

Language-Guided Time-Series Event Detection with Zero-Shot Cross-Modal Grounding

Tianjun Hou¹, Xiqing Chang¹, Sky Chenwei Wan^{1,2},

¹AI Lab, SLB, France

²LTCI, Télécom Paris, Institut Polytechnique de Paris, France

thou2@slb.com, xchang3@slb.com, cwan5@slb.com

Abstract

In this study, we address the language-guided time-series event detection problem: given a natural-language event description and a time series, the system returns event occurrences as temporal spans. Language-guided event detection leverages domain knowledge expressed in natural language to enable zero-shot detection without domain-specific fine-tuning. We introduce GroundingMOMENT¹, a zero-shot cross-modal grounding framework that aligns a time-series foundation model with a language model. GroundingMOMENT concurrently localizes both complex, high-level event occurrences and their constituent domain-agnostic structural primitives (e.g., spikes, plateaus). We further introduce GroundTSSyn², a scalable pipeline for generating synthetic time-series grounding data, and release GroundTSBench³, the first open-source benchmark for language-guided time-series grounding. Trained exclusively on GroundTSBench, GroundingMOMENT outperforms time-series multimodal large language models and achieves fast and accurate event detection. Case studies further demonstrate its zero-shot generalizability to real-world data from diverse domains.

1 Introduction

Time-series event detection is a fundamental task across many safety-critical and industrial domains. Many operational decisions depend on recognizing not only whether an event occurred in a time series, but also its precise temporal boundaries. Despite this importance, conventional time-series classification (Fawaz et al., 2019) or anomaly detection (Blázquez-García et al., 2022) is often insufficient: it merely labels a fixed window without localizing event boundaries.

¹<https://github.com/houtj/GroundingMOMENT>

²<https://github.com/houtj/GroundTSSyn>

³<https://huggingface.co/datasets/houtj19900/GroundTSBench>

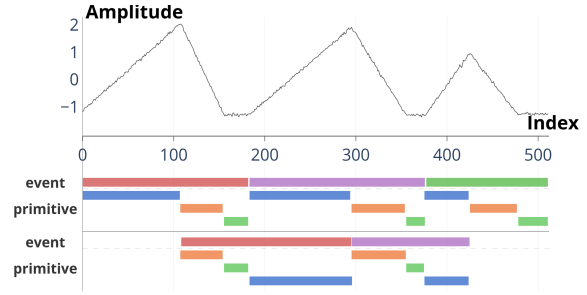


Figure 1: Illustration of the result of GroundingMOMENT on a synthetic signal. Colored bands indicate detected primitive and event spans. **Top:** Canonical query (*A steady ramp up, followed by a sharp downward slide, before a level stretch*): events and primitives are correctly localized. **Bottom:** Permuted query (*A sharp downward slide, followed by a level stretch, then a steady ramp up*): detected spans shift to align with the new primitive order, demonstrating that the model captures sequential structure.

Recent advances in time-series foundation models (Goswami et al., 2024; Garza et al., 2024; Ansari et al., 2024) show promise for broad representation learning. However, time-series data is highly domain-dependent. An identical morphological pattern (e.g., an abrupt spike) may denote a sensor artifact in one domain while signifying a catastrophic failure in another. Purely data-driven approaches thus require domain-specific fine-tuning. However, data in domains such as healthcare and aerospace is sensitive and scarce. Annotation demands deep domain expertise, making high-quality labels prohibitively expensive to obtain.

Domain expertise, however, is accessible in text, such as textbooks, operational manuals, and scientific literature. Complex time-series events can generally be decomposed into ordered sequences of domain-agnostic, low-level structural patterns, namely *primitives*, such as a monotonic increase or decrease, a transient spike, etc. (Table 1). These primitives carry no domain-specific semantics. The

same primitive types appear across drilling telemetry, electrocardiogram (ECG), and accelerometry alike. A model trained on synthetic primitives can therefore generalize to real-world events described in primitive-based natural language, enabling zero-shot learning in time-series analysis.

To this end, we make three contributions. (i) We propose GroundingMOMENT, a *zero-shot* cross-modal grounding model that aligns a time-series foundation model with a large language model to perform language-guided event detection. It executes a hierarchical two-level detection process (Figure 1), simultaneously localizing fundamental primitives and composite event occurrences. Experiments show it outperforms time-series multimodal large language models (TS-MLLMs) and achieves detection accuracy comparable to agentic LLM systems at a fraction of the latency. (ii) We introduce GroundTSSyn, a scalable pipeline for programmatic synthesis of annotated time-series grounding data from a vocabulary of domain-agnostic structural primitives, coupled with a description-generation agentic system that produces diverse, lexically rich natural-language groundings, to enable grounding-model training under the constraints of real-domain data and annotation scarcity. (iii) We release GroundTSBench, the first open-source benchmark for language-guided time-series grounding, containing 16,410 samples with diverse synthetic temporal patterns and linguistic variations. Trained exclusively on GroundTSBench, GroundingMOMENT transfers directly to real-world domains without any domain-specific fine-tuning.

2 Related Work

Time-Series Foundation Models. Pre-trained on large collections of public time-series data, foundation models such as MOMENT (Goswami et al., 2024), PatchTST (Nie et al., 2023), Chronos (Ansari et al., 2024), and Moirai (Woo et al., 2024) demonstrate broad representational capacity across forecasting, classification, and anomaly detection. Despite these advances, these models remain *domain-dependent*: the semantic meaning of a waveform pattern must be supplied by a downstream fine-tuning step on labeled domain data. Our approach overcomes this limitation by conditioning inference on natural-language event descriptions, eliminating the need for supervised domain fine-tuning.

Multimodal Large Language Models for Time Series (TS-MLLMs). Recent work has explored aligning time-series data with LLMs via patch re-programming (Jin et al., 2024), instruction tuning on synthetic captions (Xie et al., 2025), and reinforcement learning-based reasoning (Guan et al., 2026). Although these systems demonstrate strong question-answering capabilities, they are not designed for span detection: temporal boundaries are encoded as discrete text tokens, yielding poor localization precision. Agentic frameworks such as SELA (Wan et al., 2026) achieve higher detection accuracy through iterative Reasoning-and-Acting (ReAct) loops over time-series data, but their high per-query latency and token consumption make them impractical for telemetry monitoring.

Time-Series Language Alignment. TS-CLIP learns global time-series and language correspondences through contrastive alignment (Chen et al., 2025), and context alignment augments LLM time-series processing through context-level structural and logical alignment (Hu et al., 2025). CALF targets cross-modal fine-tuning for time-series forecasting (Liu et al., 2025), while TimeCAP uses LLM-generated contextual summaries for event outcome prediction (Lee et al., 2025). These approaches address sequence-level understanding, forecasting, or prediction rather than the exact start and end indices of a queried event. In contrast, GroundingMOMENT is designed for zero-shot span-level localization through joint primitive and event decoding.

Visual Grounding and Open-Vocabulary Detection. Cross-modal grounding is well established in computer vision. Grounding DINO (Liu et al., 2024) extends the DINO (Zhang et al., 2023) detector with early cross-modal fusion and language-guided query selection, serving as the primary structural precedent for our architecture.

Adapting this paradigm to 1D temporal grounding requires addressing three key discrepancies: (i) time series lack large-scale, densely annotated grounding corpora; (ii) temporal events are defined by ordered sequences of structural primitives rather than holistic spatial labels, requiring the model to learn not only the semantic-morphological correspondence between phrases and waveform shapes but also the sequential context of each primitive relative to its neighbors; and (iii) detections are inherently hierarchical. Primitives compose events, and events decompose into primitives. Therefore,

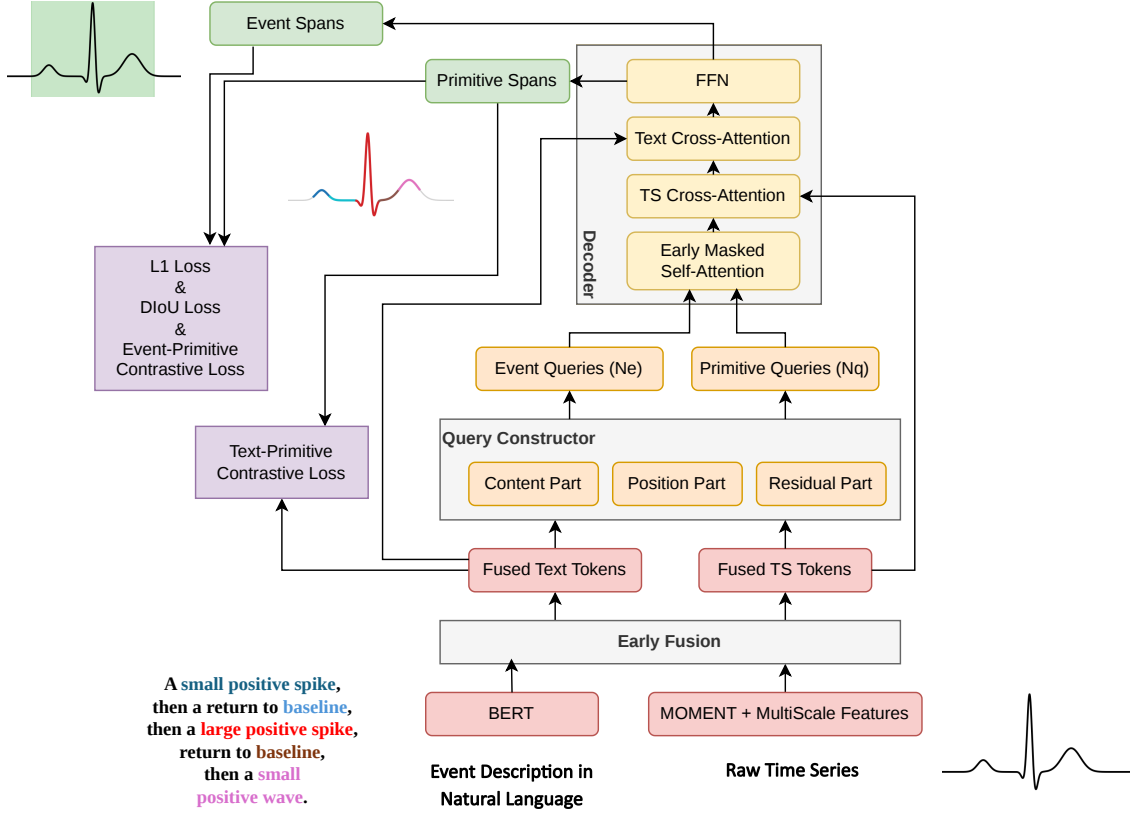


Figure 2: Overall architecture of GroundingMOMENT

the two levels are mutually dependent and must be learned jointly.

