

Article

Positivity-Preserving Neural Surrogates for Reduced-Precision Inference of a Dynamic-Energy-Budget Angiogenesis Model

Pasquale De Luca 

Department of Promotion of Human Sciences and Quality of Life, San Raffaele University of Rome,
Via di Val Cannuta 247, 00166 Rome, Italy; pasquale.deluca@uniroma5.it

Abstract

Neural surrogates of evolution equations replace many solver steps by one network evaluation, but the physical invariants they appear to respect are properties of the trained weights, and weights are what reduced-precision and integer inference perturb. We study a six-field reaction, diffusion and taxis model of angiogenesis coupled to a Dynamic Energy Budget reserve, whose fields are densities and must stay nonnegative, and we build a macro-step surrogate whose forward pass factors through operators with known sign structure. The diffusion propagator is assembled once as a dense nonnegative row-stochastic matrix, the haptotaxis step is written with explicitly nonnegative coefficients, and the learned block returns a production and a per capita destruction that enter a Modified–Patankar quotient. Every operation is then a sum or a product of nonnegative numbers, or a quotient of a nonnegative number by a positive one, so the state stays nonnegative under any arithmetic whose rounding preserves the sign of nonnegative reals. Positivity survives half precision, bfloat16 and integer quantization without clamping and without retraining, which we confirm against an unstructured surrogate sharing the backbone, the parameter count and the training protocol. Measurements on a graphics processor quantify what each format costs in accuracy, time and memory.

Keywords: structure-preserving neural networks; quantized inference; positivity preservation; Modified–Patankar methods; reaction–diffusion–taxis systems; surrogate models; edge inference; dynamic energy budget

MSC: 65M12; 68T07; 65G50; 92C17



Academic Editor: Juan Ramón Torregrosa Sánchez

Received: 16 August 2026

Revised: 8 September 2026

Accepted: 10 September 2026

Published: 20 September 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

A neural surrogate of a time-dependent partial differential equation is a learned map that advances the discrete state over an interval that a conventional integrator would cross in many steps. The appeal is plain. Once trained, the surrogate turns a sequential and stiffness-limited computation into a small number of dense algebraic kernels, which is what current accelerators execute best, and the same trained map serves an entire family of parameters, so the cost of a calibration sweep or of a many-query design study falls by the ratio between the two step sizes. The literature that established this pattern for fluid and reaction problems is by now substantial [1–8].

What is far less settled is what such a map guarantees. The equations that describe interacting populations, concentrations and energy densities are not merely equations whose solutions happen to be nonnegative. Their unknowns are densities, the reaction terms are sums of productions and of per capita losses, and a negative value is not an

inaccurate answer but a meaningless one, which propagates through saturating nonlinearities and destroys the subsequent dynamics. Numerical analysis has answered this with schemes that preserve the sign of the discrete solution unconditionally, among which the Modified–Patankar family occupies a central place [9–14], and with exponential integrators that advance Metzler transport operators exactly [15–17]. Machine learning has answered it mostly with penalties, and more recently with architectures that build a conservation law or a bound into the forward pass [18–22]. The distinction between the two answers matters more than it appears. A penalty makes the invariant a property of the particular weight vector that training produced. An architectural constraint makes it a property of the map, whatever the weights.

Three lines of work meet in this paper and it is worth separating them. The first learns the solution operator of an evolution equation, either as a neural operator trained on solution data [2,3,7] or as a correction to a coarse solver kept in the training loop [4–6], and the macro-step map studied here belongs to that family. The second builds a physical property into the architecture rather than into the loss, as with symplectic and Hamiltonian networks [20,22], with divergence-free parameterizations [19] and with the analytic constraints of [18], and the Modified–Patankar quotient used below is the analogue of those constructions for the sign of a density. The third is the engineering of inference under a budget, where quantization, pruning and accelerator-aware design are routine [23–31], and where the numerical analysis of reduced precision has its own literature [32–35]. What is missing at the intersection is a statement of the form that an invariant built into an architecture is still an invariant after the model has been quantized, which is what Theorem 1 provides for this construction.

A branch of the Modified–Patankar literature preserves the linear invariants of a production and destruction system exactly as well as its sign. There the production and the destruction terms are written as a matrix of pairwise fluxes, and the same weight that appears as a gain of one species appears as a loss of another, so that the discrete update conserves the total mass whatever the step size [9,11,12,36–39]. The construction below inherits the sign structure of that family and not its conservation. The reason is structural rather than incidental: the production and the destruction of the learned block are two groups of channels of a network, and nothing ties the gain of one field to the loss of another, so no linear invariant of the reaction is exact by construction. The energy budget is therefore measured here and not proved, the transport block alone is conservative by Proposition 1, and a conservative Patankar parameterization of the learned coefficients is the natural continuation, discussed in Section 6.

On the side of the arithmetic, the practice of preparing a network for a reduced format is quantization-aware training, in which the rounding is simulated in the forward pass and the weights, or the step sizes themselves, are learned through it [23,24,40–42], and reduced precision has an established use in scientific computing as well [33,43,44]. That literature optimizes the accuracy of the quantized model and is silent on its admissibility, and searching it for a structure-preserving surrogate of a partial differential equation trained in the deployed format returns nothing. The models compared in Section 4 are therefore quantized after training, which is the setting in which the difference between an architectural and a trained invariant is visible, and training in the target format is recommended in Section 5.2 as the way to recover the accuracy that four bits cost, not the admissibility, which costs nothing.

That distinction becomes a practical question the moment the surrogate leaves the machine that trained it. Inference on accelerators, on embedded targets and on neuromorphic hardware [45,46] is performed in reduced floating-point formats or in integer arithmetic, and the entire discipline of quantization exists to make that transition cheap in accuracy and in energy [23–28,33,35,47]. Quantization perturbs exactly what a penalty-trained invariant

depends on, namely the numerical values of the weights and of the intermediate activations. An invariant that was verified in single precision on a workstation is not thereby verified in the format the deployed model will use, and the usual remedy of projecting the output onto the admissible set repairs the symptom at the cost of the balance laws the projection ignores.

This paper takes a model in which the question is unavoidable and answers it constructively. The model is a six-field system of reaction, diffusion and taxis type describing tumour-induced angiogenesis coupled to a Dynamic Energy Budget reserve, so that proliferation and proteolysis are funded by an internal energy store rather than responding to the instantaneous oxygen level [48–52]. All six unknowns are nonnegative densities, the reaction terms are in production and destruction form, and the reference integrator is a structure-preserving exponential Modified–Patankar scheme. The surrogate learns one macro step of this system, and its forward pass is designed so that the sign structure of the discrete operators is visible to the arithmetic rather than only to the mathematics.

It is worth separating what of the construction is standard from what is not. That the exponential of a Metzler matrix is entrywise nonnegative, that the exponential of a Neumann Laplacian is row-stochastic, and that both facts make an exponential propagator a natural transport block for a positive system, are classical [15,17,53], and so is the Modified–Patankar quotient as an unconditionally positive reaction update [9,10], and a network that embeds a splitting of a partial differential equation in its forward pass is the line the present construction continues [54]. None of that is claimed here. What is new is, first, the requirement that the propagator be applied as an assembled nonnegative matrix rather than through the transform that diagonalizes it, since the two are the same operator and only one of them is free of subtractive cancellation, which Section 5.3 measures; second, a transport block whose nonnegativity is a property of the implementation rather than a condition on the data, obtained by substepping the haptotaxis according to a criterion derived in Section 3.1 and clipping its diagonal; third, the statement that the composition of these blocks with a learned Patankar quotient keeps the state in the nonnegative orthant in finite precision and under integer quantization, which is Theorem 1 and is the point of the paper; and fourth, an a priori bound on the mass drift that quantizing the transport block introduces, together with the measurements that decide which block to quantize.

The contributions are the following. We give an architecture whose transport block is a fixed nonnegative row-stochastic matrix assembled once from the exact exponential of the Metzler diffusion operator and whose haptotaxis step is rewritten so that no subtraction acts on the state, while the learned block returns a nonnegative production and a nonnegative per capita destruction that enter a Modified–Patankar quotient. We prove that the resulting map keeps every field nonnegative for every weight vector and every macro step in exact arithmetic, and that the same conclusion holds in any arithmetic whose rounding sends nonnegative reals to nonnegative numbers, which covers the IEEE binary formats in every rounding mode as well as symmetric fixed-point quantization, so that no clamping and no retraining are needed when the model is deployed in a reduced format. We bound the mass drift that quantizing the transport matrix introduces, in terms of the quantization step alone, and we verify the bound. We then measure what the guarantee costs and what it buys, against an unstructured surrogate that shares the backbone, the parameter count and the training protocol, against its penalized variant, against its clamped variant and against a network without transport blocks, across floating-point and integer formats, on a doubled time horizon, on parameters drawn outside the training box, on perturbed initial data, and in wall-clock time, throughput, memory and stored size on a consumer graphics processor. Sections 2 and 3 state the model, the discrete operators and the architecture, and prove the guarantees. Sections 4 and 5 report the measurements.

The rest of the paper is organized as follows. Section 2 presents the six-field Dynamic Energy Budget angiogenesis system, the discrete operators of the reference integrator and the sign structure that the surrogate will inherit. Section 3 defines the macro-step architecture and proves the positivity, the finite-precision and the quantization results, together with the drift bound for the transport block. Section 4 reports accuracy and invariants across arithmetic formats and outside the training conditions. Section 5 reports the cost of inference on the target accelerator and the resulting deployment recommendation. Section 6 closes with the limitations and the outlook.

2. The Model and the Discrete Operators

2.1. The Six-Field System

The unknowns are the endothelial cell density C , the protease concentration P , the inhibitor concentration I , the fibronectin density F of the extracellular matrix, the oxygen concentration O and the Dynamic Energy Budget reserve density E of the endothelial population, all functions of $x \in \Omega = [0, 1]$ and $t > 0$. Writing $f_O = O/(K_O + O)$ for the oxygen functional response, the system is

$$\partial_t C = D_C \partial_x^2 C - \partial_x (\chi C \partial_x F) + \frac{1}{E_G} (\kappa k_M E - \zeta C)_+ - d_C C, \quad (1)$$

$$\partial_t P = D_P \partial_x^2 P + a_P (1 - \kappa) k_M E - (d_P + \lambda I) P, \quad (2)$$

$$\partial_t I = D_I \partial_x^2 I + a_I C - d_I I, \quad (3)$$

$$\partial_t F = b_F C - (b_F C + \mu_F P) F, \quad (4)$$

$$\partial_t O = D_O \partial_x^2 O + S_O - \left(S_O + g_O \rho_A \frac{C}{K_O + O} \right) O, \quad (5)$$

$$\partial_t E = \rho_A f_O C - k_M E, \quad (6)$$

with homogeneous Neumann conditions for the diffusing species and $(\cdot)_+ = \max(\cdot, 0)$. The energetic content of the model is in (6) and in the first reaction term of (1). Oxygen is assimilated into the reserve at the rate $\rho_A f_O C$, which saturates in O , the reserve is mobilized at the rate $k_M E$ set by the reserve itself and not by the environment, and the mobilized flux is split by the κ -rule of Dynamic Energy Budget theory [51], a fraction κ paying maintenance ζC first and funding proliferation with what remains, the complementary fraction funding the protease production in (2). The five other fields carry the classical angiogenesis structure of [48,50], with haptotactic migration of C up gradients of F , proteolytic degradation of F opposed by an inhibitor, and oxygen supplied toward a reference level and consumed in proportion to assimilation. Table 1 lists the parameters.

Each field and each interaction in (1) to (6) stands for an observed step of a sprouting capillary. Tumour-induced angiogenesis begins when a hypoxic tumour releases diffusible factors, endothelial cells of a nearby venule are activated, and a sprout migrates through the stroma toward the source, which is the process Folkman identified as the target of an antiangiogenic therapy [55–57]. The migration of the tip is directed by adhesion to the extracellular matrix rather than only by a soluble gradient, and endothelial cells move up gradients of immobilized fibronectin in vitro [58], which is the haptotactic flux $\chi C \partial_x F$ of (1); the proliferation that follows in the stalk was already separated from the migration of the tip in the classical observations [59,60]. The proteolysis of (2) and (4) is the secretion of matrix metalloproteinases that degrade fibronectin and open a path, and the inhibitor I of (3) stands for their tissue inhibitors, a balance whose displacement is what switches the angiogenic phenotype on [61]. What the Dynamic Energy Budget reserve adds to that classical picture is the constraint that migration and proliferation are paid for. Endothelial cells are strongly glycolytic and their sprouting is limited by the flux through glycolysis

rather than by the availability of a growth factor alone, so that suppressing that flux blocks vessel sprouting without touching the signal [62]. Equation (6) is the coarsest description of that constraint: assimilation saturates in the oxygen available, the reserve is mobilized at a rate set by itself and not by the environment, and the κ -rule splits the mobilized flux between maintenance and growth on one side and proteolysis on the other [51,52]. The consequence that matters for the numerics is that a cell can be surrounded by matrix and still not proliferate, because the reserve is empty, which is a behaviour a model without an energy state cannot produce. The parameters are not fitted to a particular assay, so the system is used here as a testbed with the right structure and not as a quantitative model of a specific experiment.

Table 1. Parameters of the six-field model.

Symbol	Meaning	Value
Fixed across the family		
D_C	endothelial diffusivity	0.0001
D_O	oxygen diffusivity	0.002
D_P	protease diffusivity	0.0002
D_I	inhibitor diffusivity	0.0002
k_M	reserve mobilization rate	1
ζ	maintenance rate	0.1
E_G	energy cost of one cell	1
g_O	oxygen use per assimilation	0.5
a_P	protease yield	1
d_P	protease decay	0.5
λ	inhibition rate	1
a_I	inhibitor production	0.2
d_I	inhibitor decay	0.3
b_F	fibronectin deposition	0.5
μ_F	fibronectin degradation	1
e_P	protease energy content	1
Sampled per trajectory		
κ	allocation fraction	[0.4, 0.8]
ρ_A	assimilation rate	[0.8, 1.6]
S_O	oxygen supply	[0.4, 0.8]
χ	haptotaxis coefficient	[0.001, 0.005]
K_O	oxygen half-saturation	[0.2, 0.4]
d_C	endothelial apoptosis	[0.01, 0.04]

The property the numerics must respect is visible without solving anything. Collecting the six unknowns in $\mathbf{U} = (C, P, I, F, O, E)$, every reaction term in (1) to (6) is either a nonnegative production or a per capita loss, so the reaction part has the production and destruction form

$$\frac{d\mathbf{U}}{dt} = \mathbf{g}(\mathbf{U}) - \mathbf{d}(\mathbf{U}) \odot \mathbf{U}, \quad \mathbf{g}(\mathbf{U}) \geq 0, \quad \mathbf{d}(\mathbf{U}) \geq 0, \quad (7)$$

where $\odot$ is the entrywise product and the inequalities hold entrywise for every $\mathbf{U} \in \mathbb{R}_{\geq 0}^6$. The transport part is diffusive and advective, and the diffusion operators are Metzler, that is, their off-diagonal entries are nonnegative once discretized. The nonnegative orthant is therefore invariant for the continuous problem, and a discretization deserves the model only if the same holds for the discrete solution.

2.2. The Reference Integrator and Its Sign Structure

We discretize Ω with M cells of size $h = 1/M$ and cell centres $x_i = (i + \frac{1}{2})h$, and we collect the discrete state in $\mathbf{U}^n \in \mathbb{R}^{6 \times M}$. The reference integrator is a Strang palindrome [63] that alternates transport and reaction,

$$\mathbf{U}^{n+1} = \mathcal{T}_{\Delta t/2} \mathcal{R}_{\Delta t} \mathcal{T}_{\Delta t/2} \mathbf{U}^n, \quad (8)$$

and the three blocks are chosen so that each one preserves the sign of the state by itself.

The diffusion of a species with diffusivity D over a step τ is advanced by the exact exponential of the discrete Neumann Laplacian L . Because L is symmetric and Metzler and annihilates constants, the propagator

$$\mathcal{E}(D, \tau) = \exp(\tau DL) \quad (9)$$

is entrywise nonnegative, symmetric, and row-stochastic. It is diagonalized by the discrete cosine transform of the second kind, so that $\mathcal{E}(D, \tau) = \Phi^\top \text{diag}(e^{\tau D \mu_k}) \Phi$ with Φ the orthonormal transform matrix and $\mu_k = -2h^{-2}(1 - \cos(\pi k/M))$ the eigenvalues. Two implementations follow from this identity and they are not equivalent here. Applying the transform, scaling and transforming back costs $\mathcal{O}(M \log M)$ but computes the result as a signed sum of oscillatory contributions, so cancellation can return a negative value where the exact result is small and positive. Assembling the dense matrix once and applying it costs $\mathcal{O}(M^2)$ per species and per step, and it computes the result as a nonnegative combination of nonnegative numbers. For the grids at which a surrogate of this kind is deployed the second implementation is a single dense product per species, which is the kernel accelerators execute at peak rate, and it is the one we adopt. Section 3.2 shows that the choice is what makes the guarantee survive quantization.

