Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 13 Issue 4, April - 2026

GPU-Accelerated Lattice Boltzmann Thermal Simulation of Electrical Substations with AI-Based Real-Time Prediction

Antonis Papadakis[1,2,*], Sotiris Pigiotis[1], Filotheos Zachariadis[1], Eleni Constantinide[1]
[1]KYAMOS LTD, 37 Polyneikis Street, Strovolos, 2047, Nicosia, Cyprus
[2]Frederick University, 7. Y. Frederickou, Pallouriotisas, 1036, Nicosia, Cyprus

Abstract—This paper presents a GPU-accelerated Lattice Boltzmann Method (LBM) thermal solver with Multiple Relaxation Time (MRT) collision for coupled flow and heat transfer simulation in electrical substations, integrated with a deep learning surrogate model for real-time thermal prediction. The D3Q19 MRT solver with Boussinesq buoyancy coupling, per-region material properties (the realistic AIS case uses a global thermal-relaxation approximation discussed below), and volumetric Joule heating is implemented in CUDA C++ with CUDA-aware MPI for multi-GPU execution over InfiniBand. Validation against five independent benchmarks confirms errors below 4% on all metrics, with coupled buoyancy benchmarks below 0.5%. A 40-scenario two-dimensional parametric campaign spanning natural to forced convection regimes (Richardson number from infinity to below unity) was executed. A time-aware encoder-decoder AI architecture (TUNet-AI) achieves 0.027 K mean absolute error on the 2D test set with 23× fewer parameters than a standard U-Net baseline, and 0.0042 K MAE on a 3D scaled campaign with 8-way test-time augmentation. A realistic multi-component air-insulated substation (AIS) was generated using an automated parametric STL agent, voxelised at 33 million cells, and simulated with a phenomenological thermal relaxation approximation yielding a 24.7 K peak temperature rise under this phenomenological solver configuration. The 3D model trained on realistic AIS data achieves 0.0186 K global MAE (0.022 K hotspot-masked MAE for ΔT > 1 K) with sub-200 ms inference latency, enabling real-time surrogate prediction on GPU hardware.

Keywords—Lattice Boltzmann Method; electrical substation; thermal simulation; GPU computing; deep learning surrogate; real-time prediction; parametric geometry; CUDA-aware MPI

I. INTRODUCTION

Thermal management of electrical substations is critical for grid reliability. Copper busbars and conductors carrying kA-level currents generate Joule heat ($q = J^2/\sigma$) that must dissipate through natural and forced convection. Overheating degrades insulation, accelerates component ageing, and can cause operational failure leading to unplanned outages [1]. Current monitoring typically relies on sparse point sensors (thermocouples, fibre-optic or infrared) that cannot capture the full volumetric thermal state, motivating the development of physics-based simulation coupled with AI surrogate prediction.

Computational thermal analysis of substations has traditionally employed the finite element method (FEM) and finite volume method (FVM). Recent FEM studies have modelled busbar systems under forced convection [2] and transformer thermal behaviour [3], while commercial CFD tools have been applied to enclosed switchgear cabinets [4]. However, these approaches face scalability limitations for parametric campaigns requiring hundreds of simulations across varying operating conditions.

The Lattice Boltzmann Method (LBM) [5,6] offers an attractive alternative: its explicit, local update rule maps naturally onto GPU architectures, enabling massive parallelism. GPU-accelerated LBM solvers have demonstrated strong scaling on multi-GPU clusters [7,8], and the method's kinetic formulation naturally accommodates complex geometries through simple voxelisation. For thermal problems, the double-distribution-function approach couples a hydrodynamic population set with a separate thermal population set, enabling conjugate heat transfer simulation [9].

Parallel to solver development, deep learning surrogate models have emerged as a means to accelerate parametric CFD prediction. U-Net-based architectures [10] and their variants have shown promise for temperature field prediction in engineering applications [11,12]. Physics-informed approaches and temporal conditioning strategies offer further accuracy gains by encoding physical constraints or temporal evolution patterns into the network architecture.

