

# Global Motion Corresponder for 3D Point-Based Scene Interpolation under Large Motion

Junru Lin<sup>\*1,2</sup> Chirag Vashist<sup>\*3</sup> Mikaela Angelina Uy<sup>2,4</sup> Colton Stearns<sup>2</sup>

Xuan Luo<sup>5</sup> Leonidas Guibas<sup>2</sup> Ke Li<sup>3</sup>

<sup>1</sup>University of Toronto <sup>2</sup>Stanford University <sup>3</sup>Simon Fraser University <sup>4</sup>Nvidia <sup>5</sup>Google

<sup>\*</sup> equal contribution

<https://junrul.github.io/gmc/>

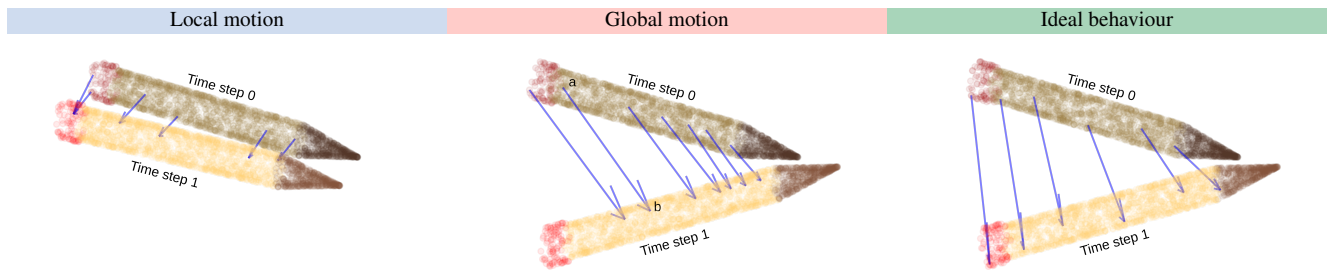


Figure 1. **Global Motion Challenge.** (1) Left: For small inter-frame motion, determining a point’s motion is effectively equivalent to matching it with a corresponding point within a small local neighborhood. In this case, local neighborhood searches yield correct correspondence and motion prediction. (2) Middle: With large global motion, local searches lead to a wrong correspondence between adjacent timesteps. (3) Right: An ideal method would be able to predict correct correspondence and achieve global motion.

## Abstract

Existing dynamic scene interpolation methods typically assume that the motion between consecutive timesteps is small enough so that displacements can be locally approximated by linear models. In practice, even slight deviations from this small-motion assumption can cause conventional techniques to fail. In this paper, we introduce **Global Motion Corresponder (GMC)**, a novel approach that robustly handles large motion and achieves smooth transitions. GMC learns unary potential fields that predict  $SE(3)$  mappings into a shared canonical space, balancing correspondence, spatial and semantic smoothness, and local rigidity. We demonstrate that our method significantly outperforms existing baselines on 3D scene interpolation when the two states undergo large global motions. Furthermore, our method enables extrapolation capabilities where other baseline methods cannot.

## 1. Introduction

Dynamic scene interpolation reconstructs continuous motion from two sets of discrete multi-view frames, which is a

fundamental challenge in computer vision driven by the increasing demand for photorealistic animation. Recent methods built on point-based representations [12, 34] are especially appealing as they enable efficient training and real-time rendering.

The core challenge in scene interpolation with point-based representations lies in establishing reliable correspondences: each point must predict its motion to a corresponding location in another frame. Most existing methods rely on a critical assumption that the motion between adjacent timesteps is small enough that point positions do not change significantly. Under this assumption, determining correspondence reduces to matching points within small local neighborhoods, which has been successfully explored by existing works [17, 20, 25, 28]. However, this small-motion assumption does not always hold in practice. Many real-world scenarios, such as recording athletes in motion, vehicles on a highway, or any dynamic scene where temporally dense capture is expensive or infeasible, routinely violate this fundamental assumption.

When motion becomes sufficiently large, dynamic scene interpolation faces a fundamental breakdown. As shown in Figure 2, existing works that rely on the motion local-

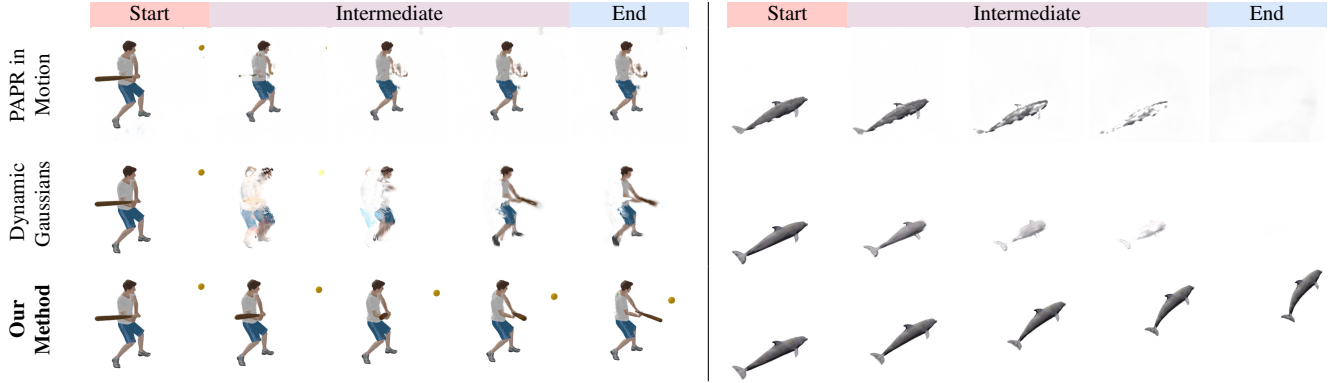


Figure 2. **Novel View Synthesis of Scene Interpolation.** Ball scene (left): A person swings a bat with dynamic body poses as a ball flies toward them. Dolphin scene (right): A dolphin jumps, undergoing large and non-rigid motion. Only two baseline methods are presented, since the other two (4DGS [25] and Deformable 3DGS [28]) fail to produce any reasonable rendering on these two scenes.

ity assumption would fail. This failure stems from the ill-posed nature of point correspondence under a large motion: 1) a point’s local neighborhood becomes unreliable as objects move far from their original positions, and 2) globally, a point may have multiple plausible matches, making the point-to-point correspondence ambiguous. As illustrated in Figure 1, naively matching nearest neighbors in these cases results in criss-cross matches, making it unusable for scene interpolation.

