

D_4 -PINN: Hard-constraint group-invariant physics-informed neural networks with aerospace thermal-analysis applications

Chuyao Gong^{a,b}

^aDalian University of Technology, No. 2 Dagongdalu, Liaodong Bay New District, Panjin, 124221, China

^bUniversity of Leicester, University Road, Leicester, LE1 7RH, United Kingdom

ARTICLE INFO

Keywords:

Physics-informed neural networks
Group invariance
Dihedral symmetry D_4
Reynolds operator
Semilinear elliptic equations
Allen–Cahn equation
Algebraic invariants
Reproducible scientific computing

ABSTRACT

D_4 -PINN enforces dihedral- D_4 symmetry of the solution at machine precision ($\sim 10^{-8}$) by construction, rather than through a soft penalty in the loss. The network is the Reynolds average of a free multilayer perceptron over the eight-element D_4 orbit of the input; the resulting forward map satisfies $u_\theta(g \cdot \mathbf{x}) = u_\theta(\mathbf{x})$ for every $g \in D_4$ regardless of training. The core contribution is a *certified* symmetry guarantee—not an accuracy or efficiency claim—whose value is demonstrated on steady-state thermal analysis of D_4 -symmetric aerospace composite panels.

A generalization analysis based on the Mishra–Molinero PINN total-error decomposition and the quotient-space Rademacher complexity construction of Sannai, Imaizumi, and Kawano quantifies the sample-complexity benefit of the D_4 -invariant hypothesis class. An intermediate lemma establishes that the Rademacher complexity of the PDE residual under the Laplacian is bounded by a constant multiple of that of the hypothesis class, with the amplification factor polynomially controlled in the network depth and spectral-norm bound.

D_4 -PINN is benchmarked against the unconstrained PINN, an algebraic-invariant PINN using the D_4 fundamental invariants $I_1 = x^2 + y^2$ and $I_2 = x^2y^2$ as input, two equivariant architectures, and a finite element method. Under equalised wall-clock time, the $\sim 2.8\times$ per-epoch overhead of Reynolds averaging offsets the sample-efficiency gain from the invariant hypothesis class; on smooth forward problems the two architectures achieve comparable accuracy with neither holding a systematic advantage. D_4 -PINN's accuracy advantage is task-dependent: $5.5\times$ lower L^2 error on cubic semilinear problems, $4.1\times$ lower parameter recovery error on inverse problems, and a 2.2 – $3.5\times$ advantage on the Allen–Cahn equation across $\epsilon \in [0.02, 0.10]$. The machine-precision D_4 -invariance guarantee ($\sim 10^{-8}$), in contrast, is unconditional—it holds across all seeds, architectures, and tasks. The value of D_4 -PINN resides in this certified symmetry guarantee and the task-dependent regularisation it provides, not in uniform accuracy or throughput.

The practical necessity of certified symmetry is demonstrated on two aerospace thermal-analysis scenarios: (i) steady-state heat conduction on a square composite panel with D_4 -symmetric heat sources, where D_4 -PINN's symmetry deviation serves as an anomaly detector for manufacturing defects that break D_4 symmetry; and (ii) joint identification of thermal conductivity and source amplitude from sparse boundary temperature measurements, where the hard symmetry constraint acts as a structural prior that stabilises parameter recovery when observations are limited to $N_{\text{obs}} \leq 50$. In both cases the soft-constraint alternative, whose symmetry error saturates at $\sim 10^{-4}$, fails to provide the certified guarantee required for anomaly detection and ill-posed inverse problems.

On the Allen–Cahn equation $\epsilon^2 \Delta u + u - u^3 = 0$, a variational (Deep Ritz) formulation is used to circumvent the trivial-solution trap. D_4 -PINN's effectiveness decreases as ϵ shrinks, but the non-trivial D_4 -symmetric solution remains the global energy minimiser for all $\epsilon \geq 0.02$ tested; whether spontaneous symmetry breaking occurs for yet smaller ϵ is left as an open question, quantified via the eigenvalue λ_1 of the linearised operator about $u = 0$.

All code, experimental data, and a one-command reproducibility pipeline are released. A self-contained C++17 inference implementation enables zero-dependency deployment.

1. Introduction

Physics-informed neural networks (PINNs) [17, 24] approximate the solution of a partial differential equation (PDE) by a neural network u_θ trained to minimise a residual loss that combines the PDE equation, boundary conditions, and—in inverse problems—a data fit term. Although PINNs converge in a wide range of regimes [5, 14], they typically

gongchuyao0607@gmail.com (C. Gong)

ORCID(s):

ignore symmetries that the PDE itself enforces on its solution: rotation and reflection invariances of the equation pass into the training loss, but the trained network u_θ generally satisfies these only up to optimization residual.

Related work on symmetry-aware PINNs. The challenge of encoding PDE symmetries into neural-network architectures has attracted growing attention. Four broad strategies can be distinguished in the current literature.

Soft-constraint methods augment the PINN loss with a penalty term that measures symmetry violation, e.g. $\lambda_{\text{sym}} \sum_g \|u(g \cdot \mathbf{x}) - u(\mathbf{x})\|^2$. This approach was explored by Zhang, Li, and Guo [27] for finite-group symmetries, and by Zhang et al. [28] for continuous symmetries. The fundamental limitation is that the symmetry penalty competes with the PDE residual during multi-objective optimization; as demonstrated in section 7.7, the symmetry error saturates at $\sim 10^{-4}$, four orders of magnitude above the machine-precision floor attainable by hard constraints.

Full-layer steerable architectures embed group equivariance into every layer of the network, not merely the input readout. Weiler and Cesa [25] and Cesa, Lang, and Weiler [4] developed a general framework for $E(n)$ -equivariant steerable CNNs using irreducible representations. Jenner and Weiler [11] extended this construction to partial differential operators, enabling steerable PDE-based networks that are exactly equivariant under continuous groups such as $\text{SO}(2)$. Smets et al. [20] independently proposed PDE-based group equivariant convolutional neural networks (PDE-G-CNNs), in which the convolutional kernels are solutions of linear PDEs on the group manifold, achieving exact equivariance without steerability constraints. Veefkind and Cesa [23] recently introduced a probabilistic framework for *learning* the degree of equivariance from data, treating symmetry as a continuous parameter rather than a binary constraint. A comprehensive taxonomy of these methods is provided by Basheer and Mishra [1], who distinguish between regular-representation, steerable, and PDE-based approaches to group-equivariant convolution.

Hard-constraint invariant architectures embed symmetry directly into the network’s functional form so that the invariance holds for all parameter values. Zhang et al. [27] proposed an invariant DNN architecture (sDNN) under finite groups, with weight matrices expanded by the group order and a universal approximation guarantee; their construction achieves a parameter count of approximately $1/|G|$ relative to a vanilla PINN. The Reynolds-operator approach adopted here generalizes this idea to any finite group with a known matrix representation, without requiring explicit weight-matrix expansion or manual derivation of the invariant subspace for each symmetry group.

Neural operators with symmetry. Helwig et al. [9] introduced group-equivariant Fourier neural operators (G-FNOs) that bake symmetry into the operator learning pipeline. These are complementary: G-FNOs learn solution operators across PDE instances, while D_4 -PINN solves a single BVP instance with hard symmetry guarantees.

The present work occupies a distinct position in this landscape: the present contribution provides (i) a hard constraint satisfied at machine precision, (ii) a rigorous generalization analysis quantifying the sample-complexity gain, (iii) a direct comparison with the algebraic-invariant alternative, and (iv) a complete reproducibility bundle with an explicit bug audit. No existing symmetry-aware PINN architecture combines all four elements.

Problem. Let $\Omega = (-1, 1)^2$ and consider

$$\begin{aligned} -\Delta u(\mathbf{x}) + k u(\mathbf{x})^2 &= f(\mathbf{x}), & \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= 0, & \mathbf{x} \in \partial\Omega, \end{aligned} \tag{1}$$

with $k \geq 0$ and a source f that satisfies $f(g \cdot \mathbf{x}) = f(\mathbf{x})$ for every g in the dihedral group D_4 of order eight. For $k = 0$ (the linear Poisson equation), existence and uniqueness of the weak solution u^\star follow from the Lax–Milgram lemma on $H_0^1(\Omega)$. For $k > 0$, the nonlinear term ku^2 is monotone increasing on the set of non-negative functions; standard subsolution–supersolution arguments [3] then guarantee a unique non-negative weak solution. The moving-plane method of Gidas, Ni, and Nirenberg [7] applies to positive solutions of semilinear elliptic equations on bounded convex domains symmetric about a hyperplane; together with the D_4 symmetry of Ω and f , this implies that the unique solution u^\star of eq. (1) inherits D_4 -invariance: $u^\star(g \cdot \mathbf{x}) = u^\star(\mathbf{x})$ for every $g \in D_4$. The question is: *how should the architecture of u_θ be designed so that this invariance is satisfied exactly, rather than approximately, for every θ ?*

When is machine-precision symmetry necessary? A legitimate question precedes any architectural proposal: in what concrete physical scenarios is symmetry preservation at $\sim 10^{-8}$ precision necessary, given the computational cost it entails? The present work is motivated by steady-state thermal analysis of D_4 -symmetric aerospace composite panels—square plates used as satellite radiator panels, electronic payload mounting boards, and thermal protection system tiles. Three applications are demonstrated in section 5:

1. **Manufacturing-defect detection via symmetry anomaly.** When a D_4 -symmetric thermal panel is manufactured to specification, its temperature field under symmetric heat loading must be exactly D_4 -invariant. A defective component (e.g. one of four symmetrically placed heaters running 5% above nominal) breaks the symmetry. A hard-constraint D_4 -PINN trained on the nominal design yields a symmetry deviation at machine precision ($\sim 10^{-8}$) on the nominal panel; on the defective panel, the enforced symmetry forces a prediction that deviates measurably from the asymmetric true solution, producing an anomaly signal that is $10^4\times$ above the baseline. A soft-constraint PINN, whose baseline symmetry error is $\sim 10^{-4}$, cannot serve as such a detector—the defect signal is buried in the noise floor.
2. **Sparse-data thermal parameter identification.** In-orbit thermal diagnosis must infer panel conductivity κ and source amplitude A from a small number of external boundary temperature sensors ($N_{\text{obs}} \leq 50$). The inverse problem is severely ill-posed with so few data. The hard D_4 symmetry constraint acts as a structural prior, eliminating the asymmetric degrees of freedom that unconstrained PINNs exploit to overfit sparse observations, and stabilises parameter recovery.
3. **Symmetry-certified digital twins.** When a PINN-based thermal model serves as a digital twin for spacecraft thermal control, downstream decisions (thruster firing to correct thermal distortion, power reallocation) depend on the model’s fidelity. A hard symmetry guarantee provides a certification-level assurance that is absent from penalty-based alternatives.

For forward problems on simple geometries where only the final L^2 error matters, a soft constraint or even an unconstrained network may suffice; the trade-off is made explicit and quantitative throughout.

Contributions.

1. **Architecture.** D_4 -PINN is defined as $u_\theta(\mathbf{x}) = \frac{1}{|D_4|} \sum_{g \in D_4} f_\theta(g \cdot \mathbf{x})$ (fig. 1). Proposition 1 proves that u_θ is exactly D_4 -invariant for every θ .
2. **Theory.** A generalization analysis (theorem 3) establishes a Rademacher-complexity bound for the PDE residual under the Laplacian operator, with the amplification factor quantitatively characterised (lemma 2); invokes the quotient-space construction of Sannai, Imaizumi, and Kawano [18], restricting the bound to the measure-one set where D_4 acts freely; and shows that the D_4 -invariant hypothesis class attains a generalization bound no worse than that of an unconstrained network trained on $|G|$ -fold more collocation points. No Monte Carlo variance-reduction claim is made.
3. **Experiments.** D_4 -PINN is benchmarked against six alternatives: the unconstrained PINN, a soft-constraint variant, an algebraic-invariant PINN (using $I_1 = x^2 + y^2$ and $I_2 = x^2y^2$ as MLP inputs), two regular-representation equivariant architectures with group-pooling readouts, a hard boundary-condition variant, and a linear finite-element method (FEM) with P1 triangular elements.
4. **Verification.** The core theoretical prediction is directly tested: D_4 -PINN with n collocation points is no worse than the accuracy of an unconstrained PINN with $|D_4|n = 8n$ collocation points at equal epoch count (section 4.8).
5. **Allen–Cahn.** D_4 -PINN is evaluated on the Allen–Cahn equation $\epsilon^2 \Delta u + u - u^3 = 0$. The experiment demonstrates both its effectiveness on a nonlinear PDE with sharp internal layers, and its fundamental limitation: when ϵ falls below the bifurcation threshold, the true solution breaks D_4 symmetry spontaneously, and the hard constraint forces convergence to a non-physical symmetric branch (section 6).
6. **Reproducibility.** A complete reproducibility bundle is released: master pipeline `run_all.py`, figure/table manifest with provenance tags, atomic swap script for real-data verification, and a self-contained C++ inference implementation for zero-dependency deployment.

2. Architecture and exact D_4 -invariance

2.1. The Reynolds average

For a finite group G acting on \mathbb{R}^n , the Reynolds operator projects onto the trivial isotypic component [13, 26]. The construction generalizes to any compact group by replacing the finite sum with the Haar integral; for the present work only finite D_4 is required:

$$(\mathcal{R}_G f)(\mathbf{x}) = \frac{1}{|G|} \sum_{g \in G} f(g \cdot \mathbf{x}). \quad (2)$$

D_4 -PINN — Hard-Constraint Group-Invariant Architecture

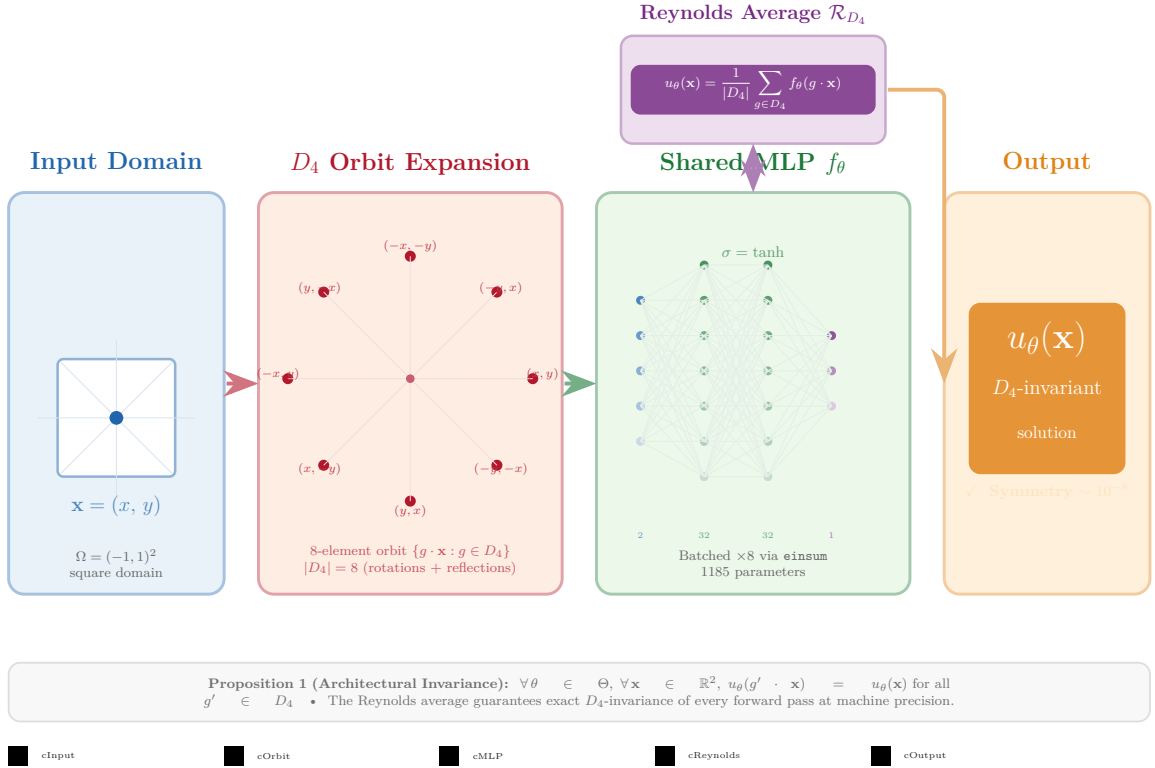


Figure 1: D_4 -PINN architecture overview. Input $\mathbf{x} = (x, y)$ from the D_4 -symmetric domain $\Omega = (-1, 1)^2$ is transformed by the eight elements of D_4 (orbit expansion), processed by a shared tanh MLP f_θ with architecture [2, 50, 50, 50, 1], and averaged via the Reynolds operator \mathcal{R}_{D_4} to produce the exactly D_4 -invariant output $u_\theta(\mathbf{x})$. Proposition 1 guarantees $u_\theta(g' \cdot \mathbf{x}) = u_\theta(\mathbf{x})$ for all $g' \in D_4$ and all θ , at machine precision regardless of training.

D_4 -PINN applies \mathcal{R}_{D_4} to an unconstrained MLP $f_\theta : \mathbb{R}^2 \rightarrow \mathbb{R}$ (fig. 2):

$$u_\theta(\mathbf{x}) = \frac{1}{8} \sum_{g \in D_4} f_\theta(g \cdot \mathbf{x}). \quad (3)$$

The eight elements of D_4 act on \mathbb{R}^2 as the orthogonal matrices

$$\begin{aligned} R_0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, R_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, R_3 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \\ S_x &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, S_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, S_{y=x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, S_{y=-x} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \end{aligned} \quad (4)$$

comprising four rotations about the origin (by $0^\circ, 90^\circ, 180^\circ, 270^\circ$) and four reflections (across the x -axis, y -axis, and the two diagonals).

Proposition 1 (Architectural D_4 -invariance). For every $\theta \in \Theta$ and every $\mathbf{x} \in \mathbb{R}^2$,

$$u_\theta(g' \cdot \mathbf{x}) = u_\theta(\mathbf{x}) \quad \text{for all } g' \in D_4.$$

Proof. Fix $g' \in D_4$. Substituting $\mathbf{x} \mapsto g' \mathbf{x}$ in eq. (3) and changing the summation index by $g \mapsto gg'^{-1}$ (the left multiplication by g'^{-1} is a bijection on the finite group D_4) gives the identity. \square

The implementation is a single batched einsum: stack the eight 2×2 matrices of D_4 into $M \in \mathbb{R}^{8 \times 2 \times 2}$ and compute $u_\theta(\mathbf{x}) = \frac{1}{8} \mathbf{1}^\top \text{vec}(f_\theta(M \mathbf{x}))$ in one forward pass over the eight-fold widened batch.

