DYNAMIC VISION OPTICAL FLOW-GUIDED GAUSSIAN SPLATTING FOR DYNAMIC SCENE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deformation-based dynamic Gaussian Splatting models every point of a monocular video with a single deformation field, whether the point is static, part of a rigid object, or part of a fluid. We found that such deformation fields cause rapidly moving objects to lose pattern details or become blurred, rather than exhibiting a clear motion trajectory. Existing event-based methods such as E-4DGS and Event-boosted D3DGS utilize events for photometric loss, which requires multiple rendering passes during training, whereas we use an event camera to decompose the scene. We estimate dynamic vision optical flow (DVOF) from events recorded by a dynamic vision sensor using contrast maximization, and derive an observability matrix from the contrast curvature to make the aperture problem explicit. Patches are grouped by whether a single affine motion explains them along their observable directions, and the groups are linked into tracks over time. This divides the scene into static, rigid, and free regions. Gaussians are assigned to the regions by rendering-weight voting, and consistency-checked spatial completion recovers event-sparse object interiors. Static Gaussians optimize time-invariant geometry and appearance, rigid ones follow explicit rigid trajectories supervised by the flow, and free ones keep a deformation field. On real RGB-event sequences, our method improves quality of full image and dynamic region over deformation-based and event-based methods, and improves the consistency of image-plane motion between rendered and observed sequences.

1 INTRODUCTION

Reconstructing dynamic scenes from monocular observations is a central problem in computer vision, with applications in immersive content creation and motion analysis. Dynamic neural radiance fields (Pumarola et al., 2021) and subsequent extensions of 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023; Yang et al., 2024b; Wu et al., 2024a) have substantially advanced this task. Yet reproducing a sequence of images does not necessarily recover the motion that produced it. This distinction is particularly consequential for a stationary monocular camera: without camera-induced parallax, the images provide limited constraints on how scene geometry and motion should be disentangled. We study this setting using RGB frames and a synchronized event stream, focusing on scenes that combine fast rigid motion with static backgrounds and nonrigid dynamics.

A common approach represents all scene motion with a shared deformation field that maps canonical Gaussians to each timestamp (Yang et al., 2024b; Wu et al., 2024a). Although expressive, this representation does not enforce that Gaussians belonging to the same rigid object move together. Image reconstruction losses constrain their aggregate rendered appearance, leaving ambiguity in the trajectories of individual primitives. Changes in Gaussian scale, orientation, and visibility can alter a texture's contribution to the image without transporting its supporting Gaussians consistently with the object. Therefore, the deformation-based baseline can lead to blurred or missing detail on fast moving objects. These observations motivate an explicit structural constraint: which primitives should move together, and which measurements constrain their shared motion?

Dynamic vision sensors (also called event cameras) provide temporally dense evidence for this purpose. They asynchronously record changes in log brightness (Gallego et al., 2020), which can serve as a constraint on object motion between RGB frames. Existing event-based dynamic reconstruction methods use this evidence in different ways. E-4DGS (Feng et al., 2025) and Event-boosted

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

D3DGS (Xu et al., 2025) impose brightness-change consistency through event formation losses, which require image rendering at event-supervision timestamps. DEGS (He et al., 2025) adapts a pretrained event optical flow network to each scene to supervise Gaussian deformation. Our approach uses event motion to determine which Gaussians should share a motion model, together with the directions in which that motion is constrained by the observations.

We first estimate patchwise image-plane velocities, termed dynamic vision optical flow (DVOF), by Contrast Maximization (Gallego et al., 2018). This estimation operates directly on the observed events and requires neither a pretrained flow network nor external flow training data. A velocity vector alone, however, conceals the aperture problem: events along a straight edge constrain motion normal to that edge but leave tangential motion ambiguous. We therefore associate each estimate with an observability matrix derived from the local curvature of the contrast objective. Its eigenstructure identifies strongly and weakly constrained velocity directions. We group patches by whether a shared affine motion field explains their velocities along the observable directions. This accommodates spatially varying velocities, such as those of a rotating object, while pooling complementary constraints across patches. When their spatial arrangement and edge orientations provide sufficient information, the pooled measurements resolve ambiguities that no individual patch can resolve.

We link these groups across time and use their pooled information to select a supported motion model for each track. A track receives a rotation-enabled shared transform when its affine information score passes the threshold; otherwise, it receives a translation-only transform if its translation information score passes. This makes short but well-observed motion usable without treating image-space affine consistency as a certificate of physical rigidity. Together with static-region detection, this yields static, shared-motion, and freely deforming regions over the time intervals.

After an initial reconstruction stage, temporally local rendering-weight voting transfers this decomposition to the Gaussians according to their contributions while each track is active. For a short-lived object that is underrepresented by the initial reconstruction, track-guided seeding adds Gaussians at high-residual pixels and associates them with the same motion and visibility interval. Static Gaussians have time-invariant geometry, free Gaussians use a deformation field conditioned on track embeddings, and each selected rigid group follows a shared rigid transform supervised by the event-derived flow. The rigid branch preserves relative geometry and ties projected Gaussian displacements to measured motion. Event motion is estimated once before reconstruction and reused during training, so our event supervision operates on projected Gaussian displacements without an event photometric rendering loss.

Our contributions are:

- We introduce an observability-aware motion analysis method that groups incomplete event-derived velocity measurements through shared affine fields and uses their pooled measurement support to assign different motion models to temporal tracks.
- We develop a motion-grouped Gaussian representation that transfers image-space tracks through temporally local rendering-weight voting and applies shared, event-supervised transforms over their supported intervals, alongside static and freely deforming branches.
- We introduce a track-guided Gaussian seeding mechanism that supplements underrepresented transient regions using training-image residuals and associates the added primitives with the track's motion and temporal visibility.

2 Related Work

Dynamic scenes via Gaussian Splatting. 3D Gaussian Splatting (Kerbl et al., 2023) is extended to dynamic scenes either by deforming a canonical set of Gaussians with a time-dependent field (Yang et al., 2024b; Wu et al., 2024a), or by jointly representing space and time with native 4D Gaussian primitives (Yang et al., 2024a). Both families let one representation account for every motion in the scene. A growing number of methods instead treat parts of the scene differently: static and dynamic primitives are separated by a learned indicator (Wu et al., 2025), by the expressiveness of two competing branches (Wang et al., 2025), or for the purpose of editing (Kwon et al., 2025), and motion is factored into global and local components (Kwak et al., 2025) or predicted at sparse anchors (Chen et al., 2026). Our decomposition is driven by events and assigns each region a motion

model. We also follow Liang et al. (2025), who report that monocular dynamic Gaussian Splatting is brittle, and evaluate every configuration over repeated runs.

Flow estimated from the frames has been used to guide dynamic Gaussians. Zhu et al. (2024) decouple optical flow into camera flow and motion flow and constrain the deformation with the latter, and Guo et al. (2024) augment several dynamic 3DGS paradigms with flow cues and an uncertainty-aware loss. Such priors inherit the temporal resolution of the frames, specifically, between two frames, the optical flow network has nothing to observe. The flow we use is estimated from events, at a temporal resolution independent of the frame rate, and it carries a per-patch statement of which of its components are observable at all.

Event-based reconstruction. Event cameras (Gallego et al., 2020) have been combined with radiance fields (Rudnev et al., 2023; Klenk et al., 2023; Qi et al., 2023) and with Gaussian Splatting (Wu et al., 2024b; Xiong et al., 2024; Zhang et al., 2025). For dynamic scene reconstruction, E-4DGS (Feng et al., 2025) and Event-boosted D3DGS (Xu et al., 2025) use event formation losses to enforce brightness-change consistency. The latter also separates dynamic and static regions using RGB feature discrepancies and lifts the resulting masks through rendered depth. DEGS (He et al., 2025) adapts a pretrained event flow network (Zhu et al., 2018) to each scene using Contrast Maximization, associates events with Gaussians through rendered depth, and uses the estimated motion to supervise deformation. Our motion estimator directly optimizes patchwise event alignment without a pretrained flow network or external flow training data.

Motion via Event-based optical flow. Estimating motion from events is a mature topic, including gradient-fitting and plane-fitting methods (Benosman et al., 2014), block matching (Liu & Delbruck, 2017), and Contrast Maximization (Gallego et al., 2018; Shiba et al., 2022). The aperture problem is known to affect event flow, and Akolkar et al. (2020) make their estimator robust to it. We instead quantify it per patch through the curvature of the contrast, and use that quantity to decide which patches move together. Events have also been segmented into several motions by fitting one motion model per cluster (Stoffregen et al., 2019), including layered affine-flow models (Farah et al., 2025). Our grouping differs in that it fits shared fields to patch-level measurements only along their observable directions, carries the resulting support into model assignment, and exists to reconstruct dynamic 3D scenes rather than only label events. Layered affine motion models have a long history in video (Wang & Adelson, 1994); we revisit them on event stream, weighted by observability.

