Surface-Aware Feed-Forward Quadratic Gaussian for Frame Interpolation with Large Motion

Zaoming Yan [1,2] Yaomin Huang[1,2] Pengcheng Lei [1,2] Qizhou Chen [1]
Guixu Zhang[1,2]
Faming Fang[1,2*]
[1] School of Computer Science and Technology, East China Normal University, Shanghai, China.
[2] The KLATASDS & Shanghai Key Laboratory of MIP.
{yan_zaoming,pengchenglei1995}@163.com {ymhuang, qizhouchen}@stu.ecnu.edu.cn
{gxzhang, fmfang}@cs.ecnu.edu.cn

Abstract

Large motion poses a critical challenge in Video Frame Interpolation (VFI) task, as it requires accurate modeling of object correspondences across frames. Existing methods primarily rely on convolutional or attention-based models, which operate at the pixel or patch level. This inherently limits them to local object correspondences, making it difficult to capture frame-level object correspondences and often leading to failure under large motion. Inspired by the fundamental theorem of surface, we explore frame-level object correspondences through the lens of differential surface. The core idea is to represent video frames as 3D surfaces and align them by matching their surface properties, thereby achieving global surface alignment and frame-level object alignment. To implement the core idea, we propose the Surface-Aware Feed-Forward Quadratic Gaussian framework, mainly consisting of the Feed-Forward Quadratic Gaussian and Surface Properties modules. Feed-Forward Quadratic Gaussian is designed to map frames to Quadratic Gaussian, which flexibly fits the object surface. Unlike previous methods that compute local correspondences, Surface Properties facilitates global surface-level alignment, which drives object correspondence alignment. Finally, we rasterize the surface properties onto the interpolated camera plane and define loss functions to supervise alignment explicitly. The outstanding performance on the large motion benchmark demonstrates the effectiveness of our framework.

1 Introduction

Video Frame Interpolation (VFI) is a fundamental low-level vision task that aims to increase the frame rate of a video by synthesizing intermediate frames between consecutive inputs. It has a wide range of real-world applications, including slow-motion video generation [1, 2, 3], video compression [4, 5], and novel view synthesis [6, 7, 8]. Despite recent progress, VFI remains challenging, particularly in large and complex motion commonly found in casually captured videos. More recently, the emergence of film agent frameworks [9, 10, 11] has introduced intelligent agents for cinematic content creation. In such scenarios, handling large motion is critical for tasks such as scene composition, camera planning, and visual continuity. These demands highlight the need for more robust VFI methods capable of modeling long-range object correspondences.

At its core, VFI requires establishing accurate correspondences between objects across frames [12]. Video frame interpolation methods can be broadly divided into two types: those based on

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

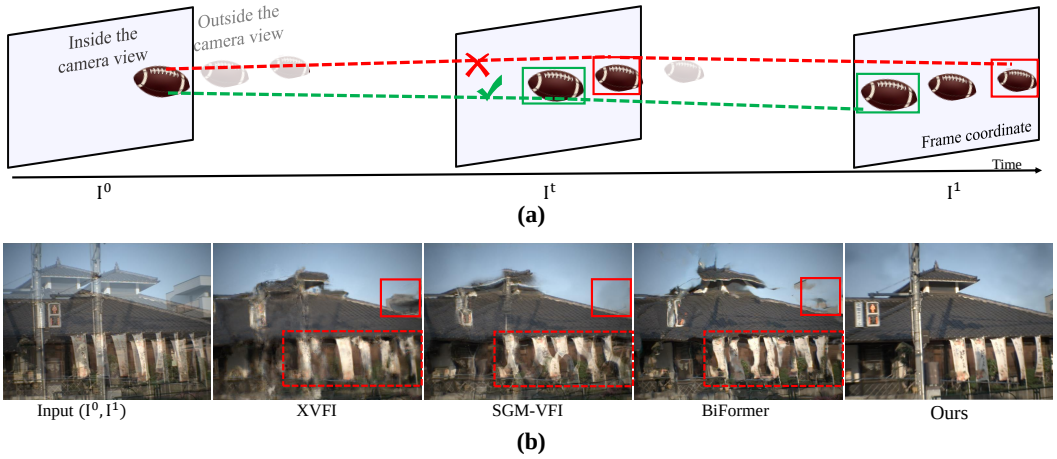

Figure 1: Challenge: Existing methods lack global frame-level object correspondences, which results in suboptimal matching.

Convolutional Neural Networks (CNNs) and those based on attention mechanisms. CNN-based methods [3, 13, 14, 15, 16, 17, 18, 12] estimate optical flow to synthesize intermediate frames, where the optical flow serves as a proxy for object correspondences. However, due to the local receptive field of convolution operations, CNN-based methods often compute object correspondences only at the pixel level, which limits their ability to capture global relationships. To overcome the locality of CNNs, attention-based architectures have been introduced for video frame interpolation [19, 20]. These methods divide input frames into patches and compute attention between them. However, this patch-level modeling still lacks a true global understanding of object correspondences across frames.

Upon scrutinizing and experimenting on the released implementations of existing methods [17, 15, 18, 12], we observe that they suffer from performance deterioration in large motion. As illustrated in Figure 1, in the large motion, existing approaches fail to establish accurate object correspondences, leading to misalignment results. We attribute these failures to the inherent locality of convolution- and attention-based methods, which operate at the pixel or patch level and thus struggle to model frame-level object correspondences. We further identify two representative failure cases (Figure 1 (b)): **(i)** Repeated objects across frames confuse optical flow estimation, resulting in incorrect local correspondences. For example, the red dashed lines show mismatched flags, causing fragmentation in the interpolated frame. **(ii)** Some objects required in the target frame are missing in adjacent frames, leading to ambiguous correspondences [21], as shown in the upper-right region.

As discussed above, overcoming the limitations of local object correspondences, particularly in the presence of large motion, necessitates the establishment of frame-level object correspondences. However, how to theoretically formulate and practically implement such correspondences requires further exploration and discussion. Inspired by the fundamental theorem of surfaces[22], we propose to represent video frames as differential surfaces (Figure 2), enabling the exploration of geometric constraints and frame-level correspondences. The core idea of this work is to map video frames onto 3D surfaces and align the overlapping regions using surface properties. This approach drives global surface alignment and promotes consistent geometric structure across frames. However, implementing this idea presents two key challenges: **(1)** how to effectively represent surface, and **(2)** which surface properties are most suitable for alignment.

To address the surface representation challenge, we propose the **Feed-Forward Quadratic Gaussian**. The Quadratic Gaussian Splatting (QGS) [23] defines a Gaussian distribution on a quadratic paraboloid, allowing smooth transitions between convex and concave forms for flexible surface fitting. However, QGS assumes accurate camera poses, which limits its applicability in real-world scenarios where videos are casually captured. To overcome this, Feed-Forward Quadratic Gaussian efficiently transforms such video frames into 3D surface representations without relying on strict assumptions such as accurate camera poses or depth maps.

To facilitate surface alignment, we leverage two key surface properties: **normal and curvature**. Surface normals are widely used to ensure geometric continuity across frames [24]. However, normal alone may become unreliable and introduce errors within large motions (surfaces with rapid bends). This motivates the incorporation of higher-order geometric descriptors such as curvature, which can

capture fine-grained surface structures. Therefore, we jointly use surface normals and curvature as consistency constraints to ensure robust surface alignment across frames.