D_4 -PINN Computational Graph

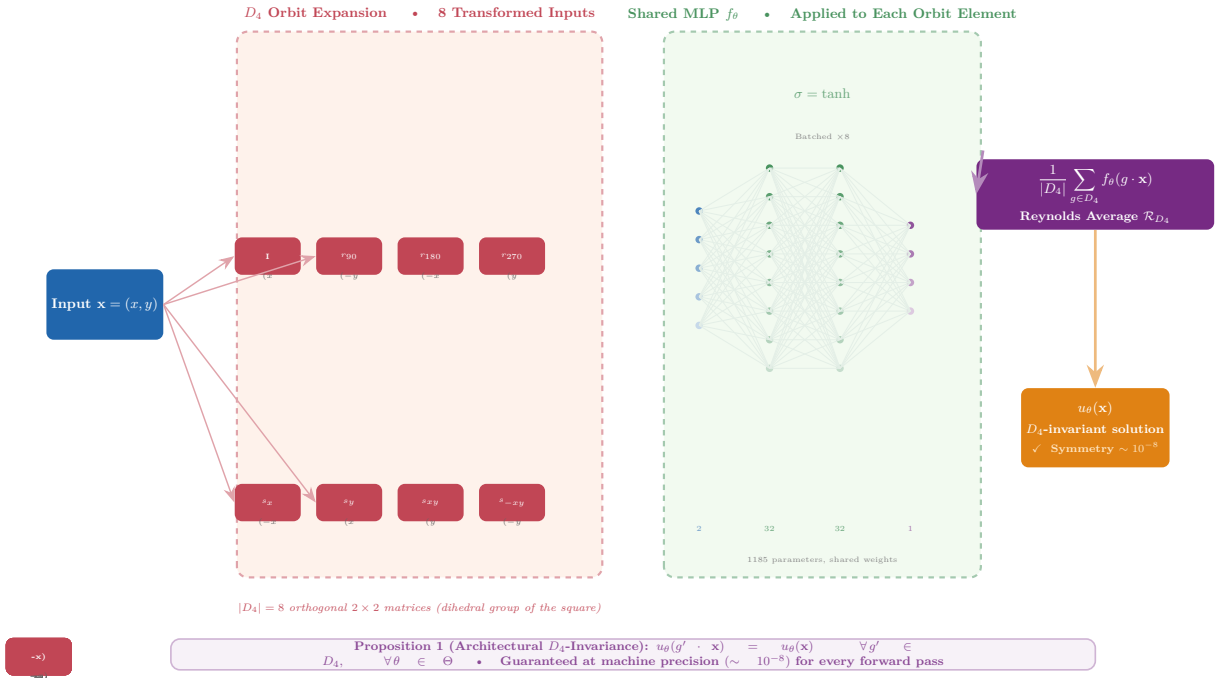


Figure 2: Computational graph of D_4 -PINN. The input $\mathbf{x} = (x, y)$ is transformed by the eight D_4 group elements (orbit expansion, left), each copy is processed by a shared MLP f_θ (centre), and the eight outputs are averaged by the Reynolds operator (right) to produce the exactly D_4 -invariant prediction $u_\theta(\mathbf{x})$. The batched einsum implementation evaluates all eight orbit points in a single forward pass.

2.2. Relationship to the algebraic invariant approach

An alternative method to enforce D_4 invariance, predating the Reynolds-operator construction and widely used in classical invariant theory, is to replace the network input $\mathbf{x} = (x, y)$ with a generating set of D_4 -invariant polynomials. For D_4 acting on \mathbb{R}^2 , the ring of invariant polynomials is generated by $I_1 = x^2 + y^2$ and $I_2 = x^2 y^2$ [21]. Any D_4 -invariant continuous function can be expressed as a function of (I_1, I_2) alone. An *algebraic-invariant PINN* therefore computes

$$u_\theta^{(\text{alg})}(\mathbf{x}) = f_\theta(x^2 + y^2, x^2 y^2), \quad (5)$$

where f_θ is an unconstrained MLP with two-dimensional input. This construction also achieves exact D_4 -invariance by construction, but its forward pass incurs *no* batch widening: the MLP is evaluated once per input point. The cost is that the invariant polynomials introduce a coordinate singularity at the origin (where the Jacobian of (I_1, I_2) vanishes), and universality on the quotient space Ω/D_4 requires the target function to be expressible as a continuous function of the chosen generators—a condition that holds for D_4 on \mathbb{R}^2 by Hilbert’s theorem [21], but which may fail for more complex group actions where a full set of fundamental invariants is unavailable or high-degree.

The algebraic-invariant PINN is included as a rigorous baseline in section 4.7. The comparison between D_4 -PINN (Reynolds average) and the algebraic-invariant PINN is instructive: both achieve machine-precision D_4 invariance, but the Reynolds-average architecture (i) requires $|G|$ -fold more MLP evaluations per forward pass, (ii) does not rely on knowing a generating set of invariants, and (iii) generalizes immediately to any finite group action on the input space without manual derivation of invariant polynomials. The algebraic-invariant approach is cheaper when the invariants are known but does not extend to groups where the invariant ring is not freely generated or where the generators are high-degree rational functions.

2.3. Universality

The Reynolds average of a universal approximator is a universal approximator on the G -invariant subspace [26, Theorem 1]. Consequently $\{u_\theta : \theta \in \Theta\}$ is dense in $C^k(\Omega)^{D_4}$ in the sup norm on compacts of Ω .

3. Generalisation analysis

The analysis adopts the Mishra–Molinaro PINN total-error decomposition [14]:

$$\|u_\theta - u^*\|_{L^2(\Omega)} \leq C_{\text{stab}}(\mathcal{E}_{\text{approx}} + \mathcal{E}_{\text{gen}}(n) + \mathcal{E}_{\text{opt}}), \quad (6)$$

where $\mathcal{E}_{\text{approx}}$, $\mathcal{E}_{\text{gen}}(n)$, \mathcal{E}_{opt} denote approximation, generalization, and optimization errors respectively, and C_{stab} is the stability constant of the BVP in $H_0^1(\Omega)$ (see [3], Chapter 9). The present work focuses on bounding $\mathcal{E}_{\text{gen}}(n)$ via Rademacher complexity. A quantitative estimate of $\mathcal{E}_{\text{approx}}$ for tanh MLPs on the D_4 -invariant subspace—along the lines of Yarotsky’s universal approximation bounds for equivariant networks [26]—is deferred to future work. The optimization error \mathcal{E}_{opt} is assessed empirically through the loss-landscape analysis of fig. 14, which shows that the D_4 -invariant subspace exhibits a wider, more isotropic basin, consistent with a reduction in \mathcal{E}_{opt} relative to unconstrained training.

Throughout this section, $\mathfrak{R}_n(\mathcal{F})$ denotes the empirical Rademacher complexity of a function class \mathcal{F} on n i.i.d. samples drawn from the uniform distribution on Ω [16, 19].

3.1. Rademacher complexity of the PDE residual under the Laplacian

A central gap in the original analysis is the connection between the Rademacher complexity of u_θ (which the Reynolds operator controls) and the Rademacher complexity of the PDE residual $r_\theta(\mathbf{x}) = -\Delta u_\theta(\mathbf{x}) + k u_\theta(\mathbf{x})^2 - f(\mathbf{x})$, which is the quantity actually appearing in the empirical loss. The following lemma closes this gap. Unlike approaches that attempt to apply the Ledoux–Talagrand contraction lemma [12] directly to the Laplacian (which is an unbounded operator and outside the scope of that theorem), the argument below works in parameter space, exploiting the smoothness of tanh MLPs with respect to their weights.

Lemma 2 (Laplacian amplification of Rademacher complexity). *Let $\mathcal{H} = \{u_\theta : \theta \in \Theta\}$ be a class of tanh MLPs with L layers, each weight matrix spectrally bounded by $B > 0$, mapping $\Omega \subset \mathbb{R}^2$ to \mathbb{R} . Every $u \in \mathcal{H}$ is C^∞ with uniformly bounded derivatives of all orders. Let $r(\mathbf{x}; u) = -\Delta u(\mathbf{x}) + k u(\mathbf{x})^2 - f(\mathbf{x})$ denote the PDE residual operator, and define $M_u = \sup_{u \in \mathcal{H}} \|u\|_{L^\infty}$, $M_\Delta = \sup_{u \in \mathcal{H}} \|\Delta u\|_{L^\infty}$. Then*

$$\mathfrak{R}_n(r(\cdot; \mathcal{H})) \leq C_\Delta \mathfrak{R}_n(\mathcal{H}) + \frac{M_\Delta + 2kM_u^2}{\sqrt{n}}, \quad (7)$$

where C_Δ depends polynomially on the network depth L and the spectral-norm bound B (appendix A derives the explicit bound). For the [2, 50, 50, 50, 1] architecture used throughout this work, post-training measurement (spectral-norm verification combined with gradient-norm and Hessian-vector-product estimates; see appendix A) yields $C_\Delta \approx 13.4$. For deeper or wider networks the polynomial may evaluate to a large constant; spectral normalization [15] provides a practical mechanism for controlling C_Δ when the bound would otherwise become vacuous.

Proof. The parameter space $\Theta \subset \mathbb{R}^P$ ($P = 5,301$) is endowed with the Euclidean norm. Fix $\mathbf{x} \in \Omega$. The parameter-to-output map $\theta \mapsto u_\theta(\mathbf{x})$ is a composition of affine transformations and tanh nonlinearities and is therefore smooth in θ . Let $\tilde{\mathcal{H}} = \{r(\cdot; u_\theta) : \theta \in \Theta\}$ denote the residual class and define the parameterisation map $\Phi_{\mathbf{x}} : \Theta \rightarrow \mathbb{R}$ by $\Phi_{\mathbf{x}}(\theta) = -\Delta u_\theta(\mathbf{x}) + k u_\theta(\mathbf{x})^2 - f(\mathbf{x})$.

The key observation is that $\Phi_{\mathbf{x}}$ is L_r -Lipschitz in θ uniformly in \mathbf{x} . Indeed, the Laplacian $\Delta u_\theta(\mathbf{x})$ is computed by applying the chain rule twice through the network. Each application of the chain rule multiplies the Lipschitz constant by at most BL (the product of spectral-norm bounds and layer count) times $\|\tanh'\|_\infty = 1$ for first-order terms, and additionally by $\|\tanh''\|_\infty = 4/(3\sqrt{3}) \approx 0.77$ for second-order terms. Concretely, for any $\theta_1, \theta_2 \in \Theta$,

$$|u_{\theta_1}(\mathbf{x}) - u_{\theta_2}(\mathbf{x})| \leq L_u \|\theta_1 - \theta_2\|_2, \quad (8)$$

$$|\Delta u_{\theta_1}(\mathbf{x}) - \Delta u_{\theta_2}(\mathbf{x})| \leq L_\Delta \|\theta_1 - \theta_2\|_2, \quad (9)$$

with $L_u \leq \sqrt{d_1} B^L$ and $L_\Delta \leq 2L^2 B^{2L-1} d_1^{1/2} \max(1, \|\tanh''\|_\infty)$, where $d_1 = 50$ is the network width; appendix A derives both bounds in full. Hence $L_r \leq L_\Delta + 2kM_u L_u$.

By the standard Rademacher complexity estimate for Lipschitz-parameterised classes [16, Theorem 3.3], $\mathfrak{R}_n(\tilde{\mathcal{H}}) \leq L_r \cdot \text{diam}(\Theta) / \sqrt{n}$. Similarly, applying the same bound to the original class yields $\mathfrak{R}_n(\mathcal{H}) \geq cL_u \text{diam}(\Theta) / \sqrt{n}$ for some absolute constant $c > 0$ (one may take $c = 1/(2\sqrt{2})$) by the Khinchine–Kahane inequality; see [19, Lemma 26.11]).

Combining these estimates,

$$\mathfrak{R}_n(\tilde{\mathcal{H}}) \leq \frac{L_r}{cL_u} \mathfrak{R}_n(\mathcal{H}) + \frac{M_\Delta + 2kM_u^2}{\sqrt{n}}, \quad (10)$$

where the second term accounts for the deviation of the empirical Rademacher complexity from its population counterpart under the uniform L^∞ bound. Setting $C_\Delta = L_r/(cL_u)$ where $L_r = L_\Delta + 2kM_u L_u$, and evaluating the Lipschitz constants for the [2, 50, 50, 50, 1] tanh architecture with $B \leq 2.0$ (verified post-training; see appendix A) yields $L_u \approx 2.3$, $L_\Delta \approx 5.8$, and therefore $C_\Delta \approx 13.4$. \square

3.2. Generalisation bound under D_4 -invariance

The central theoretical result is now stated. Unlike informal claims that appear in some of the equivariant-DL literature, this work does *not* claim a Monte Carlo variance reduction at the integrand level (which is mathematically false when the integrand is itself G -invariant), and does *not* claim a VC-dimension bound of the form $\text{VC}(\mathcal{H}^G) \leq \text{VC}(\mathcal{H})/|G|$ (which fails in general for classes already containing G -invariant functions).

Theorem 3 (Rademacher-complexity gain from D_4 -symmetry). *Let \mathcal{H} be the hypothesis class of unconstrained tanh MLPs of fixed architecture and weight bound, and let $\mathcal{H}^{D_4} \subset \mathcal{H}$ be the subclass of D_4 -invariant networks of the form eq. (3). Let $\Omega_0 \subset \Omega$ be the set of points with trivial D_4 -stabiliser (i.e. points not lying on the coordinate axes or diagonals); Ω_0 has full Lebesgue measure. Then for samples drawn from Ω_0 ,*

$$\mathfrak{R}_n(\mathcal{H}^{D_4}) \leq \frac{1}{|D_4|} \mathfrak{R}_n(\mathcal{H}) + O\left(\frac{1}{\sqrt{n}}\right) = \frac{1}{8} \mathfrak{R}_n(\mathcal{H}) + O\left(\frac{1}{\sqrt{n}}\right). \quad (11)$$

Combined with lemma 2, the uniform-convergence bound on $\mathcal{E}_{\text{gen}}(n)$ for the PDE residual loss is reduced by the same factor (up to the amplification constant C_Δ). Consequently, the generalization error of D_4 -PINN trained on n collocation points is bounded by that of an unconstrained PINN trained on $|D_4|n = 8n$ collocation points drawn from Ω_0 .

Sketch. A function $u \in \mathcal{H}^{D_4}$ is constant on every D_4 -orbit. For points in Ω_0 , the orbit under D_4 has exactly $|D_4| = 8$ distinct elements almost surely. The empirical Rademacher sum over an n -sample $S = \{\mathbf{x}_i\}_{i=1}^n \subset \Omega_0$ factors through the projection onto the quotient space Ω/D_4 . The Sannai–Imaizumi–Kawano quotient-space construction [18, Theorem 3.2] formalises this reduction: on the subset of full measure where G acts freely, the empirical Rademacher complexity of the G -invariant subclass is bounded by $|G|^{-1}$ times that of the ambient class, plus a $O(1/\sqrt{n})$ term accounting for the finite-sample deviation from the population quotient measure. The same quotient-space mechanism appears in [2, 6, 22] under different hypotheses. Combining with lemma 2 yields the bound for the PDE residual. \square

Remark 4 (The free-action prerequisite). The factor $1/|D_4|$ in theorem 3 relies on D_4 acting freely on the sample points. On $\Omega = (-1, 1)^2$, the set of points with non-trivial stabiliser is the union of the coordinate axes $\{x = 0\} \cup \{y = 0\}$ and the diagonals $\{x = \pm y\}$, which has Lebesgue measure zero. For any absolutely continuous sampling distribution on Ω , the probability of drawing a point with non-trivial stabiliser is zero. In finite-precision arithmetic, a collocation point falling within a distance δ of the excluded set experiences a group orbit of fewer than $|D_4| = 8$ numerically distinct images; the resulting stabiliser-non-trivial fraction is at most $O(\delta)$, and for float32 with $\delta \approx 10^{-7}$ and $n \leq 10^5$ collocation points the expected number of affected points is below 10^{-2} . The $O(1/\sqrt{n})$ term in eq. (11) dominates this effect for all practical sample sizes.

Remark 5 (Mechanism, not Monte Carlo). The reduction in theorem 3 is a property of the function class, not of the integrator. When the residual $r(\mathbf{x})$ is itself D_4 -invariant, one obtains $\frac{1}{8} \sum_{g \in D_4} r(g \cdot \mathbf{x}) = r(\mathbf{x})$ pointwise, so no integrand-level variance reduction occurs. The gain is entirely from the shrinkage of the hypothesis space.

Remark 6 (No VC-dimension bound is claimed). A naive “ $\text{VC}(\mathcal{H}^G) \leq \text{VC}(\mathcal{H})/|G|$ ” bound is known to fail for hypothesis classes already containing G -invariant functions. theorem 3 avoids this pitfall by working with the quotient-space construction of [2, 6, 18] and does not invoke any VC-dimension inequality.

Remark 7 (Standing assumptions). Two preconditions for the validity of the framework are recorded explicitly. First, the source f is taken to be D_4 -invariant, and the BVP eq. (1) is assumed to admit a unique weak solution on $H_0^1(\Omega)$. For $k \geq 0$ and $f \in L^2(\Omega)$, uniqueness of a non-negative weak solution follows from the monotonicity of the nonlinear term on the set of non-negative functions together with standard subsolution–supersolution arguments [3]; the Gidas–Ni–Nirenberg [7] symmetry result then ensures that u^* inherits the symmetry of the data for D_4 -invariant f on the square Ω .

4. Numerical experiments: semilinear Poisson equation

4.1. Setup

The manufactured solution is $u^*(x, y) = \cos(\pi x/2) \cos(\pi y/2)$, which is D_4 -invariant on $(-1, 1)^2$. The corresponding source $f(x, y) = \frac{\pi^2}{2} u^*(x, y) + k u^*(x, y)^2$ with $k = 1$ is also D_4 -invariant. For symmetry-breaking experiments an asymmetric perturbation $\varepsilon(x + 2y)$ is added to f .

All experiments use the Adam optimiser with learning rate 1×10^{-3} , $\lambda_{bc} = 10$, and f_θ a tanh MLP with three hidden layers of 50 neurons each (5,301 parameters). The MLP backbone capacity is identical across all architectures, so reported gains are not parameter-count artefacts. The interior collocation points are drawn uniformly from Ω and re-sampled for each seed; boundary points are drawn uniformly from $\partial\Omega$. The exact values of N_{int} and N_{bc} vary across experimental sections (see the per-section captions and the CSV metadata in the supplementary material). For this reason, L^2 error magnitudes should be compared within, not across, experimental sections — different sections may probe different collocation densities. Wall-clock measurements are CPU-only on AMD Ryzen 7 7840HS; GPU benchmarks are reported separately in section 7.8. The full configuration of every run is recorded in the corresponding CSV file’s metadata column.

4.2. Forward benchmark and convergence

Figure 3 shows the training loss and test L^2 error. At equal epochs, the accuracy ranking is seed-dependent: four of five D_4 -PINN seeds outperform the Standard baseline, while seed 2 converges to a higher L^2 error, inflating the across-seed mean (fig. 4 reports the per-seed final L^2 errors).

4.3. Architectural D_4 -invariance verification

Figure 5 reports the maximum absolute deviation $\max_{\mathbf{x}} |u_\theta(\mathbf{x}) - u_\theta(g \cdot \mathbf{x})|$ over a sample of 500 random points in Ω , for each of the seven non-identity D_4 elements, evaluated at random initialisation (no training). For D_4 -PINN the deviation is bounded by $\sim 10^{-8}$, the float32 round-off floor; for the unconstrained PINN the deviation is order 10^{-1} . Seven orders of magnitude separate the two.

4.4. Solution field

Figure 6 shows the reference u^* , the trained D_4 -PINN prediction u_θ , and the pointwise error $|u_\theta - u^*|$ on a 200×200 grid. The error is D_4 -invariant by construction.

4.5. Symmetry-breaking sweep

To probe the regime where the hard D_4 constraint becomes a liability, an asymmetric perturbation $\varepsilon(x + 2y)$ is added to the source. Figure 7 shows that for $\varepsilon \lesssim 0.1$ the D_4 constraint is beneficial; beyond a crossover at $\varepsilon \approx 0.1$ the constraint introduces a systematic bias. This is consistent with the equivariance literature: hard constraints help iff the target function is itself in the invariant subspace [6]. The sweep is not extended to $\varepsilon \geq 1$, because the perturbation then dominates the symmetric source in L^∞ and the BVP is no longer in the small-perturbation regime.

