mathematics

MDPI

Article

Benchmarking Perception to Streaming Inputs in Vision-Centric Autonomous Driving

Tianshi Jin, Weiping Ding *, Mingliang Yang, Honglin Zhu and Peisong Dai

School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China; ts.jin@my.swjtu.edu.cn (T.J.)
* Correspondence: dwp@swjtu.edu.cn

Abstract: In recent years, vision-centric perception has played a crucial role in autonomous driving tasks, encompassing functions such as 3D detection, map construction, and motion forecasting. However, the deployment of vision-centric approaches in practical scenarios is hindered by substantial latency, often deviating significantly from the outcomes achieved through offline training. This disparity arises from the fact that conventional benchmarks for autonomous driving perception predominantly conduct offline evaluations, thereby largely overlooking the latency concerns prevalent in real-world deployment. Although a few benchmarks have been proposed to address this limitation by introducing effective evaluation methods for online perception, they do not adequately consider the intricacies introduced by the complexity of input information streams. To address this gap, we propose the Autonomous driving Streaming I/O (ASIO) benchmark, aiming to assess the streaming input characteristics and online performance of vision-centric perception in autonomous driving. To facilitate this evaluation across diverse streaming inputs, we initially establish a dataset based on the CARLA Leaderboard. In alignment with real-world deployment considerations, we further develop evaluation metrics based on information complexity specifically tailored for streaming inputs and streaming performance. Experimental results indicate significant variations in model performance and ranking under different major camera deployments, underscoring the necessity of thoroughly accounting for the influences of model latency and streaming input characteristics during real-world deployment. To enhance streaming performance consistently across distinct streaming input features, we introduce a backbone switcher based on the identified streaming input characteristics. Experimental validation demonstrates its efficacy in perpetually improving streaming performance across varying streaming input features.

Keywords: vison-centric perception benchmark; online assessment; streaming inputs; two-dimensional entropy

MSC: 68T99

Citation: Jin, T.; Ding, W.; Yang, M.; Zhu, H.; Dai, P. Benchmarking Perception to Streaming Inputs in Vision-Centric Autonomous Driving. *Mathematics* **2023**, *11*, 4976. https://doi.org/10.3390/math11244976

Academic Editor: António Lopes

Received: 16 November 2023
Revised: 14 December 2023
Accepted: 15 December 2023
Published: 16 December 2023

1. Introduction

Vision-centric perception has attracted considerable attention in the field of autonomous driving in recent years. It is intuitive that vision plays the most dominant role in human driving. In principle, vision-centric perception can obtain the richest semantic information, which is essential for decision-making in autonomous driving, compared to LiDAR-based and millimeter-wave radar-based perception. Moreover, we found a large body of previous research on vision-based perception for various autonomous driving tasks in past years, such as 3D detection [1–10] in driving scenes, map construction [11–15], motion prediction [16,17], and even end-to-end autonomous driving [18–20].

Despite the remarkable achievements in vision-centric perception research, many methods suffer from high latency when deployed in real-world settings, which hinders their online performance. For example, in 3D detection, a fundamental task for autonomous

driving, camera-based 3D detectors usually have much longer (See Table 1) inference time than LiDAR-based counterparts [21–23] (on NVIDIA RTX4090). Therefore, it is essential to have evaluation metrics that balance accuracy and latency. However, most of the existing benchmarks [24–34] focus on evaluating offline performance only (e.g., Average Precision (AP), Intersection over Union (IoU), etc.). Although some studies have adopted the streaming perception paradigm to measure accuracy–latency trade-offs and Wang et al. [35] proposed an online evaluation protocol that can assess the online performance of different perception methods under various hardware conditions, they still lack prior evaluation of the streaming input. This means that for the online performance evaluation of the vision-centric perception, they are still missing the initial impact of the streaming inputs.

This paper introduces the Autonomous driving Streaming I/O (ASIO) benchmark to address the problems mentioned above. In light of the existing contributions to autonomous driving perception algorithms and hardware impacts, ASIO has directed its focus towards benchmark evaluations of the GPU perception data path. This effort involves quantifying the influence of various input sources on streaming perception performance, addressing a gap in current research. Current research only focuses on evaluating online performance without considering the influence of various streaming inputs on the perception system (e.g., different resolutions, field-of-view angles, etc.). Unlike mainstream datasets and benchmarks, our benchmark is based on the CARLA Leaderboard [36] simulator, ensuring consistency in the environment and targets encountered during testing. We gather real-time streaming input data and employ automated tools for annotating targets, a process verified through manual sampling and comparison with the nuScenes dataset. This dataset is then utilized to assess the online performance of 3D detection. Practical deployment is also investigated, specifically the problem of ASIO under different inputs. We design evaluation metrics for perception streaming inputs based on the fractional dimensional entropy computation method of time series to assess streaming perception with different models. Figure 1 illustrates the significant impact of perception inputs' variation on the streaming performance of different methods. Our approach provides a more precise characterization of the effect of perception input on the deployment of real-world autonomous driving tasks compared to classical offline benchmarks. The main contributions of this paper are summarized as follows:

(1) We present the ASIO benchmark for quantitatively evaluating the characteristics of camera-based streaming inputs and the streaming perception performance, which opens up possibilities for vision-centric perception design and performance prediction for autonomous driving.
(2) A scenario and a dataset for evaluating different streaming inputs are built based on the CARLA Leaderboard, which enables camera-based 3D detection streaming evaluation.
(3) For the implicit characteristics in streaming inputs, the computation of fractional order entropy values in one and two dimensions is proposed to construct quantitative metrics, where we investigate the streaming performance of seven modern camera-based 3D detectors under various streaming inputs.

The remainder of this paper is organized as follows: In Section 2, we analyze current offline and online evaluation methods for autonomous driving perception and identify research gaps within them. Additionally, we delve into methods for characterizing information complexity. Subsequently, in Section 3, we introduce the dataset established for assessing streaming inputs, along with the establishment of metrics for online evaluation. Meanwhile, we propose an improved method for 3D detector enhancement based on the aforementioned approaches. In Section 4, we conduct a streaming performance evaluation of seven typical 3D detectors across various cameras, extracting features of streaming perception and validating the a priori nature of our metrics. This section identifies factors that should be considered in the practical deployment of perception systems. Section 5 concludes the entire work, highlighting existing issues and providing suggestions for future improvements.

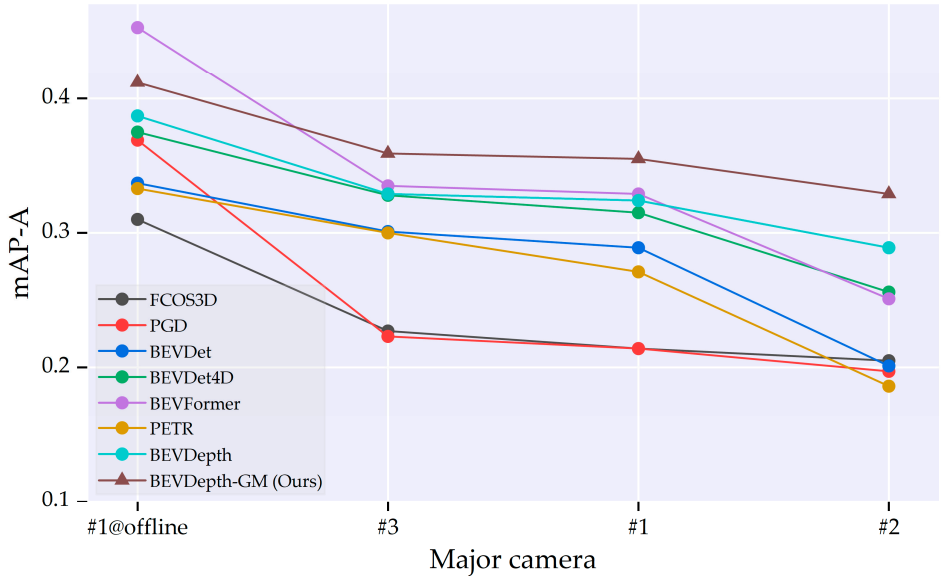

Figure 1. Comparison of streaming performances on our benchmark, where the model rank changes based on the variation in streaming inputs. The BEVDepth-GM detector (built on BEVDepth [4]) equipped with our switcher achieved better streaming performance on different major cameras. Major camera designations #1, #2, and #3 serve as the primary subjects for our online testing, with detailed specifications provided in Section 4.2. Specifically, the notation "@offline" indicates the utilization of offline evaluation.

