DC-XPIFM: AN EXPLAINABLE PHYSICS-INFORMED FOUNDATION MODEL FOR AI DATA-CENTER POWER FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Rapid power fluctuations from AI workloads challenge the short-term load forecasts used to schedule reserves and coordinate grid operation. Supporting these decisions across heterogeneous AI data centers motivates the development of generalized forecasters that combine transferable temporal representations with facility-specific physical relationships. Time-series foundation models provide the former through pretraining, but do not explicitly enforce consistency between electrical and thermal dynamics. Introducing DC-XPIFM, which, to the best of the authors' knowledge, is the first explainable physics-informed foundation model framework for short-term AI data-center power forecasting. The framework uses ordinary differential equations (ODEs) from a multi-node lumped thermal resistance-capacitance (RC) network to couple power consumption with GPU and memory temperatures, consistent with energy conservation and Newton's law of cooling. These equations regularize fine-tuning, combining patterns learned during pretraining with electro-thermal constraints informed by local measurements. DC-XPIFM is instantiated on Chronos-2, TimesFM, and Moirai and compares each backbone with its unconstrained fine-tuned variants, alongside state-of-the-art (SOTA) forecasting baselines. PI-TimesFM achieves the lowest error among the evaluated methods at every tested look-back and prediction lengths. PI-TimesFM embeddings also significantly improve the five-class workload-probe accuracy relative to vanilla TimesFM. These results show that physics-informed forecasts more accurately capture and adhere to throttle-aware power changes, abrupt drops and recovery, and post-event stability during training, fine-tuning, and online inference, behaviors essential to responsive AI data-center power management. To the best of the authors' knowledge, this is also the first Bayesian inverse-PINN method for identifying electro-thermal RC parameters from operational AI data-center data, alongside an explainability analysis that connects forecasting gains to interpretable electro-thermal dynamics and operationally informative learned representations.

1 INTRODUCTION

AI data centers are becoming a rapidly growing source of electricity demand, driven increasingly by GPU-accelerated AI and GenAI workloads. U.S. data-center electricity consumption reached approximately 176 TWh in 2023, accounting for 4.4% of national electricity use, and is projected to increase substantially as AI deployment expands Shehabi et al. (2024); International Energy Agency (2025). Beyond the growth in total consumption, the temporal variability of this demand creates challenges across several timescales. Long-term planning requires grid upgrades, siting decisions, and interconnection coordination, while day-to-day operation requires anticipating workload-driven changes in demand. Recent measurements of generative-AI training, fine-tuning, and inference workloads further demonstrate distinct power profiles whose aggregation affects whole-facility demand Vercellino et al. (2026). Consequently, understanding how much electricity AI data centers consume must be accompanied by understanding when that demand occurs and how it evolves.

Short-term power forecasting provides an actionable link between these workload dynamics and operational decisions. For grid operators, forecasts inform dispatch and reserve scheduling by providing advance estimates of demand over the relevant operating horizon. For data-center operators,

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

they can support workload scheduling, power-budget management, and coordination with on-site storage. When combined with controllable workloads and suitable operating policies, these capabilities can also help facilities participate in demand flexibility. The forecasting challenge is therefore broader than predicting an individual power trace: it is to develop models that can be adapted across heterogeneous facilities and supply useful inputs to coordinated data-center and grid operation.

Time-series foundation models offer a promising starting point by learning transferable temporal representations through large-scale pretraining. Models such as TimesFM, Moirai, and Chronos-2 support forecasting on previously unseen datasets, with Chronos-2 additionally supporting multivariate and covariate-informed prediction Das et al. (2024); Woo et al. (2024); Ansari et al. (2025). For AI data centers, this creates the possibility of beginning with a shared pretrained forecaster and adapting it to local measurements, reducing reliance on developing a dedicated model from scratch for every facility and AI workload type (training, fine-tuning, inference). At grid scale, the value of this approach lies in its potential to extend forecasting coverage across many heterogeneous sites through a common modeling framework. However, transferable statistical representations alone do not explicitly enforce the physical relationships between electrical power consumption and thermal dynamics.

Physics-informed neural networks (PINNs) provides a complementary mechanism for incorporating these relationships. PINNs augment learning from observations with residuals of governing differential equations, allowing physical knowledge to guide the learned solution Raissi et al. (2019). For AI hardware, a lumped thermal resistance–capacitance (RC) network, which was derived for this application in Saleh et al. (2026), offers a tractable representation of energy storage and heat transfer among components and their surroundings. Its governing ordinary differential equations (ODEs) couple power dissipation with GPU and memory temperatures through energy conservation and Newton's law of cooling. These relationships provide additional structure for learning from operational telemetry, although they do not by themselves describe every workload transition or cooling-control action.

Motivated by these considerations, we introduce **DC-XPiFM**, an explainable physics-informed time-series foundation model framework for short-term AI data-center power forecasting. The framework incorporates RC-based electro-thermal relationships into pretrained forecasting backbones and is instantiated on Chronos-2, TimesFM, and Moirai. We evaluate physics-informed adaptation against unconstrained fine-tuning and established supervised baselines. The study is organized around three questions: *Can a lightly adapted foundation model match a facility-specific supervised forecaster for AI data-center power prediction? Under what conditions does physics-informed adaptation improve forecasting accuracy and consistency with the governing thermal equations? Can inverse physics-informed learning recover physically plausible RC parameters? Why and how does Physics actually help the foundation models tested?* Together, these questions connect reusable forecasting with the physical calibration needed to make such models meaningful across data-center operating conditions.

Overall, the key contributions of DC-XPIFM are as follows:

- **A physics-informed foundation-model framework.** We introduce DC-XPiFM, to the best of our knowledge the first framework to embed RC electro-thermal dynamics directly into the adaptation of time-series foundation models for AI data-center power forecasting. The same formulation is instantiated on TimesFM, Chronos-2, and Moirai while preserving their original inference architecture and adding no inference-time overhead.
- **State-of-the-art forecasting across operating scales.** We benchmark vanilla and physics-informed foundation models against 16 forecasting baselines on the MIT Supercloud trace and five generative-AI workloads from the National Lab of the Rockies (NLR). PI-TimesFM achieves the lowest MAE and MAPE at every tested look-back length, reducing average MAE by 1.85%, average MAPE by 3.80%, and MAE by up to 2.46% relative to its unconstrained variant.
- **Explainable improvement of pretrained representations.** We develop a novel approach for explaining why and how the physics residual component actually impacts the embeddings, explaining the improvements in forecasting performance and physically plausible RC parameter identification. Our weight-space and embedding analysis show that the proposed RC objective reduces TimesFM's drift from its pretrained weights by approximately 51% while producing a distinct adapted solution. Physics-informed embeddings increase NLR workload balanced accuracy from 0.821 to 0.893,

raise effective rank by 30%, improve workload separability by 60%, and inference-mode recall rises by 7–22 percentage points.

- **Sensor-aware physical identification and verification.** We develop the first box-constrained Bayesian inverse PINN with a correlation-tempered likelihood and an explicit NVML measurement model for identifying physically plausible RC ODE model parameters.

2 Related Work

AI data-center power forecasting. AI workloads produce rapidly varying GPU-power profiles that shape facility demand (Vercellino et al., 2026). POLCA and TAPAS use workload information for power and thermal management (Patel et al., 2024; Stojkovic et al., 2025), while PI-DLinear adds a GPU–memory thermal residual to a locally trained power forecaster (Saleh et al., 2026). Our work instead asks whether pretrained forecasters can be adapted to local telemetry using electro-thermal constraints.

Time-series foundation models. TimesFM, Moirai, Chronos-2, and MOMENT provide pretrained backbones with different forecasting and representation-learning designs (Das et al., 2024; Woo et al., 2024; Ansari et al., 2025; Goswami et al., 2024). Energy applications have explored task-specific adaptation and physics-generated input histories (Pendyala et al., 2026; Longarini et al., 2026). We extend this direction by imposing an RC electro-thermal residual during foundation-model adaptation.

Physics-informed and inverse learning. Physics-informed learning combines observations with governing-equation residuals to estimate trajectories and unknown coefficients (Raissi et al., 2019; Braga-Neto, 2024). RC models offer a compact description of processor heat storage and transfer (Saleh et al., 2026). Because residual weighting affects optimization (Wang et al., 2021; Krishnapriyan et al., 2021), we evaluate forecasting across physics weights, horizons, models, and facilities. We also examine which electro-thermal parameters can be identified from noisy telemetry, distinguishing residual fit from parameter certainty (Simpson & Baker, 2026; Yang et al., 2021).

3 Method

3.1 Problem setting and notation

A look-back window of L samples of C telemetry channels recorded at interval Δt is written $\mathbf{X}_t = [\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t]^\top \in \mathbb{R}^{L\times C}$, and the power over the next T samples is $\mathbf{y}_t = [P_{t+1}, \ldots, P_{t+T}]^\top \in \mathbb{R}^T$. A forecaster with parameters Φ returns $\widehat{\mathbf{y}}_t$ and ϕ collects the parameters of the electro-thermal model. The MIT Supercloud trace (Samsi et al., 2021) provides $C = 5$ channels at $\Delta t = 60$ s: GPU and memory utilisation u_g, u_m (%), GPU and memory temperature T_g, T_m (°C) and power P (W). Power is summed and temperatures are averaged over the monitored V100 fleet, so the data describe one aggregate device. Its 330,508 samples are split chronologically into training, validation and test sets of 231,355, 66,101 and 33,052 samples, respectively. The NLR traces (Vercellino et al., 2026) provide, for each node, per-GPU power and temperature P_{g_i}, T_{g_i} ($i = 1, \ldots, G$, $G = 4$) and the power of two CPU packages, at $\Delta t = 0.1$ s (inference) or 0.2 s (training). Runs are assigned to the three splits (70/20/10%) in index order, by job for the training workloads and by run for the inference workloads. Since, the workloads are indexed by configuration, the resulting splits differ in composition. For both training and fine-tuning workloads, the test split contains only 16-node jobs, while the training split contains jobs of one to eight nodes (two to eight for Llama-2 LoRA). For online-rate inference, the test split contains the second repetition of request rates 81–100 req/s. The other two inference workloads span their full configuration range in every split. Hardware specifications, preprocessing, split construction, and workload power profiles are provided in Appendix E.

Every channel is standardized with the mean and standard deviation of its training split, with losses and accuracy metrics computed on standardized power. Regarding Physics residuals, they are computed after inverting the standardization, so the electro-thermal parameters keep their physical units. The target for the MIT dataset is P, and for NLR, the targets are the four per-GPU powers, with accuracy reported for the node aggregate $\sum_i P_{g_i}$. Finally, the data term is the mean squared

error (MSE) over the targets, windows, and horizon steps as

$$\mathcal{L}_{\text{data}}(\Phi) = \frac{1}{NT}\sum_{n=1}^{N}\sum_{k=1}^{T}\big(\widehat{P}_{t+k}^{(n)} - P_{t+k}^{(n)}\big)^2. \tag{1}$$

To this end, we instantiate DC-XPiFM with three pretrained forecasters, TimesFM-2.5, Chronos-2, and Moirai-1.1-R. Their architectures, forecast heads, and differentiable adaptation interfaces are described in Appendix A.

3.2 Electro-thermal prior

Two-node network (MIT). Following Saleh et al. (2026), energy balance at a GPU node and a memory node gives

$$C_g\frac{dT_g}{dt} = \alpha P - \frac{T_g - T_a}{R_{ga}} - \frac{T_g - T_m}{R_{gm}}, \tag{2}$$

$$C_m\frac{dT_m}{dt} = (1-\alpha)P - \frac{T_m - T_a}{R_{ma}} + \frac{T_g - T_m}{R_{gm}}, \tag{3}$$

where C_g and C_m are heat capacities, R_{ga} and R_{ma} are resistances to an ambient at T_a, and R_{gm} is the GPU–memory resistance. The split $\alpha \in [0,1]$ divides the single measured power between the two heat sources; without it the two balances cannot be written from one power signal. Because P is a fleet sum and T_g, T_m are fleet means, these are aggregate parameters. Nevertheless, differentiating Equations (2)–(3) and solving for the power rate gives one estimate per node as

$$\dot{P}_g^{\text{phys}} = \frac{1}{\alpha}\Big[C_g\ddot{T}_g + \Big(\frac{1}{R_{ga}} + \frac{1}{R_{gm}}\Big)\dot{T}_g - \frac{1}{R_{gm}}\dot{T}_m\Big], \tag{4}$$

$$\dot{P}_m^{\text{phys}} = \frac{1}{1-\alpha}\Big[C_m\ddot{T}_m + \Big(\frac{1}{R_{ma}} + \frac{1}{R_{gm}}\Big)\dot{T}_m - \frac{1}{R_{gm}}\dot{T}_g\Big]. \tag{5}$$

Per-GPU network (NLR). NVML takes care of every GPU together with its high-bandwidth memory, but no memory temperature was logged. Therefore, no power split was needed, so each GPU is one node described as

$$C_g\frac{dT_{g_i}}{dt} = P_{g_i} - \frac{T_{g_i} - T_a}{R_{ga}} - G_{gg}\sum_{j\neq i}\big(T_{g_i} - T_{g_j}\big), \qquad i = 1,\dots,G, \tag{6}$$

with inter-GPU conductance $G_{gg} = 1/R_{gg}$. Removing the storage term gives the static prior $P_{g_i} = (T_{g_i} - T_a)/R_{ga}$. Table 4 summarizes the two instantiations accordingly.

3.3 Physics-informed objective

On MIT, we insert forecast power into the two-node RC balances (Equations (2)–(3)). In the univariate setting, observed future temperatures and utilisation enter the residual only during training; they are neither model inputs nor needed at inference. The multivariate setting instead evaluates the residual using its own future predictions. We penalize the squared balance residual ($\mathcal{L}_{\text{th}}$), its step-to-step variation ($\mathcal{L}_{\text{rate}}$), disagreement between predicted and RC-implied power ramps ($\mathcal{L}_{\text{2nd}}$), and a heuristic high-load utilisation-consistency term ($\mathcal{L}_{\text{util}}$):

$$\mathcal{L} = \lambda_u\mathcal{L}_{\text{data}} + \lambda_r\mathcal{L}_{\text{th}} + \lambda_{\text{rate}}\mathcal{L}_{\text{rate}} + \lambda_2\mathcal{L}_{\text{2nd}} + \lambda_\theta\mathcal{L}_{\text{util}}. \tag{7}$$

On NLR, the one-node RC balance (Equation (6)) gives $\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_r\mathcal{L}_{\text{r}}$. The representation experiments use the dynamic residual, whereas the forecast panels use its static form. The residual discretisation, utilisation penalties, gate, and weight updates are specified in Appendix B.

4 Experimental Results and Discussion

We benchmark DC-XPiFM against 16 SOTA forecasting models, tuning and validating each on data cleaned of missing or corrupted entries, selected for important covariates, and Min–Max normalized

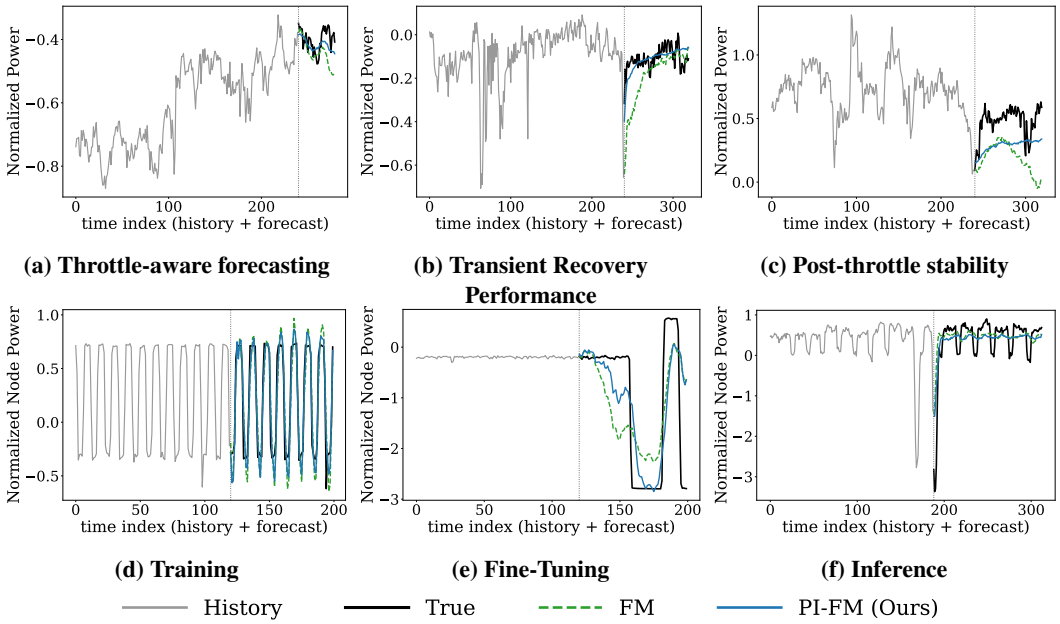


Figure 1: Comparison of FM and PI-FM across (a) power throttling, (b) transient recovery, and (c) post-event stability regimes for the MIT Supercloud dataset (first row). Second row: performance evaluation for different AI workloads, (d) training, (e) fine-tuning, (f) inference for the NLR dataset. PI-FM consistently enables accurate throttling characterization, faster recovery from different sudden AI workload changes, and stable forecasting behavior. PI reduces MAE by 37.73% during throttle-aware operation, 57.85% during transient recovery, and 31.34% during post-throttle stability on MIT, and by 30.81% for Stable Diffusion training, 29.54% for Llama-2 LoRA fine-tuning, and 3.28% for inference on NLR.

using training-set statistics. We then add physics constraints to the best-performing model and assess their effect across different input sequences and prediction lengths.