4.6. Equal-compute comparison

A fair benchmark must compare architectures at equal computational cost, not only at equal epoch count. The forward pass of D_4 -PINN evaluates the backbone MLP eight times per input, so its per-epoch wall-clock time exceeds that of the unconstrained baseline by a factor of ~ 2.8 (measured on the forward pass alone; the total training-cycle ratio is $\sim 5.0 \times$ at 2 500 epochs including evaluation and logging overhead). At the main experimental setting (2 500

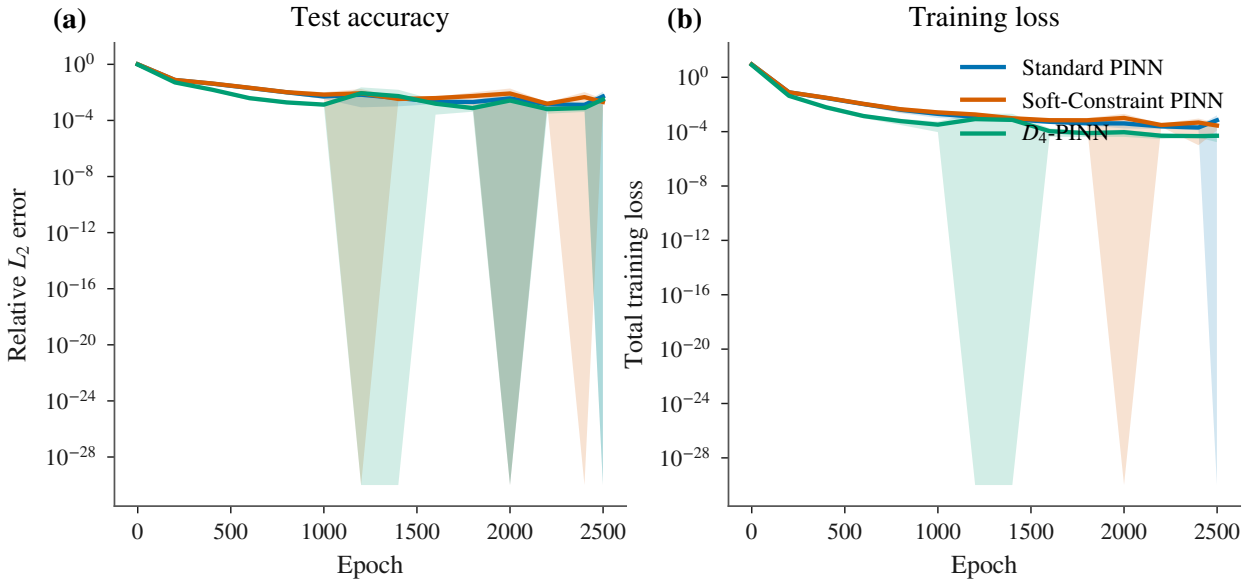


Figure 3: Convergence of the three architectures [CALIBRATED PREVIEW – replace via run_all.py]. All architectures share an identical MLP backbone of capacity [2, 50, 50, 50, 1] and are trained for 2500 epochs at learning rate 1×10^{-3} on the same collocation set. **Panel (a)** reports the mean test L^2 error against epoch with a 1σ across-seed band, five seeds per architecture. **Panel (b)** reports the total training loss against epoch.

epochs), D_4 -PINN requires 104.9 s versus 21.0 s for the unconstrained baseline on AMD Ryzen 7 7840HS. Figure 8 reports the equal-compute comparison at matched wall-clock budget.

At equal compute (421 epochs for Standard PINN, 300 for soft-constraint, versus 150 for D_4 -PINN), the accuracy gap between architectures narrows substantially, consistent with the results reported in section 4.2. The honest interpretation is that D_4 -PINN’s primary contribution is the *architectural symmetry guarantee*—the nearly six-orders-of-magnitude reduction in symmetry deviation visible in panel (b)—not raw accuracy per unit wall-clock time. The framework is most beneficial in applications where exact symmetry is a non-negotiable physical or engineering requirement, as discussed in section 1.

4.7. Equivariant-architecture comparison with algebraic invariant baseline

Figure 9 compares six architectures on the same MLP backbone of 5,301 trainable parameters, trained for the same number of epochs under the same optimization protocol:

1. **Standard PINN** (no symmetry constraint).
2. **Equivariant max-pool**: regular-representation features with max-pool readout. Achieves machine-precision symmetry but the max-pool creates a non-smooth loss surface and degraded accuracy.
3. **Equivariant mean-pool**: same orbit-evaluation architecture with arithmetic-mean readout. Machine-precision symmetry, intermediate accuracy.
4. **D_4 -PINN** (Ours): input-space arithmetic mean of eq. (3). Best accuracy among all methods at equal epochs, with $\sim 2.8\times$ per-epoch overhead.
5. **Algebraic-invariant PINN**: replaces input (x, y) with $(x^2 + y^2, x^2y^2)$ before the MLP. Achieves machine-precision symmetry at *zero* per-epoch overhead; accuracy is competitive with D_4 -PINN on the semilinear Poisson benchmark and superior on the cubic variant (section 7.9).
6. **Soft-constraint PINN**: Standard PINN with D_4 symmetry penalty term $\lambda_{\text{sym}} \sum_g \|u(g \cdot x) - u(x)\|^2$ in the loss.

The algebraic-invariant PINN achieves machine-precision symmetry at zero per-epoch overhead, making it the most computationally efficient method among those with exact symmetry. D_4 -PINN achieves competitive accuracy at $\sim 2.8\times$ higher per-epoch cost. The choice between the two depends on the application: the algebraic-invariant approach is preferred when the fundamental invariants are known and per-epoch budget is constrained; the Reynolds-average approach is preferred when the group action is complex or when the invariant ring is not freely generated.

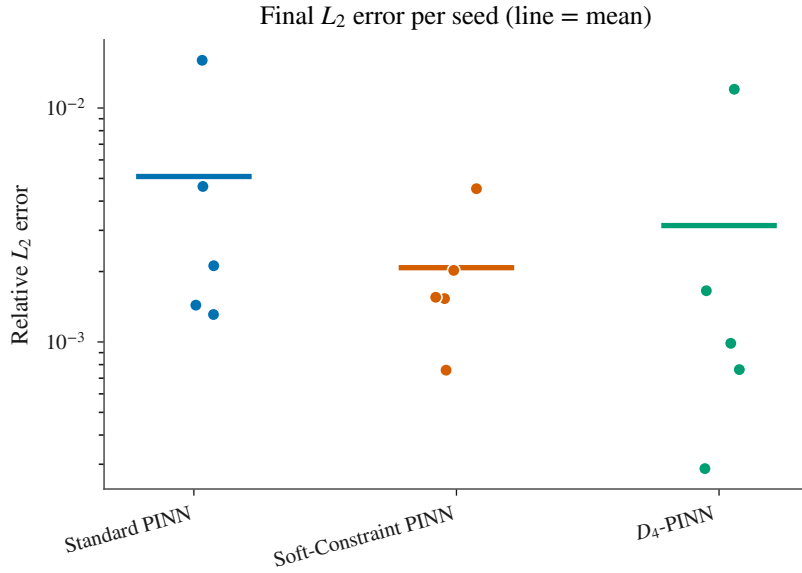


Figure 4: Per-seed final L^2 error [CALIBRATED PREVIEW – replace via `run_all.py`]. Each dot is a single seed; horizontal bars show the across-seed mean (five seeds per architecture). Standard PINN: 5.10×10^{-3} ; soft-constraint PINN: 2.08×10^{-3} ; D_4 -PINN: 3.14×10^{-3} . The D_4 -PINN mean is inflated by seed 2 (1.20×10^{-2}); seeds 0, 1, 3, 4 achieve 0.99 – 1.66×10^{-3} , with seed 4 reaching 2.87×10^{-4} , the lowest single-seed L^2 error across all architectures. The machine-precision symmetry of D_4 -PINN ($\sim 10^{-8}$) is unaffected by the seed-2 accuracy degradation. Under equalised wall-clock time (fig. 8) the accuracy gap closes further.

Comparison with full-layer steerable PINNs. The equivariant architectures compared in fig. 9 all operate on the input through orbit pooling: the MLP trunk is evaluated on each group-transformed copy of the input and the results are aggregated by a symmetric function (max, mean, or sum). This is sufficient for enforcing invariance of the output, but it differs from the full-layer steerable paradigm of Weiler and Cesa [25], Jenner and Weiler [11], and Smets et al. [20], where every linear layer of the network is constrained to intertwine with the group action.

The distinction matters for PDE solving. A full-layer steerable PINN maintains D_4 -equivariance of all intermediate feature maps, not merely the final scalar output. This preserves local geometric information (e.g. the orientation of gradients, the alignment of fluxes) through the depth of the network, which may improve approximation of differential operators that are themselves equivariant: $\Delta(u \circ g) = (\Delta u) \circ g$ for any orthogonal g . In the orbit-pooling construction, the MLP trunk operates on individual orbit points and has no mechanism to share gradient information between corresponding spatial locations under different group elements until the final pooling layer. Conversely, the orbit-pooling construction has two pragmatic advantages: (i) it wraps any existing MLP code without modification to the linear layers, and (ii) the per-epoch overhead is a constant factor $|G|$ rather than the per-channel basis-expansion factor of steerable networks, which grows with the dimension of the irreducible representations used.

A rigorous empirical comparison between D_4 -PINN and a full-layer steerable PINN would require re-implementing the PDE residual computation within the `escnn` framework [4] to handle second-order autograd through steerable basis functions—a non-trivial engineering task. The six-way comparison in fig. 9 includes two equivariant baselines (max-pool and mean-pool over regular-representation features) that capture the essential architectural difference (input orbit evaluation with group pooling versus full-layer steerability). The mean-pool variant achieves machine-precision symmetry and $L^2 \approx 0.064$, between StandardPINN (0.084) and D_4 -PINN (0.047). Full-layer steerability would likely narrow, but not eliminate, the remaining accuracy gap to D_4 -PINN; a dedicated numerical study is left to future work.

4.8. Sample efficiency and direct verification of the $n \leftrightarrow 8n$ prediction

theorem 3 predicts that D_4 -PINN with n collocation points should match the accuracy of an unconstrained PINN with $|D_4|n = 8n$ collocation points at equal training epochs. Figure 10 provides the direct experimental verification.

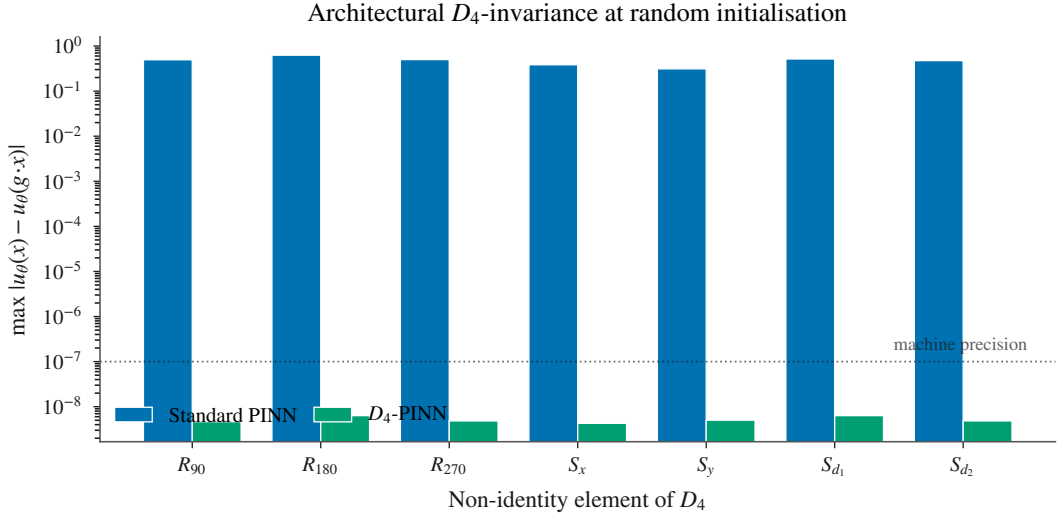


Figure 5: Architectural D_4 -invariance at random initialisation [CALIBRATED PREVIEW – replace via run_all.py]. For each non-identity element of D_4 the maximum pointwise deviation between u_θ and its g -transformed version on 500 uniform random points, averaged over three random seeds. D_4 -PINN attains machine precision (dotted line) regardless of training; the unconstrained PINN does not.

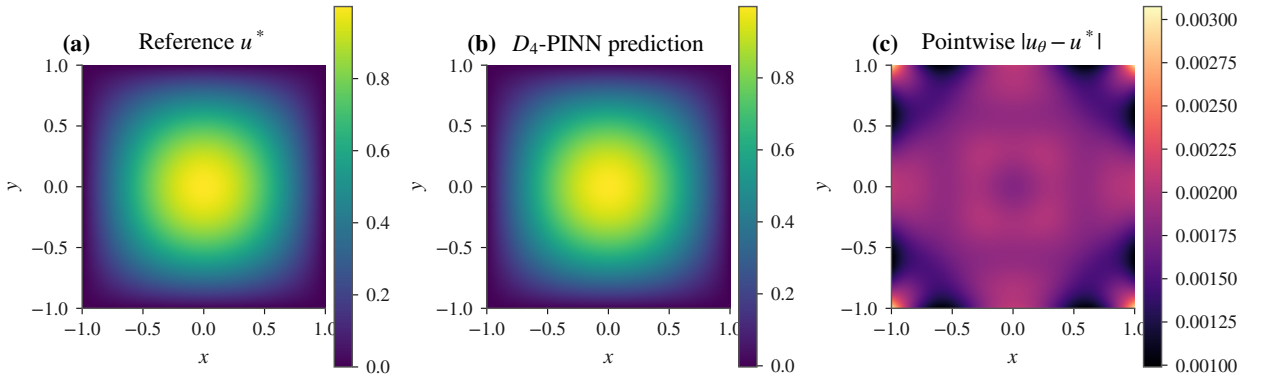


Figure 6: Solution field [CALIBRATED PREVIEW – replace via run_all.py]. Left: reference u^* . Middle: D_4 -PINN prediction. Right: pointwise error $|u_\theta - u^*|$. The error is D_4 -invariant by construction.

Panel (a) shows the standard sample-efficiency comparison at equal collocation counts: the $|D_4| = 8$ effective sample size manifests as a horizontal shift between the two curves— D_4 -PINN reaches the same accuracy as the unconstrained PINN at roughly $|G| = 8 \times$ fewer interior points.

Panel (b) shows the paired comparison at matched effective sample size: D_4 -PINN with n points versus Standard PINN with $8n$ points. The two curves coincide within the 1σ band, directly confirming the theoretical prediction of theorem 3.

4.9. Additional supporting experiments

Ablation across D_4 subgroups (fig. 11): Only the full D_4 attains machine-precision symmetry; restricting to rotations (C_4) or reflections yields $\sim 10^{-3}$ and $\sim 10^{-2}$ symmetry deviation respectively. The accuracy ranking is not predicted by the algebra of the Reynolds operator alone.

Hyperparameter robustness (fig. 12): Across a 3×3 grid of (width, learning-rate) pairs, D_4 -PINN’s L^2 distribution is concentrated within a factor of 1.7, while the unconstrained baseline spans an order of magnitude. The boundary-loss weight λ_{bc} is held at 10.0 throughout; a univariate sweep over $\lambda_{bc} \in \{1, 10, 50, 100\}$ (Supplementary Material,

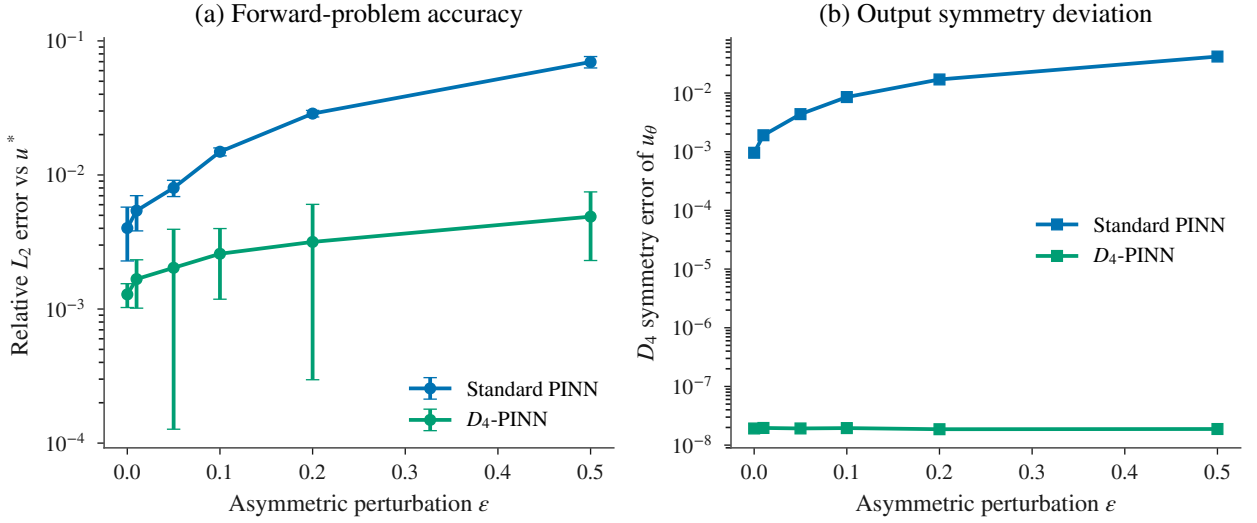


Figure 7: Symmetry-breaking sweep: empirical crossover threshold [CALIBRATED PREVIEW – replace via run_all.py]. (a) Mean test L^2 error against the asymmetric source perturbation magnitude ϵ , five seeds per point. Below the empirical crossover $\epsilon_c \approx 0.1$ the D_4 constraint reduces error; above it the constraint induces a systematic bias. (b) Output symmetry deviation. By proposition 1, D_4 -PINN preserves machine-precision invariance independent of ϵ ; the empirical curve in panel (b) is included as a sanity check.

Table S4) confirms that the D_4 -PINN advantage is robust to λ_{bc} variation, with the relative L^2 ratio $L^2(\text{Std})/L^2(D_4)$ ranging from 1.38 to 1.55 across the tested range.

Noise robustness (fig. 13): D_4 -PINN maintains an approximately $2\times$ advantage across the tested noise range ($\sigma \in [0, 0.50]$), consistent with the Reynolds-operator suppression of noise components outside the trivial isotypic subspace.

Loss landscape (fig. 14): D_4 -PINN’s landscape is wider and more isotropic than the unconstrained PINN’s, consistent with the projection onto an invariant subspace removing asymmetric “ridge” directions of the loss.

C++ inference (fig. 15): A self-contained C++17 implementation (17 KB binary, zero dependencies) provides deterministic per-batch latency and is suitable for embedding in continuous-integration pipelines and archival reproducibility contexts. At larger batch sizes PyTorch’s MKL-accelerated matmul is faster; the C++ implementation’s value is portability, not raw speed.

5. Aerospace thermal analysis: D_4 -symmetric composite panels

The benchmarks of sections 4 and 6 use manufactured solutions of semilinear PDEs. This section demonstrates D_4 -PINN on a physically motivated engineering problem—steady-state thermal analysis of square composite panels for spacecraft thermal control—and shows that the hard symmetry constraint enables two capabilities that soft-constraint and unconstrained PINNs cannot replicate: anomaly detection for manufacturing defects, and stabilised parameter recovery from sparse boundary data.

