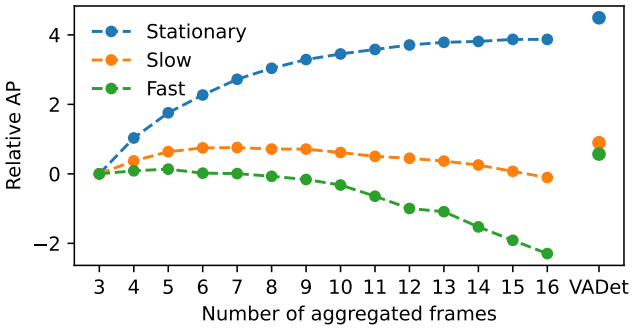
VADet: Multi-frame LiDAR 3D Object Detection using Variable Aggregation

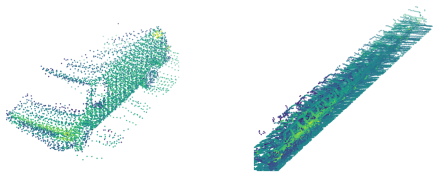
Chengjie Huang Vahdat Abdelzad Sean Sedwards Krzysztof Czarnecki
University of Waterloo
{c.huang,vahdat.abdelzad,sean.sedwards,k2czarne}@uwaterloo.ca

Abstract

Input aggregation is a simple technique used by state-of-the-art LiDAR 3D object detectors to improve detection. However, increasing aggregation is known to have diminishing returns and even performance degradation, due to objects responding differently to the number of aggregated frames. To address this limitation, we propose an efficient adaptive method, which we call Variable Aggregation Detection (VADet). Instead of aggregating the entire scene using a fixed number of frames, VADet performs aggregation per object, with the number of frames determined by an object's observed properties, such as speed and point density. VADet thus reduces the inherent trade-offs of fixed aggregation and is not architecture specific. To demonstrate its benefits, we apply VADet to three popular single-stage detectors and achieve state-of-the-art performance on the Waymo dataset.



(a) Vehicle AP performance



(b) Stationary vehicle (c) Fast vehicle

Figure 1. Performance trade-off between stationary (<0.2 m/s), slow ([0.2,10) m/s), and fast (10 m/s) vehicles. The relative AP in (a) illustrates the improvement/degradation relative to using 3-frame fixed aggregation. (b) and (c) are examples of stationary and fast-moving vehicles after 16-frame fixed aggregation.

1. Introduction

LiDAR-based methods give state-of-the-art (SOTA) 3D object detection performance in autonomous driving. While object detectors can produce accurate detections from a single LiDAR point cloud [3, 18], they have been shown to benefit from aggregated input consisting of multiple consecutive frames. A widely adopted method for incorporating multi-frame sequential point clouds is what we term fixed aggregation, where a fixed number of frames are concatenated after ego motion correction, often including timestamps as an additional feature. This is a simple and effective method to augment the input with more spatial and temporal information, without modifying the architecture [1].

However, it has been observed that the effectiveness of fixed aggregation diminishes as more frames are used, eventually degrading the detection performance [2,19]. Previous works attribute this degradation to the motion of the objects. While the point clouds of stationary objects are inherently aligned after aggregation [7], producing denser and more complete geometry (Fig. 1b), dynamic object point clouds become misaligned and distorted from motion (Fig. 1c). Yang et al. [19] notice that such misalignment makes multi-frame aggregation unhelpful, even degrading performance for fast-moving objects. Chen et al. [2] argue that this effect poses an additional challenge due to different dynamic objects having different distorted point cloud patterns. This introduces a performance trade-off, as illustrated in Fig. 1a: the detection of objects at different speeds is optimal using different numbers of frames.

To address this challenge, SOTA multi-frame detectors have made use of attention-based feature-level aggregation to more effectively utilize information from past frames [2, 14, 19]. However, our experiments reveal that there is considerably more performance to be gained from modifying the input, before resorting to specific architectural designs, with the additional complexity and computation costs they entail.

In this work, we therefore propose VADet (Variable Aggregation Detection): a simple and effective alternative to fixed aggregation that sets the level of aggregation of an object according to its properties. As VADet operates at input level, it can be integrated into existing architectures to improve detection performance without significant modifications or computational overhead. Moreover, VADet's low latency supports its use in real-time applications.

The core of VADet is a function η that maps each detected object to an empirically-optimal number of frames to aggregate. To construct η, we propose *random aggregation training* (RAT, Sec. 3.1) to efficiently study the effects of fixed aggregation on detection performance, over a wide range of configurations.

We use RAT to analyze three representative object detection architectures, establishing that in addition to speed, as illustrated in Fig. 1, object point density demonstrates another important trade-off (see Fig. 4). The function η is then constructed based on the training data to map an object's estimated speed and point density to a number of frames to aggregate, a process detailed in Sec. 3.2.

Thanks to the per-object aggregation based on η, VADet can achieve good performance for objects with different speed and density in the same scene. Our results (Sec. 5) show that VADet consistently exceeds the performance of fixed aggregation, for a given architecture, and can surpass the performance of much more complex SOTA approaches.

2. Related Work

Feature-based alignment of sequential point clouds has been explored in 3D object detection. Early work by Luo et al. [13] uses simple concatenation to combine features from multiple point clouds, which presents a trade-off between accuracy and efficiency depending on the feature layer used for fusion. Naive concatenation of feature maps inevitably introduces misalignment at the feature level due to ego motion. Huang et al. [8] instead use an LSTM to encode temporal information as hidden features and address the alignment issue by transforming the feature map using ego motion.

