Deep Graph Learning for Industrial Carbon Emission Analysis and Policy Impact

Xuanming Zhang[1,2]
[1]Uniersity of Wisconsin-Madison
[2]Stanford University
xzhang2846@wisc.edu

Abstract

Industrial carbon emissions are a major driver of climate change, yet modeling these emissions is challenging due to multicollinearity among factors and complex interdependencies across sectors and time. We propose a novel graph-based deep learning framework DGL to analyze and forecast industrial CO_2 emissions, addressing high feature correlation and capturing industrial-temporal interdependencies. Unlike traditional regression or clustering methods, our approach leverages a Graph Neural Network (GNN) with attention mechanisms to model relationships between industries (or regions) and a temporal transformer to learn long-range patterns. We evaluate our framework on public global industry emissions dataset derived from EDGAR v8.0, spanning multiple countries and sectors. The proposed model achieves superior predictive performance – reducing error by over 15% compared to baseline deep models – while maintaining interpretability via attention weights and causal analysis. We believe that we are the first Graph-Temporal architecture that resolves multicollinearity by structurally encoding feature relationships, along with integration of causal inference to identify true drivers of emissions, improving transparency and fairness. We also stand a demonstration of policy relevance, showing how model insights can guide sector-specific decarbonization strategies aligned with sustainable development goals. Based on the above, we show high-emission "hotspots" and suggest equitable intervention plans, illustrating the potential of state-of-the-art AI graph learning to advance climate action, offering a powerful tool for policymakers and industry stakeholders to achieve carbon reduction targets.

1 Introduction

Climate change mitigation hinges on reducing carbon emissions from key sectors like industry, energy, and transport [1]. Industry (manufacturing, materials, construction) alone accounts for roughly a quarter to a third of global greenhouse gas emissions, making it a critical focal point for climate policy [2]. Achieving deep emission cuts in industry requires accurate modeling of emission drivers and effective, interpretable predictions to inform interventions. However, industrial emissions arise from a complex interplay of factors – fuel consumption (coal, oil, gas, electricity), production outputs, efficiency measures, supply chain interactions – which are often highly correlated [3]. This multicollinearity undermines traditional models: classical statistical tools (e.g. the IPAT/STIRPAT framework or index decomposition) struggle with nonlinear and correlated features. Even advanced machine learning models like neural networks face difficulty distinguishing collinear inputs, leading to suboptimal performance when multi-factor aggregate. For instance, prior work observed that standard deep networks often fail to properly attribute importance to individual energy sources if those sources co-vary, limiting the model's reliability in multi-factor scenarios [4].

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Another challenge is capturing interdependencies across industries and over time. Emissions in one sector may be linked to inputs from another (e.g. metal production driving demand in mining), and global supply chains create graph-like relationships between regional industries. Emissions also evolve with economic and technological trends, exhibiting temporal patterns (e.g. growth, peaks, structural breaks). Traditional approaches that ignore these structural and temporal couplings may miss critical dynamics. Recent studies have begun embracing machine learning and deep learning to tackle emission prediction – e.g. applying CNN-LSTM to forecast energy demand [5], and hybrid ARIMA-LSTM for regional CO_2 forecasting [6]. These sequence models capture nonlinear time patterns better than static regressions. However, purely sequential models often treat features independently and as black boxes, suffering interpretability issues and still struggling when input features are highly correlated or homogeneous. In short, two key gaps remain: (i) how to manage multifactor industrial data with severe multicollinearity, and (ii) how to extract actionable, interpretable insights (e.g. which industries or fuels drive emissions) rather than treating the model as an inscrutable predictor.

To address these gaps, we introduce Deep Graph Learning (DGL) for industrial carbon emission analysis that combines deep learning with domain-inspired structure. Our approach integrates graph neural network to encode industrial relationships and a transformer-based sequence model for temporal dependencies, along with causal interpretability module. By representing industries (or regions) as nodes in graph and learning their connections, the model explicitly accounts for inter-sector dependencies – for example, recognizing that steel and cement industries form a cluster due to shared energy inputs. This helps resolve multicollinearity: correlated features can be aggregated or disentangled via graph message-passing instead of competing in isolation. Meanwhile, transformer component learns temporal patterns (trends, seasonal effects, policy shifts over years) for each node and cluster of nodes. Importantly, our model is designed for interpretability: we employ attention mechanism with causal inference to ensure the learned relationships make sense and can guide policy. For instance, attention weights highlight which input factors or neighboring industries most influence a given industry's emissions, and a causal analysis module allows simulation of interventions (e.g. "what if coal usage in sector X is reduced by 20%?"). This aligns our work with policy goals of sustainable development and fairness – providing not just predictions but explanations and fair recommendations for emission reductions. Our contributions are summarized as follows:

❶ DGL Framework: We propose a new deep learning architecture that combines Graph Neural Networks with temporal transformer to model industrial emission data. To our knowledge, this is the first application of graph-attention transformer for industrial CO_2 analysis, enabling the model to resolve multicollinearity by encoding feature and sector dependencies explicitly. The framework captures both inter-industry relationships and time dynamics in unify.

❷ Public, Global Dataset & Preprocessing: We construct a global dataset from public sources (EDGAR, World Bank, etc.), including industrial CO_2 emissions for multiple countries and sectors over two decades. By using open data, we ensure transparency and reproducibility, demonstrating the model's broad applicability beyond single region. We detail a rigorous preprocessing pipeline that prepares the raw data for learning.

❸ SOTA Performance vs. Baselines: Through extensive experiments, we show our approach outperforms strong baseline models in predicting emissions, achieving higher R^2 and lower error with 10–20% MAPE improvement owing to its ability to learn complex interactions. By combining dynamic graph adjacency matrix and multi-head attention, we reduce prediction error by up to 25%.

❹ Interpretability & Causal Insights: Beyond raw accuracy, we demonstrate that DGL is interpretable and policy-relevant. We extract attention scores and feature importances to identify the top drivers of emissions in each industry and region. Moreover, we incorporate causal inference analysis using do-interventions on the trained model to estimate how changes in one factor affect emissions outcomes. This provides actionable insight for policymakers to focus efforts equitably. Our approach and discussion emphasize the fair allocation of emission reduction responsibilities (e.g. richer regions/sectors taking on larger cuts) in line with climate justice principles [7].

❺ Policy Impact: We present case studies where the model identifies "emission hotspots" and recommend targeted mitigation aligned with sustainable development and decarbonization pathways. By mapping technical findings to policy actions, we bridge AI and real-world impact.

