Received: October 18, 2021. Revised: December 20, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

NGABUDI: Neural Guided Adaptive Bootstrapped Unsupervised Driving Intelligence

Aradea [1]*, Rianto[1], Zeni Muhamad Noer[2], Ghatan Fauzi Nugraha [3], Pandu Pangestu[1]

[1]*Department of Informatics, Faculty of Engineering University of Siliwangi, Indonesia*
[2]*Department of Informatics Management, STMIK DCI, Tasikmalaya, Indonesia*
[3]*Directorate of Human Resource, Jakarta State University, Jakarta, Indonesia*
* Corresponding author's Email: aradea@unsil.ac.id

Abstract: In this study, we introduce NGABUDI, a self-learning vision system for autonomous driving. NGABUDI combines a YOLO26s student detector, open-vocabulary pseudo-label generation, Adaptive Experience Replay, and a Self-Adaptive Cyber-Physical System control loop. This setup lets the system keep adapting to new, unlabeled sensor data without manual reannotation. We tested NGABUDI on the KITTI dataset with controlled changes in lighting and color. The student model trained on the original data reached 91.9% mAP_{50} and 71.0% $mAP_{50\text{-}95}$ using 20.8 GFLOPs. When tested on the changed data, the frozen model's mAP50 dropped to 58.90%. After running 5985 unlabeled frames through 281 updates and 844 gradient steps, NGABUDI improved mAP_{50} to 65.76% and $mAP_{50\text{-}95}$ to 41.89%. Tests across different conditions showed almost no drop in performance on clean images, and the biggest improvement happened when all changes were combined. Our ablation studies showed that source anchoring, photometric restoration, AdaER memory, and distillation are all important for the system's performance.

Keywords: AdaER, Continual Learning, SACPS, Self-learning, YOLO26.

1. Introduction

Autonomous driving depends on computer vision systems for real-time and accurate perception of the environment. While deep learning models perform well in ideal test settings, their accuracy often drops sharply when real-world conditions differ from the training data [1] [2]. Factors like weather, lighting, or traffic changes can create gaps between training and real-world data, making static models struggle to generalize [3]. This shows that static approaches are not robust enough for real-world autonomous driving.

Self-learning vision system approaches have emerged as a promising solution in which a system continuously updates its visual representation and exploits contextual knowledge from unlabeled sensory data without depending on manual annotation [4]. Several studies have attempted to improve autonomous driving perception systems through real-time object detection using YOLO model variants. The study in [5] demonstrated that YOLOv5, YOLOv7, and YOLOv8 work well under simulator-based evaluation. Beyond baseline models, other studies have modified the YOLO architecture to obtain performance aligned with autonomous driving scenarios. WGS-YOLO proposed by Yue et al. [6] showed significant improvements in vehicle and pedestrian detection, outperforming previous versions. Studies [7] and [8] modified the attention module and the loss function to maximize detection, especially for small objects in road scenes. CMS-YOLO [9], DAN-YOLO [10], and DCW-YOLO [11] made changes to improve detection in unstable environments. Beyond these YOLO versions, some recent studies have used semi-supervised methods or learning from unlabeled data to help detectors handle changing conditions [12]. Despite these advances, a gap remains in

research on vision systems that fully use an end-to-end self-learning approach. Most current object detection systems are static and need manual retraining when data changes [2]. Automatic adaptation is also often only partially included, and many methods still require re-annotation or complex computing setups, making them hard to use independently in the field [13].

This study introduces NGABUDI (Neural Guided Adaptive Bootstrapped Unsupervised Driving Intelligence), a self-learning vision system that combines real-time object detection using YOLO26 [14], pseudo-labelling with a self-learning approach, and continual learning with AdaER (Adaptive Experience Replay) [15]. The system lets an autonomous vehicle update its knowledge from unlabeled sensory data without manual re-annotation, managed by the SACPS (Self-Adaptive Cyber Physical System) control mechanism [16]. SACPS uses the MCC concept (Monitoring, Cognition, and Configuration) to guide the system, so it acts on knowledge from contextual analysis, not just simple cause-and-effect. These parts work together to help the system adapt to new data, keep previous object detections stable, and run efficiently in real time. This study offers a complete framework that brings together real-time object detection, adaptive continual learning, and SACPS-based control in one vision system.

2. Related Works

Development of perception systems for driverless vehicles has recently advanced significantly through the adoption of deep learning-based object detection models [17]. n autonomous driving, object detection is critical because the system must rapidly and accurately recognize vehicles, pedestrians, cyclists, traffic signs, and road obstacles [18]. Stable detection performance largely determines a vehicle's ability to understand its surroundings, especially when operating in dynamic real-world conditions that do not always match the training data [1]. Object detection systems must be highly accurate and operate in real time so perception results can serve as the basis for vehicle decision-making [2]. The YOLO family of models has become one of the most widely used approaches for this task because it uses single-stage detection, allowing object localisation and class classification to be performed in a single inference pass [19].

Numerous studies have developed YOLO models to improve object detection performance in autonomous vehicles. WOG-YOLO [20] builds on YOLOv5 by exploiting hyperparameter optimization, an additional prediction head, and the Ghost-C3 module. This approach improves detection performance on the KITTI dataset, but the additional prediction head may increase computational load. MCS-YOLO [21] integrates Coordinate Attention, a multi-scale detection structure, and the Swin Transformer to increase sensitivity to small objects in highway environments. This approach demonstrates the importance of multi-scale information in autonomous driving, but transformer integration can raise architectural complexity.

Another development is CMS-YOLO [9], which modifies the backbone, neck, and detection head to improve multi-object detection in road scenes. This model shows that improvements along the feature aggregation path help the model handle object variation in highway environments. However, the decoupled head structure and the more complex neck modification can add implementation overhead.

Some studies focus on detecting small objects, which is often difficult on highways. SOD-YOLOv8 [7] adds a layer for small object detection, an attention module, and PIoU loss. This helps the model find small objects better, though it can slow down inference. Li et al. created SNCE-YOLO [8] with SPD-Conv, NAM attention, C2f, and EIoU loss to improve detection in complex road scenes. These efforts show that small objects and scale changes remain major challenges for object detection in autonomous vehicles.

Robustness to environmental conditions is also a primary concern in developing object detection models. Cui et al. proposed DAN-YOLO [10], based on YOLOv7-Tiny with a Dilated Aggregation Network, Hybrid Dilated Convolution, EMA, and Wise-IoU to improve detection under challenging driving conditions. Ren et al. developed DCW-YOLO [11] based on YOLOv8s with a Dynamic Head, Coordinate Attention on SPPF, and Wise-IoU v3 to improve road object detection. Both studies show that receptive field expansion, attention mechanisms, and loss function optimisation can improve detection performance. Nevertheless, these additional mechanisms may introduce a trade-off between accuracy and computational efficiency.

Newer versions of YOLO have been used to make object detection lighter and more efficient. Yang et al. introduced LEAD-YOLO [22], based on YOLOv11n with a based on YOLOv11n, which uses a Convolutional Gated Transformer, Dilated Feature Fusion, a Hierarchical Feature Fusion Module, and a Small Feature Detection Head to detect small objects better. Liang et al. developed

YOLO11-FLC [23] or vehicle-road cooperative driving by replacing the C3k2 module with C3CTA, adding DFPN, and using a Lightweight Shared Head. Wei et al. proposed SCCA-YOLO [18], based on YOLOv8, with Spatial and Channel Collaborative Attention for highway driving. Overall, YOLO development for autonomous vehicles continues to improve accuracy, efficiency, and the ability to detect small objects and handle complex road conditions. Table 1 provides an overview of previous studies, including their methods, model changes, strengths, weaknesses, and research gaps.

