A Predict-then-Schedule framework for Power Distribution Networks with AI Data Centers

Siqi Yan, Jiebao Zhang, Xi Yao, Juan Huang and Ye Shi

***Abstract*—The surge of GPU-intensive workloads in artificial intelligence (AI) data centers drives massive energy demands, leading to soaring costs and significant stress on local power distribution networks. Coordinating delay-tolerant workload scheduling with power grid conditions via precise workload prediction can mitigate these issues. However, a critical gap remains in conventional approaches, i.e., minimizing prediction error does not necessarily lead to minimized downstream operational loss. Hence, this paper proposes an end-to-end Predict-Then-Schedule (PTS) framework that integrates upstream workload prediction with downstream scheduling optimization. By leveraging differentiable convex optimization, the PTS framework maps input features directly to optimal scheduling and enables gradient-based training. Furthermore, to respect the data center's capacity, a workload over-shifted loss combining electricity cost with a penalty for load-shedding is introduced to evaluate scheduling quality. Experiments demonstrate that the proposed framework significantly reduces operational cost and enhances system security compared to the conventional two-stage baseline.**

***Index Terms*—AI Data Center, Power Distribution Network, Workload Prediction, Predict-then-Schedule(PTS)**

I. Introduction

The explosive growth of GPU-intensive workloads, such as deep learning jobs, has driven the expansion of artificial intelligence (AI) data centers [1], [2]. The large and fluctuating power demand of AI data centers poses significant challenges to distribution grids, leading to higher electricity costs [3], [4] and increasing the risk of grid instability due to capacity constraints [5]. A promising strategy has emerged to coordinate delay-tolerant workloads with power grid conditions, utilizing their inherent spatio-temporal flexibility to reduce operational costs and enhance stability [6], [7]. However, this coordination strategy is critically dependent on precise workload forecasts [8]. Inaccurate predictions can lead to severe consequences, including high operational costs, energy waste, and distribution power grid instability [9], [10].

The predict-then-optimize (PTO) framework [11], [12] aims to avoid the consequences arising from prediction bias. Unlike the conventional two-stage method, which treats prediction and optimization as two separate, sequential procedures, the PTO framework integrates the downstream optimization problem into the training loop of the upstream predictor [13]. This allows the predictor to produce forecasts that are specifically tailored to high-quality downstream decisions [14].

In this paper, we propose an end-to-end, predict-then-schedule (PTS) framework that incorporates the downstream workload scheduling optimization directly into the upstream workload prediction training pipeline. We compute the gradients of the optimal transfer matrix with respect to the upstream predictor's parameters, inspired from differentiable convex programming. Furthermore, we introduce a workload over-shifted loss, which combines power cost and a penalty term of load-shedding variables. This loss evaluates the quality of downstream decisions, thereby guiding prediction results to satisfy physical constraints of power distribution networks and AI data centers.

Our contributions are summarized as follows: 1) To the best of our knowledge, we are the first to integrate downstream workload scheduling optimization into the training of workload prediction models in an end-to-end differentiable manner. Unlike the conventional two-stage method, the proposed framework leverages the sensitivity of scheduling decisions on upstream predictors, enabling workload forecasts that are aware of scheduling quality. 2) We propose a novel workload over-shifted loss that jointly captures electricity cost and a penalty for workload shedding, thereby evaluating scheduling decision quality in terms of both economy and system security. Case studies demonstrate that the proposed approach yields lower operational costs and higher constraint satisfaction compared to the conventional two-stage baseline.

II. Model Formulation

This section introduces the mathematical models for the power distribution network and the AI data center, and formulates a grid-coordinated spatio-temporal workload scheduling optimization problem.

A. Power Distribution Network Modeling

We consider a multi-period scheduling horizon denoted by the set $\mathcal{T} = \{1, \ldots, T\}$, where each time step $t \in \mathcal{T}$ represents a discrete interval with a resolution of $\Delta t = 1$ hour. We model the power distribution network using the linearized DistFlow model [15]. In this framework, the AI datacenters are treated as specialized, controllable loads. Each datacenter is connected to a specific bus i within the distribution network, denoted as $i \in \mathcal{N}_{\mathrm{DC}}$. The core of the grid model is to maintain power

Siqi Yan and Jiebao Zhang contributed equally to this work. Corresponding authors: Ye Shi and Xi Yao.

