Fast Reconstruction of Exact Maxwell Dynamics from Sparse Data

Dan DeGenaro[1] **Xin Li**[2] **Obed Amo**[3]

Michael Pokojovy[3,4] **Sarah Adel Bargal**[1] **Markus Lange-Hegermann**[5]

Bogdan Raiță[2]*

[1]Department of Computer Science, Georgetown University
{drd92, sarah.bargal}@georgetown.edu
[2]Department of Mathematics, Georgetown University
{xl572, bogdan.raita}@georgetown.edu
[3]Department of Mathematics and Statistics, Old Dominion University
{oamo, mpokojovy}@odu.edu
[4]School of Data Science, Old Dominion University
[5]Institute Industrial IT, Department of Computer Science and Automation,
OWL University of Applied Sciences and Arts
markus.lange-hegermann@th-owl.de

Abstract

We introduce *FLASH-MAX*, a shallow, exact-by-construction neural network architecture for predicting homogeneous electromagnetic fields from sparse pointwise observations. Each hidden neuron represents a separate exact solution to Maxwell's equations, so that the network satisfies the governing equations symbolically by construction and can be trained end-to-end from sparse data within seconds. We prove a universal approximation result showing that this exact model class remains universal on arbitrary domains. *FLASH-MAX* reaches sub-1% relative validation error from about 1K sparse pointwise observations in seconds, all while maintaining a zero PDE residual, and keeps single-digit errors even for only 100 observations sampled from 3D space. These results suggest that moving governing structure from the loss into the hypothesis class can dramatically improve the trade-off between precision and optimization speed in scientific machine learning.

1 Introduction

Maxwell's equations are a system of eight partial differential equations (PDEs) that describe the electromagnetic interaction in classical physics. Because Maxwell's equations govern electromagnetic wave propagation, reconstructing a full spatiotemporal field from sparse pointwise observations is a basic task in applications such as antenna measurement, electromagnetic compatibility diagnostics, RF sensing, subsurface exploration, tomography, and fusion plasma diagnostics. In such settings, the available data do not define a fully specified forward problem, so the task is not standard forward simulation but rather reconstruction of fields. We build the structure of Maxwell's equations directly into our model architecture and show that this can make learning both more efficient and more accurate, as opposed to enforcing admissibility approximately through the loss. In the sparse-reconstruction benchmarks considered here, the resulting training procedure reaches the target validation accuracy orders of magnitude faster than the reference implementations.

Preprint.

We propose *FLASH-MAX*, a Fast-Learning Accurate Solver for the Homogeneous MAXwell equations. *FLASH-MAX* is a *shallow neural network whose hidden neurons are themselves exact solutions of Maxwell's equations*, enabled by a suitable weight-sharing parameterization. This ensures that the field predicted by this network remains consistent with Maxwell's equations throughout training and inference. Hence, *FLASH-MAX* builds the governing structure of Maxwell's equations directly into the model architecture rather than relying on the loss to enforce approximate physical validity. We emphasize two aspects of this construction. First, *FLASH-MAX* only addresses the *homogeneous*, source-free Maxwell system with constant material parameters. Second, exactness here refers to *membership* in the Maxwell solution space, not to the accuracy of the particular solution recovered from sparse data: since the initial and boundary observations are fitted through a data loss rather than enforced symbolically, reconstruction quality remains an ordinary data-fitting problem that depends on observation coverage, network capacity, and optimization.

Among learning-based approaches, Physics-Informed Neural Networks (PINNs) are the closest comparison family, since they also fit PDE-constrained models directly from data while enforcing governing equations during optimization [1]. Ehrenpreis-Palamodov Gaussian Processes (EPGPs) are also conceptually relevant because they embed field structure directly into a Gaussian Process covariance model [2]. Numerical solvers and neural operators, by contrast, are designed for fully specified, well-posed forward problems with dense or complete boundary/initial data, and do not directly target reconstruction from sparse, possibly incomplete pointwise observations. Section 3 discusses these and further related lines of work, including hard-constrained and exact-solution-space methods, in more detail.

The exact model class remains expressive for the homogeneous system: Theorem 1 shows that it is universal on arbitrary domains. So, exact hidden-unit admissibility need not force the model into an overly rigid approximation class. The homogeneous Maxwell setting therefore gives a mathematically explicit regime in which exact solution-class parameterizations remain expressive and support very rapid sparse-data reconstruction of electromagnetic fields. Empirically, *FLASH-MAX* attains sub-1% relative validation error from about 1K sparse observations in a matter of seconds on representative and diverse sparse-reconstruction tasks while maintaining a zero PDE residual by construction (Table 1 and Figure 2). Compared with residual-based PINN training, this threshold is reached substantially sooner and more robustly. These results suggest that exact solution-class parameterizations can change the trade-off between physical validity, expressive power, and optimization speed. We summarize our contributions as follows:

- An exact-by-construction shallow neural architecture for reconstruction of solutions to the homogeneous Maxwell equations from sparse observations, with hidden units that are themselves exact solutions to Maxwell's equations (Figure 1 and Section 2);
- A universality theorem in general domains for this exact model class (Theorem 1);
- Empirical evidence of fast sparse-data reconstruction with sub-1% error and zero PDE residual from 1K data points, and single-digit error from only 100 data points (Section 4);
- Empirical and conceptual evidence that moving governing structure from the loss into the hypothesis class can improve the trade-off between physical admissibility and time to useful accuracy when the admissible geometry is explicit.

2 Method

We first motivate the idea behind *FLASH-MAX* as it might be applied to a much simpler linear PDE, the homogeneous transport equation in one spatial dimension. If we write $f(t,x) = \sigma(z_0 t + z_1 x + b)$ for a typical neuron in a 2D shallow neural network, then the transport equation yields

$$\begin{aligned} 0 = \partial_t f(t,x) - \partial_x f(t,x) &= z_0 \cdot \sigma'(z_0 t + z_1 x + b) - z_1 \cdot \sigma'(z_0 t + z_1 x + b) \\ &= (z_0 - z_1) \cdot \sigma'(z_0 t + z_1 x + b). \end{aligned}$$

Hence, $f(t,x) = \sigma(z_0 t + z_1 x + b)$ is a solution whenever $z_0 = z_1$. Thus, in this simple case, the PDE can be enforced exactly by imposing an algebraic constraint on the weights in the activation σ. We now extend this idea in several directions, such that it is applicable to electromagnetic phenomena described by Maxwell's equations.

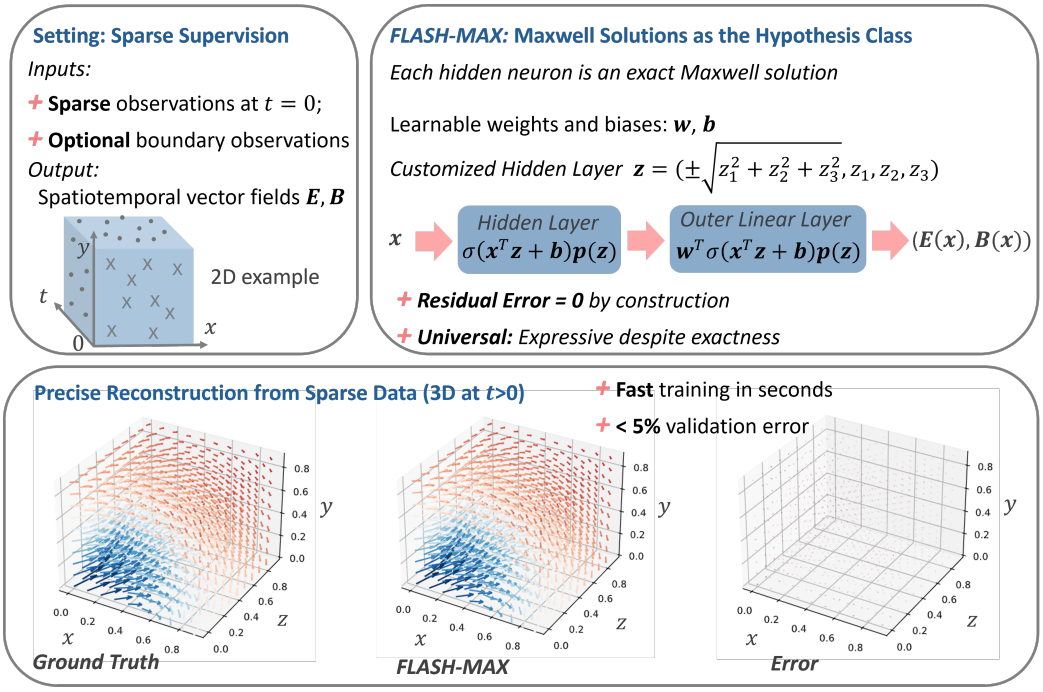

Figure 1: **Sparse-data Maxwell Reconstruction with Exact Hidden-unit Structure.** From sparse pointwise observations of a single field instance, *FLASH-MAX* fits a shallow trainable architecture whose hidden neurons are exact Maxwell solutions, so the predicted field solves Maxwell's system throughout optimization. The resulting model is exact by construction, expressive on arbitrary domains, and yields accurate reconstruction from sparse data. Exactness is ensured by using computer algebra to obtain the hidden layer restriction $z_0^2 = z_1^2 + z_2^2 + z_3^2$ and the polynomials $\boldsymbol{p}(\boldsymbol{z})$ which tie the vectorial weights. Top-left: Sparse data sampled only at $t = 0$ ("$\times$" symbols), showing that our model is predictive; data can also be sampled on the boundary of the validation domain ("$\bullet$" symbols). Bottom: Quiver plot visualization of our fast prediction of an electromagnetic knot (Hopfion) at time $t = 0.5$. The difference to ground truth is indistinguishable.

1. The domain is four-dimensional, hence we consider $\sigma(z_0 t + z_1 x + z_2 y + z_3 z + b)$.
2. Instead of the linear relation between z_0 and z_1, we get the quadratic relation[1] $z_0^2 = z_1^2 + z_2^2 + z_3^2$, which we build into our architecture as a weight-sharing constraint.
3. We use the linearity of the PDE system to write f as a linear combination: $\sum_k w_k \sigma(z_{k0} t + z_{k1} x + z_{k2} y + z_{k3} z + b_k)$.
4. To describe electric and magnetic *vector* fields, we provide two Noetherian multipliers $\boldsymbol{p}_i$, $i \in \{1, 2\}$, one of which is introduced into each summand of the linear combination.
5. We view this as a shallow neural network, where we train the parameters $w_k, \boldsymbol{z}_k, b_k$ (subject to the relation from item 2 above) on data.

More formally, let $\boldsymbol{x} = (t, x, y, z) \in \mathbb{R}^4$ be the coordinates of spacetime. We consider, respectively, electric and magnetic fields $\boldsymbol{E}, \boldsymbol{B} : \mathbb{R}^4 \to \mathbb{R}^3$, with $\boldsymbol{B}$ sometimes denoted by $\boldsymbol{H}$ in the literature, satisfying the homogeneous Maxwell equations

$$\partial_t \boldsymbol{E} - \nabla \times \boldsymbol{B} = \boldsymbol{0}, \quad \partial_t \boldsymbol{B} + \nabla \times \boldsymbol{E} = \boldsymbol{0}, \quad \nabla \cdot \boldsymbol{E} = 0, \quad \nabla \cdot \boldsymbol{B} = 0. \tag{MAX}$$

We construct an efficient solver for Initial and Initial-Boundary Value Problems using the ansatz:

$$(\boldsymbol{E}(\boldsymbol{x}), \boldsymbol{B}(\boldsymbol{x})) = \sum_{i=1}^{2} \sum_{j=\pm 1} \sum_{k=1}^{W} w_{ijk} \sigma(\boldsymbol{x}^\top \boldsymbol{z}_{ijk} + b_{ijk}) \boldsymbol{p}_i(\boldsymbol{z}_{ijk}) \tag{FLASH-MAX}$$

[1]This relation can also be explained via the connection between Maxwell's equations and the wave equation; equivalently, it is the characteristic variety of both the (1+3)D wave equation and the Maxwell system.

which is a shallow neural network of width W, customized as follows:

1. In the frequency vectors $\boldsymbol{z}_{ijk} = (z_{ijk0}, z_{ijk1}, z_{ijk2}, z_{ijk3}) \in \mathbb{R}^4$, the time weight z_{ijk0} of the time variable is tied to the spatial weights by the algebraic formula
$$z_{ijk0} = j\sqrt{z_{ijk1}^2 + z_{ijk2}^2 + z_{ijk3}^2} \quad \text{where } j = \pm 1. \tag{1}$$
2. The $\boldsymbol{p}_i \colon \mathbb{R}^4 \to \mathbb{R}^6$ are Noetherian multipliers [3–6] indexed by $i = 1, 2$ in (*FLASH-MAX*), computed via the computer algebra system Macaulay2 [7–13] as
$$\begin{aligned} \boldsymbol{p}_1(\boldsymbol{z}) &= (-z_1 z_3, -z_2 z_3, z_0^2 - z_3^2, -z_0 z_2, z_0 z_1, 0) \\ \boldsymbol{p}_2(\boldsymbol{z}) &= (z_1 z_2, -z_0^2 + z_2^2, z_2 z_3, -z_0 z_3, 0, z_0 z_1). \end{aligned} \tag{2}$$
3. The real numbers b_{ijk} are biases and the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is Lipschitz continuous and not a polynomial. In our experiments, we use the activation function $\sigma = \tanh$ and view z_{ijk1}, z_{ijk2}, z_{ijk3}, w_{ijk}, and b_{ijk} as learnable real parameters for all i, j, k that can be trained by stochastic gradient descent methods.

Since the ansatz in (*FLASH-MAX*) is obtained by means of computational algebra relying on the Ehrenpreis–Palamodov theorem [14–17], we can infer that every expression (*FLASH-MAX*) satisfies Maxwell's equations (MAX), see Lemma 4 in Appendix D. It is also worth noting that *FLASH-MAX* only requires a few hyperparameters. For additional details on *FLASH-MAX*, see Appendix B.1. A natural concern with hard architectural constraints is the loss of expressive power; if every neuron must satisfy Maxwell's equations exactly, the resulting hypothesis class could be too rigid for sparse-data fitting. The next theorem addresses this concern by proving that every solution of the homogeneous Maxwell equations can be approximated locally by *FLASH-MAX*.

Theorem 1 (Universal approximation for Maxwell's equations)**.** *Let $\sigma \colon \mathbb{R} \to \mathbb{R}$ be locally Lipschitz and non-polynomial, $t_1 < t_2$, and $\Omega \subset \mathbb{R}^3$ be a bounded open set. Let $(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}})$ be an L^2-solution of the homogeneous Maxwell equations* (MAX) *on $\mathbb{R}^4$. Then, for every $\varepsilon > 0$, there exists a finite Maxwell network solution $(\boldsymbol{E}, \boldsymbol{B})$ of the form* (*FLASH-MAX*) *such that*
$$\|(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}}) - (\boldsymbol{E}, \boldsymbol{B})\|_{L^2((t_1,t_2)\times\Omega)} < \varepsilon.$$

The approximation holds true with mild restrictions on σ even after replacing the space L^2 with spaces of continuous and smooth functions (C^0, C^m, C^∞) or Sobolev spaces H^m for $m > 0$. This theorem and its extensions are proved and explored in detail in Appendix D. The proof shows that *FLASH-MAX* can approximate the ground truth $(\boldsymbol{E}_{\mathrm{gt}}(t_1, \cdot), \boldsymbol{B}_{\mathrm{gt}}(t_1, \cdot))$ at initial time $t = t_1$ with arbitrarily small L^2-norm. It further establishes that the error at $t = t_1$ controls the spacetime L^2-error in the time span (t_1, t_2) as a consequence of the finite speed of propagation. For solving initial value problems this implies that reaching small L^2-error in *training* at the single time $t = t_1$ implies achieving L^2-error for *validation* in the time span (t_1, t_2) of the same order of magnitude.

3 Related Work

From an ML perspective, *FLASH-MAX* is related both to coordinate-based implicit neural representations such as SIREN and Fourier-feature networks [18, 19] and to random-feature or extreme-learning-machine models [20–22]; unlike both, the trainable features of *FLASH-MAX* are constrained to lie in the solution space of Maxwell's equations.

Physics-Informed Neural Networks (PINNs) solve PDEs mesh-free by embedding governing equations, initial and boundary conditions directly into the loss function. Since the original paper [1], many enhancements have been proposed to address spectral bias, optimization instability, and slow convergence, including Fourier-based embeddings, strict boundary enforcement, adaptive temporal weighting, domain decomposition, and improved optimization strategies [19, 23–29]. Prior PINN electromagnetic studies can be broadly grouped into three classes. First, several works focus on steady-state, frequency-domain, or inverse Maxwell problems, such as magnetostatics, photonic inverse design, and parameter retrieval of electromagnetic material properties [30–32]. Second, [28, 33, 34] address heterogeneous media, where interface conditions and material jumps are critical, using strategies such as first-order Maxwell formulations, interface-aware inputs, and domain

decomposition. Third, more recent studies investigate unsteady Maxwell equations, showing that convergence-enhancing strategies such as random Fourier features, strict periodicity enforcement, and causality-aware training can substantially affect performance ([26, 35]).

Beyond classical PINNs, QPINN introduces a hybrid quantum-classical PINN for two-dimensional time-dependent Maxwell equations with additional physical constraints [36], while Evo-PINN explores evolutionary and gradient-free optimization to improve PINN training robustness [37]. Despite these advances, existing work remains largely focused on 1D or 2D settings, steady-state or inverse formulations, or specially structured heterogeneous-media problems. We therefore construct a strong in-house PINN baseline for (MAX), augmenting the original PINN framework [1] with additional physics-informed constraints such as divergence-free regularization and boundary-condition enforcement. We also compare against adapted QPINN and Evo-PINN baselines in Appendix B.3; since the in-house PINN performs strongest, it is used as the primary PINN comparator in the main text. Unlike residual-based PINNs, *FLASH-MAX* builds Maxwell structure directly into the hypothesis class: every output satisfies the homogeneous Maxwell system by construction, and training only fits sparse observations within this exact class. Thus, its modeled PDE residual vanishes symbolically, whereas a PINN residual is minimized approximately. In all reported examples, *FLASH-MAX* reaches the target validation error at least one order of magnitude faster than the PINN baseline.