To explicitly supervise surface continuity, we rasterize the normal and curvature onto the camera plane and define loss functions. By leveraging these modules, we build a novel pipeline, named **Surface-Aware Feed-Forward Quadratic Gaussian**. Our method achieves state-of-the-art performance on large motion benchmarks [12]. Furthermore, we conduct extensive ablation studies to validate the contribution of each component.

The contributions of this work are summarized as: **(1)** Inspired by the Fundamental Theorem of Surfaces, we introduce a differential surface to present a video, to explore frame-level object correspondences and geometric constraints. **(2)** This paper introduces a Surface-Aware Feed-Forward Quadratic Gaussian framework that maps video frames into 3D surfaces, aiming to overcome the limited local correspondences. **(3)** Our pipeline illustrates state-of-the-art performance on the large motion benchmark.

2 Related Work

Video Frame Interpolation. Recently, advancements in deep learning have led to various methods for video frame interpolation. These methods can be broadly categorized into two main paradigms: reconstruction-based [25, 15, 14, 26, 27, 16, 28, 29, 30] and denoising diffusion probabilistic model (DDPM)-based [31, 32, 33, 34, 35]. (i) Initially, DVF [36] utilized a U-Net-like network to model two input frames and predicted the voxel flow for warping the two frames into the intermediate frame. To obtain the optical flow from the intermediate frame to the input frames, [15, 37] proposed distillation strategies to obtain the optical flow from the intermediate frame to the input frames. In large-scale motion scenarios, methods such as SGM-VFI [12], FILM[18], and XVFI [17] leverage enhanced global information in optical flow to establish accurate frame-to-frame correspondences for objects. These Kernel-based methods are implemented as separable convolutions [38], deformable convolutions [39, 40, 41]. (ii) Based on Denoising Diffusion Probabilistic Models (DDPM)[42], leverage generative techniques like DDPM to fill occlusions caused by motion. These DDPM-based approaches are implemented as score-based diffusion [43, 44, 45], motion-aware diffusion [46, 47], and Brownian bridge diffusion [48, 49]. However, most DDPM-based methods are significantly time-consuming and challenging for real-time inference.

3D Gaussian Splatting. In recent years, 3D Gaussian splatting has emerged as an active area of research in the field of 3D reconstruction. Various approaches have been proposed across different domains, broadly categorized into static scene Gaussian splatting and dynamic scene Gaussian splatting. Gaussian Splatting [50] enhances rendering quality in radiance fields. To further adapt to diverse reconstruction scenarios, [51, 52, 53] have been proposed, significantly improving the generalization capability of 3DGS-based reconstruction. To accommodate dynamic scenes, [54, 55, 56, 57, 58, 59] has been extended to handle such environments. However, the per-scene optimization of 3DGS requires densely captured images and sparse point cloud generated by SfM for initialization. Recent works [60, 61, 62, 63, 64, 65] have explored feed-forward models for sparse-view Gaussian reconstruction by capitalizing on large-scale datasets and scalable model architectures[66, 67, 68].

3 Preliminary

3.1 3D Gaussian Splatting

Kerbl et al. [69] proposed representing a scene using 3D Gaussian ellipsoids as primitives and rendering images using differentiable volume splatting. Associates a 3D Gaussian i with a position μ_i, covariance matrix Σ_i, opacity o_i and spherical harmonics (SH) coefficients h_i. The final opacity of a 3D Gaussian at any spatial point $\mathbf{p} = (x, y, z)$ is:

$$\alpha_i = o_i \underbrace{\exp\left(-\frac{1}{2}(\mathbf{p}-\mu_i)^T\Sigma^{-1}(\mathbf{p}-\mu_i)\right)}_{\mathcal{G}}, \tag{1}$$

where the covariance matrix $\Sigma = RSS^TR^T$, and $\mathcal{G}$ is Gaussian distribution.

To render an image, 3D Gaussians are first projected to 2D image space via an approximation of the perspective transformation. Specifically, the projection of a 3D Gaussian is approximated as a 2D

Gaussian with center μ_i^{2D} and covariance Σ_i^{2D}. Center μ_i^{2D} and covariance Σ_i^{2D} are computed as

$$\mu_i^{2D} = (K(W\mu_i)/(W\mu_i)_z), \quad \Sigma^{2D} = JW\Sigma_i W^T J^T, \tag{2}$$

where W is a transformation from the world space to the camera space, and J is a local affine transformation.

After sorting the Gaussians in depth order, the color at a pixel is obtained by volume rendering:

$$I(u,v) = \sum_{i=0}^{N-1} \mathbf{c}_i \alpha_i^{2D} \prod_{j=0}^{i-1} (1 - \alpha_j^{2D}). \tag{3}$$

Here, α_i^{2D} is a 2D version of Eq. (1), with μ_i, Σ_i, $\mathbf{P}$ replaced bu μ_i^{2D}, Σ_i^{2D}, (u,v) (pixel coordinate). $\mathbf{c}_i$ is the RGB color after evaluating SH with the view direction.

3.2 Quadratic Gaussian

Zhang et al. [23] proposed representing a scene using a Quadratic Gaussian as a surface and rendering images using differentiable volume splatting. For convenience, the Quadratic Gaussian distribution is expressed in cylindrical coordinates, and the opacity of a Quadratic Gaussian at any spatial point $\mathbf{p} = (\theta, \rho, z(\theta,\rho))$ is:

$$\alpha_i = o_i \underbrace{\exp\left(-\frac{(\mu_i(\theta,\rho))^2}{2(\sigma_i(\theta))^2}\right)}_{\mathcal{G}}, \tag{4}$$

$$\sigma_i(\theta) = \frac{s_1 s_2}{\sqrt{(s_2\cos\theta)^2 + (s_1\sin\theta)^2}}, \quad \mu_i(\theta,\rho) = \int_0^\rho \sqrt{1 + (2at)^2}\, dt \tag{5}$$

where $\mu_i(\cdot)$ [23] is the mean of the Gaussian distribution on the surface, and σ_i is the covariance of the Gaussian distribution on the surface. $\mathcal{G}$ is defined as the corresponding Gaussian function. $S = diag(s_1, s_2, s_3)$ denotes the scale of the Quadratic Gaussian. $a(\cdot)$ is related to the coefficient of θ.

To render an image, the Quadratic Gaussian follows the same 3D Gaussian splatting way, which is projected to 2D image space via an approximation of the perspective transformation. After splatting and sorting the Gaussians in depth order, the color at a pixel (u,v) is obtained by rendering [70]:

$$I(u,v) = \sum_{i=0}^{N-1} \mathbf{c}_i \alpha_i^{2D} \prod_{j=1}^{i-1} \left(1 - \alpha_j^{2D}\right) \tag{6}$$

where N denotes the pixel numbers of the rendered image.

4 Method

4.1 Problem formulation

Frame Interpolation.