Recently, attention mechanisms have gained popularity in feature fusion and have shown promising results. Yin et al. [20] propose a GRU module equipped with spatiotemporal attention for better feature alignment. 3D-MAN [19] and MPPNet [2] both employ attention mechanisms to combine features generated from a single or few-frame region proposal network to produce more refined detections. To better utilize the rich multi-scale features, TransPillars [14] proposes attention-based feature fusion at the voxel level to preserve the instance and contextual information.

While feature-based multi-frame methods can make effective use of longer temporal input, they often require modifications to the architecture and incur additional computation cost due to the feature transformation and fusion operations. Input-level aggregation, on the other hand, does not require architectural modifications and has been widely adopted by recent work. Caesar et al. [1] show that directly concatenating multiple consecutive ego-motion-corrected point clouds at the input level can not only improve detection performance but also enable velocity prediction for each detected object, using a velocity regression head.

We refer to this strategy as fixed aggregation, in contrast to our proposed variable aggregation. Specifically, in fixed aggregation, each point cloud from previous timestamps undergoes ego-motion correction and is then concatenated with the current frame's point cloud. More formally, let $P_\tau \in \mathbb{R}^{N_\tau \times 3}$ denote N_τ point coordinates at timestamp τ, with the corresponding ego pose $T_\tau \in \mathbb{SE}(3)$ that represents the transformation from the ego LiDAR coordinate system to a common global coordinate system. Then, the aggregated n-frame point cloud at timestamp τ is defined as

$$P^*_\tau = \bigoplus_{i=0}^{n-1} P_{\tau-i}(T_\tau^{-1} T_{\tau-i})^\mathsf{T}, \tag{1}$$

where $\oplus$ denotes the concatenation operation. Transformation $T_\tau^{-1} T_{\tau-i}$ accounts for the motion of ego between timestamps τ and $\tau - i$. In addition to spatial coordinates, intensity, and elongation as point features, a separate channel is used to encode relative timestamps.

3. Method

VADet addresses the performance trade-off associated with fixed aggregation by adaptively aggregating different types of objects with a different number of frames. To this end, we first introduce Random Aggregation Training (RAT) to enable a single detector to handle a wide range of input frame counts. We then describe our variable aggregation strategy.

3.1. Random Aggregation Training

Studying the impact of the number of input frames on different types of objects (e.g., stationary vs. dynamic) is a crucial component of our approach. Existing works tend to demonstrate the performance trade-offs of aggregation by evaluating multiple detectors trained separately with different fixed frame counts [7, 18, 19]. This is computationally expensive and thus is often done only for a few configurations. Moreover, we find that assessing performance differences between different input configurations, which can often be subtle, using multiple separately trained models is prone to high variance.

To efficiently explore the effects of frame counts on detection performance, we introduce random aggregation training (RAT), wherein a single detector is trained with input that has randomly varying numbers of aggregated

frames per scene. To compensate for the increased variety of the input, we increase the number of training epochs accordingly. We find that even though the model's capacity remains unchanged, RAT allows the model to achieve equivalent or slightly better performance than detectors trained on different fixed configurations. This is demonstrated in Table 1.

Table 1. Vehicle AP of a VoxelNeXt [3] detector trained with separate fixed configurations and RAT, evaluated with different input aggregations.

	3-frame	4-frame	8-frame	16-frame
Separate	72.90	73.45	74.70	75.06
RAT	73.38	74.13	75.45	75.74

RAT thus offers several advantages. In terms of studying the effect of the input aggregation, it significantly reduces the computational cost because training a separate detector for each input configuration is no longer required. This enables us to cover a broader range of frame counts than existing work and to more precisely determine the trade-offs for different types of object. Additionally, as the evaluation is done with a single model and varying input configurations, we find that RAT reduces the variance associated with training, providing us with more consistent results.

In this work, we also use RAT as a pre-training strategy. The detector trained with RAT serves as an ideal starting point for our proposed variable aggregation strategy, thanks to its ability to handle multiple input configurations.

3.2. Variable Aggregation

To address the performance trade-off between different types of objects, we propose per-object variable aggregation, which dynamically aggregates each detected object according to its properties, such as speed and point density.

In VADet, we first perform velocity estimation for each detected object. This serves two purposes: first, it allows us to identify the approximate locations of previously detected objects in the current frame using a constant velocity motion model, enabling us to aggregate each region separately using different aggregation strategies; second, it indicates the motion state of the object and is an important factor for determining the number of frames used in the aggregation. To estimate the velocity of the objects, we follow previous methods [1] and add channels to the regression task representing the x and y components of the velocity vector.

Formally, at timestamp τ, we consider a previously detected bounding box $b_{\tau-1}$ in the coordinate system of the current frame with position $\mathbf{x}_{\tau-1}$, dimensions $(l_{\tau-1}, w_{\tau-1}, h_{\tau-1})$, heading $\theta_{\tau-1}$, estimated velocity $\mathbf{v}_{\tau-1}$, and $n_{\tau-1}$ points from the point cloud at timestamp $\tau-1$ inside the bounding box. To achieve better performance, our strategy is to find a function $\eta(b_{\tau-1})$ that gives the empirically best number of frames to aggregate for each object at the current timestamp τ.