3 METHOD

In this section, we first briefly review Deformable 3D Gaussians (D-3DGS) (Yang et al., 2024b), which is our baseline. Then we introduce our approach (Fig. 1).

3.1 PRELIMINARIES: DEFORMABLE GAUSSIAN SPLATTING

A scene is represented by a set of anisotropic Gaussians $\{\mathcal{G}_i\}_{i=1}^{N}$ in canonical space (Kerbl et al., 2023), each with mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$, rotation $\boldsymbol{q}_i \in \mathbb{S}^3$, scale $\boldsymbol{s}_i \in \mathbb{R}^3_+$, opacity o_i, and color $\boldsymbol{c}_i$. Deformation-based methods map each Gaussian to time t through a shared field $\mathcal{D}_\theta$,

$$(\delta\boldsymbol{\mu}_i, \delta\boldsymbol{q}_i, \delta\boldsymbol{s}_i) = \mathcal{D}_\theta(\boldsymbol{\mu}_i, t), \tag{1}$$

and render the deformed Gaussians by front-to-back α-blending,

$$\hat{C}(\boldsymbol{p}, t) = \sum_i \boldsymbol{c}_i\, \alpha_i(\boldsymbol{p}, t) \prod_{j<i} (1 - \alpha_j(\boldsymbol{p}, t)) \;\triangleq\; \sum_i \boldsymbol{c}_i w_i(\boldsymbol{p}, t), \tag{2}$$

where w_i is the contribution of Gaussian i to pixel $\boldsymbol{p}$.

3.2 DYNAMIC VISION OPTICAL FLOW

We divide the image into patches Ω_j of side P and the event stream into windows centered at t_w with duration Δ. For a velocity $\boldsymbol{v}$ in pixels per window, an event $e_k = (\boldsymbol{x}_k, t_k, p_k)$ is warped to the window center as

$$\boldsymbol{x}'_k(\boldsymbol{v}) = \boldsymbol{x}_k - \tau_k \boldsymbol{v}, \qquad \tau_k = (t_k - t_w)/\Delta. \tag{3}$$

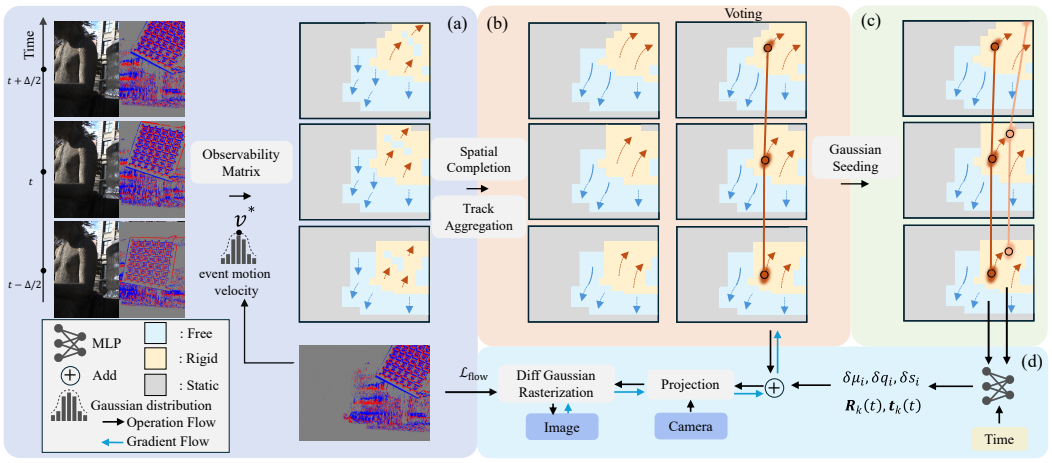


Figure 1: Overview of our approach. (a) We estimate dynamic vision optical flow from the event stream and group patches that admit a shared affine motion explanation. (b) Spatial completion extends track labels into event-sparse interiors. Track aggregation pools measurement support along each track to select a shared motion model, including for short tracks with sufficient observations. (c) Track-guided seeding adds Gaussians where the initial reconstruction underrepresents a tracked region. (d) Gaussians assigned to different regions are optimized with static, shared-rigid, or freely deforming branches.

Following Contrast Maximization (CMax) (Gallego et al., 2018), we accumulate the warped events into an unsigned image $I_{jw}(\cdot; \boldsymbol{v})$ and estimate

$$f_{jw}(\boldsymbol{v}) = \frac{\|I_{jw}(\cdot;\boldsymbol{v})\|_2^2}{\|I_{jw}(\cdot;\boldsymbol{v})\|_1^2}, \qquad \boldsymbol{v}_{jw}^\star \approx \arg\max_{\boldsymbol{v}} f_{jw}(\boldsymbol{v}). \tag{4}$$

A velocity estimate alone does not say how much of it the events actually determine. We characterize this by the curvature of the contrast at its maximum and define the observability matrix

$$\boldsymbol{H}_{jw} = \Pi_{\succeq 0}\big(-\nabla_{\boldsymbol{v}}^2 f_{jw}(\boldsymbol{v}_{jw}^\star)\big)\,, \qquad \hat{\boldsymbol{H}}_{jw} = \frac{\boldsymbol{H}_{jw}}{\lambda_{\max}(\boldsymbol{H}_{jw})}, \tag{5}$$

where $\Pi_{\succeq 0}$ projects onto the positive semi-definite cone by clipping negative eigenvalues. When $\lambda_{\max}(\boldsymbol{H}_{jw}) = 0$, $\hat{\boldsymbol{H}}_{jw} = \boldsymbol{0}$. Its dominant eigenvector is the velocity direction to which event alignment is most sensitive. For a straight edge, $\boldsymbol{H}$ is rank one and constrains only the normal-flow component; the derivation is given in App.E.

We compare a measured velocity $\boldsymbol{v}$ with a prediction $\boldsymbol{v}'$ using

$$d_{\hat{\boldsymbol{H}}}(\boldsymbol{v}, \boldsymbol{v}')^2 = (\boldsymbol{v} - \boldsymbol{v}')^\top \hat{\boldsymbol{H}} (\boldsymbol{v} - \boldsymbol{v}'). \tag{6}$$

Thus, disagreement is penalized along event-constrained directions and ignored along the null space. We retain patch-windows with sufficient event support and a clear gain from motion compensation. CMax optimization, curvature estimation, and screening details are provided in App. A.

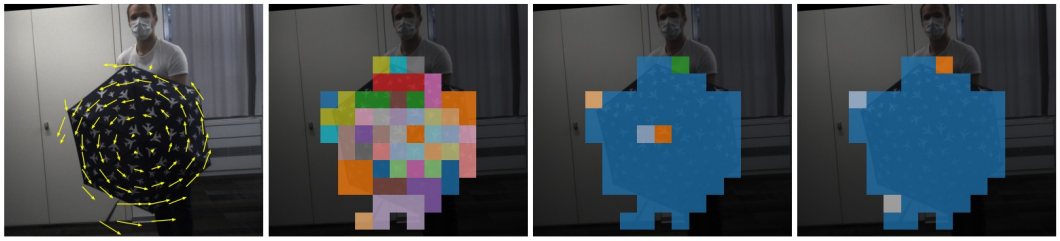
3.3 Observability-Aware Motion Grouping

Velocity similarity fragments a rotating or expanding object because its patches move in different directions. We instead ask whether their measurements admit one shared affine motion field,

$$\boldsymbol{v}(\boldsymbol{z};\boldsymbol{\theta}) = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b} = \boldsymbol{J}(\boldsymbol{z})\boldsymbol{\theta}, \qquad \boldsymbol{J}(\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{z}^\top & 1 & \boldsymbol{0}^\top & 0 \\ \boldsymbol{0}^\top & 0 & \boldsymbol{z}^\top & 1 \end{bmatrix}, \tag{7}$$

where $\boldsymbol{z} = \boldsymbol{x}/P$ is measured in patch units and $\boldsymbol{\theta} \in \mathbb{R}^6$. For a patch set $\mathcal{S}$, we fit the field in coordinates centered at $\bar{\boldsymbol{z}}_{\mathcal{S}}$,

$$\min_{\boldsymbol{A},\boldsymbol{b}_c} \sum_{j\in\mathcal{S}} d_{\hat{\boldsymbol{H}}_j}\big(\boldsymbol{v}_j^\star, \boldsymbol{A}(\boldsymbol{z}_j - \bar{\boldsymbol{z}}_{\mathcal{S}}) + \boldsymbol{b}_c\big)^2 + \gamma\|\boldsymbol{A}\|_F^2 + \eta\|\boldsymbol{b}_c\|_2^2. \tag{8}$$



(a) Dynamic vision optical flow (b) Velocity similarity (c) + Affine consistency & $\tilde{H}$ (d) + Spatial completion

Figure 2: Observability-aware motion grouping. Patchwise CMax provides velocities and their directionally constrained components. Grouping by velocity similarity fragments a rotating object, whereas a shared affine explanation aggregates patches with different velocities. Spatial completion recovers interiors that generate few events.

