SynFlow: Scaling Up LiDAR Scene Flow Estimation with Synthetic Data

Qingwen Zhang[1], Xiaomeng Zhu[1], Chenhan Jiang[2], and Patric Jensfelt[1]

[1] KTH Royal Institute of Technology, Stockholm, Sweden
[2] Hong Kong University of Science and Technology, Hong Kong, China

Abstract. Reliable 3D dynamic perception requires models that can anticipate motion beyond predefined categories, yet progress is hindered by the scarcity of dense, high-quality motion annotations. While self-supervision on unlabeled real data offers a path forward, empirical evidence suggests that scaling unlabeled data fails to close the performance gap due to noisy proxy signals. In this paper, we propose learning robust real-world motion priors entirely from scalable simulation. We introduce SynFlow, a data generation pipeline for large-scale synthetic LiDAR scene flow. Unlike prior works that prioritize sensor-specific realism, SynFlow employs a motion-oriented strategy to synthesize diverse kinematic patterns across 4,000 sequences ($\sim$940k frames), termed SynFlow-4k. This represents a $34\times$ scale-up in annotated volume over existing real-world benchmarks. Our experiments demonstrate that SynFlow-4k provides a highly domain-invariant motion prior. In a zero-shot regime, models trained only on our synthetic data generalize across multiple real-world benchmarks, comparable to in-domain supervised baselines on nuScenes and outperforming state-of-the-art methods on TruckScenes by 31.8%. Furthermore, SynFlow-4k serves as a label-efficient foundation: fine-tuning with only 5% of real-world labels surpasses models trained from scratch on the full available budget. We open-source the pipeline and dataset to facilitate research in generalizable 3D motion estimation. More detail can be found at `https://kin-zhang.github.io/SynFlow`.

Keywords: Scene Flow Estimation · Synthetic Data · Autonomous Driving

1 Introduction

Autonomous driving systems need to understand motion, not only recognize objects. Most 3D perception pipelines approach this through detection and tracking, where motion is inferred from objects belonging to predefined semantic categories [25, 35]. This object-centric formulation is effective for common traffic participants, but becomes less reliable for rare, ambiguous, or previously unseen movers. Scene flow offers a category-free alternative: it estimates dense, point-wise 3D motion directly from geometry, providing a motion-centric representation for downstream planning and interaction.

Despite its importance, progress in LiDAR scene flow is constrained by the scarcity of reliable supervision [36]. Acquiring dense and accurate 3D motion annotations for real-world LiDAR data is expensive and practically infeasible at scale [31]. To mitigate this issue, recent works [10, 18, 45] have turned to self-supervised learning on large

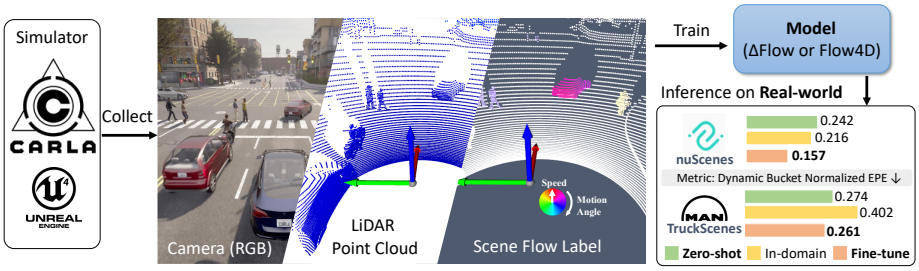

Fig. 1: Scaling up LiDAR Scene Flow with Synthetic Data. We present SynFlow, a data generation pipeline leveraging the CARLA simulator to synthesize diverse, perfectly labeled LiDAR scene flow data (center). While real-world datasets are often constrained by high annotation costs and limited scenario diversity, SynFlow provides a scalable source of dense, noise-free supervision for learning robust motion priors. As shown in the results (right), models trained on SynFlow dataset achieve strong zero-shot generalization on real-world benchmarks and significantly outperform in-domain baselines when fine-tuned on a small subset of real data.

collections of unlabeled driving data. However, these approaches rely on geometric consistency assumptions, such as rigidity cluster or temporal alignment, to construct proxy self-supervision signals. These signals are inherently noisy and under-constrained, particularly in the presence of sensor sparsity, measurement noise. Consequently, scaling up unlabeled real-world data yields diminishing returns [45], leaving a substantial performance gap compared to fully supervised methods.

This bottleneck motivates us to rethink how we source motion supervision. Instead of relying on expensive human labels or noisy self-supervision, we explore whether robust motion priors can be learned entirely from scalable simulation. We hypothesize that LiDAR scene flow learning depends primarily on capturing diverse kinematic physics rather than specific visual textures. Since simulators inherently generate precise rigid-body motion, they can provide reliable supervision even without perfect photorealism. Simulation further offers a unique advantage regarding data composition. Unlike real-world data collection [1,21,27], which is passive and limited to capturing events as they occur, simulation allows us to actively control the environment. We can easily vary sensor configurations, spawn diverse dynamic agents, and initialize traffic in specific scenarios. This ensures the model learns from a comprehensive range of motion patterns across various scenarios. However, despite these advantages, existing synthetic LiDAR pipelines [39,40] have been optimized primarily for semantic realism or sensor-specific noise to support detection and segmentation tasks. To the best of our knowledge, no framework has yet been specifically designed to prioritize the dense kinematic complexity required for scene flow.

In this work, we introduce SynFlow, a data generation pipeline for LiDAR scene flow built upon the CARLA simulator [6] (Fig. 1). SynFlow adopts a motion-oriented generation strategy: rather than attempting to perfectly reproduce real-world sensor noise, we actively proceduralize traffic densities, aggressive speed regimes, and complex topological interactions across nine maps. Leveraging this pipeline, we release SynFlow-4k, a massive synthetic dataset comprising 4,000 fully annotated sequences

(∼940k frames). This is a 34× scale-up over the annotated frames of nuScenes [2] and 46× those of TruckScenes [7], covering diverse road topologies, including roundabouts, intersections, and highways.

Through extensive experiments, we demonstrate that the model trained on SynFlow-4k generalizes zero-shot to different real-world sensors, comparable to in-domain supervised performance on nuScenes and beating the best in-domain results on TruckScenes by 31.8%. As a pre-training foundation, fine-tuning on just 5% of real labels already exceeds the performance of baselines trained from scratch on a budget four times larger. Our dataset is available at https://huggingface.co/datasets/KTH/SynFlow. Our primary contributions are as follows:

- We propose **SynFlow**, the first synthesis pipeline designed specifically for LiDAR scene flow. It shifts the focus from sensor-specific realism to geometric and temporal interaction complexity to address the lack of dense motion supervision.
- We introduce **SynFlow-4k**, a large-scale synthetic dataset comprising 4,000 sequences (∼940k frames) with dense, noise-free scene flow labels—representing a 34× scale-up over existing real-world annotated resources.
- We demonstrate that *SynFlow-4k* provides a robust, transferable motion prior: (1) models trained on it achieve strong zero-shot generalization across diverse real-world sensors; (2) it serves as a label-efficient pre-training foundation, significantly reducing real-world annotation demand; and (3) it complements real-world data by covering long-tail interactions that are rare in practice.