Table 1. Comparison between autonomous driving perception datasets.

Dataset	Stream.	Modality	Task	Model Speed
KITTI [26]	×	LiDAR & Camera	Multi-task	-
Argoverse [32]	×	LiDAR & Camera	Multi-task	-
nuScenes [24]	×	LiDAR & Camera	Multi-task	-
Waymo [31]	×	LiDAR & Camera	Multi-task	-
CARLA Leaderboard	×	LiDAR & Camera	Multi-task	-
Argoverse-HD [37]	√	Camera	2D det.	~40 FPS
nuScenes-H [35]	√	Camera	3D det.	~7 FPS
Waymo	√	LiDAR	3D det.	~25 FPS
CARLA Leaderboard	√	Camera	2D & 3D det.	~30 FPS@2D det. ~5 FPS@3D det.

2. Related Work

2.1. Autonomous-Driving Benchmark

Thanks to the various open-source benchmarks, the past decade has witnessed significant progress in autonomous driving perception. These benchmarks have shifted from 2D detection [25,28,29,33,34] tasks to 3D detection [24,26,27,30–32] tasks, which are tailored for autonomous driving scene understanding. Additionally, the data acquisition has also progressed from single RGB images to multi-modal data. Even popular datasets with 3D annotations employ surround-view images, greatly facilitating the development of vision-centric perception. However, these benchmarks primarily focus on assessing the offline performance of perception algorithms, overlooking the practical issues of perception system deployment.

2.2. Streaming Perception

The deployment of perception in autonomous driving faces the challenge of balancing accuracy and latency. In order to improve perception performance, previous works have explored the concept of streaming perception, which utilizes temporal information. For

example, Li et al. [37] introduced a benchmark for image detection algorithms and proposed a method based on Kalman filtering [38] and reinforcement learning [39] to mitigate latency. Han et al. [40] developed an efficient streaming detector for LiDAR-based 3D detection tasks, accurately predicting future frames. Wang et al. [35] presented a benchmark for various perception models under different computational constraints, with a focus on the 3D detection task. These works establish evaluation paradigms for camera-based 3D detection, as well as LiDAR-based 3D detection, highlighting the trade-off between accuracy and latency in real-world deployment. However, it is also worth noting the significant impact of streaming input on overall performance [35,37]. Therefore, there is a need for a methodology that incorporates input sources into streaming perception evaluation for autonomous driving.

2.3. Nonlinear Time Series Complexity

Autonomous driving perception systems are similar to and based on real-world, e.g., ecological, meteorological, geological, etc., systems generated by natural or physical mechanisms and are complex systems whose modes of operation are difficult to explain deterministically or by constructing analytical models [41,42]. When dealing with streaming input, it is important to quantify its complexity. The concept of Shannon's information entropy has been continuously promoted and extended, leading to the proposal of various discrete forms of entropy metrics such as Rényi entropy [43], Tsallis entropy [44], approximate entropy [45], sample entropy [46], and permutation entropy [47]. These metrics have become the main tools for measuring the complexity of a system. Ubriaco [48] introduced a new entropy measure known as fractional entropy. This measure promotes the integer order Shannon entropy in fractional dimensions, providing the possibility of entropy metrics in the application of systems with long-range correlation. Fractional entropy not only has a high sensitivity to the dynamic changes in the signal features but also reveals more details and information about the system. As a result, it demonstrates better utility in practice [49]. To explore the implicit information, we incorporated this analysis into the ASIO benchmark. We considered one- and two-dimensional aspects in the processing of the streaming input, aiming to reveal the underlying information.

3. Methods

This section begins with an introduction to the concept of ASIO. Then, we provide a test scenario and corresponding dataset to evaluate the holistic streaming perception. Finally, we present evaluation metrics to measure the streaming perception performance across various input conditions.

3.1. Autonomous Driving Streaming I/O

The evaluation of the ASIO benchmark has two aspects: evaluating the information complexity of the streaming inputs and evaluating the streaming perception performance online.

Obviously, autonomous driving perception is a multiple-input, multiple-output system at each level, and usually, the dimensions of the inputs are larger than the dimensions of the outputs. Each level can be represented by the model in Figure 2. The input vector X consists of components $X_1, X_2, \cdots, X_m$, and the output vector Y consists of components $Y_1, Y_2, \cdots, Y_l$, so the requirement of the subsystem is to maximize the information transmitted to the output about the inputs of the system, according to the principle of maximum mutual information. Based on the information-theoretic model described above, it is necessary to evaluate the complexity of the streaming input X. Specially, given streaming inputs $\{X_m\}_{m=1}^{T}$, where X_m is the image inputs at timestamp t_m and T is the total number of input timestamps. The perception algorithms are acquired to make an online response to the input instance, and the entire online predictions are $\{\hat{Y}_l\}_{l=1}^{U}$, where $\hat{Y}$ is the prediction at the timestamp t_l, and U represents the total number of predictions. Notably, the prediction timestamps are not synchronized with the input timestamps, and the model inference speed

is typically slower than the input frame rate (i.e., $U < T$). To evaluate the predictions at the input timestamp t_m, the ground truth Y_m should match the most recent prediction, yielding the pair $\left(Y_m, \hat{Y}_{\theta(m)}\right)$, where $\theta(m) = \underset{l}{argmax} t_l < t_m$. Based on the matching strategy, the ASIO benchmark evaluates the complexity of the streaming inputs:

$$\mathcal{H}_{\text{ASIO}} = \mathcal{H}(X_m), \tag{1}$$

and the online performance at every input timestamp:

$$\mathcal{O}_{\text{ASIO}} = \mathcal{O}\left(\left\{\left(Y_m, \hat{Y}_{\theta(m)}\right)\right\}_{m=1}^{T}\right), \tag{2}$$

where $\mathcal{H}(\cdot)$ and $\mathcal{O}(\cdot)$ are the evaluation metrics, which will be elaborated on in subsequent sections. Notably, ASIO instantiates the streaming paradigm on camera-based 3D detection, and the key insights also generalize to other vision-centric perceptions in autonomous driving.

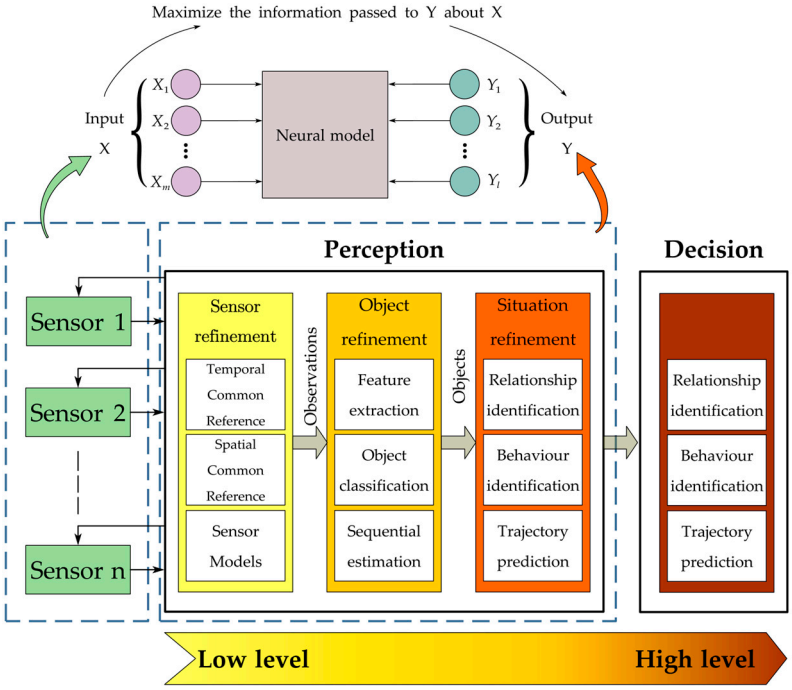

Figure 2. The perception system process encompasses various levels of data, information processing, and modeling in an overall scheme. We view the sensor's inputs and perceived performance in the model as the I/O of the streaming.

