同窗口间逐层进行深度尺度对齐，实现免训练流式处理。STream3R 和 StreamVGGT [17, 54] 采用因果 KV 缓存；LongStream [7] 增加了缓存感知训练和刷新，而 LingBot-Map [5]则将锚点、局部和轨迹上下文分离，以实现稳定的长序列重建。

不断增长的缓存促使我们进行选择性保留。在语言模型中，H_2O [53], StreamingLLM[46],和 SnapKV [19] 会保留高注意力权重令牌、注意力汇聚令牌或注意力头特定证据。对于几何，InfiniteVGGT [49]采用多样性感知剪枝；OVGGT [23]在残差引导选择期间保护锚点；RegVGGT [24]则调节输入令牌。GHOST [4] 执行几何引导驱逐，而

RetrieveVGGT [55]通过查询—键相似度和位姿检索上下文。选择可减少缓存增长，但可能丢弃有用的几何证据；空间聚合则提供了一种互补方式，用于保留来自重复观测的信息。Slot3R [52] 通过为每个空间地址保留多个状态，避免过早融合互补视图；SURE-Map [21] 则利用跨视图不确定性进行局部校正，并通过周期性的关键帧窗口推理确保几何一致性。

与我们的工作最为相关的是，STAC [41] 在固定分辨率的体素网格上结合了时间令牌保留与空间KV融合，使得无需重新训练骨干网络即可复用历史信息。我们在此设计基础上引入自适应空间分辨率和响应引导的KV合并，以控制内存增长；同时采用选择性后缀细化，利用保留的证据改进较早的预测结果。

3. 预备知识

3.1. 因果注意力

因果注意力通过限制每一帧只能利用当前帧及前一帧的信息来实现流式重建，STream3R [17] 和 StreamVGGT [54]均采用了这一方法。设 F_t 表示帧 t 的编码特征， G_t 表示其几何预测结果。对于由T帧组成的序列，该过程可概括为

$$\{G_t\}_{t=1}^{T} = \mathrm{Decoder}\big(\mathrm{TemporalSelfAttn}\big(\{F_t\}_{t=1}^{T}\big)\big)\,, \tag{1}$$

其中，TemporalSelfAttn 应用因果掩码，使帧 t 仅关注第 1、.....、 t帧；解码器则将经过注意力处理的特征映射为逐帧几何预测结果。在流式推理过程中，历史键和值会被缓存并重复利用，而无需重新处理前面的帧。然而，保留完整的KV历史会使缓存存储量随序列长度线性增长。

3.2. 时空KV缓存

STAC [41] 维护互补的时序和空间KV缓存。时序缓存 $\mathcal{M}_t^{\mathrm{temp}}$ 保留首帧参考、近期帧和锚点。空间缓存则根据预测的三维位置，将因时序淘汰的图像块词元组织成边长为 δ 的体素。从该缓存中检索可提供空间上下文 $\mathcal{M}_t^{\mathrm{spat}}$ 。

空间缓存。 每个体素 u 最多存储 P 个键值对代表项 $\mathcal{G}_u$ 和 B 个缓冲令牌 $\mathcal{E}_u$；达到容量上限时，通过聚合整合词元。代表项保留融合权重 W_i、重要性分数 S_i和所表示词元的数量 C_i，而缓冲令牌则保留重要性分数，仅通过聚合发挥作用。在帧 t，检索会在包含前一帧预测三维点的体素周围，沿每个轴的正反两个方向各延伸两个体素步长（2δ）的邻域内，搜索占用体素$\mathcal{N}_t$ 。空间上下文通过为每个注意力头选取融合权重最大的至多 R 个代表项来构建：

$$\mathcal{M}_t^{\mathrm{spat}} = \mathrm{Top}_R^{W}\left(\bigcup_{u\in\mathcal{N}_t}\mathcal{G}_u\right). \tag{2}$$

检索到的空间状态、时序缓存和当前帧KV $\mathcal{M}_t^{\mathrm{cur}}$ $\mathcal{C}_t = \mathcal{M}_t^{\mathrm{temp}} \cup$ 共同构成注意力上下文。$\mathcal{M}_t^{\mathrm{spat}} \cup \mathcal{M}_t^{\mathrm{cur}}$ 这项工作聚焦于空间缓存。

4. 方法

我们提出了**STAC++**，一个**即插即用**的免训练框架，用于控制空间KV记忆增长，并在流式三维重建中复用历史证据。STAC++由三个模块组成（图2）。**自适应空间KV缓存（ASKC）**以多种空间分辨率组织历史状态。**响应感知缓存压缩（RACC）**通过衡量各状态对注意力输出的影响，确定哪些状态可以合并。**时序上下文优化（TCR）**会在延迟一段时间后重新处理历史帧，复用缓存的中间特征，并重新计算较短的骨干网络后缀，以纳入后续观测。所有三个模块均在推理期间运行，并保留预训练主干网络权重。

4.1. 自适应空间 KV 缓存（ASKC）

在无界场景中，固定分辨率的空间缓存会随着场景覆盖范围的扩大而增长，并在相邻体素间保留冗余信息。统一降低分辨率虽能节省存储，但可能会丢弃局部差异明显的几何证据。相比之下，ASKC 将空间 KV 状态组织在不断扩展的八叉树中，在各层独立地选择性合并相邻的细粒度单元，使其成为较粗的父节点，同时在仍需详细证据的区域保留更精细的表示。

逐层建模。对于具有 L 全局注意力层的骨干网络，第 ℓ 层在时间 $\mathcal{T}_{(\ell)}$ t 维护一棵八叉树 t，其最细单元的边长为 δ （第 3.2 节）。各层共享相同的细分规则，但独立进行合并，因此同一场景区域在不同层可以采用不同的空间分辨率。层内所有 H 注意力头共享单元格，但分别存储 KV 状态。我们将该层的词元输入记为 $X_t^{(\ell)}$ 。时间维度上被淘汰的图像块词元会根据其预测的三维位置进入相应单元格，包括先前合并形成的父节点。对于落在边界外的点，我们反复将共享根节点的边长加倍，直至该点被包含在内；每次扩展时，旧根节点都会保留为子节点。

单元内存储。 每个层 ℓ 和注意力头 h中的单元格 u 最多存储 P 个代表项$\mathcal{G}_{u,\ \ell,\ h}$ 以及 B 个缓冲令牌$\mathcal{E}_{u,\ \ell,\ h}$。遵循 STAC 的在线聚合方法 [41],，我们省略单元格、层和注意力头的索引，并将每个代表项写作$m_i = (k_i,\ v_i)$，其中键向量和值向量分别为 k_i 和 v_i。当新代表项需要一个槽位，但所有 P个槽位都已占满时，

