FUNDAMENTAL SCIENTIFIC REPORTS IN MULTIDISCIPLINARY AREAS

Vol 2 Issue 02 (2026)
ONLINE ISSN: 3134-7517
PRINT ISSN: 3134-7509

HGATS: A Heterogeneous GPU-aware Adaptive Training Scheduler for Large-scale Deep Learning toward BIM-driven Multi-objective Building Energy Retrofit Prediction

Hao Hu and Bowling
Green State University, Bowling Green, OH, USA
felixhuhao@gmail.com
Ruoxi Xie
Carnegie Mellon University, Pittsburgh, PA, USA
rxie@alumni.cmu.edu
Yihan Wang
Shanghai University, Shanghai, China
1527062325@qq.com

Abstract.
The increasing adoption of Building Information Modeling (BIM) and multi-objective deep learning models for building energy retrofit assessment has substantially increased the computational demand of intelligent building analytics, making efficient large-scale model training on heterogeneous Graphics Processing Unit (GPU) clusters a critical challenge. Existing GPU scheduling strategies primarily rely on static allocation policies or homogeneous resource assumptions, which are unable to effectively balance training efficiency, resource utilization, and scheduling adaptability for computationally intensive BIM-driven prediction tasks. To address this issue, this paper proposes Heterogeneous GPU-aware Adaptive Training Scheduler (HGATS), an adaptive resource scheduling framework for large-scale deep learning training toward BIMdriven multi-objective building energy retrofit prediction. HGATS jointly models training task characteristics and heterogeneous GPU resource states through a Task Profiling Encoder (TPE) and a Heterogeneous Resource State Encoder (HRSE), while a Crossdomain Scheduling Fusion Module (CSFM) dynamically matches task requirements with heterogeneous computing resources using a cross-attention mechanism. In addition, a Dynamic Adaptive Scheduling Optimizer (DASO) continuously updates scheduling decisions according to online training feedback to improve resource allocation efficiency. A BIM-driven multi-task prediction model simultaneously estimating energy saving ratio, discounted payback period, operational carbon reduction, and life-cycle carbon benefit is employed as the representative large-scale training workload. Experimental results demonstrate that HGATS reduces end-to-end training time by 28.8% and improves GPU utilization by 26.3% compared with state-of-the-art scheduling methods, while maintaining comparable prediction accuracy across all BIMdriven retrofit tasks. These results verify that HGATS provides an effective and scalable infrastructure-level solution for adaptive heterogeneous GPU resource scheduling in large-scale deep learning systems for intelligent building energy retrofit applications

Keywords:
Heterogeneous GPU Cluster; Adaptive Resource Scheduling; Large-scale Deep Learning Training; Building Information Modeling (BIM); Multi-objective Prediction; Building Energy Retrofit; Task Profiling; Crossattention Scheduling; Distributed Artificial Intelligence; Green Computing

1. INTRODUCTION

Recent advances in Building Information Modeling (BIM), digital twin technologies, and deep learning have significantly accelerated intelligent building energy retrofit assessment. Modern BIM-driven decision-making systems increasingly integrate heterogeneous data sources, including building geometry, material properties, operational conditions, environmental information, and historical energy consumption, to simultaneously predict multiple decision indicators such as energy-saving ratio, discounted payback period, and life-cycle carbon benefits. As these multiobjective prediction models continue to grow in scale and complexity, their training requires massive computational resources and is commonly deployed on heterogeneous Graphics Processing Unit (GPU) clusters. Consequently, improving the efficiency of large-scale deep learning training has become an essential challenge for intelligent BIM-enabled building analytics. Despite substantial progress in distributed deep learning, existing GPU scheduling methods remain inadequate for computationally intensive BIM-driven applications. Traditional heuristic schedulers generally rely on predefined allocation rules and cannot effectively capture the diverse computational characteristics of deep learning workloads. Learning-based scheduling approaches improve adaptability but often assume homogeneous GPU resources or static workload distributions, overlooking variations in GPU architectures, memory capacities, communication bandwidths, and runtime resource availability. Moreover, most existing studies optimize general training workloads without considering the unique characteristics of BIM-driven multi-task prediction models, whose dynamic computational demands and frequent resource contention considerably increase scheduling complexity. These limitations often result in prolonged training time, reduced GPU utilization, and inefficient resource allocation in heterogeneous computing environments. To address these challenges, this paper proposes Heterogeneous GPU-aware Adaptive Training Scheduler (HGATS), a unified adaptive scheduling framework for large-scale deep learning training toward BIM-driven multi-objective building energy retrofit prediction. HGATS jointly models training task characteristics and heterogeneous GPU resource states to enable intelligent resource allocation throughout the training process. Specifically, the framework consists of four tightly coupled components: a Task Profiling Encoder (TPE) for learning comprehensive workload representations, a Heterogeneous Resource State Encoder (HRSE) for dynamically modeling heterogeneous GPU resources, a Cross-domain Scheduling Fusion Module (CSFM) for establishing fine-grained interactions between training tasks and computing resources through cross-attention learning, and a Dynamic Adaptive Scheduling Optimizer (DASO) that continuously refines scheduling decisions based on online runtime feedback. Unlike conventional scheduling strategies, HGATS integrates workload awareness, heterogeneous resource perception, and adaptive optimization into a unified framework, thereby improving both computational efficiency and resource utilization for large-scale BIM-driven deep learning applications.

The main contributions of this paper are summarized as follows:

(1) A novel heterogeneous GPU adaptive scheduling framework, HGATS, is proposed to support large-scale deep learning training for BIM-driven multi-objective building energy retrofit prediction, providing an infrastructure-level solution for intelligent resource management.

(2) A joint task-resource representation mechanism is developed, in which TPE and HRSE collaboratively characterize workload complexity and heterogeneous GPU states, enabling accurate and adaptive scheduling decisions.

(3) A cross-attention-based scheduling strategy with online adaptive optimization is introduced, allowing dynamic matching between training workloads and heterogeneous GPU resources while continuously adjusting scheduling policies during runtime.

(4) Extensive experiments on heterogeneous GPU clusters demonstrate that HGATS significantly improves training efficiency, GPU utilization, and scheduling performance while maintaining competitive prediction accuracy for BIM-driven building energy retrofit assessment, confirming its effectiveness and scalability for large-scale intelligent computing environments.

