DECOMPOSE-AND-SUPERPOSE: A SUPERPOSITION-INFORMED NEURAL NETWORK FOR STEADY-STATE TEMPERATURE FIELD PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The finite element method (FEM) is the standard tool for steady-state temperature field simulation, but every change of a design parameter – thermal conductivity, convective heat transfer coefficient, or domain size – requires a new mesh and a new matrix solve, which is prohibitive in parametric thermal design and in real-time thermal management. This paper proposes a superposition-informed neural network (SINN) for fast steady-state temperature field prediction. Starting from the linear operator structure of the steady-state heat conduction equation with Robin boundary conditions, the control parameters are split into operator parameters, which alter the operator itself (thermal conductivity, convective heat transfer coefficient, geometric dimensions), and load parameters, which act linearly on the solution once the operator is fixed (heat source strength, ambient reference temperature, imposed boundary values). The temperature field is accordingly decomposed into one source-driven base subfield and several boundary-response subfields; neural networks learn only the non-superposable mapping from operator parameters to these subfields, and the field under an arbitrary load is reconstructed analytically from them without further network evaluation. On a two-dimensional steady-state heat conduction dataset covering 9 boundary condition combinations and 52 rectangular geometries, the method attains an RMSE of 1.94 mK and is 229–980 times faster than COMSOL FEM at inference. Because the field under an arbitrary load is assembled from the learned subfields rather than re-predicted, load conditions absent from the training data are reconstructed at the same accuracy, without retraining and without any further network evaluation. Because the operator/load split relies only on the linear structure of the governing equation, the framework extends to other steady-state linear diffusion problems; the conditions for this transfer are analyzed in Section 6. The source code is provided as supplementary material.

1 INTRODUCTION

Fast and accurate prediction of steady-state temperature fields is a pervasive need in thermal engineering, from the design and optimization of heat sinks (Ahmed et al., 2018) and the thermal management of power electronic devices and battery packs (Qian et al., 2018; Wang et al., 2016) to the control of temperature and residual stress fields in manufacturing (Li et al., 2018). Practitioners must repeatedly solve the same conduction problem over a wide parameter space of geometry, material properties, and boundary conditions, while digital twins and real-time control demand millisecond-level solution speeds. The steady-state heat conduction equation belongs to the class of linear diffusion-type partial differential equations (PDEs) (Karniadakis et al., 2021): the field is controlled by a diffusion coefficient, a source term, an interfacial transport coefficient, and the geometry, and the same structure underlies mass diffusion, electrostatic fields, porous media flow, and neutron diffusion. We address steady-state heat conduction here and defer the transfer of the method to the other members of this class to Section 6.

The standard numerical tool is the finite element method (FEM). It is accurate for a single parameter combination, but any change of a parameter (geometric dimensions W, H, thermal conductivity k, convective heat transfer coefficient h) forces a new mesh and a new matrix solve, so previously

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

computed results cannot be reused. This cost makes FEM ill-suited to parameter sweeps and real-time control (Permann et al., 2020).

Two families of alternatives exist. Lumped-parameter and reduced-order models collapse the spatial PDE into an equivalent network or low-dimensional projection: Li et al. (2021) reduced a lithium-ion battery to isothermal nodes and used a thermal-resistance–thermal-capacitance network to estimate hot-spot temperatures. Such models replace the field by a low-dimensional approximation and cannot capture complete spatial information, a fundamental limitation when temperature uniformity, local hot spots, or stress concentration must be evaluated (He et al., 2021). The second family replaces FEM with a learned "parameter-to-field" mapping (Cai et al., 2021): physics-informed neural networks embed the governing equations in the loss (Raissi et al., 2019; Wang et al., 2021a), and Zhao et al. (2023) applied this idea in a physics-informed CNN for two-dimensional steady-state fields without FEM labels. These methods share three limitations: poor generalization outside the training domain (Cuomo et al., 2022); training samples that must densely cover the parameter space, growing exponentially in the parameter dimension (Calzolari & Liu, 2021); and no use of physical structure, since a black-box network cannot distinguish effects that superpose analytically from those that must be learned (Willard et al., 2022).

We propose a superposition-informed neural network (SINN). The control parameters fall into operator parameters (thermal conductivity, convective coefficient, geometric dimensions), which alter the operator itself, and load parameters (heat source strength, ambient reference temperature, imposed boundary values), which act linearly once the operator is fixed. For fixed operator parameters the dependence on the loads is linear and can be assembled analytically from a few basis solutions, whereas changes in operator parameters produce a non-superposable nonlinear response (Lu et al., 2021). The method therefore decomposes the field into a superposable load response, handled analytically, and a non-superposable operator response, learned by a network (Wang et al., 2021b), reducing input dimensionality, removing the need to re-invoke the network for new loads, and shrinking the space each network must learn (Raissi et al., 2019).

We validate the method on the two-dimensional steady-state heat conduction equation (i.e., the Poisson equation with Robin boundary conditions), whose parameters and boundary conditions are representative of the linear diffusion class; the operator/load classification is analyzed in Section 2. Experiments cover 9 boundary condition combinations, 52 rectangular geometries, and approximately 60,000 sets of operator parameters. The method reaches millisecond-level inference at FEM-comparable accuracy, and load conditions absent from the training data are reconstructed at the same accuracy by superposing already-learned subfields. Section 6 examines the conditions under which the framework transfers to other steady-state linear diffusion problems.

2 The Superposition Principle in Steady-State Heat Conduction

2.1 Problem Formulation and Linear Operator Structure

Consider a bounded two-dimensional domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$. The governing equation of the steady-state heat conduction problem is

$$\nabla \cdot (k\nabla T) + q = 0 \quad \text{in } \Omega \tag{1}$$

and the Robin (convective) boundary condition is imposed on the boundary

$$-k\frac{\partial T}{\partial n} = h(T - T_\infty) \quad \text{on } \partial\Omega \tag{2}$$

where T is the temperature, k the thermal conductivity, q the volumetric heat source density, h the convective heat transfer coefficient, and T_∞ the ambient reference temperature; Dirichlet and Neumann conditions are limiting cases of Robin. The same structure, with different physical meanings of the field variable and coefficients, underlies the other steady-state diffusion problems listed in Section 6; the present section is written in thermal notation.

Define the domain operator $\mathcal{L}[T] \equiv -\nabla\cdot(k\nabla T)$ and the boundary operator $\mathcal{B}[T] \equiv -k\partial T/\partial n + hT$; then Eqs. equation 1–equation 2 become the composite operator equation

$$\mathcal{A}[T] = \mathcal{S}, \qquad \mathcal{S} = (q,\, hT_\infty) \tag{3}$$

where $\mathcal{A} = (\mathcal{L}, \mathcal{B})$.

Theorem 1 (Superposition Principle): with k and h fixed, the operator $\mathcal{A}$ is linear. For scalars λ_1, λ_2, if T_1 and T_2 correspond to loads $\mathcal{S}_1$ and $\mathcal{S}_2$, then $\lambda_1 T_1 + \lambda_2 T_2$ is the solution for $\lambda_1 \mathcal{S}_1 + \lambda_2 \mathcal{S}_2$ (Evans, 2010; Brezis, 2010). The theorem uses only the linearity of $\mathcal{A}$, and therefore applies to any steady-state problem sharing the structure of equation 1–equation 2.

2.2 Linear Superposability of Load Parameters

Theorem 1 directly yields the linear dependence of the temperature field on two types of load parameters.

