



國立臺北科技大學

資訊工程研究所

碩士學位論文

自動駕駛鳥瞰視角增強軌跡預測
Autonomous Driving Bird's Eye
View-Enhanced Trajectory Prediction



研究生：鄭傳脩

指導教授：謝東儒 博士

中華民國 一百一十五年 七月

國立臺北科技大學
研究所碩士學位論文口試委員會審定書

本校_____研究所_____君

所提論文，經本委員會審定通過，合於碩士資格，特此證明。

學位考試委員會

委員：_____



指導教授：_____

所長：_____

中華民國 一百零四 年 ○ 月 ○ 日

摘要

關鍵詞：駕駛路線產生、車道偵測、地圖定位、擴增實境、自動駕駛、車輛定位。

傳統軌跡預測模型在處理複雜城市場景時，僅依賴連續影像造成不精準問題進行改良。將原本輸入模型的前一影格影像替換為由導航路徑繪製之鳥瞰圖影像，以整合為更直觀的全域空間與導航資訊。本研究建立了一套軌跡預測的鳥瞰圖影像生成流程，透過地圖匹配與優化器，將帶有雜訊的全球定位系統軌跡校正至開放街圖的道路中心線上，進而生成精準的鳥瞰圖資料。實驗採用 nuScenes 資料集，利用卷積神經網路作為骨幹網路進行特徵提取，並結合門控循環單元捕捉駕駛過程中的動態時序資訊，最後透過多軌跡預測輸出候選行駛路徑。透過將導航路徑轉換為鳥瞰圖影像並結合視覺特徵，補足影像在全域導航意圖上的缺失，並為自動駕駛系統提供更穩健的路徑規劃效能。

ABSTRACT

Keywords: Driving route generation, lane detection, map localization, augmented reality, autonomous driving.

Traditional trajectory prediction models often suffer from reduced accuracy in complex urban environments due to their reliance on sequential images. To address this issue, this study replaces the previous-frame image with a BEV image generated from the navigation route, providing richer global spatial and navigation information. A BEV image generation pipeline is proposed, where noisy GPS trajectories are aligned to OSM road centerlines through map matching and optimization. The proposed method is evaluated on the nuScenes dataset using a CNN for feature extraction and a GRU to model temporal driving dynamics, followed by multi-trajectory prediction. By integrating BEV navigation information with visual features, the proposed approach improves trajectory prediction robustness and enhances path planning performance for autonomous driving systems.

致 謝

感謝指導教授這兩年在學術研究的協助與鼓勵，讓我在研究期間獲益良多，留下許多珍貴的回憶。指導教授和林宏祥教授從研究方向的確立、實驗設計的規劃，到論文的修正與完善，教授皆提供寶貴的建議與專業的見解，使我能夠逐步克服研究過程中所遭遇的困難，並順利完成本論文。

也感謝江佩穎教授和林宏祥教授擔任我的學位考試委員，委員們從不同角度提出的見解與建議，使本論文得以進一步修正與完善，提升研究內容的完整性與嚴謹性，並提供我學術論文的修改建議。

最後，特別感謝我的父母在求學期間始終給予我無私的支持與鼓勵。在研究過程中，無論面對挫折或壓力，總是給予我最大的包容與關懷，使我能夠專心投入研究並堅持完成學業。

目 錄

口試委員會審定書	i
摘要	ii
ABSTRACT	iii
致 謝	iv
目 錄	v
圖 目 錄	vii
表 目 錄	ix
第一章 導論	1
1.1 研究動機與目的	1
1.2 問題描述	1
1.3 研究貢獻	2
第二章 文獻討論	4
2.1 開源自動駕駛平台技術	4
2.2 端到端自駕模型	5
2.3 自動駕駛資料集	6
2.4 多視角相機影像轉換為鳥瞰視角	7
第三章 方法	9
3.1 模型 OP-Deepdive 架構	9
3.2 生成 BEV 資料	11
3.3 訓練使用資料集	14
3.4 加入 BEV 模型訓練流程	15
第四章 結果與討論	18
4.1 評估指標	18
4.1.1 平均歐幾里得誤差 (Mean Euclidean Error, MEE)	18
4.1.2 多層次命中率 (Multi-level Hit Rates)	19
4.2 模型 openpilot 訓練參數及 BEV 影像參數	20
4.3 模型結果比較	22
4.3.1 模型說明	22
4.3.2 量化比較	23

4.3.3	擾動模型表現較佳之場景	26
4.3.4	修正模型表現較佳之場景	28
4.3.5	原始模型表現較佳之場景	29
4.4	對其模型表現不佳原因分析	30
4.5	場景模型表現統計表	31
第五章	結論	33
5.1	結論	33
5.2	未來展望	33
參 考 文 獻		34



圖 目 錄

3.1	由美國公司 Comma.ai 提出的 Supercombo 模型的流程 (Pipeline) 架構，說明了從輸入數據到最終預測結果的處理流程。出處: https://github.com/OpenDriveLab/Openpilot-Deepdive	9
3.2	模型 Supercombo 在資料前處理採用視角轉換將影像縮放至 128×256 的尺寸達到節省運算資源並且看到的路面特徵都是一致的。出處: https://github.com/OpenDriveLab/Openpilot-Deepdive	10
3.3	由於 Supercombo 模型並未開源，故本研究採用 OpenDriveLab 上海人工智慧實驗室，Shanghai AI Laboratory 的研究團隊所提出重新實現的模型 OP-Deepdive 的網路結構圖，效果與未開源的 Supercombo 接近。出處: https://github.com/OpenDriveLab/Openpilot-Deepdive?tab=readme-ov-file	11
3.4	影像 BEV 生成流程。讀取特定的 JSON 數據文件，包含 OpenStreetMap 地圖數據以及經過擾動的 Perturbed GPS 軌跡。原始的 GPS 採樣點通常呈現雜亂或偏移的狀態，分佈在道路線段的周圍。為了修正這些偏差，系統會執行核心的地圖匹配邏輯，將這些擾動過的 GPS 點與地圖路網進行對齊，使所有座標點精確地回歸到道路中線上。最後，對齊後的數據會被轉換成 BEV 影像，產生從垂直視角俯瞰的道路環境圖案。	13
3.5	在上圖的配置中，系統採用傳統的時間序列輸入，將當前影格與前一影格同時餵入生成的模型中。這種方式主要依賴前後影像的視覺特徵變化，來讓模型理解車輛的運動與環境動態。接著，模型會透過腳本執行車道線預測，最後將結果轉化為視覺化的影片輸出。在下圖的改進流程中，系統將輸入源進行了替換。當前影格依然保留，但原本的前一影格被更具空間資訊的 BEV 所取代。	17
4.1	本研究提出的四種模型變體輸入資料配置示意圖。(a) Original：採用傳統連續車前影像作為基準輸入。(b) Perturbed：將前一幀替換為包含 GPS 定位誤差的噪聲 BEV 影像。(c) Corrected：將前一幀替換為經幾何優化、對齊至地圖道路線後的 BEV 影像。(d) Aligned：將前一幀替換為經道路線對齊並透過變換轉為車輛視角 (Ego-view) 的 BEV 影像。	23
4.2	不同區間下各方法的命中率比較。圖中展示了 original、perturbed、corrected 與 aligned 四種方法在四個不同區間 (0-10 至 30-40) 下的命中率變化趨勢。	24

4.3	圖中展示了 original、perturbed、corrected 與 aligned 四種方法在四個區間 (0-10 至 30-40) 下的歐氏誤差變化趨勢。	25
4.4	原始模型在夜間彎道場景中容易出現預測錯誤 (藍線偏離)，而 Perturbed 算法 (橘線) 可以有效地將路徑校正回與真實情況一致的方向。這證明了該修正機制在提升夜間行車安全性上的重要性。	27
4.5	在圖 (a) 的微偏右彎道場景中，可以明顯觀察到 Perturbed (橘線) 的預測效果優於原始的 OpenPilot (藍線)，其預測方向與真實路徑更為貼近。然而，在圖 (c) 停等紅綠燈的場景下，預測效果則出現了顯著的惡化，無論是 OpenPilot (藍線) 還是 Perturbed (橘線)，在面對紅綠燈停等這種複雜的都市靜止場景時，其預測軌跡皆表現不佳，出現了嚴重的方向扭曲與偏差，未能反映情境下應有的正確行駛或靜止狀態。	27
4.6	綠線方向與紅線始終保持一致，顯示出高度的準確性。相對而言，藍線 (OpenPilot) 代表原始系統的預測，其方向明顯偏離了真實路徑，存在預測錯誤	28
4.7	綠線 (Corrected) 的方向與紅線 (Ground Truth) 保持高度一致。這表明修正算法能夠有效地識別道路趨勢，並提供與實際路徑相符的預測結果，在彎道場景中，藍線未能準確跟隨紅線的轉向趨勢。	29
4.8	模型 Openpilot 路徑與 Ground Truth 具有較高的一致性，而 Corrected 路徑則出現明顯偏移，特別是在彎道路段，偏差情況更為顯著	30
4.9	將含有雜訊之 GPS 座標對齊至參考路線後，於車輛轉彎場景中，由於定位誤差與路徑匹配不穩定，對齊後的位置點可能在轉彎前後區域產生頻繁跳動現象。Aligned BEV 表示方式係依據車輛視角對路徑進行旋轉與對齊，因此位置點的不穩定變化將導致整體路徑表示持續發生旋轉與位移。此現象會造成輸入特徵在時間序列上的一致性下降，模型辨識路徑幾何特徵之困難度，對模型訓練穩定性與預測效能產生負面影響。	31

表 目 錄

4.1	模型 OP-Deepdive 訓練超參數設定	21
4.2	影像 BEV 超參數設定	22
4.3	夜間場景 1059 量化比較	26
4.4	道路切換場景 0557 量化比較	27
4.5	直線場景 0794 量化比較	28
4.6	場景 0093 (intersection-right-straight) 量化比較	29
4.7	場景 0013 (intersection-straight) 量化比較	30
4.8	Overall Model Performance Comparison	32
4.9	Per-Scenario Model Performance Comparison (Euclidean Error in meters) . . .	32



第一章 導論

1.1 研究動機與目的

近期以 Comma.ai 為首的研究機構挑戰了傳統規則的自動駕駛範式。Santana 與 Hotz (2016) 最早提出透過生成模型建立駕駛模擬器 [1]，奠定了端到端學習的基礎。隨後發布的 Comma2k19 數據集 [2] 證明了在低成本感測器架構下，透過大規模行為克隆 (Behavioral Cloning) 亦能達成穩健的車道保持任務，Comma four 是由自動駕駛公司 Comma.ai 推出的新一代車載硬體裝置，主要用來執行 OpenPilot 自動駕駛輔助系統，安裝在車輛前擋風玻璃附近，透過攝影機與車輛資料來協助自動駕駛功能運作。值得注意的是，Comma.ai 在最新研究 (2024) 中引入了世界模型架構 [3]，有效解決了模仿學習中的協方差偏移問題與 nuScenes [4] 強調的多模態精確標註不同，Comma.ai 的方法更偏向於語義空間的潛在表徵，這為未來自動駕駛邁向通用人工智能提供了新的研究路徑。然而 OpenPilot 未公開他們的模型訓練方法，OpenDriveLab 研究團隊嘗試重現 OpenPilot 的訓練細節 (OP-DeepDive) [5]，並在公開的評估指標資料集上進行測試。OP-DeepDive 主要著重於解析 OpenPilot 的內部實作與資料流架構，包含模型推論流程、感測器資料整合機制以及控制策略運作方式。透過深入分析各模組之間的互動關係，研究者能更清楚理解系統如何將原始感測資料轉換為車輛控制指令。此外，OP-Deepdive 亦有助於開發者評估系統效能、優化演算法設計，並提供進一步擴展自動駕駛功能的技術基礎。

1.2 問題描述

路徑預測模型 OP-DeepDive 為一種以影像為輸入、生成導航路徑之深度學習模型。其原始架構設計為接收兩張連續時間影像作為輸入，包括：當前時間點之相機影像、

前一時間點之相機影像此種雙幀輸入設計的核心假設在於：透過時間序列中相鄰影像的差異，模型可隱式學習場景 [6] 中的運動資訊、物體位移與相對幾何變化，進而提升路徑預測的穩定性與準確性。然而，連續影像之時間資訊雖可提供運動線索，但仍存在以下限制：時間差異僅能反映短期運動變化 [7]，缺乏全域導航意圖資訊。模型需自行從像素變化中學習運動特徵，增加訓練複雜度。影像對齊誤差與動態物體干擾可能影響特徵學習穩定性。在實際導航應用中，系統通常可同時取得高階導航路徑資訊 [8] (例如全域規劃器產生之路徑)。因此，若能將此導航先驗資訊整合至模型輸入，可能有助於改善路徑生成效果。基於此動機，本研究提出將 OP-DeepDive 原本輸入之「前一幀相機影像」替換為「由導航路徑繪製之俯瞰視角影像 (Bird's-Eye View, BEV)」之方法，並探討其對路線生成性能之影響。

