

Article

Thermal Digital Datasheet: A Standardized Information Representation for Modular Satellite Thermal Management and Graph Neural Network-Based Temperature Prediction

Weijian Pang ^{1,2,*} , Jun Zhou ², Jingwen Xu ^{2,*} and Xinian Zhi ¹¹ Ningbo Institute of Northwestern Polytechnical University, Ningbo 315103, China² Institute of Precision Guidance and Control, School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China

* Correspondence: pangwj@mail.nwpu.edu.cn (W.P.); xjw0324@nwpu.edu.cn (J.X.)

Abstract

Modular satellite architectures can shorten development cycles and support flexible mission configurations; however, their thermal management is complicated by inter-module coupling, variable operating conditions, and stringent temperature limits. This study proposes a Thermal Digital Datasheet (TDD) framework that provides a standardized representation of module-level thermal information for data exchange, rapid system analysis, and consistent characterization across suppliers. The TDD organizes the passive thermal properties, active control capabilities, and operational constraints of each module in a unified format. To populate the datasheet with physically consistent parameters, a physics-constrained algebraic identification method is developed using excitation-based thermal response data and a structured least-squares formulation derived from the energy-balance equations. The resulting TDD representation is then integrated with a graph neural network (GNN), in which the modular interconnection topology is represented explicitly as a graph for network-wide temperature prediction. On the simulated modular satellite dataset, TDD-GNN achieves a mean absolute error of 0.096 °C and an R^2 of 0.962 for one-step temperature-increment prediction, maintaining high accuracy over autoregressive horizons of up to 2 h. In an end-to-end evaluation in which the thermal parameters of each test configuration are independently identified before GNN inference, the model retains $R^2 = 0.959$, demonstrating robustness to realistic parameter-identification errors. Perturbation-based sensitivity analysis further shows that the learned parameter ranking is consistent with the expected thermal behavior. These simulation results indicate that the proposed framework can support computationally efficient thermal-state prediction for modular satellite systems.

Keywords: Thermal Digital Datasheet; graph neural network; parameter identification; thermal-state prediction; modular satellite; parameter sensitivity



Received: 23 July 2026

Revised: 29 August 2026

Accepted: 2 September 2026

Published: 3 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

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

1. Introduction

Modular satellite systems have attracted increasing interest because they can reduce development time, support mission reconfiguration, and facilitate module replacement or reuse [1]. In contrast to monolithic spacecraft, modular platforms comprise interconnected functional units that may be assembled or reconfigured according to mission requirements. This architectural flexibility, illustrated by modular spacecraft concepts developed by

major space agencies, also creates additional thermal-management challenges that must be addressed to maintain component reliability and mission availability.

Thermal control is a fundamental aspect of spacecraft design because electronic components and scientific instruments must remain within specified temperature limits [2]. In a modular satellite, this requirement is complicated by (1) time-varying thermal coupling among modules with different thermal properties; (2) orbital variations in solar incidence, albedo, and planetary infrared loading; (3) the need for standardized thermal interfaces between modules supplied by different manufacturers; and (4) the demand for sufficiently fast state estimation and prediction to support autonomous operation. These factors motivate thermal models that are both physically interpretable and computationally efficient.

Spacecraft thermal analysis is commonly performed using finite-element models, thermal-network models, and, where appropriate, computational fluid dynamics [3]. Although high-fidelity physics-based models can provide accurate predictions, their computational cost may preclude repeated onboard or real-time evaluation. Data-driven methods, including feedforward and recurrent neural networks, have therefore been investigated for thermal prediction [4]. Nevertheless, purely data-driven models may be difficult to interpret and may not preserve known energy-balance relationships. In addition, the absence of a standardized representation of module-level thermal information complicates data exchange among tools, integration of modules from different suppliers, and system-level analysis.

To address these issues, this paper introduces a Thermal Digital Datasheet (TDD) as a standardized, physics-informed representation of the thermal characteristics of a modular spacecraft. The TDD describes the passive properties, active-control capabilities, and operating constraints of each module in a machine-readable structure. This common representation is intended to support consistent model exchange and system-level integration. The principal contributions are as follows:

1. **Standardized TDD representation** : A structured information model is defined for module-level thermal properties, control capabilities, and operational constraints, providing a common basis for thermal-data exchange and system-level modeling.
2. **Physics-constrained algebraic identification**: An excitation-based identification method is formulated to estimate the unknown TDD parameters from thermal-response data. The method constructs a structured linear least-squares problem from the discretized energy-balance equations and solves it using SVD subject to physically admissible parameter bounds.
3. **Graph-based prediction framework**: An edge-aware graph neural network is developed to predict the temperature increment of every module from the current operating state and the TDD parameters. The residual formulation supports iterative, horizon-flexible rollout while explicitly representing inter-module thermal coupling. The complete TDD-to-GNN pipeline is hereafter referred to as TDD-GNN.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 defines the thermal-network model, the TDD parameter set, and the graph representation. Section 4 presents the identification procedure, the TDD-GNN architecture, and the training methods. Section 5 reports the experimental results, and Section 6 concludes.

2. Related Work

Research relevant to modular satellite thermal management spans spacecraft thermal modeling, standardized system representation, parameter identification, and data-driven prediction. Although each topic has been studied separately, few frameworks combine physically interpretable parameter characterization with graph-based prediction for modular architectures. The following review is organized according to the methodological sequence adopted in this study.

2.1. Thermal Modeling for Spacecraft

Spacecraft thermal models range from reduced-order lumped representations to high-fidelity finite-element formulations. In general, the governing heat-transfer equations are solved subject to boundary conditions describing solar irradiation, planetary infrared radiation, internal dissipation, conductive paths, and radiation to space [2,5]. The European Space Agency has also established thermal-control standards covering analysis methods, verification procedures, and acceptance criteria for spacecraft thermal systems [6].

Lumped-parameter thermal networks, often expressed as resistance–capacitance (RC) models, provide a computationally efficient compromise between model fidelity and complexity [7]. Each node represents a region assumed to have a spatially uniform temperature, whereas the links describe conductive or linearized radiative heat exchange. Thermal capacitance accounts for energy storage and therefore enables transient analysis. RC networks have been used extensively in spacecraft thermal-balance studies and orbital simulations.

Data-driven models have also been investigated as reduced-order predictors of thermal dynamics. Neural ordinary differential equations can learn continuous-time state evolution from observations [8], whereas long short-term memory (LSTM) networks can represent temporal dependencies in measured temperature sequences [9]. More recently, deep reinforcement learning has been explored for autonomous onboard thermal control of small satellites [10]. Their use in safety-critical applications, however, may be limited when the learned dynamics lack physical interpretability or do not satisfy known conservation relationships.

2.2. Standardized Information Representation and Parameter Identification

Digital representations of physical systems have evolved from computer-aided design models toward digital-twin frameworks that integrate models, measurements, and operational services [11]. A digital twin can support simulation, monitoring, and optimization throughout the lifecycle of a physical asset. Tao et al. proposed a digital-twin framework that integrates physical models, data, and services for smart manufacturing [12]; related concepts have subsequently been considered for spacecraft monitoring, maintenance, and operational planning.

Standardized interfaces are particularly important in modular systems because interoperability must be maintained across suppliers. In aerospace applications, Platform Support Package concepts specify mechanical, electrical, and thermal interfaces to support module interchangeability [13]. Parameterized thermal models have likewise been proposed to represent module-specific behavior within a common modeling structure [7,14]. Existing approaches, however, do not always provide a complete and explicitly machine-readable description of the parameters, constraints, and uncertainty information required for system-level thermal analysis.

The TDD proposed in this paper extends these ideas by defining a physics-based, machine-readable specification of the thermal information required at module level. In addition to nominal parameter values, the representation includes operational constraints and provision for per-parameter uncertainty metadata to support consistent system assembly and analysis in future work.

Populating such a representation requires parameter-identification methods that remain consistent with the governing physics. Two broad approaches are commonly used: neural surrogate models constrained by physical residuals and algebraic methods that exploit the structure of the governing equations. Physics-informed neural networks (PINNs) incorporate observations and differential-equation residuals into a joint training objective [15] and can be useful when observational data are sparse [16,17]. Physics-informed approaches have also been explored for real-time spacecraft thermal simulation [18]. Their

identification performance nevertheless depends on the accuracy with which the neural surrogate represents the underlying state trajectory [19]. When the governing equations are linear in the unknown parameters, an algebraic formulation is often more direct: the coefficients can be estimated using structured least squares with established numerical and statistical properties. The present work adopts this second approach and solves the resulting linear system by SVD under physical bound constraints.

2.3. Graph Neural Networks for Physical Systems

Graph neural networks (GNNs) are designed for data defined on non-Euclidean structures with arbitrary connectivity [20]. Unlike convolutional networks on regular grids, GNNs update each node representation using information exchanged along graph edges. Foundational architectures include graph convolutional networks [21] and graph attention networks [22]. Message-passing neural networks provide a general formulation of this operation as repeated message computation, aggregation, and node-state update [23,24].

GNNs have been applied to several classes of physical systems. In power systems, graph convolutional networks have been used for power-flow approximation, stability assessment, and fault detection [25]. In fluid mechanics, graph-based simulators have been developed for flow-field prediction and fluid–structure interaction [26,27]. Graph models have also been applied to temperature prediction in multicore electronic systems, where spatial heat transfer follows an irregular coupling topology [28].

This relational inductive bias is well matched to modular thermal systems. Representing physical thermal contacts as graph edges allows the model to distinguish connected from unconnected modules and to condition message exchange on the corresponding conductance. The resulting architecture therefore provides a more appropriate representation of inter-module heat transfer than a fully connected model that treats all pairwise interactions identically. In addition, the model response can be interrogated through perturbation- or gradient-based sensitivity analysis.

