Toward Neuromorphic Perception: Spike-driven Lane Segmentation for Autonomous Driving using LiDAR Sensor

Genghang Zhuang[1], Zhenshan Bing[1], Kai Huang[2], and Alois Knoll[1], *Fellow, IEEE*

***Abstract*— As a prerequisite of high vehicle autonomy, lane segmentation is a significant perception task for advanced autonomous driving. In recent years, spiking neural networks (SNNs) have garnered the attention of researchers due to their appealing power efficiency, which provides the potential to improve energy consumption for the perception system on power-constrained autonomous vehicles. In this paper, we propose a spiking neural network targeted for LiDAR sensors to solve the lane segmentation problem. By encoding the LiDAR point cloud into spikes, the proposed SNN constructed in an end-to-end fully convolutional network structure is capable of processing the LiDAR input through the network to segment the lane area effectively. Experiments conducted on the KITTI dataset for urban scenes and the power consumption evaluation demonstrate the high performance and energy efficiency of the proposed SNN for LiDAR-based lane segmentation.**

I. Introduction

Autonomous driving, at the forefront of advanced interdisciplinary technology in the transportation industry, is reliant upon precise and reliable perception. Perception for autonomous vehicles encompasses the ability to accurately sense and interpret the surrounding environment from the sensor input. It is through the perception system that autonomous vehicles are empowered to make informed decisions in order to navigate complicated real-world scenarios.

In the field of autonomous navigation technology, significant advancements have been made in developing a multitude of sensors for effective perception. Various types of sensors have been successfully deployed on autonomous vehicles and integrated into the perception system. One of the most commonly used sensors is LiDAR (Light Detection and Ranging), which utilizes laser beams to generate high-resolution 3D point clouds of the surrounding environment. In comparison with cameras, LiDARs provide accurate information on the distance, exhibiting higher reliability, precision, and robustness in various environments. Therefore, LiDAR has been extensively utilized in state-of-the-art algorithms to solve perception tasks, including mapping [1], object detection [2], and semantic segmentation [3].

Among the perception tasks for autonomous driving, lane segmentation is an integral component and a pivotal role in autonomous driving. Lane segmentation performs semantic segmentation from the sensor input to identify the drivable area in the environment. It is paramount for ensuring the safety, precision, and reliability of advanced autonomous navigation in urban scenes. The significance of lane segmentation is embodied in its multifaceted implications. By segmenting the lanes, autonomous vehicles can effectively establish a structural understanding of the road layout and simultaneously acquire contextual information on their lateral position and orientation within the driving path. Accurate lane segmentation enables the vehicle to maintain a proper trajectory within its designated lane, ensuring safe and reliable navigation and minimizing the risk of collisions with other objects on the road. In addition, lane segmentation, as a fundamental input for higher-level decision-making modules, plays a crucial role in facilitating the path-planning processes of the autonomous driving system.

State-of-the-art algorithms for lane segmentation often leverage deep learning techniques, such as fully convolutional neural networks, which incorporate the encoder-decoder network structure [4], to effectively extract lane features from sensor input and infer the segmentation result as pixel-wise predictions, which have shown impressive capabilities and performance [5]. However, large-scale deep neural networks for lane segmentation are still computationally expensive and power inefficient for autonomous vehicles. Lane segmentation networks often comprise plenty of convolutional layers and up to millions of parameters [4], requiring substantial computational resources for inference, which poses significant challenges for deployment on autonomous vehicles, especially battery electric vehicles, which are more power constrained. As autonomous driving technology continues to evolve, the energy limitations in such vehicles necessitate power efficiency as a crucial factor for its large-scale applications on vehicles.

In recent years, spiking neural networks (SNNs), referred to as the third generation of neural networks, have garnered increasing attention due to their potential for high power efficiency. Compared with traditional deep neural networks that rely on continuous-valued activations, SNNs more closely mimic the spiking behavior of neurons found in natural neural networks, enabling more energy-efficient computations. The event-driven nature behind SNNs is the representation of the information through discrete and sparse spikes in the process of the forward pass. Instead of forward passing continuous values through the network, neurons of SNNs are activated by temporal spikes, resembling the communication in natural neural networks. This fundamental spike-driven principle and the inherent sparsity provide advantages with regard to computation consumption and power efficiency on neuromorphic hardware [6], reducing the overall computational overhead for deep learning.