Siqi Yan, Jiebao Zhang and Ye Shi are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China. (email:yansq2024, zhangjb2023, shiye@shanghaitech.edu.cn)

Xi Yao and Juan Huang are with China Mobile Shanghai ICT Co., Ltd. (email: yaoxi, huangjuan@cmsr.chinamobile.com)

This work was supported by National Natural Science Foundation of China (62303319), ShanghaiTech AI4S Initiative SHTAI4S202404, HPC Platform of ShanghaiTech University, and MoE Key Laboratory of Intelligent Perception and Human-Machine Collaboration (ShanghaiTech University).

979-8-3315-8138-1

balance at each bus. This is expressed through the active and reactive power balance constraints (1) and (2):

$$\sum_{g\in\Omega_i^G} P_{i,t}^g - \sum_{j\in\Omega_i^{\text{out}}} P_{ij,t}^l + \sum_{j\in\Omega_i^{\text{in}}} P_{ji,t}^l = \begin{cases} P_{i,t}^{\text{DC}}, \text{if } i \in \mathcal{N}_{\text{DC}}, \\ P_{i,t}^d, \text{otherwise}, \end{cases} \tag{1}$$

$$\sum_{g\in\Omega_i^G} Q_{i,t}^g - \sum_{j\in\Omega_i^{\text{out}}} Q_{ij,t}^l + \sum_{j\in\Omega_i^{\text{in}}} Q_{ji,t}^l = \begin{cases} 0, \text{if } i \in \mathcal{N}_{\text{DC}}, \\ Q_{i,t}^d, \text{otherwise}, \end{cases} \tag{2}$$

where $P_{i,t}^g$ and $Q_{i,t}^g$ are the active and reactive power generation, while $P_{ij,t}^l$ and $Q_{ij,t}^l$ are the line power flows. The terms $P_{i,t}^d$ and $Q_{i,t}^d$ represent the conventional active and reactive background loads at bus i, respectively. (1) and (2) couple the total power consumption $P_{i,t}^{\text{DC}}$ of the AI datacenter into the power balance equation at bus i.

The model is completed by the voltage drop constraint (3) and physical limit constraints (4)-(5), which enforce limits on bus voltages and power generation:

$$V_{i,t} - V_{j,t} = 2(r_{ij} \cdot P_{ij,t}^l + x_{ij} \cdot Q_{ij,t}^l), \tag{3}$$

$$V_{0,t} = v_r^2, \quad \underline{v}_i^2 \le V_{i,t} \le \overline{v}_i^2, \tag{4}$$

$$\underline{P}^g \le P_{i,t}^g \le \overline{P}^g, \quad \underline{Q}^g \le Q_{i,t}^g \le \overline{Q}^g. \tag{5}$$

B. AI Datacenter Power Consumption Modeling

The power consumption of an AI datacenter is primarily driven by its IT equipment load, $P_{i,t}^{\text{IT}}$, which represents the power consumed by servers and GPUs processing computational tasks. This IT load has a distinct operational range, physically bounded by the cluster's idle power P_i^{idle} and its maximum power capacity P_i^{full}:

$$P_i^{\text{idle}} \le P_{i,t}^{\text{IT}} \le P_i^{\text{full}}, \quad \forall i \in \mathcal{N}_{\text{DC}}. \tag{6}$$

However, $P_{i,t}^{IT}$ only accounts for the computational hardware. The total electricity demand drawn from the power grid, $P_{i,t}^{\text{DC}}$, must also account for auxiliary systems such as cooling, lighting, and power distribution units. We utilize the standard Power Usage Effectiveness (PUE) metric R_{PUE}, to model this relationship and map the IT load to the total cluster demand:

$$P_{i,t}^{\text{DC}} = R_{\text{PUE}} \cdot P_{i,t}^{\text{IT}}, \quad \forall i \in \mathcal{N}_{\text{DC}}. \tag{7}$$

This variable, $P_{i,t}^{\text{DC}}$, represents the total load that the datacenter imposes on the power grid, as incorporated in the power balance equation (1).