Numerical Methods. The Finite Element Method (FEM) for linear Maxwell equations has evolved well beyond standard $H(\mathrm{curl})$-conforming Nédélec elements [38], addressing challenges such as spurious modes, material discontinuities, and high-frequency pollution. Recent developments include Discontinuous Galerkin frameworks, in particular IPDG and HDG methods [39], which provide flexible mesh refinement and local conservation, as well as Finite Element Exterior Calculus (FEEC) [40], which preserves the de Rham structure and the associated divergence constraints. For high-frequency time-harmonic Maxwell equations, preconditioners and multiscale methods [41] improve iterative solver efficiency, while in the time domain, structure-preserving integrators combined with adaptive mesh refinement [42] support high-fidelity electromagnetic interference and scattering simulations. Compared to alternative techniques like the Finite-Difference Time-Domain (FDTD) method, which struggles with staircasing errors on curved boundaries, or the Boundary Element Method (BEM), which is often limited to piecewise homogeneous media, FEM offers superior flexibility in handling complex, multi-material geometries and anisotropic media through unstructured tetrahedral meshes. This geometric fidelity and availability of automated libraries make FEM the preferred choice for advanced engineering applications, such as stealth technology and photonic crystal design.

FEniCS is a library frequently used for high-performance implementation of FEM. Designed as an automated programming environment, it utilizes symbolic domain-specific languages to streamline the generation of efficient $H(\mathrm{curl})$-conforming finite element kernels [43]. In typical industrial applications, mesh sizes can reach up to a billion unknowns, necessitating the use of massively parallel solvers and large computational time on high-performance computing clusters. Unlike traditional FEMs running on HPC, *FLASH-MAX* runs on a single L40S GPU in seconds. FEMs at best approach zero PDE residuals asymptotically, whereas *FLASH-MAX* has null PDE residuals intrinsically. To separate the architectural speedup from GPU acceleration, Appendix A.2 reports CPU timings for *FLASH-MAX* under the same sparse-reconstruction setup.

Neural Operators learn mappings between function spaces rather than finite-dimensional parameter vectors and have become a standard paradigm for amortized PDE surrogate modeling. Canonical examples include DeepONet [44], FNO [45], and physics-informed variants such as PINO [46]; recent reviews emphasize their ability to generalize across discretizations, but also their dependence on the geometry, representation, and physics of the target problem [47]. For electromagnetics, operator-learning approaches exist for certain regimes: frequency-domain photonic-device simulation [48, 49], free-form 3D electromagnetic scattering and inverse design [50], 2D transient wave propagation [51], time-domain photonic rollouts [52], and Fourier-space hard-constrained electrodynamics tailored to beam-driven settings [53]. These works show that neural operators are promising for electromagnetism problems, but they do not define a directly commensurate state-of-the-art for our combination of spacetime formulation, field generality, sparse observations, and exact constraints.

We are not aware of an off-the-shelf neural-operator baseline that is directly commensurate with our per-instance sparse-reconstruction setting. A from-scratch neural-operator comparison would require substantial upfront generation of accurate Maxwell data, followed by offline training before fast inference becomes possible [47, 54]. Such a comparison would therefore probe a different

computational regime—amortized surrogate learning over many problem instances—rather than the sparse-data, per-instance regime considered here.

Gaussian Processes (GPs) provide a natural Bayesian framework for linear PDEs, as linear operators preserve Gaussian structure. Early work incorporated exact linear constraints through latent parameterizations and specialized kernels, including latent force models, divergence-/curl-free constructions, LODE-GPs, and Maxwell-adjacent magnetic-field models [55–60]. More recent work extended exact-prior ideas to constant-coefficient PDE systems and boundary conditions, including EPGP and boundary-constrained constructions [2, 61–65]. As is the case for *FLASH-MAX*, these models embed the physical law in the admissible function class rather than only inform the penalty terms. In contrast to these GPs, we realize exact Maxwell structure as an architectural inductive bias in a trainable shallow network, enabling direct gradient-based fitting from sparse data rather than expensive posterior inference over a GP prior.

Hard-constrained Scientific ML and Exact-solution Spaces. Beyond residual-based training, several lines of work restrict approximation directly to physically admissible function classes [66–68]. One family enforces a linear constraint $Au = 0$ by parameterizing $u = Bv$ for a linear differential operator B and an unconstrained learned potential v, including linearly-constrained neural networks [69] and divergence-free-by-construction operator-learning methods for incompressible flow and elasticity [70–73]. This idea covers a *single* divergence-type constraint, whereas Maxwell's system (MAX) couples $\boldsymbol{E}$ and $\boldsymbol{B}$ through *six* additional curl equations.

A second family targets the time-harmonic Helmholtz reduction of Maxwell's equations with plane-wave trial functions [74, 75], addressing a single, low-dimensional, elliptic equation rather than our fully time-dependent, hyperbolic $3+1$-dimensional system. Physics-informed kernel networks take a related but more general approach, using kernels derived from PDEs as hidden units [76], though on smaller systems. The papers [77–79] use local trial functions that satisfy the governing equations by construction, and hard-constrained Maxwell architectures also exist through potential-based or Fourier-space parameterizations [53, 80].

Architecturally closest to *FLASH-MAX* is [75], which is likewise a shallow, trainable network whose hidden units are individually exact solutions. It differs from *FLASH-MAX* in three fundamental respects: (i) it fixes a single wave number per solve and optimizes only propagation directions, whereas our trainable frequency vectors range freely over the entire light cone, (ii) it discretizes the domain into mesh elements, whereas *FLASH-MAX* is a single mesh-free network trained globally over the full spacetime domain, and (iii) its plane-wave directions and linear coefficients are found by an alternating iterative quasi-minimization scheme, whereas all *FLASH-MAX* parameters (both frequencies and weights) are optimized jointly by standard gradient descent. Beyond these architectural differences, none of the methods discussed above provide a closed-form, computer-algebra-derived hidden-unit parameterization for the complete time-dependent Maxwell system: this is what *FLASH-MAX* adds to the general principle—shared with all of the above—of moving governing PDE structure from the loss into the hypothesis class.

4 Experiments

We present experiments designed to demonstrate the efficiency and accuracy of *FLASH-MAX* as a solver of Initial Value Problems for (MAX) in comparison to prior art: PINNs, FEMs, and S-EPGP[2]. We perform this comparison in two settings: with and without boundary conditions (BC and IC, respectively), whenever possible. Note that FEMs can only operate in the BC setup. We then report and discuss the experimental results for four ground truth solutions to Maxwell's equations, which we term *Plane Waves*, *Radial Waves*, *Hopf Fibration*, and *Random Solution*. These solutions are chosen to be as different from our (*FLASH-MAX*) basis and from each other as possible, demonstrating the generalization capabilities of our network (see Appendix C for details). We first describe our experimental setup and evaluation criteria and then our three experiments: 1) *Which method can reach relative L^2-accuracy of 5% fastest?* 2) *What is the accuracy each method achieves after 70s of training time?* 3) *What is the best accuracy FLASH-MAX achieves with only 100 data samples?*

We first pick a solution $\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}} \colon [0,\infty) \times (0,1)^3 \to \mathbb{R}^3$ from the list of four in Appendix C. In the **initial value problems (IC) setup** we seek to reconstruct $\boldsymbol{E}, \boldsymbol{B} \colon [0, 0.1] \times [0.2, 0.8]^3 \to \mathbb{R}^3$, solving

[2]Code for *FLASH-MAX* and all baselines discussed in this paper are available in the supplementary material.

Table 1: **Experiment 1: Race to 5%.** Validation Error, Residual Error, and Compute training time to reach 5% relative validation error (if possible) for the ground truths: *Plane Waves*, *Radial Waves*, *Hopf Fibration*, *Random Solution* with/without boundary conditions (BCs). Animations to complement these results visually are available in the supplementary material at `flash_max/flash_max_gifs`. Details concerning these experiments are available in Table 6.

	Method	BC	Val Error (%)	Residual Error	Compute (s)
Plane Waves	FEM	✓	8.22	0.052	3,000
		✗	n/a	n/a	n/a
	PINN	✓	**<5.00**	0.007	304.2 ± 73.2
		✗	**<5.00**	0.013	119.3 ± 28.5
	S-EPGP	✓	26.65	**0**	345.7 ± 69.7
		✗	25.04	**0**	61.4 ± 25.5
	FLASH-MAX (ours)	✓	**<5.00**	**0**	**8.1** ± **0.4**
		✗	**<5.00**	**0**	**6.8** ± **0.3**
Radial Waves	FEM	✓	65.47	2.002	2.9
		✗	n/a	n/a	n/a
	PINN	✓	93.9	0.063	47.4 ± 75.0
		✗	86.7	0.072	56.4 ± 34.2
	S-EPGP	✓	did not converge	–	–
		✗	5.85	**0**	89.1 ± 15.5
	FLASH-MAX (ours)	✓	**<5.00**	**0**	**58.9** ± **2.0**
		✗	**<5.00**	**0**	**11.9** ± **0.6**
Hopf Fibration	FEM	✓	**<5.00**	0.009	157
		✗	n/a	n/a	n/a
	PINN	✓	**<5.00**	0.010	235.3 ± 80.7
		✗	**<5.00**	0.008	89.8 ± 25.9
	S-EPGP	✓	37.16	**0**	410.4 ± 78.9
		✗	**<5.00**	**0**	153.3 ± 44.8
	FLASH-MAX (ours)	✓	**<5.00**	**0**	**7.4** ± **0.3**
		✗	**<5.00**	**0**	**3.2** ± **0.0**
Random Soln.	FEM	✓	**<5.00**	0.022	13,701
		✗	n/a	n/a	n/a
	PINN	✓	5.65	0.017	355.1 ± 31.1
		✗	**<5.00**	0.011	131.9 ± 19.4
	S-EPGP	✓	75.0	**0**	516.7 ± 7.7
		✗	**<5.00**	**0**	115.9 ± 2.1
	FLASH-MAX (ours)	✓	**<5.00**	**0**	**58.3** ± **3.2**
		✗	**<5.00**	**0**	**6.1** ± **0.2**

(MAX) using 1,000 data points $(x, y, z) \in [0, 1]^3$ at $t = 0$:

$$\boldsymbol{E}(0, x, y, z) = \boldsymbol{E}_{\text{gt}}(0, x, y, z) \quad \text{and} \quad \boldsymbol{B}(0, x, y, z) = \boldsymbol{B}_{\text{gt}}(0, x, y, z). \tag{IC}$$

The evaluation domain $[0, 0.1] \times [0.2, 0.8]^3$ is chosen to lie within the domain of dependence of the initial-data region $\{0\} \times [0, 1]^3$, as prescribed by the finite speed of propagation of Maxwell's equations [81, Theorem 3.12]. This local spacetime reconstruction setting is different from the standard fully specified forward problems for which FEMs are designed.

In the **initial-boundary value problems (BC) setup** we consider tangential boundary conditions

$$\boldsymbol{E}(t, x, y, z) \times \boldsymbol{n} = \boldsymbol{E}_{\text{gt}}(t, x, y, z) \times \boldsymbol{n} \quad \text{for } t \in [0, 1] \text{ and } (x, y, z) \in \partial[0, 1]^3, \tag{BC}$$

where $\boldsymbol{n}$ is the outward normal to the boundary of the cube $\partial[0, 1]^3$ (for instance, on the top of the cube, $\boldsymbol{n} = (0, 0, 1)$). We predict over a validation set sampled from $(t, x, y, z) \in [0, 1] \times [0, 1]^3$, which is consistent with the fact that the system (MAX) with initial conditions (IC) and boundary conditions (BC) is well-posed in $L^2((0, 1)^4)$. In the BC setup we sample 1,000/7 points (t, x, y, z) on each of the 7 faces of the cube $(0, 1)^4$ which are considered in (IC) and (BC).

The evaluation metric we consider is relative L^2-error to the ground truth, defined as

$$\texttt{error}_{\text{gt}} = \text{RMSE}((\boldsymbol{E}, \boldsymbol{B}), (\boldsymbol{E}_{\text{gt}}, \boldsymbol{B}_{\text{gt}}))/\text{RMSE}((\boldsymbol{E}_{\text{gt}}, \boldsymbol{B}_{\text{gt}}), \boldsymbol{0}).$$

We validate over sets sampled from $[0, 0.1] \times [0.2, 0.8]^3$ in the IC setup and from $[0, 1]^4$ in the BC setup. For the PDE residual (PINNs and FEMs), we also report the L^2-error, namely

$$\texttt{error}_{\text{PDE}} = \text{RMSE}((\partial_t \boldsymbol{E} - \nabla \times \boldsymbol{B}, \partial_t \boldsymbol{B} + \nabla \times \boldsymbol{E}, \nabla \cdot \boldsymbol{B}, \nabla \cdot \boldsymbol{E}), \boldsymbol{0}).$$

In all experiments, we optimize each method's hyperparameters (including the number of training and/or collocation points) individually and report the best-performing setups. For example, while *FLASH-MAX* performed well with 1K data points, PINNs needed 10K, and S-EPGP needed 12K. Implementation details for all methods can be found in Appendices B.1, B.2.

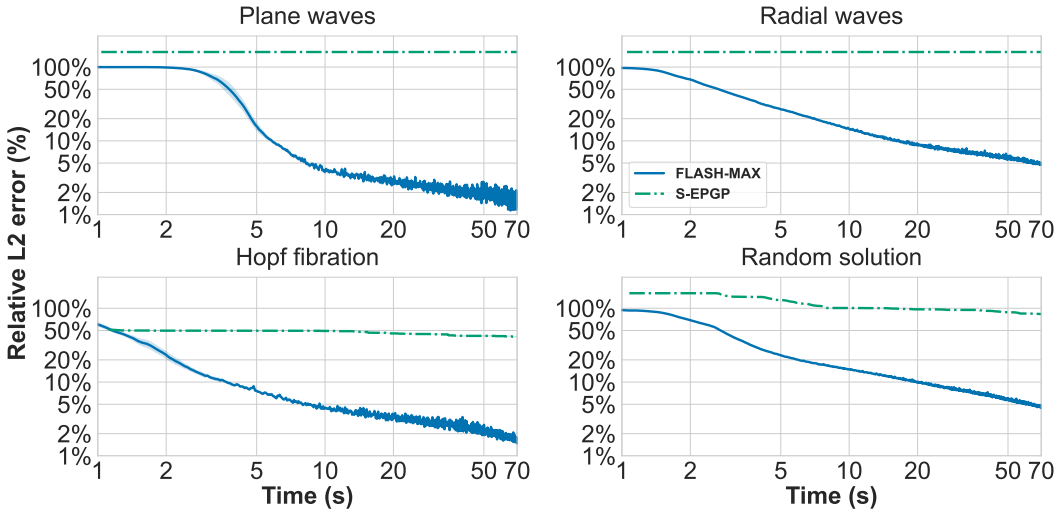

Figure 2: **Experiment 2: Performance on a Time Budget (BC setup).** The curves depict the achievable error rates (y-axis) given a specific time budget (x-axis) for the setting with boundary conditions. Uncertainties are shown (standard error of the mean over 5 random seeds). *FLASH-MAX* (ours, blue solid line) clearly converges significantly faster than S-EPGP across all four problems. FEM and PINN are excluded since they do not reach the PDE residual error threshold of 0.01 during the time budget (cf. Table 1), which is a bare minimum to claim fidelity to the physics. Plots for the IC setup can be found in the Appendix A, Figure 3 and follow a similar pattern.

The **first experiment is a race to 5%**. Given the target validation $\texttt{error}_{\text{gt}} < 5\%$, we optimize each method, stopping as soon as the target validation error is reached (if it is reached at all). We further impose the normalized residual threshold $\texttt{error}_{\text{PDE}} < 10^{-2}$, a generous threshold for normalized Maxwell equations, reflecting the standard role of residuals as indicators of physical fidelity in FEM analysis [82, 83] and neural operators [84]. In contrast, *FLASH-MAX* and S-EPGP have residual error equal to 0 by construction. For each method, we report the validation error, residual error, and compute time in both the BC and IC setups as applicable.

Table 1 presents our *Race to 5% results*, reporting validation error, residual error, and compute time (in seconds) across all methods, ground truths, and boundary condition setups. Experiments utilized different hardware: NVIDIA L40S GPU (*FLASH-MAX*, PINN), NVIDIA A100 GPU (S-EPGP), and M2 CPU (FEM), yet the comparison remains fair and underscores the algorithmic efficiency of *FLASH-MAX*. First, comparing *FLASH-MAX* and the in-house PINN on the exact same L40S GPU provides an apples-to-apples comparison against a primary machine learning baseline where our method achieves orders-of-magnitude speedup. Second, S-EPGP remains significantly slower despite utilizing a more powerful NVIDIA A100 GPU to accommodate the intensive computational requirements of Gaussian process regression. Finally, FEM ran on a CPU, which is standard for FEniCS. To eliminate GPU acceleration as a confounder, we replicated *FLASH-MAX* on the same CPU hardware (Appendix A.2). *FLASH-MAX* still reached target accuracy substantially faster than FEM, proving that the speedup is architectural rather than hardware-driven.

In the sparse-data fast-training regime discussed here, *FLASH-MAX* outperforms all competitor approaches *by one to three orders of magnitude*. Notably, *FLASH-MAX* uses identical hyperparameter settings across all ground truths and both the IC and BC setups, achieving robust performance without the need to re-tune hyperparameters on an instance-by-instance basis. PINNs and FEMs, unlike *FLASH-MAX*, do not yield identically zero residual error. PINNs are highly unstable during training, and although they sometimes converge to the 5% validation error threshold, they frequently fail to do so, even with extensive hyperparameter tuning. S-EPGPs do not reach the 5% validation error threshold in many of the analyzed setups. FEMs, which are not applicable to the IC setup, frequently require several orders of magnitude more training time to reach the 5% validation error threshold, and sometimes do not converge to the threshold at all. The **second experiment is performance on a time budget** of 70s. Figure 2 depicts the validation error $\texttt{error}_{\text{gt}}$ over time in a log-log scale. See also Figure 3 in the appendix for more comparisons. We see that our approach *FLASH-MAX* reduces the error faster and continues to consistently reduce the error during training time. FEMs and PINNs do not reach the PDE residual error threshold of 0.01 and S-EPGP shows very slow convergence, as is to be expected for a Gaussian process method.

The **third experiment is performance on a data budget**. We analyze the data requirements of *FLASH-MAX* in Table 2. We find that *FLASH-MAX* performs well even with very sparse observations, often reaching relative L^2-error below 1% with just a few hundred training pairs. In all experiments shown in Table 2, we use the IC setup shown in Table 1 (rows with ✗), varying only the number of data points used in training and fixing the random seed to 42 across runs. We find consistent, strong performance across all four ground truth solutions, with as few as 100 to 500 training observations. We refer to Appendix A for additional experiments, ablations, and sensitivity analyses.