2 Related Work

Synthetic Data as Scalable 3D Supervision. Synthetic data has increasingly been explored as a primary source of supervision for 3D learning [22, 26, 30, 33, 34, 38]. Early image-based scene flow datasets such as FlyingThings3D [22] demonstrated that dense motion fields can be learned from fully rendered 3D geometry. More recently, works such as LRM-Zero [38] and MegaSynth [11] show that large-scale synthetic data alone can learn transferable geometric priors and exhibit clear scaling behavior. These results suggest photorealistic real-world data may not be necessary for robust 3D representation learning, and suggest that controllable synthetic environments can serve as a scalable supervision source.

Simulation in Autonomous Driving. In autonomous driving, CARLA [6] have been widely used to generate labeled data for perception tasks [3, 12, 37] including detection, segmentation, and domain adaptation. Datasets such as SHIFT [28] and CarlaScenes [15] emphasize environmental diversity and cross-domain robustness, while reconstruction-simulation paradigms (e.g., ReSimAD [40]) aim to bridge sensor gaps across domains. These efforts highlight the controllability and scalability of simulation for static perception and domain transfer. However, they focus on semantic realism or sensor alignment and rarely explore the kinematic complexity required for robust, zero-shot motion transfer to real-world environments.

LiDAR Scene Flow and the Supervision Bottleneck. LiDAR scene flow [17–20, 47] estimates point-wise 3D motion between consecutive point clouds and serves as a

geometry-centric representation of dynamic environments. Due to the difficulty of obtaining dense real-world motion labels, most existing methods rely heavily on self-supervised objectives derived from geometric consistency [42], cycle constraints [32], or rigidity assumptions [8, 45]. While effective, such proxy self-supervision is inherently under-constrained and sensitive to occlusion, sparsity, and non-rigid motion. Scaling unlabeled real-world data alone does not fully close the gap to fully supervised performance.

While image-based synthetic datasets provide dense pixel-level motion annotations, prior efforts largely emphasize visual realism or domain alignment rather than motion supervision itself. We argue that 3D motion-centric tasks exhibit a different sim-to-real behavior compared to appearance-dominated tasks: since scene flow learning primarily depends on physically consistent object kinematics rather than texture or semantics, synthetic environments with accurate rigid-body states can provide transferable supervision signals for real-world LiDAR. Consequently, the key question is not whether simulation matches visual appearance, but whether scalable synthetic motion can serve as a reliable supervision source for LiDAR scene flow. Our work addresses this question by introducing a motion-oriented synthetic LiDAR data engine and systematically analyzing synthetic scaling and real-world fine-tuning behavior.

3 Task and Preliminary

Scene Flow Definition. LiDAR scene flow aims to estimate the dense 3D motion field between consecutive point clouds in dynamic environments. Given two sequential scans, a source $\mathcal{P}_t = \{\mathbf{p}_i\}_{i=1}^{N_t} \subset \mathbb{R}^3$ and a target $\mathcal{P}_{t+1}$, the objective is to predict a per-point displacement field $\mathcal{F}_t = \{\mathbf{f}_i\}_{i=1}^{N_t}$. Each vector $\mathbf{f}_i \in \mathbb{R}^3$ represents the 3D translation of $\mathbf{p}_i$ from time t to $t+1$ in continuous space.

Temporal Context. While the primary target is the forward flow from $\mathcal{P}_t$ to $\mathcal{P}_{t+1}$, modern estimators often leverage a temporal window of h past frames $\{\mathcal{P}_{t-h}, \ldots, \mathcal{P}_t, \mathcal{P}_{t+1}\}$ to improve motion reasoning. We denote a feed-forward scene flow estimator as Φ_θ to learn the mapping:

$$\Phi_\theta : \{\mathbf{T}_{\text{ego}}^{t-h\to t+1}\mathcal{P}_{t-h}, \ldots, \mathbf{T}_{\text{ego}}^{t\to t+1}\mathcal{P}_t, \mathcal{P}_{t+1}\} \to \mathcal{F}_t, \tag{1}$$

where $\mathbf{T}_{\text{ego}}^{t'\to t+1} \in \mathbb{R}^{4\times 4}$ is the odometry transformation matrix, aligning all historical point clouds into the coordinate frame of the target scan $\mathcal{P}_{t+1}$.

Backbone Architecture. In our experiments, we instantiate Φ_θ with the ΔFlow backbone [46] as default. Following multi-frame designs, the backbone voxelizes each scan into sparse 3D features, aggregates temporal context into a compact representation, and applies a 3D sparse convolutional network (e.g., MinkUNet [5]) to extract motion-aware features. Finally, voxel features are interpolated back to points in $\mathcal{P}_t$ and decoded into the forward scene flow $\mathcal{F}_t$.

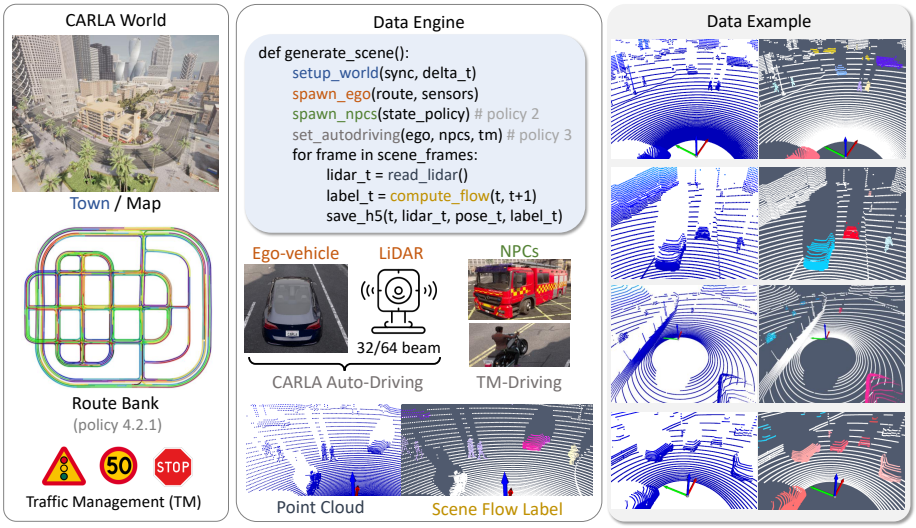

Fig. 2: Overview of our SynFlow pipeline and dataset examples. Left: A CARLA world provides diverse road topologies; we construct a route bank using topology-aware coverage to ensure broad spatial exploration and execute rollouts under Traffic Management (TM). Middle: our procedural data engine instantiates an ego vehicle with configurable LiDAR, spawns surrounding agents with controllable policies, and runs synchronized simulation steps; Right: Representative frames showing raw LiDAR point clouds alongside corresponding dense scene flow labels, spanning diverse traffic interactions and motion regimes.