There have been many studies that focus on solving perception problems using SNNs [7], including object detec-

[1]School of Computation, Information and Technology, Technical University of Munich.
[2]School of Computer Science and Engineering, Sun Yat-sen University.

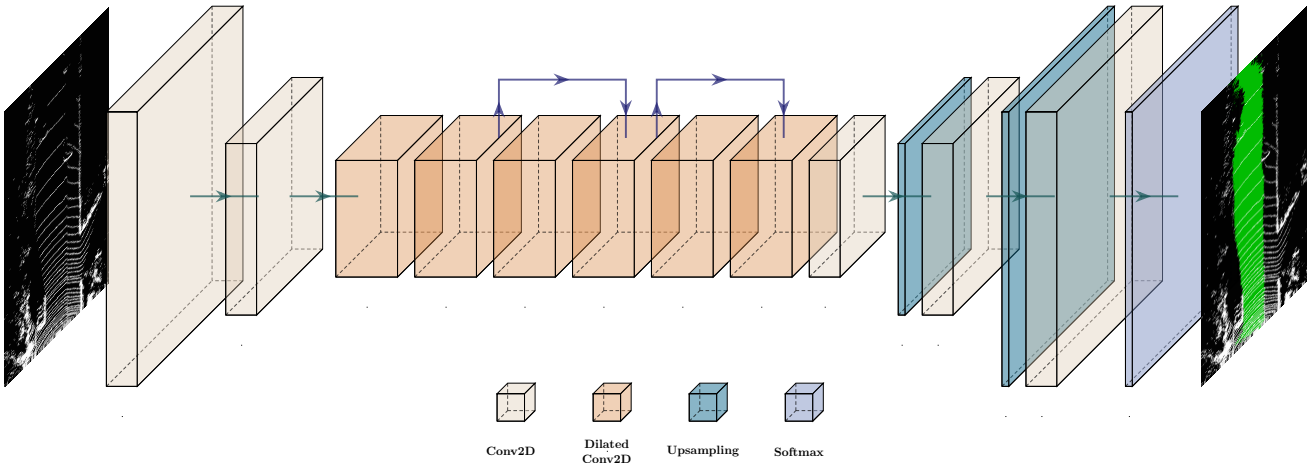


Fig. 1. Fully convolutional architecture of the lane segmentation spiking neural network, which consists of the front convolutional layers as the encoder module, dilated convolutional layers as the context module, and the up-sampling and convolutional layers as the decoder. The network processes LiDAR input and performs semantic segmentation for lane area.

tion [8] and semantic segmentation [9], based on dynamic vision sensors and RGB cameras. However, rare existing studies investigate the utilization of LiDAR sensors for lane segmentation based on SNN. In this paper, we for the first time propose a spiking neural network targeted for LiDAR sensors in order to solve lane segmentation for autonomous driving. The main process and contributions are summarized as follows:

- Spike encoding for LiDAR data and a lightweight SNN is proposed. The SNN is in an end-to-end fully convolutional network structure composed of the encoder, convolutional context module, and decoder.
- With a two-stage training approach, we incorporate an equivalent conventional neural network in the training process to facilitate the SNN training and further minimize the performance loss.
- Experiments on the KITTI dataset [10] are conducted for evaluation. The results demonstrate the high performance and efficiency of the proposed SNN for LiDAR-based lane segmentation.

