

Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis

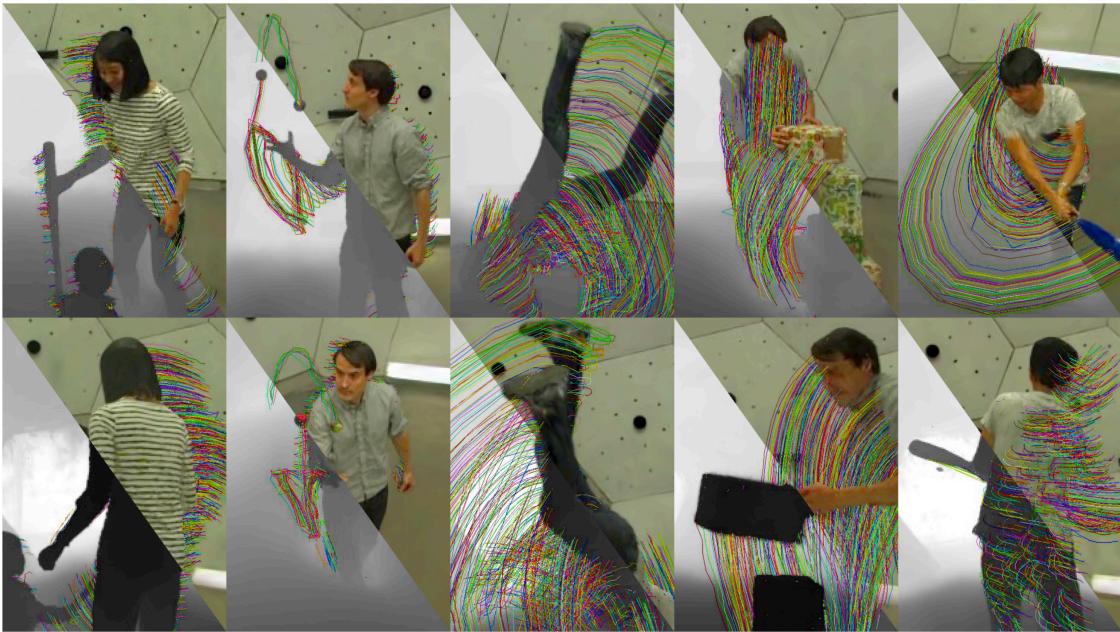


Figure 1. **Persistent Dynamic Novel-View Synthesis and Tracking Results.** Novel-view (unseen) renders of color images and depth maps across 5 scenes (columns) and 2 views (rows) at the same timestep. Each scene is parameterized by 200-300k Dynamic 3D Gaussians which move over time. We render (with occlusions) the 3D trajectories of 2.5% of these over the last 15 timesteps (0.5s). [\[Videos\]](#)

Introduction

持久动态3D世界建模在判别式和生成式人工智能中都具有变革性的潜力。在判别式方面，这能够实现场景每一部分在时间上的度量空间重建，包括物体的位置、历史运动轨迹和动态趋势。这对机器人技术、增强现实以及自动驾驶等应用至关重要。在生成式人工智能中，这种模型可以支持高分辨率动态3D资产的可控与编辑，用于电影、游戏或元宇宙内容的生成。

目前的技术还无法同时实现以下目标：对任意动态场景进行高精度轨迹跟踪、生成视觉真实的动态视角，以及实现实时训练与渲染。因此，我们提出了一种方法，将动态3D场景的重建和长时间的六自由度(6-DOF)场景跟踪任务统一到动态新视角合成的分析-通过合成框架中。

我们的主要贡献包括：

1. 扩展静态场景的3D高斯模型到动态场景。

- 利用动态3D高斯表示每个场景元素，保持其颜色、不透明度和大小随时间不变，仅允许位置和方向变化。
- 引入基于物理的局部刚性、旋转相似性和局部等距正则化约束，确保粒子间的物理一致性和轨迹稳定性。

通过我们的方法，动态场景中所有3D点的六自由度跟踪和动态重建自然产生，而无需任何外部对应关系或光流输入。

Related Work

在本文中，我们的目标是结合动态新视角合成、长时间点追踪以及动态重建任务，提出一个统一的分析-通过合成框架。以下是相关研究的分类和分析：

动态新视角合成 (Dynamic Novel-View Synthesis)