Remark 1. *The two implementations differ in cost as well as in sign behaviour, and the comparison is not settled by the exponents alone. The transform route costs $\mathcal{O}(M \log M)$ per species against $\mathcal{O}(M^2)$ for the dense product, so it wins asymptotically, but the dense product is a single general matrix multiplication, which is the operation for which accelerators and inference engines are optimized, while the transform route is a sequence of smaller and less regular kernels. At the grids at which a surrogate of this kind is deployed the dense route is competitive in time and superior in sign behaviour, and it is the one used here. At much finer grids the dense assembly becomes impractical, and the way to keep both properties is the tensor-product factorization of the propagator in higher dimension, where each factor is again a dense nonnegative matrix acting on one direction. Section 5.3 measures the three routes. The assembled product is the fastest of them up to $M = 4096$ and is overtaken by the transform at $M = 8192$, where it also needs 256 megabytes of storage per species; the transform returns negative entries at every grid tested, of the order of 10^{-7} relative to a state of order one; and the factorized route matches the sign behaviour of the assembled one at a storage that grows like dM^2 instead of M^{2d} .*

The haptotaxis of C up gradients of F is advanced by a conservative first-order upwind flux. Denoting by $v_{i+1/2} = \chi(F_{i+1} - F_i)/h$ the interface velocity, with $v_{1/2} = v_{M+1/2} = 0$ by the boundary condition, and splitting it into its positive and negative parts $v^\pm$, the flux difference can be rearranged into

$$C_i \longleftarrow \left(1 - \lambda_\tau(v_{i+1/2}^+ + v_{i-1/2}^-)\right) C_i + \lambda_\tau v_{i+1/2}^- C_{i+1} + \lambda_\tau v_{i-1/2}^+ C_{i-1}, \quad \lambda_\tau = \frac{\tau}{h}, \quad (10)$$

which is the same update as the flux form but written so that the state is combined with three coefficients that are nonnegative whenever the advective Courant number satisfies

$$\lambda_{\tau}(v_{i+1/2}^+ + v_{i-1/2}^-) \leq 1 \quad \text{for every } i. \quad (11)$$

The two forms agree to round-off, which we verify in Section 4.1, and only the second one is free of subtraction acting on the state.

The reaction is advanced by the second-order Modified–Patankar predictor and corrector of Kopecz and Meister [10] applied to (7). Writing $\mathbf{g}^0 = \mathbf{g}(\mathbf{U})$ and $\mathbf{d}^0 = \mathbf{d}(\mathbf{U})$, the predictor is

$$\mathbf{U}^* = \frac{\mathbf{U} + \tau \mathbf{g}^0}{1 + \tau \mathbf{d}^0}, \quad (12)$$

with the quotient taken entrywise, and the corrector averages the productions and the destructions evaluated at $\mathbf{U}$ and at $\mathbf{U}^*$ in the same quotient form. Both stages are quotients of a nonnegative numerator by a denominator that exceeds one, so they are unconditionally positive and no step size restriction arises from the reaction. The price of unconditional positivity is that the update is not linear in the data, which is unavoidable, since a scheme that is linear in the data and unconditionally positive cannot exceed first order [64].

3. The Macro-Step Surrogate

3.1. Architecture

Let K be the number of reference steps that one surrogate evaluation replaces and let $\tau = K\Delta t$ be the macro step. The surrogate is the map

$$\mathbf{U}^{n+1} = \mathcal{S}_{\theta}(\mathbf{U}^n, \vartheta) = \mathcal{T}^{\tau/2} \mathcal{M}_{\theta}^{\tau} \mathcal{T}^{\tau/2} \mathbf{U}^n, \quad (13)$$

where ϑ collects the physical parameters that vary across the family of problems, the transport block $\mathcal{T}^{\tau/2}$ applies the dense propagators (9) of the four diffusing species for the half macro step followed by the nonnegative form (10) of the haptotaxis, and the learned block is the Modified–Patankar quotient

$$\mathcal{M}_{\theta}^{\tau}(\mathbf{V}) = \frac{\mathbf{V} + \tau \mathbf{g}_{\theta}(\mathbf{V}, \vartheta)}{1 + \tau \mathbf{d}_{\theta}(\mathbf{V}, \vartheta)}, \quad \begin{bmatrix} \mathbf{g}_{\theta} \\ \mathbf{d}_{\theta} \end{bmatrix} = \varsigma(\mathcal{N}_{\theta}(\mathbf{V}, \vartheta)), \quad (14)$$

with $\mathcal{N}_{\theta}$ a convolutional network with twelve output channels and ς the softplus function applied channelwise. The role of ς is not smoothness but sign. It maps the network output onto the nonnegative reals, so that $\mathbf{g}_{\theta}$ is a production and $\mathbf{d}_{\theta}$ is a per capita destruction whatever the weights are, and (14) is then the discrete analogue of (7) with learned coefficients.

The transport block of the surrogate differs from the one of the reference integrator in a detail that matters for the guarantee. Over a macro half step the advective Courant number of (10) is larger than it is over a reference step by the ratio of the two step sizes, and an input that the training never produced can raise it further, since a perturbation of the matrix field at the scale of the mesh produces gradients of order h^{-1} . Two devices remove the dependence on the data. The haptotaxis step is applied in n substeps with the interface velocities frozen, which divides the Courant number by n , and the diagonal coefficient of (10) is clipped at zero, which is the analogue for a single coefficient of the limiters that enforce a maximum principle on a high-order flux [65]. Together they make the coefficients of the transport block nonnegative for every input, so the hypothesis of the results below is a property of the implementation rather than an assumption on the data.

The number of substeps follows from a criterion rather than from a trial. With the half macro step divided into n substeps the diagonal coefficient of (10) is $1 - \lambda_n(v_{i+1/2}^+ + v_{i-1/2}^-)$ with $\lambda_n = \tau/(2nh)$, and the interface velocities satisfy $|v_{i+1/2}| \leq \chi\Delta/h$ with

$$\Delta = \max_i |F_{i+1} - F_i| \quad (15)$$

the largest jump of the matrix field between adjacent cells, so the two velocities that enter one coefficient sum to at most $2\chi\Delta/h$ and the coefficient is nonnegative for every cell as soon as

$$n \geq \frac{\tau \chi \Delta}{h^2}. \quad (16)$$

The quantity Δ admits an a priori bound. Written as $\partial_t F = b_F C(1 - F) - \mu_F P F$, Equation (4) has the interval $[0, 1]$ invariant, since the right-hand side is nonnegative at $F = 0$ and equal to $-\mu_F P \leq 0$ at $F = 1$, so $\Delta \leq 1$ and

$$n \geq \left\lceil \frac{\tau \chi_{\max}}{h^2} \right\rceil = 17 \quad (17)$$

makes the coefficient nonnegative for every state, every parameter of the box and every input whatsoever, with no clip. The value used throughout is $n = 4$, which (16) justifies for every $\Delta \leq 2.44 \times 10^{-1}$ in the sense of (15), and which covers every regime examined in Section 4, and the clip covers the difference, since it changes the update only where the coefficient would have been negative and therefore makes the block nonnegative at any n .

Neither device is free, and the operation count is misleading about which one costs. The substeps are $\mathcal{O}(M)$ operations against the $\mathcal{O}(M^2)$ of the dense propagators, below one part in a thousand of the arithmetic of a macro step at $n = 4$ and three parts in a thousand at $n = 17$, but each substep is a short elementwise kernel and at this problem size the evaluation is bound by the launches rather than by the arithmetic, so raising n from four to seventeen multiplies the time of one macro step by 2.4, from 1.25 to 3.03 ms. The accuracy is unaffected, at 5.189×10^{-3} in both cases. The clip costs the exact conservation of the mass of C on the cells where it acts and nothing anywhere else.

Three design decisions deserve a word. The transport block carries no parameters, because the exponential of the Metzler operator is exact for any step size and there is nothing in it to learn, so the network spends its capacity on the reaction, the allocation and the correction of the splitting error over the macro step. The convolutional stack is local, and a receptive field of a few cells suffices because the distance that diffusion and taxis cover in one macro step is a small multiple of the mesh size in the regime of interest, which is what keeps the learned block small. The physical parameters enter as constant channels after normalization to the unit interval, so that one trained map serves the whole parameter box rather than a single trajectory.

Algorithm 1 states one macro step in the form in which it is implemented. Every line is either a nonnegative matrix acting on a nonnegative vector, a nonnegative combination of neighbouring entries, or a quotient with a numerator that is nonnegative and a denominator that is at least one, which is the property that Section 3.2 turns into a statement about arithmetic.

The comparison used throughout is deliberately narrow. The unstructured surrogate replaces (14) by the explicit update

$$\mathcal{M}_\theta^\tau(\mathbf{V}) = \mathbf{V} + \tau(\mathcal{N}_\theta^{(1)}(\mathbf{V}, \vartheta) - \mathcal{N}_\theta^{(2)}(\mathbf{V}, \vartheta)), \quad (18)$$

where $\mathcal{N}^{(1)}$ and $\mathcal{N}^{(2)}$ are the same twelve output channels split in two groups. The backbone, the number of parameters, the training data, the optimizer and the schedule are

identical, so that the only difference between the two models is whether the two channel groups are combined by a quotient or by a subtraction. The penalized surrogate adds to the loss of the unstructured model a penalty on the negative part of the prediction and a penalty on the deviation of the discrete energy budget, which is the soft-constraint counterpart of the architectural constraint. The clamped surrogate is the unstructured model composed at inference time with the projection onto the nonnegative orthant, without retraining, which is the shortcut a practitioner reaches for first. The black-box surrogate removes the transport blocks altogether and maps the state forward with the network alone.

Algorithm 1 One macro step of the structured surrogate.

Require: state $\mathbf{U} \geq 0$, parameters ϑ , propagators $\mathcal{E}_s$ for the half macro step

Ensure: state after the macro step, nonnegative by construction

```

1: for each transport half step do
2:   for each diffusing species  $s$  do
3:      $\mathbf{U}_s \leftarrow \mathcal{E}_s \mathbf{U}_s$  ▷ dense nonnegative product, one matrix multiplication
4:   end for
5:   compute the interface velocities from  $\mathbf{U}_F$  and split them into  $v^+$  and  $v^-$ 
6:    $a \leftarrow \max\{0, 1 - \lambda(v_{\rightarrow}^+ + v_{\leftarrow}^-)\}$  ▷ clipped diagonal,  $\lambda = \tau/(2nh)$ 
7:   for  $n$  substeps do
8:      $\mathbf{U}_C \leftarrow a \mathbf{U}_C + \lambda(v^- \cdot \mathbf{U}_C^{\rightarrow} + v^+ \cdot \mathbf{U}_C^{\leftarrow})$  ▷ nonnegative coefficients, no condition
9:   end for
10:  if first half step then
11:     $(\mathbf{g}, \mathbf{d}) \leftarrow \zeta(\mathcal{N}_{\vartheta}(\mathbf{U}, \vartheta))$  ▷ softplus, so  $\mathbf{g} \geq 0$  and  $\mathbf{d} \geq 0$ 
12:     $\mathbf{U} \leftarrow (\mathbf{U} + \tau \mathbf{g}) / (1 + \tau \mathbf{d})$  ▷ Modified–Patankar quotient
13:  end if
14: end for

```

3.2. Guarantees in Exact and in Finite-Precision Arithmetic

We first record the sign structure of the assembled operators, then the invariance of the nonnegative orthant, then the statement that carries the deployment claim.

Proposition 1 (Sign structure of the transport block). *Let L be the discrete Neumann Laplacian on M cells and let $D > 0$ and $\tau > 0$. Then $\mathcal{E}(D, \tau) = \exp(\tau DL)$ is entrywise nonnegative, symmetric and row-stochastic, and for a state $\mathbf{u} \geq 0$ the product $\mathcal{E}(D, \tau)\mathbf{u}$ is nonnegative and has the same discrete mass as $\mathbf{u}$. The haptotaxis update (10) with the diagonal coefficient clipped at zero is a nonnegative combination of the entries of a nonnegative state, for every velocity field and every step, and it coincides with the unclipped update whenever (11) holds.*

Proof. The matrix τDL is Metzler, since the off-diagonal entries of L are $h^{-2} \geq 0$, so $\exp(\tau DL) = e^{-s} \exp(sI + \tau DL)$ with s large enough that $sI + \tau DL$ is entrywise nonnegative, and the exponential of an entrywise nonnegative matrix is entrywise nonnegative, whence the first claim [53]. Symmetry is inherited from L . The Neumann Laplacian annihilates constants, so $L\mathbf{1} = 0$ and therefore $\exp(\tau DL)\mathbf{1} = \mathbf{1}$, which is row-stochasticity. Nonnegativity of the product is immediate, and mass is preserved because $\mathbf{1}^\top \mathcal{E} = (\mathcal{E}^\top \mathbf{1})^\top = (\mathcal{E}\mathbf{1})^\top = \mathbf{1}^\top$ by symmetry and row-stochasticity. For the last claim, the two off-diagonal coefficients in (10) are nonnegative because λ_{τ}, v^+ and v^- are, and the diagonal coefficient is nonnegative after the clip by definition, while under (11) the clip is inactive and the two updates agree. $\square$

Proposition 2 (Invariance of the nonnegative orthant). *Let $\mathbf{U}^n \geq 0$ entrywise. Then the surrogate (13) satisfies $\mathbf{U}^{n+1} \geq 0$ entrywise, for every weight vector θ , every macro step $\tau > 0$, every admissible parameter ϑ and every state, with no condition on the Courant number.*

Proof. The transport block maps the nonnegative orthant into itself by Proposition 1. In the learned block the numerator of (14) is the sum of a nonnegative state and of $\tau \mathbf{g}_\theta \geq 0$, since the softplus function takes values in the positive reals, and the denominator is $1 + \tau \mathbf{d}_\theta \geq 1$ for the same reason. The quotient of a nonnegative number by a number not smaller than one is nonnegative. Composing the three blocks gives the claim, and nothing in the argument refers to the values of the weights or to the size of τ . $\square$

The unstructured update (18) admits no such statement, and the reason is not that the network is badly trained. For any state $\mathbf{V} \geq 0$ with a component V_j smaller than $\tau \mathcal{N}_j^{(2)}$, the corresponding component of the update is negative. Since $\mathcal{N}^{(2)}$ is an unconstrained network output and V_j is arbitrarily small at the leading edge of a front or in a depleted region, the set of states on which the update leaves the orthant is not exceptional, and whether the trained model visits it is a question about the trajectory rather than about the architecture.

We now turn to the arithmetic. Let fl denote the map that a machine applies to the exact result of an elementary operation, extended to the elementary operations of the forward pass.

Definition 1 (Sign-faithful arithmetic). *An arithmetic is sign-faithful if the result of every implemented operation on nonnegative operands is nonnegative, that is, if $\text{fl}(a \circ b) \geq 0$ whenever $a, b \geq 0$ and $a \circ b \geq 0$ for $\circ \in \{+, \times, /\}$, and if the quantizers applied to stored tensors and to intermediate tensors map nonnegative entries to nonnegative entries.*

Every IEEE binary format in every rounding mode is sign-faithful, because the exact result of a sum, a product or a quotient of nonnegative numbers is nonnegative and rounding a nonnegative real returns a nonnegative representable number in each of the four directed and nearest modes, the value zero included [66]. A symmetric fixed-point quantizer with round-to-nearest is sign-faithful for the same reason, since it computes $s \cdot \text{round}(x/s)$ with $s > 0$ and the rounding of a nonnegative real is a nonnegative integer. Saturating and stochastic rounding are sign-faithful as well, provided the clipping range is symmetric or nonnegative.

Sign-faithfulness is a property of the arithmetic alone and is not by itself enough. The forward pass has to be implemented so that the arithmetic sees the sign structure, and the three requirements below are consequences of the construction of Section 3.1 rather than extra assumptions, but they are requirements on the code and they can be violated by an implementation that computes the same operator differently. They are therefore stated as hypotheses of the theorem, and Section 5.2 turns them into three checks that can be run on the deployed artefact.