3.2. Scenarios and Dataset

To validate all the proposed methods, we established a standardized visual perception benchmark using CARLA. The evaluation scenario map of Town 12 from CARLA Leaderboard 2 was chosen for this purpose (Figure 3). This city consists of various contrasting regions, including urban, residential, and rural areas, with a surrounding highway system and a ring road. The architectural styles in the city resemble those commonly seen in medium to large cities worldwide. In our study, we focused primarily on urban scenes and selected a fixed number of road participants, such as vehicles of different types and colors. The proportions of these vehicles were determined based on a relevant survey. Additionally, the traffic participants and the ratios of cars to trucks and vehicles to pedestrians were designed accordingly. The scenario's roads encompassed various types of lanes, road markings, traffic lights, signage, and even different weather conditions (Table 2). In this scenario, the vehicle travels along a fixed route at a fixed speed for a total distance of 5 km. It is equipped with a visual perception sensor that needs evaluation. The vehicle traverses

alongside the set-up traffic participants, which are equipped with their respective motion states, during the entire journey.

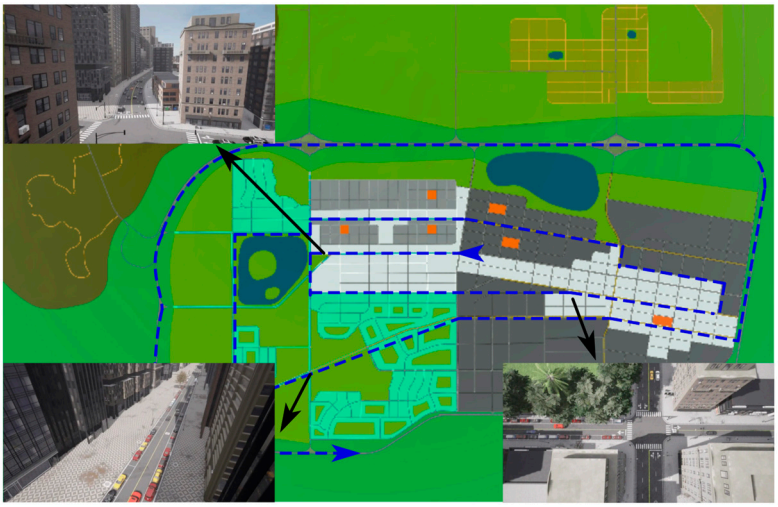

Figure 3. The simulation scenarios. The blue dashed lines and arrows represent the testing routes, while the black arrows highlight specific details within the scenarios.

Table 2. Traffic participants and weather.

Types	**Quantities/Discerptions**	
Vehicles	200	
Pedestrians	30	
Traffic lights	Default in map	
Sidewalks	Default in map	
Weather	Noon	Clear
		Mid-rainy
	Night	Clear
		Mid-rainy

Meanwhile, we set up an RGB baseline camera with a sampling frequency of 10 Hz and a resolution of 1600 × 900, as well as a 32-beam spinning LiDAR with a sampling frequency of 10 Hz, a 360° horizontal FOV, and a −30° to 10° vertical FOV for annotation purposes. For each weather condition, we annotated the data at 10 Hz for a cycle of 1000 s, resulting in 10,000 images and point cloud frames to be annotated. We conducted tests under four weather conditions while keeping the route and the targets constant so the annotation work did not need to be repeated. For 3D detection, we adopted the SUSTechPoints [50] annotation tool. For each streaming input to be tested, we combined the annotated point cloud information and used the CAT [51] model to complete the 3D box annotation and motion state assignment on the test images. In contrast to the existing streaming perception datasets (see Table 3), we built a simulator-based dataset by manually collecting and annotating data, enabling the evaluation of streaming inputs. We tested some perception models on our constructed dataset and compared them with the mainstream dataset nuScenes (see Table 4). In light of constraints pertaining to the dataset capacity and the limited variety and quantity of the target objects we incorporated, the outcomes revealed an overestimation of scores within our dataset. Nevertheless, the overall performance exhibited trends analogous to those observed in nuScenes, thereby providing partial validation of the validity of our dataset.

Table 3. Streaming perception dataset comparison.

Dataset	Construction Methods	Task	Evaluation Containing [1]		
			Inputs	Computation	Online
Argoverse-HD	1. Based on Argoverse 2. Extended with manually added annotations	2D Detection	×	×	√
nuScenes-H	1. Based on nuScenes 2. Expanding the annotations therein from 3 Hz to 12 Hz	3D Detection	×	√	√
Ours	1. Based on CARLA 2. Manually annotation of baseline 3. Automatic annotation of test object	3D Detection	√	×	√

[1] The symbol √ signifies that the benchmark includes the respective item, while the symbol × denotes its exclusion.

Table 4. Popular algorithm validation on our dataset.

Methods	nuScenes		Ours	
	mAP	NDS	mAP	NDS
FCOS3D [52]	0.358	0.428	0.459	0.538
DETR3D [53]	0.412	0.479	0.520	0.580
BEVFormer [5]	0.481	0.569	0.581	0.688
BEVDepth	0.520	0.609	0.629	0.719
SOLOFusion [10]	0.540	0.619	0.647	0.721

3.3. Evaluation Metrics

We developed evaluation metrics for streaming inputs and streaming performance, aiming to comprehensively examine the holistic streaming perception of various 3D detectors under different inputs. This section first introduces the streaming input metrics and then explains the streaming performance metrics.

3.3.1. Streaming Input Metrics

As shown in the information-theoretic model in Figure 2, we need to reveal the information complexity of the streaming input sequence. One common way to evaluate the pixel inputs is to calculate their information entropy. To describe the local structural features of the streaming inputs, we introduce two-dimensional entropy, which reveals the combined features of pixel greyscale information and greyscale distribution in the vicinity of the pixel. The feature pair (x_1, x_2) is formed by the gray level of the current pixel and the mean value of its neighborhood. Here, x_1 represents the gray level of the pixel and x_2 represents the mean value of the neighbors. The combined probability density distribution function of x_1 and x_2 is then given by the following equation:

$$p(x_1, x_2) = \frac{f(x_1, x_2)}{P \times Q} \tag{3}$$

where $f(x_1, x_2)$ is the frequency at which the feature pair (x_1, x_2) appears, and the size of X is $P \times Q$. In our implementation, x_1 is derived from the eight adjacent neighbors of the center pixel, as depicted in Figure 4. The discrete fractional two-dimensional entropy is defined as follows:

$$H = -\sum_{x_1=0}^{255} \sum_{x_2=0}^{255} p(x_1,x_2) \log_2 p(x_1,x_2) \tag{4}$$

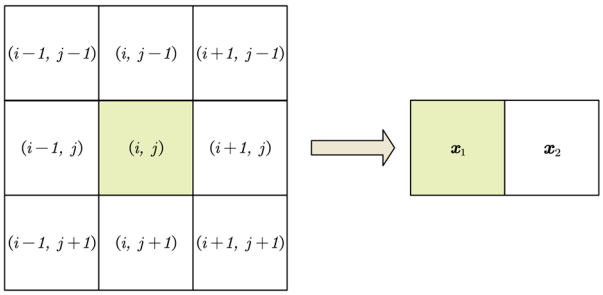

Figure 4. The pair (x_1,x_2)—a pixel and its eight neighborhoods.

To achieve a long-range correlation of information metrics in perception systems, we use a fractional–dimensional extension of information entropy and generalize it. First, based on the theory of fractional calculus, the integer-order Shannon entropy is extended to fractional dimension, leading to the proposition of a new entropy measure called fractional entropy, as:

$$S_q(P) = \sum_i^n p_i(-\log_2 p_i)^q \tag{5}$$

Proposed as an alternative measure to Shannon's information entropy, Cumulative Residual Entropy (CRE) [54] was introduced due to the difficulty in estimating the differential entropy of a continuous random variable through empirical distribution in practice. For a nonnegative discrete random variable X, its CRE is defined as:

$$\mathcal{E}(X) = -\sum \overline{F}(x)\log_2 \overline{F}(x) \tag{6}$$

where $F(x)$ is the cumulative distribution function of X and $\overline{F}(x) = 1 - F(x)$, which can be estimated through the empirical entropy value of the sample.