In the video frame interpolation task, it can be written as

$$I^t = \mathbf{F}(I^0, I^1), t \in (0,1), \tag{7}$$

where I^0 and I^1 are input frames. 0 and 1 are two input views and t is an interpolated view index between 0 and 1. To synthesize an intermediate frame I^t where $t \in (0,1)$, existing algorithms [3, 16, 12, 29, 30] typically extract object correspondences between two consecutive frames, facilitating object alignment.

Frame Interpolation under Differential Surface.

In large motion scenes, video frame interpolation methods often fall into suboptimal object correspondences, as they operate at the pixel level or patch level and thus struggle to model frame-level object correspondences.

To address the limitations of local object correspondences, we redefine the large motion problem as a global frame-level alignment task by aligning the surfaces that represent each frame. The core idea of frame-level alignment is that aligning the geometric properties between surfaces (such as normals and curvatures) can facilitate the alignment between surfaces, as illustrated in Figure 2.

In the following sections, we describe how to construct surface representations from video frames and how to select surface properties to facilitate accurate correspondences across surfaces.

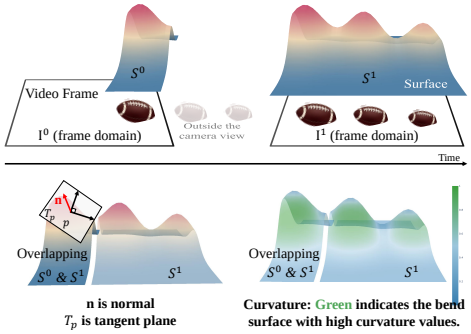

Figure 2: The core idea is mapping video frames into 3D surfaces and aligning them by matching surface properties, leading to global surface-level alignment.

4.2 Surface-Aware Feed-Forward Quadratic Gaussian

Although modeling object correspondences through differential surface representations is theoretically reasonable, it remains challenging to implement: **1)** What type of 3D primitives is suitable for representing a differential surface? **2)** Which surface properties should be selected to facilitate alignment between differential surfaces?

4.2.1 Feed-Forward Quadratic Gaussian

Modeling the texture and surface details in videos remains challenging for 3D primitives such as point clouds and 3DGS, which often struggle to accurately represent complex surfaces. Fortunately, Quadratic Gaussian Splatting (QGS) [23] is defined on a paraboloid and constructs Gaussian distributions based on geodesic distances. This enables the energy of the Gaussians to be concentrated on the surface, thereby effectively capturing complex surface and textural details. However, existing methods such as 3DGS [50], 2DGS [71], and QGS [23] still rely on accurate camera poses, which are difficult to obtain in sparse-view or unconstrained settings, thereby limiting their practical applicability. To overcome this limitation, the practical **Feed-Forward Quadratic Gaussian** is introduced that efficiently transforms frames into 3D surface representations.

Given a set of video frames, the goal of Feed-Forward Quadratic Gaussian is to generate the object surface in the QGS representation. Unlike prior methods, it does not require additional data such as camera poses, enabling single feed-forward inference. Feed-Forward Quadratic Gaussian mainly includes two sub-models: a backbone and a Quadratic Gaussian head. Formally, Feed-Forward Quadratic Gaussian can be written as:

$$f_\theta : \{I^0, I^1\} \mapsto \{\mathcal{P}^0, \mathcal{P}^1, \mathcal{C}^0, \mathcal{C}^1, \mathcal{F}^0, \mathcal{F}^1\}, \quad h_\theta : \{\mathcal{P}^0, \mathcal{P}^1, \mathcal{F}^0, \mathcal{F}^1\} \mapsto \{\mathcal{G}^0, \mathcal{G}^1\}, \tag{8}$$

f_θ is the backbone and h_θ is the Quadratic Gaussian Head. $\mathcal{F}^0, \mathcal{F}^1$ is the frame features. $\mathcal{P}^0, \mathcal{P}^1$ is the 3D point clouds. $\mathcal{C}^0, \mathcal{C}^1$ is the camera parameters. $\mathcal{G}^0, \mathcal{G}^1$ is the Quadratic Gaussian.

Backbone. Foundation models for 3D reconstruction (e.g., DUSt3R[66], MASt3R [67]) have shown remarkable competitiveness and superior performance in 3D reconstruction tasks [63]. We leverage pretrained geometric priors from foundation models to achieve a coarse alignment of 3D point clouds $\mathcal{P}$ and camera parameters $\mathcal{C}$, promoting a stable and efficient learning process. For simplicity and stability in our pipeline, we adopt a simple backbone VGGT [72]. Specifically, given a pair of input frames I^0, I^1, the backbone outputs the corresponding image features $\mathcal{F}^0, \mathcal{F}^1$, 3D point clouds $\mathcal{P}^0, \mathcal{P}^1$, and camera parameters $\mathcal{C}^0, \mathcal{C}^1$.

Quadratic Gaussian Head. Real-world object surfaces are complex, making it difficult for point clouds to capture their surface structure accurately. QGS [23] defines Gaussian distributions on a quadratic surface, which can smoothly transition between convex and concave shapes. This flexibility allows for more accurate modeling of complex object surfaces. To leverage this capability, we propose the Quadratic Gaussian Head, a module inspired by QGS, that transforms point cloud features into QGS-based 3D primitives, enabling more effective surface representation. This surface representation enables estimating surface properties in subsequent stages, thereby preserving surface consistency across frames. Specifically, QGS contains the following parameters

$$\mathcal{G}^0 = \{\mu_i^0, o_i^0, r_i^0, s_i^0, c_i^0\}_{i=1,\ldots,H\times W}, \tag{9}$$

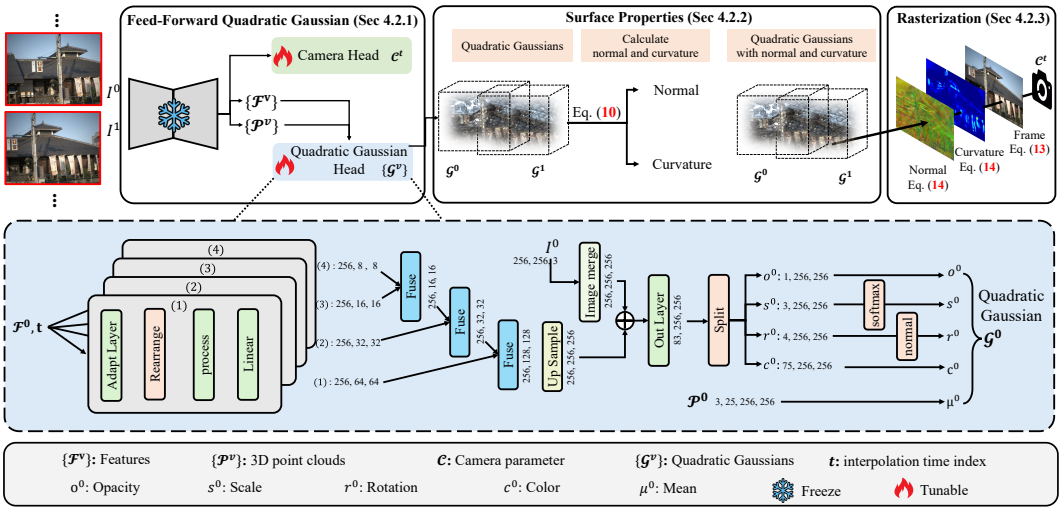

Figure 3: **Surface-Aware Feed-Forward Quadratic Gaussian** Framework. The Feed-Forward Quadratic Gaussian module transforms the input frames I^0 and I^1 into Quadratic Gaussian ($\mathcal{G}^0$ and $\mathcal{G}^1$) that represent surfaces. Then, the Surface Properties module computes the normal and curvature from the Quadratic Gaussian. Finally, the normal, curvature, and frame I^t are rasterized onto the camera plane at the interpolation time.

where opacity o, rotation r, scale s and color c and $H \times W$ pixel numbers. Subsequently, the point cloud $\mathcal{P}^0$ is the QGS's mean μ. Together with the predicted parameters, it forms the Quadratic Gaussian representation $\mathcal{G}_0$ of frame I^0.