The fit pools complementary edge constraints across patches. A patch joins a region when the fitted field explains its measured velocity in the directions selected by $\hat{\boldsymbol{H}}_j$,

$$d_{\hat{\boldsymbol{H}}_j}\left(\boldsymbol{v}_j^\star, \boldsymbol{J}(\boldsymbol{z}_j)\hat{\boldsymbol{\theta}}\right) < \max\left(\epsilon_{\text{abs}}, \epsilon_{\text{rel}}\|\boldsymbol{J}(\boldsymbol{z}_j)\hat{\boldsymbol{\theta}}\|_2\right). \tag{9}$$

We grow spatially connected regions under Eq. 9, refitting the shared field after each expansion. Regions in adjacent windows are linked when they overlap spatially and their fields predict similar motion over the union of their supports, producing motion tracks $\mathcal{T}_k$.

Track Aggregation. A short track can contain reliable motion even when it occupies only a small fraction of the sequence. We therefore consider tracks observed for at least $K_{\min}$ windows and select the model whose parameters are supported by their measurements. For a region $\mathcal{S}$ in one window, let $\tilde{\boldsymbol{z}}_j$ denote its patch coordinate relative to a local reference. We form the unregularized information matrices

$$\boldsymbol{M}_{\mathcal{S}}^{(6)} = \sum_{j\in\mathcal{S}} \boldsymbol{J}(\tilde{\boldsymbol{z}}_j)^\top \hat{\boldsymbol{H}}_j \boldsymbol{J}(\tilde{\boldsymbol{z}}_j), \qquad \boldsymbol{M}_{\mathcal{S}}^{(2)} = \sum_{j\in\mathcal{S}} \hat{\boldsymbol{H}}_j. \tag{10}$$

The first tests support for a six-parameter affine explanation; the second tests whether a shared translation is constrained. We aggregate the minimum eigenvalue of each matrix across the regions of a track. If the affine score exceeds λ_0, the track receives a rotation-enabled shared transform. Otherwise, if the translation score exceeds λ_0, it receives a translation-only transform. Remaining tracks use the free-deformation branch.

3.4 Gaussian Assignment and Track-Guided Seeding

The motion grouping yields per-window track labels. We combine low event activity with low temporal RGB variation to identify static candidates, and spatially complete eligible long-lived tracks where object interiors contain too few events. Let $\mathcal{K}_w$ be the set of track labels present in window w. The pixel label map takes values $\ell_w(\boldsymbol{p}) \in \mathcal{K}_w \cup \{\text{st}, \text{ne}, \text{unrel}\}$, denoting tracked pixels, static pixels, dynamic pixels with insufficient events, and remaining dynamic pixels without a reliable track assignment, respectively. Completion details are given in App. A.

Temporally local rendering-weight voting. After an initial reconstruction stage, we assign its Gaussians using their actual rendering contributions. Let $\mathcal{W}_k = \{w : k \in \mathcal{K}_w\}$ be the windows in which track k is active, and define the dynamic voting support as the pixel set $\mathcal{D}_w = \{\boldsymbol{p} : \ell_w(\boldsymbol{p}) \in \mathcal{K}_w \cup \{\text{unrel}\}\}$. Static and event-insufficient pixels are excluded from this set. The track vote for Gaussian i and the corresponding dynamic support are

$$V_{ik} = \sum_{w\in\mathcal{W}_k} \sum_{\boldsymbol{p}:\ell_w(\boldsymbol{p})=k} w_i(\boldsymbol{p}, t_w), \qquad D_{ik} = \sum_{w\in\mathcal{W}_k} \sum_{\boldsymbol{p}\in\mathcal{D}_w} w_i(\boldsymbol{p}, t_w), \tag{11}$$

We compare V_{ik}/D_{ik} only for tracks with $D_{ik} > 0$ and assign i to the track with the largest ratio when it is at least the assignment threshold. If no track qualifies, sequence-level votes determine

its static or free-branch assignment (App. A). Restricting both terms to $\mathcal{W}_k$ prevents a short-lived object's evidence from being diluted by windows in which it is absent. Votes are evaluated efficiently through masked rendering gradients.

Track-guided Seeding. Assignment can only reorganize existing primitives. A short-lived object occupying few training pixels may still have too few Gaussians to represent its appearance, even after its motion has been identified. For each selected track, we choose the window with its largest spatial support and measure the reconstruction residual inside its mask. We seed only when this residual is substantially higher than the image average. New Gaussians are placed at the highest residual pixels, back-projected using a foreground depth surrogate derived from the reconstruction immediately before the track appears. Their colors come from the nearest training image, their scales follow the projected sampling spacing, and they inherit the track identity.

Piecewise motion and visibility. Let $[a_k, b_k]$ be the interval covered by track k, and let $g_k(t)$ equal one inside this interval and decay smoothly to zero across a short boundary margin. Within the supported interval, assigned Gaussians follow T_k; outside it, their geometry returns to the free branch. For configurations using track-guided seeding, their opacity is also multiplied by $g_k(t)$. The same visibility schedule is applied to the assigned group so that primitives introduced for a transient observation do not remain visible throughout the sequence.

3.5 Motion-Grouped Deformation

For static Gaussians, $\delta\boldsymbol{\mu}_i = \delta\boldsymbol{q}_i = \delta\boldsymbol{s}_i = \mathbf{0}$ for $i \in \mathcal{G}_{\mathrm{st}}$, while their appearance and geometry are still optimized. Free Gaussians keep the deformation field of Eq. 1, conditioned on a learned embedding $\boldsymbol{e}_{k(i)}$ of their track so that different motions are not forced through the same function of position, $(\delta\boldsymbol{\mu}_i, \delta\boldsymbol{q}_i, \delta\boldsymbol{s}_i) = \mathcal{D}_\theta(\boldsymbol{\mu}_i, t, \boldsymbol{e}_{k(i)})$. The Gaussians of a rigid track $\mathcal{T}_k$ follow a time-varying rigid transform about the track centroid $\boldsymbol{c}_k$,

$$\boldsymbol{\mu}_i(t) = \boldsymbol{R}_k(t)\,(\boldsymbol{\mu}_i - \boldsymbol{c}_k) + \boldsymbol{c}_k + \boldsymbol{t}_k(t), \qquad \boldsymbol{q}_i(t) = \boldsymbol{q}_{\boldsymbol{R}_k(t)} \otimes \boldsymbol{q}_i, \qquad \boldsymbol{s}_i(t) = \boldsymbol{s}_i, \tag{12}$$

where $(\boldsymbol{R}_k(t), \boldsymbol{t}_k(t))$ is produced by a small network Φ_ψ of time that outputs an axis-angle vector and a translation for every track. During intervals in which the shared rigid branch is fully active, Eq. 12 preserves the relative geometry of the assigned Gaussians: their pairwise distances, relative orientations, and scales remain constant over time. This geometric constraint does not fix their image-space visibility, which can change through projection, occlusion, and the opacity gate.

