Can LLMs Access their World Knowledge for Event Prediction?

Zijun Wu[1,2]*, **Zhitian Zhang**[1], **Hossein Hajimirsadeghi**[1]
Nikita Dvornik[1], **Alexander Pashevich**[1]
[1]RBC Borealis [2]University of Alberta
zijun4@ualberta.ca
{andy.zhang,nikita.dvornik}@rbc.com
{hossein.hajimirsadeghi,alexander.pashevich}@borealisai.com

Abstract

Recent work has questioned whether large language models (LLMs) can be useful for temporal event prediction. Despite vast pretrained world knowledge, LLMs often underperform simpler specialized models. We ask whether this knowledge is truly inaccessible, or whether current methods fail to present event prediction in a form LLMs can use. We argue for the latter and show that LLMs become effective event forecasters when the interface is designed around two choices. The first grounds each event in a natural-language description, and the second separates event semantics from the prediction space to prevent semantic bias from label associations. To test this, we introduce DESCRIBEEVENT, a benchmark of ∼150K events across seven datasets and six domains with event-level descriptions. We find that capable LLMs substantially benefit from descriptions and symbolic labels, outperforming description-free variants and non-LLM baselines, while weaker models gain little. These results suggest that accessing LLM world knowledge for event prediction is an interface problem, and that event descriptions offer a practical way to elicit the knowledge LLMs already encode.[1]

1 Introduction

Temporal event prediction studies how events evolve over time and asks what event will happen next, and when. This problem is typically modeled with temporal point processes (TPPs), which represent each event as a categorical type paired with a timestamp (Hawkes, 1971; Du et al., 2016; Mei and Eisner, 2017). Recent work are adapting LLMs to temporal event prediction, motivated by the assumption that their pretrained world knowledge can help infer the next event (Liu and Quan, 2024; Kong et al., 2026; Chen et al., 2025a; Shi et al., 2023; Gruver et al., 2023; Zhou et al., 2023; Requeima et al., 2024; Gupta et al., 2025). However, recent evidence challenges this view: LLM components can often be removed with little loss, and LLM-based forecasters often fail to beat simpler specialized models (Tan et al., 2024; Chang et al., 2024). This raises our central question: *is LLM knowledge inaccessible for event prediction, or have current methods failed to make it usable?*

We argue for the latter and identify two interface choices that, together, help LLMs use their pretrained knowledge for event prediction. First, we ground each event in a natural-language description. Standard event-prediction benchmarks represent events only as (`event_type`, `event_time`) pairs (Xue et al., 2024), stripping away the event semantics that LLMs could otherwise use. Yet in real settings, events often come with text that explains what happened, such as a commit message describing a code change. We therefore move to a more realistic setting: each event is supplemented with a text description that provides context and domain cues unavailable from categorical labels alone. This makes event prediction a natural inference task for LLMs, rather than a purely symbolic sequence-modeling problem.

The second choice is to decouple event semantics from the prediction space. Semantically loaded labels can bias LLMs toward pretrained label associations (Zhao et al., 2021; Reif and Schwartz, 2024) rather than the task context, an effect we call *semantic bias*. We mitigate this by keeping event semantics in the input descriptions, while using symbolic labels for prediction. A symbolic label is a non-semantic identifier such as `type_0` or `type_1` that carries no label-name signal. This removes semantic bias and pushes an LLM to predict the next event from the descriptions of past events. Together, these two choices expose event semantics where LLMs can use them and remove semantic bias where it hurts, effectively eliciting pretrained world knowledge for event prediction.

*Work done during an internship at RBC Borealis.
[1]Our code and dataset are available at https://github.com/BorealisAI/DescribeEvent.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 17442–17457
October 24-29, 2026 ©2026 Association for Computational Linguistics

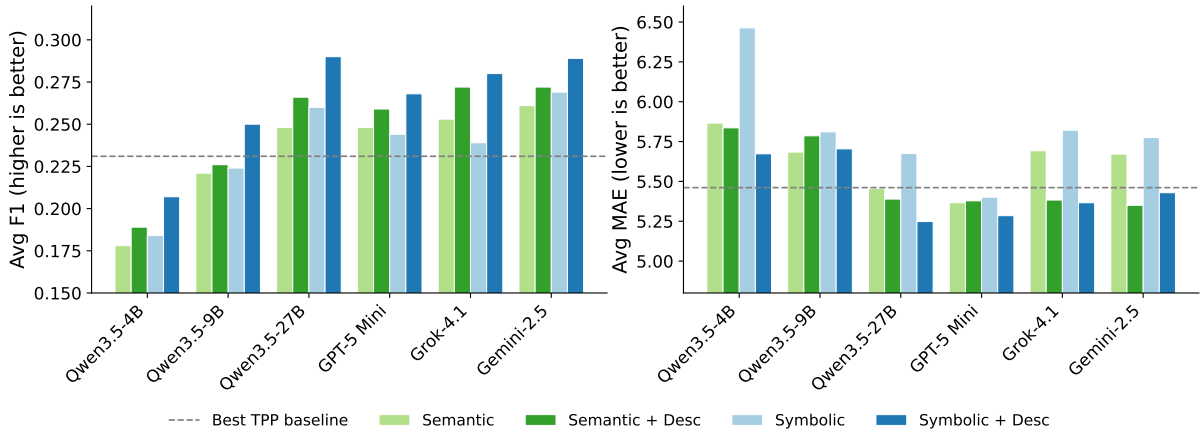


Figure 1: LLM Average performance across all datasets of DESCRIBEEVENT. Adding natural-language descriptions of events (as indicated by '+ Desc') consistently improves LLM performance, and pairing descriptions with using symbolic labels instead of semantic event names as prediction targets yields the largest gains. Dashed lines indicate the best non-LLM baseline for each metric.

To evaluate whether these choices help LLMs predict the next event, we introduce DESCRIBEEVENT, a benchmark of ∼150K events drawn from seven datasets across six domains, with time scales from minutes to weeks. While prior datasets attach at most one description per sequence (Xue et al., 2024; Liu and Quan, 2024), every event in DESCRIBEEVENT is paired with its own natural-language description. We evaluate six state-of-the-art LLMs under four settings, combining semantic or symbolic event labels with and without descriptions. We also compare against six non-LLM baselines.

Our experiments show that descriptions and symbolic labels are complementary: capable LLMs using both outperform description-free LLMs and non-LLM baselines on most datasets, as shown in Figure 1. Descriptions with semantic labels capture only part of the gain, while symbolic labels remove semantic bias and unlock the full benefit. These gains are largest for models with stronger reasoning ability (Rein et al., 2024; Phan et al., 2026), while weaker models gain little or sometimes degrade with richer context (Cao et al., 2026).

In summary, our main contributions include:

- We show that LLM knowledge is accessible for event prediction when the prediction interface exposes it properly.
- We identify per-event descriptions and symbolic labels as two interface choices that jointly produce the strongest and most consistent gains.
- We introduce DESCRIBEEVENT, a benchmark of ∼150K events across 7 datasets and 6 domains, and use it to show that the description benefit depends on model reasoning ability.

2 Related Work

We review prior work on temporal point processes, LLMs for time series, LLMs for event prediction, and the effect of label semantics on LLM predictions that motivates our use of symbolic labels.

Temporal Point Processes. Temporal point processes (TPPs) model sequences of discrete events in continuous time. The Hawkes process (Hawkes, 1971) introduced mutual excitation between event types and remains a widely used baseline for event prediction. Subsequent neural extensions replace parametric intensity functions with learned ones. RMTPP (Du et al., 2016) uses recurrent networks to encode event history, and the Neural Hawkes Process (Mei and Eisner, 2017) models continuous-time dynamics with a neural architecture. More recent work applies self-attention to event sequences. For example, the Transformer Hawkes Process (Zuo et al., 2020) and the Self-Attentive Hawkes Process (Zhang et al., 2020) replace recurrent encoders with attention mechanisms to model long-range dependencies. EasyTPP (Xue et al., 2024) standardizes evaluation across public datasets and represents each event as a type–timestamp pair. This format is compact, but it omits the natural language descriptions that often accompany real-world events.

