LIFT OUT, KICK IN: DECOUPLING EVENT EFFECTS FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world time series are often shaped by exogenous events whose effects are superimposed on the system's underlying temporal dynamics. Historical observations therefore entangle persistent dynamics with event-induced deviations, making it difficult to determine which patterns should be extrapolated and which should be attributed to specific event contexts. Existing event-aware forecasting methods can incorporate event information or model event-related variation, but typically treat historical event attribution and future event generation separately. We propose Loki (Lift Out, Kick In), a general event-aware forecasting framework built on a shared Event Response Function (ERF). The ERF models event effects according to event type and relative temporal position and is shared across the historical and forecast windows. Loki lifts out estimated event-induced deviations from historical observations before forecasting, then uses the same ERF to kick in responses to known future events. This formulation separates event-response modeling from temporal forecasting while coupling historical attribution and future generation through a shared mechanism. Across five forecasting benchmarks, Loki achieves the best standardized MAE on all five datasets, reducing error by approximately 2.8%–6.0% over the strongest non-Loki baseline. Ablations and controlled analyses further support the roles of Lift Out, Kick In, shared response modeling, and compatibility across forecasting backbones.

1 INTRODUCTION

Time series forecasting in many real-world applications is influenced not only by the system's intrinsic temporal dynamics, but also by known external events such as holidays, promotions, planned activities, and extreme weather (Taylor & Letham, 2018; Rodrigues et al., 2019; Olivares et al., 2023; Wang et al., 2024b). Such events can induce substantial and temporally structured deviations in the observed trajectory, making event information an important part of forecasting in event-rich environments.

A central challenge, however, is that **not every pattern observed in history should be extrapolated into the future**. Forecasting models extrapolate from historical variation, yet when observations have been affected by external events, some variation reflects temporal dynamics that may persist, while other variation is specific to the event context in which it occurred. The forecasting problem is therefore not only to recognize which events occur, but also to determine which observed variation should be extrapolated as underlying dynamics and which should instead be attributed to event responses. This distinction becomes particularly important when the event configuration in the forecast horizon differs from that observed in history, as illustrated by the event-response carryover scenario in Figure 1(a).For example, a temporary deviation caused by a historical event should not be propagated into an event-quiet future, whereas a known upcoming event may require a response that is absent from the recent history. In both cases, directly extrapolating the observed trajectory risks conflating event-specific variation with the dynamics expected to persist.

Existing event-aware forecasting methods address different parts of this problem. A broad class of approaches incorporates historical or known-future contextual variables as conditioning information through feature encoding, fusion, or interaction mechanisms (Salinas et al., 2020; Lim et al., 2021; Das et al., 2023; Olivares et al., 2023; Wang et al., 2024b). Other methods more explicitly isolate or correct event-related variation through baseline decomposition, historical filtering, event-specific

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.

prediction, or residual correction (Taylor & Letham, 2018; Liang et al., 2024; Wolff et al., 2025; Meng et al., 2026). While these approaches demonstrate the value of modeling external effects, **historical event attribution and future event generation are typically handled through different mechanisms**, even though both are intended to represent responses to the same underlying event process. This separation leaves the two sides of the forecasting boundary only weakly connected. A natural alternative is to use a common response model across the boundary, so that explaining historical event-associated deviations and generating future responses become two uses of the same learned mechanism.

We therefore propose **Loki (Lift Out, Kick In)**, a general framework that models event responses consistently across the forecasting boundary. As illustrated in Figure 1(b), Loki **lifts out** estimated event-associated deviations from the historical window before temporal forecasting, and then **kicks in** responses associated with known future events. A shared **Event Response Function (ERF)** models these effects according to event type and relative temporal position, coupling historical attribution and future generation through the same learned response mechanism. By separating event-response modeling from temporal extrapolation, Loki can also be combined with different forecasting backbones.

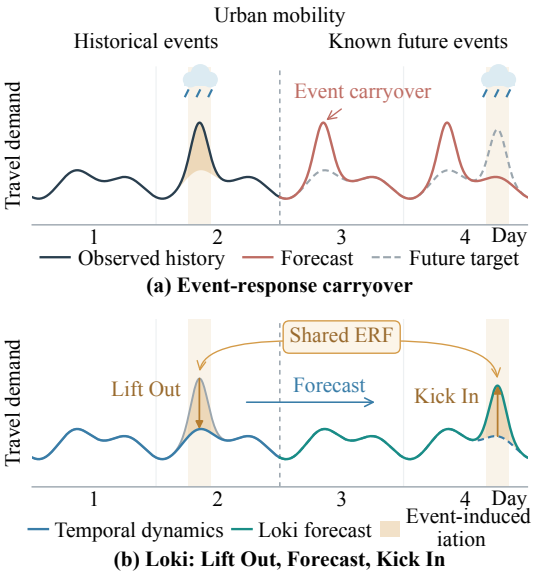


Figure 1: Event-response carryover and Loki's Lift Out–Forecast–Kick In mechanism.

We evaluate Loki across five event-aware forecasting benchmarks spanning mobility, traffic, retail, store sales, and bike demand, where it consistently outperforms strong forecasting baselines. Mechanism-oriented analyses further support the central premise of Loki: its gains arise from explicitly separating event responses from temporal dynamics rather than from event conditioning alone, the learned ERF recovers controlled event-induced variation, and the same formulation remains effective across different event regimes and forecasting backbones.

In summary, our contributions are as follows:

- **A forecasting perspective on event-aware time series modeling.** We formulate event-affected forecasting as an attribution problem: historical observations contain both temporal dynamics that may be extrapolated and event-induced variation associated with specific event contexts.
- **A shared event-response framework.** We propose Loki, whose shared ERF couples historical **Lift Out** and future **Kick In**, separating event-response modeling from temporal forecasting while learning both jointly.
- **Mechanism-oriented empirical validation.** We evaluate Loki across diverse forecasting tasks and examine not only overall forecasting performance, but also the structural necessity, response-recovery behavior, event-regime dependence, and backbone compatibility of the proposed formulation.

2 Related Work

2.1 Exogenous and Event-Conditioned Time Series Forecasting

Real-world time series are often influenced by external factors such as weather, holidays, promotions, and other contextual information, motivating extensive research on incorporating exogenous variables into forecasting models. Existing methods incorporate historical and known-future co-

variates through covariate conditioning, feature fusion, or explicit interaction mechanisms between endogenous and exogenous variables (Salinas et al., 2020; Lim et al., 2021; Das et al., 2023; Olivares et al., 2023; Wang et al., 2024b). However, conditioning on external information does not by itself explicitly separate event-induced deviations in history from temporal dynamics to be extrapolated. This motivates a more explicit separation between event-associated variation and the temporal dynamics to be extrapolated.

2.2 Decomposition and Event-Effect Modeling

Time-series forecasting has long benefited from decomposing observed variation according to distinct temporal structures, including trend and seasonality, frequency components, and stationary or non-stationary dynamics (Wu et al., 2021; Zhou et al., 2022; Liu et al., 2022; 2023; Wang et al., 2024a). Beyond intrinsic temporal structure, intervention analysis and Prophet explicitly represent intervention or holiday effects (Box & Tiao, 1975; Taylor & Letham, 2018), while promotion-aware forecasting separates baseline demand from promotion-driven variation (Wolters & Huchzermeier, 2021; Hewage et al., 2026). These works establish a long-standing precedent for modeling event-associated variation separately from the temporal dynamics to be extrapolated. Recent neural approaches pursue related ideas with learned event representations and specialized prediction mechanisms. LLM-MPE combines a non-event baseline decomposition of historical mobility with LLM-based reasoning over textual event information for future prediction (Liang et al., 2024). SPADE suppresses historical peak observations in the ordinary forecasting pathway and uses a dedicated Peak Attention module to predict future peak demand (Wolff et al., 2025). CEDAR learns an action-conditioned base rollout followed by a residual correction conditioned on external event signals (Meng et al., 2026).

These approaches demonstrate the value of explicitly modeling event-related variation. Loki differs in how the two sides of the forecasting boundary are parameterized: a shared learned Event Response Function (ERF) attributes and lifts out event-associated variation from historical observations, while the same response mechanism generates effects under future event configurations. Historical attribution and future generation are therefore coupled through a common event-response model rather than introduced as separate operations.

3 Method

We consider event-aware forecasting with an observed target history $X_{1:L} \in \mathbb{R}^{L\times F}$ and a future target $Y_{L+1:L+H} \in \mathbb{R}^{H\times F}$, where L and H denote the history and forecast lengths, respectively, and F is the number of target variables. The model is additionally given a known event schedule $E_{1:T} \in \mathbb{R}^{T\times K}$ over both windows, where $T = L + H$, and, when available, non-event contextual covariates $C_{1:T} \in \mathbb{R}^{T\times D}$.

