FROM LEAKAGE TO FIDELITY: RELIABLE BENCHMARKING FOR TEMPORAL CASCADE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal cascade prediction is widely studied, yet its empirical foundations remain fragile. Most existing works report results under random cascade splits that mix past and future signals, rely on datasets with limited features and no downstream conversion labels, and compare increasingly complex models without systematically examining whether benchmark conclusions are protocol-dependent. This paper argues that the field should move from leakage-prone evaluation toward fidelity-aware benchmarking. We introduce a protocol suite and renewed evaluation standard for temporal cascade prediction, centered on the Full Temporal protocol, overlap-based leakage diagnostics, and analyses of performance inflation and temporal drift. To broaden the scope of benchmark tasks, we also present Taoke, a real-world e-commerce cascade dataset with rich promoter/product features and observed purchase conversions, enabling both first-stage popularity forecasting and second-stage conversion forecasting under a shared benchmark asset. Finally, we include CasTemp as a lightweight reference method and additionally probe a larger same-task internal extension to verify that this pipeline remains operational at substantially greater scale. Together, these components turn cascade prediction from a protocol-sensitive leaderboard exercise into a more reliable analysis and benchmarking problem, while still providing a practical reference pipeline for large-scale evaluation and conversion-aware modeling.

1 INTRODUCTION

Temporal cascade prediction studies how the popularity of an information item evolves after an initial observation period. The problem appears in social media reposting (Weng et al., 2013; Cheng et al., 2014), citation accumulation (Min et al., 2017), traffic flow distribution (Han et al., 2024), and e-commerce promotion (Liu et al., 2016), where practitioners care not only about future diffusion volume but also about downstream business outcomes such as user engagement or purchases. Forecasting the propagation scale (i.e., popularity) of these cascades not only helps researchers better understand the mechanisms of information spread but is also crucial for numerous applications, including detecting viral misinformation (Leskovec et al., 2007; Weng et al., 2013), optimizing social media marketing strategies (Dave et al., 2011; Zhang et al., 2025), and enhancing e-commerce recommendation systems (Gao et al., 2019; Petrov et al., 2025; Peng et al., 2026b). Because of this broad relevance, the area has produced a long line of increasingly sophisticated models, ranging from feature-based regressors and recurrent architectures to graph neural networks, dynamic-memory models, and diffusion-style generators (Szabo & Huberman, 2010; Cao et al., 2017; Chen et al., 2019; Xu et al., 2021; Lu et al., 2023; Cheng et al., 2024). However, we argue that the central bottleneck of the field is no longer another modeling module, but the reliability of the benchmark itself. Three issues motivate this view.

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

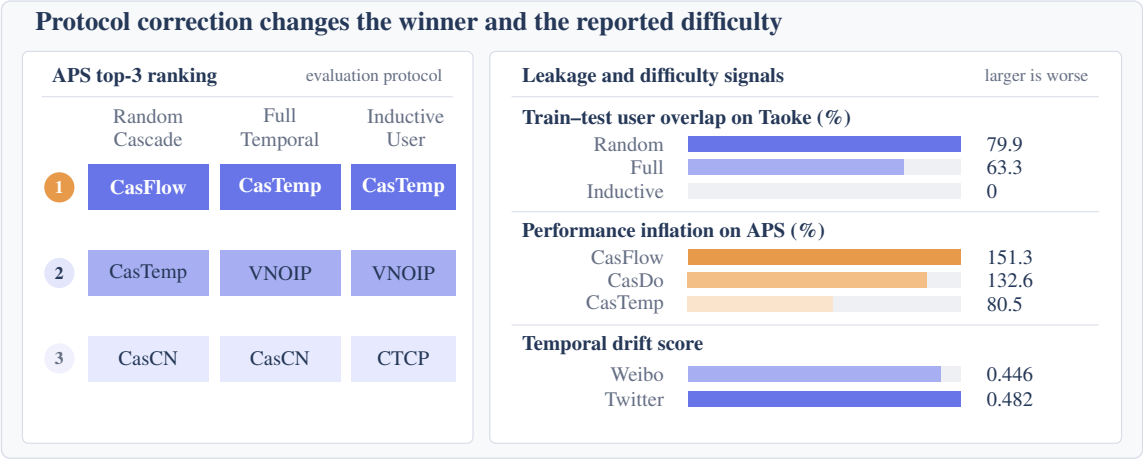

Figure 1: Correcting the evaluation protocol changes both method rankings and apparent task difficulty. **Left:** APS top-3 rankings under Random Cascade, Full Temporal, and Inductive User. **Right:** train–test user overlap on Taoke, performance inflation on APS, and temporal drift on Weibo and Twitter expose shortcuts hidden by optimistic evaluation.

Protocol validity is under-examined. Most prior work evaluates models with random cascade-level splits (Xu et al., 2021; Lu et al., 2023; Cheng et al., 2024). This practice is convenient, but it weakens the temporal semantics of forecasting: training and test cascades can share future-era activity patterns, overlapping users, and globally synchronized bursts. As a result, a model may benefit from shortcut signals that would not be available in a real deployment setting. If benchmark conclusions depend strongly on such optimistic splits, then leaderboard gains do not necessarily translate into reliable forecasting ability.

Current datasets under-support analysis and practical tasks. Widely used public datasets typically expose only propagation structures and timestamps, with limited cascade attributes, limited user features, and no downstream conversion labels. This makes it hard to study why models succeed or fail, and it also narrows the task definition to first-stage popularity only. In many real systems, though, the relevant question is not merely whether a cascade spreads, but whether it leads to actions such as clicks, comments, or purchases (Liu et al., 2007; Bodapati, 2008; Liu et al., 2010; Thelwall, 2018). Benchmarking should therefore cover richer data and richer targets.

Model complexity has outpaced benchmark scrutiny. Recent methods introduce increasingly heavy machinery to capture structural and temporal dependencies. Yet if the evaluation protocol itself is optimistic, it becomes unclear whether this complexity is learning genuine diffusion dynamics or simply exploiting artifacts in the split. For an analysis-and-benchmark paper, an important question is therefore not only "*which model wins*," but also "*which conclusions remain stable once the protocol becomes realistic*."

Thus, this paper treats temporal cascade prediction as a benchmark-design problem rather than only a model-design problem. Instead of centering the narrative on yet another state-of-the-art method, we build a benchmark-oriented framework to test whether existing conclusions remain robust under leakage-aware evaluation. Concretely, we reorganize evaluation around three representative splits that form our protocol suite: (i) **Random Cascade**, the conventional baseline that partitions cascades randomly without a global time boundary, allowing training and test sets to share future-era patterns and overlapping users; (ii) **Full Temporal**, our advocated primary protocol that uses consecutive global time windows for observation and prediction, ensuring test supervision is drawn strictly from later periods while permitting long-lived items and users to persist across adjacent windows; and (iii) **Inductive User**, the most stringent setting

that combines the temporal cut with disjoint user sets across splits, eliminating identity-based shortcuts entirely. Beyond the protocol suite, we broaden the benchmark assets with **Taoke**, a real-world e-commerce dataset containing rich promoter/product features and downstream purchase conversions, enabling tasks beyond first-stage popularity forecasting. We further introduce **CasTemp** as a lightweight reference pipeline that models self- and cross-cascade temporal evidence through temporal walks and time-aware encoding; it serves as an efficient anchor for interpreting the benchmark under stricter protocols. As illustrated in Figure 1, correcting the protocol changes the rankings on the public cascade dataset APS, exposes substantial user overlap and performance inflation, and reveals temporal drift; Taoke then broadens the benchmark to conversion forecasting, while CasTemp keeps the stricter evaluation practical to run. A fuller roadmap and detailed interpretation of these signals is provided in Section D.

Our contributions are summarized as follows.

- **A protocol suite and renewed evaluation standard for temporal cascade prediction.** We identify the protocol sensitivity of current cascade benchmarks and reorganize evaluation around chronologically faithful splits, leakage diagnostics, inflation analysis, and temporal drift analysis.
- **A conversion-rich benchmark asset.** We introduce Taoke, a real-world cascade dataset with rich promoter/product features and downstream purchase conversions, enabling benchmark tasks beyond first-stage popularity prediction.
- **A lightweight reference pipeline.** We present CasTemp as a simple and efficient reference method that remains competitive under stricter evaluation while supporting scalable benchmarking.

2 Related Work

Information Cascade Prediction. Early work on popularity prediction relies on hand-crafted features—such as user profiles, structural depth, and temporal dynamics—combined with logistic regression or similar models (Szabo & Huberman, 2010; Cheng et al., 2014). Later, RNN- and GRU-based methods model cascades as sequential events to capture temporal evolution (Cao et al., 2017; Liao et al., 2019), yet underutilized their inherent tree- or graph-structured topologies. This motivates graph-based approaches that represent cascades as evolving graphs and apply Graph Neural Networks (GNNs) for representation learning (Li et al., 2017; Xu et al., 2021), effectively capturing local propagation patterns but typically treating cascades in isolation, ignoring inter-cascade interactions like competition. Recent models address this: CTCP (Lu et al., 2023) captures cross-cascade correlations via memory updates, while CasDo (Cheng et al., 2024) employs probabilistic diffusion ODEs (Hartman, 2002) for propagation uncertainty. However, these advances incur high architectural complexity and computational overhead. Overall, early methods are structurally limited, while modern GNN-based models face scalability challenges. Moreover, most adopt cascade-based splits (Xu et al., 2021; Lu et al., 2023; Cheng et al., 2024; Wang et al., 2026) that fail to temporally separate train/test data, introducing time-travel bias through future information leakage and yielding inflated performance estimates.

