
This is the **published version** of the bachelor thesis:

Berrouhou Barrouhou, Oussama. *Predicción no supervisada de trayectorias mediante aprendizaje por imitación*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334395>

under the terms of the  license.

Predicción no supervisada de trayectorias mediante aprendizaje por imitación

Oussama Berrouhou

30 de juny de 2026

Resumen– Este proyecto tiene como objetivo desarrollar un sistema para la conducción autónoma compuesto por varios modelos basados en diferentes parámetros, utilizando técnicas de visión por computador y redes neuronales profundas. La motivación principal reside en que los accidentes de tráfico siguen provocando más de un millón de muertes anuales a nivel global, y las soluciones actuales dependen en gran medida de sensores costosos y grandes volúmenes de datos anotados, lo que limita su accesibilidad y escalabilidad. El sistema propuesto es capaz de predecir la trayectoria que debe seguir el vehículo mediante la generación de waypoints, obtenidos a partir de imágenes extraídas de un vídeo capturado por tres cámaras frontales incorporadas en el ego-vehículo. Para ello, en este trabajo se introducen los distintos enfoques que intervienen en el proceso de pseudoanotado de datos y estimaciones cinemáticas durante el entrenamiento, siendo la profundidad uno de los factores clave para la comprensión del entorno. A partir de este estudio se concluye que, a medida que se eliminan modalidades de Ground Truth, el rendimiento disminuye progresivamente; no obstante, la metodología basada en el modelo cinemático presenta un futuro prometedor, mientras que la pseudoanotación mediante Structure-from-Motion requiere trabajo adicional para mejorar la precisión de la predicción en lazo cerrado.

Palabras clave– conducción autónoma, visión por computador, redes neuronales profundas, waypoints, profundidad, seguridad vial.

Abstract– This project aims to develop a system for autonomous driving composed of several models based on different parameters, using computer vision techniques and deep neural networks. The main motivation lies in the fact that traffic accidents still cause over one million deaths annually worldwide, and current solutions rely heavily on expensive sensors and large volumes of annotated data, limiting their accessibility and scalability. The proposed system is capable of predicting the trajectory that the vehicle should follow through the generation of waypoints, obtained from images extracted from a video captured by three front cameras integrated into the ego-vehicle. To this end, this work introduces the various approaches involved in the data pseudo-labeling process and kinematic estimations during training, with depth being a key factor for understanding the environment. Based on this study, it can be concluded that as Ground Truth modalities are progressively removed, performance decreases accordingly; however, the kinematic model-based methodology shows promising potential, while pseudo-annotation through Structure-from-Motion requires further work to improve closed-loop prediction accuracy.

Keywords: Autonomous driving, computer vision, deep neural networks, trajectory prediction, waypoints, depth, road safety.

1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

LA conducción autónoma es un área en constante desarrollo dentro del campo de la inteligencia artificial, impulsada por los avances en visión por computador, sensores y toma de decisiones automática. Su impacto no se limita únicamente a la innovación tecnológica, sino que también representa un cambio profundo en la movilidad y la seguridad vial del futuro. Los vehículos

- E-mail de contacto: 1711619@uab.cat
- Menció: Computació
- Trabajo tutorizado por: Gabriel Villalonga Pineda (CVC)
- Curso 2025/26

autónomos buscan reducir los accidentes, mejorar la eficiencia del tráfico y ofrecer una conducción más segura y accesible para todos los usuarios. Esta necesidad se hace evidente al considerar que, según los últimos informes de la Organización Mundial de la Salud (OMS), los accidentes de tráfico siguen provocando aproximadamente 1,19 millones de muertes anuales a nivel global [1].

Sin embargo, el desarrollo de estos sistemas presenta aún un sinfín de retos técnicos y prácticos. Las soluciones actuales dependen en gran medida de grandes volúmenes de datos etiquetados con Ground Truth (GT), de sistemas de cámaras complejos y de procesos de calibración costosos, lo que limita su accesibilidad y escalabilidad. Además se suman las exigencias de procesamiento en tiempo real y de muy baja latencia, que restringen y limitan la implantación de estos sistemas en entornos cotidianos o en vehículos de bajo coste.

A pesar de los avances recientes que han permitido superar algunas de estas limitaciones, persisten problemas importantes relacionados con la fiabilidad y la adaptabilidad de los sistemas de navegación autónoma. Las soluciones disponibles no siempre logran equilibrar adecuadamente la precisión, la eficiencia y la facilidad de implementación, lo que dificulta su aplicación generalizada por parte de conductores o desarrolladores fuera de la investigación avanzada.



Fig. 1: Ejemplo de intersección urbana próximo al CVC-UAB real similar a los escenarios considerados en este trabajo.

Dentro del campo de la planificación de trayectorias, existen distintos enfoques: algunos se basan en el control directo de las acciones del vehículo, mientras que otros se centran en la generación de puntos de referencia o waypoints. Este último enfoque ofrece una mayor flexibilidad para adaptarse a diferentes escenarios y condiciones de conducción, permitiendo una planificación más modular y escalable.

En este contexto, el presente Trabajo de Fin de Grado tiene como objetivo abordar los desafíos mencionados mediante el diseño de un sistema de navegación autónoma basado en waypoints, que busque reducir la dependencia de infraestructuras costosas y mejorar la accesibilidad de la conducción autónoma. El propósito final es avanzar hacia soluciones más eficientes y asequibles que contribuyan al desarrollo de una movilidad inteligente y segura.

En entornos urbanos como el de la Fig. 1, el sistema debe

ser capaz de percibir correctamente la geometría de la escena, anticipar la trayectoria y tomar decisiones seguras en intersecciones y cambios de carril.

2 OBJETIVOS

Este trabajo aborda la conducción autónoma con el objetivo de realizar un estudio de la implementación de diferentes modelos basados en waypoints para resolver la predicción de trayectorias en un entorno no supervisado. Se plantea investigar la eficacia de estos modelos sin la necesidad de datos de referencia (Ground Truth) y sin que la información provenga de señales GPS, utilizando únicamente datos visuales capturados por cámara.

A partir de este objetivo general, se definen los siguientes objetivos específicos:

- Implementar un modelo basado en la cinemática del vehículo (Kinematic Bicycle Model) que, a partir de las imágenes del entorno, genere waypoints físicamente viables y coherentes con la dinámica de conducción, diseñando funciones de coste que integren *losses* cinemáticos para optimizar el entrenamiento y asegurar la suavidad y viabilidad de las trayectorias predichas sin depender de etiquetas manuales.
- Analizar de forma cuantitativa y cualitativa los resultados obtenidos, comparando el rendimiento del sistema propuesto con otros enfoques que dependen de datos supervisados, con el fin de evaluar su eficacia, escalabilidad y potencial de aplicación en entornos reales.

3 ESTADO DEL ARTE

Enfoques basados en acciones y waypoints

En los sistemas de conducción autónoma se han explorado distintos enfoques para la planificación y el control del movimiento. Entre ellos, los más destacados son los métodos basados en *acciones* y los métodos basados en *waypoints* [2, 3].

Los enfoques basados en acciones generan comandos de control directamente a partir de una observación visual, típicamente mediante una o varias cámaras monoculares dispuestas con un ángulo determinado para aproximar el *horizontal field of view* (HFOV)[4, 5]. Mediante estos *inputs*, el modelo ajusta de forma directa parámetros como la velocidad, la dirección y el ángulo de giro del *ego-vehículo*, permitiendo reaccionar ante la situación actual de manera inmediata. Este enfoque permite una toma de decisiones rápida, aunque presenta menor interpretabilidad y estabilidad a largo plazo en el trayecto. Trabajos pioneros desde el clásico ALVINN [6], hasta enfoques modernos como el de Bojarski *et al.* [7], demostraron la viabilidad de este tipo de control directo a partir de imágenes frontales.

