Máster Universitario en Inteligencia Artificial

Redes neuronales y transformer físicamente informados. Revisión, reproducción y estudio de ablación de parámetros

Trabajo Fin de Máster

Autor:
Pablo Revuelto de Miguel

Tutores:
Diego Canales Aguilera
Alejandro Bandera Moreno

Curso 2025–2026

Resumen

Las redes neuronales informadas por la física (PINN) resuelven ecuaciones en derivadas parciales imponiendo la ecuación en la propia función de pérdida, sin necesidad de malla. Su formulación estándar, sin embargo, arrastra modos de fallo bien documentados (sesgo espectral, desequilibrio de la pérdida y fallos de propagación temporal) que han motivado arquitecturas específicas para mitigarlos. Una de las más recientes es PINNsFormer, que reemplaza el perceptrón multicapa por un Transformer que opera sobre una pseudo-secuencia temporal construida alrededor de cada punto de colocación.

El origen de este trabajo está en un intento de aplicar PINNsFormer a un problema real de elasticidad, la resolución del campo de desplazamientos bidimensional de una viga empotrada. Las dificultades encontradas en esa tarea llevaron a retroceder un paso y revisar las bases, es decir, comprobar hasta qué punto los resultados publicados de la propia arquitectura se sostienen antes de exigirle problemas nuevos. Con ese fin, este trabajo somete a PINNsFormer a un examen independiente sobre los cuatro problemas de referencia empleados en el artículo que la propone (convección, reacción, onda y Navier–Stokes), con dos objetivos: reproducir las cifras publicadas y estudiar mediante una ablación de parámetros qué componentes de la arquitectura explican su comportamiento. Todos los experimentos se ejecutan a través de un único punto de entrada determinista, de modo que una misma configuración produce el mismo resultado con independencia de la máquina.

La reproducción muestra que las cifras del artículo original se recuperan solo bajo condiciones de entrenamiento muy concretas, y que factores ajenos a la arquitectura, como la precisión aritmética con la que se entrena o el valor de un paso temporal auxiliar, alteran los resultados en mayor medida que los cambios de modelo. El estudio de ablación, con sesenta y nueve ejecuciones en precisión doble, indica que la maquinaria de atención no explica el acierto de la arquitectura, que su capacidad interna está sobredimensionada y que el paso temporal es su único modo de fallo propio. Sobre estos problemas, la precisión numérica y el protocolo de entrenamiento explican una parte mayor de las diferencias publicadas que la propia arquitectura.

Palabras clave: redes neuronales informadas por la física, PINNsFormer, Transformer, ecuaciones en derivadas parciales, modos de fallo, sesgo espectral, estudio de ablación, precisión numérica, reproducibilidad.

Abstract

Physics-informed neural networks (PINNs) solve partial differential equations by enforcing the equation directly in the loss function, with no mesh required. Their standard formulation, however, suffers from well-documented failure modes (spectral bias, loss imbalance and temporal propagation failures) that have prompted architectures designed to mitigate them. One of the most recent is PINNsFormer, which replaces the multilayer perceptron with a Transformer operating on a temporal pseudo-sequence built around each collocation point.

This work originated in an attempt to apply PINNsFormer to a real elasticity problem, namely the two-dimensional displacement field of a cantilever beam. The difficulties encountered in that task motivated a step back to review the foundations, that is, to verify to what extent the published results of the architecture itself hold before demanding new problems from it. To that end, this work subjects PINNsFormer to an independent examination on the four benchmark problems used in the paper that proposes it (convection, reaction, wave and Navier–Stokes), with two aims: to reproduce the published figures and to study, through a parameter ablation, which components of the architecture explain its behaviour. Every experiment runs through a single deterministic entry point, so that a given configuration yields the same result regardless of the machine.

The reproduction shows that the figures of the original paper are recovered only under very specific training conditions, and that factors unrelated to the architecture, such as the arithmetic precision of training or the value of an auxiliary time step, alter the results more than the choice of model does. The ablation study, comprising sixty-nine double-precision runs, indicates that the attention machinery does not account for the architecture's accuracy, that its internal capacity is oversized and that the time step is its only intrinsic failure mode. On these problems, numerical precision and the training protocol explain a larger share of the published differences than the architecture itself.

Keywords: physics-informed neural networks, PINNsFormer, Transformer, partial differential equations, failure modes, spectral bias, ablation study, numerical precision, reproducibility.

Agradecimientos

A mi familia, amigos y quien siempre estuvo conmigo en este camino.

Acrónimos

Acrónimo	Significado
PINN	*Physics-Informed Neural Network* (red neuronal informada por la física)
PDE/EDP	*Partial Differential Equation* / ecuación en derivadas parciales
ODE/EDO	*Ordinary Differential Equation* / ecuación diferencial ordinaria
MLP	*Multi-Layer Perceptron* (perceptrón multicapa)
SciML	*Scientific Machine Learning*
AD	*Automatic Differentiation* (autodiferenciación)
FEM	*Finite Element Method* (método de los elementos finitos)
IC/CI	*Initial Condition* / condición inicial
BC/CC	*Boundary Condition* / condición de contorno
NTK	*Neural Tangent Kernel*
FLS	*First-Layer Sine*
QRes	*Quadratic Residual* network
NS	Navier–Stokes
L-BFGS	*Limited-memory Broyden–Fletcher–Goldfarb–Shanno*
rMAE	error absoluto medio relativo (norma ℓ_1)
rRMSE	raíz del error cuadrático medio relativo (norma ℓ_2)
WaveAct	activación periódica aprendible de PINNsFormer ($w_1 \sin x + w_2 \cos x$)

Índice de figuras

Índice de tablas

Índice general

1

Introducción

1.1. Motivación y contexto

La resolución numérica de ecuaciones en derivadas parciales (EDP) ocupa un lugar central en la ciencia y la ingeniería. La propagación de ondas, la dinámica de fluidos, la difusión del calor o la cinética de las reacciones químicas se describen con este tipo de ecuaciones, y de su resolución dependen tareas tan dispares como el diseño de una estructura, la predicción de un campo de velocidades o el control de un proceso térmico.

El método de los elementos finitos (FEM) (Zienkiewicz et al., 2005) y los métodos pseudoespectrales (Canuto et al., 2006) llevan décadas siendo la referencia para abordarlas; su madurez teórica, sus garantías de convergencia y su fiabilidad numérica están fuera de discusión. Tienen, no obstante, un inconveniente conocido, exigen discretizar el dominio en una malla y resolver sistemas algebraicos de gran tamaño, y ese coste crece deprisa cuando la dimensionalidad del problema aumenta o cuando no se busca una única simulación, sino recorrer un espacio de parámetros. En un proceso de diseño u optimización, donde conviene evaluar muchas combinaciones de geometría, material o carga, repetir la simulación completa para cada configuración resulta caro, y la malla que sirve para un caso rara vez es la óptima para el siguiente.

En la última década, el auge del aprendizaje automático científico (*scientific machine learning*, SciML) ha abierto una vía complementaria a la numérica clásica (Karniadakis et al., 2021). En el aprendizaje supervisado habitual, una red neuronal se entrena a partir de ejemplos etiquetados, se le muestran pares formados por una entrada y la salida que debería producir, y sus parámetros se ajustan hasta que la red imita ese comportamiento. La idea del SciML es distinta, se trata de aproximar la solución de una EDP mediante una red neuronal entrenada imponiendo directamente que la propia función aprendida satisfaga la ecuación física, junto con sus condiciones de contorno e iniciales, sin necesidad de conocer la solución de antemano. Esta idea, ya esbozada por Lagaris et al. (1998) a finales de

los años noventa y formalizada por Raissi et al. (2019) bajo el nombre de *Physics-Informed Neural Networks* (PINN), reformula el problema de resolver una EDP como un problema de optimización, en el que la red se entrena minimizando una función de pérdida construida con los residuos de la ecuación, evaluados de forma exacta gracias a la autodiferenciación que incorporan las bibliotecas de aprendizaje profundo habituales, como PyTorch o TensorFlow (Baydin et al., 2018). Este planteamiento aporta varias ventajas. Una vez entrenada, la red ofrece una solución continua y derivable, evaluable en cualquier punto del dominio sin necesidad de malla. A ello se suma que el conocimiento físico del problema entra en el entrenamiento como un sesgo inductivo, de modo que la red depende menos de datos previos, que suelen ser escasos o costosos de generar. Y en los problemas inversos o con observaciones parciales, precisamente aquellos en los que los métodos clásicos encajan peor, poder combinar física y datos dentro de una misma pérdida se vuelve una ventaja de peso.

Ese atractivo convive, sin embargo, con una dificultad que la literatura ha documentado con detalle. Las PINN basadas en perceptrones multicapa (MLP), que constituyen la formulación original de Raissi et al. (2019), presentan modos de fallo característicos. Tienden a converger a soluciones excesivamente suaves, o incluso triviales, cuando la solución verdadera contiene componentes de alta frecuencia o de carácter multiescala, un comportamiento ligado al sesgo espectral de las redes neuronales (Rahaman et al., 2019). Sufren además un desequilibrio acusado entre los distintos términos de la función de pérdida (residuo de la EDP, condiciones de contorno, condiciones iniciales), cuyas escalas de gradiente pueden diferir en varios órdenes de magnitud durante el entrenamiento (Wang et al., 2021a). Y propagan mal la información a lo largo del tiempo en problemas transitorios, hasta el punto de que la red puede ajustar la condición inicial y, pese a ello, fallar en reconstruir la evolución posterior (Krishnapriyan et al., 2021). Los trabajos citados coinciden en que estos fallos tienen su origen en la propia formulación del método, de modo que prolongar el entrenamiento o ajustar los hiperparámetros no basta, en general, para corregirlos.

Frente a este panorama han surgido arquitecturas que intentan atacar directamente alguno de esos modos de fallo. Una de las más recientes es PINNsFormer, propuesta por Zhao et al. (2024a) en un artículo presentado en la conferencia ICLR de 2024. Ese artículo es la pieza en torno a la cual gira todo este trabajo, y conviene precisar desde ahora el papel que desempeña. En él sus autores definen la arquitectura, que sustituye el MLP por un *encoder–decoder* de tipo Transformer (Vaswani et al., 2017) construido sobre una pseudo-secuencia temporal alrededor de cada punto de colocación; fijan los cuatro problemas de referencia sobre los que la validan (convección, reacción, onda y Navier–Stokes); y publican las cifras de error con las que afirman superar con claridad a las PINN convencionales. Esas cifras, esos problemas y ese protocolo experimental son, respectivamente, el objetivo de

la reproducción, el terreno de juego y el punto de partida metodológico de esta memoria.

La motivación para examinar ese artículo con este nivel de detalle no es abstracta. Este trabajo nació como un intento de aplicar PINNsFormer a un problema real de elasticidad, la resolución del campo de desplazamientos bidimensional de una viga empotrada. Los resultados de aquella tarea fueron consistentemente peores de lo que las cifras publicadas hacían esperar, y esa distancia entre lo prometido y lo observado condujo a la pregunta que motiva este trabajo. Antes de exigir a la arquitectura problemas nuevos, ¿se sostienen sus resultados publicados sobre sus propios problemas de referencia? La reproducibilidad de los resultados se ha convertido en una de las mayores preocupaciones del aprendizaje automático, y una arquitectura nueva no queda validada solo con las cifras que publican quienes la proponen. Importa además examinar hasta dónde llega la mejora, bajo qué condiciones se mantiene y en qué escenarios la arquitectura vuelve a degradarse.

Ese examen es el objeto de este trabajo, y su desarrollo marca la estructura de la memoria. El capítulo 2 recorre el camino que va desde las redes neuronales convencionales hasta la PINN estándar y sus modos de fallo, y el capítulo 3 explica el Transformer y la arquitectura PINNsFormer. Sobre esa base, la arquitectura se somete a dos pruebas de naturaleza distinta. La primera, en el capítulo 4, es la reproducción de los problemas con los que sus autores la validaron, comprobando si las mejoras publicadas se sostienen al reimplementar y reentrenar los modelos en un entorno controlado y documentado. La segunda, en el capítulo 5 y de carácter más exploratorio, consiste en llevar la arquitectura al límite. Se revisa la literatura que documenta los modos de fallo de las PINN y de la propia PINNsFormer, y se realiza un estudio de ablación de sus parámetros para entender qué decisiones de diseño resultan determinantes y en qué situaciones la arquitectura deja de aportar ventaja. El interés del trabajo no reside, por tanto, en una aplicación concreta de ingeniería, sino en comprender una propuesta metodológica reciente, delimitando qué resuelve, qué no resuelve y por qué.

1.2. Objetivos

Objetivo general

Estudiar a fondo la arquitectura PINNsFormer dentro del marco más amplio de las *Physics-Informed Neural Networks*, reproduciendo los resultados de su artículo original y analizando sus modos de fallo y la sensibilidad de su comportamiento a las decisiones de diseño mediante un estudio de ablación de parámetros, con el

fin de delimitar de manera fundamentada qué aporta y dónde falla respecto a la PINN convencional.

Objetivos específicos

El objetivo general se concreta en los siguientes objetivos parciales. En primer lugar, explicar las PINN estándar (su concepto, su motivación histórica desde Lagaris et al. (1998) y Raissi et al. (2019), y su formulación matemática) y situarlas frente al aprendizaje profundo supervisado convencional y frente a los métodos numéricos clásicos de resolución de EDP. En segundo lugar, recopilar y analizar los modos de fallo de las PINN documentados en la literatura (sesgo espectral, patologías de gradiente, desequilibrio de la función de pérdida y fallos de propagación temporal), que motivan las arquitecturas que pretenden mitigarlos.

Sobre esa base teórica, el tercer objetivo es estudiar en detalle la arquitectura PINNsFormer: el mecanismo de pseudo-secuencia temporal, el esquema *encoder–decoder* de tipo Transformer y la activación periódica WaveAct, relacionando la descripción del artículo con la implementación efectivamente empleada en este trabajo. El cuarto es reproducir de forma independiente los problemas de referencia del artículo original (convección, ecuación de reacción 1D, ecuación de ondas 1D y flujo de Navier–Stokes), comparando PINNsFormer con la PINN convencional y con otras variantes mediante las métricas de error relativo ℓ_1 y ℓ_2, y contrastando los resultados con las cifras publicadas.

Los tres últimos objetivos corresponden a la parte experimental propia. El quinto es diseñar y ejecutar un estudio de ablación sobre los parámetros de PINNsFormer (longitud y paso de la pseudo-secuencia, dimensión del *embedding*, número de cabezas de atención y de capas, función de activación y esquema de optimización) para determinar cuáles resultan determinantes para su rendimiento. El sexto, analizar los resultados de la reproducción y de la ablación ateniéndose a los hechos medidos, sin extrapolar más allá de lo que las cifras sostienen, y contrastarlos con la literatura para distinguir los fallos propios de esta reproducción de los ya descritos como estructurales. El séptimo y último, sintetizar en qué condiciones PINNsFormer mejora a la PINN convencional y en cuáles persisten sus modos de fallo, y señalar las líneas de trabajo futuro que el análisis deja abiertas.

1.3. Alcance y estructura del documento

El alcance de este trabajo es metodológico y experimental, no la aplicación a un caso de ingeniería concreto. Se centra en el estudio, la reproducción y el análisis

riguroso de una arquitectura de PINN (PINNsFormer) sobre los problemas de referencia con los que fue validada, sin pretender resolver un problema industrial específico ni introducir una formulación física propia. Esta delimitación responde a la naturaleza del objetivo, pues comprender las virtudes y los límites reales de la arquitectura exige un terreno controlado y comparable (los mismos *benchmarks* que la literatura emplea) antes que un escenario nuevo, en el que resultaría difícil separar las limitaciones de la red de las dificultades intrínsecas del problema. Las aplicaciones a problemas estructurales u otros casos de ingeniería quedan apuntadas, llegado el caso, como líneas de trabajo futuro.

La reproducibilidad recibe un cuidado particular. Todos los experimentos de reproducción y ablación se ejecutan a través de un único punto de entrada con semillas fijas y un entorno de dependencias controlado, de modo que una misma configuración produzca el mismo resultado con independencia de la máquina en que se entrene. Esta decisión, descrita en el apéndice B, resulta necesaria para que la comparación con las cifras publicadas tenga sentido y para que el estudio de ablación aísle el efecto de cada parámetro sin contaminarlo con la variabilidad del propio entrenamiento.

El documento se organiza en seis capítulos. El capítulo 1, que aquí concluye, ha presentado la motivación, el contexto y los objetivos del trabajo. El capítulo 2 recorre el camino desde las redes neuronales hasta las PINN, pasando por el aprendizaje profundo supervisado y la aproximación de funciones, el concepto y la historia de la PINN vainilla, su formulación matemática, su comparación con el resto del aprendizaje profundo y con los métodos numéricos clásicos, y los modos de fallo que la literatura le atribuye. El capítulo 3 se dedica a la arquitectura de PINNsFormer. Parte de una explicación del Transformer y de su mecanismo de atención, continúa con la motivación de la arquitectura, su pseudo-secuencia temporal, su esquema *encoder–decoder* y la activación WaveAct. El capítulo 4 aborda la reproducción independiente del artículo original. Detalla la metodología de reproducción y la infraestructura experimental empleada, presenta los resultados obtenidos sobre los cuatro problemas de referencia y los contrasta con las cifras publicadas, con un análisis de las discrepancias apoyado en la literatura reciente. El capítulo 5 constituye una de las aportaciones centrales del trabajo. Revisa la literatura reciente sobre los modos de fallo de las PINN y de PINNsFormer y presenta el estudio de ablación de parámetros, con sus resultados y su discusión. El capítulo 6 cierra el documento con las conclusiones y las líneas futuras que quedan abiertas.

2

De las redes neuronales a las PINN

Antes de estudiar PINNsFormer conviene fijar con cuidado el terreno sobre el que se construye. Este capítulo recorre ese terreno en orden ascendente. Parte de la red neuronal como aproximador de funciones, introduce la idea de informar a la red con la física del problema, formaliza la PINN estándar y la confronta con el resto del aprendizaje profundo y con los métodos numéricos clásicos. Termina con los modos de fallo que la literatura le atribuye, que son la verdadera motivación de las arquitecturas posteriores y, en particular, de la que ocupa el grueso de este trabajo.

2.1. Aprendizaje profundo supervisado y aproximación de funciones

Una red neuronal hacia adelante (*feedforward*) define una familia parametrizada de funciones. En su forma más común, el perceptrón multicapa (MLP), la salida se obtiene componiendo transformaciones afines con una no linealidad aplicada componente a componente. Para una red de L capas, con entrada $\mathbf{x} \in \mathbb{R}^{d_0}$, parámetros $\theta = \{(\mathbf{W}_\ell, \mathbf{b}_\ell)\}_{\ell=1}^{L}$ y función de activación σ, la red calcula

$$\mathbf{h}_0 = \mathbf{x}, \qquad \mathbf{h}_\ell = \sigma(\mathbf{W}_\ell\, \mathbf{h}_{\ell-1} + \mathbf{b}_\ell) \;\; (\ell = 1, \dots, L-1), \qquad f_\theta(\mathbf{x}) = \mathbf{W}_L\, \mathbf{h}_{L-1} + \mathbf{b}_L. \tag{2.1}$$

La elección de σ no es menor. Activaciones suaves e infinitamente derivables, como la tangente hiperbólica, son habituales en el contexto que nos ocupa, porque la red no sólo debe aproximar una función, sino también sus derivadas; una activación con derivadas nulas a trozos, como la ReLU, complica la representación de los términos diferenciales que aparecerán más adelante.

El interés de (2.1) como aproximador descansa en un resultado clásico. El teorema de aproximación universal, en la versión de Hornik et al. (1989), establece que una red con una sola capa oculta y un número suficiente de neuronas puede aproximar, con la precisión que se desee, cualquier función continua sobre un compacto.

El teorema garantiza la existencia de los parámetros, pero no dice cómo encontrarlos ni cuántas neuronas hacen falta. Esta distinción entre existencia y obtención de los parámetros conviene retenerla porque, como se verá al final del capítulo, los fallos de las PINN no se deben a una falta de capacidad expresiva de la red, sino a la dificultad de la optimización que debe encontrar esos parámetros.

En el aprendizaje supervisado convencional, los parámetros se ajustan minimizando una pérdida que mide la discrepancia entre las predicciones de la red y un conjunto de ejemplos etiquetados $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$. El caso más ilustrativo es el de la regresión. Si la red f_θ debe predecir un valor continuo, la elección habitual es el error cuadrático medio,

$$\mathcal{L}(\theta) \;=\; \frac{1}{N}\sum_{i=1}^{N}\big\| f_\theta(\mathbf{x}_i) - \mathbf{y}_i\big\|^2, \tag{2.2}$$

que penaliza el cuadrado de la distancia entre lo que la red predice para cada entrada $\mathbf{x}_i$ y la etiqueta $\mathbf{y}_i$ que debería haber producido. Minimizar (2.2) respecto a θ ajusta la red a los datos, y esta misma estructura (un promedio de discrepancias al cuadrado) reaparecerá en la función de pérdida de las PINN, con la diferencia esencial de que allí la discrepancia no se mide contra etiquetas, sino contra la propia ecuación diferencial. El ingrediente que hace viable ese ajuste a gran escala es la retropropagación, esto es, el cálculo del gradiente de la pérdida respecto a θ mediante diferenciación automática (AD). La AD no es ni diferenciación numérica por diferencias finitas ni manipulación simbólica de fórmulas, sino que explota la regla de la cadena sobre el grafo de operaciones elementales que define la red, de modo que las derivadas se obtienen con precisión de máquina y a un coste comparable al de la propia evaluación (Baydin et al., 2018). Esta herramienta, pensada inicialmente para derivar respecto a los parámetros, es justo lo que permite derivar también respecto a las entradas de la red, y de ahí que sea la pieza que hace posible todo lo que sigue.

2.2. PINN vainilla. Concepto, motivación e historia

La idea de resolver una ecuación diferencial con una red neuronal es anterior al auge reciente del aprendizaje profundo. Lagaris et al. (1998) propusieron, ya en 1998, expresar la solución de una ecuación diferencial como una solución de prueba compuesta por dos partes: un término fijo que satisface exactamente las condiciones de contorno e iniciales, y un término correctivo, dado por una red neuronal, que se anula sobre la frontera por construcción. Con ese diseño, las condiciones de contorno quedaban impuestas de forma exacta y sólo restaba ajustar la red para que el residuo de la ecuación se anulase en un conjunto de puntos del dominio.

El enfoque era elegante, pero exigía construir a mano la estructura de la solución de prueba para cada geometría, lo que limitaba su aplicación a dominios sencillos.

El planteamiento que ha terminado por imponerse relaja esa exigencia. Raissi et al. (2019) reformularon la idea bajo el nombre de *Physics-Informed Neural Networks* y propusieron imponer tanto la ecuación como sus condiciones de contorno e iniciales de forma débil, como términos adicionales de la función de pérdida, en lugar de codificarlas en la arquitectura. La red u_θ toma como entrada las coordenadas del problema (por ejemplo, posición y tiempo, (x,t)) y devuelve directamente el valor de la solución en ese punto,

$$u_\theta : (x,t) \longmapsto u_\theta(x,t) \approx u(x,t). \tag{2.3}$$

La predicción es, por tanto, punto a punto, ya que la red no consume una malla ni un estado previo, sino que aproxima el campo continuo en cualquier coordenada que se le presente. Este cambio (de codificar la física en la arquitectura a codificarla en la pérdida) es lo que dota al método de su generalidad, y también lo que abre la puerta a los modos de fallo que se estudian al final del capítulo. Al ser blandas, las restricciones físicas compiten entre sí durante el entrenamiento en lugar de cumplirse por construcción.

