FACULTAD DE INGENIERIA
PROGRAMA DE DOCTORADO EN BIOLOGÍA COMPUTACIONAL
SEDE SANTIAGO

CODIFICACIÓN PREDICTIVA DE LA TRAYECTORIA DE OBJETOS EN MOVIMIENTO USANDO FLUJO ÓPTICO

Tesis para optar al Grado de Doctor en Biología Computacional

Profesor Tutor: Dr. Tomas Pérez-Acle
Profesor Co-tutor: Dr. Cesar Ravello

Estudiante: Soraya Paz Mora Barrientos

Universidad San Sebastián

Programa de Doctorado en Biología Computacional

CODIFICACIÓN PREDICTIVA DE LA TRAYECTORIA DE OBJETOS EN MOVIMIENTO USANDO FLUJO ÓPTICO

Autora: Soraya Paz Mora Barrientos

Tutor: Dr. Tomás Pérez-Acle Co-tutor: Dr. César Ravello

AGRADECIMIENTOS

Quiero expresar mi más profundo agradecimiento a mi familia, cuyo apoyo incondicional fue fundamental para mantenerme firme durante este largo proceso de doctorado.

En particular, extiendo un reconocimiento especial a Francisco Gutiérrez, por su comprensión, paciencia y compañía constante, así como por brindarme la motivación necesaria en los momentos más desafiantes.

Asimismo, deseo agradecer a mis tutores, el Dr. Tomás Pérez-Acle y el Dr. César Ravello, por su guía académica, sus valiosas observaciones y la confianza depositada en mi trabajo. Su orientación fue decisiva para dar forma y coherencia a esta tesis, y su compromiso con la investigación ha sido para mí una fuente de inspiración.

A todos ellos, mi más sincera gratitud.

De igual forma, quiero agradecer a las instituciones que financiaron mi doctorado.

- Beca de Doctorado y Proyecto USS-FIN-24-PASD-01.
- Becas de viaje financiado por la Vicerrectoría de Investigación y Doctorados, Universidad San Sebastián.
- Proyecto FONDECYT Exploración N° 13240042.
- Centro Ciencia & Vida, FB210008, Financiamiento Basal para Centros Científicos y Tecnológicos de Excelencia de ANID.
- Air Force Office of Scientific Research FA9550-24-1-0001 – Proyecto Nevis.

Índice

Índice de figuras

Índice de tablas

.

1. Abreviaturas

ACCLIP Modelo Corrección Adaptativa con Pérdida Combinada Integrando Flujo Óptico para una Mejora en la Predicción

AE Error Angular

CNN Redes Neuronales Convolucionales

ConvGRU GRU Convolucional

EPE End-Point Error

GRU Gated Recurrent Unit

LPIPS Learned Perceptual Image Patch Similarity

LSTM Long Short-Term Memory

MAE Error Absoluto Medio

MSE Error Cuadrático Medio

PE Modelo Pre Encoder

PLS Modelo Post Latent Space

PSNR Peak Signal-to-Noise Ratio

RNN Redes Neuronales Recurrentes

SNC Sistema Nervioso Central

SSIM Índice de Similitud Estructural

PE Modelo Pre Encoder

PLS Modelo Post Latent Space

2. Resumen

La capacidad de anticipar cómo se moverán los objetos en una secuencia de vídeo se ha transformado en uno de los grandes desafíos dentro de la visión computacional ya que es escencian para generar aplicaciones como la navegación autónoma, la vigilancia y la robótica, donde la anticipación de escenarios futuros resulta esencial para la toma de decisiones. No obstante, a pesar de los avances alcanzados, los métodos actuales aún presentan limitaciones importantes. Por ejemplo, las arquitecturas convolucionales y sus variantes 3D muestran un buen rendimiento en horizontes temporales cortos y medianos, pero pierden coherencia en el movimiento cuando se extiende el número de predicciones. Por otro lado, los modelos recurrentes y aquellos basados en interpolación aportan mayor consistencia temporal, pero presentan un elevado consumo de recursos y dificultades para adaptarse a escenarios reales complejos. Dentro de este panorama, el flujo óptico aparece como una herramienta clave porque ofrece una descripción explícita del movimiento aparente en la escena bajando el coste computacional. Sin embargo, en la práctica, este recurso ha sido integrado de manera limitada dentro de las arquitecturas de predicción, lo que reduce su potencial para garantizar estabilidad y coherencia en horizontes de tiempo más largos. Esto planteó una pregunta central que guió el presente trabajo: *¿en qué punto de una arquitectura resulta más efectivo incorporar el flujo óptico para obtener predicciones confiables en entornos urbanos?* Para responder a esta pregunta, se exploraron tres modelos inspirados en codificación predictiva utilizando la siguiente estrategia: usar el flujo óptico como entrada en el modelo (Pre-Encoder, PE), estimarlo en el espacio latente tras el codificador (Post-Latent Space, PLS) o bien incorporarlo como un término regulador en la función de pérdida (ACCLIP). El objetivo general fue estudiar hasta qué punto la integración del flujo óptico mejora la capacidad de predicción de trayectorias de objetos en relación con un observador móvil, y cómo influye en la fidelidad visual y temporal de las secuencias generadas. La evaluación experimental se llevó a cabo en los conjuntos de datos KITTI, KTH–Action y Caltech Pedestrian, abarcando diferentes tipos de escenas y dinámicas. Esta investigación demostró que la incorporación explícita del flujo óptico, ya sea como señal de entrada o como supervisión, es clave para preservar la coherencia estructural y temporal en la predicción de videos. A partir de estos hallazgos, se proponen posibles mejoras arquitectónicas, como la inclusión de mecanismos de atención o módulos de refinamiento dinámico, y se identificaron desafíos abiertos relacionados con la exploración de nuevas redes y métodos de cálculo de flujo óptico , así como también la reducción de la latencia para poder generar predicciones en tiempo real.

3. Abstract

The ability to anticipate how objects will move in a video sequence has become one of the major challenges in computer vision, as it is essential for applications such as autonomous navigation, surveillance, and robotics, where the anticipation of future scenarios is crucial for decision-making. Nevertheless, despite the progress achieved, current methods still present significant limitations. For example, convolutional architectures and their 3D variants show good performance in short- and medium-term horizons, but lose motion coherence as the number of predictions increases. On the other hand, recurrent models and those based on interpolation provide greater temporal consistency, but require high computational resources and face difficulties in adapting to complex real-world scenarios.

Within this context, optical flow emerges as a key tool because it offers an explicit description of the apparent motion in the scene while reducing computational cost. However, in practice, this resource has been integrated only in a limited way within prediction architectures, which reduces its potential to ensure stability and coherence over longer time horizons. This raised a central question that guided the present work: at which point in an architecture is it most effective to incorporate optical flow in order to obtain reliable predictions in urban environments?

To address this question, three models inspired by predictive coding were explored using the following strategy: employing optical flow as input to the model (Pre-Encoder, PE), estimating it in the latent space after the encoder (Post-Latent Space, PLS), or incorporating it as a regularization term in the loss function (ACCLIP). The overall objective was to study to what extent the integration of optical flow improves the ability to predict object trajectories with respect to a moving observer, and how it influences the visual and temporal fidelity of the generated sequences.

The experimental evaluation was conducted on the KITTI, KTH–Action, and Caltech Pedestrian datasets, covering different types of scenes and dynamics.

This research demonstrated that the explicit incorporation of optical flow, whether as an input signal or as supervision, is key to preserving structural and temporal coherence in video prediction. Based on these findings, possible architectural improvements are proposed, such as the inclusion of attention mechanisms or dynamic refinement modules, and open challenges were identified related to the exploration of new networks and optical flow estimation methods, as well as reducing latency to enable real-time predictions.

4. Marco Teórico

En este capítulo se abordarán los principios teóricos de la codificación predictiva para la comprensión del procesamiento sensorial y los mecanismos mediante los cuales el Sistema Nervioso Central (SNC) anticipa estímulos sensoriales entrantes, contrastando sus predicciones con la información real percibida, y cómo dichos principios han sido trasladados al campo del aprendizaje profundo a través de modelos computacionales orientados a la predicción de secuencias de imágenes como las redes convolucionales (CNN) y redes neuronales recurrentes (RNN), así como modelos híbridos que integran ambas técnicas.

Además, se revisará el concepto de flujo óptico, entendido como una herramienta clave para representar el movimiento en las secuencias visuales. Se analizarán tanto sus fundamentos clásicos como su integración moderna en modelos de visión computacional, particularmente en aquellos que requieren una comprensión profunda de las dinámicas espaciotemporales.

Este capítulo no solo busca proporcionar una base teórica sólida para la comprensión del modelo propuesto, sino también explicar las decisiones metodológicas adoptadas desde una perspectiva interdisciplinaria, que combina principios de neurociencia computacional con herramientas de aprendizaje profundo y procesamiento visual.

4.1. Fundamentos de la teoría de la codificación predictiva

4.1.1. Origen neurocientífico y adaptación a modelos computacionales

Fundamentos neurocientíficos.

Hace casi 20 años, Jeff Hawkins y Sandra Blakeslee plantearon que la capacidad de reconocer patrones de nuestro entorno y anticipar lo que podría suceder es un aspecto esencial de lo que entendemos por inteligencia humana y también un mecanismo clave para la toma de decisiones [1]. A partir de esto, surge la teoría de la codificación predictiva, postulando que la corteza cerebral se organiza en capas que se encuentran ordenadas jerárquicamente y en el que cada nivel superior formula una hipótesis sobre la actividad del nivel inferior. Así, el sistema aprende y ajusta sus representaciones internas, volviéndose progresivamente cada vez más eficiente a la hora de anticipar la nueva información [2].

Dentro de este marco jerárquico, el cerebro compara constantemente esas hipótesis generadas con la información real que recibe para identificar diferencias entre ambas; Estas diferencias reciben el nombre de error de predicción que retroalimentan el aprendizaje en las capas superiores de la corteza, de modo que las predicciones futuras se ajusten y tengan un margen de error mínimo al enfrentarse a estímulos similares [3] [4].

Este modelo se sostiene mediante dos mecanismos: por un lado, a través de circuitos que procesan y ajustan las predicciones Figura 1a [5] [6] [7] y, por otro, por neuronas que regulan la intensidad de esos errores en función de la relevancia del estímulo [8] . De igual forma, la plasticidad de las conexiones neuronales mejora progresivamente la precisión de las predicciones permitiendo al sistema aprender y comprender mejor el mundo que lo rodea. [9] [10].

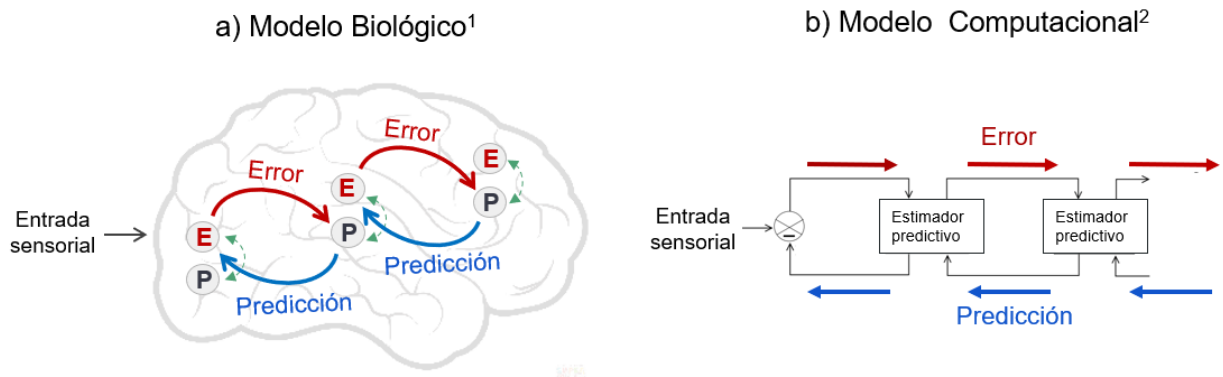


Figura 1: Modelos biológico (a) [11] y computacional (b) [12] de la teoría de la codificación predictiva : La información sensorial se procesa de forma jerárquica. En cada nivel coexisten dos módulos: un generador de predicción (flechas azules) y un estimador de error (flechas rojas). En el flujo ascendente (error), los estimadores calculan la discrepancia entre la señal real y la predicha y propagan ese error hacia los niveles superiores. Mientras que en el flujo descendente (predicción), las unidades envían predicciones hacia el nivel inmediatamente inferior, anticipando su actividad. Este ciclo descendente-ascendente se repite hasta que el error de cada capa se minimiza, logrando que las predicciones sucesivas se ajusten lo más posible a la entrada sensorial.

Conexión con señales retinianas de movimiento.

En el sistema visual, las diferencias entre lo anticipado y lo observado se inician en la retina: las células ganglionares direccionalmente selectivas y otros circuitos locales generan señales que anticipan el movimiento [13] [14] [15]. Estas respuestas tempranas pueden entenderse como un ejemplo de codificación predictiva, pues permiten al sistema reducir la incertidumbre sobre lo que ocurrirá en la escena visual [16] [17]. Así, la retina no se limita a transmitir información pasiva, sino que aporta señales con valor predictivo que luego serán integradas en niveles corticales superiores.

Adaptación a modelos de aprendizaje profundo.

A partir de esta teoría, se han desarrollado modelos de redes neuronales artificiales basadas en aprendizaje profundo que intentan replicar la forma en que el cerebro organiza jerárquicamente las predicciones y corrige sus errores de forma interna, sus pioneros fueron Rao y Ballard en 1999 [12] quienes propusieron un modelo jerárquico en el que cada capa predice la actividad de la anterior y

ajusta sus conexiones según el error de predicción (figura 1b). Luego, Friston en 2005 [10] mostró cómo reducir la energía libre, es decir, la medida de discrepancia entre los estados internos y los datos sensoriales recibidos mediante un método bayesiano de actualización constante, planteando normas basadas únicamente en la predicción local y la entrada real para ajustar los pesos de la red. En este sentido, la minimización de la energía libre equivale a reducir el error de predicción entre lo anticipado y lo percibido.

Posteriormente, Spratling en 2008 reformuló la idea de codificación predictiva global demostrando que un mecanismo local de inhibición de entradas es matemáticamente equivalente al cálculo de error y predicción propuesto inicialmente por Rao y Ballard. En lugar de requerir una única señal de error global que recorra toda la jerarquía cortical, Spratling presenta un método que compara su propia predicción con la entrada real, manteniendo la misma dinámica de ciclo de predicción y corrección pero con un esquema biológicamente más plausible. De esta manera, su propuesta unificó ambos marcos teóricos bajo un mismo andamiaje computacional en el que la corrección ocurre de forma autónoma en cada unidad de procesamiento, eliminando con esto la necesidad de un nodo centralizado de error [18].

Estos avances en modelos computacionales han abierto el camino para su aplicación directa a la predicción de secuencias de video. El reto pasó entonces buscar métodos que permitan lograr predicciones consistentes en horizontes temporales cortos a mediano plazo. Es decir, hoy en día, este enfoque ya no se limita a anticipar un estímulo aislado, sino que busca generar secuencias completas que conserven coherencia a lo largo del tiempo.

En este escenario, es relevante preguntarse si los principios de la codificación predictiva como la integración jerárquica de señales y la transmisión de errores de predicción pueden adaptarse a arquitecturas basadas en aprendizaje profundo. Esta reflexión constituye la motivación central de esta investigación, cuyo propósito es comprobar si dichos modelos son capaces de modelar relaciones espacio–temporales de mayor amplitud.

4.2. Codificacion predictiva en procesamiento de imágenes en videos

Como se vió en la sección 4.1, el paradigma de la codificación predictiva propone que el cerebro construye representaciones internas del entorno mediante la observación y la interacción, ajustando constantemente sus predicciones al compararlas con la realidad sensorial.

Inspirados en estos principios, han surgido modelos de redes neuronales artificiales de aprendizaje profundo cuyo objetivo es predecir las imágenes subsecuentes a partir de una secuencia de imágenes de vídeos como contexto, de tal forma que se genere una representación continua lo más similar posible a las imágenes reales futuras. Para lograrlo, se han desarrollado diversos modelos de predicción de imágenes de videos basados en distintos enfoques de aprendizaje profundo, tales como redes neuronales convolucionales (CNN), redes neuronales recurrentes (RNN) y modelos

generativos, los cuales se abordarán a continuación.

4.2.1. Modelos Convolucionales

Las Redes Neuronales Convolucionales (CNN) son la base de las arquitecturas de aprendizaje profundo enfocado en visión por computadora, ya que se encuentran particularmente diseñadas para el procesamiento visual, modelando con eficiencia la estructura espacial de las imágenes [19]

Sin embargo, como las operaciones convolucionales solo capturan relaciones espaciales de corto alcance, su desempeño se ve limitado por las dependencias entre fotogramas, ya que su campo receptivo se encuentra determinado por el tamaño del kernel, es decir, la matriz de filtro que se desliza sobre la imagen para extraer las características locales.

Para mitigar estas limitaciones, muchos autores han propuesto diversos métodos como:

- Apilar más capas convolucionales [20], sin embargo, esta estrategia exige enormes volúmenes de datos manualmente etiquetados y no siempre ofrece mejoras apreciables para la restauracion visual.

- Aumentar el tamaño del kernel, aunque esto genera un costo computacionalmente mas elevado.

- Usar convoluciones capaces de captar dependencias espaciales de largo alcance [21]. No obstante, como las CNN tradicionales tienden a reducir progresivamente la resolución de la imagen hasta generar mapas de características pequeños, descartan gran parte de la información espacial fina, perjudicando tanto la precisión en la clasificación como la capacidad de abordar tareas que requieren entender detalles de la escena.

Aunque estas soluciones podrían mitigarse usando conexiones residuales, no son capaces de generar mejoras en cuanto a la modelación de dependencias temporales a largo plazo. En base a esto, se han explorado convoluciones 3D buscando garantizar la consistencia temporal [22] [23] [24]. Al aplicar convoluciones 3D, se espera que la red pueda aprender transformaciones a lo largo del tiempo actuando como un estado recurrente [25].

En relacion a los modelos de convoluciones 2D y 3D, en esta última década se han propuesto varias arquitecturas basadas en CNN para enfrentar la tarea de anticipar imágenes en secuencias de vídeo, entre las más relevantes encontramos.

- **Multi-Scale AdvGDL** [26] Esta arquitectura sentó las bases para las CNN predictivas. Los autores propusieron una arquitectura multiescala que genera predicciones graduales desde baja hasta alta resolución; además, introdujeron por primera vez el concepto de entrenamiento adversarial en predicción de vídeo, que consiste en que una red discriminativa guía

al generador para producir imágenes más nítidas y coherentes. Por último, plantearon la pérdida por diferencia de gradientes de imagen, que penaliza discrepancias en los gradientes entre predicción e imagen real para preservar bordes y detalles finos. De esta forma, lograron generar predicciones coherentes en diferentes niveles de resolución y reducir artefactos locales.

- **PredCNN** [27] : Utiliza una arquitectura totalmente convolucional para la predicción de vídeos, facilitandp la propagación del gradiente y permitiendo una optimización completamente paralelizada. Para ello, introduce un nuevo modelo en bloque que mantiene una estructura jerárquica compuesta de unidades apiladas en cascada, lo que amplía su campo receptivo temporal, permitiendo capturar explícitamente las dependencias secuenciales sin recurrir a transiciones estado a estado. De esta forma, demuestra que es posible optimizar la extracción de características temporales directamente con convoluciones 2D.

- **SimVP** [28] : Demuestra que una arquitectura CNN simplificada inspirada en Vision Transformers para extraer *tokens* continuos puede igualar el rendimiento de modelos más complejos en predicción de vídeo. Este modelo introduce un traductor de dinámica temporal compuesto por módulos que permiten procesar información a múltiples escalas en paralelo, procesando los canales temporales como si fuesen dimensiones espaciales, aprendiendo transformaciones de largo alcance sin recurrir a estados recurrentes. Además, gracias a su simplicidad, SimVP reduce drásticamente el tiempo de entrenamiento. La versión 2 de este enfoque, **SimVPv2** [29], añade un módulo de atención espaciotemporal enmascarada que simplifica aún más la arquitectura, reduce el coste computacional y aumenta la velocidad de inferencia ademas de aumentar la precisión de las predicciones en métricas estándar.

En síntesis, aunque las CNN y sus variantes 3D han demostrado ser eficaces al capturar la estructura espacial y cierta consistencia temporal en secuencias de vídeo, presentan dos problemas fundamentales: por un lado, requieren un gran número de parámetros y de datos etiquetados para entrenar la red, y por otro, pierden coherencia a la hora de capturar dependencias temporales a largo plazo .

Para resolver estos problemas y explotar de la continuidad temporal de imágenes en vídeos, en lugar de ampliar indefinidamente la profundidad de la red o el tamaño de los *kernels*, resulta natural recurrir a arquitecturas creadas para datos secuenciales temporales como son las redes neuronales recurrentes Redes Neuronales Recurrentes (RNN) que se abordarán en la seccion seguiente

4.2.2. Modelos de Redes Neuronales Recurrentes en predicción de videos

Se diseñaron específicamente para capturar representaciones espacio temporales de datos secuenciales [30]. Procesan secuencias manteniendo un estado interno que se actualiza en cada paso

temporal, lo que les permite modelar dependencias entre elementos consecutivos de la serie de tiempo [31].

Entre las arquitecturas derivadas de las RNN encontramos dos arquitecturas principales: (1) *Long Short-Term Memory* LSTM [32] que añade una celda de memoria y tres puertas (de entrada, olvido y salida) que regulan de forma explícita qué información conservar, qué olvidar y qué exponer en cada paso. Gracias a este mecanismo, las LSTM capturan con eficacia dependencias a largo plazo y evitan gran parte de los problemas de gradiente que afectan a las RNN estándar [33] y (2) *Gated Recurrent Unit* GRU [34] que es un tipo de red que simplifica la arquitectura de las LSTM al combinar las puertas de entrada y de olvido en una sola puerta de actualización, y al incorporar una puerta de reinicio que controla cómo se integra la nueva información con el estado anterior [35]. Gracias a este diseño más compacto, las GRU capturan dependencias a largo plazo con menos parámetros y suelen entrenar más rápido que las LSTM.

Enfocadas en estos modelos recurrentes y sus variantes híbidras que rescatan las ventajas tanto de las CNN como las RNN es posible distinguir los siguientes modelos de predicción:

- **ConvLSTM (2015)** : emergió como un modelo clave que implemento un modelo híbrido entre una red CNN con una **ConvLSTM!** (**ConvLSTM!**) para procesar eficazmente las características de los fotogramas y capturando con precisión el movimiento y las dinámicas espaciotemporales, lo que ha influido notablemente en desarrollos de modelos posteriores mencionados a continuación.

- **PredNet (2017):** [36] modelo pionero en la exploración de redes neuronales recurrentes para la síntesis de vídeo. Tomando como inspiracion la teoría de la codificación predictiva cerebral, implementó el paradigma de codificación predictiva mediante una red convolucional recurrente organizada en capas jerárquicas que propaga el error de predicción como señal *feedforward*, es decir, una señal que se transmite únicamente en dirección ascendente, desde las capas inferiores hacia las superiores, sin retroalimentación interna, llevando la información del error hacia las capas superiores para su procesamiento. Esta red usa ConvLSTM para generar las predicciones *feedback*, esto es, una señal descendente que viaja desde las capas superiores hacia las inferiores, transportando la respuesta predicha para contrastarla con la entrada real y ajustar el modelo. De esta forma, PredNet reproduce el ciclo de hipótesis y comprobación del cerebro en una arquitectura completamente diferenciable. Sin embargo, si bien PredNet presenta una mejora en el rendimiento en datos sintéticos, no escala a escenarios reales complejos, además presenta dificultades en capturar secuencias a largo plazo.

- **PredRNN (2017):** [37] introdujo mejoras significativas al modificar la arquitectura LSTM con una estructura de doble memoria, con el fin de fortalecer el modelado espaciotemporal.

A pesar de estos avances, el modelo aún enfrenta desafíos como el desvanecimiento del gradiente en tareas de síntesis de vídeo

- **E3D-LSTM (2019): [22]** avanzó aún más incorporando convoluciones 3D en las RNN y un módulo de autoatención controlado por compuertas, mejorando sustancialmente las capacidades de síntesis de imágenes a largo plazo; sin embargo, la complejidad computacional añadida por las convoluciones 3D atenua esta ganancia ya que requiere de mucho tiempo de cómputo para procesarlas.

- **MSPred (2022):** Propone una red jerárquica convolucional recurrente que opera en múltiples frecuencias temporales para predecir fotogramas futuros, así como representaciones adicionales como poses y semántica. Sin embargo, requiere gran cantidad de datos y recursos para manejar las distintas frecuencias, lo que genera complejidad en el ajuste de las predicciones. [38]

A pesar de su eficacia para capturar dependencias temporales, las redes recurrentes enfrentan varios retos en las tareas de predicción de vídeo. Su naturaleza inherentemente secuencial, que permite modelar imagen a imagen a lo largo del tiempo, puede traducirse en una elevada complejidad computacional, especialmente en escenarios de alta resolución [39]. Además, las RNN son susceptibles al problema de desvanecimiento de gradientes, lo cual puede obstaculizar severamente su capacidad para aprender dependencias a largo plazo.

4.2.3. Modelos de predicción de imágenes basadas en flujo óptico

En contraste con los enfoques puramente convolucionales o recurrentes, los métodos de síntesis e interpolación basados en flujo óptico aprovechan estimaciones de movimiento para guiar la generación directa del siguiente fotograma. En estos modelos, técnicas de cálculo de flujo óptico como PWC-Net [40] o RAFT [41] son los más utilizados. De esta manera se reformula el problema de predicción de imágenes a un problema condicionado por el desplazamiento aparente de cada pixel entre dos imágenes consecutivas [42]. Entre los modelos más relevantes en esta categoría encontramos a:

- **FVS** (2020) [43]: Este enfoque desacopla la síntesis de fondos y objetos en movimiento, prediciendo por separado el flujo de la escena estática y las trayectorias de los objetos en movimiento. Para la predicción del fondo estático, utiliza PWC-Net como base para estimar el campo de flujo óptico entre el último fotograma observado y cada uno de los futuros; en cuanto a los objetos en movimiento, genera un módulo de predicción de trayectorias utilizando una CNN que predice parámetros como traslación y rotación; finalmente se combinan ambos módulos para obtener una síntesis donde fondo y objetos que se modelan y

actualizan de forma totalmente independiente. Si bien FSV presenta grandes avances en métodos de predicción de flujo óptico, presenta dificulades para generar predicciones en horizontes temporales grandes, y por sobre todo, presenta dificultades para estimar flujos en secuencias con movimientos rápidos como los obtenidos en un escerario real. Además, presenta problemas de oclusion y rellenado espacios vacíos con pixeles generando artefactos que degradan la calidad de las trayectorias estimadas.

- **OPT** (2022) [44]: Introduce el primer marco de optimización para predicción de vídeo que, en lugar de entrenar una red en un gran conjunto de datos, plantea la extrapolación de fotogramas futuros como un problema de optimización sobre un módulo diferenciable de interpolación de vídeo. Utilizando RAFT, estima el flujo óptico que luego es usado como optimizador para generar las etiquetas de inicialización, y luego utilizan RIFE [45] para generar estimaciones de flujo en tiempo real para interpolación de fotogramas de video. Como resultado, consigue generar imágenes realistas tanto en predicciones a corto como a mediano plazo. Sin embargo, presenta un problema de optimización, ya que para generar las predicciones de imágenes necesita generar decenas de miles de iteraciones de retropropagación a través de la red de interpolación, lo que se traduce en un coste computacional elevado y tiempos de inferencia muy superiores a los de los métodos entrenados de forma convencional.

