TEXT-GUIDED MULTIMODAL ATTENTIVE GRAPH LEARNING FOR TIME SERIES FORECASTING WITH EXOGENOUS VARIABLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting with exogenous variables predicts endogenous targets using historical observations and associated exogenous information. Existing studies mainly rely on numerical time series, leaving textual semantics largely unexplored. Furthermore, to improve efficiency, many methods simplify inter-variable modeling, failing to capture its full capacity, particularly exogenous-endogenous relationships. To address these limitations, we propose **MAGNet**, a text-guided **M**ultimodal **A**ttentive **G**raph **Net**work that integrates textual semantics into graph learning for time series forecasting with exogenous variables. MAGNet first employs an offline pretrained language model to generate contextual descriptions for exogenous and endogenous variables separately, providing global semantic representations beyond numerical observations. The resulting representations are incorporated into an asymmetric heterogeneous graph to better capture hierarchical inter-variable relationships. Building on this graph, predictive causality-aware attention mechanisms learn target-focused graph representations, improving forecasting performance. Experiments on 12 real-world datasets show that MAGNet achieves competitive short- and long-term forecasting performance compared to existing baselines, with ablation and case studies further supporting the effectiveness of the proposed framework. We provide the code accessible through this link: https://anonymous.4open.science/r/MAGNet-C049/

1 INTRODUCTION

Time series forecasting (TSF) with exogenous variables plays a crucial role in many real-world domains: finance (Zhu et al., 2025), healthcare (Li et al., 2025), traffic (Shuvo et al., 2021), and manufacturing (Yang et al., 2024), where future target outcomes depend on both historical target and related external factors. By convention, the target variable is the *endogenous* variable, while the associated observed covariates are the *exogenous* variables. This paper focuses on forecasting one endogenous variable and its multivariate exogenous variables. Given the history of both variables, the objective is to predict only the endogenous series (Wang et al., 2024b; Qiu et al., 2025), distinguishing this task from traditional multivariate TSF, in which all variables are predicted in an ensemble manner regardless of their role in the forecasting context. (Qiu et al., 2025). Effectively exploiting exogenous information is critical for accurate forecasting, making TSF with exogenous variables an important research topic.

Most existing methods for TSF with exogenous variables rely solely on numerical time series, leaving textual information largely unexplored. Multimodal TSF has shown that text can provide complementary context beyond numerical observations (Zhang et al., 2024; Gruver et al., 2023). For example, wind speed forecasting benefits from physical knowledge that explains the relationship between wind speed and power generation (Thorarinsdottir et al., 2019). Fig. 1 summarizes this evolution through four paradigms. Paradigm (a) represents multivariate TSF using only numerical data, while paradigm (b) extends it by separating exogenous and endogenous variables. Paradigm (c) further uses textual information but retains the symmetric variable modeling as adopted in methods such as Time-LLM (Jin et al., 2024b) and UniTime (Liu et al., 2024a). Motivated by these observations, we study the fourth paradigm, which jointly models variable separation and textual semantics to provide richer contextual information for forecasting, as also shown in existing models,

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

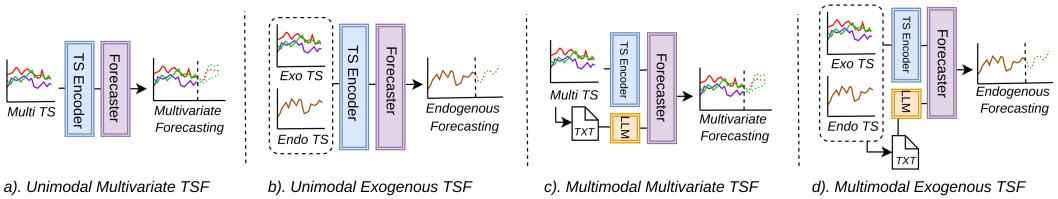


Figure 1: Four time series forecasting paradigms: (a) unimodal multivariate TSF, (b) unimodal exogenous TSF, (c) multimodal multivariate TSF, and (d) multimodal exogenous TSF (ours).

such as ExoLLM (Huang et al., 2025a), ExoTimer (Wu et al., 2026), and GraFT (Yuan et al., 2026). Beyond textual semantics, modeling inter-variable relationships is equally important. Due to the aforementioned variable-type distinctions, their interactions can be categorized into two complementary types: *exogenous-exogenous* (Exo-Exo) relationships, which capture correlations among external variables, and *exogenous-endogenous* (Exo-Endo) relationships, which characterize how external variables influence the target forecast. Existing methods typically model these interactions using attention mechanisms or numerical correlations (Qiu et al., 2025; Li et al., 2026; Wang et al., 2024b), without explicitly distinguishing their different semantic roles. Thus, effectively modeling both relationship types remains essential to appropriately exploit exogenous information.

To address this need, graph neural networks (GNNs) naturally model interactions among variables and have shown promising performance in TSF (Wu et al., 2020; Cao et al., 2020). However, existing graph-based methods are primarily designed for homogeneous, symmetric, and unimodal settings, making them less suitable for our task. These limitations motivate a text-guided, asymmetric, heterogeneous graph that explicitly preserves the distinct roles of endogenous and exogenous variables while modeling their interactions with richer semantic representations.

Therefore, we propose a text-guided **M**ultimodal **A**ttentive **G**raph **Net**work (MAGNet) for TSF with exogenous variables. MAGNet introduces textual semantics as global guidance to complement numerical observations and distinguish the semantic roles of exogenous and endogenous variables. An asymmetric heterogeneous graph explicitly models both interactions among exogenous variables and their directional influence on the forecasting target using cosine similarity and Neural Granger Causality (NGC) (Tank et al., 2022), respectively. A predictive causality-aware graph learning strategy further integrates semantic and structural information to produce target-oriented graph embedding. Together, these components provide a unified framework for multimodal TSF with exogenous variables.

Our contributions are threefold:

- We propose MAGNet, a text-guided multimodal graph learning framework for time series forecasting with exogenous variables, which jointly exploits textual semantics and asymmetric variable interactions to better model exogenous information.
- To capture complex inter-variable relationships, we further develop an asymmetric graph learning scheme that integrates heterogeneous relationship modeling with target-aware graph representation learning, enabling more accurate exploitation of exogenous information for forecasting.
- Extensive experiments on 12 real-world datasets show that MAGNet achieves the best average performance among compared methods, with the most consistent gains on backbones lacking native exogenous modeling, while comprehensive ablation studies and visualization analyses provide insights into the effectiveness of each design.

2 Related Work

2.1 Time Series Forecasting with Exogenous Variables

Time series forecasting (TSF) with exogenous variables separates the endogenous target from its external inputs, unlike multivariate TSF that treats variables symmetrically (Lima et al., 2025; Qiu et al., 2025). General forecasters: PatchTST (Nie et al., 2022), DLinear (Su, 2022), and iTransformer

(Liu et al., 2024b) do not model these roles explicitly, unlike DAG (Qiu et al., 2025), GCGNet (Li et al., 2026), and TimeXer (Wang et al., 2024b), where attention networks or numerical correlations are used to exploit exogenous information. However, they do not distinguish between bidirectional relations among exogenous variables and their directional effects on the target. Our method addresses this gap with an asymmetric, heterogeneous graph that preserves relation types and includes text-enriched nodes that provide context beyond the numerical lookback window.

2.2 Multimodal Time Series Forecasting

Multimodal TSF augments numerical sequences with information from other modalities, with text providing useful context for temporal modeling (Zhang et al., 2024; Gruver et al., 2023). Time-LLM (Jin et al., 2024b), CALF (Liu et al., 2025b), ChatTime (Wang et al., 2025a), and UniTime (Liu et al., 2024a) integrate pretrained language models into multivariate TSF, leaving the forecasting with exogenous variables aspect underexplored. MAGNet employs textual descriptions to provide global context for each series and enrich graph nodes without placing a language model in the training loop, thereby increasing efficiency. This design adapts multimodal learning to TSF with exogenous variables rather than directly following existing text-enhanced forecasters.

2.3 GNNs for Time Series Forecasting

Graph-based forecasters represent variables as nodes and learn their interactions through message passing, as shown by MTGNN (Wu et al., 2020), GCGNet (Li et al., 2026), and StemGNN (Cao et al., 2020). These methods use graph learning as a useful basis for TSF (Jin et al., 2024a), yet they are limited to homogeneous or symmetric graphs that do not accommodate distinct input and target roles. Our method builds on graph attention, incorporating a heterogeneous topology, data-driven edge weights, and an ordered two-step propagation tailored for TSF with exogenous variables.

3 Preliminaries

3.1 Problem Formulation

Formally, given an endogenous (target) series $Y \in \mathbb{R}^L$ history and at least one exogenous (covariate) $\mathbf{X} \in \mathbb{R}^{L \times N_{\text{exo}}}$ series where L denotes the sequence history length and N_{exo} is the number of exogenous series, we want to forecast the next few time steps of the endogenous series, $\hat{Y} \in \mathbb{R}^H$ (Lima et al., 2025). Note that H denotes the forecast horizon, which is dataset- and term-specific.

3.2 Neural Granger Causality

Classical Granger Causality (Granger, 1969) states that a variable X "Granger-causes" variable Y if adding X substantially affects Y's distribution.

$$Var(Y|Y_{\text{hist}}) > Var(Y|Y_{\text{hist}}, X_{\text{hist}}), \tag{1}$$

where $Y_{\text{hist}}, X_{\text{hist}}$ denote the history of variable Y and X respectively.

Neural Granger Causality (NGC) (Tank et al., 2022) extends the concept by applying a neural network. Given a base neural network, commonly an LSTM, NGC requires two models: a restricted model (M_1) and a full model (M_2). Model M_1 is restricted to learning Y just from its own history. Meanwhile, model M_2 has full knowledge of both Y and the candidate variable X. The NGC metric quantifies the improvement in the predictive ability of variable Y with the full model compared to the restricted one. As NGC aligns with MAGNet's causation needs, MAGNet employs it as a learned component in the main pipeline to infer a directional predictive causal exo-endo relationship.

4 Methodology

4.1 MAGNet Overview

MAGNet is a text-guided graph module that jointly exploits textual context while distinguishing exogenous and endogenous series, as illustrated in Fig. 2. The framework comprises three major

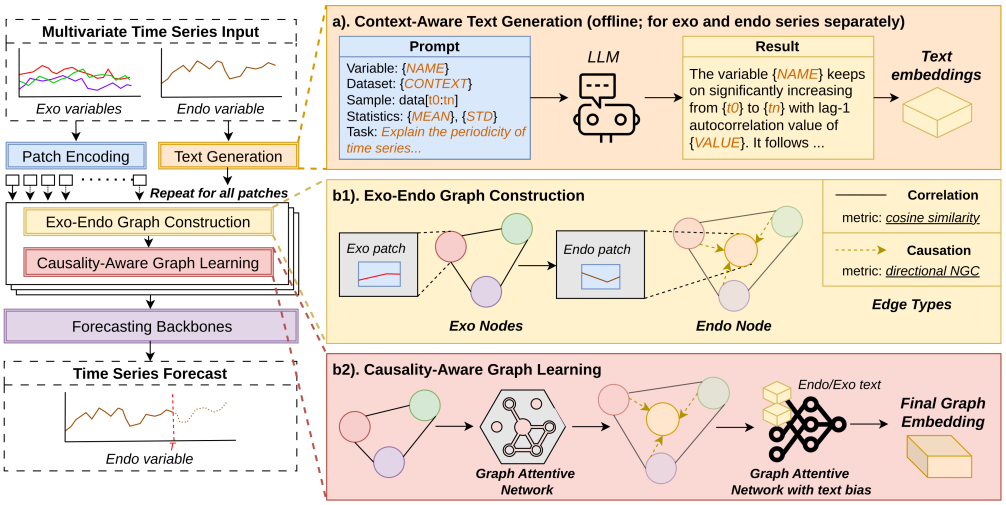


Figure 2: Overview of the MAGNet framework. Input series are patch-encoded alongside dataset-level text embeddings produced offline by an LLM (a). The graph module constructs an asymmetric, heterogeneous graph per patch and applies exo-exo and NGC-based exo-endo graph attention networks to produce causally enriched graph embeddings (b). The backbone then uses these embeddings to guide the final endo-centered prediction.

components: context-aware text generation, exo-endo graph construction, and causality-aware graph learning. First, the text generation module obtains textual embeddings from a pretrained language model by separately describing each series' overall trends and patterns. Next, the graph construction module initializes node representations using patch embeddings and constructs exo-exo and exo-endo edges based on bidirectional correlation and directional Granger-based predictive causal dependencies, respectively. Finally, the causality-aware graph learning module updates node representations through graph attention. It first propagates information over the exo-exo graph before inserting endogenous nodes, enabling them to aggregate information from exogenous-aware neighbors while incorporating text-guided semantic bias. The final forecasting representation is obtained from the learned endogenous node embeddings, creating a target-focused forecasting representation.

4.2 Context-Aware Text Embedding Generation

MAGNet divides the sequences into multiple non-overlapping patches. Afterward, each patch is flattened and projected to match the model's hidden dimension (d).

$$\mathbf{X_{enc}} = \text{Proj}(\text{Patchify}(\mathbf{X})) \in \mathbb{R}^{B\times P\times d}. \tag{2}$$

While patch embeddings capture local temporal patterns, they lack global information. When a prediction process or analysis is applied to a specific patch, each patch works independently with minimal inter-patch interaction, leaving it unaware of the overall context outside that patch. Textual representations provide complementary semantic guidance by encoding long-term behaviors that numerical aggregation alone struggles to capture (Gruver et al., 2023; Zhang et al., 2024). Consequently, MAGNet uses text embeddings as global context to facilitate graph learning for each time series patch.

To avoid additional inference overhead, we generate text embeddings offline and reuse them throughout training and inference. Specifically, the text embeddings, $\mathbf{T}_{\text{endo}}$ and $\mathbf{T}_{\text{exo}}$, are extracted from a pretrained LLM using dataset- and task-specific prompts describing the domain role and historical sample of each series. For the endogenous series, the prompt combines domain descriptions with statistical summaries (e.g., mean and standard deviation) computed from the training set.

Meanwhile, given that causal knowledge enhances graph learning (Job et al., 2025), prompts for exogenous variables describe not only their temporal characteristics but also their predictive influence on the target. MAGNet further generates separate prompts for each term, highlighting different

statistical and trend prominence. Short-term prompts emphasize recent temporal dynamics, including lag-1 autocorrelation (Box et al., 2015a), volatility, and the training set's immediate predictive influence, whereas long-term prompts capture broader trends and seasonal patterns in the training set. Appendix A provides complete prompt templates.

