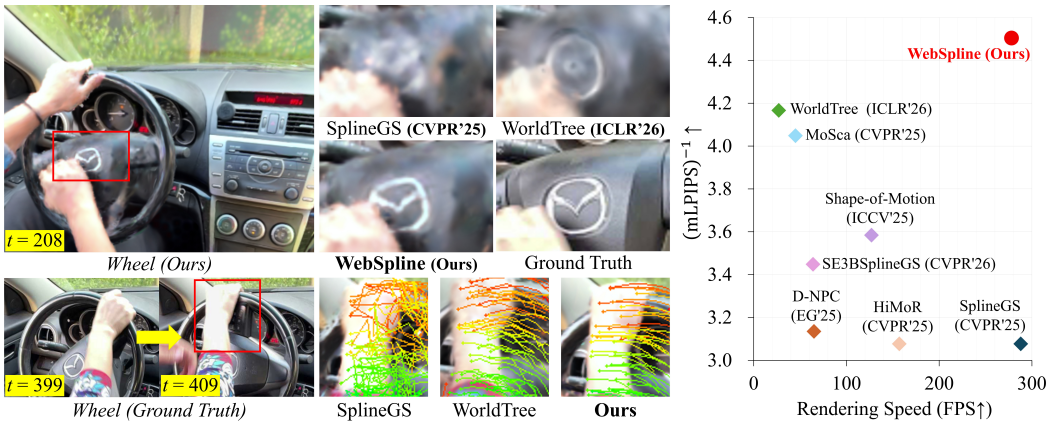
WebSpline: Structure-Informed Splines for Real-Time 3D Gaussians from Monocular Videos

Jongmin Park* **Jeonghwan Yun*** **Minh-Quan Viet Bui** **Munchurl Kim†**
KAIST
{jm.park, jeonghwan.yun, bvmquan, mkimee}@kaist.ac.kr



(a) Visual comparisons on the iPhone dataset (b) Performance gain (iPhone)

Figure 1: **WebSpline achieves high-quality dynamic scene reconstruction with fast rendering from monocular videos.** (a) Qualitative comparison with state-of-the-art methods on novel view synthesis (top) and visualized Gaussian trajectories (bottom). (b) WebSpline achieves the best rendering quality while rendering over 10× faster than the second-best method.

Abstract

Dynamic scene reconstruction from monocular videos remains highly challenging, as existing methods often struggle to balance global structural coherence and local fine-grained details under limited multi-view cues. To address this challenge, we propose WebSpline, a novel dynamic 3D Gaussian framework that enables structurally coherent and high-fidelity reconstruction from monocular videos with fast rendering. The core of WebSpline is the Structure-Informed Spline (SIS) representation, which models each dynamic Gaussian trajectory using a learnable cubic Hermite spline whose motion is structurally organized with an auxiliary Structural Proxy Graph (SPG). The proposed framework is optimized in two stages: (i) the SPG is initialized from 2D point tracks and refined with temporal rigidity regularization to establish structural coherence for moving objects across the sequence; and (ii) the SIS representation is initialized from the refined SPG and optimized under both spatial and structural neighborhood constraints. At inference, Gaussian motion is obtained solely by evaluating the learned SIS, preserving the efficiency of spline-based motion representations. Experiments on the challenging monocular dynamic scene benchmarks, iPhone and NVIDIA, demonstrate that our WebSpline achieves **state-of-the-art rendering quality** while rendering over **10× faster** than WorldTree, the second-best method on the iPhone dataset.

*Equal contribution. † Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Dynamic scene reconstruction from monocular videos is a fundamental task in 3D vision, with broad applications in virtual reality (VR), robotics, and autonomous driving. High-fidelity reconstruction in this setting requires modeling complex object deformations and time-varying geometry from limited observations while preserving spatio-temporal consistency. Recently, 3D Gaussian Splatting (3DGS) [15] has established a powerful paradigm for scene reconstruction by representing a scene with learnable 3D Gaussians and rendering them via differentiable splatting, which enables high-quality, real-time rendering. However, since 3DGS [15–17, 53, 52] is primarily designed for static scenes, extending it to monocular dynamic scene reconstruction remains highly challenging.

To tackle these challenges, several recent methods [45, 10, 49, 21, 44, 34, 43, 5, 29, 28] have extended 3DGS [15] to dynamic scene reconstruction for monocular videos by modeling the time-varying deformations of 3D Gaussians. Recent state-of-the-art (SOTA) methods can be categorized by their Gaussian motion modeling strategies: (i) scaffold-based methods [21, 44, 54] and (ii) primitive-motion-parameterized methods [34, 43]. The scaffold-based methods [21, 44, 54] model Gaussian motions through scaffold structures, where the trajectory of each Gaussian is determined by blending the motions of neighboring scaffolds over time. While effective in preserving structural consistency, such blending often introduces over-smoothing artifacts, resulting in blurry renderings and degraded perceptual quality. In contrast, the primitive-motion-parameterized methods [34, 43] model per-Gaussian motion using learnable trajectory parameters such as spline functions [34] or $SE(3)$ motion coefficients [43]. These methods can produce sharper renderings, but the motion of each Gaussian is less constrained, often leading to geometric instability for moving objects.

To address these limitations, we propose WebSpline, a novel dynamic Gaussian framework for structurally coherent and high-fidelity monocular dynamic scene reconstruction with fast rendering. At the core of WebSpline is a structurally organized spline representation that models each dynamic Gaussian with flexible per-primitive trajectories while leveraging an auxiliary graph structure during training. WebSpline consists of two components: (i) a Structural Proxy Graph (SPG) and (ii) a Structure-Informed Spline (SIS) representation, optimized through a two-stage framework. In the first stage, we construct the SPG initialized from 2D point tracks and refine it to serve as a robust structural foundation. To achieve this, we introduce an orientation-similarity-based temporal rigidity loss to refine the SPG node trajectories for structural coherence across the sequence. In the second stage, guided by the refined SPG, the SIS is introduced for parameterized representation of each dynamic Gaussian trajectory using a cubic Hermite spline [1, 7] with learnable control points. Within SIS, each dynamic Gaussian is associated with a corresponding SPG node, aligning its spline trajectory with the structurally organized SPG node trajectory. During SIS optimization, we define two types of neighborhoods, spatial and structural, and impose temporal rigidity constraints over these neighborhoods. These rigidity constraints stabilize SIS learning by preserving both local geometric consistency and structural coherence over time, while retaining the flexibility needed to capture fine-grained non-rigid motion. At inference, the SPG is no longer required, and Gaussian motion is obtained solely by evaluating the learned SIS, preserving the efficiency of spline-based motion representations. Notably, while SplineGS [34] also represents dynamic Gaussian motion with spline trajectories, WebSpline incorporates spatial and structural neighborhood constraints, thereby reducing noisy artifacts and improving geometric consistency. Extensive experiments show that WebSpline significantly outperforms the SOTA dynamic Gaussian methods, as shown in Table 1 and Figs. 1, 3, and 4. WebSpline achieves an mPSNR gain of **+0.36 dB** on the iPhone dataset [9] and a PSNR gain of **+0.61 dB** on the NVIDIA dataset [50] over the respective second-best methods. Furthermore, WebSpline enables **10× faster rendering** than WorldTree [44], the second-best method on the iPhone dataset. Our contributions are as follows:

- We propose **WebSpline**, an approach for dynamic 3D Gaussians that improves structural coherence and reconstruction fidelity while preserving the efficient inference of spline-based motion representations.
- We introduce the **Structure-Informed Spline (SIS)**, which parameterizes fine-grained Gaussian trajectories under the structural organization of a **Structural Proxy Graph (SPG)** for structurally coherent dynamic motion modeling.
- We present a two-stage optimization strategy that first structurally refines the SPG node trajectories, then uses these refined trajectories to initialize the SIS and guide its optimization.

- Our WebSpline achieves SOTA rendering quality on challenging monocular benchmarks while retaining spline-based rendering efficiency, improving the quality–efficiency trade-off. On the iPhone dataset, it renders over $\mathbf{10}\times$ faster than WorldTree, the second-best method in rendering quality.

2 Related Work

3D Scene Reconstruction. The emergence of neural radiance fields (NeRF) [32, 3, 4] has significantly advanced 3D scene reconstruction and novel view synthesis. NeRF [32, 3, 4] represents continuous 3D scenes using an MLP-based implicit representation and renders them via volumetric rendering. This process requires dense ray sampling and repeated MLP evaluations along each ray, resulting in substantial computation overhead during both training and inference. Although numerous NeRF-based approaches [6, 39, 42, 8, 33, 51] have been proposed to accelerate rendering, they often struggle to achieve a favorable balance between rendering fidelity and efficiency. Recently, 3D Gaussian Splatting (3DGS) [15] has emerged as an explicit scene representation based on anisotropic 3D Gaussians, together with a differentiable rasterization pipeline. This design enables real-time rendering while maintaining high-quality scene reconstruction. However, 3DGS [15] is primarily designed for static scene reconstruction and lacks a temporal modeling capability, which limits its applicability for dynamic scene reconstruction.

