Adaptive LiDAR Scanning: Harnessing Temporal Cues for Efficient 3D Object Detection via Multi-Modal Fusion

Sara Shoouri, Morteza Tavakoli Taba, Hun-Seok Kim
University of Michigan
{sshoouri, tmorteza, hunseok}@umich.edu

Abstract

Multi-sensor fusion using LiDAR and RGB cameras significantly enhances 3D object detection task. However, conventional LiDAR sensors perform dense, stateless scans, ignoring the strong temporal continuity in real-world scenes. This leads to substantial sensing redundancy and excessive power consumption, limiting their practicality on resource-constrained platforms. To address this inefficiency, we propose a predictive, history-aware adaptive scanning framework that anticipates informative regions of interest (ROI) based on past observations. Our approach introduces a lightweight predictor network that distills historical spatial and temporal contexts into refined query embeddings. These embeddings guide a differentiable Mask Generator network, which leverages Gumbel-Softmax sampling to produce binary masks identifying critical ROIs for the upcoming frame. Our method significantly reduces unnecessary data acquisition by concentrating dense LiDAR scanning only within these ROIs and sparsely sampling elsewhere. Experiments on nuScenes and Lyft benchmarks demonstrate that our adaptive scanning strategy reduces LiDAR energy consumption by over 65% while maintaining competitive or even superior 3D object detection performance compared to traditional LiDAR-camera fusion methods with dense LiDAR scanning.

Code — https://github.com/sarashoouri/AdaptiveLiDAR
Extended version — https://arxiv.org/abs/2508.01562

Introduction

Multisensor fusion for 3D object detection leverages complementary camera–LiDAR strengths to improve perception reliability in autonomous driving (Yan et al. 2023; Bai et al. 2022; Li et al. 2022; Liu et al. 2023; Chen et al. 2023). Cameras capture rich semantic cues, such as color and texture, while LiDAR provides spatial information through direct depth measurements. By fusing these modalities, autonomous systems significantly improve their reliability to handle challenging scenarios (Bai et al. 2022).

Despite the sophistication achieved in modern fusion techniques, fundamental inefficiency persists at the point of LiDAR data acquisition. Conventional LiDAR sensors operate in a 'stateless' or 'memoryless' manner, performing dense uniform-angle scans at each frame as if observing the scene for the very first time. This approach ignores the strong temporal continuity inherent in real-world environments, where the world does not completely rearrange itself every tenth of a second. For instance, studies in data compression demonstrate that the static background is highly predictable from prior frames. Consequently, only a small fraction of the points corresponding to dynamic objects require explicit updates (Feng, Liu, and Zhu 2020). By repeatedly re-scanning unchanged areas with full resolution, LiDAR wastes a significant portion of its energy on low-value measurements. Such redundancy imposes a substantial burden on the sensing system's power budget, a critical constraint for small form factor integration. This power demand is especially evident when comparing LiDAR with other passive sensors: a typical automotive camera requires only 1–$5W$ per unit for capture and preprocessing (Sahin 2019; Ambarella, Inc. 2019), whereas a LiDAR, as an active sensor generating beam-steered laser pulses, requires significantly more power in the range of $10 - 100W$ (Velodyne Lidar, Inc. 2018a,b). Laser emission is particularly power-demanding as each pulse must be strong enough to reach a target and return with sufficient intensity to be detected. Extending detection range forces pulse energy to rise steeply as return strength decays roughly with the fourth power of distance. Moreover, achieving finer angular resolution requires larger optics or higher launch power (Lee et al. 2020; Raj et al. 2020; Tayebati, Tulabandhula, and Trivedi 2025). In real-world deployments, these factors translate into substantial per-scan energy demands. The Velodyne HDL-32E Li-

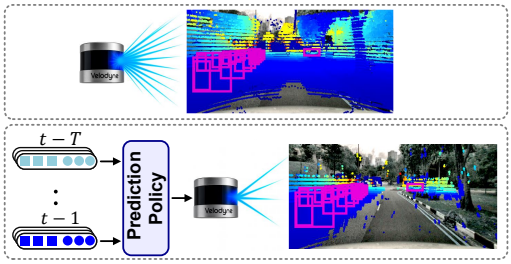

Figure 1: Conventional uniform LiDAR scanning (top) vs. our adaptive scanning (bottom), which leverages past frames to densify predicted ROIs and sparsify non-ROI areas.

DAR used for the nuScenes dataset (Caesar et al. 2020) consumes roughly $12W$, translating to approximately 0.6 J per full scan at 20 Hz (Velodyne Lidar, Inc. 2018a), whereas the Velodyne HDL-64E used in Lyft dataset (Kesten et al. 2019; Houston et al. 2021) draws about $60W$, or approximately 6 J per full scan at 10 Hz (Velodyne Lidar, Inc. 2018b). Consequently, sustaining such high power at typical frame rates is prohibitive for power-constrained sensing platforms.

These limitations demand a paradigm shift from fixed-resolution scanning to a predictive, history-aware adaptive sensing strategy. We propose that an intelligent system can anticipate regions of interest (ROI) by leveraging the memory from recent past observations. Rather than treating each frame in isolation, our method leverages a sequence of learned historical query embeddings, which encode recent spatial and temporal context, to forecast where the most salient information (or ROIs) will be in the subsequent frame. This allows the system to proactively allocate its resources for intelligent adaptive data acquisition.

To implement this concept, we introduce a two-stage adaptive scanning LiDAR and camera fusion pipeline. First, a lightweight predictor ingests historical object queries and produces refined embeddings Q' that anticipate the spatial distribution of dynamic actors in the upcoming frame. Second, these predictions are fed into a differentiable Mask Generator, which produces a binary mask via the Gumbel-Softmax trick (Jang, Gu, and Poole 2016) over the LiDAR's field of view. This mask defines the critical ROIs. The final scan pattern is dictated by this mask, activating dense LiDAR scanning within ROIs only while scanning sparsely elsewhere. By complementing this sparse LiDAR sampling with surround-view RGB camera images, we preserve detection performance even in areas with sparse LiDAR scans.

