

SDPAD: A Fully Spike-Driven Pipeline for End-to-End Autonomous Driving

Chengjun Zhang^{1,2} Yuhao Zhang^{1,2} Jie Yang^{1,2,3} Mohamad Sawan^{1,2,3}

¹Zhejiang Key Laboratory of 3D Micro/Nano Fabrication and Characterization,
Westlake Institute for Optoelectronics, Fuyang, Hangzhou, China

²Integrated-On-Chips Brain-Computer Interfaces Zhejiang Engineering Research Center,
Hangzhou, Zhejiang, China

³CenBRAIN Neurotech, School of Engineering, Westlake University,
Hangzhou, Zhejiang, China

Corresponding authors: Jie Yang and Mohamad Sawan
yangjie@westlake.edu.cn; sawan@westlake.edu.cn

October 9, 2026

Abstract

End-to-end autonomous driving demands trajectory planners that are both highly accurate and cheap enough for edge deployment. State-of-the-art artificial neural network (ANN) planners meet the accuracy requirement at the cost of heavy dense computation, while spiking neural networks (SNNs)—though promising orders-of-magnitude energy savings through sparse, event-driven arithmetic—still lag far behind in planning accuracy. We present **SDPAD**, a fully spike-driven end-to-end planning pipeline that closes this gap. SDPAD converts a pre-trained ANN perception stack into integer-spike form via quantized ANN2SNN conversion, lifts multi-view images into the bird’s-eye-view (BEV) space with a spike-driven-max (SDM) depth distribution (Spike-3D-Lift), and plans through the Spike-QFormer, a spiking query transformer in which ego, agent, and map queries distilled from the BEV scene are fused by learnable waypoint queries via cross-attention, followed by deformable spike-cross-attention refinement. Every operation is gated by integer spikes and inference is a single feed-forward pass without temporal simulation loops. On the nuScenes open-loop benchmark, SDPAD achieves an average L_2 error of 0.40 m and a collision rate of 0.12%, on par with strong ANN planners while consuming 69.9 mJ—less than 2% of recent ANN baselines. In closed-loop evaluation on the NAVSIM navtest split, SDPAD reaches 86.3 PDMS, surpassing the previous SNN planner SAD by 4.3 points and matching mainstream ANN planners at a fraction of their energy. To our knowledge, SDPAD is the first fully spike-driven planner evaluated in end-to-end autonomous driving, demonstrating that SNNs can rival dense ANNs in complex driving tasks.

1 Introduction

The evolution of end-to-end autonomous driving has seen a paradigm shift from modular pipelines to unified neural architectures capable of directly mapping raw sensor data to control signals Chen et al. [2024a]. While these dense artificial neural network (ANN) models, as shown in Fig. 1a—ranging from vectorized scene planners such as ST-P3, UniAD and VAD Hu et al. [2022, 2023], Jiang et al. [2023] to recent world-model and diffusion-based planners Li et al. [2025], Yang et al. [2025b], Zheng et al. [2025], Liao et al. [2025], Zhang et al. [2026b]—exhibit impressive closed-loop and open-loop performance, they are constrained by a fundamental performance–efficiency dilemma. The stringent latency and power envelopes of embedded vehicular hardware make the deployment of heavy transformer-based or dense convolutional backbones highly challenging, even when adopting lightweight variants.

Spiking neural networks (SNNs) present a theoretically ideal solution for embedded agents Martínez et al. [2024], Yang et al. [2025a]. By mimicking biological neurons, SNNs execute event-driven, sparse computing through discrete spikes, achieving high energy efficiency on neuromorphic hardware by leveraging spatial and temporal sparsity Zhu et al. [2024] (Fig. 1b). However, despite their success in low-dimensional control and basic perception, SNNs still lack a planning-grade architecture for autonomous driving. We identify three concrete bottlenecks. First, mainstream SNN-based driving models rely on iterative time-steps simulations, where sequential state updates introduce substantial latency overhead and hinder real-time deployment. Second, energy-efficient 2D-to-3D feature lifting, a prerequisite for BEV perception, depends on dense, floating-point softmax depth distribu-

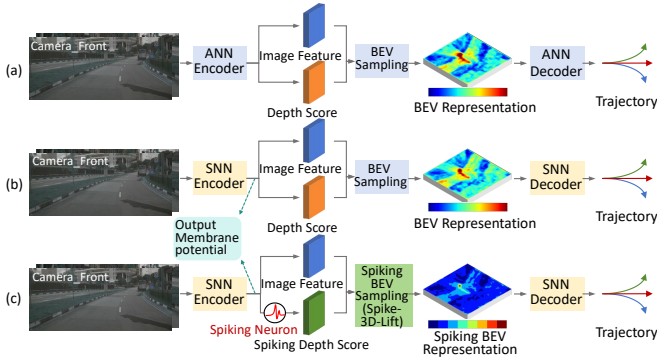


Figure 1: The comparison of different end-to-end paradigms. (a) Dense ANN pipelines compute in floating point throughout. (b) Existing SNN planners still rely on continuous membrane potentials or softmax depth scores for BEV lifting. (c) SDPAD is fully spike-driven end to end: the proposed Spike-3D-Lift turn BEV sampling into spike BEV representation.

tions that are ill-suited to spike-driven hardware. Third, existing spiking planners cannot model the multimodal interactions among the ego vehicle, surrounding agents, and map context, because their decoders lack a query-based mechanism to selectively aggregate heterogeneous scene information.

To address these bottlenecks, we introduce SDPAD, as shown in Fig. 1c, a fully spike-driven pipeline for end-to-end autonomous driving. SDPAD leverages fully integer-valued, rate-coded spike computations within a single feed-forward pass, completely dispensing with temporal unrolling. Concretely, a pre-trained ANN perception stack is converted into integer-spike form through quantized ANN2SNN conversion, and multi-view images are lifted into the BEV space by the proposed Spike-3D-Lift, which replaces the softmax depth distribution with a spike-driven-max so that frustum construction degenerates into spike-gated accumulation. On top of the Spiking BEV representation, the proposed Spike-QFormer distills three complementary query groups, an ego query, agent queries, and map queries, through spiking decoder layers, and lets learnable waypoint queries fuse them via a single-layer spiking cross-attention, producing command-conditioned trajectory candidates that are further refined by a deformable spike-cross-attention module anchored to the predicted trajectory itself. Auxiliary BEV segmentation and agent-state supervision regularize the scene representation during training. The core contributions of this work are three-fold:

1. We propose SDPAD, to our knowledge the first fully spike-driven end-to-end planner evaluated in closed loop on NAVSIM. Built on integer-spike computation with single-pass inference, it avoids the latency of temporal simulation loops entirely.
2. We propose Spike-3D-Lift, which replaces the floating-point softmax depth distribution with a spike-driven-max, turning 2D-to-3D BEV lifting into hardware-

friendly spike-gated accumulation while preserving geometric fidelity.

3. We propose the Spike-QFormer, a unified spiking query transformer that distills ego, agent, and map context into dedicated query groups and fuses them into waypoint predictions via spiking cross-attention, together with a deformable spike-cross-attention module for trajectory refinement.

Extensive experiments on nuScenes and NAVSIM show that SDPAD substantially narrows the gap between SNN and ANN planners, achieving planning accuracy comparable to mainstream ANN baselines at a small fraction of their energy consumption.

