Li-AutoFlow: Autoregressive Flow Matching for Continuous AV Scene Prediction

Anonymous CVPR submission

Paper ID *****

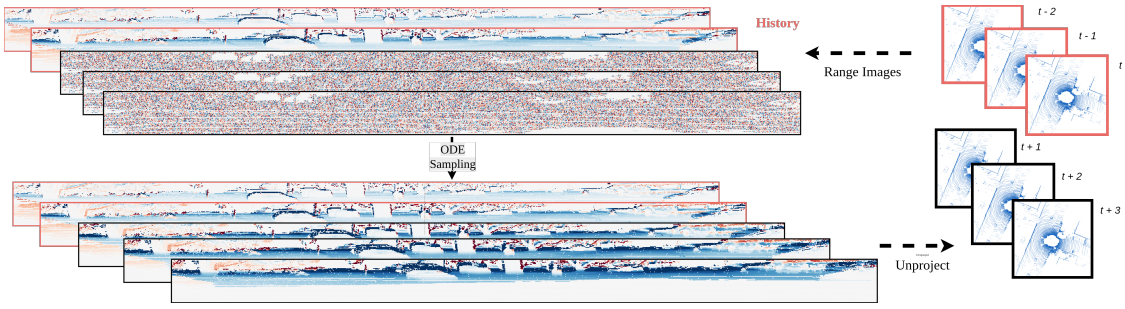

Figure 1. **Li-AutoFlow:** An autoregressive flow based method that predicts future LiDAR scans by conditioning on previous frames, utilizing an intermediate range image representation to ensure efficient generation.

Abstract

Predicting the evolution of complex 3D scenes is fundamental to safe autonomous navigation. While LiDAR provides the necessary geometric precision, existing forecasting methods often struggle with the inherent multimodality of dynamic environments. Deterministic models typically collapse to the conditional mean, resulting in "ghosting" artifacts and non-physical spatial interpolations that fail to represent discrete future contingencies.

In this paper, we introduce Li-AutoFlow, a generative framework that unifies Autoregressive Sequence Modeling with Flow Matching to produce high-fidelity, multimodal LiDAR forecasts. By operating directly on the continuous 2D range image manifold, our approach bypasses the limitations of discrete tokenization and lossy codebook bottlenecks found in prior generative works. Our formulation leverages straight-line probability flow ODEs to transport a noise prior into physically plausible future scenes, ensuring temporal causality through an autoregressive conditioning scheme.

Crucially, our tokenizer-free architecture maintains a fully differentiable pipeline from the flow-matching objective to the final reconstructed 3D point cloud. This enables direct optimization against 3D geometric metrics, such as Chamfer distance, ensuring centimeter-level precision. Experimental results on KITTI demonstrate that Li-AutoFlow significantly outperforms state-of-the-art deterministic and discrete-diffusion baselines, providing the coherent, disjoint scene samples necessary for robust motion planning in uncertain environments.

1. Introduction

Autonomous driving demands the ability to anticipate how the surrounding 3D scene will evolve over the coming seconds. Pedestrians change direction, vehicles accelerate or yield, and the observed scene geometry shifts continuously as both the ego vehicle and surrounding agents move. Accurate prediction of this evolution is critical: downstream tasks such as motion planning, collision avoidance, and scenario simulation all require reliable forecasts of future scene geometry [7, 20].

LiDAR provides a geometrically precise representation of the 3D environment, encoding surface depths with centimeter-level accuracy. Forecasting future LiDAR observations from past sweeps is therefore a key capability for autonomous systems. However, directly forecasting dense, unordered 3D point clouds is computationally prohibitive. To resolve this we formulate the problem in the native 2D range image projection [48]. By mapping sparse 3D coordinates into a dense, spherical grid, the range image natively encodes the physical constraints of the sensor while translating the problem into a structured manifold amenable to deep generative architectures. Operating in this space, LiDAR

forecasting presents two strict requirements. First, forecasts must maintain *geometric precision*, ensuring that predicted points lie on plausible physical surfaces with metric accuracy. Second, the model must capture *distributional uncertainty*, since future scenes are inherently multimodal due to the unpredictable behavior of dynamic agents.

While probabilistic forecasting is standard for object-level trajectories, the high dimensionality of dense LiDAR forecasting has largely restricted prior work [28, 29] to deterministic architectures optimized via regression losses. Consequently, these model regress towards conditional mean of the future distribution [36]. In dense 3D space, this expected-value optimization results in spatial interpolation between divergent futures, creating geometric artifacts: such as non-physical smeared surfaces and interpolated ghost objects. Modern AV planners require coherent, disjoint scene samples to evaluate discrete contingency plans (e.g., a car turning left versus going straight). A single, mean-collapsed prediction fails to provide the discrete hypotheses necessary for safe, multimodal contingency planning.

Generative models address this limitation by representing the full distribution of possible futures rather than collapsing to a single prediction [52]. We build on Flow Matching [2, 24, 26], a scalable generative framework that learns a continuous vector field transporting a simple noise prior to the target data distribution. This yields simulation-free training and efficient inference via straight-line probability flow ODEs, and has demonstrated strong results in image synthesis [15], video generation [5, 44], and robot policy learning [3].

Forecasting, however, imposes an additional constraint: temporal causality. Each predicted frame must condition on the observed history $x_{<t}$, with predictions fed back autoregressively into subsequent generations. A single-shot conditional model cannot capture this sequential dependence. Neither treatment captures the natural asymmetry of a forecasting problem, where past observations are known and clean while future frames must be generated stochastically.