En definitiva, los modelos de interpolación basados en flujo óptico combinan lo mejor de dos mundos: por un lado, aprovechan estimaciones de movimiento para guiar la generación de imágenes futuras, incrementando la coherencia entre objetos en movimiento y fondo reduciendo la creación de artefactos que deforman la imagen. Por otro lado, al formular la prediccion de la imagen futura como un problema diferenciable, permite entrenar todo el sistema de forma *end-to-end*, es decir, entrena el modelo de principio a fin. Pese a sus ventajas, estos modelos presentan limitaciones notables, ya que tal como reportan los mismos autores, dependen de estimadores de flujo y módulos de interpolación cuyo costo crece linealmente con el número de pasos futuros a predecir, sin garantías de establidad ni de calidad visual a largo plazo, además, no están diseñado para producir varias imágenes futuras en instantes sucesivos, sino que cada horizonte temporal requiere un nuevo cálculo completo del flujo óptico.

4.3. Flujo óptico

4.3.1. Flujo óptico retiniano y su integración en corteza visual

En la retina el flujo óptico puede entenderse como el conjunto de movimientos aparentes que describen cómo los puntos del entorno se desplazan sobre la superficie cuando el observador o los

objetos de la escena cambian de posición. Si bien la retina no genera un cálculo de flujo óptico como lo hacen los algoritmos de visión artificial, sí es capaz de detectar señales locales de movimiento, principalmente a través de las células ganglionares direccionalmente selectivas (DSGCs) [46] [47]. A su vez, regiones superiores del sistema visual combinan estas señales en representaciones globales de flujo óptico para percibir el movimiento propio y orientar la navegación [47] [48] [49].

4.3.2. Flujo óptico en visión artificial

El modelo de flujo óptico fue propuesto en visión artificial para describir el estímulo visual que permite a los seres vivos desplazarse por su entorno. Este modelo caracteriza una distribución de velocidades generada por un patrón de movimiento aparente de los objetos, lo cual facilita no solo tener una percepción del movimiento de los objetos, sino también percibir la forma y la distancia a la que se encuentran. De este modo, el flujo óptico se define como un campo vectorial que asigna a cada píxel (x, y) de un fotograma un desplazamiento bidimensional (u, v) proporcional al movimiento aparente de la escena entre dos instantes consecutivos [50]

La estimación del flujo óptico parte de la hipótesis de constancia de brillo: la luminancia de un punto permanece invariable durante intervalos de tiempo muy cortos. Ello conduce a la ecuación de conservación de la intensidad

$$I_x\, u + I_y\, v + I_t = 0 \tag{1}$$

La ecuación (1) relaciona los gradientes espaciales (I_x, I_y) y temporal (I_t) de la imagen con el vector de movimiento (u, v), a partir de esto, se puede inferir que la variación de los puntos (u, v) corresponden a zonas de alto gradiente espacial donde un pequeño desplazamiento genera un cambio en la posicion registrada de la intensidad, proporcionando una señal temporal que permite calcular el flujo óptico.

4.3.3. Métodos clásicos de cálculo de flujo óptico

Los métodos clásicos de flujo óptico se fundamentan en dos grandes paradigmas: los enfoques globales, que estiman un campo de flujo continuo en toda la imagen mediante criterios de suavidad, y los métodos locales, que resuelven el movimiento en pequeñas ventanas de píxeles. A continuación se presentan tres de los algoritmos más representativos:

- **Horn–Schunck** [50]: asocia el campo de flujo más suave que satisfaga la ecuación de invarianza de brillo a lo largo de toda la imagen. Este método minimiza la función de error global mediante una penalización de norma L^2 sobre la divergencia del flujo ofreciendo estructuras mas detalladas pero a un gran costo computacional.

- **Lucas–Kanade** [51]: estima localmente el flujo óptico resolviendo en ventanas pequeñas un sistema sobredeterminado basado en derivadas espaciales y temporales, este metodo es altamente eficiente para el seguimiento de esquinas y texturas.

- **Farnebäck** [52]: utiliza aproximaciones polinomiales para describir la vecindad de cada píxel y modelar los cambios de intensidad, de esta forma calcula el vector de flujo óptico comparando los polinomios entre fotogramas al derivar el desplazamiento con alta densidad.

Los métodos de flujo óptico basados en aprendizaje profundo sustituyen la formulación analítica clásica por redes neuronales que aprenden a inferir directamente el campo de movimiento a partir de grandes colecciones de datos. Pueden agruparse, de manera general, en tres familias complementarias:

- **Modelos supervisados tipo encoder–decoder:** que tratan el flujo como un problema de regresión a nivel denso; emplean convoluciones jerárquicas para extraer características y generan la predicción en una única pasada. Entre los modelos que utilizan este método encontramos a FlowNet [53] y FlowNet2 [54].

- **Redes piramidales con cost volume y warping:** refinan la estimación de correlaciones de características y mecanismos de alineamiento espacial. Su arquitectura consiste en construir una pirámide de características para analizar el movimiento y luego desplazan (*warping*) las características de la segunda imagen usando el flujo estimado en el nivel superior, alineándolas con las de la primera; en segundo lugar, generan un tensor de correlaciones locales (*cost volume*) que explicita las correspondencias tras el alineamiento, lo que permite refinar la estimación a cada escala con kernels pequeños, y así, alcanzar un mejor equilibrio entre precisión y coste computacional. Entre varios modelos de esta familia destacamos:

    - **PWC-Net** [40]: Aplica una Pirámide de características, un desplazamiento (*Warping*) progresivo y un *Cost Volume* local para, a cada escala, alinear las dos imágenes y refinar el flujo residual, alcanzando así alta precisión con un modelo compacto y rápido. Este método se entrena de forma supervisada con datasets sintéticos optimizando una función de pérdida combinada entre ℓ_1 + suavidad.
    - **SelFlow** [55]: retoma la arquitectura piramidal de PWC-Net, pero introduce un entrenamiento *auto-supervisado* mediante:
        1. Una función de pérdida multi-escala, combinando ℓ_1 y SSIM para comparar la imagen original con la reproyección de la segunda usando el flujo estimado.
        2. Un módulo de detección de oclusiones que descarta píxeles no fiables mediante consistencia bidireccional *(forward–backward)*.

En benchmarks como Sintel y KITTI, SelFlow reduce el EPE en aproximadamente un 5 % respecto a PWC-Net, especialmente en zonas con oclusiones y bordes de objetos.

- **Métodos de refinamiento recurrente o basados en Transformers:** corresponden a modelos que realizan múltiples iteraciones de actualización o usan atención global para capturar relaciones de largo alcance, alcanzando el estado del arte en benchmarks realistas, como RAFT [41] y GMFlow [56].

4.3.4. Flujo óptico en modelos de predicción

Como se mencionó anteriormente, el flujo óptico captura el movimiento aparente de los píxeles entre fotogramas consecutivos; en el contexto de predicción de vídeo, algunos métodos utilizan estos vectores para sintetizar directamente el siguiente fotograma, mientras que otros predicen el mapa de flujo futuro de forma explícita. Entre los modelos de predicción de flujo óptico encontramos:

- **OFNet** (2022) [57] : emplea una arquitectura encoder–ConvLSTM–decoder basada en UNet, donde los últimos T mapas de flujo se codifican espacialmente, luego se procesan temporalmente mediante un bloque ConvLSTM y finalmente se decodifican para generar el flujo futuro de forma autorregresiva. Gracias a este diseño puede capturar dinámicas temporales complejas y producir predicciones de hasta 3 flujos futuros. Sin embargo, los autores señalan que OFNet incurre en un coste computacional significativamente alto ya que el apilamiento de múltiples capas de ConvLSTM necesarias para modelar dependencias a largo plazo incrementa la demanda de memoria y complica la convergencia durante el entrenamiento.

- **MemFlow** (2024) [58]: presenta una arquitectura de estimación de memoria de flujo utilizando varios bloques GRUs con la que extrae un contexto y su movimiento, luego la predicción de flujo se realiza utilziando CNNs en las que integra la salida de la GRU con el flujo pasado, permitiendo predecir 1 fotograma futuro. Este método alcanza un equilibrio entre precisión y eficiencia computacional superando con creces el estado del arte en generalización cruzada. Entre sus limitaciones, los autores declaran que los enfoques de dos fotogramas no aprovechan la coherencia temporal más allá del par actual y dependen de codificadores muy costosos; ademas, al extender la memoria más allá de un solo paso no mejora el rendimiento e incluso lo degrada, lo que apunta a desafíos en la gestión del ruidos de estados muy antiguos. Por último, presenta problemas de escalabilidad de resolución, ya que sin el re-escalado adaptativo propuesto, la atención basada en memoria pierde eficacia al cambiar la resolución de entrada, degradando así la generalización del modelo.

La estimación de flujo óptico sigue siendo un tema de investigación activo y ampliamente estudiado [41] [59] [60] [58]. Sin embargo, aun presentan importantes limitaciones, muchas de las cuales tiene su origen en que los modelos de flujo óptico clásicos son entrenados con datos sintéticos y suelen diferir significativamente de los escenarios reales. Xue et al. señala que para tareas complejas es necesario entrenar desde cero una red de estimación de flujo y no basarse en modelos pre entrenados. Por otro lado, redes de flujo óptico de mayor rendimiento pueden dar lugar a peores resultados en la síntesis de imágenes reales, ya que tienden a enfatizar regiones ambiguas como oclusiones y pueden carecer de suficiente resolución espacial [61] [45].

En escenarios del mundo real, obtener etiquetas de flujo óptico *ground truth* sigue siendo un gran desafío. De igual forma, generar metodos predictivos de flujo óptico basado en imágenes del mundo real podría lograrse conforme mejoren los métodos de cálculo de flujo óptico, no obstante, integrar dichos métodos, ya sean métodos clásicos o los que estan por venir en la generación de vídeo a largo plazo sigue siendo un reto importante [62]

En definitiva, dado que el flujo óptico permite generar una representación del movimiento en la secuencia de vídeo al capturar la variación de las posiciones de los píxeles a lo largo del tiempo, utilizar métodos basados en flujo optico permitiría guiar el entrenamiento de modelos de predicción, incorporando términos de pérdida que penalizan desplazamientos inconsistentes entre fotogramas.

4.4. Función de pérdida

Una función de pérdida es un mapeo que asigna un coste $\mathcal{L}(y, y')$ al hecho de predecir y' cuando el valor verdadero es y. En definitiva, sirve como señal de error para ajustar los parámetros del modelo mediante optimización.

Por lo tanto, para mejorar la precisión de los modelos de predicción de vídeo es fundamental no solo optimizar las arquitecturas de la red, sino también diseñar estrategias de entrenamiento que mitiguen la acumulación de errores en horizontes temporales largos. En este sentido, la elección y combinación de las funciones de pérdida juega un papel decisivo al momento de guiar el aprendizaje e inducir al modelo a priorizar distintos aspectos que mejoren la calidad de la predicción.

4.4.1. Funciones de pérdida pixel-wise

Las funciones de pérdida más habituales en predicción de vídeo se basan en errores *pixel-wise*, que penalizan directamente la discrepancia entre los píxeles de las imágenes predichas y las reales, entre los más utilizados encontramos:

- **Norma L_2 / MSE**: La norma L_2 o Error Cuadrático Medio (MSE, por sus siglas en ingés) se define como la media de los errores al cuadrado entre los píxeles predichos y y los reales

x, quedando definida por la ecuación [63]:

$$\mathcal{L}_2(x, y) = \|x - y\|^2 \tag{2}$$

Es la métrica más habitual en tareas de regresión y predicción de vídeo bajo entornos controlados o sintéticos , dado que promueve un ajuste medio óptimo de las predicciones a los datos reales [64] . Es muy popular por su suavidad y propiedades de optimización, aunque tiende a generar predicciones de baja nitidez en escenarios reales [65].

- **Norma L_1 / MAE**: también conocida como Error Absoluto Medio (MAE, por sus siglas in inlés) se define como la suma de las diferencias absolutas entre los píxeles predichos y y los reales x, se define como:

$$\mathcal{L}_1(x, y) \;=\; \frac{1}{N}\sum_{i=1}^{N}\big|x_i - y_i\big|. \tag{3}$$

Donde N es el número total de píxeles de la imagen, entre sus ventajas podemos encontrar que:

- Al penalizar la magnitud de las discrepancias de manera lineal, es menos sensible a valores atípicos que la MSE, lo que favorece reconstrucciones con bordes más definidos. [66]

- Se emplea con frecuencia en tareas de restauración de imágenes y predicción de vídeo cuando la conservación de detalles finos y la robustez frente a abruptos cambios de intensidad son prioritarios [65].

Si bien estas pérdidas funcionan bien en entornos controlados o sintéticos, en vídeos naturales su uso exclusivo suele derivar en secuencias con falta de nitidez y realismo dinámico.

Para contrarrestar las limitaciones de estas funciones de pérdida, se han explorado funciones de pérdida híbridas que son producto de la combinación de dos o más funciones reguladas por hiperparámetros que permiten dar pesos de relevancia a cada función.

4.4.2. Funciones de pérdidas híbridas y perceptuales

Las funciones de pérdida híbridas, que combinan múltiples métricas simultáneamente, han surgido como una estrategia que da respuesta a ciertas limitaciones del entrenamiento tradicional, permitiendo de esta manera mejorar el entrenamiento de ANNs ya que este enfoque permite orientar el aprendizaje del modelo de manera más específica y alineada con los objetivos particulares de la tarea. Entre las funciones de pérdidas híbridas más utilizadas encontramos:

- **Pérdida Perceptual** [67]: mide la diferencia entre dos imágenes en el espacio de características extraídas por una red preentrenada [68], preservando detalles de estructura y textura [69]. De este modo, la *perceptual loss* captura diferencias semánticas y de textura que se alinean mejor con la percepción humana [70].

- **Pérdidas de integridad espaciotemporal**: es una función de pérdida que combina de forma lineal y ponderada varias funciones de pérdida simultáneamente, donde cada uno de ellas permite evalar un aspecto distinto de la calidad de la predicción [68], en términos matemáticos, esta función de pérdida se expresa como:

$$\mathcal{L} \;=\; \sum_{i=1}^{N} \alpha_i \, \mathcal{L}_i \tag{4}$$

  Donde $\mathcal{L}_i$ es el i-ésimo componente de la pérdida, por ejemplo: error pixel-a-pixel, similitud perceptual, error de flujo óptico, consistencia temporal, etc. Y α_i es el peso o ponderación asignada a ese componente, que regula su influencia relativa en el entrenamiento.

  Gracias a este esquema, el modelo puede equilibrar simultáneamente detalles estáticos, texturas y fluidez de movimiento, ajustando dinámicamente cada α_i para priorizar unos criterios sobre otros según la tarea concreta [26] [71].

- **Pérdidas basada en Índice de Similitud Estructural (SSIM)**: cuantifica la similitud entre dos imágenes considerando simultáneamente la luminancia, el contraste y la estructura local extraídas de la imagen predicha y la real, respectivamente. El índice SSIM se define como [72]:

$$\mathrm{SSIM}(x, y) = \frac{\big(2\mu_x\mu_y + C_1\big)\big(2\sigma_{xy} + C_2\big)}{\big(\mu_x^2 + \mu_y^2 + C_1\big)\big(\sigma_x^2 + \sigma_y^2 + C_2\big)} \tag{5}$$

  donde:

  - x representa a la imagen real e y a la imagen predicha.
  - μ_x y μ_y son las medias de intensidad de los pixeles
  - σ_x^2 y σ_y^2 son las varianzas
  - σ_{xy} es la covarianza entre las imágenes
  - C_1 y C_2 son constante que previenen inestabilidades en la división

  Para emplear el SSIM como función de pérdida, se invierte su valor y se promedia sobre todos los pares de imágenes:

$$\mathcal{L}_{\text{SSIM}} = \frac{1}{N} \sum \big[1 - \text{SSIM}(x, y)\big] \tag{6}$$

donde N corresponde al número de pares de imágenes predichas con las reales.

Al minimizar $\mathcal{L}_{\text{SSIM}}$, el modelo preserva mejor las estrucutras perceptuales relevantes, superando las limitaciones de las pérdida basadas puramente en *pixel-wise*.

Como se mencionó anteriormente, las funciones de pérdida híbridas pueden mejorar significativamente la calidad de las predicciones. Sin embargo, su implementación conlleva una mayor complejidad en el proceso de entrenamiento, ya que requiere ajustar varios hiperparámetros de forma simultánea. Además, al tener que ajustar múltiples pesos asociados a cada componente de la pérdida puede implicar un elevado costo computacional.

Considerando la complejidad que requieren las tareas de predicción de videos, donde intervienen múltiples dimensiones como el contenido visual que da un contexto, la dinámica temporal que determina el movimiento de los diferentes objetos en la escena y la coherencia estructural que combina ambos, se hace necesario explorar una función de pérdida híbrida cuyo diseño equilibre la *nitidez*, la *coherencia temporal* y la *estabilidad del entrenamiento*, ya que permiten combinar criterios complementarios y guiar el entrenamiento de los modelos hacia predicciones más acertadas, realistas y generalizables. Por esta razón, su exploración e incorporación resulta fundamental en el diseño de los modelos propuestos en este trabajo, con el fin de evaluar su impacto en la calidad de las predicciones y optimizar el rendimiento en escenarios complejos donde múltiples objetos se mueven simultánemante en diferentes direcciones.

La evidencia previa indica que la combinación de métricas a nivel de píxel, estructura y movimiento puede mejorar la calidad de las predicciones visuales. En esta tesis, dicha conclusión respalda el uso de funciones de pérdida híbridas que integran MSE, SSIM y flujo óptico, aspecto central en los modelos propuestos, cuyo propósito es reforzar la coherencia temporal sin alterar la arquitectura interna.

5. Visión general de los enfoques de codificación predictiva

En resumen, *los modelos puramente convolucionales* (PredNet, E3D-LSTM, SimVP) extraen características espaciales de cada fotograma mediante bloques de convoluciones 2D/3D y convierten la predicción de imágenes en una regresión directa sobre píxeles. Su principal ventaja es la eficiencia del cómputo en paralelo y la alta fidelidad local, pero carecen de mecanismos intrínsecos para capturar dependencias temporales de largo plazo y suelen requerir grandes cantidades de datos y parámetros.

Luego, los *modelos recurrentes* (PredRNN, ConvLSTM, MSPred) incorporan estados internos que memorizan el contexto pasado y modelan dinámicas secuenciales. Gracias a ello mejoran la coherencia en horizontes cortos y moderados, pero su naturaleza secuencial implica mayor latencia de inferencia, riesgo de desvanecimiento de gradientes en largas secuencias temporales y un coste computacional elevado.

Por último, las *estrategias basadas en flujo óptico* se dividen en dos subcategorías: 1. *Síntesis de fotogramas condicionada por flujo* (FVS, OPT), que usan estimadores como PWC-Net o RAFT para generar mapas de flujo internos y alimentar módulos diferenciables de *warping* o interpolación; ofrecen alta coherencia entre movimiento e imagen y calidad perceptual, pero están limitados a predicción de un paso secuencial a la vez, dependen de redes de interpolación costosas y escalan linealmente el coste al extender el horizonte de predicción. 2. *Predicción directa de flujo futuro* (MemFlow y OFNet), que generan predicciones directas de mapas de flujos ópticos, utilizan CNN y bloques ConvLSTM autoregresivos. Estas arquitecturas equilibran precisión de los vectores de flujos futuros y flexibilidad de horizonte (en especial OFNet), pero a costa de complejidad de entrenamiento, propagación de error y alto consumo de recursos.

En conjunto, el recorrido por el estado del arte muestra cómo cada familia de métodos aporta fortalezas como precisión espacial, memoria contextual y coherencia guiada por movimiento, pero también sus propias restricciones en términos de alcance temporal y eficiencia computacional. Estas observaciones motivan la exploración de arquitecturas inspiradas en mecanismos de predicción basados en la codificación predictiva cortical, con el fin de generar estimaciones confiables, capturar dependencias a largo plazo, generar predicciones confiables en escenas con movimiento de múltiples objetos, reducir el coste computacional de redes profundas y explotar verdaderamente la señal de movimiento que ofrece el flujo óptico. Estas limitaciones son las que motivan a generar las preguntas de investigación (Sección 6) que motivaron al desarrollo de esta investigación y a plantear los modelos (PE, PLS y ACCLIP) , cada uno diseñado para explorar u objetivo específico Finalmente, la validación de estos enfoques en escenarios de predicción de movimiento en entornos urbanos permitirá evaluar su aplicabilidad en el mundo real, respondiendo de manera integral a los objetivos de esta investigación.

Autor	Año	Aporte	Limitación
Rao & Ballard [12]	1999	Modelo jerárquico de codificación predictiva en la corteza visual.	No incorpora explícitamente dinámicas temporales.
Friston [10]	2005	Formaliza la minimización de la energía libre como inferencia bayesiana local.	Alto coste computacional y no gestiona directamente la temporalidad.
Spratling [18]	2008	Implementación biológicamente plausible con predicción y error locales, sin señal global.	Carece de extensión específica a secuencias de vídeo.
Lotter et al. (PredNet) [36]	2016	ConvLSTM para predicción de vídeo basado en bucles de codificación predictiva.	Escalabilidad limitada en escenarios reales y problemas de gradiente en largas secuencias.
Wu et al. (FVS) [43]	2020	Desacopla la síntesis de fondo (PWC-Net) y trayectorias de objetos (CNN) para sintetizar el siguiente fotograma usando flujos internos.	Salida limitada al fotograma futuro; depende de estimaciones de flujo costosas; predicción one-step.
Villar-Corrales et al. (MSPred) [38]	2022	Red jerárquica convolucional-recurrente que opera en múltiples frecuencias temporales para predecir fotogramas, poses y semántica.	Requiere gran cantidad de datos y recursos para manejar las distintas frecuencias; complejidad en el ajuste.
Gao et al. (SimVP) [28]	2022	Arquitectura CNN pura simplificada inspirada en ViT para extraer tokens continuos, ofreciendo alto rendimiento y eficiencia.	Enfoque determinista que no captura la incertidumbre futura y puede perder coherencia a largo plazo.
Wu et al. (OPT) [44]	2022	Reformula la predicción de vídeo como un problema de optimización e interpolación diferenciable (RAFT + RIFE) para generar el siguiente fotograma.	Optimización iterativa por fotograma muy costosa; predicción one-step; alta latencia.

Tabla 1: Tabla resumen de los modelos de codificación predictiva de imágenes futuras en el estado del arte.

6. Planteamiento del problema

La predicción a largo plazo de imágenes en secuencias de videos es clave para aplicaciones en vigilancia, robótica, navegación autónoma, entre otras, pero los métodos actuales presentan varios problemas que impiden un desempeño confiable en entornos reales, estos son:

- **Dependencias temporales limitadas:** Las arquitecturas convolucionales y sus extensiones 3D capturan bien horizontes temporales a corto y mediano plazo, pero fallan al anticipar dinámicas a largo plazo, es decir, desde 40 pasos futuros de predicción hacia el futuro.
- **Movimientos complejos y múltiples objetos:** Los modelos pierden eficacia al anticipar escenarios con varios objetos en movimiento y, más aún, en escenarios donde tanto el observador como los objetos se mueven simultáneamente en distintas direcciones.

Desde la perspectiva biológica, estas limitaciones discrepan con la eficiencia del sistema visual, el cual implementa principios de codificación predictiva para anticipar los estímulos entrantes y reducir la imprecisión en la percepción. En este enfoque, tanto la retina como la corteza visual no se limitan a reaccionar pasivamente a la información sensorial, sino que elaboran hipótesis sobre el futuro inmediato y ajustan su respuesta en función del error de predicción. De esta manera, la retina aporta señales tempranas de movimiento que pueden entenderse como un código predictivo inicial del flujo óptico y que luego en áreas superiores del sistema visual se integran estas señales en representaciones globales de flujo óptico, fundamentales para percibir el movimiento propio y orientar la navegación.

A partir de los problemas detectados y de esta motivación biológica, surgen las siguientes preguntas de investigación:

- ¿De qué manera un modelo inspirado en los principios de codificación predictiva cerebral puede capturar dependencias a largo plazo?
- ¿Cuáles son los componentes espaciotemporales más determinantes para mantener coherencia en horizontes temporales extendidos?
- ¿En qué punto de la arquitectura de la red y con qué técnica debería integrarse el flujo óptico para maximizar la calidad de las predicciones a largo plazo?
- ¿Hasta qué punto el uso de flujo óptico en la función de pérdida y/o en el diseño del modelo mejora la fidelidad y estabilidad de las predicciones?

Inspirada en el modelo biológico, esta tesis llevará estas preguntas al terreno computacional, incorporando el flujo óptico en distintos niveles de la arquitectura como un mecanismo explícito de predicción temporal, explorando el flujo óptico como señal de movimiento en tres niveles:

1. Como entrada en un codificador, para extraer la dinámica temporal aparente de movimiento de objetos y reducir el ruido de detalles irrelevantes.

2. Tras el codificador, explorando la posibilidad de calcular el flujo óptico directamente sobre las características extraídas, con el fin de verificar si dichas representaciones permiten estimar la dinámica de movimiento de manera más eficiente.

3. Como parámetro en la función de pérdida para dar coherencia temporal en la predicción, de tal forma que penalice desplazamientos inconsistentes y refuerce la continuidad entre imágenes predichas.

De esta forma, el flujo óptico servirá tanto para filtrar información útil en la entrada como para guiar el entrenamiento hacia predicciones que sean estables, realistas y coherentes a largo plazo.

7. Hipótesis y objetivos

7.1. Hipótesis

Una predicción robusta de la trayectoria de objetos en referencia a un observador móvil requiere de una codificación del flujo óptico para capturar la estructura espacio temporal de la escena.

7.2. Objetivo general

Estudiar la capacidad predictiva que agrega la codificación del flujo óptico al predecir la trayectoria de objetos en referencia de un observador movil

7.3. Objetivo específico

1. Determinar si un espacio latente que incluya el flujo óptico (arquitectura PE) permite capturar la estructura espacio-temporal de una escena.
2. Determinar si un flujo óptico derivado del espacio latente (arquitectura PLS) permite capturar la estructura espacio-temporal de una escena.
3. Determinar si un flujo óptico integrado en la función de pérdida (arquitectura ACCLIP) permite capturar la estructura espacio-temporal de una escena.
4. Evaluar la capacidad predictiva de las arquitecturas PE, PLS y ACCLIP.

8. Metodología

8.1. Componentes generales de los modelos de prediccion propuestos.

8.1.1. Notación general para la componente temporal y canales

En los modelos propuestos vamos a emplear exactamente la misma notación para el eje temporal y canales:

- $T_{\text{pasado}} = T$ **:** Número de pasos temporales anteriores (imágenes) que alimentan el modelo.
- $T_{\text{futuro}} = N$ **:** Número de pasos temporales que el modelo debe predecir.
- B **:** Representa el tamaño del *batch*, es decir, el número de muestras procesadas simultáneamente en una sola iteración.
- H y W : Son dimensiones espaciales correspondientes a la altura y el ancho respectivamente
- D **:** Número de canales de cada imagen (2 para flujo óptico, 3 para imagenes RGB y 1 para imágenes en escala de grises)
- $N_{\text{total}} = B \times T \times H \times W$: número total de vectores bidimensionales $\big(b, t, i, j\big)$ que recorren el batch y la secuencia temporal.
- Índice único $k \in \{1, 2, \ldots, N_{\text{total}}\}$: linealiza la tupla $\big(b, t, i, j\big)$, de modo que cada vector de flujo o píxel se identifique con un solo índice k. En adelante escribiremos $\mathbf{f}^{(k)}$ para indicar el mismo punto en el batch, el tiempo y la posición espacial.