Conviene distinguir desde ahora dos regímenes de uso, porque la literatura los trata de forma distinta (Raissi et al., 2019; Karniadakis et al., 2021). Lo que los separa no es la presencia o ausencia de datos de observación (ambos pueden incorporarlos) sino qué se conoce de la ecuación y qué constituye la incógnita. En el problema *directo* la ecuación está por completo determinada, con todos sus coeficientes conocidos, y lo único que se busca es su solución; la red puede entrenarse solo con la física, aunque nada impide añadir observaciones dispersas del campo, cuando se dispone de ellas, para guiar o acelerar la convergencia. En el problema *inverso*, en cambio, algún coeficiente o parámetro de la ecuación es desconocido, y son precisamente esas observaciones las que permiten recuperarlo, y la misma maquinaria ajusta entonces de forma simultánea la solución y los parámetros que faltan. La diferencia esencial es, pues, el objeto de la búsqueda (únicamente el campo, frente al campo y ciertos coeficientes) y el papel de los datos, opcionales en el primer caso pero indispensables en el segundo. Esta convivencia natural de física y observaciones dentro de una sola pérdida es una de las razones del interés que despertó el método, y reaparece en el problema de Navier–Stokes de la reproducción del capítulo 4.

2.3. Formulación matemática de la PINN

Considérese una EDP definida sobre un dominio espacio-temporal $\Omega \times [0,T]$, escrita en forma residual mediante un operador diferencial $\mathcal{N}$,

$$\mathcal{R}(x,t) \;:=\; \partial_t u(x,t) + \mathcal{N}[u](x,t) \;=\; 0, \qquad (x,t) \in \Omega \times (0,T], \tag{2.4}$$

sujeta a una condición inicial $u(x,0) = u_0(x)$ sobre Ω y a una condición de contorno $\mathcal{B}[u](x,t) = g(x,t)$ sobre $\partial\Omega \times (0,T]$. El operador $\mathcal{N}$ recoge la parte espacial de la dinámica y puede ser no lineal; en los problemas de la reproducción adoptará formas concretas (advección lineal, reacción logística, operador de ondas) que se detallan en su momento.

La PINN sustituye u por la red u_θ de (2.3) y mide hasta qué punto u_θ incumple (2.4) y sus condiciones. Para ello se evalúa el residuo en un conjunto de puntos de colocación $\{(x_r^i, t_r^i)\}_{i=1}^{N_r}$ del interior del dominio, y las condiciones inicial y de contorno en sus respectivos conjuntos de muestreo. La derivada temporal $\partial_t u_\theta$ y las derivadas espaciales que esconde $\mathcal{N}$ (que pueden ser de segundo orden o superiores) se calculan por diferenciación automática respecto a las entradas de la red, sin discretización alguna. La función de pérdida agrega entonces los tres términos en un único objetivo,

$$\mathcal{L}(\theta) \;=\; \lambda_r\, \mathcal{L}_r(\theta) + \lambda_{ic}\, \mathcal{L}_{ic}(\theta) + \lambda_{bc}\, \mathcal{L}_{bc}(\theta), \tag{2.5}$$

con

$$\mathcal{L}_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \big| \mathcal{R}(x_r^i, t_r^i) \big|^2, \qquad \mathcal{L}_{ic}(\theta) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} \big| u_\theta(x_{ic}^i, 0) - u_0(x_{ic}^i) \big|^2, \tag{2.6}$$

y un término de contorno análogo sobre la frontera espacial,

$$\mathcal{L}_{bc}(\theta) = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \big| \mathcal{B}\big(u_\theta\big)(x_{bc}^i, t_{bc}^i) \big|^2, \qquad (x_{bc}^i, t_{bc}^i) \in \partial\Omega \times [0,T], \tag{2.7}$$

donde $\mathcal{B}$ es el operador que expresa la condición de contorno (por ejemplo, $\mathcal{B}(u_\theta) = u_\theta - g$ para una condición de Dirichlet $u = g$). Cuando se dispone de observaciones de la solución, se añade un cuarto término de datos, $\mathcal{L}_{data}$, idéntico en forma a un error cuadrático medio supervisado; es lo que conecta el problema directo con el inverso.

Tres observaciones sobre (2.5) importan para lo que sigue. La primera es que la solución no se obtiene resolviendo un sistema lineal, sino minimizando un objetivo no convexo respecto a θ; el resultado, por tanto, hereda todas las dificultades de la optimización de redes neuronales. La segunda es que los pesos $\lambda_r, \lambda_{ic}, \lambda_{bc}$ fijan el equilibrio entre satisfacer la ecuación y respetar las condiciones, de modo que

de su elección depende, en buena medida, que el entrenamiento converja a la solución física o a un mínimo espurio. La tercera es que la calidad de la solución se mide, frente a una referencia $u^\star$, con errores relativos en norma ℓ_1 y ℓ_2,

$$\mathrm{rMAE} = \frac{\sum_i |u_\theta(x_i,t_i) - u^\star(x_i,t_i)|}{\sum_i |u^\star(x_i,t_i)|}, \qquad \mathrm{rRMSE} = \sqrt{\frac{\sum_i |u_\theta(x_i,t_i) - u^\star(x_i,t_i)|^2}{\sum_i |u^\star(x_i,t_i)|^2}}, \tag{2.8}$$

que son las métricas con las que se reportan los resultados de la reproducción y de la ablación en los capítulos 4 y 5.

En cuanto a la optimización en sí, la práctica habitual combina un optimizador de primer orden para una fase inicial con un método cuasi-Newton, típicamente L-BFGS, para el refinamiento final. L-BFGS aprovecha una aproximación de la curvatura de la pérdida y suele alcanzar residuos mucho menores que el descenso de gradiente estocástico en estos problemas de bajo número de parámetros y datos deterministas, a costa de una mayor sensibilidad a la condición del problema.

2.4. Las PINN frente al *deep learning* clásico y a los métodos numéricos

Aunque comparte la maquinaria del aprendizaje profundo, una PINN se aparta del paradigma supervisado habitual en un punto esencial, y es que la señal de entrenamiento no procede de datos etiquetados, sino de una ecuación. El conocimiento físico actúa como un sesgo inductivo fuerte que sustituye, total o parcialmente, a los ejemplos. Esa diferencia tiene consecuencias prácticas. Una red supervisada generaliza interpolando entre observaciones y degrada su rendimiento fuera de la distribución de los datos; una PINN, en cambio, está anclada a la física en todo el dominio de colocación, de modo que no necesita observaciones para ofrecer una solución, aunque ello no la libra (como se verá) de converger a soluciones físicamente incorrectas cuando la optimización falla. Frente al aprendizaje supervisado puro, la PINN gana en eficiencia de datos y en consistencia física. Pierde, a cambio, en la dificultad de la optimización, dado que la pérdida supervisada tampoco es convexa en los parámetros de una red, pero la pérdida de una PINN incorpora además derivadas de la propia red, lo que empeora su condicionamiento y da lugar a los modos de fallo que se describen en la sección 2.5.

La comparación con los métodos numéricos clásicos es más matizada y conviene no exagerar las ventajas de las PINN. El FEM y los métodos pseudoespectrales ofrecen garantías de convergencia, estimaciones de error y una madurez de décadas; para un problema directo bien planteado en dos o tres dimensiones, son casi

siempre más rápidos y más fiables que una PINN. Las ventajas del enfoque neuronal aparecen en nichos concretos. Uno son los problemas en dimensión alta, donde el coste del mallado se dispara, y los estudios paramétricos o de diseño, en los que interesa explorar muchas combinaciones de parámetros sin repetir desde cero una simulación completa para cada una. Otro, los problemas inversos y de asimilación de datos, en los que combinar física y observaciones resulta natural. Cabe mencionar también que la red entrenada proporciona una representación continua de la solución cuyas derivadas de cualquier orden se obtienen por diferenciación automática; un método de elementos finitos puede reconstruir a posteriori una representación evaluable en cualquier punto, pero la obtención de derivadas de orden alto exige un posproceso que aquí resulta inmediato. La PINN no viene, pues, a reemplazar al FEM, sino a cubrir escenarios en los que el mallado o la falta de datos lo penalizan; las mejoras que reportan arquitecturas como PINNsFormer (Zhao et al. (2024a)) se miden sobre problemas de referencia escogidos, no sobre el conjunto de los problemas de la física computacional, y deben leerse con esa cautela.

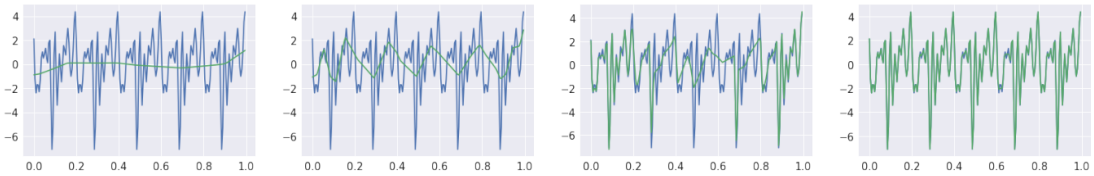
2.5. Modos de fallo documentados de las PINN vainilla

Las PINN basadas en MLP fallan, y fallan de maneras que la literatura ha sabido caracterizar. Importa subrayar que estos fallos no son de capacidad expresiva (el teorema de aproximación garantiza que una red suficientemente grande puede representar la solución) sino de optimización, pues la red podría representar la solución correcta, pero el entrenamiento no la encuentra. Esta sección recoge los modos de fallo mejor documentados y las estrategias propuestas para mitigarlos, porque juntos constituyen la motivación directa del capítulo siguiente.

2.5.1. Sesgo espectral y soluciones excesivamente suaves

El primer modo de fallo se demostró, en rigor, fuera del contexto de las PINN, y solo después se trasladó a ellas. Rahaman et al. (2019) estudiaron cómo una red totalmente conectada ajusta funciones objetivo sintetizadas a partir de un espectro de frecuencias prescrito y conocido de antemano. Al seguir, época a época, con qué rapidez la red reproduce cada componente de Fourier del objetivo, observaron un patrón sistemático. Las frecuencias bajas se aprenden primero y las altas mucho más tarde, si es que llegan a aprenderse. El hallazgo empírico lo respaldaron con un argumento analítico sobre el espectro de Fourier de una red con activaciones ReLU, cuyas amplitudes decaen con la frecuencia y cuyas componentes

de baja frecuencia resultan más robustas frente a perturbaciones de los pesos. Es esa doble evidencia (la medición directa del ritmo de aprendizaje por frecuencia y la cota teórica sobre las amplitudes) la que sostiene que el sesgo espectral es una propiedad estructural del entrenamiento por gradiente, y no un artefacto de un problema concreto. La figura 2.1 reproduce esa dinámica, donde la red ajusta primero la envolvente de baja frecuencia del objetivo y solo tras muchas más iteraciones recupera sus oscilaciones de alta frecuencia.



(a) Iteration 100 (b) Iteration 1000 (c) Iteration 10000 (d) Iteration 80000

Figura 2.1: Sesgo espectral. Función aprendida (verde) sobre la función objetivo (azul) a lo largo del entrenamiento, para un objetivo que superpone frecuencias $\kappa = (5, 10, \ldots, 45, 50)$. Las componentes de baja frecuencia se ajustan primero y las de alta frecuencia mucho más tarde. Reproducido de Rahaman et al. (2019).

Su relevancia para las PINN se estableció por separado. En una PINN, la pérdida basada en el residuo admite un mínimo trivial cuando la solución es suave, de modo que una red sesgada hacia lo suave tiende a estancarse en él, pues satisface la ecuación en promedio, pero pierde las estructuras finas (frentes abruptos, gradientes pronunciados) o colapsa a un campo casi constante que minimiza el residuo sin parecerse a la solución física. Que este mecanismo es el que limita a las PINN en problemas multiescala quedó confirmado, de manera indirecta pero convincente, por el trabajo sobre el sesgo de autovectores de las redes de características de Fourier (Wang et al., 2021b). Al precondicionar la entrada con una capa de Fourier que reequilibra el espectro del núcleo tangente, sus autores resolvieron EDP multiescala que la PINN estándar no alcanzaba. Que una corrección dirigida precisamente al espectro rescate esos problemas es la mejor prueba de que era el sesgo espectral lo que los hacía fracasar. Este mismo diagnóstico motiva de forma explícita la activación periódica que introduce PINNsFormer (capítulo 3).

2.5.2. Patologías de gradiente y desequilibrio de la pérdida

El segundo modo de fallo nace de la estructura multitérmino de (2.5) y se demostró observando directamente lo que ocurre con los gradientes durante el entrenamiento. Wang et al. (2021a) instrumentaron el entrenamiento de PINN sobre EDP rígidas (la ecuación de Helmholtz y un problema de flujo, entre otras) y registraron las distribuciones de los gradientes que la retropropagación asigna a cada término de la pérdida. La medición reveló que los gradientes del término de residuo dominan a los de las condiciones de contorno por varios órdenes de magnitud, de

manera que el optimizador ajusta bien la ecuación en el interior mientras deja las condiciones prácticamente sin optimizar, o al revés. La figura 2.2 muestra ese desequilibrio capa a capa. Lo que eleva esta observación a la categoría de modo de fallo es la relación de causa a efecto que los autores cierran a continuación. Al introducir un esquema de recocido de la tasa de aprendizaje que reescala los términos según la magnitud de sus gradientes, el desequilibrio desaparece y el error de la solución cae de forma acusada. El fallo se mide, se atribuye a una causa concreta y se corrige actuando sobre esa causa.

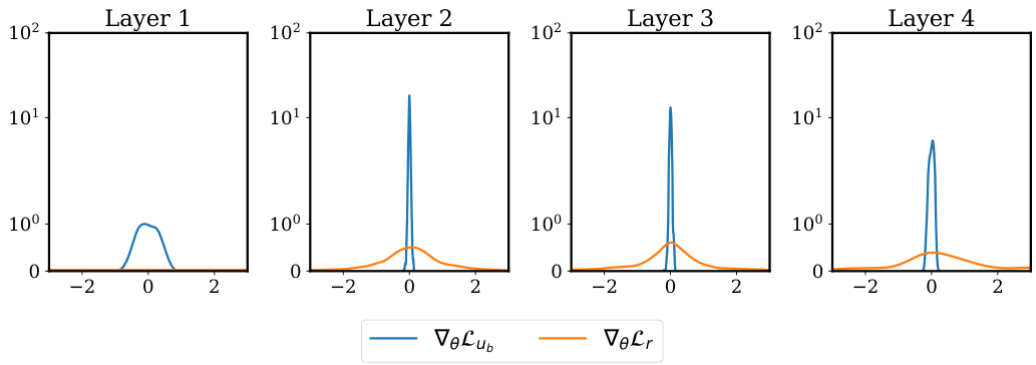


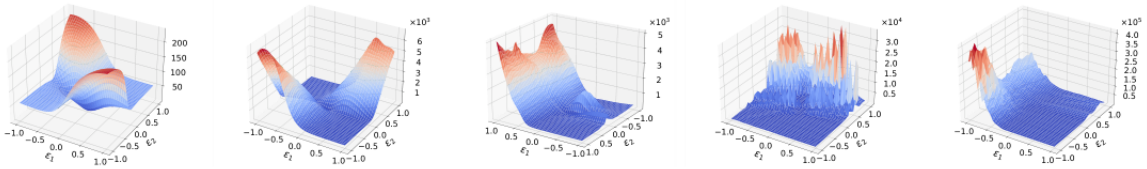
Figura 2.2: Patologías de gradiente en una PINN sobre la ecuación de Helmholtz. Histogramas de los gradientes retropropagados del residuo ($\nabla_\theta \mathcal{L}_r$) y de la condición de contorno ($\nabla_\theta \mathcal{L}_{u_b}$) en cada capa. Los gradientes del residuo son varios órdenes de magnitud mayores, lo que descompensa la optimización. Reproducido de Wang et al. (2021a).

Una segunda línea, formulada desde la teoría del *neural tangent kernel* (NTK), explica por qué el estancamiento se produce y cuándo (Wang et al., 2022). Sus autores derivaron el núcleo tangente de una PINN y probaron que su espectro de autovalores gobierna la velocidad a la que converge cada componente de la pérdida; al seguir esos autovalores durante el entrenamiento comprobaron que difieren en órdenes de magnitud, lo que condena a las componentes lentas a no aprenderse en un presupuesto de cómputo razonable. De ese análisis se deducen pesos $\lambda_r, \lambda_{ic}, \lambda_{bc}$ que igualan las tasas de convergencia. Ambos resultados interesan aquí porque la elección de los λ es uno de los factores que se controlan en la reproducción, y porque la ponderación por NTK reaparece más adelante como una de las variantes evaluadas.

2.5.3. Fallos de propagación temporal en problemas transitorios

El tercer modo de fallo es quizá el más revelador, porque desafía la intuición de que una pérdida pequeña implica una solución correcta, y se demostró con un diseño experimental especialmente limpio. Krishnapriyan et al. (2021) tomaron

ecuaciones de convección y de reacción con solución analítica conocida y barrieron de forma controlada el coeficiente de la ecuación (la velocidad de convección o la tasa de reacción). Constataron que, a partir de cierto umbral, el error relativo respecto a la solución exacta se dispara mientras la pérdida permanece baja. La red ajusta la condición inicial y el residuo en los puntos de colocación, pero no propaga la información hacia los tiempos posteriores como exige la causalidad del problema. Para sustentar que la causa es la optimización y no la representación, visualizaron el paisaje de pérdida perturbando el modelo ya entrenado a lo largo de los dos autovectores dominantes de la matriz hessiana y evaluando la pérdida resultante en cada perturbación, y mostraron que se vuelve progresivamente más rugoso y no convexo a medida que crece el coeficiente, de modo que el entrenamiento queda atrapado en mínimos ajenos a la solución física. La figura 2.3 recoge esa transición del paisaje de pérdida, de casi convexo a fuertemente accidentado, al aumentar el coeficiente.



(a) $\beta = 1.0$ (b) $\beta = 10.0$ (c) $\beta = 20.0$ (d) $\beta = 30.0$ (e) $\beta = 40.0$

β	1	10	20	30	40
Relative error	7.84×10^{-3}	1.08×10^{-2}	7.50×10^{-1}	8.97×10^{-1}	9.61×10^{-1}
Absolute error	3.17×10^{-3}	6.03×10^{-3}	4.32×10^{-1}	5.42×10^{-1}	5.82×10^{-1}

Figura 2.3: Paisaje de pérdida de una PINN sobre la convección 1D para valores crecientes del coeficiente β. Suave y casi convexo con β pequeño, se vuelve cada vez más rugoso y no convexo al crecer β, y el error relativo se dispara. Reproducido de Krishnapriyan et al. (2021).

El argumento se completa por descarte, ya que aumentar el tamaño de la red o prolongar el entrenamiento no corrige el fallo (lo que excluye la falta de capacidad), mientras que reformular el problema para respetar el orden temporal sí lo hace. Es esta combinación (barrido del coeficiente, visualización del paisaje y eliminación de las explicaciones alternativas) la que fija el fallo de propagación como un problema de optimización inducido por la forma en que la PINN trata el tiempo.

2.5.4. Estrategias de mitigación y sus costes

Frente a estos fallos se han propuesto remedios de naturaleza distinta, y conviene presentarlos indicando en cada caso cómo se ha comprobado su eficacia y qué precio tienen, porque juntos marcan el contraste con la solución arquitectónica que adopta PINNsFormer.

La vía más directa consiste en sembrar el dominio con observaciones de la solución que anclen el entrenamiento. Su eficacia es esperable, puesto que allí donde hay datos la red se ve forzada hacia la solución correcta, pero su coste es alto en el problema directo, porque exige disponer de una solución que, por definición, es lo que se quería calcular, de modo que contradice en parte el propósito de prescindir de datos.

Una segunda vía fracciona el horizonte temporal y entrena la red por tramos sucesivos, propagando la solución de un intervalo al siguiente en lugar de imponerla de golpe sobre todo el dominio. Los esquemas de tipo *sequence-to-sequence* y de aprendizaje por *curriculum* pertenecen a esta familia. Su validación es la cara complementaria del experimento de propagación de la sección anterior. Krishnapriyan et al. (2021) mostraron que un currículo que endurece de forma gradual el coeficiente de la ecuación, y una formulación secuencia a secuencia que avanza en el tiempo por ventanas, recuperan la solución correcta justo en los regímenes donde la PINN de una sola pasada fracasaba; Wight and Zhao (2021) obtuvieron un resultado análogo en ecuaciones de fase de tipo Allen–Cahn y Cahn–Hilliard, con una estrategia adaptativa de muestreo y avance temporal. El coste es de cómputo y de gestión, pues cada tramo o cada nivel del currículo es un entrenamiento encadenado con los anteriores, se multiplica el número de ejecuciones y aparecen hiperparámetros nuevos (el tamaño de la ventana, el calendario del currículo) que hay que fijar.

Una tercera vía actúa sobre la propia función de pérdida. La ponderación basada en el NTK ya mencionada (Wang et al., 2022) deriva los pesos de los autovalores del núcleo tangente; su eficacia se apoya en el mismo análisis espectral que diagnostica el fallo, y su coste es el de estimar ese núcleo, cuyo cálculo exacto es caro y obliga a aproximarlo periódicamente durante el entrenamiento. De forma más específica para el fallo de propagación, Wang et al. (2024) introducen un peso temporal que hace que el residuo en un instante solo empiece a contar cuando el residuo en los instantes anteriores ya se ha reducido, forzando a la red a aprender en el orden en que el tiempo fluye. Lo comprobaron sobre sistemas transitorios exigentes (la ecuación de Kuramoto–Sivashinsky, un flujo de Navier–Stokes y el sistema de Lorenz), donde la PINN estándar no converge a la solución de referencia y la versión causal sí. El precio es un hiperparámetro de tolerancia que gradúa la severidad del peso y una dependencia temporal en el cálculo que resta paralelismo al entrenamiento.

Todas estas estrategias comparten el rasgo de mitigar los fallos sin cambiar la arquitectura de la red, actuando sobre los datos, el muestreo o la pérdida. Comparten también la forma de su coste (más hiperparámetros que ajustar, más cómputo, o la pérdida de la simplicidad original del método) y ninguna elimina los fallos por completo. PINNsFormer representa una respuesta de naturaleza distinta a varios

de estos problemas, en particular al de propagación temporal y al sesgo espectral. En lugar de reponderar la pérdida o trocear el dominio, sustituye el MLP por una arquitectura que incorpora la estructura temporal en su propio diseño. A explicar esa arquitectura, y a comprobar mediante reproducción independiente si la promesa se sostiene, se dedican los capítulos 3 y 4.

3

El Transformer y la arquitectura PINNsFormer

El capítulo anterior dejó las PINN vainilla en un punto incómodo, capaces de representar la solución, pero con modos de fallo que la optimización no sabe sortear, en especial el sesgo espectral y el fallo de propagación temporal. Las mitigaciones revisadas en la sección 2.5 actuaban sobre la pérdida, el muestreo o los datos, sin tocar la red. PINNsFormer (Zhao et al., 2024a) ataca el problema desde otro ángulo, el de cambiar la arquitectura. Sustituye el perceptrón multicapa por un Transformer y reformula la entrada para que la red vea, en lugar de un punto aislado, un pequeño tramo de la evolución temporal. Este capítulo explica primero el Transformer y su mecanismo de atención, justifica después por qué un MLP punto a punto es ciego a la estructura temporal, y detalla a continuación las tres piezas con las que PINNsFormer responde (la pseudo-secuencia, el esqueleto *encoder–decoder* y la activación WaveAct, una activación periódica aprendible que se detalla en la sección 3.4), ateniéndose a la implementación efectivamente empleada en este trabajo. La reproducción independiente de sus resultados sobre los problemas del artículo original, junto con la metodología y la infraestructura experimental que la sustentan, se aborda en el capítulo 4.

