

One World, Dual Timeline: Decoupled Spatio-Temporal Gaussian Scene Graph for 4D Cooperative Driving Reconstruction

Yulong Chen[▲♠*] , **Xiaoyun Dong**[▲*], **Haoyu Zhang**[▲♣*], **Zongxian Yang**[▲♠], **Lewei Xie**[▲♠], **Jianping Wang**[▲], **Xinke Li**[▲▼†], **Yifan Zhang**[▲♠†], **Kai Wang**[▲♠†]

[▲] City University of Hong Kong, Hong Kong, China
[♠] City University of Hong Kong (Dongguan), Guangdong, China
[♣] SLAI, Shenzhen, China
[▼] Chengdu Research Institute, City University of Hong Kong, Sichuan, China

yulong.chen@cityu-dg.edu.cn xiaodong8@cityu.edu.hk hzhang2838-c@my.cityu.edu.hk
zongxian.yang@cityu-dg.edu.cn lewei.xie@cityu-dg.edu.cn jianwang@cityu.edu.hk
xinkeli@cityu.edu.hk yifan.zhang@cityu-dg.edu.cn kai.wang@cityu-dg.edu.cn

Abstract

Reconstructing dynamic scenes from Vehicle-to-Infrastructure Cooperative Autonomous Driving (VICAD) data is fundamentally complicated by temporal asynchrony: vehicle and infrastructure cameras operate on independent clocks, capturing the same dynamic agent such as cars and pedestrians at different physical times. Existing Gaussian Scene Graph methods implicitly assume synchronized observations and assign a single pose per agent per frame, which is an assumption that breaks in cooperative settings, where the resulting gradient conflicts cause severe ghosting on dynamic agents. Under the stated assumptions, we show that assigning asynchronous observations to a shared dynamic pose incurs an irreducible photometric loss scaling quadratically with agent velocity and cross-source time offset. To resolve this, we propose **DUST** (**D**eco**U**pled **S**patio-**T**emporal) Gaussian Scene Graph for 4D Cooperative Driving Reconstruction. DUST Gaussian scene graph shares a canonical Gaussian set per agent for appearance consistency, while maintaining decoupled pose trajectories aligned to each source's true captu re timestamps. We prove that this decoupling makes the pose-gradient kernel block-diagonal with respect to the source-specific pose parameters, while the shared Gaussian parameters remain coupled. To make DUST practical, we further introduce a static anchor-based pose correction pipeline that corrects spatio misalignment between vehicle and infrastructure annotations, and a pose-regularized joint optimization scheme that prevents trajectory jitter and drift during early training. On 26 sequences from V2X-Seq, DUST achieves state-of-the-art performance, improving dynamic-area PSNR by 3.2 dB over the strongest baseline and reducing Fréchet Video Distance by 37.7%, with keeping robustness under larger temporal asynchrony. Code is available at `https://github.com/mura1n/DUST`.

1 Introduction

High-fidelity 4D scene reconstruction [3, 7, 29, 37] has become a practical alternative to real-world data collection for autonomous driving: it converts recorded driving logs into photorealistic digital

*Indicates equal contribution.
†Corresponding authors.

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

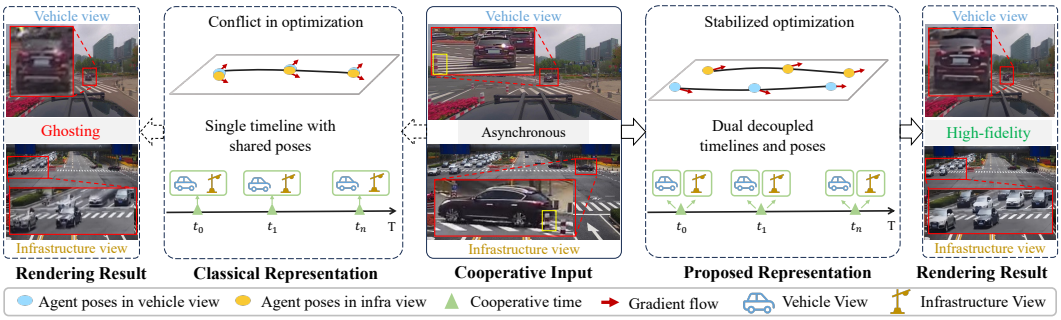


Figure 1: **Comparison of scene representations. Cooperative Input:** Vehicle and infrastructure sensors capture dynamic scenes asynchronously. **Top:** the black car has passed the pole; bottom: it has not. **Classical Representation:** Assigning these asynchronous observations to one shared pose creates conflicting gradients, leading to severe ghosting. **Proposed Representation:** Our DUST-GSG assigns separate pose timelines to each source while sharing canonical Gaussians, which separates source-specific pose updates and enables stable, high-fidelity reconstruction.

environments that support novel-view synthesis (NVS), scene editing, and scalable simulation. A particularly compelling use case arises in Vehicle-to-Infrastructure Cooperative Autonomous Driving (VICAD) [11, 16, 28, 31, 35], where both a moving vehicle and fixed roadside cameras (i.e., infrastructure) observe the same scene. Reconstructing from both sources jointly enables observations that neither platform can produce alone, while also alleviating the shortage of real paired vehicle-infrastructure data, which is costly to collect at scale.

Recent driving reconstruction methods [3, 29, 37] commonly adopt the Gaussian Scene Graph (GSG) to decompose a scene into a static background and dynamic agents, achieving strong results on single-platform data. These formulations all rest on an implicit assumption: at each frame, all observations share one timeline, and each dynamic agent is described by a single pose. This holds when sensors are driven by a common clock, but breaks down in cooperative settings, where vehicle and infrastructure cameras are triggered independently and their observations are inherently asynchronous. Forcing such asynchronous observations to share one pose induces conflicting gradients during optimization and produces ghosting on dynamic agents (Fig. 1). A concurrent effort on cooperative reconstruction [25] does not address this temporal asynchrony. The core challenge is therefore ***how to faithfully represent asynchronous multi-source observations in a unified 4D scene***.

We propose **DUST**, a 4D cooperative reconstruction framework based on a Decoupled Spatio-Temporal Gaussian Scene Graph (DUST-GSG). DUST-GSG models each dynamic agent with a shared set of canonical Gaussians to ensure appearance consistency, while assigning separate pose timelines to the vehicle and infrastructure sources (Fig. 2). Each source queries its own pose trajectory using its recorded timestamp. Our *theoretical analysis* characterizes the local incompatibility of assigning different observed states to one pose and shows that source-specific pose parameters remove the corresponding direct pose-gradient coupling. A single continuous trajectory queried at distinct capture times does not fall under the shared-pose assumption analyzed here. Beyond this theoretical design, applying our DUST-GSG to real cooperative data requires overcoming two problems. First, raw cooperative labels often contain spatio misalignment and provide poor pose initialization for DUST-GSG. We resolve this through a pose correction method. We use static vehicles as anchors to align labels from two sources, then regenerate cooperative labels for accurate initialization. Second, during reconstruction training, optimizing poses from pure image supervision easily leads to temporal drift and jitter. We address this by introducing a joint optimization scheme with pose regularization. This scheme enforces smooth motion across frames and prevents the poses from drifting away from their initial states. Combined with these two designs, DUST ensures robust and high-fidelity cooperative reconstruction under asynchronous conditions.

Our main contributions are as follows:

- We propose DUST, a cooperative reconstruction paradigm using a Decoupled Spatio-Temporal Gaussian Scene Graph. We also provide a theoretical analysis of this paradigm.
- We introduce a static anchor-based pose correction pipeline to refine vehicle-infrastructure alignment, subsequently regenerating cooperative labels offline to initialize agent poses.

- We propose a pose-regularized joint optimization scheme for Gaussian parameters and agent poses, which enforces smooth 3D motion and effectively mitigates early-stage drift.
- Evaluated on the V2X-Seq, DUST achieves state-of-the-art performance in scene reconstruction and NVS, significantly outperforming previous methods in terms of dynamic area metrics (**+3.2 dB** PSNR and **-37.7%** FVD) for reconstruction.

2 Related Work

4D Driving Scene Reconstruction. Recent driving reconstruction methods [1, 4, 5, 7, 10, 13, 22] build upon NeRF and 3D Gaussian Splatting to model dynamic urban environments. A dominant paradigm decomposes the scene into static backgrounds and dynamic agents. OMNIRE[3], StreetGS[29], and DrivingGaussian[37] track vehicles using explicit 3D tracking boxes and render them from canonical spaces. Other approaches like PVG[2] avoid tracking boxes to model general scene vibrations. Extending these representations to cooperative reconstruction is an emerging frontier. For example, CRUISE [25] explores generative scene editing for cooperative scenarios. However, all these existing frameworks rely on a single timeline constraint. They assume perfectly synchronized cameras across all observations. In real cooperative driving systems, vehicle and infrastructure sensors operate on independent hardware clocks. This hardware independence creates natural time gaps between cross observations. When existing methods force these asynchronous inputs into a single shared timeline, they inevitably produce severe artifacts. Our work directly targets this fundamental system limitation. SimULi [19] uses separate camera and LiDAR Gaussian sets linked by an anchoring constraint and accounts for measurement times in sensor rendering. DUST shares canonical Gaussians across cooperative cameras and separates their object pose trajectories.

Vehicle-Infrastructe Cooperative Autonomous Driving. Cooperative systems share sensor data across vehicles and infrastructure to overcome physical line of sight limitations. Methods like DiscoNet [9] and V2X-ViT [26] exchange deep features to improve object detection. Other works optimize communication bandwidth [6] or address spatial misalignment [12]. Spatial calibration is complementary to our treatment of asynchronous observations: pose correction improves initialization but does not establish a common sensor clock. However, developing and evaluating these algorithms requires massive amounts of driving data. Most existing models are trained on purely synthetic datasets [24, 26]. These virtual environments fail to capture natural sensor data and complex physical constraints. While datasets like DAIR-V2X [33] and V2X-Seq [34] solve this by providing real world observations, datasets cannot simulate new behaviors. Therefore, the cooperative autonomous driving industry urgently needs highly realistic simulation environments for algorithm development and testing. Our framework transforms real sequential data into fully controllable digital twins to directly serve this exact purpose.

3 Method

Our proposed DUST framework uses inputs from both the vehicle and the infrastructure, including image sequences, camera poses, and cooperative tracking bounding boxes. Our goal is to learn a shared 4D Gaussian scene representation supervised by data from both sources. This directly supports novel view synthesis, scene editing, and driving simulation. First, Sec.3.1 introduces the Decoupled Spatio-Temporal Gaussian Scene Graph (DUST-GSG). Next, Sec.3.2 analyzes the photometric residual of the shared-pose formulation and the decoupling of source-specific pose parameters under the stated assumptions. Finally, to make DUST-GSG practical, Sec.3.3 further introduces a pose correction step for accurate DUST-GSG initialization, and a joint optimization scheme that stabilizes the overall training process.

3.1 Decoupled spatio-Temporal Gaussian Scene Graph

Gaussian Scene Graph. The Gaussian Scene Graph (GSG) [3, 29, 37] organizes a driving scene into a structured graph of 3D Gaussian nodes, enabling joint reconstruction of static environments and dynamic agents. It decomposes the scene into a time-invariant background node $\mathcal{V}_{\mathrm{bg}}$, a sky node $\mathcal{V}_{\mathrm{sky}}$, and a set of dynamic agent nodes $\{\mathcal{V}_{\mathrm{ag}}\}$ for moving objects such as vehicles and pedestrians, where each node is represented as a set of 3D Gaussian primitives $\mathcal{G}$. The background and sky nodes are static and time-invariant. The dynamic agents are further classified into rigid agents, such as cars

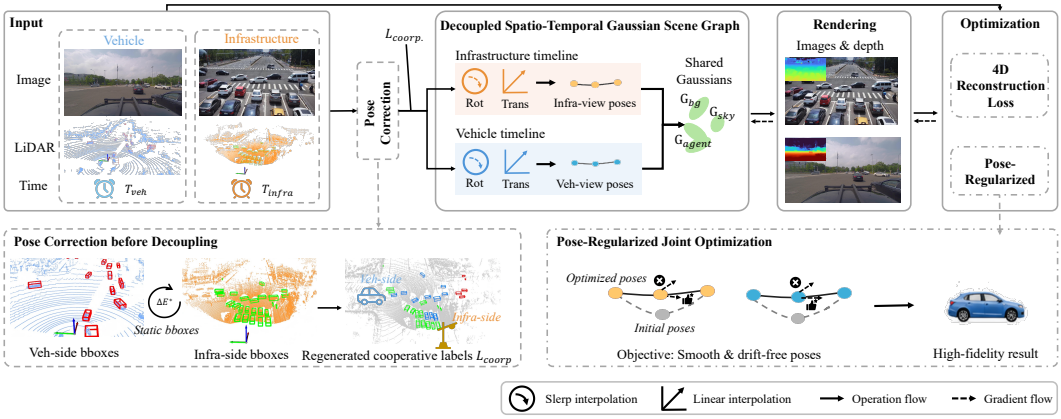


Figure 2: **Overview of DUST.** DUST first regenerates cooperative labels offline using co-visible static vehicles as anchors. The pose correction initialize a Decoupled Spatio-Temporal Gaussian Scene Graph, where each dynamic agent shares canonical Gaussians but maintains separate vehicle-side and infrastructure-side pose timelines. Joint optimization then refines Gaussian parameters and dual-timeline poses with pose regularization for stable cooperative reconstruction.

and trucks, and non-rigid agents, like pedestrians and cyclists. For these agents,the key insight is to represent each agent by a *canonical* Gaussian set $G_{ag} = \{\mathcal{G}_n\}_{n=1}^{N}$, a fixed appearance defined in the agent's local frame, decoupled from a pose trajectory $T(t) \in \mathrm{SE}(3)$ that places the agent in the world at time t:

$$\mathcal{V}_{\mathrm{ag}}(t) = T(t) \otimes f_t(G_{ag}). \tag{1}$$

Here, $f_t : \mathbb{G} \to \mathbb{G}$ models the internal shape deformation of the agent, and $\mathbb{G} = \bigcup_{N\in\mathbb{N}}\{\{\mathcal{G}_n\}_{n=1}^{N}\}$ denotes the space of 3D Gaussian sets. For rigid agents, f_t is simply the identity mapping. For non-rigid agents, f_t serves as a time-dependent deformation field [32] that captures local movements like limb motion. This factorization of a shared canonical shape optimized across all frames and a per-frame pose learned from image supervision makes GSG efficient and expressive for single-vehicle driving reconstruction.

Limitation in VICAD. In VICAD, however, vehicle and infrastructure cameras are triggered asynchronously: each source $c \in \{\mathrm{veh}, \mathrm{infra}\}$ captures the same agent at a different physical time $t_i^c \neq t_i$. For static elements this discrepancy is negligible, but dynamic agents physically occupy different world positions at t_i^{veh} and t_i^{infra}. Forcing both sources to share a single pose $T(t_i)$ thus attributes contradictory image gradients to the same Gaussian primitives, causing the canonical shape to receive conflicting supervision and the reconstructed geometry to be systematically misaligned.

