EventConnector: Mining Social Event Relations through Temporal Graphs

Anonymous ACL submission

Abstract

Understanding and retrieving related real-world events based on their temporal dynamics is a fundamental challenge in time-sensitive applications such as forecasting, information retrieval, and social analysis. Existing methods often rely on semantic similarity or global time-series alignment, which overlook the transient and directional dependencies that frequently underlie real-world correlations. In this work, we introduce **EventConnector**, a general framework for constructing a temporal event graph that captures localized co-fluctuations and lead-lag relationships between events through their time-series trajectories. The resulting graph encodes both synchronous activity and directional influence, enabling the discovery of non-obvious, cross-domain associations. To further enrich the graph structure, we incorporate a multi-hop detection mechanism that reveals transitive temporal dependencies. Experiments on real-world prediction market data show that EventConnector uncovers non-trivial temporal structures and achieves a substantial 18.89% improvement in event retrieval and time-series forecasting tasks under limited supervision. These results highlight the effectiveness of temporal graph modeling in capturing latent event relationships beyond what semantic similarity or traditional alignment techniques can offer.

1 Introduction

Real-world events rarely unfold in isolation—they are embedded within interdependent systems spanning political, economic, and cultural domains. Modeling the temporal dependencies among such events is crucial not only for forecasting, but also for understanding how societal processes co-evolve. For example, a major fiscal policy announcement can ripple through financial markets, as evidenced by studies linking President Trump's public statements to fluctuations in cryptocurrency prices (Huynh, 2021). Anticipating these cross-domain ripple effects is essential for informing public policy, risk assessment, and strategic decision-making, yet it remains a challenging and underexplored problem.

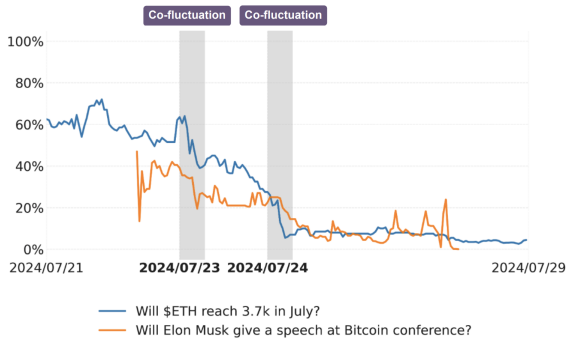

Figure 1: **Example event relations extracted by EventConnector**. Despite being semantically unrelated, the two social events—*"Will $ETH reach 3.7k in July?"* (blue) and *"Will Elon Musk give a speech at Bitcoin conference?"* (orange)—exhibit a strikingly similar temporal trajectory in their forecast probabilities. This strong short-term correlation reveals latent coupling in public sentiment or shared speculative dynamics, which are not captured by traditional semantic similarity.

One major obstacle is that correlated events are often *semantically dissimilar*. A shift in trade policy, for instance, may temporally align with movements in cryptocurrency prices, even though there is little lexical or ontological similarity to suggest a connection. As shown in Figure 1, such non-obvious dependencies may reflect latent societal dynamics or shared drivers of attention. Detecting these connections is difficult, particularly when confounded by high noise, domain heterogeneity, or lack of explicit structure. Traditional tools—such as Granger causality (Granger, 1969) or Hawkes processes (Hawkes, 1971)—often assume stationarity or low-noise environments and struggle to scale to complex, event-driven signals. Moreover, existing event forecasting datasets (e.g.,

ICEWS (Boschee et al., 2015), GDELT (Leetaru and Schrodt, 2013)) focus on discrete or single-domain events, and do not capture the full spectrum of continuous, interrelated fluctuations across domains.

Modern time-series and knowledge graph models also face limitations when applied to heterogeneous events. Deep forecasting models often assume variables can be fused into a shared latent space or rely on predefined relational structures—an assumption that holds for homogeneous systems like traffic networks (Li et al., 2018), but breaks down for semantically disjoint events. Temporal knowledge graph methods (Goel et al., 2020; Han et al., 2020) similarly embed all entities into a unified space, which risks collapsing structurally distinct signals and missing subtle, cross-domain interactions. These models often require dense supervision, assume stable dynamics, or overlook transient correlations that are critical in volatile environments like social prediction markets or public discourse.

In this work, we present **EventConnector**, a temporal graph framework for discovering and modeling dynamic dependencies between social events based on their evolving time series. Our method constructs a data-driven event graph in which nodes represent individual events and edges encode localized, statistically significant relationships derived from short-term co-fluctuation and lead-lag inference. This graph captures both synchronous and directional influence, and supports multi-hop reasoning over indirect chains of dependency.

By preserving the individuality of events while linking them through empirical temporal patterns, EventConnector enables more accurate forecasting under sparse supervision and facilitates interpretable analyses of how public attention or sentiment propagates across domains. Unlike existing approaches, it does not rely on predefined taxonomies, global embeddings, or semantic similarity. Instead, it embraces the heterogeneity of real-world signals and models structure as it emerges from data.

Our key contribution is the development of a temporal graph-based framework that elicits non-semantic connections between heterogeneous social events by identifying localized, directional dependencies in time series data. We further demonstrate that this structure significantly improves performance in event forecasting tasks—especially in inductive or low-supervision settings—outperforming both semantic and time-series retrieval baselines across domains and granularities by 18.89%, and offering a principled foundation for future extensions in social signal analysis and temporal reasoning.

2 Related Works

Time-Series Event Modeling Classical techniques like Dynamic Time Warping (DTW) (Berndt and Clifford, 1994), local correlation tracking (Papadimitriou et al., 2006), and BRAID (Sakurai et al., 2005) align or group time series based on transient or lagged patterns. Matrix profile methods (Yeh et al., 2016) efficiently detect similar or anomalous subsequences. Directional dependencies are modeled through high-dimensional Granger causality (Arnold et al., 2007) and lead-lag networks (Bennett et al., 2022). Point-process models like multidimensional Hawkes processes capture self-/cross-exciting dynamics (Zhou et al., 2013). Deep forecasting models (e.g., LSTNet (Lai et al., 2018)) leverage convolutional and recurrent layers for multiscale temporal dependencies. These approaches inform EventConnector's use of co-fluctuation and causality for graph construction.

