arXiv:2610.17856v1 [cs.LG] 13 Oct 2026

MMTA: Benchmarking Multimodal Temporal Analysis with Time Series, Text, and Vision

Ziyang Zhang[1,2]* **Shenyi Li**[2]* **Yilin Wang**[2] **Ziyun Cui**[1]
Bowen Zhou[2] **Wen Wu**[2] **Chao Zhang**[1,2]†

[1]Tsinghua University [2]Shanghai Artificial Intelligence Laboratory

ziyang-z24@mails.tsinghua.edu.cn, cz277@tsinghua.edu.cn
{lishenyi, wuwen}@pjlab.org.cn

Abstract

Time-series analysis is increasingly studied in multimodal settings, yet most benchmarks pair temporal signals with text or with visualizations derived from the same signal. This leaves open whether multimodal large language models (MLLMs) can integrate time series with real-world visual evidence. We introduce **MMTA** (**m**ulti**m**odal **t**emporal **a**nalysis), a benchmark of tri-modal tasks with diverse cross-modal complementarity patterns among time series, text, and vision. MMTA spans nine domains – robotics, autonomous driving, human action, manufacturing, AR assistance, healthcare, recommendation, finance, and climate – under *video+TS* and *image+TS* layouts, with a 3,524-sample test split covering classification and forecasting. The benchmark standardizes heterogeneous real-world sources into a shared schema with domain-specific prompts, provenance documentation, and success-weighted metrics that penalize both wrong answers and failed generations. Evaluating eight open-source and proprietary MLLMs reveals a shared bottleneck: representing time series as text creates very long token sequences and weakens fine-grained alignment with visual evidence, especially in long-video settings. We then introduce **TimeOmni-v**, a multimodal LLM extension with a dedicated time-series encoder, temporal positional alignment, modality-aware token layout, and a forecasting head. TimeOmni-v improves long-video classification and time-series inference efficiency, showing that MMTA not only exposes representation and grounding failures but also motivates targeted model design.[3]

1 Introduction

Time-series analysis, in both forecasting and classification, has long underpinned decision making in domains such as economics, meteorology, healthcare, and industrial monitoring. Recent work has therefore moved beyond unimodal temporal signals toward formulations that pair time series with natural language [1–7]. This direction is motivated by a simple limitation: numerically similar temporal patterns can mean different things depending on the task, domain, and event context. Text helps by exposing instructions, reports, metadata, or external events that are not recoverable from the sequence alone.

Text, however, is only one part of the multimodal temporal setting. Many real-world decisions also depend on visual evidence that is not a rendering of the time series. Vision encodes physical and semantic context such as objects, scenes, actions, medical imagery, product appearance, or

*Equal contribution.

†Corresponding author.

[3]Demo: `https://windowso.github.io/MMTA/`. Data: `https://huggingface.co/datasets/OpenTSLab/MMTA-v1.0`. Code for TimeOmni-v and baseline evaluation: `https://github.com/windowso/TimeOmni-v`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Track on Evaluations and Datasets.

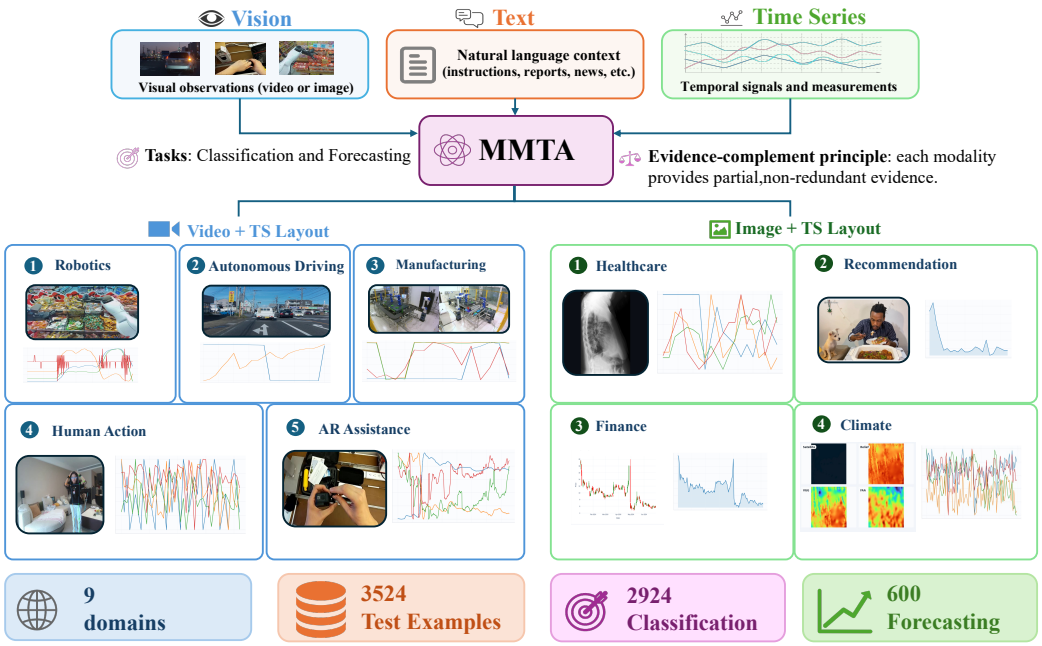


Figure 1: Overview of MMTA. The benchmark unifies nine real-world domains – Robotics, Autonomous Driving, Human Action, Manufacturing, AR Assistance, Healthcare, Recommendation, Finance, and Climate – under two cross-modal layouts (video + time series and image + time series). Vision provides real-world context, text provides semantic context, and time series provides quantitative dynamics; which modality dominates varies across domains.