Decoupled spatio-Temporal Gaussian Scene Graph (DUST-GSG). To address this, DUST-GSG keeps the canonical Gaussians G_{ag} and deformation field f_t *shared* across both sources for spatio consistency, but *splits* the pose trajectory into two source-specific sequences $\{T^{\mathrm{veh}}(t_i^{\mathrm{veh}})\}$ and $\{T^{\mathrm{infra}}(t_i^{\mathrm{infra}})\}$ aligned to each source's true capture times. The pose trajectory is initialized from cooperative labels. These labels include all visible agents from both observations. For each agent, the labels provide a sequence of 3D bounding boxes that track its position and orientation over time. To obtain the initial pose $T^{\mathrm{c}}(t_i^c)$ for a specific source c at query time t_i^c, we interpolate between these bounding boxes. We compute the position using linear interpolation and the rotation using spherical linear interpolation (Slerp [17]). Each dynamic agent node is defined as a triplet:

$$\mathcal{V}_{\mathrm{ag}}^{\mathrm{DST}} := \Big\{ G_{ag},\ \ \{T^{\mathrm{veh}}(t_i^{\mathrm{veh}})\}_{i=0}^{K-1},\ \ \{T^{\mathrm{infra}}(t_i^{\mathrm{infra}})\}_{i=0}^{K-1} \Big\}, \tag{2}$$

At render time, source c queries its own trajectory at its own capture time t_i^c:

$$\mathcal{V}_{\mathrm{ag}}^{c}(t_i^c) = T^{c}(t_i^c) \otimes f_{t_i^c}(G_{ag}), \qquad c \in \{\mathrm{veh}, \mathrm{infra}\}. \tag{3}$$

Canonical geometry and base appearance are shared across sources; only dynamic area is source-conditioned. A rendering query specifies the camera and a source-time pair (c, t), with c selecting the source pose branch. Linear translation interpolation and quaternion slerp allow queries between knots within that branch's observed time range. DUST does not enforce a unique source-independent dynamic state at a common physical time. Following OMNIRE [3], we optimize all Gaussian

parameters via differentiable rendering, but extend the optimization to jointly supervise the shared canonical Gaussians G_{ag} using image observations from both the vehicle and infrastructure sources.

3.2 Theoretical Justification of DUST-GSG

We analyze the shared-pose formulation in which asynchronous observations of a moving agent are assigned to the same dynamic pose. Let $\Delta\tau = t_i^{\text{veh}} - t_i^{\text{infra}}$ be the capture-time offset at cooperative timestamp t_i. We characterize the resulting photometric residual and the effect of source-specific pose parameters below; full proofs are in Appendix B.

Theorem 1. *Consider a dynamic agent with canonical Gaussians $G_{ag} = \{\mathcal{G}_n\}_{n=1}^{N}$ (Eq. (1)) moving at velocity $\boldsymbol{v} \in \mathbb{R}^3$ during offset $\Delta\tau$. Let $\mathcal{L}$ be the total photometric reconstruction loss over both sources. The following properties hold under the local linearized rendering, locally rigid motion, and non-degenerate photometric sensitivity assumptions stated in Appendix B, with nonzero agent velocity and time offset.*

1. Irreducible Error under a Shared Pose. *If both sources share a single pose $T(t_i) \in SE(3)$, the optimal loss satisfies*

$$\mathcal{L}^*_{\text{single}} \;\geq\; \frac{|\Delta\tau|^2\|\boldsymbol{v}\|^2}{4}\sum_{n=1}^{N}\lambda_n \;>\; 0, \tag{4}$$

where $\lambda_n > 0$ is the minimum Fisher-information eigenvalue of $\mathcal{G}_n$, quantifying its photometric sensitivity to displacement.

2. Decoupling Property of DUST-GSG. *Under $\mathcal{V}_{\text{ag}}^{\text{DST}}$ with source-specific trajectories $\{T^{\text{veh}}(t_i^{\text{veh}})\}$ and $\{T^{\text{infra}}(t_i^{\text{infra}})\}$, a zero-loss solution exists within this local model ($\mathcal{L}^*_{\text{DST}} = 0$), and the two pose sequences are optimized separately using the photometric losses of their respective sources:: a gradient update to $\{T^{\text{veh}}\}$ does not directly alter the infrastructure photometric loss, and vice versa. This separation concerns pose updates from photometric supervision; the shared canonical Gaussians remain jointly optimized using observations from both sources.*

Discussion. ❶ Property 1 establishes a positive lower bound for the shared-pose formulation under the stated assumptions, with quadratic dependence on both $\|v\|$ and $|\Delta\tau|$. For a vehicle at 10 m/s with $|\Delta\tau| = 70$ ms, the implied misalignment reaches 0.7 m, well beyond the support radius of a single Gaussian, which directly causes the ghosting in Fig. 1. ❷ Property 2 shows that DUST-GSG decouples the source-specific pose parameter blocks. This block-diagonal structure does not apply to the complete parameter space, since both sources still update the shared canonical Gaussian parameters. The analysis does not rule out a single continuous trajectory queried at each sensor's capture timestamp. This motivates the pose correction and regularization strategies described in the following.

3.3 More Techniques for DUST-GSG

Pose Correction before Decoupling. Cooperative labels, i.e., bounding boxes, in dataset [33] merge vehicle and infrastructure annotations based on provided infrastructure-to-vehicle poses. However, small pose errors often cause noticeable misalignment between the two sources. Because DUST-GSG uses these labels to initialize the pose trajectories $T^c(t_i^c)$, inaccurate labels will degrade reconstruction quality. To address this, we introduce an offline pose correction step before decoupling.

At each cooperative timestamp t_i, let $E_i \in \text{SE}(3)$ be the infrastructure-to-vehicle pose. We use co-visible static vehicles as geometric anchors to avoid dynamic interference. We match these static vehicles across both views using the Hungarian method [8], and let $\mathcal{M}_i$ denote the set of matched vehicles at timestamp t_i. For a matched vehicle $m \in \mathcal{M}_i$, let $\mathbf{X}_m^{\text{veh}}(t_i), \mathbf{X}_m^{\text{infra}}(t_i) \in \mathbb{R}^{8\times 3}$ be the 3D bounding box corner matrices in the vehicle and infrastructure coordinates, respectively. We estimate a 6-DoF [38] pose correction $\Delta E_i \in \text{SE}(3)$ by minimizing the corner alignment error:

$$\Delta E_i^* = \arg\min_{\Delta E_i \in \text{SE}(3)} \frac{1}{|\mathcal{M}_i|}\sum_{m\in\mathcal{M}_i}\left\|\mathbf{X}_m^{\text{veh}}(t_i) - \Delta E_i E_i \mathbf{X}_m^{\text{infra}}(t_i)\right\|_F^2. \tag{5}$$

When reliable static correspondences are scarce, the motion threshold can be relaxed to include slowly moving candidate anchors. If no reliable co-visible correspondence remains, we skip anchor-based

correction and retain the initial transformation. This fallback does not guarantee accurate alignment when the initial calibration is poor; the zero-anchor case has not been empirically evaluated. We optimize this using L-BFGS [15]. Multiplying the optimal correction ΔE_i^* by the original pose E_i gives the refined the pose. When reliable static anchors are scarce, the motion threshold can be relaxed to include slowly moving candidates. If no reliable correspondence remains, pose correction is skipped and the initial calibration is retained. For co-visible objects, we use this refined pose to project infrastructure annotations into the vehicle coordinate system to form cooperative annotation. Objects observed by only one source are kept after being transformed into the unified coordinate system.

Based on these regenerated labels, we interpolate poses to fill short missing segments in agent tracks. For an agent a missing a pose at time t_i, bounded by a preceding valid timestamp t_s and a subsequent valid timestamp t_e, we estimate the intermediate pose $T_a(t_i)$. We compute the translation via linear interpolation and the rotation via Slerp [17] between these two bounding poses. We only fill gaps of up to two frames. This procedure produces the pose correction for accurate DUST-GSG initialization.

Pose-Regularized Joint Optimization. DUST-GSG shares the canonical Gaussians of each dynamic agent across views while maintaining source-specific pose trajectories in dual timelines. Although pose correction provide a good initialization, directly optimizing poses using only image supervision causes drift and frame-to-frame jitter. We therefore introduce a pose regularization loss. Let $T_a^c(t_i^c)$ be the initialized pose of agent a in source $c \in \{\mathrm{veh}, \mathrm{infra}\}$, and $\tilde{T}_a^c(t_i^c)$ be its optimized pose.

First, we force the optimized 3D trajectory to be smooth over time. Specifically, an agent's position at a given frame should closely match the linear interpolation of its past and future positions. For source c and timestamp t_i^c, let $\mathcal{A}_i^c$ be the set of valid dynamic agents. We define the smoothness loss as:

$$\mathcal{L}_{\mathrm{smooth}} = \frac{1}{|\mathcal{A}_i^c|} \sum_{a \in \mathcal{A}_i^c} \left\| \tilde{T}_a^c(t_i^c) - \left[\tilde{T}_a^c(t_{i-1}^c) + w \left(\tilde{T}_a^c(t_{i+1}^c) - \tilde{T}_a^c(t_{i-1}^c) \right) \right] \right\|_2, \tag{6}$$

where $w = \frac{t_i^c - t_{i-1}^c}{t_{i+1}^c - t_{i-1}^c}$. Agents lacking valid past or future poses are excluded from $\mathcal{A}_i^c$ for this calculation. This term is applied independently to both timelines.

Second, we constrain the optimized 3D positions to remain close to their initialized positions during early training:

$$\mathcal{L}_{\mathrm{drift}} = \gamma(s) \frac{1}{|\mathcal{A}_i^c|} \sum_{a \in \mathcal{A}_i^c} \left\| \tilde{T}_a^c(t_i^c) - T_a^c(t_i^c) \right\|_2, \tag{7}$$

where s is the current training step. The weight $\gamma(s)$ linearly decays to zero during training. This prevents poses from deviating before the Gaussian attributes become stable. As training proceeds, the constraint vanishes and image supervision naturally refines the poses.

The final training objective combines image rendering, geometric supervision, standard regularization, and the proposed pose regularization:

$$\mathcal{L} = \mathcal{L}_{\mathrm{image}} + \lambda_{\mathrm{depth}} \mathcal{L}_{\mathrm{depth}} + \lambda_{\mathrm{opacity}} \mathcal{L}_{\mathrm{opacity}} + \mathcal{L}_{\mathrm{reg}} + \lambda_{\mathrm{smooth}} \mathcal{L}_{\mathrm{smooth}} + \lambda_{\mathrm{drift}} \mathcal{L}_{\mathrm{drift}}. \tag{8}$$

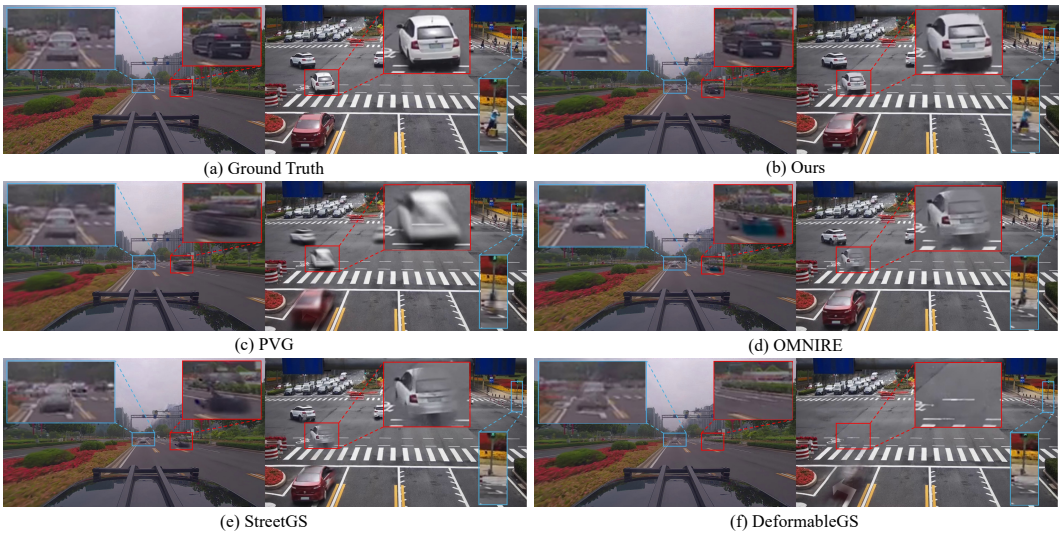
Here, $\mathcal{L}_{\mathrm{image}}$ is the image reconstruction loss based on L1 and SSIM. $\mathcal{L}_{\mathrm{depth}}$ uses sparse LiDAR supervision from both vehicle and infrastructure sensors. $\mathcal{L}_{\mathrm{opacity}}$ regularizes the rendered opacity. $\mathcal{L}_{\mathrm{reg}}$ contains the standard Gaussian regularization terms [3]. Details are provided in the Appendix C.

4 Experiement

In this section, we benchmark the reconstruction capabilities of DUST against prior methods, focusing on dynamic scene reconstruction and novel view synthesis.

4.1 Experimental Setups

Dataset. We examined 95 publicly available V2X-Seq [34] sequences, retained 67 with sufficiently complete and continuous LiDAR observations, and selected 26 based on capture period, weather, traffic density, and dynamic-object distribution, independently of reconstruction performance [30, 27]. Appendix D specifies the eligibility criteria and selected sequence IDs. Images retain their original



(a) Ground Truth (b) Ours

(c) PVG (d) OMNIRE

(e) StreetGS (f) DeformableGS

Figure 3: **Qualitative comparison on dynamic agents under asynchronous observations.** Compared to baselines, our method DUST achieves less ghosting and more high-quality reconstruction of various agents, including vehicles and pedestrians. The insets highlight the details.

resolution of 1080 by 1920. Across the 26 sequences, the median of the per-sequence median absolute recorded camera-timestamp offsets is 97.10 ms, with an interquartile range of 89.09–102.25 ms. The full statistic definition and summary are provided in the appendix.

Evaluation. Each cooperative timestamp provides a paired vehicle and infrastructure image. For scene reconstruction, we use all frames for both training and evaluation. For novel view synthesis, we hold out every 10th timestamp for testing, excluding both the vehicle and infrastructure images at these specific timestamps from the training set. We measure reconstruction quality using PSNR, SSIM, and LPIPS. Since dynamic objects represent the core challenge in asynchronous cooperative reconstruction, we specifically report these metrics on dynamic areas, masked using the provided 2D bounding boxes [34]. Furthermore, we report FVD [20] and RAFT-EPE [18] on these dynamic areas to evaluate temporal and motion consistency.

Baselines. We compare DUST against several Gaussian Splatting approaches: PVG [2], OMNIRE [3], 3DGS [7], CRUISE [25], StreetGS [29], and DeformableGS [32]. Among methods compared, for PVG, OMNIRE, StreetGS and DeformableGS, using their official implementations from the OMNIRE toolbox [3]. For a strictly fair comparison, all baselines are trained from scratch using the same data splits, evaluation protocols, and our regenerated annotations. We additionally evaluate StreetGS with source-specific pose interpolation and a unified continuous trajectory queried at each source's timestamp; details are provided in Appendix D.4.