Event-motion supervision. Let $t_w^{\pm} = t_w \pm \Delta/2$ and let π be the calibrated projection. For an active track k in window w, let $\mathcal{I}_{kw}$ be the set of sampled Gaussians assigned to it and define their projected midpoints as $\bar{\boldsymbol{p}}_{ikw} = \frac{1}{2}[\pi(T_k(\boldsymbol{\mu}_i, t_w^+)) + \pi(T_k(\boldsymbol{\mu}_i, t_w^-))]$. We evaluate the fitted affine target at

$$\boldsymbol{p}_{ikw}^{\mathrm{eval}} = \begin{cases} \bar{\boldsymbol{p}}_{ikw}, & \text{rotation-enabled track,} \\ \dfrac{1}{|\mathcal{I}_{kw}|} \displaystyle\sum_{j \in \mathcal{I}_{kw}} \bar{\boldsymbol{p}}_{jkw}, & \text{translation-only track.} \end{cases} \tag{13}$$

Thus, rotation-enabled tracks use spatially varying targets, whereas translation-only tracks share the target evaluated at the mean projected midpoint. We minimize

$$\mathcal{L}_{\mathrm{flow}} = \mathbb{E}_{(k,w,i)} \left\| \pi\big(T_k(\boldsymbol{\mu}_i, t_w^+)\big) - \pi\big(T_k(\boldsymbol{\mu}_i, t_w^-)\big) - \boldsymbol{J}(\boldsymbol{p}_{ikw}^{\mathrm{eval}}/P)\hat{\boldsymbol{\theta}}_{kw} \right\|_2, \tag{14}$$

where the expectation is over active track-window pairs and their sampled Gaussians $i \in \mathcal{I}_{kw}$. The target is predicted by the shared affine model fitted to directionally constrained patch measurements. We supervise its full two-dimensional prediction; in weakly constrained windows, this target also reflects the shared-model assumption and fitting regularization.

Training begins with the deformation-based backbone. After the densification stage, we perform temporally local assignment, add track-guided seeds, initialize the shared transforms with Eq. 14, and continue optimizing

$$\mathcal{L} = (1 - \lambda_{\mathrm{s}})\mathcal{L}_1 + \lambda_{\mathrm{s}}\mathcal{L}_{\text{D-SSIM}} + \lambda_{\mathrm{flow}}\mathcal{L}_{\mathrm{flow}}. \tag{15}$$

The RGB terms refine appearance and absolute pose, while the event term constrains motion.

Table 1: Reconstruction quality averaged equally over ten scenes. Entries are scene-level means over repeated runs. Best and second-best results are shown in **bold** and underlined, respectively.

Method	Full image			Dynamic region		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
D-3DGS	29.43	0.8499	0.2606	23.41	0.8494	0.2152
MotionGS	29.68	0.8611	0.2376	23.57	0.8452	0.2128
DeGauss	28.83	0.8486	0.2330	23.35	0.8371	0.1991
E-D3DGS	30.38	0.8626	0.2372	24.76	0.8520	0.2092
Ours w/ optical flow	30.35	0.8576	0.2591	24.92	0.8552	0.2125
Ours w/ event flow	**30.95**	**0.8649**	**0.2290**	**26.67**	**0.8996**	**0.1608**

4 Experiments

4.1 Setup

Datasets. We evaluate on ten sequences from HS-ERGB (Tulyakov et al., 2021) and CED (Scheerlinck et al., 2019). HS-ERGB covers rapid rotation, translation, and irregular motion; we add the CED jogging sequence to include articulated human motion. Both datasets provide real RGB frames and synchronized events. Because the original sequences contain long nearly static portions, we select one event-dense interval per scene. The evaluated intervals are listed in App. B.

Metrics. We report PSNR, SSIM, and LPIPS with the VGG backbone for both the full image and the dynamic region. For each test frame, the dynamic mask is the event-dense patch mask from the nearest event window and is independent of the reconstructed motion groups. For SSIM and LPIPS, we crop to the mask's bounding box and replace pixels outside the mask with ground truth before evaluation. Every method is evaluated with the same masks and code.

Motion evaluation. We complement reconstruction metrics with rendered-flow endpoint error (EPE), which measures whether the reconstructed sequence exhibits the same image-plane motion as the observations. App. C gives the full protocol.

Baselines. We compare against the deformation-based backbone D-3DGS (Yang et al., 2024b), the event-based method Event-boosted D3DGS (E-D3DGS) (Xu et al., 2025), which uses the same events as we do, the flow-guided MotionGS (Zhu et al., 2024), whose motion prior comes from RGB optical flow, and DeGauss (Wang et al., 2025), which learns an unsupervised dynamic-static decomposition. To isolate the source of motion evidence, Ours w/ optical flow replaces our event flow with optical flow estimated from the training RGB frames by GMFlow (Xu et al., 2022).

4.2 Comparisons

Quantitative average results are presented in Tab. 1, with full-image and dynamic-region metrics. Across individual scenes, Ours with event flow ranks first on dynamic PSNR and SSIM in eight of ten scenes, while the plate sequence favors DeGauss. As shown in Fig. 3, our method is capable of reconstructing rigidly moving objects in scenes involving rapid motion.

All runs use one GeForce RTX 4070 Ti Super with 16 GB of VRAM and 30,000 main training iterations. For our method using event flow, the reported time includes event preprocessing and the additional 2,000-step initialization of shared transforms when such tracks are selected. Tab. 2 summarizes rendered-flow EPE and efficiency. Our method with event flow attains the lowest mean EPE, indicating improved agreement between motion estimated from rendered and observed image sequences. Per-scene efficiency is reported in App. D.

Tab. 3 evaluates the grouping criteria on the same event flow over all temporal windows. Affine consistency reduces the median number of groups in Umbrella from 37 to 7. Adding observability weighting increases track persistence in all three scenes. Spatial completion further increases

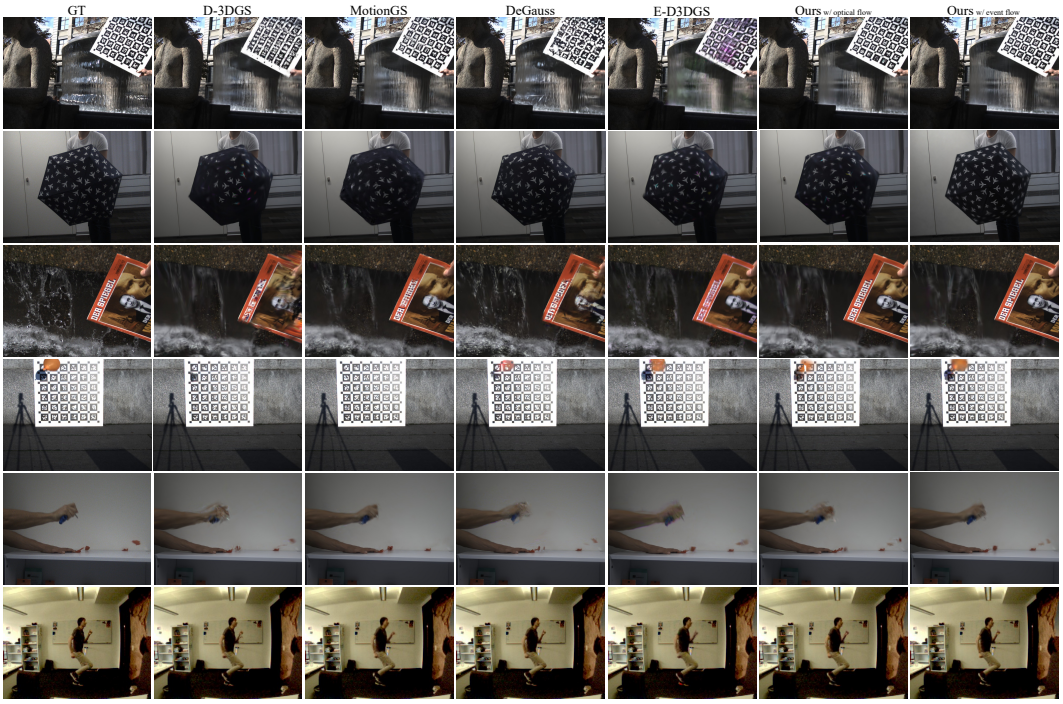


Figure 3: Qualitative comparison of different methods.

Table 2: Motion fidelity and efficiency. Rendered-flow EPE and efficiency are averaged over the scenes shared by all evaluated methods.

Method	EPE ↓	Time ↓	Gaussians (K) ↓	FPS ↑
D-3DGS	7.797	**19:09**	**73.2**	**116.2**
MotionGS	8.791	1:43:02	206.0	45.5
DeGauss	7.324	1:39:11	172.6	53.6
E-D3DGS	7.545	1:04:01	313.4	<u>110.3</u>
Ours w/ optical flow	<u>7.243</u>	24:12	85.2	100.2
Ours w/ event flow	**6.244**	<u>20:57</u>	<u>75.3</u>	108.0

coverage without changing the reported persistence. The reference region is produced by the full method, so its coverage is 100% by construction; these results measure grouping consistency rather than accuracy against independently annotated objects.

4.3 Ablations

Tab. 4 compares the full method with variants that remove motion grouping, spatial completion, or track-guided seeding, and with the RGB optical flow variant. The full method achieves the best average results across all six reconstruction metrics. Removing motion grouping causes the largest reduction in dynamic-region PSNR, from 26.67 to 23.38 dB, with dynamic SSIM falling from 0.8996 to 0.8301. Its dynamic PSNR is also close to that of D-3DGS (23.41 dB), supporting the importance of organizing the representation by motion.