Discussion. The experiments identify several advantages of *FLASH-MAX* within the sparse-reconstruction setting. First, *FLASH-MAX* is one to three orders of magnitude faster than prior art on almost every method-problem experiment (an end-to-end neural-operator baseline including solver-generated training data and hyperparameter tuning could require weeks to months of computation, so we did not include it as a primary comparator). Second, *FLASH-MAX* has a solution guarantee (zero residual error), like S-EPGP. PINNs and FEMs on the other hand do not have solution guarantees. Third, *FLASH-MAX* only requires sparse data (hundreds to a thousand data points vs. more than 10 thousand for other methods). Fourth, *FLASH-MAX* is able to obtain consistent performance reaching the validation error threshold without specific hyperparameter tuning for each problem separately. The same hyperparameters generalize to all four problems (Appendix B.2).

Although our in-house PINN provides the strongest performance among the PINN-style baselines considered in this work, its behavior remains highly problem-dependent and can become unstable on more challenging solution families. In particular, for the *Radial Waves* solution, the model exhibits very poor accuracy, with a relative error of 95.97% for the case with boundary conditions, together with a relatively large residual error of 0.063. Since well-trained runs in our setting typically attain residual errors around 0.01, this result indicates a clear optimization failure rather than merely a mismatch between residual minimization and solution accuracy. We also observe substantial variability in computational cost, e.g., 70.8 ± 123.6, which further reflects instability across runs.

Table 2: **Experiment 3: Performance on a Data Budget**. Sensitivity of *FLASH-MAX* to the number of training points (#pts) in the IC (initial conditions) setup. All runs use random seed 42. We write PW = *Plane Waves*, RW = *Radial Waves*, HF = *Hopf Fibration*, RS = *Random Solution*.

Min. RL2E (%)					**Time to min. RL2E (s)**				
# pts.	**PW**	**RW**	**HF**	**RS**	**# pts.**	**PW**	**RW**	**HF**	**RS**
100	2.6	7.4	< 1.0	2.0	100	520.4	138.0	33.2	137.0
200	< 1.0	2.9	1.1	< 1.0	200	61.4	144.6	14.2	67.7
500	< 1.0	< 1.0	< 1.0	< 1.0	500	19.2	164.7	11.5	50.2
1,000	< 1.0	< 1.0	< 1.0	< 1.0	1,000	39.6	105.8	12.5	53.3
2,000	< 1.0	< 1.0	< 1.0	< 1.0	2,000	35.0	86.6	19.0	57.2
5,000	< 1.0	< 1.0	1.5	< 1.0	5,000	33.2	94.5	11.3	70.5
10,000	< 1.0	< 1.0	< 1.0	< 1.0	10,000	40.4	97.3	21.9	73.1
12,000	1.3	< 1.0	< 1.0	< 1.0	12,000	37.7	109.6	27.4	71.3

The S-EPGP baseline also shows clear limitations, especially in the boundary-constrained setting. This behavior suggests that the finite-feature S-EPGP approximation is not expressive enough to simultaneously satisfy the initial and boundary constraints over the full space-time domain. In practice, the augmented training system appears to introduce strong competition between these two sources of supervision, and the resulting approximation quality deteriorates markedly across all four solutions. In addition, in the case without boundary conditions, since performance degrades substantially when extrapolating to longer time horizons, evaluation for this baseline is restricted to the interior region $[0, 0.01] \times [0.2, 0.8]^3$. We note that this is a shorter temporal interval than that used for *FLASH-MAX*. These results indicate that, although S-EPGP provides a useful statistical baseline, its applicability to the boundary-constrained 4D Maxwell setting considered here is limited.

The FEM baseline exhibits a different limitation from the learning-based baselines. As a structure-preserving discretization, its behavior depends directly on mesh resolution, polynomial degree, and solution regularity. In the degree-2 setup reported in Table 1, FEM reaches the validation target for the *Hopf Fibration* and *Random Solution* cases, while *Plane Waves* remain above the 5% target. The *Radial Waves* case is the most challenging FEM result, since further mesh refinement increased the validation error, consistent with a large near-origin feature of the spherical-pulse solution on $[0, 1]^3$.

Limitations. Our work focuses on the homogeneous, source-free Maxwell system with constant material parameters. The exactness guarantee of *FLASH-MAX* relies on this structure: incorporating currents, charges, conductivity, heterogeneous/dispersive media, and interface conditions would require additional exact parametrizations or constraint mechanisms. Moreover, while *FLASH-MAX*'s predictions satisfy the governing PDEs by construction, the initial and boundary observations are imposed through data fitting rather than symbolic enforcement. Thus, the accuracy of the recovered field also depends on data samples, finite network size, and nonconvex optimization strategy.

FLASH-MAX does not natively provide calibrated uncertainty: training proceeds via gradient descent rather than posterior inference, unlike S-EPGP, whose frequencies are drawn from a fixed reference measure and whose Bayesian linear-Gaussian structure yields a closed-form predictive covariance. This stems from the fact that *FLASH-MAX*'s hidden-unit frequencies $\boldsymbol{z}_{ijk}$ are trained subject to the algebraic constraint (1), whereas the linear readout weights w_{ijk} enter the model as in a finite-feature Bayesian linear model. A potential direction is therefore a two-stage procedure: train *FLASH-MAX* as usual to fit the exact-solution frequencies to the sparse data, then freeze the resulting exact-Maxwell feature map and place a Gaussian prior over the linear weights alone, yielding a closed-form Gaussian posterior predictive from a finite, Maxwell-admissible "weight-space" GP. Ensembling over multiple random frequency initializations offers a simpler, complementary alternative. We leave a full development of this direction, and comparison to kernel-based property-preserving operator learning methods that provide automatic GP uncertainty [71], to future work.

5 Conclusions

In this work, we introduced *FLASH-MAX*, an exact-by-construction architecture for reconstructing homogeneous electromagnetic fields from sparse observations. By embedding Maxwell's equations directly into the model class (*FLASH-MAX*), all predictions satisfy the governing physics identically, enabling fast training and accurate reconstruction from limited data. Empirically, the method achieves sub-1% error in seconds while guaranteeing a zero PDE residual, substantially improving optimization speed over residual-based approaches such as PINNs, see Tables 1, 2.

Theoretically, we showed in Theorem 1 that this exact solution class remains expressive via a universality result in arbitrary domains, demonstrating that strict physical admissibility need not limit approximation power. Overall, these results indicate that moving governing structure from the loss into the hypothesis class can improve the trade-off between physical consistency and computational efficiency, suggesting a broader design principle for scientific machine learning. Adding to prior exact-solution-space methods that address a single divergence-type constraint or reduction of Maxwell's equations (Section 3), *FLASH-MAX*'s specific contribution is a closed-form hidden-unit parameterization for the complete, time-dependent, eight-equation Maxwell system (MAX).

Acknowledgments and Disclosure of Funding

This work was funded by Georgetown University and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — Project Number 576384230. This work was supported by the German Federal Ministry of Research, Technology and Space (BMFTR) within the project KI-OWL under grant No. 01IS24057D. This work was supported by the SAIL project which is funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia under the grant no NW21-059C. This material is based upon work supported by the National Science Foundation under Award No. 2555785. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] 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. doi: 10.1016/j.jcp.2018.10.045.

[2] Marc Härkönen, Markus Lange-Hegermann, and Bogdan Raiţă. Gaussian Process Priors for Systems of Linear Partial Differential Equations with Constant Coefficients. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 12587–12615. PMLR, 2023. URL `https://proceedings.mlr.press/v202/harkonen23a.html`.

[3] Justin Chen, Yairon Cid-Ruiz, Marc Härkönen, Robert Krone, and Anton Leykin. Noetherian Operators in Macaulay2. *Journal of Software for Algebra and Geometry*, 12(1):33–41, 2022. doi: 10.2140/jsag.2022.12.33.

[4] Ulrich Oberst. The Construction of Noetherian Operators. *Journal of Algebra*, 222(2):595–620, 1999. doi: 10.1006/jabr.1999.8035.

[5] Alberto Damiano, Irene Sabadini, and Daniele C. Struppa. Computational Methods for the Construction of a Class of Noetherian Operators. *Experimental Mathematics*, 16(1):41–53, 2007. doi: 10.1080/10586458.2007.10128986.

[6] Katsusuke Nabeshima and Shinichi Tajima. Effective Algorithm for Computing Noetherian Operators of Zero-Dimensional Ideals. *Applicable Algebra in Engineering, Communication and Computing*, 33(6):867–899, 2022. doi: 10.1007/s00200-022-00570-7.

[7] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a Software System for Research in Algebraic Geometry, 2023. URL `https://macaulay2.com`.

[8] Rida Ait El Manssour, Marc Härkönen, and Bernd Sturmfels. Linear PDE with Constant Coefficients. *Glasgow Mathematical Journal*, 65(S1):S2–S27, 2023. doi: 10.1017/S0017089521000355.

[9] Yairon Cid-Ruiz and Bernd Sturmfels. Primary Decomposition with Differential Operators. *International Mathematics Research Notices*, 2023(14):12119–12147, 2023. doi: 10.1093/imrn/rnac178.

[10] Justin Chen and Yairon Cid-Ruiz. Primary Decomposition of Modules: A Computational Differential Approach. *Journal of Pure and Applied Algebra*, 226(10):107080, 2022.

[11] Yairon Cid-Ruiz, Roser Homs, and Bernd Sturmfels. Primary Ideals and Their Differential Equations. *Foundations of Computational Mathematics*, 21(5):1363–1399, 2021. doi: 10.1007/s10208-020-09485-6.

[12] Justin Chen, Marc Härkönen, Robert Krone, and Anton Leykin. Noetherian operators and primary decomposition. *Journal of Symbolic Computation*, 110:1–23, 2022. doi: 10.1016/j.jsc.2021.09.002.

[13] Yairon Cid-Ruiz. Noetherian operators, primary submodules and symbolic powers. *Collectanea Mathematica*, 72:175–202, 2021. doi: 10.1007/s13348-020-00285-3.

[14] Leon Ehrenpreis. *Fourier analysis in several complex variables*. Wiley, 1970.

[15] V. P. Palamodov. *Linear differential operators with constant coefficients*. Springer, 1970.

[16] Lars Hörmander. *An Introduction to Complex Analysis in Several Variables*, volume 7 of *North-Holland Mathematical Library*. North-Holland, 3 edition, 1990.

[17] J.-E. Björk. *Rings of Differential Operators*, volume 21 of *North-Holland Mathematical Library*. North-Holland, 1979. ISBN 9780444852922.

[18] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[19] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[20] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

[21] Yoh-Han Pao, Gwang Hoon Park, and Dejan J. Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2):163–180, 1994. doi: 10.1016/0925-2312(94)90053-1.

[22] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3):489–501, 2006. doi: 10.1016/j.neucom.2005.12.126.

[23] 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 *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 2019.

[24] 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, 2021. doi: 10.1137/20M1318043.

[25] 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*, volume 34, pages 26548–26560, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/df438e5206f31600e6ae4af72f2725f1-Abstract.html`.

[26] Pan Zhang, Yanyan Hu, Yuchen Jin, Shaogui Deng, Xuqing Wu, and Jiefu Chen. A Maxwell's Equations Based Deep Learning Method for Time Domain Electromagnetic Simulations. *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, 6:35–40, 2021.

[27] Pratanu Roy and Stephen T. Castonguay. Exact Enforcement of Temporal Continuity in Sequential Physics-Informed Neural Networks. *Computer Methods in Applied Mechanics and Engineering*, 430:117197, 2024. doi: 10.1016/j.cma.2024.117197.

[28] Shiyuan Piao, Hong Gu, Aina Wang, and Pan Qin. A Domain-Adaptive Physics-Informed Neural Network for Inverse Problems of Maxwell's Equations in Heterogeneous Media. *IEEE Antennas and Wireless Propagation Letters*, 23(10):2905–2909, 2024. doi: 10.1109/LAWP.2024.3413851.

[29] Jia Guo, Haifeng Wang, Shilin Gu, and Chenping Hou. TCAS-PINN: Physics-Informed Neural Networks with a Novel Temporal Causality-Based Adaptive Sampling Method. *Chinese Physics B*, 33(5):050701, 2024. doi: 10.1088/1674-1056/ad21f3.

[30] Yuan Zhang, Renxian Li, Huan Tang, Zhuoyuan Shi, Bing Wei, Shuhong Gong, Lixia Yang, and Bing Yan. Electromagnetic Scattering from a Three-Dimensional Object Using Physics-Informed Neural Network. *Applied Computational Electromagnetics Society Journal*, 40(2):103–111, feb 2025. doi: 10.13052/2025.ACES.J.400203.

[31] Yuyao Chen and Luca Dal Negro. Physics-Informed Neural Networks for Imaging and Parameter Retrieval of Photonic Nanostructures from Near-Field Data. *APL Photonics*, 7(1):010802, 2022. doi: 10.1063/5.0072969.

[32] Y. Deng, K. Fan, B. Jin, J. Malof, and W. J. Padilla. Physics-Informed Learning in Artificial Electromagnetic Materials. *Applied Physics Reviews*, 12(1):011331, 2025. doi: 10.1063/5.0232675.

[33] Michel Nohra and Steven Dufour. Physics-Informed Neural Networks for the Numerical Modeling of Steady-State and Transient Electromagnetic Problems with Discontinuous Media. *arXiv preprint arXiv:2406.04380*, 2024.

[34] Michel Nohra and Steven Dufour. Approximating Electromagnetic Fields in Discontinuous Media Using a Single Physics-Informed Neural Network. *arXiv preprint arXiv:2407.20833*, 2024.

[35] Gal G. Shaviner, Hemanth Chandravamsi, Shimon Pisnoy, Ziv Chen, and Steven H. Frankel. PINNs for Solving Unsteady Maxwell's Equations: Convergence Issues and Comparative Assessment with Compact Schemes. *Neural Computing and Applications*, 37(29):24103–24122, 2025. doi: 10.1007/s00521-025-11554-2.

[36] Ziv Chen, Gal G. Shaviner, Hemanth Chandravamsi, Shimon Pisnoy, Steven H. Frankel, and Uzi Pereg. Quantum Physics-Informed Neural Networks for Maxwell's Equations: Circuit Design, "Black Hole" Barren Plateaus Mitigation, and GPU Acceleration. *Quantum Machine Intelligence*, 8(1):21, 2026. doi: 10.1007/s42484-026-00365-w.

[37] Jian Cheng Wong, Abhishek Gupta, Chin Chun Ooi, Pao-Hsiung Chiu, Jiao Liu, and Yew-Soon Ong. Evolutionary Optimization of Physics-Informed Neural Networks: Evo-PINN Frontiers and Opportunities. *IEEE Computational Intelligence Magazine*, 21(1):16–36, 2026. doi: 10.1109/MCI.2025.3607749.

[38] Jean-Claude Nédélec. Mixed Finite Elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35(3):315–341, 1980.

[39] Ngoc Cuong Nguyen, Jaime Peraire, and Bernardo Cockburn. Hybridizable Discontinuous Galerkin Methods for the Time-Harmonic Maxwell Equations. *Journal of Computational Physics*, 230(19):7151–7175, 2011.

[40] Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Finite Element Exterior Calculus, Homological Techniques, and Applications. *Acta Numerica*, 15:1–155, 2006.

[41] Björn Engquist and Lexing Ying. Sweeping Preconditioner for the Helmholtz Equation: Hierarchical Matrix Representation. *Communications on Pure and Applied Mathematics*, 64(5):697–735, 2011. doi: 10.1002/cpa.20358.

[42] Wolfgang Bangerth and Rolf Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Basel, Basel, 2003.

[43] Anders Logg, Kent-Andre Mardal, and Garth Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin, Heidelberg, 2012. ISBN 978-3-642-23099-8. doi: 10.1007/978-3-642-23099-8.

[44] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

[45] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=c8P9NQVtmnO`.

[46] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-Informed Neural Operator for Learning Partial Differential Equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506.

[47] Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural Operators for Accelerating Scientific Simulations and Design. *Nature Reviews Physics*, 6:320–328, 2024. doi: 10.1038/s42254-024-00712-5.

[48] Jiaqi Gu, Zhengqi Gao, Chenghao Feng, Hanqing Zhu, Ray T. Chen, Duane Boning, and David Z. Pan. NeurOLight: A Physics-Agnostic Neural Operator Enabling Parametric Photonic Device Simulation. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/hash/5ddfb189c022a317ff1c72e6639079de-Abstract-Conference.html`.

[49] Hanqing Zhu, Wenyan Cong, Guojin Chen, Shupeng Ning, Ray T. Chen, Jiaqi Gu, and David Z. Pan. PACE: Pacing Operator Learning to Accurate Optical Field Simulation for Complicated Photonic Devices. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL `https://papers.nips.cc/paper_files/paper/2024/hash/7cc16e8635e6f27c295355bd214ef8d8-Abstract-Conference.html`.

[50] Yannick Augenstein, Taavi Repän, and Carsten Rockstuhl. Neural Operator-Based Surrogate Solver for Free-Form Electromagnetic Inverse Design. *ACS Photonics*, 10(5):1547–1557, 2023. doi: 10.1021/acsphotonics.3c00156.

[51] Sooyoung Oh, EungKyu Lee, and Sun K. Hong. Physics-Informed Deep Operator Networks for Modeling 2D Time-Domain Electromagnetic Wave Propagation in Various Media. *iScience*, 29(3):115076, 2026. doi: 10.1016/j.isci.2026.115076.

[52] Zaifan Wu, Yue You, Xian Zhou, and Fang Zhang. Accelerated Time-Domain Simulation of Complex Photonic Structures with a Data-Aware Fourier Neural Operator. *Optics Communications*, 604:132838, 2026. doi: 10.1016/j.optcom.2025.132838.

[53] Christopher Leon and Alexander Scheinker. Physics-Constrained Machine Learning for Electrodynamics Without Gauge Ambiguity Based on Fourier Transformed Maxwell's Equations. *Scientific Reports*, 14: 14809, 2024. doi: 10.1038/s41598-024-65650-9.

[54] Vaishnavi Iyer, Blake A. Wilson, Yuheng Chen, Alexander V. Kildishev, Vladimir M. Shalaev, and Alexandra Boltasseva. Deep learning in photonic device development: nuances and opportunities. *npj Nanophotonics*, 3:5, 2026. doi: 10.1038/s44310-025-00097-y.

[55] Mauricio Álvarez, David Luengo, and Neil D. Lawrence. Latent Force Models. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 9–16. PMLR, 2009. URL `https://proceedings.mlr.press/v5/alvarez09a.html`.

[56] Ives Macêdo and Rener Castro. Learning Divergence-Free and Curl-Free Vector Fields with Matrix-Valued Kernels. Technical report, Instituto de Matemática Pura e Aplicada (IMPA), 2010.

[57] Arno Solin, Manon Kok, Niklas Wahlström, Thomas B. Schön, and Simo Särkkä. Modeling and Interpolation of the Ambient Magnetic Field by Gaussian Processes. *IEEE Transactions on Robotics*, 34(4): 1112–1127, 2018. doi: 10.1109/TRO.2018.2830323.