We introduce two key techniques to enable end-to-end optimization of this pipeline. First, a differentiable voxelization method overcomes the limitations of traditional voxelization, which is inherently non-differentiable and blocks gradient flow. Our approach provides heuristic gradients that map voxel-level losses back to the point-level inputs, enabling end-to-end training of the entire system with adaptive non-uniform scanning. Second, we further employ a Conditional Value-at-Risk (CVaR) loss (Rockafellar, Uryasev et al. 2000) to train the scanning Mask Generator. This risk-averse objective forces the model to prioritize accurate mask generation even for small yet critical objects like pedestrians, ensuring the system remains robust in high-risk scenarios.

Our contributions can be summarized as follows:

- We introduce a history-driven adaptive LiDAR paradigm that departs from traditional memoryless acquisition, using past observations to forecast future ROI positions for dynamic sensing resource allocation.
- We propose a transformer-based LiDAR-camera fusion model integrating a history-aware Query Predictor module and a differentiable Mask Generator network trained with a risk-averse CVaR, effectively converting predictions into adaptive spatial scanning patterns.
- We facilitate a custom differentiable voxelization layer to enable end-to-end training of our adaptive sensing model. On nuScenes and Lyft, we reduce LiDAR scanning density by over 65% while matching or surpassing dense-scan 3D detectors.

Related Work

Camera and LiDAR Fusion for 3D Detection: Multi-modal camera–LiDAR fusion leverages complementary sensing capabilities, initially through feature-branch fusion networks (Chen et al. 2017; Ku et al. 2018) or point painting (Vora et al. 2020). This evolved into transformer-based models that learn cross-modal interactions by aligning BEV features (Liu et al. 2023) or extending query-based fusion from 2D detectors like DETR (Carion et al. 2020) to 3D perception (Bai et al. 2022; Wang et al. 2022; Chen et al. 2023). Most recently, CMT (Yan et al. 2023) directly encodes image and point-cloud tokens in a cross-modal transformer to achieve implicit alignment. However, these approaches assume dense LiDAR scanning at every frame, failing to exploit temporal cues that could reduce redundant sensing and power consumption.

Efficient and Adaptive Inference: Modern perception systems increasingly adopt adaptive inference to allocate computation where it matters most and avoid redundancy. At the network level, early exiting (Bolukbasi et al. 2017; Huang et al. 2017), dynamic layer skipping (Veit and Belongie 2018; Wang et al. 2018), and channel pruning (Hua et al. 2019; Yuan et al. 2020) adjust model depth or width based on input difficulty, while adaptive multi-task architectures further improve efficiency through computation sharing (Shoouri et al. 2023). For sequential inputs, recurrent policies select only informative frames (Wu et al. 2019; Panda et al. 2021; Meng et al. 2021), and image resolution can be adapted dynamically (Jang, Gu, and Poole 2016). Building upon these, our framework introduces adaptive inference at the sensor level, where LiDAR scanning density for (non-)ROIs is dynamically adjusted using historical scene context.

Point Cloud Down-Sampling: Managing the large data from LiDAR is a core challenge in 3D perception, making down-sampling essential for efficiency. Traditional geometric methods (Cheng et al. 2022; Nezhadarya et al. 2020; Wen, Yu, and Tao 2023) use voxel grids (Zhou and Tuzel 2018) or Farthest Point Sampling (Qi et al. 2017) to retain spatial coverage, while learned sampling (Lang, Manor, and Avidan 2020) and content-adaptive compression (Liu et al. 2025) further reduce point-cloud size. All of these methods operate *after* a full scan is collected, still wasting sensing energy. In contrast, our work controls the LiDAR scan density *before* capture to conserve sensing energy.

Proposed Method

In this section, we present the adaptive LiDAR-camera fusion model, as illustrated in Figure 2 (a).

Framework Overview

In the standard sensor fusion for 3D object detection, a LiDAR and synchronized cameras capture scenes at discrete

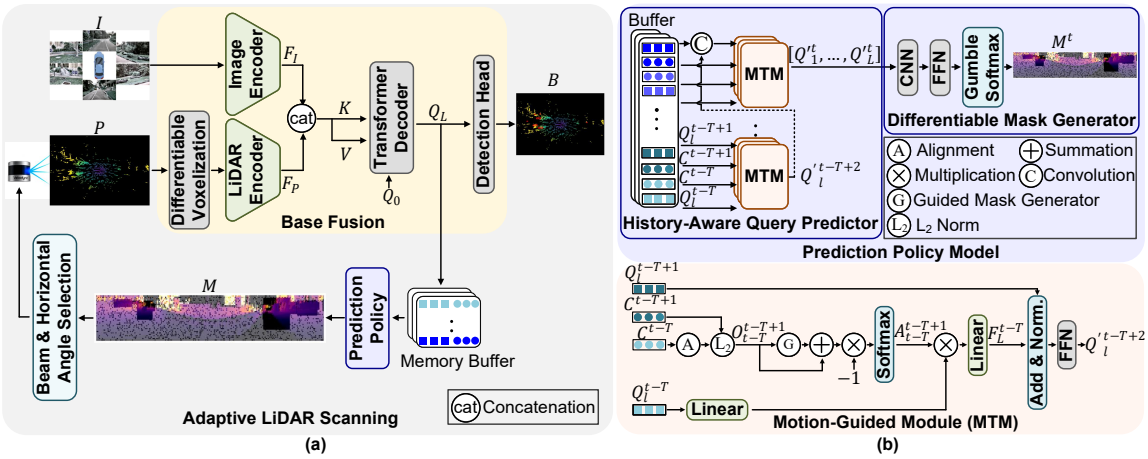

Figure 2: (a) Adaptive LiDAR scanning pipeline: historical data is leveraged to predict ROI masks for dense scanning and sparse non-ROI sampling. (b) The Query Prediction module uses queries of past frames from a memory buffer to predict the query stacks at time t, which are fed to the Mask Generator to form the final ROI mask.

timesteps. Formally, at each timestep t, the LiDAR provides a dense point cloud, $P^t = \{\, p_i^t \in \mathbb{R}^{D_p} \mid i = 1, \ldots, N\}$, where each point p_i^t encodes a 3D coordinate and additional features (e.g. intensity), and N is the number of points. Simultaneously, the cameras capture a set of M multi-view images, $I^t = \{\, I_m^t \mid m = 1, \ldots, M\}$. Given these measurements, the objective is to produce a set of 3D bounding boxes, $\mathcal{B}^t = \{b_1^t, \ldots, b_B^t\}$, that accurately localize and classify objects in the scene. Each bounding box b_i^t is parameterized by 7 degrees of freedom (DoF), consisting of its 3D center $C^t = (x_i^t, y_i^t, z_i^t)$, dimensions (l_i^t, w_i^t, h_i^t), and yaw angle θ_i^t, along with an associated class label.