4 Synthesizing the SynFlow Dataset

4.1 Overview

We introduce **SynFlow**, a synthesis pipeline designed to generate large-scale, kinematically diverse LiDAR datasets. Our design is *motion-oriented*: rather than targeting visual photorealism, we prioritize the geometric and temporal complexity of multi-agent interactions. The generation process is guided by three core policies: (1) Topological Discretization Policy: To ensure the model learns diverse road geometries (e.g., roundabouts vs. highways); (2) Speed-Regime Coverage Policy: To broaden the displacement magnitude support by including highway-structured towns/routes and leveraging road-type-dependent speed limits under Traffic Manager control; (3) Multi-Agent Interaction Policy: To enrich relative motion patterns by varying traffic density and agent behaviors, encouraging interaction-heavy scenarios beyond near-linear motion. In this work, we instantiate these policies using the CARLA simulator [6], which provides the high-fidelity physics and sensor models required to execute our kinematic rollouts.

4.2 Data Generation Pipeline

The generation of SynFlow is an automated, iterative process. We initiate the pipeline by loading a target town topology and a predefined sensor configuration. For each town,

the generation follows a sequence of "Sampling → Rollout → Export", governed by our three core policies.

1. Topological Discretization Policy (Search Space Initialization). The pipeline begins by discretizing the town's drivable topology into fine-grained lane segments $(\texttt{road\_id}, \texttt{section\_id}, \texttt{lane\_id})$. We then initialize a *route bank* $\mathcal{C}$ using a greedy search. A candidate route $\mathcal{R}$ is accepted only if it covers previously unvisited segments $|\mathcal{R} \setminus \mathcal{C}| > \tau$. This policy defines the "where" of our synthesis, ensuring geometric diversity by forcing the simulator to utilize long-tail road structures like Town04's highway loops and Town06's junctions.

2. Speed-Regime Coverage Policy (State Initialization). After selecting a route, we initialize the scene by spawning the ego vehicle and surrounding agents (NPCs) on valid waypoints along the route neighborhood. Rather than hand-tuning a target speed, we leverage the road-type-dependent speed limits and CARLA Traffic Manager control to induce diverse velocity regimes. Importantly, we include towns/routes containing highway-like segments (e.g., loop highways and long multi-lane roads), which naturally produce high-speed motion tails, complementing low-speed stop-and-go urban traffic. This policy defines the "how fast" of our synthesis and improves coverage of large displacements without requiring expensive per-scene parameter search.

3. Multi-Agent Interaction Policy (Dynamic Execution). During rollouts, we induce interaction-heavy scenes by varying local traffic density and agent behaviors under CARLA Traffic Manager control. This encourages diverse interaction regimes (e.g., merges, overtakes, and braking events) and yields complex non-linear relative motions that are essential for learning robust scene flow.

Simulation rollout and export. Given the route, kinematic states, and interaction regimes sampled by the policies above, we execute data collection in CARLA. For each town and LiDAR configuration, we initialize CARLA in a deterministic synchronous mode with a fixed simulation step $\Delta t = 0.1$, spawn the ego vehicle at the sampled route start, and attach a 32/64-beam LiDAR. We populate the scene with heterogeneous NPCs (different vehicles and pedestrians) within a local neighborhood and control them via CARLA Traffic Manager. To avoid low-motion segments caused by long stops or traffic deadlocks, we apply a lightweight deadlock-resolution rule: if the ego remains stationary beyond a short threshold due to a blocked intersection, we override the local traffic signal to release the blockage and resume the rollout, thereby maintaining a high ratio of dynamic frames. At every step, we compute ground truth scene flow supervision (Sec. 4.3) and store the rollout in HDF5 containers [29]. Each timestamp entry contains the LiDAR point cloud $\mathcal{P}_t$, ego pose, aligned flow labels $\mathcal{F}_t$, validity masks, and per-point instance metadata for training and evaluation.

4.3 Scene Flow Label Generation

After generating the dynamic scenarios, we leverage the simulator's ground-truth physical states to derive noiseless, per-point scene flow labels for all dynamic objects in the scene.

Simulator-provided information. At every timestep t, the simulator exposes the full world-coordinate rigid-body pose $\mathbf{T}_k^t \in SE(3)$ for each tracked agent k, including vehicles, pedestrians, and cyclists. In addition, the simulator provides each raw point $\mathbf{p}_i \in \mathcal{P}_t$ with a per-point instance identifier u_i that is consistent across timesteps. This allows us to associate every LiDAR point with a specific physical object and derive where that point moves between t and $t+1$.

Point-to-agent tag assignment. Since the per-point instance tag u_i is not aligned with the simulator actor ID k, we resolve the correspondence by majority voting over tags within the bounding box of agent k at time t:

$$\hat{u}_k^t = \operatorname*{argmax}_u \sum_{i:\,\mathbf{p}_i \in \mathrm{bbox}(k,t)} \mathbf{1}[u_i = u], \qquad \mathcal{M}_k^t = \{\, \mathbf{p}_i \in \mathcal{P}_t \mid u_i = \hat{u}_k^t \,\}. \quad (2)$$

where $\mathcal{M}_k^t$ is the set of LiDAR points in $\mathcal{P}_t$ that belong to agent k.

Rigid-body flow derivation. For each agent k, the simulator provides its world-coordinate pose at both timesteps, $\mathbf{T}_k^t$ and $\mathbf{T}_k^{t+1}$. The LiDAR scan $\mathcal{P}_t$ gives us the observed point positions at time t, but the corresponding positions at $t+1$ are never directly observed as they must be inferred. For any point $\mathbf{p}_i \in \mathcal{M}_k^t$, we exploit the known rigid-body motion of agent k to estimate where that surface element moves:

$$\mathbf{p}_i^\star = \mathbf{T}_k^{t+1}(\mathbf{T}_k^t)^{-1}\mathbf{p}_i. \quad (3)$$

Here $(\mathbf{T}_k^t)^{-1}$ maps $\mathbf{p}_i$ into the agent's local body frame, and $\mathbf{T}_k^{t+1}$ places it back into world space at the agent's pose at $t+1$. The ground truth flow vector is then $\mathbf{f}_i = \mathbf{p}_i^\star - \mathbf{p}_i$.

5 Training Scene Flow Estimation

In this section, we discuss the utilization of our synthesized SynFlow dataset to train a flow estimation model and define the protocols for evaluating its quality. We first describe the preparation and scaling of the SynFlow dataset, followed by our two-stage training strategy for real-world adaptation and the technical implementation details.

5.1 SynFlow Dataset Preparation

Following the procedure described in Sec. 4.2, we generate the SynFlow dataset consisting of 4k fully annotated LiDAR sequences (SynFlow-4k), totaling 939,083 frames. Data collection is performed on a desktop system equipped with an Intel i7-12700KF processor and a NVIDIA RTX 3090 GPU. Generating one sequence takes 3–6 minutes depending on the scene configuration. To study how synthetic supervision scales, we construct four training splits (1k, 2k, 3k, and 4k sequences) while controlling for map and sensor factors.