Heat source superposition: When k, h, and T_∞ are fixed, the dependence of the temperature field on the volumetric heat source q is affine-linear:

$$T[q_1 + q_2] = T[q_1] + T[q_2] - T[0] \tag{4}$$

where $T[0]$ is the background field with zero heat source, and the subtraction avoids double-counting the background field. The same affine relation holds for a uniform scaling of the total heating power, since scaling q by a factor is itself a load change.

Ambient reference temperature superposition: When k, h, and q are fixed, the dependence of the temperature field on the ambient reference temperature T_∞ is also linear:

$$T[\lambda T_{\infty,1} + (1-\lambda) T_{\infty,2}] = \lambda T[T_{\infty,1}] + (1-\lambda) T[T_{\infty,2}] \tag{5}$$

Arbitrary load variations therefore follow from a few basis solutions by analytical linear combination, without solving each load state separately.

2.3 Nonlinearity and Non-superposability of Operator Parameters

Changing an operator parameter, by contrast, breaks the linear invariance of $\mathcal{A}$ and destroys superposability. The convective coefficient h transforms the boundary operator from $\mathcal{B}_1[T] = -k\partial T/\partial n + h_1 T$ to the corresponding form with h_2, giving a hyperbolic inverse response $T \propto 1/h$. The conductivity k enters the domain operator $\mathcal{L}[T] = -\nabla \cdot (k \nabla T)$ and alters the governing differential equation itself. A change of geometric dimensions modifies the solution domain Ω, so that linear combinations of earlier solutions lose physical meaning and the problem must be solved independently.

The superposition principle therefore separates load parameters (volumetric heat source q, ambient reference temperature T_∞), which superpose linearly once the operator parameters are fixed, from operator parameters (thermal conductivity k, convective coefficient h, geometric dimensions), whose variation produces a nonlinear response. This is the basis of the decoupled SINN design: the network learns the mapping from operator parameters to the field, while load variations are handled analytically, reducing input dimensionality and enhancing generalization.

3 Simulation Model and Data Generation

3.1 Model Geometry and Parameterization Scheme

The simulation domain is $\Omega = [0, W] \times [0, H]$, discretized using a mapped quadrilateral mesh. Owing to constraints on training data scale and computational resources, only two adjacent sides are controllable active boundaries (boundary 1, bottom edge; boundary 2, right edge), each independently imposing a Dirichlet (D), Neumann (N), or Robin (R) condition; the remaining two sides (boundary 3, top edge; boundary 4, left edge) are held at the homogeneous Dirichlet value $T = 293.15$ K and serve as reference edges. The solid is silica glass, and a thermal conductivity

scaling factor c is applied so that $k = c \cdot k_0$. Its thermophysical properties and the scanning ranges of the operator parameters are given in Tables 3 and 4 in Appendix A.

Following Section 2, the operator parameters (scaling factor c, convective coefficient h, dimensions W, H) alter the problem structure and must be learned by the network, whereas the load parameters (volumetric heat source q, ambient reference temperature T_∞, fixed values on the active boundaries) act linearly when the operator is invariant and can be obtained by analytical combination of a few basis solutions.

The rectangular dimensions are deduplicated in two steps: first, the constraint $W \geq H$ is imposed to eliminate rotational symmetry duplicates; second, deduplication is performed by aspect ratio. The steady-state temperature field depends only on the normalized coordinates $(x/W, y/H)$ and the aspect ratio W/H; rectangles with the same reduced fraction W/H are reduced to a single representative, yielding 52 distinct rectangles.

3.2 Dataset Composition and Post-processing

By the superposition principle of Section 2-B, the field for a volumetric heat source with two independently controllable boundaries decomposes into three subfields:

$$T = T_{\text{source}} + \Delta T_{\text{reb1}} + \Delta T_{\text{reb2}} \tag{6}$$

where T_{source} is the base field with the source and both boundaries homogeneous, and $\Delta T_{\text{reb}i}$ is the temperature rise with no source, boundary i active, and the other boundary homogeneous. The boundary types of boundaries 1 and 2 must be consistent across the three subproblems for the superposition to be strictly valid.

Two datasets are therefore constructed: the BoundaryNet dataset (no heat source, for learning ΔT_{reb1} and ΔT_{reb2}) and the SourceNet dataset (with volumetric heat source, for learning T_{source}). The default boundary condition values are specified in Table 5 in Appendix A, and combinations are denoted bXbY, where bX and bY give the types of boundaries 1 and 2 (D/N/R correspond to b1/b2/b3); the state (active or homogeneous) is determined by the dataset.

The no-heat-source dataset scans 9 combos $\times$ 2 heating directions, giving $30,368 \times 2 = 60,736$ samples, with all 9 pairings between the active and reference boundaries covered within each combo (Table 6, Appendix A). The heat-source dataset uses a total heating power $Q = 10$ W, with the volumetric source density scaled with the rectangle volume so that this total power remains constant for every geometry; fields at other heating powers follow by linear scaling of the 10 W field, as implied by equation 4. Boundaries 1 and 2 are homogeneous with their types traversing D/N/R, and boundaries 3 and 4 are fixed at 293.15 K (Table 7, Appendix A). Its structure is isomorphic to a single direction of BoundaryNet: b1b1 scans only W, H (under homogeneous Dirichlet conditions the equation reduces to $\nabla^2 T = -Q/k$, the source density scaled by rectangle area so the total power stays constant), while the remaining 8 combos scan c and h by the same rules, giving 30,368 samples. For b1b1, c is held at its baseline value: all four edges are then at the same reference temperature, so the temperature rise satisfies $\nabla \cdot (k \nabla \Delta T) = -q$ with $\Delta T = 0$ on the entire boundary, and the dependence on c reduces to the exact amplitude factor $1/k$.

COMSOL Multiphysics outputs scattered nodal temperatures, which are interpolated onto a uniform 100×100 grid by cubic spline interpolation, with nearest-neighbor fallback for NaN values from near-boundary extrapolation. Each dataset is split into training and validation sets by stratified sampling by combo (or direction + combo) at 90% / 10% (Table 8, Appendix A).

4 SINN Network Architecture

4.1 Overall Design and Network Architecture

The two datasets of Section 3 train two networks of identical structure serving complementary tasks: **SourceNet**, which predicts the base field T_{source} under a volumetric heat source and homogeneous boundary conditions, and **BoundaryNet**, which predicts the temperature rise ΔT_{reb} with no heat