2 Related Work

2.1 End-to-End Autonomous Driving

Recent end-to-end models integrate perception, prediction, and planning into a differentiable framework. Methods like UniAD and VAD Hu et al. [2023], Jiang et al. [2023] utilize vectorized representations and bird’s-eye-view (BEV) spaces to explicitly model scene geometry and agent interactions. Subsequent works further improve planning quality through probabilistic anchor vocabularies Chen et al. [2024b], sparse scene representations Sun et al. [2025], truncated diffusion sampling Liao et al. [2025], and latent world models Li et al. [2025], Zheng et al. [2025], Zhang et al. [2026b]. While these models represent the state of the art in Planning Decision Metric Score (PDMS) on simulators like NAVSIM Dauner et al. [2024], their reliance on heavy dense decoder stacks results in immense multiply-accumulate (MAC) counts, creating a bottleneck for real-time edge deployment. SDPAD matches state-of-the-art planning performance while systematically replacing dense computation with spike-driven operations throughout the entire pipeline.

2.2 Spiking Neural Networks

SNNs have achieved competitive accuracy in image classification and object detection through both ANN-to-SNN conversion Ding et al. [2021] and direct surrogate-gradient training Zhang et al. [2022, 2026a], and recent spike-driven transformers demonstrate that attention architectures can also operate entirely on spike events at scale Huang et al. [2024], Qiu et al. [2025], Yao et al. [2025]. SNNs are also applied for navigation planning of wheeled robots Tang et al. [2020], Zhang et al. [2024]. Meanwhile, SNNs have also been extensively applied to perceptual tasks in autonomous driving, including object detection Vicente-Sola et al. [2025], Zhang et al. [2026a] and semantic segmentation Lei et al. [2025]. In autonomous driving planning, however, SNNs have predominantly been confined to open-loop, single-modality perception tasks, with SAD Zhu et al. [2024] serving as a notable benchmark for end-to-end

spiking planning on nuScenes. Other specialized endeavors have explored SNNs for spatial path finding Espino et al. [2024] or hybrid designs Fan et al. [2025] leveraging SNNs solely for low-power temporal feature. Nevertheless, current frameworks rarely rely entirely on pure SNN architectures across all processing stages. Therefore, our model aims to develop a fully spike-driven architecture specifically tailored for motion planning in autonomous driving.

2.3 Query-Based Scene Understanding for Planning

Query-based decoders, such as Q-Former Li et al. [2023] and deformable attention Zhu et al. [2021], are pivotal in modern vectorized planners Jiang et al. [2023], Sun et al. [2025] for distilling sparse structural queries from dense perception maps. However, their dependence on floating-point Softmax operators fundamentally limits their deployability on neuromorphic hardware. To mitigate this hardware constraint, recent advances in Spiking Transformers—such as Spikformer Zhou et al. [2022] and Spike-driven Transformer Yao et al. [2023]—have successfully eliminated Softmax operations by reformulating self-attention into fully spike-driven, addition-only interactions. Despite these breakthroughs, existing Spiking Transformer architectures are predominantly tailored for standard backbone feature extraction and lack specialized decoders capable of handling multi-task, query-based representation learning. To bridge this gap, we propose Spike-QFormer, which extends the query-based bottleneck paradigm to the spiking domain for end-to-end autonomous driving.

3 Method

3.1 Overall Architecture

The overall architecture of our proposed SDPAD is depicted in Fig. 2. Firstly, multi-view images ($6 \times 3 \times H \times W$) are processed by the E-Spikeformer Yao et al. [2025] backbone to yield image features and spiking depth scores, which are subsequently lifted to the BEV space via Spike-3D-Lift, forming the Spiking BEV representation. Then, this representation is fed into Spike-SegHead and Spike-AgentHead for semantic segmentation and 3D object detection, respectively. Concurrently, map, ego, and agent queries from the embedding library are integrated through Spike-QFormer modules to get waypoint query. Finally, the planning stage refines waypoints based on a 3-mode reference curve and Deformable Spike-Cross-Attention (SCA), and employs a Spike-MLP to output the final predicted trajectory for the ego vehicle.

3.2 Preliminary

Spiking Neuron Dynamics. We employ the iterative Leaky Integrate-and-Fire (I-LIF) model neuron with a soft-reset mechanism Guo et al. [2024]. At time step t , its discrete-time state transitions are formulated as:

$$U[t] = H[t - 1] + X[t], \quad (1)$$

$$S[t] = \theta(U[t] - V_{th}), \quad (2)$$

$$H[t] = U[t] - V_{th} \cdot S[t], \quad (3)$$

where $U[t]$, $H[t]$, and $X[t]$ denote the pre-fire, post-reset membrane potentials, and spatial input, respectively. V_{th} is the firing threshold, $S[t] \in \{0, 1\}$ is the output spike, and $\theta(\cdot)$ is the Heaviside step function.

Integer ANN2SNN Conversion. In rate-coded ANN-to-SNN conversion over a temporal window T , the layer- l SNN firing rate maps to the ANN activation via $a_l^T = \frac{1}{T} \sum_{t=1}^T S_l[t]$. To minimize the conversion gap, the source integer ANN is optimized using a quantized-clip activation:

$$a_l^T = h(U^l) = \frac{1}{T} \lfloor \text{clip}(U^l, 0, T) \rfloor, \quad (4)$$

where $\lfloor \cdot \rfloor$ denotes the rounding operation and U^l is the pre-activation potential. Upon conversion, the total accumulated SNN membrane potential over T steps is given by $U^l = H^l[0] + (\frac{1}{T}W^l) \sum_{d=1}^T S^{l-1}[t]$, where W^l is the weight matrix. Consequently, the sequential dynamics in Eq. (2) can be parallelized over the temporal window as:

$$\{S^l[t] = \theta(U^l - V_{th} \cdot t)\}_{t=1}^T. \quad (5)$$

This integer-driven framework effectively minimizes the mapping error and mitigates residual potential variance.

3.3 Spike-Driven-Max for Spike-3D-Lift

To achieve energy-efficient 2D-to-3D feature lifting (e.g., in LSS architectures Philion and Fidler [2020]) while preserving accurate spatial layout, we propose the Spike-Driven-Max (SDM) mechanism, as shown in Fig. 3. Unlike traditional methods that rely on dense, floating-point Softmax depth distributions, SDM quantizes the continuous depth estimation into discrete, integer-valued spike intensities, enabling hardware-friendly spike-driven matrix multiplication.