[58] Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B. Schön. Linearly Constrained Gaussian Processes. In *Advances in Neural Information Processing Systems 30*, pages 1215–1224, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/71ad16ad2c4d81f348082ff6c4b20768-Abstract.html`.

[59] Markus Lange-Hegermann. Algorithmic Linearly Constrained Gaussian Processes. In *Advances in Neural Information Processing Systems 31*, pages 2141–2152, 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/68b1fbe7f16e4ae3024973f12f3cb313-Abstract.html`.

[60] Andreas Besginow and Markus Lange-Hegermann. Constraining Gaussian Processes to Systems of Linear Ordinary Differential Equations. In Sanmi Koyejo, Shakir Mohamed, Anima Agarwal, Danielle Belgrave, Kyunghyun Cho, and Alice Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 29386–29399. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/hash/bcef27c5825d1ed8757290f237b2d851-Abstract-Conference.html`.

[61] Markus Lange-Hegermann. Linearly Constrained Gaussian Processes with Boundary Conditions. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1090–1098. PMLR, 2021. URL `https://proceedings.mlr.press/v130/lange-hegermann21a.html`.

[62] Markus Lange-Hegermann and Daniel Robertz. On Boundary Conditions Parametrized by Analytic Functions. In François Boulier, Matthew England, Timur M. Sadykov, and Evgenii V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, volume 13366 of *Lecture Notes in Computer Science*, pages 225–245. Springer, Cham, 2022. doi: 10.1007/978-3-031-14788-3_13.

[63] David Dalton, Alan Lazarus, Hao Gao, and Dirk Husmeier. Boundary constrained Gaussian processes for robust physics-informed machine learning of linear partial differential equations. *Journal of Machine Learning Research*, 25(272):1–61, 2024. URL `http://jmlr.org/papers/v25/23-1508.html`.

[64] Jianlei Huang, Marc Härkönen, Markus Lange-Hegermann, and Bogdan Raiţă. Gaussian Process Priors for Boundary Value Problems of Linear Partial Differential Equations. *arXiv preprint arXiv:2411.16663*, 2024. URL `https://arxiv.org/abs/2411.16663`.

[65] Xin Li, Markus Lange-Hegermann, and Bogdan Raiţă. Gaussian Process Regression for Inverse Problems in Linear PDEs. *IFAC-PapersOnLine*, 59(12):196–201, 2025.

[66] 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. doi: 10.1109/72.712178.

[67] Enrico Schiassi, Roberto Furfaro, Carl Leake, Mario De Florio, Hunter Johnston, and Daniele Mortari. Extreme Theory of Functional Connections: A Fast Physics-Informed Neural Network Method for Solving Ordinary and Partial Differential Equations. *Neurocomputing*, 457:334–356, 2021. doi: 10.1016/j.neucom.2021.06.015.

[68] N. Sukumar and Ankit Srivastava. Exact Imposition of Boundary Conditions with Distance Functions in Physics-Informed Deep Neural Networks. *Computer Methods in Applied Mechanics and Engineering*, 389:114333, 2022. doi: 10.1016/j.cma.2021.114333.

[69] Johannes N. Hendriks, Carl Jidling, Adrian Wills, and Thomas B. Schön. Linearly constrained neural networks. *arXiv preprint arXiv:2002.01600*, 2020.

[70] Jack Richter-Powell, Yaron Lipman, and Ricky T. Q. Chen. Neural conservation laws: A divergence-free perspective. *Advances in Neural Information Processing Systems*, 35, 2022.

[71] Ramansh Sharma, Matthew Lowery, Houman Owhadi, and Varun Shankar. Fluids you can trust: Property-preserving operator learning for incompressible flows. *arXiv preprint arXiv:2602.15472*, 2026.

[72] Xigui Li, Hongwei Zhang, Ruoxi Jiang, Deshu Chen, Chensen Lin, Limei Han, Yuan Qi, Xin Guo, and Yuan Cheng. Project and generate: Divergence-free neural operators for incompressible flows. *arXiv preprint arXiv:2603.24500*, 2026.

[73] Mohammad Sarkari Khorrami, Pawan Goyal, Jaber R. Mianroodi, Bob Svendsen, Peter Benner, and Dierk Raabe. A physics-encoded fourier neural operator approach for surrogate modeling of divergence-free stress fields in solids. *arXiv preprint arXiv:2408.15408*, 2024.

[74] Tao Cui, Ziming Wang, and Xueshuang Xiang. An efficient neural network method with plane wave activation functions for solving Helmholtz equation. *Computers & Mathematics with Applications*, 111: 34–49, 2022.

[75] Long Yuan and Qiya Hu. A discontinuous plane wave neural network method for Helmholtz equation and time-harmonic Maxwell's equations. *Advances in Computational Mathematics*, 51(2):18, 2025.

[76] Zhuojia Fu, Wenzhi Xu, and Shuainan Liu. Physics-informed kernel function neural networks for solving partial differential equations. *Neural Networks*, 172:106098, 2024.

[77] Erich Trefftz. Ein Gegenstück zum Ritzschen Verfahren. In *Proceedings of the 2nd International Congress of Applied Mechanics*, pages 131–137, Zürich, 1926.

[78] Tomi Huttunen, Petri Malinen, and Peter Monk. Solving Maxwell's Equations Using the Ultra Weak Variational Formulation. *Journal of Computational Physics*, 223(2):731–758, 2007. doi: 10.1016/j.jcp.2006.10.016.

[79] Ralf Hiptmair, Andrea Moiola, and Ilaria Perugia. A Survey of Trefftz Methods for the Helmholtz Equation. In Gabriel R. Barrenechea, Franco Brezzi, Andrea Cangiani, and Emmanuil H. Georgoulis, editors, *Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations*, volume 114 of *Lecture Notes in Computational Science and Engineering*, pages 237–279. Springer, Cham, 2016. doi: 10.1007/978-3-319-41640-3_8.

[80] Alexander Scheinker and Reeju Pokharel. Physics-Constrained 3D Convolutional Neural Networks for Electrodynamics. *APL Machine Learning*, 1(2):026109, 2023. doi: 10.1063/5.0132433.

[81] Peter D. Lax. *Hyperbolic Partial Differential Equations*, volume 14 of *Courant Lecture Notes in Mathematics*. Courant Institute of Mathematical Sciences and American Mathematical Society, 2006. ISBN 9780821835760. doi: 10.1090/cln/014.

[82] Rüdiger Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Oxford University Press, 2013.

[83] Joachim Schöberl. A posteriori error estimates for maxwell equations. *Mathematics of Computation*, 77 (262):633–649, 2008. doi: 10.1090/S0025-5718-07-02030-3.

[84] Chenkai Mao and Jonathan A. Fan. Accurate and scalable deep maxwell solvers. *Proceedings of the National Academy of Sciences*, 123(18):e2530330123, 2026. doi: 10.1073/pnas.2530330123. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2530330123`.

[85] Tosio Kato, Marius Mitrea, Gustavo Ponce, and Michael Taylor. Extension and representation of divergence-free vector fields on bounded domains. *Mathematical Research Letters*, 7(5–6):643–650, 2000.

[86] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks. *Neural Networks*, 3(5):551–560, 1990.

A Additional experiments

A.1 Sensitivity Analysis and Ablations

We evaluate *FLASH-MAX* with different network widths and activation functions in Table 3. The results are consistently good.

Table 3: *FLASH-MAX*'s sensitivity to network width (Width) and activation function (Activation). In all these runs, we use the *Random Solution* and a seed of 42. All other hyperparameters are fixed as in Table 1. Executed on a single NVIDIA L40S.

Width (thousands)	Activation	Time to <5% RL2E. (s)	Min. RL2E. (%)	Time to min. RL2E. (s)
0.1	tanh	488.2	4.9	488.2
0.2	tanh	297.0	3.2	501.2
0.5	tanh	202.8	3.1	506.4
1.0	tanh	126.3	2.6	495.1
2.0	tanh	98.8	2.1	475.2
5.0	tanh	78.9	2.0	492.2
10.0	tanh	53.9	1.6	499.3
12.0	tanh	50.4	1.6	474.7
10.0	relu	326.1	3.9	503.4
10.0	silu	118.2	2.2	496.7
10.0	cosine	13.5	1.1	227.9
10.0	gelu	69.5	2.2	538.1
10.0	sigmoid	320.9	3.0	598.6

A.2 Runtimes on CPU

We also note that *FLASH-MAX* is rather efficient even in the absence of a GPU. Table 4 shows timing results for exactly the same setups as shown in Table 1, except that all computation is handled on a CPU. We note that errors well below the 5% threshold are easily achieved in under 10 minutes of CPU training time in most cases, and in all other cases, the threshold is not far afield after that amount of training time.

Table 4: The same *FLASH-MAX* experiments as were carried out in Table 1, but executed on a CPU, an INTEL(R) XEON(R) GOLD 6542Y. We allowed at most 10 minutes of training (600 seconds) to each experiment.

Solution	BC	Minimum Val Error (%)	Compute (s)
Plane Waves	✓	1.1 ± 0.066	551.2 ± 14.3
Plane Waves	✗	1.2 ± 0.070	477.1 ± 46.8
Radial Waves	✓	5.4 ± 0.081	596.2 ± 1.40
Radial Waves	✗	1.9 ± 0.069	577.7 ± 10.2
Hopf Fibration	✓	1.6 ± 0.061	582.8 ± 10.3
Hopf Fibration	✗	1.1 ± 0.030	165.5 ± 7.90
Random Solution	✓	5.7 ± 0.015	596.4 ± 2.20
Random Solution	✗	1.2 ± 0.044	446.6 ± 78.4

A.3 Experiment 2: Performance on a Time Budget (IC setup)

The main paper includes performance over time for BC in Figure 2. We also include the same results for the experiments using only initial conditions (IC) here in Figure 3.

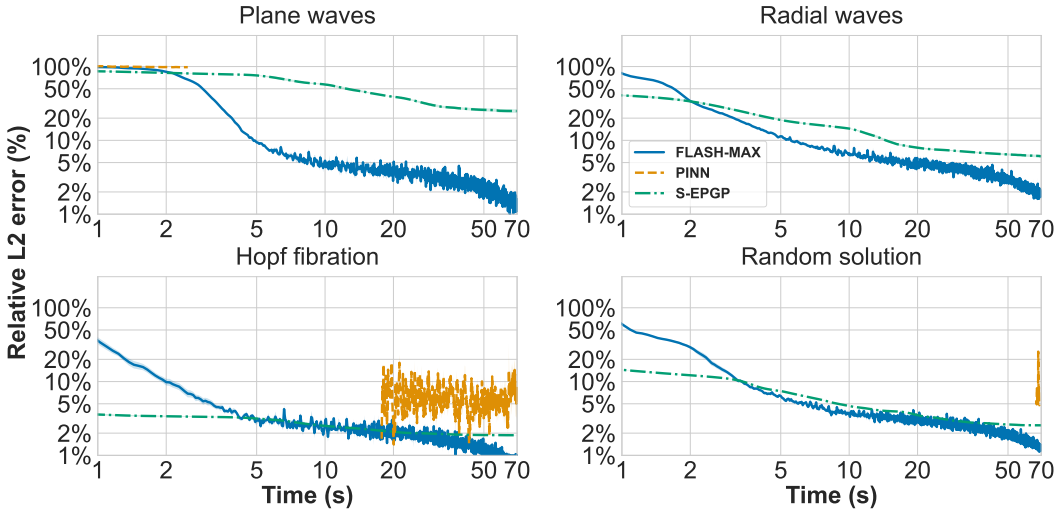

Figure 3: **Experiment 2: Performance on a Time Budget (IC setup).** The curves depict the achievable error rates (y-axis) given a specific time budget (x-axis) for the setting with initial conditions. Uncertainties are shown (standard error of the mean over 5 random seeds). *FLASH-MAX* (blue solid line) converges more reliably and usually significantly faster than existing methods across all four problems: *Plane waves*, *Radial waves*, *Hopf fibration*, and *Random solution*. Plots are analogous to those for the BC setup in Figure 2. While S-EPGP seems to perform comparably here, we note that it required an NVIDIA A100 to achieve this performance, while *FLASH-MAX* is achieving this performance on an NVIDIA L40S.

A.4 Experiment 4: Gaussian noise in training data

We performed an experiment in which we added Gaussian noise with different magnitudes of variance to the training data's targets. Results can be seen in Table 5, where "Noise Scale" means the variance of a zero-centered normal from which noise samples were drawn. Noise samples were added to all target values used in the training data (all training samples, all six components of the electromagnetic field). No noise was added to the validation set in any way. *FLASH-MAX* still achieves remarkable accuracy on the ground truth validation sets, consistently achieving 5% relative error or less in well under two minutes.

B Implementation Details

B.1 Detailed *FLASH-MAX* Description

We also express *FLASH-MAX* in matrix notation: Let $\boldsymbol{Z}_i \in \mathbb{R}^{2W\times 4}$ for $i \in \{1,2\}$ with entries $\boldsymbol{Z}_{ik\ell}$ having ℓth column denoted by $\boldsymbol{Z}_{i\bullet\ell} \in \mathbb{R}^{2W}$ for $\ell \in \{0,1,2,3\}$ and vectors $\boldsymbol{b}_i, \boldsymbol{w}_i \in \mathbb{R}^{2W}$ with entries $\boldsymbol{b}_{ik}$ resp. $\boldsymbol{w}_{ik}$. For notational simplicity, we compress the notation in (*FLASH-MAX*) by absorbing the j index in (1) directly in the definition of our special frequencies

$$\boldsymbol{Z}_{ik0} = \pm\sqrt{\boldsymbol{Z}_{ik1}^2 + \boldsymbol{Z}_{ik2}^2 + \boldsymbol{Z}_{ik3}^2}$$

with half the signs being $+$ resp. $-$. This simplification requires us to double the width parameter W from (*FLASH-MAX*) (instead of working with indices (j,k) with $j = \pm 1$, $k = 1,\dots W$, we now

Table 5: Results akin to Table 1, but with training targets noised using the noise scales given.

Solution	w/ BC	Noise Scale	Time to $< 5\%$ RL2E. (s)	Time to min. RL2E. (s)	Min. RL2E. (%)
Hopf fibration	✗	10^{-5}	3.8 ± 0.1	68.1 ± 0.9	0.50 ± 0.01
Hopf fibration	✗	10^{-4}	3.5 ± 0.2	68.8 ± 0.3	0.46 ± 0.01
Hopf fibration	✗	10^{-3}	3.8 ± 0.2	67.0 ± 1.4	0.75 ± 0.01
Hopf fibration	✓	10^{-5}	7.5 ± 0.3	69.6 ± 0.1	1.38 ± 0.03
Hopf fibration	✓	10^{-4}	7.3 ± 0.5	69.6 ± 0.2	1.41 ± 0.04
Hopf fibration	✓	10^{-3}	8.1 ± 0.5	69.7 ± 0.2	1.63 ± 0.04
Plane waves	✗	10^{-5}	25.2 ± 2.6	68.6 ± 0.3	2.22 ± 0.16
Plane waves	✗	10^{-4}	25.0 ± 2.2	69.2 ± 0.5	2.18 ± 0.17
Plane waves	✗	10^{-3}	25.6 ± 1.6	69.1 ± 0.3	2.24 ± 0.13
Plane waves	✓	10^{-5}	29.6 ± 1.4	68.1 ± 0.9	2.11 ± 0.16
Plane waves	✓	10^{-4}	28.9 ± 0.8	68.2 ± 0.6	2.08 ± 0.14
Plane waves	✓	10^{-3}	28.3 ± 0.7	69.0 ± 0.4	2.10 ± 0.17
Radial waves	✗	10^{-5}	22.1 ± 0.7	69.5 ± 0.3	2.08 ± 0.08
Radial waves	✗	10^{-4}	22.6 ± 1.2	69.6 ± 0.2	2.10 ± 0.05
Radial waves	✗	10^{-3}	22.2 ± 0.3	68.9 ± 0.3	2.11 ± 0.07
Radial waves	✓	10^{-5}	62.6 ± 4.6	71.0 ± 0.9	4.71 ± 0.14
Radial waves	✓	10^{-4}	67.5 ± 6.2	74.9 ± 3.2	4.70 ± 0.12
Radial waves	✓	10^{-3}	66.3 ± 7.2	75.0 ± 3.9	4.69 ± 0.13
Random solution	✗	10^{-5}	6.5 ± 0.4	69.2 ± 0.5	1.07 ± 0.01
Random solution	✗	10^{-4}	6.3 ± 0.2	69.0 ± 0.4	1.14 ± 0.03
Random solution	✗	10^{-3}	6.7 ± 0.3	68.4 ± 0.7	1.37 ± 0.01
Random solution	✓	10^{-5}	76.0 ± 2.3	76.0 ± 2.3	5.00 ± 0.00
Random solution	✓	10^{-4}	75.7 ± 0.9	75.7 ± 0.9	5.00 ± 0.00
Random solution	✓	10^{-3}	83.7 ± 5.7	83.7 ± 5.7	5.00 ± 0.00

only work with indices $k = 1, \dots 2W$). Furthermore, we express (2) using the $2W \times 6$ matrices

$$\boldsymbol{p}_1(\boldsymbol{Z}_1) = \begin{bmatrix} -\boldsymbol{Z}_{1\bullet 1} \odot \boldsymbol{Z}_{1\bullet 3} \\ -\boldsymbol{Z}_{1\bullet 2} \odot \boldsymbol{Z}_{1\bullet 3} \\ \boldsymbol{Z}_{1\bullet 0} \odot \boldsymbol{Z}_{1\bullet 0} - \boldsymbol{Z}_{1\bullet 3} \odot \boldsymbol{Z}_{1\bullet 3} \\ -\boldsymbol{Z}_{1\bullet 0} \odot \boldsymbol{Z}_{1\bullet 2} \\ \boldsymbol{Z}_{1\bullet 0} \odot \boldsymbol{Z}_{1\bullet 1} \\ \boldsymbol{0} \end{bmatrix}, \boldsymbol{p}_2(\boldsymbol{Z}_2) = \begin{bmatrix} \boldsymbol{Z}_{2\bullet 1} \odot \boldsymbol{Z}_{2\bullet 2} \\ -\boldsymbol{Z}_{2\bullet 0} \odot \boldsymbol{Z}_{2\bullet 0} + \boldsymbol{Z}_{2\bullet 2} \odot \boldsymbol{Z}_{2\bullet 2} \\ \boldsymbol{Z}_{2\bullet 2} \odot \boldsymbol{Z}_{2\bullet 3} \\ -\boldsymbol{Z}_{2\bullet 0} \odot \boldsymbol{Z}_{2\bullet 3} \\ \boldsymbol{0} \\ \boldsymbol{Z}_{2\bullet 0} \odot \boldsymbol{Z}_{2\bullet 1} \end{bmatrix}$$

and entry-wise application of the activation σ we obtain (*FLASH-MAX*) at $\boldsymbol{x} \in \mathbb{R}^4$

$$\begin{pmatrix} \boldsymbol{E}_\theta(\boldsymbol{x}) \\ \boldsymbol{B}_\theta(\boldsymbol{x}) \end{pmatrix} = \boldsymbol{w}_1^\top \left(\sigma(\boldsymbol{x}^\top \boldsymbol{Z}_1 + \boldsymbol{b}_1) \odot \boldsymbol{p}_1(\boldsymbol{Z}_1)\right) + \boldsymbol{w}_2^\top \left(\sigma(\boldsymbol{x}^\top \boldsymbol{Z}_2 + \boldsymbol{b}_2) \odot \boldsymbol{p}_2(\boldsymbol{Z}_2)\right) \in \mathbb{R}^6, \quad (3)$$

where $\theta = \{\boldsymbol{Z}_{ik1}, \boldsymbol{Z}_{ik2}, \boldsymbol{Z}_{ik3}, \boldsymbol{b}_{ik}, \boldsymbol{w}_{ik} \colon i = 1, 2,\ k = 1, \dots 2W\}$ are ordinary trainable weights.