Dynamic 3D Gaussian Splatting. To extend 3DGS to dynamic scenes, prior works have explored diverse strategies, including canonical-space methods [49, 26, 41, 45, 22, 2], explicit-state methods [31, 23, 20], and structured motion learning methods [10, 19, 25, 38]. More recently, SOTA methods [21, 44, 54, 43, 34] tailored for monocular video settings have been proposed. These methods incorporate various priors, such as optical flows [40, 46], monocular depths [36, 47, 27, 48], motion masks [18], or 2D point tracks [12, 13], to compensate for the limited multi-view cues and improve geometric accuracy. These methods can be divided into two groups: (i) scaffold-based methods [21, 44, 54] and (ii) primitive-motion-parameterized methods [34, 43]. For the scaffold-based methods, MoSca [21] optimizes a scaffold architecture from 2D point tracks and represents Gaussian trajectories by blending the motions of neighboring nodes within the scaffold. WorldTree [44] builds upon the scaffold-based framework of MoSca [21] by adopting a hierarchical tree-chain structure. SE3BSplineGS [54] extends MoSca [21] by adopting B-spline basis functions [7]. However, as these methods rely on motion blending across scaffold nodes, the rendering results tend to be blurry, and the inference speed is relatively slow. For the primitive-motion-parameterized methods, SoM [43] decomposes Gaussian motion into a set of shared $SE(3)$ motion bases and optimizes coefficients for each Gaussian primitive. SplineGS [34] parameterizes each Gaussian trajectory using a cubic Hermite spline function [1, 7] to ensure smooth and continuous motion while enabling high rendering speed. However, as these approaches lack explicit spatial constraints between primitives, they often struggle to maintain structural coherence and may produce noisy artifacts. Unlike these methods, our proposed WebSpline integrates a Structural Proxy Graph (SPG) as a robust structural motion foundation with a Structure-Informed Spline (SIS) representation for flexible per-primitive trajectory modeling. This design achieves structural coherence while preserving rich motion details and enabling fast rendering.

3 Method

3.1 Overview of WebSpline

Fig. 2 illustrates WebSpline, a two-stage framework that first establishes a structurally coherent motion foundation and then uses it to initialize and optimize fine-grained per-Gaussian spline trajectories. Given a monocular video $\{\boldsymbol{I}_t\}_{t=1}^{N_f}$, we represent the scene with static Gaussians $\mathcal{G}_{\text{st}}$ and dynamic Gaussians $\mathcal{G}_{\text{dy}}$, with $N_g = |\mathcal{G}_{\text{dy}}|$. In the first stage, *Structural Proxy Graph (SPG) foundation*, we lift 2D point tracks into 3D to initialize an SPG $\mathcal{K}$ and refine its node trajectories with temporal rigidity regularization. This produces a refined SPG $\tilde{\mathcal{K}}$ whose trajectories provide a structurally coherent motion foundation. In the second stage, *Structure-Informed Spline (SIS) optimization*, we initialize each dynamic Gaussian's cubic Hermite spline by least-squares fitting to its corresponding refined SPG node trajectory. We then optimize the SIS under spatial neighborhood constraints between nearby Gaussians and structural neighborhood constraints defined by the refined SPG $\tilde{\mathcal{K}}$. Thus, the

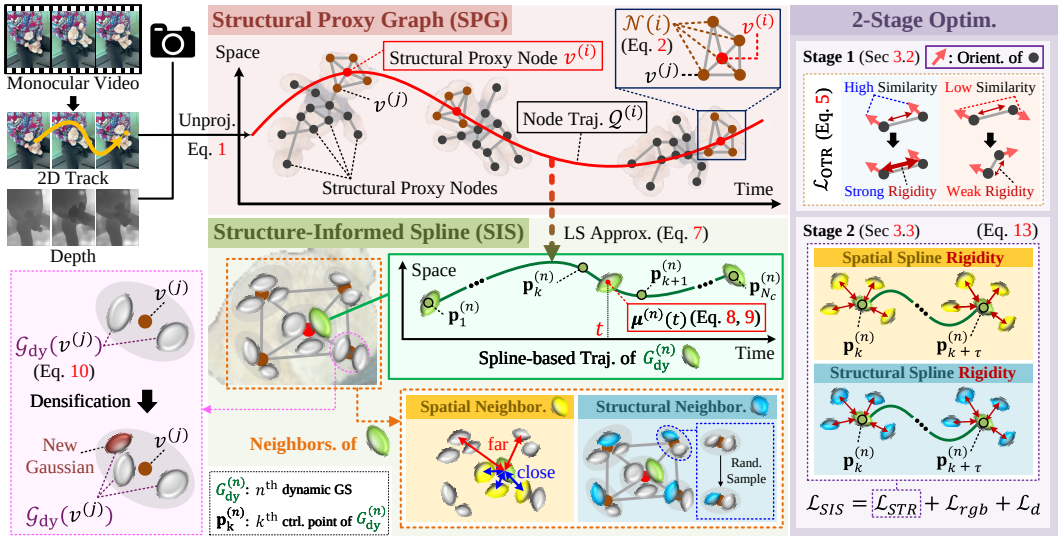


Figure 2: **Overview of WebSpline.** WebSpline models each dynamic Gaussian trajectory using the Structure-Informed Spline (SIS) representation, initialized by fitting to the refined node trajectories of the refined Structural Proxy Graph (SPG). For SIS optimization, we define two types of neighborhoods for each dynamic Gaussian, spatial and structural, to enforce coherent spline motion while capturing fine-grained dynamics.

refined SPG $\tilde{\mathcal{K}}$ has two distinct roles: it initializes the SIS with refined trajectories and provides structural neighborhoods for subsequent regularization. At inference, only the learned splines are evaluated, preserving the efficiency of spline-based motion representations.

3.2 Stage 1: SPG Foundation

SPG Initialization. To improve structural coherence in spline-based dynamic Gaussian representation of our WebSpline, we introduce the SPG $\mathcal{K} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of M structural proxy nodes and $\mathcal{E}$ denotes the edge set connecting neighboring nodes. As depicted in Fig. 2, the SPG provides a structural motion representation of dynamic components in a scene, with each structural proxy node $v^{(i)} \in \mathcal{V}$ representing a local dynamic region through its trajectory. Each $v^{(i)}$ is associated with a learnable structural proxy node trajectory $\mathcal{Q}^{(i)} = \{\mathbf{q}_t^{(i)}\}_{t=1}^{N_f}$ and a learnable auxiliary orientation sequence $\mathcal{O}^{(i)} = \{\mathbf{O}_t^{(i)}\}_{t=1}^{N_f}$, where $\mathbf{q}_t^{(i)} \in \mathbb{R}^3$ denotes its position and $\mathbf{O}_t^{(i)} \in SO(3)$ denotes its auxiliary orientation at time t. To initialize $\mathcal{Q}^{(i)}$, we first lift 2D point tracks extracted from dynamic regions of the scene into 3D using camera extrinsics and metric depth. Let $\mathcal{T}^{(i)} = \{\mathbf{u}_t^{(i)}\}_{t=1}^{N_f}$ denote the ith 2D point track, where $\mathbf{u}_t^{(i)}$ is the tracked pixel coordinate at time t. Given the metric depth $\boldsymbol{d}_t \in \mathbb{R}^{H \times W}$ and camera extrinsics $[\boldsymbol{R}_t|\boldsymbol{T}_t] \in \mathbb{R}^{3\times4}$ at time t, we initialize the 3D position of each structural proxy node by lifting the tracked pixel $\mathbf{u}_t^{(i)}$ as

$$\mathbf{q}_t^{(i)} = \boldsymbol{R}_t^\top \pi_{\boldsymbol{K}}^{-1}(\mathbf{u}_t^{(i)}, \boldsymbol{d}_t(\mathbf{u}_t^{(i)})) - \boldsymbol{R}_t^\top \boldsymbol{T}_t, \tag{1}$$

where $\pi_{\boldsymbol{K}}^{-1}$ denotes the inverse projection from image to camera coordinates under intrinsics $\boldsymbol{K}$. The initialization of $\mathcal{O}^{(i)}$ is described in Appendix B.1. We then determine $\mathcal{E}$ using a pairwise trajectory distance matrix $\mathcal{D} \in \mathbb{R}^{M\times M}$, where each entry is defined as $\mathcal{D}_{ij} = Q_{1-\eta}(\{\|\mathbf{q}_t^{(i)} - \mathbf{q}_t^{(j)}\|_2\}_{t=1}^{N_f})$. Here, $Q_{1-\eta}$ denotes the $(1-\eta)$-quantile over time. The neighborhood $\mathcal{N}(i)$ of each structural proxy node $v^{(i)}$ and the edge set $\mathcal{E}$ are defined as

$$\mathcal{N}(i) = \{j \in \{1,\dots,M\} \mid j \neq i,\ \text{rank}_i(\mathcal{D}_{ij}) \leq K\}, \quad \mathcal{E} = \bigcup_{i=1}^{M} \{(v^{(i)}, v^{(j)}) \mid j \in \mathcal{N}(i)\}, \tag{2}$$

where K denotes the number of neighboring nodes of $v^{(i)}$ and $\text{rank}_i(\mathcal{D}_{ij})$ denotes the ascending rank of $\mathcal{D}_{ij}$ among all distances from $v^{(i)}$ to others.