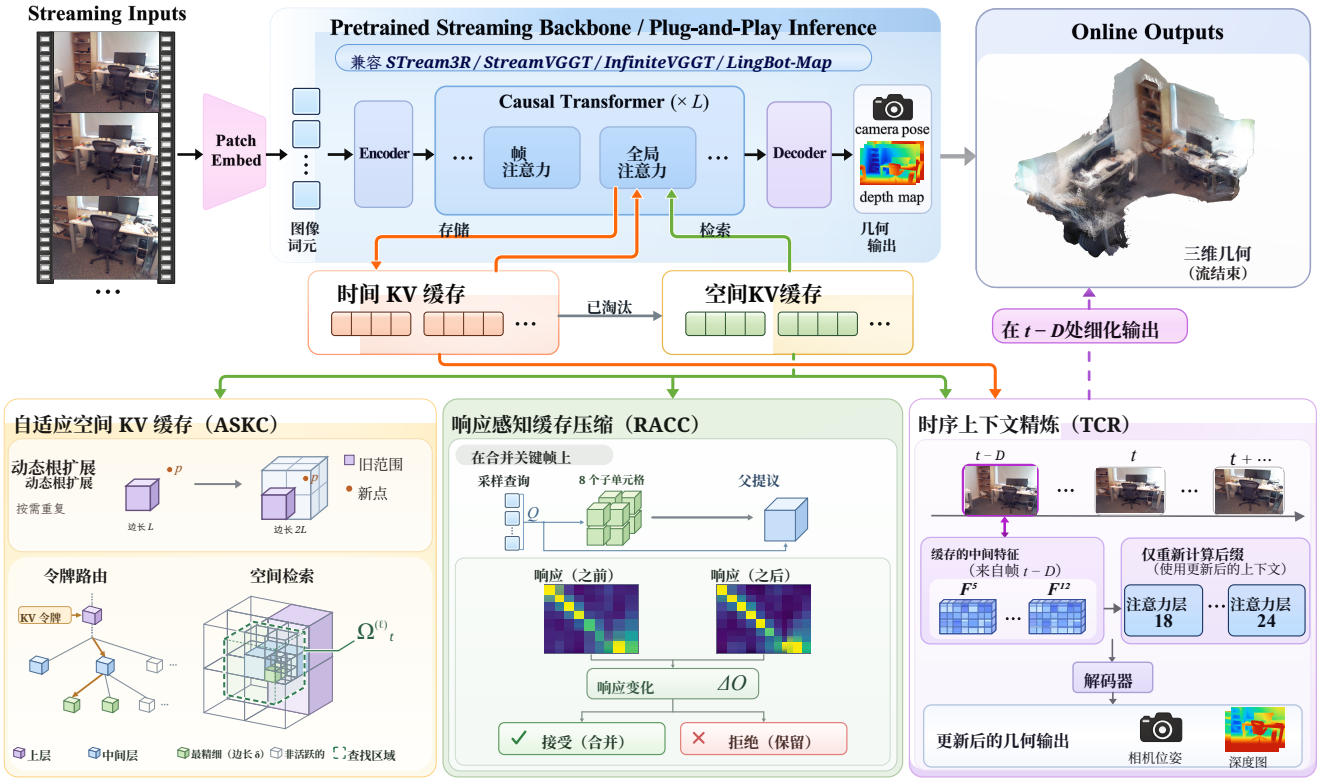

图 2. **STAC 概览++**。一种用于流式三维重建的即插即用、免训练框架。ASKC 在自适应八叉树中组织空间 KV 状态；RACC 利用注意力输出变化指导缓存压缩。TCR 复用缓存特征，并重新计算一段较短的骨干网络后缀，以结合后续观测细化历史预测。

我们将 i 与键余弦相似度最高的 W_i 剩余代表项 j 进行融合：

$$m_j \leftarrow \frac{W_j m_j + \omega_{ij} m_i}{W_j + \omega_{ij}}, \tag{3}$$

其中，$\omega_{ij} = W_i \exp(\cos(k_i, k_j) - 1)$。我们更新 $W_j \leftarrow W_j + \omega_{ij}$、$C_j \leftarrow C_j + C_i$ 和 $S_j \leftarrow S_j + S_i$。与某个代表足够相似的输入词元使用相同的更新方式，其中 $\omega_{ij} = \exp(\cos(k_i, k_j))$ 和 $C_i = 1$。足够不相似且通过重要性阈值的词元会进入缓冲区。缓冲区满时，取其中得分最高的词元作为种子，并使用指数余弦相似度权重将所有条目聚合为一个新的代表。

空间检索。令 Ω_t 为前一帧预测的三维点的查找邻域，并将其扩展 2δ（第3.2节），且 $\mathcal{N}_t^{(\ell)}$ 包括所有范围与 Ω_t 相交的活动单元格，包括合并后的父节点。对于注意力头 h，在相同的每个注意力头预算 R 下，将 $\mathcal{N}_t$ 和 $\mathcal{G}_u$ 分别替换为 $\mathcal{N}_t^{(\ell)}$ 和 $\mathcal{G}_{u,\ell,h}$，则由式 2 可得 $\mathcal{M}_{t\ell h}^{\text{spat}}$、。

层次化粗化。对于每一层，如果某个非活动父节点有 1 到 8 个活动的直接子节点，且其孙节点或更深层后代中没有键值对，则该节点在结构上符合合并条件。将这些子节点合并到父节点中属于父提议，需要通过 RACC 接受（第 4.2 节）。

4.2. 响应感知缓存压缩（RACC）

合并空间相邻状态可以节省内存，但可能改变骨干网络可利用的几何证据。一种直接的方法是显式计算候选合并 p 前后的注意力响应，并取其差值：

$$\Delta o_{p,\ell,h} = \text{Att}(\widehat{q}_{\ell,h}, \mathcal{C}_{p,\ell,h}) - \text{Att}(\widehat{q}_{\ell,h}, \mathcal{C}_{\ell,h}), \tag{4}$$

其中，Att(·) 表示注意力，$\widehat{q}_{\ell,h}$ 表示共享查询，$\mathcal{C}_{\ell,h}$ 和 $\mathcal{C}_{p,\ell,h}$ 分别表示合并前后的上下文。相应的响应分别记为 $o_{\ell,h}$ 和 $o_{p,\ell,h}$。然而，对每个候选合并直接评估这种变化，都需要在完整上下文上重复进行查询–键评分、softmax 归一化和值聚合，因此计算成本高昂。因此，我们对少量查询进行采样，并增量计算响应变化：复用参考响应和归一化因子，同时仅更新每次合并所影响的 KV 条目的贡献。RACC 在响应保真度与存储节省之间进行权衡。

合并调度。对于间隔为 M 的合并关键帧

帧，我们保留 Ω_{t-M} 以及输入 $X_{t-M}^{(\ell)}$ 的完整内容，并将其提供给每个全局注意力层。关键帧 t 完成推理后，我们从这些输入的所有特殊标记位置以及均匀的 4×4 图块网格中采样查询，以评估 Ω_{t-M} 中符合条件的父节点。这种延迟使中间帧能够在进行粗化之前使用更精细的空间表示。待处理的关键帧会在序列末尾使用可用的上下文。