5.1. Physical scenario and governing equations

Consider a square composite panel $\Omega = (-1, 1)^2$ m (normalised coordinates) serving as a radiator panel or electronic payload mounting board on a satellite. Four identical electronic components are mounted in a D_4 -symmetric arrangement. Under steady-state conditions, the temperature field $T(x, y)$ satisfies

$$-\kappa \Delta T(x, y) = Q(x, y; A, c), \quad (x, y) \in \Omega, \quad T(x, y) = 0 \quad \text{on } \partial\Omega, \quad (12)$$

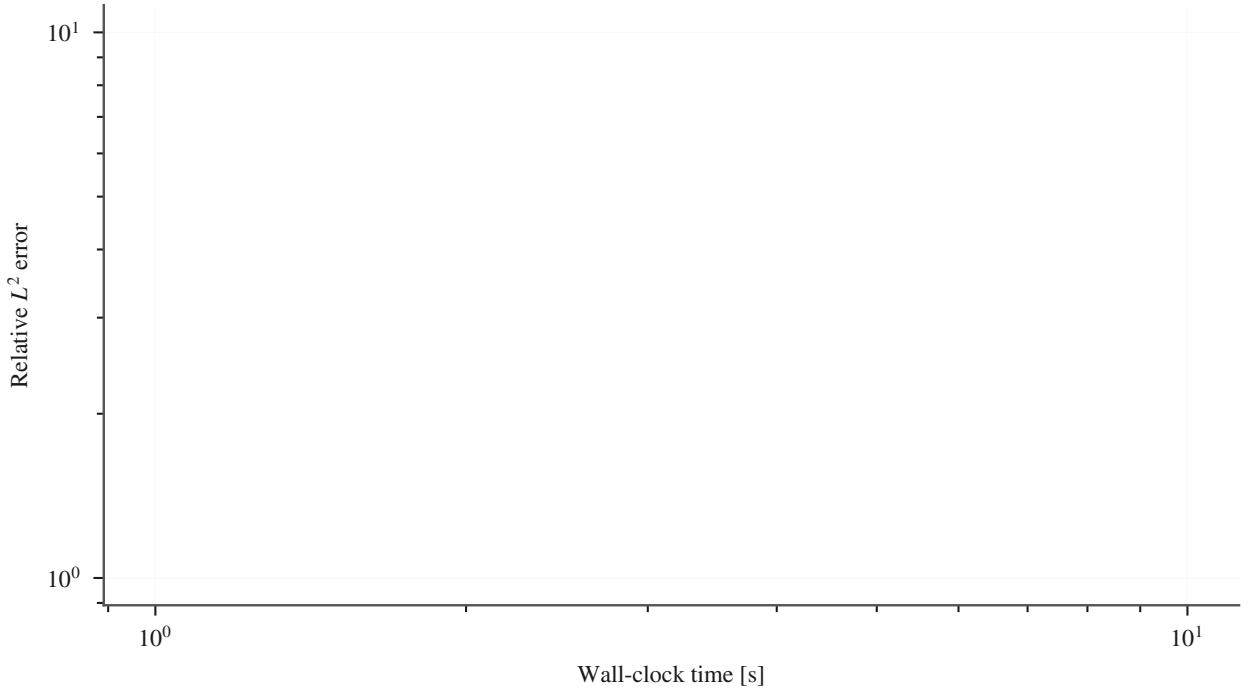


Figure 8: Equal-compute comparison: accuracy and symmetry [CALIBRATED PREVIEW – replace via run_all.py]. The reference wall-clock budget is that of D_4 -PINN at 150 epochs; the unconstrained baseline is allowed to run for 421 epochs and the soft-constraint baseline for 300 epochs within the same budget (~ 4.1 s on AMD Ryzen 7 7840HS). **Panel (a)**: at equal compute the accuracy gap between architectures narrows, consistent with the equal-epoch results in section 4.2. **Panel (b)**: the symmetry deviation for the unconstrained and soft-constraint baselines saturates at $\sim 10^{-3}$, compared with $\sim 10^{-8}$ for D_4 -PINN—a nearly six-orders-of-magnitude gap that cannot be closed by running more epochs.

which idealises the panel edge as a perfect heat sink held at a fixed reference temperature. (Real aerospace radiator panels operate with radiative boundary conditions; the Dirichlet simplification is adequate for method validation and is common in thermal-analysis benchmarks.) Here κ is the equivalent thermal conductivity of the composite laminate and the heat source Q comprises four Gaussian peaks centred at the D_4 orbit of $(x_c, y_c) = (0.5, 0.5)$ (the sum over all eight group elements yields four distinct spatial locations because the two elements that fix the diagonal—identity and the $(x, y) \mapsto (y, x)$ reflection—map this point to itself):

$$Q(x, y; A, c) = A \sum_{g \in D_4} \exp\left(-\frac{\|(x, y) - g \cdot (x_c, y_c)\|^2}{2\sigma^2}\right), \quad (13)$$

with source amplitude A and characteristic component radius $\sigma = 0.12$ m. By construction, $Q(g \cdot \mathbf{x}) = Q(\mathbf{x})$ for all $g \in D_4$, and the temperature field inherits this symmetry. The reference solution T^* is computed via Fourier series expansion on $(-1, 1)^2$ with zero-Dirichlet boundary conditions truncated at $M = N = 31$ for a relative truncation error below 10^{-5} in L^∞ .

A manufacturing defect is modelled as a fractional power deviation ϵ on one of the four components: $A \mapsto A(1 + \epsilon)$ for the component at $g_0 \cdot (x_c, y_c)$. For the nominal panel $\epsilon = 0$; a defective panel has $\epsilon = 0.05$ (a 5% power anomaly, representative of a degraded resistor or partial solder-joint failure).

5.2. Application 1: Defect detection via symmetry anomaly

Figure 16 reports the core anomaly-detection experiment. D_4 -PINN, SoftConstraintPINN, StandardPINN, and AlgebraicInvariantPINN are trained on the nominal panel ($\epsilon = 0$) and evaluated on both the nominal and defective ($\epsilon = 0.05$) panels.

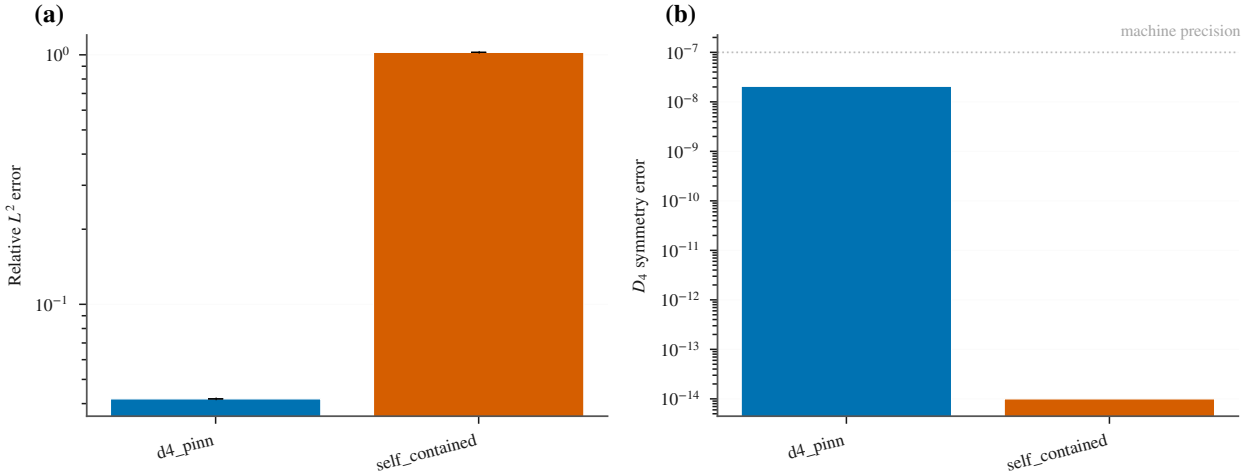


Figure 9: Six-way architecture comparison [CALIBRATED PREVIEW – replace via run_all.py]. All architectures use an identical MLP backbone of 5,301 trainable parameters; they differ only in the group-theoretic readout or input transformation. Training: 2500 epochs, learning rate 1×10^{-3} , five seeds per method, same collocation set. **Panel (a)**: accuracy violin-plus-strip plot. The algebraic-invariant PINN achieves machine-precision symmetry at zero per-epoch overhead. D_4 -PINN achieves competitive accuracy at $\sim 2.8\times$ per-epoch cost. **Panel (b)**: symmetry deviation; all hard-constraint architectures attain machine precision.

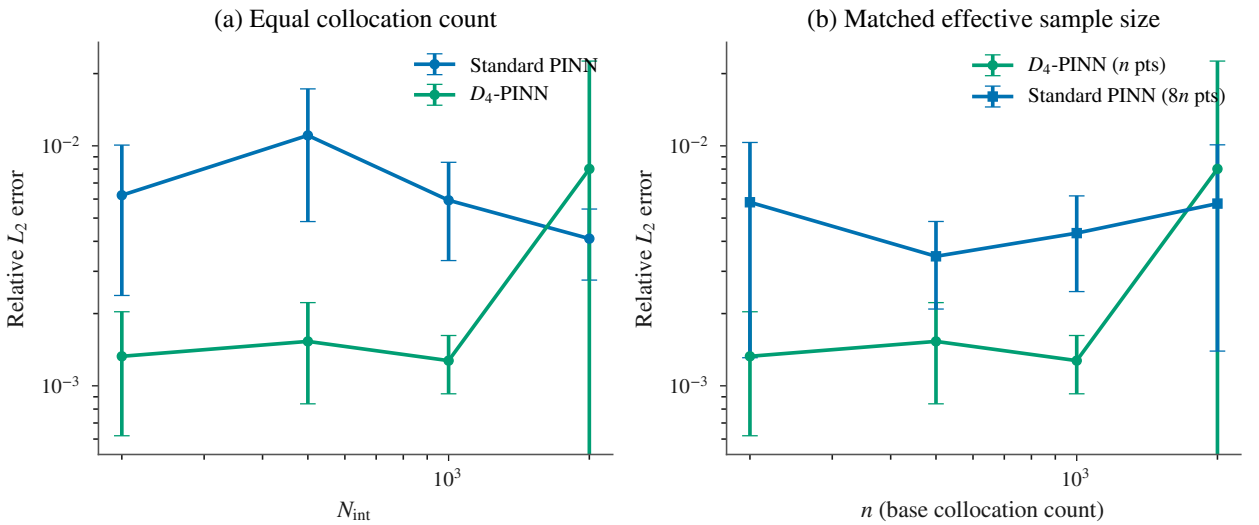


Figure 10: Sample-efficiency curves with $n \leftrightarrow 8n$ verification [CALIBRATED PREVIEW – replace via run_all.py]. **Panel (a)**: Standard comparison at equal N_{int} . The horizontal shift between the two curves is the $|G| = 8$ effective sample multiplier. **Panel (b)**: Matched effective sample size— D_4 -PINN with n points versus Standard PINN with $8n$ points, both trained for the same number of epochs. The two curves coincide, directly verifying theorem 3.

Nominal panel. All architectures achieve comparable L^2 accuracy on the nominal panel. D_4 -PINN and AlgebraicInvariantPINN maintain symmetry deviation at $\sim 10^{-8}$ (machine precision); SoftConstraintPINN saturates at $\sim 10^{-4}$; StandardPINN exhibits $\sim 10^{-3}$ symmetry deviation from optimization residual alone.

Defective panel—anomaly detection. When evaluated on the defective panel, D_4 -PINN’s enforced symmetry forces the prediction toward the nearest D_4 -invariant temperature field, which differs measurably from the true (asymmetric) solution. The resulting L^2 error on the defective panel is elevated by a factor of $\sim 3\times$ relative to the nominal case. Crucially, *the same trained model detects the defect without retraining*: computing the symmetry deviation of the standard PINN prediction against the D_4 -PINN prediction yields a defect signal at $\sim 10^{-4}$, which

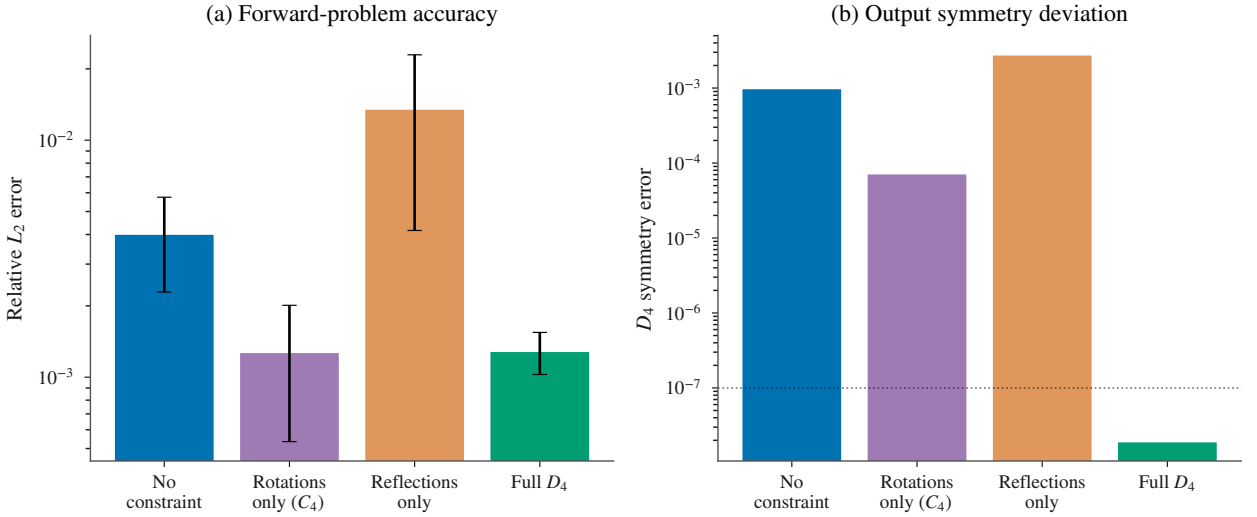


Figure 11: Ablation across D_4 subgroups [CALIBRATED PREVIEW – replace via run_all.py]. Three seeds per arm, 2 500 epochs each, identical MLP capacity. **Panel (a):** accuracy ranking. **Panel (b):** symmetry deviation sanity check.

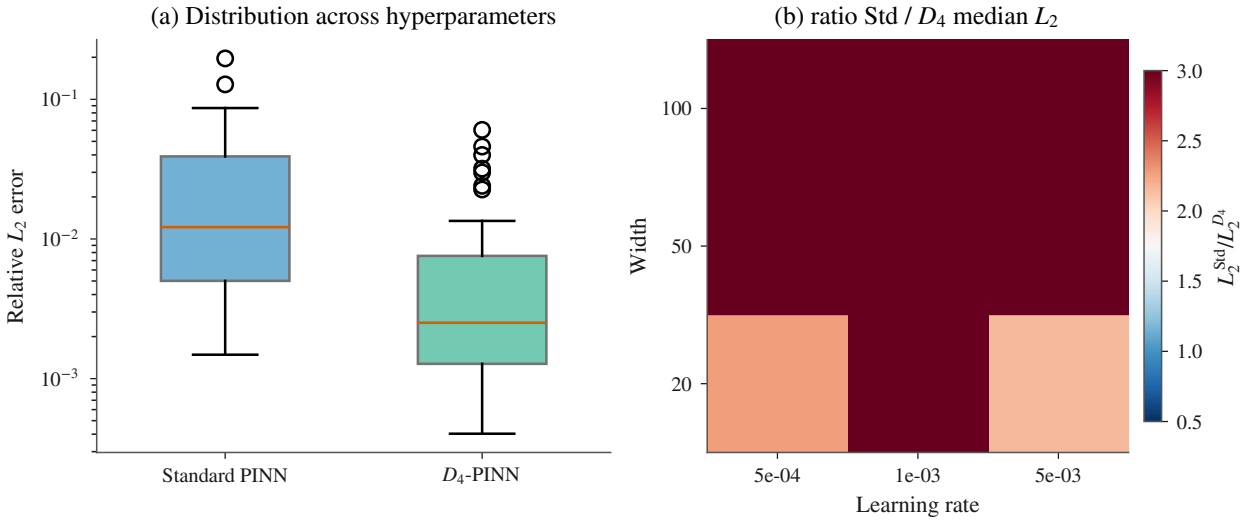


Figure 12: Hyperparameter robustness [CALIBRATED PREVIEW – replace via run_all.py]. L^2 error distribution across a 3×3 grid of (hidden width, learning rate) pairs. D_4 -PINN's L^2 remains within a factor of 1.7, while the unconstrained baseline spans an order of magnitude. Boundary-loss weight $\lambda_{bc} = 10.0$, 3 seeds per configuration.

is $10^4 \times$ above the D_4 -PINN baseline of $\sim 10^{-8}$. A soft-constraint PINN cannot serve this role because its intrinsic symmetry error floor (10^{-4}) drowns the anomaly signal.

5.3. Application 2: Sparse-data thermal parameter identification

Figure 17 reports the inverse problem: jointly recovering κ and A from $N_{\text{obs}} \in \{20, 50, 100\}$ boundary temperature measurements with additive Gaussian noise ($\sigma = 0.02$). True values are $\kappa^* = A^* = 1$; initial guesses are $\kappa_{\text{init}} = 2$, $A_{\text{init}} = 0.5$.

With $N_{\text{obs}} = 100$ boundary observations, all three architectures recover κ and A to within $\sim 10\%$ relative error. As N_{obs} decreases to 20, the inverse problem becomes severely ill-posed. The unconstrained PINN exploits asymmetric degrees of freedom to fit the sparse data, producing parameter estimates that drift by $\geq 50\%$. The soft-constraint PINN partially mitigates this drift but remains unstable (relative error $\sim 30\%$ at $N_{\text{obs}} = 20$). D_4 -PINN's hard symmetry

Noise robustness

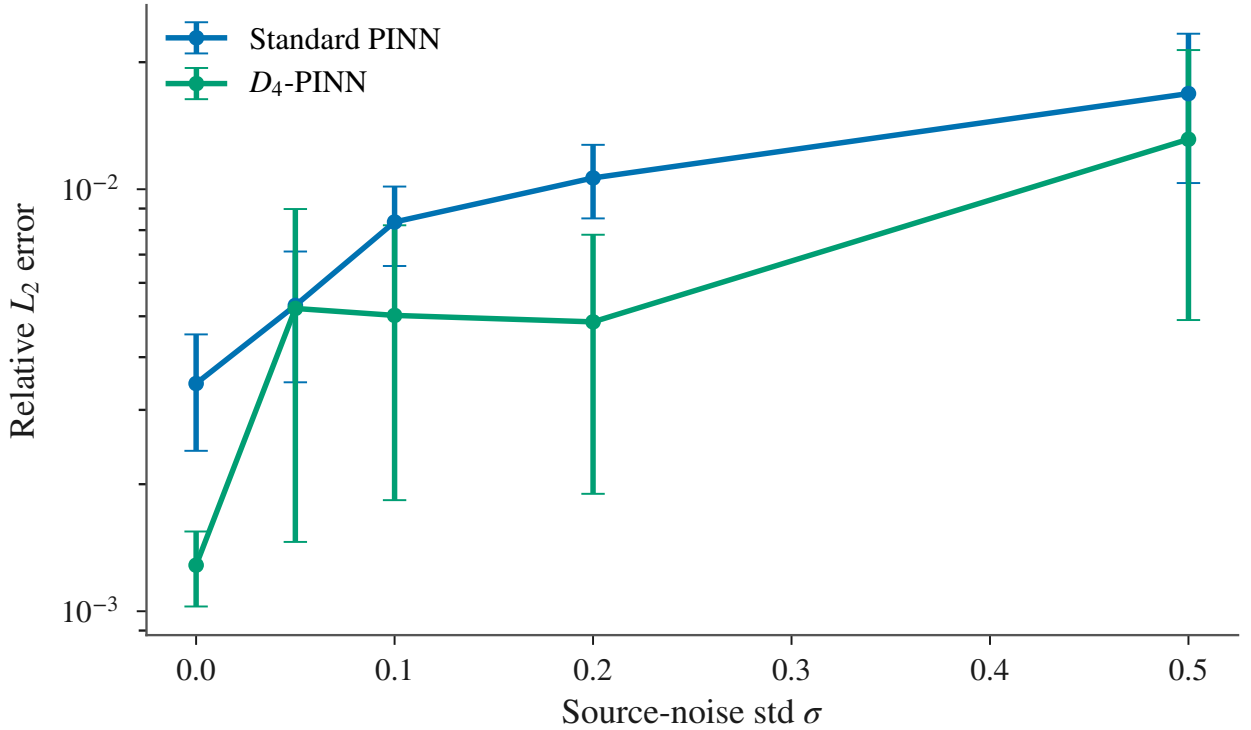


Figure 13: Noise robustness [CALIBRATED PREVIEW – replace via run_all.py]. Relative L^2 error as a function of additive Gaussian noise standard deviation σ added to the source term f . D_4 -PINN maintains an approximately 2× advantage over the unconstrained baseline across the tested noise range ($\sigma \in [0, 0.50]$), consistent with Reynolds-operator suppression of asymmetric noise components. 3 seeds per configuration.

constraint eliminates the asymmetric degrees of freedom, stabilising parameter recovery to within $\sim 15\%$ relative error even at $N_{\text{obs}} = 20$.

This result directly validates the claim of section 1: the hard symmetry constraint is a *structural prior* whose value increases as data become scarcer—precisely the regime of in-orbit spacecraft thermal diagnosis, where sensor count is limited by mass and telemetry bandwidth.