Table 1. State of the art of object detection research for autonomous vehicles using YOLO

Title and Authors	Method	Modification Techniques	Remarks
The research of a novel WOG-YOLO algorithm for autonomous driving object detection [1]	WOG-YOLO (based on YOLOv5)	Hyperparameter optimization via the modified Whale Optimization Algorithm (MWOA), addition of a fourth prediction head, and the Ghost-C3 module.	Hyperparameter optimization with MWOA together with the fourth prediction head and the Ghost-C3 module raised mAP to 94.2% and improved pedestrian mAP by +2.6% on the KITTI dataset, but increased the computational load and lengthened training time.
MCS-YOLO: A Multiscale Object Detection Method for Autonomous Driving Road Environment Recognition [21]	MCS-YOLO (based on YOLOv5)	Coordinate Attention module, a multi-scale small-object detection structure, and Swin Transformer blocks.	The Coordinate Attention module and Swin Transformer blocks improve sensitivity to small dense objects, raising mAP by 4.3% over the baseline, but add computational complexity compared with pure CNN structures.
Road Scene Multi-Object Detection Algorithm Based on CMS-YOLO [9]	CMS-YOLO (based on YOLOv5s)	*CD backbone (Cross Stage Partial DWNeck), MFFPN neck, and SDH head (Special Decoupled Head).*	The CD backbone, MFFPN neck, and SDH head improve mAP by +7.2% over YOLOv5s and resolve classification task conflicts, although the more complex decoupled-head structure adds a small number of parameters.
Real-time Object Detection in Autonomous Vehicles with YOLO [5]	Comparison of YOLO versions (v5, v7, v8)	Implementation of self-supervised learning and construction of the VSim-AV simulator dataset.	Self-supervised learning and the VSim-AV simulator dataset were successfully established, but YOLOv7 recorded notably poor performance (low recall/precision) in this simulation study.
SOD-YOLOv8 - Enhancing YOLOv8 for Small Object Detection in Traffic Scenes [7]	SOD-YOLOv8s (based on YOLOv8s)	PIoU loss, C2f-EMA attention module, and the addition of a Small Object Detection Layer (a fourth head).	PIoU loss, the C2f-EMA attention module, and the fourth head yield very high recall for small objects and superior performance on aerial imagery, but increase the computational load and inference latency.
MSSD: multi-scale self-distillation for object detection [24]	MSSD (applied to YOLOv5/X)	*Multi-scale self-distillation and Gaussian mask assisted detection.*	*Multi-scale self-distillation and Gaussian mask assisted detection are efficient in training cost because no large teacher model is required, yet self-driven learning has a performance ceiling compared with learning from a large teacher.*
SNCE-YOLO: An Improved Target Detection Algorithm in Complex Road Scenes [8]	SNCE-YOLO (based on YOLOv5s)	SPD-Conv (replacing strided convolution for low resolutions), NAM (Normalization-based Attention), the C2f structure, and EIoU loss.	SPD-Conv, NAM, the C2f structure, and EIoU loss reduce miss-detections at low resolutions with a good balance between speed (95 FPS) and accuracy, although the substantial architectural changes require further development for deployment.
DAN-YOLO: A Lightweight and Accurate Object Detector Using Dilated Aggregation Network for Autonomous Driving [25]	DAN-YOLO (based on YOLOv7-Tiny)	SPPFCSPC-E structure (Hybrid Dilated Convolution), DAN (Dilated Aggregation Network), EMA attention, and Wise-IoUv1 loss.	The SPPFCSPC-E structure, DAN, EMA attention, and Wise-IoUv1 loss improve accuracy in adverse weather through a wider receptive field, but reduce speed by 18.87% (to 142.86 FPS) and increase parameters from 6.02M to 8.10M.
DCW-YOLO: Road Object Detection Algorithms for Autonomous Driving [11]	DCW-YOLO (based on YOLOv8s)	DyHead (Dynamic Head unification), SPPFCA (Coordinate Attention on SPPF), and Wise-IoU v3 loss.	DyHead, SPPFCA, and Wise-IoU v3 loss reduce parameters while improving mAP (+2.1% on KITTI) through better local-global feature fusion, although the added attention complexity lowers detection speed to 89.2 FPS versus the YOLOv8s baseline (99.1 FPS).
LEAD-YOLO: A Lightweight and Accurate Network for Small Object Detection in	LEAD-YOLO (based on YOLOv11n)	CGF module (Convolutional Gated Transformer), DFF (Dilated Feature Fusion),	The CGF, DFF, and HFFM modules and the SFDH head are designed for small object detection, but the additional attention structure

Autonomous Driving [22]		HFFM (Hierarchical Feature Fusion), and the SFDH head.	reduces FPS to 167.5 versus the YOLO11n baseline (172.4) and slightly increases GFLOPs in certain neck stages.
Object Detection Model of Vehicle-Road Cooperative Autonomous Driving Based on Improved YOLO11 Algorithm [23]	*YOLO-FLC (improved YOLO11)*	Replacing C3k2 with C3CTA (Channel Transposed Attention), adding DFPN (Diffusion Focusing Pyramid), and LSCD (Lightweight Shared Head).	Replacing C3k2 with C3CTA together with DFPN and LSCD achieves the highest precision (85.7%) on the DAIR-V2X-I dataset with a light (2.51M parameters) and fast (167 FPS) model, but the focus on vehicle-road cooperation requires adaptation for single-vehicle scenarios.
SCCA-YOLO: A Spatial and Channel Collaborative Attention Enhanced YOLO Network for Highway Autonomous Driving Perception System [18]	SCCA-YOLO (based on YOLOv8)	SCCA module (Spatial and Channel Collaborative Attention) and Ghost module integration for parameter efficiency.	The SCCA module and Ghost integration improve detection accuracy for large animals on remote roads, such as 98.2% horse detection, with total mAP rising 0.7% to 0.844 at efficient parameter cost, but the focus on rural roads leaves dense urban traffic untested.

Many studies have shown better object detection for autonomous vehicles, but most still focus on improving the detector architecture in a fixed way. Usually, these models are trained on one dataset and then used in real-world settings without a way to update their knowledge when the data changes. This is a major drawback because the autonomous driving environment is always changing. Factors like lighting, weather, traffic, camera angles, and object types can differ from the training data. As a result, a detector that works well in ideal test conditions might still lose accuracy in real-world situations.

Based on this review, this study introduces NGABUDI to help driverless vehicles adapt by using YOLO26s for real-time object detection. YOLO26s was selected because it is more compact and efficient, which makes it suitable for deployment as the perception engine in a driverless vehicle. Unlike earlier research that often adds new modules to older YOLO versions, this study uses YOLO26s as a complete, compact perception engine. The goal is to build a real-time object detection system that can support adaptive vision in driverless vehicles.

3. Proposed Method

Based on the gap and the problems described above, we formulate NGABUDI (Neural Guided Adaptive Bootstrapped Unsupervised Driving Intelligence) as a self-learning vision system that integrates real-time object detection, autonomous label acquisition from unlabeled sensory data, and continual learning governed by an autonomous adaptation loop. Figure 1 shows the NGABUDI architecture.

Conceptually, the system runs two temporal pathways over the same camera frame stream. The first pathway executes the YOLO26-based student detector [14] on every frame to produce detections and operational alerts in real time. The second pathway activates the teacher subsystem, based on an open-vocabulary detector [26] to generate pseudo-labels for classes never annotated in the base dataset, then uses those pseudo-labels to update the student incrementally through adaptive experience replay [15]. The two pathways are bridged by a self-adaptive cyber-physical system (SACPS) control loop [16] with Monitoring, Cognition, and Configuration stages that determine when the slow pathway needs to run. Under this arrangement, system adaptation is an action derived from operational context analysis rather than the result of fixed scheduling. Throughout the following description, θ denotes the set of student parameters, t the frame index, and c the object class index.