Table 1 shows that PI-TimesFM is the strongest and most consistent forecaster among all evaluated foundation models and 16 supervised baselines. It achieves the lowest MAE and MAPE at every look-back length, with MAE values of 0.1402, 0.1403, 0.1400, and 0.1413 for histories of 240, 360, 480, and 600 minutes, respectively. Relative to unconstrained TimesFM, physics-informed adaptation reduces average MAE by 1.85%, average MAPE by 3.80%, and MAE by up to 2.46%. PI-TimesFM also outperforms DLinear, the strongest conventional baseline, in both metrics across all four history lengths, demonstrating the value of combining pretrained temporal knowledge with the proposed electro-thermal constraints.

Equally important, PI-TimesFM remains highly stable as the available history changes, where its MAE varies by only 0.0013 across the entire look-back range, while MAPE remains between 1.0350% and 1.0510%. This consistency indicates that the model does not depend on a narrowly selected input length and can operate reliably across different temporal contexts and control horizons. The benefit also extends to Chronos-2, where its physics-informed variant improves MAE in three of four settings and reduces MAPE at every look-back length. Additionally, PI-Moirai reduces MAPE by approximately 1.71%, 1.80%, and 0.71% at look-back lengths of 360, 480, and 600 minutes, respectively. It also improves MAE at the 360-minute look-back, demonstrating that the electro-thermal constraints can enhance Moirai's forecasting performance across longer temporal contexts. Overall, the results establish DC-XPiFM as an accurate, robust, and backbone-compatible framework for short-term AI data-center power forecasting.

Figure 1 illustrates these forecasting gains across selected MIT and NLR operating regimes. On MIT (a–c), PI-FM more closely tracks the power-slope change under high utilisation, the sharp drop and subsequent recovery, and the post-event level and fluctuations, complementing the lower errors in Table 1. On NLR (d–f), examples from Stable Diffusion training, Llama-2 LoRA fine-tuning, and rate-controlled online inference show closer tracking of sustained demand, rapid workload-driven fluctuations, and an abrupt serving-load transition. Overall, the sub-figures illustrate how electro-thermal adaptation can improve forecast fidelity across both sustained and rapidly changing AI workloads.

4.1 EXPLAINABILITY ANALYSIS

We probe the representation immediately before the forecast head to test whether physics-informed adaptation makes operational information more accessible where the power prediction is generated. Each window yields a 1,280-dimensional TimesFM vector from its last context patch, whereas the corresponding pooled vectors for Chronos-2 and Moirai are 768-dimensional. The RC loss changes model weights, not the embedding size or its inputs. Labels come only from the future forecast interval to avoid reading the answer from the observed history. Additionally, NLR has five workload classes, while event labels (detection problem) capture power transients, rises, or drains, plus high GPU utilization on MIT or power-cap operation on NLR. Fixed probes compare Vanilla and PI embeddings using run-grouped cross-validation on NLR and blocked-time cross-validation on MIT.

On NLR, PI embeddings raise workload balanced accuracy from 0.821 ± 0.044 to 0.893 ± 0.025 (Fig. 2), recall improves from 59% to 81% for online-rate, 82% to 89% for online-finite, and 74% to 83% for offline inference. A shallow event classifier shows that PI embeddings improve balanced accuracy (0.885 to 0.893), AUROC (0.879 to 0.894), and AUPRC (0.823 to 0.867), while the smaller gap with a stronger classifier suggests event information is easier to extract. On MIT, PI improves balanced accuracy from 0.529 to 0.557, AUROC from 0.523 to 0.568, and AUPRC from 0.234 to 0.264 at 16.9% event prevalence (Table 6). A train-period-to-test-period probe confirms the ordering (0.567 to 0.593), where high utilization adds only four distinct event windows, and absolute event accuracy remains lower than on NLR.

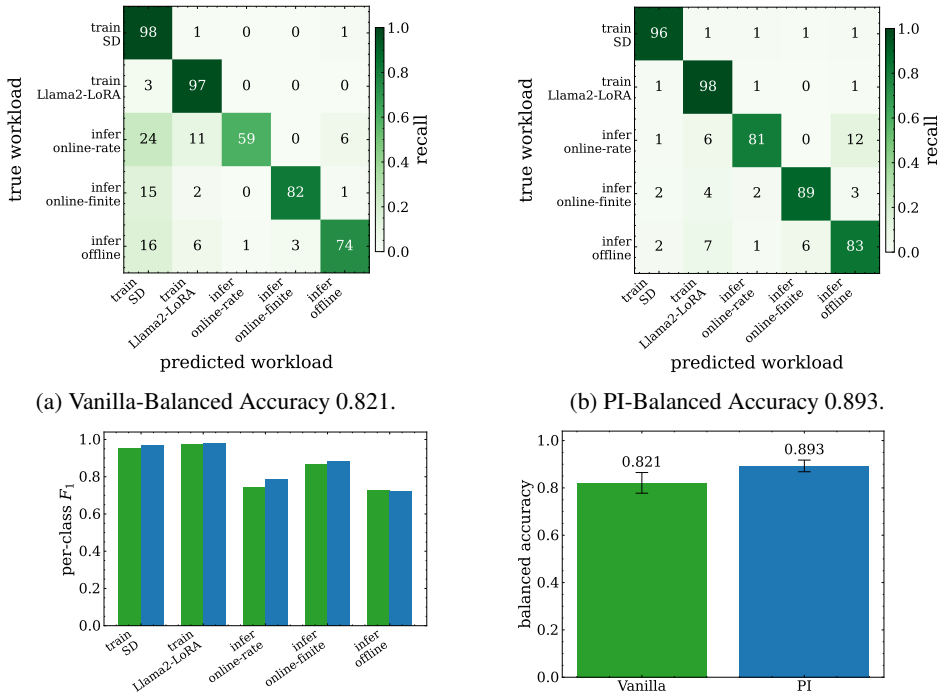
Across the model–horizon grid, physics-informed adaptation changes forecast error and embedding geometry in a backbone-dependent way. Fig. 18 shows that PI reduces average MAE by 1.78% for TimesFM and 0.79% for Chronos-2, while Moirai has 11/20 wins but a $+0.46\%$ mean change. At $L = 480, T = 20$, mean window MAE falls from 0.1340 to 0.1297, 0.1366 to 0.1334, and 0.1390 to 0.1368, respectively. The shared, standardized PCA–t-SNE projection of zero-shot, Vanilla, and PI embeddings shows diffuse changes for TimesFM and larger, architecture-specific relocations for Chronos-2 and Moirai; cooling, steady, and heating windows nevertheless remain strongly intermixed. Figs. 3 and 17c show that PI increases first-principal-component variance in all nine model–horizon comparisons and reduces MIT effective rank from 3.17 to 2.09 (TimesFM), 2.26 to 2.01 (Chronos-2), and 1.36 to 1.24 (Moirai); Moirai's active coordinates decrease from 134/768 to 118/768. Thus, PI concentrates MIT embeddings into fewer dominant directions. As Figs. 17a and 17b further show, PI improves 59% of 3,256 paired TimesFM windows and lowers mean per-window MAE by 0.0043, although displacement magnitude is only weakly associated with error change (Spearman $\rho = -0.071$): the directions of change matter more than their magnitude.

The MIT–NLR comparison shows that PI's representation changes depend on the data. On MIT, thermal-regime separability rises only 4% (Fisher ratio 0.00105 to 0.00109) while effective rank

Table 1: Average performance across prediction horizons $T \in \{5, 10, 20, 40, 80\}$ min for forecasting under different history lengths ($L \in \{240, 360, 480, 600\}$ min) for the proposed and baseline models.

	$T = 240$ min		$T = 360$ min		$T = 480$ min		$T = 600$ min	
Model	**MAE**	**MAPE**	**MAE**	**MAPE**	**MAE**	**MAPE**	**MAE**	**MAPE**
iTransformer	0.1481	1.1323	0.1527	1.1603	0.1533	1.1537	0.1555	1.2052
Transformer	0.1670	1.2211	0.1726	1.4009	0.1754	1.3291	0.1785	1.3318
TimeXer	0.1474	1.1021	0.1631	1.2364	0.1511	1.1139	0.1636	1.2195
TiDE	0.1422	1.0628	0.1436	1.0743	0.1441	1.0708	0.1451	1.0860
TSMixer	0.1507	1.0999	0.1547	1.1798	0.1549	1.1491	0.1640	1.2085
Reformer	0.1739	1.2483	0.1730	1.3543	0.1693	1.2455	0.1860	1.5094
PatchTST	0.1495	1.1203	0.1520	1.1443	0.1522	1.1335	0.1558	1.1635
Nonstationary Transformer	0.1489	1.1044	0.1505	1.1388	0.1527	1.1212	0.1580	1.2052
LightTS	0.1496	1.1490	0.1526	1.1687	0.1539	1.1695	0.1563	1.1749
FiLM	0.1432	1.0708	0.1437	1.0810	0.1440	1.0840	0.1440	1.0877
FEDformer	0.2164	1.7254	0.2747	2.3160	0.3028	2.5071	0.2741	2.2863
Pyraformer	0.1498	1.2292	0.1531	1.2034	0.1555	1.1664	0.1710	1.5822
DLinear	0.1420	1.0403	0.1426	1.0484	0.1434	1.0516	0.1439	1.0646
Crossformer	0.1491	1.1437	0.1833	2.3161	0.1504	1.1648	0.1535	1.1889
NLinear	0.1673	2.0779	0.1460	1.1018	0.1472	1.1004	0.1480	1.1144
Linear	0.1463	1.0771	0.1468	1.0863	0.1479	1.0834	0.1482	1.0906
TimesFM	0.1435	1.0680	0.1413	1.0564	0.1432	1.0916	0.1448	1.0855
Chronos-2	0.1451	1.0920	0.1469	1.1042	0.1455	1.1012	0.1474	1.1151
MOIRAI	0.1425	1.0731	0.1455	1.1166	0.1462	1.1288	0.1465	1.1215
PI-MOIRAI (Ours)	0.1437	1.0761	0.1451	1.0975	0.1467	1.1085	0.1467	1.1135
PI-Chronos-2 (Ours)	0.1427	1.0533	0.1445	1.0755	0.1459	1.0752	0.1458	1.0846
PI-TimesFM (Ours)	**0.1402**	**1.0350**	**0.1403**	**1.0440**	**0.1400**	**1.0510**	**0.1413**	**1.0402**

Note: **Bold** indicates the best (lowest) value and underlined indicates the second-best (lowest) value.



(a) Vanilla-Balanced Accuracy 0.821.

(b) PI-Balanced Accuracy 0.893.

(c) Workload Per Class (F1-score).

(d) Overall Balanced Accuracy.

Figure 2: **Five-class NLR workload identification from TimesFM forecast embeddings.** A class-balanced random forest (300 trees, depth of 8) is trained on the embedding $\mathbf{z}$. Evaluation uses 5-fold cross-validation grouped by run, with out-of-fold predictions pooled. (a) Vanilla and (b) PI confusion matrices, with each cell giving the recall (%). (c) Per-class F_1 for Vanilla (green) and PI (blue). (d) Balanced accuracy (error bars are the standard deviation over the five folds).

falls by 34%, whereas on per-GPU NLR, effective rank rises 30% (6.98 to 9.05) and five-class workload separability rises 60% (0.0413 to 0.0661, Fig. 5). These changes vary by workload, where online-rate inference gains the most rank, while event separability improves for Llama-2 LoRA and online-finite inference but declines for Stable Diffusion and offline inference (Fig. 16). Moreover, the selected PI-TimesFM checkpoint also lies closer to the pretrained weights than Vanilla (drift 0.00189 versus 0.00383) across attention, MLP, and head modules, with smaller spectral changes, yet remains distinct from Vanilla (distance 0.00319; Fig. 4). These weight differences produce a 36% mean relative change in the unchanged 1,280-dimensional forecast-head embedding, alongside higher NLR workload-probe accuracy (0.821 to 0.893). Overall, the figures support a data-dependent regularization mechanism in which physics preserves the pretrained structure at the parameter level while making physically relevant operating information more accessible in the embedding.

4.2 Inverse PINNs: Physical Parameter Identification and Verification

Forecasting gains alone cannot show that the PI term enforces electro-thermal dynamics; physically admissible parameter recovery and improved forward simulation instead test whether the model

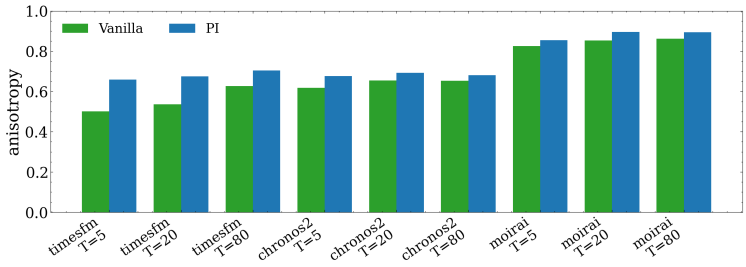
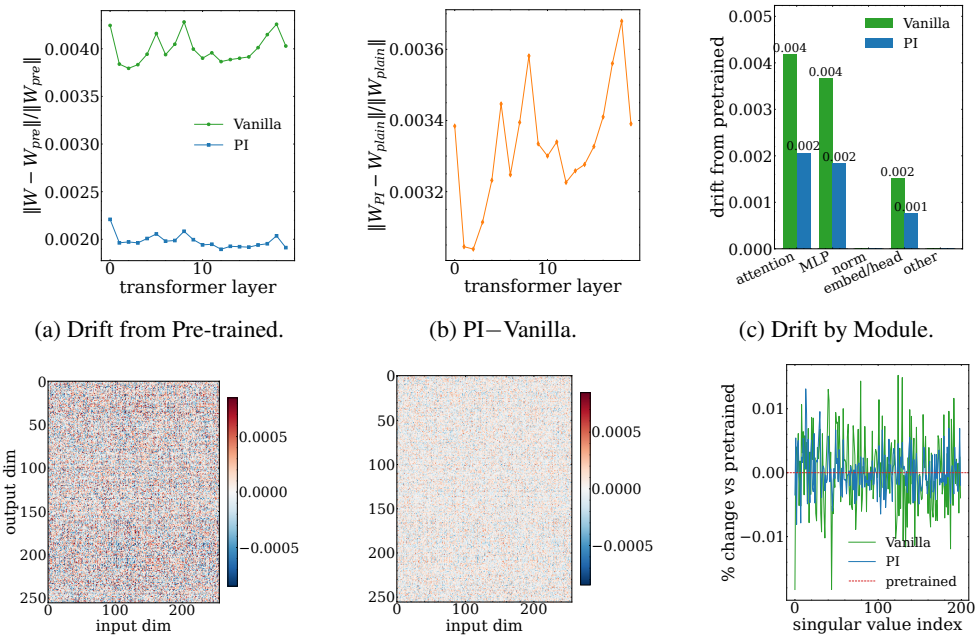


Figure 3: **Effect of physics-informed training on the tested FM models.** PI increases anisotropy in all nine settings, while its effects on compactness and weight drift depend on the model and horizon.



(a) Drift from Pre-trained. (b) PI−Vanilla. (c) Drift by Module.

(d) ΔW (Vanilla). (e) ΔW (PI). (f) Spectral Change.