Social System Modeling Foundational models—DeGroot averaging (DeGroot, 1974), threshold-based diffusion (Granovetter, 1978), and bounded-confidence dynamics (Hegselmann and Krause, 2002)—explain macro patterns from individual behavior. Data-driven systems such as EMBERS (et al., 2014), spatio-temporal forecasting (Zhao et al., 2015), and nested MIL (Ning et al., 2016) infer emergent trends from open signals. Evolving semantic graphs (Deng et al., 2019b) further capture event interplay. EventConnector builds on these ideas, defining a temporal event graph grounded in co-fluctuation and lead-lag signals for structure-aware forecasting.

Social Event Forecasting Earlier methods used social media and statistical signals (e.g., scan statistics (Chen and Neill, 2014), cascade models (Cadena et al., 2015)) to forecast unrest. Temporal event chains (Radinsky and Horvitz, 2012) and entity-centric graph models (Deng et al., 2019a, 2020; Zhou et al., 2022) incorporate cause-effect and multimodal dynamics. Our approach continues this line by modeling localized temporal dependencies via event graphs for enhanced retrieval and forecasting.

Temporal Graphs Temporal GNNs like TGAT

(Xu et al., 2020), DySAT (Sankar et al., 2020), Know-Evolve (Trivedi et al., 2017), and DyRep (Trivedi et al., 2019) embed evolving node relations via time-aware attention or event-driven dynamics. Message-passing models like TeMP (Wu et al., 2020) propagate over time-stamped knowledge graphs. EventConnector differs by defining temporal edges from time-series co-fluctuation and directional influence, enabling both inductive retrieval and forecasting.

3 Social Event Prediction

Prediction markets aggregate dispersed information and beliefs to form collective forecasts about uncertain future outcomes (Wolfers and Zitzewitz, 2004). Among them, *Polymarket* is a prominent cryptocurrency-based platform that hosts real-money prediction markets on a diverse set of social questions spanning politics, economics, entertainment, and crypto. Each market tracks beliefs over time by assigning probabilistic prices to mutually exclusive outcomes. These market-implied prices are interpretable as consensus probabilities and have been shown to produce accurate and calibrated forecasts across domains. For instance, in the context of U.S. elections, prediction market probabilities have outperformed traditional polling-based methods in estimating victory chances (Rothschild, 2009).

Let us denote by $\mathcal{E} = \{e_1, e_2, \ldots, e_N\}$ a collection of real-world *events*, where each event e_i corresponds to a temporally evolving question or proposition about the world (e.g., "Will a political candidate win the election?" or "Will Bitcoin reach $40,000 by next month?"). In our setting, an event is *observed indirectly through its time series trajectory*, which reflects public belief or collective probability estimation over time. These temporally evolving belief series allow us to cast event modeling and forecasting as a structured time-series problem grounded in human expectations and behavioral signals.

3.1 Social Events as Time-Series

We define a *social event* as a tuple

$$e = (q, \mathcal{O}), \tag{1}$$

where q is a future-uncertain question and $\mathcal{O} = \{o_1, \ldots, o_K\}$ is a set of K mutually exclusive outcomes. Each outcome $o_k \in \mathcal{O}$ is associated with a probability time series $\{p_k(t)\}_{t=1}^{T}$, where $p_k(t) \in [0, 1]$ denotes the market-implied probability of o_k at time step t. The probabilities for all outcomes are normalized at each timestamp:

$$\sum_{k=1}^{K} p_k(t) = 1, \quad \forall t \in \{1, \ldots, T\}. \tag{2}$$

This structure captures the temporal evolution of public beliefs over possible futures. For example, in a binary election event where q is *"Will candidate A win?"* and $\mathcal{O} = \{\text{Yes}, \text{No}\}$, the time series $p_{\text{Yes}}(t)$ and $p_{\text{No}}(t) = 1 - p_{\text{Yes}}(t)$ reflect belief shifts driven by campaign events, polling results, and media coverage.

Similarly, for a market-based question such as *"Will coin X exceed $P by date D?"*, the associated probability series evolves in response to price trends, market sentiment, and macroeconomic signals. These examples illustrate how time series trajectories encode real-time belief updates about evolving social outcomes.

Following this formalization, we treat each social event as a multi-outcome time series instance defined by Equation (1), and use its evolving probability dynamics as the core representation for downstream forecasting and graph construction.

3.2 Forecasting Future Social Events

Given the historical belief trajectories $\{p_k(1), \ldots, p_k(T)\}$ for each outcome $k \in \{1, \ldots, K\}$, the forecasting objective is to predict the next H values $\{p_k(T+1), \ldots, p_k(T+H)\}$. This task can be framed as a multi-horizon time-series prediction problem, where the goal is to anticipate the evolution of collective belief under ongoing information flow.

Forecasting such time series is challenging due to the non-stationary nature of belief formation, the influence of exogenous shocks (e.g., breaking news), and the heterogeneous domain contexts across events. Yet, it offers a unique and practical testbed for advancing the development of robust and adaptive forecasting models in settings that mirror real-world decision-making dynamics.

4 Social Temporal Graph

Building on the formulation of social events as time-evolving probability series introduced in the previous section, we now describe how such events can be connected into a structured representation—a *social temporal graph*—that captures inter-

event relationships grounded in their temporal dynamics.