Spatial completion and track-guided seeding address different gaps in the representation. Without spatial completion, dynamic PSNR drops by 2.32 dB. This is consistent with the role of completion in extending track assignments into event-sparse object interiors. Without seeding, dynamic PSNR decreases to 25.56 dB and dynamic LPIPS increases from 0.1608 to 0.2079. The gain from the full configuration supports supplementing the initial representation in underrepresented tracked regions,

Table 3: Grouping ablation on all temporal windows, evaluated relative to a reference region defined by the full method. Groups: number of groups overlapping the reference region (median over windows); Cover: fraction of that region covered by the largest group; Persist: length of the longest track relative to the number of active windows; Shared: whether a track is selected for a shared transform.

Scene	Grouping	Groups ↓	Cover ↑	Persist ↑	Shared
Umbrella	Velocity similarity	37	7%	33%	×
	+ Affine consistency	7	84%	67%	✓
	+ Observability $\boldsymbol{H}$	4	90%	87%	✓
	+ Spatial completion	1	100%	87%	✓
Schaff	Velocity similarity	3	85%	39%	×
	+ Affine consistency	3	84%	45%	×
	+ Observability $\boldsymbol{H}$	1	95%	82%	✓
	+ Spatial completion	1	100%	82%	✓
Bellevue	Velocity similarity	5.5	49%	39%	×
	+ Affine consistency	3	86%	28%	×
	+ Observability $\boldsymbol{H}$	1	95%	52%	✓
	+ Spatial completion	1	100%	52%	✓

Table 4: Ablation Study of key components of our method.

Method	Full image			Dynamic region		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
D-3DGS	29.43	0.8499	0.2606	23.41	0.8494	0.2152
Ours w/o motion grouping	29.27	0.8541	0.2599	23.38	0.8301	0.2292
Ours w/o spatial completion	30.31	0.8556	0.2608	24.35	0.8448	0.2153
Ours w/o seeding	29.88	0.8569	<u>0.2414</u>	<u>25.56</u>	<u>0.8698</u>	<u>0.2079</u>
Ours w/ optical flow	<u>30.35</u>	<u>0.8576</u>	0.2591	24.92	0.8552	0.2125
Ours w/ event flow	**30.95**	**0.8649**	**0.2290**	**26.67**	**0.8996**	**0.1608**

together with the associated temporal visibility schedule. Replacing event-derived flow with RGB optical flow reduces dynamic PSNR by 1.75 dB. The full-image PSNR difference is smaller, at 0.60 dB, indicating that the benefit is concentrated in the dynamic regions. This comparison supports the use of event-derived motion in our pipeline; the two variants also differ in their flow reliability and observability estimates, as detailed in App. A.

4.4 Limitations

Our method does not achieve optimal performance in the "candle" scenario, as the presence of two moving point light sources introduces complex noise into the event stream. In the "plate" scenario, the axis of rotation for the vertically rotating plate is parallel to the image plane, resulting in an 88% loss in the rigid track.

5 Conclusion

We presented a dynamic Gaussian reconstruction method that uses event-derived flow and directional observability to group motion and assign static, rigid, or freely deforming models. Spatial completion and track-guided seeding supplement event-sparse and underrepresented regions. On real monocular RGB-event sequences, our method improves average reconstruction quality and rendered motion consistency without an event photometric rendering loss.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and language polishing. All AI-assisted text was carefully reviewed and revised by the authors. LLM-generated code was verified and tested for correctness by two authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our work is designed to facilitate reproducibility. The proposed optimization scheme and its implementation details, including the optimization procedure and key training settings, are described in the main paper and supplementary material. Detailed descriptions of the datasets, data preprocessing, and experimental protocols are also provided in the main paper and supplementary material. The source code will be publicly released upon acceptance to facilitate the reproduction and further evaluation of the reported results.

REFERENCES

Himanshu Akolkar, Sio-Hoi Ieng, and Ryad Benosman. Real-time high speed motion prediction using fast aperture-robust event-driven visual flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):361–372, 2020.

Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2), 2014.

Jianing Chen, Zehao Li, Yujun Cai, Hao Jiang, Chengxuan Qian, Juyuan Kang, Shuqin Gao, Honglong Zhao, Tianlu Mao, and Yucheng Zhang. Haif-gs: Hierarchical and induced flow-guided gaussian splatting for dynamic scene. *Advances in Neural Information Processing Systems*, 38: 125539–125563, 2026.

Youssef Farah, Federico Paredes-Valles, Guido De Croon, Muhammad Ahmed Humais, Hussain Sajwani, and Yahya Zweiri. EV-LayerSegNet: Self-supervised motion segmentation using event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5165–5174, 2025.

Chaoran Feng, Zhenyu Tang, Wangbo Yu, Yatian Pang, Yian Zhao, Jianbin Zhao, Li Yuan, and Yonghong Tian. E-4dgs: High-fidelity dynamic reconstruction from the multi-view event cameras. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 7356–7365, 2025.

Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3867–3876. IEEE, 2018.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

Zhiyang Guo, Wengang Zhou, Li Li, Min Wang, and Houqiang Li. Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(4):3119–3133, 2024.

Junhao He, Jiaxu Wang, Jia Li, Mingyuan Sun, Qiang Zhang, Jiahang Cao, Ziyi Zhang, Yi Gu, Jingkai Sun, and Renjing Xu. Degs: Deformable event-based 3d gaussian splatting from rgb and event stream. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

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

Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers. E-nerf: Neural radiance fields from a moving event camera. *IEEE Robotics and Automation Letters*, 8(3):1587–1594, 2023.

Sangwoon Kwak, Joonsoo Kim, Jun Young Jeong, Won-Sik Cheong, Jihyong Oh, and Munchurl Kim. Modec-gs: Global-to-local motion decomposition and temporal interval adjustment for compact dynamic 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11338–11348, 2025.

Joohyun Kwon, Hanbyel Cho, and Junmo Kim. Instruct-4DGS: Efficient dynamic scene editing via 4d gaussian-based static-dynamic separation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Yiqing Liang, Mikhail Okunev, Mikaela Angelina Uy, Runfeng Li, Leonidas Guibas, James Tompkin, and Adam W Harley. Monocular dynamic gaussian splatting: Fast, brittle, and scene complexity rules. *Transactions on Machine Learning Research (TMLR)*, 2025.

Min Liu and Tobi Delbruck. Block-matching optical flow for dynamic vision sensors: Algorithm and FPGA implementation. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10318–10327, 2021.

Yunshan Qi, Lin Zhu, Yu Zhang, and Jia Li. E2nerf: Event enhanced neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13254–13264, 2023.

Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. Eventnerf: Neural radiance fields from a single colour event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4992–5002, 2023.

Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow. In *European Conference on Computer Vision (ECCV)*, 2022.

Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16155–16164, 2021.

John Y. A. Wang and Edward H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5), 1994.

Rui Wang, Quentin Lohmeyer, Mirko Meboldt, and Siyu Tang. Degauss: Dynamic-static decomposition with gaussian splatting for distractor-free 3d reconstruction. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6294–6303. IEEE, 2025.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320, June 2024a.

Jiahao Wu, Rui Peng, Zhiyan Wang, Lu Xiao, Luyang Tang, Jinbo Yan, Kaiqiang Xiong, and Ronggang Wang. Swift4D: Adaptive divide-and-conquer gaussian splatting for compact and efficient reconstruction of dynamic scene. In *International Conference on Learning Representations (ICLR)*, 2025.

Jingqian Wu, Shuo Zhu, Chutian Wang, and Edmund Y Lam. Ev-gs: Event-based gaussian splatting for efficient and accurate radiance field rendering. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2024b.

Tianyi Xiong, Jiayi Wu, Botao He, Cornelia Fermuller, Yiannis Aloimonos, Heng Huang, and Christopher Metzler. Event3dgs: Event-based 3d gaussian splatting for high-speed robot egomotion. In *8th Annual Conference on Robot Learning*, 2024.

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

Wenhao Xu, Wenming Weng, Yueyi Zhang, Ruikang Xu, and Zhiwei Xiong. Event-boosted deformable 3d gaussians for dynamic scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28334–28343, 2025.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations*, volume 2024, pp. 9142–9159, 2024a.

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

Zixin Zhang, Kanghao Chen, and Lin Wang. Elite-evgs: Learning event-based 3d gaussian splatting by distilling event-to-video priors. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13972–13978. IEEE, 2025.

Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.062.

Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. MotionGS: Exploring explicit motion guidance for deformable 3d gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

A Implementation Details

Event alignment. The default window duration is 20 ms. Patch width is $P = \max(32, 8\ \mathrm{round}(W/(15 \cdot 8)))$ for image width W, and a patch-window must contain at least 600 events, from which 600 are sampled. Events are bilinearly accumulated on a canvas padded by $\lfloor v_{\max} \rfloor + 24$ pixels on each side, then smoothed with a normalized 5×5 Gaussian kernel of standard deviation one pixel. The discrete contrast is $\sum I^2/((\sum I)^2 + 10^{-9})$. The Gaussian IWE used in the analysis is a continuous idealization of this discretization. Canvas cropping can alter mass and curvature and is not part of the boundary-free proof.

Translation initialization uses a 13×13 grid over $[-v_{\max}, v_{\max}]^2$, with default $v_{\max} = 0.625P$. It is followed by 40 Adam steps with learning rate 0.5, without projecting velocities back to the grid range. For coherence screening, the warp is

$$\boldsymbol{x}'_k = \boldsymbol{x}_k - \tau_k\{\boldsymbol{v} + \boldsymbol{S}(\omega)(\boldsymbol{x}_k - \boldsymbol{a}_j)\}, \qquad \boldsymbol{S}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}, \tag{16}$$

where $\boldsymbol{a}_j$ is the patch center and $\omega = 0.6 \tanh a$. Screening uses an 11×11 translation grid and two rotational initializations, $a = -0.3$ and $a = 0.3$, refined for 40 steps with learning rates 0.5 and 0.02 for translation and a, respectively. The coherence numerator is the best contrast visited, including the identity candidate. The grouping threshold ρ is loaded from the preprocessing metadata.

Per-scene results Per-scene results are shown in Tab. 5 and 6.

Table 5: Per-scene reconstruction quality (Part I of II). Entries are scene-level means over repeated runs. We highlight the **best** and second-best results. Ours w/ optical flow replaces our event flow with RGB optical flow.

Scene	Method	Full image			Dynamic region		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Bellevue	D-3DGS	21.12	0.8142	0.2203	14.71	0.7834	0.1983
	MotionGS	23.20	0.8438	0.2056	19.63	0.7591	0.2280
	DeGauss	19.20	0.7999	0.1981	15.63	0.6777	0.2737
	E-D3DGS	21.29	0.8097	0.2079	17.72	0.6932	0.3057
	Ours w/ optical flow	23.06	0.8455	0.2041	19.54	0.7626	0.2379
	Ours w/ event flow	**24.21**	**0.8543**	**0.1842**	**20.75**	**0.9221**	**0.1044**
Umbrella	D-3DGS	28.69	0.8254	0.3308	23.56	0.8540	0.2565
	MotionGS	28.28	0.8192	0.3234	23.97	0.8160	0.3076
	DeGauss	28.53	0.8275	0.2924	24.59	0.8358	**0.2415**
	E-D3DGS	29.85	0.8430	0.2896	25.39	0.8666	0.2440
	Ours w/ optical flow	29.73	0.8318	0.3316	25.60	0.8461	0.2857
	Ours w/ event flow	**31.35**	**0.8502**	**0.1918**	**27.81**	**0.8699**	0.2425
Schaff	D-3DGS	25.04	0.6955	0.4640	24.25	0.7373	0.3790
	MotionGS	24.69	0.6896	0.4245	23.64	0.7259	0.3556
	DeGauss	23.39	0.6574	**0.4136**	22.32	0.6961	0.3476
	E-D3DGS	25.56	0.7046	0.4280	24.51	0.7400	0.3590
	Ours w/ optical flow	25.40	0.6973	0.4648	24.39	0.7359	0.3889
	Ours w/ event flow	**25.74**	**0.7069**	0.4414	**30.10**	**0.9299**	**0.1264**
Balloon	D-3DGS	33.46	0.8448	0.3495	24.14	0.8935	0.2417
	MotionGS	30.79	0.8425	0.3339	21.62	0.9095	0.1981
	DeGauss	31.12	0.8375	**0.3197**	22.83	0.9085	0.1947
	E-D3DGS	**33.70**	**0.8462**	0.3312	26.29	0.9219	0.1896
	Ours w/ optical flow	32.72	0.8452	0.3516	24.83	0.9215	0.1870
	Ours w/ event flow	33.58	0.8457	0.3448	**26.84**	**0.9246**	**0.1805**
Water bomb ETH	D-3DGS	31.02	0.9336	0.1186	26.48	0.9161	0.1247
	MotionGS	31.32	**0.9454**	**0.0916**	26.32	0.9180	0.1162
	DeGauss	30.47	0.9336	0.1010	25.77	0.9051	0.1229
	E-D3DGS	31.62	0.9344	0.1168	26.80	0.9150	0.1181
	Ours w/ optical flow	31.47	0.9351	0.1161	26.67	0.9173	0.1230
	Ours w/ event flow	**32.49**	0.9444	0.0936	**27.98**	**0.9195**	**0.1128**

Optical flow ablation. Ours w/ optical flow reads only the training split of the RGB sequence. For each 20 ms window, we select the training images nearest its endpoints and estimate forward and backward flow with the GMFlow Sintel checkpoint used by MotionGS. A pixel with forward flow $\mathbf{f}$ is reliable when $\|\mathbf{f}(\boldsymbol{p}) + \mathbf{b}(\boldsymbol{p} + \mathbf{f}(\boldsymbol{p}))\|_2 \leq 1 + 0.05\|\mathbf{f}(\boldsymbol{p})\|_2$. After scaling the flow to the window duration, a patch is retained when at least 25% of its pixels are reliable, its 90th-percentile RGB change is at least 0.02, and the norm of its reliable-pixel median flow is at least two pixels. This median defines $\boldsymbol{v}_j$, we set $\boldsymbol{H}_j = \boldsymbol{I}$, and R_j is the fraction of reliable flows whose residual from the median is at most $\max(1, 0.25\|\boldsymbol{v}_j\|_2)$ pixels. The same construction is repeated at half-patch resolution for subdivision; all subsequent grouping, assignment, seeding, and training stages are unchanged.

Curvature and grouping. The negative Hessian of the translation contrast is estimated with central differences of step two pixels per window: four axial and four diagonal evaluations, in addition to the center. Negative eigenvalues are clipped to zero. Normalization divides by $\max(\lambda_{\max}, 10^{-12})$ for positive $\lambda_{\max}$ and returns zero otherwise. The quadratic interpretation is exact to second order only at a smooth interior local maximizer of the idealized objective; numerical estimates and local optimization need not satisfy this ideal.

Table 6: Per-scene reconstruction quality (Part II of II). Entries are scene-level means over repeated runs. We highlight the **best** and second-best results. Ours w/ optical flow replaces our event flow with RGB optical flow.

Scene	Method	Full image			Dynamic region		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Candle	D-3DGS	25.76	0.9213	0.1469	22.56	0.9279	0.1069
	MotionGS	29.18	0.9386	0.1394	24.83	0.9546	0.0859
	DeGauss	27.04	0.9351	**0.1209**	22.73	0.9496	**0.0791**
	E-D3DGS	29.64	0.9393	0.1312	25.06	0.9548	0.0832
	Ours w/ optical flow	**30.27**	**0.9436**	0.1388	**26.99**	**0.9598**	0.0808
	Ours w/ event flow	29.31	0.9402	0.1406	25.85	0.9564	0.0845
Confetti	D-3DGS	34.15	0.8329	0.3720	24.30	0.8557	0.2532
	MotionGS	33.97	0.8337	0.3580	26.11	0.8844	0.2051
	DeGauss	34.06	0.8335	0.3566	26.26	0.8792	0.2023
	E-D3DGS	34.24	0.8333	0.3698	26.74	0.8870	0.2039
	Ours w/ optical flow	34.23	0.8329	0.3730	26.75	0.8863	0.2067
	Ours w/ event flow	**34.75**	**0.8353**	**0.3527**	**27.99**	**0.8928**	**0.1875**
Water bomb floor	D-3DGS	25.14	0.8225	0.2240	20.41	0.7901	0.2192
	MotionGS	25.29	0.8587	0.1557	19.21	0.7882	0.2023
	DeGauss	23.14	0.8314	0.1699	17.68	0.7623	0.2003
	E-D3DGS	**26.90**	**0.8756**	**0.1461**	20.98	0.8123	**0.1924**
	Ours w/ optical flow	25.56	0.8160	0.2398	20.14	0.7856	0.2291
	Ours w/ event flow	25.60	0.8242	0.2152	**23.66**	**0.8273**	0.2003
Plate	D-3DGS	35.56	0.8606	0.3164	27.00	0.8222	0.2927
	MotionGS	35.98	0.8805	0.2943	25.22	0.7982	0.3224
	DeGauss	36.69	0.8814	0.2948	**28.89**	**0.8413**	**0.2501**
	E-D3DGS	36.59	0.8810	0.3055	27.87	0.8255	0.2929
	Ours w/ optical flow	36.54	0.8809	0.3062	27.41	0.8229	0.2991
	Ours w/ event flow	**36.92**	**0.8818**	**0.2926**	28.80	0.8354	0.2905
Jogging	D-3DGS	34.34	0.9480	0.0639	26.69	0.9142	0.0802
	MotionGS	34.13	0.9586	0.0495	25.16	0.8985	0.1069
	DeGauss	34.67	0.9491	0.0630	26.77	0.9150	0.0788
	E-D3DGS	34.40	0.9587	0.0455	26.27	0.9040	0.1031
	Ours w/ optical flow	34.54	0.9479	0.0652	26.84	0.9140	0.0866
	Ours w/ event flow	**35.55**	**0.9656**	**0.0333**	**26.88**	**0.9177**	**0.0785**