4.2.2 Surface Properties

Normals usually play a crucial role in facilitating surface alignment and continuity across frames [73, 74]. Meanwhile, curvature characterizes the degree of surface bending [75, 76]. In regions with large motion, i.e., highly curved surface areas, relying solely on normals may lead to inaccurate alignment [77, 78]. This motivates employing higher-order geometric descriptors, such as curvature, to complement normal in fine-grained alignment surfaces. Therefore, we jointly utilize normals and curvature as surface constraints to ensure accurate and consistent alignment across frames.

Given a Quadratic Gaussian, we directly compute the normal and curvature at any point on the surface. Specifically, given any 3D point $\mathbf{p}$ on the surface [23], the corresponding normal and curvature are:

$$\mathbf{n}(\mathbf{p}) = \left(2\lambda_x x, 2\lambda_y y, -\frac{1}{s_3}\right), \quad \mathbf{k}(\mathbf{p}) = \frac{4\lambda_x \lambda_y}{\left(1 + 4\lambda_x^2 x^2 + 4\lambda_y^2 y^2\right)^2}, \tag{10}$$

where $\lambda_x = \frac{d_x}{s_1^2}$, and $d_x \in \{-1, 0, 1\}$ determines whether the paraboloid is elliptic, hyperbolic, or planar. More specifically, the d_x idepends on both the positional variable x and a temporal variable t, and is defined as $d_x = \tanh(t)\exp(x)$. For the detailed computation, please refer to Appendix A.1. Formally, the calculation process is written as:

$$\{\mathcal{G}^0, \mathcal{G}^1\} \mapsto \{N^0, K^0, N^1, K^1\}, \tag{11}$$

where N and K represent the normal map and the curvature map, respectively.

4.2.3 Rasterization

Finally, we rasterize the surface properties from the 3D space into the camera plane with the interpolation camera parameter. The total process can be written as:

$$\{\mathcal{G}^0, \mathcal{G}^1, N^0, K^0, N^1, K^1, \mathcal{C}^t\} \mapsto \{I^t, N^t, K^t\}, \tag{12}$$

where $\mathcal{C}^t = \mathcal{C}^0 \times t + \mathcal{C}^1 \times (1 - t)$ is the interpolation camera parameter.

Table 1: Quantitative comparison with SOTA methods on the standard benchmark, regarding PSNR/SSIM. The best and the second best results are denoted by pink and yellow.

	Vimeo-90K [13]	UCF101[79]	SNU-FILM[80]				Average
			easy	medium	hard	extreme	
DAIN[25]	34.71/0.9756	34.39/0.9683	39.73/0.9902	35.46/0.9780	30.17/0.9335	25.09/0.8584	33.36/0.9507
AdaCoF[40]	34.47/0.9730	34.90/0.9680	39.80/0.9900	35.05/0.9754	29.46/0.9244	24.31/0.8439	33.00/0.9458
CAIN[80]	34.65/0.9730	34.91/0.9690	39.89/0.9900	35.61/0.9776	29.90/0.9270	24.78/0.8507	33.29/0.9493
Softsplat[27]	36.13/0.9805	35.17/0.9690	40.26/0.9911	36.09/0.9798	30.93/0.9365	25.16/0.8608	33.92/0.9530
XVFI[17]	35.09/0.9759	35.17/0.9685	39.93/0.9907	35.37/0.9776	29.58/0.9276	24.17/0.8450	33.22/0.9477
M2M-VFI[21]	35.20/0.9768	35.28/0.9697	40.10/0.9906	36.12/0.9797	30.63/0.9368	25.27/0.8601	33.68/0.9519
RIFE[15]	35.61/0.9779	35.29/0.9697	40.10/0.9906	36.12/0.9797	30.63/0.9368	25.27/0.8601	33.68/0.9519
IFRNet-L[81]	36.20/0.9808	35.42/0.9698	40.36/0.9910	36.12/0.9797	30.63/0.9368	25.27/0.8609	33.96/0.9531
AMT-L[82]	36.35/0.9815	35.39/0.9696	39.95/0.9913	36.09/0.9805	30.75/0.9384	25.41/0.8638	33.99/0.9542
EMA-VFI-S[20]	36.64/0.9819	35.48/0.9701	39.98/0.9905	36.09/0.9801	30.94/0.9392	25.69/0.8661	34.14/0.9547
SGM-VFI[12]	35.81/0.9793	35.40/0.9693	40.14/0.9907	36.06/0.9795	30.81/0.9375	25.69/0.8661	33.96/0.9535
VFIMamba-S[29]	36.09/0.9800	35.36/0.9696	40.21/0.9909	36.17/0.9800	30.80/0.9381	25.59/0.8655	34.14/0.9540
Ours	36.06/0.9791	35.40/0.9692	39.98/0.9906	36.10/0.9798	30.90/0.9391	25.50/0.8651	33.35/0.9475

Table 2: Quantitative comparison with VFI methods on large motion benchmark. The best and the second best results are denoted by pink and yellow.

	X-Test-L [12]		SNU-FILM-L[12]		Xiph-L[12]		Runtime (s)	FLOPs (T)
	2K	4K	hard	extreme	2K	4K		
XVFI[17]	29.82/0.8951	29.02/0.8866	27.58/0.9095	22.99/0.8260	29.17/0.8449	28.09/0.7889	0.075	0.37
FILM[18]	30.08/0.8941	29.10/0.8886	28.35/0.9156	23.06/0.8247	29.89/0.8533	27.11/0.7699	1.29	1.36
BiFormer[19]	30.32/0.9067	30.15/0.9070	28.18/0.9154	23.85/0.8393	29.61/0.8541	28.98/0.8183	1.09	0.39
RIFE[15]	29.87/0.8805	28.98/0.8756	28.19/0.9172	22.84/0.8230	30.18/0.8633	28.07/0.7982	0.20	0.2
AMT-L[82]	29.39/0.8771	28.35/0.8731	28.33/0.9184	23.14/0.8288	30.32/0.8710	28.27/0.8095	0.58	0.58
EMA-VFI-S[20]	29.51/0.8775	28.60/0.8733	28.57/0.9189	23.18/0.8292	30.54/0.8718	28.40/0.8109	0.076	0.91
SGM-VFI[12]	30.39/0.8946	29.25/0.8861	28.90/0.9209	23.19/0.8301	30.89/0.8745	28.59/0.8115	0.93	1.79
VFIMamba-S[29]	31.58/0.9169	30.50/0.9077	28.80/0.9208	23.41/0.8300	30.72/0.8780	28.62/0.8111	0.128	0.24
Ours	31.33/0.9011	30.13/0.9066	29.05/0.9213	24.20/0.8400	31.20/0.8814	29.19/0.8197	0.340	1.28