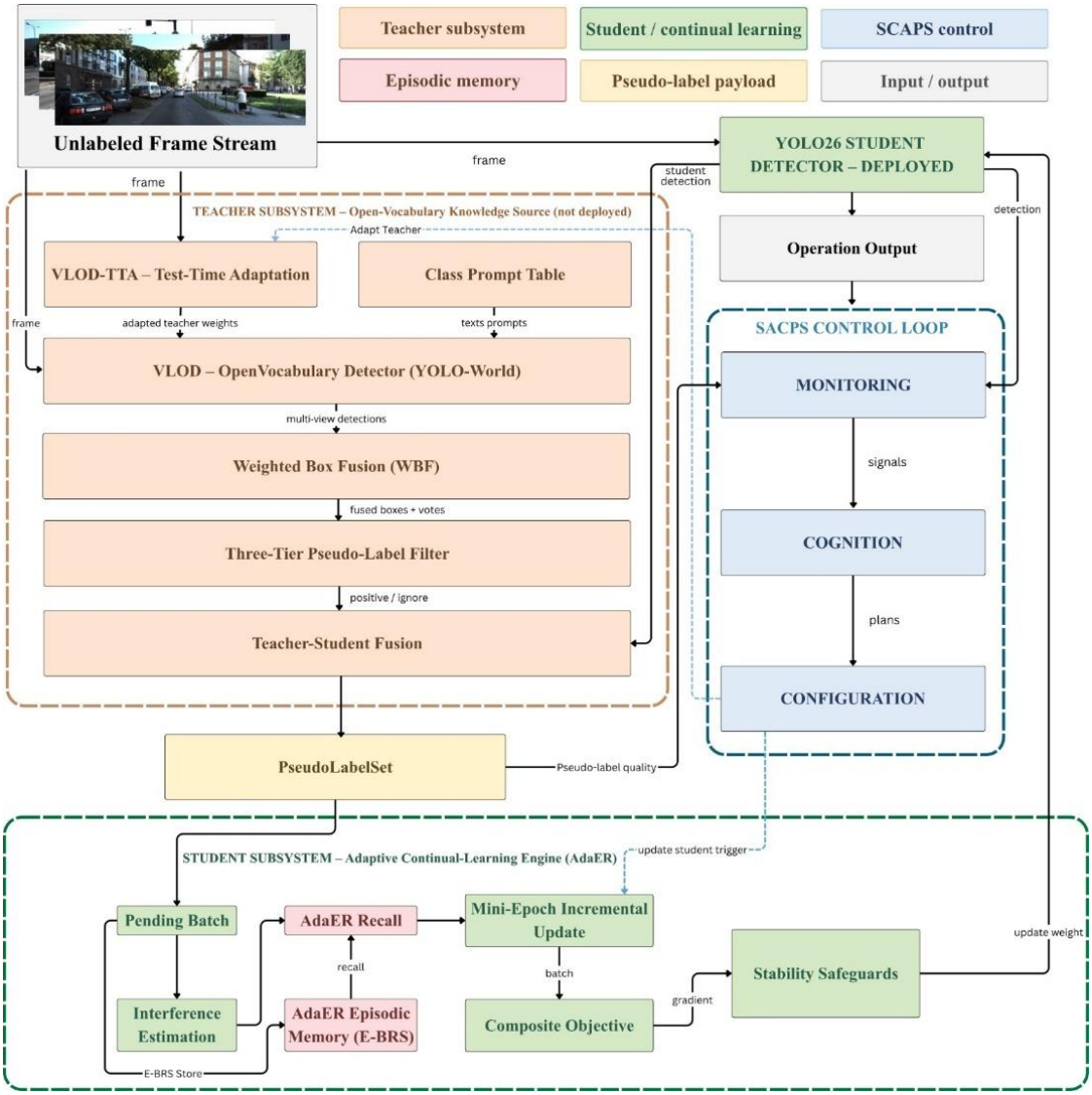

Figure. 1 Architecture of NGABUDI

The teacher subsystem operates in four sequential stages. The first stage is test-time adaptation to the frame being processed, designed as an extension of entropy minimization [27] into the object detection domain. Suppose the teacher detector produces a set of proposals, where z_i denotes the class logit vector of the i-th proposal and $p_i = \mathrm{softmax}(z_i)$ its class probability distribution, so that $\mathcal{H}_i = -\sum_c p_{ic} \log p_{ic}$ is the Shannon entropy of that proposal and $s_j = \max_c p_{jc}$ is the confidence score of the j-th proposal. A_{ij} is defined as the Intersection over Union between proposals i and j, zeroed on the diagonal and on pairs whose intersection falls below the threshold $\tau_{\mathrm{clu}} = 0.50$. The weight of each proposal is then set as its spatial density, $\omega_i = \sum_j A_{ij}\, s_j$, so that proposals located where many other high-scoring proposals overlap receive greater influence. The teacher adaptation loss is formulated as a weighted entropy in Equation (1).

$$\mathcal{L}_{\mathrm{TTA}} = \frac{\sum_i \omega_i \, \mathcal{H}_i}{\sum_i \omega_i + \epsilon} \tag{1}$$

The second stage is multi-view inference. The open-vocabulary detector is run on several variants of the frame with a deliberately lowered internal confidence threshold so that the resulting proposals are dense, since the actual filtering takes place in the following stage. The third stage merges the outputs across views using weighted box fusion [28] instead of non-maximum suppression. This choice is deliberate because NMS discards duplicate boxes, whereas WBF averages the member boxes of a cluster in a score-weighted manner, making bounding box localization more precise.

The fourth stage is tiered pseudo-label filtering, which is the main determinant of student learning quality. Filtering starts with prompt-to-class mapping and the removal of very low-scoring

candidates, followed by the application of geometric priors consisting of a minimum box area bound, a minimum side length, a maximum aspect ratio, and a height prior that rejects boxes of implausible height except for classes that are inherently tall. An acceptance threshold that adapts per class is then applied. Letting $\mathcal{Q}_c$ denote the detection score history of class c over a sliding window and $Q_q(\cdot)$ the quantile operator of order q with $q = 0.65$, the threshold of class c is set as in Equation (2).

$$\tau_c = \text{clip}\,(\min(\tau_{\text{pos}}, Q_q(\mathcal{Q}_c)), \tau_{\min}, \tau_{\text{pos}}) \quad (2)$$

Here $\tau_{\text{pos}} = 0.40$ serves as the base positive threshold as well as the upper bound, $\tau_{\min} = 0.25$ as the lower bound, and $\text{clip}(x, a, b)$ denotes clamping of x to the interval $[a, b]$. Formulation (2) addresses the inter-class balancing problem in which classes with few labels whose score scales are naturally lower are not starved of labels, in contrast to a uniform single threshold. Subsequently, a candidate with score s and vote count v is accepted as a POSITIVE label if it satisfies Equation (3).

$$(s \geq \tau c) \wedge (v \geq vmin) \wedge \big(hits \geq hmin \vee s \geq min(0{,}75, \tau c + \beta)\big) \quad (3)$$

Here $v_{\min}$ is the required minimum number of views, hits is the number of preceding frames within the temporal window containing similar boxes, $h_{\min} = 2$ is its minimum value, and $\beta = 0.15$ is the shortcut margin that admits candidates with very high scores without waiting for temporal confirmation. Candidates that fail (3) are not treated as background but are placed in the IGNORE region set ($\mathcal{I}_t$) following pedestrian detection benchmark conventions [29].

The teacher output is then fused with student detections through three rules. First, teacher and student boxes that overlap strongly and share the same class are considered correct bounding boxes and their pseudo-label scores are raised. Second, boxes that overlap strongly but belong to different classes are marked as conflicts and diverted to $\mathcal{I}_t$. Third, student boxes that overlap no teacher proposal are promoted to positive labels. ogether, these rules directly counterbalance the teacher for classes it has not mastered. The final teacher output for each frame is the pair of positive label sets P_t and ignore regions $\mathcal{I}_t$ together with a frame quality score.