Con esta notación, la forma genérica de los tensores de entrada y salida son respectivamente: $X_{\text{in}} \in \mathbb{R}^{B\times T\times D\times H\times W}$, $\quad X_{\text{out}} \in \mathbb{R}^{B\times N\times D\times H\times W}$.

8.1.2. ConvGRU: Gated Recurrent Unit Convolucional

La ConvGRU (Convolutional Gated Recurrent Unit) se encarga de integrar información temporal preservando la coherencia espacial de los mapas bidimensionales [73].

En cada paso temporal t, la ConvGRU recibe dos entradas de igual dimensión: la representación latente $x_t \in \mathbb{R}^{B\times D\times H\times W}$ y el estado oculto previo h_{t-1} que encapsula la memoria de las iteraciones previas. A partir de ellas, la celda actualiza sus puertas de actualización (z_t) que decide cuánta parte de la memoria anterior debe conservarse, y su puerta de reinicio (r_t) que regula cuánto de esa memoria debe reiniciarse antes de integrar la información nueva. Ambas puertas combinan de manera adaptativa la memoria pasada y los datos entrantes generando el

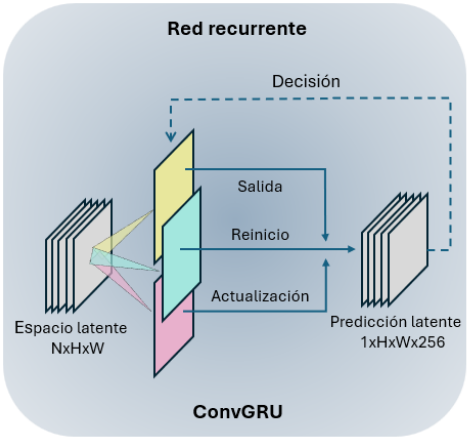


Figura 2: Esquema del bloque recurrente basado en ConvGRU integrado con el módulo de decisión autorregresiva: La celda ConvGRU recibe como entradas el estado oculto previo correspondiente al espacio latente obtenido del codificador y la representación latente, generando un nuevo estado oculto y una predicción en el espacio latente de dimensiones $[1 \times H \times W \times 256]$. El módulo de decisión descrito en la sección 8.1.3 compara la predicción con la imagen real para determinar cuál es la siguiente entrada a la red recurrente. Este mecanismo permite mitigar la acumulación de error y favorecer la estabilidad del modelo en horizontes largos de predicción.

nuevo estado oculto (h_t) , permitiendo con esto al modelo capturar patrones de movimiento a lo largo del tiempo sin sacrificar la resolución espacial.

En la implementación utilizada en este modelo, se apilan dos capas de ConvGRU con kernels de tamaño 3×3, $stride = 1$ y $padding = 1$. Además, para favorecer la generalización y mitigar el sobreajuste, se incorpora una tasa de $dropout = 0.1$ [74]. Esta técnica de regularización consiste en apagar neuronas de forma aleatoria en cada iteración durante el entrenamiento, de esta manera la red se ve forzada a distribuir mejor la información y no depender de co-adaptaciones entre neuronas que solo se activan en conjunto en determinados patrones del entrenamiento [75]. De este modo, el empleo de *dropout* en la ConvGRU permite controlar la complejidad del modelo sin comprometer la capacidad de capturar dependencias temporales a lo largo de la secuencia.

Luego, el estado oculto actualizado (h_t) se proyecta a un espacio latente de dimensión $[1 \times H \times W \times 256]$ que constituye la predicción latente. Esta prediccion es conectada finalmente con el módulo de decisión autorregresiva descrito en la Sección 8.1.3, encargado de determinar si la entrada en el siguiente paso temporal proviene de la imagen real *(ground truth)* o de la propia predicción del modelo. De este modo, la ConvGRU no solo sintetiza la dinámica temporal de la secuencia, sino que también interactúa con el mecanismo de decisión para mitigar la acumulación de error en horizontes temporales largos. Este método fue empleado en los tres modelos propuestos descritos en las Secciones 8.2, 8.3 y 8.4.

8.1.3. Módulo de Decisión Autorregresiva

El módulo de decisión autorregresiva tiene como propósito mitigar la acumulación de errores que se generan cuando un modelo genera secuencias de manera recurrente. Para ello se combinan dos criterios complementarios: un umbral de error adaptativo y un criterio de decisión basado en *scheduled sampling* [76].

1. **Umbral de error**: En cada paso temporal t, se calcula en el espacio latente el error cuadrático medio (MSE) entre la predicción $\hat{z}_t$ y el *ground truth* z_t

$$e(t) = \text{MSE}(\hat{z}_t, z_t), \tag{7}$$

   Este error se compara con un umbral U, estimado a partir de la media y desviación estándar de los errores en validación.

$$U = \mu_e + \sigma_e, \tag{8}$$

   - Si $e^{(t)} \leq U$, la predicción se considera confiable y se utiliza como entrada en el siguiente paso.
   - Si $e^{(t)} > U$, se activa el criterio de desición.

2. **Criterio de decición**: Cuando el error supera el umbral, el modelo aplica *scheduled sampling*, un mecanismo que asigna una probabilidad decreciente de usar el *ground truth* en lugar de la predicción como entrada a la ConvGRU. Al inicio del entrenamiento la probabilidad es alta, favoreciendo el uso del *ground truth*; conforme avanza, disminuye, incentivando el paso progresivamente de *teacher forcing* hacia la autoregresión

Este esquema en cascada asegura que (i) la corrección sólo se active cuando el error latente es elevado y (ii) incluso en esos casos, introduce el *ground truth* de forma controlada mediante *teacher forcing* probabilístico, lo que evita sobreajuste y facilitando una transición estable hacia la autoregresión completa. Con ello se favorece la consistencia en predicciones a largo plazo.

8.1.4. Estimación de flujo óptico con el método de Farnebäck

El método de Farnebäck [52] es un método clásico que permite calcular el flujo óptico. El método consiste en aproximar la intensidad local de los pixeles entre imágenes consecutivas mediante una función polinómica cuadrática, permitiendo modelar cada vecindario de píxeles de manera continua, para ello, se estima el desplazamiento de cada píxel en términos de un vector de movimiento (u, v).

En términos matemáticos, la relación entre dos imágenes consecutivas $I_t(\mathbf{x})$ e $I_{t+1}(\mathbf{x})$ se expresa como:

$$I_{t+1}(\mathbf{x}) \approx I_t(\mathbf{x} + \mathbf{d}), \tag{9}$$

donde $\mathbf{d} = (u, v)$ representa el vector de desplazamiento buscado. Por lo tanto, para estimar el flujo óptico se debe determinar el $\mathbf{d}$ que mejor alinea las representaciones polinómicas locales de ambas imágenes minimizando la diferencia entre dichas aproximaciones.

En este sentido, el método de Farnebäck permite estimar el flujo óptico capturando movimientos continuos en escenas dinámicas. Este método fue empleado como estimador de flujo en los modelos descritos en las Secciones 8.3 y 8.4.

8.1.5. Métricas de evaluación

La evaluación de modelos de predicción secuencial requiere métricas capaces de capturar tanto la fidelidad visual de las imágenes generadas como la consistencia del movimiento en el caso del flujo óptico. Por esta razón, para evaluar la calidad de la predicción de los modelos propuestos se implementarán métricas estructurales y perceptuales ampliamente utilizadas en visión por computadora, junto con medidas específicas de flujo. A continuación se describen las métricas seleccionadas, donde x corresponde a la imagen verdadera e y a la imagen predicha obtenida por el modelo.

Métricas de evaluación para imágenes

- **Índice de Similitud Estructural (SSIM):** El SSIM cuantifica la similitud entre dos imágenes considerando tres componentes: luminancia, contraste y estructura.

$$\mathrm{SSIM}(x, y) = \frac{\big(2\mu_x\mu_y + C_1\big)\big(2\sigma_{xy} + C_2\big)}{\big(\mu_x^2 + \mu_y^2 + C_1\big)\big(\sigma_x^2 + \sigma_y^2 + C_2\big)} \tag{10}$$

  donde μ_x, μ_y son las medias, σ_x^2, σ_y^2 las varianzas, σ_{xy} es la covarianza y C_1, C_2constantes de estabilización. Valores cercanos a 1 indican alta similitud estructural.

- **Peak Signal-to-Noise Ratio (PSNR):** [77]: El PSNR mide la diferencia pixel a pixel en relación con el rango máximo de la señal. Se expresa como:

$$\mathrm{PSNR} = 10log_{10}\left(\frac{{\mathrm{MAX}_I}^2}{\mathrm{MSE}}\right) \tag{11}$$

donde MAX_I es el valor máximo posible de intensidad y MSE el error cuadrático medio. Valores altos de PSNR implican menor distorsión.

Cuanto mayor sea el valor de PSNR, menor es el error de reconstrucción y mejor la calidad percibida de la imagen generada.

- **Learned Perceptual Image Patch Similarity (LPIPS):** [78] El LPIPS evalúa la similitud perceptual entre imágenes comparando sus activaciones en redes neuronales profundas preentrenadas. A diferencia de SSIM o PSNR, se correlaciona mejor con la percepción humana de calidad visual; valores más bajos indican mayor similitud perceptual.

$$\mathrm{LPIPS}(x, y) = \sum_{l=1}^{L} \frac{1}{H_l W_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} \left\| w_l \odot \left(\hat{x}^l_{h,w} - x^l_{h,w} \right) \right\|_2^2, \tag{12}$$

donde $x^l(x)$ representa las características extraídas en la capa l de una red preentrenada, $\hat{x}^l$ son las características normalizadas canal por canal, w_l son pesos aprendidos que ponderan la contribución de cada canal, y H_l, W_l son las dimensiones espaciales de la capa. Valores bajos de LPIPS indican mayor similitud perceptual entre las imágenes.

Métricas de evaluación para flujo óptico

- **Endpoint Error (EPE):** El EPE es la distancia euclídea promedio entre el flujo óptico estimado $\hat{\mathbf{f}}$ y el flujo real $\mathbf{f}$.

$$\mathrm{EPE} = \frac{1}{N} \sum_{x,y} \| \mathbf{f}(x, y) - \hat{\mathbf{f}}(x, y) \|_2 \tag{13}$$

Valores bajos de EPE indican mayor precisión en la estimación del movimiento.

- **Error Angular Medio (AE):**. El AE mide la diferencia angular entre los vectores de flujo estimado y real, independientemente de su magnitud:

$$\theta(x, y) = \arccos \left(\frac{\hat{\mathbf{f}}(x, y) \cdot \mathbf{f}(x, y)}{\|\hat{\mathbf{f}}(x, y)\| \cdot \|\mathbf{f}(x, y)\|} \right) \tag{14}$$

Este indicador es especialmente útil en escenas donde interesa capturar la dirección del movimiento.

Estas métricas ofrecen una evaluación integral: SSIM y PSNR miden la fidelidad estructural de las predicciones, LPIPS aproxima la percepción visual humana y EPE junto con AE evalúan la precisión del movimiento. Gracias a esta combinación, es posible evaluar el desempeño de los modelos propuestos tanto desde el punto de vista de la calidad visual como de la consistencia dinámica, aspectos fundamentales en la predicción de video y flujo óptico.

En la Tabla 2 se presenta un resumen de las métricas consideradas para evaluar los métodos propuestos, indicando su descripción y el criterio esperado en términos de valor alto (flecha hacia arriba) o bajo (flecha hacia abajo).

Tabla 2: Resumen de métricas de evaluación utilizadas en imágenes RGB y flujo óptico.

Métrica	**Descripción**	**Valor esperado**
	Métricas para imágenes	
SSIM	Similitud estructural (luminancia, contraste, estructura).	$\uparrow$ (mayor es mejor)
PSNR	Relación señal-ruido (calidad frente al error pixel a pixel).	$\uparrow$ (mayor es mejor)
LPIPS	Distancia perceptual en espacio de características profundas.	$\downarrow$ (menor es mejor)
	Métricas para flujo óptico	
EPE	Distancia euclídea promedio entre flujo estimado y real.	$\downarrow$ (menor es mejor)
AE	Diferencia angular promedio entre vectores de flujo.	$\downarrow$ (menor es mejor)

8.1.6. Conjuntos de datos de entrenamiento y validación

Para entrenar los modelos, se utilizó el dataset KITTI Dataset [79], que fue dividido en un $80\,\%$ para entrenar, $10\,\%$ para validación y $10\,\%$ para test.

Luego, para evaluar la generalización de los modelos a datos desconocidos se utilizaron dos dataset: KTH-Actions Dataset [80] y Caltech Pedestrian Dataset [81]. A continuación se describe cada uno

- **KITTI Dataset:** Corresponde a un set de imágenes que fueron grabadas desde un vehículo en movimiento, ofreciendo un amplio conjunto de información visual y de profundidad. Cubre entornos urbanos, rurales y autopistas, con diversas condiciones climáticas, de iluminación y escenas dinámicas con automóviles, peatones y ciclistas.

- **KTH-Actions Dataset:** Consiste en grabaciones de video reales de personas realizando acciones como: trotar, saludar con la mano y caminar, saltar. Se utiliza como referente para tareas de reconocimiento de acciones y predicción de video, ofreciendo patrones de movimiento naturales y variaciones dinámicas en las escenas.

- **Caltech Pedestrian Dataset:** Es un conjunto de datos centrado en la detección de peatones en entornos urbanos, ampliamente utilizado para entrenar y evaluar modelos de visión por computadora enfocados en la seguridad de la conducción autónoma.

8.2. Objetivo 1: Determinar si un espacio latente que incluya el flujo óptico (arquitectura PE) permite capturar la estructura espacio-temporal de una escena: Modelo Pre Encoder

El primer objetivo busca verificar si es posible generar predicciones del flujo óptico cuando este se incorpora directamente como entrada al modelo. Para ello se propone la arquitectura *Pre Encoder* (PE), en la cual los mapas de flujo óptico, estimados previamente mediante el método SelFlow [55], son utilizados como base para el proceso de codificación y predicción. Este diseño se construye a partir de los componentes generales descritos en la Sección 8.1, en particular la ConvGRU (Sección 8.1.2) y el módulo de decisión autorregresiva (Sección 8.1.3). En este modelo se busca identificar si al disponer del flujo óptico como insumo inicial, el modelo puede aprovechar esta información de movimiento para capturar la estructura espacio–temporal de la escena y generar predicciones consistentes de flujo óptico en horizontes futuros.

8.2.1. Estimación de flujo óptico con Selflow

Para generar los mapas de flujo que sirven de entrada al autoencoder, se utilizó Selflow [55], un método auto-supervisado que combina alta precisión en zonas ocluidas (zonas generadas por la superposición de objetos impidiendo la estimación del flujo óptico de manera directa) junto con eficiencia computacional, lo cual es esencial para pre-entrenar un autoencoder genérico con datos realistas. y movimientos complejos.

En particular, SelFlow funciona de la siguiente manera: primero obtiene la información temporal de tres imágenes consecutivas (I_{t-1}, I_t, I_{t+1}) para construir una pirámide de características en la que en cada nivel de la pirámide se reduce la resolución espacial de las imágenes creando mapas de características más compactos. Esto permite estimar flujos grandes a baja resolución lo que reduce el coste computacional; luego refina progresivamente las correcciones a medida que se asciende a las escalas superiores. En cada nivel aplica un *warping* progresivo para alinear las características de I_{t-1} e I_{t+1} con las de I_t, mejorando la precisión de las comparaciones entre imágenes al facilitar

la detección de oclusiones y posibilitar un refinado progresivo del flujo óptico en la pirámide de características.

El entrenamiento se realiza sin etiqueta real del flujo óptico, para ello, minimiza la función de pérdida multi-escala combinando ℓ_1 y SSIM. Gracias a este diseño, Selflow produce estimaciones más precisas que los métodos unidireccionales tradicionales, especialmente en zonas con oclusiones y bordes de objetos.

Finalmente, los mapas de flujo resultantes se utilizaron como datos de entrada para el pre entrenamiento del autoencoder.

8.2.2. Autoencoder de flujo óptico

El entrenamiento del autoencoder de flujo óptico constituye una etapa previa destinada a aprender representaciones latentes del flujo óptico, que luego se utilizan para inicializar el módulo *Pre Encoder*. De esta manera, se busca asegurar que el modelo principal comience con pesos adaptados al dominio del flujo, favorenciendo con esto la estabilidad del entrenamiento y la eficiencia en la convergencia.

Arquitectura del autoencoder

El autoencoder se compone de un codificador y un decodificador simétricos que recibe como entrada un campo de flujo óptico bidimensional, tal como se describe a continuación:

1. **Codificador**: Consta de una red convolucional poco profunda que extrae mapas de características a partir del flujo de entrada. Cada bloque incluye una convolución 2D, seguida de normalización `GroupNorm` y activación `LeakyReLU`.

   `GroupNorm` es una técnica de normalización que divide los canales de cada mapa de características en un número fijo de grupos y normaliza de forma independiente dentro de cada grupo, calculando la media y la varianza a partir de todos los elementos espaciales y canales de ese grupo [82].

   La configuración de las capas es la siguiente:

   - **Capa 1:** Recibe un tensor $[B, 2, H, W]$, donde B es el tamaño del batch y (u, v) representan las dos componentes del flujo y (H, W) es el tamaño espacial del mismo. Aplica una convolución 2D (kernel 3×3) con 2 canales de entrada y 32 de salida, seguida de `GroupNorm` con 8 grupos. La salida mantiene la resolución original y adopta la forma $[B, 32, H, W]$.

- **Capa 2:** Opera sobre $[B, 32, H, W]$ mediante una convolución 2D con kernel 3×3 (32→64 canales) y la misma combinación de normalización y activación. La resolución espacial se conserva, generando una salida de esta forma $[B, 64, H, W]$.
- **Capa 3:** Procesa $[B, 64, H, W]$ con una convolución 3×3 (64→128 canales) manteniendo la resolución espacial, obteniendo $[B, 128, H, W]$.

2. **Decodificador:** Su función es reconstruir un flujo óptico a resolución completa a partir de las características extraídas por el codificador. Para ello, ajusta progresivamente el número de canales hasta recuperar las dos componentes finales (u, v). Consta de tres capas como se definen a continuación:

   - **Capa 1:** Parte de $[B, 128, H, W]$ y aplica una convolución 3×3, (128→64 canales). Se utiliza nuevamente `GroupNorm` con 8 grupos y activación `LeakyReLU`.
   - **Capa 2:** Mantiene la resolución aplicando una convolución 3×3 (64→32 canales), seguida de normalización y activación como en capas previas. La salida dee esta capa es de la forma $[B, 64, H, W]$.
   - **Capa 3:** A partir de la capa anterior, proyecta el tensor resultante a $[B, 2, H, W]$ mediante una convolución 3×3 (32→2 canales). Esta última capa no incorpora normalización ni activación, para preservar el rango y la escala del flujo reconstruido.

Función de pérdida del autoencoder

El entrenamiento se guía por una función de pérdida compuesta que combina EPE y AE. El EPE mide la precisión en la magnitud de los vectores de flujo, mientras que el AE evalúa su dirección. De esta manera, la función de perdida queda definida como:

$$\mathcal{L} = \alpha \cdot \mathcal{L}(\mathrm{EPE}) + (1 - \alpha) \cdot \mathcal{L}(\mathrm{AE}) \tag{15}$$

donde $\alpha \in [0, 1]$ pondera la contribución de EPE y AE asegurando un balance entre magnitud y orientación, evitando que el modelo favorezca vectores grandes a expensas de errores angulares.

Los pesos preentrenados de este autoencoder serán utilizados en el entrenamiento del modelo principal de predicción llamado *Pre Encoder*, el cual tiene como objetivo principal generar predicciones de vectores de flujo óptico en escenarios dinámicos.

8.2.3. Arquitectura del modelo Pre Encoder

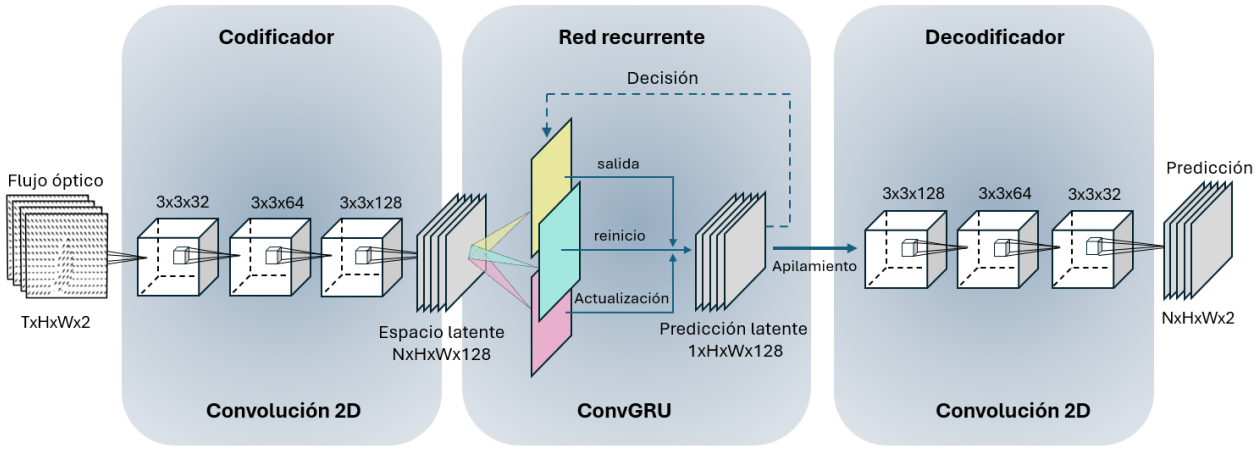


Figura 3: Arquitectura general del Modelo Pre Encoder (PE): El sistema se compone de tres módulos principales: un codificador convolucional, una red recurrente basada en ConvGRU y un decodificador convolucional. Cada uno de estos componentes se detalla en la Tabla 3, donde se especifican las capas y configuraciones empleadas en el diseño.

Módulo	Capa	Configuración	Salida
Codificador	1	Conv2D (2→32), 3×3, stride=1	$[B, T, 32, H, W]$
	2	Conv2D (32→64), 3×3, stride=1	$[B, T, 64, H, W]$
	3	Conv2D (64→128), 3×3, stride=1	$[B, T, 128, H, W]$
ConvGRU	1	kernel 3×3, 128 canales, dropout=0.1	$[B, 128, H, W]$
	2	kernel 3×3, 128 canales, dropout=0.1	$[B, 128, H, W]$
Decodificador PWC	1	Conv2D (128→128), 3×3, stride=1	$[B, 128, H, W]$
	2	Conv2D (128→64), 3×3, stride=1	$[B, 64, H, W]$
	3	Conv2D (64→32), 3×3, stride=1	$[B, 32, H, W]$
	4	Conv2D (32→2), 3×3, stride=1	$[B, 2, H, W]$

Tabla 3: Resumen de la arquitectura del modelo Pre Encoder (PE).

El modelo propuesto, denominado *Pre Encoder*, se organiza en tres módulos principales que integran codificación convolucional, predicción autoregresiva mediante una red recurrente y un codificador que genera un refinamiento jerárquico de flujo óptico tal como se describe a continuación. La arquitectura del modelo se ilustra en la Figura 3, mientras que el detalle de cada módulo se presenta en la Tabla 3.

1. **Codificador**: Se inicializa con los pesos obtenidos en el autoencoder de flujo óptico (sección 8.2.2). Durante el entrenamiento, sus parámetros son congelados con el objetivo de

mantener su capacidad de extracción de características espaciales inalterada. La salida de este módulo corresponde a representaciones latentes compactas que sirven como base para alimentar al módulo recurrente ConvGRU.

2. **Módulo recurrente ConvGRU:** Este módulo corresponde a la ConvGRU descrita en detalle en la Sección 8.1.2, empleada aquí en una configuración de dos capas apiladas con convoluciones 3×3, *stride* $= 1$, *padding* $= 1$ y una tasa de *dropout* $= 0.1$. Su función es modelar la dinámica temporal, actualizando el estado oculto h_t en cada paso a partir de la representación latente y del estado previo h_{t-1}.

3. **Módulo de decisión autorregresiva.** Tras cada predicción, se calcula el error latente y se compara con el umbral definido, tal como se explica en la Sección 8.1.3. Si el error es bajo, la propia predicción alimenta el siguiente paso; si el error supera el umbral, se aplica *scheduled sampling*, que introduce de manera probabilística el *ground truth* para mitigar la acumulación de errores.

4. **Decodificador:** Su función es reconstruir un flujo óptico a resolución completa a partir del vector latente predicho por la red recurrente. Para ello, mantiene la resolución espacial y reduce progresivamente el número de canales hasta proyectar las dos componentes finales (u, v). Consta de tres capas, definidas de la siguiente manera:

   - **Capa 1:** Parte de $[B, 128, H, W]$ y aplica una convolución 3×3 (128→64 canales). Como función de activación se usa una ReLU.
   - **Capa 2:** Mantiene la resolución $[B, 64, H, W]$ aplicando una convolución 3×3 (64→32 canales), seguida de normalización y activación como en la capa previa.
   - **Capa 3:** Proyecta el tensor resultante a $[B, 2, H, W]$ mediante una convolución 3×3 (32→2 canales). Esta última capa no incorpora normalización ni activación, con el fin de preservar el rango y la escala del flujo reconstruido.

   Este decodificador constituye el módulo final del modelo, encargado de transformar las representaciones latentes generadas tras el módulo de decisión en secuencias de flujo óptico consistentes y de alta calidad.

Esta arquitectura combina la capacidad de la ConvGRU para capturar dependencias temporales con un mecanismo de decisión autorregresiva que regula la propagación de errores. El uso del decodificador en la salida de la ConvGru permite reconstruir las predicciones de flujo para que puedan ser evaluadas en la función de pérdida.

Función de pérdida del modelo Pre Encoder

Para optimizar la predicción de flujo óptico empleamos una función de pérdida que integra tres componentes complementarios a EPE definido en la sección 8.1.5, similitud de coseno y suavidad espacial.

La **Similitud del coseno** evalúa la alineación direccional entre los vectores de flujo predicho y el flujo real sin necesidad de calcular el ángulo explícito. Cada vector se normaliza a magnitud unitaria y se mide el coseno del ángulo mediante el producto escalar, un valor cercano a 1 indica alineación perfecta, mientras que valores próximos a 0 o negativos reflejan desalineación.

$$\cos(\theta) = \frac{\vec{u}_{\text{pred}} \cdot \vec{u}_{\text{gt}}}{\|\vec{u}_{\text{pred}}\| \, \|\vec{u}_{\text{gt}}\|} \tag{16}$$

En entrenamiento se emplea como función de pérdida la forma $\mathcal{L}_{\cos} = 1 - \cos(\theta)$, que penaliza las diferencias de orientación entre ambos vectores.

La **Suavidad espacial (Smoothness)** penaliza variaciones abruptas en el campo de flujo, computando la media de las diferencias absolutas entre píxeles vecinos:

$$\text{Smooth} = \frac{1}{N} \sum_{t} \Big(|\partial_x f^{(t)}| + |\partial_y f^{(t)}| \Big) . \tag{17}$$

al unir estas tres métricas, la función de pérdida para el modelo Pre Encoder se expresa como:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{EPE}} + \beta \mathcal{L}_{\cos} + \gamma \mathcal{L}_{\text{Smooth}} \tag{18}$$

donde α, β y γ son pesos que ajustan la contribución de cada término . Esta combinación permite simultáneamente mejorar la precisión de magnitud, la fidelidad direccional y la coherencia espacial del flujo estimado.

En resumen, el modelo *Pre Encoder* representa un primer paso para poner a prueba la integración del flujo óptico en un modelo de codificación predictivo, en este caso como punto de partida en la predicción espacio temporal. Su combinación de tres bloques Codificador - Red Recurrente - Decodificador junto con un mecanismo de decisión que regula la autoregresión, permite generar una base para evaluar la capacidad predictiva de este enfoque. Con ello, se abre el camino a contrastar esta estrategia con variantes alternativas, como la que se presenta en el siguiente objetivo, donde el cálculo del flujo se traslada al espacio latente para explorar nuevas ventajas y desafíos.