In this work, we present: (1) a validated GPU-accelerated LBM-MRT thermal solver for substation simulation with explicit governing equations and boundary conditions; (2) comprehensive 2D and 3D parametric campaigns with physical-unit characterisation; (3) a parametric STL geometry agent for automated substation generation; (4) a realistic multi-component AIS simulation at 33 million cells with transparent discussion of the thermal relaxation approximation and its physical limitations; and (5) a time-aware AI surrogate model with detailed training

methodology, hotspot-specific error metrics, and honest assessment of deployment requirements.

II. GPU LBM-MRT THERMAL SOLVER

A. Governing Equations

The solver employs the D3Q19 lattice with Multiple Relaxation Time (MRT) collision [6]. The MRT collision-streaming update, written in distribution form with moment-space collision, is:

$$f_i(x+e_i\Delta t,\ t+\Delta t) = f_i(x,t) - [M^{-1}S(m-m^{eq})]_i + [M^{-1}(I-S/2)\bar{F}]_i$$

where m = M·f is the moment vector obtained from the distribution functions f via the transformation matrix M, S = diag(s_0, …, s_{18}) is the diagonal relaxation matrix in moment space, m^{eq} are equilibrium moments, and $\bar{F}$ = M·F is the forcing term in moment space following Guo's scheme [13]. The relaxation rates for the stress tensor components are set according to the kinematic viscosity v via ω_v = 1/(3v_LB + 0.5), where v_LB is the lattice kinematic viscosity.

The thermal field is solved using a separate D3Q19 population set g_i with its own relaxation parameter ω_T = 1/(3α_LB + 0.5), where α_LB is the lattice thermal diffusivity. Per-region material properties are assigned: each voxel carries a material tag determining its thermal diffusivity, density, and specific heat capacity.

Buoyancy coupling uses the Boussinesq approximation, adding a body force F_y = $\rho_0 \cdot g \cdot \beta \cdot (T - T_0)$ in the vertical direction, where β is the thermal expansion coefficient and T_0 is the reference temperature. Joule heating is applied as a volumetric source term: ΔT_src = (q_vol / (ρ·C_p)) × Δt_phys, distributed equally across thermal populations in heater-region cells. For the 2D campaign, q_vol ranges from 10^3 to 10^6 W/m³ (QVOL_SCALE = 1–1000 × base rate of 10^3 W/m³).

B. Boundary Conditions

Hydrodynamic boundaries use the half-way bounce-back scheme for solid walls (no-slip) and Zou-He velocity/pressure conditions at inlets and outlets [14]. Thermal boundaries employ anti-bounce-back for fixed-temperature (Dirichlet) walls and zero-gradient extrapolation for adiabatic (Neumann) boundaries. At fluid-solid interfaces, conjugate heat transfer is handled through the double-distribution approach: solid voxels participate in the thermal lattice (with their material-specific α), but are excluded from the hydrodynamic lattice, with bounce-back enforcing no-slip at the interface.

C. Physical-Unit Mapping

The lattice-to-physical conversion follows standard LBM practice. Given the lattice spacing Δx [m], the time step is determined from the flow velocity scaling: Δt = (Δx / u_phys) × u_LB, where u_phys [m/s] is the characteristic physical velocity and u_LB [dimensionless] is its lattice counterpart. The kinematic viscosity sets the hydrodynamic relaxation via v_LB = (2τ_v − 1)/6, and the thermal diffusivity sets the thermal relaxation via α_LB = (2τ_T − 1)/6, where τ = 1/ω. All reported simulation results have been converted to physical SI units unless explicitly stated otherwise.

D. GPU Implementation

The solver is implemented in CUDA C++ [15] with one thread per lattice cell. CUDA-aware MPI enables direct GPU-to-GPU halo exchange over InfiniBand, bypassing CPU staging entirely. The solver was deployed on a cluster of 21 Tesla V100 GPUs across 4 nodes connected via Mellanox InfiniBand. MPI consistency was verified by confirming bitwise-identical results between single-rank and multi-rank execution on identical domains.

III. SOLVER VALIDATION

Five independent benchmarks were executed to validate the flow solver, thermal solver, buoyancy coupling, and multi-GPU consistency. All benchmarks were run in lattice units and converted to physical units for error computation. Grid convergence was confirmed by running each benchmark at two resolutions (coarse and fine), verifying monotonic error reduction.

Table I. Benchmark validation summary.