Figure 4: **Effect of physics-informed training on TimesFM weights.** (a) Relative Frobenius-norm drift from the pretrained weights in each transformer layer. (b) Relative layer-wise distance between the PI and Vanilla weights. (c) Drift from pretrained weights grouped by module type. (d,e) Signed changes, $\Delta W = W_{\text{variant}} - W_{\text{pre}}$, in the same representative middle-layer weight matrix. PI makes smaller updates across layers and modules, while retaining a distinct learned weight configuration. This is consistent with the physics loss guiding more targeted updates to power–thermal behavior.

obeys GPU–memory–CPU energy balances. This matters to grid operators because credible thermal resistances and capacitances convert load forecasts into heating, cooling, and recovery time scales for safe workload shifting, power capping, and demand-response scheduling. Accordingly, we formulate a box-constrained Bayesian inverse-PINN, imposing the RC equations in weak integral form on measured power and GPU/HBM temperature trajectories to avoid unreliable derivatives of

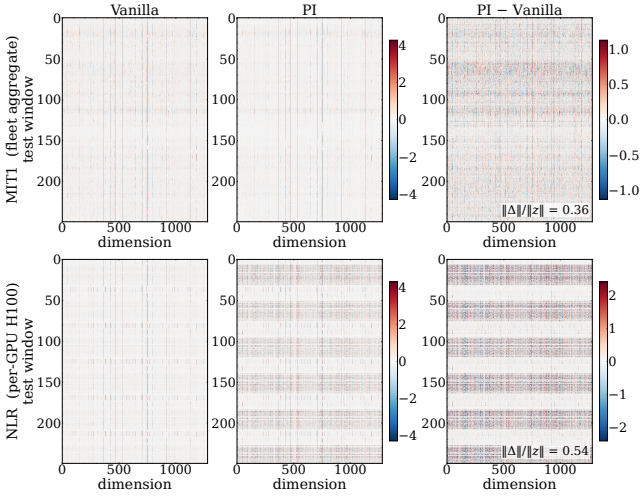


Figure 5: **Physics-informed changes in TimesFM embeddings across the MIT (top row) and NLR (bottom row) datasets.** The three columns show Vanilla embeddings, PI embeddings, and their paired difference, $\Delta \mathbf{z} = \mathbf{z}_{\text{PI}} - \mathbf{z}_{\text{Vanilla}}$. Each heatmap displays the first 250 matched test windows (vertical axis) across 1,280 embedding dimensions (horizontal axis). The annotated mean relative changes over all paired windows, $\langle \|\Delta \mathbf{z}\|_2 / \|\mathbf{z}_{\text{Vanilla}}\|_2 \rangle$, are 0.36 for MIT and 0.54 for NLR, indicating a larger PI-induced representation shift in the per-GPU data.

one-minute, quantized temperature measurements. We estimate the time-varying GPU power fraction, $\alpha(t)$, from GPU and memory utilization, then hold it fixed to remove power-partition ambiguity. For MIT (V100) and NLR (H100), the bounds are outlined in Table 7.

Against this physically bounded search space, for the MIT Supercloud dataset, Figs. 19 and 20 shows that the corrected physics objective decreases from 0.498 to 0.0297, corresponding to a 94.0% reduction, while the GPU- and memory-node residuals fall from 0.809 to 0.0473 and from 0.187 to 0.0121, respectively. Furthermore, although temperature error was excluded from the optimization objective and evaluated only as an independent forward-model diagnostic, the one-step MSE decreases from 41.61 to 4.83 K^2 and the four-hour rollout MSE from 126.01 to 22.72 K^2, thus, enforcing the energy balances improves out-of-sample thermal consistency rather than merely minimizing an auxiliary penalty. Additionally, as shown in Figs. 19 (a)–(e), the KL-regularized solution converges to $C_g = 8.91 \times 10^4$ $\mathrm{J\,K^{-1}}$, $C_m = 1.00 \times 10^4$ $\mathrm{J\,K^{-1}}$, $R_{ga} = 1.336 \times 10^{-3}$, $R_{ma} = 2.216 \times 10^{-3}$, and $R_{gm} = 1.056 \times 10^{-3}$ $\mathrm{K\,W^{-1}}$, corresponding to GPU–air and memory–air time constants of approximately 119 s and 22 s. Complementing this ablation, Fig. 21 is a conditional-identifiability stress test rather than another convergence plot. Specifically, each entry represents the normalized 95% posterior width ρ, where $\rho \approx 0$ indicates that the observations localize a parameter to a small part of its admissible range, and $\rho \approx 1$ indicates that almost the entire range remains plausible, so the columns then repeat this test while progressively treating up to four other parameters as independently known. The persistent $\rho \approx 0.89$–0.91 for C_g and C_m demonstrates that the present dataset cannot identify thermal storage even after the remaining coefficients are supplied, whereas the smaller all-free widths for R_{ga}, R_{ma}, and R_{gm} ($\rho \approx 0.11$, 0.08, and 0.28, respectively) show that heat-transfer pathways leave a stronger observable signature, which should be the case here as over 95% of the heat is dissipated and not stored in the components.

On NLR, the Bayesian inverse PINN fits the shared-loop thermal model $T_a(t) = T_{\mathrm{in}} + R_{\mathrm{loop}} \sum_j P_j(t)$ with normalized physics-residual RMS values of 5.5%, 6.1%, and 3.4% for Stable Diffusion, Llama-2 LoRA, and inference, respectively. The fitted GPU-to-coolant resistances are $R_{ga} = 0.0403$, 0.0473, and $0.0597\,\mathrm{K\,W^{-1}}$. The corresponding effective inlet temperatures are $T_{\mathrm{in}} = 40.0$, 36.8, and $26.9\,^{\circ}\mathrm{C}$, while the fitted loop resistances are $R_{\mathrm{loop}} = 1.22 \times 10^{-4}$, 1.32×10^{-4}, and $4.05 \times 10^{-4}\,\mathrm{K\,W^{-1}}$. Furthermore, the inter-GPU conductance is fitted near zero ($G_{gg} = 0.0057$, 0.0024, and $0.0016\,\mathrm{W\,K^{-1}}$), so the data do not independently establish direct thermal coupling. The capacitances $C_g = 4.81$, 3.64, and $0.79\,\mathrm{J\,K^{-1}}$ imply time constants $\tau = R_{ga} C_g$ of 0.194, 0.172, and $0.047\,\mathrm{s}$. Since these timescales are at or below the sampling interval and depend strongly on the assumed one-second NVML power-averaging model, C_g and τ are interpreted as effective model parameters.

5 Conclusion

This paper introduced DC-XPIFM, an explainable physics-informed foundation-model framework for short-term AI data-center power forecasting. DC-XPIFM embeds electro-thermal RC dynamics directly into the adaptation of TimesFM, Chronos-2, and Moirai without modifying their inference architecture. The framework was evaluated on the fleet-level MIT Supercloud trace and per-GPU H100 NLR telemetry spanning training, fine-tuning, and three inference workloads, providing a broad assessment across hardware, operating regimes, and temporal scales. The results demonstrate clear forecasting benefits, where PI-TimesFM achieved the lowest MAE and MAPE among all evaluated models at every tested look-back length, reducing average MAE by 1.85%, average MAPE by 3.80%, and MAE by up to 2.46% relative to unconstrained TimesFM. Physics also improves the TimesFM and Chronos-2 configurations, while PI-Moirai reduces MAPE across the longer look-back settings. These numerical gains translate into more robust tracking of rapid power changes, transient recovery, post-event stability, and AI workload-specific behavior across training and inference. The explainability analysis further shows that physics-informed adaptation reduces TimesFM's parameter drift by approximately 51%, increases NLR embedding effective rank by 30% and workload separability by 60%, and raises AI workload-classification balanced accuracy from 0.821 to 0.893. The inverse analysis provides complementary physical verification. Bayesian inverse PINN reduced the MIT physics objective by 94%, the one-step thermal MSE by 88%, and the four-hour rollout MSE by 82%. Overall, these findings establish DC-XPIFM as an accurate, physically robust, and explainable forecasting framework capable of supporting AI workload scheduling, power capping, reserve planning, and grid-aware operation of increasingly dynamic AI data centers.

Ethics statement

This work involves no human subjects and no new data collection. It uses two public telemetry datasets, the MIT Supercloud trace (Samsi et al., 2021) and the NLR GenAI power profiles (Vercellino et al., 2026), which contain hardware power, temperature, and utilization readings without user identities. The data and the pretrained foundation models are used under their published licenses. The intended application is short-term power forecasting for AI data-center operation and grid coordination, where a model trusted beyond its validated range could degrade reserve scheduling or power-capping decisions.

Reproducibility statement

Both datasets are public (Samsi et al., 2021; Vercellino et al., 2026). Section 3.1 and Appendix E describe the preprocessing, chronological and run-wise splits, and windowing, and Table 2 lists the configuration of each pretrained backbone. The physics residuals, loss terms and weighting schemes are specified in Sections 3.2–3.3, with the utilization term, the adapter forward passes, and the weight trajectories in Appendices B and F. For the identification of the electro-thermal parameters, Appendix G and Table 7 provide the physically admissible ranges, the power-sensor model, and the Bayesian inverse PINN. Appendix G gives all optimization settings and the sensor diagnostics. The probing protocol and its controls are described in Section F. The anonymized supplementary material contains the code accordingly.

References

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting, 2025. URL https://arxiv.org/abs/2510.15821.

P. G. Bissiri, C. C. Holmes, and S. G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016. doi: https://doi.org/10.1111/rssb.12158. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssb.12158.

Ulisses Braga-Neto. Physics-informed machine learning. In *Fundamentals of Pattern Recognition and Machine Learning*, pp. 293–324. Springer, 2024.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models, 2024. URL https://arxiv.org/abs/2402.03885.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pp. 2733–2743, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv*, 2021. doi: 10.48550/arXiv.2106.09685.

International Energy Agency. Energy and AI. Technical report, International Energy Agency (IEA), Paris, France, 2025. URL https://www.iea.org/reports/energy-and-ai/executive-summary.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

Lorenzo Longarini, Alessandro Rongoni, Simone Silenzi, Emanuele Frontoni, and Riccardo Rosati. Time series foundation models based on physics-informed synthetic histories for cold-start photovoltaic forecasting. *arXiv preprint arXiv:2606.07457*, 2026.

Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=aEFaE0W5pAd.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Íñigo Goiri, Brijesh Warrier, Nithish Mahalingam, and Ricardo Bianchini. Characterizing power management opportunities for llms in the cloud. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 207–222, 2024.

Varsha Pendyala, Yiwei Fu, Weizhong Yan, and Nurali Virani. Assessing covariate-informed grid load forecasting with a time-series foundation model. *arXiv preprint arXiv:2609.06656*, 2026.

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

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shaden Yang, Minjia Zhang, Dong Li, and Yuxiong He. ZeRO-Offload: Democratizing Billion-Scale Model Training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pp. 551–564, 2021. URL https://www.usenix.org/conference/atc21/presentation/ren-jie.

Mohammad AlShaikh Saleh, Sanjay Chawla, Sertac Bayhan, Haitham Abu-Rub, and Ali Ghrayeb. A physics-aware framework for short-term gpu power forecasting of ai data centers. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, pp. 4115–4126, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3818033. URL https://doi.org/10.1145/3770855.3818033.

Siddharth Samsi, Matthew L Weiss, David Bestor, Baolin Li, Michael Jones, Albert Reuther, Daniel Edelman, William Arcand, Chansup Byun, John Holodnack, Matthew Hubbell, Jeremy Kepner, Anna Klein, Joseph McDonald, Adam Michaleas, Peter Michaleas, Lauren Milechin, Julia Mullen, Charles Yee, Benjamin Price, Andrew Prout, Antonio Rosa, Allan Vanterpool, Lindsey McEvoy, Anson Cheng, Devesh Tiwari, and Vijay Gadepally. The mit supercloud dataset, 2021. URL https://arxiv.org/abs/2108.02037.

Arman Shehabi, Sarah J. Smith, Abigail Hubbard, Anna Newkirk, Nuoa Lei, Md Abu Bakar Siddik, Brian Holecek, Jonathan G. Koomey, Eric R. Masanet, and Dale A. Sartor. 2024 United States Data Center Energy Usage Report. Technical report, Lawrence Berkeley National Laboratory (LBNL), 2024.

Matthew J Simpson and Ruth E Baker. Parameter identifiability, parameter estimation, and model prediction for differential equation models. *SIAM Review*, 68(1):153–171, 2026.

Jovan Stojkovic, Chaojie Zhang, Íñigo Goiri, Esha Choukse, Haoran Qiu, Rodrigo Fonseca, Josep Torrellas, and Ricardo Bianchini. Tapas: Thermal-and power-aware scheduling for llm inference in cloud platforms. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 1266–1281, 2025.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors, 2022. URL https://arxiv.org/abs/2204.06507.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains, 2020. URL https://arxiv.org/abs/2006.10739.

Roberto Vercellino, Jared Willard, Gustavo Campos, Weslley da Silva Pereira, Olivia Hull, Matthew Selensky, and Juliane Mueller. Measurement of generative ai workload power profiles for whole-facility data center infrastructure planning, 2026. URL https://arxiv.org/abs/2604.07345.

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

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.

Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

Zeyu Yang, Karel Adamek, and Wesley Armour. Accurate and convenient energy measurements for gpus: A detailed study of nvidia gpu's built-in power sensor. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–17. IEEE, November 2024. doi: 10.1109/sc41406.2024.00028. URL http://dx.doi.org/10.1109/SC41406.2024.00028.

A FOUNDATION-MODEL BACKBONES

In this paper, TimesFM-2.5 (Das et al., 2024), Chronos-2 (Ansari et al., 2025) and Moirai-1.1-R (Woo et al., 2024) are leveraged and configured as in Table 2. Each model normalizes an input series $\mathbf{s} \in \mathbb{R}^L$ with statistics of the look-back ($\mathcal{N}$), splits it into non-overlapping patches of length ρ, encodes the patch tokens with a transformer, and decodes the horizon with a head g_Φ applied to a set $\mathcal{F}$ of token positions as

$$\mathbf{H} = \mathrm{Enc}_\Phi\big(\mathrm{Patch}_\rho(\mathcal{N}(\mathbf{s}))\big) \in \mathbb{R}^{M\times d}, \qquad \widehat{\mathbf{y}} = \mathcal{N}^{-1}\big(g_\Phi(\mathbf{H}_\mathcal{F})\big)_{1:T}, \qquad \mathbf{z} = \frac{1}{|\mathcal{F}|}\sum_{m\in\mathcal{F}} \mathbf{H}_m \in \mathbb{R}^d. \tag{8}$$

The vector $\mathbf{z}$, the mean of the tokens read by the head, is the representation analysed in Sec. F. The point forecast is TimesFM's point head, the median of Chronos-2's 21 quantiles, or the mean of Moirai's four-component mixture. The libraries' inference routines disable gradients, so each forward pass is made differentiable. Chronos-2's native forward pass already supports gradients and is used directly. TimesFM's decoding is re-implemented and reproduces the library output to within 9×10^{-8} in standardised units, where one pass covers every horizon because $T \leq 128$. For Moirai, we explicitly construct the context patches, masked prediction patches, and associated token identifiers in a differentiable adapter. The library offers no gradient-free reference for Moirai, so that forward pass was checked only for plausible zero-shot output.

Table 2: The time-series foundation models as configured in this study. $\mathcal{F}$ denotes the token positions read by the forecast head, over which $\mathbf{z}$ is averaged.

	TimesFM-2.5 (200M)	Chronos-2	Moirai-1.1-R (base)
Architecture	decoder, causal self-attention, rotary positions	encoder, alternating time and group attention	masked encoder, any-variate attention
Width d / layers	1280 / 20	768 / 12	768 / 12
Input patch ρ	32	16	32 (chosen from 8–128)
Normalisation $\mathcal{N}$	running mean and scale over context patches	standardisation followed by $\sinh^{-1}$	standardisation
Point forecast	point head, 128-step output patch	median of 21 quantiles	mean of a 4-component mixture
Positions $\mathcal{F}$	last context patch	output patches	prediction patches

B PHYSICS-INFORMED OBJECTIVE

MIT residuals. In the univariate PI setting, the model forecasts power from its history alone. At each forecast step k, we evaluate the coupled RC balances using predicted power $\widehat{P}_k$ (returned to

watts) and observed future temperatures $T_{g,k}, T_{m,k}$. These temperatures enter only the training loss, never the model input or inference. In the multivariate setting, the model instead receives the histories of all five channels and uses its own future power and temperature predictions in the residual. The residuals are therefore

$$r_{g,k} = C_g \dot{T}_{g,k} - \Big(\alpha \widehat{P}_k - \frac{T_{g,k} - T_a}{R_{ga}} - \frac{T_{g,k} - T_{m,k}}{R_{gm}}\Big), \tag{9}$$

$$r_{m,k} = C_m \dot{T}_{m,k} - \Big((1-\alpha) \widehat{P}_k - \frac{T_{m,k} - T_a}{R_{ma}} + \frac{T_{g,k} - T_{m,k}}{R_{gm}}\Big). \tag{10}$$

For $s \in \{\widehat{P}, T_g, T_m\}$, derivatives come from a Gaussian-smoothed series $\tilde{s}(\tau) = \sum_c w_c(\tau) s_c$, where $\tau_c = (c-1)/(T-1)$ and $w_c(\tau) \propto \exp[-(\tau - \tau_c)^2/(2\kappa^2)]$ is normalised over c, with $\kappa = 0.15$. Automatic differentiation gives $\dot{s}_k = \partial_\tau \tilde{s}(\tau_k)/[(T-1)\Delta t]$ and $\ddot{s}_k = \partial_\tau^2 \tilde{s}(\tau_k)/[(T-1)\Delta t]^2$. The heat-flow terms above use unsmoothed temperatures.