8.3. Objetivo 2: Determinar si un flujo óptico derivado del espacio latente (arquitectura PLS) permite capturar la estructura espacio-temporal de una escena: Modelo Post Latent Space

El segundo objetivo plantea evaluar si es posible generar predicciones del flujo óptico en el espacio latente. Para ello se propone un modelo *Post Latent Space* (PLS), cuya arquitectura se organiza en tres bloques principales: (i) un encoder convolucional 3D encargado de extraer representaciones espaciales del flujo, y (ii) un módulo de cálculo de flujo óptico en el espacio latente y (iii) un módulo ConvGRU que opera en dicho espacio latente para capturar dependencias temporales.

Al igual que los demás modelos, este diseño se apoya en los componentes generales presentados en la Sección 8.1, particularmente en el bloque ConvGRU (Sección 8.1.2). Sin embargo, a diferencia del modelo PE, aquí el flujo óptico no se introduce como entrada directa, sino que se estima en el espacio latente generado por el codificador mediante el método de Farnebäck descrito en la Sección 8.1.4. El objetivo es evaluar si esta estrategia resulta suficiente para capturar la dinámica de la secuencia. Cabe destacar que este modelo no incorpora el módulo de decisión autorregresiva (Sección 8.1.3), debido a que las predicciones de flujo latente no alcanzan convergencia estable, como se discute en la Sección 9.2.

8.3.1. Arquitectura del modelo Post Latent Space

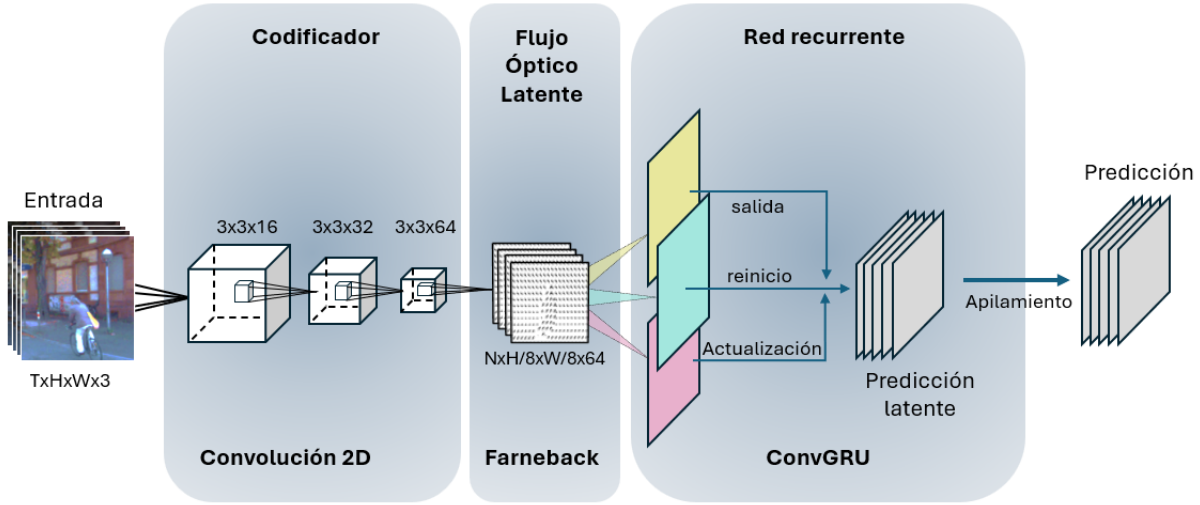


Figura 4: Arquitectura general del Modelo Post Latent Space (PLS): el modelo consta de un codificador procesa la secuencia de imágenes de entrada mediante tres bloques de convoluciones 2D, generando progresivamente mapas de características de 16, 32 y 64 canales y reduciendo sus dimensiones en H/2, H/4 y H/8, el espacio latente obtenido del decodificador es usado para generar los mapas de flujo óptico que alimentará a la red recurrente de tipo ConvGRU para generar las predicciones, los detalles del modelo se encuentran en la Tabla 4.

Módulo	Capa/Operación	Configuración	Salida
Codificador	1	Conv2D (3→16), 3×3, stride=2	$[B, T, 16, H/2, W/2]$
	2	Conv2D (16→32), 3×3, stride=2	$[B, T, 32, H/4, W/4]$
	3	Conv2D (32→64), 3×3, stride=2	$[B, T, 64, H/8, W/8]$
Cálculo de flujo óptico	Farneback	Pirámide multiescala sobre z_t, z_{t+1}	Flujo inicial $[B, 2, H/8, W/8]$
ConvGRU	Bloque recurrente	kernel 3×3, 64 canales	Predicción latente $[B, 64, H/8, W/8]$

Tabla 4: Resumen de la arquitectura del modelo Post Latent Space (PLS).

El modelo propuesto en este objetivo, denominado *Post Latent Space*, se organiza en tres módulos principales que integran codificación convolucional de imágenes RGB, cálculo de flujo óptico y Red Recurrente como se describe a continuación y se ilustra en la Figura 4, mientras que el detalle de cada módulo se presenta en la Tabla 4.

1. **Codificador**: El codificador reduce progresivamente la resolución de las imágenes RGB en una representación latente más compacta, manteniendo la información esencial a través de tres capas convolucionales con diseño similar. Cada capa aplica una convolución 2D con $\text{kernel} = 3 \times 3$, $\text{stride} = 2$ y $\text{padding} = 1$, seguida de una activación ReLU, duplicando el número de canales en cada etapa.

- **Capa 1:** recibe un tensor de entrada con dimensiones $[B, T, 3, H, W]$, donde B es el tamaño del batch, T el número de imágenes en la secuencia, 3 los canales RGB y (H, W) la resolución original. Tras la convolución (3→16 canales), la salida es $[B, T, 16, H/2, W/2]$.
- **Capa 2:** procesa el tensor $[B, T, 16, H/2, W/2]$ y, tras la convolución (16→32 canales), genera una salida $[B, T, 32, H/4, W/4]$.
- **Capa 3:** toma como entrada $[B, T, 32, H/4, W/4]$ y produce un tensor latente con dimensiones $[B, T, 64, H/8, W/8]$, duplicando nuevamente los canales y reduciendo la resolución espacial a un octavo de la original.

Con este diseño, el codificador genera una reducción de la resolución de la imagen progresivamente disminuyendo el costo comunicacional y ampliando el campo de visión de cada kernel, lo que permite integrar información de un contexto más amplio de la escena. Al mismo tiempo, el aumento de canales permite conservar los detalles locales asi como también las estructuras globales de la imagen. Al finalizar el tercer bloque, la secuencia de imágenes queda resumida en un espacio latente de dimensiones $[B, T, 64, H/8, W/8]$, listo para ser utilizado por la unidad encargada de calcular el flujo óptico

2. **Cálculo del flujo óptico**: A la salida del encoder se obtiene una representación latente $Z \in \mathbb{R}^{B\times T\times 64\times H/8\times W/8}$, donde z_t corresponde a la codificación de la imagen I_t. Para estimar el movimiento entre instantes consecutivos, se consideran los pares (z_t, z_{t+1}), que se entregan al módulo de cálculo de flujo óptico basado en el método de Farneback [52].

   Aquí, el método de Farnebäck [52] opera sobre las proyecciones en el espacio latente generadas por el codificador. Este método estima los desplazamientos entre pares consecutivos (z_t, z_{t+1}) a partir de las variaciones locales en estas representaciones. Para capturar movimientos de distinta magnitud, se utiliza una pirámide de resoluciones: primero se estiman desplazamientos globales en baja escala y luego se refinan en niveles superiores, incorporando progresivamente los detalles locales necesarios para un flujo más preciso y estable.

   Para este modelo se adoptó el método de Farnebäck debido a dos ventajas prácticas:

   - Permite estimar de manera rápida un campo de flujo inicial a partir de las activaciones del codificador, sin necesidad de entrenar módulos adicionales.
   - Al estar reducida la resolución de la imagen en el espacio latente. permite realizar cálculos de desplazamientos locales de manera mas rápida, acelerando con esto el procesamiento en comparación con operar directamente sobre la imagen original.

Finalmente, este módulo produce un flujo óptico inicial a partir de las proyecciones latentes del codificador, ofreciendo una primera estimación del movimiento entre representaciones consecutivas. Luego con el objetivo de poder generar predicciones de flujo óptico basadas en este espacio latente , los mapas de estos flujos se emplean como entrada a la RNN del tipo ConvGRU

3. **Módulo recurrente ConvGRU**: Los mapas de flujo óptico obtenidos en la etapa anterior se entregan a una unidad recurrente del tipo ConvGRU, previamente descrita en la Sección 8.1.2. En el contexto de este modelo, la ConvGRU cumple el rol de integrar la información de movimiento del flujo óptico, capturando dependencias temporales y generando predicciones de este mismo. De esta manera, el modelo extiende la estimación inicial limitada a pares consecutivos de representaciones hacia una proyección temporal más amplia, manteniendo la continuidad y consistencia del flujo óptico en el tiempo.

Función de pérdida del modelo PLS

Para supervisar la calidad del flujo óptico predicho se definió una pérdida combinada que considera tanto la magnitud como la orientación de los vectores de desplazamiento. Sea f' el flujo predicho y f el flujo de referencia (*ground truth*). La función de pérdida integra dos términos principales:

- **Norma** ℓ_1: mide la discrepancia absoluta en magnitud entre f' y f, definida como $\mathcal{L}\ell 1(f', f)$.
- **Similitud del coseno**: evalúa la coherencia direccional entre ambos flujos sin necesidad de calcular explícitamente el ángulo. Para cada píxel se normalizan los vectores y se calcula:

$$\cos(\theta) = \frac{\vec{u}_{\text{pred}} \cdot \vec{u}_{\text{gt}}}{\|\vec{u}_{\text{pred}}\| \ \|\vec{u}_{\text{gt}}\|} \tag{19}$$

  Durante el entrenamiento se utiliza la forma $\mathcal{L}_{\cos} = 1 - \cos(\theta)$, que penaliza de manera diferenciable las desalineaciones de orientación. El resultado se promedia sobre todos los píxeles y pasos temporales.

La combinación final se expresa como:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\ell 1} + \lambda \mathcal{L}_{\cos} \tag{20}$$

donde $\lambda \in [0, 1]$ regula el equilibrio entre la precisión en magnitud y la consistencia direccional. Este esquema permite ponderar de manera flexible ambos aspectos, favoreciendo estimaciones de flujo óptico que no solo sean numéricamente cercanas al valor real, sino también coherentes en su orientación espacial.

En síntesis, el modelo PLS permite explorar la viabilidad de generar predicciones de flujo óptico a partir de representaciones latentes. Sin embargo, la reducción espacial aplicada en el codificador conlleva una pérdida considerable de información, lo que dificulta obtener un flujo confiable para sostener la autoregresión. Como consecuencia, el modelo no logra una convergencia estable durante el entrenamiento, limitación que se analiza en detalle en la Sección 9.2.

8.4. Objetivo 3: Determinar si un flujo óptico integrado en la función de pérdida (arquitectura ACCLIP) permite capturar la estructura espacio-temporal de una escena:

El tercer objetivo busca analizar si el flujo óptico, considerado como información externa al modelo, es decir, no utilizado en el espacio latente ni como entrada explícita, puede generar predicciones confiables de imágenes cuando se integra únicamente a través de la función de pérdida.

Para ello se propone el modelo ACCLIP (*Corrección Adaptativa con Pérdida Combinada Integrando Flujo Óptico para una Mejora en la Predicción*), que combina codificación convolucional 3D, bloques recurrentes ConvGRU y un decodificador 3D, junto con un módulo de decisión autorregresiva.

Este modelo integra de manera explícita los componentes generales de la Sección 8.1, incluyendo la ConvGRU (Sección 8.1.2) y el módulo de decisión autorregresiva (Sección 8.1.3). A diferencia de PE y PLS, en ACCLIP el flujo óptico no forma parte de la arquitectura interna, sino que se incorpora únicamente en la función de pérdida. El propósito es analizar si esta estrategia permite obtener secuencias con mayor fidelidad visual y coherencia temporal sin depender del cálculo directo del flujo dentro del modelo.

8.4.1. Arquitectura del modelo ACCLIP

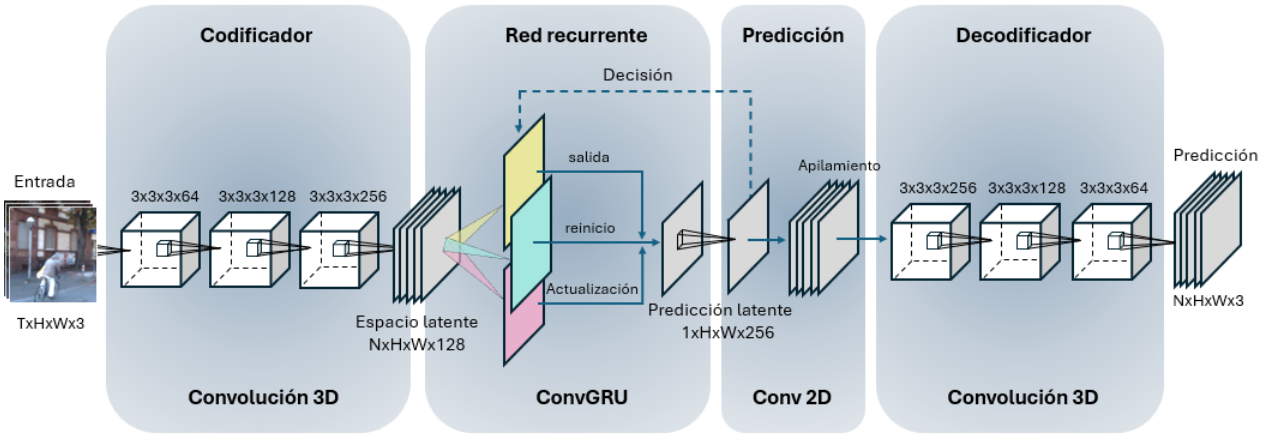


Figura 5: Arquitectura general del ACCLIP: el modelo consta de cuatro módulos, un codificador convolucional 3D que procesa la secuencia de imágenes de entrada, una red recurrente basada en ConvGRU, un módulo de prediccion de convoluciones 2D junto con una unidad de desición y un decodificador convolucional 3D. El detalle de cada uno de estos componentes se encuentran en la Tabla 5.

Módulo	Capa/Operación	Configuración	Salida
Codificador 3D	1	Conv3D (3→64), $3 \times 3 \times 3$, stride=1	$[B, 64, T, H, W]$
	2	Conv3D (64→128), $3 \times 3 \times 3$, stride=1	$[B, 128, T, H, W]$
	3	Conv3D (128→256), $3 \times 3 \times 3$, stride=1	$[B, 256, T, H, W]$
ConvGRU	1	ConvGRU, kernel 1×1, 256 canales, dropout=0.1	$[B, 256, H, W]$
	2	ConvGRU, kernel 1×1, 256 canales, dropout=0.1	$[B, 256, H, W]$
Predicción	Conv2D + decisión	Dos conv 1×1 (256→256→256) + módulo de decisión	Latente predicho $[B, 256, H, W]$
Decodificador 3D	1	Conv3D (256→128)$3 \times 3 \times 3$, stride=1	$[B, 128, T, H, W]$
	2	Conv3D (128→64), $3 \times 3 \times 3$, stride=1	$[B, 64, T, H, W]$
	3	Conv3D (64→3), $3 \times 3 \times 3$, stride=1	$[B, 3, T, H, W]$

Tabla 5: Resumen de la arquitectura del modelo ACCLIP.

El modelo propuesto, denominado *Corrección Adaptativa con Pérdida Combinada Integrando Flujo Óptico para una Mejora en la Predicción* (ACCLIP), se organiza en cuatro módulos principales que integran un codificador con bloques de convolución 3D, una red recurrente basada en ConvGRU, una unidad de predicción con bloques de convolución 2D más un mecanismo de desición y un decodifcador de convolcuiones 3D. La Figura 5 ilustra la arquitectura del modelo, mientras que el detalle de cada módulo se presenta en la Tabla 5. A continuación se describen cada uno de los módulos

1. **Codificador 3D**: Transforma bloques de video en representaciones latentes manteniendo la resolución espacial y temporal. Se compone de tres capas convolucionales 3D con $\text{stride} = 1$, $\text{padding} = 1$ y kernel $3 \times 3 \times 3$. Cada capa está seguida de BatchNorm3D y activación ReLU.

- **Capa 1**: recibe un tensor $[B, T, 3, H, W]$ (imágenes RGB) y aplica una convolución 3D con 64 filtros. La salida tiene forma $[B, 64, T, H, W]$.
- **Capa 2**: procesa $[B, 64, T, H, W]$ con una convolución 3D de 128 filtros, generando $[B, 128, T, H, W]$.
- **Capa 3**: toma $[B, 128, T, H, W]$ y aplica una convolución 3D con 256 filtros, produciendo un tensor $[B, 256, T, H, W]$.

Este diseño conserva los detalles espaciales finos al no reducir resolución, permitiendo que las representaciones latentes conserven simultáneamente información de textura y de movimiento global.

2. **Módulo recurrente ConvGRU**: Este módulo corresponde a la ConvGRU descrita en la Sección 8.1.2 , utilizada aquí en una configuración de dos capas apiladas.

   - **Entrada y estado oculto**: 256 canales.
   - **Kernel**: 1×1, preservando la estructura espacial.
   - **Dropout**: 0.1, para mejorar generalización.

   La salida de la primera ConvGRU alimenta directamente a la segunda, lo que permite capturar dependencias temporales más profundas sin alterar la resolución de los mapas de características.

3. **Módulo de predicción autoregresiva**: A partir del estado oculto de la ConvGRU se aplican dos convoluciones 2D de tamaño 1×1, con activación ReLU intermedia, que reconfiguran las 256 características latentes para generar la predicción de cada paso futuro sin modificar la resolución espacial.

   Además, en este bloque se incorpora el mecanismo de decisión autorregresiva descrito en la Sección 8.1.3. Dicho mecanismo combina un umbral adaptativo de error y *scheduled sampling* para decidir si la siguiente entrada de la ConvGRU corresponde a la predicción generada por el modelo o al *ground truth*. En particular:

   - Si el error entre la predicción latente y el *ground truth* es menor al umbral definido ($\mathrm{MSE}(f', f) < 0.024$), la propia predicción se reutiliza como entrada en el siguiente paso temporal.
   - Si el error excede el umbral ($\mathrm{MSE}(f', f) > 0.024$), el modelo aplica *scheduled sampling*, incorporando probabilísticamente el *ground truth* con una probabilidad decreciente a lo largo del entrenamiento.

El módulo de predicción genera representaciones latentes y, al mismo tiempo, controla la propagación de la información temporal, reduciendo la acumulación de errores y manteniendo la estabilidad en predicciones de largo alcance.

4. **Decodificador 3D**: Reconstruye la secuencia de imágenes a partir de las representaciones latentes predichas. Se compone de tres capas convolucionales 3D con kernel $3 \times 3 \times 3$, $\text{stride} = 1$ y $\text{padding} = 1$. Cada capa va seguida de BatchNorm3D y activación LeakyReLU.

   - **Capa 1**: transforma $[B, 256, T, H, W]$ en $[B, 128, T, H, W]$.
   - **Capa 2**: procesa $[B, 128, T, H, W]$ y reduce a $[B, 64, T, H, W]$.
   - **Capa 3**: proyecta a $[B, 3, T, H, W]$, recuperando el formato RGB original.

   De este modo, se obtiene una secuencia de imágenes predichas con la misma resolución espacial y temporal que la entrada.

Finalmente, la Tabla 5 muestra un resumen de la arquitectura ACCLIP, organizada en sus cuatro bloques principales: codificador 3D, red recurrente ConvGRU, módulo de predicción autoregresiva con mecanismo de decisión y decodificador 3D encargado de reconstruir los cuadros de salida.

8.4.2. Función de pérdida del modelo ACCLIP

El entrenamiento se guía por una pérdida híbrida que integra tres componentes complementarios:

- **MSE**: minimiza las diferencias a nivel de píxel entre cuadros reales y predichos.
- **SSIM**: preserva la similitud estructural entre ambos, priorizando la fidelidad perceptual.
- **Pérdida de flujo óptico**: garantiza la coherencia temporal penalizando discrepancias en los patrones de movimiento.

En este caso, la pérdida de flujo óptico se calcula aplicando el método denso de Farnebäck [52] sobre las secuencias de imágenes reales y predichas convertidas a escala de grises. Para cada par de cuadros consecutivos se estiman los campos de flujo correspondientes, F_t^{real} y F_t^{pred}. La pérdida se define como una norma L_1 entre ambos campos:

$$\mathcal{L}_{\text{flow}} = \| F_t^{\text{real}} - F_t^{\text{pred}} \|_1 \tag{21}$$

donde F_t^{real} corresponde al flujo calculado entre cuadros reales consecutivos y F_t^{pred} al flujo entre las predicciones realizadas modelo. De esta manera se asegura que la dinámica de movimiento

en la secuencia predicha se asemeje a la observada en la secuencia real, generando predicciones temporalmente coherentes.

La combinación final de la función de pérdida para entrenar al modelo ACCLIP se expresa como:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{MSE}} \cdot w_n + (1 - \alpha) \cdot \mathcal{L}_{\text{SSIM}} \cdot w_n + \beta \cdot \mathcal{L}_{\text{flow}} \tag{22}$$

El factor w_n, implementado como un vector linealmente decreciente, asigna mayor peso a las predicciones iniciales y lo reduce progresivamente en las más lejanas, que son intrínsecamente más difíciles de anticipar. Este mecanismo, inspirado en el concepto de proximidad temporal propuesto por [83], evita que los errores acumulados en horizontes largos dominen la señal de entrenamiento. Junto con los hiperparámetros α y β, que regulan la importancia relativa de cada término, este esquema permite equilibrar precisión numérica, fidelidad estructural y coherencia de movimiento, favoreciendo la generación de secuencias visualmente nítidas, estructuralmente consistentes y dinámicamente coherentes.

8.5. Objetivo 4: Evaluar la capacidad predictiva de los modelos PE, PLS y ACCLIP

Este objetivo busca integrar los resultados de los tres modelos propuestos (PE, PLS y ACCLIP) en un marco comparativo común, con el fin de responder a la pregunta central de esta investigación: *¿en qué etapa del modelo es más adecuado integrar el flujo óptico para generar predicciones confiables en entornos urbanos?*

Para ello, se plantea una evaluación unificada que garantice la comparabilidad entre las diferentes arquitecturas:

1. **Modelo PE (Pre–Encoder con flujo como entrada y salida):** Produce directamente mapas de flujo óptico futuros a partir de secuencias pasadas de flujo. Su evaluación se realiza en el espacio de flujo, comparando predicciones y flujo real mediante EPE y AE.

2. **Modelo PLS (Post Latent Space en espacio latente):** El flujo óptico se estima en el espacio latente generado por el codificador. Posteriormente, las predicciones latentes se transforman nuevamente en mapas de flujo para aplicar las métricas EPE y AE sobre los campos reconstruidos.

3. **Modelo ACCLIP (predicción de imágenes con flujo en la pérdida):** en este modelo tanto la entrada como la salida corresponden a imágenes RGB. Dado que la salida corresponde a imágenes futuras, se calcula flujo óptico tanto en las secuencias predichas como en las reales mediante SelFlow. De este modo, se obtienen el flujo predicho y el flujo real

al que luego se aplica EPE y AE. Además, se reportan métricas visuales (*SSIM* y *LPIPS*) para evaluar la fidelidad estructural y perceptual de las imágenes generadas.

4. **Comparación cruzada de resultados:** Una vez que todos los modelos se encuentran representado como flujo óptico, se realiza una comparación entre ellos utilizando las métricas EPE y AE. Esta evaluación cruzada permite identificar qué integración del flujo (entrada, latente o pérdida) logra el mejor equilibrio entre precisión y predicción.

De esta manera, se busca evaluar cómo la posición en la que se incorpora el flujo óptico afecta la capacidad predictiva de las arquitecturas propuestas.

9. Resultados

A continuación se presentan los resultados obtenidos a partir de la evaluación de los tres modelos propuestos, Pre Encoder (PE), Post Latent Space (PLS) y ACCLIP. El análisis se lleva a cabo sobre los conjuntos de datos descritos en la Sección 8.1.6, empleando las métricas de la Sección 8.1.5 para valorar precisión, coherencia estructural y consistencia temporal.

9.1. Objetivo 1: Evaluación del modelo Pre–Encoder

9.1.1. Evaluación Cualitativa de la Estimación de Flujo Óptico con Selflow

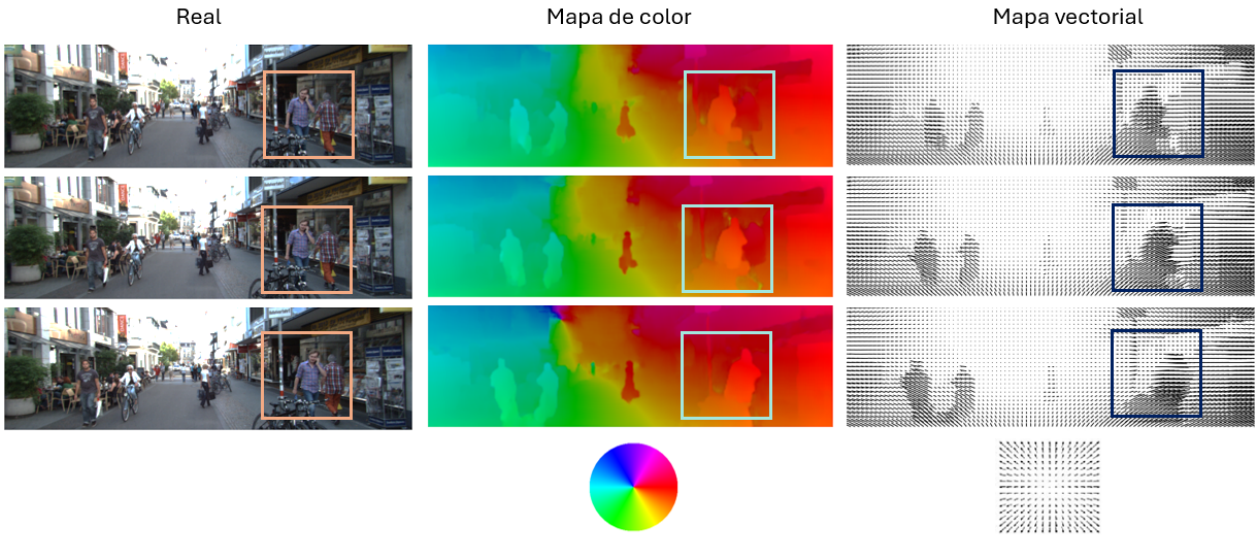


Figura 6: Comparación de las tres representaciones del flujo óptico estimado por Selflow: Columna izquierda: imagen real con recuadro naranja que marca la zona ocluida. Columna central: mapa de color HSV del flujo, donde el tono (Hue) corresponde al ángulo del flujo y la saturación a la magnitud, con la misma región destacada. Columna derecha: campo vectorial con recuadro azul, donde los vectores mantienen coherencia en el movimiento. En la parte inferior se incluye la rueda de colores HSV (centro) y un campo vectorial de referencia (derecha). Se aprecia que SelFlow filtra los píxeles no fiables en la zona de oclusión y reconstruye un flujo continuo y coherente.