3 Task Formulation

Events and Primitives. An *event* $\mathcal{E}$ is an abstract temporal concept described by a natural-language query Q , decomposed into an ordered tuple of K *primitives* $(p_1, \dots, p_K)$ each characterizing a fundamental temporal behavior (e.g., “rises sharply”). An *event occurrence* within a time series $\mathbf{x} \in \mathbb{R}^L$ is a span (s, e) whose internal structure is a contiguous, ordered partition of K *primitive occurrences* $((s_1, e_1), \dots, (s_K, e_K))$ satisfying $s = s_1 < e_1 = s_2 < \dots < e_K = e$.

Task. Given a time series $\mathbf{x} \in \mathbb{R}^L$ and an event description Q , the model predicts event spans $\mathcal{D}_{\text{event}} = \{(s_i, e_i)\}_{i=1}^N$ ($N = 0$ indicates event absence) and, optionally, their constituent primitive spans $\mathcal{D}_{\text{prim}} = \{((s_{i1}, e_{i1}, p_{i1}), \dots, (s_{iK}, e_{iK}, p_{iK}))\}_{i=1}^N$. The learning objective jointly minimizes span localization and classification errors. The classification error supervises alignment between each pre-

dicted primitive span (s_{ik}, e_{ik}) and the corresponding phrase in description (p_{ik}) .

4 GroundingMOMENT: Architecture

GroundingMOMENT extends the Detection Transformer (DETR) (Carion et al., 2020) with language-guided query selection and cross-modal decoding inspired by Grounding DINO (Liu et al., 2024), adapted to the 1D temporal domain (Figure 2).

Time-Series and Language Encoders. Given a univariate time series and a primitive-decomposed event description (Table 1), dedicated pre-trained backbones encode each modality. We employ MOMENT (Goswami et al., 2024) as the foundational time-series encoder. MOMENT processes the input signal $\mathbf{x} \in \mathbb{R}^L$ via patch embedding, dividing it into 64 non-overlapping patches of size 8.

To effectively capture temporal events across varying resolutions, we construct a multi-scale feature hierarchy from $\mathbf{F}_0$ (the patch-level MOMENT output; see Appendix C) through a sequence of L

Domain	Event	Measurement	Description
Drilling	Connection	Surface Torque (STOR)	A medium spike (detach pipe), followed by a large sharp spike (reattach pipe), then a prolonged plateau (drilling).
Drilling	Connection	Block Position (BPOS)	A sharp increase (pick up pipe), then a short plateau (connection), then a smooth linear decrease (drilling).
Healthcare	Normal Sinus Beat	MLII	A smooth positive spike (P wave), and a large sharp positive spike (QRS complex), followed by a smooth positive wave (T wave).
Healthcare	AFIB	MLII	A prolonged plateau (missing P wave); a large sharp positive spike (QRS complex), with a smooth positive wave following (T wave).

Table 1: Examples of primitive-decomposed text descriptions across different domains. Each description is composed of an ordered sequence of structural primitives. Their practical meanings are explained in parentheses.

temporal convolutions with a stride of 2:

$$\mathbf{F}_\ell = \text{Conv1D}_{\text{stride}=2}^{(\ell)}(\mathbf{F}_{\ell-1}), \quad \ell = 1, \dots, L.$$

These multi-scale features are concatenated along the temporal dimension to form the complete time-series feature sequence $\mathbf{F} \in \mathbb{R}^{N_{\text{ts}} \times D}$, where $N_{\text{ts}} = \sum_{\ell=0}^L N_\ell$. For the textual modality, we use BERT (Devlin et al., 2019) as the language encoder.

Early Cross-Modal Fusion. The time-series and textual representations are mutually enriched via a bidirectional cross-modal feature enhancer comprising three operations: (i) *self-attention*, applied independently to each modality; (ii) *time-series-to-text cross-attention* enables temporal features to assimilate the semantic intent of the query; and (iii) *text-to-time-series cross-attention* grounds linguistic phrases within the physical signal. The resulting memories $\mathbf{M}_{\text{ts}} \in \mathbb{R}^{N_{\text{ts}} \times D}$ and $\mathbf{M}_{\text{txt}} \in \mathbb{R}^{T \times D}$ are deeply integrated cross-modal representations.

Language-Guided Query Initialization. *Primitive queries* are seeded from the top- N_p time-series tokens by token-level contrastive similarity $S_{\text{prim}}(i) = \max_j (\mathbf{M}_{\text{ts}}(i) \cdot \mathbf{M}_{\text{txt}}(j)^\top)$, anchoring each query to the patch most relevant to any individual phrase token.

Event queries require holistic matching against the full description rather than individual tokens. We condense $\mathbf{M}_{\text{txt}}$ into an attention-pooled summary vector $\mathbf{d} = \sum_j \alpha_j \mathbf{M}_{\text{txt}}(j)$, where $\alpha_j = \text{softmax}(\text{MLP}(\mathbf{M}_{\text{txt}}(j)))$ weights tokens by their contribution to the global event meaning. The top- N_e positions by $S_{\text{event}}(i) = \mathbf{M}_{\text{ts}}(i) \cdot \mathbf{d}^\top$ initialize the event query reference points.

Following mixed query selection (Zhang et al., 2023), each decoder query comprises a randomly initialized content vector $\mathbf{C}$, a positional encoding $\text{PE}(\mathcal{I})$ of the selected token index, and a type-specific learnable residual $\mathbf{E}$ that disambiguates

primitive from event queries:

$$\begin{aligned} \mathbf{Q}_{\text{prim}} &= \mathbf{C}_{\text{prim}} + \text{PE}(\mathcal{I}_{\text{prim}}) + \mathbf{E}_{\text{prim}}, \\ \mathbf{Q}_{\text{event}} &= \mathbf{C}_{\text{event}} + \text{PE}(\mathcal{I}_{\text{event}}) + \mathbf{E}_{\text{event}}. \end{aligned}$$

Reference points are initialized as $\hat{\mathbf{r}} = \sigma(\hat{\delta} + \mathbf{p}) \in [0, 1]^{(N_p + N_e) \times 2}$, where $\hat{\delta}$ is an encoder-predicted offset and $\mathbf{p}$ the positional proposal. We use a (*start*, *length*) rather than the conventional (center, width) parameterization, as temporal events are most sensitive to changes at the precise start and end boundaries.

Hierarchical Decoder with Compositional Attention. Because primitives and events are mutually dependent, neither a sequential primitive-first nor an event-first architecture is tenable: a primitive-first decoder cannot accurately localize primitives without knowing where events occur, while an event-first decoder cannot resolve event boundaries without reliable primitive anchors. We therefore adopt a unified cross-modal decoder that processes both query types simultaneously, enabling joint supervision across both levels.

This unified decoder simultaneously localizes the spans of the overarching event and its constituent primitives. A key challenge is the risk of out-of-distribution generalization failure if the model memorizes specific event-primitive combinations from raw waveforms. To mitigate this, we enforce a compositional inductive bias: event-level detection must be composed from the detected primitives. We implement this via dynamic attention masking. Let $\mathbf{A} \in \{0, -\infty\}^{(N_p + N_e) \times (N_p + N_e)}$ denote the self-attention mask for the concatenated queries $\mathbf{Q} = [\mathbf{Q}_{\text{prim}}; \mathbf{Q}_{\text{event}}]$. During a warm-up phase ($t < t_{\text{warmup}}$), the mask prevents event queries from attending to primitive queries ($\mathbf{A}_{i,j} = -\infty$ for $i \in \mathcal{I}_{\text{event}}, j \in \mathcal{I}_{\text{prim}}$), forcing

the model to first learn primitive localization independently. After t_{warmup} epochs, the mask is lifted ($\mathbf{A}_{i,j} = 0$), allowing event queries to aggregate features from well-localized primitive queries to infer event boundaries. Primitive queries are permanently prevented from attending to event queries throughout training ($\mathbf{A}_{i,j} = -\infty$ for $i \in \mathcal{I}_{\text{prim}}, j \in \mathcal{I}_{\text{event}}$, for all t), ensuring primitive localization can never be shortcut through event-level context, even after the warm-up phase ends.

Prediction Heads and Training Losses. **Primitive heads** predict span boundaries (L1 + 1D DIOU losses) and supervise semantic alignment between primitive queries and text tokens via bipartite Hungarian matching. **Event heads** predict global event boundaries via a separate span-only Hungarian matching, and an additional *event-primitive correspondence head* enforces the compositional link between the two query sets. This correspondence head activates only after the warm-up mask is lifted, ensuring event representations build on stable primitives. The training objective is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{prim}}^{\text{span}} + \lambda_2 \mathcal{L}_{\text{prim}}^{\text{cls}} + \lambda_3 \mathcal{L}_{\text{event}}^{\text{span}} + \lambda_4(t) \mathcal{L}_{\text{event}}^{\text{cls}}, \quad (1)$$

where $\mathcal{L}^{\text{span}}$ comprises L1 and DIOU (Zheng et al., 2020) penalties, $\mathcal{L}_{\text{event}}^{\text{cls}}$ is the event-primitive correspondence loss, and $\lambda_4(t)$ reflects the delayed activation of the correspondence head.