The denominator of (14) deserves a separate word, because the inequality $1 + \tau \mathbf{d}_\theta \geq 1$ is used in exact and in computed arithmetic and the two are not the same statement. In exact arithmetic it holds because $\mathbf{d}_\theta \geq 0$. In a floating-point format it holds because the value one is exactly representable in every binary format, because rounding is monotone, and because the exact sum $1 + \tau d$ with $\tau d \geq 0$ is at least one, so its rounding is at least one as well and no rounding mode can bring the computed denominator below one. What the argument does need is that the operands stay in the range of the format, which is not automatic, since $\tau \mathbf{d}_\theta$ is a network output scaled by the macro step and could in principle overflow a format with a short exponent. It does not here, and the margin is wide. Over the states of the test trajectories the largest value of $\tau \mathbf{d}_\theta$ is 2.29×10^{-1} and the largest numerator is 1.21, against the largest finite value 6.55×10^4 of half precision, so more than four decades separate the computation from the range of the narrowest floating-point format used here; the computed denominator was below one in none of the 2.9×10^6 evaluations recorded,

and no evaluation reported in this paper, in any format and in any regime, produced a non-finite value. Were the margin exhausted the failure would be graceful in one direction and not in the other, since an overflowing denominator sends the quotient to zero and an overflowing numerator sends it to $+\infty$, both of which are nonnegative, while an overflow of both at once produces a value that is not a number. That is why the condition appears as a hypothesis rather than being assumed away.

Theorem 1 (Positivity under sign-faithful arithmetic). *Let the forward pass of the structured surrogate be implemented so that*

- (H1) *the diffusion step is a product by the assembled propagator, stored as an entrywise nonnegative matrix;*
- (H2) *the haptotaxis step is evaluated in the form (10), with the diagonal coefficient computed as a maximum against zero, so that its three coefficients are nonnegative;*
- (H3) *the learned block is evaluated as in (14), with an output nonlinearity that returns a nonnegative value in the working arithmetic;*
- (H4) *no intermediate quantity of the forward pass leaves the range of the format.*

Then, in any arithmetic that is sign-faithful in the sense of Definition 1, the computed state satisfies $\hat{\mathbf{U}}^n \geq 0$ for every n , for every weight vector, every macro step, every admissible parameter and every quantizer that the definition admits.

Proof. By induction on n , the base case being the initial state, and the argument is the one of Proposition 2 carried through the implemented operations rather than the exact ones. Assume $\hat{\mathbf{U}}^n \geq 0$. By (H1) the propagator is stored as a nonnegative matrix, and a quantizer that is sign-faithful returns a nonnegative matrix, so the computed product is a sum of products of nonnegative numbers and every partial sum is nonnegative, hence the computed result is nonnegative in any accumulation order. By (H2) the haptotaxis update is a sum of products of nonnegative coefficients with nonnegative states, the diagonal one because the clip is evaluated as a maximum against zero and a sign-faithful arithmetic returns a nonnegative result for it, and the same argument applies. By (H3) the computed numerator of the learned block is the sum of a nonnegative state and of a nonnegative multiple of a computed softplus, which is nonnegative because the softplus is evaluated as a nonnegative function and its quantization is sign-faithful, so the computed numerator is nonnegative. The computed denominator is one plus a nonnegative quantity, hence positive, and no rounding can bring it below one since the exact value is at least one and rounding is monotone. The computed quotient is therefore nonnegative, and it is a number rather than an overflow by (H4). Composing the blocks preserves the property, which closes the induction. $\square$

Three consequences are worth stating separately. First, no clamping is needed. A projection onto the nonnegative orthant appended to the structured surrogate would never be active, so the model never pays the accuracy and the budget error that a projection introduces on a model that does need it, which is the clamped baseline of Section 4. Second, the guarantee is uniform in the bit width, so it holds for eight, six or four bits alike, and what the bit width controls is the accuracy of the trajectory and not its admissibility. Third, and this is the point of the construction, nothing in the argument depends on the trained weights, so a model that is quantized after training carries the same guarantee as the model that was validated in single precision.

The guarantee is about the sign and not about the balance laws, and quantizing the transport matrix does perturb the discrete mass of the diffusing species. That perturbation is bounded a priori, and the bound involves the quantization step alone.

The distinction is worth stating in full, because positivity and conservation are independent properties and only the first is proved here. A map can keep every field nonnegative and still create or destroy mass, and a map can conserve mass exactly and still return a negative density. The three blocks of the surrogate sit differently with respect to the two. The diffusion propagator has both, since it is nonnegative and row-stochastic, so it conserves the discrete mass of each diffusing species exactly, to 9.84×10^{-16} in the measurements of Section 5.2. The haptotaxis step has both as long as the clip is inactive, since the upwind flux form is conservative by construction, and loses mass exactly on the cells where the clip acts, which Section 4.1 quantifies. The learned block has the first and not the second: nothing constrains a learned production of one field to be a learned destruction of another, so no linear invariant of the reaction is exact. Quantization then acts on the two properties in opposite ways. It cannot remove the first, which is Theorem 1, and it degrades the second in a way that Proposition 3 bounds for the transport block and that no theorem bounds for the learned block. The practical consequence is the budget column measured in Section 4.1, where eight-bit quantization of every tensor multiplies the error on the energy budget of the structured surrogate by 51, from 1.58×10^{-3} to 8.09×10^{-2} , while leaving the sign untouched, and where keeping the transport block in single precision brings it back to 4.82×10^{-3} . A deployment that needs a balance law and not only an admissible state has to read that column and not Theorem 1.

Proposition 3 (Mass drift of the quantized transport block). *Let $\mathcal{E}$ be the exact propagator of Proposition 1, let $\hat{\mathcal{E}}$ be its symmetric fixed-point quantization with step s and round-to-nearest, and set*

$$\delta = \max_{1 \leq j \leq M} \left| \sum_{i=1}^M (\hat{\mathcal{E}}_{ij} - \mathcal{E}_{ij}) \right|. \quad (19)$$

Then, for every $\mathbf{u} \geq 0$,

$$\left| \mathbf{1}^\top \hat{\mathcal{E}} \mathbf{u} - \mathbf{1}^\top \mathbf{u} \right| \leq \delta \|\mathbf{u}\|_1 \leq \frac{Ms}{2} \|\mathbf{u}\|_1. \quad (20)$$

Applying the estimate to each of the two half steps of every macro step, and considering the transport block alone, the mass after N macro steps differs from the initial one by at most the factor $(1 + \delta)^{2N}$ from above and $(1 - \delta)^{2N}$ from below, so the relative drift over the horizon is at most $(1 + \delta)^{2N} - 1$, which is $2N\delta$ to first order in δ and exceeds it when δ is not small. The quantity δ is computed once, offline, from the stored matrices, so the first inequality is a bound the deployment can evaluate, while the second is the worst case attained only when every entry of a column rounds in the same direction. The statement is about the exact propagator, and the matrix actually stored is the assembled one, whose repair described in Section 2.2 makes the row sums exactly one and leaves the symmetry correct to the round-off of the assembly. That discrepancy adds a term of the same order to the drift, which the verification of Section 4.1 measures and which is negligible against the quantization term whenever the format is coarser than double precision.

Proof. Since $\mathcal{E}$ is row-stochastic and symmetric, $\mathbf{1}^\top \mathcal{E} = (\mathcal{E} \mathbf{1})^\top = \mathbf{1}^\top$, so $\mathbf{1}^\top \hat{\mathcal{E}} \mathbf{u} - \mathbf{1}^\top \mathbf{u} = \mathbf{1}^\top (\hat{\mathcal{E}} - \mathcal{E}) \mathbf{u}$, whose absolute value is at most $\max_j \left| \sum_i (\hat{\mathcal{E}}_{ij} - \mathcal{E}_{ij}) \right| \|\mathbf{u}\|_1 = \delta \|\mathbf{u}\|_1$ because $\mathbf{u} \geq 0$. Each entry satisfies $|\hat{\mathcal{E}}_{ij} - \mathcal{E}_{ij}| \leq s/2$ by round-to-nearest, so $\delta \leq Ms/2$. For the statement over the horizon, a nonnegative state keeps its one norm equal to its mass, so one half step multiplies the mass by a factor in $[1 - \delta, 1 + \delta]$, and $2N$ half steps compound those factors. $\square$

The scope of the statement is narrower than the phenomenon it describes, and the difference matters when the bound is used on a deployed model. It covers the four diffusing species and the transport block alone, it assumes a symmetric fixed-point quantizer with

round-to-nearest, and it assumes that the stored matrix is row-stochastic before quantization, which the repair of Section 2.2 achieves to 3.33×10^{-16} . It does not cover the haptotaxis step when the clip is active, the two fields that do not diffuse, or the learned block, whose contribution to the drift is the one no theorem controls. It also does not transfer verbatim to a floating-point format, where the perturbation of an entry is relative rather than absolute and the second inequality of (20) has no analogue; the first inequality does transfer, since δ is defined from the stored matrix and can be computed in any format, and it is the form a deployment should use. Finally, the compounded factor $(1 + \delta)^{2N}$ is sharp only if every half step drifts in the same direction, which is what makes it a bound and not an estimate: at eight bits it exceeds the realized drift by a factor of 15 and at four bits by a factor of 1.1, so it is informative exactly where the drift is large.

Proposition 3 is what suggests the arrangement we recommend in Section 5. The transport block is four dense products and carries no parameters, so keeping it in single precision costs little memory and removes the drift term entirely, while the learned block, which is where the parameters and most of the arithmetic intensity are, is the part worth quantizing. Theorem 1 holds in both arrangements, and Section 4 measures what each one costs in accuracy.

Remark 2. *The contrast with the penalized model is sharper than the loss functions suggest. A penalty on the negative part of the prediction is evaluated on the training distribution and in the training format, so it constrains the weights to keep the trajectories it saw inside the orthant. It says nothing about a trajectory that starts outside the training box, about a horizon longer than the one used for training, and above all nothing about the same weights rounded to a coarser grid, since the rounding moves the very quantity the penalty controlled. Section 4.2 reports what this amounts to in practice.*

The guarantee says nothing about accuracy, and it is useful to record where the error of a rollout comes from, since the three sources respond to different remedies.

Proposition 4 (Error decomposition of a rollout). *Let $\mathcal{S}^\tau$ be the exact solution operator of the semidiscrete system over the macro step, let $\mathcal{S}_\theta$ be the surrogate in exact arithmetic and let $\widehat{\mathcal{S}}_\theta$ be its evaluation in the working arithmetic. Assume that on a set containing the trajectories of interest the one-step approximation error satisfies $\|\mathcal{S}_\theta(\mathbf{U}) - \mathcal{S}^\tau(\mathbf{U})\| \leq \varepsilon$, that the arithmetic contributes $\|\widehat{\mathcal{S}}_\theta(\mathbf{U}) - \mathcal{S}_\theta(\mathbf{U})\| \leq \eta$, and that the j -fold composition $\widehat{\mathcal{S}}_\theta^j$ is Lipschitz on that set with constant Λ_j , with $\Lambda_0 = 1$. Then, after N macro steps started from the same initial state,*

$$\|\widehat{\mathbf{U}}^N - \mathcal{S}^{\tau N} \mathbf{U}^0\| \leq (\varepsilon + \eta) \sum_{j=0}^{N-1} \Lambda_j \leq (\varepsilon + \eta) \sum_{k=0}^{N-1} \Lambda_1^k, \quad (21)$$

the second inequality being the specialization $\Lambda_j \leq \Lambda_1^j$.

Proof. Write $x_k = (\mathcal{S}^\tau)^k \mathbf{U}^0$ for the exact iterates and set

$$a_k = \widehat{\mathcal{S}}_\theta^{N-k}(x_k), \quad k = 0, \dots, N,$$

so that $a_0 = \widehat{\mathcal{S}}_\theta^N \mathbf{U}^0 = \widehat{\mathbf{U}}^N$ and $a_N = x_N$. Each a_k follows the exact flow for k steps and the computed map for the remaining $N - k$, so two consecutive terms differ by exactly one replacement,

$$a_k - a_{k+1} = \widehat{\mathcal{S}}_\theta^{N-1-k}(\widehat{\mathcal{S}}_\theta(x_k)) - \widehat{\mathcal{S}}_\theta^{N-1-k}(\mathcal{S}^\tau(x_k)),$$

where the computed map is composed with itself and is not applied to a difference, which matters because it is nonlinear. Applying the Lipschitz constant of the $(N - 1 - k)$ -fold

composition gives $\|a_k - a_{k+1}\| \leq \Lambda_{N-1-k} \|\widehat{\mathcal{S}}_\theta(x_k) - \mathcal{S}^\tau(x_k)\|$, and the one-step discrepancy at x_k is at most $\varepsilon + \eta$ by the triangle inequality, since x_k lies on the exact trajectory. Summing the telescoping differences,

$$\|\widehat{\mathbf{U}}^N - x_N\| = \left\| \sum_{k=0}^{N-1} (a_k - a_{k+1}) \right\| \leq (\varepsilon + \eta) \sum_{k=0}^{N-1} \Lambda_{N-1-k},$$

and reindexing gives (21). $\square$

The three terms are controlled by different means. The approximation term ε is reduced by training, by capacity and by a smaller macro step, and Section 5.1 measures how it responds to the width of the learned block. The arithmetic term η is proportional to the unit round-off of the format for the floating-point cases and to the quantization step for the integer ones, and Section 4.1 measures it as the gap between the rows of a single model. The amplification factors are a property of the dynamics and of the architecture and are not reduced by either. What Theorem 1 adds is orthogonal to all three, since it constrains the set in which the computed state lives however large the right-hand side of (21) becomes.

A decomposition whose constants are not estimated says little, so the three are measured at the states x_k of the exact trajectory, which is where the proposition evaluates them, and Table 2 collects the values. The approximation term is $\varepsilon = 1.34 \times 10^{-3}$ on average and 5.67×10^{-3} at worst. The arithmetic term is 9.83×10^{-4} for tensor float, 1.06×10^{-2} for half precision, 6.31×10^{-2} for bfloat16 and 1.02×10^{-1} for eight bits applied to every tensor, against 8.26×10^{-3} for eight bits applied to the learned block alone, so the ordering of the formats measured in Section 4.1 is already fixed by a single macro step.

The amplification admits an a priori bound and a measurement, and they agree. The transport block is almost nonexpansive: the propagator is symmetric and row-stochastic, so its spectral norm is one, and the four substeps of the taxis step have one-norm at most one and infinity-norm 1.036 over the states that occur, hence spectral norm at most 1.018. The learned block satisfies $\Lambda \leq 1 + \tau(1 + U)L_{\mathcal{N}}$ with $L_{\mathcal{N}}$ the product of the spectral norms of the four convolutions and of the Lipschitz constants of the activations, and U the largest value of the Patankar quotient, which gives $L_{\mathcal{N}} = 2.11 \times 10^3$, $U = 1.20$ and $\Lambda_1 \leq 964$. Power iteration on the Jacobian of one macro step returns 386 at worst over the states of the trajectory, so the bound computable from the weights alone overestimates the measured constant by a factor of 2.5. Two things follow. The constant is large partly because the norm is the field-normalized one in which the errors of this paper are reported: the same quantity in the plain Euclidean norm of the state is 43.4, and the factor of nine between the two is the three orders of magnitude that separate the fibronectin from the protease. And the one-step constant is not the constant a rollout obeys, since the Lipschitz constant of the whole ten-step map is 512, that is 1.87 per macro step, and a perturbation carried along the trajectory grows by 1.50 per macro step. The worst directions of successive steps do not compose, which is the reason the proposition is stated with the constants Λ_j and not with a single one.

The last block of Table 2 says what each version of the bound is worth in single precision. The geometric form built on the one-step constant gives 1.08×10^{21} and is vacuous. The form of (21) with the measured Λ_j gives 22.5. Replacing Λ_j by the amplification realized along the trajectory gives 6.34×10^{-1} , against a measured rollout error of 5.19×10^{-3} . The decomposition is therefore an upper bound of the right shape rather than a sharp estimate, and it is used in this paper to attribute the error to its three sources and not to certify a trajectory.

Table 2. Constants of the error decomposition, measured at the states of the exact trajectory.