父节点提议构造。在每一层 ℓ，考虑一个符合条件的父节点 U，其位于 Ω_{t-M}，并具有子节点位置 u_i（$i = 1, \ldots, 8$；空分支对应空集合），省略下文中的$t - M$。注意力头 h 从 $\bigcup_{i=1}^{8} \mathcal{E}_{u_i,\ell,h}$ 中保留重要性最高的至多 B 个缓冲条目，位于 $\mathcal{E}_{U,\ell,h}$中。若 $\sum_{i=1}^{8} |\mathcal{G}_{u_i,\ell,h}| \le P$，子节点代表构成 $\mathcal{G}_{U,\ell,h}$ 且保持不变；否则，我们使用公式 3，按照第 4.1 节中的代表项重新合并规则反复合并，直至剩下 P。这些配对 $p_{U,\ell,h} = (\mathcal{G}_{U,\ell,h}, \mathcal{E}_{U,\ell,h})$ 构成提议 p，从而节省

$$\Delta N_{p,\ell,h} = \sum_{i=1}^{8} \left(|\mathcal{G}_{u_i,\ell,h}| + |\mathcal{E}_{u_i,\ell,h}|\right) - |\mathcal{G}_{U,\ell,h}| - |\mathcal{E}_{U,\ell,h}|. \tag{5}$$

对注意力头求和得到 $\Delta N_{p,\ell} = \sum_{h=1}^{H} \Delta N_{p,\ell,h}$。

响应变化评估。对于采样得到的 $\widehat{q}_{\ell,h}$，计算 $o_{\ell,h} = \mathrm{Att}(\widehat{q}_{\ell,h}, \mathcal{C}_{\ell,h})$，遍历 $\mathcal{C}_{\ell,h} = \mathcal{M}_{\ell}^{\mathrm{temp}}{}_{,h} \cup \mathcal{M}_{\ell}^{\mathrm{spat}}{}_{,h} \cup \mathcal{M}_{\ell}^{\mathrm{cur}}{}_{,h}$（省略时间）。我们保留 $\Lambda_{\ell,h} = \log \sum_{i \in \mathcal{C}_{\ell,h}} \exp(z_i)$，其中 $z_i = \widehat{q}_{\ell}^{\top}{}_{,h} k_i / \sqrt{d} + \log C_i$，并省略词元上的层/注意力头索引；$d$ 是键的维度，$C_i = 1$ 用于未合并词元。提议 p 将检索到的子节点代表 $\mathcal{R}_p$ 替换为不同的父节点代表 $\mathcal{A}_p \subseteq \mathcal{G}_{U,\ell,h}$，这些代表包含其贡献，从而得到 $\mathcal{C}_{p,\ell,h} = (\mathcal{C}_{\ell,h} \setminus \mathcal{R}_p) \cup \mathcal{A}_p$。其余上下文和检索排序保持不变。对于 $i \in \mathcal{R}_p \cup \mathcal{A}_p$，使用参考归一化因子设置$\alpha_i = \exp(z_i - \Lambda_{\ell,h})$。公式 4 变为

$$\Delta o_{p,\ell,h} = \frac{\sum_{i \in \mathcal{A}_p} \alpha_i (v_i - o_{\ell,h}) - \sum_{i \in \mathcal{R}_p} \alpha_i (v_i - o_{\ell,h})}{1 + \sum_{i \in \mathcal{A}_p} \alpha_i - \sum_{i \in \mathcal{R}_p} \alpha_i}. \tag{6}$$

对于固定查询和指定替换 [9, 25]，公式6 是精确的；只有 $\mathcal{R}_p$ 和 $\mathcal{A}_p$ 需要新的分数和值累积。未变化的贡献通过 $o_{\ell,h}$ 和 $\Lambda_{\ell,h}$复用。对采样查询堆叠参考响应及其变化，可得$O_{\ell,h}$ 和$\Delta O_{p,\ell,h}$。

合并接受。 我们对各注意力头的归一化响应变化取平均：

$$E_{p,\ell} = \frac{1}{H} \sum_{h=1}^{H} \frac{\|\Delta O_{p,\ell,h}\|_F}{\max(\|O_{\ell,h}\|_F, \epsilon_m)}, \tag{7}$$

其中 $\epsilon_m = 10^{-8}$ 表示稳定性。当 p时，我们接受

$$E_{p,\ell} \le \tau_m + \lambda \frac{\Delta N_{p,\ell}}{H(P+B)}. \tag{8}$$

这里，τ_m 设定基准容差，λ 则为存储节省赋予权重。每一层都会独立作出决策，并在所有注意力头之间共享；接受后，子存储单元将由其父单元替代。

4.3. 时序上下文精炼（TCR）

在因果流式推理过程中，每一帧仅使用其到达时可用的观测进行重建。后续帧可能为先前观测到的区域提供互补视角和额外几何证据，从而丰富空间缓存。然而，这种更新后的内存虽能让后续推理受益，却不会自动修正较早的预测；这些预测仍受限于其原有且更有限的上下文。最直接的方法是利用更新后的上下文，对历史帧重新运行整个骨干网络。然而，这会重复大量特征计算。因此，我们引入 TCR，在固定延迟后利用更新后的时序和空间上下文细化历史预测。TCR 在初始推理期间保留中间层特征，并仅重新计算较短的骨干网络后缀，使较早的帧能够利用后续观测，同时限制额外计算量。

细化调度。 在初始推理期间，我们保留每帧第 5 层和第 12 层的解码器特征、后缀重计算所需的中间输入，以及该帧的空间查找区域。经过 D 帧的延迟后，我们使用更新后的上下文重新处理一次历史帧。细化延迟独立于合并间隔，合并关键帧与 RACC 共享其保留的输入。

细化过程。 在时间 t，我们从保留的中间特征处继续计算，以细化帧 $t - D$。在每个重新计算的注意力层中，来自该重新访问帧的查询会关注当前时序缓存、在其保存的查找区域内检索到的空间状态，以及根据同一帧在该层的特征重新计算得到的键和值。令X 表示保留的中间输入，$\mathcal{C}$ 表示更新后的上下文。重新计算第 18–24 层可得

$$\left(\widetilde{F}^{(18)}, \widetilde{F}^{(24)}\right) = \mathrm{Suffix}_{18:24}(X, \mathcal{C}), \tag{9}$$

其中 $\mathrm{Suffix}_{18:24}$ 表示使用更新后的上下文 $\mathcal{C}$，基于保留的输入 X 重新计算骨干网络第 18–24 层，并返回第 18 层和第 24 层的特征。Here$F^{(\ell)}$ 表示第 ℓ层的特征，波浪号表示细化，帧索引省略。解码器将第 5 层和第 12 层的缓存特征与第 18 层和第 24 层的更新特征结合：

$$\widetilde{G} = \mathrm{Decoder}\left(F^{(5)}, F^{(12)}, \widetilde{F}^{(18)}, \widetilde{F}^{(24)}\right), \tag{10}$$

表1. 在 7-Scenes、Neural RGB-D 和 ETH3D 上进行的三维重建。在四种在线骨干网络上，STAC++ 比 STAC 使用更少的内存，同时保持了有竞争力的重建质量。内存表示以吉比字节为单位的 GPU 平均峰值使用量。