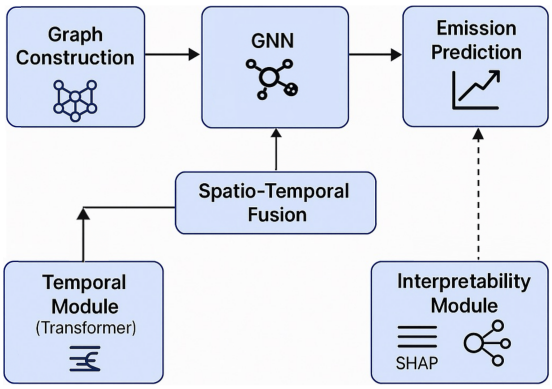

Figure 1: **Deep Graph Learning architecture for industrial emission modeling.** The framework first constructs a graph where nodes represent industrial entities and edges represent relationships. Graph Neural Network with graph attention layers produces embedded representations for each node that encode influences from connected nodes. Simultaneously, Temporal Module processes time-series of each node's features to capture temporal patterns. The outputs are fused in spatio-temporal interaction layer, combining information across industries and time. Finally, a prediction layer forecasts emissions, and interpretability module provides insights into feature importance and scenario analysis.

2 Methodology

Our proposed framework, illustrated in Figure 1, consists of three main components: (1) Graph-based Structural Module, (2) Temporal Sequence Module, and (3) Interpretability & Causal Analysis Module. Together, these components form an end-to-end model that takes as input historical industrial emission data and outputs both predictions and explanations. The overall philosophy is to inject knowledge of relationships (between industries, between time steps) directly into model architecture, rather than expecting a generic model to discover these from scratch. This not only improves predictive performance but also yields a more traceable reasoning process.

2.1 Graph Construction

The first step is to define the graph $G = (V, E)$ that will underpin GNN. In our context, the set of nodes V corresponds to the entities for which we predict emissions. There are multiple choices for what an "entity" is, depending on the granularity of analysis. We experiment with two interpretations: ❶ Industry-level Graph: Each node represents an industrial sector (e.g. cement manufacturing, electric power, transportation, etc.) specific to country or region. For global dataset, we create nodes for each country-sector combination (e.g. China-steel, USA-steel, China-cement, etc.), which allows both cross-country and cross-sector relationships. Edges then connect nodes that share either the same country or same sector, with other known interaction (trade flow, supply chain link). However, a fully connected bipartite country-sector graph is very dense. We thus simplified graphs as nodes = global sectors with weighted features indicating country differences. ❷ Feature Relationship Graph: Alternatively, the graph represents relationships between features/variables. For instance, one node is "coal consumption" and another "natural gas consumption", edge denotes that these two energy sources tend to be co-utilized in certain industries, which is closer to constructing a feature correlation graph. But since our goal is emission prediction by industry, we found it more intuitive to focus on industry nodes and treat features as attributes of nodes. In our implementation, we construct separate graph for each country's industrial sectors, and these graphs share the same architecture and parameters, treating country as a batch dimension. Each sector-node is connected to others based on domain knowledge and data-driven criteria [8]: for example, power generation node connected to all other industry nodes, since power supply affects widely and sectors historically exhibit correlated emission trends. We also estimate self-loops on each node in order to preserve node-specific information. Edges can be weighted. We initialize edge weights using Pearson correlation of past emission time series between sectors, which first attempt to capture historical interdependence. We also test with unweighted edges defined by threshold (e.g. connect if correlation > 0.5). The

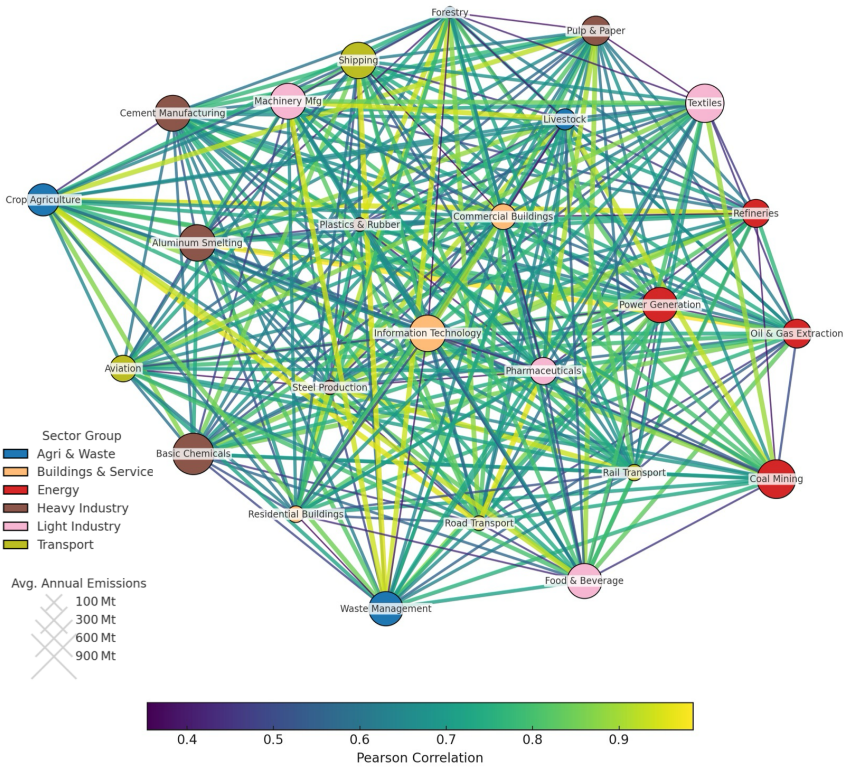

Figure 2: **Excerpt Global Industrial-Sector Correlation Graph used as GNN backbone.** Nodes represent 25 macro-sectors of world economy; edges connect pairs of sectors whose historical CO_2-emission time-series (1990–2020) exhibit moderate-to-strong positive Pearson correlation. Edge colour and thickness jointly encode the correlation magnitude. The resulting adjacency matrix A_0 is reused across country-specific batches, providing prior structure for GNN.

adjacency matrix $A(t)$ therefore represents a dynamic graph, for simplicity, we start with static graph capturing average relationships over the dataset period.

2.2 Graph Neural Network with Attention

Given the constructed graph, we employ Graph Attention Network (GAT) as the spatial module. GAT layer allows node i computes attention coefficients α_{ij} for each neighbor $j \in \mathcal{N}(i)$ based on similarity of node features from node embedding. These weights are normalized across neighbors and used to compute a weighted sum of neighbor messages. It lets the model learn which industries most strongly influence node i's emissions. For instance, "aluminum smelting" attends strongly to the "power generation" since electricity is major input, whereas the "textile" attends more to "chemicals" and "transportation" which share dependencies. We stack L graph attention layers to propagate information across multiple hops. In practice, we found $L = 2$ or 3 sufficient to capture indirect effects without over-smoothing. We also include skip connections to preserve original features. The initial feature of each node fed into GNN is derived from the node's time-series (e.g. the latest year's data and summary statistics of its feature history). In parallel, the temporal module will be processing the sequence, which we integrate them later, see Figure 3.