Let $\tilde{r} = r/\varsigma$, where $\varsigma = \mathrm{sd}(\widehat{P}) + \mathrm{sd}(T_g) + 10^{-6}$, and let $\omega_k \propto e^{-2(k-1)/T}$ have mean one. The brackets $\langle\cdot\rangle$ average over mini-batch windows and forecast steps, using $k \geq 2$ for $\mathcal{L}_{\mathrm{rate}}$. The losses are

$$\mathcal{L}_{\mathrm{th}} = \big\langle \omega_k (\tilde{r}_{g,k}^2 + \tilde{r}_{m,k}^2) \big\rangle, \tag{11}$$

$$\mathcal{L}_{\mathrm{rate}} = \big\langle \omega_k \big[(\tilde{r}_{g,k} - \tilde{r}_{g,k-1})^2 + (\tilde{r}_{m,k} - \tilde{r}_{m,k-1})^2 \big] \big\rangle, \tag{12}$$

$$\mathcal{L}_{\mathrm{2nd}} = \varsigma_2^{-2} \big\langle \omega_k \big[(\dot{\widehat{P}}_k - \dot{P}_{g,k}^{\mathrm{phys}})^2 + (\dot{\widehat{P}}_k - \dot{P}_{m,k}^{\mathrm{phys}})^2 \big] \big\rangle. \tag{13}$$

Here $\mathcal{L}_{\mathrm{th}}$ penalises energy-balance violations, $\mathcal{L}_{\mathrm{rate}}$ penalises changes in those violations, and $\mathcal{L}_{\mathrm{2nd}}$ matches the forecast power derivative to the GPU- and memory-implied derivatives of Equations (4)–(5). The point of $\mathcal{L}_{\mathrm{2nd}}$ was to constrain how quickly forecast power changes, using the temperature dynamics implied by the RC model. Differentiating each energy-balance equation gives a power-rate estimate, $\dot{P}_g^{\mathrm{phys}}$ or $\dot{P}_m^{\mathrm{phys}}$ so the loss penalizes disagreement between each estimate and $\dot{\widehat{P}}$. It was intended to make forecast ramps physically consistent, particularly during workload transitions. Both ς and $\varsigma_2 = \mathrm{sd}(\dot{\widehat{P}}) + \mathrm{sd}(\dot{P}_g^{\mathrm{phys}}) + \mathrm{sd}(\dot{P}_m^{\mathrm{phys}}) + 10^{-6}$ pool the mini-batch and are excluded from differentiation. Finally, ς is only a numerical scale as it mixes power and temperature readings.

Utilisation-consistency term. The throttle loss relates the predicted power changes $\Delta\widehat{P}_k = \widehat{P}_{k+1} - \widehat{P}_k$ to the utilization $U = \alpha u_g + (1-\alpha) u_m$, expressed as a fraction. It penalizes four patterns as follows: the predicted changes whose sign opposes the observed change of U, predicted increases while $U_k > \theta_U$, the absence of a decrease after a sustained interval with $U > \theta_U$, and predicted increases while both $U_k > \theta_U$ and $T_{g,k} > \theta_T$. The thresholds are mini-batch statistics set as $\theta_U = \max(\bar{U} + \mathrm{sd}(U), 0.85)$ and $\theta_T = \bar{T}_g + 0.5\,\mathrm{sd}(T_g)$. Thermal throttling is essentially absent from both datasets as the MIT GPU temperature stays far below the 80–90 °C limits, and the NLR GPUs peak near 74 °C. The term, therefore, regularizes ramps under high load.

Total loss and weighting. The MIT objective is

$$\mathcal{L} = \lambda_u \mathcal{L}_{\mathrm{data}} + \lambda_r \mathcal{L}_{\mathrm{th}} + \lambda_{\mathrm{rate}} \mathcal{L}_{\mathrm{rate}} + \lambda_2 \mathcal{L}_{\mathrm{2nd}} + \lambda_\theta \mathcal{L}_{\mathrm{util}}, \qquad \lambda_\theta = 0.1. \tag{14}$$

In the self-adaptive mode, the log-weights $\eta_j = \log \lambda_j$ for $j \in \{u, r, \mathrm{rate}, 2\}$ start from $\lambda = (1, 10^{-2}, 10^{-3}, 10^{-3})$. After every update of Φ, they take one Adam step with learning rate 0.01 in the ascent direction $+\nabla_\eta \mathcal{L}$ and are clipped to $[-6, 2]$. Every term is non-negative, so $\partial\mathcal{L}/\partial\eta_j = \lambda_j \mathcal{L}_j \geq 0$ and the ascent has no interior stationary point, so each weight, including the data weight λ_u, rises until it reaches the clip $e^2 \approx 7.39$.

NLR residual. On NLR, temperature is integer-valued at 0.1–0.2 s resolution, so $\dot{T}_{g_i}$ is estimated by central differences and the residual is evaluated only on the interior of the horizon as

$$r_{i,k} = C_g \dot{T}_{g_i,k} - \Big(\widehat{P}_{g_i,k} - \frac{T_{g_i,k} - T_a}{R_{ga}}\Big), \qquad \mathcal{L}_{\mathrm{r}} = \gamma \, \frac{1}{BG(T-2)} \sum_{b=1}^{B} \sum_{i=1}^{G} \sum_{k=2}^{T-1} \Big(\frac{r_{i,k}}{\varsigma}\Big)^2, \tag{15}$$

with $\varsigma = \mathrm{sd}(T_g) + \mathrm{sd}(\widehat{P}_g)$ over the mini-batch, excluded from differentiation. The optional gate $\gamma = \mathrm{clip}\big(\big\langle (\dot{T}_g/\sigma_{\dot{T}})(\dot{\widehat{P}}_g/\sigma_{\dot{\widehat{P}}}) \big\rangle, 0, 1\big)$ is a mini-batch correlation between temperature changes and

predicted power changes. Its normalising deviations are excluded from differentiation but γ itself is not, so gradients also flow through the gate. The NLR objective is $\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_r \mathcal{L}_{\text{r}}$, with the data weight fixed at one. The representation models of Section F use the full residual with the gate at the native rate, an adaptive λ_r starting at 0.05 and capped at one.

C Physics-Informed Formulation with the Proposed Thermal RC Model

The architecture sweep shows that purely data-driven models plateau on structured inference workloads, where even the best vanilla model lags burst onsets and over-predicts the tail of queue-drain phases because no standard loss function encodes the RC thermal inertia that governs how quickly a GPU rack responds to a scheduling event. To remedy this, we derive a novel set of ODEs from first principles and integrate them as physics-informed regularization constraints directly into the training loss. The goal is not only to improve MAE but to produce *interpretable* models whose learned parameters, thermal capacitances, resistances, and queue time-constants, rank consistently with the known vLLM scheduling behavior of each workload.

C.1 Derived ODEs using the Thermal Circuit RC Model

The erratic power transients observed in LLM workloads are not arbitrary noise, they are governed by the thermal inertia of the GPU package. When a new batch arrives, GPU power rises within milliseconds, but the junction temperature lags by seconds due to the thermal mass of the heatsink and coolant loop. This lag, invisible to a pure data-driven loss, is precisely what the RC ODE encodes.

Lumped-parameter RC network. Power consumption and temperature changes are interconnected through principles of energy conservation and Newton's law of cooling. In our streamlined thermal modeling approach, we utilize a lumped-parameter RC network consisting of three related thermal states: the GPU temperature $T_g(t)$, the memory temperature $T_m(t)$, and the CPU temperature T_c, as depicted in Fig. 6. Additionally, T_a represents the effective ambient or sink temperature. This study employs a lumped-parameter model to simplify a spatially distributed system by depicting it as distinct nodes with uniform characteristics. Each node (such as the GPU, memory, and CPU) is assigned a single temperature value instead of representing a continuous temperature gradient. Rather than solving the partial differential equations (PDEs) that define temperature variations at every spatial point, we derive ordinary differential equations (ODEs) to describe the temporal evolution of temperature for each lumped node. The lumped assumption is applicable here since GPUs are constructed with materials that have high thermal conductivity (like copper heat spreaders and thermal paste), leading to a relatively uniform temperature distribution within each component, thus justifying the lumped-parameter modeling approach Saleh et al. (2026).

Applying the thermal–electrical circuit analogy and enforcing energy conservation at each thermal node, the energy-balance ODEs for the GPU, memory, and CPU nodes are

$$C_g \frac{dT_g}{dt} = \alpha P_g - \frac{T_g - T_a}{R_{ga}} - \frac{T_g - T_m}{R_{gm}} - \frac{T_g - T_c}{R_{gc}}, \tag{16}$$

$$C_m \frac{dT_m}{dt} = (1-\alpha) P_g - \frac{T_m - T_a}{R_{ma}} + \frac{T_g - T_m}{R_{gm}} - \frac{T_m - T_c}{R_{mc}}, \tag{17}$$

$$C_c \frac{dT_c}{dt} = P_c - \frac{T_c - T_a}{R_{ca}} + \frac{T_g - T_c}{R_{gc}} + \frac{T_m - T_c}{R_{mc}}, \tag{18}$$

where T_g, T_m, and T_c denote the GPU, memory, and CPU thermal states, respectively, and T_a is the effective ambient/sink temperature. The parameters $C_g, C_m, C_c > 0$ denote the corresponding thermal capacitances, while $R_{ga}, R_{ma}, R_{ca} > 0$ denote the thermal resistances from each node to the ambient. The coupling resistances R_{gm}, R_{gc}, and R_{mc} model heat exchange between the GPU–memory, GPU–CPU, and memory–CPU nodes, respectively. P_g denotes the measured GPU-side power obtained from NVML, whereas P_c denotes the measured CPU-package power obtained through RAPL. Because the NVML GPU-power measurement includes the power contribution

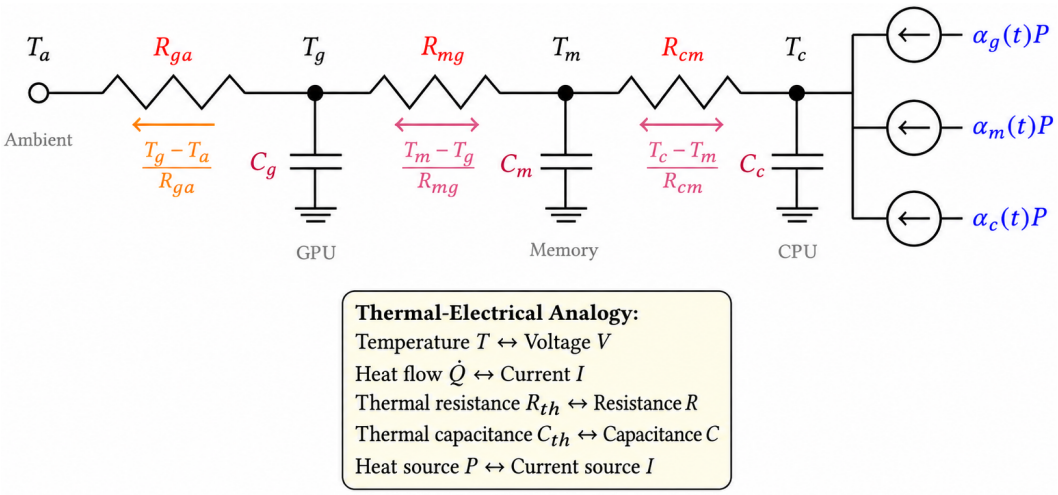


Figure 6: Equivalent three-node RC thermal network for the coupled GPU-core, HBM-memory, and CPU system. The measured GPU-side power P_g is partitioned into effective GPU-core and memory heat-generation terms αP_g and $P_m = \alpha_m(t)P = (1-\alpha)P_g$ respectively, while the independently measured CPU-package power P_c excites the CPU thermal node. Thermal capacitances represent heat storage and thermal resistances represent heat-transfer paths between components and the effective ambient.

associated with the GPU package and high-bandwidth memory (HBM), we introduce a dimensionless parameter $\alpha \in [0, 1]$ that partitions P_g into effective GPU-core and memory heat-generation terms, αP_g and $(1-\alpha)P_g$, respectively.

Solving Equation (16) for the measured GPU-side power P_g and differentiating with respect to time gives the following **GPU power-rate constraint**

$$\frac{dP_g}{dt} = \frac{1}{\alpha}\left[C_g\frac{d^2T_g}{dt^2} + \frac{1}{R_{ga}}\frac{dT_g}{dt} + \frac{1}{R_{gm}}\left(\frac{dT_g}{dt} - \frac{dT_m}{dt}\right) + \frac{1}{R_{gc}}\left(\frac{dT_g}{dt} - \frac{dT_c}{dt}\right)\right], \tag{19}$$

Compared with the two-node formulation, the additional term $(dT_g/dt - dT_c/dt)/R_{gc}$ accounts for transient thermal coupling between the GPU and CPU nodes.

Similarly, solving Equation (18) for the measured CPU/socket power P_c and differentiating with respect to time yields

$$\frac{dP_c}{dt} = C_c\frac{d^2T_c}{dt^2} + \frac{1}{R_{ca}}\frac{dT_c}{dt} - \frac{1}{R_{gc}}\left(\frac{dT_g}{dt} - \frac{dT_c}{dt}\right) - \frac{1}{R_{mc}}\left(\frac{dT_m}{dt} - \frac{dT_c}{dt}\right). \tag{20}$$

The memory temperature derivative dT_m/dt is governed by Equation equation 17, while the CPU temperature derivative dT_c/dt is governed by Equation equation 18. Therefore, Equations equation 19 and equation 20 couple the GPU, memory, and CPU thermal nodes into the physics-informed regularization loss. It is also worth noting that T_g is directly observed through the GPU temperature telemetry, whereas T_m and T_c are latent thermal states introduced by the compact RC model because memory and CPU temperature measurements are not included in the forecasting dataset. The effective thermal coefficients appearing in the state equations, including the input gains and inverse RC time constants, are identified from the training data using recursive least squares (RLS), which is commonly used for parameter identification in thermal RC models Saleh et al. (2026).

In contrast to the two-node formulation in our previous work Saleh et al. (2026), the three-node model uses separate measured power inputs whenever available. Specifically, $P_g(t)$ denotes the GPU-side power obtained from NVML, while $P_c(t)$ denotes the CPU/socket-side power obtained from RAPL.

Since the memory-side power is not directly observed, we retain a latent split parameter $\alpha \in [0, 1]$ only for the GPU–memory partition as

$$P_{\text{core}}(t) = \alpha P_g(t), \qquad P_m(t) = (1 - \alpha) P_g(t). \tag{21}$$

In practical terms, α can be interpreted either as a learned scalar value or as a variable function that adapts based on the workload. It is also crucial to emphasize that fine-tuning the Llama-2 70B parameter LLM involves the use of Low-Rank Adaptation (LoRA) Hu et al. (2021). This method is efficient in terms of parameters, as it adjusts low-rank matrices within the attention and feedforward layers without altering the original model weights. Furthermore, it incorporates DeepSpeed ZeRO-3 for parameter offloading Ren et al. (2021), which manages to distribute all model states, encompassing parameters, gradients, and optimizer states, across several ranks and offload them to CPU memory. This drastically reduces the memory demands on each GPU. CPU power is measured independently through RAPL and is retained as an explicit heat input whenever the CPU thermal path is active.

GPU and HBM memory power. NVIDIA NVML reports per-GPU power that already includes the associated HBM contribution, because the GPU die and its HBM stacks are co-packaged on the same substrate and share the same power delivery network and thermal interface material. We therefore treat the raw `gpu-g[mW]` reading as the single heat source for the g-th thermal node. CPU package power P_c is measured independently through RAPL and enters the CPU energy-balance equation as a heat-generation term. The resulting CPU thermal state T_c, rather than P_c itself, provides the thermally coupled boundary seen by the GPU and memory nodes.

C.2 Physics-Informed Loss Formulation

Let $\hat{P}_{t+k}$ and P_{t+k} denote the predicted and measured aggregate workload IT power, respectively. The supervised forecasting objective is defined as

$$\mathcal{L}_{\text{data}} = \frac{1}{NT} \sum_{n=1}^{N} \sum_{k=1}^{T} \left(\hat{P}_{t+k}^{(n)} - P_{t+k}^{(n)} \right)^2, \tag{22}$$

where N denotes the number of training samples and T is the forecasting horizon.

The physical constraints derived in the preceding section are incorporated as residual penalties. Let r_{th}, r_r, r_p, and r_q denote the thermal energy-balance, derived power-rate, power-response, and queue-response residuals, respectively. For each residual type $j \in \{\text{th}, r, p, q\}$, the corresponding physics loss is computed as

$$\mathcal{L}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} (r_{j,i})^2, \qquad j \in \{\text{th}, r, p, q\}, \tag{23}$$

where N_j denotes the number of valid temporal samples over which the corresponding residual is evaluated. Specifically, $\mathcal{L}_{\text{th}}$ penalizes violations of the thermal energy-balance equations, $\mathcal{L}_r$ penalizes violations of the derived power-rate constraints, $\mathcal{L}_p$ enforces the power-response constraint, and $\mathcal{L}_q$ penalizes violations of the first-order queue-response relation. The queue-response term is used only for the inference workloads.