1.3 研究貢獻

深度學習已廣泛應用於自動駕駛軌跡預測，其中 OP-DeepDive 能透過連續影像學習車輛行駛軌跡。然而，其於 nuScenes 資料集的實驗結果顯示，在複雜城市道路場景中的預測表現仍有改善空間。由於僅依賴連續影像作為模型輸入，難以充分獲取全域道路結構與導航意圖，進而影響路徑生成的準確性。因此，本研究導入地圖導航資訊，提出以導航路徑生成之 BEV 影像取代模型原有的前一幀影像，建立完整的導航路徑轉換流程，將地圖資訊與視覺特徵進行整合，以提供更完整的空間資訊。此外，本研究比較原始 OP-DeepDive 與導入導航路徑 BEV 影像之 OP-DeepDive 在路徑生成任務上的預測效能，分析不同輸入架構對模型性能的影響，並驗證所提出方法對提升軌跡預測準確性與路徑規劃穩健性的成效。

本研究主要貢獻如下：

3

- **改善 OP-DeepDive 在 Nuscenes 資料集表現不理想的問題：**OP-DeepDive 的實驗結果表示在 Nuscenes 資料集上表現並不理想，由於 Nuscens 的資料以平面路段較多路段較為複雜，模型預測上難度較高，我們嘗試加入地圖導航資訊來改善結果不理想的問題

- **將其中一個模型輸入更換為導航路徑繪製成的俯瞰視角影像：**提出以導航路徑俯瞰視角影像取代時間前一幀影像之輸入架構設計，建立導航路徑轉換為 BEV 影像之完整技術流程。
- **比較兩種輸入設計在路徑生成任務上的性能差異：**將原來的 OP-DeepDive 模型與修改為地圖導航的 OP-DeepDive 模型所生成的路徑進行性能比較。



第二章 文獻討論

2.1 開源自動駕駛平台技術

隨著軌跡預測技術的快速發展，開源自動駕駛平台成為研究與原型開發的重要工具。這些平台通常提供感知、規劃、控制模組，以及多車型、多場景支援，使研究者能快速進行算法測試與系統整合。OpenPilot [9] 是由 Comma.ai 開發的開源自動駕駛系統，自 2016 年起提供給社群使用 [10]。OpenPilot 支援 Level 2 自動駕駛功能，包括車道保持 (Lane Keeping Assist, LKA) [11]、自動巡航 (Adaptive Cruise Control, ACC) 以及部分自動換道。其控制策略主要基於行為克隆 (Behavior Cloning [12])，透過學習人類駕駛行為進行決策。技術上，OpenPilot 可運行於 EON Dashcam 硬體 [13]，並採用卷積神經網路 (CNN) 進行感知模組設計。此外，平台完全開源，支援多種量產車型，並由社群持續貢獻車型接口與安全監控功能。OpenPilot 主要應用於高速公路半自動駕駛及研究教育領域。

百度開發的 Apollo [14] 開源自動駕駛平台，專注於城市道路與複雜交通環境。其核心模組包括多模態感知 (Camera、LiDAR、Radar)、高精地圖路徑規劃，以及車輛控制 (如 PID 與 Model Predictive Control)。Apollo 採用模組化設計，支援多車型與仿真環境，並提供完整的感知、規劃與控制流程。該平台適用於城市道路自動駕駛測試、企業原型開發及學術研究，尤其在多車型與複雜交通情境下具有較高的靈活性。開源自動駕駛模擬平台 CARLA (CAR Learning to Act) [15]，專注於城市環境與自動駕駛算法測試。CARLA 提供高度可自訂的模擬環境，支援多車輛、多行人、多傳感器模擬，並可即時收集 Camera、LiDAR、Radar、GPS 與 IMU 等感知數據。技術上，CARLA 採用 Unreal Engine 4 [16] 渲染高精度場景，使研究者能在不同天氣、光照與交通條件下測試演算法。CARLA 主要應用於自動駕駛算法訓練、驗證及深度學習。

2.2 端到端自駕模型

傳統自動駕駛系統多採用模組化架構，將整體系統劃分為感知、定位、行為預測、路徑規劃與控制等子模組，各模組分別設計與最佳化。此類方法具備良好的可解釋性與工程可控性，然而模組間誤差易逐層累積，且系統設計高度仰賴人工特徵工程與專家經驗。為克服上述限制，近年研究逐漸關注端到端自動駕駛模型，透過深度學習方法直接由原始感測資料（如影像、LiDAR 或多模態資料）輸出駕駛控制指令或行駛軌跡，藉此實現整體系統的聯合最佳化。端到端自動駕駛模型的核心理念在於取消明確的人為模組劃分，並以單一學習目標函數進行訓練，使模型能自動學習從感測輸入到控制輸出的映射關係。相較於傳統管線式系統 [17]，端到端方法能降低系統複雜度，並避免中間模組設計不當所造成的資訊損失。Bojarski 等人 [10] 提出首個具代表性的端到端自動駕駛模型，利用卷積神經網路（CNN）直接從前方攝影機影像預測轉向角度。該研究證實深度學習模型可在無需手動特徵工程的情況下學習人類駕駛行為，奠定端到端自駕研究的基礎。此類方法主要以行為複製為訓練方式，透過模仿人類駕駛資料學習控制策略，但在罕見或極端情境下容易產生泛化問題 [18]。為提升端到端模型於實際車載系統中的可行性，Kocić 等人 [19] 提出適用於嵌入式平台的輕量化端到端深度神經網路。該模型在維持即時性的同時，仍能提供合理的控制精度，顯示端到端方法具備實務部署潛力。除監督式學習外，部分研究引入深度強化學習 [20]（Deep Reinforcement Learning, DRL）進行端到端訓練，使車輛能透過與環境互動學習最優駕駛策略。此類方法可直接輸出連續控制指令，並具備自我探索能力，適合用於複雜決策問題。然而，強化學習方法多仰賴模擬環境訓練，其在真實世界中的安全性與遷移能力仍有待進一步驗證。近年已有多篇文獻針對端到端自動駕駛進行系統性回顧。Singh [21] 從感測輸入、模型架構、訓練策略與評估方法等面向，完整整理端到端自駕研究發展脈絡，並指出安全性與可解釋性為未來關鍵挑戰。此外，Chib 與 Singh [22] 進一步比較不同端到端架構與資料形式，並探討混合方法結合模組化與端到端優點的可行性，顯示當前趨勢已逐步由純端到端轉向可解釋且可驗證的端到端架構 [23,24]。

2.3 自動駕駛資料集

目前自動駕駛研究領域中 NuScenes [4] 為廣泛使用的大型多感測器資料集之一，由 Aptiv 與 nuTonomy 團隊所建立，並於 2019 年正式公開。該資料集的主要目標是提供完整且多樣化的城市道路駕駛資料，以支援自動駕駛系統在環境感知、目標偵測、物件追蹤以及場景理解等相關研究。相較於早期僅提供影像或單一感測器資料的資料集，NuScenes 強調多感測器整合與時間同步，使研究者能夠在更接近真實自動駕駛系統的情境下進行演算法開發與測試。NuScenes 的資料主要透過實際測試車輛在城市道路環境中蒐集，涵蓋多種複雜的交通情境，例如繁忙路口、行人穿越、車輛轉彎以及不同交通密度的道路環境。整個資料集包含 1000 個完整駕駛場景，每個場景長度約為 20 秒，總計提供約 1.4 百萬張影像以及大量的感測器紀錄資料。為了提供完整的環境感知能力，NuScenes 測試車輛搭載多種感測器設備，包括六個環景 RGB 攝影機、五個毫米波雷達以及一個 32 線雷射掃描儀 (LiDAR)，並透過精確的時間同步機制確保各感測器資料能夠對齊。這樣的感測器配置使資料集能夠提供 360 度全方位的环境觀測資訊，並支援多感測器融合演算法的研究。在標註方面，NuScenes 提供相當完整的資料標記，包括 3D 物件邊界框、物件類別標籤以及跨時間的物件追蹤 ID。資料集中涵蓋多種交通參與者類別，例如車輛、行人、自行車與交通設施等。此外，NuScenes 亦提供高精地圖資訊，例如車道拓撲、道路邊界與交通區域，使研究者能夠進一步進行場景理解與地圖相關研究。由於其資料規模龐大、標註資訊完整且包含多種感測器來源，NuScenes 已成為自動駕駛感知任務的重要基準資料集之一，並被廣泛應用於 3D 目標偵測、軌跡預測以及多感測器融合等研究領域。

comma2k19 為由自動駕駛技術公司 Comma.ai 所發布的開放資料集 [2]，主要設計目的是支援車輛定位、地圖建立與感測器融合相關研究。與許多大型自動駕駛資料集著重於環境感知不同，comma2k19 的核心重點在於提供高精度的定位資料，使研究者能夠開發與評估基於視覺與慣性感測器的定位演算法。comma2k19 資料集包含約 2000 公里的實際道路行駛資料，主要蒐集於美國加州高速公路 US-280 的路段。資料蒐集方式是透過測試車輛在同一路線上多次往返行駛，藉此取得大量重複軌跡資料。這樣的資料蒐集策略使研究者能夠觀察同一路段在不同時間、不同交通條件以及不同光照環境下的影像變化，進而研究定位演算法在長期運行與環境變化下的穩定性與可靠性。在

感測器配置方面，comma2k19 主要使用車輛前方攝影機作為視覺資料來源，並搭配全球導航衛星系統（Global Navigation Satellite System, GNSS）與慣性量測單元（Inertial Measurement Unit, IMU）[25] 進行定位資料蒐集。此外，資料集中亦提供經過後處理與最佳化的高精度定位結果，作為演算法評估與比較的基準。透過 GNSS 與 IMU 的融合處理，資料集能夠提供相對準確的車輛位置與姿態資訊，使其特別適用於視覺定位、視覺里程計以及感測器融合定位等研究。相較於其他自動駕駛資料集，comma2k19 的感測器種類相對簡單，並未包含 LiDAR 或多視角攝影機系統。然而，其長距離連續行駛資料以及重複軌跡特性，使研究者能夠有效分析定位演算法在長時間運行過程中的誤差累積問題。此外，該資料集提供完整的時間同步感測器資料與高精度定位結果，因此在視覺導航、自動駕駛定位與地圖建構等研究領域中具有相當重要的應用價值。

2.4 多視角相機影像轉換為鳥瞰視角

Camera to BEV Transformer [26] 是一類能將多視角相機影像直接轉換為鳥瞰視角表示的深度學習架構，在自動駕駛的環境感知與場景理解任務中特別重要。這類方法的目標是將原本帶有透視變形的前視或環視彩色影像轉換到一個統一的平面表示，使車道線結構、物體位置與場景幾何能夠在 BEV 空間中直觀表現，更方便於後續的 3D 物體偵測、地圖建構與決策規劃流程。該方法利用 Transformer 的 Cross Attention 機制 [27]，將 BEV 空間中的每一個網格點設計為序列，並從多個相機影像提取出的特徵中作為鍵與值進行關聯計算。透過這種方式，每個 BEV 位置都能主動從不同視角影像中擷取與自己對應的資訊，進而聚合成完整的俯瞰特徵圖。這種方法不需要顯式的深度估計，而是透過學習到的注意力權重自動建立影像與空間之間的對應關係，因此能以端到端方式訓練，並對遮擋與視角差異具有較好的魯棒性。在此之前，Lift, Splat, Shoot 等人 [28] 提出了一種較早期的 Camera to BEV 架構，其流程包含三個步驟：首先將 2D 影像特徵 Lift 到 3D 空間的視錐體中，再將這些 3D 特徵 Splat 投影到 BEV 平面，最後在 BEV 空間中進行偵測任務。這種方法建立了從影像到俯瞰表示的幾何轉換基礎，後續許多 Transformer 類方法皆在此概念上發展。另一個相關工作 [29] 則著重於效能與即時性，透過視角轉換模組將影像特徵快速映射至 BEV 空間，以達到高效率的 3D 物體偵測。其中代表性的工作是 BEVFormer，這個架構利用 Transformer 的注意力機制