geophysical structure; time series provides high-resolution quantitative dynamics; and text supplies semantic context, events, reports, or instructions. These sources are structurally different and often only jointly informative. For example, as illustrated in Figure 1 for the robotics domain (case taken from the AgiBotWorld dataset [8]), deciding whether a robot completed an episode correctly draws on the language instruction to establish the goal, the egocentric video to follow the operation, and the joint and end-effector traces to verify fine-grained execution. Existing benchmarks do not systematically test whether current MLLMs can perform this kind of evidence integration over visual, textual, and temporal streams.

We address this gap with **MMTA** (**M**ulti**m**odal **T**emporal **A**nalysis), a real-world benchmark that places vision, text, and time series side by side. MMTA spans nine domains: robotics, autonomous driving, human action, manufacturing, AR assistance, healthcare, recommendation, finance, and climate. Each domain is exported under one of two cross-modal layouts, video plus time series, or image plus time series. The test split contains 3,524 samples and covers both classification and forecasting. MMTA is a collection of tri-modal tasks with diverse cross-modal complementarity patterns: depending on the domain, a modality may be decisive, supporting, or nearly redundant. MMTA is intended as an evaluation substrate for studying temporal grounding, modality reliance, efficient time-series representation, and robust reasoning under missing or weakly aligned evidence, rather than only as a leaderboard.

Our systematic evaluation of eight representative open-source and proprietary MLLMs on MMTA exposes a common failure mode: serializing time series as text produces excessively long token sequences and offers no mechanism to align temporal samples with co-occurring visual evidence. We address this bottleneck with **TimeOmni-v**, a multimodal LLM extension designed for long time series in multimodal temporal tasks. TimeOmni-v attaches a dedicated time-series encoder to an existing multimodal LLM, gives time-series tokens a temporal positional structure, and arranges them relative to the visual stream. This design improves long-video classification and inference efficiency, while the remaining difficulty of multimodal forecasting highlights open problems surfaced by MMTA.

Our contributions are summarized as follows.

- We introduce **MMTA**, a real-world collection of tri-modal tasks with diverse cross-modal complementarity patterns among vision, text, and time series, across nine domains and two layouts, video+TS and image+TS.
- We standardize heterogeneous sources into a shared evaluation suite with a 3,524-sample test split, classification and forecasting tasks, domain-specific prompts, provenance and license documentation, human and mechanical quality audits, and success-weighted metrics that penalize both incorrect answers and unparsable outputs.
- We characterize the difficulty of MMTA through eight representative MLLMs, identifying a common failure mode in which serializing time series as text creates excessive token length and weak visual-temporal alignment.
- We introduce **TimeOmni-v**, a multimodal LLM extension with a dedicated time-series encoder, temporal positional alignment, modality-aware token layout, and forecasting head. TimeOmni-v improves long-video classification and inference efficiency, can be fine-tuned on long-TS domains where the same backbone with textualized time series runs out of memory, and also runs on InternVL3.5-8B on the four datasets tested.

2 Related Work

Multimodal time-series datasets and benchmarks. Multimodal time-series datasets have grown along three lines. The first line pairs time series with free-form text such as news, reports, or task descriptions, as in MTBench [2], Context is Key [3], TimeCAP [9], Time-MMD [4], NewsForecast [10], Time-MQA [5], and TimesX [11]; Time-IMM [12] pairs irregular multivariate time series with text and adds images only as an extension in its appendix. The second line attaches rendered plots of the underlying signal [13–16], so the image is a view of the series itself rather than an observation of the scene. The most recent line introduces real images alongside text and time series, exemplified by MoTime [6]; even there, images are typically reduced to captions before consumption, only forecasting is supported, and domain coverage is concentrated on recommendation. MMTA differs from all three by using raw scene video or images as sample-level evidence, by aligning them with text and time series across nine real-world domains, and by covering both classification and forecasting.

Time-series-aware language models. Recent efforts to make language models natively consume time series stop short of fusing with vision, leaving an architectural slot that TimeOmni-v occupies. One line attaches a dedicated time-series encoder to a text LLM through token-level patch concatenation [17–20]; the resulting models reason over time series and text only, with no co-occurring visual stream. A separate line builds omni-modal LLMs with native text, image, audio, and video encoders [21, 22], yet none of them ships a time-series encoder, so long signals must still be serialized as digit strings inside the textual prompt. TimeOmni-v integrates time series into an omni-modal LLM as a first-class modality, on the same footing as vision and audio.

