



Interpreting neural operators: how nonlinear waves propagate in non-reciprocal solids

Jonathan Colen, Alexis Poncet, Denis Bartolo, Vincenzo Vitelli

► To cite this version:

Jonathan Colen, Alexis Poncet, Denis Bartolo, Vincenzo Vitelli. Interpreting neural operators: how nonlinear waves propagate in non-reciprocal solids. *Physical Review Letters*, 2024, 133 (10), pp.107301. <10.1103/PhysRevLett.133.107301>. <hal-04743537>

HAL Id: hal-04743537

<https://hal.science/hal-04743537v1>

Submitted on 18 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Interpreting neural operators: how nonlinear waves propagate in non-reciprocal solids

Jonathan Colen,^{1,2,3,*} Alexis Poncet,^{4,*} Denis Bartolo,^{4,†} and Vincenzo Vitelli^{1,2,‡}

¹*James Franck Institute, University of Chicago, Chicago, IL 60637*

²*Department of Physics, University of Chicago, Chicago, IL 60637*

³*Joint Institute on Advanced Computing for Environmental Studies,
Old Dominion University, Norfolk, Virginia 23539*

⁴*Univ. Lyon, ENS de Lyon, Univ. Claude Bernard,
CNRS, Laboratoire de Physique, F-69342, Lyon.*

(Dated: April 22, 2024)

We present a data-driven pipeline for model building that combines interpretable machine learning, hydrodynamic theories, and microscopic models. The goal is to uncover the underlying processes governing nonlinear dynamics experiments. We exemplify our method with data from microfluidic experiments where crystals of streaming droplets support the propagation of nonlinear waves absent in passive crystals. By combining physics-inspired neural networks, known as neural operators, with symbolic regression tools, we generate the solution, as well as the mathematical form, of a nonlinear dynamical system that accurately models the experimental data. Finally, we interpret this continuum model from fundamental physics principles. Informed by machine learning, we coarse grain a microscopic model of interacting droplets and discover that non-reciprocal hydrodynamic interactions stabilise and promote nonlinear wave propagation.

Introduction. — The goal of physical modeling is to build a set of rules which can predict and explain behaviors observed in experiments. In continuum theories, these rules are typically grounded in symmetries and conservation laws, which determine the relevant degrees of freedom and the equations they must satisfy. This approach faces challenges when applied to experiments that only reveal the dynamics of a finite subsets of all the interacting degrees of freedom. This situation is not exceptional, in particular it is the norm when soft matter is driven out-of-equilibrium [1, 2]. In these systems, the particles, such as colloids, droplets, or bubbles continuously exchange energy and momentum with a solvent. As a result they exhibit a variety of behaviors, driven by effective interactions that violate microscopic reversibility constraints such as detailed balance [3, 4] or Newton’s third law [5–15]. Such complications require new approaches to model building that can identify and predict the effects of these microscopic interactions in experiments [16–19].

Data-driven methods, which have shown much promise in learning physical models from observation data, have the potential to overcome these challenges. Deep neural networks [20, 21] can learn to accurately predict behavior for a variety of physical [22–30] and biological [31–35] systems. More recently, neural operators have been introduced to directly learn the solution operator for partial differential equations [36]. When incorporated within deep neural networks, they are capable of forecasting the behavior of complex dynamical systems such as the weather [36, 37]. While these methods are invaluable for efficiently and accurately predicting dynamics from data, less is known about how to interpret what they have learned about a physical system.

In this case study, we demonstrate how interpreting

neural operators can lead to data-driven model discovery by focusing on a paradigmatic out-of-equilibrium hydrodynamic problem: particles driven in a surrounding fluid [38]. Such systems exhibit a variety of phenomena: phonons in overdamped systems [39, 40], shock formation [41, 42], and symmetry-dependent melting of crystal phases [43]. We build on an experimental set-up in which water droplets advected inside a microfluidic channel form a propagating shock due to hydrodynamic interactions. Using a neural network with an internal neural operator layer [36], we transform the experimental movie into a set of variables whose dynamics are solved by an unknown partial differential equation. Using sparse regression [44–46], we then discover the equation that these machine-learned variables satisfy. We find that the discovered continuum model contains an essential term that had remained overlooked. Informed by our data-driven method, we then trace this term to the non-reciprocal nature of hydrodynamic interactions by solving suitable microscopic models. Beyond the specifics of our microfluidic experiments, we show how microscopic non-reciprocity generically results in the propagation of macroscopic solitons through non-equilibrium crystals. Our approach demonstrates how combining interpretable neural operators with theoretical modeling can provide previously-unknown insights into the physics governing complex non-linear systems.

Learning experimental dynamics. — To learn macroscopic physics from microscopic data, we consider an experimental movie [42] in which a one dimensional stream of water droplets (diameter $140\,\mu\text{m}$) are advected by an organic solvent inside a microfluidic channel, see [42] for details. The droplet stream develops a shock in response to a jam induced at $t = 0$ (Fig. 1a).

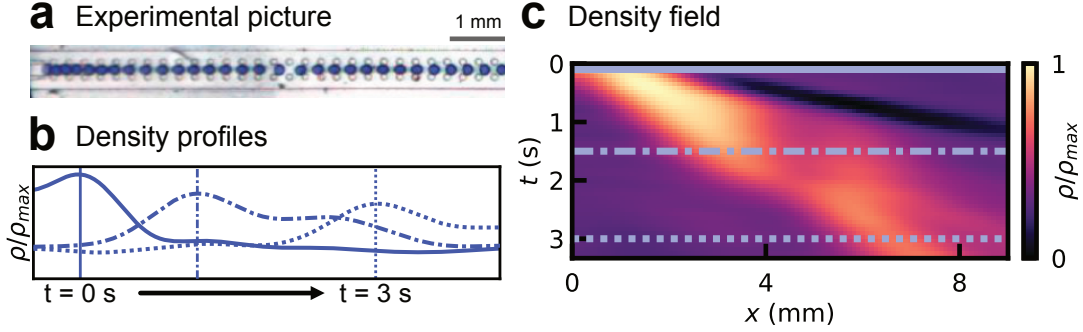


FIG. 1. **Coarse-graining microfluidic particle experiments.** (a) Experimental picture of a microfluidic particle stream [42]. (b) Experimental pictures are coarse-grained into a density profile and normalized by the maximum density ρ_{max} . (c) Spatio-temporal evolution of the density field shows formation of a shock front in response to an initial jam.

Coarse-graining the droplet positions yields a density field $\rho(x, t)$ which shows the shock deforming and propagating through space and time (Fig. 1). The system behavior is driven by microscopic interactions between the droplets, which may include a simple nearest-neighbor repulsive interaction as well as hydrodynamic backflows. Previous studies modeled this experiment at the continuum level to leading order in derivatives and nonlinearities. The resulting model reduced to a Burgers' equation obeyed by the droplet density $\rho(x, t)$ [42]

$$\partial_t \rho + (c - \alpha \rho) \partial_x \rho = 0, \quad (1)$$

where c is a basic advection speed, and α is a non-linearity that creates shocks. Here, without making any assumption about the relevant interactions, we learn to predict the dynamics of this movie using a deep neural network. To train the network, we segmented the experimental movie into overlapping $3\text{ s} \times 9\text{ mm}$ training examples. For each example, the network receives the one-dimensional slice $\rho(x) = \rho_0(x)$ and predicts the full $\rho(x, t)$ response to that initial condition (Fig. 2a). The network architecture has three components: (i) an input convolutional block which transforms the experimental density field $\rho_0(x)$ into a latent variable $\phi_0(x)$, (ii) a single-layer Fourier neural operator which evolves ϕ in time according to an unknown *linear* partial differential equation, and (iii) an output convolutional block which transforms the predicted $\phi(x, t)$ field into a predicted density field $\rho(x, t)$. Both the input and output blocks used 1-D convolutional layers which only use spatial information to construct features. The full architecture is shown in Fig. 2a. We note that the dynamics were solely predicted by the central neural operator layer. However, while the neural operator itself is linear, the input and output transformations enable the network to predict nonlinear dynamics [47]. Once trained, the neural network is capable of predicting experimental dynamics including the propagation of the initial jam in space and time from past to present (Fig. 2a-c).

Interpreting the neural operator. — The single-layer linear Fourier neural operator can be interpreted as a Green's function for the internal variable ϕ . This learned Green's function predicts how the response to an initial perturbation will propagate in space over time (Fig. 2a-b). In the co-moving frame, defined by the spatial coordinate $x - ct$ where c is the speed of the wave, we observe that the response decreases in magnitude while the width stays nearly constant (Fig. 2c-e). The encoder/decoder architecture transforms the nonlinear dynamics of the density ρ into a linear problem that can be solved using this Green's function [47].

A similar approach can exactly solve the Burgers' equation (1) previously proposed for this system. Upon adding a small diffusion term $D \partial_x^2 \rho$ and effecting a Cole-Hopf transformation, we obtain a linear partial differential equation whose Green's function is that of the standard diffusion equation (see SI). In Figure 2f-h, we plot the dynamics predicted by Burgers' equation from experimental initial conditions. As a benchmark for our machine-learned solution, we trained an identical neural network on a synthetic movie of an exact solution to Burgers' equation (SI Fig. S3). While this model learned an internal variable transformation that was distinct from Cole-Hopf, the Green's function is qualitatively similar to the diffusion kernel with a response whose amplitude decays and width grows over time.