4.3 Exo-Endo Graph Construction and Learning

4.3.1 Graph Construction

MAGNet constructs a graph for each temporal patch, denoted by $G = (V_{\text{exo}}, V_{\text{endo}}, E_{\text{corr}}, E_{\text{cau}})$. Here, V_{exo} and V_{endo} represent the exogenous and endogenous nodes, respectively, while E_{corr} and E_{cau} denote exo-exo correlation edges and exo-endo causality-inspired edges. The graph satisfies three properties: (1) **Heterogeneous**: exogenous and endogenous variables are represented as distinct node types; (2) **Asymmetric**: exo-exo and exo-endo interactions are modeled differently; and (3) **Fully-connected**: every node pair is connected by an edge.

For each node, MAGNet defines initial node embeddings:

$$\mathbf{h}_i^{(p)} = \mathbf{X}_{\mathbf{enc}}{}_i^{(p)}, \tag{3}$$

where $h_i^{(p)}$ denotes the node embedding index i and patch p.

For E_{corr}, MAGNet computes pairwise edge weights using cosine similarity to capture bidirectional correlations:

$$\mathbf{E}_{\mathbf{corr}}{}_{(i,j)}^{(p)} = \frac{\mathbf{h}_i^{(p)} \cdot \mathbf{h}_j^{(p)}}{\left\|\mathbf{h}_i^{(p)}\right\| \cdot \left\|\mathbf{h}_j^{(p)}\right\|}, \tag{4}$$

given that both node i and j are both exogenous nodes.

In contrast, E_{cau} is constructed using Neural Granger Causality (NGC) (Tank et al., 2022), a predictive metric that well represents directional causation. We adopt NGC because it explicitly quantifies how one variable contributes to predicting another (Granger, 1969). Unlike correlation, Granger-based predictive causal dependencies are inherently asymmetric.

MAGNet leverages this asymmetry to construct directed exo-endo edges that represent predictive dependencies. This property benefits graph learning, where edge weights regulate message propagation. By assigning NGC scores as directed edge weights, MAGNet enables endogenous nodes to aggregate exogenous information according to each variable's estimated contribution to the target. We compute the NGC metric as described in Section 3.2.

$$\hat{\mathbf{h}}_{\mathbf{rest}}{}_j^{(p)} = \text{LSTM}(\mathbf{h}_j^{(p)}), \hat{\mathbf{h}}_{\mathbf{full}}{}_{(i,j)}^{(p)} = \text{LSTM}(\mathbf{h}_j^{(p)}; \mathbf{h}_i^{(p-1)}), \tag{5}$$

$$\mathbf{E}_{\mathbf{cau}}{}_{(i,j)}^{(p)} = \text{MSE}(\hat{\mathbf{h}}_{\mathbf{rest}}{}_j^{(p)}) - \text{MSE}(\hat{\mathbf{h}}_{\mathbf{full}}{}_{(i,j)}^{(p)}), \tag{6}$$

where node j denotes the endogenous series and node i represents the respective exogenous series. Appendix C contains the pseudocode on how NGC is computed through graph learning during training.

4.3.2 Causality-Aware Graph Learning

MAGNet adopts a causality-aware graph learning strategy that exploits asymmetric graph structure. It first performs message passing over the exo-exo graph using a multi-head graph attention network (Vaswani et al., 2017), excluding the endogenous node. Each exogenous node attends to all other exogenous nodes, enabling each node to capture inter-exogenous correlations.

$$\mathbf{H}_{\text{exo}}^{(p)} = \text{MultiHead}\big(\mathbf{h}_{\text{exo}}^{(p)} W^Q,\ \mathbf{h}_{\text{exo}}^{(p)} W^K,\ \mathbf{h}_{\text{exo}}^{(p)} W^V; \mathbf{E}_{\text{corr}}^{(p)}\big).$$

Next, the endogenous node aggregates information through a second graph attention layer. The attention weights are modulated by the corresponding NGC scores, while text embeddings serve as semantic biases in the query and key representations, injecting global semantic context and finalizing the graph learning process. Since each graph is constructed independently for each temporal patch, message passing is limited to local observations, and graph embeddings from different patches do

not interact. By incorporating text embeddings that summarize overall trends and seasonal patterns in the full series, MAGNet enables each patch-level graph to capture both local temporal dynamics and global contextual information.

$$\mathbf{h}_{\text{exo}}^{(p)'} = \mathbf{H}_{\text{exo}}^{(p)} + \mathbf{T}_{\text{exo}}, \mathbf{h}_{\text{endo}}^{(p)'} = \mathbf{h}_{\text{endo}}^{(p)} + \mathbf{T}_{\text{endo}}, \tag{7}$$

$$\mathbf{H}_{\text{endo}}^{(p)} = \text{softmax}\left(\frac{\mathbf{Q}'\mathbf{K}'^{\top}}{\sqrt{d_h}} + \log \mathbf{E}_{\text{cau}}\right)\mathbf{H}_{\text{exo}}^{(p)}\mathbf{W}^{V}, \quad \mathbf{Q}' = \mathbf{h}_{\text{endo}}^{(p)'}\mathbf{W}^{Q},\ \mathbf{K}' = \mathbf{h}_{\text{exo}}^{(p)'}\mathbf{W}^{K} \tag{8}$$

4.3.3 Final Graph Embedding

MAGNet concludes by generating the final graph representation from the learned endogenous node embedding:

$$\mathbf{G}^{(p)} = \mathbf{h}_{\text{endo}}^{(p)} + \mathbf{H}_{\text{endo}}^{(p)} \in \mathbb{R}^{B\times 1\times d}, \tag{9}$$

thereby combining all final graph embeddings from all patches yield $\mathbf{G} \in \mathbb{R}^{B\times P\times d}$.

The resulting graph representation integrates global semantic context, conveyed by the text embeddings, with local relational information learned through heterogeneous graph propagation. This representation is then injected into the forecasting backbone, providing an explicit exo-endo relational prior rather than requiring the backbone to infer these interactions implicitly, intuitively suitable for simple backbones. Thus, the backbone can focus on temporal forecasting while MAGNet explicitly models the multimodal relationships identified in the Introduction. Our default MAGNet model adopts a dual-ensemble model with fine-tuned selection of injection methods and backbone enhancers. We selected all choices using only validation data, leaving the test set untouched. Appendix B covers the backbone-specific method in more detail.

4.4 The Training of MAGNet

MAGNet is jointly optimized with the attached backbone end-to-end during training. It mainly uses the MAE (Mean Absolute Error) and MSE (Mean Squared Error) as its main loss functions. However, for datasets with high-variance or heavy-tailed exogenous influences, we use weighted MAE (Wang et al., 2024a).

$$\mathcal{L}_{wMAE} = \frac{1}{H}\sum_{t=1}^{H}(1+|y_t|)\cdot|\hat{y}_t - y_t|\,, \tag{10}$$

where $\hat{y}_t$ denotes the predicted values over the forecasting horizon H, and y_t denotes the ground truth at time t.

Furthermore, MAGNet avoids online LLM inference by generating text embeddings offline, ensuring efficiency. Nevertheless, its complexity increases with longer time series and more variables. As the input length grows, more temporal patches are generated, increasing the graph network's needs. Although longer patches reduce the number of graphs, they may weaken the model's ability to capture local temporal dynamics. Similarly, as the number of variables increases, the fully connected graph becomes denser. A potential solution is to retain only the top-K weighted edges, although this introduces a trade-off between computational efficiency and preserving complete, informative inter-variable relationships. Appendix C contains the pseudocode for the MAGNet architecture, including its training and inference mechanism.

5 Experiments

5.1 Dataset

We evaluate MAGNet on 12 real-world datasets across multiple domains, following the prior state-of-the-art (SOTA) methods (Qiu et al., 2025): five electricity price forecasting (EPF) benchmarks (BE, DE, FR, NP, and PJM) (Lago et al., 2021), one Energy dataset, two Hydropower datasets (Colbun and Rapel), and four Wind Power datasets (Sdwpfh1, Sdwpfh2, Sdwpfm1, and Sdwpfm2).

Table 1: Test forecasting results (MAE/MSE) across 12 datasets, under short-term and long-term settings. **Red** indicates the overall best result per row; Blue indicates the second best.

Dataset	Term	MAGNet (Ours)		Unimodal ExoTSF						Multimodal TSF						Unimodal TSF							
				DAG		GCGNet		TimeXer		T3Time		TimeCMA		TimeLLM		TimeKAN		DLinear		PatchTST		TFT	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
BE	Short	**0.2475**	**0.3738**	0.2820	0.4165	0.2662	0.3905	0.2688	0.4564	0.2705	0.3997	0.6347	0.9475	0.2930	0.4775	0.2913	0.4444	0.2820	0.3993	0.2516	0.4009	0.2623	0.4308
	Long	0.3556	0.5746	0.3888	0.6112	0.3706	0.6070	0.4669	0.7121	0.6414	0.9077	0.7933	1.2214	0.3747	0.6031	0.3729	0.6191	0.3884	0.6174	**0.3339**	**0.5672**	0.3662	0.6154
DE	Short	**0.4013**	**0.4242**	0.4810	0.5685	0.4073	0.4267	0.4239	0.4834	0.4169	0.4351	0.4570	0.5167	0.5036	0.6323	0.4619	0.5173	0.4624	0.5257	0.4288	0.4751	0.4389	0.4899
	Long	**0.5763**	**0.7954**	0.5917	0.9017	0.6272	0.8912	0.5877	0.8542	0.6746	0.9877	0.6154	0.9057	0.5937	0.8639	0.6283	0.9091	0.6089	0.9026	0.5821	0.8594	0.6124	0.9254
FR	Short	0.2033	0.3614	0.2533	0.4398	**0.1965**	**0.3516**	0.2127	0.4067	0.2248	0.4192	0.5340	0.7995	0.2506	0.4625	0.2297	0.4301	0.2291	0.4038	0.1971	0.3973	0.2068	0.4368
	Long	0.3109	0.5688	0.3212	0.5876	0.3006	0.5553	0.3432	0.6354	0.5637	0.8793	0.7058	1.0781	0.3376	0.5766	0.3262	0.6015	0.3189	0.5889	**0.2954**	**0.5554**	0.3154	0.5837
NP	Short	**0.2555**	**0.2372**	0.3053	0.3036	0.2647	0.2439	0.3624	0.3611	0.3382	0.3039	0.6638	0.8052	0.4444	0.4739	0.3132	0.3159	0.3101	0.3010	0.2687	0.2565	0.3127	0.3334
	Long	0.5210	0.6445	0.5294	0.6741	0.5405	0.7062	0.7846	1.0841	0.8677	1.2816	0.9589	1.4902	0.7083	0.9486	0.5373	0.6837	0.5718	0.6995	**0.5161**	**0.6671**	0.5882	0.8351
PJM	Short	**0.1637**	**0.0740**	0.2288	0.1202	0.1685	0.0761	0.1818	0.0888	0.1799	0.0904	0.2796	0.2123	0.2323	0.1155	0.2069	0.1041	0.1864	0.0885	0.1771	0.0837	0.1822	0.0870
	Long	**0.2576**	**0.1836**	0.2866	0.2070	0.2785	0.2019	0.2669	0.1915	0.3549	0.2761	0.3208	0.2235	0.3385	0.2372	0.2863	0.2112	0.2719	0.1928	0.2752	0.2025	0.2814	0.2114
Energy	Short	0.2509	**0.1070**	0.3216	0.1653	0.2664	0.1169	0.2788	0.1274	0.2696	0.1232	0.3437	0.1919	0.2971	0.1439	0.2896	0.1395	**0.2498**	0.1074	0.2580	0.1124	0.3046	0.1557
	Long	0.3562	0.2018	0.4010	0.2526	0.3780	0.2230	**0.3544**	**0.2008**	0.6235	0.5730	0.4143	0.2714	0.3818	0.2311	0.4182	0.2785	0.3600	0.2066	0.4281	0.2840	0.4278	0.3147
Colbun	Short	**0.0962**	**0.0673**	0.1037	0.0774	0.1156	0.0855	0.1755	0.0943	0.3631	0.2895	0.9124	1.2167	0.3001	0.1926	0.1478	0.1113	0.1069	0.0807	0.1164	0.0891	0.1155	0.0872
	Long	**0.2477**	0.2778	0.2512	0.2800	0.3472	0.3930	0.3487	0.4237	0.6971	0.7340	1.0528	1.4616	0.7588	0.8920	0.2777	0.2763	0.2802	0.2996	0.3192	0.3713	0.2972	0.3242
Rapel	Short	**0.2276**	**0.2131**	0.2554	0.2417	0.2440	0.2447	0.3033	0.2538	0.3532	0.2866	0.4476	0.4743	0.3448	0.2708	0.2690	0.2763	0.2447	0.2537	0.2682	0.2852	0.2315	0.2368
	Long	**0.3859**	0.4324	0.4793	0.5357	0.4699	0.5562	0.4409	0.4400	0.6260	0.6914	0.6453	0.7648	0.5712	0.5301	0.5751	0.7166	0.4493	0.5319	0.5308	0.6544	0.4397	**0.5184**
Sdwpfh1	Short	**0.5766**	0.6685	0.6184	0.6793	0.5966	0.6813	0.6059	0.6634	0.6148	0.6539	0.6324	**0.6527**	0.6183	0.6700	0.6043	0.7476	0.5832	0.6833	0.6229	0.7480	0.5829	0.6982
	Long	**0.6879**	0.8291	0.7199	0.8978	0.6698	0.8340	0.7625	0.9152	0.7662	0.8996	0.7445	**0.7953**	0.7406	0.8529	0.6623	0.9042	0.7693	0.9423	0.8724	1.2168	0.6745	0.9239
Sdwpfh2	Short	**0.6221**	**0.7576**	0.6665	0.8142	0.6549	0.8082	0.6616	0.7709	0.6649	0.7597	0.6953	0.7737	0.6697	0.7705	0.6478	0.8303	0.6288	0.7817	0.6828	0.8626	0.6286	0.8285
	Long	0.7720	1.0067	0.7739	0.9859	**0.7450**	0.9706	0.8818	1.1960	0.8271	1.0312	0.7996	**0.9309**	0.7909	0.9900	0.7190	1.0083	0.8372	1.0163	0.9496	1.3999	0.7076	0.9840
Sdwpfm1	Short	**0.4663**	**0.5261**	0.5207	0.5406	0.5037	0.5331	0.5154	0.5329	0.5375	0.5604	0.5516	0.5496	0.5416	0.5465	0.5169	0.6026	0.4968	0.5474	0.5181	0.5989	0.5075	0.5847
	Long	**0.6846**	0.8695	0.7053	0.9025	0.6835	0.8904	0.7005	0.8541	0.7320	0.8595	0.7418	**0.7922**	0.7267	0.8270	0.7508	1.0177	0.7121	0.9112	0.7987	1.0782	0.7094	0.9921
Sdwpfm2	Short	**0.5239**	**0.5998**	0.5469	0.6249	0.5676	0.6359	0.5820	0.6100	0.5790	0.6727	0.5945	0.6154	0.5827	0.6381	0.5509	0.6604	0.5358	0.6206	0.5698	0.6804	0.5433	0.6466
	Long	0.7632	1.0292	0.7805	1.0450	**0.7278**	1.0283	0.7880	1.0199	0.7887	0.9831	0.7919	**0.9143**	0.7509	0.9630	0.7521	1.0693	0.7756	1.0582	0.9139	1.3639	0.8346	1.2994
Avg	Short	**0.3362**	**0.3675**	0.3819	0.4243	0.3543	0.3829	0.3810	0.4041	0.4010	0.4162	0.5622	0.6463	0.4232	0.4495	0.3774	0.4317	0.3597	0.3994	0.3633	0.4159	0.3598	0.4180
	Long	**0.4932**	0.6178	0.5207	0.6817	0.5116	0.6881	0.5605	0.7106	0.6802	0.8420	0.7154	0.9041	0.5895	0.7096	0.5255	0.6913	0.5286	0.6639	0.5680	0.7684	0.5212	0.7106

5.2 Baselines

By default, MAGNet is integrated with DLinear (Su, 2022), a simple unimodal backbone chosen to highlight the complementary contribution of MAGNet's multimodal graph learning. Following the paradigms illustrated in Fig. 1, we compare MAGNet against three categories of baselines: (a) Unimodal TSF, including TimeKAN (Huang et al., 2025b), DLinear (Su, 2022), PatchTST (Nie et al., 2022), and TFT (Lim et al., 2021); (b) Unimodal TSF with Exogenous Variables, including DAG (Qiu et al., 2025), GCGNet (Li et al., 2026), and TimeXer (Wang et al., 2024b); and (c) Multimodal TSF, including T3Time (Chowdhury et al., 2026), TimeCMA (Liu et al., 2025a), and TimeLLM (Jin et al., 2024b). To the best of our knowledge, MAGNet is among the first methods to instantiate the fourth paradigm by explicitly modeling the asymmetric roles of exogenous and endogenous modalities through heterogeneous graph learning.