4.1 Temporal Graphs

A *temporal graph* is a dynamic extension of a conventional graph structure in which the nodes, edges, or their associated attributes evolve over time and interact through time-dependent relationships. Formally, a temporal graph is defined as a tuple $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T)$, where $\mathcal{V}$ is a set of nodes, and $\mathcal{E}_T \subseteq \mathcal{V} \times \mathcal{V} \times R_+$ is a set of time-stamped edges. Each temporal edge $(u, v, t) \in \mathcal{E}_T$ indicates an interaction or dependency between nodes u and v that is active at time t, or over a continuous interval $[t_{\text{start}}, t_{\text{end}}]$ in cases of extended temporal influence. Unlike static graphs, $\mathcal{G}_T$ supports the analysis of causality, influence propagation, and dynamic neighborhood evolution by encoding *when* connections occur, not just *whether* they exist.

In our setting, we specialize $\mathcal{G}_T$ to the social domain. Specifically, we define a *social temporal graph* as a directed temporal graph $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T)$ where:

- Each node $v_i \in \mathcal{V}$ represents a unique social event $e_i = (q_i, \mathcal{O}_i)$, as defined earlier, together with its associated multivariate time series that reflects public belief updates over time.
- Each temporal edge $(v_i, v_j, t) \in \mathcal{E}_T$ denotes a time-specific correlation or influence between two events, inferred from the behavior of their respective time series.

The key feature of our approach is that edges in $\mathcal{E}_T$ are not constructed based on semantic similarity or textual content. Instead, each $(u, v, t) \in \mathcal{E}_T$ is derived from observed patterns in belief trajectories—specifically, when two time series exhibit meaningful, statistically significant co-fluctuations such as synchronous surges, consistent lead-lag patterns, or recurring alignment of belief shifts.

This structure captures latent social dependencies that may not be semantically obvious. For instance, a spike in belief about a major fiscal policy announcement might be followed by a shift in sentiment regarding a cryptocurrency threshold event. While the two events may differ in topic and phrasing, their co-evolution over time suggests an implicit connection shaped by external public discourse or shared information triggers.

Such patterns of temporally aligned fluctuations between events frequently reveal non-trivial connections that would be overlooked by static or text-based approaches. By representing events as nodes and their dynamic interactions as temporal edges, the social temporal graph $\mathcal{G}_T$ provides a principled framework for reasoning over social event systems. It enables the study of how public beliefs shift not just within isolated events, but across an interconnected landscape of co-evolving signals. This structured representation lays the foundation for downstream applications such as event retrieval, forecasting augmentation, and influence pathway discovery, which benefit from modeling the relational context of temporal social data.

4.2 Constructing the Social Temporal Graph

To construct the social temporal graph used in our framework, we operationalize the modeling principles described in the previous section by processing a large collection of time-series-based social events. Our construction pipeline is designed to capture non-trivial dependencies between events while suppressing redundancy and noise. The resulting temporal graph $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T)$ is built through three key stages: node construction via event filtering and merging, temporal edge construction via dynamic time-series correlation, and multi-hop enrichment via transitive similarity.

Node Construction. Each node $v \in \mathcal{V}$ in $\mathcal{G}_T$ corresponds to a unique event drawn from the Polymarket dataset. However, we observe that many events are syntactically distinct yet semantically equivalent or statistically redundant—such as phrasing variants or duplicate markets split across temporal boundaries. These event pairs often exhibit highly correlated belief trajectories with synchronized fluctuations over time. Including them as separate nodes would degrade the quality of $\mathcal{G}_T$, inflate local neighborhoods, and reduce diversity in retrieved results. To address this, we perform an event merging step that consolidates near-duplicate events into a single representative node. Merging is triggered when multiple events demonstrate consistently strong temporal alignment across sliding windows. The resulting node inherits the averaged trajectory of its constituent events, enhancing signal robustness while preserving the temporal structure.

Temporal Edge Construction. Temporal edges $(u, v, t) \in \mathcal{E}_T$ are added to capture latent inter-

event relationships that manifest through co-evolving probability dynamics. Our multi-pronged approach includes:

- **Direct temporal correlation:** We identify statistically significant co-movement between aligned time series over local windows. When two nodes exhibit synchronized sharp fluctuations across multiple intervals, we introduce a temporal edge (u, v, t) based on their lead-lag pattern.
- **Directional influence detection:** To capture asymmetric dependencies, we analyze cross-correlation structures to determine which node consistently leads the other. The direction of (u, v, t) reflects this hypothesized influence.
- **Transitive multi-hop enrichment:** For node pairs lacking strong direct correlation, we infer indirect connections by evaluating 2-hop paths through an intermediate node w where both (u, w, t) and (w, v, t') exhibit high Dynamic Time Warping (DTW) similarity. These edges are filtered by a similarity threshold to retain only meaningful second-order dependencies.

This construction ensures that $\mathcal{E}_T$ remains both expressive and sparse—highlighting informative temporal dependencies while suppressing noise.

Statistics of the Social Temporal Graph. To assess the structure of $\mathcal{G}_T$, we compute graph-level statistics, including the number of unique nodes $|\mathcal{V}|$ (post-merging), average out-degree per node (capturing neighborhood density), the ratio of directed versus bidirectional edges in $\mathcal{E}_T$, and the proportion of edges derived from multi-hop enrichment. We further analyze the distribution of temporal lags associated with edges (u, v, t) to quantify typical inter-event response times. These statistics, summarized in Table 1, offer insight into the dynamic structure of social discourse as encoded by $\mathcal{G}_T$.

This temporal graph construction framework serves as the backbone of our retrieval system, enabling structure-aware similarity and influence-aware reasoning over evolving public belief dynamics.

Table 1: **Statistics of Social Temporal Graph across Five Domains.** These statistics reflect the structural diversity and temporal dynamics captured by our graph construction pipeline.

Statistic	Politics	Sports	Crypto	Election	Other
# Unique Nodes	236	135	105	120	258
# Total Edges	6274	470	1087	1735	2589
Avg. Degree per Node	53.17	6.96	20.70	28.92	20.07
Graph Density	0.23	0.05	0.20	0.24	0.08
Average Weight	0.80	0.82	0.83	0.80	0.82

5 EventConnector

Having established the structure of the social temporal graph, we now describe how it serves as a retrieval-augmented mechanism for linking unseen events to historically grounded relational contexts. We refer to this module as *EventConnector*, which enables both inductive event retrieval and structure-aware forecasting by integrating neighborhood signals from the graph.