C. AI Datacenter Spatio-temporal Workload Scheduling

Many tasks in AI datacenters, such as deep learning jobs, are delay-tolerant, which endows their workloads with inherent spatio-temporal flexibility. We model this scheduling capability via a transfer matrix M. An element $M_{j,i,s,t}$ represents the percentage of workload that arrived at datacenter j at time s and is scheduled for execution at datacenter i at time t. This decision matrix must adhere to operational constraints:

$$M_{j,i,s,t} \in [0,1], \quad \forall i,j \in \mathcal{N}_{\text{DC}},\ s,t \in \mathcal{T}, \tag{8a}$$

$$\sum_{i\in\mathcal{N}_{\text{DC}}} \sum_{t\in\mathcal{T}} M_{j,i,s,t} = 1, \quad \forall j \in \mathcal{N}_{\text{DC}},\ s \in \mathcal{T}, \tag{8b}$$

$$M_{j,i,s,t} = 0, \quad \text{if } t < s, \tag{8c}$$

(8b) enforces load conservation, ensuring all submitted tasks are dispatched, while (8c) ensures temporal feasibility. Given a predicted workload $\hat{w}_{j,s}$ from an upstream model, the scheduling decision M determines the realized IT power consumption:

$$P_{i,t}^{\text{IT}} = \sum_{j\in\mathcal{N}_{\text{DC}}} \sum_{s\in\mathcal{T}} \hat{w}_{j,s} \cdot M_{j,i,s,t} + P_i^{\text{idle}}. \tag{9}$$

By combining the models from Sections II-A, II-B, and II-C, we formulate the downstream grid-coordinated workload scheduling problem. The objective is to minimize the total electricity generation cost (10a), subject to the complete set of system constraints:

$$\min_{\substack{M,P^{\text{IT}},P^{\text{DC}} \\ P^g,Q^g,P^l,Q^l,V}} \sum_{t\in\mathcal{T}} \sum_{i\in\Omega^g} \pi_{i,t} \cdot P_{i,t}^g \tag{10a}$$

$$\text{s.t.} \quad (1) - (9),$$

where $\pi_{i,t}$ is the electricity price. The constraints are grouped as follows: (1)-(5) represent the power grid constraints, ensuring power flow balance and voltage limits. (6)-(7) are the datacenter power constraints, linking IT load to total grid consumption. (8)-(9) are the workload transfer constraints, which define the spatio-temporal scheduling logic and its impact on IT load.

III. Methodology

This section details the proposed PTS learning framework, which integrates workload prediction, scheduling optimization, and decision evaluation into a unified closed-loop, end-to-end training pipeline.

A. The End-to-End Predict-then-Schedule Framework

In conventional two-stage approaches, prediction and optimization are treated as two separate, sequential procedures. First, an upstream prediction model is trained independently to minimize a statistical loss, such as Mean Squared Error (MSE) (11). Its predictions $\hat{w}$, are then treated as fixed, deterministic inputs for the downstream scheduling problem. This approach is limited by its failure to account for the impact of prediction errors on the final decision quality.

$$\mathcal{L}_{\text{MSE}}(\theta) = \frac{1}{N} \sum_{\{(w_i^{\text{hist}}, w_i)\}\in\mathcal{D}} \|\hat{w}_i - w_i\|^2,\ \hat{w}_i = f(w_i^{\text{hist}};\theta), \tag{11}$$

where w_i^{hist} is the historical workload used as input features, w_i is the corresponding ground-truth future workload, and (w_i^{hist}, w_i) represents a training sample drawn from the dataset $\mathcal{D}$. To overcome this limitation, we propose an end-to-end, decision-oriented PTS framework. We integrate the downstream grid-coordinated scheduling problem (10) directly into the upstream model's training pipeline by treating it as a differentiable decision layer. Rather than treating $\hat{w}$ as a constant, the framework views it as a differentiable parameter. This allows decision-based gradients from the final evaluation in Section III-B to backpropagate through the optimization layer via implicit differentiation in Section III-C. Consequently, the prediction model learns to produce decision-focused forecasts specifically optimized for high-quality, low-cost downstream scheduling.

B. Decision Evaluation Model with Over-shifted Loss

The Decision Evaluation Layer quantifies the true operational cost and ensures the differentiability of the learning pipeline. It evaluates the scheduling decision M^*, generated based on the prediction $\hat{w}$, against the ground-truth workload w. In the evaluation phase, the realized IT load $P_{i,t}^{\text{IT}*}$ upon w and M^* is defined as:

$$P_{i,t}^{\text{IT}*} = \sum_{j\in\mathcal{N}_{\text{DC}}} \sum_{s\in\mathcal{T}} w_{j,s} \cdot M_{j,i,s,t}^* + P_i^{\text{idle}}. \tag{12}$$

A fundamental challenge arises from the discrepancy between $\hat{w}$ and w. If M^* under-predicts the load and $P_{i,t}^{\text{IT}*}$ exceeds the capacity P_i^{full}, a strict optimization problem becomes infeasible, causing the gradient flow to break.

