POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

No Pose, No Problem in 4D: Feed-Forward Dynamic Gaussians from Unposed Multi-View Videos

Tesi di Laurea Magistrale in
High Performance Computing Engineering

Matteo Balice, 251648

Abstract: Recent feed-forward 3D Gaussian splatting methods have made dramatic progress on individual aspects of 3D scene reconstruction, but no existing method jointly addresses dynamic content, multi-view input, and unknown camera poses in a single feed-forward pass. Methods that handle dynamics either require accurate camera poses or accept only monocular input; pose-free multi-view methods address only static scenes; and per-scene optimization methods bridge some of these gaps but at minutes-to-hours cost per scene. This thesis introduces **NoPo4D**, the first feed-forward system that addresses this empty quadrant. Building on a pretrained geometry backbone and recent 4D Gaussian frameworks, NoPo4D introduces a velocity decomposition that splits Gaussian motion into per-pixel image-plane shifts and depth changes, allowing direct supervision from pseudo ground-truth optical flow on the 2D component. This sidesteps both the differentiable rendering that couples prior posed methods to pose accuracy and the 3D motion ground truth that prior pose-free methods require. The system is rounded out by a bidirectional motion encoder for cross-view and cross-frame feature aggregation, and view-dependent opacity that mitigates cross-view and cross-timestep Gaussian misalignments. On four multi-view dynamic benchmarks, NoPo4D consistently outperforms prior feed-forward baselines, and with an optional post-optimization stage surpasses per-scene optimization methods, while running orders of magnitude faster.

Advisor:
Prof. Matteo Matteucci

Co-advisors:
Dr. Sunghwan Hong
Prof. Marc Pollefeys

Academic year:
2025–2026

Key-words: 3D Gaussian Splatting, feed-forward reconstruction, dynamic scenes, pose-free reconstruction, optical flow, multi-view video, neural rendering

1. Introduction

The ability to reconstruct a dynamic three-dimensional scene from ordinary video and navigate through it freely from any viewpoint stands among the most compelling goals of computer vision and computer graphics. Such technology would transform how we capture, share, and relive human activity: a basketball game, a live concert, a family gathering, a surgical procedure. Today, achieving compelling free-viewpoint video typically requires a controlled studio environment, expensive calibrated camera arrays, and hours of per-scene processing, a pipeline accessible only to large production houses.

Recent years have seen dramatic progress toward removing these barriers. 3D Gaussian Splatting [24] showed that an explicit Gaussian representation enables real-time rendering of static scenes at high visual quality, while geometry foundation models such as DUSt3R [52], MASt3R [26], Fast3R [60], VGGT [48], MapAnything [23], and π^3 [53] broke the per-scene optimization bottleneck, predicting dense geometry in seconds from unordered image collections. NoPoSplat [67], AnySplat [22], Depth Anything 3 [29], and a growing body of related methods combined feed-forward geometry prediction with Gaussian rendering, achieving instant novel-view synthesis of static scenes without camera calibration.

Extending these advances to dynamic phenomena introduces an entirely new set of challenges: the reconstruction must track motion across time, maintain temporal coherence across synchronized cameras, and do so without reliable ground-truth camera poses, which are unavailable in casual multi-camera setups such as smartphones placed around a table or action cameras worn by athletes. No existing feed-forward method addresses all three challenges simultaneously.

Reconstructing dynamic 3D scenes from a handful of time-synchronized video streams, without known camera poses, without per-scene optimization, and in a single forward pass, demands solving three interlocked problems:

1. **Modeling dynamic content.** Static reconstruction methods predict a single 3D representation from a set of images, whereas dynamic scenes require a 4D representation that captures how the scene evolves over time, together with mechanisms for tracking object motion, encoding temporal relationships, and supervising predicted motion with appropriate training signals.
2. **Fusing information across multiple viewpoints.** A single camera observes the scene from one vantage point and leaves depth and occlusion ambiguities that only additional views can resolve, so jointly processing all cameras requires attention mechanisms that aggregate information across views while preserving per-view spatial structure.
3. **Estimating geometry without known camera poses.** In unposed settings, poses must be inferred jointly with the scene geometry, and estimation errors propagate into all downstream computations, including the projection and unprojection operations that map pixels to 3D space and the optical-flow supervision that grounds motion predictions.

These three problems compound when addressed simultaneously. Motion supervision must operate without ground-truth poses or 3D motion annotations. Multi-view consistency must be enforced despite per-camera pose uncertainty. Pose estimation must be robust to dynamic content that violates the static-scene assumption of most geometry pipelines. Each prior approach has solved at most two of these three problems, leaving a critical gap in the landscape of feed-forward reconstruction, which is illustrated in Table 1.

Static pose-free methods such as NoPoSplat [67], FLARE [70], and AnySplat [22] achieve impressive novel-view synthesis from unposed multi-view images in under a second, but they cannot model moving content. Feed-

Table 1: **Capability comparison of related methods.** NoPo4D is the first feed-forward method that jointly handles dynamic scenes from unposed multi-view video of general environments. **FF**: feed-forward inference. **Unposed**: no ground-truth camera poses required. **MV**: supports multi-view input. **Dyn.**: models dynamic scenes. **General**: not restricted to a specific domain.

Method	Type	FF	Unposed	MV	Dyn.	General
NoPoSplat [67]	Static	✓	✓	✓	✗	✓
FLARE [70]	Static	✓	✓	✓	✗	✓
AnySplat [22]	Static	✓	✓	✓	✗	✓
4DGT [59]	Dyn.	✓	✗	✗	✓	✓
NeoVerse [63]	Dyn.	✓	✓	✗	✓	✓
DGGT [5]	Dyn.	✓	✓	✓	✓	✗
Shape of Motion [49]	Opt.	✗	✗	✗	✓	✓
MonoFusion [56]	Opt.	✗	✗	✓	✓	✓
NoPo4D (Ours)	**Dyn.**	✓	✓	✓	✓	✓

forward dynamic methods such as 4DGT [59] model temporally coherent Gaussians with velocity attributes but require known camera poses and single-camera input. NeoVerse [63] and MoVieS [28] remove the pose requirement but supervise motion against 3D ground-truth scene flow that is unavailable in real captures, and both remain restricted to a single camera. DGGT [5] operates in the unposed multi-view dynamic regime but is trained and tested exclusively on driving scenarios. Per-scene optimization methods such as Shape of Motion [49] and MonoFusion [56] produce high-quality reconstructions but require known calibration and take minutes to hours per scene.

This thesis presents **NoPo4D**[1], the first feed-forward system that simultaneously satisfies all four desiderata: feed-forward inference, no camera pose requirement, multi-view input, and dynamic scene modeling in general environments. The system builds on the DA3 backbone [29] and extends the 4D Gaussian framework of [59, 63] to the pose-free multi-view dynamic setting through three design choices. The central contribution is a *velocity decomposition* that splits each Gaussian's motion into per-pixel image-plane shifts and a depth change. Because the image-plane component directly encodes the Gaussian's 2D projection without any rendering step, it can be supervised against pseudo-ground-truth optical flow from an off-the-shelf estimator [54], bypassing both the rendering dependence of posed methods and the 3D motion annotation requirement of pose-free methods. This is complemented by a *bidirectional motion encoder* that aggregates features jointly across all cameras and timesteps, producing motion predictions that are globally consistent rather than estimated per camera, and by *view-dependent opacity* implemented with spherical harmonic coefficients that allows the network to suppress geometrically inconsistent Gaussians from viewpoints where pose estimation errors are worst.

This thesis makes the following contributions:

1. This thesis identifies the previously unaddressed setting of feed-forward, pose-free, multi-view dynamic scene reconstruction in general environments and demonstrates via systematic comparison (Table 1) that no prior method addresses it.
2. This thesis introduces **NoPo4D**, the first feed-forward system for this setting. The key technical contribution is a velocity decomposition that splits Gaussian motion into per-pixel image-plane shifts and depth changes, enabling optical flow to supervise the image-plane component directly without rendering or 3D motion annotations. This is complemented by a bidirectional motion encoder that aggregates features across views and frames, and view-dependent opacity that handles cross-view and cross-timestep Gaussian misalignments.
3. A comprehensive experimental evaluation on four multi-view dynamic benchmarks demonstrates that NoPo4D consistently outperforms feed-forward baselines and, with optional post-optimization, surpasses per-scene optimization methods while running orders of magnitude faster. Ablation studies validate the contribution of each design choice.

[1]Project page: `https://bralani.github.io/nopo4d_html/`

2. Related Work

This chapter surveys the work most relevant to NoPo4D and introduces the technical foundations on which it is built. The discussion is organised around three themes. The first is Gaussian Splatting, tracing the evolution from static 3D Gaussian Splatting through dynamic 4D extensions and optimization-based methods, culminating in the NeoVerse parameterization that NoPo4D directly builds upon. The second is optical flow estimation, which provides the supervisory signal at the heart of the velocity decomposition and whose evolution from classical variational methods to modern feed-forward networks is reviewed in some depth. The third is feed-forward geometry estimation, covering the camera geometry and vision backbone NoPo4D builds upon, together with the landscape of static and dynamic feed-forward reconstruction methods.

2.1. Gaussian Splatting

Gaussian Splatting is an explicit scene representation that models a 3D scene as a collection of anisotropic Gaussian primitives, each carrying geometric and appearance attributes. Unlike implicit representations such as Neural Radiance Fields [35], which encode scene content in the weights of a neural network and require expensive ray-marching to render, Gaussian Splatting uses a differentiable tile-based rasterizer that projects and composites the Gaussian footprints directly onto the image plane, enabling fast rendering while remaining fully differentiable for gradient-based optimization.
This combination of rendering speed, explicit structure, and differentiability has made Gaussian Splatting a versatile foundation for a wide range of applications beyond novel-view synthesis. In robotics and embodied AI, it provides compact, updatable scene representations for navigation, manipulation, and sim-to-real transfer, where the ability to add, remove, or modify individual Gaussians in response to new observations makes it more amenable to online map updating than implicit representations that encode the scene globally in network weights. In content creation and visual effects, direct access to Gaussian attributes enables instant relighting by replacing view-dependent color with physically-based material properties, scene editing by selecting and transforming Gaussian clusters, and object insertion by compositing independently rendered Gaussian fields. In medical imaging and scientific visualization, the semi-transparent volumetric nature of the representation has shown promise for rendering complex anatomical structures and simulation data at interactive rates. In the context of this thesis, Gaussian Splatting is the core scene representation on which NoPo4D is built, as its explicit and differentiable nature makes it well-suited for feed-forward dynamic scene reconstruction.

2.1.1 Static 3D Gaussian Splatting

3D Gaussian Splatting [24] represents a scene as a finite collection of G anisotropic three-dimensional Gaussians, each defined by the tuple:

$$(\boldsymbol{\mu}_g,\ \boldsymbol{\Sigma}_g,\ \alpha_g,\ \mathbf{c}_g)_{g=1}^{G}\,. \tag{1}$$

The mean $\boldsymbol{\mu}_g \in \mathbb{R}^3$ encodes the Gaussian's center in world space, $\boldsymbol{\Sigma}_g \in \mathbb{R}^{3\times3}$ is the covariance matrix controlling the shape and orientation of the ellipsoid, $\alpha_g \in [0,1]$ is the opacity, and $\mathbf{c}_g$ is the view-dependent color encoded as spherical harmonic coefficients.
The covariance is factorized as $\boldsymbol{\Sigma}_g = \mathbf{R}_g\mathbf{S}_g\mathbf{S}_g^T\mathbf{R}_g^T$, where $\mathbf{R}_g \in SO(3)$ controls the orientation of the ellipsoid and $\mathbf{S}_g = \mathrm{diag}(s_{g,1}, s_{g,2}, s_{g,3})$ with $s_{g,i} > 0$ encodes the extent along each principal axis, giving the full matrix form:

$$\boldsymbol{\Sigma}_g = \mathbf{R}_g \begin{pmatrix} s_{g,1}^2 & 0 & 0 \\ 0 & s_{g,2}^2 & 0 \\ 0 & 0 & s_{g,3}^2 \end{pmatrix} \mathbf{R}_g^T. \tag{2}$$

This factorization guarantees positive semi-definiteness of $\boldsymbol{\Sigma}_g$ throughout optimization without requiring projected gradient steps. The full pipeline is illustrated in Figure 1. Rendering proceeds by projecting each Gaussian onto the image plane via the local affine approximation $\boldsymbol{\Sigma}_g^{2D} = \mathbf{J}_g\mathbf{W}_g\boldsymbol{\Sigma}_g\mathbf{W}_g^T\mathbf{J}_g^T$, where $\mathbf{W}_g$ is the camera rotation and $\mathbf{J}_g$ is the Jacobian of the projective transformation at $\boldsymbol{\mu}_g$. The image is partitioned into tiles; all Gaussians whose footprint overlaps a tile are sorted by depth and alpha-composited front-to-back:

$$\hat{\mathbf{C}} = \sum_{i=1}^{N} \mathbf{c}_i\,\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j). \tag{3}$$

This tile-based rasterizer is highly parallelizable and enables real-time rendering at hundreds of frames per second, a sharp contrast with the ray-marching of Neural Radiance Fields [35] and its many successors. Optimization is driven by photometric losses against training images and an adaptive densification strategy that

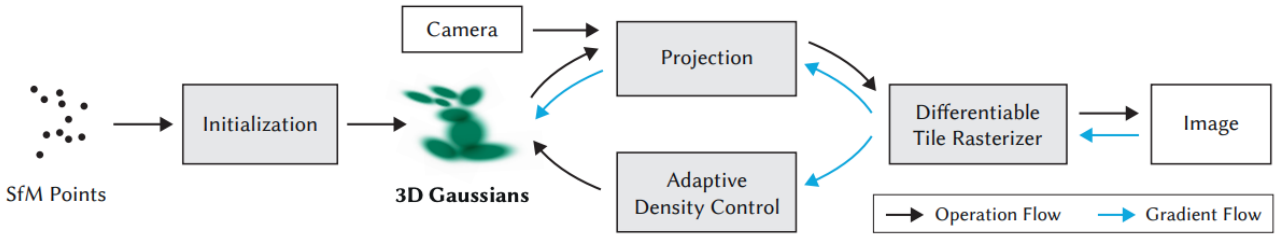


Figure 1: Overview of the 3D Gaussian Splatting pipeline. Figure taken from [24].

grows Gaussians in under-reconstructed regions and prunes redundant ones, converging in 30–60 minutes from a sparse Structure-from-Motion [39] point cloud.