5.1 Retrieving on the Social Temporal Graph

Given a query event $e_q = (q, \mathcal{O})$ with associated outcome probability time series $\{p_k^q(t)\}_{t=1}^T$, the goal of retrieval is to locate the most temporally and structurally relevant region of the graph. Although the query event is not part of the original graph (i.e., it is an inductive, out-of-graph instance), we enable linkage by first mapping it to the most similar node in the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

Query Mapping. We compute similarity between the query time series and each in-graph node using a combination of temporal similarity metrics (e.g., DTW or Pearson correlation over recent history windows). The graph node $v^* \in \mathcal{V}$ with the highest similarity score is selected as the *anchor node* for the query event.

Neighborhood Expansion. Once the anchor node v^* is identified, we perform an n-hop neighborhood expansion in $\mathcal{G}$ to collect structurally related events. This neighborhood, denoted $\mathcal{N}_n(v^*)$, includes both direct and transitive temporal correlates of the query. The flexibility in selecting n allows us to control the granularity of contextual information, where $n = 1$ focuses on strong direct relationships, while $n > 1$ enables access to higher-order latent clusters.

By leveraging the graph structure, this retrieval process enables inductive query events—those unseen during graph construction—to benefit from the relational signal embedded in the temporal event space. This is particularly valuable in real-world settings where new events emerge continuously and historical grounding is limited.

5.2 Forecasting with the Social Temporal Graph

We now describe how the retrieved events are used to enhance time-series forecasting for the query event. The key idea is to expose the forecasting model to temporally aligned supervision derived from the neighborhood of the anchor node.

Sliding Window Extraction. For each neighbor $e_i \in \mathcal{N}_n(v^*)$, we extract non-overlapping sliding windows from its time series to generate augmented training samples. This prevents temporal leakage and ensures that causal signals—such as lead-lag patterns—are preserved across windows.

Training Set Augmentation. The extracted windows from related events are then added to the training set of the base forecasting model. This augmentation allows the model to learn from structurally similar belief trajectories and generalize patterns that are aligned with the query's expected evolution.

In this way, the social temporal graph not only facilitates retrieval, but also serves as a source of inductive bias for downstream forecasting. The model becomes acquainted with the distributional properties of temporally and socially relevant events, thereby improving its ability to forecast the belief trajectory of the query event.

Table 2: **Domain-wise statistics for SocialWM under Daily and Hourly forecasting settings.** Each value reflects average statistics over all events in the domain.

Setting	Statistic	Politics	Sports	Crypto	Election	Other
Daily	# Events	98	20	23	36	87
	Avg. Length	84.43	59.45	75.17	80.47	54.69
	# Windows	1040	142	215	360	558
Hourly	# Events	135	145	49	56	163
	Avg. Length	1506.81	314.40	930.94	1298.23	780.12
	# Windows	8145	1551	1784	2892	4899

6 Experimental Setting

We conduct experiments on *SocialWM*, a real-world benchmark dataset collected from the prediction market platform Polymarket. This dataset spans five diverse domains—`Crypto`, `Election`, `Politics`, `Sports`, and `Other`—each capturing a collection of social events represented as probabilistic time series. Each event follows the definition formalized in Equation 1, where market-implied outcome probabilities evolve over time in response to real-world developments.

Data Description. We use time series data collected at both **daily** and **hourly** resolutions to evaluate forecasting performance under different temporal granularities. To create supervised forecasting samples, we apply a non-overlapping sliding window strategy with fixed-length history and prediction segments. Specifically, the daily setting uses a 7-day history to predict the next 7 days, while the hourly setting uses a 48-hour history to forecast the next 24 hours. This design maximizes the effective use of each time series while avoiding data leakage. Domain-wise statistics, including the number of events, average sequence lengths, and resulting window counts, are summarized in Table 2.

Forecasting Models. We evaluate five representative forecasting architectures: (1) **DLinear** (Zeng et al., 2023), a fast, interpretable model based on seasonal-trend decomposition; (2) **Autoformer** (Wu et al., 2021), a Transformer variant using auto-correlation to capture long-range patterns; (3) **Informer** (Zhou et al., 2021), which employs ProbSparse attention for efficient long-horizon prediction; (4) **N-BEATS** (Oreshkin et al., 2019), a deep residual network with backward and forward blocks; and (5) **TimesNet** (Wu et al., 2022), which integrates temporal and frequency-aware representations.

Retrieval-Based Comparison. To assess the impact of retrieval strategies, we benchmark our proposed **EventConnector** against: (1) **Few-Shot Forecasting**, using samples from unrelated events with minimal context; (2) **Semantic Retrieval**, retrieving events by question similarity in the natural language space; (3) **Time Series Retrieval**, identifying nearest neighbors using DTW distance over full series; and (4) **Full-Shot Forecasting**, trained on the entire domain as an oracle upper bound.

Evaluation Metrics. We report standard forecasting metrics: (1) **Mean Absolute Error (MAE)**, which measures the average absolute deviation between prediction and ground truth; and (2) **Root Mean Squared Error (RMSE)**, which emphasizes larger deviations via squared error aggregation. These jointly capture average and extreme predictive discrepancies across domains and time resolutions.

7 Experimental Results

We evaluate **EventConnector** against multiple baselines across five domains—`Politics`, `Sports`, `Crypto`, `Election`, and `Other`. Table 3 reports average RMSE and MAE scores under both **daily** (7-day horizon) and **hourly** (24-hour horizon) set-

Table 3: **Average Forecasting Performance on Out-of-Graph Events across Domains under Daily and Hourly Settings.** Each RMSE and MAE value is averaged over five forecasting models: DLinear (Zeng et al., 2023), Autoformer (Wu et al., 2021), Informer (Zhou et al., 2021), N-BEATS (Oreshkin et al., 2019), and TimesNet (Wu et al., 2022).