The neural network (Figure 2c-e) more accurately predicts the experimental movie than an exact solution to Burgers' equation (Figure 2f-h) and uses a qualitatively distinct Green's function to do so. We stress that while both Green's functions decay in amplitude over time, the diffusion kernel has a response width that grows over time while the neural operator solution remains constant (Fig. 2e,h). As a result, the neural network shock remains sharp unlike the Burgers' shock that disperses (Fig. 2a,c,f). The distinct behavior of the neural operator trained on experiments from both the exact and benchmark solutions to Burgers' equation suggests that

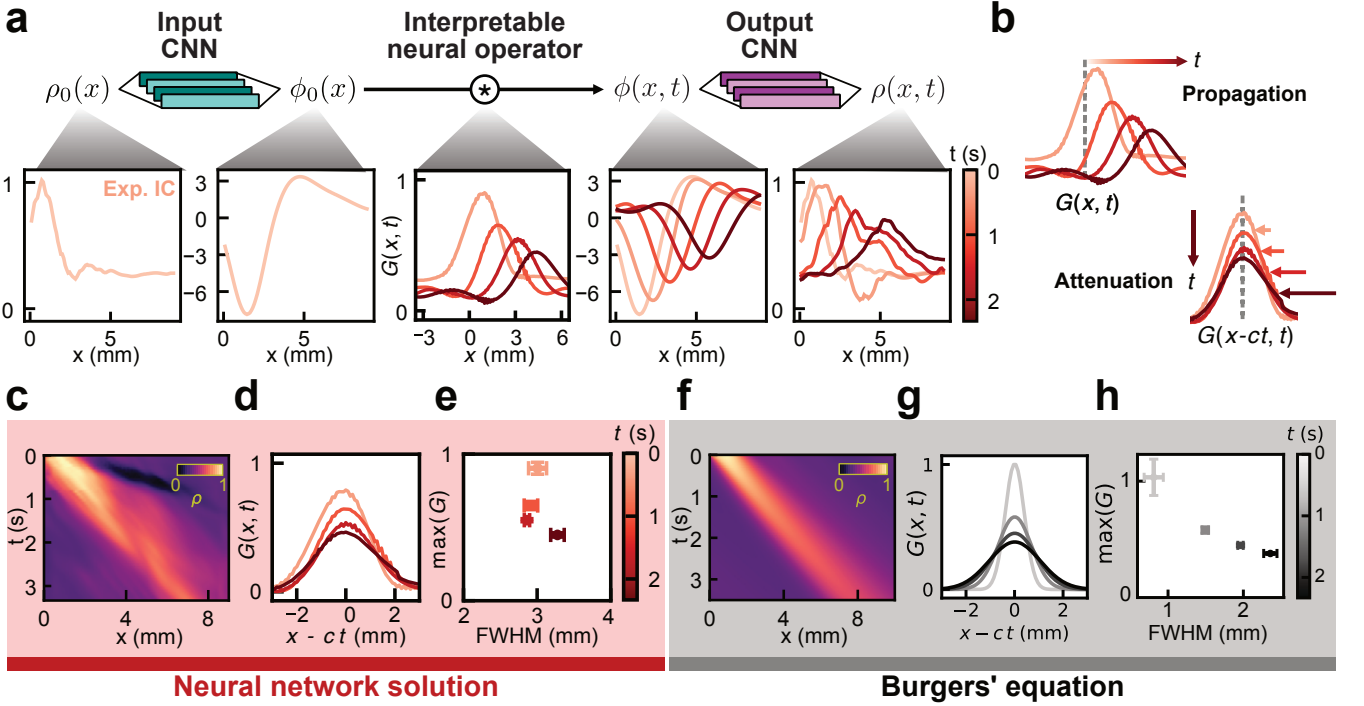


FIG. 2. Learning the dynamics of shocks in microfluidic particles experiments. (a) A neural network forecasts the particle density field $\rho(x)$ from initial conditions. The network uses a 1-D convolutional block to map the density field to an internal variable $\phi(x)$. A single-layer linear Fourier neural operator forecasts the evolution of ϕ . A second 1-D convolutional block converts $\phi(x, t)$ into a predicted density field $\rho(x, t)$. (b) The neural operator is interpretable as a Green's function for an unknown linear PDE which defines the response to an perturbation. The learned Green's function predicts a response which propagates in space and attenuates over time. (c) Neural network predictions for the density field $\rho(x, t)$. (d) Plot of the machine-learned Green's function in the co-moving frame. (e) Comparing the magnitude and width of the kernel $G(x, t)$ shows that the response to a perturbation will decrease in magnitude over time, but not disperse. (f-h) Exact solution to Burgers' equation from experimental initial conditions. $G(x, t)$ is a diffusion kernel applied after a Cole-Hopf transformation of the density ρ . The neural network learns distinct physics from Burgers', which exhibits spreading due to viscous dissipation.

the non-linear waves seen in experiments obey physics beyond Burgers' equation.

To learn an equation governing the internal variable $\phi(x, t)$ (Fig. 3a), we resort to the SINDy method [44], which has proven to effectively learn non-linear dynamical equations from experimental data [19, 28]. As the linear neural operator ensures ϕ obeys a linear PDE, we used a library of ϕ and its derivatives and found the following equation

$$\partial_t \phi - 1.86 \partial_x \phi = -0.31 \phi + 0.003 \partial_x^2 \phi - 0.008 \partial_x^3 \phi \quad (2)$$

Simulating this equation for $\phi(x, t)$ yields excellent correspondence with the neural network solution (Fig. 3a, $R^2 = 0.96$), indicating that it is indeed an excellent model for the dynamics solved by the neural operator. We applied the neural network's output CNN to this predicted ϕ field to achieve an accurate prediction of the density dynamics (SI Fig. S2).

The learned equation (2) resembles a linearized KdV-Burgers' equation, containing a dispersive $\partial_x^3 \phi$ term characteristic of the Korteweg de-Vries equation. The $\partial_x \phi$

term captures both the basic advection term and a linearization of the $\rho \partial_x \rho$ term in (1). We do note that the model has learned a damping term whose importance to the learned dynamics is weaker than the dispersive $\partial_x^3 \phi$ term but stronger than the diffusive $\partial_x^2 \phi$ term (SI Table S1). The new KdV term in the dynamics of ϕ , as well as the qualitatively distinct Green's function learned by the neural operator, corroborates our conclusion that the dynamics of the experimental density field is governed by physics beyond Burgers' equation. Hence, we consider the more general nonlinear KdV-Burgers' equation:

$$\partial_t \rho + (c - \alpha \rho) \partial_x \rho = D \partial_x^2 \rho + \beta \partial_x^3 \rho \quad (3)$$

We used the experimental movie to directly fit coefficients to a Burgers' equation ($\beta = 0$), a KdV equation ($D = 0$), and a KdV-Burgers' equation (3). We plot the predictions and compare accuracy in Figure 3b-c. Both Burgers' and KdV-Burgers' achieve similar error rates on the entire field, although neither matches the performance of the neural network. The difference between the two is more apparent near the shock front. The KdV-Burgers'

solution maintains a more pronounced shock over time (Fig. 3b) and achieves a lower error rate than Burgers' in the vicinity of this shock front (Fig. 3c).

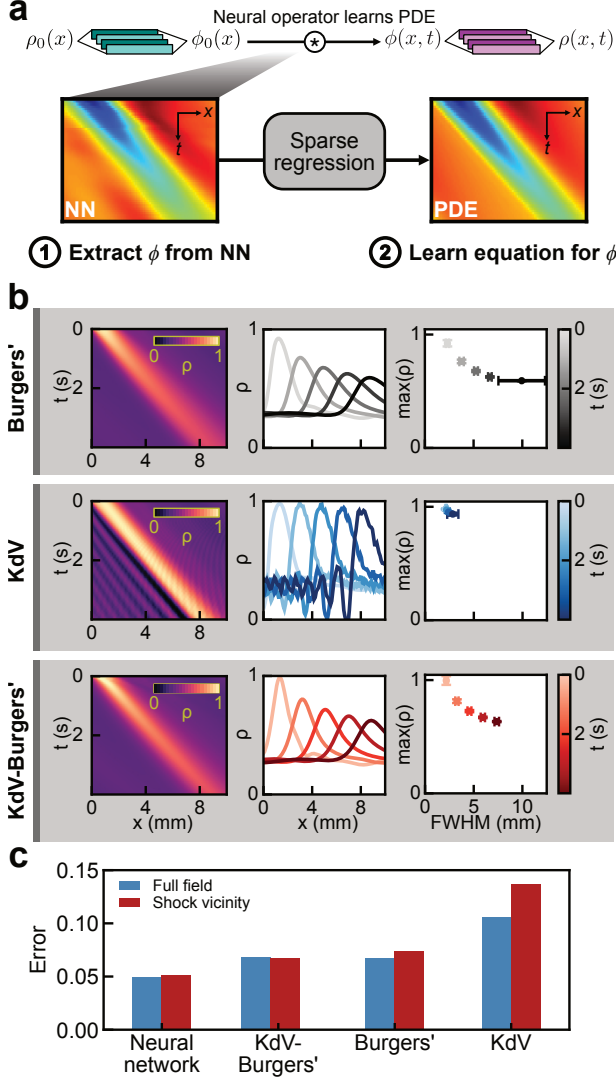


FIG. 3. Interpreting the neural operator. (a) The neural operator evolves the internal variable $\phi(x, t)$ according to a partial differential equation. Using SINDy, we learn the PDE governing $\phi(x, t)$. The resulting equation (2) includes a novel ∂_x^3 KdV term [42]. (b) Simulations of Burgers' equation (first row), KdV equation (second row), and KdV-Burgers' equation (third row) from experimental initial conditions. For each equation, we plot the density field (left) and traces at fixed time points (middle). To characterize the shock front, we plot the height vs. width at different time points (right). KdV-Burgers' preserves the sharpness of the shock and shows less spreading over time compared to Burgers'. (c) Error rates for neural network (Fig. 2) and analytic solutions (Fig. 3). Blue bars are the mean absolute error $\langle |\rho - \rho_0| \rangle$ for the full field, while red is the error within 1 mm of the shock front. Numerical values in SI Table S2.

Interpretation of learned non-linear dynamics from microscopics. — The neural operator discovered a KdV-Burgers dynamics but provided no first-principles interpretation for it. We now seek to understand its physical origin. To make progress, we build a minimal model where the dynamics of the droplets is determined by the competition between contact and hydrodynamic interactions. The equations of motion for the droplet positions R_n are

$$\zeta(\partial_t R_n - v_0) = \sum_{m \neq n} (f_{m \rightarrow n}^H + f_{m \rightarrow n}^E), \quad (4)$$

where ζ a friction coefficient and v_0 the advection speed of an isolated droplet. f^E is an elastic force that models the repulsion between the soft droplets which we approximate by linear springs $f_{n+1 \rightarrow n}^E = k(R_{n+1} - R_n - a)$ (a is the lattice spacing and k the elastic stiffness). f^H models the hydrodynamic force experienced by a droplet in response to the backflows induced by the motion of its neighbors, see Fig. 4a. Unlike f^E , hydrodynamic forces do not obey Newton's third law [6]. They are non-reciprocal: $f_{m \rightarrow n}^H = f_{n \rightarrow m}^H = f_H(R_n - R_m)$. In practice, we use the standard form $f_H(r) = A \text{csch}^2(\pi r/W)$, where W is the channel width [39].