The point cloud initialization is not merely a convenience but a practical necessity. Optimizing Gaussian means directly from random positions is extremely difficult: the photometric loss is nearly flat with respect to mean positions when a Gaussian is far from its correct 3D location, because other Gaussians already covering the same image region absorb the gradient signal. In contrast, color and opacity parameters produce well-conditioned gradients regardless of position, since they directly scale the pixel contribution of whatever region the Gaussian currently projects onto. Means must therefore start close enough to the true surface for the loss landscape to be informative, a condition that random initialization almost never satisfies. The SfM point cloud provides a coarse geometric scaffold that satisfies this condition, allowing the optimization to focus on fine-tuning shape, scale, and appearance rather than searching the full 3D space from scratch. More recently, geometry foundation models [23, 26, 29, 48, 52, 53, 60] have emerged as an alternative source of initialization, predicting dense depth and camera parameters in a single forward pass without requiring feature matching or bundle adjustment. Their output point clouds can replace the SfM initialization and still feed directly into the full per-scene 3DGS optimization, making them broadly applicable beyond feed-forward pipelines.

Several properties make 3DGS attractive as a foundation for feed-forward reconstruction: rasterization is orders of magnitude faster than integrating along rays, the explicit parameters are directly differentiable, and Gaussian means can be initialized by unprojecting pixels from a predicted depth map, making the representation compatible with feed-forward depth estimators, which is the property NoPo4D exploits.

2.1.2 Dynamic 4D Gaussian Splatting

Static 3DGS cannot represent scenes in which objects move, prompting several families of dynamic extension within the Gaussian Splatting paradigm.

Within the 3DGS paradigm, Deformable 3DGS [65] and 4D-GS [57] attach a time-conditioned deformation MLP to a canonical Gaussian set, displacing each mean to its time-varying location. Dynamic 3DGS [34] abandons the canonical representation entirely and instead stores per-timestep Gaussian parameters with rigidity constraints that encourage nearby Gaussians to move coherently, enabling explicit long-term tracking. SC-GS [18] reduces the degree of freedom of this approach by introducing a sparse set of control points whose interpolated deformation field drives the full Gaussian set, decoupling the motion representation from the number of Gaussians and enabling smoother, more physically plausible deformations. Dynamic Gaussian Marbles [42] learns long-range point trajectories that persist across the full sequence, explicitly coupling Gaussians to trackable physical points and enabling temporal consistency over hundreds of frames. MoSca [25] takes a complementary approach: it lifts monocular video to a *4D Motion Scaffold*, a compact smooth representation that encodes scene-wide deformations, anchors Gaussians onto this scaffold, and solves camera poses via bundle adjustment without external pose estimation tools, making it one of the few optimization-based methods that does not require known camera calibration as input. A separate family parameterizes Gaussians directly in (x, y, z, t) space, producing native 4D primitives with temporal covariances that govern how long each Gaussian contributes to the rendered video [10, 64]. Shape of Motion [49] represents per-Gaussian trajectories with compact SE(3) motion bases shared across the scene, enabling globally consistent long-range motion with a small number of parameters. MonoFusion [56] builds on this idea by first reconstructing each camera view independently using monocular depth priors [61, 62] and then aligning the per-view Gaussian fields into a shared 4D representation; it represents the current per-scene optimization state of the art and serves as the strongest baseline. All of these methods are optimization-based: they require known camera calibration and take minutes to hours per scene.

The parameterization NoPo4D inherits was introduced by NeoVerse [63], which extends each Gaussian to a full 4D tuple:

$$\left(\boldsymbol{\mu}_g,\ \boldsymbol{\Sigma}_g,\ \alpha_g,\ \mathbf{c}_g,\ \tau_g,\ l_g,\ \mathbf{v}_g^{+},\ \mathbf{v}_g^{-},\ \boldsymbol{\omega}_g^{+},\ \boldsymbol{\omega}_g^{-}\right)_{g=1}^{G}. \tag{4}$$

The static attributes $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \alpha_g, \mathbf{c}_g)$ are identical to those of static 3DGS described above. The dynamic

attributes introduce temporal structure: τ_g is the temporal center anchoring the Gaussian in time, l_g is a lifespan controlling its active temporal window and encoding the appearance and disappearance of transient objects, and $\mathbf{v}_g^{\pm}, \boldsymbol{\omega}_g^{\pm} \in \mathbb{R}^3$ are asymmetric forward and backward linear and angular velocities. Letting $\Delta t = |t - \tau_g|$, the position, rotation, and opacity of Gaussian g at time t are:

$$\boldsymbol{\mu}_g(t) = \boldsymbol{\mu}_g + \mathbf{v}_g^{\pm} \cdot \Delta t, \tag{5}$$

$$\mathbf{R}_g(t) = \mathbf{R}_g \cdot \mathbf{R}\big(\boldsymbol{\omega}_g^{\pm} \cdot \Delta t\big), \tag{6}$$

$$\alpha_g(t) = \alpha_g \cdot \exp\left(-\frac{\Delta t^2}{2\,\sigma_g^2}\right), \tag{7}$$

where $\mathbf{v}_g^{\pm}$ and $\boldsymbol{\omega}_g^{\pm}$ select the forward ($t > \tau_g$) or backward ($t \leq \tau_g$) velocities, $\mathbf{R}(\cdot)$ converts an axis-angle vector to a rotation matrix, and $\sigma_g^2 > 0$ is a per-Gaussian temporal variance that modulates opacity as a function of distance from the temporal center, encoding the appearance and disappearance of transient objects. The color $\mathbf{c}_g$ is time-invariant and depends only on viewing direction. The asymmetric velocity parameterization allows independent modeling of approach and departure, which is important when a scene point changes direction abruptly. NeoVerse supervises the velocity attributes directly against ground-truth 3D scene flow, which is available only in synthetic datasets, limiting its applicability to real captures. NeoVerse additionally operates on a single camera. NoPo4D adopts this parameterization but removes both constraints through the velocity decomposition described in Section 3.2.3.

2.2. Optical Flow Estimation

Optical flow is the dense 2D displacement field that maps each pixel in frame I_t to the location of the same scene point in frame I_{t+1}. Formally, for a pixel at position (x, y), the flow vector $(\Delta x, \Delta y)$ satisfies $I_t(x, y) \approx I_{t+1}(x + \Delta x, y + \Delta y)$ under the brightness constancy assumption: a surface patch retains its appearance as it moves across the image plane. This assumption is violated at occlusion boundaries, under illumination changes, and for highly specular surfaces, making optical flow estimation an inherently ill-posed inverse problem. A complementary ambiguity, known as the aperture problem, arises because local image measurements constrain only the component of motion perpendicular to the local edge orientation; full 2D velocity is underdetermined from a single local patch. Both classical and deep learning approaches resolve these ambiguities through regularization or learned priors.

The Lucas–Kanade method [33] addresses the aperture problem by assuming that flow is locally constant within a small window around each pixel and solving the resulting over-determined linear system in a least-squares sense. This local assumption breaks down near motion boundaries, where the flow field is discontinuous, but the method is highly efficient and remains the basis of many feature-tracking pipelines. Horn and Schunck [17] took the complementary global view, minimizing an energy functional that combines the brightness constancy residual with a smoothness penalty on the spatial gradient of the flow field. Their formulation produces dense, globally smooth flow but tends to blur motion discontinuities. Decades of variational methods refined these foundations by incorporating robust data terms, occlusion reasoning, and structure-adaptive regularization, yielding increasingly accurate results on classical benchmarks such as Middlebury [1] and KITTI [12].

The introduction of large synthetic datasets and end-to-end learnable architectures transformed the field. FlowNet [9] was the first convolutional network trained to regress optical flow from an image pair, demonstrating that a deep encoder-decoder architecture trained on the synthetic FlyingChairs dataset transfers to real scenes. FlowNet 2.0 [21] stacked multiple FlowNet modules in a cascade, progressively refining the estimate and substantially closing the gap to classical methods on standard benchmarks. PWC-Net [43] introduced the now-canonical combination of spatial pyramid processing, learned feature warping, and a cost volume that explicitly encodes feature-level correspondences between the two frames; this architecture proved compact, accurate, and fast, and established cost volumes as the central inductive bias for deep flow estimation.

RAFT [44] represented the next major architectural advance. Rather than predicting flow directly from a cost volume at a fixed resolution, RAFT constructs an all-pairs 4D correlation volume between every pair of feature locations in the two frames and then applies a recurrent update operator that iteratively refines a flow estimate by querying this volume at changing lookup locations. This recurrent refinement scheme achieves large improvements on standard benchmarks and generalizes substantially better to unseen data than one-shot predictors. GMFlow [58] reframed flow estimation as global feature matching: by applying a Transformer to the concatenated feature maps of both frames, each pixel can attend to every location in the second frame simultaneously, and soft correspondences are read off directly from the attention weights without any iterative refinement. This global receptive field eliminates the local-to-global limitation of cost volumes and handles large displacements more robustly, though it is less accurate than recurrent methods on fine-scale motion. FlowFormer [19] combined the strengths of both paradigms: it compresses the 4D correlation volume into a compact set of latent cost tokens via a Transformer encoder, then decodes flow estimates through cross-attention

Figure 2: Optical flow predictions from SEA-RAFT [54] on Sintel [3] (test), KITTI2015 [12] (test), and Middlebury [1] (train). Top row: input frames. Bottom row: predicted flow visualized using the standard HSV color encoding, where color encodes direction and brightness encodes magnitude. Figure taken from [54].

between query locations and these tokens, followed by RAFT-style recurrent refinement. This two-stage design achieves state-of-the-art accuracy on Sintel [3] and KITTI [12] while retaining the global context that pure cost-volume methods lack.

SEA-RAFT [54], used in NoPo4D to compute pseudo-ground-truth flow labels, improves upon the RAFT architecture with a lightweight iterative refinement scheme and a simple energy-based training objective that produces more accurate and consistent estimates, particularly in textureless and low-contrast regions where brightness constancy is weakest. The quality of these pseudo-labels directly conditions the quality of the velocity decomposition supervision, and improvements to the flow estimator propagate automatically into NoPo4D without architectural changes.

While the methods above operate on isolated frame pairs, multi-frame optical flow approaches exploit temporal context from neighboring frames to resolve correspondences that are ambiguous or occluded in a single pair. VideoFlow [40] demonstrated that propagating correlation volumes across multiple past and future frames substantially improves accuracy at occlusion boundaries, where the correct match is visible only in a neighboring frame; its multi-frame recurrence closes a persistent gap on long-range displacement benchmarks with minimal additional latency. This temporal aggregation principle motivates the bidirectional motion encoder of NoPo4D, which similarly aggregates features across multiple consecutive frames and views to produce robust velocity estimates in challenging dynamic regions.

Scene flow [46] is the three-dimensional analogue of optical flow: it assigns a 3D displacement vector to every point in the scene, representing true physical motion rather than its 2D projection onto the image plane. Recovering scene flow from a monocular video requires estimating both depth and 2D flow and reasoning about their geometric relationship, making it substantially harder than either task in isolation. FlowNet3D [31] was the first end-to-end network for point-cloud scene flow, using a flow-embedding layer that aggregates local motion cues across point neighborhoods; while accurate on depth-sensor data, it requires RGB-D or LiDAR input and cannot operate on ordinary video footage. Self-supervised methods have since attempted to estimate scene flow from monocular video by jointly optimizing depth, camera ego-motion, and residual object motion under photometric consistency constraints, but these estimates compound each other's errors: inaccurate depth produces incorrect 3D displacements, and imprecise ego-motion contaminates the residual dynamic component. Dense scene flow from video-only input thus remains unreliable, particularly under fast motion, occlusion, and low-texture regions. The connection between 3D Gaussian velocities and 2D optical flow is the key insight that NoPo4D exploits: by decomposing Gaussian motion into an image-plane component and a depth change, the image-plane component can be matched directly to 2D optical flow without requiring 3D scene flow annotations or differentiable rendering, as described in detail in Section 3.2.3.

2.3. Feed-Forward Geometry and Reconstruction

Classical 3D reconstruction pipelines chain a sequence of independently optimized stages: sparse feature extraction and matching, triangulation, and bundle adjustment that refines cameras and structure jointly. While accurate, these pipelines require minutes to hours of per-scene computation, demand sufficient visual overlap between views, and degrade on textureless or reflective surfaces. The feed-forward paradigm shifts this cost to training time: a network trained once on large and diverse scene collections learns to infer geometry directly from pixels, generalizing across scenes without any per-scene optimization at inference.

The Vision Transformer (ViT) [8] became the backbone of choice for feed-forward geometry tasks: operating on

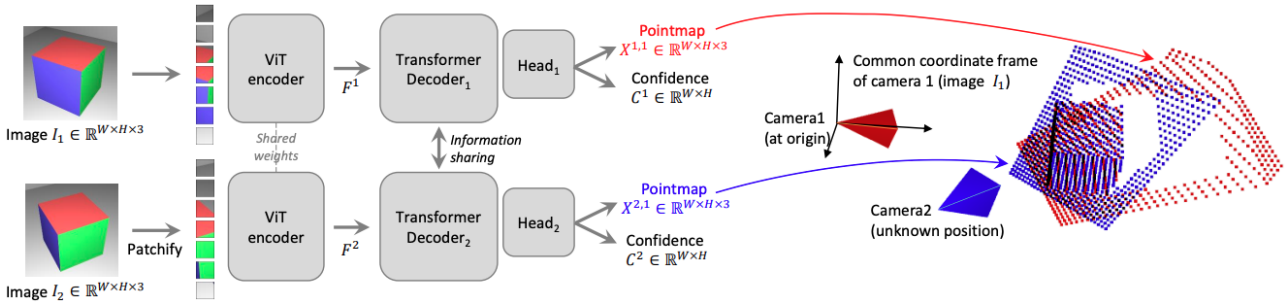


Figure 3: DUSt3R [52] pipeline: paired images are encoded by a shared ViT and decoded into dense pointmaps and confidence maps expressed in camera 1's frame.

a sequence of non-overlapping image patches with full self-attention from the first layer, it captures long-range spatial dependencies that convolutional networks model only at deeper, lower-resolution layers. DINOv2 [36] distills this into a self-supervised backbone whose features are simultaneously spatially localized and semantically rich, transferring effectively to dense prediction tasks without any task-specific pretraining. Decoding dense geometry from ViT tokens requires recovering spatial resolution from the flattened token sequence; the Dense Prediction Transformer (DPT) [37] achieves this by reassembling intermediate ViT features into multi-scale spatial grids and fusing them through convolutional blocks, yielding full-resolution output maps. Building on this architecture, Depth Anything [62] trained a ViT-DPT stack through large-scale semi-supervised distillation on tens of millions of images, producing a monocular depth prior that transfers reliably across diverse environments. Depth Anything V2 [61] refined it further with synthetic-to-real curriculum training, improving accuracy on thin structures and transparent surfaces. These monocular models produce geometrically coherent depth per image, but since each view is processed independently they cannot enforce cross-view consistency or recover metric scale without explicit camera calibration. Given a predicted depth map and camera intrinsics $\mathbf{K} = \mathrm{diag}(f_x, f_y, 1)$ with principal point offset, each pixel (u, v) with depth d can be lifted to a 3D point:

$$\mathbf{X} = d\,\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \tag{8}$$

giving a dense point cloud for each image independently. Cross-view consistency requires a shared coordinate frame, which monocular models lack.