Recent work has demonstrated that autoregressive generation can be unified with diffusion-based priors to achieve precisely this form of causally consistent sequence modeling [10, 38]. We bring this formulation to LiDAR forecasting for the first time: autoregressive conditioning enforces temporal consistency across the predicted sequence, ensuring that each frame is generated in the context of all preceding ones, while flow matching preserves the continuous structure of depth uncertainty at each step. The varying noise levels inherent in this autoregressive diffusion formulation confer an additional benefit: the model is trained to generate from observations at multiple fidelity levels, and consequently develops robustness to the measurement noise present in real LiDAR sensors.

The prior generative approach, Copilot4D [52], discretizes the scene through a VQ-VAE [43] and applies discrete diffusion over token indices. This design introduces a compounding three-fold limitation: a lossy codebook bottleneck that discards fine structural detail, a combinatorial complexity that ignores the smooth distributional structure of LiDAR data, and a structural decoupling between generation and 3D geometry. We move beyond this discrete bottleneck by operating directly on full-resolution range images without intermediate compression or discretization. Drawing on recent evidence that tokenizer-free generation is both feasible and effective [22], our approach preserves the continuous geometric relationships, sensor geometry, surface continuity, and occlusion patterns inherent to the range image manifold. Crucially, by maintaining a continuous pipeline from the flow-matching ODE to the final reconstructed point cloud, our framework is fully differentiable end-to-end. This allows us to fine-tune the generative model directly against 3D Chamfer distance, a level of geometric supervision that requires non-trivial approximations (e.g., straight-through estimators [43]) in discrete-token approaches, whereas our continuous formulation permits direct end-to-end optimization.

We present **Li-AutoFlow**, the first continuous, tokenizer-free, autoregressive flow matching model for LiDAR scene forecasting, operating directly in full-resolution range-image space. Our contributions are as follows:

- Autoregressive Flow Matching: We introduce an autoregressive flow matching scheme for multi-step prediction that maintains temporal coherence and expressiveness.
- Tokenizer-free Forecasting: We introduce the first tokenizer-free flow matching framework for LiDAR forecasting, generating full-resolution range images that preserve metric depth and sensor geometry without lossy compression.
- End-to-End Geometric Supervision: We enable 3D Chamfer fine-tuning through a fully differentiable reconstruction path, a capability unique to our continuous-space generation.

2. Related Work

LiDAR point cloud forecasting. Early approaches operated directly on unordered point sets using recurrent architectures [16], or exploited scene flow as a proxy for future state [27, 46]. Projecting each scan into a range image [11, 29, 30] reduces the problem of a structured 2D sequence prediction task, enabling convolutional and attention-based backbones at a fraction of the memory cost of volumetric representations [21, 53]. Subsequent work explored transformer-based temporal encoders [12, 28, 31], motion-augmented spatio-temporal convolutions [13], stochastic multi-modal prediction [47], state-space sequence models [40], and visual forecasting as a pre-training objective for camera-based driving [50]. All of these methods are

deterministic regressors, or at best model only a restricted parametric distribution; they collapse the future to a point estimate and cannot represent the full distribution over plausible scenes. Copilot4D [52] introduced discrete diffusion over VQ-VAE tokens as a generative alternative, but the VQ-VAE bottleneck discards fine metric structure before any generation occurs, degrading the geometric fidelity of predicted range images. We avoid this lossy compression and perform generative modeling directly in the pixel space of range-difference images, preserving full metric resolution throughout..

Diffusion and flow matching models. Denoising diffusion probabilistic models [19] and score-based models [39] learn to synthesize data by reversing a Gaussian corruption process, enabling high-fidelity generation through iterative denoising. Operating directly in pixel space is computationally expensive, so Latent Diffusion Models (LDM) [37] compress images with a VAE and perform all denoising in the resulting low-dimensional latent space; VQVAE-based tokenizers [35] push this further by quantizing the latent to a discrete codebook, enabling discrete diffusion [52] and autoregressive priors over compact token sequences. This compression pipeline dominates the field [4, 15] as the reconstruction loss of the VAE is small and it enables more efficient learnig. Flow Matching [1, 24, 25] reformulates generative modeling as regression onto an optimal transport problem. Compared to score-based diffusion, this eliminates the need for iterative Langevin corrections and reduces sampling to a single ODE integration, while providing more stable training targets and straighter trajectories. Flow matching has since become the backbone of several state-of-the-art generative model: for images in SD3 [14] and FLUX [15], and for video in Movie Gen [33], CogVideoX [51], and Wan [45].

However, these methods work on a pretrained latent space. Just image Transformers (JiT) [22] demonstrated that flow matching on raw pixels is viable at full resolution by replacing noise prediction with direct clean-image (x-)prediction. Under the manifold assumption, predicting the clean data point rather than the noised quantity allows a plain ViT operating on large pixel patches to achieve competitive generation quality without a pretrained tokeniser.

Autoregressive generative models. Large language models (LLMs) [6, 42] model joint distributions over discrete sequences by factorizing them into products of next-token conditionals, enabling scalable training and open-ended generation. This paradigm was extended to visual data by representing images as sequences of discrete VQ-VAE tokens and training autoregressive transformers [34, 35] or parallel masked predictors [8]. Visual AutoRegressive Modeling (VAR) [41] further reformulated image generation as coarse-to-fine next-scale prediction over multi-resolution token maps, achieving diffusion-level image quality with substantially faster inference. These approaches rely on a tokenizer that maps continuous signals into a discrete codebook, introducing an inherently lossy compression step. Masked Autoregressive (MAR) models [23] mitigate this limitation by replacing discrete token prediction with a per-token diffusion objective, enabling autoregressive generation directly in the continuous latent space of a VAE and improving reconstruction fidelity.