动态新视角合成领域在 NeRF 提出的背景下迅速发展。当前方法主要分为以下几类：

1. 独立时间步长表示方法

这些方法对每个时间步长分别建模，无法跨时间步获得一致的对应关系。

2. 基于欧拉表征的空间-时间网格表示

使用4D时空网格表示动态场景，例如采用平面分解或哈希函数以提高效率。然而，这些方法无法自然产生时间步之间的对应关系。

3. 基于形变场的规范时间步方法

此类方法通过形变场将规范时间步的表示映射到其他时间步，自然生成单向对应关系（通常是从某帧到参考帧的对应关系）。但也不能生成上下文对应关系，也需要高代价的反向求解，限制了动态场景的表达能力。

4. 模板引导方法

这些方法依赖预定义的模板（如人类骨架变形）建模动态场景，无法泛化到一般场景。

5. 基于点的表征方法

点云方法具有自然的拉格朗日表征特性，可以更好地捕获时间上的对应关系。然而，由于基于点的渲染性能不如基于网格或 MLP 的方法，这类方法较少受到关注。

我们的方法属于最后一类，但采用了动态3D高斯作为基础单元，使其具有旋转表征能力，并能直接重建所有3D点的六自由度运动。

长时间点追踪 (Long-Term Point Tracking)

传统的视频追踪算法通常针对以下两种任务：

1. 物体级别的追踪

使用边界框或分割掩码进行目标追踪【公式略】。

2. 稠密场景点的短期追踪

光流或场景流方法通常只能处理两个时间步之间的点追踪。

最近，一些方法尝试解决长时间的稠密点追踪问题，例如 OmniMotion 方法，但其依赖于每对时间步之间的光流作为优化目标，计算代价高且受限于输入数据的质量。

相比之下，我们的方法无需任何外部对应关系作为输入，追踪结果自然从持久的动态3D表示中涌现。

动态重建 (Dynamic Reconstruction)

针对动态重建的研究通常依赖以下假设：

1. 使用准确的深度摄像机。
2. 假设已有目标点云作为输入。
3. 针对特定场景（如驾驶场景中的移动车辆）的重建。

现有的最相似方法需要依赖大量摄像机输入或预算算的光流，而我们的方法仅使用少量摄像机数据且无需额外的预处理。

总结：现有工作中，动态新视角合成、长时间点追踪和动态重建通常被单独研究。我们首次将这些任务统一到一个基于动态3D高斯表征的框架中。

Method

Dynamic 3D Gaussians. Our dynamic scene representation (\mathcal{S}) is parameterized by a set of Dynamic 3D Gaussians, each of which has the following parameters:

- 1) a 3D center for each timestep (x_t, y_t, z_t).
- 2) a 3D rotation for each timestep parameterized by a quaternion (qw_t, qx_t, qy_t, qz_t).
- 3) a 3D size in standard deviations (consistent over all timesteps) (sx, sy, sz)
- 4) a color (consistent over all timesteps) (r, g, b)
- 5) an opacity logit (consistent over all timesteps) (o)
- 6) a background logit (consistent over timesteps) (bg)

这段描述的重建过程是一个 测试时优化 + 时间步在线优化 的方法，关键步骤如下：

1. 在 第一个时间步 完成场景属性的全面优化。
2. 后续时间步通过前一时间步的表示初始化，并仅优化与运动相关的属性。
3. 每个时间步的优化通过 可微渲染器 完成，将 3D 场景表示转化为与训练图像一致的渲染结果。

这种方法具有以下优势：

- **时间一致性：**通过时间步的逐步优化，自然捕捉场景的动态变化。
- **高效性：**仅优化运动属性，减少了参数维度，提升了计算速度。
- **灵活性：**无需额外训练数据，可以针对特定场景灵活调整。

