

차량 간 충돌 예측을 위한 실도로 및 가상 주행 데이터 기반 학습데이터 균형을 고려한 능동-커리큘럼 학습 연구

정영훈¹⁾ · 오나영¹⁾ · 송봉섭^{*2)} · 신장호³⁾

아주대학교 D.N.A.플러스융합학과 ¹⁾ · 아주대학교 미래모빌리티공학과 ²⁾ · 현대자동차 ³⁾

A Study on Active-curriculum Learning with Consideration of Dataset Balancing Based on Virtual and Real Driving Data for Vehicle Collision Prediction

Younghun Jeong¹⁾ · Nayoung Oh¹⁾ · Bongsob Song^{*2)} · Jangho Shin³⁾

¹⁾Department of Data, Networks, and AI, Ajou University, Gyeonggi 16499, Korea

²⁾Department of Mobility Engineering, Ajou University, Gyeonggi 16499, Korea

³⁾Advanced Safety Performance Development Team, Hyundai Motor Company, 150 Hyundaiyeonguso-ro,
Namyang-eup, Hwaseong-si, Gyeonggi 18280, Korea

(Received 29 October 2025 / Revised 7 January 2026 / Accepted 7 January 2026)

Abstract : This study proposes the application of active-curriculum learning methods in predicting vehicle collision that take into account dataset balancing to both virtual and field operational test (FOT) data. Since real collision events are rare and biased toward front-end collisions in the real-world environment, sole consideration of FOT data may result in data imbalance in the train and test for data-driven approaches. While several data balancing methods were proposed to overcome the imbalance due to over-sampling and under-sampling, learning strategies to consider both data structure and learning dynamics have evolved to improve prediction performance. A two-step training process is proposed to consider both data balance and learning strategy. First, both scenario-balanced and class-balanced training datasets are selected. Second, we adopt the iterative active-curriculum learning strategy, i.e., an error-driven active learning strategy that incrementally selects misclassified samples is combined with curriculum learning based on difficulty measures according to collision probability and/or road complexity. Finally, it is demonstrated in the FOT dataset that the proposed method allows enhanced collision prediction accuracy and significantly reduces the false alarm rate.

Key words : Active-curriculum learning(능동-커리큘럼 학습), Dataset balancing(데이터 균형), Collision prediction(충돌 예측), Active learning(능동 학습), Curriculum learning(커리큘럼 학습)

1. 서론

최근 자동 긴급 제동(Automated Emergency Braking, AEB)과 같은 능동형 안전 시스템(Active safety system)의 상용화로 인해 차량의 안전성이 크게 향상되었다. 실제 2015년부터 2023년까지 9년간 후방 추돌 사고의 발생률은 평균 49% 감소하였으며, 이러한 추세는 최신 차량일 수록 더욱 뚜렷하게 나타난다.¹⁾ 이러한 기술적 발전에 따라 미국 도로교통안전국(NHTSA)은 2029년까지 AEB의 적용 범위를 약 100 km/h의 고속 주행 상황까지 확대할

것을 의무화한다고 발표하였다.²⁾

최근 딥러닝 기술의 발전으로 능동형 안전 시스템에도 이를 적용하려는 시도가 이루어지고 있으나, 단 한 번의 오작동이 치명적인 사고로 이어질 수 있는 특성상 이러한 시스템을 어떻게 설계하고 신뢰성 있게 검증할 것인가가 주요한 과제로 주목을 받고 있다. 특히 딥러닝 기반 알고리즘의 성능은 네트워크 구조뿐만 아니라 데이터의 품질에 따라 성능이 좌우된다고 알려져 있다.³⁾ 나아가 실도로 주행을 통해 수집된 데이터에는 센서 노이즈나 객

*Corresponding author, E-mail: bsong@ajou.ac.kr

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium provided the original work is properly cited.

체 검출 실패 등 환경적 요인으로 인한 센서 성능 저하가 포함되어 있어 이에 대한 정밀한 검증의 중요성이 더욱 강조되고 있다.

충돌 예측 기술은 크게 물리 기반 모델과 데이터 기반 모델로 구분된다. 물리 기반 모델은 차량의 속도, 거리, 가속도 등의 물리적 정보를 이용하여 기구학적(Kinematic) 또는 확률적(Stochastic) 접근방법에 기반하여 충돌 가능성을 판단한다.^{4,5)} 이러한 모델은 해석 가능성과 계산 효율성이 높다는 장점이 있으나, 복잡한 교통상황을 충분히 반영하기 어렵다는 한계를 가진다. 데이터 기반 모델 중에서는 Multi-Layer Perceptron(MLP) 및 Convolution Neural Network(CNN) 구조를 기반으로 한 알고리즘이 제안되었다.^{6,7)} 이후 조감도(Bird's Eye View, BEV) 형태로 입력 정보를 변환한 후 CNN 구조로 충돌 위험을 예측하는 방법도 제안되었다.⁸⁾ 더 나아가, 주변 차량 수가 늘어나는 복잡한 상황에서 충돌 예측을 위한 추론 속도를 제고하기 위해 비동기 트랜스포머(Transformer) 기반 구조가 제안되었다.⁹⁾ 이러한 데이터 기반 모델은 학습 데이터의 구성과 전략에 따라 성능이 크게 달라지는 것으로 알려져 있다. 특히, 실도로 주행 데이터셋의 특성 상, 충돌 상황이 포함되지 않는 경우가 많아 안전/위험 혹은 충돌로 구분되는 클래스 간의 불균형이 불가피하게 발생한다.^{10,11)} 이러한 한계점을 극복하기 위하여 충돌 데이터의 경우 일반적으로 시뮬레이션을 통해 가상 데이터를 생성하고, 데이터의 라벨 불균형 문제를 해결하기 위해 오버샘플링과 언더샘플링을 이용한 클래스 밸런싱(Class balancing) 기법이 제안되고 있다.¹²⁻¹⁴⁾ 일반적으로 데이터 기반 접근법의 성능 개선 방법은 크게 네트워크 구조의 설계 및 최적화, 학습방법 및 데이터 증강(Data augmentation), 전처리(Pre-processing) 및 후처리(Post-processing) 알고리즘 개선의 세 가지로 나눌 수 있다.¹⁵⁾ 이에 따라 데이터의 구성과 학습 순서까지 고려한 학습 전략들이 제안되고 있는데 대표적으로 능동 학습(Active Learning, AL)과 커리큘럼 학습(Curriculum Learning, CL)이 있다.¹⁶⁻¹⁸⁾

능동 학습은 학습 알고리즘이 정보량이 높다고 판단되는 데이터를 라벨링하고 학습데이터로 선택함으로써, 최소한의 라벨링으로 최대의 학습 효율을 달성하는 방법이다. 충돌 예측 네트워크에도 이러한 학습 방식이 적용되었다.⁸⁾ 해당 연구는 알고리즘이 안전한 데이터에 대해 충돌로 잘못 판단한 데이터를 기존 데이터셋에 추가하여 학습하는 방식으로, 가상 및 실도로 주행 데이터에 대해 충돌 예측 성능을 효과적으로 향상시켰다. 그러나 실도로 주행 데이터에는 센서 노이즈로 인해 품질이 저하된 샘플이 포함될 수 있고, 이 경우 잘못 판단된 샘플을 반복적으로 능동 학습에 포함시키는 것만으로는 성능 향

상에 한계가 있다. 능동 학습은 본질적으로 모델의 성능 향상에 도움이 될 것으로 예상되는 정보성(Informativeness)이 높은 데이터를 선택하는 데에 초점을 두기 때문이다. 이러한 접근은 효율적일 수 있으나, 모델이 노이즈를 정보량이 높은 데이터로 오인할 경우 일반화 성능 저하의 위험이 있다.¹⁹⁾ 더불어 학습 데이터 전반의 분포를 충분히 반영하지 못한다는 한계가 있다.²⁰⁾ 이러한 문제를 완화하기 위해, 데이터의 난이도와 학습 순서를 체계적으로 조정함으로써 안정적인 성능 향상을 유도하는 커리큘럼 학습이 도입되었다.

커리큘럼 학습은 학습 데이터를 난이도에 따라 정렬한 뒤, 쉬운 샘플부터 점진적으로 어려운 샘플을 학습하도록 하는 방식이다. 이는 인간의 학습 과정에서 착안된 방식으로, 모델의 수렴 속도를 향상시키고 더 나은 지역 최적해(Local optimum)에 도달하게 함으로써 초기 학습 단계에서 손실 함수(Loss function)를 부드럽게(Smoothing) 만들어 일반화 성능 개선에 효과적이다.^{21,22)} 또한 센서 노이즈로 데이터 품질이 떨어지는 샘플이 학습 후반부에 배치되도록 설계함으로써, 노이즈에 강인한 모델을 학습시킬 수 있다는 장점이 있다.