The single frames H computed above are represented as fractional dimensional CREs to obtain our proposed metrics for evaluating the information complexity of streaming inputs:

$$\mathcal{E}^q(X) = \sum \overline{F}(x)\left[-\log_2 \overline{F}(x)\right]^q, q \in [0,1] \tag{7}$$

$$\mathcal{H} = \mathcal{E}^q(H) \tag{8}$$

The computed $\mathcal{H}$ exhibits notable numerical disparities when dealing with sequential inputs containing pixels of varying fields of view (FOVs). In order to more accurately assess streaming inputs, we also incorporate the density of $\mathcal{H}$ in the FOV space, which we denote as $\mathcal{D}$:

$$\mathcal{D} = \frac{\mathcal{H}}{\log_2(H \times V)}, \tag{9}$$

where $H \times V$ represents the horizontal and vertical field-of-view angles, respectively.

Figure 5 shows the information complexity calculation of the streaming inputs from the baseline camera during a complete run of our benchmark. The run used a sliding window with overlap, with the window length set to 400 and the sliding distance set to 200, and calculated its $\mathcal{H}$ values in the fractional dimension order $q \in [0,1]$ of 0.02 steps. When key targets or scenarios occur, the values of $\mathcal{H}$ at $q < 0.5$ show dramatic fluctuations with time scales. When q is close to 1, the values are smaller and vary slightly with time. Therefore, compared to $\mathcal{H}$, the classical approach (when $q = 1$) cannot effectively reveal the fluctuation of information from the streaming inputs about key targets and scenes. From this perspective, $\mathcal{H}$ is superior to classical information entropy methods. By introducing fractional order parameters q, $\mathcal{H}$ is able to capture the detailed variations of the system information, thus more accurately detecting the changes in the system's dynamic features and providing effective cues for evaluating the performance of visual perception. Numerous studies have shown that the introduction of the fractional dimension makes the

entropy metric more applicable to the study of time series compared to Shannon entropy, which is also clearly demonstrated by the above results.

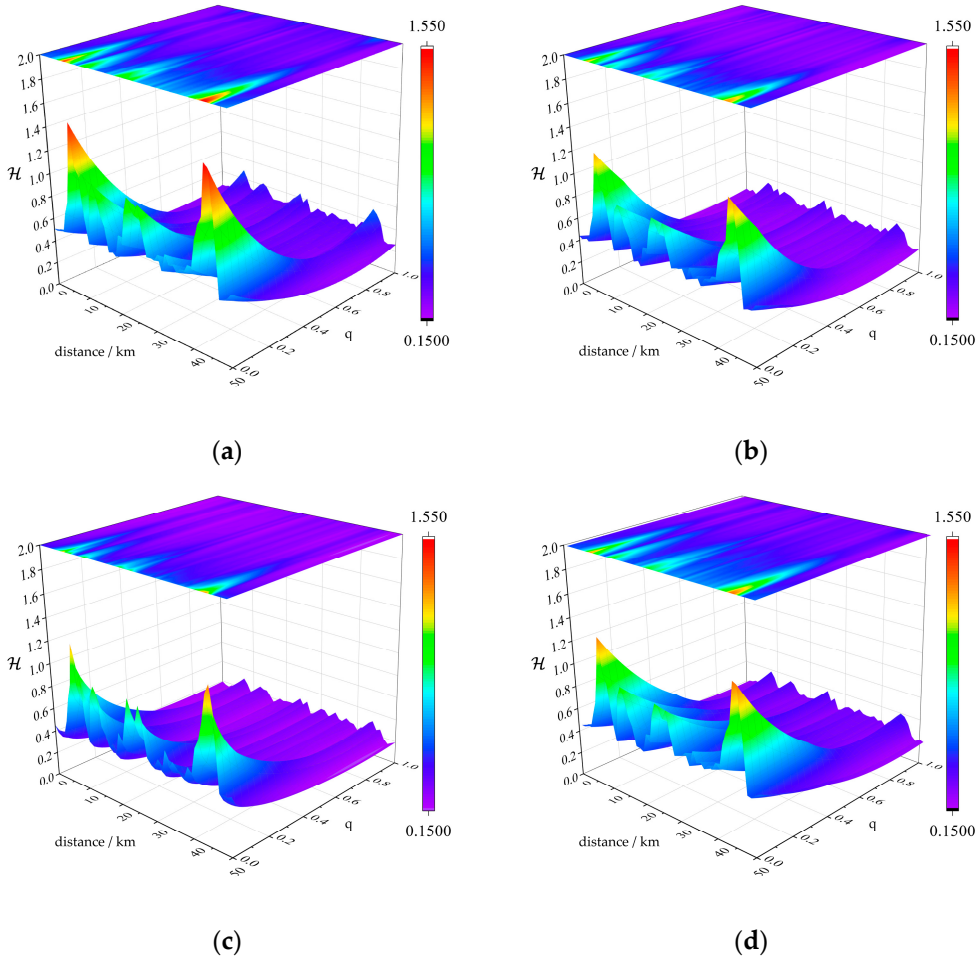

Figure 5. (**a**–**d**) depict the contour plots of the $\mathcal{H}$ distributions in w1, w2, w3, and w4, where w1 corresponds to clear noon, w2 to rainy noon, w3 to clear night, and w4 to rainy night. It is evident that, relative to the case where q is equal to 1 (Shannon's entropy description), the fractional dimensional order with q taking values within the range of (0, 1) provides a significantly more sensitive elucidation of the critical nodes in the benchmark scenario.

To obtain the q-value that is suitable for our benchmark, we expect the $\mathcal{H}$ test to show minimal variation under different weather conditions and to exhibit a larger gradient when critical scenarios occur. Therefore, the following two operators are proposed:

$$\sigma = \sqrt{\frac{\sum(\mathcal{H} - \overline{\mathcal{H}})^2}{n}}, \tag{10}$$

$$\nabla h = \sum \frac{h(s-d) - h(s+d)}{2d}, \tag{11}$$

where σ represents the standard deviation of $\mathcal{H}$ in w1–w4 under the sliding total number n, $\mathcal{H}$ is denoted by $h(s)$, and the value of d is the sliding window step size. By constructing the following optimization equation:

$$\begin{cases} \min f(\alpha) = [\sigma, -\nabla h] \\ s.t. \qquad\quad q \in [0,1] \end{cases}, \tag{12}$$

we obtain q = 0.36, which is applicable to our benchmark under the current conditions, and this result is also used by default in the experimental section later on.

3.3.2. Streaming Performance Metrics

We adopted the official nuScenes evaluation metrics, comprising Average Translation Error (ATE), Average Scale Error (ASE), Average Orientation Error (AOE), Average Velocity Error (AVE), Average Attribute Error (AAE), nuScenes Detection Score (NDS) and mean Average Precision (mAP). Due to the latency induced by the inference time in the streaming performance evaluation, there is a discrepancy between the predicted bounding boxes and the ground-truth locations, which is more pronounced for high-speed targets. Nevertheless, the AVE metric only quantifies the velocity error of true positive objects, which tend to be low-speed or static objects. Hence, we preserved the offline property of AVE while applying streaming evaluation to other metrics, denoted as mAP-A, ATE-A, ASE-A, AOE-A, and AAE-A. Inspired by existing methods [35,37], we introduce the ASIO Detection Score (ADS) as follows:

$$\mathrm{ADS} = \frac{1}{10}[5\mathrm{mAP\text{-}A} + \sum_{\mathrm{mTP}\in\mathbb{TP}}(1 - \min(1, \mathrm{mTP}))], \tag{13}$$

where $\mathbb{TP}$ = {AVE, ATE-A, ASE-A, AOE-A, AAE-A} is the set of true positive metrics.