Benchmark	Validates	Metric	Result	Reference
Poiseuille flow	Flow profile + wall BC	NRMSE	3.4%	Analytical parabola
Thermal source 1D	Diffusion + source + MPI	L2	1.73×10^{-4}	Analytical solution
Cavity Ra = 0	Pure conduction (Nu = 1)	Nu error	0.008%	Analytical Nu = 1
Cavity Ra = 10^4	Coupled buoyancy	Nu error	0.4%	de Vahl Davis [16]
MPI consistency	Multi-rank = single-rank	Max diff	0	Bitwise identity

**NRMSE denotes the normalized root-mean-square error evaluated against the analytical Poiseuille flow profile. L2 denotes the L2-norm error computed with respect to the one-dimensional analytical diffusion solution. Nu error denotes the percentage deviation of the computed Nusselt number from the reference value.*

The Poiseuille benchmark exhibits a higher NRMSE (3.4%) due to the normalisation denominator (peak velocity); the absolute velocity errors are small. The coupled buoyancy benchmark (Ra = 10^4) reproduces the reference Nusselt number of de Vahl Davis [16] to within 0.4%, confirming correct thermal-hydrodynamic coupling. In summary, errors are below 4% across all benchmarks, with coupled buoyancy cases below 0.5%.

IV. TWO-DIMENSIONAL PARAMETRIC CAMPAIGN

A 40-scenario campaign was executed using the thin-slab approach (NZ = 3, periodic z-boundaries), sweeping 5 inlet velocities (0–0.05 in lattice units, corresponding to 0–2.5 m/s physical wind speed), 4 volumetric heat source amplifications ($q_{vol} = 10^3$–10^6 W/m³), and 2 lateral boundary conditions (no-slip wall and zero-gradient Neumann). This parameter space covers a broad nondimensional operating envelope from pure natural convection (Richardson number $Ri \to \infty$) to fully forced flow ($Ri \ll 1$), spanning Reynolds numbers up to approximately 5000 (defined for the forced-convection cases) and Rayleigh numbers from 10^4 to 10^7. Energy conservation residual across all scenarios: 0.054%.

Figure 1 shows the velocity effect on the thermal plume. At zero inlet velocity, the plume rises vertically due to buoyancy. With increasing wind speed, the plume progressively deflects, demonstrating physically correct coupling between flow and thermal solvers. The plume morphology transitions smoothly from buoyancy-dominated to shear-dominated regimes, consistent with established natural-to-forced convection transition behaviour.

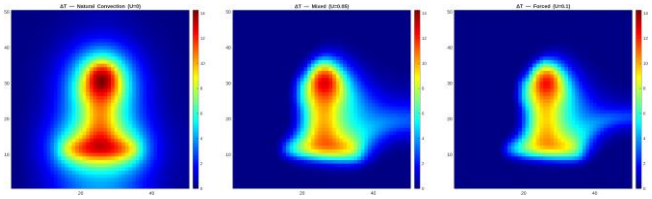

Figure 1. Velocity effect: natural convection (left), mixed (centre), forced (right).

V. REALISTIC AIS SUBSTATION SIMULATION

The substation geometry was produced by the KYAMOS parametric STL Agent, a FastAPI service built on CadQuery solid modelling with an extensible component library (busbars, breakers, clamps, transformers, jumper tubes, insulators) and three preset configurations. The agent, its component library, and the multi-scale simulation strategy it enables are described in the companion paper [17]. The AIS demo preset was used to generate a 3-phase substation section, cropped to a single bay capturing one transformer, three bus phases, one breaker bay, six clamps, and routed jumper connections. The domain (8 m × 14 m × 8 m) was voxelised at Δx = 3 cm, producing a grid of 267 × 467 × 267 = 33.3 million cells.

A. Thermal Relaxation Approximation and Physical Limitations

At the fine spatial resolution required for realistic geometry (Δx = 0.03 m), the flow-dominated time step Δt is determined by the velocity scaling and kinematic viscosity matching. For air ($\alpha_{air} = 2.2 \times 10^{-5}$ m²/s), the corresponding lattice thermal diffusivity $\alpha_{LB} = \alpha_{air} \times \Delta t/\Delta x^2$ becomes extremely small, forcing the thermal relaxation parameter $\omega_T = 1/(3\alpha_{LB} + 0.5)$ to approach its stability limit of 2.0 for all materials. At this limit, effective thermal diffusion vanishes ($\alpha_{LB} \approx 0$), rendering the thermal simulation physically meaningless.