實現多視角影像到 BEV 的轉換，並引入時序信息融合以捕捉歷史 BEV 特徵。具體而言，BEVFormer 將 BEV 平面上的每一個網格視為一個鍵，再透過空間自注意力從不同相機視角的影像特徵中提取對應信息，將這些不同視角下的資訊聚合成 BEV 特徵；同時利用 temporal self attention 將歷史 BEV 表示和當前的 BEV 進行融合以提升時序感知能力，使模型對運動物體和前後影像的連貫性有更好的理解。整個流程可以端到端地學習，不需要顯式預測深度即可在俯瞰視角上建構特徵圖。

為了驗證這種方法的有效性，作者在 nuScenes [4] 這個大規模自動駕駛基準資料集上進行了實驗。nuScenes 資料集包含大量多鏡頭 RGB 影像、激光雷達（LiDAR）掃描、雷達、GPS/IMU 和 HD Map 等豐富感知數據，並提供標準的評估指標如 nuScenes Detection Score、mAP（平均精度）及其他誤差衡量指標。實驗結果顯示，BEVFormer 在 nuScenes 測試集上達到了 56.9 的 NDS 指標，這比之前最佳的基於影像的方法如 DETR3D [30] 高出約 9.0 個百分點，而且性能達到了部分基於 LiDAR 的基線方法水平，證明了使用 BEV based Transformer 從多視角影像中學習三維空間特徵是可行且有效的。該模型在速度估計和低能見度場景的物體檢測表現上也有顯著改進，進一步體現出 BEV 表示對於理解複雜場景的重要性。影像轉 BEV 提供了一種有效的方式把多視角影像資訊映射到 BEV 空間，這種表示方式能消除視角上的干擾，使場景結構更一致並有利於 3D 空間推理，而在像 nuScenes 這樣具有挑戰性的資料集上也能取得領先的性能。這樣的 BEV 表示不只是對感知任務有用，還能在合併公開地圖、路徑規劃或定位任務中作為統一的特徵基礎，實現從影像到 BEV，再由 BEV 與全局地圖或導航路徑匹配的完整流程。

第三章 方法

3.1 模型 OP-Deepdive 架構

軌跡預測模型 OP-Deepdive [5] 是一個受 Openpilot 的 Supercombo 模型啟發而開發的端到端軌跡規劃模型。其核心目標是重新實現 Supercombo 的功能，在不需要顯式感知模組的情況下，直接透過相機影像預測車輛未來的行駛軌跡。OP-Deepdive 的整體架構與 Supercombo 高度相似，旨在保持輸入與輸出格式的一致性。Supercombo 模型的架構為輸入端：連續兩幀的影像, 前一幀與當前幀、代表高階駕駛指令的向量，以及標示左右駕交通習慣的布林值。骨幹網路：影像輸入後會經過骨幹網路進行特徵提取，隨後將特徵圖展平並與駕駛指令向量拼接。時序處理與預測頭：為了捕捉動態資訊，特徵向量會送入 GRU 模組。最後透過全連接層構成的預測頭輸出最終結果，由於不同使用者的裝置安裝位置可能存在偏差，系統要求進行相機校準以獲得外參。輸出端：最終的預測結果包含多項維度，規劃軌跡、車輛姿態、車道線、前車狀態、中繼資料以及期望狀態。

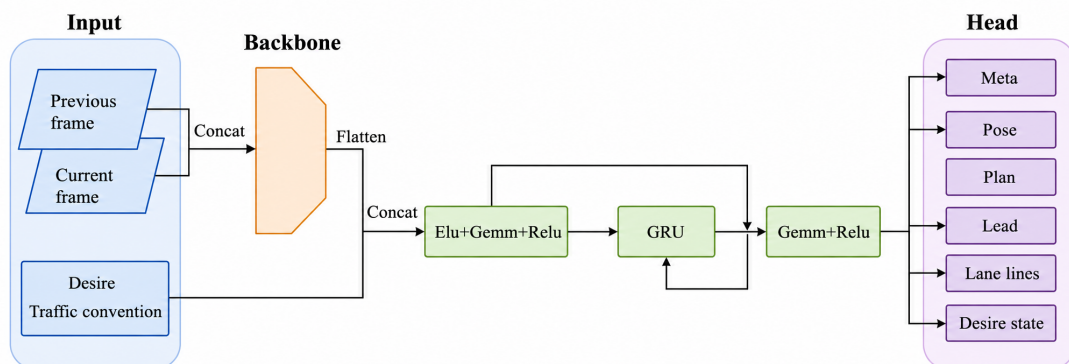


圖 3.1: 由美國公司 Comma.ai 提出的 Supercombo 模型的流程 (Pipeline) 架構，說明了從輸入數據到最終預測結果的處理流程。出處:<https://github.com/OpenDriveLab/Openpilot-Deepdive>

在影像輸入網路之前，必須經過一系列預處理步驟以確保數據的標準化。首先是視角轉換，由於不同使用者的裝置安裝位置可能存在偏差，系統要求進行相機校準以獲得外參。透過影像扭曲操作，將原始影像轉換為標準「虛擬相機」視角，這使得模型能專注於關鍵的視野區域，並將影像縮放至 128×256 的尺寸。在色彩空間方面，雖然 Openpilot 原始系統使用 YUV-422 格式以降低運算開銷，但考慮到公用數據集多以 RGB-888 為預設格式，OP-Deeptide 選擇直接採用 RGB-888 格式進行處理。最終，模型會將連續兩幀的影像進行拼接作為輸入，以提供初步的時序資訊。

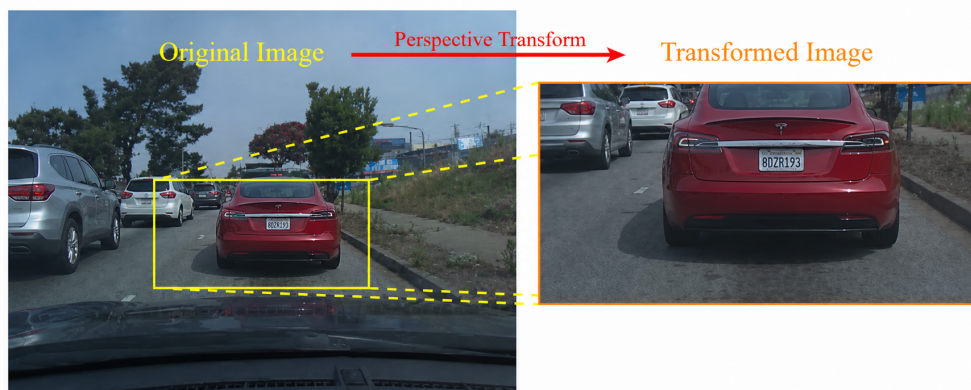


圖 3.2: 模型 Supercombo 在資料前處理採用視角轉換將影像縮放至 128×256 的尺寸達到節省運算資源並且看到的路面特徵都是一致的。出處:<https://github.com/OpenDrive-Lab/Openpilot-Deeptide>

OP-Deeptide 的網路架構主要由特徵提取骨幹、時序處理模組與預測頭三個部分組成，骨幹網路：採用 EfficientNet-B2 作為特徵提取器。該網路利用 AutoML 與模型縮放技術，在僅有 920 萬個參數的情況下展現出極高的運算效率與準確度。輸入張量經骨幹網路處理後，會透過卷積層降維並展平為長度 1024 的特徵向量。時序模組：為了捕捉駕駛過程中的動態時序資訊，系統在特徵向量後方連接了一個寬度為 512 的門控循環單元。這使得模型能夠具備記憶能力，參考過去的駕駛狀態來優化規劃結果。預測頭：GRU 的輸出會送入數個全連接層構成的預測頭。模型最終會輸出 $M=5$ 條可能的候選軌跡，每條軌跡由 $N=33$ 個在車輛座標系下的三維座標點以及一個置信度分數組成。

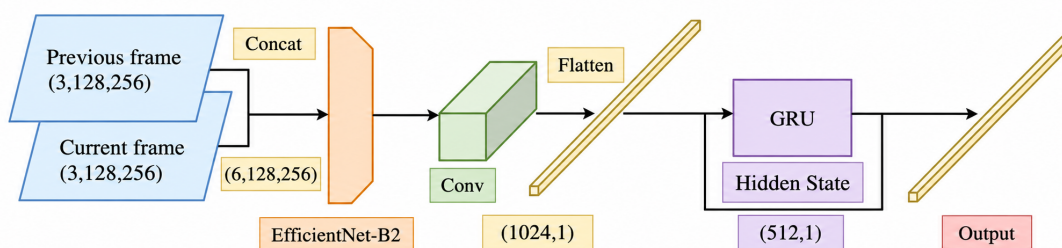


圖 3.3: 由於 Supercombo 模型並未開源，故本研究採用 OpenDriveLab 上海人工智慧實驗室，Shanghai AI Laboratory 的研究團隊所提出重新實現的模型 OP-Deepdive 的網路結構圖，效果與未開源的 Supercombo 接近。出處:<https://github.com/OpenDriveLab/Openpilot-Deepdive?tab=readme-ov-file>

3.2 生成 BEV 資料

本研究開發一套針對車輛定位軌跡的修正方法，並生成可視化 BEV 圖像。首要步驟是針對輸入的原始 JSON 資料進行解析。系統會自動掃描指定的資料路徑，從中提取兩組關鍵座標序列，第一組為 Route OSM，作為道路中心線的基準點；第二組為 GPS Perturbed，代表車輛感測器回傳且未經校正的原始行駛軌跡。隨後，系統將座標轉換為 PyTorch 的張量格式，以利後續利用 GPU 與高效率數學運算進行軌跡最佳化。接著，系統透過疊代運算尋找同時符合道路幾何限制與車輛運動特性的最佳化軌跡。程式將軌跡座標設為可優化參數，並使用 Adam 優化器進行 1000 次疊代。每一次疊代皆根據以下五項約束進行調整。首先為地圖對齊，透過地圖對齊損失將軌跡點逐步拉近至道路中心線。為提升演算法對離群值的魯棒性，本研究採用分段誤差函數，當軌跡點與最近路網線段的距離小於 2 公尺時，採用平方懲罰以提升對齊精度；當距離超過閾值時，則改為線性懲罰，以避免極端雜訊造成整體路徑受到過度拉動。其次為物理平滑，透過二階差分模擬車輛加速度，以降低感測誤差造成的座標跳動，使軌跡於轉彎處更加平順。系統藉由對連續三幀座標進行二階差分運算，對過大的曲率變化施加懲罰，確保生成軌跡於數學上具有良好的連續性，使其在視覺與物理層面皆更加合理。第三為速度穩定，藉由監控相鄰影格間的位移量，使各點之間的距離分布趨於一致，避免軌跡出現速度忽快忽慢的異常現象。系統透過限制相鄰影格位移長度的標準差，有效降低定位訊號遮蔽或漂移所造成的突發性位移變化，使車輛速度更符合實際行駛特性。

第四為正向行駛約束，系統計算局部移動向量與整體行進方向的點積，若偵測到軌跡出現後退或重疊等不合理現象，便施加較大的懲罰，以確保軌跡沿時間序列保持單向且連續的行進方向。最後為原始位置約束，確保修正後的軌跡雖逐漸貼合道路中心線，但仍維持與原始觀測值的一致性，避免因過度修正而導致軌跡變形。系統透過絕對誤差限制最佳化後的點位與原始觀測點保持適當距離，使原始定位資訊作為整體最佳化過程的參考依據，避免模型過度擬合道路幾何而產生不合理的偏移。

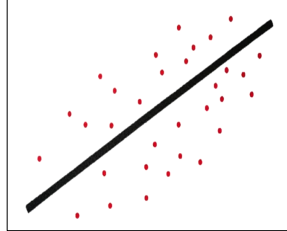
在全域優化產生初步平滑的路徑後，程式會進行最後的強制吸附處理，將點位精確投影到地圖線段上。為了防止在十字路口或對向車道產生錯誤匹配，此階段引入了「方向過濾器」：系統會判斷車輛目前的移動方向與地圖線段的方向向量是否一致。如果某個地圖線段雖然距離很近，但方向與車流相反，系統會對該路段施加極大的距離懲罰，從而引導系統選擇方向正確的車道進行投影。程式將修正後的地理座標轉化為具備研究價值的視覺資料。系統會以車輛當前座標為原點進行平移旋轉，將全球座標系轉為相對座標系。透過設定精確的物理比例尺例如每公尺對應 10 像素，程式會在黑色背景上精確繪製出白色道路邊界與彩色車輛位置。最終，這些校正後的座標會儲存為新的 JSON 檔案，並輸出每一時刻的路徑鳥瞰圖影像，提供給後續的自動駕駛感知模型或行為規劃模組使用，我們想透過調整生成圖形的參數來進行後續的效果比較。