능동 학습만의 한계를 보완하기 위해 커리큘럼 학습의 개념을 결합한 연구가 진행되었다. 기존 능동 학습은 불확실성이 높은 데이터를 위주로 학습을 진행하여 계속해서 많은 라벨링 데이터가 필요하다는 한계가 있다. 이를 개선하고자 이미지 분류 분야에서 ACAL(Adaptive Curriculum Query Strategy for Active Learning) 기법이 제안되었다.²³⁾ 이 방식은 학습 초기에 다양한 난이도의 샘플을 폭넓게 포함시켜 모델이 데이터의 일반적인 패턴과 구조를 빠르게 학습하도록 유도한 후, 이후 불확실성 기반 샘플 선택으로 전환해 점진적으로 어려운 데이터를 학습한다. 선택된 데이터의 분포가 전체 데이터의 분포와 유사한지를 판단하기 위해, DSM(Distribution Similarity Monitor) 모듈을 도입하여 학습 단계별 전략을 자동 전환하는 구조를 갖는다. 이는 적은 라벨로 높은 정확도와 빠른 일반화를 보여주었다.

이러한 연구에서 기인하여 본 연구에서는 학습방법을 개선함으로써 충돌 예측 알고리즘의 성능을 향상시킬 수 있음을 보이고자 한다. 이를 위해 선행 연구의 알고리즘을 비교 모델로 설정하고,⁸⁾ 위험과 안전으로 구분되는 이분법적 분류(Classification) 문제에서 제안하는 학습 전략의 성능 개선 효과를 살펴보고자 한다. 더 구체적으로, 능동 학습(AL)과 커리큘럼 학습(CL)의 통합 방식을 통해 “간단하고 예측 가능한(Simple & expected) 위험”에서 “복잡하고 예측이 쉽지 않은(Complex & unexpected) 위험” 데이터로의 순차적 또는 반복적 학습을 통하여 차량 간 충

돌 예측 성능이 향상됨을 보이고자 한다.

2. 기존 연구 및 문제 정의

차량 간 충돌 예측은 Fig. 1과 같이 충돌 직전의 상황을 정의한 프리-크래시(Pre-crash) 시나리오를 기반으로 수행된다. 본 연구에서는 프리-크래시 시나리오의 시작을 자차와 상대 차량 간 Time-To-Collision(TTC)이 1.5초가 되는 시점 t_1 으로 정의했다.⁸⁾ 시점 t_2 는 충돌 시점, t_c 는 처음으로 충돌을 판단하는 시점이며, t_d 는 $t_2 - t_c$ 로 정의하며 충돌 시점을 얼마나 빨리 예측했는지를 나타내는 예측 조기성 평가 지표로 사용된다. 또한 위험과 안전이라는 판단문제에 대한 성능 평가를 위해 Table 1과 같은 평가 매트릭스(Confusion matrix)를 사용하였다. 특히 본 연구에서는 학습전략에 따라 실도로 주행데이터에 대해서 False Positive(FP), 즉 안전한 상황을 충돌 상황으로 판단하는 오판단(False alarm)을 최소화하는 데 중점을 두고 집중적으로 평가를 진행하고자 한다. 동시에 충돌 시나리오를 포함하고 있는 가상 데이터셋에 대한 판단 성능 변화도 동시에 확인하고자 한다.

학습전략을 평가하기 위해 선정한 충돌 예측 네트워크(CP-CNN)는 모델 기반과 데이터 기반 접근을 통합한 구조로 구성되어 있다.⁸⁾ 구체적으로, 주어진 환경 센서 입력을 기반으로 궤적 예측 및 충돌 확률(Collision Probability, CP)을 산출한 후, 이를 종합적으로 반영한 이미지(Bird's eye view)를 생성하고, 해당 이미지를 CNN 네트워크에 입력하여 충돌 여부를 판단한다. 네트워크 학습을 위해 Table 2에 제시된 바와 같이 가상 및 실도로 주행(FOT) 데이터를 모두 포함한 복합 데이터셋(16,739개)을 구성하였다. 이 때 FOT 데이터는 Fig. 2와 같이 전방(Field of view ± 45 deg) 및 코너 레이더(Field of view ± 75 deg), 비전센서(Field of view ± 50 deg)가 장착된 실험 차량을 이용하여 취득하였으며, 20초 길이의 주행 구간을

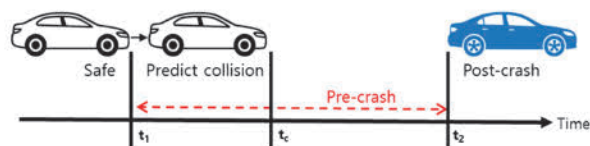


Fig. 1 Definition of pre-crash scenario for collision prediction

Table 1 Confusion matrix for collision prediction

		Annotation	
		Unsafe	Safe
Predict	Unsafe	True Positive (TP)	False Positive (FP)
	Safe	False Negative (FN)	True Negative (TN)

하나의 스니펫(Snippet)으로 정의하여 구성하였다. 전체 복합 데이터 중 18.9%의 초기(Initial) 학습 데이터셋을 랜덤하게 선정하여 학습을 하였으며, Fig. 3에서 보는 바와 같이 동일한 테스트셋에 대해 약 77~88%의 정확도 편차를 보였다. 이후 3번의 능동 학습을 반복하여 적용하였다. 2차 능동 학습 시 네 가지 다른 학습 데이터셋에 대해서 모두 성능 향상을 뚜렷하게 보여주지만 이후 동일 학습을 반복하더라도 성능이 향상되지 않았다. 이러한 한계는 정확도가 높아 능동 학습에 사용되기 위하여 추가되는 학습데이터 샘플(0.5~2% 이내)이 상대적으로 적은 경우 더욱 두드러진다. 생성형(Generative) 능동 학습 기법을 적용해 학습데이터를 증강함으로써 성능을 개선할 수도 있으나, 본 연구에서는 데이터 증강이 아닌 학습 전략적 측면에서 알고리즘 성능 향상 가능성을 검토하는 데 초점을 두고자 한다.

실도로 주행(FOT) 데이터의 경우, 특히 능동 학습만으로는 성능이 개선되지 않는 사례가 반복적으로 발생하

Table 2 Statistics of virtual and real driving dataset

	Training		Test	Sum
	Initial	Remaining		
Virtual (# of scenarios)	3,177	3,253	6,356	12,786
FOT (# of snippet)	-	1,977	1,976	3,953

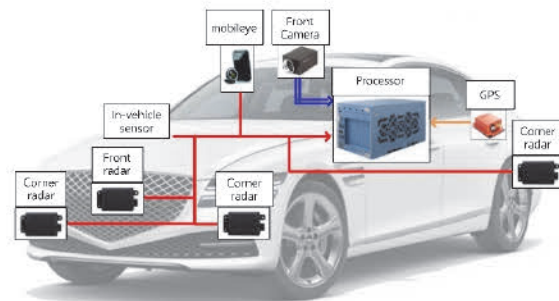


Fig. 2 Configuration of a test vehicle

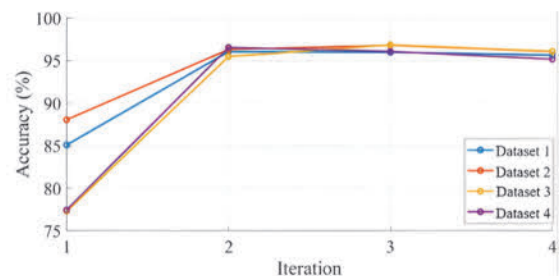


Fig. 3 Accuracy per iteration to active learning

였는데 이는 센서의 물리적 한계에서 비롯되는 두 가지 대표적 증상으로 구분할 수 있다.²⁴⁾ 첫번째는 지속적으로 발생하는 고장(Fault)과는 달리 순간적으로 발생하는 (Abrupt) 센서 노이즈이다. 해당 예시는 Fig. 4에 제시하였으며, 이는 앞서 언급한 충돌 예측 네트워크인 CP-CNN의 판단 과정 중 센서 노이즈로 인해 발생할 수 있는 오판단 사례를 보여준다. Fig. 4의 x축은 50ms 간격의 연속 샘플을 의미하며, Fig. 4(a), (b)에서 볼 수 있듯 실제로 주행 중 반대편에서 접근하는 트럭의 횡방향 위치가 몇 샘플 간 순간적으로 잘못 측정된 구간이 발생하였다. 이로부터 Fig. 4(c)의 물리기반 알고리즘(Collision Probability, CP)이 비정상적으로 높게 계산되었고, 잘못된 궤적 및 CP 정보가 이미지 형태로 네트워크에 입력되었다. 결과적으로 Fig. 4(d)와 같이 CP-CNN이 생성된 이미지를 기반으로 순간적으로 충돌로 잘못 판단한 사례를 보여준다. 두 번째는 주기적으로 발생하는(Incipient) 센서 노이즈로 일반

적인 센서가 포함하고 있는 센서 노이즈라 볼 수 있다. Abrupt/Incipient의 구분은 고장 진단(Fault detection)에서 고장의 형태를 구분하는 용어로 널리 사용되지만 이러한 센서의 불확실성이 시계열 관점에서 지속되면 고장으로, 일시적으로 나타나면 노이즈로 구분하기에 본 논문에서는 동일한 용어를 사용하였다.