Without loss of generality, we adopt the standard multi-modal 3D detection architecture, as depicted in Figure 2 (a). At each timestep t, each camera image I_m^t is passed through a 2D image backbone to produce a per-view semantic features, $F_{\mathrm{I},m}^t \in \mathbb{R}^{C_\mathrm{I} \times H_\mathrm{I} \times W_\mathrm{I}}$, which are then aggregated into a unified high-level feature map, F_I^t. Similarly, the LiDAR point cloud P^t is fed into a LiDAR backbone to extract geometric features, F_P^t. These uni-modal features are then merged using a fusion module (either by simple concatenation or via a cross-modal attention layer) to form the unified multi-modal representation, F_{fuse}^t.

This fused feature F_{fuse}^t serves as the key (K) and value (V) in a multi-layer transformer decoder. The decoder operates on a fixed set of N_q learnable queries, $Q^t = \{\, q_i^t \in \mathbb{R}^D \mid i = 1, \ldots, N_q\}$, each representing an object 'slot'. At each decoder layer l, queries Q_l^t first perform multi-head self-attention and then attend to F_{fuse}^t using the standard attention mechanism, defined as: $Attention(Q, K, V) = softmax(QK^\top/\sqrt{D})V$. The output queries from the final layer, Q_L^t, are then passed to a detection head, composed of feed-forward networks (FFN), to decode each query into 3D bounding-box parameters.

LiDAR Scanning Model: Conventional LiDAR systems operate in a stateless fashion, emitting laser pulses uniformly across all vertical and horizontal angular directions to produce a dense, uniform-angle scan at each frame. This scanning pattern is determined by the sensor's angular resolution, with H elevation angles (vertical rows) and W azimuthal angles (horizontal columns). Each beam in angular coordinate (i, j) travels a range $r_{i,j}$, and the LiDAR measures it as the sensing data. In our adaptive LiDAR scan method, we assume that the system can enable/disable any beam with arbitrary elevation and azimuthal scan patterns.

Range Image Representation: We represent the LiDAR's acquisition pattern using a 2D format known as a *range image*. This representation is common in methods that process the LiDAR data using 2D convolutional models (Milioto et al. 2019). Formally, a range image $R^t \in \mathbb{R}^{H \times W}$ is a projection where each pixel corresponds to a unique beam direction defined by an azimuth angle θ and an elevation angle ϕ. The projection from a 3D point $p_i^t = (x_i, y_i, z_i)$ to its spherical coordinates (r_i, θ_i, ϕ_i) is defined as:

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2},\ \phi_i = \arcsin\!\left(\tfrac{z_i}{r_i}\right),\ \theta_i = \mathrm{atan2}(y_i, x_i). \quad (1)$$

These spherical coordinates are then discretized to map to pixel coordinates (u, v) based on the sensor's vertical and horizontal field-of-view, such that:

$$u = \lfloor (\phi_i - \phi_{\min})/\Delta\phi \rfloor, \quad v = \lfloor (\theta_i - \theta_{\min})/\Delta\theta \rfloor, \quad (2)$$

where $\Delta\phi = (\phi_{\max} - \phi_{\min})/H$ and $\Delta\theta = (\theta_{\max} - \theta_{\min})/W$. Each pixel (u, v) of a range image R^t records the minimum r_i of all LiDAR points that belong to that pixel coordinate. By representing the scan pattern as this range image, the problem of adaptive scanning is formulated as a task of learning a binary mask that dictates which range image pixels require LiDAR sensing at each timestep.

Two-Stage History-Aware Adaptive Scanning

To estimate the optimal scanning mask M^t, we propose a two-stage Prediction Policy model. It leverages temporal context to make informed predictive decisions, and it can be plugged into any multi-modal fusion 3D detectors. The first

stage, a *History-Aware Query Prediction* module, forecasts the state of objects in the upcoming frame solely based on past observations. The second stage, a *Differentiable Mask Generator*, then translates these object-centric predictions into a mask pattern in the range image space, as shown in Figure 2 (b).

History-Aware Query Prediction using Motion-guided Temporal Module: Our model predicts queries for the current timestep t, Q'^t, using a historical buffer containing object query sets from the past T frames, denoted $\{Q^{t-T}, \dots, Q^{t-1}\}$, along with their predicted 3D bounding box centers $\{C^{t-T}, \dots, C^{t-1}\}$ and velocities $\{V^{t-T}, \dots, V^{t-1}\}$. Our architecture builds upon the core principles of the Motion-guided Temporal Module (MTM) from QTNet (Hou et al. 2023). While the original MTM fuses past queries with current frame data to 'refine' current detections, we repurpose its core concept into an autoregressive framework for a purely predictive task where no current frame sensor data is used/available.

As illustrated in Figure 2 (b), for each layer $l \in \{1, \dots, L\}$ in the transformer decoder, we sequentially unroll $T-1$ previous time steps through the historical buffer. At each step τ (from $\tau = T$ down to 2), the module takes a pair of query embeddings $Q_l^{t-\tau}$ and $Q_l^{t-\tau+1}$ along with their corresponding bounding box centers $C^{t-\tau}$ and $C^{t-\tau+1}$ as input to predict the queries for the subsequent timestep. To ensure spatial consistency, we first align the bounding box center $C^{t-\tau}$ into the coordinate frame of time $t-\tau+1$. Using the known world-to-sensor rotations $R_w^{t-\tau+1}$ and $R_{t-\tau}$, we compute the relative rotation as $R_{t-\tau}^{t-\tau+1} = R_w^{t-\tau+1}(R_w^{t-\tau})^{-1}$. Applying a constant-velocity motion model, the aligned historical center is then calculated as $C'^{t-\tau} = (C^{t-\tau} + V^{t-\tau}\Delta t)(R_{t-\tau}^{t-\tau+1})^\top$, where Δt is the inter-frame interval.