II. Related Work

There is an increasing amount of research in the literature regarding SNN solutions for perception problems [7]–[9], [11]. However, there are a very limited number of SNN studies related to semantic segmentation that only emerge in recent years [9], [12], [13]. Kucik et al. in [13] proposed an SNN model to perform space-scene classification for land cover and land use based on satellite images. With regard to autonomous driving, dynamic vision sensors (DVS) and cameras are mainly used in methods detailed in [9], [12]. In TABLE I, the employed sensors and performances of different SNN methods for semantic segmentation are listed for reference. Kim et al. introduced two SNN models in [9] solving the general multi-class semantic segmentation problem based on RGB and DVS image datasets. The models are only camera-centric. In publication [12] by Viale et al., SNN models named LaneSNNs are proposed and tested to detect lane markers in the road based on DVS. However, the method is only capable of detecting lane markers. Since the drivable area can commonly be affected by obstacles and other road users, lane segmentation including drivable area classification is more significant for the advanced autonomy of autonomous vehicles. To our best knowledge, rare studies exist regarding the SNN-based solution for the lane segmentation problem especially using LiDAR sensors, which we focus on for the first time in this work.

TABLE I
Intersection over Union (IoU) Performances of Semantic Segmentation SNNs Using Different Sensors

Method		Dataset	Mean IoU (%)
RGB Camera	DeepLab [9]	PASCAL VOC2012 [14]	22.3
	FCN [9]	PASCAL VOC2012 [14]	9.9
DVS	DeepLab [9]	DDD17 [15]	33.7
	FCN [9]	DDD17 [15]	34.2
	LaneSNNs [12]	DET [16]	62.3

III. Methodology

In this work, we solve the lane segmentation problem by performing semantic segmentation on the LiDAR point cloud input, which provides a pixel-wise classification for the lane area. As shown in Fig. 1, the proposed spiking neural network (SNN) for lane segmentation has an end-to-end basis as a fully convolutional network. We implement and adapt a lightweight fully convolutional network based on the spiking rectified linear unit with a post-synaptic filter. In the training process, a two-stage training approach is employed to facilitate the training and further improve the network performance by refinement.

A. LiDAR Input Encoding

Since the proposed SNN expects spike trains as input of the network, proper data encoding is needed and crucial for high performance. In this work, we implement the rate-based

TABLE II

SPIKE-DRIVEN FULLY CONVOLUTIONAL NETWORK LAYERS

Layer	Filters	Size, Strides	Input	Output
Input	-	-	400×200×1	400×200×1
TimeDistributedConv2D	32	3×3, 2	400×200×1	200×100×32
Spiking ReLU	-	-	200×100×32	200×100×32
TimeDistributedConv2D	32	3×3, 2	200×100×32	100×50×32
Spiking ReLU	-	-	100×50×32	100×50×32
TimeDistributedDilatedConv2D	64	3×3, 1	100×50×32	100×50×64
Spiking ReLU	-	-	100×50×64	100×50×64
TimeDistributedDilatedConv2D	64	3×3, 1	100×50×64	100×50×64
Spiking ReLU	-	-	100×50×64	100×50×64
TimeDistributedDilatedConv2D	64	3×3, 1	100×50×64	100×50×64
Spiking ReLU	-	-	100×50×64	100×50×64
TimeDistributedDilatedConv2D	64	3×3, 1	100×50×64	100×50×64
Spiking ReLU	-	-	100×50×64	100×50×64
TimeDistributedDilatedConv2D	64	3×3, 1	100×50×64	100×50×64
Spiking ReLU	-	-	100×50×64	100×50×64
TimeDistributedDilatedConv2D	64	3×3, 1	100×50×64	100×50×64
Spiking ReLU	-	-	100×50×64	100×50×64
TimeDistributedConv2D	32	1×1, 1	100×50×64	100×50×32
TimeDistributedUpSampling2D	-	2×	100×50×32	200×100×32
TimeDistributedConv2D	32	3×3, 1	200×100×32	200×100×32
Spiking ReLU	-	-	200×100×32	200×100×32
TimeDistributedUpSampling2D	-	2×	200×100×32	400×200×32
TimeDistributedConv2D	2	3×3, 1	400×200×32	400×200×2
Spiking ReLU	-	-	400×200×2	400×200×2
Softmax Output	-	-	400×200×2	400×200×2

