AutoDeep: Automatic Design of Safe, High-Performance and Compact Deep Learning Models for Autonomous Vehicles

Public Report

Project within: FFI - Electronics, software and communication

Author: Masoud Daneshtalab

Date: 2025-8-12

Table of Contents:

1. Summary in English

Deep Neural Networks (DNN) are increasingly being used to support decision-making in autonomous vehicles. While DNN holds the promise of delivering valuable results in safety-critical applications, broad adoption of DNN systems will rely heavily on how the computation-intensive DNN could be customized and deployed on the resource-limited vehicle embedded hardware platform and also how much to trust their outputs. High DNN accuracy comes at high computations, storage, and memory bandwidth requirements, which makes their deployment particularly challenging, especially for vehicle embedded computing platforms. Therefore, an efficient method is required to optimize the network complexity by efficiently exploring the design space. In addition, to trust a decision made by a DNN in an autonomous vehicle, we need assurance that it is robust in its computations and predictions and will cause no harm in real-world unintended data perturbations. There have been a lot of research on improving the accuracy of DNNs. However, nowadays, building just for accuracy is not sufficient as the DNN's main design concern. We must know how to design a compact and accurate DNN, how to optimize the computation-intensive DNN efficiently, and also how to model and evaluate noise and improve the DNN robustness for safety-critical autonomous vehicle applications.

In this project we develop an automatic framework called AutoDeep to achieve performance, compactness, and robustness in design and customization of DNN for safety-critical applications such as intention detection and behavior prediction for road and construction autonomous vehicles. The project consortium consists of three partners, including the main applicant Mälardalen University (MDU), Zenuity (now Zensact) and Volvo Construction Equipment (VCE).

Summary in Swedish:

Djupa neurala nätverk (DNN) används i allt större utsträckning för att stödja beslutsfattande i autonoma fordon. Även om DNN har potential att leverera värdefulla resultat i säkerhetskritiska applikationer, kommer en bred användning av DNN-system i hög grad att bero på hur de beräkningsintensiva DNN:erna kan anpassas och implementeras på de resursbegränsade inbyggda hårdvaruplattformarna i fordon, samt på hur mycket man kan lita på deras utdata. Hög DNN-precision medför höga krav på beräkningskapacitet, lagring och minnesbandbredd, vilket gör deras implementering särskilt utmanande, framför allt för inbyggda beräkningsplattformar i fordon. Därför krävs en effektiv metod för att optimera nätverkets komplexitet genom att på ett effektivt sätt utforska designutrymmet.

För att kunna lita på ett beslut som fattas av en DNN i ett autonomt fordon behöver vi dessutom försäkran om att nätverket är robust i sina beräkningar och prediktioner, och att det inte orsakar skada vid oavsiktliga dataförändringar i verkliga miljöer. Mycket forskning har bedrivits för att förbättra DNN:ers precision, men idag räcker det inte att endast bygga för hög noggrannhet som huvudfokus. Vi måste veta hur man designar ett kompakt och precist DNN, hur man effektivt optimerar ett beräkningsintensivt DNN, samt hur man modellerar och utvärderar brus och förbättrar DNN:ers robusthet för säkerhetskritiska applikationer i autonoma fordon.

I detta projekt utvecklar vi ett automatiserat ramverk, kallat AutoDeep, för att uppnå prestanda, kompakthet och robusthet vid design och anpassning av DNN för säkerhetskritiska applikationer såsom intentionsdetektion och beteendeprediktion för väg- och anläggningsmaskiner med autonom körning. Projektkonsortiet består av tre parter: huvudsökande Mälardalens universitet (MDU), Zenuity (numera Zenseact) och Volvo Construction Equipment (VCE).

,

2. Background and project description

Over the years, vehicle safety standards have become more stringent, e.g., today's cars and heavy equipment are safer than before by preventing crashes from happening and also by minimizing injuries and economic costs in the crashes. Yet the reports show that 94% of all crashes on the road are caused by human factors [1]. Also, in heavy equipment and earth moving industry the human factor has the most contribution in degrading system safety. Harsh working conditions and dangerous environments can affect operator's performance negatively due to fatigue which in turn leads to increased operational faults and hence is a common source of accidents that degrades safety conditions. Today there are specific institutes for vehicle safety administrations to keep people safe and reduce deaths, injuries and economic losses from vehicle crashes. For example, Euro New Car Assessment Program (EuroNCAP) exploits car assessment programs and introduces the 5-star safety ratings to provide consumers with information about the safety of new vehicles. According to the reports [2], many accidents are caused by the late driver reactions. A driver may react too late in situations that are very difficult to predict. Most people are not used to deal with such critical situations and do not make fast decisions to avoid the crash or there is not sufficient time to react. According to EuroNCAP, one of the most common aspects of safety is crash avoidance and mitigation. This aspect is considered in terms of how well a smart vehicle can prevent a crash or lessen its severity by means of accurate object detection and robust decision making. Artificial intelligence and advanced learning-based approaches have shown the capacity to model complex processes by learning and imitating them efficiently [3]. In the years and decades ahead, we need enhancement of efficient learning-based approaches to elevate the development of complex decision making that is required to be taken in time for safe autonomous vehicles. This project targets safety-critical applications such as intention detection of pedestrian, cyclist, and operator along with behaviour prediction of surrounding vehicle towards enabling autonomous functionalities for both road and construction vehicles.

Intention detection and behavior prediction are safety-critical applications for both road and construction environments that enable safe autonomous functionalities within the vehicle [4]. Autonomous driving systems require the ability to comprehend and anticipate the actions of pedestrians and cyclists that are of particular importance and are called the most vulnerable road users (VRU), especially when crossing the road. On the other hand, for a construction vehicle to be able to autonomously navigate in the complex and dynamically changing construction site, it should be able to perceive the surrounding vehicles and site operators and predict their future behavior in the environment. Behavior prediction of a vehicle is an important enabler for the vehicle to be able to navigate safely in the environment. Anticipating the intention of pedestrians and behavior of other vehicles helps the driving systems to select the correct action to avoid any potential collisions and disruption of traffic flow.

Deep Neural Networks (DNNs) are increasingly deployed in safety-critical applications of autonomous vehicles [5] which take advantage of object action anticipation in data sequences from on-board sensors inside a moving vehicle. For this, we will design a Convolutional Recurrent Neural Network (CRNN), a combination of Convolutional Neural Network (CNN) and Recurrent NN (RNN), in which the former can efficiently detect and recognize static/dynamic objects while the latter is for intention detection and behavior prediction using camera and/or LIDAR sensors.

Such Complex learning-based safety-critical applications for intention detection and behavior prediction require a great amount of processing power and memory footprint to provide desirable accuracy which makes them not suitable to be deployed on resource-limited embedded systems of the vehicle's platform. This project aims to provide optimized CRNN models that are customized for deployment on resource-limited embedded hardware platforms of vehicles with safe autonomous functionalities. To customize CRNNs for resource-limited vehicle embedded platforms, we propose the AutoDeep framework, with the aim of automatic design and optimization of deep neural architectures using multi-objective Neural Architecture Search (NAS).

Despite the advantages of deploying DNNs in safety-critical domains like self-driving applications, it is shown that DNNs often exhibit unexpected behaviors when facing unintended real-world data perturbations [6]. Therefore, when the developers do not consider these data perturbations in the design and training of the DNN, the prediction of the DNN classifier is not trustworthy. In this project, we will cope with such unintended data perturbations in order to provide robustness given a reasonable data perturbation model and mitigate data noise in real-world fail-safe systems like autonomous vehicles. AutoDeep framework solves this problem with exploring the design space considering accuracy, network architecture complexity, and robustness as the optimization objectives. To the best of our knowledge, this is the first framework which focuses not only on improving the network accuracy, but also considers network complexity, and network robustness objectives. The project overview including the work packages are shown in Figure 1.

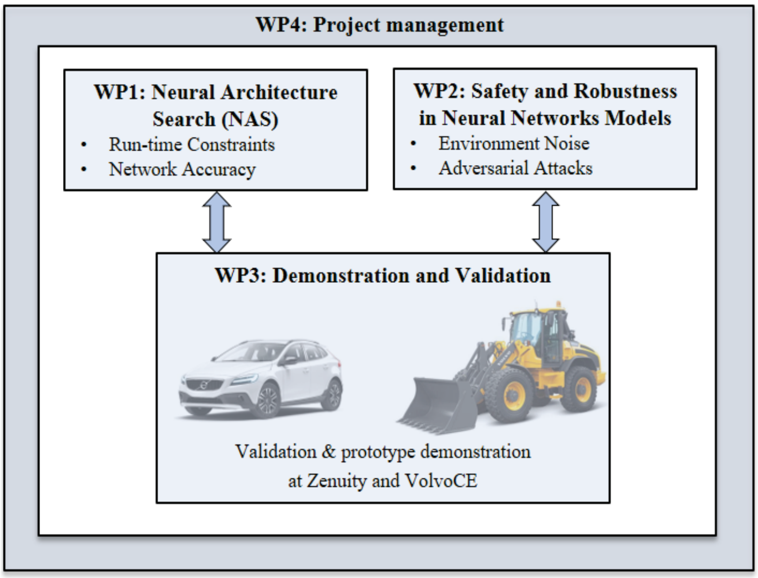

Figure 1. Project overview

3. Research challenges and objectives

3.1. Research challenges

In general, DNNs suffer from implementation difficulties:

For providing more accurate results, DNN architectures are becoming more complex models containing hundreds of deep layers and millions of floating-point operations. Thus, having too many trainable parameters negatively impacts on computational performance, particularly on embedded platforms due to the limited computing resources. High classification accuracy of DNNs comes at intensive computations and memory bandwidth requirements, which makes the DNN deployment particularly challenging for resource-limited platforms. Although there exist a variety of DNN architectures for different tasks, finding a cost-efficient architecture is still challenging due to the lack of a general designing solution. Hence, balancing a trade-off between implementation efficiency and accuracy of DNNs needs considerable efforts.

In addition, network quantization is an impressive network compression method trying to reduce the memory-footprint and computation-intensive operations of DNNs. The goal of network quantization is to represent the floating-point weights and/or activation functions with fewer bits. However, naive quantization methods lead to significant accuracy loss and unacceptable accuracy level.

For safety-critical applications, a small error can lead to catastrophic effects. Apart from accuracy level of DNNs, there is a significant need for robust machine learning systems and hardware architectures that can generate reliable and trustworthy results. It has been shown that neural networks can be vulnerable against adversarial attacks, i.e., input perturbations can cause neural networks to misclassify. To address this challenge and prove that a network is free of adversarial examples, a proper perturbation model, a good metric, and an automatic NAS framework is required for certification of the network robustness.

3.2. Project objectives

This project aims to tackle the above challenges by developing a framework, named AutoDeep, which provides resource-efficient and robust DNNs with a required performance-level for embedded computing platforms of autonomous vehicles. That is, it generates performance-efficient DNNs suitable for deployment on embedded resources-limited automotive computing platforms while enhancing the safety and robustness of DNN models considering vehicle safety standards such as EuroNCAP. AutoDeep tries to optimize the combination of complex models to efficiently detect static/dynamic objects, landmarks, and traffic signs, along with pedestrian intention detection and surrounding vehicles behavior prediction based on camera and LIDAR sensors. The key objectives of the framework are as follows:

- Providing a set of novel algorithmic techniques based on evolutionary multi-objective optimization to generate optimal end to end deep learning models in terms of accuracy, execution time, and network complexity.
- Optimizing the model with pruning and quantization techniques to reduce the huge memory footprint and bandwidth requirements and also improving the latency by designing an embedded-friendly DNN models optimized for fast and on time decision making in critical situations.
- Improving the robustness of the DNNs for safety-critical autonomous vehicle applications considering the faults in network model, the fragility in the training data, and also considering the robustness against unintended data perturbations.
- Demonstrating the effectiveness of the proposed framework in designing optimal DNNs to accurately detect the intention of moving objects

4. Results and deliverables

The project successfully achieved its intended goals, resulting in the development of advanced frameworks for designing deep neural networks (DNNs) that are efficient, compact, and robust—critical properties for deployment in autonomous vehicle (AV) systems. These frameworks were validated in two industrial use cases: (1) **trajectory prediction using 3D point cloud data from LiDAR sensors** in collaboration with Volvo Construction Equipment, and (2) **3D lane detection using camera sensors** in collaboration with Zenuity (now Zenseact). These real-world scenarios confirmed the applicability and scalability of the developed methods across diverse sensing modalities.

