Predictive Modeling of Energy Production and Graph Compression via MARL and DCRNN

W. Bernard Lee[1] Anthony G. Constantinides[2]

[1]HedgeSPA Private Limited, Singapore
bernard.lee@hedgespa.com

[2]AI & Data Analytics Lab., Imperial College London, UK
a.constantinides@imperial.ac.uk

October 2025

Presenter Profiles

Dr. W. Bernard Lee
Founder and Chief Executive, HedgeSPA Private Limited

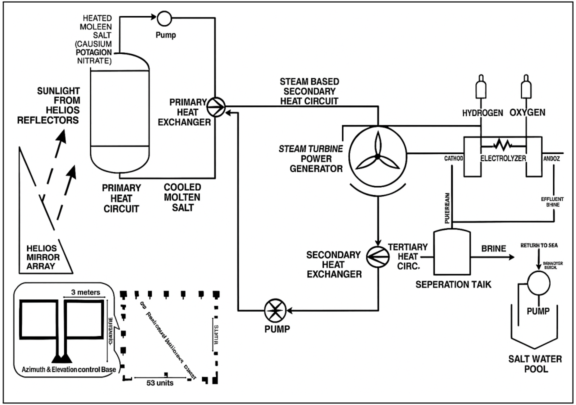
- Former Managing Director, Portfolio Management Group, BlackRock, New York
- Former Finance Professor in Singapore, the US and Hong Kong
- AI investment analytics firm seed-funded by Singapore's National Research Foundation, and is under funding round led by global Tier 1 bank

Prof. Anthony G. Constantinides
Professor Emeritus, Imperial College London
Co-Director, AI & Data Analytics Lab

- Fellow, Royal Academy of Engineering & US National Academy of Engineering
- IEEE Life Fellow, Knighthood in France, and numerous other honors
- Named by IEEE as one of the founding fathers of signal processing who authored some of the highest cited papers in signal processing and graph theory

Goal: Use AI to automate 2∼3X more-efficient solar-thermal technologies

(a) Desert deployment does not need to pay for advanced, high-efficiency technology.



(b) High-efficiency tech works to maximize energy from limited space (e.g. rooftops).

Figure: AI enables optimal placement of advanced tech by overcoming traditional control barriers.

Abstract

Modern energy infrastructures exhibit complex spatiotemporal dynamics across thermal, mass, and electrical domains. We propose a unified framework that:

- Compresses multi-layer graphs into a single effective flow network
- Applies DCRNN for one-step-ahead node state prediction
- Integrates MARL for actuator control optimization

This approach balances physical fidelity with computational tractability and supports real-time regulation in operational energy systems.

Preprint

Being processed by *MDPI Information* (IF 2.4, CiteScore 6.9) for its forthcoming Special Issue: “Applications of Information Extraction, Knowledge Graphs, and Large Language Models”

information

MDPI

Article

Predictive Modeling of Energy Production and Graph Compression via MARL and DCRNN

W. Bernard Lee [1] and Anthony G. Constantinides [2]

[1] HedgeSPA Private Limited, 12 Woodlands Square #05-70, Singapore, 737715; bernard.lee@hedgespa.com
[2] AI and Data Analytics Laboratory, Imperial College London, London, UK SW7 2AZ; a.constantinides@imperial.ac.uk

Abstract

Modern energy infrastructures—from thermal power plants and combined-cycle units to microgrids and biomass-integrated renewables—exhibit complex spatiotemporal dynamics. Variables such as temperature, pressure, viscosity, mass flow rate, and electrical load propagate through networks of pipes, heat exchangers, turbines, and transmission lines, which can be naturally modeled as layered graphs. Real-time forecasting and coordinated