LiDAR input encoding to encode the LiDAR point cloud into spike trains. In order to first reduce the complexity of the unstructured LiDAR point cloud data, a data preprocessing is carried out for the LiDAR input. The point cloud from the LiDAR is cropped based on the dataset in advance to extract points that are located in the specified region of interest (ROI) with valid labels. In the subsequent encoding process, spike generating in a 2D grid is utilized to produce the bird-view spatial representation in the form of spike trains for the LiDAR input, with a size of 400×200. The filtered point cloud is discretized into grids of the same size. Each grid is assigned to a respective spike train which encodes the LiDAR point feature within the grid to fire spikes. In the spiking process, a temporal dimension in spike trains is introduced. The frequency of the generated spikes for each grid is proportional to the height feature of the point cloud. The encoded LiDAR input in the form of spatiotemporal spike trains is subsequently fed into the SNN for lane segmentation.

B. Fully Convolutional Network

In this paper, a lightweight fully convolutional network (FCN) [4] in the SNN fashion is proposed and implemented to perform the lane segmentation. As illustrated in TABLE II, the FCN employs two convolutional layers with a stride of two as the encoder module, six dilated convolutional layers with a dilation rate of two as the context module, and two upsampling layers with two convolutional layers as the decoder module. To further improve the performance, two residual connections are introduced in the context module for every two dilated convolutional layers, as shown in Fig. 1. In the proposed spike-driven network design especially targeted for spike input, adaptations are made in comparison to common FCNs. To deal with the 3D input spike trains, time-distributed layers are introduced in order to apply convolutional and upsampling operations to every spike train through the temporal dimension of the input. Biases in layer connections are removed to avoid constant spikes injected into neurons. In addition, convolutional layers with a stride of two is preferable to max pooling for downsampling due to its simplicity and linearity. Connection bias and max pooling are achievable for SNNs with proper additional components. However, this can entail the system with higher complexity and latency. Therefore in order to reduce system complexity and improve the spike sparsity of the network, they are not applied to the lightweight network design in this work. In the last Spiking ReLU layer, the spikes are aggregated to compute the spiking rate for each neuron in order to output the lane segmentation probabilities with the Softmax function.

C. Spiking Neuron Model

In order to implement a deep neural network with close performance and characteristics, the spiking neuron model of the rectified linear unit (ReLU) activation function is employed. Moreover, the ReLU function captures one important functional characteristic of Integrate-And-Fire neurons: given the linearity when $x > 0$, the output is equal to the proposed fire rate mechanism in the IF-neuron model. This makes the conversion from ReLU to spiking activation functions more biologically plausible. The membrane potential v_m of a spiking neuron of the ReLU unit is given as follows for a given time t:

$$v_m(t) = v_m(t-1) + \Delta t \cdot \sum_k \phi\Big(x_k(t)\Big) \cdot w_k, \tag{1}$$

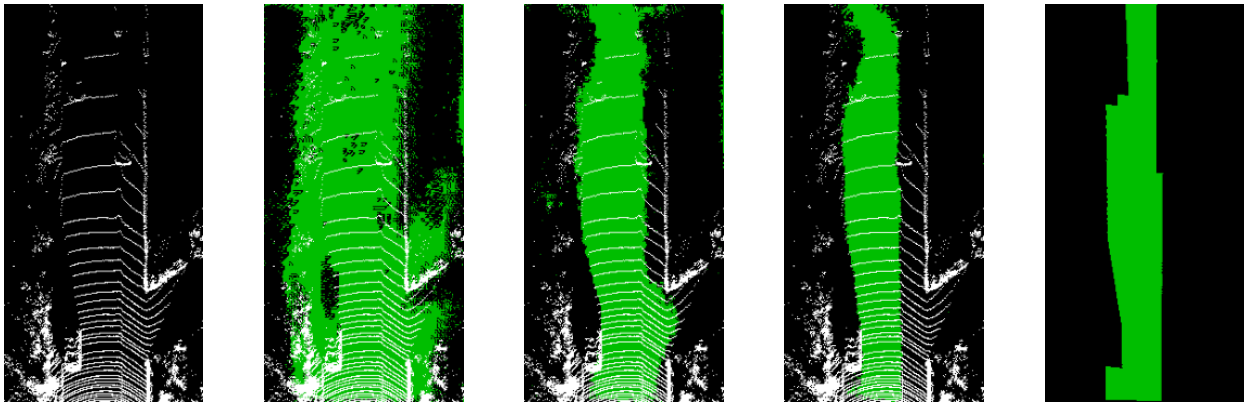
where $v_m(t-1)$ is the accumulated membrane potential of the neuron; Δt is the length of a simulation time step; $x_k(t)$ is the k-th input to the neuron via the connected synapses; $\phi(x)$ is the ReLU activation function, and w_k is the corresponding synapse weight. The membrane potential update reflects the linearity of the potential accumulation process. If the membrane potential exceeds the threshold, the spiking neuron fires a spike to its subsequently connected synapses to propagate the spikes forward.

