2025년 KSAE 전기전자시스템부문 하계 교육 프로그램

자율주행 연구자를 위한 하계 교육 워크샵

nuPlan 활용 E2E 자율주행 실습

문의 메일

hyunwookkang@hanyang.ac.kr

youngkikim@hanyang.ac.kr

seungjiryu@hanyang.ac.kr

khseok@hanyang.ac.kr

2025년 7월 11일

한양대학교 미래자동차공학과

Autonomous Intelligence (AI) Lab.

KSAE 한국자동차공학회 The Korean Society of Automotive Engineers

HANYANG UNIVERSITY AI LAB Automotive Intelligence

목차

nuPlan 환경 분석

nuPlan 프레임워크 설명

nuPlan 데이터 구조 분석

nuPlan-devkit 설명

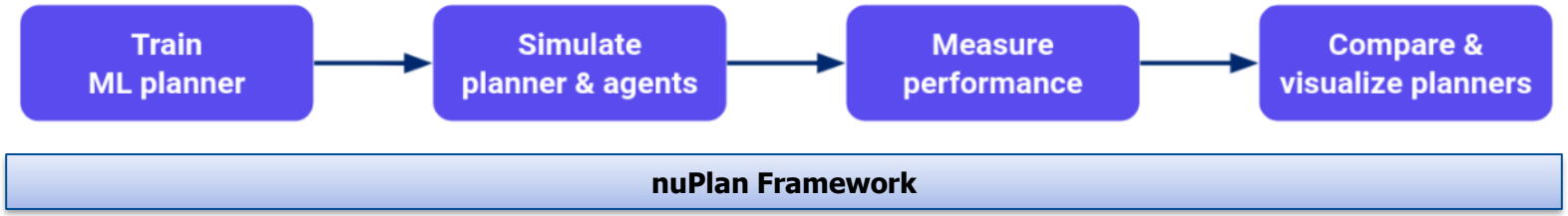
Train and Validation Framework: nuPlan

■ nuPlan 개요

- ▶ 실제 자율주행 차량 로그에서 추출한 **고정밀 시나리오와 지도 데이터 제공**
- ▶ **Planning 모듈 평가에 중점**을 둔 설계
- ▶ **경량화된 closed-loop 시뮬레이터** 제공
- ▶ 결과 시각화를 위한 **인터랙티브 도구** 제공
- ▶ **정량적 및 정성적 모델 비교**를 단일 프레임워크 내에서 지원

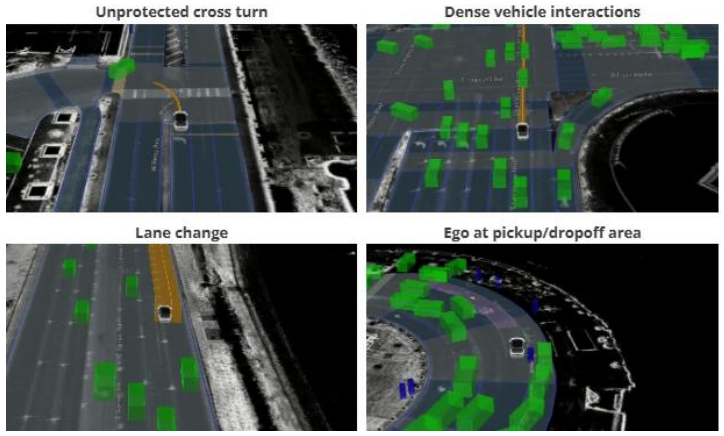
■ nuPlan Framwork Overview

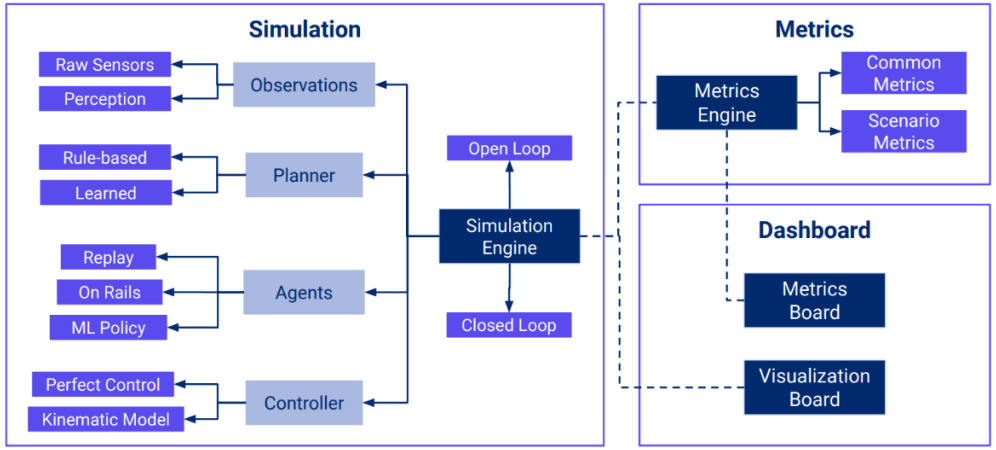
- ▶ Train (모델 학습) → **PyTorch** 기반의 ML 플래너 학습
- ▶ Simulate (시뮬레이션 평가) → **시나리오 기반 시뮬레이터**를 통한 정책 평가
- ▶ Measure (성능 측정) → 다양한 **평가 지표**를 활용한 정량적 평가
- ▶ Visualize (결과 시각화) → **nuBoard**를 통한 시각적 결과 분석