DUSt3R [52] established the feed-forward 3D reconstruction paradigm by training a transformer to directly regress dense 3D point maps from image pairs, where each entry assigns a world-space 3D coordinate to the corresponding pixel. This replaces the entire classical reconstruction pipeline of feature matching, triangulation, and bundle adjustment with a single forward pass that generalizes to unseen scenes without any per-scene optimization. Both pointmaps are expressed in the first camera's coordinate frame; for collections of more than two images, pairwise predictions are aligned into a global coordinate system by minimizing a weighted sum of squared pointmap distances across overlapping pairs, from which camera extrinsics can be recovered without any explicit camera parameterization in the network itself. This implicit treatment of camera geometry makes DUSt3R robust to completely uncalibrated inputs, a property that NoPo4D inherits via its Depth Anything 3 backbone.

MASt3R [26] extended DUSt3R by adding a dense matching head that improves correspondence quality for pairs with large viewpoint changes. Subsequent work scaled the paradigm further: Fast3R [60] introduced a parallel inference scheme for hundreds of simultaneous views.

VGGT [48] (Visual Geometry Grounded deep structured Transformers) broadens the scope of feed-forward geometry estimation by jointly predicting camera parameters, depth maps, dense 3D point maps, and 2D feature tracks from an arbitrary collection of images in a single forward pass. The architecture, illustrated in Figure 4, processes each input image with a frozen DINOv2 backbone to produce patch-level visual features. These per-image features are then concatenated and augmented with a learnable *camera token* appended to each frame's token sequence, which aggregates the geometric information needed to predict intrinsics and extrinsics. The combined token sequence is processed by a transformer that alternates between two attention regimes repeated L times: global attention layers in which all tokens across all frames attend jointly, enabling cross-view correspondence and triangulation-like reasoning, and frame attention layers in which tokens within each frame attend only to one another, refining local spatial structure. After this alternating attention stack, the camera tokens are decoded by a lightweight camera head with a small multi-layer perceptron, while the spatial tokens are decoded by a DPT head [37] into depth maps, world-space point maps, and feature tracks. This unified design avoids the cascade of independently optimized modules present in classical pipelines and replaces

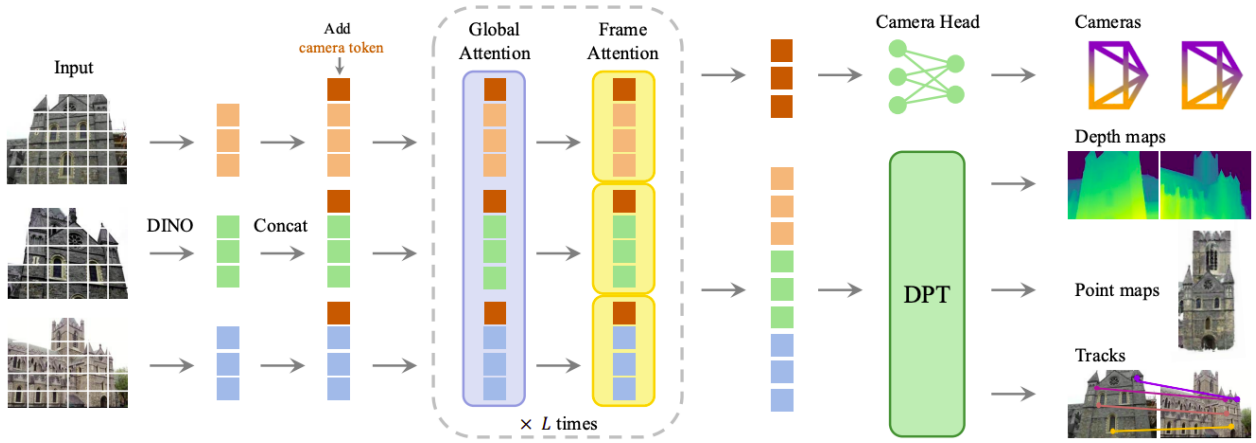


Figure 4: VGGT [48] pipeline. Each input image is encoded by a shared DINOv2 backbone and augmented with a camera token. A transformer with alternating global attention (across all frames) and frame attention (within each frame) is applied L times. The camera tokens are decoded by a camera head to predict camera parameters, while the spatial tokens are decoded by a DPT head into depth maps, point maps, and feature tracks. Figure taken from [48].

them with a single differentiable network. Compared to DUSt3R-family models, VGGT adds explicit camera prediction and feature tracks as native outputs, removing the separate global alignment step needed to recover extrinsics from pairwise point maps. NoPo4D uses VGGT-style alternating attention indirectly through its Depth Anything 3 backbone, which adopts the same within-view / cross-view attention alternation to achieve multi-view-consistent depth and camera prediction.

MoSt3R [68] adapts DUSt3R to dynamic video by estimating per-timestep point maps for each frame; the backbone processes frame pairs independently and camera parameters are recovered from the pairwise outputs without requiring poses as input, making it the first pose-free feed-forward geometry method for dynamic scenes. Its output is a time-varying point cloud rather than renderable Gaussian primitives, so novel-view synthesis still requires a subsequent rendering step. D2USt3R [15] takes a complementary approach, introducing Static-Dynamic Aligned Pointmaps with dedicated prediction heads and loss functions for static and dynamic regions separately; optical flow-based alignment losses distinguish camera-induced from object-induced motion, substantially reducing point-map error on moving objects compared to MoSt3R, though unlike MoSt3R it requires known camera poses to compute the expected ego-motion flow. Spann3R [47] extended the paradigm to online processing by managing an external spatial memory that accumulates 3D information across frames and predicts per-image point maps directly in a global coordinate system without any optimization step, enabling real-time reconstruction from ordered image streams. CUT3R [51] pushed this further with a stateful recurrent model that continuously updates a persistent scene representation as new observations arrive, handling both static and dynamic content in a unified online framework. Depth Anything 3 [29] demonstrated that a plain DINOv2 backbone with alternating within-view self-attention and cross-view attention layers achieves multi-view-consistent depth and camera prediction with fewer parameters than specialized architectures, establishing that cross-view attention is the key ingredient for consistent multi-camera geometry rather than task-specific architectural changes. Figure 5 illustrates the DA3 architecture.

A parallel thread in feed-forward reconstruction predicts Gaussian primitives rather than point maps, enabling immediate rendering. PixelSplat [4] introduced the first feed-forward model for Gaussian-based novel view synthesis from stereo pairs, predicting a dense probability distribution over 3D space and sampling Gaussian means via a differentiable reparameterization trick. Its core building block is an epipolar transformer that cross-attends features from one view onto the epipolar lines of the other, making known camera intrinsics and extrinsics a hard requirement: a pose error directly corrupts the feature matching that drives depth estimation. MVSplat [6] scales feed-forward Gaussian prediction to arbitrary multi-view inputs by replacing epipolar attention with a plane-sweeping cost volume, achieving ten times fewer parameters and twice the inference speed of PixelSplat while retaining the same pose requirement. NoPoSplat [67] removed this requirement by fine-tuning a MASt3R-style backbone to directly regress per-pixel Gaussian attributes in the coordinate frame of the first camera, achieving sub-second novel-view synthesis without any calibration. FLARE [70] follows a cascaded design: it first estimates camera poses from a geometry backbone and then conditions a separate Gaussian head on those poses, separating the geometric and appearance prediction stages for improved stability. AnySplat [22] generalizes to unconstrained collections of any size by replacing pairwise matching with a

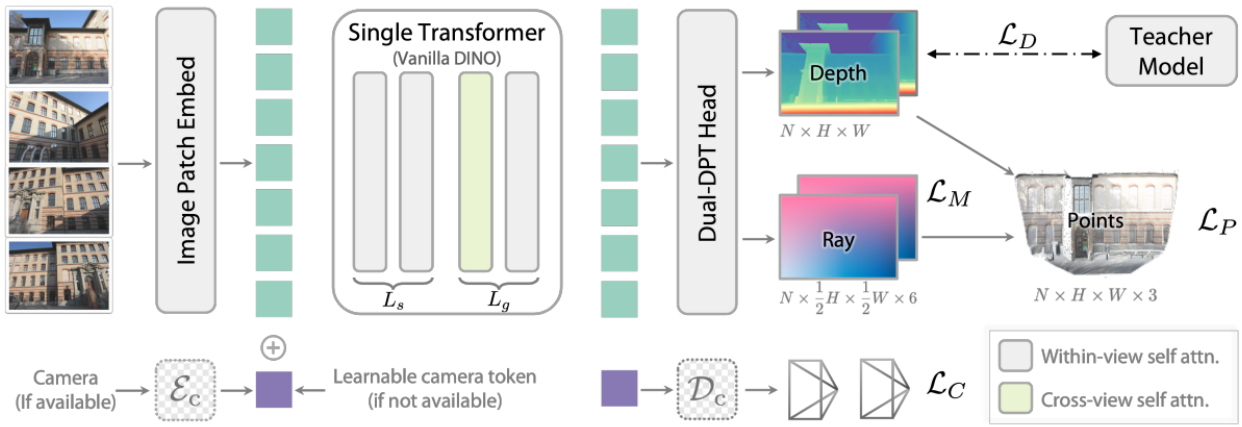


Figure 5: **Depth Anything 3 architecture.** Multiple images are patch-embedded and processed by a single transformer with L_s within-view self-attention layers followed by L_g alternating layers that interleave within-view and cross-view attention. A Dual-DPT head decodes per-view depth maps and ray maps, while a camera encoder/decoder branch predicts camera parameters. Supervision comes from a teacher depth model ($\mathcal{L}_D$), ray matching ($\mathcal{L}_M$), pointmap reconstruction ($\mathcal{L}_P$), and camera losses ($\mathcal{L}_C$). Figure taken from [29].

global attention mechanism over all views simultaneously; it also introduces a brief test-time post-optimization that refines the feed-forward output with 1000 gradient steps of photometric loss, a refinement strategy that NoPo4D adopts directly. Splatt3R [41] extends MASt3R with a Gaussian prediction head that jointly regresses per-pixel point maps and Gaussian attributes in a single forward pass; a two-stage training strategy avoids the local minima present when fitting Gaussian splats directly from stereo views. Pref3R [7] targets variable-length sequences by augmenting the DUSt3R backbone with a spatial memory network that incrementally accumulates reconstructed geometry across ordered frames, enabling pose-free novel-view synthesis at 20 FPS without any per-scene optimization. PF3plat [16] addresses pose-free video by initializing Gaussians from a pre-trained monocular depth estimator and refining cross-view alignment with geometry confidence scores that downweight unreliable depth regions, explicitly mitigating the gradient noise induced by Gaussian misalignment across views. YoNoSplat [66] unifies pose-conditioned and pose-free prediction in a single model: an Intrinsic Condition Embedding module estimates camera intrinsics when unavailable, and a mixing training strategy that gradually transitions from ground-truth to predicted poses allows the model to handle any combination of calibrated and uncalibrated inputs. All of these methods are restricted to static scenes.

Extending feed-forward reconstruction to dynamic scenes requires supervising temporal motion without dense 3D annotations. L4GM [38] proposed the first feed-forward 4D Gaussian model, conditioning a video diffusion prior on a single reference image to predict per-timestep Gaussians that animate an object over time; while it demonstrates compelling object-level dynamics, its training on curated single-object datasets with known camera trajectories means it does not generalize to unbounded multi-object environments or casual capture conditions. 4DGT [59] introduced the first feed-forward dynamic Gaussian model for general scenes, equipping each Gaussian with a temporal center, a lifespan, and asymmetric forward and backward linear and angular velocities; it achieves real-time rendering of dynamic scenes without per-scene optimization, but requires known camera poses and processes a single camera stream at a time. NeoVerse [63] relaxes the pose requirement of 4DGT but supervises 3D velocities directly against ground-truth scene flow, which is available only in synthetic datasets, limiting its applicability to real captures. MoVieS [28] extends monocular feed-forward reconstruction with motion-aware feature extraction but retains the single-camera constraint. DGGT [5] is the most directly comparable prior work, operating in the unposed multi-view dynamic setting; however, it is designed specifically for driving scenarios and does not transfer to general environments with varied motion and camera arrangements. In the experiments, DGGT is retrained on the same training data as NoPo4D to isolate its architectural limitations from distribution effects. UFO-4D [20] achieves unposed feed-forward 4D reconstruction from stereo pairs by lifting per-frame disparity maps into time-varying Gaussian fields, but its two-camera constraint and reliance on synchronized stereo capture limit applicability to wider multi-camera arrays. At the time of writing, no prior method jointly handles dynamic scenes, multi-view input, unknown camera poses, and general scene content in a single feed-forward pass; NoPo4D is the first to do so.