Dynamic Graph Learning. Cascade propagation aligns naturally with continuous-time dynamic graphs—sequences of timestamped events (Yu et al., 2023; Peng et al., 2025; 2026a). Dynamic graph learning models node representations for link prediction and classification via memory-based (Kumar et al., 2019; Wang et al., 2021), random walk-based (Starnini et al., 2012; Li et al., 2023; Lu et al., 2024), and Transformer-based approaches (Yu et al., 2023; Peng et al., 2025). For popularity prediction, CTCP (Lu et al., 2023) uses memory mechanisms to update states per event but suffers from computational bottlenecks due to frequent updates. Efficiently adapting dynamic graph techniques like temporal random walks and sequential modeling to cascade prediction remains an open challenge.

3 PRELIMINARIES

Information Cascade Graph. The definition of conventional information cascade graphs naturally aligns with that of dynamic graphs. We consider a dynamic graph $G = (V, E, T)$ characterized by sets of nodes V, edges E, and timestamps T. It captures the evolution of cascading diffusion through a sequence of chronologically ordered events $G = \{(src_i, tgt_i, c_i, t_i)\}_{i=1}^{N}$, with $0 \leq t_1 \leq \cdots \leq t_N$. Each event denotes the diffusion of a cascade c_i (e.g., a post, paper, or product) from a source node $src_i \in V$ to a target node $tgt_i \in V$ at time $t_i \in T$. Nodes src_i and tgt_i are associated with features $\mathcal{N}_i^{src}, \mathcal{N}_i^{tgt} \in \mathbb{R}^{d_n}$, and the corresponding edge has a feature vector $\mathcal{E}_i^{t_i} \in \mathbb{R}^{d_e}$ encoding timestamped cascade features, where d_n and d_e denote the dimensions of node and edge embeddings, respectively.

Information Cascade Popularity Prediction. We use $G^c(t)$ to denote the evolution process of cascade c up to time t. Given a cascade c begins at t_0^c, after observing it for time Δt_1, the task is to predict its incremental cascade popularity $\Delta P = |G^c(t_0^c + \Delta t_2)| - |G^c(t_0^c + \Delta t_1)|$ over a future prediction window $\Delta t_2 - \Delta t_1$.

Extended Cascade Graph with Popularity Conversions. To support second-stage popularity conversion prediction, we extend the conventional information cascade graph by incorporating interactions between diffusion users and downstream users. Specifically, after each diffusion event from src_j to tgt_j, the target diffusion user tgt_j may trigger conversion events involving end users, such as likes, comments, or purchases, denoted as $(tgt_j, user_j, c_j, t_j)$. We collect these as a set $H = \{(tgt_j, user_j, c_j, t_j)\}_{j=1}^{M}$ with $0 \leq t_1 \leq \cdots \leq t_M$, forming an augmented view of the cascade dynamics that includes both propagation and conversion stages.

Second-stage Popularity Conversion Prediction. We use $H^c(t)$ to denote the popularity conversion process of cascade c by time t. Given a cascade c begins at t_0^c, after observing it for time Δt_1, we predict the incremental conversions $\Delta C = |H^c(t_0^c + \Delta t_2)| - |H^c(t_0^c + \Delta t_1)|$ in the subsequent interval $\Delta t_2 - \Delta t_1$.

4 BENCHMARK ASSETS AND PROTOCOLS

4.1 WHY THE EXISTING BENCHMARK NEEDS REVISION

Random cascade-level splits mix training and test cascades from overlapping eras despite the prediction target being future growth. This misalignment introduces three leakage channels: **synchronized temporal bursts** (memorizing global activity spikes), **user behavior overlap** (exploiting repeated promoter patterns), and **future-era distribution shifts** (violating causal assumptions). Collectively, these shortcuts decouple model complexity from genuine predictive capacity.

Our controlled toy study in Table 1 confirms this failure mode: under conventional random splits, a simple MLP achieves performance comparable to or better than substantially more complex architectures by exploiting temporal shortcuts, whereas a globally time-ordered split restores the expected forecasting hierarchy. The complete toy construction, loss curves, and full results are reported in Section E. Crucially, this diagnosis isolates each leakage channel so that the same failure

Table 1: MLP vs. best complex method under Random Cascade and Full Temporal splits in the proposed toy study. Under Random Cascade, MLP matches complex models via leakage; under Full Temporal, architectural expressiveness matters. See Tables 8 and 9 in Section E for implementation details and full results of the toy study.

Split	MLP MSLE	Best Method MSLE
Random Cascade	0.451	0.625 (CasFlow)
Full Temporal	2.908	1.610 (CasTemp)

modes observed at benchmark scale in Section 6 can be attributed to protocol design instead of dataset-specific artifacts.

4.2 Evaluation Protocols

We evaluate a three-part protocol family whose full mathematical definitions and notation appear in Section E.1. **Random Cascade** reproduces the historical practice of assigning cascades randomly while using local observation and target windows. **Full Temporal** instead constructs train, validation, and test examples from consecutive global windows, ensuring that every prediction target follows its observation era; it is our default reporting protocol. **Inductive User** adds disjoint train–test user sets to the same chronology, making it a deliberately stringent diagnostic of identity transfer.

4.3 Taoke: A Conversion-Rich Benchmark Dataset

Motivation. Existing public cascade datasets typically provide only propagation structures and timestamps, lacking both rich contextual features and downstream conversion labels. This limitation restricts explanatory analysis and narrows the task scope to first-stage popularity forecasting, reducing real-world relevance.

Dataset scope. Taoke addresses this gap through a real e-commerce promotion scenario on Taobao. As illustrated in Figure 2, each cascade captures product diffusion among Taoke promoters together with exposure-to-purchase conversion events, enabling two tasks within a single asset: first-stage popularity forecasting and second-stage conversion forecasting. Taoke also provides richer side information—including promoter profiles and product attributes such as price and commission rate—allowing researchers to study how business context interacts with propagation dynamics.

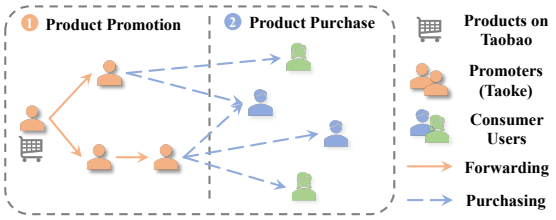

Figure 2: Overview of the Taoke benchmark. Each cascade tracks product diffusion among promoters alongside downstream purchase conversions.

As shown in Table 2, Taoke complements existing benchmarks by retaining comparable structural properties while uniquely providing conversion labels and rich features. The depth distribution in Figure 11 further confirms that Taoke remains structurally consistent with known cascade behavior despite its application-grounded setting. Beyond the released asset, we construct an internal large-scale extension, *Taoke-large*, which preserves identical semantics and schema but covers a substantially larger product pool. Because it is not publicly released and several baselines exceed runtime budgets on it, Taoke-large serves solely as an efficiency stress test rather than a second leaderboard. Together, these assets expand the experimental questions the field can ask.

Table 2: Statistical comparison across four benchmark datasets.

Metric	Twit.	Weibo	APS	Taoke
Cascade	67,760	48,693	90,768	2,862
Nodes	145K	353K	118K	29,711
Pop. Events	412K	497K	574K	979K
Conv. Events	–	–	–	2,990K
Avg. Path	3.86	3.51	5.12	5.61
Casc. Feat.	✗	✗	✗	✓
Node Feat.	✗	✗	✗	✓

4.4 Benchmark Diagnosis and Reporting Standard

The full protocol diagnosis in Section F demonstrates that model rankings, shortcut exposure, and temporal drift all change materially once the data split is made forecasting-faithful. As shown in the ranking matrix

Figure 3: Top-3 ranking matrix across three protocols. Darker shading indicates a better rank. 'R', 'F', and 'I' denote Random Cascade, Full Temporal, and Inductive User, respectively. Full numerical results are provided in Tables 13 to 16.

Twitter			
Model	**R**	**F**	**I**
CasTemp	1	1	1
CasCN	2	2	2
DeepHawkes	3	–	–
VNOIP	–	3	–
CTCP	–	–	3

Weibo			
Model	**R**	**F**	**I**
CasTemp	1	1	1
CasFlow	2	2	–
CasCN	3	–	–
DeepHawkes	–	3	3
CTCP	–	–	2

APS			
Model	**R**	**F**	**I**
CasTemp	2	1	1
CasFlow	1	–	–
CasCN	3	3	–
VNOIP	–	2	2
CTCP	–	–	3

Taoke			
Model	**R**	**F**	**I**
CasTemp	1	1	1
CasFlow	2	2	3
CasCN	3	–	–
DeepHawkes	–	3	–
CTCP	–	–	2

(Figure 3), our multi-protocol design immediately exposes ranking instability: on APS, the leading method flips from CasFlow under Random Cascade to CasTemp under Full Temporal; on Taoke, Random Cascade and Full Temporal retain 79.9% and 63.3% train–test user overlap, respectively, while Inductive User strictly removes it. These shifts prove that a single leaderboard score is insufficient to capture model robustness.