We introduce GMC to address this fundamental limitation by learning smooth global correspondences through transformation to a shared canonical space for each timestep. Our key insight is to replace direct point-to-point matching with unary potential fields that predict  $SE(3)$  transformations for each timestep. These fields take universal features such as DINO [4] as input and leverage the inductive bias of MLPs to address the ill-posed nature of global matching. By construction, this approach ensures that semantically similar points move coherently by predicting coherent transformation.

Our approach seamlessly integrates point-based representations, such as 3D Gaussian Splatting [12] (3DGS) with continuous and queryable  $SE(3)$  fields, enabling robust interpolation and extrapolation under large motion. Specifically, we train two unary potential fields alongside two sets of 3D Gaussians, where each field learns  $SE(3)$  transformations that map its respective Gaussians into a shared canonical space. By defining the GMC over both Gaussian 3D coordinates and semantic features, our  $SE(3)$  fields exhibit smoothness and continuity across both space and semantics. Forward and backward mapping through the canonical space yields dense per-point trajectories for both interpolation and extrapolation.

We evaluate GMC on *interpolation* using eight scenes from the PAPR In Motion dataset [20] and twelve additional challenging scenes with multi-object interactions and large

motions. Results demonstrate that GMC significantly outperforms baseline methods under large motion conditions. Additionally, GMC enables realistic motion *extrapolation*, a capability lacking in existing baselines. We also provide preliminary results showing that our method can not only better reason in the setting of a sparse *temporal* capture but can also improve reconstruction in sparse *spatial* capture.

In summary, our contributions are:

- **Global Motion Corresponder** (GMC), a novel method for 3D scene interpolation under large motion.
- Comprehensive experimental results demonstrating GMC’s superior performance over prior methods for interpolation on challenging large-motion scenarios.
- Further demonstration of GMC’s capabilities in enabling extrapolation and improving reconstruction quality under sparse temporal and spatial capture conditions.

## 2. Related Work

3DGS [12] is a point-based representation that rasterizes small differentiable Gaussians for novel views synthesis of a 3D scene. Unlike volumetric representations like Neural Radiance Fields [18], which require querying multiple points along each ray, 3DGS represents pixels using only a few Gaussians, achieving faster training and real-time rendering.

The challenge of dynamic scene interpolation becomes particularly acute when scenes involve complex and large motions across multiple objects. Traditional point cloud interpolation methods [16, 30, 33, 35] fail under such conditions. Numerous 3DGS-based approaches [3, 6, 9, 14, 15, 17, 25, 26, 28, 37] have emerged to tackle dynamic scene interpolation under the small-motion assumption. However, our problem requires both high-quality scene reconstruction and smooth interpolation, demanding robust performance when this assumption breaks down. Current 3DGS-based approaches fall into two categories: deformation fields and

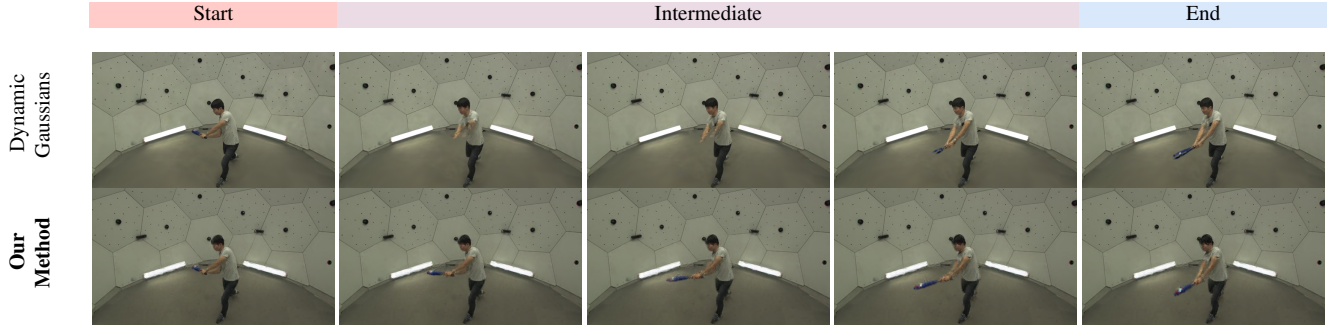


Figure 3. **Real-World Interpolation.** In the Softball scene [17], Dynamic Gaussian [17] fails on large inter-frame motion (note the missing baseball bat). The five columns correspond to five timesteps: 0.00, 0.25, 0.50, 0.75, 1.00.

iterative refinement.

**Scene Interpolation with Deformation Fields.** These methods use neural deformation fields to predict pre-Gaussian displacement at given timesteps, jointly optimizing deformation parameters alongside 3D Gaussian models. Representative works include 4DGS [25] and Deformable 3DGS [28]. However, under large motion, correspondence estimation across timesteps becomes fundamentally ill-posed. Hence, the training of the deformation network is very unstable, resulting in poor reconstruction quality and implausible motion estimates.

**Scene Interpolation by Iterative Refinement.** These approaches achieve interpolation by iteratively refining point primitives from previous timesteps. Under small-motion assumptions, point geometry and semantics do not change drastically between timesteps. These methods pre-train 3D scene representations at initial states, then update the point positions and semantics via rendering loss gradients at subsequent timesteps, storing model weights at each timestep. Interpolation proceeds by blending adjacent timestep weights. Notable examples include Dynamic Gaussian [17] and PAPR In Motion [20]. However, under large motions, simple position fine-tuning from previous states proves insufficient, causing these methods to fail.