The project delivered all key milestones:

- **M1**: Specification of industrial requirements and use cases
- **M2**: Development of methods for compact and hardware-efficient DNNs [10][12][14]
- **M3**: Enhancement of robustness against adversarial attacks and environmental perturbations [7][13][8][9]
- **M4**: Integration and validation in simulation environments and data-driven pipelines [11][12][13][10]

In the Volvo CE use case, temporal 3D LiDAR data was used for predicting future trajectories in dynamic and unstructured environments involving heavy machinery. For Zenuity, camera-based input was utilized for lightweight and accurate 3D lane detection in real-time driving scenarios, combining perception and NAS (Neural Architecture Search) techniques.

The project outcomes have been disseminated through four Q1 journals, four top-tier conference papers, and one licentiate thesis. Highlights include:

- DASS: Differentiable Architecture Search for Sparse Neural Networks (TECS, 2023)
- ProARD: Progressive adversarial robustness distillation (IJCNN 2025)
- 3DLaneNAS: Lightweight 3D lane detection via NAS (ICANN 2022)
- CLLD: Contrastive Learning for Lane Detection with cross-similarity (Pattern, 2024)
- DAT and TrajectoryNAS: Trajectory forecasting and architecture search using LiDAR data (MDPI Q1)

All project objectives were fulfilled, and the results offer reusable building blocks and validated methods that can benefit future R&D in AI-based systems.

4.1. The Differentiable Architecture Search for Sparse Neural Networks framework (DASS)

DASS [14] is a novel framework jointly developed by MDU, Zenseact, and Volvo Construction Equipment to address the challenges of deploying DNNs on highly constrained edge hardware. Unlike conventional approaches that apply pruning after architecture design, DASS integrates sparsity-aware neural architecture search (NAS) directly into the model design phase. This enables the automated discovery of compact, accurate, and hardware-efficient models optimized for extreme compression scenarios, including pruning ratios up to 99%.

1. Sparsity-Aware Search Space Design

DASS extends the standard DARTS-based NAS framework by introducing two new operations into the search space:

- SparseConv (Figure 2a) and
- SparseLinear (Figure 2b),

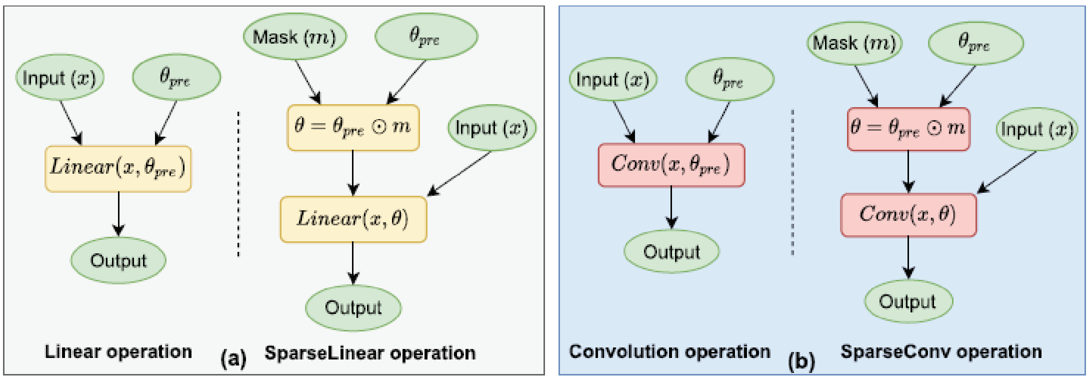

Fig. 2. Illustrating the (a) SparseLinear and (b) SparseConv operations.

which are parametric sparse variants of convolution and fully connected layers. These allow the search algorithm to explore and learn architectures that natively support unstructured sparsity, avoiding the severe performance drops observed when pruning dense backbones.

2. Joint Bi-Level Optimization

The core of DASS is a bi-level optimization strategy that jointly learns:

- Architecture parameters (α),
- Weight parameters (θ), and
- Pruning masks (m).

To address the non-triviality of this three-component problem, DASS decomposes it into a three-stage optimization pipeline (Figure 6):

1. Pre-training: Search and train the best dense architecture.
2. Pruning: Learn pruning parameters and adapt architecture accordingly.
3. Fine-tuning: Retrain the sparse model for performance recovery.

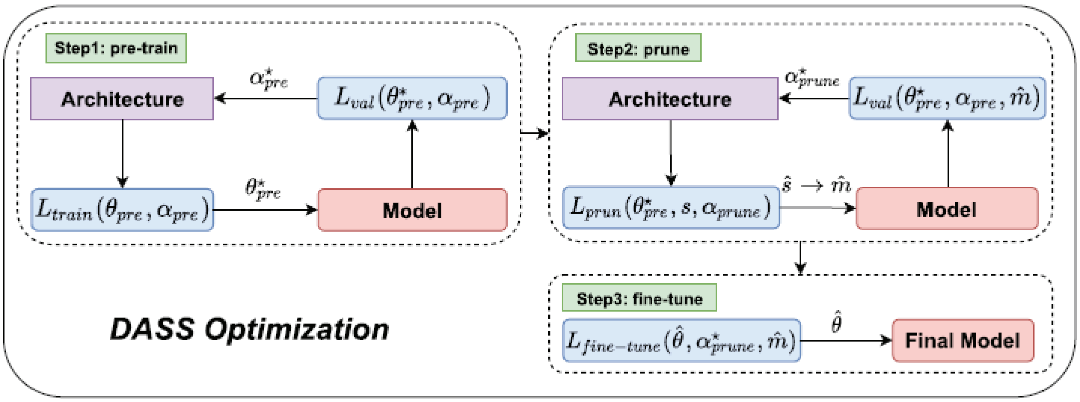

Fig. 3. The overview of the proposed optimization algorithm to find architecture parameters based on the sparse weight parameters. It consists of three main steps: (1) pre-training: search dense architecture (2) pruning: search sparse architecture (3) fine-tuning: re-train best sparse architecture.

This design ensures that the resulting architectures are structurally sparse and robust at high compression levels, unlike standard post-hoc pruning (see learning curves in Figure 4).

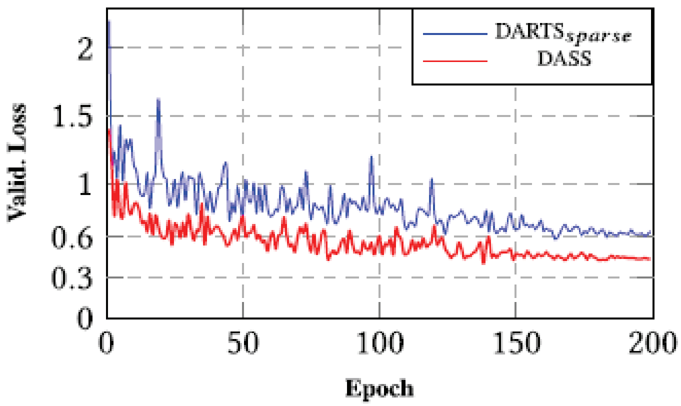

Fig. 4. Comparing learning curves (validation loss) of DASS and $\text{DARTS}_{\text{sparse}}$ on the searched architectures trained with the CIFAR-10 dataset.

3. Superior Accuracy-Compression Trade-Offs

Experimental results (see Table 1) show that DASS significantly outperforms existing dense and sparse baselines. For instance:

- DASS-Small achieves 89.1% accuracy on CIFAR-10 with only 17k parameters,

- DASS-Large achieves 73.8% top-1 accuracy on ImageNet with just 0.19M parameters, outperforming MobileNet-v2sparse and ResNet-18sparse in both accuracy and size.

Table 1. Comparing the DASS Method with Sparse Networks on the CIFAR-10 and ImageNet Datasets:

Architecture	CIFAR-10				ImageNet				
	Top-1 Acc. (%)	#Params ($\times 10^3$)	Compression Rate†	NID‡	Top-1 Acc. (%)	Top-5 Acc. (%)	#Params ($\times 10^3$)	Compression Rate†	NID‡
DARTS$_{sparse}$ [27]	81.25	21.0	100.47×	3.86	38.67	61.33	33.0	100×	1.11
MobileNet-v2$_{sparse}$ [30]	73.44	22.2	95.04×	3.30	17.97	36.72	34.87	94.63×	0.515
ResNet-18$_{sparse}$ [2]	90.62	111.6	18.90×	0.81	67.58	80.86	116.84	28.24×	0.578
EfficientNet$_{sparse}$ [31]	79.69	202.3	10.43×	0.39	-	-	-	-	-
MCUNET [8]	89.7	210.1	15.70	0.42	72.34	84.86	562.64	5.86×	0.128
DASS-Small	89.06	17.0	124.11×	5.23	46.48	68.36	28.94	114.02×	1.606
DASS-Medium	92.18	53.65	39.32×	1.71	68.34	82.24	81.95	40.26×	0.841
DASS-Large	95.31	105.5	20×	0.90	73.83	85.94	194.6	16.95×	0.38

† The baseline for comparing the compressing rate is full-precision and dense DARTS architecture.
‡ NID = Accuracy/#Parameters [86]. NID measures how efficiently each network uses its parameters.
We highlight the best results in blue color.

In terms of hardware efficiency, DASS offers up to 3.87× faster inference on CPUs and embedded GPUs, making it suitable for real-time applications on platforms like Jetson TX2 and Intel Movidius (Figure 5).

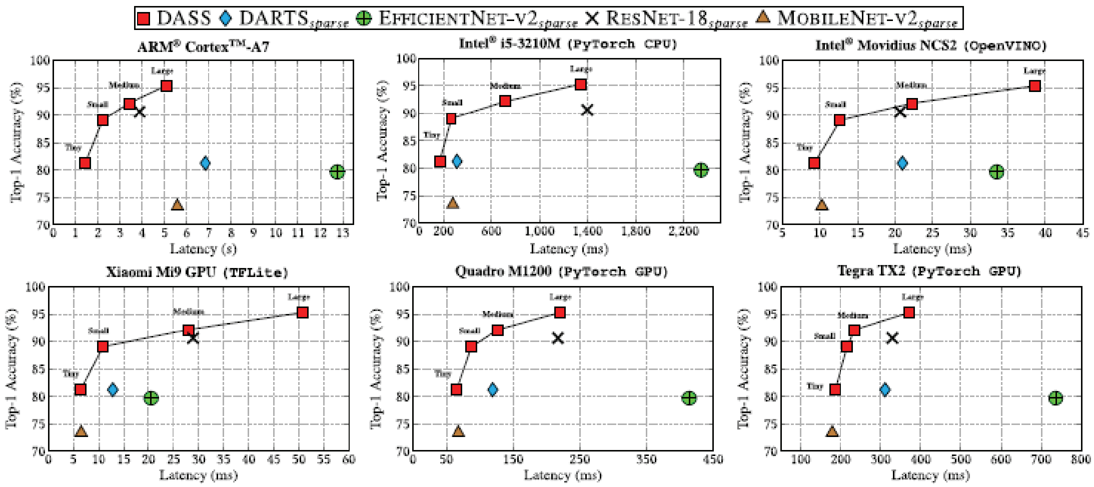

Fig. 5. Trade-off: accuracy v.s. measured latency. DASS-Tiny, DASS-Small, DASS-Medium, and DASS-Large are variants of DASS designed for different computational budgets (Table 2). DASS-Tiny consistently achieves higher accuracy with similar latency than MobileNet-v2spar se and provides lower latency while achieving better accuracy than DARTS$_{sparse}$.