Exposing performance inflation. A core diagnostic capability of our benchmark is quantifying how optimistic protocols overstate progress. By comparing Random Cascade against Full Temporal, we uncover severe performance inflation driven by temporal leakage. On APS, for instance, Random Cascade artificially improves every baseline's error metric by 37.9% to 151.3% (e.g., CasFlow inflates by +151.3%) relative to the forecasting-faithful Full Temporal split. This striking gap (Table 11 in Appendix) serves as a clear case study: conventional random splits do not merely introduce minor noise, but fundamentally distort the evaluation landscape by allowing models to memorize future cascade trajectories.

Revealing temporal non-stationarity. Furthermore, our benchmark diagnoses the invalid IID assumption inherent in random mixing. Analysis of adjacent-window temporal drift (Figure 6, Table 12) reveals that none of the datasets are temporally stationary. Twitter and Weibo exhibit high user and cascade turnover, meaning random mixing easily blurs distinct activity eras. Conversely, Taoke features persistent user identities but significant burstiness drift driven by genuine business cycles (e.g., average cascade size rises from 51.8 to 71.8 across windows). This non-stationarity physically invalidates Random Cascade, as it forces models to train on future distributions to predict past events.

Reporting standard. Based on these diagnostics, we establish a renewed evaluation standard requiring three forms of evidence: forecasting-faithful default results, explicit shortcut diagnostics, and practical executability. We adopt **Full Temporal** as the main benchmark protocol because it preserves the chronology of forecasting while remaining executable on all released datasets. Random Cascade is retained strictly as an optimistic historical comparison to measure inflation, and Inductive User serves as a stress test for user-transfer reliance. Ultimately, our benchmark shifts the field from chasing a single inflated score to providing comprehensive, protocol-aware evidence.

5 Reference Method: CasTemp

We propose **CasTemp**, a lightweight reference method designed specifically for this benchmark. Rather than introducing another highly specialized architecture, CasTemp serves as an easy-to-rerun, low-maintenance baseline that is expressive enough to avoid being a straw-man, yet simple enough to prevent benchmark conclusions from being confounded by excessive architectural complexity. CasTemp models cascade dynamics through three core components: (1) a *competition graph* constructed via promoter-set Jaccard overlap to capture lightweight inter-cascade context; (2) a *recency-aware temporal encoder* that combines self-propagation

and cross-propagation sequences with exponential decay priors; and (3) *shallow prediction heads* that fuse these representations with historical features for non-negative popularity forecasting. On Taoke, the same pipeline is seamlessly extended to second-stage conversion prediction by conditioning on the predicted popularity, tightly coupling propagation and conversion tasks. Full mathematical formulations and architectural details are provided in Section H.

Why CasTemp fits this paper. CasTemp strikes a deliberate balance for benchmarking: it remains competitive under the strict Full Temporal protocol and supports the added conversion task, ensuring that performance gains of future models can be cleanly disentangled from benchmark artifacts and attributed to transparent architectural advances instead of opaque engineering.

6 Benchmark Evaluation and Reference Studies

The analyses in Sections 4.1 and 4.4 establish why existing protocols are insufficient. What remains here are the evaluations that presuppose a sound protocol: the Full Temporal split, the richer Taoke task, and the systems-oriented behavior of a lightweight reference pipeline. The emphasis is on cross-model insight under the renewed benchmark standard: which ranking patterns persist, which disappear, and what additional properties become visible once evaluation is forecasting-faithful.

6.1 Experimental Setup

We evaluate on four real-world datasets (Twitter, Weibo, APS, Taoke), using MSLE/MALE for popularity prediction. For conversion prediction, we additionally adopt Hit@40, which measures the proportion of predictions whose error relative to the ground truth is less than 40%. We compare seven representative methods: DeepHawkes, CasCN, CasFlow, CTCP, CasDo, VNOIP, and CasTemp. All baselines are integrated from official codebases into our unified framework to ensure fair comparison. See Sections A and B for detailed information of the datasets and baselines. Following the diagnosis in Section F, Full Temporal serves as the default reporting protocol. Implementation details and hyperparameters are provided in Section I.

6.2 Performance Under the Main Protocol

Table 3: Popularity prediction performance (MSLE ↓) under Full Temporal. Best results are bolded.

Method	Twitter	Weibo	APS	Taoke
DeepHawkes	1.270 ± 0.037	1.729 ± 0.005	2.524 ± 0.053	3.363 ± 0.132
CasCN	1.194 ± 0.020	2.587 ± 0.517	2.496 ± 0.029	3.432 ± 0.054
CasFlow	1.240 ± 0.007	1.633 ± 0.022	2.659 ± 0.035	2.288 ± 0.040
CTCP	1.447 ± 0.002	1.883 ± 0.002	2.802 ± 0.016	8.384 ± 0.021
CasDo	2.101 ± 0.030	2.143 ± 0.461	4.839 ± 0.201	27.166 ± 0.172
VNOIP	1.221 ± 0.053	2.478 ± 0.061	2.399 ± 0.083	6.574 ± 0.105
CasTemp	$\mathbf{1.168 \pm 0.003}$	$\mathbf{1.478 \pm 0.009}$	$\mathbf{1.924 \pm 0.016}$	$\mathbf{0.798 \pm 0.040}$

Table 3 reports popularity-forecasting results under Full Temporal. The interpretation is intentionally benchmark-centric: instead of declaring a universal winner, the results expose a heterogeneous landscape where strengths depend strongly on dataset conditions. Four observations highlight the diagnostic value of this benchmark. First, stricter protocols compress apparent separation: on Twitter, the gap between CasCN and CasTemp narrows, consistent with removing shortcut-driven gains. Second, baseline rankings remain highly dataset-dependent (e.g., VNOIP leads on APS, CasFlow on Weibo), revealing that different datasets reward distinct inductive biases. Third, larger margins on APS and Taoke indicate that chronological fidelity exposes where temporal context matters most substantively. Fourth, no single dataset proxies the entire problem class; robustness across domains matters more than peak performance on one split.

6.3 TAOKE AS A BENCHMARK FOR SECOND-STAGE CONVERSION

Table 4 evaluates conversion forecasting under the same time-aware split. All methods reuse first-stage embeddings, testing representation transfer instead of end-to-end redesign. CasTemp reduces MSLE from 8.380 to 1.955 and raises Hit@40 from 18.42% to 54.60% over the best non-CasTemp baseline. More importantly, methods competitive on popularity do not transfer equally well to conversion, producing sharper ranking separation. The ablation rows confirm that predicted popularity ($\hat{y}_i^{\text{pop}}$) serves as a meaningful bridge: removing it causes significant deterioration, while oracle popularity yields only marginal improvement, indicating gains come from calibrated signal exposure rather than future information leakage.

Table 4: Conversion prediction on Taoke (Full Temporal). CasTemp-W/O P removes $\hat{y}_i^{\text{pop}}$; W/ GTP uses oracle popularity.

Method	Taoke (Conversion)		
	MSLE	MALE	Hit@40
DeepHawkes	42.515 ± 1.601	5.912 ± 0.235	$2.10\% \pm 0.07\%$
CasCN	8.380 ± 0.119	2.158 ± 0.068	$18.42\% \pm 1.41\%$
CasFlow	18.992 ± 0.489	3.071 ± 0.097	$12.06\% \pm 1.37\%$
CTCP	10.494 ± 0.105	2.489 ± 0.040	$16.23\% \pm 0.99\%$
CasDo	35.588 ± 1.418	5.033 ± 0.219	$2.11\% \pm 0.13\%$
VNOIP	9.339 ± 0.593	2.289 ± 0.095	$16.83\% \pm 0.60\%$
CasTemp	$\mathbf{1.955 \pm 0.120}$	$\mathbf{0.764 \pm 0.023}$	$\mathbf{54.60\% \pm 0.85\%}$
CasTemp-W/O P	3.506 ± 0.331	1.237 ± 0.100	$45.69\% \pm 1.22\%$
CasTemp-W/ GTP	1.863 ± 0.104	0.711 ± 0.046	$53.68\% \pm 0.83\%$

6.4 EFFICIENCY AND SCALABILITY

Table 6 summarizes two kinds of systems evidence. For the four released benchmark datasets, it reports the identity-run efficiency comparison between CasTemp and the fastest baseline under the same protocol. It additionally includes Taoke-large, an internal large-scale extension with the same task semantics as Taoke but substantially larger scale, to assess whether the reference pipeline remains runnable when the product pool grows well beyond the released benchmark. Its scale relative to the released Taoke benchmark is summarized in Table 5. On the released datasets, the reference pipeline is between 140× and 412× faster to train, and between 375× and 2457× faster to infer, than the fastest baseline competitor under the same protocol. Together with Figure 8, this shows that the efficiency advantage is more structural. For a benchmark study, this matters because a renewed evaluation standard must remain practical. Taoke-large strengthens that claim from a stress-test perspective: CasTemp still finishes, whereas all baseline systems exceed the 48-hour training budget and are therefore marked OOT instead of compared as a full leaderboard.