Autoregressive diffusion models. Integrating diffusion-based generation with autoregressive temporal structure has attracted growing interest. Diffusion Forcing [10] is the enabling insight: by assigning an independent, per-token noise level at training time, the model learns to condition on any mixture of clean context and partially-noised future tokens. This subsumes both teacher-forced sequence training and standard unconditional diffusion as special cases, and supports flexible inference strategies such as sliding-window generation and variable-horizon rollout. The Diffusion Forcing Transformer (DFoT) [9] instantiated this framework for high-resolution video generation with a transformer backbone. Related work on video prediction has also explored causal diffusion rollouts [18] and masked autoregressive generation with per-token diffusion heads [23], but these methods operate on natural-image pixels or VAE latents where metric precision is not a constraint. Our work is the first to apply autoregressive flow matching to LiDAR range image sequences, where the metric semantics of pixel values impose qualitatively different fidelity requirements on the generative transport process.

3. Background

3.1. Flow Matching

We formulate our generative model from the perspective of flow matching [2, 24, 26]. Consider a data distribution $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ and a noise distribution $\boldsymbol{\epsilon} \sim p_{\text{noise}}(\boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. During training, a noisy sample $\mathbf{z}_t$ is defined as an interpolation between data and noise:

$$\mathbf{z}_t = t\,\mathbf{x} + (1-t)\,\boldsymbol{\epsilon}, \quad t \in [0,1], \tag{1}$$

using a linear schedule [2, 24, 26], so that $\mathbf{z}_t \sim p_{\text{data}}$ when $t = 1$ and $\mathbf{z}_t \sim p_{\text{noise}}$ when $t = 0$. We sample t from a logit-normal distribution [15], $\text{logit}(t) \sim \mathcal{N}(\mu, \sigma^2)$, which concentrates training on intermediate noise levels where the velocity field is hardest to learn.

The flow velocity $\mathbf{v}$ is defined as the time-derivative of $\mathbf{z}_t$:

$$\mathbf{v} = \frac{d\mathbf{z}_t}{dt} = \mathbf{x} - \boldsymbol{\epsilon}. \tag{2}$$

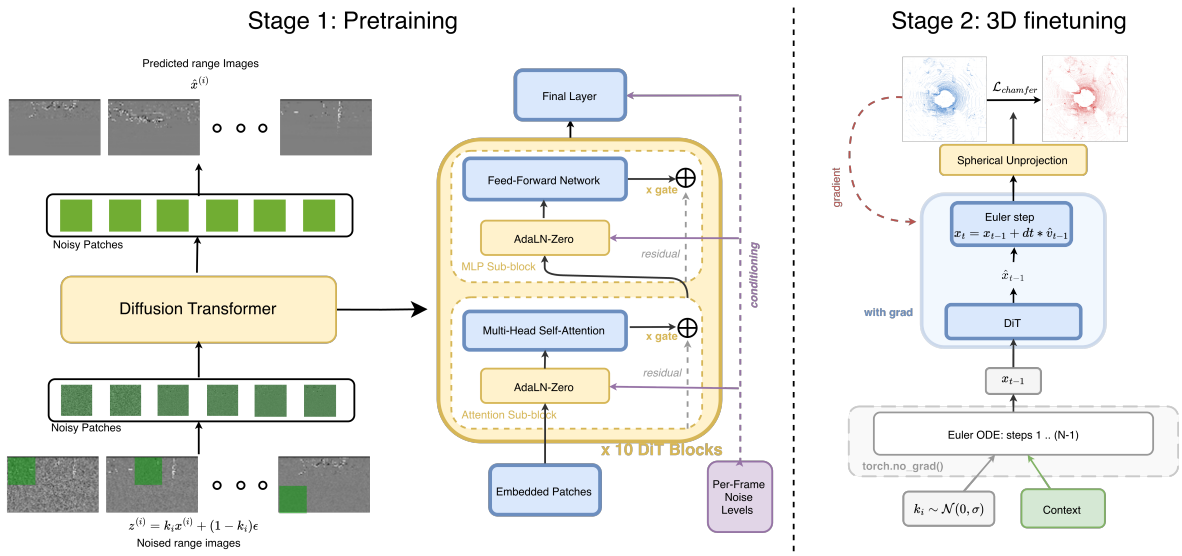

Figure 2. **Overview of the proposed two-stage pipeline for LiDAR point cloud forecasting.** Our architecture (center) utilizes a Diffusion Transformer (DiT) backbone with 10 layers, conditioned on noise levels via AdaLN-Zero. **Stage 1 (Pre-training)**: The model learns to denoise range image samples across varying noise schedules (Eq. 8). **Stage 2 (Fine-tuning)**: The pipeline incorporates a differentiable Euler sampling transition (right) supervised by a 3D Chamfer loss (Eq. 10), facilitating end-to-end optimization for geometrically consistent 3D reconstruction.