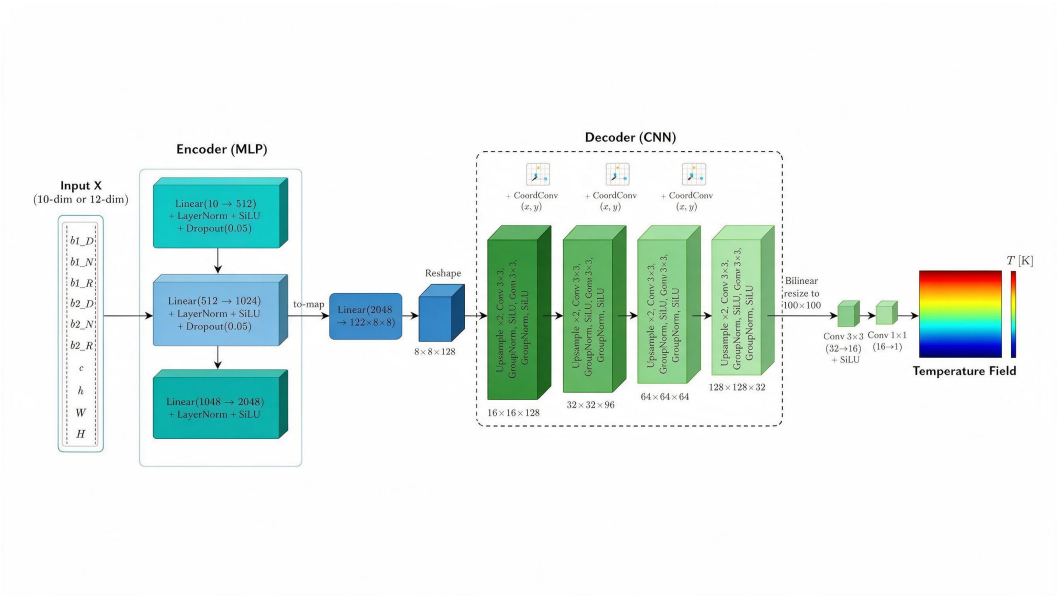

Figure 1: Schematic of the SINN network architecture (MLP encoder (input $\rightarrow$ 512 $\rightarrow$ 1024 $\rightarrow$ 2048) $\rightarrow$ To-Map layer (2048 $\rightarrow$ 128$\times$8$\times$8) $\rightarrow$ 4 CoordConv upsampling blocks $\rightarrow$ output head (32 $\rightarrow$ 16 $\rightarrow$ 1) $\rightarrow$ 100 $\times$ 100 temperature rise field).

source and single-boundary active heating. They are combined at inference through the superposition formula equation 6, so that

$$\text{SINN} = \text{SourceNet} + \text{BoundaryNet}_{\text{reb1}} + \text{BoundaryNet}_{\text{reb2}} \tag{7}$$

the complete field under arbitrary boundary conditions and volumetric heat sources is obtained without re-invoking the network for each load combination. Both networks adopt an Encoder-Decoder architecture (Fig. 1).

The design reflects the two stages of the task. The operator parameters and boundary types form a discrete/continuous hybrid condition that an MLP maps efficiently to a latent feature, while the field, a continuous spatial function with local correlations (gradients, boundary layers), is captured efficiently by a CNN through local connectivity and weight sharing. Concretely, the condition vector (10˜12 dimensions) is abstracted by the MLP encoder (input $\rightarrow$ 512 $\rightarrow$ 1024 $\rightarrow$ 2048), expanded by the To-Map layer (2048 $\rightarrow$ 128$\times$8$\times$8) into a low-resolution feature map, upsampled to 128$\times$128 by 4 CoordConv blocks (each concatenating the normalized coordinates $[x, y] \in [-1, 1]$, so the network perceives absolute position), and aligned by the output head (32 $\rightarrow$ 16 $\rightarrow$ 1) to the 100 $\times$ 100 field.

BoundaryNet and SourceNet differ slightly in input dimensionality (12 vs. 10) and parameter count, but both are on the order of 20 M (Table 9, Appendix B); they share training hyperparameters and code, requiring only a switch of the input dimensionality.

4.2 Input Features and Normalization

Both networks take the same kind of input: leading dimensions are one-hot encodings of the boundary types, and the last four are the continuous operator parameters (c, h, W, H). The one-hot part encodes the discrete problem structure; the continuous part carries the material, geometric, and heat-transfer parameters, whose nonlinear response the network learns. BoundaryNet additionally needs a 2-dimensional feature distinguishing the two heating directions (reb1 / reb2), giving 12 dimensions, whereas SourceNet encodes the homogeneous type of each of boundaries 1 and 2 in 3 dimensions, giving 10.

The input features of BoundaryNet and SourceNet are listed in Table 12 and Table 13, respectively, in Appendix B.

Both networks apply training-set Min-Max normalization to the continuous features (c, h, W, H) to map them to $[0, 1]$, placing them on the same scale as the one-hot encodings and avoiding optimization bias from differing feature magnitudes, and normalize the output temperature rise fields $\Delta T = T - 293.15$ by a global Z-score over the training set; both transformations, together with the Z-score statistics, are given in Appendix B. During inference, the physical temperature rise is recovered via the inverse normalization $\Delta T = y_{\text{pred}} \cdot \sigma_y + \mu_y$.

4.3 Loss Function

To simultaneously optimize overall fitting accuracy and local accuracy at boundaries and corners, this paper employs a multi-task loss function:

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + 0.05\mathcal{L}_{\text{Grad}} + 0.5\mathcal{L}_{\text{Boundary}} + 0.1\mathcal{L}_{\text{TopK}} \tag{8}$$

The definition and role of each loss term are given in Table 15 in Appendix B.

4.4 Training Configuration

The two networks share a common set of training hyperparameters, listed in Table 16 in Appendix B.

The training and validation sets are split using 90% / 10% stratified sampling, ensuring that all splits contain all 9 combos and 2 directions (for BoundaryNet).

5 Comparative Experiments and Accuracy Validation

This section validates SINN at three levels: network-level (whether the architecture choice is reasonable), method-level (whether decomposition-superposition is superior to a purely data-driven approach), and engineering metrics (end-to-end accuracy, inference efficiency, and recombination of learned basis responses). All experiments were conducted on the same hardware (NVIDIA GPU + COMSOL Multiphysics), and all comparison models adopted the training hyperparameters and data splits described in Section 4 to ensure fairness.

5.1 Comparison Objects and Evaluation Metrics

To evaluate SINN comprehensively, three categories of comparison objects are used (Table 20, Appendix E): the network-level comparison validates the CoordConv and CNN-decoder choices, the method-level comparison evaluates decomposition-superposition against end-to-end mapping without decomposition, and the FEM baseline provides the reference for reconstruction accuracy and inference efficiency.

Reported are three error measures – RMSE, maximum worst-point error $\max \|\text{err}\|$, and R^2 – plus parameter count and single-inference time.

(1) Root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2} \tag{9}$$

where N is the number of validation pixels (number of samples $\times 100 \times 100$) and y_i, $\widehat{y}_i$ are the ground-truth and predicted values at pixel i. RMSE is homologous to the training loss $\mathcal{L}_{\text{MSE}}$, and its quadratic penalty makes it sensitive to local instabilities at boundaries and corners.

(2) Maximum worst-point error $\max \|\text{err}\|$

$$\max\|\text{err}\| = \max_i |y_i - \widehat{y}_i| \tag{10}$$

the largest absolute error over all validation pixels.

Table 1: SINN vs. DirectNet in-domain accuracy

Metric	SINN	DirectNet	Relative advantage
RMSE (mK)	**1.944**	2.837	31% better
$max \parallel \text{err} \parallel$ (mK)	**210.510**	256.042	18% better
R^2	0.999999	0.999997	–

(3) Coefficient of determination R^2

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{11}$$

where $\bar{y} = \frac{1}{N}\sum_i y_i$ is the mean ground-truth field; values closer to 1 indicate a higher proportion of variance explained.

5.2 Network-Level Comparison: Architecture Validation

We ablate the two architectural choices – coordinate concatenation (CoordConv) and the CNN decoder – on both networks under identical configurations. CoordConv proves critical for suppressing the maximum error (removing it degrades SourceNet's $max \parallel \text{err} \parallel$ by 214%, and a pure MLP needs 2.6 times the parameters for a still-worse RMSE), and the CNN decoder markedly improves parameter efficiency. The full ablation, with the per-sample error distributions and single-sample error fields, is given in Appendix D.

