RIGHT FEATURE, RIGHT PLACE: RESIDUAL-DRIVEN CONSTRUCTION OF CAUCHY-ADAPTIVE PHYSICS-INFORMED NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-informed neural networks (PINNs) solve partial differential equations by fitting a neural representation to the equation residual. Many solutions of scientific interest, from rogue waves and solitons to anomalous diffusion, carry isolated complex poles and algebraic tails, and at a finite size a PINN's accuracy depends on whether its neurons can represent this structure. Progress on PINN accuracy has centred on optimization, and adaptive methods tune features within one representation family chosen in advance. We introduce the Cauchy-Adaptive PINN (CAPINN), which lets the PDE residual choose each neuron's class, centre, and scale from a Cauchy-enriched dictionary of Cauchy, tanh-derived, and Gaussian neurons, building a compact basis over successive time intervals without solution labels. Standard PINN benchmarks favour tanh networks through their locally tanh-like fronts, so we evaluate CAPINN on a problem roster balanced for analytic structure. CAPINN is more accurate than vanilla PINNs and physics-informed Gaussians on every problem with poles or algebraic tails, and its basis adapts to each solution: Cauchy neurons place their poles along the withheld singularity trajectories, while smooth solutions are fitted with Gaussians alone. Front-dominated benchmarks remain the territory of tanh-based solvers. At a fixed neuron budget, how the representation is allocated is a lever complementary to optimization, and residual-driven allocation yields compact, interpretable solvers.

1 INTRODUCTION

Physics-informed neural networks (PINNs) turn governing equations and initial and boundary data into a training objective (Raissi et al., 2019). Their accuracy depends on both the representation and its optimization. Work on optimization addresses ill-conditioning and difficult loss landscapes (Krishnapriyan et al., 2021; Wang et al., 2021; Rathore et al., 2024). Here we study how to build a representation suited to the solution. Numerical solvers exploit Fourier modes for periodicity, local refinement for layers, and rational functions for singularity structure. These choices are especially consequential at a small neuron budget: each neuron must capture a useful part of the solution. We use the PDE residual to choose those building blocks as the solve proceeds.

Existing methods adapt activation slopes or function combinations, frequency features, and localized geometry (Jagtap et al., 2020; Wang et al., 2023; Wu et al., 2026; Kang et al., 2025). Frozen-PINN separates representation from fitting by fixing sampled hidden neurons and updating their output coefficients (Datar et al., 2026). This separation provides a foundation for adaptive construction: choose a neuron's analytic shape, location, and scale from the residual, freeze its geometry, and jointly refit the output coefficients. A fixed dictionary of structurally different analytic functions makes these choices explicit and interpretable.

A dictionary pays off when its functions cover qualitatively different analytic behaviour, because one neuron that matches the solution's structure can replace many that do not. Gaussian profiles capture localized smooth bumps, tanh-derived profiles capture transitions between two states, and Cauchy profiles carry algebraic tails. The Benjamin–Ono soliton $u(x,t) = 4c/[1 + c^2(x - ct)^2]$ illustrates the gain: at any fixed time a single Cauchy neuron represents it exactly, with a centre and width that locate the complex pole pair $ct \pm \mathrm{i}/c$, while tanh-derived and Gaussian neurons need many terms to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

approximate its x^{-2} tails. Rational approximation rests on the same principle of placing poles near the singularities of the target (Gopal & Trefethen, 2019; Nakatsukasa et al., 2018). In a PDE solve, the solution and its poles are unknown and move in time, so the residual has to infer this geometry from the equation and the prescribed data.

We introduce the Cauchy-Adaptive Physics-Informed Neural Network (CAPINN), an adaptive frozen-feature solver with a Cauchy-enriched dictionary of analytic neurons, combining Cauchy, tanh-derived, and Gaussian pairs. On each short time interval, it selects *what shape to add, where to place it, and how wide it should be* by predicting how much the residual would drop if the candidate were added to the current basis. It freezes the selected geometry, jointly refits all output coefficients, and accepts the addition only if the actual residual decreases sufficiently. A separate residual grid determines how many neurons to retain; the whole construction uses only the equation and the prescribed initial and boundary data. Each analytic neuron consists of two spatial functions. Selection uses their responses in the PDE residual, including the current linearization for nonlinear equations.

We evaluate this construction across eight configurations of seven one-dimensional PDE problems. Under matched ablation settings, removing Cauchy raises Benjamin–Ono error by over three orders of magnitude, while mixing classes improves Voigt error by $7.7\times$ over the best single-class configuration. These results quantify the value of selecting analytic building blocks at a finite neuron budget.

Our contributions are threefold. **(1) Residual-guided construction from a Cauchy-enriched dictionary.** CAPINN selects each neuron's class, centre, and scale, freezes accepted geometry, and refits only output coefficients: by linear least squares for linear PDEs and Gauss–Newton for nonlinear PDEs (Section 3). **(2) The case for Cauchy neurons, and a benchmark balanced for analytic structure.** Cauchy neurons encode a conjugate complex pole pair through their centres and widths and capture algebraic decay (Appendix A.1). Tanh-shaped fronts in common Burgers and Allen–Cahn benchmarks motivate a broader roster spanning rational waves, algebraic tails, smooth evolution, and nonlinear fronts. **(3) Accurate and interpretable solvers.** CAPINN is more accurate than vanilla PINNs and physics-informed Gaussians on all five benchmark configurations with poles or algebraic tails (Section 4.1). Its basis adapts to each solution: Cauchy neurons dominate on rational waves, Voigt mixes classes, periodic heat uses Gaussian neurons alone, and the selected Cauchy poles, inferred from the residual alone, align closely with the withheld Peregrine poles and approximately with those of Benjamin–Ono (Sections 4.2–4.3). The same construction extends to heavy-tailed flow matching (Section 4.4).

2 Related work

Frozen-feature PDE solvers. Frozen-PINN freezes sampled tanh neurons and fits output coefficients through causal time marching (Datar et al., 2026). Physics-informed extreme learning machines first applied this idea to PDEs (Dwivedi & Srinivasan, 2020); local extreme learning machines and random-feature solvers extend it with domain decomposition and random features (Dong & Li, 2021; Chen et al., 2022). Adaptive-growth networks expand their representations using residual information (Dang et al., 2024); IFeF-PINN alternates learned Fourier features with coefficient regression (Wu et al., 2026). Physics-Informed Gaussians (PIG) combine Gaussian embeddings with a lightweight network and learn their positions and shapes (Kang et al., 2025), providing a complementary approach to spatial adaptation.

Greedy sparse approximation. Orthogonal matching pursuit (OMP) selects atoms by residual correlation and jointly refits their coefficients (Pati et al., 1993; Tropp, 2004); block OMP selects groups (Eldar et al., 2010). Orthogonal least squares (OLS) selects by the residual decrease after refitting, which accounts for overlap with the current span (Soussen et al., 2013). These established rules also inform PDE approximation, where dictionary columns describe responses in the equation residual. Nonlinear equations require linearization and verification by an actual coefficient refit.

Activation functions and analytic structure. Adaptive activations learn slope parameters (Jagtap et al., 2020); specialized activation learning also optimizes combinations of candidate functions (Wang et al., 2023). Cauchy activations and XNet introduce rational building blocks motivated by approximation theory (Li et al., 2025). Rational approximation offers related mechanisms:

AAA builds rational approximants from function samples (Nakatsukasa et al., 2018), while rational Laplace solvers place poles near domain corners and fit them from boundary data (Gopal & Trefethen, 2019). These approaches connect representation design with singularity structure, through different sources of information and pole-placement rules.

Benchmark selection and representation alignment. PDEBench and PINNacle have made PINN evaluation broad and reproducible (Takamoto et al., 2022; Hao et al., 2024): PDEBench covers advection, Burgers, diffusion–reaction, diffusion–sorption, Darcy, shallow-water, and Navier–Stokes problems, and PINNacle more than 20 tasks built on the Burgers, Poisson, heat, Navier–Stokes, wave, Gray–Scott, and Kuramoto–Sivashinsky equations. Comparisons also depend on baseline quality and reporting choices (McGreivy & Hakim, 2024). Method papers commonly evaluate on smooth trigonometric solutions and fronts: Helmholtz, convection, and wave problems with sinusoidal exact solutions appear alongside viscous Burgers in Datar et al. (2026) and Wu et al. (2026), and alongside Allen–Cahn in Kang et al. (2025). These fronts often have tanh profiles, as in viscous Burgers shocks and Allen–Cahn interfaces (Samanta & Chidella, 2025; Lee et al., 2025), matching tanh representations. None of these suites or rosters includes an equation whose solutions are rational waves or carry algebraic tails, such as the nonlinear Schrödinger, Benjamin–Ono, or fractional diffusion equations, although these structures arise in rogue waves, internal waves, and anomalous transport. Our roster adds them alongside a smooth control and nonlinear fronts (Section 4.1).

3 METHOD

CAPINN constructs its basis from a Cauchy-enriched dictionary supplied by the designer. On each short time interval, the PDE residual selects analytic basis elements from a fixed dictionary, including their representation class, centre, and scale. Accepted geometry is frozen and all output coefficients are jointly refitted. The endpoint prediction then initializes the next interval, so the basis can change as the solution evolves. Throughout, a *representation family* (Cauchy, tanh-derived, or Gaussian) refers to a pair of spatial functions such as (3), evaluated together with their derivatives and nonlocal terms in the PDE residual.

3.1 TIME-SLAB ANALYTIC REPRESENTATION

We consider one-dimensional evolution equations

$$u_t = \mathcal{N}[u] \quad \text{in } \Omega \times (0,T], \qquad \mathcal{B}u = g \quad \text{on } \partial\Omega \times (0,T], \qquad u(\cdot,0) = u_0, \tag{1}$$

where $\mathcal{N}$ may be nonlinear or nonlocal. We divide $[0,T]$ into time intervals (slabs) $[t_s, t_{s+1}]$, set $\tau = 2(t-t_s)/(t_{s+1}-t_s) - 1$, and propagate the predicted endpoint of interval $s-1$ as the entrance state u_s^{in}. On interval s, CAPINN uses

$$\begin{aligned} u_s(x,t) = b(x,t) &+ \sum_{\ell=1}^{q} \sum_{m=0}^{M-1} \beta_{\ell m}^{(s)} \psi_\ell(x) T_m(\tau) \\ &+ \sum_{k=1}^{K_s} \sum_{j=1}^{2} \sum_{m=0}^{M-1} a_{kjm}^{(s)}\, \phi_{kj}(x; f_k, c_k, d_k) T_m(\tau). \end{aligned} \tag{2}$$

Here T_m is a Chebyshev polynomial in time. The shared functions are $\{\psi_\ell\} = \{1\}$, except for periodic heat and Michelson–Sivashinsky, which use $\{1, \sin x, \cos x\}$. Their coefficients β are fitted. The prescribed term b is zero in the main benchmark runs; the Peregrine ablations retain the same plane-wave background in every configuration (Appendix B.4). Each *analytic neuron* k consists of two spatial functions ϕ_{k1}, ϕ_{k2} of the same class f_k with a common centre c_k and scale d_k; its output coefficients vary in time through the M polynomials. Each selected neuron adds $2M$ coefficients per real output channel, for a total of $(q+2K_s)M$ per channel; complex fields use two real channels. Selection fixes (f_k, c_k, d_k), and subsequent fitting updates only $a^{(s)}$ and $\beta^{(s)}$.

For a Cauchy neuron, with $r = x - c_k$,

$$\phi_{k1}(x) = \frac{r}{r^2 + d_k^2}, \qquad \phi_{k2}(x) = \frac{1}{r^2 + d_k^2}. \tag{3}$$

These channels represent a conjugate pole pair at $c_k \pm \mathrm{i}d_k$. The centre locates the profile on the real axis, and the scale sets both its width and its pole distance. The other pairs are $(\tanh(r/w), \operatorname{sech}^2(r/w))$, with $w = d/\operatorname{arcosh}\sqrt{2}$, and (rG, G), with $G = \exp[-(\log 2)r^2/d^2]$. Thus d is the half-width at half-height of each even channel. Periodic heat and Michelson–Sivashinsky use periodic counterparts; Appendix A.1 specifies their formulas.