Using the aligned centers, a geometry-guided attention map is generated to associate objects across time. First, a cost matrix is defined based on the pairwise L_2 norm (Euclidean distance): $O_{t-\tau}^{t-\tau+1} = ||C^{t-\tau+1} - C'^{t-\tau}||_2$. Next, we construct a guided mask $G_{t-\tau}^{t-\tau+1}$ to penalize improbable matches between different object classes or those separated by more than a distance threshold γ:

$$G_{t-\tau}^{t-\tau+1} = \begin{cases} 0, & O_{t-\tau}^{t-\tau+1} \le \gamma \; and \; s_{t-\tau} = s_{t-\tau+1} \\ c_m, & O_{t-\tau}^{t-\tau+1} > \gamma \; or \; s_{t-\tau} \neq s_{t-\tau+1}, \end{cases} \quad (3)$$

where s_t indicates the object category at time t and c_m is a large constant (e.g., 10^8). The attention map is then defined as $A_{t-\tau}^{t-\tau+1} = softmax(-O_{t-\tau}^{t-\tau+1} - G_{t-\tau}^{t-\tau+1})$, as shown in Figure 2 (b) (MTM module).

This attention map is used to aggregate historical query features into a context vector, formulated as $F_l^{t-\tau} = \Phi_2(A_{t-\tau}^{t-\tau+1}\Phi_1(Q_l^{t-\tau}))$, where Φ_1 and Φ_2 denote two linear layers. After aggregation, $F_l^{t-\tau}$ and $Q_l^{t-\tau+1}$ are combined to produce a provisional Query Prediction, $Q_l'^{t-\tau+2}$, calculated as c, where $Norm$ represents the layer normalization. This provisional query is the output of the MTM module (Figure 2 (b)).

The provisional $Q_l'^{t-\tau+2}$ is fused with its corresponding historical embedding ($Q_l^{t-\tau+2}$) by concatenating them and then applying a pointwise convolutional projection. This produces an updated $Q_l^{t-\tau+2}$, which serves as the input for the next iteration as shown in Figure 2 (b). This alignment-attention-aggregation-fusion procedure is repeated sequentially for each time step in the historical buffer at every decoder layer. Ultimately, this process yields the final predicted query set Q'^t for all decoder layers that carries both learned priors and motion-guided history.

Differentiable Mask Generator: To generate an adaptive LiDAR scan, we feed the predicted query stack from all L decoder layers, $Q'^t \in \mathbb{R}^{L \times N_q \times D}$, to the Differentiable Mask Generator. This module translates these object-centric representations into a LiDAR scanning policy represented by a spatial logits map, $Z^t \in \mathbb{R}^{H_B \times W_B \times 2}$, for the LiDAR range image view that consists of $H_B \times W_B$ 'block's. Each spatial location (u, v) corresponds to one angular block in the LiDAR's range image view and encodes the probability of performing either a 'full' (dense) or a 'sparse' scan for that block. Formally, the logits map is computed as $Z^t = \mathcal{F}_{\text{Head}}\big(\mathcal{F}_{\text{Enc}}(Q'^t)\big)$. The encoder $\mathcal{F}_{\text{Enc}}$ consists of depthwise convolution residual blocks that operate on each channel independently, followed by standard residual blocks, and a final 1×1 convolution to project those channels into two class logits (full vs. sparse scan). The subsequent module, $\mathcal{F}_{\text{Head}}$ applies an adaptive pooling layer and an FFN to generate the final logits. A binary scan mask, M^t, is then drawn from Z^t via the Gumbel-Softmax reparameterization (instead of non-differentiable Bernoulli draws) with a temperature (τ)-controlled softmax over Gumbel-perturbed logits: $M^t \sim \text{GumbelSoftmax}\big(Z^t; \tau\big)$. Specifically, for each block with spatial location (u, v), the probability of scanning mode $k \in \{full, sparse\}$ is obtained by:

$$\tilde{M}_k^t(u,v) = \frac{\exp\big((Z_k^t(u,v) + g_k)/\tau\big)}{\sum_{j \in \{full, sparse\}} \exp\big((Z_j^t(u,v) + g_j)/\tau\big)}, \quad (4)$$

where $g_k = -log(-logU_k)$ is a standard Gumbel distribution with $U \sim \text{Uniform}(0,1)$ and τ controls how sharply $\tilde{M}^t$ approximates a one-hot vector. The binary mask M^t is only used for forward pass while the soft probability $\tilde{M}^t$ is used to propagate gradients during training.

During inference, we generate the final adaptive LiDAR scanning pattern using a two-level sampling strategy. For blocks marked for full scans ($M^t(u,v) = 1$), we activate all LiDAR points. For blocks with sparse scans ($M^t(u,v) = 0$), we employ probabilistic subsampling based on the corresponding soft probability $\tilde{M}_{full}^t(u,v)$. This soft probability is first quantized to a predefined sparsity level (e.g., 6.25%, 12.5%, etc.). Then, a final sparse scanning pattern is generated by performing stochastic Bernoulli sampling for each beam within the block, using the quantized probability as the sampling rate. The resulting sparse point cloud is then fed into the downstream detection network. By adjusting the quantized sparsity level, the model can make a tradeoff between LiDAR efficiency and object detection accuracy.

Differentiable Voxelization

Conventional voxelization (Zhou and Tuzel 2018) is inherently non-differentiable, preventing the gradient of the final detection loss from propagating through it to the Mask Generator. To address this, we propose a heuristic differentiable voxelization method that approximates gradients through nearest-neighbor assignments. In the forward pass, we apply standard voxelization (Zhou and Tuzel 2018) to the sparse LiDAR points, $P_{\mathrm{s}}^t \in \mathbb{R}^{N_s \times D_p}$, producing a voxel tensor $V \in \mathbb{R}^{M_v \times K_v \times D_p}$, where M_v is the number of occupied voxels, K_v is the maximum points per voxel, and D_p is the per-point feature dimension. During back-propagation, we approximate the gradient $\frac{\partial Loss}{\partial P_s}$ from the voxel gradient $\frac{\partial Loss}{\partial V}$ by assigning each point to its nearest voxel center: $\frac{\partial Loss}{\partial P_{s,i}} \approx \alpha \frac{\partial Loss}{\partial V_{\mathrm{NN(i)}}}$, where $NN(i)$ is the index of the voxel whose center is closest to point $P_{s,i}$ and α is a scaling factor that stabilizes training. This simple yet effective heuristic restores gradient propagation through the voxelization step, making the entire pipeline end-to-end trainable.