To guarantee the evaluation problem remains solvable, we relax the hard capacity constraint by introducing a load shedding variable $P_{i,t}^{\text{shed}}$ and propose the Over-shifted Loss $\mathcal{L}_{\text{cost}}$, which heavily penalizes any shedding. The evaluation is formulated as:

$$\mathcal{L}_{\text{cost}} = \min_{\Xi} \sum_{t\in\mathcal{T}} \left(\sum_{i\in\Omega^g} \pi_{i,t} P_{i,t}^g + \sum_{i\in\mathcal{N}_{\text{DC}}} \gamma \left(P_{i,t}^{\text{shed}} \right)^2 \right) \tag{13a}$$

$$\text{s.t.} \quad (1) - (7)$$

$$P_{i,t}^{\text{IT}} = P_{i,t}^{\text{IT}*} - P_{i,t}^{\text{shed}}, \quad P_{i,t}^{\text{shed}} \ge 0. \tag{13b}$$

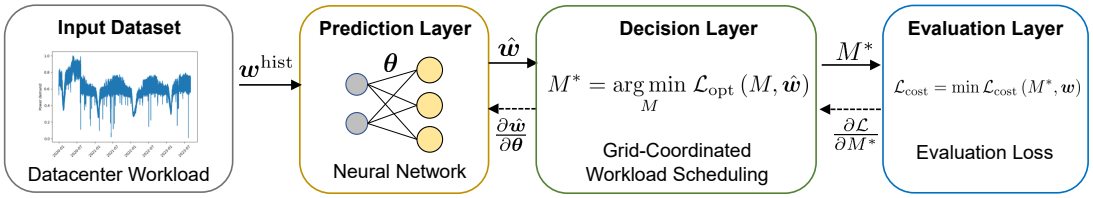


Fig. 1: End-to-end Predict-then-Schedule Framework

Here Ξ denotes the set of optimization variables. The coefficient $\gamma > 0$ is a penalty factor that governs the trade-off between economic optimality and constraint satisfaction. A sufficiently large γ prioritizes the elimination of load shedding, incentivizing the model to produce predictions that inherently hedge against grid violations. Constraints (13b) replace the hard capacity constraint with a soft relaxation to maintain gradient continuity during backpropagation. Minimizing (13a) explicitly teaches the upstream model to avoid forecasts that lead to unsafe schedules.

C. End-to-End Training via Implicit Differentiation

The key technical challenge in realizing the end-to-end framework from Section III is enabling the backpropagation of gradients from the $\mathcal{L}_{\text{cost}}$ defined in Section III-B through the decision layer to the parameters θ of the upstream predictor. We achieve this using the implicit differentiation. This technique is applicable because our downstream decision problems are convex programs with linear equality and inequality constraints. Drawing on foundational work such as [16], [17], the process works as follows:

1) Compact form: For convenience, we rewrite the decision layer in a generic form. Let y stack all primal decision variables, e.g., $\text{vec}(M)$, network flows, voltages, generations, and let ϕ collect problem parameters, e.g., the predicted load $\hat{w}(\theta)$ and other exogenous coefficients. Denote the convex objective by $J(y, \phi)$, the linear equalities by $h(y, \phi) = 0$, and the linear inequalities by $g(y, \phi) \leq 0$. The problem is

$$\min_y J(y, \phi) \quad \text{s.t.} \quad h(y, \phi) = 0, \quad g(y, \phi) \leq 0. \tag{14}$$

2) Lagrangian and KKT system: Let λ and $\mu \geq 0$ be the dual variables for $f(\cdot) = 0$ and $g(\cdot) \leq 0$, respectively. The Lagrangian is

$$\mathcal{L}(y, \lambda, \mu; \phi) = J(y, \phi) + \lambda^\top f(y, \phi) + \mu^\top g(y, \phi). \tag{15}$$

Because (14) is convex and satisfies standard regularity, strong duality holds and the KKT conditions are necessary and sufficient for optimality. The KKT system can be expressed as the vector equation

$$\mathcal{F}(y^*, \phi) = \begin{bmatrix} \nabla_y \mathcal{L}(y^*, \lambda^*, \mu^*; \phi) \\ h(y^*, \phi) \\ D(\mu^*)\, g(y^*, \phi) \end{bmatrix} = 0, \ \mu^* \geq 0, \ g(y^*, \phi) \leq 0. \tag{16}$$

where $D(\mu)$ denotes the diagonal matrix formed by μ.