4. Robust Feature Representation and Generalization

DASS exhibits strong feature similarity with dense models, reduced generalization gaps, and better class discrimination boundaries (Figure 6). These results confirm that its sparse architectures not only reduce complexity but also preserve high-quality representations essential for downstream tasks.

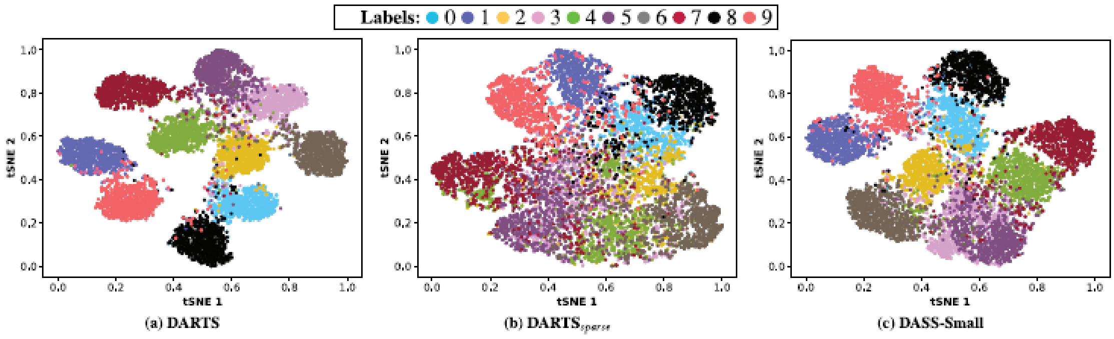

Fig. 6. Visualize decision boundary of (a) DARTS (b) DARTS$_{sparse}$ (c) DASS-Large with the t-SNE embedding method.

4.2. The Progressive adversarial robustness distillation framework (ProARD):

ProARD [7] is a novel framework developed to address a key limitation in existing Adversarial Robustness Distillation (ARD) methods — the inability to efficiently produce a wide range of robust lightweight models for diverse deployment scenarios without retraining. While conventional ARD methods transfer robustness from a large, robust teacher to a single lightweight student, ProARD trains one dynamic network capable of generating many robust student networks directly, thereby cutting computational cost and CO_2 emissions.

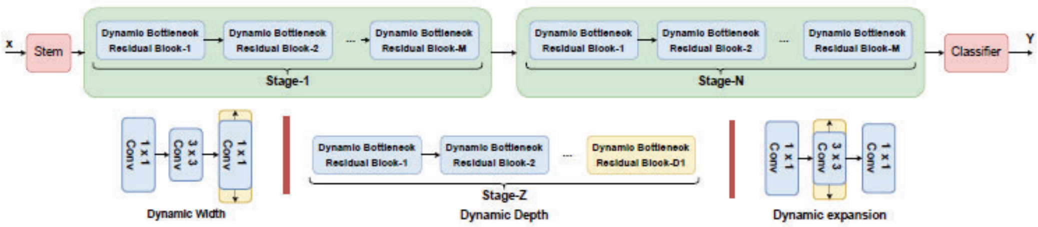

Figure 7: The architecture of Dynamic ResNet with dynamic bottleneck residual blocks. Each dynamic bottleneck

residual block has dynamic width and expansion and each stage has a dynamic depth.

1. Dynamic Network Architecture

ProARD builds a dynamic deep neural network that supports an extremely large design space (>10^{19} possible student architectures) by introducing dynamic layers with variable width (kernel size), depth, and expansion (Figure 7). This design allows a single network to encompass a broad range of architectures, from small, highly efficient models to larger, more capable ones.

- **Dynamic ResNet**: Uses dynamic bottleneck residual blocks with variable output channels, convolution kernel sizes, and stage depths.

- **Dynamic MobileNet**: Similar design, but replaces width with dynamic kernel size in inverted bottleneck blocks.

2. Progressive Adversarial Robustness Distillation

The key innovation is a progressive sampling strategy that avoids the performance degradation caused by random sampling of students. Training occurs in three progressive steps (Figure 8):

1. **Step 1** – Vary width (or kernel size) only.
2. **Step 2** – Vary both width and depth.
3. **Step 3** – Vary width, depth, and expansion simultaneously.

In each step, the largest network configuration is treated as the dynamic teacher, adversarially trained (using RSLAD loss functions) and used to distill robustness to sampled students via weight sharing. This ensures that knowledge transfers efficiently across the entire architecture space.

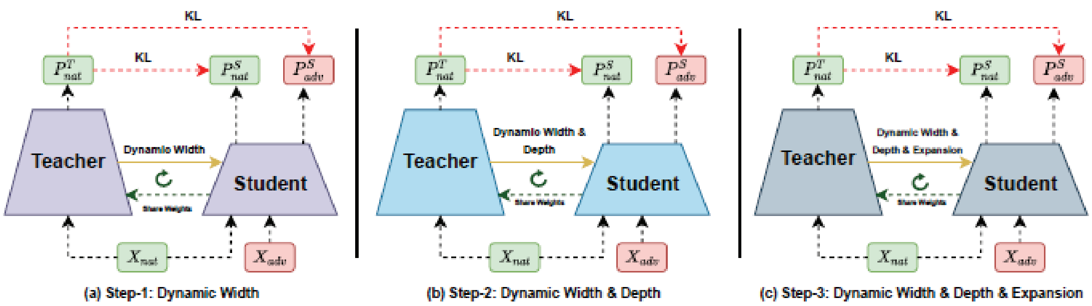

Figure 8: Three steps Progressive Adversarial Robustness Distillation Framework. (a) Step-1: train dynamic width (b) Step-2: train dynamic width and depth, and (c) Step-3: train dynamic width, depth, and expansion.

3. Multi-Objective Search with Accuracy–Robustness Predictor

After training, ProARD uses a multi-objective evolutionary algorithm (NSGA-II) to identify the optimal student architecture based on accuracy, robustness, and FLOPs constraints. To speed up the search, an accuracy–robustness predictor (trained on 2,000 evaluated students) estimates these metrics without expensive adversarial evaluation. This reduces the search time from hours to seconds per candidate.

4. Experimental Performance

- **Accuracy & Robustness Gains**: Compared to random sampling, ProARD improves accuracy by up to **13%** and robustness by up to **14%**.
- **Cost Reduction**: Cuts GPU training hours by **60×** across 50 deployment scenarios.

- **Generalization**: Produces robust students across both CIFAR-10 and CIFAR-100 datasets with competitive or better results than TRADES and standard RSLAD, without retraining (Table 2).
- **Search Quality**: NSGA-II finds final student models with better robustness–accuracy trade-offs than initial random populations.

Table. 2. Accuracy and white-box robustness results on CIFAR-10 and CIFAR-100 datasets. The best results are blodfaced. For RSLAD, we used WideResNet-34-10 as the teacher network.

Dataset	Networks	#Params(M)	Methods	Natural Acc.	FGSM	PGD20	Training
CIFAR10	ResNet50	23.52	TRADES	84.7	59.7	52.6	Yes (train from scratch)
	MBV3	4.21	TRADES	80.0	52.2	48.6	Yes (train from scratch)
	ResNet50 (WideResNet)	23.52	RSLAD	86.0	59.4	52.6	Yes (train from scratch)
	MBV3 (WideResNet)	4.21	RSLAD	**81.2**	54.5	50.5	Yes (train from scratch)
	Dyn-Resnet50	24.94	**ProARD**	**87.0**	**62.8**	**54.2**	**No (quick search)**
	Dyn-MBV3	5.28	**ProARD**	80.6	**55.0**	**50.9**	**No (quick search)**
CIFAR100	ResNet50	23.71	TRADES	54.6	28.7	25.8	Yes (train from scratch)
	MBV3	4.33	TRADES	53.4	28.9	27.2	Yes (train from scratch)
	ResNet50 (WideResNet)	23.71	RSLAD	55.3	**29.6**	**26.5**	Yes (train from scratch)
	MBV3 (WideResNet)	4.33	RSLAD	54.9	**29.1**	**28.7**	Yes (train from scratch)
	Dyn-Resnet50	27.24	**ProARD**	**60.1**	29.4	25.9	**No (quick search)**
	Dyn-MBV3	5.41	**ProARD**	**55.3**	28.6	27.8	**No (quick search)**

5. Key Technical Impact

ProARD unifies **adversarial robustness distillation**, **dynamic architecture design**, and **multi-objective optimization** into a single scalable pipeline. It enables:

- Robustness transfer to a wide spectrum of student models without per-scenario retraining.
- Efficient exploration of huge architecture spaces with minimal computational overhead.
- Deployment-ready model selection tailored to hardware and resource constraints.

4.3. Lightweight 3D lane detection via NAS (3DLaneNAS):

3DLaneNAS [10] is a multi-objective Neural Architecture Search (NAS) framework designed to improve monocular 3D lane detection for autonomous driving. In the Zenseact use case, this involves detecting lanes using front-facing camera sensors, to deliver a cost-efficient and high-performance end-to-end deep learning model.

1. Motivation

Monocular 3D lane detection offers a cost-effective alternative to LiDAR or stereo vision, but existing models often suffer from inefficient feature extraction and fusion, resulting in degraded performance—especially in long-distance lane estimation. 3DLaneNAS addresses this by optimizing the neural architecture specifically for 3D lane perception through NAS with a multi-objective energy function that jointly minimizes detection errors and inference time.

2. Search Space Design

The search space includes both:

- **Feature Extraction Module**: A stack of atomic blocks (ConvBnAct and Squeeze Blocks) across multiple resolution scales, enabling flexible encoding of front-view features (Figure 2).
- **Feature Fusion Module**: Concatenation nodes that merge multi-resolution top-view features with up-sampling or down-sampling as needed (Figure 3).

The projective transformation layer, guided by SIPM, maps front-view features into top-view space, improving lane parallelism for detection.

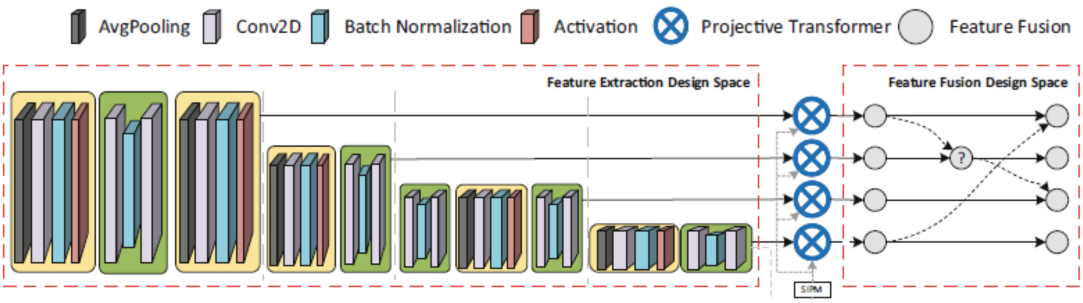

Fig. 9. The overview of 3DLaneNAS architecture. The search space of feature extraction is a stack of ConvBnAct and Squeeze Blocks. The search space of feature fusion is the combination of projective transformation outputs.

