Ensembling Object Detectors for Effective Video Query Processing

Daren Chao
University of Toronto
Toronto, Canada
drchao@cs.toronto.edu

Nick Koudas
University of Toronto
Toronto, Canada
koudas@cs.toronto.edu

Xiaohui Yu
York University
Toronto, Canada
xhyu@yorku.ca

Yueting Chen
York University
Toronto, Canada
chenyt@yorku.ca

ABSTRACT

Object detection is the foundation of video query processing systems, which have been the subject of active research in recent years. Compared to the use of a single model, model prediction ensembling is an important practical method to improve the accuracy of object detection, albeit at the expense of additional inference time. This paper focuses on video query processing with multiple object detectors, aiming to select an optimal subset of these detectors and combine their detection outputs for each video frame. We seek to strike a balance between enhancing accuracy and minimizing inference time without prior knowledge of the video or the detectors. We first introduce a method to quantify the accuracy of detection outputs utilizing reference models in the absence of ground truth, and propose an algorithm, MES, designed to effectively allocate computational resources for identifying appropriate ensembles. We then refine our proposal and present MES-B, an algorithm that performs ensemble selection within a specified budget, and SW-MES, which adapts to concept drift during ensemble selection. We comprehensively describe and analyze our proposals utilizing real datasets and present the results of a detailed experimental evaluation of varying parameters of interest. Our results demonstrate that our proposed methods significantly improve the effectiveness of ensemble selection, leading to significant optimization in both accuracy and inference time compared to other applicable methods.

1 INTRODUCTION

The rapid advances in computer vision and deep learning offer highly sophisticated algorithms for numerous applications of vast practical significance such as video object detection (OD) [51, 52], activity recognition [11, 24, 54] and other aspects of video analytics [4, 7, 62]. Such algorithms form the foundation of a new generation of data management and query processing systems and techniques that facilitate structured query processing over videos [40, 41, 64]. These systems are able to answer queries involving constraints on query-specified objects [41, 44], with numerous applications in video surveillance automation, sports analytics, news clip analysis, and autonomous driving.

Accurate OD (namely determining the label and bounding box of each object instance in each video frame) is of profound importance to the correctness of subsequent query results in a video query processing system [36, 41, 44, 65]. Depending on the mode of operation of the query processor (offline or streaming), OD is conducted in a pre-processing step or in real time. Model prediction ensembling, a well-studied approach in many domains [3, 20, 25, 43], is an important practical method to improve leaderboard accuracy of OD [31, 37, 45]. As demonstrated in Figure 1, the ensembling approach, *E*, combines the detections produced by two models to obtain a final output with higher accuracy. However, it is evident that the cost of improving accuracy by ensembling models increases inference time. As illustrated in Figure 2, while bringing higher AP (i.e., average precision), ensembling OD models trained on different datasets or environments (e.g., scene types) often increases the overall inference time.

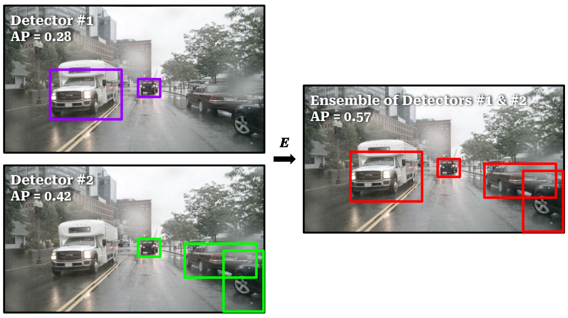


Figure 1: An example of model prediction ensembling: the ensembling approach *E* combines the detections produced by 2 deep models and produces outputs with higher AP (i.e., average precision). Both models have a YOLOv7 [60] architecture and are trained on separate datasets. They both intend to accomplish the vehicle detection task. The colored boxes represent the BBoxes of the vehicles that have been detected by the models or ensemble.

In this paper, we consider ensembling multiple object detectors to enhance overall accuracy when processing queries. Specifically, our approach involves processing a query by employing several object detectors, with the expectation that the query processing engine will select an appropriate subset of these detectors and combine their detection outputs for each video frame. The subset size is not predefined and can change with each frame. To illustrate this concept more concretely, consider the following query:

```
SELECT frameID
  FROM (PROCESS inputVideo PRODUCE frameID, Detections
       USING MES(OD1, OD2, ...; REF))
  WHERE ...
```

Here, `OD1, OD2, ...` are all object detectors; `MES(...; REF)` is the algorithm for selecting detector ensembles and conducting object detection, which is one of our proposals. Our objective is to select the best detector ensembles for each video frame to process the query, while achieving a balance between enhancing accuracy and minimizing inference time, all without prior knowledge of the video and detectors involved.

The payoff in accuracy via ensembling does not appear to increase proportionately to the resources and inference time required to utilize these models jointly. As depicted in Figure 2, the

Series ISSN: 2367-2005

10.48786/edbt.2025.06

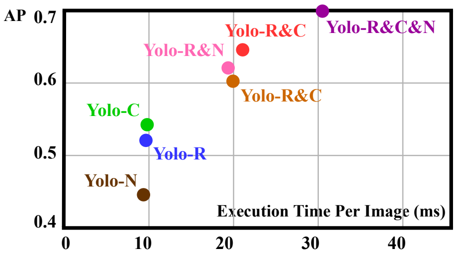


Figure 2: An example of the inference time and AP of three deep models and their ensembles on dataset nuScenes [9]. Each model has a YOLOv7-tiny [60] architecture and is trained on distinct, letter-identified datasets.

ensemble Yolo-R&C&N[1] obtains AP that is 15% higher than Yolo-C, but it requires 3 times as much time for inference. Utilizing ensembling in video query processing requires striking a balance between accuracy and inference time, which in turn necessitates addressing a number of challenges. The ***first*** challenge is to determine which set of models is appropriate for a video frame currently being processed, developing a strategy that can trade off accuracy and inference time. In addition, in contrast to a training phase, in an actual deployment setting, after selecting a set of models for a video frame and obtaining OD results, one cannot calculate the resulting accuracy of detection due to the lack of ground truth labels and bounding boxes for frames. Thus, the ***second*** challenge is the absence of effective measurement to evaluate the accuracy that can provide guidance to subsequent model selection for ensembling purposes. Furthermore, for generality, we make no assumptions regarding the workings of the various models to be ensembled. In video monitoring settings, we expect the surrounding environments, scene types, camera angles, etc. to change. This in turn will instigate the ***last*** challenge–concept drift [26], and the models in the ensemble have to be dynamically adjusted to compensate for the drift.

To address the ***first*** challenge effectively, we propose a novel approach in this paper. We develop a scoring function that incorporates both accuracy and inference time, providing a comprehensive measure of ensemble performance on a video frame. Furthermore, we develop an efficient algorithm to select the optimal ensemble that can maximize the scores obtained from the scoring function. The scoring mechanism is tunable, allowing us to focus on the aspects of detection that are more important for the application at hand (accuracy, inference time). To address the ***second*** challenge, we follow a similar approach as employed in previous studies [39]. Specifically, we require users to provide a pre-trained reference model (REF) that serves as a benchmark for quantifying OD accuracy[2]. In the field of object detection, the LiDAR OD models [46, 49, 50] have been widely leveraged as a common reference model for identifying errors in data management [39]. For the ***last*** challenge, we develop a sliding window algorithm that calibrates the ensemble scores for ensemble selection and subsequently assesses ensembles on temporally neighboring frames as opposed to the entire video, leading to vast gains on final scores.

For starters, in order to select the best ensemble of models that is applicable for a frame, a brute-force approach would enumerate all possible ensembles for each video frame. In contrast, our algorithm focuses on a subset of the ensembles that are more promising to yield higher scores. Our approach leverages the scores of ensembles selected on past frames to guide the selection in the future. We frame the task of **M**odel **E**nsemble **S**election as a decision-making problem under uncertainty and develop an algorithm, called MES, that can efficiently identify and prioritize the most promising ensembles for further examination (by biasing the selection towards such ensembles). Specifically, our algorithm selects an ensemble for each frame and conducts OD on it iteratively, balancing between exploration (i.e., trying more ensembles) vs. exploitation (i.e., focusing on the most promising ensembles). Furthermore, in consideration of the *last* challenge, we enhance the proposed MES algorithm and develop SW-MES, a **S**liding-**W**indow algorithm that adapts to concept drift dynamically and performs well on videos with non-stationary underlying distributions of surrounding environments, scene types, camera angles, etc.

In summary, this paper presents the following contributions.

- We initiate a study in the context of query processing that focuses on selecting ensembles of object detectors for each frame during the preprocessing phase of a video analysis system. The aim is to effectively allocate computational resources for identifying appropriate ensembles. We frame this goal as determining ensembles that maximize the scoring function.
- We propose an algorithm, MES, to address the ensemble selection problem, striking a balance between accuracy and inference time. Our algorithm demonstrates solid improvements in object detection when evaluated using a generic scoring function. We adopt an approach to quantify the accuracy of OD results in the absence of ground truth by utilizing pre-trained reference models. Furthermore, we refine our proposal and present SW-MES, an algorithm designed to adapt to concept drift during ensemble selection.
- We provide a comprehensive experimental evaluation using real-world datasets and various state-of-the-art object detectors trained on diverse datasets to validate the effectiveness of our proposed methods. The results reveal that our proposals lead to a 20% to 50% improvement over other techniques, substantially improving the effectiveness of ensemble selection.

This paper is organized as follows. We formally define the problem in §2 and detail the proposed solutions in §3. §4 analyzes the effectiveness of the algorithms. §5 presents the experimental results. §6 discusses related work, and §7 concludes this paper.

2 BACKGROUND AND PROBLEM DEFINITION

2.1 Object Detection and Model Ensembling

A video is a sequence of frames $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$, where $|\mathcal{V}|$ is the number of frames (length) of the video and can be fixed or unbounded. If $|\mathcal{V}|$ is unbounded, we refer to $\mathcal{V}$ as a video stream. A frame $v \in \mathcal{V}$ may hold a variety of object instances of different types, which can be detected by object detection (OD) algorithms[3] [51, 52]. Each detected object instance is assigned a Bounding Box (BBox) that describes the position of the object in this frame, a confidence value that indicates how confident the detector is

[1] Yolo-R&C&N is the ensemble of three models which are all YOLO-v7 [60] models trained on the three different datasets respectively.

[2] The case where no reference model exists presents a challenging situation and is an interesting direction for future work.

[3] Without specification, we refer to them as camera-based object detectors, distinguishing them from the LiDAR models.

for this prediction, and an estimated label indicating the object type of this instance. The accuracy of OD results on each video frame can be calculated by comparing the ground truth (i.e., GT BBoxes) with the predicted BBoxes [48, 51]. Correct OD results are a significant prerequisite for accurate query processing in a video analysis system.

Without loss of generality, we assume that there are m object detectors available. Such detectors may exhibit varying network structures (such as YOLOv7 [60], MaskRCNN [33]), encompass diverse network complexity (such as YOLOv7 and YOLOv7-Tiny [60]), or are trained on diverse datasets with varying scene types and camera angles, times of day, etc. Note that we are not concerned with the internal specifics of a detector; rather, we treat it as a black box for greater generality of our approach. This way we can easily swap models in our collection of available/applicable models, especially as more advanced detectors continuously become available given the bustling research activities in the area.

Let M_i denote the i-th detector and $\mathcal{M}$ denote the collection of all detectors, $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$. For a frame $v \in \mathcal{V}$, for each detector M_i, the OD results of applying M_i on v, $D_{M_i|v}$, is a set of triplets $\langle BBox, Conf, Label \rangle$, encompassing the coordinates of the BBoxes, the confidence value and the corresponding label for each object. We use $c_{M_i|v}$ to denote the inference time of applying M_i on v, and use $a_{M_i|v}$ to denote the Average Precision[4] (AP) [47], which is a commonly used metric in computer vision to quantify the correctness of the detection [22, 48, 51].