3.3.3. ASIO Switcher

The ASIO benchmark evaluates the current detector inference at the instant time regardless of its completion status. This way, the online inference latency significantly affects the streaming perception performance. According to the conclusion of existing methods [35,37], the online performance is largely influenced by the streaming inputs and the backbone used by the detector, and it is very different from the offline evaluation result. It can be inferred that choosing the corresponding backbone for the 3D detector according to the complexity of different inputs can improve the online performance of the detector. Therefore, in this section, we designed a switcher to different backbones for the detector by evaluating the information complexity of the current and previous streaming inputs.

For real-time sequential inputs, we selected the information complexity measure $\mathcal{H}_i$–$\mathcal{H}_j$ within a time window and observed that when the $\mathcal{H}$ of streaming inputs fluctuates significantly, its local distribution approximates an exponential distribution. Therefore, we chose the grey model GM (1, 1) [55] to predict the $\mathcal{H}$ values of the next k steps, and used the predicted values to construct a selector that decides whether to switch different backbone 3D detectors. The schematic diagram of this process is shown in Figure 6.

Assuming that we have obtained a sequence $\mathbb{H}^{(0)} = \left(\mathcal{H}^{(0)}(1), \mathcal{H}^{(0)}(2), \cdots, \mathcal{H}^{(0)}(n)\right)$, we test whether its ratio

$$\delta(k) = \frac{\mathcal{H}^{(0)}(k-1)}{\mathcal{H}^{(0)}(k)},\ k = 2, 3, \cdots, n$$

satisfies condition

$$\delta(k) \in \left(e^{-\frac{2}{n+1}}, e^{\frac{2}{n+1}}\right)$$

If this condition is satisfied, the grey model GM (1, 1) can be introduced at the present time. Let $\mathbb{H}^{(1)}$ be the 1-AGO sequence of $\mathbb{H}^{(0)}$:

$$\mathbb{H}^{(1)} = \left(\mathcal{H}^{(1)}(1), \mathcal{H}^{(1)}(2), \cdots, \mathcal{H}^{(1)}(n)\right) \tag{14}$$

$$\mathcal{H}^{(1)}(k) = \sum_{i=1}^{k} \mathcal{H}^{(0)}(i),\ k = 2, 3, \cdots, n \tag{15}$$

Additionally, the sequence $\mathbb{Z}^{(1)}$ is obtained as the mean generating sequence of $\mathbb{H}^{(1)}$ within its immediate neighborhood:

$$\mathbb{Z}^{(1)} = \left(\mathcal{Z}^{(1)}(2), \mathcal{Z}^{(1)}(3), \cdots, \mathcal{Z}^{(1)}(n)\right) \tag{16}$$

$$\mathcal{Z}^{(1)}(k) = 0.5\mathcal{H}^{(1)}(k) + 0.5\mathcal{H}^{(1)}(k-1) \tag{17}$$

We employ the elementary GM (1, 1) differential equation paradigm

$$\mathcal{H}^{(0)}(k) + a\mathcal{Z}^{(1)}(k) = b \tag{18}$$

where a is the development coefficient; parameters a and b are identified. It is assumed that:

$$\theta = \begin{bmatrix} a \\ b \end{bmatrix}^T,\ Y_n = \begin{bmatrix} \mathcal{H}^{(0)}(2) \\ \mathcal{H}^{(0)}(3) \\ \vdots \\ \mathcal{H}^{(0)}(n) \end{bmatrix},\ B_n = \begin{bmatrix} -\mathcal{Z}^{(1)}(2) & 1 \\ -\mathcal{Z}^{(1)}(3) & 1 \\ \vdots & \\ -\mathcal{Z}^{(1)}(n) & 1 \end{bmatrix}$$

Then, matrix form can be written as:

$$Y_n = B_n\theta \tag{19}$$

The objective function of least square method is established as:

$$J\left(\hat{\theta}_n\right) = \left(Y - B_n\hat{\theta}_n\right)^T\left(Y - B_n\hat{\theta}_n\right) \tag{20}$$

The least square method is used to obtain the estimated value $\hat{\theta}_n$ of $\theta = [a, b]^T$:

$$\hat{\theta}_n = \left(B_n^T B_n\right)^{-1} B_n^T Y_n \tag{21}$$

We can obtain the whitening function of the GM (1, 1) model:

$$\frac{d\mathcal{H}^{(1)}}{dt} + a\mathcal{H}^{(1)} = b \tag{22}$$

The time response function of the available Equation (20) under initial conditions is:

$$\mathcal{H}^{(1)}(k) = \left(\mathcal{H}^{(0)}(1) - \frac{b}{a}\right)e^{-a(k-1)} + \frac{b}{a} \tag{23}$$

The 1-IAGO operation is applied to Equation (7), and the simulated sequence is obtained:

$$\mathcal{H}^{(0)}(k) = \mathcal{H}^{(1)}(k) - \mathcal{H}^{(1)}(k-1),\ k = 2, 3, \cdots, n \tag{24}$$

By using the above prediction methods, we can obtain a series of predicted values when $\mathcal{H}$ shows large fluctuations. These predicted values predict the information complexity of the streaming inputs, which we use as a basis for switching the detection algorithms for different backbones (V2-99, R50, and R101 were selected for the next experiments). Notably, this switcher can be applied to any modern 3D detector (e.g., in the experiment, BEVDepth-GM is built on BEVDepth). Furthermore, experimental trials were conducted on the pipeline switch, revealing that within the overall task loop cycle of our entire benchmark process (10 ms), the pipeline switch was accomplished. Hence, we assert that the switching pipeline is lightweight, with a negligible effect on streaming latency.

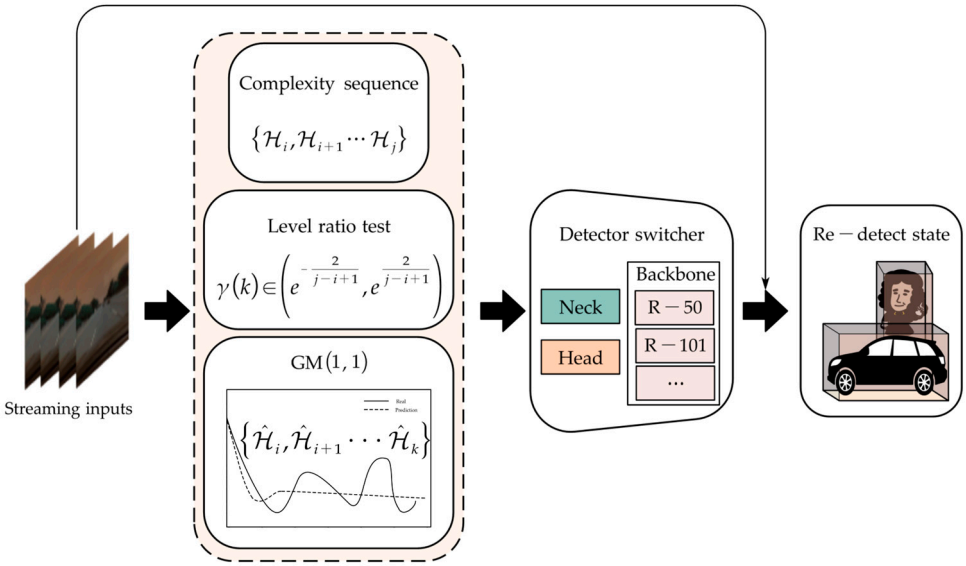

Figure 6. The streaming input updating switcher, where the GM (1, 1) is utilized to predict the future complexity of streaming inputs and is used as a basis for switching the backbone structure in the detector switcher and updating the re-detect state.

4. Experiments

In this section, the experiment setup is first provided. Afterward, we explored evaluating different levels of input complexity, such as streaming inputs and online performance evaluation. Finally, we conducted real-vehicle experiments to confirm the accuracy of our benchmark.