Taken together, this survey reveals a clear gap in the landscape of feed-forward reconstruction methods, which is quantified in Table 1 of the Introduction. The feed-forward paradigm offers two advantages that per-scene optimization cannot match: inference time measured in seconds rather than minutes, and zero-shot generalization

to scenes never seen during training. Static pose-free methods such as NoPoSplat and PF3plat demonstrate that accurate camera calibration is not a prerequisite for high-quality Gaussian prediction; feed-forward dynamic methods such as 4DGT and NeoVerse demonstrate that costly per-scene optimization is not required for temporal modeling. Yet no single method combines all four desiderata simultaneously. The three design choices of NoPo4D address each missing ingredient in turn: the velocity decomposition replaces unavailable 3D scene flow supervision with 2D optical flow, enabling motion learning without known poses or depth-sensor annotations; the bidirectional motion encoder aggregates temporal features across frames and cameras to handle ambiguous motions that are underdetermined from individual pairs; and view-dependent opacity absorbs the cross-view Gaussian misalignments that arise when camera parameters are estimated rather than prescribed, ensuring stable training across a variety of capture conditions.

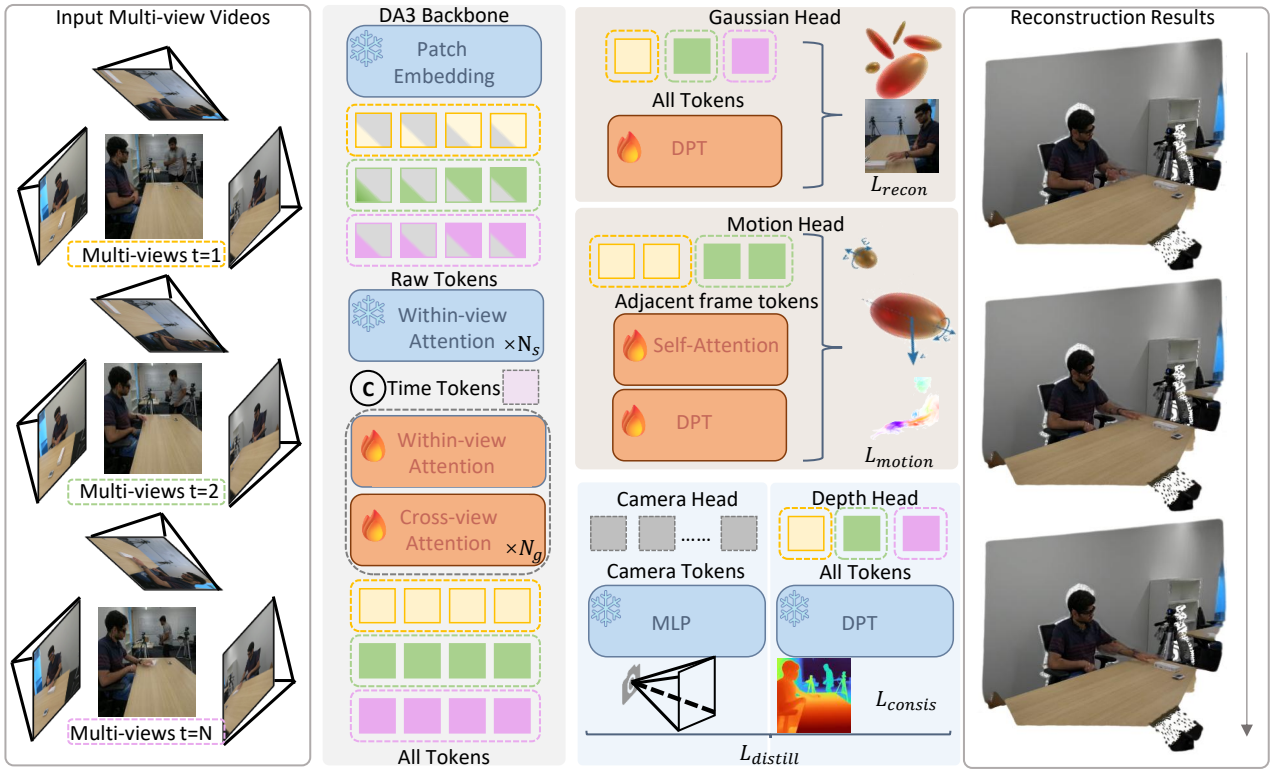


Figure 6: **NoPo4D architecture.** Given C streams of time-synchronized video, DA3 [29] extracts multi-view features through alternating within-view and cross-view attention. Frozen depth and camera heads recover per-frame geometry, which is unprojected into Gaussian means $\boldsymbol{\mu}$. A trainable Gaussian head decodes static parameters $(\mathbf{R}, \mathbf{s}, \alpha, \mathbf{c})$, while a bidirectional motion branch processes consecutive-frame features to produce velocities $(\mathbf{v}^{\pm}, \boldsymbol{\omega}^{\pm})$, motion gate ρ, and temporal variance σ_g.

3. Method

3.1. Problem Formulation

The system considers C uncalibrated, time-synchronized video streams, $\mathcal{I} = \left\{(\mathbf{I}_t^c)_{t=1}^T\right\}_{c=1}^C$, where $\mathbf{I}_t^c \in \mathbb{R}^{H\times W\times 3}$ is the frame from camera c at timestep t. The cameras are assumed to form a *static rig* (the relative pose between cameras is constant over time) as in capture setups such as Ego-Exo4D [13]. Intrinsics, extrinsics, and distortion parameters are all unknown.
NoPo4D simultaneously estimates per-camera intrinsics $\hat{\mathbf{K}}^c$ and extrinsics $(\hat{\mathbf{R}}^c, \hat{\mathbf{t}}^c)$, obtained by averaging the backbone's per-frame pose predictions across t, and a collection of G 4D Gaussians [24, 57]:

$$\left(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \alpha_g, \mathbf{c}_g, \tau_g, l_g, \mathbf{v}_g^+, \mathbf{v}_g^-, \boldsymbol{\omega}_g^+, \boldsymbol{\omega}_g^-\right)_{g=1}^G. \tag{9}$$

Each Gaussian carries static attributes [24]: mean $\boldsymbol{\mu}_g$, covariance $\boldsymbol{\Sigma}_g$, view-dependent opacity α_g, and view-dependent color $\mathbf{c}_g$ (both encoded as spherical harmonic coefficients), and dynamic attributes [57]: temporal center τ_g, lifespan l_g, and forward and backward linear and angular velocities $\mathbf{v}_g^{\pm}, \boldsymbol{\omega}_g^{\pm}$ that permit asymmetric motion around the temporal center. The total number of Gaussians is $G = C \cdot T \cdot H \cdot W$: one Gaussian is instantiated per pixel, per frame, per camera.

3.2. Model Architecture

The system has four stages, illustrated in Figure 6. A pretrained transformer backbone processes all input frames and produces dense multi-view feature tokens. Frozen depth and camera decoder heads then predict per-frame depth maps and camera parameters, which are back-projected to initialize Gaussian means. A trainable DPT Gaussian head decodes the remaining static Gaussian parameters from the backbone features. Finally, a trainable bidirectional motion encoder processes consecutive-frame features to predict the velocity

attributes and temporal parameters for each Gaussian. All outputs are assembled into the full 4D Gaussian representation and rendered via differentiable tile-based rasterization.

3.2.1 Feature Backbone

The pretrained transformer backbone from Depth Anything 3 Large [29] is employed. The backbone processes input frames through a stack of frozen within-view self-attention layers that build per-image representations independently, followed by trainable alternating layers that interleave within-view and cross-view attention to fuse information across cameras. Before the alternating layers, sinusoidal temporal tokens are injected that encode each frame's normalized timestamp in $[0, 1]$. These tokens make the backbone aware of temporal ordering, enabling the cross-view attention to aggregate across both viewpoints and timepoints. The backbone output is a set of multi-view feature tokens $\mathcal{F} = \left\{(\mathbf{F}_t^c)_{t=1}^T\right\}_{c=1}^C$, where $\mathbf{F}_t^c \in \mathbb{R}^{N\times D}$ has $N = HW/P^2$ spatial tokens, $P = 14$ is the ViT patch size of DA3, and D is the embedding dimension.
Frozen pretrained decoder heads then predict per-frame depth maps $\hat{\mathbf{D}}_t^c \in \mathbb{R}_{>0}^{H\times W}$, intrinsics $\hat{\mathbf{K}}_t^c$, and extrinsics $(\hat{\mathbf{R}}_t^c, \hat{\mathbf{t}}_t^c)$. Under the static-rig assumption, these per-frame estimates are averaged over t to yield stable per-camera parameters $\hat{\mathbf{K}}^c$, $\hat{\mathbf{R}}^c$, $\hat{\mathbf{t}}^c$.
Each depth map is then back-projected into world space. The unprojection of pixel (x, y) with depth d using camera parameters $(\hat{\mathbf{R}}^c, \hat{\mathbf{t}}^c, \hat{\mathbf{K}}^c)$ gives:

$$\hat{\mathbf{P}}_t^c = \Pi^{-1}(\hat{\mathbf{D}}_t^c, \hat{\mathbf{R}}^c, \hat{\mathbf{t}}^c, \hat{\mathbf{K}}^c), \tag{10}$$

producing a world-space point map $\hat{\mathbf{P}}_t^c \in \mathbb{R}^{H\times W\times 3}$. Each Gaussian is uniquely indexed by $g = (c, t, x, y)$, and its mean $\boldsymbol{\mu}_g$ is set to the corresponding point-map entry $\hat{\mathbf{P}}_t^c(x, y)$. This one-Gaussian-per-pixel design is not merely convenient for initialization; it is the property that makes the velocity decomposition possible, because it establishes a direct correspondence between each Gaussian and its source pixel.
Keeping the depth and camera heads frozen during training is critical, as the gradient signal from Gaussian rendering is too noisy to preserve the pretrained geometric priors.

3.2.2 Static Attribute Decoding

A DPT decoder head applied to the backbone features $\mathbf{F}_t^c$ predicts the remaining static parameters per pixel. The covariance $\boldsymbol{\Sigma}_g$ is factorized as $\boldsymbol{\Sigma}_g = \mathbf{R}_g \operatorname{diag}(\mathbf{s}_g)^2 \mathbf{R}_g^T$, where the rotation $\mathbf{R}_g$ is obtained from a predicted unit quaternion $\mathbf{q}_g \in \mathbb{R}^4$ and the scale $\mathbf{s}_g \in \mathbb{R}_{>0}^3$ is decoded with an exponential activation. Color $\mathbf{c}_g$ is decoded as spherical harmonic coefficients of order k, enabling view-dependent appearance.
Opacity is parameterized by SH coefficients $\alpha_g \in \mathbb{R}^{(k+1)^2}$ rather than a scalar, making it view-dependent. This design choice is far more important here than in monocular settings: with a single camera, all Gaussians are unprojected from the same viewpoint and spatial misalignments are minimal. In the multi-view case instead, camera parameters are estimated rather than given, so Gaussians unprojected from different cameras are inevitably slightly misaligned in 3D space: a point visible from two cameras will produce two Gaussians at slightly different world positions, and neither can be discarded without losing valid photometric information from one of the views. A scalar opacity would force a global trade-off between keeping or suppressing such a Gaussian, whereas SH-parameterized opacity allows the model to learn a per-direction confidence: the Gaussian remains fully opaque from the camera that unprojected it correctly and fades for cameras where the reprojection is inconsistent. This acts as a soft, learned data-selection mechanism that gracefully handles the geometric imprecision inherent to pose-free reconstruction, without requiring any explicit outlier detection or multi-view consistency loss. The ablation in Section 4.5 confirms this: replacing SH opacity with a scalar is the second most damaging architectural modification.

3.2.3 Decomposed Motion Parameterization

The central technical contribution of NoPo4D is a velocity decomposition that avoids both the differentiable-rendering dependence of posed methods and the 3D-motion-annotation requirement of pose-free methods.
Prior feed-forward dynamic methods [59, 63] predict 3D Gaussian velocities directly in world space. To supervise these velocities, they must either render the predicted motion as 2D optical flow and compare to ground-truth flow (which ties supervision quality to pose accuracy, as rendered flow depends on the projected Gaussian positions) or compare 3D velocities to 3D scene flow ground truth (unavailable at scale). The proposed decomposition avoids both paths.
A DPT head decodes, for each pixel (x, y) in frame t, a 2D image-plane shift $(\Delta x, \Delta y)$ and a depth displacement Δd. The pixel shifts are bounded by a tanh nonlinearity and scaled to image dimensions (W, H). These quantities define the projected position and depth of the same scene point at the next timestep:

$$x' = x + \Delta x, \quad y' = y + \Delta y, \quad \hat{\mathbf{D}}_{t+1}^c(x, y) = \hat{\mathbf{D}}_t^c(x, y) + \Delta d. \tag{11}$$

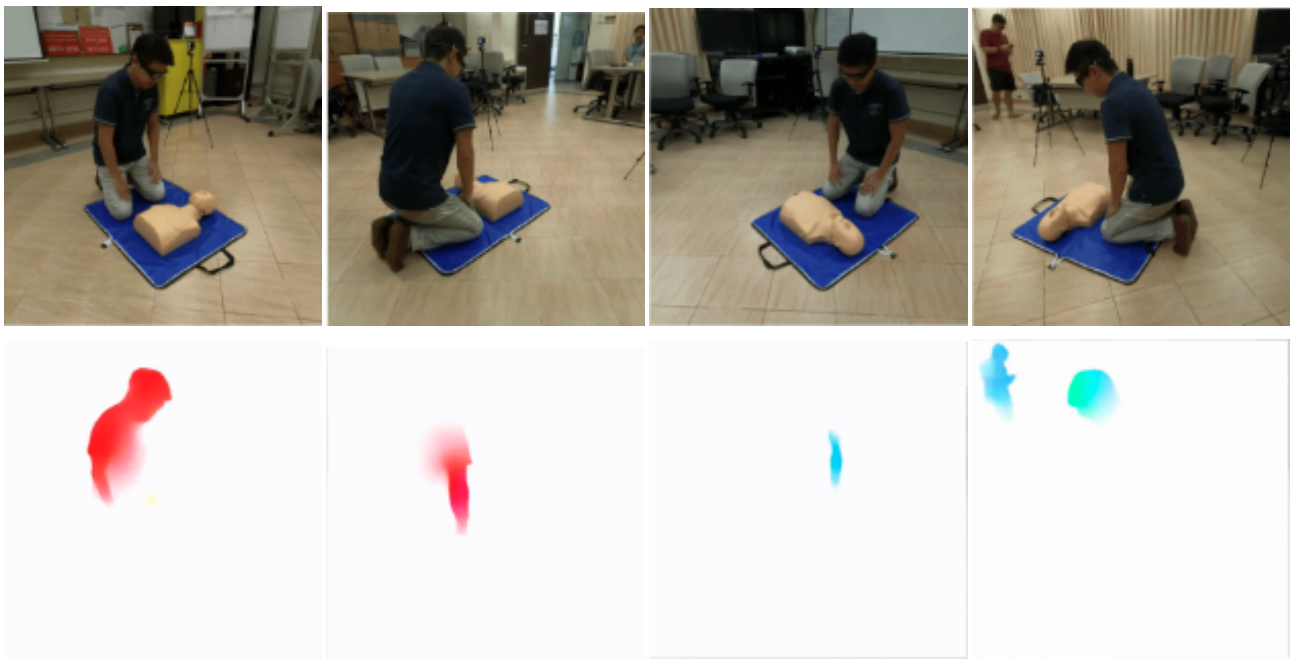


Figure 7: **Input frames and predicted optical flow.** Top row: example RGB frames from the training set. Bottom row: corresponding image-plane shift fields $(\Delta x, \Delta y)$ predicted by NoPo4D, which encode per-pixel motion directly without any rendering step.