On the whole line, Cauchy channels have algebraic tails; finite Gaussian sums decay faster, and finite tanh-derived sums approach their limiting constants exponentially. After matching these constants, a weighted far-field criterion distinguishes the tails (Appendix A.1). For Benjamin–Ono, the Hilbert transform maps one Cauchy channel to the other, so the nonlocal term of the residual is available in closed form (Appendix A.1). These properties equip the dictionary to represent algebraic decay and evaluate the Benjamin–Ono nonlocal response analytically.

Why freeze geometry? Freezing makes adaptive growth tractable: candidates are scored by projections, and each accepted addition is followed by a joint coefficient refit. The fixed centres and widths preserve a geometric interpretation throughout the solve. Frozen geometry is also the natural setting for Cauchy neurons: the universal-approximation proof for Cauchy networks (Theorem 2 and its proof in the appendix of Li et al., 2025) is constructive and two-stage, fixing the pole locations and scales before the target is seen and then determining only the outer coefficients from the target, so a Cauchy network with frozen geometry and a fitted linear readout already carries the expressivity that theorem guarantees. Classical results give the same separation for smooth activations (Mhaskar, 1996). Appendix A.3 states these results and their scope.

3.2 Residual-guided network construction

Coefficient fitting. Let X^{pde}, X^{in}, and X^{bc} denote the training collocation points for the PDE, entrance, and applicable boundary constraints. With fixed weights ω, the training residual and loss are

$$R_s(a) = \begin{bmatrix} \omega_{\mathrm{pde}}(\partial_t u_s - \mathcal{N}[u_s])|_{X^{\mathrm{pde}}} \\ \omega_{\mathrm{in}}(u_s - u_s^{\mathrm{in}})|_{X^{\mathrm{in}}} \\ \omega_{\mathrm{bc}}(\mathcal{B}u_s - g)|_{X^{\mathrm{bc}}} \end{bmatrix}, \qquad L_s(a) = \|R_s(a)\|_2^2. \tag{4}$$

Here a collects all fitted coefficients, including β in (2). The field is linear in these coefficients, but a nonlinear PDE generally produces a nonlinear residual. For nonlinear $\mathcal{N}$, we use coefficient-only Gauss–Newton updates,

$$\delta = \arg\min_{\delta} \|J_s(a)\delta + R_s(a)\|_2, \qquad a \leftarrow a + \delta, \tag{5}$$

where $J_s = \partial R_s/\partial a$. A linear PDE requires only a linear least-squares solve; no neuron geometry is optimised by gradient descent.

Candidate dictionary. Each representation family starts from prescribed grids of centres and scales. We score these candidates, refine the best candidate within each class by a local search, and compare the refined candidates across classes. The minimum scale is tied to the spatial resolution. The same search and neuron cap are used when restricting the dictionary to a single representation class; the grids and four-round refinement schedule are specified in Appendix B.

Initial neuron. We first fit one neuron to the incoming state after subtracting the background, then refit its coefficients against the full loss (4). Every subsequent neuron is scored using all three residual terms: the PDE, the incoming state, and the boundary conditions.

Scoring a new contribution. Let $r = R_s(a)$ be the current residual. The columns of $J_{\mathcal{V}}$ describe how this residual changes when the current coefficients change; the columns of C describe changes available from a candidate neuron's $2M$ new coefficients per real output channel. Write P_A for the orthogonal projector onto the column space of A. We remove directions already available through $J_{\mathcal{V}}$, then measure how much of the remaining residual the candidate can explain:

$$r_\perp = (I - P_{J_{\mathcal{V}}})r, \qquad C_\perp = (I - P_{J_{\mathcal{V}}})C, \qquad S(C) = \|P_{C_\perp} r_\perp\|_2^2. \tag{6}$$

For a linear PDE with fitted current coefficients, $S(C)$ is the exact squared-residual decrease after adding the candidate and refitting (Appendix A.2); for a nonlinear PDE it predicts the decrease

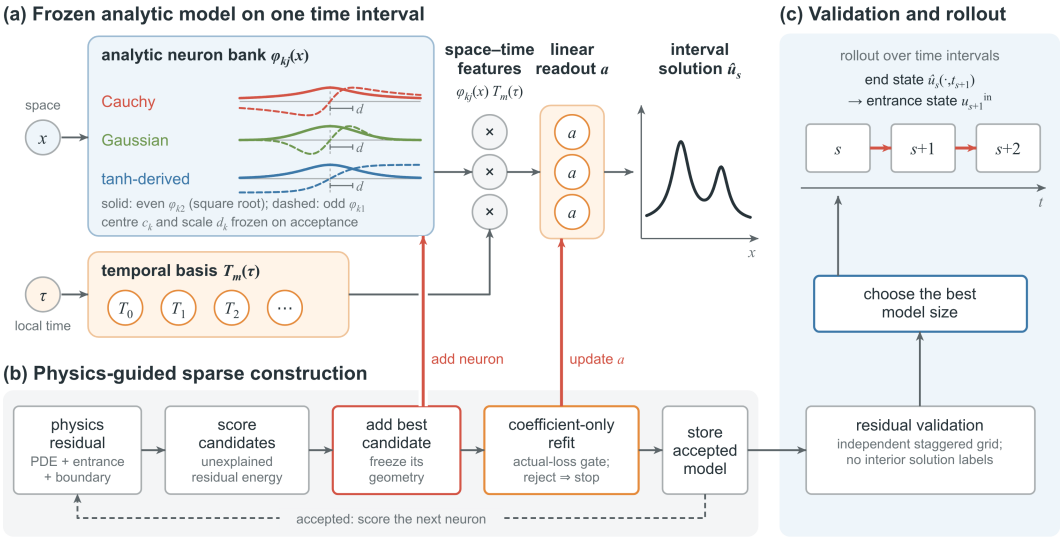

Figure 1: CAPINN on one time interval. A fixed multiscale dictionary provides Cauchy, tanh-derived, and Gaussian neurons. Shared spatial functions are fitted throughout. One analytic neuron is first fitted to the incoming state. The PDE residual then selects one neuron at a time; each selected neuron is frozen and only the output coefficients are refitted. The network size is chosen on a separate validation grid, and the end state is passed to the next interval.

in the linearized problem. This is a block OLS selection rule (Soussen et al., 2013): each neuron contributes $2M$ columns per real output channel, scored after projection away from the current span. Block OMP ranks candidates by their correlation with the residual (Eldar et al., 2010); CAPINN ranks by the refitted residual decrease, which accounts for overlap with the current span. CAPINN uses this established greedy least-squares principle to construct a Cauchy-enriched PDE basis, with geometry refinement, nonlinear acceptance checks, and residual validation. The same score applies to Student-t Flow regression (Appendix C).

Acceptance and model size. We add the highest-scoring neuron tentatively and refit all coefficients. This step checks the score's prediction against the actual PDE loss. We accept the neuron only when the loss falls below a fixed fraction $\eta < 1$ of its previous value,

$$L_s(a_{\text{trial}}) < \eta L_s(a), \tag{7}$$

otherwise growth on this interval stops. After each accepted neuron, the network is evaluated on a separate validation grid, staggered in space and with one additional time node, using the same residuals. The network size with the lowest validation residual is kept, and its value at the end of the interval becomes the initial state of the next interval. Training residuals thus propose and fit neurons, and validation residuals decide how many to keep; no solution values inside the domain are used.

Cost. Each selected neuron adds $2M$ fitted coefficients per real output channel, in addition to the qM shared coefficients. Scoring a candidate is a projection onto its $2M$ response columns, and coefficient refitting dominates the cost of each interval. On one CPU thread, the reported CAPINN fits take from about two minutes (conventional Peregrine, periodic heat) to over two hours (Burgers, Allen–Cahn) (Table 6). On the rational-wave and Voigt problems this is comparable to or shorter than the GPU baselines, but on Burgers and periodic heat Frozen-PINN fits in seconds. These timings indicate the cost of representation selection; differing hardware and stopping conditions prevent a controlled speed comparison. Per-problem settings are listed in Appendix B.

Algorithm 1 Residual-guided analytic-neuron selection

Require: PDE (1); allowed representation classes $\mathcal{F}$; fixed grids; maximum neurons K; ε, η

for $s = 0, \dots, S-1$ **do**
  obtain u_s^{in} from u_0 or the preceding predicted endpoint
  $\mathcal{V} \leftarrow \{$best one-neuron fit to $u_s^{\mathrm{in}}\}$; refit a on L_s; store model $(\mathcal{V}, a)$
  while $|\mathcal{V}| < K$ **and** $\|R_s(a)\|_2 > \varepsilon$ **do**
    refine the highest-scoring candidate of each $f \in \mathcal{F}$ under (6)
    $C^\star \leftarrow \arg\max_C S(C)$; refit a_{trial} on $\mathcal{V} \cup \{C^\star\}$
    if $L_s(a_{\mathrm{trial}}) \geq \eta L_s(a)$ **then**
      break
    end if
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{C^\star\}$; $a \leftarrow a_{\mathrm{trial}}$; store model $(\mathcal{V}, a)$
  end while
  retain the stored model with lowest independent residual validation; propagate its endpoint
end for

4 EXPERIMENTS

We evaluate accuracy and selected analytic structure on separate grids against exact or spectral references. Fitting uses prescribed initial and boundary data, with interior values and pole locations withheld (Appendix B).

Problem roster. Seven PDEs span complementary structures: Peregrine focusing on conventional and wide boxes, the travelling Benjamin–Ono soliton, the Gaussian core and algebraic tails of Voigt, periodic Michelson–Sivashinsky dynamics admitting a pole decomposition, smooth periodic heat, and the nonlinear fronts of Burgers and Allen–Cahn. This roster tests Cauchy enrichment across rational, composite, smooth, and front-dominated solutions.

Baselines. Vanilla PINN (Raissi et al., 2019), Frozen-PINN (Datar et al., 2026), and PIG (Kang et al., 2025) cover trained tanh networks, frozen tanh features, and adaptive Gaussian representations. We follow available implementations and settings, with adaptations and budgets in Appendix B.3, and include published results matching the equation, domain, horizon, and metric.

4.1 RESULTS ON BENCHMARK PDES

Table 1 reports eight configurations. CAPINN reaches relative L^2 errors below 10^{-6} on both Peregrine boxes and Benjamin–Ono, 6.7×10^{-6} on Voigt, and 4.9×10^{-9} on periodic heat. Michelson–Sivashinsky is the least accurate case at 2.0×10^{-3}. The wide Peregrine box exposes more of the algebraic tails; Appendix B specifies all problems and models.

Vanilla PINN and PIG use source recipes without target-specific searches; PIG uses fixed collocation sites instead of per-iteration resampling. Frozen-PINN is reported where its released formulation applies. Extensions to Peregrine, Benjamin–Ono, and Voigt failed the pre-registered eligibility checks and are documented separately (Appendix B.3).

4.2 CASE STUDY: ERROR PATTERNS AND POLE GEOMETRY

We examine spatial errors on wide-box Peregrine, Voigt, and Benjamin–Ono, then compare selected Cauchy geometry with exact pole trajectories. Figure labels abbreviate the tanh-derived class as tanh.

4.2.1 SPATIAL ERROR PATTERNS

Analytic structure. Peregrine has poles at $x = \pm \mathrm{i}\delta(t)$, $\delta(t) = \frac{1}{2}\sqrt{1+4t^2}$, and an x^{-2} approach to its unit background (Peregrine, 1983). Half-heat evolution produces a Voigt profile $G_\sigma * P_{\gamma t}$ with tails $\gamma t/(\pi x^2)$. Benjamin–Ono carries a pole pair along $x_0 + ct \pm \mathrm{i}/c$ over $t \in [0, 110]$ (Appendices B.4–B.6).