4.1. Experiment Setup

In our benchmark, on the one hand, inputs were subjected to complexity analysis and, on the other hand, vision-centric perception was instantiated as camera-based 3D detection. The dataset in Section 3.2 was used to evaluate the inputs as well as the 3D detection. We followed the methods in [35,37] and used a hardware simulator for the streaming evaluation. The benchmark experiments were conducted on a dedicated industrial control computer—specifically, the Nuvo-10108gc model, featuring an 8-core 16-thread Intel Core i9-12900K CPU, 64 GB RAM, and an NVIDIA RTX4090 GPU. All programs were executed on a ROS system running Ubuntu 20.04 for CPU-related tasks. For the evaluation of open-source detectors based on camera input, the measurements were performed using a batch size of 1 with the respective open-source code (GPU).

It is noteworthy that despite the utilization of an advanced hardware setup in the industrial control computer, limitations were still evident during the testing of all 3D detectors. To ensure the execution of our benchmark on hardware platforms with constrained resources, we introduced quantization techniques, as referenced in [56,57], to compress streaming inputs and the 3D detector under evaluation. This approach facilitated a more equitable performance test while maintaining computational resource constraints.

4.2. Benchmarking Results

Using the ASIO benchmark, we analyzed seven modern autonomous driving 3D detectors (FCOS3D [50], PGD [58], BEVDet [1], BEVDet4D [2], BEVFormer [5], PETR [6], BEVDepth [4]) and our proposed BEVDepth-GM with three kinds of inputs from major sensors (see Table 5). As shown in Table 6, we observed that:

(1) The 3D detectors evaluated showed a significant performance drop with the ASIO benchmark compared to the offline evaluation. When equipped with camera #1, the 3D detectors BEVFormer, FCOS3D, and PGD, which have high computational resource requirements (GFLOPs > 1300), suffered a decrease of 27.5%, 31.0%, and

42.0% in mAP-A, respectively, compared to the offline evaluation. For the efficient detectors (frame rate > 11) BEVDepth, BEVDet, and BEVDet4D, the mAP-A still dropped by 16.3%, 14.2%, and 16.0%, respectively.

(2) Streaming input metrics $\mathcal{H}$ and $\mathcal{D}$ served as an indicator of the inference speed and accuracy of streaming perception during the evaluation. As the metric $\mathcal{H}$ of streaming inputs increased from 0.210@#2 to 0.221@#1, the inference frame rates of FCOS3D, PGD, BEVDet, BEVDet4D, BEVFormer, PETR, and BEVDepth dropped by 51.4%, 47.2%, 24.4%, 49.4%, 51.5%, 53.8%, and 51.4%, respectively. As the metric $\mathcal{D}$ increased from 0.0152@#2 to 0.0183@#3, the mAP-A of the methods mentioned above increased by 10.9%, 13.4%, 49.8%, 27.9%, 33.3%, 61.0%, and 13.7%, respectively. These observations reveal that among all detectors subjected to our testing, there existed an inverse correlation between the $\mathcal{H}$-value and the inference speed when deploying detectors with different combinations of major cameras. Conversely, a positive correlation was observed between the $\mathcal{D}$-value and the online detection accuracy. Therefore, predictive assessments of their real-time inference timeliness and accuracy during actual deployment can be achieved through pre-established $\mathcal{H}$ and $\mathcal{D}$ indices.

(3) Under different types of major cameras, the magnitude of model performance variation varied widely. As illustrated in Figure 1, a substantial decline in mAP-A was evident for the efficient models BEVDet, BEVDet4D, and PETR when transitioning from high-definition cameras #1 and #3 to wide-angle camera #2. Specifically, the mAP-A values experienced significant reductions of at least 30.4%, 18.7%, and 31.4%, respectively. In contrast, models with high computational resource requirements, FCOS3D and PGD, exhibited relatively modest performance decrements, amounting to 4.7% and 7.9%, respectively. Notably, the accuracy rankings of these two models were inverted between offline and online testing. This inversion underscores the greater practical significance of ASIO benchmarking for the actual deployment of the perception system as compared to offline testing.

(4) The backbone switcher modulated the load on the model's computational resources to compensate for inference delays, thus improving streaming perception. The BEVDepth-GM, equipped with our backbone switcher, demonstrated mAP-A improvements of 9.6%, 13.7%, and 8.0% on major cameras #1, #2, and #3, respectively. Additional test results presented in Table 7 indicate that FCOS3D, PGD, and BEVFormer achieved mAP-A enhancements of 6.2%, 4.5%, and 10.6%, respectively, on camera #1. It is noteworthy that, owing to its lower model complexity, BEVFormer-GM exhibited more significant improvements (i.e., GFLOPs@BEVFormer was 1322.2, which was significantly lower than that of FCOS3D (2008.2) and PGD (2223.0)). This underscores the efficacy of switching the backbone in practical deployment scenarios, particularly for simpler models. Furthermore, our observations reveal that the backbone switcher had a more pronounced impact on the AP-A of high-speed objects. For example, on BEVFormer@#1, the AP-A for cars and buses increased by 8.5% and 8.9%, respectively, whereas the AP-A for slow-speed objects (pedestrians) saw an increase of 0.8%. This insight can inform future streaming algorithms to concurrently consider major camera selection and speed differences among different object categories.

Table 5. Major sensors for the three input types used for the experiments.

Camera Number	FOV/° H × V	Resolution	Frame Rate
#1	90 × 65	1280 × 720	25
#2	123 × 116	848 × 800	25
#3	70 × 55	1024 × 768	25

Table 6. Comparison of the results of different autonomous driving 3D detectors on our dataset validation set, where BEVDepth-GM is based on our switcher built on BEVDepth. For different major cameras, we used the metrics in Section 3.3.1. For streaming = ×, we used the offline metrics. For streaming = √, we used the online metrics in Section 3.3.2.

Methods	Major Camera	FPS	GFLOPs	Streaming	$\mathcal{H}$	$\mathcal{D}$(10×)	ADS(NDS)↑ [1]	mAP(-A)↑
FCOS3D	#1	-	2008.2	×	-	-	0.387	0.310
	#1	1.7	2008.2	√	0.221	0.177	0.326	0.214
	#2	3.5	2008.2	√	0.210	0.152	0.318	0.205
	#3	2.0	2008.2	√	0.218	0.183	0.333	0.227
PGD	#1	-	2223.0	×	-	-	0.416	0.369
	#1	1.9	2223.0	√	0.221	0.177	0.337	0.214
	#2	3.6	2223.0	√	0.210	0.152	0.312	0.197
	#3	2.3	2223.0	√	0.218	0.183	0.357	0.223
BEVDet	#1	-	215.3	×	-	-	0.415	0.337
	#1	26.0	215.3	√	0.221	0.177	0.411	0.289
	#2	34.4	215.3	√	0.210	0.152	0.353	0.201
	#3	25.2	215.3	√	0.218	0.183	0.407	0.301
BEVDet4D	#1	-	222.0	×	-	-	0.480	0.375
	#1	17.0	222.0	√	0.221	0.177	0.441	0.315
	#2	33.6	222.0	√	0.210	0.152	0.394	0.256
	#3	18.6	222.0	√	0.218	0.183	0.450	0.328
BEVFormer	#1	-	1322.2	×	-	-	0.517	0.453
	#1	3.2	1322.2	√	0.221	0.177	0.461	0.329
	#2	6.6	1322.2	√	0.210	0.152	0.404	0.251
	#3	5.0	1322.2	√	0.218	0.183	0.470	0.335
PETR	#1	-	297.2	×	-	-	0.371	0.333
	#1	8.0	297.2	√	0.221	0.177	0.347	0.271
	#2	17.3	297.2	√	0.210	0.152	0.280	0.186
	#3	8.5	297.2	√	0.218	0.183	0.351	0.300
BEVDepth	#1	-	662.6	×	-	-	0.481	0.387
	#1	11.8	662.6	√	0.221	0.177	0.469	0.324
	#2	24.3	662.6	√	0.210	0.152	0.422	0.289
	#3	14.6	662.6	√	0.218	0.183	0.506	0.329
BEVDepth-GM	#1	14.2	662.6	√	0.221	0.177	0.466	0.355
	#2	33.6	662.6	√	0.210	0.152	0.421	0.329
	#3	19.2	662.6	√	0.218	0.183	0.506	0.359

[1] The upward arrow signifies that higher numerical values are indicative of a more favorable evaluation.