Due to the linearity characteristic, the spiking rate $\overline{x}_t$ of the rectified linear neuron at the firing time t is proportional to the accumulated input, which is defined as follows:

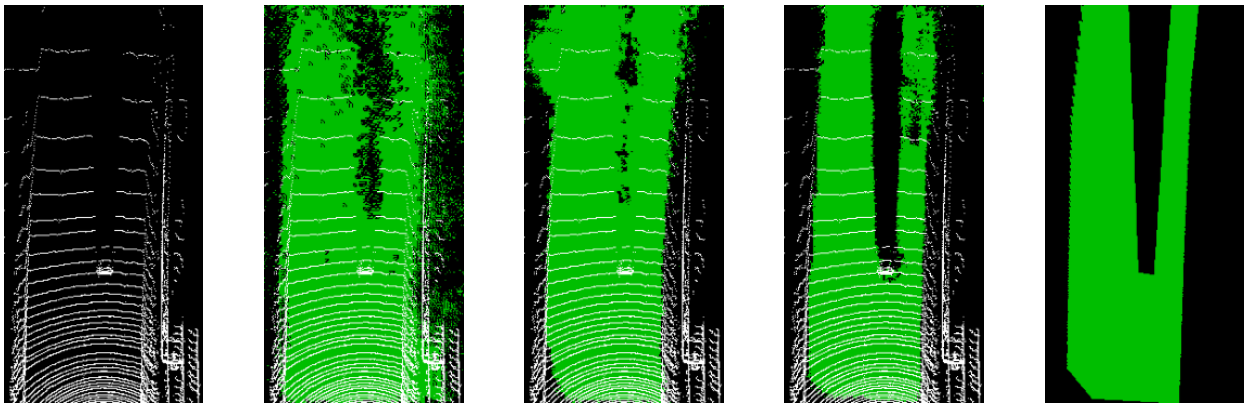
$$\overline{x}_t = \sum_t x(t) = \frac{\lfloor v_m \rfloor}{\Delta t}. \tag{2}$$

After firing, the membrane potential will be reset below the threshold:

$$v_m(t+1) = v_m(t) - x(t) \cdot \Delta t. \tag{3}$$



(a) LiDAR Input (b) $\Delta t = 0.001$ (c) $\Delta t = 0.01$ (d) $\Delta t = 0.1$ (e) Ground Truth



(f) LiDAR Input (g) $\Delta t = 0.001$ (h) $\Delta t = 0.01$ (i) $\Delta t = 0.1$ (j) Ground Truth

Fig. 2. Inference instances with different Δt. In each row, the LiDAR input data, inference outputs for different Δt, and the ground truth are listed. The closer to the ground truth, the better.

To further leverage the temporal information in the time dimension, a post-synaptic filter is introduced to aggregate and pool spikes through the output spike train. The filter provides a biologically-plausible smoothing and delay effect that corresponds to the biological post-synaptic filter, which constitutes the neuron dynamics for synaptic plasticity [17]. The post-synaptic filter is defined as a low-pass filter as follows:

$$y(t) = (1 - e^{-\Delta t/\tau}) \cdot x(t) + e^{-\Delta t/\tau} \cdot y(t-1) \quad (4)$$

in which $y(t)$ is the filter output, and τ is the trainable time constant to control the level of smoothing.