Dentro de esta familia de control directo destaca especialmente la conducción por imitación, donde la red aprende a copiar el comportamiento de un conductor humano. La

idea de fondo es sencilla: si tenemos grabadas las acciones (aceleración y *steering*) que un conductor experto tomó en cada situación, podemos usarlas como *Ground Truth* para que la red aprenda a reproducir esas mismas decisiones a partir de las imágenes. El ejemplo más representativo de esta línea es la familia CILRS [21] y su evolución CIL++ [22]. Ambos comparten la misma raíz de imitación, pero CIL++ mejora la predicción de aceleración y *steering* a partir de tres cámaras frontales (cuyos ángulos, normalmente entre 180° y 220°, simulan el campo de visión humano o HFOV), apoyándose únicamente en imágenes RGB. Internamente, no es más que una CNN (*Convolutional Neural Network*) que actúa de *encoder* extrayendo un vector de características de la imagen, seguida de una capa de regresión que predice directamente los comandos del coche, sin mapas, sin LiDAR ni proyecciones geométricas. Por su simplicidad y por ser un trabajo fundacional muy conocido, este tipo de modelo se ha convertido en una referencia habitual a la hora de comparar nuevas arquitecturas de conducción *end-to-end*.

Por otro lado, los enfoques basados en *waypoints* dividen el problema en dos pasos: primero se predice una secuencia de puntos futuros consecutivos que representan la trayectoria deseada del vehículo; posteriormente, un controlador se encarga de seguir dichos puntos. Este enfoque introduce una mayor flexibilidad, ya que permite modificar fácilmente la forma en la que se generan o se siguen los *waypoints*. Además, al trabajar con puntos intermedios en lugar de órdenes directas, es más sencillo entender qué pretende hacer el modelo y detectar posibles errores. Ejemplos representativos de esta línea de trabajo son los de Codevilla *et al.* [8] y Chitta *et al.* [9], quienes integran la predicción de *waypoints* con módulos de percepción y planificación de alto nivel.

Percepción 3D y reconstrucción del entorno

En cuanto a la percepción y reconstrucción del entorno, se han desarrollado métodos que permiten recoger información tridimensional sin necesidad de GPS. Una técnica clásica es *Structure-from-Motion* (SfM), que reconstruye una escena tridimensional a partir de una secuencia consecutiva de fotogramas captados mediante una cámara monocular, generando una nube de puntos 3D. Este método ha sido ampliamente estudiado en la literatura [10, 11] y constituye la base de muchas soluciones actuales de reconstrucción geométrica.

Otra aproximación es la estimación de profundidad monocular, donde modelos auto-supervisados como *Mono-depth* [12] o *SfMLearner* [13] aprenden a predecir mapas de profundidad a partir de secuencias de imágenes sin supervisión directa. En particular, *Monodepth* emplea un sistema de cámaras estéreo que imita la visión humana y obtiene un mapa de disparidad mediante una función de pérdida que se optimiza iterativamente para minimizar el error de reconstrucción.

Estos métodos ofrecen opciones alternas y prometedoras para generar información 3D útil en tareas de navegación y planificación de trayectorias sin depender de datos *Ground Truth*, permitiendo avanzar hacia sistemas más autónomos y eficientes. Investigaciones recientes como las de Guizili-

ni *et al.* [14], el conjunto de datos KITTI [15], o el uso de simuladores de código abierto [16], han contribuido significativamente a la evaluación y mejora de estos sistemas en entornos reales.

Estrategias de control y ejecución de trayectorias

Para hacer realidad el seguimiento de las trayectorias planificadas, existen enfoques en el control de vehículos que integran modelos físicos, como el modelo cinemático de bicicleta [17]. Esta aproximación principalmente matemática garantiza que las trayectorias generadas sean viables y respeten las restricciones dinámicas reales del automóvil a partir de la distancia entre los ejes del vehículo; su formulación completa se recoge en el **Anexo B**. En la práctica, este modelo sirve de enlace natural entre las acciones que predice el sistema y el movimiento real del vehículo, evitando que se realicen maniobras extrañas o físicamente imposibles.

Por otro lado, y siguiendo la línea de las tendencias de aprendizaje profundo aplicado a la conducción autónoma, otras arquitecturas recientes exploran el uso de redes neuronales completamente conectadas, como el Perceptrón Multicapa (MLP), para el módulo de toma de decisiones [18]. En este tipo de enfoques *End-to-End*, el MLP tiene el objetivo de procesar directamente los vectores de características extraídos por los módulos de percepción visual (transformers) [19]. Al hacerlo, el modelo es capaz de reforzar o sustituir las heurísticas de control tradicionales, estableciendo una estrategia de conducción continua y robusta que traduce directamente la información visual en comandos de navegación.

Basado en todos estos trabajos, en este trabajo se trabajará con el END2END con acciones, apoyándonos en los modelos cinemáticos como el *Kinematic Bicycle Model* para estimar las trayectorias a partir del control y el SfM para evitar la anotación y uso de *waypoints*.

4 PLANIFICACIÓN

La planificación detallada de este proyecto, incluyendo la definición de los *Workpackages* (WP) y el diagrama de Gantt asociado, está disponible en el **Anexo A**.

5 DESARROLLO

5.1 Entorno experimental y conjunto de datos

Antes de entrar a detallar cada una de las arquitecturas propuestas, es fundamental explicar con qué datos hemos alimentado a los modelos. Al fin y al cabo, por muy buena que sea la red neuronal, los resultados finales dependen directamente de la calidad de los datos de entrenamiento.

Conjunto de datos (Dataset)

Para entrenar los modelos se ha utilizado un conjunto de datos basado en el simulador CARLA [16], en concreto una variante adaptada de las secuencias de *Bench2Drive* [27]. A diferencia de otros enfoques comerciales que utilizan

multitud de sensores costosos como LiDAR o radares, este proyecto se ha limitado a usar pura visión por computador.

Cada instante de tiempo en el *dataset* está formado por las imágenes de tres cámaras frontales (izquierda, central y derecha) dispuestas para simular un amplio campo de visión. Como *Ground Truth*, no se han usado comandos directos de volante o pedales, sino coordenadas espaciales puras. Una secuencia temporal de 8 *waypoints* proyectados en metros reales que el coche debe alcanzar. Además, los datos se han dividido de forma clásica, utilizando un 94,8% para la fase de entrenamiento y reservando rutas totalmente nuevas para la validación correspondiendo al 5,2% restante. De esta forma, nos aseguramos de que la red realmente aprenda a conducir y no se limite a memorizar los mapas.

Hardware y entorno de ejecución

Entrenar redes neuronales que procesan secuencias de vídeo exige mucha potencia gráfica y memoria. Por suerte, todos los experimentos de este proyecto se han podido lanzar en un clúster de alto rendimiento equipado con múltiples unidades GPUs. Esto ha permitido paralelizar el trabajo, entrenar varios modelos pesados simultáneamente y reducir tiempos de experimentación que normalmente tardarían semanas a apenas unas horas o pocos días.