Input

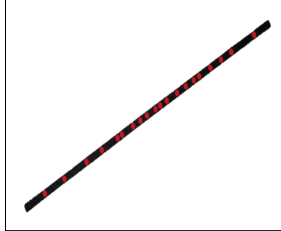
1. JSON file:
 - Open street map(OSM),
 - Perturbed GPS



2. Generate visual image



3. Align perturbed GPS (Map Matching)



Output

4. Convert to BEV image



圖 3.4: 影像 BEV 生成流程。讀取特定的 JSON 數據文件，包含 OpenStreetMap 地圖數據以及經過擾動的 Perturbed GPS 軌跡。原始的 GPS 採樣點通常呈現雜亂或偏移的狀態，分佈在道路線段的周圍。為了修正這些偏差，系統會執行核心的地圖匹配邏輯，將這些擾動過的 GPS 點與地圖路網進行對齊，使所有座標點精確地回歸到道路中線上。最後，對齊後的數據會被轉換成 BEV 影像，產生從垂直視角俯瞰的道路環境圖案。



Algorithm 1 Global Optimization-based Map Matching with Directional Snapping

Input: OSM Map Base File *osm_json*, Perturbed GPS File *perturbed_json*, Output Path *output_json*

Output: Optimized and Snapped Trajectory saved to *output_json*

```
1 if not Exists(osm_json) or not Exists(perturbed_json) then
2   return
3  $P_{osm} \leftarrow \text{LoadPoints}(\text{osm\_json})$   $D_{raw}, F_{extracted}(\text{perturbed\_json})$ ;  $\triangleright$  Coordinates and filenames
4 if  $D_{raw}$  is empty then
5   return
6  $T_{raw} \leftarrow \text{Tensor}(D_{raw})$   $S_A, S_B \leftarrow P_{osm}[0:-1], P_{osm}[1:]$ 
7  $T \leftarrow \text{Parameter}(T_{raw})$ ;  $\triangleright$  Trajectory parameters to be optimized
8 Set Hyperparameters:  $W_{align}, W_{smooth}, W_{velocity}, W_{forward}, W_{consis}, LR, ITERS$  Optimizer  $\leftarrow$ 
   Adam( $\{T\}$ ,  $lr = LR$ )
9 for  $i \leftarrow 1$  to  $ITERS$  do
10   Optimizer.zero_grad()
11    $\triangleright$  A. Alignment Loss (Distance to nearest map point)
    $M_{dist} \leftarrow \text{PairwiseDistance}(T, P_{osm})$   $d_{min} \leftarrow \min_{\text{row}}(M_{dist})$   $\mathcal{L}_{align} \leftarrow$ 
   mean( $\mathbf{I}(d_{min} < 2.0) \cdot 0.5d_{min}^2 + \mathbf{I}(d_{min} \geq 2.0) \cdot 2.0(d_{min} - 1.0)$ )
12    $\triangleright$  B. Smoothness Loss (Acceleration minimization)
    $A \leftarrow T[2:] - 2T[1:-1] + T[:-2]$   $\mathcal{L}_{smooth} \leftarrow \text{mean}(\|A\|_2^2)$ 
13    $\triangleright$  C. Velocity Consistency Loss (Uniform step length)
    $\Delta T \leftarrow T[1:] - T[:-1]$   $L_{step} \leftarrow \|\Delta T\|_2 + \epsilon$   $\mathcal{L}_{velocity} \leftarrow \text{std}(L_{step})$ 
14    $\triangleright$  D. Forward Constraint Loss (Prevent backward movement)
    $\vec{v}_{main} \leftarrow \frac{T[-1] - T[0]}{\|T[-1] - T[0]\|_2 + \epsilon}$   $\mathcal{L}_{forward} \leftarrow \text{mean}(\max(0, -(\Delta T \cdot \vec{v}_{main})))$ 
15    $\triangleright$  E. Consistency Loss (Anchor to original noisy inputs)
    $\mathcal{L}_{consis} \leftarrow \text{mean}(\|T - T_{raw}\|_1)$ 
16    $\mathcal{L}_{total} \leftarrow W_{align}\mathcal{L}_{align} + W_{smooth}\mathcal{L}_{smooth} + W_{velocity}\mathcal{L}_{velocity} + W_{forward}\mathcal{L}_{forward} + W_{consis}\mathcal{L}_{consis}$ 
    $\mathcal{L}_{total}.\text{backward}()$  Optimizer.step()
17  $T_{snap} \leftarrow T$   $\vec{V}_{seg} \leftarrow S_B - S_A$ ,  $\vec{U}_{seg} \leftarrow \frac{\vec{V}_{seg}}{\|\vec{V}_{seg}\|_2 + \epsilon}$  for  $k \leftarrow 0$  to  $\text{Len}(T_{snap}) - 1$  do
18    $\vec{v}_{ego} \leftarrow (k < \text{Len}(T_{snap}) - 1) ? (T_{snap}[k+1] - T_{snap}[k]) : (T_{snap}[k] - T_{snap}[k-1])$ 
    $\triangleright$  Vectorized projection onto all road segments
19    $t \leftarrow \text{clamp}\left(\frac{(T_{snap}[k] - S_A) \cdot \vec{V}_{seg}}{\|\vec{V}_{seg}\|_2^2 + \epsilon}, 0.0, 1.0\right)$   $P_{proj} \leftarrow S_A + t \cdot \vec{V}_{seg}$   $D_{base} \leftarrow \|T_{snap}[k] - P_{proj}\|_2^2$ 
    $\triangleright$  Direction validation
20    $\vec{d}_{dot} \leftarrow \vec{v}_{ego} \cdot \vec{U}_{seg}$  Penalty  $\leftarrow (\vec{d}_{dot} < 0) ? 100.0 : 0.0$ 
21    $idx \leftarrow \arg\min(D_{base} + \text{Penalty})$   $T_{snap}[k] \leftarrow P_{proj}[idx]$ 
22  $O_{final} \leftarrow \text{Combine}(T_{snap}, F_{extracted})$  SaveToJSON( $O_{final}$ , output_json)
```

3.3 訓練使用資料集

本研究採用目前自動駕駛領域中最具代表性的多模態資料集 nuScenes 作為實驗基準。該資料集由 Motional 團隊開發，包含了在大規模波士頓與新加坡城市道路中所採集的各類複雜場景。nuScenes 的核心優勢在於其提供了 360 度的全景感測資訊，包括 6 顆環景攝影機、長距離雷達以及高解析度光達，並針對點雲數據進行了密集的語義標註。在資料分割方面，本實驗遵循官方標準，共包含 1,000 個場景，每個場景約 20

秒長。具體分割比例為：訓練集：包含 700 個場景。驗證集：包含 150 個場景。測試集：包含 150 個場景。本研究採用的預處理資料集以 JSON 格式儲存，詳盡記錄了每一影格的多模態感測資訊與空間狀態。每一筆資料實例均包含以下核心欄位，用以建構完整的視覺導航環境：在視覺資訊方面，`imgs` 欄位指向了環景攝影機的原始影像路徑如 `CAM_FRONT`，提供了車輛前向視角的關鍵語義特徵。為了實現 2D 影像與 3D 空間之間的精確映射，資料中封裝了完整的相機幾何參數：`camera_intrinsic` 記錄了相機內參矩陣，涵蓋焦距與光心位置；而 `camera_extrinsic` 則定義了相機相對於車輛本體座標系的旋轉與平移關係。此外，為了優化座標轉換的運算效率，資料中亦預先計算並儲存了相機的外參逆矩陣包括 `camera_rotation_matrix_inv` 與 `camera_translation_inv`，這對於將特徵從影像空間投影回鳥瞰圖 BEV 空間之逆運算至關重要。在運動軌跡與地面真值方面，`future_poses` 欄位定義了車輛在未來時間步長內的空間位姿序列。每一組位姿數據皆包含三維空間座標 (x, y, z) ，代表車輛在相對座標系下的預期行駛軌跡。這些高精度的位姿數據結合了車輛自身的里程計資訊與經過校正後的導航數據，作為本研究模型訓練時的監督信號。透過整合上述視覺內外參數與未來路徑點，本研究得以在 BEV 視角下建立一個具備幾何一致性的訓練框架，使模型能夠學習從環景影像特徵到未來軌跡規劃間的非線性映射關係。為了強化模型對空間環境的理解，本研究在 nuScenes 原始數據的基礎上，進一步整合了鳥瞰圖資訊。BEV 透過將地圖拓撲與車輛軌跡投影至二維俯視平面，消除了攝影機視角中的透視縮放與遮擋問題，提供模型更為直觀的幾何導航特徵。

3.4 加入 BEV 模型訓練流程

本研究之訓練流程建構於 PyTorch 深度學習框架之上，整合了序列化的影像資訊與車輛動態約束。訓練程序首先透過 `SequencePlanningDataset` 類別進行高效能資料讀取，該機制會自動從指定的資料分割路徑中提取連續影格影像與經由全域優化校正後的未來位姿真值（Ground Truth）。為了確保模型能穩定收斂，影像輸入前會經過 `transforms` 處理，包括調整解析度至 128×256 像素並進行正則化，以利於骨幹網路進行特徵編碼。模型核心架構採用了 `SequenceBaselineV1`，其內部整合了 `SequencePlanningNetwork`。此網路具備處理時間序列特徵的能力，利用

門控循環單元 (Gated Recurrent Unit, GRU) 作為時間演化模組，其隱藏層維度設定為 512，用以記憶並傳遞跨影格的運動特徵。在訓練過程中，系統採用「多軌跡預測」(Multiple Trajectory Prediction, MTP) 策略，模型會同時輸出 M 條候選路徑 (本實驗設定 $M = 5$) 及其對應的置信度得分。損失函數則結合了分類損失與迴歸損失，後者專門針對軌跡的 x, y, z 座標進行平均平方誤差計算，並透過 `mtp_alpha` 參數調節兩者的權重平衡。針對大規模數據的優化，本實驗採用了分布式資料並行 (Distributed Data Parallel, DDP) 技術，並根據需求選用 AdamW 或 SGD 優化器。在 `main.py` 的訓練迴圈中，模型會在每一輪結束後於驗證集上計算各項指標，包含路徑點的平均位移誤差。為了確保 GRU 模組能捕捉長距離的運動趨勢，程式中設計了 `optimize_per_n_step` 機制，在連續處理 40 個時間步長後才進行梯度回傳與隱藏層狀態的截斷重設，這有效緩解了長序列訓練時的梯度消失問題。模型部署與驗證階段，系統會加載訓練完成的權重進行批量推論。推論階段，系統讀取連續兩幀影像作為輸入，模型預測出最具可能性的未來路徑點。隨後，這些相對座標點會被轉換回鳥瞰圖慣用座標系 (如將 x 與 y 軸依約定進行投影變換)，並結合 `demo.py` 中的視覺化工具，將預測軌跡、路網地圖與車體姿態同步渲染於 BEV 視圖中。這種端到端的訓練與驗證流程，確保了模型從原始感知輸入到最終規劃決策的一致性與物理合理性。