Table 5: Scale of the internal Taoke-large extension used for stress testing. It preserves the same semantics, features, and task definitions as the released Taoke dataset.

Metric	Taoke	Taoke-large
Cascades	2,862	126,515
Nodes	29,711	416,880
Pop. Events	979,635	2,496,732
Conv. Events	2,990,747	16,699,943

Table 6: Efficiency summary under Full Temporal. For the released benchmark datasets, the baseline shown is the fastest non-CasTemp method under the corresponding metric, and speedup is computed relative to that baseline. Taoke-large is an internal non-public extension with the same task semantics as Taoke but larger scale; it is reported only as a stress test, and OOT denotes the training time over 48 hours.

Dataset	Train	Baseline	Train Ref.	Infer	Baseline	Infer Ref.
Twitter	44.58s	CTCP	6219.47s / 139.5×	0.0185s	CTCP	19.76s / 1066×
Weibo	13.18s	CTCP	5426.36s / 411.8×	0.0089s	CTCP	14.33s / 1615×
APS	29.04s	CTCP	6908.98s / 237.9×	0.0135s	CTCP	33.09s / 2457×
Taoke	8.30s	CasCN	1390.60s / 167.6×	0.0104s	CasCN	3.91s / 374.9×
Taoke-large	1504.30s	N/A	All baselines OOT	30.5532s	N/A	All baselines OOT

Synthetic expansion analysis. The synthetic-expansion results in Table 7 complement the cross-model timing comparison. They isolate the scaling behavior of CasTemp itself by expanding Taoke from 1.96M to 15.67M selected events. Preprocessing grows from 12.37s to 63.65s, which is consistent with the expected increase in input volume, while train and inference remain in the narrow ranges. This suggests that, for CasTemp, the dominant cost is not the predictor itself but the data preparation around it. This separation is important: the identity-run table in Table 6 asks whether CasTemp is a practical benchmark anchor today, while the synthetic-expansion table asks whether the same pipeline remains usable as the event stream grows.

Table 7: CasTemp scalability on Taoke under synthetic event expansion.

Ratio	Events (M)	Preproc. (s)	Train (s)	Infer (s)
2×	1.96	12.37	8.58	0.0109
4×	3.92	19.54	9.03	0.0111
8×	7.84	34.58	9.37	0.0118
16×	15.67	63.65	9.99	0.0129

6.5 Reference Method Diagnostics

To ensure CasTemp's competitiveness stems from principled design instead of brittle tuning, we conduct extensive hyperparameter sensitivity and ablation studies. Results in Section J confirm stable performance across reasonable settings. Ablations further validate that the competition graph, cross-cascade sequence construction, and temporal decay each contribute non-trivially, confirming effectiveness is not driven by isolated tricks. These diagnostics reposition temporal cascade prediction as a benchmark design problem, where leakage-aware evaluation and conversion forecasting offer a rigorous testing ground for representation transfer.

6.6 Extended Discussion

Taken together, these experiments reposition temporal cascade prediction as a benchmark design problem as much as a modeling problem. Several cross-model insights now emerge more clearly. First, reliable evaluation does not preserve a single, globally dominant baseline family; instead, rankings become more dataset-specific once leakage channels are controlled. Second, expanding the task from popularity forecasting to conversion forecasting further changes the comparison surface by rewarding transferability, not only one-shot predictive fit. From a systems perspective, the benchmark remains runnable at practical cost, which is necessary if the proposed protocol suite is to function as shared infrastructure. Full Temporal serves as the default lens for answering that question, Taoke broadens the benchmark toward conversion-aware applications, and CasTemp provides a simple anchor method that helps interpret the benchmark without dominating the paper's contribution.

7 Conclusion

This paper reframes temporal cascade prediction as a benchmarking problem as much as a modeling problem. It delivers a benchmark asset, a protocol suite, and a renewed evaluation standard: Taoke expands the task scope with conversion-aware forecasting, Full Temporal and Inductive User clarify how temporal fidelity and user-transfer leakage should be handled, and the reporting policy distinguishes a default comparison protocol from diagnostic stress tests. Reliable conclusions in this area require more than another leaderboard result. They require benchmark specifications that make leakage channels visible, richer assets that test more than one target, and reference pipelines that keep trustworthy evaluation practical to run. Our additional Taoke-large stress test reinforces that practicality claim without being conflated with the released benchmark asset itself.

REFERENCES

Anand V Bodapati. Recommendation systems with purchase data. *Journal of marketing research*, 45(1): 77–93, 2008.

Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1149–1158, 2017.

Xueqin Chen, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Fengli Zhang. Information diffusion prediction via recurrent cascades convolution. In *2019 IEEE 35th international conference on data engineering (ICDE)*, pp. 770–781. IEEE, 2019.

Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pp. 925–936, 2014.

Zhangtao Cheng, Fan Zhou, Xovee Xu, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Philip S Yu. Information cascade popularity prediction via probabilistic diffusion. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Kushal Dave, Rushi Bhatt, and Vasudeva Varma. Modelling action cascades in social networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, pp. 121–128, 2011.

Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597–1600. IEEE, 2017.

Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, Lina Yao, Yang Song, and Depeng Jin. Learning to recommend with multiple cascading behaviors. *IEEE transactions on knowledge and data engineering*, 33(6):2588–2601, 2019.

Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. Bigst: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks. *Proceedings of the VLDB Endowment*, 17(5):1081–1090, 2024.

Philip Hartman. *Ordinary differential equations*. SIAM, 2002.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1269–1278, 2019.

Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5–es, 2007.

Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*, pp. 577–586, 2017.

Yiming Li, Yanyan Shen, Lei Chen, and Mingxuan Yuan. Zebra: When temporal graph neural networks meet temporal personalized pagerank. *Proceedings of the VLDB Endowment*, 16(6):1332–1345, 2023.

Dongliang Liao, Jin Xu, Gongfu Li, Weijie Huang, Weiqing Liu, and Jing Li. Popularity prediction on online articles with deep fusion of temporal process and content features. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 200–207, 2019.

Qihua Liu, Shan Huang, and Liyi Zhang. The influence of information cascades on online purchase behaviors of search and experience products. *Electronic Commerce Research*, 16(4):553–580, 2016.

Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. Arsa: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 607–614, 2007.

Yang Liu, Xiaohui Yu, Xiangji Huang, and Aijun An. S-plasa+ adaptive sentiment analysis with application to sales performance prediction. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 873–874, 2010.

Xiaodong Lu, Shuo Ji, Le Yu, Leilei Sun, Bowen Du, and Tongyu Zhu. Continuous-time graph learning for cascade popularity prediction. *arXiv preprint arXiv:2306.03756*, 2023.

Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. Improving temporal link prediction via temporal walk matrix projection. *Advances in Neural Information Processing Systems*, 37:141153–141182, 2024.

Chao Min, Jianjun Sun, and Ying Ding. Quantifying the evolution of citation cascades. *Proceedings of the Association for Information Science and Technology*, 54(1):761–763, 2017.

Jie Peng, Zhewei Wei, and Yuhang Ye. Tidformer: Exploiting temporal and interactive dynamics makes a great dynamic graph transformer. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2245–2256, 2025.

Jie Peng, Jiarui Ji, Runlin Lei, Zhewei Wei, Yongchao Liu, and Chuntao Hong. Gdgb: A benchmark for generative dynamic text-attributed graph learning. In *International Conference on Learning Representations*, volume 2026, pp. 108768–108822, 2026a.

Jie Peng, Yanping Zheng, Zhewei Zhe, Bin Tong, Guan Wang, and Bo Zheng. Can generative recommendation reach cold items? a temporal perspective on semantic-id generation. *arXiv preprint arXiv:2607.21101*, 2026b.

Aleksandr V Petrov, Craig Macdonald, and Nicola Tonellotto. Efficient recommendation with millions of items by dynamic pruning of sub-item embeddings. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2102–2111, 2025.

Michele Starnini, Andrea Baronchelli, Alain Barrat, and Romualdo Pastor-Satorras. Random walks on temporal networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 85(5):056115, 2012.

Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.

Mike Thelwall. Social media analytics for youtube comments: Potential and limitations. *International Journal of Social Research Methodology*, 21(3):303–316, 2018.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, et al. Apan: Asynchronous propagation attention network for real-time temporal graph embedding. In *Proceedings of the 2021 international conference on management of data*, pp. 2628–2638, 2021.

Yuchen Wang, Dongpeng Hou, Weikai Jing, Chao Gao, Xianghua Li, and Yang Liu. Modeling trend dynamics with variational neural odes for information popularity prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1231–1239, 2026. doi: 10.1609/aaai.v40i2.37095.

Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Virality prediction and community structure in social networks. *Scientific reports*, 3(1):2522, 2013.

Xovee Xu, Fan Zhou, Kunpeng Zhang, Siyuan Liu, and Goce Trajcevski. Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3484–3499, 2021.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023.