Specifically, given the continuous depth input D , we first restrict its range and apply a rounding operation to generate discrete spike states $\hat{D}$:

$$\hat{D} = \lfloor \text{clip}(D, D_{\min}, D_{\max}) \rfloor, \quad (6)$$

To capture the non-linear significance of deeper ranges, we employ an exponential mapping to yield the raw spike activation $M = \hat{D} \cdot 2^{\hat{D} - D_{\max}}$. To bridge the quantization gap flexibly, we introduce a learnable normalization factor $N = D_{\max} + \text{softplus}(\alpha)$, where α is a trainable parameter

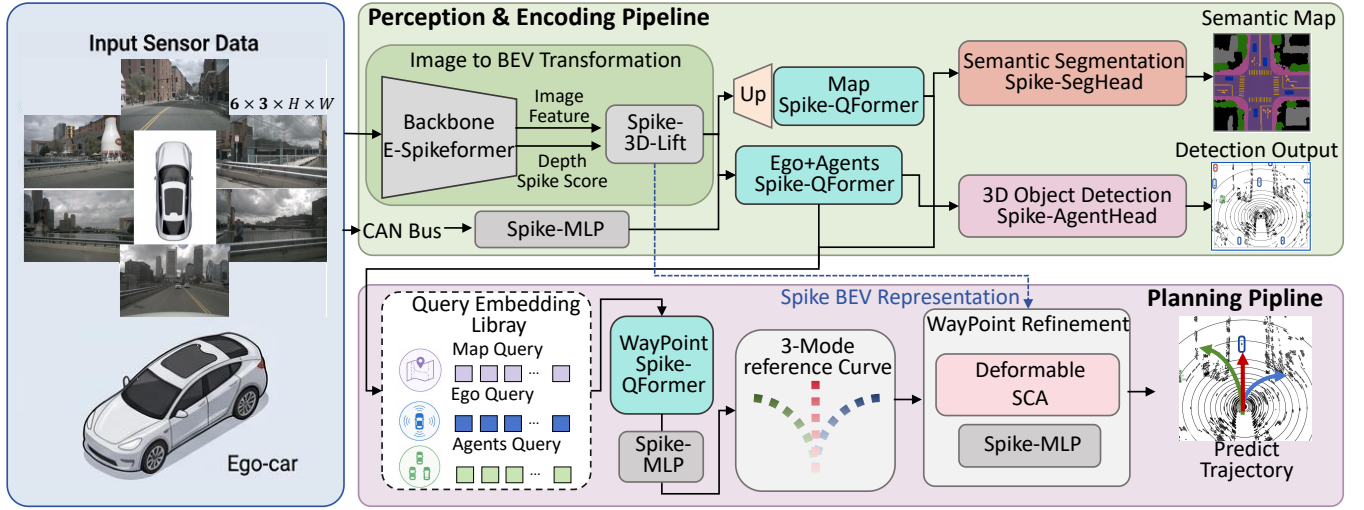


Figure 2: Overall architecture of the proposed Pipeline. Multi-view images are encoded into a Spiking BEV representation via E-Spikeformer and Spike-3D-Lift, which simultaneously supports semantic segmentation and 3D object detection. The planning stage then refines waypoints with a 3-mode reference curve and Deformable SCA, and finally outputs the ego-vehicle trajectory.

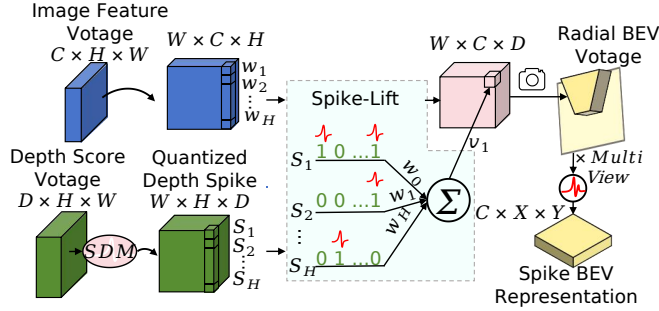


Figure 3: The illustration of the Spike-3D-Lift for Spike-BEV generation process. Image features are weighted by the accumulated spiking depth scores to form 3D frustum features (membrane potentials), followed by camera-based interpolation and spiking neural conversion to Spiking BEV Representation.

initialized dynamically. The final normalized sparse spike output S_D is formulated as:

$$S_D = \frac{\max(M, \epsilon)}{N}, \quad (7)$$

where ϵ is a small constant to prevent division by zero. Since the rounding operation in Eq. 6 is non-differentiable, we adopt the Straight-Through Estimator (STE) in the backward pass, approximating the gradient as $\frac{\partial \hat{D}}{\partial D} \approx 1$ within the active range $[D_{\min}, D_{\max}]$.

Finally, the 3D frustum membrane potential $V_{\mathcal{F}}$ is constructed via matrix multiplication between the discrete spiking depth score S_D and the continuous visual context features I :

$$V_{\mathcal{F}} = \mathcal{P}(S_D) \otimes \mathcal{P}(I), \quad (8)$$

where $\mathcal{P}(\cdot)$ represents the tensor permutation adjusting axes for batch matrix alignment. Because S_D represents scaled discrete spike states rather than dense floating-point probabilities, Eq. 9 degenerates into a spike-driven linear accumulation in neuromorphic hardware, drastically

reducing computational FLOPs while preserving representation capacity. We then draw inspiration from the Radial-Cartesian BEV Sampling technique Zhang et al. [2025]. In our approach, we first pre-define the coordinates of the Cartesian Spiking BEV features $S_{\text{bev}} \in \mathbb{R}^{C \times X \times Y}$, and subsequently project them onto the current membrane potential $V_{\mathcal{F}} \in \mathbb{R}^{C \times D \times W}$. After this projection, spike firing is triggered in the neurons. The operation can be formally expressed as:

$$S_{\text{bev}}(x, y) = \mathcal{SN}(V_{\mathcal{F}}|_{\text{Proj}(x, y)}^{\text{Bilinear}}), \quad (9)$$

where $\text{Proj}(x, y)$ denotes the coordinates of the projected point (x, y) on $V_{\mathcal{F}}$, bilinear sampling is utilized to retrieve the corresponding membrane potential, $\mathcal{SN}(\cdot)$ denotes spiking neuron layer with I-LIF neuron.

3.4 Spike-QFormer for Waypoint Decoding

Inspired by the Q-Former architecture Li et al. [2023], which employs learnable queries to extract task-relevant information from frozen visual features via cross-attention, we propose the **Spike-QFormer**, a fully spike-driven querying transformer that aggregates heterogeneous driving context into waypoint predictions. As illustrated in Fig. 2, the Spike-QFormer is built upon a single shared primitive—*spiking attention*—which is instantiated into two query decoders that progressively distill the BEV scene representation into three complementary query groups (ego, agent, and map), followed by a waypoint query decoder that fuses them for trajectory generation.

Spiking attention primitive. All query decoders in the Spike-QFormer share a unified spiking attention operator. Given a learnable embedding as input query X_q and a key-value sequence X_{kv} , spike projections are obtained as $\tilde{Q} = \mathcal{SN}(W_q \mathcal{SN}(X_q))$, $\tilde{K} = \mathcal{SN}(W_k \mathcal{SN}(X_{kv}))$

and $\tilde{V} = \mathcal{SN}(W_v \mathcal{SN}(X_{kv}))$, where the inputs are quantized into integer spike trains by $\mathcal{SN}(\cdot)$ both before and after each linear projection. The attention output is then computed without softmax normalization:

$$\mathcal{SA}(X_q, X_{kv}) = \mathcal{SN}(\tilde{Q} \tilde{K}^\top \tilde{V} \cdot 2v_{scale}) W_o, \quad (10)$$

where v_{scale} is the scaling constant. Since $\tilde{Q}$, $\tilde{K}$ and $\tilde{V}$ are all integer-valued spike tensors, both matrix multiplications degenerate into spike-gated accumulations, and the softmax, whose exponentiation and division are costly on neuromorphic hardware, is entirely removed. A spike-driven MLP follows the attention in each decoder layer, and residual connections are preserved.