Loki formulates event-aware forecasting as a decomposition of the target into an underlying trajectory and a structured event response:

$$Y_t = Y_t^0 \odot \exp(S_t), \tag{1}$$

where Y_t^0 denotes the trajectory to be modeled by the forecasting backbone and S_t denotes the event response in log space. This formulation provides a direct interpretation of the learned response in the original target space: positive values of S_t amplify the underlying trajectory, negative values suppress it, and $S_t = 0$ leaves the trajectory unchanged. Moreover, responses from multiple event sources can be combined additively in log space while corresponding to multiplicative effects on the original target scale.

To model event effects consistently across the forecasting boundary, Loki applies a shared Event Response Function (ERF) to the complete event schedule,

$$S_{1:T} = \mathrm{ERF}_\omega(E_{1:T}), \tag{2}$$

and partitions the resulting response into historical and future components, S^{hist} and S^{fut}.

As illustrated in Figure 2, Loki follows three stages: **Lift Out**, which removes the estimated historical event response S^{hist} from the observed history X to obtain the event-adjusted history X^0; **Forecast**, which predicts the underlying trajectory $\widehat{Y}^0$ from X^0 using a forecasting backbone; and

Kick In, which restores the future event response S^{fut} to obtain the final forecast $\widehat{Y}$. The same ERF generates both S^{hist} and S^{fut}, coupling historical event attribution and future event generation within a single forecasting framework.

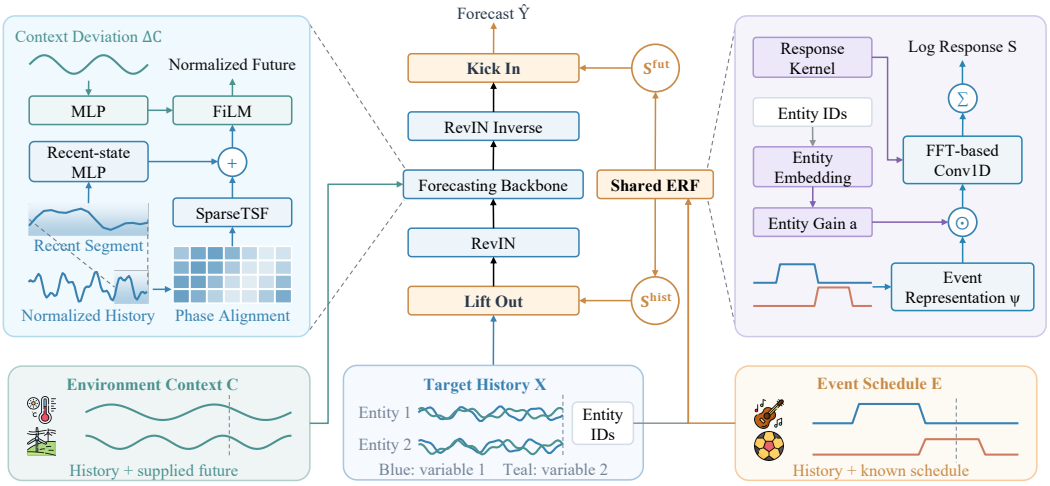


Figure 2: **Overall architecture of Loki.** The center shows the Lift Out–Forecast–Kick In pipeline, while the left and right panels detail the forecasting backbone and the shared Event Response Function (ERF), respectively.

3.1 Shared Event Response Function

The shared Event Response Function (ERF) models how a known event induces a structured response over time. For event channel k, forecasting unit i, target variable f, and time t, the response depends on the event's relative temporal position $r_{i,t,k}$, activity or intensity $d_{i,t,k}$, and forecasting-unit-specific characteristics.

We first represent the event state using a multi-scale relative-position encoding:

$$\psi_{i,t,k,m} = d_{i,t,k}\, \rho_m(r_{i,t,k}), \tag{3}$$

where m indexes the representation components and $\rho_m(\cdot)$ encodes the relative position within the event interval, as defined in Eq. (A.3). Loki then models the temporal response associated with each component using

$$g_{k,m,f}(\tau) = \sum_{j=1}^{J} W_{k,m,j,f}\, \beta_j(\tau), \tag{4}$$

where $\{\beta_j\}$ is a fixed set of multi-scale bidirectional temporal bases and W contains the learned combination weights. The basis set includes a zero-lag impulse, positive-lag exponential tails for persistent post-event effects, and negative-lag exponential tails for anticipatory effects. This bidirectional construction allows event influence to extend beyond the event interval itself: responses may emerge before an event and persist after it ends. Evaluating the same response structure across the complete history–forecast window therefore captures both anticipatory and carryover effects without introducing separate response models on the two sides of the forecasting boundary. This parameterization allows events of the same type to share a common response structure while learning target-specific temporal profiles.

The response induced by event channel k is

$$s_{i,t,k,f} = \sum_{m} a_{i,k,m}\, (g_{k,m,f} * \psi_{i,k,m})_t\,, \tag{5}$$

where $a_{i,k,m}$ provides forecasting-unit-specific modulation, as defined in Eq. (A.5). Responses from multiple event channels are aggregated additively in log space:

$$S_{i,t,f} = \sum_{k=1}^{K} s_{i,t,k,f}. \tag{6}$$

The shared ERF in Eq. (2) is evaluated over the complete observation-and-forecast window and then partitioned into S^{hist} and S^{fut}. Thus, historical event attribution and future event generation share the same response parameterization. Further details of the event representation, temporal response bases, forecasting-unit modulation, and cross-boundary computation are provided in Appendices A.1–A.3.

3.2 Lift Out, Forecast, and Kick In

Given the shared ERF, Loki applies Lift Out, forecasting, and Kick In through Eqs. (7)–(9), respectively. First, **Lift Out** removes the estimated historical event response from the observed history:

$$X^0 = X \odot \exp\left[-\operatorname{clip}\left(S^{\text{hist}}\right)\right], \tag{7}$$

where $\operatorname{clip}(\cdot)$ is used for numerical stability. This operation is intended to attribute event-associated variation to the ERF before temporal extrapolation rather than to perform generic smoothing or denoising. Window-level normalization (RevIN) (Kim et al., 2022) is then applied to the event-adjusted history before forecasting.

Next, the forecasting backbone models the remaining temporal dynamics:

$$\widehat{Y}^0 = B_\theta(X^0, C), \tag{8}$$

where C denotes non-event contextual covariates when available. Loki distinguishes bounded event variables, such as holidays, scheduled activities, and extreme-weather episodes, from continuously varying environmental variables such as temperature, precipitation, and price. The former are modeled by the ERF, whereas the latter are provided to the forecasting backbone as contextual inputs.

In our main implementation, B_θ is a lightweight backbone based on SparseTSF (Lin et al., 2024; 2026) with an additional recent-state branch. The SparseTSF branch uses an MLP to model cross-period phase relationships, while the recent-state branch uses an MLP to capture the most recent seasonal segment. The outputs of the two branches are added to form the baseline forecast. Environmental context is further incorporated through a lightweight FiLM-style modulation module (Perez et al., 2018). These components can be replaced by other forecasting architectures without changing the event-response formulation.

Finally, **Kick In** restores the response associated with known future events:

$$\widehat{Y} = \widehat{Y}^0 \odot \exp\left[\operatorname{clip}\left(S^{\text{fut}}\right)\right]. \tag{9}$$

Because S^{hist} and S^{fut} are generated by the same ERF in Eq. (2), historical event removal and future event generation are coupled through a shared response model rather than learned as independent corrections.

3.3 Learning

Loki is trained end-to-end using the final forecasting objective, so the forecasting backbone and both uses of the shared ERF are optimized jointly. Specifically, S^{hist} changes the event-adjusted trajectory presented to the forecasting backbone, whereas S^{fut} directly modifies the final prediction. The same forecasting objective therefore constrains both historical attribution and future event generation through the shared ERF. We use a scale-normalized mean absolute error:

$$\mathcal{L} = \frac{1}{BHF} \sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{f=1}^{F} \left| \frac{\widehat{Y}_{b,h,f} - Y_{b,h,f}}{\sigma_f^{\text{train}}} \right|, \tag{10}$$

where B denotes the batch size and σ_f^{train} denotes the training-set scale of target f.

At forecast positions where at least one event is active, gradients through the baseline forecasting path are stopped while gradients through the ERF branch remain active. This leaves the forward prediction unchanged while encouraging event-active deviations to be modeled through the event-response pathway rather than absorbed by the forecasting backbone.

We additionally use identity-preserving initialization: the event-response branch is initialized to produce zero response, so Lift Out and Kick In initially reduce to identity transformations. The

environmental modulation branch is initialized similarly, allowing optimization to start close to the underlying forecasting backbone and gradually learn the additional event-response structure. Further training and initialization details are provided in Appendix A.4.

4 EXPERIMENTS

Datasets. We evaluate Loki on five benchmarks spanning mobility, traffic, retail, store sales, and bike-rental demand: Ride-Hailing, TaxiBJ (Zhang et al., 2017), M5 (Makridakis et al., 2022), Rossmann (FlorianKnauer & Cukierski, 2015), and Seoul Bike (UCI Machine Learning Repository, 2020). These datasets vary in temporal resolution, number of entities and targets, available covariates, and forecasting horizons. Detailed data splits, window construction, covariate definitions, and preprocessing are provided in Appendix B.