Since the forecasting architectures operate on uniformly sampled discrete sequences rather than continuous-time functions, the temporal derivatives required by the physics residuals are evaluated using differentiable finite-difference operators. For a sampled signal z_t with sampling interval Δt we have

$$\dot{z}_t \approx \frac{z_{t+1} - z_{t-1}}{2\Delta t}, \qquad \ddot{z}_t \approx \frac{z_{t+1} - 2z_t + z_{t-1}}{\Delta t^2}, \tag{24}$$

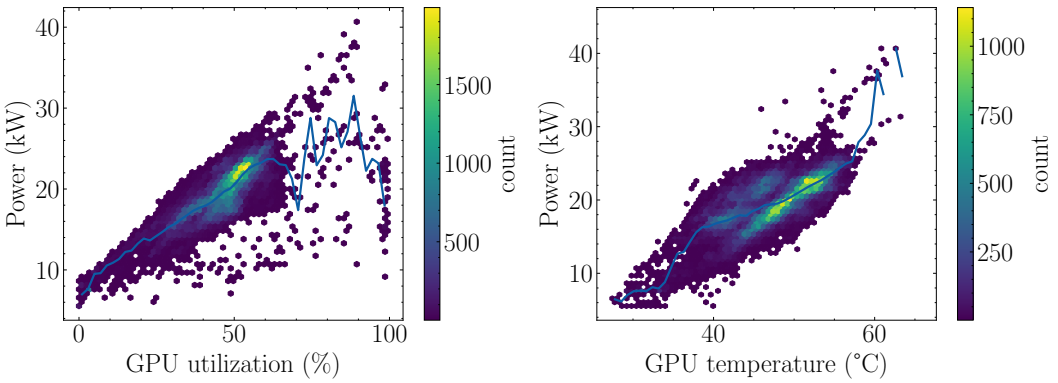
with one-sided differences used at sequence boundaries. These operations remain within the differentiable computational graph, allowing automatic differentiation to propagate the resulting physics-loss gradients back to the forecasting-model parameters.

D The MIT Supercloud Dataset

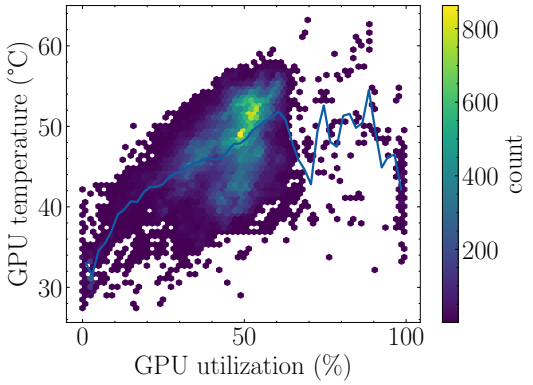
The data includes GPU power, which often accounts for a large share of AI workload power demand, together with memory utilization, GPU temperature, and storage metrics. Before forecasting, the raw records are sliced into time-series windows and scaled using Min–Max normalization.

We study data-center power forecasting using the MIT Supercloud dataset Samsi et al. (2021), collected from a high-performance computing system equipped with NVIDIA Volta V100 GPUs and Intel Xeon Gold 6248 CPUs. Cooling-system power was not recorded, so it could not be modeled alongside the measured components.

The dataset covers February through October 2021 and contains logs at 100-millisecond intervals of GPU and CPU utilization, scheduling activity, power draw, and temperature (Fig. 7). Its anonymized records are indexed by job ID and node, and aggregate GPU power peaks at 45 kW across 448 GPUs. Vision workloads dominate, led by U-Net (1,431 jobs), followed by VGG, ResNet, and Inception. The dataset also includes language models such as BERT (189 jobs) and DistillBERT (172 jobs), and graph neural networks including SchNet, DimeNetm, PNA, and conv (Fig. 8). For forecasting, the data are processed at one-minute resolution, power is aggregated by job ID and node, and Min–Max-scaled windows of varying look-back lengths are used to predict power over different minute-ahead horizons.



(a) Power vs. utilization (left) and power vs. temperature (right).



(b) Temperature vs. utilization.

Figure 7: Relationships observed in the MIT Supercloud dataset: power vs. utilization (left), power vs. temperature (right), and temperature vs. utilization (bottom). Lighter regions indicate higher count, and darker regions indicate lower count.

Table 3 summarizes the MIT Supercloud GPU cluster's long-duration, high-resolution multivariate trace. After preprocessing, seven variables were sampled at one-minute intervals, producing approximately 330,000 timesteps over 238 days for evaluating short- and longer-horizon forecasts under realistic AI workload variation. Five variables were retained for both the proposed model and the state-of-the-art baselines: memory utilization (%), memory temperature, GPU utilization (%), GPU temperature, and power draw. Memory used and memory free were excluded because they showed low observed correlation.

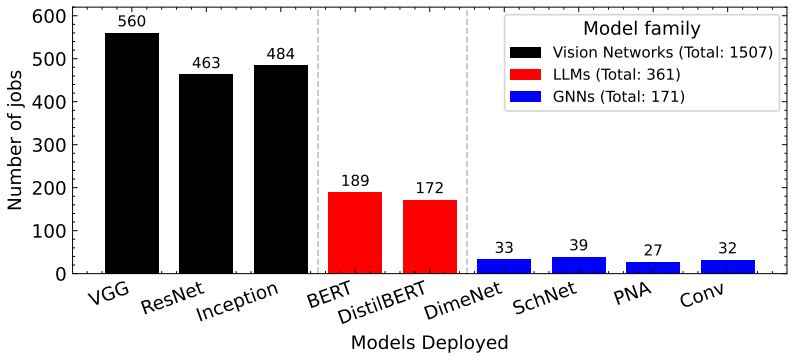


Figure 8: Computational job distribution across the AI workloads present in the MIT Supercloud dataset, namely, vision networks, large language models (LLMs), and graph neural networks (GNNs).

Table 3: Summary of the MIT Supercloud dataset used in this study.

Attribute	**MIT Supercloud GPU Trace**
Co-variates	7 (GPU utilization, memory utilization, free memory, used memory, GPU temperature, memory temperature, power draw)
# Timesteps	∼330,500
Granularity	1 min (after data processing step)
Duration	∼238 days

Table 4: Instantiation of the electro-thermal prior on the two datasets.

Parameters/Dataset	**MIT**	**NLR**
Measured power	fleet sum $P(t)$	per GPU $P_{g_i}(t)$, $G = 4$
Measured temperature	fleet means T_g, T_m	per GPU T_{g_i}, integer °C
Network	two nodes, Equations (2)–(3)	one node per GPU, Equation (6)
Power split α	required, fixed at 0.5085	not needed
Sampling Δt	60 s	0.1 s or 0.2 s; block means of 0.2–1 s where stated
$\dot{T}$ estimator	kernel smoother, automatic differentiation	central differences

E THE NLR DATASET: AN OVERVIEW OF THE PROCESSED DATA FOR EACH AI WORKLOAD

The NLR dataset provides a hierarchical view of GenAI data-center power behavior by linking node-level CPU/GPU telemetry with facility-level demand. Each GPU node consists of AMD EPYC CPUs, four NVIDIA H100 GPUs, system memory, NVMe storage, and PCIe/NVLink interconnects, which together represent the dominant workload-dependent sources of power consumption and heat generation. These nodes are installed in GPU-accelerated chassis, mounted in standard racks, and operated within environmentally controlled data-center rooms. At the facility level, the aggregated IT load is supported by centralized electrical and thermal infrastructure, including transformers, UPS systems, switchgear, generators, chillers, pumps, heat exchangers, and hybrid air–liquid or fully liquid-cooling systems. This hierarchy connects the fast power dynamics of individual computational components to the broader facility-level load observed by the forecasting model.

Table 5 summarizes the five processed workload CSVs generated from the NLR traces, including two training workloads and three inference workloads. Each CSV is resampled to one-second resolution and contains the 13-channel telemetry used for forecasting, including GPU power, GPU temperature, CPU package/core power, and the facility-level target `OT`. The workload statistics show that inference sessions are generally shorter but more frequent, whereas training jobs are longer and exhibit different power characteristics. Moreover, the mean facility power varies substantially across workload types, supporting a per-workload evaluation rather than a pooled forecasting setup. Fig. 9 illustrates these differences in sample count and mean power, while Fig. 10 shows that total GPU power is strongly coupled with facility demand, with workload-specific offsets caused by CPU activity and infrastructure overhead. GPU temperature generally follows GPU power, except in rate-limited online inference, where short idle intervals prevent the GPU from reaching thermal steady state. These observations justify using the full 13-channel telemetry and motivate the RC thermal prior adopted in the proposed physics-informed forecasting framework.

E.1 SPECIFICATIONS OF NLR'S KESTREL HPC SYSTEM

Experiments were conducted on the GPU-accelerated nodes of NLR's Kestrel HPC system, which includes 156 GPU nodes in addition to 2,314 CPU-only nodes. Each GPU node contains four NVIDIA H100 SXM GPUs, two AMD EPYC 9554 Genoa processors with 64 cores per socket, and high-speed intra- and inter-node communication through NVLink and the HPE Slingshot-11 interconnect. The nodes provide different system-memory and local-storage configurations, ranging from 384 GB to 1.5 TB of DDR5 memory and from 3.4 TB to 14 TB of NVMe storage, while each H100 GPU includes 80 GB of HBM3 memory for large-scale AI workloads. From a power perspective, each node can

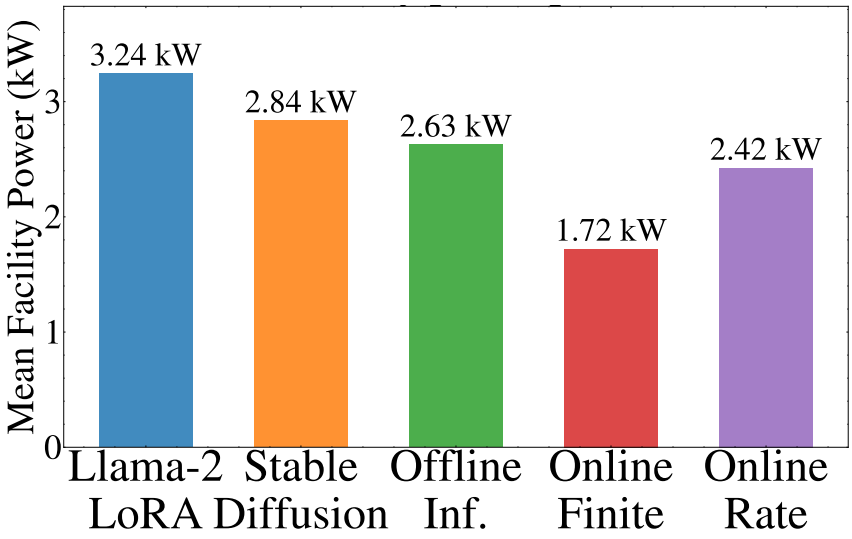


Figure 9: Workload distribution: mean facility power in kW per workload type. Training workloads are longer but run fewer sessions, while inference workloads are shorter and more numerous.

Table 5: Summary of the five NLR workload CSVs. The sessions are individual slurm jobs or vLLM request segments. Duration is total wall-clock time across all sessions.

Workload	# Timesteps	Duration	Notes
Llama-2 70B LoRA Training	58,319	16.2 h	Multi-node, checkpoint bursts
Stable Diffusion Training	61,492	17.1 h	Multi-node, diffusion cycles
Offline Inference (Llama-3 70B)	50,528	14.0 h	Batched vLLM requests
Online Finite Inference	71,674	19.9 h	Finite token-budget sessions
Online Rate Inference	64,480	17.9 h	Rate-limited token streaming

Co-variates (13): 4× GPU power [mW], 4× GPU temp [°C], 2× CPU package power [W], 2× CPU core power [W], facility OT [W].
Granularity: 100 ms. GPU + memory power joined per chip.

draw up to approximately 3.52 kW under full CPU/GPU loading, excluding peripherals, based on four 700 W GPUs and two CPU sockets. Stress-test measurements further confirmed peak GPU power near 696 W per H100 and CPU socket power around 338.6 W, while idle operation averaged 72.5 W per GPU and 64.1 W per CPU socket. These specifications make Kestrel a suitable platform for studying high-resolution power dynamics in GPU-intensive GenAI workloads.

E.2 Processed Data

Our preprocessing pipeline aggregates the power telemetry of the compute nodes participating in each workload into a common multivariate representation. For each node, we extract the four per-GPU NVML power measurements (`gpu0_mW`, . . ., `gpu3_mW`), the corresponding GPU temperatures (`gpu0_C`, . . ., `gpu3_C`), and the per-socket CPU power measurements obtained through RAPL. For multi-node training jobs, power measurements are summed across the participating compute nodes, whereas temperature measurements are averaged across nodes.

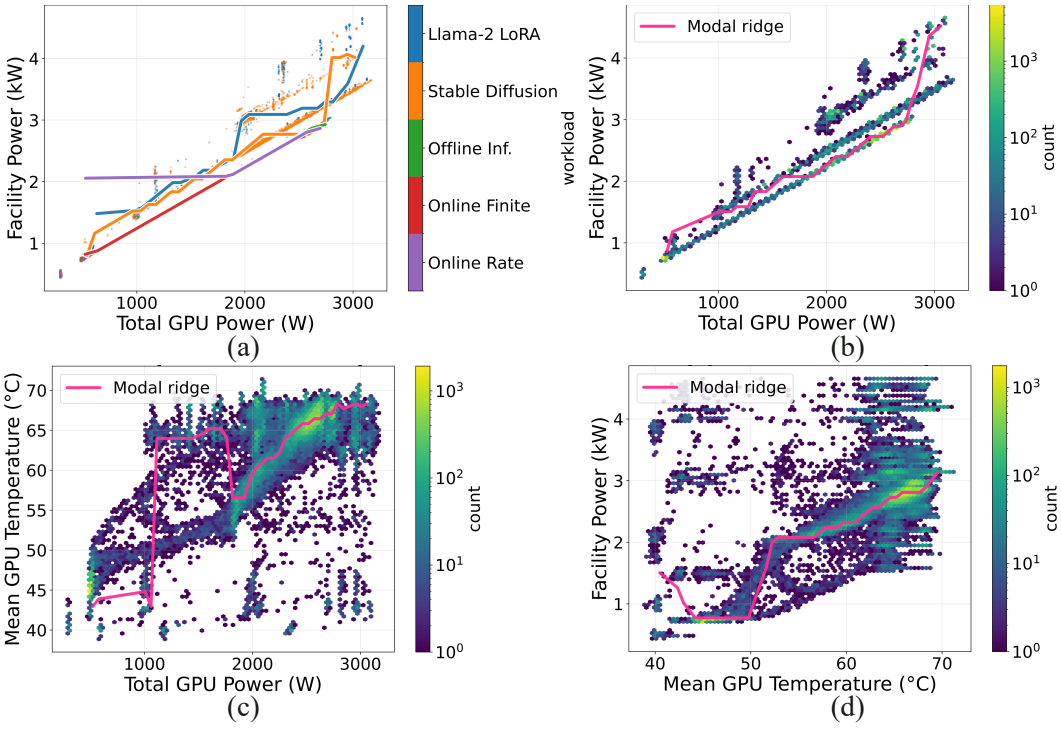


Figure 10: Scatter density plots of key physical relationships across all five NLR workloads. The strong but workload-dependent coupling motivates using the full 13-channel telemetry as input and deriving workload-specific physics constraints.

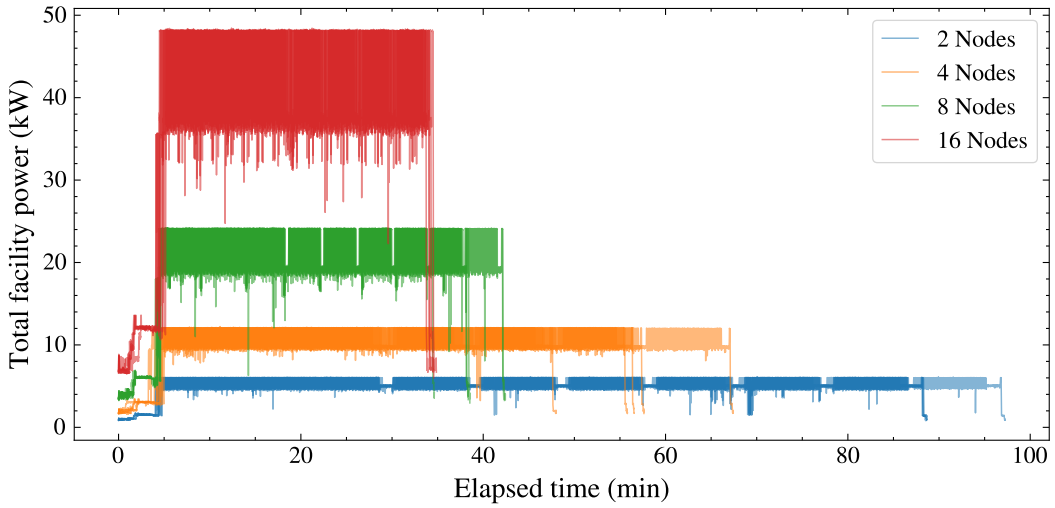


Figure 11: Power consumption profiles for fine-tuning Llama-2 70B are depicted. The replicated experiments, conducted with identical node counts, are presented as overlapping transparent curves in various colors. Each profile summarizes the total power usage across all CPU and GPU resources involved in the workload.