The SACPS control loop observes both subsystems through three quantitative signals. The first signal is the disagreement between teacher and student, computed through IoU matching. Letting $\mathcal{T}_t$ and $\mathcal{S}_t$ denote the teacher pseudo-label set and the student detection set on frame t, and n_{match} the number of successfully matched pairs, disagreement is defined as the complement of the Sørensen-Dice coefficient in Equation (4).

$$\mathcal{D}_t = 1 - \frac{2\, n_{\text{match}}}{|\,\mathcal{T}_t\,| + |\,\mathcal{S}_t\,|}. \quad (4)$$

The other two signals are derived relative to an exponentially weighted moving average (EWMA) baseline, namely the decrease of the average student confidence Δ_t^{conf} and the change of the detection rate Δ_t^{rate}. Both are computed before the baseline is updated so that deviations are not absorbed by the update on the same frame. The three signals are condensed into a single composite context score in Equation (5).

$$g_t = w_1\, \mathcal{D}_t + w_2\, \Delta_t^{\text{conf}} + w_3\, \Delta_t^{\text{rate}}, \quad (5)$$

With weights $(w_1, w_2, w_3) = (1.0;\ ;0.5\ 0.3)$ that give teacher-student disagreement the leading role. The series $\{g_t\}$ is tested with the Page-Hinkley change test [30], [31]. Let n be the number of observations since the last reset, ℓ the observation index, and μ_ℓ the running mean of g up to observation ℓ. The cumulative statistic and its alarm criterion are computed through Equation (6).

$$U_n = \sum_{\ell \leq n}(\, g_\ell - \mu_\ell - \delta),\ \text{drift}_t = \mathbb{1}[U_n - \min_{\ell \leq n} U_\ell > \Lambda \quad (6)$$

Here $\delta = 0.005$ is the magnitude tolerance, $\Lambda = 0.50$ the alarm threshold, and $\mathbb{1}[\cdot]$ the indicator function. The alarm threshold is held during the initial warm-up phase to prevent false detections while the EWMA baseline is not yet stable. Based on these signals, the Cognition stage composes an adaptation plan according to one of two mutually exclusive triggering actions. Under the deployment action, the student is trained for every N_{trig} frames of accumulated pseudo-labels, whereas under the drift/scheduled action training is triggered when drift is detected or the maximum interval is exceeded, subject to sufficient labels and frames. The Configuration stage subsequently executes the plan into concrete actions, namely resetting the teacher weights to initialization, invoking the incremental trainer over the pending batch, marking the semantic task boundary, and archiving execution statistics.

Continual learning in the student adopts the AdaER framework [15], which consists of a memory recalling stage and a memory updating stage. The contextual cue for recalling is obtained from interference measurement. Let B_t denote the pending batch of pseudo-labels accumulated since the last update, $\mathcal{M}$ the episodic memory of capacity

600 samples with x_i as its i-th sample, and $\mathcal{L}_{\text{det}}$ the detection loss. One virtual gradient step is performed on B_t with learning rate $\eta_v = 10^{-3}$, the loss change of each memory candidate is measured, and the weights are restored to their previous state through Equation (7).

$$\theta' = \theta - \eta_v \nabla_\theta \mathcal{L}_{\text{det}}(B_t; \theta),\ \Delta\mathcal{L}_i = \max\ (0, \mathcal{L}_{\text{det}}(x_i; \theta') - \mathcal{L}_{\text{det}}(x_i; \theta)) \quad (7)$$

The score $\Delta\mathcal{L}_i$ expresses how strongly an old sample would be disturbed by the new data. The replay set is composed of three complementary components, namely R_e holding $k_e = 6$ exemplars with the highest interference (data-conflicting), R_t holding up to $k_t = 4$ exemplars from previous tasks with quotas proportional to their presence in R_e (task-conflicting), and R_c holding up to $k_c = 6$ exemplars of the least represented classes in memory (class-balanced). With r as the semantic task index and n_r the number of exemplars in R_e originating from task r, the quota of task r is formulated in Equation (8).

$$R = R_e \cup R_t \cup R_c,\ \kappa_r = \lfloor k_t \cdot \frac{n_r}{\sum_{r'} n_{r'}} \rceil \quad (8)$$

The component R_c is an extension required for AdaER to operate on object detection. Memory updating itself uses reservoir sampling [32] with an eviction criterion that maximizes class entropy. Letting N_c denote the number of objects of class c in memory and π_c its proportion, the majority class and the memory entropy are formulated in Equation (9).

$$\tilde{y} = \arg\max_c N_c,\ \mathcal{H}(\mathcal{M}) = -\sum_c \pi_c \log \pi_c,\ \pi_c = N_c / \sum_{c'} N_{c'}, \quad (9)$$

The eviction victim is therefore always drawn from class $\tilde{y}$, and among its candidates the least informative sample is selected. The student update objective combines plasticity originating from pseudo-labels with stability originating from distillation and weight anchoring [33], [34] as in Equation (10).

$$\mathcal{L} = \mathcal{L}_{\text{det}}(P_t) + \lambda_{\text{KD}}\, \mathcal{L}_{\text{KD}}\big(\theta; \theta_{\text{snap}}; \mathcal{I}_t\big) + \lambda_{\text{SP}} \| \theta - \theta_{\text{base}} \|^2 \quad (10)$$

Here θ_{snap} denotes the weight snapshot frozen at the previous task boundary, θ_{base} the pre-trained base weights, $\lambda_{\text{KD}} = 1.0$ the distillation term weight, and λ_{SP} the L2-SP term weight. The distillation term is computed at the anchor level. Let a be the anchor index, z_a^{new} and z_a^{old} the class logits of the new model and the snapshot model, b_a^{new} and b_a^{old} their box predictions, $p_a^{\text{old}} = \sigma(z_a^{\text{old}})$ the class probabilities of the old model used as soft targets, and $m_a \in \{0, 1\}$ the indicator that the center of anchor a falls inside the IGNORE region. The included anchors and their weights are set as $\text{sel}_a = \max(\mathbb{1}[\max_c p_{ca}^{\text{old}} \geq \tau_{\text{KD}}], m_a)$ and $w_a = \text{sel}_a[1 + (\gamma_{\text{ign}} - 1)m_a]$ with $\tau_{\text{KD}} = 0.25$ and reinforcement factor $\gamma_{\text{ign}} = 3.0$ formulated in Equation (11).

$$\mathcal{L}_{\text{KD}} = \frac{\sum_a w_a\, [\bar{\text{BCE}}(z_a^{\text{new}}, p_a^{\text{old}}) + w_{\text{box}}\ \bar{\text{SmoothL1}}(b_a^{\text{new}}, b_a^{\text{old}})]}{\max(\sum_a \text{sel}_a, 1)} \quad (11)$$

Here $\bar{\text{BCE}}$ and $\bar{\text{SmoothL1}}$ denote averaging over the class and coordinate dimensions, respectively, with $w_{\text{box}} = 0.5$. Through this mechanism, the background pull still imposed on uncertain regions is counterbalanced by the distillation pull from the old weights, so the IGNORE region acts as compensation for teacher recall leakage. The weights used for inference are not the raw post-update weights but their exponential moving average. Letting u denote the number of gradient steps since the last task boundary, the update follows Equation (12).

$$\theta_{\text{EMA}} \leftarrow d_u\, \theta_{\text{EMA}} + (1 - d_u)\, \theta,\ d_u = \min\ (d_{\max}, \frac{1+u}{2+u}) \text{ untuk } u \leq K_{\text{ema}} \quad (12)$$

Here $d_{\max} = 0.98$ and the warm-up length is $K_{\text{ema}} = 20$. It should be emphasized that (12) is executed per gradient step and its warm-up is counted per task rather than cumulatively. This provision matters because evaluation is performed on θ_{EMA}. Additional stability safeguards include freezing the backbone and BatchNorm for online micro-batches, gradient clipping, learning-rate warm-up, and a rollback guard that compares losses on a frozen checkpoint set containing both old and new classes at each task boundary.

The overall processing flow of a single frame is summarized in Algorithm 1. Note that the student prediction is computed once per frame and reused by the operational output, the pseudo-label fusion, and Monitoring, so the additional inference cost on the adaptation pathway originates only from the teacher.