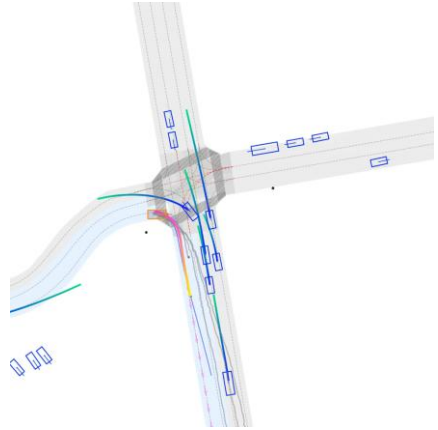


nuPlan Framework

- **1. Train – 모델 학습**
  - ▶ 입력 데이터 : 자차(Ego) 및 주변 차량 정보, 지도 데이터, (선택적으로 센서 데이터 포함)
    - PyTorch 기반의 학습 프레임워크를 통해 다양한 플래너 아키텍처에 대해 **손쉬운 확장성 제공**
- **2. Simulate – 시뮬레이션 평가**
  - ▶ 완전한 **모듈화된 아키텍처**로 각 구성 요소를 자유롭게 교체하거나 확장 가능
    - 주요 구성요소: 데이터셋, 관측 모델, 자차 플래너, 에이전트 모델, 모션 컨트롤러, 평가 지표, 시각화 도구
- **3. Measure – 성능 평가**
  - ▶ **다섯 가지 평가 범주**를 기반으로 정량적인 성능 평가 제공:
    - Traffic rule violation, Human driving similarity, Vehicle dynamics, Goal achievement, Scenario-specific metric
- **4. Visualize – 결과 시각화**
  - ▶ HD 지도, 자차, 주변 객체, 신호등 상태 등을 시각화







nuPlan Simulation Evaluation Method

1. Open-loop 모드

- 모든 차량이 **기록된 로그 그대로** 주행
- 모델은 단지 **예측된 경로(trajectory)만 출력**하며, 실제 행동은 시뮬레이션되지 않음.
- **모델의 행동이 시나리오에 영향을 주지 않기 때문에**, 실제 상호작용을 반영하기 어려움

2. Closed-loop Non-reactive 모드

- 자차(Ego 차량)는 모델의 예측에 따라 주행
- 주변 차량은 여전히 **기록된 경로(logged trajectory)** 를 따라감 → **자차 정책이 시나리오에 미치는 영향**을 관찰 가능
- 하지만 **주변 차량은 자차의 움직임에 반응하지 않음**

3. Closed-loop Reactive 모드

- 자차와 주변 차량이 **서로 상호작용함.**
- Ego 차량의 행동이 주변 차량에 영향을 주고, 그 반응이 다시 Ego 차량에 영향을 미침.
- **현실 세계 주행과 가장 유사하며**, 상호작용 및 안전성 테스트에 적합
- 단, 주변 차량은 보통 **룰 기반(rule-based)** 으로 작동

참고 자료: https://nuplan-devkit.readthedocs.io/_/downloads/en/latest/pdf/

nuPlan 환경 분석

nuPlan 프레임워크 설명

nuPlan 데이터 구조 분석

nuPlan-devkit 설명

nuPlan 데이터 구성 요소

■ nuPlan 데이터는 자율주행 차량의 센서 및 주행 기록 포괄

- ▶ **자차 주행 정보**
  - 자차의 위치 (자차 좌표계 pose), 속도 등의 운동학 정보
- ▶ **객체 트랙 (주변 차량 / 보행자 등)**
  - 주변 객체들을 후처리 추적하여 프레임마다 ID가 부여된 바운딩 박스와 클래스(차량, 보행자 등) 정보를 제공
- ▶ **신호 정보**
  - 각 프레임의 교통 신호등 상태 (빨강/초록 등) 데이터를 포함하여, 오프라인 추론된 신호 상태를 제공, 이는 실제 시뮬레이션에서 신호등 처리의 사실성을 높이는 중요한 요소
- ▶ **지도(Map) 정보**
  - HD맵 (고정밀 지도) 데이터가 지역별 .gpkg 포맷으로 제공하며, 지도에는 차선, 정지선, 교차로 등 자율주행에 필요한 지형지물 정보 포함
- ▶ **자세한 데이터 정보**: https://github.com/motional/nuplan-devkit/blob/master/docs/nuplan_schema.md

■ 시나리오 및 태그

- ▶ nuPlan은 **주행 로그를 개별 시나리오로 간주**하며, 각 시나리오에 태그를 달아 분류
- ▶ 흔한 주행 상황뿐만 아니라 다양한 이벤트 (급정거, 끼어들기, 좌회전 등)를 태그로 분류하여 흔한 상황부터 희귀 상황까지 **시나리오 형태로 정리**
- ▶ 특정 주행 상황에서 플래너를 평가하고 분석하는데 유용
- ▶ .db 파일로 관리