El Transformer, introducido por Vaswani et al. (2017) en el contexto del procesamiento del lenguaje, prescinde de la recurrencia y de la convolución y construye sus representaciones únicamente a partir del mecanismo de atención. La idea es permitir que cada elemento de una secuencia combine información de todos los demás, ponderando su relevancia de forma aprendida. Dada una secuencia de vectores agrupados en una matriz, la atención los proyecta linealmente en tres papeles (consultas $\mathbf{Q}$, claves $\mathbf{K}$ y valores $\mathbf{V}$) y calcula

$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}, \tag{3.1}$$

donde d_k es la dimensión de las claves y el factor $1/\sqrt{d_k}$ evita que los productos escalares crezcan tanto que saturen la *softmax*. La operación (3.1) admite una lectura intuitiva como una consulta a una memoria asociativa. Cada elemento de la secuencia genera tres vectores con papeles distintos: su consulta $\mathbf{q}$ expresa qué información busca, su clave $\mathbf{k}$ anuncia qué información contiene y su valor $\mathbf{v}$ es la información que aporta si alguien lo consulta. El producto $\mathbf{QK}^\top$ compara cada consulta con todas las claves, de modo que la entrada (i, j) de esa matriz mide cuánto le interesa al elemento i el contenido del elemento j; la *softmax* convierte cada fila de similitudes en una distribución de pesos que suma uno, y el producto final por $\mathbf{V}$ construye, para cada elemento, una media ponderada de los valores de toda la secuencia. El resultado es que la representación de cada elemento se actualiza mezclando la información de aquellos otros que le resultan relevantes, con pesos que no están fijados de antemano sino que dependen del contenido y se aprenden durante el entrenamiento. Esta es la diferencia esencial con una red convolucional o recurrente, ya que la atención puede conectar dos elementos cualesquiera de la secuencia en un solo paso, por alejados que estén. Para que el modelo capture distintos tipos de relación en paralelo, la atención se replica en varias cabezas, cada una con sus propias proyecciones (cada cabeza aprende así un criterio de relevancia distinto, igual que distintos filtros de una convolución aprenden rasgos distintos), y sus salidas se concatenan,

$$\begin{aligned} \mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) &= \left[\,\mathrm{head}_1 \,\|\, \cdots \,\|\, \mathrm{head}_h\,\right]\mathbf{W}^O, \\ \mathrm{head}_i &= \mathrm{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V). \end{aligned} \tag{3.2}$$

Cuando consultas, claves y valores proceden de la misma secuencia se habla de auto-atención, en la que cada elemento consulta a sus compañeros de secuencia. Cuando las consultas vienen de una secuencia y las claves y los valores de otra, se habla de atención cruzada, y es el mecanismo por el que una parte del modelo interroga a lo que otra parte ha calculado.

Sobre esta operación, la arquitectura original organiza dos módulos con funciones distintas. El *encoder* (codificador) recibe la secuencia de entrada y la transforma, mediante una pila de bloques idénticos, en una secuencia de representaciones enriquecidas donde cada elemento ya incorpora el contexto de los demás. El *decoder* (decodificador) genera la salida y, en cada uno de sus bloques, combina auto-atención sobre lo ya generado con atención cruzada sobre la salida del codificador, que actúa como la memoria que el decodificador consulta. Cada bloque de ambos módulos alterna la atención con una red *feed-forward* aplicada posición a posición (un pequeño MLP idéntico para todos los elementos, que procesa cada representación por separado), y ambas operaciones van envueltas en conexiones residuales (la entrada del bloque se suma a su salida, lo que facilita el flujo del gradiente en pilas profundas) y en una normalización por capas que estabiliza las escalas. Queda un último ingrediente. La atención (3.1) es invariante al orden de

los elementos, de modo que si se permuta la secuencia, el resultado se permuta igual, sin que el modelo note el cambio. Como en el lenguaje el orden importa, el Transformer original añade a cada elemento una codificación posicional que inyecta su posición en la secuencia. Conviene retener este detalle, porque PINNsFormer prescinde de ese término y deja que sea la propia coordenada temporal, distinta en cada elemento de su secuencia, la que aporte la información de orden. La figura 3.1 (siguiente página) resume este esqueleto *encoder–decoder* sobre el que PINNsFormer construye su propuesta.

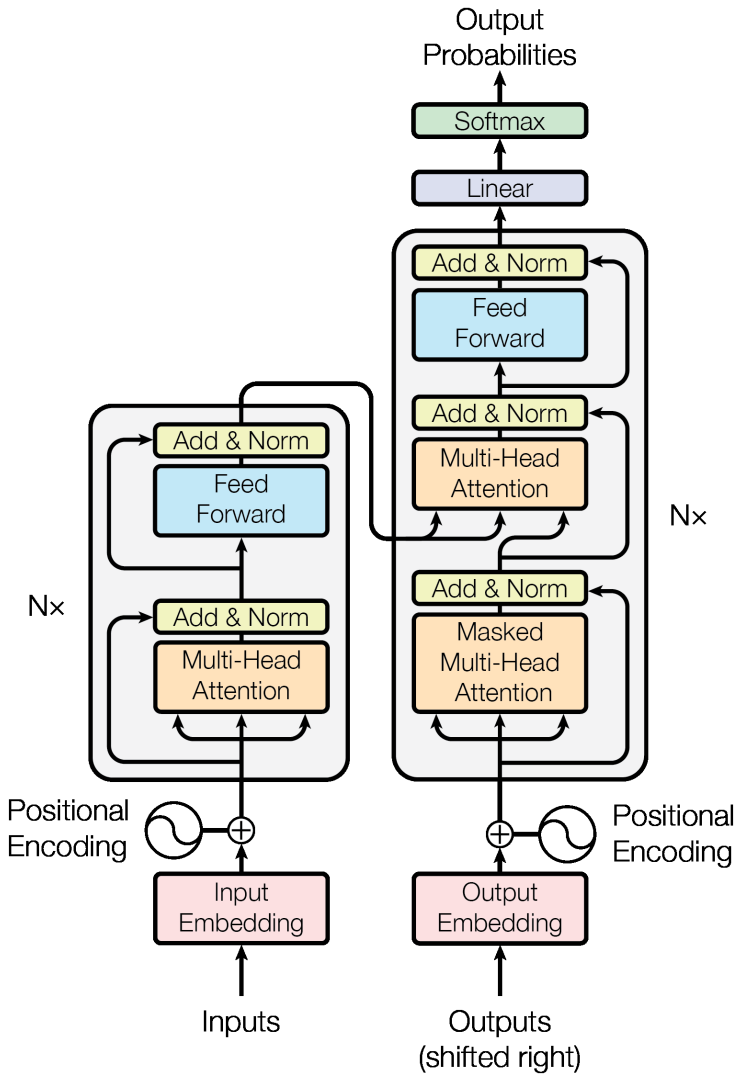


Figura 3.1: Arquitectura *encoder–decoder* del Transformer original de Vaswani et al. (2017). A la izquierda, el codificador, donde la secuencia de entrada se embebe, recibe la codificación posicional y atraviesa N bloques idénticos de auto-atención multi-cabeza seguida de red *feed-forward*, ambas con conexión residual y normalización (*Add & Norm*). A la derecha, el decodificador, cuyos bloques añaden, entre la auto-atención y la *feed-forward*, una atención cruzada cuyas claves y valores provienen de la salida del codificador. La capa lineal y la *softmax* finales producen la distribución de salida. PINNsFormer reutiliza este esqueleto sustituyendo la normalización por la activación WaveAct y omitiendo la codificación posicional.

3.1. Motivación. Las dependencias temporales que el MLP ignora

Una PINN vainilla trata cada par (x, t) como una muestra independiente. La red (2.3) recibe las coordenadas de un punto y devuelve el valor de la solución en ese punto, sin que su predicción en (x, t) guarde relación explícita con la de $(x, t + \Delta t)$. La estructura temporal del problema (el hecho de que la solución en un instante condiciona la del siguiente) no está codificada en ninguna parte de la red; queda confiada por completo a la función de pérdida, que impone la ecuación punto a punto y espera que la coherencia temporal emerja del ajuste. Este planteamiento es precisamente el que falla en los problemas transitorios estudiados por Krishnapriyan et al. (2021), pues nada en el MLP empuja a la red a propagar la información en el sentido en que fluye el tiempo, y el entrenamiento se conforma con minimizar el residuo localmente aunque la solución global sea incorrecta.

La observación de la que parte PINNsFormer es que un modelo capaz de mirar varios instantes a la vez, y de relacionarlos mediante atención, tiene al menos la posibilidad de aprender esa dependencia temporal en lugar de ignorarla. El Transformer es el candidato natural, porque su mecanismo de atención está diseñado justamente para modelar relaciones entre los elementos de una secuencia. Falta, eso sí, un detalle nada menor: en una PINN no hay secuencias, sino puntos sueltos de colocación. Construir esa secuencia a partir de cada punto es el primer paso de la arquitectura.

3.2. Pseudo-secuencia temporal

PINNsFormer convierte cada punto de colocación en una secuencia corta desplazándolo en el tiempo. A partir de (x, t) genera k copias separadas por un paso temporal Δt,

$$(x, t) \longmapsto \big\{(x, t),\ (x, t + \Delta t),\ (x, t + 2\Delta t),\ \ldots,\ (x, t + (k - 1)\Delta t)\big\}, \tag{3.3}$$

de modo que sólo se modifica la coordenada temporal y la espacial permanece fija. El artículo fija, para todos los problemas, $k = 5$ y $\Delta t = 10^{-3}, 10^{-4}$ (sin especificar exactamente cuál de los dos): cada punto se replica cinco veces y a la i-ésima copia se le suma $i\,\Delta t$ al tiempo. El resultado es lo que los autores denominan pseudo-secuencia. El prefijo merece un matiz, porque se trata de una secuencia en el sentido pleno del término, con k elementos ordenados en el tiempo sobre los que la atención puede operar relacionando el estado en t con el de los instantes inmediatamente posteriores. Lo que tiene de «pseudo» es su origen, ya que no procede de

instantáneas muestreadas del sistema ni de datos observados, sino que se construye artificialmente como un entorno temporal infinitesimal alrededor de cada punto de colocación.

La elección de un Δt muy pequeño no es casual. La pseudo-secuencia no pretende cubrir el horizonte temporal del problema, sino ofrecer a la red una vista local y densa de la evolución, lo bastante fina para que las derivadas temporales que exige el residuo de la EDP queden bien representadas y lo bastante corta para no encarecer el cómputo. La red produce una salida para cada uno de los k instantes de la secuencia; la predicción asociada al punto original es la del primer instante, y los restantes actúan como puntos de colocación adicionales que refuerzan la consistencia temporal local. De este modo, la coordenada temporal, que crece de elemento en elemento dentro de la secuencia, cumple la función que en el Transformer original desempeñaba la codificación posicional, y PINNsFormer puede por ello prescindir de ese término.

3.3. Arquitectura *encoder–decoder* de tipo Transformer

Sobre la pseudo-secuencia (3.3) opera un Transformer *encoder–decoder* de tamaño deliberadamente reducido. La descripción que sigue se corresponde con la implementación empleada en este trabajo, portada del repositorio oficial que acompaña al artículo (Zhao et al., 2024b) sin alterar la arquitectura. El flujo de datos es el siguiente. Las coordenadas de cada elemento de la secuencia (de dimensión d_{in}, igual a 2 en los problemas $1\text{D} + t$ y a 3 en Navier–Stokes) se proyectan a un espacio de trabajo de dimensión d_{model} mediante una capa lineal de *embedding*. La secuencia de vectores así obtenida atraviesa el codificador, y su salida alimenta (como claves y valores) la atención cruzada del decodificador, cuyas consultas son la propia secuencia embebida. Por último, una cabeza de salida transforma la representación del decodificador en el valor de la solución. La figura 3.2 muestra este recorrido completo, desde la pseudo-secuencia de entrada hasta la salida en los k instantes.

Cada bloque del codificador y del decodificador sigue un esquema residual con pre-activación. Si $\mathbf{z}$ es la entrada del bloque y ϕ denota la activación WaveAct de la sección siguiente, el codificador calcula

$$\begin{aligned} \mathbf{z}' &= \mathbf{z} + \mathrm{MultiHead}\big(\phi(\mathbf{z}), \phi(\mathbf{z}), \phi(\mathbf{z})\big), \\ \mathbf{z}'' &= \mathbf{z}' + \mathrm{FF}\big(\phi(\mathbf{z}')\big), \end{aligned} \tag{3.4}$$

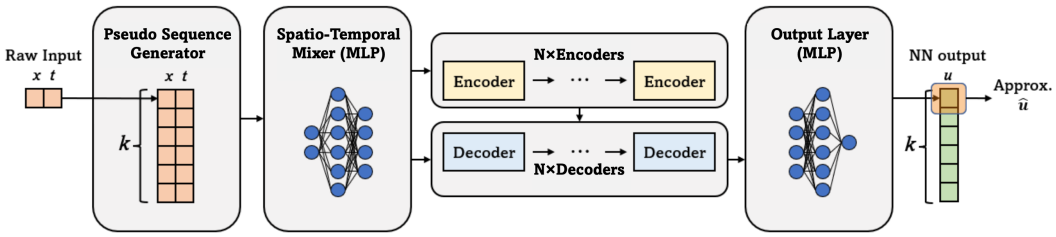


Figura 3.2: Esquema completo de PINNsFormer (Zhao et al., 2024a). Cada punto de colocación se expande en una pseudo-secuencia temporal de k instantes, se proyecta al espacio de trabajo mediante el *embedding* y atraviesa el codificador y el decodificador de atención antes de que la cabeza de salida devuelva la solución en cada instante de la secuencia.

donde FF es una red *feed-forward* de tres capas lineales $d_{\text{model}} \to d_{\text{ff}} \to d_{\text{ff}} \to d_{\text{model}}$ con WaveAct intercalada. El bloque del decodificador es idéntico salvo en que su atención es cruzada, con las consultas provenientes de la secuencia embebida y las claves y valores, de la salida del codificador. Conviene notar dos particularidades respecto al Transformer canónico. La primera es que la normalización por capas habitual se sustituye por la estructura residual con WaveAct; la segunda, ya mencionada, es la ausencia de codificación posicional. Tras la pila de bloques, el codificador y el decodificador aplican una WaveAct final, y la cabeza de salida es un perceptrón de tres capas $d_{\text{model}} \to d_{\text{hidden}} \to d_{\text{hidden}} \to d_{\text{out}}$, de nuevo con WaveAct. La figura 3.3 detalla la estructura interna de un bloque de codificador y de decodificador.

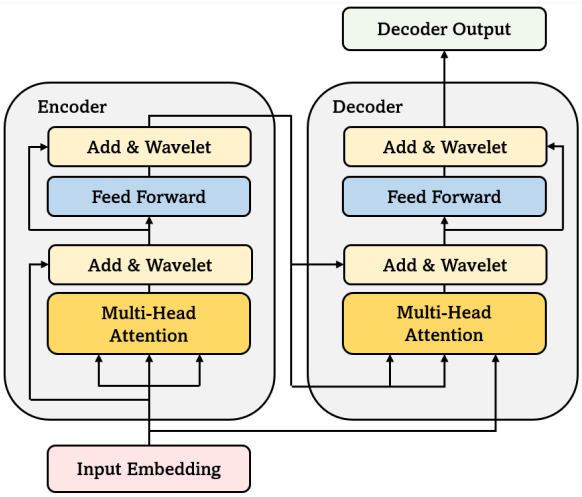


Figura 3.3: Imagen del codificador y el decodificador de PINNsFormer (Zhao et al., 2024a). Cada bloque combina atención (auto-atención en el codificador, atención cruzada en el decodificador) con una red *feed-forward*, conexiones residuales y la activación WaveAct, en lugar de la normalización por capas del Transformer canónico.

Los tamaños son pequeños para los estándares de los Transformer del lenguaje, algo razonable si se tiene en cuenta que aquí no se procesan corpus masivos de

texto, sino que se aproxima una función suave definida sobre un dominio de baja dimensión. El artículo fija $d_{\text{model}} = 32$, una sola capa en el codificador y otra en el decodificador ($N = 1$), $h = 2$ cabezas de atención, una dimensión interna de la *feed-forward* $d_{\text{ff}} = 256$ y una cabeza de salida con $d_{\text{hidden}} = 512$. Con esta configuración el modelo ronda los 4.5×10^5 parámetros, una cifra comparable a la de las PINN basadas en MLP con las que se compara, lo que hace que la comparación del capítulo 4 sea razonablemente justa en cuanto a capacidad. La dimensión d_{ff}, que en el repositorio aparece como un valor fijo, se expone aquí como un hiperparámetro más, de cara al estudio de ablación del capítulo 5.

3.4. La activación WaveAct

La última pieza, y la que conecta de forma más directa con los modos de fallo del capítulo anterior, es la función de activación. En lugar de la tangente hiperbólica habitual en las PINN, PINNsFormer emplea una activación periódica y aprendible que sus autores denominan WaveAct,

$$\phi(x) = w_1 \sin(x) + w_2 \cos(x), \tag{3.5}$$

donde w_1 y w_2 son parámetros entrenables (inicializados a la unidad) propios de cada instancia de la activación. La motivación es atacar el sesgo espectral descrito en la sección 2.5.1. Una activación construida sobre senos y cosenos introduce de partida componentes oscilatorias en la representación de la red, lo que facilita aprender soluciones con contenido de alta frecuencia que una activación suave y monótona como la tangente hiperbólica tiende a suavizar en exceso. La idea entronca con una observación recurrente en la literatura de redes neuronales para funciones, la de que dotar a la red de un sesgo hacia lo periódico u oscilatorio mitiga la lentitud con que, de otro modo, aprende las frecuencias altas (Rahaman et al., 2019).

El que w_1 y w_2 sean aprendibles permite a cada activación ajustar la amplitud relativa de sus componentes seno y coseno (y, con ello, la fase y la escala efectivas de la oscilación) en función de lo que el entrenamiento demande en ese punto de la red. WaveAct no aparece sólo en la cabeza de salida, sino de forma sistemática, dentro de la *feed-forward*, entre la atención y la *feed-forward* de cada bloque, y a la salida del codificador y del decodificador. Es, en este sentido, un rasgo transversal de la arquitectura más que un detalle local, y constituye uno de los factores cuya influencia se aísla en la ablación del capítulo 5, al contrastarla con activaciones convencionales.

4

Reproducción independiente del artículo original

Una arquitectura no queda validada por las cifras que publican sus propios autores. El capítulo anterior describió PINNsFormer tal como sus creadores la proponen y tal como se ha portado a este trabajo; queda por comprobar si las mejoras que el artículo (Zhao et al., 2024a) atribuye a la arquitectura se sostienen al reimplementar y reentrenar los modelos de forma independiente. Ese es el objeto de este capítulo. Primero se detalla la metodología de reproducción (los hiperparámetros y el protocolo exactos del artículo, las discrepancias entre el código publicado y el texto, y la infraestructura experimental sobre la que se ejecutan los experimentos). Después se presentan los resultados en precisión simple, la del código oficial, para los dos pasos temporales que el artículo deja sin precisar ($\Delta t = 10^{-3}$ y $\Delta t = 10^{-4}$), contrastados con las cifras publicadas. A continuación se analizan las discrepancias, que no son menores, apoyándose en los datos de las propias ejecuciones y en la literatura reciente que reexamina los modos de fallo de las PINN (Xu et al., 2025). Por último, y a raíz de ese análisis, se repite la reproducción en precisión doble siguiendo la tesis de Xu et al. (2025), lo que permite poner a prueba de forma directa si las diferencias observadas son de origen numérico. El resultado de fondo, adelantado aquí, es que la reproducción confirma solo parcialmente las afirmaciones del artículo, y que la explicación más parsimoniosa de las diferencias remite a la precisión numérica y al paso temporal del entrenamiento antes que a una ventaja arquitectónica.

4.1. Metodología de reproducción

Reproducir un resultado exige fijar sin ambigüedad qué se entrena, cómo se entrena y sobre qué se mide el error. Esta sección recoge esos tres elementos tal como los especifica el artículo, señala los puntos en los que el código publicado se aparta del texto (y qué decisión se ha tomado en cada uno) y describe la infraestructura

que garantiza que una misma configuración produzca el mismo número con independencia de la máquina.

4.1.1. Hiperparámetros y protocolo del artículo

La reproducción replica el protocolo descrito en la sección de experimentos del artículo y en su apéndice A. Los cuatro modelos, tanto los usados como *baselines* por el artículo original (Zhao et al. (2024a)) como PINNsFormer, se entrenan con L-BFGS y búsqueda lineal *strong Wolfe* durante 1000 iteraciones, con los tres términos de la pérdida (residuo, condición inicial y condición de contorno) ponderados por igual, $\lambda_{\text{res}} = \lambda_{\text{ic}} = \lambda_{\text{bc}} = 1$. El error se cuantifica con las métricas de error relativo ℓ_1 (rMAE) y ℓ_2 (rRMSE) definidas en la ecuación (2.8), evaluadas sobre una malla de test de 101×101 puntos común a todos los modelos.

Las arquitecturas siguen la tabla de hiperparámetros del artículo, resumida en la tabla 4.1. Un detalle deliberado del diseño experimental original es que los cuatro modelos tienen un número de parámetros del mismo orden (en torno a medio millón), de modo que ninguna eventual ventaja pueda atribuirse a una mayor sobreparametrización. La malla de puntos de colocación, en cambio, no es común, pues los *baselines* se entrenan sobre una malla de 101×101, mientras que PINNsFormer emplea una malla más gruesa de 51×51, una reducción intencionada de muestras con la que el artículo busca demostrar que la arquitectura generaliza con menos datos. Las constantes de las ecuaciones se fijan a los valores del artículo, con coeficiente de convección $\beta = 50$, de reacción $\rho = 5$, de onda $C = 4$ y, en Navier–Stokes, $\lambda_1 = 1$ y $\lambda_2 = 0.01$, con la presión evaluada en el último instante disponible.

Tabla 4.1: Hiperparámetros de los cuatro modelos, según la tabla 4 del artículo. El número de parámetros se mantiene del mismo orden a propósito. La activación de PINNsFormer es WaveAct; los *baselines* usan tangente hiperbólica (FLS sustituye la primera capa por una activación seno).

Modelo	Estructura	N.º params.
PINN	4 capas ocultas, tamaño 512	∼527 k
FLS	4 capas (1.ª seno, resto tanh), tamaño 512	∼527 k
QRes	4 capas de bloques cuadráticos, tamaño 256	∼397 k
PINNsFormer	1 enc. + 1 dec., d_{model}=32, 2 cabezas, d_{ff}=256, salida 512	∼454 k

La pseudo-secuencia de PINNsFormer usa longitud $k = 5$. Para el paso temporal Δt, el artículo indica el conjunto $\{10^{-3}, 10^{-4}\}$ sin precisar qué valor produce los resultados principales que publica. Ante esa indeterminación, la reproducción se ejecuta con ambos valores, 10^{-3} y 10^{-4}, y se contrastan por separado en la sección 4.2; anticipando el análisis, la elección de uno u otro no es inocua y llega a decidir si un problema se reproduce o no.