3. System Modeling and Parameter Definition

3.1. Problem Formulation

Consider a modular thermal system comprising N thermally interacting modules. Module i is described by its temperature $T_i(t)$ and internal heat dissipation $P_i(t)$. Under a lumped-capacitance assumption, its thermal dynamics are written as

$$C_i \frac{dT_i}{dt} = \sum_{j \in \mathcal{N}_i} G_{ij}(T_j - T_i) + P_i + Q_{\text{sol},i} - \varepsilon_i A_{\text{rad},i} \sigma_{\text{SB}} (T_i^4 - T_{\text{space}}^4) - G_{\text{ext},i} (T_i - T_{\text{space}}) \quad (1)$$

where C_i denotes the thermal capacitance of module i (J/K); G_{ij} is the thermal conductance between modules i and j (W/K); $\mathcal{N}_i$ is the set of modules thermally connected to module i ; P_i is the internal power dissipation (W); and $Q_{\text{sol},i}$ denotes the net incident environmental heat load (W), including direct solar radiation, Earth albedo, and planetary infrared radiation. In the radiative-loss term, ε_i is the surface emissivity of module i (dimensionless), $A_{\text{rad},i}$ is its effective radiating area (m^2), σ_{SB} is the Stefan–Boltzmann constant ($5.670 \times 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}$), and T_{space} is the deep-space background temperature (4 K). All temperatures in Equation (1) are expressed in kelvin, including the radiative fourth-power term and the external-leakage term; the $^{\circ}\text{C}$ values used for GNN node features (Equation (3)) are converted to kelvin before evaluating the physical equations. The final term represents effective non-radiative leakage through structural supports, harnesses, and residual conduction through the multilayer-insulation assembly, where $G_{\text{ext},i}$ (W/K) is the corresponding aggregate external conductance. The environmental load, radiating

area, and emissivity are treated as known, whereas $G_{\text{ext},i}$ is identified from response data because it depends strongly on assembly- and interface-specific details.

The same system can be represented as a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$. The node set $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ represents the modules, the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents physical thermal contacts, and the weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ satisfies $A_{ij} = G_{ij}$ for connected modules and $A_{ij} = 0$ otherwise. In vector form, the thermal dynamics are

$$\mathbf{C} \frac{d\mathbf{T}}{dt} = -\mathbf{L}\mathbf{T} + \mathbf{P} + \mathbf{Q}_{\text{sol}} - \mathbf{Q}_{\text{rad}}(\mathbf{T}) - \mathbf{G}_{\text{ext}}(\mathbf{T} - T_{\text{space}}\mathbf{1}) \quad (2)$$

where $\mathbf{T} = [T_1, T_2, \dots, T_N]^\top$ is the temperature vector, $\mathbf{C} = \text{diag}(C_1, C_2, \dots, C_N)$ is the thermal-capacitance matrix, $\mathbf{P} = [P_1, P_2, \dots, P_N]^\top$ is the internal-power vector, and $\mathbf{Q}_{\text{sol}}$ is the environmental heat-load vector. The nonlinear radiative-loss vector $\mathbf{Q}_{\text{rad}}(\mathbf{T})$ has elements $\varepsilon_i A_{\text{rad},i} \sigma_{\text{SB}}(T_i^4 - T_{\text{space}}^4)$. Furthermore, $\mathbf{G}_{\text{ext}} = \text{diag}(G_{\text{ext},1}, \dots, G_{\text{ext},N})$, and the weighted graph Laplacian $\mathbf{L}$ is defined by $L_{ii} = \sum_j G_{ij}$ and $L_{ij} = -G_{ij}$ for $i \neq j$.

The tasks considered in this study are therefore to predict the evolution of $\mathbf{T}(t)$ from the prescribed inputs $\mathbf{P}(t)$ and $\mathbf{Q}_{\text{sol}}(t)$, estimate the unknown thermal parameters, and evaluate the sensitivity of the predictions to inter-module coupling and module-level properties. The TDD provides a common representation of the parameters and constraints required by these tasks.

3.2. Thermal Digital Datasheet Definition

The Thermal Digital Datasheet is defined as a physics-based specification of the thermal information associated with an individual module. Its contents are organized into three groups: passive thermal properties, active-control capabilities, and operational constraints.

The passive properties describe the quantities that govern the uncontrolled thermal response. They include the thermal capacitance C (J/K), the internal thermal resistance R_{int} (K/W), the effective external conductance G_{ext} (W/K), the effective radiating area A_{rad} (m²), and the surface emissivity ε . The active-control fields describe available actuators, including heater power P_{heater} (W), switchable conductance G_{switch} (W/K), and active cooling capacity P_{cool} (W). The constraint envelope specifies the allowable temperature range $[T_{\text{min}}, T_{\text{max}}]$, the maximum temperature rate $|dT/dt|_{\text{max}}$ (K/min), and the maximum sustainable heat dissipation P_{max} (W). Table 1 summarizes the complete parameter set.

Table 1. Complete TDD parameter definitions.

Category	Parameter	Symbol	Units	Physical Meaning	Source
Passive	Thermal Capacitance	C	J/K	Energy storage	Identified
Passive	Internal Resistance	R_{int}	K/W	Internal gradient	Retained *
Passive	External Conductance	G_{ext}	W/K	Parasitic conductance	Identified
Passive	Radiation Area	A_{rad}	m ²	Radiative surface	Datasheet
Passive	Emissivity	ε	—	Surface property	Material spec
Active	Heater Power	P_{heater}	W	Heating capacity	Datasheet
Active	Switch Conductance	G_{switch}	W/K	Controllable link	Calibration
Active	Cooling Capacity	P_{cool}	W	Active cooling	Datasheet
Constraint	Temperature Range	$[T_{\text{min}}, T_{\text{max}}]$	°C	Safe operation	Component spec
Constraint	Rate Limit	$ dT/dt _{\text{max}}$	K/min	Shock prevention	Design margin
Constraint	Power Limit	P_{max}	W	Thermal budget	Engineering

* Retained for future multi-node module models; not used in the present single-node formulation.

Within the lumped-capacitance model used in this study, the TDD quantities are treated as follows:

- Identified from data: The thermal capacitance C , contact conductance G_{ij} , and effective external conductance G_{ext} are estimated from excitation-based thermal-response data using the algebraic method described in Section 4.2.
- Specified a priori: The radiating area A_{rad} and emissivity ε are obtained from mechanical drawings and material characterization, and enter the radiative-loss term of Equation (1) explicitly. The heater rating P_{heater} is obtained from the component datasheet and appears both as a bound on the applied heater command in the simulator and as a static node feature in the GNN (Equation (3), feature 8). The switchable conductance G_{switch} describes a controllable thermal link that is not activated in the present fixed-topology experiments but is retained in the TDD for reconfigurable thermal-architecture scenarios. The active cooling capacity P_{cool} characterizes an actively cooled module; the simulated 6U CubeSat does not include such hardware, so the field is currently unused but is included for forward compatibility with larger platforms that may carry cryocoolers or fluid-loop cold plates.
- Retained for higher-order module models: The internal resistance R_{int} characterizes temperature differences between internal heat-generating components and the module boundary. Because the present model treats each module as a single isothermal node, this internal temperature gradient is not resolved and R_{int} is therefore not used or identified. The field is retained in the TDD so that the representation can be extended to multi-node module models.
- Operational-constraint metadata: The temperature range $[T_{\text{min}}, T_{\text{max}}]$, the rate limit $|dT/dt|_{\text{max}}$, and the power limit P_{max} do not appear in the prediction equations. They define the safe operating envelope of each module and are intended to support downstream thermal-control decisions, safety-override logic in the excitation-sequence generator (Section 4.2.3), and cross-supplier documentation exchange.

Parameters that do not appear in the governing equations therefore fall into one of three categories: (i) GNN input features such as P_{heater} ; (ii) engineering metadata for documentation, safety, and supplier exchange; and (iii) forward-compatible fields reserved for higher-fidelity or reconfigurable module models.

The TDD specification is designed to accommodate per-parameter uncertainty metadata for use in downstream probabilistic analysis; however, the present study treats all parameters as deterministic and does not propagate uncertainty through the identification or prediction stages. Calibration of uncertainty fields using measured or supplier-certified data, and their integration with a probabilistic predictor, are identified as directions for future work (Section 6).

3.3. Graph Representation

The physical module connections define a natural graph that is used by both the RC network and the neural predictor. This common topology ensures that the data-driven model exchanges information only along physically meaningful thermal paths.

3.3.1. Node Definition

Each thermal module is represented by one graph node. The feature vector $\mathbf{v}_i \in \mathbb{R}^8$ combines the instantaneous operating state with the module-specific TDD quantities:

$$\mathbf{v}_i = [T_i, P_i, Q_{\text{sol},i}, s_i, C_i, G_{\text{ext},i}, \varepsilon_i A_{\text{rad},i}, P_{\text{heater},i}]^T \quad (3)$$

where T_i is the current module temperature ($^{\circ}\text{C}$), P_i is the internal power dissipation (W), $Q_{\text{sol},i}$ is the environmental heat input (W), and $s_i \in \{0, 1\}$ indicates the heater state. The remaining entries are the thermal capacitance C_i (J/K), effective external conductance $G_{\text{ext},i}$ (W/K), radiative area–emissivity product $\varepsilon_i A_{\text{rad},i}$ (m^2), and rated heater power $P_{\text{heater},i}$.

(W). The first four entries vary with time, whereas the final four describe static module properties. Their combination allows the network to condition the predicted dynamics on both the current operating point and the thermal characteristics of each module.

3.3.2. Edge Definition

A graph edge represents a physical thermal contact. Its scalar feature $\mathbf{e}_{ij} \in \mathbb{R}^1$ is the corresponding contact conductance:

$$\mathbf{e}_{ij} = [G_{ij}]^\top \quad (4)$$

where G_{ij} is the contact conductance between modules i and j (W/K). Each physical contact is represented by two directed edges so that heat exchange can be modeled in both directions. The weighted adjacency matrix is defined by

$$A_{ij} = \begin{cases} G_{ij}^{\text{nominal}} & \text{if modules } i \text{ and } j \text{ are thermally connected} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

For graph operations that require normalized connectivity, the degree-normalized adjacency matrix is

$$\tilde{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \quad (6)$$

where $\mathbf{D}$ is the weighted degree matrix, with $D_{ii} = \sum_j A_{ij}$.