We perform numerical simulations of Eq. 4 with periodic boundary conditions, $\zeta = a = 1$, $W = 2$, and $A = -(\pi/W)^2$. To probe the relevance of KdV-Burgers physics, we choose initial conditions that correspond to a 2-soliton solution of the KdV equation (Fig. 4b-e, see also SI for different initial conditions). When ignoring the soft contact interactions ($k = 0$), the particle dynamics is perfectly captured by a continuum model having the form of a KdV equation. Two solitons cross each other and propagate freely (Fig. 4c). In the limit of strong elastic forces ($k = 0.2$), the initial shape is not preserved, shocks develop and their dynamics is well described by a Burgers equation (Fig. 4e). In the intermediate regime, where both contact and hydrodynamic forces compete on equal footing ($k = 0.02$), we find that the propagation of the initial density fluctuation is quantitatively predicted by the KdV-Burgers equation Eq. (3), (Fig. 4d).

We are now equipped to interpret the KdV-Burgers dynamics revealed by our neural operator. We focus on large distances ($\tilde{x} = \epsilon(n - ct)$ with $\epsilon \rightarrow 0$) and large times ($\tilde{t} = \epsilon^3 t$), we then can recast Eq. (4) into a PDE obeyed by the density field $\rho(x, t)$. Informed by our Neural operator, we do not restrain our expansion to leading order in gradients and find that the dynamics of $\rho(x, t)$ is ruled by a KdV-Burgers equation (3). We provide a detailed derivation of our continuum model in SI and generalize it beyond the specifics of hydrodynamic interactions.

This procedure makes it possible to understand the physical origin of all terms in Eq. (3). The damping (D) is solely controlled by the elastic interactions f^E . Conversely, the advection, the non-linear and the

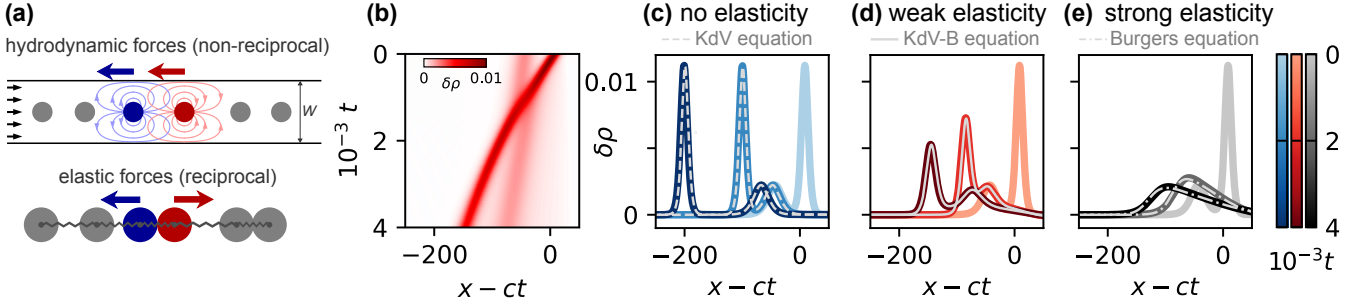


FIG. 4. Microscopic modeling of the experiments and confirmation of the KdV-Burger behavior. (a) We model the droplets as an overdamped chain of beads that interact by two kinds of interactions: hydrodynamic forces which are non-reciprocal ($f_{m \rightarrow n}^H = f_{n \rightarrow m}^H$) and elastic forces which correspond to usual springs ($f_{m \rightarrow n}^E = -f_{n \rightarrow m}^E$). (b) We run numerical simulations starting from small density fluctuations $\delta\rho = \rho - a^{-1}$ corresponding to a two-soliton solution of the KdV equation. At weak elasticity ($k = 0.02$) the solitons propagate by crossing one-another, and are damped by the elasticity. (c-e) The colored lines show the time-evolution in the simulations at no elasticity ($k = 0$), weak elasticity ($k = 0.02$, corresponding to panel b) and strong elasticity ($k = 0.2$). In the three cases, this evolution is in excellent agreement with the numerical integration of the KdV-Burgers equation (shown in panel (d) with thin gray lines). Dashed lines in panels (c) and (e) are respectively the analytical solutions of the KdV equation and the Burgers equation, showing the two limit cases.

dispersive terms (c , α and β) originate from the hydrodynamic forces ($c = v_0 - 2 \sum_{m=1}^{\infty} m f_H'(ma)$, $\alpha = -2 \sum_{m=1}^{\infty} m^2 f_H''(ma)$, and $\beta = \sum_{m=1}^{\infty} m f_H'(ma)/3$). Unlike in [48], the dispersive and non-linear terms rooted in hydrodynamic interactions conspire to stabilize and propel solitary waves without relying on inertia.

Our first principles modeling and machine learning algorithms play a mutually beneficial role. While the neural operator can suggest the presence of a KdV term, it could not provide a mechanism justifying its existence. Our simulations and theoretical argument help rationalizing these findings which had been hitherto overlooked. They provide insight on the origin of the KdV-like physics: beyond the specifics of our driven emulsions, non-reciprocal interactions generically stabilize and propagate solitary waves in overdamped one-dimensional systems. Neural networks' role as universal approximators provides an alternative route to characterizing experiments. In learning to predict dynamics, they build a “maximal model” which can then serve as a target for more principled theoretical investigations.

This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 101019141) (D.B. and A.P.).

* These two authors contributed equally

† denis.bartolo@ens-lyon.fr

‡ vitelli@uchicago.edu

- [1] S. Ramaswamy, *Advances in Physics* **50**, 297 (2001).
- [2] M. C. Marchetti, J. F. Joanny, S. Ramaswamy, T. B. Liverpool, J. Prost, M. Rao, and R. A. Simha, *Reviews of Modern Physics* **85**, 1143 (2013).
- [3] M. Fruchart, C. Scheibner, and V. Vitelli, *Annual Re-*

view of Condensed Matter Physics **14**, 471 (2023).

- [4] C. R. Scheibner, A. Souslov, D. Banerjee, P. Surowka, W. T. Irvine, and V. Vitelli, *Nature Physics* **16**, 475 (2020).
- [5] A. V. Ivlev, J. Bartnick, M. Heinen, C.-R. Du, V. Nosenko, and H. Löwen, *Phys. Rev. X* **5**, 011035 (2015).
- [6] A. Poncet and D. Bartolo, *Physical Review Letters* **128**, 048002 (2022).
- [7] F. Brauns and M. C. Marchetti, “Non-reciprocal pattern formation of conserved fields,” (2023), [arXiv:2306.08868](https://arxiv.org/abs/2306.08868) [nlin.PS].
- [8] S. Saha and R. Golestanian, “Effervescent waves in a binary mixture with non-reciprocal couplings,” (2022), [arXiv:2208.14985](https://arxiv.org/abs/2208.14985) [cond-mat.stat-mech].
- [9] S. Saha, J. Agudo-Canalejo, and R. Golestanian, *Physical Review X* **10**, 041009 (2020).
- [10] Z. You, A. Baskaran, and M. C. Marchetti, *Proceedings of the National Academy of Sciences* **117**, 19767 (2020).
- [11] R. Mandal, S. S. Jaramillo, and P. Sollich, “Robustness of travelling states in generic non-reciprocal mixtures,” (2022), [arXiv:2212.05618](https://arxiv.org/abs/2212.05618) [cond-mat.stat-mech].
- [12] Y. Avni, M. Fruchart, D. Martin, D. Seara, and V. Vitelli, “The non-reciprocal ising model,” (2023), [arXiv:2311.05471](https://arxiv.org/abs/2311.05471) [cond-mat.stat-mech].
- [13] C. Weis, M. Fruchart, R. Hanai, K. Kawagoe, P. B. Littlewood, and V. Vitelli, “Exceptional points in nonlinear and stochastic dynamics,” (2023), [arXiv:2207.11667](https://arxiv.org/abs/2207.11667) [nlin.CD].
- [14] D. Martin, D. Seara, Y. Avni, M. Fruchart, and V. Vitelli, “The transition to collective motion in nonreciprocal active matter: coarse graining agent-based models into fluctuating hydrodynamics,” (2024), [arXiv:2307.08251](https://arxiv.org/abs/2307.08251) [cond-mat.stat-mech].
- [15] M. Fruchart, R. Hanai, P. B. Littlewood, and V. Vitelli, *Nature* **592**, 363 (2021).
- [16] H. Li, X.-q. Shi, M. Huang, X. Chen, M. Xiao, C. Liu, H. Chaté, and H. P. Zhang, *Proceedings of the National Academy of Sciences* **116**, 777 (2019).
- [17] W. Van Saarloos, V. Vitelli, and Z. Zeravcic, *Soft Matter: Concepts, Phenomena, and Applications* (Princeton