SPG Structural Refinement. We refine the SPG node trajectories initialized above to establish structural coherence over time among proxy nodes representing dynamic regions. This coherent SPG enables stable SIS learning in the subsequent SIS optimization stage. To this end, we introduce an orientation-similarity-based temporal rigidity (OTR) loss $\mathcal{L}_{\text{OTR}}$. $\mathcal{L}_{\text{OTR}}$ encourages temporal rigidity among neighboring nodes that exhibit similar rigid motion by considering both positional consistency and orientation similarity over time. For efficient optimization, we compute this temporal rigidity within a temporal window $\mathcal{W} = [t_1, t_2]$. We first compute an orientation similarity score $s^{(i,j)}$ for each neighboring node pair $(v^{(i)}, v^{(j)})$ at the initial time t_1 of the temporal window as:

$$s^{(i,j)} = (\text{tr}((\mathbf{O}_{t_1}^{(i)})^\top \mathbf{O}_{t_1}^{(j)}) - 1)/2, \tag{3}$$

where $\text{tr}(\cdot)$ denotes the trace operator. We then compute the orientation similarity weight $w^{(i,j)}$ by applying a softmax to $s^{(i,j)}$ over the neighborhood $\mathcal{N}(i)$ of the structural proxy node $v^{(i)}$ as:

$$w^{(i,j)} = \exp(s^{(i,j)}) / \textstyle\sum_{j' \in \mathcal{N}(i)} \exp(s^{(i,j')}). \tag{4}$$

Using $w^{(i,j)}$, $\mathcal{L}_{\text{OTR}}$ is computed within $\mathcal{W}$ as:

$$\begin{aligned} \mathcal{L}_{\text{OTR}} = \sum_{t=t_1}^{t_2-\Delta} \sum_{i=1}^{M} \sum_{j \in \mathcal{N}(i)} w^{(i,j)} \Big(\Big\| \mathbf{l}_t^{(i,j)} - \mathbf{l}_{t+\Delta}^{(i,j)} \Big\|_2 + \Big| d_t^{(i,j)} - d_{t+\Delta}^{(i,j)} \Big| \Big), \\ \text{where} \quad \mathbf{l}_t^{(i,j)} = (\mathbf{O}_t^{(j)})^\top (\mathbf{q}_t^{(i)} - \mathbf{q}_t^{(j)}), \quad d_t^{(i,j)} = \|\mathbf{l}_t^{(i,j)}\|_2. \end{aligned} \tag{5}$$

In Eq. 5, $\mathbf{l}_t^{(i,j)}$ denotes the relative offset from $v^{(j)}$ to $v^{(i)}$, expressed in the local coordinate system of node $v^{(j)}$ and Δ is a temporal offset over frames. Here, positional consistency refers to preserving the local relative offset $\mathbf{l}_t^{(i,j)}$ and distance $d_t^{(i,j)}$ between neighboring nodes across time, while orientation similarity is used to compute $w^{(i,j)}$. The overall objective for the SPG foundation stage is defined as:

$$\mathcal{L}_{\text{SPG}} = \lambda_{\text{OTR}} \mathcal{L}_{\text{OTR}} + \lambda_{\text{vel}} \mathcal{L}_{\text{vel}} + \lambda_{\text{acc}} \mathcal{L}_{\text{acc}}, \tag{6}$$

where $\mathcal{L}_{\text{vel}}$ and $\mathcal{L}_{\text{acc}}$ denote the velocity and acceleration regularizations for the structural proxy nodes $\mathcal{V}$, respectively. Details are provided in Appendix B.2. After optimizing $\mathcal{L}_{\text{SPG}}$, we denote the refined SPG by $\tilde{\mathcal{K}}$ and its resulting node trajectories by $\tilde{\mathcal{Q}}^{(i)} = \{\tilde{\mathbf{q}}_t^{(i)}\}_{t=1}^{N_f}$.

3.3 Stage 2: SIS Optimization

SIS Representation. Building on the refined SPG $\tilde{\mathcal{K}}$, we propose the SIS representation for the motion of each dynamic 3D Gaussian, combining structural motion coherence with fine-grained deformation modeling. The motion of $G_{\text{dy}}^{(n)}$ is parameterized by the pair $(c^{(n)}, \mathbf{P}^{(n)})$, where $c^{(n)} \in \{1, \ldots, M\}$ denotes the node index of the associated structural proxy node and $\mathbf{P}^{(n)} = \{\mathbf{p}_k^{(n)}\}_{k=1}^{N_c}$ denotes a learnable control point set of a cubic Hermite spline representing the trajectory of $G_{\text{dy}}^{(n)}$. Specifically, $c^{(n)} = i$ indicates that $G_{\text{dy}}^{(n)}$ is assigned to the structural proxy node $v^{(i)}$. At initialization, we set $c^{(n)} = n$, yielding a one-to-one correspondence between dynamic Gaussians and structural proxy nodes. With this one-to-one correspondence, we initialize the control points $\mathbf{P}^{(n)}$ by fitting them to the corresponding refined structural proxy node trajectory $\tilde{\mathcal{Q}}^{(n)}$ as:

$$\mathbf{P}^{(n)} = \arg\min_{\mathbf{P}} \sum_{t=1}^{N_f} \Big\| \tilde{\mathbf{q}}_t^{(n)} - S(t, \mathbf{P}) \Big\|_2^2, \tag{7}$$

where $S(t, \mathbf{P})$ denotes the position of the cubic Hermite spline at time t which is defined as:

$$S(t, \mathbf{P}) = \begin{bmatrix} t_r^3 & t_r^2 & t_r & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{m}_k \\ \mathbf{m}_{k+1} \end{bmatrix}, \tag{8}$$

where $t_s = (t-1)(N_c-1)/(N_f-1)$, $k = \min(\lfloor t_s \rfloor + 1, N_c - 1)$, $t_r = t_s - (k-1)$, and $\mathbf{m}_k \approx (\mathbf{p}_{k+1} - \mathbf{p}_{k-1})/2$ for $2 \le k \le N_c - 1$. At the endpoints, we use one-sided differences: $\mathbf{m}_1 = \mathbf{p}_2 - \mathbf{p}_1$ and $\mathbf{m}_{N_c} = \mathbf{p}_{N_c} - \mathbf{p}_{N_c-1}$. The position of $G_{\text{dy}}^{(n)}$ at time t is then defined as:

$$\boldsymbol{\mu}^{(n)}(t) = S(t, \mathbf{P}^{(n)}). \tag{9}$$

During Gaussian densification, a newly split or cloned dynamic Gaussian $G_{\text{dy}}^{(n')}$ inherits the structural proxy node assignment of its source Gaussian $G_{\text{dy}}^{(n)}$ as $c^{(n')} = c^{(n)}$. As a result, multiple dynamic Gaussians may share a specific structural proxy node, enabling newly generated Gaussians to inherit the structural guidance of the refined SPG $\tilde{\mathcal{K}}$. We thus define the Gaussian set associated with $v^{(i)}$ as:

$$\mathcal{G}_{\text{dy}}(v^{(i)}) = \{G_{\text{dy}}^{(n)} \mid c^{(n)} = i\}. \tag{10}$$

Notably, the association between each $G_{\text{dy}}^{(n)}$ and its assigned node $v^{(i)}$ is used only in training to enforce structural coherence in SIS learning. At inference time, Gaussian positions are computed solely by evaluating the learned spline functions, enabling fast rendering.

SIS Optimization. After initializing the spline control points via Eq. 7, we further optimize them along with the other attributes of each dynamic Gaussian $G_{\text{dy}}^{(n)}$. To enhance the structural consistency of the SIS representation, we constrain the spline trajectories of dynamic Gaussians $\mathcal{G}_{\text{dy}}$ using two types of neighborhoods: spatial and structural. For each dynamic Gaussian $G_{\text{dy}}^{(n)}$, we define its *spatial neighborhood* $\mathcal{N}_{\text{sp}}(n)$ using the K_{sp} nearest dynamic Gaussians measured from the first spline control point $\mathbf{p}_1^{(n)}$ as:

$$\mathcal{N}_{\text{sp}}(n) = \{m \mid m \neq n,\ \text{rank}_n(\|\mathbf{p}_1^{(n)} - \mathbf{p}_1^{(m)}\|_2) \leq K_{\text{sp}}\}, \tag{11}$$

where $\text{rank}_n(\|\mathbf{p}_1^{(n)} - \mathbf{p}_1^{(m)}\|_2)$ is defined analogously to $\text{rank}_i(\mathcal{D}_{ij})$ in Eq. (2). We also define the *structural neighborhood* $\mathcal{N}_{\text{st}}(n)$ of $G_{\text{dy}}^{(n)}$ based on the connectivity of its associated structural proxy node $v^{(c^{(n)})}$ with respect to the neighboring proxy nodes $\mathcal{N}(c^{(n)})$ in Eq. (2) as:

$$\mathcal{N}_{\text{st}}(n) = \bigcup_{j \in \mathcal{N}(c^{(n)})} \{m \mid G_{\text{dy}}^{(m)} \sim \mathcal{U}(\mathcal{G}_{\text{dy}}(v^{(j)}))\}, \tag{12}$$

where $\mathcal{U}(\mathcal{G}_{\text{dy}}(v^{(j)}))$ denotes uniform sampling over the dynamic Gaussian set $\mathcal{G}_{\text{dy}}(v^{(j)})$ associated with $v^{(j)}$. We then define a spline temporal rigidity (STR) loss $\mathcal{L}_{\text{STR}}$ over $\mathcal{N}_{\text{sp}}(n)$ and $\mathcal{N}_{\text{st}}(n)$ as:

$$\mathcal{L}_{\text{STR}} = \underbrace{\alpha \cdot \sum_{k=1}^{N_c-\tau} \sum_{n=1}^{N_g} \sum_{m \in \mathcal{N}_{\text{sp}}(n)} \Psi_k^{(n,m)}}_{\text{Spatial Spline Rigidity } (\mathcal{L}_{\text{STR}}^{\text{sp}})} + \underbrace{\beta \cdot \sum_{k=1}^{N_c-\tau} \sum_{n=1}^{N_g} \sum_{m \in \mathcal{N}_{\text{st}}(n)} \Psi_k^{(n,m)}}_{\text{Structural Spline Rigidity } (\mathcal{L}_{\text{STR}}^{\text{st}})}, \tag{13}$$

where α and β denote balancing coefficients for the spatial and structural spline rigidity terms, $\mathcal{L}_{\text{STR}}^{\text{sp}}$ and $\mathcal{L}_{\text{STR}}^{\text{st}}$, respectively. The spline rigidity $\Psi_k^{(n,m)}$ is defined as:

$$\Psi_k^{(n,m)} = \|\mathbf{r}_k^{(n,m)} - \mathbf{r}_{k+\tau}^{(n,m)}\|_1 + |\rho_k^{(n,m)} - \rho_{k+\tau}^{(n,m)}|, \tag{14}$$

where $\mathbf{r}_k^{(n,m)} = \mathbf{p}_k^{(m)} - \mathbf{p}_k^{(n)}$, $\rho_k^{(n,m)} = \|\mathbf{r}_k^{(n,m)}\|_2$ and τ denotes an index offset between control points within the same spline control point set. The two spline rigidity terms, $\mathcal{L}_{\text{STR}}^{\text{sp}}$ and $\mathcal{L}_{\text{STR}}^{\text{st}}$, provide complementary constraints for SIS optimization. $\mathcal{L}_{\text{STR}}^{\text{sp}}$ preserves local geometric consistency among spatially nearby dynamic Gaussians, while $\mathcal{L}_{\text{STR}}^{\text{st}}$ promotes coherent motion among potentially distant but structurally related Gaussians associated with neighboring SPG nodes. The resulting $\mathcal{L}_{\text{STR}}$ is combined with the RGB and depth losses, $\mathcal{L}_{\text{rgb}}$ and $\mathcal{L}_{\text{d}}$, to optimize the SIS representation for both structural coherence and reconstruction fidelity. Here, $\mathcal{L}_{\text{rgb}}$ and $\mathcal{L}_{\text{d}}$ are L1 losses on the rendered frame and rendered depth, respectively. The overall objective for SIS optimization is defined as:

$$\mathcal{L}_{\text{SIS}} = \lambda_{\text{STR}}\mathcal{L}_{\text{STR}} + \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_{\text{d}}\mathcal{L}_{\text{d}}. \tag{15}$$

4 Experiments

4.1 Experimental Setup

Implementation Details. Our framework is built on the widely adopted open-source 3D Gaussian Splatting (3DGS) codebase [15]. We obtain 2D point tracks using CoTracker3 [13]. For depth,

Table 1: Quantitative comparisons of novel view synthesis on the iPhone [9] and NVIDIA [50] datasets. Best, second-best, and third-best results are highlighted in red, orange, and yellow, respectively. Rendering speed (FPS) and training time (Tr. h) are measured using a single NVIDIA RTX 3090 Ti GPU. Training times are measured on the NVIDIA dataset [50]. All methods use the same depth and tracking priors where applicable.

Method	iPhone [9]			NVIDIA [50]			FPS↑		Tr. (h)↓
	mPSNR↑	mSSIM↑	mLPIPS↓	PSNR↑	SSIM↑	LPIPS↓	iPhone	NVIDIA	
T-NeRF (NeurIPS'22) [9]	16.96	0.577	0.379	18.33	0.436	0.511	0.43	0.28	> 24
HyperNeRF (ACM-TG'21) [35]	16.81	0.569	0.332	17.60	0.377	0.367	0.27	0.33	> 24
NSFF (CVPR'21) [24]	15.46	0.551	0.396	24.33	0.751	0.199	0.30	0.38	> 24
4DGS (CVPR'24) [45]	13.87	0.459	0.407	22.89	0.731	0.237	83	95	**0.25**
RoDynRF (CVPR'23) [30]	17.10	0.534	0.517	25.89	0.854	0.067	0.62	0.45	> 24
D-NPC (Eurographics'25) [11]	16.41	0.582	0.319	25.64	0.845	0.109	65	70	0.27
MarbleGS (SIGGRAPH'24) [38]	16.08	0.568	0.433	24.56	0.667	0.110	16	10	13
HiMoR (CVPR'25) [25]	16.02	0.558	0.325	24.89	0.694	0.118	157	145	1
MoDec-GS (CVPR'25) [19]	14.65	0.320	0.461	23.94	0.628	0.132	15	10	1
SoM (ICCV'25) [43]	17.13	0.674	0.279	24.58	0.651	0.124	127	255	1
SplineGS (CVPR'25) [34]	18.70	0.663	0.325	27.12	0.872	0.052	**288**	**400**	2
MoSca (CVPR'25) [21]	19.54	0.719	0.247	26.48	0.848	0.072	45	40	0.8
SE3BSplineGS (CVPR'26) [54]	19.36	0.705	0.290	26.15	0.845	0.085	64	56	2
WorldTree (ICLR'26) [44]	19.75	0.728	0.240	26.28	0.845	0.065	27	47	1
WebSpline (Ours)	**20.11**	**0.738**	**0.222**	**27.73**	**0.881**	**0.050**	278	390	0.5

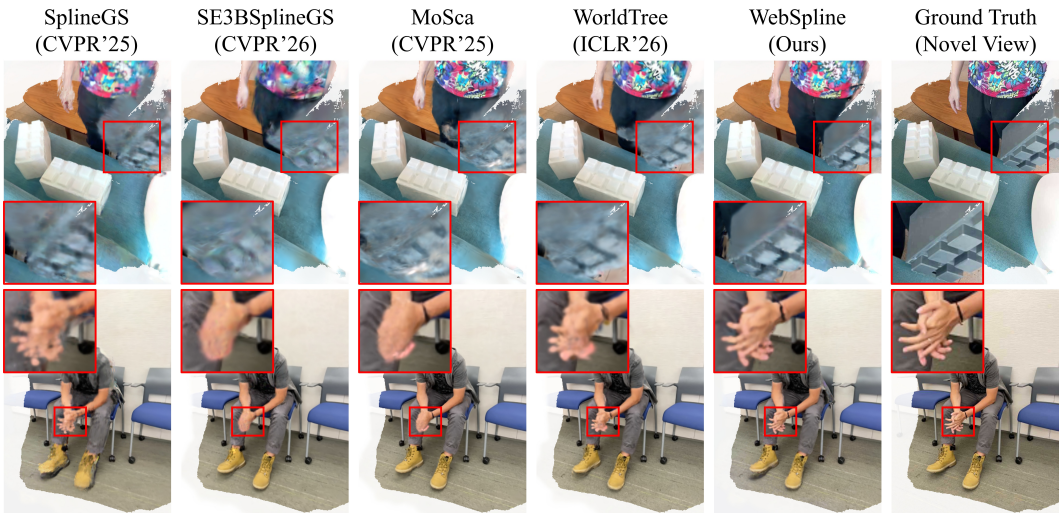


Figure 3: **Visual comparisons for novel view synthesis on the iPhone dataset [9].**

we use the pre-trained UniDepth model [36] on the NVIDIA dataset [50] and sensor depth on the iPhone dataset [9]. For all compared methods that require depth or tracking priors, we use these same priors, respectively. WebSpline is trained for 2K and 20K iterations in the SPG foundation and SIS optimization stages, respectively. In the SPG foundation stage, we linearly decay the current window length $|\mathcal{W}|$ from N_f to $0.1 \times N_f$ to speed up optimization. The starting frame t_1 is sampled from $[1, N_f - |\mathcal{W}| + 1]$ via a Beta distribution to provide balanced supervision across the entire sequence. We set $K = 16$ in Eq. 2 and $K_{\text{sp}} = 8$ in Eq. 11. We set the number of control points to $N_c = \lfloor N_f/4 \rfloor$ for the iPhone dataset [9] and $N_c = N_f$ for the NVIDIA dataset [50]. We set the temporal offset Δ in Eq. 5 and the index offset τ between control points in Eq. 13 to 1. We set α and β values in Eq. 13 to 0.1. For $\mathcal{L}_{\text{SPG}}$ and $\mathcal{L}_{\text{SIS}}$, we set all λ values in Eqs. 6 and 15 to 1, respectively.

Datasets. We evaluate WebSpline on two widely used monocular video benchmarks: the iPhone [9] and NVIDIA [50] datasets. The iPhone dataset consists of 7 casually captured scenes featuring complex camera and object motions. The NVIDIA dataset includes 7 scenes with diverse motion patterns, originally recorded with a 12-camera rig. Following the protocol of RoDynRF [30], we simulate a monocular video sequence by selecting one camera view per time step.

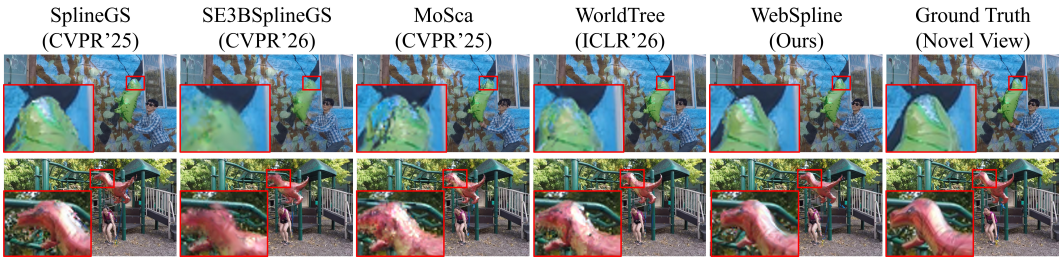


Figure 4: **Visual comparisons for novel view synthesis on the NVIDIA dataset [50].**

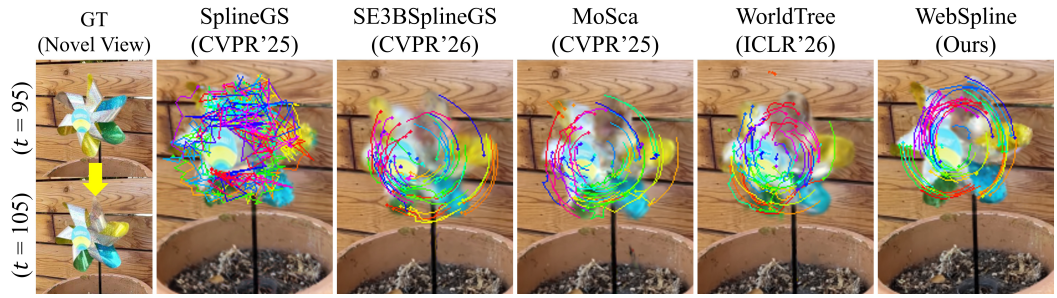


Figure 5: **Visual comparisons of dynamic Gaussian trajectories on the iPhone dataset [9].**

Metrics. For the iPhone dataset [9], we follow DyCheck [9] and report co-visibility-masked mPSNR, mSSIM, and mLPIPS to exclude regions unobserved during training. For the NVIDIA dataset [50], we report PSNR, SSIM, and LPIPS. We also report FPS to assess rendering efficiency.