Specifically, the rasterization of color is performed according to the following equation:

$$I^t(u,v) = \sum_{i=0}^{N-1} \mathbf{c}_i \alpha_i^{2D} \prod_{j=0}^{i-1} (1 - \alpha_i^{2D}). \tag{13}$$

Similarly, [23] renders the normal and curvature on the camera plane as follows:

$$N^t(u,v) = \sum_{i=0}^{N-1} \mathbf{n}_i \alpha_i^{2D} \prod_{j=0}^{i-1} (1 - \alpha_i^{2D}), \quad K^t(u,v) = \sum_{i=0}^{N-1} \mathbf{k}_i \alpha_i^{2D} \prod_{j=0}^{i-1} (1 - \alpha_i^{2D}), \tag{14}$$

where N denotes the pixel number of the rendered image.

4.3 Loss Function

Finally, we minimize the following loss function:

$$\mathcal{L} = \mathcal{L}_c + \alpha \mathcal{L}_{kn}, \quad \mathcal{L}_{kn} = (1 - \text{sigmoid}(\ln(|K(u,v)|) + \varepsilon)) \mathcal{L}_n \tag{15}$$

where $\mathcal{L}_c = \|I_{gt}^t - I^t\|_2$ is an RGB reconstruction loss function. $\mathcal{L}_{kn}(u,v)$ denotes the curvature-aware normal loss, which enforces surface alignment [23]. And $\mathcal{L}_n$ [71] is the normal consistency loss to ensure primitives are locally aligned with the surface.

5 Experiments

Metrics. We use common quantitative metrics: Peak Signal-To-Noise Ratio (PSNR) and Structural Similarity Image Metric (SSIM), where higher scores indicate better image quality. To assess temporal consistency between frames, we additionally employ the tOF metric [17].

To further illustrate the effectiveness of our algorithm in addressing large motion scenarios, we present a statistical analysis of the relationship between motion magnitudes [8, 18] and the corresponding

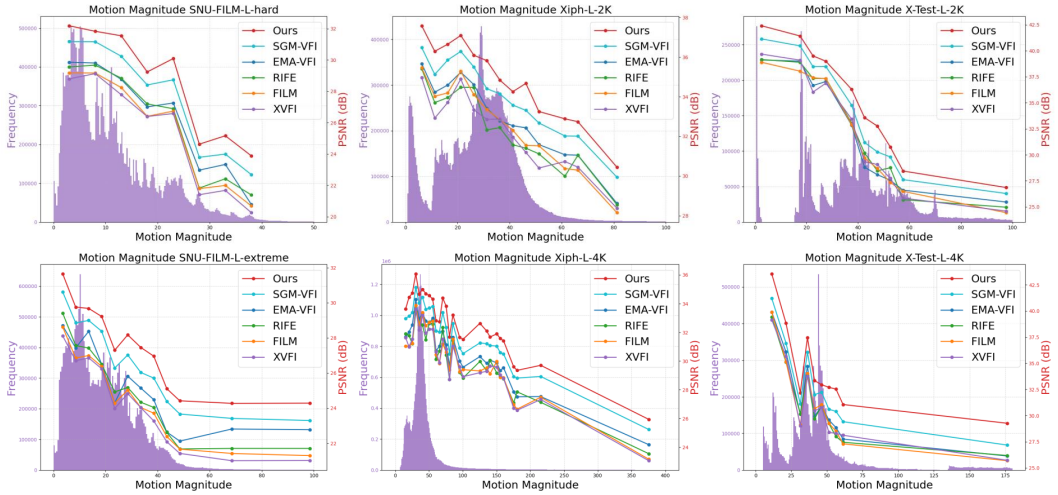

Figure 4: PSNR versus motion magnitude. Higher motion magnitudes correspond to larger inter-frame displacements, representing more challenging motion scenarios.

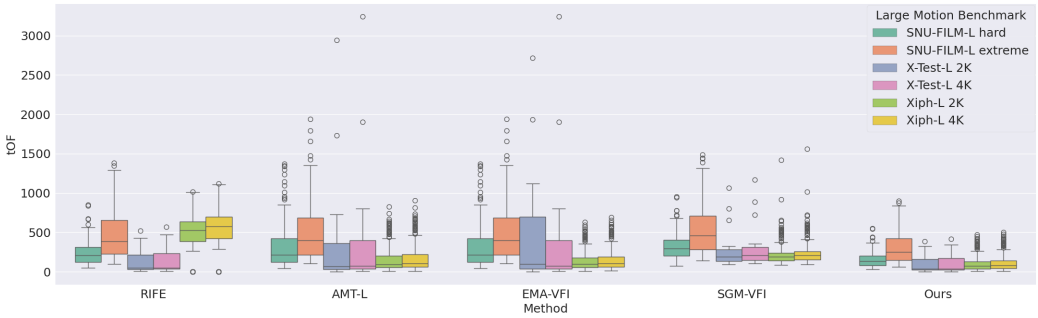

Figure 5: Comparison of temporal consistency under the tOF metric.

PSNR performance, as shown in Figure 4. Note that a higher motion magnitude corresponds to larger inter-frame displacement, indicating more challenging motion scenarios.

Datasets. For fair comparison, we follow the training and testing datasets established by the large motion benchmark [12]. For training, we follow the setting [12], utilizing both the Vimeo90K and X4K1000FPS (X-Train) datasets. Vimeo90K [13] consists of 51,312 triplets with a resolution of 448×256, featuring an average motion magnitude between 1 and 8 pixels. X4K1000FPS (X-Train) [17] contains 4,408 clips at a resolution of 768×768, with each clip comprising 65 consecutive frames.

We evaluate its performance following the large motion benchmark introduced by SGM-VFI [12]. X-Test-L [17, 12] with the largest temporal gap, as our primary benchmark for evaluating large motion scenarios. We also choose the 0th and 32nd frames as input and evaluate the quality of the synthesized 16th output frame. SNU-FILM-L [80, 12] is the most challenging half of the SNU-FILM hard and extreme, with 155 triplets each. Xiph-L [12] is constructed based on the original Xiph dataset [83] by doubling the input temporal intervals and retaining the most challenging half of the data to form this benchmark.

Implementation Detail. We optimize the loss using Adam in PyTorch framework. The cosine scheduler schedules the learning rate from $1e{-}4$ to $1e{-}6$. Standard data augmentation techniques, such as flipping, rotation, and cropping, are applied to the data with a size of 518×280. We train our model on the training datasets with a batch size 16 for 800 epochs.