nuPlan Training & Validation Scenarios

'accelerating_at_crosswalk'
'accelerating_at_stop_sign'
'accelerating_at_stop_sign_no_crosswalk'
'accelerating_at_traffic_light'
'accelerating_at_traffic_light_with_lead'
'accelerating_at_traffic_light_without_lead'
'behind_bike'
'behind_long_vehicle'
'behind_pedestrian_on_driveable'
'behind_pedestrian_on_pickup_dropoff'
'changing_lane'
'changing_lane_to_left'
'changing_lane_to_right'
'changing_lane_with_lead'
'changing_lane_with_trail'
'crossed_by_bike'
'crossed_by_vehicle'
'following_lane_with_lead'
'following_lane_with_slow_lead'
'following_lane_without_lead'
'high_lateral_acceleration'
'high_magnitude_jerk'
'high_magnitude_speed'
'low_magnitude_speed'
'medium_magnitude_speed'
'near_barrier_on_driveable'
'near_construction_zone_sign'
'near_high_speed_vehicle'
'near_long_vehicle'
'near_multiple_bikes'
'near_multiple_pedestrians'
'near_multiple_vehicles'
'near_pedestrian_at_pickup_dropoff'
'near_pedestrian_on_crosswalk'
'near_pedestrian_on_crosswalk_with_ego'
'near_trafficcone_on_driveable'
'on_all_way_stop_intersection'
'on_carpark'
'on_intersection'
'on_pickup_dropoff'
'on_stopline_crosswalk'
'on_stopline_stop_sign'
'on_stopline_traffic_light'
'on_traffic_light_intersection'
'starting_high_speed_turn'
'starting_left_turn'
'starting_low_speed_turn'
'starting_protected_cross_turn'
'starting_protected_noncross_turn'
'starting_right_turn'
'starting_straight_stop_sign_intersection_traversal'
'starting_straight_traffic_light_intersection_traversal':
'starting_u_turn'
'starting_unprotected_cross_turn'
'starting_unprotected_noncross_turn'
'stationary'
'stationary_at_crosswalk'
'stationary_at_traffic_light_with_lead'
'stationary_at_traffic_light_without_lead'
'stationary_in_traffic'
'stopping_at_crosswalk'
'stopping_at_stop_sign_no_crosswalk'
'stopping_at_stop_sign_with_lead'
'stopping_at_stop_sign_without_lead'
'stopping_at_traffic_light_with_lead'
'stopping_at_traffic_light_without_lead'
'stopping_with_lead'
'traversing_crosswalk'
'traversing_intersection'
'traversing_narrow_lane'
'traversing_pickup_dropoff'
'traversing_traffic_light_intersection'
'waiting_for_pedestrian_to_cross'

nuPlan 환경 분석

nuPlan 프레임워크 설명

nuPlan 데이터 구조 분석

nuPlan-devkit 설명

nuPlan-devkit 개요

■ **자율주행 차량 플래닝을 위한 종합 개발 키트로, 디렉토리 구조가 각 모듈별로 체계적으로 구성**

▶ nuPlan 데이터셋을 로드, 처리하고 시뮬레이션 및 학습을 수행하는 소프트웨어 프레임워크

■ **nuPlan-devkit 경로: https://github.com/motional/nuplan-devkit**

■ **nuPlan-devkit 폴더 구조**

```
nuplan_devkit/                  <--- 레포지토리 루트
├── ci/                         (CI 설정 관련 코드들 – 일반 사용자에겐 불필요)
├── docs/                       (리포지토리와 데이터셋 관련 문서 및 README들)
├── nuplan/                     (※ 메인 소스 코드 폴더)
│   ├── common/                 (공통 유틸리티 모듈: database와 planning에서 공유되는 코드)
│   ├── database/               (데이터셋 로드 및 맵 렌더링 등의 코어 devkit 기능)
│   ├── planning/               (플래닝 프레임워크: 시뮬레이션, 학습(training), 평가 관련 코드)
│   ├── submission/             (플래닝 챌린지용 제출 엔진 코드)
│   └── cli/                    (커맨드라인 인터페이스 도구: 예를 들어 데이터베이스 확인용)
└── tutorials/                  (인터랙티브 튜토리얼 노트북 모음)
```

주요 디렉토리 설명: nuPlan-devkit/nuplan

nuplan/common

- Common 모듈은 데이터베이스와 플래닝에서 공통으로 활용되는 **유틸리티 코드**

nuplan/database

- Database 모듈은 nuPlan **데이터셋 로드와 지도 렌더링**을 담당하는 핵심 모듈
- nuPlan의 방대한 주행 로그와 맵 데이터를 불러오고 쿼리하는 API, 시나리오 추출 기능 제공

nuplan/planning

- Planning 모듈은 자율주행 플래닝 프레임워크의 핵심으로, 시뮬레이션 엔진, 모델 학습(training), 플래너 평가(evaluation) 로직을 모두 포함한 standalone 플래닝 엔진
- **자율주행 차량의 경로 계획 알고리즘**, **시뮬레이션 실행**, **평가 지표 계산** 등을 구현

nuplan/submission

- Submission 모듈은 **nuPlan 챌린지 참가자를 위한 제출 엔진**, 평가 서버에 제출할 패키지를 구성하고 검증

nuplan/cli

- Cli 모듈은 nuPlan 데이터베이스와 상호작용하는 커맨드라인 도구 제공
- **데이터베이스 확인**이나 **변환 작업** 수행

주요 디렉토리 설명: nuPlan-devkit/tutorials

- **폴더 내에 여러 가지 주피터 노트북 예제가 제공되어 있어, 초기 학습과 실습에 활용 가능**

- **nuplan_framework.ipynb**
  - ▶ **nuPlan 프레임워크 종합 튜토리얼**로, 머신러닝 기반 플래너의 학습부터 시뮬레이션, 성능 측정, 결과 시각화까지 전체 과정 실습
    - 간단한 모델 학습 → 시뮬레이터 실행 후 metric 확인 → nuboard를 통한 시각화

