# Appendix

## Overview

This supplementary material provides additional details and comprehensive benchmarks omitted from the main text due to space constraints. The contents are organized as follows:

- **Section A:** Details our training data composition, hyperparameters, data formatting templates, exact inference prompt designs, and statistical variance analysis across multiple random seeds.
- **Section B:** Presents a macro-level survey and zero-shot performance leaderboard of existing VTG-MLLMs.
- **Section C:** Provides extended qualitative visualizations to further illustrate the intrinsic behaviors of different output paradigms.

Upon acceptance, we will publicly release the full training and evaluation codebase, including all paradigm implementations, data formatting pipelines, and benchmark configurations.

## A. Additional Implementation Details

**Hardware and Optimization.** Training is conducted on 2 nodes, each equipped with 4 NVIDIA H200 GPUs. We use a per-GPU batch size of 4 and a gradient accumulation step of 4, yielding a global effective batch size of 128. For LoRA fine-tuning, we set the rank $r = 16$ and $\alpha = 32$, targeting the $q, k, v, o$ projections in the LLM attention blocks. The learning rate is initialized at $1 \times 10^{-4}$ with a cosine decay schedule and a 3% linear warmup ratio.

**Training Data Composition.** Table S1 details the source-level composition of our training set. The combined corpus comprises approximately 1.2M temporally annotated samples from ∼400K unique videos across 11 sources, spanning moment retrieval (66.8%), grounded video question answering (21.5%), and dense video captioning (11.6%) tasks.

Table S1. Training data composition by source.

| Source | Samples | Videos |
|---|---|---|
| InternVid [27] | 608K | 87K |
| YTTemporal [36] | 278K | 21K |
| Valley [17] | 229K | 229K |
| DiDeMo [1] | 33K | 8K |
| ShareGPT4Video [3] | 24K | 24K |
| ViTT [10] | 8K | 5K |
| TextVR [31] | 8K | 8K |
| COIN [24] | 8K | 8K |
| ActivityNet [2] | 7K | 7K |
| QueryD [21] | 5K | 1K |
| VideoChat [13] | 2K | 2K |
| **Total** | **∼1.2M** | **∼400K** |

**Data Formatting Setup.** To strictly isolate the output formulation as the sole experimental variable, all paradigms share the exact same raw video-text pairs but undergo paradigm-specific prompt and target formatting. We detail these exact input-output formats for a specific training sample in Table S3.

Table S2. Backbone-level specifications.

| Backbone | Vision Enc. | LLM | $d$ |
|---|---|---|---|
| SmolVLM2-0.5B | SigLIP-B/16 | SmolLM2-360M | 960 |
| SmolVLM2-2.2B | SigLIP2-SO400M | SmolLM2-1.7B | 2048 |
| FastVLM-1.5B | FastViTHD | Qwen2-1.5B | 1536 |
| Molmo2-4B | SigLIP2-SO400M | Qwen3-4B | 2560 |
| Molmo2-8B | SigLIP2-SO400M | Qwen3-8B | 4096 |

**Inference Prompt Templates.** In the era of Multimodal Large Language Models, the formulation of textual prompts profoundly impacts evaluation performance. To ensure rigorous reproducibility of our empirical study, we provide the exact inference prompt templates used across the three video temporal grounding tasks. During evaluation, the placeholder {`query`} is dynamically replaced with the specific textual query from the corresponding dataset. The detailed templates are presented in Table S4.

**Module Architectures and Loss.** Table S2 summarizes the backbone-level specifications. Let $d$ denote the LLM hidden dimension. For the *Continuous* paradigm, the Time Decoder is a 3-layer MLP with ReLU activations ($d \to d \to d \to 2(R{+}1)$, Xavier initialization) that maps the hidden state at the ⟨TIME_STAMP⟩ position to $2{\times}(R{+}1)$ distribution logits over $R{=}32$ bins. A symmetric Time Encoder converts ground-truth timestamps into Gaussian distributions ($\sigma{=}1.0$) over $R{+}1$ anchor points and maps them back to $\mathbb{R}^d$ via an inverse MLP ($2(R{+}1) \to d \to d \to d$). The total loss is $\mathcal{L} = \mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{DFL}} + \mathcal{L}_{\text{DIoU}}$ with equal weights. For the *Generative* paradigm, TimeTower and ScoreTower are each an embedding layer over a 13-token character vocabulary (`Embedding`$(13, d)$), and SyncTower is a single learnable embedding (`Embedding`$(1, d)$); all three heads use standard cross-entropy. The *Text* paradigm adds no modules and uses the standard next-token loss. Additional hyperparameters: weight decay 0.01, LoRA dropout 0.05, seed 42, max sequence length 4096.