A network $\mathbf{v}_\theta$ is trained to predict this velocity by minimising:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,\,\mathbf{x},\,\boldsymbol{\epsilon}}\left[\left\|\mathbf{v}_\theta(\mathbf{z}_t,\,t) - (\mathbf{x}-\boldsymbol{\epsilon})\right\|_2^2\right]. \tag{3}$$

Sampling is performed by solving the ODE:

$$\frac{d\mathbf{z}_t}{dt} = \mathbf{v}_\theta(\mathbf{z}_t,\,t), \tag{4}$$

starting from $\mathbf{z}_0 \sim p_{\text{noise}}$ and integrating to $t = 1$

3.2. Autoregressive Diffusion

Standard diffusion models apply a uniform noise level across all tokens in a sequence. Diffusion Forcing [10] relaxes this by assigning *independent* noise levels $k_{1:T} \in [0,1]^T$ to individual tokens. Each token is noised independently via Eq. (1):

$$\mathbf{z}_t^{(i)} = k_i\,\mathbf{x}^{(i)} + (1-k_i)\,\boldsymbol{\epsilon}^{(i)}, \quad \boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0},\mathbf{I}), \tag{5}$$

and a single network is trained to denoise any combination of noise levels simultaneously:

$$\mathcal{L}_{\text{DF}} = \mathbb{E}_{k_{1:T},\,\mathbf{x}_{1:T},\,\boldsymbol{\epsilon}_{1:T}}\Big[\frac{1}{T}\sum_{i=1}^{T}\left\|\mathbf{v}_\theta(\mathbf{z}_{1:T},\,k_{1:T})^{(i)} - (\mathbf{x}^{(i)} - \boldsymbol{\epsilon}^{(i)})\right\|_2^2. \tag{6}$$

This enables flexible conditioning at inference: any subset of tokens can be pinned at $k_i = 1$ (fully clean) while others are initialized at $k_i = 0$ and refined via Eq. (4), with no architectural difference between training and inference. The Diffusion Forcing Transformer (DFoT) [9] scales this to high-resolution sequences via a transformer backbone, injecting per-token noise levels through AdaLN-Zero modulation [32].

4. Method

We present **Li-AutoFlow** (Fig 2, an autoregressive generative model for LiDAR scene forecasting that operates directly on full-resolution range difference images. Given a reference range image $\mathbf{r}_0$ and T_c past range differences $\Delta\mathbf{r}_{1:T_c}$, the model predicts K future differences $\Delta\mathbf{r}_{T_c+1:T_c+K}$ using a flow-matching generative process. Future range images are then recovered through a differentiable reconstruction step.

Our model is implemented using a Diffusion Forcing Transformer (DFoT) that jointly models spatial geometry within each frame and temporal dynamics across frames. Unlike prior approaches based on latent diffusion [52], we operate directly in the native sensor resolution, preserving the one-to-one correspondence between pixels and LiDAR measurements. Training proceeds in two stages: we first pretrain the model on clean next-frame predictions using the flow-matching objective, then fine-tune on a 3D-reprojection loss to improve geometric fidelity in the output point cloud.

4.1. Preprocessing

Each LiDAR sweep $\mathcal{P}_k \subset \mathbb{R}^3$ is projected onto a spherical range image $\mathbf{r}_k \in \mathbb{R}^{H\times W}$, where pixel (i,j) stores the

Euclidean range $r_{ij} = \|\mathbf{p}_{ij}\|_2$. Pixels with no LiDAR return are marked invalid and assigned $r_{ij} = 0$. To align frames, we warp $\mathbf{r}_k$ into the coordinate frame of the reference image $\mathbf{r}_0$ using the relative pose $\mathbf{T}_{0\leftarrow k}$, yielding $\mathbf{r}_k^{(0)} = \mathrm{Warp}(\mathbf{r}_k, \mathbf{T}_{0\leftarrow k})$. Rather than modeling raw range images directly, we operate on the *range difference*

$$x_t = \Delta\mathbf{r}_k = \mathbf{r}_k^{(0)} - \mathbf{r}_0, \quad k = 1, \ldots, N, \tag{7}$$

which removes static background and concentrates signal on dynamic objects. The resulting representation is sparse and near-zero for the majority of pixels, substantially reducing the complexity of the generative manifold compared to modeling raw range images directly.

4.2. Full-Resolution Autoregressive Flow Matching

Each pixel of a range image corresponds to a physically grounded depth measurement: its value determines the 3D location of a LiDAR return through the spherical projection. Spatial compression disrupts this one-to-one correspondence. When convolutional VAEs downsample the range image, neighbouring laser beams are merged into shared latent cells, erasing fine angular structures such as sharp object boundaries, thin objects, and precise range discontinuities. To preserve this geometric fidelity, we perform generative modelling directly in the full-resolution range difference space $\mathbf{x}_k \in \mathbb{R}^{H\times W}$.

We learn this distribution using flow matching. For a ground-truth range difference frame $\mathbf{x}_k$, we construct noisy interpolants

$$\mathbf{z}_t = t\mathbf{x}_k + (1-t)\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

and train a denoising network D_θ to predict the clean signal $\hat{\mathbf{x}}^{(i)}$. This x-prediction parameterization provides a well-conditioned regression target across the noise spectrum, avoiding the large target variance encountered by noise- or velocity-prediction objectives in high-dimensional pixel spaces [22]. The loss however, is evaluated in velocity space, which proves to be a more favorable learnable objective [22]. Training then minimizes the velocity-space regression error

$$\mathcal{L} = \mathbb{E}\left[\frac{1}{K}\sum_{i=T_c+1}^{T_c+K}\left\|\hat{\mathbf{v}}^{(i)} - \mathbf{v}^{(i)}\right\|_2^2\right], \tag{8}$$

where the loss is applied only to future frames $i > T_c$.

Operating at full resolution remains tractable due to the properties of the range difference representation. Subtracting a reference frame suppresses the static background, producing sparse signals that concentrate on dynamic scene elements. The resulting generative manifold is therefore substantially simpler than that of raw range images, enabling stable training without a latent compression bottleneck.

Pacification. To process frames with a transformer backbone, each $\mathbf{x}_k \in \mathbb{R}^{H\times W}$ is partitioned into non-overlapping $p \times p$ patches and linearly projected into C-dimensional tokens, yielding $\frac{H}{p} \times \frac{W}{p}$ tokens per frame. This patch projection reorganizes the computation for transformer processing but does not alter the underlying signal resolution, and thus preserves the full geometric fidelity of the range image.