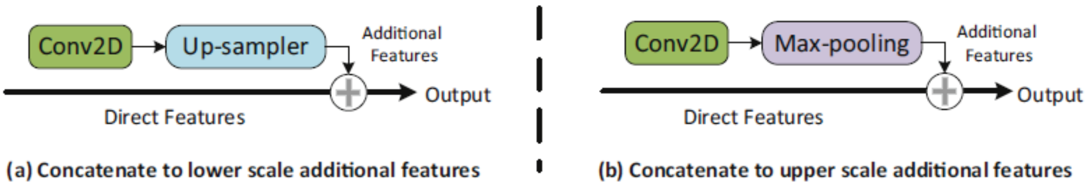

Fig. 10. (a) The Up-sampler is used for concatenating the additional features with lower resolution. (b) The Max-pooling is used for concatenating the additional features with higher resolution.

3. Search Algorithm

3DLaneNAS employs Multi-Objective Simulated Annealing (MOSA) for fast convergence in the discrete search space. The energy function (*Eq. 1*) integrates:

- Lateral and longitudinal lane detection errors (near: 0–40m, far: 40–100m)
- Measured inference time on the target hardware (NVIDIA RTX A4000)

This ensures an optimal **error–latency trade-off** rather than relying on proxy metrics like FLOPs.

$$E = \frac{1}{2n}\sum_{i=1}^{n}((LatE_i + LongE_i)t_i) \times \max(1, \frac{1}{n}\sum_{i=1}^{n}(t_i - \alpha)) \quad (1)$$

We consider a multi-objective energy function (Eq. 1) to improve the 3D lane detection accuracy in addition to reducing the network inference time. The energy function (E) is the multiplication of the network inference time (t) and the average value of lateral (LatE) and longitudinal (LongE) errors. n indicates the number of test samples in each batch. We do not use any proxy such as Floating-Point-Operations-per-Second (FLOPs) for the inference time estimation.

4. Training Strategy

To reduce the heavy computational cost (~10 GPU hours per candidate), the framework uses:

- **Partial training** (5 epochs) during search for faster evaluation
- **Weight transfer** between candidate architectures to accelerate convergence

The final architecture is fully trained for performance evaluation (Table 3).

Table 3. Comparing the performance of 3DLaneNAS with the states-of-the-art.

Architecture	AP (%)	F-score (%)	Lateral error (cm)		Longitudinal error (cm)		#Params (M)	Inference Time (ms)
			0–40 m	40–100 m	0–40 m	40–100 m		
3DLaneNet [10]	74.9	77.7	11.5	60.1	3.2	23.0	20.8	14.5
GenLaneNet [11]	87.2	83	7.4	53.8	1.5	23.2	3.36	16
3DLaneNAS	**92.4**	**92.1**	**3.7**	**35.8**	**0.5**	**19.2**	**1.75**	**12**

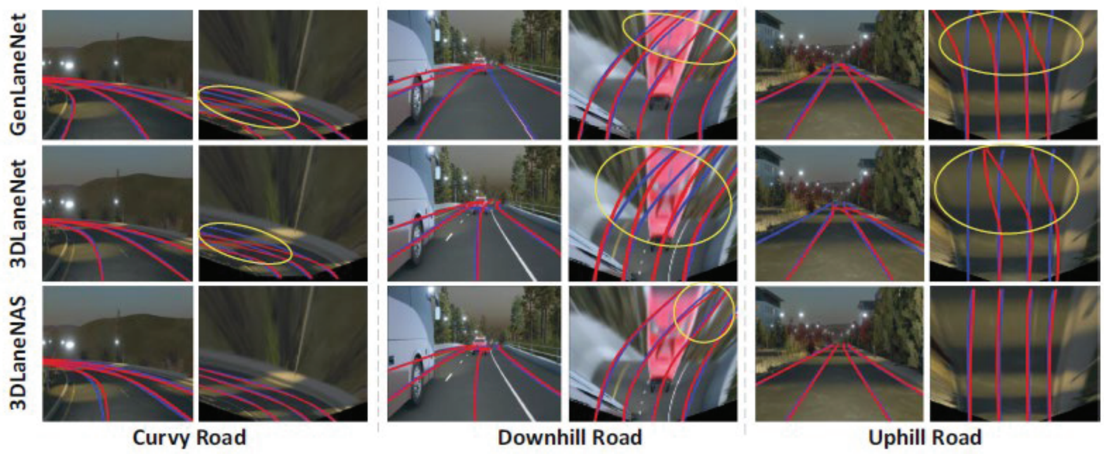

Fig. 11. Illustrating the performance of 3DLaneNet [10], GenLaneNet [11] and 3DLaneNAS in three different road scenarios. Blue lines are ground-truth, red lines are network predictions. Left column: curvy roads' results. Middle column: downhill sample results. Right column: uphill road sample. Yellow circles show low-confidence estimations. (Color figure online)

5. Experimental Results

On the synthetic-3D-lanes dataset, 3DLaneNAS achieves:

- **+17.5% AP** over 3DLaneNet and **+5.2% AP** over GenLaneNet
- **1.2–1.33× lower inference time**
- **41.9–50.4% lower longitudinal error** and **44–59% lower lateral error**

Qualitative results in Figure 11 show improved robustness in **curvy, uphill, and downhill** road scenarios.

6. Search Method Efficiency and Reproducibility

3DLaneNAS demonstrates steady energy function convergence (Figure 12a) and superior error–latency performance (Figure 12b) compared to random and local search. Multiple search runs with different seeds yield consistent results, with only **2.74% standard deviation** (STDEV), proving reproducibility.

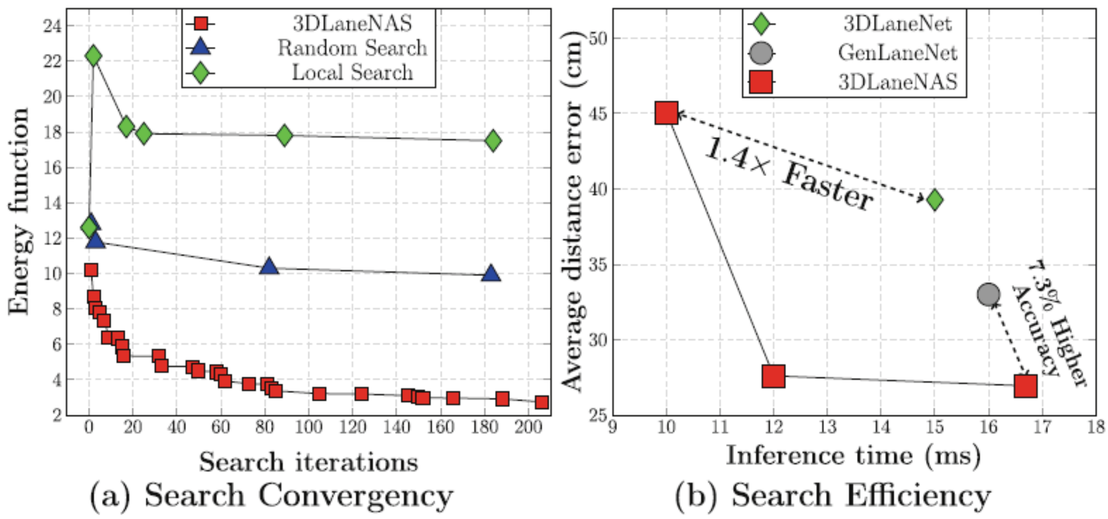

Fig. 12. Convergency of energy function (b). Error-latency trade-off of 3DLaneNAS in comparison with 3DLaneNet and GenLaneNet.

4.4. Contrastive Learning for Lane Detection with cross-similarity (CLLD):

The CLLD framework [13] was developed through close collaboration between MDU and Zenseact, targeting the creation of robust lane detection models for camera-based perception systems in autonomous driving. Lane detection from camera sensors is highly susceptible to adverse lighting, weather, occlusion from other vehicles, and worn lane markings. This makes the ability to recover missing or partially visible lane segments critical for safety and reliability.

CLLD is a self-supervised multi-task contrastive learning framework designed to pretrain lane detection models without the need for large-scale labeled datasets. Its novelty lies in introducing a cross-similarity operation that captures relationships between visible and occluded parts of an image, enabling lane reconstruction even when visibility is impaired. This design allows the model to maintain long-range dependencies between spatially distant but contextually related lane features. Figure 13 presents the **overall CLLD pipeline**, where an input image is masked to simulate occlusion, passed through an encoder–decoder backbone, and trained with multiple self-supervised losses to reconstruct both lane geometry and contextual features. The approach is designed to improve resilience to missing visual information without relying on large labeled datasets.

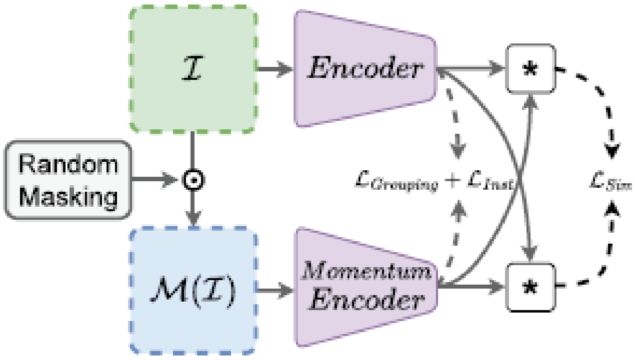

Fig. 13. The CLLD framework.

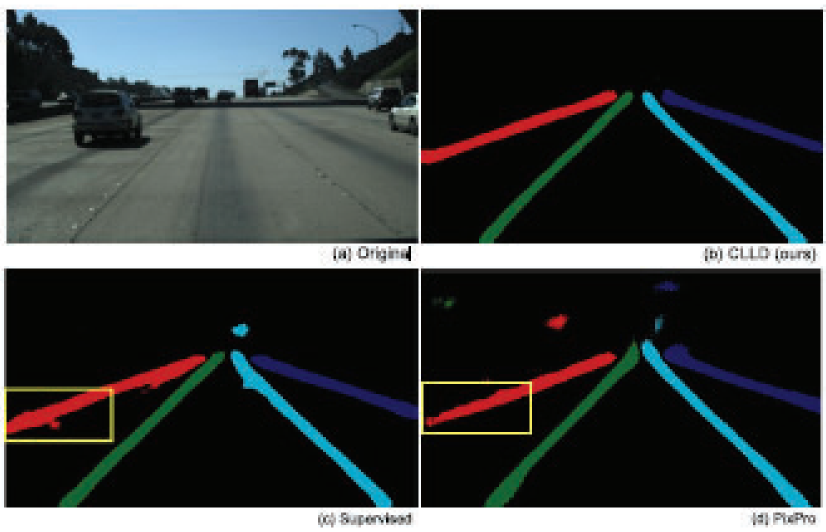

Fig. 14. Comparison of the state-of-the-art segmentation-based lane detection RESA with three different pretraining strategies. (a) Input image (b) RESA output with CLLD (ours) and (c) RESA output with supervised and (d) RESA output with PixPro pretraining. Yellow boxes represent accuracy drops in the detection of lanes that are occluded by cars.

1. Positioning Within Lane Detection Methods

Figure 14 compares CLLD's learning paradigm with existing self-supervised methods, highlighting its local patch-level processing and the novel cross-similarity mechanism that links corresponding regions between masked and unmasked feature maps. This distinguishes CLLD from prior global contrastive learning approaches by enabling fine-grained structural alignment in lane patterns.