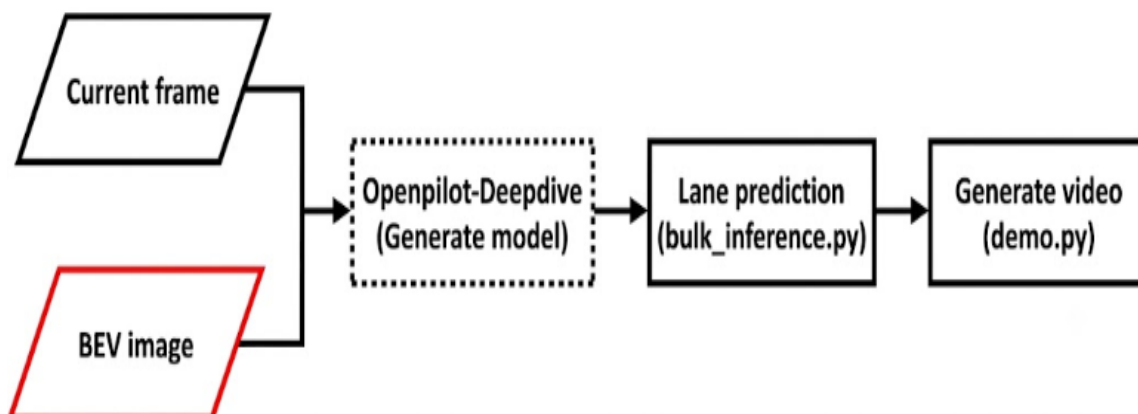
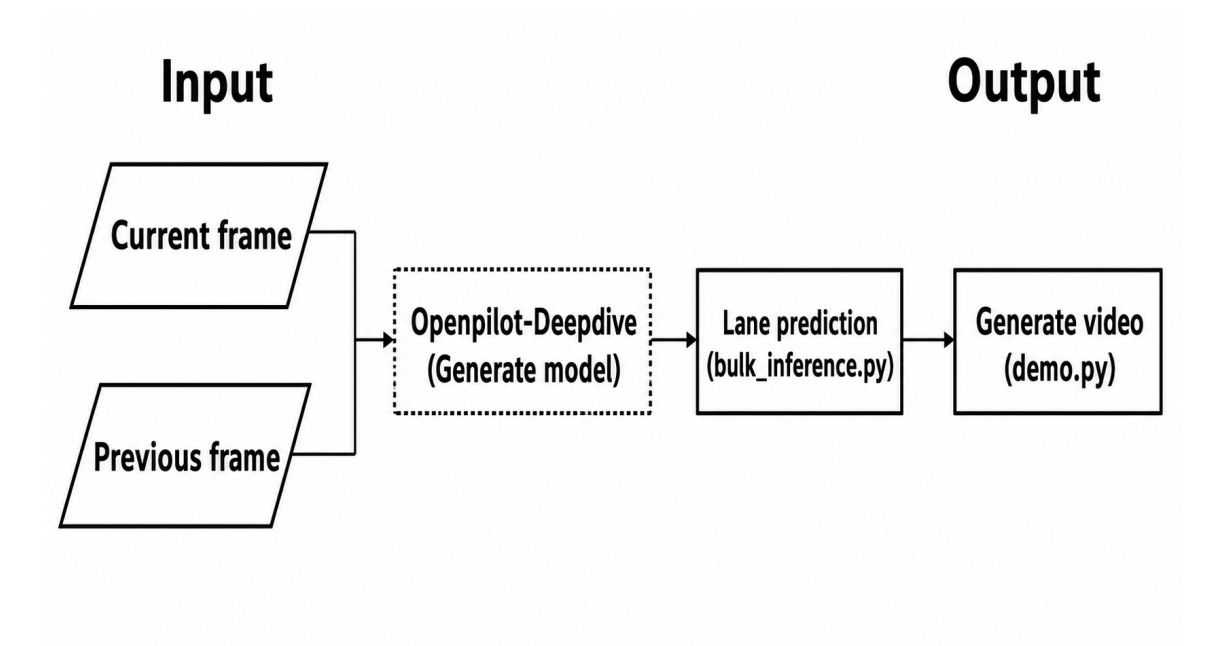


圖 3.5: 在上圖的配置中，系統採用傳統的時間序列輸入，將當前影格與前一影格同時餵入生成的模型中。這種方式主要依賴前後影像的視覺特徵變化，來讓模型理解車輛的運動與環境動態。接著，模型會透過腳本執行車道線預測，最後將結果轉化為視覺化的影片輸出。在下圖的改進流程中，系統將輸入源進行了替換。當前影格依然保留，但原本的前一影格被更具空間資訊的 BEV 所取代。

第四章 結果與討論

4.1 評估指標

本章針對所提出方法進行實驗驗證與結果分析，以評估導入導航路徑 BEV 影像後對 OP-DeepDive 軌跡預測模型之影響。首先介紹實驗環境與評估方式，並以 nuScenes 資料集進行模型訓練與測試。接著比較原始 OP-DeepDive 與導入導航路徑 BEV 影像之 OP-DeepDive 在不同評估指標下的預測表現，包括平均歐幾里得誤差及多層次命中率。最後，透過實驗結果分析不同輸入架構對模型預測能力與路徑生成效能之影響，並探討所提出方法於複雜城市道路場景中的優勢與限制，作為後續研究與模型改進之參考。

4.1.1 平均歐幾里得誤差 (Mean Euclidean Error, MEE)

平均歐幾里得誤差 (Mean Euclidean Error, MEE) 為衡量預測值與真實值之間空間差異之常用評估指標，廣泛應用於軌跡預測、目標定位及自駕車路徑生成等任務。其核心概念在於計算每一筆預測座標與對應真實座標之間的歐幾里得距離，並對所有樣本取平均，以反映模型在空間位置預測上的整體偏差程度。

設共有 N 筆樣本，第 i 筆樣本之預測座標為 $(\hat{x}_i, \hat{y}_i)$ ，對應之真實座標為 (x_i, y_i) ，則平均歐幾里得誤差可定義為

$$\text{MEE} = \frac{1}{N} \sum_{i=1}^N \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2} \quad (4.1)$$

對所有樣本取平均得到整體誤差值。應用於三維空間時，可進一步擴展為

$$\text{MEE} = \frac{1}{N} \sum_{i=1}^N \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 + (\hat{z}_i - z_i)^2}. \quad (4.2)$$

平均歐幾里得誤差具有直觀之幾何意義，其數值可直接對應於實際空間距離（如公尺），因此相較於均方誤差（Mean Squared Error, MSE）更易於解釋模型在實際應用場景中的預測偏差。然而，該指標對資料尺度具敏感性，故於不同資料集或座標系統間進行比較時，需確保單位一致。此外，平均歐幾里得誤差也常被用在時間序列軌跡評估中，例如計算整條路徑的平均偏移量，這在自駕車路徑預測中特別重要。若進一步細分，還可以延伸出其他指標，例如最終位置誤差（Final Displacement Error, FDE），專門評估軌跡終點的偏差。

4.1.2 多層次命中率 (Multi-level Hit Rates)

多層次命中率是一種用於評估模型預測結果在不同誤差容忍尺度下表現之指標，廣泛應用於軌跡預測、定位估計與自駕車路徑規劃等任務中。相較於單一誤差指標（例如平均歐幾里得誤差），多層次命中率能更細緻地描述模型在「不同精度要求」下的穩定性與可靠性，因此在實務與學術研究中均具有重要意義。其基本概念在於：對每一個預測結果，首先計算其與真實標註之間的距離誤差，通常採用歐幾里得誤差 (Euclidean distance) 作為衡量標準。接著設定一組遞增的距離閾值集合，例如 $r_1, r_2, \dots, r_K$ ，分別代表不同的容忍誤差範圍（如 1 公尺、2 公尺、5 公尺等）。若某一筆預測的誤差小於等於某個閾值，則視為該層級下的命中。

設共有 N 筆樣本，第 i 筆樣本之預測位置為 $(\hat{x}_i, \hat{y}_i)$ ，真實位置為 (x_i, y_i) ，則其誤差定義為：

$$d_i = \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2} \quad (4.3)$$

在此基礎上，對於任一距離閾值 r_k ，其命中率（Hit Rate）定義為：

$$HR(r_k) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(d_i \leq r_k) \quad (4.4)$$

其中， $\mathbf{1}(\cdot)$ 為指示函數（indicator function），當條件成立時取值為 1，否則為 0。此定義表示在所有樣本中，誤差小於等於閾值 r_k 的比例，即為該層級的命中率。

進一步地，若考慮多個閾值集合 $\{r_1, r_2, \dots, r_K\}$ ，則可得到一組多層次命中率：

$$\{HR(r_1), HR(r_2), \dots, HR(r_K)\}$$

此結果可視為一條「誤差容忍曲線」，用以描述模型在不同精度需求下的表現分佈。例如，在低閾值（如 1m）下的命中率反映模型的高精度能力，而在較高閾值（如 5m 或 10m）下的命中率則反映模型的整體穩定性與可接受範圍內的預測能力。相較於僅使用平均誤差的評估方式，多層次命中率具有以下優勢：首先，它能避免平均值對離群值過度敏感的問題；其次，它能同時呈現模型在不同精度需求下的完整性能輪廓；最後，它特別適合用於安全關鍵系統（如自駕車或機器人導航），因為這些應用通常對「是否落在可接受誤差範圍內」比平均誤差更為敏感。總結而言，多層次命中率提供了一種由粗到細的分層評估方式，使研究者能更全面理解模型在不同空間精度要求下的行為特性，並作為補充平均誤差類指標的重要評估工具。

4.2 模型 openpilot 訓練參數及 BEV 影像參數

本實驗 OP-Deepdive 模型訓練中，超參數的設定直接影響了模型的收斂速度、記憶體消耗與最終的泛化能力。訓練總共規劃了 1000 個 Epoch，整個訓練資料集將會完整通過神經網絡進行 1000 次的前向傳播與反向傳播。在批次大小方面設定為 6，模型每次更新權重前只會讀取 6 筆訓練樣本，較小的批次大小雖然會使梯度估計產生較大的隨機波動，但這種波動有時能幫助模型跳出局部最佳解，同時也能大幅降低對硬體顯示記憶體的即時需求。為了引導模型在錯誤表面上穩定下滑，學習率設定為 5×10^{-4} ，能有效避免訓練初期梯度爆炸或權重更新幅度過大而導致訓練發散。在資料載入與預處理的效率優化上，設置了 4 個 N workers，系統會啟動 4 個並行線程來加速資料的讀取與傳輸，確保硬體運算時不會因為等待資料而出現閒置。優化器則採用了經典的隨機梯度下降法，透過計算每個批次的梯度來更新參數，雖然路徑較為曲折，但在許多視覺與深度學習任務中被證實具有良好的泛化效能。在損失函數或多任務學習的權重平衡上，Mtp alpha 設定為 1，用以平衡不同損失項之間的比例，保持各任務在訓練過程中的影響力均等。本訓練採用了梯度累積的策略，將 Optimize per n step 設定為 40，模型在讀取資料時，並不會在每個批次結束後立刻更新權重，而是會連續累積 40 個步驟（即 $6 \times 40 = 240$ 筆樣本）的梯度後，才進行一次實質的參數優化，這種做法能在有限的硬體資源下，模擬出大批次訓練的穩定梯度效果。所有的訓練運算皆在 NVIDIA RTX A2000 GPU 硬體環境下執行，完整訓練超參數如表 4.1 所示：

表 4.1: 模型 OP-Deepdive 訓練超參數設定

Parameter	Value
Epoch	1000
Batch size	6
Learning rate	5×10^{-4}
N workers	4
Optimizer	SGD
MTP alpha	1
Optimize per n step	40
Hardware	NVIDIA RTX A2000 GPU

本實驗之鳥瞰圖影像與軌跡規劃超參數設定主要分為「影像幾何配置」與「優化權重損失」兩大部分。在影像幾何配置方面，輸出影像寬度設定為 800 像素，高度設定為 600 像素，用以定義 BEV 地圖的解析度大小。空間比例尺設定為 10，表示現實世界中每 1 公尺距離會對應至影像中的 10 個像素，因此 BEV 影像可涵蓋約 80 公尺 $\times$ 60 公尺的實際範圍。此外，線寬設定為 2 像素，用於繪製車道線、軌跡與道路邊界，以確保影像特徵具有清晰且一致的線條表示。在優化權重設定方面，各參數用於調整軌跡規劃過程中的損失函數比例。對齊權重設定為 6.0，用以控制預測軌跡與目標路徑之間的貼合程度；平滑權重設定為 60.0，為所有權重中最高的參數，代表模型更加重視軌跡連續性與行駛舒適度，以降低劇烈轉向或速度變化。速度權重設定為 25.0，用於約束車輛行駛速度，使其維持合理的速度分布。前向引導權重設定為 40.0，用以強化車輛向前行駛的趨勢，避免產生停滯或不合理的路徑規劃。一致性權重設定為 0.3，主要用於維持相鄰時間幀間預測結果的連續性。完整訓練超參數設定如表 4.2 所示：

表 4.2: 影像 BEV 超參數設定

Parameter	Value
Img width	800 (pixels)
Img height	600 (pixels)
Pixels per meter	10 (pixels)
Line width	2 (pixels)
Align weight	6.0
Smooth weight	60.0
Velocity weight	25.0
Forward weight	40.0
Consistency weight	0.3