Algorithm 1 Processing of a single frame in the proposed architecture

Input: frame t; weights θ, θ_{EMA}; memory $\mathcal{M}$; pending batch B_t

1: $\mathcal{S}_t \leftarrow$ student inference using θ_{EMA}

2: adapt the teacher with $\mathcal{L}_{\text{TTA}}$ in Eq. (1)

3: $\mathcal{D} \leftarrow$ multi-view teacher inference fused via WBF

4: $(P_t, \mathcal{J}_t) \leftarrow$ tiered filtering over $\mathcal{D}$ using Eqs. (2)–(3)
5: $(P_t, \mathcal{J}_t) \leftarrow$ fusion with $\mathcal{S}_t$ (agreement/conflict/missed object)
6: if the frame passes the quality gate and $P_t \neq \emptyset$ then $B_t \leftarrow B_t \cup \{(P_t, \mathcal{J}_t)\}$
7: issue operational alerts from $\mathcal{S}_t$
8: compute $\mathcal{D}_t$, Δconf_t, Δrate_t, then g_t with Eqs. (4)–(5)
9: $\text{drift}_t \leftarrow$ Page–Hinkley test over g_t with Eq. (6)
10: if the plan includes teacher adaptation then reset the teacher weights to their initialization
11: if the plan includes a student update then
12: compute $\Delta\mathcal{L}_i$ for memory candidates with Eq. (7)
13: $R \leftarrow R_e \cup R_t \cup R_c$ with Eq. (8)
14: for each chunk of B_t, perform one gradient step on $\mathcal{L}$ in Eqs. (10)–(11)
15: update θ_{EMA} with Eq. (12)
16: run the rollback guard, insert B_t into $\mathcal{M}$ with Eq. (9), then $B_t \leftarrow \emptyset$

4. Experiment

The experiment begins by building the initial student model (YOLO26s) trained on the KITTI dataset [35][36]. We split the dataset's 7481 training images using an 80/20 ratio, with 80% for training and 20% for validation. We trained the student model with a learning rate of 8.33×10^{-4}, momentum of 0.9, an image size of 640 × 640, 8 dataloader workers, and 60 epochs using the AdamW optimizer on an NVIDIA GeForce RTX 4090 24GB. Figure 2 shows the training results.

Table 2. Comparison of the student model with other detector methods for autonomous vehicles

Model	Precission	Recall	mAP_{50}	mAP_{50-95}	C
WOG-YOLO [20]	**93.7**	**87.8**	**94.2**	-	3
SNCE-YOLO [8]	93.4	85.5	91.9	62.9	3
DCW-YOLO [11]	88.0	86.6	-	69.3	**8**
Our Student (YOLO26s)	90.5	85.7	91.9	**71.0**	**8**

As shown in Figure 2, YOLO26s performs well with mAP_{50} and mAP_{50-95} of approximately 71.0% and 91.9%, even exceeding several other detector methods listed in Table 2. Although WOG-YOLO [20] achieves 94.2% on mAP_{50}, that result was measured on only three classes. In addition to comparing against other detectors that use KITTI as a benchmark, we evaluated the student model against several earlier YOLO models of the s (small) series. Table 3 presents this comparison, showing that YOLO26s is not the most optimal model, yet its GFLOPs value of 20.8 is the lowest among the compared YOLO models. This supports compatibility for deployment on the low-end devices that will host this student model.

Table 3. Performance comparison of the student model (YOLO26s) with other YOLO versions on the KITTI dataset

Model	Parameters (M)	GFLOPs	mAP_{50}	mAP_{50-95}
YOLOv5s [37]	**9,125,288**	24.1	91.8	69.8
YOLOv8s [38]	11,138,696	28.7	92.2	70.4
YOLOv9s [39]	7,290,504	27.4	91.5	69.6
YOLOv10 [40]	8,072,544	24.8	91.8	70.7
YOLO11s [41]	9,430,888	21.6	**92.4**	**71.5**
YOLO12s [42]	9,233,976	21.2	91.4	69.3
YOLO26s	9,468,276	**20.8**	91.9	71.0

Next, we tested NGABUDI's adaptability to various domain shifts that may occur when the detection model is deployed in the real world. We used Equation (13) to engineer possible domain shifts on the incoming input. Figure 3 shows the results of these domain shifts.

$$x' = clip(\alpha \cdot 255 \cdot (x / 255)\text{^}\gamma + \beta + n, 0, 255), \quad n \sim N(0, \sigma^2) \qquad (13)$$

Figure 3 decomposes the shift into several components according to Equation (13). Panel (b) shows that gamma correction removes shadow contrast before highlights, causing dark vehicles to blend into the asphalt. Panel (c) compresses the dynamic range without shifting the black point. Panel (d) is the only component that pushes the black point below zero and therefore clips information. Panel (e) attenuates the signal with noise, rendering small and distant objects invisible. Panel (f) combines the four, with mean luminance falling from 0.448 to 0.140 and RMS contrast from 0.213 to 0.121. Images in panel (f) are pseudo-labelled and used as training data to adapt the student detector.

Table 4. Parameters used for restoration

Stage	**Parameter**	**Value**
Gate	Luminance lower/upper	0.30/0.68
Gate	RMS contrast	0.11
Gamma	Target L_{target}	0.45
Gamma	Bounds of γ_r	0.30–1.60
CLAHE	Tile size	8 × 8
CLAHE	Base clip/gain/maximum	1.6/2.4/ 4.0
CLAHE	Target C_{target}	0.20
Denoise	σ_{eff}	0.030
Denoise	Diameter$\sigma_{space}/\sigma_{color}$ multiplier	5/7/2.5

To test how well NGABUDI adapts to changing visual conditions, we transformed 5985 KITTI images using Equation (13) and used them as an unlabeled adaptation stream. We also transformed 1496 validation images in the same way to create

the target domain for evaluation. To avoid overlap, we made sure that none of the image stems in the evaluation set appeared in the adaptation stream. During adaptation, we applied photometric restoration on the teacher pathway to help form pseudo-labels when the input condition was outside the normal range. This process relies only on the statistics of the current frame and does not use clean images or the shift parameters from Equation (13). Restoration includes adaptive gamma correction, CLAHE on the luminance channel [43], and bilateral filtering [44] applied conditionally based on noise estimated with the Immerkaer method [45]. The module activates if the mean luminance is outside 0.30 to 0.68 or if the RMS contrast is below 0.11. Table 4 lists all the parameters used for restoration in the experiment.

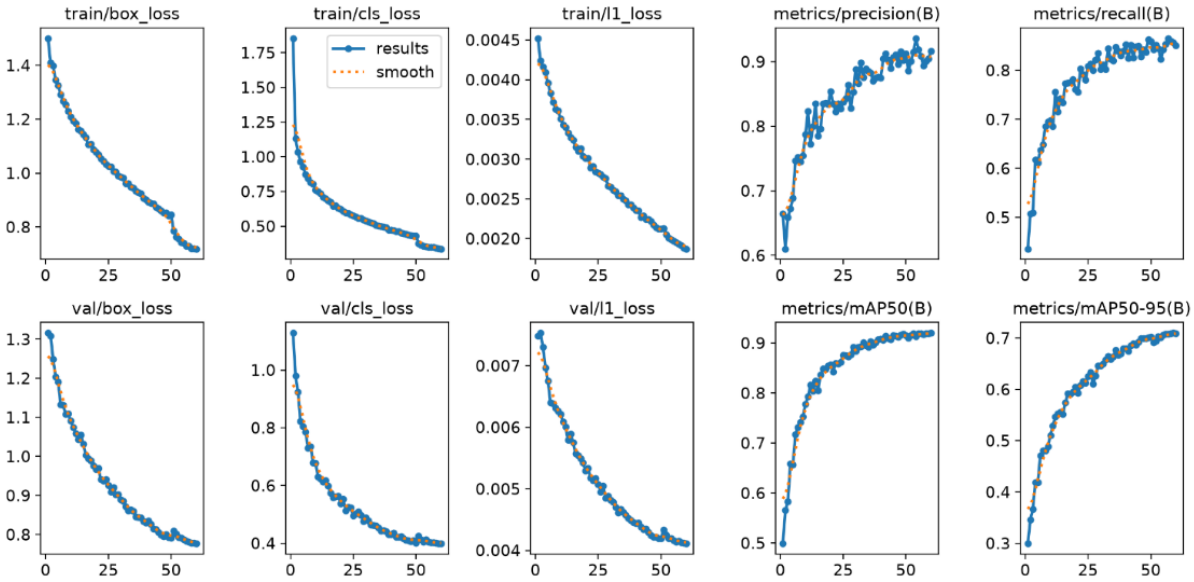

Figure. 2 Training results of the student ($YOLO26_s$)

Figure. 3 Domain shift results produced by Equation (13)

Received: October 18, 2021. Revised: December 20, 2021.