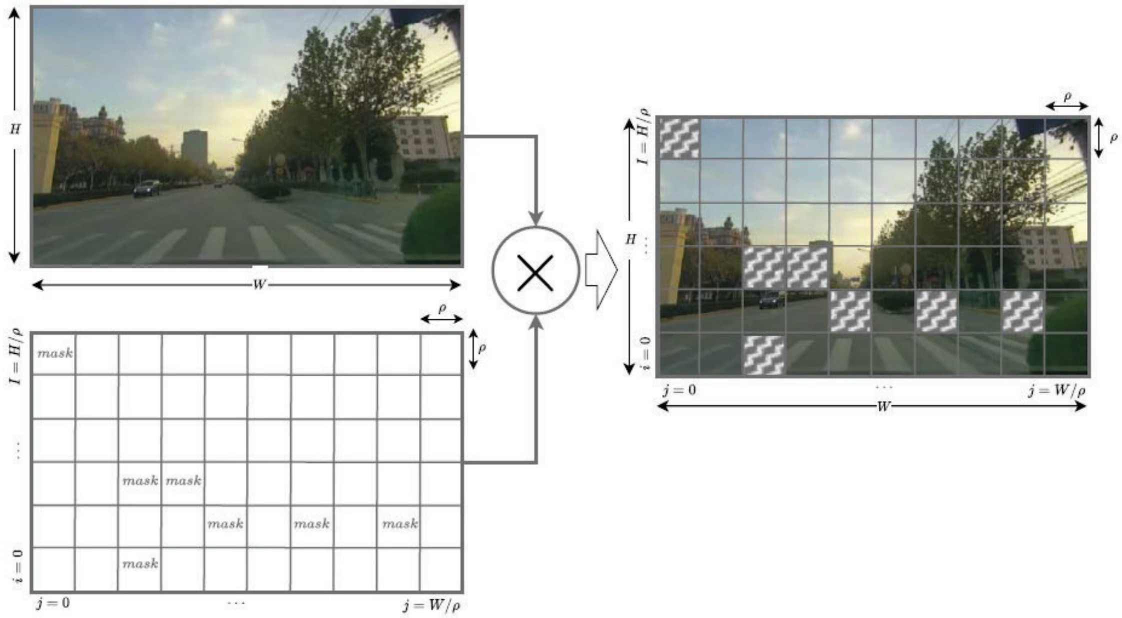

Fig. 15. Masking input image; The given input with size $H \times W$ is divided into $\rho \times \rho$ patches. Each pixel of the masked patch got a random value from a zero-mean normal distribution N(0, 1).

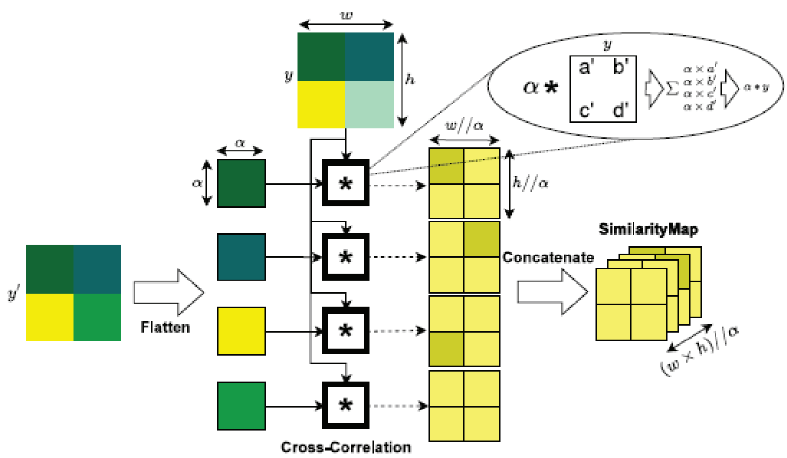

Fig. 16. The cross-similarity operation.

2. Methodology:

a) **Local Feature Contrastive Learning:** Instead of comparing entire images (global features), CLLD works at the patch level, enabling the model to detect lane segments in localized regions.

b) **Masking Strategy (Fig. 15):** Portions of the input image are randomly masked, simulating occlusion. The model is then tasked with predicting the masked areas based on surrounding context.

c) **Cross-Similarity Operation (Fig. 16):** For each patch in the masked image's feature map, CLLD computes similarity scores with all patches in the original image's feature map, and vice versa. This bidirectional comparison ensures that the network learns structural correlations and positional relationships critical for reconstructing occluded lane segments.

d) **Multi-Loss Training:** The optimization combines ($L_{clld} = L_{cons} + L_{sim} + L_{inst}$):
   - **Consistency Loss:** Aligns features between masked and unmasked views.
   - **Similarity Loss:** Matches cross-similarity structures across views.
   - **Instance-Level Loss:** Improves discriminative representation learning.

3. Experimental Highlights

- **Benchmarks:** Evaluated on TuSimple and CuLane, two standard lane detection datasets.
- **Architectures:** Applied to U-Net, RESA, and CLRNet backbones.
- **Results:**
  - Achieved up to **4% improvement** in challenging "shadow" conditions on CuLane (Fig. 17).
  - Outperformed state-of-the-art contrastive learning baselines (PixPro, VICRegL, DenseCL) in F1-score and accuracy while maintaining competitive precision.

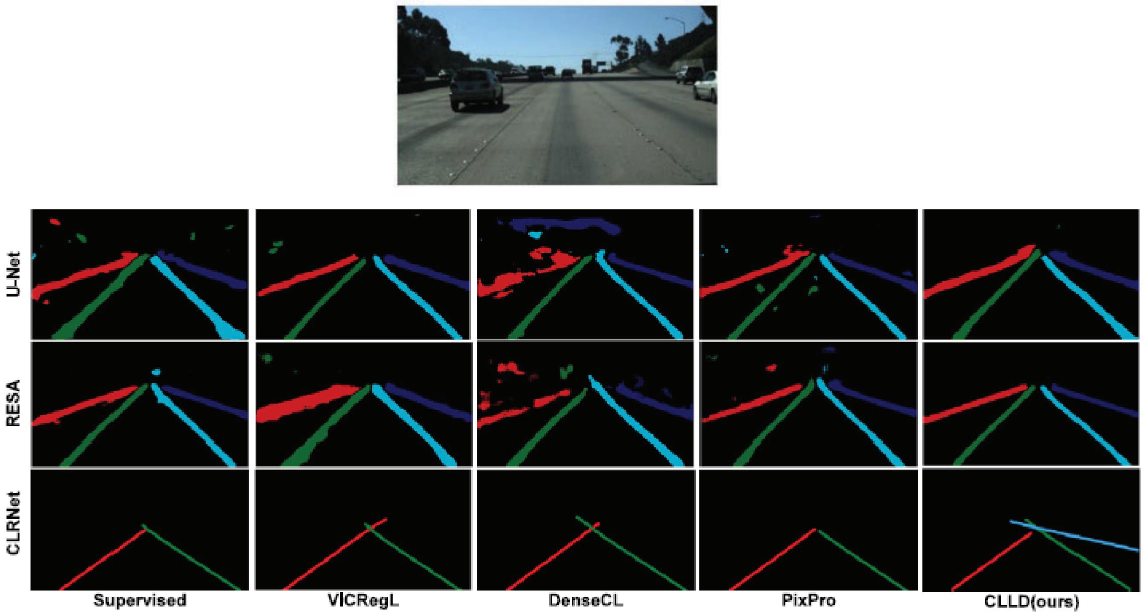

Fig. 17. Qualitative comparison of the results of CLLD with prior SSL methods and supervised learning.

4. Conclusion

CLLD demonstrates that self-supervised pretraining with cross-similarity can significantly enhance camera-based lane detection under challenging visibility. By reducing dependency on labeled datasets and improving resilience to occlusion, it serves as a scalable approach for both research and production-grade ADAS and autonomous driving pipelines.

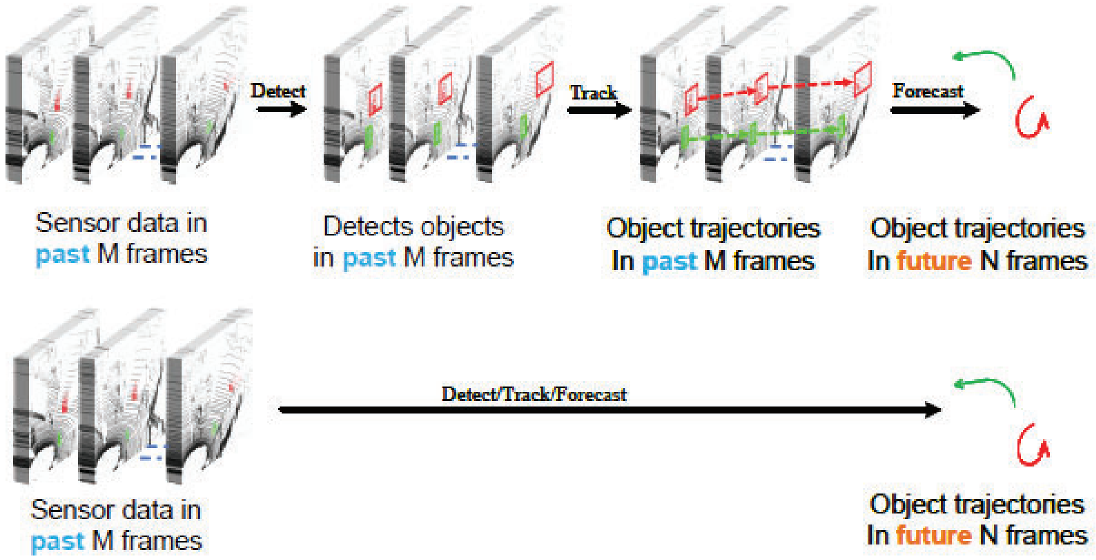

Fig. 18 (Top Row) Cascade methods, which handle detection, tracking, and forecasting in a sequential pipeline, are vulnerable to error propagation. In the diagram, the arrows indicate the direction of processing for the Lidar data, moving from raw input to the final output. The input data, represented in blue, is gathered from past observations, while the future output is shown in orange. This is because each stage assumes error-free input from the previous one, which is often unrealistic in real-world applications. As a result, errors can accumulate and negatively impact the final predictions. (Bottom Row) End-to-end methods, on the other hand, directly predict future trajectories from raw data. This unified approach allows for the joint optimization of detection, tracking, and forecasting, leading to more accurate and reliable results.

4.5. A Neural Architecture Search Framework for Trajectory Prediction (TrajectoryNAS):

The TrajectoryNAS framework [12] was developed through collaboration between MDU and Volvo Construction Equipment (VCE) to address the challenge of accurate, real-time trajectory forecasting from LiDAR sensory data in dynamic and often unstructured industrial environments. In VCE's work sites—populated with heavy machinery such as wheel loaders, dump trucks, and excavators—predicting the movement of surrounding agents based on 3D point cloud data is critical for operational safety, collision avoidance, and efficient autonomous navigation.

Conventional cascaded perception pipelines for LiDAR data perform detection, tracking, and forecasting in separate stages, often leading to error propagation and increased latency. TrajectoryNAS instead employs an end-to-end 3D deep learning model for point cloud processing, jointly optimizing object detection, tracking, and future motion prediction in a single pipeline. Fig. 18 contrasts the limitations of cascaded LiDAR pipelines with the benefits of end-to-end design. The architecture is automatically designed using Neural Architecture Search (NAS) to maximize both accuracy and runtime efficiency.

1. Proposed framework:

TrajectoryNAS is the first trajectory prediction framework to apply NAS end-to-end for LiDAR-based forecasting. It optimizes all components—from feature extraction to detection heads—using a multi-objective energy function that balances accuracy (AP, ADE, FDE) and latency. A hybrid exploration–exploitation strategy is used:

- Phase 1 – Exploration: Candidate architectures are rapidly evaluated on a miniaturized NuScenes LiDAR dataset (~10× faster than full training).
- Phase 2 – Exploitation: The best architecture from Phase 1 is fully trained on the complete dataset.
- Phase 3 – Deployment: The model is evaluated on target hardware to verify latency constraints.