**Visual Features for Correspondence.** Features extracted from self-supervised vision transformers [4, 8] serve as powerful visual descriptors that preserve local semantic information. These features excel in downstream tasks such as zero-shot segmentation and semantic correspondence [2]. Zhang et al. [31] demonstrate semantic correspondence using features from large vision models such as DINO [4] and Stable Diffusion [5, 21]. However, while achieving impressive 2D correspondence results, these methods fail to extend to 3D representations or dynamic scenes due to their lack of 3D awareness under complex object motions

and occlusion. To address this issue, Zero-Shot 3D Shape Correspondence [1] proposes a method for establishing correspondence between 3D shapes without explicit supervision. Although innovative, the method is not specifically designed to capture temporal dynamics. In contrast, our method leverages DINO [4] features to learn motion-aware point-based correspondences, uniquely enabling both 3D scene interpolation and extrapolation under large motion.

### 3. Method

In the previous sections, we explained that traditional methods assume spatial positions of 3D points change minimally between adjacent timesteps, which holds only under small, local motion. Since each Gaussian’s position remains nearly constant, it is easy to establish correspondences between points from adjacent timesteps by simply identifying the Euclidean nearest neighbors. Using these correspondences, existing methods either refine point positions from previous timesteps or train neural deformation fields to predict per-point displacements.

However, nearest-neighbor correspondence fails when scenes undergo substantial motion. Under this premise, finding correspondences between adjacent timesteps is not as straightforward. When correspondences rely solely on spatial proximity, severe mismatches can occur (Figure 1), causing existing methods to produce implausible trajectories that violate spatial rigidity and physical constraints (Figure 2). Hence, the core challenge under large motion is *actually* analogous to the challenge of establishing smooth global correspondences between primitives across time.

Although establishing global correspondences remains challenging for current works in literature, it is intuitive for human perception. Humans can effortlessly perceive the geometry and physical properties of dynamic 3D scenes, intuitively predicting intermediate or future motions from just two timesteps. This is because humans have inherent *semantic* knowledge about the physical world and understanding of *local spatial rigidity*. The latter property is

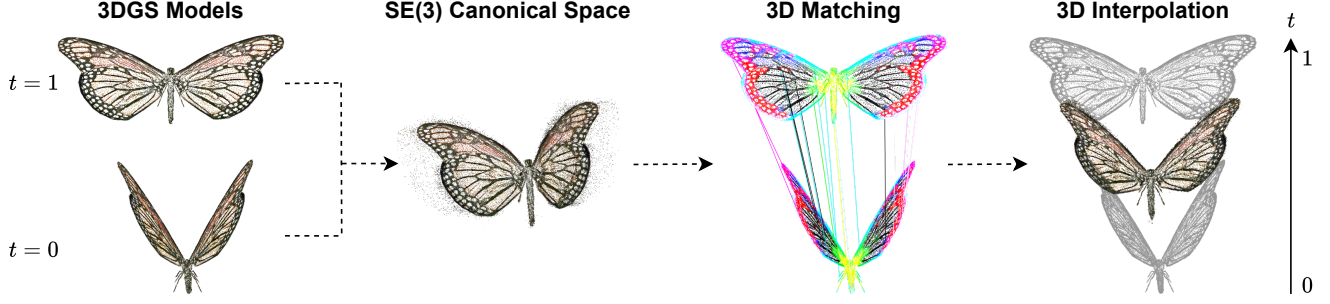


Figure 4. **Method Overview.** (1) Left: 3DGS models at  $t = 0$  and  $t = 1$ . (2) Middle Left: Alignment in a canonical space through SE(3) transformation. (3) Middle Right: 3D matching (colored by PCA-DINO features) is established based on the alignment. (4) Right: Continuous 3D interpolation is derived from the 3D transformation and 3D matching.

important, which means that even though the overall motion may be non-rigid, the movement of nearby object parts move similarly.

### 3.1. Challenges in Learning Correspondence

Since a naive nearest-neighbor search in Euclidean space using point coordinates does not work under the large motion setup, we need to find a way to conduct a global search to obtain point-to-point correspondences. For instance, we can try to define a function that allows us to measure the distance between points of two arbitrary Gaussian  $g_i$  and  $g_j$ . One option is to use a weighted sum of color ( $c$ ), semantic ( $f$ ), and spatial ( $\mu$ ) features to define the following distance function:

$$D_{i,j}^{\text{toy}} = w_c \|c_i - c_j\|_2^2 + w_f \|f_i - f_j\|_2^2 + w_\mu \|\mu_i - \mu_j\|_2^2. \quad (1)$$

It can be reasoned that even this distance function is not enough to establish reliable correspondences. This is because there may be many Gaussian elements within a scene that can have the same color and semantic features. For instance, consider point “a” in column 2 of Figure 1: by the distance function defined in Eq. 1, its closest match in timestep 1 is point “b”, because both of them have the same color and semantic features. In general, it is difficult to find analytical distance metrics that yield the correct correspondences.

Since it is hard to come up with a pre-defined analytical function, we propose to learn these correspondences through a differentiable optimization-based method. Before describing our approach, we first highlight key properties essential for tackling the challenges posed in establishing correspondences and consequently recovering the scene motion under the large-motion assumption:

1. The semantics and color of corresponding points should be in agreement.
2. Semantically similar points in a neighborhood should move coherently.

### 3.2. Global Motion Corresponder

We give a quick overview of our approach to tackle the challenge of finding smooth and accurate global correspondence (Figure 4 and 5). We propose transforming both sets of Gaussians into a learnable shared canonical space where corresponding Gaussians occupy identical spatial locations:

$$\underbrace{R_i^{(0)} \mu_i^{(0)} + t_i^{(0)}}_{\hat{\mu}_i^{(0)}} = \underbrace{R_j^{(1)} \mu_j^{(1)} + t_j^{(1)}}_{\hat{\mu}_j^{(1)}}, \quad (2)$$

where  $R$  and  $t$  represent *learnable* point-wise transformations for each timestep. The transformations  $(R_i^{(0)}, t_i^{(0)})$  and  $(R_j^{(1)}, t_j^{(1)})$  are obtained from our Unary Potential Fields  $\mathcal{F}_0$  and  $\mathcal{F}_1$ , which are parameterized as MLPs. The parameters of these MLPs are optimized using our proposed Energy-based loss. Now, we go into the technical details of the design of our Unary Potential Fields and Energy-based loss.