Implementation Details. We train all methods for 30,000 iterations per scene on a single NVIDIA A100 GPU. For DUST, the regularization coefficents are $\lambda_{\text{smooth}} = 0.01$ and $\lambda_{\text{drift}} = 0.01$. During inference, our model achieves real-time rendering at $\sim$20 FPS at 1080×1920 resolution.

Details on the datasets, metrics, baselines, and implementation can be found in Appendix C and D.

4.2 Main Results

As shown in Tab. 1, DUST achieves state-of-the-art performance in dynamic areas and full images. For instance, our dynamic PSNR in scene reconstruction exceeds the second-best PVG by 3.2 dB. Previous methods like OMNIRE and StreetGS force asynchronous observations into a single timeline. This shared-pose assumption creates optimization conflicts for dynamic agents. By decoupling the timelines, our DUST-GSG directly resolves these conflicts and recovers accurate appearances. Multi-frame metrics further confirm this advantage. DUST improves scene reconstruction FVD by 37.7%, dropping it from 63.7 to 39.7. This temporal stability comes from both the DUST-GSG representation and our pose regularization, which jointly prevent trajectory jitter. Visual comparisons in Fig. 3 align with these numbers. While baselines suffer from severe ghosting and blurring on moving vehicles and pedestrians, DUST renders sharp and artifact-free dynamic agents.

Table 1: **Performance comparison of scene reconstruction and NVS on the V2X-Seq dataset.** DUST outperforms prior Gaussian methods under all conditions, achieving substantial improvement in dynamic areas. Cls and Cond denote sequence categories and conditions. Full averages all 26 sequences. **Red** and **Blue** indicate the best and second-best results in a Cond line or Full line.

Method	Cls.	Cond.	Scene Reconstruction							Novel View Synthesis						
			Full Image			*Dynamic Area*				*Full Image*			*Dynamic Area*			
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	FVD↓	EPE↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	FVD↓	EPE↓
3DGS [7] (2023)	Time	Noon	21.81	0.784	0.318	14.88	0.376	555.0	26.92	18.14	0.736	0.358	13.51	0.362	613.3	56.56
		Dusk	22.10	0.809	0.310	14.03	0.336	465.3	25.84	17.92	0.755	0.358	12.30	0.311	1110.6	63.48
	Weather	Normal	22.57	0.805	0.313	16.67	0.449	345.3	16.90	19.20	0.754	0.342	15.56	0.436	475.9	50.47
		Rainy	23.98	0.853	0.285	17.20	0.512	231.6	20.17	18.62	0.786	0.327	14.95	0.498	612.2	62.22
	Crowd	Low	26.03	0.842	0.280	17.02	0.465	225.6	24.65	22.77	0.803	0.309	15.73	0.459	380.3	71.86
		High	20.92	0.784	0.328	14.98	0.397	794.6	27.28	18.20	0.743	0.357	13.95	0.384	893.1	59.59
	Full		22.72	0.812	0.307	15.82	0.425	433.2	22.62	18.91	0.761	0.343	14.32	0.409	682.2	58.52
DeformGS [32] (2024)	Time	Noon	26.90	0.860	0.141	18.29	0.563	501.2	3.61	24.08	0.768	0.166	16.76	0.418	536.3	15.95
		Dusk	26.35	0.875	0.128	16.24	0.427	445.1	5.54	24.80	0.823	0.140	15.56	0.346	931.1	19.41
	Weather	Normal	27.83	0.884	0.121	19.68	0.625	294.6	3.55	25.09	0.803	0.145	18.43	0.501	419.9	15.33
		Rainy	29.22	0.899	0.131	20.36	0.640	181.2	3.81	26.22	0.835	0.150	18.56	0.511	469.7	16.70
	Crowd	Low	30.51	0.908	0.099	20.13	0.644	194.2	3.04	27.09	0.831	0.120	18.50	0.497	369.3	14.88
		High	25.28	0.841	0.174	17.09	0.504	608.5	5.45	23.38	0.771	0.182	16.22	0.397	724.6	19.53
	Full		27.61	0.877	0.135	18.74	0.572	360.9	4.3	25.04	0.804	0.152	17.46	0.450	564.2	16.84
StreetGS [29] (2024)	Time	Noon	27.96	0.870	0.124	20.81	0.729	172.9	1.97	25.21	0.781	0.146	18.40	0.518	210.1	12.29
		Dusk	28.38	0.889	0.106	20.84	0.715	102.6	2.41	26.48	0.832	0.121	19.29	0.589	234.8	13.80
	Weather	Normal	28.90	0.890	0.108	23.93	0.788	99.5	1.54	26.07	0.809	0.124	21.01	0.592	203.4	7.23
		Rainy	29.68	0.900	0.124	23.23	0.765	60.6	2.00	26.91	0.839	0.145	20.56	0.593	207.4	12.52
	Crowd	Low	30.89	0.903	0.102	23.14	0.771	46.5	2.11	27.71	0.828	0.121	20.28	0.561	123.7	12.21
		High	27.61	0.868	0.131	21.58	0.737	137.6	2.13	25.09	0.784	0.156	19.29	0.550	253.2	13.29
	Full		28.88	0.886	0.116	22.54	0.758	103.0	2.01	26.20	0.812	0.135	20.01	0.572	214.8	11.47
OMNIRE [3] (2025)	Time	Noon	28.16	0.871	0.122	20.97	0.736	158.7	1.85	25.33	0.783	0.141	18.43	0.517	205.5	11.75
		Dusk	28.70	0.889	0.105	21.13	0.727	46.7	2.24	26.74	0.833	0.121	19.53	0.599	71.4	13.22
	Weather	Normal	29.14	0.888	0.108	24.12	0.791	56.6	1.46	26.20	0.810	0.122	21.02	0.590	112.7	7.05
		Rainy	30.06	0.902	0.122	24.04	0.779	35.1	1.79	27.20	0.838	0.146	21.20	0.604	138.2	12.25
	Crowd	Low	31.27	0.904	0.101	23.95	0.792	25.8	2.02	27.95	0.828	0.120	20.94	0.585	78.5	11.76
		High	28.01	0.869	0.129	21.83	0.741	89.3	2.03	25.34	0.788	0.146	19.42	0.551	170.5	12.58
	Full		29.20	0.887	0.114	22.90	0.767	63.7	1.87	26.41	0.813	0.132	20.25	0.579	125.2	10.93
CRUISE [25] (2025)	Time	Noon	23.40	0.799	0.209	17.15	0.490	378.1	5.44	22.60	0.765	0.220	16.71	0.445	440.6	23.65
		Dusk	24.01	0.821	0.193	18.91	0.591	147.9	5.45	23.51	0.804	0.196	16.67	0.444	452.5	24.50
	Weather	Normal	24.19	0.822	0.184	19.39	0.601	277.4	5.33	23.00	0.793	0.199	18.39	0.514	336.7	25.08
		Rainy	25.48	0.856	0.176	18.77	0.569	134.4	5.84	24.62	0.836	0.187	18.27	0.518	313.5	24.81
	Crowd	Low	23.18	0.802	0.204	17.84	0.513	134.8	4.95	25.75	0.831	0.168	18.61	0.542	265.7	23.83
		High	26.98	0.861	0.149	18.96	0.584	360.7	6.38	22.57	0.779	0.215	17.24	0.466	440.7	26.48
	Full		24.47	0.826	0.187	18.57	0.561	240.9	5.74	23.63	0.801	0.199	17.68	0.488	397.6	24.49
PVG [2] (2026)	Time	Noon	28.42	0.858	0.174	22.36	0.734	185.6	2.21	25.87	0.789	0.193	19.70	0.549	323.5	12.38
		Dusk	27.82	0.883	0.148	20.47	0.665	134.9	3.62	25.64	0.830	0.161	18.20	0.491	306.4	17.83
	Weather	Normal	29.01	0.880	0.150	23.74	0.772	137.3	2.23	26.28	0.817	0.159	20.90	0.591	226.5	12.36
		Rainy	30.46	0.900	0.149	24.61	0.789	51.5	2.39	27.37	0.848	0.164	21.38	0.609	194.6	13.75
	Crowd	Low	31.43	0.903	0.120	24.85	0.808	46.1	1.76	27.53	0.835	0.137	21.12	0.602	121.4	14.57
		High	27.06	0.852	0.189	21.27	0.675	248.3	3.86	24.72	0.787	0.203	19.15	0.520	410.0	16.14
	Full		28.97	0.879	0.157	22.91	0.742	131.9	2.77	26.16	0.817	0.171	20.09	0.562	262.3	14.24
Ours (DUST)	Time	Noon	28.75	0.872	0.122	24.11	0.839	78.1	1.65	25.94	0.789	0.137	20.04	0.584	167.5	9.14
		Dusk	29.05	0.892	0.102	24.57	0.844	27.8	2.04	27.03	0.834	0.119	21.50	0.667	69.5	11.90
	Weather	Normal	29.42	0.891	0.105	26.85	0.871	45.1	1.31	26.72	0.817	0.120	22.67	0.641	111.0	6.52
		Rainy	30.36	0.904	0.119	27.86	0.880	21.8	1.62	27.49	0.842	0.141	22.44	0.640	84.6	11.82
	Crowd	Low	31.43	0.906	0.098	27.61	0.878	18.1	1.74	28.18	0.831	0.118	22.40	0.627	46.0	11.44
		High	28.56	0.872	0.125	24.49	0.829	47.4	1.90	25.90	0.794	0.145	21.01	0.613	105.3	10.80
	Full		29.55	0.889	0.112	26.11	0.860	39.7	1.70	26.82	0.818	0.130	21.80	0.631	97.2	9.94

4.3 Robustness under Complex Driving Conditions

Tab. 1 shows our method is robust in complex environments. The gap is clearest in crowded scenes, where dense traffic creates more asynchronous conflicts. Baselines struggle here, but DUST achieves an FVD of 47.4, reducing OMNIRE's score of 89.3 by nearly 47%. We also maintain a strong lead in dynamic areas during rainy weather and at dusk.

While DUST ranks second in a few full-image metrics, the difference is marginal. For example, our full-image PSNR in rainy conditions is only 0.1 dB lower than PVG. This slight drop is expected because full-image scores are dominated by static backgrounds, which existing methods already model well.

4.4 Robustness against Asynchronous Observations

To evaluate robustness against asynchronous data, we introduce time delays between the two observation sources. Specifically, we pair the vehicle frames with infrastructure frames shifted by one to three time steps. This setup creates physical time gaps ranging from zero to 300 milliseconds. Fig. 4 compares DUST and StreetGS using PSNR and SSIM for both full images and dynamic areas. As the time gap increases, StreetGS performance drops rapidly. In contrast, DUST shows a much slower decay. This demonstrates that DUST has stronger robustness to temporal asynchrony. The figure also plots our cumulative advantage. This metric grows steadily with larger time delays. The widening gap confirms that our decoupled design effectively handles severe asynchrony. It successfully prevents the motion artifacts common in single timeline methods.

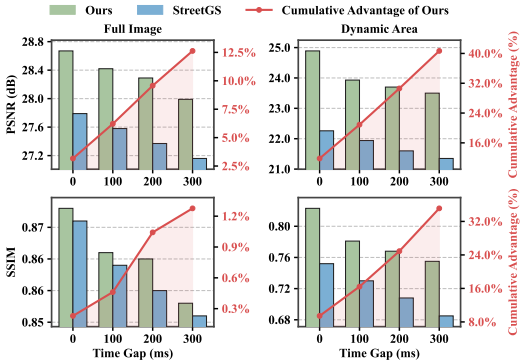


Figure 4: Performance comparison between Ours and StreetGS under increasing temporal asynchrony of 0-300 (Time Gap) ms. The cumulative growth rate of our method over StreetGS on the specified metrics is illustrated by the red curve.

4.5 Ablation Studies

We conduct comprehensive ablation studies to evaluate the contribution of each module, with results shown in Tab. 2 and Fig. 5. The findings demonstrate the critical role of our proposed DUST-GSG representation. Removing either the dual timelines or the separated poses causes severe performance drops, especially in dynamic areas where the vehicle PSNR falls from 25.12 to 23.35 and 22.34, respectively. This confirms that decoupling spatio and temporal states is essential for handling asynchronous cooperative data. The *w/o Dual Timelines* variant retains interpolation at source timestamps and subsequent pose optimization, and therefore does not isolate timestamp interpolation alone. Furthermore, the static anchor-based pose correction provides crucial initialization. Without it, the dynamic PSNR decreases by 1.14 dB. This shows that refining the raw dataset labels improves pose initialization in the evaluated scenes. Without the pose regularation ($\mathcal{L}_{\text{smooth}}$), the LPIPS falls from 0.125 to 0.134. This demonstrates that applying the smooth regularization to the pose is crucial for refined reconstruction. Finally, while removing the pose regularization ($\mathcal{L}_{\text{drift}}$) causes only a slight drop in PSNR, this term is vital for dynamic stability. As shown in Fig. 5e, without this constraint, unconstrained joint optimization leads to trajectory drift and jittering during early training. The regularization ensures the Gaussians learn correct appearances rather than compensating for incorrect positions. Together, these modules allow DUST to effectively reconstruct dynamic objects and establish a strong decoupled paradigm for VICAD reconstruction.

Table 2: Ablations on four randomly selected sequences from the High Crowd subset, using the same training settings as the main experiments. Dual Timelines and Separated Poses are components of DUST-GSG.

Variant	Full Image			Dynamic Area	
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
Full Model	**28.75**	**0.874**	**0.125**	**25.12**	**0.837**
w/o Pose Correction	27.91	0.873	0.128	23.98	0.821
w/o Dual Timelines	28.40	0.872	0.128	23.35	0.783
w/o Separated Poses	28.19	0.870	0.131	22.34	0.742
w/o $\mathcal{L}_{\text{drift}}$	28.72	0.873	0.126	24.99	0.832
w/o $\mathcal{L}_{\text{smooth}}$	28.62	0.869	0.134	24.96	0.833



(a) Full Model (b) w/o Pose Correction (c) w/o Dual Timelines

(d) w/o Separated Pose (e) w/o $\mathcal{L}_{\text{drift}}$ (f) w/o $\mathcal{L}_{\text{smooth}}$

Figure 5: **Qualitative ablation study** results across infrastructure camera from the scene reconstruction task. (a) Our complete model. (b) Without pose initialization before decoupling. (c) Without dual timelines. (d) Without separated poses. (e) Without $\mathcal{L}_{\text{drift}}$. (f) Without $\mathcal{L}_{\text{smooth}}$.

Additional comparisons. On the four sequences used in Table 2, StreetGS with source-specific pose interpolation achieves 21.76 dB dynamic-area PSNR, compared with 25.12 dB for DUST. In the unified continuous trajectory comparison, DUST improves dynamic-area PSNR from 21.11 to 24.89 dB and SSIM from 0.718 to 0.823. Full results are reported in Appendix D.4.

Pose correction robustness. We examine sparse anchors, relaxed motion thresholds, calibration noise, box noise, and incorrect matches. With three static agents in sequence 0007, pose correction improves dynamic-area PSNR from 25.89 to 26.67 dB. It improves alignment IoU under the tested perturbations, although stronger noise degrades the final alignment; details are in Appendix D.5.