5 GroundTSSyn and GroundTSBench: Synthetic Data Generation

We present GroundTSSyn, a scalable pipeline for synthetic grounding data generation, and use it to construct GroundTSBench, the first open-source benchmark for language-guided time-series grounding. All annotations are programmatically derived, ensuring perfect ground-truth accuracy. Full pipeline details are provided in Appendix A. Comprehensive benchmark statistics appear in Appendix B.

5.1 Primitive Pool

We establish a vocabulary of six primitive types: *Increase*, *Decrease*, *Plateau*, *Spike*, *Dip*, and *Oscillation*. Each primitive is parameterized by (i) *generative features*, which are randomized prior to waveform synthesis and directly control waveform morphology (amplitude, duration, and type-specific stochastic parameters such as rise duration for spikes or frequency for oscillations), and by (ii) *derived features*, which are computed analytically

after waveform synthesis (e.g., slope, sharpness, cycle count) and serve as descriptors for natural language grounding. These features vary magnitude, steepness, duration, curvature and symmetry, without expanding the primitive vocabulary.

5.2 Signal Synthesis Pipeline

Signal synthesis proceeds through four stages. A *layout planner* first samples event counts, per-event primitive structure, inter-event gap structure, and durations for each primitive. Each primitive is then *synthesized* from its sampled generative features. Primitives are *concatenated* by aligning consecutive endpoints, z-score normalized, and optionally corrupted with white Gaussian noise. Two perceptual visibility scores, *primitive visibility* and *change-point visibility*, are computed post-normalization to identify boundaries a human observer might perceptually conflate. These scores guide downstream description generation without influencing synthesis. Finally, bounded stochastic *jitter* perturbs the generative features of each in-event primitive to produce multiple semantically equivalent occurrences per event template, providing the morphological variation present in real-world data.

To capture two prevalent structural morphologies, we additionally synthesize *periodic variants*, in which the canonical event pattern tiles the full signal with no inter-event gap, and *baseline variants*, in which the start and end values of each event and inter-event segment are constrained within a capped absolute difference.

5.3 Description Generation

Each generated signal is paired with natural-language descriptions via an agentic pipeline powered by a vision-language model (VLM). *Hard negatives* are first constructed by applying stochastic structural mutations to the canonical event template and verifying by exhaustive scan that the mutated pattern is absent from the signal. A *base description agent* then produces a canonical primitive-by-primitive paraphrase alongside fluent negative descriptions, varying the lexicons used. A *variant description agent* subsequently generates additional positive descriptions, increasing lexical and structural diversity. All candidate descriptions are reviewed by a *critic agent* that verifies description correctness.

During description generation, adjacent primitives are potentially merged into familiar abstrac-

Method	Event Level				Primitive Level				Lat. (s)
	mIoU@0.3	mIoU@0.5	mIoU@0.7	mAP	mIoU@0.3	mIoU@0.5	mIoU@0.7	mAP	
<i>TS-MLLMs</i>									
ChatTS (Xie et al., 2025)	5.3	2.4	1.0	2.3	6.7	4.4	1.5	4.0	10.8
TimeOmni-1 (Guan et al., 2026)	6.3	5.4	0.9	4.9	8.6	4.6	2.5	5.1	7.8
<i>LLM-Only</i>									
GPT-4.1 (Text)	30.8	24.6	10.1	22.6	44.0	36.5	21.9	32.5	4.5
GPT-5.4 (Text)	49.1	24.8	18.4	23.8	57.5	36.1	20.6	32.5	5.2
Qwen3-30B (Text) (Yang et al., 2025)	16.2	10.4	4.6	11.0	22.6	14.7	8.6	13.0	8.4
<i>Agentic (Wan et al., 2026)</i>									
SELA (Qwen3-30B)	54.6	38.5	27.5	41.2	59.9	41.5	27.4	40.5	38.4
SELA (GPT-4.1)	90.6	82.9	75.4	76.7	98.5	89.3	80.5	79.1	45.7
SELA (GPT-5.4)	92.1	85.2	80.4	83.6	98.8	95.1	89.2	88.3	49.0
<i>GroundingMOMENT (Ours)</i>									
BERT_L+MOMENT_B	85.2	82.6	77.5	79.2	98.4	96.5	91.5	88.7	1.2
BERT _B +MOMENT _B	81.2	80.8	78.8	77.1	96.6	95.2	89.8	85.0	1.0
BERT _L +MOMENT _S	68.6	63.5	57.4	55.0	75.4	69.4	57.2	58.1	0.9
BERT _B +MOMENT _S	65.3	61.6	55.8	55.9	70.4	63.7	52.9	54.7	0.7

Table 2: Main results on GroundTSBench. GroundingMOMENT achieves zero-shot accuracy comparable to the best agentic system while operating at $\sim 40\times$ lower latency. BERT_L/_B denotes BERT-large/base; MOMENT_B/_S denotes MOMENT-base/small. Best results per method group are **bolded**.

tions such as a *step-up* or a *table* by the agentic pipeline. Meanwhile, all temporal boundaries are analytically derived before description generation and are never modified by an LLM. The agents receive structured primitive types and feature summaries and vary only the surface wording of the corresponding grounding description.

5.4 Benchmark Overview

GroundTSBench comprises 16,410 samples, each pairing a synthesized signal with on average 8.8 descriptions (1 canonical, 3.4 positive variants, and 4.4 hard negatives). Signals exhibit 4.4 ± 2.0 event occurrences per sample, with each event composed of 4.7 ± 1.7 primitives, spanning a vocabulary of 4,441 tokens with a mean description length of 18.9 ± 7.7 words (Appendix B).

6 Experiments

GroundingMOMENT is trained on GroundTSBench (Section 5). MOMENT and BERT backbones are loaded from their respective pre-trained checkpoints. Full implementation details are provided in Appendix C.

6.1 Evaluation Protocol

The synthetic dataset is partitioned into training and validation splits. To evaluate event and primitive grounding performance, we report the mean Intersection-over-Union (mIoU) at thresholds $\theta \in \{0.3, 0.5, 0.7\}$ (denoted as mIoU@ θ) and the mean Average Precision (mAP) computed across the IoU range of $[0.3, 0.75]$ with a step size of 0.05. We also report inference latency in seconds per sample.

6.2 Baselines

We compare GroundingMOMENT against three classes of baselines, covering the space of available strategies for language-guided event detection (full prompting details in Appendix D).

TS-MLLMs. ChatTS (Xie et al., 2025) and TimeOmni-1 (Guan et al., 2026) are the most direct competitors. These models jointly process time-series and natural language. They establish whether existing multimodal time-series instruction-tuned models can address grounding without modification. Prompted baselines are instructed to return a strict JSON array of start-end spans. An output that cannot be parsed into this schema is treated as an explicit failure and receives zero on the grounding metrics.

LLM-Only Approaches. GPT-4.1, GPT-5.4, and Qwen3-30B-A3B (Yang et al., 2025) serve as strong text-only baselines, where the time series is serialized as text and passed to a general-purpose LLM. These baselines quantify how well frontier models achieve grounding through in-context reasoning alone, without time-series-specific architecture.

Agentic Time-Series Analysis. SELA (Wan et al., 2026) represents the agentic time-series analysis solution, where an agentic framework orchestrates VLMs within iterative ReAct loops. We evaluate it with both proprietary (GPT-4.1, GPT-5.4) and open-weight (Qwen3-30B (Yang et al., 2025)) backends.

Ablation	Ev. mIoU@0.7	Ev. mAP	Pr. mIoU@0.7	Pr. mAP
GroundingMOMENT (full)	77.5	79.2	91.5	88.7
w/o query-class residual	69.5	71.1	92.4	88.1
w/o separate ref-point	67.6	70.5	86.4	87.6
w/o permanent prim→event mask	72.5	71.0	82.3	82.4
w/o warm-up attention mask	71.9	73.2	85.0	86.3
w/o event class head	70.3	71.5	90.4	87.6
Non-hierarchical (event only)	62.8	67.4	—	—
(center, width) parameterization	70.9	74.4	87.6	82.5
Lexical holdout (train: no type names → test: type names)	73.4	76.1	91.7	87.9
Compositional length (train: <4 prim. → test: >5 prim.)	62.1	65.0	67.3	63.9

Table 3: Ablation and generalizability study for GroundingMOMENT (BERT_L+MOMENT_B). Best ablation results per column are **bolded**.

6.3 Main Results

Table 2 reports performance across event-level and primitive-level grounding metrics for all methods.

TS-MLLMs are insufficient for temporal grounding. ChatTS and TimeOmni-1 achieve event-level mIoU@0.3 of only 5.3 and 6.3 and mAP of 2.3 and 4.9, far below all other baselines. Both models are trained for question-answering and captioning rather than span detection. Their outputs are free-form text whose correspondence to temporal boundaries is incidental. Predicted spans frequently reflect the lexical content of the query rather than the signal morphology. This is a typical hallucination behavior driven by weak cross-modal alignment. Therefore, many outputs cannot be parsed into the required span schema. In an audit of ChatTS validation outputs, 70.0% violate the strict schema and only 20.0% contain parseable spans. This parseable subset reaches only 17.3 mAP, confirming that weak cross-modal alignment prevents fine-grained boundary localization (Appendix D).

