MESSENGER: Memory-Enhanced Sequential Scene Flow Estimation via Autoregressive Next-Frame ForecastinG

Jiuming Liu[1], Jianing Li[2], Mengmeng Liu[3], Hongyang He[4], Hesheng Wang[2], Per Ola Kristensson[1]
[1] University of Cambridge [2] Shanghai Jiao Tong University
[3] University of Twente [4] University of Warwick
liujiuming123@gmail.com pok21@cam.ac.uk

Abstract

Scene flow can capture low-level 3D motion displacements in dynamic scenarios. Early pairwise estimators relying on instantaneous two-frame motion lack long-term temporal correlation and also struggle with poor extrapolation ability in future prediction. Although some recent methods attempt to explore multi-frame scene flow estimation in a sequence-to-sequence manner, they typically suffer from heavy computational overhead with increasing input frames and long-horizon prediction degradation due to ineffective motion propagation. To address these problems, we propose a novel memory-enhanced sequential scene flow pipeline, called **MESSENGER**. To sufficiently mine long-term temporal dependencies naturally within consecutive sequences, a memory buffer is designed by explicitly storing multiple history flow estimates and latent states. For each input frame, the temporally stored flows and states are correlated and retrieved to predict the current initialized flow in a next-frame forecasting manner. Furthermore, we develop an uncertainty-aware reweighting module to filter unreliable retrievals and mitigate accumulated errors. Extensive experiments on nuScenes and Argoverse 2 demonstrate state-of-the-art performance of our MESSENGER, reducing EPE3D by **71.6%** on nuScenes and **67.7%** on Argoverse 2 in long-horizon future extrapolation. This superiority can be attributed to our designed autoregressive forecasting paradigm, which naturally forces the network to progressively learn the next-frame distribution based on history observations. Code will be released at `https://github.com/liujiuming123/Messenger`.

1 Introduction

Scene flow indicates the 3D dense displacement field across consecutive image or point cloud frames [1, 2, 3, 4, 5, 6], which is a fundamental task capturing low-level scene dynamics. Scene flow estimation can be applied in various downstream tasks such as dynamic reconstruction and generation [7, 8, 9, 10, 11], motion segmentation [12, 13, 14], and SLAM systems [15, 16, 17, 18].

With the rapid development of deep learning architectures, concurrent methods [2, 19, 20, 21] commonly leverage learning-based matching for scene flow estimation in a pairwise paradigm. As a pioneering method, FlowNet3D [1] adopts hierarchical feature extraction for iterative flow estimation and refinement. PWCNet [2] introduces the Pyramid-Warping-Cost Volume (PWC) structure [22] to estimate flows end-to-end. Subsequent approaches strengthen matching reliability [23, 24, 19], exert rigidity constraints [20], or capture robust probabilistic distributions with generative models [25, 26]. However, these pairwise-only methods are typically constrained by limited temporal receptive field

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

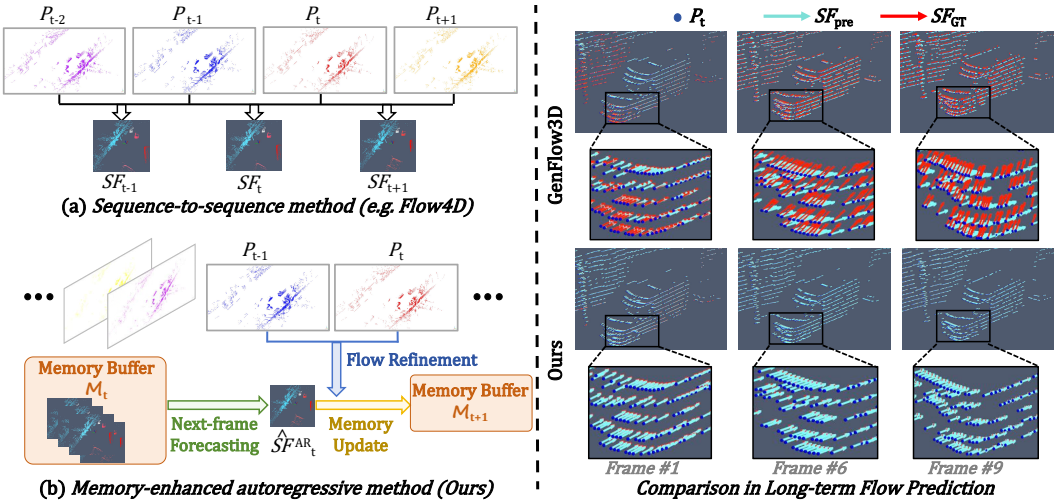

Figure 1: **Comparison with previous multi-frame scene flow methods.** (***Left***) Compared to sequence-to-sequence methods with heavy computational costs, our method adopt an autoregressive sequential estimation paradigm, which retrieves multiple history flow priors from a memory buffer. (***Right***) Compared to recent SOTA flow prediction method GenFlow3D [26] suffering from degraded accuracy in longer-horizon predictions, our uncertainty-aware memory reweighting can mitigate long-term compounding errors by measuring history-present motion discrepancy.

[27], neglecting long-term motion consistency intrinsically within LiDAR sequences, while also struggling with occlusions, motion boundaries, and large displacements [28, 29, 30].

To effectively unlock spatio-temporal dependencies in LiDAR sequences, some researchers recently focus on studying multi-frame scene flow estimation [31, 32, 27] as in Fig. 1. However, sequence-to-sequence methods [32, 31, 33] commonly struggle with inevitable increasing computational costs when consuming more frames because they stack multi-frame features in an additional temporal dimension [31, 32] or in the channel dimension [33] in Fig. 1 (***Left***) and Fig. 2. To address this problem, DeltaFlow [34] computes residual flows, rather than full flows, to maintain a constant feature size. Nevertheless, it still requires fixed-length input and also relies on ground truth rigid poses from odometry. Furthermore, previous methods rarely investigate future flow extrapolation potential beyond the observed history contexts. Unlike estimation-only approaches, recent work GenFlow3D [26] proposes a diffusion-based scene flow estimator that allows future scene flow prediction. However, it faces the challenge of degraded prediction performance along longer-interval extrapolations as in Fig. 1 (***Right***) due to temporally accumulated compounding errors and ineffective implicit motion propagation from the GRU module. Furthermore, the GPU memory usage further restrains its scalability to longer sequences as in Fig. 2.

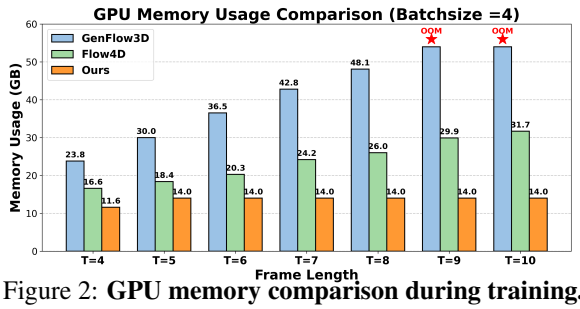

Figure 2: **GPU memory comparison during training.**

In this paper, we propose a novel **M**emory-**E**nhanced **S**equential **S**cene flow **E**stimation framework, via autoregressive **N**ext-frame forecastin**G**, called **MESSENGER**. To sufficiently mine temporal dependencies, a memory buffer is developed to dynamically store explicit coarse flows and implicit latent states from multiple nearest history estimates with minimal computational overhead. When a new frame from LiDAR sequences is delivered, an initial flow is extrapolated by purely correlating and retrieving history memories. Consequently, the forecasting paradigm explicitly enforces the model to predict future motions based on observations. However, the historical retrieval is not equally trustworthy for each point, leading to potential compounding errors during long-horizon predictions. Additionally, it is non-trivial for the forecasting-only flows to consider sudden motion variations and unpredictable dynamics. To alleviate these problems, we also design an uncertainty quantification module to indicate per-point reliability which is used to reweight the memory merging process. In this

way, outliers in historical memories that may cause compounding errors will be effectively filtered. Finally, multi-scale diffusion refinement decoders recover fine-grained flow details iteratively.