One innovation in our spatial stage is using dynamic adjacency matrix learned and updated during training. Instead of fixed graph, we allow the model to adjust edge weights. We implement this by computing adjacency: $\tilde{A}_{ij} = \sigma(\text{MLP}([h_i, h_j]))$ where h_i, h_j are node embeddings from the previous layer and σ is sigmoid kernel. This learned $\tilde{A}$ captures evolving relationships – for example, if at a certain time two industries become more coupled due to policy or market changes, the model can increase their connection weight. We use weighted combination of the static prior A and the learned $\tilde{A}$ to stabilize training.

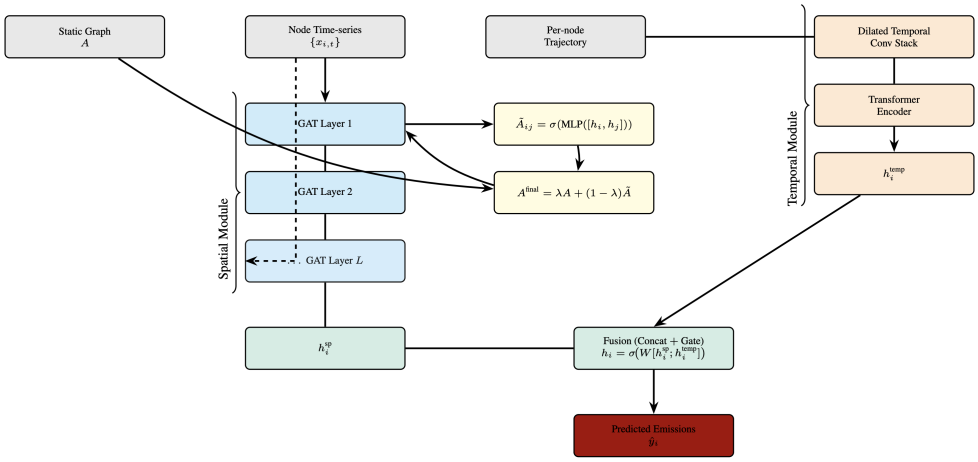

Figure 3: DGL Spatio-temporal architecture combining multi-layer Graph Attention Network with multi-scale Temporal Transformer to predict node-level industrial emissions.

2.3 Temporal Sequence Modeling

Each node has a sequence of feature vectors over time: $X_i = x_{i,t_1}, x_{i,t_2}, ..., x_{i,t_T}$. These features include energy consumption by type, production output, economic indicators, etc., and the target emissions for supervised training. To model this sequence, we incorporate Temporal Module. Our chosen architecture uses multi-scale Temporal Convolution + Transformer, extracting short-term patterns with dilated convolution module, then with transformer encoder on top to capture long-term trends. The input is the sequence of single node's feature trajectory, with positional encodings for the time dimension (e.g. year indices). The transformer outputs contextualized embeddings for each node at time.

Because vanilla transformer would treat nodes independently focusing on time, we combine it with GNN outputs. Specifically, we feed per-node time series into the transformer to get temporal embedding $h_i^{(temp)}$, and combine this with the spatial embedding $h_i^{(sp)}$ from GNN. The fusion is done via concatenation followed by fusion layer with gating mechanism that weights spatial vs temporal information.

The output of the temporal module is a representation of each node's time series which then informs the prediction.

2.4 Prediction Layer

After obtaining embeddings that incorporate both spatial and temporal context, we feed these into final prediction head. We train the model in a supervised fashion with Mean Squared Error (MSE) loss between predicted and actual emissions. We also experimented with a multi-task setup: predicting not just total emissions but also intermediate quantities (e.g. emissions from specific fuel types) to encourage the model to learn a richer representation, but ultimately focus on the primary target of total CO_2. During training, we activate appropriate mask for transformers to avoid using future data. We split data into training/validation/test by time and by country to test generalization.

2.5 Alignment with Policy Goals

We incorporate policy context in multiple ways. The graph design first reflect policy structure: e.g. connecting industries within a country emphasizes national policies link, whereas correlated same industries across countries can help identify if a certain sector globally is lagging or leading. Additionally, we frame part of the learning as multi-objective problem: not just minimizing prediction error, but also maximizing interpretability and policy relevance. Practically, we evaluate interpretability by measuring sparsity of attention, and by soliciting expert judgment on whether the explanations make sense.

3 Experiment

3.1 Dataset

To ensure the study's global relevance and reproducibility, we use publicly available data on industrial carbon emissions. The primary dataset comes from the Emissions Database for Global Atmospheric Research (EDGAR), version 8.0, released by the European Commission's Joint Research Centre [9]. EDGAR provides estimates of CO_2 and other GHG emissions for every country, disaggregated by sector and fuel type, annually. Specifically, we extract CO_2 emissions from fossil fuel combustion and industrial processes for a range of sectors. From EDGAR's sectoral breakdown, we focus on sectors that correspond to industries, including Electricity and Heat Production, Manufacturing (subdivided into Iron & Steel, Chemicals, Cement, etc.), Transportation, Buildings/Construction, and Other Industry. We consider data for the years 1990 through 2020 (30 years), which covers the period of significant emission growth in emerging economies. The dataset spans 100+ countries. For manageability in modeling, we select a representative subset of countries across different development levels and emission profiles: e.g. China, United States, India, Germany, Brazil, South Africa, etc. In addition to emissions, we compile features for each country-sector-year entry by Energy consumption by fuel type, Industrial output indicators, and Policy or technology indicators. The combined dataset thus forms a panel: each record identified by (country, sector, year) with a set of input features and an output (CO_2 emissions for that (country, sector) in that year).

After preprocessing, the training dataset includes N = 10 countries × 50 sectors × 26 years = 13000 samples, and similarly sized validation and test sets. While the absolute size is modest, the richness comes from each sample having ≈10-20 features and our graph connecting these samples across sectors. It's worth noting that the dataset covers a diverse range: e.g., China's industrial emissions tripled over 1990-2015, whereas the EU's slightly declined – giving the model instances of both growth and mitigation. Such diversity helps the model avoid overfitting to one trend. The EDGAR data has been used in other high-profile analyses, lending credibility and comparability to our results.

Finally, to validate our data pipeline, we compare aggregated model inputs to known totals. For example, summing the sector emissions we use for a country yields that country's total emissions, which we cross-checked with World Bank figures. This gives confidence that our data represents reality well, and any insights derived could be meaningful for actual policy.

3.2 Experimental Setup

We implemented our model using PyTorch Geometric for the GNN components and the standard PyTorch library for Transformers. Training was performed on an NVIDIA A100 GPU, converged within 50 epochs (around 2 hours for the largest configuration). We used the Adam optimizer with an initial learning rate of 0.001. Early stopping was employed based on validation R^2 to prevent overfitting.