5.3 Method-Level Comparison: SINN vs. Purely Data-Driven Approach

To validate the core value of the superposition decomposition, we compare SINN with a purely data-driven full-field network, DirectNet, which learns the end-to-end mapping $(\text{BC}, \text{heat source}, c, h, W, H) \rightarrow T(x, y)$ with no subproblem decomposition. Its backbone, loss function, and training hyperparameters are identical to those of the subnetworks; only the input layer is expanded to a 16-dimensional all-one-hot encoding (Table 17 in Appendix B). It was trained on the complete field dataset (30,368 samples) with the same split as SINN (27,328 / 3,040), 20.05 M parameters, and a training time of 2.91 h.

On the same 3,040 validation conditions SINN outperforms DirectNet on every metric (Table 1): an RMSE of 1.944 mK against 2.837 mK (31% better), a $max \parallel \text{err} \parallel$ of 210.5 mK against 256.0 mK, and a comparable R^2. Decomposing the problem leaves each subproblem with a simpler solution space, whereas DirectNet must fit all coupling effects of the volumetric heat source, dual-boundary heating, and operator parameters within a single network. The single-sample error fields of Fig. 6 (Appendix E) show the same pattern: SINN attains an RMSE of 1.841 mK and an MAE of 1.135 mK, 24% and 34% below DirectNet, with a more compact spatial error distribution, while both models reach their worst error (approximately 41 mK) near the corners of the active Dirichlet boundary – a common bottleneck caused by the physical singularity rather than by network structure.

5.3.1 Recombination of Learned Basis Responses on Unseen Load Combinations

To examine whether SINN can recombine basis responses that it has already learned, boundary 2 is switched from the active state to the homogeneous state while boundary 1 remains active and the volumetric heat source is retained, giving 3,040 load conditions. This combination of boundary states occurs nowhere in the training data, yet for each condition the true field is exactly the superposition of two responses on which the subnetworks have already been trained: the source-driven base field for that boundary combination and the boundary-1 response. SINN therefore reconstructs the field by adding these two learned components, with no retraining and without introducing any new network capacity.

SINN reconstructs these conditions with an RMSE of 1.46 mK – essentially its in-domain accuracy – and a max|err| of 213.3 mK, likewise of the same order as in domain (Table 2); since neither

Table 2: SINN reconstruction and DirectNet output on the unseen load-state test set

Model	RMSE (mK)	max|err| (mK)	R^2
SINN	**1.46**	**213.3**	**0.999998**
DirectNet	1,396.07	10,062	-0.593

subnetwork is re-fitted or re-evaluated, the result comes purely from recombining already-learned components. DirectNet yields an RMSE of 1,396 mK, an MAE of 929 mK, a max|err| of 10,062 mK, and an R^2 of -0.593. This should not be read as a measure of its generalization ability: the boundary-2 state is a one-hot input whose second entry is identically zero throughout its training set, so the corresponding first-layer weights receive no gradient and remain at their initialization; the figure reflects the absence of any training signal for that dimension, not a property of end-to-end learning in general. What the design establishes is the complementary result in the first row of Table 2: basis responses already learned by the subnetworks can be recombined into a field the method was never trained on, at millikelvin accuracy. Figure 7 (Appendix E) shows a representative sample.

5.4 Inference Efficiency Comparison

We compare the inference time of SINN and FEM. The FEM baseline is a single COMSOL solution of the complete field, including mesh generation, steady-state solution, and result interpolation; SINN involves three forward inferences plus analytical superposition, timed after warm-up as the median of multiple runs (Table 19, Appendix E).

SINN achieves a 229$\times$ speedup over FEM for single-sample inference on GPU and a nearly 1000$\times$ speedup for batch inference (parameter-sweep scenarios); even on CPU, a 41$\times$ speedup is attained. Moreover, because arbitrary load variations (heat source intensity, active boundary values) under the same operator parameters (c, h, W, H) require no re-inference, whereas FEM must re-solve for each load state, the practical speedup in multi-load exploratory design is further amplified.

6 Discussion

6.1 The Contribution Lies in the Decomposition, Not in the Network

The method has been validated for one instance of the problem: a two-dimensional rectangular domain, a single solid, and two adjacent controllable edges with independent Dirichlet, Neumann, or Robin conditions, covering 9 combinations and 52 geometries. Within that scope, the contribution is the decomposition rather than the network. The subnetworks are ordinary regression networks, each mapping a low-dimensional vector of operator parameters to a two-dimensional field – a design choice, not the contribution. The contribution is the separation of the operator and load parameters and the analytical recombination of the resulting subfields, a construction external to the network: any backbone with the same input–output signature, among them a Fourier neural operator or a DeepONet, can replace the MLP–CoordConv decoder without altering the parametric sweep, the datasets, the superposition step, or the reconstruction of an arbitrary load.

The ablation of Section 5-B is consistent with this: the three architectures tested on SourceNet differ only slightly in accuracy (RMSE 0.423, 0.522, and 0.508 mK), and the most accurate is not the one with the most parameters. A systematic comparison against operator-learning backbones is left as a limitation. What changes accuracy decisively is whether the field is predicted end to end or assembled from subfields: at a fixed backbone, SINN improves on DirectNet on every in-domain metric (Section 5-C), so the reported accuracy should be read as a property of the decomposition rather than of the network that implements it.

6.2 Transfer to Other Steady-State Linear Diffusion Problems

The decomposition relies on no thermal-specific property of Eqs. equation 1–equation 2, only on the governing equation being linear in the field variable and on its parameters separating into those

entering the operator and those entering the load. It therefore applies to any steady-state problem satisfying four conditions: linearity in the field variable with field-independent coefficients; form-preserving differential and boundary operators once the operator parameters are fixed; linear entry of the load terms; and steadiness, since a transient problem would add time as a coordinate. The conditions, the correspondence with other physical systems (Table 18), and the cases in which the framework does not apply – nonlinear coefficients, coupled multiphysics, or geometry that changes between operating conditions – are detailed in Appendix C.

The two parameter classes are distinguished by where a parameter appears in the composite operator equation, not by its physical name. In equation 2, h multiplies T and so changes $\mathcal{B}$, making it an operator parameter that must be a network input, whereas T_∞ enters as a constant right-hand-side term and is a load parameter, superposable like the heat source; Appendix C gives the criterion and its borderline cases.

7 CONCLUSION

This paper proposes a superposition-informed neural network (SINN) for fast steady-state temperature field prediction. The field is decomposed into linearly combinable subfields; the networks fit only the non-superposable mapping from operator parameters to these subfields, and the field under arbitrary loads is reconstructed by analytical superposition. Taking the two-dimensional steady-state heat conduction problem (i.e., the Poisson equation with Robin boundary conditions) as the target problem, the contributions are:

1. **Theoretical analysis**. The control parameters are classified into operator parameters (thermal conductivity k, convective coefficient h, geometric dimensions W, H) and load parameters (volumetric heat source q, ambient reference temperature T_∞, values imposed on the active boundaries), and the nonlinearity of the response to operator-parameter changes together with the linear superposability of load parameters is established.
2. **Network design**. Both subnetworks use an Encoder-Decoder architecture of an MLP encoder plus a CoordConv CNN decoder: the MLP maps low-dimensional conditions to a latent space, and the CoordConv CNN addresses spatial perception at boundaries and corners through explicit coordinate injection.