5.3 Implementation Details

We evaluate all methods under a historical-only protocol, withholding future exogenous values from every baseline, even when they are available. We train all models for up to 200 epochs with early stopping and a patience of 15 epochs. We run all experiments with five seeds, and the result tables report the average metric across these seeds. We also follow the experimental settings of prior SOTA methods for the lookback window and forecasting horizon. Specifically, all datasets except Hydro use $(168, 24)$ for short- and $(720, 360)$ for long-term forecasting, while the Hydro datasets adopt $(60, 10)$ and $(180, 30)$ due to their limited size. In addition, the model uses simple Z-score normalization. Appendix D provides additional details on the datasets, baseline implementations, and experimental settings, including hyperparameter search windows and results.

5.4 Main Results

Table 1 shows that MAGNet achieves the best average MAE of 0.3362 and 0.4932 for short- and long-term forecasting, respectively, demonstrating competitive performance. Improvements are more consistent in the short-term setting, while long-term results remain positive despite greater variability. Notably, MAGNet shows particularly strong gains on several Wind datasets, where meteorological exogenous variables provide informative external signals through clear physical causal relationships with the target. This suggests that modeling exo-endo interactions helps when strong predictive dependencies exist between external variables and the target.

In addition, Table 2 summarizes how MAGNet integrates as a plug-and-play module with other backbones. MAGNet consistently improves simple Unimodal TSF models with no exogenous awareness, with DLinear benefiting the most. This is expected because DLinear, like other Unimodal TSF models such as PatchTST and TFT, lacks mechanisms for exogenous modeling, allowing MAGNet to provide beneficial inductive biases. In contrast, gains appear only in short-term scenar-

Table 2: Summary statistics of MAGNet plug-and-play augmentation across seven backbone forecasters. **Vanilla** denotes the standalone result and **+MAGNet** denotes the MAGNet-augmented result. Win Count: number of datasets (out of 12) where the model outperforms its counterpart (by MAE). **Red** indicates the overall best result (Vanilla vs +MAGNet within each backbone pair).

Metric	Term	Unimodal ExoTSF						Unimodal TSF							
		DAG		GCGNet		TimeXer		TimeKAN		DLinear		PatchTST		TFT	
		Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet
Avg MAE	Short	0.3820	**0.3642**	0.3543	**0.3514**	0.3810	**0.3810**	0.3774	**0.3719**	0.3597	**0.3362**	0.3633	**0.3507**	0.3598	**0.3456**
	Long	**0.5191**	0.5526	**0.5116**	0.5165	**0.5605**	0.5807	**0.5255**	0.5436	0.5286	**0.4932**	0.5680	**0.5671**	**0.5212**	0.5215
Avg MSE	Short	0.4160	**0.4002**	0.3829	**0.3786**	0.4041	**0.3986**	0.4317	**0.4217**	0.3994	**0.3675**	0.4159	**0.3916**	0.4180	**0.3985**
	Long	**0.6568**	0.7274	**0.6548**	0.6591	**0.7106**	0.7357	**0.6913**	0.6965	0.6639	**0.6178**	0.7684	**0.7218**	0.7106	**0.7083**
Win Count	Short	3/12	9/12	3/12	9/12	6/12	6/12	4/12	8/12	0/12	12/12	3/12	8/12	2/12	10/12
	Long	8/12	4/12	8/12	4/12	9/12	3/12	8/12	4/12	0/12	12/12	8/12	4/12	5/12	7/12

ios for unimodal TSFs with exogenous-aware backbones, such as GCGNet, DAG, and TimeXer, because these models already incorporate specialized exogenous-modeling modules, resulting in greater overlap with MAGNet's graph signal, as reflected in their detrimental long-term results. Detailed plug-and-play results across all 12 datasets under short- and long-term forecasting settings are in Appendix E.

6 Interpretability Analysis: Predictive Causal Weight

To understand how MAGNet models variable interactions, we visualize the NGC-based edge weights as heatmaps. These figures provide an interpretable view of which exogenous variables contribute more strongly to the endogenous target. We present representative datasets under the short-term setting here, while additional heatmap visualizations for more datasets and the long-term setting appear in Appendix F.1.

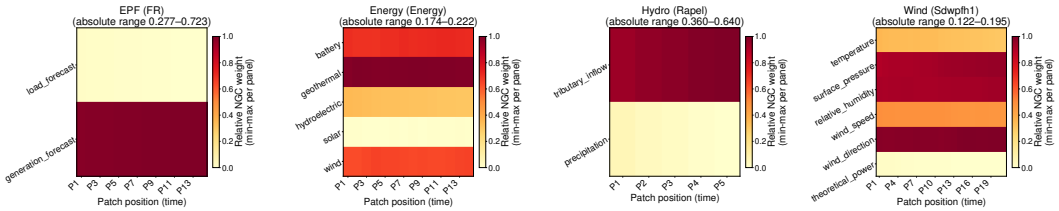


Figure 3: Relative graph causal weights learned by MAGNet+DLinear across FR, Energy, Rapel, and Sdwpfh1 under the short-term forecasting settings.

Fig. 3 summarizes the heatmaps, which reveal how the learned NGC scores separate the importance assigned to each exogenous variable. For example, in the FR dataset, which analyzes the electricity price dataset in France, *generation_forecast* receives substantially higher predictive causal weights than *load_forecast*, indicating a stronger predictive contribution to short-term forecasting.

7 Ablation Studies

7.1 Main Modules Ablation

To assess whether each module benefits MAGNet, we remove each core module from the main MAGNet architecture and evaluate its impact on model performance. Table 3 reports the MAE of several MAGNet variants: Full MAGNet, w/o Text (removing text embeddings), w/o Exo GAT (removing exogenous correlation), w/o NGC (removing predictive causation), and w/o Patching (removing patch-based preprocessing). Across datasets, the full MAGNet achieves the lowest overall MAE, with performance varying by dataset, showing that each component contributes to overall performance. The table also shows the Wilcoxon and t-test p-values, all below the common threshold of 0.05, indicating that the full model differs significantly from its variants across the evaluation settings.

Table 3: Ablation study: Category-level average MAE (short-term + long-term) and statistical significance. **Red** denotes the best result per row. p-values are from paired tests over 24 settings (Full vs. each ablation).

Category	Full	w/o Text	w/o Exo GAT	w/o NGC	w/o Patch
EPF avg	**0.3293**	0.3309	0.3298	0.3304	0.3371
Energy avg	0.3035	0.3024	0.3021	**0.3013**	0.4063
Hydro avg	**0.2393**	0.2428	0.2396	0.2436	0.2545
Wind avg	**0.6371**	0.6431	0.6414	0.6431	0.6506
Overall avg	**0.4148**	0.4179	0.4163	0.4177	0.4336
Wilcoxon p	–	< 0.001	0.006	< 0.001	< 0.001
t-test p	–	0.022	0.021	0.020	0.009

7.2 Case Study Analysis

We visualize MAGNet's prediction results compared to vanilla DLinear on the FR dataset in the short-term setting. This shows that MAGNet consistently improves prediction by producing forecasts closer to the ground truth. Appendix F.2 includes more case studies across datasets and a long-term forecasting setting.

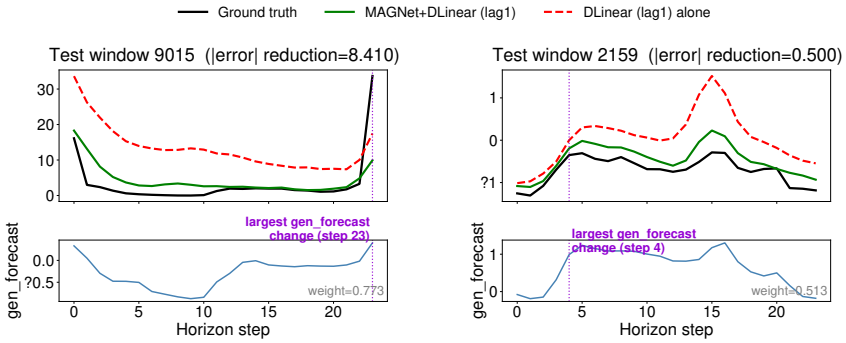


Figure 4: Case studies comparing MAGNet+DLinear against a vanilla DLinear on selected test windows from FR. Each panel shows ground truth (black), MAGNet+DLinear prediction (green), and DLinear-alone prediction (red dashed), with the most influential exogenous variable plotted below.

Fig. 4 shows that compared with the vanilla DLinear model (red), MAGNet more accurately tracks the ground-truth trajectory, particularly during periods of rapid fluctuation where DLinear exhibits larger deviations. MAGNet reduces the absolute error, demonstrating its beneficial contribution. The corresponding *generation_forecast* series is shown below the main performance figure, as it has the highest NGC in Fig. 3. The consistency between its temporal variations and MAGNet's improved predictions suggests that MAGNet effectively leverages highly influential exogenous variables to better capture short-term changes.

8 Conclusion

This paper presents MAGNet, a text-guided heterogeneous graph framework for time-series forecasting with exogenous variables. MAGNet explicitly models the asymmetric roles of exogenous and endogenous variables while incorporating global semantic context through text embeddings. Experiments demonstrate that this formulation achieves competitive performance across both short- and long-term forecasting tasks. However, MAGNet currently assumes a single endogenous forecasting target and relies on offline-generated text embeddings. Future work may extend the framework to multiple endogenous variables, incorporate human-made external knowledge sources, and investigate adaptive text generation to better capture evolving temporal regimes.

AI USE STATEMENT

In this work, we used generative AI tools to check spelling and grammar and to review existing research ideas to avoid plagiarism. We did not use generative AI tools to write the main arguments of this paper or to generate or implement ideas. For writing the paper, we used generative AI to write mathematical equations and algorithms, ensuring consistent variables and notation throughout. We have reviewed all AI-assisted work, verifying mathematical correctness and consistency against the main text. We take responsibility for the final content of this work, including text, claims, or artifacts produced with generative AI.

ETHICS STATEMENT

Our work is based on honesty and transparency, with no intention to cause harm, and aims to add value to the current research field. In addition, we make all datasets used in this work available online.

REPRODUCIBILITY STATEMENT

We submitted supplemental material alongside this manuscript; the link at the end of the abstract section includes the code for our model as a standalone (default), plug-and-play, its ablation studies, and visualization analysis. We have anonymized the repository to follow the double-blind review process. The code includes only a few representative datasets and seeds to keep the file size below the maximum allowed. To fully reproduce the main table result, please import the remaining datasets and re-train the MAGNet default model.

REFERENCES

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015a.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015b.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17766–17778. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/cdf6581cb7aca4b7e19ef136c6e601a5-Paper.pdf.

Abdul Monaf Chowdhury, Rabeya Akter, and Safaeid Hossain Arib. T3time: Tri-modal time series forecasting via adaptive multi-head alignment and residual fusion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(25):20597–20605, Mar. 2026. doi: 10.1609/aaai.v40i25.39196. URL https://ojs.aaai.org/index.php/AAAI/article/view/39196.

Clive W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 19622–19635. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3eb7ca52e8207697361b2c0fb3926511-Paper-Conference.pdf.

Qihe Huang, Zhengyang Zhou, Kuo Yang, and Yang Wang. Exploiting language power for time series forecasting with exogenous variables. In *Proceedings of the ACM Web Conference 2025*, pp. 4043–4052, 2025a. doi: 10.1145/3696410.3714793.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting. *arXiv preprint arXiv:2502.06910*, 2025b.

Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 46(12):10466–10485, 2024a.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23857–23880, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/680b2a8135b9c71278a09cafb605869e-Paper-Conference.pdf.

Simi Job, Xiaohui Tao, Taotao Cai, Haoran Xie, Lin Li, Qing Li, and Jianming Yong. Exploring causal learning through graph neural networks: An in-depth review. *WIREs Data Mining and Knowledge Discovery*, 15(2):e70024, 2025. doi: https://doi.org/10.1002/widm.70024. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.70024. e70024 DMKD-00854.R1.

Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, and Rafał Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2021.116983. URL https://www.sciencedirect.com/science/article/pii/S0306261921004529.

Hao Li, Bowen Deng, Chang Xu, ZhiYuan Feng, Viktor Schlegel, Yu-Hao Huang, Yizheng Sun, Jingyuan Sun, Kailai Yang, Yiyao Yu, and Jiang Bian. Mira: Medical time series foundation model for real-world health data. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 98657–98685. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/8e12ba543adc673da5b89c9311fcf72c-Paper-Conference.pdf.

Zhengyu Li, Xiangfei Qiu, Yuhan Zhu, Xingjian Wu, Jilin Hu, Chenjuan Guo, and Bin Yang. Gcgnet: Graph-consistent generative network for time series forecasting with exogenous variables. *arXiv preprint arXiv:2603.08032*, 2026.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4): 1748–1764, 2021.