	Politics		Sports		Crypto		Election		Other	
Model (Daily)	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Few-Shot	0.3604	0.3008	0.3400	0.2772	0.3169	0.2568	0.3710	0.3117	0.3533	0.2946
Semantic Retrieval	0.3088	0.2546	0.4260	0.3891	0.3267	0.2670	0.3598	0.3042	0.3244	0.2709
Time-Series Retrieval	0.3041	0.2526	0.3394	0.2845	0.2941	0.2379	0.3633	0.3038	0.3371	0.2809
EventConnector	**0.2655**	**0.2226**	**0.2629**	**0.2160**	**0.2319**	**0.1896**	**0.2833**	**0.2351**	**0.2837**	**0.2321**
Full-Shot	0.2181	0.1789	0.2617	0.2090	0.2185	0.1745	0.2523	0.2082	0.2508	0.1949
Model (Hourly)	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Few-Shot	0.4139	0.3569	0.4090	0.3528	0.3712	0.3174	0.4216	0.3643	0.4154	0.3549
Semantic Retrieval	0.2520	0.2059	0.2471	0.2151	0.2653	0.2257	0.2888	0.2370	0.2681	0.2183
Time-Series Retrieval	0.2479	0.2048	0.3207	0.2747	0.3137	0.2696	**0.2502**	**0.2183**	0.2718	0.2309
EventConnector	**0.2241**	**0.1828**	**0.2125**	**0.1703**	**0.2299**	**0.1898**	0.2665	0.2237	**0.2465**	**0.1996**
Full-Shot	0.1602	0.1238	0.1856	0.1505	0.1748	0.1352	0.1868	0.1457	0.1739	0.1373

tings on five forecasting models: DLinear (Zeng et al., 2023), Autoformer (Wu et al., 2021), Informer (Zhou et al., 2021), N-BEATS (Oreshkin et al., 2019), and TimesNet (Wu et al., 2022), enabling model-agnostic comparison.

EventConnector consistently outperforms all retrieval-based and few-shot baselines. across both time granularities and all domains. In the daily setting, it achieves the lowest RMSE and MAE throughout, e.g., in `Politics`, EventConnector yields an RMSE of **0.2655**, outperforming time-series retrieval (0.3041) and semantic retrieval (0.3088). Similarly, in the volatile `Crypto` domain, it achieves **0.2319**, a significant margin over DTW (0.2941) and semantic methods (0.3267).

The trend holds in the hourly setting, where EventConnector leads in four of five domains. For instance, it records RMSE scores of **0.2241** and **0.2465** in `Politics` and `Other`, outperforming both semantic and time-series retrieval. These results affirm that graph-based inductive signals enhance forecasting regardless of temporal resolution or domain volatility.

Semantic retrieval, while leveraging textual similarity, overlooks directional and temporal dynamics critical for forecasting. Time-series retrieval improves alignment but lacks structured context. EventConnector mitigates both limitations by capturing short-term co-fluctuation and directional influence within a temporal event graph.

EventConnector also approaches full-shot performance using limited supervision. In the daily `Election` domain, it achieves an RMSE of **0.2833**, within 12.3

Importantly, EventConnector remains robust across time granularities and domains. Its consistent gains in both daily and hourly resolutions confirm the ability of temporal graphs to capture multi-scale dependencies. Moreover, the method generalizes well across topics from stable political questions to dynamic crypto markets, demonstrating strong inductive bias and generalization.

In summary, EventConnector delivers consistent and substantial improvements across all evaluated settings. It outperforms retrieval-based and few-shot baselines across both **daily** and **hourly** time granularities, confirming its ability to capture fine-grained as well as coarse temporal dependencies. Its performance advantage generalizes across diverse domains, demonstrating strong inductive bias and robustness to domain variation. Furthermore, the observed gains hold across five distinct forecasting architectures, highlighting the **model-agnostic** nature of the framework.

8 Discussion

Why do semantic and time-series retrieval baselines underperform? While both semantic and time-series-based retrieval methods provide useful baselines, they fail to capture the relational richness exploited by EventConnector. Semantic retrieval relies solely on textual similarity between event descriptions and is agnostic to the actual evolution of belief over time. As a result, it often retrieves

events that are topically similar but temporally uncorrelated. Time-series retrieval based on full DTW distances considers the entire trajectory, which may overly emphasize global alignment while ignoring localized or transient co-fluctuations. In contrast, EventConnector focuses on dynamically co-evolving segments and incorporates lead-lag relationships into its graph structure, effectively grounding retrieval in both temporal dynamics and relational context. Thus, although EventConnector can be seen as a special form of time-series retrieval, its structural awareness and focus on causal temporal patterns allow it to surpass naive DTW-based methods.

Does hop number affect downstream performance? An important design choice in our framework is the neighborhood depth—i.e., the number of hops used during retrieval. We find that increasing the number of hops generally leads to improved forecasting performance across domains, as more relational signals are incorporated into the inductive prediction. Notably, the most significant gains occur when moving from 1-hop to 2-hop retrieval, suggesting that immediate and second-order neighbors together capture the majority of useful contextual information. However, as shown in Figure 2, the performance gains begin to plateau beyond 2-hop, indicating a clear diminishing return effect. This trend highlights that while incorporating additional hops can be beneficial, most of the forecasting signal is concentrated in the first two levels of the event graph.

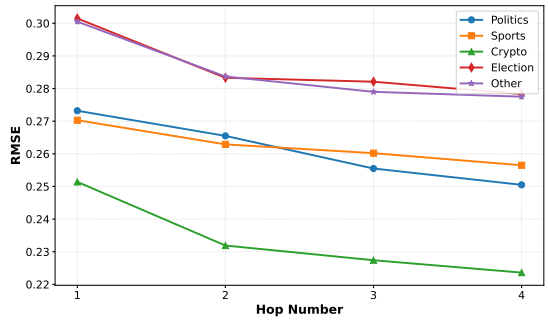

Figure 2: Forecasting performance comparison across different hop numbers used for retrieval.