Table 1: Relative L^2 error over the full space–time domain. Bold marks the lowest error per row, excluding the numerical reference. Budgets, stopping reasons, and model sizes are in Appendix B.3.

Problem	Domain, horizon	CAPINN	Vanilla PINN	Frozen-PINN	PIG	Published[a]	Numerical ref.[b]
Peregrine (conv.)[c]	$x \in [-2, 2]$ $t \in [-2, 2]$	**5.77e-7**	2.32e-5	n/a	2.84e-3	2.48e-5 ESA	8.0e-9
Peregrine (wide)	$x \in [-20, 20]$ $t \in [-3, 3]$	**4.23e-7**	5.66e-1†	n/a	5.20e-1†	—	6.3e-9
Benjamin–Ono soliton	$x \in [-100, 100]$ $t \in [0, 110]$	**4.22e-7**	6.97e-2	n/a	7.21e-1†	—	2.6e-6
Voigt	$x \in [-20, 20]$ $t \in [0, 2]$	**6.74e-6**	3.78e-3	n/a	1.57e-3	—	2.7e-6
Periodic heat	$x \in [-\pi, \pi]$ $t \in [0, 2]$	**4.87e-9**‡	4.69e-5	2.1e-5	1.85e-4	—	8.2e-16
Michelson–Sivashinsky	$x \in [-\pi, \pi]$ $t \in [0, 4]$	**2.01e-3**‡	8.45e-1†	n/a	9.88e-1†	—	1.7e-11
Burgers	$x \in [-1, 1]$ $t \in [0, 1]$	3.74e-6*	9.91e-4	3.43e-7	1.15e-1†	**8.25e-9** vRBA	1.40e-6
Allen–Cahn	$x \in [-1, 1]$ $t \in [0, 1]$	4.09e-5*	9.92e-1†	2.07e0†,e	1.04e-4 ±4.1e-5 (Kang et al., 2025)	**1.88e-6** vRBA	3.71e-6

[a] Authors' reported result on the same equation, domain, horizon, and metric: ESA (Hu et al., 2025), vRBA (Toscano et al., 2026).
[b] Error of a spectral or Chebyshev solver against the exact solution, or its coarse–fine difference where no exact solution exists. The Burgers and Allen–Cahn gaps exceed some of the reported neural errors (Frozen-PINN and vRBA on Burgers; vRBA on Allen–Cahn), so the reference cannot rank those methods.
[c] Error on the amplitude $|u|$, as in the published result.
[e] Adaptation of the released periodic nonlinear example; its basis and derivative-matching initialization are incompatible with parts of these initial data (Appendix B.10).
* Burgers uses 769 spatial nodes (initially 513) and Allen–Cahn uses time-interval entrance weight 30 (initially 3); the same change is applied to every representation configuration (Appendix B.3).
‡ Later tuned Mixed configuration; single-class configurations were not rerun at this setting (Appendix B).
† Not converged. n/a: unsupported by the released Frozen-PINN formulation. For Peregrine, Benjamin–Ono, and Voigt, pre-registered Frozen-PINN-style extensions did not pass the comparison gates (Appendix B.3).

Figure 2 compares spatial error maps across the three problems. On Peregrine, Gaussian errors concentrate near the core and remain visible in the tails, while tanh develops oscillations extending into the far field. On Benjamin–Ono, both leave errors across the domain. Cauchy-only and Mixed networks achieve relative L^2 errors near 4×10^{-7} on both rational waves. Voigt separates the representation classes differently: Gaussian errors are largest toward the tails, Cauchy errors concentrate around the core, and mixing reduces both, reaching 6.7×10^{-6}, $7.7\times$ lower than the best single-class configuration.

4.2.2 Emergent pole geometry

Figure 3 compares the selected neuron geometry directly with the solution's pole structure. For a Cauchy neuron with poles $z = a \pm iw$, the centre a and width w can be compared with the exact pole's real part and distance from the real axis. Neurons with larger fitted contributions align closely with $(0, \delta(t))$ on Peregrine and lie near $(x_0 + ct, 1/c)$ on Benjamin–Ono. The Benjamin–Ono widths form several bands, so this second agreement is approximate. Related work fits rational networks to numerical solutions to track singularities (Derevianko et al., 2026); here the plotted parameters are selected during the PDE solve.

4.3 Analytic-representation ablations

Table 2 tests the Cauchy-enriched dictionary by holding the fitting procedure, collocation grids, shared spatial functions, and neuron cap fixed within each problem, while changing the allowed classes. Validation determines the retained size, so the comparison matches the neuron cap. On Benjamin–Ono, Mixed retains only Cauchy elements and reaches 4.22×10^{-7} error. Removing the Cauchy representation raises the error to 5.93×10^{-4}, a factor of 1404. On wide Peregrine,

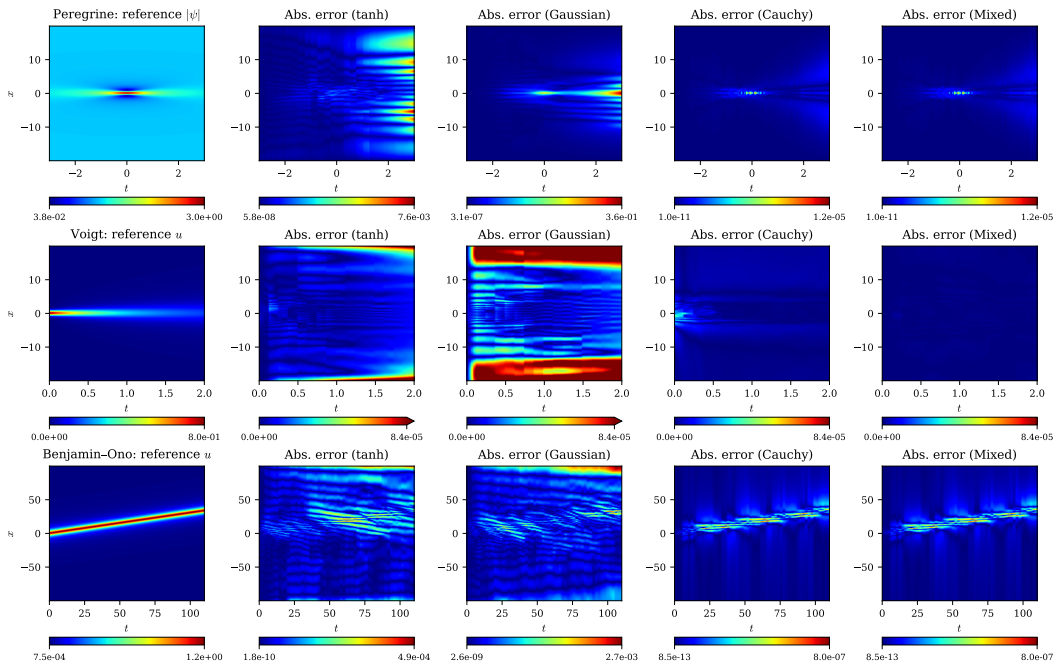

Figure 2: Reference fields and absolute errors for wide-box Peregrine (top), Voigt (middle), and Benjamin–Ono (bottom). Domains and horizons follow Table 1. Peregrine displays reference amplitude and complex-field error. Color scales are linear and individual, except the Voigt errors share a scale capped at the tanh 99th percentile (8.4×10^{-5}); marked panels saturate above this value.

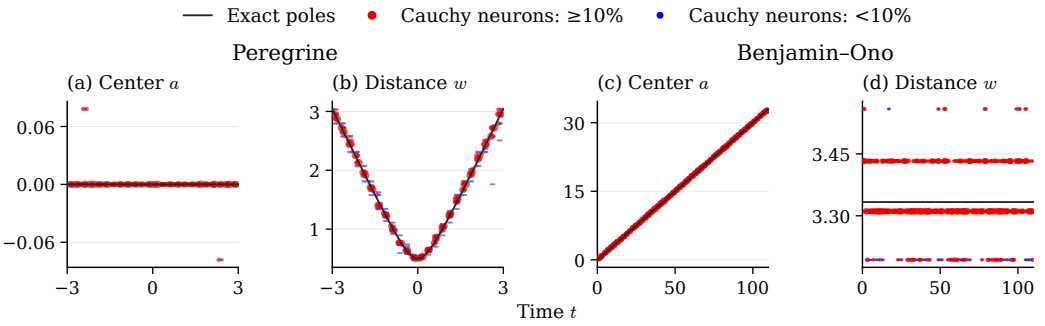

Figure 3: Cauchy-neuron pole geometry in the mixed models: Peregrine center and pole distance (left), then Benjamin–Ono center and pole distance (right). Black curves are exact trajectories used only after fitting. Red points have contribution share $q_j \geq 0.1$ and blue points have $q_j < 0.1$, where $q_j = \|v_j\|_{L^2} / \sum_k \|v_k\|_{L^2}$ across all selected neurons; size increases with q_j.

Cauchy-only and Mixed agree to within 3%, and excluding Cauchy raises the error by a factor of about 4900. These rational-wave results identify Cauchy elements as the key building blocks at the tested budgets.

Voigt tests a different possibility: several representation classes can be useful together. Over the full horizon $t \in [0, 2]$, Mixed reduces error by $7.7\times$ relative to the best single-class configuration. Periodic heat provides the complementary outcome: CAPINN selects only Gaussian elements and exactly matches the Gaussian-only error of 3.23×10^{-7} under the matched ablation setting. The lower heat error in Table 1 uses the separately disclosed tuned setting. Appendix B reports the ablation for every problem, including the short-horizon Voigt result.

Table 2: Matched analytic-representation ablations (relative L^2 error). "No C" allows the tanh-derived and Gaussian representations; Mixed allows all three under the matched settings of this table; Table 1 uses additional tuning for periodic heat. C/T/G gives the numbers of selected Cauchy/tanh/Gaussian elements summed over time intervals. Each column evaluates a complete analytic-representation configuration. Bold marks the lowest error in each row.

Problem	Cauchy	tanh	Gaussian	No C	Mixed	C/T/G
Peregrine, wide	**4.13e-7**	1.22e-3	3.24e-2	2.08e-3	4.23e-7	109/3/3
Benjamin–Ono	**4.22e-7**	4.52e-4	2.55e-3	5.93e-4	**4.22e-7**	275/0/0
Voigt, $t \in [0, 2]$	5.19e-5	2.65e-4	8.54e-4	2.32e-4	**6.74e-6**	85/31/87
Periodic heat	2.85e-5	9.47e-7	**3.23e-7**	**3.23e-7**	**3.23e-7**	0/0/72

Across the remaining problems, the best configuration is problem-dependent: Burgers favours the full mixture, Allen–Cahn favours tanh-derived plus Gaussian, and Michelson–Sivashinsky favours Cauchy alone. The Burgers run without Cauchy stops early at the acceptance test, so its error reflects early stopping as well as the dictionary. On Burgers, Frozen-PINN on the same grid is $10.9\times$ more accurate than Mixed.

4.4 Beyond PDEs: heavy-tailed generative transport

Flow matching fits a velocity field transporting source samples to a target distribution (Lipman et al., 2023). We apply CAPINN to the regression residual of Student-t Flow (Pandey et al., 2025), using a heavy-tailed Student-t source and a two-dimensional, four-mode heavy-tailed target with a 120-coefficient budget. Generated samples use a common ODE sampler and are evaluated by sliced Wasserstein-1 (SW1), independently of selection.

In a pre-registered study over 20 paired seeds, Mixed CAPINN lowers SW1 relative to the same dictionary without Cauchy neurons in 18 runs (exact paired sign-flip test, $p = 3.8 \times 10^{-6}$) and reduces mean SW1 by 24% (0.185 versus 0.243; 95% bootstrap interval of the paired difference $[0.042, 0.074]$). Residual selection retains a Cauchy neuron in every run and chooses one first in 19 of 20. Mixed also has the lowest mean SW1 of all five configurations, and both held-out tail audits favour it on average. Appendix C gives the protocol, the full results, and the selected neurons.