Model ensembling can be done utilizing various approaches, such as Non-Maximum Suppression (NMS) [29, 52], its variants [8, 34, 38] or other BBox-based IOU approaches [12, 56, 61, 67]. Although these approaches for computing the ensemble of OD results need far less resource (incl. time and GPU) than model inference, they are not free. Formally, for a frame v, let S be the set of detectors in an ensemble, where $S \subseteq \mathcal{M}$. We use $D_{S|v}$ to denote the model ensemble results using S on v. Similarly, we use $c_{S|v}$ to denote the inference time of applying ensemble S on v,

$$c_{S|v} = \sum_{M_i \in S} c_{M_i|v} + c^e_{S|v} \quad , \tag{1}$$

where $c_{M_i|v}$ is the inference time of each model M_i and $c^e_{S|v}$ is the extra inference time used for computing the ensemble. Typically, ensembling approaches only need to do lightweight calculations on BBoxes, hence their inference time is much less than that of detectors composed of deep network structures (given the same computing resources) thus $c_{M|v} \gg c^e_{S|v}$. The average precision of the ensemble is denoted by $a_{S|v}$. If there is no ambiguity, we will use S to denote the ensemble of models and consider applying single detectors as a special case of the ensemble where $|S| = 1$.

2.2 Scoring Mechanism

Since the payoff in accuracy gains does not increase proportionately to the resources and inference time required to run all models in the ensemble (as illustrated in Figure 2), we wish to strike a balance between accuracy and inference time. To combine these two measurements together, we propose to use an aggregate *scoring function* to evaluate the ensemble on a video frame. Let SC denote a generic scoring function and $r_{S|v}$ denote the aggregate score of applying ensemble S on frame v, which is computed by the scoring function. The specific form of SC is not crucial for our ensuing proposals as long as the following intuitive criteria are met:

- The score $r_{S|v}$ computed by SC must exhibit a positive (linear or non-linear) correlation with $a_{S|v}$ and a negative (linear or non-linear) correlation with $c_{S|v}$,

$$r_{S|v} a_{S|v} \quad , \quad r_{S|v} - c_{S|v} \quad ;$$

- The scores are normalized in [0, 1];

where represents a linear or non-linear positive correlation. This generic scoring function can assist our ensuing proposals in identifying the best ensembles for specific user requirements. The purpose of the first criterion is to assess the OD results of an ensemble by having the score rise with increasing accuracy or decreasing inference time, and vice versa. The second criterion is employed to calibrate the upper bounds of ensemble scores in our ensuing proposals. Users can assign different weights to the two components involving $a_{S|v}$ and $c_{S|v}$ to make the scoring function tunable to balance their impact; for example, in circumstances where the resource (e.g., available time) is limited, users may set a higher weight for inference time. We will outline the scoring function employed in the experiments in Section 5, which is an example satisfying the criteria presented in this section.

2.3 AP Estimation

Average Precision (AP) is a commonly adopted metric to quantify the accuracy of the detection results. According to [22, 23, 47], the general definition for AP identifies the area under the precision-recall curve representing the value of precision against the recall of the OD results for different confidence threshold values. Given the OD results of an ensemble S on a frame v (denoted by $D_{S|v}$) and the ground truth BBoxes on v (denoted by $BBox_{\mathrm{GT}|v}$), the true AP of S on v, denoted by $a_{S|v}$, is defined as

$$a_{S|v} = AP\left(D_{S|v}\, , BBox_{\mathrm{GT}|v}\right), \tag{2}$$

where $AP(\cdot)$ represents the formula for computing AP as detailed in [22, 23, 47][5]. However, computing the true AP necessitates ground truth labels, which are not always available in our setting. In the absence of ground truth, we will estimate AP utilizing the user-specified reference model, denoted as `REF`. An important observation is that our ensuing proposals do not rely on precise absolute values of AP for ensembles; rather, the AP values for each ensemble only need to be calibrated to reflect their relative accuracy *ranking* among all ensembles. Consequently, we will estimate the AP by comparing $D_{S|v}$ with the BBoxes generated by `REF` on v (denoted by $BBox_{\mathtt{REF}|v}$),

$$\hat{a}_{S|v} = AP\left(D_{S|v}\, , BBox_{\mathtt{REF}|v}\right). \tag{3}$$

LiDAR Model as `REF`. An example of a practical reference model is the LiDAR model, i.e., `REF`:=`LiDAR`. In applications such as autonomous driving, LiDAR has been widely employed to capture surrounding three-dimensional (3D) point clouds, producing LiDAR sweeps, as an alternative to cameras. Generally, given information such as the shooting time and LiDAR rotation angle, it is feasible to project a point in the LiDAR sweep to its corresponding image captured by cameras (for details on how this is done consult [9]). LiDAR OD models [46, 49, 50] are proposed to

[4] AP assesses not just Precision but rather evaluates the model's Precision performance across various levels of *Recall*. Mean Average Precision (mAP) is used when more than one object type is evaluated.

[5] AP is calculated as the weighted mean of precisions achieved at each threshold, where the weighting is the increase in recall from the previous threshold: AP = $\int_0^1 p(r)\, dr$, where r denotes the recall at various thresholds and $p(r)$ denotes the precision at recall equal to r.

conduct 3D object detection on LiDAR sweeps. Prior works (such as [39]) have utilized LiDAR models to label video databases for identifying errors in data labels produced by the (camera-based) object detector, via comparing the OD results of the LiDAR- and camera-based OD models. The BBoxes generated by LiDAR can be obtained by applying the LiDAR model to the corresponding LiDAR sweep of frame v, producing 3D OD BBoxes. These 3D BBoxes on the LiDAR sweep can then be converted into 2D BBoxes on frame v, denoted as $BBox_{\texttt{LiDAR}|v}$.

It has been observed [63] that LiDAR-based OD models offer a significant advantage in speed compared to camera-based OD models; i.e. if $c_{\texttt{LiDAR}|v}$ represents the inference time for a LiDAR model on a frame v, then $c_{\texttt{LiDAR}|v} \ll c_{M_i|v}$, $\forall M_i \in \mathcal{M}$.

2.4 Problem Definition

Time-Unrestricted Video Ingestion (*TUVI*). Our objective is to select a subset (ensemble) of models $S \subseteq \mathcal{M}$ for each frame $v \in \mathcal{V}$ that will produce OD results maximizing the specific choice of the scoring function. Formally, let $\mathscr{G}$ represent a selection strategy that determines which ensemble $\mathscr{G}_v \subseteq \mathcal{M}$ is selected on a frame $v \in \mathcal{V}$. Given a scoring function SC, we are interested in determining a selection strategy $\hat{\mathscr{G}}$ that identifies an ensemble for each video frame in the target video $\mathcal{V}$ by maximizing the sum of scores obtained when applying the selected ensembles on the frames,

$$\hat{\mathscr{G}} = \arg\max_{\mathscr{G}} \sum_{v \in \mathcal{V}} r_{\mathscr{G}_v|v}. \tag{4}$$

We refer to the problem described above as Time-Unrestricted Video Ingestion (*TUVI*) and will present a confidence-bound-based greedy algorithm, MES, in §3 to solve this problem.

Time-Constrained Video Ingestion (*TCVI*). In practice, when pre-processing a large number of videos for object detection and subsequent querying in a video analytics system, one may be able to invest a fixed amount of time units (up to B) for accurate OD utilizing model prediction ensembling. As a result, we also consider a variant of *TUVI*, namely Time-Constrained Video Ingestion (*TCVI*): Given a scoring function SC and a time budget B, we identify a selection strategy $\hat{\mathscr{G}}$ that maximizes the aggregate scores of all video frames processed within an allotted time constraint,

$$\hat{\mathscr{G}} = \arg\max_{\mathscr{G}} \sum_{v \in \mathcal{V}_B} r_{\mathscr{G}_v|v} \quad \text{s.t.} \sum_{v \in \mathcal{V}_B} c_{\mathscr{G}_v|v} \leq B, \tag{5}$$

where $\mathcal{V}_B$ is the sequence of frames processed exhausting the budget B under the strategy $\hat{\mathscr{G}}$. We will present a variant algorithm of MES, named MES-B, in §3 to solve this problem. In such a case, after exhausting B to process $\mathcal{V}_B$ utilizing ensembles, if users wish to continue processing the remaining frames with the same strategy $\hat{\mathscr{G}}$, they can allocate an additional budget B_{extra}. We will present a method to estimate the B_{extra} required to process the remaining video frames in §3.

***TUVI* under Concept Drift (*TUVI-CD*).** Furthermore, in practice, a streaming video (such as surveillance video) often experiences concept drifts, i.e., the underlying distribution of surrounding environments, scene types, etc. is constantly evolving. Concept drift occurs abruptly, in accordance with the definition in [28]: the distribution of scores remain constant during certain periods, and they change at unknown time instants called breakpoints, which do not depend on the ensemble selection strategy or on the sequence of frames. Let ξ represent the number of breakpoints throughout the entire video $\mathcal{V}$. We refer to the versions of *TUVI* under concept drift as *TUVI-CD*. We will provide algorithms in §3 to solve it as well.

Algorithm 1: MES

```
   Input: $\mathcal{V}$; $\mathcal{M}$; $\gamma$; $SC$;
1  Initialize $T_S$ and $\hat{\mu}_S$ $\forall$ $S \subseteq \mathcal{M}$.
2  for $t = 1, \ldots, \gamma$ do
3      Apply all ensembles on the frame $v_t$ and calculate the
         estimated scores for all ensembles, $\{\hat{r}_{S'|v_t}\}_{S' \subseteq \mathcal{M}}$.
4  for $t = \gamma+1, \ldots, |\mathcal{V}|$ do
5      for each ensemble $S' \subseteq \mathcal{M}$ do
6          $U_{S'|v_t} = \hat{\mu}_{S'|v_{t-1}} + \Gamma_{S'|v_{t-1}}$.
7      $\hat{\mathscr{G}}_{v_t} = \arg_{S' \subseteq \mathcal{M}} \max U_{S'|v_t}$.
8      Apply $\hat{\mathscr{G}}_{v_t}$ on $v_t$ and collect the est. reward $\hat{r}_{\hat{\mathscr{G}}_{v_t}|v_t}$.
9      for each subset $S_{sub} \subset \hat{\mathscr{G}}_{v_t}$ do
10         Apply $S_{\text{sub}}$ on $v_t$ and collect the est. reward
             $\hat{r}_{S_{\text{sub}}|v_t}$.
```

3 METHODOLOGY

In this section, we first present an approach for quantifying the accuracy of OD results on a frame in the absence of ground truth. For the *TUVI* problem, we present MES that can effectively select ensembles for each video frame in an iterative manner, serving as an OD pre-processor for video analysis systems. In addition, we propose MES-B, a variant of MES, that handles the *TCVI* problem in which ensemble-based processing and annotation of the video take place subject to a limited time budget, as well as an improved selection algorithm, SW-MES, which adapts to concept drifts naturally.

3.1 MES

To address the *TUVI* problem, we propose an algorithm called MES, which selects an ensemble for each frame and processes the frames iteratively, maximizing the sum of scores computed by a scoring function SC achieved under the selection. The basic idea is that MES will select the ensembles for future frames based on the accuracy and inference time of the ensemble selection for frames processed in the past. Specifically, MES adapts optimism under uncertainty [28, 55]. The entire procedure of MES is presented in Algorithm 1. It accepts as input the sequence of frames $\mathcal{V}$, the set of models $\mathcal{M}$, the number of the initial frames γ, and a scoring function SC.

3.1.1 Initialization. Throughout the execution of the algorithm, we materialize two placeholders to record the performance of each ensemble $S \subseteq \mathcal{M}$ over all the frames that have been processed in previous iterations: 1) the number of times an ensemble has been utilized for inference in previous iterations, denoted by T_S, and 2) the mean value of the estimated score obtained in previous iterations in which the ensemble was utilized, denoted by $\hat{\mu}_S$. Their values after processing the frame v_t at time t are denoted as $T_{S|v_t}$ and $\hat{\mu}_{S|v_t}$.