Affine fitting uses patch coordinates and subtracts the current region's mean position. The regularizers in Eq. 8 are $\gamma = 0.1$ and $\eta = 10^{-4}$. The global intercept is restored after solving. The membership tolerances are $\epsilon_{\text{abs}} = 2$ pixels per window and $\epsilon_{\text{rel}} = 0.25$. Seeds are ordered by the smallest eigenvalue of normalized $\hat{\boldsymbol{H}}$, with $10^{-3}R$ added as a secondary preference. Velocities with $\|\boldsymbol{v}\|_2 \geq 0.9v_{\max}$ are excluded from initial growth and attached to neighboring fields when their cosine similarity exceeds 0.9; this is a heuristic large-velocity check, not detection of a hard optimization bound. Optional subdivision re-estimates mixed-motion candidates on a half-patch grid. The temporal matching factor is $\kappa = 1.5$.

Track selection and spatial completion. Tracks observed in at least $K_{\min} = 3$ windows are candidates. Each original patch is counted once even when represented by multiple half-patch cells. For the six-parameter test, coordinates are expressed relative to the mean integer cell index of the region; the half-cell center offset is retained. No ridge term is included in either information matrix. The median per-region minimum eigenvalue must be at least $\lambda_0 = 0.1$, first for $\boldsymbol{M}^{(6)}$ and, if that test fails, for $\boldsymbol{M}^{(2)}$. The threshold is tied to this coordinate and normalization convention.

Spatial completion is applied to long-lived candidate tracks; it does not change the measurements used for model assignment, and short tracks retain their original support. Closing uses a 3×3 structuring element for two iterations on the half-patch label grid. Event-free cells are added only within enclosed holes. Cells assigned to other long-lived tracks are protected; cells from other groups must pass the flow-consistency test, with tolerance multiplied by three inside holes and by one in closing gaps. Completed labels are used for training assignment; the original labels are retained separately.

Static masks and voting. The static mask excludes any patch reaching the event-count threshold in any window, or whose 90th percentile of pixelwise temporal grayscale standard deviation exceeds 0.02. RGB statistics use subsampled training frames only, with index stride $\max(1, \lfloor N_{\text{train}}/40 \rfloor)$. Thus the static mask is fixed across time. Pixels without sufficient events in a window can still be assigned to free deformation if RGB variation rules out the static label.

Votes are accumulated separately at window centers containing a motion group. The implementation differentiates the masked red-channel rendering with respect to each Gaussian's degree-zero red SH coefficient and divides by $Y_{00} = 0.28209479177387814$, with appearance correction disabled. For the standard lower-clamped SH color conversion, the effective contribution is w_i times the derivative of the color clamp. This is why the gradient score equals the ideal blending-weight vote only when that channel is unclamped.

For each track k, the denominator D_{ik} in Eq. 11 sums tracked and unreliable dynamic contributions only over that track's active windows. The track with the highest V_{ik}/D_{ik} is assigned when this ratio is at least 0.5. Static and dynamic-but-event-insufficient labels are first determined from their sequence-level votes; a qualifying temporally local track assignment then takes precedence. Total support below 0.5 produces an unseen label. Unseen, unreliable, and dynamic no-event labels remain in the free branch. Track ties follow the implementation's first-maximum rule.

Track-guided seeding. For each selected track, the reference window is the one with the largest label support, and the nearest RGB training camera supplies the reference image. The method seeds only when the mean absolute RGB residual inside the track mask is at least twice the full-image mean. It selects the top 25% residual pixels within the mask, provided this yields at least 16 samples. Their depth is set to 0.9 times the median projected depth of Gaussians falling inside the mask two windows before the track first appears; at least ten such Gaussians are required. Each selected pixel is back-projected at this shared depth, initialized from its reference RGB value, assigned an isotropic scale matching the sampling spacing, and given opacity 0.1. The event loss is then prefitted and the learned translation at the reference time is subtracted from the seeded canonical positions; this exactly fixes the reference location for translation-only tracks.

Motion branches and optimization. Track embeddings have 16 dimensions and are zero-initialized; unassigned Gaussians share an additional embedding. Static assignment zeros time-dependent deformation, while canonical scene parameters remain trainable by RGB reconstruction. The rigid network uses six-frequency time encoding, two width-64 ReLU hidden layers, and a $6K$-dimensional output for K tracks. Its final layer is zero-initialized. Reference centroids are computed once from the assigned canonical means and stored as fixed buffers. Translation-only tracks set the rotation output to zero. The piecewise gate is one on the interval between the first and last active window centers and follows a cubic smoothstep to zero over a normalized-time margin of 0.05. Within the transition margin, position, rotation offset, and scale offset are blended between the shared and free branches.

The default switch is at iteration 15,000. Rigid transforms are first optimized for 2,000 steps using only the event-derived displacement term, with Adam learning rate 5×10^{-3} and at most 2,048 sampled Gaussians per track-window.

The joint stage uses $\lambda_{\text{flow}} = 10^{-3}$ and an initial rigid-network learning rate of 10^{-3}. Canonical positions are detached for this loss; RGB gradients still update canonical parameters and rigid transforms. The displacement residual uses the unsquared Euclidean norm.

Table 7: Sequences and evaluated frame intervals. Endpoints are inclusive.

Dataset	Sequence	Frame interval
HS-ERGB	fountain_bellevue2	000420–000570
HS-ERGB	spinning_umbrella	000300–000399
HS-ERGB	fountain_schaffhauserplatz_02	000000–000100
HS-ERGB	baloon_popping	000085–000125
HS-ERGB	water_bomb_eth_01	002950–003040
HS-ERGB	candle	000515–000565
HS-ERGB	confetti	000374–000550
HS-ERGB	water_bomb_floor_01	000350–000399
HS-ERGB	spinning_plate	000220–000420
CED	jogging	000000–000099

Table 8: Rendered-flow EPE in dynamic regions. OF means optical flow and EF means event flow.

Scene	D-3DGS	MotionGS	DeGauss	E-D3DGS	Ours w/ OF	Ours w/ EF
Bellevue	11.647	11.580	11.502	12.038	11.940	**10.858**
Umbrella	8.177	9.746	6.083	**2.001**	3.418	3.388
Schaff	5.850	6.052	6.312	**5.145**	5.917	5.785
Balloon	30.783	40.031	27.935	37.153	31.520	**23.520**
Water bomb ETH	2.833	2.793	2.870	2.179	**1.854**	2.015
Candle	1.149	0.891	1.623	1.492	1.059	**0.803**
Confetti	7.533	7.525	7.513	**7.345**	7.479	7.372
Water bomb floor	8.850	7.994	8.379	**6.872**	8.116	7.493
Plate	0.441	0.444	**0.419**	0.432	0.439	0.448
Jogging	0.709	0.858	**0.605**	0.795	0.692	0.753
Average	7.797	8.791	7.324	7.545	7.243	**6.244**

B Dataset Intervals

Tab. 7 lists the contiguous frame intervals used in all comparisons. The intervals retain the event-dense dynamic portion of each original sequence.