The forward pass of *FLASH-MAX* described above is also represented in Figure 4.

We view (*FLASH-MAX*) and its implementation as a shallow neural network with two bells and whistles attached: the weight sharing in the $\boldsymbol{z}_{ik0}$ being determined by other entries and the multiplication by the $\boldsymbol{p}(\boldsymbol{z}_i)$. These two bells and whistles ensure the properties regarding Maxwell's equations (MAX) Most importantly, the model in (*FLASH-MAX*) only yields solutions of (MAX), which is proved by direct calculation in Lemma 4). Moreover, *FLASH-MAX* has very few hyperparameters, as we do not have additional layers, nor a need for collocation points to enforce the equations.

***FLASH-MAX* training setup.** In all *FLASH-MAX* experiments in Table 1, we use the AdamW optimizer with a learning rate of 5×10^{-2}, a weight decay of 5×10^{-5} and $(\beta_1, \beta_2) = (0.9, 0.95)$. The activation function is $\sigma = \tanh$. We initialize all weights with the Xavier normal initialization with a standard gain of $5/3$, except for the biases which are initialized with 0. We use a model width

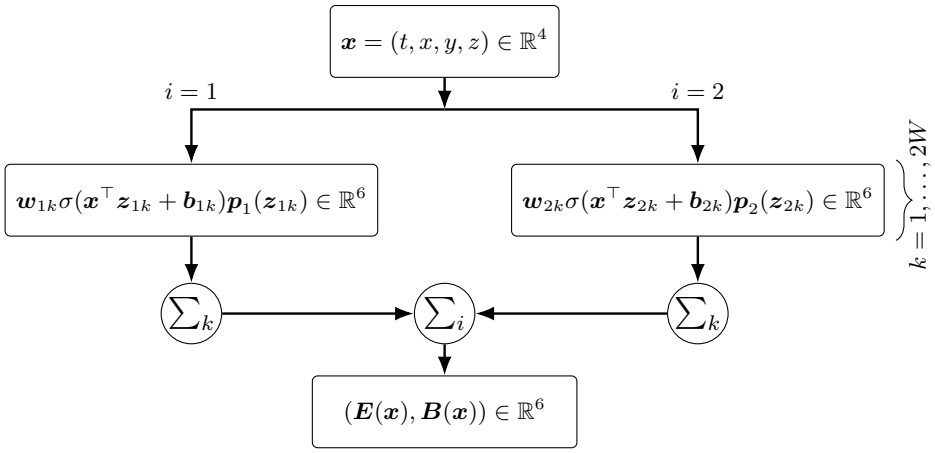

Figure 4: Graphical representation of the forward pass (3) of *FLASH-MAX*, here expressed much like in (*FLASH-MAX*) in terms of each timespace frequency $\boldsymbol{z}_{ik} = \boldsymbol{Z}_{ik\bullet} \in \mathbb{R}^4$ for $i = 1, 2$, $k = 1, \ldots, 2W$.

Table 6: Details concerning the experiments in Table 1.

Method	**BC**	**# pts**	**Device**	**Eval. domain**
FEM	✓	n/a	CPU	$[0,1]^4$
FEM	✗	n/a	CPU	$[0,0.1] \times [0.2,0.8]^3$
PINN	✓	10k	L40S	$[0,1]^4$
PINN	✗	10k	L40S	$[0,0.1] \times [0.2,0.8]^3$
S-EPGP	✓	12k	A100	$[0,1]^4$
S-EPGP	✗	12k	A100	$[0,0.1] \times [0.2,0.8]^3$
FLASH-MAX (ours)	✓	1k	L40S	$[0,1]^4$
FLASH-MAX (ours)	✗	1k	L40S	$[0,0.1] \times [0.2,0.8]^3$

of $w = 10$K (corresponding to 200K trainable parameters) and train with randomized batches of size 1K over 2K training points. Validation error calculations are over 10K points sampled from the validation domain. A cosine learning rate scheduler was employed over the first 10K epochs of training with $\eta_{\min} = 0$.

In the boundary condition setup, $1/7$ of the training points are sampled from the interior of the training spatial domain $(x, y, z) \in [0,1]^3$ with $t = 0$ (initial condition points - IC points), while $1/7$ of the training points are sampled from each of the six faces of the training spatial domain $(x, y, z) \in \partial[0,1]^3$ with $t \in [0,1]$ (boundary condition points - BC points). Validation data are drawn from $(t, x, y, z) \in [0,1]^4$. We mask the targets during training such that loss for BC points is computed only for the non-normal components of the electric field, effectively minimizing only MSE $(\boldsymbol{E} \times \boldsymbol{n}, \boldsymbol{E}_{\text{gt}} \times \boldsymbol{n})$ for BC points. For instance, considering a BC point from the top face of $[0,1]^3$, we compute the loss during training only over the components $\boldsymbol{E}_x$ and $\boldsymbol{E}_y$ as the normal direction $\boldsymbol{E}_z$ is not part of the boundary condition. We never include the $\boldsymbol{B}$ field for BC points–it is only used for the IC points.

In the IC setup, all training points are sampled from the interior of the training spatial domain $(x, y, z) \in [0,1]^3$ with $t = 0$. Validation data are drawn from $(x, y, z) \in [0.2, 0.8]^3$ with $t \in [0, 0.1]$. No special masking is done.

B.2 State-of-the-art Alternative Approaches

The three main approaches we compare with are PINNs, Gaussian Processes (S-EPGP), and FEMs.

PINNs. We benchmark *FLASH-MAX* against a Physics-Informed Neural Network (PINN) [1] on the time-dependent Maxwell system in Eq. MAX. Unless otherwise specified, all PINN experiments use

the same architecture and optimization setup: a 6-layer MLP with hidden size 128, Adam optimizer with learning rate 10^{-2}, and joint optimization of all active loss terms. For physics supervision, we construct a uniform Cartesian grid over $[0,1]^4$ and randomly sample 50K collocation points. For data supervision, initial-condition points are sampled from the $t=0$ slice using a $50 \times 50 \times 50$ Cartesian grid over $(x,y,z) \in [0,1]^3$, from which 10K training points are randomly selected.

In this work, we consider two settings that align with the problem formulations. In the IC setting, the model is trained with initial data, PDE, and divergence losses, with weights $\lambda_{\text{Data}} = 10$, $\lambda_{\text{PDE}} = 1$, and $\lambda_{\text{div}} = 2$, and no boundary supervision is imposed. In the BC setting, we additionally enforce the boundary loss with weight $\lambda_{\text{BC}} = 10$, while keeping the other loss weights unchanged. Here, PDE loss refers to the residuals of the first two equations in the MAX; and λ_{div} is the weight for divergence loss corresponding to the residuals for the last two equations in the MAX. Boundary samples are drawn from the spatial boundary of the unit cube over $t \in [0,1]$, and validation includes 10K sampled boundary points together with evaluation over the full space-time domain $[0,1]^4$.

S-EPGP. We benchmark *FLASH-MAX* against the S-EPGP baseline [2], following the same formulation and Maxwell-consistent feature construction as in [2, 65]. In all experiments, S-EPGP is trained by minimizing the negative log marginal likelihood (NLML), and relative error is computed jointly over all six electromagnetic components. Unless otherwise specified, all S-EPGP experiments use the same training setup: Adam optimization on an A100 GPU, $n_{\text{MC}} = 300$ frequency samples, and $8n_{\text{MC}} = 2400$ spectral features. Initial-condition supervision is constructed from the same $t = 0$ Cartesian grid as above, from which we sample 2K locations, corresponding to 12K scalar observations over the six field components.

We consider two settings aligned with the formulations in this work. In the IC setting, the model is trained using only observations at $t = 0$. Since performance degrades substantially when extrapolating to longer time horizons, evaluation for this baseline is restricted to the interior region $[0, 0.01] \times [0.2, 0.8]^3$. We note that this is a shorter temporal interval than that used for *FLASH-MAX*, reflecting the limited stability of the finite-feature S-EPGP approximation. In the BC setting, we additionally impose boundary constraints using 1K sampled points per spatial face. The tangential electric-field constraints are incorporated by augmenting the S-EPGP feature matrix and observation vector, and evaluation is performed over the full space-time domain $[0,1]^4$.

The relatively weak performance of S-EPGP, especially in the BC setting, likely arises from the finite feature approximation. While the Maxwell-consistent basis is theoretically expressive in the infinite-feature limit, $n_{\text{MC}} = 300$ appears insufficient to accurately represent the target solution over the full constrained domain. As a result, when initial and boundary constraints are combined in the augmented system, basis mismatch can lead to tension between the two objectives, whereas fitting the initial slice alone is substantially easier.

FEM. We benchmark *FLASH-MAX* against a classical structure-preserving finite element baseline [38, 40] implemented in DOLFINx/FEniCS [43]. The FEM baseline is used only in the boundary condition setting, since the no boundary condition setting considered in this paper is a sparse-data extrapolation task rather than a fully specified forward initial-boundary value problem. Consistent with the structure-preserving FEM perspective discussed in Section 2, the solver uses a mixed $H(\mathrm{curl}) \times H(\mathrm{div})$ discretization of the source-free Maxwell system (MAX), where the electric field $\boldsymbol{E}$ is represented with first-kind Nédélec elements and the magnetic field $\boldsymbol{B}$ with Raviart–Thomas elements.

Time integration is performed with a Crank–Nicolson scheme. Since the benchmark problem is linear and the mesh is fixed, the system matrix is assembled once and reused throughout the time-stepping loop. All FEM runs use a structured tetrahedral mesh of $[0,1]^3$, with N subdivisions per coordinate direction and $6N^3$ tetrahedra. We use matching polynomial degree-2 spaces for $\boldsymbol{E}$ and $\boldsymbol{B}$, time step $\Delta t = 10^{-2}$, and a PETSc GMRES solver with block-Jacobi ILU preconditioning.

Initial data are imposed by interpolating the exact electric and magnetic fields at $t = 0$ into the corresponding finite element spaces. The tangential electric boundary condition $\boldsymbol{E} \times \boldsymbol{n}$ is imposed strongly on $\partial[0,1]^3$ at every time step using the exact trace; no boundary condition is imposed directly on $\boldsymbol{B}$. Validation is performed on 10K randomly sampled spacetime points in $[0,1]^4$, using the same relative error metric as the other methods and computing the error jointly over all six

electromagnetic components. We additionally report a final-step FEM residual computed from the last two Crank–Nicolson time levels.

The results of our FEM implementation on the four ground truths (Appendix C) are reported in Table 1. Note that the performance of FEM on the *Radial Waves* solution was very poor. This is due to a numerical artifact at $(0, 0, 0, 0)$, see Figure 6 (top-left). The ML methods we use do not seem to be affected by the artifact due to the random sampling of data. To demonstrate that our FEM converges, albeit slowly, we approximate the *Radial Waves* ground truth by shifting the domain away from the singularity, taking $\Omega = (0.1, 1.1)^3$. FEM reaches sub-5% validation error, but stabilizes well above the PDE residual threshold of 0.01 reported (Table 7), thus remaining far from the performance of *FLASH-MAX* for *Radial Wave* with boundary conditions in Table 1.

Table 7: Additional experiment for FEM. We repeat the *Race to 5%* experiment in Table 1 on the *Radial Waves* ground truth. Spacetime domain $(0, 1) \times (0.1, 1.1)^3$, designed to avoid the numerical artifact at the origin in Figure 6(top-left).

Ground Truth	Method	Space	Val Error (%)	Residual Error	Compute (s)
Radial Waves	FEM	$(0.1, 1.1)^3$	**<5.00**	0.140	287.4

B.3 Physics-Informed Neural Network (PINN) Baseline: QPINN and Evo-PINN

As discussed in Section 3, we further compare our in-house PINN baseline with two representative PINN-style methods, QPINN [36] and Evo-PINN [37]. Among these PINN-based baselines, our in-house PINN consistently achieves the strongest performance and is therefore used as the primary PINN comparator in the main text. For completeness, we report the detailed results of QPINN and Evo-PINN in Table 8.

To the best of our knowledge, no prior PINN work with publicly available code directly targets the general homogeneous, constant-coefficient 4D time-dependent Maxwell problem in Equation (MAX). We therefore include QPINN and Evo-PINN as the closest reproducible alternatives we could identify. QPINN is the most closely related task-specific baseline, but it is developed for a 2D hybrid quantum-classical Maxwell formulation with several problem-specific design choices. Evo-PINN, in contrast, is a recent general framework aimed at improving PINN optimization rather than a method specialized for Maxwell equations in our formulation. Although these baselines are not fully matched

Table 8: Validation Error, Residual Error, and Compute training time to reach the best relative validation error.

	Method	BC	Validation Error	Residual Error	Compute	Device
PlaneWave	QPINN[36]	✓	48.07%	9.16×10^{-2}	348.55s	A100
		✗	96.06%	8.38×10^{-2}	37.26s	
	Evo-PINN[37]	✓	100.17%	7.7×10^{-3}	35.87s	CPU
		✗	99.10%	1.6×10^{-1}	$146.29s$	
RadialWave	QPINN[36]	✓	99.68%	6.46×10^{-4}	7.14s	A100
		✗	52.37%	2.47×10^{-1}	36.23s	
	Evo-PINN[37]	✓	101.81%	5.1×10^{-3}	35.81s	CPU
		✗	96.30%	7.3×10^{-2}	$87.77s$	
Hopf	QPINN[36]	✓	36.92%	6.42×10^{-2}	379.71s	A100
		✗	8.71%	4.84×10^{-2}	1186.49s	
	Evo-PINN[37]	✓	83.86%	1.0×10^{-3}	18.85s	CPU
		✗	46.88%	1.9×10^{-2}	73.1s	
Random	QPINN[36]	✓	23.60%	3.83×10^{-2}	956.27s	A100
		✗	37.30%	4.46×10^{-2}	913.69s	
	Evo-PINN[37]	✓	103.06%	9.9×10^{-4}	20.95s	CPU
		✗	73.90%	2.9×10^{-1}	$2549s$	

to our problem, they provide useful reference points for evaluating how related or more general PINN methodologies transfer to our setting. In our in-house PINN experiments, we do not add any additional techniques, but follow the tight alignment with the exact MAX formulation and training regime.

We adapt QPINN by retaining its core idea of inserting a parameterized quantum circuit near the output of an otherwise classical PINN, while replacing the original 2D three-field formulation (E_z, H_x, H_y) with our six-component representation $(B_1, B_2, B_3, E_1, E_2, E_3)$. Our implementation uses a 6-layer MLP with hidden width 128, whose penultimate features are projected to a 7-qubit, 4-layer quantum circuit with Pauli-Z readout, using the strongly entangling ansatz and `acos` angle scaling. We also retain the additional global energy regularization used in the original QPINN study. Since several components of the original 2D setup are not directly transferable to our problem, this baseline should be viewed as an approximate transfer of the QPINN design rather than a matched reproduction.

We adapt Evo-PINN by replacing standard gradient-based PINN training with a CMA-ES optimizer implemented through EvoJAX, while keeping the PINN itself as a classical network over the same Maxwell variables and residual terms used in our benchmark. The resulting model is a smaller MLP trained by evolutionary search under a physics-informed objective composed of data, PDE, divergence, and, when applicable, boundary losses. This baseline follows the optimization-oriented spirit of Evo-PINN, but should be interpreted as an Evo-PINN-inspired evolutionary PINN rather than a like-for-like reproduction of a method designed for our Maxwell formulation. Moreover, under our computational budget, the evolutionary baseline uses both a smaller model and a more limited training budget than our in-house PINN, which likely further constrains its final performance.

C List of Ground Truths

We test each solver (ours, PINN, GP, FEM) using the four ground truths that we describe in detail below. These vector fields are chosen to challenge every solver and, in particular, none of them favor *FLASH-MAX* a priori. Moreover, these ground truths are very different mathematically from each other. *FLASH-MAX* substantially outperformed competitors for any choice of ground truth and across all experiments we performed.

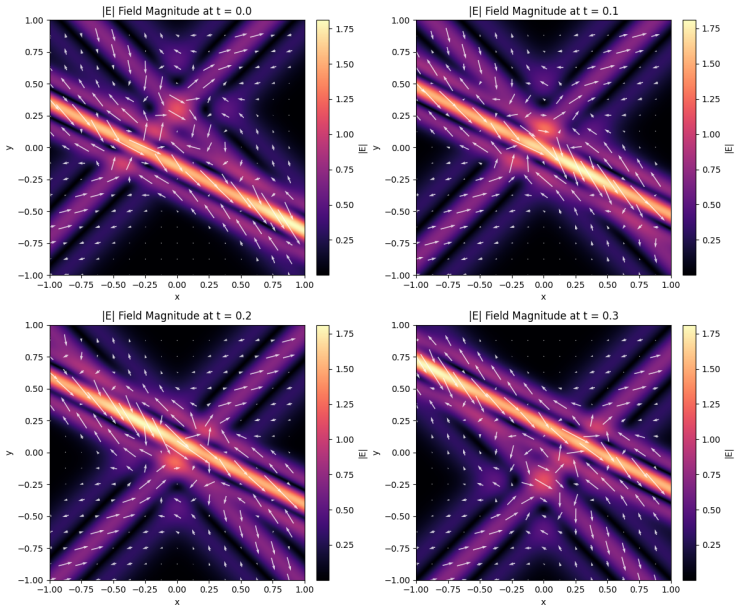

Figure 5: Time evolution of the magnitude of the electric field for the *Plane Waves* solution.