**Inference Post-Processing.** For *Continuous*, the $2(R{+}1)$ logits are split into start/end groups, softmax-normalized, and decoded via expectation $\hat{t} = \sum_{i=0}^{R} p_i \cdot i$, then rescaled by duration$/R$ to absolute seconds. For *Generative*, the head-switching state machine accumulates character-level time tokens, splits by ⟨sep⟩, and parses floating-point timestamps; scores are decoded analogously. For *Text*, timestamps are extracted from the generated string via pattern matching (e.g., "from X to Y seconds").

**Shared annotation:** `video: split_video_7w/RP6nWBaB2K8.mp4` ground truth: $[52.0, 63.0]$ s

**(a) Text Numeral Generation**

INPUT: `<video>\n`With the input: 'The worker cleans up excess plaster from the ground, ensuring a tidy workspace.', identify the precise second in the video where this event occurs.

OUTPUT: The event happens from 52.0 to 63.0 seconds.

SUPERVISION: standard next-token cross-entropy (no additional temporal modules)

**(b) Temporal Token Generation**

INPUT: `<video>\n`With the input: 'The worker cleans up excess plaster from the ground, ensuring a tidy workspace.', identify the precise second in the video where this event occurs.

OUTPUT: `<sync>`$\underbrace{\texttt{<time>}\cdots\texttt{<time>}}_{14}$`<score>`The worker cleans up excess plaster from the ground, ensuring a tidy workspace.

SUPERVISION: `times: [[52.0, 63.0]], scores: [[]]` $\rightarrow$ dedicated 13-token vocabulary

**(c) Continuous Temporal Decoding**

INPUT: `<video>\n`With the input: 'The worker cleans up excess plaster from the ground, ensuring a tidy workspace.', identify the precise second in the video where this event occurs.

OUTPUT: The event happens in `<TIME_STAMP>`.

SUPERVISION: `times: [[52.0, 63.0]]` $\rightarrow$ distribution-based time decoder

Table S3. Training data format comparison across the three paradigms for the same moment retrieval annotation. All paradigms share identical video inputs and queries; only the output format and temporal supervision differ.

Table S4. Exact inference prompt templates utilized for evaluating the models across three temporal grounding tasks. All paradigms receive the identical text prompt alongside the visual input to ensure fair comparison.

| **Task (Dataset)** | **Inference Prompt Template** |
|---|---|
| **Moment Retrieval** (Charades-STA) | `Localize the visual content described by the given textual query '{query}' in the video, and output the start and end timestamps in seconds.` |
| **Highlight Detection** (QVHighlights) | `Please find the highlight contents in the video described by a sentence query, determining the highlight timestamps and its saliency score on a scale from 1 to 5. Now I will give you the sentence query: '{query}'. Please return the query-based highlight timestamps and salient scores.` |
| **Dense Video Captioning** (YouCook2) | `Scrutinize the video and determine multiple occurrences, providing their initial and final timestamps as well as a summary of each action.` |

**Statistical Variance Analysis.** To verify the robustness and reproducibility of our findings, we repeat training with 3 different random seeds on two representative backbones: SmolVLM2-2.2B (a compact model) and Molmo2-4B (a mid-scale model), covering the two primary scales studied in this work. Table S5 reports the mean and standard deviation of mIoU on Charades-STA and QVHighlights for all three paradigms.

As shown in Table S5, the standard deviations across all configurations are consistently small (typically $\leq$0.5 mIoU), demonstrating that our conclusions are robust to random initialization. Notably, the performance gap between paradigms (e.g., *Cont.* vs. *Text*: $\sim$26 mIoU on SmolVLM2-2.2B) is an order of magnitude larger than the within-paradigm variance, confirming that the observed differences are attributable to the output formulation rather than stochastic training variation.