This process yields architectures that are both accurate in prediction and efficient in inference, crucial for real-time LiDAR perception systems.

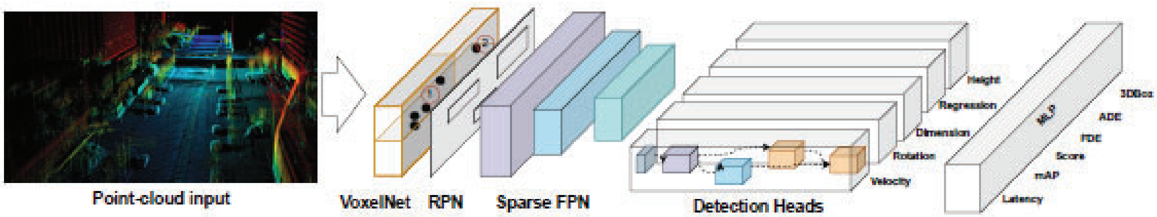

Fig. 19. The overview of TrajcetoryNAS process.

TrajectoryNAS builds on the **VoxelNet 3D object detection backbone** to process raw LiDAR point cloud input and is extended with several key components designed specifically for the VCE use case (Fig. 19):

- **NAS-Generated Multi-Head Detection Modules** – Instead of manually designing the detection heads, TrajectoryNAS uses its Neural Architecture Search process to automatically configure multiple parallel prediction heads. Each head specializes in estimating a specific property of detected objects, including velocity vectors, orientation (rotation), physical dimensions, bounding box regression offsets, and height above ground. This automated design ensures that each head is optimally structured for point cloud–based perception, delivering both accuracy and efficiency (Fig. 19, right-hand side).

- **Future Feature Prediction Network** – Beyond detecting objects in the current LiDAR frame, TrajectoryNAS includes a lightweight temporal prediction module that learns to transform present-frame feature maps into predicted feature maps for future timesteps. This enables the model to perform **direct detection in future frames** without requiring iterative simulation of the scene or external tracking modules, thereby reducing computational overhead while improving temporal consistency (Fig. 19, middle section).

- **Backcasting Association Mechanism** – Once objects are detected in future frames, the model projects them back ("backcasting") to the present reference frame using a constant-velocity motion model. This step ensures **consistent multi-step trajectory alignment** between present and future detections, which is essential for building coherent trajectories over the entire prediction horizon (Fig. 19, temporal linking arrows).

The NAS process searches over an expansive design space of more than 2,300 candidate 3D deep model architectures, covering variations in backbone configuration, detection head design, and temporal prediction layers. This allows TrajectoryNAS to produce architectures highly specialized for VCE's LiDAR-based, point cloud–driven operational scenarios, balancing accuracy, robustness, and real-time performance.

2. Search Algorithm

TrajectoryNAS uses a **Multi-Objective Simulated Annealing (MOSA)** strategy to design an optimal end-to-end 3D deep model for LiDAR-based trajectory prediction in the VCE use case. Candidate architectures are first evaluated on a mini NuScenes LiDAR dataset for fast exploration, then the best design is retrained on the full dataset for exploitation, and finally its real latency is measured on an RTX A4000. The search optimizes the multi-objective energy function:

$$E = Latency \times mAP^{\alpha} \times ADE^{\beta} \times FDE^{\gamma}$$

where mAP measures detection and forecasting precision, and ADE/FDE capture trajectory displacement errors. This process yields architectures that meet the high-accuracy and low-latency needs of real-time point cloud processing in industrial environments.

3. Experimental Results (Tables 4 & 5, and Fig. 20)

- **Dataset:** NuScenes LiDAR, 3-second forecasting horizon.
- **Performance (Cars):**
    - Outperforms Fast and Furious and FutureDet in mAPf for static, linear, and nonlinear motion.
    - Achieves highest APf for nonlinear trajectories while maintaining competitive latency (~22 ms).
- **Performance (Pedestrians):**
    - Significant gains in predicting complex, non-linear movement patterns.
    - Maintains high APf across K=1 and K=5 settings.
- **Qualitative Results:** Fig. 20 shows the predictions for both vehicles and pedestrians closely follow ground-truth trajectories, with minimal false motion for static objects.

Table 4. Comparison TrajcetoryNAS and state-of-the-art trajectory prediction model on cars according to accuracy and latency metrics.

Method	Time (ms)	K = 1								K = 5							
		$AP^{stat.}$		$AP^{lin.}$		$AP^{non-lin.}$		mAP		$AP^{stat.}$		$AP^{lin.}$		$AP^{non-lin.}$		mAP	
		$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f
Detection + Constant Velocity	21	70.3	66.0	65.8	21.2	90.0	6.5	75.4	31.12	70.3	66.0	65.8	21.2	90.0	6.5	75.4	31.2
Detection + Forecast [21]	20	69.1	64.7	66.1	22.2	86.3	7.5	73.8	31.5	69.1	64.7	66.1	22.2	86.3	7.5	73.8	31.5
FutureDet [25]	24	70.0	65.5	62.9	24.9	91.8	10.1	74.9	33.5	70.1	67.3	62.9	27.7	91.7	11.7	74.9	35.6
TrajectoryNAS (ours)	22	71.0	65.6	63.8	26	91.2	10.3	75	34	71	67.4	63.8	29.2	91.1	12.1	75.3	36.2

Table 5. Comparison TrajcetoryNAS and state-of-the-art trajectory prediction model on pedestrian according to accuracy and latency metrics.

Method	Time (ms)	K = 1								K = 5							
		$AP^{stat.}$		$AP^{lin.}$		$AP^{non-lin.}$		mAP		$AP^{stat.}$		$AP^{lin.}$		$AP^{non-lin.}$		mAP	
		$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f
Detection + Constant Velocity	21	55.1	33.3	73.5	27.8	96.9	12.4	75.2	25.5	55.1	33.3	73.5	27.8	96.9	12.4	75.2	24.5
Detection + Forecast [21]	20	53.7	35.0	73.9	30.8	97.2	13.3	74.9	26.4	53.7	35.0	73.9	30.8	97.2	13.3	74.9	26.4
FutureDet [25]	24	53.1	33.3	72.4	32.6	95.2	14.7	73.6	26.9	53.1	35.1	72.4	34.0	95.2	15.0	73.6	28.0
TrajectoryNAS (ours)	22	55.8	37.1	77.9	39.9	95.2	17.7	76.3	31.3	55.8	38.6	77.9	40.9	95.2	17.9	76.3	32.5

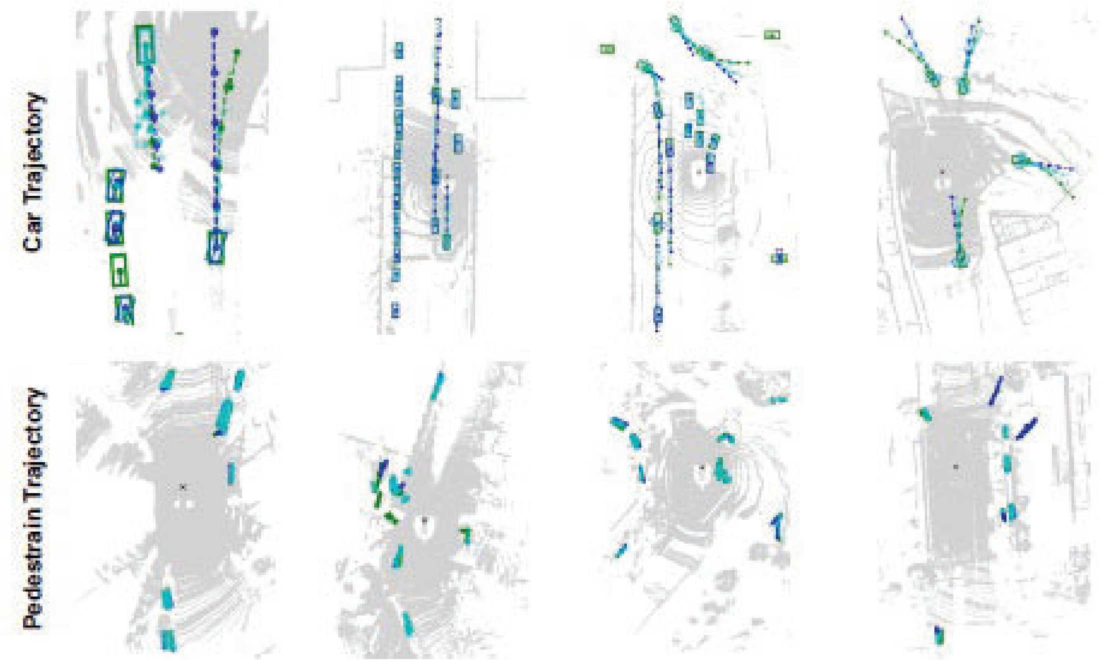

Figure 20. The visual demonstration of TrajectoryNAS; the first row is the trajectory prediction for cars, and the second row is the trajectory prediction for the pedestrian. Green lines are ground-truth. Blue lines are trajectory prediction with highest probability. Cyan lines are trajectory predictions with the highest probability.

4. Conclusion and advantages for LiDAR-Based Industrial Autonomy

TrajectoryNAS delivers a **NAS-optimized, LiDAR-based end-to-end trajectory forecasting pipeline** that jointly addresses detection, tracking, and prediction. By automatically finding the best architecture under real-world latency and accuracy constraints, it achieves **superior prediction performance** for both vehicles and pedestrians. This makes it highly suitable for safety-critical, real-time applications in autonomous driving and industrial autonomy.

In a **Volvo CE-style use case**, TrajectoryNAS can be deployed on heavy machinery equipped with LiDAR sensors to:

- Predict the paths of nearby machines and workers with high temporal precision.
- Operate within real-time constraints on embedded GPUs.
- Maintain robust forecasting for both structured and unstructured environments.

4.6. Deep Learning-Based Acceleration-Aware Trajectory Forecasting (DAT):

The DAT framework [11] was developed through collaboration between MDU and Volvo Construction Equipment (VCE) to address the challenge of accurate, real-time trajectory forecasting from LiDAR sensory data in dynamic and often unstructured industrial environments. In heavy machinery operations, such as those involving wheel loaders, dump trucks, and excavators, predicting the motion of surrounding agents based on LiDAR observations is critical for operational safety, collision avoidance, and efficient autonomous navigation. Conventional cascaded perception pipelines process LiDAR data in separate detection, tracking, and forecasting stages, which often causes error propagation between components. DAT instead adopts an end-to-end LiDAR-based learning paradigm that jointly detects objects and predicts their future trajectories in a single network, reducing intermediate noise and improving accuracy. Fig. 18 contrasts the limitations of cascaded LiDAR pipelines with the benefits of DAT's unified design.

1. Acceleration-Aware Forecasting

Most forecasting networks assume constant velocity, which is insufficient for modeling real-world, nonlinear motion such as acceleration, deceleration, or sudden maneuvers. DAT introduces an acceleration prediction head alongside initial velocity prediction, enabling accurate modeling of both linear and nonlinear motion patterns. This enhancement is vital in Volvo CE's work sites, where machine interactions involve frequent changes in speed and direction.

2. Deriving Ground-Truth Acceleration

Public LiDAR datasets like NuScenes lack acceleration labels, which are essential for modeling realistic motion changes. DAT addresses this by applying **Second-Order Regression (SOR)** to sequential LiDAR detections, fitting a second-order motion model to estimate acceleration directly.