Loss Function for Training

Our framework is trained end-to-end by minimizing a single composite loss averaged across $T-1$ consecutive frames:

$$\mathcal{L} = \frac{1}{T-1} \sum_{t=2}^{T} \Big(\mathcal{L}_{3D}^t + \lambda_1 \mathcal{L}_{\mathrm{distill}}^t + \lambda_2 \mathcal{L}_{\mathrm{mask}}^t + \lambda_3 \mathcal{L}_{\mathrm{CVaR}}^t \Big). \quad (5)$$

The 3D object detection loss $\mathcal{L}_{3D}^t$ combines classification and regression terms to compare the predicted bounding boxes ($\mathcal{B}^t$) with the ground-truth ($\mathcal{B}_{gt}^t$). To enforce temporal consistency, the distillation loss aligns the predicted queries with the reference queries Q_{ref}^t generated by the same fusion model on the full-dense point clouds: $\mathcal{L}_{\mathrm{distill}} = \left\| {Q'}^t - Q_{ref}^t \right\|_1$. To guide the Mask Generator, we define the mask loss, $\mathcal{L}_{\mathrm{mask}}^t$. We first project the ground-truth bounding box positions onto the LiDAR range image to produce a binary guidance mask, M_{GT}^t, then compute a per-pixel Focal Loss (FL) (Lin et al. 2017) to handle the class imbalance between foreground (object) and background pixels:

$$\mathcal{L}_{\mathrm{mask}}^t = \frac{1}{H_B\, W_B} \sum_{u=1}^{H_B} \sum_{v=1}^{W_B} \mathrm{FL}\big(M_{(u,v)}^t, M_{GT(u,v)}^t\big). \quad (6)$$

Finally, to improve the robustness of the Mask Generator for small yet critical objects (e.g. pedestrians), we introduce a Conditional Value at Risk (CVaR) loss that focuses on worst-case prediction errors. For each sample, we extract the per-pixel mask losses $\{\ell_k\}_{k=1}^{N_{small}}$ from $\mathcal{L}_{\mathrm{mask}}^t$ for regions corresponding to small objects and sort them in descending order. We define the 'Value at Risk', m^*, as the loss of the $\lceil (1-\beta)\, N_{small} \rceil$-th worst sample, such that: $m^* = \ell_{(\lceil (1-\beta)\, N_{small} \rceil)}$. Then, we formulate the CVaR loss:

$$\mathcal{L}_{\mathrm{CVaR}}^t = m^* + \frac{1}{\beta\, N_{small}} \sum_{k=1}^{N_{small}} \max\big(\ell_k - m^*, 0\big). \quad (7)$$

By penalizing the average of the worst β-fraction of losses, CVaR drives the network to improve performance on the most challenging and small-object regions.

Experiments

Experimental Setup

Dataset and Metrics: We evaluate our framework on the nuScenes (Caesar et al. 2020) and Lyft (Kesten et al. 2019) datasets. NuScenes contains 1000 driving scenes (700/150/150 train/val/test) captured with a 32-beam LiDAR at 20 Hz and six cameras providing 360° coverage, with 3D annotations for 10 classes at 2 Hz. We follow the official 2 Hz evaluation protocol for fair comparison. Lyft features a 64-beam LiDAR, six cameras, and 5 Hz annotation frequency across complex urban environments, offering finer temporal cues for evaluating our history-aware model.

For evaluation, we follow the official nuScenes metrics (mAP and NDS). For Lyft, we report AP and mAP, and we introduce a weighted-mAP (w-mAP) that weights each class AP by its frequency, reducing the influence of rare categories.

Implementation Details: Our adaptive LiDAR scanning module is a plug-in that integrates with any query-based camera–LiDAR fusion architecture. We use CMT (Yan et al. 2023) as the base model, which has six transformer decoder layers with 900 object queries. We use a ResNet (He et al. 2016) (low image resolution) or VoVNet (Lee and Park 2020) (high image resolution) model as image backbone, and VoxelNet (Zhou and Tuzel 2018) as the LiDAR backbone. For temporal context, we utilize a buffer of $T = 4$ frames for both datasets.

We employ a three-stage training strategy to ensure stable convergence. First, we train the Query Predictor in isolation by attaching the pre-trained (frozen) CMT and supervising it to generate accurate 3D bounding boxes from the Query Predictor output at future time t predicted based on the buffered queries. Then, we train the CMT to operate on sparse LiDAR inputs by initializing the Query Predictor with weights from the first stage and integrating it with the Mask Generator that sparsifies the LiDAR input. Although voxelization remains non-differentiable at this point, this step solidifies the ability of the fusion model to process sparse LiDAR inputs. Finally, the entire pipeline is fine-tuned end-to-end by replacing the standard voxelizer with the proposed differentiable voxelization layer.

Comparison to the State of the Art

We compare our adaptive scanning approach in two scenarios: 'next-frame prediction' and 'entire-sequence prediction'. In the next-frame prediction case, T past frames in the query queue are obtained by full scan to perform the adaptive scan for the next frame. On the other hand, the entire-sequence prediction always uses adaptive scan for all frames including the T recent past frames in the query buffer.