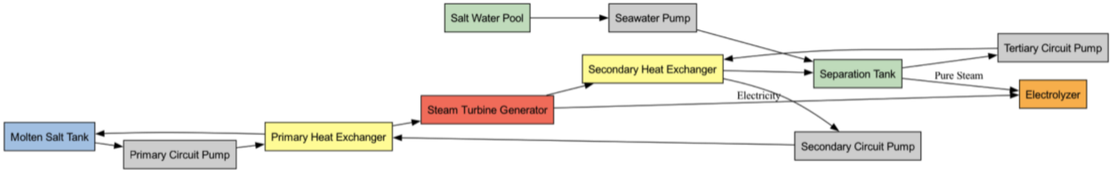
Motivation

- Traditional PDE-based models are accurate but hard to scale
- Energy systems are heterogeneous and topology evolves frequently
- Data-driven graph models offer flexibility but face:
  1. Multi-layer graph complexity
  2. Long-range temporal dependencies
  3. Non-stationary operating regimes
- Practically, this represents a significant and ambitious undertaking:
  1. Several governments have expressed formal interest in deployment
  2. UK has committed to providing access to a designated test site at a symbolic rental rate, alongside support for establishing an R&D center at Bayes Centre, Univ of Edinburgh
  3. A Special Purpose Vehicle (SPV) is being structured to consolidate assets and enable scalable investment

Design Principles

1. This is based on layered graphs (e.g. thermal, mass, electrical, etc.); compressed into a single directed graph based on energy flow
2. Use DCRNN (initially developed at Caltech for controlling traffic flows in LA) to forecast next-step node states
3. Close the loop with MARL agents (used in manufacturing to control overflows) using learned forecasts

This preserves locality, reduces memory, and enables fast policy improvement.



Background Research

1. Multi-Agent Reinforcement Learning (MARL)
    - Agents interact with environment to maximize expected return
    - MARL settings: Dec-POMDPs, CTDE
    - Coordination via mixing networks, shared critics, communication
    - Applied to edge computing and industrial control, so it is a natural fit with energy systems
    - Rewards reflect efficiency, stability, and safety
2. Diffusion Convolutional RNN (DCRNN)
    - Combines graph diffusion with GRUs
    - Models directional transport via biased random walks
    - Learns spatiotemporal signal propagation
    - Proven effective in traffic and recommendation systems
    - Physically consistent for heat and mass flow modeling

Problem Formulation

- Graph: $G = (V, E, A)$, with compressed adjacency A
- Node state: $x_t^{(i)} = [T_t^{(i)}, q_t^{(i)}, p_t^{(i)}, \dots]$
- System dynamics: $X_{t+1} = f(X_t, u_t; G) + \varepsilon_t$
- Objectives:
  - Forecast X_{t+1} from history
  - Compute control u_t minimizing stage cost or maximum total output objective
- Energy Transfer Approximation: Local energy balance given by:

$$E_t^{(i)} = m^{(i)} c_p T_t^{(i)}, \quad \dot{E}_t^{(i)} \approx \sum_j \alpha_{ji} q_t^{(j \to i)} c_p (T_t^{(j)} - T_t^{(i)}) - \beta^{(i)} (T_t^{(i)} - T_{\mathrm{amb}}) + \gamma^{(i)} u_t^{(i)}$$

  - α_{ji}: directional coupling
  - $\beta^{(i)}$: energy loss coefficient
  - $\gamma^{(i)} u_t^{(i)}$: actuation input

Graph Compression Strategy

- Aggregate multi-layer graphs into one:

$$A_{ij} = \sum_{\ell \in \{\text{thermal, mass, electrical}\}} w_\ell \cdot \phi_\ell(A_{ij}^{(\ell)})$$

- Optional learned correction:

$$A^\star = A + \Delta A, \quad \|\Delta A\|_1 \leq \lambda$$

- Preserves topology and directionality

DCRNN for One-Step Forecasting

- Combines diffusion convolution with GRU architecture
- Predicts next-step node states X_{t+1} from current state and control
- Enables conditional forecasting with control-aware dynamics