4.2 Comparison with State-of-the-Art Methods

Novel View Synthesis. Table 1 presents quantitative comparisons for dynamic novel view synthesis between our WebSpline and existing methods [9, 35, 24, 45, 30, 11, 38, 25, 19, 43, 34, 21, 54, 44] on the iPhone [9] and NVIDIA [50] datasets. Since SE3BSplineGS [54] only reported performance for a subset of the iPhone dataset [9], we evaluated it across all seven scenes using its official code under the same experimental setting. As shown in Table 1, WebSpline consistently outperforms all the SOTA methods across all rendering-quality metrics on both datasets. The scaffold-based methods [21, 54, 44] show lower perceptual quality than our WebSpline on both datasets, as reflected by mLPIPS and LPIPS, indicating the limitation of their reliance on motion blending. SplineGS [34] achieves the second-best results on the NVIDIA dataset [50], but its performance degrades on the longer and more challenging iPhone [9] sequences, implying the instability of its unconstrained motion representation. Notably, WebSpline achieves 278 FPS on the iPhone dataset, rendering more than **10**$\times$ **faster** than WorldTree [44] that is the second-best method in rendering quality but achieves only 27 FPS in rendering speed. While WorldTree [44] computes each Gaussian's motion by blending neighboring scaffold-node motions using dual quaternion blending (DQB) [14], our SIS representation directly evaluates spline-based Gaussian motion, avoiding this blending overhead. This efficiency stems from direct spline evaluation, as in SplineGS; our contribution is to preserve this efficiency while improving structural coherence and rendering quality. Figs. 3 and 4 show visual comparisons between WebSpline and SOTA methods [34, 54, 21, 44] on the iPhone [9] and NVIDIA [50] datasets, respectively. As shown, WebSpline produces novel view renderings with clearer boundaries and sharper details of dynamic objects than the SOTA methods [34, 54, 21, 44]. More results are provided in Appendix C.

Gaussian Trajectory Visualization. To further evaluate WebSpline's motion modeling, Fig. 5 visualizes dynamic Gaussian trajectories projected onto the 2D image plane and compares them with SOTA methods [34, 54, 21, 44]. SplineGS [34] fails to accurately model the trajectories of moving dynamic regions, resulting in noisy Gaussian motions. The scaffold-based methods [21, 54, 44] tend to produce jittering trajectories, indicating that motion blending across neighboring scaffold nodes may lead to temporally unstable Gaussian motion. In contrast, our WebSpline produces clear trajectories, showing both structurally coherent and temporally smooth Gaussian motion representation.

Table 2: **Ablation studies.** We ablate the key framework design components and loss functions of WebSpline on the iPhone dataset [9] under the same setting as Table 1.

(a) Framework Design

Method	mPSNR↑	mSSIM↑	mLPIPS↓
w/o SPG	19.68	0.728	0.236
w/o SIS	19.62	0.724	0.239
w/o 2-stage	19.75	0.730	0.230
Ours	**20.11**	**0.738**	**0.222**

(b) Loss Functions

Method	mPSNR↑	mSSIM↑	mLPIPS↓
w/o $\mathcal{L}_{\text{OTR}}$	19.79	0.728	0.232
w/o $\mathcal{L}_{\text{STR}}$	19.76	0.729	0.245
w/o $\mathcal{L}^{\text{sp}}_{\text{STR}}$	20.04	0.738	0.226
w/o $\mathcal{L}^{\text{st}}_{\text{STR}}$	20.01	0.737	0.227
Ours	**20.11**	**0.738**	**0.222**



Figure 6: **Visual results of ablation study on the iPhone dataset [9].**

4.3 Ablation Study

To validate the effectiveness of WebSpline's framework design and loss functions, Table 2 and Fig. 6 present ablation results on the iPhone dataset [9].

Framework Design. Table 2-(a) reports ablation results for the framework design of WebSpline, including the SPG, SIS, and two-stage optimization strategy. For 'w/o SPG', we initialize splines directly from lifted 3D point tracks [13] via least-squares fitting and remove the structural spline rigidity term $\mathcal{L}^{\text{st}}_{\text{STR}}$ while retaining the spatial spline rigidity term $\mathcal{L}^{\text{sp}}_{\text{STR}}$ in Eq. 13. The resulting performance drop highlights the importance of the SPG for coherent Gaussian motion. For 'w/o SIS', we remove the SIS representation and directly use the structural proxy node trajectories obtained after the SPG foundation stage to represent dynamic Gaussian motion. The resulting performance drop shows that SIS plays a crucial role in providing sufficient motion expressiveness for fine-grained dynamic Gaussian trajectories. For 'w/o 2-stage', we remove the SPG foundation stage and use the SPG initialized from 2D point tracks without pre-optimization. The SIS is initialized from this unoptimized SPG, and the SPG and SIS are then jointly optimized in a single stage. The resulting degraded performance indicates that a pre-optimized SPG provides more reliable structural guidance for stable SIS learning, highlighting the rationale behind our two-stage optimization strategy. These trends align with the qualitative results in Fig. 6, where all three ablated variants exhibit noisier and less coherent motion than the full WebSpline.

Loss Functions. Table 2-(b) shows the effectiveness of $\mathcal{L}_{\text{OTR}}$ and $\mathcal{L}_{\text{STR}}$, used in the SPG foundation and SIS optimization stages, respectively. Without $\mathcal{L}_{\text{OTR}}$, the SPG fails to maintain structurally coherent node trajectories over time, weakening the structural guidance for SIS learning. Without $\mathcal{L}_{\text{STR}}$, neighboring Gaussians lose temporal rigidity, resulting in less coherent spline trajectories. This indicates that explicit rigidity constraints are necessary beyond mere SPG initialization. Fig. 6 further shows that removing either $\mathcal{L}_{\text{OTR}}$ or $\mathcal{L}_{\text{STR}}$ leads to noisier renderings, consistent with the quantitative degradation. Furthermore, as shown in Table 2-(b), we ablate the two spline rigidity terms $\mathcal{L}^{\text{sp}}_{\text{STR}}$ and $\mathcal{L}^{\text{st}}_{\text{STR}}$ in $\mathcal{L}_{\text{STR}}$ to analyze their contributions. Ablating either $\mathcal{L}^{\text{sp}}_{\text{STR}}$ or $\mathcal{L}^{\text{st}}_{\text{STR}}$ degrades performance, while using both terms achieves the best results. This demonstrates their complementary roles in achieving local geometric consistency and SPG-induced structural coherence, respectively.

Analysis of Local Structural Distortion (LSD). To further analyze the structural consistency of WebSpline, Fig. 7 visualizes the local structural distortion (LSD) distribution across dynamic Gaussians. LSD measures the temporal variation of distances between neighboring Gaussians, allowing us to assess how the SPG and $\mathcal{L}_{\text{STR}}$ contribute to local structural consistency in the learned SIS trajectories. Specifically, for each dynamic Gaussian $G^{(n)}_{\text{dy}}$, we define $\mathcal{N}_{\text{LSD}}(n)$ as the set of its $K = 8$ nearest neighbors based on Euclidean distances between Gaussian centers in the first frame. For each neighbor $m \in \mathcal{N}_{\text{LSD}}(n)$, the distance at time t is $d_{n,m}(t) = \|\boldsymbol{\mu}^{(n)}(t) - \boldsymbol{\mu}^{(m)}(t)\|_2$, where Gaussian centers are computed from the learned SIS using Eq. 9. We define LSD as the temporal

variance of these distances averaged over the neighbors:

$$\mathrm{LSD}^{(n)} = \frac{1}{K} \sum_{m \in \mathcal{N}_{\mathrm{LSD}}(n)} \frac{1}{N_f - 1} \sum_{t=1}^{N_f} \left(d_{n,m}(t) - \bar{d}_{n,m} \right)^2, \quad \bar{d}_{n,m} = \frac{1}{N_f} \sum_{t=1}^{N_f} d_{n,m}(t). \tag{16}$$

Lower LSD indicates more consistent distances to neighboring Gaussians over time, while higher LSD indicates greater local structural distortion. In Fig. 7, we compare WebSpline against two variants: (i) 'w/o $\mathcal{L}_{\mathrm{STR}}$' and (ii) 'w/o $\mathcal{L}_{\mathrm{STR}}$ & SPG'. The vertical axis ('% of Gaussians') indicates the percentage of dynamic Gaussians in each LSD bin. We also report the median ($\mathcal{M}$) and standard deviation (σ) of each distribution. For 'w/o $\mathcal{L}_{\mathrm{STR}}$ & SPG' (orange), the LSD's $\mathcal{M}$ reaches 20.927, indicating severe local structural distortion caused by unconstrained Gaussian trajectories. While 'w/o $\mathcal{L}_{\mathrm{STR}}$' (pink) lowers the LSD's $\mathcal{M}$, it still exhibits a large standard deviation ($\sigma = 4.423$), showing its insufficiency in consistently constraining Gaussian trajectories. In contrast, the LSD values of our full model (blue) are concentrated in the low-value range, confirming that WebSpline enables dynamic Gaussians to move in a structurally stable manner with respect to their neighboring Gaussians.