Compared to the Extended Kalman Filter (EKF), SOR is:

- **More robust to LiDAR noise** and labeling errors.

- **Effective with short observation windows**, common in industrial sites.
- **Independent of strict noise assumptions**, making it broadly applicable.

On NuScenes, SOR consistently outperformed EKF in both accuracy and stability (Figs. 21 and Table 6), providing reliable ground-truth acceleration for training DAT's forecasting model.

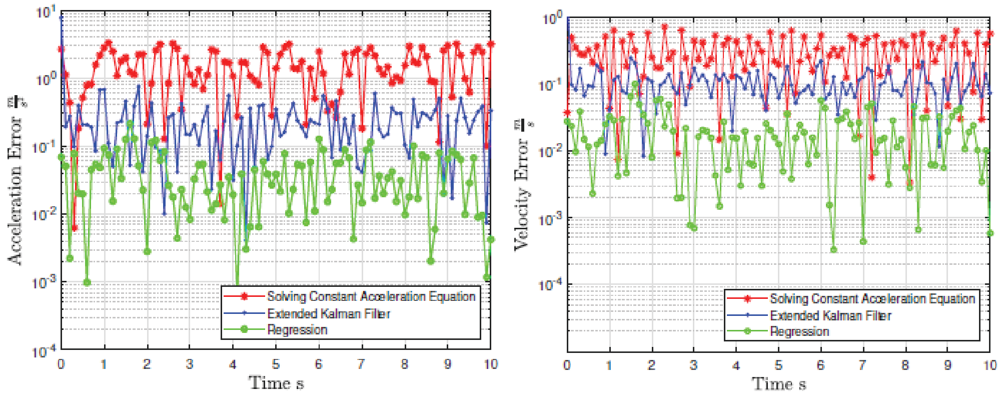

Figure 21. (left)Acceleration error comparison across different methods, (right) Initial velocity error comparison across different methods.

Table 6. Comparison of velocity and acceleration error. The cyan color specified the best errors gathered:

	V_0		A	
	μ_v	σ_v^2	μ_a	σ_a^2
SOR	0.02	0.03^2	0.04	0.06^2
EKF	0.12	0.17^2	0.30	0.79^2
Equation	0.30	0.19^2	1.57	1.84^2

3. Architecture and Methodology

As presented in Fig. 22, DAT extends the CenterPoint LiDAR-based 3D object detection backbone with three enhancements tailored for joint perception and forecasting

- **Acceleration & Initial Velocity Heads** – Beyond standard properties (size, orientation, position), the network directly estimates each object's instantaneous acceleration and initial velocity to capture non-linear motion typical in industrial sites.
- **Feature Transformation Module** – A lightweight temporal block that maps current LiDAR-frame features to predicted future-frame features, encoding short-term dynamics without long history buffers.
- **Backcasting Mechanism** – Uses a constant-acceleration motion model to project future detections back to the present, enabling accurate association between predicted and observed objects across frames.

Integrating these components lets DAT jointly optimize LiDAR-based perception and forecasting in a single end-to-end pipeline—removing separate tracking stages and reducing error propagation.

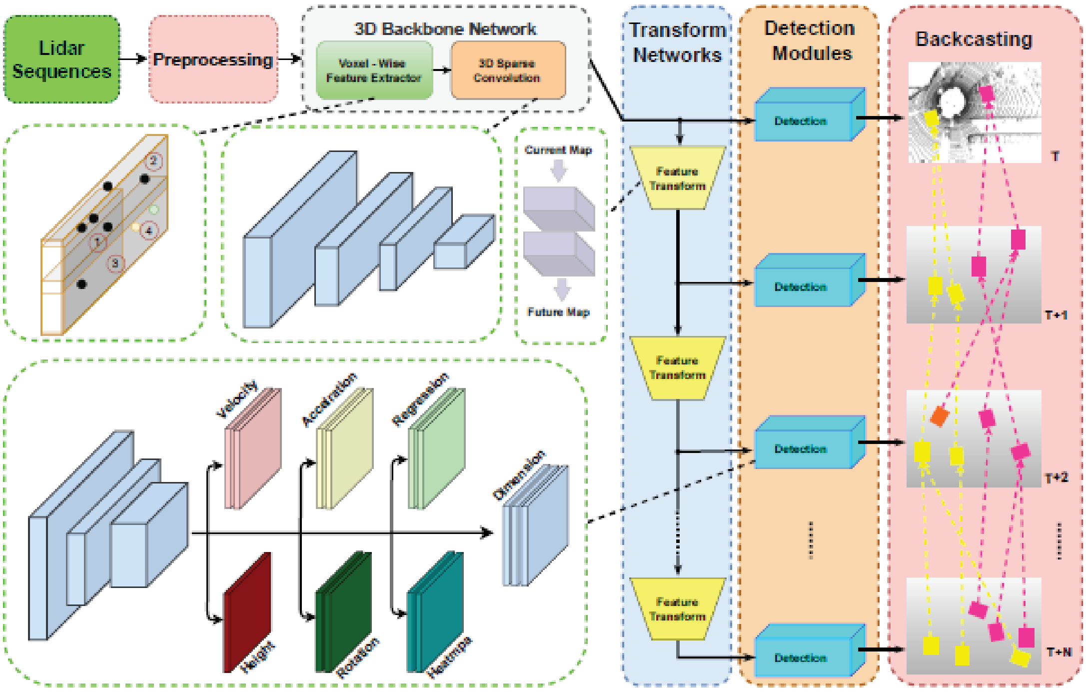

Figure 22. DAT: based on a LiDAR sequence, DAT detects objects in both the present frame (t) and future frames (up to t + T). These future detections are projected back to the current frame allowing for alignment with detections in the present moment.

4. Experimental Results

DAT was evaluated on the NuScenes dataset using LiDAR sensory data, with a prediction horizon of **3 seconds**. Two popular LiDAR backbones, VoxelNet and PointPillars, were used to validate generality across architectures.

a) Performance Highlights (Table 7)**:**

- For **nonlinear motion** (e.g., accelerating or turning objects), DAT achieved up to **82% higher forecasting accuracy** (APf) than TrajectoryNAS [12] under the single-prediction setting (K=1).
- Improvements were consistent across motion types—static, linear, and nonlinear—and across multiple prediction horizons. Under the top-5 predictions setting (K=5), DAT continued to outperform all baselines, as shown in Table 7.
- The integration of acceleration prediction proved particularly effective in reducing forecast drift over longer horizons, a common issue in LiDAR-based trajectory prediction.

b) Qualitative Analysis (Fig. 23)**:**

- Figure 23 shows the visualization of representative NuScenes scenes, showing that DAT's LiDAR-based predicted trajectories align closely with the ground truth for all motion categories. In complex situations—such as vehicles taking sharp turns or abruptly changing speed—DAT maintained smooth and accurate trajectory curves, whereas baseline methods often diverged significantly.

- These results demonstrate that DAT not only delivers **quantitative gains** in forecasting accuracy but also produces **qualitatively more reliable** motion predictions, making it well-suited for deployment in VCE's LiDAR-driven autonomy stack.

Table 7. Evaluation of the car object detection and trajectory forecasting pipeline on the NuScenes datasets. Rows with cyan color are our model's results:

	K = 1								K = 5							
	$AP^{stat.}$		$AP^{lin.}$		$AP^{non-lin.}$		mAP		$AP^{stat.}$		$AP^{lin.}$		$AP^{non-lin.}$		mAP	
	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f	$AP_{det.}$	AP_f
Detection + Constant Velocity	70.3	66.0	65.8	21.2	90.0	6.5	75.4	31.12	70.3	66.0	65.8	21.2	90.0	6.5	75.4	31.2
Detection + Forecast (Luo et al. [31])	69.1	64.7	66.1	22.2	86.3	7.5	73.8	31.5	69.1	64.7	66.1	22.2	86.3	7.5	73.8	31.5
FutureDet (Peri et al. [4])	70.0	65.5	62.9	24.9	**91.8**	10.1	74.9	33.5	70.1	67.3	62.9	27.7	**91.7**	11.7	74.9	35.6
FutureDet–PointPillars (Peri et al. [4])	70.1	64.1	63.4	24.8	92.4	9.6	75.4	32.8	70.7	67.5	63.4	28.8	92.0	11.9	75.4	36.1
FutureDet + Map (Peri et al. [4])	70.2	65.5	62.7	24.3	91.7	9.4	74.9	33.1	70.2	67.5	62.7	27.1	91.7	11.0	74.9	35.2
TrajectoryNAS (Sharifi et al. [35])	71.2	65.6	63.8	26	91.2	10.3	75	34	71	67.4	63.8	29.2	91.1	12.1	75.3	36.2
Ours	72.1	**66.2**	**70.8**	**30.1**	91.3	**18.7**	**78.0**	**38.3**	72.1	**69.4**	**70.8**	**36.3**	91.3	23.5	**78.0**	**43.0**
Ours + PointPillars	70.0	62.7	65.5	27.1	89.3	17.5	75.0	35.8	70.0	66.5	65.5	33.9	89.3	**24.0**	75.0	41.1
Ours + MAP	**72.3**	66.0	70.2	29.6	91.5	18.1	**78.0**	37.9	**72.3**	69.3	70.2	35.5	91.5	22.8	**78.0**	42.5

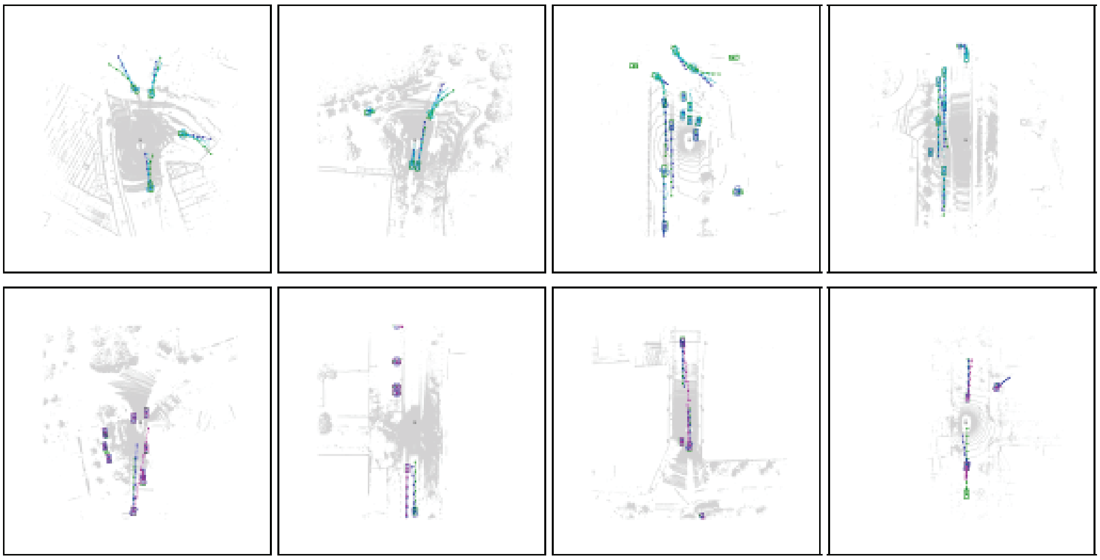

Figure 23. Qualitative evaluation of trajectory forecasts using DAT. In the first row, ground-truth trajectories are depicted in green, the highest confidence forecast in blue, and other potential future trajectories in cyan. The second row compares the highest confidence forecasts of DAT (blue) with those of TrajectoryNAS (magenta), alongside the ground-truth trajectories (green). The results illustrate that DAT predictions are closer to the ground truth.

