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.

Advances in differentiable rendering approaches such as Neural Radiance Field (NeRF) [26] and 3D Gaussian Splatting (3DGS) [17] have enabled high-quality reconstruction of dynamic scenes [53, 50, 62, 39, 18]. 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.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

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 [31] or cycle consistency [52], or performing multi-stage training [16]. Another strategy is to leverage foundation models for additional semantic features or priors [31, 7, 52]. 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, 42, 2, 5, 13, 44, 59] 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 [33]. When handling large scenes with many dynamic elements, they rely on costly annotations [6, 34], limiting their scalability. Most recently, DrivingRecon [25] and STORM [51] 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 [48] and WOD [36] 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 ($\geq$ 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 [26, 17], recent approaches use deformation fields [32, 30, 56, 46] 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 and dynamic elements [29, 54, 40, 50, 9, 12], they remain limited by annotation quality and availability. Self-supervised alternatives using motion cues and physics-informed priors [47, 52, 7, 16, 31] 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, 42, 2, 5, 13, 44, 59], leveraging large training datasets

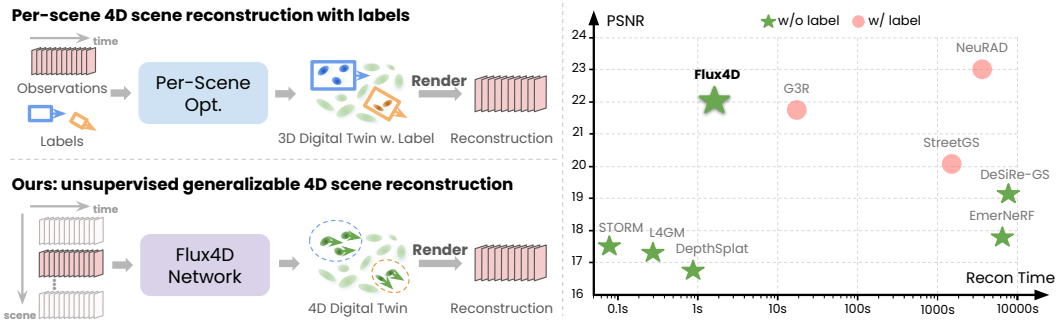

Figure 1: ***Flux4D* is a simple and scalable framework for unsupervised 4D reconstruction. Left:** Paradigms for 4D reconstruction. **Right:** realism-speed comparisons with existing works.

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 [63] or iterative refinement [6], 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 [14, 11, 43, 61, 27] 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 [14, 11, 22] or LiDAR [60, 55, 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 [25] and STORM [51] 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 [20], ViT-Adapter [8]) 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

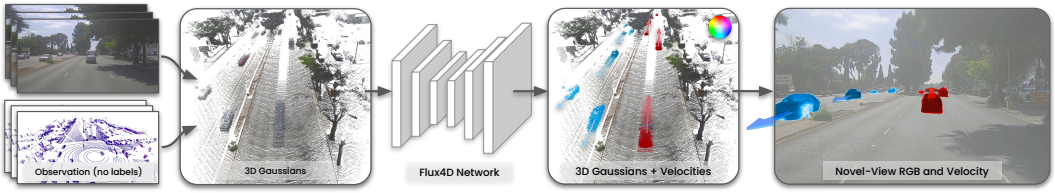

Figure 2: **Model overview.** *Flux4D* reconstructs 4D world by predicting 3D Gaussians with velocities given unlabelled sensor observations, and trained with the photometric reconstruction objective. The resultant model can be used for RGB and flow synthesis from novel views.

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 ($\mathbb{R}^1$) [17]. 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 [47, 52, 31, 58, 25, 51], 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 [31, 25, 21, 51]. 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 [17] 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 [39]	✗	✗	23.01	0.734	1.98	–	24.61	0.685	2.30	–	∼60min
StreetGS [50]	✗	✗	20.06	0.605	1.02	–	23.38	0.680	0.84	–	∼28min
G3R [6]	✗	✓	21.85	0.670	2.33	–	24.35	0.686	1.96	–	17s
Unsupervised recon.											
EmerNeRF† [52]	✓	✗	17.79	0.411	6.09	0.318	22.80	0.624	4.24	0.432	∼100min
DeSiRe-GS† [31]	✓	✗	19.08	0.477	3.36	0.297	22.25	0.608	24.89	0.322	∼120min
DepthSplat* [49]	✓	✓	16.87	0.425	6.18	–	21.40	0.595	2.73	–	0.87s
L4GM [33]	✓	✓	17.36	0.343	–	–	19.38	0.465	–	–	0.32s
STORM [51]	✓	✓	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**	3.9s

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

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 [47, 52, 7, 16, 31]. 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 [6], 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 [21] 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

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.