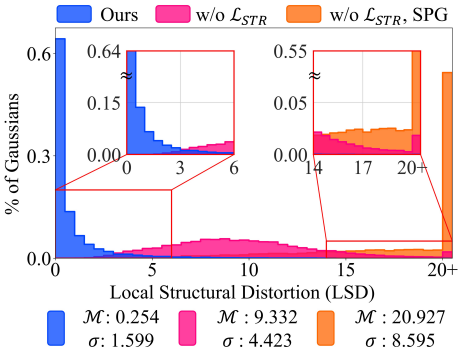


Figure 7: **Distributions of LSD Values.** We use the *paper-windmill* scene of the iPhone dataset [9]. '20+' indicates all LSD values above 20 are accumulated at the LSD value of 20.

5 Conclusion

We present WebSpline, a dynamic 3D Gaussian framework for structurally coherent and high-fidelity reconstruction from monocular videos. At its core, WebSpline explicitly couples a Structure-Informed Spline (SIS) representation with a Structural Proxy Graph (SPG), achieving a highly expressive yet robust motion model that captures complex, fine-grained Gaussian dynamics while reducing structural degradation. With two-stage optimization and SIS-only inference, WebSpline achieves **state-of-the-art rendering quality** on the iPhone and NVIDIA datasets. On the iPhone dataset, it renders **over** $10\times$ **faster** than WorldTree, the second-best method in rendering quality.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government [Ministry of Science and ICT (Information and Communications Technology)] (Project Number: RS-2022-00144444, Project Title: Deep Learning Based Visual Representational Learning and Rendering of Static and Dynamic Scenes).

References

[1] J Harold Ahlberg, Edwin Norman Nilson, and Joseph Leonard Walsh. *The Theory of Splines and Their Applications: Mathematics in Science and Engineering: A Series of Monographs and Textbooks, Vol. 38*, volume 38. Elsevier, 2016.

[2] Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. In *European Conference on Computer Vision*, pages 321–335. Springer, 2024.

[3] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, 2021. URL `https://arxiv.org/abs/2103.13415`.

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields, 2022. URL `https://arxiv.org/abs/2111.12077`.

[5] Minh-Quan Viet Bui, Jongmin Park, Juan Luis Gonzalez, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Mobgs: Motion deblurring dynamic 3d gaussian splatting for blurry monocular video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 2480–2489, 2026.

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022.

[7] C De Boor. A practical guide to splines. *Springer-Verlag*, 1978.

[8] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022.

[9] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022.

[10] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. *CVPR*, 2024.

[11] Moritz Kappel, Florian Hahlbohm, Timon Scholz, Susana Castillo, Christian Theobalt, Martin Eisemann, Vladislav Golyanik, and Marcus Magnor. D-npc: Dynamic neural point clouds for non-rigid view synthesis from monocular video. *arXiv preprint arXiv:2406.10078*, 2024.

[12] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv*, 2023.

[13] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6013–6022, 2025.

[14] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46, 2007.

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. URL `https://arxiv.org/abs/2308.04079`.

[16] Leonid Keselman and Martial Hebert. Approximate differentiable rendering with algebraic surfaces. In *European Conference on Computer Vision*, pages 596–614. Springer, 2022.

[17] Leonid Keselman and Martial Hebert. Flexible techniques for differentiable rendering with 3d gaussians. *arXiv preprint arXiv:2308.14737*, 2023.

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, 2023. doi: 10.1109/ICCV51070.2023.00371.

[19] Sangwoon Kwak, Joonsoo Kim, Jun Young Jeong, Won-Sik Cheong, Jihyong Oh, and Munchurl Kim. Modec-gs: Global-to-local motion decomposition and temporal interval adjustment for compact dynamic 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[20] Junoh Lee, Chang-Yeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. In *NeurIPS*, 2024.

[21] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024.

[22] Deqi Li, Shi-Sheng Huang, Zhiyuan Lu, Xinran Duan, and Hua Huang. St-4dgs: Spatial-temporally consistent 4d gaussian splatting for efficient dynamic scene rendering. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705250. doi: 10.1145/3641519.3657520. URL `https://doi.org/10.1145/3641519.3657520`.

[23] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *CVPR*, 2024.

[24] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

[25] Yiming Liang, Tianhan Xu, and Yuta Kikuchi. Himor: Monocular deformable gaussian reconstruction with hierarchical motion representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 886–895, 2025.

[26] Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. *arXiv*, 2023.

[27] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[28] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21136–21145, 2024.

[29] Qingming Liu, Yuan Liu, Jiepeng Wang, Xianqiang Lyu, Peng Wang, Wenping Wang, and Junhui Hou. Modgs: Dynamic gaussian splatting from casually-captured monocular videos with depth priors. In *The Thirteenth International Conference on Learning Representations*, 2025.

[30] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *CVPR*, 2023.

[31] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis, 2023. URL `https://arxiv.org/abs/2308.09713`.

[32] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. URL `https://arxiv.org/abs/2003.08934`.

[33] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[34] Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez Bello, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Splinegs: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 26866–26875, June 2025.

[35] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 2021.

[36] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *CVPR*, 2024.

[37] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv*, 2018.

[38] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[39] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5459–5469, 2022.

[40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

[41] Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction. *arXiv preprint arXiv:2406.03697*, 2024.

[42] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13524–13534, 2022.

[43] Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *International Conference on Computer Vision (ICCV)*, 2025.

[44] Qisen Wang, Yifan Zhao, and Jia Li. Worldtree: Towards 4d dynamic worlds from monocular video using tree-chains. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=mVo6cyFR6C`.

[45] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering, 2024. URL `https://arxiv.org/abs/2310.08528`.

[46] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.

[47] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[48] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[49] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023.

[50] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*, 2020.

[51] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5752–5761, 2021.

[52] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19447–19456, 2024.

[53] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics (ToG)*, 43(6):1–13, 2024.

[54] Xuankai Zhang, Junjin Xiao, Shangwei Huang, Wei-shi Zheng, and Qing Zhang. Learning explicit continuous motion representation for dynamic gaussian splatting from monocular videos. *arXiv preprint arXiv:2603.25058*, 2026.

Appendix

A Additional Analysis

A.1 Analysis of SPG

The optimized SPG supports SIS learning in two ways: its refined node trajectories provide a structurally coherent initialization, and its structural neighborhoods guide subsequent SIS optimization through $\mathcal{L}_{\mathrm{STR}}^{\mathrm{st}}$. Table 3 separately evaluates these two roles on the iPhone dataset [9], while retaining the spatial regularization term $\mathcal{L}_{\mathrm{STR}}^{\mathrm{sp}}$ in all variants.

Table 3: Analysis of the two roles of the optimized SPG on the iPhone dataset [9]. "SIS init." and "Structural reg." denote SIS initialization and structural regularization, respectively. "Yes/No" indicates whether the optimized SPG is used for each role. Spatial regularization $\mathcal{L}_{\mathrm{STR}}^{\mathrm{sp}}$ is retained in all variants.

Optimized SPG usage	SIS init.	Structural reg.	mPSNR↑	mSSIM↑	mLPIPS↓
Neither role	No	No	19.68	0.728	0.236
Regularization only	No	Yes	19.74	0.730	0.230
Initialization only	Yes	No	20.01	0.737	0.227
Both roles (Ours)	Yes	Yes	**20.11**	**0.738**	**0.222**

In *Neither role*, the SPG is removed: spline control points are initialized by least-squares fitting directly to the lifted 3D point tracks, following SplineGS [34], and $\mathcal{L}_{\mathrm{STR}}^{\mathrm{st}}$ is omitted. In *Regularization only*, the SPG is constructed and optimized to define structural neighborhoods for $\mathcal{L}_{\mathrm{STR}}^{\mathrm{st}}$, but the spline control points are still fitted directly to the lifted 3D point tracks. In *Initialization only*, the spline control points are fitted to the refined SPG node trajectories, while $\mathcal{L}_{\mathrm{STR}}^{\mathrm{st}}$ is omitted. Finally, *Both roles (Ours)* uses the optimized SPG for both initialization and structural regularization. All variants use least-squares fitting for spline initialization; the difference is whether the fitting targets are the lifted 3D point tracks or the refined SPG node trajectories.

As shown in Table 3, *Initialization only* improves mPSNR from 19.68 to 20.01 dB over *Neither role*, whereas *Regularization only* improves it to 19.74 dB. These results indicate that structurally refined initialization accounts for the larger performance gain in this experiment. *Both roles (Ours)* further improves mPSNR to 20.11 dB and achieves the best mSSIM and mLPIPS, showing that structural regularization provides an additional benefit beyond initialization.

A.2 Analysis of $\mathcal{L}_{\mathrm{OTR}}$

The orientation-similarity-based temporal rigidity loss $\mathcal{L}_{\mathrm{OTR}}$ is related to the ARAP-style rigidity regularization used in the scaffold-based methods [21, 54, 44]. Both losses encourage neighboring proxy nodes to preserve local geometric relationships over time. Unlike the original ARAP formulation, which treats neighboring relations uniformly, $\mathcal{L}_{\mathrm{OTR}}$ uses the orientation similarity weight $w^{(i,j)}$ to adaptively emphasize neighboring node pairs that exhibit similar local rigid motion. This weighting reduces the influence of neighboring nodes that are spatially close but follow different motion patterns. In WebSpline, $\mathcal{L}_{\mathrm{OTR}}$ is used to optimize the SPG as structural guidance for subsequent SIS optimization, rather than to directly drive Gaussian deformation or rendering.