따라서 본 연구에서는 Fig. 3에 제시된 단순(Vanilla) 능동 학습의 한계를 보완하고자 센서의 특성, 즉 불확실성의 특징을 고려한 학습 전략을 제안하고자 한다. 위험 상황을 포함한 가상 데이터에 대한 성능 저하를 최소화함과 동시에 실제로 주행 데이터, 즉 안전 시나리오만 포함한 데이터에 대한 충돌 예측 성능의 False Positive Rate (FPR)를 제고할 수 있는 학습 전략을 제시한다. 이를 통해 모델이 위험 상황에 대해서는 민감하게 반응하면서도, 안전 상황에 대해서는 불필요한 경고를 줄일 수 있는 균형 잡힌 학습 구조를 구현하고자 한다.

3. 학습데이터 균형 및 커리큘럼 학습

3.1 충돌 예측 네트워크와 충돌 확률

차량 간 충돌 예측 성능 제고를 위한 학습 방법의 개선 가능성을 검토하고자, 선행연구인 CP-CNN⁸⁾을 본 연구의 기준 네트워크로 선정하였으며 전체적인 구조는 Fig. 5와 같다. 입력의 경우 Fig. 2에 제시한 차량 플랫폼의 데이터와 비전 센서의 트랙(Track) 정보를 입력으로 하며, 센서융합(Sensor fusion)을 통해 주변 차량의 통합된 트랙 정보를 생성한다. 이후 칼만필터(Kalman filter)를 통해 상대 차량의 미래 위치 정보를 예측(Trajectory prediction)하고, 이를 기반으로 충돌 확률(Collision Probability, CP)을 계산한다. 다음 단계로 센서융합 결과, 예측 미래 위치 그리고 충돌 확률 정보를 종합하여 이미지 형태의 단순조감도(Simplified Bird Eye's View, SBEV)로 추상화(Abstraction)한다. 추상화된 정보는 세 가지 계층으로 구성되는데, 비전 센서에서 제공되는 차선 관련 정보인 정적(Static) 계층 정보, 센서융합에서 제공되는 위치 및 속도 정보인 동적(Dynamic) 계층 정보, 그리고 미래 위치 및 충돌 확률 정보인 메타(Meta) 계층 정보로 분류된다. 마지막으로 SBEV 이미지는 CNN 구조 기반 분류(Classification) 모듈에 입력되어, 최종적으로 충돌이 발생하는 경우 Fig. 6과 같이 12개로 구분되어 있는 충돌 부위(Collision mode) 중 1개의 부위만을 출력한다. 만약 11, 12, 13과 같이 여러 부위가 동시에 포함되는 상황에서는 가장 큰 면적을 차지하는 부위 1개만을 선정하여 출력한다. 본 연구에서 충돌 확률(CP)을 센서 노이즈와 직·간접적으로 연관된 지표로 보고, 더 나아가 데이터 품질이 낮

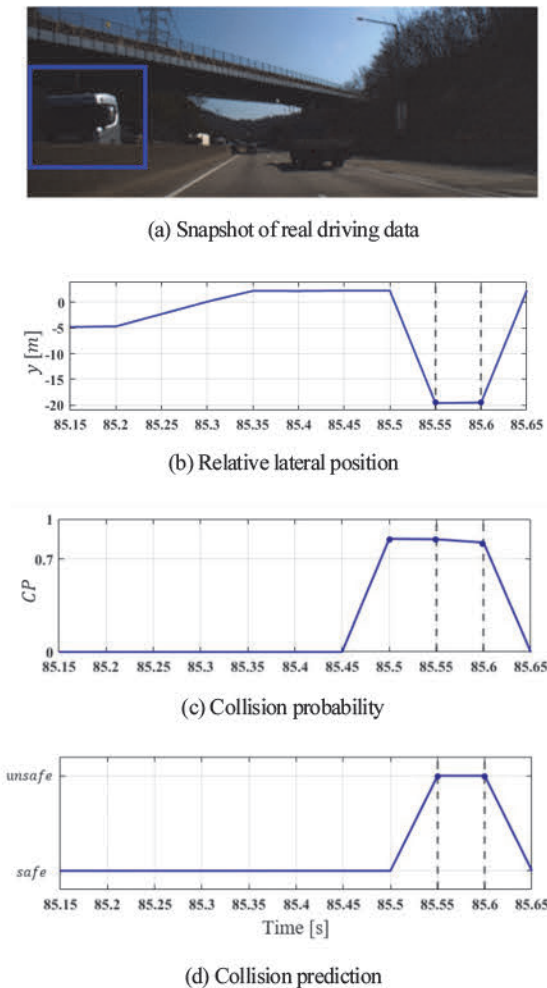


Fig. 4 A false alarm example due to abrupt sensor noise

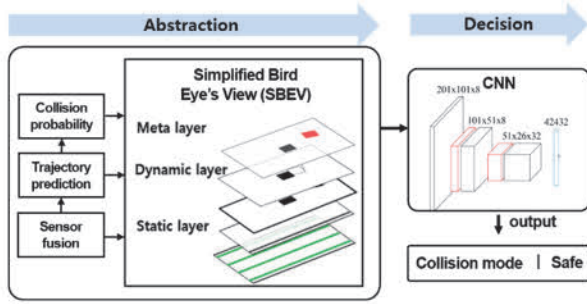


Fig. 5 Schematic diagram for collision prediction

13	23	33	43	53
12				52
11	21	31	41	51

Fig. 6 Definition of collision mode

거나 분포상 상대적으로 희귀한 데이터를 선정할 수 있는 지표(Measure)로 사용하고자 한다. 더불어 데이터의 난이도(Difficulty)를 정량적으로 표현하는 지표로 확장하여, 이후 커리큘럼 학습에서 난이도 기반 학습순서를 설정하는 방법의 하나로 응용하고자 한다. 충돌 확률은 자차와 상대차량 간 미래 위치 확률분포가 일정 충돌 영역 D 에 속할 확률을 의미하며 다음과 같이 정의된다.⁵⁾

$$CP(t) = \max_{i=1, \dots, N} \Pr(x_{t+iT} \in D) \quad (1)$$

여기서 i 는 미래 예측 단계, N 은 총 예측 단계 수를 나타낸다. 상대 상태 벡터 $x_{t+iT} = [\tilde{p}_x \tilde{p}_y]^T \in \mathfrak{R}^2$ 로 정의되며, 자차량 기준 좌표계에서의 종방향(Longitudinal) 및 횡방향(Lateral) 상대 위치를 포함한다. 상대 차량의 미래 상태는 등가속도(Constant acceleration) 운동모델을 기반으로 한 칼만 필터를 통해 추정되며, 이로부터 평균벡터 μ_i 와 공분산행렬 Σ_i 가 산출된다. 미래 상대 상태의 확률 밀도함수(Probability Density Function, PDF)는 μ_i 와 Σ_i 를 갖는 이변량 정규분포로 가정된다. 이에 따라, 시점 $t+iT$ 에서의 충돌 확률은 다음과 같이 정의된다.

$$\Pr(x_{t+iT} \in D) = \iint_{(\tilde{p}_x, \tilde{p}_y) \in D} p_{t+iT}(\tilde{p}_x, \tilde{p}_y | Y_t) d\tilde{p}_x d\tilde{p}_y \quad (2)$$

여기서 $p_{t+iT}(\tilde{p}_x, \tilde{p}_y | Y_t)$ 는 관측치 Y_t 가 주어졌을 때 $[\tilde{p}_x \tilde{p}_y]^T$ 의 확률분포를 나타내는 확률밀도함수이며, 확

률분포의 적분은 영역 D 의 x 축 및 y 축 방향의 경계 값에서 평가된 해당 누적분포함수(Cumulative Distribution Function, CDF)를 이용한다. 충돌 영역 D 는 자차와 상대 차량의 실제 차체 형상 및 예측 상대 위치 관계를 뜻하며, 본 연구에서는 차체 형상의 교차 영역을 자차 기준(Body-fixed) 좌표계에 대하여 사각 영역으로 간소화하였다.

$$D = [x_{min}, x_{max}] \times [y_{min}, y_{max}] \quad (3)$$

구체적으로, 식(2)에서 주어진 이변량 적분을 각 축의 1차원 CDF로 분리하여 충돌확률(CP)는 다음과 같이 근사하여 계산하였다.

$$\Pr(x_{t+iT} \in D) = [F_x(x_{max}) - F_x(x_{min})] \times [F_y(y_{max}) - F_y(y_{min})] \quad (4)$$

여기서 F 는 각 축에 대한 정규 누적분포함수이다. 최종적으로 CP 는 식 (1)과 같이 각 예측단계에서 계산된 확률값 중 최대값으로 정의되며, 이에 대한 예시로 Fig. 4(c)에서 CP 계산 결과를 보여주고 있다. 특히 Fig. 4(b)에서 횡방향 위치의 잘못된 추정값이 CP 의 잘못된 계산을 유도하여 Fig. 4(d)와 같이 판단 알고리즘에서도 충돌로 잘못 판단함을 확인할 수 있다.