## B. Comprehensive Survey of VTG-MLLMs

To situate our empirical findings within the broader landscape of Video-LLM research, we compile a comprehensive performance leaderboard of existing state-of-the-art multimodal large language models capable of video temporal grounding. As presented in Table S6, we detail their

Table S5. Variance analysis across 3 random seeds on two representative backbones. We report mean±std of mIoU. The consistently small standard deviations confirm that the observed paradigm gaps are robust and not attributable to stochastic training variation.

| Backbone | Para. | Charades-STA mIoU | QVHighlights mIoU |
|---|---|---|---|
| SmolVLM2-2.2B | Text | 20.6±0.4 | 16.3±0.3 |
| | Cont. | 46.8±0.3 | 54.2±0.4 |
| | Gen. | 27.9±0.5 | 23.7±0.4 |
| Molmo2-4B | Text | 24.1±0.3 | 18.8±0.4 |
| | Cont. | 56.1±0.2 | 59.8±0.3 |
| | Gen. | 32.5±0.4 | 27.3±0.3 |

zero-shot performance across standard benchmarks, strictly grouped by their underlying temporal output paradigms.

This macro-level overview highlights the extreme heterogeneity in current VTG-MLLM evaluation setups, strongly validating the necessity of our controlled, variable-isolating study presented in the main text.

## C. Extended Failure Taxonomy and Qualitative Examples

To complement the failure case analysis in Section 5.4 of the main paper, we first provide a quantitative taxonomy of error types to systematically dissect paradigm weaknesses, followed by additional visual examples that concretely illustrate these behaviors.

### C.1. Quantitative Taxonomy of Errors

We conduct a systematic diagnosis on the Charades-STA dataset to analyze the failure distribution across the three representative paradigms: DisTime (*Continuous*), TRACE (*Temporal Token*), and VtimeLLM (*Text Numeral*). For each paradigm, we collect the failed predictions (i.e., IoU $< 0.5$) and categorize them into three distinct error types:

- **Type A (Temporal Hallucination):** The model predicts a temporal window completely disjoint from the ground truth, often occurring when LLMs fail to ground abstract numerals to continuous video frames.
- **Type B (Boundary Jitter):** The model correctly identifies the macroscopic event, but the temporal boundaries are overly bloated or truncated, failing the strict 0.5 IoU threshold.
- **Type C (Semantic Failure):** The model fundamentally misunderstands the query (e.g., confusing "opening" with "closing") or the causal logic of the video.

As visually detailed in Figure S1, the text numeral *VtimeLLM* exhibits a staggering 61.4% of its errors stemming from Temporal Hallucination (Type A). The token-based *TRACE* also suffers heavily from hallucinations

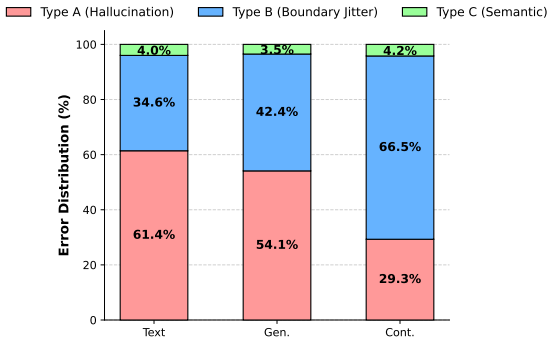


Figure S1. Quantitative error taxonomy across representative paradigms on Charades-STA. The composition explicitly breaks down the failure cases (where IoU $< 0.5$) into three types. The *Continuous* paradigm drastically shifts the error mode from severe hallucinations to minor boundary jitters.

(54.1%). Conversely, the continuous *DisTime* model fundamentally shifts the error distribution: its hallucinations are drastically suppressed to 29.3%, with the majority of its errors (66.5%) being minor Boundary Jitters (Type B). This proves that regressing a continuous distribution prevents the LLM from arbitrarily "guessing" wild timestamps.

### C.2. Qualitative Visualizations

Guided by the quantitative taxonomy above, we significantly expand our qualitative analysis by dedicating three comprehensive figures to concretely illustrate these distinct behavioral patterns. We systematically dissect the performance of three representative methods, across our proposed error types: Temporal Hallucination (Figure S2), Boundary Jitter (Figure S3), and Semantic Confusion (Figure S4).

These exhaustive visual examples highlight a critical divergence in underlying failure modes. The ***Continuous*** paradigm gracefully manages fuzzy action boundaries and naturally suppresses severe temporal hallucinations. As shown in the visualizations, even in challenging scenarios where DisTime fails the strict evaluation threshold (e.g., Figure S3 Bottom), its errors are predominantly benign *Type B (Boundary Jitters)*, typically manifesting as an overly bloated temporal distribution that still directionally covers the actual event.