En cuanto a la precisión aritmética, el entrenamiento se realiza en precisión simple (coma flotante de 32 bits), que es la del código publicado y la del entorno oficial, y sobre la que se obtienen los resultados de referencia del artículo. No obstante, por motivos que el análisis de la sección 4.3 hará evidentes (y siguiendo la tesis de Xu et al. (2025) sobre el papel de la precisión en los modos de fallo), la matriz de reproducción se repitió después por completo en precisión doble (64 bits). Esa segunda reproducción, y las conclusiones que de ella se extraen, se presentan por separado en la sección 4.4.

4.1.2. Discrepancias entre el código publicado y el texto

La reproducción fiel de un artículo tropieza casi siempre con desajustes entre lo que el texto afirma y lo que el código hace. Documentarlos (en lugar de corregirlos en silencio) forma parte del rigor del ejercicio, porque cada uno es una fuente potencial de divergencia en los números. Se han identificado y resuelto los siguientes, dando prioridad en todos los casos al texto del artículo sobre el código heredado de los cuadernos.

El primero afecta al número de iteraciones. Los cuadernos de convección y reacción del repositorio entrenaban durante 500 iteraciones, mientras que el texto especifica 1000 para todos los modelos, y se adopta el valor del texto. El segundo, al tamaño de QRes: el código fijaba un tamaño oculto de 512, pero la tabla de hiperparámetros del artículo indica 256, que es el valor adoptado y el que además explica su recuento de parámetros inferior. El tercero, a la malla de entrenamiento, ya que el código aplicaba la misma malla a todos los modelos, mientras que el texto reserva la malla gruesa de 51×51 para PINNsFormer y la fina de 101×101 para los *baselines*, asimetría que aquí se respeta.

El cuarto desajuste es el coeficiente de la ecuación de onda. El artículo escribe $C = 3$ como velocidad de onda, pero la solución analítica que él mismo proporciona solo satisface la ecuación con coeficiente $C = 4$; el «3» corresponde en realidad al modo de la condición inicial. Usar 3 haría que la referencia no resolviera la propia ecuación e impediría reproducir los resultados, de modo que se usa $C = 4$, valor del código oficial y coherente con la solución publicada. El quinto es el paso temporal de Navier–Stokes, cuyo cuaderno empleaba $\Delta t = 10^{-2}$, valor que contradice el conjunto $\{10^{-3}, 10^{-4}\}$ del artículo, y se descarta en favor de 10^{-4}. El sexto, la capa de *embedding* en Navier–Stokes. El código la fija a dos entradas, pero este problema aporta tres coordenadas (x, y, t), así que la capa se generaliza a d_{in} entradas preservando exactamente el comportamiento del caso bidimensional.

Quedan dos desajustes de alcance menor. Los mejores resultados del artículo en la ecuación de onda (tabla 2) emplean una reponderación de la pérdida basada en el *neural tangent kernel* que no está implementada en el código portado; la

reproducción de onda se limita, por tanto, a las variantes sin NTK, y así se contrasta en su tabla. Y la dimensión interna $d_{\text{ff}} = 256$ del *feed-forward* de PINNsFormer no aparece en la tabla de hiperparámetros del artículo, de modo que se toma del código oficial y se anota como tal.

4.1.3. Infraestructura experimental

Los experimentos se ejecutan sobre una infraestructura reproducible y desatendida, diseñada con dos objetivos, que el coste de cómputo esté acotado y que el mismo experimento dé el mismo número en cualquier máquina. Todo el entrenamiento pasa por un único punto de entrada que fija las semillas, construye el modelo y el problema, entrena con L-BFGS, evalúa las métricas, genera las figuras de solución y vuelca un registro con la configuración completa y los resultados. Ese mismo código se ejecuta sin cambios tanto en la nube como en un cuaderno interactivo, de modo que la paridad de resultados entre entornos es verificable y no una promesa.

El camino principal de cómputo es una imagen de contenedor construida sobre la versión oficial de la biblioteca de aprendizaje profundo (2.0.1, con CUDA 11.7 y cuDNN 8), la misma del entorno con el que se obtuvieron las cifras publicadas, para minimizar las diferencias atribuibles a la versión del software. Esa imagen se lanza como un *pod* de GPU bajo demanda en un proveedor de cómputo en la nube; las ejecuciones de esta reproducción se ejecutaron sobre una GPU NVIDIA RTX A6000. El control de coste se articula por partida doble. El contenedor se autoapaga al terminar pase lo que pase, y el lanzador, además, espera a que la ejecución figure como concluida y termina el *pod* por su cuenta. Las métricas, las figuras y los registros de cada ejecución se guardan en una plataforma de seguimiento de experimentos, agrupados por marca de tiempo, de forma que ningún intento sobrescribe a otro y que los resultados persisten aunque el *pod* desaparezca; una utilidad autónoma los descarga después al repositorio local.

La reproducibilidad se apuntala fijando todas las fuentes de aleatoriedad (los generadores de la biblioteca numérica y del framework, en CPU y en GPU) y forzando el modo determinista de las rutinas de cálculo, con las versiones de las dependencias ancladas. Con estas garantías, una misma semilla debe producir el mismo número en la nube y en el cuaderno interactivo. La figura 4.1 resume la estructura de esta infraestructura de extremo a extremo.

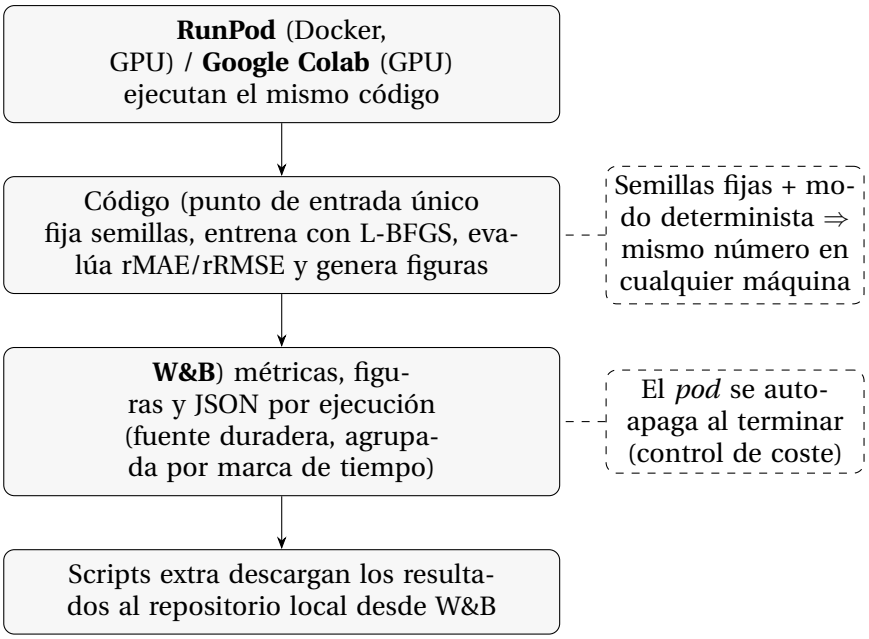


Figura 4.1: Estructura de la infraestructura MLOps de la reproducción. Toda ejecución (en la nube o en el cuaderno) pasa por el mismo punto de entrada, que entrena, evalúa y sube sus resultados a una plataforma de seguimiento duradera; una utilidad autónoma los baja al repositorio local. El determinismo garantiza la paridad de resultados entre entornos y el autoapagado del *pod* acota el coste.

4.1.4. Matriz experimental, agregación y reproducibilidad verificada

La reproducción no se limita a una sola configuración. Combinando las dos precisiones aritméticas y los dos pasos temporales que el artículo deja abiertos, la matriz de experimentos abarca tres configuraciones sobre los cuatro problemas y los cuatro modelos: precisión simple con $\Delta t = 10^{-3}$, precisión simple con $\Delta t = 10^{-4}$ y precisión doble con $\Delta t = 10^{-4}$. Las dos primeras se analizan en la sección 4.2 y la tercera en la sección 4.4; en conjunto suman varias decenas de ejecuciones, todas con la semilla 0.

Para no fiar los resultados a una sola ejecución, cada configuración se lanzó de forma independiente más de una vez, en *pods* distintos y en momentos distintos, con la intención de promediar las repeticiones y amortiguar cualquier variabilidad residual del entrenamiento. En la práctica, las repeticiones de una misma configuración devolvieron métricas byte a byte idénticas en todos los problemas y modelos, hasta la última cifra significativa. Las tablas recogen, por tanto, un valor que coincide exactamente con cada repetición, y esa coincidencia es en sí un resultado, pues confirma que el determinismo del protocolo funciona (semillas fijas, modo determinista de las rutinas de cálculo y dependencias ancladas) y que las cifras no son producto de una realización afortunada del azar del entrenamiento, sino la

salida estable de una configuración fija. Esta propiedad importa de modo particular para el argumento del capítulo. Si una misma configuración es perfectamente reproducible, cualquier diferencia entre configuraciones (entre un paso temporal y otro, o entre una precisión y otra) es atribuible a ese factor y no a la aleatoriedad.

4.2. Resultados de la reproducción en precisión simple

Esta sección recoge los resultados obtenidos en precisión simple, la del código oficial y la de las cifras de referencia del artículo. Como el texto no precisa qué paso temporal de la pseudo-secuencia produce sus resultados principales, se reproduce con los dos valores que menciona ($\Delta t = 10^{-3}$ y $\Delta t = 10^{-4}$), y cada uno se contrasta por separado con las mismas cifras publicadas. Conviene leer las tablas recordando que un error relativo próximo a la unidad indica que el modelo no ha aprendido la solución (se queda en una predicción trivial), mientras que un valor de orden 10^{-2} o inferior corresponde a una solución correcta. En negrita, el mejor modelo de cada columna en la reproducción.

4.2.1. Paso temporal $\Delta t = 10^{-3}$

Con el paso grueso, PINNsFormer reproduce sus dos resultados unidimensionales más llamativos. En convección alcanza un rMAE de 0.030, prácticamente el 0.023 publicado, y en 1D-reacción un 0.011 que iguala o mejora el 0.015 del artículo (tabla 4.2). La ecuación de onda, en cambio, se le resiste ya en esta configuración, pues obtiene 0.492, lejos del 0.270 publicado y peor que cualquiera de los *baselines* (tabla 4.3). En Navier–Stokes (tabla 4.4) ningún modelo recupera la presión con error relativo menor que uno (tampoco PINNsFormer, cuya casilla se completa aquí con un rMAE de 16.3 frente al 0.384 publicado).

Tabla 4.2: Convección y 1D-reacción con $\Delta t = 10^{-3}$ (precisión simple). Cifras publicadas (tabla 1 del artículo) frente a la reproducción. Errores relativos rMAE y rRMSE.

	Convección				1D-reacción			
	Artículo		Reproducción		Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE	rMAE	rRMSE	rMAE	rRMSE
PINN	0.778	0.840	0.655	0.735	0.982	0.981	0.043	0.079
QRes	0.746	0.816	0.750	0.820	0.979	0.977	0.993	0.990
FLS	0.674	0.771	0.064	0.074	0.984	0.985	0.982	0.981
PINNsFormer	0.023	0.027	**0.030**	**0.035**	0.015	0.030	**0.011**	**0.023**

Tabla 4.3: 1D-onda con $\Delta t = 10^{-3}$ (precisión simple). Cifras publicadas (tabla 2 del artículo, filas sin NTK) frente a la reproducción. Las variantes con NTK del artículo no se reproducen (no implementadas) y se omiten de la comparación.

	Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE
PINN	0.326	0.335	**0.057**	**0.058**
QRes	—	—	0.390	0.392
FLS	—	—	0.067	0.072
PINNsFormer	0.270	0.283	0.492	0.481

Tabla 4.4: Navier–Stokes con $\Delta t = 10^{-3}$ (precisión simple). Error relativo de la presión (evaluada en el último instante) publicado (tabla 3 del artículo) frente a la reproducción.

	Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE
PINN	13.08	9.08	9.66	6.61
QRes	6.41	4.45	42.07	28.77
FLS	3.98	2.77	20.45	13.99
PINNsFormer	0.384	0.280	16.32	11.16

Un matiz merece subrayarse ya aquí. En la reacción, los *baselines* PINN y (en convección) FLS no fracasan como el artículo asegura. La PINN vainilla resuelve la reacción con un rMAE de 0.043 donde la tabla publicada le asigna 0.982, y FLS resuelve la convección con 0.064 donde el artículo le atribuye 0.674. La diferencia dramática entre un *baseline* que fracasa y un PINNsFormer que triunfa se atenúa, por tanto, incluso en la configuración que mejor reproduce a PINNsFormer. Que FLS resuelva la convección con tan poco esfuerzo tiene, además, una explicación conocida, y es que sustituye la primera capa por una activación seno, que introduce de partida las características de Fourier necesarias para representar una solución de frente viajero y contenido oscilatorio (velocidad $\beta = 50$), precisamente el tipo de función que una activación tangente hiperbólica aprende con lentitud por el sesgo espectral descrito en la sección 2.5.1.

4.2.2. Paso temporal $\Delta t = 10^{-4}$

Las tablas de esta configuración recogen únicamente a PINNsFormer, el único modelo cuyo entrenamiento depende del paso temporal. Como se justifica en la sección 4.2.3, Δt es el paso de la pseudo-secuencia y no interviene en los *baselines*, cuyas cifras (idénticas con cualquier paso) ya figuran en las tablas de la sección 4.2.1. Basta reducir el paso a 10^{-4}, sin tocar ningún otro hiperparámetro, para

que PINNsFormer se derrumbe en convección, donde su rMAE pasa de 0.030 a 0.751 (tabla 4.5), de modo que el modelo estrella queda tan estancado como los peores *baselines* del paso grueso. La reacción, en cambio, aguanta (0.011) y la onda sigue fallando (0.511). En Navier–Stokes (tabla 4.7) PINNsFormer tampoco recupera la presión, con un rMAE de 16.58 prácticamente igual al que obtiene con $\Delta t = 10^{-3}$ (16.32, tabla 4.4).

Tabla 4.5: Convección y 1D-reacción con $\Delta t = 10^{-4}$ (precisión simple). Cifras publicadas (tabla 1 del artículo) frente a la reproducción. Solo se incluye PINNsFormer: los *baselines* no dependen de Δt y sus cifras figuran en la tabla 4.2.

	Convección				1D-reacción			
	Artículo		Reproducción		Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE	rMAE	rRMSE	rMAE	rRMSE
PINNsFormer	0.023	0.027	0.751	0.818	0.015	0.030	0.011	0.022

Tabla 4.6: 1D-onda con $\Delta t = 10^{-4}$ (precisión simple). Cifras publicadas (tabla 2 del artículo, filas sin NTK) frente a la reproducción. Solo PINNsFormer; los *baselines* no dependen de Δt (tabla 4.3).

	Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE
PINNsFormer	0.270	0.283	0.511	0.502

Tabla 4.7: Navier–Stokes con $\Delta t = 10^{-4}$ (precisión simple). Error relativo de la presión (evaluada en el último instante) publicado (tabla 3 del artículo) frente a la reproducción. Solo PINNsFormer; los *baselines* no dependen de Δt (tabla 4.4).

	Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE
PINNsFormer	0.384	0.280	16.58	11.34

El contraste entre las dos configuraciones es el hallazgo que más dice de toda la reproducción en precisión simple, y merece enunciarse sin rodeos. El resultado estrella de PINNsFormer en convección (un rMAE de 0.023 que el artículo exhibe como prueba de la superioridad de la arquitectura) se reproduce con $\Delta t = 10^{-3}$ (0.030) y se desploma con $\Delta t = 10^{-4}$ (0.751), sin que cambie nada más que el paso temporal auxiliar de la pseudo-secuencia, una cantidad sin significado físico directo. Que el mismo modelo, con los mismos pesos iniciales y el mismo número de iteraciones, pase del éxito al fracaso al mover un solo hiperparámetro secundario indica que lo que se está midiendo deja de ser una propiedad estable de la arquitectura para ser la posición de cada entrenamiento respecto a una frontera inestable. La reacción, más benigna, se reproduce en ambos pasos; la onda no se

reproduce en ninguno; y Navier–Stokes queda fuera del alcance de todos los modelos (también de PINNsFormer) en cuanto a la presión, mientras que los campos de velocidad sí se recuperan. Sobre por qué el paso temporal, y no la arquitectura, decide el desenlace en convección versa la sección siguiente.

4.2.3. Los dos pasos temporales, frente a frente

Reunir en una sola tabla el efecto del paso temporal sobre PINNsFormer permite leer de un vistazo su sensibilidad a Δt (tabla 4.8). La comparación se restringe a esta arquitectura de forma deliberada, porque Δt es el paso de la pseudo-secuencia y, en esta implementación, solo interviene en el entrenamiento de PINNsFormer; los *baselines* (PINN, QRes y FLS) no construyen pseudo-secuencia y no usan Δt en ningún punto, de modo que sus cifras son idénticas con cualquier paso y ya figuran en las tablas 4.2, 4.3 y 4.4. Contrastar el paso temporal solo tiene sentido, por tanto, para PINNsFormer.

Tabla 4.8: Sensibilidad de PINNsFormer al paso temporal en precisión simple. Error relativo (semilla 0) con $\Delta t = 10^{-3}$ y $\Delta t = 10^{-4}$ sobre los tres problemas unidimensionales y sobre la presión de Navier–Stokes. Para los problemas 1D es rMAE; para Navier–Stokes es el rMAE de la presión. Solo se incluye PINNsFormer porque es el único modelo cuyo entrenamiento depende de Δt.

	Convección		1D-reacción		1D-onda		Navier–Stokes	
Modelo	10^{-3}	10^{-4}	10^{-3}	10^{-4}	10^{-3}	10^{-4}	10^{-3}	10^{-4}
PINNsFormer	0.030	0.751	0.011	0.011	0.492	0.511	16.32	16.58

El salto que domina la tabla es el de PINNsFormer en convección, que pasa del acierto al fracaso ($0.030 \rightarrow 0.751$) sin más cambio que Δt. Como es el único modelo que emplea el paso temporal, el efecto queda perfectamente aislado, sin ninguna otra variable en juego que pueda explicarlo. Que el mismo modelo, con los mismos pesos iniciales y el mismo número de iteraciones, cruce del éxito al fracaso al mover un hiperparámetro auxiliar sin significado físico directo indica que lo que se mide no es una propiedad estable de la arquitectura, sino la posición de cada entrenamiento respecto a una frontera inestable (el argumento que desarrolla la sección 4.3). La reacción, por contraste, es indiferente al paso (0.011 con ambos), y la onda falla en los dos casos (0.492 y 0.511). La columna de Navier–Stokes refuerza la lectura desde su métrica particular. La presión no baja de la unidad con ningún paso, y PINNsFormer apenas se mueve entre 16.32 y 16.58, una variación mínima que confirma que, salvo en convección, el paso temporal no altera el desenlace de este modelo.

4.3. Análisis de las discrepancias

Las diferencias entre las tablas publicadas y las reproducidas son demasiado grandes y demasiado sistemáticas para despacharlas como ruido de entrenamiento. Que PINNsFormer pase de un rMAE de 0.030 a uno de 0.751 en convección al mover únicamente el paso temporal de la pseudo-secuencia de 10^{-3} a 10^{-4}, o que la PINN vainilla resuelva con holgura la reacción que el artículo declara irresoluble para ella, no son fluctuaciones: son cambios de régimen. Esta sección argumenta, a partir de los datos de las propias ejecuciones y de la literatura reciente (Xu et al., 2025; Krishnapriyan et al., 2021), que esos cambios de régimen tienen una causa identificable en la mecánica de la optimización (sensible al paso temporal y, como se verá en la sección 4.4, a la precisión numérica), y no en la arquitectura, y que ello obliga a matizar qué es exactamente lo que las tablas del artículo miden.

4.3.1. La firma del fallo, residuo convergido con solución incorrecta

El primer indicio lo dan las componentes de la pérdida al final del entrenamiento, en la configuración de precisión simple con $\Delta t = 10^{-4}$ donde PINNsFormer fracasa en convección y onda. En esos casos, el residuo de la ecuación (el término que impone la física) está de hecho satisfecho, y lo que queda alto es el ajuste de las condiciones de contorno o inicial. En convección, la pérdida total de PINNsFormer al terminar es de 1.5×10^{-2}, pero descompuesta revela un residuo de apenas 2.2×10^{-4} frente a un término de contorno de 1.4×10^{-2}, señal de que la red ha aprendido a satisfacer la ecuación diferencial, pero no a respetar los datos de contorno, y su error de solución es del $75\,\%$. En onda ocurre lo mismo con la condición inicial, con un residuo de 6×10^{-4} frente a un término inicial de 8.5×10^{-2} que domina la pérdida y deja la solución a un error del $51\,\%$.

Esta combinación (residuo convergido, error de solución grande) es precisamente la firma que caracteriza los modos de fallo de las PINN descritos por Krishnapriyan et al. (2021) y que motivan la existencia misma de PINNsFormer. La observación relevante es que, en la reproducción, esa firma no aparece en la PINN vainilla (que resuelve la reacción) sino en PINNsFormer (que fracasa en convección y onda). El modo de fallo no ha desaparecido, se ha desplazado de un modelo a otro. Cualquier explicación de las discrepancias tiene que dar cuenta de esa movilidad.

4.3.2. La precisión numérica como causa. La tesis del FP64

Una línea de trabajo reciente ofrece exactamente esa explicación. Xu et al. (2025) sostienen que los modos de fallo canónicos de las PINN sobre estos mismos problemas de transporte no son mínimos locales insalvables separados de la solución por barreras de la función de pérdida, sino estancamientos inducidos por la precisión. Su argumento es que, con precisión simple (32 bits), el criterio de convergencia y la búsqueda lineal del optimizador L-BFGS se satisfacen prematuramente, porque el gradiente en esa precisión finita se vuelve indistinguible de cero antes de que el modelo alcance la solución; el entrenamiento se congela entonces en una «fase de fallo» en la que el residuo ya ha convergido pero el error de solución sigue siendo grande. Los autores describen una dinámica de tres fases (no convergido, fallo, éxito) cuyas fronteras se desplazan con la precisión aritmética, y muestran que basta con pasar a precisión doble (64 bits) para que las PINN vainilla crucen a la fase de éxito y resuelvan los problemas sin ningún modo de fallo.

Este marco encaja con cada anomalía de la reproducción. La firma descrita en la subsección anterior (residuo convergido, solución incorrecta) es literalmente la definición de la «fase de fallo» de Xu et al. (2025); los estancamientos de PINNsFormer en convección y onda dejan de parecer mínimos genuinos del objetivo completo y se revelan como paradas del optimizador. Y el dato determinante es que toda la reproducción, como aparentemente el código oficial del artículo, se ejecuta en precisión simple, y se sitúa por ello exactamente en el régimen en el que, según esta tesis, la frontera entre el fallo y el éxito es frágil y estos efectos dominan.

4.3.3. Fragilidad de la frontera y el papel de la semilla única

Si la frontera entre la fase de fallo y la de éxito es fina y sensible a la trayectoria exacta del optimizador, entonces qué modelo cae de qué lado depende de detalles que nada tienen que ver con la arquitectura, como la implementación concreta de la búsqueda lineal, el orden de reducción de las operaciones en una GPU dada, la versión del *framework* o la semilla. Esto explica las dos anomalías simétricas de la reproducción sin necesidad de postular nada más. Que la PINN vainilla resuelva la reacción (0.043) donde el artículo dice que fracasa (0.982) significa que la ejecución de la reproducción cayó del lado del éxito de una frontera que la del artículo no cruzó; que PINNsFormer fracase en convección y onda (0.75 y 0.51) donde el artículo dice que triunfa (0.023 y 0.27) es el fenómeno inverso. Ninguno de los dos resultados es erróneo como tal, ya que ambos son ejecuciones en precisión simple que muestrean lados distintos de una frontera inestable.