Each split aggregates complete rollouts across multiple towns and route banks, ensuring broad topological coverage. To maintain balanced map composition across scales, we subsample routes from Town12 (the largest route bank) and include them

Table 1: Summary of SynFlow-4k data splits. Scaling annotated volume via rollouts across CARLA towns and route banks. Training sets comprise mixed 32/64-beam LiDAR sequences.

Split	#Annotated Frames	Composition	Scenarios
1k	271,148	Town06, 07, 10	arterials, complex junctions, rural roads
2k	449,407	Town01–05	roundabouts, multi-lane intersections
3k	720,555	1k + 2k	(1k + 2k)
4k	939,083	2k + Town12	mixed zones (urban/suburban/rural)

only in the 4k split. All splits utilize a mixture of 32/64-beam LiDAR configurations to ensure sensor-agnostic motion learning. The exact composition and dominant scene characteristics of these splits are summarized in Tab. 1. Sample SynFlow-4k visualizations are in Sec 6.3, with more detailed video samples at `https://kin-zhang.github.io/SynFlow`.

5.2 Real-World Evaluation Strategy

To evaluate the quality of SynFlow-4k and its utility for real-world tasks, we define two primary evaluation regimes: *1) Zero-shot Generalization:* We evaluate models trained only on SynFlow-4k to measure how well they transfer to different real-world datasets without any domain adaptation; *2) Label-Efficient Fine-tuning:* To quantify the reduction in required real-world annotations, we use SynFlow-4k as a pre-training initialization. Models are first trained on our synthetic ground truth labels and then fine-tuned on restricted subsets (5–20%) of the target real-world benchmarks. By starting from the SynFlow-4k checkpoint and continuing supervised training only on limited real samples, we evaluate the model's ability to transfer a mature motion prior to data-scarce real-world environments.

5.3 Implementation Details

Backbone and Loss. We adopt the ΔFlow [46] backbone and follow its default architecture settings, using 5-frame LiDAR sequences voxelized at 0.15m within a 38.4m range. We train with the supervised scene flow loss introduced in ΔFlow: motion-awareness, category-balanced, and instance-consistency terms. For synthetic data, ground-truth is obtained via simulator states with the target construction described in Sec. 4.3.
Optimization and Augmentation. We train for 15 epochs using Adam with a learning rate of 0.002 and a total batch size of 20. We apply standard augmentations (e.g., z-height perturbation, random x-y flips) without additional domain adaptation.
Evaluation Datasets. We evaluate our method on three real-world LiDAR benchmarks. *nuScenes* [2] provides urban driving scenes captured with a 32-beam LiDAR; its training split contains 700 scenes totaling 137,575 frames. *TruckScenes* [7] focuses on long-haul highway driving with a dual 64-beam LiDAR platform; its training split contains 524 scenes and 101,902 frames. We choose nuScenes and TruckScenes as our primary benchmarks for label efficiency because their annotated ground-truth is available for

Table 2: Performance on the **nuScenes** validation set. *Supervision*: *unlab.* = self-supervised on unlabeled real sequences; *lab.* = supervised on available (20%) labeled real data; *synth.* = supervised on SynFlow synthetic data only (no real data). **Bold** denotes the best result in each column.

Methods	Supervision	Dynamic Bucket-Normalized ↓					Three-way EPE (cm) ↓			
		Mean	CAR	OTHER	PED.	VRU	Mean	FD	FS	BS
Ego Motion Flow	–	1.000	1.000	1.000	1.000	1.000	12.34	35.94	1.07	0.00
Self-supervised (100% unlabeled real-world data)										
SeFlow [45]	100% unlab.	0.544	0.396	0.635	0.726	0.419	8.19	16.15	3.97	4.45
VoteFlow [18]	100% unlab.	0.538	0.355	0.605	0.780	0.410	7.80	15.65	3.51	4.24
SeFlow++ [43]	100% unlab.	0.509	0.327	0.583	0.716	0.409	6.13	14.59	1.96	1.86
TeFlow [42]	100% unlab.	0.395	0.303	0.461	0.474	0.344	4.64	10.92	1.49	1.51
Fully supervised (20% labeled real-world data)										
DeFlow [44]	20% lab.	0.314	0.163	0.286	0.533	0.275	3.98	6.99	3.45	1.50
Flow4D [14]	20% lab.	0.279	0.204	0.312	0.379	0.222	3.82	8.05	1.82	1.58
ΔFlow [46]	20% lab.	0.216	0.138	0.219	0.327	0.181	2.33	4.83	1.37	0.79
SynFlow pre-training (ours, synthetic data only → optional fine-tune)										
SynFlow-4k	0% (synth. only)	0.242	0.177	0.307	0.300	0.183	2.60	6.06	1.28	0.46
SynFlow-4k + FT	20% lab.	**0.157**	**0.110**	**0.173**	**0.247**	**0.098**	**1.65**	**3.48**	**1.19**	**0.29**

only a small subset ($\sim$20%) of the full sequences. Finally, the *Aeva* [23] dataset provides FMCW LiDAR sequences covering both urban and highway driving; we evaluate on 67 sequences after performing a flow generation consistency check.

Evaluation Metrics. We follow prior work and report *Dynamic Bucket-Normalized EPE* [13] as our primary metric, which normalizes errors by motion magnitude across velocity buckets. We additionally report *Three-way EPE* [4] for completeness to facilitate comparison with existing methods.

Baselines. We compare against representative LiDAR scene flow baselines from prior works, including both self-supervised methods (SeFlow [45], VoteFlow [18], SeFlow++ [43], TeFlow [42]) and fully supervised feed-forward estimators (DeFlow [44], Flow4D [14], ΔFlow [46]). All methods are trained with the best configurations as reported in their papers in the in-domain real-world datasets.

6 Results

6.1 Baseline Comparison

Tab. 2 and Tab. 3 report the primary results on nuScenes and TruckScenes across three learning regimes: self-supervised learning on the full unlabeled training split, supervised training on the human-labeled data, and zero-shot transfer from synthetic training.

Zero-shot Transfer. Without observing any real-world data, SynFlow-4k achieves strong zero-shot generalization on both benchmarks. On nuScenes, our zero-shot prior achieves a Dynamic EPE of 0.242, outperforming the best self-supervised baseline (TeFlow, 0.395) by 38.7% and approaching the performance of supervised methods. On Truck--Scenes, it attains 0.274, outperforming TeFlow (0.425) by 35.5% and even surpassing

Table 3: Performance on the **TruckScenes** validation set. TruckScenes features large commercial vehicles with distinct LiDAR configurations unseen during SynFlow pre-training, providing a stringent test of cross-domain zero-shot generalization. Notation follows Tab. 2.

