Flux4D: Flow-based Unsupervised 4D Reconstruction

Jingkang Wang[1,2*]   **Henry Che**[1,3*†]   **Yun Chen**[1,2*]   **Ze Yang**[1,2]
Lily Goli[1,2†]   **Sivabalan Manivasagam**[1,2]   **Raquel Urtasun**[1,2]

Waabi[1]   University of Toronto[2]   UIUC[3]

https://waabi.ai/flux4d

Abstract

Reconstructing large-scale dynamic scenes from visual observations is a fundamental challenge in computer vision. While recent differentiable rendering methods such as NeRF and 3DGS have achieved impressive photorealistic reconstruction, they suffer from scalability limitations and require annotations to decouple moving actors from the static scene, such as in autonomous driving scenarios. Existing self-supervised methods attempt to eliminate explicit annotations by leveraging motion cues and geometric priors, yet they remain constrained by per-scene optimization and sensitivity to hyperparameter tuning. In this paper, we introduce *Flux4D*, a simple and scalable framework for 4D reconstruction of large-scale dynamic driving scenes. *Flux4D* directly predicts 3D Gaussians and their motion dynamics to reconstruct sensor observations in a fully unsupervised manner. By adopting only photometric losses and enforcing an "as static as possible" regularization, *Flux4D* learns to decompose dynamic elements directly from raw data without requiring pre-trained supervised models or foundational priors simply by training across many scenes. Our approach enables efficient reconstruction of dynamic scenes within seconds, scales effectively to large datasets, and generalizes well to unseen environments, including rare and unknown objects. Experiments on outdoor driving datasets show *Flux4D* significantly outperforms existing methods in scalability, generalization, and reconstruction quality.

1   Introduction

Reconstructing the 4D physical world from visual observations captured in the wild is a key goal in computer vision, with applications in virtual reality and robotics, including autonomous driving. High-quality reconstructions provide the foundation for scalable simulation environments that enable safer and more efficient autonomy development. Unlike artist-created environments, environments built automatically with data collected by sensor-equipped vehicles are more realistic, are more cost-efficient, and capture the diversity of the real world [50, 62, 31].

Advances in differentiable rendering approaches such as Neural Radiance Field (NeRF) [32] and 3D Gaussian Splatting (3DGS) [18] have enabled high-quality reconstruction of dynamic scenes [62, 59, 72, 45, 19, 6, 46]. These methods decompose scenes into a static background and a set of dynamic actors using human annotations such as 3D tracklets or dynamic masks, and then perform rendering on the composed representation, optimizing to reconstruct the input observations. While they achieve impressive visual fidelity, their reliance on manual annotations to decompose static and dynamic elements increases costs and time, preventing these methods from scaling to large sets of unlabelled data. Some approaches leverage pre-trained perception models to generate annotations automatically, but this can cause artifacts when the model predictions are noisy or incorrect, which can be difficult to recover from during reconstruction. Moreover, these methods typically require hours to reconstruct

---

*Equal contributions.

†Work done while a research intern at Waabi.

each scene on consumer GPUs. These two main issues, expensive annotation costs and slow per-scene optimization, limit the scalability of these methods.

Recent works have explored self-supervised approaches to eliminate the reliance on human annotations and learn the decomposition of static and dynamic actors directly from data. This is a challenging task due to the ambiguity of actor motion over time, coupled with spatial geometry and appearance variations. One strategy attempts to improve the decomposition by incorporating additional regularization terms such as geometric constraints [37] or cycle consistency [61], or performing multi-stage training [17]. Another strategy is to leverage foundation models for additional semantic features or priors [37, 8, 61]. However, the resulting complex models can be sensitive to hyperparameters, slow to train, and unable to generalize to new scenes. Moreover, they often have poor decomposition results, and struggle to render novel views, limiting their usability.

As an alternative to costly per-scene optimization, generalizable approaches [3, 51, 2, 5, 14, 53, 69] use feed-forward neural networks to predict scene representations directly from observations, enabling efficient reconstruction within seconds. However, these approaches are designed for small-scale environments, can only process a few low-resolution images (typically 1-4 views with resolutions below 512px), and primarily focus on static scenes [2, 5] or only dynamic objects [40]. When handling large scenes with many dynamic elements, they rely on costly annotations [7, 41], limiting their scalability. Most recently, DrivingRecon [30] and STORM [60] propose feed-forward, self-supervised approaches for driving scenes. While promising, these methods focus on the sparse reconstruction setting and can only handle a small number (≤ 12) of low-resolution (≤ 360px) input views before reaching compute limits, and still depend on pre-trained vision models for semantic guidance, constraining their fidelity, scalability and applicability to downstream simulation.

In this paper, we propose *Flux4D*, an *unsupervised* and *generalizable* reconstruction approach that enables accurate and efficient 4D driving scene reconstruction at scale. Without any annotations, *Flux4D* predicts 3D Gaussians along with motion parameters directly in 3D space from multi-sensor observations within seconds, enabling efficient scene reconstruction. Our reconstruction paradigm is illustrated in Fig. 1. *Flux4D* uses a remarkably minimalist design that employs only photometric losses and a simple static-preference prior, without requiring complex regularization schemes or external supervision to learn the motion that prior works leverage. We find that the key ingredient for *Flux4D* to accurately recover geometry, appearance, and motion flow comes from learning across a diverse range of scenes. Moreover, *Flux4D*'s use of LiDAR data, commonly available in the autonomous driving domain, enable handling of a large number (≥ 60) of high-resolution (1080px) input multi-view images, achieving high-fidelity reconstruction and scalable simulation. Our 3D design yields a compact and geometrically consistent representation across views, improving efficiency, enabling explicit multi-view flow reasoning and reducing appearance-motion ambiguity.

Experiments on outdoor driving datasets PandaSet [57] and WOD [42] demonstrate that *Flux4D* achieves better scene decomposition and novel view synthesis than previous state-of-the-art annotation-free reconstruction methods, and is competitive with per-scene optimization methods that use human annotations. We also show that *Flux4D* can be trained to predict sensor observations in future frames, akin to next-token prediction, but applied to dynamic 3D scenes. Finally, we showcase using *Flux4D*'s reconstruction for controllable camera simulation via scene editing and novel view rendering at high resolution (≥ 1080px). *Flux4D* highlights the power of unsupervised learning for 4D scene reconstruction, enabling efficient scaling to vast unlabeled datasets.