At the beginning, as there is no information on which ensemble performs the best, MES will conduct the initialization for all the ensembles, evaluating them on the initial frames, estimating their accuracy, and updating T_S and $\hat{\mu}_S$. In Lines 2-3 of Alg. 1, MES conducts the initialization for each ensemble $S \subseteq \mathcal{M}$. We detail the processing of one of the ensembles S' as an example. Given a hyper-parameter γ, MES first applies S' onto the first γ frames of the video, i.e. $\{v_t\}_{t\in[\gamma]}$, conducts object detection, and obtains the OD results $\{D_{S'|v_t}\}_{t\in[\gamma]}$. Then, the estimated AP of S' on the initial frames, $\{\hat{a}_{S'|v_t}\}_{t\in[\gamma]}$, is calculated using the approach indicated as in §2.3. For each frame $v\in\{v_t\}_{t\in[\gamma]}$, combining the estimated AP $\hat{a}_{S'|v}$ and the inference time $c_{S'|v}$ obtained with Equation (1), the estimated score $\hat{r}_{S'|v}$ for S' on v is calculated using the scoring function SC,

$$\hat{r}_{S'|v} = SC(\hat{a}_{S'|v}, c_{S'|v}).$$

For each ensemble $S \subseteq \mathcal{M}$, MES conducts the above process for initialization purposes. After the initialization, for each ensemble $S \subseteq \mathcal{M}$, the two placeholders, T_S and $\hat{\mu}_S$, are updated as follows,

$$T_S = T_{S|v_\gamma} = \gamma; \quad \hat{\mu}_S = \hat{\mu}_{S|v_\gamma} = \frac{\sum_{s=1}^{\gamma} \hat{r}_{S|v_s}}{T_{S|v_\gamma}}. \tag{6}$$

3.1.2 Iterative Exploration-Exploitation Approach. Next, MES continues processing the frames iteratively, utilizing an exploration and exploitation approach [55]. Exploration, refers to the process of selecting ensembles that have hardly been selected in previous iterations, while exploitation refers to selecting ensembles with higher estimated scores on previously processed frames (i.e. $\hat{\mu}_S$). To determine which ensembles need exploration, we compute an exploration bonus for each ensemble, which will be large if an ensemble has not been selected often. Let $\Gamma_{S|v_t}$ represent the exploration bonus for ensemble S after processing v_t,

$$\Gamma_{S|v_t} = \sqrt{2\ln t \;/\; T_{S|v_t}}.$$

For the t-th iteration, as shown in Lines 5-6 of Alg. 1, MES computes the Upper Confidence Bound (UCB) [55] of the estimated score $\hat{\mu}_{S|v_{t-1}}$ for each ensemble $S \subseteq \mathcal{M}$, denoted as $U_{S|v_t}$, by summing up the estimated score and the exploration bonus,

$$U_{S|v_t} = \hat{\mu}_{S|v_{t-1}} + \Gamma_{S|v_{t-1}}. \tag{7}$$

If the UCB $U_{S|v_t}$ of an ensemble S is larger than that of others, it indicates: either (1) $\hat{\mu}_{S|v_{t-1}}$ is larger, which means the ensemble got higher scores in previous iterations and is more likely to get a higher score in the future, or (2) $\Gamma_{S|v_{t-1}}$ is larger, which means the ensemble has not been selected and explored much, or a combination of (1) and (2).

In Line 7 of Alg. 1, MES then selects the ensemble with the highest UCB, denoted as $\hat{\mathscr{G}}_{v_t}$. During the early stages of the iteration procedure, MES primarily concentrates on exploring the ensembles that have been selected the least. As time progresses, the exploration bonus term decreases (since as n goes to infinity, $\frac{\log n}{n}$ goes to zero), making the algorithm shift its attention to exploitation and tends to select ensembles based mainly on the estimated score $\hat{\mu}_{S|v_{t-1}}$.

3.1.3 Placeholder Update. In Line 8 of Alg. 1, MES applies the selected ensemble $\hat{\mathscr{G}}_{v_t}$ onto frame v_t, conducts object detection, and obtains the OD results $D_{\hat{\mathscr{G}}_{v_t}|v_t}$. Then, MES calculates the estimated score $\hat{r}_{\hat{\mathscr{G}}_{v_t}|v_t}$ using the scoring function SC by combining the estimated AP $\hat{a}_{\hat{\mathscr{G}}_{v_t}|v_t}$ obtained using the approach indicated in §2.3 and the inference time $c_{\hat{\mathscr{G}}_{v_t}|v_t}$ obtained via Equation (1). Then, the values of $T_{\hat{\mathscr{G}}_{v_t}}$ and $\hat{\mu}_{\hat{\mathscr{G}}_{v_t}}$ for the selected ensemble are updated as follows,

$$\begin{aligned} T_{\hat{\mathscr{G}}_{v_t}|v_t} &= T_{\hat{\mathscr{G}}_{v_t}|v_{t-1}} + 1, \\ \hat{\mu}_{\hat{\mathscr{G}}_{v_t}|v_t} &= \frac{\hat{\mu}_{\hat{\mathscr{G}}_{v_t}|v_{t-1}} T_{\hat{\mathscr{G}}_{v_t}|v_{t-1}} + \hat{a}_{\hat{\mathscr{G}}_{v_t}|v_t}}{T_{\hat{\mathscr{G}}_{v_t}|v_t}}. \end{aligned} \tag{8}$$

Particularly, the inference time $c_{\hat{\mathscr{G}}_{v_t}|v_t}$ of applying ensemble $\hat{\mathscr{G}}_{v_t}$ on frame v_t is calculated utilizing Equation (1),

$$c_{\hat{\mathscr{G}}_{v_t}|v_t} = \sum_{M_i \in \hat{\mathscr{G}}_{v_t}} c_{M_i|v_t} + c^e_{\hat{\mathscr{G}}_{v_t}|v_t} \quad ,$$

where the inference time for each model in ensemble $\hat{\mathscr{G}}_{v_t}$ is far more than the inference time for computing the ensemble (i.e. ensembling the OD results produced by the models in $\hat{\mathscr{G}}_{v_t}$ utilizing the ensembling approaches),

$$c_{M_i|v_t} \gg c^e_{\hat{\mathscr{G}}_{v_t}|v_t} \quad \forall\, M_i \in \hat{\mathscr{G}}_{v_t}.$$

Similarly, for the ensembles corresponding to all subsets of ensemble $\hat{\mathscr{G}}_{v_t}$, the inference time for calculating ensemble is much lower than the inference time for each model,

$$c_{M_i|v_t} \gg c^e_{S_{\text{sub}}|v_t} \quad \forall\, M_i \in \hat{\mathscr{G}}_{v_t},\ \forall\, S_{\text{sub}} \subset \hat{\mathscr{G}}_{v_t}.$$

Consequently, at the end of each iteration, as shown in Lines 9-10 of Alg. 1, it applies each of the ensembles corresponding to all subsets of $\hat{\mathscr{G}}_{v_t}$ to the current frame v_t. Since the OD results of every single model of $\hat{\mathscr{G}}_{v_t}$ on frame v_t are materialized during Line 8, they can be *reused* for calculating the ensembles of all subsets of $\hat{\mathscr{G}}_{v_t}$. Thus, the extra time spent is only $\sum_{S_{\text{sub}} \subset \hat{\mathscr{G}}_{v_t}} c^e_{S_{\text{sub}}|v_t}$, which is $\ll \sum_{M_i \in \hat{\mathscr{G}}_{v_t}} c_{M_i|v_t}$. Then, the values of $T_{S_{\text{sub}}}$ and $\hat{\mu}_{S_{\text{sub}}}$ for each ensemble $S_{\text{sub}} \subset \hat{\mathscr{G}}_{v_t}$ are updated as follows,

$$\begin{aligned} T_{S_{\text{sub}}|v_t} &= T_{S_{\text{sub}}|v_{t-1}} + \mathbb{1}[S_{\text{sub}} \subset \hat{\mathscr{G}}_{v_t}], \\ \hat{\mu}_{S_{\text{sub}}|v_t} &= \frac{\hat{\mu}_{S_{\text{sub}}|v_{t-1}} T_{S_{\text{sub}}|v_{t-1}} + \mathbb{1}[S_{\text{sub}} \subset \hat{\mathscr{G}}_{v_t}] \cdot \hat{a}_{\hat{\mathscr{G}}_{v_t}|v_t}}{T_{S_{\text{sub}}|v_t}}, \end{aligned} \tag{9}$$

where $\mathbb{1}[\cdot]$ is an indicator function that takes the value of 1 when the event is true and 0 otherwise.

Combining the updates of T_S and $\hat{\mu}_S$ given by Equations (8) and (9), we present the following formula,

$$\begin{aligned} T_{S|v_t} &= \sum_{s=1}^{t} \mathbb{1}[S \subseteq \hat{\mathscr{G}}_{v_s}], && \forall S \subseteq \mathcal{M} \\ \hat{\mu}_{S|v_t} &= \frac{\sum_{s=1}^{t} \mathbb{1}[S \subseteq \hat{\mathscr{G}}_{v_s}]\, \hat{r}_{S|v_s}}{T_{S|v_t}}, && \forall S \subseteq \mathcal{M} \end{aligned} \tag{10}$$

where we assume that the ensemble selected during initialization (i.e., the first γ iterations) is always $\mathcal{M}$, i.e. $\{\hat{\mathscr{G}}_{v_t} = \mathcal{M}\}_{t\in[\gamma]}$.

3.1.4 Discussion. The algorithm we developed does not require any specific prerequisites for the detectors within $\mathcal{M}$. It effectively accommodates a variety of detector models with diverse structures, types, or configurations. In addition, the hyper-parameter γ dictates the number of frames to be iterated during the initialization process. Choosing a γ value that is too large can result in a loss of efficiency. However, if the γ value is too small, it may lead to practical issues such as no object appearing within the first γ frames, resulting in inaccurate estimation of AP during the initialization phase. This can cause the selection strategy in MES to deviate from the optimal choice and ultimately result in inaccurate results. The SW-MES algorithm, which will be discussed in §3.3, addresses this issue by implementing a forgetting strategy.

```
Algorithm 2: MES-B
   Input: $\mathcal{V}$; $\mathcal{M}$; $\gamma$; $SC$; $\boldsymbol{B}$;
 1 $\boldsymbol{C = 0}$.
 2 for $t = 1, ..., \gamma$ do
 3     Apply all ensembles on the frame $v_t$ and calculate the
         estimated scores for all ensembles, $\{\hat{r}_{S'|v_t}\}_{S' \subseteq \mathcal{M}}$
 4     $\boldsymbol{C = C + \sum_{M' \in \mathcal{M}} c_{M'|v_t} + \sum_{S' \subseteq \mathcal{M}} c^e_{S'|v_t}}$.
 5 $\boldsymbol{t = \gamma}$.
 6 while $\boldsymbol{C \leq B}$ do
 7     $\boldsymbol{t = t + 1}$.
 8     for each ensemble $S' \subseteq \mathcal{M}$ do
 9         $U_{S'|v_t} = \hat{\mu}_{S'|v_{t-1}} + \Gamma_{S'|v_{t-1}}$.
10     $\hat{\mathcal{G}}_{v_t} = \arg_{S' \subseteq \mathcal{M}} \max U_{S'|v_t}$.
11     Apply $\hat{\mathcal{G}}_{v_t}$ on $v_t$ and collect the est. reward $\hat{r}_{\hat{\mathcal{G}}_{v_t}|v_t}$.
12     for each subset $S_{sub} \subseteq \hat{\mathcal{G}}_{v_t}$ do
13         Apply $S_{\text{sub}}$ on $v_t$ and collect the est. reward
             $\hat{r}_{S_{\text{sub}}|v_t}$.
14     $\boldsymbol{C = C + \sum_{M' \in \hat{\mathcal{G}}_{v_t}} c_{M'|v_t} + \sum_{S_{\text{sub}} \subseteq \hat{\mathcal{G}}_{v_t}} c^e_{S_{\text{sub}}|v_t}}$.
```

3.2 MES-B

For the *TCVI* problem where we can invest up to B time units for conducting more accurate OD utilizing ensembles, we modify MES and propose a new algorithm MES-B as presented in Alg. 2, where the modification is highlighted with **bold** text. This section focuses mostly on the adaptation of MES to the *TCVI* problem.