By replacing the ReLU neurons with the spiking neuron model combined with the post-synaptic filters, the fully convolutional network is converted to a spiking neural network operating based on spikes.

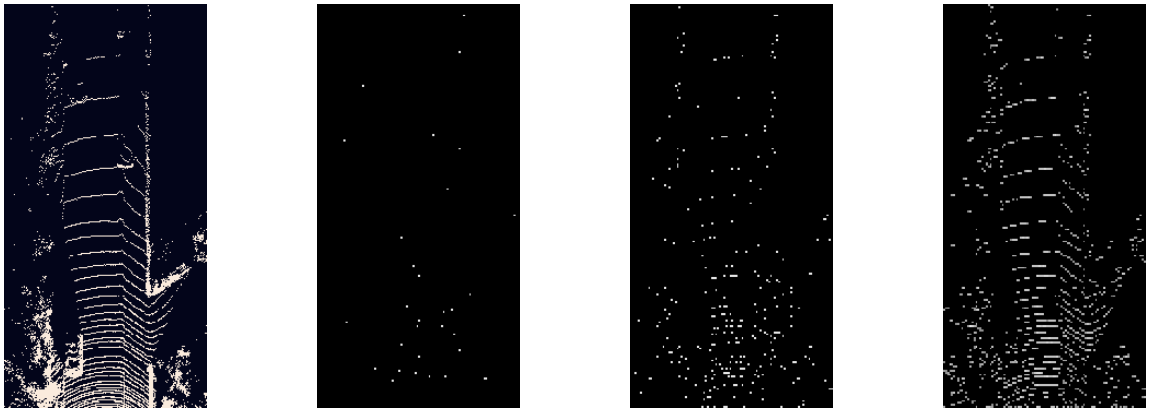
D. Training Mechanism

Since the spiking activation neural model is not differentiable due to the discrete spikes, conventional training based on backpropagation for SNNs is not directly feasible. In this work, the proposed SNN is trained in a two-stage training fashion. In the first stage, coarse training with the equivalent non-spiking DNN is performed. To facilitate the training, the equivalent DNN based on the non-spiking ReLU units is first trained with backpropagation. The binary cross-entropy loss is utilized for loss computation in the training. Due to the linear characteristic of the spiking rate of the spiking ReLU neurons, the connection weights are transferred from the equivalent trained DNN model to the SNN, as a starting point for further improvement.

In the second refinement stage, the spike-aware training [18] derived from the quantization-aware training technique [19] is exploited in this work. During the forward pass, the proposed SNN is simulated for the loss computation. In the subsequent backpropagation phase, the equivalent DNN is incorporated for gradient calculation and weight updating. The updated weights are then applied to the SNN for model refinement. With the two-stage training method, the SNN is trained efficiently and effectively.

IV. Experiments

To evaluate the effectiveness and performance of the proposed method, experiments with different settings are carried out based on a public dataset for autonomous driving scenes. In this section, we present the implementation settings and discuss the experimental results to demonstrate the applicability and performance of the lane segmentation SNN.



(a) Original LiDAR Input (b) $\Delta t = 0.001$ (c) $\Delta t = 0.01$ (d) $\Delta t = 0.1$

Fig. 3. Visualization of spiking activity in the encoder module of the SNN for different Δt values, with the original LiDAR input at the left for reference.

TABLE III
PERFORMANCES OVER TIME STEP LENGTHS

Model		Pixel Accuracy (%)	IoU (%)	Energy (J/Inf)
Non-spiking FCN		94.90	88.04	CPU: 1.13×10^{1} GPU: 3.93×10^{-1}
SNN	Δt = 0.01	88.45	76.68	7.39×10^{-4}
	Δt = 0.1	94.39	86.42	4.71×10^{-3}
	Δt = 1	94.88	87.43	4.44×10^{-2}

A. Experimental Setup

We implement and train the SNN on *TensorFlow* [20] and *KerasSpiking* [18], with the spiking ReLU units and the post-synaptic filters involved for better performance. To improve the generalizability of the proposed SNN in different autonomous scenes, the public KITTI dataset [10] is utilized for training and evaluation. The point cloud data captured with a 64-channel 3D *Velodyne* LiDAR and the semantic segmentation labels from the dataset are extracted and calibrated for training. The extracted samples are split into a training (80%), validation (10%), and testing (10%) set. The model is trained with an Adam optimizer and the learning rate is set to 0.0005. A batch size of 4 is used for training iterations. Besides, in order to investigate the influence of the time step length Δt hyperparameter, different values of Δt: 0.001, 0.01, 0.1, and 1 second(s) are examined in the experiments for training and testing.