Configuración del entrenamiento (Hiperparámetros)

Aunque cada arquitectura tiene sus propias particularidades, la base del entrenamiento se ha mantenido lo más consistente posible para que las comparativas fuesen justas:

- **Optimizador:** Se ha utilizado *AdamW* por su excelente equilibrio entre velocidad de convergencia y control del sobreajuste mediante la regularización de los pesos (*weight decay*).
- **Tamaño de lote (Batch Size):** Se ajustó a 30 muestras por iteración, un tamaño ideal para no llegar a saturar la memoria y provocar cuellos de botella.
- **Tasa de aprendizaje (Learning Rate):** Para los modelos iniciales se utilizó una tasa de 1×10^{-4} .

Toda esta configuración ha servido como un terreno de juego estable sobre el que se han construido y evaluado las diferentes arquitecturas que se detallan en los siguientes apartados.

5.2 Arquitectura

Para empezar con este proyecto, es fundamental tener una primera aproximación y que esta sea a su vez un modelo base, es decir, un modelo referencia que nos permita comparar en un futuro con otros modelos para observar diferentes métricas, tendencias y resultados. Por ende, el primer paso ha sido la elección del conjunto de modelos o arquitecturas baselines. Esta elección debe ser hecha de forma consciente y de forma coherente. Esto se debe a que una mala elección de los baselines puede suponer cambios significativos en nuestros resultados al ser la referencia comparativa. Bien, para este proyecto se ha llegado a la conclusión de que la base de los baselines es el modelo CIL++, que a su vez

será uno de los tres baselines que se llevarán a cabo. Como ya se ha presentado en el Estado del Arte, CIL++ es una evolución directa del CILRS [21], y su enfoque de imitación por visión nos ofrece una base sólida y conocida sobre la que comparar con las arquitecturas que se desarrollarán posteriormente [22].

5.2.1 Baseline basado en CIL++

Recordando lo ya explicado en el Estado del Arte, CIL++ es un modelo de conducción autónoma por imitación que, a partir de las imágenes de tres cámaras frontales, predice directamente el *steering* y la aceleración usando solo imágenes RGB, sin mapas ni sensores adicionales. En este apartado no nos interesa volver a desglosar su funcionamiento interno, sino justificar por qué se ha elegido como pilar de los baselines.

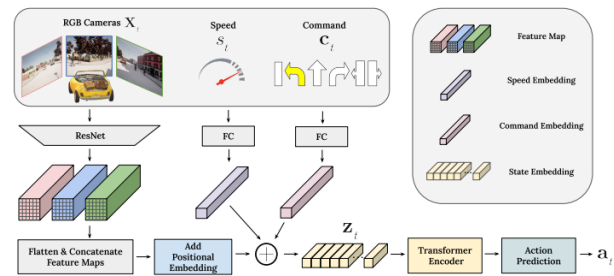


Fig. 2: Arquitectura de CIL++, compuesto principalmente de tres partes.[23]

La Fig. 2 muestra esta arquitectura: el *encoder* convolucional que resume la imagen en un vector de características y la capa de regresión que produce los comandos del coche.

Más allá de su relevancia como referente de la conducción end-to-end, la principal razón para elegir CIL++ es su arquitectura compacta y fácil de entrenar, que garantiza unos resultados de baseline estables, sin complejidades que introduzcan ruido en la comparación. Además, al apoyarse únicamente en imágenes RGB, constituye un punto de partida ideal para medir el impacto de incorporarle la profundidad, la predicción de pose y la de waypoints que se verán más adelante. Sus limitaciones son, precisamente, lo que motiva su uso: es un modelo que reacciona pero no anticipa, y no incorpora geometría alguna. Son esas carencias las que este proyecto pretende explorar, lo que hace que la comparación resulte correlativa y significativa.

5.2.2 Baseline basado en predicción de waypoints: enfoque MLP

Entrando en el corazón de este proyecto, tras seleccionar el modelo base de los baselines del apartado anterior y tras trabajar con el modelo CIL++ clásico, se ha considerado necesario adaptar esta arquitectura a las necesidades específicas del proyecto. Para ello, se ha modificado CIL++ con el propósito de que deje de predecir directamente los comandos de control (*steering* y *aceleración*) para pasar a predecir una secuencia de waypoints futuros que describen la trayec-

toria que debería seguir el vehículo durante los próximos metros.

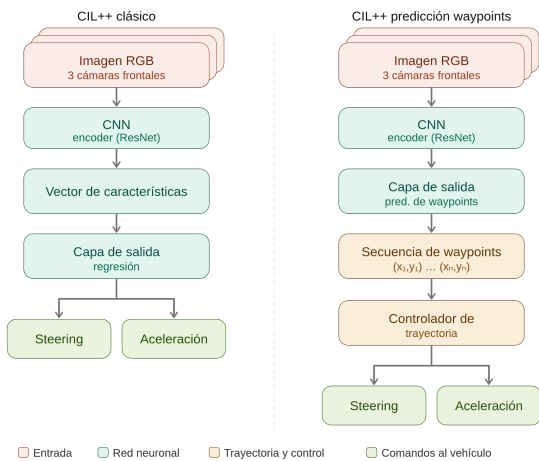


Fig. 3: Comparativa entre el modelo CIL++ clásico (salida en comandos de control) y la versión modificada con salida en *waypoints* futuros y control posterior.

Como se aprecia en la Fig. 3, la arquitectura original se mantiene prácticamente intacta en el encoder, pero la capa de salida de regresión orientada a comandos instantáneos se sustituye por un bloque que predice una secuencia de *waypoints* (x_i, y_i) . Esta modificación transforma la filosofía y enfoque del modelo original. Profundizando un poco más, la red toma una imagen y produce controles instantáneos por lo que, pese a ser un enfoque correcto como baseline, presenta limitaciones cuando se pretende obtener un comportamiento estable en escenarios y situaciones complejas debido a la alta sensibilidad al ruido, el exceso de dependencia del frame actual y carece de estructura para admitir la incorporación de información geométrica.

En cambio, la variante modificada de CIL++ empleada en este proyecto genera unos *waypoints* (coordenadas) relativos que representan puntos de la trayectoria futura a seguir. Estos *waypoints*, una vez generados, permiten construir un camino que puede ser usado para diferentes aplicaciones. En este proyecto, la aplicación directa del camino generado que más nos interesa es que a partir del camino con un algoritmo de seguimiento de trayectoria se puede transformar en *steering* y *aceleración* de manera más fiable y estable.

Para llevar a cabo esta etapa final de transformación de coordenadas a comandos de control físicos, en este proyecto se ha delegado esta fase a una red neuronal profunda, concretamente a un **Perceptrón Multicapa (MLP)**. En este caso, el MLP recibe el *waypoint* predicho por la red y aprende a mapearlo directamente hacia las acciones de control (*steering*, *aceleración* y *freno*). El entrenamiento es completamente supervisado y directo: para cada muestra se construye una etiqueta de control a partir de la geometría del propio *waypoint* objetivo (el ángulo hacia el punto determina el *steering* y la distancia al mismo determina la *aceleración* deseada), y el MLP se optimiza minimizando una función de pérdida L1 entre sus salidas y dicha etiqueta: el módulo de control aprende su comportamiento únicamente de la relación geométrica entre la trayectoria predicha y los comandos que la harían viable, manteniendo a la vez la lige-

reza y rapidez computacional necesarias para su ejecución en tiempo real en el simulador.

Cabe destacar que esta variante predice una secuencia completa de *waypoints* a través de la rama de percepción (CIL++), pero el módulo de control se alimenta únicamente del primer *waypoint* de dicha secuencia, que es el más cercano e inmediato al vehículo. Además, ambas ramas se entrenan de forma independiente: el gradiente del módulo de control no se propaga hacia la red de predicción de *waypoints*, de modo que cada cabeza aprende exclusivamente de su propia supervisión.