Quantity	Mean	Largest
one-step approximation error ε over the states of the exact trajectory	1.34×10^{-3}	5.67×10^{-3}
arithmetic term η		
tf32	4.11×10^{-4}	9.83×10^{-4}
bf16	1.98×10^{-2}	6.31×10^{-2}
fp16	3.72×10^{-3}	1.06×10^{-2}
int8	4.06×10^{-2}	1.02×10^{-1}
int6	1.33×10^{-1}	2.78×10^{-1}
int4	4.06×10^{-1}	9.16×10^{-1}
int8 mixed	4.49×10^{-3}	8.26×10^{-3}
int6 mixed	1.70×10^{-2}	3.41×10^{-2}
int4 mixed	9.02×10^{-2}	1.52×10^{-1}
amplification Λ		
bound from the layer norms	–	964.02
one macro step, field-normalized norm	233.47	386.05
one macro step, Euclidean norm	26.25	43.38
the whole rollout, Λ_{10}	–	512.04
the same, per macro step	–	1.87
growth of a perturbation, per macro step	1.38	1.50
the bound of Proposition 4, single precision		
with Λ^k and the one-step constant	–	1,082,582,249,284,126,244,864.00
with the constants Λ_k of the composed maps	–	22.47
with the amplification realized along the trajectory	–	6.34×10^{-1}
measured	–	5.19×10^{-3}

4. Accuracy and Invariants

4.1. Setup and Behaviour Across Arithmetic Formats

The domain is discretized with 128 cells and the reference integrator uses the step 2.00×10^{-3} , so that one macro step of size 0.2 replaces 100 reference steps and a trajectory over the horizon 2 consists of 10 macro steps. The parameters of Table 1 that are marked as sampled are drawn uniformly from their box, one draw per trajectory, and the initial data are randomized in the position and the width of the initial vasculature, in the position and the width of the oxygen source and in the initial reserve. The training set has 512 trajectories, the validation set 96 and the test set 128, all produced in double precision. Training minimizes a field-normalized error accumulated over 3 unrolled macro steps, with the Adam optimizer, a cosine schedule, 300 epochs and batches of 64 windows.

The generation of the data is worth stating in full, since the reported accuracy is only as representative as the distribution behind it. The grid is 128 cells of size $h = 1/128$ with the state at the cell centres and homogeneous Neumann conditions, the reference step is $\Delta t = 2.00 \times 10^{-3}$, one macro step is $\tau = K\Delta t = 0.2$ with $K = 100$, and a trajectory is 10 macro steps to the final time 2, stored as eleven snapshots one macro step apart. Each trajectory carries its own draw of the six sampled parameters of Table 1, uniform in the intervals listed there and constant along the trajectory, and its own initial state, with the centre of the initial vasculature uniform in $[0.04, 0.09]$, its width in $[0.04, 0.07]$ and its amplitude in $[0.6, 1.0]$, the centre of the oxygen source uniform in $[0.90, 0.97]$ and its width in $[0.12, 0.20]$, the initial reserve uniform in $[0.05, 0.15]$ and the matrix field equal to one. The three splits are 512, 96 and 128 consecutive draws of a single random stream, so no parameter vector, no initial state and no trajectory is shared between them, and all three are

integrated in double precision by the reference scheme, the surrogates being trained and evaluated in single precision or below. Training uses every window of three consecutive macro steps inside a training trajectory, which is 8 windows per trajectory and 4096 in all, while validation and test are always the full ten-step rollout from the initial state, so the error reported in every table is a rollout error over the whole horizon and not the three-step error that is optimized. The field-normalized error divides each field by the mean of its absolute value over the training set, floored at 10^{-3} , and the checkpoint kept is the one with the best validation rollout error among those evaluated every ten epochs.

The penalized model adds two terms to that loss, a penalty on the mean squared negative part of the prediction in the same normalization as the data term, with weight $\lambda_+ = 10$, and a penalty on the mean squared deviation of the discrete energy budget from the reference, with weight $\lambda_B = 1$. The two were set so that at initialization neither term dominates the data term, and they are swept in Section 4.1 over three orders of magnitude, which is the measurement that decides whether the conclusion drawn from the penalized model is a conclusion about penalties or about one choice of weights.

Every model is trained with five seeds, and the two families of quantities are aggregated in the two ways that suit them. Accuracy and budget error are continuous quantities measured on five independent runs, so the tables of this section report the mean and the sample standard deviation, and the intervals quoted in the text are Student intervals at the ninety-five per cent level on four degrees of freedom. The invariant is not a continuous quantity: a run either leaves the nonnegative orthant or does not, so the same tables report the number of seeds in which a violation occurs, together with the worst value attained over the seeds. Averaging the second family would be meaningless, and reporting only the worst case of the first would be pessimistic, which is why the two columns are aggregated differently and the convention is stated in the header. The tables that examine one regime at a time report the first seed, and the spread over the seeds is quantified at the end of this section.

Table 3 collects the verification of the ingredients that the surrogate inherits, all performed before any training. The batched implementation of the reference scheme agrees with the independent implementation of the model to 1.52×10^{-11} in relative terms after two hundred steps, so the targets are the reference solution and not an approximation of it. The assembled propagator has a largest negative entry of 4.56×10^{-15} before the repair described in Section 2.2 and rows that sum to one within 3.33×10^{-16} after it, both at the level of the round-off of the assembly, and its asymmetry is 2.18×10^{-15} . Its exactness is confirmed in two independent ways, by comparing it with the thirty-two-fold composition of the propagator of one thirty-second of the step, which agrees to 2.55×10^{-13} , and by comparing it with two hundred thousand explicit substeps of the same operator, which agree to 3.93×10^{-7} , the residual being the error of the explicit reference rather than of the propagator. The reference integrator keeps every field nonnegative over the whole horizon, the smallest value attained being 8.91×10^{-9} . Finally the flux form and the nonnegative-coefficient form of the haptotaxis step agree to 1.11×10^{-16} when both are applied in one step, and the unclipped diagonal coefficient of the latter, computed on the macro half step in a single substep so as to isolate the effect of the step size, never falls below 9.37×10^{-1} over the training distribution, which corresponds to an advective Courant number of 6.27×10^{-2} .

The clip of Section 3.1 was promised a measurement and here it is. Recording the smallest unclipped diagonal coefficient over every rollout of the structured surrogate, with the four substeps in place, gives 9.77×10^{-1} on the test set, 9.49×10^{-1} on the doubled horizon, 9.67×10^{-1} outside the parameter box and 2.64×10^{-1} on perturbed initial data. The clip is therefore never active, in any regime, and the nonnegativity of the transport

block is delivered by the substeps alone. The last of the four numbers is what justifies them. It corresponds to an advective Courant number of 7.36×10^{-1} per substep, so on the whole macro half step the Courant number of that input reaches 2.94, and a transport block written in one step would have had a negative diagonal coefficient there and would have produced a negative density. Four substeps cost $\mathcal{O}(M)$ operations against the $\mathcal{O}(M^2)$ of the propagators and they move that number below one with a margin, which is why the construction of Section 3.1 is not a precaution without a case.

Table 3. Verification of the full-order model and of the assembled operators.

Quantity	Value
agreement with the reference implementation after 200 steps, relative	1.52×10^{-11}
largest negative propagator entry before repair	4.56×10^{-15}
largest row sum deviation before repair	7.55×10^{-15}
largest row sum error after repair	3.33×10^{-16}
largest asymmetry of the assembled propagator	2.18×10^{-15}
propagator against thirty-two-fold composition of itself	2.55×10^{-13}
propagator against 200,000 explicit substeps	3.93×10^{-7}
smallest value of any field over 500 full-order steps	8.91×10^{-9}
difference between the flux form and the nonnegative form of the taxis step	1.11×10^{-16}
smallest diagonal coefficient of the taxis step on the macro half-step	9.37×10^{-1}
largest advective Courant number on the macro half-step	6.27×10^{-2}

Table 4 tests the criterion (16) directly, by raising the amplitude of the multiplicative noise on the initial data until the clip is forced to act. At each amplitude the largest jump Δ of the matrix field between adjacent cells is measured on the full-order trajectory, which is where the criterion reads it, and (16) returns the number of substeps below which the clip must act: $n \geq 1$ at zero noise, $n \geq 4$ at ten per cent, $n \geq 9$ at twenty-five and $n \geq 17$ at fifty. The measured column of smallest diagonal coefficients crosses zero between $n = 2$ and $n = 4$, between $n = 4$ and $n = 8$, and between $n = 8$ and $n = 17$ respectively, so the criterion locates the transition to within one value of the ladder in every case, and the four substeps used throughout are exactly what it requires at the ten per cent amplitude of Section 4.2. Where the clip does act its cost is measured rather than assumed. At ten per cent noise a single substep clips 7.9 per cent of the cells and removes 23.5 per cent of the mass of the endothelial density over the horizon, two substeps clip 0.42 per cent and remove 3.2 per cent, and four substeps clip nothing and conserve the mass to 2.4×10^{-7} , which is the round-off of the sum. At an amplitude of one hundred per cent the criterion asks for thirty-three substeps and the ladder stops at seventeen, where 1.4 per cent of the cells are clipped and 5.7 per cent of the mass is lost. The smallest value of the state is zero in every row of the table, clipped or not, which is Theorem 1 once more: the clip changes the accuracy and the mass and never the sign.

Table 5 is the central measurement of the paper. Its four columns report the field-normalized error of the rollout over the whole horizon, the smallest value attained by any field at any cell and any time, the fraction of entries of the predicted trajectories that are negative, and the relative error on the discrete energy budget at the final time. The error and the budget error are averaged over the three seeds, while the two columns that concern the invariant report the worst case over them, since an invariant that fails in one seed has failed. Its rows group the five models by the arithmetic in which inference is performed, the three integer rows marked as mixed being the arrangement in which the learned block is quantized and the transport block stays in single precision.

Table 4. Substeps of the haptotaxis step against the criterion, and the cost of the clip when it acts.

Setting	Smallest Diagonal	Clipped Cells	Mass Defect of C	Smallest Value
noise 0 per cent, $\Delta = 1.29 \times 10^{-2}$, criterion $n \geq 1$				
$n = 1$	9.112×10^{-1}	0	2.20×10^{-7}	4.40×10^{-3}
$n = 2$	9.556×10^{-1}	0	2.22×10^{-7}	4.40×10^{-3}
$n = 4$	9.778×10^{-1}	0	2.14×10^{-7}	4.40×10^{-3}
$n = 8$	9.889×10^{-1}	0	2.35×10^{-7}	4.40×10^{-3}
$n = 17$	9.948×10^{-1}	0	4.72×10^{-7}	4.40×10^{-3}
noise 10 per cent, $\Delta = 1.98 \times 10^{-1}$, criterion $n \geq 4$				
$n = 1$	−1.864	7.95×10^{-2}	2.35×10^{-1}	0
$n = 2$	-4.318×10^{-1}	4.20×10^{-3}	3.16×10^{-2}	0
$n = 4$	2.841×10^{-1}	0	2.39×10^{-7}	3.14×10^{-4}
$n = 8$	6.421×10^{-1}	0	2.14×10^{-7}	7.78×10^{-4}
$n = 17$	8.316×10^{-1}	0	2.36×10^{-7}	1.04×10^{-3}
noise 25 per cent, $\Delta = 4.96 \times 10^{-1}$, criterion $n \geq 9$				
$n = 1$	−10.973	2.96×10^{-1}	1.16	0
$n = 2$	−2.579	1.28×10^{-1}	3.78×10^{-1}	0
$n = 4$	-7.897×10^{-1}	1.75×10^{-2}	6.51×10^{-2}	0
$n = 8$	1.051×10^{-1}	0	2.34×10^{-7}	4.33×10^{-8}
$n = 17$	5.789×10^{-1}	0	2.68×10^{-7}	8.09×10^{-6}
noise 50 per cent, $\Delta = 9.91 \times 10^{-1}$, criterion $n \geq 17$				
$n = 1$	−93,106,036,736.000	4.47×10^{-1}	12,414,215,168.00	0
$n = 2$	−6.282	2.96×10^{-1}	1.24	0
$n = 4$	−2.579	1.28×10^{-1}	3.87×10^{-1}	0
$n = 8$	-7.897×10^{-1}	1.77×10^{-2}	6.60×10^{-2}	0
$n = 17$	1.578×10^{-1}	0	2.85×10^{-7}	4.52×10^{-13}

The first observation concerns single precision, where the architectural constraint costs nothing. The structured surrogate reaches 5.07×10^{-3} against 5.52×10^{-3} for the unstructured model with the same 28956 parameters, 5.24×10^{-3} for the penalized model and 5.02×10^{-3} for the black-box model, so the four accuracies lie within ten per cent of one another. The ordering is not resolved by five training seeds either: the Student intervals at the ninety-five per cent level are $[4.80, 5.34] \times 10^{-3}$, $[5.14, 5.89] \times 10^{-3}$, $[4.60, 5.88] \times 10^{-3}$ and $[4.82, 5.23] \times 10^{-3}$ respectively, and every pair of them overlaps. The second observation is that in single precision none of the three models with a transport block violates positivity on the test set, the smallest value of the unstructured model being 0, which is precisely why the question cannot be settled by validating in the format used for training.

The third observation is the one the paper is about, and the right statistic for it is not an average over the seeds but the number of seeds in which a violation occurs at all, since an invariant either holds or does not. Under eight-bit quantization of every tensor the unstructured model leaves the nonnegative orthant in three of the five seeds, between -5.86×10^{-3} and -2.85×10^{-2} , and under six-bit quantization in all five, between -5.80×10^{-2} and -1.02×10^{-1} . The penalized model survives eight bits in all five seeds and leaves the orthant at six bits in all five, between -7.60×10^{-2} and -1.03×10^{-1} , and at four bits in four of them. The black-box model, which carries no transport block, leaves the orthant already in single precision, in two seeds of five, at -8.02×10^{-5} . The structured model attains a smallest value of exactly zero in every seed and every format of the table, including six and four bits, with no clamping applied at any point. That is the practical content of Theorem 1. The intermittency of the baselines across seeds is itself the argument. A property that appears in one seed and not in another is not a property of the method, and no amount of validation on the seed that happened to behave establishes it for the one that did not.

Table 5. Rollout accuracy and invariants of the surrogates across arithmetic formats.

Format	Error	Seeds with a Violation	Smallest Value	Budget Error
structured				
fp32	$5.07 \times 10^{-3} \pm 1.9 \times 10^{-4}$	0/5	0	1.58×10^{-3}
fp16	$1.84 \times 10^{-2} \pm 8.9 \times 10^{-4}$	0/5	0	2.27×10^{-3}
int8	$8.35 \times 10^{-2} \pm 2.6 \times 10^{-3}$	0/5	0	8.09×10^{-2}
int8 mixed	$1.34 \times 10^{-2} \pm 1.3 \times 10^{-3}$	0/5	0	4.82×10^{-3}
int4 mixed	$2.84 \times 10^{-1} \pm 4.1 \times 10^{-2}$	0/5	0	1.84×10^{-1}
unstructured				
fp32	$5.52 \times 10^{-3} \pm 2.7 \times 10^{-4}$	0/5	0	1.94×10^{-3}
fp16	$4.81 \times 10^{-2} \pm 2.8 \times 10^{-3}$	0/5	0	1.33×10^{-2}
int8	$2.00 \times 10^{-1} \pm 3.1 \times 10^{-2}$	3/5	-2.85×10^{-2}	2.47×10^{-1}
int8 mixed	$2.00 \times 10^{-1} \pm 3.4 \times 10^{-2}$	4/5	-1.87×10^{-2}	2.03×10^{-1}
int4 mixed	$2.12 \times 10^0 \pm 1.4 \times 10^{-1}$	5/5	-6.18×10^{-2}	1.28×10^0
penalized				
fp32	$5.24 \times 10^{-3} \pm 4.6 \times 10^{-4}$	0/5	0	1.80×10^{-3}
fp16	$3.65 \times 10^{-2} \pm 1.2 \times 10^{-2}$	0/5	0	1.56×10^{-2}
int8	$1.66 \times 10^{-1} \pm 1.2 \times 10^{-2}$	0/5	0	1.52×10^{-1}
int8 mixed	$1.76 \times 10^{-1} \pm 3.3 \times 10^{-2}$	3/5	-5.26×10^{-3}	1.07×10^{-1}
int4 mixed	$2.15 \times 10^0 \pm 2.9 \times 10^{-1}$	5/5	-7.15×10^{-2}	1.21×10^0
clamped				
fp32	$5.52 \times 10^{-3} \pm 2.7 \times 10^{-4}$	0/5	0	1.94×10^{-3}
fp16	$4.81 \times 10^{-2} \pm 2.8 \times 10^{-3}$	0/5	0	1.33×10^{-2}
int8	$1.99 \times 10^{-1} \pm 3.0 \times 10^{-2}$	0/5	0	2.38×10^{-1}
int8 mixed	$2.00 \times 10^{-1} \pm 3.4 \times 10^{-2}$	0/5	0	1.97×10^{-1}
int4 mixed	$2.12 \times 10^0 \pm 1.3 \times 10^{-1}$	0/5	0	1.30×10^0
black box				
fp32	$5.02 \times 10^{-3} \pm 1.5 \times 10^{-4}$	2/5	-8.02×10^{-5}	1.62×10^{-3}
fp16	$5.10 \times 10^{-3} \pm 1.4 \times 10^{-4}$	2/5	-7.93×10^{-5}	1.64×10^{-3}
int8	$3.98 \times 10^{-2} \pm 4.2 \times 10^{-3}$	1/5	-2.99×10^{-4}	2.76×10^{-2}
int8 mixed	$2.07 \times 10^{-2} \pm 5.5 \times 10^{-3}$	1/5	-2.99×10^{-4}	7.51×10^{-3}
int4 mixed	$4.40 \times 10^{-1} \pm 1.2 \times 10^{-1}$	5/5	-1.47×10^{-1}	1.68×10^{-1}

The penalty is instructive here. It is worth 5.24×10^{-3} in single precision, where it was trained, and it does buy something under mild quantization, since the penalized model survives eight bits where the unstructured one does not. It buys nothing at six bits, where both fail in every seed, because quantization moves the weights and the invariant the penalty was protecting moves with them, which is the content of Remark 2.