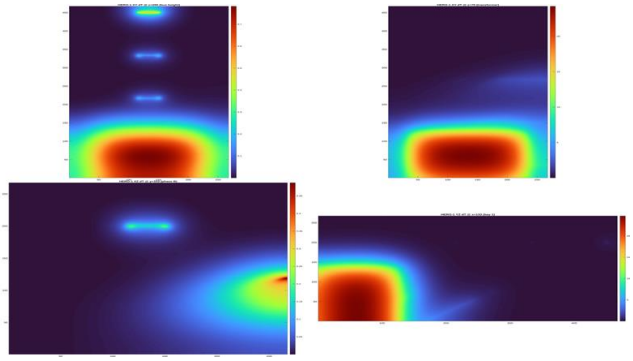

Figure 2. Voxel QA at four slice positions confirming correct resolution of all components. Axes represent lattice voxel indices.

To enable stable thermal simulation at this resolution, a global override THERMAL_OMEGA = 1.25 was applied. Since $\tau_T = 1/\omega_T = 1/1.25 = 0.8$, the corresponding uniform lattice thermal diffusivity is $\alpha_{LB} = (2\tau_T - 1)/6 = (2\times0.8 - 1)/6 = 0.1$, which is identical for all material regions. This approximation has an important physical consequence: it assigns the same effective thermal diffusivity to ambient air, steel transformer bodies, and copper busbars. It therefore cannot reproduce the large material contrast in thermal conductivity (e.g., copper versus air) nor the associated transient heat-spreading behaviour. Consequently, the simulation results are phenomenological rather than strictly quantitative: the spatial temperature patterns (plume shapes, hotspot locations, thermal interaction zones) are physically meaningful, but the absolute temperature magnitudes and transient response rates do not accurately reflect conjugate heat transfer between materials with vastly different conductivities.

Alternatives considered include thermal subcycling (executing multiple thermal steps per flow step), multi-rate relaxation schemes, and re-scaling the problem to match thermal and hydrodynamic time scales. These approaches are being investigated for future work. For the present study, the global override enables a consistent dataset for AI surrogate training, with the understanding that the trained model learns

the phenomenological thermal evolution of this specific solver configuration. The 0.59% energy budget residual confirms global energy conservation under the approximation.

B. Simulation Results

Figures 3–6 show cross-sectional temperature fields at four diagnostic planes. The simulation ran to 10,000 steps with a peak ΔT of 24.7 K (T_max = 317.9 K, domain-averaged ΔT_mean = 2.82 K). Figure 3 (XY midplane at bus height, z = 6.0 m) shows three parallel phase busbars resolved as distinct heated features with the transformer thermal dome visible below. Figure 4 (XY plane at transformer height, z = 2.1 m) reveals the 22 K hot zone at the transformer tank surface with thermal stratification extending upward. Figure 5 (XZ midplane through phase B) shows the busbar as a horizontal heated feature with a concentric thermal halo. Figure 6 (YZ midplane at bay 1) shows the transformer hot zone (25 K) at the lower left with heat rising through jumper connections toward the three-phase region above.

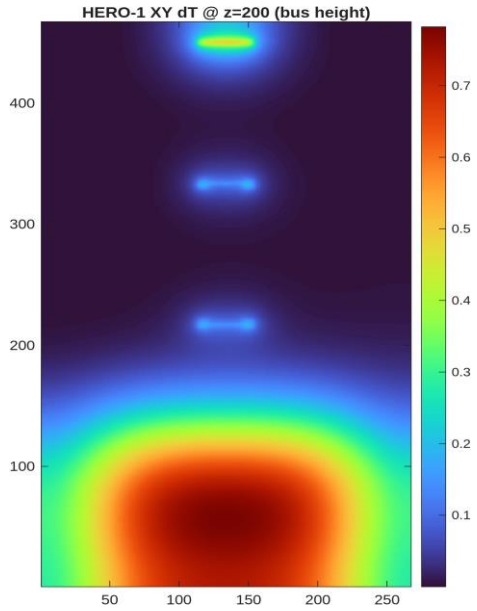

Figure 3. XY midplane at bus height (z = 6.0 m): three phase busbars with transformer dome below. Colour bar: ΔT (K). Axes: lattice voxel indices.