5 Conclusion

We present **DUST**. It is a cooperative reconstruction framework designed for asynchronous autonomous driving scenarios. Under the stated assumptions, assigning asynchronous observations to a shared dynamic pose introduces an irreducible photometric residual. To address this, we introduce a decoupled scene graph. We combine this core design with anchor based pose correction and spatio pose regularization. Extensive experiments on the V2X Seq dataset show that our method achieves superior performance across various asynchronous conditions. Our evaluation is limited to V2X-Seq and two camera sources, and does not include an empirical evaluation of the case without reliable anchors. Our current framework also has a limitation in modeling rapid non-rigid motion. Our decoupled design successfully resolves the trajectory mismatch for dynamic agents. However, it does not explicitly handle the shape changes of deformable objects like pedestrians under asynchronous observations. Time delays can still cause appearance blur during rapid limb movements. Future research will explore feature based fine grained temporal deformation modeling.

6 Acknowledgements

This paper is supported in part by the Department of Science and Technology of Guangdong Province (2025QN12X071); the Guangdong and Hong Kong Universities 1+1+1 Joint Research Collaboration Scheme (R002026G0116, R002026B0121 and 658532950); the Seed Project under the City University of Hong Kong (Dongguan) Technology and Innovation Center (TIC2026SP005); the National Natural Science Foundation of China (NSFC, Grant 62606428); start-up funding from the City University of Hong Kong (Dongguan) (B01040000108 and 512780827); the Hong Kong Research Grants Council (GRF 11219624); the Natural Science Foundation of Sichuan Province (2026NSFSC1471); and Young Scientists Fund Program C of the National Natural Science Foundation of China (Grant 62502406).

References

[1] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023.

[2] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *International Journal of Computer Vision*, 134(3):83, 2026.

[3] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban scene reconstruction. In *The Thirteenth International Conference on Learning Representations*, 2025.

[4] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12479–12488, 2023.

[5] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.

[6] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35:4874–4886, 2022.

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

[8] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, page 83–97, 1955.

[9] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34:29541–29552, 2021.

[10] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6498–6508, 2021.

[11] Yang Lu, Jie Wang, Xiaoyun Dong, Ziyao Huang, Bingyi Liu, Jen-Ming Wu, and Jianping Wang. Vi-planning: Infrastructure-assisted real-time planning optimization for autonomous driving. In *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*, pages 923–937, 2025.

[12] Yifan Lu, Quanhao Li, Baoan Liu, Mehrdad Dianati, Chen Feng, Siheng Chen, and Yanfeng Wang. Robust collaborative 3d object detection in presence of pose errors. *arXiv preprint arXiv:2211.07214*, 2022.

[13] 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*, pages 10318–10327, 2021.

[14] Jiayu Qian, Yuwu Lu, Wuyuan Xie, Zhihui Lai, Miaohui Wang, and Xuelong Li. CPSR-CLIP: Conditional prompt-induced style reconstruction for zero-shot domain adaptation. *IEEE Transactions on Multimedia*, 27:7714–7728, 2025.

[15] Dewi Retno Sari Saputro and Purnami Widyaningsih. Limited memory broyden-fletcher-goldfarb-shanno (l-bfgs) method for the parameter estimation on geographically weighted ordinal logistic regression model (gwolr). In *AIP conference proceedings*, volume 1868, page 040009. AIP Publishing LLC, 2017.

[16] Shuyao Shi, Jiahe Cui, Zhehao Jiang, Zhenyu Yan, Guoliang Xing, Jianwei Niu, and Zhenchao Ouyang. Vips: Real-time perception fusion for infrastructure-assisted autonomous driving. In *Proceedings of the 28th annual international conference on mobile computing and networking*, pages 133–146, 2022.

[17] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

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

[19] Haithem Turki, Qi Wu, Xin Kang, Janick Martinez Esturo, Shengyu Huang, Ruilong Li, Zan Gojcic, and Riccardo de Lutio. Simuli: Real-time lidar and camera simulation with unscented transforms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 95374–95387, 2026.

[20] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[21] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[22] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024.

[23] Tao Wu, Alexandra Gomez-Villa, Senmao Li, Yaxing Wang, Joost van de Weijer, and Kai Wang. DiDE: Direct injection with color-texture DEcoupling for 3D stylization. *arXiv preprint arXiv:2610.02044*, 2026.

[24] Hao Xiang, Zhaoliang Zheng, Xin Xia, Runsheng Xu, Letian Gao, Zewei Zhou, Xu Han, Xinkai Ji, Mingxi Li, Zonglin Meng, et al. V2x-real: a largs-scale dataset for vehicle-to-everything cooperative perception. In *European Conference on Computer Vision*, pages 455–470. Springer, 2024.

[25] Haoran Xu, Saining Zhang, Peishuo Li, Baijun Ye, Xiaoxue Chen, Huan-ang Gao, Jv Zheng, Xiaowei Song, Ziqiao Peng, Run Miao, et al. Cruise: Cooperative reconstruction and editing in v2x scenarios using gaussian splatting. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12518–12525. IEEE, 2025.

[26] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision*, pages 107–124. Springer, 2022.

[27] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589. IEEE, 2022.

[28] Yunjiang Xu, Lingzhi Li, Jin Wang, Yupeng Ouyang, and Benyuan Yang. Instinct: Instance-level interaction architecture for query-based collaborative perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25464–25473, 2025.

[29] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pages 156–173. Springer, 2024.

[30] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023.

[31] Lei Yang, Kaicheng Yu, Tao Tang, Jun Li, Kun Yuan, Li Wang, Xinyu Zhang, and Peng Chen. Bevheight: A robust framework for vision-based roadside 3d object detection. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, March 2023.

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

[33] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21361–21370, 2022.

[34] Haibao Yu, Wenxian Yang, Hongzhi Ruan, Zhenwei Yang, Yingjuan Tang, Xu Gao, Xin Hao, Yifeng Shi, Yifeng Pan, Ning Sun, Juan Song, Jirui Yuan, Ping Luo, and Zaiqing Nie. V2x-seq: A large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[35] Haibao Yu, Wenxian Yang, Jiaru Zhong, Zhenwei Yang, Siqi Fan, Ping Luo, and Zaiqing Nie. End-to-end autonomous driving through v2x cooperation. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 2025.

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[37] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024.

[38] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

Appendices

A Notation **14**

B Proof **16**

B.1 Formal Rendering Model and Assumptions 16

B.2 Jacobian Structure and Rendering Fisher Information 17

B.3 Proof of Part (i): Irreducible Residual under Single-Timeline 17

B.4 Proof of Part (ii): Gradient Incompatibility 19

B.5 Proof of Part (iii): NTK Block-Diagonalization 19

C Implementation Details **21**

C.1 Initialization 21

C.2 Training 21

C.3 Optimization 22

D Evaluation **22**

D.1 Datasets 22

D.2 Metrics 23

D.3 Baselines 24

D.4 Additional Asynchrony-Aware Baselines 25

D.5 Pose-Correction Robustness 25

E Scope and Limitations **26**

F Broader Impacts **27**

G Safeguards **27**

A Notation

Table 3: Notation used in the main paper and appendix.

Notation	*Definition*
INDICES & TEMPORAL QUANTITIES	
$c \in \{v, f\}$	Source index: v for vehicle, f for infrastructure
$n = 1, \ldots, N$	Index over canonical Gaussians of a single dynamic agent
t_i	i-th cooperative anchor timestamp
τ_c	Actual sensor capture time of source c
$\Delta\tau$	Temporal offset between sources: $\Delta\tau = \tau_v - \tau_f$
$v \in \mathbb{R}^3$	Agent velocity vector under locally rigid motion
CANONICAL-SPACE PARAMETERS (ORANGE)	

Table 3 (continued)

Notation	*Definition*
$\bar{\mu}_n \in \mathbb{R}^3$	Canonical mean of Gaussian n in the agent's body frame
$\bar{\mu}_n^* \in \mathbb{R}^3$	Ground-truth canonical mean of Gaussian n; shared across all sources
$\bar{\Sigma}_n \in \mathbb{S}^3_{++}$, $\bar{\alpha}_n$, $\bar{c}_n$	Canonical covariance, opacity, and color coefficient of Gaussian n
$w_n^c(u)$	Alpha-compositing blending weight of Gaussian n at pixel u from source c
$\phi_n^c(u) \in \mathbb{R}^3$	Pixel-space influence vector of Gaussian n at pixel u from source c (Eq. 15)
$\mathbf{A}_n^c \in \mathbb{S}^3_+$	Rendering Fisher information matrix: $\mathbf{A}_n^c = \sum_{u \in \mathcal{U}^c} \phi_n^c(u)\, \phi_n^c(u)^\top$ (Def. 5)
$\lambda_{\min,n}, \lambda_n$	Minimum Fisher eigenvalue: $\min\big(\lambda_{\min}(\mathbf{A}_n^v),\ \lambda_{\min}(\mathbf{A}_n^f)\big)$
POSE & WORLD-SPACE QUANTITIES (BLUE)	
$R^c \in SO(3)$, $T^c \in \mathbb{R}^3$	Rotation and translation of source c at capture time τ_c
R_i, T_i	Shared rotation and translation in the single-timeline formulation at anchor t_i
$\mu_n^c \in \mathbb{R}^3$	World-space mean of Gaussian n from source c: $\mu_n^c = R^c \bar{\mu}_n + T^c$
$\mu_n^{*,c} \in \mathbb{R}^3$	Ground-truth world position of Gaussian n at capture time τ_c
$\mu_n^\dagger \in \mathbb{R}^3$	Fisher-weighted optimal world position under single-timeline (Eq. 24)
$\delta_n^c \in \mathbb{R}^3$	World-space positional error: $\delta_n^c = \mu_n^c - \mu_n^{*,c}$
$\delta_n^{c\dagger} \in \mathbb{R}^3$	Residual positional error at the single-timeline optimum $\mu_n^\dagger$
$J_{\pi,n}^c \in \mathbb{R}^{2\times 3}$	Jacobian of camera projection at μ_n^c: $J_{\pi,n}^c = \partial \hat{u}_n^c / \partial \mu_n^c$
$\hat{u}_n^c \in \mathbb{R}^2$, $\hat{\Sigma}_n^c \in \mathbb{S}^2_{++}$	Projected 2D mean and covariance of Gaussian n in source c's image plane (EWA splatting)
$\xi^v, \xi^f \in \mathbb{R}^6$	Source-specific pose parameters in $\mathfrak{se}(3)$ under DUST-GSG; collapsed to a single shared ξ under single-timeline
NTK QUANTITIES	
$\theta = (\bar{\boldsymbol{\mu}}, \boldsymbol{\xi})$	Full optimizable parameter vector
$\Theta \in \mathbb{R}^{M\times M}$	Empirical Neural Tangent Kernel matrix
$\Theta^{\bar{\mu}}$, Θ^{ξ}	Canonical-parameter and pose-parameter additive components of Θ
$\Theta_{vv}^{\xi^v}$, $\Theta_{ff}^{\xi^f}$	Vehicle and infrastructure diagonal blocks of the DUST-GSG pose kernel
$\mathbf{r}^c(t)$	Pixel residual vector of source c at training step t: $\mathbf{r}^c(t) = \hat{\mathbf{I}}^c(t) - \mathbf{I}^{c*}$
RENDERING & PHOTOMETRIC LOSS	
$\hat{I}^c(u)$	Rendered color at pixel u from source c
$I^{c*}(u)$	Ground-truth image captured at time τ_c
$\mathcal{U}^c$	Set of pixels observing the agent from source c
$\mathcal{L}^c$	Per-source photometric loss
$\mathcal{L}_n^c$	Per-Gaussian per-source loss (quadratic form)
$\mathcal{L}^*_{\text{single}}$	Optimal reconstruction loss under single-timeline
$\mathcal{L}^*_{\text{DST}}$	Optimal reconstruction loss under DUST
LOSS FUNCTIONS & REGULARIZATION	
$\mathcal{L}_{\text{image}}$	Image loss: $(1-\lambda_r)\mathcal{L}_1 + \lambda_r \mathcal{L}_{\text{SSIM}}$
$\mathcal{L}_1$	L1 reconstruction loss
$\mathcal{L}_{\text{SSIM}}$	SSIM reconstruction loss
$\lambda_r = 0.2$	Weight of the SSIM loss
$\mathcal{L}_{\text{depth}}$	Depth map loss: $\frac{1}{hw}\sum \lVert \mathcal{D}^s - \hat{\mathcal{D}} \rVert_1$
$\mathcal{D}^s$	Inverse of the sparse LiDAR depth map
$\hat{\mathcal{D}}$	Inverse of the predicted depth map
$\mathcal{L}_{\text{opacity}}$	Mask/opacity loss (sky + occupancy)
O_g	Rendered opacity map
M_{sky}	Sky mask
$\mathcal{L}_{\text{reg}}$	Pose smoothness regularization
$\boldsymbol{\theta}(t)$	Human body poses at time t
δ	Randomly chosen integer from $\{1,2,3,4,5\}$ for pose regularization
$\lambda_{\text{depth}} = 0.01$	Weight of depth loss
$\lambda_{\text{opacity}} = 0.05$	Weight of opacity loss
$\lambda_{\text{reg}} = 0.01$	Weight of pose smoothness loss
EVALUATION METRICS	
$\hat{\mathbf{I}}$	Rendered image
$\mathbf{I}$	Ground-truth image
N	Number of valid pixels (in PSNR computation)
μ_x, μ_y	Local means of image patches $\mathbf{x}, \mathbf{y}$ (SSIM)
σ_x, σ_y	Local standard deviations (SSIM)

Table 3 (continued)

Notation	*Definition*
σ_{xy}	Local cross-covariance (SSIM)
$c_1,\ c_2$	Small constants for numerical stability (SSIM)
$\phi_\ell(\cdot)$	Feature map from layer ℓ of AlexNet (LPIPS)
$\mathbf{w}_\ell$	Learned channel-wise weights at layer ℓ (LPIPS)
$H_\ell,\ W_\ell$	Spatial dimensions of feature map at layer ℓ (LPIPS)
$\boldsymbol{\mu}_{\text{gen}},\ \boldsymbol{\Sigma}_{\text{gen}}$	Mean and covariance of generated video clip features (FVD)
$\boldsymbol{\mu}_{\text{real}},\ \boldsymbol{\Sigma}_{\text{real}}$	Mean and covariance of real video clip features (FVD)
T	Total number of video frames
$T_c = 16$	Length of each non-overlapping video clip (FVD)
$\mathbf{F}_t \in \mathbb{R}^{H\times W\times 2}$	Dense optical flow field between frames t and $t+1$
(u, v)	Horizontal and vertical components of optical flow
$(\hat{u}, \hat{v})$	Predicted flow components (rendered video)
$\mathcal{M}$	Union of dynamic masks from two consecutive frames (EPE)

B Proof

We analyze the photometric residual of the shared-pose formulation and the decoupling of source-specific pose parameters under the assumptions stated below. The proof proceeds in five stages: we first formalize the rendering model and state the working assumptions (§B.1); then derive the Jacobian structure and the rendering Fisher information matrix (§B.2); next prove the irreducible-residual lower bound (§B.3); then establish gradient incompatibility (§B.4); and finally show how DUST-GSG resolves both issues via NTK block-diagonalization (§B.5).