Diffusion Convolution:
Given row-normalized transition matrices $D^{-1}A$ and $\tilde{D}^{-1}A^T$, the K-step bidirectional diffusion convolution is:

$$\mathrm{DiffConv}(X) = \sum_{k=0}^{K} D^{-1}A^k X \Theta_k^{(f)} + \tilde{D}^{-1}(A^T)^k X \Theta_k^{(b)}$$

Captures anisotropic transport aligned with physical flow and supports directional modeling in energy networks

GRU Integration:
Embed diffusion convolution in GRU:

$$Z_t = \sigma(\mathrm{DiffConv}([X_t, U_t])W_z + H_{t-1}R_z + b_z)$$
$$R_t = \sigma(\mathrm{DiffConv}([X_t, U_t])W_r + H_{t-1}R_r + b_r)$$
$$\tilde{H}_t = \tanh(\mathrm{DiffConv}([X_t, U_t])W_h + (R_t \odot H_{t-1})R_h + b_h)$$
$$H_t = (1 - Z_t) \odot H_{t-1} + Z_t \odot \tilde{H}_t$$

Final prediction: $X_{t+1} = H_t W_o + b_o$

Forecasting Objective and MARL Control

Loss and Training Objective

$$\mathcal{L}_{\text{forecast}} = \sum_t \|X_{t+1}^{\text{true}} - X_{t+1}^{\text{pred}}\|_1 + \eta\|\Delta A\|_1$$

$$+ \rho \textstyle\sum_i \max(0, -A^{\star}_{ii})$$

- L1 loss for robustness to outliers
- Penalizes self-loops and encourages sparsity

MARL Control with One-Step Horizon

- Each actuator node $i \in U$ is assigned an agent
- Observes local and neighborhood states: $o_t^{(i)} = \psi(X_t, N(i; G))$
- Uses DCRNN as learned transition model for planning

Reward Shaping and Bellman Update

Reward Shaping

- Actuation cost: $-\lambda_u \|u_t\|_2^2$
- Smoothness: $-\lambda_\Delta \|u_t - u_{t-1}\|_2^2$
- Tracking:

$$r_t = -\sum_i w_T^{(i)} (T_{t+1}^{(i)} - T_{\text{ref}}^{(i)})^2$$

- $\sum_{(i,j)\in C} w_q^{(i,j)} (q_{t+1}^{(i\to j)} - q_{\text{ref}}^{(i\to j)})^2$

Bellman Equation with DCRNN

$$Q(s_t, u_t) = r_t + \gamma \mathbb{E}_{s_{t+1}}[V(s_{t+1})],$$

$$s_{t+1} = \text{Enc}(X_{t+1}) = \text{Enc}(\text{DCRNN}(X_t, u_t))$$

- Reduces compounding model error
- Stabilizes training with one-step lookahead

CTDE and One-Step Forecasting Power

Centralized Training, Decentralized Execution (CTDE)

- Centralized critic $Q(s, u)$ trained with joint rollouts
- Each agent executes local policy $\pi^{(i)}(a|o^{(i)})$
- Optional parameter sharing and neighbor communication

One-Step Forecasting Power

$$X_{t+1} = A_x X_t + B_u u_t + \varepsilon_t, \quad \rho(A_x) < 1$$

- DCRNN learns local Jacobians via diffusion kernels
- Bellman backups over one step sufficient for greedy policy improvement
- Prevents error accumulation over long rollouts