Multi-source query construction. We construct three complementary query groups from the heterogeneous driving context. All groups are learnable embeddings that extract information from the BEV representation through spiking decoder layers, each consisting of spiking self-attention, spiking cross-attention and a spike-driven MLP. The ego query $\mathbf{Q}_{ego} \in \mathbb{R}^{1 \times D}$ and the agent queries $\mathbf{Q}_{agent} \in \mathbb{R}^{N_a \times D}$ ($N_a=30$) summarize the ego vehicle’s kinematic state and the instance-level motion of surrounding participants, respectively. To inject the ego status, the CAN-bus signals and the one-hot navigation command are encoded into a status token $\mathbf{v}_{status} = W_s[\mathbf{c}_{can}; \mathbf{c}_{cmd}]$, which is appended to the downsampled 10×10 BEV grid $\mathbf{F}_{bev}^a \in \mathbb{R}^{100 \times D}$:

$$[\mathbf{Q}_{ego}; \mathbf{Q}_{agent}] = D_{sn}^3(\mathbf{E}_{ta}, [\text{flt}(\mathbf{F}_{bev}^a); \mathbf{v}_{status}] + \mathbf{E}_{kv}), \quad (11)$$

where $\text{flt}(\cdot)$ denotes the flattening operation, $\mathbf{E}_{ta}$ and $\mathbf{E}_{kv}$ are learned query and key embeddings and D_{sn}^3 stacks three spiking decoder layers. The agent queries are additionally supervised by an auxiliary agent-state prediction head, encouraging explicit instance-level representations. Meanwhile, the map queries $\mathbf{Q}_{map} \in \mathbb{R}^{N_b \times D}$ ($N_b=16$) distill navigation-relevant semantics, road geometry, lane topology and drivable area, from the full-resolution BEV features, which are fused with the agent-view BEV features through a spike-driven projection:

$$\mathbf{Q}_{map} = D_{sn}^3(\mathbf{E}_{map}, \text{flt}(\hat{\mathbf{F}}_{bev})), \quad (12)$$

where $\hat{\mathbf{F}}_{bev}$ is the fused BEV feature map.

Waypoint querying via cross-attention. The waypoint decoder treats learnable waypoint tokens as information-seeking queries attending to the fused multi-source context. We maintain waypoint embeddings $\mathbf{E}_{wp}, \mathbf{E}_{pos} \in \mathbb{R}^{K \times L \times 2D}$, where $K=3$ denotes the navigation commands (*left*, *straight*, *right*) and L is the prediction length. The three query groups are concatenated into a unified context $\mathbf{Q}_{fused} = [\mathbf{Q}_{ego}; \mathbf{Q}_{agent}; \mathbf{Q}_{map}]$, and a single-layer spiking cross-attention generates the waypoint representations:

$$\bar{\mathbf{W}} = \mathcal{SA}(\mathbf{E}_{wp} + \mathbf{E}_{pos}, \mathbf{Q}_{fused}). \quad (13)$$

Finally, a Spike-MLP decodes each waypoint token into per-step displacements, which are cumulatively summed

to construct K candidate trajectories. The trajectory corresponding to the active navigation command is then selected as the final planning output.

3.5 Deformable Spike-Cross-Attention

To ground the waypoint tokens in fine-grained spatial evidence, we refine them with a Deformable Spike-Cross-Attention (Deformable SCA) module, which couples the adaptive spatial sampling of deformable attention Zhu et al. [2021] with an all-spike computation paradigm. Unlike conventional deformable attention that relies on continuous softmax weights, Deformable SCA replaces every floating-point attention computation with spike-driven operations.

Given waypoint tokens $\mathbf{q} \in \mathbb{R}^{N_q \times D}$, BEV membrane potential $\mathbf{v} \in \mathbb{R}^{N_v \times D}$ and reference points $\mathbf{p} \in [0, 1]^{N_q \times 2}$ given by the initial trajectory, both inputs are first quantized by the spiking neuron layer $\mathcal{SN}(\cdot)$, and the sampling offsets, attention weights and value projection are predicted from the spiking features:

$$\Delta \mathbf{p} = W_p \tilde{\mathbf{q}}, \quad \mathbf{A} = \mathcal{SN}(W_a \tilde{\mathbf{q}}), \quad \mathbf{V} = W_v \tilde{\mathbf{v}}, \quad (14)$$

where $\tilde{\mathbf{q}} = \mathcal{SN}(\mathbf{q})$ and $\tilde{\mathbf{v}} = \mathcal{SN}(\mathbf{v})$. The sampling locations are obtained by combining the reference points with the predicted offsets, normalized by the spatial shape (X_l, Y_l) of each feature level: $\hat{\mathbf{p}}_{hlk} = \mathbf{p} + \Delta \mathbf{p}_{hlk} \odot (X_l, Y_l)$. Deformable SCA then aggregates the sampled features with the spiking attention weights through bilinear sampling:

$$\text{DeformSCA}(\mathbf{q}, \mathbf{v}, \mathbf{p}) = \mathcal{SN}\left(\sum_{h=1}^H \sum_{l=1}^L \sum_{k=1}^P \mathbf{A}_{hlk} \odot \mathbf{V}_{hlk}^{\text{bilinear}}\right) W_o + \mathbf{q}, \quad (15)$$

where h indexes H attention heads, k indexes P sampling points, and a residual connection with the input tokens is preserved.

3.6 Loss

The overall training objective combines four task-specific losses.

BEV segmentation loss. A cross-entropy loss over the 200×200 BEV grid for semantic map supervision:

$$\mathcal{L}_{seg} = - \sum_c y_c \log(\hat{y}_c), \quad (16)$$

where c indexes the semantic classes.

Agent detection loss. A binary cross-entropy loss on the existence logits of the $N_a=30$ agent slots plus an L_1 regression loss on their states:

$$\mathcal{L}_{agent} = \lambda_{cls} \mathcal{L}_{BCE}(\hat{p}, p^*) + \lambda_{box} \mathcal{L}_{L1}(\hat{\mathbf{s}}, \mathbf{s}^*), \quad (17)$$

with $\lambda_{cls}=0.85$ and $\lambda_{box}=0.1$.

Planning loss. A command-conditioned L_1 loss over valid future horizons, applied to both the initial trajectories ($\mathcal{L}_{plan}^{\text{init}}$) and the refined ones ($\mathcal{L}_{plan}^{\text{ref}}$):

$$\mathcal{L}_{plan} = \frac{1}{Z} \sum_t w_t^{\text{cmd}} \cdot m_t \cdot |\hat{\tau}_t - \tau_t^*|_1, \quad (18)$$

where w_t^{cmd} is the command-guided mode weight, m_t masks invalid timesteps, and Z is the number of valid steps.

Depth loss. A binary cross-entropy loss $\mathcal{L}_{\text{depth}}$ supervising the depth distribution in Spike-3D-Lift. The total loss is

$$\mathcal{L} = \lambda_{\text{seg}}\mathcal{L}_{\text{seg}} + \mathcal{L}_{\text{agent}} + \lambda_{\text{plan}}(\mathcal{L}_{\text{plan}}^{\text{init}} + \mathcal{L}_{\text{plan}}^{\text{ref}}) + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}, \quad (19)$$

with $\lambda_{\text{seg}}=1.1$, $\lambda_{\text{plan}}=10.0$ and $\lambda_{\text{depth}}=0.1$.

4 Experiments

4.1 Benchmarks

Open-Loop Metrics on nuScenes. Developed to capture diverse real-world driving environments, the nuScenes benchmark Caesar et al. [2020] provides 1,000 driving scenes for open-loop testing. We evaluate the predicted future trajectories, sampled at 2 Hz for a duration of 3 seconds, by calculating the L_2 displacement error and Collision Rate (CR), remaining consistent with existing methodologies Hu et al. [2023], Jiang et al. [2023].