Rafael Diniz Toscano de Lima, Sergio Murilo Maciel Fernandes, and Sidney Marlon Lopes de Lima. Time series forecasting with exogenous variables: a literature review to identify promising gaps in computational research. *Revista Principia*, 62, maio 2025. doi: 10.18265/2447-9187a2025id8575. URL https://periodicos.ifpb.edu.br/index.php/principia/article/view/8575.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18):18780–18788, Apr. 2025a. doi: 10.1609/aaai.v39i18.34067. URL https://ojs.aaai.org/index.php/AAAI/article/view/34067.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025b. doi: 10.1609/aaai.v39i18.34095.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 4095–4106, New York, NY, USA, 2024a. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645434. URL https://doi.org/10.1145/3589334.3645434.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 11116–11140, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/2ea18fdc667e0ef2ad82b2b4d65147ad-Paper-Conference.pdf.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Jilin Hu, Chenjuan Guo, and Bin Yang. A comprehensive survey of deep learning for multivariate time series forecasting: a channel strategy perspective (2025). *URL https://arxiv. org/abs/2502*, 10721, 2025.

Xiangfei Qiu, Yuhan Zhu, Zhengyu Li, Xingjian Wu, Bin Yang, and Jilin Hu. DAG: A Dual Correlation Network for Time Series Forecasting with Exogenous Variables. *arXiv e-prints*, art. arXiv:2509.14933, September 2025. doi: 10.48550/arXiv.2509.14933.

Mohammmad Shuvo, Muhtadi Zubair, Afsara Purnota, Sarowar Hossain, and Muhammad Iqbal Hossain. Traffic forecasting using time-series analysis. 01 2021. doi: 10.1109/ICICT50816.2021.9358682.

Chaoqun Su. Dlinear makes efficient long-term predictions. In *Proceedings of ACM Conference (Baidu KDD Cup)*, 2022.

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B. Fox. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2022. doi: 10.1109/TPAMI.2021.3065601.

Thordis Thorarinsdottir, Anders Løland, and Alex Lenkoski. Probabilistic forecasting of temporal trajectories of regional power production-part 1: Wind. *arXiv preprint arXiv:1903.01186*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, and Jianxin Liao. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(12):12694–12702, 2025a. doi: 10.1609/aaai.v39i12.33384. URL https://doi.org/10.1609/aaai.v39i12.33384.

Meng Wang, Jintao Yang, Bin Yang, Hui Li, Tongxin Gong, Bo Yang, and Jiangtao Cui. Towards lightweight time series forecasting: A patch-wise transformer with weak data enriching. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 1278–1291. IEEE, 2025b.

Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. Card: Channel aligned robust blend transformer for time series forecasting. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 11245–11283, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/2f4d6f8e0f4f543db12260696b2a3551-Paper-Conference.pdf.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 469–498. Curran Associates, Inc., 2024b. doi: 10.52202/079017-0015. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0113ef4642264adc2e6924a3cbbdf532-Paper-Conference.pdf.

Lan Wu, Xuebin Wang, Chenglong Ge, Ruijuan Chu, and Linyu Wang. Exotimer: Leveraging large language models for time series forecasting with exogenous variables. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(32):26956–26964, 2026. doi: 10.1609/aaai.v40i32.39908.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 753–763, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-7998-4. doi: 10.1145/3394486.3403118.

Chaolin Yang, Jingdong Yan, and Guangming Wang. Time series trends forecasting for manufacturing enterprises in the digital age. *Journal of Organizational and End User Computing (JOEUC)*, 36(1):1–22, 2024.

Yuqi Yuan, Xiong Luo, Qiaojuan Peng, and Wenbing Zhao. Graft: Infusing pre-trained transformers with relational structure for time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(33):28041–28049, 2026. doi: 10.1609/aaai.v40i33.40029.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*, 2024.

Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jiantao Su, Junfu Lyu, Yanjun Ma, and Dejing Dou. Sdwpf: A dataset for spatial dynamic wind power forecasting challenge at kdd cup 2022. *arXiv preprint arXiv:2208.04360*, 2022.

Zhuohang Zhu, Haodong Chen, Qiang Qu, and Vera Chung. Fincast: A foundation model for financial time-series forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, pp. 4539–4549, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720406. doi: 10.1145/3746252.3761261. URL https://doi.org/10.1145/3746252.3761261.

A CONTEXT-AWARE TEXT GENERATION

We prompt the pretrained language model (LLM) to generate text descriptions. We provide a dataset example from FR (one of the Electricity Price datasets), under short- and long-term forecasting settings, to show a text description generated by the LLM.

A.1 PROMPT TEMPLATES

Prompt templates are provided to send to the LLM to generate text descriptions, which are then encoded via the BERT encoder into context-aware text embeddings. Table 4 and Table 5 present prompts for endogenous and exogenous variables under short- and long-term settings, respectively. More details on the variable description (*var_desc*) and statistical description (*stats_str*) are in the supplemental code in *text_generation.py*.

A.2 TEXT DESCRIPTION RESULT FROM LLM

The LLM output is a text description retrieved before the BERT encoder encodes it. Table 6 and Table 7 display examples of text descriptions produced by the LLM for short- and long-term settings, respectively.

Table 4: Prompt templates for endogenous variables delivered to the LLM to generate text descriptions and embeddings.

Variable Type	**Term**	**Prompt**
Endogenous	Short	You are an expert in *DATASET_CONTEXT*. Variable: *var_name* (target variable to forecast) Description: *var_desc* *stats_str* Based on both the description and the observed statistics, answer concisely (1-2 sentences each): 1. What is the typical short-term temporal pattern? Is it persistent, mean-reverting, or irregular? Does the lag-1 autocorrelation support this? 2. How noisy is this variable within a day or week? Does the coefficient of variation reflect high volatility? 3. At what sub-daily or daily time scale does it change most? 4. What short-term conditions cause sudden spikes or drops? Reply with a paragraph containing ONLY the answers. No preamble.
Endogenous	Long	You are an expert in *DATASET_CONTEXT*. Variable: *var_name* (target variable to forecast) Description: *var_desc* *stats_str* Based on both the description and the observed long-term statistics, answer concisely (1-2 sentences each): 1. What is the long-term trend of this variable? Increasing, decreasing, or flat? Does the trend strength support a meaningful directional movement over months? 2. How strong is the seasonal pattern? Does this variable repeat reliably week-over-week or month-over-month? Do the weekly and monthly autocorrelations support this? 3. Over a horizon of weeks to months, what drives the largest changes in this variable? 4. What long-term conditions or structural changes cause this variable to deviate from its seasonal pattern? Reply with a paragraph containing ONLY the answers. No preamble.

Table 5: Prompt templates for exogenous variables delivered to the LLM to prepare text descriptions and embeddings.

Variable Type	Term	Prompt
Exogenous	Short	You are an expert in *DATASET_CONTEXT*. Variable: *var_name* (exogenous variable) Description: *var_desc* *stats_str* Based on both the description and the observed statistics, answer concisely (1-2 sentences each): 1. How strongly does *var_name* causally influence *endo_name*? Rate the causal strength as strong/moderate/weak and explain the physical or economic mechanism behind this relationship. 2. Does an increase in *var_name* typically increase or decrease *endo_name*? Is this relationship linear or are there nonlinear regimes or threshold effects? 3. Are there specific threshold values of *var_name* beyond which its effect on *endo_name* changes dramatically? 4. Is *var_name* a leading indicator (changes before *endo_name*) or a coincident indicator (changes simultaneously)? What is the typical lag in hours or days? 5. What is the short-term temporal pattern of *var_name* itself? Persistent, volatile, or irregular? Does lag-1 autocorrelation support this? 6. How predictable is *var_name* step-to-step, and at what sub-daily or daily time scale does it change most? Reply with a paragraph containing ONLY the answers. No preamble.
Exogenous	Long	You are an expert in *DATASET_CONTEXT*. Variable: *var_name* (exogenous variable) Description: *var_desc* *stats_str* Based on both the description and the observed long-term statistics, answer concisely (1-2 sentences each): 1. How strongly does *var_name* causally influence *endo_name* over long horizons of weeks to months? Rate as strong/moderate/weak and explain the physical or economic mechanism. 2. Does an increase in *var_name* typically increase or decrease *endo_name* over the long term? Is this relationship linear or are there nonlinear regimes or threshold effects? 3. Are there threshold values of *var_name* beyond which its long-term effect on *endo_name* changes dramatically? 4. Is the long-term causal influence of *var_name* on *endo_name* stronger or weaker than its short-term influence? Under what long-term conditions does it become a strong predictor vs a weak predictor? 5. What is the long-term trend of *var_name* itself? Increasing, decreasing, or stable over months? Does it repeat reliably at weekly or monthly cycles? 6. How does the seasonal pattern of *var_name* align with or drive the seasonal pattern of *endo_name*? Reply with a paragraph containing ONLY the answers. No preamble.

Table 6: Example text descriptions for the FR EPF dataset under short-term forecasting settings.

Variable	Generated Description
electricity_price (endogenous)	The typical short-term temporal pattern is persistent, as indicated by the lag-1 autocorrelation of 0.4909, which suggests that the price tends to follow a step-to-step pattern. This persistence is supported by the lag-1 autocorrelation value. The variable is relatively noisy, with a coefficient of variation of 0.4882, indicating high volatility...
load_forecast (exogenous)	The causal strength of load_forecast on price is strong, as indicated by the high lag-1 autocorrelation (0.9760) and the coefficient of variation (0.2284), suggesting a significant and relatively stable relationship between the two variables. This relationship is likely driven by the physical mechanism of electricity demand and supply...
generation_forecast (exogenous)	The causal strength of generation_forecast on price is strong, as the lag-1 autocorrelation is 0.9860, indicating a high degree of persistence in the relationship. This physical/economic mechanism is likely due to the cross-border supply conditions that influence electricity prices, where changes in French power generation can impact the overall supply...

Descriptions are abbreviated; full prompts are provided in the supplemental code.

Table 7: Example text descriptions for the FR EPF dataset under long-term forecasting settings.

Variable	Generated Description
electricity_price (endogenous)	The long-term trend of this variable is decreasing, with a slope of -0.0004 per time step, indicating a downward movement over time. The trend strength (R^2 of linear fit) of 0.038 supports a meaningful directional movement, albeit with some variability. The seasonal pattern is strong, with a period-to-period variation of 0.515 ...
load_forecast (exogenous)	The long-term causal influence of load_forecast on price is strong, driven by the physical mechanism that higher electricity demand (load_forecast) typically leads to higher prices due to increased competition for limited supply. This relationship is generally linear, with an increase in load_forecast typically leading to an increase in price. The long-term relationship between load_forecast and price is generally linear, with an increase ...
generation_forecast (exogenous)	The causal influence of generation_forecast on price is strong, driven by its role as an exogenous input variable capturing expected domestic supply conditions that influence electricity prices. This mechanism is physical, as changes in generation_forecast directly impact the supply of electricity, which in turn affects prices. The long-term relationship between generation_forecast and price is nonlinear, with an increase in generation_forecast typically leading to a decrease in price ...

Descriptions are abbreviated; full prompts are provided in the supplemental code.

B BACKBONE-SPECIFIC INJECTION DETAILS

B.1 MAGNET+DLINEAR INJECTION DETAILS

For our default MAGNet, with DLinear as the chosen backbone forecaster, we apply both injection methods and backbone enhancers to increase performance and robustness. We also apply ensemble methods: we train and hyperparameter-search two separate models with distinct injection methods and backbone enhancers, then average their predictions at inference time.

Table 8 lists the three injection methods used for MAGNet+DLinear along with their descriptions. More specifically, under the short-term setting, MAGNet+DLinear employs an ensemble of Dual Injection and Cross-attention Injection methods, while under the long-term setting, the model employs an ensemble of Gated Mixing Injection and Cross-attention Injection.

In addition, the DLinear model, as MAGNet's default backbone, is also enhanced by the backbone enhancer. However, we cherry-pick these enhancers for each dataset and term to achieve optimal performance. Table 9 displays the list of possible enhancers applied to both MAGNet and DLinear.

Table 10 and Table 11 show the cherry-picked backbone enhancers applied to the MAGNet+DLinear final model for both short- and long-term settings, respectively.

B.2 MAGNET+OTHERS INJECTION DETAILS

As mentioned, MAGNet can also be seen as a plug-and-play module attached to other existing Unimodal backbones, including DAG, TimeXer, TimeKAN, PatchTST, and TFT. This experiment aims to show how MAGNet attachment improves the backbones and achieves more robust performance than the standalone backbone, not necessarily to achieve SOTA. Since we don't consider other backbones our default, we apply one injection method per backbone (no ensemble and no backbone enhancers) as seen in Table 12.

Table 8: Finalized injection methods used in MAGNet+DLinear.

Method Name	Description
Dual Injection	Collect graph bias from the MAGNet model, then inject it separately into the trend and seasonal components of the DLinear input (post-decomposition), each with its own gate. This lets MAGNet's signal matter differently for slow trends and fast seasonal structure.
Cross-attention Injection	A multi-head attention network attends over MAGNet's full per-patch graph bias to enhance each graph bias, which a simple linear layer then projects into a context vector. This vector is used in a FiLM (Frequency-improved Legendre Memory Model) to enhance the DLinear input sequence.
Gated Mixing Injection	DLinear commonly sums the seasonal and trend outputs with a fixed 1:1 mix. This method allows MAGNet's graph bias to derive a per-sample mixing weight instead (bounded via a tanh/sigmoid function).

Table 9: List of finalized backbone enhancer methods used in MAGNet+DLinear.

Method Name	Description
MLP Enhancer	Instead of the single linear layer used in DLinear to map trend and seasonal patterns to predictions, this method uses a 2-layer MLP (Linear, GELU, Linear). This improves non-linear prediction capacity.
Residual Head	Employs a small zero-initialized residual MLP after trend and seasonal prediction outputs are combined by DLinear. The linear layers used as the main predictor in DLinear remain untouched.
Cross-attention Deep FiLM	This method is specifically used when the Cross-attention Injection method is employed previously. After the attention network attends to all graph biases in a sequence, it is projected to a context vector (pre-FiLM) with a 2-layer MLP similar to the MLP enhancer instead of a linear layer.
NGC Dimension Override ($= k_1$)	Override the hyperparameter-searched value of dimension used in NGC computation to k_1.
Patch Length (PL) Override ($= k_2$)	Override the hyperparameter-searched value of patch length to k_2.
Gated Warmup (GW)	Training follows a gated-based warmup for 30 epochs instead of the usual plain cosine warmup.
Attention Warmup (AW)	Training follows an attention-based warmup for 30 epochs instead of the usual plain cosine warmup.
Weighted Loss and Warmup (WL)	Losses are weighted, and warmup is applied for 5 epochs.

Table 10: List of backbone enhancers used for MAGNet+DLinear under short-term settings specialized for each dataset and each ensemble model.