5 Conclusion and limitations

CAPINN lets the governing equation build the basis of a frozen-feature PINN: the PDE residual chooses each neuron's analytic class, centre, and scale, and only the output coefficients are fitted. The bases it builds follow the structure of the solution. Rational waves receive almost purely Cauchy bases, and removing Cauchy neurons raises Benjamin–Ono error by three orders of magnitude; the Voigt profile receives a mixture that is $7.7\times$ more accurate than any single class; periodic heat receives a purely Gaussian basis. Although no pole locations are ever supplied, the selected Cauchy neurons land on the complex poles of the Peregrine and Benjamin–Ono solutions. The same rule carries over to regression: in a pre-registered Student-t Flow study, the mixed dictionary beats the same dictionary without Cauchy on generated SW1 in 18 of 20 paired seeds. Under a fixed neuron budget, the equation can decide what a PINN is made of, as well as how well it fits.

The evidence is one-dimensional and uses three neuron classes; for nonlinear PDEs the selection score is a local prediction confirmed by refitting; and front-dominated problems, where tanh already matches the solution, stay with tanh-based Frozen-PINN on Burgers and with spectral methods on periodic problems. Extending the dictionary to other singularity types, to higher dimensions, and to gradient refinement of the selected geometry are the natural next steps.

Use of AI tools

Large language model assistants were used for editing and proofreading the text, for drafting and refactoring parts of the experiment and analysis code under the authors' direction, and for literature search whose results were verified against the original sources. The research questions, method,

theoretical statements and their derivations, experimental design, and the interpretation of results are the authors' own, and the authors take full responsibility for all content.

REPRODUCIBILITY STATEMENT

Section 3 specifies the dictionary, selection score, acceptance rule, and validation procedure, and Appendix B lists every problem, grid, budget, and baseline setting. All CAPINN runs are deterministic given their configuration. The Student-t Flow experiment, its pre-registered protocol, raw per-seed outputs, and a reproduction command are included in the supplementary material.

REFERENCES

Jingrun Chen, Xurong Chi, Weinan E, and Zhouwang Yang. Bridging traditional and machine learning-based algorithms for solving PDEs: The random feature method. *Journal of Machine Learning*, 1(3):268–298, 2022.

Haoning Dang, Fei Wang, and Song Jiang. Adaptive-growth randomized neural networks for PDEs: Algorithms and numerical analysis. *arXiv preprint arXiv:2408.17225*, 2024.

Chinmay Datar, Taniya Kapoor, Abhishek Chandra, Qing Sun, Erik Lien Bolager, Iryna Burak, Anna Veselovska, Massimo Fornasier, and Felix Dietrich. Fast training of accurate physics-informed neural networks without gradient descent. In *International Conference on Learning Representations*, 2026.

Nadiia Derevianko, Hans-Joachim Bungartz, and Felix Dietrich. Rational neural networks for tracking complex singularities of nonlinear PDEs. *arXiv preprint arXiv:2608.08134*, 2026. URL https://arxiv.org/abs/2608.08134v1.

Suchuan Dong and Zongwei Li. Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 387:114129, 2021.

Vikas Dwivedi and Balaji Srinivasan. Physics informed extreme learning machine (PIELM)–a rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391:96–118, 2020. doi: 10.1016/j.neucom.2019.12.099.

Yonina C. Eldar, Patrick Kuppinger, and Helmut Bölcskei. Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Transactions on Signal Processing*, 58(6):3042–3054, 2010. doi: 10.1109/TSP.2010.2044837.

Abinand Gopal and Lloyd N. Trefethen. Solving Laplace problems with corner singularities via rational functions. *SIAM Journal on Numerical Analysis*, 57(5):2074–2094, 2019.

Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, and Jun Zhu. PINNacle: A comprehensive benchmark of physics-informed neural networks for solving PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 76721–76774, 2024. doi: 10.52202/079017-2442. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8c63299fb2820ef41cb05e2ff11836f5-Abstract-Datasets_and_Benchmarks_Track.html.

Wei Hu, Yi Cheng, and Chao Dong. Comment on "mix-training physics-informed neural networks for the rogue waves of nonlinear Schrödinger equation" [chaos, solitons and fractals 164 (2022) 112712]. *Chaos, Solitons & Fractals*, 200:117069, 2025. doi: 10.1016/j.chaos.2025.117069. URL https://github.com/wickweidr/ESA-PINNs-PT/tree/fe39356753cd87b7b566941ad1ce8f5de8fc6682. Reported numerical result from the authors' ESA-PINNs-PT repository.

Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.

Namgyu Kang, Jaemin Oh, Youngjoon Hong, and Eunbyung Park. PIG: Physics-informed gaussians as adaptive parametric mesh representations. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=y5B0ca4mjt.

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

Hyun Geun Lee, Youngjin Hwang, Yunjae Nam, Sangkwon Kim, and Junseok Kim. Benchmark problems for physics-informed neural networks: The Allen–Cahn equation. *AIMS Mathematics*, 10(3):7319–7338, 2025. doi: 10.3934/math.2025335.

Xin Li, Zhihong Xia, and Hongkun Zhang. Cauchy activation function and XNet. *Neural Networks*, 188:107375, 2025. doi: 10.1016/j.neunet.2025.107375.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence*, 6 (10):1256–1269, 2024. doi: 10.1038/s42256-024-00897-5.

H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, 8(1):164–177, 1996.

Yuji Nakatsukasa, Olivier Sète, and Lloyd N. Trefethen. The AAA algorithm for rational approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, 2018.

Kushagra Pandey, Jaideep Pathak, Yilun Xu, Stephan Mandt, Michael Pritchard, Arash Vahdat, and Morteza Mardani. Heavy-tailed diffusion models. In *International Conference on Learning Representations*, 2025.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, volume 1, pp. 40–44, 1993. doi: 10.1109/ACSSC.1993.342465.

D. H. Peregrine. Water waves, nonlinear Schrödinger equations and their solutions. *Journal of the Australian Mathematical Society. Series B. Applied Mathematics*, 25(1):16–43, 1983. doi: 10.1017/S0334270000003891.

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

Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in training PINNs: A loss landscape perspective. In *International Conference on Machine Learning*, 2024.

Pintu Samanta and Srinivasa Rao Chidella. Exact and asymptotic solutions of Burgers' equation on half-line. *The Quarterly Journal of Mechanics and Applied Mathematics*, 78(1):hbaf001, 2025. doi: 10.1093/qjmam/hbaf001.

Charles Soussen, Rémi Gribonval, Jérôme Idier, and Cédric Herzet. Joint k-step analysis of orthogonal matching pursuit and orthogonal least squares. *IEEE Transactions on Information Theory*, 59(5):3158–3174, 2013. doi: 10.1109/TIT.2013.2238606.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

O. Thual, U. Frisch, and M. Hénon. Application of pole decomposition to an equation governing the dynamics of wrinkled flame fronts. *Journal de Physique*, 46(9):1485–1494, 1985.

Juan Diego Toscano, Daniel T. Chen, Vivek Oommen, Jérôme Darbon, and George Em Karniadakis. A variational framework for residual-based adaptivity in neural PDE solvers and operator learning. *npj Artificial Intelligence*, 2:32, 2026. doi: 10.1038/s44387-026-00084-4.

Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

Honghui Wang, Lu Lu, Shiji Song, and Gao Huang. Learning specialized activation functions for physics-informed neural networks. *Communications in Computational Physics*, 34(4):869–906, 2023. doi: 10.4208/cicp.OA-2023-0058.

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

Yulun Wu, Miguel Aguiar, Karl H. Johansson, and Matthieu Barreau. Iterative training of physics-informed neural networks with fourier-enhanced features. In *International Conference on Learning Representations*, 2026.

A Theory

The method rests on two separate questions. The *right-feature* question asks why a Cauchy representation can be preferable to a finite tanh-derived or Gaussian representation for a rational, algebraically decaying solution. The *right-place* question asks how the residual can decide whether a particular representation–centre–scale basis element adds a direction that the current network does not already contain. Sections A.1 and A.2 give limited, explicit answers to these questions; Section A.3 relates them to frozen-feature approximation and states what they do not prove.

A.1 Right feature: pole geometry and algebraic tails

Dictionary formulas and shared coefficients. For the nonperiodized pairs, let $r = x - c$, $d > 0$, $w_T = d/\operatorname{arcosh}\sqrt{2}$, and $G(r) = \exp[-(\log 2)r^2/d^2]$. The implementation uses

class	ϕ_1	ϕ_2
Cauchy	$r/(r^2+d^2)$	$1/(r^2+d^2)$
tanh-derived	$\tanh(r/w_T)$	$\operatorname{sech}^2(r/w_T)$
Gaussian	$rG(r)$	$G(r)$

(8)

The parameter d is the half-width at half-height of the even channel. These pairs are used for Peregrine, Benjamin–Ono, Voigt, Burgers, and Allen–Cahn. For the last two, periodic boundary conditions are enforced through the residual rather than by periodizing each feature.

Periodic heat and Michelson–Sivashinsky instead use periodic pairs of period $L = 2\pi$. For Cauchy, put $\rho = e^{-2\pi d/L}$ and $z = \rho e^{2\pi \mathrm{i} r/L}$; the odd and even channels are respectively the imaginary and real parts of $(1+z)/(1-z)$. For tanh-derived features, set $z_n = (r + nL)/w_T$ and use

$$\phi_1 = \sum_{n=-J}^{J} \tanh z_n - \frac{2r}{L}, \qquad \phi_2 = \sum_{n=-J}^{J} \operatorname{sech}^2 z_n, \quad J = \lceil 20 w_T/L \rceil + 2. \tag{9}$$

The linear correction removes the mean slope of the periodized tanh primitive. For Gaussian features, set $w_G = d/\sqrt{2\log 2}$, $z_n = (r+nL)/w_G$, and $J = \lceil 20 w_G/L \rceil + 2$. The channels are $\sum_{n=-J}^{J} z_n e^{-z_n^2/2}$ and $\sum_{n=-J}^{J} e^{-z_n^2/2}$. Derivatives are evaluated from these formulas; finite image sums approximate the periodic features.

Every configuration also fits shared spatial functions: $\{1\}$ for the nonperiodized dictionaries and $\{1, \sin x, \cos x\}$ for the periodic dictionaries. These functions are always present and are excluded from neuron counts and the C/T/G totals. Including their Chebyshev time coefficients gives $(q + 2K_s)M$ fitted coefficients per real output channel, with $q = 1$ or 3. This yields, for example, $(1 + 2 \cdot 5)7 = 77$ for Benjamin–Ono and $(3 + 2 \cdot 8)7 = 133$ for periodic heat at the maximum retained sizes in Table 7. The Peregrine field has two real output channels.

Pole geometry and operator compatibility. For $r = x - c$ and $d > 0$,

$$\frac{1}{x - (c + \mathrm{i}d)} = \frac{r}{r^2 + d^2} + \mathrm{i}\frac{d}{r^2 + d^2}. \tag{10}$$

The two real Cauchy channels in (3) therefore encode a conjugate pole pair whose real-axis location and pole distance are (c, d). This gives every selected Cauchy neuron an interpretable geometry. It also matches the nonlocal Benjamin–Ono operator. With the convention $\widehat{Hf}(k) = -\mathrm{i}\,\mathrm{sign}(k)\widehat{f}(k)$,

$$H\left[\frac{d}{r^2 + d^2}\right] = \frac{r}{r^2 + d^2}, \qquad H\left[\frac{r}{r^2 + d^2}\right] = -\frac{d}{r^2 + d^2}, \qquad H(\partial_{xx} u) = \partial_{xx} H u. \tag{11}$$

Thus the Hilbert part of the residual exchanges the two channels before differentiation instead of introducing a different spatial representation. These identities motivate a Cauchy option in the dictionary; they do not say that every problem should select it.