Methods	Supervision	Dynamic Bucket-Normalized ↓					Three-way EPE (cm) ↓			
		Mean	CAR	OTHER	PED.	VRU	Mean	FD	FS	BS
Ego Motion Flow	–	1.000	1.000	1.000	1.000	1.000	61.43	184.30	0.00	0.00
Self-supervised (100% unlabeled real-world data)										
SeFlow [45]	100% unlab.	0.681	0.494	0.752	0.886	0.591	37.41	103.97	1.56	6.68
VoteFlow [18]	100% unlab.	0.680	0.517	0.737	0.895	0.573	36.47	103.48	1.29	4.64
SeFlow++ [43]	100% unlab.	0.653	0.519	0.738	0.860	0.497	33.35	95.62	1.20	3.23
TeFlow [42]	100% unlab.	0.425	0.254	0.455	0.636	0.353	41.97	116.55	1.32	8.06
Fully supervised (20% labeled real-world data)										
DeFlow [44]	20% lab.	0.570	0.180	0.410	0.970	0.730	7.30	16.47	1.67	3.77
Flow4D [14]	20% lab.	0.456	0.176	0.351	0.885	0.413	16.14	44.87	1.71	1.85
ΔFlow [46]	20% lab.	0.402	0.196	0.400	0.690	0.323	7.28	16.26	1.36	4.52
SynFlow pre-training (ours, synthetic data only → optional fine-tune)										
SynFlow-4k	0% (synth. only)	0.274	0.109	0.220	0.467	0.300	25.25	74.16	1.02	0.58
SynFlow-4k + FT	20% lab.	**0.266**	**0.082**	**0.232**	**0.464**	**0.285**	**6.75**	**15.70**	**0.98**	**3.58**

the SOTA supervised ΔFlow baseline (0.402) by 31.8%. These results across different sensors suggest that physically consistent motion transfers well for LiDAR scene flow. Our motion-oriented synthetic data effectively bridges the sim-to-real gap without explicit domain adaptation.

Fine-tuning Transfer. As shown in Tab. 4, pre-training on SynFlow-4k yields a stronger motion prior, leading to more effective downstream fine-tuning. Fine-tuning on 20% of real labels achieves 0.157 on nuScenes, outperforming the supervised ΔFlow baseline (0.216) by 27.3%. On TruckScenes, the gain reaches 33.8% (0.266 vs. 0.402). These gains are achieved in the practical regime where dense annotation is expensive and only a fraction of frames can be labeled. Overall, the results indicate that SynFlow-4k provides a strong initialization that significantly reduces the demand for real-world labels, fine-tuning on just 5% of real labels already exceeds baselines trained from scratch on $4\times$ more data.

6.2 Ablation Studies

Zero-shot Scaling. To understand the effect of synthetic data scale on zero-shot performance, we evaluate training volumes ranging from 1k to 4k sequences (Fig. 3). Performance improves consistently across both benchmarks, with the most significant gains observed between 1k and 2k. Beyond this point, accuracy begins to stabilize, suggesting the core motion distribution for rigid-body kinematics is effectively captured within the 4k split. Among categories, *CAR* benefits most from additional data. Its rigid and predictable kinematics transfer well from simulation, reaching 0.109 on TruckScenes at 4k without any real labeled data. These results confirm that synthetic scale is a reliable lever for improving motion priors, particularly for vehicle-class objects.

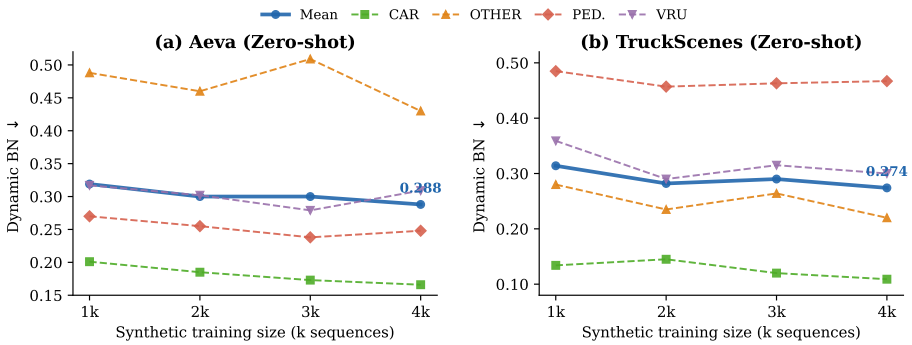

Fig. 3: Zero-shot scaling performance (1k–4k sequences). Evaluation on Aeva (a) and TruckScenes (b) using Dynamic Bucket-Normalized EPE (lower is better). Solid blue line indicates the overall mean; dashed lines represent per-category breakdowns.

Fine-tuning Scaling. To evaluate how real-world data scale affects adaptation, we ablate the labeling budget required for fine-tuning. As shown in Tab. 4, SynFlow-4k provides a high-fidelity initialization that significantly reduces label demand. On nuScenes, fine-tuning our prior with just 5% of real labels (0.201) already outperforms the supervised baseline trained from scratch on 20% of the data (0.216). Increasing the fine-tuning budget to 10% and 20% further reduces Dynamic Bucketed-Normalized EPE to 0.175 and 0.157, respectively. On TruckScenes, this advantage is even more pronounced: our zero-shot model already outperforms the 20% supervised baseline (0.274 vs. 0.402), and adding a 10% labeling budget further reduces error to 0.261. Across both datasets, the most rapid performance gains occur within the first 10% of real-world exposure, with further improvements scaling more gradually up to 20%. This demonstrates that SynFlow-4k serves as a strong pre-training foundation, significantly reducing the need for real-world labels while outperforming training from scratch.

Synthetic-Real Complementarity. To investigate how synthetic supervision can further push the limits of zero-shot transfer, we evaluate SynFlow-4k in combination with UniFlow [16]: a state-of-the-art model pre-trained on a massive union of real-world datasets (Argo-v2, nuScenes, and Waymo). This experiment tests whether synthetic data is merely a "low-cost substitute" for real labels or an orthogonal source of knowledge (Tab. 5). We find that combining both sources (0.263) consistently outperforms either individual model, yielding a striking 37% improvement in the PED. category (0.398 $\rightarrow$ 0.251). This large gain suggests that even mega-scale real datasets lack the kinematic density needed for small, dynamic agents. While UniFlow captures real sensor characteristics (e.g., beam layouts, range-dependent sparsity, and measurement noise), SynFlow-4k contributes dense, oracle kinematics for "long-tail" interactions that are rarely seen in real-world logs. These results demonstrate that synthetic and real-world pre-training are highly complementary: simulation provides the fundamental motion "rules", while real data provides the sensor "realism".

Table 4: Fine-tuning scaling on nuScenes (upper) and TruckScenes (lower). Comparison between from-scratch supervised baselines and SynFlow-4k across varying real-world label budgets. Zero-shot (0%) denotes purely synthetic supervision; +FT rows indicate fine-tuning on the specified real-world fraction. **Bold** denotes best per column.