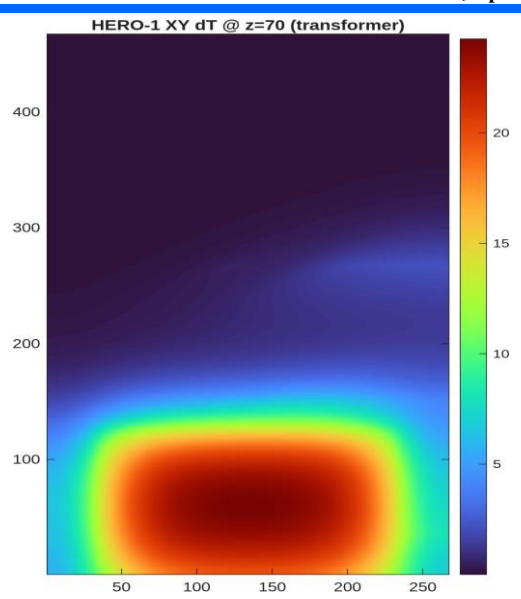

Figure 4. XY midplane at transformer height (z = 2.1 m): 22 K hot zone at transformer tank surface. Colour bar: ΔT (K). Axes: lattice voxel indices.

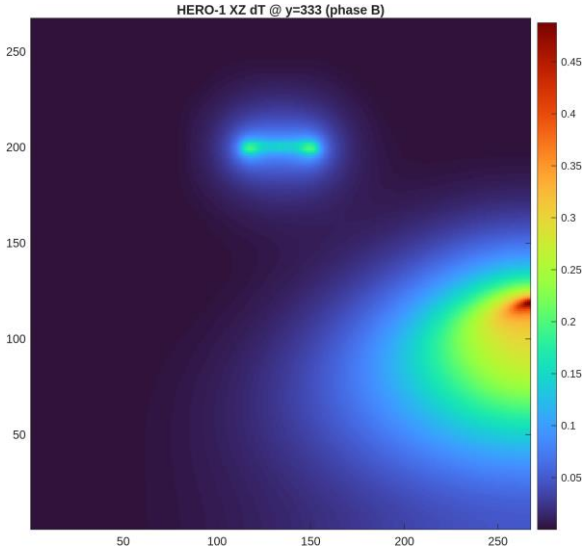

Figure 5. XZ midplane through phase B: busbar thermal halo. Colour bar: ΔT (K). Axes: lattice voxel indices.

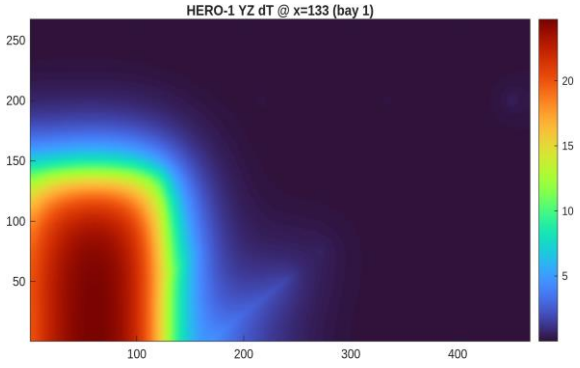

Figure 6. YZ midplane at bay 1: transformer to busbar thermal interaction. Colour bar: ΔT (K). Axes: lattice voxel indices.

VI. AI SURROGATE MODEL

A. Architecture Overview

TUNet-AI is a time-aware encoder-decoder architecture that extends the U-Net paradigm [10] with a temporal conditioning mechanism. The key innovation is the use of time-image conditioning: rather than treating temporal evolution through recurrent units (ConvLSTM) or 3D convolutions, TUNet-AI encodes temporal context by stacking four consecutive temperature-difference (ΔT) frames as input channels alongside a static volumetric heat source field. This provides the network with explicit velocity-of-change information without the computational overhead or vanishing-gradient issues of recurrent architectures. The encoder-decoder backbone uses standard convolutional blocks with skip connections, but with significantly fewer filters than the reference U-Net, resulting in 0.34M parameters (23× fewer than the 7.76M-parameter standard U-Net [10]). The architecture details beyond this high-level description are subject to a pending patent application.