A finite-budget tail mismatch. Suppose the matched far-field constant has been removed from a target profile and

$$f(x) = Ax^{-p} + o(|x|^{-p}), \qquad |x| \to \infty, \tag{12}$$

with $A \neq 0$ and $p > 0$. Any finite Gaussian sum decays exponentially. A finite tanh-derived sum also approaches zero exponentially after its limiting constant is matched and nondecaying components cancel. Hence either finite representation g satisfies

$$\lim_{|x| \to \infty} (1 + |x|^p)|f(x) - g(x)| = |A|. \tag{13}$$

The weighted criterion makes the leading algebraic tail visible even after the central profile has been fitted. Equation (13) is an asymptotic statement on the whole line, not a compact-domain error lower bound and not a contradiction of universal approximation. Together, (10)–(13) identify a regime in which the Cauchy representation carries useful structure per basis element; the residual still has to decide its location, scale, and necessity.

A.2 Right place: exact residual decrease for a linear problem

Let V be the current linear residual-response span, r the residual, and B the response columns of a candidate neuron. Write $Q = I - P_V$ and $B_\perp = QB$. Refitting over the enlarged span gives

$$\min_{v \in V,\, \beta} \|r - v - B\beta\|_2^2 = \|(I - P_V)r\|_2^2 - \|P_{B_\perp}(I - P_V)r\|_2^2. \tag{14}$$

The second term is exactly $S(B)$ in (6). It measures the decrease attributable to the candidate after the existing coefficients readjust. For a single candidate column b, it reduces to $|b^\top Q r|^2 / \|Qb\|_2^2$ when $Qb \neq 0$, the OLS selection criterion (Soussen et al., 2013); dependent candidates receive zero score. Projection onto the full residualized block span gives its block extension. Standard block OMP uses block correlations (Eldar et al., 2010) and need not make the same selection. For a linear PDE with fitted current coefficients, maximizing $S(B)$ therefore chooses the largest squared-residual decrease among the candidates scored. This is an established least-squares identity, applied here to PDE residual responses.

For a nonlinear PDE, the same identity applies only to the Gauss–Newton linearisation. CAPINN treats it as a proposal, performs the actual nonlinear coefficient refit, and applies the acceptance test (7). Independent residual validation then chooses among the accepted network sizes. The exact statement is consequently about linear residuals, not a global optimality guarantee for the nonlinear algorithm.

A.3 Frozen geometry and theoretical scope

Constructive approximation motivates separating hidden geometry from outer coefficients. An approximant has the form

$$f_N(x) = \sum_{k=1}^{N} a_k\, \phi(x; \theta_k), \tag{15}$$

where θ_k denotes neuron geometry and a_k the outer coefficients. With fixed geometry, fitting sampled function values is a linear least-squares problem in a_k. Choosing useful θ_k remains nonlinear, and fitting a nonlinear PDE residual generally also requires nonlinear coefficient updates. Cauchy neurons make this separation explicit because each $\theta_k = (c_k, d_k)$ specifies a pole location and distance to the real axis. The universal-approximation theorem for Cauchy networks (Theorem 2 of Li et al., 2025) is proved constructively in exactly this form: the proof places the poles on a target-independent set of locations and scales and then obtains the outer coefficients from the target, so the approximant it builds is a frozen-geometry Cauchy network with a linear readout. CAPINN keeps this structure and replaces the target-independent placement by residual-guided selection from a fixed grid.

CAPINN instantiates this separation algorithmically: it proposes (f_k, c_k, d_k) from a fixed multiscale dictionary, freezes accepted geometry, and jointly refits the outer coefficients. Unlike an existential universal-approximation argument, its ranking uses the equation residual without interior solution values. The approximation result explains why a basis with frozen geometry and a fitted linear readout can remain expressive; it does not prove convergence of the interval-wise nonlinear solver, certify an optimal dictionary, or guarantee recovery of the exact poles. Pole alignment in Section 4.2.2 is therefore an empirical post-hoc test of the selected representation rather than a theorem assumed by the method.

B Experimental details

B.1 Problem set and evaluation protocol

This appendix specifies the benchmark problems, the CAPINN configuration used for each, and the implementation of every baseline in Table 1. Section 4.1 introduces the eight configurations; here we state them formally and report, problem by problem, the settings, error maps, and baseline outcomes behind each table entry.

The roster has two parts. The first five configurations—the Peregrine rogue wave on a conventional and a wide box, the Benjamin–Ono soliton, Gaussian-to-Voigt half-heat evolution, and the Michelson–Sivashinsky equation—have solutions whose analytic structure consists of isolated complex poles or algebraic tails. They are the problems a Cauchy neuron is designed to represent. The last three—periodic heat, viscous Burgers, and Allen–Cahn—are controls: a smooth solution without finite singularities, and two established benchmarks dominated by steep fronts. They test whether the same construction generalizes when that structure is absent.

Evaluation. Every entry of Table 1 is the relative L^2 error $\|\hat{u} - u\|_2/\|u\|_2$ over the complete space–time scoring grid of its problem (Table 3). For conventional-box Peregrine the metric is the amplitude error $\||\hat{\psi}| - |\psi|\|_2/\||\psi|\|_2$ on the 151×257 time–space grid of the published comparison; complex fields otherwise use the full complex error. Scoring grids are separate evaluation grids, not the collocation or validation points. References are exact solutions where available and spectral solutions otherwise; the numerical-reference column of Table 1 reports the error of that spectral solver against the exact solution, or its coarse–fine difference when no exact solution exists. All computations use float64 or complex128.

Table 3: Scoring grids and references.

Problem	Scoring grid (space × time)	Reference	Numerical reference
Peregrine, conventional	512×201; 257×151 (amplitude)	exact	Chebyshev, 8.0e-9
Peregrine, wide	512×201	exact	Chebyshev, 6.3e-9
Benjamin–Ono	2048×56	exact	padded Fourier, 2.6e-6
Voigt	2048×65	exact	padded Fourier, 2.7e-6
Michelson–Sivashinsky	1024×65	Fourier spectral	coarse–fine, 1.7e-11
Periodic heat	1024×33	exact	8.2e-16
Burgers	2048×201	Fourier spectral	coarse–fine, 1.40e-6
Allen–Cahn	2048×201	Fourier spectral	coarse–fine, 3.71e-6

B.2 CAPINN HYPER-PARAMETERS AND TRAINING GUIDE

CAPINN divides the horizon into N equal time intervals. On each interval, the solution is a linear combination of analytic basis elements, each a representation–centre–width tuple contributing two spatial functions with Chebyshev time coefficients. At the start of an interval one element is fitted to the incoming state (the prescribed initial data on the first interval, the previous prediction afterwards). Further elements are then proposed from a candidate grid, refined locally, and accepted while they reduce the training residual. Accepted geometry is frozen; only output coefficients are fitted, by Gauss–Newton. A separate residual grid selects how many of the accepted elements to retain. Table 4 groups the hyper-parameters that shape the model; settings within a group act together.

Table 4: CAPINN hyper-parameters. Per-problem values are in Table 5.

Group	Settings	Role	Guidance
Time resolution	number of time intervals N; M Chebyshev modes on n_t time nodes per interval	time variation each interval can represent	An error floor on a smooth problem that more neurons do not lower indicates too coarse a time representation; increase M/n_t or N (Sections B.7 and B.8).
Spatial collocation	n_x spatial nodes	resolution at which the residual is enforced	Must resolve the narrowest feature being fitted; error that grows only after a front steepens indicates too few nodes (Section B.9).
Candidate dictionary	n_c centres $\times$ 7 geometrically spaced widths in $[w_{\min}, w_{\max}]$; 4 local refinement rounds	scales and positions available to each representation class	The width interval should span the narrowest expected feature up to the domain scale; for rational solutions, down to the smallest pole distance from the real axis.
Basis growth	growth tolerance ε; neuron cap K per interval	size of the constructed basis	Growth stops when the training residual norm falls below ε or at K neurons; on smooth problems it usually stops at ε, so a lower ε matters more than a larger K (Section B.8).
Interval coupling	entrance weight λ_e	continuity at the start of each interval versus the PDE residual	Error that accumulates across interval boundaries calls for a larger weight (Section B.10).

The remaining settings are fixed across all problems and are part of the released code: coefficients are solved by Gauss–Newton with at most 8 iterations and a relative singular-value cutoff of 10^{-10}; a neuron is accepted only if it lowers the squared training residual by at least 0.5%; the validation grid uses spatial nodes disjoint from training and one extra time node. Raising the iteration limit to 32 or lowering the cutoff to 10^{-12} did not change the fitted residual on the problems where we tested it.

Table 5: Per-problem CAPINN settings, grouped as in Table 4.

Problem	N	M/n_t	n_x	n_c	$[w_{\min}, w_{\max}]$	ε	K	λ_e
Peregrine, conventional	16	7/11	128	17	[0.31, 4.2]	10^{-6}	14	5
Peregrine, wide	24	7/11	512	33	[0.31, 4.2]	10^{-6}	10	5
Benjamin–Ono	55	7/11	512	65	[0.78125, 25]	10^{-6}	16	3
Voigt	16	7/11	512	65	[0.15625, 5]	10^{-6}	16	3
Michelson–Sivashinsky	32	7/11	512	12	[0.06, 3.84]	10^{-6}	20	3
Periodic heat	16	7/11	512	12	[0.06, 3.84]	10^{-8}	20	3
Burgers	20	6/8	769	101	[0.008, 1]	10^{-6}	32	3
Allen–Cahn	20	6/8	513	101	[0.008, 1]	10^{-6}	32	30

The problem subsections state the adjustments made for specific problems and the diagnostics behind them.

Replication. Given its configuration, a CAPINN run is deterministic: collocation grids and candidate grids are fixed, and no parameter is randomly initialized. We report one run per configuration. The only random generator in the pipeline, a fixed-seed post-hoc health check, does not affect fitting. Fit times are wall-clock times of the reported run with one BLAS thread. Runs shared hosts with concurrent jobs, so times indicate cost but do not rank methods (Table 6).

Table 6: Wall time of the reported solutions. CAPINN: fitting time of the Table 1 run, one BLAS thread. Baseline: the best in-house baseline shown in the problem's figure; "(cap)" marks a fit that ran to its wall-clock budget.

Problem	CAPINN	Host	Best in-house baseline	Wall time	Host
Peregrine, conventional	113 s	A	Vanilla PINN	7200 s (cap)	GPU
Peregrine, wide	1142 s	A	PIG	1691 s	GPU
Benjamin–Ono	2103 s	B	Vanilla PINN	3300 s (cap)	GPU
Voigt	2398 s	A	PIG	2633 s	GPU
Michelson–Sivashinsky	1445 s[a]	A	Vanilla PINN	3600 s (cap)	B
Periodic heat	119 s	A	Frozen-PINN	0.8 s per seed	A
Burgers	9116 s	A	Frozen-PINN	3.3 s	A
Allen–Cahn	7615 s	A	— (published value)	—	—

Hosts: A = Intel Core i7-11700K; B = Intel Core Ultra 7 265K; GPU = NVIDIA RTX 4080 SUPER.
[a] Includes an interrupted first attempt; the run was resumed from its saved intervals.

B.3 Baseline implementations

Reporting principle. Each baseline starts from the available published code and reported settings, without target-specific searches. If a method does not support a problem feature, we mark the entry n/a; if it does not converge under its recommended settings, we report the outcome (†) and its reason. Baseline architectures follow the available implementations; equation-specific adaptations, sampling changes, and stopping budgets are stated below. Loss evaluation in memory-sized chunks preserves the objective. In-house baselines are single-seed float64 fits, except Frozen-PINN on periodic heat (five seeds). An independent check reproduced every saved error from the saved model and verified the equation, data, and derivatives.

Vanilla PINN (Raissi et al., 2019). A float64 PyTorch port of the authors' TensorFlow code: tanh networks with 8×20 hidden units (4×100 with two outputs for the nonlinear Schrödinger equation, as in the source), L-BFGS-B with the source tolerances, and Adam first where the source uses it. The same prescribed initial and boundary data as CAPINN are imposed as penalties; the Peregrine networks add the plane-wave background to their output, while the main CAPINN runs fit the full field (Section B.4). Budgets are wall-clock caps of 3300–7200 s (problem subsections).