C Rendered-Flow EPE Protocol

Rendered-flow EPE evaluates the image-plane motion reproduced by each reconstruction. For every held-out frame pair separated by four native frames, let I_t, I_{t+4} be the ground-truth images and $\hat{I}_t, \hat{I}_{t+4}$ the corresponding renders. We apply the same fixed GMFlow model with Sintel weights to both pairs:

$$\mathbf{F}_t^{\mathrm{gt}} = \mathcal{G}(I_t, I_{t+4}), \qquad \mathbf{F}_t^{m} = \mathcal{G}(\hat{I}_t^{m}, \hat{I}_{t+4}^{m}). \tag{17}$$

Images are resized only as required by the estimator, with the longer side capped at 896 pixels and both dimensions made divisible by 16; the resulting flows are rescaled to the original pixel coordinates. For the dynamic mask Ω_t from the nearest event window, the per-pair error is

$$\mathrm{EPE}_t^{m} = \frac{1}{|\Omega_t|} \sum_{\boldsymbol{p} \in \Omega_t} \left\| \mathbf{F}_t^{m}(\boldsymbol{p}) - \mathbf{F}_t^{\mathrm{gt}}(\boldsymbol{p}) \right\|_2 . \tag{18}$$

We average the per-pair values so that frames with larger masks do not receive more weight. The same estimator, temporal offset, masks, and resizing procedure are used for every method.

Table 9: Per-scene efficiency (Part 1 of 2). Time is training time, Gaussian count is the final representation size, and FPS is rendering speed.

Scene	Method	Time	Gaussians	FPS
Bellevue	D-3DGS	21:03	154380	56
	MotionGS	3:24:18	512622	24
	DeGauss	1:27:46	230448	47
	E-D3DGS	2:31:07	809610	20
	Ours w/ RGB optical flow	43:15	244891	38
	Ours w/ Event optical flow	28:59	176360	43
Umbrella	D-3DGS	19:40	61347	109
	MotionGS	1:45:09	122480	63
	DeGauss	1:54:11	164619	50
	E-D3DGS	1:00:06	213025	69
	Ours w/ RGB optical flow	25:33	57772	88
	Ours w/ Event optical flow	22:51	58763	90
Schaff	D-3DGS	14:27	52521	137
	MotionGS	1:40:41	190650	44
	DeGauss	1:53:57	189741	41
	E-D3DGS	1:27:46	382886	27
	Ours w/ RGB optical flow	23:34	55574	99
	Ours w/ Event optical flow	20:14	57035	98
Balloon popping	D-3DGS	22:25	57055	109
	MotionGS	1:35:31	87831	71
	DeGauss	2:06:42	154118	44
	E-D3DGS	1:08:14	144313	109
	Ours w/ RGB optical flow	27:24	55080	89
	Ours w/ Event optical flow	22:55	56048	112
Water bomb ETH	D-3DGS	19:45	128910	73
	MotionGS	1:12:32	445690	21
	DeGauss	1:20:29	224132	61
	E-D3DGS	51:31	459577	48
	Ours w/ RGB optical flow	24:07	126847	60
	Ours w/ Event optical flow	20:19	124470	67

D Per-Scene Efficiency

E Aperture Ambiguity in an Ideal Edge Model

An infinite uniform edge has infinite total event mass, so its unnormalized two-dimensional L^1 and L^2 norms cannot be used directly in Eq. 4. We instead use a finite periodic tangential coordinate $\eta \in [0, L)$ and an unbounded normal coordinate $\xi \in \mathbb{R}$. This preserves tangential translation invariance while keeping the contrast finite. Let $\boldsymbol{x} = \xi \boldsymbol{n} + \eta \boldsymbol{e}$, with unit vectors $\boldsymbol{e} \perp \boldsymbol{n}$. Take a normalized temporal density ψ on $[-1/2, 1/2]$ and event density

$$\varrho(\xi, \eta, \tau) = \nu_0 \psi(\tau) \delta\big(\xi - c_0 - \tau \boldsymbol{n}^\top \boldsymbol{u}\big), \qquad \nu_0 > 0. \tag{19}$$

A candidate warp introduces normal mismatch $d = \boldsymbol{n}^\top(\boldsymbol{u} - \boldsymbol{v})$. Translation of the uniform periodic tangential coordinate has no effect. After Gaussian smoothing in the normal direction and normalized periodic smoothing in the tangential direction, the IWE is

$$I_{\boldsymbol{v}}(\xi, \eta) = \nu_0 \phi_d(\xi - c_0), \qquad \phi_d(s) = \int \psi(\tau) G_\sigma(s - \tau d)\, d\tau. \tag{20}$$

Table 10: Per-scene efficiency (Part 2 of 2). Time is training time, Gaussian count is the final representation size, and FPS is rendering speed.

Scene	Method	Time	Gaussians	FPS
Candle	D-3DGS	7:37	36335	209
	MotionGS	1:28:36	78793	25
	DeGauss	1:24:44	174038	53
	E-D3DGS	17:41	96288	217
	Ours w/ optical flow	8:25	38044	201
	Ours w/ event flow	8:09	39348	199
Confetti	D-3DGS	24:44	58503	106
	MotionGS	1:39:43	80126	26
	DeGauss	1:28:30	112886	88
	E-D3DGS	51:04	124420	130
	Ours w/ optical flow	29:56	61981	87
	Ours w/ event flow	28:23	63161	96
Water bomb floor	D-3DGS	19:07	67562	91
	MotionGS	1:35:23	285252	30
	DeGauss	1:50:19	200386	49
	E-D3DGS	1:13:06	588740	41
	Ours w/ optical flow	30:57	104629	54
	Ours w/ event flow	22:44	69563	87
Plate	D-3DGS	18:11	48236	147
	MotionGS	1:24:52	71926	97
	DeGauss	2:09:19	129070	44
	E-D3DGS	54:49	136188	118
	Ours w/ optical flow	18:24	43738	156
	Ours w/ event flow	20:38	44651	143
Jogging	D-3DGS	24:30	67059	125
	MotionGS	1:23:36	185004	54
	DeGauss	55:57	146492	59
	E-D3DGS	24:49	178993	324
	Ours w/ optical flow	10:24	63454	130
	Ours w/ event flow	14:15	63324	145

Since $\int \phi_d = 1$, its contrast is

$$f(\boldsymbol{v}) = \frac{\nu_0^2 L \|\phi_d\|_2^2}{(\nu_0 L)^2} = \frac{1}{L}\|\phi_d\|_2^2 = \frac{1}{L}\mathbb{E}_{\tau,\tau'\sim\psi}\left[G_{\sqrt{2}\sigma}\big((\tau-\tau')d\big)\right], \tag{21}$$

where τ, τ' are independent. Every summand is maximized at $d = 0$. If $\mathrm{Var}_\psi(\tau) > 0$, this is the unique maximizing normal mismatch, while every tangential velocity remains equivalent.

Writing $f(\boldsymbol{v}) = g(\boldsymbol{n}^\top \boldsymbol{v})$ and differentiating at any maximizer gives

$$-\nabla_{\boldsymbol{v}}^2 f(\boldsymbol{v}^\star) = \kappa_\psi \boldsymbol{n}\boldsymbol{n}^\top, \qquad \kappa_\psi = \frac{G_{\sqrt{2}\sigma}(0)}{L\sigma^2}\,\mathrm{Var}_\psi(\tau). \tag{22}$$

For positive temporal variance this matrix has rank one, with null space exactly the edge direction. If all event times coincide, $\kappa_\psi = 0$ and no velocity component is constrained. This proves the ideal aperture claim without treating a singular precision matrix as an invertible Gaussian covariance.

Real patches have boundaries, nonuniform event density, noise, and discrete sampling. These effects can introduce tangential curvature, and the ideal rank-one result is not an exact statement about every finite patch. The argument does not imply an additive Hessian decomposition for arbitrary intersecting edges, since the squared IWE objective contains cross terms.

F Measurement Rank and Affine Fitting

For any parameter perturbation $\boldsymbol{d} \in \mathbb{R}^6$,

$$\boldsymbol{d}^\top \boldsymbol{M}_{\mathcal{S}} \boldsymbol{d} = \sum_{j \in \mathcal{S}} \|\hat{\boldsymbol{H}}_j^{1/2} \boldsymbol{J}_j \boldsymbol{d}\|_2^2. \tag{23}$$

Thus $\boldsymbol{M}_{\mathcal{S}}$ is positive definite exactly when the stacked matrices $\hat{\boldsymbol{H}}_j^{1/2} \boldsymbol{J}_j$ have column rank six. Each ideal edge patch supplies at most one scalar constraint. Diverse normals alone are insufficient: their spatial arrangement must also constrain the affine coefficients. The positive diagonal regularizer in the implemented centered fit guarantees a unique numerical minimizer even when measurement rank is deficient; it supplies no additional measured motion information.