Xiaoqing Zhang, Yuhan Liu, Jianzhou Wang, Zhenxing Hu, Xiuying Chen, and Rui Yan. Sagraph: A large-scale social graph dataset with comprehensive context for influencer selection in marketing. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3733–3742, 2025.

APPENDIX CONTENTS

General Information

- AI Use Statement 14
- Ethics Statement 14
- Reproducibility Statement 14

Datasets and Baselines

- **Appendix A** Datasets 14
- **Appendix B** Baselines 15
- **Appendix C** Data Privacy and Ethical Considerations 15

Motivation and Evaluation

- **Appendix D** Extended Motivation and Figure Interpretation 16
- **Appendix E** Evidence of Information Leakage in Random Cascade Splits 16
- **Appendix F** Analysis of the Evaluation Protocols 18
- **Appendix G** Evaluation and Reporting Notes 20

Methodology and Experiments

- **Appendix H** Reference Method Details 20
- **Appendix I** Implementation Details and Hyperparameters 22
- **Appendix J** Ablation Study 22
- **Appendix K** Additional Dataset Characterization 22
- **Appendix L** Additional Experimental Results 24

AI USE STATEMENT

We used generative AI tools to aid and polish the writing of this paper, including improvements to clarity, grammar, and phrasing. We also used such tools for retrieval and discovery, for example, to identify potentially relevant related work. All retrieved references were checked by the authors against the original sources before inclusion, and all AI-assisted text was reviewed and revised by the authors. Generative AI was not used to originate the core research questions, benchmark design, evaluation protocols, methodological ideas, experimental conclusions, or scientific claims of this work. These contributions are original to the authors, who take full responsibility for the paper and its contents.

ETHICS STATEMENT

The primary ethical considerations of this work concern the collection, processing, and release of the Taoke dataset, which contains observational records of e-commerce promotion and purchase behavior. As detailed in Section C, all personally identifiable information has been removed, user identifiers have been replaced with irreversible pseudonymous tokens, and no raw text or specific brand information is released. Sensitive auxiliary attributes are provided only in desensitized form. The dataset and its collection and release procedures were reviewed and approved by the Legal and Security Department of the data-source company. The benchmark is intended for research on aggregate diffusion and conversion forecasting; users should comply with the accompanying license and must not attempt to re-identify individuals or use the data for individual-level profiling. These safeguards mitigate privacy, security, and misuse risks while retaining research utility.

REPRODUCIBILITY STATEMENT

We will make all materials required to reproduce the released benchmark results available through an anonymous repository during review (`https://anonymous.4open.science/r/Cascade_Bench-CCE2`) and release them publicly thereafter. The artifact will include the Taoke dataset introduced in this paper, preprocessing and protocol-construction scripts, fixed data splits, model implementations, training and evaluation code, experiment configurations, and analysis scripts.

A DATASETS

We evaluate our method on four real-world datasets spanning social media, academic networks, and e-commerce: three public datasets-**Twitter**, **Weibo**, **APS**—and a newly introduced dataset, **Taoke**.

- **Twitter (Weng et al., 2013)** contains tweets posted between March 24 and April 25, 2012, along with their retweet cascades. Each cascade represents the diffusion of a specific hashtag.
- **Weibo (Cao et al., 2017)** is collected from Sina Weibo, the most popular Chinese microblogging platform, and includes posts published on July 1, 2016, and their subsequent reposts. Each cascade corresponds to the propagation of a single post.
- **APS** [1] consists of papers published in journals of the American Physical Society (APS) prior to 2017 and their citation relationships. Each cascade models the accumulation of citations for a given paper. Following prior work, we reformulate citation prediction as a cascade forecasting task through appropriate transformation and preprocessing.

[1] `https://journals.aps.org/datasets`

- **Taoke** is a new real-world dataset introduced in this work. It captures the cascade-like diffusion of product promotions among Taobao promoters (Taoke), where each cascade represents the forwarding of a certain product. Crucially, the dataset also records downstream conversion behavior—i.e., actual consumer purchases—enabling research on second-stage popularity conversion and bridging the gap between cascade prediction and practical applications. It contains all forwarding and transaction records of selected high-sales products on Taobao from August 9, 2025 to August 12, 2025. We also construct an internal extension, **Taoke-large**, with the same semantics, features, and task definitions but a much larger product pool; it is used only for large-scale efficiency stress testing. The released artifact package is provided in our code repository.

B BASELINES

We compare our model against a range of established baselines:

- **DeepHawkes** (Cao et al., 2017): Models cascades as multiple diffusion paths and employs a GRU network to capture temporal dynamics in diffusion sequences.
- **CasCN** (Chen et al., 2019): Represents cascades as evolving graph sequences and uses a GNN-LSTM architecture to learn dynamic cascade representations.
- **CasFlow** (Xu et al., 2021): A state-of-the-art approach that learns user representations from both social and cascade graphs, and generates cascade embeddings using a GRU-based sequential model combined with a Variational Autoencoder (VAE).
- **CTCP** (Lu et al., 2023): A continuous-time graph learning framework that models inter-cascade dependencies by maintaining dynamic user and cascade representations. It encodes diffusion events as messages and updates node states through an evolutionary learning module with recurrent fusion.
- **CasDo** (Cheng et al., 2024): Integrates probabilistic diffusion models to capture uncertainty in information spread, by injecting noise during forward propagation and reconstructing cascade embeddings via a reverse denoising process.
- **VNOIP** (Wang et al., 2026): A variational neural ODE-based method that jointly models cascade sequences and popularity-trend trajectories in continuous time. In our benchmark it serves as a strong recent baseline representing trend-aware temporal forecasting.

C DATA PRIVACY AND ETHICAL CONSIDERATIONS

We confirm that the Taoke dataset has undergone rigorous privacy protection processing to ensure ethical and legal compliance. Specifically, we have implemented the following measures: (1) User Anonymization: All personally identifiable information has been removed and user IDs have been replaced with irreversible pseudonymous tokens. (2) Content Sanitization: The dataset contains no raw text data (e.g., specific production brand) and focuses solely on structured behavioral records (e.g., timestamps, promotions, transactions). (3) Feature Desensitization: Auxiliary features (e.g., product titles) are provided as processed embeddings, preventing reverse-engineering of sensitive semantics. (4) Institutional Review: The dataset and collection methodology were reviewed and approved by the Legal and Security Department of the data source company, confirming no risk of sensitive information leakage. These steps ensure the dataset maintains research utility while strictly preserving user privacy and meeting industrial security standards.

D Extended Motivation and Figure Interpretation

This appendix provides a detailed roadmap and interprets the preview signals shown in Figure 1, complementing the protocol definitions and benchmark descriptions in the main text.

Figure 1 gives a compact preview aligned with our three-part framework. *(Left)* Protocol correction reshapes both ranks and apparent difficulty: on APS, the top method changes from CasFlow under Random Cascade to CasTemp under Full Temporal and Inductive User, showing that even the winner depends on the split definition. *(Right-top)* Leakage remains substantial under conventional protocols: on Taoke, Random Cascade and Full Temporal preserve 79.9% and 63.3% train–test user overlap respectively, whereas Inductive User removes it entirely. *(Right-middle)* This leakage translates into significant performance inflation: compared to Full Temporal, APS scores under Random Cascade are inflated by up to 151.3% for CasFlow. *(Right-bottom)* The underlying data exhibits strong non-stationarity: the composite temporal-drift score reaches 0.482 on Twitter and 0.446 on Weibo, making random partitioning difficult to justify as a default.

Beyond the released benchmark, we use a larger internal extension, Taoke-large, that preserves the same semantics and task definitions while scaling up the product pool; its role is limited to operational stress testing rather than released benchmark comparison. Meanwhile, the reference pipeline remains practical under the stricter benchmark, achieving 139×–412× training speedups and 375×–2457× inference speedups over the fastest baseline competitors on the released benchmark, and successfully completing on Taoke-large where baseline systems run out of budget.

Overall, the main text first examines the evaluation protocols and shows why benchmark conclusions can change once chronology and leakage channels are treated explicitly. We then introduce the richer Taoke benchmark asset and its expanded task scope. Finally, we use the lightweight reference pipeline to study what changes under the resulting benchmark specification, including cross-model ranking shifts, transfer behavior on the added conversion task, and the cost of maintaining a practical default standard. This appendix supplements that narrative with quantitative evidence and operational details that support the claims made in the introduction.

E Evidence of Information Leakage in Random Cascade Splits

Optimistic cascade-level splits. Most prior cascade prediction papers adopt a random cascade split, typically assigning cascades to train/validation/test partitions without enforcing a global temporal boundary (Xu et al., 2021; Lu et al., 2023; Cheng et al., 2024). This is problematic because the target itself is future growth. When train and test cascades are sampled from overlapping periods, models may indirectly access synchronized bursts, overlapping users, or future-era activity patterns that would not be available in an actual forecasting pipeline. For rigorous benchmarking, this matters because even a small amount of leakage can distort both absolute scores and relative model rankings.