Methods	Real labels	Mean	CAR	OTHER	PED.	VRU
nuScenes validation set						
ΔFlow [46]	20%	0.216	0.138	0.219	0.327	0.181
SynFlow-4k (Zero-shot)	0%	0.242	0.177	0.307	0.300	0.183
SynFlow-4k + FT	5%	0.201	0.151	0.240	0.272	0.139
SynFlow-4k + FT	10%	0.175	0.127	0.201	0.253	0.119
SynFlow-4k + FT	20%	**0.157**	**0.110**	**0.173**	**0.247**	**0.098**
TruckScenes validation set						
ΔFlow [46]	20%	0.402	0.196	0.400	0.690	0.323
SynFlow-4k (Zero-shot)	0%	0.274	0.109	0.220	0.467	0.300
SynFlow-4k + FT	5%	0.271	0.101	0.214	0.471	0.299
SynFlow-4k + FT	10%	**0.261**	0.088	**0.215**	0.478	**0.262**
SynFlow-4k + FT	20%	0.266	**0.082**	0.232	**0.464**	0.285

Table 5: Complementarity of SynFlow-4k and UniFlow. Zero-shot evaluation on the Aeva dataset. UniFlow [16] is trained on merged real-world data (Argo-v2 [36], nuScenes, Waymo). Combining synthetic and real-world sources consistently outperforms individual training.

Training data	Dynamic Bucket-Normalized ↓					Three-way EPE (cm) ↓			
	Mean	CAR	OTHER	PED.	VRU	Mean	FD	FS	BS
UniFlow only	0.303	0.112	**0.359**	0.398	0.344	6.65	13.17	5.16	1.62
SynFlow-4k only	0.288	0.166	0.430	**0.248**	0.309	9.08	21.41	4.88	0.94
Both	**0.263**	**0.102**	0.401	0.251	**0.298**	**5.95**	**13.15**	**3.85**	**0.84**

Synthetic Generation Strategy. To investigate the architectural design of *SynFlow* and identify the key factors driving its strong generalization, we isolate our data generation policies using a 1k-sequence ablation split (Tab. 6). We find that our Topological Discretization Policy (Policy 1), represented by Route Coverage in the table, provides the largest individual gain (0.346 $\rightarrow$ 0.330). By replacing random route sampling with a greedy search over unvisited lane segments, the model encounters a more diverse set of road structures, leading to more robust motion contexts. Furthermore, including Highway towns to support our Speed Regime Policy (Policy 2) provides a complementary benefit (0.330 $\rightarrow$ 0.319). Its impact is most visible in the Three-way EPE metrics, where it significantly reduces EPE Foreground Dynamic (FD) error (50.21 $\rightarrow$ 48.17,cm). This confirms that exposing the model to the high-speed displacements found in highway loops is essential for generalizing to the high-velocity tails of real-world datasets. The combination of both policies ensures that SynFlow dataset covers both long-tail road structures and a broad support of motion magnitudes.

Backbone Agnosticism. To verify that the benefits of SynFlow dataset are not architecture-specific, we evaluate zero-shot performance across multiple scene flow backbones on

Table 6: Ablation of data generation design choices (1k sequences trained), zero-shot evaluated on Aeva dataset. "Topology Coverage (Policy 1)" toggles greedy lane-segment coverage route bank; otherwise, random-start fixed-length routes are used. "Speed Regime (Policy 2)" toggles whether highway-structured towns (Town04, Town06, Town07) are included; otherwise only urban towns are used.

P1:	P2:	Dynamic Bucket-Normalized ↓					Three-way EPE (cm) ↓			
Topology	Speed	Mean	CAR	OTHER	PED.	VRU	Mean	FD	FS	BS
		0.346	0.225	0.512	0.311	0.334	18.96	52.34	3.51	1.02
	✓	0.331	0.211	0.487	0.282	0.342	18.45	51.25	3.10	1.01
✓		0.330	0.215	0.501	0.281	0.324	17.89	50.21	2.82	**0.65**
✓	✓	**0.319**	**0.201**	**0.488**	**0.270**	**0.317**	**17.11**	**48.17**	**2.52**	0.66

Table 7: Ablation of backbone architectures. Zero-shot evaluation on Aeva after training on SynFlow-4k dataset. The consistent improvement across different estimators demonstrates the backbone-agnostic transferability of our synthetic supervision.

Backbone	Dynamic Bucket-Normalized ↓					Three-way EPE (cm) ↓			
	MEAN	CAR	OTHER	PED.	VRU	Mean	FD	FS	BS
Ego Motion Flow	1.000	1.000	1.000	1.000	1.000	44.31	132.00	0.92	0.00
Trained on SynFlow-4k									
DeFlow	0.426	0.249	0.514	0.448	0.500	10.99	25.85	6.60	0.84
Flow4D	0.308	0.173	0.452	0.286	0.320	9.37	21.87	6.00	0.94
ΔFlow	0.288	0.166	0.430	0.248	0.309	9.08	21.41	4.88	0.94

the Aeva dataset (Tab. 7). As an initial reference, an Ego Motion Flow baseline (assigning flow via odometry only) results in maximum Dynamic Normalized-Bucket EPE (1.000), demonstrating that ego-compensation alone cannot account for the dynamic scene elements. In contrast, after pre-training on SynFlow-4k, all learned backbones, including feed-forward estimators like DeFlow [44] and Flow4D [14], as well as our default ΔFlow, reduce both Dynamic Bucket-Normalized error and Three-way EPE. The performance gains across these diverse architectures indicate that SynFlow dataset provides a transferable and backbone-agnostic motion prior, validating its utility as a general-purpose supervisory source for LiDAR scene flow.

6.3 Qualitative Comparison

Fig. 4 visualizes representative samples from SynFlow-4k and real-world datasets. While real-world datasets provide essential sensor realism (patterns and noise), they are often tailored to specific operational domains: nuScenes focuses on urban driving with a 32-beam configuration, while TruckScenes targets highway environments with commercial vehicles. SynFlow-4k complements these specialized distributions by incorporating a variety of road topologies, including roundabouts and merging zones, across both 32 and 64-beam sensor profiles. It intentionally increases the density and diversity of dynamic agents, particularly pedestrians and small movers in sidewalks and crossing areas. Our pipeline also provides dense, point-wise supervision for all dynamic