5.1 Comparison with Previous Methods

As shown in the Table 1 and 2, methods are compared, which are tested on the standard and large motion benchmark. To comprehensively evaluate the capability of our model, we conduct experiments on benchmarks with varying motion magnitudes. We compare our method against recent video frame

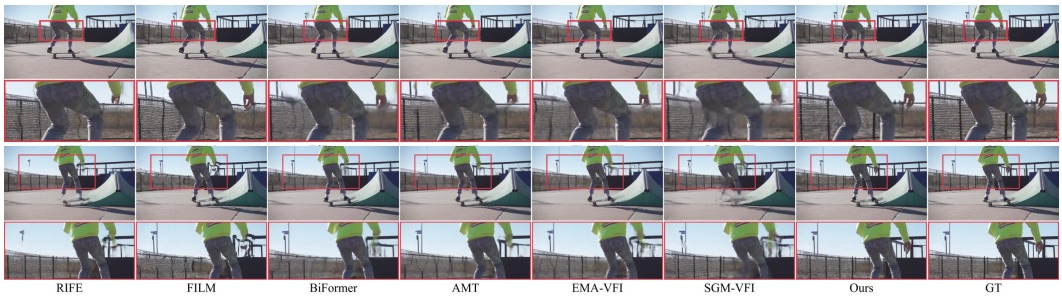

Figure 6: Visual comparison with state-of-the-art methods.

interpolation (VFI) approaches, including those specifically designed for large motion, as well as methods that perform well on standard benchmarks.

Large Motion. The large motion benchmark contains a significant number of scenes with large inter-frame displacements, resulting in motion magnitudes that are higher than those in standard benchmarks. As shown in Table 2, our method consistently outperforms state-of-the-art approaches on the large motion benchmark, demonstrating its effectiveness in handling complex motion scenarios. To further analyze performance under varying motion magnitude, we examine the PSNR across different motion magnitude intervals, as illustrated in Figure 4.

Comparison of Temporal Consistency. We employ the tOF metric [17] to evaluate the temporal consistency. As shown in Figure 5, our method consistently outperforms existing approaches on the large motion benchmark in terms of both the mean and variance of tOF, indicating more stability. This superior temporal consistency can be attributed to our method's accurate modeling of surface properties, which enables fine alignment of object correspondences across frames. We employ the tOF metric [17] to evaluate the temporal consistency. Figure 5 reports the quality of motion reconstruction across several existing models on the large motion benchmark. We can clearly observe that our method consistently outperforms existing methods in both the mean and variance of tOF. Outstanding temporal consistency is due to our method's fine alignment of object correspondence by modeling the surface properties, which improves motion temporal consistency.

Comparison of Visual Results. We further compare the visualization results in large motion. Figure 6 compares our method and several state-of-the-art approaches. CNN-based methods estimate pixel-level object correspondences, which often fall into local optima under large motion, leading to subpar interpolation results. Attention-based methods estimate patch-level object correspondences, which improves interpolation results under large motion.

5.2 Ablation Study

In this section, we present experimental insights to analyze and discuss the questions raised in the previous section: which 3D primitives are suitable for representing differential surfaces, and which surface properties facilitate surface alignment.

Table 3: Ablation studies of Architecture on SNU-FILM-L extreme.

	Backbone			Gaussian Head		Perfermence	
Setting	CUT3R	MonST3R	VGGT	3DGS head	QGS head	PSNR	tOF(↓)
(i)	✓			✓		23.27	308
(ii)	✓				✓	24.11	240
(i)		✓		✓		23.25	297
(ii)		✓			✓	24.12	236
(i)			✓	✓		23.30	286
(ii)			✓		✓	24.20	226

Architecture. To demonstrate the superiority of the QGS head in capturing complex geometric structures compared to the 3DGS head, we first compare their texture representations. Figure 7 provides qualitative evidence, while Table 3 offers quantitative validation of the QGS head's effectiveness.

Figure 7 presents a visual comparison between the two heads, highlighting the QGS head's ability to preserve fine-grained geometric details. As shown in the error maps of Figure 7, the surface-aware QGS head achieves significantly better reconstruction quality, particularly in regions with complex textures.

To further isolate the contribution of the QGS head from that of the backbone, we conduct an ablation study by combining different backbones, CUT3R [84], VGGT [72], and MonST3R [85], with both the 3DGS and QGS heads. Table 3 summarizes the performance of these combinations. The results show that the QGS head consistently outperforms all other configurations, especially under challenging

large motion scenarios, demonstrating its effectiveness in improving both reconstruction quality and temporal consistency.

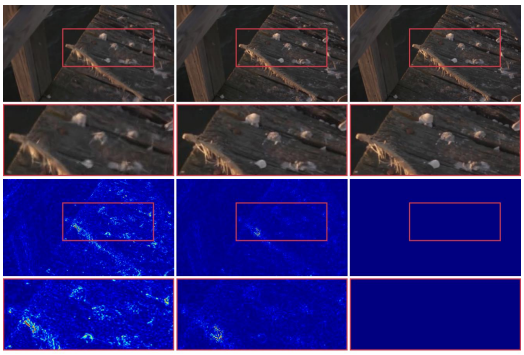

Figure 7: A visual comparison between the 3DGS head and the QGS head. The bottom part shows a heatmap of the interpolated frame error.

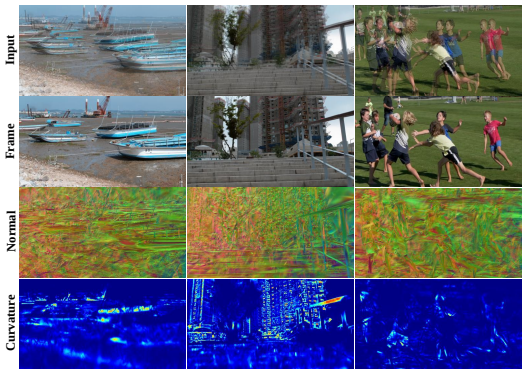

Figure 8: We visualize the surface properties, and notably, as discussed earlier, the curvature highlights motion regions, particularly those corresponding to rapidly bending surfaces.

Surface properties. We conduct an ablation study to further investigate the role of surface properties in enhancing surface alignment. Both qualitative and quantitative results are presented in Figure 8 and Table 4. Figure 8 visualizes the surface normals and curvature maps rendered by the QGS head. Notably, the curvature map highlights regions with high surface variation, such as bends and folds. This observation supports our earlier analysis that curvature serves as a higher-order geometric descriptor, complementing surface normals and facilitating more accurate surface alignment. These additional insights contribute to improved video frame interpolation performance in large motion.

Table 4: Ablation studies of Surface properties on X-Test-L 2K.

	Surface Properties			**Perfermence**	
Setting	RGB	Normal	Curvature	PSNR	tOF (↓)
(i)	✓			29.07	183
(ii)	✓	✓		30.47	117
(iii)	✓	✓	✓	31.33	96

6 Conclusion

This work is the first to analyze frame-level object correspondence under large motion from the perspective of differential surface. Building on this insight, we propose an explicit Surface-Aware Feed-Forward Quadratic Gaussian pipeline to mitigate the challenge. Specifically, the proposed method transforms video frames into Quadratic Gaussians representing differential surfaces. Within this representation, we compute corresponding surface properties, such as normal and curvature. These properties are rendered onto the camera plane for explicit supervision and alignment. Extensive experiments demonstrate that our framework achieves state-of-the-art performance on the large motion benchmark, highlighting its effectiveness and robustness in handling complex motion scenarios. This framework opens new avenues for incorporating differential surface into the video frame interpolation task, particularly under large motion conditions.