2 Related Work

Optimization-based 4D reconstruction: Inspired by differentiable rendering [32, 18], recent approaches use deformation fields [38, 36, 66, 55] to model dynamic scenes but still struggle with real-world complexity due to overparameterization and poor static-dynamic decomposition. While some methods address this by using human annotations (3D tracklets, semantic models) to explicitly separate static [49, 63] and dynamic elements [35, 62, 29, 47, 39, 59, 10, 13, 64], they remain limited by annotation quality and availability. Self-supervised alternatives using motion cues and physics-informed priors [56, 61, 8, 17, 37] reduce dependence on annotations but typically require complex regularization schemes and expensive per-scene optimization. In contrast, our approach reconstructs dynamic 4D scenes without explicit supervision or per-scene optimization, achieving scalable reconstruction through simple photometric losses with minimal regularization.

Generalizable reconstruction: Generalizable methods infer scene representations directly from observations without per-scene optimization [3, 51, 2, 5, 14, 53, 69], leveraging large training datasets
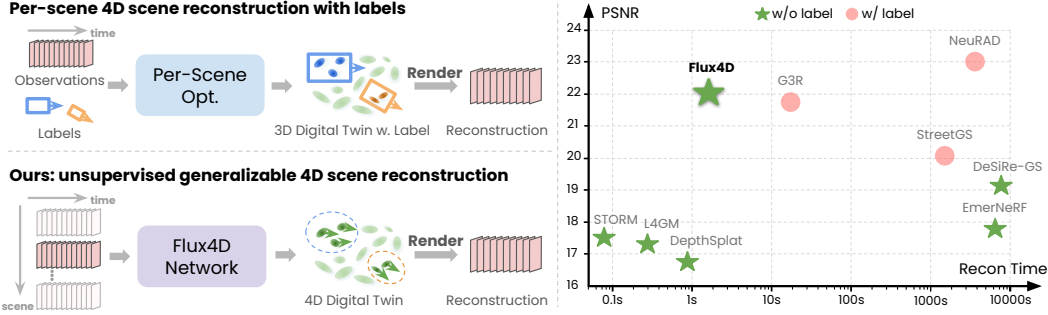
Figure 1: *Flux4D* **is a simple and scalable framework for unsupervised 4D reconstruction. Left:** Paradigms for 4D reconstruction. **Right:** realism-speed comparisons with existing works.

to improve reconstruction quality in novel environments. However, existing approaches primarily target static scenes, struggling with dynamic environments due to computational constraints and dependence on sparse, low-resolution inputs. Recent advances attempt to overcome these limitations using efficient architectures [73] or iterative refinement [7], but still rely on 3D annotations. In contrast, *Flux4D* generalizes to unseen dynamic scenes by predicting 3D Gaussians with their motion directly from raw observations without external supervision.

Unsupervised world models: Our work relates to recent advances in unsupervised world models, which learn predictive representations of environments without explicit supervision. These approaches typically tokenize visual data into discrete or continuous representations [15, 12, 52, 71, 33] processed by autoregressive or diffusion-based models to predict future states. While demonstrating impressive visual quality, such methods generally lack interpretable 3D structure, limiting precise control over generated content. Existing solutions often produce lower-resolution outputs with reduced temporal consistency, are typically restricted to single modalities (*e.g.*, camera [15, 12, 26] or LiDAR [21, 70, 65, 1]), and require substantial computational resources. While our primary focus is reconstruction, *Flux4D*'s ability to simultaneously model motion dynamics and predict future frames shares conceptual similarities with world models. Unlike these approaches, *Flux4D* uses explicit 3D representation, providing 3D interpretability, controllability and spatiotemporal consistency.

Unsupervised generalizable reconstruction: Most recently, DrivingRecon [30] and STORM [60] explore unsupervised generalizable 4D reconstruction for driving scenes, using feed-forward networks to predict the velocities of 3D Gaussians. Despite impressive performance, they can process only sparse (3-4), low-resolution ($\leq 256 \times 512$) frames with substantial computational requirements and rely on pre-trained vision models (DeepLabv3+ [4], SAM [23], ViT-Adapter [9]) for additional supervision, limiting their scalability and applicability. *Flux4D* achieves better performance with a simpler and more scalable approach, and through our novel incorporation of LiDAR to initialize the scene, can handle full HD images with denser views (> 60) while being computationally efficient. Please see supp. for more discussions.

3 Scalable 4D Reconstruction with *Flux4D*

Given a sequence of camera and LiDAR data captured by a robot sensor platform, we aim to reconstruct the underlying 4D scene representation that disentangles static and dynamic entities and supports high-quality rendering at novel viewpoints. Such a representation can enable future prediction and counterfactual simulation. To achieve scalable 4D scene reconstruction, our method should be unsupervised, meaning it uses no annotations, and fast, running in seconds. Towards this goal, we propose *Flux4D*, an unsupervised and generalizable approach that learns to reconstruct 4D scenes via three simple steps (Fig. 2). We first lift the sensor observations at each timestep to a set of initial 3D Gaussians. We then feed the initial representation to a network to predict 3D flow and refined attributes for each 3D Gaussian. Finally, we supervise the network solely through reconstruction and static-preference losses.

3.1 Scene Representation

Our approach takes a set of posed camera images $\mathcal{I} = \{\mathbf{I}_k\}_{1 \leq k \leq K}$ and LiDAR point clouds $\mathcal{P} = \{\mathbf{P}_k\}_{1 \leq k \leq K}$ captured over time by a moving platform and outputs a scene representation
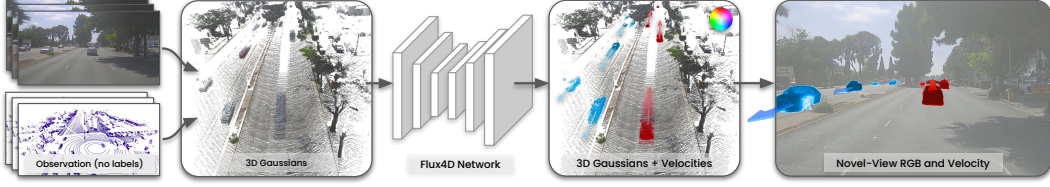
Figure 2: **Model overview.** *Flux4D* reconstructs 4D world by predicting 3D Gaussians with velocities given unlabelled sensor observations, and trained with the photometric reconstruction objective. The resultant model can be used for RGB and flow synthesis from novel views.

with geometry, appearance, and 3D flow. We represent the scene using a set of 3D Gaussians $\mathcal{G} = \{\mathbf{g}_i\}_{1 \leq i \leq M}$. Each Gaussian point g_i is parameterized by its center position $\mathbf{p}_i$ ($\mathbb{R}^3$), scale ($\mathbb{R}^3$), orientation ($\mathbb{R}^4$), color ($\mathbb{R}^3$) and opacity ($\mathbb{R}^1$) [18]. Additionally, we augment each Gaussian with a learnable instantaneous velocity $\mathbf{v}_i \in \mathbb{R}^3$ and a fixed capture time t_i. We denote the sets of velocities and timestamps for all Gaussians as $\mathcal{V} = \{\mathbf{v}_i\}_{1 \leq i \leq M}$ and $\mathcal{T} = \{t_i\}_{1 \leq i \leq M}$.