In stark contrast, constrained by the inherent mismatch between discrete token generation and the continuous temporal space, the discrete generative paradigms frequently struggle with strict timestamp regression. Both text-based and token-based approaches are highly susceptible to outputting confidently incorrect, completely disjoint timeframes (*Type A*) or suffering from fundamental semantic misalignments (*Type C*) when reasoning over complex videos.

Table S6. Comprehensive performance leaderboard of existing VTG-MLLMs. Metrics represent zero-shot performance reported in their respective papers or official repositories. ”-” indicates the absolute metric was not reported or evaluated by the original authors. Methods are grouped by their temporal output paradigm to facilitate direct comparison.

| **Method** | **Backbone** | **Charades-STA** *(Moment Retrieval)* | | | **QVHighlights** *(Highlight Detection)* | | **YouCook2** *(Dense Video Captioning)* | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **R1@0.5** | **R1@0.7** | **mIoU** | **mAP** | **HIT@1** | **CIDEr** | **SODA$_c$** | **F1** |
| ***Text Numeral Generation*** | | | | | | | | | |
| TimeChat [23] | LLaMA-2-7B | 32.2 | 13.4 | – | 14.5 | 23.9 | 1.2 | 3.4 | 12.6 |
| VTimeLLM [8] | Vicuna-7B | 27.5 | 11.4 | 31.2 | – | – | 3.4 | 0.9 | – |
| GroundingGPT [14] | Vicuna-7B | 29.6 | 11.9 | – | – | – | – | – | – |
| HawkEye [28] | Vicuna-7B | 31.4 | 14.5 | 33.7 | – | – | – | – | – |
| Chrono-GPT [18] | GPT-4o | 28.8 | 11.0 | 33.0 | – | – | – | – | – |
| ***Temporal Token Generation*** | | | | | | | | | |
| SeViLA [35] | Flan-T5-XL (3B) | 15.0 | 5.8 | 18.3 | – | – | – | – | – |
| Momentor [22] | LLaMA-7B | 26.6 | 11.6 | 28.5 | 7.6 | 17.0 | – | – | – |
| VTG-LLM [6] | LLaMA-2-7B | 33.8 | 15.7 | – | 16.5 | 33.5 | 5.0 | 1.5 | 17.5 |
| Grounded-VideoLLM [25] | Phi-3.5 (4B) | 36.4 | 19.7 | 36.8 | – | – | – | – | – |
| TRACE [5] | Mistral-7B | 40.3 | 19.4 | 38.7 | 42.7 | 26.8 | 8.1 | 2.2 | 22.4 |
| ***Continuous Temporal Decoding*** | | | | | | | | | |
| TimeRefine [26] | Vicuna-7B | 38.6 | 16.4 | 36.2 | – | – | – | – | – |
| InternVideo2.5 [29] | InternLM2.5-7B | 43.3 | 22.8 | 41.7 | 26.5 | 54.1 | – | – | – |
| VideoMind [16] | Qwen2-VL-7B | 59.1 | 31.2 | 50.2 | – | – | – | – | – |
| DisTime [37] | InternVL2.5-8B | 60.3 | 30.8 | 53.1 | – | – | 31.0 | 6.9 | 26.4 |