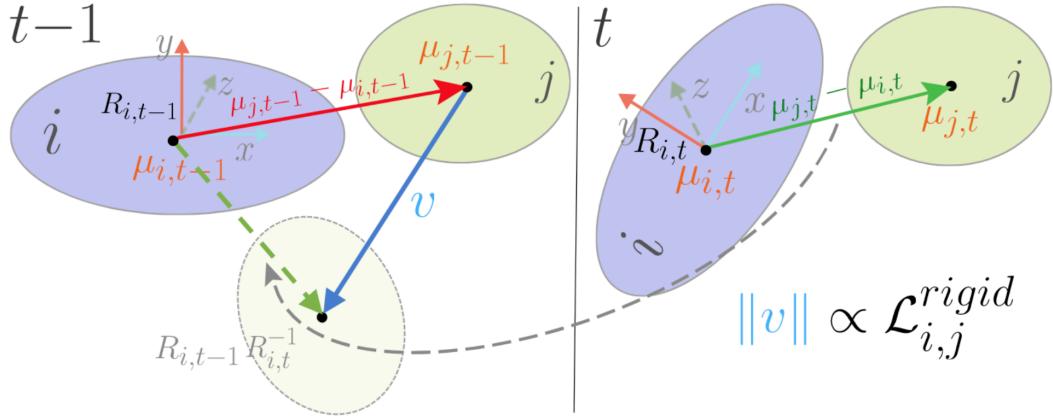


Figure 4. **Local Rigidity Loss.** For each Gaussian i , nearby Gaussians j should move in a way that follows the rigid-body transform of the coordinate system of i between timesteps.

动态3D高斯的数学表示

动态3D高斯定义为一个时间步 t 的3D空间点 $p = [x, y, z]^\top$ 的概率分布，其核函数为：

$$f_{i,t}(p) = \text{sigm}(o_i) \cdot \exp\left(-\frac{1}{2}(p - \mu_{i,t})^\top \Sigma_{i,t}^{-1}(p - \mu_{i,t})\right) \quad (1)$$

1. 中心点表示

高斯中心 $\mu_{i,t}$ 是粒子 i 在时间步 t 的位置：

$$\mu_{i,t} = \begin{bmatrix} x_{i,t} \\ y_{i,t} \\ z_{i,t} \end{bmatrix} \quad (2)$$

2. 协方差矩阵表示

协方差矩阵 $\Sigma_{i,t}$ 表示高斯核的形状和方向，由旋转矩阵 $R_{i,t}$ 和缩放矩阵 S_i 组成：

$$\Sigma_{i,t} = R_{i,t} S_i S_i^\top R_{i,t}^\top \quad (3)$$

2.1 缩放矩阵

$$S_i = \text{diag}(\sigma_{x,i}, \sigma_{y,i}, \sigma_{z,i}) \quad (4)$$

2.2 旋转矩阵

通过四元数 $q_{i,t} = [q_{w,i,t}, q_{x,i,t}, q_{y,i,t}, q_{z,i,t}]$ 构造旋转矩阵 $R_{i,t}$:

$$R_{i,t} = \text{q2R}(q_{i,t}) = \begin{bmatrix} 1 - 2q_{y,i,t}^2 - 2q_{z,i,t}^2 & 2q_{x,i,t}q_{y,i,t} - 2q_{z,i,t}q_{w,i,t} & 2q_{x,i,t}q_{z,i,t} + 2q_{y,i,t}q_{w,i,t} \\ 2q_{x,i,t}q_{y,i,t} + 2q_{z,i,t}q_{w,i,t} & 1 - 2q_{x,i,t}^2 - 2q_{z,i,t}^2 & 2q_{y,i,t}q_{z,i,t} - 2q_{y,i,t}q_{w,i,t} \\ 2q_{x,i,t}q_{z,i,t} - 2q_{y,i,t}q_{w,i,t} & 2q_{y,i,t}q_{z,i,t} + 2q_{x,i,t}q_{w,i,t} & 1 \end{bmatrix}$$

可微渲染的数学推导

动态高斯通过摄像机的投影矩阵渲染到2D图像平面，步骤如下：

1. 3D中心投影到2D

给定相机的内参矩阵 K 和外参矩阵 E , 3D点 $\mu_{i,t}$ 投影到2D平面上:

$$\mu_{i,2D} = K \cdot \frac{E\mu_{i,t}}{(E\mu_{i,t})_z} \quad (6)$$

其中, $\mu_{i,t}$ 是3D点, E 是世界到相机的变换矩阵, K 是相机内参矩阵, $(E\mu_{i,t})_z$ 是深度。

2. 协方差矩阵投影到2D

3D协方差矩阵 $\Sigma_{i,t}$ 通过投影矩阵转换为2D协方差矩阵 $\Sigma_{i,2D}$:

$$\Sigma_{i,2D} = J \cdot E \cdot \Sigma_{i,t} \cdot E^\top \cdot J^\top \quad (7)$$

其中, J 是投影公式 $\mu_{i,2D}$ 对 $\mu_{i,t}$ 的雅可比矩阵:

$$J = \frac{\partial \mu_{i,2D}}{\partial \mu_{i,t}} \quad (8)$$

3. 颜色融合

每个像素的颜色通过深度优先的渲染公式计算:

$$C_{\text{pix}} = \sum_{i \in S} c_i f_{i,2D} \prod_{j=1}^{i-1} (1 - f_{j,2D}) \quad (9)$$