Initialization: We initialize Gaussian positions from LiDAR points $\mathbf{P}_k$ from each source frame in the sequence, set scales based on the average distance to nearby points, and assign colors by projecting these points onto the corresponding camera image $\mathbf{I}_k$. Each Gaussian's timestamp t_i is assigned the capture time of its source LiDAR frame, and velocities are initialized to zero. We aggregate source frame Gaussians to create $\mathcal{G}_{\text{init}}$.

3.2 Predicting Flow and Rendering

Inspired by recent advances in 4D reconstruction [56, 61, 37, 68, 30, 60], we propose to learn a time-dependent velocity field to model the dynamics of driving scenes. Given the initial velocity-augmented Gaussians $\mathcal{G}_{\text{init}}$, we leverage a neural reconstruction function f_θ that outputs the refined Gaussian parameters $\mathcal{G}$ and the predicted velocities $\mathcal{V}$:

$$\mathcal{G}, \mathcal{V} = f_\theta(\mathcal{G}_{\text{init}}, \mathcal{T}). \tag{1}$$

With the predicted velocities $\mathcal{V}$, each Gaussian can be propagated from its initial timestep t_i to any target timestep t' using a linear motion model:

$$\mathbf{p}_i^{t'} = \mathbf{p}_i^{t_i} + \mathbf{v}_i \cdot (t' - t_i), \tag{2}$$

where $\mathbf{p}_i^{t'}$ is the Gaussian position at time t', $\mathbf{v}_i$ and t_i are its velocity and capture time. This formulation enables continuous, temporally consistent reconstruction under a constant velocity assumption. We find this simple motion model can already achieve reasonable performance when reconstructing outdoor driving scenes with short time horizons ($\sim 1s$), an observation aligned with existing works [37, 30, 24, 60]. Moreover, we investigate higher-order polynomial motion models, as discussed in Sec. 3.4 and Table 7.

3.3 Unsupervised Learning of Dynamics

We now describe how the method learns to disentangle the scene dynamics. The network f_θ is trained in a fully self-supervised manner, without requiring explicit 3D annotations. Given the predicted Gaussians $\mathcal{G}$, we move the Gaussians to target time t' using Eqn. (2), render the scene using differentiable rasterization [18] to generate color and depth images, and compare them against the real sensor observations $\mathcal{I}$ and $\mathcal{P}$. To prevent unnecessary motion and encourage stability, we introduce an "*as static as possible*" regularization. The total loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{vel}} \mathcal{L}_{\text{vel}}, \tag{3}$$

where $\mathcal{L}_{\text{recon}}$ represents the reconstruction loss, consisting of L_1 and structural similarity losses w.r.t the images, and an L_1 depth loss in the image plane compared to the projected LiDAR, and $\mathcal{L}_{\text{vel}}$ serves as a velocity regularization term that minimizes motion magnitudes:

$$\mathcal{L}_{\text{recon}} = \lambda_{\text{rgb}} \mathcal{L}_{\text{rgb}} + \lambda_{\text{SSIM}} \mathcal{L}_{\text{SSIM}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}}, \tag{4}$$

Table 1: **Comparison to SoTA unsupervised methods on novel view synthesis.** We evaluate photorealism, geometry, and speed metrics against per-scene optimization methods and generalizable methods. † denotes the need for pre-trained vision models. *Flux4D* surpasses unsupervised and achieves competitive performance with supervised methods (top block), without requiring 3D labels.

Methods	*Unsup.*	*Gen.*	Dynamic-only				Full image				Recon speed
			PSNR↑	SSIM↑	D_{RMSE}↓	V_{RMSE}↓	PSNR↑	SSIM↑	D_{RMSE}↓	V_{RMSE}↓	Time↓
Recon. with labels (reference)											
NeuRAD [45]	✗	✗	23.01	0.734	1.98	–	24.61	0.685	2.30	–	∼60min
StreetGS [59]	✗	✗	20.06	0.605	1.02	–	23.38	0.680	0.84	–	∼28min
G3R [7]	✗	✓	21.85	0.670	2.33	–	24.35	0.686	1.96	–	17s
Unsupervised recon.											
EmerNeRF† [61]	✓	✗	17.79	0.411	6.09	0.318	22.80	0.624	4.24	0.432	∼100min
DeSiRe-GS† [37]	✓	✗	19.08	0.477	3.36	0.297	22.25	0.608	24.89	0.322	∼120min
DepthSplat* [58]	✓	✓	16.87	0.425	6.18	–	21.40	0.595	2.73	–	0.87s
L4GM [40]	✓	✓	17.36	0.343	–	–	19.38	0.465	–	–	0.32s
STORM† [60]	✓	✓	17.65	0.367	5.24	0.203	20.79	0.508	4.80	0.238	**0.07s**
Flux4D (Ours)	✓	✓	**21.99**	**0.662**	**1.63**	**0.157**	**23.84**	**0.675**	**1.07**	**0.182**	1.8s

$$\mathcal{L}_{\text{vel}} = \frac{1}{M} \sum_i \|\mathbf{v}_i\|_2. \tag{5}$$

We train f_θ across a diverse set of scenes. Notably, we find that training across many scenes enables the network to *automatically* decompose static and dynamic components in urban scenes without requiring the complex regularizations used in prior per-scene optimization techniques [56, 61, 8, 17, 37]. This highlights the effectiveness of data-driven priors as a powerful form of implicit regularization and the scalability of this simple framework.

3.4 Improving Realism and Flow

The aforementioned components form the core of our approach, termed *Flux4D*-base. *Flux4D*-base can already disentangle motion and render novel views with high quality. We further improve *Flux4D*-base through two enhancements that further recover more fine-grained appearance and refined flow, resulting in our final model, *Flux4D*.