Using the predicted intrinsics $\hat{\mathbf{K}}^c$ with focal lengths (f_x, f_y) and principal point (c_x, c_y), source and displaced pixels are unprojected into camera-space coordinates, and their difference is computed:

$$\Delta \mathbf{X}_{\text{cam}} = \begin{pmatrix} \dfrac{(x' - c_x)\,\hat{\mathbf{D}}^c_{t+1} - (x - c_x)\,\hat{\mathbf{D}}^c_t}{f_x} \\ \dfrac{(y' - c_y)\,\hat{\mathbf{D}}^c_{t+1} - (y - c_y)\,\hat{\mathbf{D}}^c_t}{f_y} \\ \hat{\mathbf{D}}^c_{t+1} - \hat{\mathbf{D}}^c_t \end{pmatrix}. \tag{12}$$

The forward linear velocity $\mathbf{v}_g^+$ is obtained by rotating this camera-space displacement into world coordinates and normalizing by the temporal interval Δt:

$$\mathbf{v}_g^+ = \frac{\hat{\mathbf{R}}^c\,\Delta \mathbf{X}_{\text{cam}}}{\Delta t}. \tag{13}$$

The backward velocity $\mathbf{v}_g^-$ is computed symmetrically from the preceding frame pair.
The key property of this parameterization is that the image-plane shift $(\Delta x, \Delta y)$ encodes the same quantity as the 2D optical flow at pixel (x, y): it is the displacement of the source Gaussian's projected position between consecutive frames, computed directly from the predicted quantities without any rendering step. This means it can be compared directly to pseudo-ground-truth optical flow estimated by an off-the-shelf model, giving a training signal that is completely decoupled from the quality of the predicted camera poses. Figure 7 shows example input frames and their corresponding SEA-RAFT optical flow fields used as supervision targets.
Empirical verification in the ablation study of Section 4.5 confirms that this coupling is necessary. When both the decomposition and the optical-flow loss are removed simultaneously (ablation *No decomposition + no flow loss*), training diverges: velocity magnitudes grow unboundedly because there is no rendering-free signal to constrain them in the pose-free setting. Replacing only the decomposition while retaining flow supervision through rendering (ablation *No decomposition*) also degrades performance, confirming that the direct pixel-level pathway provides a cleaner gradient.

3.2.4 Bidirectional Motion Encoder

The image-plane shifts, depth changes, angular velocities, and temporal parameters are all produced by a bidirectional motion encoder $\mathcal{M}$. The key design principle is that motion estimation requires *two* sources of context that the backbone alone cannot provide: temporal context (what changed between frames) and cross-view context (how the same motion looks from different cameras). The encoder is purpose-built to aggregate both simultaneously.

For each consecutive frame pair $(t, t+1)$, the encoder concatenates feature tokens from all C cameras at frame t with tokens from all C cameras at frame $t+1$, yielding a combined sequence of $2CN$ tokens. Two learnable temporal embeddings are added to distinguish source-frame tokens from neighbor-frame tokens, giving the attention mechanism an explicit signal about which tokens belong to which timestep. A standard multi-head self-attention block then performs joint attention over the full sequence, allowing every spatial location in every camera to attend to every other location across both frames and all viewpoints in a single pass. This joint attention is what makes the encoder bidirectional in the multi-view sense: a token from camera c at time t can directly attend to tokens from camera c' at time $t+1$, resolving motion ambiguities that would be unresolvable from a single stream. The resulting tokens are split back into forward ($t \to t+1$) and backward ($t \leftarrow t+1$) halves and decoded by a shared DPT head into pixel-aligned motion fields.

For each Gaussian g, the motion encoder produces: the forward and backward image-plane shifts and depth displacements used to compute $\mathbf{v}_g^{\pm}$ via Equations (11)–(13); forward and backward angular velocities $\boldsymbol{\omega}_g^{\pm} \in \mathbb{R}^3$ encoding rotational motion around the Gaussian's center; a per-pixel motion gate $\rho_g \in (0, 1)$, implemented as a sigmoid-activated scalar, that suppresses velocity predictions in static scene regions where motion should be zero; a temporal covariance $\sigma_g > 0$ controlling how long the Gaussian remains visible around its temporal center; and a temporal center τ_g anchored to the source frame's timestamp. The motion gate is particularly important: without it, the model tends to hallucinate small spurious velocities for background Gaussians, which accumulates into visible flickering artifacts.

The ablation *No motion branch* (Section 4.5) evaluates the alternative of predicting velocities directly from backbone features without the cross-frame, cross-view attention step. Without the joint attention, each camera's velocity is estimated independently from single-frame features, producing predictions that are inconsistent across views and that cannot exploit the temporal context needed to resolve motion ambiguities.

3.3. Training

3.3.1 Losses

The total training loss combines four terms:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{motion}}\mathcal{L}_{\text{motion}} + \lambda_{\text{consis}}\mathcal{L}_{\text{consis}} + \mathcal{L}_{\text{distill}}. \tag{14}$$

The *reconstruction loss* $\mathcal{L}_{\text{recon}} = \lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}} + \lambda_{\text{SSIM}}\mathcal{L}_{\text{SSIM}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}}$ is evaluated on target frames rendered from the predicted Gaussians and poses, aligned into the scene coordinate frame via the two-pass procedure described in Section 3.3.2.

The *motion loss* supervises Gaussian dynamics by comparing predicted pixel shifts to pseudo-ground-truth optical flow $\mathbf{f}^*$ from SEA-RAFT [54]. To avoid training on static background noise, supervision is applied only to pixels where the ground-truth shift magnitude exceeds 2 pixels, denoting this set $\Omega' = \{p \in \Omega \mid \|\mathbf{f}^*(p)\|_2 > 2\}$:

$$\lambda_{\text{motion}}\mathcal{L}_{\text{motion}} = \lambda_{\text{flow}} \frac{1}{|\Omega'|} \sum_{p \in \Omega'} \left\| (\Delta x, \Delta y)(p) - \mathbf{f}^*(p) \right\|_1 . \tag{15}$$

This direct pixel-level comparison bypasses rendering entirely, decoupling motion supervision from pose estimation errors.

The *depth consistency loss* penalizes deviations of rendered depth from the backbone's own predictions:

$$\lambda_{\text{consis}}\mathcal{L}_{\text{consis}} = \lambda_{\text{consis}} \frac{1}{|\Omega|} \sum_{p \in \Omega} \|\hat{\mathbf{D}}(p) - \mathbf{D}(p)\|_2^2. \tag{16}$$

This loss acts as a regularizer that prevents Gaussians from drifting away from their initialized positions, but it cannot stand alone: without geometric grounding from the other losses, the model finds degenerate solutions that collapse all Gaussians onto a flat surface (as the loss ablation in Table 5 shows).

The *distillation loss* preserves the pretrained geometric priors of the DA3 backbone by distilling from the frozen DA3 Giant model:

$$\mathcal{L}_{\text{distill}} = \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{normal}}\mathcal{L}_{\text{normal}}, \tag{17}$$

where $\mathcal{L}_{\text{pose}}$ is a Huber loss on predicted versus teacher poses, and depth and normal losses are ℓ_2 distances $\|\mathbf{D} - \mathbf{D}^*\|_2^2$ and $\|\nabla\mathbf{D} - \nabla\mathbf{D}^*\|_2^2$. Removing distillation is the single most damaging modification in the loss ablation, confirming that the pretrained DA3 representations are easily corrupted by the noisy gradients from dynamic Gaussian training if not anchored to their original predictions. Each auxiliary loss thus addresses a distinct failure mode: distillation preserves priors, flow grounds dynamics, and consistency enforces coherence.

Multi-View Epipolar Consistency Loss (optional). A 3D point observed by two cameras simultaneously must carry the same world-space velocity. For each pixel (x, y) in camera c, the predicted depth $\hat{\mathbf{D}}_t^c(x, y)$ and the estimated camera parameters are used to unproject it to a world point, which is then reprojected into camera c' to find the corresponding pixel (x', y'). Denoting by $\Omega_{cc'}$ the set of pixels whose reprojection falls inside camera c', and by $g = (c, t, x, y)$, $g' = (c', t, \lfloor x' \rceil, \lfloor y' \rceil)$ the two Gaussians, the loss penalizes their velocity difference:

$$\mathcal{L}_{\text{mv}} = \frac{1}{C(C-1)} \sum_{\substack{c,c'=1 \\ c \neq c'}}^{C} \frac{1}{|\Omega_{cc'}|} \sum_{(g,\, g') \in \Omega_{cc'}} \left\| \mathbf{v}_g^+ - \mathbf{v}_{g'}^+ \right\|_1, \tag{18}$$

with backward velocities $\mathbf{v}^-$ handled symmetrically. A critical subtlety is that reprojection alone does not guarantee a valid correspondence: the reprojected pixel (x', y') may be occluded in camera c', in which case g' observes a different surface entirely. Detecting occlusions requires comparing the reprojected depth against the depth predicted for (x', y') in camera c', but both depth values are themselves estimated and noisy, making this check unreliable in practice. Accurate correspondences could instead be obtained with a pretrained dense matcher such as LightGlue [30] or RoMa [11], which produce geometrically robust matches without relying on depth. However, running such a model for every camera pair at every training step would substantially increase training cost, making this loss impractical for large-scale training. We therefore did not include it in the final objective.

3.3.2 Training Curriculum

Training proceeds in two stages of 20,000 steps each. The first stage trains on single-camera viewpoints, which initializes the static scene representation and the motion heads on individual streams before the added complexity of multi-camera fusion is introduced. The second stage extends to batches of 16 frames sampled across 1–4 cameras simultaneously, with temporal stride between consecutive frames sampled from 4–8 and warmed up linearly over the first 2,000 steps. Starting with closely-spaced frames allows the motion encoder to build reliable predictions from small displacements, where optical flow estimates are most accurate, before gradually extending to wider temporal baselines where motion ambiguity is higher.

Since NoPo4D is pose-free, rendering target views during training requires mapping the predicted target poses into the scene's coordinate frame, which is defined implicitly by the backbone's context-frame predictions. A two-pass strategy is employed. In the first pass (with gradients), only context frames are processed, producing Gaussians and context-frame poses that define the scene frame. In the second pass (without gradients), all frames including target frames are processed, yielding fresh context and target poses. The Umeyama algorithm [45] then finds the closed-form Sim(3) mapping from the second-pass context poses to the first-pass context poses, and this transformation is applied to the target poses to bring them into the scene frame. Because the second pass is detached from the computational graph, the rendering loss supervises only the first-pass Gaussians, not the encoder's pose predictions. The same two-pass procedure is used at inference time when rendering from novel viewpoints.

3.3.3 Backbone Fine-Tuning Strategy

Among the backbone's components, NoPo4D fine-tunes only the alternating cross-view attention layers, keeping the within-view layers and the pretrained depth and camera heads frozen. The rationale is that the alternating layers govern cross-camera feature fusion, which is the aspect of the backbone's computation that most differs between DA3's static-scene pretraining and the dynamic multi-view fine-tuning objective. Adapting these layers allows the backbone to integrate temporal tokens and handle dynamic content, while the frozen within-view layers preserve the broad per-image representations that make DA3 features generally useful. The ablation study in Section 4.5.3 compares six configurations and shows a monotonic improvement as more alternating layers are unfrozen, with fine-tuning all 16 layers achieving the best result.

3.3.4 Post-Optimization

An optional test-time refinement stage, following AnySplat [22], takes the feed-forward output as initialization and applies 100 gradient steps of photometric optimization against the input context views, after pruning Gaussians with opacity below 0.01. Each Gaussian attribute is optimized with a dedicated learning rate: 1.6×10^{-4} for $\boldsymbol{\mu}_g$, 5×10^{-3} for $\mathbf{s}_g$, 1×10^{-3} for $\mathbf{q}_g$, 5×10^{-2} for α_g, 2.5×10^{-3} for $\mathbf{c}_g$, 5×10^{-3} for $(\hat{\mathbf{R}}^c, \hat{\mathbf{t}}^c)$, 1×10^{-4} for τ_g, 5×10^{-3} for σ_g, and 1×10^{-2} for ρ_g. These per-parameter rates reflect the different scales and sensitivities of each attribute: opacity benefits from aggressive updates to quickly suppress floaters, while position requires a small rate to avoid disrupting the geometrically accurate initialization. Because the feed-forward initialization is already geometrically accurate, convergence is rapid (under 5 seconds per scene) and

the refinement corrects residual Gaussian placement errors that the feed-forward pass cannot resolve due to its single-inference nature. This stage is entirely optional; the feed-forward-only results already outperform all prior feed-forward baselines.