#### 3.2.1. Unary Potential Field

We define a single Unary Potential Field  $\mathcal{F}$ , as function that maps 3D coordinates  $\mu \in \mathbb{R}^3$  and feature vectors  $\tilde{f} \in \mathbb{R}^4$  to SE(3) transformations. It outputs a quaternion representation  $R \in \mathbb{R}^4$  and translation  $t \in \mathbb{R}^3$ :

$$\mathcal{F}(\tilde{f}, \mu) = (R, t). \quad (3)$$

In practice, we use PCA-projected DINO features for  $\tilde{f}$  to capture semantic information efficiently, and  $R$  is converted to a  $3 \times 3$  rotation matrix when applied in subsequent transformations.

It is a well-known fact that neural networks are able to learn smooth functions such that the output does not change abruptly. Since we wish to keep the predicted motion smooth with respect to both semantic features and spatial positions, we leverage the inductive bias of MLPs for our unary potential. Hence,  $\mathcal{F}$  is parameterized as an MLP.



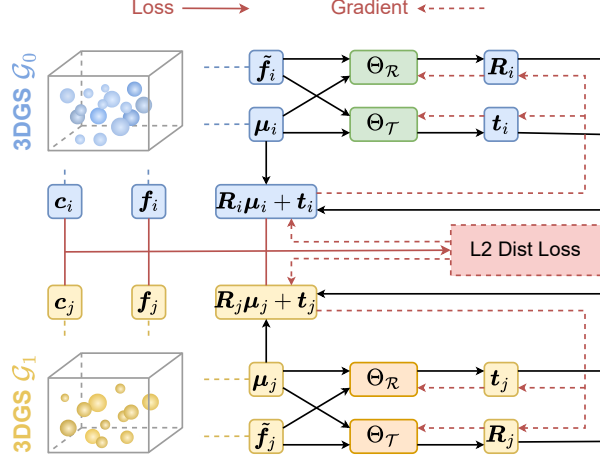


Figure 5. **GMC Learning Illustration.** Each GMC uses two MLPs ( $\Theta_R$  and  $\Theta_T$ ), which input a Gaussian’s mean ( $\mu$ ) and PCA-DINO feature ( $\tilde{f}$ ) and output rotation  $R$  and transformation  $t$ , to calculate the new position  $\hat{\mu} = R\mu + t$  in the shared canonical space. The energy loss is the L2 distance between nearest neighbors in the joint space of color, feature, and  $\hat{\mu}$ , with gradient backpropagated to the MLPs.

Since  $\mathcal{F}$  takes features  $\tilde{f}$  as input, semantically similar points will naturally predict similar transformations by the inductive bias of the MLP.

We term these potential fields “unary” because each timestep requires individual mapping to the canonical space. The same object at different timesteps needs distinct transformations to the shared canonical space. Hence, we need to have separate MLPs per timestep ( $\mathcal{F}_0, \mathcal{F}_1$ ).

### 3.2.2. Energy-based Loss

We now define our energy-based loss used to optimize unary potential fields for global correspondence learning. Concretely, using the means of Gaussians mapped from  $\mathcal{G}_0$  and  $\mathcal{G}_1$  to the canonical space, we have:

$$E_{i,j} = w_c \|c_i - c_j\|_2^2 + w_f \|f_i - f_j\|_2^2 + w_\mu \|\hat{\mu}_i - \hat{\mu}_j\|_2^2, \quad (4)$$

where  $\hat{\mu}_i$  and  $\hat{\mu}_j$  represent the transformed means in the canonical space. Comparing this term with Eq. 1, rather than using spatial distance in the original Euclidean space, we are now using spatial distance in the canonical space.

To train the parameters of  $\mathcal{F}_0$  and  $\mathcal{F}_1$ , we define a bidirectional loss using the aforementioned energy term:

$$\mathcal{L}_E = \sum_{g_i \in \mathcal{G}_0} \min_{g_j \in \mathcal{G}_1} E_{i,j} + \sum_{g_j \in \mathcal{G}_1} \min_{g_i \in \mathcal{G}_0} E_{j,i}. \quad (5)$$

This loss encourages Gaussians that are similar in color and feature space, and that are close in the canonical space, to be

matched, prompting smooth and realistic correspondences. It is important to note two crucial designs:

1. The *bidirectional* loss that operates on both sets of Gaussians  $\mathcal{G}_0, \mathcal{G}_1$  is essential. Optimizing the energy term for just one timestep can lead to a scenario similar to column 2 of Figure 1, where the back of the pencil at timestep 1 does not have any correspondence. The two-way nature of the loss ensures that all points in both timesteps find correspondences.
2. Color  $c$  and features  $f$  terms in Eq. 4 are crucial. Since unary potential fields are randomly initialized, the contribution of  $w_\mu \|\hat{\mu}_i - \hat{\mu}_j\|_2^2$  at the beginning of the training is arbitrary. Hence, the color and feature terms help to establish initial “soft” correspondences.

### 3.2.3. Global Motion

Given learned unary potentials, we can then recover the global motion for Gaussian  $g_i^0$  in  $\mathcal{G}_0$  with position  $\mu_i^0$  by first mapping it to the shared canonical space using its learned unary potential field  $\mathcal{F}_0$ :  $\hat{\mu}_i^{(0)} = R_i^{(0)} \mu_i^{(0)} + t_i^{(0)}$ . Next, we can find its corresponding Gaussian  $g_j$  in  $\mathcal{G}_1$  using the energy term defined in Eq 4 in the shared canonical space. We refer to the mean of  $g_j$  as  $\mu_j^{(1)}$ . Finally, we can see the inverse transformation from its learned unary potential field  $\mathcal{F}_1$  (refer Eq. 2):

$$\begin{aligned} \mu_i^{(0),t=1} &= \left(R_j^{(1)}\right)^{-1} \left(R_i^{(0)} \mu_i^{(0)} + t_i^{(0)} - t_j^{(1)}\right) \\ &= \left(R_j^{(1)}\right)^\top \left(R_i^{(0)} \mu_i^{(0)} + t_i^{(0)} - t_j^{(1)}\right). \end{aligned} \quad (6)$$

The overall global motion for  $g_i^0$  is then the displacement vector  $|\mu_i^{(0),t=1} - \mu_i^{(0)}|$ .