Plane Waves. The solution of (MAX) with simplest structure that we use is a superposition of three plane wave solutions with profile derived from a squared exponential $f(s) = 0.01e^{-10(s-0.3)^2}$. Let

$$F(s) = f''(s) = -0.2e^{-10(s-0.3)^2}\left[1 - 20(s-0.3)^2\right]$$

The directions of the three plane waves $\boldsymbol{z}$ must satisfy the light cone restriction $z_0^2 = z_1^2 + z_2^2 + z_3^2$. We consider $\boldsymbol{z} \in \{(\sqrt{3}, 1, 1, 1), (\sqrt{3}, -1, 1, 1), (\sqrt{6}, -1, -2, 1)\}$. Let

$$\begin{aligned} s_1 &= \sqrt{3}t + x + y + z \\ s_2 &= \sqrt{3}t - x + y + z \\ s_3 &= \sqrt{6}t - x - 2y + z \end{aligned}$$

Let us define the evaluated functions as $F_1 = F(s_1)$, $F_2 = F(s_2)$, and $F_3 = F(s_3)$. Recall that $\boldsymbol{x} = (t, x, y, z)$. The **electric field** $\boldsymbol{E}$ is given by

$$\begin{aligned} E_1(\boldsymbol{x}) &= (1-\sqrt{3})F_1 - (1+\sqrt{3})F_2 + (1-2\sqrt{6})F_3 \\ E_2(\boldsymbol{x}) &= (1+\sqrt{3})F_1 + (1-\sqrt{3})F_2 + (2+\sqrt{6})F_3 \\ E_3(\boldsymbol{x}) &= -2F_1 - 2F_2 + 5F_3, \end{aligned}$$

while the **magnetic field** $\boldsymbol{B}$ is given by

$$\begin{aligned} B_1(\boldsymbol{x}) &= (\sqrt{3}+1)F_1 + (\sqrt{3}-1)F_2 + (2\sqrt{6}+1)F_3 \\ B_2(\boldsymbol{x}) &= (1-\sqrt{3})F_1 + (1+\sqrt{3})F_2 + (2-\sqrt{6})F_3 \\ B_3(\boldsymbol{x}) &= -2F_1 - 2F_2 + 5F_3. \end{aligned}$$

Snapshots of the evolution of the magnitude of the electric field can be found in Figure 5.

Radial Waves. We next consider the evolution of a symmetric radial wave, modelling a Gaussian pulse at $(0, 0, 0)$ in space leading to waves which dissipate outward. This solution is fundamentally different than our network (*FLASH-MAX*).

Let $r = \sqrt{x^2 + y^2 + z^2}$ and $s = r - t$. The radial profile is given by

$$\begin{aligned} f(s) &= 0.01e^{-10(s-0.7)^2} \\ f'(s) &= -0.2(s-0.7)e^{-10(s-0.7)^2} \\ f''(s) &= -0.2e^{-10(s-0.7)^2}\left[1 - 20(s-0.7)^2\right] \end{aligned}$$

We will use the following auxiliary scalar fields

$$\begin{aligned} g(s) &= \frac{f'(s)}{r^2} - \frac{f(s)}{r^3} \\ h(s) &= \frac{f''(s)}{r^3} - \frac{3f'(s)}{r^4} + \frac{3f(s)}{r^5} \\ q(s) &= -\frac{f''(s)}{r^2} + \frac{f'(s)}{r^3} \end{aligned}$$

With this notation, the **electric field** $\boldsymbol{E}$ is given by

$$\begin{aligned} E_1(\boldsymbol{x}) &= 10\left[xzh - yq\right] \\ E_2(\boldsymbol{x}) &= 10\left[yzh + xq\right] \\ E_3(\boldsymbol{x}) &= 10\left[g + z^2h - \frac{f''(s)}{r}\right] \end{aligned}$$

and the **magnetic field** $\boldsymbol{B}$ is

$$\begin{aligned} B_1(\boldsymbol{x}) &= 10\left[yq + xzh\right] \\ B_2(\boldsymbol{x}) &= 10\left[-xq + yzh\right] \\ B_3(\boldsymbol{x}) &= 10\left[-2g - (x^2+y^2)h\right] \end{aligned}$$

Snapshots of the evolution of the magnitude of the electric field can be found in Figure 6.

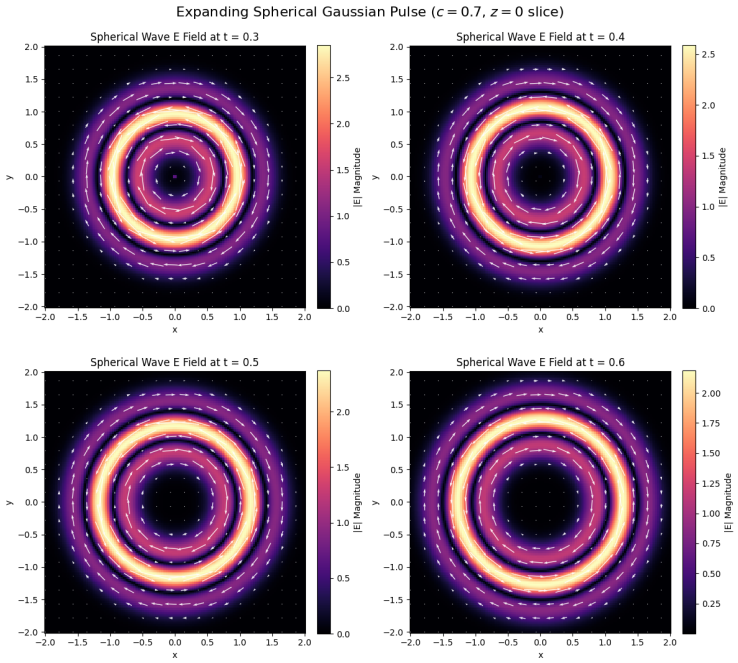

Figure 6: Time evolution of the magnitude of the electric field for the *Radial Waves* solution.

Hopf Fibration. We consider an electromagnetic knot or Hopfion, which is a smooth rational function (quotient of polynomials without singularities). Thus its nature is algebraic, which makes it completely different from the other three solutions we consider. Consider the following auxiliary polynomial quantities:

$$A = 1 + x^2 + y^2 + z^2 - t^2, \qquad T_1 = A^3 - 12t^2A,$$
$$T_2 = 8t^3 - 6tA^2 \qquad D = (A^2 + 4t^2)^3.$$

With this notation, the **electric field** $\boldsymbol{E}$ is given by

$$E_1(\boldsymbol{x}) = D^{-1}[T_1\left[(t-z)^2 - 1 - x^2 + y^2\right] - T_2\left[2xy - 2(t-z)\right]]$$
$$E_2(\boldsymbol{x}) = D^{-1}[T_1\left[-2xy - 2(t-z)\right] - T_2\left[1 - (t-z)^2 - x^2 + y^2\right]]$$
$$E_3(\boldsymbol{x}) = D^{-1}[T_1\left[2x(t-z) - 2y,\right] + T_2\left[2x + 2y(t-z)\right]]$$

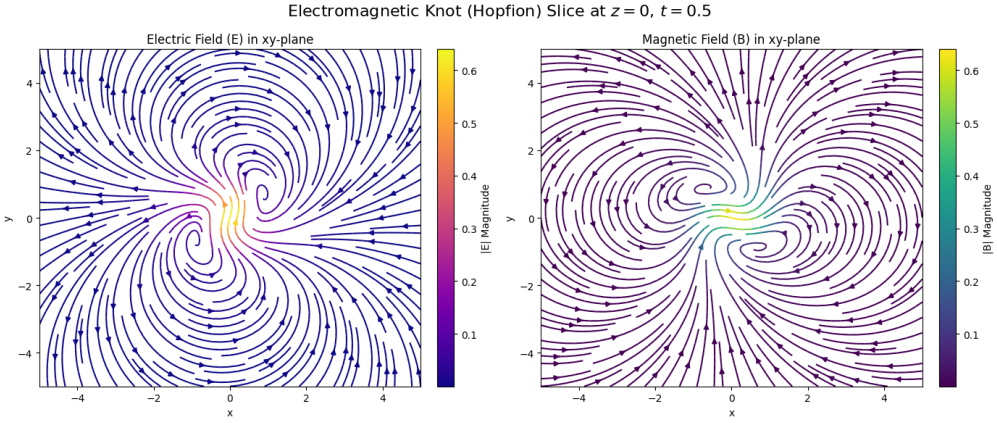

Figure 7: Planar streamlines of the *Hopf Fibration* solution.

and the **magnetic field** $\boldsymbol{B}$ is

$$
\begin{aligned}
B_1(\boldsymbol{x}) &= D^{-1}[T_2\left[(t-z)^2 - 1 - x^2 + y^2\right] + T_1\left[2xy - 2(t-z)\right]] \\
B_2(\boldsymbol{x}) &= D^{-1}[T_2\left[-2xy - 2(t-z)\right] + T_1\left[1 - (t-z)^2 - x^2 + y^2\right]] \\
B_3(\boldsymbol{x}) &= D^{-1}[T_2\left[2x(t-z) - 2y\right] - T_1\left[2x + 2y(t-z)\right]].
\end{aligned}
$$

Beautiful planar snapshots of the streamlines of the electric and magnetic fields can be found in Fig. 7.

Random Solution. For the last solution we add 100 plane waves similar to those used in the *Plane Waves* solution. The direction and shifts of these plane waves are chosen randomly as we explain below. This ensures that the solutions we obtain are indistinguishable from an arbitrary solution of (MAX).

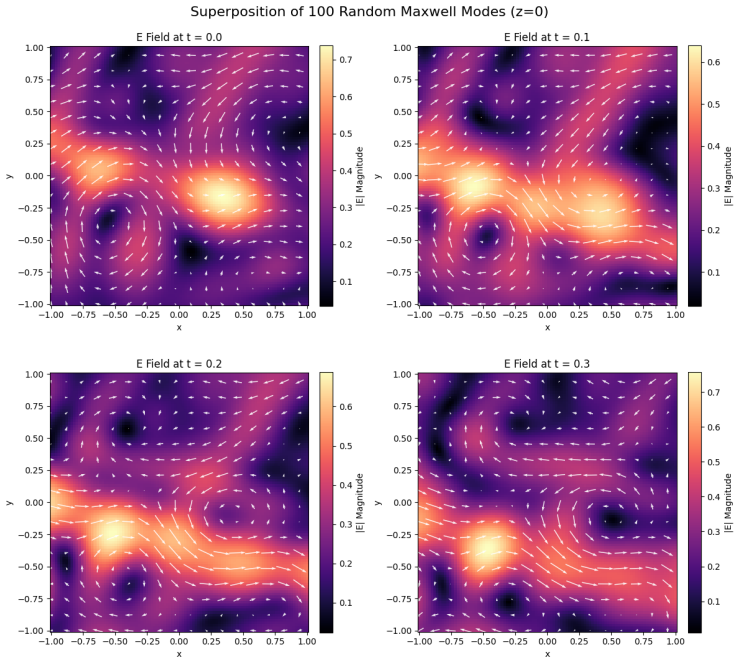

Figure 8: Time evolution of the magnitude of the electric field for the *Random Solution*.

For each $k \in \{1, 2, \ldots, 100\}$, we sample independently $z_{k1}, z_{k2}, z_{k3} \sim \mathcal{N}(0, 0.1)$ and $b_k \sim \mathcal{N}(0, 1)$ and let $z_{k0} = \sqrt{z_{k1}^2 + z_{k2}^2 + z_{k3}^2}$. Let

$$
\begin{aligned}
s_k &= z_{k0}t + z_{k1}x + z_{k2}y + z_{k3}z + b_k \\
F(s) &= -0.2e^{-10(s-0.3)^2}\left[1 - 20(s-0.3)^2\right]
\end{aligned}
$$

With this notation, the **electric field** $\boldsymbol{E}$ is given by

$$
\begin{aligned}
E_1(\boldsymbol{x}) &= \sum_{k=1}^{100}\left[z_{k1}z_{k3} - z_{k2}z_{k0}\right]F(s_k) \\
E_2(\boldsymbol{x}) &= \sum_{k=1}^{100}\left[z_{k2}z_{k3} + z_{k1}z_{k0}\right]F(s_k) \\
E_3(\boldsymbol{x}) &= \sum_{k=1}^{100}\left[z_{k3}^2 - z_{k0}^2\right]F(s_k)
\end{aligned}
$$

and the **magnetic field** $\boldsymbol{B}$ is

$$B_1(\boldsymbol{x}) = \sum_{k=1}^{100} \left[z_{k0}z_{k2} + z_{k1}z_{k3}\right] F(s_k)$$
$$B_2(\boldsymbol{x}) = \sum_{k=1}^{100} \left[-z_{k0}z_{k1} + z_{k2}z_{k3}\right] F(s_k)$$
$$B_3(\boldsymbol{x}) = \sum_{k=1}^{100} \left[-z_{k1}^2 - z_{k2}^2\right] F(s_k).$$

Snapshots of the evolution of the magnitude of the electric field can be found in Figure 8.

D Proof of the Universal Approximation Theorem

We first clarify what we mean by L^2-solutions of (MAX). To do so, we define the spaces we are working with: Let $\Omega \subset \mathbb{R}^d$ be an open set; in our work $d = 3, 4$. We will work with spaces of smooth functions

$$C^m(\Omega, \mathbb{R}^n) = \{u \colon \Omega \to \mathbb{R}^n \colon \partial^\alpha u \text{ uniformly continuous whenever } |\alpha| \leq m\}.$$

To be clear, here $\alpha \in \mathbb{N}^d$ is a multi-index so

$$\partial^\alpha = \partial_1^{\alpha_1}\partial_2^{\alpha_2}\dots\partial_d^{\alpha_d}$$

and $|\alpha| = \alpha_1 + \alpha_2 + \ldots + \alpha_d$. The spaces $C^m(\Omega, \mathbb{R}^n)$ are Banach spaces under the norms

$$\|u\|_{C^m(\Omega,\mathbb{R}^n)} = \max_{|\alpha|\leq m, \boldsymbol{x}\in\Omega} |\partial^\alpha u(\boldsymbol{x})|,$$

where $|\cdot|$ denotes the ℓ^2-norm on a finite dimensional inner product space (like $\mathbb{R}^n$). No confusion will arise from using the same notation $|\cdot|$ for two types of length. The space

$$C^\infty(\Omega, \mathbb{R}^n) = \bigcap_{m\geq 0} C^m(\Omega, \mathbb{R}^n)$$

is a Frechét space under the C^m-seminorms. Finally, we will work with the space $C_c^\infty(\Omega, \mathbb{R}^n)$ of compactly supported smooth functions, meaning that $u \in C_c^\infty(\Omega, \mathbb{R}^n)$ if $u \in C^\infty(\Omega, \mathbb{R}^n)$ and the set $\{\boldsymbol{x} \in \Omega \colon u(\boldsymbol{x}) \neq \boldsymbol{0}\}$ is compact (closed and bounded).

The typical space for error measurement is

$$L^2(\Omega, \mathbb{R}^n) = \{u \colon \Omega \to \mathbb{R}^n \colon u \text{ measurable, } \int_\Omega |u(\boldsymbol{x})|^2 d\boldsymbol{x} < \infty\},$$

We will also work with the Sobolev spaces

$$H^m(\Omega, \mathbb{R}^n) = \{u \in L^2(\Omega, \mathbb{R}^n) \colon \partial^\alpha u \in L^2(\Omega, \mathbb{R}^n) \text{ whenever } |\alpha| \leq m\},$$

the space of functions with all derivatives of order at most m belonging to L^2. In particular, $H^0 = L^2$. We say that $\partial^\alpha u \in L^2(\Omega)$ if there is a constant $C > 0$ such that

$$\int_\Omega u(\boldsymbol{x})^\top \partial^\alpha v(\boldsymbol{x}) d\boldsymbol{x} \leq C\|v\|_{L^2(\Omega,\mathbb{R}^n)} \quad \text{for all } v \in C_c^\infty(\Omega, \mathbb{R}^n).$$

The spaces $H^m(\Omega, \mathbb{R}^n)$ are Hilbert spaces with the inner product

$$\langle u, v\rangle_{H^m(\Omega,\mathbb{R}^n)} = \sum_{|\alpha|\leq m} \int_\Omega \partial^\alpha u(\boldsymbol{x})^\top \partial^\alpha v(\boldsymbol{x}) d\boldsymbol{x},$$

which induces the norm

$$\|u\|_{H^m(\Omega,\mathbb{R}^n)} = \sqrt{\sum_{|\alpha|\leq m} \int_\Omega |\partial^\alpha u(\boldsymbol{x})|^2 d\boldsymbol{x}}.$$

An L^2-solution $(\boldsymbol{E}, \boldsymbol{B})$ of (MAX) in $\Omega \subset \mathbb{R}^4$ is simply a function $(\boldsymbol{E}, \boldsymbol{B}) \in L^2(\Omega, \mathbb{R}^6)$ that is a distributional solution. To define this here, write

$$A(\partial)(\boldsymbol{E}, \boldsymbol{B}) = \begin{pmatrix} \partial_t \boldsymbol{E} - \nabla \times \boldsymbol{B} \\ \partial_t \boldsymbol{B} + \nabla \times \boldsymbol{E} \\ \nabla \cdot \boldsymbol{E} \\ \nabla \cdot \boldsymbol{B} \end{pmatrix},$$

where $\partial = (\partial_t, \partial_x, \partial_y, \partial_z)$ and $\nabla = (\partial_x, \partial_y, \partial_z)^\top$. Then $A(\partial)(\boldsymbol{E}, \boldsymbol{B}) = \mathbf{0}$ in Ω if and only if

$$\int_\Omega (\boldsymbol{E}(\boldsymbol{x}), \boldsymbol{B}(\boldsymbol{x}))^\top A(\partial)^\top \varphi(\boldsymbol{x}) d\boldsymbol{x} = 0 \quad \text{for all } \varphi \in C_c^\infty(\Omega, \mathbb{R}^8).$$

In particular, if $(\boldsymbol{E}, \boldsymbol{B}) \in H^1(\Omega, \mathbb{R}^6)$, this distributional relation reduces by integration by parts to (*FLASH-MAX*) expressed almost everywhere in Ω.

We can now state our main theorem, extending Theorem 1 to various norms. We will only work with time-space domains of the form $(t_1, t_2) \times \Omega$ as it seems unphysical to allow for gaps in the time domain. However, the result below holds for any open set in $\mathbb{R}^4$.

Theorem 2 (Universal approximation for Maxwell's equations)**.** *Let $m \geq 0$ be an integer, $\sigma \in C^m(\mathbb{R})$ be non-polynomial, $t_1 < t_2$, and $\Omega \subset \mathbb{R}^3$ be a bounded open set. Let $(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}})$ be a H^m-solution of the homogeneous Maxwell equations* (MAX) *on $\mathbb{R}^4$. Then, for every $\varepsilon > 0$, there exists a finite Maxwell network solution $(\boldsymbol{E}, \boldsymbol{B})$ of the form* (*FLASH-MAX*) *such that*

$$\|(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}}) - (\boldsymbol{E}, \boldsymbol{B})\|_{H^m((t_1,t_2)\times\Omega)} < \varepsilon.$$

If $m > 1$, the same holds true with the H^m-norm replaced by the C^{m-2}-norm.