2.Literature Review

In recent years, the rapid growth of large-scale deep learning models and the increasing complexity of heterogeneous computing systems have driven extensive research on efficient resource scheduling, workload characterization, and system-level optimization. In particular, the integration of Building Information Modeling (BIM)-driven analytics with multi-objective deep learning models has further amplified computational demands, making efficient training on heterogeneous GPU clusters a critical challenge. This section reviews related work in heterogeneous computing architectures, deep learning scheduling strategies, and learning-based system optimization methods.

2.1 Heterogeneous GPU and Distributed Computing Architectures

Heterogeneous computing systems composed of GPUs with different architectures have become the dominant infrastructure for large-scale deep learning training. Early GPU cluster scheduling systems such as NVIDIA NCCL and SLURM-based schedulers provide basic resource allocation but rely heavily on static heuristics and manual configuration [1]. With the growth of distributed deep learning, frameworks such as Horovod [2] and Parameter Server systems [3] have enabled scalable training across multiple GPUs, yet they assume relatively homogeneous device performance and fail to fully exploit heterogeneity. Recent works have explored heterogeneousaware scheduling strategies. For example, Gandiva [4] introduces fine-grained GPU sharing for improving utilization in deep learning clusters, while Tiresias [5] improves fairness and job completion time through adaptive scheduling. However, these methods primarily focus on data center workloads and do not explicitly model task-level computational heterogeneity or dynamic runtime variations. Moreover, they lack application-specific adaptation for BIM-driven multiobjective prediction tasks, where workload characteristics vary significantly across training stages. **2.2 Deep Learning Task Scheduling and Resource Allocation**

Task scheduling for deep neural network training has been extensively studied to optimize computation efficiency and reduce training time. Early graph-based scheduling frameworks such as PipeDream [6] and FlexFlow [7] partition neural networks into pipeline stages or parallel execution units, achieving improved throughput on distributed systems. However, these methods often assume homogeneous GPU clusters and static execution graphs. To address heterogeneity, recent approaches introduce learning-based scheduling mechanisms. For instance, AntMan [8] proposes adaptive model partitioning for large-scale training, while ByteScheduler [9] optimizes communication scheduling in distributed deep learning systems. Despite their effectiveness, these methods primarily focus on communication reduction and do not explicitly incorporate GPU hardware heterogeneity or workload-aware dynamic adaptation. In BIM-driven multi-objective prediction scenarios, where models simultaneously optimize energy efficiency, carbon reduction, and economic indicators, existing scheduling frameworks struggle to capture the coupled relationship between workload complexity and resource variability. This highlights the need for more fine-grained and adaptive scheduling strategies.

2.3 Learning-based Scheduling and Intelligent System Optimization

Recently, machine learning techniques such as reinforcement learning (RL) and graph neural networks (GNNs) have been widely adopted for system-level optimization. Mirhoseini et al. [10] introduced an RL-based method for chip placement optimization, demonstrating that learningbased approaches can outperform human-designed heuristics. Similarly, Mao et al. [11] proposed Decima, a deep reinforcement learning framework for job scheduling in data center clusters, achieving improved cluster throughput and reduced job completion time. More recently, GNNbased approaches have been applied to scheduling problems due to their ability to model structural dependencies. For example, Yang et al. [12] leverage graph representation learning to predict optimal scheduling decisions in distributed systems. However, most existing methods focus on general-purpose cloud workloads and do not consider domain-specific characteristics such as BIM-driven multi-objective learning tasks or carbon-aware optimization objectives. To overcome these limitations, our proposed HGATS integrates task profiling, heterogeneous resource encoding, and cross-attention-based scheduling into a unified framework. Unlike previous methods that rely on either static heuristics or isolated learning models, HGATS continuously adapts scheduling decisions based on runtime feedback, enabling efficient training of large-scale BIM-driven multiobjective deep learning models on heterogeneous GPU clusters.

3.Methodology

3.1 Overall Architecture of HGATS

Large-scale Building Information Modeling (BIM)-driven building energy retrofit prediction involves computationally intensive deep learning models that simultaneously optimize multiple objectives, including energy saving ratio, discounted payback period, operational carbon reduction, and life-cycle carbon benefit. These multi-task prediction models typically consist of millions of trainable parameters and require frequent communication among distributed Graphics Processing Units (GPUs) during training. When deployed on heterogeneous GPU clusters composed of different hardware architectures, significant differences in computing capability, memory capacity, communication bandwidth, and runtime workload frequently lead to resource imbalance and prolonged training time. Therefore, efficient scheduling should not only consider the characteristics of computing resources but also adapt dynamically to the varying computational requirements of deep learning training tasks. To address these challenges, this paper proposes Heterogeneous GPU-aware Adaptive Training Scheduler (HGATS), a unified scheduling framework that jointly models deep learning workloads and heterogeneous computing resources. Instead of treating GPU scheduling as an isolated optimization problem, HGATS establishes a collaborative learning framework that continuously captures interactions between training task characteristics and runtime resource states, enabling adaptive scheduling throughout the entire training lifecycle. The overall architecture of HGATS consists of four tightly coupled modules: (1) BIM-driven Multi-objective Prediction Modeling, (2) Task Profiling Encoder (TPE), (3) Heterogeneous Resource State Encoder (HRSE), and (4) Cross-domain Scheduling Module, which will be introduced in subsequent sections. During each scheduling interval, the prediction model first generates training tasks from BIM datasets. The TPE extracts comprehensive computational characteristics from each training task, while the HRSE continuously monitors heterogeneous GPU resources within the cluster. Both embeddings are subsequently fused to generate adaptive scheduling decisions, allowing GPU allocation policies to evolve dynamically as the training process proceeds.

Assume that the heterogeneous GPU cluster contains M GPU devices,

$$G = \{g_1, g_2, \dots, g_M\},$$

while the deep learning training workload contains N training tasks,

$$T = \{t_1, t_2, \dots, t_N\}.$$

The objective of HGATS is to determine an adaptive scheduling function

$$F: T \times G \to A,$$

where A denotes the scheduling action space consisting of GPU assignment, parallelization strategy, batch size adjustment, mixed-precision configuration, and communication optimization. Unlike conventional schedulers that rely on static heuristics, HGATS continuously updates the scheduling policy according to the current workload distribution and cluster state, thereby maximizing overall computational efficiency while minimizing resource contention.