Valve Dynamics and Safety Constraints

Valve Control and Energy Flow

- Mass flow: $q_{t+1}^{(i\to j)} = g(u_t^{(i\to j)})$
- Pressure drop: $\Delta p_t^{(i,j)} \approx k_v^{(i\to j)} u_t^{(i\to j)} \Delta p_t^{(i,j)}$
- Heat transfer:

$$\dot{Q}_{t+1}^{(i\to j)} = q_{t+1}^{(i\to j)} c_p (T_t^{(i)} - T_t^{(j)})$$

- Temperature update:

$$T_{t+1}^{(j)} = T_t^{(j)} + \frac{\Delta t}{m^{(j)} c_p} \left(\sum_i \dot{Q}_{t+1}^{(i\to j)} - \sum_k \dot{Q}_{t+1}^{(j\to k)} - h^{(j)} A^{(j)} (T_t^{(j)} - T_{\text{amb}}) \right)$$

Safety and Constraints

- Temperature bounds: $T_{\min}^{(i)} \leq T_t^{(i)} \leq T_{\max}^{(i)}$
- Control bounds: $0 \leq u_t^{(i)} \leq 1$
- Flow limits: $\sum_j q_t^{(i\to j)} \leq q_{\max}^{(i)}$
- Smoothness: $|\Delta u_t^{(i)}| \leq \delta_u$
- Training: penalize violations with barrier terms
- Execution: project actions onto feasible set

Integrated Algorithm Overview

1. **Graph Compression**
   - Input: Layered adjacencies $A^{(\ell)}$, physical metadata
   - Output: Effective directed adjacency $A^{\star}$
2. **DCRNN Training**
   - Data: Sequences $(X_{t-k+1:t}, u_{t-k+1:t}) \rightarrow X_{t+1}$
   - Objective: Minimize $\mathcal{L}_{\text{forecast}}$; early stopping on validation
3. **Critic and Policy Training (CTDE)**
   - Target: $y_t = r(s_t, u_t, s_{t+1}) + \gamma V_\theta(s_{t+1})$
   - Critic: Minimize $(Q_\theta(s_t, u_t) - y_t)^2$
   - Policy: Maximize $Q_\theta(s_t, \pi_\phi(o_t))$ with entropy regularization
4. **Deployment**
   - Execution: Observe X_t, compute $u_t = \pi(o_t)$
   - Forecast: Optionally predict X_{t+1} for monitoring
   - Logging: Store transitions for periodic retraining

MPC Initialization and Theoretical Foundations

MPC Initialization and MARL Fine-Tuning

- Initialize MARL with model predictive control (MPC) using DCRNN as one-step model
- Fine-tune agents with MARL to adapt to nonlinear effects
- Combines model-based and model-free learning for robust control

Consistency of Diffusion Convolution

- Continuous transport: $\frac{\partial \phi}{\partial t} = -v \cdot \nabla \phi + \kappa \nabla^2 \phi$
- Discretized: $\phi_{t+1} \approx [(1-\alpha)I + \alpha P_f + \beta P_b]\phi_t$
- DCRNN approximates polynomial functions of P_f, P_b

Stability Under Model Error

- Assume Lipschitz error: $\|f(X,u) - f(X,u)\| \leq \varepsilon$
- If policy satisfies: $V(f(X_t,u_t)) - V(X_t) \leq -\alpha\|X_t\|^2 + c\varepsilon$
- Then system remains input-to-state stable for small ε

Training Pipeline

Data Preprocessing

- Normalize features per node using Z-score
- Initialize $A^{\star}$ with physical priors; allow small corrections ΔA
- History window $k \in [20, 100]$ for context vs. stability

Architecture Choices

- Diffusion steps $K = 2$ or 3; hidden size: 64256
- Control conditioning: concatenate U_t, FiLM-style gating

Optimization Strategy

- Forecasting: Adam optimizer, cosine decay, teacher forcing scheduled sampling
- MARL: actor-critic, target networks, prioritized replay, entropy annealing

Regularization and Constraints

- Spectral norm on diffusion kernels; graph sparsity penalties
- Safety layers: project actions, enforce feasibility

Evaluation Framework

Methodology

- Datasets: synthetic PDE pipe simulations, historical SCADA/PI telemetry
- Baselines: ARIMA, LSTM, TCN; GCN-GRU, Graph WaveNet, STGCN; PID/MPC, single-agent RL