Iterative refinement: *Flux4D*-base recovers the overall scene appearance, but often lacks fine-grained details. We hypothesize that this limitation stems from the constrained capacity of a single-step feedforward network, and imperfect initialization due to occlusions. To mitigate this, we introduce an iterative refinement mechanism inspired by G3R [7], leveraging 3D gradients as feedback to enhance reconstruction quality. Specifically, after each forward pass and generation of rendered color and depth at the supervision views, we compute the 3D gradients of the Gaussians according to the loss function Eqn. (3), and provide the generated Gaussians and gradients as input to a network f_ϕ to further refine them. This process progressively corrects color inconsistencies and sharpens details within as few as two iterations. By incorporating iterative feedback, our method achieves higher-fidelity reconstruction, particularly in regions with complex appearance variations, while preserving the efficiency and scalability of *Flux4D*-base.

Motion enhancement: *Flux4D*-base recovers the overall scene flow accurately (Table 7). We further introduce *polynomial motion* parameterizations to better model actor behaviors like acceleration, braking or turning. Please see supp. for more details and comparisons. Exploring more advanced velocity models [24] or implicit flow representations is an exciting direction for future work. To further improve the flow and appearance quality of dynamic actors, we modify the loss function to focus on dynamic regions. Specifically, we render the flow in the image plane and apply pixel-wise re-weighting to the photometric loss. This gives higher importance to faster-moving regions during training, which typically occupy fewer pixels and would contribute less to the overall loss.

4 Experiments

We evaluate *Flux4D* against the current state-of-the-art (SoTA) self-supervised scene reconstruction methods, including both per-scene optimization and generalizable approaches. We also report the performance of supervised methods that do require annotations to model dynamics as a reference. We perform experiments on multiple outdoor dynamic datasets and assess novel view appearance

| GT | DepthSplat | L4GM | EmerNeRF | DeSiRe-GS | Ours |

Figure 3: **Qualitative results for NVS on PandaSet**. Rendered RGB images from novel views show that our method achieves better image quality across a variety of urban scenes, with crisper edges and sharper dynamic actors compared to baselines.

| GT | NeuRAD | G3R | EmerNeRF | DeSiRe-GS | Ours |

Figure 4: **NVS on longer-horizon logs.** Qualitative comparison shows that our method outperforms SoTA unsupervised baselines, by maintaining better estimation of actor movements in longer horizon. We shrink the gap in quality to supervised methods.

and depth, as well as recovered flow. We also ablate *Flux4D*'s design and show that *Flux4D* scales with more data. Finally, we demonstrate the controllability of our predicted scene representation for realistic camera simulation.

4.1 Experimental Details

Experiment setup: We conduct experiments on outdoor driving scenes from PandaSet [57] and Waymo Open Dataset (WOD) [42]. From PandaSet's 103 dynamic scenes (1080p cameras, 64-beam LiDARs, 10Hz), we select 10 diverse scenes for validation and use the rest for training. We use the front camera and 360° LiDAR, both collected at 10 Hz. To compare against existing feed-forward generalizable reconstruction methods that can only take a small number of frames as input, we report scene reconstruction results on short 1.5s windows within the validation sequences. Each method takes as input frames 0, 2, 4, 6, 8, 10, and is evaluated on frames 1, 3, 5, 7, 9 (*interpolation*) and 11-15 (*future prediction*). We sample a new snippet every 20 frames, yielding four non-overlapping evaluation snippets per log. We also evaluate against per-scene optimization methods over the full duration of the validation sequence (8 seconds) in the interpolation setting (every other frame is held out). For WOD evaluation, we follow the NVS setting in DrivingRecon [30], using the Waymo-NOTR subset with three front cameras, taking $\{t-2, t-1, t+1\}$ frames as input, and generating the interpolated frame at time t, where t is every tenth frame in each sequence. Finally, we evaluate scene flow estimation perpformance on PandaSet and WOD (official validation set with 202 logs). As existing scene flow estimation methods cannot directly predict flows at novel timesteps, we evaluate scene flow on the input frames. We restrict evaluation to LiDAR points within the camera field of view (FoV) following [60].

Baselines: We compare against SoTA unsupervised scene reconstruction approaches: (1) *Self-supervised per-scene optimization:* EmerNeRF [61] and DeSiRe-GS [37], which reconstruct dynamic scenes using geometry priors, cycle consistency, and pre-trained vision models (FiT3D [67] and DI-
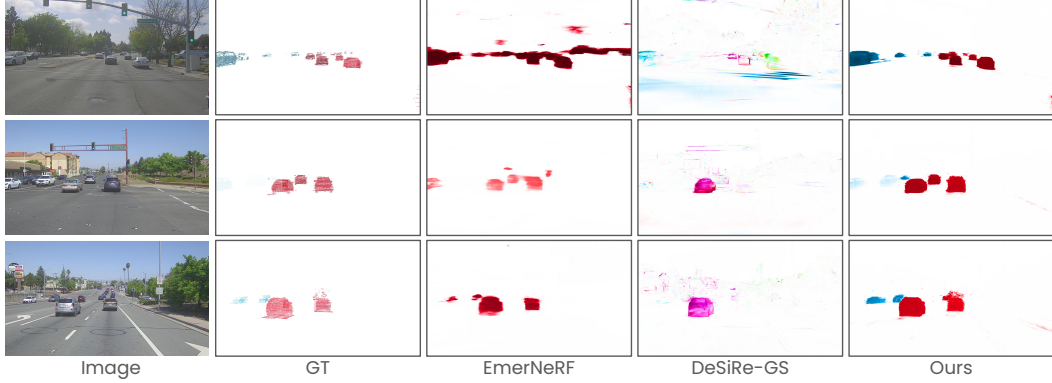
6

Figure 5: **Estimating motion flows.** We compare our estimated motion with prior unsupervised methods through rendered flow, showing accurate static region detection and sharper actor flow edges.

Table 2: **Full sequence reconstruction.** *Flux4D* outperforms unsupervised methods for 8-second reconstructions on dynamic regions and full image, closing the gap with supervised methods.