Baselines. We compare Loki with eleven forecasting baselines: DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), SparseTSF (Lin et al., 2026), TimeMixer (Wang et al., 2024a), iTransformer (Liu et al., 2024), TimeXer (Wang et al., 2024b), CrossLinear (Zhou et al., 2025), XLinear (Chen et al., 2026), TiDE (Das et al., 2023), DAG (Qiu et al., 2026), and TFT (Lim et al., 2021). Models are evaluated under their supported information settings, including pure endogenous, historical-exogenous, and future-exogenous settings. Loki, TiDE, DAG, and TFT use future covariates without accessing future target values. We report standardized MAE and MSE, with scaling statistics computed solely from the training set and fixed across validation and test sets. Detailed protocols and configurations are provided in Appendices B.1 and C.2.

Implementation details. All experiments are implemented in Python 3.11 with PyTorch and conducted on eight NVIDIA RTX A6000 GPUs. All models are optimized with AdamW and selected based on validation MAE, using model-specific optimization configurations rather than a unified setting. Training and implementation details are provided in Appendix C.

4.1 MAIN RESULTS

Table 1: Forecasting performance across five datasets. The best results are highlighted in **bold** and the second best are underlined.

Dataset	Ride-Hailing		TaxiBJ		M5		Rossmann		Seoul Bike	
Metrics	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
DLinear	0.1890	0.2271	0.1264	0.0563	0.1068	0.0309	0.3352	0.2702	0.6722	0.8312
PatchTST	<u>0.1618</u>	0.2171	0.1323	0.0698	0.1022	0.0298	0.2482	0.2069	0.5343	0.7825
SparseTSF	0.1804	0.2265	0.1357	0.0645	0.1026	0.0298	0.2753	0.2439	0.5616	0.6540
TimeMixer	0.1653	0.2212	0.1311	0.0677	0.1025	0.0304	0.2490	0.2019	0.5047	0.6679
iTransformer	0.1670	0.2208	0.1310	0.0669	0.1037	0.0314	0.2507	0.2028	0.5547	0.7451
TFT	0.1692	0.2113	<u>0.1191</u>	<u>0.0538</u>	0.1067	0.0334	0.1591	0.0669	<u>0.3310</u>	**0.2076**
TiDE	0.1625	<u>0.2008</u>	0.1333	0.0597	0.1080	0.0309	0.1597	0.0600	0.6079	0.7618
TimeXer	0.1713	0.2232	0.1324	0.0612	0.1068	0.0321	0.2529	0.2018	0.5204	0.6860
CrossLinear	0.1649	0.2222	0.1276	0.0619	0.1023	<u>0.0294</u>	0.2551	0.2037	0.5050	0.7205
XLinear	0.1659	0.2193	0.1391	0.0723	<u>0.1019</u>	0.0304	0.2518	0.1973	0.5097	0.7058
DAG	0.1655	0.2198	0.1321	0.0641	0.1083	0.0338	<u>0.1551</u>	**0.0551**	0.4647	0.5499
Loki	**0.1573**	**0.1996**	**0.1120**	**0.0469**	**0.0961**	**0.0258**	**0.1508**	<u>0.0573</u>	**0.3122**	<u>0.2463</u>

Table 1 compares Loki against eleven strong forecasting baselines across five heterogeneous benchmarks spanning mobility, traffic, retail, and bike-demand forecasting. Loki achieves the best standardized MAE on all five datasets, consistently outperforming the strongest non-Loki baseline by approximately 2.8%–6.0%. The gains extend beyond MAE: under standardized MSE, Loki ranks first on Ride-Hailing, TaxiBJ, and M5, and second on Rossmann and Seoul Bike, while attaining the best average rank across all compared methods. Importantly, Loki remains competitive with TiDE, DAG, and TFT under the same future-covariate setting, indicating that its gains are not attributable to an easier information regime.

The consistent improvements across datasets and evaluation metrics suggest that Loki's effectiveness generalizes across diverse forecasting domains rather than being driven by a particular benchmark

or metric. On the original data scale, Loki also achieves the lowest WMAPE on all five datasets (Table 9), further supporting its predictive advantage. Appendix D.1 reports the complete RMSE and WMAPE results.

4.2 Ablation Study

As shown in Table 2, Loki consistently outperforms the parameter-matched **Direct Fusion** variant, which uses the same event information but injects it directly into the forecasting pathway without explicit Lift Out/Kick In decomposition. The improvement holds across both datasets and metrics and is particularly substantial on M5, with reductions of 6.52% in MAE and 15.13% in MSE. This controlled comparison suggests that the gains arise from the proposed decomposition rather than event conditioning alone.

Table 2: Ablation study of Loki.

Dataset	Ride-Hailing		M5	
Metrics	MAE	MSE	MAE	MSE
Only Backbone	0.1635	0.2163	0.1033	0.0307
Direct Fusion	0.1607	0.2125	0.1028	0.0304
History-only	0.1634	0.2200	0.1144	0.0403
Future-only	0.1578	0.2005	0.1009	0.0298
Independent ERF	0.1582	0.1997	0.1016	0.0277
Full Loki	**0.1573**	**0.1996**	**0.0961**	**0.0258**

Loki also consistently outperforms both one-sided variants, **History-only** and **Future-only**, which retain only Lift Out and Kick In, respectively. On M5, Loki improves over the stronger one-sided variant by 4.76% in MAE and 13.42% in MSE, demonstrating that jointly modeling historical and future event effects is important for capturing the full response pattern.

Loki outperforms **Independent ERF**, which replaces the shared ERF with two architecturally identical but independently parameterized response functions for the historical and forecast windows. Despite introducing additional response parameters, Independent ERF performs consistently worse across both datasets and metrics. On M5, the shared ERF yields 5.41% lower MAE and 6.86% lower MSE, supporting the effectiveness of parameter sharing for learning a unified response function across historical and future windows.

4.3 Further Analyses

4.3.1 Controlled Event-Effect Recovery

While the forecasting results show that Loki improves predictive accuracy, they do not directly reveal whether the learned ERF actually recovers the event-associated variation that the framework is designed to separate from underlying temporal dynamics. This is difficult to verify on standard forecasting benchmarks because event effects are generally unobserved. We therefore construct controlled event-injected versions of Ride-Hailing and M5. Given the original reference series Y^{ref}, we construct

$$Y = Y^{\mathrm{ref}} \odot \exp(S^*), \tag{11}$$

where S^* is a known injected event response in log space. The original series otherwise remains unchanged, preserving its native temporal dynamics and noise. The model observes the corresponding artificial event descriptors, but neither S^* nor its temporal profile is provided as input or supervision.

We evaluate three conditions. **Full** uses both the perturbed series and the artificial event descriptors. **Event-zeroed** uses the same perturbed series but removes these descriptors, testing whether the event-response mechanism can recover the injected component from event information. A **null control** retains the descriptors but injects no corresponding effect, allowing us to measure spurious attribution. Full construction, control settings, and evaluation metrics are provided in Appendix E.

As shown in Figure 3, Loki closely recovers both the magnitude and temporal structure of the injected responses. Compared with the event-zeroed control, Loki reduces event-effect MAE by 86.71%–87.69% and history-restoration MAE by 93.26%–93.72% across the two datasets, while the recovered response profiles achieve cosine similarities of 0.9966–0.9989. The null control produces only small spurious responses when no effect is injected. Together, these results support the intended interpretation of the ERF as a predictive attribution of event-associated variation rather than a causal effect estimator. Full quantitative results are reported in Appendix E.4.

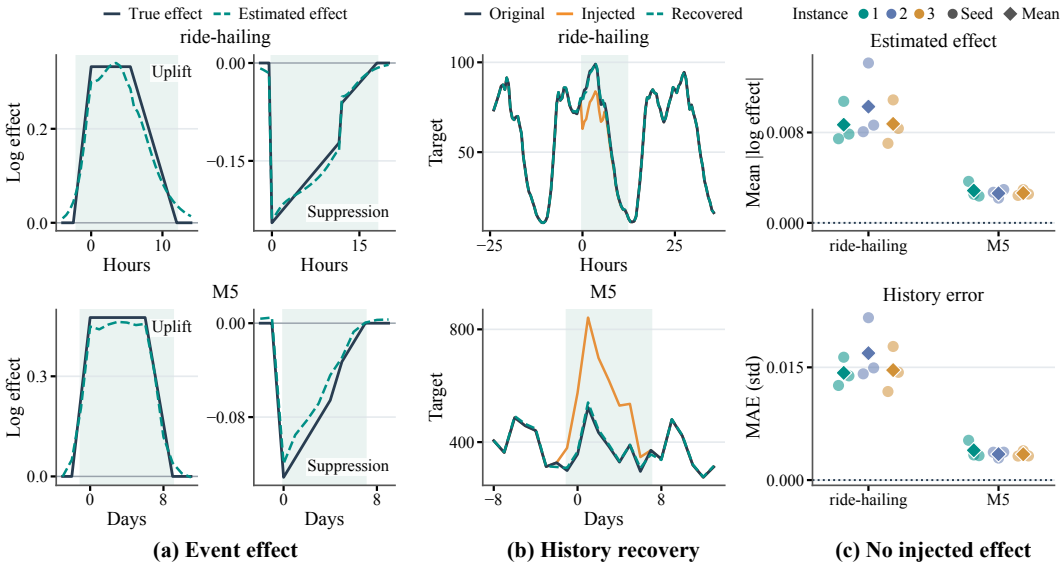


Figure 3: Controlled event-effect recovery on event-injected Ride-Hailing and M5 data. (a) Injected log effects and Loki estimates for uplift and suppression examples. (b) Reference, event-injected, and Lift Out–recovered histories. (c) Null-control results with event descriptors retained but no effect injected.