### 3.3. Training Details

**Local Isometry Loss.** To further enforce the smoothness and rigidity, we incorporate a local isometry loss applied to the transformations into the canonical frame, inspired by prior works [17, 20, 23]. For each Gaussian  $g_i$ , we find its  $k$  nearest neighbors  $\text{NN}_i$  in the original space and enforce that distances are preserved after transformation with Local Isometry Loss  $\mathcal{L}_{\text{iso}} =$

$$\frac{1}{kN} \sum_{g_i \in \mathcal{G}_k} \sum_{g_j \in \text{NN}_i} \left| \|\mu_i - \mu_j\|_2^2 - \|\hat{\mu}_i - \hat{\mu}_j\|_2^2 \right|, \quad (7)$$

where  $N$  is the total number of Gaussians. This loss penalizes changes in local geometric relationships, promoting locally rigid transformations. In practice, we apply this loss to the transformation into the canonical space, as well as the full transformation between frames  $t = 0$  and  $t = 1$  (see Sec. 3.4 for this full transformation). The total loss for training each GMC combines the energy and the local isometry terms:

$$\mathcal{L} = \mathcal{L}_E + \alpha \mathcal{L}_{\text{iso}}, \quad (8)$$

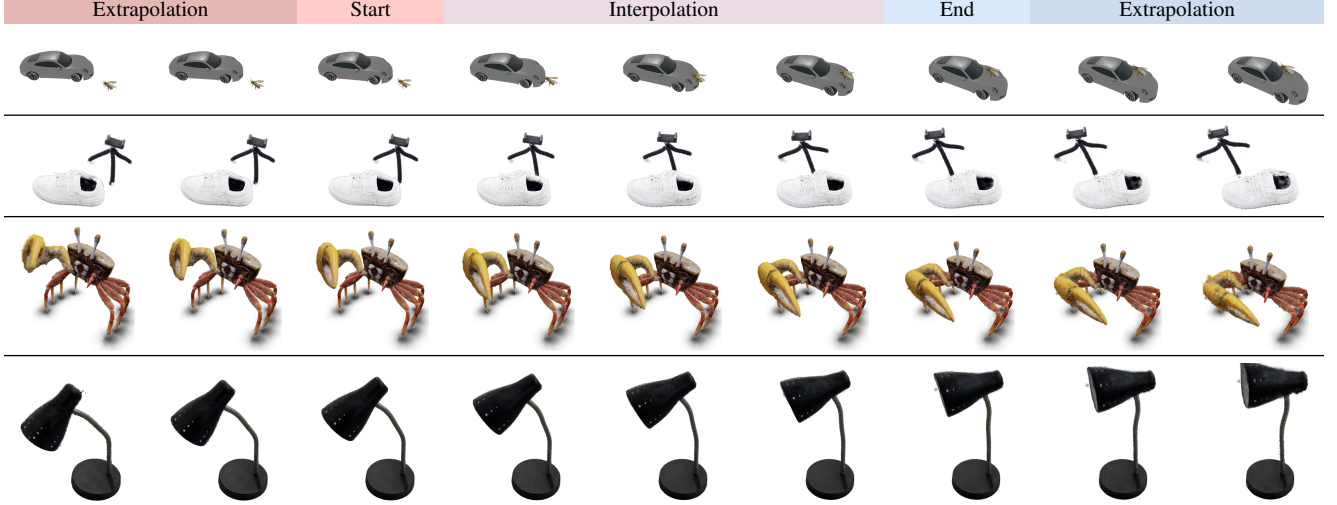


Figure 6. **Interpolation and Extrapolation Results.** (1) Rows 1-2: Multiple objects with global motion (synthetic Car and the real-world Shoe). (2) Rows 3-4: Single objects with local motion [20] (synthetic Crab and real-world Lamp). Our method provides plausible interpolation and extrapolation given multi-view input of two states.

where  $\alpha$  is a weight that balances the two terms. We gradually increase  $\alpha$  during training, starting from zero, to allow the model to first establish correspondences before enforcing rigidity.

**Joint Refinement.** After GMC learning, we perform a joint refinement to update both GMC and the Gaussian sets  $\mathcal{G}_0$  and  $\mathcal{G}_1$  simultaneously. Without loss of generality, at each iteration, we render Gaussians from  $\mathcal{G}_0$  at both  $t = 0$  and  $t = 1$  (where Gaussians are appropriately transformed into  $t = 1$  before rendering). Let  $\hat{\mathbf{I}}_0$  and  $\hat{\mathbf{I}}_1$  represent the set of rendered images at timestep  $t_0$  and  $t_1$  respectively. Similarly, let  $\mathbf{I}_0$  and  $\mathbf{I}_1$  represent the set of ground truth images at timestep  $t_0$  and  $t_1$  respectively. We use the following RGB loss that combines L1 and LPIPS [32]:

$$\mathcal{L}_{\text{render}} = \beta \mathcal{L}_{\text{RGB}}(\mathbf{I}_0, \hat{\mathbf{I}}_0) + \mathcal{L}_{\text{RGB}}(\mathbf{I}_1, \hat{\mathbf{I}}_1), \quad (9)$$

where  $\beta$  is a weight balancing the rendering contributions from each state. By backpropagating this loss, the parameters of both the Gaussian models and the unary potential fields are refined, thereby improving rendering quality and transformation coherence.

**Dropout.** Notice that a trivial solution to Eq 2 occurs when all Gaussians are mapped to the origin in the canonical space, collapsing to a single point to achieve minimal positional difference. This can happen when the potential fields simply make the outputs satisfy  $\mathbf{R}\boldsymbol{\mu} = -\mathbf{t}$ . To avoid this trivial solution, random dropout is applied to the position input  $\boldsymbol{\mu}$  during the training.