Methods	Dynamic-only				Full image			
	PSNR↑	SSIM↑	D_{RMSE} ↓	V_{RMSE} ↓	PSNR↑	SSIM↑	D_{RMSE} ↓	V_{RMSE} ↓
Recon. with labels (reference)								
NeuRAD [45]	22.99	0.719	1.71	–	24.99	0.679	2.29	–
StreetGS [59]	21.63	0.701	0.94	–	23.89	0.708	0.87	–
G3R [7]	20.60	0.573	2.16	–	23.15	0.636	2.01	–
Unsupervised recon.								
EmerNeRF† [61]	18.65	0.437	4.48	0.478	23.42	0.627	3.09	0.975
DeSiRe-GS† [37]	19.76	0.544	4.08	0.312	22.91	0.659	4.07	0.395
***Flux4D* (Ours)**	**21.94**	**0.658**	**1.57**	**0.162**	**23.72**	**0.670**	**1.10**	**0.186**

NOv2 [34]); (2) *Generalizable methods:* L4GM* [40], a 4D reconstruction model adapted to driving scenes using depth supervision; DepthSplat*, an extension of [58] that unprojects LiDAR points using estimated depth for 3D Gaussian prediction; DrivingRecon [30], which builds a 4D feed-forward model utilizing learned priors from pre-trained vision models (SAM [23] and DeepLab-v3 [4]); and STORM [60] which predicts per-pixel Gaussians and their motion in a feed-forward manner. For reference, we also include SoTA methods that use ground-truth 3D tracklets: StreetGS [59] and NeuRAD [45] (compositional 3DGS/NeRF), as well as G3R [7] (iterative refinement of composi-tional 3DGS). Apart from reconstruction methods, we also compare with representative scene flow estimation methods NSFP [27] and FastNSF [28] as a reference.

Metrics: We report standard metrics to measure the photorealism, geometric and motion accuracy using PSNR, SSIM, and depth RMSE (V_{RMSE}) and velocity RMSE (V_{RMSE}). Results are reported on both full images and dynamically moving regions for a comprehensive assessment. For scene-flow quality, we report EPE3D, Acc_5 and Acc_{10} (fraction of points with error $\leq$ 5/10 cm), angular error in radians (θ_ϵ), three-way EPE [11]: background-static (BS), foreground-static (FS), and foreground-dynamic (FD), bucketed normalized EPE [20], and inference speed. On WOD, where semantic labels are coarse, we follow EulerFlow [48] and report bucketed normalized EPE for *Background (incl. Signs)*, *Vehicles*, *Pedestrians*, and *Cyclists* only.

***Flux4D* implementation details:** We adopt a 3D U-Net with sparse convolutions [43] for f_θ. To handle unbounded scenes, we place random points on a spherical plane at a far distance to model sky and far-away regions. We also add random points within a 3D sphere following [59] to increase model robustness. Our model processes full-resolution images ($\geq 1920 \times 1080$) in all experiments and can be efficiently scaled to higher resolutions without significant overhead. Unless otherwise stated, all models are trained for 30,000 iterations on $4\times$ NVIDIA L40S (48G) GPUs, taking approximately 2 days. The reconstruction loss weights $\lambda_{rgb}, \lambda_{SSIM}, \lambda_{depth}$ are set as 0.8, 0.2 and 0.01 respectively. The velocity regularization weight λ_{vel} is set as 5e-3.

Table 3: **NVS on WOD [42].** We achieve significant improvements over generalizable baselines.

Methods	Full Image			Dynamic		Static	
	PSNR	SSIM	LPIPS	PSNR	SSIM	PSNR	SSIM
LGM [44]	17.49	0.47	0.33	17.79	0.49	15.37	0.39
PixelSplat [2]	18.24	0.56	0.30	18.63	0.58	16.96	0.44
MVSplat [5]	19.00	0.57	0.28	19.29	0.58	17.35	0.47
L4GM [40]	17.63	0.54	0.31	18.58	0.56	16.78	0.43
DrivingRecon [30]	20.63	0.61	0.21	20.97	0.62	19.70	0.51
Flux4D	**26.62**	**0.82**	**0.18**	**26.86**	**0.83**	**26.09**	**0.80**

Table 4: **Future prediction.** We surpass unsupervised and supervised methods.

Methods	PSNR↑	SSIM↑	D_{RMSE} ↓	V_{RMSE} ↓
Recon. with labels				
NeuRAD [45]	18.63	0.458	2.82	–
StreetGS [59]	16.35	0.398	1.67	–
G3R [7]	18.93	0.501	2.41	–
Unsupervised recon.				
EmerNeRF[†] [61]	15.40	0.313	7.92	0.304
DeSiRe-GS[†] [37]	15.69	0.346	14.74	0.303
STORM[†] [60]	16.61	0.338	5.82	0.218
Flux4D (Ours)	**19.07**	**0.489**	**2.57**	**0.162**

Figure 6: **High-fidelity flow and RGB reconstruction.** *Flux4D* not only provides photorealistic reconstruction of the dynamic scene but also estimates actors' motion flow with high precision.

4.2 Scalable 4D Reconstruction

Novel view synthesis on PandaSet: Table 1 and Fig. 3 compare *Flux4D* against SoTA unsupervised methods on 1s PandaSet snippets in the interpolation setting, with supervised approaches included for reference. Reconstruction speed is measured on a single RTX A5000 GPU (24GB). *Flux4D* achieves superior photorealism and geometric accuracy with fast reconstruction speed. We further evaluate our method on longer-horizon reconstruction of 8 second logs (Table 2 and Fig. 4), using iterative processing of 1s snippets. Our approach outperforms unsupervised per-scene optimization methods by a large margin on both 1s and 8s reconstruction tasks, without requiring pre-trained models or complex regularization. Our quantitative results as reported in these tables also indicate that *Flux4D* is competitive even against supervised approaches. Qualitatively, as shown in Fig. 3 and 4, *Flux4D* achieves high-fidelity camera rendering in both static and dynamic regions, while existing unsupervised approaches usually suffer from noticeable artifacts on dynamic actors due to inaccurate learned dynamics.

Novel view synthesis on WOD: We further compare *Flux4D* with SoTA generalizable methods on WOD in Table 3, where we follow the setup in [30]. The baseline results are from DrivingRecon [30] paper and we confirmed the setup and results with the authors to ensure accurate comparison. *Flux4D* surpasses DrivingRecon by +5.99 dB in PSNR and +0.21 in SSIM, demonstrating its effectiveness for unsupervised dynamic scene reconstruction. Please see supp. for qualitative comparisons.

Flow estimation: We compare the estimated motion flows of *Flux4D* with existing unsupervised per-scene optimization methods EmerNeRF [61] and DeSiRe-GS [37]. As shown in Table 1, 2 and Fig. 5, *Flux4D* significantly outperforms prior approaches, learning accurate motion direction and magnitude without any supervision. In contrast, existing methods struggle to learn consistent motion flows and fully decompose dynamic scenes, leading to inaccurate and incoherent motion predictions, limiting their applicability in downstream tasks.