Does cross-domain retrieval help with single-domain prediction? One surprising and important finding is that cross-domain retrieval—retrieving events from different topical domains—can still improve single-domain forecasting performance. For example, a political event may exhibit strong co-fluctuation with a cryptocurrency market event, even though the two are semantically unrelated. This suggests that social signals often transcend traditional domain boundaries and that latent belief dynamics may be driven by shared external shocks or macro-level sentiment flows. The ability of EventConnector to uncover and exploit such cross-domain linkages points toward the promise of scaling up the social temporal graph across domains to further enhance generalization and retrieval coverage.

Can this method generalize beyond SocialWM-Bench? Although our experiments are conducted on SocialWM-Bench, the proposed retrieval and forecasting framework is not limited to prediction market data. Any domain where entities are associated with belief-like or attention-driven time series—such as financial instruments (e.g., stock prices, currency exchange rates), public opinion polls, or media engagement metrics—could potentially benefit from a similar temporal graph construction and retrieval approach. This opens the door to generalizing EventConnector to broader real-world forecasting tasks where transient and directional dependencies between time series are prevalent. In future work, we plan to extend our framework to these domains and investigate how relational inductive biases can be adapted to non-social, high-frequency temporal systems.

9 Conclusion

We tackle the challenge of forecasting public opinion and retrieving real-world events based on their temporal dynamics—a critical task amid rapidly evolving social discourse. Traditional models often falter due to the transient and interdependent nature of social signals. To address this, we propose **EventConnector**, a temporal graph framework that captures localized, directional co-fluctuations between events. By linking belief-evolving time series through statistically grounded lead-lag dependencies, our method enables effective retrieval and forecasting under sparse supervision. Empirical results across domains and granularities show that EventConnector consistently outperforms few-shot and retrieval baselines, rivaling full-shot forecasting while requiring less data. This highlights the value of graph-structured temporal context in improving generalization and predictive accuracy.

10 Limitations

While **EventConnector** demonstrates strong performance in capturing temporal dependencies and improving forecasting accuracy, several limitations remain. First, the construction of the temporal graph relies on observed co-fluctuations in time-series data, which may miss latent but semantically relevant relationships not reflected in belief dynamics. Second, the current edge construction mechanism assumes stationarity of lead-lag patterns over time, which may not hold in rapidly evolving domains. Third, our method focuses on pairwise and transitive interactions, but does not yet model higher-order dependencies such as joint influence from multiple events. Finally, while the framework is model-agnostic, its effectiveness can vary depending on the quality and resolution of the underlying time-series signals, which may be sparse or noisy in certain real-world applications.

References

Andrew Arnold, Yan Liu, and Naoki Abe. 2007. Temporal causal modeling with graphical granger methods. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 66–75.

Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. 2022. Lead–lag detection and network clustering for multivariate time series. *Machine Learning*, 111(8):4497–4538.

David J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *AAAI-94 Workshop on Knowledge Discovery in Databases (KDD)*, pages 359–370.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS coded event data. Harvard Dataverse, V12.

Cesar A Cadena and 1 others. 2015. Forecasting social unrest using activity cascades. In *ICWSM*.

Fanglan Chen and Daniel B. Neill. 2014. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *ACM SIGKDD*.

Morris H. DeGroot. 1974. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121.

Han Deng and 1 others. 2019a. A graph-based event propagation model for civil unrest forecasting. In *ICDM*.

Han Deng and 1 others. 2020. Glean: Event forecasting with context graph modeling. In *ACL*.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2019b. Learning dynamic context graphs for predicting social events. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1000–1008.

Naren Ramakrishnan et al. 2014. 'beating the news' with embers: Forecasting civil unrest using open source indicators. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1799–1808.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3988–3995.

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

Mark Granovetter. 1978. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations (ICLR)*.

Alan G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90.

Rainer Hegselmann and Ulrich Krause. 2002. Opinion dynamics and bounded confidence: Models, analysis and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3):2.

Toan Luu Duc Huynh. 2021. Does bitcoin react to trump's tweets? *Journal of Behavioral and Experimental Finance*, 31:100546.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long- and short-term temporal patterns with deep neural networks. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 95–104.

Kalev Leetaru and Philip A. Schrodt. 2013. GDELT: Global data on events, location, and tone, 1979–2012. In *Proceedings of the International Studies Association Annual Convention*.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Yue Ning, Sathappan Muthiah, Huzefa Rangwala, and Naren Ramakrishnan. 2016. Modeling precursors for event forecasting via nested multi-instance learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1415–1424.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

Spiros Papadimitriou, Jimeng Sun, and Philip S. Yu. 2006. Local correlation tracking in time series. In *IEEE International Conference on Data Mining (ICDM)*, pages 456–465.

Kira Radinsky and Eric Horvitz. 2012. Modeling and predicting behavioral dynamics on the web. In *WWW*.

David Rothschild. 2009. Forecasting elections: Comparing prediction markets, polls, and their biases. *Public Opinion Quarterly*, 73(5):895–916.

Yasushi Sakurai, Spiros Papadimitriou, and Christos Faloutsos. 2005. Braid: Stream mining through group lag correlations. In *ACM SIGMOD International Conference on Management of Data*, pages 599–610.

Aravind Sankar and 1 others. 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *WSDM*.

Rakshit Trivedi, Mehrdad Farajtabar, and 1 others. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *ICML*.

Rakshit Trivedi and 1 others. 2019. Dyrep: Learning representations over dynamic graphs. In *ICLR*.

Justin Wolfers and Eric Zitzewitz. 2004. Prediction markets. *Journal of Economic Perspectives*, 18(2):107–126.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430.

Xinyi Wu and 1 others. 2020. Temp: Temporal message passing for temporal knowledge graph completion. In *EMNLP*.

Da Xu and 1 others. 2020. Inductive representation learning on temporal graphs. In *ICLR*.