方法	Type	7-Scenes 准确率 ↓	7-Scenes 完整度 ↓	7-Scenes F1 ↑	Neural RGB-D 准确率 ↓	Neural RGB-D 完整度 ↓	Neural RGB-D F1 ↑	ETH3D 准确率 ↓	ETH3D 完整度 ↓	ETH3D F1 ↑	内存 ↓ (吉比字节)
ViPE [13]	优化	0.061	0.032	85.56	0.183	0.105	61.05	2.514	3.493	12.47	11.38
MegaSaM [20]	优化	0.027	0.026	94.25	0.047	0.018	93.81	1.910	1.981	19.84	10.70
VGGT [38]	离线	0.022	0.022	96.46	0.017	0.012	99.04	0.199	0.128	86.91	16.88
VGG-T^3 [11]	离线	0.046	0.029	92.06	0.059	0.025	86.15	0.441	0.292	70.47	14.99
CUT3R [40]	在线	0.093	0.038	81.09	0.210	0.095	62.44	0.933	0.730	58.57	10.34
TTT3R [6]	在线	0.035	0.026	92.99	0.101	0.031	83.84	0.607	0.597	63.20	6.02
ZipMap [14]	在线	0.114	0.054	76.14	0.368	0.271	47.50	0.807	0.570	60.83	17.15
STream3R [17][a]	在线	**0.043**	0.021	**95.52**	0.033	0.010	97.30	0.553	0.288	70.33	27.18
STream3R-W	在线	0.082	0.025	89.63	0.085	0.017	88.86	0.556	0.293	70.24	11.67
STream3R-STAC	在线	0.048	**0.020**	95.38	0.066	**0.013**	**94.56**	**0.492**	0.242	**70.59**	14.28
STream3R-STAC++	在线	0.047	**0.020**	95.32	**0.063**	0.014	94.14	0.543	**0.234**	69.74	**10.84**
StreamVGGT [54][b]	在线	**0.046**	**0.025**	**90.74**	0.132	0.064	70.44	0.674	0.504	56.96	23.41
StreamVGGT-W	在线	0.071	0.031	85.59	0.165	0.075	62.85	0.676	0.503	56.87	11.41
StreamVGGT-STAC	在线	0.056	0.026	89.21	0.141	0.056	68.12	0.604	0.471	**59.59**	15.17
StreamVGGT-STAC++	在线	0.053	0.027	89.76	**0.126**	**0.046**	**73.16**	**0.596**	**0.458**	56.64	**11.11**
InfiniteVGGT [49]	在线	**0.048**	**0.025**	**90.27**	0.126	0.059	69.40	0.637	0.474	57.14	13.57
InfiniteVGGT-STAC	在线	0.051	0.027	89.88	0.129	0.060	69.42	0.627	0.463	**58.23**	15.43
InfiniteVGGT-STAC++	在线	0.049	0.028	90.09	**0.121**	**0.050**	**72.18**	**0.610**	**0.442**	57.35	**12.75**
LingBot-Map [5]	在线	0.029	**0.020**	94.96	0.057	0.023	89.08	0.487	0.514	69.97	**13.35**
LingBot-Map-STAC	在线	**0.028**	0.021	**95.13**	0.055	**0.021**	90.93	0.487	0.485	69.53	17.59
LingBot-Map-STAC++	在线	**0.028**	0.021	95.08	**0.054**	**0.021**	**91.07**	**0.480**	**0.479**	**70.68**	15.12

a STream3R 在 9 个 Neural RGB-D 序列中的 4 个上发生内存溢出；
b StreamVGGT 在 9 个 Neural RGB-D 序列中的 1 个上发生内存溢出

~其中 G 表示由原生的稠密注意力头和相机注意力头生成的细化几何预测。

5. 实验

5.1. 实现细节

STAC++ 是一个即插即用、免训练的框架。我们将其集成到四种在线重建骨干网络中：STream3R [17]，、StreamVGGT [54]，、InfiniteVGGT [49],和 LingBot-Map [5]，，并冻结其预训练权重。作为对比，我们还将 STAC 的空间缓存 [41] 扩展到 InfiniteVGGT 和 LingBot-Map，同时保留它们原生的时间记忆策略。InfiniteVGGT 按令牌预算而非固定时间窗口进行驱逐；在预测出被驱逐补丁的三维位置后，我们会将其键值对存档到空间内存中。

数据集与指标。 我们在 7-Scenes [33], Neural RGB-D [1], 和 ETH3D [32],上评估三维重建，并在 Long3D [49]和牛津尖塔 [36]（不包括 christ- church-05）上评估大型场景和长序列流的重建。对于相机位姿估计，我们在 ETH3D、Sintel [2],和 TUM RGB-D [35]上进行评估。我们报告准确率（Acc）和完整度（Comp），分别表示预测结果到参考结果以及参考结果到预测结果的最近邻距离均值（m）。F1值（%）在 7-Scenes 上采用 0.10m 阈值在 Neural RGB-D 上采用 0.10m 阈值，在 ETH3D、Long3D 和牛津尖塔上采用 0.25m 阈值。对于相机，我们报告经过 Sim(3) 对齐后的绝对轨迹误差（ATE，m）和角度 AUC（%），在 3° 和 30°处使用每个有效帧对中较大的相对旋转误差或平移方向误差。

基线。 我们将 STAC++集成到 STream3R [17],、StreamVGGT [54], 、InfiniteVGGT [49], 和 LingBot-Map [5] 中。对于每种骨干网络，我们将 STAC++与原生推理进行比较，并另外纳入窗口化 STream3R 和 StreamVGGT 基线。

实现细节。 按照 STAC [41], ，每个单元格在每个注意力头中最多存储 $P = 4$ 个代表项和 $B = 8$ 个缓冲令牌。我们使用的最细单元格大小为 $\delta = 0.05$，合并参数为 $\tau_m = \lambda = 5 \times 10^{-4}$，后缀深度为$k = 8$，默认细化延迟和合并间隔为$D = M = 8$ 帧（大型场景中为 $D = M = 16$）。STream3R-W 和 StreamVGGT-W 分别使用 36 帧和 48 帧的窗口，而它们的 STAC 和 STAC++变体使用 8 帧窗口。InfiniteVGGT 原生的 1,200,000 令牌预算，在 STAC 和 STAC++中降至 650,000。所有 LingBot-Map 变体均使用包含 16 个关键帧的时间窗口，并保留 8 个尺度锚点。这样，STAC 和 STAC++在每种骨干网络中共享相同的时序记忆设置。所有实验均在一台英伟达