That conclusion would be worth little if it held only at one choice of the two penalty weights, so they are swept over three orders of magnitude around the value used, both weights together, with the same backbone, the same data, the same schedule and the first seed. With both set to one the model already leaves the orthant at eight bits, at -2.83×10^{-2} , and its mixed eight-bit variant reaches -1.58×10^{-2} . With both set to ten, the value used throughout, it survives eight bits in both arrangements and fails at six, at -7.60×10^{-2} . With both set to one hundred it survives eight bits and still fails at six, at -7.71×10^{-2} , at the cost of a single-precision accuracy that falls from 5.00×10^{-3} to 7.19×10^{-3} . With both set to one thousand the accuracy falls to 1.21×10^{-2} , the mixed eight-bit arrangement fails again at -8.33×10^{-3} and six bits fail at -7.81×10^{-2} . Six bits are therefore lost at every weight tested, and raising the weight buys at most one bit of margin while costing accuracy, which is what a penalty evaluated on the training distribution and in the training format is expected to do: it moves the weights so that the trajectories it saw stay inside the orthant, and the quantizer moves the weights again.

The fourth observation concerns accuracy under quantization, where the two blocks behave differently. Quantizing every tensor to eight bits costs the structured model a factor of 16 in accuracy, while quantizing the learned block alone and keeping the transport block in single precision costs a factor of 2.6 and yields 1.34×10^{-2} , which remains a usable trajectory. Section 5.2 explains the difference in terms of Proposition 3 and recommends the mixed arrangement. The clamped model deserves a separate comment. Projecting the unstructured prediction onto the nonnegative orthant removes the negative values by construction and changes little else, since the error moves from 2.00×10^{-1} to 1.99×10^{-1} in eight bits while the budget error stays at 2.38×10^{-1} against 4.82×10^{-3} for the structured model in the mixed arrangement. The projection repairs the symptom that is visible and leaves the balance law that is not.

A fifth observation concerns the choice among the floating-point formats, and it goes against the habit acquired in training. The bfloat16 format, which is the default reduced format of most training pipelines because its exponent range matches that of single precision, is markedly worse here than half precision, 7.28×10^{-2} against 1.84×10^{-2} for the structured model and 4.29×10^{-1} against 4.81×10^{-2} for the unstructured one. The reason is that the two formats trade the same sixteen bits differently. The state of this system is of order one in every field after the normalization, so the exponent range of bfloat16 buys nothing, while its eight significant bits against the eleven of half precision make its unit round-off eight times larger, which the rollout then accumulates over the macro steps. Tensor-float arithmetic sits between the two, at 5.81×10^{-3} against 5.07×10^{-3} , and it is the setting under which the timings of Section 5.1 are taken. The practical rule is that a surrogate of a well-scaled physical state should be reduced along the mantissa and not along the exponent, which is the opposite of the rule that governs the choice of a training format.

Figure 1 shows the endothelial density, the oxygen and the reserve at half of the horizon and at the final time for one test trajectory, with the full-order solution and the prediction of the structured surrogate superimposed. Figure 2 follows the error of the four models along the trajectory in single and in half precision, and shows that the error grows smoothly with the number of macro steps rather than through a sudden loss of stability. Figure 3 reports the smallest value attained and the error of each model as the inference format is degraded from single precision to four integer bits, and it is the graphical form of the statement that only one of the four curves of the left panel stays on the axis. The depth of the violation of the other three grows with the coarseness of the quantizer without ever returning to zero, while the right panel shows that at six and four bits the unstructured and the penalized models have left the accurate regime altogether, with errors above one, so at those widths the depth of the excursion below zero is no longer an informative quantity and only the fact of the excursion is.

Table 6 resolves the sign statistic field by field and identifies where a violation appears first. The fields of the model differ in magnitude by three orders, since the fibronectin and the oxygen are of order one while the protease and the inhibitor are three orders smaller in most of the domain, and a quantizer with one scale per tensor would destroy the small fields before the large ones. The tables reported here therefore use one scale per field for states and activations and one scale per output channel for the convolution weights, which is the standard arrangement of an integer inference engine, and even with it the fields that leave the orthant for the unstructured model at eight bits are the endothelial density, at -3.25×10^{-3} , and the reserve, at -1.64×10^{-4} , while the protease and the inhibitor are held at zero. The structured model attains exactly zero in every field and every format, and zero is attained rather than approached because the protease and the inhibitor start from zero and are produced from it.

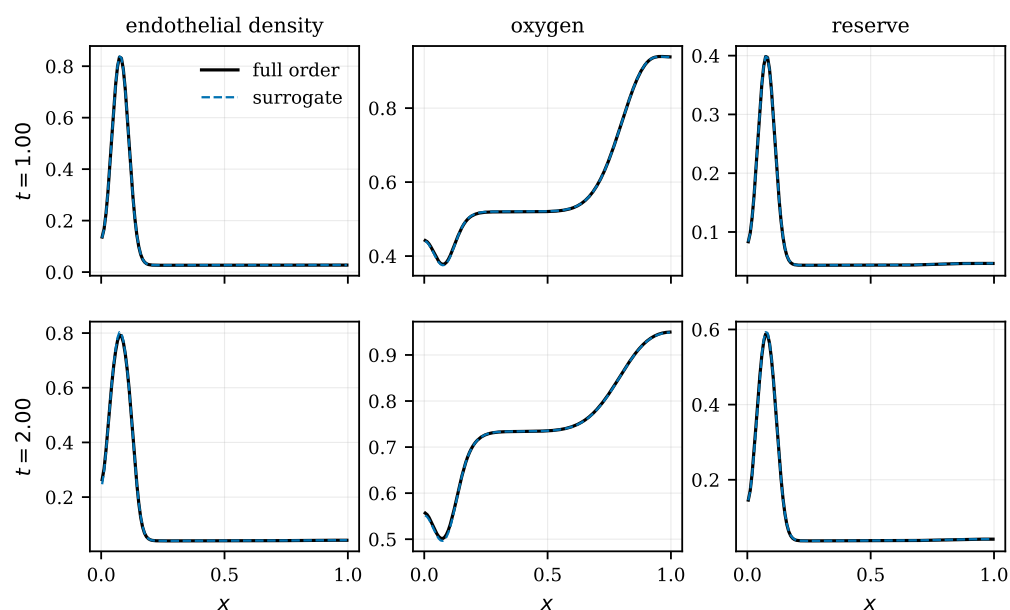


Figure 1. Full-order solution and prediction of the structured surrogate on a test trajectory.

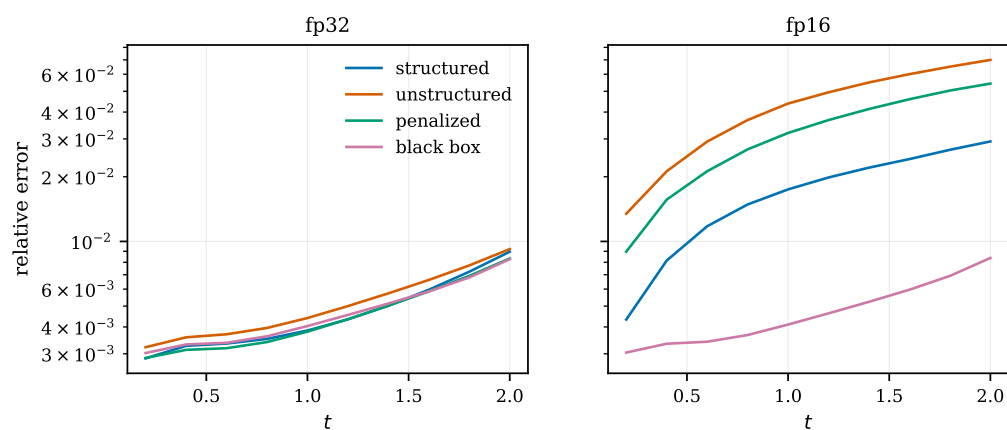


Figure 2. Growth of the rollout error along the trajectory for the four models.

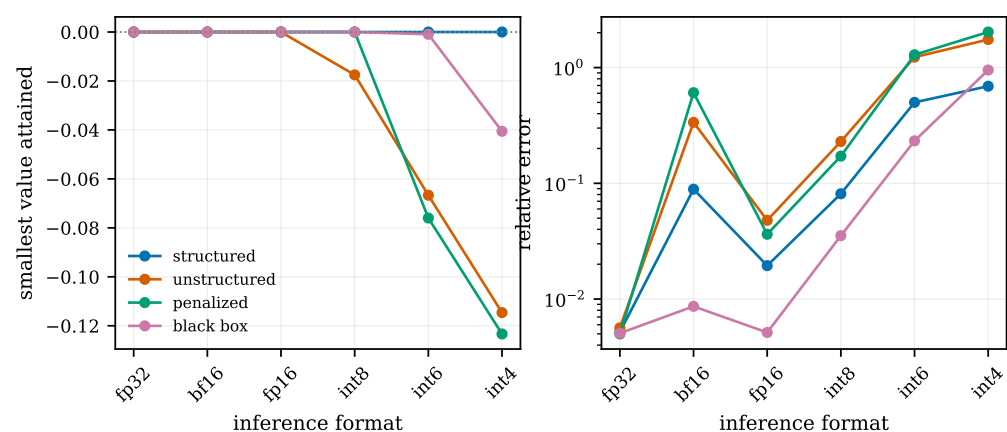


Figure 3. Smallest value attained and rollout error as the inference format is degraded.

Table 6. Smallest value attained by each field over the test trajectories.

Model	<i>C</i>	<i>P</i>	<i>I</i>	<i>F</i>	<i>O</i>	<i>E</i>
fp32						
structured	0	0	0	8.21×10^{-1}	1.00×10^{-1}	1.66×10^{-2}
unstructured	0	0	0	8.12×10^{-1}	1.00×10^{-1}	1.60×10^{-2}
penalized	0	0	0	8.15×10^{-1}	1.00×10^{-1}	1.63×10^{-2}
black box	0	0	-2.72×10^{-5}	8.16×10^{-1}	1.00×10^{-1}	1.62×10^{-2}
fp16						
structured	0	0	0	8.21×10^{-1}	1.00×10^{-1}	1.66×10^{-2}
unstructured	0	0	0	8.10×10^{-1}	1.00×10^{-1}	1.57×10^{-2}
penalized	0	0	0	8.13×10^{-1}	1.00×10^{-1}	1.70×10^{-2}
black box	0	0	-2.75×10^{-5}	8.15×10^{-1}	1.00×10^{-1}	1.62×10^{-2}
int8						
structured	0	0	0	8.15×10^{-1}	1.00×10^{-1}	1.26×10^{-2}
unstructured	-3.25×10^{-3}	0	0	7.99×10^{-1}	1.00×10^{-1}	-1.64×10^{-4}
penalized	0	0	0	8.06×10^{-1}	1.00×10^{-1}	7.56×10^{-3}
black box	0	0	0	8.13×10^{-1}	1.00×10^{-1}	5.68×10^{-3}
int8 mixed						
structured	0	0	0	8.21×10^{-1}	1.00×10^{-1}	1.68×10^{-2}
unstructured	-1.31×10^{-3}	0	0	7.95×10^{-1}	1.00×10^{-1}	5.79×10^{-3}
penalized	-2.45×10^{-3}	0	0	8.04×10^{-1}	1.00×10^{-1}	1.30×10^{-2}
black box	0	0	0	8.18×10^{-1}	1.00×10^{-1}	1.66×10^{-2}

The spread over the five training seeds is small enough not to affect any of these statements. In single precision the rollout error of the structured model varies between 4.70×10^{-3} and 5.25×10^{-3} across the seeds and that of the unstructured model between 5.20×10^{-3} and 5.97×10^{-3} , so the two models overlap in accuracy. The difference in the sign statistic is categorical rather than statistical, since the structured model has zero violations in five seeds out of five in every format of Table 5 while the unstructured model has three out of five at eight bits and five out of five at six. No interval on a proportion is needed to separate a property that never fails from one that fails in the majority of the runs.

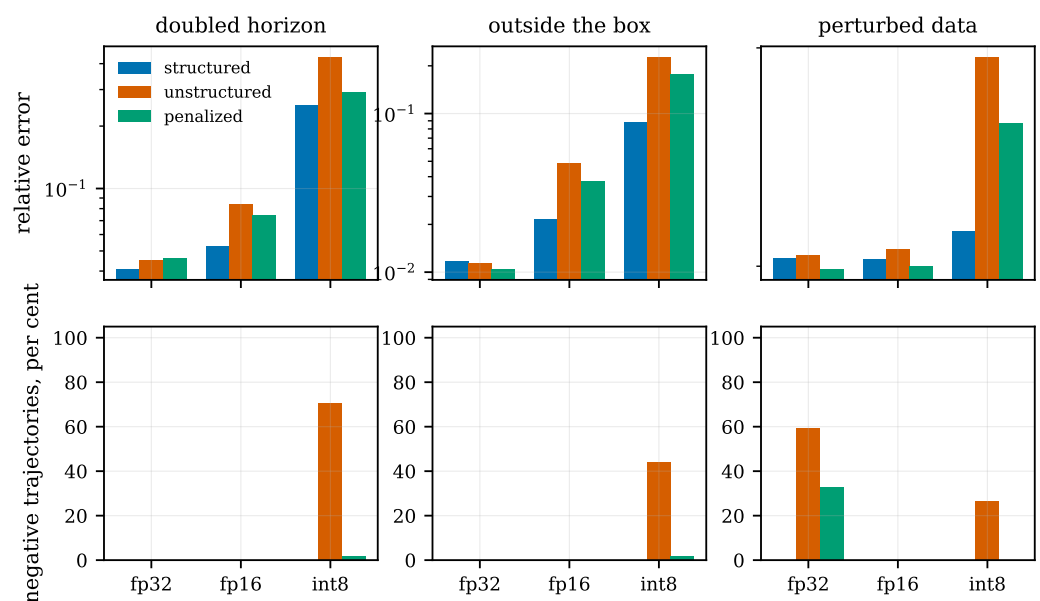
The training cost is one-off and is not part of the inference budget, and it is quoted here for completeness. One model of the reference width takes 131 s on the same processor, and the generation of the whole training, validation and test set takes a comparable time, so the entire study behind Table 5 is a matter of hours rather than of days. The comparison with the reference scheme in Section 5.1 ignores that cost, which is the right convention when the surrogate is reused across a campaign and the wrong one when a single trajectory is required.

4.2. Behaviour Outside the Training Conditions

Table 7 and Figure 4 report three regimes that the training never saw. The horizon is doubled, so that the models are asked for twice as many macro steps as they were trained to produce. The sampled parameters are drawn from a box extended by one quarter of its width on each side, so that a quarter of the draws fall outside the training range. The initial data are perturbed by multiplicative noise of ten per cent, which is the situation of a surrogate driven by measured rather than by synthetic input. Each regime carries its own full-order reference, computed with the same integrator.

Table 7. Behaviour of the surrogates outside the conditions used for training.