B. Training Methodology

All models were trained using mean squared error (MSE) loss with the Adam optimiser (initial learning rate 10^{-3}, cosine annealing schedule). Input and output fields were normalised by the per-scenario maximum ΔT to bring all scenarios into a comparable range. For held-out test scenarios, normalisation used the maximum ΔT from the input frames (available at inference time), not from future target frames, avoiding target leakage. During autoregressive startup when all input frames are zero, a fixed operating-condition-based scale factor derived from the training set is used in place of the input-frame maximum. For the 2D campaign (40 scenarios), the data was split by scenario: 32 training, 4 validation, 4 test, ensuring no temporal leakage between splits. Each scenario contributes approximately 200 temporal snapshots (sampled every 50 solver steps during the transient approach to steady state), yielding approximately 6,400 training samples. The 3D scaled campaign (40 scenarios at 80^3 resolution) used the same 80/10/10% scenario-level split. The 3D realistic AIS campaign (20 scenarios at 128^3 subvolume crops centred on the transformer hotspot region) used a 14/3/3 scenario split. Training was performed on NVIDIA V100 GPUs: 40 epochs for 2D (approximately 2 hours), 40 epochs for 3D scaled (approximately 8 hours on 4 GPUs), and 60 epochs for 3D realistic (approximately 12 hours on 4 GPUs). Batch sizes were 16 (2D), 4 (3D scaled), and 2 (3D realistic). All reported test metrics are from the held-out test scenarios not seen during training or validation.

C. Reference Architecture

The U-Net baseline follows the original architecture of Ronneberger et al. [10] adapted for regression (replacing the final softmax with a linear output layer), with 4 encoder levels, 64 initial filters doubling at each level, and the same input/output configuration as TUNet-AI. This is a standard, widely-used architecture rather than a state-of-the-art compact design; future work will include comparisons against modern lightweight architectures (e.g., MobileNet-style encoders with depthwise separable convolutions) for a more comprehensive efficiency assessment.

D. Results and Error Analysis

Table 2. Three-level AI comparison.

**The 38.1 K maximum error occurs at voxel (1, 44, 75) in the near-boundary 2-voxel shell of scenario 029; excluding that shell reduces the maximum to 23.5 K. The validation MAE in Figure 8 (0.002 K) is from the validation split; Table 2 MAE (0.0042 K) is from the held-out test split. See Section VI-D body text for detailed analysis.*

All MAE and RMSE values presented in Table 2 were computed over the full volumetric domain. Hotspot MAE was computed only over voxels, where ground-truth ΔT > 1.0 K. The 2D and 3D tasks differ in dimensionality, domain complexity, and target distribution. Thus, their MAE values are contextual rather than directly comparable.

The global MAE values in Table 2 are computed over the entire volumetric domain, including the large fraction of ambient-air voxels where ΔT ≈ 0. Because these near-zero voxels constitute approximately 95% of the domain volume, the global MAE is dominated by the model's trivially accurate predictions in empty space. The hotspot-masked MAE (evaluated only in voxels where ground-truth ΔT > 1.0 K) provides a more operationally relevant metric. For the 3D Realistic AIS model, the hotspot MAE is 0.022 K, demonstrating excellent accuracy at heated components. However, the 3D Scaled model exhibits a hotspot MAE of 1.394 K, indicating systematic underestimation of peak temperatures in the simpler scaled geometry. Specifically, the 38.1 K maximum error in the 3D Scaled model occurs at voxel (1, 44, 75) in scenario 029 (step 1000), where the ground-truth ΔT is 38.1 K but the model predicts only 0.04 K. This voxel lies in the near-boundary 2-voxel shell (x = 1); excluding that shell reduces the maximum error to 23.5 K, indicating that prediction failures are concentrated near domain edges and at the most extreme temperature gradients. This contrast suggests that richer geometric context in the realistic domain may help the network localise hotspots more accurately, and that the scaled-domain campaign would benefit from increased training diversity or architectural modifications targeting extreme-gradient regions. Full-domain multi-subvolume inference covering the complete 33-million-cell AIS domain is left for future deployment work.