Overall, our key contributions in this paper are as follows:

- We propose a novel memory-enhanced autoregressive pipeline for joint sequential scene flow estimation and prediction based on the next-frame forecasting mechanism.
- A memory buffer stores multi-frame history flows and flow-related hidden states in the coarsest stage that are historically correlated and retrieved to forecast the next-frame flow. This strategy strengthens the long-term temporal consistency while boosting the future flow extrapolation capability.
- To rectify erroneous propagation compounding along multiple history memories, we design an uncertainty-aware measurement module to quantify per-point reliability and reweight the memory fusion process.
- Our method achieves state-of-the-art performance in both flow estimation and prediction tasks. Notably, our MESSENGER outperforms recent leading method GenFlow3D, reducing EPE3D by 71.6% on nuScenes and 67.7% on Argoverse 2 in long-horizon future extrapolation.

2 Related Work

Pair-wise Scene Flow Estimation. Early methods utilize stereo [35, 36, 37] or RGB-D images [38, 39] as inputs, lacking 3D awareness. As a milestone work, FlowNet3D [1] first introduces learnable neural networks for end-to-end flow estimation built on PointNet++ [40]. PointPWC-Net [2] further introduces the PWC design, which iteratively warps the first frame and calculates cost volume with the second frame. FLOT [19] proposes an optimal-transport-based method for point matching. Merging point and voxel representations together, PV-RAFT [41] designs both local and global feature extraction. To enhance matching reliability, 3DFlow [23], Bi-PointFlowNet [24], and MSBRN [42] introduce bi-directional association layers or backward verification modules. To enlarge the receptive field for cross-frame correlation, GMSF [43] develops transformer-based scene flow estimation with global matching. Recently, some approaches such as DifFlow3D [25, 26] focus on generative probabilistic distributions based on diffusion models. Although these methods have achieved great advancements in the pairwise setting, they cannot capture long-range motion consistency due to limited temporal awareness [4]. Additionally, pairwise methods commonly suffer from suboptimal matching performance in challenging scenarios, e.g, occlusion, high dynamics.

Multi-Frame Scene Flow Estimation and Prediction. There is increasing research attention pivoting to multi-frame scene flow estimation enhanced with long-term temporal coherence. Flow4D [32] proposes a 4D voxel network to extract spatio-temporal information from point cloud sequences. For efficiency, 4D processing is decomposed into 1D and 3D convolutions. MambaFlow [31] further reduces computational costs with Mamba backbone [44, 45] achieving linear complexity. To avoid growing features sizes from multiple frames, DeltaFlow [34] develops a Δ mechanism by computing residual flows rather than directly processing all raw frames. They also design Category-Balanced Loss and Instance Consistency Loss to improve estimation performance. Without requiring labor-intensive ground truth labeling, TEFlow [27] introduces a temporal ensembling strategy by retrieving flow-related supervisions from a multi-frame candidate pool. However, all these methods can only estimate in-between flows restrained in given observations, which fail to explore the effectiveness of future motion prediction. Also, they mostly rely on preliminary odometry registration results to obtain rigid flow. The most similar paper is GenFlow3D [26], which jointly supports estimation and prediction. However, their future prediction performance is extremely limited (e.g., one future frame in their default settings) due to accumulated errors compounding in long-horizon rollout in Fig. 1.

Memory Mechanism. Memory is widely explored to tackle occluded regions, sensor degradation, and establish temporal dependencies in various computer vision tasks, ranging from semantic segmentation [46, 47], object detection [48, 49], pose estimation [50, 51, 52, 53], etc. The effectiveness of memory has also been proved in optical flow [54, 55, 56], which facilitates recognition of occluded pixels or motion boundaries. MemFlow [54] designs an attention-based memory read-out and update module combined with a GRU-based refinement layer. MEMFOF [55] proposes a memory-efficient RAFT-based flow iteration module, with multi-frame context and correlation features cached and

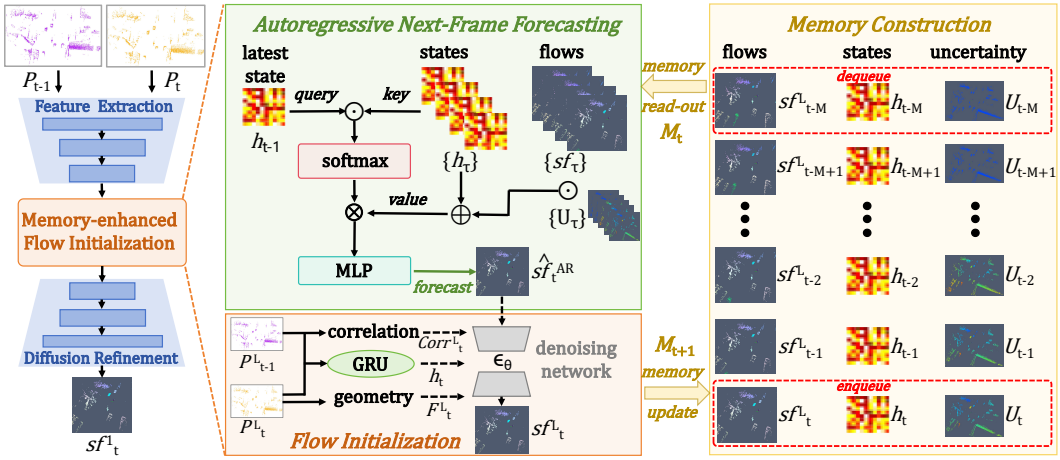

Figure 3: **Overall pipeline of our proposed MESSENGER.**

reused in the current cross-frame matching. ARFlow [56] further strengthens the memory construction based on an autoregressive Mamba refinement [57, 58], focusing on geometry-agnostic dynamics across frames. However, we notice that the memory in the corresponding 3D version of optical flow, namely scene flow, is still underexplored. In this paper, we for the first time combine history flows as memory priors for progressive next-frame flow forecasting to learn robust long-term temporal consistency.

3 Methodology

3.1 Preliminaries

Problem Definition. We propose a novel autoregressive framework on point cloud sequences for ***joint scene flow estimation and prediction***. Given H-length history observations $\mathcal{P} = \{P_1, P_2, \ldots, P_H\}$ where $P_t \in N \times 3$, our goal is to estimate the scene flows ***in between*** $\mathcal{SF}_t (t \leq H)$, representing the per-point displacements from P_{t-1} to P_t. Due to the explicit sequential memory storage and next-frame forecasting strategy, our method can also extrapolate well to K-length ***future*** scene flow predictions $\mathcal{SF}_t$ $(H < t \leq H + K)$ based on historical observations.

Pipeline Overview. As in Fig. 3, we adopt a sequential online estimation paradigm where all point sequences are delivered to the network frame-by-frame. For each timestamp t, only the current frame P_t is additionally inputted. For P_t, we follow [25, 26] to extract L-stage hierarchical point features as $\{F_t^l\}_{l=1}^{L}$. To effectively capture underlying spatio-temporal cues within LiDAR sequences and improve extrapolation capability, we dynamically maintain a memory buffer storing M-length temporally nearest history flows. These stored history flows are correlated by a transformer-based memory retrieval module to predict the current coarsest flow in a next-frame forecasting manner as flow initialization. To alleviate the erroneous history fusion and compounding errors, an uncertainty quantification module is designed to filter unreliable history motions. Finally, a multi-scale diffusion decoder [26] iteratively refines coarse flow in the upper layers.