La reproducción aporta ella misma la evidencia más directa de esta fragilidad. El paso de la convección de PINNsFormer de un rMAE de 0.030 con $\Delta t = 10^{-3}$ a uno de 0.751 con $\Delta t = 10^{-4}$ (documentado en las tablas 4.2 y 4.5) es exactamente eso, dos entrenamientos idénticos salvo en un paso temporal auxiliar que caen a lados opuestos de la frontera fallo/éxito. Y como cada configuración es perfectamente reproducible (subsección 4.1.4), la diferencia no puede atribuirse al azar, sino que es el propio Δt el que empuja al optimizador de un lado al otro.

No hace falta salir del propio artículo para corroborar esta lectura. Su estudio de sensibilidad a los hiperparámetros de la pseudo-secuencia muestra que, en la 1D-reacción, mantener la longitud $k = 3$ y mover el paso temporal de 10^{-3} a 10^{-4} hace que PINNsFormer salte de un error de 0.037 a uno de 0.997 (de éxito rotundo a fallo total) por un cambio mínimo de un hiperparámetro. El propio artículo documenta, pues, que su modelo se asienta justo sobre esa frontera inestable, aunque lo presente como una virtud (flexibilidad ante un abanico amplio de hiperparámetros). Interpretado a la luz de Xu et al. (2025), ese mismo dato es la evidencia de que lo que separa el éxito del fracaso en estos problemas depende menos de la arquitectura que de qué lado de la frontera (de paso temporal y de precisión) cae cada entrenamiento.

Importa acotar bien el alcance de esta observación. Tanto el artículo como esta reproducción reportan resultados de una sola semilla por modelo. Con una frontera fallo/éxito inestable, una única realización es insuficiente para establecer una jerarquía robusta entre modelos, porque la cifra publicada puede reflejar un sorteo favorable. No se afirma aquí que haya habido una selección deliberada de resultados; se constata una limitación metodológica (la que comparten, por lo demás, el artículo y esta reproducción) que la evidencia interna de la tabla de sensibilidad vuelve especialmente pertinente.

4.3.4. Implicaciones para la comparación arquitectónica

De lo anterior se sigue una conclusión mesurada pero de peso. La diferencia que el artículo publica entre PINNsFormer y los *baselines* sobre convección y onda es, en buena parte, la diferencia entre un entrenamiento que cruzó a la fase de éxito y otros que no lo hicieron; y ese cruce, en precisión simple, depende de factores ajenos a la arquitectura. La reproducción, por tanto, no puede confirmar que la ventaja de PINNsFormer en esos problemas sea de naturaleza arquitectónica y no un artefacto de qué lado de la frontera de precisión cayó cada ejecución. Lo que sí sostiene la reproducción es más modesto. Sobre la 1D-reacción, y con la salvedad de que la ventaja frente a una PINN bien entrenada es pequeña, PINNsFormer alcanza una solución correcta; sobre convección y onda, con este protocolo y en

precisión simple, no mejora (y a menudo empeora) lo que consigue una variante tan simple como FLS.

Esto no refuta el valor de PINNsFormer, pero reordena las preguntas. La cuestión deja de ser «¿cuánto mejora PINNsFormer a la PINN?» y pasa a ser «¿qué parte de la mejora publicada sobrevive cuando se controla la precisión numérica y se promedia sobre varias semillas?». La primera mitad de esa pregunta admite una respuesta experimental directa, y es la que aborda la sección siguiente. Si las discrepancias son estancamientos inducidos por la precisión simple, reentrenar la matriz completa en precisión doble debería hacer que los fracasos que dependen de la precisión (y solo esos) se conviertan en aciertos. El promediado sobre múltiples semillas, en cambio, queda como continuación natural en conexión con el estudio de ablación del capítulo siguiente.

4.4. Segunda reproducción en precisión doble (FP64)

El análisis anterior deja una predicción falsable. Si las discrepancias en convección y onda son estancamientos del optimizador inducidos por la precisión simple, y no limitaciones de las arquitecturas, entonces reentrenar en precisión doble debería convertir en aciertos precisamente esos fracasos (y solo esos, dejando intactos los que respondan a causas distintas). Motivada por la tesis de Xu et al. (2025) y por los propios resultados en precisión simple, se repitió la matriz completa de reproducción en precisión doble (coma flotante de 64 bits). Se fija el paso temporal en $\Delta t = 10^{-4}$, que es el valor para el que se dispone de la cobertura más completa de ejecuciones en doble precisión y, no por casualidad, el que en precisión simple provocaba el fracaso de la convección, por ser ahí donde la prueba es más exigente.

Tabla 4.9: Convección y 1D-reacción en precisión doble ($\Delta t = 10^{-4}$): cifras publicadas (tabla 1 del artículo) frente a la reproducción en FP64. Compárese la fila de PINNsFormer en convección con la de la tabla 4.5 (misma configuración en FP32).

	Convección				1D-reacción			
	Artículo		Reproducción		Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE	rMAE	rRMSE	rMAE	rRMSE
PINN	0.778	0.840	0.713	0.786	0.982	0.981	0.035	0.062
QRes	0.746	0.816	0.682	0.761	0.979	0.977	0.988	0.988
FLS	0.674	0.771	0.135	0.167	0.984	0.985	0.036	0.062
PINNsFormer	0.023	0.027	**0.022**	**0.026**	0.015	0.030	**0.013**	**0.026**

La predicción se cumple, y lo hace sin ambigüedad. El caso que mejor lo ilustra es la convección de PINNsFormer (tabla 4.9). El mismo problema, con el mismo

Tabla 4.10: 1D-onda en precisión doble ($\Delta t = 10^{-4}$). Cifras publicadas (tabla 2 del artículo, filas sin NTK) frente a la reproducción en FP64.

	Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE
PINN	0.326	0.335	**0.011**	**0.011**
QRes	—	—	0.346	0.357
FLS	—	—	0.030	0.031
PINNsFormer	0.270	0.283	0.229	0.241

Tabla 4.11: Navier–Stokes en precisión doble ($\Delta t = 10^{-4}$). Error relativo de la presión publicado (tabla 3 del artículo) frente a la reproducción en FP64. PINNsFormer es el de menor error de los cuatro modelos.

	Artículo		Reproducción	
Modelo	rMAE	rRMSE	rMAE	rRMSE
PINN	13.08	9.08	17.56	12.01
QRes	6.41	4.45	38.01	26.00
FLS	3.98	2.77	14.33	9.80
PINNsFormer	0.384	0.280	**7.21**	**4.93**

$\Delta t = 10^{-4}$ que en precisión simple daba un rMAE de 0.751, pasa en precisión doble a 0.022 (esto es, recupera con exactitud la cifra publicada de 0.023). El único cambio entre un entrenamiento y otro es el número de bits de la aritmética. Es difícil imaginar una demostración más directa de que aquel fracaso no era un defecto de la arquitectura ni un mínimo insalvable de la pérdida, sino un estancamiento de L-BFGS inducido por la precisión, tal como sostiene la tesis del FP64; la figura 4.2 lo hace visible sobre el propio campo de solución.

La ecuación de onda cuenta una historia paralela, con PINNsFormer mejorando de 0.511 a 0.229, un valor ya comparable (incluso algo mejor) al 0.270 que publica el artículo, y, como se detalla más abajo, los *baselines* bien optimizados bajan también a errores pequeños en doble precisión.

Ahora bien, la doble precisión no borra todas las diferencias, y las que deja en pie son tan informativas como las que corrige. QRes sigue sin resolver la reacción (0.988 en doble precisión, como ya ocurría en simple, tabla 4.2), de modo que su fracaso no depende de la aritmética y apunta a una limitación genuina de esa variante en ese problema. En Navier–Stokes, el error relativo de la presión permanece por encima de la unidad para todos los modelos (también para PINNsFormer, que en doble precisión baja hasta 7.21), porque su origen no es la precisión sino la indeterminación de la presión salvo una constante aditiva, que ninguna mejora numérica puede eliminar; con todo, PINNsFormer es ahí el de menor error de los cuatro, netamente por debajo de unos *baselines* que se sitúan entre 14 y 38, aunque

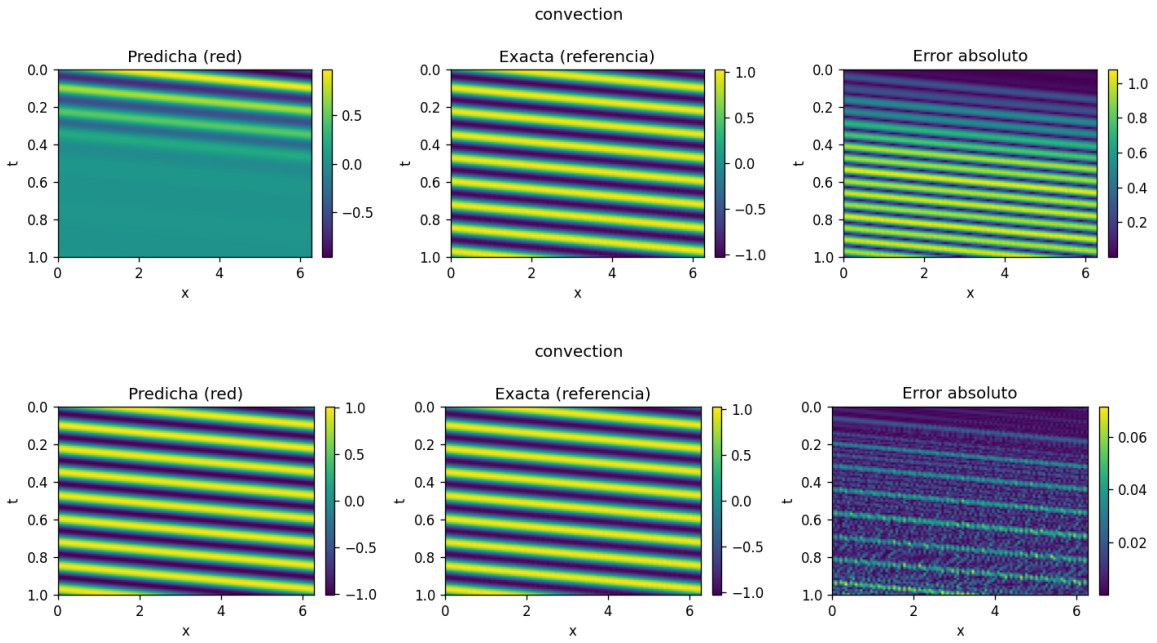


Figura 4.2: PINNsFormer en convección con $\Delta t = 10^{-4}$. Solución predicha, solución exacta y error absoluto. Arriba, en precisión simple (rMAE 0.751), donde la predicción no reconstruye el frente de transporte y el error satura el dominio. Abajo, en precisión doble (rMAE 0.022), la misma configuración, sin más diferencia que los bits de la aritmética, recupera la solución del artículo.

ninguno se acerque al 0.384 publicado. Y, sobre todo, cuando la precisión deja de penalizar a unos y a otros, los *baselines* bien optimizados se revelan competitivos. FLS resuelve la convección con 0.135 y la onda con 0.030, y la PINN vainilla resuelve la onda con 0.011, mejor que PINNsFormer. La lectura combinada es, por tanto, matizada. La precisión doble confirma que PINNsFormer puede alcanzar sus cifras publicadas (la arquitectura no está rota), pero al nivelar el terreno también muestra que su ventaja sobre alternativas mucho más simples es, en estos benchmarks, estrecha y dependiente del problema. La superioridad tajante que dibuja el artículo se sostiene sobre una comparación en precisión simple que penalizaba de forma desigual a unos modelos frente a otros.

4.5. Síntesis

La reproducción independiente del artículo deja tres conclusiones firmes. La primera es de método, ya que el protocolo resulta reproducible en sentido estricto, hasta el punto de que las repeticiones independientes de cada configuración arrojaron métricas idénticas, lo que convierte cualquier diferencia entre configuraciones en una señal atribuible a un factor concreto y no al azar. La segunda es de resultado, y es que las cifras de referencia de PINNsFormer no se reproducen de forma

incondicional, sino solo bajo condiciones precisas de entrenamiento. Su resultado en convección aparece con $\Delta t = 10^{-3}$ y en precisión doble, pero se desploma con $\Delta t = 10^{-4}$ en precisión simple; la reacción se reproduce de forma robusta; la onda no se reproduce sin la reponderación por NTK que el código no implementa. Además, en todos los casos el contraste con los *baselines* se atenúa porque estos no fracasan como el artículo afirma, y cuando el terreno se nivela (en precisión doble) variantes tan simples como FLS o la PINN vainilla resultan competitivas o superiores. La tercera es de interpretación, y ha dejado de ser una mera hipótesis, porque el reentrenamiento en precisión doble confirma de forma directa que los fracasos de convección y onda en precisión simple eran estancamientos inducidos por la precisión (la convección de PINNsFormer pasa de 0.751 a 0.022 sin más cambio que la aritmética), en línea exacta con la tesis del FP64 (Xu et al., 2025). Se sigue de todo ello que una comparación en precisión simple y con una sola semilla no aísla limpiamente una ventaja arquitectónica. Esta lectura no cierra el estudio de PINNsFormer, sino que fija la agenda del resto de la memoria, pues el capítulo siguiente lleva la arquitectura al límite mediante un estudio de ablación que, entre otras cosas, permite someter a prueba precisamente qué componentes (y qué condiciones de entrenamiento) explican su comportamiento.

A la vista del conjunto de resultados, prácticamente la totalidad de las cifras publicadas se ha logrado reproducir bajo unas condiciones u otras de precisión y paso temporal, con una sola excepción persistente, la ecuación de Navier–Stokes, donde el error de presión de PINNsFormer no se acerca al valor de referencia (0.384) ni en precisión simple ni doble, ni con $\Delta t = 10^{-3}$ ni con $\Delta t = 10^{-4}$. Como último recurso, y contradiciendo de forma deliberada los hiperparámetros que el artículo declara (el conjunto $\{10^{-3}, 10^{-4}\}$ de su tabla 4), se lanzaron dos ejecuciones adicionales con $\Delta t = 10^{-2}$, que era el valor que figuraba en el repositorio oficial en el momento de descargarlo. La tabla 4.12 recoge el resultado.

Tabla 4.12: Navier–Stokes con $\Delta t = 10^{-2}$ (el valor del repositorio oficial, ajeno al conjunto declarado en el artículo). Error relativo de la presión de PINNsFormer publicado (tabla 3 del artículo) frente a la reproducción, en precisión simple y doble. Las cifras coinciden en la práctica con las de $\Delta t = 10^{-4}$ (tablas 4.7 y 4.11), de modo que el paso temporal no altera el resultado.

	Artículo		Reproducción	
Precisión	rMAE	rRMSE	rMAE	rRMSE
Simple (FP32)	0.384	0.280	16.58	11.34
Doble (FP64)	—	—	7.21	4.93

El error de presión permanece esencialmente invariable respecto al obtenido con $\Delta t = 10^{-4}$ (16.58 en simple, 7.21 en doble), de modo que ni siquiera el paso temporal del propio código oficial recupera la cifra publicada. Esto descarta que el paso

temporal sea la causa de la discrepancia y refuerza que el origen está en las contradicciones internas entre el texto y el código de este benchmark (instante de evaluación de la presión y forma del término viscoso), ya señaladas en la sección 4.1.2. Navier–Stokes queda, por tanto, como el único caso que resiste la reproducción bajo todas las condiciones ensayadas.

5

Llevando PINNsFormer al límite. Modos de fallo y estudio de ablación

La reproducción del capítulo 4 dejó abierta una pregunta que este capítulo se propone responder de forma ordenada. Si las cifras publicadas de PINNsFormer solo se recuperan bajo condiciones precisas de entrenamiento (un paso temporal concreto, o la precisión doble), y si su ventaja sobre alternativas simples se desdibuja en cuanto se controlan esos factores y la semilla, entonces vale la pena entender qué promete exactamente la arquitectura, frente a qué fallos se propone como remedio y qué dice la literatura más reciente sobre si ese remedio se sostiene. La sección 5.1 construye ese recorrido. Parte de los modos de fallo documentados de la PINN estándar, sigue con la respuesta que PINNsFormer ofrece a esos fallos y su recepción académica, y termina con los trabajos que, ya publicado el modelo, han puesto en duda tanto el diagnóstico como la solución. Sobre ese marco se apoya el estudio de ablación propio (secciones 5.2–5.5), que somete cada pieza de la arquitectura a variación controlada para separar lo que aporta estructura de lo que aporta el simple aumento de capacidad o un artefacto de precisión.

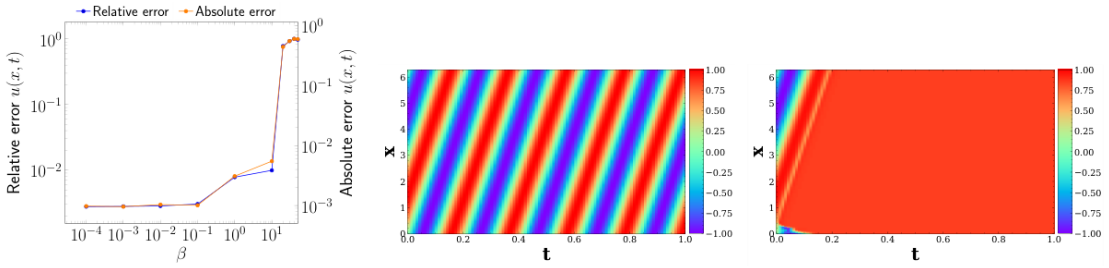
5.1. Revisión de la literatura reciente sobre modos de fallo

La historia de PINNsFormer no se entiende sin la historia de los fracasos que vino a resolver. Por eso esta revisión no se limita a enumerar trabajos, sino que sigue un hilo conductor que va desde por qué falla la PINN estándar, pasando por la mejora que PINNsFormer promete sobre esos fallos, hasta los reveses documentados de la propia arquitectura una vez sometida a escrutinio independiente. Al lector le debe quedar claro, al terminar la sección, que el ecosistema de estas arquitecturas es menos un catálogo de victorias sucesivas que un debate abierto sobre qué es realmente un modo de fallo y cuándo una arquitectura lo resuelve.

5.1.1. Los modos de fallo de la PINN vainilla

Los tres modos de fallo de la PINN estándar (el paisaje de optimización rugoso, el desequilibrio entre los términos de la pérdida y el sesgo espectral) quedaron caracterizados, con sus mecanismos y su evidencia experimental, en la sección 2.5. No se repite aquí ese análisis. Lo que interesa a esta revisión es otra cosa, releerlos como el punto de partida del que PINNsFormer deriva su razón de ser y contra el que habrá que medir, en las secciones siguientes, tanto sus promesas como los resultados propios de este trabajo.

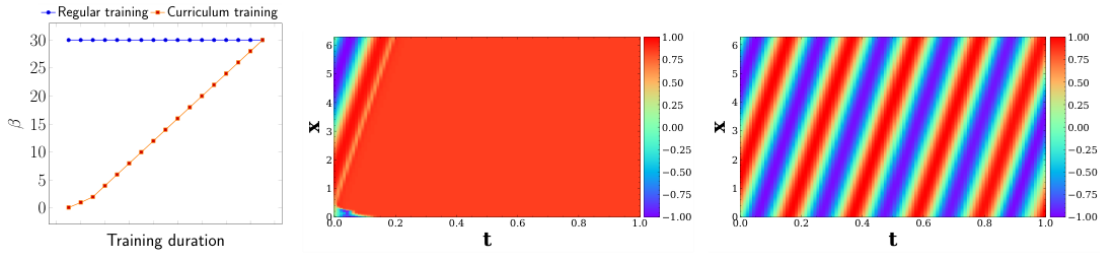
De los tres, el que fija el terreno de juego es el fallo de propagación temporal de Krishnapriyan et al. (2021), y no solo por su contenido. Las ecuaciones de convección y de reacción con las que esos autores barrieron el coeficiente de la EDP son, literalmente, dos de los benchmarks sobre los que el artículo de PINNsFormer se valida y sobre los que operan la reproducción del capítulo 4 y la ablación de este capítulo. La figura 5.1 muestra el comportamiento que convirtió esos problemas en referencia. A partir de cierto umbral del coeficiente, el error respecto a la solución exacta se dispara y la predicción deja de reconstruir el frente viajero, aunque la pérdida permanezca baja.



(a) *Error for different* β **(b)** *Exact solution for* $\beta = 30$ **(c)** *PINN solution for* $\beta = 30$

Figura 5.1: Fallo de una PINN vainilla en la convección 1D al crecer el coeficiente β: (a) el error relativo respecto a la solución exacta se dispara a partir de cierto umbral; (b) solución exacta y (c) solución de la PINN para $\beta = 30$, que ya no reconstruye el frente viajero. Reproducido de Krishnapriyan et al. (2021).

Los remedios que esos mismos autores proponen, y que la figura 5.2 ilustra para el caso del currículo, importan aquí por lo que revelan del diagnóstico. Ninguno toca la arquitectura. Tanto la regularización por currículo (empezar con un problema fácil e ir endureciéndolo) como el aprendizaje secuencia-a-secuencia que avanza en el tiempo por tramos, en la línea de Wight and Zhao (2021), actúan sobre el protocolo de entrenamiento, lo que sitúa la causa del fallo en la optimización y deja abierta la pregunta de si un cambio de arquitectura puede lograr lo mismo sin fraccionar el problema. Esa es exactamente la casilla que PINNsFormer pretende ocupar.



(a) *Curriculum regularization schematic*

(b) *Regular training PINN solution for* $\beta = 30$

(c) *Curriculum training PINN solution for* $\beta = 30$

Figura 5.2: Regularización por currículo como remedio al fallo anterior. (a) Esquema del entrenamiento por dificultad creciente del coeficiente; (b) la PINN entrenada de forma estándar falla en $\beta = 30$, mientras que (c) la entrenada con currículo recupera la solución. Reproducido de Krishnapriyan et al. (2021).

Los otros dos modos de fallo entran en esta historia por las huellas que dejan en los experimentos de esta memoria. El desequilibrio de gradientes de Wang et al. (2021a) tiene una firma medible (el residuo interior converge mientras las condiciones de contorno o iniciales quedan sin ajustar) que es precisamente la que la reproducción encontró en los fracasos de PINNsFormer en convección y onda (sección 4.3.1); conviene retener que esa firma, descrita para la PINN vainilla, puede aparecer también en la arquitectura que venía a evitarla. El sesgo espectral, por su parte, es el fallo que PINNsFormer afirma atacar con su activación WaveAct (sección 3.4), de modo que la eficacia real de esa pieza (uno de los ejes de la ablación) es también un veredicto indirecto sobre cuánto pesa ese fallo en estos benchmarks. Sobre este trasfondo triple, y no contra un único defecto bien delimitado, hay que juzgar la promesa de la arquitectura.

5.1.2. La promesa de PINNsFormer y su recepción

PINNsFormer (Zhao et al., 2024a) se presenta como una respuesta arquitectónica y no meramente algorítmica a esos fallos. Su hipótesis de partida es que la PINN vainilla, al evaluar la red punto a punto de forma independiente, ignora las dependencias temporales del sistema físico, y que ese olvido está en la raíz de los fracasos de propagación que documentó Krishnapriyan et al. en convección y reacción. La propuesta consiste en transformar cada punto de colocación en una pseudosecuencia temporal de k instantes separados por un paso Δt (sección 3.2) y procesarla con un codificador–decodificador de atención, de modo que la predicción en cada instante quede condicionada por los vecinos temporales. A ello se añade la activación WaveAct, $\phi(x) = w_1 \sin x + w_2 \cos x$ con w_1, w_2 aprendibles, pensada como remedio al sesgo espectral. La arquitectura se ofrece, por tanto, como un remedio de propósito general, ya que no ajusta pesos ni reordena el entrenamiento,

sino que cambia el aproximador para que capture de fábrica la estructura que a la PINN vainilla se le escapa.