3.2 학습데이터 균형

본 연구에서는 가상 데이터와 실도로 주행 데이터를 모두 포함하는 복합(Mixed) 프리-크래시(Pre-crash) 데이터셋을 학습 및 평가에 사용하였다.⁸⁾ 가상 데이터의 경우 국내 교통사고 분석시스템(TASS)와 유럽의 교통사고 데이터(IGLAD)를 기반으로 교차로 및 비교차로의 14개 시나리오를 선정하고, IPG CarMaker를 이용해 안전과 충돌 데이터를 같은 비율로 생성하였다.^{25,27} 반면 실도로 주행 데이터의 경우 Fig. 2와 같이 레이더 및 카메라를 장착한 차량 플랫폼을 이용하여 수집하였으며 사용한 데이터의 상세 구성은 Table 2에 요약되어 있다. 이 때 실도로 데이터에는 모두 안전한 데이터만 포함되어 있다. 학습 데이터셋은 1차적으로 복합(Mixed) 데이터셋에 대해 일정 비율에 따라 학습(Training) 데이터셋과 평가(Test) 데이터셋으로 구분하여 구성한다. 다음으로 선정된 학습 데이터셋을 다시 초기(Initial) 및 후반(Remaining) 학습 데이터셋으로 구분한다. 초기 학습 데이터셋은 충돌에 대한 정보를 충분히 학습하고자 가상 데이터셋에 대해서만 데이터 균형(Data balancing) 전략을 고려해 학습데이터를 선정하고, 후반 학습 데이터셋은 가상 및 실도로 주행 데이터를 모두 포함하는 복합 데이터셋으로 구성하였다. 이 때

실제 데이터 분포는 시나리오별 빈도와 충돌 모드별 클래스 분포 상 차이가 존재하기 때문에, 두 관점을 동시에 고려하지 않을 경우 특정 시나리오 또는 특정 충돌 모드가 과도하게 학습되는 편향이 발생할 수 있다. 이를 위해 우선 시나리오 관점에서, 초기 학습 데이터셋을 선정할 시나리오에 대해 모두 포함하도록 구성하였다. Fig. 7에서 보는 바와 같이 미리 정의된 14개의 시나리오(Logical scenario)에 대해 상세(Concrete) 시나리오가 초기 학습 데이터셋에 골고루 포함되도록 학습 데이터를 선정한다.

다음으로 시나리오 별 데이터 균형뿐만 아니라 충돌 데이터가 상대적으로 희귀하다는 점을 고려해 클래스 관점에서 데이터 균형을 고려하였다. 이를 위해 먼저 안전 및 충돌 데이터를 500 ms, 50 ms와 같이 서로 다른 샘플링 간격으로 추출하였고, 그 결과 Table 3과 같이 53 %와 47%의 비율로 이미지 학습 데이터셋(SBEV)을 구성하였다. 더 나아가, 실제 충돌은 360도 전방위에서 발생할 수 있으며 충돌 부위(Collision mode)도 Fig. 6에서 보이는 바와 같이 정의할 수 있다. 이에 따라 본 연구에서는 초기 데이터셋 선정에 한해 충돌 유무뿐만 아니라 12개의 충돌 모드 분포에 대한 다양성을 고려한 Class-balanced 학습 데이터셋을 구성하였다. 구체적으로, Table 3에서 보는 바와 같이 12개의 충돌 모드별로 각 2,000개의 학습 데이터를 랜덤하게 선정하여 충돌에 해당하는 총 24,000개의 학습 데이터를 재구성하였다. 최종적으로 초기 학습 데이터셋을 구성하기 위하여 시나리오관점의 균형

(Scenario-balanced)과 클래스 관점(즉, 충돌 모드)의 균형(Class-balanced)을 동시에 고려하였다.

3.3 커리큘럼 학습 설계

커리큘럼 학습의 핵심은 데이터에 난이도를 부여하는 방식과 학습 스케줄링 방식에 있다.¹⁸⁾ 데이터의 난이도를 정량화 한 난이도 지표(Difficulty measure)는 주로 도로 환경, 주변 차량 밀도 등의 정량적 지표가 사용된다. 학습 스케줄링은 이러한 난이도 지표를 바탕으로, 학습 시점 별로 어떤 데이터를 언제 얼마나 학습할지 결정하는 역할을 하며 일반적으로 이산형 또는 연속형 방식으로 구분된다.

3.3.1 난이도 지표

본 논문에서는 난이도 지표를 구성하기 위해 위험도, 복잡도, 도로지형 지표를 사용하였다. 우선 위험도 지표는 충돌 위험을 정량적으로 계산하는 데 사용되며, 본 연구에서는 대표적인 예측 기반 지표인 차량 간 충돌 예측 시간을 계산하는 TTC(Time-To-Collision)와 충돌 확률 CP를 사용하였다. 난이도는 Table 1에서와 같이 충돌이 발생하는 경우($GT = unsafe$)와 그렇지 않은 경우($GT = safe$)로 구분되며, 위험도 계산 결과에 따라 TP나 TN의 경우에 낮은 난이도를, FP나 FN에 높은 난이도를 부여하는 방법을 제안한다. 이를 반영하기 위하여 TTC에 기반한 난이도 지표 $C_{TTC}(z)$ 를 다음과 같이 정의할 수 있다.

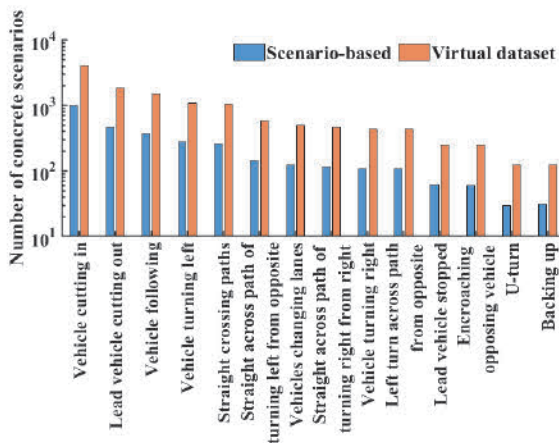


Fig. 7 Distribution of initial training data with respect to scenario

Table 3 Number of samples in the initial training dataset

	Unsafe	Safe	Total
Scenario-balanced	22,749	20,838	43,587
Class-balanced	24,000	20,838	44,838

$$C_{TTC}(z) = \begin{cases} \max\{0, 1 - TTC(z)^{-1}\}, & \text{if } GT(z) = unsafe \\ \min\{1, TTC(z)^{-1}\}, & \text{if } GT(z) = safe \end{cases} \quad (3)$$

여기서 샘플 데이터 z 에 대한 TTC의 계산 결과를 역수로 변환하여 안전한 상황에서 $TTC(z)^{-1}$ 를 이용하여 C_{TTC} 가 작은 값이 나오도록 하고, 충돌이 발생하는 경우는 $1 - TTC(z)^{-1}$ 를 이용하여 작은 값이 나오도록 설정하였다. 비슷한 방식으로 CP를 활용한 난이도 지표 또한 안전한 상황에서는 CP를 이용하고 충돌이 발생하는 경우는 $1 - CP$ 를 이용하여 해당 난이도 지표를 정의하였다.

$$C_{CP}(z) = \begin{cases} 1 - CP(z), & \text{if } GT(z) = unsafe \\ CP(z), & \text{if } GT(z) = safe \end{cases} \quad (4)$$

여기서 $CP(z)$ 는 식 (1)을 이용한 CP의 예측 결과이다. 다음으로 복잡도 지표를 데이터 난이도 지표로 활용하였다. 복잡도 지표는 주행 환경 내 상호작용 정도를 반영하는 요소이며 본 연구에서는 주변 차량이나 보행자 등의 교통 밀집도(Crowdedness)를 평균적으로 정량화하

는 방법을 사용하였다.²⁸⁾

$$C_{crowd} = \frac{1}{T} \sum_{t=1}^T |D_t| \quad (5)$$

여기서 자차 주변 관심 영역(ROI) 내 객체 수를 시간 평균하여 산출하며, T 는 전체 프레임 수, D_t 는 프레임 t 에서 탐지된 객체의 수를 의미한다. 자차 주변에 존재하는 객체의 수가 많을수록 복잡하다고 판단하여 데이터 난이도를 높게 설정하였다.