3.2 Memory Buffers for Autoregressive Next-Frame Forecasting

In real-world LiDAR scans, dynamics mostly tend to be consistent across consecutive frames [27, 54]. Motivated by this observation, we develop a memory buffer $\mathcal{M}$ that explicitly stores multi-frame history coarsest flows to effectively capture long-term spatio-temporal priors. By correlating these history flows, the initial current flow is forecast with enhanced motion propagation. This design offers two advantages: (1) Memory is constructed only in the coarsest layer with ***minimal computational overhead***. (2) The progressive next-frame forecasting paradigm naturally learns how to predict future motions from history priors, leading to significantly ***enhanced flow extrapolation ability***.

Memory Construction. At any timestep t, the memory buffer maintains a fixed-length queue of historical explicit flows and implicit states from the nearest M frames:

$$\mathcal{M}_t = \{(\mathbf{h}_\tau, \mathcal{SF}_\tau^L)\}_{\tau=t-M}^{t-1}, \tag{1}$$

where $\mathbf{h}_\tau \in \mathbb{R}^{n\times C}$ denotes the GRU-based history hidden states following [26] in the coarsest stage. $\mathcal{SF}_\tau^L \in \mathbb{R}^{n\times 3}$ is the estimated history flows in the coarsest stage, where n is the point number.

Memory Retrieval for Next-Frame Forecasting. To retrieve useful motion priors for strengthening the current matching, we formulate the temporal querying process by a cross-attention mechanism. Assuming that motion states are consistent in a short period [26], the implicit states can reveal high-level motion tendencies while the latest state retains the highest similarity with the current one. Therefore, We first generate the *query* ($\mathbf{Q}_t$) from the latest flow state and *key* ($\mathbf{K}_{\text{mem}}$) from multiple stacked hidden states:

$$\mathbf{Q}_t = \mathbf{W}_Q \mathbf{h}_{t-1}, \quad \mathbf{K}_{\text{mem}} = [\mathbf{W}_K \mathbf{h}_{t-M} \oplus \cdots \oplus \mathbf{W}_K \mathbf{h}_{t-1}], \tag{2}$$

where $\oplus$ indicates feature concatenation. Crucially, the *value* ($\mathbf{V}_{\text{mem}}$) matrix should consider both low-level explicit historical flows and high-level implicit motion tendencies as:

$$\mathbf{V}_{\text{mem}} = \mathbf{W}_V[[\mathbf{h}_{t-M} \oplus \mathcal{SF}_{t-M}^L] \oplus \cdots \oplus [\mathbf{h}_{t-1} \oplus \mathcal{SF}_{t-1}^L]], \tag{3}$$

where $\mathbf{W}_Q, \mathbf{W}_K$, and $\mathbf{W}_V$ are learnable linear projections. Afterwards, by computing the scaled dot-product attention, the network aggregates ***historical*** flows and states from the memory buffer, generating the predicted ***current-timestamp*** flow $\hat{\mathcal{SF}}_t^{\text{AR}}$ in a next-frame forecasting paradigm:

$$\hat{\mathcal{SF}}_t^{\text{AR}} = \text{MLP}\left(\text{Softmax}\left(\frac{\mathbf{Q}_t \mathbf{K}_{\text{mem}}^\top}{\sqrt{d_k}}\right)\mathbf{V}_{\text{mem}}\right), \tag{4}$$

where d_k is the feature dimension in transformer. The next-frame forecasting can guarantee long-horizon motion consistency and smooth trajectories by attending to past flow priors.

Memory Update. The coarse flow directly predicted from history above lacks current-frame geometry details and motion variations. Therefore, we also refine it with a diffusion-based initialization module [28] conditioned by current geometric features F_t^L, correlation features [2] $Corr_t^L = \text{CostVolume}(F_{t-1}^L, F_t^L)$, and current hidden states [26] $h_t = \text{GRU}(h_{t-1}, F_t^L, Corr_t^L)$. The refined flow $\mathcal{SF}_t^L$ is then stored to dynamically update our memory buffer in a First-In-First-Out (FIFO) manner in Fig. 3. The newest generated couple $(\mathbf{h}_t, \mathcal{SF}_t^L)$ is pushed into $\mathcal{M}_{t+1}$, while the farthest-frame memory $(\mathbf{h}_{t-M}, \mathcal{SF}_{t-M}^L)$ is discarded for the next-frame (P_{t+1}) flow estimation.

3.3 Uncertainty-Guided Memory Reweighting

Forecasting the current flow from history memories fails to consider cross-frame chronological inconsistency or even introduce outliers, especially when objects undergo ***sudden non-rigid deformations***, ***abrupt acceleration***, or ***prolonged occlusions*** [59]. Furthermore, the autoregressive paradigm commonly suffers from ***compounding errors***: erroneous predictions from previous time steps accumulate and propagate to later frames [60]. To prevent our progressive flow estimation from being misled by flawed historical estimates, we propose a learnable uncertainty quantification map to dynamically reweight per-frame feature reliability during memory retrieval.

Discrepancy-based Uncertainty Measurement. We hypothesize that the uncertainty of historical memory can be quantified by the discrepancy between temporal motion priors and currently-refined dynamics. We compute the historical-current flow divergence with a flow residual as:

$$\mathbf{Res}_t = \hat{\mathcal{SF}}_t^{\text{AR}} - \mathcal{SF}_t^L, \tag{5}$$

where $\mathbf{Res}_t \in \mathbb{R}^{n\times 3}$ is per-point corresponding to the current point cloud P_t. A large magnitude of $\mathbf{Res}_t$ indicates that the historically interacted flow $\hat{\mathcal{SF}}_t^{\text{AR}}$ strongly contradicts the current geometric evidence, which reveals potential tracking failure or unpredictable inconsistent motions. We then map the residual to a continuous uncertainty scalar $\mathbf{U}_t^{(i)} \in [0, 1]$ for each point i via a L2-Normalization followed by a clamping function:

$$\mathbf{U}_t^{(i)} = \min\left(\max\left(\text{Norm}\|\mathbf{Res}_t^{(i)}\|_2, 0\right), 1\right). \tag{6}$$

Table 1: **Comparison of scene flow estimation results on nuScenes [61] and Argoverse 2 [62]**. We follow [26], inputting $H = 3$ consecutive frames to estimate two corresponding flows in between.

Method	Reference	Dataset	Scene Flow Estimation					
			$\mathcal{SF}_1$			$\mathcal{SF}_2$		
			EPE3D ↓	Acc3DS ↑	Out3D ↓	EPE3D ↓	Acc3DS ↑	Out3D ↓
PointPWC-Net [2]	ECCV'20	nuScenes	0.0386	0.7668	0.6255	0.0386	0.7695	0.6212
FLOT [19]	ECCV'20		0.0297	0.8561	0.5071	0.0293	0.8604	0.5049
PV-RAFT [41]	CVPR'21		0.0500	0.7570	0.6055	0.0492	0.7598	0.6024
Bi-PointFlowNet [24]	ECCV'22		0.0291	0.8676	0.4924	0.0291	0.8696	0.4913
PT-FlowNet [63]	RA-L'23		0.0264	0.8776	0.4753	0.0260	0.8821	0.4695
DiffFlow3D [25]	CVPR'24		0.0222	0.9205	0.4132	0.0221	0.9216	0.4118
GenFlow3D [26]	ICCV'25		<u>0.0161</u>	<u>0.9580</u>	<u>0.3399</u>	<u>0.0138</u>	<u>0.9756</u>	<u>0.3071</u>
Ours	—		**0.0154**	**0.9629**	**0.3283**	**0.0133**	**0.9767**	**0.2973**
PointPWC-Net [2]	ECCV'20	Argoverse 2	0.0553	0.6726	0.3681	0.0556	0.6742	0.3682
FLOT [19]	ECCV'20		0.0489	0.7681	0.3545	0.0491	0.7678	0.3569
PV-RAFT [41]	CVPR'21		0.0968	0.6588	0.4599	0.0971	0.6626	0.4587
Bi-PointFlowNet [24]	ECCV'22		0.0298	0.8995	0.2522	0.0301	0.8980	0.2543
PT-FlowNet [63]	RA-L'23		0.0386	0.8161	0.3163	0.0391	0.8154	0.3195
DiffFlow3D [25]	CVPR'24		0.0190	0.9635	0.1872	0.0189	0.9627	0.1887
GenFlow3D [26]	ICCV'25		<u>0.0175</u>	<u>0.9660</u>	<u>0.1823</u>	<u>0.0164</u>	<u>0.9711</u>	<u>0.1771</u>
Ours	—		**0.0167**	**0.9691**	**0.1760**	**0.0156**	**0.9743**	**0.1712**