3.2.1 Learning Function η

To determine the number of aggregated frames that provide the best detection performance for each object detection, we consider two important factors: speed and point density (number of points per unit surface area). Aggregation changes the appearance of the point clouds for objects of different speeds due to motion distortion, as illustrated in Fig. 1b and Fig. 1c, and thus the optimal number of frames for aggregation varies with the object's speed. Additionally, as more frames are aggregated, the point density increases proportionally, affecting the number of points representing each object and consequently impacting the detection performance.

In practice, since neither of these factors can be accurately determined for a given object, we use the velocity prediction from the object detector to estimate its speed $\|v_{\tau-1}\|$, and approximate its point density $\rho_{\tau-1}$ using the predicted bounding box dimensions:

$$\rho_{\tau-1} = n_{\tau-1} / \left(l_{\tau-1} \cdot w_{\tau-1} + l_{\tau-1} \cdot h_{\tau-1} + w_{\tau-1} \cdot h_{\tau-1}\right) \tag{2}$$

Since the trade-offs for different types of objects can be different for each dataset and architecture, we obtain η empirically by evaluating the object detector's performance on different types of objects over a wide range of input configurations on the training split. This is feasible thanks to RAT allowing the use of a single model for evaluation. Specifically, we implement η as a piecewise function using a lookup table. The table is constructed by dividing the training set into subcategories of objects with different speeds and densities, and determining the frame count that leads to the highest average precision for each subcategory.

While existing works have observed the effects of aggregation on objects with different speeds [2, 7, 19], the way aggregation interacts with objects with different point densities is not well-studied. In VADet, we establish point density as an additional factor that should be considered and evaluated.

3.2.2 Input construction

For each object, we first determine the approximate location $\hat{\mathbf{x}}^*_\tau$ of the object at the current timestamp τ according to the constant velocity model. This is given by

$$\hat{\mathbf{x}}^*_\tau = \mathbf{x}_{\tau-1} + \mathbf{v}_{\tau-1}/f, \tag{3}$$

where f is the frame rate of the LiDAR point clouds.

To encompass all the points belonging to the object, including past points that could potentially fall outside of the

Algorithm 1: Variable Aggregation

Input: point clouds $P_{\tau-n_{\max}+1}, \ldots, P_\tau$; ego poses $T_{\tau-n_{\max}+1}, \ldots, T_\tau$; previous detections $\mathbf{b}_{\tau-1}$

Output: aggregated object points P_τ^{obj}

Initialize P_τ^{obj} to be an empty point cloud
Compute $\hat{\mathbf{b}}_\tau$ from $\mathbf{b}_{\tau-1}$ using Eqs. (3) to (8)
for $i \leftarrow 0$ **to** $n_{\max} - 1$ **do**
  $P_{\tau-i}^{\text{corr}} \leftarrow P_{\tau-i}(T_\tau^{-1} T_{\tau-i})^\mathsf{T}$
  $\hat{\mathbf{b}}_\tau^i \leftarrow \left\{ \hat{b}_\tau \mid \hat{b}_\tau \in \hat{\mathbf{b}}_\tau, \eta(b_{\tau-1}) > i \right\}$
  $P_\tau^{\text{obj}} \leftarrow P_\tau^{\text{obj}} \oplus \text{Crop}(P_{\tau-i}^{\text{corr}}, \hat{\mathbf{b}}_\tau^i)$
return P_τ^{obj}

bounding box, we enlarge the region of aggregation based on the speed of the object and the number of frames used in the aggregation. The final region of aggregation for an object, denoted $\hat{b}_\tau := (\hat{\mathbf{x}}_\tau, \hat{l}_\tau, \hat{w}_\tau, \hat{h}_\tau, \hat{\theta}_\tau)$, is given by

$$\hat{\mathbf{x}}_\tau = \hat{\mathbf{x}}_\tau^* - \frac{\mathbf{v}_{\tau-1} \cdot (\eta(b_{\tau-1}) - 1)}{2f}, \tag{4}$$

$$\hat{l}_\tau = \sigma \cdot l_{\tau-1} + \frac{|\mathbf{v}_{\tau-1}| \cdot (\eta(b_{\tau-1}) - 1)}{f}, \tag{5}$$

$$\hat{w}_\tau = \sigma \cdot w_{\tau-1}, \tag{6}$$

$$\hat{h}_\tau = \sigma \cdot h_{\tau-1}, \tag{7}$$

$$\hat{\theta}_\tau = \theta_{\tau-1}, \tag{8}$$

where $\sigma \geq 1$ is the enlargement factor that adds a margin to the aggregation region. Note that the second term in Eq. (5) is used to enlarge the length of the object to include the misaligned "smudges" from object motion. Accordingly, the center of the aggregation region is adjusted in Eq. (4).

Finally, for each region constructed from the process above, we aggregate the respective number of past frames in that region. In practice, this operation can be efficiently implemented by cropping the regions $\hat{b}_\tau$ for each previous frame $\tau - i$, $i < \eta(b_{\tau-1})$. Point clouds are corrected for ego motion by transformation $T_\tau^{-1} T_{\tau-i}$ before aggregation. This process is detailed in Algorithm 1. The aggregated objects are then combined with the remaining background points outside of the aggregation regions.

4. Experimental Setup

To demonstrate our method is effective and can be easily applied to different architectures, we evaluate VADet using CenterPoint [21], VoxelNeXt [3], and DSVT [18] on the large scale Waymo Open Dataset [16]. CenterPoint, VoxelNeXt, and DSVT respectively represent the SOTA in dense voxel-based, fully sparse, and transformer-based 3D object detectors.