- University Press, 2024).
- [18] D. S. Seara, J. Colen, M. Fruchart, Y. Avni, D. Martin, and V. Vitelli, “Sociohydrodynamics: data-driven modelling of social behavior,” (2024), [arXiv:2312.17627 \[cond-mat.soft\]](#).
 - [19] R. Supek, B. Song, A. Hastewell, G. P. T. Choi, A. Mietke, and J. Dunkel, *Proceedings of the National Academy of Sciences* **120**, e2206994120 (2023).
 - [20] Y. Lecun, Y. Bengio, and G. Hinton, *Nature* **521**, 436 (2015).
 - [21] J. Schmidhuber, *Neural Networks* **61**, 85 (2015).
 - [22] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Journal of Computational Physics* **378**, 686 (2019).
 - [23] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, *Reviews of Modern Physics* **91**, 045002 (2019).
 - [24] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, *Nature Machine Intelligence* **2**, 94 (2020).
 - [25] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, *Nature Reviews Physics* **3**, 422 (2021).
 - [26] J. Carrasquilla and R. G. Melko, *Nature Physics* **13**, 431 (2017).
 - [27] V. Bapst, T. Keck, A. Grabska-Barwińska, C. Donner, E. D. Cubuk, S. Schoenholz, A. Obika, A. Nelson, T. Back, D. Hassabis, and P. Kohli, *Nature Physics* **16**, 448 (2020).
 - [28] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, *Annual review of fluid mechanics* **52**, 477 (2020).
 - [29] Z. Zhou, C. Joshi, R. Liu, M. M. Norton, L. Lemma, Z. Dogic, M. F. Hagan, S. Fraden, and P. Hong, *Soft Matter* **17**, 738 (2021).
 - [30] J. Colen, M. Han, R. Zhang, S. A. Redford, L. M. Lemma, L. Morgan, P. V. Ruijgrok, R. Adkins, Z. Bryant, Z. Dogic, M. L. Gardel, J. J. de Pablo, and V. Vitelli, *Proceedings of the National Academy of Sciences* **118**, e2016708118 (2021).
 - [31] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Židek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, *Nature* **596**, 583–589 (2021).
 - [32] C. J. Soelistyo, G. Vallardi, G. Charras, and A. R. Lowe, *Nature Machine Intelligence* **4**, 636 (2022).
 - [33] A. Zaritsky, A. R. Jamieson, E. S. Welf, A. Nevarez, J. Cillay, U. Eskiocak, B. L. Cantarel, and G. Danuser, *Cell Syst* **12**, 733 (2021).
 - [34] M. S. Schmitt, J. Colen, S. Sala, J. Devany, S. Seetharaman, A. Caillier, M. L. Gardel, P. W. Oakes, and V. Vitelli, *Cell* **187**, 481 (2024).
 - [35] M. Lefebvre, J. Colen, N. Claussen, F. Brauns, M. Raich, N. Mitchell, M. Fruchart, V. Vitelli, and S. J. Streichan, “Learning a conserved mechanism for early neuroectoderm morphogenesis,” (2023), [bioRxiv 2023.12.22.573058](#).
 - [36] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Fourier neural operator for parametric partial differential equations,” (2021), [arXiv:2010.08895 \[cs.LG\]](#).
 - [37] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar, “Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators,” (2022), [arXiv:2202.11214 \[physics.aoph\]](#).
 - [38] T. Beatus, R. H. Bar-Ziv, and T. Tlusty, *Physics Reports* **516**, 103 (2012).
 - [39] T. Beatus, R. Bar-Ziv, and T. Tlusty, *Physical Review Letters* **99**, 124502 (2007).
 - [40] N. Desreumaux, J.-B. Caussin, R. Jeanneret, E. Lauga, and D. Bartolo, *Physical Review Letters* **111**, 118301 (2013).
 - [41] T. Beatus, T. Tlusty, and R. Bar-Ziv, *Physical Review Letters* **103**, 114502 (2009).
 - [42] N. Champagne, E. Lauga, and D. Bartolo, *Soft Matter* **7**, 11082 (2011).
 - [43] I. Saeed, H. K. Pak, and T. Tlusty, *Nature Physics* **19**, 536 (2023).
 - [44] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Proceedings of the National Academy of Sciences* **113**, 3932 (2016).
 - [45] K. Kaheman, J. N. Kutz, and S. L. Brunton, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **476**, 20200279 (2020).
 - [46] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, *IEEE Access* **8**, 169259 (2020).
 - [47] C. R. Gin, D. E. Shea, S. L. Brunton, and J. N. Kutz, *Scientific Reports* **11**, 21614 (2021).
 - [48] J. Veenstra, O. Gamayun, X. Guo, A. Sarvi, C. V. Meiners, and C. Coullais, *Nature* **627**, 528 (2024).

Interpreting neural operators: how nonlinear waves propagate in non-reciprocal solids

Supplementary Information

Jonathan Colen^{*,1,2,3} Alexis Poncet^{*,4} Denis Bartolo,⁴ and Vincenzo Vitelli^{1,2}

¹*James Franck Institute, University of Chicago, Chicago, IL 60637*

²*Department of Physics, University of Chicago, Chicago, IL 60637*

³*Joint Institute on Advanced Computing for Environmental Studies,
Old Dominion University, Norfolk, Virginia 23539*

⁴*Univ. Lyon, ENS de Lyon, Univ. Claude Bernard,
CNRS, Laboratoire de Physique, F-69342, Lyon.*

CONTENTS

I. Neural network architecture and training	1
II. Trained network analysis	2
III. Linearizing Burgers' equation	4
IV. Applying KdV-Burgers' to experiments	4
V. Mapping the microscopic model onto the KdV-Burgers equation	6
A. Derivation of the KdV-Burgers equation	6
B. Some properties of the KdV and KdV-Burgers equation	7
C. Cubic non-linearities and modified KdV-Burgers equation	8
D. Discrete chain and Volterra lattice	8
VI. Numerical simulations	9
A. Numerical solution of the microscopic equations	9
B. Additional numerical simulations	9
C. Integration of KdV, KdV-Burgers and Burgers equations	10
D. Hydrodynamic interactions	10
References	11

I. NEURAL NETWORK ARCHITECTURE AND TRAINING

The neural network used in this paper was implemented in Pytorch [1]. It consisted of a read-in network, a single-layer linear Fourier neural operator, and a read-out network. The read-in and read-out networks were convolutional neural networks which learn nonlinear functions of their input fields. These used stacked ConvNext-style blocks which lifted the inputs to a high-dimensional feature space and learned the transformations in this representation. Each ConvNext block contained a grouped spatial convolution followed by two convolutions with kernel size 1 and an inverse bottleneck structure with expansion ratio 4. For example, a block with input and output channel size 16 had 3 convolutional layers: [grouped 16 \rightarrow 16, k1 16 \rightarrow 64, k1 64 \rightarrow 16] [2]. This structure enabled aggregation of short-range spatial information such as derivatives, and then subsequent local nonlinear transformations of those features. The read-in and read-out networks had identical structure, with two ConvNext blocks with channel sizes [1,64,64] and a final convolutional layer with channel size [64,1] and kernel size 1 to push the data back into the one-dimensional representation. These ConvNext blocks used 1D convolutions and acted only along the spatial dimension of the data.

The linear neural operator can be interpreted as a Green's function $G(\mathbf{r})$ and can be represented numerically as a d -dimensional vector, where each entry sets the interaction strength at separation \mathbf{r} [3]. We choose an alternative representation as in [4] and learn the Fourier-transformed Green's function $G(\mathbf{q})$. This reduces computational overhead

Term	Coef.	L2 Norm	Importance
ϕ	-0.31	1.64×10^2	51.1
$\partial_x \phi$	1.86	1.56×10^2	290
$\partial_x^2 \phi$	0.003	9.97×10^2	3.28
$\partial_x^3 \phi$	-0.008	1.43×10^4	113

TABLE S1. Importance values for each term in the machine-learned equation (2). Importance is the product of the coefficient and the L2 Norm of the corresponding feature column in the SINDy feature library.

for the convolution operation and enables measurement noise mitigation via a regularized loss function.

$$\mathcal{L} = \underbrace{\sum_i (F_i - F_i^{NN})^2}_{\text{Reconstruction}} + \beta \underbrace{\sum_{\mathbf{q}} |G(\mathbf{q})|^2}_{\text{Regularization}} \quad (\text{S1})$$

Such regularization improves interpretability by ensuring that the model learns a general and smooth solution to the PDE, rather than overfitting to noise appearing as high-wavenumber terms in $f(\mathbf{q})$.

In a standard deep convolutional neural network, such as a U-Net [5], each layer computes features from a local neighborhood of each pixel and the network propagates long-range information by stacking many convolutional layers. The dual role of each layer in determining local and non-local features, as well as the number of layers needed to obtain a large receptive field, makes it difficult to interpret what deep networks have learned.

The presented architecture simplifies interpretation by separating the problem into three manageable pieces which can be analyzed separately. The read-in and read-out modules compute local information at each point while the learned Green’s function describes all of the long-range information propagation. The network trains end-to-end, meaning a model that has learned to predict experimental data has also found a representation of the system that is governed by a PDE. Even if the inputs and outputs do not directly correspond to the fields in the continuum description, the read-in and read-out layers allow the network to learn a suitable transformation and a solution to the PDE governing those transformed variables. These features make this approach well-adapted for analyzing, testing, and identifying models of experiments with complex and noisy data.

We used this neural network to analyze the dynamics of a one-dimensional microfluidic-particle stream which develops a shock in response to a jam induced at $t = 0$ and is predicted to be described by a Burger’s equation [6]. We coarse-grained the droplet positions by first performing background subtraction (with the background defined as the average image frame over the entire movie) and averaging in the transverse direction. We next convolved the movie with a Gaussian kernel with width $\sigma_t = 0.05$ s (3 frames sampled at 60 Hz) and $\sigma_x = 0.5$ mm ≈ 3 particle diameters, and downsampled by a factor of 4 in space and time into a movie of size $7 \text{ s} \times 18 \text{ mm}$ (112×200 pixels). We segmented the resulting ρ field into 6,200 overlapping $T' \times L' = 3 \text{ s} \times 9 \text{ mm}$ (50×100 pixel) segments to use as training examples. We trained the network using the Adam optimizer with learning rate $\lambda = 10^{-4}$ on the convolutional layers and $\lambda = 10^{-2}$ on the Green’s function. We used a regularization parameter $\beta = 10^{-7}$ and trained for 1000 epochs using the Pytorch ReduceLROnPlateau scheduler (patience 10 epochs).

II. TRAINED NETWORK ANALYSIS

We used the PySINDy sparse regression package [7, 8] to learn a dynamical rule for the machine-learned field $\phi(x, t)$ in Fig. 3. We used a library of linear spatial derivatives of the field ϕ up to fifth order and the sequentially-thresholded least-squares optimizer with $\tau = 10^{-3}$ and $\alpha = 1$. The result was the equation (2) with the coefficients of ∂_x^4 , ∂_x^5 thresholded to zero by the SINDy algorithm.

We note that the coefficients are lower for the higher-order derivative terms – this is a consequence of the derivatives $\partial_x^n \phi$ having larger magnitudes as n increases. To quantify the relative importance of each term in (2), we multiply the absolute value of the learned coefficient by the L2 norm of the corresponding feature column in the SINDy library. This represents the typical weight that each term contributes to the dynamics $\partial_t \phi$. The results, reported in Table S1, show the significance of the dispersive term ∂_x^3 as greater than that of the diffusive or damping terms.

To test the accuracy of (2), we used the read-in network to compute the initial condition $\phi(x, t = 0)$ from the data. We simulated the above equation using a 4th order Runge-Kutta integration scheme over a $T = 3$ s window with $dt = 0.06$ s (both T and dt were the same as used by the neural network). This produced the ϕ_{SINDy} prediction shown in Fig. S1b. We then used the neural network read-out transformation to convert ϕ_{SINDy} to ρ_{SINDy} shown in Fig. S1d.