Limitation. While our pipeline can cover most cases of large motion, there are many other cases beyond that coverage. The main reason for the limitations is that our definition of large motion and the proposed ideas are somewhat naive, which makes the solution sub-optimal for geometry. Our current definition focuses more on static correspondences in the background regions across different frames. For dynamic correspondences, due to the relatively short time interval between adjacent frames, we adopt a simplified assumption of linear motion in this work. At present, we employ a relatively basic differential surface theory to model the problem. We believe that, in the future, a more unified modeling of camera motion and object motion within a comprehensive differential geometry framework could lead to a more accurate characterization of complex dynamic scenes.

Acknowledgment. This work was supported by the National Key R&D Program of China (2022ZD0161800), the National Natural Science Foundation of China under Grant 62271203, AI-Empowered Research Paradigm Reform and Discipline Leap Plan under Grant 2024AI01012 and the Open Research Fund of KLATASDS-MOE, ECNU.

References

[1] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang. Videoinr: Learning video implicit neural representation for continuous space-time super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2047–2057, 2022.

[2] Zhewei Huang, Ailin Huang, Xiaotao Hu, Chen Hu, Jun Xu, and Shuchang Zhou. Scale-adaptive feature aggregation for efficient space-time video super-resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4228–4239, 2024.

[3] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018.

[4] Zhaoyang Jia, Yan Lu, and Houqiang Li. Neighbor correspondence matching for flow-based video frame synthesis. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5389–5397, 2022.

[5] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 416–431, 2018.

[6] Zhihang Zhong, Mingdeng Cao, Xiang Ji, Yinqiang Zheng, and Imari Sato. Blur interpolation transformer for real-world motion from blur. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5713–5723, 2023.

[7] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016.

[8] Tim Brooks and Jonathan T Barron. Learning to synthesize motion blur. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6840–6848, 2019.

[9] Zhenran Xu, Jifang Wang, Longyue Wang, Zhouyi Li, Senbao Shi, Baotian Hu, and Min Zhang. Filmagent: Automating virtual film production through a multi-agent collaborative framework. In *SIGGRAPH Asia 2024 Technical Communications*, pages 1–4. 2024.

[10] Sangmin Kim, Seunguk Do, and Jaesik Park. Showmak3r: Compositional tv show reconstruction. *CVPR*, 2025.

[11] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning*, 2024.

[12] Chunxu Liu, Guozhen Zhang, Rui Zhao, and Limin Wang. Sparse global matching for video frame interpolation with large motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19125–19134, 2024.

[13] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019.

[14] Xin Jin, Longhai Wu, Jie Chen, Youxin Chen, Jayoon Koo, and Cheul-hee Hahm. A unified pyramid recurrent network for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1578–1587, 2023.

[15] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*, pages 624–642. Springer, 2022.

[16] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Advances in Neural Information Processing Systems*, 32, 2019.

[17] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14489–14498, 2021.

[18] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *European Conference on Computer Vision*, pages 250–266. Springer, 2022.

[19] Junheum Park, Jintae Kim, and Chang-Su Kim. Biformer: Learning bilateral motion estimation via bilateral transformer for 4k video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1568–1577, 2023.

[20] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5682–5692, 2023.

[21] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3553–3562, 2022.

[22] Elsa Abbena, Simon Salamon, and Alfred Gray. *Modern differential geometry of curves and surfaces with Mathematica*. Chapman and Hall/CRC, 2017.

[23] Ziyu Zhang, Binbin Huang, Hanqing Jiang, Liyang Zhou, Xiaojun Xiang, and Shunhan Shen. Quadratic gaussian splatting for efficient and detailed surface reconstruction. *arXiv preprint arXiv:2411.16392*, 2024.

[24] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.

[25] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3703–3712, 2019.

[26] Xin Jin, Longhai Wu, Guotao Shen, Youxin Chen, Jie Chen, Jayoon Koo, and Cheul-hee Hahm. Enhanced bi-directional motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5049–5057, 2023.

[27] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020.

[28] Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 41–56. Springer, 2020.

[29] Guozhen Zhang, Chuxnu Liu, Yutao Cui, Xiaotong Zhao, Kai Ma, and Limin Wang. Vfimamba: Video frame interpolation with state space models. *Advances in Neural Information Processing Systems*, 37:107225–107248, 2024.

[30] Zujin Guo, Wei Li, and Chen Change Loy. Generalizable implicit motion modeling for video frame interpolation. *Advances in Neural Information Processing Systems*, 37:63747–63770, 2024.

[31] Junhwa Hur, Charles Herrmann, Saurabh Saxena, Janne Kontkanen, Wei-Sheng Lai, Yichang Shih, Michael Rubinstein, David J Fleet, and Deqing Sun. High-resolution frame interpolation with patch-based cascaded diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3868–3876, 2025.

[32] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22930–22941, 2023.

[33] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024.

[34] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

[35] Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18444–18455, 2023.

[36] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE international conference on computer vision*, pages 4463–4471, 2017.

[37] Mengshun Hu, Kui Jiang, Zhihang Zhong, Zheng Wang, and Yinqiang Zheng. Iq-vfi: Implicit quadratic motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6410–6419, 2024.

[38] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017.

[39] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7029–7045, 2021.

[40] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5316–5325, 2020.

[41] Pengcheng Lei, Zaoming Yan, Tingting Wang, Faming Fang, and Guixu Zhang. Three-stage temporal deformable network for blurry video frame interpolation. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024.

[42] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[43] Siddhant Jain, Daniel Watson, Eric Tabellion, Ben Poole, Janne Kontkanen, et al. Video interpolation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7341–7351, 2024.

[44] Duolikun Danier, Fan Zhang, and David Bull. Ldmvfi: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1472–1480, 2024.

[45] Wen Wang, Qiuyu Wang, Kecheng Zheng, Hao Ouyang, Zhekai Chen, Biao Gong, Hao Chen, Yujun Shen, and Chunhua Shen. Framer: Interactive frame interpolation. *arXiv preprint arXiv:2410.18978*, 2024.

[46] Zhilin Huang, Yijie Yu, Ling Yang, Chujun Qin, Bing Zheng, Xiawu Zheng, Zikun Zhou, Yaowei Wang, and Wenming Yang. Motion-aware latent diffusion models for video frame interpolation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1043–1052, 2024.

[47] Zhihang Zhong, Gurunandan Krishnan, Xiao Sun, Yu Qiao, Sizhuo Ma, and Jian Wang. Clearer frames, anytime: Resolving velocity ambiguity in video frame interpolation. In *European Conference on Computer Vision*, pages 346–363. Springer, 2024.

[48] Zonglin Lyu, Ming Li, Jianbo Jiao, and Chen Chen. Frame interpolation with consecutive brownian bridge diffusion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3449–3458, 2024.

[49] Danyeong Lee, Dohoon Lee, Dongmin Bang, and Sun Kim. Disco: Diffusion schrödinger bridge for molecular conformer optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13365–13373, 2024.