其中：

- $f_{i,2D}$ 是2D高斯核的影响值，计算方式类似 $f_{i,t}(p)$ 。
 - $c_i = [r_i, g_i, b_i]$ 是高斯的颜色。
-

基于物理的正则化公式

We restrict the set of Gaussians j to be the k-nearest-neighbours of i ($k=20$), and weight the loss by the a weighting factor for the Gaussian pair:

$$w_{i,j} = \exp\left(-\lambda_w \|\mu_{j,0} - \mu_{i,0}\|_2^2\right)$$

which is an (unnormalized) isotropic Gaussian weighting factor. We set λ_w to 2000, which gives a standard deviation of $\sim 2.2\text{cm}$, and calculate this with the distance between the Gaussian centers in the first timestep and fix it over the rest of the timesteps. This results in the rigidity loss only being enforced locally, while still allowing global non-rigid reconstruction.

加权因子和局部刚性损失

1. 限制为最近邻的高斯粒子集合

我们将高斯粒子 j 的集合限制为 i 的 k 个最近邻粒子，其中 $k = 20$ 。即：

$$j \in \text{knn}_i \quad (\text{k-nearest neighbors}) \tag{10}$$

- 目的：

1. 减少计算开销，只计算与粒子 i 空间距离最近的 k 个粒子。
 2. 遵循局部刚性假设，即刚性约束仅在相邻粒子间有效。
-

2. 损失的加权因子

损失计算时，采用一个基于距离的加权因子 $w_{i,j}$ ，其公式为：

$$w_{i,j} = \exp\left(-\lambda_w \|\mu_{j,0} - \mu_{i,0}\|_2^2\right) \tag{11}$$

- 公式解析：

- $w_{i,j}$ 是粒子 i 和 j 的加权系数。
- $\mu_{i,0}$ 和 $\mu_{j,0}$ 分别表示粒子 i 和 j 在初始时间步 ($t = 0$) 的中心位置。
- $\|\mu_{j,0} - \mu_{i,0}\|_2^2$ 表示两粒子间的欧氏距离平方。
- λ_w 是控制距离衰减速率的超参数。

高斯加权的意义

1. 局部性保证：

- 距离越远，加权因子 $w_{i,j}$ 越小，从而减弱远处粒子的影响。
- 确保刚性约束只在局部范围内有效。

2. 平滑性：

- $w_{i,j}$ 是一个等方性 (isotropic) 的高斯分布，对方向不敏感，保证了约束的平滑性。

3. 加权因子的超参数设置

- 公式中的参数 λ_w 设置为：

$$\lambda_w = 2000 \quad (12)$$

- 对应的标准差：该值对应于高斯分布的标准差约为 ~ 2.2 cm。

作用

- 控制衰减范围： λ_w 决定高斯权重的影响范围。较大的 λ_w 会使距离较远的粒子贡献减弱，确保刚性约束的局部性。
- 柔性调整：通过调整 λ_w ，可以控制局部刚性的约束强度。

4. 初始时间步的加权计算

我们使用初始时间步 ($t = 0$) 的高斯粒子中心位置来计算距离，并将计算得到的加权因子 $w_{i,j}$ 固定在整个时间序列中：

$$w_{i,j} = \exp(-\lambda_w \|\mu_{j,0} - \mu_{i,0}\|_2^2) \quad (\text{固定值}) \quad (13)$$

- 原因：
 1. **计算效率**: 只需在初始时间步计算一次加权因子，后续时间步直接使用，减少了动态计算的开销。
 2. **时间一致性**: 固定的加权因子可以避免粒子动态变化导致的约束不稳定。
-

5. 局部刚性与全局非刚性

最终的刚性损失只在局部范围内施加，从而实现：

- **局部刚性**: 通过 k 个最近邻粒子和加权因子 $w_{i,j}$ 的约束，使得局部粒子间的相对运动保持刚性。
 - **全局非刚性**: 刚性约束只作用于局部范围，允许场景在全局范围内表现出非刚性动态形变。
-

总结

加权因子的设计通过距离和高斯分布实现：

1. **局部性控制**: 仅约束相邻粒子之间的相对运动，确保局部刚性。
2. **高效计算**: 基于初始时间步的距离计算加权因子并固定，减少动态计算开销。
3. **柔性调整**: 通过 λ_w 控制约束强度，支持不同的动态场景建模需求。