4.3.2 Event-Regime Analysis

We partition test windows according to whether active discrete-event indicators appear in the historical and forecast windows. Quiet (Q) denotes no active indicator, while Event (E) denotes at least one, yielding four history-to-forecast regimes: Q→Q, E→Q, Q→E, and E→E. As shown in Figure 4(a), Loki's relative MAE reduction becomes larger as event involvement increases, rising from 1.59% in Q→Q to 2.34% in E→Q, 3.88% in Q→E, and 5.60% in E→E. This pattern suggests that Loki's gains are not uniform across samples, but become more pronounced in regimes where event-aware modeling is more relevant.

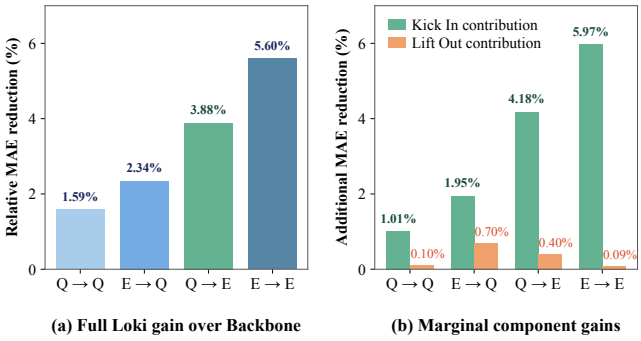


Figure 4: **Event-regime analysis on Ride-hailing.** Regimes are denoted as history→forecast, with Q and E indicating quiet and event-active windows. (a) Relative MAE reduction of Full Loki over the Backbone. (b) Marginal Kick In and Lift Out contributions, measured against History-only and Future-only, respectively.

Figure 4(b) further separates the contributions of the two components. Kick In, measured by Loki's additional gain over History-only, contributes substantially more when the forecast window is event-active, reaching 4.18% in Q→E and 5.97% in E→E, compared with 1.01% in Q→Q and 1.95% in E→Q. This is consistent with its role in modeling responses associated with future events. Lift Out, measured relative to Future-only, has its largest incremental contribution in E→Q (0.70%), where event-affected history transitions into an event-quiet future and historical event-related variation is less directly transferable to the forecast horizon. Overall, the regime-dependent patterns align with the intended division of roles between Kick In and Lift Out across the two sides of the forecasting boundary.

4.3.3 BACKBONE COMPATIBILITY

Loki decouples event-response modeling from temporal forecasting, allowing its Event Response Function to plug into different forecasting backbones. To evaluate whether the proposed event-response mechanism depends on the default forecasting architecture, we instantiate Loki with three representative backbones: SparseTSF, PatchTST, and TimeMixer. These models differ substantially in their temporal modeling strategies, providing a controlled test of whether Loki's gains persist beyond a particular backbone design. We evaluate all three variants on Ride-Hailing and M5.

Table 3: Compatibility of Loki with different forecasting backbones.

Dataset	Ride-Hailing		M5	
Metrics	MAE	MSE	MAE	MSE
PatchTST	0.1618	0.2171	0.1022	0.0298
+Loki	**0.1583**	**0.2002**	**0.0985**	**0.0262**
Improve	2.12%	7.78%	3.62%	12.08%
SparseTSF	0.1804	0.2265	0.1026	0.0298
+Loki	**0.1708**	**0.2043**	**0.0988**	**0.0268**
Improve	5.32%	9.80%	3.68%	10.07%
TimeMixer	0.1653	0.2212	0.1025	0.0304
+Loki	**0.1586**	**0.2005**	**0.1009**	**0.0274**
Improve	4.08%	9.36%	1.61%	9.87%

As shown in Table 3, Loki consistently improves all six backbone–dataset combinations. Standardized MAE decreases by 2.12%–5.32% on Ride-Hailing and 1.61%–3.68% on M5, while MSE is reduced by 7.78%–9.80% and 9.87%–12.08%, respectively. The gains persist across diverse temporal forecasting architectures, indicating that Loki's event-response modeling is complementary to the forecasting backbone rather than tied to a specific implementation.

5 LIMITATIONS

Loki assumes that relevant event descriptors are available for both historical and forecast windows. While suitable for scheduled or externally observed events, this assumption does not cover latent, unrecorded, or previously unseen events whose occurrence or semantics are unavailable to the model. The shared Event Response Function (ERF) further assumes that response patterns learned from historical events remain informative for future ones, which may be less appropriate under substantial shifts in event effects, unseen event types, or changing entity-specific responses. Extending the response model to adapt to distribution shifts or previously unseen events is an important direction for more non-stationary settings.

The learned ERF should be interpreted as a predictive decomposition of event-associated variation rather than a causal effect estimator. In observational data, the underlying dynamics and event-induced component are generally not separately identifiable without additional assumptions or interventions. Controlled or semi-synthetic experiments can evaluate recovery of known injected event components by construction, but such recovery does not establish causal identifiability in real-world settings.

6 CONCLUSION

In this paper, we introduced Loki, a general framework for event-aware time series forecasting that separates event-induced variation from the temporal dynamics to be extrapolated. Loki uses a shared Event Response Function across the forecasting boundary: event-related deviations are lifted out from historical observations before temporal forecasting, while the same response mechanism generates and kicks in the effects associated with known future events. This formulation couples historical attribution and future generation while keeping event-response modeling distinct from the forecasting backbone. Experiments show that this separation improves forecasting performance, and mechanism-oriented analyses consistently support the intended Lift Out/Kick In interpretation. The learned ERF recovers controlled event responses, Loki provides larger gains in event-relevant regimes, and the same mechanism remains effective across different forecasting backbones. Overall, our results suggest that event effects are better treated as a shared response process across history and future, rather than being absorbed into the underlying temporal dynamics or modeled separately on the two sides of the forecasting boundary.

AI USE STATEMENT

Generative AI tools were used in this work for the following tasks: literature exploration and summarization, refinement of the conceptual framing, feedback on research methodology and experimental design, interpretation of experimental results, manuscript structuring, and drafting and editing portions of the paper for clarity and readability.

Generative AI tools were not used for the following disclosure-relevant tasks: generating synthetic datasets, implementing the proposed method, formulating mathematical claims, or developing or verifying formal proofs.

The authors reviewed and verified all AI-assisted outputs by checking cited references against the original sources, independently reviewing methodological and experimental suggestions, validating all reported experimental results against the underlying experiment outputs, and manually reviewing and revising all manuscript text. The authors take full responsibility for all text, claims, code, results, and other artifacts contained in this work.

REPRODUCIBILITY STATEMENT

The main paper specifies Loki's architecture, learning objective, evaluation setting, and computational environment. Appendix B documents the dataset construction, chronological splits, forecasting windows, covariates, and preprocessing. Appendix A provides detailed model and learning specifications, while Appendix C provides training and implementation details. Appendix D defines the additional metrics and uncertainty reporting. The controlled event-effect recovery study further reports its generation procedure, experimental conditions, evaluation metrics, and random seeds in Appendix E. Together, these materials specify the data pipeline, training protocol, and evaluation procedures used in the reported experiments.

REFERENCES

George E. P. Box and George C. Tiao. Intervention analysis with applications to economic and environmental problems. *Journal of the American Statistical Association*, 70(349):70–79, 1975. doi: 10.1080/01621459.1975.10480264. URL https://doi.org/10.1080/01621459.1975.10480264.

Xinyang Chen, Huidong Jin, Yu Huang, and Zaiwen Feng. XLinear: A lightweight and accurate MLP-based model for long-term time series forecasting with exogenous inputs. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence*, volume 40, pp. 20325–20335, 2026. doi: 10.1609/aaai.v40i24.39121. URL https://doi.org/10.1609/aaai.v40i24.39121.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with TiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=pCbC3aQB5W.

FlorianKnauer and Will Cukierski. Rossmann store sales. Kaggle competition dataset, 2015. URL https://www.kaggle.com/competitions/rossmann-store-sales. Kaggle.