4. Methodology

The proposed methodology consists of three connected components. First, excitation data are used to identify the unknown module and interface parameters required by the TDD. Second, the resulting graph-structured representation is supplied to the TDD-GNN for one-step residual prediction and autoregressive rollout. Third, the trained model is evaluated through parameter perturbations and computational profiling. Figure 1 summarizes the complete workflow.

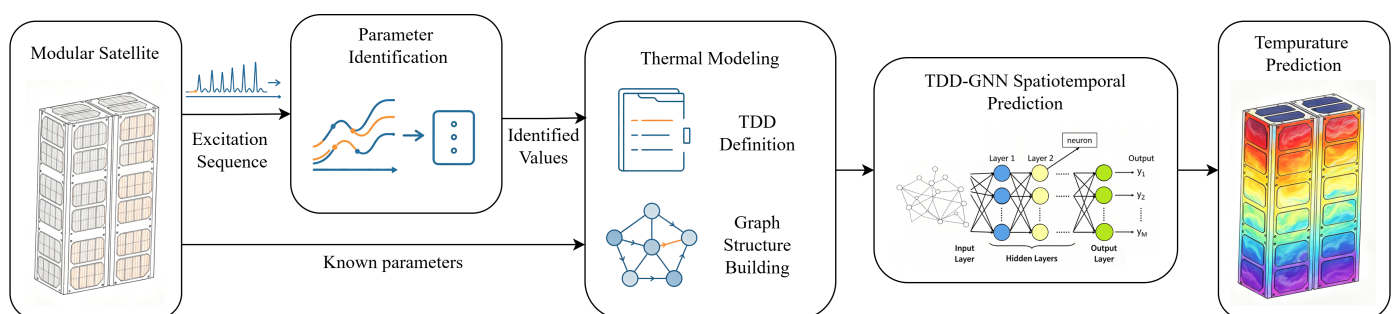


Figure 1. Overview of the proposed workflow, comprising thermal-system representation, physics-constrained parameter identification, TDD construction, and GNN-based temperature prediction.

4.1. Overall Framework

The modular spacecraft is first represented as a lumped-parameter RC network with the TDD providing a common schema for module and interface data (Section 3). The unknown capacitances and conductances are then estimated from excitation-based thermal-response trajectories, and the identified parameters are encoded as graph features for the TDD-GNN. Repeated application of the one-step predictor provides multi-step forecasts, and perturbation analysis assesses whether the learned parameter dependence is physically plausible.

4.2. Physics-Constrained Algebraic Identification

The unknown TDD parameters must be estimated from experimental measurements or high-fidelity simulation data without violating the structure of the thermal model. The proposed method exploits the fact that the discretized network equations are linear in the unknown capacitances and conductances, thereby reducing the identification problem to structured least squares.

4.2.1. Discretized Thermal Network Formulation

Using a sampling interval Δt , Equation (1) is discretized at time index n as

$$C_i \frac{T_i(n+1) - T_i(n)}{\Delta t} = \sum_{j \in \mathcal{N}_i} G_{ij} [T_j(n) - T_i(n)] + P_i(n) + Q_{\text{sol},i}(n) - \varepsilon_i A_{\text{rad},i} \sigma_{\text{SB}} [T_i(n)^4 - T_{\text{space}}^4] - G_{\text{ext},i} [T_i(n) - T_{\text{space}}] \quad (7)$$

where $T_i(n)$ is the measured temperature of module i at time index n . The unknowns are the module capacitances $\{C_i\}_{i=1}^N$, the contact conductances $\{G_{ij}\}_{(i,j) \in \mathcal{E}}$, and the effective external conductances $\{G_{\text{ext},i}\}_{i=1}^N$. The environmental heat input is decomposed as

$$Q_{\text{sol},i}(n) = q_{\text{solar},i}(n) + q_{\text{albedo},i}(n) + q_{\text{IR},i}(n) \quad (8)$$

where σ_{SB} is the Stefan–Boltzmann constant and $T_{\text{space}} = 4 \text{ K}$ is the assumed deep-space background temperature. Both the radiative-loss term and the external-leakage term are evaluated with temperatures converted to kelvin, consistent with Equation (1). By contrast, $G_{\text{ext},i}$ represents aggregate non-radiative leakage through supports, harnesses, and residual insulation conduction; because these paths depend on assembly details, they are estimated from the response data.

4.2.2. Structured Least-Squares Identification

Equation (7) is linear in the unknown parameter vector

$$\boldsymbol{\theta} = [C_1, \dots, C_N, G_{ij}^{(1)}, \dots, G_{ij}^{(M)}, G_{\text{ext},1}, \dots, G_{\text{ext},N}]^\top,$$

where M is the number of physical thermal contacts. Stacking the equations for all modules and sampled time indices gives

$$\mathbf{H} \boldsymbol{\theta} = \mathbf{b} \quad (9)$$

Each row of $\mathbf{H}$ contains coefficients computed from the measured temperatures and the prescribed power and environmental inputs, whereas $\mathbf{b}$ contains the known heat-input and radiative terms. The stacked system is generally overdetermined, although insufficiently excited data can produce strong collinearity or rank deficiency.

The parameter estimate is obtained using an SVD-based least-squares solution:

$$\hat{\boldsymbol{\theta}} = \mathbf{H}^+ \mathbf{b} = \mathbf{V} \boldsymbol{\Sigma}^+ \mathbf{U}^\top \mathbf{b} \quad (10)$$

where $\mathbf{H}^+$ denotes the Moore–Penrose pseudoinverse and $\mathbf{H} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^\top$ is the singular value decomposition. Singular values below the prescribed numerical tolerance are discarded, yielding a stable minimum-norm estimate when the design matrix is ill-conditioned or rank-deficient.

4.2.3. Excitation Sequence Design

Parameter identifiability depends on the dynamic content of the measurements. Under nearly constant power input, the system can approach a quasi-steady state, in which case the capacitance terms become weakly observable because $dT_i/dt \approx 0$. A three-phase heater sequence is therefore designed to emphasize different groups of parameters:

1. Phase 1: Individual-module pulses for capacitance identification. Each of the $N = 6$ modules receives a short pulse consisting of three on-steps followed by five off-steps. Sequential excitation of all modules produces 48 time steps per cycle. The initial temperature slope following each pulse provides direct information about C_i .
2. Phase 2: Differential heating for contact-conductance identification. For each of the $M = 7$ contacts, one endpoint is heated for three steps and then left unheated for three steps, generating a temperature difference across that interface. The resulting 42-step phase increases the information available for estimating the associated G_{ij} .
3. Phase 3: Passive cooling for external-conductance identification. All heaters are disabled for 30 steps. Because the radiative term is computed from the specified surface properties, the remaining cooling response provides information about $G_{\text{ext},i}$.

One excitation cycle contains 120 time steps, corresponding to 2 h for $\Delta t = 60$ s, and is repeated over the trajectory. A safety override disables or modifies the commanded heater state when a module approaches $T_{\text{max}} - 2$ K or $T_{\text{min}} + 2$ K. The resulting excitation matrix is intended to provide sufficient rank for the simultaneous estimation of 19 parameters ($6C + 7G_{ij} + 6G_{\text{ext}}$).

4.2.4. Physical Bound Constraints

The unconstrained least-squares estimate is projected onto physically admissible intervals:

$$\hat{\theta}_j \leftarrow \text{clip}(\hat{\theta}_j, \theta_{j,\text{min}}, \theta_{j,\text{max}}) \quad (11)$$

The bounds are selected from module mass and heat-capacity estimates, contact geometry and material conductivity, structural leakage paths, and interface-resistance information. This projection prevents negative or otherwise nonphysical parameter values, although it can also introduce boundary bias when the unconstrained estimate lies outside the selected range.

4.2.5. Shared-Parameter Identification

For a fixed satellite configuration, trajectories generated under different initial conditions and environmental inputs share one underlying parameter vector. These trajectories are therefore stacked into a single identification system. Consolidation increases the number of equations and reduces the influence of trajectory-specific measurement noise, provided that the assumed parameters remain constant across the combined data.

4.3. TDD-GNN Prediction Architecture

The TDD-GNN represents the instantaneous thermal state on the module graph and predicts the corresponding one-step temperature increment. Multi-step forecasts are obtained by autoregressive application of the same residual model.

4.3.1. Residual Prediction and Autoregressive Rollout

The predictor does not require an explicit history window. Instead, it estimates the module-wise increment $\Delta T_i = T_i(t + \Delta t) - T_i(t)$ from the current graph state:

$$\Delta \hat{\mathbf{T}} = f_{\text{GNN}}(\mathbf{V}(t), \mathbf{E}, \mathbf{G}) \quad (12)$$

Here, $\mathbf{V}(t)$ is the node-feature matrix, $\mathbf{E}$ contains the edge features, and $\mathbf{G}$ specifies the graph connectivity. The next temperature is reconstructed as $\hat{T}_i(t + \Delta t) = T_i(t) + \Delta \hat{T}_i$.

For an H -step forecast, the predicted temperature at one step is used as the temperature input at the next step, while the prescribed power, heater, and environmental inputs are advanced according to the known input sequence. Because the same one-step predictor is reused, the forecast horizon can be varied without retraining.

4.3.2. Message Passing Framework

The graph layers follow the message-passing formulation of Gilmer et al. [23]:

1. Message computation: For each directed edge (j, i) , a message from node j to node i is computed as

$$\mathbf{m}_{ij} = \text{MLP}\left(\left[\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij}\right]\right) \quad (13)$$

2. Message aggregation: The incoming messages at node i are summed:

$$\mathbf{a}_i^{(l)} = \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ij} \quad (14)$$

3. Node update: The aggregated message is combined with the current node state:

$$\mathbf{h}_i^{(l+1)} = \text{UPDATE}\left(\mathbf{h}_i^{(l)}, \mathbf{a}_i^{(l)}\right) \quad (15)$$

In the implemented network, the update has a GraphSAGE-like concatenation form:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \text{CONCAT}\left(\mathbf{h}_i^{(l)}, \mathbf{a}_i^{(l)}\right)\right) \quad (16)$$

where $\sigma(\cdot)$ denotes the ReLU activation function and $\mathbf{W}^{(l)}$ is a learnable weight matrix.