- **nuplan_scenario_visualization.ipynb**
  - ▶ **시나리오 시각화 튜토리얼**로, nuPlan 데이터셋에 포함된 다양한 시나리오 유형을 필터링하여 불러오고, 이를 지도 위에 그려보는 예제 제공

- **nuplan_planner_tutorial.ipynb**
  - ▶ **플래너 개발 튜토리얼**로, 플래너를 구현하여 nuPlan 시뮬레이션에서 실행하는 방법 안내

- **nuplan_advanced_model_training.ipynb**
  - ▶ **고급 모델 학습 튜토리얼**로, nuPlan 아키텍쳐 상 심화된 내용과 확장 포인트를 안내
    - 데이터 캐싱 메커니즘, 벡터화 입력 생성, 커스텀 모델 파이프라인 통합 등을 제공을 통한 연구 지향적인 실습 가능

[실습] nuplan_scenario_visualization.ipynb

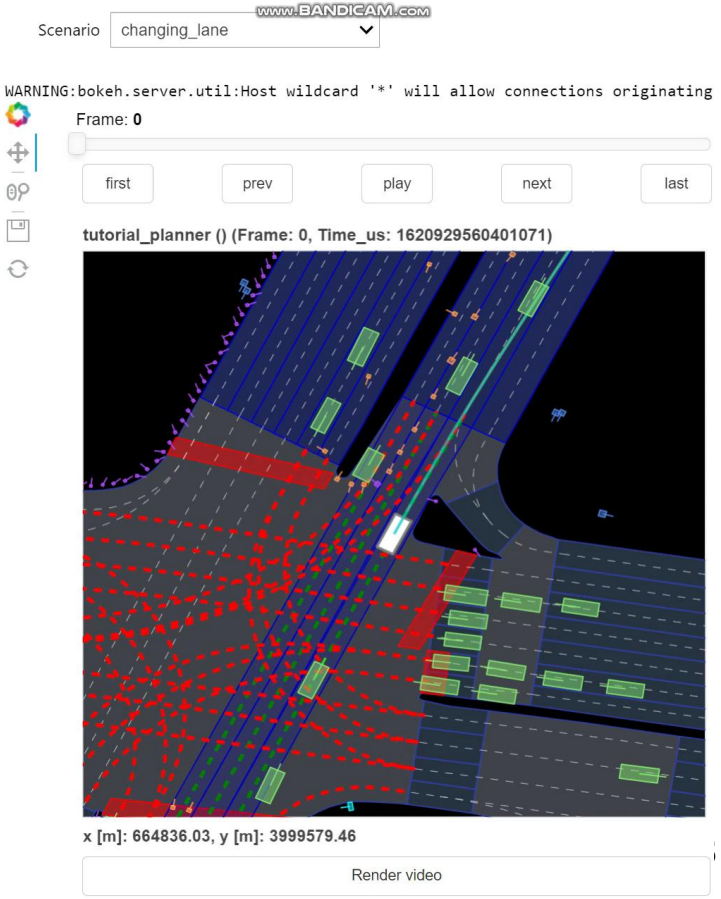
■ 목적: 시나리오 시각화 튜토리얼

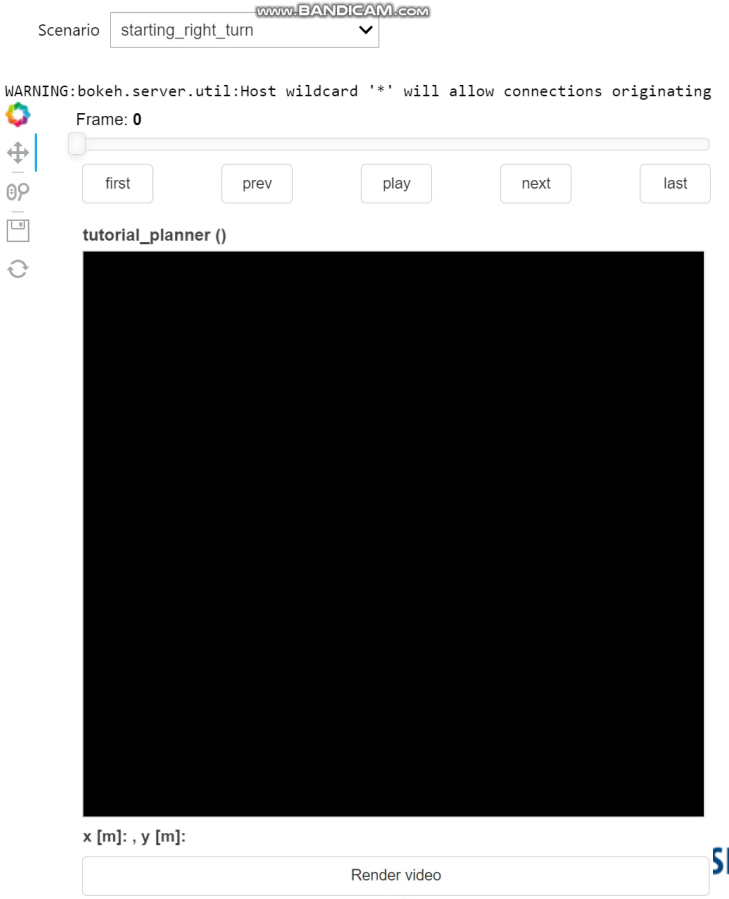
▶ nuPlan 데이터셋에 포함된 다양한 시나리오 유형을 필터링하여 불러오고, 이를 지도 위에 그려보는 예제 제공