5. Conclusion

DAT introduces a robust, acceleration-aware end-to-end trajectory forecasting framework that overcomes the limitations of constant-velocity models. By integrating acceleration prediction and unifying detection with forecasting, it delivers superior accuracy for complex motion patterns, making it well-suited for Volvo CE's industrial autonomy roadmap.

5. Dissemination and publications

How are the project results planned to be used and disseminated		Comments
Increase knowledge in the field	x	The project's dissemination activities have advanced knowledge in the field by sharing its findings, methodologies, and technological advances with both academia and industry through publications, conferences, workshops, and targeted engagement. All developed codes are released as open source, providing the community with practical, reusable tools that support reproducibility, adaptation, and faster innovation. These efforts have strengthened the state of the art and fostered collaboration across research, industry, and policy sectors.
Be passed on to other advanced technological development projects	x	The project's results are designed to be transferable to other advanced technological development initiatives. Its methods, open-source tools, and validated workflows can be readily adapted to new domains, enabling future projects to build on proven solutions rather than starting from scratch. This accelerates development cycles, reduces costs, and promotes knowledge continuity across research and innovation activities.
Be passed on to product development **Projects Introduced on the market**	x	The project's outcomes have strong potential to be integrated into product development pipelines, providing tested algorithms, open-source code, and scalable workflows that can shorten time-to-market. By embedding these innovations into commercial solutions, they can enhance product performance, enable new features, and support the introduction of advanced, market-ready technologies.
Used in investigations / regulatory / licensing / political decisions		

5.1 Publications

During the course of the project, we published four Q1 journal papers, four conference papers, and one licentiate thesis (midway report), with a second licentiate thesis expected to be completed within a month of this project report. The majority of these publications resulted from close collaboration with industry partners.

1. S. H. Mousavi, S. A. Mousavi, and M. Daneshtalab, "ProARD: progressive adversarial robustness distillation: provide wide range of robust students," *in Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2025, Italy.

2. S. A. Mousavi, H. Mousavi, and M. Daneshtalab, "FARMUR: Fair Adversarial Retraining to Mitigate Unfairness in Robustness" *in proc. of 27th Springer European Conference on Advances in Databases and Information Systems (ADBIS)*, 2023, Spain.

3. H. Mousavi, A. Zoljodi, and M. Daneshtalab, "Analysing robustness of tiny deep neural networks" *in proc. of 27th Springer European Conference on Advances in Databases and Information Systems (ADBIS)*, 2023, Spain.

4. Zoljodi, S. Abadijou, M. Loni, M. Alibeigi, M. Daneshtalab, "3DLaneNAS: Neural Architecture Search for Accurate and Light-Weight 3D Lane Detection," *in Proceedings of 31st Springer International Conference on Artificial Neural Networks (ICANN)*, 2022, UK

5. (Journal) Zoljodi, S Abadijou, M Alibeigi, M Daneshtalab, "Contrastive Learning for Lane Detection via cross-similarity," *Elsevier Journal of Pattern Recognition Letters* (*PRL-Elsevier*), 2024. (IF: 3.3, Q1)

6. (Journal) H. Mousavi, M. Loni, M. Alibeigi, M. Daneshtalab, "DASS: Differentiable Architecture Search for Sparse Neural Networks," *ACM Transactions on Embedded Computing Systems* (*ACM-TECS*), Vol. 22, pp. 1-21, 2023 (IF: 2, Q1).

7. (Journal) A. Sharifi, A. Zoljodi, M Daneshtalab, "DAT: Deep Learning-Based Acceleration-Aware Trajectory Forecasting," *Multidisciplinary Digital Publishing Institute Journal of Imaging* (*MDPI-imaging*), 2024. (IF: 3.3, Q1)

8. (Journal) A. Sharifi, A. Zoljodi, M Daneshtalab, "TrajectoryNAS: A Neural Architecture Search for Trajectory Prediction," *Multidisciplinary Digital Publishing Institute Journal of Sensors* (*MDPI-Sensors*), 2024. (IF: 3.4, Q1)

There are additional relevant publications that were inspired and driven by this project, although they are not directly linked to the specific use cases and were not carried out in collaboration with the involved industrial partners.

9. (Journal) M Chakraborty, M Aryapoor, and M Daneshtalab, "Frequency Domain Complex-Valued Convolutional Neural Network" *Elsevier Journal of Expert Systems With Applications* (*ESWA*), 2025. (IF: 7.5, Q1)

10. (Journal) MH Ahmadilivani, M Taheri, J Raik, M Daneshtalab, M Jenihhin, "A systematic literature review on hardware reliability assessment methods for deep neural networks," *ACM Computing Surveys* (*ACM-CSUR*), Vol. 65, pp. 1-39, 2024. (IF: 28, Q1)

11. (Journal) M. Loni, A. Zoljodi, A. Majd, B. Ahn, M. Daneshtalab, M. Sjödin, H. Esmailzadeh "FastStereoNet: A Fast Neural Architecture Search for Improving the Inference of Disparity Estimation on Resource-Limited Platforms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (*IEEE TSMCS*), Vol. 52, No. 8, pp. 5222–5234, 2022. (IF: 11, Q1)

12. M. Loni, H. Mousavi , M. Riazati, M. Daneshtalab, M. Sjödin, "TAS: Ternarized Neural Architecture Search for Resource-Constrained Edge Devices," *in Proceedings of 25th ACM/IEEE Design, Automation, and Test in Europe (DATE)*, 2022, Virtual.

13. Z. Yigit, H. Forsberg, M. Daneshtalab, "Machine Learning-Based Prognostic Approaches for Construction Equipment Powertrain Systems", *in Proceedings of 36th IEEE Intelligent Vehicles Symposium (IV), 2025, Romania.*

14. J. Lindén, A. Ermedahl, H. Salomonsson, M. Daneshtalab, B. Forsberg, P. Carbone, "Autonomous Realization of Safety-and Time-Critical Embedded Artificial Intelligence" *in Proc. of 27th ACM/IEEE Design, Automation, and Test in Europe (DATE), 2024, Spain.*

15. M. H. Ahmadilivani, M. Taheri, J. Raik, M. Daneshtalab, M. Jenihhin, "DeepVigor: Vulnerability Value Ranges and Factors for DNNs' Reliability Assessment", *in Proc. of 28th IEEE European Test Symposium (ETS)*, 2023, USA.

5.2 Availability of Project Resources and open access

All source codes developed as part of this project, including those linked to the published research, are openly accessible via the DeepHERO Lab webpage: https://www.es.mdu.se/deephero/, supporting transparency, reproducibility, and knowledge sharing within the research community.

All journal publications are available in open-access format, and preprints of the conference papers are also openly accessible.

6. Conclusions and future research

The project has successfully achieved its primary objectives, delivering high-quality scientific outcomes, including multiple Q1 journal publications, conference papers, and licentiate theses. These results were largely enabled by strong collaboration between academic researchers and industrial partners, ensuring both scientific impact and practical relevance. The core technical achievements include the development of advanced AI frameworks for the project's main use cases: efficient end-to-end models for trajectory prediction and lane detection. These frameworks have been extensively validated on real-world datasets and benchmarked against state-of-the-art approaches, demonstrating notable improvements in accuracy, robustness, and computational efficiency. Designed for real-time, resource-constrained environments, they address practical deployment challenges in autonomous systems while maintaining strong scientific rigor.

Looking ahead, future research will focus on extending the developed methods to broader application areas, enhancing scalability and adaptability to diverse industrial contexts. This includes further validation on real-world datasets, integration with emerging technologies, and exploration of new architectures and algorithms for improved performance, robustness, and sustainability. We also aim to expand this work into federated settings, investigating how to improve the robustness and reliability of edge-aware large models—such as LLMs and VLMs—to enable their deployment in more specialized and federated environments. Additionally, expanding collaborations with both existing and new industrial partners will be key to translating research outcomes into deployable solutions, ensuring continued impact beyond the scope of this project.

7. Participating parties

The project consortium comprised Mälardalen University (MDU), Volvo Construction Equipment (VCE), and Zenseact (formerly Zenuity), bringing together leading academic expertise in AI and industrial leaders in construction machinery and autonomous driving technologies. The project was coordinated by Prof. Masoud Daneshtalab, Director of the DeepHERO Laboratory at MDU. The MDU team included Prof. Masoud Daneshtalab, Hamid Mousavi, and Ali Zoljodi, contributing to the research design, development, and evaluation of the AI frameworks. From Zenseact, Mina Alibeigi provided expertise in lane detection systems, while from Volvo Construction Equipment (VCE), Sara Afshar, Mohammad Loni, and Andreas Hjertström contributed domain knowledge, industrial datasets, and validation support for the targeted use cases.

8. References:

[1] Bruce Brown (October, 2017). Evidence stacks up in favor of self-driving cars in 2016 NHTSA fatality report. [Online]. Available: https://www.digitaltrends.com/cars/2016-nhtsa-fatality-report/

[2] The Official Site of The European New Car Assessment Programme, (2019), [Online]. Available: https://www.euroncap.com/en/

[3] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," *In Proceedings of the 40th International Conference on Software Engineering,* pp. 303–314, 2018.

[4] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 5079–5087, 2015.

[5] P. Viswanath, et al., "End to end learning for self-driving cars", *IEEE International Conference on Consumer Electronics (ICCE)*, 2016.

[6] W. Wu, H. Xu, S. Zhong, M. R. Lyu & I. King, "Deep validation: Toward detecting real-world corner cases for deep neural networks," *In 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN),* pp. 125-137, 2019.

[7] S. H. Mousavi, S. A. Mousavi, and M. Daneshtalab, "ProARD: progressive adversarial robustness distillation: provide wide range of robust students," *in Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2025, Italy.

[8] S. A. Mousavi, H. Mousavi, and M. Daneshtalab, "FARMUR: Fair Adversarial Retraining to Mitigate Unfairness in Robustness" *in proc. of 27th Springer European Conference on Advances in Databases and Information Systems (ADBIS)*, 2023, Spain.

[9] H. Mousavi, A. Zoljodi, and M. Daneshtalab, "Analysing robustness of tiny deep neural networks" *in proc. of 27th Springer European Conference on Advances in Databases and Information Systems (ADBIS)*, 2023, Spain.

[10] A. Zoljodi, S. Abadijou, M. Loni, M. Alibeigi, M. Daneshtalab, "3DLaneNAS: Neural Architecture Search for Accurate and Light-Weight 3D Lane Detection," *in Proceedings of 31st Springer International Conference on Artificial Neural Networks (ICANN)*, 2022, UK

[11] A. A. Sharifi, A. Zoljodi, M Daneshtalab, "DAT: Deep Learning-Based Acceleration-Aware Trajectory Forecasting," *Multidisciplinary Digital Publishing Institute Journal of Imaging* (*MDPI-imaging*), 2024. (IF: 3.3, Q1)

[12] A. A. Sharifi, A. Zoljodi, M Daneshtalab, “TrajectoryNAS: A Neural Architecture Search for Trajectory Prediction,” *Multidisciplinary Digital Publishing Institute Journal of Sensors* (*MDPI-Sensors*), 2024. (IF: 3.4, Q1)
[13] A. Zoljodi, S Abadijou, M Alibeigi, M Daneshtalab, “Contrastive Learning for Lane Detection via cross-similarity,” *Elsevier Journal of Pattern Recognition Letters* (*PRL-Elsevier*), 2024. (IF: 3.3, Q1)
[14] H. Mousavi, M. Loni, M. Alibeigi, M. Daneshtalab, “DASS: Differentiable Architecture Search for Sparse Neural Networks,” *ACM Transactions on Embedded Computing Systems* (*ACM-TECS*), Vol. 22, pp. 1-21, 2023 (IF: 2, Q1).