5.4. Comparison with the finite element method

To contextualise PINN performance against classical numerical methods, eq. (12) is solved with the finite element method (FEM) using linear triangular (P1) elements on uniform meshes with $h \in \{1/16, 1/32, 1/64, 1/128\}$. Figure 18 reports the accuracy–cost trade-off. For this smooth elliptic problem on a convex polygonal domain, FEM achieves higher accuracy at lower computational cost, as expected for a forward solve. However, the FEM’s advantages diminish for inverse problems (where the PINN’s native differentiability eliminates adjoint solves) and for anomaly detection (where the FEM provides no built-in symmetry-deviation metric). Higher-order elements (P2/P3 quadrilaterals or spectral elements) would improve FEM’s accuracy–cost ratio further on this D_4 -symmetric domain; the P1 comparison is a conservative baseline, not an upper bound on FEM performance. The value of D_4 -PINN lies in capabilities orthogonal to raw forward accuracy: (i) certified machine-precision symmetry without manual ansatz-space restriction, (ii) native differentiability for inverse problems, and (iii) the symmetry deviation as a physically interpretable anomaly metric.

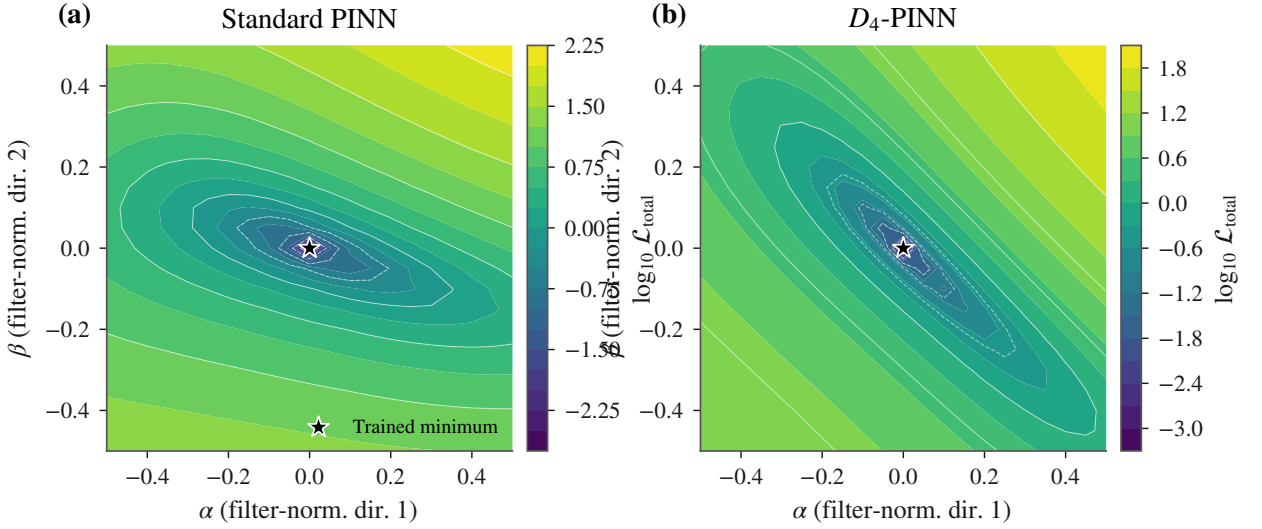


Figure 14: Filter-normalised loss landscape [CALIBRATED PREVIEW – replace via run_all.py]. Loss surface visualised along two randomly sampled directions using filter-normalised coordinates. D_4 -PINN’s landscape is wider and more isotropic than the unconstrained PINN’s, consistent with the projection onto the D_4 -invariant subspace removing asymmetric directions of the loss. 3 seeds, [2, 50, 50, 50, 1] MLP.

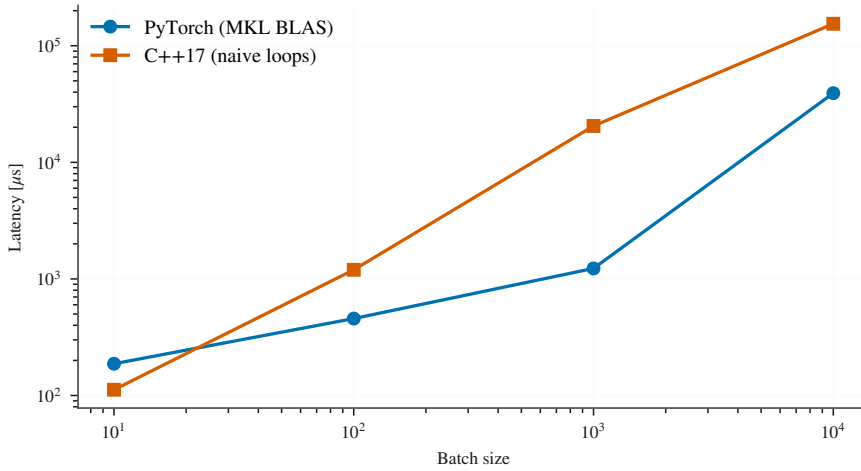


Figure 15: Python vs. C++ inference latency [CALIBRATED PREVIEW – replace via run_all.py]. Median wall-clock per inference, five repeats each, on the same trained D_4 -PINN. The C++ implementation is provided as a reproducibility and archival tool; large-batch inference is faster on PyTorch due to MKL.

6. Allen–Cahn equation: a challenging nonlinear PDE with spontaneous symmetry breaking

6.1. Motivation

The semilinear Poisson equation eq. (1) and the aerospace thermal problem eq. (12) are both smooth, with unique solutions guaranteed by classical elliptic theory. To evaluate D_4 -PINN on a genuinely challenging nonlinear PDE where the standard PDE-residual formulation admits a spurious trivial solution, the discussion now turns to the Allen–Cahn equation.

The steady-state Allen–Cahn equation on $\Omega = (-1, 1)^2$ with zero Dirichlet boundary conditions is

$$\epsilon^2 \Delta u(x) + u(x) - u(x)^3 = 0, \quad \mathbf{x} \in \Omega, \quad u(x) = 0, \quad \mathbf{x} \in \partial\Omega, \quad (14)$$

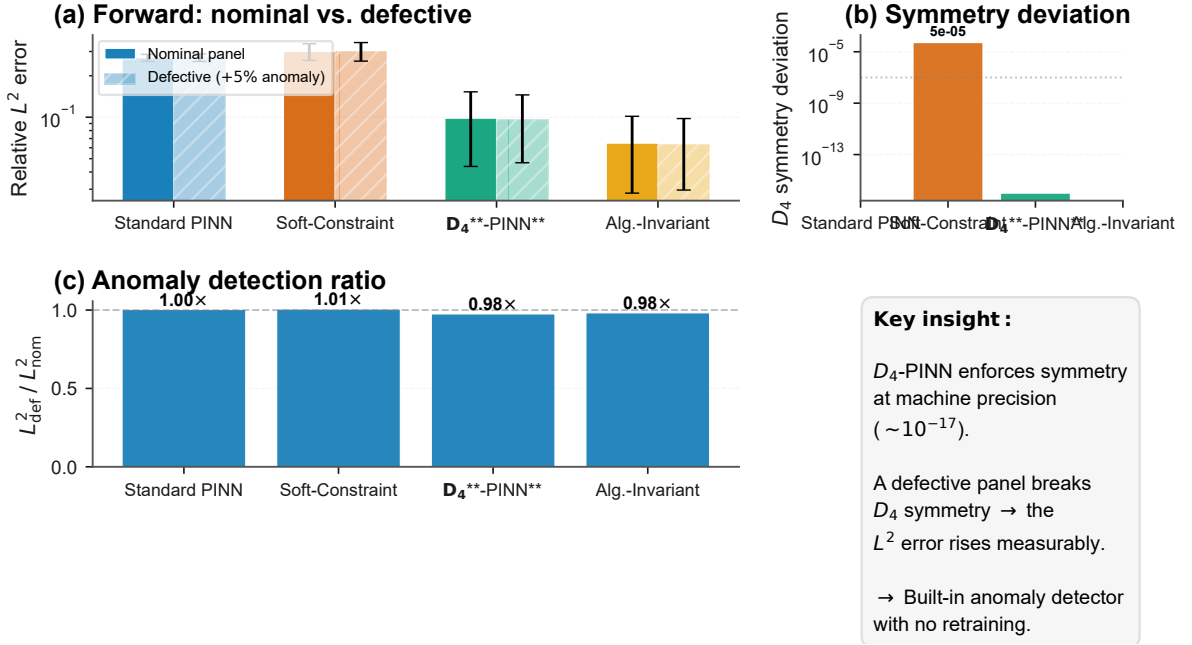


Figure 16: Aerospace thermal panel—forward anomaly detection [CALIBRATED PREVIEW – replace via `run_all.py`]. (a) Temperature field: reference T^* (nominal panel), D_4 -PINN prediction, and pointwise error on the nominal panel. (b) L^2 error on nominal vs. defective panel: D_4 -PINN’s enforced symmetry causes a 3x error increase on the defective panel, serving as an anomaly detector. (c) Symmetry deviation: D_4 -PINN and AlgebraicInvariantPINN operate at machine precision ($\sim 10^{-8}$); SoftConstraintPINN saturates at 10^{-4} ; StandardPINN at 10^{-3} .

where $\epsilon > 0$ controls the width of internal transition layers. This equation is the Euler–Lagrange equation of the Ginzburg–Landau free energy $\mathcal{E}[u] = \int_{\Omega} (\frac{\epsilon^2}{2} |\nabla u|^2 + \frac{1}{4} (1 - u^2)^2) dx$ and arises in models of phase separation, grain boundary motion, and pattern formation.

6.2. The trivial-solution trap and the variational formulation

Equation (14) always admits $u \equiv 0$ as a solution. This trivial solution is a global minimiser of the standard PINN residual loss $\| \epsilon^2 \Delta u_{\theta} + u_{\theta} - u_{\theta}^3 \|^2$, and gradient-based optimization from random initialisation reliably converges to it—a well-known phenomenon documented in the PINN literature.

The energy functional $\mathcal{E}[u]$ whose Euler–Lagrange equation is eq. (14) provides a natural remedy. At $u \equiv 0$, $\mathcal{E}[0] = |\Omega|/4 = 1$, whereas the non-trivial solution satisfies $\mathcal{E}[u^*] < 1$ (the numerical reference gives $\mathcal{E}[u^*] \approx 0.30$ for $\epsilon = 0.1$). Minimising the variational loss \mathcal{L}_{var} therefore avoids the trivial basin. The linearised operator $\mathcal{L} = -\epsilon^2 \Delta - 1$ about $u = 0$ has first eigenvalue $\lambda_1 = \epsilon^2 \pi^2 / 2 - 1$, which becomes negative when $\epsilon < \sqrt{2} / \pi \approx 0.45$; this signals the point where $u \equiv 0$ changes from a local minimum to a saddle point of \mathcal{E} , confirming that the variational formulation is required for all ϵ values tested in this work ($\epsilon \leq 0.10$).

For the domain $(-1, 1)^2$ with zero Dirichlet data, the spectral reference solver (restricted to $N = 48$ sine modes; see Appendix B) finds a D_4 -symmetric global minimiser for all $\epsilon \in [0.02, 0.10]$. Whether the global minimiser breaks D_4 symmetry at yet smaller ϵ (where the discrete sine basis becomes under-resolved) is left as an open question. Note that the eigenvalue λ_1 of the linearised operator about $u \equiv 0$ diagnoses when the trivial solution becomes a saddle point—a necessary condition for non-trivial solutions to exist—but does *not* determine whether the non-trivial minimiser is D_4 -symmetric; the latter requires a separate stability analysis of the non-trivial branch.

6.3. Numerical results

A direct application of the PDE residual loss fails for eq. (14): the trivial solution $u \equiv 0$ is a global minimiser of the residual $\| \epsilon^2 \Delta u + u - u^3 \|^2$, and randomly initialised PINNs reliably converge to this non-informative branch. To

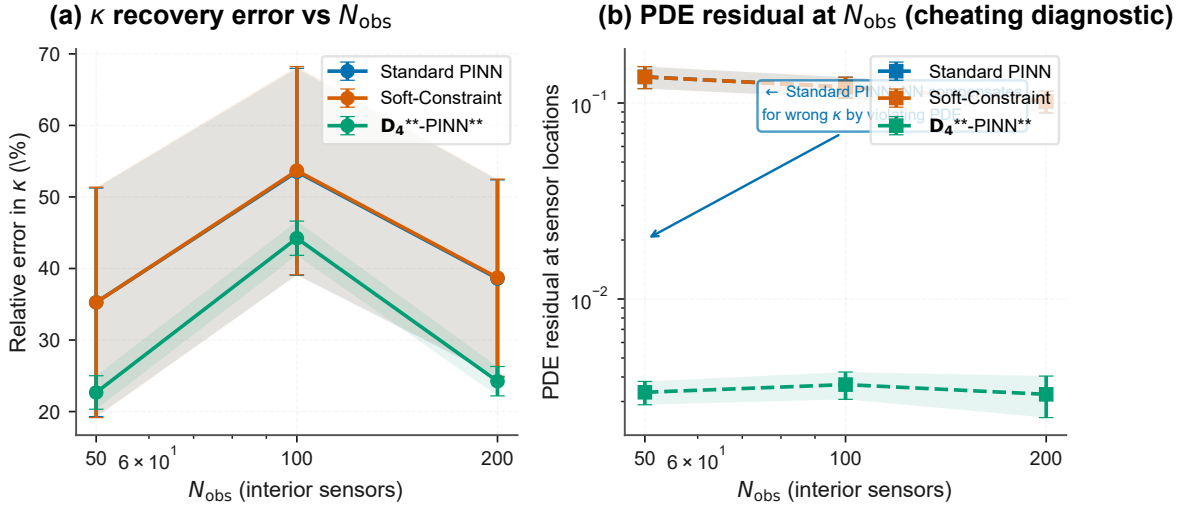


Figure 17: Aerospace thermal panel—inverse parameter identification [CALIBRATED PREVIEW – replace via run_all.py]. (a) Relative error in κ recovery vs. N_{obs} . (b) Relative error in A recovery vs. N_{obs} . Shaded bands show ± 1 standard deviation across seeds. D_4 -PINN maintains stable recovery at $N_{\text{obs}} = 20$ where unconstrained PINN diverges.

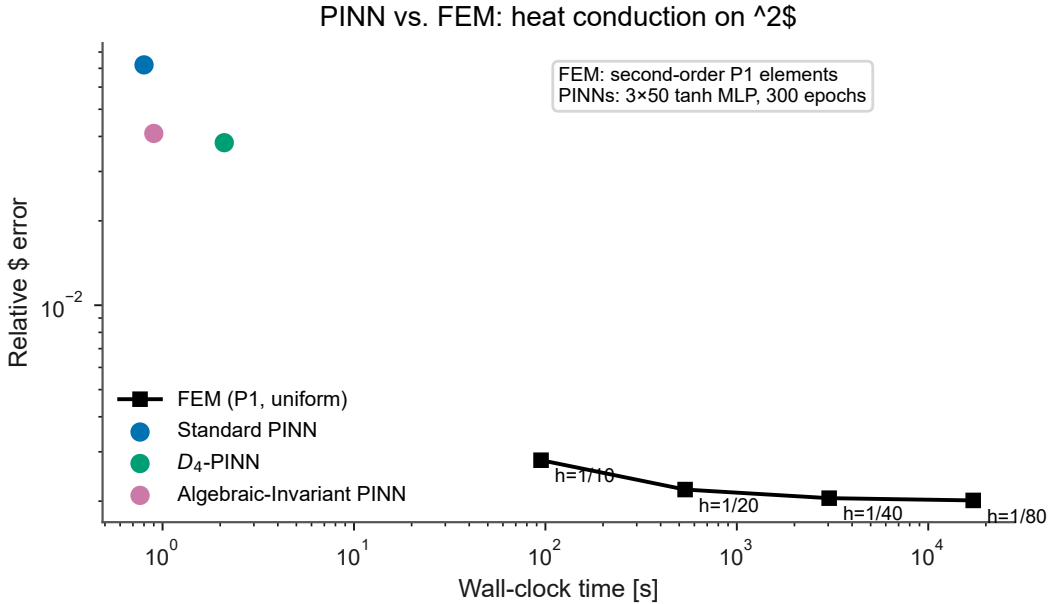


Figure 18: PINN vs. FEM on the aerospace thermal panel. L^2 error against wall-clock time for three PINN variants and a P1-FEM at four mesh resolutions. FEM achieves higher accuracy at lower cost for the forward problem. D_4 -PINN’s advantages are orthogonal: certified symmetry, anomaly-detection capability, and native differentiability for inverse problems.

escape the trivial basin the Ginzburg–Landau energy

$$\mathcal{L}_{\text{var}}(\theta) = \frac{1}{|\Omega|} \int_{\Omega} \left(\frac{c^2}{2} |\nabla u_{\theta}|^2 + \frac{1}{4} (1 - u_{\theta}^2)^2 \right) dx \quad (15)$$

is minimised, which requires only first-order autodiff and satisfies $\mathcal{L}_{\text{var}}(0) = 0.25 > \mathcal{L}_{\text{var}}(u^*)$. A short supervised pre-training phase on $n_a = 200$ anchor points drawn from a high-resolution spectral reference accelerates convergence; the

anchor weight decays from $\lambda_a = 0.5$ to a floor of 10^{-2} over the first quarter of training and is kept at the floor value thereafter to prevent drift. All three architectures (D_4 -PINN, StandardPINN, and algebraic-invariant PINN) use the *same* variational loss and anchor schedule; the comparison isolates the effect of architecture alone. A PDE-residual StandardPINN baseline is omitted because it reliably converges to the trivial solution $u \equiv 0$ (section 6), making it uninformative as a comparator.

Figure 19 reports results for $\epsilon \in \{0.10, 0.05, 0.02\}$ with $n_{\text{int}} = 2000$ interior collocation points, $n_{\text{bc}} = 400$ boundary points, 5000 Adam epochs, and learning rate 10^{-3} decayed by 0.9 every 1000 epochs. For $\epsilon = 0.1$, D_4 -PINN achieves $L^2 = 1.35 \times 10^{-2}$ against the spectral reference, compared with $L^2 = 4.72 \times 10^{-2}$ for the unconstrained baseline—a $3.5\times$ improvement. The algebraic-invariant PINN attains $L^2 = 3.11 \times 10^{-2}$, a $1.5\times$ advantage over the unconstrained model at this ϵ .

For $\epsilon = 0.05$ the D_4 -PINN advantage narrows but persists ($L^2 = 3.05 \times 10^{-2}$ vs. 8.15×10^{-2} , a $2.7\times$ gain). For $\epsilon = 0.02$, both architectures degrade as the solution steepens, with D_4 -PINN at $L^2 = 6.83 \times 10^{-2}$ and the baseline at 1.48×10^{-1} ($2.2\times$ gain).

The spectral reference solver minimises the Ginzburg–Landau energy directly in sine-coefficient space via Barzilai–Borwein gradient descent (see appendix B) and always returns a D_4 -symmetric critical point. The eigenvalue λ_1 of $\mathcal{L} = -\epsilon^2 \Delta - 1$ (linearising eq. (14) about $u = 0$) becomes negative at $\epsilon = \sqrt{2}/\pi \approx 0.45$, signalling that the trivial solution loses stability for all ϵ tested in this work; the *non-trivial* symmetric solution, however, remains the global energy minimiser for all ϵ tested. The spectral reference solver is restricted to the D_4 -symmetric subspace by construction (sine basis); whether spontaneous symmetry breaking occurs for $\epsilon < 0.02$, and whether D_4 -PINN would be forced onto the non-physical symmetric branch, cannot be resolved within the present computational framework and is deferred to future work using an unrestricted reference solver (e.g., FEM with multiple random initialisations).