■ 장점

▶ 각 시나리오 별 주변 환경 확인 가능

■ 실행 결과





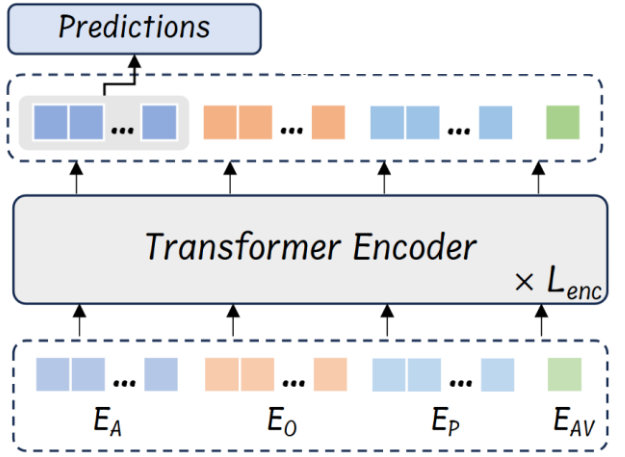
PLUTO 실습

PLUTO

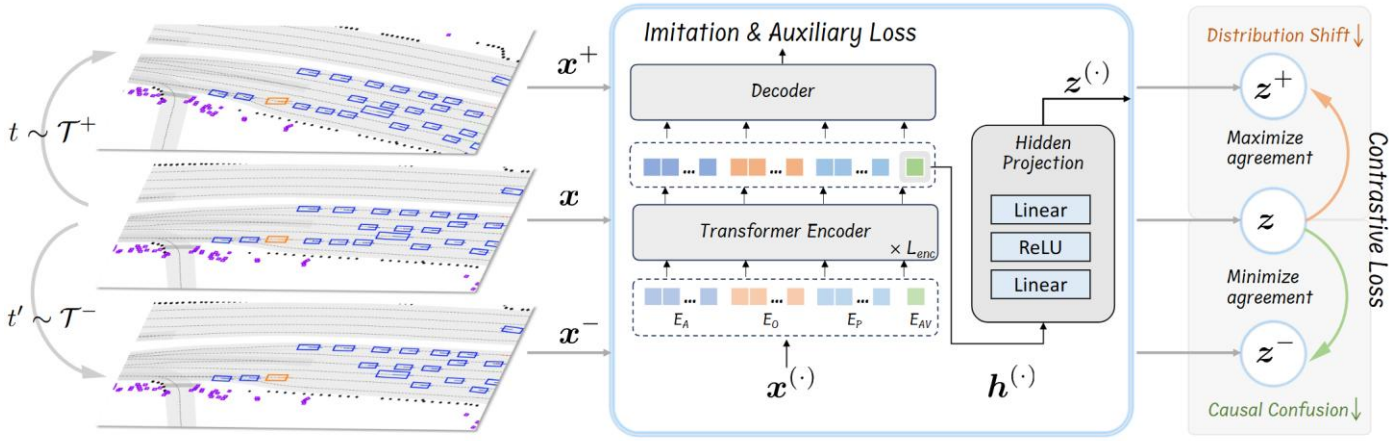
Title	PLUTO: Pushing the Limit of Imitation Learning-based Planning for Autonomous Driving		
Published	CoRR, 2024		
Author	*Jie Cheng, Yingbing Chen, Qifeng Chen*	Institution	*Hong Kong University of Science and Technology*

Contribution

- 종횡방향 거동을 유연하게 다루기 위해 **Longitudinal-lateral aware model architecture** 도입
- 주행 행동을 올바르게 제어하고 상호작용 학습을 강화하기 위한 **Contrastive Imitation Learning (CIL)** 적용
  - 모방학습의 대표적인 한계인 Causal confusion을 해결하기 위함
- Differential interpolation을 기반으로 충돌 방지, 도로 이탈 방지를 위한 **Auxiliary loss** 사용
- PlanTF 성능을 넘어 nuPaln closed-loop planning에서 SOTA 달성