LLMs for Time Series. Large language models have recently been adapted to temporal data.

Specifically, Time-LLM (Jin et al., 2024) reprograms LLMs for time series forecasting by aligning time series representations with language tokens. Chronos (Ansari et al., 2024) tokenizes time series values into a fixed vocabulary and trains transformer language models directly on them. Nevertheless, Tan et al. (2024) show that the LLM component can often be removed without hurting performance, challenging the assumption that pretrained language knowledge helps for temporal prediction. Similarly, Chang et al. (2024) find that zero-shot LLMs lag behind conventional methods for temporal event forecasting. Williams et al. (2025) offer a counterpoint and introduce a benchmark where textual context is essential for time series forecasting. They show that their LLM prompting outperforms statistical and foundation model baselines.

LLMs for Event Prediction. In LLM-based event prediction, TPP-LLM (Liu and Quan, 2024) fine-tunes LLMs on structured event records using the same type–timestamp format as classical TPPs. Several methods go further by incorporating textual information. LAMP (Shi et al., 2023) uses abductive reasoning with event metadata to let LLMs explain candidate future events proposed by a TPP predictor, Language-TPP (Kong et al., 2026) converts time intervals into byte-tokens that integrate descriptions with a language model, and LAST SToP (Gupta et al., 2025) designs specialized prompts with event descriptions for asynchronous time series. TESRBench (Liu and Quan, 2025) encodes event sequences and descriptions into a shared space for retrieval rather than next-event prediction. In a related direction, Cao et al. (2026) study whether additional context improves LLM temporal reasoning on a repurchase prediction task, and Tan et al. (2025) study whether LLMs can infer event descriptions from time series rather than use descriptions to improve prediction. Ye et al. (2024) introduce MIRAI, a benchmark for evaluating LLM agents on international event forecasting, but focus on agent tool use within a single domain rather than on interface design across domains.

Label Semantics in LLMs. Prior work suggests that LLMs can carry non-trivial priors over label tokens. For example, Zhao et al. (2021) show that in few-shot classification, LLMs tend to favor certain label strings regardless of the input, and propose calibration methods to correct this. Reif and Schwartz (2024) systematically measure and mitigate label bias across tasks. Wei et al. (2023) show that replacing natural language labels with arbitrary symbols, both during fine-tuning and at inference, improves in-context learning for NLP tasks. Our use of symbolic labels draws on this insight and extends it to event prediction, where symbolic event labels make descriptions the main source of event semantics rather than label priors.

3 Approach

In this section, we present two interface choices for improved LLM-based event prediction. We ground events in natural-language descriptions and decouple event semantics from prediction labels to reduce misleading pretrained label associations. Together, these choices expose useful context while preserving a cleaner prediction space.

3.1 Preliminaries

Next-event prediction. An event sequence is an ordered list of events observed in time. In the standard representation used by prior event-prediction work (Xue et al., 2024), each event is a pair of a timestamp and a categorical type,

$$e_i = (t_i, k_i), \tag{1}$$

where $t_i \in \mathbb{R}$ is the timestamp and $k_i \in \mathcal{K}$ is a categorical type drawn from a finite vocabulary $\mathcal{K}$. A sequence of length N is $S = (e_1, \ldots, e_N)$. The history prefix of length n is $H_n = (e_1, \ldots, e_n)$, and the next-event prediction task is to produce (t_{n+1}, k_{n+1}) given H_n.

Prediction with an off-the-shelf LLM. We serialize each event into text as "$[k_i]$ $+\Delta t_i$", where $\Delta t_i = t_i - t_{i-1}$ is the inter-event gap (Liu and Quan, 2024; Kong et al., 2026). The prompt pairs this serialized history with a system message, which states the task, lists the valid labels in $\mathcal{K}$, and asks the model to reason before returning the next event as `{event_type, time}`. At step n, the model produces

$$(r_{n+1}, \hat{t}_{n+1}, \hat{k}_{n+1}) = \mathrm{LLM}(H_n), \tag{2}$$

where r_{n+1} is model reasoning. We follow teacher forcing by appending the ground-truth e_{n+1} to the history before the next prediction (Mei and Eisner, 2017; Xue et al., 2024; Gupta et al., 2025).

3.2 Text Descriptions

Event types alone (typically used for event prediction) do not carry enough semantic content for LLMs to leverage their pretrained knowledge. An

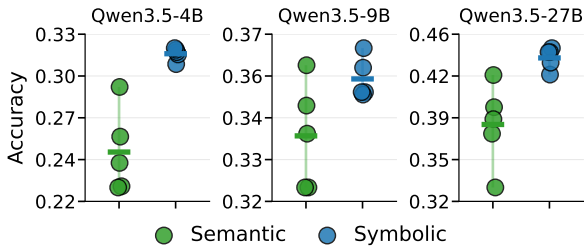


Figure 2: Accuracy under random label permutations. Semantic labels are permutation-sensitive, while symbolic labels are not.

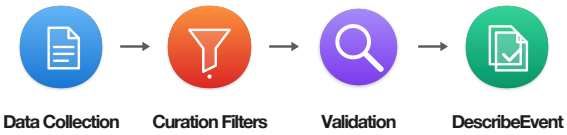


Figure 3: DESCRIBEEVENT data curation pipeline.

event type like push is merely a name with no domain context behind it. Without that context, the model has little information to ground its pretrained knowledge on, and works only from the label tokens themselves.

A per-event description surfaces this context in the model's input. We extend each event with a natural-language description d_i, so an event becomes a triple

$$e_i = (t_i, k_i, d_i). \tag{3}$$

For a code-repository event, d_i can mention the commit message, the number of commits, or the files changed. For a seismic event, it can mention the magnitude, depth, or location.

Descriptions enter only as input context, and the prediction target stays (t_{n+1}, k_{n+1}) as in Section 3.1. With this added context, each event in the history reads as text rather than the label name alone, so the model can apply its pretrained world knowledge to next-event prediction.

3.3 Removing Semantic Bias

Event type names are themselves natural-language tokens, and an LLM can carry sequential associations between such names from pretraining text (Zhao et al., 2021; Reif and Schwartz, 2024). For example, in code-repository text push often appears near `merge`, so an LLM expects `merge` after push before reading any event history. We call this effect *semantic bias*: the model predicts the next event from these pretrained label-name associations, not from the actual past events. As a result, the model can produce a plausible-sounding prediction from labels alone, ignoring the description even when the description carries decisive information.

We run a label-permutation test to expose semantic bias. Specifically, we omit descriptions so that the label sequence is the only source of signal, and we randomly relabel the type vocabulary across runs while leaving the underlying event sequence unchanged. For example, under semantic labels, push might be relabeled as `pull` and `commit` as push. Under symbolic labels, push maps to `type_1` in one run and `type_4` in another. As shown in Figure 2, symbolic accuracy is more stable than semantic accuracy across permutations. This gap reveals semantic bias, since the label names themselves carry predictive signal beyond structure.

To prevent semantic bias from competing with descriptions, we adopt a fixed bijective mapping ℓ_{sym} that rewrites each semantic type as an anonymous token. Specifically, the i-th event type in the vocabulary $\mathcal{K}$ is renamed to the string `type_i`, so push and `commit` become `type_0` and `type_1`. With semantic bias removed, the model is encouraged to rely on the description d_i to predict the next event.

4 The DESCRIBEEVENT Benchmark

We introduce DESCRIBEEVENT, a benchmark that pairs every event with a natural-language description, to test whether LLMs can use these descriptions for next-event prediction. We first describe the data collection, then the curation filters, and finally the validation diagnostics and dataset statistics. Figure 3 summarizes the full pipeline.

4.1 Data Collection

DESCRIBEEVENT covers six domains across seven datasets. As shown in Figure 4, it includes Amazon product reviews, USGS earthquake records, GDELT geopolitical interactions, NBA play-by-play, Wikipedia article edits, and GitHub activity under two different views, per repository and per user. These domains span a wide range of actors, from individuals to institutions to natural processes. The temporal granularities range from minutes to weeks, as shown in Figure 5. Details about the dataset sources and prediction tasks are provided in Table 4 in Appendix A.1.

