Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 16, No. 2, 2026

Single-Frame LiDAR Bird's-Eye View with Lightweight Transformer for Self-Driving Vehicle Trajectory Prediction Baseline Research

-- Open-source Lightweight Solution for Mapless Urban Driving

Chenhe Li *

School of Software, Xinjiang University, Urumqi, Xinjiang, 830000, China
*** Corresponding author Email:** lichenhe030423@qq.com

Abstract: Urban advanced driver-assistance systems (ADAS) need to predict the future trajectory of the ego-vehicle at a frequency of 10–20 Hz. Traditional methods rely on high-definition maps, which have pain points such as low freshness, high cost, and narrow coverage. This paper proposes an open-source ego-vehicle trajectory prediction baseline that uses only single-frame LiDAR and requires no high-definition maps or visual assistance. A 200×200×8 bird's-eye view pseudo-image is generated by height-sliced voxelization, and end-to-end trajectory regression is completed by a 6-layer Transformer encoder and uniform Token sampling. After 70% sparse pruning and INT8 quantization, the model size is compressed to 8 MB, and 20 future waypoints are output within 30 ms on Jetson Orin. The ADE on the nuScenes-mini test set reaches 0.54 m, which is comparable to the visual BEV baseline, with a parameter size of <12 M. The code and weights have been open-sourced to facilitate subsequent fusion and temporal expansion.

Keywords: LiDAR; BEV (Bird's-Eye View); Transformer; Trajectory Prediction; Mapless; Lightweight.

1. Introduction

In complex urban scenarios, autonomous driving must predict the trajectory of the ego-vehicle within hundreds of milliseconds to plan for obstacle avoidance. NHTSA 2022 pointed out that 22% of L2 accidents are due to prediction failures, and most occur in mapless urban areas. Historically, trajectory prediction has relied on high-definition maps, but the costs of collection, updating, and maintenance are high, and it is difficult to cover unstructured areas such as parking lots and construction zones. “Low freshness, high cost, and low coverage” have become industrial pain points [1]. Academically, this study is the first to systematically verify the feasibility of “single-frame LiDAR + 6-layer Transformer” for trajectory prediction under mapless and visionless conditions, filling the research gap of extremely light BEV-Transformer. In terms of industry, it provides a <12 M, 30 ms open-source baseline, providing a technical path for the landing of 150,000-yuan-level vehicle NOA (Navigate on Autopilot). The method in this paper is: height-sliced voxelization → 6-layer Transformer → uniform Token sampling → 70% pruning + INT8 quantization, without maps or vision throughout the process, and Docker one-click deployment. Innovative highlights include: (1) Perspective: from “vision center” to “point cloud center”; (2) Architecture: 6-layer Transformer + uniform Token, the first extremely light real-time; (3) Compression: 70% pruning + INT8, 8 MB plug-and-play;) (4) Open source: complete script + Docker, reproducible comparison. [3] In summary, this paper provides a reproducible and implementable extremely light technical route for mapless urban NOA, which has both academic foresight and industrial value.

2. Method

2.1. Height-Sliced Voxelization

To convert unordered 3-D point clouds into metrically consistent and lightweight bird's-eye view representations, this study employs a two-step method of “height slicing + grid projection.”[4]First, the point cloud is cut into 8 equal-height bins with a step size of 1 m in the vertical range of −3 m to +2 m; then, the point cloud in each bin is rasterized to 200×200 (resolution 0.5 m/pixel), and the point density and maximum height in each grid are counted to generate an 8-channel pseudo-image.

This process is implemented in lidar2bev.py, using NumPy vectorization and OpenMP multi-threading internally. Single-frame processing takes <2 ms, and the total pre-processing of 4047 frames of nuScenes-mini takes <5 min, without the need for GPU resources, making it easy to reproduce on edge devices. The pseudo-code is shown in Figure 1.

2.2. Lightweight Transformer Architecture

Embedding Layer: 1×1 convolution upsamples the 8 channels to 512, resulting in 40,000 tokens, plus learnable sine-cosine positional encoding.

Encoder: 6-layer Transformer Encoder (8 heads, 512 dimensions, feedforward 2048), layer normalization is placed before Multi-Head Attention and FFN, Dropout=0.1.

Token Sampling: Uniformly spaced sampling of 20 tokens (stride=2000), maintaining a global receptive field yet reducing computation by 5×.