LLM-only approaches achieve intermediate accuracy with coarse localization. Serializing the time series as an indexed two-column table and prompting a frontier LLM yields a large improvement over TS-MLLMs. GPT-4.1 and GPT-5.4 reach event-level mIoU@0.3 of 30.8 and 49.1; Qwen3-30B achieves 16.2, confirming that frontier model capability is a prerequisite. However, a pronounced gap between lenient and strict thresholds persists: GPT-5.4 reaches 49.1 at mIoU@0.3 yet only 18.4 at the 0.7 threshold, and its mAP of 23.8 is barely above GPT-4.1’s 22.6 despite the higher coarse recall. LLM-only approaches identify approximate regions of interest but struggle with precise boundary localization, a fundamental

limitation of discrete token-level encoding of continuous data. We also observe that the performance is sensitive to the input representation. Providing explicit timestep indices alongside signal values is essential for the reported accuracy. Therefore, LLM-only approaches depend heavily on the input format.

Agentic systems achieve the highest accuracy at prohibitive cost. SELA with GPT-5.4 and GPT-4.1 attains event-level mIoU@0.3 of 92.1 and 90.6 and mAP of 83.6 and 76.7, representing the empirical ceiling. The iterative ReAct loop enables multi-granularity inspection and self-correction, but at substantial cost: each query consumes $\approx 200,000$ tokens at 45 to 49 seconds per sample, making continuous telemetry monitoring impractical. SELA with Qwen3-30B (mIoU@0.3 of 54.6, mAP of 41.2) is notably weaker, underscoring the dependence of agentic approaches on frontier model capacity.

GroundingMOMENT achieves agentic-level accuracy at a fraction of the latency. BERT_L+MOMENT_B achieves event-level mIoU@0.3 of 85.2 and mAP of 79.2, approaching the empirical ceiling. At the primitive level it attains mAP of 88.7, *exceeding* SELA (GPT-5.4) at 88.3, at an inference latency of 1.2 seconds, roughly 40 $\times$ faster and without per-query token overhead. This makes GroundingMOMENT the only approach suitable for continuous telemetry monitoring.

The time-series backbone is the primary performance driver. Replacing MOMENT_B with MOMENT_S degrades event mAP by ≈ 24 points, whereas switching BERT_L to BERT_B causes a more modest 2–4 mAP reduction, indicating that scaling the time-series encoder yields higher returns than scaling the language encoder.

Domain	Method	mIoU@0.3	mIoU@0.5	mIoU@0.7	mAP	Lat. (s)
ECG	ChatTS	6.6	4.7	2.1	2.9	11.8
	GPT-5.4 (Text)	55.7	30.3	17.1	23.1	5.9
	Agentic (GPT-5.4)	76.8	69.7	60.0	65.3	55.6
	GroundingMOMENT	70.5	63.5	58.2	63.0	1.3
Drilling	ChatTS	3.2	1.4	0.6	0.9	11.5
	GPT-5.4 (Text)	60.7	42.1	29.6	34.2	5.5
	Agentic (GPT-5.4)	88.4	76.8	72.0	73.9	68.6
	GroundingMOMENT	81.1	70.1	63.6	67.2	2.5

Table 4: Real-world zero-shot grounding results. GroundingMOMENT is trained exclusively on GroundTSBench. Best results per domain are **bolded**.

6.4 Ablation and Generalizability Study

Table 3 summarizes an ablation and generalizability analysis comparing with BERT_L+MOMENT_B; full per-component analysis is provided in Appendix E.

Architectural components. The query-class residual and separate reference-point initialization each improve event-level mAP by ≈ 8 –9 points with minimal impact on primitives, confirming they act specifically on the event branch. The permanent prim $\rightarrow$ event mask yields the largest gain in primitive accuracy (+9.2 mIoU@0.7), as it permanently prevents primitive localization from being shortcut through event-level context. The warm-up attention mask contributes a complementary stabilization effect (+6.5 primitive mIoU@0.7): by briefly isolating primitives from events at the very start of training, it lets primitive representations converge before event queries begin drawing on them, avoiding the noisy early-training signal that would otherwise propagate into the event branch. The event classification head and the full hierarchical structure are jointly critical. Removing all primitive supervision (non-hierarchical) produces the worst event mAP (67.4), since the model must memorize event patterns without compositional grounding. The (start, length) parameterization outperforms (center, width) by 4.8 mAP at the event level, as temporal start and end boundaries carry change point salience.

Generalization. Training without canonical primitive-type names and testing exclusively on those names yields event/primitive mAP comparable to the full model (76.1/87.9 vs. 79.2/88.7), confirming vocabulary-agnostic grounding. Restricting training to events with <4 primitives and testing on >5 reveals a structural gap (event mAP 65.0), caused by the model overestimating span widths for shorter primitives it has not encountered;

compositional structure transfers, but boundary precision depends on training-distribution coverage of primitive durations.

7 Real-World Case Studies

To demonstrate cross-domain transferability, we evaluate GroundingMOMENT trained on GroundTSBench on real-world data without fine-tuning. Table 4 reports the quantitative comparison across ECG (Appendices F) and drilling (Appendices G). GroundingMOMENT remains within 2.3 and 6.7 mAP points of the agentic system on ECG and drilling, respectively, while operating $42.8\times$ and $27.4\times$ faster. ChatTS and text-only LLMs remain substantially weaker, whereas the agentic system attains its accuracy at prohibitive latency.

7.1 Cardiac Arrhythmia Detection (ECG)

We evaluate the MIT-BIH Arrhythmia Database (Moody and Mark, 2001) across 47 records containing 75,052 normal and 7,130 atrial fibrillation (AFIB) beats (Type V). Inference is performed with a sliding window of 1 000 samples (50% overlap), each interpolated to 512 points and z-score normalized. Using the descriptions from Table 1, GroundingMOMENT applies two queries to the same signal: the normal sinus beat query and the AFIB query. Figure 3 illustrates this description-driven discrimination on a representative segment from record 201: the normal sinus query yields no detections, whereas the AFIB query correctly localizes all four beats in the window.

7.2 Drilling Connection Detection

Drilling connections are the cyclic operations in which the drill string is extended by adding a new pipe stand. These are key operational efficiency bottlenecks in oil well construction. Each connection

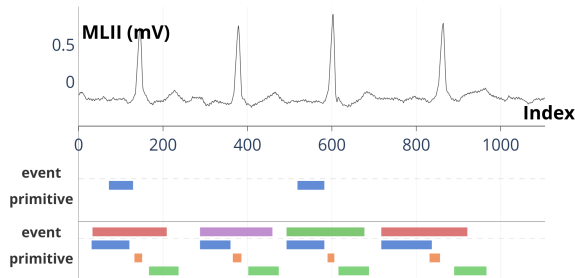


Figure 3: ECG case study on MIT-BIH record 201. Both panels show the same AFIB signal segment. **Top (normal sinus query)**: no events are detected—the expected P wave primitive is absent from the AFIB morphology, suppressing all detections. **Bottom (AFIB query, P wave replaced by a prolonged plateau)**: all four beats are correctly localized, with primitive spans for the plateau (blue), QRS complex (orange), and T wave (green) identified per beat. Applying two descriptions to identical input demonstrates GroundingMOMENT’s description-driven rhythm discrimination.

follows a mandatory three-phase sequence: pipe detachment, pipe pickup, and pipe reattachment.

We evaluate 52 days of multivariate telemetry⁴ containing 179 expert-annotated connection events, processing the surface torque and block-position channels independently with the descriptions from Table 1. The surface torque channel captures the dual-spike signature of detachment and reattachment, while the sharp increase in the block position channel traces the pickup. This evaluation demonstrates multi-channel transfer through one query per channel without architectural modification.

Per-phase duration analysis on this subset identifies reattachment as the highest-variance phase (CV = 85.0%), identifying it as the primary target for operational improvement. This finding is accessible only through primitive-level localization.

8 Conclusion

We proposed GroundingMOMENT, a zero-shot model that grounds primitive-decomposed text descriptions in time-series data. We introduced GroundTSSyn as a scalable data generation pipeline and GroundTSBench as the first open-source benchmark for this task. Together, they enable GroundingMOMENT to bridge domain expertise and raw telemetry, achieving high accuracy at low latency across diverse real-world domains.

⁴The drilling data are derived from the Utah FORGE Well 56-32 Drilling Data and Logs dataset: <https://catalog.data.gov/dataset/utah-forge-well-56-32-drilling-data-and-logs>.

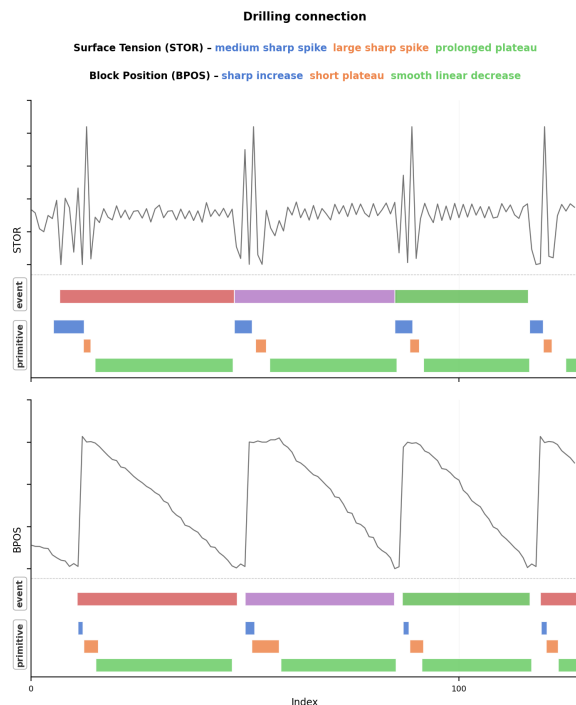


Figure 4: Drilling connection case study. **Top (STOR)**: the first medium spike (blue) marks pipe detachment; the second large spike (orange) marks reattachment; the prolonged plateau (green) spans the connection hold. **Bottom (BPOS)**: a sharp increase (blue) followed by a short plateau (orange) and a smooth decrease (green) captures the pipe pickup arc. Each channel is processed independently with its own query, demonstrating zero-shot multi-channel analysis.⁵

Limitations

Modality and input variability. While GroundingMOMENT adapts image-grounding techniques to time series, several modality differences remain. Time series often have variable sampling rates, whereas images have a fixed pixel grid. Time series possess a variable number of channels with domain-specific meanings (requiring permutation invariance), unlike the fixed RGB channels of images. Furthermore, time series vary widely in sequence length, and simple resampling distorts frequency information. Currently, GroundingMOMENT handles these issues via preprocessing and post-processing wrappers. Developing an architecture that natively accommodates these variations is an important direction for future work.