Antes de entrenar el modelo PE fue necesario evaluar si SelFlow [55] resultaba apropiado para calcular el flujo óptico en escenarios urbanos. Sobre todo observar y analizar cómo este método maneja las oclusiones en este contexto, ya que estas son frecuentes en secuencias de tránsito peatonal y vehicular y representan un reto importante para mantener la coherencia del movimiento. Para este análisis se utilizó el conjunto de datos KITTI, ampliamente empleado en visión por computador en entornos urbanos. Como ejemplo, en la Figura 6 se muestra una secuencia donde dos peatones cruzan en direcciones opuestas. En cada instante se presentan la

imagen real (columna izquierda), el mapa de color HSV (*Hue-Saturation-Value*) del flujo (columna central) y el campo vectorial (columna derecha). Las regiones ocluidas se resaltaron con recuadros naranja (imagen real), cian (mapa de color) y azul (mapa vectorial), mostrando que SelFlow detecta y filtra correctamente los píxeles ocluidos, preserva la dirección del flujo sin introducir artefactos y mantiene la coherencia con las áreas adyacentes, lo que confirma que es una buena opción para calcular el flujo óptico en escenarios urbanos y para el entrenamiento del modelo PE.

9.1.2. Resultados del Autoencoder de flujo óptico

Con el fin de construir la base del modelo Pre Encoder , se entrenó un autoencoder utilizando como datos de entrada los flujos ópticos estimados previamente con SelFlow sobre el dataset KITTI. Su desempeño se evaluó en un subconjunto de validación no visto durante el entrenamiento, considerando distintos valores del parámetro α de la función de pérdida descrita en la Sección 8.2.2, con el objetivo de identificar la configuración más confiable.

Las métricas empleadas fueron EPE y AE, calculados entre los flujos originales y sus reconstrucciones. Los resultados se observan en la Tabla 6, donde se reportan los valores promedio y desviación estándar, lo que permite identificar el rango de α que ofrece un mejor equilibrio entre magnitud y dirección del flujo reconstruido.

A partir de la tabla es posible observar que:

1. Al poner más énfasis en minimizar el error de desplazamiento (EPE) aumentando α, se observa una caída pronunciada del EPE desde valores muy altos de 0.9 px con $\alpha = 0.1$ hasta aproximadamente 1.5 px con $\alpha = 0.9$

2. En cuanto al comportamiento del error angular, cuando α es muy bajo, el modelo prioriza la corrección de la dirección del flujo y logra errores angulares alrededor de $2.3°$ pero a expensas de un EPE muy elevado, al incrementar α, el error angular desciende tambien, alcanzando un mínimo de $1.40°$ en $\alpha = 0.8$ antes de elevarse levemente.

A partir de estos resultados se escogió el valor de $\alpha = 0.8$ para continuar con el entrenamiento del modelo ya que este valor genera un punto óptimo de compromiso entre EPE y el error angular, permitiendo mantener un bajo error de magnitud sin sacrificar la precisión direccional.

ALFA	EPE px↓	Error Angular (°)↓
0.1	9.0545 ± 3.3606 px	2.3344 ± 0.7070°
0.2	4.1740 ± 1.3241 px	1.5960 ± 0.4258°
0.3	2.6145 ± 0.9581 px	1.6328 ± 0.4239°
0.4	2.6740 ± 0.8732 px	1.8739 ± 0.4059°
0.5	2.8418 ± 1.0974 px	1.8409 ± 0.3861°
0.6	2.1408 ± 1.0340 px	1.8052 ± 0.4128°
0.7	1.9569 ± 0.7129 px	1.5654 ± 0.4244°
0.8	1.8266 ± 0.7426 px	1.4082 ± 0.3712°
0.9	1.5720 ± 0.5546 px	1.4412 ± 0.3579°

Tabla 6: Evaluación del autoencoder entrenado con flujos de SelFlow para diferentes valores de α en la función de pérdida. Los resultados corresponden a EPE y AE promediados en el conjunto de validación.

La figura 7 muestra una comparación visual entre el flujo óptico real y el flujo reconstruido por el autoencoder. En (b) se observa el flujo óptico calculado directamente a partir del par de imágenes, mientras que en (c) se presenta la reconstrucción generada por el autoencoder. La barra de color indica la magnitud del desplazamiento del vector de flujo óptico: los más oscuros representan vectores de mayor longitud, es decir, mayor desplazamiento, mientras que los tonos claros corresponden a desplazamientos menores. Puede notarse que el autoencoder logra reproducir de forma consistente las principales estructuras de movimiento presentes en la escena, lo que evidencia que el espacio latente aprendido por el modelo es capaz de identificar y representar los objetos de interés de manera efectiva.

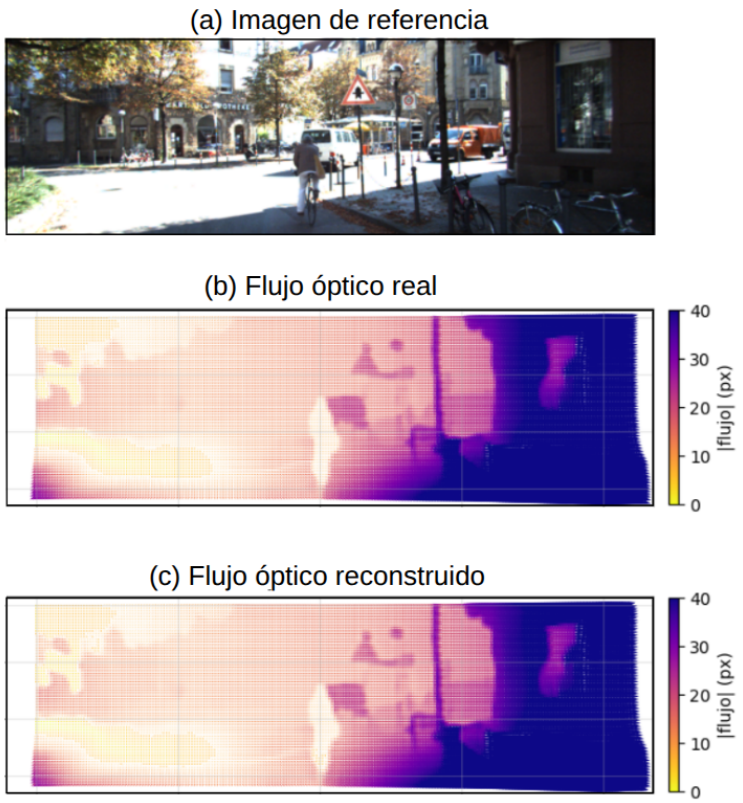


Figura 7: Comparación cualitativa entre flujo real y flujo reconstruido por el autoencoder (a) Imagen de referencia de la secuencia KITTI. (b) Flujo óptico real calculado a partir del par de imágenes. (c) Flujo óptico reconstruido por el autoencoder propuesto. La barra de color indica la magnitud del desplazamiento en píxeles por frame.

9.1.3. Resultados del modelo Pre Encoder

El modelo Pre Encoder (PE) recibe como entrada bloques de secuencias formados por 10 mapas de flujo óptico pasados y, de manera autoregresiva, genera las estimaciones de los siguientes cinco flujos a partir de las representaciones procesadas por la red recurrente ConvGRU y el mecanismo de decisión.

Para determinar la configuración más adecuada de los hiperparámetros α, β y γ de la función de pérdida descrita en la Sección 8.2.2, se entrenó el modelo explorando distintas combinaciones de dichos valores. Los resultados completos del barrido de hiperparámetros (tablas y métricas por configuración) se presentan en el Anexo A.1. Asimismo, se evaluaron dos alternativas respecto al uso del codificador previamente entrenado en el autoencoder: mantener sus pesos congelados o permitir su ajuste durante el entrenamiento completo.

La comparación, resumida en la Tabla 7, mostró que la variante con el codificador congelado entregaba un entrenamiento más estable y un mejor equilibrio entre magnitud y dirección del flujo, alcanzando valores de EPE $= 1.20$ y AE $= 1.90°$ para las configuraciones de $\alpha = 0.8$,

$\beta = 0.2$ y $\gamma = 0.05$. En contraste, la opción sin congelar tuvo valores mínimos de EPE en torno a 1.42 y un valor de AE de $2.72°$ para configuraciones de $\alpha = 0.9$, $\beta = 0.2$ y $\gamma = 0.01$. Con base en estos resultados, en todos los experimentos posteriores se adoptó la configuración con codificador congelado junto con sus parámetros de α, β y γ, asegurando que el modelo aproveche las representaciones latentes previamente aprendidas sin comprometer la estabilidad del entrenamiento. La tabla completa con todas las combinaciones exploradas de α, β y γ se incluye en el material suplementario.

Codificador	α	β	γ	**EPE (px)**	**AE (°)**
Congelado	0.8	0.2	0.05	1.20	1.90
No congelado	0.9	0.2	0.01	1.42	2.72

Tabla 7: Comparación de las mejores configuraciones de α, β y γ de la función de pérdida, con encoder congelado y sin congelar. Se muestran los valores de EPE (px) y AE (°) para las combinaciones de hiperparámetros más representativas.

Una vez definidos los hiperparámetros, se procedió a entrenar el modelo PE con los flujos ópticos del conjunto KITTI y a evaluarlo en el conjunto de validación, considerando un horizonte de 10 pasos de predicción. Para estimar su precisión, se calcularon los valores promedio de EPE y AE en cada uno de los frames generados, a partir de la comparación entre el flujo estimado por el modelo y el flujo real, tal como muestra la Tabla 8. A partir de esta tabla, se observa una tendencia creciente en ambas métricas a medida que aumenta el horizonte de predicción. En particular, se observa que EPE aumenta de 1.23 píxeles en el primer paso hasta 11.107 píxeles en el paso 10, mientras que el error angular se incrementa de $1.142°$ a $32.74°$ en el mismo intervalo. Este comportamiento refleja la acumulación progresiva de errores en la predicción.

predicción	**EPE (px)** $\downarrow$	**AE (°)** $\downarrow$
1	1.233 $\pm$ 0.0201	1.142 $\pm$ 0.0271
2	2.512 $\pm$ 0.0232	1.935 $\pm$ 0.0222
3	3.544 $\pm$ 0.0190	3.751 $\pm$ 0.0711
4	4.621 $\pm$ 0.0230	4.932 $\pm$ 0.0391
5	6.432 $\pm$ 0.0075	6.456 $\pm$ 0.0224
6	7.502 $\pm$ 0.0106	10.93 $\pm$ 0.0280
7	8.621 $\pm$ 0.0681	18.09 $\pm$ 0.0380
8	9.044 $\pm$ 0.0640	29.16 $\pm$ 0.0331
9	9.496 $\pm$ 0.0608	30.09 $\pm$ 0.0439
10	11.107 $\pm$ 0.0725	32.74 $\pm$ 0.0442

Tabla 8: Evaluación de EPE (px) y AE (°) para 10 predicciones $\pm$ desviación estándar.

Luego, para establecer un umbral que determine el horizonte temporal de predicción confiable, se tomó como referencia el protocolo de evaluación oficial del benchmark KITTI. Dicho protocolo considera que un píxel se encuentra correctamente estimado cuando el valor promedio de EPE es ≤ 4. Valores superiores a este umbral implican una pérdida significativa de fidelidad en la estimación del flujo óptico. Este criterio ha sido adoptado ampliamente en la literatura como estándar de evaluación del flujo óptico [55] [40] [59], ya que permite identificar el rango en el cual se mantiene la fidelidad entre el flujo estimado por el modelo y el flujo óptico real.

Los resultados de la Tabla 8 muestran que durante las primeras 3 predicciones el valor de EPE se mantiene por debajo del umbral con valores de EPE entre 1.2 y 3.5 píxeles, lo que indica que las estimaciones del flujo óptico en esos horizontes temporales conservan una alta fidelidad en relación al flujo verdadero. A partir de la cuarta predicción, se supera el valor de referencia para EPE con un valor de 4.621 y continúa aumentando progresivamente en los pasos siguientes, alcanzando valores superiores a 11 px en la décima predicción. Este incremento se acompaña de un deterioro en el AE, que pasa de 1.14° en la primera predicción a más de 32° en la décima. Estos resultados permiten concluir que el horizonte de predicción confiable del modelo se limita a las tres primeras predicciones, mientras que a partir de la cuarta la fidelidad de las estimaciones se ve comprometida.

La Figura 8 representa una visualización entre los flujos ópticos reales y los generados por el modelo para las primeras 5 predicciones. En la fila superior, se ilustran los mapas de color del flujo óptico real en los instantes iniciales ($n = 1$ hasta $n = 10$) que es la entrada que alimenta al modelo. Luego, las tres filas inferiores representan los resultados obtenidos por el modelo para los pasos de predicción $n = 11 \ldots 15$, donde la fila superior corresponde al flujo óptico real, fila intermedia corresponde al flujo estimado por el modelo, y la tercera representa el mapa de error $|real - pred|$, calculado como la magnitud de la diferencia vectorial entre los vectores de flujo optico real y los predichos y visualizado en escala logarítmica. En este último, los tonos verdes indican alta similitud entre flujo real y predicho, mientras que los tonos amarillos resaltan regiones de mayor discrepancia.

A partir de esta figura se aprecia que el modelo es capaz de capturar la estructura global del flujo, reproduciendo de manera adecuada la dirección general del movimiento de los objetos en la escena. Sin embargo, se observan inconsistencias en los bordes, donde las predicciones aparecen ligeramente difuminadas , lo que se resalta en las líneas amarillas en el mapa de error.

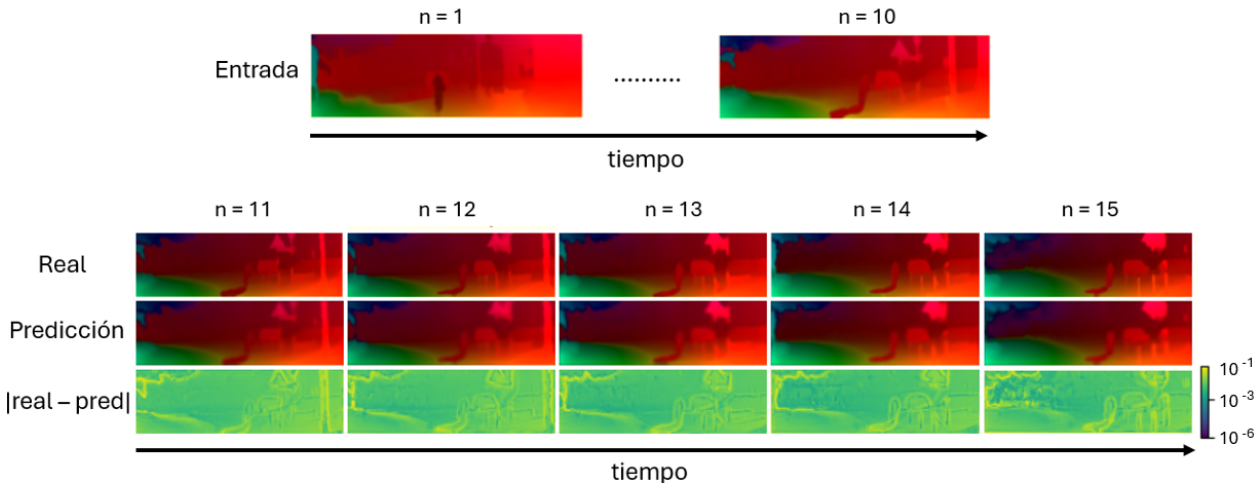


Figura 8: Comparación cualitativa de los cinco primeros pasos de predicción del flujo óptico con el modelo pre encoder: Fila superior "Entrada": muestra los mapas de color del flujo real en los primeros pasos de la secuencia (n $= 1 \ldots 10$). Filas centrales (n $=$ 11–15): fila superior: mapas de color del flujo real en cada instante de predicción. Fila intermedia: predicción de flujo realizada por el modelo para los mismos instantes de tiempo. Fila inferior: $|real–pred|$: magnitud de la diferencia pixel a pixel entre flujo real y predicción, gráficada en escala logarítmica, en amarillo se resaltan las zonas de mayor error o diferencias entre la imagen de flujo real con la imagen de flujo predicho. A lo largo de los pasos n $= 11$ a n $= 15$ se aprecia que el modelo reproduce correctamente la estructura global del flujo, mientras que los mapas de error evidencian la acumulación progresiva de discrepancias, especialmente en los bordes de objetos y transiciones de alto contraste. Es importante destacar que a partir del n $= 13$ comienza a evidenciarse más detalles en los contornos debido a la acumulación de error autoregresivo, lo que se hace más visible al observar las zonas de color amarillo en el mapa de error.

Tabla 9: Resultados de generalización en el dataset Caltech para los primeros cinco pasos de predicción. Se reportan el *End-Point Error* (EPE) en píxeles y el error angular (AE) en grados como valores medios.

predicción	EPE (px)	AE (°)
1	2.58	1.64
2	3.80	3.69
3	5.87	5.86
4	7.02	8.46
5	11.05	10.76

Tabla 10: Resultados de generalización en el dataset KTH para los primeros cinco pasos de predicción. Se reportan el *End-Point Error* (EPE) en píxeles y el error angular (AE) en grados como valores medios.

predicción	EPE (px)	AE (°)
1	0.56	0.87
2	1.72	1.57
3	3.23	2.92
4	5.93	4.71
5	5.61	6.21

Adicionalmente, con el fin de evaluar la capacidad de generalización del modelo, se evaluó el desempeño del método para las primeras 5 predicciones en dos conjuntos de datos: KTH-Action y Caltech Pedestrian.

La Tabla 9 muestra los resultados obtenidos para el conjunto de imágenes Caltech que esta compuesto de imágenes de entornos urbanos, a partir de esta tabla es posible observar que se obtiene un $EPE = 2.58$ px en el primer paso y un valor de $EPE = 11.05$ px en el quinto, mientras que el error angular se incrementa desde $1.64°$ hasta $10.76°$ en el mismo intervalo, lo que sugiere que el modelo es capaz de generalizar en las primeras dos predicciones pero desde la cuarta hasta adelante genera predicciones poco confiables. .

De manera complementaria, se evaluó la generalización en el conjunto de datos KTH - Action, que está compuesto por secuencias de acciones humanas en entornos controlados. Los resultados presentados en la Tabla 10 muestran valores de EPE entre 0.56 para la primera predicción y 5.61 píxeles en la quinta, junto con errores angulares que varían entre $0.87°$ a $6.21°$ en los primeros cinco pasos. A diferencia de Caltech, en KTH el modelo alcanza un desempeño significativamente mejor, lo cual se explica por la simplicidad de las escenas y la menor variabilidad visual del conjunto de imágenes en relación a Caltech. La evaluación del modelo PE realizada para Caltech y KTH ponen de manifiesto que éste puede generar predicciones pero en horizontes temporales cortos.

En resumen, es posible determinar el modelo es capaz de extraer características espacio temporales a partir de vectores de flujo óptico y generar predicciones coherentes a partir de ellos. Si bien, se evidenció que el modelo puede generar predicciones confiables para 3 pasos futuros en el conjunto de imágenes de KITTI, punto a partir del cual comienza a perder la capacidad de reflejar la evolución real de la secuencia, al evaluar la generalización del modelo en otros conjuntos de imágenes no vistos en el entrenamiento, ese rango disminuyó a 2 pasos temporales futuros, lo que demostraría una limitancia del modelo al no poder reflejar la evolución del movimiento de los objetos en set de imáganes no vistas previamente.

Por lo tanto, es posible concluir que los resultados del modelo Pre Encoder (PE) confirman que el uso del flujo óptico como entrada permite capturar de manera efectiva las dependencias espacio temporales y generar predicciones coherentes en horizontes cortos; el análisis de EPE, AE mostró que en general el modelo mantiene un desempeño confiable hasta aproximadamente 2 pasos futuros, punto a partir del cual comienza a perder la capacidad de generalizar y reflejar la evolución del movimiento de los objetos presentados en la secuencia de imágenes. Estos hallazgos validan la hipótesis planteada en el Objetivo 1 y, al mismo tiempo, evidencian las limitaciones del enfoque, lo que motiva la exploración de arquitecturas alternativas para mejorar el modelo.

9.2. Objetivo 2: Evaluación del modelo Post Latent Space (PLS)

Para comprender cómo varía el campo de flujo óptico según la dimensión de reescalado de la imagen, aplicamos un factor de escala N, de modo que cada dimensión (ancho y alto) se reduce a $1/N$ de su tamaño original. De esta forma, es posible observar de forma sistemática el impacto que tiene la resolución de entrada en la densidad y fidelidad del flujo óptico.

En la Figura 9 se muestran los campos de flujo óptico reales tras aplicar un pre-muestreo de la imagen original tras aplicar un factor de escala de 1, 2 y 4, respectivamente, esto corresponde a reducir la resolución original a $1/1$, $1/2$ y $1/4$ de su tamaño original. Para la escala $= 1$ (columna izquierda) sin pre-muestreo, el flujo presenta en la resolución completa 375×1242, capaz de capturar movimientos destacados como el vehículo, ciclista, etc. Al emplear la escala $= 2$ (columna central), la imagen de entrada se reduce a 187×621 ; en este caso se conservan las corrientes principales, aunque con la mitad de densidad y un ligero suavizado en zonas homogéneas. Con la escala $= 4$ (columna derecha), al pre-muestrear a 93×310, el flujo pierde casi por completo los detalles finos, quedando únicamente un patrón grueso y espaciado de vectores.

En conjunto, esta figura sirve de referencia para entender hasta qué punto el pre-muestreo y el factor de escala afecta la fidelidad del flujo: cuanto mayor es la reducción, más rápido resulta el cálculo, pero menor la resolución y cobertura del campo de desplazamientos.

Tras evaluar en la Figura 9 la influencia del premuestreo sobre la densidad y fidelidad del flujo óptico, resulta natural preguntarse cómo estos mismos factores de escala afectan las representa-

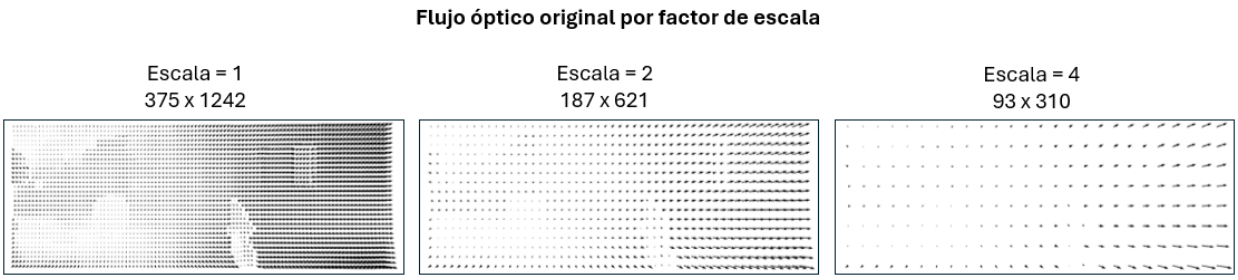


Figura 9: Campo de flujo óptico original calculado con Farneback tras pre-muestrear la imagen de entrada con factores de escala 1, 2 y 4, correspondientes a resoluciones de 375×1242, 187×621 y 93×310 píxeles. A medida que aumenta el factor de escala, la densidad de vectores disminuye y los detalles finos del movimiento se desvanecen, ilustrando el compromiso entre eficiencia y fidelidad en la estimación del flujo óptico.

ciones internas de este modelo. Es por esto que se analizó el impacto del reescalado de la imagen de entrada en las salidas latentes de cada capa del encoder.

En la Figura 10 se comparan las salidas de la imagen original y sus mapas latentes en cada una de las tres capas de convolución 2D del encoder, tras aplicar factores de escala N $= 1$, 2 y 4 durante el preprocesado.

En particular para N $= 1$ (columna izquierda), la entrada conserva sus dimensiones completas de 375×1242 px y el encoder genera sucesivamente mapas de 188×621, 94×311 y 47×156px, preservando la mayoría de bordes y texturas.

Con N $= 2$ (columna central), la imagen se reduce a 187×621 px, lo que produce mapas de 94×311, 47×156 y 24×78 px; en estas resoluciones intermedias ya se aprecia cierta pérdida de detalle en contornos finos y zonas texturizadas.

Al usar N $= 4$ (columna derecha), la entrada de 93×310 px da lugar a mapas de 47×155, 24×28 y 12×39 px; en estas dimensiones tan reducidas se observa un marcado pixelado y la desaparición de estructuras pequeñas.

En resumen, al analizar como varía cada capa de la convolución segun el factor de escala aplicado a la imagen, es posible distinguir que si bien un factor de escala mayor acelera el cálculo al trabajar con tensores más pequeños, penaliza la fidelidad espacial y dificulta la preservación de detalles finos en las activaciones latentes.

sobretodo al analizar la ultima capa de 64 canales es posible observar que para los 3 tipos de reescalados, la representación se vuelve tan abstracta que apenas se aprecian formas coherentes, debido a que ya no dispone de información suficiente para resolver texturas finas ni bordes curvos, sino que solo es capaz de generar un patrón muy grueso y disperso. lo que sugiere que a medida que se avanza en las capas del encoder y se aumenta el factor de escala, la red pasa de capturar detalles finos a modelar únicamente la geometría más global, sacrificando riqueza espacial en favor de eficiencia computacional.

En la práctica, ese colapso de detalles de la imagen implica que la red pierde capacidad para discriminar pequeños desplazamientos de flujo (por ejemplo, movimiento de ruedas o ramas) y acaba basándose solo en patrones gruesos de movimiento global. Por ello, aunque escalar a N = 4 acelera drásticamente el cómputo, se sacrifica la capacidad de la red para modelar flujos ópticos de pequeña magnitud o cambios de dirección sutiles, lo cual puede comprometer la exactitud de predicciones de movimiento en escenas complejas.

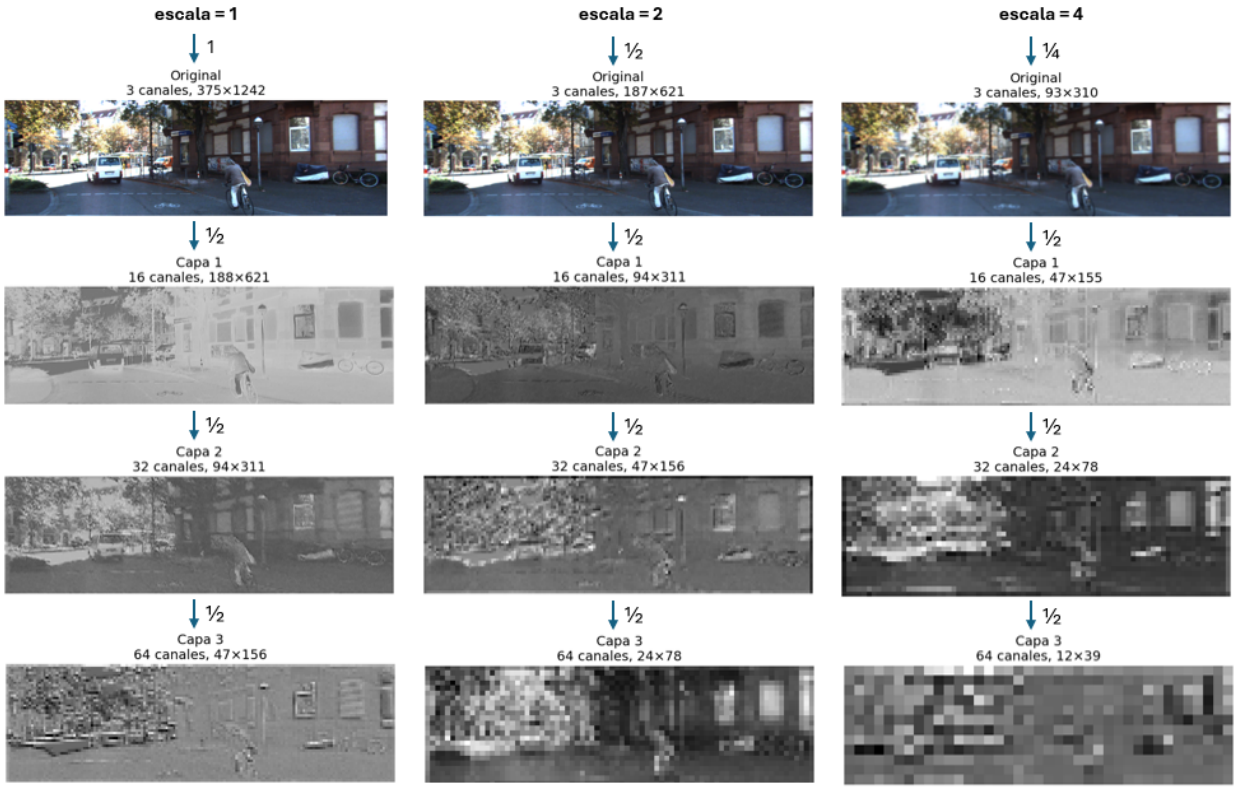


Figura 10: Degradación de la resolución en el encoder según el factor de escala y el stride de convolución. Cada columna muestra el pre-muestreo de la imagen de entrada aplicando un factor de escala($\times 1$, $\times \frac{1}{2}$, $\times \frac{1}{4}$); las flechas "$\downarrow \frac{1}{2}$" indican la reducción de resolución por 'stride=2' en cada bloque convolucional. Los títulos especifican canales y dimensiones resultantes en la imagen original y en las capas 1, 2 y 3.