Dataset	Dual Injection Model	Cross-attention Injection Model
BE, FR, PJM	MLP Enhancer	MLP Enhancer
DE	MLP Enhancer	None
NP	Residual Head	Residual Head
Sdwpfh1	PL Override ($= 8$)	PL Override ($= 8$)
Sdwpfh2	PL Override ($= 8$)	None
Others	None	None

"Others" refers to the datasets that are not mentioned.

Table 11: List of backbone enhancers used for MAGNet+DLinear under long-term settings specialized for each dataset and each ensemble model.

Dataset	Mixing Gate Injection Model	Cross-attention Injection Model
BE	MLP Enhancer + GW	–
DE	–	MLP Enhancer
Energy	PL Override ($= 8$)	None
NP	WL	Residual Head + WL
PJM	None	MLP Enhancer
Rapel	MLP Enhancer + GW	–
Sdwpfh1	–	AW
Sdwpfh2	–	Deep FiLM + NGC Override ($= 64$)
Sdwpfm2	MLP Enhancer	MLP Enhancer
Others	None	None

"Others" refers to the datasets that are not mentioned.

Table 12: List of finalized injection methods used in MAGNet+Other backbones.

Backbone	Injection Method
PatchTST	Project MAGNet's causal graph bias to d_model and add it to the endogenous patch embeddings before the patch encoder.
TimeKAN	Add the graph bias to the finest-resolution patch embeddings before the multi-scale pyramid and mixing process through one learned sigmoid gate.
TFT	Interpolate the graph bias along time to cover both the history and forecast length, then add to the inputs of the LSTM encoder-decoder.
TimeXer	Project the causal graph bias to TimeXer's width and add it to its patch embeddings through a per-patch sigmoid gate multiplied by its global sigmoid scale.
DAG and GCGNet	Squeeze the graph bias into one number token per patch, then stretch it to match the full sequence length before passing it through the model.

C MAGNET ALGORITHM / PSEUDOCODE

Algorithm 1 shows the full pseudocode mechanism of MAGNet's forward pass. It is split into several steps, following the overall framework division. In addition, Algorithm 2 shows the NGC training mechanism implemented for each exo-endo predictive causal relationship.

Algorithm 1 MAGNet Forward Pass (`MAGNETGraphOnly`)

Require: Endogenous patches $X_{\text{endo}} \in \mathbb{R}^{B\times P\times L_{\text{patch}}\times 1}$; exogenous patches $X_{\text{exo}} \in \mathbb{R}^{B\times P\times L_{\text{patch}}\times N_{\text{exo}}}$; static text embeddings $\tau_{\text{endo}}, \tau_{\text{exo}}$ (offline LLM $\rightarrow$ BERT-encoded, pooled `[CLS]`)
Ensure: Causal graph bias $G \in \mathbb{R}^{B\times P\times d}$

// — Step 1: Patch Embedding —
$h^{(p)}_{\text{endo}} \leftarrow \text{Linear}_{\text{endo}}(X^{(p)}_{\text{endo}})$ ▷ $L_{\text{patch}} \rightarrow d$, no positional encoding
$h^{(p)}_{\text{exo}} \leftarrow \text{Linear}_{\text{exo}}(X^{(p)}_{\text{exo}})$

// — Step 2: Text Conditioning (computed once, reused across all p) —
$g \leftarrow \sigma(\theta_{\text{gate}})$ ▷ learned scalar gate, cold-started at $\theta_{\text{gate}} = -2.0$
$T_{\text{endo}} \leftarrow g \cdot \text{LayerNorm}(\text{Proj}(\tau_{\text{endo}}))$
$T_{\text{exo}} \leftarrow g \cdot \text{LayerNorm}(\text{Proj}(\tau_{\text{exo}}))$ ▷ $\in \mathbb{R}^{N_{\text{exo}}\times d}$

for each patch $p = 1, \ldots, P$ **do**
  // — Step 3: Node Construction —
  $h^{(p)}_{\text{endo}} \leftarrow \text{LayerNorm}\big(h^{(p)}_{\text{endo}}\big)$
  $h^{(p)}_{\text{exo}} \leftarrow \text{LayerNorm}\big(h^{(p)}_{\text{exo}}\big)$
  // — Step 4: Exo–Exo Graph (ExoGAT) —
  $s^{(p)}_{ij} \leftarrow \dfrac{h^{(p)}_{\text{exo},i} \cdot h^{(p)}_{\text{exo},j}}{\|h^{(p)}_{\text{exo},i}\|\,\|h^{(p)}_{\text{exo},j}\|}$ ▷ pairwise cosine similarity
  $E^{(p)}_{\text{corr}} \leftarrow \text{softmax}_j(s^{(p)}_{ij})$ ▷ row-normalized edge weights
  $Q, K, V \leftarrow h^{(p)}_{\text{exo}}W^Q,\ h^{(p)}_{\text{exo}}W^K,\ h^{(p)}_{\text{exo}}W^V$
  $\alpha^{(p)} \leftarrow \text{softmax}\Big(\dfrac{QK^\top}{\sqrt{d_{\text{head}}}} + \log E^{(p)}_{\text{corr}}\Big)$ ▷ content logits + structural bias
  $H^{(p)}_{\text{exo}} \leftarrow \text{LayerNorm}\big(h^{(p)}_{\text{exo}} + \alpha^{(p)}V\big)$
  // — Step 5a: Neural Granger Causality —
  for each exogenous variable $i = 1, \ldots, N_{\text{exo}}$ **do**
    $\hat{y}_{\text{rest}} \leftarrow \text{LSTM}(h^{(p)}_{\text{endo}})$
    $\hat{y}_{\text{full},i} \leftarrow \text{LSTM}(h^{(p)}_{\text{endo}} \,\|\, H^{(p-1)}_{\text{exo},i})$
    $c^{(p)}_i \leftarrow \text{MSE}(\hat{y}_{\text{rest}}) - \text{MSE}(\hat{y}_{\text{full},i})$ ▷ positive $\Rightarrow$ Granger-causal
  end for
  $E^{(p)}_{\text{cau}} \leftarrow \text{softmax}_i(c^{(p)}_i)$
  // — Step 5b: Text-Biased, Causality-Weighted Attention Aggregation —
  $Q' \leftarrow (h^{(p)}_{\text{endo}} + T_{\text{endo}})\,W^Q$
  $K' \leftarrow (H^{(p)}_{\text{exo}} + T_{\text{exo}})\,W^K$
  $V' \leftarrow H^{(p)}_{\text{exo}}W^V$ ▷ text does *not* enter V: routing signal, not content
  $\beta^{(p)} \leftarrow \text{softmax}\Big(\dfrac{Q'K'^\top}{\sqrt{d_{\text{head}}}} + \log E^{(p)}_{\text{cau}}\Big)$
  $H^{(p)}_{\text{endo}} \leftarrow \text{LayerNorm}\big(h^{(p)}_{\text{endo}} + \text{Proj}(\beta^{(p)}V')\big)$
  // — Step 6: Patch Output —
  $G^{(p)} \leftarrow H^{(p)}_{\text{endo}} \in \mathbb{R}^{B\times 1\times d}$
end for
$G \leftarrow \text{Stack}\big(G^{(1)}, \ldots, G^{(P)}\big) \in \mathbb{R}^{B\times P\times d}$
return G ▷ handed to backbone-specific injection, cf. Appendix B

Algorithm 2 Neural Granger Causality (NGC) Edge-Weight Computation

Require: Current-patch endogenous embedding $\mathbf{h}_{\text{endo}}^{(p)}$; previous-patch exogenous embeddings $\mathbf{h}_{\text{exo},i}^{(p-1)}$ for $i = 1, \ldots, N_{\text{exo}}$; shared restricted model $\text{LSTM}_{\text{rest}}$; shared unrestricted model LSTM_{unr}; shared output head g

Ensure: Causal edge weights $\mathbf{E}_{\text{cau}}^{(p)} \in \mathbb{R}^{N_{\text{exo}}}$

$\hat{\mathbf{y}}_{\text{rest}} \leftarrow g\big(\text{LSTM}_{\text{rest}}(\mathbf{h}_{\text{endo}}^{(p)})\big)$ ▷ single-step input, no patch history

$\mathcal{L}_{\text{rest}} \leftarrow \mathcal{L}\big(\hat{\mathbf{y}}_{\text{rest}}, \mathbf{h}_{\text{endo}}^{(p)}\big)$

// Fold all N_{exo} candidate variables into the batch dimension of a single call

$\hat{\mathbf{y}}_{\text{unr},1:N_{\text{exo}}} \leftarrow g\big(\text{LSTM}_{\text{unr}}\big(\big[\mathbf{h}_{\text{endo}}^{(p)} \,\|\, \mathbf{h}_{\text{exo},i}^{(p-1)}\big]_{i=1}^{N_{\text{exo}}}\big)\big)$ ▷ one shared LSTM, batched over i

for $i = 1, \ldots, N_{\text{exo}}$ **do**

  $\mathcal{L}_{\text{unr},i} \leftarrow \mathcal{L}\big(\hat{\mathbf{y}}_{\text{unr},i}, \mathbf{h}_{\text{endo}}^{(p)}\big)$

  $s_i^{(p)} \leftarrow \mathcal{L}_{\text{rest}} - \mathcal{L}_{\text{unr},i}$ ▷ positive $\Rightarrow$ variable i is Granger-causal for patch p

end for

$\mathbf{E}_{\text{cau}}^{(p)} \leftarrow \text{softmax}\big(\mathbf{s}^{(p)}\big)$

Training. $\text{LSTM}_{\text{rest}}$, LSTM_{unr}, and g are not pretrained, frozen, or separately warmed up; all NGC parameters share one optimizer with the rest of MAGNet and the backbone and are trained end-to-end from scratch. $\mathcal{L}_{\text{rest}}$ and $\{\mathcal{L}_{\text{unr},i}\}$ are computed only to form $\mathbf{s}^{(p)}$ and are not minimized as auxiliary objectives; the NGC networks receive gradients solely through $\mathbf{E}_{\text{cau}}^{(p)}$'s effect on the attention logits (Eq. 8) and, from there, the forecasting loss.

D Dataset, Baseline, and Experimental Setting Details

D.1 Dataset Details

D.1.1 Dataset Description Table

Table 13 summarizes exogenous and endogenous variable names.

Below is a list of official GitHub repositories that host these datasets, grouped by dataset.

- EPF (BE, DE, FR, NP, PJM) (Lago et al., 2021): https://github.com/jeslago/epftoolbox
- Wind (Sdwpfh1, Sdwpfh2, Sdwpfm1, Sdwpfm2) (Zhou et al., 2022): https://github.com/PaddlePaddle/PaddleSpatial
- Energy and Hydro (Colbun, Rapel) (Qiu et al., 2025): https://github.com/decisionintelligence/DAG

The DAG model first found and applied both the Energy and Hydro datasets, making the DAG GitHub repository the original source we know of. In addition, the original wind datasets do not split the data into 4 smaller datasets by granularity. The DAG model also applies this dataset processing, which we adopt.

D.1.2 Preprocessing Details

MAGNet employs conventional z-score normalization using the StandardScaler command (Pedregosa et al., 2011), allowing a fair comparison with existing baselines, as other models also use a similar mechanism.

$$\hat{x} = \frac{x - \mu}{\sigma}, \tag{11}$$

where μ denotes the mean of the series and σ denotes the standard deviation of the series. We apply this normalization before patch splitting and encoding, so it uses global series statistics rather than per-patch normalization.

One additional detail concerns the distributional shift in the Colbun dataset. Because of its small size, the training and validation distributions differ from the test distribution, resulting in a non-generalizable model and high MSE/MAE. To address this issue, we train the MAGNet model not on the data itself, but on the gradient of a data point and its neighbors by taking discrete first-order differentiation (Box et al., 2015b)

$$x'_t = x_t - x_{t-1} \tag{12}$$

The model is trained on the new series x'_t, and predictions are reconstructed to the original scale via cumulative summation:

$$\hat{x}_t = x_{t_0} + \sum_{i=1}^{t} \hat{x}'_i \tag{13}$$

where x_{t_0} is the last observed value in the lookback window.

D.2 Baseline Implementation Details

As mentioned, we implement multiple backbones both to attach to MAGNet and to compare as standalones. Below is the list of existing time series forecasting models that are used as baselines:

- Unimodal TSF for Exogenous Variables
  - DAG (Qiu et al., 2025): A dual correlation network that uses separate encoding branches to capture both inter-variable correlations and temporal dynamics.

Table 13: Statistics of datasets. Ex. and En. are abbreviations for Exogenous and Endogenous variables, respectively.

Dataset	#Ex.	Ex. Descriptions	En. Description	Sampling Freq.	Length	Split
BE	2	Generation, System Load	Belgium's Electricity Price	1 Hour	52,416	7:1:2
DE	2	Wind Power, Amprion Zonal Load	Germany's Electricity Price	1 Hour	52,416	7:1:2
FR	2	Generation, System Load	France's Electricity Price	1 Hour	52,416	7:1:2
NP	2	Grid Load, Wind Power	Nord Pool Electricity Price	1 Hour	52,416	7:1:2
PJM	2	System Load, SyZonal COMED Load	PJM Electricity Price	1 Hour	52,416	7:1:2
Energy	5	Battery, Geothermal, Hydroelectric, Solar, Wind	Thermoelectric	1 Hour	13,064	7:1:2
Colbun	2	Precipitation, Tributary Inflow	Water Level	1 Day	2,958	7:1:2
Rapel	2	Precipitation, Tributary Inflow	Water Level	1 Day	3,366	7:1:2
Sdwpfh1	6	Climate Features	Active Power	1 Hour	14,641	7:1:2
Sdwpfh2	6	Climate Features	Active Power	1 Hour	14,641	7:1:2
Sdwpfm1	6	Climate Features	Active Power	30 Min	29,281	7:1:2
Sdwpfm2	6	Climate Features	Active Power	30 Min	29,281	7:1:2

  - GCGNet (Li et al., 2026): A variational generative forecaster producing coarse forecasts with a CVAE-based generator, graph-structure aligner, and graph refiner.
  - TimeXer (Wang et al., 2024b): A Transformer-based forecaster that extends the standard patch-based mechanism using cross-attention between endogenous and exogenous representations.

- Unimodal General TSF
  - TimeKAN (Huang et al., 2025b): Applies Kolmogorov-Arnold Networks (KANs) to decompose input series through frequency domains, replacing standard MLP layers.
  - DLinear (Su, 2022): One-layer simple linear model that decomposes input series into trend and seasonal components before applying a separate linear forecaster.
  - PatchTST (Nie et al., 2022): Splits time series into patches and applies patch-independent forecasting via a Transformer-based forecaster.
  - TFT (Lim et al., 2021): Applies LSTM-based sequence encoding to jointly model short- and long-term dependencies.
- Multimodal General TSF
  - T3Time (Chowdhury et al., 2026): A forecasting framework with three parallel branches (time, frequency, and LLM-prompt), fused via a multi-head cross-modal attention.
  - TimeCMA (Liu et al., 2025a): A dual-branch design with a time series encoder and an LLM-prompt encoder that produces robust embeddings from the LLM branch's disentangled representation.
  - TimeLLM (Jin et al., 2024b): Employs a frozen large language model (LLM) by converting numerical time series into text representations.