4.3 模型結果比較

本實驗將比較四種模型的量化與質化的能力，首先進行模型的基本說明，接著比較量化表現，最後進行質化分析。



4.3.1 模型說明

本研究共構建並訓練了四種不同的模型變體，用以系統性地評估鳥瞰圖（BEV）資訊及其空間校正策略對模型效能的影響。各模型之架構與訓練設定說明如下：

- 原始模型（Original）：本模型採用 OP-Deepdive 的原始基準設定，未納入 BEV 資訊。其輸入採傳統的時間序列連續影像，透過將前一幀（Frame $t-1$ ）與當前幀（Frame t ）之影像資料同時輸入網絡進行訓練，作為後續實驗對比的基準線（Baseline）。
- 擾動模型（Perturbed）：此架構引入了包含全球定位系統（GPS）定位誤差的噪聲 BEV 資訊。在訓練過程中，原連續影像中的前一幀被替換為此帶有空間噪聲的 BEV 影像，用以評估未經校正的感知干擾對模型預測能力的負面效能影響。
- 修正模型（Corrected）：為了消除定位誤差帶來的干擾，本模型採用經幾何優化演算法將 GPS 噪聲對齊至地圖道路線後的 BEV 影像。訓練時同樣將前一幀影像

替換為該修正後的 BEV 資料，旨在探討二維空間特徵幾何校正後對模型定位與規劃的效益。

- 車輛視角模型 (Aligned)：此模型在修正模型的基礎上進一步實施了視角變換。其輸入之 BEV 影像不僅經過道路線幾何對齊，更透過仿射變換將其座標系由全局視角 (Global view) 轉換為與當前車輛一致的自我視角 (Ego-vehicle view)，並將其替換為前一幀輸入，用以驗證時空特徵一致性對模型特徵融合的優化效果。

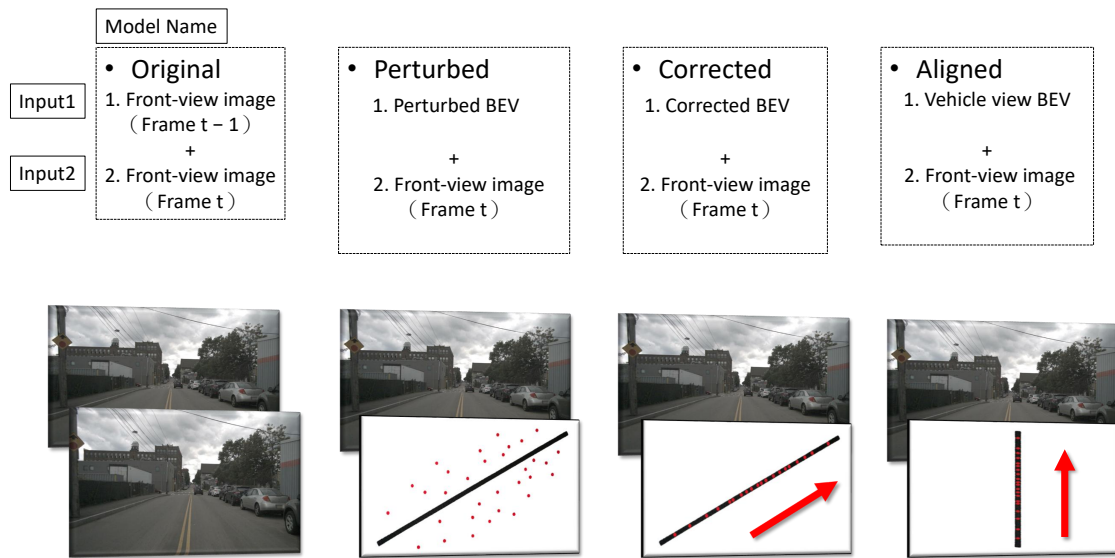


圖 4.1: 本研究提出的四種模型變體輸入資料配置示意圖。(a) Original：採用傳統連續車前影像作為基準輸入。(b) Perturbed：將前一幀替換為包含 GPS 定位誤差的噪聲 BEV 影像。(c) Corrected：將前一幀替換為經幾何優化、對齊至地圖道路線後的 BEV 影像。(d) Aligned：將前一幀替換為經道路線對齊並透過變換轉為車輛視角 (Ego-view) 的 BEV 影像。

4.3.2 量化比較

將 Original (原始)、Perturbed (受擾)、Corrected (修正) 與 Aligned (對齊真值) 這四種方法在不同場景下的消長與表現，進行分析與結論，圖 4.2 為 Hit rate 量化結果。

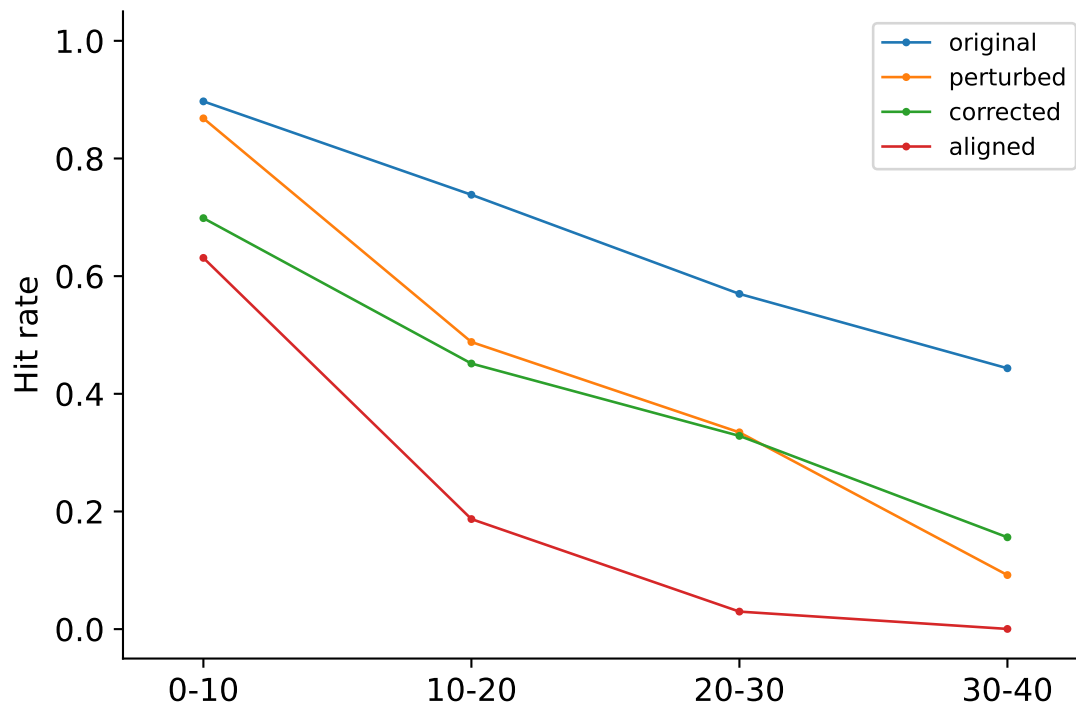


圖 4.2: 不同區間下各方法的命中率比較。圖中展示了 original、perturbed、corrected 與 aligned 四種方法在四個不同區間（0-10 至 30-40）下的命中率變化趨勢。

在 0-10 的初始區間中，四種方法分成了兩個領先群組：original 與 perturbed 的表現最為優異，命中率高達約 0.95 左右；而 corrected 與 aligned 則處於較低的水平，命中率大約在 0.81 附近。然而，進入 10-20 區間後，各方法的表現開始出現分歧。original 依然維持最優異的性能，命中率僅微幅下滑至約 0.82；perturbed 則出現明顯衰退，降至約 0.67；corrected 下滑至約 0.53；而 aligned 的衰退幅度最大，此時命中率已陡降至約 0.30。到了 20-30 區間，趨勢發生了有趣的交叉。original 雖然持續下滑，但仍以約 0.70 的命中率高居第一。值得注意的是，perturbed 在此區間的衰退速度加劇，命中率跌至約 0.41，這使得原本表現較差的 corrected（此時約為 0.43）在此處完成了黃金交叉，反超 perturbed 並位居第二。此時 aligned 的表現則持續探底，命中率已經逼近於零（約 0.04）。進入最後的 30-40 區間，所有方法的性能都降至各自的最低點。original 最終命中率下滑至約 0.58，在全階段皆保持最穩健且最優秀的預測能力。corrected 則展現了比 perturbed 更好的抗退化能力，最終維持在約 0.23。相比之下，perturbed 在後期潰不成軍，最終命中率僅剩約 0.13；而 aligned 則在該區間完全歸零（0.0），表現最為遜色。總結來說，此圖表證明了 original 在全規模區間下都具備壓倒性的最優準確度。

perturbed 雖然在初期與其並駕齊驅，但隨後便迅速惡化；corrected 儘管初始表現一般，但其抗衰退的斜率較為平緩，並在後期成功超越了受干擾的 perturbed 方法；而 aligned 則在各階段的準確度與耐受性表現上都顯著落後於其他三者。

圖 4.3 呈現了四種不同狀態（original、perturbed、corrected、aligned）下，歐氏誤差隨著某個分組區間增加而變化的趨勢。整體而言，所有狀態的歐氏誤差都隨著區間的遞增而呈現上升趨勢，且各組別在 0-10 區間的初始誤差都非常接近，均在 1.0 公尺以下，但隨著區間推進，彼此之間的差距大幅拉開。藍線代表的原始狀態（original）在所有區間中表現最佳，其誤差增長最為平緩，在 30-40 區間時，其歐氏誤差僅上升至約 4.0 公尺左右。橘線代表的受擾動狀態（perturbed）其誤差增長速度略高於原始狀態，到了最後的 30-40 區間時，誤差值達到了約 7.7 公尺。綠線代表的校正後狀態（corrected）誤差增長幅明顯大於前兩者，在 10-20 區間時與紅線接近，但隨後持續顯著上升，最終在 30-40 區間達到約 11.6 公尺。紅線代表的對齊狀態（aligned）在所有狀態中擁有最高的誤差與最陡峭的上升斜率，自 10-20 區間起便逐漸與其他曲線拉開距離，在最後的 30-40 區間，其歐氏誤差急遽飆升至大約 18.0 公尺。

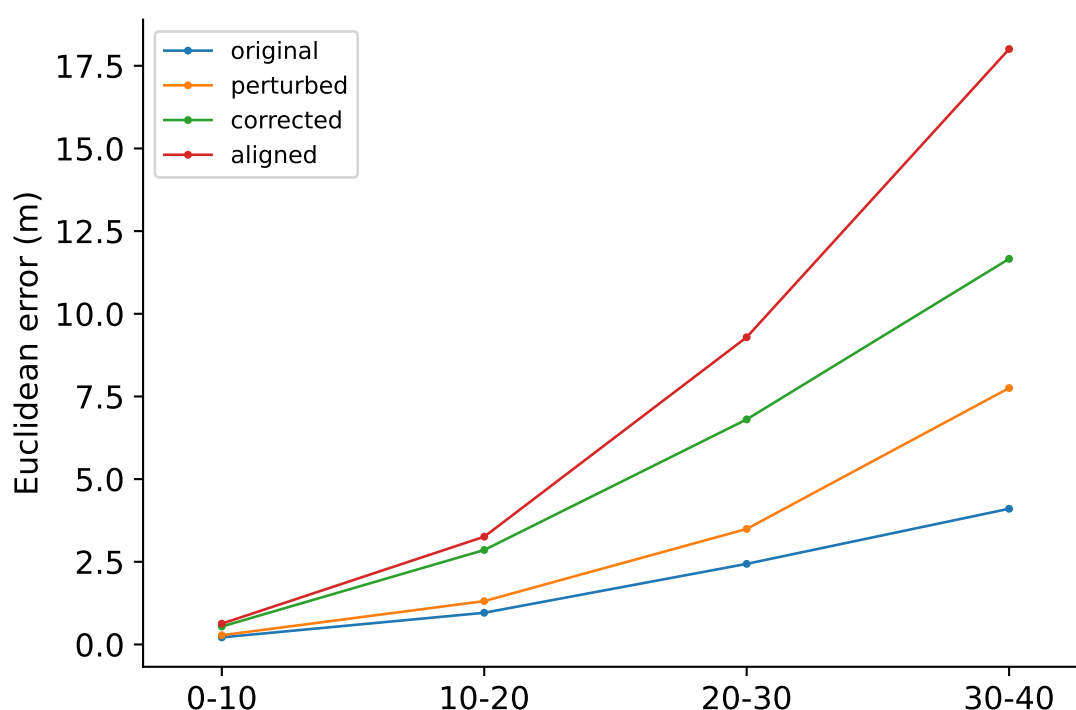


圖 4.3: 圖中展示了 original、perturbed、corrected 與 aligned 四種方法在四個區間（0-10 至 30-40）下的歐氏誤差變化趨勢。