Toy evidence of protocol-induced shortcuts. To make the issue concrete and intuitively verify the presence of information leakage under Random Cascade split, we construct a toy benchmark illustrated in Figure 4. It contrasts two evaluation views of the same diffusion process: the conventional Random Cascade-level partition and a Full Temporal-style globally time-ordered split. Specifically, we set the observation window to one unit of time (gray dashed box) and define the incremental popularity up to the cutoff as the prediction target. We design two scenarios to expose the failure mode: In Scenario 1, the test cascade c_1 (orange) shares similar propagation dynamics with the training cascade c_0 (blue), including identical burst patterns between T_1–T_2 and decay phases from T_2–T_3. Conversely, in Scenario 2, the test cascade c_2 (green) exhibits fundamentally different dynamics from c_0, originating from a distinct node (u_{12} vs. u_0) with misaligned temporal phases.

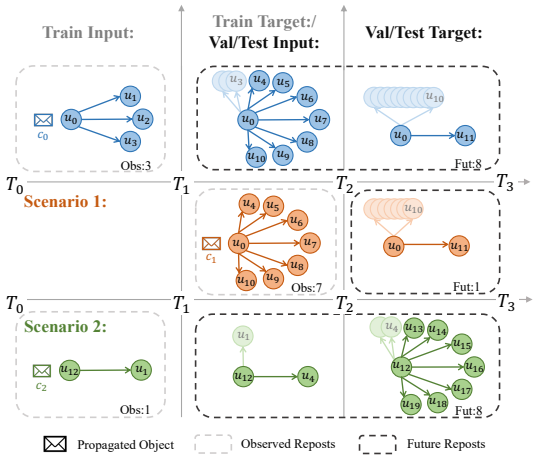

Figure 4: Toy illustration of protocol sensitivity. Horizontally: conventional Random Cascade splitting with per-cascade observation and target windows. Vertically: a Full Temporal-style globally time-ordered split that separates train/validation/test by consecutive time segments.

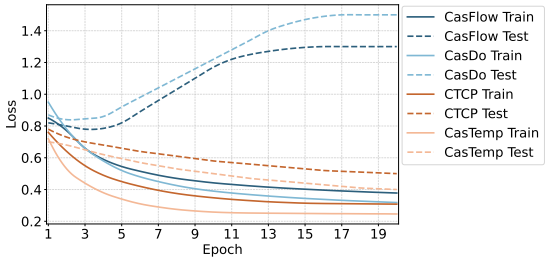

Figure 5: Training and test loss under the time-ordered toy split.

Table 8: Toy benchmark under the conventional cascade-based split. Lower is better.

Model	Scen. 1		Scen. 2	
	MSLE	MALE	MSLE	MALE
MLP	6.3621	2.5498	0.4510	0.6731
DeepHawkes	5.5863	2.0954	0.6449	0.7812
CasCN	5.4622	1.9477	0.7831	0.8502
CasFlow	5.8906	1.9985	0.6251	0.9053
CTCP	5.1063	2.3519	0.8546	0.9415
CasDo	6.0415	2.2980	0.6831	0.9392
VNOIP	5.5802	2.0474	0.6247	0.9010
CasTemp	5.4821	2.0891	0.7500	0.9044

Table 9: Toy benchmark under the corrected Full Temporal-style split. Lower is better.

Model	Scen. 1		Scen. 2	
	MSLE	MALE	MSLE	MALE
MLP	2.9078	1.8136	2.9884	1.5250
DeepHawkes	1.6632	1.3905	2.2501	1.3193
CasCN	2.9038	1.7192	2.4778	1.5804
CasFlow	4.6893	2.1320	4.9673	2.0780
CTCP	1.9387	1.5913	1.8842	1.3765
CasDo	5.6574	2.4400	4.6387	2.1285
VNOIP	3.7326	1.9001	2.0455	1.1901
CasTemp	1.6104	1.4027	1.8661	1.5412

Under genuine forecasting semantics, a model trained on c_0 should perform well on the similar c_1 but struggle with the dissimilar c_2. However, under the Random Cascade protocol, future information is blurred into a shared pool, causing models to behave inconsistently with these intended semantics: they may exploit temporal shortcuts to perform poorly on apparently similar cases (Scenario 1) while achieving artificially high accuracy on dissimilar ones (Scenario 2). Notably, even the simplest MLP achieves performance comparable to sophisticated baselines in Table 8, suggesting that the apparent gains of complex models are largely driven by leaked signals rather than genuine predictive capacity. We generate 1,000 synthetic cascades following this construction (70%/30% train-test split under the Random Cascade protocol). The results in Table 8 should therefore be read not as a standalone benchmark, but as motivation for protocol revision, demonstrating that standard splits can invalidate evaluation even when test cascades appear semantically hard.

Correcting the split changes the qualitative conclusion. The corrected toy results in Table 9 show the expected qualitative change. Once the protocol is forecasting-consistent, over-parameterized models lose much of their apparent advantage, while lightweight methods generalize more stably; notably, several baselines now underperform even the simple MLP, whose relative standing improves markedly compared to Table 8. The loss curves in Figure 5 further suggest that some previously strong methods are sensitive to shortcut-driven overfitting. This toy diagnosis is useful because it isolates the failure mode; the next section shows that the same issue reappears at benchmark scale.

E.1 Definitions of the Evaluation Protocols

Notation. Let $\mathcal{C}$ denote the set of cascades and let t_0^c be the creation time of cascade c. For any time interval $I = [a, b)$, we write $G^c[I]$ for the cascade-c subgraph induced by diffusion events whose timestamps lie in I, and

$$N_c[I] = |\{(src_i, tgt_i, c_i, t_i) : c_i = c,\ a \le t_i < b\}|$$

for the number of cascade-c diffusion events in that interval. For protocols that split at the cascade level, every cascade uses the same local observation and target windows:

$$x^c = G^c[t_0^c, t_0^c + \Delta t_1), \qquad y^c = N_c[t_0^c + \Delta t_1, t_0^c + \Delta t_2).$$

The distinction between protocols therefore lies in how the cascade set is partitioned and in which overlap channels remain accessible.

Protocol family. We study three protocols. **(i) Random Cascade** partitions $\mathcal{C}$ uniformly at random into $\mathcal{C}_{\mathrm{tr}}, \mathcal{C}_{\mathrm{va}}, \mathcal{C}_{\mathrm{te}}$ and applies the local (x^c, y^c) definition above. For the recommended **(ii) Full Temporal** protocol, we choose global boundaries $\beta_0 < \beta_1 < \beta_2 < \beta_3 < \beta_4$ and define:

$$\mathcal{D}_q^{\mathrm{FT}} = \left\{ \left(G^c[\beta_{q-1}, \beta_q),\ N_c[\beta_q, \beta_{q+1})\right) :\ c \in \mathcal{C},\ N_c[\beta_{q-1}, \beta_q) > 0 \right\},$$

for $q \in \{1, 2, 3\}$, where $q = 1, 2, 3$ correspond to train, validation, and test. Finally, **(iii) Inductive User** keeps the same global windows but additionally requires:

$$U(\mathcal{D}_{\mathrm{train}}^{\mathrm{IU}}) \cap U(\mathcal{D}_{\mathrm{test}}^{\mathrm{IU}}) = \varnothing,$$

where $U(\cdot)$ denotes the set of diffusion users appearing in a split. In terms of benchmark role, Random Cascade serves as the optimistic historical baseline, Full Temporal is our main reporting protocol because it preserves forecasting semantics while remaining executable on all datasets, and Inductive User is best interpreted as a stress test for user-transfer reliance rather than as the default setting.

F Analysis of the Evaluation Protocols

Protocol sensitivity at benchmark scale. The results in Figure 3 show that the real benchmark behaves fundamentally differently under strict protocols. For example, on APS, the best random-split baseline is CasFlow, but once chronology is enforced the winner changes to CasTemp and the entire runner-up group is reordered. Twitter, Weibo, and Taoke keep the same winner across the three non-degenerate protocols, but their second and third positions still change materially, indicating that ranking stability cannot be assumed from a single split.

Leakage is multi-channel rather than scalar. Different protocols suppress different shortcut channels. Random Cascade removes item reuse across splits but still keeps non-trivial user overlap. Full Temporal enforces chronological forecasting, yet it intentionally preserves item continuity because the same item

Table 10: Protocol properties extracted from the actual split statistics. ρ_u denotes train–test user overlap.

Dataset	Train–Test User Overlap (ρ_u)		
	Random Cascade	Full Temporal	Inductive User
Twitter	0.257	0.285	0.000
Weibo	0.170	0.209	0.000
APS	0.072	0.175	0.000
Taoke	0.799	0.633	0.000

may remain active across adjacent global windows; in fact, the train–test item overlap ratio is 1.0 on all four datasets under this protocol. Inductive User removes train–test user reuse by construction. Table 10 quantifies these differences.

The preferred protocol need not minimize every scalar leakage statistic. Because Full Temporal preserves forecasting-faithful item continuity, its aggregate leakage score is slightly higher than Random Cascade on Twitter, Weibo, and APS. This reflects the benchmark design choice that removing temporal invalidity is more important than forcing all overlap channels to zero. Moderate user overlap also reflects real deployment conditions where long-active users naturally span training and test periods. Inductive User remains useful precisely because it isolates the user-transfer channel that Full Temporal intentionally leaves available.