B.1 Formal Rendering Model and Assumptions

Throughout the proof, orange denotes *canonical-space* quantities (those defined in the agent's body frame, independent of capture time), and blue denotes *pose/world-space* quantities (those that depend on the source-specific rigid-body transform or the camera).

Definition 2 (Canonical Gaussian Agent)**.** *A dynamic agent is represented by N canonical 3D Gaussians with means $\bar{\boldsymbol{\mu}} = (\bar{\mu}_1, \dots, \bar{\mu}_N) \in \mathbb{R}^{3N}$, covariances $\bar{\boldsymbol{\Sigma}} = (\bar{\Sigma}_1, \dots, \bar{\Sigma}_N)$ with each $\bar{\Sigma}_n \in \mathbb{S}^3_{++}$, opacities $\bar{\boldsymbol{\alpha}} = (\bar{\alpha}_1, \dots, \bar{\alpha}_N) \in (0,1)^N$, and view-dependent color coefficients $\bar{\mathbf{c}} = (\bar{c}_1, \dots, \bar{c}_N)$.*

Definition 3 (Source-Specific World Projection)**.** *For source $c \in \{v, f\}$ (vehicle, infrastructure) with rigid-body pose $P^c = (R^c, T^c) \in SE(3)$ at capture time τ_c, the world-space parameters of Gaussian n are*

$$\mu_n^c = R^c\,\bar{\mu}_n + T^c, \qquad \Sigma_n^c = R^c\,\bar{\Sigma}_n\,(R^c)^\top. \tag{9}$$

Definition 4 (Differentiable Splatting Renderer)**.** *Given camera intrinsics $K^c \in \mathbb{R}^{3\times 3}$ and extrinsics $W^c \in SE(3)$, the rendered color at pixel $u \in \mathbb{R}^2$ from source c is*

$$\hat{I}^c(u) = \sum_{n=1}^{N} \underbrace{\bar{\alpha}_n \prod_{m<n} \big(1 - \bar{\alpha}_m\,\mathcal{G}_m^c(u)\big)}_{w_n^c(u)}\ \bar{c}_n\ \mathcal{G}_n^c(u), \tag{10}$$

where $\mathcal{G}_n^c(u) = \exp\!\big(-\frac{1}{2}(u-\hat{u}_n^c)^\top(\hat{\Sigma}_n^c)^{-1}(u-\hat{u}_n^c)\big)$ is the 2D Gaussian evaluated at pixel u, with projected mean $\hat{u}_n^c = \pi(K^c W^c \mu_n^c)$ and projected covariance $\hat{\Sigma}_n^c$ obtained via EWA splatting. The blending weight $w_n^c(u)$ encapsulates the front-to-back alpha-compositing logic.

Assumption 1 (Linearized Rendering Regime)**.** *We approximate the rendering function by its first-order Taylor expansion around the current parameters θ_0. This characterizes the early training phase where gradient conflicts are most damaging:*

$$\hat{I}^c(u;\theta) \ \approx\ \hat{I}^c(u;\theta_0) + \big\langle \nabla_\theta \hat{I}^c(u;\theta_0),\ \theta - \theta_0 \big\rangle. \tag{11}$$

Assumption 2 (Locally Rigid Motion)**.** *During the temporal offset $\Delta\tau = \tau_v - \tau_f$ between the two sources, the agent undergoes approximately constant-velocity motion with velocity $v \in \mathbb{R}^3$:*

$$\mu_n^*(\tau_f) \ \approx\ \mu_n^*(\tau_v) + v\cdot\Delta\tau, \qquad \forall\, n = 1, \dots, N, \tag{12}$$

where $\mu_n^*(\tau_c)$ denotes the ground-truth world position of Gaussian n at time τ_c.

B.2 Jacobian Structure and Rendering Fisher Information

From the rendering equation (10), the gradient of $\hat{I}^c(u)$ with respect to $\bar{\mu}_n$ factors through three stages via the chain rule:

$$\nabla_{\bar{\mu}_n} \hat{I}^c(u) \;=\; \underbrace{(R^c)^\top}_{\text{world}\to\text{canonical}} \cdot \underbrace{(J^c_{\pi,n})^\top}_{\text{image}\to\text{world}} \cdot \underbrace{\nabla_{\hat{u}^c_n} \hat{I}^c(u)}_{\text{color}\to\text{image}} \;\in \mathbb{R}^3, \tag{13}$$

where $J^c_{\pi,n} = \frac{\partial \hat{u}^c_n}{\partial \mu^c_n} \in \mathbb{R}^{2\times 3}$ is the Jacobian of the camera projection evaluated at μ^c_n, and $\nabla_{\hat{u}^c_n} \hat{I}^c(u) \in \mathbb{R}^2$ is the sensitivity of the rendered color to the projected Gaussian center.

Differentiating Eq. (10) with respect to $\hat{u}^c_n$:

$$\nabla_{\hat{u}^c_n} \hat{I}^c(u) \;\approx\; w^c_n(u)\, \bar{c}_n\, \mathcal{G}^c_n(u)\, (\hat{\Sigma}^c_n)^{-1}(u - \hat{u}^c_n) \;\in \mathbb{R}^2. \tag{14}$$

The sensitivity is proportional to the blending weight $w^c_n(u)$, the Gaussian kernel value $\mathcal{G}^c_n(u)$, and the Mahalanobis displacement $(u - \hat{u}^c_n)$ scaled by the inverse projected covariance. Pixels far from the Gaussian center or occluded by other Gaussians contribute negligibly.

Substituting Eq. (14) into Eq. (13) and defining the *pixel-space influence vector*

$$\phi^c_n(u) \;:=\; (J^c_{\pi,n})^\top\, w^c_n(u)\, \bar{c}_n\, \mathcal{G}^c_n(u)\, (\hat{\Sigma}^c_n)^{-1}(u - \hat{u}^c_n) \;\in \mathbb{R}^3, \tag{15}$$

we obtain the compact form

$$\nabla_{\bar{\mu}_n} \hat{I}^c(u) \;=\; (R^c)^\top\, \phi^c_n(u). \tag{16}$$

By Eq. (16), the gradient in canonical space equals the world-space influence vector $\phi^c_n(u)$ rotated back by $(R^c)^\top$. This factorization is the structural key of the entire proof: since $R^v \neq R^f$ in general, the same canonical parameter $\bar{\mu}_n$ is pulled in two *different rotated directions* by the two sources—the geometric root of the gradient conflict exposed in Part (ii).

Consider the per-source photometric loss $\mathcal{L}^c = \frac{1}{2}\sum_{u\in\mathcal{U}^c}(\hat{I}^c(u) - I^{c*}(u))^2$, where $\mathcal{U}^c$ is the set of pixels observing the agent from source c and I^{c*} is the ground-truth image captured at time τ_c. The gradient with respect to $\bar{\mu}_n$ is

$$\nabla_{\bar{\mu}_n} \mathcal{L}^c = (R^c)^\top \sum_{u\in\mathcal{U}^c} \big(\hat{I}^c(u) - I^{c*}(u)\big)\, \phi^c_n(u). \tag{17}$$

Under Assumption 1, the photometric residual at pixel u is dominated by the world-space positional error $\delta^c_n = \mu^c_n - \mu^{*,c}_n$ of Gaussian n. A first-order expansion of the rendering equation around the true position yields $\hat{I}^c(u) - I^{c*}(u) \approx \phi^c_n(u)^\top \delta^c_n$ (contribution from Gaussian n). Substituting into Eq. (17), the gradient becomes

$$\nabla_{\bar{\mu}_n} \mathcal{L}^c \;\approx\; (R^c)^\top \underbrace{\left(\sum_{u\in\mathcal{U}^c} \phi^c_n(u)\, \phi^c_n(u)^\top\right)}_{\mathbf{A}^c_n \;\in\; \mathbb{R}^{3\times 3}} \delta^c_n. \tag{18}$$

Definition 5 (Rendering Fisher Information Matrix)**.** *The* rendering Fisher information matrix *of Gaussian n from source c is*

$$\mathbf{A}^c_n \;=\; \sum_{u\in\mathcal{U}^c} \phi^c_n(u)\, \phi^c_n(u)^\top \;\in\; \mathbb{S}^3_+. \tag{19}$$

Being a sum of rank-1 outer products, $\mathbf{A}^c_n$ is positive semi-definite by construction. It becomes strictly positive definite ($\mathbf{A}^c_n \succ 0$) whenever $\{\phi^c_n(u)\}_{u\in\mathcal{U}^c}$ spans $\mathbb{R}^3$, which holds when the agent is observed from a non-degenerate viewpoint with sufficient pixel coverage. We assume $\mathbf{A}^c_n \succ 0$ throughout. Combining Eq. (19) with the residual approximation, the per-source loss for Gaussian n takes the quadratic form

$$\mathcal{L}^c_n \;\approx\; \frac{1}{2}\, (\delta^c_n)^\top\, \mathbf{A}^c_n\, \delta^c_n, \tag{20}$$

and the total per-source loss is $\mathcal{L}^c = \sum_{n=1}^N \mathcal{L}^c_n$. This quadratic structure is the foundation for all subsequent analyses.

B.3 Proof of Part (i): Irreducible Residual under Single-Timeline

In the single-timeline formulation, both sources share pose $P(t_i) = (R_i, T_i)$ at the anchor timestamp t_i, so the world-space position of every Gaussian is the same for both sources: $\mu^v_n = \mu^f_n = R_i\, \bar{\mu}_n + T_i$. However, the ground-truth positions differ between the two capture times. Under Assumption 2:

$$\mu^{*,v}_n = \mu^*_n(t_i) + v(\tau_v - t_i), \qquad \mu^{*,f}_n = \mu^*_n(t_i) + v(\tau_f - t_i). \tag{21}$$

Denoting the shared world position as $\mu_n = R_i\bar{\mu}_n + T_i$, the positional errors for the two sources are

$$\delta_n^v = \mu_n - \mu_n^{*,v} = \big(\mu_n - \mu_n^*(t_i)\big) - v(\tau_v - t_i),$$
$$\delta_n^f = \mu_n - \mu_n^{*,f} = \big(\mu_n - \mu_n^*(t_i)\big) - v(\tau_f - t_i). \tag{22}$$

Crucially, the difference between the two errors is *independent of the parameters*:

$$\delta_n^v - \delta_n^f \;=\; -v\,\Delta\tau, \qquad \Delta\tau = \tau_v - \tau_f. \tag{23}$$

This identity is the root cause of the optimization conflict: no matter how we adjust $\bar{\mu}_n$, the two positional errors always differ by the fixed vector $-v\Delta\tau$. The optimizer therefore faces an irreconcilable constraint—reducing δ_n^v necessarily increases δ_n^f by the same amount, and vice versa.

The total loss for Gaussian n under single-timeline is $\mathcal{L}_n = \mathcal{L}_n^v + \mathcal{L}_n^f = \frac{1}{2}(\delta_n^v)^\top \mathbf{A}_n^v \delta_n^v + \frac{1}{2}(\delta_n^f)^\top \mathbf{A}_n^f \delta_n^f$. We minimize over $\mu_n \in \mathbb{R}^3$ by setting $\nabla_{\mu_n}\mathcal{L}_n = \mathbf{A}_n^v\delta_n^v + \mathbf{A}_n^f\delta_n^f = \mathbf{0}$, and solving for the optimal position:

$$\mathbf{A}_n^v\,(\mu_n - \mu_n^{*,v}) + \mathbf{A}_n^f\,(\mu_n - \mu_n^{*,f}) = \mathbf{0}$$
$$\big(\mathbf{A}_n^v + \mathbf{A}_n^f\big)\mu_n = \mathbf{A}_n^v\,\mu_n^{*,v} + \mathbf{A}_n^f\,\mu_n^{*,f}$$
$$\mu_n^\dagger = \big(\mathbf{A}_n^v + \mathbf{A}_n^f\big)^{-1}\big(\mathbf{A}_n^v\,\mu_n^{*,v} + \mathbf{A}_n^f\,\mu_n^{*,f}\big). \tag{24}$$

This is the Fisher-weighted average of the two target positions—a compromise that is optimal for neither source individually. Substituting $\mu_n^\dagger$ back, the residual errors at the optimum are

$$\begin{aligned}\delta_n^{v\dagger} &= \mu_n^\dagger - \mu_n^{*,v}\\ &= \big(\mathbf{A}_n^v + \mathbf{A}_n^f\big)^{-1}\big(\mathbf{A}_n^v\,\mu_n^{*,v} + \mathbf{A}_n^f\,\mu_n^{*,f}\big) - \mu_n^{*,v}\\ &= \big(\mathbf{A}_n^v + \mathbf{A}_n^f\big)^{-1}\mathbf{A}_n^f\,(\mu_n^{*,f} - \mu_n^{*,v})\\ &= \big(\mathbf{A}_n^v + \mathbf{A}_n^f\big)^{-1}\mathbf{A}_n^f\,(-v\Delta\tau),\end{aligned} \tag{25}$$

where the last step uses $\mu_n^{*,f} - \mu_n^{*,v} = -v\Delta\tau$ from Eqs. (21)–(23). Similarly, $\delta_n^{f\dagger} = (\mathbf{A}_n^v + \mathbf{A}_n^f)^{-1}\mathbf{A}_n^v(v\Delta\tau)$. Both residual errors are non-zero whenever $v\Delta\tau \neq \mathbf{0}$, confirming that the Fisher-weighted compromise leaves both sources with unresolved positional error.

We now derive the lower bound in two steps.

Step A (eigenvalue inequality). For any $M \succ 0$ and any vector x, $x^\top Mx \geq \lambda_{\min}(M)\|x\|^2$. Applying this at the optimum:

$$\begin{aligned}\mathcal{L}_n^* &= \frac{1}{2}(\delta_n^{v\dagger})^\top \mathbf{A}_n^v\,\delta_n^{v\dagger} + \frac{1}{2}(\delta_n^{f\dagger})^\top \mathbf{A}_n^f\,\delta_n^{f\dagger}\\ &\geq \frac{\lambda_{\min}(\mathbf{A}_n^v)}{2}\|\delta_n^{v\dagger}\|^2 + \frac{\lambda_{\min}(\mathbf{A}_n^f)}{2}\|\delta_n^{f\dagger}\|^2\\ &\geq \frac{\lambda_{\min,n}}{2}\big(\|\delta_n^{v\dagger}\|^2 + \|\delta_n^{f\dagger}\|^2\big),\end{aligned} \tag{26}$$

where $\lambda_{\min,n} = \min\big(\lambda_{\min}(\mathbf{A}_n^v), \lambda_{\min}(\mathbf{A}_n^f)\big) > 0$.