The network maps the eight raw node features through a linear input projection to a 64-dimensional hidden representation, applies three edge-aware message-passing layers with BatchNorm and ReLU, and produces per-node outputs through a two-layer projection with dropout. The full network contains 85,249 trainable parameters; Table 2 lists the principal hyperparameters. The implementation also includes two switches used in the ablation study. The first removes the static TDD node features (features 5–8), and the second removes the edge conductance from the message function. These switches preserve the remaining topology and training procedure so that the contribution of each information source can be evaluated consistently.

Table 2. TDD-GNN hyperparameters.

Component	Parameter	Value
Input	Node feature dimension	8
Input	Edge feature dimension	1
Input	Prediction target	ΔT (residual)
Input Projection	Hidden dimension	64
GNN Layers	Number of layers	3
GNN Layers	Message MLP	[64, 64]
GNN Layers	Update MLP	[64, 64]
GNN Layers	Aggregation	Sum
GNN Layers	Self-loops	Yes
GNN Layers	Normalization	BatchNorm
GNN Layers	Activation	ReLU

Table 2. Cont.

Component	Parameter	Value
Output	FC dimensions	[128, 1]
Output	Dropout rate	0.2
Output	Final activation	Linear
Total	Trainable parameters	85,249

4.4. Sensitivity Analysis and Training Optimization

4.4.1. Definition of the Sensitivity Index

The sensitivity analysis measures how changes in the TDD parameters alter the predicted temperatures. It is used both to examine whether the learned input–output relationship is physically plausible and to identify the parameters for which estimation accuracy is most important.

For a parameter group $\theta \in \{C, G_{\text{ext}}, G_{\text{contact}}\}$, the raw parameter values are perturbed by a relative amount δ , and the normalized prediction change is computed as

$$S(\theta, \delta) = \frac{1}{|\delta|} \cdot \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \left\| \hat{\mathbf{T}}_{\text{perturbed}}^{(n)} - \hat{\mathbf{T}}_{\text{baseline}}^{(n)} \right\|_1 \quad (17)$$

where $\hat{\mathbf{T}}_{\text{baseline}}^{(n)}$ is the prediction for test sample n obtained with the nominal parameters and $\hat{\mathbf{T}}_{\text{perturbed}}^{(n)}$ is the corresponding prediction after perturbation. Perturbations are applied before feature normalization so that the modified inputs are transformed using the same statistics as the training data.

A finite-perturbation measure is used instead of an infinitesimal gradient because it captures the net response over the selected parameter range, including nonlinear interactions among the input features.

4.4.2. Training Protocol

The TDD-GNN is trained by minimizing the mean squared error between the predicted and reference temperature increments :

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N_s \cdot N} \sum_{i=1}^{N_s} \sum_{j=1}^N \left(\Delta \hat{T}_j^{(i)} - \Delta T_j^{(i)} \right)^2 \quad (18)$$

where N_s is the number of training samples, N is the number of modules, and $\Delta T_j^{(i)} = T_j(t+1) - T_j(t)$ is the reference increment for module j in sample i . Predicting increments rather than absolute temperatures reduces the scale of the target and anchors the reconstructed next state to the measured current temperature.

The training configuration is as follows:

- Optimizer: Adam with an initial learning rate of 10^{-3} and weight decay of 10^{-5} ;
- Batch size: 256 one-step graph samples;
- Maximum number of epochs: 200, with early stopping after 20 epochs without validation improvement;
- Learning-rate scheduler: ReduceLROnPlateau with a reduction factor of 0.5 and a patience of 10 epochs;
- Gradient clipping: maximum norm of 1.0;
- Trajectory split: 160 training, 20 validation, and 20 test trajectories; and
- Feature normalization: zero-mean, unit-variance standardization using statistics computed only from the training set.

5. Experiments and Discussion

The experiments address three questions: (1) can the unknown thermal parameters be recovered from excitation-based temperature-response data; (2) how accurately and efficiently can the TDD-GNN predict the network-wide thermal state; and (3) how much performance is retained when identified rather than ground-truth parameters are used at inference time?

5.1. Simulation Setup

A modular 6U CubeSat representation is used as the thermal-simulation test case. It comprises $N = 6$ lumped thermal modules arranged in a 2×3 grid and connected by $M = 7$ thermal contacts corresponding to adjacent mounting interfaces. Each module has an individual capacitance, power-dissipation profile, heater, and radiating surface. Figure 2 shows the simulated layout and coupling topology.

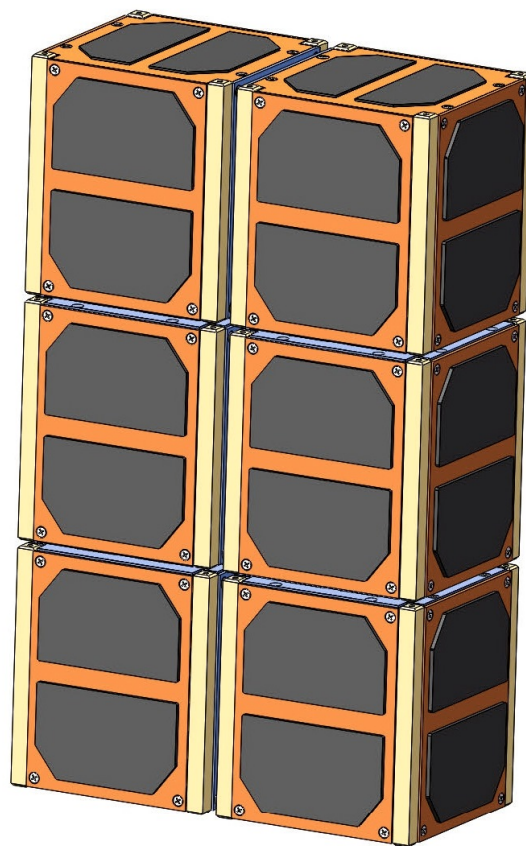


Figure 2. Simulated six-module 6U CubeSat layout and thermal-contact topology.

The trajectories are generated synthetically, but the parameter intervals are selected to be representative of CubeSat and small-satellite thermal models reported in the literature [2,29]. The sampled ranges cover module heat capacities, interface conductances, environmental heat loads, and surface properties; the complete list is given in Table 3.

Zero-mean Gaussian noise with a standard deviation of $\sigma_T = 0.10$ K is added to all simulated temperature measurements. This value is intended to represent the order of magnitude of spacecraft thermistor measurement uncertainty. The same noise level is used for all three datasets.

Table 3. Principal simulation parameters.

Parameter	Symbol	Value
Number of modules	N	6
Module layout	—	2×3 grid
Thermal contacts	M	7
Thermal capacitance	C	$[200, 1000]$ J/K
Contact conductance	G_{ij}	$[0.3, 5.0]$ W/K
External conductance	G_{ext}	$[0.001, 0.008]$ W/K
Internal power	P	$[1, 5]$ W
Sampling period	Δt	60 s
Trajectory length	T_{sim}	720 steps (12 h)
Identification trajectories	N_{id}	50 (shared params)
GNN trajectories	N_{gnn}	200 (independent params)
GNN split (train/val/test)	—	160/20/20
Deep-space temperature	T_{space}	4 K

Figure 3 illustrates how the physical module arrangement is converted into a graph. Modules become nodes, thermal contacts become edges, and the TDD quantities provide the corresponding node and edge attributes.

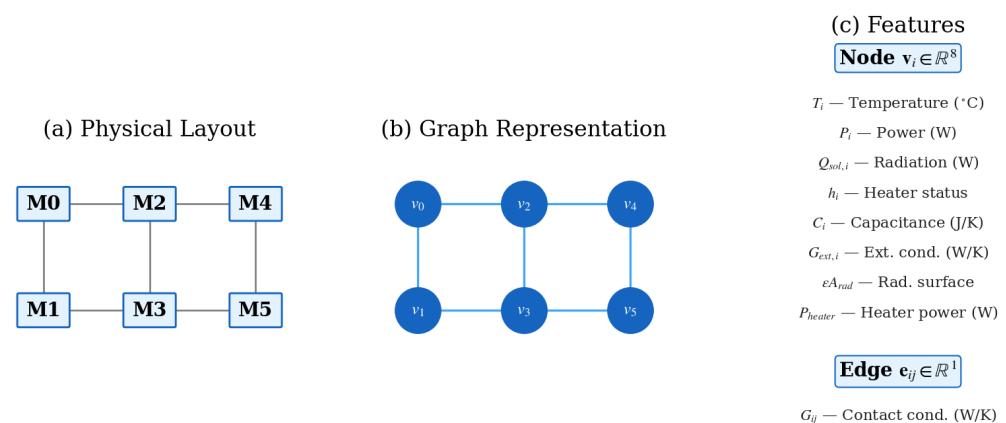


Figure 3. Construction of the modular thermal graph. (a) Physical module layout and thermal contacts. (b) Graph representation in which modules are nodes and contacts are edges. (c) Dynamic and static node features and the conductance-based edge feature.

Three datasets are generated from the same simulator for distinct evaluation purposes; their main characteristics are summarized in Table 4.

1. Dataset A (identification validation): Fifty excitation-mode trajectories are generated for one fixed parameter configuration. Each trajectory contains 720 samples with $\Delta t = 60$ s (12 h). The data are used only to evaluate shared-parameter identification (Section 5.2).
2. Dataset B (GNN development): Two hundred trajectories with independently sampled parameter sets and hysteresis-based heater control, split into 160 training, 20 validation, and 20 test trajectories (115,200/14,400/14,400 one-step pairs). Dataset B is used for model training, baseline comparison, and ablation analysis (Sections 5.3.1–5.3.3).
3. Dataset C (end-to-end identification data): For each of the 20 test configurations in Dataset B, ten additional excitation-mode trajectories are generated using the same underlying thermal parameters but independently sampled initial conditions and noise realizations. These trajectories do not overlap in time with the corresponding

hysteresis-controlled test trajectory and are used only to estimate the parameters supplied to the end-to-end evaluation (Section 5.4).