Every event extends the EasyTPP format (Xue et al., 2024) with one text field for the description. Specifically, we inherit the timestamp t_i and event

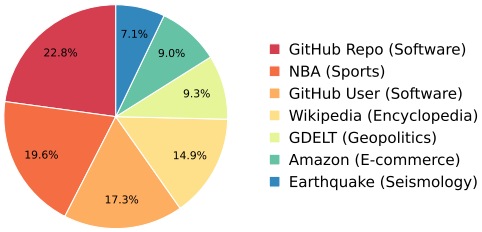


Figure 4: Dataset distribution of DESCRIBEEVENT by total events. The benchmark spans seven datasets across six diverse domains.

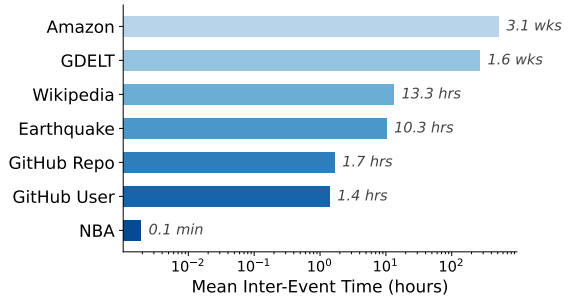


Figure 5: Mean inter-event time across datasets. DESCRIBEEVENT covers temporal granularities from minutes to weeks.

type k_i for each event, and add the natural-language description d_i defined in Section 3.2. Each description d_i is taken from the raw source text attached to the event at the time it occurred, mirroring a real-time collection setting: when an event and its description are recorded, later events have not yet happened. This temporal ordering prevents information about future events from entering earlier descriptions. The label space for k_i differs across datasets and is listed in Appendix A.2.

4.2 Curation Filters

We apply three universal filters to every dataset to improve the quality of each sequence, since a high-quality sequence should not be solvable by trivial baselines, such as always predicting the most common type or exploiting long runs of identical events. The three filters we applied are a *dominant-type cap* on the fraction of events sharing a single type, a *consecutive-repeat cap* on the fraction of adjacent events sharing the same type, and a *minimum type diversity* on the number of distinct types per sequence.

Additionally, each dataset passes through domain-specific filters to further improve the quality of the sequences, such as magnitude thresholds for earthquakes and bot-account exclusion for GitHub. Those per-dataset thresholds are detailed in Appendix A.3.

	Recoverability			Similarity		
Dataset	Random	R	Δ_{gain}	$\text{cos}_{\text{other}}$	cos_{true}	Δ_{leak}
GitHub Repo	0.125	0.905	0.780	0.114	0.187	0.073
GitHub User	0.125	0.897	0.772	0.122	0.228	0.106
Amazon	0.100	0.904	0.804	0.155	0.358	0.203
Earthquake	0.125	0.884	0.759	0.257	0.388	0.131
GDELT	0.143	0.611	0.468	0.112	0.227	0.115
NBA	0.111	0.992	0.881	0.249	0.399	0.150
Wikipedia	0.167	0.602	0.435	0.233	0.351	0.118

Table 1: Description validation diagnostics across the seven datasets in DESCRIBEEVENT. Δ_{gain} measures recovery above chance, where higher is better. Δ_{leak} measures label leakage, where lower is better and indicates that descriptions do not paraphrase labels.

4.3 Validation and Statistics

We validate each dataset with two diagnostics. The first is *Recoverability*, denoted by R. It is the accuracy of a validator LLM (Gilardi et al., 2023) predicting the type from the description alone. High R shows that the description is informative enough to identify the event type. As shown in Table 1, R exceeds random guess by a large margin on every dataset.

To check that a high R does not come from descriptions that merely paraphrase the labels, we report a second diagnostic, *Similarity*. For each description, we compute two cosine similarities under a sentence encoder (Reimers and Gurevych, 2019), cos_{true} against its true label name and $\text{cos}_{\text{other}}$ against the other labels. Descriptions that paraphrase their labels would yield a large gap. As shown in Table 1, this gap is small on every dataset. Together with the large recoverability gain above, this shows that descriptions in DESCRIBEEVENT carry event content beyond what the label name already provides, as intended in Section 3.2.

After the above curation processes, DESCRIBEEVENT contains 148,185 events organized into 1,940 sequences across seven datasets. Detailed dataset statistics are reported in Table 5.

5 Experiments

5.1 Experimental Setup

Choice of LLMs. In our main experiments, we evaluate six large language models that span open and closed weights and a range of parameter scales. From the open-source Qwen3.5 family (Yang et al., 2025), we include three sizes at 4B, 9B, and 27B

Model	GitHub Repo		GitHub User		Amazon		Earthquake		GDELT		NBA		Wikipedia	
	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓
Semantic Event Type														
Qwen3.5-4B	0.226	3.002	0.206	2.327	0.140	6.728	0.150	0.406	0.141	10.303	0.191	0.128	0.190	18.171
Qwen3.5-9B	0.290	2.832	0.245	1.830	0.148	6.622	0.167	0.370	0.133	10.409	0.262	0.092	0.304	17.634
Qwen3.5-27B	0.349	2.334	0.305	1.563	0.159	5.959	0.225	0.365	0.137	10.188	0.278	0.082	0.280	17.696
GPT-5 Mini	0.332	2.252	0.317	1.473	0.131	6.012	0.220	0.344	0.146	10.139	0.281	0.075	0.311	17.268
Grok-4.1	0.342	2.487	0.328	1.626	0.149	6.612	0.187	0.345	0.168	10.112	0.307	0.077	0.287	18.585
Gemini-2.5	0.351	2.295	0.309	1.585	0.168	5.957	0.231	0.350	0.156	10.201	0.320	0.077	0.292	19.233
Symbolic Event Type														
Qwen3.5-4B	0.216	3.709	0.232	2.893	0.117	7.005	0.174	0.379	0.150	10.715	0.163	0.482	0.233	20.061
Qwen3.5-9B	0.281	2.947	0.265	1.990	0.157	6.507	0.214	0.402	0.161	10.748	0.216	0.438	0.271	17.647
Qwen3.5-27B	0.352	2.422	0.327	1.570	0.161	5.975	0.235	0.367	0.149	10.816	0.282	0.250	0.315	18.323
GPT-5 Mini	0.336	2.353	0.301	1.537	0.142	6.021	0.211	0.348	0.148	10.187	0.264	0.094	0.305	17.268
Grok-4.1	0.358	2.518	0.306	1.669	0.139	6.815	0.175	0.353	0.150	10.556	0.285	0.086	0.261	18.749
Gemini-2.5	0.349	2.329	0.338	1.568	0.183	6.123	0.229	0.356	0.163	10.597	0.308	0.083	0.311	19.372
Semantic Event Type ***w/ Description***														
Qwen3.5-4B	0.229	2.844	0.182	1.949	0.127	6.095	0.207	0.397	0.120	12.057	0.236	0.106	0.225	17.406
Qwen3.5-9B	0.307	2.724	0.246	1.790	0.111	6.600	0.194	0.415	0.156	10.706	0.282	0.089	0.289	18.181
Qwen3.5-27B	0.353	2.332	0.288	1.489	0.160	5.806	0.219	0.359	0.160	10.392	0.317	0.078	0.362	17.268
GPT-5 Mini	0.330	2.445	0.277	1.647	0.157	5.841	0.252	0.345	0.151	10.087	0.316	0.072	0.330	17.206
Grok-4.1	0.344	2.459	0.285	1.576	0.169	6.084	0.238	**0.334**	0.158	10.073	0.329	0.072	0.378	17.082
Gemini-2.5	0.333	2.332	0.283	1.489	**0.187**	5.827	0.240	0.336	0.173	**9.539**	0.359	0.072	0.329	17.852
Symbolic Event Type ***w/ Description***														
Qwen3.5-4B	0.263	2.452	0.222	1.808	0.155	5.922	0.185	0.406	0.168	10.985	0.227	0.112	0.229	18.024
Qwen3.5-9B	0.310	2.774	0.261	1.700	0.164	6.118	0.206	0.391	0.173	10.266	0.294	0.080	0.340	18.601
Qwen3.5-27B	**0.383**	2.277	0.340	**1.453**	0.171	5.801	**0.255**	0.362	0.158	10.154	0.344	0.074	**0.379**	**16.615**
GPT-5 Mini	0.372	**2.237**	**0.342**	1.459	0.149	**5.758**	0.235	0.348	0.156	10.058	0.288	0.077	0.331	17.056
Grok-4.1	0.355	2.365	0.323	1.526	0.177	6.087	0.242	**0.334**	0.156	10.289	0.341	**0.071**	0.365	16.893
Gemini-2.5	0.361	2.295	0.324	1.492	0.174	5.851	0.242	0.349	**0.182**	10.080	**0.369**	**0.071**	0.368	17.864