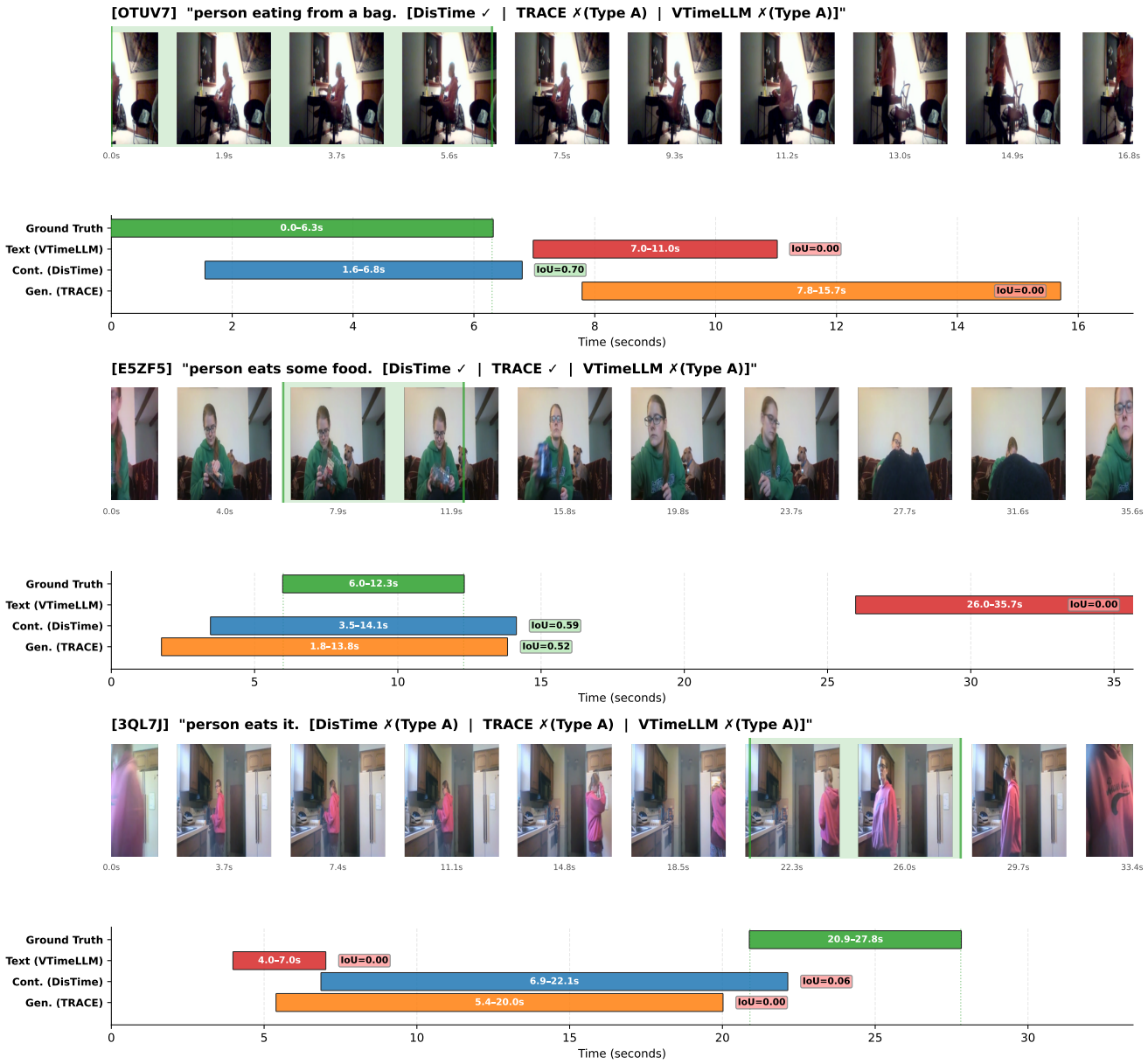


Figure S2. **Qualitative comparisons focusing on Type A (Temporal Hallucination) errors.** Green bars denote Ground Truth. **(Top)** DisTime successfully localizes the event, whereas both discrete paradigms (TRACE and VTimeLLM) completely hallucinate early timeframes, demonstrating the continuous paradigm's robustness against blind guessing. **(Middle)** DisTime and TRACE accurately capture the action, leaving VTimeLLM as the sole model exhibiting a severe Type A hallucination (predicting the very end of the video). This underscores the text-numeral paradigm's extreme susceptibility to temporal hallucinations. **(Bottom)** A universally challenging query where all paradigms fail via hallucination, predicting disjoint early segments. This highlights that while continuous decoding mitigates hallucinations, extreme causal complexity can still trigger Type A errors across all architectures.