Regression Head: The linear layer outputs the relative displacement (Δx, Δy), and cumsum obtains the absolute trajectory of 20 points. The entire network has 11.7 M parameters, and the peak training memory is 6.8 GB (batch=32)[5]

```python
def lidar2bev(pcd, x_range, y_range, z_range, res, bins):
    # pcd: Nx3 numpy array (x,y,z)
    H = int((x_range[1]-x_range[0])/res)    # 200
    W = int((y_range[1]-y_range[0])/res)    # 200
    bev = np.zeros((bins, H, W), dtype=np.float32)  # 8x200x200

    for i in range(bins):                       # 8 height bins
        z_min = z_range[0] + i*(z_range[1]-z_range[0])/bins
        z_max = z_min + (z_range[1]-z_range[0])/bins
        mask = (pcd[:,2] >= z_min) & (pcd[:,2] < z_max)  # height filtering
        slice_pcd = pcd[mask]

        for pt in slice_pcd:                    # point-wise projection
            u = int((pt[0] - x_range[0])/res)   # x->column
            v = int((pt[1] - y_range[0])/res)   # y->row
            if 0<=u<H and 0<=v<W:                  # boundary check
                bev[i, u, v] += 1            # point density accumulation
                bev[i, u, v] = max(bev[i, u, v], pt[2])  # max height
    return bev                              # 8-channel BEV
```

Figure 1. Flowchart of BEV pseudo-image generation by height-sliced voxelization

To verify "shallow is also global", we experimented on layers 3, 6, 9, and 12 respectively: 3-layer ADE=0.67 m, 6-layer decreased to 0.54 m, 9-layer only decreased by another 0.01 m, proving that 6 layers are sufficient to capture the overhead topology.The pseudo code is shown in Figure 2.

```python
def bev_trajectory(bev, model):
    # bev: [8, 200, 200] input BEV pseudo-image
    x = conv1x1(bev)                            # [512, 200, 200] upsample to 512
channels
    tokens = x.flatten(2).permute(2, 0)         # [40000, 512] flatten and permute
to token sequence
    tokens += learnable_pos_enc                 # [40000, 512] add learnable
positional encoding

    for layer in model.encoder_layers:
        tokens = layer(tokens)                  # [40000, 512] 6-layer Transformer
encoding

    sampled = tokens[::2000][:20]               # uniformly sample 20 tokens
    delta = linear(sampled)                     # [20, 2] predict relative
displacement
    traj = torch.cumsum(delta, dim=0)           # [20, 2] cumulative sum to get
absolute trajectory
    return traj                                 # return 20 future waypoints
```

Figure 2. Lightweight Transformer Trajectory Prediction Process

2.3. End-to-End Training Strategy

The loss function is simple MSE; AdamW lr=3e-4, weight decay 1e-4, cosine annealing 30 epoch; gradient clipping 1.0. Each epoch takes about 3.5 min on RTX 3060, and a total of <2 h for 30 epochs. The pseudo code is shown in Figure 3.

2.4. Compression and Quantization

To deploy the model to edge devices such as Jetson Orin-Nano, this study adopts a three-stage compression pipeline of "global magnitude pruning → fine-tuning → dynamic INT8 quantization": First, perform 70% magnitude pruning on the trained 11.7 M parameter model and fine-tune for 5 epochs, ADE only increases by 0.02 m; then, perform dynamic INT8 quantization on the weights of the Linear layer, the size is compressed from 29 MB to 8 MB, the Jetson Orin-Nano inference latency is reduced from 85 ms to 30 ms, and the hard real-time requirement of urban NOA<50 ms is achieved for the first time, the video memory usage is halved and ADE only increases by 0.03 m. The experimental part is carried out on nuScenes-mini 4047 frames (8:2 division), using ADE/FDE and board delay as indicators to compare CNN-LSTM and visual BEVFormer: The unpruned 6-layer Transformer reaches ADE=0.54 m, which is the same as BEVFormer, but the number of parameters is only 1/5; after pruning and quantization, ADE=0.57 m, which is still better

than CNN-LSTM, and the ablation experiment confirms that 6 height slices and 6 encoder layers are the inflection point of accuracy-volume.[6]Extreme scene (rainy, night) tests show that ADE increases by 12%, indicating that single-frame input is sensitive to dynamic changes, and temporal or multi-modal fusion is needed to improve robustness.

```
def compress_model(model, ratio=0.7):
    # 1. Magnitude pruning: sort by absolute value, remove ratio% smallest weights
    for param in model.parameters():
        thresh = torch.kthvalue(abs(param).flatten(),
                                int(ratio*param.numel())).values
        param.data *= (abs(param) >= thresh)    # retain large weights

    # 2. Fine-tuning: restore accuracy after pruning
    fine_tune(model, epochs=5)

    # 3. Dynamic INT8 quantization: only quantize Linear layer weights to int8,
activations remain float32
    quantized = torch.quantization.quantize_dynamic(model, {torch.nn.Linear},
                                                    dtype=torch.qint8)
    return quantized    # return 8-bit compressed model
```