Las cifras publicadas respaldaban esa ambición. Sobre los mismos benchmarks de convección y reacción en los que la PINN vainilla fracasa, PINNsFormer reportaba errores relativos de aproximadamente 2×10^{-2} frente a valores cercanos a la unidad de los baselines, y extendía la mejora a la ecuación de onda y a un problema de Navier–Stokes. El trabajo fue aceptado como comunicación en ICLR 2024, y su versión final incorpora un estudio de sensibilidad de hiperparámetros que examina el efecto de la longitud de secuencia k y del paso Δt sobre el error. Las cuestiones que de forma natural suscita una propuesta así (el coste computacional de multiplicar por k los puntos evaluados en cada iteración, la sensibilidad a esos dos hiperparámetros y el alcance del conjunto de EDP probadas) son las que ese estudio de sensibilidad aborda solo en parte. De hecho, como se razonó en el capítulo 4, ese mismo estudio contiene una evidencia interna incómoda. Con $k = 3$, pasar de $\Delta t = 10^{-3}$ a $\Delta t = 10^{-4}$ hace saltar el error relativo de 0.037 a 0.997, es decir, del éxito al fracaso casi total por un cambio de un único hiperparámetro. La frontera entre acierto y fracaso, ya en los datos de los propios autores, es angosta.

Esa fragilidad no invalida la propuesta, pero obliga a leerla con cautela. Una arquitectura que promete resolver de raíz los modos de fallo de la PINN, pero cuyo acierto depende de forma tan sensible de un paso temporal auxiliar sin significado físico directo, deja margen razonable para preguntarse si la mejora proviene de la estructura de atención, del aumento de parámetros que esta conlleva o de haber caído, en cada benchmark publicado, en el lado afortunado de una frontera estrecha. Son precisamente estas preguntas las que la literatura posterior ha empezado a responder.

5.1.3. Los reveses documentados de PINNsFormer

Una vez publicado el modelo, varios trabajos independientes han cuestionado sus dos pilares (el diagnóstico (que el fallo se debe a dependencias temporales ignoradas) y la solución (que la atención sobre pseudo-secuencias las recupera)) desde ángulos complementarios.

El cuestionamiento más directo a la arquitectura procede del trabajo de Arni y Blanco sobre el *Spectral PINNsformer* (Arni and Blanco, 2025). Su análisis identifica dos debilidades del diseño codificador–decodificador original. La primera es de economía, pues sostienen que el codificador es redundante cuando la captura de correlaciones espaciotemporales descansa por completo en la autoatención, de modo que buena parte del recuento de parámetros de PINNsFormer no contribuye a la precisión. La segunda es de fondo, ya que mantienen que PINNsFormer no ataca realmente el sesgo espectral, que era uno de los fallos que decía resolver, y

que WaveAct no basta para ello. Su propuesta (eliminar el codificador e incorporar características de Fourier explícitas en la entrada) iguala o supera a PINNsFormer en todos los benchmarks con una fracción de los parámetros. La lectura es reveladora. Si un rediseño más pequeño y con un tratamiento espectral explícito mejora los resultados, entonces la contribución neta de la maquinaria codificador–decodificador de PINNsFormer al problema espectral era menor de lo que sugería su presentación.

El segundo revés, ya discutido en profundidad en el capítulo 4, es el de Xu et al. (2025). Su tesis reinterpreta los modos de fallo canónicos de convección y reacción (los mismos sobre los que PINNsFormer construye su ventaja) no como mínimos genuinos del paisaje de optimización, sino como artefactos de la precisión FP32 en el optimizador L-BFGS. En simple precisión, el criterio de convergencia de la búsqueda lineal se satisface de forma prematura y el entrenamiento se detiene en una fase de fallo con residuo aparentemente convergido pero solución incorrecta. Al reentrenar en FP64, buena parte de esos fracasos de los baselines desaparecen. La implicación para PINNsFormer es delicada. Si los baselines contra los que se compara no fracasan por una limitación de fondo sino por un detalle de precisión numérica, la ventaja relativa de la arquitectura de atención se mide contra un rival artificialmente debilitado. La reproducción del capítulo 4 no solo es coherente con esta lectura, sino que la confirma de forma directa, ya que al reentrenar la matriz en precisión doble, el fracaso de PINNsFormer en convección se convierte en acierto sin más cambio que la aritmética (un rMAE de 0.751 pasa a 0.022), y los baselines, que ya en FP32 resolvían la reacción que el artículo declaraba irresoluble, se revelan competitivos en cuanto se nivela la precisión.

El tercer ángulo es de contraste de diseño más que de refutación. La familia de los *Physics-Informed Transformers* de Nagda et al. (2024) aplica la atención a la física por una vía distinta de la pseudo-secuencia. En lugar de fabricar instantes temporales auxiliares alrededor de cada punto, tratan el problema como aprendizaje de operadores sobre conjuntos, con una arquitectura invariante a la discretización del dominio que aprende relaciones entre condiciones iniciales y puntos de consulta. Validada sobre las ecuaciones de Burgers 1D y del calor 2D, esta línea muestra que la pseudo-secuencia de PINNsFormer no es la única (ni necesariamente la más robusta) manera de introducir atención en un resolutor de EDP. Que existan diseños de transformador para física que prescinden por completo del truco del Δt auxiliar refuerza la sospecha de que ese truco es una elección de ingeniería frágil antes que un principio necesario.

Puestos en fila, estos trabajos dibujan un ecosistema mucho menos concluyente que el que sugiere la lectura aislada del artículo original. La PINN vainilla falla por razones de optimización, equilibrio y espectro que se solapan; PINNsFormer promete resolverlas con atención sobre pseudo-secuencias y una activación de onda;

y la literatura posterior sostiene, con evidencia, que su codificador es prescindible y su tratamiento espectral insuficiente (Arni and Blanco, 2025), que las victorias sobre los baselines se apoyan en parte en un artefacto de precisión numérica (Xu et al., 2025), y que la pseudo-secuencia es solo una entre varias maneras (frágil, además) de llevar la atención a las EDP (Nagda et al., 2024). Este es el mapa que motiva el estudio de ablación de las secciones siguientes. En vez de aceptar o rechazar la arquitectura en bloque, se trata de desmontarla pieza a pieza (longitud y paso de la pseudo-secuencia, profundidad y anchura del transformador, activación, optimizador y precisión numérica) y medir cuánto aporta realmente cada una, separando la estructura del mero aumento de capacidad y del efecto de la aritmética de punto flotante.

5.2. Diseño del estudio de ablación

La revisión de la sección anterior y la reproducción del capítulo 4 convergen en una misma conclusión provisional, que las cifras publicadas de PINNsFormer dependen de condiciones de entrenamiento (el paso temporal, la precisión numérica) de un modo que su presentación original no deja ver, y el contraste con los *baselines* se atenúa en cuanto esas condiciones se controlan. Queda por responder una pregunta más fina. De todas las piezas que componen la arquitectura y su protocolo, ¿cuáles explican realmente su comportamiento, cuáles son indiferentes y dónde está exactamente la frontera entre el éxito y el fallo? El estudio de ablación que se diseña en esta sección se propone responderla de forma empírica y con comparaciones justas. El estudio se ha ejecutado en su totalidad (su coste de cómputo se detalla más abajo) y sus resultados ocupan las secciones 5.3 y 5.4; lo que sigue es el protocolo cerrado con el que se han obtenido.

5.2.1. Protocolo OFAT alrededor del punto del artículo

El estudio adopta un protocolo de un factor cada vez (OFAT, por sus siglas en inglés). Existe un punto central único, que no es otro que la configuración exacta del artículo tal como se fijó en la reproducción, y cada ejecución del estudio modifica de ese punto un solo hiperparámetro, dejando todos los demás en su valor central. Esta disciplina es lo que garantiza que cualquier diferencia de error observada sea imputable al eje que se ha variado y no a interacciones ocultas entre varios cambios simultáneos. El punto central se ejecuta una única vez por cada problema y actúa como fila de referencia de todas las tablas de ese problema, de modo que las comparaciones se hacen siempre contra un mismo origen y no contra cifras heterogéneas.

Tres decisiones adicionales refuerzan la equidad de la comparación. La primera es que el valor de referencia de cada hiperparámetro no se duplica en ningún archivo del estudio, sino que se toma directamente de la configuración de la reproducción, y el eje que cada ejecución ha variado se infiere después, en la fase de agregación, comparando su configuración con la del artículo; así el informe es reconstruible sin depender de metadatos frágiles. La segunda es que cada configuración (el centro y cada variación) se entrena con tres semillas distintas (0, 1 y 2), lo que permite reportar cada celda como una media acompañada de su desviación muestral y separar la variabilidad debida a la inicialización del efecto genuino del eje. Esta es, además, la respuesta directa a la limitación de semilla única señalada en el capítulo anterior (subsección 4.3.3). La tercera es que el optimizador (L-BFGS con búsqueda lineal *strong Wolfe*), la malla de evaluación (101×101), las métricas rMAE y rRMSE y la precisión aritmética se mantienen fijos en todo el estudio, de manera que ninguno de ellos contamine la lectura de los ejes que sí se varían.

5.2.2. La decisión sobre la precisión y la elección de FP64

Que la ablación se ejecute en precisión doble es una consecuencia obligada del análisis del capítulo 4. Allí se vio que en precisión simple la convergencia de estos problemas es frágil y sensible al paso temporal: la convección de PINNsFormer, sin ir más lejos, fracasa con un rMAE de 0.75 en FP32 con $\Delta t = 10^{-4}$, y la onda con 0.51, mientras que en precisión doble esos mismos casos recuperan las cifras publicadas. Anclar el estudio en precisión simple arriesgaría situar el punto central (o alguna de sus variaciones) del lado equivocado de esa frontera, de modo que lo medido no sería la sensibilidad a los ejes sino el artefacto de precisión de L-BFGS que describen Xu et al. (2025), y las conclusiones quedarían confundidas. Fijar la precisión en FP64 mantiene el punto central en la fase de éxito de forma estable y permite que lo que se mida sea, limpiamente, el efecto de cada eje. La interacción entre precisión e hiperparámetros (esto es, si algún eje desplaza la frontera FP32/FP64) se reserva como una segunda pasada opcional del mismo estudio forzando precisión simple, ajena al núcleo aquí diseñado.

5.2.3. Ejes de variación

Los ejes se eligen para cubrir las dos familias de decisiones que definen a PINNsFormer, las de su arquitectura de atención (dimensión del embedding, número de cabezas, número de capas y capacidad de las subredes internas) y las de su pseudo-secuencia temporal (longitud, paso y malla de muestreo), más el presupuesto de optimización. El estudio se limita deliberadamente a PINNsFormer, sin ablacionar la PINN vainilla, porque esta última es un *baseline* ya caracterizado a

Tabla 5.1: Ejes del estudio de ablación de PINNsFormer y valores probados. El valor central (en negrita) es el de la configuración del artículo; cada ejecución cambia un único eje respecto a ese centro. Entre paréntesis, la referencia del artículo de la que procede cada eje.

Eje	Valores probados
d_{model} (embedding)	16, **32**, 64
cabezas de atención	1, **2**, 4
N (capas enc./dec.)	**1**, 2
d_{ff} (*feed-forward*)	128, **256**, 512
d_{hidden} (MLP de salida)	256, **512**, 1024
k (longitud pseudo-secuencia)	3, **5**, 7
Δt (paso pseudo-secuencia)	10^{-2}, $\mathbf{10^{-3}}$, 10^{-4}
malla de entrenamiento	26, **51**, 101
iteraciones L-BFGS	500, **1000**

fondo en los capítulos 2 y 4 (y en la literatura que allí se revisa), de modo que incluirla solo multiplicaría el coste de cómputo sin aportar información nueva; el contraste entre ambas arquitecturas lo proporciona ya la reproducción del capítulo anterior. La tabla 5.1 recoge el conjunto de ejes; el valor central de cada uno, en negrita, es el del artículo.

Dos ejes merecen un comentario por su conexión con capítulos anteriores. La rejilla de k y Δt reproduce deliberadamente la del estudio de sensibilidad del propio artículo (su tabla 7), lo que permitirá un contraste directo, comprobar si la región estable que allí se reporta ($k \geq 5$, $\Delta t \leq 10^{-3}$) y el salto al fallo en $k = 3$, $\Delta t = 10^{-4}$ (que en el artículo dispara el error hasta 0.997) se reproducen de forma independiente. La malla de entrenamiento, por su parte, aísla una asimetría del diseño original. Dado que PINNsFormer se entrena sobre una malla más gruesa (51×51) que los *baselines* (101×101), llevar su malla a 101 (la de aquellos) permite medir si su ventaja sobrevive a igualdad de muestreo o si parte de ella se explica por esa diferencia. El eje de iteraciones, en fin, pone a prueba la hipótesis, apuntada al auditar el código, de que la diferencia entre las 500 iteraciones de los cuadernos oficiales y las 1000 del texto sea material; la reproducción del capítulo anterior la dejó planteada y aquí se mide directamente sobre PINNsFormer en los tres problemas.

5.2.4. Reparto por problema y control de coste

La matriz completa de todos los ejes sobre los tres problemas superaría las treinta horas de cómputo, por lo que el estudio concentra cada eje en el problema donde resulta más informativo, guiado por los tiempos por ejecución medidos en la reproducción en precisión doble. El reparto, resumido en la tabla 5.2, obedece a un criterio de coste-información. La reacción, el problema más barato y el que el

Tabla 5.2: Reparto de los ejes por problema y presupuesto de ejecuciones. Cada configuración se entrena con tres semillas. Navier–Stokes se excluye por incomparabilidad documentada.

Problema	Ejes cubiertos (solo PINNsFormer)	Ejecuciones
Reacción	los nueve ejes (matriz completa)	51
Convección	paso temporal Δt e iteraciones	12
Onda	iteraciones	6
Navier–Stokes	excluido	-
Total	23 configuraciones $\times$ 3 semillas	**69**

artículo emplea para su propio estudio de sensibilidad, soporta la matriz completa de los nueve ejes; la convección, de coste medio, recibe los dos ejes que comprueban si la sensibilidad hallada en reacción se transfiere al régimen de fallo por transporte ($\beta = 50$) (el paso temporal y las iteraciones) ; y la onda, el problema más caro, se limita a la hipótesis de las iteraciones. El problema de Navier–Stokes se excluye del estudio porque la auditoría del código documentó varias contradicciones internas entre el texto del artículo y su implementación oficial (el instante en que se evalúa la presión, el tratamiento del término viscoso y la convención de derivadas) que vuelven sus cifras estructuralmente incomparables, de modo que una ablación sobre esa base no admitiría comparaciones justas.

El plan resultante asciende a 23 configuraciones que, con tres semillas cada una, suman 69 ejecuciones. A partir de los tiempos por ejecución medidos en precisión doble (del orden de 180 s en reacción, 1000 s en convección y 3100 s en onda), y corrigiendo por el efecto de cada configuración sobre el coste, el estudio completo se estima en unas 9 a 11 horas en una única GPU de clase A100 (en torno a 17 dólares), o en unas 5 a 6 horas de tiempo de pared si los tres problemas se lanzan en *pods* independientes en paralelo. La infraestructura es la misma que la de la reproducción (capítulo 4), en la que cada ejecución sube su registro y sus figuras nada más terminar, de modo que la caída de un *pod* a mitad de camino no destruye lo ya calculado, y el informe se agrega automáticamente por eje en forma de media y desviación sobre las tres semillas.

5.2.5. Métricas y análisis previsto

El análisis de los resultados, que ocupará las secciones siguientes cuando el estudio se ejecute, se organiza en torno a cuatro preguntas fijadas de antemano para no incurrir en lectura sesgada de los datos. La primera es la sensibilidad por eje. Para cada hiperparámetro se medirá la degradación relativa del rMAE respecto al punto central, distinguiendo los ejes planos (en los que el error apenas cambia, señal de robustez) de los ejes con salto, en los que una pequeña variación cruza la frontera del fallo. La segunda es el contraste con la tabla 7 del artículo en la

reacción, ya descrito. La tercera es el efecto de las iteraciones sobre los tres problemas, como prueba de la hipótesis de la auditoría. La cuarta es de coste-beneficio, pues se enfrentará el error no solo al eje variado, sino al número de parámetros y al tiempo de cómputo que ese eje implica, para juzgar si la ventaja de PINNsFormer sobrevive a igualdad de malla de entrenamiento. Una norma metodológica atraviesa todo el análisis, la de que no se extraerán conclusiones mezclando precisiones ni problemas distintos; solo se comparan celdas de una misma tabla, con el mismo centro y la misma configuración de fondo.

5.3. Resultados de la ablación

El estudio diseñado en la sección anterior se ha ejecutado en su totalidad, con **69 ejecuciones en precisión doble**, con tres semillas por configuración, que barren la matriz completa de los nueve ejes de PINNsFormer sobre la 1D-reacción y, sobre la convección y la onda, los ejes señalados como potencialmente decisivos (el paso temporal y las iteraciones). Todas las cifras que siguen son medias sobre las tres semillas acompañadas de su desviación muestral, y se leen contra la fila del punto central de cada problema, que es la configuración exacta del artículo. Al juzgar diferencias pequeñas hay que tener presente la magnitud de esas desviaciones, pues una variación de una o dos décimas en la escala de 10^{-2} cae dentro del ruido de inicialización y no debe sobreinterpretarse. Los resultados se presentan por problema, de la sensibilidad arquitectónica más completa (la de la reacción) a las pruebas dirigidas sobre convección y onda.

5.3.1. Sensibilidad arquitectónica completa sobre la 1D-reacción

La reacción es el único problema sobre el que se han recorrido los nueve ejes, y por tanto el que ofrece el mapa completo de sensibilidad de la arquitectura (tabla 5.3). Su punto central rinde un rMAE de $(1.86 \pm 0.24) \times 10^{-2}$, una solución plenamente correcta.

La primera lectura de la tabla 5.3 es que, sobre la 1D-reacción, ninguna variación de un solo eje saca a PINNsFormer de la solución correcta, y el rMAE oscila entre 1.2 y 2.6×10^{-2} en todas las filas, sin un solo fracaso. La arquitectura es, en este problema, notablemente robusta a sus propios hiperparámetros. Dentro de esa estabilidad, los ejes se separan en tres grupos de comportamiento bien diferenciado.

Los ejes de la maquinaria de atención (número de cabezas, dimensión del embedding por encima de su valor central y número de capas) apenas mueven la aguja.

Tabla 5.3: Ablación OFAT de PINNsFormer sobre la 1D-reacción (precisión doble, media $\pm$ desviación sobre tres semillas). Cada fila varía un único eje respecto al punto central (configuración del artículo). rMAE en unidades de 10^{-2}; el número de parámetros y el tiempo medio por ejecución acompañan a cada fila para el análisis de coste-beneficio.

Eje	Valor	rMAE ($\times 10^{-2}$)	N.º params	t (s)
centro (paper)	—	1.86 ± 0.24	453 561	173
d_{model}	16	2.41 ± 0.88	422 633	333
d_{model}	64	1.88 ± 0.72	527 705	180
cabezas	1	1.74 ± 0.46	453 561	191
cabezas	4	2.01 ± 0.09	453 561	178
N	2	1.58 ± 0.37	626 953	211
d_{ff}	128	1.48 ± 0.86	338 361	241
d_{ff}	512	2.61 ± 0.08	880 569	493
d_{hidden}	256	1.24 ± 0.57	247 993	171
d_{hidden}	1024	2.54 ± 0.11	1 257 913	533
k (num. pasos)	3	1.95 ± 0.27	453 561	166
k (num. pasos)	7	1.57 ± 0.38	453 561	1028
Δt (paso)	10^{-4}	2.09 ± 1.10	453 561	159
Δt (paso)	10^{-2}	1.73 ± 0.36	453 561	210
malla	26	1.27 ± 0.35	453 561	171
malla	101	1.62 ± 0.08	453 561	969
iteraciones	500	1.86 ± 0.24	453 561	146

Pasar de dos a una o a cuatro cabezas deja el error en 1.74 y 2.01×10^{-2}, ambos solapados con el centro dentro de la desviación; duplicar d_{model} a 64 no cambia nada (1.88); y añadir una segunda capa de codificador–decodificador ($N = 2$) rebaja el error a 1.58 a costa de un 38 % más de parámetros. La estructura de atención admite, pues, un abanico amplio de configuraciones sin que el resultado se resienta.

El segundo grupo, el de la capacidad interna, ofrece la señal más nítida y también la más llamativa, y es que ampliarla empeora el resultado. Elevar la dimensión del *feed-forward* de 256 a 512 lleva el rMAE de 1.86 a 2.61×10^{-2} y casi duplica los parámetros; ampliar el MLP de salida de 512 a 1024 lo lleva a 2.54 con casi el triple de parámetros. En sentido contrario, reducir esas mismas dimensiones tiende a mejorar la precisión y abarata el modelo, pues $d_{\text{ff}} = 128$ baja a 1.48×10^{-2} con un tercio menos de parámetros, y $d_{\text{hidden}} = 256$ ofrece el mejor rMAE de toda la tabla, 1.24×10^{-2}, con casi la mitad de parámetros y el mismo tiempo. El valor central $d_{\text{ff}} = 256$ (que, recuérdese, no figura en el artículo y procede del código) no es siquiera el óptimo.

El tercer grupo lo forman los ejes de la pseudo-secuencia y el muestreo, cuyo efecto es pequeño en la reacción pero de coste muy desigual. Alargar la secuencia a

$k = 7$ mejora levemente el error (1.57) pero multiplica por seis el tiempo de cómputo; refinar la malla a 101 lo baja a 1.62 a costa de casi seis veces más tiempo; y el paso temporal Δt, que en este problema es indiferente (2.09 y 1.73×10^{-2} para 10^{-4} y 10^{-2}), resultará decisivo en convección. Merece subrayarse el dato de la malla por su lectura de coste-beneficio. Al entrenar sobre la misma rejilla de 101×101 que emplean los *baselines*, PINNsFormer no solo no se degrada, sino que mejora ligeramente respecto a su malla reducida de 51 (1.62 frente a 1.86×10^{-2}). Su ventaja en la reacción, por tanto, no descansa en la reducción intencionada del muestreo. Un último dato, sobre las iteraciones, se comenta en su contexto más adelante, y es que pasar de 1000 a 500 iteraciones de L-BFGS deja el error exactamente igual que el centro, señal de que el optimizador ha convergido mucho antes del medio millar de iteraciones en este problema.

5.3.2. El paso temporal y las iteraciones en la convección

Trasladado a la convección (el problema de transporte con $\beta = 50$ que en precisión simple hacía fracasar a PINNsFormer), el eje del paso temporal deja de ser inocuo (tabla 5.4). Con el valor central 10^{-3} y con 10^{-4} el modelo resuelve el problema (2.57 y 1.93×10^{-2}), pero elevar el paso a 10^{-2} lo hunde en un rMAE de 1.110 (mayor que uno, esto es, solución trivial). El paso temporal se revela así como un umbral entre el éxito y el fallo, con un salto de casi dos órdenes de magnitud entre dos valores contiguos de la rejilla. El eje de iteraciones, en cambio, produce aquí un efecto pequeño pero ya no nulo, pues reducirlas a 500 empeora levemente el error, de 2.57 a 2.94×10^{-2}, lo que indica que en convección L-BFGS todavía progresa algo entre las 500 y las 1000 iteraciones, al contrario que en la reacción.