Table 4. Overview of the simulation datasets.

Dataset	Purpose	Trajectories	Parameters	Heater Mode
A	Identification validation	50	Shared	Excitation
B	GNN training/evaluation	200	Independent	Hysteresis
C	E2E deployment	20×10	Per-configuration	Excitation

Dataset A is disjoint from the GNN-development data. Dataset C shares the underlying parameter configurations of the Dataset B test set to emulate configuration-specific calibration before deployment, but its excitation trajectories, initial conditions, and noise realizations are generated separately. No Dataset C trajectory is used for network optimization.

The simulator is implemented in Python (version 3.12.13) using NumPy (version 1.26.4) for numerical integration. The neural models are implemented in PyTorch (version 2.10.0) and PyTorch Geometric (version 2.8.0). Training and inference measurements are obtained on a workstation equipped with an NVIDIA GeForce RTX 4070 GPU using CUDA (version 12.8).

5.2. Parameter Identification Validation (Dataset A)

Dataset A provides a controlled evaluation of the algebraic identification procedure for one parameter configuration observed under 50 independently initialized excitation trajectories.

The stacked algebraic system is constructed from all 50 excitation trajectories in Dataset A. Table 5 reports the resulting parameter errors.

Table 5. Parameter identification accuracy.

Parameter	MAPE	Description	Count
Thermal Capacitance (C)	16.22%	Per-module heat capacity	6
Contact Conductance (G_{ij})	16.01%	Inter-module conduction	7
External Conductance (G_{ext})	18.01%	Parasitic conductance	6
Overall	16.71%	All parameters	19

Across the 19 estimated quantities, the overall mean absolute percentage error (MAPE) is 16.71%. The capacitances and contact conductances have similar errors (16.22% and 16.01%, respectively), indicating that the pulse and differential-heating phases provide comparable information for these parameter groups. The external conductance has the largest MAPE (18.01%) because this term contributes only a small fraction of the heat balance in the simulated operating range, resulting in a lower signal-to-noise ratio.

The three columns in Table 6 resolve the apparent discrepancy noted during review. The $R^2 = 0.585$ value is a one-step, 60 s increment score produced by the explicit-Euler physics update embedded in the algebraic identifier; it was labeled as a trajectory-reconstruction metric in the original manuscript but does not correspond to a free-run simulation. When the same identified parameters are integrated over full trajectories using the generator-consistent semi-implicit scheme, the free-run absolute-temperature R^2 rises to 0.960. An oracle run with the ground-truth parameters attains $R^2 = 0.996$ (MAE = 0.29 °C), confirming that the RC structure and integrator reproduce the simulator accurately; the gap between 0.996 and 0.960 is therefore attributable to parameter-identification error rather than to model-form error. The identified heat capacities are systematically underestimated (by 7–28%), and one contact conductance saturates at the upper identification bound of

5.0 W/K; these biases produce a mean cold offset of $-0.23\text{ }^{\circ}\text{C}$ and larger residuals in eclipse (MAE = $1.42\text{ }^{\circ}\text{C}$) than in sunlit phases (MAE = $1.09\text{ }^{\circ}\text{C}$). The errors remain localized: the dominant $0\text{--}15\text{ }^{\circ}\text{C}$ operating band retains $R^2 = 0.94$, while the cold tail (-20 to $0\text{ }^{\circ}\text{C}$) carries most of the residual variance. These structured residuals are precisely what the downstream GNN is trained to absorb; the end-to-end experiment in Section 5.4 confirms that the GNN retains $R^2 = 0.959$ when fed the identified parameters. Figure 4 compares representative reconstructed and reference trajectories.

Table 6. Prediction performance of the RC model under three evaluation settings. The one-step increment R^2 is produced by the explicit-Euler update embedded in the algebraic identifier and is retained for traceability with the original manuscript; the two free-run columns integrate the model over full held-out trajectories using the generator-consistent six-substep semi-implicit scheme, with identified or ground-truth parameters. One-step MAE/RMSE are omitted because their units (K per 60 s increment) are not comparable to the absolute-temperature errors of the free-run columns.

Metric	One-Step ΔT (Identified)	Free-Run T (Identified)	Free-Run T (True Params)
MAE ($^{\circ}\text{C}$)	—	1.2070	0.2918
RMSE ($^{\circ}\text{C}$)	—	1.4538	0.4402
R^2	0.5849	0.9603	0.9964

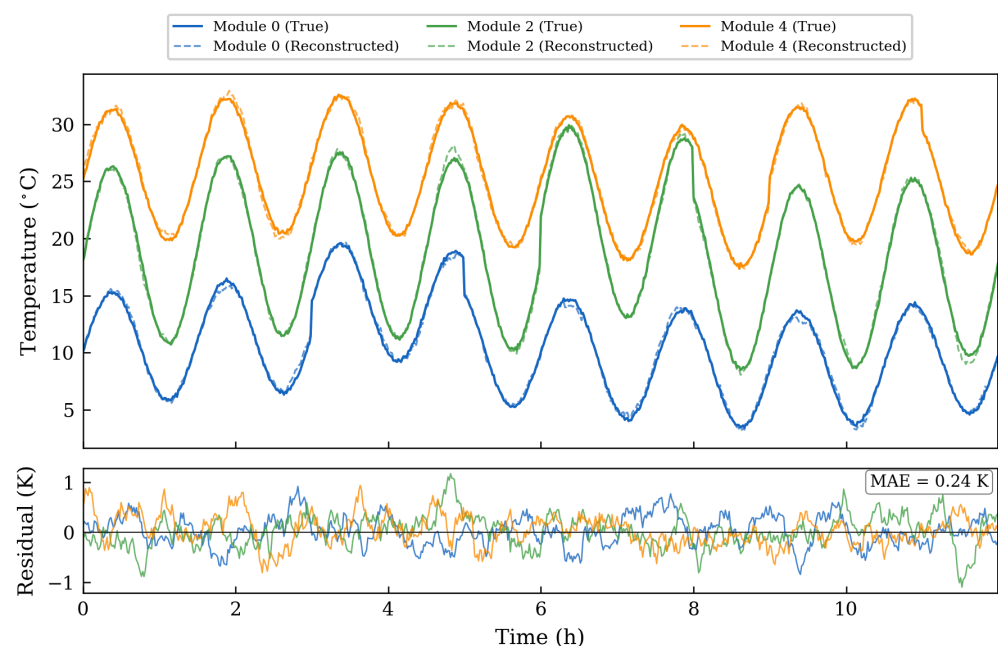


Figure 4. Representative temperature trajectories reconstructed using the identified parameters and the corresponding reference simulation.

For reference, a neural physics-informed identification procedure was also tested. A neural surrogate was trained to represent $T(t)$, and automatic differentiation was used to evaluate the derivatives in the physical residual. Under the tested training configuration, the surrogate did not reproduce the temperature trajectories adequately ($R^2 = 0.000$, MAE = 3.96 K), and the resulting parameter MAPE was 53.04%. This result is specific to the implemented surrogate and training protocol; it nevertheless illustrates the additional approximation error that can arise when an intermediate neural trajectory model is used for a problem that is directly linear in the unknown parameters.

5.3. GNN Prediction Performance (Dataset B)

The predictive model is evaluated on Dataset B using ground-truth thermal parameters as TDD features.

The network is trained using 115,200 one-step samples from 160 trajectories and evaluated on 14,400 samples from the 20 held-out test trajectories. Early stopping selects epoch 117, at which the validation loss is 0.015101. Table 7 reports the one-step residual-prediction metrics.

Table 7. One-step temperature-increment prediction performance.

Metric	Value	Unit	Description
MAE	0.0957	°C	Mean absolute error
RMSE	0.1221	°C	Root mean square error
R^2	0.9620	—	Coefficient of determination
Inference time	0.012	ms/sample	Per-sample latency

For one-step prediction of ΔT , the TDD-GNN obtains an MAE of 0.0957 °C and an R^2 of 0.9620. MAE and RMSE are reported in degrees Celsius because they retain the physical unit of the predicted quantity, which is standard practice in temperature-prediction tasks and allows direct comparison with engineering tolerances; the dimensionless R^2 is additionally provided as a scale-invariant indicator. The measured latency is 0.012 ms per graph sample on the stated workstation, which is small relative to the 60 s sampling interval.

5.3.1. Multi-Step Rollout Evaluation

Autoregressive forecasts are evaluated for horizons of 1, 10, 30, 60, and 120 steps, corresponding to 1 min through 2 h. Unlike the one-step results in Table 7, which are computed for the increment target ΔT , the rollout metrics are computed for reconstructed absolute temperatures. Table 8 summarizes the results.

Table 8. Autoregressive absolute-temperature prediction performance.

Horizon	Time Ahead	MAE (°C)	RMSE (°C)	R^2
1 step	1 min	0.1683	0.2511	0.9878
10 steps	10 min	0.1825	0.2216	0.9916
30 steps	30 min	0.1220	0.1534	0.9899
60 steps	1 h	0.1519	0.1882	0.9928
120 steps	2 h	0.1130	0.1401	0.9907

The absolute-temperature coefficient of determination remains above 0.98 for all reported horizons. The errors are not monotonic in the forecast horizon, which reflects differences in the orbital phase and temperature-variation amplitude represented by each evaluation window. Consequently, the table should not be interpreted as showing that longer forecasts are intrinsically easier. Because the results are obtained from one training run, repeated-seed experiments would be required to quantify training variability and determine whether the observed differences among horizons are statistically robust.