3) Implicit differentiation: By the implicit function theorem, the optimizer y^* is a differentiable function of ϕ and its sensitivity is obtained by differentiating (16):

$$\frac{\partial \mathcal{F}}{\partial \phi} + \frac{\partial \mathcal{F}}{\partial y} \frac{\mathrm{d}y^*}{\mathrm{d}\phi} = 0 \quad \Rightarrow \quad \frac{\mathrm{d}y^*}{\mathrm{d}\phi} = -\left[\frac{\partial \mathcal{F}}{\partial y}\right]^{-1} \left[\frac{\partial \mathcal{F}}{\partial \phi}\right]. \tag{17}$$

Here $\frac{\partial \mathcal{F}}{\partial y}$ is the nonsingular KKT matrix composed of the Hessian blocks of (15) and the constraint Jacobians under LICQ and strict complementarity. Instead of forming an explicit inverse, we solve the linear system defined by (17) to obtain Jacobian–vector products efficiently.

4) Backpropagation to the predictor: Since the parameters ϕ depend on the predictor via $\phi = \hat{w}(\theta)$, the gradient of the over-shifted loss $\mathcal{L}_{\text{cost}}$ in (13a) with respect to the prediction network weights θ follows from the chain rule:

$$\frac{\mathrm{d}\mathcal{L}_{\text{cost}}}{\mathrm{d}\theta} = \frac{\mathrm{d}\mathcal{L}_{\text{cost}}}{\mathrm{d}y^*} \frac{\mathrm{d}y^*}{\mathrm{d}\phi} \frac{\mathrm{d}\phi}{\mathrm{d}\theta}. \tag{18}$$

(16)–(18) enable end-to-end training of the predict then schedule pipeline. The downstream optimizer remains an exact convex program, while its solution is fully differentiable with respect to the upstream forecast parameters.

D. Complete Training Framework

In summary, our proposed PTS architecture adopts the three-layer, end-to-end framework illustrated in Fig. 1. This framework connects workload prediction, decision-making, and decision evaluation in a closed loop. The layers are:

- Prediction Layer (Section III-A): An upstream neural network $f(w^{\text{hist}}; \theta)$, implemented as an LSTM, maps system features w^{hist} to a workload prediction $\hat{w}$.
- Decision Layer (Section II-C): A differentiable convex optimization layer that takes the prediction $\hat{w}$ as a parameter, solves the grid-coordinated scheduling problem (10), and outputs the optimal decision M^*.
- Decision Evaluation Layer (Section III-B): This layer assesses the true quality of the decision M^* by testing it against the ground-truth workload w. It solves the decision evaluation problem (13) to compute the final over-shifted loss $\mathcal{L}_{\text{cost}}$, which quantifies both economic cost and penalties for any safety-violating infeasibilities.

During training, this end-to-end pipeline enables the $\mathcal{L}_{\text{cost}}$ to be backpropagated from the evaluation layer, through the differentiable decision layer via Section III-C, and into the prediction layer to update θ. This feedback loop trains a predictor that is inherently aware of the downstream task. It learns to produce forecasts that proactively avoid high-cost, operationally infeasible regions of the solution space, thereby directly minimizing the true operational objectives of cost and safety.

IV. Simulation

To validate the proposed end-to-end PTS framework, we evaluate its performance against the conventional two-stage method in terms of operational cost, prediction accuracy, and constraint satisfaction.

A. Experimental Setup

1) System Configuration: Simulations are performed on a modified IEEE 33-bus distribution test system. One AI data center is integrated at bus 24. The parameters of the AI data center include an idle power consumption of 0.00 units and a full power consumption capacity of 0.15 units. The scheduling horizon is 24 hours with hourly discretization. The hourly electricity price profile, which is shown in Fig. 4, is generated through a transformation of a sinusoidal function, with values ranging between 0.5 and 1, reflecting the diurnal fluctuations in electricity prices.