Uncertainty-Guided Memory Reweighting. Quantified uncertainty is then used to dynamically gate the history flows retrieved from the memory buffer, filtering unreliable outliers and disabling inconsistent motion propagation. Specifically, multi-frame reliability $\mathbf{\Gamma}_\tau = \mathbf{1} - \mathbf{U}_\tau$ is also stored in the memory buffer (Section 3.2) as memory triplets: $(\mathbf{h}_\tau, \mathcal{SF}_\tau^L, \mathbf{\Gamma}_\tau)$. During the memory retrieval, historical flows are reweighted by this reliability factor per-point, obtaining uncertainty-aware *value*:

$$\mathbf{V}_\tau = \mathbf{W}_V[\mathbf{h}_\tau \oplus (\mathcal{SF}_\tau^L \odot (\mathbf{1} - \mathbf{U}_\tau))]. \tag{7}$$

Through reliability-guided reweighting, the network adaptively penalizes the influence of highly uncertain memory (e.g., with occlusion or dynamics). Consequently, it forces the model to rely more heavily on the current correlation when the history is deemed untrustworthy, while still exploiting robust historical priors to smooth trajectories in confident regions. Finally, multiple cascaded diffusion refinement decoders [28] iteratively upsample and refine coarse flow estimates with more fine-grained details, yielding multi-scale scene flow estimates $\{\mathcal{SF}_t^l\}_{l=1}^{L-1}$.

3.4 Loss Functions

Following previous diffusion-based scene flow methods [26, 28], we adopt multi-scale supervised loss functions on both estimated flows and flow residuals. Specifically, for each point frame t at each stage l, two loss functions are formulated as:

$$L_{res,t}^l = ||\Delta\overline{\mathcal{SF}}_t^l - \Delta\mathcal{SF}_t^l||, \quad L_{flow,t}^l = ||\overline{\mathcal{SF}}_t^l - \mathcal{SF}_t^l||, \tag{8}$$

where $\Delta\overline{\mathcal{SF}}_t^l$ and $\overline{\mathcal{SF}}_t^l$ denote the ground truth flow residual and full flow. For the layer l from $L-1$ to 1, full flows are recovered by adding estimated flow residuals derived from diffusion refinement onto upsampled coarse flows [25]: $\mathcal{SF}_t^l = \Delta\mathcal{SF}_t^l + up(\mathcal{SF}_t^{l+1})$. During training, both H observed frames and K future predicted frames are calculated:

$$L_{total} = \sum_{t=1}^{H+K-1} \sum_{l=1}^{L} \alpha^l (L_{res,t}^l + L_{flow,t}^l), \tag{9}$$

where $\{\alpha^l\}_{l=1}^L$ are weights in different stages.

4 Experiments

4.1 Datasets and Implementation Details

Evaluation Datasets. We leverage real-world LiDAR sequences in nuScenes [61] and Argoverse 2 [62] for evaluation. The nuScenes dataset captures LiDAR points in a 20 HZ frequency and chooses

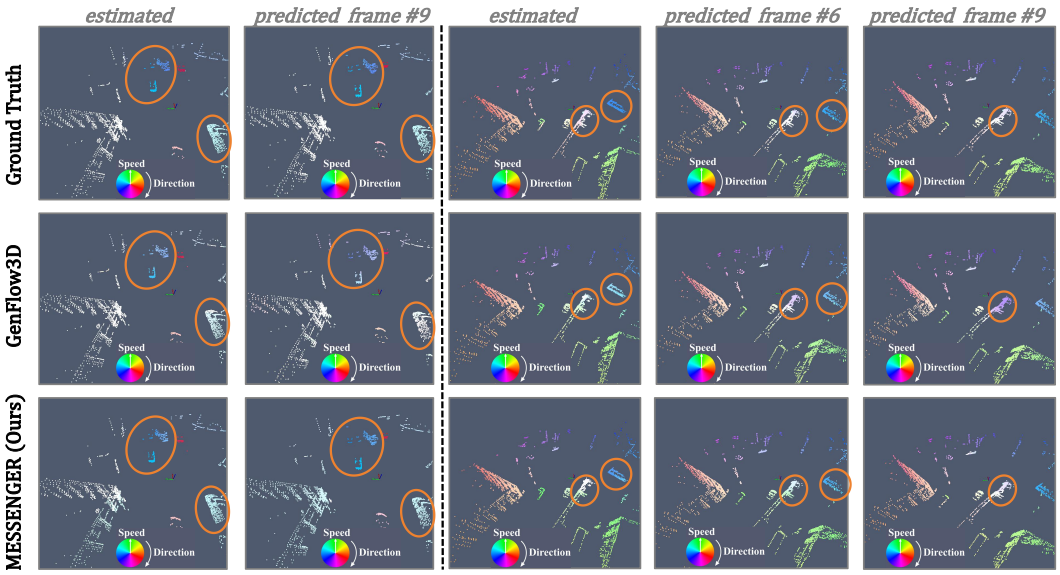

Figure 4: **Qualitative comparison in scene flow estimation and prediction on nuScenes.** From up to bottom, rows show ground truth, GenFlow3D [26], and our predictions across time. Scene flow is visualized with hue indicating direction and saturation representing speed. Comparison is highlighted with orange circles. Please zoom in for details.

Table 2: **Comparison of scene flow prediction results on nuScenes [61] and Argoverse 2 [62]**. We set the observed frames as $H = 6$ and predict future frames with the intervals of $k = 1, 2, 3, 4$.

	Methods	$\mathcal{SF}_{H+1}$		$\mathcal{SF}_{H+2}$		$\mathcal{SF}_{H+3}$		$\mathcal{SF}_{H+4}$	
		EPE3D ↓	Acc3DS ↑	EPE3D ↓	Acc3DS ↑	EPE3D ↓	Acc3DS ↑	EPE3D ↓	Acc3DS ↑
nuScenes	GenFlow3D [26]	0.029	0.948	0.046	0.889	0.065	0.830	0.088	0.754
	Ours	**0.015**	**0.969**	**0.018**	**0.938**	**0.021**	**0.915**	**0.025**	**0.883**
	Methods	$\mathcal{SF}_{H+1}$		$\mathcal{SF}_{H+2}$		$\mathcal{SF}_{H+3}$		$\mathcal{SF}_{H+4}$	
		EPE3D ↓	Acc3DS ↑	EPE3D ↓	Acc3DS ↑	EPE3D ↓	Acc3DS ↑	EPE3D ↓	Acc3DS ↑
Argoverse 2	GenFlow3D [26]	0.049	0.651	0.105	0.279	0.140	0.172	0.192	0.130
	Ours	**0.025**	**0.915**	**0.046**	**0.713**	**0.054**	**0.636**	**0.062**	**0.555**

keyframes with 10-frame intervals. We follow [26] to construct evaluation sequences, using 650 frames for training and the other 200 frames for testing. For the Argoverse 2 dataset, it scans in a 10 HZ frequency. Following [26], we curate training (700 scenes) and evaluation (150 scenes) subsequences, and exclude invalid points into loss computation. Odometry and tracking results are used to calculate ground truth labels. We follow [1, 19, 41], choosing EPE3D, Acc3DS, and Out3D as our main metrics, representing average estimation errors, accuracy, and outlier rate, respectively.