In particular, for C^0-approximations, convergence is uniform. Moreover, if we have a smooth solution $(\boldsymbol{E}, \boldsymbol{B}) \in C^\infty(\mathbb{R}^4, \mathbb{R}^6)$, we obtain an approximation in the Frechét topology on $C^\infty((t_1, t_2) \times \Omega)$.

The fact that functions in (*FLASH-MAX*) satisfy Maxwell's equations (MAX) follows by construction, since we use the solvepde command in Macaulay2 [7]. We will prove this directly in the sequel.

Theorem 2 follows from the next statement which incorporates boundary conditions. We will need to define the set

$$\Omega_T = \{\boldsymbol{x} \in \mathbb{R}^3 \colon \operatorname{dist}(\boldsymbol{x}, \Omega) < T\},$$

which is the region of $\mathbb{R}$ where initial conditions uniquely determine the solution of (MAX) up to time T. This is an aspect of the so-called *finite speed of propagation* property [81, Theorem 3.12].

Theorem 3 (Universal approximation for Maxwell's equations with initial conditions)**.** *Let $m \geq 0$ be an integer, $\sigma \in C^m(\mathbb{R})$ be non-polynomial, $T > 0$, and $\Omega \subset \mathbb{R}^3$ be a convex bounded open set. Let $\boldsymbol{E}_0, \boldsymbol{B}_0 \in H^m(\Omega_T, \mathbb{R}^3)$ be divergence-free. Let $(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}})$ be the unique solution in $H^m((0,T) \times \Omega, \mathbb{R}^6)$ of the homogeneous Maxwell equations* (MAX)*. Then, for every $\varepsilon > 0$, there exists a finite Maxwell network solution $(\boldsymbol{E}, \boldsymbol{B})$ of the form* (*FLASH-MAX*) *such that*

$$\|(\boldsymbol{E}_0, \boldsymbol{B}_0)(0, \cdot) - (\boldsymbol{E}, \boldsymbol{B})(0, \cdot))\|_{H^m(\Omega_T)} < \varepsilon,$$

so that we can further ensure

$$\|(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}})(t, \cdot) - (\boldsymbol{E}, \boldsymbol{B})(t, \cdot)\|_{H^m((0,T)\times\Omega)} < \varepsilon.$$

If $m > 1$, the second estimate holds true with the H^m-norm replaced by the C^{m-2}-norm.

In particular, given C^∞ initial conditions, we obtain approximations in the Frechét topology on $C^\infty((0, T) \times \Omega)$.

Before we prove this theorem, we need to ensure that our model makes sense, i.e. that (*FLASH-MAX*) indeed generates solutions of Maxwell's system (MAX).

Lemma 4. (*FLASH-MAX*) *gives solutions to Maxwell's equations* (MAX)*.*

Proof. By linearity of the equations, it suffices to verify this claim for

$$(\boldsymbol{E}(\boldsymbol{x}), \boldsymbol{B}(\boldsymbol{x})) = \sigma(\boldsymbol{x}^\top \boldsymbol{z}) \boldsymbol{p}_i(\boldsymbol{z})$$

where

$$z_0 = \pm\sqrt{z_1^2 + z_2^2 + z_3^2}$$

and

$$\boldsymbol{p}_1(\boldsymbol{z}) = (-z_1 z_3, -z_2 z_3, z_1^2 + z_2^2, -z_0 z_2, z_0 z_1, 0)$$
$$\boldsymbol{p}_2(\boldsymbol{z}) = (z_1 z_2, -z_1^2 - z_3^2, z_2 z_3, -z_0 z_3, 0, z_0 z_1).$$

We thus write $z^\pm := (\pm|\xi|, \xi)$ for $\xi \in \mathbb{R}^3$, so that a direct computation gives

$$\boldsymbol{p}_1(z^\pm) = \big(-\xi \times (\xi \times e_3),\ \mp|\xi|(\xi \times e_3)\big), \qquad \boldsymbol{p}_2(z^\pm) = \big(\xi \times (\xi \times e_2),\ \pm|\xi|(\xi \times e_2)\big),$$

First let $i = 1$. By the chain rule, we obtain

$$\partial_t \boldsymbol{E} - \nabla \times \boldsymbol{B} = \sigma'(\boldsymbol{x}^\top z^\pm)(\pm|\xi|(-\xi \times (\xi \times e_3)) - \xi \times (\mp|\xi|(\xi \times e_3)) = \boldsymbol{0}$$
$$\partial_t \boldsymbol{B} + \nabla \times \boldsymbol{E} = \sigma'(\boldsymbol{x}^\top z^\pm)(\pm|\xi|(\mp|\xi|(\xi \times e_3) + \xi \times (-\xi \times (\xi \times e_3))) = \boldsymbol{0}$$
$$\nabla \cdot \boldsymbol{E} = \sigma'(\boldsymbol{x}^\top z^\pm)(-\xi \times (\xi \times e_3)) \cdot \xi = \boldsymbol{0}$$
$$\nabla \cdot \boldsymbol{B} = \sigma'(\boldsymbol{x}^\top z^\pm)(\mp|\xi|(\xi \times e_3) \cdot \xi = \boldsymbol{0},$$

where all calculations follow by elementary vector calculus identities. The case $i = 2$ is similar. □

Proof of Theorem 2 assuming Theorem 3. If $(\boldsymbol{E}_{\rm gt}, \boldsymbol{B}_{\rm gt}) \in H^m(\mathbb{R}^4)$, then $(\boldsymbol{E}_{\rm gt}(t,\cdot), \boldsymbol{B}_{\rm gt}(t,\cdot)) \in H^m(\mathbb{R}^3)$ for almost every $t \in \mathbb{R}$. This follows by Fubini's theorem. Pick such a $t_0 \le t_1$. By translation invariance of linear pde with constant coefficients, we can assume that $t = 0$ and $t_2 = T$ in the statement of Theorem 3. By replacing Ω with its convex hull, we can assume that Ω is convex. We also let $\boldsymbol{E}_0 = \boldsymbol{E}_{\rm gt}(0,\cdot)$ and $\boldsymbol{B}_0 = \boldsymbol{B}_{\rm gt}(0,\cdot)$. Apply Theorem 3 to conclude. □

Of course, the difficulty is moved to the:

Proof of Theorem 3. We first establish the approximation of $\boldsymbol{E}_0, \boldsymbol{B}_0$ in H^m using activation function $\sigma = \cos$. Then we extend the approximation to general σ. Finally, we recall estimates for (MAX) that enable us to control the spacetime error by the error at $t = 0$.

Step 1. We can assume $\boldsymbol{E}_0, \boldsymbol{B}_0 \in L^2(\mathbb{R}^3)$ and $\boldsymbol{E}_0 = \boldsymbol{0} = \boldsymbol{B}_0$ outside a cube $Q \subset \mathbb{R}^3$, without loss of generality:
By [85], there exist $\tilde{\boldsymbol{E}}_0, \tilde{\boldsymbol{B}}_0 \in L^2(\mathbb{R}^3)$ that are both divergence-free in $\mathbb{R}^3$ and each equal $\boldsymbol{E}_0, \boldsymbol{B}_0$ in Ω_T. By Poincaré's lemma, $\tilde{\boldsymbol{E}}_0 = \nabla \times v_1$, $\tilde{\boldsymbol{B}}_0 = \nabla \times v_2$ for some $v_1,\ v_2 \in H^1(\mathbb{R}^3)$. Consider $\rho \in C_c^\infty(Q)$ such that $\rho = 1$ in Ω_T and $Q \ni \Omega_T$ is a sufficiently large cube. Then $e_0 = \nabla \times (\rho v_1)$, $b_0 = \nabla \times (\rho v_2)$ are divergence-free in $\mathbb{R}^3$, compactly supported inside Q, equal $\boldsymbol{E}_0, \boldsymbol{B}_0$ respectively in Ω_T. Let (e, b) be the unique solution in $L^2((0,T) \times \mathbb{R}^3)$ of (MAX) with initial conditions e_0, b_0. By the finite speed of propagation property (see also *Step 4* below), we have that $e = \boldsymbol{E}_{\rm gt}$ and $b = \boldsymbol{B}_{\rm gt}$ in $(0,T) \times \Omega$.

Step 2. $\boldsymbol{E}_0, \boldsymbol{B}_0$ can be approximated in H^m by Fourier sums with "good" frequencies:
By the translation-, dilation-, and rotation-invariance of (MAX), we can assume the cube $Q = (0, 2\pi)^3$, so that we can write the Fourier series

$$\boldsymbol{E}_0(x,y,z) = \sum_{\boldsymbol{k} \in \mathbb{Z}^3} c_{\boldsymbol{k}} \exp(\sqrt{-1}(xk_1 + yk_2 + zk_3))$$

where $c_{\boldsymbol{k}} \in \mathbb{C}^3$ are complex coefficients, with a similar formula for $\boldsymbol{B}_0$.

We next use the fact that $\boldsymbol{E}_0$ is divergence-free to conclude, first, that $c_{\boldsymbol{0}} = \boldsymbol{0}$. This is so since

$$(2\pi)^3 c_{\boldsymbol{0}} = \int_Q \boldsymbol{E}_0 dV = \int_Q \nabla(x,y,z)\boldsymbol{E}_0 dV = -\int_Q (x,y,z)\nabla \cdot \boldsymbol{E}_0 dV = \boldsymbol{0}.$$

We moreover have

$$0 = \nabla \cdot \boldsymbol{E}_0(x,y,z) = \sqrt{-1} \sum_{\boldsymbol{0} \neq \boldsymbol{k} \in \mathbb{Z}^3} c_{\boldsymbol{k}} \cdot \boldsymbol{k} \exp(-\sqrt{-1}(xk_1 + yk_2 + zk_3)),$$

so that the scalar product $c_{\boldsymbol{k}} \cdot \boldsymbol{k} = 0$ for all $\mathbf{0} \neq \boldsymbol{k} \in \mathbb{Z}^3$ by linear independence over $\mathbb{R}$ of sine and cosine waves. In particular, if $c_{\boldsymbol{k}} = a_{\boldsymbol{k}} + \sqrt{-1} b_{\boldsymbol{k}}$ for $a_{\boldsymbol{k}}, b_{\boldsymbol{k}} \in \mathbb{R}^3$, then $a_{\boldsymbol{k}}, b_{\boldsymbol{k}} \in \boldsymbol{k}^\perp$ for any $\mathbf{0} \neq \boldsymbol{k} \in \mathbb{Z}^3$. After taking real part we are left with

$$\boldsymbol{E}_0(x,y,z) = \sum_{\mathbf{0}\neq\boldsymbol{k}\in\mathbb{Z}^3} a_{\boldsymbol{k}} \cos(xk_1 + yk_2 + zk_3) - b_{\boldsymbol{k}} \sin(xk_1 + yk_2 + zk_3),$$

and $\sin$ is a shift of $\cos$ by $\pi/2$. We have a similar formula for B_0,

$$\boldsymbol{B}_0(x,y,z) = \sum_{\mathbf{0}\neq\boldsymbol{k}\in\mathbb{Z}^3} \tilde{a}_{\boldsymbol{k}} \cos(xk_1 + yk_2 + zk_3) - \tilde{b}_{\boldsymbol{k}} \sin(xk_1 + yk_2 + zk_3),$$

with $\tilde{a}_{\boldsymbol{k}}, \tilde{b}_{\boldsymbol{k}} \in \boldsymbol{k}^\perp$. We denote by $\boldsymbol{E}_{0j}, \boldsymbol{B}_{0j}$ the partial sums of $\boldsymbol{E}_0, \boldsymbol{B}_0$ above, truncated at $|\boldsymbol{k}| \leq j$. Thus $\boldsymbol{E}_{0j}, \boldsymbol{B}_{0j}$ are smooth divergence-free fields, which converge respectively to $\boldsymbol{E}_0, \boldsymbol{B}_0$ in $H^m(Q, \mathbb{R}^3)$ as $j \to \infty$.

We next approximate $\boldsymbol{E}_{0j}, \boldsymbol{B}_{0j}$ with similar trigonometric polynomials such that no frequency used has $k_1 = 0$.[3] To this end, we will show that any derivative ∂^α, $|\alpha| \leq m$ of the term $a\cos(k_2 y + k_3 z)$ with $a \in \mathbb{R}^3$ and $k_2^2 + k_3^2 \leq j^2$ can be approximated uniformly with a good frequency. Note that $a_2 k_2 + a_3 k_3 = 0$ and one of k_2, k_3 must be non-zero. Suppose without loss of generality that $k_2 \neq 0$. Note that for any $\delta \in (0,1)$

$$\nabla \cdot [(a_1, a_2 - a_1\delta/k_2, a_3)^\top \cos(\delta x + k_2 y + k_3 z)] = 0.$$

We aim to estimate for $\alpha \in \mathbb{N}^3$ with $|\alpha| \leq m$ the difference

$$\mathbf{I}_\alpha = |\partial^\alpha [(a_1, a_2 - a_1\delta/k_2, a_3)^\top \cos(\delta x + k_2 y + k_3 z) - a\cos(k_2 y + k_3 z)]|.$$

To do so, we introduce a bit of notation. We write f_k for the kth derivative of f. We write $v_1 = (\delta, k_2, k_3)^\top$, $v_2 = (0, k_2, k_3)^\top$, $\vec{x} = (x,y,z)^\top$, and $\tilde{a} = (a_1, a_2 - a_1\delta/k_2, a_3)^\top$. We then have by chain rule

$$\begin{aligned}
\mathbf{I}_\alpha &= |\partial^\alpha [\tilde{a}\cos(v_1^\top \vec{x}) - a\cos(v_2^\top \vec{x})]| \\
&= |v_1^\alpha \tilde{a} \cos_{|\alpha|}(v_1^\top \vec{x}) - v_2^\alpha a \cos_{|\alpha|}(v_2^\top \vec{x})| \\
&\leq |v_1^\alpha \tilde{a} \cos_{|\alpha|}(v_1^\top \vec{x}) - v_1^\alpha a \cos_{|\alpha|}(v_1^\top \vec{x})| + |v_1^\alpha a \cos_{|\alpha|}(v_1^\top \vec{x}) - v_2^\alpha a \cos_{|\alpha|}(v_1^\top \vec{x})| \\
&\quad + |v_2^\alpha a \cos_{|\alpha|}(v_1^\top \vec{x}) - v_2^\alpha a \cos_{|\alpha|}(v_2^\top \vec{x})| \\
&\leq |v_1^\alpha||\tilde{a} - a| + |v_1 - v_2||a| + |v_2^\alpha||a||\delta x| \\
&\leq j^m |a|\delta + \delta|a| + 2\pi\delta|a||j|^m \\
&\leq 10\delta|a|j^m
\end{aligned}$$

where we used that $|k_2| \geq 1$ and the fact that $\cos$ is 1-Lipschitz. It follows that

$$\|\tilde{a}\cos(v_1^\top \cdot) - a\cos(v_2^\top \cdot)\|_{C^m(Q)} \leq C_0(m)\delta|a|j^m;$$

In particular, replacing the terms in the *finite* expansions of $\boldsymbol{E}_{0j}, \boldsymbol{B}_{0j}$ with frequencies $\boldsymbol{k}$ with $k_1 = 0$ as above by choosing δ small enough, we can obtain

$$\|(\boldsymbol{E}_{0j}, \boldsymbol{B}_{0j}) - (\tilde{\boldsymbol{E}}_{0j}, \tilde{\boldsymbol{B}}_{0j})\|_{C^m(Q)} \leq 1/j,$$

where

$$(\tilde{\boldsymbol{E}}_{0j}, \tilde{\boldsymbol{B}}_{0j})^\top = \sum_{\xi\in F_j} A_\xi \cos(\xi_1 x + \xi_2 y + \xi_3 z) + B_\xi \sin(\xi_1 x + \xi_2 y + \xi_3 z)$$

with $\xi_1 \neq 0$ for all ξ in a finite set $F_j \subset \mathbb{R}^3 \setminus \{\mathbf{0}\}$ and $A_\xi, B_\xi \in \ker R(\xi)$, where

$$R(\xi) = \begin{pmatrix} \xi_1 & \xi_2 & \xi_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \xi_1 & \xi_2 & \xi_3 \end{pmatrix}.$$

Then $(\tilde{\boldsymbol{E}}_{0j}, \tilde{\boldsymbol{B}}_{0j})$ give the desired finite approximations of $(\boldsymbol{E}_0, \boldsymbol{B}_0)$ in $H^m(Q, \mathbb{R}^6)$.

[3] The reason why this is necessary is far from transparent and will become clear in *Step 3*. In short, the vectors $\boldsymbol{p}_i(\boldsymbol{z}) \in \mathbb{R}^6$ from (2) contain too many zeroes at $\boldsymbol{z} \in \mathbb{R}^4$ with $\boldsymbol{z}_1 = 0$.

Step 3. $(\boldsymbol{E}_0, \boldsymbol{B}_0)$ *can be approximated in* H^m *by (FLASH-MAX) with* $\sigma = \cos$ *at* $t = 0$*:*
To achieve this, we will express $(\tilde{\boldsymbol{E}}_{0j}, \tilde{\boldsymbol{B}}_{0j})$ in terms of

$$P(\xi) = (p_1(|\xi|, \xi)^\top,\, p_2(|\xi|, \xi)^\top,\, p_1(-|\xi|, \xi)^\top,\, p_2(-|\xi|, \xi)^\top) \in \mathbb{R}^{6\times 4}.$$

with p_i as in (2).

We claim that $\operatorname{im} P(\xi) = \ker R(\xi)$ if $\xi_1 \neq 0$. To this end, we note that the inclusion $\subset$ follows from a simple computation. Once we prove that $\ker P(\xi) = \{\mathbf{0}\}$, the rank-nullity theorem implies that $\operatorname{rank} P(\xi) = 4$, which clearly equals the dimension of $\ker R(\xi)$, yielding the claim. To show that $\ker P(\xi) = \{\mathbf{0}\}$ for $\xi_1 \neq 0$, we simply write

$$\alpha_1 p_1(|\xi|, \xi) + \alpha_2 p_2(|\xi|, \xi) + \alpha_3 p_1(-|\xi|, \xi) + \alpha_4 p_2(-|\xi|, \xi) = \mathbf{0}.$$

Equation (2) implies $\alpha_1 = \alpha_3$, $\alpha_2 = \alpha_4$. If the $(1,1),(3,2)$-minor is nonzero, we are done. Else, $\xi_2 = 0$ and then the $(2,1),(3,2)$ minor is nonzero, which proves the claim.