Setting	Error	Smallest Value	Affected Trajectories, per Cent	Budget Error
doubled horizon				
structured, fp32	4.08×10^{-2}	0	0	7.92×10^{-3}
structured, int8	2.51×10^{-1}	0	0	1.01×10^{-1}
unstructured, fp32	4.51×10^{-2}	0	0	1.14×10^{-2}
unstructured, int8	4.31×10^{-1}	-3.74×10^{-2}	70	4.57×10^{-1}
penalized, fp32	4.59×10^{-2}	0	0	9.39×10^{-3}
penalized, int8	2.93×10^{-1}	-8.22×10^{-3}	2	2.36×10^{-1}
parameters outside the training box				
structured, fp32	1.17×10^{-2}	0	0	5.44×10^{-3}
structured, int8	8.87×10^{-2}	0	0	8.74×10^{-2}
unstructured, fp32	1.13×10^{-2}	0	0	4.83×10^{-3}
unstructured, int8	2.27×10^{-1}	-1.46×10^{-2}	44	3.24×10^{-1}
penalized, fp32	1.04×10^{-2}	0	0	4.34×10^{-3}
penalized, int8	1.78×10^{-1}	-6.63×10^{-3}	2	1.21×10^{-1}
perturbed initial data				
structured, fp32	3.03×10^{-1}	0	0	5.63×10^{-3}
structured, int8	3.14×10^{-1}	0	0	1.75×10^{-1}
unstructured, fp32	3.04×10^{-1}	-1.58×10^{-4}	59	4.97×10^{-3}
unstructured, int8	3.95×10^{-1}	-1.46×10^{-2}	27	2.80×10^{-1}
penalized, fp32	2.99×10^{-1}	-1.17×10^{-4}	33	4.48×10^{-3}
penalized, int8	3.62×10^{-1}	0	0	1.29×10^{-1}

**Figure 4.** Accuracy and sign violations of the surrogates outside the training conditions.

The pattern is consistent across the three regimes. In single precision the structured and the unstructured models are comparable in accuracy, the structured one reaching 4.08×10^{-2} against 4.51×10^{-2} on the doubled horizon and 1.17×10^{-2} against 1.13×10^{-2} outside the parameter box. On perturbed initial data both lose accuracy to about 3.03×10^{-1} and 3.04×10^{-1} respectively, which is the effect of the noise on the reference itself rather than a failure of either model. What separates them is the sign. On perturbed initial data the unstructured model produces a negative value in 59 per cent of the trajectories already in single precision and the penalized model in 33 per cent, so the violation is not confined to reduced precision but appears as soon as the input leaves the distribution the penalty was fitted on. Under eight-bit quantization the unstructured model produces a negative

value in 70 per cent of the trajectories on the doubled horizon, in 44 per cent outside the parameter box and in 27 per cent on perturbed data. The structured model produces none, in every regime and every format, which is again the statement of Theorem 1 and not a property of the particular training run.

The doubled horizon deserves a separate reading, because it is where Proposition 4 can be checked against a measurement rather than used as a bound. Asking the structured surrogate for twice as many macro steps as it was trained to produce raises its error from the in-distribution value of the same seed, 5.19×10^{-3} , to 4.08×10^{-2} , a factor of 7.9. The decomposition predicts a factor of two from the number of steps alone, so the remainder is the geometric amplification, and equating the two gives an amplification per macro step of about 1.16. That the estimate is close to one and not far above it is the reason the rollout degrades smoothly rather than diverging, which is what Figure 2 shows in the in-distribution case, and it is a property of the dynamics that neither architecture can change. What the architecture does change is that the degraded trajectory is still a trajectory of densities.

The mechanism behind the perturbed-data column is worth stating, because it is what made the transport block of Section 3.1 unconditional. Multiplicative noise applied cell by cell to the matrix field produces gradients of order h^{-1} , and the interface velocities of the haptotaxis step grow with them, so the advective Courant number over a macro half step rises above one on inputs of this kind although it is 6.27×10^{-2} on the training distribution. A transport block whose nonnegativity rested on the Courant condition would lose it there, which is not a failure of the theorem but a failure of its hypothesis, and it is why the implementation substeps the haptotaxis and clips its diagonal coefficient instead of assuming the condition. The learned block meets the same unfamiliar input and produces equally unreliable coefficients, but those coefficients enter a quotient whose numerator is a sum of nonnegative terms, so an unfamiliar input degrades the accuracy without moving the prediction out of the admissible set. This is the difference the tables record, and it is a difference in algebra and not of training.

5. Cost of Inference

5.1. Cost on the Target Processor and the Classical Alternative

All timings are wall-clock medians over repeated runs after warm-up, with the device synchronized around each timed region, on a single NVIDIA GeForce RTX 4080 SUPER with the propagators and the weights assembled once outside the timed region. Table 8 reports the time for one macro step and for one trajectory, the throughput and the peak memory, for the reference integrator and for the structured surrogate at four batch sizes and four formats.

The learned map replaces 100 reference steps by one evaluation, and the measured saving follows. One macro step of the reference integrator costs 49.06 ms at batch one and one evaluation of the surrogate costs 0.668, a factor of 73, and the factor is 73 at batch two hundred and fifty six, where the throughput reaches 37,170 trajectories per second against 507 for the reference. The time of the surrogate is almost independent of the format, since 0.668 ms in single precision becomes 0.713 in half precision, and the reason is that at this problem size the evaluation is dominated by the latency of the kernel launches rather than by arithmetic. The benefit of the reduced formats is therefore in the stored size and in the memory traffic rather than in the time. Table 9 reports the stored size of the learned block of the structured surrogate in each format.

Table 8. Inference cost of one macro step and of one trajectory on the target processor.

Setting	Macro Step, ms	Trajectory, ms	Trajectories per Second	Speed-Up
Full-order model				
fp64, batch 1	49.421	494.2	2.0	–
fp64, batch 256	50.389	503.9	508.0	–
fp32, batch 1	49.057	490.6	2.0	–
fp32, batch 256	50.448	504.5	507.5	–
Structured surrogate				
fp32, batch 1	0.6684	6.68	150	73
fp32, batch 8	0.6840	6.84	1170	74
fp32, batch 64	0.7056	7.06	9071	73
fp32, batch 256	0.6887	6.89	37,170	73
tf32, batch 1	0.6715	6.71	149	73
tf32, batch 8	0.7001	7.00	1143	72
tf32, batch 64	0.7118	7.12	8991	72
tf32, batch 256	0.6972	6.97	36,720	72
bf16, batch 1	0.7020	7.02	142	70
bf16, batch 8	0.7351	7.35	1088	69
bf16, batch 64	0.7208	7.21	8880	71
bf16, batch 256	0.7354	7.35	34,813	69
fp16, batch 1	0.7134	7.13	140	69
fp16, batch 8	0.7408	7.41	1080	68
fp16, batch 64	0.7335	7.33	8726	70
fp16, batch 256	0.7333	7.33	34,912	69

Table 9. Stored size of the learned block of the structured surrogate.

Format	Stored Size, kB
fp32	115.8
tf32	115.8
bf16	57.9
fp16	57.9
int8	29.0

A claim that a reduced format pays is worth only the conditions attached to it, so they are measured. At the size of Table 8 the macro step is a handful of small kernels acting on a state of 6×128 numbers, the device is far from saturated, and the time is set by the launches, so half precision buys nothing in time. What it buys there is footprint: the learned block occupies 115.8 kilobytes in single precision, 57.9 in half precision and 29.0 at eight bits, while the four transport matrices occupy 256 kilobytes and are shared by every trajectory of a batch, so a batched sweep amortizes them and a single-query deployment does not. Table 10 sweeps the grid and the batch until the arithmetic dominates, and there the ordering changes. At $M = 2048$ with a batch of five hundred and twelve one macro step costs 14.08 ms in single precision, and half precision divides it by 1.90 and bfloat16 by 1.59 while tensor float leaves it unchanged, which is what the traffic predicts, since the two sixteen-bit formats halve the bytes moved and tensor float does not. The same batch at $M = 512$ gives 1.73 and 1.50. The reduced formats of this paper are therefore worth their accuracy when the deployment is bound by throughput on a large grid or a large batch, and worth only their footprint when it is bound by latency on one small state, which is the situation of an embedded target answering one query at a time and for which the operative number is the 29.0 kilobytes.

Table 10. Time of one macro step in single precision and the factor by which a reduced format divides it. The three rows at $M = 128$ with a small batch are the ones on which the library selects the fallback algorithm of Table 11, so the factor there measures the kernel and not the arithmetic.

Setting	fp32, ms	tf32	bf16	fp16
$M = 128$, batch 1	12.195	16.93	16.61	16.81
$M = 128$, batch 64	38.726	41.16	52.69	53.13
$M = 128$, batch 512	9.591×10^{-1}	0.98	1.25	1.29
$M = 512$, batch 1	8.849×10^{-1}	0.96	0.95	0.95
$M = 512$, batch 64	9.276×10^{-1}	1.00	0.97	0.99
$M = 512$, batch 512	3.176	0.99	1.50	1.73
$M = 2048$, batch 1	1.034	1.13	1.11	1.11
$M = 2048$, batch 64	1.572	0.97	1.13	1.19
$M = 2048$, batch 512	14.079	1.01	1.59	1.90

What moves the time at this size is the kernel that the library selects for the convolutions of the learned block, and the measurement below is a case worth reporting. With tensor-float arithmetic enabled the macro step costs between 0.673 and 0.716 ms at every width tested. With strict single precision the same evaluation costs 0.668 ms at width thirty-two and 12.23 at width forty-eight, a factor of 18, and 46.3 at width ninety-six, because the library falls back on a slow algorithm for those channel counts. Enabling the autotuning of the library does not repair it. The observation is not about the method and it does not affect any accuracy reported here, but it is the kind of fact that decides whether a deployment meets its latency budget, and it says that the width of the learned block has to be chosen against the kernels of the target rather than on parameter count alone.

The pattern is the signature of algorithm selection and not of arithmetic, and four facts identify it. The widths sixteen and thirty-two cost the same in both settings, so the fallback is not a cost of single precision as such. The widths forty-eight and ninety-six cost an order of magnitude more in strict single precision only, so it is those shapes that lack a fast candidate once the tensor-core path is excluded. Enabling the exhaustive autotuner moves nothing, at 12.23 against 12.23 ms at width forty-eight, so the selection is not a mis-prediction of the heuristic but the absence of a faster algorithm to select. And the effect disappears with the batch: the same evaluation at width forty-eight in strict single precision costs 12.23 ms at batch one and 9.59×10^{-1} at batch five hundred and twelve, where a different algorithm becomes admissible. Table 11 reports the four settings. Whichever algorithm the library substitutes, the arithmetic performed is the same and no accuracy in this paper depends on it, which is why the effect appears here as a deployment fact and not as a property of the method.

Table 11. Time of one macro step of the structured surrogate under four library settings.

Setting	$w = 16$	$w = 32$	$w = 48$	$w = 96$
heuristic selection, single precision	0.663	0.668	12.232	46.277
autotuning, single precision	0.680	0.673	12.227	46.277
heuristic selection, tensor float	0.670	0.671	0.708	0.719
autotuning, tensor float	0.673	0.677	0.715	0.716

Figure 5 shows how the two costs respond to the batch, and its two panels say different things. In the left panel the time of one macro step is nearly flat for the surrogate, from 0.668 ms at batch one to 0.689 at batch two hundred and fifty six, and equally flat for the reference integrator, from 49.06 to 50.45, because at these sizes neither computation saturates the device and the cost is set by the number of kernel launches rather than by the work inside them. In the right panel the throughput therefore grows almost linearly with the batch for both methods, and at batch two hundred and fifty six it reaches 37,170 trajectories

per second for the surrogate against 507 for the reference integrator, at a peak memory of 7743 megabytes. The practical reading is that a parameter sweep should be batched whichever method is used, and that the advantage of the surrogate comes from the steps it removes and not from a better use of the device at a fixed number of steps.

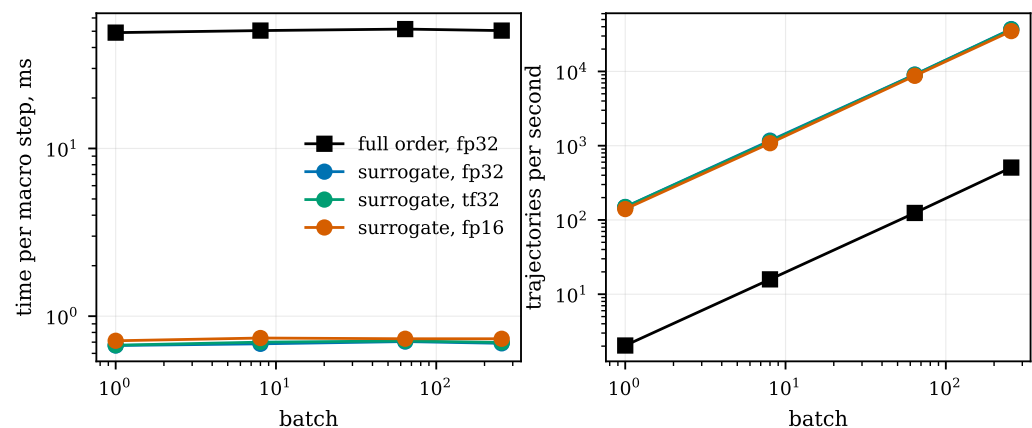


Figure 5. Time per macro step and throughput of the surrogate and of the reference integrator.

The split of the arithmetic between the two blocks explains why the recommendation of Section 5.2 is the one it is. One macro step performs 3.82 million multiply and accumulate operations, of which 3.69 million are the convolutions of the learned block and only 132 thousand are the two transport half steps, so the learned block carries 97 per cent of the arithmetic and all of the parameters. Reducing the format of the transport block therefore saves almost nothing and costs the exact conservation of mass, while reducing the format of the learned block is where a reduced format pays. The dense propagators are four matrices shared by every trajectory of a batch, so their memory is amortized over the batch, which is a further reason to leave them alone.

A speed-up against the reference integrator run at its own step is not by itself an argument, because the reference integrator has no stability restriction that forces that step. Its diffusion is exact for any step and its reaction is unconditionally positive, so the step is limited by accuracy alone, and the comparison that decides the question is with the same scheme run at the macro step. Table 12 and Figure 6 make that comparison. Run with the macro step itself, and therefore at the same number of steps as the surrogate, the reference scheme reaches 1.95×10^{-2} against 5.19×10^{-3} for the structured surrogate, a factor of 3.8, at 5.02 ms per trajectory against 6.86. Halving its step brings the reference scheme to 5.79×10^{-3} , an accuracy the surrogate already exceeds, at 10.03 ms, so the surrogate delivers that accuracy in 1.5 times less time, and the widest learned block reaches 3.29×10^{-3} at 6.82 ms. The advantage is real and it is bounded. The reference scheme is second order in the step, so its curve descends steeply and overtakes the surrogate as soon as an accuracy better than about 1.05×10^{-3} is required, which is what Figure 6 shows. The trivial predictor that returns the initial state and holds it over the horizon has an error of 1.31, which fixes the scale against which all of these numbers are to be read.

Figure 7 completes the picture on the side of the surrogate, where increasing the width of the learned block from 4540 to 103,980 parameters improves the error from 1.44×10^{-2} to 3.29×10^{-3} while the unstructured model of the same size stops improving at 4.84×10^{-3} . The same figure shows that they separate rather than converge. The structured model improves monotonically over the range tested, so its capacity is still the limiting factor at the largest width, while the unstructured model of the same size gains nothing from the last refinement, which is consistent with it spending capacity on reproducing a constraint that the structured model obtains from its algebra. The comparison is not a general statement

about capacity, since both models remain far from the accuracy of the reference scheme at a small step, but it does say that the architectural constraint is not paid for in parameters.

Table 12. Accuracy and cost of the surrogate against the reference scheme run with larger steps.

Method	Steps	Error	Smallest Value	Trajectory, ms
Reference scheme at decreasing step				
$\tau = 0.2$	10	1.95×10^{-2}	0	5.02
$\tau = 0.1$	20	5.79×10^{-3}	0	10.03
$\tau = 0.04$	50	1.05×10^{-3}	0	25.03
$\tau = 0.02$	100	2.72×10^{-4}	0	50.48
$\tau = 0.008$	250	4.28×10^{-5}	0	125.91
$\tau = 0.002$	1000	5.74×10^{-8}	0	510.81
Structured surrogate at increasing width				
width 16	10	1.44×10^{-2}	0	6.60
width 32	10	7.40×10^{-3}	0	6.58
width 48	10	5.19×10^{-3}	0	6.86
width 96	10	3.29×10^{-3}	0	6.82
Initial state held	0	1.31×10^0	0	0

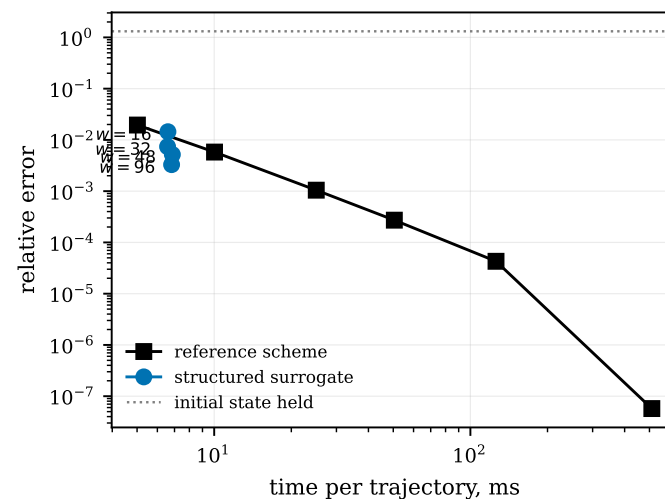


Figure 6. Rollout error against time per trajectory for the two families of methods.