Detailed performance inflation case study. As highlighted in the main text, inflation is particularly striking on APS. Table 11 provides the complete breakdown. Relative to Full Temporal, Random Cascade improves every APS method by a large margin, serving as our clearest evidence of how optimistic protocols materially overstate progress.

Table 11: APS case study for performance inflation when moving from Random Cascade to Full Temporal. Positive inflation means the random split reports an easier (overly optimistic) result.

Model	Random Cascade	Full Temporal	Inflation
CasCN	1.192	2.496	+109.3%
CasDo	2.081	4.839	+132.6%
CasFlow	1.058	2.659	+151.3%
CasTemp	1.066	1.924	+80.5%
CTCP	1.356	2.802	+106.7%
DeepHawkes	1.431	2.524	+76.4%
VNOIP	1.740	2.399	+37.9%

Detailed temporal drift profiles. Even if one ignores direct overlap contamination, the data are not temporally stationary. Figure 6 summarizes adjacent-window drift across all four datasets. Twitter and Weibo exhibit the highest average composite drift, while Taoke is more stable in aggregate but still far from IID: its user and cascade identities are persistent, yet burstiness and conversion-relevant activity still shift over time. This is precisely the setting in which random cascade mixing of windows becomes hard to justify.

As shown in Table 12, Twitter and Weibo have the weakest user continuity, with average adjacent-window user Jaccard scores of only 0.157 and 0.101, respectively, which helps explain why random mixing can easily blur distinct activity eras. APS shows lower composite drift overall, but its user and cascade identities still turn over enough to violate a naive IID interpretation. Taoke is different again: its user and cascade identities are substantially more persistent, yet its average burstiness JSD is the largest among the four datasets. Looking back at the underlying window statistics, this comes from genuine popular business-cycle variation: the average cascade size rises from 51.8 in window 1 to 71.8 in window 4, while active users

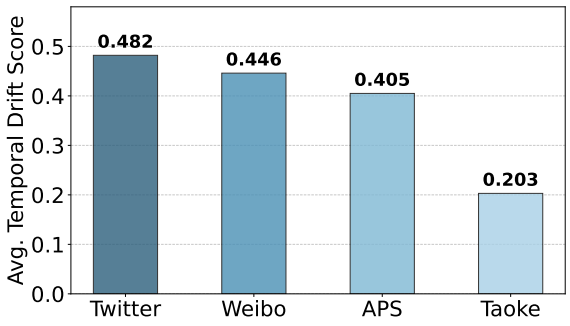

Figure 6: Average adjacent-window temporal drift across datasets. Drift is computed as the mean of $(1 -$ user Jacc.$)$, $(1 -$ cascade Jacc.$)$, and burstiness JSD. Higher bars indicate stronger non-stationarity.

Table 12: Drift profiles averaged over adjacent windows. User/cascade Jacc. measure identity continuity between consecutive windows, and burstiness JSD measures temporal-pattern shift. Higher Jaccard indicates greater continuity; higher JSD indicates stronger pattern change.

Dataset	Per-Metric Drift Components			
	User Jacc.	Cascade Jacc.	Burst. JSD	Avg. Drift
Twitter	0.157	0.403	0.0049	0.482
Weibo	0.101	0.564	0.0021	0.446
APS	0.257	0.529	0.0005	0.405
Taoke	0.498	0.916	0.0223	0.203

increase from 16.1K to 20.0K. In other words, Taoke is temporally smoother in who participates, but still non-stationary in how strongly campaigns propagate.

G Evaluation and Reporting Notes

For reproducibility, we note that the artifact label 'benchmark_default' corresponds to *Full Temporal*; the main paper reports popularity results under this protocol, while Random Cascade and Inductive User are used mainly for benchmark diagnosis. The Taoke conversion table is a task-specific subset rather than a duplicate of the full popularity roster. Taoke-large shares the same semantics and task definitions as Taoke, but it is a non-public internal extension used only for large-scale efficiency stress testing, not as a second full benchmark leaderboard. The released artifact exposes the released data, the split scripts, the training/evaluation scripts, and the analysis pipeline for the released benchmark asset.

H Reference Method Details

Competition graph and temporal sequences. Given the dynamic cascade graph in Section 3, CasTemp first constructs an inter-cascade competition graph $\mathcal{G}_c = (\mathcal{C}, \mathcal{E}_c, \mathbf{w}_c)$. Each node is a cascade, and the edge weight between cascades c_i and c_j is the Jaccard overlap of their promoter sets:

$$w_{ij} = \frac{|U_i \cap U_j|}{|U_i \cup U_j|}.$$

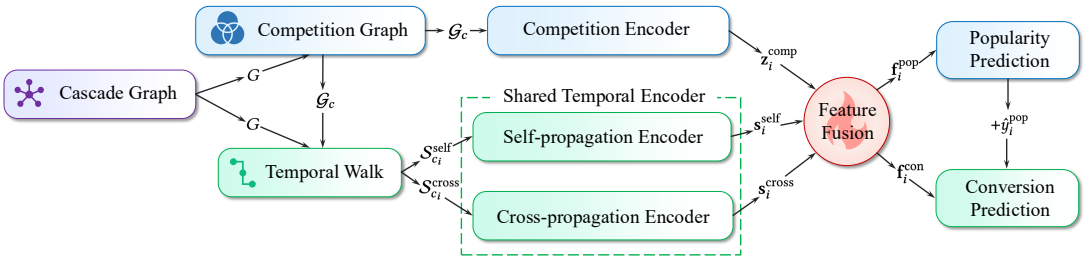

Figure 7: CasTemp as a benchmark reference pipeline. It combines a cascade competition graph, self- and cross-cascade temporal sequences, and lightweight predictors for popularity and conversion forecasting.

Only edges above a threshold τ_1 are retained. This graph provides a lightweight proxy for competition or collaboration among cascades and is simple to update through set maintenance.

For each target cascade c_i, CasTemp then extracts two ordered sequences: a self-propagation sequence $\mathcal{S}_{c_i}^{\text{self}}$ containing the recent events of c_i, and a cross-propagation sequence $\mathcal{S}_{c_i}^{\text{cross}}$ sampled from neighboring cascades in $\mathcal{G}_c$ by temporal random walks. The former describes the cascade's own growth trajectory, while the latter captures nearby external activity without introducing a heavy graph-diffusion module.

Temporal sequence encoder. Each event is represented by promoter features together with a sinusoidal encoding of normalized timestamps. To reflect the stronger predictive value of recent behavior, CasTemp applies an exponential decay prior

$$\gamma_k = \exp\left(-\lambda(t_{\max} - t_k)\right),$$

where $t_{\max}$ is the latest timestamp in the sequence. The decay-weighted events are encoded by a GRU (Dey & Salem, 2017) with additive attention, yielding $\mathbf{s}_i^{\text{self}}$ and $\mathbf{s}_i^{\text{cross}}$ for the self and cross sequences, respectively. This keeps the temporal module compact while making the recency bias explicit.

Competition-aware cascade representation. CasTemp further applies a lightweight GAT (Veličković et al., 2017) layer on $\mathcal{G}_c$ to produce a competition-aware cascade embedding $\mathbf{z}_i^{\text{comp}}$. This step is intentionally shallow: it is not meant to be a heavy graph learner, only a compact summary of the local inter-cascade context associated with c_i.

Popularity predictor. For first-stage popularity forecasting, CasTemp concatenates the competition-aware embedding, the self- and cross-sequence summaries, historical popularity features, and dataset-specific side information:

$$\mathbf{f}_i^{\text{pop}} = \left[\mathbf{z}_i^{\text{comp}}; \mathbf{s}_i^{\text{self}}; \mathbf{s}_i^{\text{cross}}; \mathbf{h}_i^{\text{his}-\text{pop}}; \mathbf{h}_i^{\text{fused}}\right]. \tag{1}$$

An MLP with Softplus output then predicts non-negative incremental popularity:

$$\hat{y}_i^{\text{pop}} = \text{MLP}_{\text{pop}}(\mathbf{f}_i^{\text{pop}}).$$

The simple head guarantees that the goal of testing whether a compact representation already suffices under the benchmark's stricter evaluation setting.

Conversion predictor. On Taoke, the same reference pipeline is extended to second-stage conversion forecasting. The conversion predictor reuses the same structural and temporal representations, then augments them with historical conversion signals and the predicted popularity:

$$\mathbf{f}_i^{\text{con}} = \left[\mathbf{z}_i^{\text{comp}}; \mathbf{s}_i^{\text{self}}; \mathbf{s}_i^{\text{cross}}; \mathbf{h}_i^{\text{his}-\text{con}}; \mathbf{h}_i^{\text{fused}}; \hat{y}_i^{\text{pop}}\right]. \tag{2}$$

The final prediction is

$$\hat{y}_i^{\text{con}} = \text{MLP}_{\text{con}}(\mathbf{f}_i^{\text{con}}).$$

As illustrated in Figure 7, this keeps the second-stage task tightly coupled to the first-stage setup and makes Taoke useful as a benchmark for propagation-to-conversion modeling rather than an unrelated auxiliary task.