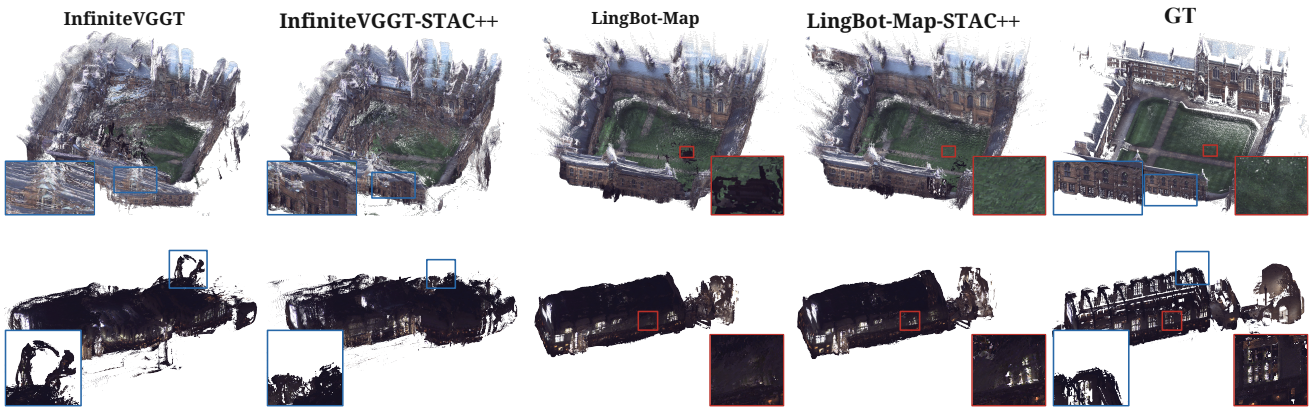

图 3. 大场景重建的定性比较。蓝色插图比较 InfiniteVGGT 与其 STAC++ 变体，红色插图比较 LingBot-Map 与其 STAC++ 变体。最右侧一列显示激光扫描真实值。

表 2. Long3D 和牛津尖塔上的定量结果。STAC++ 使 InfiniteVGGT 和 LingBot-Map 完成了所有评估序列，而它们的 STAC 变体则因内存不足而停止运行。内存以吉比字节为单位，推理时间以毫秒/帧为单位。

方法	准确率 ↓	完整度 ↓	F1 ↑	内存 ↓	推理 ↓
Long3D					
STream3R [17]	*OOM*	*OOM*	内存溢出	内存溢出	*OOM*
STream3R-W	**2.815**	**0.768**	**26.06**	13.32	224.26
STream3R-STAC	3.210	0.917	22.57	15.69	116.84
STream3R-STAC++	3.192	0.858	22.63	**10.75**	**190.52**
StreamVGGT [54]	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
StreamVGGT-W	2.717	**0.648**	23.76	13.27	285.96
StreamVGGT-STAC[d]	2.691	0.533	25.82	29.45	118.49
StreamVGGT-STAC++	**2.713**	1.003	**24.30**	**11.39**	**205.04**
InfiniteVGGT [49]	**3.136**	**0.767**	**22.08**	15.93	**419.29**
InfiniteVGGT-STAC[e]	3.276	1.106	23.87	21.19	386.82
InfiniteVGGT-STAC++	3.242	1.559	20.67	**15.72**	472.00
LingBot-Map [5]	**2.007**	0.968	**29.89**	**11.92**	252.26
LingBot-Map-STAC	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
LingBot-Map-STAC++	2.354	**0.948**	29.39	16.48	**239.67**
牛津尖塔					
InfiniteVGGT [49]	18.436	5.907	2.00	**15.54**	**464.58**
InfiniteVGGT-STAC	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
InfiniteVGGT-STAC++	**11.594**	**1.686**	**4.96**	17.40	628.46
LingBot-Map [5]	2.951	1.730	**19.59**	**15.43**	**154.54**
LingBot-Map-STAC[c]	3.489	1.665	14.27	35.47	257.88
LingBot-Map-STAC++	**2.783**	**1.466**	19.28	15.65	204.08

c LingBot-Map-STAC 在 3/9 个牛津尖塔场景上发生内存溢出
d StreamVGGT-STAC 在 1/5 个 Long3D 场景上发生内存溢出
e InfiniteVGGT-STAC 在 1/5 个 Long3D 场景上发生内存溢出

配备 48 吉比字节内存的 GeForce RTX 4090。

5.2. 三维重建。

表1 比较了在 7-Scenes、Neural RGB-D 和 ETH3D 上的重建效果。相较于窗口化 STream3R 和 StreamVGGT，以及原生的 InfiniteVGGT 和 LingBot-Map，STAC++在 12 种设置中的 9 种设置下提升了 F1 值。四种骨干网络在 Neural RGB-D 上均有提升，增幅为 1.99–10.31 个百分点。在与 STAC 相同的时间窗口或令牌预算下，STAC++ 使各骨干网络的平均峰值 GPU 内存降低了 14.0%–26.8%。这表明，自适应压缩可以减少空间冗余，同时保留对重建有用的证据。将 STAC++集成到 LingBot-Map 后，其在这些数据集上的准确率、完整度和 F1 值均优于 CUT3R [40], TTT3R [6], 和 ZipMap [14]。

对于大型场景和长序列流，表2 报告了 Long3D [49] 和牛津尖塔 [36]上的结果。VGGT [38],VGG-T^3 [11], CUT3R [40], 和 ZipMap [14] 在这些大型场景基准测试中会发生内存溢出（OOM）故障。对于 InfiniteVGGT 和 LingBot-Map，STAC++能够完成所有评估序列，而 STAC 变体则在多个序列中发生内存溢出故障。这一差异凸显了随着轨迹延长、场景覆盖范围扩大，控制空间内存增长的重要性。STAC++ 还提升了牛津尖塔上的重建质量，降低了 InfiniteVGGT 和 LingBot-Map 的准确率和完整度。从推理时间来看，STAC++ 在 Long3D 上比窗口化 STream3R 和 StreamVGGT 基线更快，而相较于原生 InfiniteVGGT 和 LingBot-Map，其运行时间因数据集而异。这些结果表明，自适应空间缓存能够在保留几何证据的同时，支持更可靠的长序列流推理；在这些基准测试评估的不同场景以及不同预训练骨干网络中，质量与内存之间的权衡各不相同。

如图 3所示，STAC++ 能够生成高质量的

表 3. Sintel、TUM RGB-D 和 ETH3D 上的相机位姿估计。在四种在线骨干网络上，STAC++ 通常能降低 ATE，同时比 STAC 使用更少内存。我们报告 ATE（米）、角度 AUC（%）和平均峰值 GPU 内存（吉比字节）。