5.3.2. Baseline Comparison

Table 9 compares the TDD-GNN with four neural baselines and a direct RC-model baseline on the same test data. The neural models use the same train-validation-test split and optimization protocol. The RC baseline evaluates the governing thermal model using the ground-truth parameter values.

Table 9. One-step comparison with baseline methods.

Method	MAE (°C)	RMSE (°C)	R^2
TDD-GNN	0.0957	0.1221	0.9620
Physics-Based RC Model (true params)	0.1222	0.1996	0.8986
Physics-Based RC Model (identified params)	0.1203	0.1958	0.9024
GCN	0.1506	0.2020	0.8961
Vanilla GNN	0.1515	0.2049	0.8931
LSTM	0.1543	0.2113	0.8863
1D-CNN	0.1577	0.2156	0.8817

The TDD-GNN reduces the MAE by approximately 38% relative to the LSTM and by approximately 20–22% relative to the two RC variants. It also outperforms the vanilla GNN and GCN variants, which indicates that the static TDD features and edge conductance provide useful information beyond the instantaneous operating state. To ensure a fair comparison, Table 9 reports the RC model both with ground-truth parameters and with parameters identified by the algebraic procedure of Section 4.2. The identified-parameter RC performs comparably to (and marginally better than) the true-parameter RC on this test set, likely owing to compensatory fitting between the estimated parameters and the unmodeled residuals of the lumped formulation. Importantly, the TDD-GNN using identified parameters (Section 5.4, MAE = 0.0999 °C) remains substantially more accurate than either RC variant, confirming that the GNN provides predictive value beyond simply replicating the governing RC equations. Figure 5 shows the one-step residual predictions and the rollout metrics.

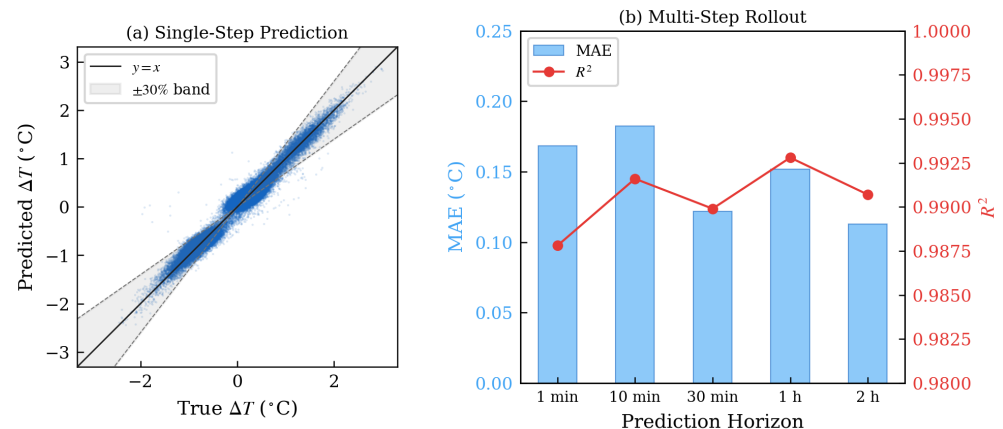


Figure 5. Prediction results for the TDD-GNN. (a) Predicted and reference one-step temperature increments on the test set. (b) MAE and R^2 for autoregressive absolute-temperature forecasts from 1 to 120 steps.

5.3.3. Ablation Study

The contribution of the TDD node features and edge attributes is examined using the two architecture switches described in Section 4.3.2. Each ablated model is trained and evaluated using the same data split and optimization settings. Table 10 reports the results.

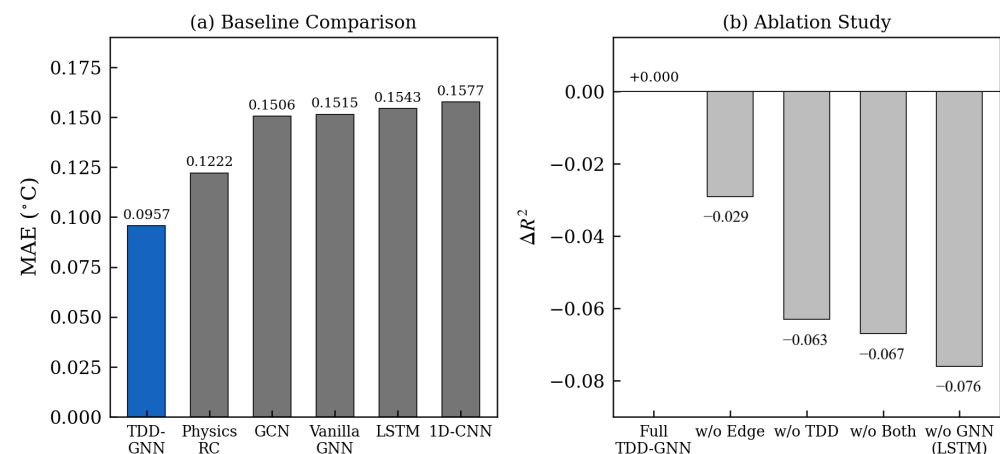
The ablation results support the following observations:

- TDD node features: Removing C , G_{ext} , $\varepsilon A_{\text{rad}}$, and P_{heater} decreases R^2 by 0.063 and increases the MAE by approximately 53%. The static thermal descriptors therefore provide substantial information about module-specific dynamics.

- Graph message passing: Replacing the graph model with the reported LSTM baseline decreases R^2 by 0.076. Under the implemented baseline configuration, explicit exchange of information between thermally connected modules is beneficial.
- Edge conductance: Removing the edge feature while retaining the TDD node features decreases R^2 by 0.029. Thus, explicit contact-conductance information improves prediction, although part of the coupling can still be inferred from the temperatures and node-level properties.
- Removal of both parameter sources: The GCN-like variant uses only temperature, power, environmental input, and heater state. Its R^2 of 0.8953 is close to those of the other unstructured or weakly structured baselines, showing that graph connectivity alone is insufficient to distinguish the sampled thermal configurations. Figure 6 visualizes the baseline and ablation comparisons.

Table 10. Ablation results for the TDD-GNN.

Configuration	MAE (°C)	RMSE (°C)	R^2	ΔR^2
TDD-GNN	0.0957	0.1221	0.9620	—
w/o Edge	0.1256	0.1628	0.9325	−0.029
w/o TDD	0.1468	0.1989	0.8992	−0.063
w/o Both	0.1511	0.2027	0.8953	−0.067
w/o GNN	0.1543	0.2113	0.8863	−0.076

**Figure 6.** Baseline and ablation comparisons. (a) One-step MAE of the proposed and baseline methods. (b) Reduction in R^2 after removing TDD node features, edge attributes, or graph message passing.

5.4. End-to-End Identification-to-Prediction Evaluation (Dataset C)

The preceding experiments evaluate parameter identification and prediction separately. The following end-to-end experiment instead supplies configuration-specific identified parameters to the trained TDD-GNN. For each test configuration, additional excitation trajectories are generated, the algebraic method estimates the corresponding TDD parameters, and these estimates replace the ground-truth parameter features during inference on the hysteresis-controlled test trajectory.

This design captures variation in identification accuracy across the sampled parameter space. For each of the 20 Dataset B test configurations, ten independent excitation trajectories of 720 steps are generated with $\sigma_T = 0.10$ K measurement noise. Identification is then performed separately for each configuration. The reported implementation requires approximately 3.7 s per configuration (73.9 s in total). Table 11 summarizes the distribution of parameter errors over the 20 configurations.

Table 11. Configuration-specific identification accuracy (mean $\pm$ standard deviation over 20 configurations).

Parameter	MAPE (Mean)	MAPE (Std)	Count
Thermal Capacitance (C)	7.35%	4.46%	6
Contact Conductance (G_{ij})	15.50%	7.32%	7
External Conductance (G_{ext})	9.31%	5.08%	6

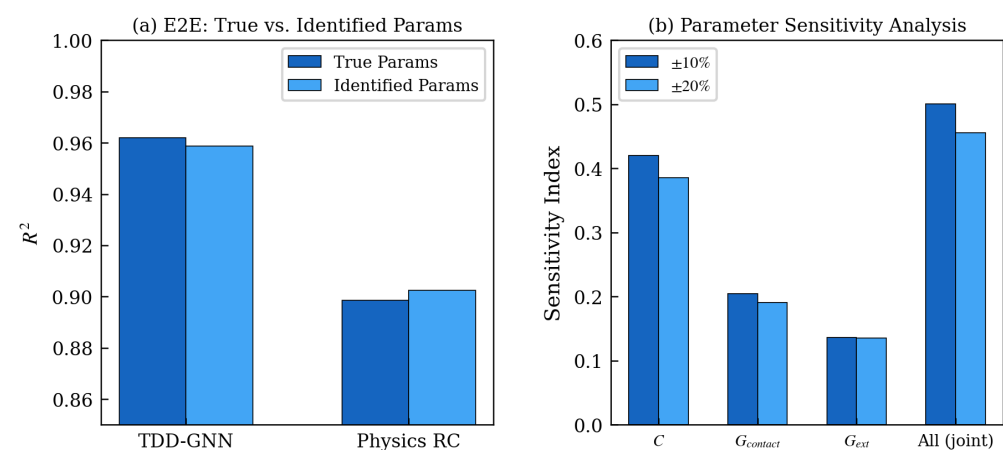
The overall MAPE across all parameters is about 10.97%, with moderate variation among configurations. This error level is somewhat lower than that observed in Dataset A, though the comparison is not direct due to different parameter realizations. The result provides the parameter-error level used in the subsequent end-to-end prediction test.

Table 12 reports the prediction performance of both TDD-GNN and the RC baseline using either ground-truth or identified parameters.

Table 12. End-to-end performance with ground-truth and identified parameters.