PIG (Kang et al., 2025). The authors' released two-dimensional recipe on every problem: four feature channels of 800 diagonal Gaussians (3,200 Gaussians), initial width 0.1, and a $4 \to 16 \to$output tanh decoder (16,097 parameters; 16,114 with two outputs). Coordinates are normalized to $[-1, 1]$. Training uses Torch L-BFGS with strong-Wolfe line search for at most 2001 outer iterations and one hour, on fixed collocation sites instead of the source's per-iteration resampling. The Helmholtz-specific PDE weight of the source is not transferred.

Frozen-PINN (Datar et al., 2026). The released solvers. The published formulation (Eq. 1 of that paper) covers real scalar fields with linear spatial-derivative operators. We use the released diffusion and Burgers solvers; the Allen–Cahn entry adapts the released periodic nonlinear example and has the structural limitations stated in Section B.10. Peregrine (complex field) and Benjamin–Ono, Voigt, and Michelson–Sivashinsky (nonlocal operators) fall outside it and are marked n/a. Frozen-feature extensions to these problems were evaluated separately under pre-registered eligibility gates; none passed, and they are not reported as results of the published method. The reasons are given per problem.

Published values are taken from the cited papers for the same equation, domain, horizon, and metric. Their scoring grids, seeds, and budgets follow the source and can differ from ours.

Table 7: Model size. CAPINN: maximum retained analytic neurons and fitted real coefficients per interval, including shared spatial functions, and number of intervals.

Problem	CAPINN			Vanilla PINN	PIG	Frozen-PINN
	neurons	coeff.	intervals	hidden layers	Gaussians	features
Peregrine, conventional	9	266	16	4×100	4×800	n/a
Peregrine, wide	8	238	24	4×100	4×800	n/a
Benjamin–Ono	5	77	55	8×20	4×800	n/a
Voigt	16	231	16	8×20	4×800	n/a
Michelson–Sivashinsky	20	301	32	8×20	4×800	n/a
Periodic heat	8	133	16	8×20	4×800	1024 inner / 256 outer
Burgers	32	390	20	8×20	4×800	550
Allen–Cahn	32	390	20	8×20	4×800	2000 inner / 100 outer

Figures. Each problem subsection shows one row of five panels on the scoring grid: the reference, the CAPINN prediction and absolute error, and the prediction and absolute error of the best in-house baseline (lowest error among our vanilla PINN, PIG, and Frozen-PINN runs). Field panels share the reference colour range. The baseline error panel uses a linear scale capped at its 99.5th percentile; an arrow marks saturation. Where CAPINN is more accurate, the CAPINN error panel uses a logarithmic scale from its own 1st percentile up to that same baseline maximum, so its colours indicate how much smaller its errors are while its pattern stays visible. For Burgers, where the baseline is more accurate, both error panels use their own linear scales. Allen–Cahn shows CAPINN only, because Table 1 reports the published PIG value rather than an in-house replication.

B.4 Peregrine rogue wave

Problem. The focusing nonlinear Schrödinger equation $i\psi_t + \frac{1}{2}\psi_{xx} + |\psi|^2\psi = 0$ with the Peregrine solution (Peregrine, 1983)

$$\psi(x,t) = e^{it}\left(1 - \frac{4(1+2it)}{1+4x^2+4t^2}\right),$$

on the conventional box $x \in [-2,2]$, $t \in [-2,2]$ and the wide box $x \in [-20,20]$, $t \in [-3,3]$. The initial trace and the Dirichlet traces at both spatial boundaries are prescribed from the exact solution; interior values are never supplied.

Analytic structure. For fixed t, ψ has simple poles at $x = \pm i\delta(t)$ with $\delta(t) = \frac{1}{2}\sqrt{1+4t^2}$, closest to the real axis at the focusing time $t = 0$. Away from the core, $|\psi|$ approaches the unit background algebraically, as x^{-2}. The wide box exposes this algebraic approach over a larger range.

Configuration. Table 5. No background term is supplied: CAPINN fits the full complex field. An earlier configuration added the exact plane wave e^{it} and fitted only the correction; removing it changes the error by less than 0.002% (conventional box, amplitude $5.7658665 \times 10^{-7} \to 5.7659186 \times 10^{-7}$; wide box $4.2290426 \times 10^{-7} \to 4.2291146 \times 10^{-7}$), because the constant spatial column with Chebyshev time coefficients represents the background. Fit times are 113 s (conventional) and 1142 s (wide).

Results. Conventional box: amplitude error 5.77×10^{-7} (whole field 8.84×10^{-7} on the 512×201 grid). Wide box: 4.23×10^{-7}. The analytic-representation ablation below uses the same configuration and cap; these runs retain the supplied background, identically for every representation class.

Box	tanh	Gaussian	Cauchy	tanh + Gaussian	Mixed	C/T/G
Conventional	9.71e-7	4.89e-5	5.83e-7	8.77e-7	5.77e-7	73/8/3
Wide	1.22e-3	3.24e-2	4.13e-7	2.08e-3	4.23e-7	109/3/3

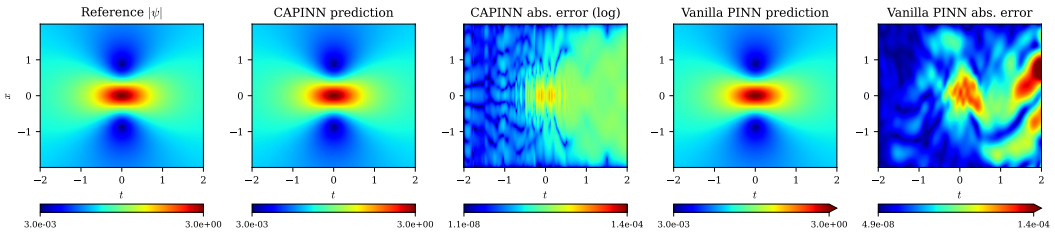

Figure 4: Peregrine, conventional box: reference amplitude $|\psi|$, CAPINN and vanilla PINN predictions and complex-field errors. Amplitude relative L^2 error: CAPINN 5.77×10^{-7}, vanilla PINN 2.32×10^{-5} ($40\times$ larger).

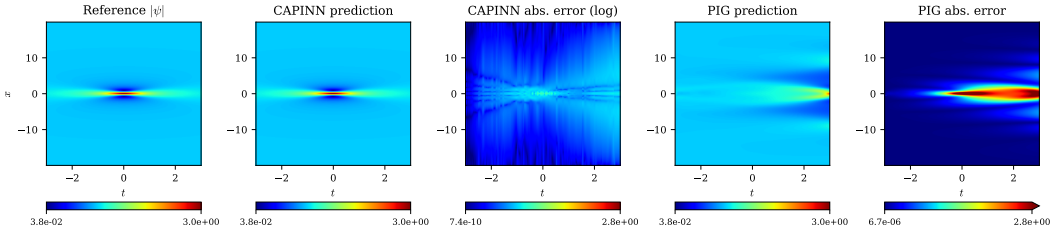

Figure 5: Peregrine, wide box: reference amplitude, CAPINN and PIG predictions and complex-field errors. Relative L^2 error: CAPINN 4.23×10^{-7}, PIG 5.20×10^{-1} (not converged).

Baselines.

- *Vanilla PINN.* 4×100 network with 513 initial sites, 129 boundary times, and 20,000 Latin-hypercube residual sites. Conventional box: 50,000 Adam steps, then 31,813 L-BFGS-B iterations within the 7200 s cap, giving amplitude error 2.32×10^{-5}. Wide box: 33,626 L-BFGS-B iterations after Adam, reaching $5.66 \times 10^{-1\dagger}$; the fit misses the focusing event, with peak amplitude 1.18 at $t = 0$ against the exact value 3.
- *PIG.* Conventional box: the optimizer stalls after 526 s at 2.84×10^{-3}; the field at the focus is $-2.999 - 0.004i$ against the exact -3. Wide box: the optimizer stalls after 1691 s at $5.20 \times 10^{-1\dagger}$; the focus value is $0.73 + 0.83i$, and a narrow temporal Gaussian produces large residuals between collocation sites.
- *Frozen-PINN.* n/a: complex field. A frozen-feature extension for the coupled real and imaginary parts passed its initial-projection check, but its error changed by 18% when the spatial domain was doubled, failing the pre-registered robustness gate.
- *Published.* ESA-PINNs-PT reports $(2.48 \pm 0.38) \times 10^{-5}$ amplitude error on the same box and grid (Hu et al., 2025): a mean over eight float32 runs of a network that learns the full field.

B.5 Benjamin–Ono soliton

Problem. The Benjamin–Ono equation $u_t + uu_x - \mathcal{H}u_{xx} = 0$, where $\mathcal{H}$ is the Hilbert transform, with the travelling rational soliton

$$u(x,t) = \frac{4c}{1 + c^2(x - x_0 - ct)^2}, \qquad c = 0.3,\ x_0 = 0,$$

on $x \in [-100, 100]$, $t \in [0, 110]$. The initial trace is prescribed.

Analytic structure. For fixed t, u has simple poles at $x = x_0 + ct \pm i/c$ and decays algebraically, as x^{-2}. The soliton travels 33 length units over the horizon, so the peak and its tails must be transported over 55 intervals. Because $\mathcal{H}$ is nonlocal, the operator is evaluated on the whole line: analytically for Cauchy neurons, and for tanh and Gaussian neurons by a 65,536-node FFT on $[-800, 800)$ with cubic interpolation to the residual sites.

Configuration. Table 5. Fit time 2103 s.

Results. 4.22×10^{-7}. Mixed retains only Cauchy elements.

tanh	Gaussian	Cauchy	tanh + Gaussian	Mixed	C/T/G
4.52e-4	2.55e-3	4.22e-7	5.93e-4	4.22e-7	275/0/0

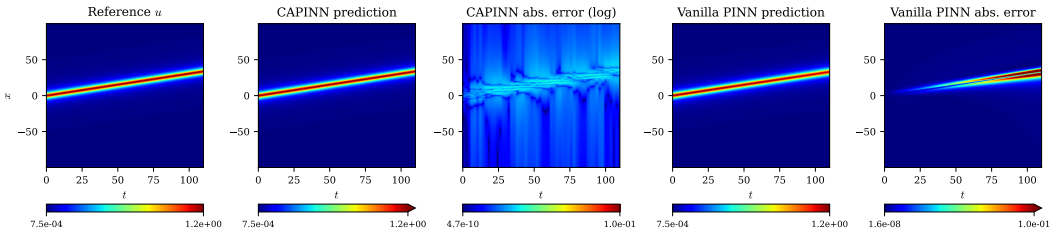

Figure 6: Benjamin–Ono: reference, CAPINN and vanilla PINN predictions and errors. Relative L^2 error: CAPINN 4.22×10^{-7}, vanilla PINN 6.97×10^{-2}.

Baselines.

- *Vanilla PINN.* 8×20 network, 1,025 initial sites (513 in the box, 512 beyond it), 1,485 spatial residual sites over 129 times, soft zero-boundary penalties, and a 3300 s cap. Error 6.97×10^{-2}, with final-time error 0.137; this is a capped finite-box approximation.
- *PIG.* The optimizer stalls after 2230 s at $7.21 \times 10^{-1\dagger}$. The fitted soliton decays and slows: at $t = 110$ its peak is 0.33 at $x = 13.3$, against 1.2 at $x = 33$.
- *Frozen-PINN.* n/a: nonlocal operator. A frozen-feature extension did not meet the pre-registered initial-projection threshold in physical coordinates.
- *Published.* No matched published result.