Escala = 1 (columna izquierda): la imagen original conserva sus dimensiones completas (375×1242 px) y el downsampling en el encoder produce sucesivamente mapas de 188×621, 94×311 y 47×156 px.

Escala = 2 (columna central): al reducir la entrada a la mitad (187×621 px), las capas intermedias pasan a 94×311, 47×156 y 24×78 px, con pérdida de detalle visible en bordes finos y texturas.

Escala = 4 (columna derecha): la entrada cae a 93×310 px, y el encoder genera mapas de sólo 47×155, 24×78 y 12×39 px; en estas resoluciones tan bajas ya se aprecia un fuerte pixelado y la desaparición de estructuras pequeñas.

De este modo, el parámetro de escala controla directamente la resolución de la imagen en todas las etapas posteriores: un factor de escala alto acelera el cómputo al tratar con tensores más pequeños, pero sacrifica la fidelidad espacial y dificulta la preservación de detalles finos en las activaciones latentes.

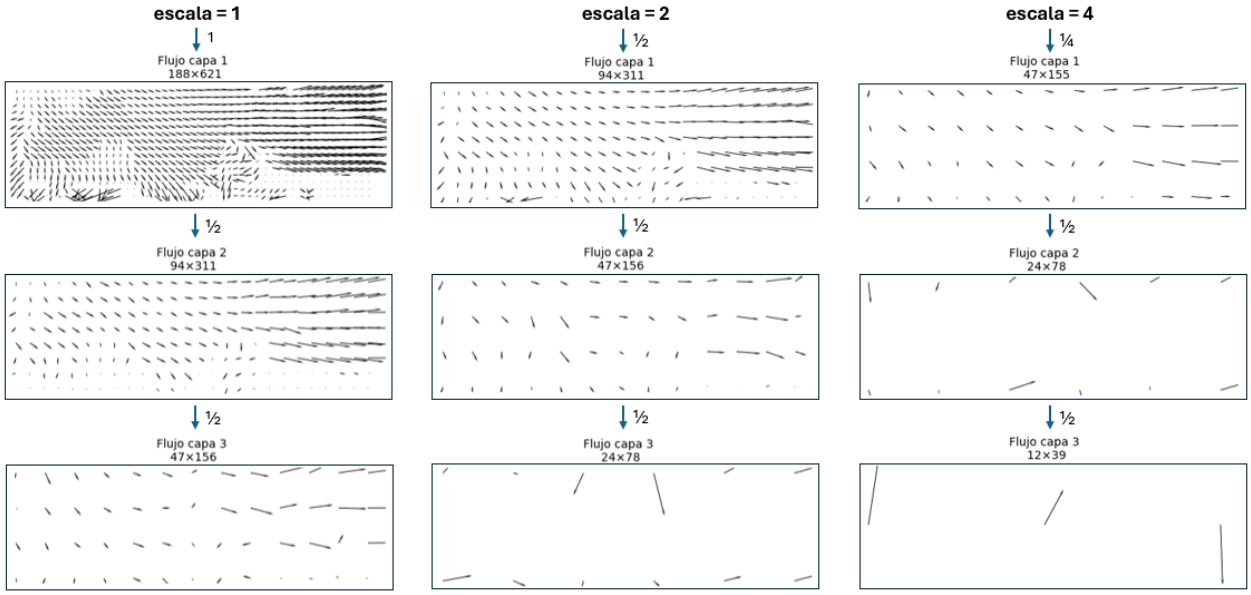


Figura 11: Campos de flujo óptico obtenidos sobre las activaciones de las tres capas del encoder para diferentes factores de escala. Cada columna corresponde a la entrada pre procesada con *factor de escala* = 1, 2 y 4, respectivamente, y cada fila muestra el flujo extraído sobre la salida de la Capa 1, Capa 2 y Capa 3 (con las resoluciones indicadas). Se aprecia la degradación progresiva del detalle del flujo al crecer la reducción de resolución.

Además, al calcular el flujo óptico directamente sobre los mapas latentes de cada capa (fig 11), se observa cómo la degradación de resolución inducida por el factor de escala impacta también a los vectores de movimiento internos del encoder:

En la escala N = 1 (columna izquierda), los flujos en las capas 1 y 2 reproducen la dirección y magnitud de los desplazamientos originales; los vectores marcan con claridad las direcciones principales del movimiento aunque los bordes no aparecen tan nítidos como en la imagen original. En cuanto a la capa 3, aunque la densidad es baja, aún se intuyen las orientaciones globales de los desplazamientos, sin que se aprecien detalles finos.

Al duplicar el factor de escala a N = 2 (columna central), la capacidad de capturar movimientos finos se reduce drásticamente: en la capa 1 las direcciones dominantes siguen siendo visibles, pero los vectores aparecen más dispersos y con menor densidad. En la capa 2 las zonas de interés se reducen a parches de flujo, donde sólo persiste la señal de movimiento más marcado. y en la capa 3 el flujo solo se pueden observar indicios de las trayectorias principales.

Finalmente cuando se observa el flujo optico con factor de escala 4 (columna derecha) la pre-reducción de 47×155 empobrece a tal punto el flujo de la capa 1 que solo es posible observar trazos gruesos de movimiento, donde los contornos de objetos apenas se distinguen. Por último en las capas 2 y 3 el flujo se reduce a tal nivel que los vectores se ven aislados, sin continuidad ni forma reconocible.

Con todo lo anterior, es posible concluir que, incluso en la resolución completa (N = 1), las capas del codificador pierden definición de forma progresiva y, a medida que el submuestreo se vuelve más agresivo, la red solo conserva señales de movimiento muy gruesas, sin generar detalles finos en ningún nivel. Esta degradación irreparable de la información necesaria para estimar con precisión los vectores de desplazamiento provoca que, al reducirse el tamaño de los mapas latentes, el flujo óptico pierda densidad, detalle angular y coherencia en zonas de movimiento claves para la deteccion de objetos, impidiendo la obtención de un campo lo suficientemente fiel para que a partir d ello se puedan generar predicciones futuras. Sin una estimación robusta y completa en las capas profundas, este modelo no dispone de las bases temporales y espaciales necesarias para predecir flujos futuros con confianza, ya que la reducción progresiva de la resolución descarta la información crítica para una predicción confiable.

9.3. Objetivo 3: Evaluación del modelo ACCLIP

9.3.1. Ajuste de hiperparámetros

Para evaluar el efecto del balance entre MSE y SSIM se exploraron distintos valores de α, que controla la ponderación relativa entre MSE y SSIM en la función de pérdida descrita en la sección 8.4.2. El objetivo principal fue determinar cómo dichos valores de α afectan la consistencia estructural de las imágenes predichas a medida que aumenta el número de predicciones. Además, para identificar cómo afecta la incorporación del flujo óptico en la función de pérdida se exploraron varios valores de β. Las superficies resultantes del promedio de SSIM se muestran en el mapa de contornos de la Figura 12; en ella se presentan solo dos escenarios: la Figura 12(a) con $\beta = 0.0$ (sin término de flujo) y la Figura 12(b) con $\beta = 0.7$, correspondiendo esta última al caso de mejor desempeño. Los resultados para los demás valores de β se incluyen en la Sección de Anexos, Figura A.1. La escala de valores de SSIM está representada por la barra de color ubicada a la derecha del gráfico. En general, los colores más claros (en la parte superior de la barra) indican valores más altos de SSIM, lo que corresponde a una mayor similitud estructural, mientras que los colores más oscuros representan valores más bajos, es decir, menor similitud con la secuencia real.

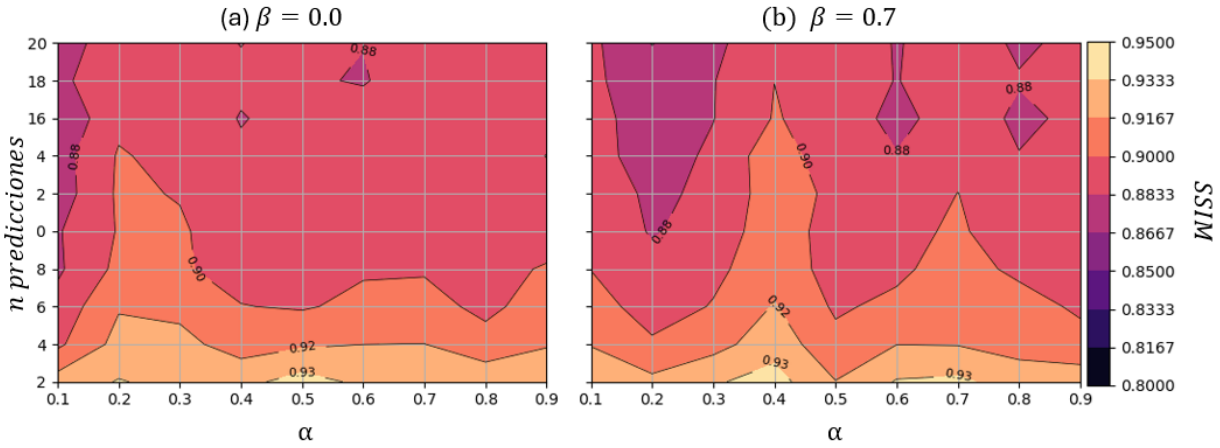


Figura 12: Mapa de contornos del SSIM medio en función de α y el horizonte de predicción . (a) SSIM sin el parámetro de flujo óptico en la función de pérdida ($\beta = 0.0$), mientras que (b) incorpora el parámetro de flujo óptico con ($\beta = 0.7$) correspondiente al hiperparámetro con mejor desempeño. La barra de colores a la derecha representa el valor de SSIM: los tonos más oscuros corresponden a valores bajos y los más claros a valores altos.

En ambas condiciones se observa un descenso del SSIM a medida que aumenta el número de predicciones, lo cual es esperable dada la acumulación de error en predicciones a largo plazo. Sin embargo, es posible observar patrones distintos según los valores de α. Cuando $\beta = 0.0$ (Fig. 12(a), los valores altos de SSIM se concentran principalmente en $\alpha = 0.2$ en horizontes de predicción cortos, es decir, de 1 a 3 imágenes, y a medida que α aumenta, el SSIM tiende a disminuir rápidamente en predicciones más largas, lo que sugiere que dar mayor peso al SSIM no compensa la ausencia de información de movimiento en la función de pérdida. Por el contrario, para $\beta = 0.7$ (Fig. 12b) muestra dos regiones claramente definidas donde el SSIM se mantiene consistentemente alto: una alrededor de $\alpha = 0.7$ y otra en $\alpha = 0.4$ donde mantiene un SSIM > 0.93 en las primeras predicciones; de igual forma es posible observar que mantiene un rango más amplio de longitudes de predicción superiores a 0.9 cercano a 18 predicciones futuras.

9.3.2. Evaluación del modelo con los mejores hiperparámetros

Luego de identificar los valores óptimos $\alpha = 0.4$ y $\beta = 0.7$, se evaluó la capacidad del modelo para generar predicciones a largo plazo utilizando el conjunto de datos KITTI. Los resultados presentados en la Tabla 11 muestran que para predicciones a corto plazo (2 y 4 imágenes), las métricas SSIM y PSNR son más altas y LPIPS es más bajo, indicando que el modelo preserva mejor los detalles estructurales y perceptuales inmediatos. Sin embargo, al extender el horizonte de predicción a 10, 12 y 14 imágenes, se observa una disminución gradual en SSIM y PSNR y un leve incremento en LPIPS, reflejando cómo los errores acumulados afectan progresivamente la calidad

perceptual y la fidelidad píxel a píxel. A pesar de esto, incluso con 14 imágenes predichas, el SSIM permanece relativamente alto, sugiriendo que el modelo mantiene la consistencia estructural.

Predicciones	**SSIM↑**	**PSNR↑**	**LPIPS↓**
2	0.9361 ± 0.0025	26.8498 ± 0.1837	0.0856 ± 0.0032
4	0.9067 ± 0.0026	25.5064 ± 0.1159	0.0944 ± 0.0033
6	0.9002 ± 0.0023	25.2136 ± 0.0939	0.0970 ± 0.0027
8	0.8997 ± 0.0018	25.1331 ± 0.0886	0.0959 ± 0.0020
10	0.8942 ± 0.0019	24.6696 ± 0.0723	0.1011 ± 0.0022
12	0.8921 ± 0.0014	24.8942 ± 0.0704	0.1089 ± 0.0704
14	0.8824 ± 0.0015	24.2341 ± 0.0716	0.1066 ± 0.0019

Tabla 11: Evaluación de SSIM, PSNR y LPIPS para diferentes número de predicciones , usando los mejores hiper parámetros($\alpha = 0.4$ and $\beta = 0.7$).

La figura 13 muestra una comparación visual de las imágenes reales y predichas del conjunto de datos KITTI en un horizonte temporal de 23 pasos futuros; en la fila superior de la imagen se muestra la secuencia de entrada de 10 imágenes, en la fila intermedia se muestra la secuencia de imágenes reales que debería predecir el modelo y en la fila inferior la predicción realizada para la predicciones 21 a la 23. Al realizar una evaluación comparativas entre las imágenes reales y las predichas, se puede observar que a mayores distancias temporales (predicción 21 a 23), la reproducción de los detalles se vuelve menos precisa, comprometiendo la relación entre la coherencia de la escena y detalles locales. Los artefactos producidos por el modelo están marcados con recuadros amarillos para indicar las áreas donde la predicción se desvía de la estructura visual esperada. En particular, es posible observar que el modelo falla cuando se genera una imagen predicha con regiones oscuras como sombras. En conclusión, el modelo ofrece resultados más fiables hasta aproximadamente diez cuadros. Sin embargo, conforme aumenta el número de predicciones, se hace algo más difícil mantener la coherencia espacial (SSIM, PSNR) así como también la calidad perceptual (LPIPS), lo que conduce a una degradación en la calidad de predicción en horizontes a largo plazo.

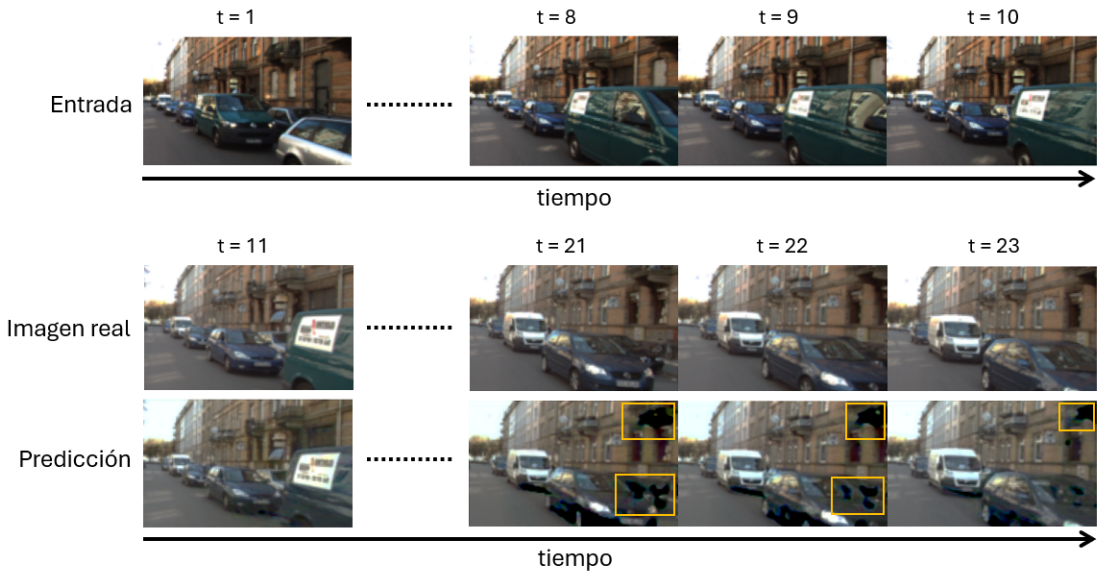


Figura 13: Resulados modelo ACCLIP: Predicciones en el KITTI-Dataset La fila superior (t=1 a t=10) representa la secuencia de entrada, la fila intermedia (t=11 a t=23) muestra el objetivo correspondiente a la secuencia real, y la fila inferior presenta las imágenes predichas por ACCLIP. Los recuadros amarillos observados en la predicción, resaltan los artefactos generados por el modelo.

9.3.3. Comparación con modelos del estado del arte

Para identificar si el modelo es capaz de generalizar bien en cualquier escenario, se evaluó el modelo con dos conjunto de imágenes que no fueron visualizados durante el entrenamiento: KTH-Action y Caltech Pedestrian dataset. En cada caso se presentan primero los resultados cuantitativos mediante métricas objetivas (SSIM, PSNR, LPIPS) presentadas en una tabla comparativa, y luego una evaluación cualitativa con imágenes representativas.

Conjunto de datos KTH Actions

La Tabla 12 presenta los resultados en el conjunto de datos KTH-Action comparando ACCLIP con distintos métodos del estado del arte. Como es posible observar, los resultados indican que ACCLIP ACCLIP logra el mejor desempeño en términos de similitud estructural (SSIM), tanto en escenario de predicción a corto plazo (10→20, SSIM $= 0.967$) como en el de horizonte más largo (10→40, SSIM $= 0.974$). Estos resultados confirman la capacidad del modelo para preservar la estructura espacial y la coherencia de las secuencias en el tiempo.

Sin embargo, cuando se evalúa la fidelidad a nivel de intensidad de los píxeles (PSNR), ACCLIP se sitúan por debajo de varios modelos del estado del arte en la evaluación 10→20 e inferior a SimVP en la evaluación 10→40. Esto indica que, aunque ACCLIP conserva la geometría global de la escena, existen variaciones entre la imagen predicha y la real cuando son comparadas pixel a pixel, lo que se traduce en una reconstrucción de texturas menos precisa y con mayor nivel de ruido.

Estos resultados sugieren que, aunque la secuencia predicha conserva correctamente la estructura global de la escena, en los detalles como bordes y texturas tienden a ser mas borrosos o ruidosos, detalles reflejados por los valores de PSRN. Esto indica que ACCLIP tiende a privilegiar la coherencia estructural sobre la fidelidad pixel a pixel, lo cual resulta ventajoso en tareas donde la forma y el movimiento son más relevantes que la precisión absoluta de cada textura.

La inspección visual de las predicciones, Figura 14 revela discrepancias entre la imagen real y la predicha señaladas con flechas amarillas. En estas regiones aparecen artefactos como siluetas que no están presentes en la imagen real. Estos artefactos son los que afectan directamente al valor del PSRN al introducir diferencias en la intensidad pixel a pixel.

Es probable que estos artefactos provengan de imprecisiones en el cálculo del flujo óptico de la función de pérdida, sobre todo en zonas con contornos o movimientos , donde el modelo tiende a generar bordes distorsionados o duplicados. Aun así, la estructura general de la acción se conserva, lo que explica por qué el SSIM se mantiene elevado pese a la pérdida de detalle en áreas específicas.

Los experimentos en KTH muestran que ACCLIP consigue reproducir de manera convincente la dinámica general de las acciones humanas, manteniendo la continuidad y la forma de los movimientos a lo largo del tiempo. Por otro lado, la inspección visual de las secuencias de predicción revela que, en zonas de bordes o movimientos, aparecen pequeños artefactos locales, como contornos duplicados o difusos, que explican el bajo valor de PSNR. Como conclusión, el modelo parece privilegiar la conservación de la coherencia global de la escena aunque ello implique sacrificar parte del detalle. Este equilibrio puede ser útil en escenarios donde lo importante es captar la acción y su ritmo, antes que la exactitud minuciosa de cada píxel.

Método	**KTH (10 → 20)**		**KTH (10 → 40)**	
	SSIM↑	**PSNR↑**	**SSIM↑**	**PSNR↑**
ConvLSTM [84]	0.712	23.58	0.639	22.85
SAVP [85]	0.746	25.38	0.650	23.97
VPN [86]	0.746	25.23	-	-
DFN [87]	0.794	27.26	0.652	23.21
fRNN [88]	0.771	26.12	0.678	23.77
Znet [89]	0.817	27.58	-	-
MCnet [71]	0.804	25.95	0.730	23.89
SV2Pi [90]	0.826	27.56	0.778	25.92
SV2Pv [90]	0.820	27.52	-	-
PredRNN [37]	0.839	27.55	0.739	25.37
VarNet [91]	0.843	28.48	0.739	25.31
SVAP-VAE [85]	0.852	28.75	-	-
PredRNN++ [92]	0.859	28.90	0.746	25.44
MSNets [93]	0.879	29.30	0.810	25.14
E3d-LSTM [22]	0.879	29.81	0.851	27.94
STMFANet [94]	0.881	30.00	-	-
SimVP [28]	0.905	**33.72**	0.886	**32.93**
ACCLIP	**0.967**	27.80	**0.974**	28.00

Tabla 12: Comparación entre ACCLIP y los modelos de predicción de vídeo más avanzados en el conjunto de datos KTH-Action, mostrando SSIM y PSNR para los imágenes 10→20 y 10→40. Las flechas hacia arriba indican que valores más altos indican un mejor rendimiento..

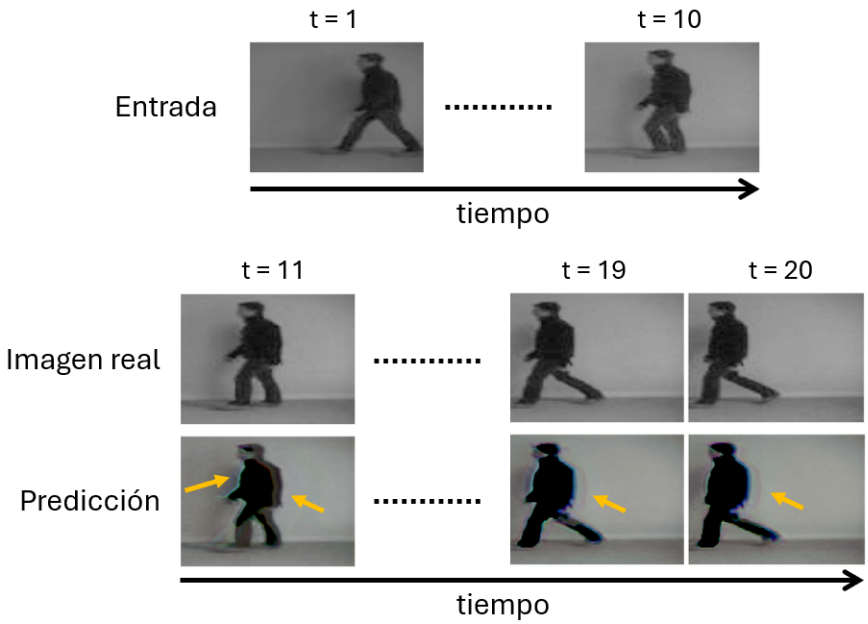


Figura 14: Resulados modelo ACCLIP: Predicciones en KTH-Action Dataset La fila superior (t=1 a t=10) muestra las imágenes de entrada, la fila intermedia (t=11 a t=20) corresponde a las imágenes reales, y la fila inferior muestra la secuencia predicha por ACCLIP. Las flechas amarillas destacan discrepancias o artefactos en la silueta del sujeto.

Conjunto de datos Caltech - Pedestrian

El desempeño de ACCLIP también se evaluó en el conjunto Caltech Pedestrian considerando una única predicción hacia el futuro.

La Tabla 13 presenta los resultados comparativos con otros métodos del estado del arte. Como se muestra en la tabla, ACCLIP obtiene los valores más altos de SSIM (0.987) superando a los modelos del estado del arte y mostrando su capacidad para mantener la estructura y la coherencia de la escena. En cuanto a PSNR (32.4) aunque el valor es competitivo, se mantiene por debajo de SimVP (33.1), lo que sugiere que pese a que la forma general de la iamgen se conserva, aún existen pequeñas diferencias a nivel de detalle fino y fidelidad pixel a pixel. Estos resultados indican que ACCLIP puede generar predicciones en entornos dinámicos y urbanos, logrando un equilibrio entre precisión estructural y apariencia visual.

Al generar una inspección visual en el conjunto Caltech (Figura 15) se observa que a simple vista la predicción de ACCLIP es muy similar a la imagen real, resultado que concuerda con los altos valores de SSIM reportados en la Tabla 13. Sin embargo, al observar la columna derecha que representa las diferencias entre la imagen predicha y la real, se distinguen zonas mas claras que marcan las áreas donde la predicción y la imagen real no coinciden. Es posible distinguir que estas discrepancias se concentran principalmente en regiones oscuras y con sombras, donde el modelo tiende a generar artefactos o pérdidas de detalle, un comportamiento ya observado en las evaluaciones sobre KITTI.

En síntetis, ACCLIP demostró un desempeño competitivo al compararse con modelos de del estado del arte como SimVP, E3D-LSTM y STMFANet, alcanzando los valores más altos de SSIM en todos los conjuntos de datos evaluados. Confirmando su capacidad para mantener la estructura y la dinámica temporal de las secuencias, incluso en entornos complejos como el urbano. Por otro lado, es posible distinguir que el modelo presenta limitaciones claras, como generar artefactos en zonas con bordes y sombras, perdiendo definición en contornos. Este comportamiento explica por qué el PSNR se mantiene por debajo de algunos métodos, lo que puede estar relacionado con el uso del flujo óptico en la función de pérdida, que prioriza la coherencia del movimiento por sobre la fidelidad pixel a pixel.

En conclusión, ACCLIP tiene potencial como modelo de predicción de secuencias, especialmente en aplicaciones donde lo esencial es anticipar patrones de movimiento más que reconstruir texturas con exactitud. Estos resultados instan a generar trabajos futuros orientados a mejorar el equilibrio entre coherencia estructural y fidelidad visual, clave tanto para el avance teórico en codificación predictiva como para aplicaciones prácticas en visión por computador.

Método	Caltech Pedestrian (10 → 1)	
	SSIM↑	PSNR↑
MCnet [71]	0.879	-
DVF [42]	0.897	26.2
CtrlGen [95]	0.900	26.5
PredNet [96]	0.905	27.6
SDC-Net [97]	0.918	-
rCycleGan [98]	0.919	29.2
ContextVP [99]	0.921	28.7
DPG [100]	0.923	28.2
CrevNet [101]	0.925	29.3
STMFANet [94]	0.927	29.1
SimVP [28]	0.940	**33.1**
ACCLIP	**0.983**	32.4

Tabla 13: Comparación entre ACCLIP y los modelos de predicción de vídeo en el conjunto de datos Caltech Pedestrian, mostrando SSIM y PSNR para las imágenes 10→1, donde valores más altos indican un mejor rendimiento.