4.1. Dataset

The Waymo dataset is a large-scale autonomous driving dataset collected under a variety of traffic conditions in San Francisco, Phoenix, and Mountain View. It consists of 798 sequences for the training split, 202 sequences for the validation split, and 150 sequences for the held-back test split. Each sequence is approximately 20 seconds long.

Waymo uses multiple LiDAR sensors operating at 10 Hz, resulting in approximately 200 point clouds per sequence. The main sensor is a top-mounted proprietary 64-beam rotating LiDAR. There are also four close-range LiDARs mounted to the side of the vehicle. In addition to the intensity channel, Waymo's sensors also produce elongation for each point. We use points from all five LiDARs by concatenating them in the ego vehicle coordinate system.

4.2. Evaluation Metrics

For overall object detection performance, we use the official Waymo evaluation suite. Unless specified otherwise, we report the level 2 average precision (AP) for Vehicle, which includes very sparse objects (≤ 5 points). The IoU threshold used for matching true positives is 0.7.

Evaluating a specific subset of the objects (e.g., dynamic objects) for more detailed analysis requires more careful handling due to false positives that cannot be matched with any ground truth. Waymo evaluation suite provides functionalities to perform such evaluation. Specifically, the subset precision and recall are defined as follows:

$$\text{Prec}_{\text{waymo}} = \frac{\text{TP}_{\text{subset}}}{\text{TP}_{\text{subset}} + \text{FP}_{\text{subset}} + \text{FP}_{\text{unknown}}}, \tag{9}$$

$$\text{Rec}_{\text{waymo}} = \frac{\text{TP}_{\text{subset}}}{\text{TP}_{\text{subset}} + \text{FN}_{\text{subset}}}. \tag{10}$$

$\text{TP}_{\text{subset}}$ and $\text{FN}_{\text{subset}}$ are the number of true positives and false negatives within the subset of objects. $\text{FP}_{\text{subset}}$ is the number of false positives that overlap with some objects in the subset but do not meet the IoU threshold. $\text{FP}_{\text{unknown}}$ is the number of objects that do not overlap with any ground truth.

However, we argue that this formulation cannot correctly reflect the detection performance of a subset because $\text{FP}_{\text{unknown}}$ is independent of the subset and biases the precision depending on the size of the subset. The consequence is that subsets of different sizes are incomparable and the weighted sum of subset precisions underestimates the precision of the union of subsets.

To address this issue, we introduce a definition of precision that weights $\text{FP}_{\text{unknown}}$ by the proportion of objects in

Table 2. Dynamic vehicle AP of a CenterPoint model on the Waymo dataset using different subset evaluation metrics.

Metric	Dynamic ≥0.2 m/s	Subsets (m/s) [0.2,1)	[1,3)	[3,5)	[5,7)	[7,9)	[9,11)	[11,13)	[13,15)	[15,17)	[17,20)	≥20	Weighted average
Waymo	72.9	63.0	62.4	62.9	66.5	68.9	66.8	64.1	61.8	64.0	67.0	62.9	64.7
Ours	74.4	71.6	69.8	72.2	74.6	76.6	76.3	75.7	75.3	78.3	81.6	78.3	74.4

the subset:

$$\mathrm{Prec}_{\mathrm{subset}} = \frac{\mathrm{TP}_{\mathrm{subset}}}{\mathrm{TP}_{\mathrm{subset}} + \mathrm{FP}_{\mathrm{subset}} + \frac{N_{\mathrm{subset}}}{N_{\mathrm{total}}} \cdot \mathrm{FP}_{\mathrm{unknown}}} \tag{11}$$

N_{subset} and N_{total} are the number of objects in the subset and the total number of objects, respectively. The definition of recall remains unchanged. In contrast to the standard Waymo metric, this formulation allows the comparison and combination of subset APs, as illustrated in Table 2. We note, in particular, that when using our formulation the weighted average of subset APs weighted by subset size is closer to the AP of the union of the subsets.

4.3. Implementation Details

4.3.1 Architectures

CenterPoint [21] and VoxelNeXt [3] are both single stage CNN-based 3D object detectors. The input point cloud first undergoes voxelization and is then fed to a sparse convolutional backbone. Multiple detection heads are used to separately produce bounding box attributes, including confidence score, location, and box dimensions. DSVT [18] is an emerging transformer-based 3D object detector. In DSVT, the traditional convolution backbone is replaced with multiple transformer blocks consisting of shifted window and partition-based self-attention operations.

Following [1], we add a two-layer regression head to predict an object's velocity vector for all architectures. For DSVT, we use the pillar variant (which we denote DSVT-P). As the dynamic voxelization adopted by the original work cannot be scaled beyond 4-frame aggregation on our hardware, we use a traditional static voxelization where for each voxel, at most 40 points are randomly selected and processed. To make the computation tractable, we further reduce the input channels from 192 to 96, and the hidden channels from 384 to 192.

4.3.2 Training

The baseline models are trained on the entire Waymo training split for 20 epochs across 8 NVIDIA A6000 GPUs using the proposed RAT strategy. We use the Adam optimizer and a one-cycle learning rate schedule, with an initial learning rate of 0.0003 and a maximum learning rate of 0.003. We use a total batch size of 32 for CenterPoint, 32 for VoxelNeXt, and 16 for DSVT-P.