Let

$$\begin{aligned} v_j(x,y,z) = \sum_{\xi\in F_j} &[P(\xi)^\top P(\xi)]^{-1} P(\xi)^\top A_\xi \cos(x\xi_1 + y\xi_2 + z\xi_3) \\ &+ [P(\xi)^\top P(\xi)]^{-1} P(\xi)^\top B_\xi \sin(x\xi_1 + y\xi_2 + z\xi_3), \end{aligned}$$

where the left-inverse is actually an inverse because $\ker R(\xi) = \operatorname{im} P(\xi)$ for $\xi_1 \neq 0$. This implies that

$$\begin{aligned} &\sum_{\xi\in F_j} P(\xi)[P(\xi)^\top P(\xi)]^{-1} P(\xi)^\top A_\xi \cos(x\xi_1 + y\xi_2 + z\xi_3) \\ &\quad + P(\xi)[P(\xi)^\top P(\xi)]^{-1} P(\xi)^\top B_\xi \sin(x\xi_1 + y\xi_2 + z\xi_3) = (\tilde{\boldsymbol{E}}_{0j}, \tilde{\boldsymbol{B}}_{0j})^\top (x,y,z), \end{aligned} \tag{4}$$

which means that indeed arbitrary divergence-free initial conditions can be approximated in H^m using (*FLASH-MAX*) with activation function $\sigma = \cos$.

Step 4. General activation function for H^m*-approximation at* $t = 0$*:*
Because the sum in (4) is finite, it suffices to approximate a single term

$$v(\vec{x}) = P(\xi) A \cos(\xi^\top \vec{x} + b)$$

in $H^m(Q)$ using an arbitrary activation σ. In particular, by our assumption on σ not being a polynomial, we can use [86] to find parameters $w_j,\, z_j,\, b_j \in \mathbb{R}$ such that

$$\left\| \cos - \sum_{j=1}^{W} w_j \sigma(z_j \cdot + b_j) \right\|_{H^m(-4\pi|\xi|+b, 4\pi|\xi|+b)} < \delta.$$

Our approximation for v will then be

$$\tilde{v}(\vec{x}) = \sum_{j=1}^{W} w_j P(\xi) A \sigma(z_j \xi^\top \vec{x} + z_j b + b_j) = \sum_{j=1}^{W} P(z_j\xi)(w_j z_j^{-2} A)\sigma(z_j \xi^\top \vec{x} + \underbrace{z_j b + b_j}_{B_j}),$$

so $\tilde{v}$ is in the form (*FLASH-MAX*) at $t = 0$. We will next show that we can control the α-derivative of the error in approximating v with the error of approximating the α-derivative of cos. Let $\alpha \in \mathbb{N}^3$.

$$\begin{aligned} &\|\partial^\alpha (v - \tilde{v})\|^2_{L^2(Q)} \\ &= \int_Q \left| \xi^\alpha P(\xi) A \left(\cos_{|\alpha|}(\xi^\top \vec{x} + b) - \sum_{j=1}^{W} w_j z_j^{|\alpha|} \sigma_{|\alpha|}(z_j \xi^\top \vec{x} + z_j b + b_j) \right) \right|^2 dV \\ &\leq |\xi^\alpha P(\xi) A|^2 \int_{(-4\pi, 4\pi)^3} \left| \cos_{|\alpha|}(|\xi| t_1 + b) - \sum_{j=1}^{W} w_j z_j^{|\alpha|} \sigma_{|\alpha|}(z_j(|\xi| t_1 + b) + b_j) \right|^2 dt_1 dt_2 dt_3 \\ &\leq 64\pi^2 |\xi^\alpha P(\xi) A|^2 |\xi|^{-1} \int_{-4\pi|\xi|+b}^{4\pi|\xi|+b} \left| \cos_{|\alpha|}(s) - \sum_{j=1}^{W} w_j z_j^{|\alpha|} \sigma_{|\alpha|}(z_j s + b_j) \right|^2 ds \\ &\leq 64\pi^2 |\xi^\alpha P(\xi) A|^2 |\xi|^{-1} \delta^2. \end{aligned}$$

where the first change of variable is a rotation making ξ the direction of the t_1 axis (4π is a generous bound to ensure that the rotation of Q is contained in $(-4\pi, 4\pi)^3$) and the last inequality follows from the approximation of cosine. By summing these estimates of $\partial^\alpha(v - \tilde{v})$ for $|\alpha| \le m$, we can conclude that $\|v - \tilde{v}\|_{H^m(Q)}$ can be made arbitrarily small.

Step 5. Dependence of solutions of Maxwell's equations on initial conditions in H^m or C^m: Let $(\boldsymbol{E}, \boldsymbol{B}) \in L^2(\mathbb{R}^3, \mathbb{R}^6)$ solve (MAX), and let $\hat{\boldsymbol{E}}(\xi, t), \hat{\boldsymbol{B}}(\xi, t)$ denote the spatial Fourier transform of $\boldsymbol{E}, \boldsymbol{B}$, normalized so that Plancherel's theorem reads $\|f\|_{L^2(\mathbb{R}^3)} = \|\hat{f}\|_{L^2(\mathbb{R}^3)}$. Let

$$[\xi]_\times := \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix}$$

denote the real, skew-symmetric matrix representing $v \mapsto \xi \times v$. Writing $z(\xi, t) = (\hat{\boldsymbol{E}}, \hat{\boldsymbol{B}})(\xi, t) \in \mathbb{C}^6$, (MAX) becomes, for a.e. $\xi \in \mathbb{R}^3$, the linear ODE

$$\partial_t z(\xi, t) = A(\xi) z(\xi, t), \qquad A(\xi) := \begin{pmatrix} 0 & \sqrt{-1}\,[\xi]_\times \\ -\sqrt{-1}\,[\xi]_\times & 0 \end{pmatrix}.$$

Writing $M^* := \bar{M}^\top$ for the conjugate transpose, a direct computation using $[\xi]_\times^\top = -[\xi]_\times$ gives $A(\xi)^* = -A(\xi)$ for every ξ, i.e. $A(\xi)$ is skew-Hermitian. Consequently

$$\partial_t |z(\xi, t)|^2 = z(\xi, t)^* A(\xi)^* z(\xi, t) + z(\xi, t)^* A(\xi) z(\xi, t) = 0,$$

so $|z(\xi, t)| = |z(\xi, 0)|$ for every t and a.e. ξ. By Plancherel's theorem,

$$\int_{\mathbb{R}^3} |\boldsymbol{E}(t, \cdot)|^2 + |\boldsymbol{B}(t, \cdot)|^2 dV = \int_{\mathbb{R}^3} |z(\xi, t)|^2 d\xi = \int_{\mathbb{R}^3} |z(\xi, 0)|^2 d\xi = \int_{\mathbb{R}^3} |\boldsymbol{E}_0|^2 + |\boldsymbol{B}_0|^2 dV$$

for every t; note that no differentiability of $(\boldsymbol{E}, \boldsymbol{B})$ in space, and no compact support of $(\boldsymbol{E}, \boldsymbol{B})(t, \cdot)$, is needed for this identity. In particular,

$$\sup_{t \in (0,T)} \int_{\mathbb{R}^3} |\boldsymbol{E}(t, \cdot)|^2 + |\boldsymbol{B}(t, \cdot)|^2 dV \le \int_{\mathbb{R}^3} |\boldsymbol{E}_0|^2 + |\boldsymbol{B}_0|^2 dV. \tag{5}$$

Since all derivatives of $(\boldsymbol{E}, \boldsymbol{B})$ are also solutions of (MAX), we can consider $\alpha \in \mathbb{N}^3$ with $|\alpha| \le m$ and only spatial derivatives $\partial = (\partial_x, \partial_y, \partial_z)$ to obtain directly from (5) that

$$\sup_{t \in (0,T)} \int_{\mathbb{R}^3} |\partial^\alpha \boldsymbol{E}(t, \cdot)|^2 + |\partial^\alpha \boldsymbol{B}(t, \cdot)|^2 dV \le \int_{\mathbb{R}^3} |\partial^\alpha \boldsymbol{E}_0|^2 + |\partial^\alpha \boldsymbol{B}_0|^2 dV,$$

which implies

$$\sup_{t \in (0,T)} \|(\boldsymbol{E}, \boldsymbol{B})(t, \cdot)\|_{H^m(\Omega)} \le \|(\boldsymbol{E}_0, \boldsymbol{B}_0)\|_{H^m(\Omega_T)}. \tag{6}$$

We next incorporate time derivatives. Let $a \in \mathbb{N}$ and $\alpha \in \mathbb{N}^3$ be such that $a + |\alpha| \le m$. We claim that $\partial_t^a \partial^\alpha (\boldsymbol{E}, \boldsymbol{B})$ can be expressed only in terms of spatial derivatives of $(\boldsymbol{E}, \boldsymbol{B})$. To this end, we note that by direct calculation we have for integers $j \ge 0$

$$\partial_t^{2j}(\boldsymbol{E}, \boldsymbol{B}) = \Delta^j (\boldsymbol{E}, \boldsymbol{B}) \quad \text{so} \quad \partial_t^{2j+1}(\boldsymbol{E}, \boldsymbol{B}) = \Delta(\nabla \times \boldsymbol{B}, -\nabla \times \boldsymbol{E}),$$

where $\Delta = \partial_x^2 + \partial_y^2 + \partial_z^2$. Therefore we can express

$$\partial_t^a \partial^\alpha (\boldsymbol{E}, \boldsymbol{B}) = A_{a,\alpha}(\partial)(\boldsymbol{E}, \boldsymbol{B}),$$

where $A_{a,\alpha}$ is a linear partial differential operator in spatial variables only with constant coefficients and having order $a + |\alpha| \le m$. In particular we can estimate

$$\begin{aligned} \|\partial_t^a \partial^\alpha (\boldsymbol{E}, \boldsymbol{B})\|_{L^2((0,T)\times\Omega)} &= \|A_{a,\alpha}(\partial)(\boldsymbol{E}, \boldsymbol{B})\|_{L^2((0,T)\times\Omega)} \\ &\le \sqrt{T} \sup_{t \in (0,T)} \|A_{a,\alpha}(\partial)(\boldsymbol{E}, \boldsymbol{B})(t, \cdot)\|_{L^2(\Omega)} \\ &\le C_1(m)\sqrt{T} \sup_{t \in (0,T)} \|(\boldsymbol{E}, \boldsymbol{B})(t, \cdot)\|_{H^m(\Omega)} \\ &\le C_1(m)\sqrt{T} \|(\boldsymbol{E}_0, \boldsymbol{B}_0)\|_{H^m(\Omega_T)}. \end{aligned}$$

where $C(m) > 0$ is a constant depending on m only and the last inequality follows from (6). Since $a + |\alpha| \leq m$ we can infer that

$$\|(\boldsymbol{E}, \boldsymbol{B})\|_{H^m((0,T)\times\Omega)} \leq C_2(m)\sqrt{T}\|(\boldsymbol{E}_0, \boldsymbol{B}_0)\|_{H^m(\Omega_T)} \tag{7}$$

We derive a similar estimate on the C^m-scale. We write for $a + |\alpha| \leq m$

$$\begin{aligned}
\|\partial_t^a \partial^\alpha (\boldsymbol{E}, \boldsymbol{B})\|_{C^0((0,T)\times\Omega)} &= \|A_{a,\alpha}(\partial)(\boldsymbol{E}, \boldsymbol{B})\|_{C^0((0,T)\times\Omega)} \\
&= \sup_{t\in(0,T)} \|A_{a,\alpha}(\partial)(\boldsymbol{E}, \boldsymbol{B})(t,\cdot)\|_{C^0(\Omega)} \\
&\leq C_3(m) \sup_{t\in(0,T)} \|(\boldsymbol{E}, \boldsymbol{B})(t,\cdot)\|_{C^m(\Omega)} \\
&\leq C_1(m,\Omega) \sup_{t\in(0,T)} \|(\boldsymbol{E}, \boldsymbol{B})(t,\cdot)\|_{H^{m+2}(\Omega)} \\
&\leq C_1(m,\Omega)\|(\boldsymbol{E}_0, \boldsymbol{B}_0)\|_{H^{m+2}(\Omega_T)}.
\end{aligned}$$

where the penultimate inequality follows from the Morrey–Sobolev embedding and the last inequality follows from (6). Finally, we get

$$\|(\boldsymbol{E}, \boldsymbol{B})\|_{C^m((0,T)\times\Omega)} \leq C_2(m,\Omega)\|(\boldsymbol{E}_0, \boldsymbol{B}_0)\|_{H^{m+2}(\Omega_T)}. \tag{8}$$

Step 6. Conclusion in H^m spaces:
We now collect most of the facts above. Let $\varepsilon > 0$, $m \geq 0$, and $\boldsymbol{E}_0, \boldsymbol{B}_0 \in H^m(\Omega_T)$ be divergence-free. Let $\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}} \in H^m((0,T)\times\Omega)$ be the unique solution of Maxwell equations (MAX) with initial conditions $\boldsymbol{E}_0, \boldsymbol{B}_0$.

According to *Steps 1-4*, there exists a finite neural network (*FLASH-MAX*) which we name here $(\boldsymbol{E}, \boldsymbol{B})$ such that

$$\|(\boldsymbol{E}_0, \boldsymbol{B}_0) - (\boldsymbol{E}, \boldsymbol{B})(0,\cdot)\|_{H^m(\Omega)} < \varepsilon. \tag{9}$$

Since $(\boldsymbol{E}, \boldsymbol{B})$ solves (MAX), so does its difference to $(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}})$, by linearity. We apply *Step 1*'s construction to the error in the initial data. Let

$$(\boldsymbol{E}_0', \boldsymbol{B}_0') := (\boldsymbol{E}, \boldsymbol{B})(0,\cdot) - (\boldsymbol{E}_0, \boldsymbol{B}_0),$$

restricted to Ω_T; this pair is divergence-free and, by *Steps 2-4* (whose $H^m(Q)$-convergence, $Q \Supset \Omega_T$, gives (9) with Ω replaceable by Ω_T), has $H^m(\Omega_T)$-norm $< \varepsilon$. *Step 1*'s construction, applied to $(\boldsymbol{E}_0', \boldsymbol{B}_0')$ instead of to $(\boldsymbol{E}_0, \boldsymbol{B}_0)$, yields a compactly supported, divergence-free field in $H^m(\mathbb{R}^3, \mathbb{R}^6)$, which we continue to denote $(\boldsymbol{E}_0', \boldsymbol{B}_0')$; since the construction is continuous and maps $\mathbf{0} \mapsto \mathbf{0}$, $\|(\boldsymbol{E}_0', \boldsymbol{B}_0')\|_{H^m(\mathbb{R}^3)} < C_{\mathrm{ext}}(m)\varepsilon$ for some constant $C_{\mathrm{ext}}(m)$. Let $(\boldsymbol{E}', \boldsymbol{B}')$ solve (MAX) with initial condition $(\boldsymbol{E}_0', \boldsymbol{B}_0')$; by *Step 5*,

$$\|(\boldsymbol{E}', \boldsymbol{B}')\|_{H^m((0,T)\times\mathbb{R}^3)} < C_2(m)C_{\mathrm{ext}}(m)\sqrt{T}\varepsilon.$$

Since $(\boldsymbol{E}_0', \boldsymbol{B}_0') = (\boldsymbol{E}, \boldsymbol{B})(0,\cdot) - (\boldsymbol{E}_0, \boldsymbol{B}_0)$ on Ω_T, finite speed of propagation (*Step 1*) gives $(\boldsymbol{E}', \boldsymbol{B}') = (\boldsymbol{E}, \boldsymbol{B}) - (\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}})$ on $(0,T)\times\Omega$, so that

$$\|(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}}) - (\boldsymbol{E}, \boldsymbol{B})\|_{H^m((0,T)\times\Omega)} \leq \|(\boldsymbol{E}', \boldsymbol{B}')\|_{H^m((0,T)\times\mathbb{R}^3)} < C_2(m)C_{\mathrm{ext}}(m)\sqrt{T}\varepsilon.$$

Perhaps by making ε smaller, we obtain the claim.

Step 7. Conclusion in C^m spaces:
Now let $m > 1$ and assume that the initial condition is in $C^m(\Omega_T) \subset H^m(\Omega_T)$. As in *Step 6*, we work with the auxiliary field $(\boldsymbol{E}', \boldsymbol{B}')$ rather than with $(\boldsymbol{E}, \boldsymbol{B}) - (\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}})$ directly, since the latter need not be compactly supported. The same argument establishing (8) (Morrey–Sobolev embedding, applied with $\mathbb{R}^3$ in place of Ω) gives

$$\|(\boldsymbol{E}', \boldsymbol{B}')\|_{C^{m-2}((0,T)\times\mathbb{R}^3)} < C_2(m,\mathbb{R}^3)\|(\boldsymbol{E}_0', \boldsymbol{B}_0')\|_{H^m(\mathbb{R}^3)} < C_2(m,\mathbb{R}^3)C_{\mathrm{ext}}(m)\varepsilon,$$

and since $(\boldsymbol{E}', \boldsymbol{B}') = (\boldsymbol{E}, \boldsymbol{B}) - (\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}})$ on $(0,T)\times\Omega$ (*Step 6*),

$$\|(\boldsymbol{E}_{\mathrm{gt}}, \boldsymbol{B}_{\mathrm{gt}}) - (\boldsymbol{E}, \boldsymbol{B})\|_{C^{m-2}((0,T)\times\Omega)} \leq \|(\boldsymbol{E}', \boldsymbol{B}')\|_{C^{m-2}((0,T)\times\mathbb{R}^3)} < C_2(m,\mathbb{R}^3)C_{\mathrm{ext}}(m)\varepsilon.$$

This concludes the proof of our main result.

□

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims align with the method and results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of our approach are discussed in the conclusions section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The full set of assumptions and complete correct proof can be found in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All details required to reproduce experimental results are available in the manuscript, collected in Appendix B.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is provided with the submission and will be made publicly available upon acceptance. Code will be accompanied by sufficient instructions to reproduce main experimental results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The main manuscript and supplementary material provide all details necessary to replicate experiments and understand results, collected in Appendix B.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the standard deviation of five trained models for tables and figures.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our experiments section clearly states computational resources used to run the experiments, the amount of data and parameters needed to run the experiments, and the time of execution.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Societal impacts NA to the work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not involve releasing new models or datasets. We propose an approach of moving governing structure from the loss into the hypothesis class and show that this can improve the trade-off between physical consistency and computational efficiency.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Code is the only asset used in this work. It is properly cited, and their licenses and terms of use are respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code is provided with clear and detailed documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not include crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not include research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used to develop the methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.