Implementation Details. Unlike GenFlow3D [26] which can only process short fixed-length sequences, our autoregressive formulation can support arbitrary-length sequential input. Thus, we train our model on all LiDAR frames within the same sequence. When testing, for scene flow estimation, we set the existing length as $H = 3$. For scene flow prediction, we set the existing length as $H = 6$ as default and varies the future frame length K from 1 to 4. Following [2, 19, 64, 28], 8192 points are randomly sampled, and ground points and points farther than 35m are filtered. The learning rate is initialized as 0.001 and reduced by 0.7 every 40 epochs until convergence. We adopt the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$. The diffusion decoder has $L = 4$ stages and their loss weights are set as $\{\alpha^l\}_{l=1}^4 = \{0.02, 0.04, 0.08, 0.16\}$, respectively. We leverage DDIM [65] with the total timestamp of 1000 while using 1-step denoising during the inference following [25]. The length of our designed memory buffer is $M = 6$.

4.2 Comparison in Scene Flow Estimation

To assess the effectiveness of our proposed scene flow estimator, we follow [26] to deliver three frames as input and estimate two corresponding flows in between on both nuScenes and Argoverse 2. As in Table 1, our method outperforms all previous scene flow estimators in terms of all evaluation

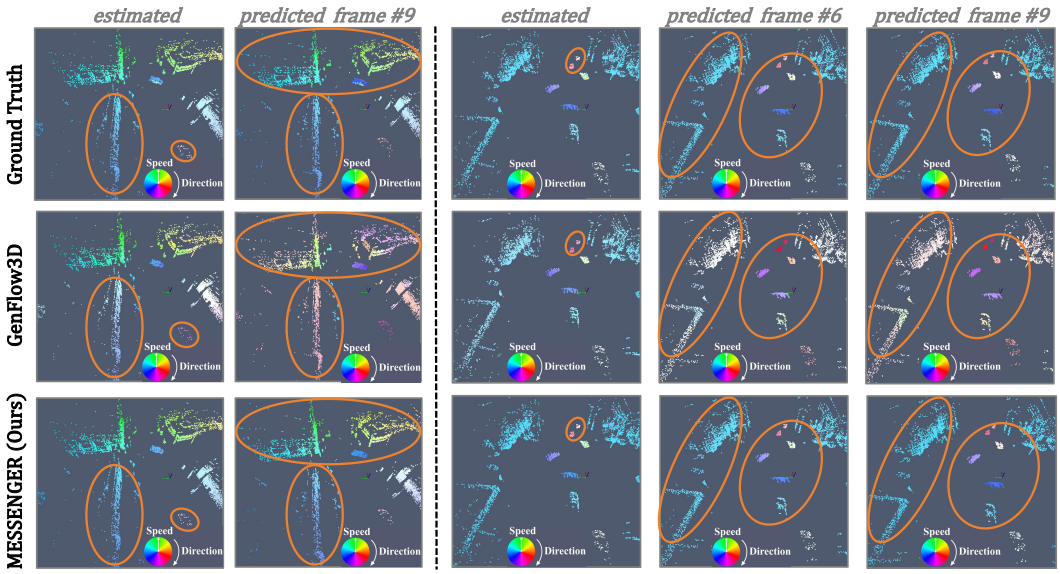

Figure 5: **Qualitative comparison in scene flow estimation and prediction on Argoverse 2.** From up to bottom, rows show ground truth, GenFlow3D [26], and our predictions across time. Scene flow is visualized with hue indicating direction and saturation representing speed. Comparison is highlighted with orange circles. Please zoom in for details.

metrics, including EPE3D, Acc3DS, Out3D. Compared to the recent SOTA approach GenFlow3D [26], our MESSENGER has lower EPE3D and Out3D metrics, and also achieves higher Acc3DS. This advantage is further demonstrated in Fig. 4 and Fig. 5, where our estimated flows are more accurate than GenFlow3D's output, especially in challenging scenarios such as dynamic cars and repetitive patterns. However, we notice that the discrepancy between GenFlow3D and ours is small in 3-frame input, mainly because the temporal information is limited. Therefore, we further increase the input frames in Fig. 6, where GenFlow3D has triple higher errors, while ours has even lower errors.

4.3 Comparison in Future Scene Flow Prediction

We also compare the extrapolation ability of future scene flow prediction in Table 2. We input $H = 6$ observed frames and then predict the future flows with a length of $K = 4$. On both nuScenes and Argoverse 2, our method has superior extrapolation ability, especially in much longer predictions. When forecasting one future frame, our method has only half of the EPE3D estimation error of GenFlow3D [26]. The performance gap is further enlarged with the extrapolation frame getting longer from $k = 1$ to 4. As for the 4-th frame forecasting, our method has less than 1/3 estimation errors compared to GenFlow3D. On Argoverse dataset, GenFlow3D witnesses obvious decreasing Acc3DS from 0.651 to 0.130. In contrast, ours has way lower drop from 0.915 to 0.555. The significantly improved extrapolation capability is attributed to our designed autoregressive next-frame forecasting module, which naturally learns how to predict future flows based on history observed ones. Furthermore, our uncertainty-aware weighting can filter outliers in historically stored flows, ensuring high temporal consistency. In Fig. 4 and Fig. 5, we visualize the long-term prediction results. For the 6-th frame prediction and the 9-th frame, GenFlow3D witnesses huge performance degradation with much larger drifts on both foreground and background objects. Nevertheless, our method still keeps consistently accurate predictions with much longer rollouts.

4.4 Efficiency Analysis

In Fig. 2, we have already compared the GPU memory where our GPU consumption is constant to the input frame length. Here, we also additionally compare the latency with previous diffusion-based methods DifFlow3D and GenFlow3D. As in Table 3, compared to the other two methods, our method achieves the highest efficiency in terms of one forward latency.

Table 3: **Latency comparison.**

Methods	DifFlow3D [25]	GenFlow3D [26]	Ours
Latency	216.7ms	88.7ms	55.9ms

Table 4: **Ablation studies on the nuScenes dataset.** We evaluate the impact of core architecture designs, memory lengths, and memory stored contents in the 4-th future prediction setting.

(a) Core novelty and uncertainty.

	EPE3D ↓	Acc3DS ↑
w/o Memory	0.088	0.754
w/o Uncertainty	0.045	0.668
MLP-based Uncertainty	0.029	0.844
Norm-based Uncertainty	**0.025**	**0.883**

(b) Memory lengths.

M	EPE3D ↓	Acc3DS ↑
3	0.040	0.764
5	0.035	0.806
6	**0.025**	**0.883**
8	0.031	0.823

(c) Memory Construction.