마지막으로 주행경로의 기하학적 난이도를 반영하는 척도로 도로곡률 C_{curve} 을 난이도 지표로 사용하였다. 이는 관심 영역 내 차선 중심선 또는 계획 경로의 곡률 크기를 집계해 도로 기하학적 복잡도를 정량화한 값으로, 시점 p 에 대해 이산 샘플 $\{z_t\}_{t=1}^N$ 가 주어질 경우 다음과 같이 근사된다.²⁸⁾

$$C_{curve} = \mu(\mathcal{K}_C) + \mu(\dot{\mathcal{K}}_C) \quad (6)$$

여기서 \mathcal{K}_C 는 각 시점에서의 곡률(Curvature)을, $\dot{\mathcal{K}}_C$ 는 곡률 변화율을 의미한다. 일반적으로 곡률이 클수록 주행 경로의 조향 난이도가 높아지므로, 해당 지표는 도로의 기하학적 형태가 운전자의 주행 난이도에 미치는 복합적인 영향을 반영한 지표라고 할 수 있다. 위와 같이 계산된 C_{TTC} , C_{crowd} , C_{curve} 는 정규화를 통해 $[0, 1]$ 범위로 조정한 후, 이후 스케줄러 규칙에 따라 활용될 수 있도록 난이도 지표로 정의하였다.

3.3.2 학습 스케줄러

커리큘럼 학습에 사용되는 학습 스케줄러는 크게 이산형(Discrete)과 연속형(Continuous) 방식으로 구분된다.³⁰⁾ 대표적인 이산형 방식으로 Baby-step 스케줄러가 사용되고 있으며, 전체 데이터를 난이도에 따라 여러 단계로 나누고(Split) 각 단계별로 모델이 특정 조건을 만족할 때 다음 단계로 넘어가도록 하며,²⁹⁾ 본 연구에서는 정확도(Accuracy) 임계값을 90%로 설정하였다. Baby-step 스케줄러는 난이도 지표 $\mathcal{C}(z)$ 에 대하여 m 개의 구간으로 나누고, 각 구간의 임계값을 $\lambda_k \in (0, 1]$ ($k = 1, \dots, m$)로 정의한다. 이에 따라 각 학습 단계 s 에서 사용되는 학습 데이터셋은 다음과 같이 정의된다.

$$D_{tr}^{(s)} = \bigcup_{k=1}^s D^{(k)} \quad (7)$$

여기서 $D_{tr}^{(s)}$ 는 학습 단계 s 까지 누적된 학습 데이터셋을 의미하며, $D^{(k)}$ 는 식 (8)에서 정의한 각 단계의 데이터 집합으로, 이를 단계별로 누적함으로써 $D_{tr}^{(s)}$ 를 구성한다.

$$D^{(k)} = \{z | \lambda_{k-1} \leq \mathcal{C}(z) < \lambda_k\} \quad (8)$$

여기서 $\lambda_0 = 0$ 이며 $\lambda_m = 1$ 으로 정의한다.

Fig. 8은 식 (4)를 기반으로 CP를 난이도 지표로 선정하였을 때 학습 데이터셋을 10단계로 ($m = 10$) 구분한 예시를 보여준다. Fig. 8(a)는 자차량이 차로를 유지하며 주행할 때 우측 차로의 차량이 끼어드는 상황으로, CP가 0.1 미만으로 계산되며 D^1 데이터로 분류되는 경우를 보여주고 있다. Fig. 8(b)의 경우 상대차량의 끼어들기로 인해 실제 충돌이 발생한 가상 시나리오 데이터로, CP가 0.8~0.9 사이 값을 가지며, 식 (4)에서 정의한 난이도 기준에 따라 D^2 데이터셋으로 분류된 사례를 보여준다. 반면 Fig. 8(c)는 Fig. 4에서 제시된 사례로, 실제 안전한 상황이지만 (a)와 다르게 반대편 차로에서 접근하는 트럭의 횡방향 위치가 잘못 측정되어 CP가 0.8~0.9 사이 값으로 높게 계산된 상황이다. 이를 주어진 식(4)에 적용하면 데이터 난이도가 높은 D^8 데이터로 분류된다.

다음으로 연속형 방식은 난이도 임계값을 선형(Linear) 혹은 제곱근(Root) 함수 형태로 증가시켜 학습 범위를 점진적으로 확장하는 방식이다. 학습 진행 단계 s 에 따라 난이도 임계 함수 $\lambda(s)$ 을 연속적으로 증가시키며 학습에 포함되는 데이터 난이도의 상한선을 단계적으로 확장한다. 즉, 이산형 방식의 λ_k 와 같이 난이도 구간을 고정된 개수로 분할하지 않고, $\mathcal{C}(z) \leq \lambda(s)$ 를 만족하는 데이터만을 학습에 포함시키며 학습이 진행됨에 따라 난이도가 높은 샘플이 추가된다. 연속형 스케줄러에서의 초기 임계 값 λ_0 는 학습 초기에 허용되는 난이도의 시작점을 의미하며, 이산 스케줄러에서의 λ_0 와는 개념적으로 구분된다.

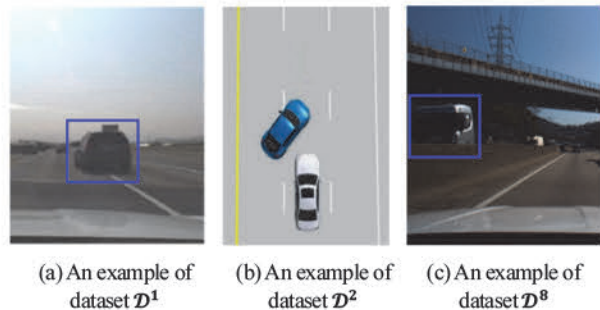


Fig. 8 Examples of training dataset based on collision probability

$$\lambda_{linear}(s) = \min\left(1, \lambda_0 + \frac{1 - \lambda_0}{T_{grow}} \cdot s\right) \quad (9)$$

$$\lambda_{root}(s) = \min\left(1, \sqrt{\frac{1 - \lambda_0^2}{T_{grow}} \cdot s + \lambda_0^2}\right) \quad (10)$$

여기서 T_{grow} 는 난이도 임계값이 최댓값 1에 도달하기까지의 총 학습 단계 수이며, s 는 현재 학습 단계를 의미한다. 매 시점 s 에서 학습에 사용되는 데이터셋은 다음과 같이 정의된다.

$$\mathcal{D}_{tr}(s) = \{z | \mathcal{C}(z) \leq \lambda(s)\} \quad (11)$$

이로써 커리큘럼 학습 진행 단계에 따라 학습 구조가 자동적으로 반영되며, 난이도 기반 데이터 샘플 선택이 연속적으로 이루어진다.

4. 능동-커리큘럼 학습

본 절에서는 Fig. 3에서 제시한 단순(Vanilla) 능동학습의 한계점을 개선하기 위하여, 능동학습과 커리큘럼 학습을 혼용함으로써 실제로 주행 데이터에 대한 오판단을 줄이기 위한 방법을 제시하고자 한다. 즉, 순차적인 혼용과 반복적인 혼용 방법을 각각 제시하고 이후에 성능 비교를 수행하고자 한다.

4.1 순차형 능동-커리큘럼 학습

순차형(Sequential) 능동-커리큘럼 학습(Active-curriculum Learning, ACL)은 크게 세 단계로 구성되어 있다. 우선 데이터 균형을 고려한 초기 학습데이터를 이용하여 네트워크 학습을 진행한다. 다음으로는 후반 학습데이터에 대해서 능동 학습을 진행하여 커리큘럼 학습에 이용될 학습 데이터셋을 구성한다. 마지막으로 난이도 지표와 학습 스케줄러를 고려하여 커리큘럼 학습을 적용한다. 이를 Pseudo code 형태로 정리하면 Algorithm 1과 같다. 먼저 가상 데이터로만 구성된 Table 2에 제시한 초기 데이터셋(\mathcal{D}^{ini})으로 모델을 학습한다. 이후 2차 학습으로 가상 및 실도로 데이터로 구성된 후반 데이터셋(\mathcal{D}^{rem})에서 특정 데이터를 선별(query)하여 \mathcal{D}^{set} 을 구성한다. 이 때 특정 데이터는 초기 평가결과를 반영하여 구성하였는데, Table 1에 제시한 지표에서 FN(False Negative)의 비율보다 FP(False Positive)가 매우 큰 것을 반영하여 Fig. 4와 같은 FP 데이터만을 선별하였다. 이렇게 선별한 \mathcal{D}^{set} 을 기존 \mathcal{D}^{ini} 에 추가하여 학습 데이터를 확장하는 능

Algorithm 1: Sequential active-curriculum learning

Input : \mathcal{D}^{ini} : virtual dataset for initial training,
 \mathcal{D}^{rem} : virtual and real dataset for remaining training;
Output : \mathcal{M}^* : optimal model