Table 5. Adaptation performance of NGABUDI on the photometrically shifted KITTI domain

Checkpoint	Images processed	Updates	Gradient steps	Precision	Recall	mAP_{50}	mAP_{50-95}
Source domain	0	0	0	0.9051	0.8569	0.9189	0.7122
Frozen model	0	0	0	0.7525	0.5552	0.5890	0.3946
Frozen model + restoration	0	0	0	0.7692	0.6064	0.6476	0.4441
Adaptation 1	498	23	70	0.7570	0.6104	0.6457	0.4122
Adaptation 2	996	46	139	0.7676	0.5978	0.6457	0.4141
Adaptation 3	1.494	70	211	0.7924	0.5947	0.6528	0.4149
Adaptation 4	1.992	93	280	0.8013	0.5858	0.6603	0.4174
Adaptation 5	2.490	117	352	0.7779	0.5829	0.6427	0.4044
Adaptation 6	2.988	141	424	0.7367	0.6006	0.6390	0.4009
Adaptation 7	3.486	164	493	0.7611	0.6058	0.6533	0.4112
Adaptation 8	3.984	188	565	0.7701	0.5888	0.6516	0.4141
Adaptation 9	4.482	212	637	0.7800	0.5799	0.6469	0.4116
Adaptation 10	4.980	235	706	0.7647	0.5881	0.6420	0.4075
Adaptation 11	5.478	258	775	0.7656	0.5896	0.6464	0.4117
Final adaptation	5.985	281	844	0.7834	0.5959	0.6576	0.4189

Restoration is used only on the teacher pathway during adaptation, while the student continues performing inference on frames without additional preprocessing. As a reference, we also evaluate the frozen model + restoration configuration to separate the influence of input improvement from NGABUDI's knowledge updating. The comparison in Table 5 separates the contribution of photometric restoration from NGABUDI's adaptive learning.

Photometric shift lowers the student mAP_{50} from 91.9% in the source domain to 58.90% in the target domain, and recall drops from 85.7% to 55.52%. Applying photometric restoration to the frozen model increases mAP_{50} to 64.76% and recall to 60.64%. This shows that normalizing input statistics helps reduce some of the domain shift, though it cannot recover lost visual information. After processing 5,985 unlabeled frames with 281 updates and 844 gradient steps, NGABUDI achieves an mAP_{50} of 65.76% and mAP_{50-95} of 41.89%. These are improvements of 6.86 and 2.43 points over the frozen model. This means about 20.9% of the mAP_{50} gap is recovered without manual annotation in the target domain. The final mAP_{50} is also 1.00 point higher than the frozen model with restoration, showing that the student weights absorb the adapted knowledge instead of relying only on preprocessing during inference. Over 12 checkpoints, mAP_{50} ranges from 63.90% to 66.03%, with an average of 64.87%. There is no downward trend up to 844 gradient steps, so the adaptation process remains stable.

Next, to test whether adaptation compromises source-domain capability, we re-evaluated the base and adapted models on image 400 under the various shifts shown in Table 6. Overall, the adapted model's performance remains very close to that of the base model under clean and single-perturbation conditions, with relatively small changes. Small gains appear under the gamma, brightness, and noise conditions, whereas small drops occur under clean and contrast. The most prominent difference arises under the combined shift condition, where the adapted model achieves the largest gain of +0.0786 mAP_{50}, showing that NGABUDI adaptation is most effective on target domains that actually contain combined shifts (the delta values are shown in Figure 4). These results indicate that the stabilization mechanisms of distillation, weight anchoring, and experience replay in AdaER can limit catastrophic forgetting during adaptation.

.

Table 6. Performance of the base and adapted models on each shift component.

Input condition	Base model	Adapted model	Difference
No shift (source domain)	0.9272	0.9267	−0.0005
Gamma (γ = 1.64)	0.9066	0.9131	+0.0065
Contrast (α = 0.88)	0.9265	0.9236	−0.0029
Brightness (β = −14)	0.9260	0.9281	+0.0021
Noise (σ = 4.8)	0.9052	0.9148	+0.0096
Combined (s = 0.4)	0.5977	0.6763	+0.0786

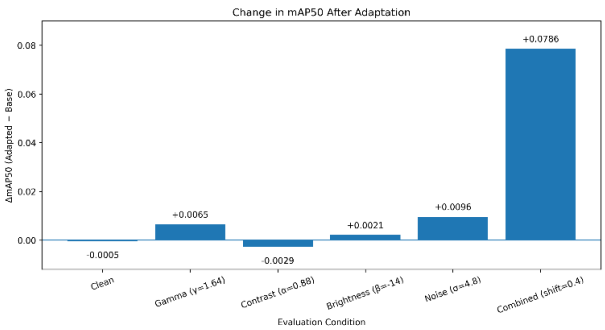

Figure 4. Comparison of ΔmAP_{50} between the base model (Base/Frozen) and the adapted model (Adapted/Ours) under the six photometric evaluation conditions.

Received: October 18, 2021. Revised: December 20, 2021.

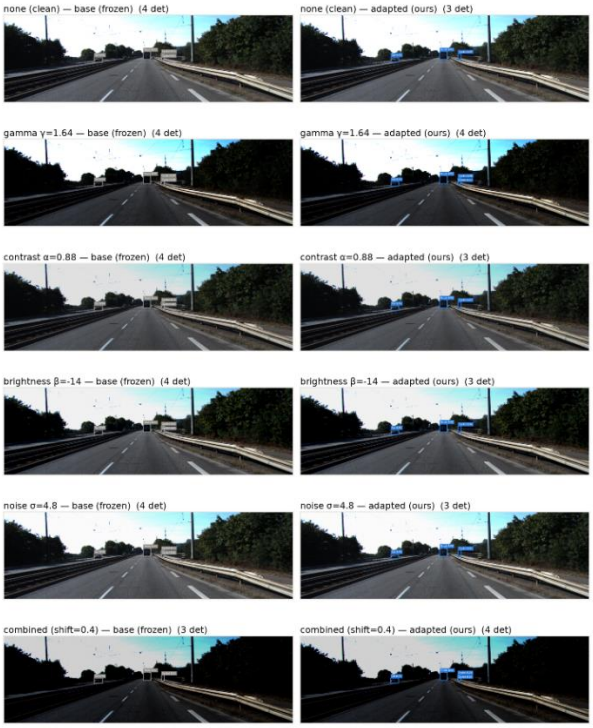

Figure 5. Qualitative comparison of detection results between the base model (base/frozen) and the adapted model (adapted/ours) under clean, gamma, contrast, brightness, noise, and combined shift conditions. The adapted model maintains a similar detection pattern under most conditions and shows better recovery of detections under the combined shift.

To identify each component's contribution to overall performance, we conducted an ablation study by removing one component in each configuration and comparing the results against the full NGABUDI. We evaluated the photometrically shifted KITTI evaluation data using precision, recall, and mAP50, and used the change in mAP50 to quantify each component's impact on final performance. The results of this experiment are presented in Table 7.

The ablation results reveal that each part of NGABUDI affects its final performance in different ways. Removing the anchor source causes the largest drop (0.1120 mAP_{50}), while removing photometric restoration (0.0858) and AdaER memory (0.0809) also lowers performance. Distillation and the learning-rate schedule have notable impacts too, with drops of 0.0727 and 0.0665. On the other hand, removing the rollback guard only reduces mAP_{50} by 0.0005, which suggests it mainly serves as a safety feature rather than a key factor in performance. Figure 6 compares the effects of removing each NGABUDI component. Overall, these findings show that NGABUDI relies most on source-domain information, photometric restoration, and the AdaER-based continual learning mechanism.