	State	Flow	Uncertainty	EPE3D ↓	Acc3DS ↑
a)	✗	✓	✓	0.029	0.834
b)	✓	✗	✓	0.031	0.827
c)	✗	✓	✗	0.033	0.807
d)	✓	✗	✗	0.066	0.415
e)	✓	✓	✓	**0.025**	**0.883**

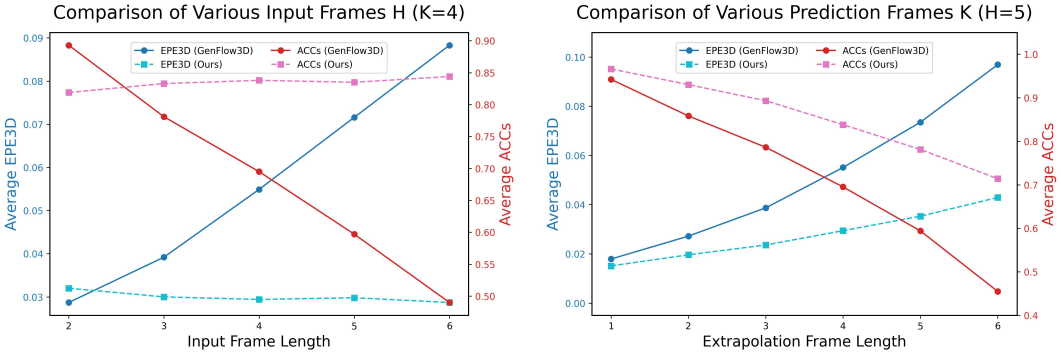

Figure 6: **Comparison of different input frames (H) and extrapolation frames (K) for future scene flow prediction on nuScenes.**

4.5 Ablation Studies

Memory Construction. In Table 4a, we remove the designed memory buffer, leading to more than triple increasing errors. This demonstrates the extrapolation ability relies significantly on explicit memory storage with next-frame prediction. We also conduct ablations with various memory length in Table 4b, where from $M = 3$ to 6, the prediction quality becomes better. However, storing more frames leads to a slight performance degradation partly because farther frames share scarce semantic similarities with the current one, which can be viewed as outliers. We also compare various storage contents in memory buffers as in Table 4c. Comparing a) and b) with e), both explicit flow storage and implicit state storage are vital to the prediction performance.

Uncertainty. Without designed uncertainty, there is almost double EPE3D increase in Table 4a. This proves that uncertainty can effectively rectify long-term error accumulations. We also compare different construction methods where Normalization-based manner is better than the MLP-based one. In memory buffer, we explicitly store prior-frame uncertainty estimates. Without storing uncertainty in Table 4c, there is obviously worse prediction comparing c) to a), and d) to b).

5 Discussions

Why MESSENGER Excels Significantly in Prediction? Our method has significantly superior extrapolation ability as in Table 2 compared to recent SOTA method GenFlow3D [26]. This advantage can be attributed to three designs: (1) Compared to implicit motion propagation, we explicitly store memory both flow estimates and hidden states, providing strong semantic and motion priors for future prediction. (2) Our designed autoregressive next-frame forecasting paradigm can force the model to unlock temporal dependencies from stored memories. (3) Furthermore, the proposed uncertainty quantification can mitigate long-term drifts commonly existing in autoregressive models. We further demonstrate this by varying the input and extrapolation frame length in Fig. 6. Evaluating the effect of input frames, we freeze the prediction step as $K = 4$ and change input frames. From Fig. 6 (Left), the baseline method GenFlow3D [26] witnesses a significant performance drop with more than triple EPE3D increase. In contrast, our method has even improved Acc3Ds and much lower EPE3D. This proves that the baseline method cannot effectively propagate history priors from temporal observations while ours could store and retrieve multiple useful memories, thereby strengthening the future prediction. We also compare different forecasting steps in Fig. 6 (Right), where the observed frame length is kept as $H = 5$. Both GenFlow3D and our method have a performance drop with

Table 5: **Per-class analysis of estimation errors on nuScenes.**

(a) Dynamic Bucket-Normalized EPE.

$\mathcal{SF}_1$	CAR	OTHER	PED.	VRU
GenFlow3D	0.0234	0.0257	0.0353	0.0418
Ours	**0.0209**	**0.0228**	**0.0291**	**0.0321**

(b) Three-way EPE.

$\mathcal{SF}_{H+4}$	FD	FS	BS	Class Average
GenFlow3D	0.137	0.072	0.086	0.098
Ours	**0.051**	**0.013**	**0.024**	**0.029**

longer rollouts due to potential new emerging motions or abrupt dynamics. However, our method holds higher stability with lower metric fluctuations.

How Uncertainty Works? We visualize two representative samples with colored uncertainty in Fig. 7, where uncertainty tends to exist in dynamic regions. This is partly because dynamic cars commonly introduce inconsistent motions across frames [59], where retrieved memory flows may have lower reference reliability to forecast the next frame. In this case, our method will adaptively force the network to learn from current geometric and correlation features, rather than forecasting.

Fine-grained Per-class Error Analysis. We additionally evaluate the Dynamic Bucket-Normalized EPE metric, proposed in TrackFlow [66], which normalizes per-point EPE by the mean speed within predefined motion buckets and classifies estimates into four categories including regular cars (CAR), other vehicles (OTHER), pedestrians (PED.), and wheeled vulnerable road users (VRU). A direct EPE3D comparison with GenFlow3D on the nuScenes scene flow estimation is shown in Table 5a. Generally, pedestrians and vulnerable road users, such as bicycles or motorcycles, have higher estimation errors than regular cars for both methods. This is partly because PED. and VRU have smaller size, non-rigid motions, and fewer LiDAR points scanning on them, making consistent point correlation across consecutive frames much harder. Our method surpasses GenFlow3D in all dynamic classes, especially in challenging classes PED. and VRU. We also report the Three-way EPE metric [67], which computes the unweighted average EPE over three categories: foreground dynamic (FD), foreground static (FS) and background static (BS). As in Table 5b, our particular advantage in dynamic regions can be attributed to our designed memory buffer, which can store multi-frame historical flows and features, ensuring stronger motion consistency.

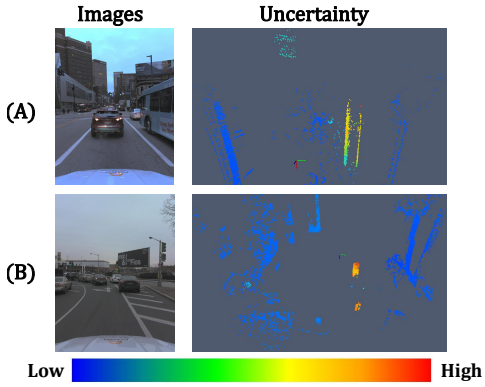

Figure 7: **Visualization of uncertainty.** We map the learned uncertainty values to a corresponding color strip.

6 Conclusion

In this paper, we proposed a novel memory-enhanced sequential scene flow estimation pipeline based on autoregressive next-frame forecasting. The core design is a dynamically maintained memory buffer which stores explicit coarse flows, implicit states, and reliability scores from multiple history timestamps. The current flow is initialized by progressively forecasting the next frame motions based on retrieved memory features. To alleviate potential compounding errors and filter unreliable outliers, we develop an uncertainty measurement to indicate per-point trustworthy score when weighting history memories. Extensive experiments demonstrate our method outperforms existing scene flow methods in both estimation and prediction, especially in long-horizon future predictions.

Limitations and Broader Impacts. Although our work based on attention-based autoregressive forecasting has future prediction capability within a limited frame length, future work could investigate how to incorporate large-scale pretrained backbone such as Wan [68] for long-horizon world modeling [69, 70, 71, 72, 73, 74]. Scene flow represents dense 4D world dynamics, which can strengthen motion modeling in recent world models.

Acknowledgments. We thank Qingwen Zhang from KTH for her assistance in running open-source codes for efficiency comparison. In addition, Jiuming Liu appreciates the Trinity Hall and Cambrdige Trust for the International Scholarship they provided.

References

[1] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019.

[2] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 88–107. Springer, 2020.

[3] Zhiheng Feng, Jiuming Liu, and Hesheng Wang. Optimizing scene flow with neural rigidity prior. *Robot Learning*, 1(1):1–14, 2024.