Table 4: Effect of orientation-similarity weighting in $\mathcal{L}_{\mathrm{OTR}}$.

Method	mPSNR↑	mSSIM↑	mLPIPS↓
w/o OTR	19.79	0.728	0.232
OTR w/o $w^{(i,j)}$	20.00	0.736	0.225
Ours	**20.11**	**0.738**	**0.222**

As shown in Table 4, using $\mathcal{L}_{\mathrm{OTR}}$ without the orientation similarity weight $w^{(i,j)}$ already improves over removing $\mathcal{L}_{\mathrm{OTR}}$, confirming the benefit of temporal rigidity for SPG refinement. Adding $w^{(i,j)}$ further improves mPSNR from 20.00 to 20.11 and reduces mLPIPS from 0.225 to 0.222, indicating that adaptive weighting provides more reliable structural guidance. Overall, $\mathcal{L}_{\mathrm{OTR}}$ serves as an effective refinement objective for the SPG, while the main contribution of WebSpline lies in the SPG-to-SIS framework that uses this refined structural foundation to learn structurally coherent yet flexible spline trajectories for dynamic Gaussians.

A.3 Analysis of τ

We conduct an ablation study on the index offset τ between control points in the spline temporal rigidity loss $\mathcal{L}_{\text{STR}}$ defined in Eq. 13. Specifically, the spline rigidity term $\Psi_k^{(n,m)}$ compares relative position vectors and distances between neighboring Gaussians at control point indices k and $k+\tau$. As shown in Table 5, the performance gradually decreases as τ increases. A larger τ enforces rigidity over a longer temporal span of control points, introducing long-range temporal constraints that can reduce the flexibility of SIS in modeling fine-grained Gaussian motion. Therefore, we set $\tau = 1$ as the default setting.

Table 5: Effect of the index offset τ between control points in $\mathcal{L}_{\text{STR}}$.

Method	mPSNR↑	mSSIM↑	mLPIPS↓
$\tau = 10$	19.97	0.736	0.225
$\tau = 7$	19.98	0.736	0.225
$\tau = 4$	20.02	0.737	0.224
$\tau = 2$	20.06	0.738	0.222
Ours ($\tau = 1$)	**20.11**	**0.738**	**0.222**

B Additional Method Details

This section provides additional details on the auxiliary components used in the SPG foundation stage. These components stabilize the SPG $\mathcal{K}$ before learning the SIS representation.

B.1 Auxiliary Orientation Initialization

To initialize the auxiliary orientation sequence $\mathcal{O}^{(i)} = \{\mathbf{O}_t^{(i)}\}_{t=1}^{N_f}$, we employ a local Procrustes alignment strategy consistent with recent scaffold-based methods [21, 54, 44]. Let $\mathcal{N}(i)$ denote the valid neighbors of node i, and let $\mathbf{r}_t^{(i,j)} = \mathbf{q}_t^{(j)} - \mathbf{q}_t^{(i)}$ be the relative position vector from node i to node j. We center these vectors by subtracting their mean over $\mathcal{N}(i)$, yielding $\tilde{\mathbf{r}}_t^{(i,j)} = \mathbf{r}_t^{(i,j)} - |\mathcal{N}(i)|^{-1} \sum_{\ell \in \mathcal{N}(i)} \mathbf{r}_t^{(i,\ell)}$. Starting with $\mathbf{O}_1^{(i)} = \mathbf{I}$, we estimate the relative orientation update from frame t to frame $t+1$, for $t = 1, \ldots, N_f - 1$, by aligning the centered local configurations:

$$\Delta\mathbf{O}_t^{(i)} = \arg\min_{\mathbf{R} \in SO(3)} \sum_{j \in \mathcal{N}(i)} \left\| \mathbf{R}\tilde{\mathbf{r}}_t^{(i,j)} - \tilde{\mathbf{r}}_{t+1}^{(i,j)} \right\|_2^2. \tag{17}$$

We solve this alignment problem using the standard SVD-based closed-form solution, with a determinant correction to ensure a proper rotation, and recursively update the orientation as

$$\mathbf{O}_{t+1}^{(i)} = \Delta\mathbf{O}_t^{(i)} \mathbf{O}_t^{(i)}. \tag{18}$$

These auxiliary orientations are used to compute the orientation similarity weights $w^{(i,j)}$ and relative position vectors in local coordinates within $\mathcal{L}_{\text{OTR}}$, as well as the orientation regularization terms described in Sec. B.2.

B.2 Velocity and Acceleration Regularization

We apply temporal smoothness regularization terms to the SPG node positions and auxiliary orientations [21, 54, 44]. For each structural proxy node $v^{(i)}$, we regularize both the position sequence $\{\mathbf{q}_t^{(i)}\}_{t=1}^{N_f}$ and the auxiliary orientation sequence $\{\mathbf{O}_t^{(i)}\}_{t=1}^{N_f}$.

For the position sequence, the velocity and acceleration terms are defined as

$$\mathcal{L}_{\text{vel}}^{\text{pos}} = \sum_{t=1}^{N_f-1} \sum_{i=1}^{M} \left\| \mathbf{q}_{t+1}^{(i)} - \mathbf{q}_t^{(i)} \right\|_2, \qquad \mathcal{L}_{\text{acc}}^{\text{pos}} = \sum_{t=1}^{N_f-2} \sum_{i=1}^{M} \left\| \mathbf{q}_{t+2}^{(i)} - 2\mathbf{q}_{t+1}^{(i)} + \mathbf{q}_t^{(i)} \right\|_2. \tag{19}$$

For the auxiliary orientation sequence, we compute the relative orientation update from frame t to frame $t+1$ and its rotation angle:

$$\Delta\mathbf{O}_t^{(i)} = \mathbf{O}_{t+1}^{(i)} \left(\mathbf{O}_t^{(i)} \right)^\top, \qquad \theta_t^{(i)} = \angle \left(\Delta\mathbf{O}_t^{(i)} \right), \qquad t = 1, \ldots, N_f - 1, \tag{20}$$

where $\angle(\cdot)$ denotes the rotation angle of an $SO(3)$ matrix. The orientation velocity and acceleration terms are then defined as

$$\mathcal{L}_{\text{vel}}^{\text{ori}} = \sum_{t=1}^{N_f-1} \sum_{i=1}^{M} \theta_t^{(i)}, \qquad \mathcal{L}_{\text{acc}}^{\text{ori}} = \sum_{t=1}^{N_f-2} \sum_{i=1}^{M} \left| \theta_{t+1}^{(i)} - \theta_t^{(i)} \right|. \tag{21}$$

The orientation velocity term penalizes the magnitude of inter-frame rotation, while the orientation acceleration term penalizes changes in this magnitude; the latter does not directly penalize changes in the rotation axis. Finally, we combine the position and orientation terms as

$$\mathcal{L}_{\text{vel}} = \alpha_{\text{pos}}\mathcal{L}_{\text{vel}}^{\text{pos}} + \alpha_{\text{ori}}\mathcal{L}_{\text{vel}}^{\text{ori}}, \qquad \mathcal{L}_{\text{acc}} = \alpha_{\text{pos}}\mathcal{L}_{\text{acc}}^{\text{pos}} + \alpha_{\text{ori}}\mathcal{L}_{\text{acc}}^{\text{ori}}. \tag{22}$$

We set $\alpha_{\text{pos}} = 0.3$ and $\alpha_{\text{ori}} = 0.1$ for both regularization terms. These internal weights are distinct from the overall coefficients λ_{vel} and λ_{acc} in Eq. 6, which are set to 1. Together, these terms encourage smooth temporal evolution of the SPG during the SPG foundation stage, complementing $\mathcal{L}_{\text{OTR}}$ in Eq. 6.

C Additional Qualitative Results

Figs. 9, 10, 11, 12, and 13 present additional qualitative comparisons for novel view synthesis on the iPhone dataset [9].

D Limitations

Although WebSpline achieves high-quality dynamic scene reconstruction with fast rendering, it requires scene-specific optimization for each video sequence. This limits its applicability to immediate reconstruction scenarios. Extending WebSpline to a feed-forward framework that directly predicts the SIS representation for a given video sequence is an important direction for future work.



Figure 8: **Failure case on the *kite-surf* scene.** The input view and novel-view rendering show the difficulty of reconstructing highly non-rigid water splashes and droplets. The highlighted region shows blurred splash details.

Failure Case. The structural rigidity prior can be unreliable when local relationships change substantially over time. As shown in Fig. 8, a representative case is the water splashes and droplets in the *kite-surf* scene of the DAVIS dataset [37]. Their highly non-rigid motion violates the local structural consistency assumption, weakening optimized-SPG-based regularization and degrading the learned SIS representation.

E Broader Impact

WebSpline may support applications in AR/VR, telepresence, robotics, and digital content creation by enabling efficient dynamic 3D reconstruction from monocular videos. It can help reduce the need for specialized multi-camera capture systems and make dynamic 3D content creation more accessible. However, potential risks include privacy concerns when capturing people or private spaces, as well as the misuse of synthesized novel views. As such accessible rendering technologies advance, it becomes increasingly critical to establish robust consent protocols and clear disclosure mechanisms for synthesized visual content.