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 [48] and Waymo Open Dataset (WOD) [36]. 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 [25], 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 [51].

Baselines: We compare against SoTA unsupervised scene reconstruction approaches: (1) *Self-supervised per-scene optimization:* EmerNeRF [52] and DeSiRe-GS [31], which reconstruct dynamic scenes using geometry priors, cycle consistency, and pre-trained vision models (FiT3D [57] and DI-

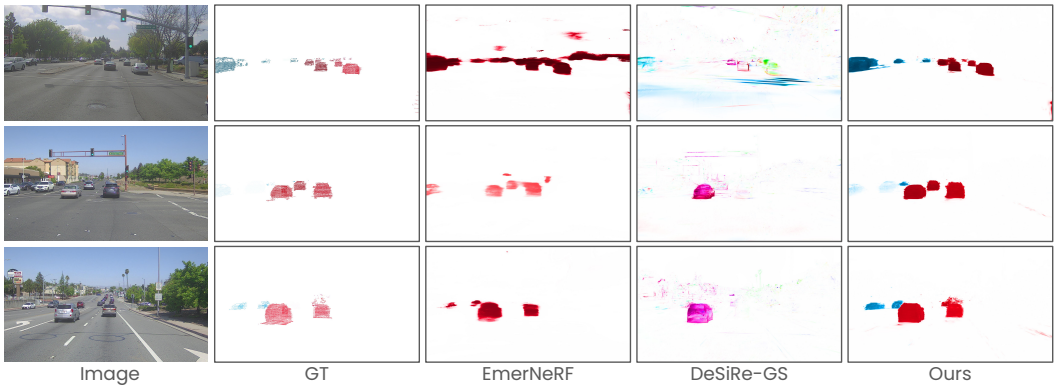

Figure 5: **Estimating motion flows.** We compare our estimated motion with prior unsupervised methods through rendered flow, showing accurate static region detection and sharper actor flow edges.

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 [39]	22.99	0.719	1.71	–	24.99	0.679	2.29	–
StreetGS [50]	21.63	0.701	0.94	–	23.89	0.708	0.87	–
G3R [6]	20.60	0.573	2.16	–	23.15	0.636	2.01	–
Unsupervised recon.								
EmerNeRF† [52]	18.65	0.437	4.48	0.478	23.42	0.627	3.09	0.975
DeSiRe-GS† [31]	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 [28]); (2) *Generalizable methods:* L4GM* [33], a 4D reconstruction model adapted to driving scenes using depth supervision; DepthSplat*, an extension of [49] that unprojects LiDAR points using estimated depth for 3D Gaussian prediction; DrivingRecon [25], which builds a 4D feed-forward model utilizing learned priors from pre-trained vision models (SAM [20] and DeepLab-v3 [4]); and STORM [51] 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 [50] and NeuRAD [39] (compositional 3DGS/NeRF), as well as G3R [6] (iterative refinement of compositional 3DGS). Apart from reconstruction methods, we also compare with representative scene flow estimation methods NSFP [23] and FastNSF [24] 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 [10]: background-static (BS), foreground-static (FS), and foreground-dynamic (FD), bucketed normalized EPE [19], and inference speed. On WOD, where semantic labels are coarse, we follow EulerFlow [41] 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 [37] 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 [50] 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 $\lambda_{\text{rgb}}, \lambda_{\text{SSIM}}, \lambda_{\text{depth}}$ 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 [36].** We achieve significant improvements over generalizable baselines.