Below is also a list of official GitHub repositories for each backbone:

- DAG: https://github.com/decisionintelligence/DAG
- GCGNet: https://github.com/decisionintelligence/GCGNet
- TimeXer: https://github.com/thuml/TimeXer
- TimeKAN: https://github.com/huangst21/timekan
- DLinear: https://github.com/vivva/DLinear
- PatchTST: https://github.com/yuqinie98/patchtst
- TFT: https://github.com/google-research/google-research/tree/master/tft
- T3Time: https://github.com/monaf-chowdhury/T3Time
- TimeCMA: https://github.com/ChenxiLiu-HNU/TimeCMA
- TimeLLM: https://github.com/KimMeen/Time-LLM

Table 14: Training configuration for all MAGNet plug-and-play experiments. Settings are consistent across all six backbone architectures unless noted.

Setting	Value	Applies To
Optimizer		
Optimizer	AdamW	All
Weight decay	1×10^{-4}	Short-term
	1×10^{-3}	Long-term
LR schedule	Cosine annealing	All
LR warmup	5-epoch linear warmup	FR, NP, Energy, Rapel, Sdwpfh1
Training		
Max epochs	200	All
Patience	15	All
Gradient		
Gradient clipping	1.0 (max norm)	All except Sdwpfh1
	0.5 (max norm)	Sdwpfh1
Loss Function		
Short-term	MAE ($\mathcal{L}_{\text{MAE}}$)	EPF, Energy, Hydro
Long-term	Smooth L1 ($\beta = 0.5$)	EPF, Energy
Long-term	MAE ($\mathcal{L}_{\text{MAE}}$)	Hydro
All terms	Weighted MAE ($\mathcal{L}_{\text{wMAE}}$)	Wind
Preprocessing		
Normalization	Z-score normalization	All
Differencing	First-order	Colbun (long-term only)
Patch length	12	All
Hardware		
GPU	NVIDIA RTX 4090 (24 GB)	—
No. of GPUs	4	—
Framework	PyTorch 2.12.1 (CUDA 13.0)	—

D.3 Experimental Settings

D.3.1 Training Configuration

Table 14 shows the training configuration parameters and types used for our MAGNet training, including the PyTorch version and GPU hardware used in all our experiments and ablation studies. Some configurations are dataset-specific and manually searched to achieve the best performance for each dataset, as shown in the table. MAGNet and all other backbones are run under 5 distinct pre-selected seeds (seed = 0, 42, 123, 2024, 3407) under the same experimental protocol. The current reported results are the averages of these 5 runs.

In addition, to better match real-world scenarios and practical industry use, MAGNet does not use future values of exogenous variables to predict the endogenous series, even when such values are available in some datasets (Wang et al., 2025b). This ensures that the baselines mentioned above are evaluated under the same circumstances and do not receive future exogenous values, ensuring a fair comparison.

D.3.2 Hyperparameter Search Space

As mentioned, attaching MAGNet as a plug-and-play to other backbones (including DAG) requires a joint end-to-end hyperparameter search to acquire its full potential. Table 15 shows which hyper-parameters to search and their search ranges or possible values per backbone.

D.3.3 Best Hyperparameters per Backbone per Dataset

Tables 16, 17, 18, 19, 24, 25, 20, 21, 22, 23, 26, 27, 28, 29 show the hyperparameters that are ultimately chosen as the optimized values, yielding the best performance based on validation MAE. These values serve as the default for subsequent experiments and studies in this paper.

Table 15: Hyperparameter search space per backbone. Total configs per dataset–term shown in parentheses.

Backbone	Hyperparameter	Description	Options
PatchTST (18)	`ptst_d_model`	PatchTST width	64 / 128 / 256
	`ptst_e_layers`	Encoder depth	2 / 3 / 4
	`lr`	Learning rate	1×10^{-4} / 5×10^{-4}
TFT (6)	`tft_d_model`	TFT hidden width	32 / 64 / 128
	`lr`	Learning rate	1×10^{-4} / 5×10^{-4}
DLinear (18)	`moving_avg`	Decomp. kernel size	13 / 25 / 51
	`magnet_d_model`	MAGNet graph width	32 / 64 / 128
	`lr`	Learning rate	1×10^{-4} / 5×10^{-4}
TimeXer (24)	`tx_d_model`	TimeXer width	256 / 512
	`tx_e_layers`	Encoder depth	2 / 3
	`magnet_d_model`	MAGNet graph width	32 / 64 / 128
	`lr`	Learning rate	1×10^{-4} / 5×10^{-4}
DAG (24)	`dag_d_model`	DAG encoder width	128 / 256
	`dag_e_layers`	Encoder depth	1 / 2
	`magnet_d_model`	MAGNet graph width	32 / 64 / 128
	`lr`	Learning rate	1×10^{-4} / 5×10^{-4}
TimeKAN (12)	`tk_d_model`	TimeKAN width	32 / 64 / 128
	`tk_e_layers`	Encoder depth	1 / 2
	`lr`	Learning rate	1×10^{-4} / 5×10^{-4}
GCGNet (24)	`gcg_d_model`	GCGNet width	32 / 64 / 128
	`magnet_d_model`	MAGNet graph width	64 / 128 / 256 / 512
	`lr`	Learning rate	1×10^{-4} / 1×10^{-3}

Table 16: MAGNet+DLinear fusion hyperparameters — short-term setting. Each dataset lists the two ensemble models.

Dataset	Variant	moving_avg	magnet_d_model	lr
BE	dual_only	13	32	5×10^{-4}
	crossattn_mq	13	32	1×10^{-4}
DE	dual_only	25	32	5×10^{-4}
	crossattn_mq	51	32	5×10^{-4}
FR	dual_only	25	128	5×10^{-4}
	crossattn_mq	51	32	1×10^{-4}
NP	dual_only	13	32	5×10^{-4}
	crossattn_mq	13	32	5×10^{-4}
PJM	dual_only	13	64	1×10^{-4}
	crossattn_mq	13	32	1×10^{-4}
Energy	dual_only	25	32	5×10^{-4}
	crossattn_mq	13	64	5×10^{-4}
Colbun	dual_only	51	32	5×10^{-4}
	crossattn_mq	25	32	5×10^{-4}
Rapel	crossattn_mq	51	32	5×10^{-4}
Sdwpfh1	dual_only	51	64	5×10^{-4}
	crossattn_mq	51	64	5×10^{-4}
Sdwpfh2	dual_only	25	128	5×10^{-4}
	crossattn_mq	13	32	5×10^{-4}
Sdwpfm1	dual_only	25	32	1×10^{-4}
	crossattn_mq	51	32	1×10^{-4}
Sdwpfm2	dual_only	13	32	1×10^{-4}
	crossattn_mq	25	128	5×10^{-4}

Table 17: MAGNet+DLinear fusion hyperparameters — long-term setting. Each dataset lists the two explored fusion variants.

Dataset	Variant	moving_avg	magnet_d_model	lr
BE	gated_mixing_only	25	64	1×10^{-4}
DE	crossattn_mq	13	32	1×10^{-4}
FR	gated_mixing_only	51	64	1×10^{-4}
	crossattn_mq	25	64	1×10^{-4}
NP	gated_mixing_only	51	32	5×10^{-4}
	crossattn_mq	25	64	1×10^{-4}
PJM	gated_mixing_only	51	32	1×10^{-4}
	crossattn_mq	13	32	1×10^{-4}
Energy	gated_mixing_only	25	32	1×10^{-4}
Colbun	gated_mixing_only	25	32	1×10^{-4}
	crossattn_mq	51	64	1×10^{-4}
Rapel	gated_mixing_only	13	128	5×10^{-4}
Sdwpfh1	crossattn_mq	25	64	1×10^{-4}
Sdwpfh2	crossattn_mq	13	32	1×10^{-4}
Sdwpfm1	gated_mixing_only	25	128	5×10^{-4}
	crossattn_mq	13	32	1×10^{-4}
Sdwpfm2	gated_mixing_only	13	128	5×10^{-4}
	crossattn_mq	51	32	1×10^{-4}

Table 18: MAGNet+PatchTST fusion hyperparameters — short-term setting.

Dataset	ptst_d_model	ptst_e_layers	lr
BE	64	3	1×10^{-4}
DE	64	2	5×10^{-4}
FR	64	2	1×10^{-4}
NP	64	2	5×10^{-4}
PJM	256	2	1×10^{-4}
Energy	256	3	1×10^{-4}
Colbun	256	2	5×10^{-4}
Rapel	256	4	5×10^{-4}
Sdwpfh1	64	4	1×10^{-4}
Sdwpfh2	64	3	1×10^{-4}
Sdwpfm1	128	4	1×10^{-4}
Sdwpfm2	64	4	5×10^{-4}

Table 19: MAGNet+PatchTST fusion hyperparameters — long-term setting.

Dataset	ptst_d_model	ptst_e_layers	lr
BE	256	4	1×10^{-4}
DE	64	2	1×10^{-4}
FR	256	4	5×10^{-4}
NP	128	2	1×10^{-4}
PJM	256	2	5×10^{-4}
Energy	64	2	1×10^{-4}
Colbun	256	4	5×10^{-4}
Rapel	256	4	5×10^{-4}
Sdwpfh1	256	2	5×10^{-4}
Sdwpfh2	64	4	1×10^{-4}
Sdwpfm1	128	2	5×10^{-4}
Sdwpfm2	128	2	5×10^{-4}

Table 20: MAGNet+TimeKAN fusion hyperparameters — short-term setting.

Dataset	tk_d_model	tk_e_layers	lr
BE	32	1	5×10^{-4}
DE	128	1	5×10^{-4}
FR	64	1	5×10^{-4}
NP	128	1	5×10^{-4}
PJM	64	1	5×10^{-4}
Energy	64	1	5×10^{-4}
Colbun	64	1	5×10^{-4}
Rapel	64	1	5×10^{-4}
Sdwpfh1	32	1	5×10^{-4}
Sdwpfh2	64	1	5×10^{-4}
Sdwpfm1	32	1	5×10^{-4}
Sdwpfm2	128	1	5×10^{-4}

Table 21: MAGNet+TimeKAN fusion hyperparameters — long-term setting.

Dataset	tk_d_model	tk_e_layers	lr
BE	32	2	5×10^{-4}
DE	64	1	1×10^{-4}
FR	64	1	5×10^{-4}
NP	128	1	1×10^{-4}
PJM	32	1	5×10^{-4}
Energy	64	2	1×10^{-4}
Colbun	32	1	1×10^{-4}
Rapel	32	1	1×10^{-4}
Sdwpfh1	64	2	5×10^{-4}
Sdwpfh2	128	2	1×10^{-4}
Sdwpfm1	32	2	1×10^{-4}
Sdwpfm2	64	1	1×10^{-4}

Table 22: MAGNet+TimeXer fusion hyperparameters — short-term setting.

Dataset	tx_d_model	tx_e_layers	magnet_d_model	lr
BE	256	2	64	1×10^{-4}
DE	256	2	128	1×10^{-4}
FR	256	2	32	1×10^{-4}
NP	512	2	128	5×10^{-4}
PJM	256	2	128	1×10^{-4}
Energy	512	2	32	1×10^{-4}
Colbun	512	2	32	1×10^{-4}
Rapel	512	2	64	5×10^{-4}
Sdwpfh1	256	2	128	1×10^{-4}
Sdwpfh2	256	2	32	1×10^{-4}
Sdwpfm1	256	2	32	5×10^{-4}
Sdwpfm2	256	2	32	1×10^{-4}

Table 23: MAGNet+TimeXer fusion hyperparameters — long-term setting.

Dataset	tx_d_model	tx_e_layers	magnet_d_model	lr
BE	256	3	32	1×10^{-4}
DE	256	2	32	1×10^{-4}
FR	256	2	32	1×10^{-4}
NP	256	2	64	1×10^{-4}
PJM	256	2	32	1×10^{-4}
Energy	256	2	32	1×10^{-4}
Colbun	512	3	32	5×10^{-4}
Rapel	256	3	128	1×10^{-4}
Sdwpfh1	512	3	128	5×10^{-4}
Sdwpfh2	256	3	128	1×10^{-4}
Sdwpfm1	256	2	64	5×10^{-4}
Sdwpfm2	256	3	64	5×10^{-4}

Table 24: MAGNet+TFT fusion hyperparameters — short-term setting.

Dataset	`tft_d_model`	`lr`
BE	32	5×10^{-4}
DE	32	5×10^{-4}
FR	32	5×10^{-4}
NP	32	5×10^{-4}
PJM	32	5×10^{-4}
Energy	64	5×10^{-4}
Colbun	64	5×10^{-4}
Rapel	32	5×10^{-4}
Sdwpfh1	64	1×10^{-4}
Sdwpfh2	32	1×10^{-4}
Sdwpfm1	128	1×10^{-4}
Sdwpfm2	64	5×10^{-4}

Table 25: MAGNet+TFT fusion hyperparameters — long-term setting.

Dataset	`tft_d_model`	`lr`
BE	64	5×10^{-4}
DE	64	5×10^{-4}
FR	128	5×10^{-4}
NP	64	5×10^{-4}
PJM	32	1×10^{-4}
Energy	128	5×10^{-4}
Colbun	128	5×10^{-4}
Rapel	32	5×10^{-4}
Sdwpfh1	32	5×10^{-4}
Sdwpfh2	32	1×10^{-4}
Sdwpfm1	32	5×10^{-4}
Sdwpfm2	32	1×10^{-4}

Table 26: MAGNet+DAG fusion hyperparameters — short-term setting.

Dataset	`dag_d_model`	`dag_e_layers`	`magnet_d_model`	`lr`
BE	128	2	32	1×10^{-4}
DE	128	1	32	5×10^{-4}
FR	128	2	64	1×10^{-4}
NP	128	1	64	5×10^{-4}
PJM	128	2	32	1×10^{-4}
Energy	128	2	32	5×10^{-4}
Colbun	256	1	64	1×10^{-4}
Rapel	256	1	32	5×10^{-4}
Sdwpfh1	128	2	32	5×10^{-4}
Sdwpfh2	256	1	64	1×10^{-4}
Sdwpfm1	128	2	32	1×10^{-4}
Sdwpfm2	128	1	128	1×10^{-4}

Table 27: MAGNet+DAG fusion hyperparameters — long-term setting.