We initialize our VADet models with the weights from the baseline models, then fine-tune them for an additional epoch using cosine learning rate decay with an initial learning rate of 0.0001. Since our models rely on information cached from previous frames, including point clouds and predictions, the standard frame-based shuffling cannot be applied during training. Instead, we divide the dataset into mini-sequences with a maximum of 32 frames and shuffle the mini-sequences. This introduces randomness while ensuring that frames appear in their correct sequence. Furthermore, as the performance of a model can fluctuate during training (due to ongoing optimization), following [5], we load the offline predictions from the baseline models using 3-frame fixed aggregation during training.

4.3.3 Specifying η

We implement η as a lookup table: we subdivide the training dataset based on speed and point cloud density and empirically determine the frame count that leads to the highest average precision for each subcategory (Sec. 3.2). The speed and density thresholds are set to [0.00, 0.20, 1.55, 3.63, 5.90, 81.6, 11.34, 17.53] m/s and [0.00, 0.68, 1.86, 3.86, 8.02, 18.81, 71.37] pts/m^2 respectively. They are chosen based on the training set object statistics to ensure a sufficient number of objects in each bin for evaluation. For each speed and density combination, we evaluate the baseline model on 3–16-frame input and select the frame count with the highest AP performance. Background points undergo a fixed 3-frame aggregation.

5. Results and Analysis

We present the overall performance of VADet compared to baseline models and SOTA methods in Section 5.1. To demonstrate the better trade-offs achieved by VADet, in Section 5.3 we perform a detailed evaluation based on speed and point cloud density.

5.1. Overall Performance

Table 3 shows that, for all three architectures, VADet demonstrates superior performance compared to baseline models using different fixed frame counts. In particular, VADet using CenterPoint achieves 71.5 AP compared to 71.0 AP using 12, 13, 14, or 15-frame aggregation. For VoxelNeXt, VADet achieves 76.5 AP compared to 75.8 AP

Table 3. Overall vehicle AP on the Waymo validation split. The best performance is in bold and the second best is underlined.

	VADet	3f	4f	5f	6f	7f	8f	9f	10f	11f	12f	13f	14f	15f	16f
CenterPoint	**71.5**	68.1	68.9	69.5	70.0	70.3	70.5	70.7	70.8	70.9	71.0	71.0	71.0	71.0	70.9
VoxelNeXt	**76.5**	73.4	74.1	74.6	75.0	75.3	75.4	75.6	75.7	75.7	75.8	75.8	75.8	75.8	75.7
DSVT-P	**74.5**	69.5	70.3	70.8	71.4	71.7	72.0	72.2	72.4	72.6	72.7	72.8	73.0	73.0	73.2

Table 4. Overall vehicle performance on the Waymo validation split compared with SOTA multi-frame detectors.

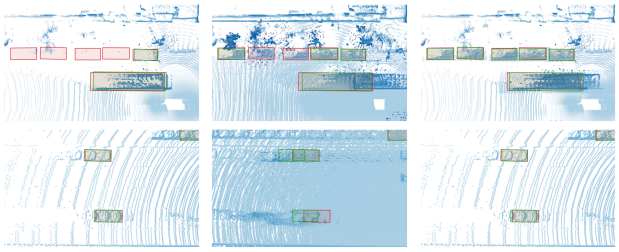
Method	# frames	L1 AP/APH	L2 AP/APH
PillarNeXt-B [9]	3	80.6/80.1	72.9/72.4
DSVT-pillar [18]	4	81.7/81.2	73.8/73.4
DSVT-voxel [18]	4	81.8/81.4	74.1/73.6
FSD++ [5]	7	81.4/80.9	73.3/72.9
CenterFormer [22]	8	78.8/78.3	74.3/73.8
3D-MAN [19]	16	74.5/74.0	67.6/67.1
MPPNet [2]	16	82.7/82.3	75.4/75.0
VADet-CenterPoint	3–16	79.0/78.5	71.7/71.3
VADet-DSVT-P	3–16	82.0/81.6	74.5/74.1
VADet-VoxelNeXt	3–16	**83.9/83.4**	**76.6/76.1**

Table 5. Overall vehicle performance on the Waymo test split compared with other methods (without TTA or ensemble).

Method	# frames	Modality	L2 AP/APH
AFDetV2 [6]	2	L	74.3/73.9
PV-RCNN++ [15]	2	L	76.3/75.9
SWFormer [17]	3	L	75.0/74.7
PillarNeXt-B [9]	3	L	76.2/75.8
FSD++ [5]	7	L	77.1/76.7
3D-MAN [19]	16	L	70.4/70.0
MPPNet [2]	16	L	77.3/76.9
CenterFormer [22]	16	L	78.7/78.3
BEVFusion [12]	3	C+L	77.9/77.5
DeepFusion [11]	5	C+L	76.1/75.7
HorizonLiDAR3D [4]	5	C+L	78.2/77.8
LoGoNet [10]	5	C+L	79.7/79.3
VADet-VoxelNeXt	3–16	L	**79.8/79.4**

using 12, 13, 14, or 15-frame aggregation. DSVT-P with VADet achieves 74.5 AP compared to 73.2 AP using 16-frame aggregation.