Harsha Chamara Hewage, H. Niles Perera, and Kasun Bandara. Enhancing demand forecasting in retail: A comprehensive analysis of sales promotional effects on the entire demand life cycle. *Journal of Forecasting*, 45(1):293–315, 2026. doi: 10.1002/for.70039. URL https://doi.org/10.1002/for.70039.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Yuebing Liang, Yichao Liu, Xiaohan Wang, and Zhan Zhao. Exploring large language models for human mobility prediction under public events. *Computers, Environment and Urban Systems*,

112:102153, 2024. doi: 10.1016/j.compenvurbsys.2024.102153. URL https://doi.org/10.1016/j.compenvurbsys.2024.102153.

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. doi: 10.1016/j.ijforecast.2021.03.012. URL https://doi.org/10.1016/j.ijforecast.2021.03.012.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with 1k parameters. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30211–30226. PMLR, 2024. URL https://proceedings.mlr.press/v235/lin24n.html.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and C. L. Philip Chen. SparseTSF: Lightweight and robust time series forecasting via sparse modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(1):170–183, 2026. doi: 10.1109/TPAMI.2025.3602445. URL https://doi.org/10.1109/TPAMI.2025.3602445.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4054556fcaa934b0bf76da52cf4f92cb-Abstract-Conference.html.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/28b3dc0970fa4624a63278a4268de997-Abstract-Conference.html.

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 *International Conference on Learning Representations*, volume 2024, pp. 11116–11140, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2ea18fdc667e0ef2ad82b2b4d65147ad-Abstract-Conference.html.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364, 2022. doi: 10.1016/j.ijforecast.2021.11.013. URL https://doi.org/10.1016/j.ijforecast.2021.11.013.

Junjie Meng, Ranxu Zhang, Zi-an Zhang, Shujun Liu, Xiaoning Qi, Xiaozhou Xu, Yanyong Zhang, Hui Xiong, and Chao Wang. CEDAR: Controlled and event-driven demand forecasting via residual decomposition. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 7797–7808. ACM, August 2026. doi: 10.1145/3770855.3818338. URL https://doi.org/10.1145/3770855.3818338.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Kin G. Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39(2):884–900, 2023. doi: 10.1016/j.ijforecast.2022.03.001. URL https://doi.org/10.1016/j.ijforecast.2022.03.001.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018. doi: 10.1609/aaai.v32i1.11671. URL https://doi.org/10.1609/aaai.v32i1.11671.

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. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.14933.

Filipe Rodrigues, Ioulia Markou, and Francisco C. Pereira. Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. *Information Fusion*, 49: 120–129, 2019. doi: 10.1016/j.inffus.2018.07.007. URL https://doi.org/10.1016/j.inffus.2018.07.007.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001. URL https://doi.org/10.1016/j.ijforecast.2019.07.001.

Sean J. Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1): 37–45, 2018. doi: 10.1080/00031305.2017.1380080. URL https://doi.org/10.1080/00031305.2017.1380080.

UCI Machine Learning Repository. Seoul bike sharing demand. Dataset, 2020. URL https://doi.org/10.24432/C5F62R.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, volume 2024, pp. 38626–38652, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/a7ac8a21e5a27e7ab31a5f42a0117bdb-Abstract-Conference.html.

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 *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/hash/0113ef4642264adc2e6924a3cbbdf532-Abstract-Conference.html.

Malcolm Wolff, Kin G. Olivares, Boris Oreshkin, Sunny Ruan, Sitan Yang, Abhinav Katoch, Shankar Ramasubramanian, Youxin Zhang, Michael W. Mahoney, Dmitry Efimov, and Vincent Quenneville-Bélair. ♠ SPADE ♠ split peak attention DEcomposition. arXiv preprint arXiv:2411.05852, 2025. URL https://arxiv.org/abs/2411.05852v2.

Jannik Wolters and Arnd Huchzermeier. Joint in-season and out-of-season promotion demand forecasting in a retail environment. *Journal of Retailing*, 97(4):726–745, 2021. doi: 10.1016/j.jretai.2021.01.003. URL https://doi.org/10.1016/j.jretai.2021.01.003.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://doi.org/10.1609/aaai.v37i9.26317.

Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1655–1661, 2017. doi: 10.1609/aaai.v31i1.10735. URL https://doi.org/10.1609/aaai.v31i1.10735.

Pengfei Zhou, Yunlong Liu, Junli Liang, Qi Song, and Xiangyang Li. CrossLinear: Plug-and-play cross-correlation embedding for time series forecasting with exogenous variables. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp.

4120–4131. Association for Computing Machinery, 2025. doi: 10.1145/3711896.3736899. URL https://doi.org/10.1145/3711896.3736899.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

A Additional Method Details

A.1 Event Representation

Loki represents each event through three complementary aspects: its **sustained activity**, its **proximity to the event onset**, and its **proximity to the event end**. Historical and future event covariates are concatenated and processed jointly over the complete $L+H$ window before the response is split into historical and future components.

Let $e_{i,t,k}$ denote the value of event channel k for forecasting unit i in the event schedule E at time t. For continuous or flag-valued event channels, nonzero values are first normalized using the training-set nonzero RMS statistic $\mathrm{RMS}_k^{\mathrm{train}}$,

$$\bar{e}_{i,t,k} = \frac{e_{i,t,k}}{\mathrm{RMS}_k^{\mathrm{train}}}. \tag{A.1}$$

The activity or intensity variable $d_{i,t,k}$ used in the main text is then obtained through a learnable event-type-specific power transform,

$$d_{i,t,k} = \mathrm{sign}(\bar{e}_{i,t,k})|\bar{e}_{i,t,k}|^{\gamma_k}, \qquad \gamma_k = \exp(\eta_k). \tag{A.2}$$

Consecutive nonzero values in the same event channel are treated as one event interval. Let $p_{i,t,k}$ and $q_{i,t,k}$ denote the distances from the beginning and end of that interval, respectively, and let $v^p_{i,t,k}$ and $v^q_{i,t,k}$ indicate whether the corresponding boundary is observable within the current window. These quantities instantiate the relative-position state $r_{i,t,k}$ introduced in the main text. We define its multi-scale encoding directly as

$$\rho(r_{i,t,k}) = \left[1,\ \left\{e^{-p_{i,t,k}/s} v^p_{i,t,k}\right\}_{s\in\mathcal{S}_{\mathrm{pos}}},\ \left\{e^{-q_{i,t,k}/s} v^q_{i,t,k}\right\}_{s\in\mathcal{S}_{\mathrm{pos}}}\right], \tag{A.3}$$

where

$$\mathcal{S}_{\mathrm{pos}} = \mathrm{unique}\left\{\text{steps-per-day}, \text{season-length}\right\}.$$

Here $\rho_m(r_{i,t,k})$ denotes the m-th component of the vector $\rho(r_{i,t,k})$. The final event representation is therefore

$$\psi_{i,t,k,m} = d_{i,t,k}\rho_m(r_{i,t,k}).$$

The position encoder itself describes the event interval; anticipatory and post-event effects are produced by the bidirectional temporal response function below.

A.2 Shared Event Response Function

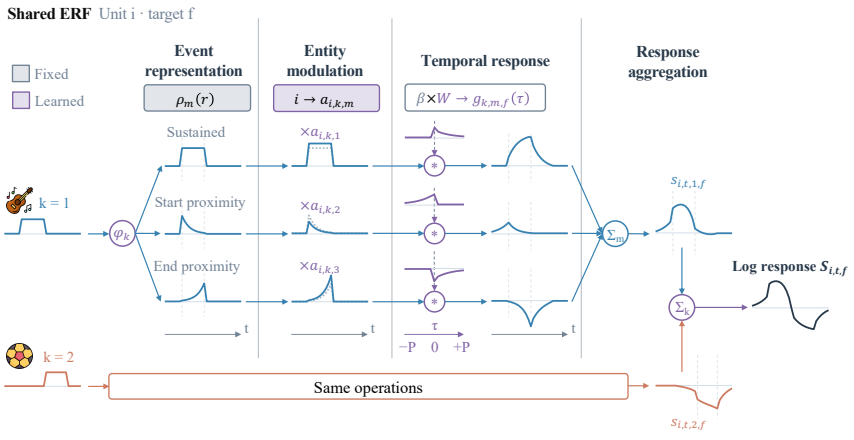


Figure 5: Computation of the shared Event Response Function (ERF).

Figure 5 summarizes the computation of the shared ERF from event representation through response aggregation. The ERF uses a compact fixed basis bank to represent **instantaneous**, **anticipatory**,

and **post-event carryover** event responses. For response support P, the basis contains a zero-lag impulse,

$$\beta_0(\tau) = \mathbf{1}[\tau = 0],$$

together with positive- and negative-lag exponential tails,

$$\beta_s^+(\tau) = \frac{\mathbf{1}[\tau > 0]e^{-\tau/s}}{\sum_{r=1}^{P} e^{-r/s}}, \qquad \beta_s^-(\tau) = \frac{\mathbf{1}[\tau < 0]e^{\tau/s}}{\sum_{r=1}^{P} e^{-r/s}}, \tag{A.4}$$

for exponentially spaced scales

$$s \in \{1, 2, 4, \ldots \leq P\}.$$

In the reported experiments, P is set to the dataset seasonal length.