Scene flow evaluation: While *Flux4D* primarily focuses on reconstruction and is not specifically designed for scene flow estimation, we further evaluate its performance on PandaSet compared with representative scene flow estimation methods using standard scene flow metrics in Table 5 and 6. Please see supp. for comparisons on WOD. Although not designed for scene flow estimation, Flux4D achieves superior performance across most scene flow metrics using only reconstruction-based supervision (RGB + depth). Notably, it outperforms other methods on smaller or less common object categories such as wheeled VRUs, other vehicles, and pedestrians, as shown in bucketed evaluations. These results highlight a promising path to unifying state-of-the-art scene flow estimation [20, 20, 22, 25] and reconstruction within a single framework.

8

Table 5: **Comparison with scene flow estimation methods.**

Method	EPE3D $\downarrow$	Acc_5 $\uparrow$	Acc_{10} $\uparrow$	θ_ϵ $\downarrow$	EPE-BS $\downarrow$	EPE-FS $\downarrow$	EPE-FD $\downarrow$	EPE-3way $\downarrow$	Inference time $\downarrow$
NSFP [27]	0.183	0.558	0.713	0.510	0.106	0.103	0.573	0.227	∼5.57 s/frame
FastNSF [28]	0.194	0.571	0.714	0.471	0.155	0.134	0.428	0.211	∼0.68 s/frame
STORM [60]	0.120	0.757	0.782	0.489	**0.009**	**0.098**	0.536	0.201	**∼0.01 s/frame**
Flux4D	**0.094**	**0.775**	**0.807**	**0.123**	0.019	0.117	**0.391**	**0.165**	∼0.31 s/frame

Table 6: **Bucketed scene flow error on PandaSet.** Normalized EPE3D ($\downarrow$) per class, split into static (S) and dynamic (D) regions. Mean S/D are averages across all buckets. Abbrev.: BG = Background, CAR = Car, WVRU = Wheeled VRU, VEH = Other Vehicles, PED = Pedestrian.

Method	BG-S$\downarrow$	CAR-S$\downarrow$	CAR-D$\downarrow$	WVRU-S$\downarrow$	WVRU-D$\downarrow$	VEH-S$\downarrow$	VEH-D$\downarrow$	PED-S$\downarrow$	PED-D$\downarrow$	Mean S$\downarrow$	Mean D$\downarrow$
NSFP [27]	0.128	0.093	0.668	0.046	0.975	0.060	0.819	0.071	0.945	0.080	0.852
FastNSF [28]	0.196	0.153	**0.581**	0.043	0.960	0.075	0.701	0.041	**0.894**	0.102	**0.784**
STORM [60]	**0.005**	0.087	0.713	**0.000**	1.000	0.195	1.000	0.093	1.012	0.076	0.931
Flux4D	0.019	**0.078**	0.701	0.011	**0.866**	**0.021**	**0.661**	**0.027**	0.966	**0.031**	0.800

Future prediction: We evaluate *Flux4D*'s capability for future frame prediction beyond the observed frames. This challenging task requires precise motion estimation, temporal consistency, occlusion reasoning, and a comprehensive 4D scene understanding. As shown in Table 4, *Flux4D* outperforms existing unsupervised methods in both photometric accuracy and geometric consistency. Moreover, *Flux4D* even outperforms supervised approaches that rely on imperfect explicit annotations for extrapolation, demonstrating the robustness of our predicted scene representation and the effectiveness of unsupervised scene flow prediction. This highlights *Flux4D*'s ability to model scene dynamics, which is critical for world modeling, simulation, and scene understanding in autonomous systems. We report dynamic-only metrics in Table 4 and defer full-image metrics to supp.

Ablation: Table 7 evaluates *Flux4D*'s key design components. Iterative refinement significantly enhances image quality and geometric accuracy metrics. Polynomial motion modeling improves motion prediction performance. Table 8 demonstrates that our static-preference prior is essential to learning accurate flow, and that velocity reweighting improves performance on the dynamic elements. Please refer to supp. for qualitative comparisons.

LiDAR-free *Flux4D*: We show that *Flux4D* can also operate in a LiDAR-free mode at inference similar to DrivingRecon [30] and STORM [60] by using off-the-shelf monocular depth estimation model [16]. As shown in Table 9, the flow estimation performance remains comparable, and in some cases, the visual realism improves in background regions (*e.g.*, buildings) due to the broader coverage provided by monocular depth, particularly in areas where LiDAR sparsity limits reconstruction quality. Combining both LiDAR and points lifted by monocular depth yields the best overall realism.

Scaling analysis: *Flux4D*'s effectiveness stems from multi-scene training, leveraging diverse driving data as implicit regularization. Unlike per-scene methods that require complex regularizations or pre-trained models, increasing the amount of training data naturally improves scene decomposition and motion estimation. Analysis on PandaSet and WOD shows consistent improvements in photometric accuracy and motion estimation as training data scale. This confirms unsupervised 4D reconstruction benefits significantly from diverse real-world scenarios, suggesting *Flux4D* can continue improving with additional data, making it promising for scalable scene reconstruction.

Camera Simulation: We showcase applying *Flux4D* for high-fidelity camera simulation in large-scale driving scenarios. *Flux4D* produces high-quality motion flows in diverse, large-scale dynamic scenes on PandaSet (Fig. 6), Argoverse 2 [54], and WOD (Fig. 7). This allows accurate scene decomposition across diverse environments which is critical for instance extraction and direct manipulation of dynamic elements (Fig. 9). Compared to existing self-supervised per-scene methods, *Flux4D* is better suited for interactive and controllable applications, as it reconstructs an editable representation that supports instance mask extraction, scene editing and object manipulation for various downstream tasks. In Fig. 9, we demonstrate *Flux4D*'s capability to render realistic images of the modified scene representation. Notably, our approach achieves this without requiring labels.