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego F. Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile i: All pairs similarity joins for time series. In *IEEE International Conference on Data Mining (ICDM)*, pages 1317–1322.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

Liang Zhao, Qiuhong Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2015. Multi-task learning for spatio-temporal event forecasting. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1503–1512.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115.

Kai Zhou and 1 others. 2022. Eventgraph: Forecasting emerging events via temporal knowledge graphs. In *EMNLP*.

Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 641–649.

Appendix A: Detailed Forecasting Results

This appendix provides full quantitative results under the **inductive setting**, where models forecast unseen out-of-graph social events using limited or retrieved supervision. We report performance across five domains—`Politics`, `Sports`, `Crypto`, `Election`, and `Other`—under two temporal granularities: **daily** and **hourly**.

Table 4 presents the forecasting performance on the **daily** setting, where models use 7 days of history to predict the next 7 days. Table 5 shows the corresponding results for the **hourly** setting, using 48 hours of history to forecast the next 24 hours. Each row corresponds to a forecasting model evaluated with **Root Mean Squared Error (RMSE)** and **Mean Absolute Error (MAE)** across all domains and retrieval strategies.

Appendix B: Experiment Details

This appendix provides detailed information regarding the experimental setup, datasets, baseline implementations, proposed model configurations, and evaluation metrics used in Section 6.

Model Size and Computational Budget. We conduct experiments using five representative time-series forecasting architectures—DLinear (Zeng et al., 2023), Autoformer (Wu et al., 2021), Informer (Zhou et al., 2021), N-BEATS (Oreshkin et al., 2019), and TimesNet (Wu et al., 2022). All experiments were run on a single NVIDIA A6000 GPU (48GB), totaling approximately 200 GPU hours. This budget includes time for graph construction, hyperparameter tuning, model training, retrieval-based augmentation, and result visualization.

Experimental Setup and Hyperparameters. We adopt a consistent forecasting pipeline under both **daily** (7-day prediction) and **hourly** (24-hour prediction) settings. For each forecasting model, we perform a grid search over learning rates, batch sizes, and other related hyperparameters. The best-performing configuration is selected based on validation RMSE.

Baseline Models. For each baseline model, we utilized publicly available implementations where possible, adhering to the hyperparameter settings recommended in their original publications or widely adopted in benchmark studies. These models include:

- **DLinear** (Zeng et al., 2023): A lightweight and interpretable model based on seasonal-trend decomposition of time series. It assumes a linear mapping from decomposed components to future values, making it highly efficient and robust on short sequences. We used the original authors' open-source PyTorch implementation with default training settings.

- **Autoformer** (Wu et al., 2021): A Transformer-based model tailored for long-term time-series forecasting. It introduces an auto-correlation mechanism to replace traditional attention, effectively capturing periodic patterns in high-resolution sequences. We tuned context length and dropout based on validation RMSE.

- **Informer** (Zhou et al., 2021): This model uses ProbSparse self-attention to improve the efficiency of long sequence forecasting. It reduces the quadratic complexity of full attention and allows fast modeling of long temporal dependencies. We followed the original hyperparameter setup with minor adjustments to input length and learning rate.

- **N-BEATS** (Oreshkin et al., 2019): A deep residual forecasting architecture that uses backward and forward fully connected blocks to model temporal signals in a non-recurrent, interpretable fashion. Its block-based design allows for learning of trend and seasonality patterns with minimal assumptions. We used the official implementation with recommended forecast and backcast lengths.

- **TimesNet** (Wu et al., 2022): A recent model that integrates frequency-domain and temporal-domain representations via temporal 2D variation blocks. It achieves strong performance across general forecasting tasks by capturing multi-scale dependencies. We used the official implementation with input lengths set to match our daily and hourly configurations.

All baseline models were retrained on the same forecasting tasks and time windows as our proposed method to ensure fair comparison. Wherever applicable, early stopping and validation-based model selection were applied to avoid overfitting.

Graph Construction and Visualization Tooling. Graph generation, storage, and visualization

are implemented using `networkx`, `plotly`, and `torch_geometric`. These tools support topological processing, interactive visualization, and GPU-accelerated graph batching. The event graph is serialized in PyTorch format and serves as a backbone for both retrieval and inductive forecasting tasks.

Table 4: **Inductive** Setting: Forecasting Performance on out-of-graph events across Domains on **daily** time frequency, 7 history length and 7 prediction length.