Table 7. mAP-A of FCOS3D, PGD, and BEVFormer and the corresponding models with backbone switchers. The experiments were conducted under the deployment of major camera #1 while we reported AP-A for high-speed categories (e.g., cars, buses) and slow-speed categories (e.g., pedestrians).

Method	mAP-A↑ [1]	AVE↓ [2]	Car AP-A	Bus AP-A	Ped. AP-A
FCOS3D	0.214	1.283	0.238	0.105	0.289
FCOS3D-GM	0.227 (+6.2%)	1.233	0.265	0.129	0.312
PGD	0.213	1.220	0.233	0.099	0.297
PGD-GM	0.223 (+4.5%)	1.216	0.256	0.127	0.299
BEVFormer	0.331	0.379	0.366	0.313	0.398
BEVFormer-GM	0.366 (+10.6%)	0.380	0.397	0.341	0.401

[1,2] The upward arrow signifies that higher numerical values are indicative of a more favorable evaluation, whereas a downward arrow indicates that a more favorable evaluation is associated with lower numerical values.

The aforementioned experimental findings underscore the substantial impact of different types of inputs on the performance of streaming perception. Although efficient

detectors such as BEVDet4D, BEVDet, and PETR exhibited excellent performance in offline testing and high-resolution inputs, their performance experienced significant degradation when exposed to wide-field-of-view streaming inputs. In contrast, complex models such as FCOS3D, PGD, and BEVDepth demonstrated more consistent performance across various streaming inputs. Furthermore, by conducting pre-assessments of complexity through evaluations of $\mathcal{H}$ and $\mathcal{D}$ for streaming inputs from different major cameras, one can effectively estimate the efficiency and accuracy of streaming perception for these 3D detectors under such input conditions.

4.3. Analysis on Computational Resource Sharing

Typically, the same major camera may be employed for multiple tasks as shared streaming inputs. To analyze the performance fluctuations induced by shared computational resources, we evaluated the 3D detectors (BEVFormer and BEVDepth) on a GPU (RTX4090) concurrently processing N classification tasks based on ResNet18. As illustrated in Figure 7, as the number of classification tasks increased, the performance of BEVFormer and BEVDepth declined due to the reduction in computational resources allocated to 3D detection tasks. Specifically, as the number of classification tasks increased from 0 to 10, the mAP-S of BEVFormer and BEVDepth decreased by 49.5% and 20.2%, respectively. It is noteworthy that the proposed backbone switcher consistently enhanced streaming performance under computational sharing conditions. When executing 10 classification tasks, the mAP-A of BEVDepth-GM and BEVFormer-GM increased by 10.9% and 15.9%, respectively.

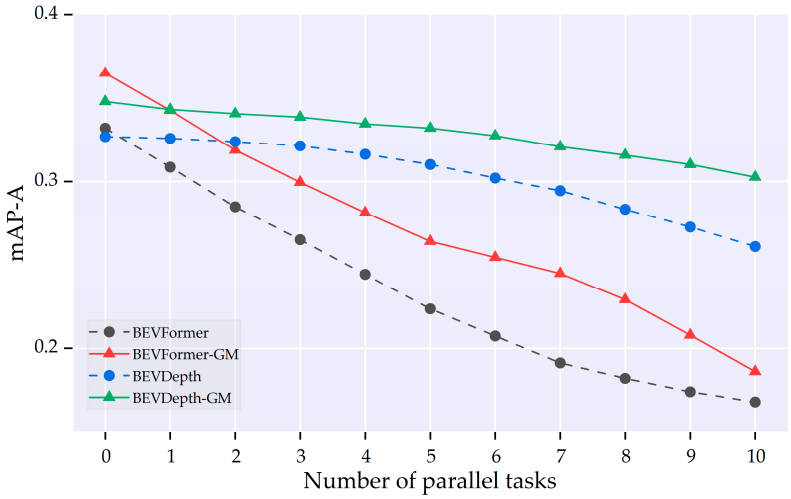

Figure 7. Comparison of the streaming performance of BEVFormer, BEVFormer-GM, BEVDepth, and BEVDepth-GM on major camera #1 under computational resource sharing. The x-axis denotes the number of parallel ResNet18-based classification tasks.

5. Discussion and Conclusions

This paper introduces the ASIO benchmark for evaluating the online performance of vision-centric autonomous driving perception systems under various streaming inputs. Specifically, we establish an evaluation dataset based on the CARLA Leaderboard, serving dual purposes: estimating the information complexity of streaming inputs in advance and validating camera-based streaming 3D detection. The evaluation metrics encompass two components—an information complexity assessment metric involving a fractional–dimensional two-dimensional entropy specifically tailored to input information from different major cameras, and a performance evaluation metric based on ground truth different than offline methods. Additionally, we propose the ASIO switcher based on the real-time input's information complexity to address abrupt changes in input information for 3D detectors, consistently achieving superior streaming performance across three major cameras. Leveraging the ASIO benchmark, we investigate the online performance of seven

representative 3D detectors under different streaming inputs. The experimental results demonstrate that the information complexity of streaming inputs can be utilized to predict the online practical deployment performance of 3D detectors. Furthermore, considerations of the model's parallel computational budget and the selection of backbones based on varying information complexities should be incorporated into the design considerations for practical deployment. Although the proposed ASIO benchmark represents a significant stride towards practical vision-centric perception in autonomous driving, several limitations warrant further investigation in future research: (1) Autonomous driving extends beyond the GPU perception data path, and future research endeavors should encompass a comprehensive evaluation of the entire autonomous driving perception pipeline, encompassing streaming inputs, perception algorithms, and hardware constraints; (2) the establishment of more comprehensive and enriched datasets is needed to adequately address performance testing of streaming perception from input to algorithm and computational platforms; (3) in real-world deployment, an extremely diverse sensor configuration is adopted, encompassing multi-camera setups, infrared sensors, and even event cameras, necessitating the development of a more generalized and unified description for such configurations; (4) the assessment of real-time inputs and corresponding strategies for 3D detectors merit further research; and (5) algorithms geared towards multitasking and end-to-end approaches should encompass a broader spectrum of autonomous driving tasks, such as depth estimation and dynamic tracking, requiring inclusion in the computation of evaluation metrics.

Author Contributions: Conceptualization, T.J. and W.D.; formal analysis, M.Y.; investigation, T.J.; methodology, T.J.; resources, M.Y.; supervision, W.D. and M.Y.; validation, T.J., H.Z. and P.D.; visualization, P.D. and H.Z.; writing—original draft, T.J.; writing—review and editing, T.J. and W.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Sichuan Province under grant/award number 2023NSFSC0395 and the SWJTU Science and Technology Innovation Project under grant/award number 2682022CX008.

Data Availability Statement: No new data was created or analyzed in this manuscript, data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Huang, J.; Huang, G.; Zhu, Z.; Ye, Y.; Du, D. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv* **2021**, arXiv:2112.11790.
2. Huang, J.; Huang, G. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv* **2022**, arXiv:2203.17054.
3. Li, Y.; Bao, H.; Ge, Z.; Yang, J.; Sun, J.; Li, Z. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 1486–1494.
4. Li, Y.; Ge, Z.; Yu, G.; Yang, J.; Wang, Z.; Shi, Y.; Sun, J.; Li, Z. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 1477–1485.
5. Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; Dai, J. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 1–18.
6. Liu, Y.; Wang, T.; Zhang, X.; Sun, J. Petr: Position embedding transformation for multi-view 3d object detection. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 531–548.
7. Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, A.; Wang, T.; Zhang, X. Petrv2: A unified framework for 3d perception from multi-camera images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 3262–3272.
8. Jiang, Y.; Zhang, L.; Miao, Z.; Zhu, X.; Gao, J.; Hu, W.; Jiang, Y.-G. Polarformer: Multi-camera 3d object detection with polar transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 1042–1050.
9. Li, Y.; Chen, Y.; Qi, X.; Li, Z.; Sun, J.; Jia, J. Unifying voxel-based representation with transformer for 3d object detection. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 18442–18455.