Closed-Loop Assessment in NAVSIM. Based on the extensive OpenScene dataset Contributors [2023] ($> 100,000$ keyframes across 1,192 train and 136 test scenarios), the NavSim simulator Dauner et al. [2024] serves as our closed-loop validation environment. To align with the official baseline, an LQR controller is applied to interpolate the 4-second, 2 Hz predicted trajectories. Ultimately, the framework evaluates driving behavior via the Planning Decision Metric Score (PDMS), which integrates five essential driving criteria: No at-fault Collision (NC), Drivable Area Compliance (DAC), Time-to-Collision (TTC), Comfort (Comf.), and Ego Progress (EP).

4.2 Implementation Details

nuScenes. We adopt the large model of E-SpikeFormer Yao et al. [2025] (19M parameters) as the feature extractor for our model. Multi-view images downsampled to 256×704 are used as input. We adopt Radial-Cartesian-Sampling Zhang et al. [2025] as the BEV feature projection method, where membrane potentials are projected into BEV features based on spike proportion before computing spike firing. For navigation commands, we followed the work Hu et al. [2023], comprising three types: left, right, and straight. In the decoder, we generate three planning trajectories (each with 6 points over 3 seconds) and select the one corresponding to the navigation command. The model is trained on 2 NVIDIA A800 GPUs with a total batch size of 20 for 30 epochs. We use the AdamW optimizer with an initial learning rate of 6×10^{-4} , decayed using a step schedule. We also verify the effect of auxiliary tasks on the final waypoint prediction. We further conduct ablation studies on the nuScenes benchmark

to evaluate the effectiveness of the proposed components and the impact of spiking neural networks on performance.

NAVSIM. For experiments conducted on NAVSIM benchmark, we use two E-SpikeFormer backbone networks equipped with spiking attention to process the concatenated images and LiDAR BEV maps, similar to Tansfuser Chitta et al. [2022]. Based on the fusion of LiDAR and image information in the backbone with spiking attention, this architecture can directly use the final LiDAR features as BEV features for Spike-Qformer. Waypoints are predicted as an 8-waypoint trajectory over 4 seconds. The model is also trained on 2 NVIDIA A800 GPUs for 100 epochs with a learning rate of 6×10^{-4} .

4.3 Main Results

nuScenes. We conducted comprehensive comparisons between SDPAD and existing autonomous driving methods on the nuScenes benchmark as shown in Tab. 1. It can be found that SDPAD achieves highly competitive planning accuracy compared to state-of-the-art ANN methods. Specifically, our method delivers comparable performance to heavy ANN baselines such as SparseDrive Sun et al. [2025] and DiffusionDrive Liao et al. [2025], while inherently benefiting from the ultra-low energy consumption and temporal sparsity of SNNs. Crucially, when compared with recent SNN-based driving models, SDPAD demonstrates a significant advantage over SAD Zhu et al. [2024], achieving superior trajectory prediction accuracy. On nuScenes, the use of auxiliary tasks does not lead to a significant improvement in waypoint prediction, with the 3-second L2 error differing by merely 0.01. This indicates that our fully spike-driven framework retains robust spatiotemporal feature extraction capabilities, successfully bridging the performance gap between neuromorphic architectures and dense ANN models without relying on power-hungry computations.

NAVSIM. We also evaluated the closed-loop planning accuracy of SDPAD on the NAVSIM benchmark, and the experimental results are presented in Tab. 2. To conduct a fair comparison, we evaluated our model against both ANN-based and SNN-based planners. Notably, SDPAD achieves 86.3% PDMS, exhibiting planning accuracy that is highly comparable to mainstream ANN methods like Hydra-MDP Li et al. [2024] and World4Drive Zheng et al. [2025], yet operating at a fraction of their computational footprint. Furthermore, our proposed method outperforms existing spiking models such as SAD by a notable margin of 4.3 PDMS points, proving the efficacy of our architectural design in handling complex closed-loop scenarios. On NAVSIM, our method with auxiliary tasks contributes a 1% improvement to the final PDMS. The complete SDPAD leverages the integer-spike representation of spiking neurons to represent the dynamic information of the driving scene, demonstrating that a fully spike-driven pipeline can maintain exceptional safety and planning precision in realistic navigation tasks.

Table 1: Comparison on nuScenes dataset with open-loop metrics. metric calculated in the same way as VAD

Method	SNN	Img. Backbone	Auxiliary Task	L2 (m) ↓				Collision Rate (%) ↓				Energy (mJ) ↓
				1s	2s	3s	Avg.	1s	2s	3s	Avg.	
ST-P3 Hu et al. [2022]	✗	EffNet-b4	Det& Map	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71	3520.40
UniAD Hu et al. [2023]	✗	ResNet-101	Det&Track&Map&Motion&Occ	0.45	0.70	1.04	0.73	0.62	0.58	0.63	0.61	-
VAD Jiang et al. [2023]	✗	ResNet-50	Det&Map&Motion	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22	-
SparseDrive Sun et al. [2025]	✗	ResNet-50	Det&Track&Map&Motion	0.29	0.58	0.96	0.61	0.01	0.05	0.18	0.08	-
DiffusionDrive Liao et al. [2025]	✗	ResNet-50	Det&Track&Map&Motion	0.27	0.54	0.90	0.57	0.03	0.05	0.16	0.08	-
ResWorld Zhang et al. [2026b]	✗	ResNet-50	None	0.14	0.27	0.49	0.30	0.01	0.03	0.14	0.06	4103.74
SAD Zhu et al. [2024]	✓	STM	Det&Map	1.53	2.35	3.21	2.36	0.62	1.26	2.38	1.41	46.92
SDPAD(ours)	✓	E-SpikeFormer	None	0.20	0.37	0.65	0.41	0.02	0.09	0.27	0.13	65.88
			Det& Map	0.19	0.36	0.64	0.40	0.02	0.09	0.26	0.12	69.93

Table 2: Comparison of state-of-the-art methods on the NAVSIM navtest split.

Method	SNN	Img. Backbone	Auxiliary Task	NAC ↑	DAC ↑	TTC ↑	Comf. ↑	EP ↑	PDMS ↑	Energy (mJ) ↓
Transfuser Chitta et al. [2022]	✗	ResNet-34	Det&Map	97.7	92.8	92.8	100	79.2	84.0	114.46
UniAD Hu et al. [2023]	✗	ResNet-34	Det&Map	97.8	91.9	92.9	100	78.8	83.4	-
VADv2 Chen et al. [2024b]	✗	ResNet-34	Det&Map	97.2	89.1	91.6	100	76.0	80.9	-
Hydra-MDP-W-EP Li et al. [2024]	✗	ResNet-34	Det&Map	98.3	96.0	94.6	100	78.7	86.5	-
LAW Li et al. [2025]	✗	ResNet-34	None	96.4	95.4	88.7	99.9	81.7	84.6	-
World4Drive Zheng et al. [2025]	✗	ResNet-34	None	97.4	94.3	92.8	100	79.9	85.1	-
DiffusionDrive Liao et al. [2025]	✗	ResNet-34	Det&Map	98.2	96.2	94.7	100	82.2	88.1	153.22
ResWorld Zhang et al. [2026b]	✗	ResNet-34	Det&Map	98.9	96.5	95.6	100	83.1	89.0	-
SAD Zhu et al. [2024]	✓	STM	Det&Map	96.3	92.2	91.0	98.9	77.7	82.0	20.92
SDPAD(ours)	✓	E-SpikeFormer	None	97.8	93.9	93.6	99.9	79.9	85.3	32.34
			Det&Map	97.7	95.1	93.8	100	80.7	86.3	34.38