The response kernel is parameterized as

$$g_{k,m,f}(\tau) = \sum_{j=1}^{J} W_{k,m,j,f}\beta_j(\tau),$$

where k, m, j, and f index event type, relative-position component, temporal basis, and target variable, respectively.

To allow forecasting-unit heterogeneity while retaining a shared response structure, Loki applies a positive low-rank modulation,

$$a_{i,k,m} = \exp\left(\mathrm{clip}_{[-2,2]}\left(\langle V_i, U_{k,m}\rangle\right)\right). \tag{A.5}$$

where $V_i \in \mathbb{R}^R$ is a learnable forecasting-unit embedding and $U_{k,m} \in \mathbb{R}^R$ is a learnable vector for event type k and representation channel m, shared across forecasting units.

The response associated with event type k is then

$$s_{i,t,k,f} = \sum_m a_{i,k,m}\left(g_{k,m,f} * \psi_{i,k,m}\right)_t,$$

and the total log response is obtained by summing across event types,

$$S_{i,t,f} = \sum_k s_{i,t,k,f}.$$

Thus, event types share the same response formulation while retaining event-specific, target-specific, and forecasting-unit-specific structure.

A.3 Cross-Boundary Computation

The complete history-and-forecast interval is treated as a single response domain. The ERF is evaluated once on the full event schedule and only then partitioned into

$$S^{\mathrm{hist}} = S_{1:L}, \qquad S^{\mathrm{fut}} = S_{L+1:L+H}.$$

The forecasting boundary is therefore *not* a response boundary: a historical event may have a persistent effect extending into the forecast horizon, while a future event may produce an anticipatory response extending into the historical window. Only the external boundaries of the complete $L + H$ window truncate the response.

For efficiency, the bidirectional response is implemented using zero-padded FFT-based linear convolution.

Lift Out and Kick In are both applied in the original target space. Writing the clipping operator in the main text explicitly as $\mathrm{clip}_{[-8,8]}$, we have

$$X^0 = X \odot \exp\left[-\mathrm{clip}_{[-8,8]}\left(S^{\mathrm{hist}}\right)\right],$$

and

$$\widehat{Y} = \widehat{Y}^0 \odot \exp\left[\mathrm{clip}_{[-8,8]}\left(S^{\mathrm{fut}}\right)\right].$$

Window-level RevIN is applied only between these two operations: after Lift Out and before forecasting, and inverted before Kick In.

A.4 Forecasting and Learning Details

The default temporal forecaster follows a lightweight SparseTSF-style design and is augmented with a recent-state branch that encodes the most recent seasonal segment. The two forecasting branches are combined additively.

For datasets with continuously varying environmental covariates, Loki uses their future deviation from the historical mean,

$$\Delta C_t = C_t - \frac{1}{L}\sum_{\ell=1}^{L} C_\ell, \tag{A.6}$$

where t indexes a forecast position. Let χ_t denote the corresponding calendar/time representation and let z_t denote the backbone latent state at that position. A lightweight network h_ϕ produces the FiLM parameters through zero-reference subtraction,

$$[\lambda_t, \delta_t] = h_\phi([\Delta C_t, \chi_t]) - h_\phi([0, \chi_t]), \tag{A.7}$$

$$z'_t = (1 + \lambda_t) \odot z_t + \delta_t. \tag{A.8}$$

The reference subtraction ensures zero modulation when the future environment matches the historical reference state.

Loki is trained only through the final scale-normalized forecasting loss,

$$\mathcal{L} = \frac{1}{BHF}\sum_{b=1}^{B}\sum_{h=1}^{H}\sum_{f=1}^{F}\left|\frac{\widehat{Y}_{b,h,f} - Y_{b,h,f}}{\sigma_f^{\text{train}}}\right|,$$

with no direct supervision on the event response or decomposition components.

At forecast positions where at least one event is active, gradients through the baseline forecasting path are stopped while gradients through the ERF remain active. This modifies only the backward computation and leaves the forward prediction unchanged.

Finally, the ERF is initialized to produce the identity multiplicative effect. The temporal response weights are initialized to zero, yielding $S = 0$ and hence $\exp(S) = 1$; forecasting-unit modulation is initialized to one; the event-intensity power is initialized to $\gamma_k = 1$; and the environmental FiLM branch is initialized to zero modulation.

B Dataset and Forecasting Protocol Details

B.1 Forecasting Protocol

We construct rolling-origin samples under fixed chronological splits. Forecast horizons lie entirely within one split, while historical inputs may precede its boundary. Table 4 summarizes the statistics of the five benchmark datasets. Table 5 summarizes the window configurations and sample counts.

TaxiBJ and Rossmann contain temporal discontinuities. We require the complete $L+H$ span of each sample to be temporally contiguous and discard any window crossing a gap. No temporal padding or missing-timestamp interpolation is used.

We evaluate future-exogenous models under a **known-future covariate** setting: exogenous variables supplied by each benchmark protocol are treated as available throughout the forecast horizon, whereas future target values are never provided as model inputs.

B.2 Covariates and Dataset Construction

Table 6 summarizes the dataset- specific assignment of covariates to modeling roles based on semantics rather than a universal variable-type rule. For ERF inputs, categorical events are expanded into event-type channels, while binary and continuous values encode activity and intensity, respectively.

We retain only preprocessing steps that change the forecasting problem definition. TaxiBJ windows crossing temporal gaps are removed. M5 is evaluated at the store–department level using Level-9 aggregated sales, with corresponding price, promotion, SNAP, and calendar covariates. For Rossmann,

Table 4: Statistics of the five benchmark datasets. Observations denote entity–time rows.

Dataset	Ride-hailing	TaxiBJ	M5	Rossmann	Seoul Bike
Resolution	30 min	30 min	Daily	Daily	Hourly
# Observations	14,025,600	21,872,640	137,829	1,017,209	8,760
# Entities	200	1,024	70	1,115	1
# Targets	9	2	1	1	1

Table 5: Forecasting protocol across datasets. History and forecast lengths are reported in time steps. Sample counts are forecasting windows after split and continuity filtering.

Dataset	History L	Forecast H	Stride	Samples Train / Val / Test
Ride-hailing	1344	336	48	212,200 / 35,200 / 35,600
TaxiBJ	336	48	48	150,528 / 29,696 / 20,480
M5	364	28	1	106,539 / 70 / 70
Rossmann	364	28	1	275,599 / 62,645 / 151,470
Seoul Bike	168	24	24	248 / 36 / 73

Table 6: High-level covariate roles used by Loki across datasets.

Dataset	ERF channels	Backbone context	Other known inputs
Ride-hailing	event and calendar signals	–	calendar, entity
TaxiBJ	holidays	temperature, wind, weather	calendar, grid
M5	SNAP, calendar events	price, promotion aggregates	calendar, entity
Rossmann	promotion, holidays	–	open status, calendar, store
Seoul Bike	holiday	weather variables	operating status, calendar

unobserved sales records are excluded while closed-store periods are retained; same-day customer counts are not used as model inputs. Seoul Bike retains non-functioning periods and provides operating status as auxiliary information.

C Experimental Implementation Details

C.1 Training and Model Selection

Models are validated each epoch, with model-specific early-stopping patience. The checkpoint minimizing normalized validation MAE is used for final evaluation. DAG adds a causality auxiliary loss with coefficient $\beta = 0.1$; all other models use only the primary forecasting objective.

C.2 Baseline Implementations and Configurations

Most baselines use official implementations integrated into a common training and evaluation pipeline through lightweight adapters. TiDE and TFT use PyTorch reimplementations or adaptations based on the cited original implementations. Table 7 summarizes the implementation sources and key configurations.

For TimeXer, the patch length is 24 for Ride-hailing, TaxiBJ, and Seoul Bike, and 7 for M5 and Rossmann.

Table 7: Baseline implementation sources and key model configurations.

Model	Implementation source	Key configuration
DLinear	Official implementation	individual = false
PatchTST	Official implementation	patch = 48, stride = 48, $d_{\text{model}} = 128$, 8 heads, 3 layers
SparseTSF	Official implementation	linear variant; period set according to dataset periodicity
TimeMixer	Official implementation	$d_{\text{model}} = 32$, $d_{\text{ff}} = 64$, 2 layers, 2 downsampling layers
iTransformer	Official implementation	$d_{\text{model}} = 128$, 8 heads, 2 layers
TimeXer	Official implementation	$d_{\text{model}} = 128$, 8 heads, 2 layers; dataset-dependent patch length
CrossLinear	Official implementation	patch = 24, $d_{\text{model}} = 128$, $d_{\text{ff}} = 256$
XLinear	Official implementation	$d_{\text{model}} = 1024$, temporal FF = 256
TiDE	PyTorch reimplementation	hidden = 256, 2 layers, series embedding = 16
DAG	Official implementation	patch = 48, stride = 48, $d_{\text{model}} = 64$, 4 heads, 1 layer
TFT	PyTorch adaptation	hidden = 64, 4 heads, dropout = 0.1

Table 8: Standardized RMSE across five datasets. Results are reported as mean ± sample standard deviation across seeds. The best results are highlighted in **bold** and the second best are underlined.