Step B (parallelogram identity). For any two vectors $a, b \in \mathbb{R}^3$, $\|a\|^2 + \|b\|^2 = 2\|(a+b)/2\|^2 + \|a-b\|^2/2 \geq \|a-b\|^2/2$. Setting $a = \delta_n^{v\dagger}$, $b = \delta_n^{f\dagger}$, and using $\delta_n^{v\dagger} - \delta_n^{f\dagger} = -v\Delta\tau$:

$$\|\delta_n^{v\dagger}\|^2 + \|\delta_n^{f\dagger}\|^2 \;\geq\; \frac{\|\delta_n^{v\dagger} - \delta_n^{f\dagger}\|^2}{2} = \frac{\|v\Delta\tau\|^2}{2} = \frac{\|v\|^2|\Delta\tau|^2}{2}. \tag{27}$$

Combining Steps A and B and summing over all N Gaussians:

$$\mathcal{L}_{\text{single}}^* = \sum_{n=1}^N \mathcal{L}_n^* \;\geq\; \frac{\|v\|^2|\Delta\tau|^2}{4}\sum_{n=1}^N \lambda_{\min,n} \;>\; 0. \tag{28}$$

This completes the proof of Part (i). The bound is strict and *representation-level*: no choice of canonical parameters can reduce the photometric loss below this threshold, regardless of the optimizer used. The irreducible error scales quadratically with both the agent velocity $\|v\|$ and the temporal offset $|\Delta\tau|$, and linearly with the aggregate Fisher information $\sum_n \lambda_{\min,n}$—all quantities that are large in typical high-speed VICAD scenarios.

The formula for the theorem 1 in the Sec 3.2 is obtained by reformulating $\lambda_{\min,n}$ as λ_n.

$$\mathcal{L}_{\text{single}}^* \;\geq\; \frac{|\Delta\tau|^2\|v\|^2}{4}\sum_{n=1}^N \lambda_n \;>\; 0, \tag{29}$$

where $\lambda_n > 0$ is the minimum Fisher-information eigenvalue of $\mathcal{G}_n$, quantifying its photometric sensitivity to displacement.

B.4 Proof of Part (ii): Gradient Incompatibility

We now show that the two per-source gradients cannot simultaneously vanish at any canonical parameter, and quantify the separation between their individual optima. From Eq. (18), the gradient of $\mathcal{L}^c$ with respect to $\bar{\mu}_n$ vanishes when

$$\begin{aligned}\nabla_{\bar{\mu}_n}\mathcal{L}^c = \mathbf{0} &\iff (R_i)^\top \mathbf{A}_n^c\, \delta_n^c = \mathbf{0} \\ &\iff \mathbf{A}_n^c\, \delta_n^c = \mathbf{0} \\ &\iff \delta_n^c = \mathbf{0} \\ &\iff \mu_n = \mu_n^{*,c},\end{aligned} \tag{30}$$

where the second equivalence uses $R_i \in SO(3)$ (invertible) and the third uses $\mathbf{A}_n^c \succ 0$ (invertible). Thus $\mathcal{L}^v$ is minimized at $\mu_n = \mu_n^{*,v}$ and $\mathcal{L}^f$ is minimized at $\mu_n = \mu_n^{*,f}$.

Since $\mu_n = R_i\bar{\mu}_n + T_i$ is a bijection in $\bar{\mu}_n$, the per-source canonical optima are $\bar{\mu}_n^{*,v} = {R_i}^\top(\mu_n^{*,v} - T_i)$ and $\bar{\mu}_n^{*,f} = {R_i}^\top(\mu_n^{*,f} - T_i)$. Their separation is

$$\begin{aligned}\left\|\bar{\mu}_n^{*,v} - \bar{\mu}_n^{*,f}\right\| &= \left\|{R_i}^\top(\mu_n^{*,v} - \mu_n^{*,f})\right\| \\ &= \left\|\mu_n^{*,v} - \mu_n^{*,f}\right\| \qquad (R_i \in SO(3) \text{ is isometric}) \\ &= \|v\,\Delta\tau\| \;=\; \|v\|\,|\Delta\tau|.\end{aligned} \tag{31}$$

Since $v \neq \mathbf{0}$ and $\Delta\tau \neq 0$, this separation is strictly positive, so no single $\bar{\mu}_n$ can simultaneously satisfy both $\nabla_{\bar{\mu}_n}\mathcal{L}^v = \mathbf{0}$ and $\nabla_{\bar{\mu}_n}\mathcal{L}^f = \mathbf{0}$. The separation $\|v\|\,|\Delta\tau|$ is the physical displacement of the agent between the two capture times projected into canonical space. For a vehicle traveling at $\|v\| = 10$ m/s with $|\Delta\tau| = 70$ ms, this equals 0.7 m—far exceeding the typical Gaussian kernel radius in driving scenes. The optimizer is forced to place $\bar{\mu}_n$ at the Fisher-weighted compromise (Eq. (24)), creating blurred, ghosted reconstructions that satisfy neither view.

Gradient anti-alignment in the worst case. When t_i lies between the two capture times, i.e., $(\tau_v - t_i)(\tau_f - t_i) < 0$, the positional offsets δ_n^v and δ_n^f at the anchor position $\mu_n = \mu_n^*(t_i)$ point in opposite directions along v. Under the isotropic Fisher approximation $\mathbf{A}_n^v \approx \mathbf{A}_n^f \approx \lambda_n I_3$, the per-source gradients from Eq. (18) become

$$\begin{aligned}\nabla_{\bar{\mu}_n}\mathcal{L}^v &\approx (R_i)^\top \mathbf{A}_n^v\, \delta_n^v \\ &= \lambda_n (R_i)^\top \left(\mu_n^*(t_i) - \mu_n^{*,v}\right) \\ &= \lambda_n (R_i)^\top \left(-v(\tau_v - t_i)\right) \quad \text{[from Eq. (21)]} \\ &= -\lambda_n (R_i)^\top v(\tau_v - t_i),\end{aligned} \tag{32}$$

and analogously $\nabla_{\bar{\mu}_n}\mathcal{L}^f \approx -\lambda_n (R_i)^\top v(\tau_f - t_i)$. The cosine similarity between the two gradient vectors is

$$\begin{aligned}\cos\angle\left(\nabla_{\bar{\mu}}\mathcal{L}^v,\ \nabla_{\bar{\mu}}\mathcal{L}^f\right) &= \frac{\nabla_{\bar{\mu}_n}\mathcal{L}^v \cdot \nabla_{\bar{\mu}_n}\mathcal{L}^f}{\left\|\nabla_{\bar{\mu}_n}\mathcal{L}^v\right\|\left\|\nabla_{\bar{\mu}_n}\mathcal{L}^f\right\|} \\ &= \frac{\lambda_n^2\left\|(R_i)^\top v\right\|^2(\tau_v - t_i)(\tau_f - t_i)}{\lambda_n^2\left\|(R_i)^\top v\right\|^2|\tau_v - t_i|\,|\tau_f - t_i|} \\ &= \frac{(\tau_v - t_i)(\tau_f - t_i)}{|\tau_v - t_i|\,|\tau_f - t_i|} \;=\; -1,\end{aligned} \tag{33}$$

where the last equality holds because $(\tau_v - t_i)$ and $(\tau_f - t_i)$ have opposite signs, and $\|(R_i)^\top v\| = \|v\| > 0$ since $R_i \in SO(3)$ is isometric. This is the most severe form of gradient conflict: the two sources provide diametrically opposed update signals to the shared canonical parameters. Even when the anchor does not lie between the capture times, the cosine similarity satisfies

$$\cos\angle\left(\nabla_{\bar{\mu}}\mathcal{L}^v,\ \nabla_{\bar{\mu}}\mathcal{L}^f\right) = \frac{(\tau_v - t_i)(\tau_f - t_i)}{|\tau_v - t_i|\,|\tau_f - t_i|} < +1 \quad \text{whenever } \Delta\tau \neq 0, \tag{34}$$

ensuring persistent gradient interference regardless of anchor placement. □

B.5 Proof of Part (iii): NTK Block-Diagonalization

We now show that DUST-GSG simultaneously eliminates the irreducible residual and the gradient conflict by introducing source-specific pose degrees of freedom.

Under DUST-GSG, source c uses its own pose $P^c(\tau_c) = (R^c, T^c)$ at its actual capture time τ_c, so the world-space position of Gaussian n is $\mu_n^c = R^c\,\bar{\mu}_n + T^c$. When the poses match the ground truth, the positional error for source c becomes

$$\begin{aligned}\delta_n^c &= \mu_n^c - \mu_n^{*,c} \\ &= R^*(\tau_c)\,\bar{\mu}_n + T^*(\tau_c) - \big(R^*(\tau_c)\,\bar{\mu}_n^* + T^*(\tau_c)\big) \\ &= R^*(\tau_c)\big(\bar{\mu}_n - \bar{\mu}_n^*\big),\end{aligned} \tag{35}$$

where $\bar{\mu}_n^*$ is the true canonical position, shared by both sources since it encodes the agent's intrinsic geometry independent of capture time. Setting $\bar{\mu}_n = \bar{\mu}_n^*$ therefore yields $\delta_n^c = \mathbf{0}$ for *both* sources simultaneously:

$$\bar{\mu}_n = \bar{\mu}_n^* \implies \delta_n^v = \delta_n^f = \mathbf{0} \implies \mathcal{L}_{\text{DST}}^* = \sum_{n=1}^{N}\big(\mathcal{L}_n^v + \mathcal{L}_n^f\big) = 0. \tag{36}$$

The key mechanism is that the source-specific poses P^v, P^f absorb the temporal discrepancy: each pose places the agent at the correct world position for its own capture time, so the canonical parameters need only encode the agent's intrinsic shape. This is in direct contrast to the single-timeline case, where no canonical parameter can simultaneously satisfy $\delta_n^v = \mathbf{0}$ and $\delta_n^f = \mathbf{0}$ (Part (ii)).

To analyze the optimization dynamics, we employ the Empirical Neural Tangent Kernel (NTK) adapted to the Gaussian splatting renderer. Let $\theta \in \mathbb{R}^d$ be the full optimizable parameter vector; the empirical NTK is the $M \times M$ matrix with entries

$$\Theta(\mathbf{x}_k, \mathbf{x}_l) = \big\langle \nabla_\theta \hat{I}^{c_k}(u_k;\theta),\ \nabla_\theta \hat{I}^{c_l}(u_l;\theta)\big\rangle, \tag{37}$$

where $\mathbf{x}_k = (u_k, c_k)$ indexes pixel u_k from source c_k. Under gradient flow on the squared loss, the residual vector $\mathbf{r}(t) = \hat{\mathbf{I}}(t) - \mathbf{I}^*$ evolves as $\dot{\mathbf{r}}(t) \approx -\Theta(0)\,\mathbf{r}(t)$ in the linearized regime (Assumption 1), so the spectral properties of Θ govern the optimization dynamics. Partitioning the parameter vector as $\theta = (\bar{\boldsymbol{\mu}}, \boldsymbol{\xi})$, where $\bar{\boldsymbol{\mu}} \in \mathbb{R}^{3N}$ collects the canonical means and $\boldsymbol{\xi}$ collects the pose parameters, the NTK decomposes additively: $\Theta = \Theta^{\bar{\mu}} + \Theta^{\boldsymbol{\xi}}$.

In the single-timeline formulation, the shared pose $\xi \in \mathbb{R}^6$ couples both sources, giving the pose Jacobian $J_\xi^{\text{single}} = (\nabla_\xi \hat{\mathbf{I}}^v;\, \nabla_\xi \hat{\mathbf{I}}^f) \in \mathbb{R}^{M\times 6}$ and the pose kernel

$$\Theta_{\text{single}}^{\xi} = J_\xi^{\text{single}}\,(J_\xi^{\text{single}})^\top = \begin{pmatrix} \nabla_\xi \hat{\mathbf{I}}^v (\nabla_\xi \hat{\mathbf{I}}^v)^\top & \nabla_\xi \hat{\mathbf{I}}^v (\nabla_\xi \hat{\mathbf{I}}^f)^\top \\ \nabla_\xi \hat{\mathbf{I}}^f (\nabla_\xi \hat{\mathbf{I}}^v)^\top & \nabla_\xi \hat{\mathbf{I}}^f (\nabla_\xi \hat{\mathbf{I}}^f)^\top \end{pmatrix}. \tag{38}$$

The off-diagonal blocks $\nabla_\xi \hat{\mathbf{I}}^v (\nabla_\xi \hat{\mathbf{I}}^f)^\top$ are generically non-zero: a pose update driven by the vehicle loss simultaneously perturbs the infrastructure residuals, and vice versa. This is the NTK-level manifestation of the gradient conflict identified in Part (ii).

Under DUST-GSG, the poses are parameterized by separate $\xi^v \in \mathbb{R}^6$ and $\xi^f \in \mathbb{R}^6$. Since the vehicle renderer uses only $P^v(\xi^v)$ and the infrastructure renderer uses only $P^f(\xi^f)$, there is no functional dependence between them: $\partial \hat{I}^v(u)/\partial \xi^f = \mathbf{0}$ and $\partial \hat{I}^f(u)/\partial \xi^v = \mathbf{0}$ for all u. The pose Jacobian therefore has a block-diagonal structure:

$$J_\xi^{\text{DST}} = \begin{pmatrix} \nabla_{\xi^v} \hat{\mathbf{I}}^v & \mathbf{0} \\ \mathbf{0} & \nabla_{\xi^f} \hat{\mathbf{I}}^f \end{pmatrix} \in \mathbb{R}^{M\times 12}, \tag{39}$$

and the resulting pose kernel is

$$\begin{aligned}\Theta_{\text{DST}}^{\boldsymbol{\xi}} &= J_\xi^{\text{DST}}\,(J_\xi^{\text{DST}})^\top \\ &= \begin{pmatrix} \nabla_{\xi^v} \hat{\mathbf{I}}^v (\nabla_{\xi^v} \hat{\mathbf{I}}^v)^\top & \mathbf{0} \\ \mathbf{0} & \nabla_{\xi^f} \hat{\mathbf{I}}^f (\nabla_{\xi^f} \hat{\mathbf{I}}^f)^\top \end{pmatrix} \\ &= \operatorname{diag}\Big(\Theta_{vv}^{\xi^v},\ \Theta_{ff}^{\xi^f}\Big),\end{aligned} \tag{40}$$

where $\Theta_{vv}^{\xi^v} := \nabla_{\xi^v} \hat{\mathbf{I}}^v (\nabla_{\xi^v} \hat{\mathbf{I}}^v)^\top \in \mathbb{R}^{M_v \times M_v}$ and $\Theta_{ff}^{\xi^f} := \nabla_{\xi^f} \hat{\mathbf{I}}^f (\nabla_{\xi^f} \hat{\mathbf{I}}^f)^\top \in \mathbb{R}^{M_f \times M_f}$. The off-diagonal blocks vanish identically, establishing the block-diagonal structure asserted in Eq. (40).