Model	Metric	True Params	Identified Params
TDD-GNN	MAE ($^{\circ}\text{C}$)	0.0957	0.0999
TDD-GNN	RMSE ($^{\circ}\text{C}$)	0.1221	0.1273
TDD-GNN	R^2	0.9620	0.9587
Physics-Based RC	MAE ($^{\circ}\text{C}$)	0.1222	0.1203
Physics-Based RC	RMSE ($^{\circ}\text{C}$)	0.1996	0.1958
Physics-Based RC	R^2	0.8986	0.9024

With identified TDD parameters, the GNN maintains accuracy very close to that achieved with ground-truth parameters, indicating only limited degradation under the observed mean parameter error. The RC baseline shows a slight improvement with identified parameters, possibly due to compensation between fitted parameters and unmodeled dynamics, but the TDD-GNN still outperforms the RC model in all metrics. Figure 7 summarizes the end-to-end results together with the sensitivity analysis.

**Figure 7.** End-to-end prediction and parameter sensitivity. (a) R^2 for the TDD-GNN and RC baseline using ground-truth and identified parameters. (b) Sensitivity indices for $\pm 10\%$ and $\pm 20\%$ parameter perturbations.

5.5. Parameter Sensitivity Analysis

To examine the dependence of the learned predictor on the thermal parameters, each parameter group (C, G_{ext} , and $G_{contact}$) is perturbed independently by $\pm 10\%$ and

$\pm 20\%$. The sensitivity index is computed as defined in Equation (17). Table 13 lists the resulting indices.

Table 13. Perturbation-based parameter sensitivity indices.

Parameter	$\pm 10\%$ S.I.	$\pm 20\%$ S.I.
C (thermal capacitance)	0.4205	0.3857
G_{ext} (external conductance)	0.1359	0.1356
G_{contact} (contact conductance)	0.2043	0.1911
All (joint perturbation)	0.5007	0.4556

For the tested perturbations, the inferred ordering is $C > G_{\text{contact}} > G_{\text{ext}}$:

- Thermal capacitance: C has the largest index, consistent with its direct role in thermal inertia and the temperature rate dT/dt .
- Contact conductance: G_{contact} has an intermediate index, reflecting its role in redistributing heat between connected modules.
- External conductance: G_{ext} has the smallest index, consistent with its relatively weak contribution to the short-horizon energy balance.
- Joint perturbation: The joint index is smaller than the sum of the individual indices, indicating interaction or partial cancellation among the parameter effects.

The agreement between this ranking and the qualitative structure of the RC equations provides evidence that the model responds to the TDD features in a physically plausible manner. It does not, by itself, prove that the learned dynamics satisfy the governing equations over untested conditions.

5.6. Computational Efficiency

Table 14 compares the parameter counts, inference latency, and training time of the evaluated methods.

Table 14. Computational cost comparison.

Method	Parameters	Inference (ms/Sample)	Training (s/Epoch)
TDD-GNN	85,249	0.012	19.43
GCN	85,057	0.012	20.06
Vanilla GNN	85,249	0.012	19.26
LSTM	54,337	0.003	17.51
1D-CNN	9313	0.002	16.90
RC Model	0	0.002	0

All neural models have sub-millisecond inference latency. For a 60 s sampling interval, the TDD-GNN latency is negligible at the six-node graph scale. The LSTM and 1D-CNN are faster but have higher prediction errors (Table 9).

The TDD-GNN converges in about 120 epochs, corresponding to less than 40 min of offline training on the RTX 4070. These measurements are hardware- and implementation-dependent and should not be interpreted as onboard performance benchmarks.

5.7. Robustness Under Model Mismatch

The results reported above are obtained on trajectories generated by the same lumped RC equations used for parameter identification. To examine whether the trained predictor retains accuracy when the test data deviate from the training assumptions, the frozen one-step TDD-GNN is evaluated on three deliberately perturbed test sets. Each set comprises

20 hysteresis-controlled trajectories of 720 steps generated with a random seed independent of the training data; the GNN receives the same nominal TDD features as in the standard pipeline, while the data-generating process is altered. The baseline ($\sigma_T = 0.10$ K, in-range parameters, constant G_c) yields $\text{MAE} = 0.097$ °C and $R^2 = 0.959$, consistent with the held-out test results in Table 7.

Sensor-noise stress. Measurement noise is increased from the training value of $\sigma_T = 0.10$ K to 0.30 K and 0.50 K. As Table 15 shows, MAE increases approximately linearly with the noise level, reaching 0.496 °C at $\sigma_T = 0.50$ K. Because each one-step prediction is anchored to the most recent (noisy) temperature measurement, input noise propagates directly to the predicted increment; this behavior is intrinsic to any measurement-conditioned predictor and is not specific to the proposed architecture. Even at the highest noise level, the MAE remains below the lower end of typical small-satellite heater deadbands (0.5–2 °C, Section 5.8.1), so the predictor remains usable for state estimation and look-ahead monitoring.

Parameter extrapolation. Thermal capacitances, contact conductances, and external conductances are sampled from envelopes extended by 20% beyond each endpoint of the training range (e.g., $C \in [160, 1200]$ J/K instead of [200, 1000] J/K). The MAE changes by only +0.001 °C (Table 15), indicating that the one-step predictor does not rely critically on the TDD parameter values falling within the training distribution when the current measured temperature is supplied as an input.

Structural mismatch. Each constant contact conductance is replaced by a temperature-dependent form

$$G_c(T) = G_{c,0} \left[1 + \alpha \left(\frac{T_i + T_j}{2} - T_{\text{ref}} \right) \right], \quad (19)$$

with $\alpha = 0.005$ K^{−1} and $T_{\text{ref}} = 10$ °C, producing approximately $\pm 10\%$ variation across the simulated operating range. The GNN is still given the constant $G_{c,0}$ as its edge feature. The resulting MAE increase is +0.001 °C (Table 15), confirming that mild unmodeled temperature dependence of interface conductance is well tolerated at the 60 s prediction scale.

Table 15. Robustness of the frozen one-step TDD-GNN under three model-mismatch conditions. Each condition uses 20 independent hysteresis trajectories (720 steps each). ΔMAE is relative to the in-distribution baseline. All error metrics are one-step ΔT errors in °C, consistent with Table 7.

Condition	MAE (°C)	RMSE (°C)	R^2	ΔMAE (°C)
Baseline ($\sigma_T = 0.10$ K)	0.0969	0.1235	0.9586	—
High noise, $\sigma_T = 0.30$ K	0.2839	0.3610	0.7538	+0.1870
High noise, $\sigma_T = 0.50$ K	0.4963	0.6373	0.5217	+0.3995
Parameter extrapolation ($\pm 20\%$)	0.0976	0.1246	0.9505	+0.0007
Structural mismatch, $G_c(T)$, $\alpha = 0.005$ K ^{−1}	0.0975	0.1246	0.9578	+0.0006

These results indicate that the residual formulation provides inherent robustness to parameter-distribution shift and mild structural mismatch: because the current measured state dominates the one-step temperature increment, inaccuracies in the static TDD features and small unmodeled nonlinearities produce only bounded one-step residuals. The dominant sensitivity is to sensor noise, which affects all data-driven thermal predictors equally and can be mitigated in practice through sensor selection and filtering. Stronger structural mismatch—such as phase-change materials, heat pipes, or large radiative nonlinearities—would require extension of the TDD schema and GNN architecture and is discussed in Section 5.8.4.

5.8. Discussion

5.8.1. Engineering Interpretation

The reported one-step MAE of 0.096 °C should be interpreted in the context of practical spacecraft thermal control. Typical heater switch-on/switch-off deadbands for small satellites lie in the range 0.5–2 °C, whereas radiator louvers and fluid-loop control loops commonly operate with tolerances of 1–3 °C. The prediction error of TDD-GNN is therefore approximately one to two orders of magnitude smaller than the decision thresholds against which thermal-control actions are taken, indicating that the predictor is sufficiently accurate to support state estimation, look-ahead monitoring, and model-based heater scheduling without inducing false control transitions. During autoregressive rollout, R^2 remains above 0.98 for horizons up to 2 h; although small error accumulation is unavoidable in any recursive predictor, the residual formulation anchors every step to the most recent temperature measurement, so an abrupt disturbance (e.g., an unexpected change in external heat flux) is re-tracked within one sampling interval (60 s) rather than being amplified through a long open-loop horizon.

The end-to-end experiment further shows that realistic parameter-identification errors of approximately 11–17% produce only a limited degradation of GNN accuracy (MAE from 0.0957 to 0.0999 °C, R^2 from 0.962 to 0.959). This robustness is directly relevant to operational scenarios in which module parameters are obtained from supplier datasheets or one-off calibration tests rather than from exhaustive identification campaigns. A probabilistic treatment of parameter uncertainty, for example using Bayesian neural networks or Monte-Carlo dropout, could provide calibrated prediction intervals and is identified as a direction for future work.

5.8.2. Reduced-Order Validation Platform and Model Mismatch

The RC network used in this study provides an efficient environment for algorithm development but does not reproduce the spatial detail of a high-fidelity finite-element model; it should therefore be regarded as a reduced-order validation platform. An oracle free-run with ground-truth parameters attains $R^2 = 0.996$, confirming that the lumped RC structure and semi-implicit integrator reproduce the simulator accurately and that the residual gap to $R^2 = 0.960$ with identified parameters is dominated by parameter-estimation error rather than model-form error. The end-to-end experiment confirms that the trained GNN is only weakly affected by these errors within the tested range. Nevertheless, the one-step increment $R^2 = 0.585$, the localized free-run residuals in eclipse and cold-temperature segments, and the difficulty of estimating G_{ext} indicate that parameter identifiability and model mismatch remain relevant.