[4] Mengmeng Liu, Jiuming Liu, Michael Ying Yang, Chaokang Jiang, Jiangtao Li, Yunpeng Zhang, Hesheng Wang, Francesco Nex, and Hao Cheng. Streamvlo: Streaming visual-lidar odometry with cumulative drift compensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 39086–39097, 2026.

[5] Xiaole Tang, Jiayi Xu, Xiang Gu, Yan Yang, and Jian Sun. Binding multiple modalities via multimodal wasserstein barycenter. *arXiv preprint arXiv:2609.33800*, 2026.

[6] Yifei Dong, Fengyi Wu, Yilong Dai, Lingdong Kong, Guangyu Chen, Xu Zhu, Qiyu Hu, Tianyu Wang, Johnalbert Garnica, Feng Liu, et al. Language-conditioned world modeling for visual navigation. *arXiv preprint arXiv:2603.26741*, 2026.

[7] Jiuming Liu, Zheng Huang, Mengmeng Liu, Tianchen Deng, Francesco Nex, Hao Cheng, and Hesheng Wang. Topolidm: Topology-aware lidar diffusion models for interpretable and realistic lidar point cloud generation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8180–8186. IEEE, 2025.

[8] Xinjin Li, Yu Ma, Kaisen Ye, Jinghan Cao, Minghao Zhou, and Yeyang Zhou. Hy-facial: Hybrid feature extraction by dimensionality reduction methods for enhanced facial expression classification. In *Eighteenth International Conference on Machine Vision (ICMV 2025)*, volume 14114, pages 206–213. SPIE, 2026.

[9] Mengmeng Liu, Jiuming Liu, Yunpeng Zhang, Jiangtao Li, Michael Ying Yang, Francesco Nex, and Hao Cheng. 4dstr: Advancing generative 4d gaussians with spatial-temporal rectification for high-quality and consistent 4d generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7224–7232, 2026.

[10] Mengmeng Liu, Diankun Zhang, Jiuming Liu, Jianfeng Cui, Hongwei Xie, Guang Chen, Hangjun Ye, Francesco Nex, Hao Cheng, and Michael Ying Yang. Universe: Unified video action models for autonomous driving with flexible mask-modulated modality generation. *arXiv preprint arXiv:2607.05133*, 2026.

[11] Tianchen Deng, Xuefeng Chen, Yi Chen, Qu Chen, Yuyao Xu, Lijin Yang, Le Xu, Yu Zhang, Bo Zhang, Wuxiong Huang, and Hesheng Wang. Gaussiandwm: 3d gaussian driving world model for unified scene understanding and multi-modal generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10656–10667, June 2026.

[12] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Björn Ommer, and Andreas Geiger. Slim: Self-supervised lidar scene flow and motion segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13126–13136, 2021.

[13] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 424–443. Springer, 2022.

[14] Mengmeng Liu, Hao Cheng, and Michael Ying Yang. Tracing the influence of predecessors on trajectory prediction. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3245–3255. IEEE, 2023.

[15] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. Sni-slam: Semantic neural implicit slam. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21167–21177. IEEE, 2024.

[16] Jiuming Liu, Guangming Wang, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Translo: A window-based masked point transformer framework for large-scale lidar odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1683–1691, 2023.

[17] Tianchen Deng, Chang Nie, Shuhong Liu, Wenhua Wu, Jianfei Yang, Shenghai Yuan, Jiuming Liu, Zhe Liu, Danwei Wang, and Hesheng Wang. Voxelized 3d gaussian representation for dense visual slam on embedded vision system. *International Journal of Computer Vision*, 134(10):445, 2026.

[18] Jiuming Liu, Dong Zhuo, Zhiheng Feng, Siting Zhu, Chensheng Peng, Zhe Liu, and Hesheng Wang. Dvlo: Deep visual-lidar odometry with local-to-global feature fusion and bi-directional structure alignment. In *European Conference on Computer Vision*, pages 475–493. Springer, 2024.

[19] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *European conference on computer vision*, pages 527–544. Springer, 2020.

[20] Ruibo Li, Chi Zhang, Guosheng Lin, Zhe Wang, and Chunhua Shen. Rigidflow: Self-supervised scene flow learning on point clouds by local rigidity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16959–16968, 2022.

[21] Yair Kittenplon, Yonina C Eldar, and Dan Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4114–4123, 2021.

[22] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[23] Guangming Wang, Yunzhe Hu, Zhe Liu, Yiyang Zhou, Masayoshi Tomizuka, Wei Zhan, and Hesheng Wang. What matters for 3d scene flow network. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 38–55. Springer, 2022.

[24] Wencan Cheng and Jong Hwan Ko. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. In *European Conference on Computer Vision*, pages 108–124. Springer, 2022.

[25] Jiuming Liu, Guangming Wang, Weicai Ye, Chaokang Jiang, Jinru Han, Zhe Liu, Guofeng Zhang, Dalong Du, and Hesheng Wang. 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*, pages 15109–15119, 2024.

[26] Hanlin Li, Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Genflow3d: Generative scene flow estimation and prediction on point cloud sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27488–27497, 2025.

[27] Qingwen Zhang, Chenhan Jiang, Xiaomeng Zhu, Yunqi Miao, Yushan Zhang, Olov Andersson, and Patric Jensfelt. Teflow: Enabling multi-frame supervision for self-supervised feed-forward scene flow estimation. *arXiv preprint arXiv:2602.19053*, 2026.

[28] Jiuming Liu, Weicai Ye, Guangming Wang, Chaokang Jiang, Lei Pan, Jinru Han, Zhe Liu, Guofeng Zhang, and Hesheng Wang. Difflow3d: Hierarchical diffusion models for uncertainty-aware 3d scene flow estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2025.

[29] Zhiheng Fu, Zixu Li, Zhiwei Chen, Fangxu Liu, Yupeng Hu, Weili Guan, and Liqiang Nie. Egoaction: Egocentric action composition with reliability-aware temporal fusion for the epic-kitchens action detection challenge at cvpr 2026. *arXiv preprint arXiv:2605.24496*, 2026.

[30] Mengmeng Liu, Michael Ying Yang, Jiuming Liu, Yunpeng Zhang, Jiangtao Li, Sander Oude Elberink, George Vosselman, and Hao Cheng. Dvlo4d: Deep visual-lidar odometry with sparse spatial-temporal fusion. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9740–9747. IEEE, 2025.

[31] Jiehao Luo, Jintao Cheng, Qingwen Zhang, Bohuan Xue, Rui Fan, and Xiaoyu Tang. Mambaflow: A novel and flow-guided state space model for scene flow estimation. *IEEE Transactions on Intelligent Vehicles*, 2026.

[32] Jaeyeul Kim, Jungwan Woo, Ukcheol Shin, Jean Oh, and Sunghoon Im. Flow4d: Leveraging 4d voxel network for lidar scene flow estimation. *IEEE Robotics and Automation Letters*, 2025.

[33] Qingwen Zhang, Yi Yang, Heng Fang, Ruoyu Geng, and Patric Jensfelt. Deflow: Decoder of scene flow network in autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2105–2111. IEEE, 2024.

[34] Qingwen Zhang, Xiaomeng Zhu, Yushan Zhang, Yixi Cai, Olov Andersson, and Patric Jensfelt. Deltaflow: An efficient multi-frame scene flow estimation method. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[35] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7. IEEE, 2007.

[36] Zixu Li, Yupeng Hu, Zhiwei Chen, Haokun Wen, Xuemeng Song, and Liqiang Nie. Combiner: Composed image retrieval guided by attribute-based neighbor relations. *IEEE Transactions on Image Processing*, 2026.

[37] Zixu Li, Yupeng Hu, Zhiheng Fu, Zhiwei Chen, Yongqi Li, and Liqiang Nie. Tema: Anchor the image, follow the text for multi-modification composed image retrieval. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24421–24442, 2026.