5 Limitations

Although *Flux4D* achieves SoTA 4D reconstruction without any annotations or pre-trained models, three key limitations remain: (1) flow estimation for highly dynamic actors with complex motion
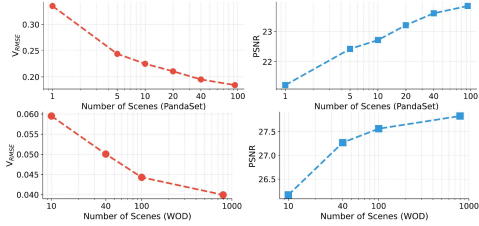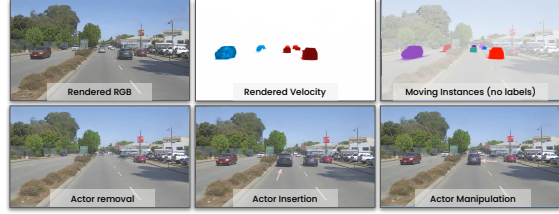
Table 7: **Ablation study on *Flux4D* designs.**

Methods	Dynamic-only			
	PSNR↑	SSIM↑	$D_{RMSE}\downarrow$	$V_{RMSE}\downarrow$
Flux4D-base	18.89	0.472	1.98	**0.165**
+ iterative refinement	21.32	0.636	1.66	0.167
+ polynomial motion	**21.45**	**0.641**	**1.55**	0.167

Table 8: **Ablation study on training strategy.**

Methods	Dynamic-only			
	PSNR↑	SSIM↑	$D_{RMSE}\downarrow$	$V_{RMSE}\downarrow$
Flux4D	**21.99**	**0.662**	1.63	**0.157**
− vel. reweighting	21.45	0.641	1.55	0.167
− vel. regularization	21.08	0.614	**1.44**	0.532

Table 9: **LiDAR-free *Flux4D* using off-the-shelf monocular depth estimation model [16].**

Methods	Dynamic-only				Full image				Scene Flow
	PSNR↑	SSIM↑	$D_{RMSE}\downarrow$	$V_{RMSE}\downarrow$	PSNR↑	SSIM↑	$D_{RMSE}\downarrow$	$V_{RMSE}\downarrow$	EPE-3way↓
Flux4D (monocular depth only)	21.71	0.668	**1.45**	0.159	23.87	0.688	1.23	0.186	0.165
Flux4D (LiDAR, Table 1)	**21.99**	0.662	1.63	**0.157**	23.84	0.675	**1.07**	**0.182**	0.165
Flux4D (LiDAR + monocular depth)	**21.99**	**0.682**	1.52	0.158	**24.55**	**0.726**	1.11	0.184	**0.161**

Figure 7: ***Flux4D* reconstruction on Argoverse 2 and WOD.**

Figure 8: **Scaling analysis.** Increasing number of training scenes for *Flux4D* consistently improves performance.

Figure 9: **Simulation applications.** Flux4D can be applied suc- cessfully to different camera simulation tasks, e.g., actor removal, insertion and manipulation.

patterns is challenging, which could be mitigated by leveraging larger and more diverse training data; (2) iterative approach for long-horizon reconstruction creates visible inconsistencies at transition points; and (3) the method assumes a simple pinhole camera model with clean LiDAR data, limiting applicability with rolling shutter cameras or noisy sensor inputs. Please see supp. for more examples. Future work will focus on scaling to larger datasets, developing a unified temporal representation for seamless long-term reconstruction, and improving robustness to real-world sensor imperfections. Furthermore, *Flux4D*'s explicit 3D representation offers interpretable structure for world models. Overall, we believe that our simple and scalable design serves as a foundation for the community to build upon, enabling further advancements in 4D reconstruction.

6 Conclusion

We present *Flux4D*, a scalable flow-based unsupervised framework for reconstructing large-scale dynamic scenes by directly predicting 3D Gaussians and their motion dynamics. By relying solely on photometric losses and enforcing an "as static as possible" regularization, *Flux4D* effectively decomposes dynamic elements without requiring any supervision, pre-trained models, or foundational priors. Our method enables fast reconstruction, scales efficiently to large datasets, and generalizes well to unseen environments. Extensive experiments on outdoor driving datasets demonstrate state-of-the-art performance in scalability, generalization, and reconstruction quality. We hope this work paves the way for efficient, unsupervised 4D scene reconstruction at scale.

Acknowledgement

References

[1] Ben Agro, Quinlan Sykora, Sergio Casas, Thomas Gilles, and Raquel Urtasun. Uno: Unsupervised occupancy fields for perception and forecasting. In *CVPR*, 2024. 3

[2] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 2, 8

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, 2021. 2

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3, 7

[5] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, 2024. 2, 8

[6] Yun Chen, Matthew Haines, Jingkang Wang, Krzysztof Baron-Lis, Sivabalan Manivasagam, Ze Yang, and Raquel Urtasun. Salf: Sparse local fields for multi-sensor rendering in real-time. *arXiv preprint arXiv:2507.18713*, 2025. 1

[7] Yun Chen, Jingkang Wang, Ze Yang, Sivabalan Manivasagam, and Raquel Urtasun. G3R: Gradient guided generalizable reconstruction. In *ECCV*, 2025. 2, 3, 5, 7, 8

[8] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv preprint arXiv:2311.18561*, 2023. 2, 5

[9] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *ICLR*, 2023. 3

[10] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024. 2

[11] Nathaniel Chodosh, Deva Ramanan, and Simon Lucey. Re-evaluating lidar scene flow for autonomous driving. In *WACV*, 2024. 7

[12] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *arXiv preprint arXiv:2405.17398*, 2024. 3

[13] Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving. *arXiv preprint arXiv:2411.16816*, 2024. 2

[14] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. In *ICLR*, 2024. 2

[15] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023. 3

[16] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. In *TPAMI*, 2024. 9, 10

[17] Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. S3gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024. 2, 5

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. In *TOG*, 2023. 1, 2, 4

[19] Mustafa Khan, Hamidreza Fazlali, Dhruv Sharma, Tongtong Cao, Dongfeng Bai, Yuan Ren, and Bingbing Liu. Autosplat: Constrained gaussian splatting for autonomous driving scene reconstruction. *arXiv preprint arXiv:2407.02598*, 2024. 1

[20] Ishan Khatri, Kyle Vedder, Neehar Peri, Deva Ramanan, and James Hays. I can't believe it's not scene flow! In *ECCV*, 2024. 7, 8

[21] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4d occupancy forecasting. In *CVPR*, 2023. 3

[22] Jaeyeul Kim, Jungwan Woo, Ukcheol Shin, Jean Oh, and Sunghoon Im. Flow4d: Leveraging 4d voxel network for lidar scene flow estimation. In *RA-L*, 2025. 8

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 3, 7

[24] Jinxi Li, Ziyang Song, Siyuan Zhou, and Bo Yang. Freegave: 3d physics learning from dynamic videos by gaussian velocity. In *CVPR*, 2025. 4, 5

[25] Siyi Li, Qingwen Zhang, Ishan Khatri, Kyle Vedder, Deva Ramanan, and Neehar Peri. Uniflow: Towards zero-shot lidar scene flow for autonomous vehicles via cross-domain generalization. *arXiv preprint arXiv:2511.18254*, 2025. 8