We also attempted to learn rules for the read-in and read-out transformations in hopes that they might yield sparse representations similar to the neural operator. For each transformation $\phi \rightarrow \rho$, $\rho \rightarrow \phi$, we used a library of the

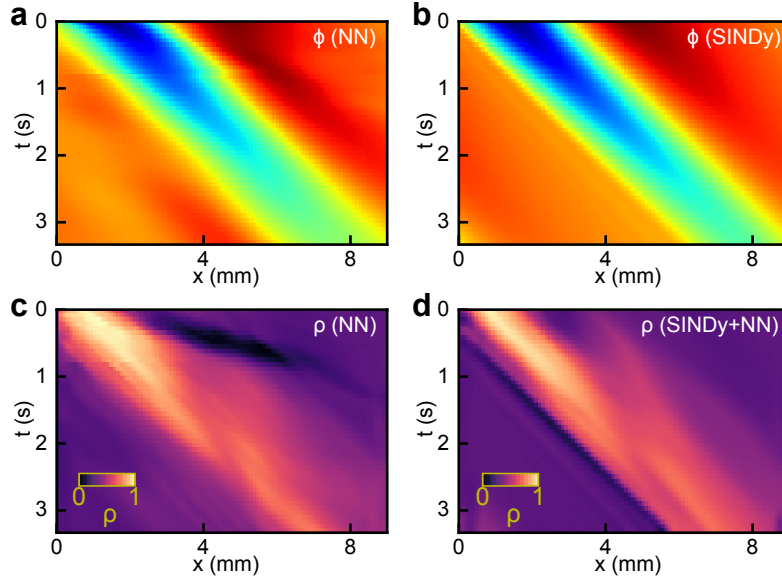


FIG. S1. (a-b) Comparison of ϕ predicted by NN (a) and predicted by the learned equation (2) (b). The equation achieves an $R^2 = 0.96$. (c-d) Comparison of ρ predicted by NN (c) and obtained by passing (b) through the NN read-out transformation (d). The mean absolute error between (c) and (d) is 0.063. The prediction in (d) achieves an MAE of 0.072 relative to experiments.

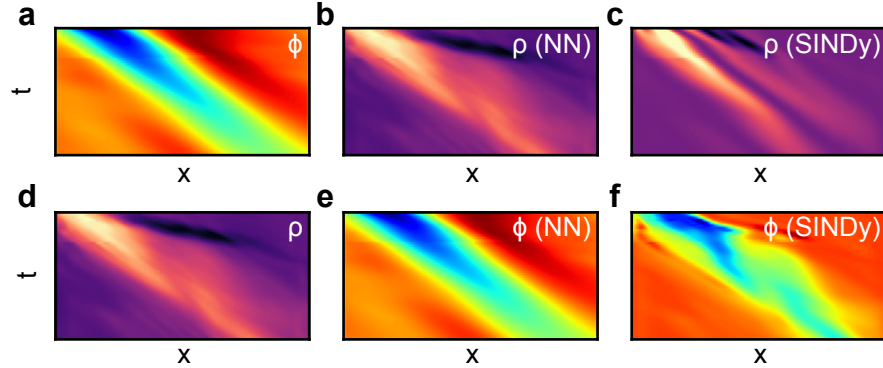


FIG. S2. Learning the transformations $\rho \rightarrow \phi$ and $\phi \rightarrow \rho$ for microfluidics experiment. (a) Plot of ϕ obtained via the read-in transformation and propagated using the neural operator layer. (b) Plot of ρ predicted by applying the read-out transformation to ϕ . (c) ρ_{SINDy} obtained by fitting the transformation $\phi \rightarrow \rho$ using the SINDy method. This method produces an equation with 14 terms and reaches an MAE of 0.085 relative to ρ (NN) and an MAE of 0.100 relative to experiments. (d-f) Repeat of top row for learning the read-in transformation $\rho \rightarrow \phi$. Note that $\phi(x, t)$ is not predicted by integrating the PDE (2) as in Fig. S1, but is instead predicted by applying a learned equation to the experimental density field $\rho(x, t)$. The identified equation contains 23 non-zero coefficients and achieves an $R^2 = 0.70$ relative to ϕ (NN).

input field, derivatives, nonlinear powers to order 3, sin, and cos. Our best learned models were not sparse. For the $\phi \rightarrow \rho$ transformation, we learned an equation which contained 14 terms and produced an MAE of 0.100 relative to experiments (MAE=0.085 relative to the ρ field predicted by the neural network), see Fig. S2a-c). For the $\rho \rightarrow \phi$, we learned an equation which contained 23 terms and achieved an $R^2 = 0.70$ relative to the ϕ field predicted by the NN, see SI Fig. S2d-f).

III. LINEARIZING BURGERS' EQUATION

Burgers' equation is a paradigmatic example describing shock generation in fluids. Consider an initial value problem for the viscous Burger's equation.

$$\partial_t u + u \partial_x u = \nu \partial_x^2 u \quad (\text{S2})$$

$$u(x, t = 0) = f(x) \quad (\text{S3})$$

On the surface, this is a nonlinear PDE and thus is not tractable using Green's function methods. However, it can be mapped onto a linear problem via a Cole-Hopf transformation.

$$u = -2\nu \frac{1}{\phi} \partial_x \phi \quad (\text{S4})$$

This change of variables turns (S2) into a linear diffusion equation

$$\partial_t \phi - \nu \partial_x^2 \phi = 0 \quad (\text{S5})$$

$$\phi(x, t = 0) = \exp \left(-\frac{1}{2\nu} \int f(y) dy \right) \quad (\text{S6})$$

This analogous initial value problem is equivalent to the response to an instantaneous impulse $f(x)$ at $t = 0$, and thus it can be solved by integration with the Green's function for the one-dimensional diffusion equation

$$G(x, t) = \frac{1}{\sqrt{4\pi\nu t}} \exp \left(-\frac{x^2}{4\nu t} \right) \Theta(t) \quad (\text{S7})$$

The Cole-Hopf result is a standard example of how nonlinear systems can become tractable to Green's function methods via a suitable transformation.

IV. APPLYING KDV-BURGERS' TO EXPERIMENTS

In Figure S3 we compared simulations of a Burgers' equation, a KdV equation, and a KdV-Burgers' equation from experimental initial conditions. We used a physics-informed neural network (PINN) [9] to fit the coefficients of each equation, as we found other approaches had trouble fitting the $\partial_x^3 \rho$ term, likely due to its amplification of experimental noise. Each PINN was each fully-connected neural network with 5 hidden layers of 50 neurons with sine activations and trained on 10000 randomly selected points from the dataset. They minimized the joint loss function

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N [\rho_i - \hat{\rho}_i]^2 + \frac{\beta}{N} \sum_{i=1}^N [\partial_t \rho_i + (c - \alpha) \rho_i \partial_x \rho_i - D \partial_x^2 \rho_i - \beta \partial_x^3 \rho_i]^2 \quad (\text{S8})$$

The first term is a reconstruction loss using the mean-squared deviation between prediction and experiments. The second term is a physics-informed loss which penalizes the deviation from the prescribed equation (3). We used $\beta = 10$ to weigh the physics-informed loss more heavily than the reconstruction loss. Following [9], we minimized the joint objective function using the L-BFGS optimizer. The final components of the loss function for each equation are summarized in Table S2.

Equation	MSE Loss	Phys. Loss
Burgers'	3.03×10^{-3}	1.66×10^{-5}
KdV	3.48×10^{-3}	2.91×10^{-5}
KdV-Burgers'	2.98×10^{-3}	1.37×10^{-5}

TABLE S2. PINN loss terms for variants of the general KdV-Burgers' equation. The full equation (3) achieves the lowest reconstruction error and the lowest physics error.

The Burgers' and KdV-Burgers' PINNs achieved similar reconstruction accuracy, while the KdV PINN performed worse. By construction, PINNs allow a tradeoff between faithfulness to the data and adherence to the prescribed

Solution	MAE	Shock-MAE
Neural network	0.049	0.051
Burgers'	0.067	0.074
KdV	0.11	0.14
KdV-Burgers'	0.068	0.067
Baseline	0.17	0.34

TABLE S3. Error rates for neural network (Fig. 2) and analytic solutions (Fig. 3). We compute the mean absolute error (MAE) $\langle |\rho - \rho_0| \rangle$ for the whole field and within a 1-mm region centered at the shock (Shock-MAE). Baseline reports the error rates of predicting a uniform density field given by the mean of the movie.

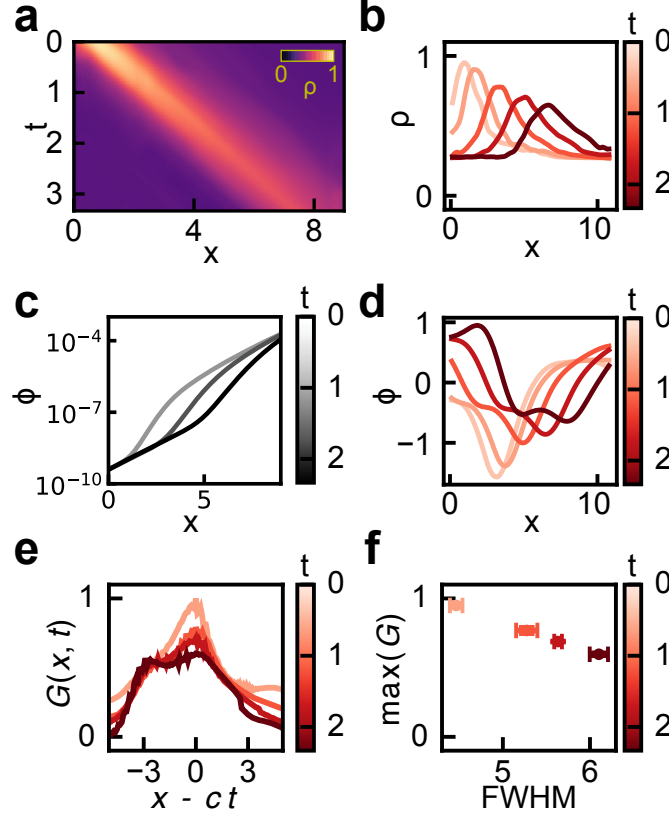


FIG. S3. Neural operator learns to predict density field that obeys Burgers' equation. (a) Neural network trained on Burgers' data in Fig. 3b predicts diffusive spreading of initial jam. (b) Neural network predictions of the density profile $\rho(x, t)$ at different time points. (c) Cole-Hopf transformation of the density field into an intermediate variable. (d) The intermediate variable $\phi(x, t)$ predicted by the neural operator is distinct from the Cole-Hopf variable in (c). (e) Neural network Green's function in the comoving frame at different time points. (f) The neural network's learned Green's function decreases in magnitude and increases in width over time.

physics. We observe that the KdV-Burgers' PINN achieves a lower physics-informed loss than the other two models. The combination of low reconstruction error while also closely satisfying the physical constraint suggests that the KdV-Burgers' equation may be more suitable to the experimental data than either the Burgers' or KdV limits.