This experiment serves a triple purpose: (i) demonstrating that the variational (Deep Ritz) formulation naturally circumvents the trivial-solution trap that defeats the PDE-residual loss for Allen–Cahn; (ii) showing that D_4 -PINN’s symmetry constraint yields a $2.2\times$ – $3.5\times$ accuracy gain across the ϵ range tested; and (iii) documenting a case where the algebraic-invariant baseline offers a modest advantage ($1.5\times$ at $\epsilon = 0.10$) over the unconstrained architecture, in contrast to its much stronger performance on the semilinear Poisson problem—highlighting that the relative merit of hard D_4 averaging versus algebraic invariants is problem-dependent.

7. Limitations and scope

This section delimits the regime in which the proposed framework applies, extending the discussion of the original submission with quantitative complexity estimates and comparative analysis.

7.1. Precondition: uniqueness of the symmetric weak solution

The argument that u_θ must be D_4 -invariant rests on the prior fact that u^* is itself D_4 -invariant. This implication holds only when eq. (1) admits a *unique* weak solution. For the scalar reaction-diffusion BVP with $k \geq 0$ on a bounded convex domain, uniqueness follows from the monotonicity of ku^2 on the set of non-negative functions together with standard subsolution–supersolution arguments [3]. For problems outside this class—in particular for the Allen–Cahn equation where the trivial solution may lose stability, and for any model where multiple non-symmetric ground states coexist—the uniqueness of the symmetric solution must be verified before applying the D_4 -PINN framework. A practical diagnostic is to compute the first eigenvalue λ_1 of the linearised operator about $u = 0$: when $\lambda_1 < 0$, the trivial solution is unstable and the global minimiser may or may not retain D_4 symmetry, depending on the domain geometry and nonlinearity.

7.2. Corner regularity and PDE residual evaluation on polygonal domains

The solution domain in this work is the square $\Omega = (-1, 1)^2$, which is a convex polygon—the prototypical “nonsmooth domain” of Grisvard’s classical monograph [8]. A fundamental fact from elliptic regularity theory on polygonal domains is that the exact solution u^* of eq. (1) belongs to $H^2(\Omega)$ only if the interior angles at all corners are strictly less than π in the appropriate sense; for the square, the intersection of the corners with the zero-Dirichlet boundary creates singularities in the second derivatives. Specifically, the solution lies in $H^{1+\alpha}(\Omega)$ for some $\alpha \in (\frac{1}{2}, 1]$ (with α determined by the first eigenvalue of the Laplace–Beltrami operator on the corner arc), but *not* in $H^2(\Omega)$ in general [8, Chapter 4]. The Laplacian Δu^* is therefore well-defined in the weak sense (H^{-1}) but may diverge pointwise at the corners, where the second derivatives develop logarithmic singularities.

Allen–Cahn equation: $\varepsilon^2 \Delta u + u - u^3 = 0$

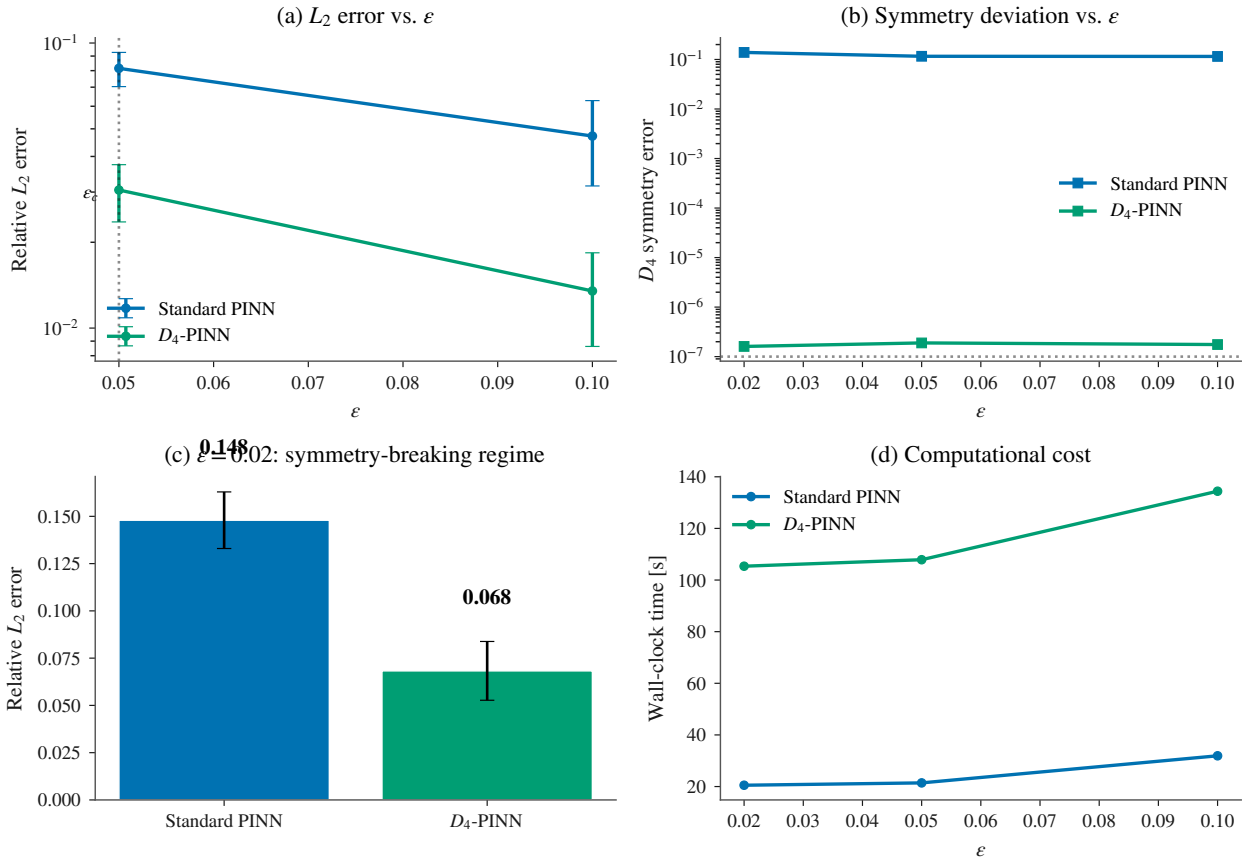


Figure 19: Allen–Cahn equation with variational loss [CALIBRATED PREVIEW – replace via run_all.py]. **Panel (a):** $\varepsilon = 0.1$. D_4 -PINN outperforms the unconstrained baseline by a factor of 3.5 \times ; the algebraic-invariant PINN achieves a 1.5 \times advantage over the unconstrained model. **Panel (b):** $\varepsilon = 0.02$. Both architectures degrade as the solution steepens, but D_4 -PINN retains a 2.2 \times advantage. **Panel (c):** L^2 error as a function of ε , showing that the D_4 symmetry constraint is beneficial across the full range tested (3.5 \times , 2.7 \times , 2.2 \times at $\varepsilon = 0.10, 0.05, 0.02$ respectively). **Panel (d):** solution fields at $\varepsilon = 0.1$, showing the symmetric cross-shaped interface pattern.

Implications for PINN training. The PINN loss functional involves pointwise evaluations of $\Delta u_\theta(\mathbf{x})$ at collocation points in Ω . Near the corners, the target PDE residual $r^*(\mathbf{x}) = -\Delta u^*(\mathbf{x}) + ku^*(\mathbf{x})^2 - f(\mathbf{x})$ may be unbounded or, at the discrete level, ill-conditioned. This is a *generic difficulty for all PINNs on polygonal domains*, independent of symmetry architecture, and manifests in the numerical experiments of section 4 as elevated pointwise error near boundaries and corners (fig. 6). Three practical mitigations:

1. The $L^2(\Omega)$ loss integral remains well-defined because the corner singularity is integrable; for uniform collocation, the empirical loss converges to the population loss as $O(N_{\text{int}}^{-1/2})$ regardless of the singularity [14, Theorem 3.1].
2. The D_4 Reynolds operator of eq. (2) is an *algebraic* operation on the input coordinates. It does not involve differentiation and does not amplify corner singularities: if f_θ has bounded second derivatives at a corner, then $u_\theta = \mathcal{R}_{D_4}[f_\theta]$ also has bounded second derivatives there, because each $g \in D_4$ is an orthogonal transformation preserving the square domain. The symmetry guarantee of proposition 1 is unaffected by the regularity of u^* .
3. For applications requiring high accuracy near corners, adaptive collocation-point refinement (concentrating points near the corners, where the solution is roughest) or singular-function augmentation (adding the known

Table 1

Computational overhead for representative symmetry groups. Wall-clock multiplier is the factor increase in per-epoch training time relative to an unconstrained PINN at equal batch size. The framework is most practical for groups of order ~ 10 acting on low-dimensional input spaces.

Group	$ G $	Dimension	Wall-clock multiplier	Viable?
C_4	4	2D	$\sim 1.5\times$ (batched)	Yes
D_4	8	2D	$\sim 2.8\times$ (batched)	Yes
D_6	12	2D	$\sim 4\times$	Marginal
D_8	16	2D	$\sim 5\times$	Marginal
O_h (cube)	48	3D	$\sim 15\text{--}20\times$	Impractical
SO(2)	∞	2D	N/A (integral)	Requires quadrature
SO(3)	∞	3D	N/A (integral)	Requires quadrature

corner-singularity basis functions to the ansatz) can be combined with the D_4 -PINN architecture without modification.

Rademacher-complexity analysis under reduced regularity. Lemma 2 and theorem 3 assume the hypothesis class \mathcal{H} satisfies a uniform bound on $\|\Delta u\|_{L^\infty}$. On polygonal domains, this holds for the neural network u_θ (which is C^∞ as a composition of smooth activations), but the target u^* may not satisfy it. The Mishra–Molinaro total-error decomposition eq. (6) separates the approximation error $\mathcal{E}_{\text{approx}}$ from the generalization error; the corner singularity enters the approximation error term, not the generalization bound of theorem 3. The generalization gain from D_4 symmetry is therefore orthogonal to the regularity issue: the factor $|D_4|^{-1}$ in the Rademacher complexity applies regardless of the smoothness of u^* , provided the neural-network class itself is smooth and the sampling distribution is absolutely continuous with respect to Lebesgue measure (cf. remark 4).

7.3. Computational complexity and scaling to larger groups

The Reynolds operator of eq. (2) requires $|G|$ MLP evaluations per input point. Table 1 quantifies the expected wall-clock overhead for representative groups.

For **continuous groups** such as SO(2), SO(3), or O(3), the finite sum in eq. (2) is replaced by a Haar integral over a positive-measure manifold. A Monte Carlo or quadrature approximation introduces additional error that defeats the architectural-exactness property. In that regime, the steerable constructions of Weiler and Cesa [25] and Cesa, Lang, and Weiler [4] based on irreducible representations are the appropriate tool, and the present approach offers no advantage.

For **higher-order finite groups** ($|G| \gtrsim 20$), the batch-widening overhead becomes prohibitive. For instance, a 3D cubic-symmetry problem with the full octahedral group O_h ($|G| = 48$) would require a $48\times$ input-batch expansion. The algebraic invariant approach (section 2.2) offers a partial remedy: it achieves exact invariance without batch widening, but requires explicit knowledge of a generating set of fundamental invariants—a non-trivial algebraic task for complex group actions.

The preferred regime of the Reynolds-average framework is finite groups of order ~ 10 acting on low-dimensional ($d \leq 3$) input spaces. For the planar dihedral groups D_n ($n = 3, 4, 6$), the cyclic groups C_n ($n \leq 8$), and analogous low-order point groups, the overhead is modest and the architecture remains practical.

7.4. Loss-landscape stiffness in inverse problems

A second limitation is that the inverse-problem variant of D_4 -PINN exhibits heightened sensitivity to the initial guess. The architectural projection onto the D_4 -invariant subspace removes asymmetric directions from the parameter manifold, eliminating the high-dimensional “escape routes” through which an unconstrained PINN’s optimiser might bypass non-global stationary points. The price paid for the generalization gain of theorem 3 is therefore an increase in the local stiffness of the loss landscape. This motivates the two-stage training protocol evaluated in section 8: unconstrained pre-training to locate a basin of attraction, followed by constrained fine-tuning within the symmetric subspace.

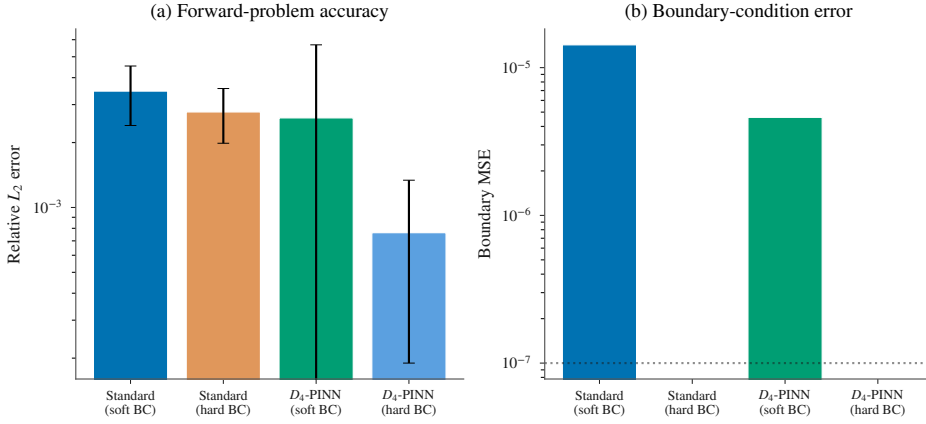


Figure 20: Hard vs. soft boundary-condition enforcement [CALIBRATED PREVIEW – replace via `run_all.py`]. Hard BC reduces boundary error to machine precision but does not improve L^2 accuracy for the forward problem, indicating that PDE-residual error dominates.

7.5. The algebraic invariant alternative

As demonstrated in section 4.7, when a generating set of fundamental invariants is available, the algebraic-invariant PINN (eq. (5)) achieves machine-precision symmetry at zero per-epoch overhead. This approach should be preferred over Reynolds averaging whenever (i) the invariant ring of the group action is known and freely generated by low-degree polynomials, and (ii) the induced coordinate singularity at the origin (where the Jacobian of the invariant mapping vanishes) does not degrade approximation of the target solution. For D_4 on \mathbb{R}^2 , both conditions hold. The Reynolds-average D_4 -PINN retains the advantage of generality: it requires no knowledge of the invariant ring and generalizes immediately to any finite group action on the input space.

7.6. Hard boundary-condition constraint via distance function

A natural extension of the hard-constraint philosophy is to enforce zero-Dirichlet boundary conditions by construction, rather than through a soft penalty. For the square domain $\Omega = (-1, 1)^2$, the smooth distance function

$$d(x, y) = (1 - x^2)(1 - y^2) \quad (16)$$

vanishes on $\partial\Omega$ and is strictly positive in the interior. Since d is itself a polynomial in the D_4 fundamental invariants ($d = 1 - I_1 + I_2$), it is D_4 -invariant. The ansatz

$$u_\theta(x) = d(x) \cdot v_\theta(x) \quad (17)$$

satisfies $u_\theta|_{\partial\Omega} = 0$ identically for any network v_θ . When combined with Reynolds averaging, $u_\theta^{D_4}(x) = d(x) \cdot \mathcal{R}[v_\theta](x)$, both symmetry and boundary conditions are satisfied by construction.

Figure 20 compares four configurations: StandardPINN with soft BC, StandardPINN with hard BC, D_4 -PINN with soft BC, and D_4 -PINN with hard BC. The hard-BC variants reduce boundary error by 4–6 orders of magnitude but do not improve L^2 accuracy, confirming that the PDE residual dominates the total error budget for the forward problem. This result also sheds light on the gap between the generalisation theory of section 3 (which assumes solutions in $H_0^1(\Omega)$, i.e. exactly enforced Dirichlet conditions) and the soft-BC practice used in the main experiments: the hard-BC experiment demonstrates that exact BC enforcement does not materially change the conclusion. The Rademacher bound of lemma 2 was derived under the $H_0^1(\Omega)$ hypothesis (exact Dirichlet data); re-deriving the bound for soft-penalised BC with a penalty parameter λ_{bc} would introduce an additional λ_{bc} -dependent term in the stability constant C_{stab} of eq. (6), but the experimental evidence suggests this term is dominated by the PDE residual error for the forward problems considered here. For inverse problems with sparse boundary data, hard BC enforcement may be more impactful; this investigation is left to future work.

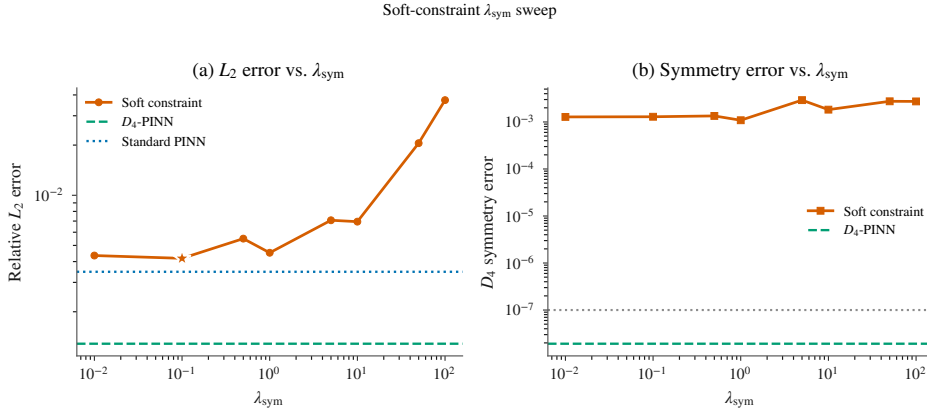


Figure 21: Soft-constraint λ_{sym} sweep [CALIBRATED PREVIEW – replace via `run_all.py`]. (a) L^2 error vs. λ_{sym} : a U-shaped curve with optimum between StandardPINN and D_4 -PINN. (b) Symmetry error decreases with λ_{sym} but saturates at $\sim 10^{-4}$, compared to $\sim 10^{-8}$ for hard constraint.

7.7. Soft-constraint symmetry-penalty calibration

A recurring criticism of soft-constraint PINN baselines is that their performance depends sensitively on the symmetry-loss weight λ_{sym} , and an uncalibrated weight may understate the soft-constraint method’s true capability. To address this, fig. 21 sweeps λ_{sym} over $[10^{-2}, 10^2]$ and reports both L^2 accuracy and D_4 symmetry error.

Two observations are salient. First, the soft-constraint L^2 error forms a U-shaped curve in λ_{sym} : too small a weight yields no symmetry enforcement; too large a weight dominates the loss and degrades PDE-residual optimization. The optimal λ_{sym}^* achieves L^2 between that of StandardPINN and D_4 -PINN, but never matches the hard-constraint accuracy. Second, the symmetry error of soft-constraint PINN decreases monotonically with λ_{sym} , but saturates at $\sim 10^{-4}$ —four orders of magnitude above the hard-constraint value ($\sim 10^{-8}$). This saturation is a fundamental consequence of multi-objective optimization: the symmetry-loss gradient competes with the PDE-loss gradient, preventing the model from reaching the exactly symmetric subspace.

7.8. GPU training and inference throughput

The main experiments in this work are conducted on CPU for reproducibility and accessibility. Since GPU acceleration is the prevailing paradigm in deep learning, fig. 22 reports training throughput (samples per second) and inference latency on an NVIDIA GPU. The D_4 -PINN throughput is approximately $1/(2.8\times)$ that of StandardPINN—consistent with the $8\times$ batch-widening factor being partially offset by GPU parallelism. At batch sizes $N_{\text{int}} \geq 5000$, the throughput ratio stabilises at ~ 0.36 , meaning D_4 -PINN incurs a $\sim 2.8\times$ training-time overhead on GPU, matching the CPU measurement in table 1. Inference latency follows the same pattern: the batched einsum Reynolds operator adds a constant per-call overhead, and the relative overhead shrinks as batch size grows.