2) Data Preprocessing and Model Parameters: The workload dataset used in this experiment is synthesized from real-world power demand forecasting data and has been normalized to [0,1], which is shown in Fig. 2. It consists of hourly power load values that represent the electrical consumption. This dataset spans 3.5 years of hourly data, totaling approximately 30,660 data points. This duration was selected to capture diverse seasonal and diurnal load patterns for robust generalization, and is partitioned into 80% for training and 20% for testing.

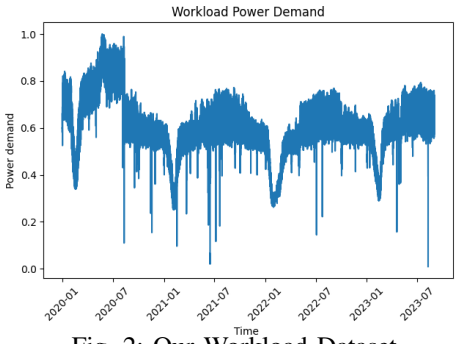


Fig. 2: Our Workload Dataset

TABLE I: Comparison of metrics among different training methods.

Method	MSE Loss	Cost Loss	Violation
Two-Stage	0.0106	31.8381	0.7518
Two-Stage (Reserve)	0.0106	14.9977	0.3190
PTS (γ=10)	0.0490	10.7189	0.0007
PTS (γ=100)	0.0931	10.7205	0.0007

3) Models for Comparison: We benchmark our proposed approach against the traditional two-stage baseline, evaluating performance across various metrics:

- Two-Stage: The traditional baseline framework, where the predictive model is trained solely to minimize MSE in (11).
- Two-Stage (Reserve): This method sets a new operational limit at 80% of the rated capacity while reserving the remaining 20% as a safety margin.
- PTS: Our proposed framework, where the predictor is trained to minimize the over-shifted loss (13a) that balances operational cost and power system security.

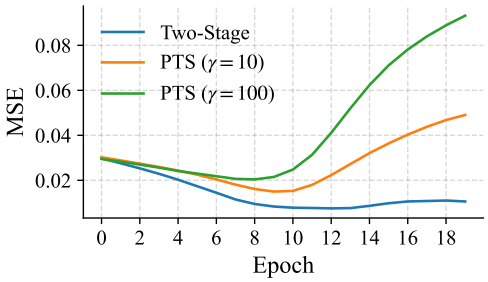
4) Implementation and Training Details: The prediction model is a LSTM neural network implemented in PyTorch. The downstream optimization is formulated in CVXPY and embedded as a differentiable layer. Experiments are conducted on an HPC node equipped with an Intel Core i5-13600K CPU and one NVIDIA RTX-3060 GPU. Models are trained for 20 epochs with learning rate of 1×10^{-3} and batch size of 30.

B. Results and Analysis

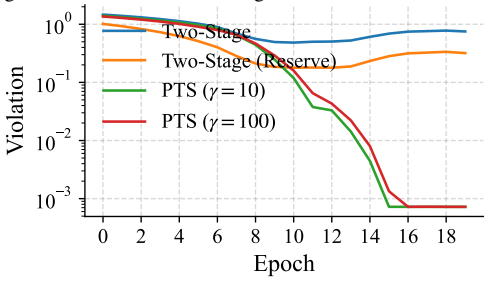
We evaluate the two training strategies on the test dataset, comparing operational cost, prediction accuracy, and the quality of the resulting scheduling decisions.

1) Operational Cost and Accuracy Trade-off: The comparative performance during training is evaluated across three dimensions. Figure 3b illustrates the magnitude of constraint violations, Figure 3a displays the mean squared error between predicted and ground-truth workloads, and Figure 3c presents the total operational cost. Table I further summarizes the final quantitative results.