Table 7. Ablation study

Configuration	**P ↑**	**R ↑**	**mAP@0.5 ↑**
NGABUDI (full)	0.7939	0.5769	**0.6506**
† w/o anchor source	0.6665	0.5075	0.5386
† w/o photometric restoration	0.7458	0.5307	0.5648
w/o AdaER memory ($\mathcal{M}$)	0.7894	0.6064	0.5697
† w/o distillation ($\mathcal{L}_{KD}$)	0.7145	0.5677	0.5779
w/o learning-rate schedule ($\eta \equiv const$)	0.8205	0.6037	0.5841
Control: student updates disabled	0.7525	0.5552	0.5890
w/o cross-domain input to $\mathcal{L}_{KD}$	0.7599	0.5353	0.5962
w/o C-CMR selection	0.7693	0.5972	0.5985
w/o $\mathcal{L}_{det}$ decay ($\lambda_{det} \equiv 1$)	0.7275	0.5614	0.6091
w/o BN statistics adaptation	0.7648	0.5957	0.6148
Adaptation scope: head only	0.7769	0.5596	0.6204
w/o two-source gating ($n_{src} \geq 2$)	0.7293	0.5895	0.6240
w/o rollback guard	0.7605	0.5925	0.6501

5. Conclusion

This study proposes NGABUDI as a self-learning vision system for autonomous vehicles that integrates YOLO26s based real-time detection, pseudo-label generation from unlabeled data, AdaER based continual learning, and SACPS adaptation control within a single end-to-end framework. On the source domain, YOLO26s achieves an mAP_{50} of 91.9% and mAP_{50-95} of 71.0% with a computational cost of 20.8 GFLOPs. Under photometric shift, the frozen-model mAP_{50} drops to 58.90%. After processing 5,985 unlabeled frames through 281 updates and 844 gradient steps, NGABUDI increases mAP_{50} to 65.76% and mAP_{50-95} to 41.89% without using manual annotations in the target domain.

Cross-condition evaluation shows that adaptation preserves source-domain performance, with mAP_{50} changing only from 92.72% to 92.67%, while under the combined shift performance improves from 59.77% to 67.63%. The ablation study shows that the anchor source, photometric restoration, AdaER memory, and distillation are the components that most influence adaptation effectiveness. These results indicate that NGABUDI is able to improve perception capability on the target distribution while limiting catastrophic forgetting of previously acquired knowledge.

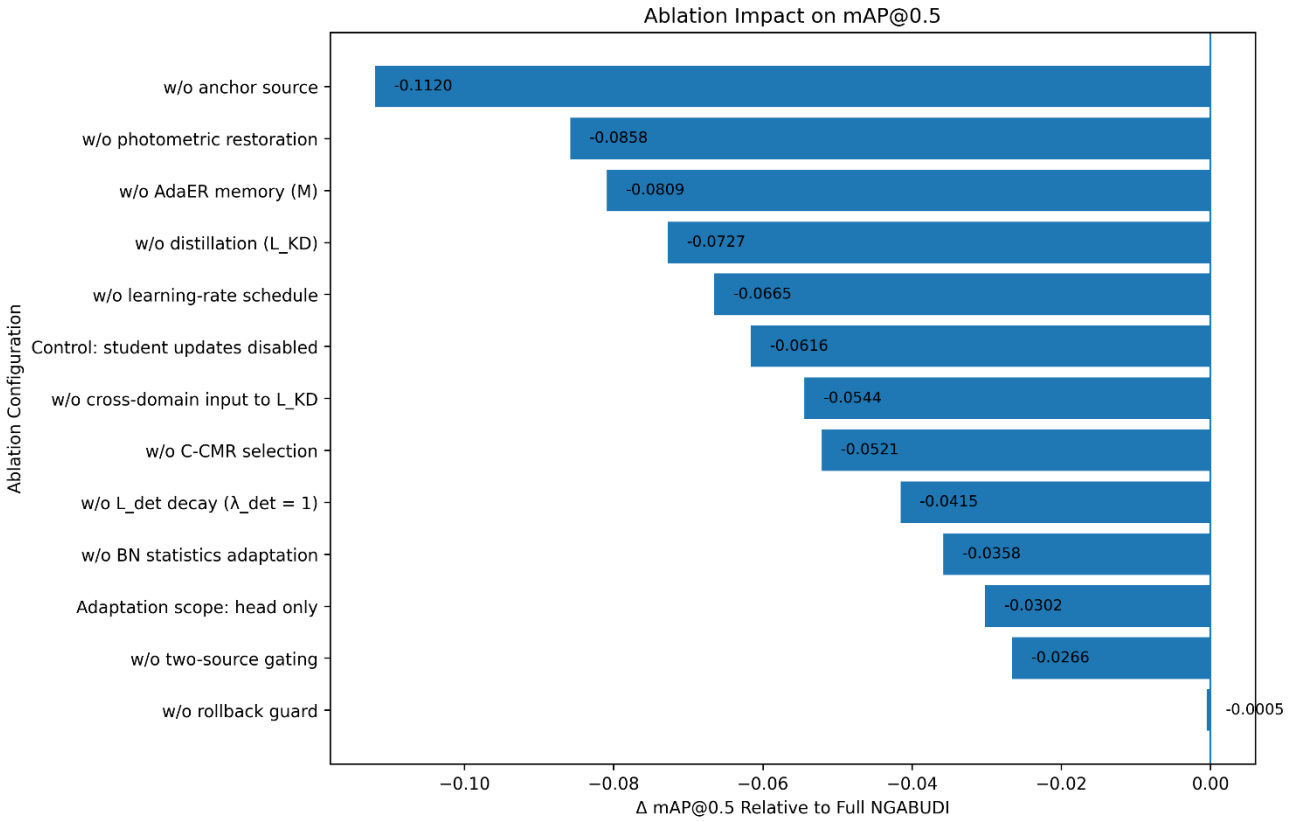

Figure 6. Impact of removing each NGABUDI component on mAP@0.5 relative to the full configuration.

Nevertheless, this study still has several limitations. All experiments were conducted at laboratory scale using the KITTI dataset and controlled synthetically generated photometric domain shifts, so the system has not yet been tested directly on a vehicle or on real-world sensor streams. Operational conditions such as weather changes, time of day, location, camera characteristics, vehicle vibration, traffic dynamics, and sensor disturbances are not fully represented. In addition, the overall computational cost of the teacher pathway and the student update process has not been evaluated on real in-vehicle hardware. Therefore, future work should carry out in-vehicle or on-road validation, extend testing to real-world domain shifts and more diverse sensors, and evaluate accuracy, latency, memory consumption, and adaptation stability during continuous operation.

Conflicts of interest

Authors must declare any conflicts of interest or explicitly state, “The authors declare no conflict of interest.” Any personal circumstances or interests that could be perceived as having an undue influence on the presentation or interpretation of the research findings must be clearly disclosed.

Author contributions

Methodology, investigation, formal analysis, validation, Aradea; Conceptualization, methodology, reviewing & supervision, Rianto; Data curation, self-Learning vision system, model evaluation, Zeni Muhamad Noer; Resources, formal analysis, validation, Ghatan Fauzi Nugraha; software development, writing original draft preparation, Pandu Pangestu; preprocessing data, visualization data.

Acknowledgments

The present study is supported by the Institute for Research and Community Services of Siliwangi University, and the Research Group of Intelligent Systems and Informatics (ISI) at the University of Siliwangi. Likewise, this study is a manifestation of the strategic program of The Ministry of Higher Education, Science, and Technology of the Republic of Indonesia (No. 265/C3/DT.05.00/PL-BARU/2026) related to the research developments in Indonesia.

References

[1] L. Xu, “Data Shift of Object Detection in Autonomous Driving,” *arXiv preprint*

arXiv:2508.11868, May 2025, [Online]. Available: http://arxiv.org/abs/2508.11868

[2] J. Yoo, D. Lee, I. Chung, D. Kim, and N. Kwak, "What, How, and When Should Object Detectors Update in Continually Changing Test Domains?," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, May 2024, pp. 23354–23363. doi: 10.1109/CVPR52733.2024.02204.

[3] D. Hendrycks and T. Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," Mar. 2019, [Online]. Available: http://arxiv.org/abs/1903.12261

[4] B. R. Lamichhane, A. Aueawatthanaphisut, G. Srijuntongsiri, and T. Horanont, "Context-aware decision making in autonomous vehicles: Integrating social behavior modeling with large language models," *Array*, vol. 27, Sep. 2025, doi: 10.1016/j.array.2025.100420.