Table 2: LLM performance on DESCRIBEEVENT. Best results per dataset are in **bold**.

parameters. We also include three close-source models, GPT-5 Mini (OpenAI, 2025), Grok-4.1 Fast (xAI, 2025), and Gemini-2.5 Flash (Comanici et al., 2025). In our analysis, we probe how reasoning ability shapes the benefit of descriptions, and additionally include two less reasoning-capable models, Llama-3-8B (Grattafiori et al., 2024) and GPT-4o-mini (Hurst et al., 2024).

Inference with LLMs. We evaluate the four settings defined by two binary factors (with/without description and symbolic/semantic labels) as described in Section 3. We extend the base serialization from Section 3.1 so that each event is written as "$[k_i] + \Delta t_i \mid d_i$". In the symbolic setting, k_i is replaced by $\ell_{\text{sym}}(k_i)$, and in the description-off settings, "$\mid d_i$" is removed. All four settings share the same prompting protocol, including the brief reasoning step before the final answer, so they differ only in how events are represented. One example per setting is given in Figure 9 in Appendix D. Additional inference configuration is detailed in Appendix C.

Evaluation Metrics. We evaluate predictions on next event type and time. For type, we report the macro-averaged F1 over the type vocabulary $\mathcal{K}$,

$$\text{Macro-F1} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \text{F1}_k, \tag{4}$$

where F1_k is the harmonic mean of precision and recall for type k. For time, we report Mean Absolute Error on the predicted inter-event gap, in each dataset's native time unit as detailed in Table 5,

$$\text{MAE} = \frac{1}{|\mathcal{P}|} \sum_{(s,n) \in \mathcal{P}} \left| \widehat{\Delta t}_{n+1}^{(s)} - \Delta t_{n+1}^{(s)} \right|, \tag{5}$$

where $\mathcal{P}$ is the set of (sequence, position) pairs predicted across the test set.

5.2 Performance of Large Language Models

Table 2 reports per-dataset F1 and MAE for each LLM on DESCRIBEEVENT, and Figure 1 shows each model's average across the seven datasets.

We first observe that adding descriptions to semantic event labels generally improves performance. Table 2 shows that F1 increases and MAE drops for nearly every LLM when descriptions are added to semantic labels. This indicates that descriptions give the model information beyond what

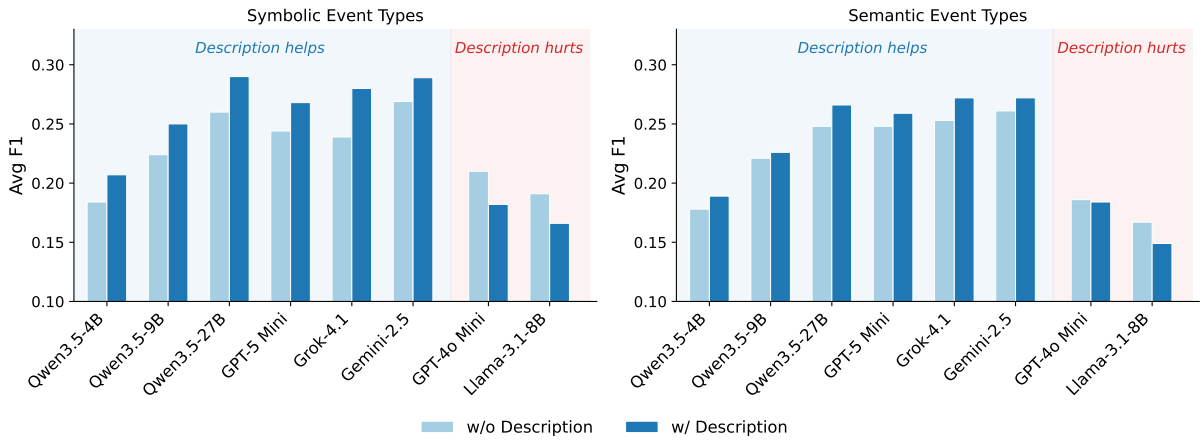


Figure 6: Effect of adding event descriptions on Avg F1 under Symbolic and Semantic label settings. Models with strong reasoning capability benefit from descriptions, while less capable models see degraded performance.

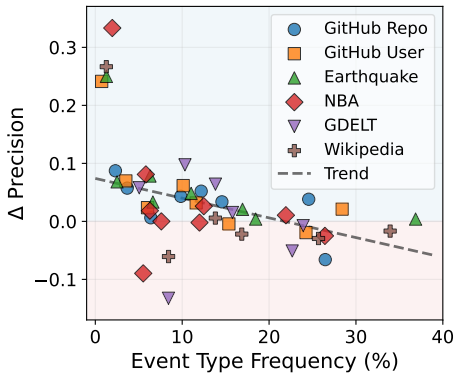


Figure 7: Precision gain from using Symbolic over Semantic labels plotted against event type frequency.

types and timestamps alone provide, even when semantic bias is still present. Removing label semantics by using symbolic event types helps suppress this bias, but without descriptions it also removes the semantic cues that the LLM could use. The model is then left mainly with sequence patterns in labels and timestamps, which explains why symbolic labels alone perform similarly to semantic labels for most models.

We also observe that descriptions are most effective when paired with symbolic labels, achieving the best per-dataset F1 and MAE on most datasets in Table 2 and the highest average F1 and lowest average MAE in Figure 1. These results support the semantic bias intuition from Section 3.3: when label names no longer provide semantic cues, the model is encouraged to use the event descriptions, allowing the information they carry, and any pretrained knowledge they activate, to contribute more directly to prediction.

5.3 Description Benefit Tracks Reasoning Capability

Predicting from a description is itself a non-trivial reasoning task, since the model must extract causal signal from free-form text and integrate it with the temporal pattern of past events. We thus examine the relationship between a model's reasoning capability and its ability to use descriptions. We use two reasoning benchmarks as proxies[2] : GPQA Diamond (Rein et al., 2024) tests graduate-level domain reasoning, and Humanity's Last Exam (HLE) (Phan et al., 2026) tests multi-disciplinary multi-step reasoning.

Figure 6 compares average F1 with and without descriptions and reveals a split. And as shown in Table 8 in Appendix B.1, Models that benefit from descriptions score at least 77% on GPQA Diamond and at least 7% on HLE, while GPT-4o Mini and Llama-3.1-8B perform significantly worse on both and see overall F1 drop when descriptions are added. We hypothesize that without strong reasoning capabilities, these weaker models are distracted by the description and lose the temporal patterns they would otherwise follow. Appendix B.3 and Table 10 provide further details about the performance of the weaker models.

5.4 Symbolic Labels Improve Rare Event Precision

We next investigate why symbolic labels outperform semantic labels when both are paired with descriptions. Our intuition follows the semantic bias defined in Section 3.3: an LLM may predict an

[2]Benchmark scores are sourced from Artificial Analysis, https://artificialanalysis.ai/.