Model	Ride-hailing	TaxiBJ	M5	Rossmann	Seoul Bike
DLinear	0.4766 ± 0.0007	0.2372 ± 0.0009	0.1757 ± 0.0051	0.5198 ± 0.0007	0.9116 ± 0.0167
PatchTST	0.4660 ± 0.0004	0.2642 ± 0.0065	0.1725 ± 0.0036	0.4548 ± 0.0020	0.8846 ± 0.0087
SparseTSF	0.4759 ± 0.0004	0.2538 ± 0.0077	0.1725 ± 0.0034	0.4939 ± 0.0008	0.8085 ± 0.0202
TimeMixer	0.4703 ± 0.0015	0.2602 ± 0.0055	0.1742 ± 0.0089	0.4493 ± 0.0067	0.8172 ± 0.0091
iTransformer	0.4698 ± 0.0013	0.2585 ± 0.0090	0.1770 ± 0.0075	0.4503 ± 0.0014	0.8631 ± 0.0077
TFT	0.4596 ± 0.0056	0.2317 ± 0.0110	0.1825 ± 0.0128	0.2586 ± 0.0079	$\mathbf{0.4548 \pm 0.0315}$
TiDE	0.4481 ± 0.0023	0.2443 ± 0.0062	0.1758 ± 0.0069	0.2449 ± 0.0037	0.8669 ± 0.1137
TimeXer	0.4725 ± 0.0007	0.2473 ± 0.0106	0.1788 ± 0.0112	0.4492 ± 0.0062	0.8282 ± 0.0073
CrossLinear	0.4714 ± 0.0016	0.2488 ± 0.0046	0.1711 ± 0.0138	0.4513 ± 0.0012	0.8487 ± 0.0125
XLinear	0.4683 ± 0.0009	0.2688 ± 0.0090	0.1741 ± 0.0065	0.4442 ± 0.0062	0.8401 ± 0.0020
DAG	0.4689 ± 0.0009	0.2529 ± 0.0105	0.1837 ± 0.0088	$\mathbf{0.2347 \pm 0.0027}$	0.7410 ± 0.0317
Loki	$\mathbf{0.4468 \pm 0.0010}$	$\mathbf{0.2166 \pm 0.0034}$	$\mathbf{0.1605 \pm 0.0027}$	0.2392 ± 0.0088	0.4961 ± 0.0172

Table 9: Original-scale WMAPE across five datasets. Results are reported as mean ± sample standard deviation across seeds. The best results are highlighted in **bold** and the second best are underlined.

Model	Ride-hailing	TaxiBJ	M5	Rossmann	Seoul Bike
DLinear	0.1258 ± 0.0010	0.1796 ± 0.0007	0.1027 ± 0.0026	0.2145 ± 0.0003	0.5647 ± 0.0142
PatchTST	0.1083 ± 0.0003	0.1880 ± 0.0033	0.0982 ± 0.0013	0.1588 ± 0.0023	0.4488 ± 0.0041
SparseTSF	0.1216 ± 0.0002	0.1928 ± 0.0094	0.0986 ± 0.0021	0.1762 ± 0.0006	0.4718 ± 0.0185
TimeMixer	0.1110 ± 0.0002	0.1862 ± 0.0031	0.0985 ± 0.0040	0.1594 ± 0.0041	0.4239 ± 0.0054
iTransformer	0.1120 ± 0.0003	0.1861 ± 0.0053	0.0996 ± 0.0031	0.1604 ± 0.0007	0.4659 ± 0.0066
TFT	0.1117 ± 0.0012	0.1691 ± 0.0065	0.1025 ± 0.0059	0.1018 ± 0.0041	0.2780 ± 0.0160
TiDE	0.1081 ± 0.0013	0.1894 ± 0.0081	0.1038 ± 0.0044	0.1022 ± 0.0027	0.5106 ± 0.0581
TimeXer	0.1143 ± 0.0004	0.1881 ± 0.0070	0.1026 ± 0.0055	0.1618 ± 0.0018	0.4371 ± 0.0101
CrossLinear	0.1106 ± 0.0003	0.1813 ± 0.0026	0.0983 ± 0.0074	0.1632 ± 0.0016	0.4242 ± 0.0058
XLinear	0.1111 ± 0.0003	0.1975 ± 0.0081	0.0979 ± 0.0035	0.1611 ± 0.0033	0.4282 ± 0.0020
DAG	0.1107 ± 0.0008	0.1876 ± 0.0071	0.1041 ± 0.0038	0.0992 ± 0.0010	0.3903 ± 0.0147
Loki	$\mathbf{0.1047 \pm 0.0002}$	$\mathbf{0.1591 \pm 0.0025}$	$\mathbf{0.0924 \pm 0.0010}$	$\mathbf{0.0965 \pm 0.0039}$	$\mathbf{0.2623 \pm 0.0110}$

D Additional Experimental Results

D.1 Additional Evaluation Metrics

In addition to the standardized MAE and MSE reported in the main paper, we report standardized RMSE and original-scale WMAPE as complementary evaluation metrics. Tables 8 and 9 report the results. For each random seed, standardized RMSE is computed as

$$\mathrm{RMSE}_{\mathrm{std}} = \sqrt{\mathrm{MSE}_{\mathrm{std}}},$$

and is then averaged across seeds. Original-scale WMAPE is computed independently for each target channel as

$$\mathrm{WMAPE}_c = \frac{\sum_{n,h} |\widehat{y}_{n,h,c} - y_{n,h,c}|}{\sum_{n,h} |y_{n,h,c}|}, \tag{D.1}$$

followed by an equal average across target channels with nonzero denominators. Predictions and targets are evaluated in the original target scale. Reported uncertainties are sample standard deviations across seeds.

Loki achieves the lowest original-scale WMAPE on all five datasets, consistent with the standardized MAE results reported in the main paper.

D.2 Event-Regime Analysis

We further analyze forecasting performance according to whether active discrete-event indicators appear in the historical and forecast windows. Each test window is assigned to one of four regimes: Quiet→Quiet (Q→Q), Event→Quiet (E→Q), Quiet→Event (Q→E), or Event→Event (E→E). Table 10 reports the corresponding standardized MAE.

Table 10: Indicator-defined event-regime results on Ride-hailing. Standardized MAE values are reported as mean ± sample standard deviation across seeds. Lower MAE is better.

Regime	# Samples	Backbone	History-only	Future-only	**Full Loki**
Q→Q	4,492	0.14879 ± 0.00017	0.14793 ± 0.00010	0.14658 ± 0.00048	$\mathbf{0.14643 \pm 0.00025}$
E→Q	15,793	0.14772 ± 0.00015	0.14714 ± 0.00023	0.14528 ± 0.00061	$\mathbf{0.14427 \pm 0.00021}$
Q→E	1,269	0.15490 ± 0.00014	0.15538 ± 0.00031	0.14949 ± 0.00080	$\mathbf{0.14889 \pm 0.00046}$
E→E	14,046	0.18661 ± 0.00013	0.18734 ± 0.00020	0.17631 ± 0.00044	$\mathbf{0.17616 \pm 0.00055}$

The gain over the Backbone increases as active-event involvement grows, from 1.59% in Q→Q to 5.60% in E→E. The contribution of future-event modeling is larger when the forecast window is event-active, while the incremental benefit of historical Lift Out is largest in the E→Q transition. These patterns are consistent with the intended roles of Kick In and Lift Out.

E Controlled Event-Effect Recovery Details

E.1 Semi-Synthetic Event Construction

We construct controlled event-injected versions of Ride-hailing and M5 by injecting known multiplicative responses into the original reference series,

$$Y_{i,t,f} = Y^{\mathrm{ref}}_{i,t,f} \exp\left(S^{*}_{i,t,f}\right),$$

where i, t, and f index forecasting unit, time, and target variable. The reference series retains its original temporal dynamics, noise, covariates, and native event-related variation; the experiment adds only the controlled artificial response S^{*}.

Each forecasting unit receives non-overlapping artificial events $j \in \mathcal{E}_i$. Event j has type $k_j \in \{+, -\}$, corresponding to uplift and suppression, onset o_j, duration ℓ_j, and dose $q_j \in \{0.5, 1.0, 1.5\}$. Its injected log response is

$$S^{*}_{i,t,f} = \sum_{j \in \mathcal{E}_i} a_{i,k_j}\, b_{k_j,f}\, A_{k_j}\, q_j\, h_{k_j}(t; o_j, \ell_j), \tag{E.1}$$

where $a_{i,k} \sim U(0.7, 1.3)$ is a forecasting-unit- and event-type-specific gain. For Ride-hailing, $b_{k,f} \sim U(0.85, 1.15)$ is an event-type- and target-specific gain; M5 has one target and uses $b_{k,f} = 1$. The base log amplitudes are $A_{+} = \log(1.3)$ and $A_{-} = \log(0.75)$.