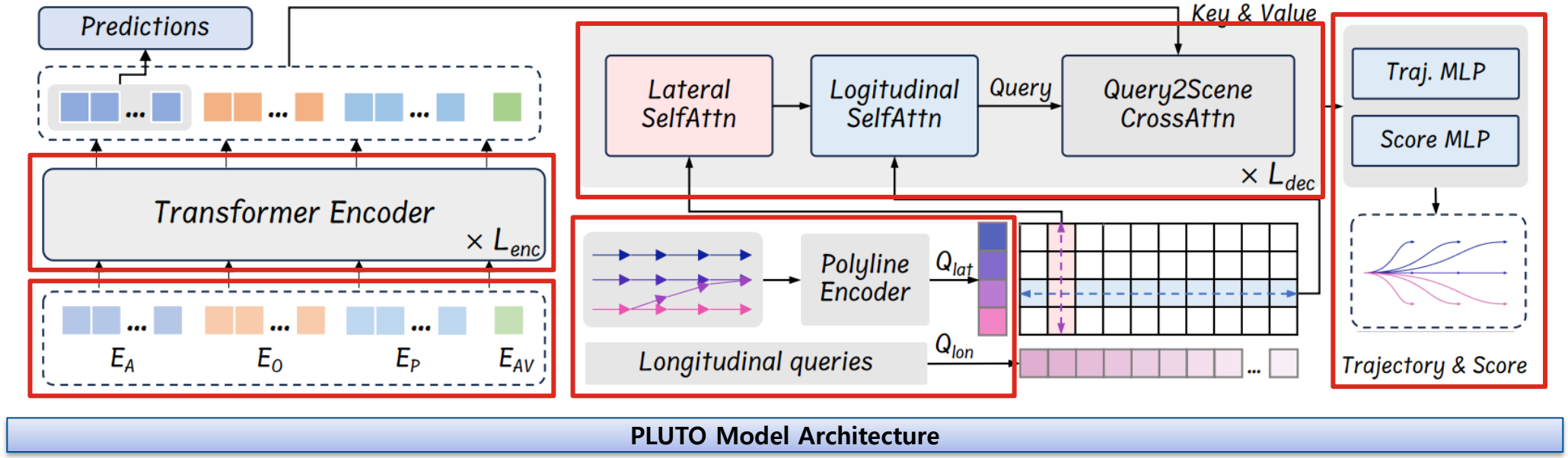
PLUTO Encoder



PLUTO Contrastive Imitation Learning

PLUTO Model Overview

- **Input vectors → Transformer encoder → Anchor-based query → Transformer decoder → Multi-modal prediction + planning**
  - **Input vectors**: 자차(현재 시점), 주변의 동적/정적 객체(과거 및 현재 시점), 그리고 차선 피처 정보를 벡터화 및 임베딩 적용
  - **Transformer encoder**: 자차, 객체, 차선 feature에 어텐션(attention)을 적용하여 그 상호작용을 학습
  - **Anchor-based query**: reference line을 반영한 횡방향 쿼리와 학습 가능한 종방향 쿼리 생성
  - **Transformer decoder**: 인코딩된 장면 피처(scene features)와 생성된 쿼리 간에 어텐션을 적용하여 상호작용을 학습



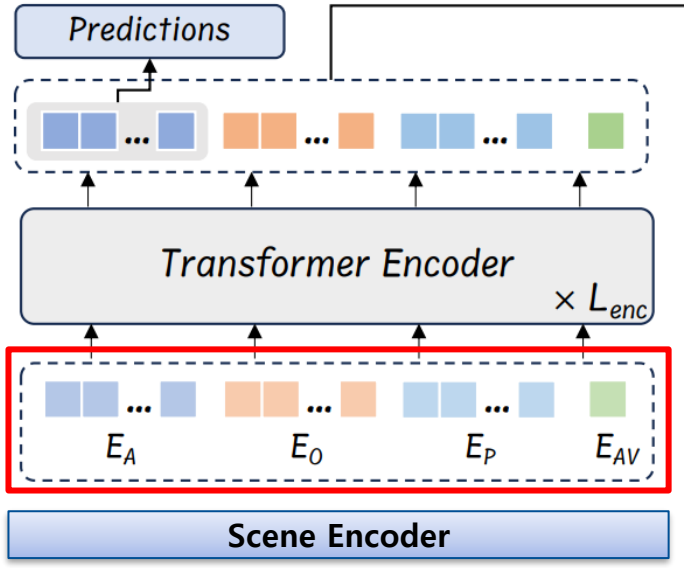
PLUTO Model Architecture

PLUTO | Input Representation

- **목적: 도심 환경에서의 자율주행을 위한 자차 다중 모드 궤적과 주변 에이전트의 움직임 예측**
- **모델 입력**
  - $\mathcal{A}$: 자율주행 차량의 feature (위치, 속도, 가속도, 조향각)와 주변 에이전트 feature
  - $\mathcal{O}$: 정적 장애물 feature
  - M: HD map feature
  - C: other traffic-related contexts (에이전트 타입, 제한 속도, global route, 신호등)

- **최종 Feature 표현**
  - PE: Positional Embedding
  - E_{attr} : 에이전트 타입, 제한 속도, global route 정보, 신호등 정보 등을 포함한 learnable embedding