- 1: $\mathcal{M} \leftarrow \text{train}(\mathcal{M}, \mathcal{D}^{ini});$
- 2: $\mathcal{D}^{set} = \text{query}(\mathcal{M}, \mathcal{D}^{rem});$
- 3: $\mathcal{D}_{tr} = \mathcal{D}^{ini} \cup \mathcal{D}^{set};$
- 4: $\mathcal{D}' = \text{sort}(\mathcal{D}_{tr}, \mathcal{C});$
- 5: $\{\mathcal{D}^1, \mathcal{D}^2, \dots, \mathcal{D}^m\} = \text{split}(\mathcal{D}');$
- 6: **for** $k=1 \dots m$ **do**
- 7: **While** not converged **for** p epochs **do**
- 8: train($\mathcal{M}, \mathcal{D}^k$); **end while**
- 9: **end for**

Table 4 Number of training data in the sequential ACL

		Unsafe	Safe	Sum
Class-balanced (\mathcal{D}^{ini})		24,000	20,838	44,838
\mathcal{D}^{set}	SIM	-	1,187	2,477
	FOT	-	1,290	
\mathcal{D}_{tr}		24,000	23,315	47,315

동 학습을 적용하였다. 다음으로 순차적으로 커리큘럼 학습을 적용한다. 즉 확장된 학습 데이터셋(\mathcal{D}_{tr})을 $\mathcal{C}(z)$ 에 따라 오름차순 정렬(Sort)하고, 이산형 스케줄러 Baby-step을 채택하여 난이도 기준으로 m 개의 구간으로 분할(Split)한다. 각 구간 \mathcal{D}^s 에 대해 학습을 진행하며, 각 단계에서는 p epoch 동안 모델이 수렴할 때까지 반복적으로 학습을 수행한다. 해당 학습 방법을 통해 사용된 전체 샘플 수는 Table 4와 같다.

4.2 반복형 능동-커리큘럼 학습

두번째 학습전략으로 커리큘럼 학습을 먼저 적용한 후 능동학습을 적용하는 방식의 반복형 능동-커리큘럼 학습 방법을 제안하고자 한다. Fig. 9는 반복형(Iterative) 능동-커리큘럼 학습 전략을 도식화한 것으로 기존 능동 학습에 커리큘럼 학습을 초기 학습 과정에 결합하여 확장된 구조를 보여준다. 즉 제안된 학습전략은 우선 초기 학습 데이터셋에 난이도 지표(Difficulty measure)와 학습 스케줄러(Training scheduler)를 적용하여 데이터 분할 및 학습 스케줄러에 기반한 커리큘럼 학습을 각각 수행한다. 다음으로, 능동 학습 방법을 적용하여 질의(Query) 과정을 통해 오판단 데이터를 선별하고 새로운 데이터셋으로 확장(Annotate & append)하는 절차를 반복한다. 좀 더 구

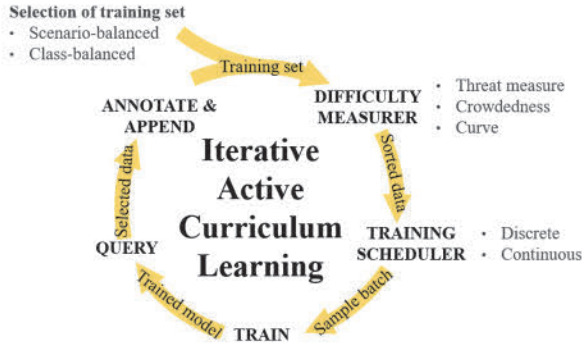


Fig. 9 Overview of active curriculum learning

Algorithm 2: Iterative active-curriculum learning	
Output : \mathcal{M}^* : optimal model	
1:	$\mathcal{D}_{tr}^1 = \mathcal{D}^{ini}$;
2:	for $j = 1 \dots n$ do
3:	$\mathcal{D}' = \text{sort}(\mathcal{D}_{tr}^j, \mathcal{C})$
4:	$\{\mathcal{D}^1, \mathcal{D}^2, \dots, \mathcal{D}^m\} = \text{split}(\mathcal{D}')$
5:	for $k = 1 \dots m$ do
6:	while not converged for p epochs do
7:	$\text{train}(\mathcal{M}, \mathcal{D}^k)$; end while
8:	end for
9:	$\mathcal{D}^{sel} = \text{query}(\mathcal{M}, \mathcal{D}^{rem})$;
10:	$\mathcal{D}_{tr}^{j+1} = \mathcal{D}_{tr}^j \cup \mathcal{D}^{sel}$;
11:	end for

체적으로 Baby-step 방식의 학습 스케줄러를 기반으로 한 능동-커리큘럼 학습과정을 Pseudo code 형태로 정리하면 Algorithm 2와 같다.

학습 데이터셋(\mathcal{D}_{tr})는 난이도 지표 $\mathcal{C}(\mathbf{z})$ 를 이용하여 오름차순으로 정렬(Sort)된 후 m 개의 난이도 구간으로 분할(Split)되며, 난이도가 낮은 데이터부터 순차적으로 학습에 포함된다. 각 학습 단계 s 에 이전까지 학습한 데이터와 현재 단계의 데이터를 누적하며 \mathcal{D}_{tr} 을 구성하고, 설정된 epoch동안 수렴할 때까지 반복 학습을 수행한다. 이후 모델은 query 단계를 통해 오판단 데이터를 탐색 후 선정(\mathcal{D}^{sel})하고, 이를 학습 데이터에 추가(\mathcal{D}_{tr})하여 다음 Iteration의 학습에 사용한다. 첫번째 Iteration에서는 가상 데이터로(\mathcal{D}^{ini})만 구성된 Table 3의 Class-balanced 데이터셋을 사용하지만, 이후 반복 단계부터는($j \geq 2$) 가상 및 실도로 주행 데이터를 모두 포함하는 \mathcal{D}^{rem} 에 대하여 알고리즘 평가가 이루어지고, 오판단 데이터에 대하여 query로 선별된 데이터를 확장시킨 데이터셋(\mathcal{D}_{tr})으로 학습을 진행하며, 전체 학습에 사용된 샘플 수는 Table 5에 제시하였다. 이와 같은 반복 구조를 통해 모델은 스스

Table 5 Number of training data in the iterative ACL

Iteration	Training data	Unsafe	Safe	Sum
$j = 1$	$\mathcal{D}_{tr}^1 = \mathcal{D}^{ini}$	24,000	20,838	44,838
$j = 2$	Virtual	-	612	851
	Real	-	239	
$j = 3$	\mathcal{D}_{tr}^2	24,000	21,689	45,689
	Virtual	-	1,249	
	Real	-	76	
$j = 4$	\mathcal{D}_{tr}^3	24,000	23,014	47,014
	Virtual	-	451	
	Real	-	5	
	\mathcal{D}_{tr}^4	24,000	23,470	

로 학습에 유효한 데이터를 선정하고, 난이도에 따라 학습 범위를 점진적으로 확장함으로써 안정적이고 효율적인 학습이 가능하다.

5. 실험 결과

5.1 실험 환경

Table 2에서 요약된 복합 데이터셋을 기반으로 모델을 학습한 뒤, 평가 데이터셋에 대하여 서로 다른 모델의 학습 방법론에 따른 성능을 비교하였다. 평가를 위한 데이터셋으로는 가상 및 실도로 주행 데이터에 대하여 각각 6,356개와 1,976개의 상세 시나리오를 사용하며, 실도로 주행 데이터는 20초 단위로 분할된 Snippet으로 구성된다.

커리큘럼 학습에서 사용된 이산형 스케줄러의 경우, 설정한 난이도 단계 내에서 정확도(Accuracy)가 90%로 수렴하면 다음 단계 학습을 진행하도록 하였다. 각 단계는 사전에 정의된 난이도 구간에 따라 데이터가 분할되어 있으며, 각 단계의 데이터셋이 모두 학습된 후에 다음 단계로 넘어가는 구조를 가진다. 반면 연속형 스케줄러의 경우 난이도 임계값을 점진적으로 증가시키는 방식으로 구성하였다. 학습 초기에는 총 데이터 중 1%만을 활용했으며, 이후 임계값은 설정된 T_{grow} 동안 식 (9)와 (10)에서 제시한 Linear, root 방식에 따라 일정한 속도로 증가하며 T_{grow} 이후 전체 데이터를 포함하도록 하였다. 임계값이 최대에 도달한 이후에는, 설정한 epoch 수만큼 전체 학습 데이터를 반복적으로 사용하여 모델을 학습시켰다.

5.2 비교 평가

학습전략에 따른 성능 비교를 위해 두가지 주요 평가 지표를 고려하였다. 즉, 실도로 주행 데이터셋의 성능 개선과 동시에 위험 상황을 포함한 가상 데이터에 대한 성능 저하를 최소화하는 것을 목표로 하고자 한다. 이를 위

하여 우선 Table 1에서 정의한 오판단(FP)에 해당하는 평가 지표로 FPR(False Positive Rate)과 발생 건수(# of FP)를 함께 선정하였다. 다음으로 가상 데이터에 대한 성능 향상 또는 저하를 판단하기 위해서 정확도(ACC)와 FNR(False Negative Rate)를 평가 지표로 선정하였다.