Linguistic variation and semantic scope. Our zero-shot claim concerns transfer across signal domains rather than invariance to every linguistic register. The description-generator study in Appendix B finds performance degradation for longer,

formal statistical phrasing, despite preserving semantic correctness. Training with a mixture of generators is a promising mitigation. Moreover, the present task targets morphological primitives and their temporal relations. Rich contextual constraints, such as comparisons with preceding amplitudes, require additional structured constraints; high-level domain concepts can be rewritten manually or by an LLM pre-compiler into a primitive sequence before grounding.

Deployment risks. The method is intended as decision support, not an autonomous control mechanism. In clinical ECG monitoring, false negatives for AFIB can delay intervention, while false positives can cause alarm fatigue. In drilling monitoring, missed connections can cause costly operational disruptions. Joint event and primitive predictions provide an auditable structural trace for expert review, but domain-specific validation and human oversight remain necessary before deployment.

Acknowledgments

We thank the members of the SLB AI Lab for their support and valuable feedback throughout this work. We are especially grateful to Josselin Kheroubi, Manager of SLB AI Lab, for providing the resources that made this research possible. We also thank the anonymous reviewers, for their thoughtful comments and constructive suggestions, which helped improve the paper.

References

- Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Türkmen, 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](#). *Trans. Mach. Learn. Res.*, 2024.
- Ane Blázquez-García, Angel Conde, Usue Mori, and José Antonio Lozano. 2022. [A review on outlier/anomaly detection in time series data](#). *ACM Comput. Surv.*, 54(3):56:1–56:33.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. [End-to-end object detection with transformers](#). In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, Lecture Notes in Computer Science, pages 213–229. Springer.
- Ziwen Chen, Xiaoyuan Zhang, and Ming Zhu. 2025. [TS-CLIP: time series understanding by CLIP](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 4646–4664. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. [Deep learning for time series classification: a review](#). *Data Min. Knowl. Discov.*, 33(4):917–963.
- Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. 2024. [Timegpt-1](#). *Preprint*, arXiv:2310.03589.
- Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. [MO-MENT: A family of open time-series foundation models](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 16115–16152. PMLR / OpenReview.net.
- Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuo Zhu Liu, Sabato Marco Siniscalchi, Ming Jin, and Shirui Pan. 2026. [Timeomni-1: Incentivizing complex reasoning with time series in large language models](#). *Preprint*, arXiv:2509.24803.
- Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. 2025. [Context-alignment: Activating and enhancing llms capabilities in time series](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- 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 re-programming large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Geon Lee, Wenchao Yu, Kijung Shin, Wei Cheng, and Haifeng Chen. 2025. [Timecap: Learning to contextualize, augment, and predict time series events with large language model agents](#). In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of*

- Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 18082–18090. AAAI Press.
- Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. 2025. [CALF: aligning llms for time series forecasting via cross-modal fine-tuning](#). In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 18915–18923. AAAI Press.
- Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. 2024. [Grounding DINO: marrying DINO with grounded pre-training for open-set object detection](#). In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLVII*, Lecture Notes in Computer Science, pages 38–55. Springer.
- George B Moody and Roger G Mark. 2001. The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3):45–50.
- Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. [A time series is worth 64 words: Long-term forecasting with transformers](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Sky Chenwei Wan, Yifei Y. Wang, Tianjun Hou, Xiqing Chang, and Aymeric Jan. 2026. [Grammar of the wave: Towards explainable multivariate time series event detection via neuro-symbolic vlm agents](#). *Preprint*, arXiv:2603.11479.
- Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. [Unified training of universal time series forecasting transformers](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 53140–53164. PMLR / OpenReview.net.
- Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. 2025. [Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning](#). *Proc. VLDB Endow.*, 18(8):2385–2398.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. 2023. [DINO: DETR with improved denoising anchor boxes for end-to-end object detection](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. 2020. [Distance-iou loss: Faster and better learning for bounding box regression](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 12993–13000. AAAI Press.

A GroundTSSyn Pipeline Details

A.1 Primitive Pool

Each primitive is parameterized by two classes of features.

Generative features directly control waveform synthesis and fall into three sub-categories: *definitive* features, whose values are fixed by the layout (start value and duration); *configurable-range* features, whose values are sampled at synthesis time (amplitude, i.e., peak-to-peak range); and *stochastic* features, drawn from per-type distributions—e.g., rise duration and recovery fraction for spikes; frequency, envelope behavior, and waveform shape for oscillations; curvature shape for ramps.

Derived features are computed analytically from the synthesized waveform (e.g., slope, sharpness, asymmetry, and cycle count) and serve as descriptors for downstream natural language grounding. They are not inputs to the synthesizer.

Although spikes, dips, and oscillations could theoretically be expressed as sequences of simpler ramps and plateaus, we retain them as atomic primitives because they constitute distinct, frequently named concepts in natural language descriptions of time-series data.

Under these settings, morphological diversity is expressed through parameter settings rather than additional primitive labels: magnitude, steepness, duration, and curvature or symmetry specify the continuous and categorical variation within each type. Complex morphological patterns may be generated by the description pipeline by combining adjacent primitives into abstractions, such as a *step-up*, which is a sequence of increase and plateau, or a *table*, which is a sequence of increase, plateau, and decrease.

A.2 Signal Synthesis Pipeline

Layout planning. Layout planning proceeds in three sequential stages. First, *counts* are sampled: the number of event occurrences K , the number of primitives per event P , and the number of background primitives in each of the $K+1$ inter-event gaps. Second, *types* are assigned for every primitive, using weighted-categorical sampling subject to adjacency constraints (e.g., two consecutive plateaus are forbidden). Third, *durations* are allocated: a total event length is drawn and sequentially subdivided among the P in-event

primitives respecting per-type minimum lengths; the remaining signal budget is similarly partitioned among the inter-event segments. Throughout, the total length of the signal T is maintained.

Feature sampling and realization. With the layout fixed, definitive generative features are bound: the start value is fixed to zero (the absolute level will be established later by concatenation shifting), and the duration is taken directly from the layout. Each primitive is realized in two phases. In the first phase, the amplitude is independently sampled from configuration-defined ranges. Stochastic generative features are drawn according to the per-type schema. For example, a spike draws rise duration and recovery fraction, while an oscillation draws frequency, envelope behavior, and waveform shape. In the second phase, the waveform is synthesized from these inputs and derived features are computed from the resulting sample values. A geometric validity check is then applied: the slope must change at the junction of every pair of adjacent primitives, ensuring a discernible inflection point at each transition.

Concatenation, shifting, and normalization. Primitives are sequentially concatenated by aligning consecutive endpoints at their shared boundary sample. Because every primitive was synthesized from a start value of zero, a running shift is applied so that each primitive’s initial value matches the preceding one’s terminal value. The assembled signal is then z-score normalized, after which all features are recomputed from the normalized waveform. This recomputation requires tracking whether each feature is *level-dependent*, such as absolute amplitude values, which change across occurrences as the local baseline shifts; or *level-independent*, such as duration, slope ratio, and frequency, which are invariant to baseline offsets. The distinction is essential for the subsequent jitter stage: only level-independent features are perturbed to generate new occurrences, while level-dependent features are re-derived from the normalized signal after each re-synthesis.

Noise injection and perceptual visibility. White Gaussian noise is optionally injected into the normalized signal. The noise standard deviation σ informs two perceptual scores assigned to each primitive post-normalization. The *primitive visibility* score measures how detectable a primitive is in isolation: for spikes and dips it is a sigmoidal

function of the amplitude-to-noise ratio; for ramps, plateaus, and oscillations it additionally accounts for duration saturation and neighbor masking. The *change-point visibility* score measures how clearly the boundary between two adjacent primitives appears as a visible inflection, combining slope mismatch and level mismatch between the two primitives relative to σ . These scores do not influence signal synthesis; they guide the description generation pipeline by identifying which primitives a human observer might perceptually conflate.

Occurrence variation via jitter. A single event template produces only one canonical realization, yet a grounding model must learn to associate a description with the event regardless of natural variation in amplitude or timing. To produce multiple distinct yet semantically equivalent occurrences, we apply bounded stochastic perturbations to the generative features of each in-event primitive, including durations, amplitudes, and other stochastic parameters. Each perturbed primitive is re-synthesized and validated to confirm that all previously defined geometric and layout constraints remain satisfied. The result is a set of signal occurrences that share the same abstract event identity but differ in their fine-grained waveform details, providing the model with the variability present in real-world data.

A.3 Periodic and Baseline Signal Variants

Empirical observations of time series highlight two prevalent structural morphologies.

Periodic signals. Numerous physiological and mechanical processes generate cyclically repeating event patterns, such as R-R intervals in electrocardiograms or valve actuation cycles in mechanical drilling. To model these, we explicitly synthesize periodic variants wherein the canonical event pattern tiles the full signal length, with no inter-event gap.