4. Experiments

4.1. Implementation Details

NoPo4D is trained on approximately 2,900 Ego-Exo4D [13] scenes, a large-scale dataset of everyday human activities captured by synchronized multi-camera rigs in diverse environments including kitchens, sports facilities, and medical settings. Frames are undistorted and downsampled to a maximum of 448 pixels on the longer side. Following AnySplat [22], random aspect-ratio jittering is applied in $[0.5, 1.0]$, center-cropping at 77–100%, and horizontal flipping. Training runs in two stages of 20,000 steps each: a single-camera stage followed by a multi-camera stage with 16-frame batches across 1–4 cameras. AdamW [32] is used with separate learning rates of 5×10^{-6} for the backbone alternating layers, 5×10^{-5} for the motion encoder, and 1×10^{-4} for the velocity DPT head and Gaussian head. All rates follow a 1,000-step linear warmup and cosine annealing to one-tenth of the initial value. Loss weights are $\lambda_{\text{MSE}} = 1$, $\lambda_{\text{LPIPS}} = 0.5$, $\lambda_{\text{SSIM}} = 0.05$, $\lambda_{\text{consis}} = 0.1$, $\lambda_{\text{flow}} = 0.02$, $\lambda_{\text{pose}} = 1.0$, $\lambda_{\text{depth}} = 0.1$, $\lambda_{\text{normal}} = 0.1$. Training uses batch size 1 per GPU on 4 NVIDIA GH200 GPUs with data parallelism.

4.2. Experimental Setup

The evaluation covers four multi-view dynamic benchmarks with different properties. *ExoRecon* [56] derives sparse multi-view captures from Ego-Exo4D and is thus the in-distribution benchmark, using the same test-scene split as MonoFusion to allow direct comparison. *Immersive Light Field (ILF)* [2] contains real-world dynamic scenes captured by a spherical 46-camera array, with scenes spanning concerts, sports events, and outdoor activities. *Kubric* [14] is a synthetic dataset with physically-simulated multi-object scenes rendered from multiple viewpoints, providing controlled large-displacement motion. *N3DV* [27] offers high-quality captures with 18–21 cameras, enabling evaluation under varying input-view counts and from novel viewpoints significantly removed from the input cameras. The latter three benchmarks are out-of-distribution relative to the Ego-Exo4D training data.

All benchmarks are evaluated with the same chunk-based protocol. Each scene is partitioned into non-overlapping chunks of 5 consecutive timestamps; the 4 surrounding frames serve as context input, and the middle frame is the held-out target. The target frame is synthesized at each input camera viewpoint and we report PSNR, SSIM [55], and LPIPS [69] averaged across cameras, chunks, and test scenes, capping evaluation at 60 chunks per scene for ExoRecon and 100 for the others. On N3DV, evaluation is additionally performed at moderate (5.7°–27.7°) and extreme (34.6°–71.9°) novel viewpoints.

Comparison is made against feed-forward and per-scene optimization baselines. Among feed-forward methods, NeoVerse [63] and MoVieS [28] are monocular methods extended to the multi-view setting by independently reconstructing each stream and aligning the per-view outputs. DGGT [5] is the most directly comparable method; it is retrained on the same Ego-Exo4D-derived training data as NoPo4D to eliminate distribution effects. The per-scene optimization baselines are MonoFusion [56], MV-SOM [50], and Dyn3D-GS [34], all of which require known camera poses.

4.3. Quantitative Results

Table 2 reports results on the in-distribution ExoRecon benchmark. NoPo4D substantially outperforms all feed-forward baselines, leading the nearest competitor (NeoVerse) by 9.1 dB PSNR. The gap over DGGT retrained on the same data (8.7 dB) isolates the architectural contribution from training-distribution effects. With the optional 100-step post-optimization, NoPo4D surpasses MonoFusion, the current per-scene optimization state of the art, by 1.5 dB PSNR, despite requiring no camera calibration. This demonstrates that a strong feed-forward initialization enables brief refinement to match or exceed methods built on full per-scene optimization.

Tables 3 and 4 report out-of-distribution performance on the remaining three benchmarks. On ILF and Kubric, NoPo4D consistently leads the strongest feed-forward baseline (DGGT), with a particularly pronounced advantage on Kubric (+3.5 dB feed-forward), which contains large-displacement motion that benefits most from the explicit motion parameterization. Post-optimization delivers further gains of 2–5 dB across both datasets, indicating that the feed-forward output generalizes well even outside the training distribution.

On N3DV (Table 4), NoPo4D leads by nearly 4 dB at input cameras and remains competitive at moderate novel viewpoints. At extreme viewpoints (34.6°–71.9°), pixel-aligned metrics begin to misrepresent perceptual quality: a blurry but pixel-aligned prediction can score higher than a sharper but slightly-shifted one. The qualitative comparisons of Figure 9 confirm that NoPo4D produces visually cleaner reconstructions with better-preserved geometry and texture at extreme angles, while DGGT exhibits semi-transparent ghosting and NeoVerse and MoVieS over-smooth.

Table 2: **Held-out view synthesis on ExoRecon [56]. FF** denotes feed-forward inference.

Method	FF	PSNR ↑	SSIM ↑	LPIPS ↓
Dyn3D-GS [34]	✗	24.28	0.692	0.539
MV-SOM [50]	✗	26.91	0.890	0.138
MonoFusion [56]	✗	30.43	**0.930**	**0.060**
NoPo4D (Ours, post-opt.)	✗	**31.95**	0.928	0.075
MoVieS [28]	✓	18.78	0.725	0.288
DGGT [5]	✓	19.67	0.645	0.417
DGGT (Ego-Exo4D) [5]	✓	20.44	0.704	0.418
NeoVerse [63]	✓	20.03	0.714	0.354
NoPo4D (Ours)	✓	**29.15**	**0.886**	**0.125**

Table 3: **Generalization to out-of-distribution datasets.** NoPo4D was not trained on either benchmark. NeoVerse was trained on Kubric and is therefore in-distribution on that benchmark.

	Immersive Light Field [2]			Kubric [14]		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MoVieS [28]	16.12	0.650	0.410	14.49	0.720	0.260
NeoVerse [63]	19.83	0.680	0.360	16.86	0.500	0.520
DGGT [5]	20.65	0.700	0.360	20.14	0.530	0.430
DGGT (Ego-Exo4D) [5]	20.23	0.685	0.453	18.22	0.551	0.522
NoPo4D (Ours)	21.63	0.740	0.280	23.61	0.739	0.256
NoPo4D (Ours, post-opt.)	**24.04**	**0.817**	**0.245**	**28.17**	**0.838**	**0.159**

4.4. Qualitative Evaluation

Figures 8 and 9 compare rendering quality across methods on three distinct scenarios, each exposing a different axis of difficulty.

ExoRecon: dynamic foregrounds. On real-world Ego-Exo4D captures, NoPo4D reconstructs fine-grained motion such as hand gestures, tool manipulation, and body pose changes that baselines consistently smear into blurry or doubled regions. The difference is most visible at object boundaries: NoPo4D produces clean, sharp edges around moving limbs, whereas NeoVerse and MoVieS blend foreground and background together because their monocular per-stream reconstruction cannot reason about the shared 3D structure. DGGT, despite being retrained on the same data, produces noticeably lower-frequency outputs and frequently hallucinates structure in occluded regions.

Kubric: large-displacement motion. The synthetic Kubric sequences contain fast-moving rigid objects with large inter-frame displacements, which stress-test the velocity decomposition. NoPo4D correctly localizes moving objects and preserves their shape across frames. Competing methods either misplace the objects (due to flow estimation failures at large displacements) or produce artifacts at their boundaries where the dynamic and static Gaussians are incorrectly merged. The explicit bidirectional velocity representation allows NoPo4D to handle these cases because the motion prediction is grounded in optical flow rather than implicit deformation fields.

N3DV: extreme novel viewpoints. At viewpoints 34.6°–71.9° away from the input cameras, NoPo4D's explicit 3D Gaussian representation maintains recognizable geometry and texture. The Gaussian primitives, initialized from accurate depth predictions, provide a geometrically consistent scaffold that supports synthesis far from the training views. In contrast, DGGT shows pronounced ghosting artifacts where overlapping Gaussians from different cameras are not coherently merged, and NeoVerse/MoVieS produce over-smoothed outputs that lose fine texture detail because their implicit representations cannot extrapolate reliably at large angular offsets. These qualitative differences also illustrate a known limitation of pixel-aligned metrics: at extreme viewpoints, numerical scores can favor blurry-but-aligned predictions over sharper-but-shifted ones, which partly explains why the quantitative gap understates the perceptual advantage visible in Figure 9.

Table 4: **View synthesis on N3DV [27],** evaluated at input cameras, moderate novel views, and extreme novel views.

	Input Cameras			Novel Cameras (Moderate)			Novel Cameras (Extreme)		
Method	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
MoVieS [28]	14.10	0.556	0.505	14.54	0.435	0.582	9.63	0.162	0.716
NeoVerse [63]	24.50	0.789	0.313	18.63	0.530	0.449	13.32	0.423	0.601
DGGT [5]	23.08	0.749	0.268	17.93	0.532	0.435	15.87	0.449	0.557
DGGT (Ego-Exo4D) [5]	22.49	0.713	0.356	6.84	0.223	0.626	10.43	0.419	0.631
NoPo4D (Ours)	28.43	0.905	0.144	**19.84**	**0.578**	**0.315**	15.69	**0.467**	**0.520**
NoPo4D (Ours, post-opt.)	**31.79**	**0.956**	**0.089**	19.93	0.570	0.320	**15.91**	0.449	0.531

Table 5: **Ablation studies on ExoRecon.** Each row in the left table removes one architectural component. Each row in the right table shows which auxiliary losses are present (✓) or absent (✗); the reconstruction loss is always active.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
No opacity SH	21.61	0.758	0.247
No motion branch	22.97	0.856	0.161
No decomposition	27.67	0.865	0.129
No temporal encoding	27.69	0.861	**0.125**
NoPo4D (Ours)	**29.15**	**0.886**	**0.125**

(a) Architectural components.

Recon.	Distill.	Consis.	Flow	PSNR	SSIM	LPIPS
✓	✗	✗	✗	25.09	0.829	0.162
✓	✗	✓	✗	3.51	0.238	0.803
✓	✗	✓	✓	25.91	0.815	0.159
✓	✓	✗	✓	28.02	0.867	**0.122**
✓	✓	✓	✗	28.19	0.869	0.125
✓	✓	✓	✓	**29.15**	**0.886**	0.125

(b) Auxiliary losses.

4.5. Ablation Studies

Table 5 isolates the contribution of each architectural component and each auxiliary loss on ExoRecon.

4.5.1 Architectural Components

The architectural ablation reveals that removing the bidirectional motion encoder (*No motion branch*, −6.2 dB) and replacing SH opacity with scalar opacity (*No opacity SH*, −7.5 dB) are by far the most damaging modifications. The magnitude of the motion-branch drop reflects the fact that independent per-camera velocity estimation cannot produce globally consistent motion: each camera reaches different conclusions about the direction of moving objects, and the resulting conflicting Gaussian velocities degrade rendering quality drastically. The opacity drop is comparably large because scalar opacity forces a binary choice (keep or discard a Gaussian globally) whereas SH opacity allows the network to suppress geometrically inconsistent Gaussians only from the specific viewpoints where they cause artifacts, preserving their contributions elsewhere. Replacing the velocity decomposition with direct 3D velocity prediction (*No decomposition*, −1.5 dB) confirms that the direct pixel-level supervision pathway is meaningfully cleaner than the rendered alternative. Removing the sinusoidal temporal tokens (*No temporal encoding*, −1.5 dB) shows that the alternating attention layers benefit from explicit temporal context.

4.5.2 Auxiliary Losses

The loss ablation reveals complementary roles for the three auxiliary losses. The consistency loss alone (*recon + consis*) is catastrophic (3.51 PSNR): without geometric grounding from distillation or flow, the model satisfies the depth constraint trivially by collapsing Gaussians onto a degenerate flat surface. This confirms that the consistency loss functions as a regularizer rather than a primary supervisory signal. In the leave-one-out comparisons, distillation (−3.2 dB) prevents the alternating attention layers from drifting away from their pretrained DA3 representations; flow (−1.0 dB) keeps velocity predictions from collapsing toward zero; and consistency (−1.1 dB) prevents multi-view geometric disagreements from persisting. The full configuration combines all three to address each failure mode simultaneously.

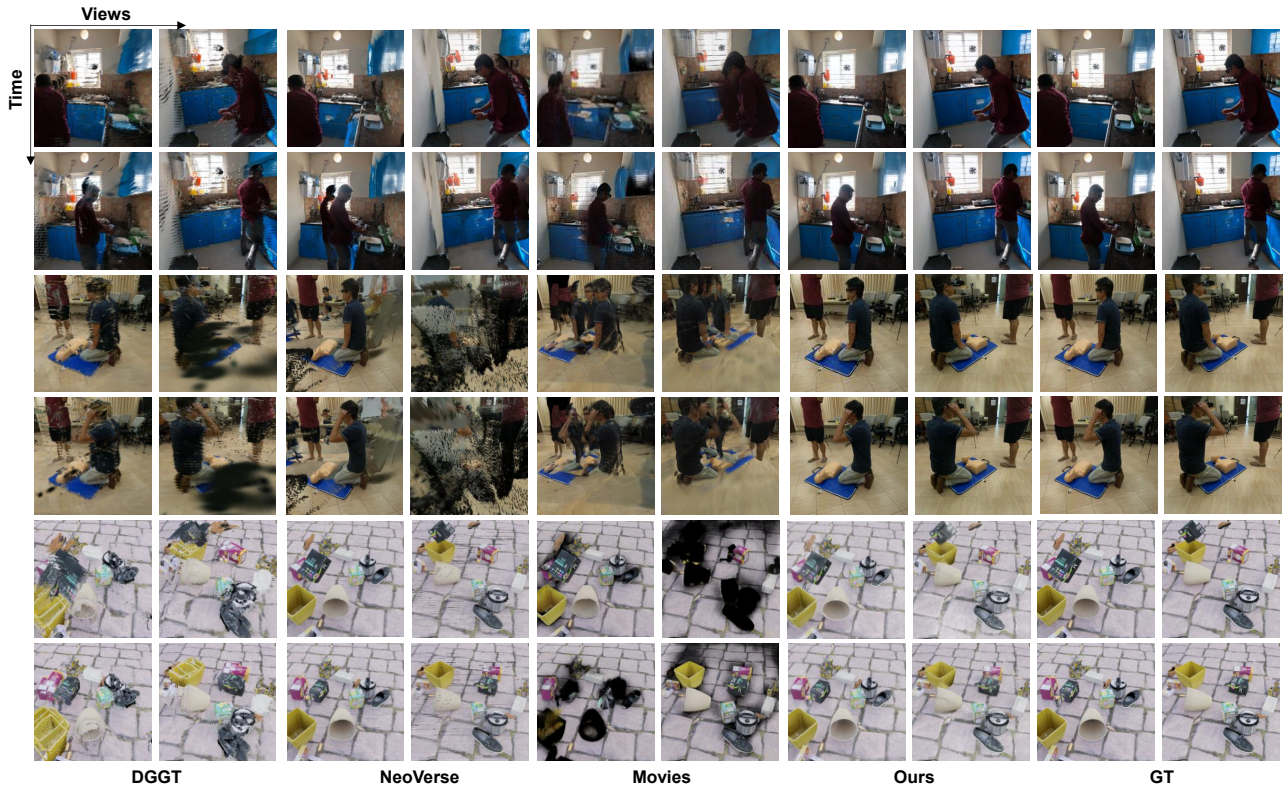


Figure 8: **Qualitative comparison on ExoRecon [56] and Kubric [14].** NoPo4D recovers fine-scale structure with sharp dynamic foregrounds while baselines exhibit blurring, geometric distortion, or missing motion.