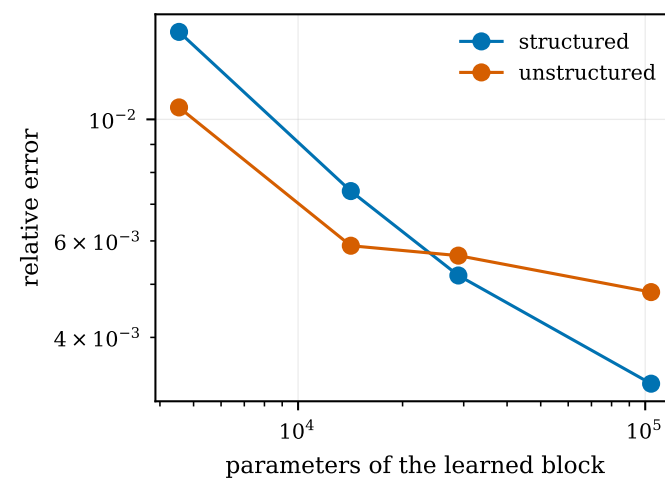


Figure 7. Rollout error against the number of parameters of the learned block.

5.2. What to Quantize

Proposition 3 predicts that quantizing the transport block trades the exact conservation of the mass of the diffusing species for a drift controlled by the quantization step, and Table 13 measures both sides of that prediction. In eight bits the quantization step of the propagators is 6.77×10^{-3} , the computable constant δ of (19) is 5.98×10^{-3} , the worst case $Ms/2$ is 4.33×10^{-1} , and the drift realized over the whole horizon is 8.24×10^{-3} against a bound of 1.27×10^{-1} built from δ . The bound holds with a wide margin, as it must, since it assumes that the entries of a column all round in the same direction. Left unquantized, the propagator conserves the mass to 9.84×10^{-16} , which is round-off.

Table 13. Mass drift of the quantized transport block against the bound.

Format	Step s	δ	$Ms/2$	Bound over Horizon	Realized Drift
int8	6.77×10^{-3}	5.98×10^{-3}	4.33×10^{-1}	1.27×10^{-1}	8.24×10^{-3}
int6	2.77×10^{-2}	2.93×10^{-2}	1.77×10^0	7.83×10^{-1}	2.19×10^{-1}
int4	1.23×10^{-1}	4.76×10^{-2}	7.86×10^0	1.53×10^0	1.34×10^0
exact	0	0	0	0	9.84×10^{-16}

The budget column of Table 5 tracks the same effect on the quantity the model is about, the total energy of the reserve, of the structure and of the protease. In single precision the structured surrogate closes it to 1.58×10^{-3} and the unstructured one to 1.94×10^{-3} , so at that precision the budget is a matter of approximation and not of architecture. Half precision leaves both at 2.27×10^{-3} and 1.33×10^{-2} . Eight-bit quantization of every tensor separates them, at 8.09×10^{-2} for the structured model against 2.47×10^{-1} , and the mixed arrangement recovers most of the difference, at 4.82×10^{-3} . Figure 8 follows the same quantity along the trajectory rather than at the final time, in single precision and after eight-bit quantization of every tensor, and it shows that the budget error of the structured model grows smoothly with the macro steps in both formats while the two unstructured models separate from it as soon as the quantizer is coarse. Two mechanisms are superimposed here and the drift table separates them. The transport block contributes the quantity bounded by Proposition 3, which the mixed arrangement removes entirely, and the learned block contributes an error that no theorem controls, since nothing constrains a learned production and destruction to respect a balance law. The paper claims the first and measures the second, and the gap between the two is the boundary of the construction. An architecture that also made the budget exact would need the learned coefficients to be constrained pairwise, in the way a conservative Patankar scheme constrains them, which is a natural next step and is not what is proved here.

The practical consequence is the arrangement recommended here. The transport block carries no parameters, occupies the memory of four dense matrices that are shared by every trajectory of a batch, and is responsible for the exact conservation, so quantizing it saves little and costs the balance law, and its cost is the subject of Remark 1. The learned block carries all the parameters and most of the arithmetic, so it is the part worth reducing. Table 5 confirms the reasoning in accuracy, where the fully quantized structured model reaches 8.35×10^{-2} and the mixed arrangement 1.34×10^{-2} with a budget error of 4.82×10^{-3} against 8.09×10^{-2} . Theorem 1 covers both arrangements, so the choice between them is about accuracy and about the balance law, never about admissibility.

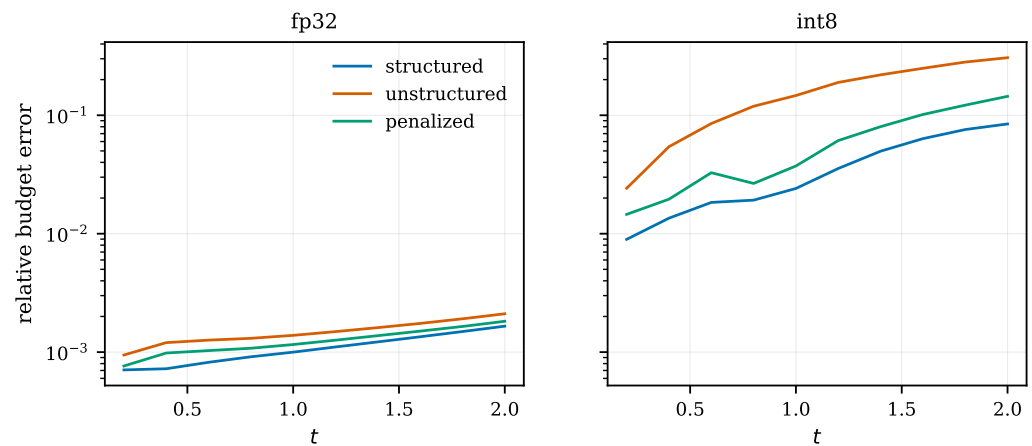


Figure 8. Relative error on the discrete energy budget along the trajectory.

The same reasoning determines what to do with the two blocks when the budget is tighter still. Reducing the learned block from eight to six and then to four bits costs accuracy in the way Proposition 4 predicts, since the term η grows with the quantization step while ε and the amplification are unchanged, and the measured sequence 1.34×10^{-2} , 4.76×10^{-2} and 2.84×10^{-1} follows the growth of the step rather than any loss of stability. The trajectory at four bits is no longer usable for a quantitative answer, and it is still a trajectory of nonnegative densities, which is what separates a degraded prediction from an inadmissible one. If a deployment has to operate at that width, the remedy is to recover accuracy by training in the target format rather than to relax the architecture, since the guarantee is what costs nothing and the accuracy is what has to be paid for.

It is then possible to say which formats a quantitative prediction can use, rather than to report the accuracy of each and leave the choice open. The yardstick is not an absolute tolerance but the accuracy the surrogate has to beat to be worth training, which is the error of the reference scheme run at the same macro step and at comparable cost, 1.95×10^{-2} . By that criterion single precision at 5.07×10^{-3} , tensor float at 5.81×10^{-3} , half precision at 1.84×10^{-2} and the mixed eight-bit arrangement at 1.34×10^{-2} are usable, and the rest are not: bfloat16 at 7.28×10^{-2} , eight bits applied to every tensor at 8.35×10^{-2} , the mixed six-bit arrangement at 4.76×10^{-2} and the mixed four-bit arrangement at 2.84×10^{-1} all cost more accuracy than the surrogate buys. Read as a field-normalized error the same list says that the usable formats keep the trajectory within two per cent of the reference solution and that the others do not. The practically relevant range is therefore narrow and real, from half precision to eight bits on the learned block with the transport block left in single precision. Six and four bits appear in this paper as a test of the invariant under a quantizer coarse enough to destroy the accuracy, which is the regime in which the difference between an architectural and a trained invariant is visible, and not as a recommendation.

Three checks make the guarantee auditable on the target rather than only on paper, and none of them requires running the model. The first is that the stored propagators are entrywise nonnegative after quantization, which holds for any sign-faithful quantizer and is verified by inspecting the smallest entry. The second is that the advective Courant number of the taxis step stays below one by more than the unit round-off of the working format, which is a property of the parameter box and of the macro step and is verified once, the margin measured here being wide. The third is that the output nonlinearity of the learned block is evaluated as a nonnegative function in the target arithmetic, which is the only place where an implementation could break the argument, for instance by replacing the softplus with a piecewise approximation that undershoots. If the three hold, Theorem 1 transfers to the deployed artefact without further assumptions, and no output projection is needed. The same three checks fail immediately for the unstructured architecture, since

there is no step at which the sign of the update is determined by the form of the expression rather than by the values it combines.

5.3. Higher Dimensions and the Cost of the Transport Block

The dense assembly that makes the sign structure visible to the arithmetic is also the part of the construction that does not scale, and the two facts have to be separated, because the first is what the paper proves and the second is a property of one implementation of one operator. On a grid of M cells per direction in d dimensions the assembled propagator of one species is an $M^d \times M^d$ matrix, so it holds M^{2d} numbers and costs as many multiplications, which is 256 megabytes and 6.7×10^7 multiplications per species already at $M = 8192$ in one dimension, and 256 gigabytes at $M = 64$ in three. Two routes avoid it and they are not equivalent. Table 14 and Figure 9 measure both against the assembly, for one species and one macro half step, on the processor of Section 5.1.

Table 14. Cost in microseconds, storage and sign behaviour of the three routes to the same diffusion propagator, for one species and one macro half step. The storage column is the assembled operator, against M numbers for the spectral route and dM^2 for the factorized one.

M	Assembled	Factorized	Spectral	Storage	Smallest	Smallest, Spectral
one dimension						
128	7.34	–	97.6	64 kiB	3.3×10^{-16}	-3.0×10^{-8}
256	9.47	–	128	256 kiB	5.3×10^{-16}	-6.0×10^{-8}
512	9.39	–	129	1 MiB	1.4×10^{-15}	-8.9×10^{-8}
1024	8.68	–	96.1	4 MiB	2.6×10^{-15}	-1.2×10^{-7}
2048	12.8	–	100	16 MiB	1.7×10^{-14}	-1.5×10^{-7}
4096	77	–	96.6	64 MiB	5.4×10^{-14}	-1.2×10^{-7}
8192	391	–	104	256 MiB	9.8×10^{-14}	-1.3×10^{-7}
two dimensions						
32	9.16	27.9	195	4 MiB	6.0×10^{-16}	-1.2×10^{-7}
48	14.8	27.7	193	20.2 MiB	4.8×10^{-16}	-1.2×10^{-7}
64	29.6	28	193	64 MiB	5.2×10^{-16}	-6.0×10^{-8}
96	493	27.9	194	324 MiB	4.4×10^{-16}	-6.0×10^{-8}
128	–	24.4	194	1 GiB	3.3×10^{-16}	-1.5×10^{-7}
256	–	38.1	260	16 GiB	4.9×10^{-16}	-1.8×10^{-7}
512	–	57.8	260	256 GiB	1.4×10^{-15}	-1.8×10^{-7}
1024	–	185	385	4 TiB	2.5×10^{-15}	-1.6×10^{-7}
three dimensions						
16	–	65.4	–	64 MiB	2.8×10^{-16}	–
32	–	70	–	4 GiB	5.3×10^{-16}	–
64	–	63.1	–	256 GiB	5.4×10^{-16}	–

The first route diagonalizes the propagator with the transform that Section 2.2 describes, at a cost of $\mathcal{O}(M^d \log M)$ and a storage of M numbers. It is asymptotically the cheaper of the two and it is the wrong one here. In one dimension it is slower than the assembled product up to $M = 4096$, where the two are 7.70×10^{-2} and 9.66×10^{-2} ms, and it wins only at $M = 8192$, where they are 3.91×10^{-1} and 1.04×10^{-1} ; the assembly is a single general matrix multiplication and the transform is a sequence of smaller kernels, which at these sizes decides the comparison against the exponents. More to the point, it returns negative entries at every grid tested, from -2.98×10^{-8} at $M = 128$ to -1.34×10^{-7} at $M = 8192$ on a state of order one, and in two dimensions from -5.96×10^{-8} to -1.79×10^{-7} , while the assembled product returns a smallest entry of 3.35×10^{-16} to 9.78×10^{-14} , which is the round-off of a sum of nonnegative numbers. The two computations agree to that same round-off. They are the same operator and only one of them satisfies hypothesis (H1) of Theorem 1, which is the content of the choice made in Section 2.2.

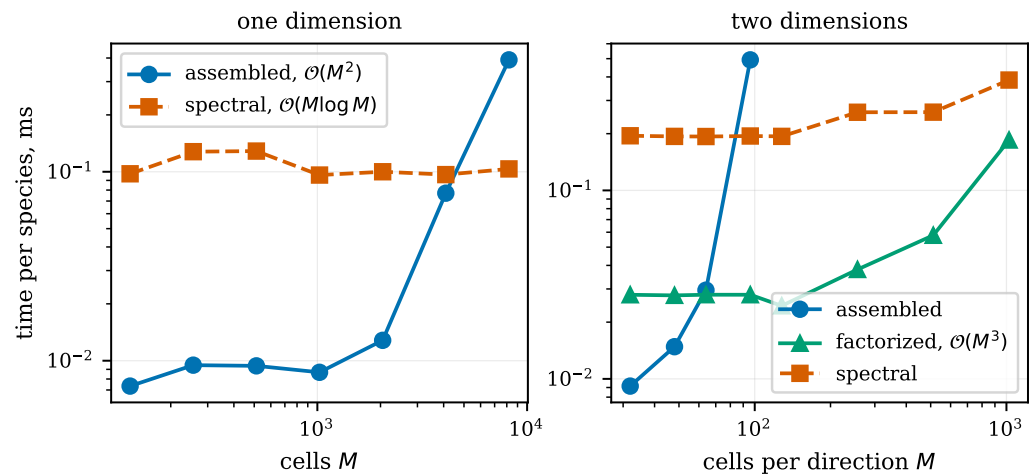


Figure 9. Cost of the three routes to the same diffusion propagator, in one and in two dimensions.

The second route is the tensor-product factorization. The Neumann Laplacian on a Cartesian grid is a sum of one-dimensional operators acting on different directions, which commute, so the propagator of a species factors as $\mathcal{E}^{(d)} = \mathcal{E} \otimes \cdots \otimes \mathcal{E}$ and is applied one direction at a time [67]. A Kronecker product of entrywise nonnegative matrices is entrywise nonnegative, and a Kronecker product of row-stochastic matrices is row-stochastic, since the row sums multiply, so $\mathcal{E}^{(d)}$ has exactly the two properties of Proposition 1 and every intermediate of the directional application is a nonnegative combination of nonnegative numbers. The haptotaxis step splits by direction in the same way, each direction carrying the form (10) with its own clipped diagonal, so (16) applies per direction with the same Δ and the same h , and (17) gives the same unconditional value of n . Hypotheses (H1) to (H3) therefore hold verbatim in any dimension, and Theorem 1 holds with them. The cost is dM^{d+1} multiplications and dM^2 numbers of storage instead of M^{2d} and M^{2d} , and the measurements follow: in two dimensions the factorized route is already faster than the assembled one at $M = 64$ and is faster by a factor of 17.6 at $M = 96$, which is the last grid at which the assembly fits in the memory of the device, and it runs at 1.85×10^{-1} ms at $M = 1024$, where the assembled operator would need four terabytes. In three dimensions it costs 6.31×10^{-2} ms at $M = 64$, where the assembled operator would need 256 gigabytes, and its smallest output entry is 5.42×10^{-16} .

That the guarantee is dimension-independent is an argument and not a measurement, so it is checked as well. Table 15 builds the structured block and its unstructured counterpart on a 128×128 and on a $32 \times 32 \times 32$ grid, with the factorized propagators, the direction-split clipped haptotaxis and the same convolutional backbone in two and in three dimensions, and rolls each of them over ten macro steps in single precision and after eight, six and four bit quantization. The weights are drawn at random and are not trained, which is the appropriate test for a statement quantified over every weight vector, and eight draws are used at two scales of the output layer. The structured block attains a nonnegative state in all 8 draws, in both dimensions, at both scales and in all four formats, without a clip ever being needed. The unstructured block, with the same weights, leaves the orthant in six draws of eight in two dimensions in single precision at the smaller scale and in seven of eight at the larger, and in eight of eight at four bits. What is not claimed is accuracy: no two-dimensional or three-dimensional surrogate is trained here, the full-order data for such a study is not generated, and the extension of the accuracy results of Section 4 to higher dimensions remains open. The guarantee, the cost of the transport block and the storage are what transfer, and they are what Tables 14 and 15 report.