[26] Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. Drivingdiffusion: Layout-guided multi-view driving scenarios video generation with latent diffusion model. In *ECCV*, 2024. 3

[27] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. In *NeurIPS*, 2021. 7, 9

[28] Xueqian Li, Jianqiao Zheng, Francesco Ferroni, Jhony Kaesemodel Pontes, and Simon Lucey. Fast neural scene flow. In *CVPR*, 2023. 7, 9

[29] Jeffrey Yunfan Liu, Yun Chen, Ze Yang, Jingkang Wang, Sivabalan Manivasagam, and Raquel Urtasun. Real-time neural rasterization for large scenes. In *ICCV*, 2023. 2

[30] Hao Lu, Tianshuo Xu, Wenzhao Zheng, Yunpeng Zhang, Wei Zhan, Dalong Du, Masayoshi Tomizuka, Kurt Keutzer, and Yingcong Chen. Drivingrecon: Large 4d gaussian reconstruction model for autonomous driving. *arXiv preprint arXiv:2412.09043*, 2024. 2, 3, 4, 6, 7, 8, 9

[31] Sivabalan Manivasagam, Ioan Andrei Bârsan, Jingkang Wang, Ze Yang, and Raquel Urtasun. Towards zero domain gap: A comprehensive study of realistic LiDAR simulation for autonomy testing. In *ICCV*, 2023. 1

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[33] Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, et al. Driveworld: 4d pre-trained scene understanding via world models for autonomous driving. In *CVPR*, 2024. 3

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7

[35] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs. In *CVPR*, 2021. 2

[36] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021. 2

[37] Chensheng Peng, Chengwei Zhang, Yixiao Wang, Chenfeng Xu, Yichen Xie, Wenzhao Zheng, Kurt Keutzer, Masayoshi Tomizuka, and Wei Zhan. Desire-gs: 4d street gaussians for static-dynamic decomposition and surface reconstruction for urban driving scenes. *arXiv preprint arXiv:2411.11921*, 2024. 2, 4, 5, 6, 7, 8

[38] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2

[39] Ava Pun, Gary Sun, Jingkang Wang, Yun Chen, Ze Yang, Sivabalan Manivasagam, Wei-Chiu Ma, and Raquel Urtasun. Neural lighting simulation for urban scenes. In *NeurIPS*, 2023. 2

[40] Jiawei Ren, Cheng Xie, Ashkan Mirzaei, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, Huan Ling, et al. L4gm: Large 4d gaussian reconstruction model. In *NeurIPS*, 2025. 2, 5, 7, 8

[41] Xuanchi Ren, Yifan Lu, Hanxue Liang, Jay Zhangjie Wu, Huan Ling, Mike Chen, Francis Fidler, Sanja annd Williams, and Jiahui Huang. Scube: Instant large-scale scene reconstruction using voxsplats. In *NeurIPS*, 2024. 2

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 2, 6, 8

[43] Haotian Tang, Shang Yang, Zhijian Liu, Ke Hong, Zhongming Yu, Xiuyu Li, Guohao Dai, Yu Wang, and Song Han. Torchsparse++: Efficient training and inference framework for sparse convolution on gpus. In *MICRO*, 2023. 7

[44] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *ECCV*, 2024. 8

[45] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. NeuRAD: Neural rendering for autonomous driving. In *CVPR*, 2024. 1, 5, 7, 8

[46] 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. *arXiv preprint arXiv:2510.12901*, 2025. 1

[47] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *CVPR*, 2023. 2

[48] Kyle Vedder, Neehar Peri, Ishan Khatri, Siyi Li, Eric Eaton, Mehmet Kemal Kocamaz, Yue Wang, Zhiding Yu, Deva Ramanan, and Joachim Pehserl. Neural eulerian scene flow fields. In *ICLR*, 2025. 7

[49] Jingkang Wang, Sivabalan Manivasagam, Yun Chen, Ze Yang, Ioan Andrei Bârsan, Anqi Joyce Yang, Wei-Chiu Ma, and Raquel Urtasun. CADSim: Robust and scalable in-the-wild 3d reconstruction for controllable sensor simulation. In *CoRL*, 2022. 2

[50] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *CVPR*, 2021. 1

[51] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2

[52] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *ECCV*, 2024. 3

[53] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality meshes. *arXiv preprint arXiv:2404.12385*, 2024. 2

[54] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 9

[55] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 2024. 2

[56] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. In *NeurIPS*, 2022. 2, 4, 5

[57] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *ITSC*, 2021. 2, 6

[58] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*, 2024. 5, 7

[59] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. In *ECCV*, 2024. 1, 2, 5, 7, 8

[60] Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Maximilian Igl, Apoorva Sharma, Peter Karkus, Danfei Xu, Boris Ivanovic, Yue Wang, and Marco Pavone. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. *arXiv preprint arXiv:2501.00602*, 2025. 2, 3, 4, 5, 6, 7, 8, 9

[61] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 2, 4, 5, 6, 7, 8

[62] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *CVPR*, 2023. 1, 2

[63] Ze Yang, Sivabalan Manivasagam, Yun Chen, Jingkang Wang, Rui Hu, and Raquel Urtasun. Reconstructing objects in-the-wild for realistic sensor simulation. In *ICRA*, 2023. 2

[64] Ze Yang, Jingkang Wang, Haowei Zhang, Sivabalan Manivasagam, Yun Chen, and Raquel Urtasun. Genassets: Generating in-the-wild 3d assets in latent space. In *CVPR*, 2025. 2

[65] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *CVPR*, 2024. 3

[66] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024. 2

[67] Yuanwen Yue, Anurag Das, Francis Engelmann, Siyu Tang, and Jan Eric Lenssen. Improving 2D Feature Representations by 3D-Aware Fine-Tuning. In *ECCV*, 2024. 6

[68] Haiming Zhang, Wending Zhou, Yiyao Zhu, Xu Yan, Jiantao Gao, Dongfeng Bai, Yingjie Cai, Bingbing Liu, Shuguang Cui, and Zhen Li. Visionpad: A vision-centric pre-training paradigm for autonomous driving. *arXiv preprint arXiv:2411.14716*, 2024. 4

[69] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large reconstruction model for 3D gaussian splatting. In *ECCV*, 2025. 2

[70] Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Learning unsupervised world models for autonomous driving via discrete diffusion. In *ICLR*, 2024. 3

[71] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *ECCV*, 2024. 3

[72] 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 *CVPR*, 2024. 1

[73] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint arXiv:2410.12781*, 2024. 3