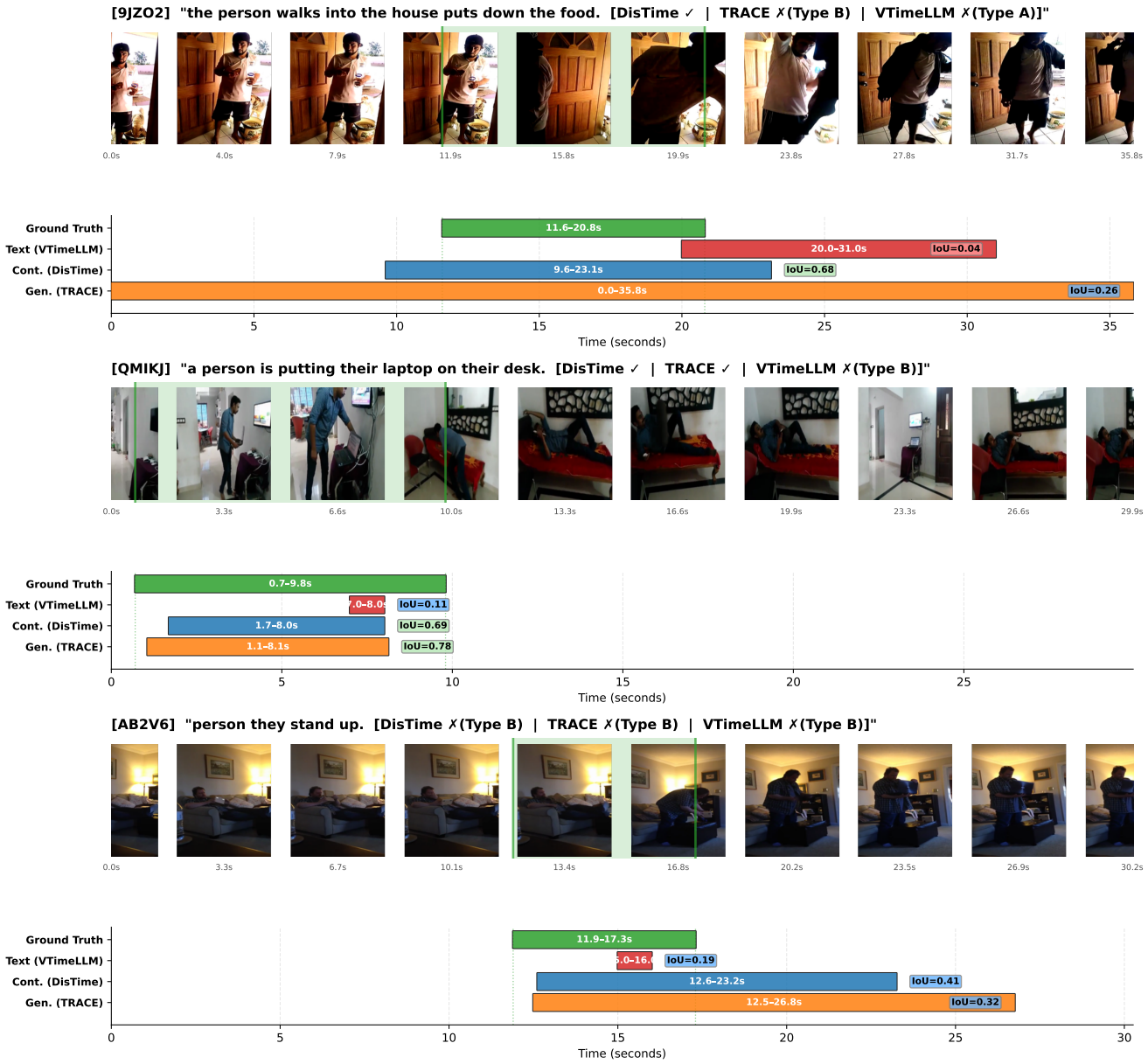


Figure S3. **Qualitative comparisons focusing on Type B (Boundary Jitter) errors. (Top)** DisTime accurately covers the true temporal window. TRACE, however, exhibits massive "Token Bloat" (a severe Type B error spanning almost the entire video), while VTimeLLM hallucinates entirely (Type A). **(Middle)** Both DisTime and TRACE successfully localize the action. VTimeLLM identifies the general vicinity but severely truncates the boundary (7.0-8.0s vs. GT 0.7-9.8s), yielding a failing IoU of 0.11. **(Bottom)** A scenario characterized by ambiguous action boundaries where all models suffer from Type B errors. Notably, DisTime and TRACE regress overly bloated durations (over-extension), whereas the text-based VTimeLLM predicts an extremely narrow window (truncation), reflecting their distinct mechanisms for handling temporal uncertainty.