Tabla 5.4: Ablación de PINNsFormer sobre la convección ($\beta = 50$, precisión doble, media $\pm$ desviación sobre tres semillas). El punto central es $\Delta t = 10^{-3}$ con 1000 iteraciones. Un rMAE mayor que uno indica que el modelo no ha aprendido la solución.

Configuración	rMAE	t (s)
centro ($\Delta t = 10^{-3}$, 1000 it.)	0.0257 ± 0.0046	1093
$\Delta t = 10^{-4}$	0.0193 ± 0.0022	1139
$\Delta t = 10^{-2}$	1.110 ± 0.023	1647
iteraciones $= 500$	0.0294 ± 0.0071	843

5.3.3. El presupuesto de optimización en la onda

La ecuación de onda es el problema donde PINNsFormer rinde peor de los tres, y donde el eje de iteraciones tiene su efecto más acusado (tabla 5.5). Su punto central, con 1000 iteraciones y en precisión doble, alcanza un rMAE de $(2.13 \pm 0.42) \times 10^{-1}$; reducir el presupuesto a 500 iteraciones lo empeora de forma marcada, hasta $(3.55 \pm 0.06) \times 10^{-1}$. A diferencia de la reacción, donde el margen entre 500 y 1000 era ocioso, en la onda cada iteración cuenta, porque L-BFGS sigue lejos de la convergencia al llegar a las 500. Conviene, no obstante, poner la cifra central en perspectiva. Un rMAE de 0.21 es más de un orden de magnitud peor que el que alcanzan sobre este mismo problema, y en la misma precisión doble, tanto la PINN vainilla (0.011) como la variante FLS (0.030), según se vio en la reproducción (capítulo 4). En la onda, por tanto, PINNsFormer no fracasa de forma catastrófica, pero es la peor de las arquitecturas evaluadas, y las mil iteraciones del artículo apenas bastan para dejarlo cerca de su propia cifra publicada. La figura 5.3 contrasta ambos casos sobre el campo de solución.

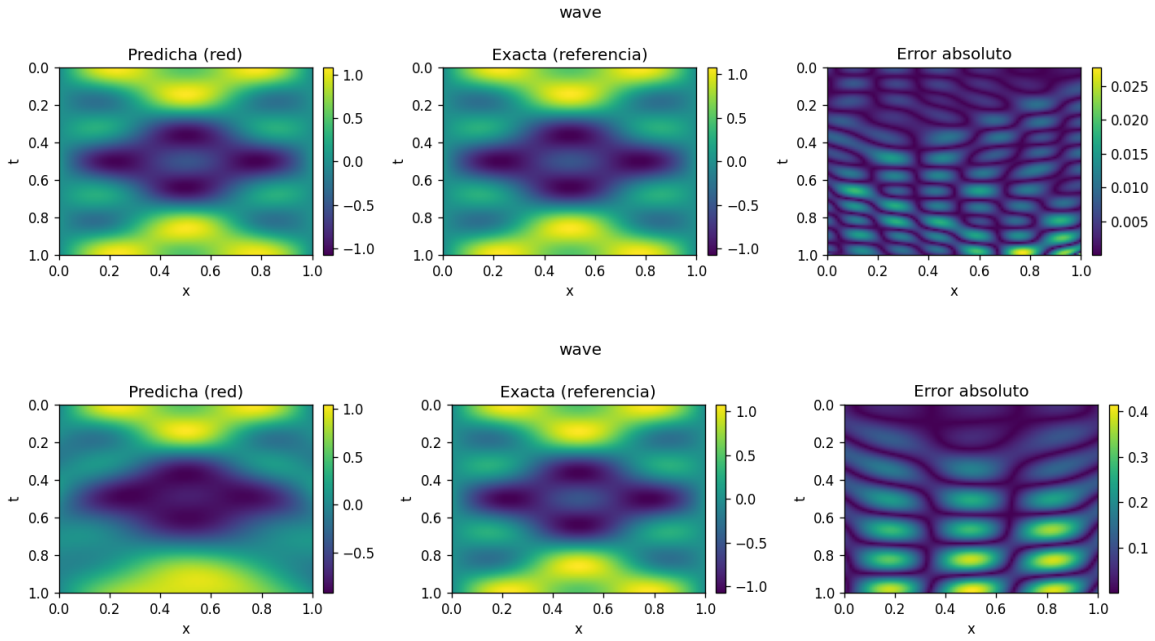


Figura 5.3: Ecuación de onda en precisión doble. Solución predicha, exacta y error absoluto. Arriba, la PINN vainilla (rMAE 0.011), que reconstruye la solución oscilatoria con fidelidad. Abajo, PINNsFormer (rMAE 0.229), cuya predicción pierde amplitud y deja un error un orden de magnitud mayor. La debilidad de PINNsFormer en la onda persiste con la aritmética, el paso y el presupuesto a su favor, es genuina y no un artefacto de precisión.

Tabla 5.5: Ablación del presupuesto de iteraciones de PINNsFormer sobre la onda (precisión doble, media $\pm$ desviación sobre tres semillas). El punto central usa 1000 iteraciones. rMAE en unidades de 10^{-1}.

Iteraciones	rMAE ($\times 10^{-1}$)	t (s)
1000 (centro)	2.13 ± 0.42	3277
500	3.55 ± 0.06	1651

5.4. Discusión. Modos de fallo confirmados e interpretación

Con el estudio ya completo, los resultados dialogan de forma directa con las tres partes de este capítulo (los modos de fallo de la PINN que motivan a PINNsFormer (sección 5.1.1), los reveses documentados de la arquitectura (sección 5.1.3) y las discrepancias de la reproducción (capítulo 4)) y permiten un juicio de conjunto sobre qué explica el comportamiento de la arquitectura y qué no. Se ordena en torno a las preguntas que el diseño fijó de antemano.

La atención no es lo que decide el resultado. La robustez de PINNsFormer a sus ejes de atención (cabezas, dimensión del embedding, número de capas) es un hallazgo con lectura doble. Por un lado es una virtud, porque el modelo no exige un ajuste fino de esos hiperparámetros. Por otro, y más revelador, indica que la configuración concreta de la atención que el artículo prescribe no es la causa de su acierto, ya que un abanico amplio de configuraciones rinde prácticamente igual, y ni siquiera duplicar la dimensión del embedding o el número de cabezas cambia el resultado de forma perceptible. La observación coincide con la línea del *Spectral PINNsformer* de Arni y Blanco (Arni and Blanco, 2025), que sostenía sobre bases teóricas que buena parte de la maquinaria codificador–decodificador de PINNsFormer es prescindible; la ablación aporta ahora evidencia empírica directa en la misma dirección.

La capacidad interna está sobredimensionada. La señal más firme del estudio es que ampliar la dimensión del *feed-forward* o del MLP de salida degrada la precisión mientras multiplica el coste, y que reducirlas la mejora. El efecto es inequívoco en los extremos ($d_{\text{ff}} = 512$ y $d_{\text{hidden}} = 1024$ superan al centro por márgenes mayores que su desviación) y confirma que el punto central heredado del código carga con parámetros que no solo no ayudan, sino que estorban. Que el mejor rMAE de toda la tabla se obtenga con la mitad de parámetros ($d_{\text{hidden}} = 256$) es la prueba más contundente de redundancia, es decir, que la arquitectura, tal como se publica, no está en su frontera de eficiencia, y una versión más pequeña la mejoraría en precisión y en coste a la vez. Esta conclusión, unida a la anterior, socava la narrativa según la cual el mérito de PINNsFormer reside en la riqueza de su arquitectura de atención; los datos apuntan más bien a que esa riqueza es, en parte, lastre.

El paso temporal es el único modo de fallo propio de la arquitectura. De todos los ejes probados, solo Δt produce un fracaso genuino, y lo hace en convección al valor 10^{-2}. Este resultado cierra el argumento que venía construyéndose desde el capítulo 4. El dispositivo que PINNsFormer introduce para capturar las dependencias temporales (la pseudo-secuencia) es también su punto único de fragilidad. La frontera de fallo no vive en la profundidad de la red ni en el número de cabezas, sino en un paso temporal auxiliar sin significado físico, y su posición depende del problema, inocuo en la reacción, decisivo en la convección. Es la misma fragilidad que el propio artículo documenta en su tabla 7 y que la reproducción en precisión simple ya había hecho aflorar por otra vía. La lectura conjunta es que la contribución más singular de PINNsFormer (convertir cada punto en una pequeña secuencia temporal) introduce, junto con su supuesta ventaja, un hiperparámetro nuevo y frágil que la PINN vainilla sencillamente no tiene.

El efecto de las iteraciones es real, pero depende del problema. La hipótesis de la auditoría (que la diferencia entre las 500 iteraciones de los cuadernos oficiales y las 1000 del texto sea material) recibe una respuesta matizada al medirse sobre los tres problemas. En la reacción es nula, ya que 500 y 1000 iteraciones dan el mismo número, porque L-BFGS converge mucho antes. En la convección es pequeña (0.029 frente a 0.026). En la onda, en cambio, es más notoria, pues pasar de 500 a 1000 iteraciones mejora el rMAE de 0.355 a 0.213, casi a la mitad. El presupuesto de optimización, por tanto, sí importa (y mucho) en el problema más difícil y más caro, donde el optimizador aún está lejos de converger al llegar a las 500 iteraciones. La consecuencia metodológica es directa, y es que comparar arquitecturas con un presupuesto de iteraciones insuficiente puede penalizar a unas más que a otras, y buena parte de las discrepancias entre implementaciones que usan 500 o 1000 iteraciones se explica por este efecto en los problemas costosos.

La onda es una debilidad genuina, no un artefacto. El capítulo 4 ya mostraba que, en precisión doble, PINNsFormer quedaba muy por detrás de la PINN vainilla y de FLS en la ecuación de onda. La ablación lo confirma y lo precisa, pues ni siquiera agotando el presupuesto de mil iteraciones (donde el rendimiento aún está mejorando) PINNsFormer se acerca a las cifras de los *baselines*; se estanca en un rMAE de 0.21, un orden de magnitud peor. A diferencia de la convección, cuyo fracaso en precisión simple resultó ser un artefacto rescatable con FP64, la mediocridad en la onda persiste con la aritmética, el paso y el presupuesto en su favor. Lleva a pensar, por tanto, que es una limitación real de la arquitectura sobre este tipo de solución oscilatoria, y no un efecto de precisión ni de optimización insuficiente. Que el propio artículo solo alcanzara sus mejores cifras de onda recurriendo a una reponderación por NTK que su código no acompaña (y que aquí no se reproduce), apunta en la misma dirección.

El corredor de fallo de la tabla 7 no se reproduce con variaciones aisladas. El artículo reporta que la combinación $k = 3$ con $\Delta t = 10^{-4}$ dispara el error de la reacción hasta 0.997. La ablación no reproduce ese fracaso, ya que, por separado, $k = 3$ rinde 1.95×10^{-2} y $\Delta t = 10^{-4}$ rinde 2.09×10^{-2}, ambos plenamente exitosos. La explicación es doble y coherente con todo lo anterior. Primero, el protocolo OFAT varía un solo eje cada vez y por diseño nunca prueba la esquina $k = 3 \wedge \Delta t = 10^{-4}$, que es una interacción, no un efecto marginal. Segundo, la ablación corre en precisión doble, donde (como mostró el capítulo 4) desaparecen los estancamientos que en precisión simple pueblan justamente esas esquinas. Que un fracaso tan aparatoso en el artículo se evapore al aislar los factores y elevar la precisión es una pieza más a favor de la lectura del FP64 (Xu et al., 2025).

Puestas en común, las cinco observaciones componen un retrato coherente y desmitificador de PINNsFormer. Sobre el problema en el que gana con claridad (la

reacción), su acierto es robusto, pero no proviene de las decisiones de atención que lo singularizan, sino que sobrevive a casi cualquier variación de ellas; su arquitectura está sobredimensionada, hasta el punto de que recortarla mejora resultado y coste; su único modo de fallo propio es el paso de la pseudo-secuencia, el hiperparámetro que la arquitectura añade y la PINN no tiene; y en el problema oscilatorio queda claramente por detrás de alternativas mucho más simples. La ablación no niega que PINNsFormer resuelva los benchmarks que resuelve, pero sí desplaza el crédito. Lo que lo hace funcionar no es la sofisticación de su atención, y lo que lo hace fallar es justamente su aportación más distintiva.

5.5. Limitaciones del estudio

El alcance de las conclusiones anteriores está acotado por varias limitaciones que conviene declarar sin tapujos. La primera es la cobertura desigual entre problemas, pues la sensibilidad a la totalidad de los ejes arquitectónicos se ha medido solo sobre la reacción, mientras que en convección y onda el estudio se restringe al paso temporal y a las iteraciones. Nada garantiza que la robustez a la atención o el sobredimensionamiento observados en la reacción se reproduzcan idénticos en un problema de transporte o en uno oscilatorio; extender el barrido completo de ejes a esos dos benchmarks es la continuación natural del trabajo. Por la misma razón, la comparación frente a la PINN vainilla se toma de la reproducción (capítulo 4) y no de la propia ablación, que por decisión de diseño se limita a PINNsFormer.

La segunda limitación es metodológica y es inherente al protocolo OFAT. Al variar un factor cada vez, el estudio mide efectos marginales pero es ciego a las interacciones entre ejes. La esquina $k = 3 \wedge \Delta t = 10^{-4}$ de la tabla 7 del artículo ilustra el punto, pues un diseño OFAT nunca la visita; un fallo que solo se manifieste en la combinación de dos valores pasaría inadvertido. Un diseño factorial completo lo capturaría, a un coste de cómputo mucho mayor.

La tercera atañe a la precisión. El estudio se ejecuta en FP64 por las razones expuestas en la sección 5.2.2, lo que aísla el efecto de los ejes del artefacto de precisión, pero a cambio no cuantifica la interacción entre precisión e hiperparámetros; esa segunda pasada en precisión simple (prevista en el diseño) queda pendiente. La cuarta es de representatividad, ya que los cuatro benchmarks son problemas canónicos de baja dimensión, y las conclusiones no deben extrapolarse sin más a EDP de mayor dimensión o a geometrías complejas, ni a otros regímenes de los coeficientes. Por último, aunque tres semillas superan la única del artículo y permiten cuantificar la variabilidad, siguen siendo una muestra pequeña, y las desviaciones observadas (considerables en algunos ejes) recomiendan prudencia al ordenar configuraciones cuyas diferencias caen dentro de ese margen.

Ninguna de estas limitaciones invalida los hallazgos firmes (la robustez a la atención, el sobredimensionamiento de la capacidad interna, la fragilidad del paso temporal y la debilidad genuina en la onda), pero sí delimitan con honestidad hasta dónde llegan y qué faltaría para convertir esta caracterización de PINNsFormer en un veredicto definitivo sobre la familia de las arquitecturas de atención para PINN.

6

Conclusiones y trabajo futuro

Este trabajo se planteó como un estudio metodológico que recorriera el camino completo desde las redes neuronales informadas por la física hasta la arquitectura PINNsFormer, con un propósito que no era anunciar una mejora más, sino comprobar con rigor qué se sostiene de las que se publican. Los objetivos fijados en el capítulo 1 se han cumplido en ese orden. Se explicaron las PINN estándar y se las situó frente al aprendizaje profundo y a los métodos numéricos clásicos (capítulo 2); se recopilaron sus modos de fallo documentados; se estudió en detalle la arquitectura PINNsFormer relacionando el artículo con su implementación (capítulo 3); se reprodujo de forma independiente sobre sus cuatro problemas de referencia (capítulo 4); y se diseñó y ejecutó un estudio de ablación de sus parámetros (capítulo 5). Este capítulo recoge lo que esa trayectoria permite afirmar, ateniéndose a las cifras medidas, y señala las líneas que el análisis deja abiertas.

6.1. Conclusiones

Las conclusiones se ordenan en tres planos, lo que la reproducción independiente reveló sobre las cifras publicadas, lo que la ablación reveló sobre la arquitectura, y la lección metodológica que atraviesa a ambas y que constituye, quizá, la aportación más transferible del trabajo.

6.1.1. La reproducción, una superioridad condicionada

La reproducción independiente del artículo, ejecutada sobre una infraestructura determinista y en varias configuraciones de precisión y paso temporal, no confirma la superioridad tajante que el artículo original exhibe. Las cifras de referencia de PINNsFormer se recuperan, pero solo bajo condiciones de entrenamiento precisas. Su resultado estrella en convección (un error relativo de 0.023) se reproduce con paso temporal $\Delta t = 10^{-3}$ en precisión simple (0.030) y en precisión doble (0.022), pero se desploma hasta 0.751 en precisión simple con $\Delta t = 10^{-4}$, sin que

cambie nada más que un paso temporal auxiliar sin significado físico directo. La reacción se reproduce de forma robusta, mientras que la ecuación de onda no reproduce la ventaja publicada bajo ninguna de las condiciones ensayadas, y la presión de Navier–Stokes resulta incomparable por su indeterminación aditiva. Más aún, el contraste dramático que el artículo dibuja entre unos *baselines* que fracasan y un PINNsFormer que triunfa se atenúa al reproducirlo, pues la PINN vainilla resuelve la reacción que el artículo declaraba irresoluble para ella (0.043 frente al 0.982 publicado), y en precisión doble también FLS la resuelve.

La causa de estas divergencias quedó establecida de forma directa al reentrenar la matriz completa en precisión doble. Los fracasos de convección y onda que aparecían en precisión simple se convierten en aciertos con solo cambiar los bits de la aritmética (la convección de PINNsFormer pasa de 0.751 a 0.022, recuperando con exactitud la cifra publicada), en confirmación empírica exacta de la tesis de Xu et al. (2025), pues en precisión simple, el criterio de convergencia de L-BFGS se satisface de forma prematura y el entrenamiento se estanca en una fase de fallo con residuo convergido y solución incorrecta, no en un mínimo insalvable del problema. La reproducción muestra, así, que buena parte de la ventaja publicada de PINNsFormer sobre sus *baselines* no es de naturaleza arquitectónica, sino la diferencia entre entrenamientos que cruzaron a la fase de éxito y otros que se quedaron en la de fallo, un cruce que en precisión simple depende de factores ajenos a la arquitectura. Lo que sí falla con independencia de la precisión (la variante QRes en reacción, que se estanca en 0.988 en ambas) delimita, por contraste, dónde hay una limitación genuina y dónde solo un artefacto numérico.

6.1.2. Qué explica, según la ablación, el comportamiento de PINNsFormer

El estudio de ablación, con sus sesenta y nueve ejecuciones en precisión doble sobre PINNsFormer, desplaza el crédito de su rendimiento lejos de donde el artículo lo sitúa. Sobre la reacción, el problema que la arquitectura resuelve con holgura, ninguna variación aislada de sus hiperparámetros la saca de la solución correcta, pero ese acierto no proviene de las decisiones que la singularizan. Los ejes de la maquinaria de atención (número de cabezas, dimensión del embedding, número de capas) son prácticamente planos, ya que un abanico amplio de configuraciones rinde igual, de modo que la sofisticación concreta de la atención no es la causa del éxito. Esta evidencia empírica coincide con la objeción de redundancia del *Spectral PINNsformer* de Arni y Blanco (Arni and Blanco, 2025), que sostenía sobre bases teóricas que buena parte del codificador es prescindible.

La señal más firme del estudio va un paso más allá. La capacidad interna de la arquitectura está sobredimensionada. Ampliar la dimensión del *feed-forward* o del

MLP de salida degrada la precisión mientras multiplica los parámetros, y reducirlas la mejora (el mejor error de toda la tabla se obtiene con la mitad de los parámetros del punto central). La arquitectura, tal como se publica, no está en su frontera de eficiencia, y una versión más pequeña la superaría en precisión y en coste a la vez. De todos los ejes, solo el paso temporal de la pseudo-secuencia produce un fracaso genuino, y lo hace de forma dependiente del problema, inocuo en la reacción y catastrófico en la convección con $\Delta t = 10^{-2}$. La contribución más distintiva de PINNsFormer (convertir cada punto en una pequeña secuencia temporal) introduce, junto con su supuesta ventaja, un hiperparámetro nuevo y frágil que la PINN vainilla sencillamente no tiene, y que resulta ser su único modo de fallo propio. A ello se suma que en la ecuación de onda PINNsFormer queda un orden de magnitud por detrás de alternativas mucho más simples (y sin que el presupuesto de mil iteraciones baste para acercarlo a su cifra publicada, aunque las iteraciones, a diferencia de la reacción, sí importan en este problema costoso), lo que revela una debilidad estructural y no un artefacto de precisión. En conjunto, la ablación obliga a revisar la imagen habitual de la arquitectura, pues lo que hace funcionar a PINNsFormer difícilmente puede ser la riqueza de su atención, cuando su acierto sobrevive a casi cualquier variación de ella; y lo que lo hace fallar es justamente su aportación más característica.

6.1.3. La lección metodológica sobre precisión, semilla y reproducibilidad

Por encima de lo que la reproducción y la ablación digan sobre PINNsFormer en particular, el trabajo deja una conclusión de método que trasciende a esta arquitectura concreta. Sobre los benchmarks canónicos de las PINN, la precisión numérica y el protocolo de entrenamiento (el paso temporal, el presupuesto de iteraciones, la semilla) explican una parte mayor de las diferencias publicadas que la propia arquitectura. Una comparación que no controle estos factores puede atribuir a una red lo que en realidad corresponde a la aritmética de coma flotante o a una realización afortunada del azar. De aquí se siguen tres exigencias para cualquier evaluación rigurosa de estas arquitecturas. Conviene ejecutar en precisión doble para no medir el artefacto de L-BFGS que documentan Xu et al. (2025); promediar sobre varias semillas, porque una sola es insuficiente cuando la frontera entre el fallo y el éxito es tan estrecha como la que el propio estudio de sensibilidad del artículo (su tabla 7) ya evidenciaba; e igualar los presupuestos de cómputo, dado que un número de iteraciones insuficiente penaliza de forma desigual a unos modelos frente a otros.

Que esta reproducción pudiera afirmar tales cosas se apoya en una propiedad que no debe pasarse por alto, pues las repeticiones independientes de cada configuración devolvieron métricas idénticas hasta la última cifra significativa, de modo que

cualquier diferencia entre configuraciones es atribuible a un factor concreto y no al ruido del entrenamiento. Ese determinismo estricto es la condición que convierte las comparaciones del trabajo en argumentos y no en impresiones. La lección enlaza, por lo demás, con la línea que arranca en la caracterización de los modos de fallo de las PINN de Krishnapriyan et al. (2021) y culmina en la reinterpretación de esos fallos como estancamientos inducidos por la precisión, de modo que lo que a primera vista parece una virtud o un defecto de una arquitectura puede ser, examinado de cerca, un fenómeno de la optimización. Leer las afirmaciones del estado del arte sobre estos problemas con esa cautela (y con reproducción, ablación y control de la precisión por delante) es la actitud que este trabajo reivindica.

6.1.4. Aportaciones del trabajo

De lo anterior se desprenden las aportaciones concretas de este Trabajo Fin de Máster. La primera es una reproducción independiente de PINNsFormer en varias configuraciones de precisión y paso temporal que, además de contrastar las cifras publicadas, aporta una confirmación empírica directa de la tesis del FP64 sobre la propia arquitectura, y no solo sobre las PINN vainilla en las que se formuló. La segunda es un estudio de ablación sistemático de PINNsFormer en precisión doble que identifica, con datos, la sobreparametrización de su capacidad interna y la fragilidad de su paso temporal como los dos hechos que mejor explican su comportamiento. La tercera es la infraestructura experimental reproducible y determinista sobre la que descansan ambos, cuya paridad de resultados entre entornos es verificable. Y la cuarta, de carácter transversal, es una lectura de PINNsFormer apoyada en los hechos y en la literatura reciente (la del *Spectral PINNsformer* (Arni and Blanco, 2025), la del FP64 (Xu et al., 2025) y la de los transformadores para física por otras vías (Nagda et al., 2024)) que distingue sus méritos reales de los que un examen superficial le atribuiría.