When compared with SOTA object detection methods in Table 4 and Table 5, VADet-equipped models also have competitive performance on both the validation and test splits. Most notably, our single-stage VADet-VoxelNeXt achieves 76.1 and 79.4 level 2 APH on the validation and test split respectively, outperforming two-stage multi-frame methods such as MPPNet [2] by a large margin, with lower computation overhead. Specifically, we measure that the second stage proposed by MPPNet introduces an additional 900–2500 ms latency over the base detector, while VADet only requires an additional 50 ms overhead for input aggregation, which we believe can be further optimized with a GPU-accelerated implementation. Furthermore, with a powerful backbone such as VoxelNeXt, our method can match the performance of recent SOTA camera-LiDAR fusion methods such as LoGoNet [10].



(a) 3-frame (b) 16-frame (c) VADet

Figure 2. Qualitative results comparing 3-frame and 16-frame fixed aggregation with VADet. Red and green bounding boxes are ground truth and predictions, respectively. Predictions are filtered with 0.5 confidence threshold for visual clarity.

Our results demonstrate that VADet can effectively utilize multiple frames to achieve SOTA detection performance, suggesting that by addressing various performance trade-offs with carefully constructed input, a simple single-stage object detection architecture such as VoxelNeXt can outperform much more complex SOTA methods.

5.2. Qualitative Results

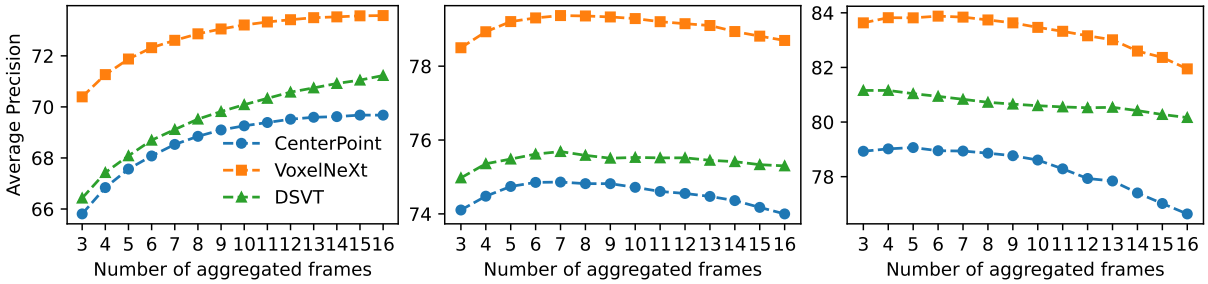
Qualitative results comparing 3-frame and 16-frame fixed aggregation with VADet are shown in Fig. 2. Overall, we observe that 16-frame input appears much denser than 3-frame and VADet. This is especially noticeable for background points, which may not be beneficial to detecting objects and introduce unnecessary computation.

The 3-frame model is unable to detect the occluded stationary vehicles depicted in the upper images, due to the lack of points. While these objects appear more complete with 16 frames, the model can only partially detect them. We hypothesize this is due to the dense background points leading to confusion. VADet, on the other hand, aggregates only object points while keeping the background sparse, and therefore can accurately detect all vehicles.

Similarly, for the fast-moving vehicles seen in the lower images, 16-frame aggregation results in long "smudges" around each vehicle, leading to inaccurate localization. In these situations, VADet does not over-aggregate these objects and benefits from the more accurate detections

Table 6. Vehicle AP breakdown by speed.

	Speed	VADet	3f	4f	5f	6f	7f	8f	9f	10f	11f	12f	13f	14f	15f	16f
CenterPoint	Stationary	**70.3**	65.8	66.8	67.6	68.1	68.5	68.8	69.1	69.3	69.4	69.5	69.6	69.6	69.7	69.7
	Slow	**75.0**	74.1	74.5	74.7	74.9	74.9	74.8	74.8	74.7	74.6	74.6	74.5	74.4	74.2	74.0
	Fast	**79.5**	78.9	79.0	79.1	79.0	78.9	78.9	78.8	78.6	78.3	77.9	77.8	77.4	77.0	76.6
VoxelNeXt	Stationary	**74.3**	70.4	71.3	71.9	72.3	72.6	72.9	73.1	73.2	73.3	73.4	73.5	73.5	73.6	73.6
	Slow	**79.6**	78.5	78.9	79.2	79.3	79.4	79.4	79.3	79.3	79.2	79.2	79.1	78.9	78.8	78.7
	Fast	**84.2**	83.6	83.8	83.8	83.9	83.8	83.7	83.6	83.5	83.3	83.2	83.0	82.6	82.4	81.9
DSVT-P	Stationary	**72.3**	66.4	67.4	68.1	68.7	69.1	69.5	69.8	70.1	70.3	70.6	70.7	70.9	71.0	71.2
	Slow	**77.3**	75.0	75.4	75.5	75.6	75.7	75.6	75.5	75.5	75.5	75.5	75.5	75.4	75.3	75.3
	Fast	**82.6**	81.2	81.2	81.0	80.9	80.8	80.7	80.7	80.6	80.6	80.5	80.5	80.4	80.3	80.2



(a) Stationary (b) Slow (c) Fast

Figure 3. AP vs. the number of frames for stationary (<0.2 m/s), slow ([0.2,10) m/s), and fast-moving (≥10 m/s) vehicles.

achieved by 3-frame input.

5.3. Breakdown Analysis

The overall performance degradation at 16-frame fixed aggregation, if any, appears to be minuscule for the baseline models: just 0.1% degradation for CenterPoint and VoxelNeXt. This is due to the extreme imbalance between different types of objects in the dataset. For instance, the overall results are skewed in favour of the approximately 80% stationary vehicles in the dataset, which negatively affects the performance of dynamic objects. As a result, we perform a breakdown analysis to highlight the performance trade-offs and demonstrate how VADet can benefit objects that are negatively impacted by fixed aggregation.