10. Park, J.; Xu, C.; Yang, S.; Keutzer, K.; Kitani, K.; Tomizuka, M.; Zhan, W. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. *arXiv* **2022**, arXiv:2210.02443.
11. Li, Q.; Wang, Y.; Wang, Y.; Zhao, H. Hdmapnet: An online hd map construction and evaluation framework. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 4628–4634.
12. Pan, B.; Sun, J.; Leung, H.Y.T.; Andonian, A.; Zhou, B. Cross-view semantic segmentation for sensing surroundings. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4867–4873. [CrossRef]
13. Peng, L.; Chen, Z.; Fu, Z.; Liang, P.; Cheng, E. BEVSegFormer: Bird's Eye View Semantic Segmentation From Arbitrary Camera Rigs. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 5935–5943.
14. Liu, Y.; Yuan, T.; Wang, Y.; Wang, Y.; Zhao, H. Vectormapnet: End-to-end vectorized hd map learning. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 22352–22369.
15. Liao, B.; Chen, S.; Wang, X.; Cheng, T.; Zhang, Q.; Liu, W.; Huang, C. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv* **2022**, arXiv:2208.14437.
16. Akan, A.K.; Güney, F. Stretchbev: Stretching future instance prediction spatially and temporally. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 444–460.
17. Hu, A.; Murez, Z.; Mohan, N.; Dudas, S.; Hawke, J.; Badrinarayanan, V.; Cipolla, R.; Kendall, A. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15273–15282.
18. Kendall, A.; Hawke, J.; Janz, D.; Mazur, P.; Reda, D.; Allen, J.-M.; Lam, V.-D.; Bewley, A.; Shah, A. Learning to drive in a day. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8248–8254.
19. Arshad, S.; Sualeh, M.; Kim, D.; Nam, D.V.; Kim, G.-W. Clothoid: An integrated hierarchical framework for autonomous driving in a dynamic urban environment. *Sensors* **2020**, *20*, 5053. [CrossRef]
20. Zhu, Z.; Zhao, H. Learning Autonomous Control Policy for Intersection Navigation With Pedestrian Interaction. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3270–3284. [CrossRef]
21. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
22. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef]
23. Yin, T.; Zhou, X.; Krahenbuhl, P. Center-based 3d object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11784–11793.
24. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
25. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
26. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
27. Geyer, J.; Kassahun, Y.; Mahmudi, M.; Ricou, X.; Durgesh, R.; Chung, A.S.; Hauswald, L.; Pham, V.H.; Mühlegg, M.; Dorn, S. A2d2: Audi autonomous driving dataset. *arXiv* **2020**, arXiv:2004.06320.
28. Huang, X.; Wang, P.; Cheng, X.; Zhou, D.; Geng, Q.; Yang, R. The apolloscape open dataset for autonomous driving and its application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2702–2719. [CrossRef] [PubMed]
29. Neuhold, G.; Ollmann, T.; Rota Bulo, S.; Kontschieder, P. The mapillary vistas dataset for semantic understanding of street scenes. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4990–4999.
30. Scheel, O.; Bergamini, L.; Wolczyk, M.; Osiński, B.; Ondruska, P. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In Proceedings of the Conference on Robot Learning, Auckland, New Zealand, 14–18 December 2022; pp. 718–728.
31. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
32. Wilson, B.; Qi, W.; Agarwal, T.; Lambert, J.; Singh, J.; Khandelwal, S.; Pan, B.; Kumar, R.; Hartnett, A.; Pontes, J.K. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv* **2023**, arXiv:2301.00493.
33. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2636–2645.
34. Zhang, S.; Benenson, R.; Schiele, B. Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3213–3221.

35. Wang, X.; Zhu, Z.; Zhang, Y.; Huang, G.; Ye, Y.; Xu, W.; Chen, Z.; Wang, X. Are We Ready for Vision-Centric Driving Streaming Perception? The ASAP Benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 9600–9610.
36. Carla Autonomous Driving Leaderboard. 2021. Available online: https://leaderboard.carla.org/leaderboard/ (accessed on 17 November 2021).
37. Li, M.; Wang, Y.-X.; Ramanan, D. Towards streaming perception. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part II 16, 2020; pp. 473–488.
38. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
39. Ghosh, A.; Nambi, A.; Singh, A.; Yvs, H.; Ganu, T. Adaptive streaming perception using deep reinforcement learning. *arXiv* **2021**, arXiv:2106.05665.
40. Han, W.; Zhang, Z.; Caine, B.; Yang, B.; Sprunk, C.; Alsharif, O.; Ngiam, J.; Vasudevan, V.; Shlens, J.; Chen, Z. Streaming object detection for 3-d point clouds. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 423–441.
41. Peng, C.-K.; Buldyrev, S.V.; Havlin, S.; Simons, M.; Stanley, H.E.; Goldberger, A.L. Mosaic organization of DNA nucleotides. *Phys. Rev. E* **1994**, *49*, 1685. [CrossRef]
42. Warfield, J.N. Societal systems planning, policy and complexity. *Cybern. Syst.* **1978**, *8*, 113–115. [CrossRef]
43. Rényi, A. On measures of entropy and information. In Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics, Berkeley, CA, USA, 20 June–30 July 1960; pp. 547–562.
44. Tsallis, C. Possible generalization of Boltzmann-Gibbs statistics. *J. Stat. Phys.* **1988**, *52*, 479–487. [CrossRef]
45. Pincus, S.M. Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. USA* **1991**, *88*, 2297–2301. [CrossRef]
46. Pincus, S.M.; Goldberger, A.L. Physiological time-series analysis: What does regularity quantify? *Am. J. Physiol. Heart Circ. Physiol.* **1994**, *266*, H1643–H1656. [CrossRef]
47. Richman, J.S.; Moorman, J.R. Physiological time-series analysis using approximate entropy and sample entropy. *Am. J. Physiol. -Heart Circ. Physiol.* **2000**, *278*, H2039–H2049. [CrossRef]
48. Ubriaco, M.R. Entropies based on fractional calculus. *Phys. Lett. A* **2009**, *373*, 2516–2519. [CrossRef]
49. Machado, J.T. Fractional order generalized information. *Entropy* **2014**, *16*, 2350–2361. [CrossRef]
50. Li, E.; Wang, S.; Li, C.; Li, D.; Wu, X.; Hao, Q. Sustech points: A portable 3d point cloud interactive annotation platform system. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1108–1115.
51. Qian, X.; Liu, C.; Qi, X.; Tan, S.-C.; Lam, E.; Wong, N. Context-Aware Transformer for 3D Point Cloud Automatic Annotation. *arXiv* **2023**, arXiv:2303.14893. [CrossRef]
52. Wang, T.; Zhu, X.; Pang, J.; Lin, D. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 913–922.
53. Wang, Y.; Guizilini, V.C.; Zhang, T.; Wang, Y.; Zhao, H.; Solomon, J. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In Proceedings of the Conference on Robot Learning, Auckland, New Zealand, 14–18 December 2022; pp. 180–191.
54. Xiong, H.; Shang, P.; Zhang, Y. Fractional cumulative residual entropy. *Commun. Nonlinear Sci. Numer. Simul.* **2019**, *78*, 104879. [CrossRef]
55. Liu, S.; Chunwu, Y.; Dazhi, C. Weapon equipment management cost prediction based on forgetting factor recursive GM (1, 1) model. *Grey Syst. Theory Appl.* **2020**, *10*, 38–45. [CrossRef]
56. Huang, Q. Weight-quantized squeezenet for resource-constrained robot vacuums for indoor obstacle classification. *AI* **2022**, *3*, 180–193. [CrossRef]
57. Huang, Q.; Tang, Z. High-Performance and Lightweight AI Model for Robot Vacuum Cleaners with Low Bitwidth Strong Non-Uniform Quantization. *AI* **2023**, *4*, 531–550. [CrossRef]
58. Wang, T.; Xinge, Z.; Pang, J.; Lin, D. Probabilistic and geometric depth: Detecting objects in perspective. In Proceedings of the Conference on Robot Learning, Auckland, New Zealand, 14–18 December 2022; pp. 1475–1485.