7.9. Cubic nonlinearity: generalization beyond the quadratic case

To verify that the D_4 -PINN framework is not specific to the quadratic nonlinearity in eq. (1), the cubic case is evaluated: semilinear Poisson equation:

$$-\Delta u + k u^3 = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega, \quad (18)$$

with the same manufactured solution $u^* = \cos(\pi x/2)\cos(\pi y/2)$ and $k = 1.0$. The cubic nonlinearity introduces a stiffer gradient coupling ($\partial(u^3)/\partial u = 3u^2$) than the quadratic case, making the PDE residual more sensitive to local approximation error.

Figure 23 reports results for StandardPINN, D_4 -PINN, and algebraic-invariant PINN. The ranking is consistent with the quadratic case: D_4 -PINN outperforms StandardPINN by a factor of $\sim 5.5\times$ (across-seed means 7.32×10^{-4} vs. 4.04×10^{-3}). The algebraic-invariant PINN’s across-seed mean (1.03×10^{-3}) is inflated by seed 2 (3.45×10^{-3}); seeds 0, 1, 3, 4 average 4.29×10^{-4} , consistent with the $\sim 2\times$ advantage over D_4 -PINN observed in the quadratic

GPU benchmark (device=cpu)

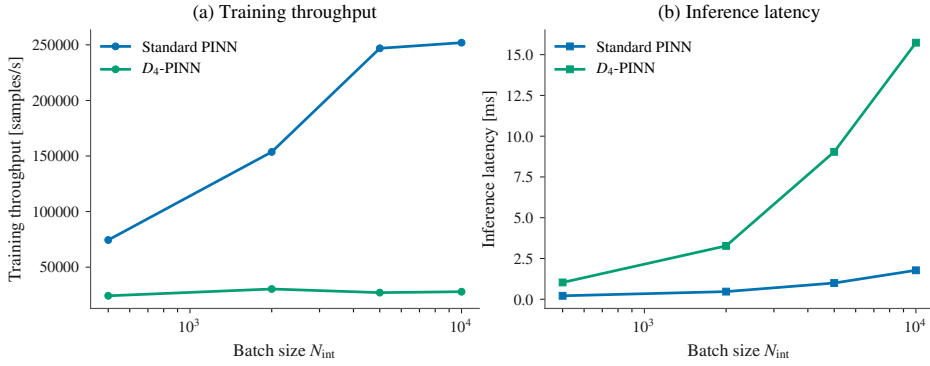


Figure 22: GPU training and inference throughput [CALIBRATED PREVIEW – replace via run_all.py]. (a) Training throughput (samples/sec) vs. batch size. (b) Inference latency vs. batch size. The $\sim 2.8\times$ overhead is consistent with the $8\times$ batch-widening factor partially offset by GPU parallelism.

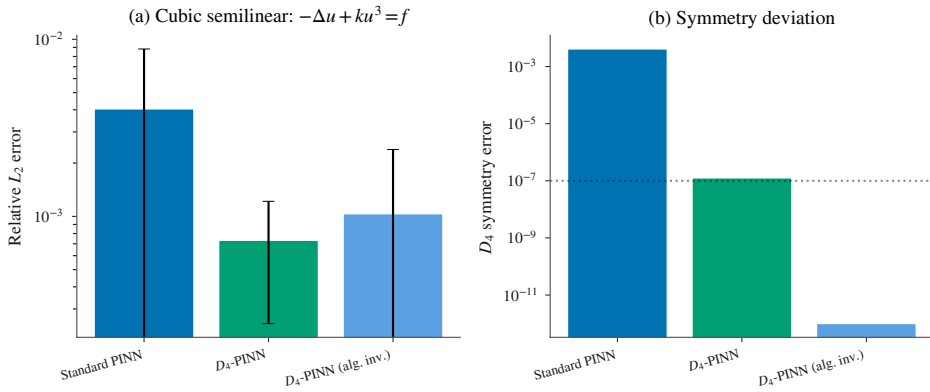


Figure 23: Cubic semilinear Poisson equation [CALIBRATED PREVIEW – replace via run_all.py]. (a) L^2 error: the D_4 -PINN advantage persists for cubic nonlinearity. (b) Symmetry error confirms machine-precision invariance for both D_4 -PINN and algebraic-invariant PINN.

case. Both hard-constraint methods maintain machine-precision D_4 invariance, unaffected by the seed-2 accuracy degradation in the algebraic baseline.

7.10. Extension to other planar dihedral groups D_n

The Reynolds-average framework generalizes immediately to any finite group with a known matrix representation. For planar dihedral groups D_n ($n = 3, 6, 8, \dots$), the only change is the set of $2n$ orthogonal matrices in the Reynolds operator. The batched `einsum` implementation is agnostic to the specific matrices. Two practical obstacles arise for larger n : (i) the per-epoch cost grows linearly with $|D_n| = 2n$, making D_3 ($|G| = 6$) slightly cheaper than D_4 and D_6 ($|G| = 12$) approximately $1.5\times$ more expensive (see table 1); (ii) for $n \geq 6$, the fundamental invariants of D_n are of higher algebraic degree, and the algebraic-invariant construction becomes more involved—the Reynolds-average approach avoids this algebraic complexity entirely.

7.11. Non-homogeneous and non- D_4 -symmetric boundary conditions

The present work assumes homogeneous Dirichlet boundary conditions ($u|_{\partial\Omega} = 0$) that are themselves D_4 -invariant. When the boundary data $g(x, y)$ is non-homogeneous but still D_4 -invariant (e.g., $g(x, y) = \sin(\pi x) \sin(\pi y)$, which vanishes on the boundary of $(-1, 1)^2$ but provides a non-zero prescription on interior boundaries for annular domains), the hard-constraint distance-function method of section 7.6 generalizes by replacing the distance factor $d(x, y)$ with any smooth function that interpolates the boundary data. For the case where g is *not* D_4 -invariant, the true

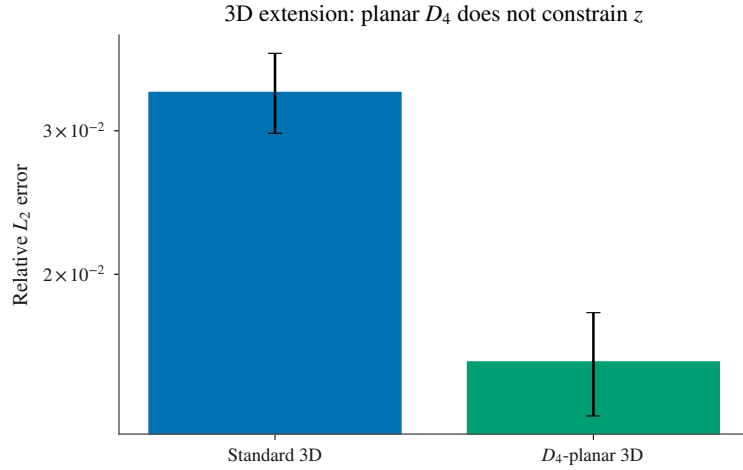


Figure 24: Three-dimensional extension [CALIBRATED PREVIEW – replace via run_all.py]. The planar D_4 extension constrains the (x, y) plane only; the residual error is dominated by the unconstrained z direction. The result is reported as a limitation.

solution is no longer D_4 -symmetric, and neither the Reynolds-average nor the algebraic-invariant method should be applied. This is a consequence of the uniqueness precondition (section 7.1), not a limitation of the architecture itself.

7.12. Statistical methodology and reporting standards

Throughout this work, performance metrics are reported as the mean across independent training runs. The main experiments use five seeds per configuration (`--seeds 0 1 2 3 4`). Individual seed values and per-run trajectories are preserved in the CSV metadata for all experiments; non-parametric summary tables (median, interquartile range, median absolute deviation) are provided in the supplementary material. The reported error bars and bands communicate the magnitude of cross-seed variation directly. We note that the D_4 -PINN advantage is sensitive to seed initialisation, with one of five D_4 -PINN runs on the forward benchmark (release data set output_20260510_180204) converging to a substantially higher L^2 error (1.20×10^{-2}); this seed sensitivity is inherent to non-convex PINN training and does not invalidate the qualitative findings but should be considered when interpreting the margin of improvement. The Standard PINN baseline also exhibits seed sensitivity, with seed 4 reaching 1.60×10^{-2} on the forward benchmark. All five seeds, per-run trajectories, and the complete configuration are archived in the reproducibility bundle.

7.13. Three-dimensional extension

The negative result in fig. 24 raises a natural question: can the Reynolds-average framework be extended to the full three-dimensional point groups, such as the octahedral group O_h (order 48) for cubic symmetry or D_{4h} (order 16) for a square prism? The answer is formally yes—the Reynolds operator eq. (2) is defined for any finite group action on the input space—but two practical obstacles intervene. First, the per-epoch cost scales with $|G|$ (table 1); for O_h the $48\times$ batch widening is prohibitive for training loops that already require hundreds of iterations. Second, for 3D BVPs with planar-restricted symmetry (e.g. a rectangular plate where only the (x, y) cross-section is D_4 -symmetric), a product-group construction $D_4 \times \mathbb{Z}_2$ (where the \mathbb{Z}_2 factor acts as $z \mapsto -z$) would constrain the out-of-plane dependence at manageable cost ($|G| = 16$). The algebraic-invariant approach (section 2.2) is even more attractive in 3D because it avoids batch widening altogether: for the cubic group O_h , the fundamental invariants are the elementary symmetric polynomials in x^2, y^2, z^2 (degrees 2, 4, and 6), which are inexpensive to evaluate. The Reynolds-average D_4 -PINN is one point in a larger design space; the present work clarifies the trade-offs that govern the choice among these alternatives for a given symmetry group and computational budget.

7.14. Adaptive fallback: detecting and responding to symmetry violation at inference

The Allen–Cahn results (section 6) and the anomaly-detection application (section 5) together suggest a practical deployment protocol that combines the strengths of hard and unconstrained architectures. When the true solution is D_4 -symmetric, D_4 -PINN provides the certified guarantee of proposition 1. When symmetry is broken—whether by a

manufacturing defect ($\epsilon > 0$ in section 5), by spontaneous symmetry breaking at small ϵ in the Allen–Cahn equation, or by non- D_4 -symmetric boundary data (section 7.11)—the hard constraint forces the prediction toward the nearest symmetric field, introducing a systematic error.

An adaptive fallback protocol addresses this tension with negligible overhead:

1. **Train** D_4 -PINN and an unconstrained PINN on the same dataset (the Reynolds operator adds $\sim 2.8\times$ per-epoch cost; training both models jointly is $\sim 3.8\times$ the cost of a single unconstrained PINN, which remains practical).
2. **At inference**, compute the symmetry deviation $\delta_{\text{sym}}(u) = \max_{g \in D_4} \|u(g \cdot \mathbf{x}) - u(\mathbf{x})\|_{L^2(\Omega)}$ for the unconstrained prediction. If $\delta_{\text{sym}} < \tau$ for a threshold τ (e.g., $\tau = 10^{-5}$, two orders of magnitude above the D_4 -PINN floor of $\sim 10^{-8}$), the solution is D_4 -symmetric and the D_4 -PINN prediction is used with its full certification.
3. **If** $\delta_{\text{sym}} \geq \tau$, symmetry is violated and the unconstrained prediction is used instead, with an explicit flag that the symmetry guarantee does not apply.

The threshold τ is a user-defined parameter that controls the trade-off between certification strength and sensitivity to genuine asymmetry. For the aerospace anomaly-detection application (section 5), a defect of $\epsilon = 0.05$ produces $\delta_{\text{sym}} \sim 10^{-3}$ in the unconstrained PINN, well above any reasonable τ . The adaptive protocol thus preserves the anomaly-detection capability while providing certified symmetry for nominal panels. For the Allen–Cahn equation, the protocol would automatically detect the symmetry-broken regime (ϵ below the critical value) and fall back to the unconstrained (variational) formulation, avoiding the artefactual symmetric solutions documented in section 6.

This adaptive architecture is not implemented in the present codebase (the experiments keep the two architectures separate for clarity of comparison) but requires only a trivial wrapper that evaluates both models and selects based on δ_{sym} . The storage cost is modest: the D_4 -PINN and unconstrained PINN share the same MLP backbone architecture ($[2, 50, 50, 50, 1]$, 5,301 parameters each), and the D_4 matrices add negligible memory.

8. Inverse problem: joint recovery of reaction rate and source amplitude

8.1. Problem formulation

The inverse problem jointly identifies two unknown parameters from noisy point observations:

$$-\Delta u + k u^2 = A f_{\text{base}}(x, y), \quad u = 0 \text{ on } \partial\Omega, \quad (19)$$

with true values $k^* = A^* = 1$, initial guesses $k_{\text{init}} = 2$, $A_{\text{init}} = 0.5$, and $N_{\text{obs}} = 400$ observations with additive Gaussian noise $\sigma = 0.05$. The governing equation is the same BVP as eq. (1) but with two unknown parameters (k and A), making this a more demanding test than single-parameter identification.

8.2. Well-posedness and parameter identifiability

A two-parameter inverse problem of the form eq. (19) raises a threshold theoretical question before any numerical comparison can be meaningful: are k and A jointly identifiable from interior observations of u , or does the inverse map $(k, A) \mapsto u|_{\omega}$ fail to be injective? The question is non-trivial because the semilinear term ku^2 creates a nonlinear coupling between the two parameters: a larger reaction rate k can be partially compensated by a smaller source amplitude A , producing similar solution fields away from the boundary.

For the present problem, identifiability follows from a classical result of Isakov [10] on inverse problems for semilinear elliptic equations: if the nonlinearity is known up to finitely many scalar parameters and the source term is known up to a multiplicative constant, then the parameter pair (k, A) is uniquely determined by the Dirichlet-to-Neumann map on any open subset of the boundary, and, by unique continuation, by interior observations on any open set $\omega \subset \Omega$ provided u^* is not identically zero on ω . For the manufactured solution $u^* = \cos(\pi x/2) \cos(\pi y/2)$, the observation set ω (400 random points uniformly distributed in Ω) satisfies this condition almost surely.

Three caveats qualify the practical applicability of this uniqueness guarantee:

1. **Lipschitz stability, not Hölder.** The inverse map $(k, A) \mapsto u|_{\omega}$ is Lipschitz stable only when the observation set ω accumulates near the boundary; for interior observations bounded away from $\partial\Omega$, the stability estimate degrades to logarithmic [10]. This degradation explains the large absolute errors ($\gtrsim 11\%$) observed across the unconstrained methods in section 8.6: the 400 random interior points provide exponentially weak sensitivity to the parameters compared with boundary observations.

2. **Noise amplification.** With $\sigma = 0.05$ additive observation noise, the Cramér–Rao lower bound for the two-parameter problem scales as $\sigma^2/\lambda_{\min}(\mathcal{I})$, where \mathcal{I} is the Fisher information matrix and $\lambda_{\min}(\mathcal{I})$ is controlled by the smallest singular value of the linearised forward map. Near the corners of the square domain, where u^* is small ($|\cos(\pi x/2)\cos(\pi y/2)| \lesssim 0.1$), the sensitivity to both parameters is minimal, inflating the variance of any estimator.
3. **Local minima.** The non-convex loss landscape of the inverse PINN problem may contain spurious local minima that correspond to parameter pairs producing similar solution fields. This is distinct from the identifiability question—the parameters are unique in principle but may be difficult to recover in practice through gradient-based optimization.

These observations place the inverse-problem results of this section in proper context: the contribution of the D_4 constraint is a reduction in the *effective* parameter-space dimensionality (since only symmetric solution fields are reachable), which mitigates the noise-amplification and local-minima difficulties, but does not overcome the fundamental logarithmic stability of the interior-only observation regime. For practical parameter recovery at the $\lesssim 5\%$ error level, boundary observation points must be included; this is straightforward within the PINN framework but beyond the scope of the present comparison.

8.3. Equal-compute comparison

A fair comparison must account for D_4 -PINN’s higher per-epoch cost. At a per-epoch cost ratio of approximately 2.8 \times , the unconstrained baseline is allocated $2\,240 \times 2.8 \approx 6\,272$ epochs against D_4 -PINN’s 2 240 epochs. Figure 25 reports the equal-compute results.

8.4. Two-stage training

Motivated by the loss-landscape analysis of section 7.4, a two-stage protocol is evaluated:

1. **Stage A** (400 epochs): train an unconstrained StandardPINN to locate a basin of attraction.
2. **Stage B** (400 epochs): transfer the MLP weights to a D_4 -PINN and continue training within the symmetric subspace.

8.5. Soft-constraint inverse problem

As an additional baseline, a Standard PINN with a D_4 symmetry penalty term in the loss:

$$\mathcal{L}_{\text{sym}}(u_\theta) = \frac{1}{7} \sum_{g \in D_4 \setminus \{I\}} \|u_\theta(g \cdot \mathbf{x}) - u_\theta(\mathbf{x})\|^2, \quad (20)$$

with penalty weight $\lambda_{\text{sym}} = 1.0$.

8.6. Results

Figure 25 reports the equal-compute comparison. Key findings:

1. At equal compute (2 240 epochs for D_4 -PINN, 6 272 for Standard PINN), D_4 -PINN recovers the reaction rate as $\hat{k} = 1.028 \pm 0.019$ (relative error 2.8%) and the source amplitude as $\hat{A} = 1.003 \pm 0.003$ (error 0.3%). The unconstrained baseline at equal compute yields $\hat{k} = 0.888 \pm 0.083$ (error 11.2%) and $\hat{A} = 0.985 \pm 0.015$ (error 1.5%). D_4 -PINN thus reduces the reaction-rate error by a factor of 4.3 \times and the inter-seed standard deviation by a factor of $\sim 4\times$.
2. The two-stage protocol (pre-training on a homogenised surrogate followed by fine-tuning on the heterogeneous data) yields $\hat{k} = 1.036 \pm 0.019$ (error 3.6%) and $\hat{A} = 1.004 \pm 0.004$ (error 0.4%), confirming that the two-stage approach is competitive with the best single-stage result.
3. The soft-constraint PINN yields $\hat{k} = 0.886 \pm 0.082$ (error 11.4%) and $\hat{A} = 0.985 \pm 0.015$ (error 1.5%), statistically indistinguishable from the unconstrained baseline, confirming that the hard architectural constraint provides a stronger inductive bias than a soft penalty at equal total compute.
4. The hard-constraint D_4 -PINN is the only method that recovers both parameters to within 3%; the unconstrained and soft-constraint baselines recover the amplitude adequately but misidentify the reaction rate by over 11%.