The forecasting target, denoted by (OT), represents the aggregate workload IT power rather than the total electrical demand of the data-center facility. Specifically, for a workload executing on the set of compute nodes ($\mathcal{N}$), we define

$$P_{\mathrm{IT}}(t) = \sum_{n \in \mathcal{N}} \left[\sum_{g=1}^{4} P_{n,g}^{\mathrm{GPU}}(t) + \sum_{s=1}^{2} P_{n,s}^{\mathrm{CPU,pkg}}(t) \right], \tag{25}$$

where ($P_{n,g}^{\mathrm{GPU}}$) is the NVML-reported power of GPU (g) on node (n), ($P_{n,s}^{\mathrm{CPU,pkg}}$) is the package-level power of CPU socket (s), and $\mathrm{OT}(t) \equiv P_{\mathrm{IT}}(t)$. GPU-temperature and CPU-core-domain measurements are retained as explanatory covariates because they provide information about thermal and processor activity, but they are not separately added to (OT). Thus, (OT) captures the time-varying IT power attributable to the active AI workload and should not be interpreted as whole-facility

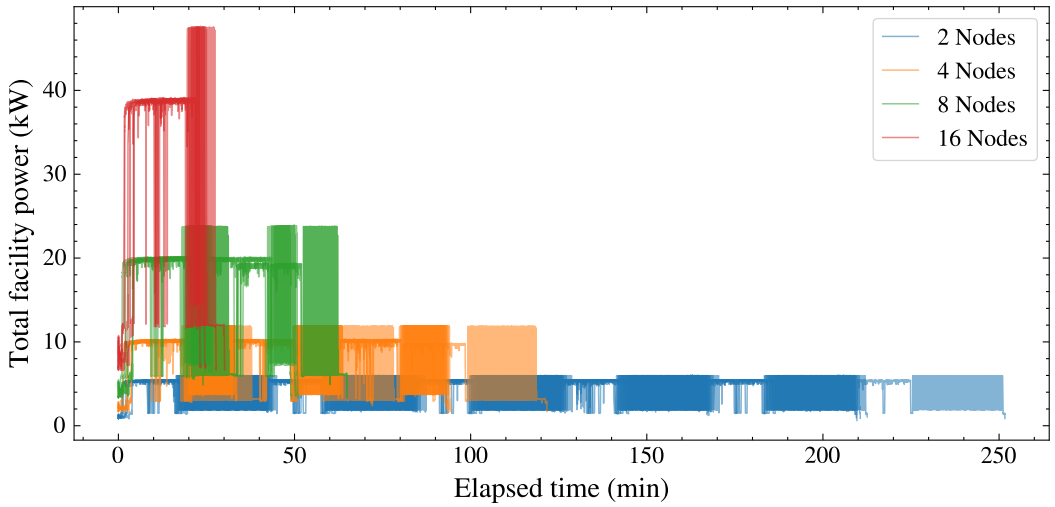


Figure 12: The power consumption patterns for Stable Diffusion training are illustrated. Experiments that were duplicated with identical node counts appear as overlapping translucent curves in various colors. Each profile represents the total power usage across all CPU and GPU resources engaged by the workload.

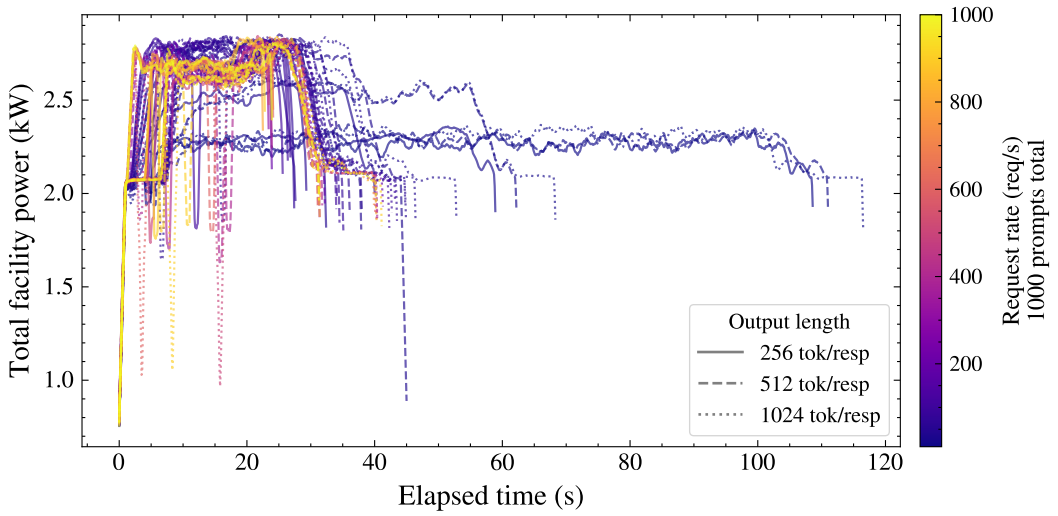


Figure 13: The benchmark results for the online inference of Llama-3 70B power series are analyzed based on the dataset, request frequency, and the length of output tokens, with a fixed batch size set at 1,000 prompts. The median values are represented by lines, while the shaded areas indicate the 10th to 90th percentiles.

electrical demand, which would additionally require modeling cooling, power-conversion losses, networking, storage, and other infrastructure loads. This distinction is consistent with the NLR dataset methodology, in which measured workload power profiles are separately scaled to representative whole-facility demand using the DIPLOEE model Vercellino et al. (2026).

All time series are subsequently aligned to a common one-second grid using nearest-neighbor alignment with a five-second forward-fill limit. Each Slurm job therefore produces one $(T \times 13)$ table, while each inference session produces one $(T \times 13)$ table sliced by request. A `site_id` identifies the corresponding Slurm job or inference session and is used to form the train, validation, and test partitions.

E.3 Processed Results

E.3.1 Llama-2 70B (LLM) Fine-Tuning Workload Power Profiles

The Llama-2 70B fine-tuning workload exhibits clear transient power behavior that is directly relevant to facility-level forecasting as shown in Fig. 11. Across different node counts, the power traces begin with an approximately three-minute ramp-up phase associated with checkpoint loading, dataset initialization, CUDA context creation, and ZeRO-3 state partitioning, followed by a fine-tuning phase marked by high-frequency fluctuations and intermittent low-activity intervals. These variations arise from step-synchronous data-parallel training, where GPU computation alternates with collective communication and CPU–GPU transfers related to ZeRO-3 optimizer and state offloading. As the number of nodes increases, the fluctuations become more pronounced due to the larger number of data-parallel ranks and higher communication and synchronization overhead. Although runtime per epoch showed strong scaling up to 16 nodes, the fixed per-device batch size caused the global batch size and average power consumption to increase with node count, leading to higher total training energy under the tested configuration. These observations show that scaling GenAI fine-tuning does not automatically improve energy efficiency and that accurate short-term power forecasting is important for data-center provisioning, scheduling, and real-time load management.

E.3.2 Stable Diffusion (Image Generation) Training Workload Power Profiles

The Stable Diffusion training workload exhibited power dynamics that were notably different from those of Llama-2 fine-tuning and therefore important for facility-level forecasting as observed in Fig. 12. In particular, execution times were longer, especially at smaller node counts, and the workload showed a much wider range of time-resolved power fluctuations, driven mainly by the computationally expensive and iterative diffusion sampling used during evaluation. For example, at 16 nodes, facility

power varied substantially as the workload alternated between lower-activity periods and short bursts of intensive computation associated with repeated denoising and synchronization steps. Although average power and power variability increased roughly linearly with node count, these aggregate trends masked the strong transient behavior visible in the full power traces, underscoring the need for time-resolved analysis. Unlike the Llama-2 case, total energy consumption tended to decrease with increasing node count because shorter execution times offset the additional power draw, although this behavior was closely tied to the evaluation schedule and is not necessarily general. Overall, the results show that model architecture strongly influences facility-level load profiles, reinforcing the importance of workload-specific forecasting models for GenAI data centers.

E.3.3 Llama-3 70B (LLM) Offline & and Online Inference Workload Power Profiles

The Llama-3 70B offline inference experiments were conducted on one compute node using vLLM by varying both the number of input prompts and the maximum output-token length. Specifically, the offline tests considered prompt counts of 25, 100, 500, 750, and 1000, with output-token lengths of 512 and 1024. Across these cases, the power traces showed a rapid ramp-up followed by either

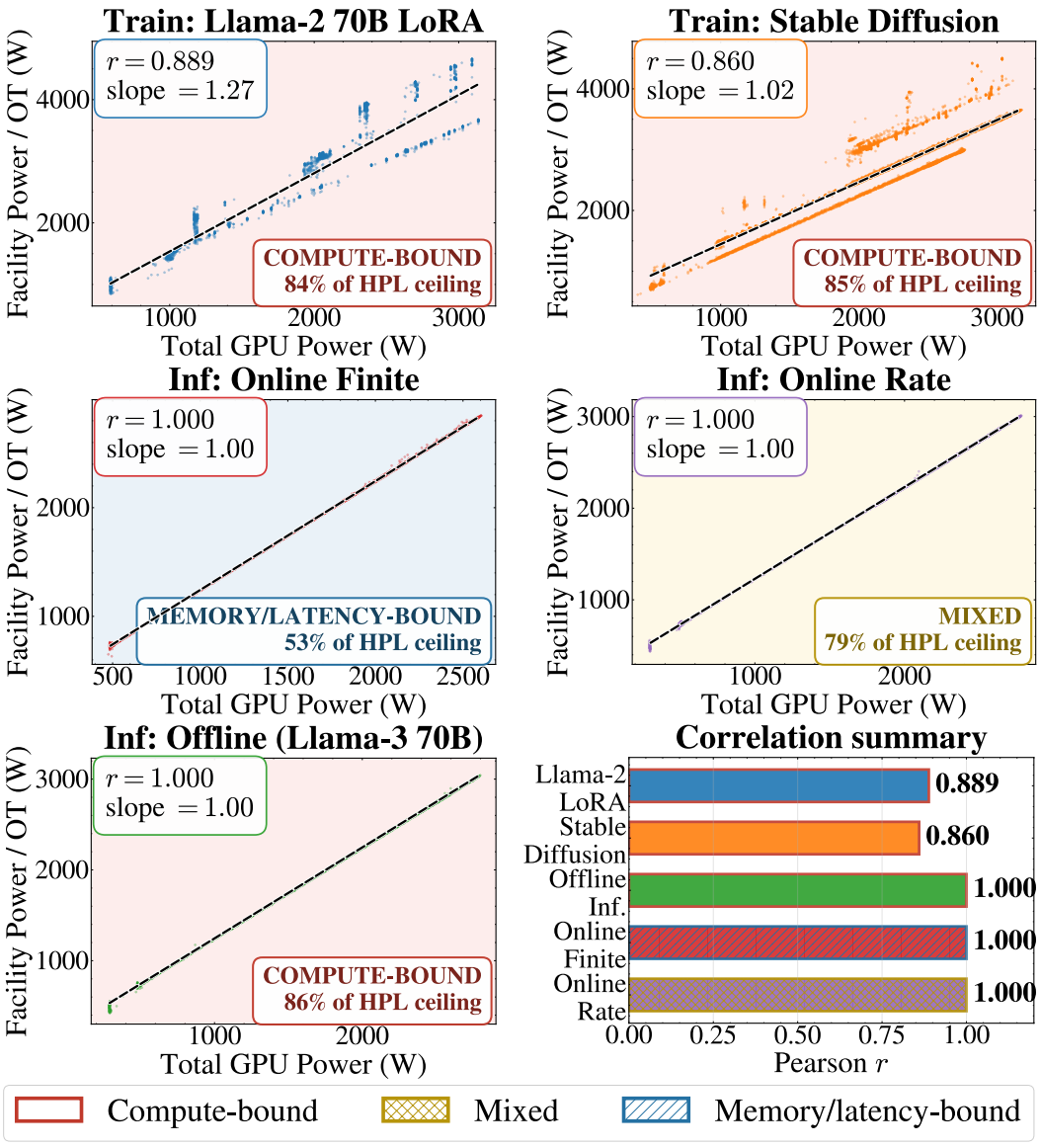


Figure 14: GPU Power vs. Facility power-NLR GenAI Workloads.

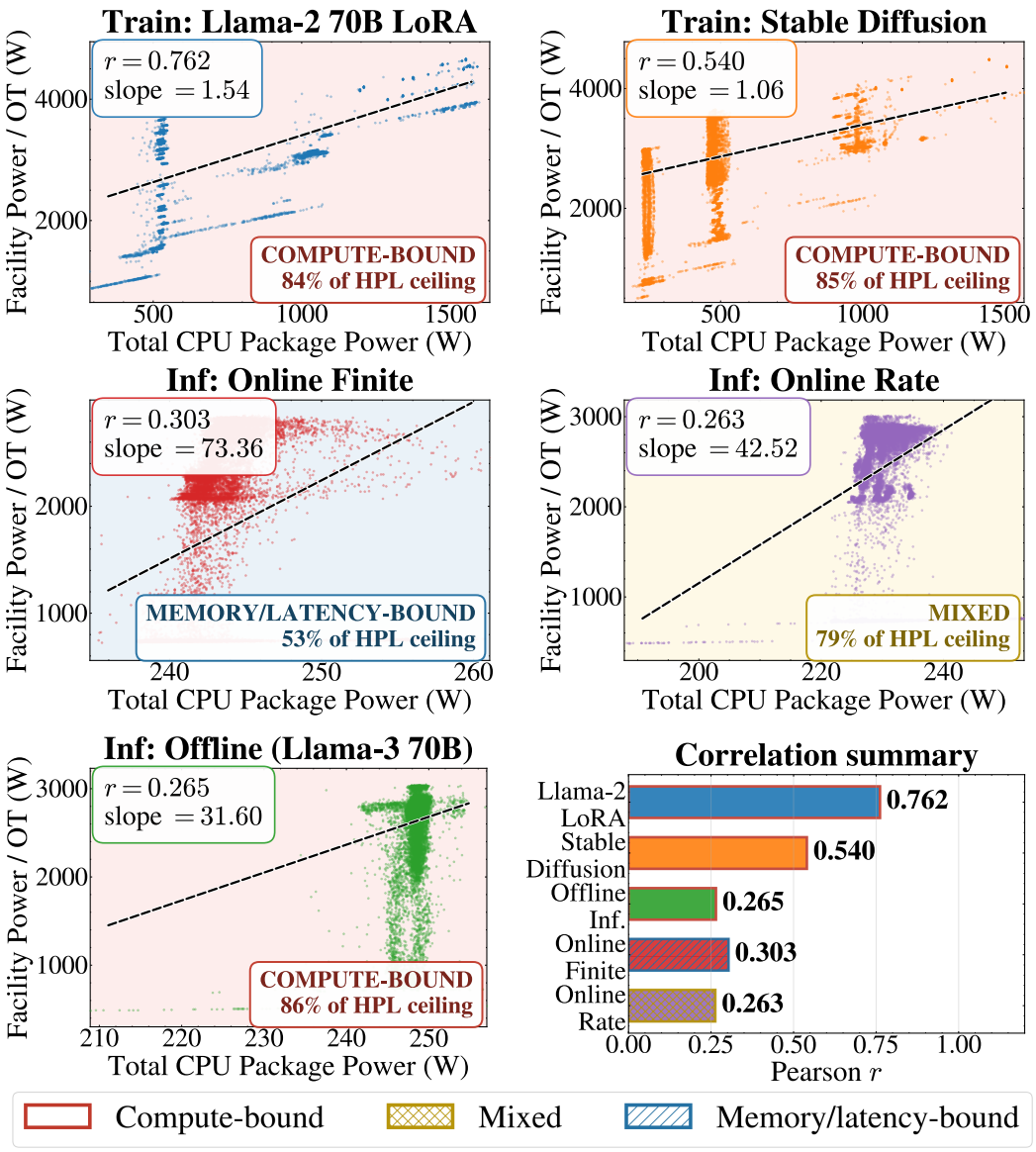


Figure 15: CPU Power vs. Facility power-NLR GenAI Workloads.

sustained or gradually decreasing consumption, with total execution time ranging from approximately 10 s for the shortest runs to about 85 s for the longest runs. Increasing the output-token length had an almost proportional effect on runtime, while increasing the number of prompts raised power only up to a saturation point: power consumption leveled off near 2.8 kW after roughly 325 prompts. High-prompt cases also showed larger variability and intermittent dips, indicating that inference power is governed by nonlinear saturation effects rather than scaling directly with prompt count or token length.

The online inference experiments better represent production LLM serving, where requests arrive dynamically rather than as a fixed batch as shown in Fig. 13. These tests used 1000 prompts and varied the dataset, request rate, output-token length, and random seed, with three repetitions per configuration. The evaluated request rates included 10, 20, 100, and 1000 prompts/s in the finite online-serving tests, while the sustained three-minute tests considered request rates such as 1, 10, 25, 50, and 100 prompts/s. Output-token lengths of 256, 512, and 1024 were used. The results show that request rate was the dominant factor shaping the online power traces, while output-token length mainly affected the response tail and became more influential at higher request rates. Throughput increased with request rate but saturated as the system approached its serving limit, while power and

power variability rose rapidly and saturated around 2.8–2.9 kW, with the sustained tests reaching near 2.8 kW at approximately 25 prompts/s. These findings demonstrate that LLM inference produces bursty, token-dependent, rate-dependent, and saturation-driven power profiles, reinforcing the need for workload-specific, physics-informed forecasting models for realistic GenAI facility-level demand prediction.