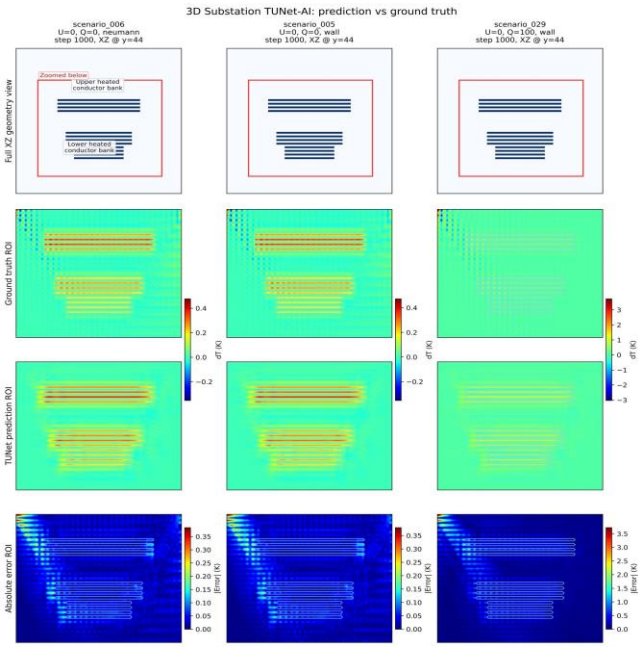

Figure 7. 3D TUNet-AI comparison: geometry/ROI view (top), ground truth and prediction (middle rows), absolute error (bottom).

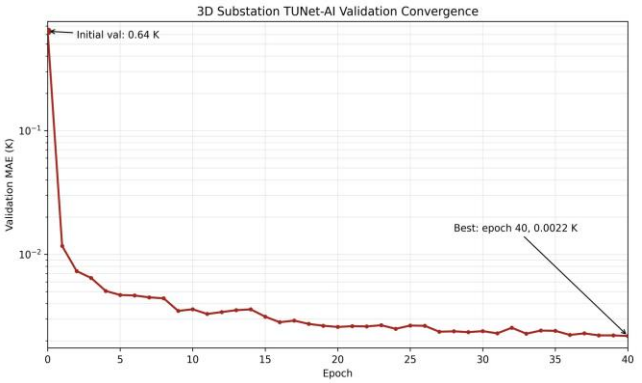

Figure 8. Training convergence (log scale): validation MAE from 0.6 K to 0.002 K over 40 epochs.

Table II. Three-level AI comparison

Model	MAE (K)	RMSE (K)	Max (K)	Hotspot MAE (K)	P99 (K)	Note
2D Champion	0.027	0.043	0.908	0.111	0.286	Baseline
3D Scaled (TTA)	0.0042	0.045	38.1*	1.394	0.046	—
3D Realistic (TTA)	0.0186	0.037	0.861	0.022	0.160	—

E. Inference Performance and Deployment Considerations

The TUNet-AI 2D champion achieves 4.6 ms inference latency on a single V100 GPU, supporting surrogate inference at over 200 Hz. The 3D realistic subvolume (128^3) requires 163.8–168.7 ms per forward pass (approximately 6 Hz). The 0.34M-parameter checkpoint (1.4 MB) is compact enough for potential edge deployment, though CPU/edge-device latency has not yet been benchmarked and remains a subject of ongoing work.

An important consideration for real-world deployment is the gap between the model's input requirements (four consecutive full-field ΔT volumes) and the data available from physical substations (sparse point sensors). In the intended deployment workflow, the model operates autoregressively: starting from an initial ambient state (ΔT = 0), it feeds its own predictions back as inputs for subsequent steps, driven by the known heat source configuration and boundary conditions. The sparse sensor data serves as periodic correction points for data assimilation, constraining drift in the autoregressive rollout. Quantitative characterisation of long-horizon autoregressive stability and the data-assimilation step is deferred to future work.

VII. CONCLUSIONS

A GPU-accelerated LBM-MRT thermal solver has been developed, validated across five benchmarks (errors below 4% overall, below 0.5% for coupled buoyancy cases), and applied to electrical substation simulation at three progressively challenging scales. A parametric STL agent service enables automated geometry generation from engineering specifications. The thermal relaxation approximation used in the 3D realistic simulation produces phenomenologically correct spatial patterns but does not capture true conjugate heat transfer between materials; addressing this limitation through thermal subcycling or multi-rate schemes is ongoing work.