SINN outperforms the end-to-end DirectNet on every in-domain evaluation metric, with an RMSE 31% lower (1.94 mK vs. 2.84 mK). Beyond the training distribution, already-learned responses can be recombined analytically: for load conditions whose boundary-state combination does not occur in the training data, SINN reconstructs the field with an RMSE of 1.46 mK, the same level as in domain, without retraining or any new network capacity. Inference is substantially faster than FEM – 229$\times$ for single-sample GPU inference, nearly 1000$\times$ for a batch of 256, and 41$\times$ on CPU – and because load variations under the same operator parameters follow from analytical combination of the basis fields, no re-inference is required, whereas FEM must re-solve for each load state. The following limitations warrant further investigation:

1. Only two adjacent rectangle sides with three boundary condition types are considered; more complex geometries and boundary conditions would require expanding the dataset and modifying the network structure.
2. Only steady-state problems are addressed; transient heat conduction would require sequential networks or operator learning methods.
3. The method is validated solely on steady-state heat conduction; Appendix C analyzes the conditions for transfer to other steady-state linear diffusion problems, but none is solved numerically here.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and LaTeX formatting of the manuscript. We have not used generative AI tools for the research ideas, the experimental design, the simulation code, or the generation of any numerical results reported in this paper, and these tasks are not applicable to the use of such tools in this work. We have reviewed all AI-assisted work:

every suggested edit was checked against the underlying simulation outputs and the cited literature, and all reported quantities were verified against the simulation results by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This work is a computational study of steady-state heat conduction. It relies exclusively on numerically generated data obtained from finite-element simulation of an idealized solid, and involves no human subjects, no animal subjects, and no personally identifiable information; no datasets were collected from human participants or from online sources. We foresee no ethical concerns, conflicts of interest, or dual-use risks arising from this work.

Reproducibility statement

The simulation model, the parameter ranges, the dataset composition, the network architectures, and the training configuration are specified in Sections 3 and 4 and detailed in Appendices A and B. Appendix A gives the geometry parameterization, the boundary-condition value conventions, the mesh and interpolation procedure, the per-combination sample counts, and the training/validation split; Appendix B lists the input features, the parameter counts, the loss terms and their weights, and the shared training hyperparameters. Ground-truth fields are produced by COMSOL Multiphysics finite-element solves. The source code is provided as supplementary material.

References

Hamdi E. Ahmed, B. H. Salman, A. Sh. Kherbeet, and M. I. Ahmed. Optimization of thermal design of heat sinks: A review. *International Journal of Heat and Mass Transfer*, 118:129–153, March 2018. ISSN 0017-9310. doi: 10.1016/j.ijheatmasstransfer.2017.10.099.

Haïm Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer, 2010. ISBN 978-0-387-70913-0. doi: 10.1007/978-0-387-70914-7.

Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-Informed Neural Networks for Heat Transfer Problems. *Journal of Heat Transfer*, 143(6), April 2021. doi: 10.1115/1.4050542.

Giovanni Calzolari and Wei Liu. Deep learning to replace, improve, or aid CFD analysis in built environment applications: A review. *Building and Environment*, 206:108315, December 2021. ISSN 0360-1323. doi: 10.1016/j.buildenv.2021.108315.

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. *Journal of Scientific Computing*, 92(3):88, July 2022. ISSN 1573-7691. doi: 10.1007/s10915-022-01939-z.

Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 2 edition, 2010. ISBN 978-0-8218-4974-3. doi: 10.1090/gsm/019.

C. X. He, Q. L. Yue, M. C. Wu, Q. Chen, and T. S. Zhao. A 3d electrochemical-thermal coupled model for electrochemical and thermal analysis of pouch-type lithium-ion batteries. *International Journal of Heat and Mass Transfer*, 181:121855, 2021. doi: 10.1016/j.ijheatmasstransfer.2021.121855.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, June 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5.

Changlong Li, Naxin Cui, Chunyu Wang, and Chenghui Zhang. Simplified electrochemical lithium-ion battery model with variable solid-phase diffusion and parameter identification over wide temperature range. *Journal of Power Sources*, 497:229900, June 2021. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2021.229900.

Table 3: Thermophysical properties of silica glass

Parameter	Symbol	Value	Unit
Baseline thermal conductivity	k_0	1.38	W/(m·K)
Density	ρ	2203	kg/m^3
Specific heat capacity	C_p	703	J/(kg·K)

Yingli Li, Kun Zhou, Pengfei Tan, Shu Beng Tor, Chee Kai Chua, and Kah Fai Leong. Modeling temperature and residual stress fields in selective laser melting. *International Journal of Mechanical Sciences*, 136:24–35, February 2018. ISSN 0020-7403. doi: 10.1016/j.ijmecsci.2017.12.001.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5.

Cody J. Permann, Derek R. Gaston, David Andrš, Robert W. Carlsen, Fande Kong, Alexander D. Lindsay, Jason M. Miller, John W. Peterson, Andrew E. Slaughter, Roy H. Stogner, and Richard C. Martineau. MOOSE: Enabling massively parallel multiphysics simulation. *SoftwareX*, 11: 100430, January 2020. ISSN 2352-7110. doi: 10.1016/j.softx.2020.100430.

Cheng Qian, Amir Mirza Gheitaghy, Jiajie Fan, Hongyu Tang, Bo Sun, Huaiyu Ye, and Guoqi Zhang. Thermal Management on IGBT Power Electronic Devices and Modules. *IEEE Access*, 6: 12868–12884, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2793300.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. doi: 10.1016/j.jcp.2018.10.045.

Qian Wang, Bin Jiang, Bo Li, and Yuying Yan. A critical review of thermal management models and solutions of lithium-ion batteries for the development of pure electric vehicles. *Renewable and Sustainable Energy Reviews*, 64:106–128, October 2016. ISSN 1364-0321. doi: 10.1016/j.rser.2016.05.033.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 43 (5):A3055–A3081, January 2021a. doi: 10.1137/20M1318043.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40), October 2021b. doi: 10.1126/sciadv.abi8605.

Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. *ACM Computing Surveys*, 55(4):66:1–66:37, November 2022. ISSN 0360-0300. doi: 10.1145/3514228.

Xiaoyu Zhao, Zhiqiang Gong, Yunyang Zhang, Wen Yao, and Xiaoqian Chen. Physics-informed convolutional neural networks for temperature field prediction of heat source layout without labeled data. *Engineering Applications of Artificial Intelligence*, 117:105516, January 2023. ISSN 0952-1976. doi: 10.1016/j.engappai.2022.105516.

A Dataset Construction Details

This appendix collects the simulation settings and dataset statistics summarized in Section 3. Table 3 lists the thermophysical properties of the simulated solid, Table 4 the scanning ranges of the operator parameters, and Table 5 the boundary condition value conventions.

Table 6 reports the per-direction sample counts of the no-heat-source dataset, Table 7 the scanned parameters and sample counts per combo for the heat-source dataset, and Table 8 the resulting training/validation split.