[38] Simon Hadfield and Richard Bowden. Kinecting the dots: Particle based scene flow from depth sensors. In *2011 International Conference on Computer Vision*, pages 2290–2295. IEEE, 2011.

[39] Zixu Li, Yupeng Hu, Zhiwei Chen, Qinlei Huang, Guozhi Qiu, Zhiheng Fu, and Meng Liu. Retrack: Evidence-driven dual-stream directional anchor calibration network for composed video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23373–23381, 2026.

[40] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[41] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. Pv-raft: Point-voxel correlation fields for scene flow estimation of point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6954–6963, 2021.

[42] Wencan Cheng and Jong Hwan Ko. Multi-scale bidirectional recurrent network with hybrid correlation for point cloud based scene flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10041–10050, 2023.

[43] Yushan Zhang, Johan Edstedt, Bastian Wandt, Per-Erik Forssén, Maria Magnusson, and Michael Felsberg. Gmsf: Global matching scene flow. *Advances in Neural Information Processing Systems*, 36:64415–64427, 2023.

[44] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[45] Jiuming Liu, Ruiji Yu, Yian Wang, Yu Zheng, Tianchen Deng, Weicai Ye, and Hesheng Wang. Point mamba: A novel point cloud backbone based on state space model with octree-based ordering strategy. *arXiv preprint arXiv:2403.06467*, 2024.

[46] Enxu Li, Sergio Casas, and Raquel Urtasun. Memoryseg: Online lidar semantic segmentation with a latent memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 745–754, 2023.

[47] Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8219–8228, 2021.

[48] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10337–10346, 2020.

[49] Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 4086–4096, 2017.

[50] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[51] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Haoang Li, Mengmeng Liu, Tianchen Deng, Marc Pollefeys, Michael Ying Yang, and Hesheng Wang. Regformer++: An efficient large-scale 3d lidar point registration network with projection-aware 2d transformer. *arXiv preprint arXiv:2603.14290*, 2026.

[52] Mengmeng Liu, Hao Cheng, Lin Chen, Hellward Broszio, Jiangtao Li, Runjiang Zhao, Monika Sester, and Michael Ying Yang. Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2039–2049. IEEE, 2024.

[53] Hao Cheng, Mengmeng Liu, Lin Chen, Hellward Broszio, Monika Sester, and Michael Ying Yang. Gatraj: A graph-and attention-based multi-agent trajectory prediction model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 205:163–175, 2023.

[54] Qiaole Dong and Yanwei Fu. Memflow: Optical flow estimation and prediction with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19068–19078, 2024.

[55] Vladislav Bargatin, Egor Chistov, Alexander Yakovenko, and Dmitriy Vatolin. Memfof: High-resolution training for memory-efficient multi-frame optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8187–8196, 2025.

[56] Jiuming Liu, Mengmeng Liu, Siting Zhu, Yunpeng Zhang, Jiangtao Li, Michael Ying Yang, Francesco Nex, Hao Cheng, and Hesheng Wang. Arflow: Auto-regressive optical flow estimation for arbitrary-length videos via progressive next-frame forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026.

[57] Jiuming Liu, Jinru Han, Lihao Liu, Angelica I Aviles-Rivero, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Mamba4d: Efficient 4d point cloud video understanding with disentangled spatial-temporal state space models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17626–17636. IEEE, 2025.

[58] Igor Mihajlovic, Jiuming Liu, and Hesheng Wang. Pcpmamba: Large-scale point cloud prediction based on spatio-temporal state space models. In *2025 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 793–799. IEEE, 2025.

[59] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang. Regformer: an efficient projection-aware transformer network for large-scale point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8451–8460, 2023.

[60] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[61] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[62] 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.

[63] Jingyun Fu, Zhiyu Xiang, Chengyu Qiao, and Tingming Bai. Pt-flownet: Scene flow estimation on point clouds with point transformer. *IEEE Robotics and Automation Letters*, 8(5):2566–2573, 2023.

[64] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3254–3263, 2019.

[65] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[66] Ishan Khatri, Kyle Vedder, Neehar Peri, Deva Ramanan, and James Hays. I can't believe it's not scene flow! In *European Conference on Computer Vision*, pages 242–257. Springer, 2024.

[67] Nathaniel Chodosh, Deva Ramanan, and Simon Lucey. Re-evaluating lidar scene flow. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5993–6003. IEEE, 2024.

[68] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[69] Chaokang Jiang, Desen Zhou, Jiuming Liu, and Li Sun. Vectorworld: Efficient streaming world model via diffusion flow on vector graphs. In *Forty-third International Conference on Machine Learning*, 2026.

[70] Jiuming Liu, Chaojun Ni, Mengmeng Liu, Chensheng Peng, Fangjinhua Wang, Sitian Shen, Marc Pollefeys, Masayoshi Tomizuka, Ayush Tewari, and Per Ola Kristensson. Towards interactive video world modeling: Frontiers, challenges, benchmarks, and future trends. *arXiv preprint arXiv:2606.01164*, 2026.

[71] Yifei Dong, Fengyi Wu, Guangyu Chen, Lingdong Kong, Qiyu Hu, Yuxuan Zhou, Xu Zhu, Jingdong Sun, Jun-Yan He, Qi Dai, Alexander G. Hauptmann, and Zhi-Qi Cheng. Towards unified world models for visual navigation via memory-augmented planning and foresight. In *European Conference on Computer Vision*, pages 276–295. Springer, 2026.

[72] Yifei Dong, Fengyi Wu, Qi He, Lingdong Kong, Heng Li, Minghan Li, Zebang Cheng, Yuxuan Zhou, Jingdong Sun, Qi Dai, Alexander G Hauptmann, and Zhi-Qi Cheng. Ha-vln 2.0: An open benchmark and leaderboard for human-aware navigation in discrete and continuous environments with dynamic multi-human interactions, 2026.

[73] Mengmeng Liu, Diankun Zhang, Jiuming Liu, Jianfeng Cui, Hongwei Xie, Guang Chen, Hangjun Ye, Michael Ying Yang, Francesco Nex, and Hao Cheng. Driveva: Video action models are zero-shot drivers. In *European Conference on Computer Vision*, pages 315–335. Springer, 2026.

[74] Shuo Lu, Yinuo Xu, Kecheng Yu, Siru Jiang, Yongcan Yu, Yubin Wang, Haitao Yang, Yuxiang Zhang, Bin Wang, Ran He, et al. Worldcoder-bench: Benchmarking physically grounded 3d world synthesis. *arXiv preprint arXiv:2606.01869*, 2026.

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 main claims in the abstract and introduction faithfully reflect the paper's contributions and scope and match theoretical and experimental results.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have included a discussion about the paper's 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: [N/A]

   Justification: The paper does not present new theoretical results, theorems, or formal guarantees. The equations in the method section define the model architectures and loss functions.

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: The paper fully disclose all the information needed to reproduce the main experimental results. We also supplement the implementation code in the supplementary materials.

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

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

   Answer: [Yes]

   Justification: The experiments use publicly available datasets and benchmarks, including nuScenes and Argoverse 2, following their official access procedures and licenses. The code will be open-sourced after publication.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies the datasets, validation splits, evaluation metrics, optimizer, learning rate, weight decay, batch size, feature resolutions, and architecture details. We also supplement the implementation code in the supplementary materials.

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 hasn't included the error bars due to highly fixed results under our default settings.

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: The paper has provided sufficient information on the computer resources such as GPU memory usage in the main 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: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

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 have included a broader impact clarification in the paper.

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper has communicated the details of the dataset/code/model as part of submissions via structured templates.

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

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

    Answer: [N/A]

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

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

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

Answer: [N/A]

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.