Dataset	dag_d_model	dag_e_layers	magnet_d_model	lr
BE	128	1	128	1×10^{-4}
DE	256	1	32	1×10^{-4}
FR	256	1	64	1×10^{-4}
NP	128	1	32	1×10^{-4}
PJM	256	1	64	1×10^{-4}
Energy	128	1	64	5×10^{-4}
Colbun	256	2	64	5×10^{-4}
Rapel	128	2	32	5×10^{-4}
Sdwpfh1	256	2	64	5×10^{-4}
Sdwpfh2	256	2	64	5×10^{-4}
Sdwpfm1	256	2	64	5×10^{-4}
Sdwpfm2	128	2	128	1×10^{-4}

Table 28: MAGNet+GCGNet fusion hyperparameters — short-term setting.

Dataset	gcg_d_model	magnet_d_model	lr
BE	128	64	1×10^{-4}
DE	64	32	1×10^{-3}
FR	64	32	1×10^{-3}
NP	128	32	1×10^{-3}
PJM	256	32	1×10^{-3}
Energy	64	64	1×10^{-3}
Colbun	512	32	1×10^{-3}
Rapel	64	32	1×10^{-3}
Sdwpfh1	64	128	1×10^{-3}
Sdwpfh2	64	64	1×10^{-3}
Sdwpfm1	64	64	1×10^{-4}
Sdwpfm2	512	32	1×10^{-4}

Table 29: MAGNet+GCGNet fusion hyperparameters — long-term setting.

Dataset	gcg_d_model	magnet_d_model	lr
BE	128	64	1×10^{-4}
DE	64	64	1×10^{-3}
FR	64	128	1×10^{-3}
NP	128	32	1×10^{-3}
PJM	256	128	1×10^{-3}
Energy	64	32	1×10^{-3}
Colbun	512	64	1×10^{-3}
Rapel	64	64	1×10^{-3}
Sdwpfh1	64	32	1×10^{-3}
Sdwpfh2	64	64	1×10^{-3}
Sdwpfm1	64	128	1×10^{-4}
Sdwpfm2	512	128	1×10^{-4}

E DETAILED PLUG-AND-PLAY RESULTS

As mentioned, by default, MAGNet is attached to DLinear with its dataset-specific injection methods explained in Appendix B. In addition, MAGNet can also be used as a plug-and-play model with other Unimodal TSF backbones, both general and specialized for exogenous variables. Table 30 and Table 31 compare MAE and MSE, respectively, between each vanilla backbone and the MAGNet-attached backbone.

Table 30: Forecasting results (MAE) across 12 datasets under short-term and long-term settings. **Bold** indicates the better result within each Vanilla/+MAGNet pair.

Dataset	Term	Unimodal ExoTSF						Unimodal TSF							
		DAG		GCGNet		TimeXer		TimeKAN		DLinear		PatchTST		TFT	
		Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet
BE	Short	0.2820	**0.2705**	0.2662	**0.2607**	**0.2688**	0.2889	0.2913	**0.2614**	0.2820	**0.2475**	0.2516	**0.2516**	0.2623	**0.2604**
	Long	**0.3888**	0.3955	**0.3706**	0.3873	**0.4669**	0.5073	0.3729	**0.3702**	0.3884	**0.3556**	**0.3339**	0.3896	0.3662	**0.3509**
DE	Short	0.4810	**0.4186**	**0.4073**	0.4076	0.4239	**0.4116**	0.4619	**0.4300**	0.4624	**0.4013**	0.4288	**0.4116**	0.4389	**0.4222**
	Long	0.5917	**0.5877**	0.6272	**0.6107**	**0.5877**	0.6054	**0.6283**	0.6710	0.6089	**0.5763**	**0.5821**	0.5850	0.6124	**0.6069**
FR	Short	0.2533	**0.2395**	0.1965	**0.1963**	**0.2127**	0.2205	0.2297	**0.2148**	0.2291	**0.2033**	**0.1971**	0.1995	**0.2068**	0.2094
	Long	**0.3212**	0.4065	**0.3006**	0.3164	**0.3432**	0.3677	**0.3262**	0.3413	0.3189	**0.3109**	**0.2954**	0.3136	**0.3154**	0.3352
NP	Short	0.3053	**0.2694**	0.2647	**0.2598**	0.3624	**0.3496**	0.3132	**0.2890**	0.3101	**0.2555**	**0.2687**	0.2720	0.3127	**0.2663**
	Long	0.5294	**0.5202**	**0.5405**	0.5765	0.7846	**0.6841**	**0.5373**	0.5506	0.5718	**0.5210**	**0.5161**	0.5364	0.5882	**0.5588**
PJM	Short	0.2288	**0.1672**	0.1685	**0.1660**	0.1818	**0.1749**	0.2069	**0.1812**	0.1864	**0.1637**	0.1771	**0.1709**	0.1822	**0.1756**
	Long	0.2866	**0.2823**	**0.2785**	0.2804	0.2669	**0.2665**	0.2863	**0.2813**	0.2719	**0.2576**	**0.2752**	0.2759	**0.2814**	0.2908
Energy	Short	0.3216	**0.2787**	**0.2664**	0.2698	**0.2788**	0.2920	**0.2896**	0.2962	**0.2498**	0.2509	**0.2580**	0.2775	0.3046	**0.2713**
	Long	**0.4010**	0.4415	**0.3780**	0.3972	**0.3544**	0.3597	**0.4182**	0.4333	0.3600	**0.3562**	**0.4281**	0.5644	**0.4278**	0.5304
Colbun	Short	**0.1037**	0.1136	**0.1156**	0.1158	**0.1755**	0.1902	**0.1478**	0.1727	0.1069	**0.0962**	0.1164	**0.1116**	0.1155	**0.1042**
	Long	**0.2512**	0.2958	**0.3472**	0.3718	0.3487	**0.2719**	**0.2777**	0.3695	0.2802	**0.2477**	**0.3192**	0.3206	0.2972	**0.2634**
Rapel	Short	0.2554	**0.2521**	0.2440	**0.2394**	0.3033	**0.2774**	**0.2690**	0.3307	0.2447	**0.2276**	0.2682	**0.2517**	**0.2315**	0.2398
	Long	0.4793	**0.4562**	0.4699	**0.4509**	**0.4409**	0.5355	0.5751	**0.5737**	0.4493	**0.3859**	0.5308	**0.4899**	**0.4397**	0.4439
Sdwpfh1	Short	0.6184	**0.6032**	0.5966	**0.5916**	0.6059	**0.6030**	0.6043	**0.5947**	0.5832	**0.5766**	0.6229	**0.5781**	0.5829	**0.5568**
	Long	**0.7199**	0.7616	**0.6698**	0.6730	**0.7625**	0.7834	**0.6623**	0.6931	0.7693	**0.6879**	0.8724	**0.7666**	0.6745	**0.6659**
Sdwpfh2	Short	**0.6665**	0.6711	0.6549	**0.6484**	**0.6616**	0.6680	0.6478	**0.6373**	0.6288	**0.6221**	0.6828	**0.6417**	0.6286	**0.6125**
	Long	**0.7739**	0.8292	0.7450	**0.7133**	**0.8818**	0.9525	**0.7190**	0.7427	0.8372	**0.7720**	0.9496	**0.9077**	**0.7076**	0.7325
Sdwpfm1	Short	0.5207	**0.5202**	0.5037	**0.4983**	**0.5154**	0.5211	0.5169	**0.4990**	0.4968	**0.4663**	0.5181	**0.4953**	0.5075	**0.4891**
	Long	**0.7053**	0.8502	**0.6835**	0.6944	**0.7005**	0.7953	**0.7508**	0.7716	0.7121	**0.6846**	**0.7987**	0.8019	0.7094	**0.7004**
Sdwpfm2	Short	**0.5469**	0.5659	0.5676	**0.5626**	0.5820	**0.5754**	**0.5509**	0.5552	0.5358	**0.5239**	0.5698	**0.5474**	0.5433	**0.5395**
	Long	**0.7805**	0.8045	0.7278	**0.7263**	**0.7880**	0.8391	0.7521	**0.7243**	0.7756	**0.7632**	0.9139	**0.8541**	0.8346	**0.7792**
Avg	Short	0.3820	**0.3642**	0.3543	**0.3514**	0.3810	**0.3810**	0.3774	**0.3719**	0.3597	**0.3362**	0.3633	**0.3507**	0.3598	**0.3456**
	Long	**0.5191**	0.5526	**0.5116**	0.5165	**0.5605**	0.5807	**0.5255**	0.5436	0.5286	**0.4932**	0.5680	**0.5671**	**0.5212**	0.5215

Table 31: Forecasting results (MSE) across 12 datasets under short-term and long-term settings. **Bold** indicates the better result within each Vanilla/+MAGNet pair.

Dataset	Term	Unimodal ExoTSF						Unimodal TSF							
		DAG		GCGNet		TimeXer		TimeKAN		DLinear		PatchTST		TFT	
		Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet	Vanilla	+MAGNet
BE	Short	0.4165	**0.4059**	0.3905	**0.3849**	0.4564	**0.4498**	0.4444	**0.4082**	0.3993	**0.3738**	**0.4009**	0.4125	**0.4308**	0.4315
	Long	**0.6112**	0.6354	**0.6070**	0.6146	**0.7121**	0.7524	0.6191	**0.5919**	0.6174	**0.5746**	**0.5672**	0.6291	0.6155	**0.5882**
DE	Short	0.5685	**0.4417**	0.4267	**0.4203**	0.4834	**0.4614**	0.5173	**0.4577**	0.5257	**0.4242**	0.4751	**0.4405**	0.4899	**0.4681**
	Long	0.9017	**0.8339**	0.8912	**0.8778**	**0.8542**	0.8909	**0.9091**	0.9984	0.9026	**0.7954**	0.8594	**0.8370**	**0.9254**	0.9403
FR	Short	**0.4398**	0.4511	**0.3516**	0.3575	**0.4067**	0.4119	0.4301	**0.4049**	0.4038	**0.3614**	**0.3973**	0.4013	0.4368	**0.4246**
	Long	**0.5876**	0.7477	**0.5553**	0.5744	**0.6354**	0.6511	**0.6015**	0.6380	0.5889	**0.5688**	**0.5554**	0.5833	**0.5837**	0.6132
NP	Short	0.3036	**0.2393**	0.2439	**0.2355**	0.3611	**0.3149**	0.3159	**0.2748**	0.3010	**0.2372**	0.2565	**0.2526**	0.3334	**0.2475**
	Long	0.6741	**0.6511**	**0.7062**	0.7635	1.0841	**0.8919**	**0.6837**	0.7256	0.6995	**0.6445**	**0.6672**	0.6871	0.8351	**0.7480**
PJM	Short	0.1202	**0.0775**	0.0761	**0.0737**	**0.0888**	0.0967	0.1041	**0.0857**	0.0885	**0.0740**	0.0837	**0.0798**	**0.0870**	0.0879
	Long	0.2070	**0.1985**	0.2019	**0.2012**	**0.1915**	0.1926	0.2112	**0.2028**	0.1928	**0.1836**	0.2025	**0.1984**	**0.2114**	0.2194
Energy	Short	0.1653	**0.1303**	**0.1169**	0.1192	**0.1274**	0.1389	**0.1395**	0.1467	0.1074	**0.1070**	**0.1124**	0.1304	0.1557	**0.1255**
	Long	**0.2526**	0.3073	**0.2230**	0.2514	**0.2008**	0.2067	**0.2785**	0.2975	0.2066	**0.2018**	**0.2840**	0.4833	**0.3147**	0.4353
Colbun	Short	**0.0774**	0.0826	0.0855	**0.0827**	**0.0943**	0.1005	**0.1113**	0.1533	0.0807	**0.0673**	0.0891	**0.0812**	0.0872	**0.0811**
	Long	**0.2800**	0.3402	**0.3930**	0.4189	0.4237	**0.2879**	**0.2763**	0.3482	0.2996	**0.2778**	0.3713	**0.3288**	0.3242	**0.3090**
Rapel	Short	0.2417	**0.2398**	0.2447	**0.2428**	0.2538	**0.2290**	**0.2763**	0.3751	0.2537	**0.2131**	0.2852	**0.2390**	0.2368	**0.2264**
	Long	0.5357	**0.4887**	0.5562	**0.4868**	**0.4400**	0.5323	0.7166	**0.6534**	0.5319	**0.4324**	0.6544	**0.4883**	0.5184	**0.4719**
Sdwpfh1	Short	**0.6793**	0.7105	0.6813	**0.6714**	0.6634	**0.6527**	0.7476	**0.7182**	0.6833	**0.6685**	0.7480	**0.6854**	0.6982	**0.6916**
	Long	**0.8978**	1.0168	0.8340	**0.8310**	**0.9152**	0.9238	**0.9042**	0.9211	0.9423	**0.8291**	1.2168	**0.9106**	**0.9239**	0.9723
Sdwpfh2	Short	0.8142	**0.8054**	0.8082	**0.7935**	**0.7709**	0.7737	0.8303	**0.8122**	0.7817	**0.7576**	0.8626	**0.7798**	0.8285	**0.7855**
	Long	**0.9859**	1.1718	**0.9706**	1.0011	**1.1960**	1.4784	1.0083	**0.9331**	1.0163	**1.0067**	1.3999	**1.3482**	**0.9840**	1.0312
Sdwpfm1	Short	**0.5406**	0.5637	0.5331	**0.5265**	**0.5329**	0.5489	0.6026	**0.5766**	0.5474	**0.5261**	0.5989	**0.5678**	0.5847	**0.5602**
	Long	**0.9025**	1.1972	0.8904	**0.8694**	**0.8541**	0.9538	**1.0177**	1.0374	0.9112	**0.8695**	1.0782	**1.0123**	**0.9921**	1.0257
Sdwpfm2	Short	**0.6249**	0.6550	0.6359	**0.6353**	0.6100	**0.6043**	0.6604	**0.6471**	0.6206	**0.5998**	0.6804	**0.6292**	**0.6466**	0.6517
	Long	**1.0450**	1.1401	1.0283	**1.0194**	**1.0199**	1.0666	1.0693	**1.0102**	1.0582	**1.0292**	1.3639	**1.1549**	1.2994	**1.1453**
Avg	Short	0.4160	**0.4002**	0.3829	**0.3786**	0.4041	**0.3986**	0.4317	**0.4217**	0.3994	**0.3675**	0.4159	**0.3916**	0.4180	**0.3985**
	Long	**0.6568**	0.7274	**0.6548**	0.6591	**0.7106**	0.7357	**0.6913**	0.6965	0.6639	**0.6178**	0.7684	**0.7218**	0.7106	**0.7083**