A legitimate concern is that both the identification procedure and the GNN are evaluated on trajectories generated by the same governing equations, which creates a favorable evaluation scenario. Three considerations mitigate this concern. First, the end-to-end experiment deliberately re-identifies parameters from independently generated excitation trajectories before GNN inference, so that the predictor never sees the ground-truth parameter vector at test time. Second, the GNN is trained to predict one-step temperature increments rather than absolute trajectories; because each prediction is anchored to the measured current state, unmodeled dynamics manifest as a bounded one-step residual rather than as a slowly growing trajectory bias. Third, the robustness experiments in Section 5.7 explicitly inject three forms of deviation—increased sensor noise, 20% parameter extrapolation beyond the training envelope, and a temperature-dependent contact conductance not represented in the TDD edge features—and confirm that the frozen predictor degrades by less than 0.001 °C in MAE under the latter two conditions. The dominant sensitivity is to measurement noise, which is intrinsic to all data-driven predictors and is not specific to the

proposed framework. Stronger structural deviations (phase-change materials, heat pipes, large radiative nonlinearities) remain beyond the scope of the present lumped formulation and are discussed in Section 5.8.4. As one reviewer observed, validation through thermal-vacuum tests or a high-fidelity finite-element model is a natural extension rather than a prerequisite for demonstrating the methodological contribution of the present study.

5.8.3. Generalization Across Topologies

Dataset B demonstrates generalization across independently sampled thermal parameters for a fixed six-node topology. It does not yet demonstrate transfer to a different number of modules or a new connection topology. Because the message-passing layers share weights across all nodes and condition edge messages on the scalar conductance feature, the architecture is in principle agnostic to graph size and connectivity; the robustness to parameter extrapolation and structural perturbations demonstrated in Section 5.7 provides preliminary evidence that the learned message-passing weights may transfer across topologies, although explicit verification using variable-topology simulations (e.g., four- and eight-module configurations with different contact graphs) and, ultimately, physical test data remains as future work.

5.8.4. Limitations

Several limitations remain. The module topology and environmental inputs are assumed to be known; the identification uses synthetic excitation data and a fixed lumped-node model. The robustness tests in Section 5.7 cover mild temperature-dependent conductance and parameter extrapolation, but strongly nonlinear mechanisms such as phase-change materials, heat pipes, loop heat pipes, and large radiative nonlinearities are not represented. The sensitivity study is based on deterministic parameter perturbations and does not constitute a probabilistic uncertainty-propagation analysis. In addition, the reported accuracy and timing results are obtained from synthetic data on one hardware platform. Validation against high-fidelity models and thermal-vacuum measurements is therefore required before operational use can be assessed.

6. Conclusions

This paper introduced a Thermal Digital Datasheet framework for representing the passive properties, active-control capabilities, and operational constraints of modules in a modular satellite thermal system. The representation supplies a common set of node and interface quantities for parameter identification, system assembly, and data-driven temperature prediction.

A physics-constrained algebraic method was developed to estimate the module capacitances, contact conductances, and effective external conductances from excitation trajectories. The method constructs a linear least-squares system from the discretized energy-balance equations, solves it by SVD, and projects the estimates onto admissible bounds. The overall parameter MAPE is 16.71% in the shared-parameter experiment and 10.97% in the configuration-specific end-to-end experiment.

Using the TDD values as graph features, the proposed GNN achieves an MAE of 0.0957 °C and an R^2 of 0.9620 for one-step temperature-increment prediction, with R^2 remaining above 0.98 for autoregressive horizons of up to 2 h. The method reduces the one-step MAE by approximately 20% relative to the strongest baseline. Ablation results confirm that both the static TDD node features and the contact-conductance edge attributes contribute to the reported performance, and the perturbation study yields a physically plausible sensitivity ordering. When configuration-specific identified parameters are used instead of ground-truth values, the TDD-GNN retains an MAE of 0.0999 °C and an R^2 of

0.9587, supporting the feasibility of coupling the algebraic calibration procedure with the learned predictor. Deliberate mismatch tests further show that 20% parameter extrapolation and mild temperature-dependent contact conductance increase MAE by less than 0.001 °C, while the dominant sensitivity is to measurement noise as expected for any measurement-conditioned predictor.

Future work should evaluate the framework using high-fidelity finite-element models and thermal-vacuum measurements, examine transfer across module counts and connection topologies, and integrate the predictor with closed-loop thermal-control algorithms. The current lumped model does not account for strongly nonlinear heat-transfer mechanisms such as phase-change materials, heat pipes, or loop heat pipes; extending the TDD schema and the GNN architecture to accommodate such effects is a further priority. Finally, fixed uncertainty fields should be replaced by statistically calibrated parameter distributions, potentially combined with Bayesian neural-network layers to provide probabilistic prediction intervals.

Author Contributions: Conceptualization, W.P. and J.X.; Methodology, W.P. and J.X.; Software, W.P. and X.Z.; Validation, W.P. and X.Z.; Formal analysis, W.P. and X.Z.; Investigation, W.P.; Resources, W.P. and J.X.; Data curation, W.P.; Writing—original draft, W.P.; Writing—review & editing, W.P.; Visualization, W.P.; Supervision, J.Z.; Project administration, W.P. and J.Z.; Funding acquisition, W.P. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ningbo Commonweal Programme, Zhejiang Province, under Grant 2025S061.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Post, M.A.; Yan, X.T.; Letier, P. Modularity for the future in space robotics: A review. *Acta Astronaut.* **2021**, *189*, 530–547. [\[CrossRef\]](#)
2. Gilmore, D.G. *Spacecraft Thermal Control Handbook: Volume 1: Fundamental Technologies*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2002.
3. Meseguer, J.; Pérez-Grande, I.; Sanz-Andrés, Á. *Spacecraft Thermal Control*; Woodhead Publishing: Cambridge, UK, 2012.
4. Xiong, Y.; Guo, L.; Zhang, Y.; Xu, M.; Tian, D.; Li, M. Surrogate modeling for spacecraft thermophysical models using deep learning. *Neural Comput. Appl.* **2022**, *34*, 16577–16603. [\[CrossRef\]](#)
5. Gao, Z.; Qiao, K.; Bai, J.; Wang, Z.; Liu, H.; Li, P. Numerical investigation of natural circulation loop with supercritical CO₂ on the thermal control system of micro spacecrafts. *Int. J. Heat Mass Transf.* **2023**, *217*, 124661. [\[CrossRef\]](#)
6. European Cooperation for Space Standardization. *ECSS-E-ST-31C: Space Engineering—Thermal Control General Requirements*; Technical Report; European Space Agency: Paris, France, 2008.
7. Consolo, R.C.; Boetcher, S.K.S. Advances in Spacecraft Thermal Control. In *Advances in Heat Transfer*; Academic Press: Cambridge, MA, USA, 2023; Volume 56, pp. 1–50. [\[CrossRef\]](#)
8. Chen, R.T.Q.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D.K. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems 31*; NeurIPS: Sydney, Australia, 2018.
9. Xu, Y.; Yao, W.; Zheng, X.; Chen, J. A hybrid Monte Carlo quantile EMD-LSTM method for satellite in-orbit temperature prediction and data uncertainty quantification. *Expert Syst. Appl.* **2024**, *255*, 124875. [\[CrossRef\]](#)
10. Mousist, A.D. Autonomous payload thermal control. *arXiv* **2023**, arXiv:2307.15438. [\[CrossRef\]](#)
11. Glaessgen, E.; Stargel, D. The digital twin paradigm for future NASA and U.S. Air Force vehicles. In Proceedings of the Collection of Technical Papers—AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, HI, USA, 23–26 April 2012; pp. 1–14.

12. Tao, F.; Cheng, J.; Qi, Q.; Zhang, M.; Sui, F. Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3563–3576. [\[CrossRef\]](#)
13. Ren, J.; Kong, N.; Zhuang, Y.; Zhang, J.; Ma, S.; Wang, B.; Liu, W. A review on the interfaces of orbital replacement unit: Great efforts for modular spacecraft. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2021**, *235*, 1941–1967. [\[CrossRef\]](#)
14. Xie, Y.; Yao, W.; Li, X.; Wang, N.; Zheng, X.; Chen, X. Hybrid digital twin for satellite temperature field perception and attitude control. *Adv. Eng. Inform.* **2024**, *59*, 102405. [\[CrossRef\]](#)
15. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
16. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [\[CrossRef\]](#)
17. Cai, S.; Wang, Z.; Wang, S.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks for heat transfer problems. *J. Heat Transf.* **2021**, *143*, 060801. [\[CrossRef\]](#)
18. Oddiraju, M.; Hasnain, Z.; Bandyopadhyay, S.; Sunada, E.; Chowdhury, S. Physics-Informed Machine Learning Towards A Real-Time Spacecraft Thermal Simulator. In Proceedings of the AIAA Aviation 2024 Forum, Las Vegas, NV, USA, 29 July–2 August 2024. [\[CrossRef\]](#)
19. Raissi, M.; Wang, Z.; Triantafyllou, M.S.; Karniadakis, G.E. Deep learning of vortex-induced vibrations. *J. Fluid Mech.* **2020**, *861*, 119–137. [\[CrossRef\]](#)
20. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
22. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
23. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1263–1272.
24. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. Relational inductive biases, deep learning, and graph networks. *arXiv* **2018**, arXiv:1806.01261.
25. Liao, W.; Bak-Jensen, B.; Pillai, J.R.; Wang, Y.; Wang, Y. A review of graph neural networks and their applications in power systems. *J. Mod. Power Syst. Clean Energy* **2022**, *10*, 345–360. [\[CrossRef\]](#)
26. Pfaff, T.; Fortunato, M.; Sanchez-Gonzalez, A.; Battaglia, P.W. Learning mesh-based simulation with graph networks. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
27. Sanchez-Gonzalez, A.; Godwin, J.; Pfaff, T.; Ying, R.; Leskovec, J.; Battaglia, P.W. Learning to simulate complex physics with graph networks. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 8459–8468.
28. Miao, D.; Duan, G.; Chen, D.; Zhu, Y.; Zheng, X. Real-Time Temperature Prediction for Large-Scale Multi-Core Chips Based on Graph Convolutional Neural Networks. *Electronics* **2025**, *14*, 1223. [\[CrossRef\]](#)
29. Bennett, N.S.; Hawchar, A.; Cowley, A. Thermal Control of CubeSat Electronics Using Thermoelectrics. *Appl. Sci.* **2023**, *13*, 6480. [\[CrossRef\]](#)

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