$$E_0 = \texttt{concat}(E_{AV}, E_A, E_O, E_P) + PE + E_{attr}$$

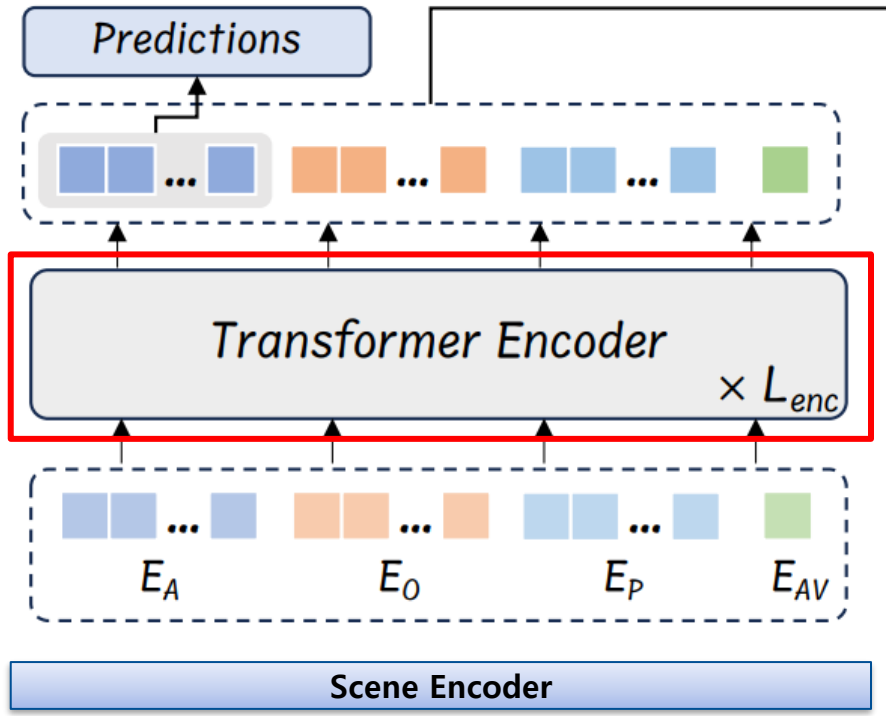


PLUTO | Transformer Encoder

Scene encoding

- Multi-head attention을 통해 토큰 간의 상호작용 학습
- $L_{enc}(=4)$개의 Transformer layer로 구성
- MHA: Standard multi-head attention
- FFN: Feedforward network layer

$$\hat{E}_{i-1} = \texttt{LayerNorm}(E_{i-1}),$$
$$E_i = E_{i-1} + \texttt{MHA}(\hat{E}_{i-1}, \hat{E}_{i-1}, \hat{E}_{i-1}),$$
$$E_i = E_i + \texttt{FFN}\left(\texttt{LayerNorm}(E_i)\right),$$



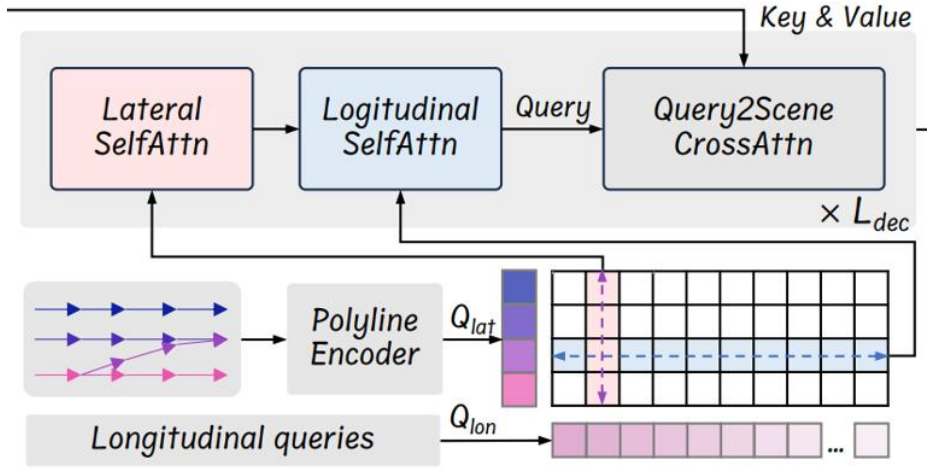
PLUTO | Semi-Anchor-based Decoder

PlanTF의 문제점

- 종횡방향 주행의 불균형: learning-based planning이 종방향 테스크는 잘 학습하지만, 횡방향 테스크는 잘 학습하지 못하는 문제
- Trajectory diversity 부족: trajectory를 한 번에 직접 예측하여 다양한 주행 행동을 반영하지 못함

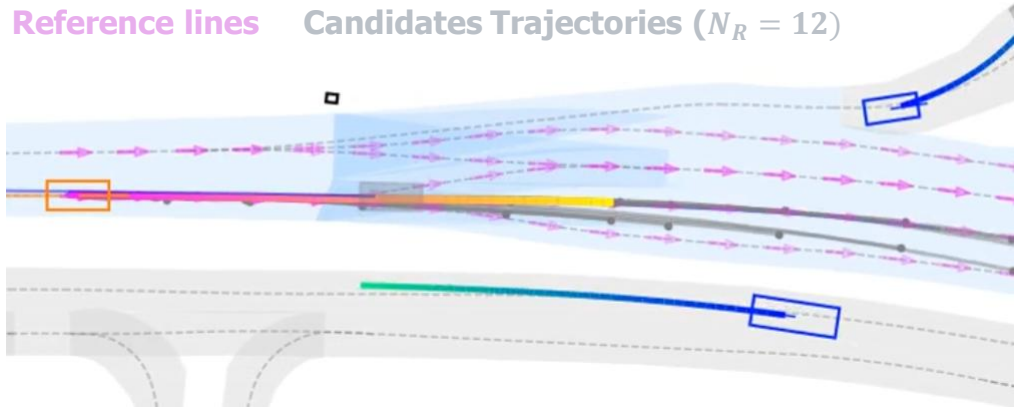
(해결책) Longitudinal-lateral aware model architecture (Semi-anchor-based decoding)

- Lateral query: AV 주변 lane segment를 탐색하여 **reference line** 추출
  즉, topological connection을 고려한 lane centerline을 lateral query로 사용
- Longitudinal query: $N_L(=12)$**개의 anchor-free 방식**(다양한 종방향 속도, 행동 표현 가능)으로 query 생성



Architecture Part of Semi-Anchor-based Decoding

Inference Result



An Example of Inference Result

PLUTO | Imitation Loss & Prediction Loss

Imitation Loss

- For migrating model collapse,
- $\hat{\tau}$: 이전에 계산한 여러 Reference line 중에서 횡방향 오차가 가장 작은 target reference line
- τ^{free}: 모델 자체의 학습 결과로 나온 trajectory

$$\mathcal{L}_i = \mathcal{L}_{reg} + \mathcal{L}_{cls}$$

$$\mathcal{L}_{reg} = \texttt{L1}_{smooth}(\hat{\tau}, \tau^{gt}) + \texttt{L1}_{smooth}(\tau^{free}, \tau^{gt})$$
$$\mathcal{L}_{cls} = \texttt{CrossEntropy}(\boldsymbol{\pi}_0, \boldsymbol{\pi}_0^*),$$