### 3.4. Interpolation and Extrapolation

Eq. 6 provides relative rotation  $\mathbf{R}_r = \left(\mathbf{R}_j^{(1)}\right)^\top \mathbf{R}_i^{(0)}$  and relative translation  $\mathbf{t}_r = \left(\mathbf{R}_j^{(1)}\right)^\top (\mathbf{t}_i^0 - \mathbf{t}_j^1)$ . Interpolation between  $t = 0$  and  $t = 1$  proceeds by interpolating between the identity transformation and this relative transformation. Note that our method is flexible with the interpolation strategy, resulting in diverse motion speeds, and we will demonstrate simple linear interpolation. Specifically, for interpolation parameter  $t \in [0, 1]$ , the interpolated rotation  $\mathbf{R}_t$  is computed using spherical linear interpolation (SLERP) [22] between the identity matrix and  $\mathbf{R}_r$ , while the interpolated translation  $\mathbf{t}_t$  employs linear interpolation between zero and  $\mathbf{t}_r$ . For extrapolation, we simply set parameter  $t < 0$  or  $t > 1$ , extending motion beyond observed timesteps.

## 4. Experiments

### 4.1. Dataset

We evaluate our method on both synthetic and real-world scenes, including single-object and multi-object scenes with local or global motion. Our evaluation dataset comprises: (1) the local-motion dataset from PAPR [20] containing six synthetic scenes and two real-world scenes, (2) real-world scenes from Dynamic Gaussian [17] with frame gaps of 5 timesteps to generate large-motion scenarios, (3) eight synthetic scenes created using the Objaverse objects [7], including three single-object scenes with large global motion and pose changes, and five multi-object scenes with global motion exhibiting rigid or non-rigid behavior, and (4) three

| Metric    | Method                | Ball          | Boat          | Butterfly     | Car           | Dolphin       | Knight        | Microwave     | Seagull       | Avg           |
|-----------|-----------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| SI-FID ↓  | 4DGS [25]             | -             | 328.84        | -             | 460.02        | -             | -             | 258.16        | 294.02        | -             |
|           | Deformable 3DGS [28]  | -             | 811.06        | -             | 800.08        | -             | -             | 709.55        | 633.55        | -             |
|           | Dynamic Gaussian [17] | 192.16        | 267.84        | 303.63        | 290.16        | 281.37        | 298.67        | 270.54        | 278.00        | 283.53        |
|           | PAPR in Motion [20]   | 154.02        | 288.67        | 269.90        | 339.79        | 297.43        | 360.81        | 289.02        | 291.50        | 315.75        |
|           | <b>Ours</b>           | <b>109.37</b> | <b>171.32</b> | <b>100.75</b> | <b>170.87</b> | <b>201.01</b> | <b>262.10</b> | <b>210.15</b> | <b>166.94</b> | <b>224.42</b> |
| SI-EMD ↓  | 4DGS [25]             | -             | <b>59.90</b>  | -             | 1404.70       | -             | -             | <b>45.79</b>  | <b>65.63</b>  | -             |
|           | Deformable 3DGS [28]  | -             | 329.22        | -             | 392.66        | -             | -             | 195.4         | 324.78        | -             |
|           | Dynamic Gaussian [17] | 84.19         | 197.36        | 568.30        | 264.01        | 1402.41       | 55.89         | 106.24        | 224.20        | 521.51        |
|           | PAPR in Motion [20]   | 78.18         | 130.42        | 495.24        | 334.69        | 458.82        | 116.72        | 164.60        | 141.64        | 246.71        |
|           | <b>Ours</b>           | <b>62.62</b>  | <b>98.37</b>  | <b>435.32</b> | <b>192.94</b> | <b>321.59</b> | <b>52.96</b>  | <b>73.60</b>  | <b>125.20</b> | <b>149.38</b> |
| SI-MPED ↓ | 4DGS [25]             | -             | 436.14        | -             | 5202.3        | -             | -             | 173.04        | 489.79        | -             |
|           | Deformable 3DGS [28]  | -             | 153.18        | -             | 310.48        | -             | -             | 155.94        | 122.13        | -             |
|           | Dynamic Gaussian [17] | 92.31         | 268.38        | 1036.32       | 223.32        | 2325.01       | 54.96         | 93.53         | 231.85        | 824.50        |
|           | PAPR in Motion [20]   | 42.83         | 67.34         | 285.55        | 88.42         | 242.89        | 67.86         | 32.60         | 57.50         | 114.45        |
|           | <b>Ours</b>           | <b>14.37</b>  | <b>18.96</b>  | <b>45.74</b>  | <b>35.10</b>  | <b>20.13</b>  | <b>15.31</b>  | <b>13.98</b>  | <b>16.95</b>  | <b>16.47</b>  |

Table 1. **Scene Interpolation Evaluation.** This table compares methods on synthetic global-motion scenes (“-” indicates failure). Our method achieves the lowest SI-FID and SI-MPED scores, indicating smoother interpolation of rendered images and geometry. In most scenes, our method also achieves lower SI-EMD scores, demonstrating better overall geometry fidelity.

| Method                | SI-FID ↓      | SI-EMD ↓     | SI-MPED ↓   |
|-----------------------|---------------|--------------|-------------|
| 4DGS [25]             | 150.42        | 15.35        | 156.64      |
| Deformable 3DGS [28]  | 404.46        | 27.35        | 40.38       |
| Dynamic Gaussian [17] | 228.07        | 64.27        | 114.91      |
| PAPR in Motion [20]   | 117.94        | 12.99        | 9.38        |
| <b>Ours</b>           | <b>112.62</b> | <b>12.96</b> | <b>9.27</b> |

Table 2. **Synthetic Local-Motion Evaluation.** Comparison of average metrics on synthetic local-motions scenes [20] against baselines.

manually captured real-world scenes, including one single-object and two multi-object scenes with global motion.