4.5.3 Backbone Fine-Tuning Strategy

Table 6: **Backbone fine-tuning strategy comparison on ExoRecon.**

Configuration	Trainable params	PSNR ↑	SSIM ↑	LPIPS ↓
Unfreeze camera and depth heads	525 M	16.94	0.654	0.354
Full fine-tuning	621 M	25.09	0.829	0.162
Freeze backbone	266 M	25.68	0.804	0.151
Fine-tune last 4 layers	318 M	26.32	0.841	0.152
Fine-tune last 8 layers	369 M	27.28	0.853	0.140
Ours (16 alternating layers)	470 M	**29.15**	**0.886**	**0.125**

Table 6 compares six fine-tuning configurations. Unfreezing the depth and camera heads is catastrophic (−12.2 dB), confirming that these heads encode strong geometric priors that are easily destabilized by the noisy gradients from dynamic losses. Full fine-tuning also underperforms (−4.1 dB): the deeper backbone layers contain general-purpose representations that transfer well from DA3's large-scale pretraining and should be preserved. Completely freezing the backbone under-fits (−3.5 dB) because the dynamic multi-view setting requires task-specific adaptation that the frozen weights cannot provide. Progressive unfreezing shows monotonic improvement, and fine-tuning all 16 alternating attention layers while keeping the within-view layers and pretrained heads frozen achieves the best result. The alternating layers govern cross-view attention, which is precisely where the static-scene pretraining objective diverges from the dynamic multi-view fine-tuning objective.

4.6. Scalability Analysis

NoPo4D is trained with at most 4 cameras but evaluated here with $C \in \{2, \ldots, 12\}$ on N3DV, testing whether the cross-view attention generalizes to unseen camera configurations. Figure 10 reveals three key findings.

Quality. NoPo4D maintains a consistent and substantial quality lead over all baselines across the full range of camera counts on all three metrics. As C increases, all methods experience a moderate quality decrease: with

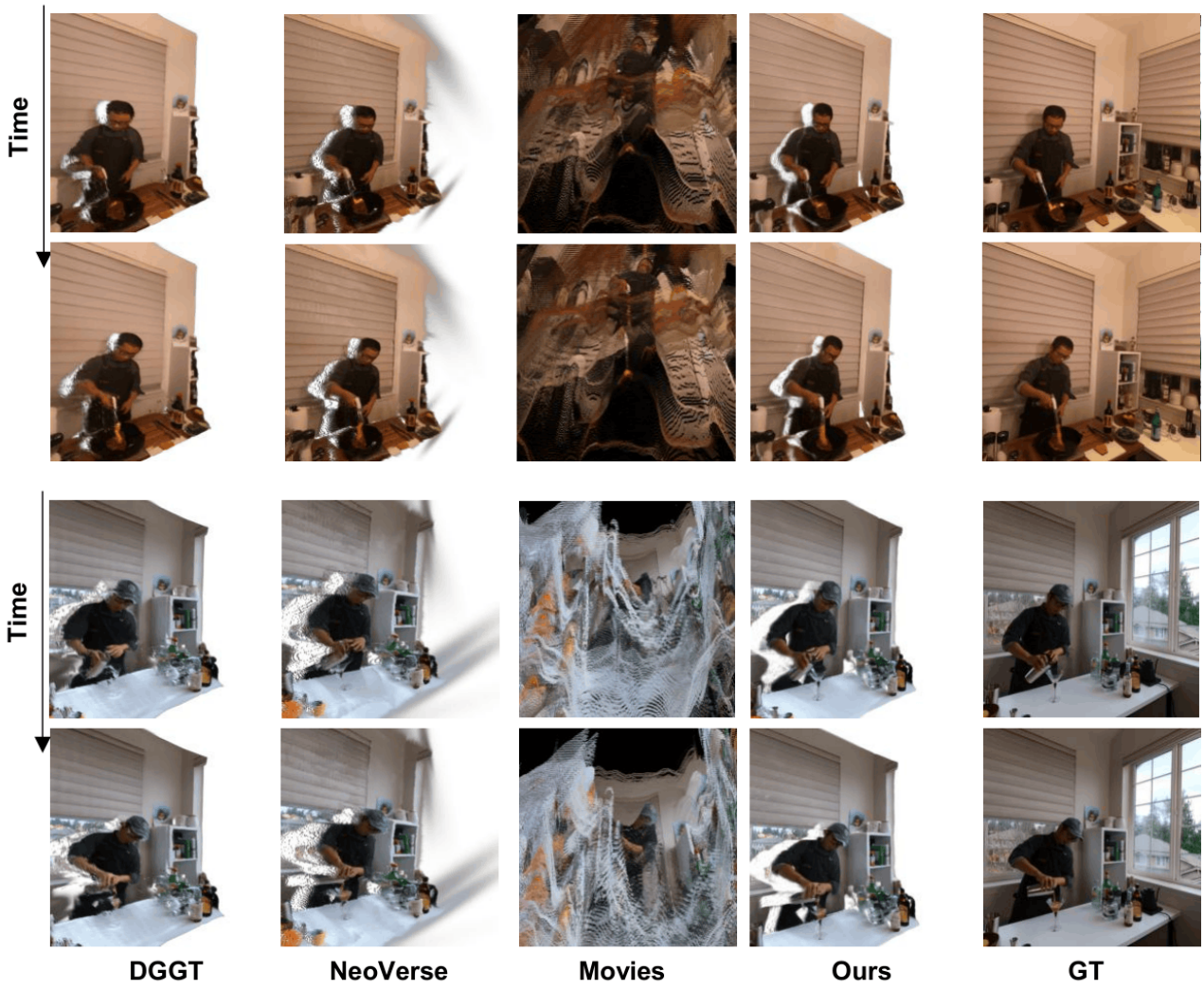


Figure 9: **Qualitative comparison on N3DV [27] under extreme viewpoint changes.** NoPo4D maintains recognizable geometry and texture at large viewpoint offsets, while DGGT shows ghosting and NeoVerse/MoVieS over-smooth.

more cameras, the held-out target views are further from any input camera, making the synthesis task harder. NoPo4D degrades most gracefully, losing only ∼4.7 dB PSNR from $C = 2$ to $C = 12$, compared to ∼6.9 dB for NeoVerse. DGGT loses ∼3.6 dB but starts from a much lower baseline, so the absolute gap to NoPo4D remains large throughout. The SSIM and LPIPS curves confirm the same ordering: NoPo4D consistently achieves higher structural similarity and lower perceptual distance than any baseline at every camera count.

Graceful generalization. The fact that quality does not collapse beyond $C = 4$ demonstrates that the cross-view attention mechanism in the backbone generalizes naturally to configurations unseen during training. This is a non-trivial property: the model must correctly fuse features from up to 12 cameras using attention weights learned exclusively from 1–4 camera inputs. The smooth degradation curve suggests that the learned attention patterns are robust to varying numbers of context tokens, likely because self-attention is permutation-invariant and the temporal embeddings provide sufficient structure even with more views.

Computational cost. Memory and inference time scale near-linearly with camera count for NoPo4D, NeoVerse, and MoVieS, reaching under 27 GB and 5 seconds at 12 cameras. DGGT, in contrast, grows super-linearly in both memory (∼50 GB) and inference time (∼15 seconds) at 12 cameras, making it impractical for dense multi-camera deployments. The near-linear scaling of NoPo4D follows directly from the one-Gaussian-per-pixel design: the total number of Gaussians grows linearly with C, and the cross-view attention, while quadratic in sequence length, is bounded in practice by the fixed input resolution.

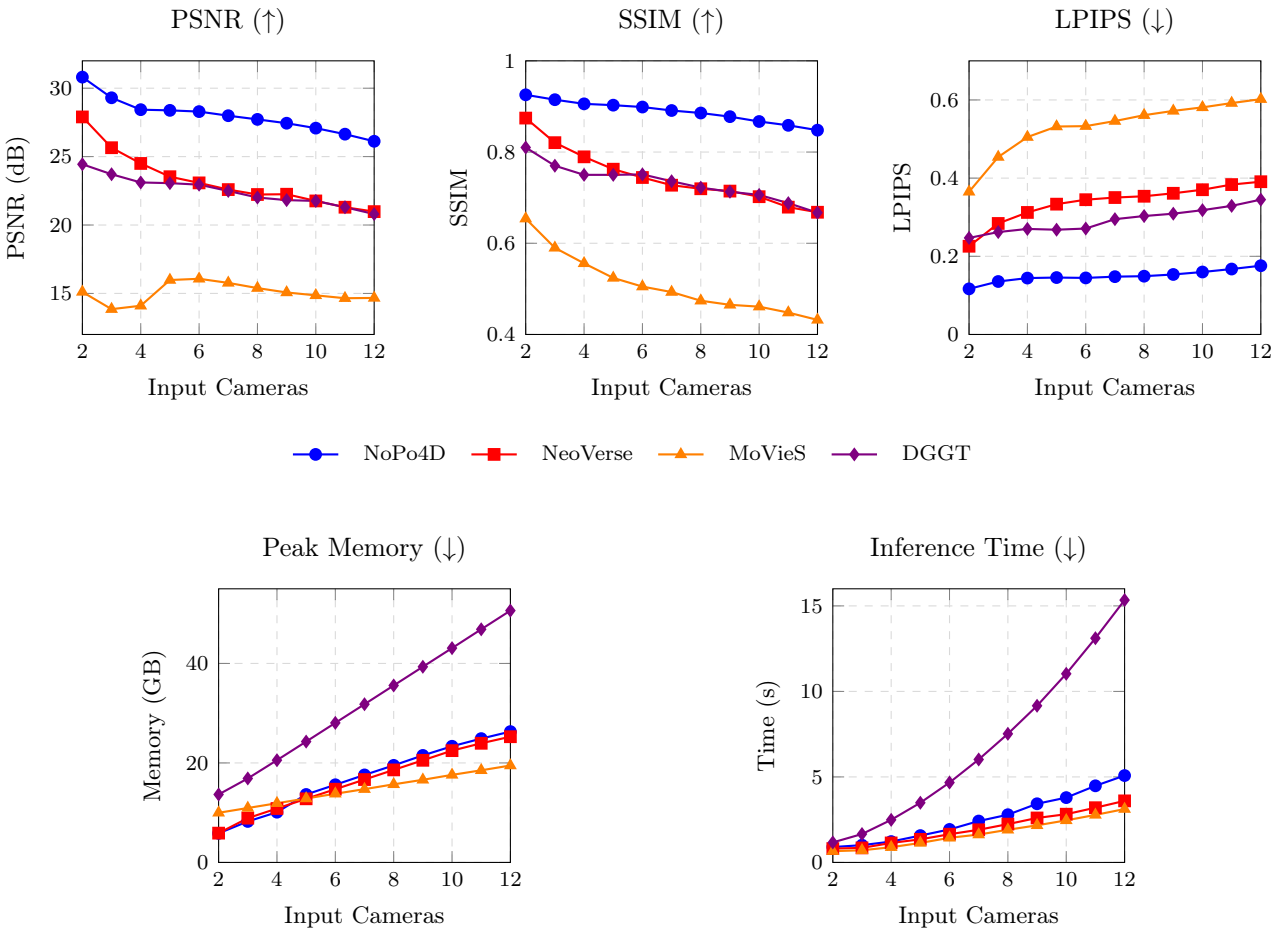


Figure 10: **Scalability on N3DV as the number of input cameras varies from 2 to 12.** Top row: rendering quality (PSNR, SSIM, LPIPS). Bottom row: peak GPU memory and inference time. NoPo4D was trained with at most 4 cameras and generalizes gracefully, while DGGT memory and inference time grow super-linearly.

5. Conclusions

This thesis introduced NoPo4D, the first feed-forward system for 4D Gaussian reconstruction from unposed multi-view video in general environments. Prior work had addressed at most two of the three key challenges simultaneously: modeling dynamic content, fusing multi-view information, and estimating geometry without known camera poses. NoPo4D fills the remaining gap by combining three design choices that each address a distinct obstacle in the joint problem.

The central contribution is the velocity decomposition, which splits each Gaussian's 3D motion into a 2D image-plane shift and a depth change. Because the image-plane component encodes the Gaussian's projected displacement without any rendering step, it can be supervised directly against pseudo-ground-truth optical flow from an off-the-shelf estimator. This creates a training signal that is completely decoupled from pose estimation accuracy, sidestepping the circular dependency that plagues posed dynamic methods and the synthetic-data requirement that limits unposed alternatives. This is the first work to establish a direct, render-free connection between 3D Gaussian velocities and 2D optical flow, opening a practical pathway for supervising dynamic reconstruction at scale.

The bidirectional motion encoder provides the multi-view consistency that the decomposition alone cannot guarantee. By performing joint self-attention over all cameras at two consecutive timesteps, it produces velocity predictions that are globally coherent across the camera rig, addressing the failure mode in which independently estimated per-camera velocities assign conflicting motion to the same scene point. View-dependent opacity, implemented with spherical harmonic coefficients, complements the encoder by allowing the network to suppress Gaussians from viewpoints where geometric estimation errors are worst, without globally discarding them.