Model	GitHub Repo		GitHub User		Amazon		Earthquake		GDELT		NBA		Wikipedia	
	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓
non-LLM Baselines														
ARIMA	0.166	3.075	0.193	2.152	**0.130**	7.010	0.149	0.374	0.151	10.363	0.059	0.105	0.175	24.561
KNN	**0.332**	2.906	**0.327**	2.244	0.117	6.875	**0.175**	0.397	**0.153**	11.139	0.260	0.089	**0.256**	21.239
Hawkes	0.067	4.215	0.109	3.805	0.056	6.427	0.067	0.561	0.055	18.596	0.046	0.154	0.084	102.286
NHP	0.159	2.462	0.131	1.665	0.044	5.934	0.148	0.343	0.095	10.772	0.148	0.105	0.084	18.227
RMTPP	0.258	2.688	0.298	1.944	0.026	6.326	0.161	0.360	0.058	**9.688**	0.338	**0.073**	0.138	23.022
MLP	0.307	2.637	0.287	1.903	0.048	6.501	0.163	0.359	0.066	10.306	0.329	0.080	0.090	24.188
non-LLM Baselines ***w/ Description***														
NHP	0.131	**2.288**	0.117	**1.563**	0.034	**5.780**	0.151	**0.335**	0.138	11.423	0.078	0.115	0.177	**16.720**
RMTPP	0.248	2.944	0.227	2.075	0.028	6.339	0.161	0.351	0.109	10.265	**0.362**	0.079	0.199	20.202
MLP	0.196	2.733	0.180	2.149	0.043	5.982	0.145	0.359	0.115	10.406	0.347	0.075	0.097	23.427

Table 3: Per-dataset performance of non-LLM baselines. Best result per dataset is in **bold**.

event type from its label name alone even when the description does not support that prediction (McCoy et al., 2024). This can create bias-driven false positives. Symbolic labels remove the semantic signal from the label name, encouraging the model to base its prediction on the description instead. Since precision is $\mathrm{TP}_t/(\mathrm{TP}_t + \mathrm{FP}_t)$, the same number of false positives has a smaller effect for common events, where there are many true positives, but a much larger effect for rare events, where the true-positive base is small. Therefore, if symbolic labels reduce semantic-bias-driven false positives, the precision gain should be strongest for rare events.

To test this hypothesis, we compute performance differences between symbolic and semantic labels, with descriptions included in both settings. For each event type, we measure the change in precision, recall, and F1, and compare it with that type's frequency within its dataset. Figure 7 shows the precision results, while Figure 8 in Appendix B.4 shows recall and F1. We exclude Amazon because its large label space would dominate the results. Across the remaining 46 event types, the precision gain from symbolic labels increases sharply as frequency decreases, with a Pearson correlation of $r = -0.51$ ($p \approx 0.0003$). In contrast, recall shows almost no relationship with frequency ($r = -0.01$), suggesting that symbolic labels mainly reduce false positives rather than changing how often true events are recovered. This does not mean that rare events carry stronger semantic bias. Prior work links label bias to properties of the label tokens and their pretraining distribution, rather than to a label's frequency in a downstream dataset (Zhao et al., 2021; Reif and Schwartz, 2024). Thus, the frequency effect is better explained by precision: rare events have fewer true positives, so the same number of false positives has a larger impact.

5.5 Comparison with non-LLM Baselines

We compare our approach with six baselines spanning statistical, nonparametric, temporal point process (TPP), and neural sequence-modeling approaches. The TPP baselines include one classical model, Hawkes Process (Hawkes, 1971), and two neural TPPs: NHP (Mei and Eisner, 2017) and RMTPP (Du et al., 2016). We also include ARIMA (Box et al., 2015) and KNN as non-TPP baselines, and an MLP as a simple neural sliding-window baseline. We use no descriptions for ARIMA, KNN, and Hawkes. For the neural baselines, NHP, RMTPP, and MLP, we additionally evaluate description-aware variants that encode each event's description with Sentence-BERT (Reimers and Gurevych, 2019) and concatenate the embedding with the type and time features at the model input. We exclude fine-tuned LLM event predictors (Liu and Quan, 2024; Kong et al., 2026; Gupta et al., 2025) because they require a large training corpus, which lies beyond the scope of our evaluation.

ARIMA forecasts inter-event times with small-order models chosen by grid search and predicts the most frequent type as the label. KNN matches the most similar sliding windows in the sequence's history by type and temporal similarity. Hawkes Process models conditional intensity as background rates with exponentially decaying excitation from past events, and predicts the highest-intensity type along with the reciprocal of total intensity as the time. NHP is a neural TPP based on a continuous-time LSTM whose hidden state decays between events, while RMTPP is a neural TPP based on a GRU with separate output heads for type and time. Finally, MLP is a two-layer feedforward network over a fixed sliding window of recent events. We train the parametric baselines on the first 80% to 90% events in each sequence (same length that our

LLM method receives in its prompt), and evaluate autoregressively on the remaining events.

Table 3 provides the non-LLM comparison point for our results. Overall, the strongest LLM setting in Table 2, which combines descriptions with symbolic labels, outperforms the best non-LLM baselines on most datasets for both metrics. Among the non-LLM methods, KNN is the strongest baseline for event-type prediction, while description-aware NHP is strongest for time prediction. The effect of descriptions on non-LLM baselines is mixed: adding Sentence-BERT embeddings often improves MAE, but does not consistently improve F1. This suggests that description embeddings help these baselines capture timing information more reliably than event-type semantics.

6 Conclusion

In this paper, we showed that LLM world knowledge can improve event prediction when the prediction interface exposes it. We identified two key interface choices: grounding events in natural-language descriptions and using symbolic labels as prediction targets. Their combination yields the strongest results, with capable LLMs outperforming description-free variants and non-LLM baselines on most datasets, while weaker models benefit little or even degrade. To support this evaluation, we introduce DESCRIBEEVENT, a benchmark of 150K events across seven datasets and six domains, and show that the benefit of descriptions is closely tied to model reasoning ability.

7 Limitations

Our work has several limitations. First, DESCRIBEEVENT is designed as an evaluation benchmark; we do not explore using it as training data to fine-tune LLMs for description-grounded prediction, e.g., via parameter-efficient adaptation (Hu et al., 2022; Wu et al., 2026b,a), which is a promising direction for future work. Second, while we identify a correlation between reasoning capability, as measured by GPQA Diamond, and description utilization, establishing a causal mechanism requires further investigation. Third, LLM inference may take seconds per prediction versus tens of milliseconds for specialized event-prediction non-LLM models in our experiments; we view this added latency as a trade-off for better performance and the interpretability offered by the model's natural-language reasoning. Finally, we represent inter-arrival times as raw floating-point numbers in text using standard tokenizers while improving their representations is worth exploring further (Singh and Strouse, 2024).

References

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. 2024. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. 2015. *Time Series Analysis: Forecasting and Control*, 5th edition. John Wiley & Sons.

Yanan Cao, Farnaz Fallahi, Murali Mohana Krishna Dandu, Lalitesh Morishetti, Kai Zhao, Luyi Ma, Sinduja Subramaniam, Jianpeng Xu, Evren Korpeoglu, Kaushiki Nag, Sushant Kumar, and Kannan Achan. 2026. Is more context always better? Examining LLM reasoning capability for time interval prediction. In *Proceedings of the ACM Web Conference 2026*.

He Chang, Chenchen Ye, Zhulin Tao, Jie Wu, Zhengmao Yang, Yunshan Ma, Xianglin Huang, and Tat-Seng Chua. 2024. A comprehensive evaluation of large language models on temporal event forecasting. *arXiv preprint arXiv:2407.11638*.

Lili Chen, Wensheng Gan, Shuang Liang, and Philip S. Yu. 2025a. Enhancing temporal awareness in LLMs for temporal point processes. *arXiv preprint arXiv:2601.00845*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025b. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564. Association for Computing Machinery.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, pages 19622–19635.

Shubham Gupta, Thibaut Durand, Graham W. Taylor, and Lilian Bialokozowicz. 2025. LAST SToP for modeling asynchronous time series. In *International Conference on Machine Learning*.

Alan G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, pages 83–90.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*.

Quyu Kong, Yixuan Zhang, Yang Liu, Panrong Tong, Enqi Liu, and Feng Zhou. 2026. Byte-token enhanced language models for temporal point processes analysis. In *Proceedings of the ACM Web Conference 2026*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Zefang Liu and Yinzhu Quan. 2024. TPP-LLM: Modeling temporal point processes by efficiently fine-tuning large language models. *arXiv preprint arXiv:2410.02062*.