To evaluate and analyze the performance of the SNN, two main evaluation metrics, pixel accuracy and Intersection over Union (IoU) are used in the result analysis, which are the most common metrics for lane segmentation and general semantic segmentation tasks. In addition, we evaluate the energy consumption of the SNN based on *TensorFlow* and *KerasSpiking* on different devices including CPU, GPU, and dedicated neuromorphic hardware.

B. Experimental Results

We first present the qualitative results for two testing samples in Fig. 2, and explore the influence of different time step lengths. The tested samples listed in two rows show the inference results for different time step lengths Δt, with the original LiDAR and the ground truth for reference. As shown in the figure, in the case of $\Delta t = 0.001$, the SNN is not capable of effectively segmenting the drivable lane area, generating false-positive areas for each input sample. As Δt increases to 0.01 and 0.1, the SNN yields better lane segmentation results with the ground truths as reference. In the case of $\Delta t = 0.01$, the trained model is able to segment the close-range lane area with some irregular false-positive areas at different distances. It is also capable of detecting the vehicle in close range which blocks the drivable area behind, as shown in Fig. 2c. However, as shown in Fig. 2h, the model fails to detect the vehicle in mid-range. The inference results with $\Delta t = 0.1$ are better compared to the cases of $\Delta t = 0.001$ and $\Delta t = 0.01$, with fewer false predictions. As shown in Fig. 2d and Fig. 2i, blocking vehicles in close and mid-range are successfully detected. The qualitative results of $\Delta t = 1$ are close to the case of $\Delta t = 0.1$, therefore they are not listed in Fig. 2.

To investigate the significance of the time step length Δt, further quantitative studies for different values of Δt are conducted. In Fig. 3 we visualize the output spikes of the first ReLU activation layer to inspect the underlying spiking activity in the encoder module of the SNN during the second stage of training. Given a spiking rate, the number of generated spikes is proportional to the period of time. Therefore, an inappropriate Δt value can cause the SNN model to generate either over-sparse spikes which are not abundant enough to perform inference, or over-dense spikes which lead to additional energy consumption. As shown in Fig. 3b, in the case of $\Delta t = 0.001$, rare spikes are generated to represent the LiDAR input, which corresponds to the poor inference performance shown in the second column of Fig. 2. In the case of $\Delta t = 0.01$, the spike activity shown in Fig. 3c is sparse and sub-optimal to fully reconstruct the LiDAR input. As Δt increases to 0.1, the spikes are dense enough to produce a close representation for the LiDAR point cloud input and yield a good performance of inference as shown in Fig. 2.

In TABLE III, we report the quantitative results of the

proposed SNN's performance. The mean results of pixel accuracy, intersection over union (IoU), and the estimated energy consumption per inference on *Intel Loihi* are listed for the SNN models with different Δt values, respectively. The results of the equivalent non-spiking FCN with its energy consumption on CPU and GPU are also estimated and listed in the table as the baseline [21]. As shown in the table, the increase of Δt corresponds to a tendency of increasing performances as well as energy consumption. In the case of $\Delta t = 1$, the SNN has subtle improvements in the pixel accuracy and the IoU result. However, the time length $\Delta t = 1$ can introduce a long undesired latency to the perception system for real-time application. In comparison, the results for $\Delta t = 0.1$ exhibit that the SNN has a performance close to the equivalent non-spiking CNN, showing a very small performance loss at only 0.51% for pixel accuracy and 1.62% for IoU. At the same time, the SNN model with $\Delta t = 0.1$ embodies a large advantage in energy consumption compared to the non-spiking FCN, consuming only 0.042% and 1.2% of energy on CPU and GPU per inference, respectively.