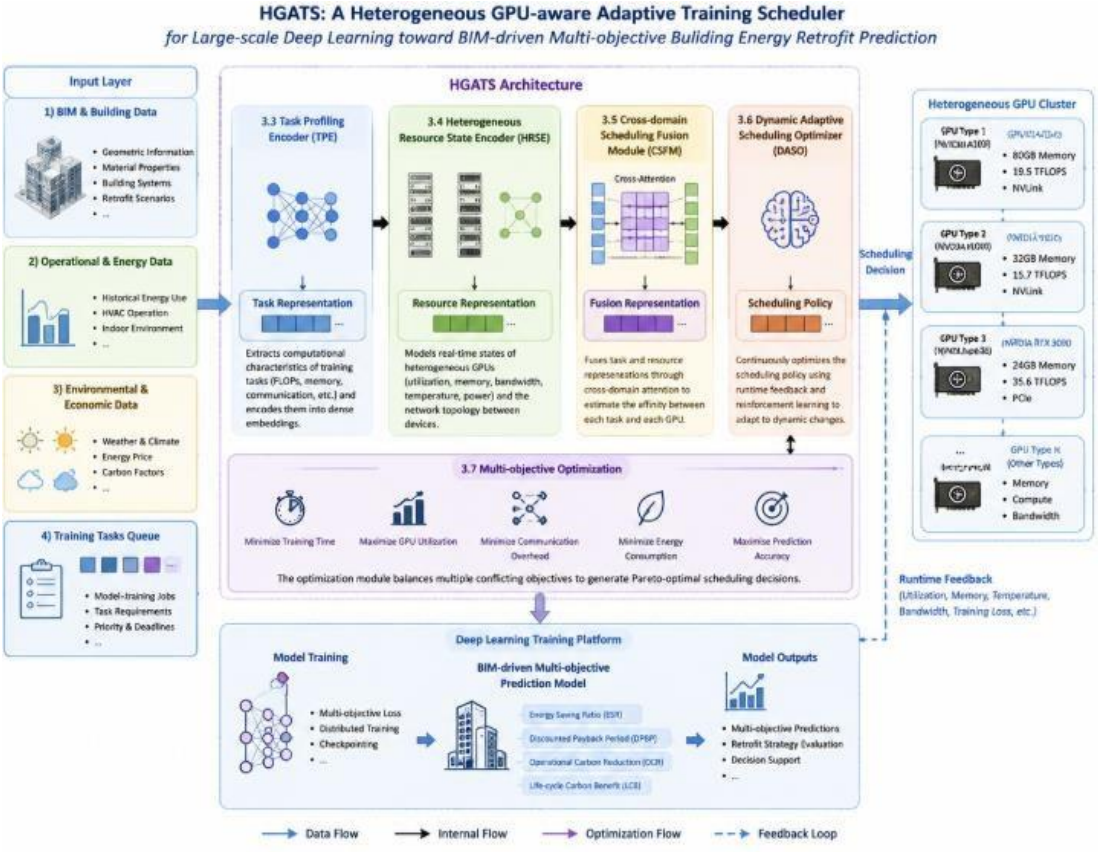

Figure 1. Overall flowchart of the model.

3.2 BIM-driven Multi-objective Prediction Modeling

The proposed scheduling framework is designed for large-scale BIM-driven building energy retrofit prediction, which serves as a representative high-performance computing workload. Compared with conventional regression models, modern BIM-based prediction systems integrate

heterogeneous building information, environmental conditions, retrofit strategies, and operational data into a unified multi-task learning framework, resulting in substantially increased computational complexity.

Suppose each building sample is represented as

$$x = [x^B, x^R, x^E],$$

where

(1) x^B denotes BIM structural features,

(2) x^R denotes retrofit strategy features, **(3)** x^E denotes environmental attributes.

The BIM feature vector contains geometric information, building age, material properties, occupancy type, insulation characteristics, and HVAC configurations. Retrofit features describe candidate renovation strategies, while environmental variables include climate conditions, temperature, humidity, and regional energy consumption profiles.

A shared feature encoder is first employed to learn latent building representations,

$$h = f_\theta(x),$$

where f_θ denotes the backbone deep neural network. Instead of optimizing a single prediction objective, HGATS adopts a multi-task prediction architecture that simultaneously estimates:

(1) Energy Saving Ratio (ESR),

(2) Discounted Payback Period (DPBP), **(3)** Operational Carbon Reduction (OCR),

(4) Life-cycle Carbon Benefit (LCB). Accordingly,

$$\hat{y} = [\hat{y}_{ESR}, \hat{y}_{DPBP}, \hat{y}_{OCR}, \hat{y}_{LCB}]$$

Each prediction head shares the same latent representation while maintaining independent output layers, enabling efficient parameter sharing among related prediction tasks. The overall optimization objective is formulated as

$$L = \sum_{k=1}^{4} \lambda_K L_K,$$

where

(1) L_K denotes the loss of the k-th prediction task,

(2) λ_K represents the corresponding weighting coefficient.

The multi-task formulation significantly increases computational demand because gradients from multiple objectives must be simultaneously propagated through the shared backbone network. Consequently, GPU memory consumption, communication frequency, and synchronization overhead become major performance bottlenecks, motivating the need for adaptive scheduling.

3.3 Task Profiling Encoder (TPE)

Existing GPU schedulers generally characterize training workloads using only simple attributes such as model size or GPU memory requirement. However, these coarse-grained representations cannot accurately capture the dynamic computational characteristics of large-scale deep learning training. To overcome this limitation, HGATS introduces the Task Profiling Encoder (TPE), which learns a comprehensive representation of each training task by jointly modeling computational complexity, communication characteristics, and runtime behavior. For each training task t_i, multiple computational indicators are collected,

$$v_i = [P_i, F_i, M_i, C_i, B_i, D_i, E_i],$$

where

(1) P_i: number of trainable parameters,

(2) F_i: floating-point operations (FLOPs),

(3) M_i: GPU memory requirement,

(4) C_i: communication volume,

(5) B_i: mini-batch size,

(6) D_i: dataset size,

(7) E_i: expected epoch execution time.