Methods	Full Image			Dynamic		Static	
	PSNR	SSIM	LPIPS	PSNR	SSIM	PSNR	SSIM
LGM [38]	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 [33]	17.63	0.54	0.31	18.58	0.56	16.78	0.43
DrivingRecon [25]	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 [39]	21.52	0.557	3.03	–
StreetGS [50]	19.09	0.499	1.49	–
G3R [50]	21.13	0.570	2.09	–
Unsupervised recon.				
EmerNeRF† [52]	19.64	0.516	5.00	0.346
DeSiRe-GS† [31]	18.86	0.513	26.07	0.325
STORM [51]	19.63	0.489	5.19	0.251
***Flux4D* (Ours)**	**21.81**	**0.598**	**1.42**	**0.193**

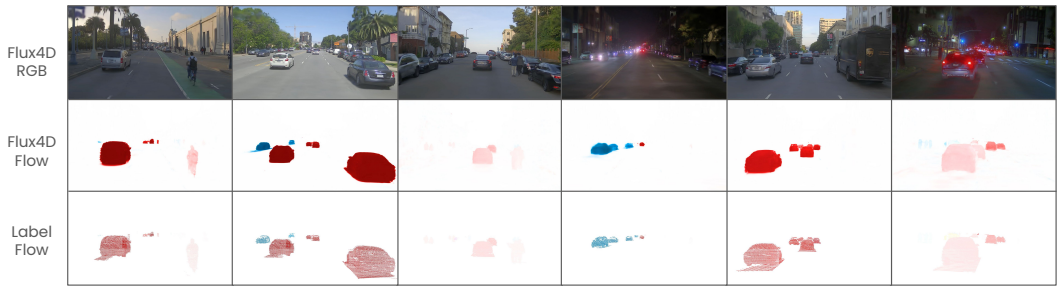

Figure 6: **High-fidelity flow and RGB reconstruction.** *Flux4D* not only provides photorealistic reconstruction of the dynamic scene but also estimates actors' motion flow with high precision.

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 [25]. The baseline results are from DrivingRecon [25] 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 [52] and DeSiRe-GS [31]. 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 and reconstruction within a single framework.

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

Method	EPE3D ↓	Acc_5 ↑	Acc_{10} ↑	θ_ϵ ↓	EPE-BS ↓	EPE-FS ↓	EPE-FD ↓	EPE-3way ↓	Inference time ↓
NSFP [23]	0.183	0.558	0.713	0.510	0.106	0.103	0.573	0.227	∼5.57 s/frame
FastNSF [24]	0.194	0.571	0.714	0.471	0.155	0.134	0.428	0.211	∼0.68 s/frame
STORM [51]	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 (↓) 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↓	CAR-S↓	CAR-D↓	WVRU-S↓	WVRU-D↓	VEH-S↓	VEH-D↓	PED-S↓	PED-D↓	Mean S↓	Mean D↓
NSFP [23]	0.128	0.093	0.668	0.046	0.975	0.060	0.819	0.071	0.945	0.080	0.852
FastNSF [24]	0.196	0.153	**0.581**	0.043	0.960	0.075	0.701	0.041	**0.894**	0.102	**0.784**
STORM [51]	**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 full-image metrics in Table 4 and report dynamic-only metrics in 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 [25] and STORM [51] by using off-the-shelf monocular depth estimation model [15]. 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 [45], 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} ↓	V_{RMSE} ↓
Flux4D-base	18.89	0.472	1.98	**0.165**
+ iterative refine	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} ↓	V_{RMSE} ↓
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 [15].**

Methods	Dynamic-only				Full image				Scene Flow
	PSNR ↑	SSIM ↑	D_{RMSE} ↓	V_{RMSE} ↓	PSNR ↑	SSIM ↑	D_{RMSE} ↓	V_{RMSE} ↓	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.**

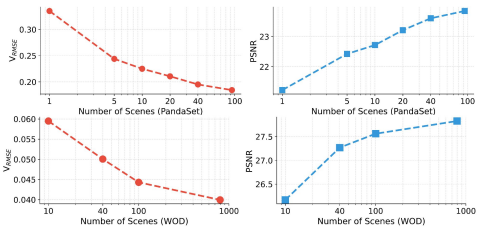

Figure 8: **Scaling analysis.** Increasing number of training scenes for *Flux4D* consistently improves performance.