[5] N. M. Alahdal, F. Abukhodair, L. H. Meftah, and A. Cherif, "Real-time Object Detection in Autonomous Vehicles with YOLO," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 2792–2801. doi: 10.1016/j.procs.2024.09.392.

[6] S. Yue, Z. Zhang, Y. Shi, and Y. Cai, "WGS-YOLO: A real-time object detector based on YOLO framework for autonomous driving," *Computer Vision and Image Understanding*, vol. 249, p. 104200, Dec. 2024, doi: 10.1016/J.CVIU.2024.104200.

[7] B. Khalili and A. W. Smyth, "SOD-YOLOv8 - Enhancing YOLOv8 for Small Object Detection in Traffic Scenes," *ArXiv*, Aug. 2024, [Online]. Available: http://arxiv.org/abs/2408.04786

[8] F. Li, Y. Zhao, J. Wei, S. Li, and Y. Shan, "SNCE-YOLO: An Improved Target Detection Algorithm in Complex Road Scenes," *IEEE Access*, vol. 12, pp. 152138–152151, 2024, doi: 10.1109/ACCESS.2024.3481642.

[9] Z. Lv, R. Wang, Y. Wang, F. Zhou, and N. Guo, "Road Scene Multi-Object Detection Algorithm Based on CMS-YOLO," *IEEE Access*, vol. 11, pp. 121190–121201, 2023, doi: 10.1109/ACCESS.2023.3327735.

[10] S. Cui *et al.*, "DAN-YOLO: A Lightweight and Accurate Object Detector Using Dilated Aggregation Network for Autonomous Driving," *Electronics (Switzerland)*, vol. 13, no. 17, May 2024, doi: 10.3390/electronics13173410.

[11] H. Ren, F. Jing, and S. Li, "DCW-YOLO: Road Object Detection Algorithms for Autonomous Driving," *IEEE Access*, vol. 13, pp. 125676–125688, 2025, doi: 10.1109/ACCESS.2024.3364681.

[12] W. Chen, J. Yan, W. Huang, W. Ge, H. Liu, and H. Yin, "Robust object detection for autonomous driving based on semi-supervised learning," *Security and Safety*, vol. 3, 2024, doi: 10.1051/sands/2024002.

[13] R. Sathyam and Y. Li, "Foundation Models for Autonomous Driving Perception: A Survey Through Core Capabilities," *IEEE Open Journal of Vehicular Technology*, vol. 6, pp. 2554–2582, May 2025, doi: 10.1109/OJVT.2025.3604823.

[14] G. Jocher and J. Qiu, "Ultralytics YOLO26," 2026. [Online]. Available: https://github.com/ultralytics/ultralytics

[15] X. Li, B. Tang, and H. Li, "AdaER: An adaptive experience replay approach for continual lifelong learning," *Neurocomputing*, vol. 572, p. 127204, May 2024, doi: 10.1016/j.neucom.2023.127204.

[16] A. Aradea, I. Darmawan, R. Rianto, H. Mubarok, and G. F. Nugraha, "A Self-Adaptive CPS-Based Object Recognition Framework for Smart Glasses Using Dist-YOLOv3-Xception with Attention," *Informatica*, vol. 49, no. 15, Nov. 2025, doi: 10.31449/inf.v49i15.8853.

[17] L. Liu *et al.*, "Deep Learning for Generic Object Detection: A Survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020, doi: 10.1007/s11263-019-01247-4.

[18] F. Wei and W. Wang, "SCCA-YOLO: A Spatial and Channel Collaborative Attention

Enhanced YOLO Network for Highway Autonomous Driving Perception System," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-90743-4.

[19] S. Kang, Z. Hu, L. Liu, K. Zhang, and Z. Cao, "Object Detection YOLO Algorithms and Their Industrial Applications: Overview and Comparative Analysis," Mar. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/electronics14061104.

[20] L. Xu, W. Yan, and J. Ji, "The research of a novel WOG-YOLO algorithm for autonomous driving object detection," *Sci. Rep.*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-30409-1.

[21] Y. Cao, C. Li, Y. Peng, and H. Ru, "MCS-YOLO: A Multiscale Object Detection Method for Autonomous Driving Road Environment Recognition," *IEEE Access*, vol. 11, pp. 22342–22354, 2023, doi: 10.1109/ACCESS.2023.3252021.

[22] Y. Yang, S. Yang, and Q. Chan, "LEAD-YOLO: A Lightweight and Accurate Network for Small Object Detection in Autonomous Driving," *Sensors*, vol. 25, no. 15, Aug. 2025, doi: 10.3390/s25154800.

[23] E. Liang, D. Wei, F. Li, H. Lv, and S. Li, "Object detection model of vehicle-road cooperative autonomous driving based on improved YOLO11 algorithm," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-18263-9.

[24] Z. Jia, S. Sun, G. Liu, and B. Liu, "MSSD: multi-scale self-distillation for object detection," *Visual Intelligence*, vol. 2, no. 1, Dec. 2024, doi: 10.1007/s44267-024-00040-3.

[25] S. Cui *et al.*, "DAN-YOLO: A Lightweight and Accurate Object Detector Using Dilated Aggregation Network for Autonomous Driving," *Electronics (Switzerland)*, vol. 13, no. 17, Sep. 2024, doi: 10.3390/electronics13173410.

[26] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "YOLO-World: Real-Time Open-Vocabulary Object Detection," Feb. 2024, [Online]. Available: http://arxiv.org/abs/2401.17270

[27] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully Test-time Adaptation by Entropy Minimization," Mar. 2021, [Online]. Available: http://arxiv.org/abs/2006.10726

[28] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," Feb. 2021, doi: 10.1016/j.imavis.2021.104117.

[29] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, 2012, doi: 10.1109/TPAMI.2011.155.

[30] E. S. Page, "Continuous Inspection Schemes," *Biometrika*, vol. 41, no. 1/2, p. 100, Jun. 1954, doi: 10.2307/2333009.

[31] D. V Hinkley, "Inference about the change-point from cumulative sum tests," 1971. [Online]. Available: https://about.jstor.org/terms

[32] J. S. Vitter, "Random Sampling with a Reservoir," *ACM Transactions on Mathematical Software (TOMS)*, vol. 11, no. 1, pp. 37–57, Mar. 1985, doi: 10.1145/3147.3165;WGROUP:STRING:ACM.

[33] Z. Li and D. Hoiem, "Learning without Forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018, doi: 10.1109/TPAMI.2017.2773081;TAXONOMY:TAXONOMY:ACM-PUBTYPE;PAGEGROUP:STRING:PUBLICATION.

[34] X. Li, Y. Grandvalet, and F. Davoine, "Explicit Inductive Bias for Transfer Learning with Convolutional Networks," *35th International Conference on Machine Learning, ICML 2018*, vol. 6, pp. 4408–4419, Feb. 2018, Accessed: Aug. 23, 2026. [Online]. Available: https://arxiv.org/pdf/1802.01483

[35] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

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

[37] G. Jocher *et al.*, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation ," Nov. 2022, *Zenodo*. doi: 10.5281/zenodo.7347926.

[38] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[39] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *ArXiv*, Feb. 2024, [Online]. Available: http://arxiv.org/abs/2402.13616

[40] A. Wang *et al.*, "YOLOv10: Real-Time End-to-End Object Detection," *ArXiv*, Oct. 2024, [Online]. Available: http://arxiv.org/abs/2405.14458

[41] G. Jocher and J. Qiu, "Ultralytics YOLO11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[42] M. A. R. Alif and M. Hussain, "YOLOv12: A Breakdown of the Key Architectural Features," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2502.14740

[43] K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," *Graphics Gems*, pp. 474–485, Jan. 1994, doi: 10.1016/B978-0-12-336156-1.50061-6.

[44] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 839–846, 1998, doi: 10.1109/ICCV.1998.710815.

[45] J. Immerkær, "Fast Noise Variance Estimation," *Computer Vision and Image Understanding*, vol. 64, no. 2, pp. 300–302, Sep. 1996, doi: 10.1006/CVIU.1996.0060.