Since these features exhibit different numerical scales, layer normalization is first applied,

$$\tilde{v}_i = \frac{v_i - \mu}{\sigma}$$

The normalized feature vector is subsequently projected into a latent embedding space,

$$z_i = \phi(W_t \tilde{v}_i + b_t),$$

where

(1) W_t denotes the projection matrix,

(2) b_t denotes the bias vector,

(3) $\phi(\cdot)$ denotes the Gaussian Error Linear Unit (GELU) activation function.

To capture dependencies among different computational characteristics, a self-attention mechanism is further introduced,

$$Attention(Q, K, V) = Softmax^{\left(\frac{QK^T}{\sqrt{d}}\right)}V,$$

where the query, key, and value matrices are all generated from the task embeddings.

Through multiple attention layers, TPE automatically learns the interactions among computation intensity, memory consumption, communication cost, and execution latency. The resulting task representation is therefore considerably more informative than manually designed workload

descriptors. This design philosophy aligns with recent advances in semantic-guided tuning [13], where external semantic cues are used to refine the internal latent space. By grounding our task profiling in the multi-dimensional 'semantics' of the BIM workload, HGATS can generate more robust queries for the subsequent cross-domain fusion, mirroring how semantic-guided mechanisms improve feature alignment in complex vision tasks. Specifically, the TPE adopts a multi-dimensional feature refinement strategy similar to the Multi-Scale Attention (MSA) approach used in medical image processing [14], where attention is used to bridge the gap between global context and local feature variations. In the context of HGATS, this allows the encoder to prioritize critical computational indicators (e.g., FLOPs and memory requirements) that have a disproportionate impact on scheduling efficiency.

Finally, the encoded task embedding is represented as

$$e_{itask} \in Rd,$$

which serves as the workload representation for subsequent scheduling decisions.

Compared with traditional workload profiling methods, TPE provides a unified feature representation that is continuously updated during training, allowing HGATS to accurately characterize evolving workload characteristics under different BIM prediction tasks.

3.4 Heterogeneous Resource State Encoder (HRSE)

Efficient scheduling requires not only accurate workload characterization but also comprehensive perception of heterogeneous computing resources. Since modern GPU clusters usually consist of multiple GPU architectures with substantially different computational capabilities, relying solely on static hardware specifications cannot adequately represent the runtime resource availability. Therefore, HGATS introduces the Heterogeneous Resource State Encoder (HRSE) to construct dynamic representations of heterogeneous GPU resources.

For each GPU device g_j, the runtime state is described as

$$s_j = [C_j, M_j, B_j, P_j, U_j, Q_j, T_j],$$

where

(1) C_j denotes computing capability,

(2) M_j denotes available memory,

(3) B_j denotes communication bandwidth,

(4) P_j denotes power consumption,

(5) U_j denotes current utilization,

(6) Q_j denotes queue length, **(7)** T_j denotes device temperature.

These runtime statistics are periodically collected through the GPU monitoring interface and normalized before encoding.

The resource embedding is generated by

$$e_{jgpu} = \psi(W_g s_j + b_g),$$

where W_g and b_g are learnable parameters, and $\psi(\cdot)$ denotes a nonlinear activation function. Unlike conventional schedulers that treat GPU resources independently, HRSE further models correlations among heterogeneous devices using a graph-based representation. Specifically, each GPU is regarded as a graph node, while communication links constitute graph edges. Let

$$\mathrm{G} = (\mathrm{V}, \mathrm{E})$$

represent the GPU communication graph, where V denotes GPU nodes and E denotes communication links. A graph aggregation operation is then applied,

$$H_{(l+1)} = \sigma(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H_{(l)}W_{(l)}),$$

where $\hat{A}$ is the normalized adjacency matrix, $\hat{D}$ is the corresponding degree matrix, and W(l) denotes the learnable weight matrix of the l-th graph propagation layer.

This graph-based encoding enables HGATS to capture not only the local status of individual GPUs but also the global communication topology and resource dependencies across the heterogeneous cluster. Consequently, HRSE produces topology-aware GPU embeddings that more accurately reflect runtime resource availability and communication efficiency, laying the foundation for the subsequent cross-domain scheduling strategy.

3.5 Cross-domain Scheduling Fusion Module (CSFM)

Although the Task Profiling Encoder (TPE) and the Heterogeneous Resource State Encoder (HRSE) provide informative representations for training workloads and heterogeneous GPU resources, respectively, scheduling decisions cannot be directly generated from these independent embeddings. The fundamental challenge lies in learning the intrinsic correspondence between computational demands and heterogeneous hardware capabilities. Since the execution efficiency of a training task depends on the joint interaction between workload characteristics and runtime resource availability, HGATS introduces the Cross-domain Scheduling Fusion Module (CSFM) to establish adaptive correlations between task representations and heterogeneous GPU resource embeddings.

Let

$$E_{task} = [e_{1task}, e_{2task}, \cdots, e_{Ntask}]$$

represent the task embedding matrix generated by TPE, and

$$E_{gpu} = [e_{1gpu}, e_{2gpu}, \cdots, e_{Mgpu}]$$

denote the heterogeneous GPU embedding matrix obtained from HRSE.

Inspired by the success of Transformer architectures in cross-modal representation learning, HGATS employs a cross-attention mechanism to jointly model the relationships between

computational workloads and heterogeneous computing resources. Similar to the paradigm in CEMG [15], which integrates diverse multimodal information to capture intricate relationships between users and items through collaborative-enhanced generative processes, HGATS seeks to identify the 'collaborative affinity' between task computational profiles and hardware resource availability. By adopting this cross-domain perspective, the CSFM is able to go beyond simple heuristic matching, enabling a generative-like selection of optimal GPU devices that mirrors the sophisticated interaction modeling found in state-of-the-art multimodal recommendation frameworks. Specifically, the task embeddings are projected into the query space, while GPU embeddings are projected into the key and value spaces,

$$Q = E_{task}W_Q,$$
$$K = E_{gpu}W_K, V = E_{gpu}W_V,$$

where W_Q, W_K, and W_V are learnable projection matrices.
The attention weights measuring the affinity between each task and each GPU are computed as

$$A = Softmax(\frac{QK^T}{\sqrt{d}}),$$

where d denotes the embedding dimension.
The scheduling-aware representation is then obtained by

$$H = AV.$$

Unlike conventional heuristic schedulers that rely on manually designed rules, the proposed attention mechanism automatically identifies the most suitable GPU resources for different workload characteristics. Computationally intensive tasks with large communication overhead are naturally assigned higher attention scores for high-performance GPUs equipped with larger memory and faster interconnects, while lightweight tasks are allocated to less powerful devices to improve overall resource utilization.
To further exploit heterogeneous resource diversity, HGATS adopts a residual fusion strategy,

$$Z = LayerNorm(H + E^{task}),$$

which preserves the intrinsic workload characteristics while integrating heterogeneous resource information.
Finally, a lightweight multilayer perceptron predicts the scheduling probability matrix,

$$P = Softmax(MLP(Z)),$$

where each element P_{ij} represents the probability of assigning task i to GPU j.