Zefang Liu and Yinzhu Quan. 2025. Retrieval of temporal event sequences from textual descriptions. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 37–49.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. 2024. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*.

Hongyuan Mei and Jason Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, pages 6754–6764.

OpenAI. 2025. GPT-5 system card.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, and 1 others. 2026. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, pages 1139–1146.

Yuval Reif and Roy Schwartz. 2024. Beyond performance: Quantifying and mitigating label bias in llms. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 6784–6798.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

James Requeima, John Bronskill, Dami Choi, Richard E. Turner, and David Duvenaud. 2024. LLM processes: Numerical predictive distributions conditioned on natural language. In *Advances in Neural Information Processing Systems*, pages 109609–109671.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y. Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. 2023. Language models can improve event prediction by few-shot abductive reasoning. In *Advances in Neural Information Processing Systems*.

Aaditya K. Singh and DJ Strouse. 2024. Tokenization counts: the impact of tokenization on arithmetic in frontier llms. *arXiv preprint arXiv:2402.14903*.

Mingtian Tan, Mike A. Merrill, Zack Gottesman, Tim Althoff, David Evans, and Tom Hartvigsen. 2025. Inferring events from time series using language models. *arXiv preprint arXiv:2503.14190*.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. 2024. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc Le. 2023. Symbol tuning improves in-context learning in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 968–979.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. 2025. Context is key: A benchmark for forecasting with essential textual information. In *International Conference on Machine Learning*, pages 66887–66944.

Zijun Wu, Yongchang Hao, and Lili Mou. 2026a. Tokmem: One-token procedural memory for large language models. In *International Conference on Learning Representations*, pages 32460–32480.

Zijun Wu, Yongchang Hao, and Lili Mou. 2026b. Ultra-low-dimensional prompt tuning via random projection. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1284–1303.

xAI. 2025. Grok 4.1 model card.

Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao Jiang, Chen Pan, James Y. Zhang, Qingsong Wen, Jun Zhou, and Hongyuan Mei. 2024. EasyTPP: Towards open benchmarking temporal point processes. In *International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Chenchen Ye, Ziniu Hu, Yihe Deng, Zijie Huang, Mingyu Derek Ma, Yanqiao Zhu, and Wei Wang. 2024. MIRAI: Evaluating LLM agents for event forecasting. *arXiv preprint arXiv:2407.01231*.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. 2020. Self-attentive hawkes process. In *International Conference on Machine Learning*, pages 11183–11193.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, pages 43322–43355.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer hawkes process. In *International Conference on Machine Learning*, pages 11692–11702.

Dataset	Source	Prediction Task
GitHub Repo	GH Archive	Repository event
GitHub User	GH Archive	User action
Amazon	Amazon Reviews'23	Product category
Earthquake	USGS API	Tectonic zone
GDELT	GDELT 2.0 Database	Diplomatic action
NBA	NBA Stats API	Play type
Wikipedia	MediaWiki API	Edit type

Table 4: Dataset sources and prediction tasks in DESCRIBEEVENT.

A Datasets

A.1 Dataset Sources and Statistics

Table 4 lists the seven datasets in DESCRIBEEVENT over six domains, together with their sources of data and prediction task.

For the details of their sources, we access each through its public interface and use it within their terms. Specifically, GitHub Repo and GitHub User come from the GH Archive under CC BY-4.0 license; Amazon Reviews from the McAuley Lab's Amazon Reviews 2023 release (Hou et al., 2024) for academic research; Earthquake records from the USGS Earthquake Catalog API, a U.S. public-domain federal source; GDELT events from the GDELT 2.0 Event Database, free for academic and non-commercial use with attribution; NBA play-by-play from the public NBA Stats endpoints available for non-commercial use; and Wikipedia edits from the MediaWiki Revisions API, where article content is licensed under CC BY-SA 4.0.

Table 5 reports statistics over all datasets of DESCRIBEEVENT after our curation, which contains 1,940 sequences and 148,185 events in total, with six to ten event types per dataset and average sequence lengths from 48.9 to 101.8 events. Time units range from minutes for NBA plays to weeks for Amazon purchases, and average description length ranges from 8.6 words for GDELT to 62.3 words for GitHub User. For Amazon, the full label space contains 40 product categories. However, each sequence only draws from a single parent category. Therefore in our evaluation, a model is exposed to at most 10 event types per sequence.

A.2 Semantic Label Space

Table 6 lists the semantic event types for each dataset, together with the value of $|\mathcal{Y}|$, which denotes the per-sequence label space, namely the number of distinct labels a model can encounter within a single sequence.

For most datasets, $|\mathcal{Y}|$ matches the dataset-wide label space since every sequence draws from the same set of types. But the Amazon dataset is an exception, because each Amazon sequence is anchored to a single parent category, and a model therefore sees only the 10 sub-categories within that category rather than the full 40. We set $|\mathcal{Y}| = 10$ accordingly.

A.3 Sequence Filters

This appendix details the curation filters introduced in Section 4.2 with the per-dataset threshold values used to produce DESCRIBEEVENT.

There are three sequence-level filters apply to every dataset: **(i)** The dominant-type cap, denoted Dominant%, bounds the maximum fraction of events drawn from a single `type_event`, **(ii)** The consecutive-repeat cap, denoted Repeat%, bounds the maximum fraction of adjacent events sharing the same type; and **(iii)** The minimum type diversity requirement, or Min types, which sets a floor on the number of distinct types per sequence.

Each sequence must also fall within a per-dataset length range and pass a small set of domain-specific quality filters that remove sources of label noise, such as bot accounts on GitHub and Wikipedia, unverified Amazon reviews, and low-magnitude earthquakes. Any sequence that fails one or more of these checks is discarded before the splits are formed. Table 7 reports those threshold values for each dataset.

B Additional Analyses and Results

B.1 Reasoning Benchmark Scores

We use GPQA Diamond and HLE scores as references for the models' reasoning capabilities in Section 5.3. Table 8 lists those scores for each model, along with whether adding descriptions improves event prediction.

We see a clear split on both benchmarks. Every model that benefits from descriptions scores at least 77% on GPQA Diamond and at least 7% on HLE. GPT-4o Mini and Llama-3.1-8B, for which descriptions hurt, sit far below at 43% and 26% on GPQA Diamond and at 4% and 5% on HLE.

Interestingly, since Qwen3.5-4B is the smallest model in our evaluation but it sits in the helper group with 77.1% on GPQA Diamond, whereas Llama-3.1-8B has twice as many parameters and sits in the non-helper group at 25.9%. Together with Figure 6 in Section 5.3, these findings support the claim that description utilization tracks general reasoning capability rather than model size.

Dataset Statistics	GitHub User	GitHub Repo	Wikipedia	NBA	Earthquake	GDELT	Amazon	Total
Number of Sequences	382	373	276	286	215	179	229	1,940
Number of Events	33,856	25,576	22,063	29,102	10,510	13,779	13,299	148,185
Number of Event Types	8	8	6	9	8	7	10	-
Time Unit	Hours	Hours	Hours	Minutes	Days	Days	Weeks	-
Avg Sequence Length	88.6	68.6	79.9	101.8	48.9	77.0	58.1	-
Avg Description Words	62.3	59.9	20.8	23.8	42.9	8.6	39.4	-

Table 5: Dataset statistics for DESCRIBEEVENT.