Partitioning the full residual vector as $\mathbf{r}(t) = (\mathbf{r}^v(t)^\top, \mathbf{r}^f(t)^\top)^\top$, the gradient flow restricted to pose parameters evolves as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\mathbf{r}^v \\ \mathbf{r}^f\end{pmatrix} = -\Theta_{\text{DST}}^{\boldsymbol{\xi}}\begin{pmatrix}\mathbf{r}^v \\ \mathbf{r}^f\end{pmatrix} = -\begin{pmatrix}\Theta_{vv}^{\xi^v} & \mathbf{0} \\ \mathbf{0} & \Theta_{ff}^{\xi^f}\end{pmatrix}\begin{pmatrix}\mathbf{r}^v \\ \mathbf{r}^f\end{pmatrix}. \tag{41}$$

The block-diagonal structure decouples the two subsystems completely:

$$\dot{\mathbf{r}}^v(t) = -\Theta^{\xi^v}_{vv}\,\mathbf{r}^v(t), \qquad \dot{\mathbf{r}}^f(t) = -\Theta^{\xi^f}_{ff}\,\mathbf{r}^f(t). \tag{42}$$

These two ODEs are independent: the closed-form solution $\mathbf{r}^v(t) = e^{-\Theta^{\xi^v}_{vv}t}\mathbf{r}^v(0)$ depends only on ξ^v and $\mathbf{r}^v(0)$, with no dependence on ξ^f or $\mathbf{r}^f$, and vice versa. Consequently, a gradient step on ξ^v driven by $\mathbf{r}^v$ produces *exactly zero* change in $\mathbf{r}^f$, and vice versa—the precise sense in which DUST-GSG eliminates cross-source gradient interference at the pose level. Combined with the zero-residual result of Eq. (36), this establishes all three claims of Part (iii). □

C Implementation Details

C.1 Initialization

For the background model, we combine 8×10^5 LiDAR points with 2×10^5 random samples, divided into 1×10^5 near samples uniformly distributed by distance to the scene's origin and 1×10^5 far samples uniformly distributed by inverse distance. We filter out LiDAR samples belonging to dynamic objects. For rigid nodes and non-rigid deformable nodes, we utilize their tracking bounding boxes to accumulate LiDAR points. Each dynamic instance is initialized with at most 5 000 canonical Gaussians, and we only model moving instances, with trajectory length thresholds of 1.0 for rigid nodes and 0.5 for deformable nodes. To determine the initial color, LiDAR points are projected onto the image plane, whereas random samples are initialized with random colors.

Prior to training, we perform offline Pose Correction to initialize the dual-timeline poses. Co-visible static vehicles serve as geometric anchors, matched across the vehicle and infrastructure views Euclidean distance and the Hungarian method. A 6-DoF pose correction is estimated by minimizing corner alignment error using L-BFGS. The refined poses regenerate cooperative labels, and missing tracks are filled via linear interpo lation for translation and Slerp for rotation for gaps up to two frames. These labels initialize separate pose trajectories for the vehicle-side and infrastructure-side timelines, with all timestamps normalized to $[0, 1]$.

C.2 Training

Our method trains for 30 000 iterations with all scene nodes optimized jointly. The learning rate for background Gaussian properties aligns with the default settings of 3DGS: positions at 1.6×10^{-4} (d ecaying to 1.6×10^{-6}), SH DC at 2.5×10^{-3}, SH rest at 1.25×10^{-4}, opacity at 5×10^{-2}, scaling at 5×10^{-3}, and rotation at 1×10^{-3}. The degrees of spherical harmonics a reset to 3 for background nodes, rigid nodes, and deformable nodes. For deformable nodes, the canonical position learning rate is scaled by a factor of 2.0.

The learning rate for the rotation of instance poses is 1×10^{-5}, decreasing exponentially to 5×10^{-6}. The learning rate for the translation of instance poses is 5×10^{-4}, decreasing exponentially to 1×10^{-4}. For deformable nodes, the deformation network (8 layers, 256 channels) uses a learning rate of 1.6×10^{-3}, decaying to 1.6×10^{-4}, and the temporal embedding uses 1×10^{-3}, decaying to 1×10^{-4}. The sky environment map is optimized at 0.01. Affine transformation and camera pose refinement modules both use 1×10^{-5}.

For the Gaussian densification strategy, we utilize the absolute gradient of Gaussians. We set the densification threshold of the position gradient to 5×10^{-4} and the densification size threshold to 3×10^{-3}. We reset the opacity of Gaussians to 0.01 every 3 000 iterations, with an opacity warm-up of 500 steps and refinement every 100 steps. We stop splitting background Gaussians at 15 000 steps, rigid nodes at 30 000 steps, a nd deformable nodes at 20 000 steps.

The photometric loss combines L1 and SSIM with weights 0.8 and 0.2. Additional regularization includes mask (0.05), depth (0.01), affine (1×10^{-5}), and dynamic region loss (0.05, with 0.1 for vehicles and 0.05 for humans, activated from iteration 3 000). For pose regularization, temporal smoothness uses 0.01 over 5 frames; effective-translational temporal smoothness uses 0.002 over 1 frame; time-offset velocity loss is 0.01; filled residual loss is 0.02; and group residual loss is 0.01 (stopped after 12 000 steps). The drift loss weight linearly decays to zero during training.

The training process commences from a $1/4$ image resolution, doubling every 250 iterations. We supervise with two asynchronous cameras (vehicle-side cam0 and roadside cam1), using independently provided timestamp files, with the roadside view as the reference for pose-group alignment. The sky environment map resolution is 1024.

Our method runs on NVIDIA A100 Tensor Core GPU, with training for each scene taking about one hour. Training time varies with different training settings.

C.3 Optimization

Since our newly constructed regularization term for pose optimization has already been introduced in Sec 3.3, only the remaining regularization terms are detailed here. Specifically, following the implementation of OMNIRE [3], they are defined as follows:

We utilize the loss function introduced in Eq 43 to jointly optimize all learnable parameters. The image loss is computed as:

$$\mathcal{L}_{\text{image}} = (1 - \lambda_r)\mathcal{L}_1 + \lambda_r \mathcal{L}_{\text{SSIM}} \tag{43}$$

due to sparse temporal-spatial observation of the dynamic part, its supervision signal is insufficient. To address this, we apply a higher image loss weight to the dynamic regions identified by the rendered dynamic mask. This weight is set to 5. The depth map loss is computed as:

$$\mathcal{L}_{\text{depth}} = \frac{1}{hw} \sum \left\| \mathcal{D}^s - \hat{\mathcal{D}} \right\|_1 \tag{44}$$

where $\mathcal{D}^s$ is the inverse of the sparse depth map. We project LiDAR points onto the image plane to generate the sparse LiDAR map, and $\hat{\mathcal{D}}$ is the inverse of the predicted depth map.

The mask loss $\mathcal{L}_{\text{opacity}}$ is computed as:

$$\mathcal{L}_{\text{opacity}} = -\frac{1}{hw} \sum O_g \cdot \log O_g - \frac{1}{hw} \sum M_{\text{sky}} \cdot \log(1 - O_g) \tag{45}$$

where M_{sky} is the sky mask, and O_g is the rendered opacity map.

In addition to the reconstruction losses, we introduce various regularization terms for different Gaussian representations to improve quality. Among these, an important regularization term is $\mathcal{L}_{\text{pose}}$, designed to ensure smooth human body poses $\boldsymbol{\theta}(t)$. This term is defined as:

$$\mathcal{L}_{\text{reg}} = \frac{1}{2} \left\| \boldsymbol{\theta}(t - \delta) + \boldsymbol{\theta}(t + \delta) - 2\boldsymbol{\theta}(t) \right\|_1 \tag{46}$$

where δ is a randomly chosen integer from $\{1, 2, 3, 4, 5\}$. We set the weight of the SSIM loss, λ_r, to 0.2, the depth loss, λ_{depth}, to 0.01, the opacity loss, λ_{opacity}, to 0.05, and the pose smoothness loss, λ_{reg}, to 0.01.

D Evaluation

D.1 Datasets

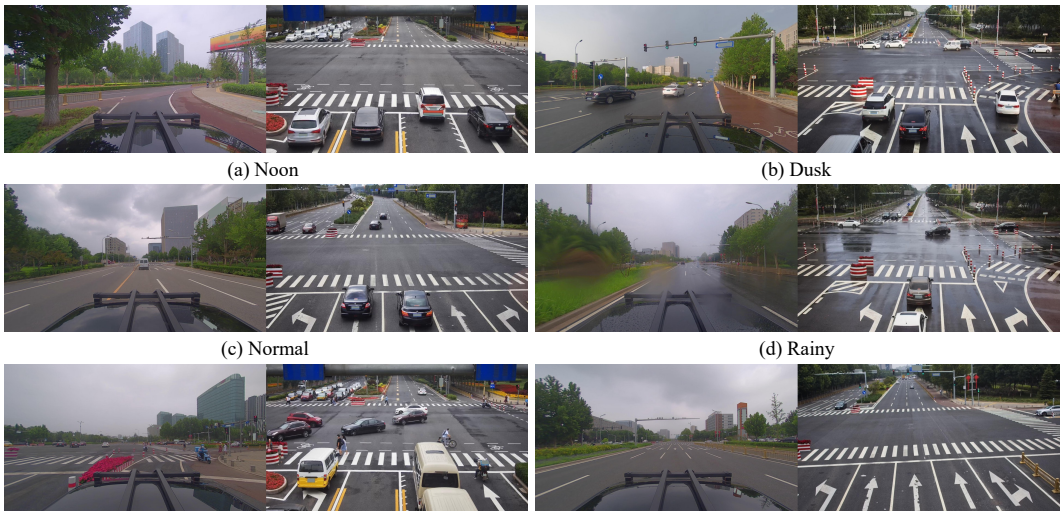
We examine all 95 publicly available V2X-Seq sequences and retain 67 with sufficiently complete and continuous LiDAR data. Sequences with clearly missing or discontinuous LiDAR observations are excluded. From the retained sequences, we select 26 according to capture period, weather, traffic density, and dynamic object distribution, independently of model performance. Weather is checked manually, while capture period and traffic density are supported by dataset metadata. The condition subsets overlap, and the Full results use the 26 unique selected sequences. Specifically, for the "times of data" category, night-time V2I scenarios are not provided in the V2X-Seq [34] dataset. Thus, the period from 11:30 to 12:30 is defined as "Noon". The period from 16:30 to 17:30 is defined as "Dusk". Four sequences are selected for each time period. For the "Weather" category, only normal and rainy conditions are available in the dataset. Therefore, two subcategories are defined: "Normal" and "Rainy". Six sequences are allocated to each subcategory. For the "traffic densities" category, referred to as "Crowd", the number of agents in each sequence was counted. A higher number of agents indicates higher traffic densities. The 6 sequences with the most agents are selected as "High". The 6 sequences with the fewest agents are selected as "Low". The selected sequence id, average frames, average agent counts, and data formats for each category are detailed in Tables 4. Examples for each category are illustrated in Fig 6.

Reconstruction and NVS protocols. For scene reconstruction, all frames are used for training and evaluation. For NVS, both the vehicle and infrastructure images at every 10th cooperative timestamp are excluded from training and used for testing. The two settings are reported separately, and neither evaluates generalization to new scenes. Our use of V2X-Seq for cooperative reconstruction follows prior work such as CRUISE. All methods in our evaluation are trained from scratch with the common data splits, regenerated annotations, and 30,000 training iterations stated in Section 4.1.

Original camera timestamp offsets. For each paired vehicle and infrastructure image, we compute the absolute difference between their recorded camera timestamps. We first take the median within each sequence and then summarize the resulting 26 values in Table 5. This gives each sequence equal weight. These statistics describe the original dataset offsets; the delays in Figure 4 are additional temporal perturbations.

Table 4: Detailed descriptions of the sequences selected for classification are provided. The specific condition of the sequences is denoted by Cond. The average number of image frames across all sequences under each condition is represented by Frames. The average number of agents in all sequences under each condition is indicated by Agent.

Cond.	Sequence ID	Frames	Agent	Data Format
Noon	70,71,72,73	181	27	Image, LiDAR, Bounding Box
Dusk	55,56,57,58	204	22	Image, LiDAR, Bounding Box
Normal	21,23,29,30,60,63	176	17	Image, LiDAR, Bounding Box
Rainy	36,40,41,42,47,48	166	10	Image, LiDAR, Bounding Box
Low	7,21,23,36,42,79	156	5	Image, LiDAR, Bounding Box
High	2,4,59,66,71,73	240	48	Image, LiDAR, Bounding Box



(a) Noon (b) Dusk

(c) Normal (d) Rainy

(e) High (f) Low

Figure 6: The images from both the vehicle and infrastructure perspectives of the dataset are presented under each condition.

D.2 Metrics

We assess reconstruction quality using five metrics: PSNR [21], SSIM, LPIPS [36], FVD [20], and RAFT-EPE [18]. For the image-based metrics (PSNR, SSIM, and LPIPS), we report results on the full image as well as within dynamic regions. For temporal consistency, FVD and RAFT-EPE are computed exclusively within dynamic masks to avoid static back grounds dominating the motion-related evaluation.

PSNR. The Peak Signal-to-Noise Ratio measures pixel-wise reconstruction fidelity between the rendered image $\hat{\mathbf{I}}$ and the ground-truth image $\mathbf{I}$. It is defined as:

$$\text{PSNR} = -10 \log_{10} \left(\frac{1}{N} \sum_{i=1}^{N} (\hat{I}_i - I_i)^2 \right), \tag{47}$$

where N is the number of valid pixels and pixel intensities are normalized to $[0, 1]$. A higher PSNR indicates lower reconstruction error.

SSIM. The Structural Similarity Index Measure evaluates local structural consistency by comparing luminance, contrast, and structure within a sliding Gaussian window. For two image patches $\mathbf{x}$ and $\mathbf{y}$, it is formulated as:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{48}$$

where μ_x, μ_y and σ_x, σ_y are the local means and standard deviations, σ_{xy} is the cross-covariance, and c_1, c_2 are small constants for numerical stability. We compute the full SSIM map using an 11×11 Gaussian window and average the values over the valid pixel region.

Table 5: Distribution of the per-sequence median absolute camera timestamp offsets in the 26 evaluated sequences.

Statistic over the 26 sequence medians	Offset (ms)
Minimum	68.36
25th percentile	89.09
Median	97.10
75th percentile	102.25
Maximum	138.36
Mean $\pm$ standard deviation	96.39 ± 17.39

LPIPS. The Learned Perceptual Image Patch Similarity measures perceptual distance using deep network features. Given rendered and ground-truth images, we extract multi-scale features from a pre-trained AlexNet and c ompute a normalized ℓ_2 distance:

$$\text{LPIPS}(\hat{\mathbf{I}}, \mathbf{I}) = \sum_{\ell} \frac{1}{H_\ell W_\ell} \sum_{h,w} \left\| \mathbf{w}_\ell \odot \left(\phi_\ell(\hat{\mathbf{I}})_{h,w} - \phi_\ell(\mathbf{I})_{h,w} \right) \right\|_2^2, \tag{49}$$

where $\phi_\ell(\cdot)$ denotes the feature map from layer ℓ of the network, $\mathbf{w}_\ell$ are learned channel-wise weights, and H_ℓ, W_ℓ are the spatial dimensions of the feature map. The input images are normalized from $[0, 1]$ to $[-1, 1]$ before feature extraction. Lower LPIPS indicates better perceptual similarity.

FVD. The Fréchet Video Distance measures temporal consistency by computing the Fréchet distance between the feature distributions of rendered and ground-truth videos. We extract spatio-temporal features using a pre-trained I3D network (trained on Kinetics) that yields 400-dimensional embeddings. For a video with T frames, we build non-overlapping clips of length $T_c = 16$ frames. Each clip is pre-processed by resizing the shorter side to 224 p ixels, center-cropping to 224×224, and normalizing to $[-1, 1]$. Let $(\boldsymbol{\mu}_{\text{gen}}, \boldsymbol{\Sigma}_{\text{gen}})$ and $(\boldsymbol{\mu}_{\text{real}}, \boldsymbol{\Sigma}_{\text{real}})$ denote the mea n and covariance of the generated and real clip features, respectively. The FVD is defined as:

$$\text{FVD} = \|\boldsymbol{\mu}_{\text{gen}} - \boldsymbol{\mu}_{\text{real}}\|_2^2 + \text{Tr}\Big(\boldsymbol{\Sigma}_{\text{gen}} + \boldsymbol{\Sigma}_{\text{real}} - 2\sqrt{\boldsymbol{\Sigma}_{\text{gen}}\boldsymbol{\Sigma}_{\text{real}}}\Big), \tag{50}$$

where $\text{Tr}(\cdot)$ is the matrix trace and $\sqrt{\cdot}$ denotes the matrix square root. Following standard practice, FVD is computed only within dynamic regions so that temporal consistency of moving agents is emphasi zed.