Unlike MES (Alg. 1), MES-B's input adds a new parameter, the time budget B. Due to the limited budget, it may only be able to perform ensemble selection on some of the video frames before the time budget is exhausted. It records and updates the elapsed time consumption, denoted as C, on each iteration. MES-B first initializes C to a scalar variable with a value of 0 at Line 1 of Alg. 2. In Line 4 of Alg. 2, since MES-B needs to explore all ensembles, according to the definition in Equation (1), for each of the initialization frames $v_t \in \{v_s\}_{s \in [\gamma]}$, the time that MES-B needs to spend, C_{v_t}, is calculated as

$$\begin{aligned} C_{v_t} &= \sum_{S' \subseteq \mathcal{M}} c_{S'|v_t} = \sum_{S' \subseteq \mathcal{M}} \left(\sum_{M' \in S'} c_{M'|v_t} + c^e_{S'|v_t} \right) \\ &= \left(2^{m-1} - 1 \right) \cdot \sum_{M' \in \mathcal{M}} c_{M'|v_t} + \sum_{S' \subseteq \mathcal{M}} c^e_{S'|v_t}. \end{aligned} \tag{11}$$

Since the object detection results of each single model can be *reused* and thus every single model needs only one inference. The time consumed shown in Equation (11) can be optimized as follows,

$$C_{v_t} = \sum\nolimits_{M' \in \mathcal{M}} c_{M'|v_t} + \sum\nolimits_{S' \subseteq \mathcal{M}} c^e_{S'|v_t}. \tag{12}$$

Similarly, in Line 14, MES-B updates C, adding to it the time consumed in this iteration (processing v_t), C_{v_t}, which is the sum of the time consumed for applying the ensembles in this iteration,

$$\begin{aligned} C_{v_t} &= \sum_{S_{\text{sub}} \subseteq \hat{\mathcal{G}}_{v_t}} c_{S_{\text{sub}}|v_t} = \sum_{S_{\text{sub}} \subseteq \hat{\mathcal{G}}_{v_t}} \left(\sum_{M' \in S_{\text{sub}}} c_{M'|v_t} + c^e_{S_{\text{sub}}|v_t} \right) \\ &= \left(2^{|\hat{\mathcal{G}}_{v_t}|-1} - 1 \right) \sum_{M' \in \hat{\mathcal{G}}_{v_t}} c_{M'|v_t} + \sum_{S_{\text{sub}} \subseteq \hat{\mathcal{G}}_{v_t}} c^e_{S_{\text{sub}}|v_t} \quad , \end{aligned} \tag{13}$$

where $|\hat{\mathcal{G}}_{v_t}|$ represents the number of models in $\hat{\mathcal{G}}_{v_t}$. Similarly, the OD results of each model can be reused, and the time consumed during this iteration shown in Equation (13) can be optimized as

$$C_{v_t} = \sum\nolimits_{M' \in \hat{\mathcal{G}}_{v_t}} c_{M'|v_t} + \sum\nolimits_{S_{\text{sub}} \subseteq \hat{\mathcal{G}}_{v_t}} c^e_{S_{\text{sub}}|v_t}. \tag{14}$$

The iteration of MES-B will terminate when the budget is exhausted, i.e., $C \leq B$, as demonstrated in Line 6 of Alg. 2.

Let $\mathcal{V}_B$ represent the frames processed with MES-B exhausting the time budget B. While some approaches [16, 41] can increase processing throughput by skipping frames based on the similarity of adjacent frames in videos, these methods are orthogonal to our work and will not be elaborated upon in this paper. In the case where any frames remain unlabelled, i.e., $\mathcal{V} \backslash \mathcal{V}_B \neq \emptyset$, there are a couple of possible remedial approaches. For example, they can be processed 1) with a lightweight detector, such as the lightest model in $\mathcal{M}$, or 2) using an extra budget B_{extra} to select ensembles from $\mathcal{M}$ under the same strategy $\hat{\mathcal{G}}$. For the second approach, we provide a linear regression-based method, named LRBP, for estimating the extra budget B_{extra} required to process the remaining video frames $\mathcal{V} \backslash \mathcal{V}_B$ under the same selection strategy:

1. During processing the frames in $\mathcal{V}_B$, obtain the pairs of processed frames and the budget consumed after each iteration, i.e. $\left\{ \langle t, \sum_{s=1}^{t} C_{v_s} \rangle \right\}_{t \in [|\mathcal{V}|_B]}$;
2. Fit the pairs using linear regression and predict the budget $\hat{B}_{\text{extra}}$ required to process $|\mathcal{V}|$ frames.

We will evaluate the accuracy of LRBP for predicting the extra budget B_{extra} in §5.

3.3 SW-MES

For the problem *TUVI-CD*, previous work [32] and our empirical evidence demonstrate that MES is not appropriate for abruptly changing environments. To address this issue, we propose an improved selection algorithm, called SW-MES, that adopts a sliding-window technique, in which the past scores utilized to affect the present selection decision, are derived only from a fixed-length time window, rather than from all previously processed frames.

This section focuses mainly on how SW-MES differs from MES. SW-MES introduces a new hyper-parameter λ that indicates the size of the time window. After initializing SW-MES in the same manner as MES, at the t-th iteration, instead of averaging the scores over all the frames that have been processed as per Equation (10), SW-MES relies on a local empirical average of the observed scores derived from only recent λ preceding frames. Specifically, for each $S \subseteq \mathcal{M}$, SW-MES updates the local values of T_S and $\hat{\mu}_S$ after each iteration in the following way,

$$\begin{aligned} T^{\lambda}_{S|v_t} &= \sum_{s=t-\lambda+1}^{t} \mathbb{1}[S \subseteq \hat{\mathcal{G}}_{v_s}], \\ \hat{\mu}^{\lambda}_{S|v_t} &= \frac{\sum_{s=t-\lambda+1}^{t} \mathbb{1}[S \subseteq \hat{\mathcal{G}}_{v_s}] \ \hat{r}_{S|v_s}}{T^{\lambda}_{S|v_t}}, \end{aligned} \tag{15}$$

and constructs a UCB $U_{S'|v_t}$ for the estimated score through replacing Line 6 in Alg. 1 by the following,

$$U_{S'|v_t} = \hat{\mu}^{\lambda}_{S'|v_{t-1}} + \Gamma^{\lambda}_{S'|v_{t-1}} \quad , \tag{16}$$

where the exploration bonus $\Gamma^{\lambda}_{S'|v_{t-1}}$ is calculated as

$$\Gamma^{\lambda}_{S'|v_{t-1}} = \sqrt{2 \frac{\ln \min(t-1, \lambda)}{T^{\lambda}_{S'|v_{t-1}}}}.$$

The window size λ should capture the necessary temporal dynamics without introducing excessive noise. It can be determined based on expert knowledge of the video or grid search [5, 6] on a validation set. We will evaluate the outcomes of SW-MES and compare it with MES in §5.

4 ANALYSIS

In this section, we analyze the effectiveness of the algorithms presented in §3.

To evaluate the effectiveness of MES, MES-B and SW-MES, we define the *regret* that quantifies the amount of score they lose due to not selecting an optimal ensemble at each iteration, denoted by $R_{\texttt{MES}}$, $R_{\texttt{MES-B}}$ and $R_{\texttt{SW-MES}}$, respectively,

$$\begin{aligned} R_{\texttt{MES}} = R_{\texttt{SW-MES}} &= \sum_{t=1}^{|\mathcal{V}|} \left(r_{S^*_{v_t}|v_t} - r_{\hat{\mathscr{G}}_{v_t}|v_t} \right), \\ R_{\texttt{MES-B}} &= \sum_{t=1}^{|\mathcal{V}_B|} \left(r_{S^*_{v_t}|v_t} - r_{\hat{\mathscr{G}}_{v_t}|v_t} \right), \end{aligned} \tag{17}$$

where $\hat{\mathscr{G}}_{v_t}$ represents the ensemble selected by each of them for frame v_t; $S^*_{v_t}$ represents the optimal ensemble for frame v_t in the sense of achieving the highest score on v_t,

$$S^*_{v_t} = \arg\max_{S' \subseteq \mathcal{M}} r_{S'|v_t}.$$

Clearly, a smaller value of $R_{\texttt{MES}}$/$R_{\texttt{MES-B}}$/$R_{\texttt{SW-MES}}$ indicates a more effective MES/MES-B/SW-MES algorithm.

Analysis of MES. According to the definition of *TUVI*, we know that for MES, the underlying distribution of surrounding environments, scene types, etc. remains *consistent* throughout the entire video. Thus, to facilitate the forthcoming proof, we make the following reasonable assumption for MES:

Assumption 1. *The scores $\{r_{S|v}\}_{v\in\mathcal{V}}$ for each ensemble $S \subseteq \mathcal{M}$ over the video are identical random variables from a stationary distribution that are unknown and potentially different for each ensemble.*

Theorem 4.1. *For problem TUVI, given $\mathcal{V}$ and $\mathcal{M}$ satisfying Assumption 1, the expected regret of* MES *has the following upper bound,*

$$E\left[R_{\texttt{MES}}\right] \le O(|\mathcal{M}| \log |\mathcal{V}|). \tag{18}$$

Proof sketch. We adapt the proof for algorithm UCB and several findings in [1], omitting some intermediate steps.

We first introduce some notations for the proof. For simplicity, we use n to represent $|\mathcal{V}|$. Let $N_{S|v_t}$ represent the number of times that an ensemble S has been selected before the t-th iteration[6],

$$N_{S|v_t} = \sum_{s=1}^{t} \mathbb{1}\left[\hat{\mathscr{G}}_{v_s} = S\right]. \tag{19}$$

Let μ^* represent the average score of the optimal ensemble on frames over the entire video, $\mu^* = \frac{1}{n}\sum_{v\in\mathcal{V}} r_{S^*_v|v}$; let μ_S represent the mean value of the actual score of an ensemble S on the frames on which the ensemble is inferred, $\mu_S = \frac{\sum_{s=1}^{n} \mathbb{1}[S \subseteq \hat{\mathscr{G}}_{v_S}]\, r_{S|v_S}}{T_{S|v_n}}$. We have

$$E\left[R_{\texttt{MES}}\right] = n \cdot \mu^* - \sum_{S\subseteq\mathcal{M}} E\left[N_{S|v_n}\right] \cdot \mu_S \quad . \tag{20}$$

We bound $N_{S|v_n}$ for each ensemble $S \subseteq \mathcal{M}$ as follows. Let $\Gamma_x^y = \sqrt{2 \ln y/x}$ and ℓ be an arbitrary positive integer.

$$\begin{aligned} N_{S|v_n} &\le \ell + \sum_{t=1}^{n} \mathbb{1}\left[\hat{\mu}_{S^*_{v_t}|v_{t-1}} + \Gamma_{S^*_{v_t}|v_{t-1}} \le \hat{\mu}_{S|v_{t-1}} + \Gamma_{S|v_{t-1}}, N_{S|v_{t-1}} \ge \ell\right] \\ &\le \ell + \sum_{t=1}^{n} \mathbb{1}\left[\mu_{S^*_{v_t}|v_{t-1}} + \Gamma_{S^*_{v_t}|v_{t-1}} \le \mu_{S|v_{t-1}} + \Gamma_{S|v_{t-1}}, T_{S|v_{t-1}} \ge \ell\right] \\ &\le \ell + \sum_{t=1}^{n}\sum_{s=1}^{t-1}\sum_{s'=\ell}^{t-1} \mathbb{1}\left[\mu_{S^*_{v_s}|v_s} + \Gamma_s^t \le \mu_{S|v_{s'}} + \Gamma_{s'}^t\right]. \end{aligned} \tag{21a}$$