Signals with baseline. Sensor measurements in real-world environments frequently exhibit variation around a baseline. We simulate this by imposing a capped absolute difference between the starting value and ending value of each event occurrence and each inter-event primitive sequence. If the threshold is set to 0, no baseline shift is allowed.

A.4 Description Generation

Negative grounding generation. A grounding model trained only on positive examples could trivially learn to always predict a match; hard negatives are therefore essential. We generate them by applying stochastic mutation to the canonical event template: removing an interior primitive, substituting a primitive type, or swapping two non-adjacent primitives. Each mutated template is verified by exhaustive scan to confirm it does not align with any contiguous segment of the signal. Accepted mutations are forwarded to the description agents as negative seeds.

Base description agent. The base agent operates on structured text only. It receives the primitive type sequence and bucketized feature summaries, and produces a canonical paraphrase: a short primitive phrase per position, assembled in template order. In the same pass, it paraphrases each negative seed, producing fluent descriptions of events that are structurally absent from the signal. The agent varies the lexicons used.

Variant description agent. To increase lexical and structural diversity, a separate multimodal agent generates multiple positive variant descriptions per sample. In a first *observation* stage, the agent receives the global signal plot, a zoomed view of one event occurrence, and the precomputed primitive and change-point visibility scores with candidate perceptual operations. Based on this visual evidence, it proposes abstractions a human observer might naturally adopt: merging two adjacent primitives whose shared boundary has low change-point visibility into a single description chunk, splitting a spike or dip whose onset blends into a neighboring ramp into two sub-phrases, or reinterpreting a low-visibility primitive under an alternative type (e.g., a very gradual rise as a plateau). In a second *generation* stage, a subset of these observations is applied to the base description to produce each variant.

Quality review. All candidate descriptions—including base, variants, and negatives—are reviewed by a multimodal critic agent. The critic receives the global signal plot alongside each description with its claimed grounding spans and verifies that every primitive phrase names the correct type and direction, that quantitative claims are visually consistent with the signal in the grounded span, and that the temporal ordering of spans is

correct. The critic assigns a verdict of either accept, borderline, or reject. For spans that are difficult to assess from the global view, the critic may request a single zoomed slice before issuing a verdict. Descriptions receiving either a borderline or reject verdict are discarded. Only accepted descriptions are included in the released training and evaluation sets.

B GroundTSBench Dataset

GroundTSBench is constructed by running GroundTSSyn (Appendix A) at scale. The resulting 16,410 samples span a broad range of event morphologies and description styles. Signals are drawn from three structural categories: 40% are free-form signals with no inter-event constraints; 30% are periodic variants in which the event pattern tiles the full signal length; and 30% are baseline-bounded variants in which consecutive event and inter-event segments are constrained to remain within a capped absolute level difference. Table 7 reports the comprehensive statistics of GroundTSBench.

B.1 Primitive Type Distribution

Table 5 reports the instance counts for each primitive type across all in-event primitives.

Primitive	Instances	%
Decrease	20,416	26.5
Increase	20,174	26.2
Plateau	14,191	18.4
Dip	8,788	11.4
Spike	8,113	10.5
Oscillation	5,448	7.1
Total	77,130	100

Table 5: Primitive type distribution across all in-event primitives in GroundTSBench (16,410 samples).

B.2 Per-Type Vocabulary

Table 6 lists the top-10 most frequent content words in the grounding phrases per primitive type.

Primitive	Top-10 words
Increase	upward, ramp, rise, steady, climb, rising, stretch, gradual, gentle, upswing
Decrease	downward, drop, slope, decline, stretch, slide, descending, descent, falling, gentle
Plateau	flat, stretch, level, segment, lasting, brief, steady, rise, patch, baseline
Spike	upward, sharp, sudden, jolt, brief, abrupt, peak, spike, quick, rise
Dip	downward, brief, dip, trough, notch, drop, quick, sharp, small, low
Oscillation	wavy, segment, stretch, section, undulating, wavering, rippling, oscillating, patch, oscillation

Table 6: Top-10 most frequent content words (stop-words removed) in the grounding phrases per primitive type, across all 16,410 synthesised samples (positive descriptions only).

B.3 Description Generator Robustness

We evaluate the BERT_B+MOMENT_B model on held-out test sets whose descriptions are independently generated by GPT-5.4, GPT-5.2, GPT-5.4-mini, GPT-4.1, and GPT-4.1-mini. The underlying signals, event spans, primitive spans, and evaluation protocol are held fixed; only the description generator changes.

We conduct a manual audit of 100 descriptions from each generator (500 descriptions total). Every audited description preserves the programmatically specified primitive types, temporal order, and feature behaviors. Thus, performance differences in Table 8 cannot be attributed to semantic hallucinations or incorrect temporal ground truth.

Generator-dependent performance. GPT-5.4 and GPT-5.2 closely match the default description distribution: their event mAP values are 77.1 and 75.8, and their primitive mAP values are 85.0 and 84.5, respectively. GPT-5.4-mini produces a moderate decrease to 66.0 event mAP and 72.6 primitive mAP. In contrast, GPT-4.1 and GPT-4.1-mini reduce event mAP to 51.2 and 42.4 and primitive mAP to 51.1 and 42.4. The larger decline at both levels indicates that linguistic variation affects not only the composition of an event description but also the model’s alignment of individual primitive phrases to waveform morphology.

Register drift. The observed differences reflect a shift in register rather than a change in semantics. GPT-5.4 and GPT-5.2 generally use concise qualitative phrases, such as *a brief dip followed by a steady rise*; GPT-4.1 variants more often use formal statistical constructions, such as *exhibiting asymmetry at the 75th percentile*, and produce descriptions approximately 15% longer. The shared frequent vocabulary and the manual audit support this interpretation: the descriptions remain correct, but their phrasing departs from the training distribution. Consequently, the zero-shot capability demonstrated here is strongest for domain transfer under structurally equivalent descriptions, rather than complete invariance to arbitrary linguistic styles. Training with mixtures of generators is a natural direction for improving robustness to this register drift.

Statistic	Value
<i>Composition</i>	
Total samples	16,410
Train / Val (90 / 10 split)	14,769 / 1,641
Signal variants (free / periodic / baseline)	40 / 30 / 30 %
Base / pos. variant / negative desc.	1 / 3.4 / 4.4 per sample
Critic verdict (accept / borderline / reject)	44.9 / 42.7 / 12.4 %
<i>Signal structure</i>	
Signal length (timesteps)	512
Noisy samples ($\sigma > 0$)	66.7 %
Noise σ range	[0.01, 0.05]
Event occurrences per sample	4.4 ± 2.0
Event coverage (fraction)	0.71 ± 0.21
<i>Primitive structure</i>	
Primitives per event	4.7 ± 1.7 (range 2–7)
Event span (timesteps)	82.2 ± 44.9
<i>Language</i>	
Vocabulary size	4,441 tokens
Description length (words)	18.9 ± 7.7
Variant ops (merge / reinterpret / split)	65.2 / 25.4 / 9.4 %
<i>Hard negatives</i>	
switch_type / swap / remove	43.6 / 32.3 / 24.1 %

Table 7: Comprehensive statistics of GroundTSBench. Critic verdicts and variant operations apply to positive descriptions only; the reject rate is reported over pre-filtered candidates, and only accepted descriptions enter the benchmark.

Description Generator	Event Level				Primitive Level			
	mIoU@0.3	mIoU@0.5	mIoU@0.7	mAP	mIoU@0.3	mIoU@0.5	mIoU@0.7	mAP
GPT-5.4 (default)	81.2	80.8	78.8	77.1	96.6	95.2	89.8	85.0
GPT-5.2	79.9	79.2	77.2	75.8	95.2	94.4	90.2	84.5
GPT-5.4-mini	70.0	69.2	67.3	66.0	82.4	81.2	76.7	72.6
GPT-4.1	51.8	54.1	52.2	51.2	58.6	57.4	53.3	51.1
GPT-4.1-mini	44.7	44.1	42.5	42.4	48.6	47.6	44.3	42.4

Table 8: Description-generator robustness on GroundTSBench. Best result per column is **bolded**.

C Training Configuration

C.1 Architecture

Both modalities are projected to a shared hidden dimension of $D = 256$.

Time-series encoder. MOMENT processes a fixed-length input of 512 timesteps as 64 non-overlapping patches of size 8, producing per-patch embeddings of dimension $d_{\text{MOMENT}} = 1024$ (base) or $d_{\text{MOMENT}} = 384$ (small), which are linearly projected to D . A multi-scale feature hierarchy is constructed from the base level $\mathbf{F}_0 \in \mathbb{R}^{64 \times D}$ by applying 6 successive stride-2 Conv1D layers, yielding levels $\ell = 1, \dots, 6$ with sequence lengths 32, 16, 8, 4, 2, 1. All 7 levels are concatenated to form the complete time-series feature sequence of $N_{\text{ts}} = 127$ tokens.

Language encoder. BERT-base ($d_{\text{BERT}} = 768$) or BERT-large ($d_{\text{BERT}} = 1024$) tokenizes the event description with a maximum sequence length of 128 tokens; the output embeddings are projected to D .

Feature enhancer. The bidirectional cross-modal feature enhancer comprises 6 transformer layers with 8 attention heads, FFN dimension 1024, and dropout 0.1.

Decoder. The hierarchical cross-modal decoder comprises 6 layers with 8 attention heads, FFN dimension 1024, and dropout 0.1. The span prediction head is shared across all decoder layers (and with the encoder’s two-stage output). The decoder is initialized with $N_p = 64$ primitive queries and $N_e = 16$ event queries selected from the 127 available time-series tokens.

C.2 Training

Optimizer. AdamW is used with three differential learning rates: 10^{-4} for all non-backbone parameters, 10^{-5} for the MOMENT time-series encoder, and 10^{-5} for the BERT text encoder. Weight decay of 10^{-4} is applied to all parameters except biases and LayerNorm weights.