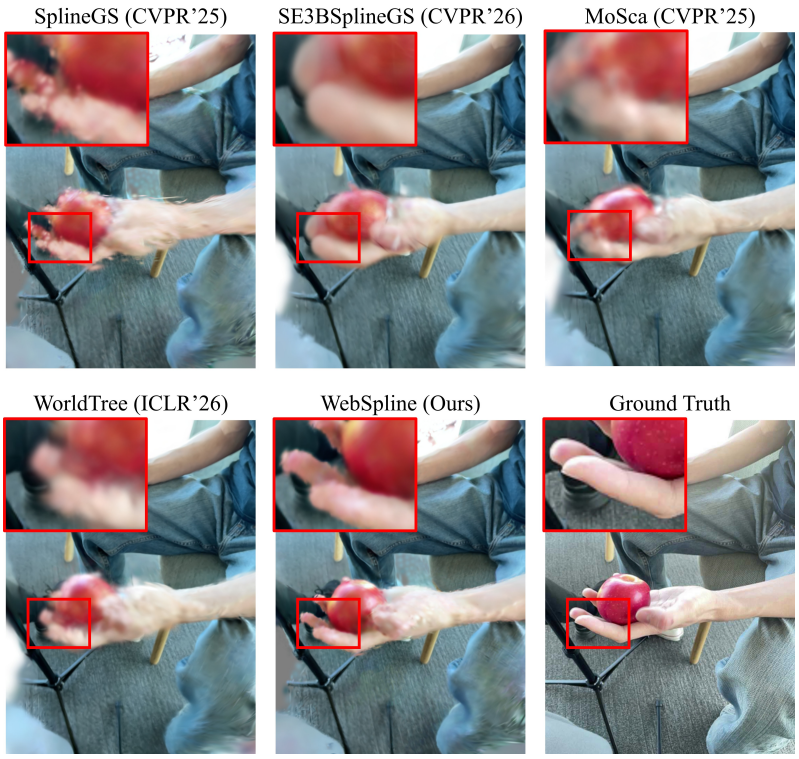


Figure 9: **Visual comparisons for novel view synthesis on the *apple* scene from the iPhone dataset [9].**

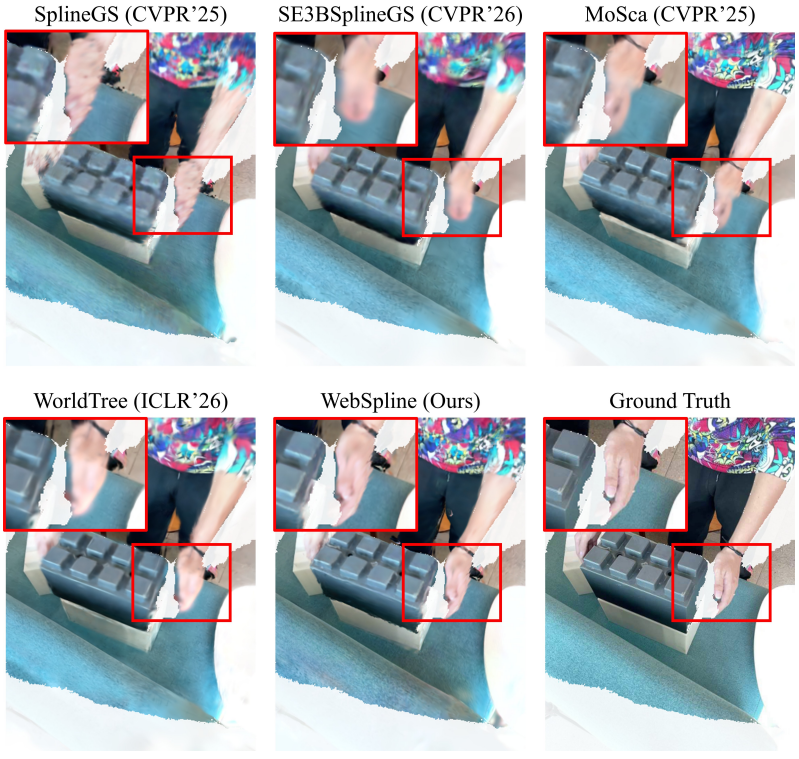


Figure 10: **Visual comparisons for novel view synthesis on the *block* scene from the iPhone dataset [9].**

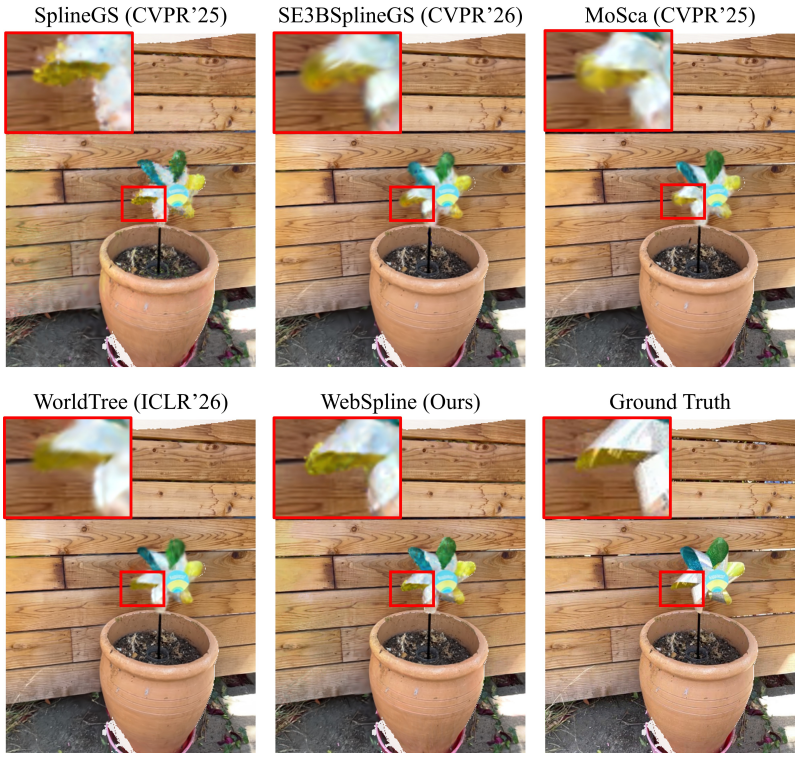


Figure 11: **Visual comparisons for novel view synthesis on the *paper-windmill* scene from the iPhone dataset [9].**

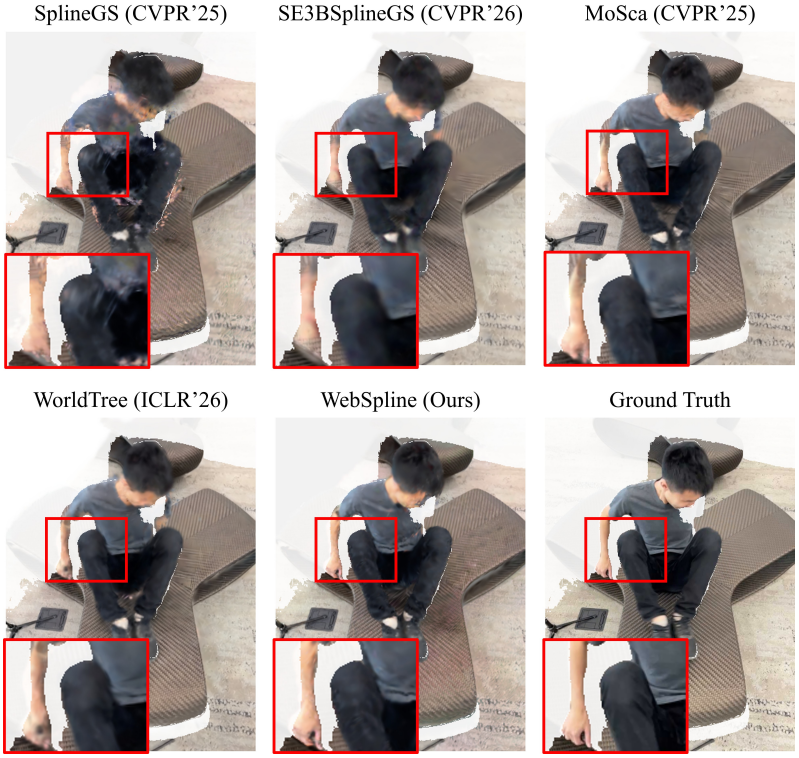


Figure 12: **Visual comparisons for novel view synthesis on the *spin* scene from the iPhone dataset [9].**



Figure 13: **Visual comparisons for novel view synthesis on the *wheel* scene from the iPhone dataset [9].**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly reflect the paper's contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a Limitations and Future Work section in the Appendix, where we discuss that WebSpline requires per-scene optimization and identify feed-forward extension as a future direction.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper does not include theoretical assumptions and proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the method details in the main paper and include additional implementation details in the Appendix, including the codebase, 2D tracking and depth estimation tools, training iterations, neighborhood sizes, number of control points, offset values, and loss weights used in our experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We do not provide open access to the code at submission time. However, we provide detailed method descriptions and implementation details in the paper and Appendix to support reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the experimental setting in the main paper and provide additional implementation details and hyperparameters in the Appendix, including training iterations, neighborhood sizes, control-point settings, offset values, and loss weights.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report quantitative results averaged over standard benchmark scenes, but do not include error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the GPU used for our experiments, along with training time and rendering speed for the evaluated methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We properly follow the code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive and negative societal impacts of this work in the Appendix.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release high-risk models or datasets that require special safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite and credit the existing codebase, pretrained models, and benchmark datasets used in this work, including 3DGS, CoTracker3, UniDepth, and the iPhone and NVIDIA datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new datasets, code, or model assets at submission time.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development does not involve LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.