Backbone. We instantiate the denoiser D_θ using a Diffusion Forcing Transformer (DFoT) [9]. The model processes frame-token sequences through alternating spatial and temporal attention layers. Spatial attention operates within each frame, capturing local geometric structure and the relationships between neighbouring laser beams. Temporal attention propagates information across frames, enabling the model to track the motion of surfaces over time. Causality is enforced by masking future frames during temporal attention so that each generated frame conditions only on its preceding history. This factorized design scales more efficiently with sequence length than full spatiotemporal attention while explicitly separating spatial coherence from temporal dynamics.

Per-frame noise conditioning. Diffusion Forcing assigns independent noise levels to different frames within the same forward pass. Context frames are fixed at $k_i = 1$ (clean), while future frames are sampled with $k_i \in [0, 1)$ from the training noise schedule. Each frame's noise level is encoded using a sinusoidal embedding and injected into every attention block via AdaLN-Zero modulation [32]. This conditioning provides the network explicit awareness of each frame's noise state, allowing it to exploit clean context frames while denoising noisy future frames during autoregressive generation.

4.3. 3D Geometric Fine-Tuning

The flow matching stage optimises a purely 2D surrogate objective: the network receives no direct signal about the 3D geometry its predictions imply. Since $\hat{\mathbf{x}}_k$ lives in full-resolution range image space, differentiable reconstruction to $\hat{\mathcal{P}}_k$ is exact and requires no decoder—we exploit this to introduce a fine-tuning stage that closes the supervision gap by minimising Chamfer distance between predicted and ground-truth point clouds.

Given $\hat{\mathbf{x}}_k$, we recover $\hat{\mathcal{P}}_k$ by applying Eq. (7) to obtain $\hat{\mathbf{r}}_k^{(0)}$, then unprojecting and applying $\hat{\mathbf{T}}_{k\leftarrow 0}$. Now, backpropagating through all S ODE steps is memory-prohibitive. We instead run $S-1$ steps under `torch.no_grad()`, detach the penultimate state $\bar{\mathbf{z}} = \mathrm{sg}(\mathbf{z}_{t_{S-1}})$, and perform a single gradient-enabled final step:

$$\hat{\mathbf{x}}_k = \bar{\mathbf{z}} + \Delta t \cdot \mathbf{v}_\theta(\bar{\mathbf{z}},\, t_{S-1},\, \mathbf{c})\,, \tag{9}$$

so that gradients from $\mathcal{L}_{3D}$ flow into $\mathbf{v}_\theta$ through exactly one denoising step, keeping memory overhead constant in S.

The fine-tuning objective combines the Chamfer distance with an invalid-pixel regularizer:

$$\mathcal{L}_{3D} = \mathrm{CD}(\hat{\mathcal{P}}_k, \mathcal{P}_k) + \lambda_{\text{inv}}\, \mathcal{L}_{\text{inv}}, \tag{10}$$

where the Chamfer distance is:

$$\mathrm{CD}(\hat{\mathcal{P}}_k, \mathcal{P}_k) = \frac{1}{|\hat{\mathcal{P}}_k|} \sum_{\hat{\mathbf{p}} \in \hat{\mathcal{P}}_k} \min_{\mathbf{p} \in \mathcal{P}_k} \|\hat{\mathbf{p}} - \mathbf{p}\|_2 + \frac{1}{|\mathcal{P}_k|} \sum_{\mathbf{p} \in \mathcal{P}_k} \min_{\hat{\mathbf{p}} \in \hat{\mathcal{P}}_k} \|\mathbf{p} - \hat{\mathbf{p}}\|_2, \tag{11}$$

and $\mathcal{L}_{\text{inv}}$ penalises non-zero predictions at pixels where the ground-truth has no valid return:

$$\mathcal{L}_{\text{inv}} = \frac{1}{|\mathcal{I}|} \sum_{(i,j) \in \mathcal{I}} H_\delta(\hat{r}_{ij}), \qquad \mathcal{I} = \{(i,j) : r_{ij} = 0\}, \tag{12}$$

where H_δ is the Huber loss,

$$H_\delta(x) = \begin{cases} \frac{1}{2}x^2, & |x| \le \delta, \\ \delta\left(|x| - \frac{1}{2}\delta\right), & |x| > \delta, \end{cases} \tag{13}$$

suppressing hallucinated returns in occluded and empty regions that Chamfer distance alone does not penalise. Unlike a squared penalty, the linear tail of H_δ prevents the loss from being dominated by the few pixels with large spurious predictions, while still providing a smooth L_2 signal near zero.

4.4. Implementation Details

Architecture. The DFoT backbone uses hidden dimension 384, patch size 8×8, depth 8, and 6 attention heads with MLP ratio 4. Temporal position embeddings are learned 1D embeddings along the sequence axis.

Datasets and splits. We evaluate on KITTI Odometry [17], using sequences 00–05 for training, 06–07 for validation, and 08–10 for testing, and on nuScenes [7]. Range images are at resolution 64×2048. We use $T_c = 2$ context frames and $K = 3$ future frames.

Training. The flow model is optimised with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0) at learning rate 10^{-3} with gradient norm clipping at 1.0. The flow model is trained for 300k steps with batch size 8 on L40s GPUs. The timestep sampling follows a logit norm based noise schedule [22], $k_i \sim \mathcal{N}(\mu_p, \sigma_p^2)$, which biases training toward intermediate noise levels. 3D fine-tuning runs for up to 10k steps at learning rate 10^{-4} with $S = 5$ ODE steps with 5 steps of ODE iterations and $\lambda_{\text{inv}} = 0.1$.