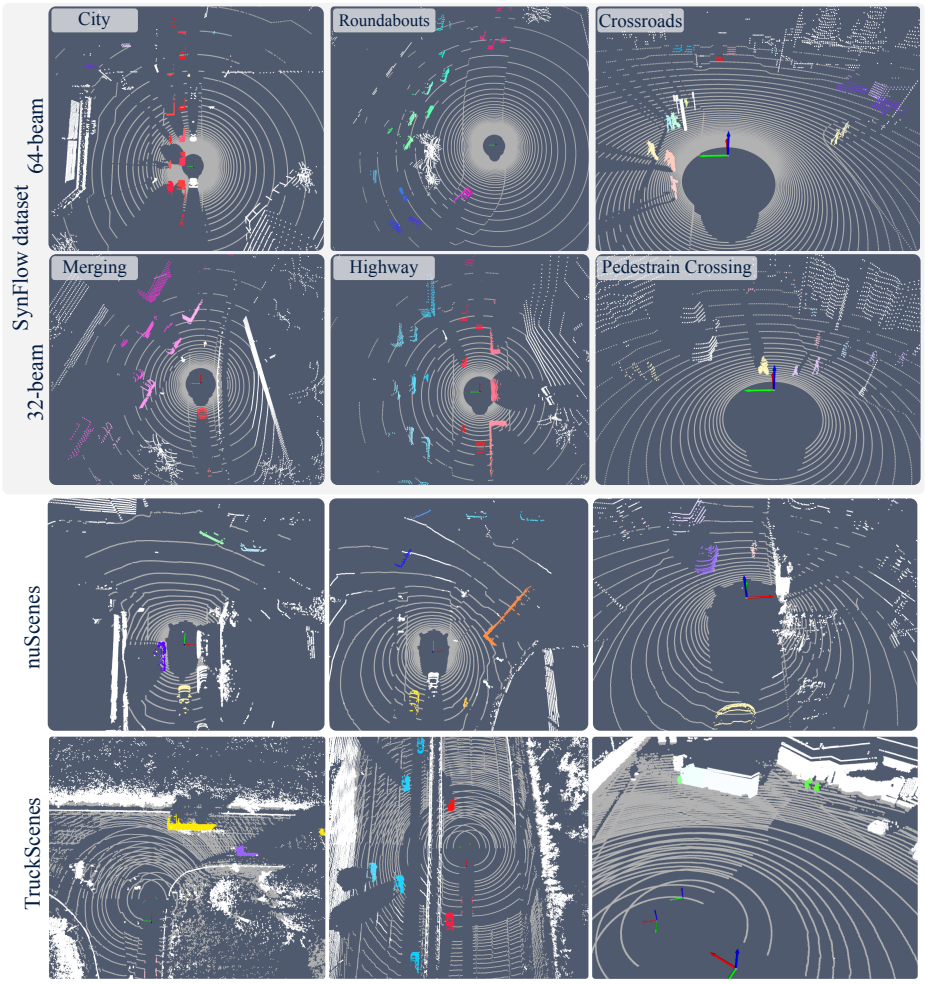

Fig. 4: Dataset visualization. Top: SynFlow-4k samples under 64-beam (row 1) and 32-beam (row 2) configurations, spanning city, roundabout, highway, and merging scenarios. Per-point scene flow labels are rendered as colored vectors. Direction is encoded as hue, and magnitude as saturation. Bottom: representative samples from real-world datasets (nuScenes and TruckScene).

agents simultaneously, capturing kinematic details that can be difficult to annotate in real-world logs. Together, these properties allow the SynFlow dataset to serve as a comprehensive motion prior that complements the scene-specific coverage of individual real-world datasets.

6.4 Limitations and Future Works

From Open-loop to Feedback-driven Synthesis. Our current pipeline follows a "generate-then-train" paradigm: data is synthesized based on predefined policies and then used

for training. This open-loop process does not dynamically adapt to the model's learning state. Future work could explore a *closed-loop* or *cascade learning* framework, where the model's failure cases (e.g., specific occlusion patterns or rare motion speeds) act as feedback signals to trigger targeted re-simulation. By actively generating "hard examples" adversarial to the current model, the pipeline could achieve higher data efficiency and continuous improvement.

Expanding Domains. While the presented methods and experiments focus on autonomous driving, the motion-oriented synthesis principle underlying SynFlow can generalize naturally to other domains where 3D motion labels are scarce. A natural extension is to apply this methodology to other simulation environments (e.g., Isaac Sim [24]) for embodied robotics [9, 41], covering indoor navigation, tabletop manipulation, and human-robot interaction scenarios.

7 Conclusion

In this work, we introduced SynFlow, a motion-oriented generation pipeline, and its resulting large-scale dataset, SynFlow-4k, to address the critical bottleneck of dense annotation in LiDAR scene flow. Rather than chasing perfect sensor realism, our approach prioritizes geometric and temporal multi-agent complexity, showing that physically consistent motion relations are highly transferable across domains.

Through extensive evaluations, SynFlow-4k is a robust, backbone-agnostic pretraining source. In a zero-shot regime, models trained on it generalize across diverse benchmarks, comparable to in-domain supervised performance. Fine-tuned on 5% of real labels, it surpasses supervised baselines trained from scratch on four times the budget. As a complement to real-world data, it further supplies kinematic density for long-tail interactions that real-world logs lack. SynFlow provides a scalable, label-efficient data engine for generalizable dynamic 3D scene understanding. We hope that releasing SynFlow and its extensible data engine will facilitate further research on generalizable 3D motion estimation and accelerate progress toward reliable dynamic perception in diverse real-world environments.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations and data handling were enabled by the Berzelius supercomputing resource provided by the National Supercomputer Centre at Linköping University and the Knut and Alice Wallenberg Foundation, Sweden, as well as by resources provided by Chalmers e-Commons at Chalmers and the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

1. Alibeigi, M., Ljungbergh, W., Tonderski, A., Hess, G., Lilja, A., Lindstrom, C., Motorniuk, D., Fu, J., Widahl, J., Petersson, C.: Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
3. Cai, X., Jiang, W., Xu, R., Zhao, W., Ma, J., Liu, S., Li, Y.: Analyzing infrastructure lidar placement with realistic lidar simulation library. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 5581–5587. IEEE (2023)
4. Chodosh, N., Ramanan, D., Lucey, S.: Re-evaluating lidar scene flow. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 5993–6003 (2024). https://doi.org/10.1109/WACV57701.2024.00590
5. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
6. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
7. Fent, F., Kuttenreich, F., Ruch, F., Rizwin, F., Juergens, S., Lechermann, L., Nissler, C., Perl, A., Voll, U., Yan, M., et al.: Man truckscenes: A multimodal dataset for autonomous trucking in diverse conditions. Advances in Neural Information Processing Systems **37**, 62062–62082 (2024)
8. Hoffmann, D.T., Raza, S.H., Jiang, H., Tananaev, D., Klingenhoefer, S., Meinke, M.: Floxels: Fast unsupervised voxel based scene flow estimation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22328–22337 (2025)
9. Huang, W., Chao, Y.W., Mousavian, A., Liu, M.Y., Fox, D., Mo, K., Fei-Fei, L.: Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20765–20779 (June 2026)
10. Jiang, C., Wang, G., Liu, J., Wang, H., Ma, Z., Liu, Z., Liang, Z., Shan, Y., Du, D.: 3dsflabelling: Boosting 3d scene flow estimation by pseudo auto-labelling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
11. Jiang, H., Xu, Z., Xie, D., Chen, Z., Jin, H., Luan, F., Shu, Z., Zhang, K., Bi, S., Sun, X., et al.: Megasynth: Scaling up 3d scene reconstruction with synthesized data. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16441–16452 (2025)
12. Jiang, W., Xiang, H., Cai, X., Xu, R., Ma, J., Li, Y., Lee, G.H., Liu, S.: Optimizing the placement of roadside lidars for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18381–18390 (2023)
13. Khatri, I., Vedder, K., Peri, N., Ramanan, D., Hays, J.: I can't believe it's not scene flow! In: European Conference on Computer Vision. pp. 242–257. Springer (2024)
14. Kim, J., Woo, J., Shin, U., Oh, J., Im, S.: Flow4D: Leveraging 4d voxel network for lidar scene flow estimation. IEEE Robotics and Automation Letters pp. 1–8 (2025). https://doi.org/10.1109/LRA.2025.3542327
15. Kloukiniotis, A., Papandreou, A., Anagnostopoulos, C., Lalos, A., Kapsalas, P., Nguyen, D.V., Moustakas, K.: Carlascenes: A synthetic dataset for odometry in autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4520–4528 (2022)