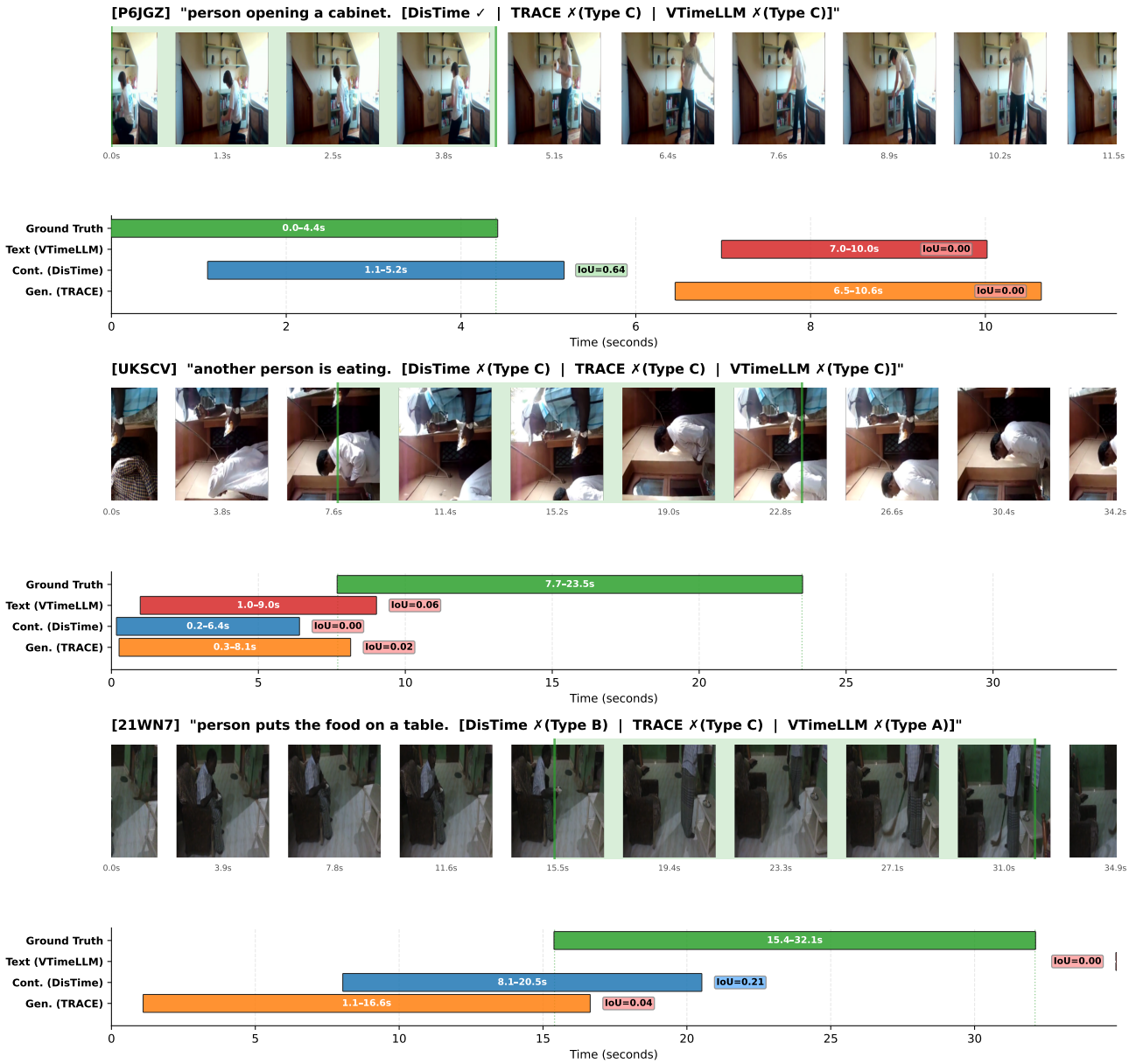


Figure S4. **Qualitative comparisons focusing on Type C (Semantic Confusion) errors. (Top)** DisTime perfectly localizes "opening a cabinet". In contrast, both TRACE and VTimeLLM confuse this action with a semantically similar but chronologically later event, resulting in disjoint predictions (Type C). **(Middle)** A highly deceptive case where all three paradigms are misled by earlier visual cues of a person eating, completely missing the ground-truth timeframe. This represents a universal semantic failure across current VTG-MLLMs. **(Bottom)** A complex failure case demonstrating diverse error manifestations. DisTime successfully captures the event but fails due to benign boundary jitter (*Type B*, IoU=0.21). Meanwhile, TRACE predicts a semantically incorrect early action (*Type C*), and VTimeLLM hallucinates entirely off-target (*Type A*). This perfectly visually encapsulates our taxonomy findings.