Pese a que dicha evolución de la arquitectura pueda parecer superficial y sin un impacto significativo a simple vista, ocurre todo lo contrario, ya que este cambio implica:

- Sustituir la capa de salida de regresión de CIL++ por una que produce un conjunto de waypoints (x_i, y_i) .
- Ajustar la función de pérdida para medir el error entre la trayectoria predicha y la trayectoria real obtenida del GT: Adentrándonos un poco más en este segundo punto, donde se menciona acerca de que la modificación del modelo nos obliga a también redefinir la función de pérdida del modelo. En lugar de un error sobre valores escalares de *steering* y *aceleración*, se utiliza una pérdida L1/L2 sobre cada *waypoint* predicho.
- Adaptar la estructura del modelo para que el vector de características generado por el encoder alimente un bloque de predicción de trayectorias capaz de generar coordenadas espaciales bidimensionales en lugar de comandos instantáneos.
- Acoplar un módulo independiente a la salida de la red (el modelo neuronal MLP).
- Integrar esta arquitectura modificada en el pipeline de evaluación del simulador, de manera que las decisiones del vehículo dependan de una trayectoria completa y no de solamente un par de comandos.

Con estas modificaciones, la arquitectura deja de estar limitada como se ha dicho anteriormente al procesamiento independiente de cada imagen y se convierte en una base adecuada y mejorada. A partir de aquí, el siguiente paso consiste en extender el modelo hacia un enfoque que aproveche la *moción*, es decir, secuencias de imágenes y predicciones dependientes del tiempo, lo que se aborda en detalle en el siguiente subapartado.

5.3 Adaptación Unsupervisada

Tras establecer y evaluar los baselines, el siguiente paso dentro de este proyecto es avanzar hacia un sistema de conducción progresivamente más autónomo, reduciendo su dependencia de datos anotados. Esta evolución se aborda en dos etapas complementarias: primero, se elimina la necesidad de disponer de etiquetas de control (*steering* y *aceleración*) durante el entrenamiento, delegando la generación de acciones a un modelo físico del vehículo; posteriormente, se suprime también la dependencia del GPS como fuente

de *waypoints* de referencia, sustituyéndolo por un módulo de visión tridimensional basado en *Structure-from-Motion*. La combinación de ambas estrategias da lugar a un sistema puramente visual, capaz de aprender a conducir sin etiquetas de control ni coordenadas GPS.

5.3.1 Entrenamiento sin acciones

Una vez demostrada la viabilidad del enfoque basado en *waypoints* con control MLP, el siguiente paso lógico consiste en preguntarse si es posible eliminar la dependencia de etiquetas de control (*steering*, aceleración) durante el entrenamiento. La respuesta es el **modelo cinemático de bicicleta** (*Kinematic Bicycle Model*), ya introducido brevemente en el Estado del Arte y cuya formulación completa se recoge en el **Anexo B**.

La idea central es la siguiente: en lugar de supervisar las acciones de control de forma directa con un *Ground Truth* de acciones, se incorpora el modelo cinemático como una **capa diferenciable** dentro de la propia red. La arquitectura resultante funciona de la siguiente manera: CIL++ procesa las imágenes y, en vez de generar *waypoints*, produce directamente una predicción de *steering* y aceleración. Estas dos magnitudes alimentan al modelo cinemático de bicicleta, que las proyecta hacia adelante en el tiempo aplicando las ecuaciones físicas del movimiento del vehículo (posición, orientación y velocidad), generando así una secuencia de *waypoints* futuros. El entrenamiento se realiza comparando estos *waypoints* proyectados contra los *waypoints* reales del *Ground Truth* (obtenidos del GPS), mediante una pérdida L1.

La ventaja fundamental de este diseño es que el *steering* y la aceleración se aprenden de forma **implícita**, sin necesidad de disponer de etiquetas de control. Al retropropagar el error de los *waypoints* a través de las ecuaciones cinemáticas (que son diferenciables), la red ajusta sus predicciones de control hasta que la trayectoria proyectada coincida con la real. Además, al imponer las restricciones físicas del vehículo (distancia entre ejes, radio de giro máximo, inercia), el modelo cinemático garantiza que las trayectorias generadas sean físicamente viables, evitando maniobras imposibles que un MLP libre podría producir.

5.3.2 Entrenamiento sin waypoints

Resuelto el problema de prescindir de etiquetas de control, queda por abordar la segunda dependencia: los *waypoints* de referencia obtenidos del GPS. Para eliminarla, se ha incorporado información geométrica del entorno mediante la técnica de *Structure-from-Motion* (SfM), tomando como referencia la arquitectura de *Monodepth2* [24], que es ampliamente reconocida por su capacidad para estimar profundidad y la posición de la cámara en situaciones con cámaras monoculares sin supervisión directa.

El propósito de integrar un modelo a partir de *Monodepth2* es aportar al sistema una representación espacial más detallada, capaz de capturar la estructura 3D de la escena a partir de una secuencia de imágenes consecutivas. Esto es

fundamental para mejorar la comprensión del entorno y proporcionar al modelo de predicción de trayectorias una base geométrica sobre la que predecir de forma más coherente. Por lo tanto, más que un refuerzo, es un complemento necesario que garantiza una alta robustez a nuestro sistema.

En esta aproximación, se emplea un par de redes neuronales. Un *Depth Network* que se encarga de estimar un mapa de profundidad relativo para cada fotograma, y un *Pose Network* que este predice la posición de la cámara en un frame mediante la transformación 3D entre imágenes adyacentes. Esta combinación de ambas redes nos permite reconstruir una representación espacial en 3D aproximada del entorno, siguiendo una línea propia del SfM pero mediante un aprendizaje auto-supervisado.

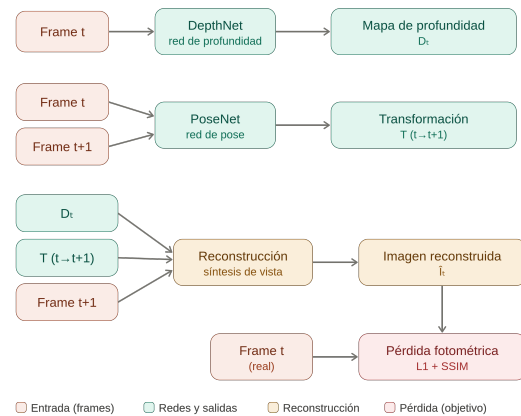


Fig. 4: Esquema del modelo basado en *Structure-from-Motion*, inspirado en *Monodepth2*, con red de profundidad y red de pose.

Antes de seguir y en este contexto, el funcionamiento del *Structure-from-Motion* [11] se basa en la triangulación. La idea es simple. Cuando un mismo punto de la escena aparece en varios fotogramas consecutivos, cada uno de esos puntos en la imagen define una recta o rayo tridimensional que parte del centro de la cámara y atraviesa ese punto que vendría a ser un píxel. Y si además conocemos cómo se ha movido la cámara entre esos fotogramas en el sentido de sus transformaciones geométricas (rotaciones y traslaciones), podemos colocar todos esos rayos en un mismo sistema de referencia en 3D empezando a reconstruir punto tras punto en el entorno 3D. El punto 3D real se define como la intersección de dichas rectas. Es conveniente remarcar que este proceso no tiene nada que ver con los puntos de horizonte ya que, debido a la naturaleza del SfM, es propenso y normal que se llegue a ver de dicha forma. En resumen, SfM aprovecha cómo cambia la proyección de un mismo punto a lo largo del tiempo para recuperar su profundidad y reconstruir la estructura tridimensional de la escena.