For the next-frame prediction setting on the nuScenes validation set, we achieve 66.0% LiDAR sparsity at low resolution while incurring marginal performance degradation of 0.59% in mAP and 0.14% in NDS, as shown in Table 1. Higher image resolution yields 66.8% sparsity while actually improving mAP by 1.0% and NDS by 0.27%. On the nuScenes test split, our model achieves sparsity level of 63.8% (low resolution) and 64.2% (high resolution) with

Model	Image Res.	Backbone	mAP% ↑	NDS% ↑	Scan Sparsity% ↑
AutoAlignV2 (Chen et al. 2022)	1600×900	CSPNet & VoxelNet	64.4	69.5	0
FUTR3D (Chen et al. 2023)	1600×900	VoVNet & VoxelNet	64.5	68.3	0
UVTR (Li et al. 2022)	1600×900	R50 & VoxelNet	65.4	70.2	0
MSMDFusion (Jiao et al. 2023)	448×800	R50 & VoxelNet	66.9	68.9	0
MVP (Yin, Zhou, and Krähenbühl 2021)	1600×900	DLA34 & VoxelNet	67.1	70.8	0
TransFusion (Bai et al. 2022)	448×800	R50 & VoxelNet	67.5	71.3	0
BEVFusion (Liang et al. 2022)	448×800	Swin-Tiny & VoxelNet	67.9	71.0	0
BEVFusion (Liu et al. 2023)	448×800	Swin-Tiny & VoxelNet	68.5	71.4	0
DeepInteraction (Yang et al. 2022)	1600×900	R50 & VoxelNet	69.9	72.6	0
CMT (Yan et al. 2023)	800×320	R50 & VoxelNet	67.9	70.7	0
+Adaptive Scan	800×320	R50 & VoxelNet	67.5	70.6	**66.0**
CMT	1600×640	VoVNet & VoxelNet	70.3	72.9	0
+Adaptive Scan	1600×640	VoVNet & VoxelNet	**71.0**	**73.1**	**66.8**

Table 1: Performance comparison for next-frame prediction on nuScenes validation.

Model	mAP% ↑	w-mAP% ↑	Sparsity% ↑	Car	Other Veh.	Bus	Truck	Motorcycle	Bicycle	Pedestrian
CMT	48.4	79.8	0	85.5	75.7	50.2	54.7	31.4	21.1	20.5
+Adaptive Scan	48.4	**80.4**	**68.7**	**87.1**	**77.5**	46.6	50.2	23.0	**29.3**	**25.2**

Table 2: Performance comparison for next-frame prediction on Lyft validation. 'Other Veh.' is other vehicle.

Model	mAP% ↑	NDS% ↑	Sparsity% ↑
PointPaint (Vora et al. 2020)	54.1	61.0	0
PointAug (Wang et al. 2021)	66.8	71.1	0
UVTR (Li et al. 2022)	67.1	71.1	0
FusionPaint (Xu et al. 2021)	68.1	71.6	0
TransFusion (Bai et al. 2022)	68.9	71.7	0
BEVFusion (Liu et al. 2023)	70.2	72.9	0
CMT* (800×320)	68.6	71.2	0
+Adaptive Scan	68.3	71.0	**63.8**
CMT* (1600×640)	70.4	73.0	0
+Adaptive Scan	70.0	72.9	**64.2**

Table 3: Performance comparison for next-frame prediction on nuScenes test. * denotes our reproduced results.

competitive performances (Table. 3). Note that the competitive performance is attained despite the dataset's low (2Hz) frame rate, which makes the next frame prediction challenging. On the higher frame rate (5Hz) Lyft dataset (Table 2), we attain 68.7% sparsity without sacrificing mAP and enhancing w-mAP by 0.75%. By selectively removing unnecessary background points, our approach surpasses the full-scan baseline in w-mAP.

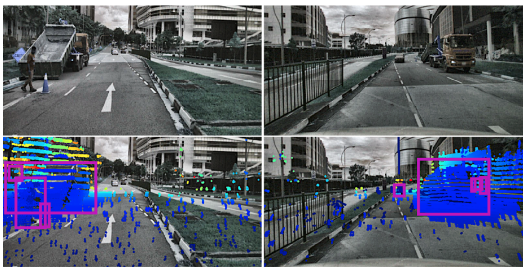

Figure 3: Top: original images. Bottom: corresponding selected selected LiDAR points, densely scanned in ROIs and sparsely elsewhere, with color encoding distance.

We also evaluate the 'entire-sequence prediction', where only the initial buffer frames ($T = 4$) use a full scan and the remaining frames always use the proposed adaptive LiDAR scanning. On the nuScenes validation dataset (excluding the initial buffer), our method maintains sparsity rates of 65.3% and 66.4% at both resolutions, maintaining comparable or even improved detection accuracy (Table 4). Similarly, on the Lyft dataset (Table 5), we achieve 67.2% sparsity and boost performance by $0.41\%/0.88\%$ in mAP/w-mAP. Crucially, by scanning only a sparse subset of beams, far fewer points are fed into the LiDAR backbone, which reduces the overall model GFLOPs. Specifically, our method reduces the GFLOPs by 3.54% (low resolution) and 2.63% (high resolution) on nuScenes, and by 1.84% on Lyft. This demonstrates that our framework not only avoids power consumption overhead in LiDAR sensing but also actively reduces the computational demands of the perception model despite that introducing additional modules such as the Query Predictor and Mask Generator. Figure 3 provides visualization examples of our adaptive scanning strategy on two frames from nuScenes, illustrating how the model densely scans regions containing objects and their immediate surroundings, while significantly reducing scanning density in non-ROI areas.

Ablation Study

Performance vs. Sparsity: We analyze the trade-off between scan sparsity and perception accuracy for 'next-frame prediction' on nuScenes and Lyft. As shown in Figures 4 and 5, denser scans (lower sparsity) improve mAP and NDS (nuScenes) or w-mAP (Lyft). On nuScenes, our adaptive scan surpasses the full-scan CMT baseline once sparsity is below 71.6% for mAP and 70.9% for NDS. On Lyft, we match or exceed full-scan mAP at 68.7% sparsity and outperform w-mAP at 71.2%.

Impact of Buffer Length: We evaluate how temporal buffer length affects accuracy and sparsity on nuScenes for 'next-frame prediction'. Table 6 shows that as we increase

Model	mAP% ↑	NDS% ↑	Sparsity% ↑	GFLOPs
CMT (800×320)	67.9	70.7	0	769.4
+Adaptive Scan	67.5	70.6	**65.3**	**742.2**
CMT (1600×640)	70.3	72.9	0	3260.9
+Adaptive Scan	**71.0**	**73.1**	**66.4**	**3175.2**

Table 4: Evaluation of entire sequence prediction on nuScenes validation.