方法	Type	Sintel			TUM			ETH3D			内存 ↓
		ATE ↓	AUC@3↑	AUC@30↑	ATE ↓	AUC@3↑	AUC@30↑	ATE ↓	AUC@3↑	AUC@30↑	（吉比字节）
STream3R [17]	在线	0.214	**7.65**	**53.02**	**0.025**	**7.10**	**48.62**	1.906	14.26	62.45	13.49
STream3R-W	在线	0.214	7.41	52.76	0.032	3.93	41.80	1.855	14.08	61.80	10.61
STream3R-STAC	在线	0.204	6.12	53.01	0.032	6.06	44.70	**1.433**	16.78	**67.25**	12.33
STream3R-STAC++	在线	**0.201**	6.27	51.75	0.031	4.78	44.89	1.551	**17.00**	66.80	**9.07**
StreamVGGT [54]	在线	0.233	6.00	47.94	0.026	8.12	53.48	1.898	13.59	58.83	11.83
StreamVGGT-W	在线	0.233	6.00	47.94	0.029	7.76	50.71	1.900	13.51	58.58	10.35
StreamVGGT-STAC	在线	0.226	**10.03**	**52.63**	0.026	**12.32**	**58.44**	1.649	**16.46**	62.25	13.22
StreamVGGT-STAC++	在线	**0.214**	8.26	49.42	**0.025**	10.38	56.11	**1.607**	15.85	**64.20**	**9.65**
InfiniteVGGT [49]	在线	0.236	6.36	47.21	0.027	**8.25**	52.60	1.742	12.87	59.63	**12.15**
InfiniteVGGT-STAC	在线	0.238	6.33	47.25	0.027	8.13	**52.74**	1.755	12.61	59.67	16.02
InfiniteVGGT-STAC++	在线	**0.229**	**7.40**	**48.00**	**0.026**	7.66	52.61	**1.724**	**13.70**	**61.79**	12.63
LingBot-Map [5]	在线	0.126	3.56	50.44	0.021	11.72	59.20	0.712	35.99	75.49	**14.26**
LingBot-Map-STAC	在线	0.130	3.47	**50.97**	0.019	13.43	62.39	0.685	38.08	**77.40**	18.66
LingBot-Map-STAC++	在线	**0.121**	**3.70**	49.68	**0.018**	**14.41**	**63.55**	**0.623**	**38.87**	77.38	14.56

表 4. 在牛津尖塔数据集上对 LingBot-Map-STAC++ 的 ASKC、RACC 和 TCR 进行消融实验。内存以吉比字节为单位，推理时间以毫秒/帧为单位。

配置	准确率 ↓	F1 ↑	内存 ↓	推理 ↓
无 ASKC（STAC）[c]	3.4888	14.27	35.47	257.88
无 RACC	2.9131	18.62	**14.76**	172.13
随机合并	2.8929	19.22	14.87	190.95
不含 TCR	2.7927	19.22	15.16	**163.54**
TCR（2 层）	2.7995	**19.52**	15.69	204.31
TCR（14 层）	2.8118	18.99	15.44	216.01
TCR（21 层）	2.8031	19.06	15.44	239.34
TCR（24 层）	2.8010	18.94	15.38	239.96
完整（8 层）	**2.7828**	19.28	15.65	204.08

借助 InfiniteVGGT 和 LingBot-Map 进行大场景重建。结果保留了连贯几何结构和局部结构细节，表明紧凑空间记忆能够维持视觉重建质量。

5.3. 相机位姿估计。

遵循以往工作 [5, 41], , 我们在 Sintel [2], 、TUM RGB-D [35], 和 ETH3D [32]上评估相机位姿估计性能。如表3所示，STAC++ 在全部 12 项比较中均降低了 ATE，相应的原生基线采用了 STream3R 和 StreamVGGT 的窗口化变体。与相同时间窗口或令牌预算下的 STAC 相比，它在 12 种设置中的 11 种设置下进一步降低了 ATE，并使不同骨干网络的平均峰值 GPU 内存减少了 21.2%–27.0%。这表明，更紧凑的空间缓存能够保留对相机定位有用的证据，同时提升内存效率。不同数据集和骨干网络上的角度 AUC 增益有所不同，其中两项指标在 TUM RGB-D 上的 LingBot-Map 中均有所提升。

5.4. 消融研究

我们在牛津尖塔数据集上开展消融研究，以评估 ASKC、RACC 和 TCR。ASKC 和 RACC 的消融实验考察空间缓存压缩，TCR 的消融实验考察时序上下文细化。

ASKC 的作用。 w/o ASKC 变体对应于 STAC，它仅在固定分辨率的空间单元格内聚合词元（表4）。与这一基线相比，分层合并显著减少了内存占用。

RACC 的作用。 w/o RACC 变体接受所有符合条件的合并，而随机合并采用响应引导策略的逐序列接受率，结果取三个随机种子的平均值。在内存占用相近的情况下，响应引导合并的准确率和 F1值均优于随机合并和 w/o RACC（表 4）。

TCR 的作用。 表 4显示，在 TCR 变体中，细化八个层可取得最高准确率。较浅的细化几乎没有益处，而更深的后缀会增加计算量，却无法提升质量，这支持将 $k = 8$ 作为默认设置。

6. 结论

我们提出了 STAC++，这是一个面向无界场景中长期流式三维重建的即插即用、免训练框架。自适应空间 KV 缓存（ASKC）利用八叉树组织空间 KV 状态，并在不同区域和 Transformer 层之间采用自适应分辨率。响应感知缓存压缩（RACC）通过注意力输出变化引导键值合并，在节省内存与保持几何结构之间实现平衡。时序上下文细化（TCR）复用缓存的中间特征，并重新计算较短的骨干网络后缀，以利用后续观测细化历史预测

结果。对四种预训练骨干网络的实验表明，该方法能够降低内存消耗，同时保持具有竞争力的重建质量和相机位姿估计质量，并提升对长序列和大型场景的可扩展性。延迟细化还能在有限计算开销下进一步提高重建质量。

大语言模型使用声明

OpenAI Codex 协助整理手稿、润色语言、检查描述与实现是否对应，以及准备 LaTeX 源文件。作者负责验证方法、实验、引文及最终论述。

参考文献

[1] Dejan Azinovi´c, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner 和 Justus Thies。神经 RGB-D 表面重建。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 6290–6301 页，2022年。6[2] Daniel J. Butler、Jonas Wulff、Garrett B. Stanley 和 Michael J. Black。用于光流评估的自然风格开源影片。载于《欧洲计算机视觉会议论文集》，第 611–625 页，2012。6, 8[3] Yohann Cabon、Lucas Stoffl、Leonid Antsfeld、Gabriela Csurka、Boris Chidlovskii、Jerome Revaud 和 Vincent Leroy。MUSt3R：用于立体三维重建的多视图网络。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 1050–1060 页，2025年。2[4] Leyang Chen、Junyi Wu、Zhiteng Li 和 Yulun Zhang。GHOST：面向高效三维重建的几何分层在线流式令牌淘汰。载于《神经信息处理系统进展》，2026年。即将发表。2[5] Lin-Zhuo Chen、Jian Gao、Shangzhan Zhang、Yihang Chen、Nan Xue、Jianyuan Wang、Christian Rupprecht、Xun Cao、Xing Zhu、Yujun Shen、Yao Yao 和 Yinghao Xu。用于流式三维重建的几何上下文变换器。载于《欧洲计算机视觉会议论文集》，第 293–314 页，2026年。1, 2, 6, 7, 8。[6] Xingyu Chen、Yue Chen、Yuliang Xiu、Andreas Geiger 和 Anpei Chen。TTT3R：将三维重建视为测试时训练。载于《国际学习表征会议》，2026年。2, 6, 7[7] Chong Cheng、Xianda Chen、Tao Xie、Wei Yin、Weiqiang Ren、Qian Zhang、Xiaoyang Guo 和 Hao Wang。LongStream：长序列流式自回归视觉几何。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 272–283 页，2026年。2[8] Leo Kaixuan Cheng、Abdus Shaikh、Ruofan Liang、Zhijie Wu、Yushi Guan 和 Nandita Vijaykumar。MERG3R：面向大规模神经视觉几何的分治方法。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 28969–28978 页，2026年。2