If the event in (21a), i.e., $\mu_{S^*_{v_s}|v_s} + \Gamma_s^t \le \mu_{S|v_{s'}} + \Gamma_{s'}^t$, is true, then at least one of the following three conditions must be true,

$$Evt_1 : \mu_{S^*_{v_s}|v_s} + \Gamma_s^t \le \mu^*; \tag{22a}$$

$$Evt_2 : \mu^* < \mu_S + 2\Gamma_{s'}^t; \tag{22b}$$

$$Evt_3 : \mu_{S|v_{s'}} - \mu_S \ge \Gamma_{s'}^t. \tag{22c}$$

According to Chernoff-Hoeffding Inequality [35], the probability that Inequality (22a) holds is $\Pr\left[Evt_1\right] \le t^{-4}$; similarly, the probability that Inequality (22c) holds is $\Pr\left[Evt_3\right] \le t^{-4}$. For Inequality (22b), when $\ell = \lceil \frac{8\ln n}{(\mu^*-\mu_S)^2} \rceil$, (22b) fails, $\mu^* - \mu_S - 2\Gamma_{s'}^t \ge 0$. Consequently, $\Pr\left[\mu_{S^*_{v_s}|v_s} + \Gamma_s^t > \mu_{S|v_{s'}} + \Gamma_{s'}^t\right] \le 2t^{-4}$. In the end,

$$E\left[R_{\texttt{MES}}\right] \le \sum_{S:\mu_S<\mu^*} \frac{8\ln n}{\mu^* - \mu_S} + \left(1 + \frac{\pi}{3}\right) \sum_{S'\subseteq\mathcal{M}} (\mu^* - \mu_{S'}). \tag{23}$$

Thus, $E\left[R_{\texttt{MES}}\right] \le O(|\mathcal{M}| \log n)$. □

Analyses of MES-B. To show the upper bound of $R_{\texttt{MES-B}}$, we adapt the proof in [21], omitting some intermediate steps. For simplicity, we use $n(B)$ and m to represent $|\mathcal{V}_B|$ and $|\mathcal{M}|$. Let $\bar{c}_S$ represent the average cost of the ensemble S, $\bar{c}_S = \frac{1}{n}\sum_{t=1}^{n} c_{S|v_t}$; let $S^\dagger$ represent the ensemble with the highest average score, $S^\dagger = \arg\max_{S\subseteq\mathcal{M}} \frac{1}{n}\sum_{t=1}^{n} r_{S|v_t}$.

[21] shows the following lemma, which characterizes $n(B)$ of the proposed algorithm MES-B.

Lemma 4.2. *For problem TCVI, if $\forall S \subseteq \mathcal{M}, \exists \delta_S > 0, \rho_S > 0$, s.t., $E\left[N_{S|v_{n(B)}}\right] \le \delta_S \ln n(B) + \rho_S$, then we have*

$$\begin{aligned} E[n(B)] &\le \frac{B+1}{\bar{c}^\dagger} + \delta \ln 2\left(\frac{B+1}{\bar{c}^\dagger} + \delta \ln 2\delta + \rho\right) + \rho; \\ E[n(B)] &> \frac{B-\rho'}{\bar{c}^\dagger} - \frac{\delta'}{\bar{c}^\dagger} \ln 2\left(\frac{B+1}{\bar{c}^\dagger} + \delta\ln 2\delta + \rho\right) - 1. \end{aligned} \tag{24}$$

where $\delta = \sum_{S\neq S^\dagger} \delta_S$, $\rho = \sum_{S\neq S^\dagger} \rho_S$, $\delta' = \sum_{S\neq S^\dagger} \bar{c}_S\delta_S$, and $\rho' = \sum_{S\neq S^\dagger} \bar{c}_S\rho_S$.

Theorem 4.3. *For problem TCVI, given $\mathcal{V}$, $\mathcal{M}$ and B satisfying Assumption 1, the expected regret of* MES-B *has the following upper bound,*

$$E\left[R_{\texttt{MES-B}}\right] \le O(|\mathcal{M}| \log B). \tag{25}$$

Proof sketch. Based on Theorem 4.1 and Lemma 4.2, for *TCVI*, we can similarly derive

$$\begin{aligned} E\left[R_{\texttt{MES-B}}\right] &= n(B)\cdot\mu^* - \sum_{S\subseteq\mathcal{M}} E\left[N_{S|v_{n(B)}}\right]\cdot\mu_S \\ &\le \sum_{S:\mu_S<\mu^*} \frac{8\ln E[n(B)]}{\mu^*-\mu_S} + \left(1+\frac{\pi}{3}\right)\sum_{S'\subseteq\mathcal{M}}(\mu^*-\mu_{S'}) \\ &\le \alpha \ln\left(\frac{B+1}{\bar{c}^*} + \delta\ln 2\left(\frac{B+1}{\bar{c}^*} + \delta\ln 2\delta + \rho\right) + \rho\right) + \beta, \end{aligned} \tag{26}$$

[6] The difference between $N_{S|v}$ and $T_{S|v}$ is that the former counts the number of times S (i.e. $S = \hat{\mathscr{G}}_v$) is selected and the latter counts the number of times a superset of S (i.e. $S \subseteq \hat{\mathscr{G}}_v$) is selected.

Table 1: Information of Dataset nuScenes.

Group	# of Scenes	# of Samples	Duration (min)
nuScenes	850	42,500	354
nusc-clear	274	13,700	114
nusc-night	79	3,950	33
nusc-rainy	184	9,200	77

Table 2: Information of Dataset BDD.

Group	# of Sequences	# of Samples	Duration (min)
BDD	300	30,000	200
bdd-rainy	120	5,070	80
bdd-snow	132	5,549	90

Table 3: Information of OD model structures.

Structures	# of Params	Avg. Inference Time (ms)
YOLOv7	37.2M	49.5
YOLOv7-tiny	6.03M	10.0
YOLOv7-micro	2.68M	7.7
Faster R-CNN	42.1M	212

where $\alpha = \sum_{S:\mu_S<\mu^*} \delta_S(\mu^* - \mu_S)$, $\beta = \sum_{S' \subseteq \mathcal{M}} \rho_S(\mu^* - \mu_S)$.

Thus, $E\left[R_{\texttt{MES-B}}\right] \le O(|\mathcal{M}| \log B)$. □

Analyses of SW-MES. According to the definition of problem *TUVI-CD*, the underlying distribution may *change* abruptly. Consequently, Assumption 1 made for the proof of Theorem 4.1 can be relaxed where the scores $\{r_{S|v}\}_{v \in \mathcal{V}}$ for each ensemble are modeled by a sequence of random variables from potentially different distributions that may vary across time.

Theorem 4.4. *For problem TUVI-CD, given $\mathcal{V}$ and $\mathcal{M}$, if an appropriate λ is chosen, the expected regret of* SW-MES *has the following upper bound,*

$$E\left[R_{\texttt{SW-MES}}\right] \le O\left(|\mathcal{M}|\sqrt{\xi \cdot |\mathcal{V}| \cdot \log |\mathcal{V}|}\right), \tag{27}$$

where ξ is the number of breakpoints throughout the entire video $\mathcal{V}$.

Proof sketch. Utilizing the regret guarantee for algorithm SW-UCB [28], we can derive an upper bound of $E\left[R_{\texttt{SW-MES}}\right]$,

$$E\left[R_{\texttt{SW-MES}}\right] \le C \sum_{S \subseteq \mathcal{M}} \left(C_S \frac{n \log \lambda}{\lambda} + \lambda \cdot \xi + \log^2 \lambda\right), \tag{28}$$

where C and $\{C_S\}_{S \subseteq \mathcal{M}}$ are all constants that do not involve n or λ.

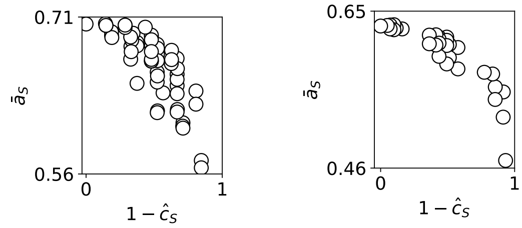
Choosing $\lambda = \sqrt{n \log n / \xi}$,

$$E\left[R_{\texttt{SW-MES}}\right] \le \ C \sum_{S \subseteq \mathcal{M}} \left(C_S\sqrt{\xi n \log n} + \sqrt{\xi n \log n} + \log^2 \sqrt{\frac{n \log n}{\xi}}\right). \tag{29}$$

Thus, $E\left[R_{\texttt{SW-MES}}\right] \le O\left(|\mathcal{M}|\sqrt{\xi \cdot n \cdot \log n}\right)$. □

5 EXPERIMENTAL EVALUATION

In this section, we present the results of our experimental evaluation utilizing real datasets varying settings and parameters of interest.



(a) $\mathcal{V}_{\textbf{nusc}}$.

(b) $\mathcal{V}_{\textbf{nusc}}^{\textbf{night}}$.

Figure 3: $\langle \bar{a}_S, 1 - \hat{c}_S \rangle$ values of all the ensembles for datasets $\mathcal{V}_{\textbf{nusc}}$ and $\mathcal{V}_{\textbf{nusc}}^{\textbf{night}}$. Each circle represents an ensemble.

5.1 Datasets

5.1.1 nuScenes [9]. nuScenes is a large-scale autonomous driving dataset consisting of 850 scenes with object annotations for 23 object classes. Each scene contains a sequence of images consecutively captured by cameras mounted on a moving vehicle, together with their corresponding LiDAR sweeps captured by LiDAR mounted on the same vehicle.

Utilizing the scene information provided by nuScenes, we group all the scenes based on the environmental circumstances (such as weather and time) at the time of capture and divide them into 3 categories: *clear*, *night*, and *rainy*, as shown in Table 1. We use $\mathcal{V}_{\text{nusc}}$ to represent the video dataset consisting of all the frames from videos in nuScenes, and $\mathcal{V}_{\text{nusc}}^{\text{clear}}$, $\mathcal{V}_{\text{nusc}}^{\text{night}}$ and $\mathcal{V}_{\text{nusc}}^{\text{rainy}}$ to represent 3 additional specialized datasets consisting of all frames from the groups of corresponding categories respectively.

In order to evaluate the effectiveness of our proposals for *TUVI-CD*, we create new video datasets, $\mathcal{V}_{\text{c\&n}}$, $\mathcal{V}_{\text{n\&r}}$ and $\mathcal{V}_{\text{c\&n\&r}}$, by segmenting the specialized datasets (each identified with a corresponding letter subscript) to 10 segments and combining the segments in a random order to introduce concept drift. For example, to create $\mathcal{V}_{\text{c\&n}}$, we divide $\mathcal{V}_{\text{nusc}}^{\text{clear}}$ and $\mathcal{V}_{\text{nusc}}^{\text{night}}$ into 10 segments, and then shuffle and combine the segments together.

5.1.2 BDD [66]. BDD is a video dataset consisting of 100,000 sequences of frames, annotated with various object types, and enriched with detailed information of each sequence such as the location (e.g., city streets or parking lots), time (e.g., daytime or nighttime), and weather conditions (e.g., snowy or rainy). We randomly select some sequences from BDD for evaluation, denoted as $\mathcal{V}_{\text{bdd}}$. Additionally, we utilize sequences with labels *rainy* or *snow* to train different specialized detectors, as shown in Table 2, which are used to form model candidates $\mathcal{M}$ (to be introduced in §5.2).

5.2 Models

Object Detectors. We utilize three model structures based on YOLOv7 [60] (YOLOv7, YOLOv7-tiny, and YOLOv7-micro in descending order of network complexity) and a structure based on Faster R-CNN [52] for training object detection (OD) models. The structures of YOLOv7-tiny are built based on the description in [60]. The novel structure of YOLOv7-micro is derived from YOLOv7-tiny by reducing the number of neural network layers and prediction grids[7]. Generally, as shown in Table 3, under the same training settings, the order of detection accuracy and inference time are YOLOv7 > YOLOv7-tiny > YOLOv7-micro > Faster

[7] The shallower structures of complicated models have been used in many works: for example, [40, 41, 44, 65] refer to them as proxy models or filters for preliminary filtering of frames that satisfy the specified requirements.