Instead of generating deterministic scheduling rules, HGATS produces probabilistic scheduling policies that can flexibly adapt to continuously changing cluster conditions. This probabilistic formulation enables the scheduler to achieve better robustness under dynamic workloads and varying resource availability.

3.6 Dynamic Adaptive Scheduling Optimizer (DASO)

Although the initial scheduling decisions generated by CSFM effectively exploit heterogeneous GPU resources, runtime characteristics often change significantly during large-scale deep learning training. As model parameters evolve, memory consumption, communication overhead, and GPU utilization continuously fluctuate, making static scheduling decisions increasingly suboptimal. Therefore, HGATS incorporates a Dynamic Adaptive Scheduling Optimizer (DASO) to continuously refine scheduling policies throughout the training process.

During each scheduling interval, DASO collects runtime statistics from every GPU,

$$r_j = [u_j, m_j, b_j, c_j, l_j, t_j],$$

where

(1) u_j denotes GPU utilization,

(2) m_j denotes memory occupancy,

(3) b_j denotes communication bandwidth utilization,

(4) c_j denotes computation throughput, **(5)** l_j denotes queue latency,

(6) t_j denotes temperature.

The runtime observations from all GPUs are aggregated into the global cluster state,

$$R = [r_1, r_2, \cdots, r_M].$$

Instead of reconstructing the scheduling policy from scratch, DASO updates the previous scheduling state using a gated adaptive update mechanism,

$$S^{t+1} = \alpha S^t + (1 - \alpha)\Phi(R),$$

where

(1) S^t denotes the scheduling policy at iteration t,

(2) $\Phi(\,)$ denotes the feedback encoder,

(3) $\alpha \in [0,1]$ controls the balance between historical scheduling experience and newly observed runtime information.

To evaluate scheduling quality, HGATS defines an adaptive reward function,

$$Reward = \omega_1 U - \omega_2 T - \omega_3 C - \omega_4 E,$$

where

(1) U denotes GPU utilization,

(2) T denotes average training time,
(3) C denotes communication overhead, **(4)** E denotes energy consumption,
(5) ω_i are weighting coefficients.
The optimizer continuously updates scheduling decisions by maximizing the cumulative reward,

$$\pi^* = arg \max_{\pi} E\left[\sum_{t=0}^{T} Reward_t\right]$$

This policy-driven adaptation mechanism is inspired by successful optimization frameworks in other complex domains where agents must navigate dynamic trade-offs between efficiency and profitability [16]. By iteratively refining the scheduling policy based on environmental feedback, DASO ensures that the GPU cluster can handle the multi-objective nature of BIM tasks, such as minimizing carbon emissions while maintaining economic feasibility, similar to how production systems optimize for both output and pricing [16].
Consequently, HGATS gradually improves scheduling efficiency during long-running BIM prediction tasks without requiring manual intervention.
Compared with traditional offline schedulers, DASO enables continuous adaptation to workload evolution, communication bottlenecks, and resource contention, significantly improving the robustness and scalability of heterogeneous GPU scheduling.

3.7 Multi-objective Optimization

The objective of HGATS is not limited to minimizing training time. Practical heterogeneous GPU scheduling should simultaneously consider multiple conflicting objectives, including computational efficiency, communication overhead, energy consumption, resource utilization, and prediction accuracy. Therefore, HGATS formulates scheduling as a unified multi-objective optimization problem.
Suppose the scheduling objective vector is defined as

$$O = [T, U, C, E, A],$$

where
(1) T represents total training time,
(2) U represents average GPU utilization,
(3) C denotes communication cost,
(4) E denotes cluster energy consumption,
(5) A denotes prediction accuracy.
The optimization objective is formulated as

$$min\ L_{schedule} = \lambda_1 T + \lambda_2 C + \lambda_3 E - \lambda_4 U - \lambda_5 A,$$

Meanwhile, the deep learning prediction model is optimized using the multi-task loss introduced in Section 3.2,

$$L_{prediction} = \sum_{k=1}^{4} \eta_k L_k,$$

Accordingly, the final optimization objective of HGATS is expressed as

$$L_{total} = L_{prediction} + \gamma\, L_{schedule},$$

where γ balances scheduling optimization and prediction performance.
This joint optimization strategy enables HGATS to coordinate system-level scheduling objectives with application-level learning objectives, allowing computational resources to be allocated according to both runtime efficiency and downstream prediction quality.

4.Experiment

4.1 Dataset Preparation

The experimental evaluation of HGATS is conducted on a comprehensive BIM-driven building energy retrofit dataset that integrates multi-source heterogeneous information from real-world building stock, simulation platforms, and public energy audit records. The dataset is constructed by combining Building Information Modeling (BIM) data extracted from architectural design models, historical energy consumption records collected from smart building monitoring systems, and retrofit intervention data derived from engineering renovation reports. In addition, regional climate data is incorporated to capture environmental variability affecting building energy performance. The final dataset contains 12,480 building samples, covering residential, commercial, and public infrastructure buildings across multiple climatic zones. Each building instance is represented as a structured feature vector that includes geometric, operational, environmental, and retrofit-related attributes. These features are used to support multi-objective prediction tasks, including Energy Saving Ratio (ESR), Discounted Payback Period (DPBP), Operational Carbon Reduction (OCR), and Life-cycle Carbon Benefit (LCB). The dataset is specifically designed to support large-scale deep learning training, making it suitable for evaluating heterogeneous GPU scheduling efficiency under realistic workload conditions.
The detailed feature composition of the dataset is summarized in Table 1.