RAFT-EPE. To evaluate motion plausibility, we compute the End-Point Error (EPE) between optical flows estimated from consecutive rendered frames and the corresponding ground-truth frames. We use RAFT with the pre-tr ained weights on FlyingThings (raft-things.pth) and run inference with 20 update iterations. For consecutive frames $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$, RAFT predicts a dense flow field $\mathbf{F}_t \in \mathbb{R}^{H \times W \times 2}$ with horizontal and vertical components (u, v). The EPE is computed as:

$$\text{EPE} = \frac{1}{|\mathcal{M}|} \sum_{(h,w) \in \mathcal{M}} \sqrt{\left(\hat{u}_{h,w} - u_{h,w}\right)^2 + \left(\hat{v}_{h,w} - v_{h,w}\right)^2}, \tag{51}$$

where $\mathcal{M}$ denotes the union of dynamic masks from both frames, ensuring evaluation focuses strictly on moving regions.

D.3 Baselines

To evaluate the effectiveness of our approach, our method is compared with several state-of-the-art baselines.

- **3DGS [7]**: 3D Gaussian Splatting (3DGS) is utilized for real-time and high-quality novel-view synthesis. Continuous volumetric radiance fields are represented by 3D Gaussians. These Gaussians are initialized from sparse calibration points. Interleaved optimization and density control are performed to accurately represent the scene geometry. A fast visibility-aware rendering algorithm is developed to support anisotropic splatting and real-time rendering.
- **PVG [2]**: Periodic Vibration Gaussian (PVG) is introduced to model dynamic urban scenes. A unified representation model is presented to capture the synergistic interactions of static and dynamic elements. Periodic vibration-based temporal dynamics are incorporated into the 3D Gaussian splatting technique. To handle sparse training data, a temporal smoothing mechanism and a position-aware adaptive control strategy are proposed. High-quality reconstruction is achieved without manually labeled bounding boxes or optical flow estimation.
- **OMNIRE [3]**: OMNIRE is presented as a comprehensive system for the creation of high-fidelity digital twins from on-device logs. Diverse dynamic objects, including vehicles and pedestrians, are fully reconstructed. Scene graphs are built on 3DGS. Multiple Gaussian representations are constructed in canonical spaces to

model various dynamic actors. Advanced simulations with human-participated scenarios are supported by this holistic reconstruction capability.

- **CRUISE [25]**: CRUISE is proposed as a comprehensive reconstruction and synthesis framework for Vehicle-to-everything (V2X) driving environments. Real-world scenes are accurately reconstructed using decomposed Gaussian Splatting. Dynamic traffic participants are decomposed into editable Gaussian representations. Images from both ego-vehicle and infrastructure views are rendered to facilitate large-scale V2X dataset augmentation.
- **StreetGS [29]**: Street Gaussians (StreetGS) is introduced to model dynamic urban streets. The dynamic urban scene is explicitly represented as a set of point clouds equipped with semantic logits and 3D Gaussians. Foreground vehicles and the background are modeled separately. Tracked poses and a 4D spherical harmonics model are optimized to capture the dynamics and appearance of object vehicles. Fast scene editing and rendering are achieved by this explicit representation.
- **DeformableGS [32]**: Deformable 3D Gaussians Splatting (DeformableGS) is proposed to model monocular dynamic scenes. 3D Gaussians are learned in a canonical space. A deformation field is applied to capture the intricate details of moving objects. An annealing smoothing training mechanism is introduced to mitigate the impact of inaccurate poses. Real-time rendering and high visual quality are achieved through a differential Gaussian rasterizer.

D.4 Additional Asynchrony-Aware Baselines

Both comparisons use the same four randomly selected sequences from the High Crowd subset as the ablation study in Table 2. All methods use the regenerated cooperative annotations and the same training and evaluation settings within each comparison.

StreetGS with source-specific pose interpolation. This baseline interpolates dynamic-object poses using the capture timestamps of the vehicle and infrastructure cameras. It does not use DUST's dual-timeline decoupling or source-specific joint pose optimization. In contrast, the w/o Dual Timelines ablation still interpolates poses at the source timestamps and retains the subsequent pose optimization in DUST, so it does not isolate interpolation alone.

Table 6: Comparison DUST with StreetGS using source-specific pose interpolation.

Variant	Full Image			Dynamic Area	
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
DUST	**28.75**	**0.874**	**0.125**	**25.12**	**0.837**
StreetGS variant	27.73	0.872	0.130	21.76	0.731

Unified continuous trajectory. This baseline uses one source-independent pose trajectory for each dynamic object. Vehicle and infrastructure observations query this trajectory at their respective timestamps. Translation uses linear interpolation and rotation uses quaternion Slerp. The canonical Gaussians, losses, training iterations, and evaluation settings are otherwise the same as in DUST.

Table 7: Comparison with a unified continuous trajectory on the same four sequences used in Table 2. Both methods use the same canonical Gaussians, losses, training iterations, and evaluation settings.

	Full Image			Dynamic Area	
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
Unified continuous trajectory	25.93	0.856	0.149	21.11	0.718
DUST	**28.67**	**0.873**	**0.126**	**24.89**	**0.823**

Table 7 shows that DUST improves full-image PSNR by 2.74 dB and dynamic-area PSNR by 3.78 dB. Dynamic-area SSIM increases from 0.718 to 0.823. These results support the effectiveness of source-specific pose trajectories under the evaluated settings.

D.5 Pose-Correction Robustness

Sparse static anchors. A vehicle is treated as a static anchor when its cumulative displacement in the global coordinate frame over the sequence is below 2.5 m. We examine sequences 0071 and 0007; sequence 0007 contains three static agents out of five agents. Table 8 shows that Pose Correction improves dynamic-area PSNR by 0.84 dB on sequence 0071 and 0.78 dB on sequence 0007.

Table 8: Pose correction results in scenes with a limited number of static agents.

	Full Image			Dynamic Area	
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
Before Pose Correction (0071)	28.84	0.892	0.108	23.67	0.813
DUST (0071)	29.43	0.894	0.106	24.51	0.838
Before Pose Correction (0007)	31.02	0.898	0.089	25.89	0.873
DUST (0007)	31.36	0.898	0.088	26.67	0.885

Relaxed motion thresholds. When strictly static vehicles are scarce, slowly moving vehicles can be included by relaxing the displacement threshold. On sequence 0066, we compare thresholds of $< 1\,\mathrm{m}$ and $< 3.5\,\mathrm{m}$. As shown in Table 9, the stricter threshold produces better results, while the relaxed threshold still improves dynamic-area PSNR from 24.56 to $25.26\,\mathrm{dB}$. The case without reliable anchors has not been evaluated.

Table 9: Pose correction on sequence 0066 using different static-agent thresholds.

	Full Image			Dynamic Area	
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
Before Pose Correction	29.61	0.890	0.106	24.56	0.820
DUST ($< 3.5\,\mathrm{m}$)	29.64	0.890	0.106	25.26	0.840
DUST ($< 1\,\mathrm{m}$)	29.70	0.891	0.104	25.35	0.845

Perturbation settings. We conduct the following perturbation experiments on sequence 0071, uniformly sampling 100 of its 204 frames. Each setting runs the full 6-DoF Pose Correction, and results are reported as the mean and standard deviation over three random seeds. We measure the 3D IoU of correctly matched pairs before and after optimization.

Table 10: Sensitivity to calibration noise on sequence 0071.

Calibration noise (σ_t, σ_r)	M1: Pre-opt	M2: Post-opt	ΔIoU
$(0.0,\ 0.0)$	0.3295 ± 0.0000	0.5529 ± 0.0000	$+0.2234$
$(0.1,\ 1.0)$	0.1589 ± 0.0245	0.5470 ± 0.0052	$+0.3882$
$(0.3,\ 3.0)$	0.0488 ± 0.0063	0.4752 ± 0.0359	$+0.4264$
$(0.5,\ 5.0)$	0.0155 ± 0.0058	0.3669 ± 0.0031	$+0.3514$
$(1.0,\ 10.0)$	0.0014 ± 0.0009	0.1817 ± 0.0174	$+0.1803$

As shown in Table 10, the pre-optimization alignment is sensitive to calibration noise, and M1 drops markedly as the initial extrinsic error grows. Pose Correction improves 3D IoU under all tested conditions. Under $(0.1\,\mathrm{m}, 1^\circ)$ and $(0.3\,\mathrm{m}, 3^\circ)$ perturbations, IoU rises by 0.39 and 0.43. Even under the extreme $(1\,\mathrm{m}, 10^\circ)$ setting, the gain is 0.18. The post-optimization IoU, however, is still below the result without additional perturbation, which means the module can partially correct large initial extrinsic errors but cannot always recover accurate calibration.

Table 11 reports the sensitivity of Pose Correction to perturbations of bounding-box centers and yaw angles. Pose Correction improves mean 3D IoU at every tested noise level. However, the post-optimization IoU decreases from 0.5529 without additional noise to 0.1949 at $(1\,\mathrm{m}, 10^\circ)$. These results indicate partial tolerance to the tested annotation perturbations, while substantial noise still degrades the final alignment.

With one to four incorrect anchor matches per frame, the IoU gain on correctly matched pairs remains approximately $+0.20$. The post-optimization IoU decreases from 0.5529 without replacements to 0.5133 with four replacements. These results show tolerance to the tested number of incorrect correspondences when other reliable matches remain available.

E Scope and Limitations

More than two sources. Additional sensors would introduce pose trajectories for the dynamic agents they observe, while sharing the static scene and canonical Gaussians. The number of pose parameters therefore increases with the number of observing sources. Our experiments use one vehicle camera and one infrastructure camera; efficiency and optimization stability with more sources remain unverified.

Table 11: Sensitivity of Pose Correction to 3D box annotation noise.

3D box noise (σ_c, σ_y)	M1: Pre-opt	M2: Post-opt	ΔIoU
$(0.0,\ 0.0)$	0.3295 ± 0.0000	0.5529 ± 0.0000	$+0.2234$
$(0.2,\ 2.0)$	0.3038 ± 0.0044	0.4773 ± 0.0048	$+0.1735$
$(0.5,\ 5.0)$	0.1960 ± 0.0105	0.3423 ± 0.0030	$+0.1463$
$(1.0,\ 10.0)$	0.0835 ± 0.0054	0.1949 ± 0.0264	$+0.1114$

Table 12: Sensitivity of Pose Correction to incorrect anchor matches.

n_{swaps}	M1: Pre-opt	M2: Post-opt	ΔIoU
0	0.3295 ± 0.0000	0.5529 ± 0.0000	$+0.2234$
1	0.3272 ± 0.0025	0.5287 ± 0.0067	$+0.2015$
2	0.3200 ± 0.0053	0.5227 ± 0.0199	$+0.2027$
3	0.3189 ± 0.0013	0.5164 ± 0.0021	$+0.1975$
4	0.3131 ± 0.0029	0.5133 ± 0.0050	$+0.2002$

Queries at arbitrary times. DUST uses linear interpolation for translation and Slerp for rotation within each source's time range. Rendering requires specifying the source as well as the timestamp and query camera. The source-specific trajectories do not guarantee a unique dynamic world state determined by time alone.

Appearance differences. Spherical harmonics and the existing affine appearance transformation account for some changes in viewing direction, brightness, and color. Strong differences in exposure, white balance, camera response, reflections, or motion blur can still produce conflicting supervision for the shared canonical appearance and reduce texture quality. CPSR-CLIP [14] studies conditional style reconstruction for zero-shot domain adaptation, while DiDE [23] separately controls color and texture in 3D stylization. Whether these approaches can improve appearance consistency across vehicle and infrastructure cameras remains to be evaluated.

Experimental scope. The evaluation is limited to the selected V2X-Seq sequences. Generalization to other cooperative datasets and the case without reliable anchors have not been empirically evaluated. The additional delays in Figure 4 test robustness to temporal asynchrony; arbitrary timestamp errors, clock drift, and continuous frame drops remain unexamined.

F Broader Impacts

A practically significant yet under-explored problem is addressed by this work. The reconstruction of dynamic scenes from Vehicle-to-Infrastructure Cooperative Autonomous Driving data is fundamentally complicated by temporal asynchrony. Such asynchronous observations are routinely generated in real-world cooperative systems. Vehicle and infrastructure sensors are driven by independent hardware clocks. The same moving agents are captured at different physical times. A genuine operational need is therefore created for asynchronous-aware reconstruction. A principled foundation is provided for deploying cooperative reconstruction models in un-synchronized settings. Cross-source interference is entirely eliminated by the Decoupled Spatio-Temporal Gaussian Scene Graph. This decoupled dual-timeline design is broadly applicable beyond autonomous driving. Any multi-sensor spatio-temporal domain subject to hardware asynchrony is supported by this architecture. Multi-camera sports broadcasting, drone swarm perception, and large-scale urban surveillance are included. Reusable components are further offered by the static anchor-based label regeneration and the pose-regularized joint optimization schem. Cooperative models can be adapted to new un-synchronized deployments by downstream practitioners. The community is anticipated to be encouraged by the formalization of this asynchronous problem. Idealized synchronized evaluation protocols can be surpassed. Highly realistic simulation environments can be developed to serve the multi-source operational lifetimes of real cooperative systems.

G Safeguards

Several safeguards are implemented for the responsible release of the models and data. Personally identifiable information is completely removed from the utilized dataset. Specific sensitive elements like pedestrian faces and vehicle license plates are fully anonymized. A Croissant file is included in the released directory. This file describes the original data and dataset resources securely. The regenerated cooperative annotations and model weights are restricted to academic research purposes. Commercial use is strictly prohibited by the data

agreement. The synthesized driving scenes are intended solely for simulation and algorithmic evaluation. Any malicious application is explicitly forbidden by the release license.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: In the abstract and introduction sections, we delineate the problems addressed by this work. and outline our contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the conclusion section, we highlight the limitations of the current work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: We provide the complete theoretical proofs in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide detailed experiment settings in Appendix C. Besides, code is anonymously available at https://github.com/mura1n/DUST.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: Code is anonymously available at https://github.com/mura1n/DUST.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide experiment settings and hyperparameters in Appendix C.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A]

   Justification: Not applicable.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources in Appendix C

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics, and we confirm our research in the paper conforms with Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Please refer to the Appendix F.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [Yes]

    Justification: Please refer to the Appendix G.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We provide source links for the baselines in Appendix C. The dataset introduction can be found in Appendix D.1.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We release our code anonymously at `https://github.com/mura1n/DUST`.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLM is used only for formatting purposes and does not impact the core methodology.

Guidelines:

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