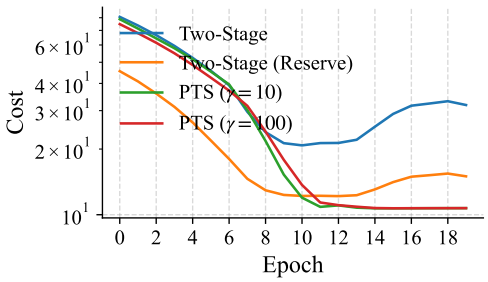
The results reveal a distinct trade-off between statistical fidelity and decision quality. While the Two-Stage model achieves the lowest MSE, it suffers from significant constraint violations and high operational costs. The Reserve baseline improves system safety relative to the standard Two-Stage approach but results in suboptimal costs due to its inflexible capacity limits. In contrast, the PTS models prioritize the more critical downstream decision objective over conventional statistical accuracy. The prediction MSE increases in the later stages of training and thus successfully reduces the violation metric by nearly an order of magnitude to a stable and near-zero level. This performance demonstrates that internalizing downstream constraints and cost objectives enables the model to generate safe, feasible, and significantly more cost-effective decisions.



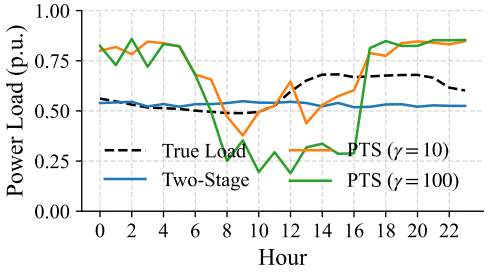
(a) The mean squared error between predicted and ground-truth workload among three models.



(b) The magnitude of constraint violations among four methods.



(c) The cost comparison among four methods.



(d) The prediction comparison between true load and predicted load from three models.

Fig. 3: Performance Comparison of Different Methods

The sensitivity analysis regarding the penalty coefficient γ further demonstrates the structural robustness of the proposed framework. As illustrated in Fig. 3b, both $\gamma = 10$ and $\gamma = 100$ successfully reduce violations to a comparable stable floor. Although a larger γ of 100 imposes a stricter penalty that leads to higher initial operational costs in Fig. 3c, both configurations eventually converge to similar cost levels. This outcome indicates that the framework is insensitive to the specific value of γ provided the coefficient is sufficiently large to maintain system constraints.

2) Analysis of Prediction Behavior: As shown in Fig. 3d, the predictive behaviors differ significantly. The Two-Stage model adheres closely to the true load, displaying high statistical fidelity. In contrast, the PTS models produce decision-oriented predictions with intentional deviations, for example, over-forecasting during hours 0-

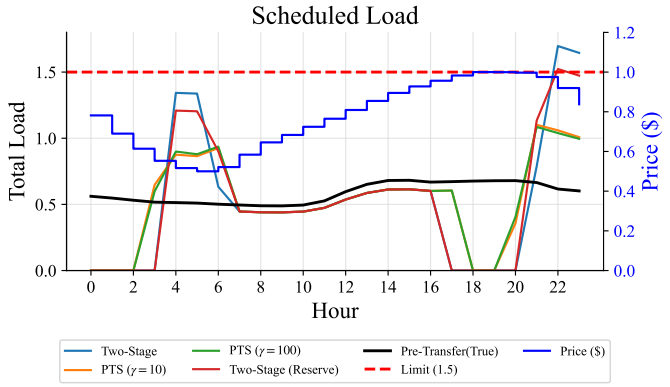


Fig. 4: Comparison of task transfer results among four methods.

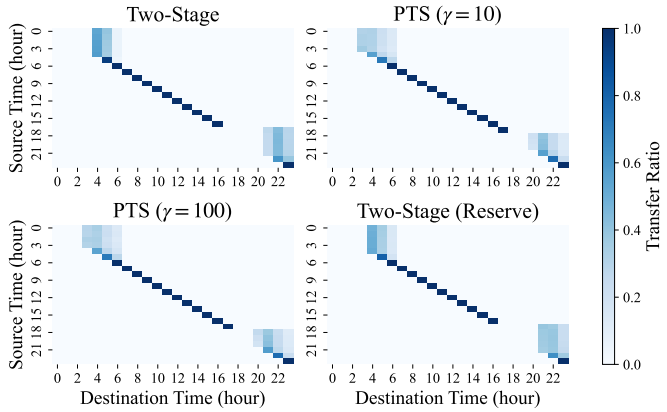


Fig. 5: Heatmaps of learned transfer matrices $M_{d,c,s,t}$ among four methods.

6, 17-23 and under-forecasting during hours 11-16. This sacrifice in statistical accuracy is not a flaw, but rather a learned strategy, as the PTS models recognize this distorted prediction is essential for guiding the downstream optimization process away from infeasible or high-cost decisions.