如果需要进一步细化公式推导或实例应用，请继续说明！

正则化约束确保动态高斯的轨迹和形状符合物理规律：

1. 局部刚性损失

每个高斯 i 的运动与其邻域 j 的刚体变换一致：

$$L_{\text{rigid}} = \quad (14)$$

$$\frac{1}{k|S|} \sum_{i \in S} \sum_{j \in \text{knn}_i} w_{i,j} \|(\mu_{j,t-1} - \mu_{i,t-1}) - R_{i,t-1} R_{i,t}^{-1} (\mu_{j,t} - \mu_{i,t})\|_2^2 \quad (15)$$

$$\mathcal{L}^{\text{rigid}} = \frac{1}{k|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{j \in \text{knn}_{i;k}} \mathcal{L}_{i,j}^{\text{rigid}}$$

2. 旋转相似性损失

相邻高斯的旋转保持一致：

$$L_{\text{rot}} = \frac{1}{k|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{j \in \text{knn}_i} w_{i,j} \|\hat{q}_{j,t} \hat{q}_{j,t-1}^{-1} - \hat{q}_{i,t} \hat{q}_{i,t-1}^{-1}\|_2^2 \quad (16)$$

3. 局部等距损失

在所有时间步中，高斯之间的距离保持一致：

$$L_{\text{iso}} = \frac{1}{k|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{j \in \text{knn}_i} w_{i,j} |\|\mu_{j,0} - \mu_{i,0}\|_2 - \|\mu_{j,t} - \mu_{i,t}\|_2| \quad (17)$$

4. 损失权重

权重 $w_{i,j}$ 定义为高斯 i 和 j 初始距离的高斯加权：

$$w_{i,j} = \exp(-\lambda_w \|\mu_{j,0} - \mu_{i,0}\|_2^2) \quad (18)$$

其中， λ_w 控制权重的衰减范围。

这些公式进一步揭示了动态3D高斯的优化细节及其物理合理性。如果需要深入实验部分的推导或应用场景，请继续说明。

Experiment

数据集准备 (Dataset Preparation)

我们使用了 **CMU Panoptic Studio** 数据集，并构建了一个名为 **PanopticSports** 的子集。数据集准备的步骤如下：

1. 序列选择
 - 从 Panoptic Studio 数据集中选择六个子序列，包含复杂的动态场景，分别命名为：*juggle, box, softball, tennis, football, basketball*。
 - 每个序列包含 150 帧，帧率为 30 FPS。
2. 摄像机设置
 - 使用 31 个摄像机中的 27 个作为训练摄像机，其余 4 个用于测试。
 - 提供了相机的内参和外参，用于视角对齐。
3. 图像预处理
 - 图像经过失真矫正，尺寸调整为 640×360 。
4. 伪真值生成
 - 初始点云：从深度相机采样，结合训练摄像机投影生成。
 - 前景-背景分割：通过帧差分生成伪真值分割掩码。
5. 轨迹真值生成
 - 使用 CMU 数据集提供的高质量面部和手部关键点，手动筛选出 21 条真实 3D 轨迹。

评估指标 (Evaluation Metrics)

1. 新视角合成 (Novel View Synthesis)

使用以下标准评估从未见视角生成的图像质量：

- **PSNR (峰值信噪比)**：衡量图像与参考图像的差异。
- **SSIM (结构相似性指数)**：衡量图像的结构相似性。
- **LPIPS (感知相似性)**：基于感知特征的距离。

2. 3D 长时间点追踪 (3D Long-Term Point Tracking)

在 3D 空间中对点的长时间追踪，采用以下指标：

- **MTE (中值轨迹误差)**：点的 3D 位置与伪真值的中值欧氏距离。

- δ 精度：指定误差阈值下，预测点与真值点的比例。
- 生存率 (Survival Rate)：轨迹始终保持在一定距离范围内的比例。

3. 2D 长时间点追踪 (2D Long-Term Point Tracking)