Tal y como se muestra en la Fig. 4, las predicciones de profundidad y pose se combinan para sintetizar la vista objetivo y definir la pérdida fotométrica.

En cuanto a la función de pérdida utilizada en este modelo se basa en la reconstrucción del frame objetivo a partir de la vista generada mediante las predicciones de profundidad y pose. Para ello, se proyectan los píxeles del fotograma de

referencia al fotograma objetivo utilizando la transformación geométrica estimada, y se comparan ambas imágenes mediante una pérdida fotométrica compuesta por un L1 y un *Structural Similarity Index* (SSIM) [25]. De esta manera, el modelo aprende a generar mapas de profundidad coherentes con la geometría real de la escena sin necesidad de valores de profundidad en el *Ground Truth*.

Una vez entrenado, el módulo SfM proporciona información geométrica que puede emplearse en varias etapas del proyecto. En particular, la profundidad estimada permite enriquecer la entrada de los modelos propuestos a continuación, permitiendo que la red capture relaciones espaciales que no pueden extraerse únicamente de la intensidad RGB de las imágenes. Asimismo, la estimación del movimiento entre imágenes añade una referencia útil sobre cómo se está desplazando el vehículo. Esto ayuda a generar trayectorias más estables y coherentes, especialmente en situaciones con cambios de velocidad, curvas pronunciadas o entornos menos habituales.

6 RESULTADOS

6.1 Evaluación

En esta sección se describirá cómo se evaluarán los modelos desarrollados a lo largo del proyecto y de qué manera se compararán entre sí. El objetivo no es solamente obtener una cifra de error aislada, sino entender hasta qué punto cada componente introducido aporta algo tangible y perceptible al comportamiento final del sistema de conducción autónoma.

La evaluación combinará dos tipos de análisis complementarios:

- Evaluación cuantitativa: basada en métricas numéricas calculadas sobre conjuntos de datos de validación y en simulaciones cerradas en el entorno de pruebas CARLA.
- Evaluación cualitativa: centrada en la inspección y verificación visual de trayectorias, mapas de profundidad y escenas concretas donde los modelos aciertan o fallan de manera significativa.

6.1.1 Evaluación de los modelos de conducción (CIL++ y predicción de *waypoints*)

Para analizar el rendimiento de los modelos encargados directamente de la conducción (tanto el baseline CIL++ clásico como sus variantes basadas en trayectorias), la evaluación se dividirá en dos fases: *offline* y *online*.

En la fase *offline*, el rendimiento se medirá sobre un conjunto de secuencias de validación no empleadas durante el entrenamiento. En el caso del modelo CIL++ original, se monitorizará la evolución de la función de pérdida (L1/L2) aplicada directamente sobre los comandos escalares de aceleración y *steering*. Por el contrario, para el modelo modificado, el error se medirá sobre las coordenadas espaciales (x_i, y_i) de cada *waypoint* predicho frente al *Ground Truth*,

estudiando gráficamente cómo crece este error posicional a medida que aumenta la distancia temporal en la trayectoria.

En la fase *online*, se llevarán a cabo episodios de conducción reales dentro del simulador CARLA, enfrentando a los modelos a diversos escenarios (rectas, curvas suaves, intersecciones y presencia de tráfico). Para los modelos basados en *waypoints*, estos se transformarán en acciones físicas mediante el módulo de control posterior descrito anteriormente (MLP). Al someter a todos los modelos a las mismas rutas, se podrán comparar de forma justa extrayendo las siguientes métricas:

- Promedio de la puntuación final de conducción.
- Promedio del porcentaje de rutas completadas sin intervención.
- Promedio de penalizaciones por infracciones indeseadas, como frenazos bruscos, salidas de carril o giros demasiado agresivos.

Desde el punto de vista cualitativo, se analizarán ejemplos concretos para comprobar si el enfoque por *waypoints* supone una mejora real frente al baseline reactivo en situaciones complejas (como curvas cerradas o zonas de baja visibilidad), identificando también aquellos escenarios donde aún se produzcan fallos o patrones recurrentes.

6.1.2 Evaluación de los modelos basados en profundidad (SfM)

En los modelos basados en *Structure-from-Motion*, la evaluación tiene un enfoque más geométrico y visual, centrándose principalmente en la calidad de los mapas de profundidad estimados y en la coherencia de la pérdida durante el entrenamiento.

En primer lugar, se analizará la curva de la pérdida fotométrica (L1 + SSIM) a lo largo de los epochs para comprobar que el modelo auto-supervisado converge de forma estable, detectando posibles estancamientos o problemas de *overfitting*.

En segundo lugar, se seleccionarán varios fotogramas de prueba para visualizar cualitativamente los mapas de profundidad predichos junto a sus imágenes RGB originales. Aunque el entrenamiento se realiza sin *Ground Truth* de profundidad explícito, las predicciones se compararán visualmente con mapas de profundidad generados por el propio simulador a modo de referencia. El objetivo es verificar si la red comprende la estructura global de la escena (carretera, edificios, vehículos, vegetación) y distingue con claridad los objetos cercanos de los lejanos.

Finalmente, se evaluará el impacto real de esta información geométrica una vez se integre con el modelo de predicción de *waypoints*, observando si la profundidad ayuda a estabilizar las trayectorias y a reducir errores en situaciones complejas.

6.1.3 Comparativa global

Una vez analizados los componentes por separado, se realizará una comparativa global definitiva que confronte:

- El baseline CIL++ clásico.
- Los modelos con predicción de *waypoints* (con control MLP).
- El sistema completo que enriquece la predicción integrando el módulo SfM.

Esta síntesis final se apoyará tanto en tablas resumen de resultados cuantitativos como en comparativas visuales. El propósito último de esta sección es demostrar empíricamente si la evolución desde un enfoque end-to-end directo hacia una arquitectura modular (con representaciones geométricas de trayectoria y profundidad) justifica la complejidad añadida y aporta la robustez esperada al vehículo autónomo.

6.2 Resultados

6.2.1 Resultados comparativos de los baselines

Tras definir la metodología de evaluación, en esta subsección se presentan los resultados empíricos obtenidos en el simulador *CARLA* para todos los modelos desarrollados en las fases previas del proyecto. La TABLA 1 recoge el promedio de la puntuación de conducción (*Avg. driving score*), el porcentaje de ruta completada (*Avg. route completion*) y el multiplicador de penalización por infracciones (*Avg. infraction penalty*).

Model	Prediction (WP/Actions)	GT (GPS)	Control GT	Avg. driving score	Avg. route completion	Avg. infraction penalty
CIL++ Vanilla	Actions	No	Sí	63,63	73,73	0,86
CIL++ WP (MLP)	Ambos	Sí	Sí	70,03	81,61	0,83
CIL++ CINEMATICO	Ambos	Sí	No	46,59	73,50	0,63
CIL++ MLP SFM	Ambos	No	Sí	30,41	63,34	0,51
CIL++ CINEMATICO SFM	Ambos	No	No	37,67	65,93	0,60

TABLA 1: RESULTADOS OBTENIDOS EN LA EVALUACIÓN *online* DE TODOS LOS MODELOS CONSIDERADOS EN EL SIMULADOR *CARLA*.

Observando los datos obtenidos, se pueden extraer conclusiones muy claras sobre el comportamiento de cada arquitectura. En primer lugar, el modelo **CIL++ Vanilla** demuestra ser un punto de partida muy sólido, logrando completar casi el 74% de las rutas en promedio y obteniendo una puntuación final de 63,63. Esto confirma que el enfoque *end-to-end* clásico es funcional, aunque presenta un claro margen de mejora en escenarios complejos.