理論上，Aligned 方法透過將空間座標系動態對齊至車輛本體座標系，使預測軌跡與車輛航向方向保持一致，應能降低模型對空間轉換的學習負擔，進而提升預測效能。然而，實驗結果顯示其表現並未如預期優於其他方法。本研究推測，主要原因可能來自 GPS 雜訊於座標轉換過程中的累積影響。當車輛執行轉彎等高動態駕駛行為時，GPS 雜訊可能造成車輛位置估計產生瞬時偏移。即使經過後端校正與濾波處理，此類誤差仍可能在座標系轉換後轉化為縱向位置震盪與航向角變化，使連續幀之間產生非預期的空間旋轉與位移。對於依賴時間序列資訊的深度學習模型而言，此類不穩定特徵會增加輸入資料的複雜度，降低模型學習軌跡運動規律與駕駛行為先驗的能力，進而造成實際效能與理論預期存在差異。此外，在 30-40 公尺的遠距離預測區間中，由於未來駕駛行為具有較高的不確定性，預測誤差會隨時間累積，使 corrected 方法的改善幅度逐漸降低，最終與 original 方法接近。然而，Aligned 方法在遠距離區間仍維持較穩定的表現，顯示其具備較佳的理想化空間對齊能力。綜合而言，corrected 方法透過後端幾何校正有效改善中短距離預測中的軌跡偏移問題，提升模型於複雜環境下的穩定性；但若欲進一步提升長距離預測能力，仍需結合更完整的動態駕駛意圖建模，以降低長期預測中的不確定性。



4.3.3 擾動模型表現較佳之場景

場景 1059 (straight-right-night) 中，perturbed 方法於所有 Hit Rate threshold 下皆取得最佳表現，且具有最低 Euclidean Error，顯示適度的軌跡擾動能有效提升模型於夜間右轉場景中的泛化能力與誤差容忍性。相較之下，corrected 方法並未帶來預期改善，其 Hit Rate 甚至低於 original，推測原因可能與夜間場景中特徵不穩定、軌跡過度修正以及時序一致性破壞有關。此外，aligned 方法表現顯著下降，顯示幾何對齊程序可能導致座標系統偏移或軌跡方向失真，使預測結果與 ground truth trajectory 產生大幅偏差。

表 4.3: 夜間場景 1059 量化比較

模型	Original	Perturbed	corrected	aligned
hit rate@0.5	0.3739	0.4172	0.3170	0.0578
hit rate@1.0	0.4322	0.4962	0.3994	0.0988
hit rate@2.0	0.5278	0.5799	0.4802	0.1554
Euclidean Error	3.7000	2.8736	3.8061	9.3513

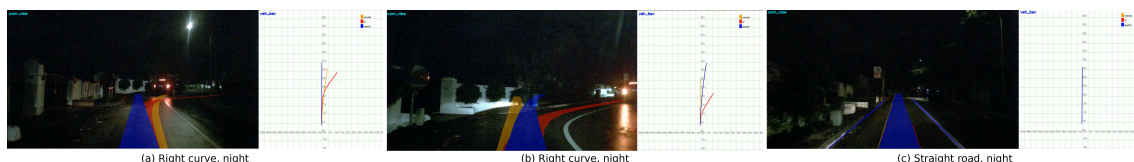


圖 4.4: 原始模型在夜間彎道場景中容易出現預測錯誤 (藍線偏離)，而 Perturbed 算法 (橘線) 可以有效地將路徑校正回與真實情況一致的方向。這證明了該修正機制在提升夜間行車安全性上的重要性。

場景 0557 (straight-lane-change) 中，perturbed 方法於所有 Hit Rate threshold 下皆取得最佳表現，並同時具有最低 Euclidean Error，顯示 trajectory perturbation 能有效提升模型對 lateral motion 與 lane transition 的建模能力。特別是在 HR@0.5 下，perturbed 較 original 提升約 9.8%，顯示其預測軌跡與 ground truth centerline 之間具有更高的幾何一致性。相較之下，corrected 方法並未帶來明顯改善，推測可能由於過度平滑化導致 lane-change trajectory 的橫向特徵被削弱。此外，aligned 方法表現顯著下降，顯示 rigid alignment 可能破壞 lane-change 過程中的方向與位移資訊，進而導致 trajectory prediction 失效。

表 4.4: 道路切換場景 0557 量化比較

模型	Original	Perturbed	corrected	aligned
hit rate@0.5	0.5745	0.6728	0.5508	0.1887
hit rate@1.0	0.6645	0.7291	0.6365	0.2552
hit rate@2.0	0.7038	0.7382	0.6870	0.3331
Euclidean Error	2.4904	2.2108	2.6134	6.7485

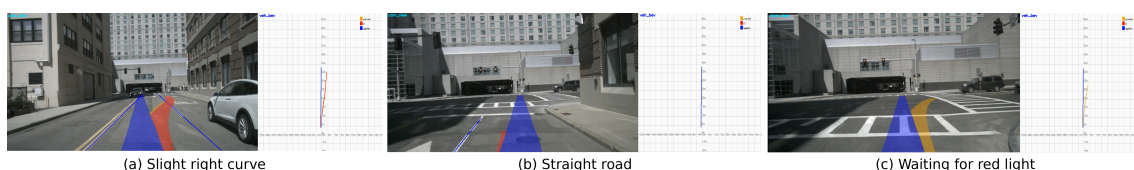


圖 4.5: 在圖 (a) 的微偏右彎道場景中，可以明顯觀察到 Perturbed (橘線) 的預測效果優於原始的 OpenPilot (藍線)，其預測方向與真實路徑更為貼近。然而，在圖 (c) 停等紅綠燈的場景下，預測效果則出現了顯著的惡化，無論是 OpenPilot (藍線) 還是 Perturbed (橘線)，在面對紅綠燈停等這種複雜的都市靜止場景時，其預測軌跡皆表現不佳，出現了嚴重的方向扭曲與偏差，未能反映情境下應有的正確行駛或靜止狀態。

4.3.4 修正模型表現較佳之場景

場景 0794 屬於 straight 類型，為典型低曲率（low-curvature）且幾何結構相對單純的軌跡場景。相較於 lane-change 或 intersection 類型場景，此類場景具有較小的 heading variation、較低的 lateral displacement，以及較穩定的 trajectory continuity，因此更容易觀察不同方法對軌跡幾何修正能力的影響。從量化結果來看，corrected 方法在此場景中表現最佳，不僅於所有 Hit Rate threshold 下皆達到近乎完美的命中率，同時也取得最低的 Euclidean Error，顯示 correction 對於低複雜度場景具有高度有效性。

表 4.5: 直線場景 0794 量化比較

模型	Original	Perturbed	corrected	aligned
hit rate@0.5	0.9979	0.9979	1.0000	0.9958
hit rate@1.0	0.9979	0.9979	1.0000	0.9979
hit rate@2.0	1.0000	1.0000	1.0000	1.0000
Euclidean Error	0.0268	0.0249	0.0214	0.0358

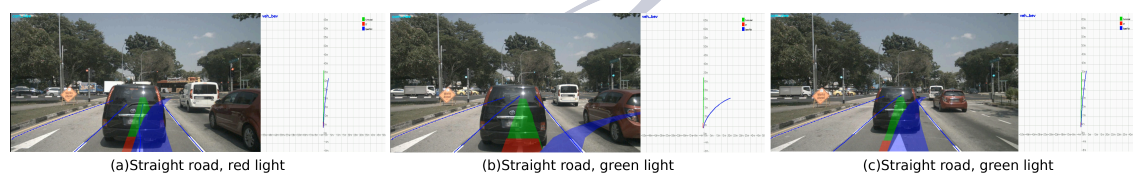


圖 4.6: 綠線方向與紅線始終保持一致，顯示出高度的準確性。相對而言，藍線 (Open-Pilot) 代表原始系統的預測，其方向明顯偏離了真實路徑，存在預測錯誤

場景 0093 屬於 intersection-right-straight 類型，該場景包含路口通過、右轉以及後續直行等複合軌跡特徵，因此相較於純直線場景，其軌跡幾何變化較大，對於模型的方向估計與軌跡連續性要求也更高。從量化結果來看，corrected 方法雖未在所有 Hit Rate threshold 下取得最佳表現，但在較寬鬆的評估條件與整體軌跡誤差上具有明顯優勢。具體而言，corrected 於 HR@2.0 達到 0.5833，優於 original 的 0.5521，同時 Euclidean Error 由 3.4428 降低至 3.1219，顯示 correction 過程能有效降低整體軌跡偏移，使預測軌跡在全域幾何位置上更接近 ground truth。

表 4.6: 場景 0093 (intersection-right-straight) 量化比較

模型	Original	Perturbed	corrected	aligned
hit rate@0.5	0.3854	0.3812	0.4038	0.1126
hit rate@1.0	0.4583	0.4479	0.3474	0.1693
hit rate@2.0	0.5521	0.5312	0.5833	0.2450
Euclidean Error	3.4428	3.5081	3.1219	7.4384

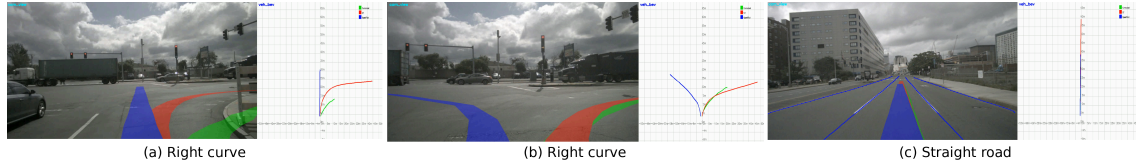


圖 4.7: 綠線 (Corrected) 的方向與紅線 (Ground Truth) 保持高度一致。這表明修正算法能夠有效地識別道路趨勢，並提供與實際路徑相符的預測結果，在彎道場景中，藍線未能準確跟隨紅線的轉向趨勢。

4.3.5 原始模型表現較佳之場景

原始基準模型 (Original) 在場景 0013 下展現了極高且穩定的預測精度，其平均歐氏誤差 (Euclidean Error) 僅有 0.2311 公尺，且在高容忍度門檻下的命中率 (hit rate@2.0) 達到了 0.9936，意味著將近百分之百的預測點都能精準控制在兩公尺的誤差範圍內。然而，當系統引入環境擾動 (Perturbed) 之後，模型的表現受到了顯著的負面衝擊，在嚴格門檻下的 hit rate@0.5 從 0.8674 大幅降至 0.5176，同時整體的平均歐氏誤差亦由 0.2311 驟升至 1.5199 公尺，這項數據變化深刻表明，在十字路口這種具備複雜幾何拓撲結構與潛在干擾源的場景中，基礎預測模型對於外在噪聲或軌跡擾動具備相當高的敏感性與不穩定性。校正機制 (corrected) 在命中率指標 (hit rate) 上展現出了一定的修復成效。相較於受擾狀態 (Perturbed)，校正後的 hit rate@0.5 成功提升至 0.5609，且在 hit rate@1.0 與 hit rate@2.0 上也分別進步至 0.6913 與 0.7521，這證明了該校正算法能有效修正部分偏離的軌跡，並將其重新拉回至合理的誤差門檻之內。然而，一項值得高度關注的異常現象在於其整體的平均歐氏誤差 (Euclidean Error) 反而顯著上升至 2.9690 公尺，此一矛盾的趨勢揭示了該校正機制的核心限制：雖然它成功優化了多數正常幀的近距離偏差 (進而拉高了整體的命中率)，但在某些特定幀中可能產生了較為嚴重的極端預測值或軌跡發散現象，這些離群值 (Outliers) 在計算平均歐氏誤差時放大了幾何距離，導致整體平均誤差不降反升。

相較之下，對齊模型（aligned）在場景 0013 中的表現則呈現嚴重的全面性退化。其平均歐氏誤差大幅攀升至 5.6573 公尺，且在最嚴格的預測門檻下，hit rate@0.5 僅剩餘 0.3457，其餘各階段的命中率亦均為所有評估模型中的最低值。這項結果指出，目前所採用的時空或特徵對齊策略，在「十字路口直行」這種幾何方向相對單一但高度依賴軌跡連續性的場景中，產生了顯著的負面效果，可能存在嚴重的過度校正、特徵扭曲或方向對齊失效的問題，反而破壞了原本已具備一定準確度的基礎軌跡。

表 4.7: 場景 0013 (intersection-straight) 量化比較

模型	Original	Perturbed	corrected	aligned
hit rate@0.5	0.8674	0.5176	0.5609	0.3457
hit rate@1.0	0.9661	0.6129	0.6913	0.4235
hit rate@2.0	0.9936	0.7247	0.7521	0.5097
Euclidean Error	0.2311	1.5199	2.9690	5.6573