Prediction Loss

- PlanTF와 동일하게 모든 에이전트의 trajectory에 대한 L1 loss 계산

$$\mathcal{L}_p = \texttt{L1}_{smooth}(\boldsymbol{P}_{1:N_A}, \boldsymbol{P}^{gt}_{1:N_A})$$

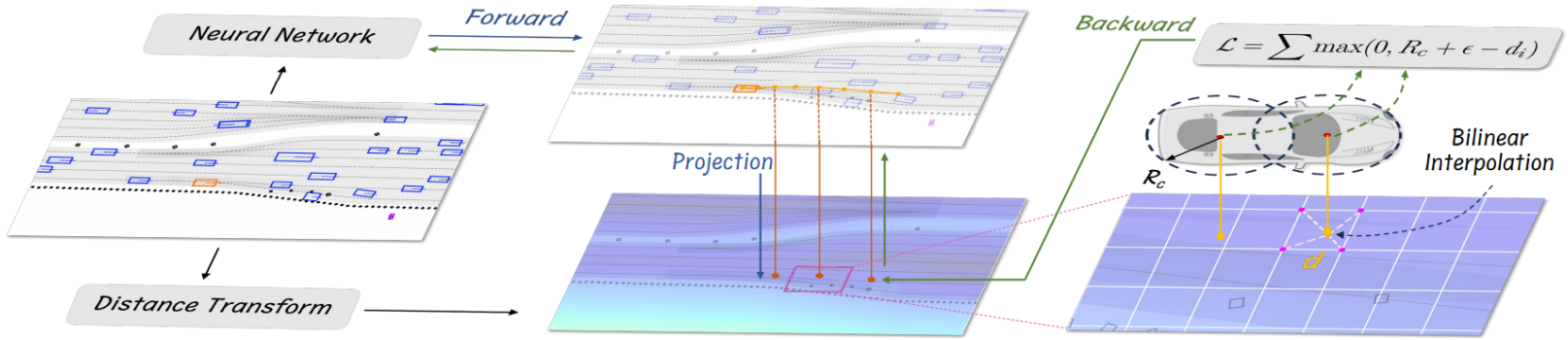
$$\boldsymbol{P}_{1:N_A} = \texttt{MLP}(E'_A)$$

PLUTO | Efficient Differentiable Auxiliary Loss

- **충돌 상황과 도로를 이탈하는 상황과 같은 discrete한(미분 불가능한) 문제 상황을 연속적이며 미분 가능하도록 Loss를 설계**
- **Cost map construction**
  - 주행 불가능 영역에 대한 constraint를 위해 Euclidean Signed Distance Field (ESDF) 사용하여 rasterized binary mask로 Image space에 매핑한 후, 이 마스크에 대해 거리 변환(distance transform) 수행
- **Auxiliary loss calculation**
  - 에고 차량의 3개의 원으로 모델링하여 각 원에 대해 가장 가까운 주행 불가능 영역까지의 거리를 계산
  - Signed distance value, d_i가 양수이면 주행 가능 영역, 음수이면 주행 불가능 영역
    → trajectory가 주행 가능 영역을 선택하도록 유도

$$\mathcal{L}_{aux} = \frac{1}{T_f}\sum_{t=1}^{T_f}\sum_{i=1}^{N_c}\max(0, R_c + \epsilon - d_i^t),$$

$R_c: circle's\ radius$
$\epsilon: safety\ threshold$
$d_i: signed\ distance\ value$



Proposed auzxiliary loss computation

PLUTO | Training Loss

■ 모델 학습을 위한 최종 Loss

- ▶ Imitation loss $\mathcal{L}_i$, prediction loss $\mathcal{L}_p$는 PlanTF와 동일
- ▶ 충돌 및 도로 이탈 방지를 위한 auxiliary loss와 causal confusion 해결을 위한 contrastive loss 추가

$$\mathcal{L} = w_1\mathcal{L}_i + w_2\mathcal{L}_p + w_3\mathcal{L}_{aux} + w_4\mathcal{L}_c$$

- $\mathcal{L}_i$: imitation loss
- $\mathcal{L}_p$: prediction loss
- $\mathcal{L}_{aux}$: auxiliary loss
- $\mathcal{L}_c$: contrastive loss

PLUTO 학습 실습

목표

- nuPlan 샘플 데이터셋을 통한 PLUTO 학습 프로세스 확인

방법

- Git repository를 통한 실습
- 모델 학습 코드 이해 및 실행
- 모델 학습 및 결과 확인

실습 코드

- https://github.com/ailab-hanyang/nuPlan_practice.git
- 해당 git repo의 readme 참고하여 실습 환경 구성

PLUTO 검증 실습

목표

- nuPlan 샘플 데이터셋을 통한 PLUTO 검증 프로세스 확인
  - 개발 모델 Closed-Loop 성능 검증

방법

- Git repository를 통한 실습
- 모델 추론 코드 이해 및 실행
- 모델 추론 및 결과 확인

실습 코드

- https://github.com/ailab-hanyang/nuPlan_practice.git
- 해당 git repo의 readme 참고하여 실습 환경 구성

THANK YOU
FOR YOUR ATTENTION