16. Li, S., Zhang, Q., Khatri, I., Vedder, K., Ramanan, D., Peri, N.: UniFlow: Towards zero-shot lidar scene flow for autonomous vehicles via cross-domain generalization. arXiv preprint arXiv:2511.18254 (2025)
17. Lin, Y., Caesar, H.: ICP-Flow: Lidar scene flow estimation with icp. In: CVPR (2024)
18. Lin, Y., Wang, S., Nan, L., Kooij, J., Caesar, H.: Voteflow: Enforcing local rigidity in self-supervised scene flow. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 17155–17164 (2025)
19. Liu, J., Wang, G., Ye, W., Jiang, C., Han, J., Liu, Z., Zhang, G., Du, D., Wang, H.: Difflow3d: Toward robust uncertainty-aware scene flow estimation with iterative diffusion-based refinement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15109–15119 (2024)
20. Luo, J., Cheng, J., Zhang, Q., Xue, B., Fan, R., Tang, X.: MambaFlow: A novel and flow-guided state space model for scene flow estimation. IEEE Transactions on Intelligent Vehicles **11**(4), 511–521 (2026). https://doi.org/10.1109/TIV.2026.3663171
21. Mao, J., Niu, M., Jiang, C., Liang, H., Chen, J., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., Yu, J., Xu, C., Xu, H.: One million scenes for autonomous driving: Once dataset. In: NeurIPS Datasets and Benchmarks (2021)
22. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016)
23. Narasimhan, G.N., Vhavle, H., Vishvanatha, K.B., Reuther, J.: Aevascenes: A dataset and benchmark for fmcw lidar perception (2025), https://scenes.aeva.com/, accessed: 2026-06-28
24. NVIDIA: Isaac Sim (2024), https://github.com/isaac-sim/IsaacSim, accessed: 2026-06-28
25. Pang, Z., Li, Z., Wang, N.: Simpletrack: Understanding and rethinking 3d multi-object tracking. In: European conference on computer vision. pp. 680–696. Springer (2022)
26. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: CVPR (2025)
27. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
28. Sun, T., Segu, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., Yu, F.: Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21371–21382 (2022)
29. The HDF Group: Hierarchical Data Format, version 5, https://github.com/HDFGroup/hdf5, accessed: 2026-06-28
30. Van Hoorick, B., Wu, R., Ozguroglu, E., Sargent, K., Liu, R., Tokmakov, P., Dave, A., Zheng, C., Vondrick, C.: Generative camera dolly: Extreme monocular dynamic novel view synthesis. In: ECCV (2024)
31. Vedder, K., Peri, N., Chodosh, N., Khatri, I., Eaton, E., Jayaraman, D., Ramanan, Y.L.D., Hays, J.: ZeroFlow: Fast Zero Label Scene Flow via Distillation. International Conference on Learning Representations (ICLR) (2024)
32. Vedder, K., Peri, N., Khatri, I., Li, S., Eaton, E., Kocamaz, M.K., Wang, Y., Yu, Z., Ramanan, D., Pehserl, J.: Neural eulerian scene flow fields. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=0CieWy9ONY

33. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotný, D.: VGGT: visual geometry grounded transformer. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5294–5306 (2025)
34. Wang, J., Chen, M., Zhang, S., Karaev, N., Schönberger, J., Labatut, P., Bojanowski, P., Novotny, D., Vedaldi, A., Rupprecht, C.: VGGT-Ω. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
35. Wang, Z., Chen, F., Lertniphonphan, K., Chen, S., Bao, J., Zheng, P., Zhang, J., Huang, K., Zhang, T.: Technical report for argoverse challenges on unified sensor-based detection, tracking, and forecasting. arXiv preprint arXiv:2311.15615 (2023)
36. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021) (2021)
37. Wilson, J., Song, J., Fu, Y., Zhang, A., Capodieci, A., Jayakumar, P., Barton, K., Ghaffari, M.: Motionsc: Data set and network for real-time semantic mapping in dynamic environments. IEEE Robotics and Automation Letters **7**(3), 8439–8446 (2022)
38. Xie, D., Bi, S., Shu, Z., Zhang, K., Xu, Z., Zhou, Y., Pirk, S., Kaufman, A., Sun, X., Tan, H.: Lrm-zero: Training large reconstruction models with synthesized data. Advances in Neural Information Processing Systems **37**, 53285–53316 (2024)
39. Yang, D., Cai, X., Liu, Z., Jiang, W., Zhang, B., Yan, G., Gao, X., Liu, S., Shi, B.: Realistic rainy weather simulation for lidars in carla simulator. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 951–957 (2024)
40. Zhang, B., Cai, X., Yuan, J., Yang, D., Guo, J., Yan, X., Xia, R., Shi, B., Dou, M., Chen, T., Liu, S., Yan, J., Qiao, Y.: ResimAD: Zero-shot 3d domain transfer for autonomous driving with source reconstruction and target simulation. In: The Twelfth International Conference on Learning Representations (2024)
41. Zhang, J., Zheng, C., Yang, Y., Argus, M., Soraki, R., Han, W., Anderson, T., Li, C.L., Liu, S., Duan, J., Ren, Z., Zhang, J., Krishna, R.: Molmomotion: Forecasting point trajectories in 3d with language instruction. arXiv preprint arXiv:2606.18558 (2026)
42. Zhang, Q., Jiang, C., Zhu, X., Miao, Y., Zhang, Y., Andersson, O., Jensfelt, P.: TeFlow: Enabling multi-frame supervision for self-supervised feed-forward scene flow estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3667–3676 (2026)
43. Zhang, Q., Khoche, A., Yang, Y., Ling, L., Mansouri, S.S., Andersson, O., Jensfelt, P.: HiMo: High-speed objects motion compensation in point cloud. IEEE Transactions on Robotics **41**, 5896–5911 (2025). https://doi.org/10.1109/TRO.2025.3619042
44. Zhang, Q., Yang, Y., Fang, H., Geng, R., Jensfelt, P.: DeFlow: Decoder of scene flow network in autonomous driving. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 2105–2111 (2024). https://doi.org/10.1109/ICRA57147.2024.10610278
45. Zhang, Q., Yang, Y., Li, P., Andersson, O., Jensfelt, P.: SeFlow: A self-supervised scene flow method in autonomous driving. In: European Conference on Computer Vision (ECCV). p. 353–369. Springer (2024). https://doi.org/10.1007/978-3-031-73232-4_20
46. Zhang, Q., Zhu, X., Zhang, Y., Cai, Y., Andersson, O., Jensfelt, P.: DeltaFlow: An efficient multi-frame scene flow estimation method. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
47. Zhang, Y., Edstedt, J., Wandt, B., Forssén, P.E., Magnusson, M., Felsberg, M.: GMSF: Global matching scene flow. Advances in Neural Information Processing Systems **36** (2024)