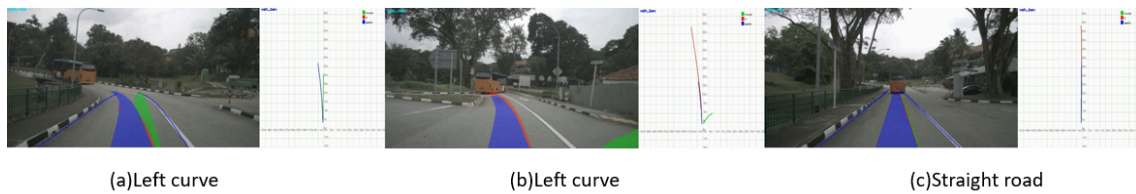


圖 4.8: 模型 Openpilot 路徑與 Ground Truth 具有較高的一致性，而 Corrected 路徑則出現明顯偏移，特別是在彎道路段，偏差情況更為顯著

4.4 對其模型表現不佳原因分析

在自駕車路徑預測任務中，為了提升軌跡表示與車輛運動方向之對齊性，常將原始 GPS 資訊投影至參考路線後建立以車輛視角為基準之 Aligned Bird's-Eye View (Aligned BEV) 表示。然而，當輸入之 GPS 訊號受到定位誤差、多路徑效應或感測雜訊影響時，經路徑匹配（map matching）或路線對齊後的位置估計仍可能存在不穩定現象。此問題於車輛通過轉彎區域時尤其明顯，因為彎道附近通常具有較高曲率與較複雜的幾何結構，使得帶有雜訊之定位點容易在轉彎前段與轉彎後段之路徑節點間反覆切換。當位置點於彎道區域產生頻繁跳動時，對於以車輛朝向作為座標基準之 Aligned BEV 表示將產生額外影響。由於 Aligned BEV 需根據當前車輛位置與朝向資訊對場景進行旋轉與重新對齊，因此位置估計的不穩定會直接導致 BEV 空間中的路徑表示於連續時間步間

產生非預期之旋轉與位移變化。換言之，即使實際車輛運動軌跡保持平滑，輸入至模型之路徑特徵仍可能因定位跳動而呈現高頻率的空間擾動。此類由定位誤差所導致之 BEV 表示不穩定性，將進一步破壞時間序列資料應具備之連續性與一致性。對於依賴時序資訊學習運動模式之模型而言，不連續的路徑表示容易造成特徵分布偏移 (feature distribution shift)，使模型難以有效建立車輛運動狀態與路徑幾何之間的對應關係。此外，持續性的旋轉擾動亦可能使模型誤將定位誤差學習為有效特徵，增加訓練過程中的優化難度與收斂不穩定性。因此，在使用 Aligned BEV 作為路徑表示方式時，降低 GPS 雜訊造成之位置跳動問題具有重要性。若未有效處理定位誤差對路徑對齊流程之影響，則可能導致輸入特徵缺乏穩定性，進而降低模型對轉彎場景之辨識能力與整體路徑預測效能。

Challenges of Vehicle-Centric Perception During Turning Scenarios

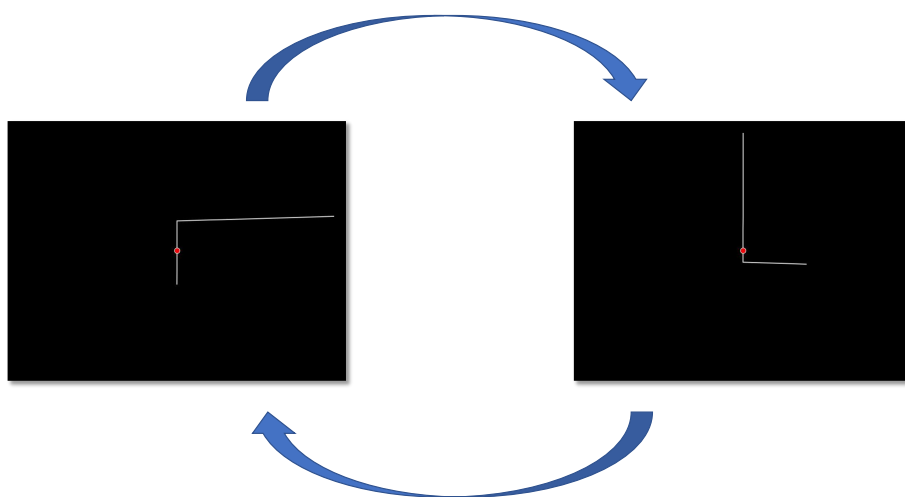


圖 4.9: 將含有雜訊之 GPS 座標對齊至參考路線後，於車輛轉彎場景中，由於定位誤差與路徑匹配不穩定，對齊後的位置點可能在轉彎前後區域產生頻繁跳動現象。Aligned BEV 表示方式係依據車輛視角對路徑進行旋轉與對齊，因此位置點的不穩定變化將導致整體路徑表示持續發生旋轉與位移。此現象會造成輸入特徵在時間序列上的一致性下降，模型辨識路徑幾何特徵之困難度，對模型訓練穩定性與預測效能產生負面影響。

4.5 場景模型表現統計表

表 4.8: Overall Model Performance Comparison

Model / Method	Hit Rate @ 1.0m (↑)	Euclidean Error (m) (↓)	Error (0–10m)	Error (30–40m)
original	77.78%	1.56 m	0.20 m	3.81 m
perturbed	57.79%	2.62 m	0.27 m	7.24 m
corrected	52.65%	4.59 m	0.53 m	10.92 m
aligned	32.57%	6.46 m	0.61 m	16.90 m

表 4.9: Per-Scenario Model Performance Comparison (Euclidean Error in meters)

Category	Scenario Class	original	corrected	perturbed	aligned
Intersection	intersection-straight	0.36	5.36	1.67	5.87
	intersection-straight-left	0.18	1.47	2.56	9.17
	intersection-straight-lane_change	0.65	6.79	1.39	5.79
	intersection-right-straight	2.26	1.92	3.58	5.47
	intersection-right	4.01	5.00	5.45	7.06
	intersection-left	3.68	8.29	4.52	7.49
	intersection-left-stop	1.18	8.56	2.89	4.41
	intersection-lane_change-left	1.84	3.40	2.41	7.35
Straight & Maneuvers	straight	0.18	2.25	1.07	3.79
	straight-lane_change	0.76	1.39	0.86	3.60
	straight-left	0.52	2.42	1.54	3.92
	straight-right	6.26	3.97	8.06	7.51
	straight-left-right-left	0.81	2.99	1.96	6.56
	straight-lane_change-right-lane_change	0.53	3.79	1.80	6.82
Night	intersection-left-night	3.31	3.54	3.32	8.00
	intersection-straight-night	0.14	0.32	0.58	10.32
	straight-right-night	2.12	2.03	2.45	4.42
	straight-right-right-night	1.22	1.68	2.25	7.64

第五章 結論

5.1 結論

本研究針對端到端自動駕駛框架 OP-Deepdive 進行優化，提升了其在大型公開自動駕駛資料集 nuScenes 上的表現。本研究的核心創新在於打破傳統單一透視視覺輸入的限制，創新性地將基於 OpenStreetMap 導航路徑資訊所繪製的鳥瞰圖影像引入模型輸入端。實驗與對比結果證實，透過多源資訊的融合，模型在路徑生成任務上的預測精度、軌跡平滑度以及對複雜路網的適應力均獲得顯著改善。此成果不僅驗證了導航先驗資訊在端到端規劃中的關鍵作用，開創了具備高可行性與前瞻性的研究路徑。

5.2 未來展望

綜觀本研究之成果與局限，未來之研究深化將朝向「卡爾曼濾波器」與「連續 2 張圖片與 BEV 訓練」兩方向開展。首先，針對車輛於轉彎路段時，定位點在鄰近路段間易出現軌跡跳動與不連續之瓶頸，後續研究將導入卡爾曼濾波器。藉由引入車輛動力學先驗知識，對定位序列進行狀態估測與雜訊過濾，以期在不犧牲即時性的前提下，顯著提升 aligned 軌跡之平滑度與魯棒性。其次，鑑於現有模型架構對於連續雙幀影像之時間序列依賴性，未來將深入評估不同幀數輸入對預測精度之敏感度。在此基礎上，本研究亦計畫引入 BEV，將前視角影像轉化為具備拓撲一致性的空間特徵，並與原有的雙幀時序資訊進行深度融合。預期此種架構創新，能有效改善物件在轉彎或遮蔽場景下的幾何表徵能力，進而大幅提升整體模型的泛化性能。

參 考 文 獻

- [1] Eder Santana and George Hotz. Learning a driving simulator, 2016.
- [2] Harald Schafer, Eder Santana, Andrew Haden, and Riccardo Biasini. A commute in data: The comma2k19 dataset, 2018.
- [3] Mitchell Goff, Greg Hogan, George Hotz, Armand du Parc Locmaria, Kacper Raczy, Harald Schäfer, Adeeb Shihadeh, Weixing Zhang, and Yassine Yousfi. Learning to drive from a world model, 2025.
- [4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020.
- [5] Li Chen, Tutian Tang, Zhitian Cai, Yang Li, Penghao Wu, Hongyang Li, Jianping Shi, Junchi Yan, and Yu Qiao. Level 2 autonomous driving on a single device: Diving into the devils of openpilot, 2022.
- [6] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks, 2015.
- [7] Felipe Codevilla, Antonio M. López, Vladlen Koltun, and Alexey Dosovitskiy. On offline evaluation of vision-based driving models, 2018.
- [8] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst, 2018.

- [9] Comma.ai. Openpilot: Open source adas. <https://github.com/commaai/openpilot>, 2020.
- [10] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseem Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016.
- [11] Yuhang Wang, Abdulaziz Alhuraish, Shengming Yuan, Shuyi Wang, and Hao Zhou. Openlka: an open dataset of lane keeping assist from market autonomous vehicles, 2025.
- [12] Dean A. Pomerleau. Alvin: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems*, pages 305–313, 1989.
- [13] comma.ai. Announcing the EON Dashcam DevKit. <https://blog.comma.ai/announcing-the-eon-dashcam-devkit/>, 2017.
- [14] Baidu Apollo. Apollo autonomous driving platform. <https://github.com/ApolloAuto/apollo>, 2023.
- [15] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017.
- [16] Epic Games. Unreal engine 4. https://en.wikipedia.org/wiki/Unreal_Engine_4, 2014. Accessed: 2026-02-26.
- [17] Michael Montemerlo, Jan Becker, Suraj Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.
- [18] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning, 2018.
- [19] Jelena Kocić, Nenad Jovičić, and Vujo Drndarević. An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms. *Sensors*, 2019.

- [20] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *arXiv preprint*, 2020.
- [21] Apoorv Singh. End-to-end autonomous driving using deep learning: A systematic review. *arXiv*, 2023.
- [22] Pranav Singh Chib and Pravendra Singh. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *arXiv*, 2023.
- [23] Pei Wang, Mustafa Sevil, and Chris Urmson. Safe end-to-end autonomous driving enhanced by uncertainty estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [24] Rahul Mehta, Anush Srivastava, Ahmed Abdel-Aziz, Jaeheung Hwang, and Anthony Kim. Towards interpretable end-to-end models for self-driving vehicles. *IEEE Transactions on Intelligent Vehicles*, 5(4):700–710, 2020.
- [25] Paul D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.
- [26] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers, 2022.
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.
- [28] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d, 2020.
- [29] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view, 2022.

- [30] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries, 2021.
- [31] Ziang Guo, Konstantin Gubernatorov, Selamatwit Asfaw, Zakhar Yagudin, and Dzmitry Tsetserukou. Vdt-auto: End-to-end autonomous driving with vlm-guided diffusion transformers, 2025.
- [32] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning, 2022.
- [33] Zhe Ming Chng, Joseph Mun Hung Lew, and Jimmy Addison Lee. Ronelddv2: A faster, improved lane tracking method, 2022.
- [34] Lucas Tabelini, Rodrigo Berriel, Thiago M. Paixão, Claudine Badue, Alberto F. De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression, 2020.
- [35] Markus Schreiber, Carsten Knöppel, and Uwe Franke. Laneloc: Lane marking based localization using highly accurate maps. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 449–454, 2013.
- [36] Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Buló, Richard Newcombe, Peter Kotschieder, and Vasileios Balntas. Orienternet: Visual localization in 2d public maps with neural matching, 2023.