Dataset	$\lvert\mathcal{Y}\rvert$	Semantic Label Space
Amazon	10	**Beauty & Personal Care** covers foot/hand/nail care, fragrance, hair care, makeup, men's grooming, personal care, salon & spa, shave & hair removal, skin care, and tools & accessories. **Books** covers arts & photography, biographies & memoirs, children's books, Christian books & bibles, crafts/hobbies/home, history, literature & fiction, mystery/thriller/suspense, politics & social sciences, and teen & young adult. **Electronics** covers accessories & supplies, camera & photo, car & vehicle electronics, computers & accessories, GPS finders & accessories, headphones/earbuds/accessories, home audio, portable audio & video, security & surveillance, and television & video. **Home & Kitchen** covers bath, bedding, event & party supplies, furniture, heating/cooling/air quality, home décor, kitchen & dining, seasonal décor, storage & organization, and wall art.
Earthquake	8	North America; Central and South America; Northwest Pacific; Southeast Asia and Oceania; Southwest Pacific; Mediterranean and Middle East; Central and South Asia; Atlantic and Africa.
GDELT	7	public statement; consultation; diplomatic engagement; material cooperation; disapproval; coercion; armed conflict.
NBA	9	made 2-point; made 3-point; missed field goal; free throw; turnover; foul; rebound; steal or block; timeout.
Wikipedia	6	revert; reference; content addition; content removal; copy edit; minor edit.
GitHub	8	issue opened; issue closed; PR opened; PR merged; push; release; PR reviewed; comment.

Table 6: Label space `type_event` for each dataset. $\lvert\mathcal{Y}\rvert$ denotes the number of semantic event labels.

Dataset	Dominant %	Repeat %	Min types	Length	Domain-specific filters
Amazon	30%	35%	3	50–80	Verified-purchase reviews only; user time span 3–60 months; *other*-typed events stripped.
Earthquake	50%	–	4	30–50	Magnitude at least 5.0.
GDELT	35%	35%	3	40–100	Root events only; at least 2 mentions, with at least 3 required for armed conflict; business, media, and educational actors excluded from armed conflict; same-country pairs excluded; span 30–1200 days; Goldstein swing of at least 12 on at most 15% of adjacent events.
NBA	40%	30%	4	90–110	All four regulation quarters present; overtime excluded.
Wikipedia	40%	40%	4	60–80	Bot accounts excluded; chunk duration at most 4380 hours which is about six months.
GitHub Repo	50%	–	5	60–80	Repositories with more than 20% bot events dropped; at least 150 characters per event on average; 2–5 distinct actors per repository.
GitHub User	50%	–	5	80–100	Bot accounts excluded; 2–5 distinct repositories per user; at least 150 characters per event on average; adjacent events with the same type and text collapsed.

Table 7: Filtering thresholds for each dataset. Dominant % denotes the maximum share of events with the same `type_event` within a sequence, and Repeat % denotes the maximum share of adjacent event pairs with the same type. Min types specifies the minimum number of distinct event types per sequence, and Length gives the allowed sequence-length range.

Model	GPQA Diamond	HLE	Desc. helps?
Qwen3.5-27B	85.8%	22.2%	✓
Qwen3.5-9B	80.6%	13.3%	✓
Qwen3.5-4B	77.1%	7.8%	✓
GPT-5 Mini	80.3%	14.6%	✓
Grok-4.1	85.3%	17.6%	✓
Gemini-2.5	79.0%	11.1%	✓
GPT-4o Mini	42.6%	4.0%	✗
Llama-3.1-8B	25.9%	5.1%	✗

Table 8: Reasoning benchmark scores on GPQA and HLE, and whether adding descriptions improves event prediction.

B.2 Average Performance Across Datasets

Table 9 reports the dataset-averaged macro F1 and MAE for every row of Tables 2 and 3, summarizing the numbers that underlie Figure 1 across the datasets. Notably, Qwen3.5-27B with symbolic labels and descriptions attains the highest Avg F1 of 0.290 and the lowest Avg MAE of 5.248 overall, and the four LLM settings (2×2 design that crosses description inclusion and label setups) rank in the same order as the finding in our main experiment, with symbolic labels plus descriptions on top and semantic labels alone at the bottom. Adding descriptions raises the best per-group F1 from 0.261 to 0.272 under semantic labels and from 0.269 to 0.290 under symbolic labels, with the larger gain on the symbolic side, consistent with the semantic bias account. The non-LLM baselines sit well below the LLM rows on F1, with KNN strongest at 0.231, and pairing NHP, RMTPP, and MLP with descriptions yields a small MAE improvement without closing the F1 gap.

B.3 Detailed Results on Smaller Models

This appendix expands the results in Section 5.3 with two smaller models, GPT-4o Mini and Llama-3.1-8B, run under the same four-setting protocol as the capable LLMs in Table 2. Table 10 reports macro F1 and MAE for all four settings on the seven datasets in DESCRIBEEVENT.

Adding descriptions does not improve overall macro F1 for either model. Specifically, on GPT-4o Mini, descriptions reduce overall F1 by 0.028 under symbolic labels and by 0.002 under semantic labels. On Llama-3.1-8B, the drop is 0.025 under symbolic labels and 0.018 under semantic labels. The drop is therefore larger under symbolic than under semantic labels for both models, inverting the pattern seen in the capable LLMs in our main Table 2, where symbolic labels paired with descriptions is the strongest setting.

B.4 Detailed Analysis about Rare Events

Figure 7 in Section 5.4 only provide plots for the precision panel of the per-event gain analysis. In this appendix, we shows all three panels in Figure 8, with precision, recall, and F1 from left to right. Each point is one event type from the six datasets, plotted at its within-domain frequency against the symbolic-over-semantic performance gain. Precision and F1 both rise as frequency decreases, with correlations of $r = -0.51$ and $r = -0.40$, while recall stays flat at $r = -0.01$. This matches our semantic bias intuition. Symbolic labels remove the bias-driven false positives that dilute precision on rare events, while the description that drives recall is shared by both label settings.

C LLM Inference Configuration

We serve the open-weight Qwen3.5 models locally with vLLM (Kwon et al., 2023), and access the closed-weight models GPT-5 Mini, Grok-4.1 Fast, Gemini-2.5 Flash, and GPT-4o Mini through the OpenRouter API. For each model we keep the provider's default sampling temperature and do not tune any decoding hyperparameters, so the comparison reflects out-of-the-box behavior. For models that support a deep thinking mode, we disable it because extended traces tend to inflate intermediate computation on simple inputs without improving accuracy (Chen et al., 2025b), and we observed the same degradation in our preliminary experiments. We instead instruct the model to write a concise chain of thought of two to three sentences before emitting the final `{event_type, time}` answer, as shown in the system message in Figure 9.

For a sequence of length N, we set the initial history length to $n_0 = \lfloor \rho N \rfloor$, where the fraction $\rho \in (0, 1)$ is chosen per dataset. Specifically, for the datasets with shortest averaged sequence length shown in in Table 5, i.e., Earthquake and Amazon, we set $\rho = 0.8$, and for the rest, we set $\rho = 0.9$. This yields roughly eight prediction steps per sequence on average across all datasets.

D Prompt Templates

We illustrate the four evaluation settings using a GitHub User event sequence as a running example, shown in Figure 9. The settings differ along two axes, whether event types use *semantic* labels such as `push` and `comment` or *symbolic* labels such as

Model	F1↑	MAE↓
Semantic Event Type		
Qwen3.5-4B	0.178	5.866
Qwen3.5-9B	0.221	5.684
Qwen3.5-27B	0.248	5.455
GPT-5 Mini	0.248	5.366
Grok-4.1	0.253	5.692
Gemini-2.5	0.261	5.671
*Semantic **w/ Description***		
Qwen3.5-4B	0.189	5.836
Qwen3.5-9B	0.226	5.786
Qwen3.5-27B	0.266	5.389
GPT-5 Mini	0.259	5.378
Grok-4.1	**0.272**	5.383
Gemini-2.5	**0.272**	**5.350**

Model	F1↑	MAE↓
Symbolic Event Type		
Qwen3.5-4B	0.184	6.463
Qwen3.5-9B	0.224	5.811
Qwen3.5-27B	0.260	5.675
GPT-5 Mini	0.244	5.401
Grok-4.1	0.239	5.821
Gemini-2.5	0.269	5.775
*Symbolic **w/ Description***		
Qwen3.5-4B	0.207	5.673
Qwen3.5-9B	0.250	5.704
Qwen3.5-27B	**0.290**	**5.248**
GPT-5 Mini	0.268	5.285
Grok-4.1	0.280	5.366
Gemini-2.5	0.289	5.429

Model	F1↑	MAE↓
non-LLM Baselines		
KNN	**0.231**	6.413
ARIMA	0.146	6.806
Hawkes	0.069	19.435
NHP	0.116	5.644
RMTPP	0.182	6.300
MLP	0.184	6.568
*non-LLM Baselines **w/ Description***		
NHP	0.118	**5.461**
RMTPP	0.191	6.036
MLP	0.160	6.447

Table 9: Dataset-averaged macro F1 and MAE for every row of Tables 2 and 3, organized into Semantic, Symbolic, and non-LLM groups with their description-aware variants. The best Avg F1 and Avg MAE within each group are in **bold**.