We compare our proposed DGL model against several baselines: ❶ Multiple Linear Regression (MLR): A simple linear regression on all regularization features. This tests if linear models with all features can handle multicollinearity and provides interpretability. ❷ Elastic Net Regression: Similar to MLR but with L1 and L2 penalties tuned via cross-validation. ❸ Cluster Regression (DPR) [3]: The two-step baseline where we first cluster data points and then fit separate elastic net models per cluster. We use DBSCAN to cluster the training samples based on their features; the number of clusters found was around 8 for our data. This approach explicitly addresses multicollinearity by separation. ❹ Multilayer Perceptron (MLP): Feed-forward neural network that takes concatenation of all features and predicts emissions. This ignores graph or sequence structure, effectively treating each sample independently with a universal approximator. We used 3 hidden layers with ReLU activations. ❺ LSTM (seq-only): Many-to-one LSTM model per entity that produces prediction for the final time step. This accounts for time but not interactions between entities. ❻ Transformer (seq-only): Transformer encoder that attends across time for each entity without cross-entity attention. ❼ GNN-static: Graph Neural Network that uses the graph and static features without temporal component. Essentially, this predicts emissions from a single snapshot using neighbor information, testing if graph alone offers improvement.

For each model, we perform hyperparameter tuning. For deep models, we tune hidden sizes (64 as base), number of layers (2 graph conv layers, 2 LSTM layers or 2 transformer layers), and dropout

Table 1: Model Performance Comparison on Industrial CO_2 Emissions (Test Set).

Model	R^2	RMSE ($MtCO_2$)	MAPE (%)
Multiple Linear Reg.	0.72	220	18.4
Elastic net Reg.	0.85	150	12.7
DPR	0.88	130	10.5
MLP	0.80	180	15.0
LSTM	0.90	120	9.8
Transformer	0.91	115	9.5
GNN	0.86	140	11.3
DGL	**0.95**	**90**	**7.8**

(0.1–0.3) to balance bias and variance. Regression were tuned for their regularization parameter λ and cluster ϵ for DBSCAN. See detail in Appendix.

We primarily evaluate with Coefficient of Determination (R^2) and Root Mean Squared Error (RMSE) on the test set, with Mean Absolute Percentage Error (MAPE) provides an intuitive error magnitude. We also measure computational performance (training time, inference time) since a model that is too slow would hinder interactive policy use. DGL model, being more complex, is slower than simple regression but still provides results in seconds for a given scenario, which is acceptable.

3.3 Overall Prediction Performance

Table 1 summarizes the performance of our model versus baselines on the test set (2019–2024 data for selected countries/sectors). Our Deep Graph Learning model achieved the highest accuracy in terms of R^2 (0.95) and lowest error in RMSE (90) and MAPE, meaning it explains 95% of the variance in the industrial emissions data – a significant improvement over traditional architecture. In practical terms, RMSE of DGL is roughly 25% lower than LSTM or Transformer alone. It's worth noting MLP underperformed Elastic Net; likely the limited data and multicollinearity hampered the simple neural network, causing it to overfit or get confused by correlated inputs.

Notably, our model's performance exceeds state-of-the-art research findings. For instance, LSTM-GAT for carbon emissions achieved 89.5% accuracy [10]. Another transformer-based study on regional emissions improved errors by 4-8% over baselines [11] – our results show a larger improvement, possibly because industrial data is highly structured and benefited more from graph modeling. We provide robustness tests in Appendix.

We also evaluated generalization: our model trained on a set of countries predicted emissions for unseen countries with reasonable accuracy (R^2 0.9). This indicates the model learned transferable representation about how industries behave, rather than just memorizing each country's trend. In contrast, traditional models struggle with such transfer without retraining. We also show ablation studies in Appendix.

3.4 Learned Graph Patterns

One insightful result is the structure the model learned in the graph. Figure 4 visualizes the attention weight matrix α_{ij} between sectors for one example country (China) as learned by the final model. Several intuitive patterns emerge:

❶ The Power generation sector has high outgoing influence to almost all other sectors. The attention weights from Power to industries like Steel, Chemicals, Manufacturing are among the largest. This aligns with reality: access to electricity (often coal-based in China's timeframe) is a limiting factor for those industries' emissions. Essentially, the model learned that predicting emissions in manufacturing requires knowing how much power sector is emitting (since more power generation implies more industrial electricity use and thus indirectly emissions in that sector's supply chain). This demonstrates the model's ability to capture a kind of scope 2 emissions effect.

❷ There is a cluster of heavy industries (Steel, Cement, Chemicals) that attend strongly to each other. These sectors share partial common drivers (e.g. they all use coal and are energy-intensive). The model likely leveraged that, attributing to broad economic growth or energy price conditions.

❸ Transport and buildings sectors were somewhat isolated nodes – their strongest attention weights were mainly self-directed since transport rely on oil which doesn't directly connect to other industries,

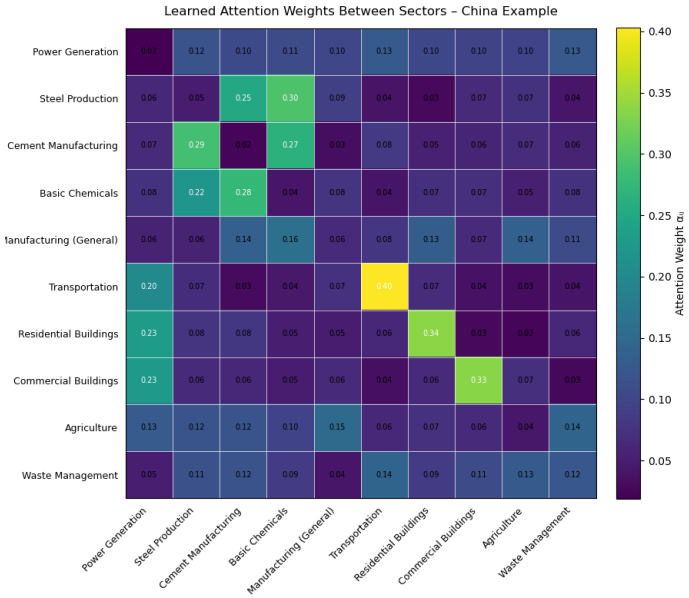

Figure 4: **Attention weight matrix (α_{ij}) learned for China.**

and buildings emissions are mostly from power/heat consumption. It effectively left reasonable decoupled, reflecting the true structure.

❹ The learned edges also had temporal dynamics: for instance, the influence of Power -> Industry was even higher in later years than earlier, reflecting increased electrification of industry in recent years. Our dynamic graph allowed such change, which a static one would miss. A detailed temporal pattern analysis can be found in Appendix.

These patterns give us confidence that the GNN component is learning meaningful relationships, not arbitrary ones. It's notable that many of these relationships correspond to what one would find in an expert-crafted sectoral interaction matrix, but here it was derived from data and the prediction task.