[9] Tri Dao、Daniel Y. Fu、Stefano Ermon、Atri Rudra 和 Christopher R´e。FlashAttention：具备 IO 感知能力的快速且内存高效的精确注意力。在《神经信息处理系统进展》中，第 16344–16359 页，2022年。5[10] Tianye Ding、Yiming Xie、Yiqing Liang、Moitreya Chatter-jee、Pedro Miraldo 和 Huaizu Jiang。LASER：用于免训练流式 4D 重建的逐层尺度对齐。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第 36433 36443 页，2026年 2 – 。[11] Sven Elflein、Ruilong Li、S´ergio Agostinho、Zan Gojcic、Laura Leal-Taix´e、Qunjie Zhou 和 Aljosa Osep。VGG-T^3：大规模离线前馈式三维重建。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第 36464–36474 页，2026年。2、6、7[12] Ziren Gong、Xiaohan Li、Fabio Tosi、Ninghui Xu、Stefano Mattoccia、Jianfei Cai 和 Matteo Poggi。MAGiSt3R：从单目 RGB 视频进行多智能体前馈3D重建。在《欧洲计算机视觉会议》中，第 417–436 页，2026年。2[13] Jiahui Huang、Qunjie Zhou、Hesam Rabeti、Aleksandr Ko-rovko、Huan Ling、Xuanchi Ren、Tianchang Shen、Jun Gao、Dmitry Slepichev、Chen-Hsuan Lin、Jiawei Ren、Kevin Xie、Joydeep Biswas、Laura Leal-Taix´e 和 Sanja Fidler。ViPE：用于 3D几何感知的视频姿态引擎。arXiv 预印本 arXiv:2508.10934，2025年。6[14] Haian Jin、Rundi Wu、Tianyuan Zhang、Ruiqi Gao、Jonathan T. Barron、Noah Snavely 和 Aleksander Hoły´nski。Zi pMap：通过测试时训练实现的线性时间有状态3D重建。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第 21748–21759 页，2026年。2、6、7[15] Seonghyun Jin 和 Jong Chul Ye。FILT3R：用于流式三维重建的潜状态自适应卡尔曼滤波器。在《欧洲计算机视觉会议》中，第 289–307 页，2026年。2[16] Bernhard Kerbl、Georgios Kopanas、Thomas Leimk¨uhler 和 George Drettakis。用于实时辐射场渲染的 3D高斯泼溅。《ACM图形学汇刊》，42（4）：139:1–139:14，2023年。1[17] Yushi Lan、Yihang Luo、Fangzhou Hong、Shangchen Zhou、Honghua Chen、Zhaoyang Lyu、Bo Dai、Shuai Yang、Chen Change Loy 和 Xingang Pan。STream3R：基于因果变换器的可扩展顺序3D重建。在《国际学习表征会议》中，2026年。1、2、3、6、7、8[18] Vincent Leroy、Yohann Cabon 和 J´erˆome Revaud。利用 MASt3R 将图像匹配锚定于 3D。在《欧洲计算机视觉会议》中，第 71–91 页，2024年。2[19]Yuhong Li、Yingbing Huang、Bowen Yang、Bharat Venkitesh、Acyr Locatelli、Hanchen Ye、Tianle Cai、Patrick Lewis 和 Deming Chen。SnapKV：大语言模型在生成前就知道你要找什么。在《神经信息处理系统进展》中，第 22947–22970 页，2024年。2[20] Zhengqi Li、Richard Tucker、Forrester Cole、Qianqian Wang、Linyi Jin、Vickie Ye、Angjoo Kanazawa、Aleksander Holyn-