Metrics

- Forecasting: MAE, MAPE, RMSE, calibration error
- Control: cumulative reward, energy efficiency, constraint violations, settling time
- Ablations: graph compression, diffusion steps K, control conditioning, DCRNN vs. GRU

Testbed Overview

- Closed-loop hydrogen production simulation
- Thermal, mass, electrical dynamics; realistic deployment validation

System Design and Control

Graph Construction

- Directed graph $G = (V, E)$: electrolyzers, compressors, heat exchangers, mirror arrays
- Edges encode thermal, fluid, electrical flow
- Feature vector $x_t^{(i)} \in \mathbb{R}^d$: temp, pressure, voltage, flow, actuator
- Global schema with semantic sparsity

Temporal Forecasting

- PyTorch implementation; input: $X_{t-50:t}$, output: $X_{t+1:t+3}$
- Diffusion convolution + GRU for directional transport and temporal dependencies

Closed-Loop Control

1. DCRNN predicts $X_{t+1:t+3}$
2. AIController selects u_t
3. PlantSimulator computes X_{t+1}
4. Loop repeats for adaptive control

Control Simulation Results (Starting Temp = 90°C, Target = 110°C)

Iteration	11x11 Adjacency Matrix		17x17 Adjacency Matrix		22x22 Adjacency Matrix	
	Tank Temp (°C)	Mirror Action	Tank Temp (°C)	Mirror Action	Tank Temp (°C)	Mirror Action
0	89.71	On	89.80	On	89.69	On
10	87.15	On	88.20	On	88.09	On
20	84.98	On	86.16	On	86.05	On
30	85.80	On	86.82	On	86.14	On
40	90.55	On	91.96	On	90.93	On
50	98.32	On	99.58	On	98.61	On
60	105.31	On	106.88	On	105.05	On
70	109.45	On	110.21	Off	109.68	On
80	109.21	On	109.18	On	108.62	On
90	107.68	On	106.88	On	106.82	On
Comp. Time (sec)	4.832946		4.684803		4.706305	

Control Simulation Results (Target = 110°C, Various Starting Temperatures)

Iteration	Start at 90°C		Start at 100°C		Start at 108°C		Start at 115°C	
	Tank Temp (°C)	Mirror Action	Tank Temp (°C)	Mirror Action	Tank Temp (°C)	Mirror Action	Tank Temp (°C)	Mirror Ac
0	89.71	On	99.76	On	107.67	On	114.98	On
10	87.15	On	98.05	On	105.88	On	113.38	On
20	84.98	On	95.84	On	103.86	On	111.02	On
30	85.80	On	95.83	On	103.93	On	110.94	On
40	90.55	On	101.15	On	108.95	On	115.70	On
50	98.32	On	108.87	On	115.29	Off	120.89	Off
60	105.31	On	109.88	Off	114.01	Off	120.18	Off
70	109.45	On	109.65	Off	112.71	Off	119.11	Off
80	109.21	On	108.55	On	110.61	Off	117.55	Off
90	107.68	On	106.48	On	109.14	On	115.82	Off
omp. Time (sec)	4.832946		4.802022		4.779279		4.807150	

Simulation Insights

Adjacency Matrix Augmentation

- Duplicate sensor nodes for intake-output dynamics
- 17×17 and 22×22 matrices preserve topology

Results: Matrix Size

- Predictive accuracy stable; runtime: $11\times11 \to 4.83\,\mathrm{s}$, $17\times17 \to 4.68\,\mathrm{s}$, $22\times22 \to 4.71\,\mathrm{s}$

Real-Time Deployment

- 100 slices/day $\to$ 14 m 24 s/slice; history: 50 slices; horizon: 3 slices
- Inference: <5s on 2-core Xeon