Table 4: Scanning ranges of operator parameters

Parameter	Range	Step	Count
c ($k = c \cdot 1.38$ W/(m·K))	1.0–2.0	0.1	11
h (W/(m^2·K))	10–100	10	10
W (m)	0.010–0.020	0.001	11
H (m)	0.010–0.019	0.001	10

Table 5: Boundary condition value conventions

State	Dirichlet (D, b1)	Neumann (N, b2)	Robin (R, b3)
Active	$T_0 = 297.15$ K	$q_0 = 1000$ W/m^2	$T_{\text{ext}} = 303.15$ K
Homogeneous	$T_0 = 293.15$ K	$q_0 = 0$ W/m^2	$T_{\text{ext}} = 293.15$ K

Table 6: Sample counts per direction for the no-heat-source dataset (by combo)

Direction	Active boundary	Ref boundary	Total samples (9 combos)
reb1	Boundary 1 active	Boundary 2 homoge-neous	30,368
reb2	Boundary 1 homoge-neous	Boundary 2 active	30,368
Total			**60,736**

Table 7: Scanning parameters and sample counts per combo for the heat-source dataset

Combo	Boundary 1	Boundary 2	Scanned parameters	Samples
b1b1	D(0)	D(0)	W, H	52
b1b2	D(0)	N(0)	c, W, H	572
b1b3	D(0)	R(0)	h, c, W, H	5,720
b2b1	N(0)	D(0)	c, W, H	572
b2b2	N(0)	N(0)	c, W, H	572
b2b3	N(0)	R(0)	h, c, W, H	5,720
b3b1	R(0)	D(0)	h, c, W, H	5,720
b3b2	R(0)	N(0)	h, c, W, H	5,720
b3b3	R(0)	R(0)	h, c, W, H	5,720
Total				**30,368**

Table 8: Training and validation set split for the two datasets

Dataset	Total samples	Training / Validation
SourceNet	30,368	27,328 / 3,040
BoundaryNet	60,736	54,662 / 6,074

B Network, Training, and Ablation Details

This appendix collects the architecture, training, and ablation details summarized in Sections 4 and 5. Table 9 gives the per-module parameter counts of the two subnetworks, Tables 12 and 13 their input features, Table 14 the Z-score statistics, Table 15 the loss terms and their weights, and

Table 9: Parameter count per module for BoundaryNet and SourceNet

Module	BoundaryNet	SourceNet
MLP Encoder	**2,637,824**	**2,635,776**
To-Map	16,793,600	16,793,600
Block 1 (128→128)	296,128	296,128
Block 2 (128→96)	189,024	189,024
Block 3 (96→64)	91,200	91,200
Block 4 (64→32)	27,936	27,936
Head (32→16→1)	5,137	5,137
Total	**20,040,849**	**20,043,585**

Table 10: SourceNet architecture ablation study

Model	Parameters (M)	RMSE (mK)	max|err| (mK)	R^2
SourceNet	20.0	**0.423**	**13.96**	**0.9999985**
No-CC CNN	20.0	0.522	43.87	0.9999977
MLP	52.0	0.508	12.71	0.9999979

Table 11: BoundaryNet architecture comparison (temperature rise in mK)

Model	Parameters	RMSE (mK)	$\mathbf{max}$ $\|$err$\|$ (mK)	$\mathbf{R^2}$
BoundaryNet	20.0 M	1.361	**208.54**	0.999998
No-cc CNN	20.0 M	1.757	231.29	0.999997
MLP	52.0 M	1.687	224.44	0.999997

Table 16 the shared training hyperparameters. Table 17 lists the input features of DirectNet, and Tables 10 and 11 report the architecture ablations of Section 5-B.

The continuous input features (c, h, W, H) are Min-Max normalized over the training set,

$$x_{\text{cont}}^{\text{norm}} = \frac{x_{\text{cont}} - x_{\min}}{x_{\max} - x_{\min} + \epsilon} \tag{12}$$

which places them on the same scale as the one-hot encodings; the one-hot encodings require no further normalization. The output temperature rise fields $\Delta T = T - 293.15$ are normalized by a global Z-score over the training set,

$$y_{\text{norm}} = \frac{\Delta T - \mu_y}{\sigma_y} \tag{13}$$

with the statistics of Table 14.

C Transfer Conditions and Applicability Limits

The decomposition of Section 2 applies to any steady-state problem satisfying four conditions: (1) the governing equation is linear in the field variable, and the material coefficients do not depend on the field; (2) once the operator parameters are fixed, both the differential and the boundary operator retain their form; (3) the load terms enter linearly, i.e., the source term, the imposed non-homogeneous boundary values, and the ambient reference value appear on the right-hand side of the composite operator equation; and (4) the problem is steady-state. A transient problem would add time as an additional coordinate, and the decomposition would then have to be applied in a space–time domain or coupled with a time-stepping scheme.

Table 12: BoundaryNet input features

No.	Feature	Type	Dim.	Range	Description
1–2	direction	one-hot	2	[1,0]=reb1, [0,1]=reb2	Heating boundary direction
3–5	active_type	one-hot	3	[1,0,0]=D, [0,1,0]=N, [0,0,1]=R	Active boundary type
6–8	ref_type	one-hot	3	Same as above	Ref boundary type
9	c	float	1	$[1.0,\ 2.0]$	Conductivity scaling factor $k = c \cdot 1.38$
10	h	float	1	$[0,\ 100]$	Convective heat transfer coefficient (valid for Robin)
11	W	float	1	$[0.010,\ 0.020]$	Rectangle length (m)
12	H	float	1	$[0.010,\ 0.019]$	Rectangle width (m)

Table 13: SourceNet input features

No.	Feature	Type	Dim.	Range	Description
1–3	b1_type	one-hot	3	[1,0,0]=D, [0,1,0]=N, [0,0,1]=R	Boundary 1 homogeneous type
4–6	b2_type	one-hot	3	Same as above	Boundary 2 homogeneous type
7	c	float	1	$[1.0,\ 2.0]$	Conductivity scaling factor
8	h	float	1	$[0,\ 100]$	Convective heat transfer coefficient
9	W	float	1	$[0.010,\ 0.020]$	Rectangle length (m)
10	H	float	1	$[0.010,\ 0.019]$	Rectangle width (m)

Table 14: Dataset Z-score statistical parameters

Network	$\mu_{\mathbf{y}}$ (K)	$\sigma_{\mathbf{y}}$ (K)	Numerical range (K)
BoundaryNet	0.839	0.995	$[-0.26,\ 9.63]$
SourceNet	0.456	0.352	$[0,\ 2.14]$

Table 15: Definitions and weights of the multi-task loss function terms

Loss term	Weight	Formula	Role
MSE main loss	1.0	$\| y_{\text{pred}} - y_{\text{true}} \|_2^2$	Overall temperature field fitting
Gradient loss	0.05	$\| \nabla y_{\text{pred}} - \nabla y_{\text{true}} \|_1$	Spatial smoothness of the temperature field
Boundary-weighted MSE	0.5	$\| (y_{\text{pred}} - y_{\text{true}}) \odot W_{\text{boundary}} \|_2^2$	Increase weight on boundary/corner regions
Top-K loss	0.1	$\text{topk}(\| y_{\text{pred}} - y_{\text{true}} \|, k = 1\%)$	Reduce maximum error

Table 18 lists the correspondence between the thermal notation used in this paper and that of several other steady-state linear diffusion systems. For each, the workflow of Section 4 is unchanged – one base subfield with all sources active and every controllable boundary homogeneous, one boundary-response subfield per controllable boundary with the other loads removed, and superposition for an arbitrary load – and only the input layer must be rewritten, with that system's operator parameters. For mass diffusion a reaction or injection source takes the role of the heat source and the film mass transfer coefficient that of the convective coefficient; for the electrostatic problem permittivity, charge density, and surface conductivity replace k, q, and h; for porous media flow and neutron diffusion the permeability and diffusion coefficients play the role of k. Because the subfields are