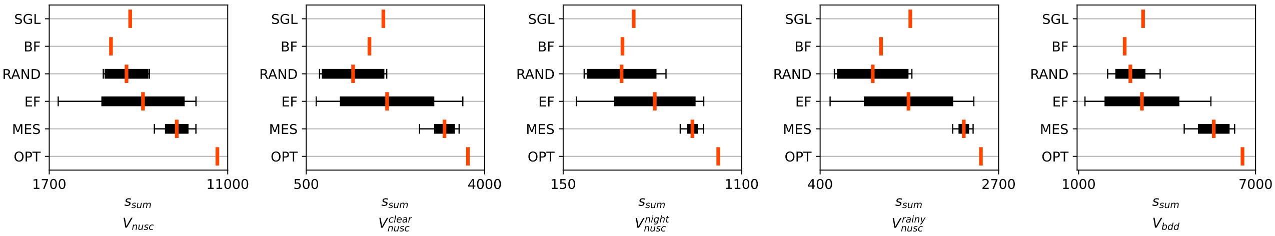
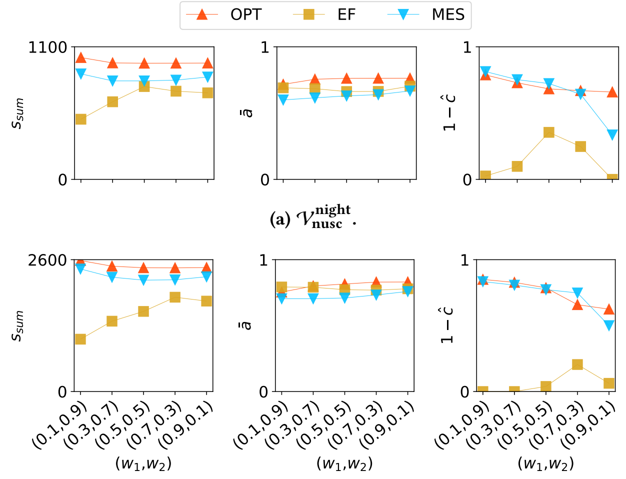


Figure 4: Scores (s_{sum}) obtained by varying algorithms for *TUVI* on datasets $\mathcal{V}_{\text{nusc}}$, $\mathcal{V}_{\text{nusc}}^{\text{clear}}$, $\mathcal{V}_{\text{nusc}}^{\text{night}}$, $\mathcal{V}_{\text{nusc}}^{\text{rainy}}$ and $\mathcal{V}_{\text{bdd}}$. The red markers represent the mean values of s_{sum} for each algorithm; the black rectangles represent standard deviations of the results under 100 independent trials; the extended lines represent extreme values (minimum/maximum) under 100 trials.



(a) $\mathcal{V}_{\text{nusc}}^{\text{night}}$.

(b) $\mathcal{V}_{\text{nusc}}^{\text{rainy}}$.

Figure 5: s_{sum}, $\bar{a}$ and $1-\hat{c}$ under varying weight combinations of the scoring function for *TUVI* on datasets $\mathcal{V}_{\text{nusc}}^{\text{night}}$ and $\mathcal{V}_{\text{nusc}}^{\text{rainy}}$.

R-CNN. Before conducting our experiments, we train models using these four structures on the datasets specified in Tables 1&2, aiming to develop as many pre-trained object detection models as possible. For each dataset, we use a proper set of relevant pre-trained object detectors to form the pool of their corresponding model candidates $\mathcal{M}$.

Ensemble Approaches. We attempt several ensemble approaches (including NMS [29, 52], Softer-NMS [34], WBF [56], NMW [67] and Fusion [61]) for the pre-trained OD models and select WBF, which produces the most accurate OD outputs, as the ensemble approach for the subsequent experiments.

Figure 3 presents the $\langle \bar{a}_S, 1-\hat{c}_S \rangle$ values of all the ensembles $\subseteq \mathcal{M}$ for datasets $\mathcal{V}_{\text{nusc}}$ and $\mathcal{V}_{\text{nusc}}^{\text{night}}$, where $\bar{a}_S$ represents the average AP of ensemble S over the video; $\hat{c}_S$ represents the normalized average inference time of ensemble S over the video. Similar trends are observed on the other datasets.

Reference Models. We utilize a pre-trained state-of-the-art LiDAR model, named MEGVII [68], for quantifying the accuracy of OD results in the absence of ground truth (i.e. `REF`:=`LiDAR`).

5.3 Approaches Compared

We compare through experiments the proposed algorithms, `MES`, `MES-B`, and `SW-MES`, with the following algorithms.

Table 4: Prediction of B_{extra} utilizing `LRBP`.

Datasets	$\lvert\mathcal{V}\rvert$	B	$\lvert\mathcal{V}_B\rvert$	B_{lrbp}	B_{extra}
$\mathcal{V}_{\text{nusc}}$	200,000	100	11,115	1,578	1,500
		200	23,160	1,456	1,400
		400	48,165	1,221	1,200
$\mathcal{V}_{\text{nusc}}^{\text{clear}}$	40,000	500	6,360	1,750	1,750
$\mathcal{V}_{\text{nusc}}^{\text{night}}$	18,500	30	2,745	191	170
$\mathcal{V}_{\text{nusc}}^{\text{rainy}}$	42,800	150	6,105	327	320

1. *OPT*: OPT represents an OPTimal approach in which an "oracle" selects the best ensemble based on score (which takes into account) both accuracy and inference time) for each video frame, i.e. $\{\hat{\mathcal{G}}_v^{\text{opt}}=S_v^*\}_{v\in\mathcal{V}}$, where $\hat{\mathcal{G}}_v^{\text{opt}}$ is the ensemble chosen for frame v and S_v^* is the ensemble achieving the highest score on v, $S_v^* = \arg_{S\subseteq\mathcal{M}} \max r_{S|v}$. OPT signifies the best score any algorithm can achieve, and we include it as a reference. However, it is impractical on its own, as it is impossible to know which ensemble is the best for each frame in advance.
2. *BF* and *SGL*: BF is a Brute-Force approach applying the largest ensemble consisting of all detectors[8] on each frame, $\{\hat{\mathcal{G}}_v=\mathcal{M}\}_{v\in\mathcal{V}}$; SGL always applies a specific SinGLe detector (which is the most accurate on average across all frames) to each frame.
3. *RAND*: It applies an ensemble RANDomly selected from $\mathcal{M}$ on each frame.
4. *EF*: Explore-First (EF) implements a Multi-Armed Bandit (MAB) strategy [55], which initially explores all ensembles in $\mathcal{M}$ by applying each to the first δ_{EF} frames of video $\mathcal{V}$. Subsequently, the ensemble with the highest estimated score based on these frames is selected and applied to the remaining frames.

5.4 Settings

We utilize the following scoring function in the experiments,

$$r_{S|v} = SC(a_{S|v}, c_{S|v}) = w_1\cdot \log_2\left(a_{S|v}+1\right) + w_2\cdot \log_2\left(2-\hat{c}_{S|v}\right) \tag{30}$$

where the weights w_1+w_2=1; $\hat{c}_{S|v}$ is the normalized inference time of S on v, $\hat{c}_{S|v}=\frac{c_{S|v}}{c_{\max}}$, where $c_{\max}$ is the maximum inference time among all the ensembles, $c_{\max}=\max_{\forall S'\subseteq\mathcal{M}} c_{S'|v}$. Any function that adheres to the criteria identified in §2 can be easily adopted. We utilize w_1=w_2=0.5 in what follows unless stated otherwise.

Each experiment reports the average of the results of 100 independent trials for each algorithm compared. For each trial,

[8] The largest ensemble $\mathcal{M}$ usually has the best accuracy but the longest inference time; thus, BF is not expected to achieve a high score when the time component plays a significant role in the scoring function.

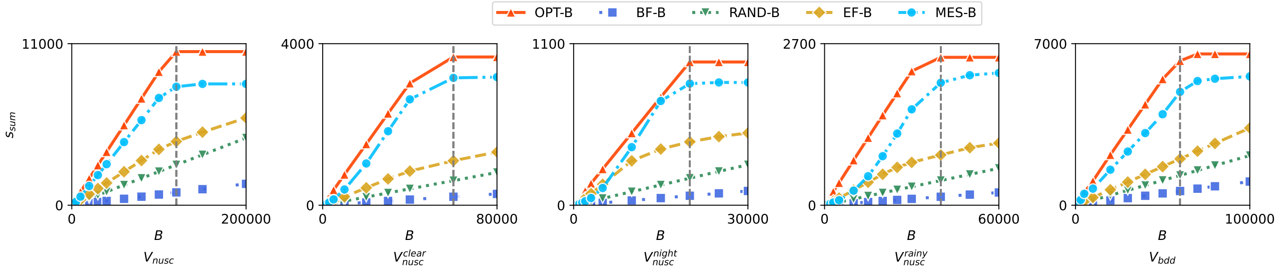


Figure 6: s_{sum}-B curves for *TCVI* on datasets $\mathcal{V}_{\text{nusc}}$, $\mathcal{V}_{\text{nusc}}^{\text{clear}}$, $\mathcal{V}_{\text{nusc}}^{\text{night}}$, $\mathcal{V}_{\text{nusc}}^{\text{rainy}}$ and $\mathcal{V}_{\text{bdd}}$.

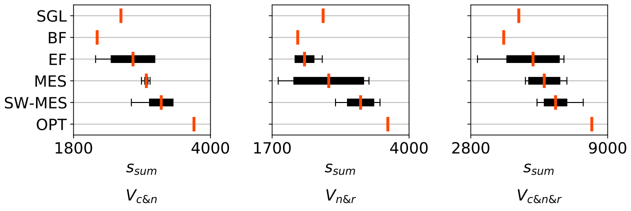


Figure 7: Scores (s_{sum}) obtained by varying algorithms for *TUVI-CD* on datasets $\mathcal{V}_{\text{c\&n}}$, $\mathcal{V}_{\text{n\&r}}$ and $\mathcal{V}_{\text{c\&n\&r}}$.

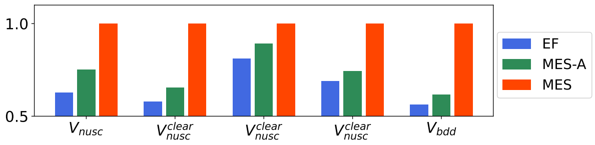
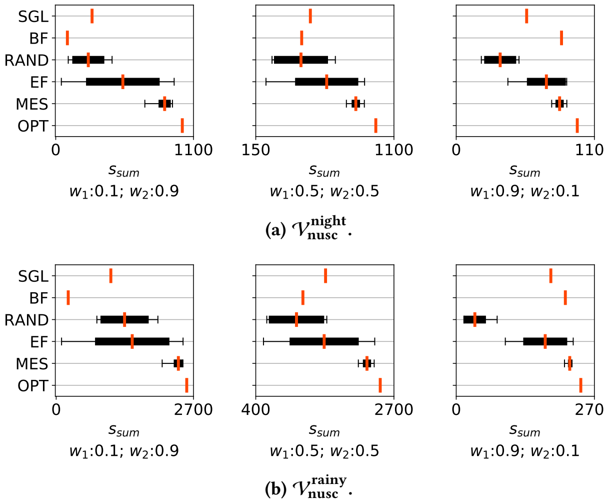


Figure 8: The sum of scores (normalized by the score of MES) of algorithms EF, MES and its ablation MES-A on all the datasets.



(a) $\mathcal{V}_{\text{nusc}}^{\text{night}}$.

(b) $\mathcal{V}_{\text{nusc}}^{\text{rainy}}$.

Figure 9: Scores obtained by the algorithms with varying weight combinations in the scoring function.

we *re-sample* the video datasets following the procedures outlined in §5.1. All algorithms were implemented in Python and run on a Linux server with Intel Xeon Gold 6244 3.60GHz CPU, 64GB memory, and an NVIDIA TITAN Xp GPU.

5.5 Measurements

We utilize the sum of scores, s_{sum}, to evaluate the results produced by applying the selection algorithms on each frame in $\mathcal{V}$. For problems *TUVI* and *TUVI-CD*,

$$s_{\text{sum}} = \sum_{t=1}^{|\mathcal{V}|} r_{\hat{\mathcal{G}}_{v_t}|v_t} \quad ,$$

and for problem *TCVI*,

$$s_{\text{sum}} = \sum_{t=1}^{|\mathcal{V}_B|} r_{\hat{\mathcal{G}}_{v_t}|v_t} \quad ,$$

where $\hat{\mathcal{G}}_{v_t}$ represents the ensemble identified by the selection algorithms for frame v_t; $r_{\hat{\mathcal{G}}_{v_t}|v_t}$ is calculated by the scoring function specified in Equation (30).