To further test the accuracy of the PINN-identified coefficients, we integrated each equation from experimental initial conditions (Fig. 3b). These results agreed with the PINNs, with Burgers' and KdV-Burgers' performing well and KdV achieving much lower accuracy. The KdV-Burgers' equation achieved higher accuracy near the shock front (Fig. 3c, Table S3). Together, the PINN results and Fig. 3 corroborate the conclusion that the neural operator learned KdV-like physics in order to preserve the shock sharpness over time.

V. MAPPING THE MICROSCOPIC MODEL ONTO THE KDV-BURGERS EQUATION

In this section, we derive a PDE that describes the long wavelength dynamics of our microscopic model. The method follows the derivation of the KdV equation from the standard microscopic model offered by the Fermi-Pasta-Ulam-Tsingou chain, see [10] for a thorough presentation.

A. Derivation of the KdV-Burgers equation

We start from the microscopic equations of motion (Eq. (4) of the main text),

$$\zeta(\partial_t R_n - v_0) = \sum_{m \neq n} (f_{m \rightarrow n}^H + f_{m \rightarrow n}^E). \quad (\text{S9})$$

In the following we set $\zeta = 1$ without loss of generality. We define the local density $\rho(x)$ as

$$\rho(x = na) = \frac{1}{R_{n+1} - R_n - a} \quad (\text{S10})$$

where a is the lattice spacing, and x is the continuous analog of n . Our goal is to derive the dynamical equation obeyed by $\rho(x, t)$ in the limit of small density fluctuations and long wavelengths.

We start from a reference configuration on the infinite line with uniform lattice spacing a , so that the reference positions of the particles are $R_n^0 = na$. In term of the displacements $u_n(t) = R_n(t) - R_n^0$, Eq. (S9) becomes

$$\partial_t u_n = \sum_{j \geq 1} [f_H(ja + u_{n+j} - u_n) + f_H(ja + u_n - u_{n-j})] + \sum_{j \geq 1} [f_E(ja + u_{n+j} - u_n) - f_E(ja + u_n - u_{n-j})], \quad (\text{S11})$$

where $f_H(r)$ is the distance-dependent hydrodynamic force. In the narrow channels of our experiments, the hydrodynamic interactions are exponentially screened. $f_E(r)$ is the elastic force. Within a linear spring approximation, we have $f_E(a + \Delta r) = k\Delta r$, and the sum on $j \geq 1$ reduces to the $j = 1$ term only. We stress that the elastic interactions are reciprocal, in the sense that they obey Newton's third law of motion (action-reaction principle). Reciprocity manifests itself by the minus sign difference between the two forces in the second sum of Eq. (S11). By contrast, the hydrodynamic interactions are fully non-reciprocal, they do not obey the action-reaction principle, which results in a + sign in the first sum of Eq. (S11).

When the deformations $(u_{n \pm j} - u_n)$ are small, we can expand f_H and f_E up to second order,

$$\begin{aligned} \partial_t u_n = \sum_{j \geq 1} & \left[2f_H(ja) + f'_H(ja)(u_{n+j} - u_{n-j}) + \frac{f''_H(ja)}{2} ((u_{n+j} - u_n)^2 + (u_n - u_{n-j})^2) + \dots \right] \\ & + \sum_{j \geq 1} \left[f'_E(ja)(u_{n+j} + u_{n-j} - 2u_n) + \frac{f''_E(ja)}{2} ((u_{n+j} - u_n)^2 - (u_n - u_{n-j})^2) + \dots \right]. \end{aligned} \quad (\text{S12})$$

To build a continuous limit, we consider deformations of long wavelength, i.e of order a/ϵ with $\epsilon \rightarrow 0$. We can then define a (dimensionless) continuous spatial variable $\tilde{x} = \epsilon \frac{x}{a} = \epsilon n$. We then define the displacement field as $u_n = \nu(\epsilon) a \tilde{u}(\tilde{x})$. $\nu(\epsilon)$ is a scale factor that defines the magnitude of the displacements described by our continuum theory. The magnitude of the linear deformations are then of order $\nu\epsilon$. More precisely, we can expand the local deformations in powers of ϵ as

$$u_{n \pm j} - u_n = \nu a [\tilde{u}(\tilde{x} \pm j\epsilon) - \tilde{u}(\tilde{x})] = \nu a \left[\pm(j\epsilon) \partial_{\tilde{x}} \tilde{u} + \frac{(j\epsilon)^2}{2} \partial_{\tilde{x}\tilde{x}} \tilde{u} \pm \frac{(j\epsilon)^3}{6} \partial_{\tilde{x}\tilde{x}\tilde{x}} \tilde{u} + O(\epsilon^4) \right]. \quad (\text{S13})$$

Injecting this expansion in Eq. (S12), we find that \tilde{u} obeys

$$\partial_t \tilde{u}(\tilde{x}) = \left[V - c \partial_{\tilde{x}} \tilde{u} - \epsilon^2 \nu \frac{\alpha}{2} (\partial_{\tilde{x}} \tilde{u})^2 + \epsilon^3 \beta \partial_{\tilde{x}\tilde{x}\tilde{x}} \tilde{u} \right] + [\epsilon^2 K_1 \partial_{\tilde{x}\tilde{x}} \tilde{u} + \epsilon^3 \nu K_2 (\partial_{\tilde{x}} \tilde{u})(\partial_{\tilde{x}\tilde{x}} \tilde{u})] + O(\epsilon^4), \quad (\text{S14})$$

where the parameters $V, c, \alpha, \beta, K_1, K_2$ are defined as

$$V = 2 \sum_{j \geq 1} f_H(ja), \quad c = -2 \sum_{j \geq 1} j f'_H(ja), \quad \alpha = -2a \sum_{j \geq 1} j^2 f'_H(ja), \quad (\text{S15})$$

$$\beta = \frac{1}{3} \sum_{j \geq 1} j^3 f'''_H(j\epsilon), \quad K_1 = \sum_{j \geq 1} j^2 f'_E(ja), \quad K_2 = \frac{a}{2} \sum_{j \geq 1} j^3 f''_E(ja). \quad (\text{S16})$$

We note that in the case of nearest neighbor interactions, only the term $j = 1$ remains. In particular, for linear springs, we have $K_1 = k$ and $K_2 = 0$.

We can now write an equation of motion for the local deformations, and disregard the advection term $(\partial_{\tilde{x}}\tilde{u})$, as it can be eliminated by a change of reference frame. In all that follows, we use the new coordinate $\tilde{y} = \tilde{x} - \epsilon ct = \epsilon(n - ct)$. The equation for the deformation field $\tilde{w}(\tilde{y}) = \partial_{\tilde{y}}\tilde{u}$ then reduces to

$$\partial_t \tilde{w}(\tilde{y}) = \epsilon^2 K_1 \partial_{\tilde{y}\tilde{y}} \tilde{w} - (\epsilon^2 \nu) \alpha \tilde{w} \partial_{\tilde{y}} \tilde{w} + \epsilon^3 \beta \partial_{\tilde{y}\tilde{y}\tilde{y}} \tilde{w} + (\epsilon^3 \nu) K_2 \partial_{\tilde{y}} (\tilde{w} \partial_{\tilde{y}} \tilde{w}) + O(\epsilon^4). \quad (\text{S17})$$

We note that the K_2 term that originates from the elastic interactions is of order $\epsilon^3 \nu$. It is therefore negligible compared to the non-linear correction that originates from the hydrodynamic interactions (of order $\epsilon^2 \nu$). In other words, without loss of generality, the linear-spring approximation is sufficient to explain the large-scale physics of our system.

The non-reciprocal non-linear term $(\tilde{w} \partial_{\tilde{y}} \tilde{w})$ is known to yield shocks. The question is how are these shocks regularized. When the elastic interactions are strong ($K_1 \sim 1$), the elasticity regularizes perturbations of order $\nu \sim 1$, leading to the Burgers equation (at order ϵ^2)

$$\partial_{\tilde{t}} \tilde{w}(\tilde{y}) = K_1 \partial_{\tilde{y}\tilde{y}} \tilde{w} - \alpha \tilde{w} \partial_{\tilde{y}} \tilde{w}, \quad (\text{S18})$$

with the rescaled time $\tilde{t}' = \epsilon^2 \tau$.

When elastic interactions are vanishingly small ($K_1 = K_2 = 0$), the shocks are regularized by the dispersive term induced by the non-reciprocal interactions. For perturbations of order $\nu \sim \epsilon$, we find the well-known KdV equation (at order ϵ^3),

$$\partial_{\tilde{t}} \tilde{w} = -\alpha \tilde{w} \partial_{\tilde{x}} \tilde{w} + \beta \partial_{\tilde{x}\tilde{x}\tilde{x}} \tilde{w}, \quad (\text{S19})$$

with the rescaled time $\tilde{t} = \epsilon^3 t$.

Finally, when the elastic interactions are weak, $K_1 = \epsilon \tilde{K}_1$ with \tilde{K}_1 of order 1, the dispersion and the damping both contribute to regularize shocks. Eq. (S17) then takes the generic form of a KdV-Burgers equation (at order ϵ^3),

$$\partial_{\tilde{t}} \tilde{w} = \tilde{K}_1 \partial_{\tilde{y}\tilde{y}} \tilde{w} - \alpha \tilde{w} \partial_{\tilde{y}} \tilde{w} + \beta \partial_{\tilde{y}\tilde{y}\tilde{y}} \tilde{w}. \quad (\text{S20})$$

To compare the prediction of our continuum theory with microscopic simulations and experiments, we recast this dynamics into an equation of motion for the density field. For small density fluctuations $\rho - a^{-1} = \epsilon^2 \tilde{\rho}$ (see Eq. (S10)), and noticing that $\tilde{\rho} = -\tilde{w}$, we find

$$\partial_{\tilde{t}} \tilde{\rho} = \tilde{K}_1 \partial_{\tilde{y}\tilde{y}} \tilde{\rho} + \alpha \tilde{\rho} \partial_{\tilde{y}} \tilde{\rho} + \beta \partial_{\tilde{y}\tilde{y}\tilde{y}} \tilde{\rho}, \quad (\text{S21})$$

We stress that the scaling of all physical quantities with ϵ was essential to control our perturbative analysis. However, none of our prediction explicitly depend on ϵ . We can come back to the original variables, $t = \epsilon^{-3} \tilde{t}$, $x = \epsilon^{-1} \tilde{y} + ct$, $K_1 = \epsilon^{-1} \tilde{K}_1$, $\rho = a^{-1} + \epsilon^2 \tilde{\rho}$ and obtain

$$\partial_t \rho = -c \partial_x \rho + K_1 \partial_{xx} \rho + \alpha \rho \partial_x \rho + \beta \partial_{xxx} \rho, \quad (\text{S22})$$

which corresponds to Eq. (3) of the main text with $K_1 = D$. This equation is a priori valid for small perturbations, but we show in Fig. S4 that it gives reasonable predictions even for large perturbations.