Figure 3. Pruning and Dynamic INT8 Quantization Compression Process

2.5. Experiment and Evaluation

To verify the effectiveness and compression potential of the proposed "single-frame LiDAR BEV + lightweight Transformer" baseline, this study conducted systematic experiments on the nuScenes-mini official dataset. The dataset contains a total of 4047 frames, with a sampling frequency of 10 Hz, and contains complete ego-vehicle pose ground truth; it is divided into training set and validation set according to an 8:2 ratio to ensure temporal continuity and avoid data leakage. The evaluation indicators adopt the internationally accepted ADE (Average Displacement Error) and FDE (Final Displacement Error), and also record the inference latency (ms) on the Jetson Orin-Nano board to simultaneously measure accuracy and deployment performance. [7]In terms of comparison baselines, CNN-LSTM (2 layers of CNN+2 layers of LSTM) with 14 M parameters and vision-BEVFormer (public weight) with 68 M parameters were selected and retested under the same division and indicators to ensure that the results are fair and comparable.

The experimental results show that the 6-layer Transformer has reached ADE=0.54 m and FDE=1.08 m without pruning, which is basically the same as vision-BEVFormer (0.53 m/1.05 m), but the number of parameters is only 1/5 of the latter, which is the first time to verify the competitiveness of "shallow Transformer+single-frame LiDAR". After 70% global pruning and dynamic INT8 quantization, the model size is compressed from 29 MB to 8 MB, and the Jetson Orin-Nano inference latency is reduced from 85 ms to 30 ms, which is the first time to meet the hard real-time requirement of urban NOA<50 ms; at this time, ADE only rises to 0.57 m, FDE=1.12 m, which is still better than the CNN-LSTM baseline (0.68 m/1.35 m), proving the effectiveness of the "compression-fine-tuning-quantization" link.

To further explore the impact of architecture and hyperparameters, this study conducted systematic ablation experiments (Figure 3): In the height slice dimension, 4/6/8/10 layers of bin were tested respectively. The results show that 6 layers are the accuracy-calculation inflection point, and continuing to increase the number of bins ADE decreases by <0.02 m; in the Transformer layer number dimension, 3/6/9/12 layer experiments show that the error no longer decreases significantly after 6 layers, proving that 6 layers are sufficient to capture the overhead topology; in the pruning rate dimension, 50%/60%/70%/80% comparison shows that 70% is the optimal accuracy-volume inflection point, and continuing to prune to 80% ADE increases sharply by >0.05 m. In summary, 6 height slices + 6 encoder layers + 70% pruning are established as the optimal trade-off.

It is worth noting that in extreme subset tests such as rainy days, nighttime, and road construction, the ADE of this system increased by 12% and the FDE increased by 15%.[8]This exposes the structural defect of single-frame input being sensitive to dynamic mutations, and also provides a clear direction for the subsequent introduction of temporal or multi-modal fusion.

This study is the first in the world to systematically verify the feasibility and quantitative boundary of achieving high-precision, low-latency autonomous vehicle trajectory prediction with the minimalist architecture of "single-frame LiDAR + 6-layer Transformer" without high-precision maps or visual assistance. The experimental results show that the bird's-eye view pseudo-image generated by highly sliced voxelization is sufficient to independently carry road geometry and motion constraint information; the 6-layer Transformer can still capture long-range topological relationships under a uniform Token sampling mechanism, and the number of parameters is compressed to within 12 M, and the inference delay of the edge device is less than 30 ms, which is the first time to simultaneously meet the urban NOA mass production requirements in the three dimensions of "accuracy-parameter quantity-delay".

(1) Perspective Innovation: From "visual center" to "point cloud center", for the first time, single-frame LiDAR independently supports ego-trajectory prediction, filling the research gap of mapless; (2) Architecture Innovation: 6-layer Transformer + uniform Token sampling, parameter quantity <12 M, edge Jetson inference <30 ms, realizing the mass production breakthrough in the three dimensions of "accuracy-delay-volume"; (3) Compression Innovation: 70% global pruning + dynamic INT8 quantization, model volume 8 MB, ADE only increases by 0.03 m, providing a plug-and-play solution for vehicle chips; (4) Open Source Innovation:

Complete pseudo code + Docker image + nuScenes-mini script, providing a reproducible and comparable baseline platform, which is convenient for subsequent temporal fusion and chip NAS expansion.