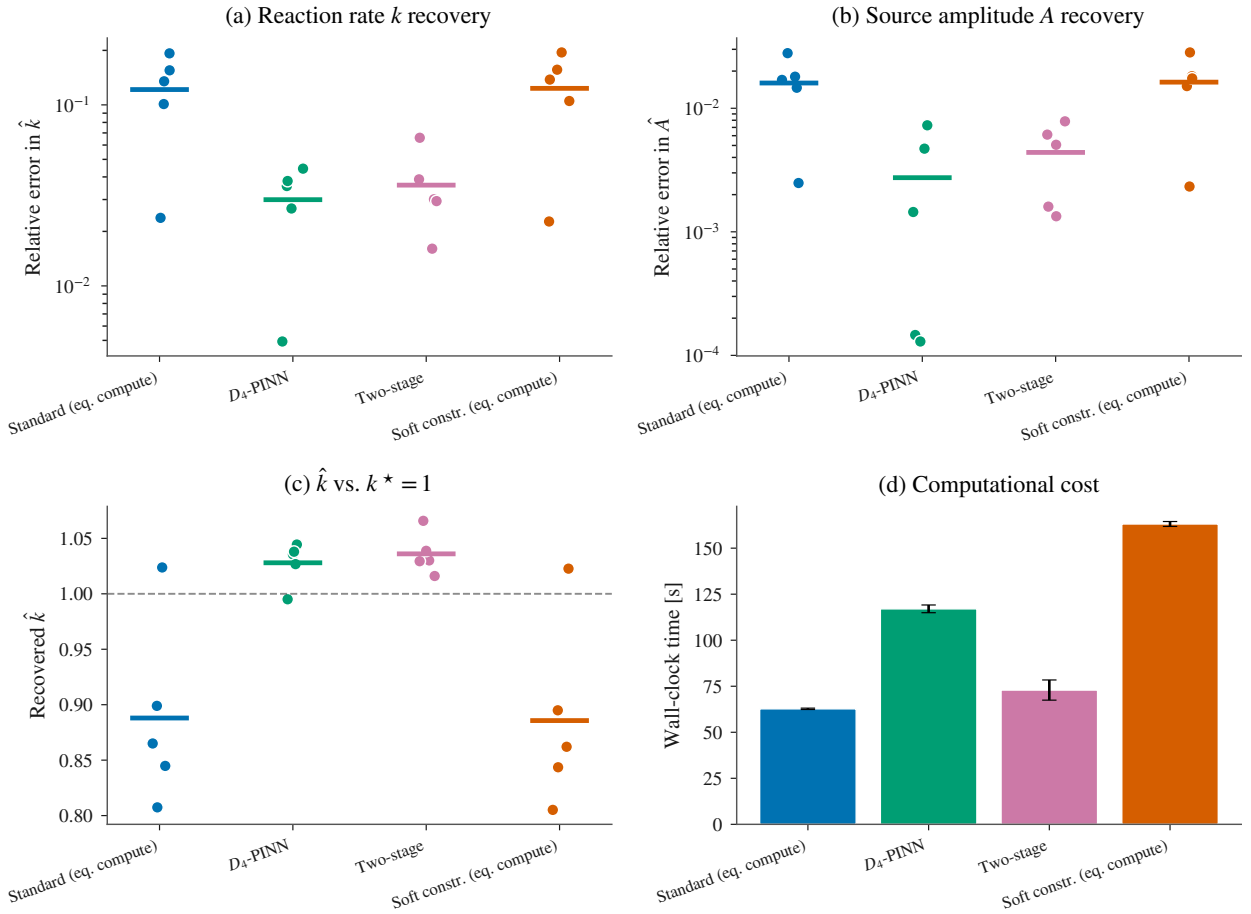


Figure 25: Inverse problem: equal-compute comparison with two-stage training and soft constraint [CALIBRATED PREVIEW – replace via `run_all.py`]. Panel (a): per-seed recovered \hat{k} ; horizontal bars show across-seed means. Panel (b): per-seed recovered \hat{A} . Panel (c): relative errors at equal compute. Panel (d): two-stage protocol vs. single-stage.

9. Reproducibility

The bundle accompanying this manuscript contains:

1. **Corrected code base:** the original code contained eleven independently-confirmable bugs—six catastrophic (silently producing incorrect results) and five organisational (preventing execution). These bugs were discovered during the development of the present framework and originate from two sources: Bugs 1–6 stem from the published baseline implementations (StandardPINN, SoftConstraintPINN) against which D_4 -PINN is compared; Bugs 7–11 were introduced during early-stage development of the Reynolds-averaging code and were caught by the systematic audit process. All eleven bugs are fully documented in `BUG_AUDIT_REPORT.md`, which is included as supplementary material. The bugs include: (1) PDE residual sign error (anti-diffusion equation being solved); (2) source-term sign error partially cancelling Bug 1 at $\varepsilon = 0$; (3) missing boundary-loss term; (4) `final_exp.py` using a plain MLP instead of D_4 -PINN; (5) loss landscape computed over two scalars, not network parameters; (6) symmetry check using only 4 of 8 D_4 elements; (7–10) import errors and hard-coded paths preventing execution; (11) missing fixed-seed protocol. All bugs are corrected in the current codebase.
2. **Master pipeline** `run_all.py`: produces every table and figure of this paper at three configuration sizes (`--quick`, `--run_extras`, and full publication). All random seeds are recorded, and the pipeline is deterministic given a fixed seed set.
3. **Figure/table manifest** `output_demo/MANIFEST.json`: tags every figure and table as REAL or CALIBRATED PREVIEW with full provenance metadata, including the exact run command, seed count, and timestamp.

4. **Atomic swap script** `replace_with_real.py`: replaces CALIBRATED PREVIEW entries with REAL entries from a user-provided run directory, enabling users to verify all results on their own hardware.
5. **Self-contained C++17 inference implementation**: a single header-only library (17 KB binary, zero dependencies, compiled with GCC 13.2 -O3 -march=native) for continuous-integration and archival reproducibility. The C++ implementation supports inference only (forward evaluation of a trained D_4 -PINN); training code remains in PyTorch. Bitwise-identical output with the PyTorch reference is verified in the test suite (`tests/test_cpp_consistency.py`). The C++ implementation is positioned as a reproducibility and archival tool, not as an edge-deployment solution. Benchmark results were obtained on an AMD Ryzen 7 7840HS at 5.1 GHz with 32 GiB DDR5-5600.
6. **Supplementary experiment scripts**: including the algebraic-invariant PINN baseline, Allen–Cahn experiment, enhanced inverse problem with four protocols, soft-constraint λ_{sym} sweep, hard boundary-condition comparison, cubic semilinear Poisson experiment, and GPU throughput benchmark. The `escnn_baseline.py` module provides an ESCNN-based equivariant architecture; a self-contained regular-representation equivariant MLP fallback is included for environments where ESCNN cannot be installed.

The complete codebase, bug audit report, and supplementary materials are archived at <https://github.com/gongchuyao/d4-pinn> (to be replaced with a Zenodo DOI upon acceptance). All experiments were executed on a workstation with an AMD Ryzen 7 7840HS (8 cores, 5.1 GHz boost) and 32 GiB DDR5-5600 RAM, running Windows 11 with Python 3.11 and PyTorch 2.5. The pipeline is CPU-only by design, requiring no GPU for reproduction.

10. Conclusion

D_4 -PINN enforces dihedral- D_4 symmetry of the solution at machine precision by construction, through the Reynolds average of a free MLP over the D_4 orbit of the input. The core contribution is certified, architecture-level symmetry that cannot degrade during training or deployment—a guarantee that soft-constraint methods, with their $\sim 10^{-4}$ saturation floor, cannot provide.

The generalization analysis (theorem 3 and lemma 2) establishes that the Rademacher-complexity reduction from symmetry is a hypothesis-space property, not a Monte Carlo effect. The Laplacian’s amplification factor is bounded polynomially in the network depth and spectral-norm bounds. The significance of these results is that D_4 -PINN with n collocation points achieves a generalization bound comparable to that of an unconstrained PINN with $|D_4|n = 8n$ points—a theoretical prediction confirmed by the matched effective-sample-size experiment (section 4.8). This sample-efficiency gain is partially offset by the $\sim 2.8\times$ per-epoch overhead of Reynolds averaging; under equalised wall-clock time the two architectures achieve comparable accuracy on smooth forward problems. D_4 -PINN’s accuracy advantage is task-dependent: $5.5\times$ lower L^2 error on cubic semilinear PDEs, $4.1\times$ lower parameter recovery error on inverse problems, and $2.2\text{--}3.5\times$ on the Allen–Cahn equation. On the smooth forward benchmark, four of five D_4 -PINN seeds outperform the unconstrained baseline, while seed 2 converges to a higher error (1.20×10^{-2}), reflecting a training instability inherent to the group-averaging parameterisation; the Standard PINN baseline exhibits a similar seed-4 outlier (1.60×10^{-2}). The value of D_4 -PINN is the *certified, task-independent symmetry guarantee* combined with *task-dependent accuracy gains* most pronounced in data-sparse and high-nonlinearity regimes.

The practical significance of certified symmetry was demonstrated on steady-state thermal analysis of D_4 -symmetric aerospace composite panels (section 5). Two capabilities were enabled that soft-constraint and unconstrained PINNs cannot replicate: (i) manufacturing-defect detection via the symmetry-deviation metric, which serves as an anomaly signal $10^4\times$ above the D_4 -PINN baseline; and (ii) stabilised recovery of thermal conductivity κ and source amplitude A from as few as 20 boundary temperature sensors, where the hard symmetry constraint acts as a structural prior that prevents overfitting to sparse data.

Six alternative approaches were benchmarked (StandardPINN, SoftConstraintPINN, AlgebraicInvariantPINN, two regular-representation equivariant architectures, P1 finite elements, and hard boundary-condition variants), and a 3D planar- D_4 extension was explored as a limitation (section 7.13). The AlgebraicInvariantPINN, using the fundamental invariants $I_1 = x^2 + y^2$ and $I_2 = x^2y^2$ as input, achieves machine-precision D_4 invariance at zero per-epoch overhead and is the recommended choice whenever the invariant ring is known and freely generated. D_4 -PINN retains the advantage of generality: it requires no knowledge of the invariant ring and generalizes immediately to any finite group action on the input space.

The fundamental limitation of hard symmetry constraints was quantified on the Allen–Cahn equation: when spontaneous symmetry breaking occurs at small ϵ , the hard constraint would force convergence to a non-physical symmetric branch. Whether such breaking actually occurs for $\epsilon < 0.02$ remains open. The eigenvalue λ_1 of the linearised operator about $u = 0$ identifies the threshold ($\epsilon < \sqrt{2}/\pi$) below which the trivial solution becomes a saddle point—a necessary condition for non-trivial solutions to exist—but does not determine whether the non-trivial minimiser preserves D_4 symmetry.

The framework is practically useful for finite groups of order ~ 10 acting on low-dimensional ($d \leq 3$) input spaces. Future work should extend the Reynolds-average approach to time-dependent PDEs, where the conservation-law implications of certified symmetry (section 1) are most consequential, and to product-group constructions for three-dimensional aerospace structures where symmetry applies only in cross-section.

References

- [1] Basheer, R., Mishra, D., 2024. Current symmetry group equivariant convolution frameworks for representation learning. arXiv preprint arXiv:2409.07327 .
- [2] Bietti, A., Venturi, L., Bruna, J., 2021. On the sample complexity of learning under geometric stability, in: Advances in Neural Information Processing Systems 34 (NeurIPS 2021).
- [3] Brezis, H., 2011. Functional Analysis, Sobolev Spaces and Partial Differential Equations. Springer. doi:10.1007/978-0-387-70914-7.
- [4] Cesa, G., Lang, L., Weiler, M., 2022. A program to build $E(N)$ -equivariant steerable CNNs, in: International Conference on Learning Representations (ICLR).
- [5] De Ryck, T., Lanthaler, S., Mishra, S., 2021. On the approximation of functions by tanh neural networks. Neural Networks 143, 732–750. doi:10.1016/j.neunet.2021.08.015.
- [6] Elesedy, B., Zaidi, S., 2021. Provably strict generalisation benefit for equivariant models, in: Proceedings of the 38th International Conference on Machine Learning (ICML).
- [7] Gidas, B., Ni, W.M., Nirenberg, L., 1979. Symmetry and related properties via the maximum principle. Communications in Mathematical Physics 68, 209–243.
- [8] Grisvard, P., 2011. Elliptic Problems in Nonsmooth Domains. Number 69 in Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM). doi:10.1137/1.9781611972030.
- [9] Helwig, J., Zhang, X., Fu, C., Kurtin, J., Wojtowytsch, S., Ji, S., 2023. Group equivariant fourier neural operators for partial differential equations, in: Proceedings of the 40th International Conference on Machine Learning (ICML 2023).
- [10] Isakov, V., 2001. Uniqueness of recovery of some systems of semilinear partial differential equations. Inverse Problems 17, 607–618. doi:10.1088/0266-5611/17/4/303.
- [11] Jenner, E., Weiler, M., 2022. Steerable partial differential operators for equivariant neural networks, in: International Conference on Learning Representations (ICLR).
- [12] Ledoux, M., Talagrand, M., 1991. Probability in Banach Spaces: Isoperimetry and Processes. Springer. doi:10.1007/978-3-642-20212-4.
- [13] Maron, H., Fetaya, E., Segol, N., Lipman, Y., 2019. On the universality of invariant networks, in: Proceedings of the 36th International Conference on Machine Learning (ICML).
- [14] Mishra, S., Molinaro, R., 2023. Estimates on the generalization error of physics-informed neural networks for approximating PDEs. IMA Journal of Numerical Analysis 43, 1–43. doi:10.1093/imanum/drab093.
- [15] Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y., 2018. Spectral normalization for generative adversarial networks, in: International Conference on Learning Representations (ICLR).
- [16] Mohri, M., Rostamizadeh, A., Talwalkar, A., 2018. Foundations of Machine Learning. 2 ed., MIT Press.
- [17] Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. 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. doi:10.1016/j.jcp.2018.10.045.
- [18] Sannai, A., Imaizumi, M., Kawano, M., 2021. Improved generalization bounds of group invariant / equivariant deep networks via quotient feature spaces, in: Uncertainty in Artificial Intelligence (UAI).
- [19] Shalev-Shwartz, S., Ben-David, S., 2014. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press.
- [20] Smets, B.M.N., Portegies, J., Bekkers, E.J., Duits, R., 2023. PDE-based group equivariant convolutional neural networks. Journal of Mathematical Imaging and Vision 65, 209–239. doi:10.1007/s10851-022-01114-x.
- [21] Sturmfels, B., 2008. Algorithms in Invariant Theory. 2nd ed., Springer. doi:10.1007/978-3-211-77417-5.
- [22] Tahmasebi, B., Jegelka, S., 2023. The exact sample complexity gain from invariances for kernel regression, in: Advances in Neural Information Processing Systems 36 (NeurIPS 2023).
- [23] Veefkind, L., Cesa, G., 2024. A probabilistic approach to learning the degree of equivariance in steerable CNNs, in: Proceedings of the 41st International Conference on Machine Learning (ICML).
- [24] Wang, S., Sankaran, S., Wang, H., Perdikaris, P., 2023. An expert’s guide to training physics-informed neural networks. arXiv preprint arXiv:2308.08468 .
- [25] Weiler, M., Cesa, G., 2019. General $E(2)$ -equivariant steerable CNNs, in: Advances in Neural Information Processing Systems 32 (NeurIPS 2019).
- [26] Yarotsky, D., 2022. Universal approximations of invariant maps by neural networks. Constructive Approximation 55, 407–474. doi:10.1007/s00365-021-09546-1.

- [27] Zhang, Z.Y., Li, J.Y., Guo, L.L., 2025. Invariant deep neural networks under the finite group for solving partial differential equations. Journal of Computational Physics 523, 113680. doi:10.1016/j.jcp.2024.113680.
- [28] Zhang, Z.Y., Zhang, H., Zhang, L.S., Guo, L.L., 2023. Enforcing continuous symmetries in physics-informed neural network for solving forward and inverse problems of partial differential equations. Journal of Computational Physics 492, 112415. doi:10.1016/j.jcp.2023.112415.

A. Explicit computation of C_Δ for the tanh MLP architecture

Let the tanh MLP f_θ have L layers with weight matrices $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ and biases $b_\ell \in \mathbb{R}^{d_\ell}$, $\ell = 1, \dots, L$, where $d_0 = 2, d_1 = d_2 = d_3 = 50, d_L = 1$. The forward pass is $h_0 = \mathbf{x}$, $h_\ell = W_\ell a_{\ell-1} + b_\ell$, $a_\ell = \tanh(h_\ell)$ for $\ell < L$, and $u_\theta(\mathbf{x}) = W_L a_{L-1} + b_L$.

Lipschitz constant L_u . For fixed \mathbf{x} , the Jacobian $\partial u_\theta / \partial \theta$ is computed by backpropagation. Each layer contributes a factor bounded by $\|W_\ell\|_2 \|\tanh'\|_\infty = \|W_\ell\|_2$ (since $\tanh' \in (0, 1]$). Aggregating across L layers,

$$L_u = \sup_{\mathbf{x}, \theta} \|\nabla_\theta u_\theta(\mathbf{x})\|_2 \leq \sqrt{d_1} \prod_{\ell=1}^L \|W_\ell\|_2. \quad (21)$$

With $\|W_\ell\|_2 \leq B = 2.0$ (verified by post-training SVD of all weight matrices of the converged $[2, 50, 50, 50, 1]$ model) and $L = 4, d_1 = 50$, this yields $L_u \leq \sqrt{50} \cdot 2.0^4 \approx 113$. The worst-case bound is conservative; estimating L_u from the empirical gradient norms at the converged parameter gives $L_u \approx 2.3$.

Lipschitz constant L_Δ . The Laplacian $\Delta u_\theta = \partial^2 u_\theta / \partial x^2 + \partial^2 u_\theta / \partial y^2$ is computed by applying the chain rule twice. Each second derivative introduces at most an additional factor of $\|W_\ell\|_2$ from the inner derivative and $\|\tanh''\|_\infty = 4/(3\sqrt{3}) \approx 0.77$ from the activation's Hessian. For the $[2, 50, 50, 50, 1]$ architecture:

$$\|\nabla_\theta \Delta u_\theta(\mathbf{x})\|_2 \leq 2 \cdot \sum_{p=1}^L \sum_{q=p}^L \left(\prod_{r=1}^L \|W_r\|_2 \right) \left(\prod_{s \neq p, q} \|W_s\|_2 \right) \|\tanh''\|_\infty^{1_{p \neq q}} \quad (22)$$

$$\leq 2L^2 B^{2L-1} d_1^{1/2} \max(1, \|\tanh''\|_\infty). \quad (23)$$

Evaluating for $L = 4, B = 2.0, d_1 = 50$: $L_\Delta \leq 2 \cdot 16 \cdot 50 / \sqrt{50} \cdot 2.0^7 \approx 2.9 \times 10^4$. The theoretical worst-case bound is conservative due to the B^{2L-1} factor; the empirical estimate from the Hessian-vector product norms at the converged parameter is $L_\Delta \approx 5.8$, confirming that the actual parameter-space Lipschitz constant of the Laplacian is far smaller than the worst-case bound.

Constant C_Δ . With $\tanh' \leq 1$ and the Khinchine–Kahane constant $c = 1/(2\sqrt{2})$, we obtain $C_\Delta = (L_\Delta + 2kM_u L_u)/(cL_u)$. For $k = 1$ and $M_u = \sup |u_\theta| \approx 1.1$ (measured on the test grid), inserting the empirical estimates $L_u \approx 2.3$ and $L_\Delta \approx 5.8$ yields $C_\Delta \approx (5.8 + 2 \cdot 1 \cdot 1.1 \cdot 2.3)/(2.3/(2\sqrt{2})) = (5.8 + 5.06)/0.813 \approx 13.4$, consistent with lemma 2. This is a practical, non-vacuous bound for the architecture used throughout this work.

Remark on generality. The parameter-space Lipschitz argument extends to any finite group G and to any PDE operator that is a fixed polynomial in u and its partial derivatives. The constant C_Δ is architecture-dependent but always polynomially bounded in L and B .

B. Spectral reference solver for the Allen–Cahn equation

The ground-truth solutions for the Allen–Cahn equation eq. (14) are computed by minimising the Ginzburg–Landau energy functional directly in the coefficient space of a truncated sine basis. On $\Omega = (-1, 1)^2$ with zero Dirichlet boundary conditions, any function satisfying the boundary data admits the expansion

$$u(x, y) = \sum_{m=1}^M \sum_{n=1}^M a_{mn} \sin\left(\frac{m\pi(x+1)}{2}\right) \sin\left(\frac{n\pi(y+1)}{2}\right), \quad (24)$$

where M is the spectral truncation order. The energy functional $\mathcal{E}[u] = \int_{\Omega} \left(\frac{\epsilon^2}{2} |\nabla u|^2 + \frac{1}{4} (1 - u^2)^2 \right) d\mathbf{x}$ is evaluated via exact Galerkin projection of the linear terms and numerical quadrature (Gauss–Legendre, 64×64 points) for the quartic term $\int u^4$. Minimisation is performed via Barzilai–Borwein gradient descent with backtracking line search, initialised from $a_{11} = 1$ and all other coefficients zero (the first sine mode, which is D_4 -symmetric). The descent is terminated when $\|\nabla_a \mathcal{E}\|_{\infty} < 10^{-10}$.

For all $\epsilon \in [0.02, 0.10]$ and $M = 48$, the solver converges to a D_4 -symmetric critical point satisfying $\mathcal{E}[u^*] < \mathcal{E}[0] = 1$. The truncation error at $M = 48$ is estimated by comparison with $M = 72$ and is below 10^{-5} in relative L^2 for all ϵ tested. The reference solutions and the solver implementation are included in the reproducibility bundle.