Autoregressive inference. During inference, the model generates K future frames jointly within a single flow trajectory. The context frames are fixed at $k_i = 1$ and future frames are initialised from standard Gaussian noise at $k_i = 0$. The probability flow ODE of Eq. (4) is then integrated from $t = 0$ to $t = 1$ using an ODE solver num of samples = 5], producing the predicted range differences $\widehat{\mathbf{x}}_{T_c+1:T_c+K}$.

5. Experiments

5.1. Evaluation

Dataset. We evaluate on the KITTI Odometry benchmark [17], a standard testbed for LiDAR-based autonomous driving research. The dataset provides 22 sequences of outdoor driving data captured with a Velodyne HDL-64E at 10 Hz. We follow the standard split used in prior forecasting work [28, 29, 31]: sequences 00–07 for training and 08-10 for testing.

Metrics. We evaluate performance using two complementary metrics over the $K = 3$ predicted future steps. *Chamfer Distance (CD)* measures the mean bidirectional nearest-neighbour distance between predicted and ground-truth 3D point clouds (Eq. 11). We report both per-step values and the mean across the prediction horizon; lower values indicate better geometric accuracy. *Range Image Loss (L1)* computes the mean absolute error between predicted and ground-truth range images, Lower values indicate improved reconstruction fidelity.

5.2. Baselines

We compare against three range-image-based point cloud forecasting methods that span the major architectural families. **Mersch et al. [29]** project each LiDAR scan to a range image and stack consecutive frames into a 3D tensor, processing it with an encoder-decoder of 3D spatio-temporal convolutions in a fully self-supervised manner, establishing the foundational range-image baseline.

PCPNet [28] replaces convolutions with a Transformer that operates on range image tokens compressed along both spatial dimensions, and augments training with a semantic consistency loss derived from SemanticKITTI labels to improve downstream utility.

ATPPNet [31] processes range image sequences through Conv-LSTM blocks gated by dual channel-wise and spatial attention, complemented by a parallel 3D-CNN branch for global spatio-temporal context, yielding the strongest recurrent baseline.

CoPilot4D [52] is a generative world model that first tokenizes point clouds into discrete BEV tokens via a VQ-VAE, then forecasts future tokens using a masked discrete diffusion Transformer, enabling scalable unsupervised training