Figs. 14 and 15 show that GPU power is strongly correlated with aggregate workload IT power, whereas CPU power exhibits substantially weaker and more workload-dependent correlations. The workload characterization further distinguishes compute-bound regimes (Llama-2, Stable Diffusion, and Online Finite), where computation dominates power demand, from memory/latency-bound Offline Inference, where data movement and memory-access latency are more limiting, and mixed Online Rate inference, where both effects contribute. These trends explain the near-unity GPU–IT-power correlation for the inference workloads and the much weaker CPU correlations, particularly for Offline and Online inference.

F Representation and Weight-Space Explainability Analysis

Labels. Labels are computed from the forecast window only, so a probe cannot recover them by reading the look-back from $\mathbf{z}$. Let $\Delta P_{t+k} = P_{t+k+1} - P_{t+k}$, and let $\varsigma_{\Delta P}$ be the standard deviation of one-step power changes in the training data (on NLR, of per-GPU power, with the label computed on the four-GPU mean). The labels are: *transient*: $\mathbf{1}\{\max_k |\Delta P_{t+k}| > 2\varsigma_{\Delta P}\}$; *event*: transient, or an operating condition c. On MIT, c indicates GPU utilisation above 90% within the horizon. On NLR, c indicates power-limit operation, $\max_{k,i} P_{g_i,t+k} > 0.97 P_{\max}$, where $P_{\max}$ is the largest per-GPU power in the first eight training runs; *thermal regime*: cooling, steady or heating, from the change of T_g over the horizon with a ± 0.25 K dead band; *workload identity*: on NLR, one of five AI workload classes.

Geometry. With $\bar{\mathbf{z}} = \mathbf{z}/\|\mathbf{z}\|_2$ and class prototypes $\boldsymbol{\mu}_c$, the normalized class means of $\bar{\mathbf{z}}$, we have compactness $\mathcal{C} = \langle \bar{\mathbf{z}}_n^\top \boldsymbol{\mu}_{c(n)} \rangle$ and dispersion $\mathcal{D} = \binom{K}{2}^{-1} \sum_{c<c'} \boldsymbol{\mu}_c^\top \boldsymbol{\mu}_{c'}$ (Ming et al., 2023). Both are computed without centering, so both include the direction common to all embeddings. Moreover, we have the prototype energy $E(\mathbf{z}) = \tau \log \sum_c \exp(\bar{\mathbf{z}}^\top \boldsymbol{\mu}_c/\tau)$ with $\tau = 1$, and the k-nearest-neighbour score with $k = 50$ (Sun et al., 2022), both referenced to training embeddings. From the eigenvalues ν_i of the centered covariance, the anisotropy $\nu_1/\sum_i \nu_i$ is computed along with the effective rank $(\sum_i \nu_i)^2/\sum_i \nu_i^2$. Finally, the Fisher ratio $\mathrm{tr}\,\mathbf{S}_B/\mathrm{tr}\,\mathbf{S}_W$ and the cosine distance between the mean embedding of each arm and that of (Z) are obtained.

Weight space. We measure fine-tuning drift from the pretrained weights Φ_0 as $\Delta(\Phi) = \|\Phi - \Phi_0\|_F/\|\Phi_0\|_F$, separately by layer and module type. We also compare the physics-informed and baseline/vanilla weights directly and inspect changes in the singular values of a representative matrix. Since drift depends on training duration, the optimization steps for each checkpoint are taken into account accordingly.

Probes and controls. We test whether the embeddings encode the forecast-window labels using a decision tree, random forest, and linear probe, with identical settings for all. Five-fold group cross-validation holds out the entire NLR runs or contiguous MIT time blocks. Additionally, the pooled held-out predictions yield balanced accuracy, macro-F1, AUROC, and AUPRC. Following Hewitt & Liang (2019), each probe is compared with the same classifier trained on six look-back statistics per channel or on the permuted labels. Finally, the probes are trained on the training-period embeddings and evaluated on the test period.

G Identification of the electro-thermal parameters

For MIT (V100) and NLR (H100), the bounds are outlined in Table 7, where $C_g \in [100, 400]\ \mathrm{J,K^{-1}}$ (die-copper-heat-pipe assembly), $C_m \in [10, 50]\ \mathrm{J,K^{-1}}$ (HBM stacks and heat spreader), $R_{ga} \in [0.2, 0.6]\ \mathrm{K,W^{-1}}$ (forced-air die-to-ambient path), $R_{ma} \in [0.3, 1.0]\ \mathrm{K,W^{-1}}$ (memory-baseplate-to-air path), and $R_{gm} \in [0.05, 0.5]\ \mathrm{K,W^{-1}}$ (interposer-mediated GPU–memory coupling). Because MIT Supercloud aggregates power across 448 GPUs, we apply $C_{\mathrm{agg}} = NC_{\mathrm{dev}}$ and $R_{\mathrm{agg}} = R_{\mathrm{dev}}/N$,

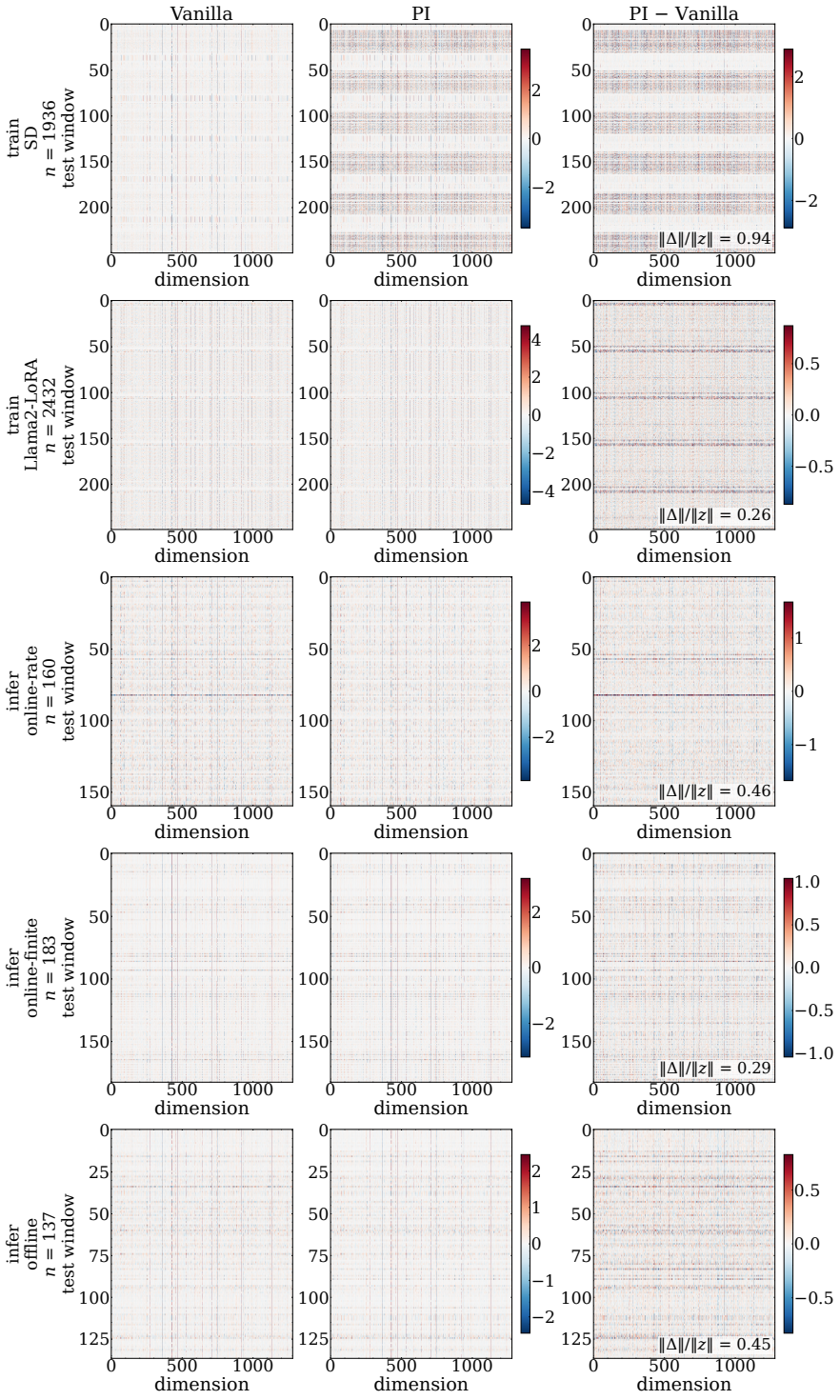


Figure 16: **Workload-dependent effect of physics-informed training on the TimesFM's embeddings for the NLR dataset.** Rows show Stable Diffusion and Llama2-LoRA training, followed by Llama3 online-rate, online-finite, and offline inference. The three columns show Vanilla TimesFM embeddings, PI embeddings, and their paired difference, $\Delta\mathbf{z} = \mathbf{z}_{\mathrm{PI}} - \mathbf{z}_{\mathrm{Vanilla}}$. Each heatmap displays up to 250 matched test windows (vertical axis) across 1,280 embedding dimensions (horizontal axis), where n is the total number of paired windows for that workload. Vanilla and PI share a color scale within each row, whereas the difference has its own scale. The annotated mean relative changes, $\langle \|\Delta\mathbf{z}\|_2 / \|\mathbf{z}_{\mathrm{Vanilla}}\|_2 \rangle$, are 0.94, 0.26, 0.46, 0.29, and 0.45 from top to bottom, showing that physics-informed training reshapes the representation to different degrees across the AI workloads.

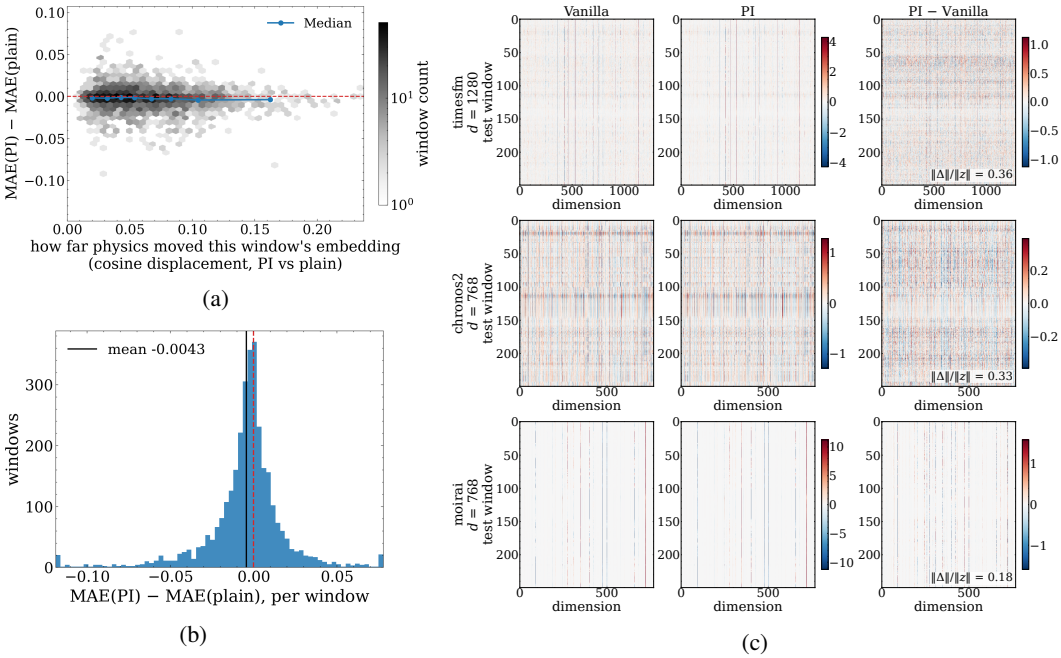
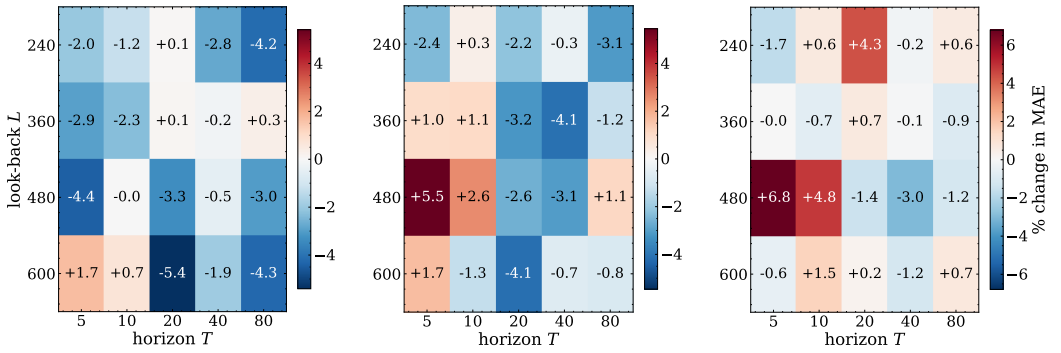


Figure 17: **Effect of physics-informed training on the embedding vectors.** On the MIT test set ($L = 480$, $T = 20$), (a) relates the cosine displacement between each window's Vanilla and PI embeddings to its change in forecast error, $\Delta\text{MAE} = \text{MAE}_{\text{PI}} - \text{MAE}_{\text{Vanilla}}$. Shading counts windows, blue points show binned medians, and negative ΔMAE favors PI. (b) shows ΔMAE across 3,256 paired windows, where PI improves 59% of forecasts, with mean -0.0043 (black line; the red line marks zero). (c) shows Vanilla embeddings, PI embeddings, and their difference for 250 paired windows per backbone (rows: TimesFM, Chronos-2, Moirai; horizontal axis: embedding dimension). Vanilla and PI share a color scale within each row. Their mean relative ℓ_2 changes are 0.36, 0.33, and 0.18, respectively.

giving $C_g \in [4.48 \times 10^4, 1.792 \times 10^5]$ J, K^{-1}, $C_m \in [4.48 \times 10^3, 2.24 \times 10^4]$ J, K^{-1}, $R_{ga} \in [4.46 \times 10^{-4}, 1.34 \times 10^{-3}]$ K, W^{-1}, $R_{ma} \in [6.70 \times 10^{-4}, 2.23 \times 10^{-3}]$ K, W^{-1}, and $R_{gm} \in$



(a) TimesFM-PI better in 15/20 (mean -1.78%).

(b) Chronos-2-PI better in 13/20 (mean -0.79%).

(c) MOIRAI-PI better in 11/20 (mean +0.46%).

Figure 18: **Effect of physics-informed training on forecasting accuracy.** Figs (a)–(c) show TimesFM, Chronos-2, and Moirai, respectively. Each cell reports the percentage change in test MAE relative to Vanilla, $100(\text{MAE}_{\text{PI}} - \text{MAE}_{\text{Vanilla}})/\text{MAE}_{\text{Vanilla}}$, for the indicated look-back length L and forecast horizon T. Negative (blue) values favor PI; positive (red) values favor Vanilla. Across the 20 settings per model, PI improves 15 for TimesFM, 13 for Chronos-2, and 11 for Moirai, with mean MAE changes of -1.78%, -0.79%, and $+0.46\%$, respectively.

Table 6: Event detection performance on the MIT Supercloud and NLR datasets. The physics-informed (PI) formulation consistently improves balanced accuracy, AUROC, and AUPRC compared with the corresponding vanilla model. Δ denotes the absolute improvement obtained using the PI formulation.

Dataset	**Model**	**Bal. Acc. (%)**	**AUROC (%)**	**AUPRC (%)**
MIT Supercloud	Vanilla	52.9	52.3	23.4
	PI	**55.7**	**56.8**	**26.4**
	Δ	+2.8	+4.5	+3.0
NLR	Vanilla	88.5	87.9	82.3
	PI	**89.3**	**89.4**	**86.7**
	Δ	+0.8	+1.5	+4.4

Table 7: Free parameters and admissible ranges for identification. MIT ranges are per device and are scaled to the fleet as described in the text.

Dataset	Parameter	Range	Scale
MIT (V100, per device)	C_g, C_m	$[100, 400]$, $[10, 50]$ $\mathrm{J\,K^{-1}}$	log
	R_{ga}, R_{ma}, R_{gm}	$[0.2, 0.6]$, $[0.3, 1.0]$, $[0.05, 0.5]$ $\mathrm{K\,W^{-1}}$	log
NLR (H100)	C_g	$[0.2, 400]$ $\mathrm{J\,K^{-1}}$	log
	R_{ga}	$[0.02, 0.06]$ $\mathrm{K\,W^{-1}}$	log
	G_{gg}	$[0, 20]$ $\mathrm{W\,K^{-1}}$	linear
	R_{loop}	$[10^{-4}, 2 \times 10^{-2}]$ $\mathrm{K\,W^{-1}}$	log
	T_{in}	$[15, 40]$ °C	linear
Both	σ_R	$[10^{-4}, 1]$ (fraction of S)	log

$[1.12 \times 10^{-4}, 1.12 \times 10^{-3}]$ $\mathrm{K, W^{-1}}$; thus, $\tau = RC$ remains invariant under aggregation, while log-space parameterization prevents negative or otherwise non-physical solutions.