In practice, in our numerical simulations, we choose $f_H(r) = -(\frac{\pi}{W})^2 \text{csch}^2(\frac{\pi r}{W})$ with $W = 2$ (and $a = 1$). This yields the following values for the parameters: $c = -3.44$, $\alpha = -13.4$, $\beta = 0.713$.

B. Some properties of the KdV and KdV-Burgers equation

In this section we recall some known properties of the KdV and KdV-Burgers equations [10, 11].

Let us first consider the KdV equation,

$$\partial_t \rho = \alpha \rho \partial_x \rho + \beta \partial_{xxx} \rho. \quad (\text{S23})$$

It is an integrable system, and its solution (given an initial condition) can be derived analytically by the inverse scattering method. In general, a solution is the sum of solitons propagating in one direction (to the left in Fig. S4), and dispersive waves that move in the opposite direction (to the right in Fig. S4). For some specific initial conditions,

the solutions reduce to a superposition of solitons only. In particular, the one-soliton solution of the KdV equation is parametrized by a single parameter $\kappa > 0$ that sets the width, the amplitude and the speed of the soliton:

$$\rho(x, t) = \frac{12\beta\kappa^2}{\alpha} \text{sech}^2 [\kappa (x + 4\beta\kappa^2 t)]. \quad (\text{S24})$$

The soliton propagates at velocity $-4\beta\kappa^2$. In our case $\beta > 0$, solitons propagate to the left.

The exact two-soliton solution of the KdV equation is governed by two parameters $\kappa_1, \kappa_2 > 0$ setting the speeds of the solitons (when they are well separated):

$$\rho(x, t) = \frac{12\beta}{\alpha} \frac{\kappa_1^2 - \kappa_2^2}{(\kappa_1 \coth X_1 - \kappa_2 \tanh X_2)^2} \left(\frac{\kappa_1^2}{\sinh^2 X_1} + \frac{\kappa_2^2}{\sinh^2 X_2} \right) \quad (\text{S25})$$

$$X_n = \kappa_n (x + 4\beta\kappa_n^2 t), \quad (\text{S26})$$

where $t = 0$ is the time at which the two solitons cross one another. We use this solution to initialize our numerical simulations. By noting that $\rho - \rho_0 \sim -\partial_x u$, we find

$$u(x, t) = -\frac{12\beta}{\alpha} \frac{\kappa_1^2 - \kappa_2^2}{\kappa_1 \coth X_1 - \kappa_2 \tanh X_2}. \quad (\text{S27})$$

We use this result to initialize our discrete chain in the simulations presented in Fig. 4 of the main text. In passing, we note that one can build an exact N -soliton solution of the KdV equation [11].

The KdV-Burgers equation, Eq. (S21), is no longer an integrable system. We choose to solve it by numerical integration (using a spectral solver described below).

C. Cubic non-linearities and modified KdV-Burgers equation

We make a technical remark that allows us to put our results in a broader context. In principle the expansion of the non-reciprocal interactions could solely include odd order terms:

$$f_H(ja + u_{n+j} - u_n) = f'_H(ja)(u_{n+j} - u_n) + \frac{f'''_H(ja)}{6}(u_{n+j} - u_n)^3 + \dots \quad (\text{S28})$$

In this case, Eqs. (S14) and (S17) would become

$$\partial_t \tilde{u}(\tilde{x}) = \left[-\epsilon c \partial_{\tilde{x}} \tilde{u} + \epsilon^3 \nu^2 \frac{\gamma}{3} (\partial_{\tilde{x}} \tilde{u})^3 + \epsilon^3 \beta \partial_{\tilde{x}\tilde{x}\tilde{x}} \tilde{u} \right] + [\epsilon^2 K_1 \partial_{\tilde{x}\tilde{x}} \tilde{u} + \epsilon^3 \nu K_2 (\partial_{\tilde{x}} \tilde{u}) (\partial_{\tilde{x}\tilde{x}} \tilde{u})] + O(\epsilon^4), \quad (\text{S29})$$

$$\partial_t \tilde{w}(\tilde{y}) = \epsilon^2 K_1 \partial_{\tilde{y}\tilde{y}} \tilde{w} + (\epsilon^3 \nu^2) \gamma \tilde{w}^2 \partial_{\tilde{y}} \tilde{w} + \epsilon^3 \beta \partial_{\tilde{y}\tilde{y}\tilde{y}} \tilde{w} + (\epsilon^3 \nu) K_2 \partial_{\tilde{y}} (\tilde{w} \partial_{\tilde{y}} \tilde{w}) + O(\epsilon^4), \quad (\text{S30})$$

with $\gamma = 3a^2 \sum_{j \geq 1} j^3 f'_H(ja)$. When the elasticity is small $K_1 = \epsilon \tilde{K}_1, K_2 = \epsilon \tilde{K}_2$, we may take $\nu \sim 1$ and obtain the following equation at order ϵ^3 ,

$$\partial_{\tilde{t}} \tilde{w}(\tilde{y}) = K_1 \partial_{\tilde{y}\tilde{y}} \tilde{w} + \gamma \tilde{w}^2 \partial_{\tilde{y}} \tilde{w} + \beta \partial_{\tilde{y}\tilde{y}\tilde{y}} \tilde{w}, \quad (\text{S31})$$

with $\tilde{t} = \epsilon^3 t$. The same equation holds for the scaled density fluctuations $\tilde{\rho}$. This PDE is known as the modified KdV-Burgers equation, which reduces to the mKdV (modified Korteweg-de Vries) equation when there is no elasticity [10, 11]. Like the KdV equation, the mKdV equation is integrable and enjoys exact soliton solutions.

D. Discrete chain and Volterra lattice

Deriving continuous descriptions as we did in the previous sections is appealing, but we note for completeness that some results can already be obtained at the discrete level for a specific choice of forces. Let us consider exponentially decaying non-reciprocal forces

$$f_H(r) = A e^{-|r|/b}, \quad (\text{S32})$$

with no potential forces $f_E = 0$. In this case, Eq. (S9) (with $v_0 = 0$ for simplicity) reads

$$\partial_t R_n = \tilde{A} \left[e^{-\Delta_n/b} + e^{-\Delta_{n-1}/b} \right], \quad (\text{S33})$$

where $\tilde{A} = Ae^{-a}/\zeta$ and the local deformation is $\Delta_n = R_{n+1} - R_n - a$, with $R_{n+1} - R_n > 0$ at all times. This equation takes a simple form when written in terms of the quantity $v_n = -(b/\tilde{A})e^{-\Delta_n/b}$,

$$\partial_t v_n = v_n(v_{n+1} - v_{n-1}). \quad (\text{S34})$$

This set of equations is known as the Volterra lattice, as it describes a minimal predators and preys model involving an infinite number of species.

The Volterra lattice is an integrable system. We can define a Lax pair (M, L) satisfying $\partial_t L = [M, L]$, where L and M are operators acting on sequences f_n as $(Lf)_n = v_n f_{n-1} + f_{n+1}$ and $(Mf)_n = (v_n + v_{n+1})f_n + f_{n+2}$. Importantly, the Volterra lattice enjoys solitonic solutions [12].

We note here the strong similarity between the Volterra lattice and the Toda chain. The Toda chain is an integrable discrete model akin to the Fermi-Pasta-Ulam-Tsingou (FPUT) chain [13]. From this perspective, instead of thinking of the non-linear waves seen in our simulations as KdV solitons, we may alternatively see them as (approximate) excitations of a Volterra lattice for suitably chosen parameters.

VI. NUMERICAL SIMULATIONS

A. Numerical solution of the microscopic equations

We discuss the details of the numerical simulations reported in Fig. 4 (Main text) and Fig. S4 (below). We initially set $N = 500$ particles on a line at positions $R_n(t=0) = i + u_n^0$ for $i = n, \dots, N$ where we used a lattice spacing $a = 1$ and u_n^0 is the initial perturbation. Periodic boundary conditions are enforced, in a box length $L = N + u_N - u_1$, so that there is no deformation at the edge of the box. In the main text (Fig. 4), we input an exact two-soliton solution of the KdV equation: $u_n^0 = u(na, t_0)$ using Eq. (S27) with $t_0 = -10^3$ and the soliton speeds $V_1 = 4\beta\kappa_1^2 = 0.05$ and $V_2 = 4\beta\kappa_2^2 = 0.01$. The parameters α and β are chosen according to the derivation of subsection V A.

In Fig. S4, we chose an “arbitrary” initial condition as the sum of two gaussians:

$$\rho^0(x) - a^{-1} \approx h_1 e^{-\frac{(x-x_1)^2}{2w_1^2}} + h_2 e^{-\frac{(x-x_2)^2}{2w_2^2}}. \quad (\text{S35})$$

More precisely, we used

$$u_n^0 = -\sqrt{\frac{\pi}{2}} \left[w_1 h_1 \operatorname{erf} \left(\frac{n-x_1}{\sqrt{2}w_1} \right) + w_2 h_2 \operatorname{erf} \left(\frac{n-x_2}{\sqrt{2}w_2} \right) \right], \quad (\text{S36})$$

with $x_1 - x_2 = 30$, $w_1 = w_2 = 10$. In Fig. S4a, $(h_1, h_2) = (0.015, -0.005)$ while in Fig. S4b, $(h_1, h_2) = (0.15, -0.02)$. The x axes of Fig. 4 and Fig. S4 are shifted so that the perturbation is initially around the origin.

The equations of motion are

$$\partial_t u_n = \sum_{n \neq m} f_H(d_{n,m}) + k(u_{n+1} + u_{n-1} - 2u_n) \quad (\text{S37})$$

where $d_{n,m}$ is the distance between particles n and m (for $m > n$, $d_{n,m} = \text{per}(m - n + u_m - u_n)$ where per periodizes the distance in the interval $(-L/2, L/2]$). The hydrodynamic force is $f_H(r) = -(\frac{\pi}{W})^2 \operatorname{csch}^2(\frac{\pi r}{W})$ with $W = 2$ (see subsection VID). The elastic constant is $k = 0, 0.02$ or 0.2 depending on the panels of the figures.

The N coupled equations of motion (S37) are integrated using the Julia package `DifferentialEquations.jl` [14] with algorithm `AutoVern7(Tsit5())` and maximal timestep 0.1 .

B. Additional numerical simulations

In Fig. S4, we present numerical simulations with two other initial conditions in addition to the ones shown in the main text. Fig. S4a, shows an generic small perturbation that decays into two solitons traveling to the left, and dispersive waves travelling to the right. The agreement with the KdV-Burgers equation is quantitative and the strong elasticity limit is described by Burgers’ equation.