## 4.2. Motion Interpolation

**Implementation Details.** For the 3DGS pre-training, we build upon Feature 3DGS [36] codebase and incorporate gsplat [29] for accelerated rasterization. For each state, we train for 30k iterations. Before the MLPs training, normalization scalars for the DINO, PCA-DINO, RGB, and Gaussian means are pre-calculated from the first 3DGS model. These normalizations are applied to both models during the query and database construction. The weights are set as  $w_c = 1$ ,  $w_f = 10$ ,  $w_\mu = 10$ ,  $k = 256$ ,  $\beta = 1$ , with  $\alpha$  starting at 0 and linearly increasing to 10 over 10k iterations. The four MLPs are trained for 20k iterations, followed by 20k joint refinement iterations with enabled pruning and densification for both 3DGS models. More details can be found in the supplementary material.

**Scene Interpolation Metrics.** When objects undergo large motions, determining the motion trajectories is an

ill-posed problem, as there can be infinitely many paths for an object to transit from one state to another. Since we do not have access to a single “ground truth” trajectory, we cannot define metrics that measure the *correctness* of the predicted trajectory. One way to evaluate the trajectory is by the smoothness of interpolation. PAPR in Motion [20] proposed various such metrics to measure the *smoothness* of the overall interpolation by using only start and end state information. Specifically, rendering quality is evaluated using Scene Interpolation Fréchet inception distance [10] (SI-FID), while geometry quality is assessed using Scene Interpolation Earth Mover’s Distance (SI-EMD). Furthermore, we propose using Scene Interpolation Multiscale Potential Energy Discrepancy [27] (SI-MPED) to measure the local geometry preservation (details in supplementary material). To achieve a low SI-MPED value, a method needs to predict the correct end-state while maintaining good local geometry. Therefore, a high SI-MPED indicates either poor motion learning or loss of local geometry, or both. Note that SI-EMD and SI-MPED results are reported in units of  $10^{-3}$  in this paper, and for all three metrics, lower scores indicate better quality.

**Note about Baselines.** Methods that use deformation fields, namely 4DGS [25] and Deformable 3DGS [28], jointly optimize over the parameters of the deformation field and the Gaussian model. For large motion, this problem is ill-posed, and thus the training of these methods is very unstable. Hence, in tables and figures, we do not report results where the methods fail to train.

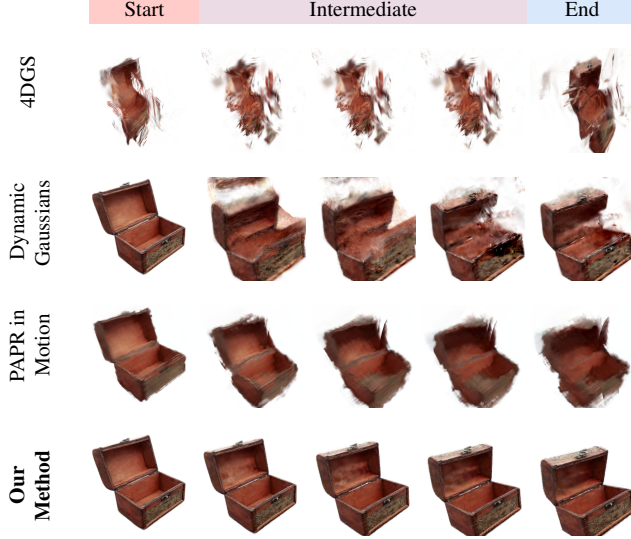


Figure 7. **Real-World Scene Interpolation.** In the real-world `Box` scene, the box undergoes global motion while its lid exhibits local motion. Our method accurately captures both motion types and delivers realistic interpolation.

**Quantitative Results.** Evaluation results of interpolation on synthetic scenes are reported in Table 1 and Table 2, and on real-world scenes in Table 3. For both synthetic and real-world scenes, our method outperforms the baselines by a significant margin, indicating better geometry and rendering quality during interpolation.

**Qualitative Results.** Qualitative results are shown in Figure 2, Figure 7, Figure 3 and Figure 6. In Figure 2, our method is compared with the baseline methods PAPR in Motion [20] and Dynamic Gaussian [17] on novel-view synthesis during interpolation for synthetic scenes. Consider the `Dolphin` scene in Figure 2, where the baselines fail because they update the point positions locally and are not able to correspond to true positions that are far away. Figure 7 shows the results of novel-view synthesis during interpolation of the real-world scene `Box`. While the interpolation given by the baseline methods is suboptimal, our method correctly identifies such complex motion and produces feasible interpolation of the lid while moving the box body. Furthermore, Figure 3 shows that our method is able to produce plausible interpolation in real-world scenes when the bat has large motion between start and end states, while the bat disappears during the interpolation for Dynamic Gaussian [17]. Figure 6 presents qualitative results on both interpolation and extrapolation. Since neither baseline method can perform extrapolation, we focus on our method, demonstrating that it performs well on synthetic and real-world data, single and multiple objects, and both local and global motion.

| Metric    | Method                | Box           | Shoe          | Tapeline     | Avg           |
|-----------|-----------------------|---------------|---------------|--------------|---------------|
| SI-FID ↓  | 4DGS [25]             | 418.6         | 350.09        | -            | -             |
|           | Deformable 3DGS [28]  | -             | -             | -            | -             |
|           | Dynamic Gaussian [17] | 293.20        | 302.67        | 330.08       | 308.65        |
|           | PAPR in Motion [20]   | 371.51        | 339.24        | 425.28       | 378.68        |
|           | <b>Ours</b>           | <b>182.13</b> | <b>168.65</b> | <b>148.2</b> | <b>166.33</b> |
| SI-EMD ↓  | 4DGS [25]             | 67.73         | 57.68         | -            | -             |
|           | Deformable 3DGS [28]  | -             | -             | -            | -             |
|           | Dynamic Gaussian [17] | 368.93        | 524.19        | 697.19       | 530.10        |
|           | PAPR in Motion [20]   | 724.65        | 531.33        | 1577.02      | 944.33        |
|           | <b>Ours</b>           | <b>81.44</b>  | <b>46.43</b>  | <b>84.69</b> | <b>70.85</b>  |
| SI-MPED ↓ | 4DGS [25]             | 748.67        | 345.71        | -            | -             |
|           | Deformable 3DGS [28]  | -             | -             | -            | -             |
|           | Dynamic Gaussian [17] | 279.00        | 627.56        | 290.62       | 399.06        |
|           | PAPR in Motion [20]   | 566.24        | 268.97        | 334.86       | 390.02        |
|           | <b>Ours</b>           | <b>36.27</b>  | <b>20.38</b>  | <b>41.70</b> | <b>32.78</b>  |