Por su parte, la variante **CIL++ WP (MLP)** se consolida como vencedora inequívoca de esta etapa. Gracias a su módulo de control basado en aprendizaje, el sistema no solo fue capaz de seguir los waypoints de forma fluida, sino que superó ampliamente al modelo Vanilla. Alcanzó un 81,61% de ruta completada y elevó la puntuación global a 70,03. Aunque su penalización por infracción (0,830) refleja que aún comete pequeños errores conductuales puntuales, la notable mejora en la capacidad de avance y adaptación al

entorno justifica con creces el salto hacia este enfoque modular guiado por aprendizaje automático.

6.2.2 Modelos Avanzados: Enfoque Cinemático e Integración de SfM

Continuando con el análisis de la TABLA 1, el modelo **CIL++ CINEMATICO** (que emplea el *Kinematic Bicycle Model* en lugar del MLP; véase **Anexo B**) presenta un comportamiento muy revelador. Obtiene una puntuación global de 46,59 y completa el 73,50% de las rutas. Contrario a la hipótesis inicial de que la imposición de restricciones cinemáticas aportaría mayor seguridad, los resultados demuestran lo contrario: el modelo cinemático presenta una mayor tasa de colisiones y fallos en comparación con el modelo MLP. Como se observa en el análisis cualitativo, esta arquitectura sufre de una rigidez excesiva; al intentar seguir una trayectoria geométrica pura, el vehículo carece de la flexibilidad necesaria para adaptarse a imprevistos dinámicos o ajustar su trayectoria ante errores de percepción, derivando en choques que el enfoque MLP, al ser más "plástico" y basado en aprendizaje directo de la relación entre trayectoria y control, logra mitigar con mayor eficacia.

El punto de inflexión del proyecto llega al evaluar los modelos sin GPS, es decir, las variantes que dependen exclusivamente de la visión por computador para guiarse: **CIL++ MLP SFM** y **CIL++ CINEMATICO SFM**.

Al observar la tabla, el rendimiento de estos modelos cae drásticamente a puntuaciones de 30,41 y 37,67 respectivamente. Durante el análisis exhaustivo de estos resultados atípicos, se descubrió que el origen no era un fallo de la red neuronal, sino un problema clásico de ingeniería de datos: un error de inconsistencia en la escala.

Dado que la red basada en *Structure-from-Motion* predice coordenadas relativas de cámara, durante el preprocesamiento se aplicó un factor de escala (aprox. $\times 130.98$) para trasladar estos valores a metros reales del entorno de *CARLA*. Sin embargo, un error en la pipeline de evaluación provocó que este multiplicador se aplicara por duplicado durante la inferencia en el simulador. Como resultado, la red estaba ordenando al vehículo alcanzar waypoints proyectados a más de 5 kilómetros de distancia en apenas fracciones de segundo.

Ante este alto valor de error, el modelo **CIL++ MLP SFM** intentó ejecutar aceleraciones inasumibles, lo que colapsó el sistema y hundió su puntuación. Resulta bastante interesante comparar esto con el modelo **CIL++ CINEMATICO SFM** (37,67). A pesar de sufrir exactamente el mismo error de escala astronómico, su puntuación fue cercana a la del MLP. Esto demuestra empíricamente que, aunque el modelo cinemático es propenso a colisiones por su rigidez, sus ecuaciones físicas (véase **Anexo B**) actuaron como una barrera de seguridad natural, recortando las aceleraciones absurdas demandadas por la red y evitando un fallo catastrófico total.

Una vez identificado y subsanado el error de escalado en el código, el sistema demostró la verdadera capacidad del módulo SfM. Tal y como se ilustra en la **Fig. 5**, tras la corrección matemática, los waypoints proyectados por la red de

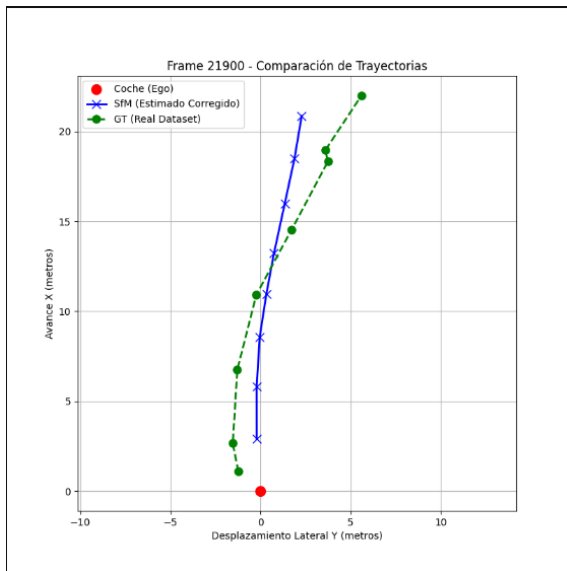


Fig. 5: Visualización del Frame 21900: Comparativa entre la trayectoria real del GT (verde) y la reconstrucción corregida mediante SfM (azul). Se observa la alta fidelidad geométrica obtenida exclusivamente por métodos visuales.

estimación de pose (SfM) logran calcar con una precisión decente la curvatura de la carretera dictada por el Ground Truth (GPS). Esta gráfica es la prueba definitiva de que es posible reconstruir la geometría del mundo real y anticipar la trayectoria de un vehículo de forma robusta, segura y puramente visual, eliminando por completo la dependencia estructural de sensores externos de posición.

Al evaluar los modelos con dicha corrección, se observó que, a pesar de que la geometría estática es prácticamente idéntica al Ground Truth (como se vio en la Fig. 5), las puntuaciones en CARLA no experimentaron la subida esperada, manteniéndose en valores iguales o inferiores a los de la TABLA 1. A pesar de haber arreglado la escala geométrica, las puntuaciones en simulación online confirmaron la presencia de error acumulado, que viene a ser el mayor reto de la odometría visual. El problema no está en predecir la trayectoria en un fotograma suelto, donde la red es realmente precisa, sino en encadenar esas predicciones a lo largo de una ruta. Al ser un sistema de lazo cerrado, los pequeños fallos al estimar la pose no se descartan, sino que se suman. Un desvío inicial de unos pocos centímetros genera un cálculo erróneo para el siguiente tramo, causando una pérdida de control progresiva.

7 CONCLUSIONES

Tras todo el desarrollo y la experimentación de las diferentes arquitecturas, este proyecto permite extraer conclusiones claras y fundamentadas cuantitativamente. La principal es que la transición de un modelo puramente reactivo hacia una arquitectura modular basada en waypoints y guiada por aprendizaje justifica plenamente su implementación, alcanzando una puntuación de conducción de 70,03 y un 81,61% de ruta completada, frente al 63,63 y 73,73% del baseline CIL++. En contraste, el modelo cinemático obtuvo una puntuación de 46,59 con un 73,50% de ruta completada,

demostrando que no toda complejidad añadida implica una mejora en entornos dinámicos. La rigidez de este modelo (tal y como se anticipó en el apartado 5.3.1 al proponer el entrenamiento sin acciones) resultó contraproducente, restando al vehículo la flexibilidad necesaria para reaccionar ante imprevistos o recuperarse de errores. Finalmente, la integración de Structure-from-Motion (SfM) constituye la aportación más ambiciosa. Las variantes CIL++ MLP SfM y CIL++ CINEMATICO SfM obtuvieron puntuaciones de 30,41 y 37,67 respectivamente. Aunque el análisis ha corroborado que el módulo SfM es capaz de reconstruir la geometría con fidelidad altísima cuando se evalúa fotograma a fotograma, su integración en simulación demostró que esta precisión estática es insuficiente para un entorno de lazo cerrado (closed-loop). El problema reside en el error acumulado: desviaciones minúsculas en la estimación de pose se arrastran y suman de forma continua, provocando una deriva que acaba inevitablemente en la pérdida de control del vehículo.