I Implementation Details and Hyperparameters

All models are trained for up to 100 epochs with early stopping patience of 30, and any single run exceeding 48 hours is forcibly terminated and evaluated at its last checkpoint. Baselines are integrated from their original open-source implementations with published hyperparameters into our unified codebase to ensure fair comparison. CasTemp is trained with learning rate 0.01; the sequence encoder uses hidden dimension 16, temporal decay coefficient $\lambda = 0.1$, self-sequence length $L_{\max} = 10$, competition-graph threshold $\tau_1 = 0.1$, and temporal random walk budget $(\tau_2, \tau_3) = (5, 10)$. These settings are chosen to balance efficiency and stability and are further diagnosed in Section J. All methods are evaluated on the same released splits and reporting windows. Experiments are conducted on an NVIDIA RTX A6000 48GB GPU.

J Ablation Study

Figure 9 studies the sensitivity of four core hyperparameters on CasTemp. The results suggest that the method is reasonably stable across several nearby settings, with $\lambda = 0.1$ and $(L_{\max}, \tau_2, \tau_3) = (10, 5, 10)$ providing a good efficiency-accuracy trade-off. This matters because it indicates that the reference method is not winning only through brittle hyperparameter tuning.

We also report a component-level ablation in Figure 10. Its role is supportive rather than central: it confirms that the competition graph, cross-cascade sequence construction, and temporal decay mechanism each contribute non-trivially to the reference pipeline's effectiveness, so the model's behavior is not driven by one isolated trick.

K Additional Dataset Characterization

To further illustrate the structural heterogeneity across benchmarks, we present the cascade depth distributions for all four datasets in Figure 11.

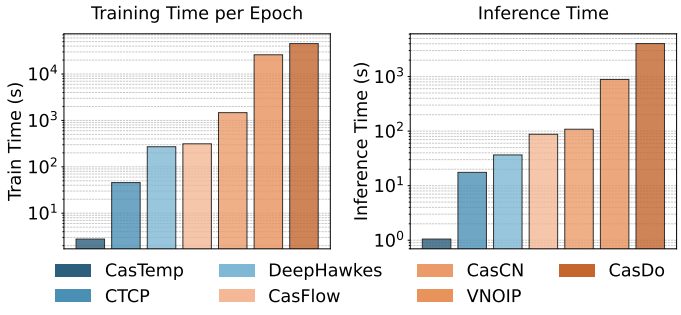

Figure 8: Training time per epoch and inference time of each method on Twitter.

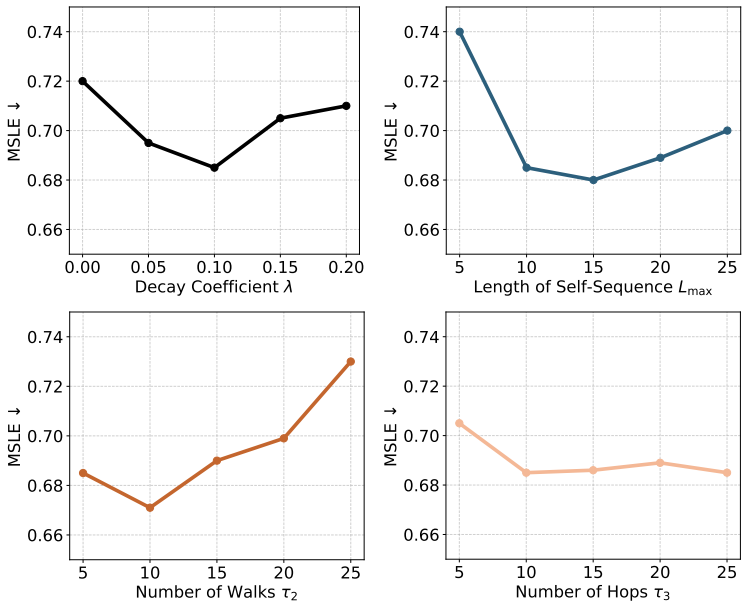

Figure 9: Hyperparameter analysis of the CasTemp reference method on Taoke.

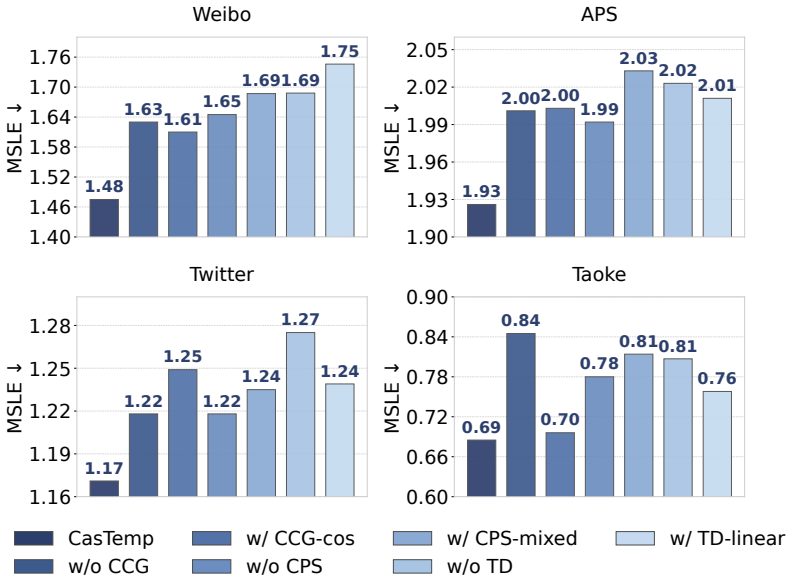

Figure 10: Ablation of key components in the CasTemp reference method.

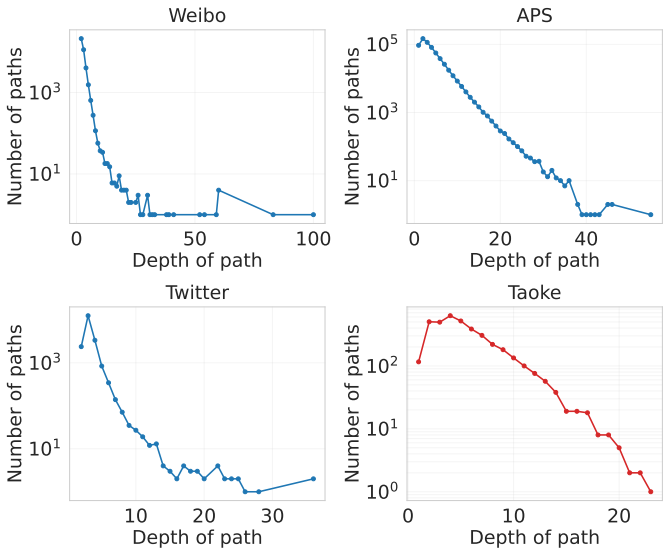

Figure 11: Distribution of cascade-depth statistics across datasets.

L Additional Experimental Results

To assess computational efficiency, we report both the training time per epoch and inference latency for each method on the Twitter dataset in Figure 8.

For completeness, we provide the full numerical MSLE results corresponding to the ranking matrix in Figure 3. The detailed performance of all seven baselines under the three evaluation protocols (R: Random Cascade, F: Full Temporal, I: Inductive User) on each dataset is reported in Tables 13 to 16 for Twitter, Weibo, APS, and Taoke, respectively. These tables enable direct verification of the rank assignments and facilitate fine-grained comparison beyond the top-3 summary presented in the main text.

Table 13: Full MSLE results on Twitter across three protocols. Top-3 are color-coded matching Figure 3.

Model	**R**	**F**	**I**
CasTemp	1.2353	1.1685	1.0801
CasCN	1.2418	1.1939	1.0997
DeepHawkes	1.2724	1.2702	2.0850
VNOIP	1.3280	1.2210	1.1127
CasFlow	1.3035	1.2401	1.1252
CTCP	1.5688	1.4474	1.1085
CasDo	1.9118	2.1014	2.3206

Table 14: Full MSLE results on Weibo across three protocols. Top-3 are color-coded matching Figure 3.

Model	R	F	I
CasTemp	1.5777	1.4783	1.5586
CasFlow	1.6268	1.6332	1.9174
CTCP	1.7932	1.8832	1.6359
CasCN	1.6669	2.5874	1.7878
DeepHawkes	1.8345	1.7295	1.7298
VNOIP	2.4950	2.4779	1.8095
CasDo	2.9273	2.1425	4.4178

Table 15: Full MSLE results on APS across three protocols. Top-3 are color-coded matching Figure 3.

Model	R	F	I
CasFlow	1.0585	2.6594	1.8379
CasTemp	1.0662	1.9240	1.3914
CasCN	1.1924	2.4956	2.6877
VNOIP	1.7396	2.3991	1.4960
CTCP	1.3559	2.8021	1.8317
DeepHawkes	1.4312	2.5240	2.3433
CasDo	2.0807	4.8391	4.7325

Table 16: Full MSLE results on Taoke across three protocols. Top-3 are color-coded matching Figure 3.

Model	R	F	I
CasTemp	1.0113	0.7984	0.6715
CasFlow	2.1328	2.2877	1.6461
CasCN	3.2639	3.4317	4.8026
DeepHawkes	4.3000	3.3632	2.9352
CTCP	6.3728	8.3837	1.5410
VNOIP	11.8504	6.5742	1.7713
CasDo	28.5780	27.1664	3.9857