Table 15. Random weight vectors that leave the nonnegative orthant, out of eight draws, on a two-dimensional and a three-dimensional grid.

Model	fp32	int8	int6	int4
two dimensions				
structured, gain 1	0/8	0/8	0/8	0/8
structured, gain 5	0/8	0/8	0/8	0/8
unstructured, gain 1	6/8	2/8	1/8	0/8
unstructured, gain 5	7/8	7/8	6/8	8/8
three dimensions				
structured, gain 1	0/8	0/8	0/8	0/8
structured, gain 5	0/8	0/8	0/8	0/8
unstructured, gain 1	5/8	2/8	0/8	3/8
unstructured, gain 5	5/8	4/8	3/8	7/8

6. Conclusions

We have built a macro-step neural surrogate of a six-field reaction, diffusion and taxis model of angiogenesis coupled to a Dynamic Energy Budget reserve, whose invariant is a property of the map, not of the trained weights. Three ingredients carry it, an exponential propagator assembled as a dense nonnegative row-stochastic matrix, an upwind haptotaxis step made independent of the data by substepping and by clipping its diagonal, and a learned block returning a production and a per capita destruction that enter a Modified–Patankar quotient. Every operation is a sum or a product of nonnegative numbers, or a quotient of a nonnegative number by a positive one, so the state stays nonnegative in any arithmetic whose rounding preserves the sign of nonnegative reals. The measurements confirm Theorem 1 from single precision down to four integer bits, in all five seeds and in every regime, without clamping or retraining, while the counterparts of equal size lose the property under quantization.

Two results are worth isolating. The constraint is free in accuracy, since the four models agree to within ten per cent in single precision. The advantage in cost is bounded, since the reference scheme run at the same macro step is less accurate at comparable cost and overtakes the surrogate once high accuracy is required.

The limits are of four kinds and two of them are narrower than they look. The accuracy study is one-dimensional; the guarantee and the cost of the transport block do extend to two and to three dimensions, which Section 5.3 argues and measures, but no higher-dimensional surrogate is trained here and nothing is claimed about the accuracy of one. The assembled propagator costs $\mathcal{O}(M^{2d})$ in storage and in operations and is the part of the construction that does not scale; the tensor-product factorization removes that wall at dM^2 storage and dM^{d+1} operations while keeping the hypothesis the theorem needs, and it is what a deployment on a finer grid should use. The integer arithmetic is emulated by a symmetric quantizer evaluated in single precision, so the timings of Section 5.1 are floating-point timings and the integer rows of Table 5 measure accuracy and admissibility, not speed; the argument of Theorem 1 does not depend on the distinction, since a genuine integer engine is sign-faithful as well, but the cost of an integer deployment is not measured here. And the guarantee concerns the sign and not the balance laws, the transport block being exactly conservative and the learned block not.

Three continuations follow from those limits. A conservative parameterization of the learned coefficients, in the manner of the conservative Patankar schemes, would make a linear invariant exact by construction and close the gap that the budget column of Table 5 measures. Training in the deployed format is the standard remedy for the accuracy that six and four bits cost and would say whether the range of usable formats widens. And

the construction applies unchanged to any production and destruction system with a Metzler transport operator, of which the chemotaxis and reaction-diffusion families are the immediate testbeds.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

CFL	Courant–Friedrichs–Lewy
DCT	Discrete cosine transform
DEB	Dynamic Energy Budget
GEMM	General matrix multiplication
IEEE	Institute of Electrical and Electronics Engineers
MAC	Multiply and accumulate operation
PDE	Partial differential equation
PINN	Physics-informed neural network

References

1. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [\[CrossRef\]](#)
2. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229. [\[CrossRef\]](#)
3. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Fourier neural operator for parametric partial differential equations. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 3–7 May 2021.
4. Bar-Sinai, Y.; Hoyer, S.; Hickey, J.; Brenner, M.P. Learning data-driven discretizations for partial differential equations. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 15344–15349. [\[CrossRef\]](#)
5. Kochkov, D.; Smith, J.A.; Alieva, A.; Wang, Q.; Brenner, M.P.; Hoyer, S. Machine learning-accelerated computational fluid dynamics. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2101784118. [\[CrossRef\]](#)
6. Um, K.; Brand, R.; Fei, Y.; Holl, P.; Thuerey, N. Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. In Proceedings of the Advances in Neural Information Processing Systems 33, Vancouver, BC, Canada, 6–12 December 2020; pp. 6111–6122.
7. Brandstetter, J.; Worrall, D.; Welling, M. Message passing neural PDE solvers. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
8. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
9. Burchard, H.; Deleersnijder, E.; Meister, A. A high-order conservative Patankar-type discretisation for stiff systems of production–destruction equations. *Appl. Numer. Math.* **2003**, *47*, 1–30. [\[CrossRef\]](#)
10. Kopecz, S.; Meister, A. On order conditions for modified Patankar–Runge–Kutta schemes. *Appl. Numer. Math.* **2018**, *123*, 159–179. [\[CrossRef\]](#)
11. Kopecz, S.; Meister, A. Unconditionally positive and conservative third order modified Patankar–Runge–Kutta discretizations of production–destruction systems. *BIT Numer. Math.* **2018**, *58*, 691–728. [\[CrossRef\]](#)
12. Öffner, P.; Torlo, D. Arbitrary high-order, conservative and positivity preserving Patankar-type deferred correction schemes. *Appl. Numer. Math.* **2020**, *153*, 15–34. [\[CrossRef\]](#)
13. Izgin, T.; Kopecz, S.; Meister, A. On Lyapunov stability of positive and conservative time integrators and application to second order modified Patankar–Runge–Kutta schemes. *ESAIM Math. Model. Numer. Anal.* **2022**, *56*, 1053–1080. [\[CrossRef\]](#)
14. Sandu, A. Positive numerical integration methods for chemical kinetic systems. *J. Comput. Phys.* **2001**, *170*, 589–602. [\[CrossRef\]](#)

15. Moler, C.; Van Loan, C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.* **2003**, *45*, 3–49. [\[CrossRef\]](#)
16. Al-Mohy, A.H.; Higham, N.J. A new scaling and squaring algorithm for the matrix exponential. *SIAM J. Matrix Anal. Appl.* **2010**, *31*, 970–989. [\[CrossRef\]](#)
17. Hochbruck, M.; Ostermann, A. Exponential integrators. *Acta Numer.* **2010**, *19*, 209–286. [\[CrossRef\]](#)
18. Beucler, T.; Pritchard, M.; Rasp, S.; Ott, J.; Baldi, P.; Gentine, P. Enforcing analytic constraints in neural networks emulating physical systems. *Phys. Rev. Lett.* **2021**, *126*, 098302. [\[CrossRef\]](#)
19. Richter-Powell, J.; Lipman, Y.; Chen, R.T.Q. Neural conservation laws: A divergence-free perspective. In Proceedings of the Advances in Neural Information Processing Systems 35, New Orleans, LA, USA, 28 November–9 December 2022; pp. 38075–38088.
20. Greydanus, S.; Dzamba, M.; Yosinski, J. Hamiltonian neural networks. In Proceedings of the Advances in Neural Information Processing Systems 32, Vancouver, BC, Canada, 8–14 December 2019; pp. 15379–15389.
21. Chen, R.T.Q.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. Neural ordinary differential equations. In Proceedings of the Advances in Neural Information Processing Systems 31, Montreal, QC, Canada, 3–8 December 2018; pp. 6571–6583.
22. Hairer, E.; Lubich, C.; Wanner, G. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed.; Springer: Berlin, Germany, 2006.
23. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 2704–2713.
24. Nagel, M.; Amjad, R.A.; van Baalen, M.; Louizos, C.; Blankevoort, T. Up or down? Adaptive rounding for post-training quantization. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 13–18 July 2020; pp. 7197–7206.
25. Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M.W.; Keutzer, K. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2022; pp. 291–326.
26. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
27. Sze, V.; Chen, Y.-H.; Yang, T.-J.; Emer, J.S. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* **2017**, *105*, 2295–2329. [\[CrossRef\]](#)
28. Deng, L.; Li, G.; Han, S.; Shi, L.; Xie, Y. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proc. IEEE* **2020**, *108*, 485–532. [\[CrossRef\]](#)
29. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
30. Banbury, C.; Reddi, V.J.; Torelli, P.; Holleman, J.; Jeffries, N.; Kiraly, C.; Montino, P.; Kanter, D.; Ahmed, S.; Pau, D.; et al. MLPerf Tiny benchmark. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, Virtual, 6–14 December 2021.
31. David, R.; Duke, J.; Jain, A.; Janapa Reddi, V.; Jeffries, N.; Li, J.; Kreeger, N.; Nappier, I.; Natraj, M.; Wang, T.; et al. TensorFlow Lite Micro: Embedded machine learning for TinyML systems. *Proc. Mach. Learn. Syst.* **2021**, *3*, 800–811.
32. Higham, N.J. *Accuracy and Stability of Numerical Algorithms*, 2nd ed.; SIAM: Philadelphia, PA, USA, 2002.
33. Higham, N.J.; Mary, T. Mixed precision algorithms in numerical linear algebra. *Acta Numer.* **2022**, *31*, 347–414. [\[CrossRef\]](#)
34. Higham, N.J.; Pranesh, S. Simulating low precision floating-point arithmetic. *SIAM J. Sci. Comput.* **2019**, *41*, C585–C602. [\[CrossRef\]](#)
35. Abdelfattah, A.; Anzt, H.; Boman, E.G.; Carson, E.; Cojean, T.; Dongarra, J.; Fox, A.; Gates, M.; Higham, N.J.; Li, X.S.; et al. A survey of numerical linear algebra methods utilizing mixed-precision arithmetic. *Int. J. High Perform. Comput. Appl.* **2021**, *35*, 344–369. [\[CrossRef\]](#)
36. Broekhuizen, N.; Rickard, G.J.; Bruggeman, J.; Meister, A. An improved and generalized second order, unconditionally positive, mass conserving integration scheme for biochemical systems. *Appl. Numer. Math.* **2008**, *58*, 319–340. [\[CrossRef\]](#)
37. Huang, J.; Shu, C.-W. Positivity-preserving time discretizations for production–destruction equations with applications to non-equilibrium flows. *J. Sci. Comput.* **2019**, *78*, 1811–1839. [\[CrossRef\]](#)
38. Ávila, A.I.; Kopecz, S.; Meister, A. A comprehensive theory on generalized BBKS schemes. *Appl. Numer. Math.* **2021**, *157*, 19–37. [\[CrossRef\]](#)
39. Blanes, S.; Iserles, A.; Macnamara, S. Positivity-preserving methods for ordinary differential equations. *ESAIM Math. Model. Numer. Anal.* **2022**, *56*, 1843–1870. [\[CrossRef\]](#)
40. Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv* **2018**, arXiv:1806.08342.
41. Choi, J.; Wang, Z.; Venkataramani, S.; Chuang, P.I.-J.; Srinivasan, V.; Gopalakrishnan, K. PACT: Parameterized clipping activation for quantized neural networks. *arXiv* **2018**, arXiv:1805.06085.

42. Esser, S.K.; McKinstry, J.L.; Bablani, D.; Appuswamy, R.; Modha, D.S. Learned step size quantization. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
43. Klöwer, M.; Düben, P.D.; Palmer, T.N. Number formats, error mitigation, and scope for 16-bit arithmetics in weather and climate modeling. *J. Adv. Model. Earth Syst.* **2020**, *12*, e2020MS002246. [\[CrossRef\]](#)
44. Klöwer, M.; Hatfield, S.; Croci, M.; Düben, P.D.; Palmer, T.N. Fluid simulations accelerated with 16 bits: Approaching 4x speedup on A64FX by squeezing ShallowWaters.jl into Float16. *J. Adv. Model. Earth Syst.* **2022**, *14*, e2021MS002684. [\[CrossRef\]](#)
45. Roy, K.; Jaiswal, A.; Panda, P. Towards spike-based machine intelligence with neuromorphic computing. *Nature* **2019**, *575*, 607–617. [\[CrossRef\]](#)
46. Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99. [\[CrossRef\]](#)
47. Micikevicius, P.; Narang, S.; Alben, J.; Diamos, G.; Elsen, E.; Garcia, D.; Ginsburg, B.; Houston, M.; Kuchaiev, O.; Venkatesh, G.; et al. Mixed precision training. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
48. Anderson, A.R.A.; Chaplain, M.A.J. Continuous and discrete mathematical models of tumour-induced angiogenesis. *Bull. Math. Biol.* **1998**, *60*, 857–899. [\[CrossRef\]](#)
49. De Luca, P.; Marcellino, L. Analytical and numerical properties of an extended angiogenesis PDEs model. *J. Math. Biol.* **2025**, *91*, 62. [\[CrossRef\]](#)
50. Levine, H.A.; Pamuk, S.; Sleeman, B.D.; Nilsen-Hamilton, M. Mathematical modeling of capillary formation and development in tumor angiogenesis: Penetration into the stroma. *Bull. Math. Biol.* **2001**, *63*, 801–863. [\[CrossRef\]](#)
51. Kooijman, S.A.L.M. *Dynamic Energy Budget Theory for Metabolic Organisation*, 3rd ed.; Cambridge University Press: Cambridge, UK, 2010.
52. Nisbet, R.M.; Muller, E.B.; Lika, K.; Kooijman, S.A.L.M. From molecules to ecosystems through dynamic energy budget models. *J. Anim. Ecol.* **2000**, *69*, 913–926. [\[CrossRef\]](#)
53. Berman, A.; Plemmons, R.J. *Nonnegative Matrices in the Mathematical Sciences*; SIAM: Philadelphia, PA, USA, 1994.
54. De Luca, P.; Marcellino, L. Towards numerical method-informed neural networks for PDE learning. *Mathematics* **2025**, *13*, 2392. [\[CrossRef\]](#)
55. Folkman, J. Tumor angiogenesis: Therapeutic implications. *N. Engl. J. Med.* **1971**, *285*, 1182–1186. [\[CrossRef\]](#)
56. Carmeliet, P.; Jain, R.K. Molecular mechanisms and clinical applications of angiogenesis. *Nature* **2011**, *473*, 298–307. [\[CrossRef\]](#)
57. Potente, M.; Gerhardt, H.; Carmeliet, P. Basic and therapeutic aspects of angiogenesis. *Cell* **2011**, *146*, 873–887. [\[CrossRef\]](#)
58. Bowersox, J.C.; Sorgente, N. Chemotaxis of aortic endothelial cells in response to fibronectin. *Cancer Res.* **1982**, *42*, 2547–2551.
59. Ausprunk, D.H.; Folkman, J. Migration and proliferation of endothelial cells in preformed and newly formed blood vessels during tumor angiogenesis. *Microvasc. Res.* **1977**, *14*, 53–65. [\[CrossRef\]](#)
60. Stokes, C.L.; Lauffenburger, D.A. Analysis of the roles of microvessel endothelial cell random motility and chemotaxis in microvascular network formation. *J. Theor. Biol.* **1991**, *152*, 377–403.
61. Rundhaug, J.E. Matrix metalloproteinases and angiogenesis. *J. Cell. Mol. Med.* **2005**, *9*, 267–285. [\[CrossRef\]](#)
62. De Bock, K.; Georgiadou, M.; Schoors, S.; Kuchnio, A.; Wong, B.W.; Cantelmo, A.R.; Quaegebeur, A.; Ghesquière, B.; Cauwenberghs, S.; Eelen, G.; et al. Role of PFKFB3-driven glycolysis in vessel sprouting. *Cell* **2013**, *154*, 651–663. [\[CrossRef\]](#)
63. Strang, G. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.* **1968**, *5*, 506–517. [\[CrossRef\]](#)
64. Bolley, C.; Crouzeix, M. Conservation de la positivité lors de la discrétisation des problèmes d'évolution paraboliques. *RAIRO Anal. Numér.* **1978**, *12*, 237–245. [\[CrossRef\]](#)
65. Zhang, X.; Shu, C.-W. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.* **2010**, *229*, 3091–3120. [\[CrossRef\]](#)
66. IEEE Std 754-2019; IEEE Standard for Floating-Point Arithmetic. IEEE: Piscataway, NJ, USA, 2019.
67. Van Loan, C.F. The ubiquitous Kronecker product. *J. Comput. Appl. Math.* **2000**, *123*, 85–100. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.