In addition, we use $\bar{a}$ (average AP) to measure the accuracy of the OD results produced by applying the selection algorithms:

$$\text{for } \mathit{TUVI}\text{: } \bar{a} = \text{avg}_{t=1}^{|\mathcal{V}|} a_{\hat{\mathcal{G}}_{v_t}|v_t}; \quad \text{for } \mathit{TCVI}\text{: } \bar{a} = \text{avg}_{t=1}^{|\mathcal{V}_B|} a_{\hat{\mathcal{G}}_{v_t}|v_t}.$$

We use $1-\hat{c}$ to measure the performance of the ensembles selected by the algorithms, where $\hat{c}$ is the average normalized inference time:

$$\text{for } \mathit{TUVI}\text{: } \hat{c} = \text{avg}_{t=1}^{|\mathcal{V}|} \hat{c}_{\hat{\mathcal{G}}_{v_t}|v_t}; \quad \text{for } \mathit{TCVI}\text{: } \hat{c} = \text{avg}_{t=1}^{|\mathcal{V}_B|} \hat{c}_{\hat{\mathcal{G}}_{v_t}|v_t}.$$

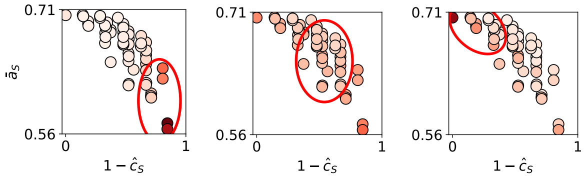
5.6 Overall Evaluation

This section evaluates the overall performance of the algorithms proposed in this paper, alongside comparative approaches, on all the three distinct problem definitions described in §2.4.

5.6.1 *Evaluation of Problem TUVI.*

Sum of Scores. Under the problem setting of *TUVI*, we run the algorithms on each specialized dataset and present Figure 4, which illustrates the sum of scores s_{sum} obtained by the algorithms. For approaches RAND, EF and MES, Figure 4 displays the mean values of s_{sum}, as well as their standard deviation and minimum/maximum results from 100 independent trials for each algorithm. Since the results for OPT, BF and SGL are the same across all trials, they are represented as single points in the figure. Figure 4 reveals MES consistently achieves scores higher than 85% of OPT's scores across all datasets and outperforms SGL, BF, RAND and EF. Furthermore, the range (between the minimum and maximum) and standard deviation of MES results are both smaller compared to EF, indicating that MES is more stable.

Average Precision and Inference Time Cost. To display the AP and Cost of each algorithm directly, we plot s_{sum}, $\bar{a}$ and $1-\hat{c}$ (defined in §5.5) under varying weight combinations of accuracy and time cost $\langle w_1, w_2 \rangle$ for the scoring function *SC* for *TUVI* on datasets $\mathcal{V}_{\text{nusc}}^{\text{night}}$ and $\mathcal{V}_{\text{nusc}}^{\text{rainy}}$ in Figure 5. We observe that MES consistently achieves a higher s_{sum} than EF for all weight combinations. In terms of $\bar{a}$ and $1-\hat{c}$, OPT and MES exhibit similar



(a) w_1: 0.1; w_2: 0.9. (b) w_1: 0.5; w_2: 0.5. (c) w_1: 0.9; w_2: 0.1.

Figure 10: A demonstration of the distribution of the number of times the ensembles are selected at various weight combinations in the scoring function for MES on $\mathcal{V}_{\text{nusc}}$. Darker color represents more selections.

trends are are closer to each other than EF: as w_1 increases and w_2 decreases, $\bar{a}$ increases and $1-\hat{c}$ decreases. Compared to EF, MES can adapt ensemble selection to various weight combinations to balance accuracy and cost. Similar trends are observed on the other datasets.

5.6.2 Evaluation of Problem TCVI.

Sum of Scores. For problem *TCVI*, Figure 6 shows the total scores obtained by each algorithm as budget B varies. MES outperforms SGL, BF and EF not only when B is sufficient (to the right of the dashed vertical lines), but also when B is small (which might provide enough budget to process the entire video dataset).

Extra Budget Prediction. To evaluate the approach for extra budget prediction LRBP (introduced in §3.2), we present Table 4. B denotes the initial budget, $|\mathcal{V}_B|$ represents the frames processed upon exhausting B, and $B_{\text{lrbp}}/B_{\text{extra}}$ indicates the predicted/actual extra budget needed to process the entire video dataset. The table shows that the errors of B_{lrbp} over B_{extra} are generally within 10% for the evaluated datasets. As B increases in the $\mathcal{V}_{\text{nusc}}$ prediction, the errors of B_{lrbp} over B_{extra} decrease, suggesting improved prediction accuracy.

5.6.3 Evaluation of Problem TUVI-CD.

Sum of Scores. For problem *TUVI-CD*, Figure 7 depicts the s_{sum} obtained by MES, SW-MES, and other algorithms on datasets $\mathcal{V}_{\text{c\&n}}$, $\mathcal{V}_{\text{n\&r}}$ and $\mathcal{V}_{\text{c\&n\&r}}$. While MES remains superior to SGL, BF, and EF for *TUVI-CD*, its performance declines compared to *TUVI* due to its inability to adapt to the concept drift in the datasets. Conversely, SW-MES, which adjusts to concept drift using the sliding-window mechanism, consistently achieves higher scores than MES and other approaches (except OPT) while maintaining a relatively smaller standard deviation and a narrower range between the minimum and maximum scores.

5.7 Detailed Evaluation Analysis

In this section, we conduct a detailed evaluation of the algorithms introduced in this paper, focusing on the *TUVI* problem definition, including the ablation study, parameters analyses, and resource utilization analyses.

5.7.1 Ablation Study. We also investigate a variant of MES, named MES-A, which excludes the step of applying ensembles corresponding to all subsets of the selected ensemble in each iteration (as shown in Lines 9-10 of Alg. 1). Comparing MES-A with EF and MES on each dataset, Figure 8 depicts the sum of scores for each algorithm (normalized by the score of MES) across all datasets. Despite outperforming EF, MES-A experiences a significant drop in performance across all datasets.

5.7.2 Relative Importance of Components in Scoring Functions. We further explore the relative importance of the accuracy and time cost components within the scoring function by varying the weight combinations $\langle w_1, w_2 \rangle$ and provide insights on choosing weights.

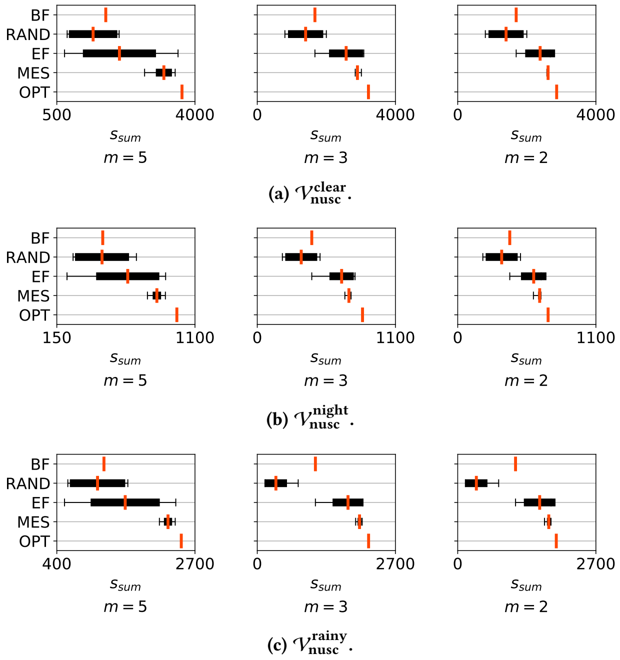
Relative Importance. Figure 9 shows the scores obtained by the algorithms under different weight combinations. RAND results are unsatisfactory and erratic across all combinations. When w_1=0.1 and w_2=0.9 (i.e., when the scoring function is dominated by the Cost component), BF performs significantly worse than MES, as it always selects the most complex and expensive ensembles. SGL's performance is similar. As w_1 increases and w_2 decreases (i.e., the weight of the Accuracy component grows), the scores of SGL, BF and EF gradually approach those of MES and OPT. For w_1=0.9 and w_2=0.1, MES still has higher s_{sum} than EF, but the advantage diminishes. In summary, our proposed methods naturally adapt to different application scenarios with varying importance of accuracy and inference time.

Insights on Choosing Weights. In addition, a suitable weight combination should be assigned to the scoring function to influence ensemble selection based on actual application requirements. If the weight of accuracy is greater than that of efficiency, our ensemble selection strategy favors ensembles with high accuracy; otherwise, it leans toward ensembles with faster inference time. Figure 10 displays the distribution of the number of times the ensembles are selected in MES under varying weights on dataset $\mathcal{V}_{\text{nusc}}$, with each circle representing an ensemble and darker colors representing more selections. When w_2>w_1, MES tends to select ensembles in the lower right part of the figure; when w_2=w_1, MES favors ensembles in the middle part; and when w_2<w_1, MES prefers ensembles in the upper left part. Similar trends are observed in the other datasets, allowing users to choose weights based on these patterns.

5.7.3 Analysis of Ensemble Quantity. We analyze the influence of the number of ensembles, 2^m-1, where m=$|\mathcal{M}|$ is the number of detectors. By reducing the number of detectors in $\mathcal{M}$ (from m=5 to m=3 and m=2), we present Figure 11, which displays the sum of scores obtained by algorithms over varying m for *TUVI* on datasets $\mathcal{V}^{\text{clear}}_{\text{nusc}}$, $\mathcal{V}^{\text{night}}_{\text{nusc}}$ and $\mathcal{V}^{\text{rainy}}_{\text{nusc}}$. It can be observed that, compared with m=5 (i.e. a total of 31 ensembles), the gap between BF/EF and MES becomes smaller as m decreases. When m=2 (i.e. a total of 3 ensembles), the mean value of the total scores of EF is equal to MES. This is because the difficulty of selecting the optimal ensemble diminishes as the number of ensembles decreases for all approaches. Similar trends are observed on the other datasets.

5.7.4 Impact of Hyper-parameter Gamma. It depicts the sum of scores of MES on datasets $\mathcal{V}^{\text{clear}}_{\text{nusc}}$, $\mathcal{V}^{\text{night}}_{\text{nusc}}$ and $\mathcal{V}^{\text{rainy}}_{\text{nusc}}$ varying the initialization parameter γ (as shown in Lines 2-3 of Alg. 1) in Figure 12. In general, a smaller γ may lead to less accurate AP estimation for each ensemble during the initialization phase, potentially impacting the selection in subsequent iterations. Conversely, since the initialization step is computationally demanding, requiring inference on all models, choosing an excessively large value for γ could result in lower overall scores. The figure illustrates that as γ progresses from smaller to larger values, the scores initially increase and subsequently decrease.

5.7.5 Details of Resource Utilization. It presents the proportion of the time spent on each of the components of MES throughout the entire process on dataset $\mathcal{V}_{\text{nusc}}$ in Figure 13. As can be observed from the figure, the time spent on ensembling and other



(a) $\mathcal{V}_{\mathbf{nusc}}^{\mathbf{clear}}$.

(b) $\mathcal{V}_{\mathbf{nusc}}^{\mathbf{night}}$.

(c) $\mathcal{V}_{\mathbf{nusc}}^{\mathbf{rainy}}$.

Figure 11: Scores obtained by algorithms over varying $|\mathcal{M}|$.

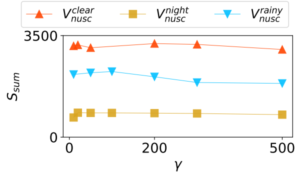


Figure 12: s_{sum} of MES varying γ.

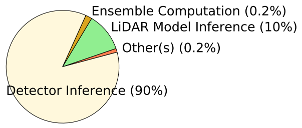


Figure 13: Proportion of runtime overhead of each MES component.

optimization components (such as the computations in Lines 5-6 of Alg. 1 and optimizations in Lines 9-10 of Alg. 1) is negligible (0.4%). This demonstrates that the algorithms we proposed incur almost negligible overhead. The dominant time spent is on detector inference (90%), followed by the LiDAR model inference (10%), which are necessary overheads inherent to the process.