ID	Ego Query	Agent Query	Map Query	Deformable -SCA	Param. (M)↓	L2 (m) ↓			Collision Rate (%) ↓			Energy (mJ) ↓
						1s	2s	3s	1s	2s	3s	
1	✗	✗	✓	✗	34.91	0.28	0.57	0.98	0.04	0.19	0.44	69.27
2	✓	✗	✓	✗	38.11	0.27	0.56	0.97	0.02	0.16	0.51	68.71
3	✗	✓	✓	✗	38.38	0.20	0.39	0.70	0.02	0.12	0.35	69.58
4	✓	✓	✗	✗	35.21	0.21	0.38	0.67	0.02	0.11	0.34	70.02
5	✓	✓	✓	✗	38.39	0.20	0.38	0.66	0.02	0.12	0.30	69.64
6	✓	✓	✓	✓	38.70	0.19	0.36	0.64	0.02	0.09	0.26	69.93

Table 3: Ablation for design choices.

quant- steps	L2 (m) ↓			Collision Rate (%) ↓			Energy (mJ) ↓	Depth Att.	MFLOPs (Att.)	mIoU (%) ↑	mAP:0.5 (%) ↑	L2 (m) ↓			Collision Rate (%) ↓		
	1s	2s	3s	1s	2s	3s						1s	2s	3s	1s	2s	3s
2	2.32	3.81	5.32	2.05	4.94	6.12	79.08	Softmax	45.96	38.79	15.30	0.19	0.37	0.65	0.02	0.09	0.26
3	0.22	0.40	0.66	0.02	0.09	0.31	84.74	SpareMax	10.25	29.06	10.37	0.20	0.37	0.66	0.04	0.11	0.32
4	0.20	0.38	0.66	0.02	0.11	0.26	76.64	I-LIF	23.06	17.48	2.23	0.21	0.39	0.68	0.04	0.12	0.39
8	0.19	0.36	0.64	0.02	0.09	0.26	69.93	SDM	35.13	36.60	13.94	0.19	0.36	0.64	0.02	0.09	0.26

Table 4: Ablation study of SNN quantization steps.**Table 5: Ablation study of depth attention for BEV.**

4.4 Ablation Studies

Effect of designs in spiking decoder. Tab. 3 shows the contribution of each component in our proposed decoder. ID-1 (only Map Query) serves as the baseline, achieving an L2 of 0.98 *m* and a collision rate of 0.44%. Adding the Ego Query (ID-2) or the Agent Query (ID-3) individually reduces the L2 to 0.97 *m* and 0.70 *m* respectively, demon-

strating the importance of ego and agent information. When both ego and agent query are included without the Map Query (ID-4), the L2 further drops to 0.67 *m*. The full combination of Ego, Agent and Map Query (ID-5) yields an L2 of 0.66 *m* and a collision rate of 0.30%. Finally, incorporating Deformable-SCA in ID-6 achieves the best performance: 0.64 *m* L2 and 0.26% collision rate,

with only a modest increase in parameters (39M vs. 35M of baseline). This validates that rich query interactions and attention mechanisms are crucial for high-quality planning.

Efficiency of quantization steps in SNN. Since our integer-spike representation encodes the firing count of T conversion steps in a single value, varying T_{max} is equivalent to varying the effective time window of the converted SNN, without actually unrolling it at inference. Tab. 4 studies the effect of the number of quantization steps in our spiking neural network. Increasing the quantization steps from 2 to 8 consistently improves the planning quality: L2 decreases from 5.32 m to 0.64 m , and the collision rate drops from 6.12% to 0.26%. However, higher quantization steps do not increase energy: the energy cost even decreases slightly from 84.73 mJ (quant-steps 3) to 69.93 mJ (quant-steps 8), because coarser quantization produces denser spike firing. Notably, even with only 2 quantization steps, our model already achieves reasonable results, and additional steps provide further gains in complex scenarios.

Depth attention for BEV construction. To validate the efficacy of our SDM within the Spike-3D-Lift mechanism, we conduct comparative experiments against Softmax, SparseMax Martins and Astudillo [2016], and I-LIF variants, with the results summarized in Tab. 5. Compared to the conventional Softmax attention, our SDM module reduces computational overhead by 23.6% (from 45.96 to 35.13 MFLOPs) while maintaining competitive perception performance (36.60% mIoU and 13.94% mAP@0.5). Crucially, SDM achieves superior or on-par motion planning accuracy across all prediction horizons (1s–3s), yielding an L2 error of 0.64m and a low collision rate of 0.26% at 3s. While SparseMax and I-LIF lower computational complexity further, they suffer from severe degradation in perception fidelity—most notably I-LIF’s drop to 17.48% mIoU—which adversely propagates to planning safety. These results demonstrate that SDM strikes an optimal trade-off between energy/computational efficiency and end-to-end task performance.

5 Conclusion

In this work, we have presented SDPAD, a fully spike-driven pipeline for end-to-end autonomous driving, by incorporating integer ANN2SNN conversion, a spike-driven-max mechanism for BEV lifting, and the Spike-QFormer planning decoder with deformable spike-cross-attention refinement. SDPAD performs the entire perception-to-planning computation on integer spike representations in a single feed-forward pass, without any temporal simulation loops. Comprehensive experiments on nuScenes and NAVSIM validate the superiority of SDPAD in planning quality, energy efficiency, and closed-loop safety, demonstrating that spiking architectures can rival dense ANN planners in complex driving tasks at a small fraction of their energy cost. We believe SDPAD provides a blueprint

for future ultra-efficient edge AI vehicles, and leave direct surrogate-gradient training and on-chip neuromorphic deployment as future work.

References

- Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024a.
- Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xingang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024b.
- Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.
- OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Vancouver, Canada*, pages 18–22, 2023.
- Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.
- Jianhao Ding, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. *arXiv preprint arXiv:2105.11654*, 2021. URL <https://arxiv.org/abs/2105.11654>.
- Harrison Espino, Robert Bain, and Jeffrey L Krichmar. A rapid adapting and continual learning spiking neural network path planning algorithm for mobile robots. *IEEE Robotics and Automation Letters*, 9(11):9542–9549, 2024.
- Yuqi Fan, Zhiyong Cui, Zhenning Li, Yilong Ren, and Haiyang Yu. Sah-drive: A scenario-aware hybrid planner for closed-loop vehicle trajectory generation. *arXiv preprint arXiv:2505.24390*, 2025.

- Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 12244–12252, 2024.
- Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022.
- Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhui Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.
- Zihan Huang, Xinyu Shi, Zecheng Hao, Tong Bu, Jianhao Ding, Zhaoqi Yu, and Tiejun Huang. Towards high-performance spiking transformers from ann to snn conversion. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 10688–10697, 2024.
- Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.
- Zhenxin Lei, Man Yao, Jiakui Hu, Xinhao Luo, Yanye Lu, Bo Xu, and Guoqi Li. Spike2former: Efficient spiking transformer for high-performance image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1364–1372, 2025.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.
- Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *International Conference on Learning Representations*, volume 2025, pages 42942–42959, 2025.
- Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.
- Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12037–12047, 2025.
- Fernando S Martínez, Jordi Casas-Roma, Laia Subirats, and Raúl Parada. Spiking neural networks for autonomous driving: A review. *Engineering Applications of Artificial Intelligence*, 138:109415, 2024.
- Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016.
- Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision*, pages 194–210. Springer, 2020.
- Xuerui Qiu, Jieyuan Zhang, Wenjie Wei, Honglin Cao, Junsheng Guo, Rui-Jie Zhu, Yimeng Shan, Yang Yang, Malu Zhang, and Haizhou Li. Quantized spike-driven transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=5J9B7Sb8rO>.
- Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8795–8801. IEEE, 2025.
- Guangzhi Tang, Neelesh Kumar, and Konstantinos P. Michmizos. Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware. 2020.
- Alex Vicente-Sola, Davide L Manna, Paul Kirkland, Gaetano Di Caterina, and Trevor J Bihl. Spiking neural networks for event-based action recognition: A new task to understand their advantage. *Neurocomputing*, 611: 128657, 2025.
- Bo Yang, Shibo Zhou, Chaohui Lin, Qingao Chai, Rui Yan, De Ma, Gang Pan, and Huajin Tang. Hsrl: A hierarchical control system based on spiking deep reinforcement learning for robot navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13384–13390. IEEE, 2025a.
- Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9327–9335, 2025b.
- Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36: 64043–64058, 2023.
- Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu,

- and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2973–2990, 2025.
- Chengjun Zhang, Ka Wa Yip, Bo Yang, Zhiyong Zhang, Mengwen Yuan, Rui Yan, and Huajin Tang. Casrl: Collision avoidance with spiking reinforcement learning among dynamic, decision-making agents. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8031–8038, 2024.
- Chengjun Zhang, Yuhao Zhang, Jisong Yu, Jie Yang, and Mohamad Sawan. Event-fused hybrid ann-snn architecture for low-latency object detection in automotive vision. *IEEE Robotics and Automation Letters*, 11, 2026a.
- Duzhen Zhang, Shuncheng Jia, and Qingyu Wang. Recent advances and new frontiers in spiking neural networks. *arXiv preprint arXiv:2204.07050*, 2022. URL <https://arxiv.org/abs/2204.07050>.
- Jinqing Zhang, Yanan Zhang, Yunlong Qi, Zehua Fu, Qingjie Liu, and Yunhong Wang. Geobev: Learning geometric bev representation for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9960–9968, 2025.
- Jinqing Zhang, Zehua Fu, Zelin Xu, Wenying Dai, Qingjie Liu, and Yunhong Wang. Resworld: Temporal residual world model for end-to-end autonomous driving. *arXiv preprint arXiv:2602.10884*, 2026b.
- Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, et al. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28632–28642, 2025.
- Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spik-former: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*, 2022.
- Rui-Jie Zhu, Ziqing Wang, Leilani Gilpin, and Jason K Eshraghian. Autonomous driving with spiking neural networks. *Advances in Neural Information Processing Systems*, 37:136782–136804, 2024.
- Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021.

Supplementary Material

This supplementary material provides implementation details, detailed model architecture specifications, additional experiments for 3D object detection and qualitative visualizations that could not be included in the main paper due to space limitations.

A Implementation Details

A.1 ANN pre-training and conversion.

We first train the dense ANN counterpart of the perception stack (spike neurons replaced by ReLU, attention by standard multi-head attention) on nuScenes. The pre-trained weights are then converted by replacing every activation with the integer-spike operator $\mathcal{SN}(\cdot)$ (I-LIF with $V_{\max}=8$, Eq. (4) of the main paper) while keeping all linear layers unchanged.

A.2 Dataset and Input Settings

nuScenes. Six surround-view images are downsampled from 900×1600 to 256×704 . During training we apply random scaling in $[-0.2, 0.2]$ and horizontal flipping. The BEV grid covers $[-15 \text{ m}, 15 \text{ m}] \times [-30 \text{ m}, 30 \text{ m}]$ with $0.15 \text{ m} \times 0.3 \text{ m}$ resolution (100×100 cells); the depth range is $[1 \text{ m}, 35 \text{ m}]$ discretized with 0.5 m intervals (68 bins). Depth supervision comes from LiDAR point clouds projected onto the image planes. The planner predicts $T=6$ waypoints over 3 s at 2 Hz, conditioned on three navigation commands (*left*, *straight*, *right*) obtained from the CAN bus.

NAVSIM. We follow the official NAVSIM protocol: the input is the concatenation of the front and front-left/right camera images (and the rasterized BEV map for the map-enhanced variant), and the planner predicts 8 waypoints over 4 s. The model is trained for 100 epochs on 2 NVIDIA A800 GPUs with a learning rate of 6×10^{-4} .

A.3 Energy Measurement Methodology

We follow the standard SNN energy model based on 45 nm hardware characterization:

$$E = E_{\text{MAC}} \cdot N_{\text{MAC}} + E_{\text{AC}} \cdot N_{\text{AC}}, \quad (20)$$

with $E_{\text{MAC}}=4.6 \text{ pJ}$ and $E_{\text{AC}}=0.9 \text{ pJ}$ per 32-bit operation. Before counting, all BatchNorm layers are folded into their preceding convolutions. For each linear/convolutional layer, the input tensor is examined on the fly: if it is an integer-valued spike tensor, the operation count is scaled by the measured spike firing rate and counted as accumulate (AC) operations; otherwise it is counted as MAC operations. Firing rates are measured per layer over the full validation set and averaged. Energy figures reported in the main paper (Tables 1 and 2) correspond to

processing a single frame (nuScenes) or a single simulation step (NAVSIM). ANN baselines are measured with the identical pipeline, where all operations are MACs.

B Model Architecture Details

The detailed architecture and parameter specifications of our model are summarized in Tab B.6.

Module	Specification
Image backbone	Spike-driven transformer (large), 19.0M embed dims [64, 128, 256, 360], stride-16 output
Spike-3D-Lift	Radial-Cartesian sampling Zhang et al. [2025]
BEV encoder	SDM depth neuron, $V_{\max}=8$, BEV channels 80 channels [128, 256, 360, 256, 256], output 256 agent-view features 10×10
Query library	1 ego + 30 agent + 16 map queries, $D=256$
Ego/Agent decoder	3 spiking layers over 10×10 BEV + status
Map decoder	3 spiking layers over fused BEV
Waypoint queries	$K \times T = 3 \times 6$ tokens, dim $2D=512$
Waypoint decoder	1 spiking cross-attention layer, 8 heads
Deformable SCA	8 heads, 8 sampling points, single BEV level
Spike neuron	I-LIF, $V_{\min}=0$, $V_{\max}=8$

Table B.6: Per-stage architecture specification of SDPAD.

B.1 Spike-Driven-Max (SDM) Initialization

The learnable normalization factor of SDM is $N = D_{\max} + \text{softplus}(\alpha)$ with α initialized to -5 , so that $N \approx D_{\max}$ at the beginning of training and the normalization can adapt during fine-tuning.

B.2 Deformable SCA Initialization

Following Zhu et al. [2021], the sampling-offset weights are initialized to zero and the biases are arranged as concentric rings: head h is assigned angle $\theta_h = 2\pi h/H$, and sampling point k scales the radius linearly with $k+1$. This provides a structured prior that covers diverse spatial neighborhoods around each reference point before task-specific deformations are learned. The reference point of each waypoint token is given by the corresponding coordinate of the initial (unrefined) trajectory, so spatial sampling is anchored to the model’s own geometric prediction.