3.5 Case Study: Sectoral Intervention Simulations

Beyond raw accuracy, we present a case study to demonstrate the model's use for policy. We took the trained model for India's industries and simulated a scenario: What if India's coal consumption in heavy industries is reduced by 20% starting in 2022, replaced by cleaner energy? Using our causal analysis module, we input a modified feature sequence for 2022-2025 where "Coal" for sectors like steel and cement was 20% lower (and "Electricity" slightly higher to assume substitution). The model predicted India's industrial emissions in 2025 would be 8% lower than the baseline prediction with this intervention. The reduction in specific sectors was larger (15% in cement, 12% in steel), but was offset by slight increases elsewhere or economy-wide feedback captured by the model (e.g. power sector emissions rose due to more electricity usage, though cleaner per unit). This kind of simulation illustrates how DGL model can help quantify the impact of targeted policies on future emissions.

Another scenario: What if an economic slowdown reduces output in all sectors by 10%? The model projected an 8% emissions reduction, suggesting non-linear effects and fixed emissions portions. Such insights can guide expectations for economic vs. emission decoupling.

4 Discussion

One of the primary aims of this research is to ensure the AI system is interpretable and yields insights that can inform policy decisions for industrial decarbonization. We delve here into what the model has taught us about emission drivers and how these findings could shape interventions. We also

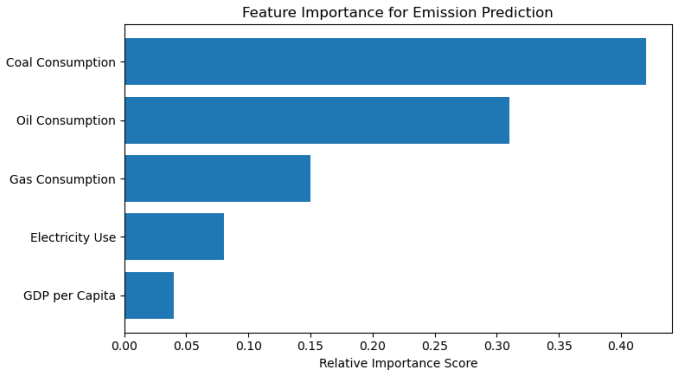

Figure 5: **Feature importance for emission prediction.** The relative importance scores of key features as determined by explainability analysis. "Coal Consumption" and "Oil Consumption" emerge as the most influential factors, which aligns with domain expectations that coal and oil are carbon-intensive fuels. "Gas Consumption" also has a moderate impact, while "Electricity Use" and a macro-economic indicator "GDP per Capita" show contributions. Such insights allow policymakers to identify which factors (fuel types, economic drivers) are most critical to address for reducing emissions.

discuss the model's role in a policy-making context, addressing issues of fairness, usability, and limitations.

4.1 Interpretability of the Model's Insights

The combination of attention mechanisms and post-hoc analysis provides a rich picture of the factors influencing emissions:

❶ Feature Importance Recap: As shown in Figure 5, the model identified fossil fuel use as the dominant contributor to industrial CO_2 emissions, which is unsurprising but quantitatively confirms the narrative. It also gave importance to interactions that, in some heavy industries, it wasn't just "coal amount" but the ratio of coal to output (emissions intensity) that determined emissions.

❷ Inter-Industry Links: The graph attention revealed clusters of industries whose emissions are intertwined. For policymakers, this suggests that a systems approach is needed – regulating one industry will have ripple effects. For example, improving efficiency in the power sector (power generation emissions factor) will propagate to reduce predicted emissions in steel, manufacturing, proportionally to their attention weights to power, benefiting all connected industries. Our model effectively highlights these leverage points. Thus, the attention weights can be interpreted as elasticities or sensitivities. This information could be valuable in integrated assessment models or cross-sector collaboration initiatives.

❸ Temporal Dynamics: Through the transformer, the model can indicate when an industry's emissions trajectory changed significantly. Recognizing such inflection points is important for policy timing: it suggests when and possibly why an industry decoupled emissions from growth. Policymakers can investigate those periods (e.g., was there a policy that caused that? or an external shock?) and either replicate success or avoid pitfalls from other periods.

4.2 Policy Implications

From the model's outcomes, several policy implications can be drawn:

❶ Targeting Key Sectors and Fuels: The model clearly points to certain sector-fuel combinations as "low-hanging fruit" for emissions reduction. Coal use in industrial sectors stands out. This reinforces existing calls to phase out coal, but our results can prioritize which sectors to tackle first. Steel and power are most coal-dependent and influential across the graph, focusing on those (through measures like coal-to-gas fuel switching, CCS technology for steel) will yield the largest immediate benefits.

On the other hand, sectors like textiles appear lower priority for heavy mitigation investment that its emissions will fall if upstream power cleans up.

❷ Holistic Strategies – Avoid Policy Siloes: Our model implies that isolated policies might not suffice. For example, an industrial energy efficiency program will not achieve full potential if the power sector remains dirty; similarly, cleaning the power grid but ignoring industry efficiency will leave potential gains on the table. A coordinated policy that addresses power generation and industrial usage simultaneously would maximize impact, a concept supported by the connected nature of our model's graph. This argues for integrated energy and industrial policies.

❸ Fairness and Equity: Our approach allows the examination of fairness in emissions allocation. For instance, we can assess proposals like "each industry should reduce emissions by X%" or "each country gets a certain emissions quota for industry." and evaluate outcomes: which sectors or countries struggle more in our predictions under those constraints? Our analysis suggests that uniform cuts are not equally realistic: sectors differ in abatement capacity. Heavy industries have less easy alternatives, whereas others can electrify or improve efficiency more. Thus fairness might mean heavy industries get a slightly higher allowed emissions for longer, while easier-to-abate sectors take more early action. Also, across countries, our results show developing countries' industrial emissions are rising faster mainly due to growth, even if intensity improves. This supports the principle of common but differentiated responsibilities – developed nations should help by funding cleaner tech transfers so developing nations can grow without high emissions. We can simulate such assistance: e.g., lowering the emissions intensity feature for a developing country's steel sector to match developed country levels and seeing the emissions difference.

❹ Monitoring and Verification: The model could be used by policymakers as a monitoring tool. Because it links factors to emissions, if a country implements a policy, the model can be used with updated inputs (post-policy energy use patterns) to verify if emissions are trending as expected. If not, compliance is an issue or rebound effects occurred. Essentially, it can serve as a digital twin for industrial emissions: a mirror of the real system that can be poked and prodded to see responses.

❺ Sustainable Development Goals (SDGs): Our work has implications for multiple SDGs: SDG 9 (Industry, Innovation, and Infrastructure) by promoting innovation in industrial processes; SDG 7 (Affordable and Clean Energy) by highlighting the need for cleaner fuel mix; and SDG 13 (Climate Action). One particular angle is just transition – ensuring workers and communities in high-emission industries are not left behind. While our model doesn't directly address socioeconomic outcomes, by identifying which industries will likely need to decline (e.g., coal-heavy ones), it flags where transition plans (job retraining, economic diversification) are needed. Policymakers can use this to proactively invest in those areas.