B.6 Gaussian-to-Voigt evolution

Problem. The half-heat equation on the whole line, $u_t = -\gamma(-\partial_{xx})^{1/2}u$ with $\gamma = 1$, from Gaussian initial data with standard deviation $\sigma = 0.5$, evaluated on the box $x \in [-20, 20]$ for $t \in [0, 2]$. The solution is the Voigt profile

$$u(x,t) = (G_\sigma * P_{\gamma t})(x) = \frac{\operatorname{Re}\, w(z)}{\sigma\sqrt{2\pi}}, \qquad z = \frac{x + i\gamma t}{\sigma\sqrt{2}},$$

where $P_{\gamma t}$ is the Cauchy kernel and w the Faddeeva function.

Analytic structure. A Gaussian core coexists with algebraic tails $u \sim \gamma t/(\pi x^2)$ for $|x| \to \infty$ and $t > 0$. The tails emerge immediately and carry a growing share of the mass. The operator is evaluated as for Benjamin–Ono.

Configuration. Table 5. Fit time 2398 s. A check with fourfold padding and eightfold nodes on the short-horizon fits changed $\gamma|D|u$ by less than 5×10^{-7}.

Results. 6.74×10^{-6}. Mixed combines all three representation classes and is $7.7\times$ more accurate than the best single-class configuration. On the shorter horizon $t \in [0, 0.5]$ the factor is $14.7\times$.

Horizon	tanh	Gaussian	Cauchy	tanh + Gaussian	Mixed	C/T/G
$t \in [0, 2]$	2.65e-4	8.54e-4	5.19e-5	2.32e-4	6.74e-6	85/31/87
$t \in [0, 0.5]$	7.49e-5	4.76e-4	6.36e-5	5.02e-5	4.34e-6	23/15/20

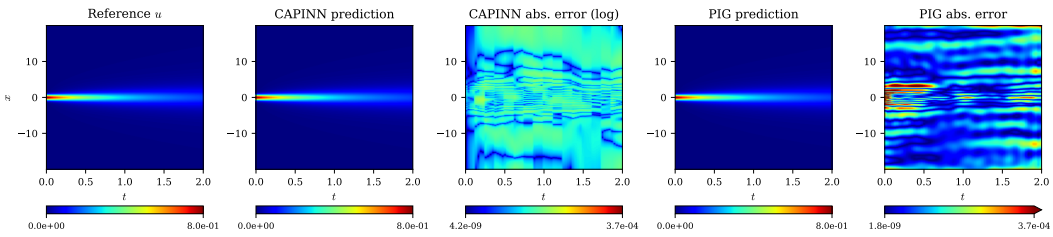

Figure 7: Voigt: reference, CAPINN and PIG predictions and errors. Relative L^2 error: CAPINN 6.74×10^{-6}, PIG 1.57×10^{-3} (230× larger).

Baselines.

- *Vanilla PINN.* 8×20 network, 1,367 spatial residual sites, and a 3600 s cap. Error 3.78×10^{-3}, a capped finite-box approximation with far-boundary discrepancy 9.2×10^{-5}.
- *PIG.* 1.57×10^{-3}. The optimizer stops after 2633 s, and the gradient at the stop is not stationary. Accuracy is limited by the initial-data fit (error 1.7×10^{-3}) and by the algebraic tail near the box edge. The first attempt exceeded GPU memory before its first evaluation; the reported run evaluates the identical objective in chunks of eight time slices.
- *Frozen-PINN.* n/a: nonlocal operator. No frozen-feature extension candidate passed the pre-registered initial-projection gate.
- *Published.* No matched published result.

B.7 Michelson–Sivashinsky equation

Problem. The slope form of the Michelson–Sivashinsky equation, $u_t + uu_x = \nu u_{xx} + |D|u$ with $\nu = 0.12$, where $|D|$ has Fourier multiplier $|k|$, on the periodic domain $x \in [-\pi, \pi]$ with $t \in [0, 4]$ and $u(x, 0) = 0.35 \sin x + 0.2 \sin(2x + 0.7) - 0.1 \cos 3x$.

Analytic structure. The equation admits exact pole-decomposition solutions, in which the slope is a finite sum of contributions from complex-conjugate poles whose positions evolve by a closed system of ordinary differential equations (Thual et al., 1985). Our initial data are trigonometric, not of pole form, so no closed-form solution is used. The reference is a Fourier spectral solution; a refined Fourier ETDRK4 solve with 2048 modes and $\Delta t = 3.125 \times 10^{-5}$ agrees with it to 1.7×10^{-11}.

Configuration. Table 5. Fit time 1445 s, including an interrupted first attempt.

Tuning. With the initial $M/n_t = 5/7$, the error plateaus at 3.07×10^{-2}. Increasing the time resolution to $7/11$ reduces it to 2.01×10^{-3} (15.3×). This change was applied to the Mixed configuration only, after inspecting the result; the single-class ablation below uses the initial setting.

Results. 2.01×10^{-3}, the least accurate CAPINN case. The remaining error is dominated by a front crossing the periodic boundary, which all representation classes share. At the initial setting the classes are close; Cauchy alone is most accurate.

tanh	Gaussian	Cauchy	tanh + Gaussian	Mixed	C/T/G
2.94e-2	2.96e-2	2.72e-2	3.60e-2	3.07e-2	252/160/184

Baselines.

- *Vanilla PINN.* 8×20 network with periodic value and derivative penalties, 10,000 Adam steps (phase cap 1800 s) followed by L-BFGS-B, and a 3600 s cap. Error $8.45 \times 10^{-1\dagger}$; the PDE residual on unused sites has RMS 0.128.

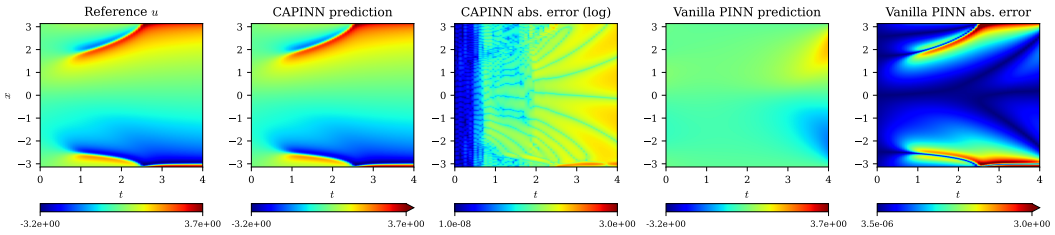

Figure 8: Michelson–Sivashinsky: reference, CAPINN and vanilla PINN predictions and errors. Relative L^2 error: CAPINN 2.01×10^{-3}, vanilla PINN 8.45×10^{-1} (not converged).

- *PIG.* Reaches the 3600 s cap at $9.88 \times 10^{-1\dagger}$, with the optimizer stalled; the PDE residual on unused sites has RMS 0.138.
- *Frozen-PINN.* n/a: nonlocal operator. In a frozen-feature extension, 10 of 15 configurations integrated to $t = 4$ and 5 stopped with step-size failures; the completed runs did not pass the PDE-residual and periodic-derivative validation.
- *Published.* No matched published result for a single-instance solve.

B.8 Periodic heat

Problem. $u_t = 0.02\, u_{xx}$ on the periodic domain $x \in [-\pi, \pi]$, $t \in [0, 2]$, from a periodized normalized Gaussian with centre 0.37 and standard deviation 0.12.

Analytic structure. The Fourier coefficients of the solution decay like $\exp(-(\sigma^2/2 + 0.02\, t)k^2)$, so the solution is entire in x, with no finite singularities. It serves as a smooth control, in which Gaussian neurons are expected to suffice. The reference evolves the Fourier modes exactly.

Configuration. Table 5. Fit time 119 s.

Tuning. With the initial settings $M/n_t = 5/7$ and $\varepsilon = 10^{-6}$, the error is 3.23×10^{-7}. Increasing the time resolution to $7/11$ and lowering the growth tolerance to 10^{-8} reduce it to 4.87×10^{-9} ($66\times$). As for Michelson–Sivashinsky, this applies to the Mixed configuration only; the ablation below uses the initial settings.

Results. 4.87×10^{-9}. CAPINN selects only Gaussian elements (0/0/91 at the tuned setting) and matches the Gaussian-only error at the initial setting.

tanh	Gaussian	Cauchy	tanh + Gaussian	Mixed	C/T/G
9.47e-7	3.23e-7	2.85e-5	3.23e-7	3.23e-7	0/0/72

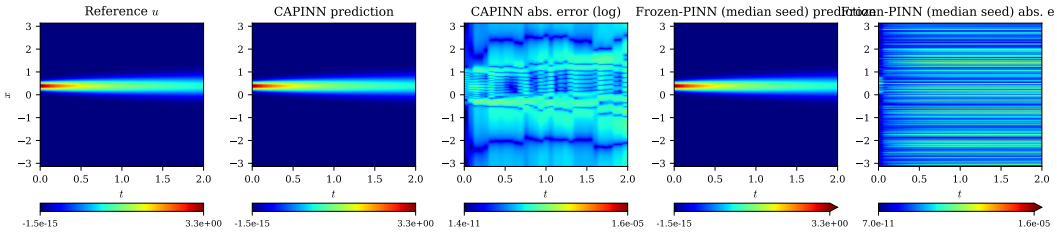

Figure 9: Periodic heat: reference, CAPINN and Frozen-PINN (median of five seeds) predictions and errors. Relative L^2 error: CAPINN 4.87×10^{-9}, Frozen-PINN 1.2×10^{-5} for the seed shown (2.1×10^{-5} mean over five seeds).

Baselines.

- *Vanilla PINN.* 8×20 network, 512 initial and residual sites over 129 times, periodic value and derivative penalties, and a 3600 s L-BFGS-B cap. Error 4.69×10^{-5}.
- *PIG.* The optimizer stalls after 1407 s at 1.85×10^{-4}.
- *Frozen-PINN.* The released diffusion solver with 1024 inner and 256 outer features, 512 construction points, ODE tolerance 10^{-8}, and SVD cutoff 10^{-10}. The solver's unit diffusion is mapped exactly by the time scaling $s = 0.02t$. This configuration was fixed before scoring on our grid. Error $(2.1 \pm 1.7) \times 10^{-5}$ over seeds 1–5, about 0.8 s per fit.
- *Published.* No matched published result.

B.9 Viscous Burgers equation

Problem. $u_t + uu_x = (0.01/\pi)u_{xx}$ on $x \in [-1, 1]$, $t \in [0, 1]$, with $u(x, 0) = -\sin \pi x$. We solve it on the periodic domain; the solution is odd, so it satisfies the homogeneous Dirichlet conditions used in the published comparisons.

Properties. A steep front forms at $x = 0$ near $t = 1/\pi \approx 0.32$, the breaking time of the inviscid problem; its width then scales with the viscosity. The reference is a Fourier ETDRK4 solution; the coarse–fine difference between 1024 and 2048 modes is 1.40×10^{-6}.

Configuration. Table 5. Fit time 9116 s.

Tuning. With the initial $n_x = 513$, the error is 3.94×10^{-5} and grows after the front forms. Two diagnostics narrowed down the cause. First, the width grid already contains widths down to 0.008, so the front width was available. Second, in a four-interval diagnostic at 1026 nodes, enforcing exact continuity at interval entrances increased the validation residual by a geometric mean of $5.2\times$, so the weighted continuity treatment was kept. Increasing the spatial collocation to $n_x = 769$ for every representation configuration reduces the error to 3.74×10^{-6} ($10.5\times$). A small shared sawtooth near $t \approx 0.3$ remains.