Results: Start Temperatures

- Policy delays mirror deactivation in cold conditions; safety overrides learned behavior
- **Runtime:** $90^\circ\mathrm{C} \to 4.83\,\mathrm{s}$, $100^\circ\mathrm{C} \to 4.80\,\mathrm{s}$, $108^\circ\mathrm{C} \to 4.78\,\mathrm{s}$, $115^\circ\mathrm{C} \to 4.81\,\mathrm{s}$

Summary and Future Work

Section Summary

- DCRNN: accurate short-horizon forecasts
- MARL: stable, constraint-respecting policies
- Fast inference on standard hardware
- CFD too slow for short-horizon control

Deployment Limitations

- Fixed $A^\star$ may miss regime shifts $\rightarrow$ learn $A^\star(X_t)$ via attention
- Sensor noise $\rightarrow$ Kalman/Neural-Kalman filters
- Partial observability $\rightarrow$ Dec-POMDPs, shared critics
- Safety certification $\rightarrow$ control barrier functions, reachability analysis

Toward Physics-Informed Optimization

- **Physics-informed loss**:

$$\mathcal{L}_{\text{phys}} = \sum_i \left\| \frac{m^{(i)} c_p}{\Delta t} (T_{t+1}^{(i)} - T_t^{(i)}) - \sum_j q_t^{(j \to i)} c_p (T_t^{(j)} - T_t^{(i)}) + \beta^{(i)} (T_t^{(i)} - T_{\text{amb}}) - \gamma^{(i)} u_t^{(i)} \right\|$$

- **Differentiable MPC head**: Solve constrained quadratic subproblem with backprop
- **Multi-horizon rollout**: Validate closed-loop behavior to detect drift or compounding error
- **Next step**: Set up R&D Center at Bayes Centre, University of Edinburgh

Conclusions

- Unified framework for forecasting and control in energy systems
- Combines graph compression, DCRNN prediction, and MARL control
- Diffusion convolution captures directed energy transport
- One-step forecasting reduces complexity and supports real-time optimization
- CTDE-style MARL integrates naturally with physical constraints

Real-Time Viability and Simulation Integration

Real-Time Viability

- All test computations complete in < 5 seconds on standard desktop hardware
- Enables decisions within narrow temporal windows
- CFD models are too slow for short-horizon control
- Runtime and accuracy confirm commercial viability

Bridging Simulation and Control

- Future work: Evaluate with CFD simulations and real-world hardware
- Physics-informed regularization strengthens reliability
- Goal: Bridge gap between high-fidelity simulation and sensor-driven control
- Supports next-generation energy infrastructure

References

- Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). *Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting*. International Conference on Learning Representations (ICLR).
- Zhang, K., Yang, Z., & Baar, T. (2021). *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*. In Handbook of Reinforcement Learning and Control. Springer.
- Oroojlooy, A., & Hajinezhad, D. (2020). *A Review of Cooperative Multi-Agent Deep Reinforcement Learning*. arXiv preprint arXiv:1908.03963.
- Rosenberger, J., Metzger, A., & Pohl, K. (2022). *Deep Reinforcement Learning Multi-Agent System for Resource Allocation in Industrial Internet of Things*. Sensors, 22(11), 4099. https://www.mdpi.com/1424-8220/22/11/4099
- Zhou, J., & Yu, Q. (2023). *DCRNN: A Deep Cross Approach Based on RNN for Partial Parameter Sharing in Multi-task Learning*. arXiv preprint arXiv:2310.11777. https://arxiv.org/abs/2310.11777
- Sutton, R. S. (1990). *Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming*. Proceedings of the Seventh International Conference on Machine Learning, pp. 216224.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.
- Lee, W. B., & Constantinides, A. G. (2023). *Green Hydrogen Production, Transportation, Securitization, and Tokenization using Graph-Neural-Network AI-Automated Control*. International Journal on Intelligent Systems, 16(12), 118. IARIA. https://www.iariajournals.org/intelligent_systems/tocv16n12.html