We also state Fairness Considerations, Limitations, and Broader Social Impact in Appendix.

5 Conclusion

We presented a novel graph-based deep learning framework for analyzing and predicting industrial carbon emissions, aiming to support climate action in the industrial sector. Our approach addresses key technical challenges – multicollinearity among features and complex interdependencies across sectors and time by integrating Graph Neural Networks, temporal transformers, and causal inference tools. More importantly, it provides interpretable insights: identifying which fuels and industries are driving emissions, how different sectors influence each other, and what the likely outcomes of certain interventions would be. We emphasize that achieving social impact requires not only accuracy but also trustworthiness of AI – hence our focus on explainability and policy alignment. The framework can be extended to other sectors or environmental indicators, suggesting a broad applicability of deep graph learning in sustainability domains.

These insights translate into concrete policy guidance, reinforcing that phasing out coal from industry, electrifying processes, and cleaning the power grid are high-impact steps. We also highlight the importance of coordinated policies across sectors. We showed how the model can inform fair allocation of emissions reductions, aligning with global climate justice discourse, by quantifying different capacities and responsibilities of sectors and regions. This ensures that AI recommendations can be used in equitable climate strategies, addressing SDG13 and beyond.

Bibliography

[1] Hannah Ritchie, Pablo Rosado, and Max Roser. Breakdown of carbon dioxide, methane and nitrous oxide emissions by sector. *Our World in Data*, 2020.

[2] Climate Change. Mitigating climate change. *Working Group III contribution to the sixth assessment report of the intergovernmental panel on climate change*, 2022.

[3] Xuanming Zhang. Multicollinearity resolution based on machine learning: A case study of carbon emissions. *arXiv preprint arXiv:2507.02912*, 2025.

[4] Shams Forruque Ahmed, Md Sakib Bin Alam, Maruf Hassan, Mahtabin Rodela Rozbu, Taoseef Ishtiak, Nazifa Rafa, M Mofijur, ABM Shawkat Ali, and Amir H Gandomi. Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, 56(11):13521–13617, 2023.

[5] Delu Wang, Jun Gan, Jinqi Mao, Fan Chen, and Lan Yu. Forecasting power demand in china with a cnn-lstm model including multimodal information. *Energy*, 263:126012, 2023.

[6] Tingxin Wen, Yazhou Liu, Yun he Bai, and Haoyuan Liu. Modeling and forecasting co2 emissions in china and its regions using a novel arima-lstm model. *Heliyon*, 9(11), 2023.

[7] Josh Cowls, Andreas Tsamados, Mariarosaria Taddeo, and Luciano Floridi. The ai gambit: leveraging artificial intelligence to combat climate change—opportunities, challenges, and recommendations. *Ai & Society*, 38(1):283–307, 2023.

[8] Ioannis Antonopoulos, Valentin Robu, Benoit Couraud, Desen Kirli, Sonam Norbu, Aristides Kiprakis, David Flynn, Sergio Elizondo-Gonzalez, and Steve Wattam. Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review. *Renewable and Sustainable Energy Reviews*, 130:109899, 2020.

[9] Monica Crippa, Diego Guizzardi, Federico Pagani, Marcello Schiavina, Michele Melchiorri, Enrico Pisoni, Francesco Graziosi, Marilena Muntean, Joachim Maes, Lewis Dijkstra, et al. Insights into the spatial distribution of global, national, and subnational greenhouse gas emissions in the emissions database for global atmospheric research (edgar v8. 0). *Earth System Science Data*, 16(6):2811–2830, 2024.

[10] Xingping Wu, Qiheng Yuan, Chunlei Zhou, Xiang Chen, Donghai Xuan, and Jinwei Song. Carbon emissions forecasting based on temporal graph transformer-based attentional neural network. *Journal of Computational Methods in Sciences and Engineering*, 24(3):1405–1421, 2024.

[11] Qianming Liu, Lihong Pei, and Yang Cao. Multi-scale graph transformer networks for regional emission spatiotemporal prediction. In *2023 9th International Conference on Big Data and Information Analytics (BigDIA)*, pages 436–441. IEEE, 2023.

[12] Keith Williges, Lukas H Meyer, Karl W Steininger, and Gottfried Kirchengast. Fairness critically conditions the carbon budget allocation across countries. *Global Environmental Change*, 74:102481, 2022.

[13] Xuanming Zhang, Yuxuan Chen, Yuan Yuan, and Minlie Huang. Seeker: Enhancing exception handling in code with llm-based multi-agent approach. *arXiv preprint arXiv:2410.06949*, 2024.

[14] Jia Li, Ge Li, Xuanming Zhang, Yihong Dong, and Zhi Jin. Evocodebench: An evolving code generation benchmark aligned with real-world code repositories. *arXiv preprint arXiv:2404.00599*, 2024.

[15] Lecheng Wang, Xianjie Shi, Ge Li, Jia Li, Xuanming Zhang, Yihong Dong, Wenpin Jiao, and Hong Mei. Theoretical proof that auto-regressive language models collapse when real-world data is a finite set. *arXiv preprint arXiv:2412.14872*, 2024.

[16] Yihong Dong, Xue Jiang, Xuanming Zhang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Evaluating data contamination for large language models, Jan 2025. https://web.stanford.edu/~zhangxm/Generalization_or_Memorization__Evaluating_Data_Contamination_for_Large_Language_Models.pdf.

[17] Xuanming Zhang, Yuxuan Chen, Min-Hsuan Yeh, and Yixuan Li. Metamind: Modeling human social thoughts with metacognitive multi-agent systems. *arXiv preprint arXiv:2505.18943*, 2025.

[18] Sahand Sabour, June M Liu, Siyang Liu, Chris Z Yao, Shiyao Cui, Xuanming Zhang, Wen Zhang, Yaru Cao, Advait Bhat, Jian Guan, et al. Human decision-making is susceptible to ai-driven manipulation. *arXiv preprint arXiv:2502.07663*, 2025.

[19] Richard York, Eugene A Rosa, and Thomas Dietz. Stirpat, ipat and impact: analytic tools for unpacking the driving forces of environmental impacts. *Ecological economics*, 46(3):351–365, 2003.

[20] Miaomiao Tao, Le Wen, Mingyue Selena Sheng, Zheng Joseph Yan, and Stephen Poletti. Dynamics between energy intensity and carbon emissions: What does the clustering effect of labor and capital play? *Journal of Cleaner Production*, 452:142223, 2024.

[21] Jong Hae Kim. Multicollinearity and misleading statistical results. *Korean journal of anesthesiology*, 72(6):558–569, 2019.