将 3D 点投影到 2D 图像平面，计算以下指标：

- MTE：2D 像素误差。
- δ 精度：与 3D 追踪相同。
- 生存率：与 3D 追踪相同。

Task	Metrics	Method	Juggle	Boxes	Softball	Tennis	Football	Basketball	Mean
View Synthesis	PSNR↑	3GS-O [17] Ours	28.19 29.48	28.74 29.46	28.77 28.43	28.03 28.11	28.49 28.49	27.02 28.22	28.21 28.7
	SSIM↑	3GS-O [17] Ours	0.91 0.92	0.91 0.91	0.91 0.91	0.90 0.91	0.90 0.91	0.89 0.91	0.90 0.91
	LPIPS↓	3GS-O [17] Ours	0.15 0.15	0.15 0.17	0.14 0.19	0.16 0.17	0.16 0.19	0.18 0.18	0.16 0.17
3D Tracking	3D MTE↓	3GS-O [17] Ours	32.81 1.90	39.95 1.97	64.94 2.02	75.54 2.33	45.57 2.45	76.71 2.56	55.9 2.21
	3D δ ↑	3GS-O [17] Ours	13.6 77.2	3.5 75.9	5.9 70.3	4.2 69.0	9.8 69.4	3.5 66.3	6.8 71.4
	3D Surv↑	3GS-O [17] Ours	56.3 100	60.8 100	37.2 100	16.9 100	59.6 100	31.9 100	43.8 100
2D Tracking	2D MTE↓	3GS-O [17] PIPS [12] Ours	23.86 5.76 1.54	29.88 8.42 1.42	51.6 13.3 1.69	58.15 21.0 1.36	35.15 23.2 1.48	64.29 22.6 1.93	43.8 15.7 1.57
	2D δ ↑	3GS-O [17] PIPS [12] Ours	17.1 55.9 80.4	10.5 39.5 82.5	8.9 37.0 77.3	6.5 28.4 80.2	15.0 43.5 79.7	7.2 33.2 73.9	10.9 39.6 78.4
	2D Surv↑	3GS-O [17] PIPS [12] Ours	71.3 91.6 100	74.4 61.3 100	42.7 88.6 100	23.0 72.2 100	69.6 79.8 100	47.1 77.6 100	54.7 79.0 100

Table 1. Results on our prepared PanopticSports dataset. See text for details on the dataset, metrics, tasks and methods.

Method	PSNR↑	SSIM↑	LPIPS↓
TiNeuVox-S [9]	26.64	0.92	0.14
TiNeuVox [9]	27.28	0.91	0.13
InstantNGP [26]	24.69	0.91	0.12
Particle-NeRF [1]	27.47	0.94	0.08
Ours	39.49	0.99	0.02

Table 2. Result on the Particle-NeRF dataset. See text for details on the dataset, metrics, tasks and methods.

实验结果与分析 (Results and Analysis)

1. PanopticSports 数据集结果

(1) 新视角合成

我们的方法与现有的 3D Gaussian Splatting 方法 (3GS-O) 相比，性能有所提升：

- **PSNR**: 从 28.21 提升到 28.7。
- **SSIM**: 从 0.90 提升到 0.91。
- **LPIPS**: 保持在 0.16 左右。

(2) 3D 追踪

- **MTE**: 我们的方法实现了 2.21cm 的中值误差，比 3GS-O 的 55.9cm 大幅降低。
- δ 精度: 在 2cm 阈值下，我们达到了 71.4%。
- 生存率: 保持在 100%。

(3) 2D 追踪

- **MTE**: 我们的误差为 1.57 像素，远优于现有方法 15.7 像素。
 - δ 精度: 达到 78.4%。
 - 生存率: 保持在 100%。
-

2. Particle-NeRF 数据集结果

我们的方法在简单的合成数据集上几乎达到了完美表现：

- **PSNR**: 39.49。
- **SSIM**: 0.99。
- **LPIPS**: 0.02。

相比其他方法（如 Particle-NeRF 和 TiNeuVox），我们的结果在所有指标上都有显著提升。

3. 消融实验 (Ablation Study)

在 *juggle* 场景中，对方法的关键组件进行了消融分析，结果如下：

1. 局部刚性损失 (LRigid)

去除此项后，3D追踪误差从 1.90cm 增至 4.32cm，生存率下降。

2. 旋转相似性损失 (LRot)

去除后影响较小，但视觉效果有所下降。

3. 局部等距损失 (LIso)

去除后导致点追踪的长时间一致性较差。

4. 背景分割损失 (LBg)

去除后追踪性能大幅下降，PSNR 降至 24.14。

5. 参数固定 (Fixing Parameters)

如果不固定高斯的不透明度、大小和颜色，则 PSNR 降至 27.14。

6. 前向传播初始化 (Forward Propagation)

缺乏前向初始化导致优化不稳定，3D误差增加至 6.32cm。

总结 (Summary)

我们的实验表明，基于动态3D高斯的表征不仅在新视角合成任务中表现优异，还在长时间点追踪中实现了前所未有的准确性和稳定性。通过对物理约束的全面分析，验证了各组件对最终性能的贡献。