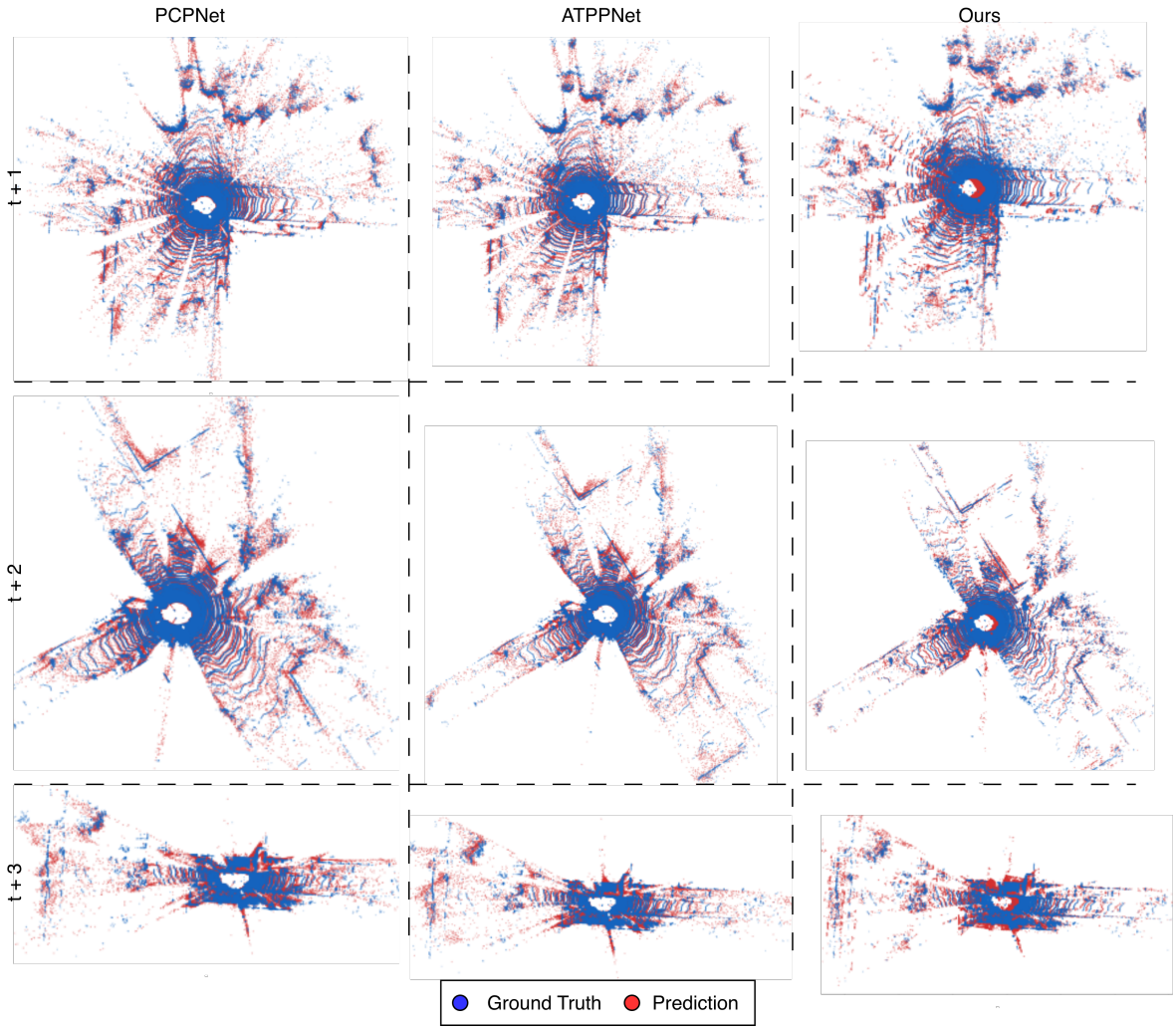

Figure 3. Qualitative comparison of predicted (red) versus ground truth (blue) point clouds at horizons $t{+}1$, $t{+}2$, and $t{+}3$ (rows) for PCPNet [28], ATPPNet [31], and our method (columns, left to right).

across multiple datasets. *As the code is not publicly available, we report the mean Chamfer distance for the 1-second prediction horizon directly from their paper.

5.3. Quantitative Results

Table 1 compares our method with prior point cloud forecasting approaches on KITTI Odometry. Our approach consistently outperforms all baselines across all future horizons. At $t{+}1$, our model achieves a CD of 0.087, improving over PCPNet (0.280) and ATPPNet (0.221) by **68.9%** and **60.6%**, respectively. At $t{+}2$, the gap further widens, where our model obtains 0.117, corresponding to **65.6%** and **57.3%** improvements over PCPNet (0.340) and ATPPNet (0.274). At $t{+}3$, our method achieves 0.155, reducing the error by **62.4%** and **54.9%** compared to PCPNet (0.412) and ATPPNet (0.344). Against CoPilot4D, a strong generative baseline based on discrete diffusion, our method achieves a mean CD of 0.120 compared to their reported mean of 0.180, a **33.3%** improvement, despite CoPilot4D operating on tokenized BEV representations trained across multiple large-scale datasets. Overall, our approach substantially outperforms all prior methods while maintaining stable performance as the prediction horizon increases.

5.4. Qualitative Analysis

Figure 3 shows predicted range images and their corresponding 3D back-projections for representative test sequences. Li-AutoFlow produces sharper object boundaries and more coherent predictions of dynamic objects (e.g., vehicles and cyclists) compared to baseline methods, which tend to blur high-motion regions due to the averaging effects of deterministic regression. Our method preserves fine local geometric

Table 1. Comparison of point cloud forecasting performance on KITTI Odometry (sequences 08–10). † denotes our method

Method	Chamfer Distance (m) ↓			
	t+1	t+2	t+3	Mean
PCPNet [28]	0.280	0.340	0.412	0.344
ATPPNet [31]	0.221	0.274	0.344	0.280
Copilot4D* [52]	-	-	-	0.180
Li-AutoFlow (ours)†	**0.087**	**0.117**	**0.155**	**0.120**
w/o 3D fine-tuning	0.177	0.234	0.271	0.234

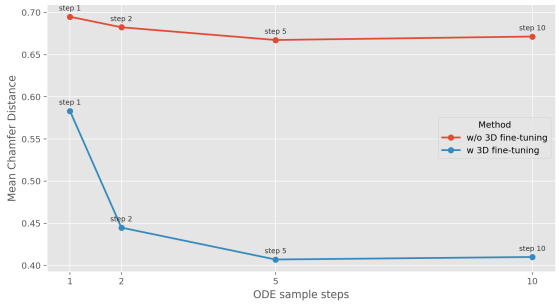

Figure 4. Figure show the Chamfer distance with varying number of ODE sampling steps. Results are computed on a single test sequence.

changes and demonstrates improved understanding of scene dynamics, resulting in more accurate motion structures in the predicted frames. Furthermore, the stochastic nature of the flow model enables the generation of geometrically plausible future states, particularly around occluded or uncertain regions.

5.5. Ablation Studies

Effect of chamfer distance fine-tuning Table 1 shows fine-tuning with a 3D Chamfer distance objective yields substantial improvements across all prediction horizons, reducing mean CD from 0.227 to 0.120, a relative improvement of over 47%. Qualitatively, the fine-tuned model produces significantly fewer outlier points, with predictions adhering much more tightly to planar surfaces such as roads and building facades. Without fine-tuning, stray points are scattered around object boundaries and flat regions, whereas fine-tuning encourages the predicted geometry to collapse onto the correct surfaces. The gains are consistent across time steps, with CD improving from 0.177 to 0.087 at t+1 and from 0.271 to 0.155 at t+3, suggesting that 3D supervision corrects systematic geometric errors rather than simply reducing noise at a single horizon.

Number of ODE sampling steps. In Figure 4 we evaluate generation quality as a function of the number of ODE integration steps (1, 2, 5, 10) with and without 3D fine-tuning, reporting Mean Chamfer Distance in Figure 4. Without 3D fine-tuning, performance improves only marginally as the number of steps increases, plateauing beyond step 5 with consistently high CD throughout. With 3D fine-tuning, quality improves sharply between steps 1 and 5, after which gains become negligible. This suggests that 5 ODE steps strike a favorable balance between inference speed and prediction accuracy, and that 3D fine-tuning is the dominant factor in performance improvement rather than increasing the number of sampling steps.