Table 11: Semi-synthetic event-generation settings used for controlled recovery.

Setting	Ride-hailing	M5
Temporal resolution	30 minutes	1 day
Event duration ℓ	12 / 24 / 48 steps	3 / 5 / 7 days
Uplift anticipation ℓ_{pre}	4 steps	1 day
Post-event tail ℓ_{post}	12 steps	2 days
Gap between response supports	48–144 steps	7–14 days

Table 12: Experimental conditions for controlled event-effect recovery.

Condition	Target	Artificial event descriptors
Full	event-injected	retained in history and forecast
Event-zeroed	same event-injected target	zeroed in history and forecast
Null control	original reference target	retained in history and forecast

Let ℓ_{pre} and ℓ_{post} denote the anticipation and post-event support lengths. Uplift uses a linear anticipatory ramp, a constant response during the event interval, and a linear decay after the event:

$$h_+(t;o,\ell) = \begin{cases} \dfrac{t-o+\ell_{\text{pre}}+1}{\ell_{\text{pre}}+1}, & o-\ell_{\text{pre}} \le t < o, \\ 1, & o \le t < o+\ell, \\ \dfrac{o+\ell+\ell_{\text{post}}-t}{\ell_{\text{post}}+1}, & o+\ell \le t < o+\ell+\ell_{\text{post}}, \\ 0, & \text{otherwise.} \end{cases} \tag{E.2}$$

Suppression has no anticipatory component; its response decreases linearly from 1 to 0.5 during the event and then enters a shorter tail beginning at one quarter of the base response:

$$h_-(t;o,\ell) = \begin{cases} 1 - \dfrac{t-o}{2(\ell-1)}, & o \le t < o+\ell, \\ \dfrac{1}{4}\dfrac{o+\ell+\ell_{\text{post}}-t}{\ell_{\text{post}}}, & o+\ell \le t < o+\ell+\ell_{\text{post}}, \\ 0, & \text{otherwise.} \end{cases} \tag{E.3}$$

Table 11 summarizes the dataset-specific temporal settings.

For each forecasting unit, events are scheduled independently within the original train, validation, and test splits. Artificial response supports neither overlap nor cross split boundaries. Event type, duration, and dose are sampled uniformly from their respective choices, and the gap between consecutive supports is sampled uniformly over the integer range in Table 11. Missing dates are handled by generating on the complete temporal grid and then mapping back to observed rows; no values are imputed and no additional observation noise is introduced.

We generate three independent semi-synthetic configurations, indexed by 4101–4103. Within a configuration, the generated schedules, doses, and gains are fixed across training seeds. The model receives two artificial event channels, uplift and suppression. For an event of type k, the corresponding input value is $e_{i,t,k} = q_j$ during $o_j \le t < o_j + \ell_j$ and zero otherwise. The anticipatory ramp and post-event tail are not directly encoded in these inputs. The gains a and b, temporal profiles h_k, and ground-truth response S^* are never provided as model inputs or direct supervision.

E.2 Experimental Conditions

We compare three conditions that isolate whether recovery depends on the artificial event descriptors. Table 12 summarizes how the target and descriptors differ across conditions.

In the *Full* condition, Loki is trained and evaluated on the injected series with the uplift and suppression channels available. *Event-zeroed* uses exactly the same injected targets and retains all native covariates, but sets both artificial channels to zero in the historical and forecast windows of the train,

validation, and test splits. The *null control* retains the same artificial schedules and doses while leaving the target equal to Y^{ref}, so its artificial-response ground truth is zero. This condition measures false attribution when event descriptors are present but no corresponding effect is injected.

Results are first averaged across the training seeds within each generated configuration and then averaged across the three configurations. Target scaling statistics are fitted using the training portion of the corresponding condition, and checkpoints are selected by validation forecasting MAE.

E.3 Recovery Metrics

Event-effect recovery. Let $\widehat{S}_{\text{syn}}$ denote the sum of the responses attributed by Loki to the uplift and suppression channels, and let S^* denote the known injected log response. For target f, event-effect MAE is

$$\text{MAE}_{\text{effect},f} = \frac{1}{|\mathcal{V}_f|} \sum_{(i,t)\in\mathcal{V}_f} \left| \widehat{S}_{\text{syn},i,t,f} - S^*_{i,t,f} \right|, \tag{E.4}$$

where $\mathcal{V}_f$ contains future positions that are both visible and identifiable. A position in an artificial response support is visible only when the corresponding event boundaries are fully observed in the joint history–forecast window; a position is identifiable when $Y^{\text{ref}}_{i,t,f} > 0$. The metric includes both artificial-response positions and valid quiet positions where $S^* = 0$. We first average within each target variable and then average equally across targets with valid evaluation points.

Temporal-shape recovery. For every complete artificial event and target pair, we compare the estimated and injected response curves over the event's full response support using cosine similarity,

$$\cos\left(\widehat{\mathbf{s}}, \mathbf{s}^*\right) = \frac{\widehat{\mathbf{s}}^\top \mathbf{s}^*}{\|\widehat{\mathbf{s}}\|_2 \|\mathbf{s}^*\|_2}, \tag{E.5}$$

where the vectors contain the estimated and ground-truth log responses along that event support. Curves are not mean-centered or smoothed. Similarities are averaged over eligible events within each run, then over training seeds within a generated configuration, and finally over configurations. This evaluates temporal-shape agreement without treating a high cosine value as evidence of exact magnitude recovery.

History restoration. Because Loki simultaneously models native and artificial event channels, let $\widehat{S}_{\text{native}}$ and $\widehat{S}_{\text{syn}}$ denote their respective estimated log responses. We isolate the effective artificial contribution after the response clipping used by the model,

$$\widehat{\Delta} = \text{clip}_{[-8,8]}\left(\widehat{S}_{\text{native}} + \widehat{S}_{\text{syn}}\right) - \text{clip}_{[-8,8]}\left(\widehat{S}_{\text{native}}\right). \tag{E.6}$$

The Lift Out–adjusted history is then

$$Y^{\text{hist}}_{\text{recovered}} = Y_{\text{hist}} \odot \exp\left(-\widehat{\Delta}_{\text{hist}}\right). \tag{E.7}$$

History-restoration error is the standardized MAE between $Y^{\text{hist}}_{\text{recovered}}$ and the pre-injection reference $Y^{\text{ref}}_{\text{hist}}$ over visible, identifiable historical positions within the artificial response support. Errors are divided by the corresponding training-target scale σ_f^{train}, averaged within targets, and then averaged equally across valid targets.

For the null control, $S^* = 0$. We report the mean absolute effective artificial log response $|\widehat{\Delta}|$ over the forecast window together with the standardized history-restoration error. These quantities measure false attribution rather than recovery of an injected component.

E.4 Recovery Results

Table 13 reports the quantitative recovery results on the semi-synthetic Ride-hailing and M5 datasets. Full Loki substantially improves event-effect recovery over Event-zeroed, reducing event-effect MAE by 87.69% on Ride-hailing and 86.71% on M5. The recovered response trajectories also closely match the injected temporal profiles, with cosine similarities ranging from 0.9966 to 0.9989 across the evaluated settings.

Table 13: Controlled event-effect recovery on semi-synthetic datasets. Results are averaged over three synthetic instances and multiple training seeds per instance. Zeroed denotes Event-zeroed; Full denotes Full Loki. Improv. reports relative MAE reductions from Zeroed, computed using unrounded means. Cosine similarities are reported for Full Loki.

	Event-effect MAE ↓			History-restoration MAE ↓			Temporal-shape cosine ↑ (Full)	
Dataset	Zeroed	Full	Improv.	Zeroed	Full	Improv.	Uplift	Suppression
Ride-hailing	0.0558	**0.0069**	87.69%	0.3180	**0.0200**	93.72%	0.9978	0.9966
M5	0.0522	**0.0069**	86.71%	0.1853	**0.0125**	93.26%	0.9989	0.9987

The recovered responses also perform the intended role of Lift Out. After removing the estimated artificial response from the perturbed history, Full Loki reduces history-restoration MAE by 93.72% on Ride-hailing and 93.26% on M5 relative to Event-zeroed, indicating that most of the injected variation can be removed before temporal extrapolation.

Finally, the null control evaluates false attribution when artificial event descriptors are present but no corresponding response is injected. The estimated artificial responses remain small but non-zero, with mean absolute effective log-response magnitudes of 0.00925 on Ride-hailing and 0.00271 on M5. These results reinforce that the learned ERF provides a predictive decomposition of event-associated variation rather than an exactly identified or causal estimate of event effects.