Figura 15: Resulados modelo ACCLIP: Predicciones en Caltech-Pedestrian Dataset Visualización de resultados en el conjunto Caltech Pedestrian para una predicción a un paso. La columna izquierda muestra la última imagen de entrada al modelo, mientras que las columnas centrales corresponden a la imagen real y a la predicción generada por ACCLIP. La columna derecha presenta el mapa de diferencias entre ambas, donde las discrepancias se resaltan con colores más claros.

9.4. Objetivo 4: Evaluar la capacidad predictiva de las arquitecturas PE, PLS y ACCLIP.

Este objetivo buscó responder a la pregunta de investigación: *¿En qué punto de la arquitectura conviene integrar el flujo óptico para obtener predicciones confiables en escenas urbanas?* Para ello se compararon tres alternativas: incorporarlo directamente en la entrada (*arquitectura PE*), calcularlo en el espacio latente a la salida del codificador (*arquitectura PLS*) o bien usarlo como término adicional en la función de pérdida (*arquitectura ACCLIP*). Las características principales de cada modelo se resumen en la tabla 14.

Modelo	**Punto de integración FO**	**Salida primaria**	**Métricas de medición**	$\mathbf{N}_{\text{futuro}}$
Pre-Encoder (PE)	Entrada (mapas de flujo)	Flujo futuro	EPE, AE	3
Post-Latent (PLS)	Latente (tras codificador)	Flujo futuro	EPE, AE	-
ACCLIP	en la función de pérdida	Imágenes RGB	SSIM, LPIPS, PSRN	20

Tabla 14: Resumen de los 3 modelos de predicción PE, PLS y ACCLIP: *Punto de integración FO* indica en qué lugar de la arquitectura se integra el flujo óptico, *salida primaria* indica el tipo de salida que tiene el modelo, estas pueden ser flujo óptico (para PE y PLS) o imágenes RGB para ACCLIP. N_{futuro} indica los pasos temporales futuros que indica el numero de predicciones que puede generar el modelo.

Para poder garantizar la coherencia en la comparación entre modelos, se fijó un horizonte de evaluación temporal de 4 predicciones; en el caso de ACCLIP, como su salida son imágenes RGB, fue necesario calcular el flujo óptico a partir de dichas predicciones utilizando el mismo método aplicado en PE llamado SelFlow. Los resultados comparativos entre los modelos se presentan en la Tabla 15

Modelo	$\mathbf{N}_{\text{futuro}}$	**EPE** $\downarrow$	**AE (°)** $\downarrow$
Pre-Encoder (PE)	3	3.2	3.5
Post-Latent (PLS)	-	-	-
ACCLIP	3	1.4	1.1

Tabla 15: Resultados comparativos de los tres modelos evaluados. En PE se reportan directamente métricas de flujo (EPE y AE). En el caso de ACCLIP, dado que la salida primaria son imágenes RGB, se calcularon mapas de flujo óptico a partir de sus predicciones usando el método SelFlow, lo que permitió estimar EPE y AE de manera coherente con PE.La columna N_{futuro} indica el horizonte de predicciones evaluado. Cabe destacar que el modelo PLS no logró converger, por lo que no fue posible reportar resultados.

A partir de la tabla es posible identificar que el modelo PE funciona bien en horizontes cortos, con un error promedio de EPE $= 4.6$ px y un error angular de $4.9°$. Esto confirma que usar directamente el flujo como entrada permite capturar patrones de movimiento expresados en flujo

óptico, aunque su desempeño se degrada al intentar predecir más pasos hacia adelante, tal como se evidenció en la sección 9.1.

Por otro lado, el modelo ACCLIP se destacó con claridad. No solo obtuvo mejores métricas de flujo estimado (EPE $= 1.8$ px y AE $= 1.3°$), sino que también, como pudo observarse en la sección 9.3, alcanzó valores altos en las métricas perceptuales de SSIM y LPIPS en las imágenes predichas. Estos resultados indican que integrar el flujo en la función de pérdida ayuda a mantener tanto la coherencia de movimiento como la calidad visual de las predicciones, incluso en horizontes más largos.

Finalmente, el modelo PLS no consiguió converger, lo que sugiere que la integración del flujo en el espacio latente no es una estrategia adecuada para este tipo de tareas.

En resumen, los resultados sugieren que la mejor opción no es usar el flujo como entrada ni en el espacio latente, sino incorporarlo como un elemento regulador en la función de pérdida. Esta estrategia, como lo mostró ACCLIP, ofrece predicciones más precisas y visualmente coherentes en escenarios urbanos complejos.

10. Discusión y líneas futuras

Modelo Pre Encoder (PE)

El análisis realizado en el marco del primer objetivo permitió validar la hipótesis de que el flujo óptico, al incorporarse como entrada directa al modelo, constituye una señal valiosa para capturar la dinámica espacio–temporal en secuencias de video. El uso de SelFlow como estimador externo demostró ser adecuado en contextos urbanos, al manejar correctamente las oclusiones y entregar mapas de flujo coherentes, que sirvieron como entrada para el entrenamiento del autoencoder inicial. A partir de este preentrenamiento, el modelo Pre Encoder (PE) fue capaz de generar predicciones de flujo óptico en horizontes futuros de durante los primeros cinco pasos temporales.

Los experimentos mostraron que la configuración con el codificador congelado ofreció un comportamiento más estable y un mejor equilibrio entre magnitud y dirección del flujo que la alternativa sin congelar. Asimismo, la exploración sistemática de los hiperparámetros de la función de pérdida evidenció que valores de $\alpha = 0.8$, $\beta = 0.2$ y $\gamma = 0.05$ son los hiperparámetros que generan predicciones más confiables, indicando que el modelo se estabiliza cuando se da mas relevancia a la similitud vectorial regida por EPE que a la coherencia direccional regida por el error angular, revelando un compromiso natural entre ambas métricas. En conjunto, estas pruebas permitieron establecer una configuración de pérdida adecuada para el modelo final. Finalmente se pudo estimar que el horizonte temporal de confianza del modelo es de hasta 3 pasos temporales futuros, en los cuales se logra mantener una consistencia estructural comparable al flujo real.

En conclusión, el modelo Pre Encoder (PE) logró demostrar que el flujo óptico puede ser aprovechado como entrada para guiar predicciones en el dominio temporal, confirmando la validez del Objetivo 1. Sin embargo, sus limitaciones al buscar generar predicciones de flujo en horizontes medios y largos evidencian la necesidad de explorar variantes arquitectónicas capaces de superar la dependencia exclusiva del flujo externo y mejorar la capacidad de generalización.

Modelo Post Lantent Space (PLS)

El modelo Post Latent Space (PLS) buscaba evaluar si era posible calcular el flujo óptico directamente en el espacio latente, con la idea de reducir el costo computacional y evitar el uso explícito de mapas de flujo en la entrada. Sin embargo, los resultados mostraron que este enfoque tiene limitaciones claras.

En primer lugar, el análisis de los factores de escala mostró al reducir progresivamente la resolución durante el proceso de codificación genera pérdida de detalle y densidad del flujo óptico. Aunque en las primeras capas es posible reconocer patrones de movimiento globales, a medida que se avanza hacia capas más profundas las representaciones se vuelven demasiado abstractas y

pierden información espacial. Como consecuencia, el flujo estimado en el espacio latente conserva únicamente trazos gruesos de movimiento, lo que no permite identificar desplazamientos de pequeña magnitud o variaciones locales en los bordes de objetos.

Esta pérdida de fidelidad explica por qué el modelo no logró converger de manera estable: al no presentar un campo de flujo óptico confiable en el espacio latente, no se puede sostenter una coherencia temporal que permita generar predicciones con la red recurrente. A diferencia del modelo Pre Encoder, donde el flujo se incorpora explícitamente en la entrada y se refina mediante mecanismos adicionales, en el PLS la estimación latente resulta insuficiente para capturar dependencias espacio temporales.

En resumen, los resultados del PLS muestran que mover el cálculo del flujo al espacio latente puede ahorrar cómputo, pero a costa de perder fidelidad en las predicciones. Esto refuerza la idea de que es necesario encontrar un equilibrio entre eficiencia y calidad, y abre la puerta a explorar variantes que mantengan más detalle espacial, que combinen resoluciones intermedias o que incluyan mecanismos de refinamiento para mejorar la estabilidad del modelo.

Modelo Corrección Adaptativa con Pérdida Combinada Integrando Flujo Óptico para una Mejora en la Predicción Pre Encoder (ACCLIP)

Los resultados de ACCLIP muestran un gran desempeño al compararse con modelos del estado del arte, como SimVP, E3D-LSTM, STMFANet entre otros. En todos los conjuntos de datos (KITTI, KTH-Action y Caltech-Pedestrian), el modelo mantiene la coherencia estructural y temporal de las secuencias de imágenes predichas, obteniendo valores de SSIM mayores en comparación con los modelos presentados en la literatura.

Sin embargo, existen limitaciones claras del modelo, ya que tiende a generar artefactos en zonas con bordes o sombras, perdiendo definición en los contornos. Esto explica por qué, pese a tener buenos resultados en SSIM, el PSNR se mantiene por debajo de otros modelos. Una posible causa podría estar asociada a las limitaciones del método elegido para el cálculo del flujo óptico como parte de la función de pérdida, ya que este método prioriza la coherencia del movimiento por sobre la reconstrucción precisa de cada píxel, lo que se hace notorio al observar los artefactos generados, especialmente en zonas con texturas o en regiones con variaciones de intensidad. Esto resalta la importancia de explorar métodos alternativos de cálculo de flujo óptico, de tal forma que complemente la coherencia del movimiento con la apariencia local, con el objetivo de mejorar la fidelidad pixel a pixel sin sacrificar la coherencia temporal.

En esa línea, también se puede proponer, en investigaciones futuras, explorar mecanismos de atención adaptativa y la incorporación de funciones de pérdida perceptual más sensibles a texturas y contrastes, que ayuden a robustecer el modelo frente a condiciones de iluminación variables.

Ahora bien, desde una perspectiva práctica, ACCLIP resulta especialmente eficiente en contextos donde el objetivo no es la fidelidad exacta de cada píxel, sino la capacidad de anticipar patrones de movimiento. Por ejemplo, en sistemas de seguridad urbana, en la predicción de trayectorias de peatones y vehículos o en escenarios de movilidad autónoma, la preservación de la acción y su continuidad pesa más que la nitidez de las texturas.

En síntesis, pese a las limitaciones encontradas, ACCLIP logra conservar la forma y el movimiento general en imágenes de secuencias urbanas, algo que en estudios previos se ha señalado como fundamental para la predicción de secuencias. Esta tendencia también encaja con las ideas de la codificación predictiva, donde se prioriza la geometría y el movimiento global como claves para anticipar la dinámica de la escena.

En conclusión, ACCLIP no solo demuestra un rendimiento competitivo frente al estado del arte, sino que además abre la puerta a una reflexión más amplia: los modelos de predicción de secuencias deben encontrar un mejor balance entre coherencia estructural y fidelidad visual. Este equilibrio es clave no solo para la investigación en predicción basada en flujo, sino también para aplicaciones concretas de la visión por computador en entornos reales.

Desde una perspectiva biológica, el principio de codificación predictiva plantea que el sistema ajusta sus predicciones comparando la información esperada con la percibida, generando señales de error de predicción que permiten ajustar al sistema para generar predicciones cada vez más certeras. Los modelos desarrollados en esta tesis aplican este principio de forma análoga mediante un mecanismo de decisión donde se genera un cálculo de error que compara la discrepancia entre la predicción y la entrada real actuando como un mecanismo de ajuste que optimiza los parámetros del sistema para generar predicciones confiables en un horizonte temporal más amplio. Este proceso permite que las redes aprendan de la experiencia, refinando progresivamente sus predicciones y reproduciendo, en términos computacionales, el modo en que los sistemas biológicos mejoran su capacidad anticipatoria a partir del error.

De igual manera, el proceso de aprendizaje en los modelos propuestos se basa en la retroalimentación iterativa donde la función de pérdida actúa como un mecanismo de ajuste que permite al sistema aprender en base a su experiencia. En el modelo PE, esta función pondera la coherencia entre magnitud y dirección del flujo óptico predicho, permitiendo ajustar la magnitud del desplazamiento y también la coherencia direccional del movimiento. El modelo ACCLIP, por otra parte, integra la función de pérdida en términos de reconstrucción visual (SSIM, PSNR) y de flujo óptico, permitiendo que el sistema aprenda simultáneamente la apariencia y la dinámica de la escena. Estos mecanismos de aprendizaje reproducen el principio biológico de la codificación predictiva, donde las predicciones se ajustan continuamente en función de la experiencia con el entorno. De esta manera, los modelos propuestos en términos computacionales, reflejan la capacidad adaptativa del sistema para refinar sus predicciones a partir de la interacción con el entorno.

Discusión general

Como se resume en la Tabla 16, de los tres enfoques , solo dos pueden representar predicciones del flujo óptico, aunque reflejan distintos compromisos entre confianzaen la predicción y alcance predictivo.

Modelo	Aspecto clave	Puntos fuertes	Limitaciones	Predictivo
Pre Encoder	Incluye el flujo óptico antes del encoder.	Estabilidad en horizontes cortos	Alcance limitado, 3 predicciones	✓
Post Latent Space	Infiere el flujo en el espacio latente	✗	Pérdida de detalle fino a gran escala	✗
ACCLIP	Incluye el flujo óptico en la función de perdida.	Robustez en la predicción a largo plazo	Mayor complejidad computacional	✓

Tabla 16: Comparativa de resultados de los modelos evaluados

Estos resultados sugieren que existen dos ubicaciones claves para integrar el flujo óptico dentro de la arquitectura: antes del codificador como se presenta en el modelo PE y como componente regulador en la función de pérdida como se presenta en el modelo ACCLIP.

La hipótesis central de esta tesis sostiene que una predicción robusta de trayectorias en un observador móvil requiere codificar información de flujo óptico para representar de manera efectiva la estructura espaciotemporal de la escena. Los resultados obtenidos respaldan esta afirmación: tanto el modelo PE (EPE $\approx 1{,}23$ px, AE $\approx 1{,}14°$) como la arquitectura completa ACCLIP (SSIM $\approx 0{,}98$, PSNR $\approx 32{,}4$ dB) validan que la integración del flujo óptico, tanto en el diseño del modelo como en la función de pérdida, es esencial para alcanzar predicciones estables, precisas y coherentes en entornos dinámicos.

11. Conclusión

Modelo Pre Encoder (PE)

El modelo Pre Encoder (PE) demostró que el flujo óptico puede emplearse como entrada para generar predicciones consistentes en horizontes cortos, confirmando la hipótesis del Objetivo 1.

Los experimentos en el conjunto de datos KITTI mostraron que la configuración con codificador congelado además de una ponderación alta de α y baja de β en la función de pérdida ofrecieron el mejor equilibrio entre precisión en magnitud y coherencia en dirección.

Asimismo, se mantuvo la estabilidad de las métricas de error durante los primeros tres pasos de predicción, con un *End Point Error (EPE)* promedio cercano a 3.5 px y un *Angular Error (AE)* de $\sim 3.8°$. La inspección cualitativa en este horizonte temporal corroboró que las estructuras globales de movimiento se reproducen con fidelidad en este rango y que los errores se concentran principalmente en los bordes de los objetos.

Estos hallazgos validan la pertinencia del enfoque, aunque ponen de manifiesto limitaciones en horizontes largos que motivan la exploración de arquitecturas alternativas.

Modelo Post Laten Space (PLS)

El modelo Post Latent Space (PLS) no logró generar predicciones confiables debido a la pérdida de detalle al calcular el flujo óptico en el espacio latente, lo que impidió su convergencia estable. Si bien la estrategia reduce el costo computacional, compromete la fidelidad del flujo y limita la capacidad predictiva, estableciendo un horizonte poco práctico para su aplicación. Estos resultados evidencian que el cálculo de flujo en el espacio latente no es suficiente por sí solo y refuerzan la necesidad de explorar arquitecturas alternativas que preserven mayor información espacial.

Modelo Corrección Adaptativa con Pérdida Combinada Integrando Flujo Óptico para una Mejora en la Predicción Pre Encoder (ACCLIP)

Los resultados obtenidos con ACCLIP muestran un avance claro frente a varios de los modelos actuales. En los tres conjuntos de datos evaluados: KTH-Action, Caltech-Pedestrian y KITTI, el modelo alcanzó los valores muy superiores en SSIM: en KITTI tuvo un resultado de 0.8824 para 14 predicciones, en KTH-Action un resultado de 0.967 para 20 predicciones y 0.974 para 40 predicciones y en Caltech-Pedestrian un valor de 0.983 para 1 predicción. Confirmando su capacidad para mantener una continuidad de las escenas en el tiempo. En cambio, al evaluar PSNR, el modelo tuvo un resultado inferior en relación a algunos métodos, lo que refleja que,

aunque la estructura general de la escena se conserva, los detalles como bordes y texturas no siempre se reconstruyen con precisión. En otras palabras, ACCLIP privilegia la coherencia del movimiento por sobre la fidelidad de cada píxel.

Este comportamiento conecta con la idea de la codificación predictiva, donde lo fundamental es anticipar la dinámica de la escena a partir de la geometría y el movimiento, más que de la apariencia exacta de cada punto. Así, ACCLIP no solo entrega métricas competitivas, sino que también aporta una forma distinta de pensar la función de pérdida para generar modelos de predicción de secuencias, poniendo en el centro en el movimiento del objeto mas que en el detalle perceptual. Este nuevo paradigma de incorporar el flujo óptico en la función de pérdida abre un camino interesante tanto para la investigación teórica como para aplicaciones reales en visión por computador.

Como conclusión final, en primer lugar, es posible determinar que los tres modelos desarrollados en esta investigación permiten identificar las implicancias sobre el papel del flujo óptico en la predicción de secuencias de video. El modelo Pre-Encoder validó que incorporar el flujo como entrada mejora la consistencia estructural en horizontes cortos, aunque evidenció limitaciones al extender la predicción en el tiempo. El enfoque Post Latent Space, pese a su promesa de eficiencia, mostró que calcular el flujo únicamente en el espacio latente compromete la fidelidad y no resulta suficiente para sostener coherencia temporal. Finalmente, el modelo ACCLIP se consolidó como el mejor método, alcanzando métricas de similitud estructural (SSIM) superiores al estado del arte en todos los conjuntos de datos evaluados y confirmando que la preservación de la coherencia espacial y del movimiento es clave incluso cuando se sacrifica detalle fino en la reconstrucción pixel a pixel.

En segundo lugar, esta investigación demuestra que el flujo óptico no solo es un recurso útil como entrada o como restricción en la función de pérdida, sino que además puede guiar diseños arquitectónicos que equilibren estructura general y fidelidad local en modelos de predicciones.

En tercer lugar, los hallazgos presentados sugieren que modelos como ACCLIP tienen un alto potencial en aplicaciones donde lo importante es anticipar patrones de movimiento, como en seguridad urbana, predicción de trayectorias o sistemas de movilidad autónoma. No obstante, persisten desafíos en la representación de texturas y bordes, lo que abre líneas claras para investigaciones futuras orientadas a integrar mecanismos de atención adaptativa y funciones perceptuales en la función de pérdida que podrían permitir mejorar el desempeño en condiciones reales.

12. Bibliografía

Referencias

[1] Jerome A Feldman. On intelligence as memory. *Artificial Intelligence*, 169(2):181–183, 2005.

[2] Andy Clark. Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204, 2013.

[3] Harriet Feldman and Karl J Friston. Attention, uncertainty, and free-energy. *Frontiers in human neuroscience*, 4:215, 2010.

[4] Jakob Hohwy. Attention and conscious perception in the hypothesis testing brain. *Frontiers in psychology*, 3:96, 2012.

[5] Manuel S Malmierca, Lucy A Anderson, and Flora M Antunes. The cortical modulation of stimulus-specific adaptation in the auditory midbrain and thalamus: a potential neuronal correlate for predictive coding. *Frontiers in systems neuroscience*, 9:19, 2015.

[6] Peter F Liddle and Musa B Sami. The mechanisms of persisting disability in schizophrenia: imprecise predictive coding via cortico-striatal-thalamo-cortical loop dysfunction. *Biological Psychiatry*, 2024.

[7] William H Alexander and Joshua W Brown. Frontal cortex function as derived from hierarchical predictive coding. *Scientific reports*, 8(1):3843, 2018.

[8] Andre M Bastos, W Martin Usrey, Rick A Adams, George R Mangun, Pascal Fries, and Karl J Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

[9] Wolfram Schultz, Peter Dayan, and P Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

[10] Karl Friston. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836, 2005.

[11] Gábor Stefanics, Jan Kremláček, and István Czigler. Visual mismatch negativity: a predictive coding view. *Frontiers in human neuroscience*, 8:666, 2014.

[12] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[13] H. B. Barlow and R. M. Hill. Selective sensitivity to direction of movement in ganglion cells of the rabbit retina. *Journal of Physiology*, 173(3):377–407, 1963.

[14] Jason Johnston, Huayu Ding, Sara-Helen Seibel, Federico Esposti, and Leon Lagnado. A retinal circuit generating a dynamic predictive code for motion. *Current Biology*, 29(17):2893–2906.e8, 2019.

[15] Richard H. Masland. The neuronal organization of the retina. *Neuron*, 76(2):266–280, 2012.

[16] Taro Hosoya, Stephen A. Baccus, and Markus Meister. Dynamic predictive coding by the retina. *Nature*, 436(7047):71–77, 2005.

[17] Bing Liu, Matthew V. Macellaio, and Luke C. Osborne. Predictive encoding of motion begins in the primate retina. *Nature Communications*, 12:3375, 2021.

[18] Michael W Spratling. Reconciling predictive coding and biased competition models of cortical function. *Frontiers in computational neuroscience*, 2:300, 2008.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Viren Jain, Joseph F Murray, Fabian Roth, Srinivas Turaga, Valentin Zhigulin, Kevin L Briggman, Moritz N Helmstaedter, Winfried Denk, and H Sebastian Seung. Supervised learning of image restoration with convolutional networks. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[21] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[22] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International conference on learning representations*, 2018.

[23] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29, 2016.

[24] Sandra Aigner and Marco Körner. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans. *arXiv preprint arXiv:1810.01325*, 2018.

[25] Joost Van Amersfoort, Anitha Kannan, Marc'Aurelio Ranzato, Arthur Szlam, Du Tran, and Soumith Chintala. Transformation-based models of video sequences. *arXiv preprint arXiv:1701.08435*, 2017.

[26] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[27] Ziru Xu, Yunbo Wang, Mingsheng Long, Jianmin Wang, and M KLiss. Predcnn: Predictive learning with cascade convolutions. In *IJCAI*, pages 2940–2947, 2018.

[28] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3170–3180, 2022.

[29] Cheng Tan, Zhangyang Gao, Siyuan Li, and Stan Z Li. Simvpv2: Towards simple yet powerful spatiotemporal predictive learning. *IEEE Transactions on Multimedia*, 2025.

[30] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[31] Wei Fang, Yupeng Chen, and Qiongying Xue. Survey on research of rnn-based spatio-temporal sequence prediction algorithms. *Journal on Big Data*, 3(3):97, 2021.

[32] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[33] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008.

[34] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[35] Jingxia X Chen, DM Jiang, and YN Zhang. A hierarchical bidirectional gru model with attention for eeg-based emotion classification. *Ieee Access*, 7:118530–118540, 2019.

[36] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *arXiv preprint arXiv:1605.08104*, 2016.

[37] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017.

[38] Angel Villar-Corrales, Ani Karapetyan, Andreas Boltres, and Sven Behnke. Mspred: Video prediction at multiple spatio-temporal scales with hierarchical recurrent networks. *arXiv preprint arXiv:2203.09303*, 2022.

[39] Cheng Tan, Siyuan Li, Zhangyang Gao, Wenfei Guan, Zedong Wang, Zicheng Liu, Lirong Wu, and Stan Z Li. Openstl: A comprehensive benchmark of spatio-temporal predictive learning. *Advances in Neural Information Processing Systems*, 36:69819–69831, 2023.

[40] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of CVPR*, pages 8934–8943, 2018.

[41] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020.

[42] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE international conference on computer vision*, pages 4463–4471, 2017.

[43] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5539–5548, 2020.

[44] Yue Wu, Qiang Wen, and Qifeng Chen. Optimizing video prediction via video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17814–17823, 2022.

[45] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*, pages 624–642. Springer, 2022.

[46] David I. Vaney, Sheng He, and W. Rowland Taylor. Direction-selective ganglion cells in the retina: Symmetry and asymmetry in structure and function. *Nature Reviews Neuroscience*, 13(3):194–208, 2012.

[47] Shai Sabbah, Timothy C. Gemmer, Brian B. Bhatia, Shai D. Hillier, and Markus Meister. A retinal code for motion along the gravitational and body axes during locomotion. *Nature Neuroscience*, 20:196–204, 2017.

[48] Jonathan Samir Matthis, Jonathan L. Yates, and Mary M. Hayhoe. Retinal optic flow during natural locomotion. *Current Biology*, 32(6):1238–1249.e4, 2022.

[49] John C. Rasmussen, Joseph M. Barrett, and John H. Singer. Contributions of retinal direction selectivity to central visual motion processing. *Current Biology*, 30(20):R1180–R1192, 2020.

[50] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[51] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of IJCAI*, pages 674–679, 1981.

[52] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the Scandinavian Conference on Image Analysis (SCIA)*, volume 2749 of *Lecture Notes in Computer Science*, pages 363–370. Springer, 2003.

[53] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Can Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *Proceedings of ICCV*, pages 2758–2766, 2015.

[54] Eddy Ilg, Nikolaus Mayer, Tsung-Yi Saikia, Marcin Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of CVPR*, pages 1647–1655, 2017.

[55] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4571–4580, 2019.

[56] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.

[57] Andrea Ciamarra, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Forecasting future instance segmentation with learned optical flow and warping. In *International Conference on Image Analysis and Processing*, pages 349–361. Springer, 2022.

[58] Qiaole Dong and Yanwei Fu. Memflow: Optical flow estimation and prediction with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19068–19078, 2024.

[59] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *European conference on computer vision*, pages 668–685. Springer, 2022.

[60] Deqing Sun, Charles Herrmann, Fitsum Reda, Michael Rubinstein, David J Fleet, and William T Freeman. Disentangling architecture and training for optical flow. In *European Conference on Computer Vision*, pages 165–182. Springer, 2022.

[61] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5437–5446, 2020.

[62] Jingyun Liang, Yuchen Fan, Kai Zhang, Radu Timofte, Luc Van Gool, and Rakesh Ranjan. Movideo: Motion-aware video generation with diffusion model. In *European Conference on Computer Vision*, pages 56–74. Springer, 2024.

[63] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, 2 edition, 2001. Capítulo 5.

[64] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. Sección 1.4.

[65] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Sección 6.2.1.

[66] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

[67] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016.

[68] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[69] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11-14, 2016, proceedings, part v 14*, pages 597–613. Springer, 2016.

[70] Ruikun Wang et al. Research on image generation and style transfer algorithm based on deep learning. *Open Journal of Applied Sciences*, 9(08):661, 2019.

[71] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

[72] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[73] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015.

[74] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 1019–1027, 2016.

[75] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[76] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

[77] Dilbag Singh, Deepak Garg, and Husanbir Singh Pannu. Efficient landsat image fusion using fuzzy and stationary discrete wavelet transform. *The Imaging Science Journal*, 65(2):108–114, 2017.

[78] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[79] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

[80] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004.

[81] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *2009 IEEE conference on computer vision and pattern recognition*, pages 304–311. IEEE, 2009.