3) Impact on Scheduling Decisions: The difference in predictive behavior directly impacts the feasibility of the downstream scheduling decisions, as shown in Fig. 4. The Two-Stage models, based on its statistically accurate forecast, attempts to exploit low prices during hours 4-6, 22-23, but this results in severe violations of the capacity limit. The Reserve method avoids extreme violations but shows a minor violation. Conversely, the PTS framework demonstrates its superiority. By leveraging its awareness of the downstream optimization, it accurately identifies the capacity limit as a critical bottleneck. Its distorted prediction was specifically designed to avoid this. The final schedule intelligently smooths the workload scheduling, both exploiting the low prices and, crucially, never breaching the operational limit.

4) Analysis of Learned Transfer Policies: Fig. 5 visualizes the learned transfer matrices M. While both models learn to transfer workload to low-cost periods, the policy learned by the PTS model holds relatively conservative transfer strategies. For instance, to prevent constraint violations, the PTS will not shift the workload from 17:00 to later time slots, even if this could improve overall resource utilization.

V. Conclusion

This paper introduces an end-to-end PTS framework for AI datacenter workload scheduling coordinated with power distribution networks. By integrating the downstream convex scheduling optimization as a differentiable layer using implicit differentiation, our framework aligns workload prediction with final operational objectives. The proposed workload over-shifted loss ensures that the model is trained to balance operational cost with power system security. Experimental results demonstrate that our framework significantly reduces operational cost and improves constraint satisfaction compared to the conventional two-stage baseline.

References

[1] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020.

[2] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 3645–3650.

[3] J. Sun, M. Xu, M. Cespedes, and M. Kauffman, "Data center power system stability—part i: Power supply impedance modeling," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 2, pp. 403–419, 2022.

[4] M. S. Munir, S. F. Abedin, N. H. Tran, Z. Han, E.-N. Huh, and C. S. Hong, "Risk-aware energy scheduling for edge computing with microgrid: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3476–3497, 2021.

[5] X. Chen, X. Wang, A. Colacelli, M. Lee, and L. Xie, "Electricity demand and grid impacts of ai data centers: Challenges and prospects," 2025. [Online]. Available: https://arxiv.org/abs/2509.07218

[6] J. Li and W. Qi, "Toward optimal operation of internet data center microgrid," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 971–979, 2016.

[7] A. Safari, K. Taghizad-Tavana, and M. Tarafdar Hagh, "Artificial intelligence-driven optimization of internet data center energy consumption in active distribution networks: a transformer-based robust control model with spatio-temporal flexibility analytics," *Available at SSRN 5056252*, 2024.

[8] Z. Chen, J. Hu, G. Min, A. Y. Zomaya, and T. El-Ghazawi, "Towards accurate prediction for high-dimensional and highly-variable cloud workloads with deep learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 4, pp. 923–934, 2019.

[9] B. D. Dimd, S. Völler, U. Cali, and O.-M. Midtgård, "A review of machine learning-based photovoltaic output power forecasting: Nordic context," *IEEE Access*, vol. 10, pp. 26 404–26 425, 2022.

[10] T. Schmitt, T. Rodemann, and J. Adamy, "The cost of photovoltaic forecasting errors in microgrid control with peak pricing," *Energies*, vol. 14, no. 9, p. 2569, 2021.

[11] A. N. Elmachtoub and P. Grigas, "Smart "predict, then optimize"," *Management Science*, vol. 68, no. 1, pp. 9–26, 2022.

[12] P. Donti, B. Amos, and J. Z. Kolter, "Task-based end-to-end model learning in stochastic optimization," *Advances in neural information processing systems*, vol. 30, 2017.

[13] H. Zhang, R. Li, Q. Du, J. Tao, S. Pineda, G. Kariniotakis, S. Camal, C. B. Monroc, M. Sun, C. Wan, W. Xu, and F. Teng, "Decision-Focused Learning for Power System Decision-Making Under Uncertainty," pp. 1–18.

[14] B. Wilder, B. Dilkina, and M. Tambe, "Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 1658–1665.

[15] M. Farivar and S. H. Low, "Branch flow model: Relaxations and convexification—part i," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2554–2564, 2013.

[16] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International conference on machine learning*, 2017, pp. 136–145.

[17] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," *Advances in neural information processing systems*, vol. 32, 2019.