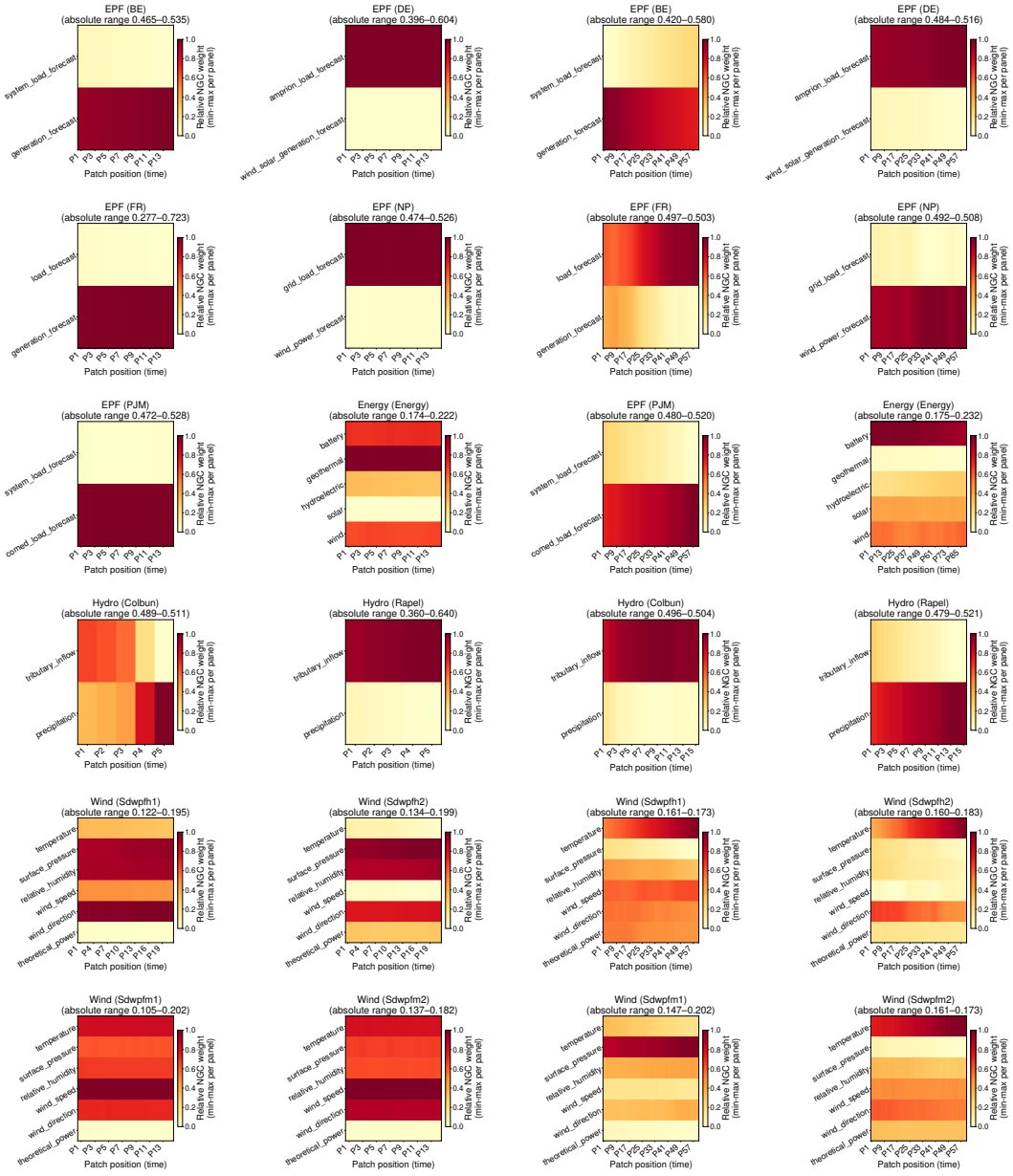
F Additional Visualizations

F.1 Additional NGC Causal Weight Heatmaps

Fig. 5 shows heatmaps of graph causal weights (Tank et al., 2022) for our interpretability analysis across all 12 datasets. Fig. 5(a) shows the result under short-term settings, while (b) shows the result for long-term settings. These figures provide an interpretable view of which exogenous variables contribute more strongly to the endogenous target.

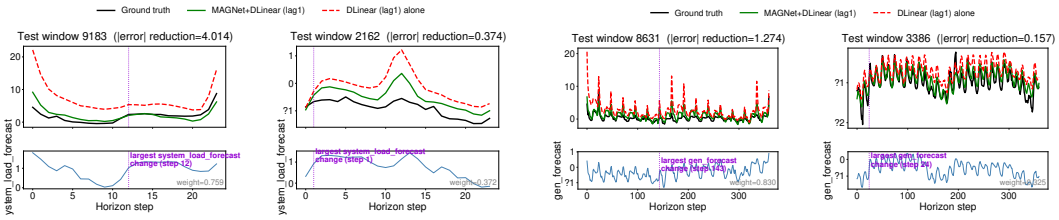
F.2 Additional Case Studies

Fig. 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 show all the case study for both short- and long-term forecasting settings, (a) and (b), respectively, for the 2 distinct test windows selected for all 12 datasets. This shows that MAGNet consistently improves prediction by yielding a forecast closer to the ground truth than vanilla DLinear under both forecasting terms and across all datasets.



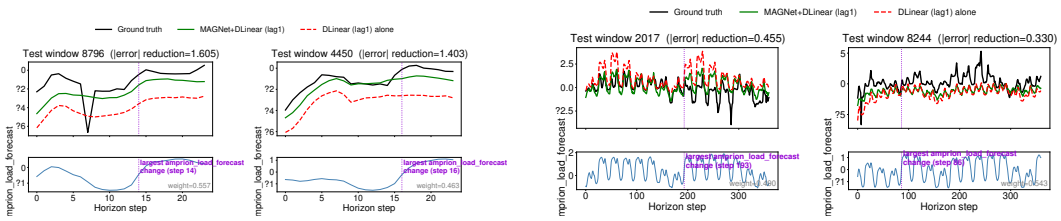
(a) Short-term (b) Long-term

Figure 5: NGC causal weights by domain (MAGNet+DLinear), relative min-max scaled per panel.



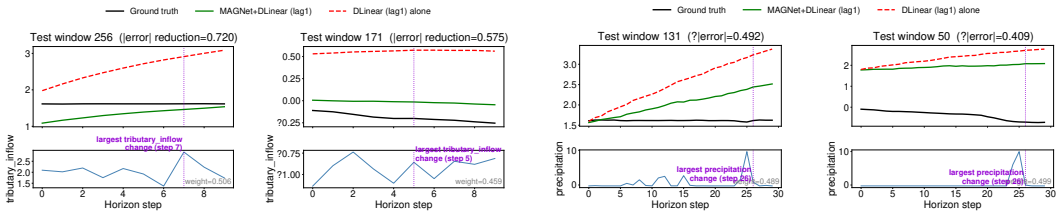
(a) Short-term (b) Long-term

Figure 6: Case study on BE: MAGNet+DLinear vs. DLinear alone.



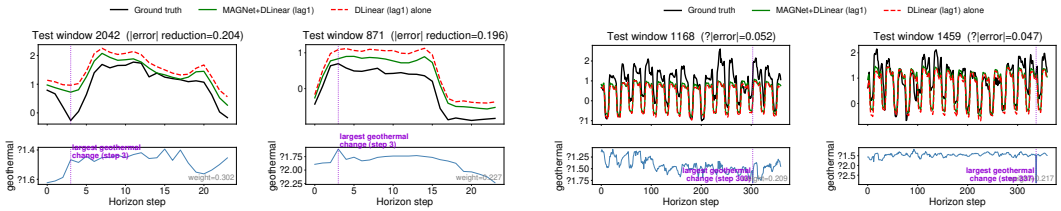
(a) Short-term (b) Long-term

Figure 7: Case study on DE: MAGNet+DLinear vs. DLinear alone.



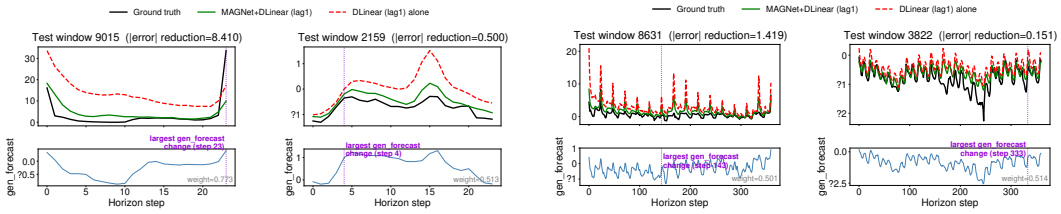
(a) Short-term (b) Long-term

Figure 8: Case study on Colbun: MAGNet+DLinear vs. DLinear alone.



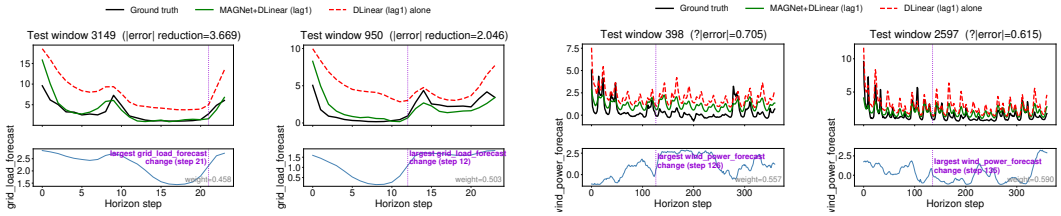
(a) Short-term (b) Long-term

Figure 9: Case study on Energy: MAGNet+DLinear vs. DLinear alone.



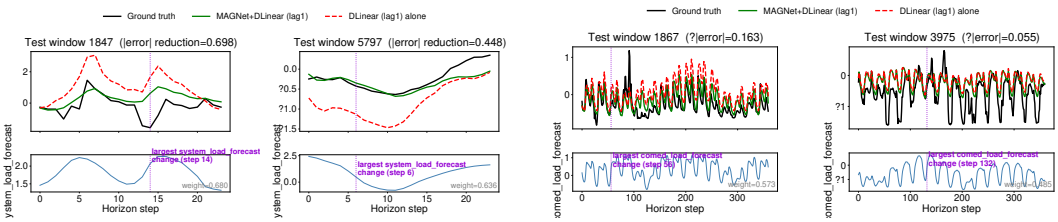
(a) Short-term (b) Long-term

Figure 10: Case study on FR: MAGNet+DLinear vs. DLinear alone.



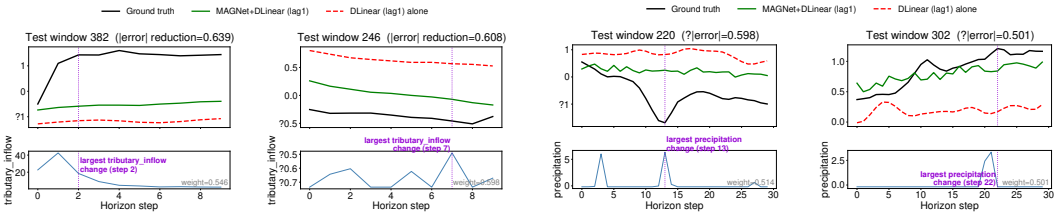
(a) Short-term (b) Long-term

Figure 11: Case study on NP: MAGNet+DLinear vs. DLinear alone.



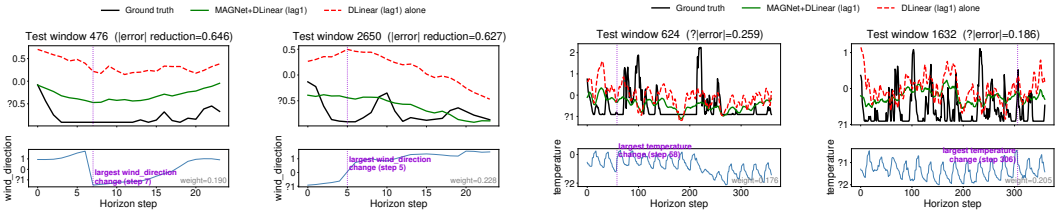
(a) Short-term (b) Long-term

Figure 12: Case study on PJM: MAGNet+DLinear vs. DLinear alone.



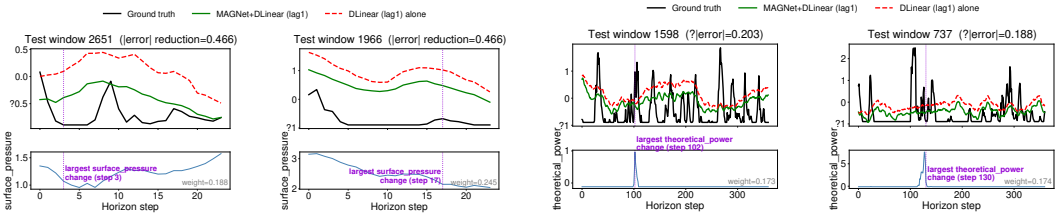
(a) Short-term (b) Long-term

Figure 13: Case study on Rapel: MAGNet+DLinear vs. DLinear alone.



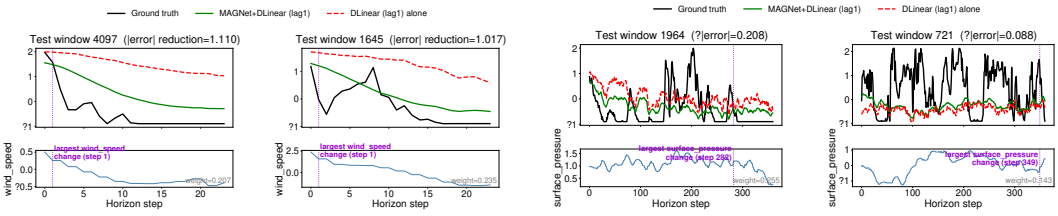
(a) Short-term (b) Long-term

Figure 14: Case study on Sdwpfh1: MAGNet+DLinear vs. DLinear alone.



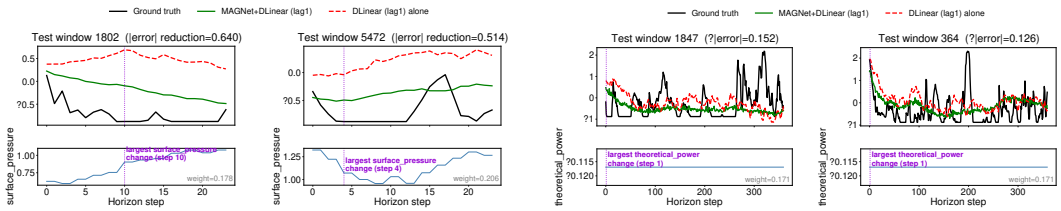
(a) Short-term (b) Long-term

Figure 15: Case study on Sdwpfh2: MAGNet+DLinear vs. DLinear alone.



(a) Short-term (b) Long-term

Figure 16: Case study on Sdwpfm1: MAGNet+DLinear vs. DLinear alone.



(a) Short-term (b) Long-term

Figure 17: Case study on Sdwpfm2: MAGNet+DLinear vs. DLinear alone.