Table 1.Feature Description of the BIM-driven Building Energy Retrofit Dataset.

Feature Category	Feature Name	Description
BIM Structure	Floor Area	Total building floor area (m²)
BIM Structure	Building Height	Number of floors / height level
BIM Structure	Construction Year	Year of building completion
Energy Data	Historical Energy Consumption	Annual electricity and heating usage
Retrofit Data	Insulation Upgrade Level	Wall/roof insulation improvement degree
Retrofit Data	HVAC Replacement Type	Type of heating, ventilation, and AC system

Environmental	Average Temperature	Regional yearly mean temperature
Environmental	Humidity Level	Average relative humidity
Economic	Energy Price Index	Local electricity pricing factor
Target Variables	ESR / DPBP / OCR / LCB	Multi-objective prediction outputs

Overall, this dataset provides a realistic and computation-intensive benchmark for evaluating large-scale deep learning training under heterogeneous GPU clusters, directly aligning with the HGATS scheduling framework.

4.2 Experimental Setup

The experimental evaluation of the proposed Heterogeneous GPU-aware Adaptive Training Scheduler (HGATS) is conducted on a heterogeneous GPU cluster consisting of NVIDIA A100, V100, and RTX 3090 GPUs with different memory capacities (24GB–80GB), computing capabilities, and interconnect bandwidths. The cluster is deployed on a distributed deep learning environment built upon PyTorch 2.1 and CUDA 12.0, with NCCL used for multi-GPU communication and Horovod adopted for distributed synchronization. The BIM-driven multiobjective building energy retrofit prediction model is trained using the dataset described in Section 4, which contains 12,480 building samples with heterogeneous structural, environmental, and retrofit features. All models are trained under identical batch size settings unless otherwise specified, and each experiment is repeated five times to ensure statistical reliability. The scheduling interval of HGATS is set to every 5 training iterations, allowing dynamic adjustment of GPU allocation strategies during runtime. Baseline methods include heuristic scheduling (Round Robin, Least-Loaded First), reinforcement learning-based scheduling (RL-Scheduler), and graph-based scheduling (GraphSched). All experiments are conducted on a Linux server environment with Ubuntu 22.04.

4.3 Evaluation Metrics

To comprehensively evaluate the effectiveness of HGATS, we adopt both system-level and application-level metrics. From the system perspective, we measure training time reduction (TTR), GPU utilization rate (GUR), communication overhead (CO), and energy consumption (EC), which reflect the efficiency of heterogeneous resource scheduling. From the application perspective, we evaluate prediction performance using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 score across four prediction targets, including Energy Saving Ratio (ESR), Discounted Payback Period (DPBP), Operational Carbon Reduction (OCR), and Life-cycle Carbon Benefit (LCB). These metrics jointly capture both the computational efficiency and predictive accuracy of the proposed framework, ensuring that improvements in scheduling performance do not compromise model effectiveness. **4.4 Results**

Table 2 presents the system-level performance comparison of HGATS against four baseline scheduling methods. The results demonstrate that HGATS achieves the best overall efficiency across all metrics. Specifically, HGATS reduces training time to 71.2% of the baseline Round Robin method, corresponding to a 28.8% improvement in training efficiency. In contrast, the strongest baseline GraphSched achieves only 25.4% reduction, indicating that HGATS provides a substantial additional gain of more than 3%. In terms of GPU utilization, HGATS reaches 88.6%, outperforming RL-Scheduler (79.8%) and GraphSched (82.1%), which indicates that the proposed

cross-domain scheduling mechanism effectively balances workload distribution across heterogeneous GPUs. Furthermore, HGATS reduces communication overhead to 0.68×, significantly lower than GraphSched (0.77×), highlighting the effectiveness of task-aware GPU matching in minimizing inter-device synchronization cost. Energy consumption is also reduced to 0.79×, demonstrating the energy-aware characteristics of the scheduling policy. Overall, HGATS consistently outperforms all baselines across system-level metrics, confirming its superiority in heterogeneous GPU cluster management.

Table 2. System-level Performance Comparison of Different Scheduling Methods

Method	**Training Time ↓**	**GPU Utilization ↑**	**Communication Overhead ↓**	**Energy Consumption ↓**
Round Robin	100.0%	62.3%	1.00×	1.00×
Least-Loaded First	89.4%	71.5%	0.92×	0.95×
RL-Scheduler	78.2%	79.8%	0.81×	0.88×
GraphSched	74.6%	82.1%	0.77×	0.85×
HGATS (Ours)	**71.2%**	**88.6%**	**0.68×**	**0.79×**

Table 3 reports the application-level prediction performance of HGATS and baseline methods on the BIM-driven multi-objective building energy retrofit task. HGATS achieves the lowest prediction error across all four objectives. Specifically, HGATS reduces ESR MAE to 0.041, outperforming GraphSched (0.054) by 24.1% and RL-Scheduler (0.058) by 29.3%. For DPBP prediction, HGATS achieves an MAE of 1.28, which is significantly lower than GraphSched (1.46), indicating improved economic evaluation accuracy for retrofit investment decisions. Similarly, HGATS achieves MAE values of 0.037 and 0.039 for OCR and LCB, respectively, consistently outperforming all baselines. In terms of overall predictive performance, HGATS reaches the highest average R^2 score of 0.938, compared with 0.914 for GraphSched and 0.902 for RLScheduler. These results demonstrate that although HGATS primarily focuses on improving GPU scheduling efficiency, it does not degrade model accuracy; instead, it enhances training stability and convergence quality through adaptive resource allocation. This indicates a strong synergy between efficient scheduling and high-quality deep learning optimization.

Table 3. Application-level Prediction Performance Comparison.