학습 전략의 비교 평가를 위해서 Table 6에서 보는 바와 같이 능동학습(AL), 순차적 및 반복적 능동 커리큘럼 학습으로 구분을 하였다. 특히 순차적 능동 커리큘럼 학습의 경우 많은 조합이 가능한데 그 중 대표적인 9개 조합만을 선정하여 결과를 비교하였다. 이를 위해 Table 2의 Test에 해당하는 데이터로 성능을 평가하였으며, 가상 데이터셋과 실도로 데이터셋에 대한 성능 결과를 Tables 6과 7에 각각 제시하였다. 이는 우선 가상 데이터셋을 기반으로 한 결과로 난이도 지표와 학습 스케줄러를 선정하고 이후에 실도로 주행 데이터셋에 대하여 FPR(False Positive Rate)과 FP 발생 건수(# of FP)를 모두 저감할 수 있는 학습전략을 선정하기 위함이다.

우선 Table 6의 결과를 기반으로 난이도 지표와 학습 스케줄러를 선정하는 과정을 설명하고자 한다. 난이도 지표의 경우 3.3.1절에서 제안된 C_{TTC} , C_{CP} , C_{crowd} 와 C_{curve} 가 모두 고려되었다. 3.3.2절에서 설명된 Baby-step(B-S) 스케줄러의 경우 선행 연구²⁾를 참고하여 데이터를 Easy와 Hard와 같이 2단계(*i.e.*, $m=2$)로 나누는 경우(Case)를 포함시켰고, 이를 점진적으로 증가시켜 10단계($m=10$)까지 고려하였다. 그 외에 연속형 스케줄러로 선형(Linear) 및 제곱근(Root) 함수 형태로 학습 범위를 점진적으로 확장하는 방식이 고려되었다.

다음으로 능동학습(Case 1)만 적용된 경우 네 가지 평가 지표에 대한 결과를 모두 보여주고 있으며 이후 난이도 지표와 학습 스케줄러의 다양한 조합 중 해당 성능보다

낮은 경우는 후보에서 제외되었다. 이러한 과정을 통하여 선택된 대표적인 9가지의 조합을 순차적 능동 커리큘럼 학습에 적용된 결과를 Case 2부터 Case 10까지 보여주고 있다. 특히 난이도 지표를 C_{CP} 로 설정하고 스케줄러를 Baby-step의 10단계로 분할한 Case 5의 경우 ACC, FPR, FP의 개수와 같은 성능 관점에서 우수함을 보여주고 있다. 단, FNR 관점에서 Case 3에 비해 0.1%가 부족하지만 이는 FN의 개수 관점에서 4개 정도에 해당하며 통계적으로 성능 저하 편차(Deviation)에 해당한다고 볼 수 있다. 이러한 결과 분석을 바탕으로 선정된 조합을 반복형 능동 커리큘럼 학습에 적용한 결과를 Case 11에서 보여주고 있다. Case 1와의 성능 비교를 통해 FNR 6.6% 증가 외에 다른 평가 지표에서 모두 성능이 개선되었음을 보여주고 있다.

실도로 주행 데이터셋의 경우 FNR에 해당하는 데이터를 포함하고 있지 않기에, 평가지표에서 FNR을 제외하고 Table 6과 동일한 조합에 대하여 Table 7을 구성하였다. Table 7의 Case 5와 Case 1을 비교하였을 때 Accuracy와 FPR에서 각 0.3%, 16.7% 개선된 결과를 보여주었다. 해당 조합을 Iterative ACL에 적용한 결과는 Case 11에서 확인 가능하며, Accuracy와 FPR에서 각 0.7%, 58.3% 개선된 결과를 보여주었다. 더불어 Measure를 C_{CP} 로, 스케줄러를 Linear로 설정한 Case 7 조합의 경우 가상 데이터셋에서도 성능 향상을 이루었으며, 실도로 주행 데이터에 대해서 Accuracy와 FPR에서 각 0.4%, 33.3%의 향상이 이루어졌다. 따라서 이를 Iterative ACL에 적용해보았고, 결과는 Case 12와 같다. 이 경우 Accuracy와 FPR에서 각 0.9%, 75% 개선된 결과를 보여주었다. 다만 Case 12의 경우 가상 데이터셋 평가 시 FNR이 33.3% 증가하는 경향을 보여 실도로 주행 데이터에 한하여 큰 성능 향상을 보임을 확인하였다.

Table 6 Performance comparison of AL, sequential and iterative ACL with respect to virtual dataset

Case	Method	Measure	Scheduler (m)	ACC↑ (%)	FNR↓ (%)	FPR↓ (%)	# of FP↓
1	AL	-	-	95.2	1.5	6.8	259
2	Sequential ACL	C_{crowd}	B-S (10)	95.0	1.5	7.3	275
3		C_{curve}	B-S (10)	94.9	1.0	7.8	295
4		C_{TTC}	B-S (10)	95.2	1.4	7.0	266
5		C_{CP}	B-S (10)	96.0	1.1	5.7	217
6		C_{CP}	B-S (2)	95.3	1.4	6.8	258
7		C_{CP}	Linear	95.4	1.3	6.8	257
8		C_{TTC}	Linear	95.3	1.4	6.6	249
9		C_{curve}	Linear	95.2	1.8	6.8	259
10		C_{CP}	Root	95.6	1.5	6.2	236
11	Iterative ACL	C_{CP}	B-S (10)	96.2	1.6	5.1	194
12		C_{CP}	Linear	95.9	2.0	5.3	202

Table 7 Performance comparison of AL, sequential and iterative ACL with respect to real-driving datasets

Case	Method	Measure	Scheduler (<i>m</i>)	ACC ↑ (%)	FPR ↓ (%)	# of FP↓
1	AL	-	-	98.7	1.2	22
2	Seq. ACL	C_{crowd}	B-S (10)	98.7	1.2	23
3		C_{curve}	B-S (10)	99.1	0.9	17
4		C_{TTC}	B-S (10)	99.0	1.0	18
5		C_{CP}	B-S (10)	99.0	1.0	19
6		C_{CP}	B-S (2)	98.9	1.1	20
7		C_{CP}	Linear	99.1	0.8	15
8		C_{TTC}	Linear	98.6	1.2	25
9		C_{curve}	Linear	98.7	1.2	22
10		C_{CP}	Root	98.9	1.1	20
11		Iter. ACL	C_{CP}	B-S (10)	99.4	0.5
12	C_{CP}		Linear	99.6	0.3	8

이와 같은 결과를 바탕으로 실도로 주행데이터를 기반으로 오판단(False alarm) 비율을 줄이기 위한 학습전략이 제안되었고, Measure를 C_{CP} 로, 스케줄러를 Baby-step의 10단계로 설정하였을 경우 기존 가상 데이터셋에 대한 성능 저하가 거의 없이 실도로 주행 데이터에 대한 FPR 및 FP 발생횟수를 줄일 수 있음을 확인하였다. 더 나아가, 반복형 ACL은 학습 결과를 기반으로 난이도 조정과 데이터 선택을 반복적으로 수행함으로써 학습 전략을 점진적으로 정교화할 수 있다는 특징을 가진다. 이는 실제 차량 적용 환경에서 충돌 예측 시스템의 신뢰성 확보에 중요한 요소를 충족하였음을 의미한다.

마지막으로 Fig. 10에서는 Fig. 3에서 보여주었던 능동 학습의 한계점을 개선하여 반복을 통하여 정확도 성능을 더 높일 수 있음을 보여주고 있다. 비교를 위해서 3가지 학습전략이 선택되었다. 능동학습(AL)의 경우 Fig. 3의 Dataset 1에 대한 학습 성능을 선정하였으며, 순차형(Sequential) ACL의 경우 C_{CP} , Baby-step 스케줄러를 적용한(Tables 6, 7의 Case 5 해당) 학습 전략을 선정하였다. 반복형 ACL의 경우 Case 11에 해당하는 난이도 지표와 스케줄러를 선정하였다. 첫 번째 학습(Iteration = 1) 후에는 커리큘럼 학습이 적용되어 있는 반복형 ACL이 가장 높은 정확도를 보여주고 있다. 두 번째 학습(Iteration = 2) 후에는 모두 비슷한 성능을 보여주고 있다. 세 번째 및 네 번째 학습(Iteration = 3, 4) 후에는 반복형 ACL 전략이 정확성 관점에서 꾸준히 상승하고 있음을 보여주고 있다. 즉, 실도로 주행 환경에서의 신뢰성 향상과 위험 상황 판단 성능의 균형 유지라는 두 관점에서 제안하는 기법이 우수한 성능을 보였음을 확인할 수 있다.