[82] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[83] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018.

[84] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[85] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[86] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning*, pages 1771–1779. PMLR, 2017.

[87] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.

[88] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3163–3172, 2021.

[89] Jianjin Zhang, Yunbo Wang, Mingsheng Long, Wang Jianmin, and S Yu Philip. Z-order recurrent neural networks for video prediction. In *2019 IEEE International conference on multimedia and expo (ICME)*, pages 230–235. IEEE, 2019.

[90] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[91] Beibei Jin, Yu Hu, Yiming Zeng, Qiankun Tang, Shice Liu, and Jing Ye. Varnet: Exploring variations for unsupervised video prediction. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5801–5806. IEEE, 2018.

[92] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International conference on machine learning*, pages 5123–5132. PMLR, 2018.

[93] Jungbeom Lee, Jangho Lee, Sungmin Lee, and Sungroh Yoon. Mutual suppression network for video prediction using disentangled features. *arXiv preprint arXiv:1804.04810*, 2018.

[94] Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4554–4563, 2020.

[95] Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7854–7863, 2018.

[96] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[97] Fitsum A Reda, Guilin Liu, Kevin J Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro. Sdc-net: Video prediction using spatially-displaced convolution. In *Proceedings of the European conference on computer vision (ECCV)*, pages 718–733, 2018.

[98] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019.

[99] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 753–769, 2018.

[100] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9006–9015, 2019.

[101] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations*, 2020.

A. Anexo

A.1. Barrido de hiperparámetros modelo Pre Encoder

A continuación se presenta barrido completo de de hiperparámetros α, β y γ de la función de pérdida (véase Sección 8.2.2), con todas las combinaciones probadas y sus métricas (EPE, AE). Las tablas están organizadas acorde a la variación del hiperparámetro α y sus combinaciones con β y γ. Se reportan promedios de validación cuando el modelo se entrena con el codificador congelado y sin congelar

Tabla A.1: Comparación de resultados para $\alpha = 0.1$

Parámetros			No Congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.10	0.10	0.01	1.57	14.29	2.09	11.14
0.10	0.10	0.05	1.81	13.68	1.39	12.07
0.10	0.10	0.10	1.40	13.83	4.93	12.69
0.10	0.20	0.01	1.65	12.42	2.65	15.03
0.10	0.20	0.05	2.21	13.82	1.88	10.14
0.10	0.20	0.10	4.44	19.58	1.61	9.66
0.10	0.30	0.01	1.82	16.00	2.48	9.69
0.10	0.30	0.05	1.67	15.02	2.44	12.79
0.10	0.30	0.10	2.40	15.87	2.17	10.01
0.10	0.40	0.01	2.02	17.02	2.78	16.12
0.10	0.40	0.05	2.57	16.13	3.42	14.25
0.10	0.40	0.10	3.24	15.27	2.26	14.01
0.10	0.50	0.01	1.86	14.49	1.89	13.91
0.10	0.50	0.05	1.84	13.32	2.08	16.53
0.10	0.50	0.10	1.83	14.28	2.26	14.02
0.10	0.60	0.01	2.42	14.02	1.18	16.29
0.10	0.60	0.05	4.17	15.12	6.44	15.79
0.10	0.60	0.10	2.40	15.87	2.17	15.24
0.10	0.70	0.01	1.72	14.29	2.14	12.97
0.10	0.70	0.05	2.05	14.55	2.22	11.01
0.10	0.70	0.10	2.17	15.66	2.03	14.56
0.10	0.80	0.01	4.12	14.97	3.48	14.49
0.10	0.80	0.05	5.27	14.29	1.35	14.74
0.10	0.80	0.10	3.01	16.24	1.27	13.31
0.10	0.90	0.01	1.74	14.66	1.93	15.42
0.10	0.90	0.05	1.84	16.43	2.05	16.02
0.10	0.90	0.10	1.77	13.81	2.01	10.31

Tabla A.2: Comparación de resultados para $\alpha = 0.2$

Parámetros			No Congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.20	0.10	0.01	2.54	14.62	2.30	10.61
0.20	0.10	0.05	2.81	14.68	2.34	14.07
0.20	0.10	0.10	1.54	18.84	4.01	13.69
0.20	0.20	0.01	1.74	15.98	3.42	12.03
0.20	0.20	0.05	1.34	14.01	2.89	14.14
0.20	0.20	0.10	2.48	16.13	3.12	13.66
0.20	0.30	0.01	2.45	16.01	2.21	16.69
0.20	0.30	0.05	1.42	15.52	2.01	15.79
0.20	0.30	0.10	2.47	14.16	3.47	15.01
0.20	0.40	0.01	2.61	16.01	3.64	16.12
0.20	0.40	0.05	2.84	17.13	3.13	14.25
0.20	0.40	0.10	2.61	15.87	3.26	14.01
0.20	0.50	0.01	1.12	15.29	1.89	13.91
0.20	0.50	0.05	1.61	14.31	1.78	16.53
0.20	0.50	0.10	1.62	14.46	1.27	14.62
0.20	0.60	0.01	2.46	14.13	5.08	16.29
0.20	0.60	0.05	3.17	13.92	6.64	13.79
0.20	0.60	0.10	3.14	15.27	5.87	15.24
0.20	0.70	0.01	2.13	15.94	4.13	14.24
0.20	0.70	0.05	2.21	14.01	5.67	16.01
0.20	0.70	0.10	2.34	14.97	4.96	14.56
0.20	0.80	0.01	2.61	13.01	3.84	14.49
0.20	0.80	0.05	3.74	14.29	3.34	14.74
0.20	0.80	0.10	3.94	14.21	6.51	13.31
0.20	0.90	0.01	3.46	15.97	3.84	13.42
0.20	0.90	0.05	2.84	14.02	4.05	16.02
0.20	0.90	0.10	2.47	14.35	4.94	15.31

Tabla A.3: Comparación de resultados con $\alpha = 0.3$

Parámetros			No congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.30	0.10	0.01	1.50	5.35	1.65	4.39
0.30	0.10	0.05	4.25	19.52	2.08	7.12
0.30	0.10	0.10	1.35	13.59	2.69	10.36
0.30	0.20	0.01	2.01	8.63	1.97	9.55
0.30	0.20	0.05	1.73	15.84	1.30	9.02
0.30	0.20	0.10	1.81	9.02	1.44	13.55
0.30	0.30	0.01	2.42	7.65	1.82	8.72
0.30	0.30	0.05	1.87	13.29	2.30	9.10
0.30	0.30	0.10	1.54	10.34	1.33	12.19
0.30	0.40	0.01	1.87	18.62	1.42	14.08
0.30	0.40	0.05	3.87	13.11	2.34	9.22
0.30	0.40	0.10	1.58	15.64	2.45	12.98
0.30	0.50	0.01	1.87	10.75	2.49	8.18
0.30	0.50	0.05	3.87	14.21	1.74	11.42
0.30	0.50	0.10	1.58	13.94	1.45	12.54
0.30	0.60	0.01	4.54	8.75	2.49	8.08
0.30	0.60	0.05	3.21	14.21	1.74	9.22
0.30	0.60	0.10	1.85	15.54	2.25	9.98
0.30	0.70	0.01	1.57	14.21	2.33	13.03
0.30	0.70	0.05	1.70	13.73	1.54	9.91
0.30	0.70	0.10	1.65	10.12	1.56	14.12
0.30	0.80	0.01	2.64	13.74	2.12	9.08
0.30	0.80	0.05	3.84	19.23	1.64	15.01
0.30	0.80	0.10	2.26	13.94	1.45	19.84
0.30	0.90	0.01	1.69	13.81	1.68	13.27
0.30	0.90	0.05	1.63	13.54	1.71	9.51
0.30	0.90	0.10	1.61	10.42	1.59	13.62

Tabla A.4: Comparación de resultados con $\alpha = 0.4$

Parámetros			No congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.40	0.10	0.01	2.40	9.95	2.63	7.34
0.40	0.10	0.05	3.25	9.52	2.94	8.32
0.40	0.10	0.10	3.24	12.06	2.34	9.31
0.40	0.20	0.01	4.41	11.64	3.97	9.23
0.40	0.20	0.05	3.61	9.34	2.93	8.64
0.40	0.20	0.10	2.81	9.34	2.34	11.31
0.40	0.30	0.01	2.13	11.87	2.82	8.95
0.40	0.30	0.05	1.87	10.62	1.24	7.34
0.40	0.30	0.10	3.52	11.34	3.33	8.94
0.40	0.40	0.01	2.87	12.64	2.24	12.31
0.40	0.40	0.05	1.97	11.11	1.93	10.12
0.40	0.40	0.10	2.64	12.64	3.32	12.98
0.40	0.50	0.01	2.84	10.75	3.30	8.18
0.40	0.50	0.05	2.66	14.21	2.07	11.42
0.40	0.50	0.10	3.58	13.94	2.45	12.54
0.40	0.60	0.01	4.24	8.75	3.64	9.87
0.40	0.60	0.05	3.30	9.91	2.76	10.22
0.40	0.60	0.10	2.95	12.01	3.25	9.97
0.40	0.70	0.01	2.62	11.32	3.02	10.03
0.40	0.70	0.05	1.98	13.01	2.54	10.21
0.40	0.70	0.10	4.63	12.13	2.36	11.31
0.40	0.80	0.01	3.42	8.74	3.21	9.08
0.40	0.80	0.05	2.93	9.23	2.32	10.81
0.40	0.80	0.10	2.32	9.94	2.45	11.34
0.40	0.90	0.01	2.74	9.81	1.67	10.27
0.40	0.90	0.05	1.92	7.54	1.97	9.64
0.40	0.90	0.10	1.93	10.22	2.53	11.34

Tabla A.5: Comparación de resultados con $\alpha = 0.5$

Parámetros			No congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.50	0.10	0.01	1.56	6.51	1.35	7.65
0.50	0.10	0.05	1.28	6.07	1.24	4.06
0.50	0.10	0.10	2.47	6.14	1.55	6.14
0.50	0.20	0.01	1.71	7.42	1.29	5.72
0.50	0.20	0.05	1.48	8.92	2.54	5.01
0.50	0.20	0.10	1.66	5.83	1.46	6.17
0.50	0.30	0.01	1.30	6.77	4.52	5.75
0.50	0.30	0.05	1.46	6.32	1.36	6.84
0.50	0.30	0.10	1.56	8.08	1.75	7.86
0.50	0.40	0.01	2.34	7.84	2.42	7.98
0.50	0.40	0.05	1.62	6.64	2.34	5.82
0.50	0.40	0.10	1.34	9.84	1.34	9.31
0.50	0.50	0.01	1.62	7.69	6.76	7.55
0.50	0.50	0.05	1.54	6.42	1.63	5.36
0.50	0.50	0.10	1.68	8.07	2.43	7.67
0.50	0.60	0.01	2.34	9.77	2.52	8.36
0.50	0.60	0.05	1.97	6.32	1.34	4.34
0.50	0.60	0.10	2.34	7.08	1.44	6.86
0.50	0.70	0.01	2.72	5.63	2.68	4.71
0.50	0.70	0.05	1.61	5.05	1.80	4.53
0.50	0.70	0.10	1.57	3.99	1.47	3.89
0.50	0.80	0.01	1.30	6.77	4.52	5.75
0.50	0.80	0.05	2.34	5.34	2.34	4.64
0.50	0.80	0.10	1.32	7.97	1.37	6.61
0.50	0.90	0.01	2.47	9.34	1.92	6.68
0.50	0.90	0.05	1.51	6.60	1.67	5.27
0.50	0.90	0.10	1.53	8.76	1.45	7.44

Tabla A.6: Comparación de resultados con $\alpha = 0.6$

Parámetros			No congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.60	0.10	0.01	2.24	6.35	1.97	6.12
0.60	0.10	0.05	1.67	5.24	1.34	4.24
0.60	0.10	0.10	2.37	4.34	2.03	4.36
0.60	0.20	0.01	1.97	4.61	1.29	3.84
0.60	0.20	0.05	1.61	3.92	1.51	2.37
0.60	0.20	0.10	1.67	5.68	2.23	4.67
0.60	0.30	0.01	2.31	4.87	2.35	4.32
0.60	0.30	0.05	1.36	5.36	1.97	3.34
0.60	0.30	0.10	1.98	5.68	1.65	3.48
0.60	0.40	0.01	2.32	4.84	2.34	4.38
0.60	0.40	0.05	1.35	3.91	1.34	3.71
0.60	0.40	0.10	1.61	3.97	1.95	3.67
0.60	0.50	0.01	1.37	4.67	1.36	4.57
0.60	0.50	0.05	1.54	2.62	1.37	3.94
0.60	0.50	0.10	1.36	5.37	2.43	3.67
0.50	0.60	0.01	2.32	4.68	2.52	3.41
0.60	0.60	0.05	1.68	3.99	1.24	3.95
0.60	0.60	0.10	2.32	2.38	2.44	2.31
0.60	0.70	0.01	1.79	3.57	2.68	3.21
0.60	0.70	0.05	1.62	2.89	1.80	2.35
0.60	0.70	0.10	1.34	2.94	1.47	1.87
0.60	0.80	0.01	2.68	3.47	2.52	2.32
0.60	0.80	0.05	1.39	2.29	1.34	3.36
0.60	0.80	0.10	1.48	4.02	1.34	4.24
0.60	0.90	0.01	2.03	3.97	1.62	4.05
0.60	0.90	0.05	1.67	4.21	1.27	3.67
0.60	0.90	0.10	1.94	4.79	1.46	3.31

Tabla A.7: Comparación de resultados con $\alpha = 0.7$

Parámetros			No Congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.70	0.10	0.01	1.70	8.66	1.77	4.63
0.70	0.10	0.05	1.63	4.93	1.38	4.01
0.70	0.10	0.10	1.51	4.24	1.26	3.75
0.70	0.20	0.01	3.05	8.97	1.37	4.53
0.70	0.20	0.05	3.84	3.50	1.42	3.09
0.70	0.20	0.10	1.96	3.79	1.37	6.35
0.70	0.30	0.01	2.50	5.21	1.79	4.10
0.70	0.30	0.05	1.57	3.63	1.51	3.02
0.70	0.30	0.10	2.64	5.90	1.85	4.52
0.70	0.40	0.01	2.34	5.34	1.79	4.10
0.70	0.40	0.05	1.61	3.46	1.34	3.32
0.70	0.40	0.10	2.31	5.90	1.86	4.52
0.70	0.50	0.01	2.40	6.13	1.58	4.92
0.70	0.50	0.05	2.31	6.45	1.79	3.16
0.70	0.50	0.10	2.15	4.55	1.53	4.08
0.70	0.60	0.01	1.97	5.81	1.95	4.32
0.70	0.60	0.05	1.24	3.63	1.34	3.02
0.70	0.60	0.10	2.35	5.90	2.34	4.52
0.70	0.70	0.01	2.72	7.14	3.91	4.32
0.70	0.70	0.05	1.59	3.48	5.07	5.16
0.70	0.70	0.10	1.72	3.91	2.32	5.14
0.70	0.80	0.01	2.50	5.68	1.35	3.10
0.70	0.80	0.05	1.57	3.63	1.21	3.02
0.70	0.80	0.10	1.84	5.90	1.84	4.52
0.70	0.90	0.01	1.68	3.83	1.73	4.59
0.70	0.90	0.05	1.34	3.64	1.63	3.26
0.70	0.90	0.10	1.60	5.15	1.80	3.28

Tabla A.8: Comparación de resultados con $\alpha = 0.8$

Parámetros			No congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.80	0.10	0.01	1.68	4.71	1.21	3.60
0.80	0.10	0.05	1.60	3.70	1.20	3.23
0.80	0.10	0.10	2.12	4.83	1.47	3.44
0.80	0.20	0.01	1.73	3.86	1.25	2.22
0.80	0.20	0.05	2.42	2.87	1.20	1.90
0.80	0.20	0.10	2.75	3.23	3.51	2.76
0.80	0.30	0.01	1.51	3.43	1.56	2.01
0.80	0.30	0.05	2.53	2.04	2.09	1.91
0.80	0.30	0.10	2.58	2.30	2.72	1.91
0.80	0.40	0.01	1.54	3.51	1.23	3.13
0.80	0.40	0.05	2.35	2.32	1.33	1.97
0.80	0.40	0.10	1.36	3.45	1.34	2.21
0.80	0.50	0.01	2.90	5.88	1.43	2.67
0.80	0.50	0.05	2.56	2.23	1.32	1.92
0.80	0.50	0.10	1.60	3.95	1.54	2.76
0.80	0.60	0.01	1.32	4.32	1.68	3.01
0.80	0.60	0.05	2.64	2.38	2.42	1.96
0.80	0.60	0.10	1.84	3.74	2.34	2.32
0.80	0.70	0.01	1.64	3.75	1.17	3.52
0.80	0.70	0.05	1.50	3.40	1.19	2.63
0.80	0.70	0.10	1.62	3.66	1.17	3.27
0.80	0.80	0.01	2.32	2.34	1.26	2.38
0.80	0.80	0.05	1.21	2.61	2.09	2.13
0.80	0.80	0.10	1.38	3.31	2.72	2.35
0.80	0.90	0.01	1.53	3.49	1.65	3.42
0.80	0.90	0.05	1.47	3.28	1.51	3.38
0.80	0.90	0.10	1.49	3.30	1.53	3.55

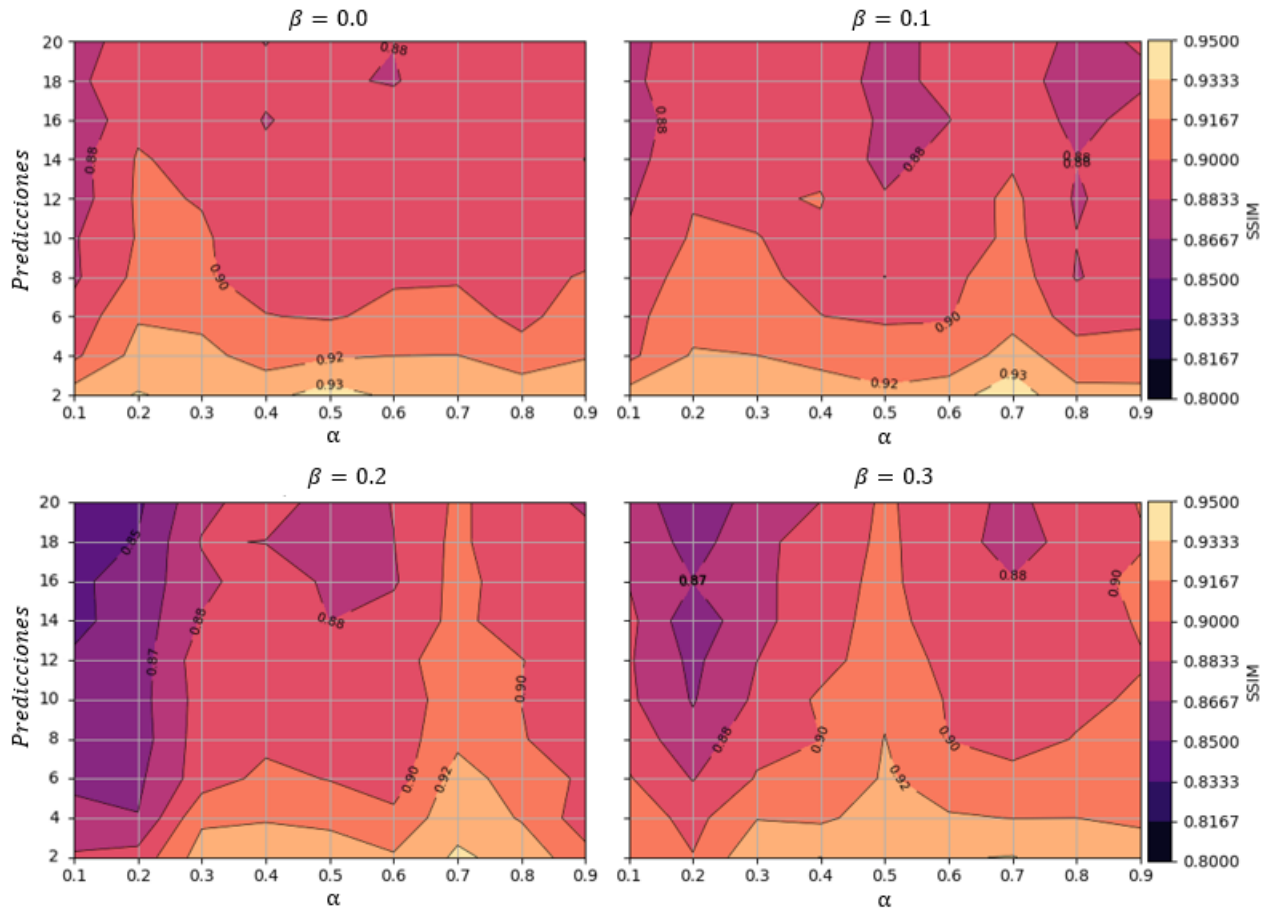
Tabla A.9: Comparación de resultados con $\alpha = 0.9$

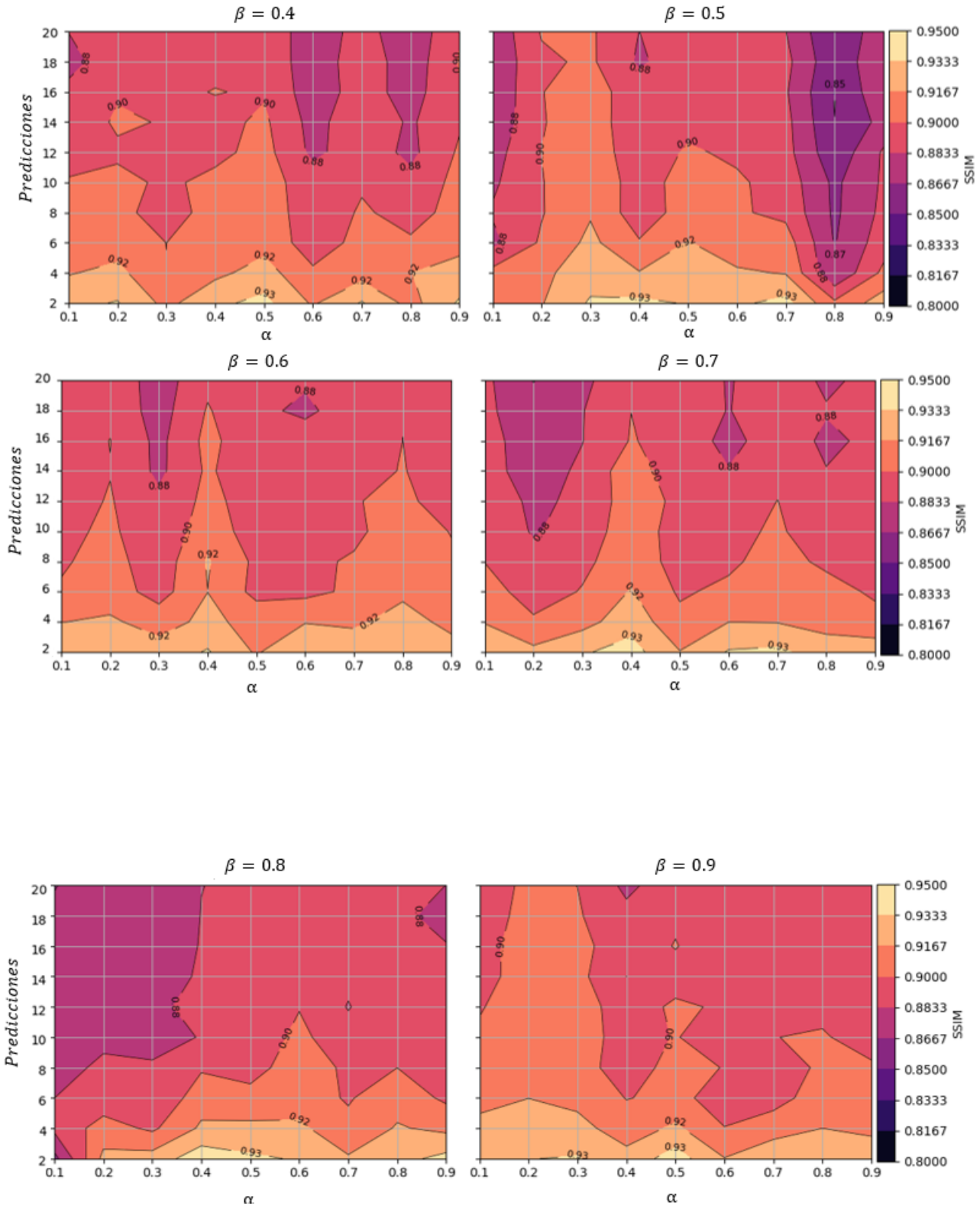
Parámetros			No Congelado		Congelado	
α	β	γ	EPE	AE	EPE	AE
0.90	0.10	0.01	2.57	5.51	2.33	5.43
0.90	0.10	0.05	2.86	7.62	2.41	4.69
0.90	0.10	0.10	1.40	3.47	2.26	3.97
0.90	0.20	0.01	1.42	2.72	2.34	1.93
0.90	0.20	0.05	1.81	3.76	1.89	2.13
0.90	0.20	0.10	3.25	5.69	2.43	3.36
0.90	0.30	0.01	2.42	5.11	1.39	4.33
0.90	0.30	0.05	1.75	5.10	1.82	4.47
0.90	0.30	0.10	2.03	7.18	1.93	3.57
0.90	0.40	0.01	1.32	5.21	1.31	4.21
0.90	0.40	0.05	1.42	4.01	1.12	3.35
0.90	0.40	0.10	1.38	4.24	1.36	2.14
0.90	0.50	0.01	1.92	5.98	1.39	2.77
0.90	0.50	0.05	1.62	4.89	1.80	2.49
0.90	0.50	0.10	2.08	5.56	1.95	2.66
0.90	0.60	0.01	1.14	3.01	1.11	2.32
0.90	0.60	0.05	2.34	2.12	1.12	1.97
0.90	0.60	0.10	2.74	2.38	1.47	1.96
0.90	0.70	0.01	1.55	4.19	1.67	3.02
0.90	0.70	0.05	1.63	6.25	1.81	4.48
0.90	0.70	0.10	1.59	3.91	1.77	2.12
0.90	0.80	0.01	1.75	4.01	1.48	2.33
0.90	0.80	0.05	2.45	2.34	1.22	1.95
0.90	0.80	0.10	1.38	5.34	1.57	3.52
0.90	0.90	0.01	1.48	3.77	1.58	3.05
0.90	0.90	0.05	1.52	3.95	1.63	3.33
0.90	0.90	0.10	1.49	3.81	1.71	3.12

A.2. Barrido de hiperparámetros modelo ACCLIP

Mapas de contorno del SSIM promedio en función del parámetro de compromiso α y del horizonte de predicción (número de cuadros futuros), para distintos valores del peso del flujo β. Cada subpanel muestra una configuración específica de β (desde 0.0 hasta 0.9), dispuestos de izquierda a derecha y de arriba hacia abajo. Estos resultados ilustran cómo, al aumentar β, las regiones de alto SSIM se desplazan y se expanden a lo largo de diferentes valores de α y longitudes de predicción. La barra de color a la derecha representa los valores de SSIM: los colores más oscuros indican valores bajos, mientras que los colores más claros corresponden a valores más altos.

Figura A.1: Mapas de contorno del SSIM promedio en función del parámetro α y del horizonte de predicción para distintos valores del peso del flujo β.




β = 0.4
β = 0.5
β = 0.6
β = 0.7
β = 0.8
β = 0.9
Predicciones
α
SSIM
0.9500
0.9333
0.9167
0.9000
0.8833
0.8667
0.8500
0.8333
0.8167
0.8000