5.3.1 Speed

To illustrate the impact of input aggregation on objects with different speeds, we divide the vehicle class into stationary (<0.2 m/s), slow ([0.2,10) m/s), and fast (≥10 m/s) subcategories. Stationary, slow, and fast vehicles respectively amount to 79.7%, 14.2%, and 6.1% of the Waymo validation set. The AP performance for each subcategory is reported in Table 6.

Our results for the baseline models are consistent with the trade-off observed in existing work and further suggest that this effect is consistent across different architectures. Moreover, the results show that the optimal number of frames can be different for objects with different speeds. Specifically, the stationary vehicle performance (Fig. 3a) increases as more frames are used for aggregation: all baseline models achieve the best performance with 16-frame input, suggesting more aggregation is beneficial. Slow vehicles (Fig. 3b), on the other hand, reach maximum performance at 7 frames, while for fast-moving vehicles (Fig. 3c), a large degradation can be observed when more frames are used, indicating aggregating fewer frames is more favourable.

VADet, on the other hand, does not exhibit such a performance trade-off. For all three architectures and subcategories, VADet achieves higher AP than the best fixed aggregation configuration in each respective subcategory (underlined in Table 6). This not only highlights the effectiveness of VADet at mitigating the performance trade-off between objects at different speeds but also demonstrates the applicability of VADet to various architectures.

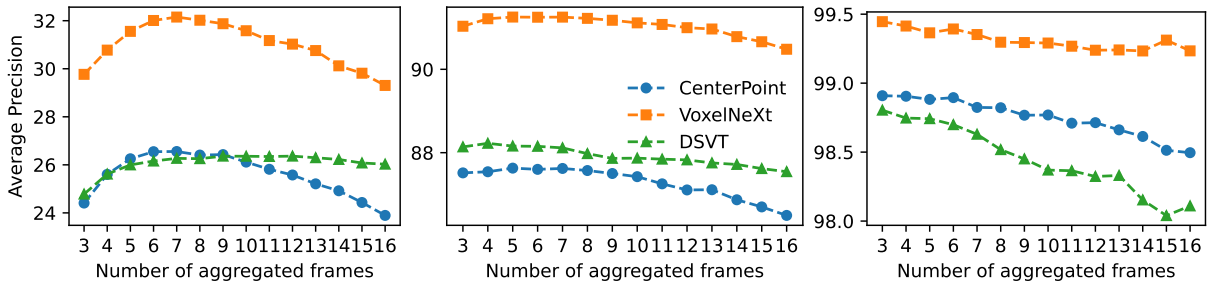
5.3.2 Point density

To demonstrate that input aggregation can also lead to a performance trade-off between objects with different point cloud densities, we evaluate and report the performance of vehicles with different point cloud densities. Using Eq. (2), we divide the vehicle class into sparse (<2 pts/m^2), medium ([2,100) pts/m^2), and dense (≥100 pts/m^2) subcategories.

For stationary objects, we observe the same trend seen

Table 7. Dynamic vehicle AP performance breakdown by point cloud density.

	Density	VADet	3f	4f	5f	6f	7f	8f	9f	10f	11f	12f	13f	14f	15f	16f
CenterPoint	Sparse	**27.6**	24.4	25.6	26.3	26.5	26.6	26.4	26.4	26.1	25.8	25.6	25.2	24.9	24.4	23.9
	Medium	**87.8**	87.5	87.5	87.6	87.6	87.6	87.6	87.5	87.4	87.3	87.1	87.1	86.9	86.7	86.5
	Dense	**98.9**	**98.9**	**98.9**	**98.9**	**98.9**	98.8	98.8	98.8	98.8	98.7	98.7	98.7	98.6	98.5	98.5
VoxelNeXt	Sparse	**32.8**	29.8	30.8	31.6	32.0	32.2	32.0	31.9	31.6	31.2	31.0	30.8	30.1	29.8	29.3
	Medium	**91.5**	91.0	91.2	91.3	91.2	91.3	91.2	91.2	91.1	91.1	91.0	91.0	90.8	90.7	90.5
	Dense	99.3	**99.4**	**99.4**	**99.4**	**99.4**	**99.4**	99.3	99.3	99.3	99.3	99.2	99.2	99.2	99.3	99.2
DSVT-P	Sparse	**28.0**	24.8	25.6	26.0	26.2	26.3	26.3	26.4	26.4	26.3	26.4	26.3	26.2	26.1	26.0
	Medium	**89.9**	88.1	88.2	88.2	88.2	88.1	88.0	87.9	87.9	87.8	87.8	87.8	87.7	87.6	87.5
	Dense	**99.0**	98.8	98.7	98.7	98.7	98.6	98.5	98.5	98.4	98.4	98.3	98.3	98.2	98.0	98.1



(a) Sparse (b) Medium (c) Dense

Figure 4. AP vs. the number of frames for sparse ($<$2 pts/m^2), medium ((2,100] pts/m^2), and dense ($>$100 pts/m^2) dynamic vehicles.

in Fig. 3a, suggesting the performance is not influenced by the point cloud density. The following analysis therefore focuses on dynamic objects ($\geq$0.2 m/s), with the full results for stationary objects in the supplementary material. The dynamic vehicles in the validation set contain 23.9% sparse, 67.9% medium, and 8.2% dense vehicles according to our definition. The AP performance for each subcategory is detailed in Table 7.