Parameters and admissible ranges. Each free parameter is confined to a range $[\ell_j, h_j]$ through $\theta_j = \ell_j(h_j/\ell_j)^{\sigma(\zeta_j)}$ (log-scaled) or $\theta_j = \ell_j + (h_j - \ell_j)\,\sigma(\zeta_j)$ (linear), where σ is the logistic function and no value of ζ_j can leave the range. Table 7 lists the free parameters and their ranges, including the physics noise scale σ_R.

On MIT, the per-device ranges describe an air-cooled V100. They are scaled to the fleet by $C \to NC$ and $R \to R/N$ with $N = 448$, which leaves every time constant RC unchanged. The ambient is held at $T_a = 27\,^{\circ}\mathrm{C}$. The split is held at $\alpha(t) = b_g u_g(t)/[b_g u_g(t) + b_m u_m(t)]$, clipped to $[0.05, 0.95]$, from the least-squares regression $P \approx b_0 + b_g u_g + b_m u_m$ on the training split.

On NLR, the ambient is the node's shared coolant loop,

$$T_a(t) = T_{\mathrm{in}} + R_{\mathrm{loop}}\Big(\sum_{i=1}^{G} P_{g_i}(t) + \sum_{j=1}^{2} P_{c_j}(t)\Big), \tag{26}$$

where P_{c_j} are the CPU-package powers. A variant with a fixed $T_a = 27\,^{\circ}\mathrm{C}$ and free parameters $\{C_g, R_{ga}, G_{gg}\}$ is also fitted.

Power measurement model (NLR). The NLR per-GPU power was logged with `nvmlDeviceGetPowerUsage`, which on the H100 returns the mean power over the preceding second (Yang et al., 2024); the temperature is an instantaneous integer reading. In agreement with this, temperature increments lead power increments by one to two samples in all three workloads, and a pointwise least-squares fit of Equation (6) returns $C_g < 0$ (Appendix G). Let $\mathcal{A}_w x(t) = w^{-1}\int_{t-w}^{t} x(s)\,ds$, so that the measured power is $\tilde{P}_{g_i} = \mathcal{A}_w P_{g_i}$ with $w = 1$ s. Equations (6) and (26) are linear with constant coefficients, so $\mathcal{A}_w$ commutes with them and the balance that holds between measured

quantities is

$$C_g \frac{T_{g_i}(t) - T_{g_i}(t-w)}{w} = \tilde{P}_{g_i}(t) - \frac{\bar{T}_{g_i}(t) - \bar{T}_a(t)}{R_{ga}} - G_{gg} \sum_{j \neq i} \big(\bar{T}_{g_i}(t) - \bar{T}_{g_j}(t)\big), \tag{27}$$

with $\bar{T} = \mathcal{A}_w T$ and $\bar{T}_a = T_{\text{in}} + R_{\text{loop}}\big(\sum_i \tilde{P}_{g_i} + \sum_j \mathcal{A}_w P_{c_j}\big)$. The left side is the exact derivative of $\bar{T}_{g_i}$; this is the energy balance tested against the sensor's own box kernel, and it adds no parameter. Both w and C_g delay the temperature relative to the measured power, so they are confounded, and the estimates are repeated for $w \in \{0, 0.8, 1.2\}$ s, where $w = 0$ is the instantaneous form of Equation (6).

Bayesian Inverse PINN. Following Yang et al. (2021), the node temperatures of each identification window b are represented by a neural network, and the energy balance enters the likelihood as a residual evaluated on that network,

$$u_b(t) = m_b + s\,\mathrm{NN}_b\big(\boldsymbol{\varphi}(\tau)\big), \qquad \boldsymbol{\varphi}(\tau) = \big(\tau,\ \sin 2\pi f_j \tau,\ \cos 2\pi f_j \tau\big)_{j=1}^{32}, \tag{28}$$

where $\tau \in [0, 1]$ is the normalised time in the window, f_j are log-spaced from 0.5 to $f_{\max}$ cycles per window (Tancik et al., 2020), NN_b has two $\tanh$ layers of width d and one output per node, m_b is the window mean of the measurements and s their pooled within-window standard deviation. The time derivative of u_b is propagated exactly through the network in forward mode. The residual r_k is Equations (2)–(3) on MIT and Equation (27) on NLR, evaluated on u_b at every sample time t_k; on NLR, samples in the first w of a window have no history and carry no residual. Residuals are divided by a scale computed from the data only, the mean measured power S (per GPU on NLR); a scale computed from the current parameters would make the objective nearly insensitive to them (Appendix G). The likelihood is

$$\log p(\mathcal{D} \mid \mathbf{u}, \boldsymbol{\theta}, \sigma_R) = \eta_T \sum_k \log \mathcal{N}\big(T_k^{\text{obs}};\, u(t_k),\, \sigma_T^2\big) + \eta_R \sum_k \log \mathcal{N}\big(r_k/S;\, 0,\, \sigma_R^2\big). \tag{29}$$

The physics noise σ_R is inferred. The temperature noise σ_T is fixed, because with a point-estimated network a free σ_T is not identified: the network interpolates the samples and σ_T falls to the lower end of its range. On MIT, σ_T is the error on withheld samples (every fifth) of a fit of the network to the temperatures alone, made before the physics term is introduced. On NLR it is $1/\sqrt{12} \approx 0.29$ K, the quantisation noise of integer-degree readings.

Posterior. The physical parameters and σ_R receive a factorised Gaussian $q(\boldsymbol{\zeta}) = \prod_j \mathcal{N}(\mu_j, s_j^2)$ over the unconstrained coordinates, with the maps and ranges of Table 7 and the prior $\mathcal{N}(\mathbf{0}, 1.5^2\mathbf{I})$, whose median is the midpoint of each range (geometric for log-scaled, arithmetic for linear). The network weights Φ are a point estimate under a Gaussian prior, implemented as weight decay. The network and q are fitted jointly by maximising the evidence lower bound

$$\mathcal{E}(\boldsymbol{\mu}, \mathbf{s}, \Phi) = \mathbb{E}_q\Big[\log p\big(\mathcal{D} \mid \mathbf{u}_\Phi, \boldsymbol{\theta}(\boldsymbol{\zeta}), \sigma_R(\boldsymbol{\zeta})\big)\Big] - \mathrm{KL}\big(q \,\|\, p\big), \tag{30}$$

with the KL term at full weight. Because Φ is not sampled, the reported intervals exclude the uncertainty of the latent temperature path; this is a partial treatment compared with sampling the network weights (Yang et al., 2021).

Tempering. Residuals of neighbouring samples are correlated, so an i.i.d. likelihood over 10^5–10^6 residuals overstates the information in the data. After a burn-in with $\eta_T = \eta_R = 1$, we estimate the integrated autocorrelation time τ_{int} of each residual type with Geyer's initial positive sequence, removing only the global mean so that persistent per-window offsets count as correlation, and set $\eta = \max(1/\tau_{\text{int}}, 1/H)$, where H is the number of samples per window. The result is a tempered (power) posterior (Bissiri et al., 2016), whose effective sample size reflects the residual correlation. It does not reflect structural error in the model, which is assessed by the sensitivity analyses below. Both the i.i.d. and the tempered posteriors are reported.

Windows and surrogate capacity. The windows are those of the weak-form fit below. On MIT, $d \in \{16, 32, 48\}$ and $f_{\max} \in \{15, 30, 60\}$ cycles per window are chosen by the error on withheld samples of the temperature-only fit; the smallest network within 2% of the best is kept. On NLR, the

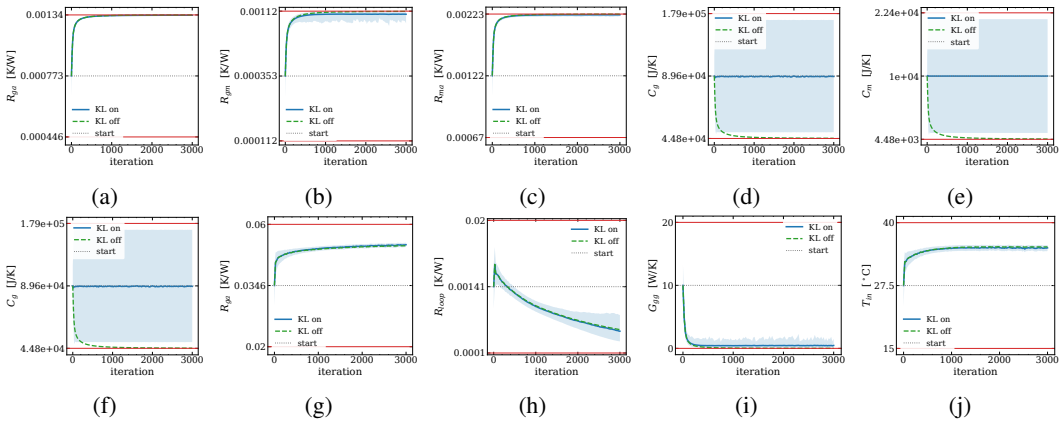


Figure 19: **Parameter convergence for the MIT Supercloud (first row) and NLR (second row) datasets' Inverse-PINN model.** Optimization histories for the aggregate thermal capacitances, C_g and C_m, and resistances, R_{ga}, R_{ma}, and R_{gm}. Solid blue curves show the posterior medians with KL regularization, shaded regions denote the corresponding 95% credible intervals, and dashed green curves show the KL-ablated solutions. Dotted gray lines indicate the initial values, while red lines mark the physically admissible bounds. The resistances converge near their upper bounds under both settings, whereas removing KL regularization drives the capacitances toward their lower bounds, revealing their weaker identifiability from the available data.

temperature follows the power within one or two samples, so withholding samples removes signal. Each 600-sample window is therefore represented by four networks of 150 samples, with $d = 32$ and $f_{\max}$ at the Nyquist frequency (74.5 cycles per window), and all samples enter the likelihood. Optimisation settings are given in Appendix G.

Weak-form check. As an estimate that uses no network, each balance is also integrated over sub-windows $\Omega_k = [a_k, b_k]$, which replaces the derivative by a measured temperature difference. For the MIT GPU node,

$$\epsilon_{g,k} = C_g\big[T_g(b_k) - T_g(a_k)\big] - \Delta t \sum_{t\in\Omega_k} \Big(\alpha(t)P(t) - \frac{T_g(t) - T_a}{R_{ga}} - \frac{T_g(t) - T_m(t)}{R_{gm}}\Big), \tag{31}$$

and the memory node and each NLR GPU are treated in the same way, with four sub-windows per window. Residuals are divided by S_0, the mean gross heat entering the nodes in a sub-window, computed from the data. The mean of $(\epsilon/S_0)^2$ is minimised under the same prior and ranges with a KL weight of $10^{-3}/n$, where n is the number of free parameters, so the result is a pseudo-posterior whose widths rank the parameters but are not calibrated. This fit treats the NLR power as instantaneous; the sensor delay then enters only at sub-window edges, but it is not modelled. On MIT the windows are 400 training-split windows of 240 min. On NLR they are 300 windows of 600 native samples from the first runs of each of three workloads (the two training workloads and online-rate inference); these are training runs, except for 20 validation windows in the online-rate case.

Identifiability diagnostics. For each parameter we report the posterior median, its position in the range, and the normalised width $\rho_j = \log(\theta_{j,97.5}/\theta_{j,2.5})/\log(h_j/\ell_j)$, or $(\theta_{j,97.5} - \theta_{j,2.5})/(h_j - \ell_j)$ for linear ranges. A value $\rho_j \approx 1$ means the posterior fills the range, and a median within 2% of either end of its range is reported as at the bound. The following comparisons separate information supplied by the data from the effect of the prior, the range and the model:

1. the i.i.d. and tempered widths of the Bayesian inverse PINN;
2. the medians of the Bayesian inverse PINN and of the weak-form fit, which use a strong-form and an integral residual (neither is a ground truth);
3. on NLR, the Bayesian inverse PINN with $w \in \{0, 0.8, 1, 1.2\}$ s;
4. the data error at the posterior median relative to σ_T; a value well below σ_T means that the network is fitting measurement noise;

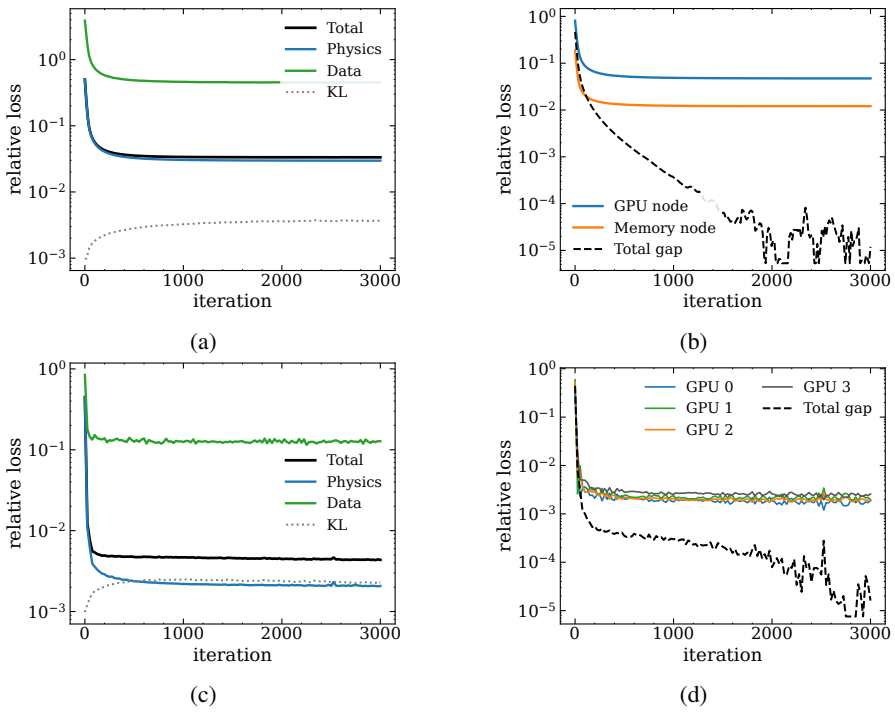
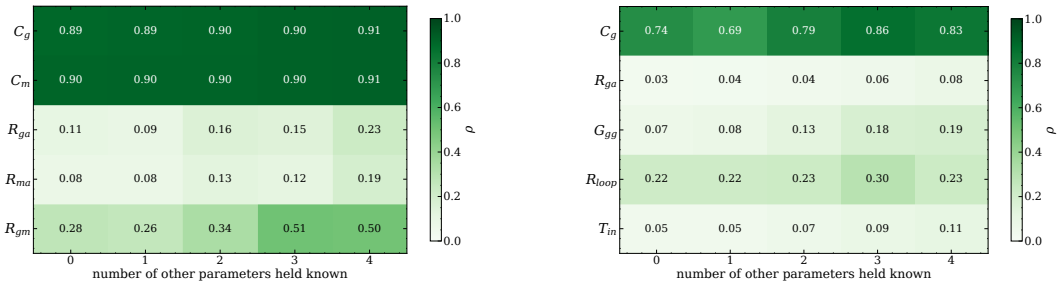


Figure 20: **Inverse PINN model convergence on the (a) and (b) MIT Supercloud and (c) and (c) NLR datasets.** (a) and (c) Total objective, physics residual, and KL regularization during the box-constrained fit. It is a forward-model diagnostic and does not enter the objective. (b) The physics residual is separated into GPU and memory energy-balance violations. The dashed curve shows the absolute distance of the total objective from its mean over the final 10% of iterations, indicating convergence. (d) The physics residual is separated into the four GPUs per node. All physics residuals use the corrected, fixed normalization scale.

5. for the weak-form fit, a refit without the KL term (comparing medians only, because without the KL term the widths collapse), the distance of each median from its initial midpoint, conditional slices of the objective across each range with the other parameters at the fitted point (not profile likelihoods), and refits with every subset of k parameters fixed at the $k = 0$ medians (31 fits for five parameters).

As a diagnostic outside the objective, the equations with the weak-form parameters are integrated by implicit Euler from each window's first measurement, and we report the one-step temperature error and, on MIT, the free-running error. All of these are computed on the identification windows and are therefore in-sample.



(a) (b)

Figure 21: **Conditional identifiability of (a) MIT and (b) NLR datasets' thermal parameters.** Each cell reports the median normalized width of the parameter's 95% credible interval, $\rho = \log(\theta_{97.5}/\theta_{2.5})/\log(\theta_{\max}/\theta_{\min})$, across fits in which the indicated number of other parameters is held known. Smaller ρ indicates stronger identification; $\rho \approx 1$ means that nearly the entire physically admissible range remains plausible. The thermal capacitances C_g and C_m remain weakly identified even when four other parameters are known, whereas the heat-transfer resistances are more tightly constrained by the MIT dataset measurements.