Learning rate schedule. Learning rates follow a linear warmup over 3 epochs followed by cosine annealing to zero over the remaining 197 epochs (200 epochs total).

Backbone unfreeze schedule. Both the MOMENT and BERT backbones are frozen for the first 5 epochs. At epoch 5 their parameters are

unfrozen and added to the optimizer at their respective backbone learning rates, allowing stable decoder warm-up before committing to end-to-end fine-tuning.

Batch and hardware. Training uses a batch size of 32 and mixed-precision computation (bfloat16). Gradients are clipped to a maximum ℓ_2 -norm of 1.0. A CUDA-stream prefetcher overlaps host-to-device transfers with GPU computation. Distributed training over multiple GPUs uses PyTorch Distributed Data Parallel (DDP) with the NCCL backend; single-GPU experiments use standard DataLoader with 4 worker processes. All training and GroundingMOMENT inference experiments are conducted on a node equipped with 2×NVIDIA H100 (80 GB) GPUs and Intel Xeon Gold CPUs.

Reproducibility. All experiments use random seed 42.

C.3 Loss Configuration

Primitive detection losses. Primitive spans are matched to ground-truth primitives via bipartite Hungarian matching with costs: classification $\lambda_{\text{cls}}^{\text{match}} = 5$, L1 $\lambda_{\text{L1}}^{\text{match}} = 10$, and DIOU $\lambda_{\text{IoU}}^{\text{match}} = 2$. The training losses are weighted as: L1 regression ($\lambda_{\text{L1}} = 10$), DIOU regression ($\lambda_{\text{DIOU}} = 2$), and focal contrastive classification ($\lambda_{\text{cls}} = 1$; focal $\alpha = 0.25$, $\gamma = 2.0$).

Event detection losses. Event spans are matched via a separate Hungarian step with costs L1 (= 10) and DIOU (= 2), matching the span-only supervision described in Section 4. Training losses use the same λ_{L1} and λ_{DIOU} weights as the primitive branch.

Event-primitive correspondence loss ($\mathcal{L}_{\text{event}}^{\text{cls}}$). The event classification term in Eqn. 1 is realised by an *event-primitive correspondence head*. The supervision target is a binary positive map $\mathbf{Y} \in \{0, 1\}^{N_e \times N_p}$ derived deterministically from the two independently-solved span-matching assignments already used for the span losses: for each matched event query $e \leftrightarrow$ ground-truth event k_e and each matched primitive query $p \leftrightarrow$ ground-truth primitive belonging to event k_p , we set $\mathbf{Y}_{e,p} = 1$ whenever $k_e = k_p$, and zero otherwise. The loss is a focal binary cross-entropy over $\mathbf{S}$ against $\mathbf{Y}$ with the same focal parameters as the primitive classification loss. Its overall weight $\lambda_4(t)$ is gated by the warm-up schedule. $\lambda_4(t) = 0$

for $t < t_{\text{warmup}}$ and $\lambda_4(t) = 1$ thereafter. The correspondence head only receives gradient signal once the primitive and event branches have stabilised. The same head and loss are applied to every intermediate decoder layer as auxiliary supervision.

Auxiliary supervision. Intermediate decoder layers and the encoder output each receive auxiliary span supervision (same loss weights), which stabilizes gradient flow through the full decoder stack.

Denoising training. Contrastive DeNoising (CDN) (Zhang et al., 2023) is applied: 2 denoising groups are constructed per sample with label noise ratio 0.5 and span noise scale 0.4, contributing to training with a denoising loss weight of 1.0.

Warm-up attention mask. Event queries are blocked from attending to primitive queries in self-attention for the first 8 epochs (the *event isolation* warm-up phase described in Section 4). Primitive queries are permanently blocked from attending to event queries.

C.4 Inference

At inference time, predictions are filtered by a classification score threshold of 0.5. A 1D non-maximum suppression with IoU threshold 0.5 is applied over primitive spans within each phrase group to remove duplicates. The decoder is seeded with the top- $N_p = 64$ primitive and top- $N_e = 16$ event queries selected by language-guided query initialization.

C.5 Summary Table

Table 9 consolidates all architectural dimensions, training settings, loss weights, and inference parameters for easy reference.

Category	Hyperparameter	Value
<i>Architecture</i>		
	Shared hidden dimension D	256
	Input length (timesteps)	512
	Patch size	8
	Number of patches (L0)	64
	Multi-scale feature levels	7 (L0–L6)
	Total TS tokens N_{ts}	127
	MOMENT output dim (base / small)	1024 / 384
	BERT output dim (large / base)	1024 / 768
	Max text token length	128
	Feature enhancer layers / heads	6 / 8
	Feature enhancer FFN dim	1024
	Feature enhancer dropout	0.1
	Decoder layers / heads	6 / 8
	Decoder FFN dim	1024
	Decoder dropout	0.1
	Primitive queries N_p	64
	Event queries N_e	16
<i>Training</i>		
	Optimizer	AdamW
	Learning rate (non-backbone)	10^{-4}
	Learning rate (MOMENT backbone)	10^{-5}
	Learning rate (BERT backbone)	10^{-5}
	Weight decay	10^{-4}
	LR warmup epochs	3
	Total epochs	200
	LR schedule	Cosine annealing
	Batch size	32
	Gradient clipping (ℓ_2 -norm)	1.0
	Mixed precision	bfloat16
	Backbone freeze epochs	5
	Event isolation warm-up epochs	8
<i>Loss weights</i>		
	Hungarian cost: classification λ_{cls}^{match}	5
	Hungarian cost: L1 λ_{L1}^{match}	10
	Hungarian cost: DIoU λ_{IoU}^{match}	2
	Training loss: primitive span L1 λ_{L1}	10
	Training loss: primitive span DIoU λ_{DIoU}	2
	Training loss: primitive classification λ_{cls}	1
	Training loss: event span L1 / DIoU λ_3	10 / 2
	Training loss: event classification $\lambda_4(t)$	$0 \ (t < t_{warmup}), 1 \ \text{otherwise}$
	Focal loss α / γ	0.25 / 2.0
	Denoising groups	2
	Denoising label noise ratio	0.5
	Denoising span noise scale	0.4
	Denoising loss weight	1.0
<i>Inference</i>		
	Classification score threshold	0.5
	NMS IoU threshold	0.5

Table 9: Complete hyperparameter and configuration summary for GroundingMOMENT. Values correspond to the BERT_L+MOMENT_B variant; backbone output dimensions for other variants are listed in the Architecture block.

D Baseline Prompting Details

All prompted baselines receive a unified five-component prompt structure: (i) the serialized time-series signal, (ii) a plain-English description of the event to be detected, (iii) the detection objective, (iv) a brief contextual framing of the grounding task, and (v) a specification of the required output format. Models are explicitly instructed to produce a JSON array of temporal span objects of the form `[{"start": s, "end": e}, ...]`, where indices are expressed in the original timestep domain. Responses that cannot be parsed into this schema are treated as failures and receive zero on the grounding metrics.

TS-MLLMs. ChatTS (Xie et al., 2025) receives raw numpy arrays as a native time-series modality: patches are encoded via a lightweight MLP and concatenated with text tokens at the token level, bypassing textual serialization entirely. TimeOmni-1 (Guan et al., 2026) adopts a text-based strategy, serializing the signal as a comma-delimited numerical sequence (e.g., `[2001.46, 2003.03, ...]`) in component (i).

ChatTS output parsing audit. On the validation set, 70.0% of ChatTS responses violate the strict JSON schema, 10.0% are valid empty arrays, and 20.0% contain parseable spans. Of the invalid responses, 95.7% wrap the array in conversational prose and 4.3% use incorrect JSON keys. The parseable subset attains mIoU@0.3/@0.5/@0.7 of 22.1/19.0/15.0 and mAP of 17.3. Thus, schema violations are frequent, but correcting them alone would not resolve the underlying boundary-localization error.

LLM-Only Approaches. GPT-4.1, GPT-5.4, and Qwen3-30B (Yang et al., 2025) receive the same five-component prompt structure, with one key difference in the time-series representation: the signal is serialized as an indexed two-column tabular format, where each row contains a timestep index in the first column and the corresponding signal value in the second. Among several serialization strategies evaluated, this indexed tabular representation consistently yields the best localization accuracy; omitting the explicit index column produces markedly inferior results. All reported LLM-only numbers adopt the indexed tabular format. Qwen3-30B serves as the primary open-weight representative.

Agentic Time-Series Analysis. SELA (Wan et al., 2026) orchestrates VLM agents within iterative ReAct loops, accessing time-series data exclusively through active visualization tools (e.g., `view_window` calls that render signal plots). The agents never receive the raw signal as a serialized array; instead they inspect the waveform visually and refine temporal localizations by zooming into sub-windows augmented with view-dependent normalization.

Latency measurement. Per-sample latency is measured end-to-end, including serialization and output parsing. GPT-4.1, GPT-5.4, and SELA variants that use these backends are served via Azure OpenAI; their latency includes the full round-trip over a 100 Mbps network connection. Qwen3-30B, ChatTS, TimeOmni-1, and GroundingMO-MENT are executed locally on the 2×NVIDIA H100 (80 GB) / Intel Xeon Gold node described in Appendix C.

E Ablation and Generalizability Study Details

Full per-component analysis for the ablation and generalizability study summarized in Section 6.4 (Table 3).