Table 3. **Real-World Global-Motion Evaluation.** Comparison of our method with the baselines on real-world scenes with global motion (“-” indicates failure). Overall, our method outperforms the baselines in rendering quality and geometry fidelity.

| Model                  | SI-FID ↓      | SI-EMD ↓      | SI-MPED ↓    |
|------------------------|---------------|---------------|--------------|
| No DINO Input          | 203.38        | 230.04        | 50.35        |
| No Position Input      | 198.72        | 233.68        | 54.29        |
| No Local Isometry Loss | 252.09        | 219.81        | 69.38        |
| No Joint Refinement    | 194.42        | 217.70        | 28.08        |
| <b>Full</b>            | <b>188.98</b> | <b>215.28</b> | <b>26.05</b> |

Table 4. **Ablation Study.** Average metrics on synthetic global-motion scenes for various model variants are reported to reveal the impact of each method component on interpolation performance.

### 4.3. Ablation Study

Each component is removed separately from the full model, and the average resulting metrics are reported in Table 4. Specifically, we study the importance of (1) the DINO input to the Motion Network, (2) the position (Gaussian mean) input to the Motion Network, (3) local distance preservation loss, and (4) the appearance refinement stage. Table 4 shows that the full model achieves the best SI-FID, SI-EMD, and SI-MPED.

## 5. Conclusion

We present Global Motion Corresponder (GMC), a novel method for 3D scene interpolation and extrapolation. We show that predicting large motion between timesteps is analogous to establishing smooth global correspondences between points across time. Next, we present our methods with the desired interpolation properties. Experimental results demonstrate that GMC significantly outperforms prior work in interpolation tasks, especially when dealing with large changes between captures. Moreover, GMC enables extrapolation beyond the captured states, a capability lacking in prior work.



## References

- [1] Ahmed Abdelreheem, Abdelrahman Eldesokey, Maks Ovsjanikov, and Peter Wonka. Zero-shot 3d shape correspondence, 2023. [3](#)
- [2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors, 2022. [3](#)
- [3] Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting, 2024. [2](#)
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. [2](#), [3](#)
- [5] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. In *Advances in Neural Information Processing Systems*, pages 9011–9023. Curran Associates, Inc., 2021. [3](#)
- [6] Devikalyan Das, Christopher Wewer, Raza Yunus, Eddy Ilg, and Jan Eric Lenssen. Neural parametric gaussians for monocular non-rigid object reconstruction, 2024. [2](#)
- [7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects, 2022. [6](#)
- [8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. [3](#)
- [9] Bardienu P. Duisterhof, Zhao Mandi, Yunchao Yao, Jia-Wei Liu, Jenny Seidenschwarz, Mike Zheng Shou, Deva Ramanan, Shuran Song, Stan Birchfield, Bowen Wen, and Jeffrey Ichnowski. Deforms: Scene flow in highly deformable scenes for deformable object manipulation, 2024. [2](#)
- [10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6629–6640, 2017. [7](#)
- [11] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. [1](#)
- [12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. [1](#), [2](#)
- [13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. [1](#)
- [14] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. *arXiv preprint arXiv:2312.16812*, 2023. [2](#)
- [15] 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 (CVPR)*, pages 21136–21145, 2024. [2](#)
- [16] Fan Lu, Guang Chen, Sanqing Qu, Zhijun Li, Yinlong Liu, and Alois Knoll. Pointnet: Point cloud frame interpolation network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. [2](#)
- [17] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [18] 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. [2](#)
- [19] G. Papandreou and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 193–200, Barcelona, Spain, 2011. [1](#)
- [20] Shichong Peng, Yanshu Zhang, and Ke Li. Papr in motion: Seamless point-level 3d scene interpolation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. [3](#)
- [22] Ken Shoemake. Animating rotation with quaternion curves. *SIGGRAPH Comput. Graph.*, 19(3):245–254, 1985. [6](#)
- [23] 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, 2024. [5](#)
- [24] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. [2](#)
- [25] 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 (CVPR)*, pages 20310–20320, 2024. [1](#), [2](#), [3](#), [7](#), [8](#)
- [26] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *Computer Vision – ECCV 2024*, pages 156–173. [2](#)
- [27] Qi Yang, Yujie Zhang, Siheng Chen, Yiling Xu, Jun Sun, and Zhan Ma. Mped: Quantifying point cloud distortion based on multiscale potential energy discrepancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6037–6054, 2023. [7](#), [1](#), [3](#)
- [28] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20331–20341, 2024. [1](#), [2](#), [3](#), [7](#), [8](#)

- [29] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. [7](#)
- [30] Yiming Zeng, Yue Qian, Qijian Zhang, Junhui Hou, Yixuan Yuan, and Ying He. Idea-net: Dynamic 3d point cloud interpolation via deep embedding alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. [2](#)
- [31] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence, 2023. [3](#)
- [32] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. [6](#), [1](#), [2](#)
- [33] Tianyu Zhang, Guocheng Qian, Jin Xie, and Jian Yang. Fast-pci: Motion-structure guided fast point cloud frame interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. [2](#)
- [34] Yanshu Zhang, Shichong Peng, Seyed Alireza Moazenipourasil, and Ke Li. Papr: Proximity attention point rendering. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [1](#)
- [35] Zehan Zheng, Danni Wu, Ruisi Lu, Fan Lu, Guang Chen, and Changjun Jiang. Neuralpci: Spatio-temporal neural field for 3d point cloud multi-frame non-linear interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [2](#)
- [36] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suyu You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. [7](#)
- [37] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21634–21643, 2024. [2](#)