	GitHub Repo		GitHub User		Amazon		Earthquake		GDELT		NBA		Wikipedia	
Setting	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓	F1↑	MAE↓
GPT-4o Mini														
Symbolic	0.216	2.152	0.243	1.421	0.163	5.711	0.219	0.348	0.175	10.222	0.177	0.115	0.280	16.603
+ Description	0.199	2.276	0.183	1.518	0.187	5.780	0.202	0.391	0.162	10.139	0.088	0.107	0.255	16.701
Semantic	0.206	2.177	0.197	1.452	0.121	5.781	0.206	0.355	0.135	10.065	0.202	0.089	0.232	16.621
+ Description	0.218	2.325	0.171	1.563	0.131	5.849	0.199	0.371	0.155	10.179	0.163	0.104	0.249	16.639
Llama-3.1-8B														
Symbolic	0.224	3.958	0.216	3.084	0.141	7.251	0.190	1.294	0.175	12.454	0.154	11.877	0.237	24.908
+ Description	0.161	3.809	0.173	2.978	0.152	6.570	0.178	0.857	0.151	12.978	0.118	0.242	0.231	19.344
Semantic	0.187	3.983	0.182	2.712	0.139	7.717	0.165	4.639	0.153	12.413	0.117	0.171	0.225	31.863
+ Description	0.139	4.354	0.125	2.616	0.141	6.699	0.182	0.880	0.155	15.325	0.110	0.118	0.195	19.094

Table 10: Performance of smaller models across datasets. Adding descriptions does not consistently help, suggesting that sufficient model capability is needed to leverage descriptive information.

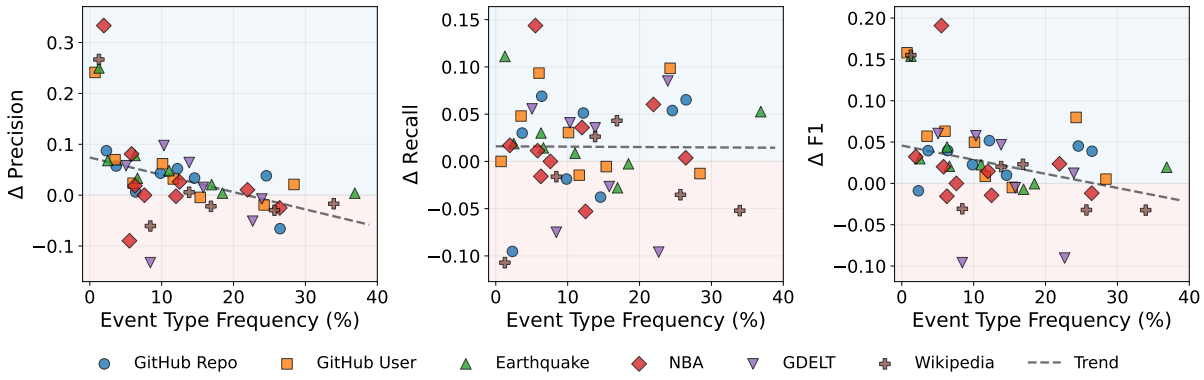


Figure 8: Per-type gain from Symbolic over Semantic labels, both with descriptions, versus event type frequency, for precision, recall, and F1.

`type_0` and `type_1`, and whether natural-language descriptions are included.

The system message tells the model that it must provide a brief reasoning with two to three sentences and predict the next event type and its time. It also lists the valid event types for the sequence, and fixes the response format so that we can parse a single event type and a positive number of hours.

The user message then serializes the history as a chronological list of past events, with each event written as `[type] +Xh`, where `+Xh` is the time gap from the previous event. The description variants append a short natural-language sentence after each event, drawn from DESCRIBEEVENT.

System

You predict the next event in a sequence. Given the event history, predict the next event type and time.

Valid event types: push, comment, pr_merged, pr_reviewed, pr_opened, issue_closed, issue_opened, release.

Each event shows: [type] +Xh (hours since previous event). Use both the event types and the +Xh time gaps to predict what happens next and when.

Response format (follow exactly):
Brief reasoning here (2–3 sentences max).
{"event_type": "<type>", "time_hours": <positive_number>}

User

Event history (chronological):
1. [pr_reviewed] +0.000h
2. [push] +0.881h
3. [comment] +5.280h
4. [comment] +0.386h
5. [pr_reviewed] +0.503h

Predict the next event_type and time_hours.

(a) **Semantic** setting. Human-readable names, no descriptions.

System

You predict the next event in a sequence. Given the event history, predict the next event type and time.

Valid event types: type_0, type_1, type_2, type_3, type_4, type_5, type_6, type_7.

Each event shows: [type] +Xh (hours since previous event). Use both the event types and the +Xh time gaps to predict what happens next and when.

Response format (follow exactly):
Brief reasoning here (2–3 sentences max).
{"event_type": "<type>", "time_hours": <positive_number>}

User

Event history (chronological):
1. [type_3] +0.000h
2. [type_0] +0.881h
3. [type_1] +5.280h
4. [type_1] +0.386h
5. [type_3] +0.503h

Predict the next event_type and time_hours.

(b) **Symbolic** setting. Anonymous labels, no descriptions.

System

You predict the next event in a sequence. Given the event history, predict the next event type and time.

Valid event types: push, comment, pr_merged, pr_reviewed, pr_opened, issue_closed, issue_opened, release.

Each event shows: [type] +Xh | description. Use the event types, their descriptions, and the +Xh time gaps to predict what happens next and when.

Response format (follow exactly):
Brief reasoning here (2–3 sentences max).
{"event_type": "<type>", "time_hours": <positive_number>}

User

Event history (chronological):
1. [pr_reviewed] +0.000h | PR approved in org/main-repo: docs(changelog): fix change log entry of #13021
2. [push] +0.881h | Code pushed to org/sub-module: fixup! feat(header_bulk_carrier): add new FFI interfaces for getting request/response headers in bulk
3. [comment] +5.280h | Comment on PR in org/main-repo: Rebased on master
4. [comment] +0.386h | Review comment on PR in org/main-repo: Improved error logging when having inconsistent labels count.
5. [pr_reviewed] +0.503h | PR approved in org/main-repo: fix(http-log): add port information to the host header

Predict the next event_type and time_hours.

(c) **Semantic + Description** setting. Human-readable names with descriptions.

System

You predict the next event in a sequence. Given the event history, predict the next event type and time.

Valid event types: type_0, type_1, type_2, type_3, type_4, type_5, type_6, type_7.

Each event shows: [type] +Xh | description. Use the event types, their descriptions, and the +Xh time gaps to predict what happens next and when.

Response format (follow exactly):
Brief reasoning here (2–3 sentences max).
{"event_type": "<type>", "time_hours": <positive_number>}

User

Event history (chronological):
1. [type_3] +0.000h | PR approved in org/main-repo: docs(changelog): fix change log entry of #13021
2. [type_0] +0.881h | Code pushed to org/sub-module: fixup! feat(header_bulk_carrier): add new FFI interfaces for getting request/response headers in bulk
3. [type_1] +5.280h | Comment on PR in org/main-repo: Rebased on master
4. [type_1] +0.386h | Review comment on PR in org/main-repo: Improved error logging when having inconsistent labels count.
5. [type_3] +0.503h | PR approved in org/main-repo: fix(http-log): add port information to the host header

Predict the next event_type and time_hours.

(d) **Symbolic + Description** setting. Anonymous labels with descriptions.

Figure 9: Prompt templates for the four evaluation settings, illustrated on a GitHub User event sequence.