Event query design. Both the query-class residual connection and the separate reference-point initialization for event queries primarily affect event-level performance, with primitive-level metrics remaining largely stable in their absence. Removing the query-class residual reduces event mIoU@0.7 from 77.5 to 69.5 and event mAP from 79.2 to 71.1, while primitive mIoU@0.7 and mAP are nearly unchanged at 92.4 and 88.1, respectively. A comparable pattern holds for the separate reference-point mechanism: its removal degrades event mIoU@0.7 to 67.6 and event mAP to 70.5, with primitive metrics showing a more moderate decline to 86.4 and 87.6. These results confirm that the two mechanisms act specifically on the event branch, providing distinct initializations that allow event queries to seek holistic event boundaries independently of the primitive queries.

Primitive and Event Isolation Masks. Two independent self-attention masks isolate the primitive and event branches, and we ablate each in turn. Removing the *permanent* prim→event mask—which otherwise prevents primitive queries from ever attending to event queries—has the most pronounced effect on primitive-level accuracy, reducing primitive mIoU@0.7 from 91.5 to 82.3 and primitive mAP from 88.7 to 82.4, with event mAP dropping from 79.2 to 71.0. Without this permanent block, primitive localization can be shortcut through event-level context at any point in training, allowing primitive queries to inherit the same memorized event-primitive combinations that the mask was designed to prevent, rather than remaining independently grounded in the raw waveform. Removing the *warm-up* attention mask—which blocks event queries from attending to primitive queries only for the first 8 epochs—produces a smaller but still substantial degradation (primitive mIoU@0.7 85.0, primitive mAP 86.3, event mAP 73.2). This stems from training instability: without the brief isolation period, event queries begin drawing on primitive representations before those representations have stabilized, propagating noisy early-training gradients into both branches and preventing stable convergence. The two masks are complementary: the

permanent mask protects primitive independence for the entire run and accounts for the majority of the primitive-accuracy gain, while the warm-up mask specifically protects the earliest, most volatile phase of joint training and contributes an additional stabilization benefit on top.

Event classification head and compositional generalization. Omitting the event classification head, which supervises alignment between projected primitive queries and event queries, reduces event mIoU@0.7 to 70.3 and event mAP to 71.5, while leaving primitive metrics largely intact. The more severe degradation appears in the non-hierarchical variant, which removes all primitive supervision and trains a single decoder to predict event spans directly from time-series features. This variant attains only 62.8 event mIoU@0.7 and 67.4 event mAP—the lowest event-level performance of any configuration. Without compositional grounding through primitives, the model must memorize event-level patterns directly from the waveform, requiring exhaustive coverage of all event morphologies in training data. Since such coverage is infeasible at scale, the model fails to transfer to unseen primitive sequences. Hierarchical primitive-to-event supervision is therefore not merely an auxiliary signal, but the mechanism through which the model achieves compositional generalization.

Span parameterization. Replacing the (start, length) parameterization with the conventional (center, width) representation used in image-domain detectors yields a consistent drop across both event and primitive metrics, with event mAP decreasing from 79.2 to 74.4 and primitive mAP from 88.7 to 82.5. Temporal events are particularly sensitive to the precise location of their onset change-point; the (start, length) formulation anchors one prediction coordinate directly to this boundary, whereas the (center, width) formulation distributes localization error symmetrically around the midpoint, which is less aligned with the asymmetric perceptual salience of temporal transitions.

Lexical Holdout. We test whether grounding generalizes across vocabulary by removing from the training set all descriptions containing any of the six canonical primitive-type names (*spike*, *increase*, *decrease*, *plateau*, *oscillation*, *dip*) and evaluating exclusively on descriptions that use these names. The lexical holdout model achieves 76.1 event mAP and 87.9 primitive mAP, comparable

to the full model’s 79.2 and 88.7. This confirms that the model has learned the semantic meaning of primitive descriptions rather than relying on specific surface forms: the cross-modal feature enhancer associates waveform morphology with embedding-space representations of primitive concepts, which remain consistent across paraphrases and canonical type names alike.

Compositional Length Generalization. We evaluate whether the model composes primitive detectors for event lengths unseen during training by restricting training to events with fewer than 4 constituent primitives and evaluating on events with more than 5 primitives. The held-out model achieves 65.0 event mAP and 63.9 primitive mAP, a notable drop from the full model. Analysis reveals that the model tends to predict longer primitive spans for longer events: because events with more constituent primitives are synthesized with shorter individual primitives on average, the model—trained only on wider primitive spans—overestimates each span’s duration, accumulating boundary errors across the sequence. The compositional structure generalizes, but precise boundary localization depends on training-distribution coverage of primitive durations.

F Healthcare Case Study Details

Database. The aggregate evaluation covers 47 records from the MIT-BIH Arrhythmia Database (Moody and Mark, 2001), comprising 75,052 normal and 7,130 AFIB beats. Record 201 is a representative detailed subset from the standard public benchmark of half-hour two-lead ECG recordings digitised at 360 Hz with 11-bit resolution over a ± 5 mV range. It contains episodes of both normal sinus rhythm and atrial fibrillation (AFIB), making it well-suited for illustrating description-driven rhythm discrimination. We use the MLII lead, which provides the clearest P-QRS-T morphology.

Signal statistics. The full record spans 650 000 samples (≈ 30 min). The displayed segment spans rows 103 735–104 839 (≈ 3 s), a window that contains several consecutive heartbeat cycles with clearly visible P, QRS, and T components.

Preprocessing. A sliding window of 1 000 samples (2.78 s at 360 Hz) is applied with 50% overlap (500-sample stride). Each window is linearly interpolated to 512 points and z-score normalized before being passed to the model.

Description templates. The two natural-language descriptions (Table 1) used for inference are:

- **Normal sinus beat:** “A smooth positive spike; a large sharp positive spike; a smooth positive wave.”
Primitives: P wave $\rightarrow$ QRS complex $\rightarrow$ T wave.
- **AFIB:** “A prolonged plateau; a large sharp positive spike; a smooth positive wave.”
Rationale: AFIB replaces the organized P wave with irregular fibrillatory baseline; the P wave primitive is therefore substituted by a prolonged plateau. No event is detected when this description is applied to a normal sinus segment, confirming that P wave absence is sufficient to suppress detection.

Inference parameters: top- k events = 16, top- k primitives = 64, detection threshold = 0.5.

Post-processing. Detected spans are mapped back from the 512-point normalised domain to the original 1 000-sample window via inverse linear interpolation, then offset by the window start index to recover absolute sample positions at 360 Hz.

Detailed Quantitative evaluation. For the record-201 subset, we apply both descriptions across all sliding windows of the full 30-minute record and classify each beat cycle as AFIB or normal based on which description yields a detection. Record 201 contains 879 annotated AFIB beat cycles and 1,640 normal beat cycles. Table 10 reports the resulting beat-level confusion matrix. GroundingMOMENT achieves an overall **accuracy of 92.1%**, with AFIB precision of 91.0%, recall of 86.0%, and F1 of 88.4%. This representative zero-shot result confirms that primitive-level descriptions can discriminate pathological rhythm changes in clinical recordings.

	Pred. AFIB	Pred. Normal
True AFIB	756 (TP)	123 (FN)
True Normal	75 (FP)	1,565 (TN)

Table 10: Beat-level confusion matrix for AFIB detection on MIT-BIH record 201 (2,519 beat cycles total). Accuracy = 92.1%, Precision = 91.0%, Recall = 86.0%, Specificity = 95.4%, F1 = 88.4%.

G Drilling Case Study Details

Connection event analysis. Drilling connections—the cyclic operations in which the drill string is extended by adding a new pipe stand—are a critical efficiency bottleneck. Each connection follows a mandatory sequence of three operations: (i) *pipe detachment* from the rotary, (ii) *pipe pickup*, and (iii) *pipe reattachment*. In the STOR channel, detachment manifests as a first medium-amplitude spike and reattachment as a second, larger sharp spike; the BPOS channel captures the pickup as a sustained increase in block position. The total duration of each step directly reflects operational efficiency.

We describe the connection event with two complementary queries—one per channel—taken from Table 1 and run inference independently on each channel. As shown in Figure 4, GroundingMOMENT correctly localizes the ordered primitive sequence on both channels without any domain-specific training.

Detailed Quantitative evaluation. The detailed operational analysis below uses a 7-hour subset, where domain experts identified 91 connection events. GroundingMOMENT detects 85 of these, yielding a **recall of 93.4%** without domain-specific training. For each detected connection, the model localizes all three constituent phases, enabling per-phase duration analysis. Table 11 summarizes the phase durations across the 85 detected connections. Detachment and pickup are executed consistently (CV of 37.6% and 48.2%, respectively), while reattachment exhibits a coefficient of variation (CV) of 85.0%—identifying it as the primary target for operational improvement.

Phase	Mean (s)	Std (s)	CV (%)
Detachment	25.0	9.4	37.6
Pickup	5.6	2.7	48.2
Reattachment	41.3	35.1	85.0
Total	71.9	—	—

Table 11: Phase duration statistics (in seconds) across 85 detected drilling connections. CV = std/mean. The high CV for reattachment identifies it as the primary operational bottleneck.

Data. The aggregate dataset contains 52 days of multivariate drilling telemetry and 179 expert-annotated connection events. The phase-analysis

subset is a 7-hour recording sampled at 1 s, containing 25 359 timesteps.

Preprocessing. A sliding window of 650 timesteps (10.8 min) with 50% overlap (325 step stride) is applied to each channel independently. Each window is linearly interpolated to 512 points to match the model’s fixed input length, then z-score normalized.

Description templates. The two natural-language descriptions (Table 1) used for inference are:

- **STOR:** “A medium spike; a large sharp spike; a prolonged plateau.”
- **BPOS:** “A sharp increase; a short plateau; a smooth linear decrease.”

Inference parameters: top- k events = 16, top- k primitives = 64, detection threshold = 0.5.

Post-processing. Detected spans are mapped back from the 512-point normalised domain to the original 650-sample window via inverse linear interpolation, then offset by the window start index to recover absolute timestamps.