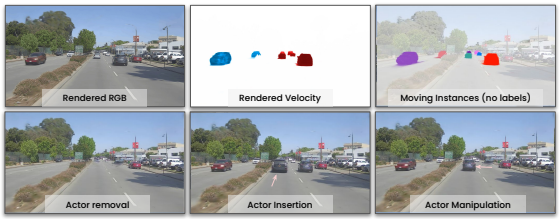

Figure 9: **Simulation applications.** Flux4D can be applied suc- cessfully to different camera simulation tasks, e.g., actor removal, insertion and manipulation.

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

We sincerely thank the anonymous reviewers for their insightful suggestions especially on scene flow evaluation, paper framing, and additional experiments using monocular depth estimation models. We would like to thank Andrei Bârsan and Joyce Yang for their feedback on the early draft. We also thank the Waabi team for their valuable assistance and support.

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

[7] 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

[8] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[9] 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

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

[11] 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

[12] 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

[13] 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

[14] 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

[15] 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

[16] 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

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

[18] 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

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

[20] 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

[21] Jinxi Li, Ziyang Song, and Bo Yang. GVFi: Learning 3d gaussian velocity fields from dynamic videos, 2025. 4, 5

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

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

[24] Xueqian Li, Jianqiao Zheng, Francesco Ferroni, Jhony Kaesemodel Pontes, and Simon Lucey. Fast neural scene flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9878–9890, 2023. 7, 9

[25] 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

[26] 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

[27] 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

[28] 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

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

[30] 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

[31] 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

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

[33] 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

[34] 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

[35] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. *CVPR*, 2020. 17

[36] 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

[37] 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

[38] 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

[39] 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

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

[41] 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

[42] 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

[43] 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

[44] 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

[45] 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, 17

[46] 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

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

[48] 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, 17

[49] 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

[50] 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

[51] 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

[52] 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

[53] 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

[54] 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. 2

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

[56] 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

[57] 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

[58] 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

[59] 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

[60] Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Learning unsupervised world models for autonomous driving via discrete diffusion. *arXiv preprint arXiv:2311.01017*, 2023. 3

[61] 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

[62] 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

[63] 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

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: The claims in the abstract and introduction accurately reflect the contributions and scope of our work. The stated claims are well aligned with both empirical results presented.

   Guidelines:

   - The answer NA 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 NA 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: We discuss our limitations in Sec. 5. We further provide qualitative examples of failure cases and more discussions about the limitations in the supplemental material.

   Guidelines:

   - The answer NA 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: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA 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: To the best of our knowledge, we have provided all the information needed to reproduce the main experimental results of the paper. We provide the complete architectural designs, data splits, experiment settings, training hyperparameters, loss weights, and more in both the main paper (Sec. 4) and the supplementary material.

Guidelines:

- The answer NA 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: [No]

   Justification: We use public datasets [48, 45, 35] in this work. We are unable to release code at the time of submission. We recognize the importance of reproducibility and are actively exploring the possibility of releasing the code with the camera-ready version. In the meantime, we provide pseudocode and implementation details, as well as training and evaluation procedures in the main paper or supplementary.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all implementation, training, testing, and data details in both the main paper and supplementary material. This includes data splits, hyperparameters, loss weights, pseudocode, optimizer type, architectural design, experimental settings, and more.

   Guidelines:

   - The answer NA 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: [No]

   Justification: The paper does not include error bars or statistical significance tests. This is consistent with common practice in the 3D vision field, where such results are typically considered self-evident due to the large performance gaps.

   Guidelines:

- The answer NA 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 the full breakdown of our experiments compute resources for each experiment in the supplementary material. We also provide our rough estimate of the compute resources needed for development.

   Guidelines:

   - The answer NA 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: To the best of our knowledge, our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: We discuss the broader impact in the supplementary material.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper does not pose such risks.

Guidelines:

- The answer NA 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 a complete summary of the licenses of all assets we used (datasets, codebases) in the supplementary material.

Guidelines:

- The answer NA 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: [NA]

    Justification: Our paper experiments with exisitng, open-source datasets and does not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Our core method does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.