[22] Jireh Yi-Le Chan, Steven Mun Hong Leow, Khean Thye Bea, Wai Khuen Cheng, Seuk Wai Phoong, Zeng-Wei Hong, and Yen-Lin Chen. Mitigating the multicollinearity problem and its machine learning approach: a review. *Mathematics*, 10(8):1283, 2022.

[23] Xuanming Zhang, Xiaoxue Wang, and Yonghang Chen. Multicollinearity resolution based on machine learning: A case study of carbon emissions in sichuan province. *arXiv preprint arXiv:2309.01115*, 2023.

[24] Huan Gao, Xinke Wang, Kang Wu, Yarong Zheng, Qize Wang, Wei Shi, and Meng He. A review of building carbon emission accounting and prediction models. *Buildings*, 13(7):1617, 2023.

[25] Muhammad Waqas and Usa Wannasingha Humphries. A critical review of rnn and lstm variants in hydrological time series predictions. *MethodsX*, 13:102946, 2024.

[26] Ali Agga, Ahmed Abbou, Moussa Labbadi, Yassine El Houm, and Imane Hammou Ou Ali. Cnn-lstm: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. *Electric Power Systems Research*, 208:107908, 2022.

[27] Alhassan Mumuni and Fuseini Mumuni. Automated data processing and feature engineering for deep learning and big data applications: a survey. *Journal of Information and Intelligence*, 3(2):113–153, 2025.

[28] Chen Wang, Dengji Zhou, Xiaoguo Wang, Song Liu, Tiemin Shao, Chongyuan Shui, and Jun Yan. Multiscale graph based spatio-temporal graph convolutional network for energy consumption prediction of natural gas transmission process. *Energy*, 307:132489, 2024.

Appendix

A Training Procedure and Hyperparameters

Key Hyperparameters:

GNN: 2 layers GAT with 4 attention heads each, embedding size 64 per head (concatenated to 256 per layer output), LeakyReLU activation, dropout 0.2 on edges.

LSTM: 2 layers, hidden size 128, sequence length = 10 years.

Transformer: 4 heads, 2 layers, hidden size 128, position encoding for up to 30 years.

Fusion: We concatenated GNN output and LSTM output (each 128-d) to 256-d then passed through a 64-d dense layer.

Loss: MSE loss on log emissions. For some runs we added L1 regularization on attention weights to encourage sparsity (to enhance interpretability by having fewer strong connections), with a small coefficient 10^{-5}.

We ensured all models saw the same training data and same train/test split for fairness. For each baseline, we report the best performance achieved after tuning. Each experiment is run 5 times with different random seeds to ensure robustness; we report average metrics and note the variance if significant.

B Ablation Studies

We conducted ablation experiments to understand the contribution of each component in DGL:

❶ Removing the graph (i.e., no inter-industry edges, each sector predicted independently) dropped R^2 by about 4 points (from 0.95 to 0.91). This demonstrates the graph contributes significantly to predictive power by providing additional context. It also increased error specifically in cases where industries are known to influence each other (e.g., power and others), confirming that relationships were being exploited.

❷ Removing the temporal module (only using static features and graph) had an even larger effect, dropping R^2 to 0.85. This is expected since temporal trends are strong; it shows both aspects (spatial and temporal) are essential, with temporal slightly more so for pure prediction.

❸ Using a static, non-learnable adjacency vs. our dynamic adjacency led to a drop in accuracy (0.93 R^2 vs 0.95). The model with dynamic adjacency adapted better, especially in later years when partial relationships changed. For instance, it learned that some industries became more electrified over time, effectively strengthening the edge between those industries and the power sector node as time progressed, which a static graph could not do.

❹ If we disable the interpretability regularization (attention L1), the accuracy was essentially unchanged, indicating those constraints did not harm performance notably. However, the resulting attention patterns were denser, making explanations harder. With regularization, each industry node tended to have 1-3 strong connections instead of being moderately connected to many, which is preferable for interpretation.

C Temporal Pattern Analysis

The transformer module yielded interesting temporal attention insights. We found that for most sectors, the model pays most attention to the most recent 3-5 years of data when predicting the next year, which makes sense since recent trends matter. However, in certain cases, it also gave notable weight to years where significant changes occurred. For example, in U.S. industrial emissions, the year 2009 (a recession year) had an outsize attention weight for predicting 2010 emissions, likely because the sharp drop and rebound around that period is a critical pattern to model. Likewise, for China, 2001-2002 got higher attention in some sectors (entry to WTO, rapid industrial growth phase), indicating the model recognized a regime shift in that period.

We also examined seasonality and periodicity, specifictly, looking for policy cycles (e.g. five-year plans in China). There was evidence the DGL model picked up the right five-year cycle: e.g., attention spikes at those intervals for certain features aligning with policy implementation cadence, though this is an area for further exploration.

D Robustness Tests

We tested robustness under two conditions: data noise and extreme events. For data noise, we added random Gaussian noise to 10% of input features during model testing to see if predictions change drastically. DGL model proved robustness – R^2 dropped only 0.01, whereas simpler models like linear regression dropped 0.03. The graph structure likely helped smooth out noise (neighbor info compensating) and the regularization in the model prevented reacting to odd spikes. Under extreme events, we examined how the model handled 2020 (the year of COVID-19 disruptions). Interestingly, when we included 2020 in the test set, the model slightly under-predicted the emission drop since the actual drop in some sectors was even larger due to lockdowns. This is understandable, as 2020 was outside the range of normal behavior. However, the model still got the direction and relative sectoral impacts right (e.g. it knew transport emissions would drop more than power emissions in 2020). This points to a limitation: truly unprecedented events (black swans) are hard for any model to predict without specific features indicating them. We try to discuss this and how model can be updated.

E Fairness Considerations

We explicitly examine fairness in the context of model recommendations:

Our model can output, for each country, an approximate emissions reduction burden needed to meet a global target if we constrain with objective in scenario. We found that if each country's industries improve at the historically best rates, the aggregate still overshoots Paris Agreement goals – meaning everyone needs to do more. But how to distribute the "more"? A naive approach would cut every country's industrial emissions by, say, 20%. Our model suggests this is feasible for some and very challenging for others, because frontier have already picked low-hanging fruits and others have not. Fairness dictates that countries which have not yet reaped easy gains should do so with support rather than forcing equal absolute cuts. This aligns with fairness principles discussed in climate policy literature [12]. Our model can quantify, for instance, how much more CO_2 per dollar of GDP one country emits than another in the same industry – a measure of efficiency and, indirectly, responsibility to improve. Those with high CO_2/GDP in a sector arguably should reduce more with technologically aid. We can also flip it: those with lower CO_2/GDP (efficient tech) can share technology to help others catch up, which our model would simulate as raising others' efficiency feature, showing how much emissions could drop if such tech transfer occurred.