	Politics		Sports		Crypto		Election		Other	
Model	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
few-shot forecasting										
DLinear	0.4308	0.3861	0.3826	0.3343	0.2938	0.2469	0.4438	0.4007	0.4002	0.3556
Autoformer	0.2561	0.2226	0.2688	0.2303	0.2530	0.2185	0.2575	0.2253	0.2167	0.1876
Informer	0.8061	0.6265	0.7601	0.5722	0.6865	0.5117	0.8372	0.6552	0.7435	0.5729
N-BEATS	0.1640	0.1385	0.1399	0.1165	0.1447	0.1197	0.1675	0.1422	0.1262	0.1027
TimesNet	0.1450	0.1303	0.1489	0.1329	0.2068	0.1872	0.1494	0.1351	0.2801	0.2545
Semantic Retrieval based forecasting										
DLinear	0.4306	0.3859	0.3828	0.3346	0.2939	0.2470	0.4439	0.4008	0.4001	0.3555
Autoformer	0.1081	0.0902	0.4179	0.3731	0.2067	0.1789	0.2058	0.1767	0.1243	0.1041
Informer	0.7531	0.5771	1.0781	1.0228	0.6798	0.5326	0.8418	0.6744	0.6950	0.5416
N-BEATS	0.1080	0.0903	0.1000	0.0803	0.2440	0.1871	0.1593	0.1350	0.1244	0.1008
TimesNet	0.1443	0.1297	0.1510	0.1347	0.2092	0.1896	0.1482	0.1341	0.2782	0.2525
Time-series Retrieval based Forecasting										
DLinear	0.4299	0.3850	0.3827	0.3344	0.2938	0.2468	0.4435	0.4004	0.3998	0.3551
Autoformer	0.1161	0.0985	0.2196	0.1888	0.1448	0.1204	0.0884	0.0738	0.1116	0.0956
Informer	0.7191	0.5549	0.8439	0.6843	0.6788	0.5091	0.8169	0.6375	0.7268	0.5559
N-BEATS	0.1113	0.0949	0.1000	0.0803	0.1451	0.1248	0.3191	0.2725	0.1671	0.1431
TimesNet	0.1446	0.1301	0.1510	0.1348	0.2080	0.1884	0.1487	0.1347	0.2803	0.2546
EventConnector-based forecasting										
DLinear	0.4235	0.3770	0.3815	0.3328	0.2926	0.2447	0.4402	0.3961	0.3937	0.3470
Autoformer	0.2113	0.1836	0.2057	0.1725	0.2187	0.1817	0.2618	0.2308	0.1857	0.1585
Informer	0.4754	0.3883	0.4963	0.3893	0.3460	0.2759	0.4932	0.3753	0.5051	0.3879
N-BEATS	0.0935	0.0680	0.0999	0.0777	0.1250	0.1019	0.0944	0.0729	0.1115	0.0881
TimesNet	0.1236	0.0960	0.1312	0.1079	0.1773	0.1440	0.1271	0.1005	0.2223	0.1791
Full-shot forecasting										
DLinear	0.4083	0.3408	0.3737	0.3208	0.2859	0.2346	0.4242	0.3727	0.3812	0.3196
Autoformer	0.0608	0.0500	0.1219	0.0977	0.0924	0.0764	0.0871	0.0730	0.0710	0.0602
Informer	0.4196	0.3471	0.5777	0.4365	0.4227	0.3279	0.5303	0.4250	0.4993	0.3752
N-BEATS	0.0863	0.0671	0.0969	0.0741	0.1101	0.0856	0.0936	0.0710	0.0988	0.0758
TimesNet	0.1156	0.0896	0.1384	0.1157	0.1814	0.1479	0.1263	0.0991	0.2037	0.1438

Table 5: **Inductive** Setting: Forecasting Performance on out-of-graph events across Domains on **hourly** time frequency, 48 history length and 24 prediction length.

	Politics		Sports		Crypto		Election		Other	
Model	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
few-shot forecasting										
DLinear	0.3998	0.3478	0.3932	0.3364	0.2767	0.2301	0.4125	0.3597	0.3856	0.3338
Autoformer	0.2359	0.2119	0.2290	0.2086	0.1421	0.1259	0.2686	0.2442	0.1961	0.1761
Informer	0.8958	0.7902	0.9534	0.8362	0.9269	0.8175	0.8760	0.7731	0.8597	0.7518
N-BEATS	0.3963	0.3227	0.3685	0.3029	0.3151	0.2578	0.3954	0.3234	0.3857	0.3161
TimesNet	0.1415	0.1118	0.1008	0.0800	0.1954	0.1557	0.1557	0.1212	0.2498	0.1968
Semantic Retrieval-based Forecasting										
DLinear	0.3780	0.3042	0.3842	0.3256	0.2695	0.2210	0.3977	0.3391	0.3648	0.2993
Autoformer	0.0288	0.0255	0.0667	0.0594	0.0442	0.0397	0.0323	0.0286	0.0354	0.0312
Informer	0.5709	0.4833	0.3645	0.2724	0.4937	0.4164	0.7184	0.5858	0.5535	0.4576
N-BEATS	0.1406	0.1075	0.1459	0.1135	0.1478	0.1188	0.1390	0.1081	0.1374	0.1073
TimesNet	0.1417	0.1090	0.1014	0.0806	0.1943	0.1533	0.1565	0.1235	0.2495	0.1963
Time-Series Retrieval-based Forecasting										
DLinear	0.3826	0.3219	0.3887	0.3312	0.2729	0.2256	0.4043	0.3495	0.3724	0.3164
Autoformer	0.0365	0.0307	0.0889	0.0731	0.0736	0.0636	0.0583	0.0488	0.0588	0.0508
Informer	0.5505	0.4769	0.8933	0.7819	0.9149	0.8093	0.5127	0.4707	0.5982	0.5175
N-BEATS	0.1383	0.1025	0.1360	0.1063	0.1364	0.1060	0.1349	0.1047	0.1362	0.1038
TimesNet	0.1316	0.1119	0.0968	0.0809	0.1706	0.1437	0.1409	0.1178	0.1933	0.1660
EventConnector-based Forecasting										
DLinear	0.3784	0.3069	0.3853	0.3270	0.2710	0.2231	0.3988	0.3410	0.3640	0.2961
Autoformer	0.0575	0.0523	0.0582	0.0521	0.0572	0.0520	0.0478	0.0427	0.0358	0.0319
Informer	0.4308	0.3585	0.5547	0.5036	0.6822	0.5984	0.6035	0.5140	0.4927	0.4038
N-BEATS	0.1357	0.1044	0.1496	0.1223	0.1430	0.1144	0.1423	0.1045	0.1348	0.1046
TimesNet	0.1180	0.0919	0.0879	0.0707	0.1732	0.1406	0.1401	0.1163	0.2052	0.1614
Full-shot forecasting										
DLinear	0.3771	0.2990	0.3695	0.2904	0.2610	0.1999	0.3891	0.3093	0.3625	0.2863
Autoformer	0.0276	0.0242	0.0419	0.0372	0.0360	0.0325	0.0233	0.0209	0.0298	0.0262
Informer	0.1363	0.1038	0.2946	0.2493	0.2426	0.1946	0.2114	0.1676	0.1572	0.1289
N-BEATS	0.1364	0.0979	0.1346	0.1053	0.1342	0.1028	0.1398	0.1032	0.1372	0.0988
TimesNet	0.1234	0.0943	0.0873	0.0701	0.2003	0.1462	0.1706	0.1276	0.1829	0.1461