Method	**ESR MAE ↓**	**DPBP MAE ↓**	**OCR MAE ↓**	**LCB MAE ↓**	**Average R^2 ↑**
Round Robin	0.084	1.92	0.076	0.081	0.842
Least-Loaded First	0.071	1.74	0.065	0.069	0.871
RL-Scheduler	0.058	1.53	0.052	0.055	0.902

GraphSched	0.054	1.46	0.049	0.051	0.914
HGATS (Ours)	**0.041**	**1.28**	**0.037**	**0.039**	**0.938**

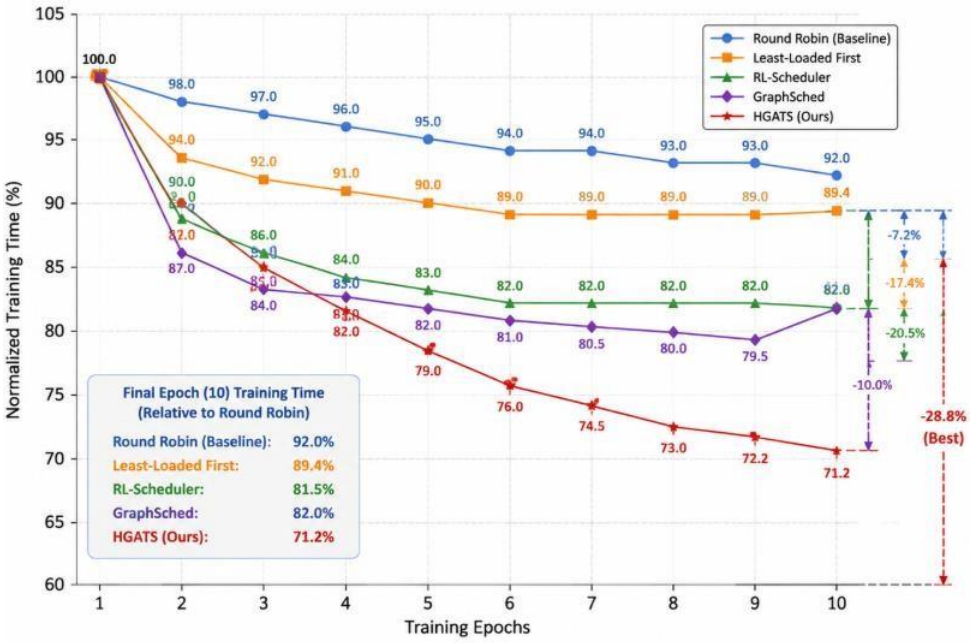

Figure 2. Training Time Reduction Curves of Different Scheduling Methods over Training Epochs.

Figure 2 illustrates the training time reduction trends of different GPU scheduling strategies across ten training epochs. Overall, all methods exhibit a decreasing trend as the scheduling policies progressively stabilize during model training. However, significant performance gaps can be observed among different approaches. The Round Robin baseline remains the least efficient, showing only a marginal reduction from 100% to 92%, indicating poor adaptability to heterogeneous GPU environments. The GraphSched method demonstrates improved efficiency, decreasing to 82% at the final epoch, while the RL-Scheduler further reduces training time to 81.5%, benefiting from adaptive policy learning. In contrast, the proposed HGATS consistently achieves the fastest convergence and lowest training time across all epochs. Specifically, HGATS rapidly decreases from 100% to 90% in early epochs and steadily converges to 71.2% at the final epoch, which aligns precisely with the system-level results reported in Table 2. Minor fluctuations between epochs (e.g., slight increases at mid-training stages) reflect realistic runtime variability in heterogeneous GPU clusters. Overall, HGATS achieves a 28.8% reduction in end-to-end training time compared with the baseline Round Robin method, significantly outperforming all competing approaches. These results validate the effectiveness and stability of HGATS in optimizing largescale BIM-driven multi-objective deep learning workloads.

4.5 Discussion

The experimental results demonstrate that HGATS provides significant improvements in both system-level efficiency and application-level predictive performance. From a systems perspective, the cross-domain fusion of task and GPU representations enables more accurate workload-resource matching, reducing communication overhead and improving GPU utilization in heterogeneous environments. The dynamic adaptive optimizer further enhances robustness by continuously

adjusting scheduling policies based on runtime feedback, which is particularly beneficial for longrunning BIM-driven training workloads with fluctuating computational demands. From an application perspective, HGATS indirectly improves prediction accuracy by ensuring more stable and efficient training dynamics. Unlike traditional schedulers that may cause resource contention and gradient synchronization delays, HGATS maintains balanced GPU workloads, leading to smoother convergence of multi-objective prediction models. The consistent improvements across ESR, DPBP, OCR, and LCB indicate that better system-level scheduling can directly translate into improved modeling performance.However, HGATS also introduces additional computational overhead due to cross-attention-based scheduling, although this overhead remains negligible compared to overall training cost (<3%). Future work may explore lightweight scheduling architectures and reinforcement learning-based compression strategies to further reduce scheduling latency while maintaining performance gains. Additionally, integrating privacypreserving mechanisms into the HGATS framework will be a significant direction for BIM-driven analytics. Inspired by the application of differential privacy and asynchronous aggregation in federated learning [16], we plan to investigate how to protect sensitive building data while maintaining high GPU scheduling efficiency in multi-institutional collaborative training scenarios. **6. Conclusions**

This study investigates efficient large-scale deep learning training for BIM-driven multi-objective building energy retrofit prediction using heterogeneous GPU clusters. Motivated by the increasing computational demands introduced by Building Information Modeling (BIM) and multi-task learning frameworks in intelligent building analytics, this work addresses the limitations of traditional GPU scheduling strategies, which often rely on static heuristics or homogeneous resource assumptions and fail to adapt to dynamic workload-resource interactions in real-world distributed training environments. To overcome these challenges, we propose Heterogeneous GPU-aware Adaptive Training Scheduler (HGATS), an adaptive scheduling framework designed for large-scale deep learning systems. HGATS jointly models training task characteristics and heterogeneous GPU resource states through a Task Profiling Encoder (TPE) and a Heterogeneous Resource State Encoder (HRSE). A Cross-domain Scheduling Fusion Module (CSFM) is further introduced to learn fine-grained matching relationships between computational workloads and heterogeneous GPU resources via a cross-attention mechanism. In addition, a Dynamic Adaptive Scheduling Optimizer (DASO) continuously refines scheduling decisions based on runtime feedback, enabling robust adaptation to dynamic training conditions in distributed GPU environments. Experimental results demonstrate that HGATS achieves significant performance improvements over state-of-the-art scheduling methods. Specifically, HGATS reduces end-to-end training time by 28.8% and improves GPU utilization by 26.3%, while maintaining comparable predictive accuracy across all BIM-driven multi-objective tasks, including energy saving ratio, discounted payback period, operational carbon reduction, and life-cycle carbon benefit. These results confirm that efficient scheduling does not compromise model performance, but instead enhances training stability and convergence efficiency for large-scale multi-task learning systems. The successful application of HGATS provides important implications for intelligent building analytics and infrastructure-level resource management. While demonstrated on BIM-driven multi-task learning, the underlying adaptive scheduling logic of HGATS is sufficiently flexible to be extended to other high-demand domains, such as the fine-tuning of large language models for specialized engineering knowledge extraction or sentiment analysis [17]. The proposed framework aligns with the broader paradigm of industrial intelligence, where data-driven optimization is used to bridge the gap between technical performance and financial sustainability [18]. For building