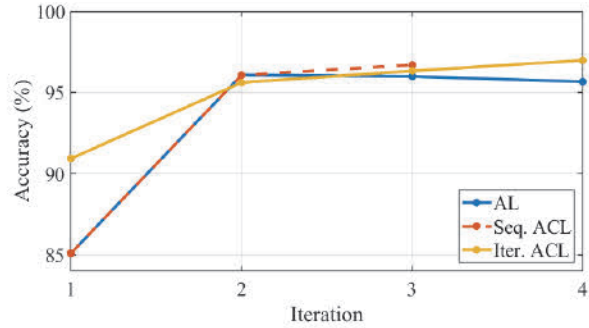


Fig. 10 Accuracy per iteration: AL vs. Seq. ACL vs. Iter. ACL

6. 결론

본 연구에서는 복합 데이터를 활용하여 차량 간 충돌 예측 알고리즘의 성능을 향상시키기 위하여 데이터 균형을 고려한 능동-커리큘럼 학습 전략을 제안하였다. 가상 위험 시나리오를 포함하는 복합 데이터셋을 초기 데이터셋과 후반 데이터셋으로 구성을 하고 초기 데이터셋에 시나리오와 클래스 관점에서 데이터 균형을 고려한 학습 데이터를 선정한다. 다음으로 기존 능동 학습 커리큘럼 학습을 결합한 순차형 및 반복형 능동-커리큘럼 학습 전략을 제안하여 노이즈에 의한 오판단(False alarm) 문제를 완화하고 학습 안정성을 높일 수 있음을 보여주었다. 구체적으로, 제안하는 기법인 반복형 능동 커리큘럼 학습(Iterative active-curriculum learning)이 기존 선행연구인 능동학습(Active learning) 대비 실도로 주행 데이터의 경우 Accuracy와 FPR에서 각 0.7%, 58.3% 개선된 결과를 보여주었다. 더불어 충돌을 포함하는 가상 데이터의 경우 Accuracy와 FPR에서 각 1.05%, 25% 개선되었음을 보여주었다.

추후 연구로는 제안하는 학습 방식을 충돌 예측뿐만 아니라 궤적 예측, 충돌 회피 등 다양한 목적의 네트워크에 대한 학습 전략으로 확대하고자 한다. 또한 생성형 능동학습과 같은 데이터 증강을 통한 알고리즘 성능 제고를 고려하고 있다.

후 기

본 연구는 국토교통부 국토교통 DNA플러스 융합기술대학원 육성사업의 연구비 지원(과제번호: RS-2022-00156089)에 의해 수행되었습니다.

References

- 1) MITRE, A Study on Real-World Effectiveness of Model Year 2015–2023 Advanced Driver Assistance Systems, <https://www.mitre.org/sites/default/files/2025-01/PR-25->

- 0114-Study-Real-world-Effectiveness-Model-year-2015-%E2%80%932023-ADAS.pdf, 2025.
- 2) National Highway Traffic Safety Administration (NHTSA), Final Rule - Automatic Emergency Braking Systems for Light Vehicles, https://www.nhtsa.gov/sites/nhtsa.gov/files/2024-04/final-rule-automatic-emergency-braking-systems-light-vehicles_web-version.pdf, 2025.
 - 3) Y. Gong, G. Liu, Y. Xue, R. Li, and L. Meng, "A Survey on Dataset Quality in Machine Learning," *Information and Software Technology*, Vol.162, 107268, 2023.
 - 4) K. Lee and D. Kum, "Collision Avoidance/Mitigation System: Motion Planning of Autonomous Vehicle via Predictive Occupancy Map," *IEEE Access*, Vol.7, pp. 52846–52857, 2019.
 - 5) J. Jansson, Collision Avoidance Theory with Application to Automotive Collision Mitigation, Ph.D. Dissertation, Linköping University, Linköping, 2005.
 - 6) D. Lee and H. Yeo, "Real-Time Rear-End Collision-Warning System Using a Multilayer Perceptron Neural Network," *IEEE Transactions on Intelligent Transportation Systems*, Vol.17, No.11, pp.3087–3097, 2016.
 - 7) X. Wang, J. Liu, T. Qiu, C. Mu, C. Chen, and P. Zhou, "A Real-Time Collision Prediction Mechanism with Deep Learning for Intelligent Transportation System," *IEEE Transactions on Vehicular Technology*, Vol.69, No.9, pp.9497–9508, 2020.
 - 8) S. Lee, B. Song, and J. Shin, "Collision Prediction in an Integrated Framework of Scenario-Based and Data-Driven Approaches," *IEEE Access*, Vol.12, pp.55234–55247, 2024.
 - 9) S. Lee, Y. Jeong, and B. Song, "Multi-Task Prediction of Collision and Trajectories Based on Transformer Network for Safety-Critical Scenarios of Automated Vehicles," *Transactions of KSAE*, Vol.32, No.10, pp.843–852, 2024.
 - 10) Z. Wang, S. Lan, and X. Sun, "Enhancing Autonomous Driving Safety with Collision Scenario Integration," *arXiv preprint*, arXiv:2503.03957, 2025.
 - 11) X. Zhang, Q. Zhang, L. Han, Q. Qu, and X. Chen, "AccidentSim: Generating Physically Realistic Vehicle Collision Videos from Real-World Accident Reports," *arXiv preprint*, arXiv:2503.20654, 2025.
 - 12) W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, "A Survey on Imbalanced Learning: Latest Research, Applications and Future Directions," *Artificial Intelligence Review*, Vol.57, No.6, pp.137, 2024.
 - 13) M. Buda, A. Maki, and M. A. Mazurowski, "A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks," *Neural Networks*, Vol.106, pp.249–259, 2018.
 - 14) L. Li and X. Zhang, "Addressing Data Imbalance in Collision Risk Prediction with Active Generative Oversampling," *Scientific Reports*, Vol.15, No.1, Paper No.9133, 2025.
 - 15) International Organization for Standardization, ISO/PAS 8800:2024—Road Vehicles—Safety and Artificial Intelligence, ISO, Geneva, 2024.
 - 16) E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecky, H. Xu, D. Roy, and J. M. Alvarez, "Scalable Active Learning for Object Detection," *IEEE Intelligent Vehicles Symposium (IV)*, pp.1430–1435, 2020.
 - 17) P. Ren, Y. Xiao, X. Chang, P. Y. Chang, Z. Li, B. B. Gupta, and X. Wang, "A Survey of Deep Active Learning," *ACM Computing Surveys (CSUR)*, Vol.54, No.9, pp.1–40, 2021.
 - 18) X. Wang, Y. Chen, and W. Zhu, "A Survey on Curriculum Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.44, No.9, pp.4555–4576, 2021.
 - 19) M. Mottaghi and K. Baek, "Deep Active Learning in the Presence of Label Noise: A Survey," *arXiv preprint*, arXiv:2302.11075, 2023.
 - 20) V. Menden, Y. Saleh, and A. Iske, "Bounds on the Generalization Error in Active Learning," *arXiv preprint*, arXiv:2409.09078, 2024.
 - 21) Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, Vol.382, pp.41–48, 2009.
 - 22) B. Jafarpour, D. Sepehr, and N. Pogrebnayakov, "Active Curriculum Learning," *First Workshop on Interactive Learning for Natural Language Processing*, pp.40–45, 2021.
 - 23) S. Ma, H. Du, K. M. Curran, A. Lawlor, and R. Dong, "Adaptive Curriculum Query Strategy for Active Learning in Medical Image Classification," *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp.48–57, 2024.
 - 24) D. Narzary and K. C. Veluvolu, "Multiple Sensor Fault Detection Using Index-Based Method," *Sensors*, Vol.22, No.20, 2022.
 - 25) KoROAD, Traffic Accident Analysis System, <http://taas.koroad.or.kr>
 - 26) 2016.IGLAD, Initiative for the Global Harmonisation of Accident Data, <https://www.iglad.org/>, 2025.
 - 27) J. Lee, U. I. Jung, and B. Song, "Critical Scenario Generation for Collision Avoidance of Automated

- Vehicles Based on Traffic Accident Analysis and Machine Learning,” Transactions of KSAE, Vol.28, No.11, pp.817–826, 2020.
- 28) A. Sadat, S. Segal, S. Casas, Tu. J. Yang, B. Urtasun, and E. Yumer, “Diverse Complexity Measures for Dataset Curation in Self-Driving,” 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.8609–8616, 2021.
- 29) V. Cirik, E. Hovy, and L. Morency, “Visualizing and Understanding Curriculum Learning for Long Short-Term Memory Networks,” arXiv preprint, arXiv:1611.06204, 2016.
- 30) E. A. Platanios, O. Stretcu, G. Neubig, B. Póczos, and T. M. Mitchell, “Competence-Based Curriculum Learning for Neural Machine Translation,” arXiv preprint, arXiv:1903.09848, 2017.