Table 16: Shared training hyperparameters for the two subnetworks

Parameter	Value
Optimizer	AdamW
Initial learning rate	1×10^{-3}
Learning rate schedule	ReduceLROnPlateau, factor=0.5, patience=10
Batch size	64
Max epochs	200
Early stopping patience	30
Gradient clipping	max_norm = 1.0

Table 17: DirectNet input features (16 dimensions)

No.	Feature	Type	Dim.	Description
1–3	b1_type	one-hot	3	Boundary 1 type (D/N/R)
4–6	b2_type	one-hot	3	Boundary 2 type (D/N/R)
7–8	b1_state	one-hot	2	Boundary 1 state (active / homogeneous)
9–10	b2_state	one-hot	2	Boundary 2 state (active / homogeneous)
11–12	heat_flag	one-hot	2	Whether volumetric heat source is present
13–16	c, h, W, H	float	4	Operator parameters (Min-Max normalized)

Table 18: Correspondence of the diffusion-type equation across physical systems

Physical System	Field Variable ϕ	Diffusion Coefficient $\mathbf{D}$	Source Term $\mathbf{S}$	Transport Coefficient α
Heat conduction	Temperature T	Thermal conductivity k	Heat source q	Convective heat transfer coefficient h
Mass diffusion	Concentration c	Diffusion coefficient D_m	Reaction source R	Film mass transfer coefficient k_c
Electrostatic potential	Electric potential V	Permittivity ε	Charge density ρ_e	Surface conductivity σ
Porous media flow	Hydraulic head H	Permeability coefficient K	Source/sink Q	–
Neutron diffusion	Neutron flux ψ	Diffusion coefficient D_n	Fission source s	Extrapolation coefficient

combined analytically, the number of network evaluations is independent of the number of loads, the property exploited in Section 5.

Three classes of problem fall outside the framework. Nonlinear problems, where a material coefficient depends on the field or the boundary condition is itself nonlinear (e.g., thermal radiation), leave the operator varying with the solution, so no fixed set of subfields can be recombined. Coupled multiphysics problems, where the temperature enters the coefficients of another field equation or vice versa, lose the separation between operator and load parameters. Problems with a moving boundary or a geometry that changes between operating conditions are also excluded, since geometry is itself an operator parameter.

The two parameter classes are distinguished by where a parameter appears in the composite operator equation, not by its physical name: one that multiplies the field variable, or enters the differential or boundary operator, is an operator parameter, whereas one forming the right-hand side is a load parameter. The same criterion settles borderline cases – a temperature-dependent conductivity and a radiation boundary condition violate conditions (1) and (2), and a fixed non-homogeneous Dirichlet value is a load parameter just as the heat source is.

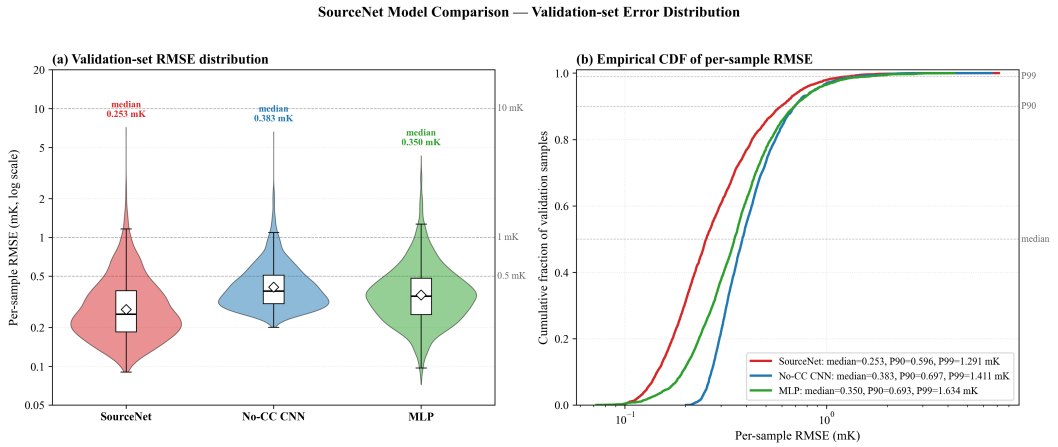

Figure 2: Per-sample RMSE distribution of SourceNet and comparison models on the validation set.

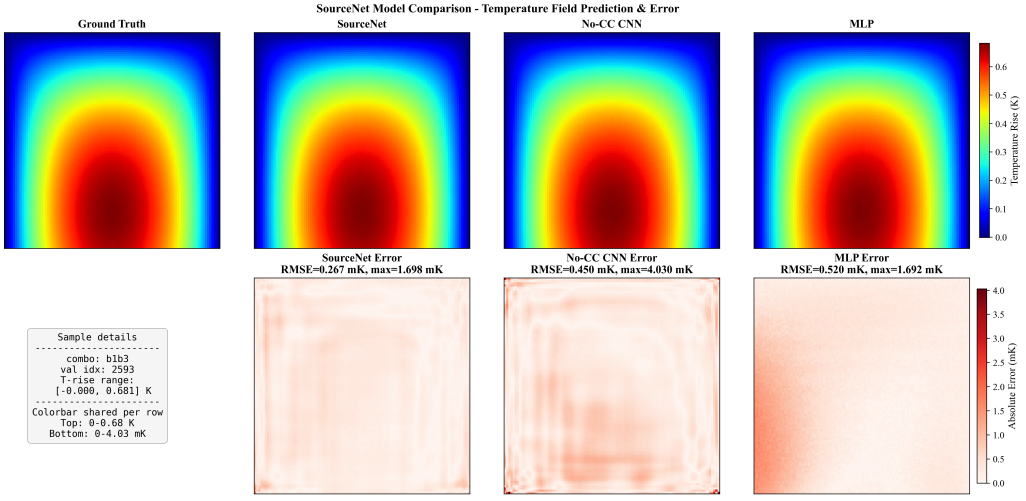

Figure 3: Temperature field prediction and error comparison of SourceNet and comparison models on a single validation sample.

D Ablation Study Details

Tables 10 and 11 and Figs. 2–5 give the full architecture ablations summarized in Section 5-B. On SourceNet, removing CoordConv raises $max \parallel$ err $\parallel$ from 13.96 mK to 43.87 mK, a deterioration of 214%, because without explicit position cues the predictions at boundaries and corners become severely inaccurate; the median per-sample RMSE is 0.253 mK, about 34% and 28% below the No-cc CNN and MLP baselines, with 85.4% of samples below 0.5 mK against 73.0% and 76.9%. A pure MLP needs 2.6 times the parameter count (52 M vs. 20 M) for a comparable RMSE that is still 20% worse. On BoundaryNet the same ordering holds: its RMSE of 1.361 mK leads the No-cc CNN (1.757 mK) and the MLP (1.687 mK) by 22.6% and 19.3%, and its median per-sample RMSE of 0.823 mK is about 38% and 29% lower, with 59.7% of samples below 1 mK against 33.0% and 40.4%. In both cases the ECDF curve of the proposed model lies entirely to the upper left of the two baselines, i.e. it exhibits first-order stochastic dominance. The single-sample error fields of Figs. 3 and 5 show the same pattern in space: the proposed models have the most uniform error fields (RMSE 0.267 mK for SourceNet, 0.809 mK for BoundaryNet), while No-cc CNN develops a pronounced high-error band at the active boundary, with $max \parallel$ err $\parallel$ 2.4 and 2.2 times larger respectively.