6. Conclusion

In this work, we introduced Li-AutoFlow, a LiDAR scene forecasting framework designed to bypass the quantization artifacts and information loss typical of discrete latent-space architectures. By formulating the prediction task as a flow-matching process directly on the full-resolution range-image manifold, we maintain the inherent sensor topology and metric depth necessary for high-fidelity 3D reconstruction. Our experiments demonstrate that a tokenizer-free, Diffusion Forcing Transformer can effectively capture complex scene dynamics, while the fully differentiable nature of our pipeline allows for direct optimization via a 3D Chamfer distance (CD) objective. This dense geometric supervision results in a 47% improvement in CD over baseline models that lack 3D-aware fine-tuning, highlighting the importance of alignment between the generative space and the physical 3D world.

Limitations and future work. The current system relies on ground-truth odometry during training (KITTI poses) and a pose estimator at test time, introducing an external dependency that should ideally be removed by jointly learning ego-motion estimation directly from raw scans. Future work will transition to a fully end-to-end architecture via joint odometry learning directly from raw scans, while adopting consistency distillation to collapse the sampling process into 1–2 steps without sacrificing structural fidelity. Finally, extending evaluation to nuScenes [7] and Waymo [49] would strengthen generalization claims beyond the KITTI odometry benchmark. We believe continuous-space autoregressive generation on raw sensor data represents a promising direction for scalable world modeling in autonomous systems. The absence of a tokenization bottleneck makes this class of architectures well-suited for large-scale pre-training on unlabeled LiDAR corpora, with subsequent 3D fine-tuning for safety-critical multi-step forecasting.

References

[1] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023. 3

[2] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. 2, 3

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2

[4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3

[5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 2

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. 3

[7] Holger Caesar et al. nuScenes: A multimodal dataset for autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 6, 8

[8] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[9] Boyuan Chen et al. Diffusion forcing transformer for long-horizon video generation. *arXiv preprint*, 2024. 3, 4, 5

[10] Boyuan Chen et al. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024. 2, 3, 4

[11] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds. In *Int. Symp. Visual Computing*, 2020. 2

[12] Soham Dasgupta, Kshitij Aphale, Kaustab Pal, and Avinash Sharma. Accurate and real-time lidar point cloud forecasting for autonomous driving. New York, NY, USA, 2025. Association for Computing Machinery. 2

[13] Jue Dong et al. Self-supervised point cloud prediction for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 25(2), 2024. 2

[14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 3

[15] Patrick Esser, Sumith Kulal, Andreas Blattmann, Robin Rombach, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024. 2, 3

[16] Hehe Fan and Yi Yang. PointRNN: Point recurrent neural network for moving point cloud processing. In *arXiv preprint arXiv:1910.08287*, 2019. 2

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012. 6

[18] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.*, 2020. 3

[20] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4d occupancy forecasting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[21] Alex H Lang, Sourabh Vora, Holger Caesar, Lubomir Lublin, Oscar Spong, Susien Peters, and Juan Nuñez-Iglesias. PointPillars: Fast encoders for object detection from point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[22] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. 2, 3, 5, 6

[23] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3

[24] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Int. Conf. Learn. Represent.*, 2023. 2, 3

[25] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. 3

[26] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. 2, 3

[27] Fan Lu, Guang Chen, Zhijun Li, Lijun Zhang, Yinlong Liu, Sanqing Qu, and Alois Knoll. MoNet: Motion-based point cloud prediction network. *IEEE Trans. Intell. Transp. Syst.*, 23(8):13794–13804, 2021. 2

[28] Zhen Luo, Junyi Ma, Zijie Zhou, and Guangming Xiong. PCPNet: An efficient and semantic-enhanced transformer network for point cloud prediction. *IEEE Robot. Autom. Lett.*, 8(7):4267–4274, 2023. 2, 6, 7, 8

[29] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conf. on Robot Learning*, 2022. 2, 6

[30] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet++: Fast and accurate LiDAR semantic seg-

mentation. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2019. 2

[31] Kaustab Pal, Aditya Sharma, Avinash Sharma, and K Madhava Krishna. ATPPNet: Attention based temporal point cloud prediction network. *arXiv preprint arXiv:2401.17399*, 2024. 2, 6, 7, 8

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023. 4, 5

[33] Adam Polyak, Amit Zohar, Andrew Brown, Andros Swaminathan, Oran Metzer, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 3

[34] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2021. 3

[35] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

[36] Nicholas Rhinehart, Kris M. Kitani, and Paul Vernaza. R2p2: A ReparameteRized pushforward policy for diverse, precise generative path forecasting. In *European Conference on Computer Vision*, 2018. 2

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ömmer. High-resolution image synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 3

[38] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion, 2025. 2

[39] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Int. Conf. Learn. Represent.*, 2021. 3

[40] Yue Song et al. PCPMamba: Point cloud prediction with mamba-based state space models. *arXiv preprint*, 2025. 2

[41] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. Best Paper Award. 3

[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[43] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[44] 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. 2

[45] Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.01502*, 2025. 3

[46] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Inverting the pose forecasting pipeline with SPF2: Sequential pointcloud forecasting for sequential pose forecasting. In *Conf. on Robot Learning*, 2021. 2

[47] Xinshuo Weng, Junyu Nan, Kuan-Hui Lee, Rowan McAllister, Adrien Gaidon, Nicholas Rhinehart, and Kris M Kitani. S2Net: Stochastic sequential pointcloud forecasting. In *Eur. Conf. Comput. Vis.*, 2022. 2

[48] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893, 2018. 1

[49] Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. *arXiv preprint arXiv:2510.26125*, 2025. 8

[50] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 2

[51] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3

[52] Anthony Zhang et al. Copilot4D: Learning unsupervised world models for autonomous driving via discrete diffusion, 2023. 2, 3, 4, 6, 8

[53] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3d object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2