While we have previously observed that dynamic objects favour fewer input frame counts, we notice in Fig. 4a that sparse objects can still benefit from more input aggregation: up to 7-frame aggregation for CenterPoint and VoxelNeXt, and up to 12-frame aggregation for DSVT-P. On the other hand, for denser objects (Figs. 4b and 4c), higher frame counts become harmful to the detection performance. All three architectures achieve the best performance for medium density vehicles with 5-frame input, and dense vehicles with 3-frame input. Our results highlight a trade-off that has not been studied in existing work and underscores the necessity of considering point density in our proposed approach.

Compared to fixed aggregation, VADet is comparable to the best performance in each subcategory, surpassing fixed aggregation in many cases. This demonstrates that VADet is also effective at addressing the trade-off between objects with different point densities for different architectures.

6. Limitations and Extensions

Input aggregation adds information to sparse detections, up to the point that an object's motion or other characteristics cause confusion. VADet therefore improves the detection of certain objects by not over-aggregating them, but their detections may nevertheless be sparse and could benefit from aggregation. For such objects, the addition of a different aggregation approach would be necessary.

In this work, we have identified speed and point density as important features to provide as inputs to function η. We hypothesize that a future implementation of η as a more abstract learned function from point clouds to numbers of frames will produce even better results.

In the preceding text, we have focused on the Waymo vehicle class—a common expedient to simplify comparisons. In the supplementary material, we give further results that show VADet is also effective on the Waymo pedestrian class. While VADet's benefits will be class and dataset dependent, we do not anticipate any obvious limitations.

7. Conclusion

We have addressed the inherent performance trade-off of fixed aggregation by proposing VADet, a variable aggregation approach that can be easily applied to different architectures with minimal modifications. Our extensive evaluation shows that VADet can effectively combat the trade-off and achieve SOTA performance.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3, 5

[2] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. MPPNet: Multi-frame feature intertwining with proxy points for 3D temporal object detection. In *European Conf. on Computer Vision (ECCV)*, pages 680–697, Cham, 2022. Springer Nature Switzerland. 1, 2, 3, 6

[3] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. VoxelNeXt: Fully sparse VoxelNet for 3D object detection and tracking. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 21674–21683, 2023. 1, 3, 4, 5

[4] Zhuangzhuang Ding, Yihan Hu, Runzhou Ge, Li Huang, Sijia Chen, Yu Wang, and Jie Liao. 1st place solution for waymo open dataset challenge–3d detection and domain adaptation. *arXiv preprint arXiv:2006.15505*, 2020. 6

[5] Lue Fan, Yuxue Yang, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Super sparse 3D object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12490–12505, 2023. 5, 6

[6] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. AFDetV2: Rethinking the necessity of the second stage for object detection from point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 969–979, 2022. 6

[7] Chengjie Huang, Vahdat Abdelzad, Sean Sedwards, and Krzysztof Czarnecki. SOAP: Cross-sensor domain adaptation for 3D object detection using Stationary Object Aggregation Pseudo-labelling. In *IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, 2024. 1, 2, 3

[8] Rui Huang, Wanyue Zhang, Abhijit Kundu, Caroline Pantofaru, David A. Ross, Thomas Funkhouser, and Alireza Fathi. An LSTM approach to temporal 3D object detection in LiDAR point clouds. In *European Conf. on Computer Vision (ECCV)*, pages 266–282, Cham, 2020. Springer International Publishing. 2

[9] Jinyu Li, Chenxu Luo, and Xiaodong Yang. PillarNeXt: Rethinking network designs for 3D object detection in LiDAR point clouds. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 17567–17576, 2023. 6

[10] Xin Li, Tao Ma, Yuenan Hou, Botian Shi, Yuchen Yang, Youquan Liu, Xingjiao Wu, Qin Chen, Yikang Li, Yu Qiao, et al. Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 17524–17534, 2023. 6

[11] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 17182–17191, 2022. 6

[12] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *Int. Conf. on Robotics and Automation (ICRA)*, pages 2774–2781, 2023. 6

[13] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and Furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[14] Zhipeng Luo, Gongjie Zhang, Changqing Zhou, Tianrui Liu, Shijian Lu, and Liang Pan. TransPillars: Coarse-to-fine aggregation for multi-frame 3D object detection. In *IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, pages 4230–4239, January 2023. 1, 2

[15] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection. *International Journal of Computer Vision*, 131(2):531–551, 2023. 6

[16] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo Open Dataset. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 4

[17] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. SWFormer: Sparse window transformer for 3D object detection in point clouds. In *European Conf. on Computer Vision (ECCV)*, pages 426–442. Springer, 2022. 6

[18] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. DSVT: Dynamic sparse voxel transformer with rotated sets. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 13520–13529, June 2023. 1, 2, 4, 5, 6

[19] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3D-MAN: 3D multi-frame attention network for object detection. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1863–1872, June 2021. 1, 2, 3, 6

[20] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. LiDAR-based online 3D video object detection with graph-based message passing and spatiotemporal transformer attention. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[21] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3D object detection and tracking. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, 2021. 4, 5

[22] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. CenterFormer: Center-based transformer for 3D object detection. In *European Conf. on Computer Vision (ECCV)*, pages 496–513, Cham, 2022. Springer, Springer Nature Switzerland. 6