7.1 Líneas de trabajo futuro

A raíz de lo anterior, la línea de mejora más evidente para continuar este trabajo pasa por atacar directamente ese error acumulado. Más allá de conseguir una predicción estática perfecta, para mejorar en CARLA hace falta introducir algún mecanismo que establezca lo que ve la red a lo largo de toda la secuencia y no deje que los pequeños fallos se arrastren. La opción más natural sería incorporar filtros temporales, como el Filtro de Kalman [26], que permitan suavizar y corregir la estimación de la trayectoria fotograma a fotograma, fusionando la predicción de la red con un modelo del movimiento esperado del vehículo. Con un componente de este tipo, es normal esperar que el sistema puramente visual reduzca la incertidumbre y se acerque mucho más al rendimiento de las variantes que sí disponen de GPS, que sería el siguiente paso lógico para cerrar del todo este proyecto.

8 USO DE LA IA GENERATIVA

El detalle completo del uso de la IA generativa en este trabajo se recoge en el **Anexo C**.

REFERÈNCIES

- [1] World Health Organization, *Global Status Report on Road Safety 2023*, World Health Organization, Geneva, 2023.
- [2] A. Tampuu, T. Matiisen, M. Kodelja, I. Illium, M. Mandre, D. Fischgardt, and K. Korjus, *A Survey of End-to-End Driving: Architectures and Training Methods*, IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 10, pp. 4577–4595, 2020.
- [3] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, *A Survey of Deep Learning Techniques for Autonomous Driving*, Journal of Field Robotics, vol. 37, no. 3, pp. 362–386, 2020.

- [4] J. Janai, F. Güney, A. Behl, and A. Geiger, *Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art*, Foundations and Trends® in Computer Graphics and Vision, vol. 12, no. 1–3, pp. 1–308, 2020.
- [5] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, *nuScenes: A Multimodal Dataset for Autonomous Driving*, Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11621–11631, 2020.
- [6] D. A. Pomerleau, *ALVINN: An Autonomous Land Vehicle in a Neural Network*, Advances in Neural Information Processing Systems (NeurIPS), vol. 1, 1988.
- [7] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al., *End to End Learning for Self-Driving Cars*, arXiv preprint arXiv:1604.07316, 2016.
- [8] F. Codevilla, M. Müller, A. Lopez, V. Koltun, and A. Dosovitskiy, *End-to-End Driving via Conditional Imitation Learning*, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- [9] K. Chitta, A. Prakash, and A. Geiger, *TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving*, Conference on Robot Learning (CoRL), 2021.
- [10] N. Snavely, S. M. Seitz, and R. Szeliski, *Photo Tourism: Exploring Photo Collections in 3D*, ACM Transactions on Graphics (SIGGRAPH), vol. 25, no. 3, pp. 835–846, 2006.
- [11] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, *A Survey of Structure from Motion*, Acta Numerica, vol. 26, pp. 305–364, 2017.
- [12] C. Godard, O. Mac Aodha, and G. J. Brostow, *Unsupervised Monocular Depth Estimation with Left-Right Consistency*, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [13] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, *Unsupervised Learning of Depth and Ego-Motion from Video*, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [14] V. Guizilini, R. Ambruş, S. Pillai, A. Raventos, and A. Gaidon, *3D Packing for Self-Supervised Monocular Depth Estimation*, Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2485–2494, 2020.
- [15] A. Geiger, P. Lenz, and R. Urtasun, *Vision Meets Robotics: The KITTI Dataset*, International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.
- [16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, *CARLA: An Open Urban Driving Simulator*, Proc. of the 1st Annual Conference on Robot Learning (CoRL), pp. 1–16, 2017.
- [17] B. Paden, M. Cáp, S. Z. Yong, D. Yershov, and E. Frazzoli, *A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles*, IEEE Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 33–55, 2016.
- [18] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, *Trajectory-guided Control Prediction for End-to-end Autonomous Driving: A Simple yet Strong Baseline*, Advances in Neural Information Processing Systems (NeurIPS), 2022.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is All you Need*, Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.
- [20] S. Sachdeva, *Scrum Methodology*, International Journal of Engineering and Computer Science, vol. 5, no. 10, pp. 18471–18474, 2016.
- [21] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, *Exploring the Limitations of Behavior Cloning for Autonomous Driving*, Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
- [22] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, *End-to-End Driving via Conditional Imitation Learning*, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- [23] Repository / Model architecture reference for CIL++, *CARLA Conditional Imitation Learning implementation*.
- [24] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, *Digging Into Self-Supervised Monocular Depth Estimation*, Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
- [25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image Quality Assessment: From Error Visibility to Structural Similarity*, IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.
- [26] R. E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, Journal of Basic Engineering, vol. 82, no. 1, pp. 35–45, 1960.
- [27] W. Cao et al., *Bench2Drive: Towards Multi-Ability Autonomous Driving in Closed-Loop Simulation*, arXiv preprint arXiv:2406.01367, 2024.

A ANEXO: PLANIFICACIÓN

Este proyecto se divide principalmente en 4 workpackages (WP) donde en cada uno se trata un elemento fundamental y más general del proyecto. Cada WP puede durar desde 3 semanas hasta 9 semanas en función de la complejidad y dedicación que requiera cada uno. Por último, antes de ir a una explicación más detallada de cada WP, es fundamental mencionar que cada workpackage está formado, como

mínimo, por 2 tareas para favorecer la diversificación y la buena gestión del tiempo durante la estancia del proyecto.

WP1 Contextualización:

Al inicio de este proyecto se llevará a cabo una investigación a fondo de modelos recientes, actuales y prometedores (T1.1) para, posteriormente, definir los objetivos a cumplir, desde los más generales hasta los más específicos. A su vez, también se realizará una planificación detallada, donde se asignará a cada tarea un intervalo de semanas en el que se desarrollará (T1.2). La duración de este WP1 es adherente a su nivel de complejidad y, por ello, tendrá una duración de 4 semanas.

WP2 Implementación:

A partir de este punto, se comenzará con la parte de la implementación. En este WP2 nos centraremos inicialmente en la creación de modelos *baselines* (T2.1). Dentro de los *baselines*, tendremos tres principales modelos reutilizables posteriormente. Los tres modelos serán una selección de entre modelos basados en temporalidad de cámaras y modelos basados en profundidad. Además de los *baselines*, durante este WP2 también se llevarán a cabo los modelos *proposal*, que, como bien indica el nombre son modelos propuesta que tienen la intención de superar a los *baselines*. Estos modelos son: Modelo basado en MLP (Multi-Layer-Perceptron) (T2.3) y por último, Modelo basado en modelo cinemático (T2.4). Asimismo, este WP2 promete ser el más complejo de entre los existentes y, como consecuencia, será el que mayor dedicación reciba, con 9 semanas.

WP3 Evaluación:

Para concluir la parte práctica de este proyecto, se llevará a cabo este tercer WP3. En esta parte nos centraremos más en la evaluación de los tres últimos modelos (T2.2 y T2.3), comparándolos con el modelo *baseline* (T2.1). De esta comparación nacen las tareas de pruebas (T3.1) y análisis (T3.2). Un aspecto importante a mencionar es que este WP3, a diferencia de otros, se empieza a mediados del anterior WP2, solapando ciertas semanas en las que se trabajará tanto el WP2 como el WP3.