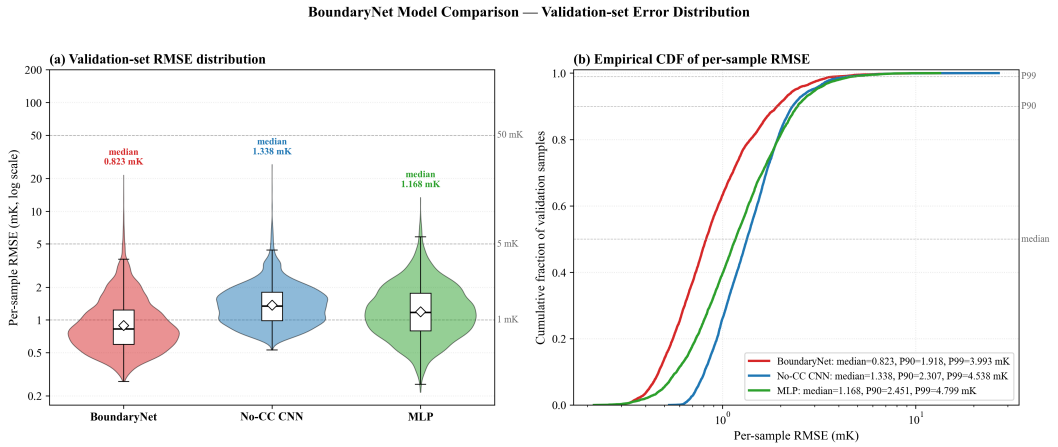

Figure 4: Per-sample RMSE distribution of BoundaryNet and comparison models on the validation set.

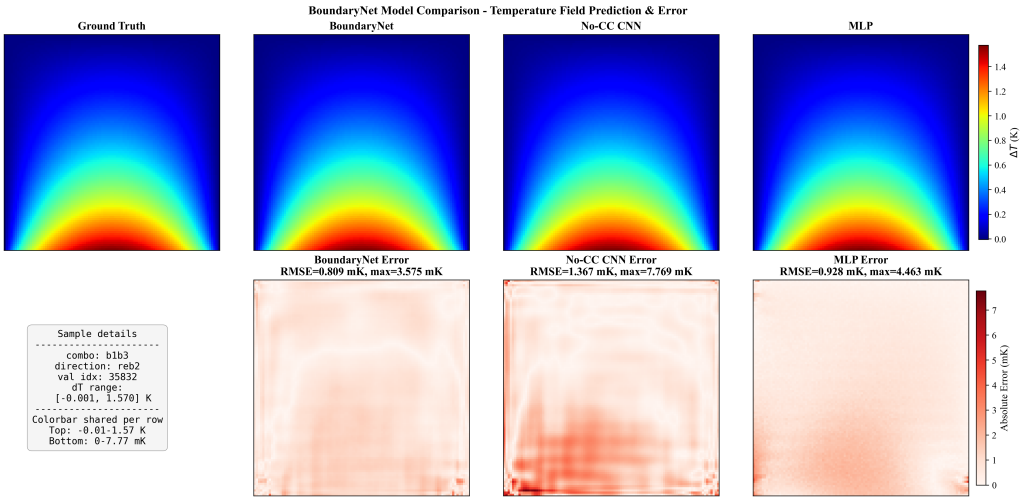

Figure 5: Temperature field prediction and error comparison of BoundaryNet and comparison models on a single validation sample.

Table 19: Inference efficiency and speedup ratio of SINN vs. COMSOL FEM

Method	Single-run time	Speedup ratio
FEM (COMSOL, single complete solution)	2119 ms	1×
SINN (GPU, single sample)	9.27 ms	**229**×
SINN (GPU, batch of 256)	2.16 ms/sample	**980**×
SINN (CPU, single sample)	52.2 ms	41×

E Additional Figures and Tables

Figures 6 and 7 show the predicted temperature and error fields of SINN and DirectNet on the in-domain and unseen-load samples discussed in Section 5-C; Table 20 lists the categories of comparison objects used in Section 5-A, and Table 19 the inference timings of Section 5-D.

Table 20: Classification of comparison objects

Category	Comparison object
Network-level	No-cc CNN (No CoordConv CNN), MLP
Method-level	DirectNet (purely data-driven full-field prediction)
Baseline	FEM (COMSOL, finite element ground truth)

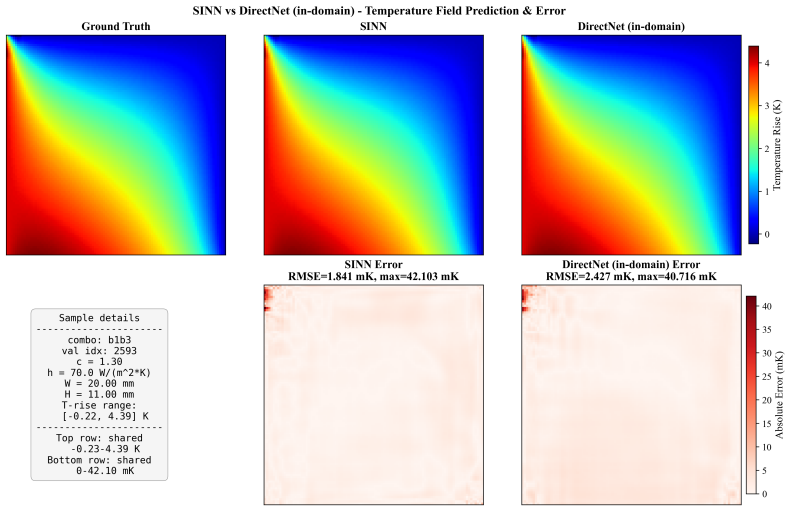

Figure 6: Temperature field prediction and error comparison of SINN and DirectNet on a single validation sample.

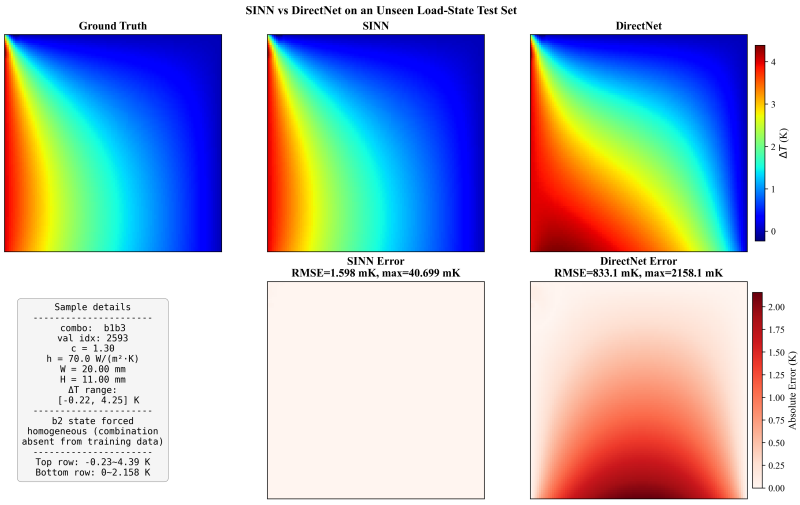

Figure 7: Temperature field prediction and error comparison of SINN and DirectNet on a sample from the unseen load-state test set.