Ninguna de estas conclusiones niega que PINNsFormer resuelva los problemas que resuelve. Lo que el trabajo establece es más matizado y, por ello, más útil. Su ventaja sobre alternativas simples y bien optimizadas es estrecha y condicionada, su arquitectura admite un adelgazamiento que la mejoraría, y su aportación más singular es a la vez el origen de su única fragilidad propia. El valor de la pseudosecuencia con atención para las PINN, lejos de quedar refutado, queda situado en su justa medida.

6.2. Líneas futuras

El propio carácter acotado del estudio conduce a continuaciones naturales, que se agrupan en tres direcciones.

La primera es completar la caracterización empírica de PINNsFormer. La sensibilidad a la totalidad de los ejes arquitectónicos se ha medido de forma exhaustiva solo sobre la reacción; extender ese barrido completo a la convección y a la onda comprobaría si la robustez a la atención y la sobreparametrización observadas son generales o propias del problema más benigno. A ello se añade un diseño factorial que, a diferencia del protocolo de un factor cada vez empleado aquí, capture las interacciones entre ejes (en particular la esquina $k = 3$ con $\Delta t = 10^{-4}$ que el artículo reporta como fallo y que, por separado, ninguno de los dos valores reproduce) y una segunda pasada en precisión simple que cuantifique la interacción entre precisión e hiperparámetros, cartografiando la frontera FP32/FP64 sobre cada eje. La consolidación de las jerarquías mediante un mayor número de semillas cerraría la cuestión de la variabilidad que este trabajo solo ha podido acotar con tres.

La segunda dirección es arquitectónica. Las debilidades detectadas (capacidad interna sobredimensionada y mal comportamiento en la ecuación de onda) apuntan a variantes concretas que merecería la pena evaluar con el mismo protocolo. Es el caso del *Spectral PINNsformer*, que elimina el codificador redundante e incorpora características de Fourier explícitas para atacar el sesgo espectral (Arni and Blanco, 2025); la combinación de PINNsFormer con la reponderación por núcleo tangente neuronal (Wang et al., 2022) y con el entrenamiento causal (Wang et al., 2024), especialmente indicada para el problema oscilatorio donde la arquitectura flaquea; y las aproximaciones que llevan la atención a la física por vías distintas de la pseudo-secuencia, como el aprendizaje de operadores invariante a la discretización de los *Physics-Informed Transformers* (Nagda et al., 2024). Comparar estas alternativas entre sí, en precisión doble y con presupuestos igualados, permitiría dilucidar qué ingredientes de la atención resultan realmente útiles para resolver EDP.

La tercera dirección lleva el estudio más allá de los benchmarks canónicos de baja dimensión. La utilidad práctica de estas arquitecturas se juega en problemas de mayor dimensión, en geometrías complejas y en regímenes de los coeficientes donde los métodos numéricos clásicos se encarecen, que son precisamente los nichos en los que una PINN tiene algo que ofrecer. Un caso aplicado de elastodinámica (como el de una viga empotrada, no abordado en esta memoria) constituiría un banco de pruebas concreto para contrastar si las conclusiones obtenidas sobre problemas de referencia se trasladan a un escenario de ingeniería realista. En última instancia, la contribución más valiosa que este trabajo sugiere quizá no sea una arquitectura concreta, sino un modo de evaluarlas, una batería de comparación con precisión controlada, múltiples semillas y presupuestos equitativos, que permita separar de una vez lo que aporta el diseño de lo que aporta la aritmética.

Bibliografía

Rohan Arni and Carlos Blanco. Physics-informed neural networks with Fourier features and attention-driven decoding. NeurIPS 2025 Workshop on Machine Learning and the Physical Sciences (AI4Science), 2025. arXiv:2510.05385.

Atılım Güneş Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.

Claudio Canuto, M. Yousuff Hussaini, Alfio Quarteroni, and Thomas A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer, Berlin, 2006.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. doi: 10.1016/0893-6080(89)90020-8.

George E. Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6): 422–440, 2021.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Isaac E. Lagaris, Aristidis Likas, and Dimitrios I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

Mayank Nagda et al. PITs: Physics-informed transformers for predicting chemical phenomena. In *Workshop on Machine Learning for Chemistry and Chemical Engineering (ML4CCE), ECML-PKDD*, 2024.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021a.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021b.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449: 110768, 2022.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421:116813, 2024. Preprint arXiv:2203.07404, 2022.

Colby L. Wight and Jia Zhao. Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks. Communications in Computational Physics, 2021.

Chenhui Xu, Dancheng Liu, Amir Nassereldine, and Jinjun Xiong. FP64 is all you need: Rethinking failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.10949.

Zhiyuan Zhao, Xueying Ding, and B. Aditya Prakash. Pinnsformer: A transformer-based framework for physics-informed neural networks. In *International Conference on Learning Representations (ICLR)*, 2024a.

Zhiyuan Zhao, Xueying Ding, and B. Aditya Prakash. PINNsFormer: official repository. `https://github.com/AdityaLab/pinnsformer`, 2024b. Consultado en junio de 2026.

Olgierd C. Zienkiewicz, Robert L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Butterworth-Heinemann, Oxford, 6 edition, 2005.

Apéndices

Apéndice A

Hiperparámetros y configuración experimental

Este apéndice recoge, en un solo lugar, todos los hiperparámetros y constantes con los que se ejecutaron la reproducción del capítulo 4 y el estudio de ablación del capítulo 5. Los valores proceden de un único fichero de configuración que actúa como fuente de verdad del proyecto experimental, y están verificados línea a línea contra la tabla 4 y el apéndice A del artículo original (Zhao et al., 2024a). Se separan en tres bloques, la arquitectura de cada modelo, el protocolo de entrenamiento común y las constantes propias de cada ecuación.

A.1. Arquitectura de los modelos

La tabla A.1 detalla la estructura de los cuatro modelos. El recuento de parámetros se mantiene del mismo orden de magnitud a propósito (en torno a medio millón), de modo que ninguna diferencia de rendimiento pueda atribuirse a una sobreparametrización de uno de ellos frente a los demás.

Tabla A.1: Arquitectura de cada modelo (tabla 4 del artículo). Los *baselines* usan tangente hiperbólica como activación; FLS sustituye la primera capa por una activación seno y PINNsFormer emplea la activación aprendible WaveAct. La dimensión interna $d_{\text{ff}} = 256$ de PINNsFormer no figura en el artículo y se toma del código oficial.

Modelo	Estructura	N.º params.
PINN	4 capas ocultas, tamaño 512, tanh	∼527 k
FLS	4 capas (1.ª seno, resto tanh), tamaño 512	∼527 k
QRes	4 capas de bloques cuadráticos, tamaño 256	∼397 k
PINNsFormer	1 enc. + 1 dec., d_{model}=32, h=2 cabezas, d_{ff}=256, salida d_{hidden}=512	453 561

A.2. Protocolo de entrenamiento

El protocolo de optimización es común a los cuatro modelos y a los cuatro problemas. Todos se entrenan con el optimizador cuasi-Newton **L-BFGS** con búsqueda lineal *strong Wolfe* y tasa de aprendizaje 1.0, durante **1000 iteraciones**. Los tres términos de la pérdida (residuo de la ecuación, condición inicial y condición de contorno) se ponderan por igual, $\lambda_{\text{res}} = \lambda_{\text{ic}} = \lambda_{\text{bc}} = 1$. El error se mide con las métricas de error relativo ℓ_1 (rMAE) y ℓ_2 (rRMSE) de la ecuación (2.8), evaluadas sobre una malla de test de 101×101 puntos común a todos los modelos.

La tabla A.2 resume los parámetros de muestreo y de la pseudo-secuencia. La malla de puntos de colocación no es común, pues los *baselines* se entrenan sobre una malla fina de 101×101 y PINNsFormer sobre una gruesa de 51×51, una reducción intencionada del artículo. La pseudo-secuencia de PINNsFormer usa longitud $k = 5$ en todos los problemas; para el paso temporal Δt el artículo indica el conjunto $\{10^{-3}, 10^{-4}\}$ sin precisar cuál produce sus resultados principales, por lo que la reproducción se ejecuta con ambos (sección 4.2).

Tabla A.2: Parámetros de muestreo, pseudo-secuencia y optimización. La malla de colocación depende del modelo; el resto es común. El paso temporal Δt solo afecta a PINNsFormer (parámetro de su pseudo-secuencia).

Parámetro	Valor
Optimizador	L-BFGS, búsqueda lineal *strong Wolfe*, lr $= 1.0$
Iteraciones	1000 (todos los modelos)
Pesos de la pérdida	$\lambda_{\text{res}} = \lambda_{\text{ic}} = \lambda_{\text{bc}} = 1$
Malla de colocación (*baselines*)	101×101 ($N_{ic} = N_{bc} = 101$)
Malla de colocación (PINNsFormer)	51×51 ($N_{ic} = N_{bc} = 51$)
Malla de test	101×101 (común)
Longitud de pseudo-secuencia	$k = 5$ (PINNsFormer)
Paso de pseudo-secuencia	$\Delta t \in \{10^{-3}, 10^{-4}\}$ (PINNsFormer)
Muestreo de Navier–Stokes	2500 puntos del dominio (x, y, t)

A.3. Constantes de las ecuaciones

La tabla A.3 fija las constantes, los dominios y las condiciones de cada uno de los cuatro problemas, con los valores del apéndice B del artículo. El coeficiente de la ecuación de onda merece una aclaración. El artículo lo escribe como $\beta = 3$, pero la solución analítica que él mismo proporciona solo satisface la ecuación con coeficiente $C = 4$; el valor 3 corresponde en realidad al modo de la condición inicial. Se adopta $C = 4$, coherente con la solución publicada y con el código oficial (véase la discusión en la sección 4.1.2).

Tabla A.3: Ecuaciones, constantes, dominios y condiciones de los cuatro problemas de referencia (apéndice B del artículo). En Navier–Stokes la presión se evalúa en el último instante disponible.

Problema	Ecuación y dominio	Constantes / condiciones
Convección	$u_t + \beta u_x = 0$;	$\beta = 50$; IC $\sin(x)$;
	$x \in [0, 2\pi]$, $t \in [0, 1]$	BC periódica
1D-reacción	$u_t - \rho u(1-u) = 0$;	$\rho = 5$; IC gaussiana;
	$x \in [0, 2\pi]$, $t \in [0, 1]$	BC periódica
1D-onda	$u_{tt} - C\,u_{xx} = 0$;	$C = 4$; IC $\sin(\pi x) + \frac{1}{2}\sin(3\pi x)$,
	$x \in [0, 1]$, $t \in [0, 1]$	$u_t(x, 0) = 0$; BC Dirichlet
Navier–Stokes	$u_t + \lambda_1(u \cdot \nabla u) + \nabla p - \lambda_2 \Delta u = 0$	$\lambda_1 = 1$, $\lambda_2 = 0.01$;
		datos del cilindro

Los ejes y valores concretos que recorre el estudio de ablación (variaciones de d_{model}, cabezas, N, d_{ff}, d_{hidden}, k, Δt, malla e iteraciones alrededor del punto central de esta tabla) se detallan en la tabla 5.1 del capítulo 5 y no se repiten aquí.

Apéndice B

Reproducibilidad. Código, entorno y ejecución

La reproducibilidad no es un añadido de este trabajo, sino la condición que da sentido a sus comparaciones, porque solo si una misma configuración devuelve el mismo número puede atribuirse una diferencia entre dos ejecuciones al factor que las distingue y no al azar del entrenamiento. Este apéndice describe la infraestructura que garantiza esa propiedad (el punto de entrada único, el control del determinismo y el entorno fijado) y detalla cómo relanzar cada experimento de la memoria.

B.1. Punto de entrada único

Toda ejecución, sea en la nube o en un cuaderno interactivo, pasa por un mismo programa que fija las semillas, construye el modelo y el problema a partir de la configuración del apéndice A, entrena con L-BFGS, evalúa las métricas rMAE y rRMSE, genera las figuras de solución y vuelca un registro con la configuración completa y los resultados. Ese registro se guarda como un fichero por ejecución y, en paralelo, se sube a una plataforma de seguimiento de experimentos. La lógica de entrenamiento vive por entero en ese programa, de modo que ni el cuaderno ni la imagen de contenedor contienen cálculo propio, de modo que la paridad entre entornos es una propiedad del diseño y no una coincidencia.

Sobre ese entrypoint se apoyan dos orquestadores. El primero recorre la matriz completa de la reproducción (los cuatro problemas por los cuatro modelos) en una sola ejecución; el segundo expande el estudio de ablación siguiendo el protocolo de un factor cada vez a partir de un diseño declarativo, sin duplicar los valores de referencia, que se toman de la configuración central. Ambos agrupan sus ejecuciones bajo una marca de tiempo, de forma que ningún intento sobrescribe a otro.

B.2. Control del determinismo y entorno

El determinismo se apuntala fijando todas las fuentes de aleatoriedad (los generadores de la biblioteca numérica y del *framework* de aprendizaje profundo, tanto en CPU como en GPU) y forzando el modo determinista de las rutinas de cálculo subyacentes. Las versiones de las dependencias se anclan al entorno con el que se obtuvieron las cifras publicadas (tabla B.1) para minimizar las diferencias atribuibles al *software*. Con estas garantías, una misma semilla produce el mismo número en la nube y en el cuaderno interactivo.

Tabla B.1: Entorno de ejecución fijado para la reproducibilidad.

Componente	Versión / valor
Framework de aprendizaje profundo	2.0.1 (CUDA 11.7, cuDNN 8)
Biblioteca numérica	1.22.4
Modo determinista de las rutinas de cálculo	activado
Semillas	fijadas (numérica y *framework*, CPU y GPU)
GPU de la reproducción	NVIDIA RTX A6000
GPU de la ablación	clase A100 (precisión doble por hardware)

La prueba de que este control funciona es directa y aparece en el propio capítulo 4, en el que las repeticiones independientes de cada configuración, lanzadas en *pods* distintos y en momentos distintos, devolvieron métricas idénticas hasta la última cifra significativa. Esa coincidencia byte a byte es lo que convierte cualquier diferencia entre configuraciones (entre un paso temporal y otro, o entre una precisión y otra) en una señal atribuible a ese factor.

B.3. Persistencia y recuperación de resultados

El camino principal de cómputo es una imagen de contenedor que se lanza como un *pod* de GPU bajo demanda. El coste se acota por partida doble. El contenedor se autoapaga al terminar pase lo que pase, y el lanzador espera a que la ejecución figure como concluida y termina el *pod* por su cuenta. Como cada ejecución sube su registro y sus figuras nada más completarse, la caída de un *pod* a mitad de camino no destruye lo ya calculado, pues una utilidad autónoma descarga después al repositorio local todo lo que se llegó a completar, incluidas las ejecuciones parciales, y reconstruye el informe agregado.

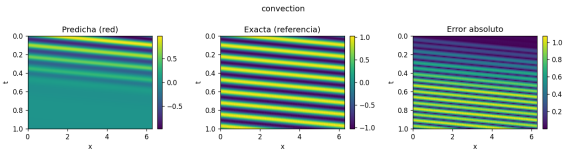
Apéndice C

Galería de figuras de soluciones

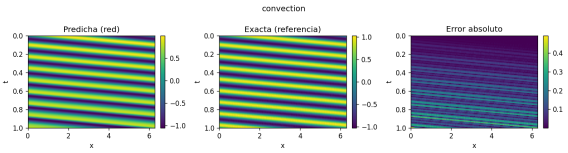
Cada ejecución del proyecto experimental genera, además de sus métricas, una figura de tres paneles con la solución predicha por el modelo, la solución de referencia y el mapa de error absoluto entre ambas. Estas figuras son útiles porque hacen visible lo que una cifra de error resume, de forma que un error relativo próximo a la unidad se corresponde con un panel de error que satura todo el dominio (la predicción no se parece a la solución), mientras que un error de orden 10^{-2} deja un mapa de error casi plano. Dos de ellas se han incorporado ya al cuerpo de la memoria para ilustrar hallazgos concretos, como el rescate de la convección al pasar a precisión doble (figura 4.2) y la debilidad de PINNsFormer en la onda (figura 5.3).

Esta galería reúne el resto de figuras de la reproducción en **precisión doble** con $\Delta t = 10^{-4}$ (la configuración que recupera las cifras publicadas de PINNsFormer sobre los tres problemas unidimensionales) para los cuatro modelos. Los errores relativos que acompañan a cada panel son los de las tablas 4.9 y 4.10. Se incluye además una lámina de Navier–Stokes (figura C.4) con el campo de presión de los cuatro modelos; su error relativo, recogido en la tabla 4.11, queda por encima de la unidad a causa de la indeterminación aditiva de la presión, de modo que la lámina ilustra esa limitación más que un acierto (aunque PINNsFormer sea el de menor error).

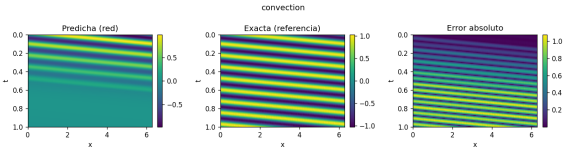
Leídas en conjunto, las cuatro láminas condensan el argumento del capítulo 4. En convección y reacción, PINNsFormer y los *baselines* bien optimizados comparten un mapa de error plano; los fracasos que quedan (QRes en reacción, la mediocridad general en la onda) se ven a simple vista en sus paneles de error.



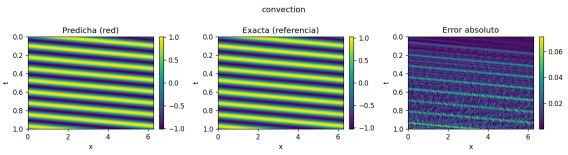
PINN: rMAE 0.713



FLS: rMAE 0.135

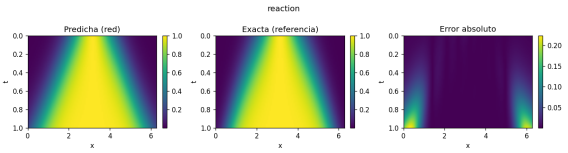


QRes: rMAE 0.682

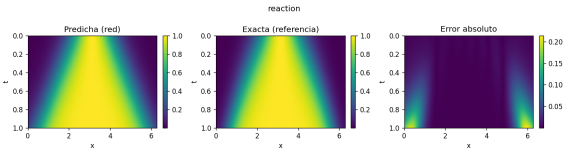


PINNsFormer: rMAE 0.022

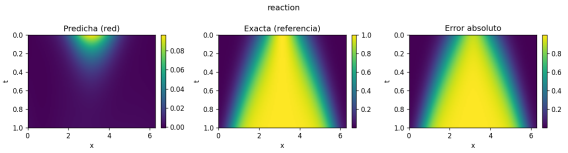
Figura C.1: Convección en precisión doble ($\Delta t = 10^{-4}$). Solución predicha, exacta y error absoluto de los cuatro modelos. Solo PINNsFormer y FLS resuelven el problema de transporte; PINN y QRes se quedan en una predicción trivial.



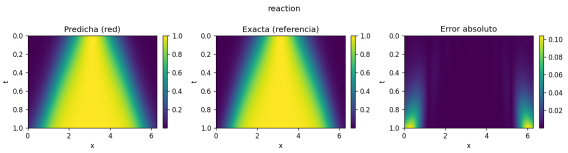
PINN: rMAE 0.035



FLS: rMAE 0.036

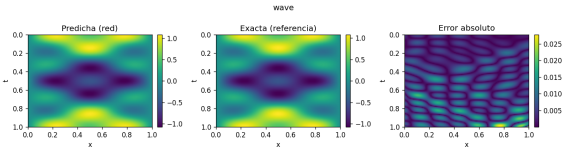


QRes: rMAE 0.988

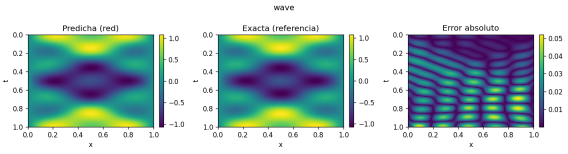


PINNsFormer: rMAE 0.013

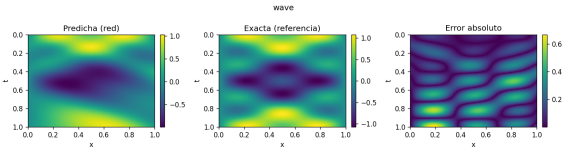
Figura C.2: 1D-reacción en precisión doble ($\Delta t = 10^{-4}$). PINN, FLS y PINNsFormer resuelven el problema; solo QRes fracasa, y lo hace con independencia de la precisión, lo que apunta a una limitación genuina de esa variante y no a un artefacto numérico.



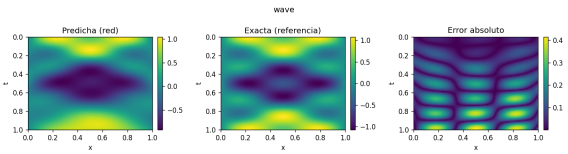
PINN: rMAE 0.011



FLS: rMAE 0.030

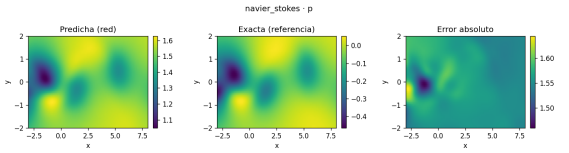


QRes: rMAE 0.346

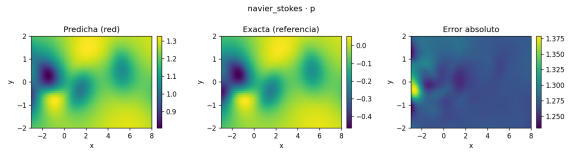


PINNsFormer: rMAE 0.229

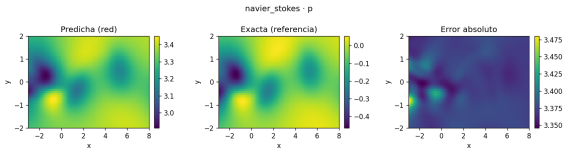
Figura C.3: 1D-onda en precisión doble ($\Delta t = 10^{-4}$). La PINN vainilla y FLS reconstruyen la solución oscilatoria; PINNsFormer queda un orden de magnitud por detrás, la peor de las cuatro arquitecturas sobre este problema.



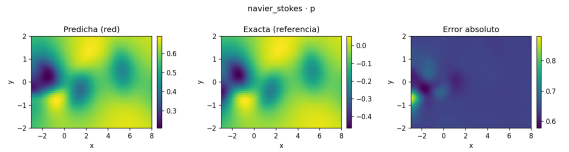
PINN: rMAE presión 17.56



FLS: rMAE presión 14.33



QRes: rMAE presión 38.01



PINNsFormer: rMAE presión 7.21

Figura C.4: Navier–Stokes en precisión doble ($\Delta t = 10^{-4}$). Campo de presión predicho, exacto y error absoluto de los cuatro modelos. El error relativo supera la unidad en todos ellos por la indeterminación aditiva de la presión (que se recupera salvo una constante); PINNsFormer es el de menor error, pero ninguno reproduce la cifra publicada.