[50] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[51] Wenbo Chen and Ligang Liu. Deblur-gs: 3d gaussian splatting from camera motion blurred images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1):1–15, 2024.

[52] Lingzhe Zhao, Peng Wang, and Peidong Liu. Bad-gaussians: Bundle adjusted deblur gaussian splatting. In *European Conference on Computer Vision*, pages 233–250. Springer, 2024.

[53] Yutong Chen, Marko Mihajlovic, Xiyi Chen, Yiming Wang, Sergey Prokudin, and Siyu Tang. Splatformer: Point transformer for robust 3d gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2025.

[54] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024.

[55] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024.

[56] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20331–20341, 2024.

[57] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024.

[58] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:101790–101817, 2024.

[59] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024.

[60] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19457–19467, 2024.

[61] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024.

[62] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10208–10217, 2024.

[63] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024.

[64] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024.

[65] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2(3):4, 2024.

[66] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[67] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[68] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024.

[69] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[70] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[71] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.

[72] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*, 2025.

[73] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13137–13146, 2021.

[74] Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9535–9545, 2024.

[75] He Chen and Gregory S Chirikjian. Curvature: A signature for action recognition in video sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 858–859, 2020.

[76] Wenchong He, Zhe Jiang, Chengming Zhang, and Arpan Man Sainju. Curvanet: Geometric deep learning based on directional curvature for 3d shape analysis. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2214–2224, 2020.

[77] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[78] Zhuoxiao Li, Shanliang Yao, Yijie Chu, Angel F Garcia-Fernandez, Yong Yue, Eng Gee Lim, and Xiaohui Zhu. Mvg-splatting: Multi-view guided gaussian splatting with adaptive quantile-based geometric consistency densification. *arXiv preprint arXiv:2407.11840*, 2024.

[79] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[80] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 10663–10671, 2020.

[81] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2022.

[82] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9801–9810, 2023.

[83] Christopher Montgomery and H Lars. Xiph. org video test media (derf's collection). *Online, https://media. xiph. org/video/derf*, 6, 1994.

[84] Qianqian Wang*, Yifei Zhang*, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025.

[85] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024.

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: **[TODO]**

   Guidelines:

   - The answer NA 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 NA 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: **[TODO]**

   Guidelines:

   - The answer NA 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: [Yes]

Justification: **[TODO]**

Guidelines:

- The answer NA 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: **[TODO]**

   Guidelines:

   - The answer NA 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: **[TODO]**

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA 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: **[TODO]**

   Guidelines:

   - The answer NA 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: **[TODO]**

   Guidelines:

   - The answer NA 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: **[TODO]**

   Guidelines:

   - The answer NA 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: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA 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: **[TODO]**

    Guidelines:

    - The answer NA 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: [No]

Justification: **[TODO]**

Guidelines:

- The answer NA 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: [No]

Justification: **[TODO]**

Guidelines:

- The answer NA 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: [No]

Justification: **[TODO]**

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: **[TODO]**

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Theoretical Supplement

A.1 Surface Properties of Quadratic Gaussian

A paraboloid is defined as:

$$f(x,y,z) = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \frac{d_x}{s_1^2} & 0 & 0 & 0 \\ 0 & \frac{d_y}{s_2^2} & 0 & 0 \\ 0 & 0 & 0 & -\frac{d_z}{2s_3} \\ 0 & 0 & -\frac{d_z}{2s_3} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \frac{d_x}{s_1^2}x^2 + \frac{d_y}{s_2^2}y^2 - \frac{d_z}{s_3}z = 0. \tag{16}$$

$\mathbf{S} = diag(s_1, s_2, s_3)$, which denotes the orientation and scale of the quadric in the object space. The matrix D defines the surface shape: $\mathbf{D} = diag(1,1,1,1)$ yields an ellipsoid, while $\mathbf{D} = diag(1,0,0,0)$ produces a plane. $d_{ii} \in \{0, \pm 1\}$ determines whether the paraboloid is elliptic, hyperbolic, or planar. For convenience in writing and subsequent derivations, we simplify Equation 16 as follows:

$$f(x,y,z) = \lambda_x x^2 + \lambda_y y^2 - \frac{1}{s_z}z = 0 \tag{17}$$

A.1.1 Normal

QGS is a surface-based representation that naturally possesses multiview consistent geometric properties, making it straightforward to compute surface normals. Given any point $\mathbf{p} = (x, y, z)$ on the surface, we can take the partial derivatives of Eq. 17, yielding:

$$\mathbf{n}(\mathbf{p}) = \left(2\lambda_x x, 2\lambda_y y, -\frac{1}{s_z}\right), \tag{18}$$

A.1.2 Curvature

Here, we compute the Gaussian curvature analytically using a standard differential geometry approach [22]. By the way, throughout the entire paper, the parameter domain is expressed using (u, v) coordinates, while the surface is represented using (x, y, z) coordinates. We simplify Eq. 17 as $z = \lambda_x x^2 + \lambda_y y^2$. Given the point $\mathbf{p} = (x, y, z)$, the partial derivatives are:

$$x_u = (1, 0, 2\lambda_x x) \tag{19}$$

$$x_v = (0, 1, 2\lambda_y y) \tag{20}$$

The first fundamental form is:

$$E = \langle x_u, x_u \rangle = 1 + 4\lambda_x^2 x^2 \tag{21}$$

$$F = \langle x_u, x_v \rangle = 4\lambda_x \lambda_y xy \tag{22}$$

$$G = \langle x_v, x_v \rangle = 1 + 4\lambda_y^2 y^2 \tag{23}$$

The second fundamental form is:

$$n = \frac{x_u \times x_v}{\|x_u \times x_v\|} = \frac{(-2\lambda_x x, -2\lambda_y y, 1)}{\sqrt{1 + 4\lambda_x^2 x^2 + 4\lambda_y^2 y^2}} \tag{24}$$

$$x_{uu} = (0, 0, 2\lambda_x) \tag{25}$$

$$x_{uv} = (0, 0, 0) \tag{26}$$

$$x_{vv} = (0, 0, 2\lambda_y) \tag{27}$$

$$L = \langle n, x_{uu} \rangle = \frac{2\lambda_x}{\sqrt{1 + 4\lambda_x^2 x^2 + 4\lambda_y^2 y^2}} \tag{28}$$

$$M = \langle n, x_{uv} \rangle = 0 \tag{29}$$

$$N = \langle n, x_{vv} \rangle = \frac{2\lambda_y}{\sqrt{1 + 4\lambda_x^2 x^2 + 4\lambda_y^2 y^2}} \tag{30}$$

Finally, the Gaussian curvature can be computed as:

$$\mathbf{k}(\mathbf{p}) = \frac{LN - M^2}{EG - F^2} = \frac{\frac{4\lambda_x \lambda_y}{1 + 4\lambda_x^2 x^2 + 4\lambda_y^2 y^2}}{1 + 4\lambda_x^2 x^2 + 4\lambda_y^2 y^2} = \frac{4\lambda_x \lambda_y}{\left(1 + 4\lambda_x^2 x^2 + 4\lambda_y^2 y^2\right)^2} \tag{31}$$

B More Visual Results

The anonymous GitHub repository provides visualization results in both video and 3D formats.