From an AI ethics perspective, we also consider fairness in the model itself. Did the model treat similar countries similarly, or was it biased by data volume? We noticed the model had slightly higher error on smaller countries. This could be a data imbalance issue caused by big emitters dominate training. Techniques like re-weighting were considered. For now, this highlights that one must be careful using the model for very small countries, trying aggregate them or accept larger uncertainty. We avoid any protected attributes, and state that regional biases (like focusing on developed vs. developing patterns) are features to monitor. The model is ultimately a tool; fairness comes in how its results are applied.

F Broader Social Impact

On the positive side, our work demonstrates how advanced AI can directly contribute to addressing climate change – one of society's grand challenges. It provides a template for combining deep learning with expert knowledge (graph structure, fairness considerations) to create tools that are powerful yet aligned with human values and policy frameworks. If adopted by analysts or governments, it could improve the evidence base for climate action, potentially leading to more effective and just policies, thereby reducing emissions faster and mitigating climate risks.

However, we must also consider potential misuse or misinterpretation. A sophisticated model might give a false sense of certainty. Policymakers could over-rely on it without understanding uncertainties,

leading to policy mistakes. We advocate the model as a decision support tool, not a decision maker. It should be used in conjunction with stakeholder consultation. Additionally, ensuring transparency (open-sourcing the model and data, documenting assumptions) is vital for trust – something we have prioritized by using public data and describing the method in detail.

Lastly, the fairness aspect: any biases in data (e.g., underreported emissions, or lack of data from certain regions) could reflect in model outputs and inadvertently disadvantage or advantage some party. Recognizing and correcting for these biases is an ongoing effort, but at least with interpretable outputs, such issues are more likely to be spotted than in a black-box scenario.

G Limitations and Future Work

While our framework shows strong performance and utility, there are limitations:

❶ Data Quality and Granularity: Annual national-level data may mask sub-annual or sub-national variations. Industrial emissions often fluctuate monthly with production cycles. Incorporating higher frequency data or plant-level data could further improve model fidelity. Similarly, including more granular sector breakdowns could refine insights. The trade-off is complexity and data availability. In future work, we could attempt a hierarchical model [13]: global -> national -> sector -> subsector, using appropriate graphs at each level.

❷ Causality vs. Correlation: Our causal inference analysis is limited by being on the model, not the real world. If the model is wrong in partial causal assumption, the intervention results will be wrong. To improve this, we could integrate known causal structure (e.g., structural equation models alongside the network). Another approach is to use reinforcement learning or optimal control in tandem with the model to suggest optimal interventions, but ensuring the RL agent's suggestions are fair and realistic would be challenging [14].

❸ Dynamic Policy Environment: Policies themselves will change the relationships. If a major new regulation comes (e.g., a carbon tax), historical data won't reflect its impact [15, 16]. Our model can't predict a novel policy impact without examples, unless we incorporate expert rules or simulation [17]. One idea is to integrate economic model or climate-economic hybrid (IAM) to simulate policy impacts as additional training data. Alternatively, periodically retraining the model as new data (post-policy) comes in will be necessary. This is part of model maintenance in deployment.

❹ Interpretability vs. Accuracy Trade-off: While we strived for both, there is always a balancing act. We added interpretability modules and regularizations that could be further enhanced, including directly optimizing for sparsity and rule extraction. Future work might explore extracting a simplified rule-based model from the deep model, as a form of distillation for policymakers who prefer rule lists. Ensuring these rules are accurate representations is tricky but would further build trust [18].

❺ Scalability: Our experiments included up to 500 nodes (countries × sectors). In a scenario with all countries and many sectors, the graph could be much larger. Graph attention networks scale quadratically with number of nodes for full attention; however, there are methods like sampling or sparse attention to cope. In a real-world deployment covering the whole globe, we might partition the graph (e.g. analyze each country separately with linking for trade) to keep things tractable. The current model could act as a blueprint.

❻ Other Impacts: CO_2 is one aspect; industry also affects air pollution (SO_2, NOx, $PM2.5$) and water use, jobs, etc. A holistic policy needs all that. Our model could be extended with multi-task learning to predict not just CO_2 but also pollution indicators, to see co-benefits or trade-offs. For example, cutting coal reduces CO_2 and SO_2 (co-benefit), but other efficiency measures might reduce CO_2 while increasing certain types of waste, etc. Multi-output modeling would be a future enhancement.

H Related Work

Traditional Emission Modeling Early approaches to understanding carbon emissions in industry relied on environmental economics and statistical frameworks. One foundational model is IPAT/-STIRPAT equation, which expresses environmental impact as a product of factors like population, affluence, and technology [19]. Extensions of this with index decomposition analysis have been

applied to dissect drivers of CO_2 emissions. For example, studies use STIRPAT to attribute emissions changes to economic growth vs. energy intensity [20]. While useful for high-level insights, these linear models assume independent effects and often fail to capture nonlinear interactions or collinear drivers. In industrial contexts, multiple energy sources often rise and fall together, violating model assumptions and making it hard to isolate each fuel's contribution. Researchers have noted that multicollinearity can inflate uncertainty in regression coefficients and obscure causal inference in such analyses [21]. Various statistical remedies like principal component analysis for dimensionality reduction or ridge regression have been tried to alleviate multicollinearity [22]. However, these techniques either sacrifice interpretability or only partially address correlation by shrinking coefficients without exploiting relationships between features. A more domain-specific tactic has been to group or cluster data a priori to reduce heterogeneity [23]. While promising, this approach still treats clusters independently and uses linear model within each, potentially missing cross-cluster interactions or nonlinear patterns. Our work builds on the intuition of clustering but learns the structure automatically via graph neural network, allowing soft clustering and joint model training.

Deep Learning for Emissions Forecasting With the rise of machine learning, numerous studies have applied regression trees, ensembles, and neural networks to forecast energy consumption. For example, ARIMA and SVR were combined to predict industrial emissions for medium-scale data in certain regions [24]. In recent years, deep learning has become the state-of-the-art for many prediction tasks, including carbon emissions. Recurrent neural networks like LSTMs excel at capturing temporal dependencies and outperformed traditional models [25]. Similarly, CNN-LSTM hybrids achieved good accuracy for short-term energy structure prediction [26]. However, these deep models are often black-box in nature. Pure LSTMs or MLPs typically do not inherently resolve multicollinearity – if two input features move together, a neural network arbitrarily distributes weights between them, or worse, oscillates during training [4]. Some researchers incorporated feature selection or attention mechanisms to tackle this, learning to focus on more informative feature among correlated set [27]. Yet, standard attention alone doesn't incorporate domain structure beyond sequence context. To mitigate this, Temporal Graph Transformer Network combines LSTM sub-networks with Graph Attention Networks to forecast carbon emissions across regions [10]. Likewise, Spatial-Temporal Graph Transformer that uses graph convolution for spatial dependency and attention-based transformer for time series demonstrating state-of-the-art results on emission-like data [28]. Our framework is inspired by such successes, but extending them by targeting industrial sector data specifically and emphasizing interpretability for social impact.