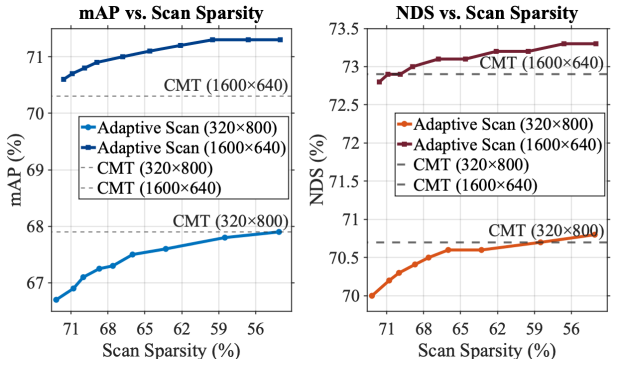

Figure 4: Ablation study comparing performance under different LiDAR levels on nuScenes validation set.

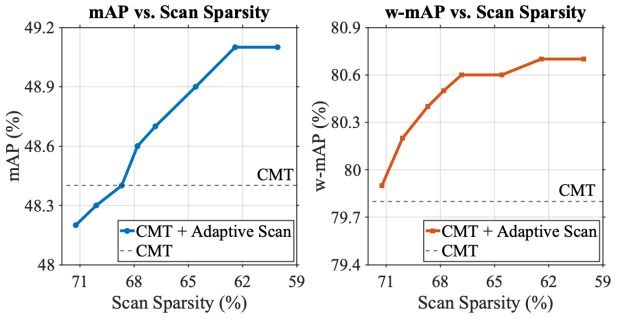

Figure 5: Sparsity ratios vs. Lyft validation set performance

the number of buffer frames, performance steadily improves, while scan sparsity naturally declines. Leveraging more historical context enables the model to anticipate objects that appear further away or re-enter the scene, but it also requires additional computation. However, even with a buffer of $T = 4$ frames, the overall cost remains lower than the full-scan CMT thanks to the sparse LiDAR input.

Buffer size (T)	mAP% ↑	NDS% ↑	Sparsity% ↑	GFLOPs
2	66.7	70.0	69.9	736.3
3	67.0	70.2	67.4	738.7
4	**67.5**	**70.6**	66.0	742.2

Table 6: Buffer size (T) vs. nuScenes performance

Impact of Loss Terms: We investigate the effects of the proposed losses, $\mathcal{L}_{\text{CVaR}}$ and $\mathcal{L}_{\text{distill}}$, on both detection accuracy and scan sparsity. Table 7 demonstrates that without these losses, both performance and sparsity suffer significantly. Introducing the distillation loss, $\mathcal{L}_{\text{distill}}$, alone yields substantial gains as it guides the predicted queries to mimic

Model	mAP% ↑	w-mAP% ↑	Sparsity% ↑	GFLOPs
CMT	48.4	79.8	0	3351.2
+Adaptive Scan	**48.6**	**80.5**	**67.2**	**3289.6**

Table 5: Evaluation of entire sequence prediction on Lyft validation.

full-scan queries. The CVaR loss, $\mathcal{L}_{\text{CVaR}}$, further enhances robustness by specifically forcing the model to improve its predictions on small and rare objects. Combining both losses yields the best detection performance with a slight decrease in sparsity. This is an expected trade-off, as $\mathcal{L}_{\text{CVaR}}$ directs the model to scan more diverse areas.

$\mathcal{L}_{\text{CVaR}}$	$\mathcal{L}_{\text{distill}}$	mAP% ↑	NDS% ↑	Sparsity% ↑
		66.5	69.9	65.4
✓		67.0	70.3	65.8
	✓	67.3	70.4	**66.3**
✓	✓	**67.5**	**70.6**	66.0

Table 7: Impact of loss terms on nuScenes validation set

Impact of Differentiable Voxelization: Table 8 evaluates the effect of differentiable voxelization that allows the gradient propagating through the end-to-end datapath. It confirms that, without differentiable voxelization, the detection performance drops significantly. Incorporating the proposed approach notably improves accuracy, though sparsity slightly decreases. This minor reduction in sparsity is expected, as the model now identifies additional ROI areas that are critical to the task performance.

Diff. voxel	mAP% ↑	NDS% ↑	Sparsity% ↑
	66.7	70.0	68.1
✓	**67.5**	**70.6**	66.0

Table 8: Impact of differentiable voxelization on nuScenes

Conclusion

We introduce an adaptive LiDAR scanning framework that enhances sensor fusion systems' efficiency by leveraging temporal history to predict and selectively scan salient regions of a scene. Our two-stage approach uses a history-aware Query Predictor to predict object locations and a differentiable Mask Generator to create an efficient scanning policy. Experiments on nuScenes and Lyft datasets demonstrate that this method can reduce LiDAR scanning by over 65%, achieving competitive performance, all while reducing computational overhead. This work represents a significant step towards more intelligent, efficient, and sustainable perception systems for autonomous vehicles.

Acknowledgments

This work was supported in part by COGNISENSE, one of seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

References

Ambarella, Inc. 2019. Ambarella Introduces CV22AQ Automotive Camera SoC for Advanced Driver Assistance Systems (ADAS). Press Release.

Bai, X.; Hu, Z.; Zhu, X.; Huang, Q.; Chen, Y.; Fu, H.; and Tai, C.-L. 2022. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1090–1099.

Bolukbasi, T.; Wang, J.; Dekel, O.; and Saligrama, V. 2017. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, 527–536. PMLR.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1907–1915.

Chen, X.; Zhang, T.; Wang, Y.; Wang, Y.; and Zhao, H. 2023. Futr3d: A unified sensor fusion framework for 3d detection. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 172–181.

Chen, Z.; Li, Z.; Zhang, S.; Fang, L.; Jiang, Q.; and Zhao, F. 2022. Deformable feature aggregation for dynamic multi-modal 3D object detection. In *European conference on computer vision*, 628–644. Springer.

Cheng, T.-Y.; Hu, Q.; Xie, Q.; Trigoni, N.; and Markham, A. 2022. Meta-sampler: Almost-universal yet task-oriented sampling for point clouds. In *European Conference on Computer Vision*, 694–710. Springer.

Feng, Y.; Liu, S.; and Zhu, Y. 2020. Real-time spatio-temporal lidar point cloud compression. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 10766–10773. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, J.; Liu, Z.; Zou, Z.; Ye, X.; Bai, X.; et al. 2023. Query-based temporal fusion with explicit motion for 3d object detection. *Advances in Neural Information Processing Systems*, 36: 75782–75797.