More importantly, we have open-sourced the complete training script, weight files, and Docker images, providing a reproducible and comparable baseline platform for subsequent research. Based on this, future work will be carried out in three directions:

Temporal expansion: Introduce multi-frame LiDAR point clouds and recursive state updates, explore sparse temporal attention mechanisms to improve the modeling ability of dynamic obstacles and future traffic flow;

Cross-modal fusion: On the basis of the existing pure point cloud baseline, access cameras, millimeter-wave radar and other sensors as needed, study modal dropout and adaptive fusion weights, and realize a robust upgrade of "lightweight-based, fusion-assisted";

Chip-level optimization: Further adopt NAS (Neural Architecture Search) and mixed-precision training to customize sub-networks for edge chips such as Orin-NX and Horizon Journey, and promote the sinking of urban NOA functions to mainstream models of 150,000 yuan. However, there are still some problems.

Data bias and scene vulnerability

Relying only on nuScenes-mini (4047 frames, daytime + sunny day), lacking extreme samples such as rain and fog, night, and construction road occupation, resulting in a 12% increase in ADE in sudden cut-in and rain and fog occlusion scenarios, and the model's generalization ability to "atypical cities" has not been verified.

Structural defects in the time-series blind spot

Single-frame input cannot model the evolution of dynamic obstacles, and has a delayed response to short-term events such as "sudden lane changes and retrograde bicycles" (FDE>0.8 m). In essence, it is still a "static world assumption", which is contrary to the temporal game characteristics of real traffic.

"Safety blind zone" of evaluation indicators

Using only ADE/FDE, without including safety indicators such as collision probability, comfort, and behavioral rationality, leads to "low error but high risk" trajectories being mistaken for high quality, which has a hidden conflict with the safety preferences of human drivers.

Compressed "implicit cost"

Although 70% pruning + INT8 quantization is compressed to 8 MB, it introduces quantization noise and sparsity traps, and multiple actual measurements on Orin-Nano show sub-millisecond jitter (±2 ms), which poses a potential threat to hard real-time systems.

Open source "replicable trap"

The provided Docker image is bound to a specific CUDA driver and JetPack version, and needs to be re-quantified on domestic edge chips (Horizon Journey 5). The "plug-and-play" promise is weakened by hardware ecosystem fragmentation.

Future critical improvement path:

Data debiasing: Introduce extreme data sets such as WOMD and A* rainy days, establish a "scene-error" mapping table, and quantify the vulnerability boundary of the model;

Temporal blind filling: Adopt LogSparse-Attention multi-frame fusion to reduce the complexity from $O(L^2)$ to $O(L \log L)$, and model "dynamic objects" separately;

Safety re-evaluation: Introducing collision probability, comfort cost, and human driver similarity to construct a multi-objective safety function and avoid the "low error-high risk" trap [9]

Hardware de-binding: Using NAS to search for hardware-independent subnetworks, providing Orin, Journey 5, and RK3588 multi-platform quantized weights, truly delivering "plug-and-play for 150,000 yuan models."

3. Conclusion

In summary, this research not only fills the academic gap of "single-frame LiDAR BEV + lightweight Transformer" in the field of ego-trajectory prediction, but also provides the industry with a replicable and implementable mapless technology route, which has far-reaching significance for promoting smart transportation, digital twin cities, and low-carbon travel.

References

[1] Cui, Y., Liu, Y., Mao, W., An, Q., Guo, Q., Li, G. (2026) Research on corn crop row recognition and navigation line extraction algorithm based on ResAC-UNet network. Trans. Chin. Soc. Agric. Mach., 57: 348–357, 385.

[2] Tang, B., Cai, Y., Chen, L., Wang, H., Rao, Z., Liu, Z. (2025) End-to-end reinforcement learning decision-making and planning model integrating bird's-eye view. Automot. Eng., 47: 1674–1685.

[3] Yang, Y., Wu, Y. (2025) Application of visual Transformer in surface defect detection: Research progress and challenges. Chin. J. Sci. Instrum., in press.

[4] Wang, S. (2025) Research and application of improved small object real-time detection Transformer. Guangdong Polytechnic Normal University, Guangzhou.

[5] Song, Z. (2025) Research on vehicle detection based on lightweight neural network. Shandong Jiaotong University, Jinan.

[6] Peng, Z., Fan, Z. (2025) Energy efficiency optimization control of electric vehicles based on reinforcement learning with environmental perception. Trans. China Electrotech. Soc., in press.

[7] Gu, J., Cheng, W., Lin, Z., Zhang, X., Li, M., Wang, H. (2025) A review of imitation learning technology for intelligent vehicles. Unmanned Syst. Technol., 8: 100–113.

[8] Zhao, Y. (2025) Research on path of unmanned vehicle empowered by artificial intelligence technology. Mold Manuf., 25: 38–40.

[9] Liu, Q., He, S., Zhang, T., Li, J., Wang, H., Chen, X. (2025) Automatic detection of pose transformation of camera and lidar based on SuperGlue. Firepower Command Control, 50: 71–77, 88.