C Additional Experimental Results

Furthermore, to evaluate the representation capacity of our Spiking BEV features, we conduct experiments on 3D object detection. Specifically, we attach a spike-driven CenterNet head to the BEV features extracted by SDPAD, following the training protocol of GeoBEV. The detection results are summarized in Tab C.7. While the quantized

model exhibits a performance gap compared with state-of-the-art ANN-based baselines, it outperforms existing methods such as PETRv2. These results demonstrate that the Spiking BEV features extracted via our SDM effectively encode rich spatial and semantic representations in the Bird’s-Eye-View space.

D Qualitative Results

In this section, we provide extended qualitative evaluations on representative scenarios from both the nuScenes and NAVSIM benchmarks (Fig D.4 and Fig D.5). The visual comparisons cover intermediate Spiking BEV feature representations, multi-task perception outputs (segmentation and detection), and predicted trajectory quality against the ground truth, demonstrating faithful scene encoding and stable planning."

Method	Backbone	Image Size	Frames	mAP $\uparrow$	NDS $\uparrow$	mATE $\downarrow$	mASE $\downarrow$	mAOE $\downarrow$	mAVE $\downarrow$	mAAE $\downarrow$
BEVDet (arXiv. 2021)	ResNet50	256 $\times$ 704	1	0.298	0.379	0.725	0.279	0.589	0.860	0.245
PETrv2 (ICCV. 2023)	ResNet50	256 $\times$ 704	2	0.349	0.456	0.700	0.275	0.580	0.437	0.187
BEVDepth (AAAI. 2023)	ResNet50	256 $\times$ 704	2	0.351	0.475	0.639	0.267	0.479	0.428	0.198
BEVStereo (AAAI. 2023)	ResNet50	256 $\times$ 704	2	0.372	0.500	0.598	0.270	0.438	0.367	0.190
SA-BEV (ICCV. 2023)	ResNet50	256 $\times$ 704	2	0.387	0.512	0.613	0.266	0.352	0.382	0.199
BEFormerv2 (CVPR. 2023)	ResNet50	-	-	0.423	0.529	0.618	0.273	0.413	0.333	0.188
SOLOFusion (ICLR. 2023)	ResNet50	256 $\times$ 704	17	0.427	0.534	0.567	0.274	0.511	0.252	0.181
StreamPETR (ICCV. 2023)	ResNet50	256 $\times$ 704	8	0.450	0.550	0.613	0.267	0.413	0.265	0.196
BEVNext (CVPR. 2024)	ResNet50	256 $\times$ 704	8	0.456	0.560	0.530	0.264	0.424	0.252	0.206
RayDN (CVPR. 2024)	ResNet50	256 $\times$ 704	8	0.469	0.563	0.579	0.264	0.433	0.256	0.187
GeoBEV (AAAI. 2025)	ResNet50	256 $\times$ 704	2	0.415	0.535	0.533	0.265	0.419	0.298	0.214
SDPAD+SpikeCenterNetHead	E-Spikeformer	256 $\times$ 704	2	0.352	0.473	0.615	0.276	0.483	0.437	0.218

Table C.7: Comparison with previous state-of-the-art multi-view 3D detectors on the nuScenes *val* set.

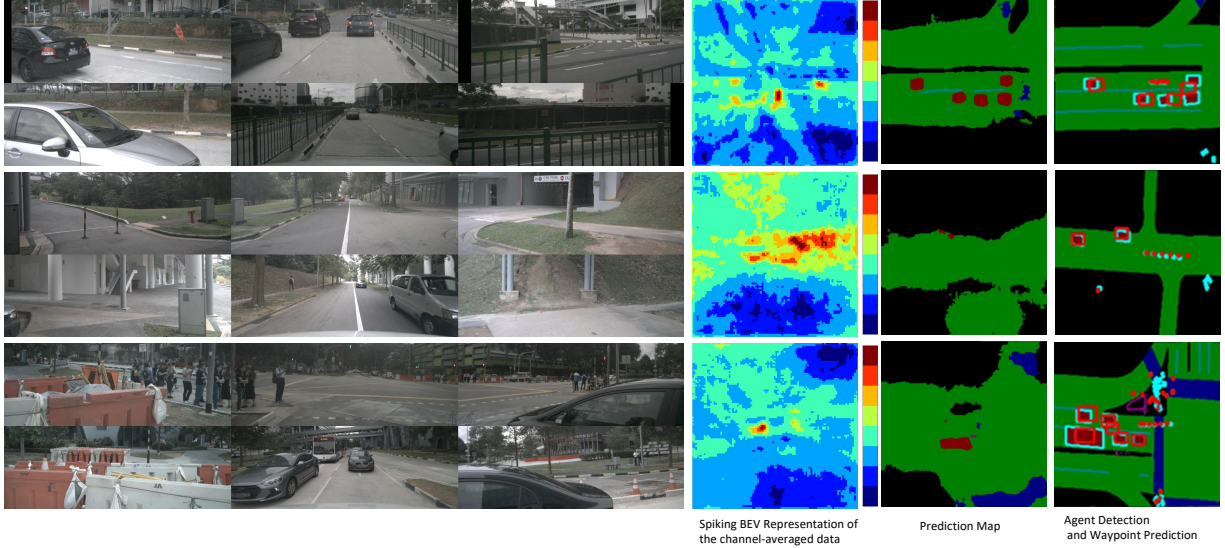


Figure D.4: Predicted trajectories of SDPAD versus ground truth in diverse nuScenes scenarios, together with the BEV segmentation and agent detection outputs. We also show the intermediate Spiking BEV features.

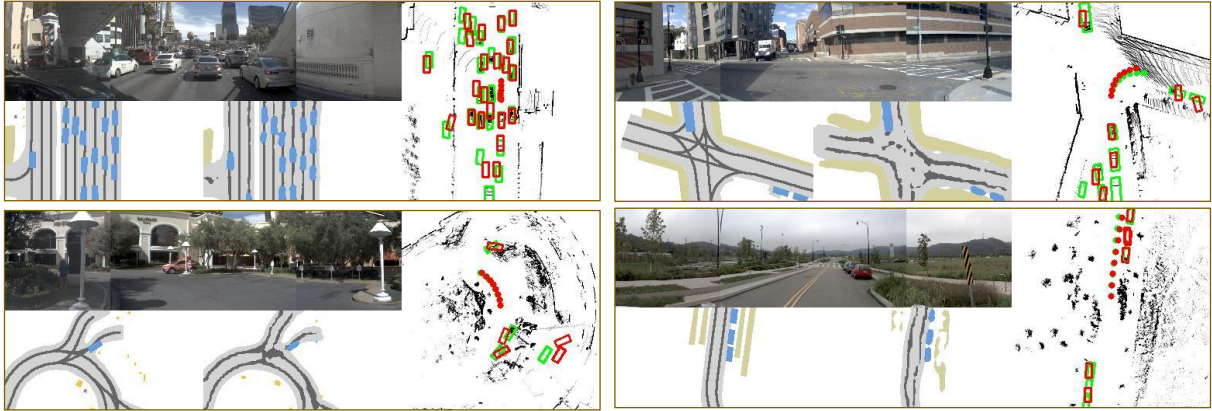


Figure D.5: Predicted trajectories of SDPAD versus ground truth in diverse NAVSIM.