Taking the reported performances of SNN methods based on different sensors that we illustrate in TABLE I into consideration, we can see the comparison of the proposed method with the other methods. Since there are rare state-of-art SNN models solving the lane area segmentation problem, only methods reported in [9], [12] in the segmentation purpose targeted for driving scenes are considered in comparison with our method, in which dynamic vision sensors (DVS) are used. Models proposed in [9] perform semantic segmentation on the DDD17 dataset [15] achieve 33.7% and 34.2% IoU. The LaneSNNs model introduced in [12] which only detects lane markers in the road achieves higher IoU performance at 62.3%, tested on the DET dataset [16]. Based on LiDAR, the proposed SNN in this work shows 24.1%-52.7% improvements with regard to IoU, which showcases the advantage in segmentation effectiveness of the LiDAR-based SNN.

The experimental results demonstrate the high applicability and high performance of the proposed SNN on lane segmentation based on LiDAR. The results also exhibit tremendous potential regarding power efficiency for the lane segmentation application on power-constrained vehicles.

V. Conclusion

This paper proposes a spiking neural network (SNN) designed specifically for LiDAR sensors, with the objective of addressing lane segmentation in autonomous driving. The proposed SNN is constructed in an end-to-end fully convolutional structure, incorporating spike trains input generated from the LiDAR point cloud. To facilitate the training process and minimize performance loss, a two-stage training approach is employed with an equivalent conventional artificial neural network. Experiments are conducted on the KITTI dataset to evaluate the effectiveness and performance of the proposed method. The results and power consumption evaluation demonstrate the high performance and efficiency of the proposed SNN for LiDAR-based lane segmentation.

References

[1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[2] Y. Wu, Y. Wang, S. Zhang, and H. Ogai, "Deep 3d object detection networks using lidar data: A review," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1152–1171, 2020.

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

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[5] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," in *2017 ieee intelligent vehicles symposium (iv)*.

[6] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *Ieee Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[7] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: opportunities and challenges," *Frontiers in neuroscience*, p. 774, 2018.

[8] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-yolo: spiking neural network for energy-efficient object detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 270–11 277.

[9] Y. Kim, J. Chough, and P. Panda, "Beyond classification: Directly training spiking neural networks for semantic segmentation," *Neuromorphic Computing and Engineering*, vol. 2, no. 4, p. 044015, 2022.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[11] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, pp. 54–66, 2015.

[12] A. Viale, A. Marchisio, M. Martina, G. Masera, and M. Shafique, "Lanesnns: Spiking neural networks for lane detection on the loihi neuromorphic processor," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 79–86.

[13] A. S. Kucik and G. Meoni, "Investigating spiking neural networks for energy-efficient on-board ai applications. a case study in land cover and land use classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2020–2030.

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[15] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "Ddd17: End-to-end davis driving dataset," *arXiv preprint arXiv:1711.01458*, 2017.

[16] W. Cheng, H. Luo, W. Yang, L. Yu, S. Chen, and W. Li, "Det: A high-resolution dvs dataset for lane extraction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[17] R. Moreno-Bote and N. Parga, "Role of synaptic filtering on the firing response of simple model neurons," *Physical review letters*, vol. 92, no. 2, p. 028102, 2004.

[18] T. Bekolay, J. Bergstra, E. Hunsberger, T. DeWolf, T. Stewart, D. Rasmussen, X. Choo, A. Voelker, and C. Eliasmith, "Nengo: a Python tool for building large-scale functional brain models," *Frontiers in Neuroinformatics*, vol. 7, no. 48, pp. 1–13, 2014.

[19] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *arXiv preprint arXiv:2109.12894*, 2021.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *Osdi*, vol. 16, no. 2016. Savannah, GA, USA, 2016, pp. 265–283.

[21] B. Degnan, B. Marr, and J. Hasler, "Assessing trends in performance per watt for signal processing applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 1, pp. 58–66, 2015.