Houston, J.; Zuidhof, G.; Bergamini, L.; Ye, Y.; Chen, L.; Jain, A.; Omari, S.; Iglovikov, V.; and Ondruska, P. 2021. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, 409–418. PMLR.

Hua, W.; Zhou, Y.; De Sa, C. M.; Zhang, Z.; and Suh, G. E. 2019. Channel gating neural networks. *Advances in neural information processing systems*, 32.

Huang, G.; Chen, D.; Li, T.; Wu, F.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jiao, Y.; Jie, Z.; Chen, S.; Chen, J.; Ma, L.; and Jiang, Y.-G. 2023. Msmdfusion: Fusing lidar and camera at multiple scales with multi-depth seeds for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 21643–21652.

Kesten, R.; Usman, M.; Houston, J.; Pandya, T.; Nadhamuni, K.; Ferreira, A.; Yuan, M.; Low, B.; Jain, A.; Ondruska, P.; Omari, S.; Shah, S.; Kulkarni, A.; Kazakova, A.; Tao, C.; Platinsky, L.; Jiang, W.; and Shet, V. 2019. Lyft Level 5 AV Dataset 2019. https://woven.toyota/en/dataset.

Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; and Waslander, S. L. 2018. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 1–8. IEEE.

Lang, I.; Manor, A.; and Avidan, S. 2020. Samplenet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7578–7588.

Lee, S.; Lee, D.; Choi, P.; and Park, D. 2020. Accuracy–power controllable LiDAR sensor system with 3D object recognition for autonomous vehicle. *Sensors*, 20(19): 5706.

Lee, Y.; and Park, J. 2020. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13906–13915.

Li, Y.; Chen, Y.; Qi, X.; Li, Z.; Sun, J.; and Jia, J. 2022. Unifying voxel-based representation with transformer for 3d object detection. *Advances in Neural Information Processing Systems*, 35: 18442–18455.

Liang, T.; Xie, H.; Yu, K.; Xia, Z.; Lin, Z.; Wang, Y.; Tang, T.; Wang, B.; and Tang, Z. 2022. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35: 10421–10434.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, B.; Chen, Y.; Wang, B.; Yang, M.; and Kim, H.-S. 2025. H-PCC: Point Cloud Compression with Hybrid Mode Selection and Content Adaptive Down-sampling. *IEEE Robotics and Automation Letters*.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D. L.; and Han, S. 2023. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, 2774–2781. IEEE.

Meng, Y.; Panda, R.; Lin, C.-C.; Sattigeri, P.; Karlinsky, L.; Saenko, K.; Oliva, A.; and Feris, R. 2021. Adafuse: Adaptive temporal fusion network for efficient action recognition. *arXiv preprint arXiv:2102.05775*.

Milioto, A.; Vizzo, I.; Behley, J.; and Stachniss, C. 2019. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 4213–4220. IEEE.

Nezhadarya, E.; Taghavi, E.; Razani, R.; Liu, B.; and Luo, J. 2020. Adaptive hierarchical down-sampling for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12956–12964.

Panda, R.; Chen, C.-F. R.; Fan, Q.; Sun, X.; Saenko, K.; Oliva, A.; and Feris, R. 2021. Adamml: Adaptive multimodal learning for efficient video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7576–7585.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Raj, T.; Hanim Hashim, F.; Baseri Huddin, A.; Ibrahim, M. F.; and Hussain, A. 2020. A survey on LiDAR scanning mechanisms. *Electronics*, 9(5): 741.

Rockafellar, R. T.; Uryasev, S.; et al. 2000. Optimization of conditional value-at-risk. *Journal of risk*, 2: 21–42.

Sahin, F. E. 2019. Long-range, high-resolution camera optical design for assisted and autonomous driving. In *Photonics*, volume 6, 73. MDPI.

Shoouri, S.; Yang, M.; Fan, Z.; and Kim, H.-S. 2023. Efficient computation sharing for multi-task visual scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17130–17141.

Tayebati, S.; Tulabandhula, T.; and Trivedi, A. R. 2025. Generative Sensing: Pre-training LiDAR with Masked Autoencoders for Ultra-Frugal Perception. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Veit, A.; and Belongie, S. 2018. Convolutional networks with adaptive inference graphs. In *Proceedings of the European conference on computer vision (ECCV)*, 3–18.

Velodyne Lidar, Inc. 2018a. *HDL-32E Datasheet*. Revision M, Document 97-0038.

Velodyne Lidar, Inc. 2018b. *HDL-64E S3 Specification Sheet*. Revision J, Document 63-9194.

Vora, S.; Lang, A. H.; Helou, B.; and Beijbom, O. 2020. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4604–4612.

Wang, C.; Ma, C.; Zhu, M.; and Yang, X. 2021. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11794–11803.

Wang, X.; Yu, F.; Dou, Z.-Y.; Darrell, T.; and Gonzalez, J. E. 2018. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European conference on computer vision (ECCV)*, 409–424.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, 180–191. PMLR.

Wen, C.; Yu, B.; and Tao, D. 2023. Learnable skeleton-aware 3D point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17671–17681.

Wu, Z.; Xiong, C.; Ma, C.-Y.; Socher, R.; and Davis, L. S. 2019. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1278–1287.

Xu, S.; Zhou, D.; Fang, J.; Yin, J.; Bin, Z.; and Zhang, L. 2021. Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 3047–3054. IEEE.

Yan, J.; Liu, Y.; Sun, J.; Jia, F.; Li, S.; Wang, T.; and Zhang, X. 2023. Cross modal transformer: Towards fast and robust 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 18268–18278.

Yang, Z.; Chen, J.; Miao, Z.; Li, W.; Zhu, X.; and Zhang, L. 2022. Deepinteraction: 3d object detection via modality interaction. *Advances in Neural Information Processing Systems*, 35: 1992–2005.

Yin, T.; Zhou, X.; and Krähenbühl, P. 2021. Multimodal virtual point 3d detection. *Advances in Neural Information Processing Systems*, 34: 16494–16507.

Yuan, Z.; Wu, B.; Sun, G.; Liang, Z.; Zhao, S.; and Bi, W. 2020. S2dnas: Transforming static cnn model for dynamic inference via neural architecture search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 175–192. Springer.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4490–4499.