6 RELATED WORK

Model Prediction Ensembling. Object detection (OD) is one of the most important topics in computer vision since it has many applications in several fields [12]. Model prediction ensembling approaches have been used in object detection to improve accuracy [31, 37, 45]. Non-Maximum Suppression (NMS) [29, 52] is proposed to eliminate redundant bounding boxes predicted by an OD model, which can be directly used to ensemble the predictions among various models. Some improved variants of NMS are proposed, such as Soft-NMS [8] and Softer-NMS [34], which can also be applied to model ensembling. In addition, other BBoxes-based IOU ensembling approaches have also been proposed to ensemble OD results, such as Fusion [61], NMW [67] and WBF [56].

Automated Video Analytics. Automated video analytics utilizing deep learning models (such as object detection models and multiple object tracking models) as primitives is an area of growing research interest in the community [2, 10, 13–15, 19, 42, 44, 64, 65]. Numerous recent works present query processing frameworks that include both frame content (object types, positions in frames, etc.) and temporal constraints (object tracking outputs, etc.) as query primitives [14, 17, 18, 65]. For example, NoScope and BlazeIt [40, 41] utilize proxy models to detect objects accelerating queries via inference-optimized model search; SVQ and SVQ++ [14, 44, 64, 65] provide filters to accelerate queries involving complicated constraints on/between objects present in the frames. A prerequisite for answering the queries accurately is the correct extraction of OD metadata by deep learning algorithms (i.e., object detection models). In this work, leveraging the model prediction ensembling approaches, we propose a general algorithm to improve the OD accuracy, which is orthogonal to many of the downstream query processing techniques used in previous work.

Model/Ensemble Selection. Selecting the appropriate models or ensembles (from a collection of models/ensembles) for processing input is essential. [30] and [53] employ decision trees to detect drift that occurs in a data stream and select models that can deliver higher accuracy on recent concepts. ODIN [57] proposes a selector for picking ensembles of specialized models for processing a given image input. In our work, due to the lack of knowledge about the dataset and our treatment of models as black boxes for generality, it is impossible to adopt the aforementioned algorithms, which select ensembles by classifying concepts.

Multi-objective Query Optimization (MOQO). MOQO models the cost of a query plan as a vector instead of a scalar value, to accommodate multiple (often conflicting) execution metrics such as time and resource use [59]. The first category of approaches to the MOQO problem simplifies this by using weighted sums to collapse these dimensions into a single objective [27], while the second category of methods identifies Pareto-optimal plans, where no alternative plan is superior across all metrics [58, 59]. By consolidating multiple criteria into a single scoring function, the methodology adopted in this paper is akin to the first category of approaches to MOQO and provides a computationally efficient and manageable solution. Exploring the full spectrum of optimal solutions by redefining the problem within the MOQO framework, including identifying Pareto-optimal ensembles, represents a promising direction for future research.

7 CONCLUSIONS

Object detection algorithms serve as the foundation for video query frameworks that involve various object constraints. Model prediction ensembling techniques can enhance object detection accuracy, but they also incur additional inference costs. In this paper, we address the problem of selecting suitable ensembles that optimize a score comprising both accuracy and inference time, without requiring prior knowledge about the video and detectors. We propose an algorithm, MES, which effectively allocates computational resources for identifying appropriate ensembles. We further refine our approach by introducing the MES-B algorithm for conducting ensemble selection within a specified budget and the SW-MES algorithm for adapting to concept drifts during ensemble selection. Comprehensive experimental results on real video datasets confirm the effectiveness of our proposed algorithms under a range of settings. We believe that improving the quality of preprocessing results in video analysis systems is a crucial research direction that warrants further exploration.

REFERENCES

[1] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2 (2002), 235–256.

[2] Favyen Bastani and Samuel Madden. 2022. OTIF: Efficient Tracker Preprocessing over Large Video Datasets. In *Proceedings of the International Conference on Management of Data.* 2091–2104.

[3] Eric Bauer and Ron Kohavi. 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning* 36, 1 (1999), 105–139.

[4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. 2019. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 941–951.

[5] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems* 24 (2011).

[6] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research* 13, 2 (2012).

[7] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP).* IEEE, 3464–3468.

[8] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. 2017. Soft-NMS–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision.* 5561–5569.

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 11621–11631.

[10] Jiashen Cao, Karan Sarkar, Ramyad Hadidi, Joy Arulraj, and Hyesoon Kim. 2022. FiGO: Fine-Grained Query Optimization in Video Analytics. (2022).

[11] Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[12] Ángela Casado-García and Jónathan Heras. 2020. Ensemble methods for object detection. In *ECAI 2020.* IOS Press, 2688–2695.

[13] Daren Chao, Yueting Chen, Nick Koudas, and Xiaohui Yu. 2023. Track Merging for Effective Video Query Processing. In *2023 IEEE 39th International Conference on Data Engineering.*

[14] Daren Chao, Nick Koudas, and Ioannis Xarchakos. 2020. Svq++: Querying for object interactions in video streams. In *Proceedings of the International Conference on Management of Data.* 2769–2772.

[15] Daren Chao, Nick Koudas, and Xiaohui Yu. 2023. Marshalling Model Inference In Video Streams. In *2023 IEEE 39th International Conference on Data Engineering.*

[16] Yueting Chen, Nick Koudas, Xiaohui Yu, and Ziqiang Yu. 2022. Spatial and temporal constrained ranked retrieval over videos. *Proceedings of the VLDB Endowment* 15, 11 (2022), 3226–3239.

[17] Yueting Chen, Xiaohui Yu, and Nick Koudas. 2020. TQVS: Temporal Queries over Video Streams in Action. In *Proceedings of the International Conference on Management of Data.* 2737–2740.

[18] Yueting Chen, Xiaohui Yu, Nick Koudas, and Ziqiang Yu. 2021. Evaluating Temporal Queries Over Video Feeds. In *Proceedings of the International Conference on Management of Data.* 287–299.

[19] Pramod Chunduri, Jaeho Bang, Yao Lu, and Joy Arulraj. 2022. Zeus: Efficiently Localizing Actions in Videos using Reinforcement Learning. (2022).

[20] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems.* Springer, 1–15.

[21] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. 2013. Multi-armed bandit with budget constraint and variable costs. In *Twenty-Seventh AAAI Conference on Artificial Intelligence.*

[22] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision* 111, 1 (2015), 98–136.

[23] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88, 2 (2010), 303–338.

[24] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional Two-Stream Network Fusion for Video Action Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[25] Yoav Freund, Robert E Schapire, et al. 1996. Experiments with a new boosting algorithm. In *icml*, Vol. 96. Citeseer, 148–156.

[26] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 1–37.

[27] Sumit Ganguly, Waqar Hasan, and Ravi Krishnamurthy. 1992. Query optimization for parallel execution. In *Proceedings of the 1992 ACM SIGMOD international conference on management of data.* 9–18.

[28] Aurélien Garivier and Eric Moulines. 2011. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory.* Springer, 174–188.

[29] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 580–587.

[30] Heitor M Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfharinger, Geoff Holmes, and Talel Abdessalem. 2017. Adaptive random forests for evolving data stream classification. *Machine Learning* 106 (2017), 1469–1495.

[31] Ruoyu Guo, Cheng Cui, Yuning Du, Xianglong Meng, Xiaodi Wang, Jingwei Liu, Jianfeng Zhu, Yuan Feng, and Shumin Han. 2019. 2nd Place Solution in Google AI Open Images Object Detection Track 2019. *arXiv preprint arXiv:1911.07171* (2019).

[32] Cédric Hartland, Sylvain Gelly, Nicolas Baskiotis, Olivier Teytaud, and Michele Sebag. 2006. Multi-armed bandit, dynamic environments and meta-bandits. (2006).

[33] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision.* 2961–2969.

[34] Yihui He, Xiangyu Zhang, Marios Savvides, and Kris Kitani. 2018. Softer-nms: Rethinking bounding box regression for accurate object detection. *arXiv preprint arXiv:1809.08545* 2, 3 (2018), 69–80.

[35] Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association* 58, 301 (1963), 13–30.

[36] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B Gibbons, and Onur Mutlu. 2018. Focus: Querying large video datasets with low latency and low cost. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18).* 269–286.

[37] Zehao Huang, Zehui Chen, Qiaofei Li, Hongkai Zhang, and Naiyan Wang. 2020. 1st Place Solutions of Waymo Open Dataset Challenge 2020–2D Object Detection Track. *arXiv preprint arXiv:2008.01365* (2020).

[38] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. 2018. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European conference on computer vision (ECCV).* 784–799.

[39] Daniel Kang, Nikos Arechiga, Sudeep Pillai, Peter D Bailis, and Matei Zaharia. 2022. Finding label and model errors in perception data with learned observation assertions. In *Proceedings of the 2022 International Conference on Management of Data.* 496–505.

[40] Daniel Kang, Peter Bailis, and Matei Zaharia. 2019. BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. *Proceedings of the VLDB Endowment* 13, 4 (2019), 533–546.

[41] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: Optimizing Deep CNN-Based Queries over Video Streams at Scale. *Proceedings of the VLDB Endowment* 10, 11 (2017), 1586–1597.

[42] Daniel Kang, John Guibas, Peter Bailis, Tatsunori Hashimoto, and Matei Zaharia. 2022. TASTI: Semantic Indexes for Machine Learning-based Queries over Unstructured Data. (2022).

[43] Josef Kittler, Mohamad Hatef, Robert PW Duin, and Jiri Matas. 1998. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence* 20, 3 (1998), 226–239.

[44] Nick Koudas, Raymond Li, and Ioannis Xarchakos. 2020. Video monitoring queries. In *IEEE International Conference on Data Engineering.* 1285–1296.

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

[46] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2117–2125.

[47] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision.* Springer, 740–755.

[48] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. 2020. A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP).* IEEE, 237–242.

[49] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. 2018. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 918–927.

[50] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 652–660.

[51] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 779–788.

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

[53] Siqi Ren, Bo Liao, Wen Zhu, and Keqin Li. 2018. Knowledge-maximized ensemble algorithm for different types of concept drift. *Information Sciences* 430 (2018), 261–281.

[54] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems.* 568–576.

[55] Aleksandrs Slivkins et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.

[56] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. 2021. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image*

and Vision Computing 107 (2021), 104117.

[57] Abhijit Suprem, Joy Arulraj, Calton Pu, and Joao Ferreira. 2020. Odin: Automated drift detection and recovery in video analytics. *arXiv preprint arXiv:2009.05440* (2020).

[58] Immanuel Trummer and Christoph Koch. 2014. Approximation schemes for many-objective query optimization. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 1299–1310.

[59] Immanuel Trummer and Christoph Koch. 2017. Multi-objective parametric query optimization. *Commun. ACM* 60, 10 (2017), 81–89.

[60] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* (2022).

[61] Pan Wei, John E Ball, and Derek T Anderson. 2018. Fusion of an ensemble of augmented image detectors for robust object detection. *Sensors* 18, 3 (2018), 894.

[62] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, 3645–3649.

[63] Yutian Wu, Yueyu Wang, Shuwei Zhang, and Harutoshi Ogai. 2020. Deep 3D object detection networks using LiDAR data: A review. *IEEE Sensors Journal* 21, 2 (2020), 1152–1171.

[64] Ioannis Xarchakos and Nick Koudas. 2019. Svq: Streaming video queries. In *Proceedings of the International Conference on Management of Data*. 2013–2016.

[65] Yannis Xarchakos and Nick Koudas. 2021. Querying for interactions. In *IEEE International Conference on Data Engineering*. 2153–2158.

[66] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2636–2645.

[67] Huajun Zhou, Zechao Li, Chengcheng Ning, and Jinhui Tang. 2017. Cad: Scale invariant framework for real-time object detection. In *Proceedings of the IEEE international conference on computer vision workshops*. 760–768.

[68] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. 2019. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492* (2019).