WP4 Documentación:

Por último, y no menos importante, finalizaremos este proyecto con una de las partes más significativas, como lo es la documentación de todo el proceso y las tareas que se han mencionado anteriormente. Empezaremos con el informe final (T4.1), en el que se plasmarán y refinarán los resultados y conclusiones obtenidos del anterior WP3. A continuación, se pasará a la preparación de la presentación (T4.2), que se expondrá ante el jurado. Este WP recibirá una duración de 3 semanas.

La **Fig. 6**, muestra el diagrama de Gantt con detalle de todos los WP y tareas planificadas que se realizarán a lo largo de las 19 semanas de este proyecto.

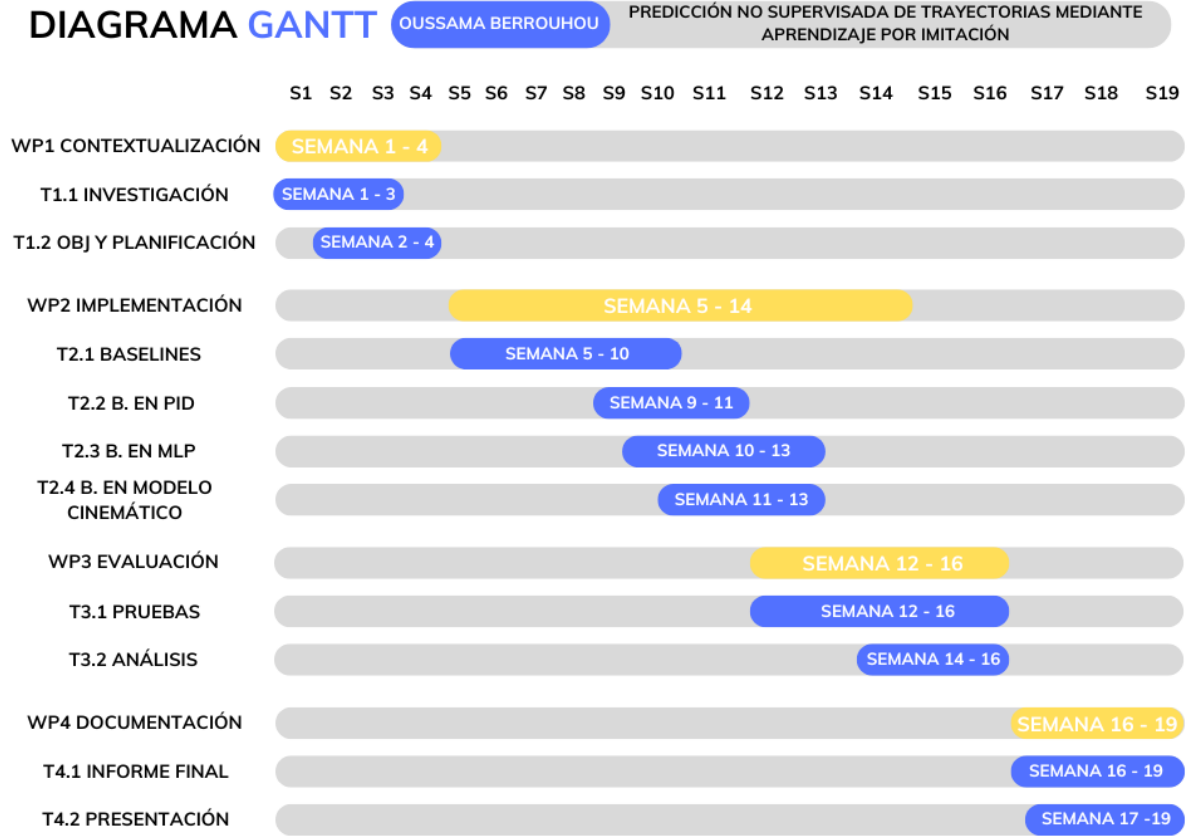


Fig. 6: Diagrama de Gantt

B ANEXO: FÓRMULAS MATEMÁTICAS

En este anexo se recoge la formulación matemática del modelo cinemático de bicicleta, referenciada a lo largo del cuerpo principal del informe.

B.1 Modelo Cinemático de Bicicleta (*Kinematic Bicycle Model*)

El modelo cinemático de bicicleta describe el movimiento de un vehículo de cuatro ruedas transformando cada par de ruedas como una sola rueda centrada en el eje correspondiente. Bajo la hipótesis de no deslizamiento lateral y velocidad reducida, la evolución del estado del vehículo queda definida por el siguiente sistema de ecuaciones:

$$\dot{x} = v \cos(\psi + \beta) \quad (1)$$

$$\dot{y} = v \sin(\psi + \beta) \quad (2)$$

$$\dot{\psi} = \frac{v}{l_r} \sin(\beta) \quad (3)$$

$$\beta = \arctan\left(\frac{l_r}{l_f + l_r} \tan(\delta_f)\right) \quad (4)$$

donde las variables y parámetros son:

- (x, y) : coordenadas de posición del centro de gravedad del vehículo en el plano horizontal.
- ψ : ángulo de orientación respecto a un eje de referencia fijo.
- v : velocidad lineal del vehículo (módulo).
- δ_f : ángulo de dirección de la rueda delantera, salida directa del módulo de control.
- β : ángulo de deslizamiento lateral del centro de gravedad, calculado a partir de δ_f y la geometría del vehículo.
- l_f : distancia desde el centro de gravedad hasta el eje delantero.
- l_r : distancia desde el centro de gravedad hasta el eje trasero.
- $L = l_f + l_r$: distancia total entre ejes.

Este modelo se integra en la arquitectura CIL++ Cinemática para restringir las trayectorias predichas a maniobras físicamente posibles, garantizando que el vehículo no realice giros o aceleraciones imposibles acorde con sus dimensiones y dinámica real.

C ANEXO: USO DE LA IA GENERATIVA

La inteligencia artificial generativa se ha utilizado como herramienta de apoyo en la redacción y revisión del presente trabajo, empleándose cautelosamente y verificando cada respuesta posteriormente. En ningún caso se ha empleado para generar contenido técnico, resultados ni conclusiones. Concretamente, se ha empleado para:

- La sustitución de palabras repetidas mediante sinónimos adecuados. Por ejemplo, en lugar de repetir constantemente términos como "enfoque" o "método" en el apartado de Estado del Arte, se han introducido alternativas equivalentes como "estrategia", "propuesta" o "línea de trabajo" para mejorar la fluidez del texto.
- La corrección de faltas ortográficas y gramaticales detectadas durante la redacción, como la normalización de acentos en palabras técnicas (por ejemplo, "autónoma", "visión", "parámetros").
- La mejora de la estructura y el formato del documento, ayudando a reorganizar párrafos para lograr una mayor coherencia. Un ejemplo de ello fue la reestructuración del apartado de Introducción, donde se diferenciaron claramente los bloques dedicados al contexto, las limitaciones actuales y los objetivos del trabajo.
- La elección de la palabra más idónea dentro de un conjunto de opciones propuestas por el autor. Por ejemplo, en la sección de Metodología, se evaluaron alternativas como "metodología ágil", "metodología iterativa" o "metodología inspirada en SCRUM" [20], optando finalmente por esta última.

Algunos ejemplos de las consultas realizadas son:

- *"¿Podrías sugerir mejoras de coherencia y fluidez en este párrafo sin cambiar su contenido?"*
- *"Busca un sinónimo para evitar repetir 'enfoque' tres veces seguidas en este apartado."*
- *"Ordena estas dos frases para que la idea avance de forma lógica."*
- *"¿Esta oración suena demasiado informal para un TFG?"*