Across four multi-view dynamic benchmarks—ExoRecon, Immersive Light Field, Kubric, and N3DV— NoPo4D consistently outperforms all feed-forward baselines and, with 100 steps of post-optimization, surpasses per-scene optimization methods that require known camera calibration, while running orders of magnitude faster.

5.1. Future Work

Several directions remain open:

- **Moving camera rigs.** NoPo4D assumes a static camera rig; when cameras themselves move, the per-camera pose averaging collapses distinct positions into a single incorrect estimate, degrading reconstruction quality.
- **Pretrained teacher dependency.** The model relies on pseudo-ground-truth signals from DA3 [29] and SEA-RAFT [54]; as these models improve, NoPo4D benefits automatically, but training quality is ultimately bounded by their accuracy.
- **Higher-order motion.** The velocity decomposition is currently first-order in time; capturing acceleration or periodic trajectories within the same feed-forward framework is a natural next direction.
- **Monocular and asynchronous capture.** Extending the system beyond synchronized multi-camera rigs would substantially broaden its applicability.

In summary, NoPo4D demonstrates that pose-free, feed-forward 4D reconstruction is not only feasible but competitive with methods that rely on known calibration and minutes of per-scene optimization. By grounding dynamic supervision in optical flow rather than rendered outputs, and by fusing multi-view information through a shared attention mechanism, the system opens a practical path toward scalable, real-time dynamic scene reconstruction from the increasingly common multi-camera capture hardware found in robotics, sports analytics, and mixed reality applications.

References

[1] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[2] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics*, 39(4), August 2020.

[3] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, volume 7577, pages 611–625, 2012.

[4] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction, December 2023. arXiv:2312.12337.

[5] Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, BING WANG, Guang Chen, et al. Feedforward 4d reconstruction for dynamic driving scenes using unposed images.

[6] 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, March 2024. arXiv:2403.14627.

[7] Zequn Chen, Jiezhi Yang, and Heng Yang. Pref3r: Pose-free feed-forward 3d gaussian splatting from variable-length image sequence. *arXiv preprint arXiv:2411.16877*, 2024.

[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. In *International Conference on Learning Representations (ICLR)*, 2021.

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.

[10] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4D-Rotor Gaussian Splatting: Towards Efficient Novel View Synthesis for Dynamic Scenes, July 2024. arXiv:2402.03307 [cs].

[11] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361. IEEE, 2012.

[13] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J. Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard

Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-Exo4D: Understanding Skilled Human Activity from First- and Third-Person Perspectives, September 2024. arXiv:2311.18259.

[14] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti, Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator, March 2022. arXiv:2203.03570.

[15] Jisang Han, Honggyu An, Jaewoo Jung, Takuya Narihira, Junyoung Seo, Kazumi Fukuda, Chaehyun Kim, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. D^2USt3R: Enhancing 3D Reconstruction with 4D Pointmaps for Dynamic Scenes, April 2025. arXiv:2504.06264.

[16] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. PF3plat: Pose-Free Feed-Forward 3D Gaussian Splatting, October 2024. arXiv:2410.22128.

[17] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, 1981.

[18] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4220–4230, 2024.

[19] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 668–685, 2022.

[20] Junhwa Hur, Charles Herrmann, Songyou Peng, Philipp Henzler, Zeyu Ma, Todd Zickler, and Deqing Sun. Ufo-4d: Unposed feedforward 4d reconstruction from two images. *arXiv preprint arXiv:2602.24290*, 2026.

[21] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017.

[22] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D Gaussian Splatting from Unconstrained Views, May 2025. arXiv:2505.23716.

[23] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*. IEEE, 2026.

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering, August 2023. arXiv:2308.04079.

[25] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic Gaussian Fusion from Casual Videos via 4D Motion Scaffolds, November 2024. arXiv:2405.17421.

[26] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding Image Matching in 3D with MASt3R, June 2024. arXiv:2406.09756 [cs].

[27] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5521–5531, 2022.

[28] Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. Movies: Motion-aware 4d dynamic view synthesis in one second. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2026.

[29] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the Visual Space from Any Views, November 2025. arXiv:2511.10647.

[30] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023.

[31] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. FlowNet3D: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–537, 2019.

[32] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, November 2017.

[33] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[34] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis, August 2023. arXiv:2308.09713 [cs].

[35] 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, August 2020. arXiv:2003.08934.

[36] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, April 2023. arXiv:2304.07193 [cs].

[37] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction, March 2021. arXiv:2103.13413.

[38] Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, and Huan Ling. L4GM: Large 4D Gaussian Reconstruction Model, June 2024. arXiv:2406.10324.

[39] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[40] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. VideoFlow: Exploiting temporal cues for multi-frame optical flow estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 12324–12333, 2023.

[41] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3R: Zero-shot Gaussian Splatting from Uncalibrated Image Pairs, August 2024. arXiv:2408.13912.

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

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.

[44] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419, 2020.

[45] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[46] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 722–729, 1999.

[47] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pages 78–89. IEEE, 2025.

[48] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer, March 2025. arXiv:2503.11651.

[49] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of Motion: 4D Reconstruction from a Single Video, July 2024. arXiv:2407.13764 [cs].

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

[51] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[52] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy, December 2023. arXiv:2312.14132.

[53] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

[54] Yihan Wang, Lahav Lipson, and Jia Deng. SEA-RAFT: Simple, Efficient, Accurate RAFT for Optical Flow, May 2024. arXiv:2405.14793.

[55] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[56] Zihan Wang, Jeff Tan, Tarasha Khurana, Neehar Peri, and Deva Ramanan. MonoFusion: Sparse-View 4D Reconstruction via Monocular Fusion, July 2025. arXiv:2507.23782 [cs].

[57] 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, July 2024. arXiv:2310.08528 [cs].

[58] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning optical flow via global matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8121–8130, 2022.

[59] Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4DGT: Learning a 4D Gaussian Transformer Using Real-World Monocular Videos, June 2025. arXiv:2506.08015 [cs].

[60] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D Reconstruction of 1000+ Images in One Forward Pass, January 2025. arXiv:2501.13928.

[61] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data, April 2024. arXiv:2401.10891.

[62] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2, June 2024. arXiv:2406.09414.

[63] Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. NeoVerse: Enhancing 4D World Model with in-the-wild Monocular Videos, January 2026. arXiv:2601.00393.

[64] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting, February 2024. arXiv:2310.10642 [cs].

[65] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction, November 2023. arXiv:2309.13101 [cs].

[66] Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting. *arXiv preprint arXiv:2511.07321*, 2025.

[67] 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, October 2024. arXiv:2410.24207.

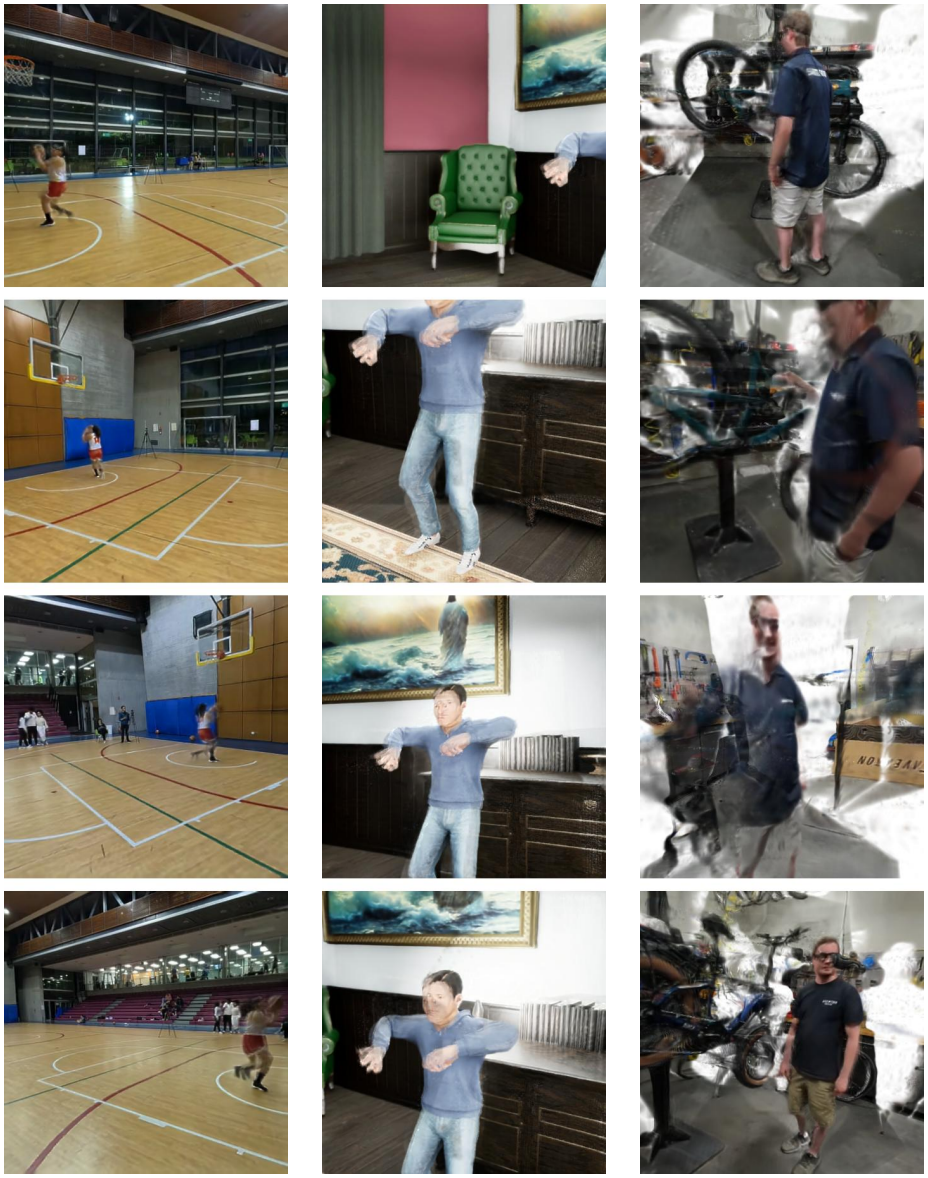
[68] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A Simple Approach for Estimating Geometry in the Presence of Motion, October 2024. arXiv:2410.03825.

[69] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[70] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. FLARE: Feed-forward Geometry, Appearance and Camera Estimation from Uncalibrated Sparse Views, February 2025. arXiv:2502.12138.

A. Failure Cases

Figure 11 documents three representative failure modes. In fast-moving regions such as rapidly swinging limbs or thrown objects, the pseudo-ground-truth optical flow from SEA-RAFT becomes unreliable, and the motion encoder receives incorrect supervision, producing cross-view Gaussian misalignments that manifest as semi-transparent floating artifacts around the moving region. When the camera rig itself moves, the static-rig assumption is violated, and averaging per-frame pose predictions collapses distinct camera positions into an incorrect single estimate, misaligning the unprojected Gaussian positions. Finally, scene regions that are visible from only a narrow range of input viewpoints may carry depth estimation errors from the monocular backbone, with no cross-view geometric constraint to correct them; at extreme novel viewpoints, these depth errors surface as floating Gaussian artifacts.



(a) Fast-moving regions. (b) Moving camera rig. (c) Far-from-input viewpoints.

Figure 11: **Failure cases.** (a) Cross-view misalignments and floating artifacts in fast-moving regions. (b) Degradation under camera motion, violating the static-rig assumption. (c) Floating Gaussian artifacts far from input camera viewpoints.

Abstract in lingua italiana

I recenti metodi feed-forward per il 3D Gaussian Splatting hanno compiuto progressi notevoli su singoli aspetti della ricostruzione di scene tridimensionali, ma nessun metodo esistente affronta congiuntamente contenuti dinamici, input multi-vista e pose delle telecamere sconosciute in un unico passaggio feed-forward. I metodi che gestiscono la dinamica richiedono pose accurate delle telecamere o accettano solo input monoculare; i metodi multi-vista senza pose affrontano unicamente scene statiche; i metodi di ottimizzazione per-scena colmano alcune di queste lacune, ma con un costo da minuti a ore per scena. Questa tesi introduce **NoPo4D**, il primo sistema feed-forward che affronta questo quadrante inesplorato. Basandosi su un backbone geometrico pre-addestrato e sui recenti framework 4D Gaussian, NoPo4D introduce una scomposizione della velocità che separa il moto delle Gaussiane in spostamenti nel piano immagine per pixel e variazioni di profondità, consentendo la supervisione diretta della componente 2D tramite flusso ottico pseudo-ground-truth. Questo approccio evita sia la dipendenza dal rendering differenziabile sia la necessità di annotazioni di moto 3D. Il sistema è completato da un encoder di moto bidirezionale per l'aggregazione di feature cross-vista e cross-frame, e da un'opacità dipendente dalla vista che mitiga i disallineamenti delle Gaussiane tra viste e istanti temporali diversi. Su quattro benchmark multi-vista dinamici, NoPo4D supera costantemente i baseline feed-forward precedenti e, con una fase opzionale di post-ottimizzazione, raggiunge o supera i metodi per-scena pur essendo significativamente più veloce.

Parole chiave: ricostruzione 3D gaussiana, ricostruzione feed-forward, scene dinamiche, ricostruzione senza posa, flusso ottico, video multi-vista, rendering neurale

Acknowledgements

I am deeply grateful to my co-advisor Dr. Sunghwan Hong, whose day-to-day mentorship, patience, and technical insight shaped every aspect of this work. His willingness to engage with problems at any hour and his constant push for rigour made this project what it is.

I am equally thankful to Prof. Matteo Matteucci, Prof. Marc Pollefeys, and Chenyangguang Zhang for their guidance, encouragement, and scientific vision.

I owe a great deal to my collaborator Yanik Künzi, who contributed essential ideas and effort at every stage of the project. More broadly, I want to thank the entire CVG team for making my time at ETH Zürich such a memorable experience: from the daily intellectual energy in the lab to the ski retreat, which I will not forget.

Finally, I thank my family and friends for their constant support throughout the long road of a master's thesis. This work would not have been possible without them.