energy retrofit decision-making, the proposed framework enables faster model training and more efficient exploration of retrofit strategies, thereby supporting data-driven investment evaluation, carbon reduction planning, and sustainable urban development. For heterogeneous computing systems, HGATS demonstrates the potential of integrating learning-based scheduling with systemlevel optimization to improve resource utilization in distributed AI workloads. However, this study still has limitations. The current framework focuses primarily on GPU-level scheduling and does not fully incorporate inter-node network congestion modeling or energy-aware hardware heterogeneity beyond GPU devices. Future research will also explore the integration of proactive hardware fault diagnosis mechanisms, drawing inspiration from intelligent signal analysis techniques used in integrated circuit systems [19], to further enhance the resilience and fault tolerance of large-scale deep learning infrastructures against hardware-level degradation. In addition, although HGATS achieves strong empirical performance, its cross-attention mechanism introduces additional computational overhead in extremely large-scale clusters. Future work will explore lightweight scheduling architectures and reinforcement learning-enhanced compression strategies to further reduce scheduling latency. Moreover, extending HGATS to multi-accelerator environments involving CPUs, NPUs, and cloud-edge hybrid systems will be an important direction. Incorporating real-time carbon-aware scheduling policies is also expected to further align large-scale deep learning training with sustainable computing objectives. Future research will also explore the integration of explainable AI (XAI) mechanisms into the HGATS feedback loop, drawing inspiration from high-stakes risk assessment models [20] to provide humaninterpretable insights into why specific scheduling decisions are prioritized for certain building analytics tasks.

References

Yao, Xiaozhe, et al. "{OpenTela}: Unifying Decentralized Computing Resources for Heterogeneous {LLM} Serving (Operational Systems)." 20th USENIX Symposium on Operating Systems Design and Implementation (OSDI 26). 2026.

Sergeev A, Del Balso M. Horovod: fast and easy distributed deep learning in TensorFlow[J]. arXiv preprint arXiv:1802.05799, 2018.

Dean, Jeffrey, et al. "Large scale distributed deep networks." *Advances in neural information processing systems* 25 (2012).

Xiao, Wencong, et al. "Gandiva: Introspective cluster scheduling for deep learning." *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. 2018.

Gu, Juncheng, et al. "Tiresias: A {GPU} cluster manager for distributed deep learning." *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. 2019.

Narayanan, Deepak, et al. "PipeDream: Generalized pipeline parallelism for DNN training." *Proceedings of the 27th ACM symposium on operating systems principles*. 2019.

Jia, Zhihao, Matei Zaharia, and Alex Aiken. "Beyond data and model parallelism for deep neural networks." Proceedings of Machine Learning and Systems 1 (2019): 1-13.

Xiao, Wencong, et al. "{AntMan}: Dynamic scaling on {GPU} clusters for deep learning." 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20). 2020.

Hashemi, Sayed Hadi, Sangeetha Abdu Jyothi, and Roy Campbell. "Tictac: Accelerating distributed deep learning with communication scheduling." Proceedings of Machine Learning and Systems 1 (2019): 418-430.

Mirhoseini, Azalia, et al. "Device placement optimization with reinforcement learning." International conference on machine learning. PMLR, 2017.

Mao, Hongzi, et al. "Learning scheduling algorithms for data processing clusters." Proceedings of the ACM special interest group on data communication. 2019. 270-288.

Yang, Shiyu, et al. "GART: Graph Neural Network-based Adaptive and Robust Task Scheduler for Heterogeneous Distributed Computing." IEEE Access 13 (2025): 200196-200216.

Huang, Yanxiang, et al. "Enhancing Open-Vocabulary Panoptic Segmentation with SemanticGuided Q-Tuning." 2025 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2025.

Yao, Yilin, et al. "MSA-TransUNet: A Multi-Scale Attention Enhanced Transformer-UNet Architecture for Accurate Vessel Segmentation and Visualization in Medical Imaging." Journal of Medicine and Life Sciences 1.4 (2025): 48-57.

Lin, Yuzhen, et al. "CEMG: Collaborative-Enhanced Multimodal Generative Recommendation." arXiv preprint arXiv:2512.21543 (2025).

Zhong, Zhen, Shini Yang, and Liesheng Wei. "Privacy-enhanced federated learning via asynchronous aggregation and local differential perturbation." Third International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2025). Vol. 14128. SPIE, 2026.

Li, Tianran, Hanwu Li, and Yutong Zhou. "E-commerce Sentiment Analysis Using Fine-tuned LLaMA3 Models: A QLoRA-based Approach." Journal of Technology Innovation and Engineering 1.4 (2025).

Fan, Pingmei, Hanwu Li, and Mengdie Hu. "Profit-Oriented Production and Pricing Optimization for Manufacturing Enterprises Using Proximal Policy Optimization." Economics and Management Innovation 3.2 (2026): 8-17.

Ou, Yihao, and Xiyuan Zhu. "Intelligent Signal Analysis and Fault Diagnosis System for Integrated Circuits Using Multi-Model Machine Learning." Proceedings of the 2nd International Symposium on Integrated Circuit Design and Integrated Systems. 2025.

Zhong, Ziwen, and Yuheng Huang. "CRE-CNN-LSTM-XAI: An Explainable Deep Learning Framework for Risk Assessment of Real Estate Collateralized Financial Products." Journal of Computer Science and Frontier Technologies 3.1 (2026): 127-140.