ski 和 Noah Snavely。MegaSaM：从随意动态视频中进行准确、快速且稳健的结构与运动恢复。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第 10486–10496, 2025 – 页。[21] Mingkai Liu、Hao Zhao 和 Xingxing Zuo。SURE-Map：自校正流式几何基础模型。arXiv 预印本 arXiv:2609.15795, 2026年。3[22] Yuzheng Liu、Siyan Dong、Shuzhe Wang、Yingda Yin、Yan-chao Yang、Qingnan Fan 和 Baoquan Chen。SLAM3R：从单目 RGB 视频进行实时稠密场景重建。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第 16651–16662 页，2025年。2[23] Si-Yu Lu、Po-Ting Chen、Hui-Che Hsu、Sin-Ye Jhong、Wen-Huang Cheng 和 Yung-Yao Chen。OVGGT：O(1) 常数开销的流式视觉几何变换器。在《欧洲计算机视觉会议》中，第 112–129 页，2026年。2[24] Hongbo Mao、Junjun Jiang、Youyu Chen、Jiaxin Zhang、Zhe-meng Dong 和 Xianming Liu。RegVGGT：通过受约束内存实现可持续的流式视觉几何锚定。在《欧洲计算机视觉会议》中，第 1–18 页，2026年。2[25] Maxim Milakov 和 Natalia Gimelshein。softmax 的在线归一化因子计算。arXiv 预印本 arXiv:1805 02867，2018年 5 。 。[26]Ben Mildenhall、Pratul P. Srinivasan、Matthew Tancik、Jonathan T. Barron、Ravi Ramamoorthi 和 Ren Ng。NeRF：将场景表示为用于视图合成的神经辐射场。在《欧洲计算机视觉会议》中，第 405–421 页，2020年。1[27]Riku Murai、Eric Dexheimer 和 Andrew J. Davison。MASt3R-SLAM：利用 3D 重建先验进行实时稠密同步定位与建图。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第 16695–16705 页，2025年。2[28] Richard A. Newcombe、Shahram Izadi、Otmar Hilliges、David Molyneaux、David Kim、Andrew J. Davison、Pushmeet Kohli、Jamie Shotton、Steve Hodges 和 Andrew Fitzgibbon。KinectFusion：实时稠密表面映射与跟踪。在《IEEE混合与增强现实国际研讨会》中，第 127–136 页，2011年。1[29] Kerui Ren、Lihan Jiang、Tao Lu、Mulin Yu、Linning Xu、Zhangkai Ni 和 Bo Dai。Octree-GS：迈向采用基于LOD结构的三维高斯的一致实时渲染。《IEEE模式分析与机器智能汇刊》，2025年。2[30] Johannes Lutz Sch¨onberger 和 Jan-Michael Frahm。重新审视运动恢复结构。在《IEEE计算机视觉与模式识别会议论文集》中，第 4104–4113 页，2016年。1[31] Johannes Lutz Sch¨onberger、Enliang Zheng、Marc Pollefeys 和 Jan-Michael Frahm。面向非结构化多视图立体的逐像素视图选择。在《欧洲计算机视觉会议》中，第 501–518, 2016 – 页。[32] Thomas Schops、Johannes L. Schonberger、Silvano Galliani、Torsten Sattler、Konrad Schindler、Marc Pollefeys 和 Andreas Geiger。具有高分辨率图像和多摄像机视频的多视图立体基准。在《论文集》中 IEEE计算机视觉与模式识别会议，第3260–3269页，2017年。6, 8[33] Jamie Shotton、Ben Glocker、Christopher Zach、Shahram Izadi、Antonio Criminisi 和 Andrew Fitzgibbon。用于 RGB-D图像相机重定位的场景坐标回归森林。载于 IEEE计算机视觉与模式识别会议论文集，第2930–2937页，2013年。6[34] Zhijian Shu、Cheng Lin、Tao Xie、Wei Yin、Ben Li、Zhiyuan Pu、Weize Li、Yao Yao、Xun Cao、Xiaoyang Guo 和 Xiao-Xiao Long。LiteVGGT：通过几何感知缓存令牌合并增强原版 VGGT。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第36422–36432页，2026年。2[35] J¨urgen Sturm、Nikolas Engelhard、Felix Endres、Wolfram Burgard 和 Daniel Cremers。RGB-D SLAM 系统的评测基准。载于 IEEE/RSJ智能机器人与系统国际会议，第573–580页，2012年。6, 8[36] Yifu Tao、Miguel Angel Mu˜noz-Ba˜n´on、Lintong Zhang、Jia- hao Wang、Lanke Frank Tarimo Fu 和 Maurice Fallon。牛津尖塔数据集：大规模 LiDAR-视觉定位、重建与辐射场方法的基准。《国际机器人研究杂志》，45(6)：839–857，2026年。6, 7[37] Hengyi Wang 和 Lourdes Agapito。利用空间记忆进行三维重建。载于三维视觉国际会议，第78–89页，2025年。2[38] Jianyuan Wang、Minghao Chen、Nikita Karaev、Andrea Vedaldi、Christian Rupprecht 和 David Novotny。VGGT：视觉几何基础变换器。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第5294–5306页，2025年。1, 2, 6, 7[39] Jianyuan Wang、Minghao Chen、Shangzhan Zhang、Nikita Karaev、Johannes Sch¨onberger、Patrick Labatut、Piotr Bo- janowski、David Novotny、Andrea Vedaldi 和 Christian Rupprecht。VGGT-Ω。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第21486–21499页，2026年。2[40] Qianqian Wang、Yifei Zhang、Aleksander Holynski、Alexei A. Efros 和 Angjoo Kanazawa。具有持久状态的连续三维感知模型。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第10510–10522页，2025年。2, 6, 7[41] Runze Wang、Yuxuan Song、Youcheng Cai 和 Ligang Liu。STAC：面向流式三维重建的即插即用时空感知缓存压缩。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第7567–7576页，2026年。1, 3, 6, 8[42] Shuzhe Wang、Vincent Leroy、Yohann Cabon、Boris Chidlovskii 和 Jerome Revaud。DUSt3R：让几何三维视觉变得简单。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第20697–20709页，2024年。1, 2[43] Yifan Wang、Jianjun Zhou、Haoyi Zhu、Wenzheng Chang、Yang Zhou、Zizun Li、Junyi Chen、Jiangmiao Pang、Chun- hua Shen 和 Tong He。 π^3：置换等变视觉

几何学习。载于国际学习表征会议，2026年。2[44] Thomas Whelan、Stefan Leutenegger、Renato Salas Moreno、Ben Glocker 和 Andrew J. Davison。ElasticFusion：无需位姿图的稠密同步定位与建图。载于《机器人：科学与系统》论文集，2015年。1[45] Yuqi Wu、Wenzhao Zheng、Jie Zhou 和 Jiwen Lu。Point3R：利用显式空间指针记忆进行流式三维重建。载于神经信息处理系统进展，第77686–77710页，2025年。2[46] Guangxuan Xiao、Yuandong Tian、Beidi Chen、Song Han 和 Mike Lewis。利用注意力汇实现高效的流式语言模型。载于国际学习表征会议，2024年。2[47] Tao Xie、Peishan Yang、Yudong Jin、Yingfeng Cai、Wei Yin、Weiqiang Ren、Qian Zhang、Wei Hua、Sida Peng、Xiaoyang Guo 和 Xiaowei Zhou。Scal3R：面向大规模三维重建的可扩展测试时训练。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第21760–21771页，2026年。2[48] Jianing Yang、Alexander Sax、Kevin J. Liang、Mikael Henaff、Hao Tang、Ang Cao、Joyce Chai、Franziska Meier 和 Matt Feiszli。Fast3R：迈向单次前向传播的 1000+ 图像三维重建。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第21924–21935页，2025年。2[49] Shuai Yuan、Yantai Yang、Xiaotian Yang、Xupeng Zhang、Zhonghao Zhao、Lingming Zhang 和 Zhipeng Zhang。InfiniteVGGT：面向无尽流的视觉几何基础变换器。arXiv预印本 arXiv:2601.02281，2026年。1, 2, 6, 7, 8[50] Yuheng Yuan、Qiuhong Shen、Shizun Wang、Xingyi Yang 和 Xinchao Wang。Test3R：学习在测试时进行三维重建。载于神经信息处理系统进展，第111375–111394页，2025年。2[51] Shangzhan Zhang、Jianyuan Wang、Yinghao Xu、Nan Xue、Christian Rupprecht、Xiaowei Zhou、Yujun Shen 和 Gor- don Wetzstein。FLARE：从未标定稀疏视图进行前馈几何、外观与相机估计。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第21936–21947页，2025年。2[52] Xiyuan Zhang、Yanming Yang、Kaiyuan Xu、Ruibo Li 和 Chi Zhang。Slot3R：用于流式三维重建的组相联空间记忆。arXiv预印本 arXiv:2610.12282，2026 年。3[53] Zhenyu Zhang 、Ying Sheng、Tianyi Zhou、Tianlong Chen、Lianmin Zheng、Ruisi Cai、Zhao Song、Yuandong Tian、Christopher R´e、Clark Barrett、Zhangyang Wang 和 Beidi Chen。H_2O：用于大语言模型高效生成式推理的重击者预言机。载于神经信息处理系统进展，第34661–34710页，2023年。2[54] Dong Zhuo、Wenzhao Zheng、Jiahe Guo、Yuqi Wu、Jie Zhou 和 Jiwen Lu。流式视觉几何变换器。载于国际学习表征会议，2026年。1, 2, 3, 6, 7, 8[55] Zichen Zou、Xiaosong Jia、Zuxuan Wu 和 Yu-Gang Jiang。注意力本身即可进行检索。RetrieveVGGT：免训练通过查询-键相似度检索实现长上下文流式三维重建。收录于 神经信息处理系统进展，2026年。即将出版。 3