The TUNet-AI surrogate model achieves 0.0042 K global MAE on idealised 3D geometry and 0.0186 K global MAE (0.022 K hotspot-masked MAE) on

realistic multi-component AIS geometry; this is lower than the 2D reference MAE, although the tasks are not directly comparable due to differences in dimensionality, domain complexity, and target distribution. The 3D Scaled model, while achieving excellent global MAE, exhibits a hotspot-masked MAE of 1.394 K, revealing that hotspot prediction accuracy depends strongly on geometric complexity and training diversity. Inference latency ranges from 4.6 ms (2D) to 168 ms (3D subvolume) on V100 GPU hardware. The complete platform, from engineering specification to surrogate thermal prediction, demonstrates TRL5 readiness (laboratory-validated integrated system), with progression to TRL6 requiring demonstration on a physical substation with real sensor feedback and validation of the autoregressive deployment pipeline.

ACKNOWLEDGMENT

This work is part of the **REALISATION-THERMAL-AI** Project (Grant number: COM-CONCEPT-ENERGY/0624/0163), funded by the EU Recovery and Resilience Facility of the European Union – NextGenerationEU, and the Republic of Cyprus through the Research and Innovation Foundation within the framework of the «Restart 2016–2020» Programmes for Research, under Component 6.1 «REPowerEU» of the Cyprus Recovery and Resilience Plan. Computations were performed on the KYAMOS V100 InfiniBand cluster.

REFERENCES

[1] IEEE Std C57.91-2011, IEEE Guide for Loading Mineral-Oil-Immersed Transformers and Step-Voltage Regulators, Institute of Electrical and Electronics Engineers, 2011.

[2] M. Bedkowski et al., "Thermal analysis of power station busbar systems," IEEE Trans. Power Deliv., vol. 31, no. 3, pp. 1116–1124, 2016.

[3] G. Swift, T. S. Molinski, and W. Lehn, "A fundamental approach to transformer thermal modeling — Part I: Theory and equivalent circuit," IEEE Trans. Power Deliv., vol. 16, no. 2, pp. 171–175, 2001.

[4] X. Zhong, C. Wang, and Y. Liu, "CFD analysis of thermal behaviour in medium-voltage switchgear cabinets under enclosed operating conditions," Appl. Therm. Eng., vol. 165, p. 114586, 2020.

[5] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford Univ. Press, 2001.

[6] P. Lallemand and L.-S. Luo, "Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability," Phys. Rev. E, vol. 61, no. 6, pp. 6546–6562, 2000.

[7] L. Grinberg et al., "Lattice Boltzmann simulations on massively parallel supercomputers," SC Conf., IEEE, 2011.

[8] M. Schönherr, K. Kucher, M. Geier, M. Stiebler, S. Freudiger, and M. Krafczyk, "Multi-thread implementations of the lattice Boltzmann method on non-uniform grids for CPUs and GPUs," Comput. Math. Appl., vol. 61, no. 12, pp. 3730–3743, 2011.

[9] A. A. Mohamad, Lattice Boltzmann Method: Fundamentals and Engineering Applications. Springer, 2011.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," MICCAI, pp. 234–241, 2015.

[11] N. Thuerey, K. Weißenow, L. Prantl, and X. Hu, "Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows," AIAA J., vol. 58, no. 1, pp. 25–36, 2020.

[12] S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," J. Fluid Mech., vol. 879, pp. 217–254, 2019.

[13] Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the LBM," Phys. Rev. E, vol. 65, p. 046308, 2002.

[14] Q. Zou and X. He, "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model," Phys. Fluids, vol. 9, no. 6, pp. 1591–1598, 1997.

[15] NVIDIA Corp., "CUDA C++ Programming Guide," docs.nvidia.com/cuda, 2024.

[16] G. de Vahl Davis, "Natural convection of air in a square cavity: A bench mark numerical solution," Int. J. Numer. Meth. Fluids, vol. 3, pp. 249–264, 1983.

[17] A. Papadakis, S. Pigiotis, F. Zachariadis, and E. Constantinide, "Parametric geometry generation and multi-scale GPU-accelerated thermal simulation of air-insulated substations," JMEST, 2026 (companion manuscript, submitted).