Fig. S4b shows a perturbation of larger amplitude. In the absence of elasticity, a large number of solitons is generated in agreement with the KdV prediction. While the perturbation is rather large, it is still well described the KdV-Burgers equation (and the Burgers equation at large damping).

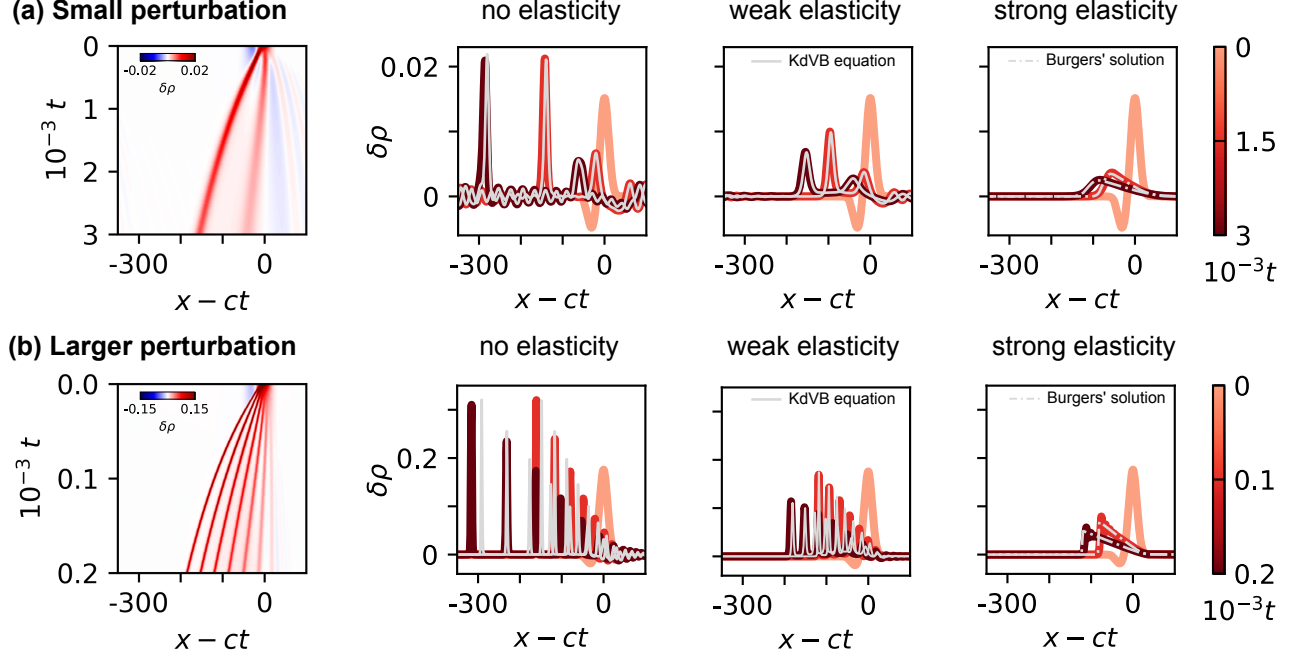


FIG. S4. Numerical simulations with different initial conditions. (a) Small perturbation ($\delta\rho \sim 0.2$). The left panel is the spatiotemporal plot corresponding to a weak elasticity ($k = 0.02$). The initial condition decomposes into two solitons and dispersive waves as expected from the KdV phenomenology. The three other panels correspond to no elasticity ($k = 0$), weak elasticity ($k = 0.02$) and strong elasticity ($k = 0.2$). In the three cases the evolution is well described by the KdV-Burgers equation derived from the microscopic parameters (thin gray line), see text (without elasticity, this reduces to the KdV equation). The large elasticity case is well described by the Burgers equation (dashed lines). (b) Larger perturbation ($\delta\rho \sim 0.15$). The four panels correspond to the same parameters as (a). In this case, the initial condition decays in a large number of solitons, well predicted by the KdV-Burgers equation. The strong elasticity limit is still predicted by the Burgers equation.

C. Integration of KdV, KdV-Burgers and Burgers equations

We now explain how we solve the nonlinear PDEs (KdV, Burgers and KdV-Burgers) to compare our continuum models to the discrete numerical simulations.

KdV equation. The dashed gray curves in Fig. 4e correspond to the analytical 2-soliton solution of the KdV equation, Eq. (S25), at times $t_0 + 2 \cdot 10^3$ and $t_0 + 4 \cdot 10^3$.

KdV-B equation. The continuous gray curves in Fig. 4d and Fig. S4 correspond to a numerical integration of the KdV-Burgers equation (S22) with the parameters derived in subsection V A. We use the Dedalus package (spectral solver) [15] with standard parameters.

Burgers equation. The dashdotted gray line in Fig. 4e and Fig. S4 is the solution of the Burgers equation, with the parameters of subsection V A. We use the results of section III to compute the time evolution of the initial condition: Cole-Hopf transformation, and (numerical) convolution with the Green's function of the diffusion equation.

D. Hydrodynamic interactions

In this last section, we provide more details on the specifics of the hydrodynamic interactions relevant to model our experiments. More details can be found in Ref. [16]. When squeezed by the confining walls, the droplets do not act as passive tracers. The viscous friction with the walls reduce their advection speed. We can therefore think of the droplets as intruders driven through a confined viscous fluid. Their motion perturb the mean flow and induce a velocity field $\mathbf{v}(x, y)$ that contribute to the advection of the other droplets.

We consider the 2D Stokes flow $\mathbf{v}_1(x, y)$ around a droplet in infinite space. It is incompressible and satisfies $\nabla \cdot \mathbf{v}_1(\mathbf{r}) = 2\pi\sigma\hat{\mathbf{e}}_x \cdot \nabla\delta(\mathbf{r})$ (source dipole). The solution can be expressed either in term of a potential $\phi_1(x, y)$ or a streamfunction $\psi_1(x, y)$. The complex potential defined as $w_1(z) = \phi_1(z) + i\psi_1(z)$, and the complex velocity

$v_1(z) = v_{1,x}(z) - iv_{1,y}(z)$ are related by the formula $v(z) = \frac{\partial w}{\partial z}$. Their expressions are

$$w_1(z) = \frac{\sigma}{z}, \quad v_1(z) = -\frac{\sigma}{z^2}. \quad (\text{S38})$$

Now instead of considering infinite space, we assume that the droplet is at the origin, in the middle of a channel of width W . The tangential velocity must vanish at the channel walls. To enforce this condition, we can use the method of images to compute the complex potential, and the corresponding complex velocity

$$w(z) = \sum_{k=-\infty}^{\infty} w_1(z - ikW) = \sigma \frac{\pi}{W} \coth\left(\frac{\pi z}{W}\right), \quad (\text{S39})$$

$$v(z) = -\sigma \left(\frac{\pi}{W}\right)^2 \sinh^{-2}\left(\frac{\pi z}{W}\right). \quad (\text{S40})$$

In terms of the coordinates (x, y) , this gives

$$\mathbf{v}(x, y) = \frac{-2\sigma(\pi/W)^2}{[\cosh(2\pi x/W) - \cos(2\pi y/W)]^2} \begin{pmatrix} \cosh(2\pi x/W) \cos(2\pi y/W) - 1 \\ \sinh(2\pi x/W) \sin(2\pi y/W) \end{pmatrix}. \quad (\text{S41})$$

Ignoring the fluctuations transverse to the x axis, we only need to consider

$$v_x(x, 0) = -\sigma \left(\frac{\pi}{W}\right)^2 \text{csch}^2\left(\frac{\pi x}{W}\right). \quad (\text{S42})$$

This velocity fields acts as an effective distance-dependant force on the other droplets. This explains our choice $f_H(r) = -\left(\frac{\pi}{W}\right)^2 \text{csch}^2\left(\frac{\pi r}{W}\right)$, where we set the strength to $\sigma = 1$ without loss of generality.

Two comments are in order: (i) since $v_x(x, 0) = v_x(-x, 0)$, the hydrodynamic interactions are strongly non-reciprocal, this fact is at the heart of the phenomenology that we detail in the main text ; (ii) the finite width of the channel implies a screening of the interactions: the hydrodynamic forces are short-ranged (exponential decay).

-
- [1] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” (2019), [arXiv:1912.01703 \[cs.LG\]](#).
 - [2] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022) pp. 11966–11976.
 - [3] C. R. Gin, D. E. Shea, S. L. Brunton, and J. N. Kutz, *Scientific Reports* **11**, 21614 (2021).
 - [4] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Fourier neural operator for parametric partial differential equations,” (2021), [arXiv:2010.08895 \[cs.LG\]](#).
 - [5] O. Ronneberger, P. Fischer, and T. Brox, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, edited by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Springer International Publishing, Cham, 2015) pp. 234–241.
 - [6] N. Champagne, E. Lauga, and D. Bartolo, *Soft Matter* **7**, 11082 (2011).
 - [7] B. d. Silva, K. Champion, M. Quade, J.-C. Loiseau, J. Kutz, and S. Brunton, *Journal of Open Source Software* **5**, 2104 (2020), publisher: The Open Journal.
 - [8] A. A. Kaptanoglu, B. M. d. Silva, U. Fasel, K. Kaheman, A. J. Goldschmidt, J. Callahan, C. B. Delahunt, Z. G. Nicolaou, K. Champion, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, *Journal of Open Source Software* **7**, 3994 (2022).
 - [9] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Journal of Computational Physics* **378**, 686 (2019).
 - [10] T. Dauxois and M. Peyrard, *Physics of Solitons* (Cambridge University Press, 2006).
 - [11] A. D. Polyanin and V. F. Zaitsev, *Handbook of Nonlinear Partial Differential Equations: Exact Solutions, Methods, and Problems* (Chapman and Hall/CRC, 2003).
 - [12] V. E. Adler and A. B. Shabat, *Theoretical and Mathematical Physics* **201**, 1442 (2019).
 - [13] M. Toda, *Journal of the Physical Society of Japan* **22**, 431 (1967).
 - [14] C. Rackauckas and Q. Nie, *Journal of Open Research Software* **5** (2017), 10.5334/jors.151.
 - [15] K. J. Burns, G. M. Vasil, J. S. Oishi, D. Lecoanet, and B. P. Brown, *Physical Review Research* **2**, 023068 (2020).
 - [16] T. Beatus, R. Bar-Ziv, and T. Tlusty, *Physical Review Letters* **99**, 124502 (2007).