Results. 3.74×10^{-6}. The full mixture is most accurate among the representation configurations. With tanh and Gaussian only, growth stops at interval 6 after a candidate fails the acceptance test; that error reflects the stalled growth rather than the excluded representation class alone.

tanh	Gaussian	Cauchy	tanh + Gaussian	Mixed	C/T/G
9.29e-6	1.81e-3	1.24e-5	3.22e-2	3.74e-6	212/135/163

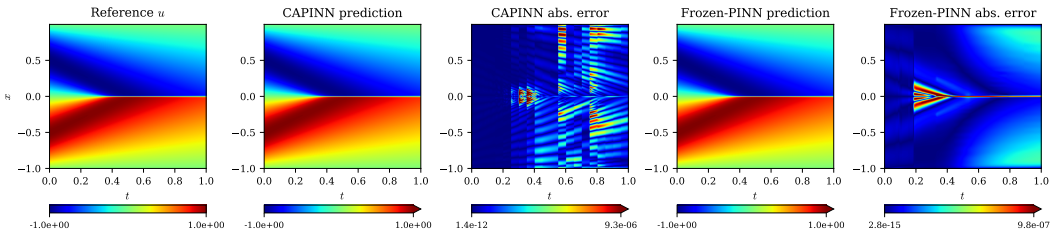

Figure 10: Burgers: reference, CAPINN and Frozen-PINN predictions and errors. Relative L^2 error: CAPINN 3.74×10^{-6}, Frozen-PINN 3.43×10^{-7} ($10.9\times$ smaller than CAPINN).

Baselines.

- *Vanilla PINN.* The authors' Burgers recipe: 8×20 network, 100 initial and boundary observations, and 10,000 Latin-hypercube residual sites plus 456 initial and boundary sites. L-BFGS-B stops on the source gradient tolerance after 262 s at 9.91×10^{-4}.

- *Frozen-PINN.* The released Burgers solver and accuracy configuration: width 550, nine time blocks, 4,000 initial and 1,000 later interior points, 20,000 resampling points, and LSODA with tolerance 10^{-6}. It reaches 3.43×10^{-7} in 3.29 s, $10.9\times$ lower error than CAPINN. The authors report $(2.27 \pm 0.69) \times 10^{-7}$.
- *PIG.* Reaches the 3600 s cap at $1.15 \times 10^{-1\dagger}$; initial, boundary, and PDE errors remain substantial.
- *Published.* vRBA with SSBroyden reports 8.25×10^{-9} on the authors' grid and budget (Toscano et al., 2026). This is below our reference's coarse–fine difference, so our scoring cannot rank methods at that level.

B.10 Allen–Cahn equation

Problem. $u_t = 10^{-4}u_{xx} + 5(u - u^3)$ on the periodic domain $x \in [-1, 1]$, $t \in [0, 1]$, with $u(x, 0) = x^2 \cos \pi x$.

Properties. The reaction drives the solution toward the stable states ± 1. Interfaces sharpen to width of order $\sqrt{10^{-4}/5} \approx 0.0045$ and then evolve slowly. The initial data are continuous but not smooth under periodic extension: $u_x(\pm 1, 0) = \mp 2$. The reference is a Fourier ETDRK4 solution; the coarse–fine difference between 1024 and 2048 modes is 3.71×10^{-6}.

Configuration. Table 5. Fit time 7615 s.

Tuning. With the initial entrance weight $\lambda_e = 3$, the error is 3.55×10^{-4} and accumulates across interval boundaries. Raising λ_e to 30 for every representation configuration reduces it to 4.09×10^{-5} ($8.7\times$). Localized error at the periodic seam at early times, caused by the derivative jump in the initial data, remains.

Results. 4.09×10^{-5}. Here the configuration without the Cauchy representation is most accurate, at 2.24×10^{-5}: the problem tests interface resolution and serves as a counterexample to a uniform benefit from Cauchy enrichment.

tanh	Gaussian	Cauchy	tanh + Gaussian	Mixed	C/T/G
3.81e-5	3.22e-4	2.08e-4	2.24e-5	4.09e-5	180/217/212

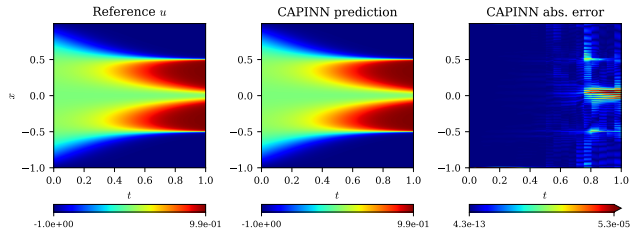

Figure 11: Allen–Cahn: reference, CAPINN prediction and error. The error scale is linear and capped at the 99.5th percentile; the saturated values are the early-time seam spikes near $x = \pm 1$. Relative L^2 error: CAPINN 4.09×10^{-5}; published PIG 1.04×10^{-4} and vRBA 1.88×10^{-6}.

Baselines.

- *Vanilla PINN.* 8×20 network with periodic penalties and a 3600 s cap (1,930 Adam steps, then 1,619 L-BFGS-B iterations). Error $9.92 \times 10^{-1\dagger}$.
- *PIG.* Table 1 reports the authors' published $(1.04 \pm 0.41) \times 10^{-4}$ over three seeds for exactly this problem (Kang et al., 2025, Table 1). Their Allen–Cahn recipe (4,000 Gaussians with feature dimension one, Adam, a causal loss, and NTK-based loss balancing) is not included in the released code, and several of its settings are unstated. Our replication with

the released two-dimensional recipe stalls after 3279 s at $9.91 \times 10^{-1\dagger}$; Gaussian widths collapse and produce narrow residual spikes between collocation sites.

- *Frozen-PINN.* No released Allen–Cahn solver exists. Applying the released periodic nonlinear construction (the Kuramoto–Sivashinsky example) with the Allen–Cahn right-hand side gives $2.07^{\dagger}$ on all five seeds, for two independent reasons: the periodic outer basis contains no constant mode, while u_0 has mean $-2/\pi^2$; and the example's initial fit also matches the derivatives u_x, u_{xx}, and u_{xxxx}, while for this u_0, u_x jumps from $+2$ to -2 across the periodic boundary.
- *Published.* vRBA with SSBroyden reports 1.88×10^{-6} on the authors' grid and budget (Toscano et al., 2026); our reference's coarse–fine difference (3.71×10^{-6}) is of the same order.

C STUDENT-T FLOW: OUT-OF-DOMAIN EVIDENCE

We apply CAPINN to Student-t Flow (Pandey et al., 2025), the heavy-tailed variant of flow matching (Lipman et al., 2023), to test residual-guided representation construction beyond PDE solving. The path is $x_t = tx_1 + (1-t)n$, with source noise $n \sim t_3(0, I_2)$ and a four-mode, heavy-tailed Student-t_3 target. Every configuration has one forced affine neuron and at most four analytic neurons from the same fixed dictionary: centres on a 5×5 grid, endpoint widths $\{0.45, 0.9, \sqrt{3}\}$, and fixed trajectories tc. In this two-dimensional setting each analytic neuron has three spatial functions (a radial profile and two directional components; for the tanh-derived class, $\tanh r_1$, $\tanh r_2$, and their product), four temporal polynomials, and two output dimensions, hence 24 output coefficients; the affine neuron also has 24, giving every pure and mixed dictionary the same 120-coefficient cap. CAPINN scores neurons from the source-noise residual on a selection split, freezes accepted geometry, refits coefficients on fit data by ridge-regularized least squares (ridge 10^{-7}, scaled by the mean diagonal of the Gram matrix), and chooses model size by validation noise MSE. The 12,000 paired samples are split 60/15/15/10 into fit, selection, validation, and independent audit sets; generated metrics use a common 80-step RK4 sampler and are never used for selection.

An exploratory five-seed pilot (seeds 20261560–20261564) found the same direction in four paired runs, although pure Cauchy had the lowest mean SW1. It motivated a 20-seed confirmation whose protocol, seeds, primary comparison, and success criteria were fixed before it was run (included in the supplementary material); the pilot is not pooled with the confirmation. On the 20 new paired seeds, Mixed CAPINN retains a Cauchy element in every run and lowers generated sliced Wasserstein-1 (SW1) relative to the otherwise identical mixed dictionary without the Cauchy representation in 18 of them. The paired difference, defined as no-Cauchy SW1 minus Mixed CAPINN SW1, is 0.0574 ± 0.0374 (mean $\pm$ sample standard deviation); its 95% bootstrap interval is $[0.0420, 0.0740]$, and an exact one-sided sign-flip test gives $p = 3.8 \times 10^{-6}$. This controlled comparison demonstrates the contribution of Cauchy enrichment to the selected velocity basis in this fixed-budget transport task.

The single-class rows provide descriptive context rather than a second registered comparison. Mixed has the lowest mean SW1, but pure Cauchy's mean includes one high-error seed (0.853); its median SW1 is 0.171, close to Mixed's 0.163, and Mixed is lower than pure Cauchy in 12 of the 20 paired seeds. Both tail audits favour Mixed on average, but neither was registered as a hypothesis test and both have substantial variation. The registered comparison establishes the gain from Cauchy enrichment; the single-class and tail results characterize its behaviour across seeds.

Figure 12 visualizes the analytic elements retained by residual validation for the 20 Mixed runs. The affine element is forced and omitted from the display. Cauchy is retained in every run and is the first analytic choice in 19 of 20 runs; tanh-derived and Gaussian elements then complement the residual when validation keeps them. Empty cells indicate proposed additions excluded when validation retained a smaller model. Residual-guided fitting determines this representation allocation, with generated metrics reserved for independent evaluation.

Table 8: Confirmatory Student-t Flow audit over 20 new paired seeds. Every configuration uses the same fixed dictionary construction, forced affine neuron, and 120-coefficient cap; residual validation chooses model size. SW1 is the registered primary generated-distribution metric. The two tail columns are descriptive held-out audits, not registered hypothesis tests. Values are mean ± sample standard deviation.

Method	coeffs	SW1 ↓	$q_{99.5}$ rel. err.	tail mass err.	paired result
Pure Cauchy	111.6 ± 22.4	$.2188 \pm .1716$	$.1207 \pm .2822$	$.00451 \pm .01186$	—
Mixed CAPINN	90.0 ± 23.2	$.1851 \pm .0986$	$.0829 \pm .0660$	$.00228 \pm .00187$	18/20; $p = 3.8\text{e}{-6}$
Mixed without Cauchy	93.6 ± 26.9	$.2426 \pm .0890$	$.1533 \pm .2293$	$.00308 \pm .00490$	—
Pure Gaussian	110.4 ± 22.6	$.2263 \pm .0530$	$.1199 \pm .1056$	$.00238 \pm .00182$	—
Pure tanh-derived	98.4 ± 25.7	$.3078 \pm .1637$	$.1699 \pm .3662$	$.00762 \pm .01440$	—

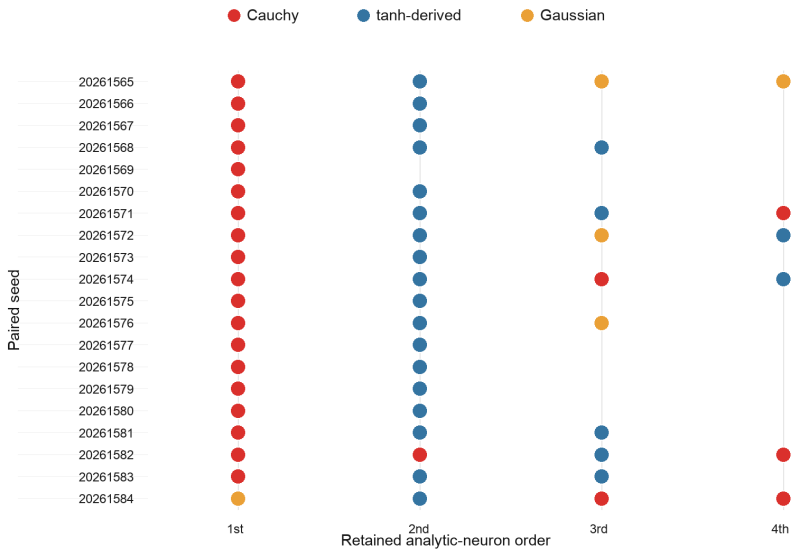

Figure 12: Frozen analytic neurons retained by residual validation in the 20 confirmatory Mixed Student-t Flow runs. Each row lists retained neurons after the forced affine neuron; empty cells are candidates excluded when validation chose a shorter prefix. Cauchy is retained in every run and selected first in 19 of 20.