3 The MMTA Benchmark

MMTA is a collection of vision+TS+text tri-modal tasks with diverse cross-modal complementarity patterns, built from the openly available datasets in which the three modalities are natively present. Nine real-world domains are aligned into a single corpus under two cross-modal layouts (Figure 1), *video+TS* and *image+TS*, and each domain is evaluated separately on a 3,524-sample test split (2,924 classification and 600 forecasting samples). The domains differ by an order of magnitude or more in time-series length (up to 5×10^3 steps), channel count (1 to 36), video duration (up to $155.7\,\mathrm{s}$), and image count (1 to 7 per sample). Text is counted as a sample-level modality when it varies across samples and carries scene, event, or background information; seven domains carry such text, whereas the prompts of CoVLA and CUHK-X contain only the task description and the label space (Appendix B.1). Provenance, licenses, and prompt templates are documented in Appendix A, and the release includes a Croissant metadata file with the Responsible AI fields.

3.1 Video+TS Domains

Robotics (AgiBotWorld). The source is AgiBotWorld2026 [8]. Each instance is one Task Frame within a manipulation episode. The visual stream is the top-head egocentric clip covering the full task-frame interval, and the temporal stream is a 10-channel state trace combining six arm-position channels with four end-effector velocity channels. The text modality exposes the task instruction and the ordered step sequence. The model is asked to classify the dominant anomaly cause among four labels: *paused*, *meaningless operation*, *fall*, or *other*.

Autonomous Driving (CoVLA). The source is the CoVLA Dataset [23], which contains driving videos recorded on Japanese public roads with synchronized vehicle-state and lead-vehicle annotations. Each instance is a braking segment, defined as a continuous interval where the brake pedal stays pressed for at least six frames. The visual stream is the cropped dashcam clip over that interval, and the temporal stream is five synchronized channels: ego speed, ego acceleration, steering angle, lead-vehicle speed, and lead-vehicle acceleration. The model classifies the brake reason into three categories: lead-vehicle deceleration, ego turning, or other. The labels are derived by rules from upstream signals. The brake-pedal and turn-signal variables that locate the braking interval and define the ego-turning class are not among the five released channels, whereas lead-vehicle acceleration also enters the lead-vehicle-deceleration rule, so the label and the input time series overlap partially.

Human Action (CUHK-X). The source is the CUHK-X HAR subset [24]. Each instance is one action trial. The visual stream is a synthetic skeleton-rendered clip, so pose evolution is encoded directly in the video rather than through raw RGB; the temporal stream is five wearable IMU streams, each containing three accelerometer channels and three gyroscope channels. The model classifies the action over 40 categories.

Manufacturing (Future Factories). The source is the Future Factories Platform [25]. Each instance is one production cycle on an industry-grade assembly line. The visual stream is the on-site surveillance video captured by two synchronized cameras monitoring the assembly station; the temporal stream is 36 PLC channels (12 gripper signals plus 24 robot joint angles); and the text modality is a cycle report describing the process. Defects are introduced by manually removing components from a modular rocket assembly. The model classifies the cycle into one of four labels: *Correct*, *NoNose*, *NoNose,NoBody2*, or *NoNose,NoBody2,NoBody1*.

AR Assistance (HoloAssist). The source is HoloAssist [26]. Each instance is one coarse-grained action segment with a duration between 20 and 80 seconds. The visual stream is the egocentric RGB clip cropped to the segment, the temporal stream is nine synchronized HoloLens channels (three-axis gaze direction, three-axis head translation, three-axis gyroscope), and the text modality lists the ordered fine-grained actions inside the segment. The model is asked a binary correctness question: whether all contained fine-grained actions were executed correctly.

3.2 Image+TS Domains

Healthcare (MIMIC). The sources are MIMIC-IV [27], MIMIC-CXR [28], and MIMIC-IV-Note [29]. Each instance is a clinically anchored cohort point: either a chest X-ray acquisition or a fixed 48-hour mark after admission. The visual stream is the chest radiograph at the anchor; the temporal stream is six routinely recorded vital signs (heart rate, respiratory rate, SpO_2, temperature, NIBP systolic, NIBP diastolic) over the observation window; the text modality is the radiology report aligned to the same study, which records the radiologist's interpretation, clinical history, and prior-exam comparisons. Two prognosis-style tasks are evaluated: *in-hospital mortality prediction* (predict whether the patient ultimately dies during the hospital stay using the first 48 hours of data) and *discharge readiness prediction* (predict whether the patient should be discharged within the next 48 hours from a 48-hour lookback window).

Recommendation (PixelRec). The source is the PixelRec dataset [30], which captures video-consumption behaviors in lifestyle and entertainment categories. Each instance is one item. The visual stream is the item cover thumbnail; the temporal stream is the daily interaction count series (*e.g.*, daily views or clicks); the text modality concatenates title, tag, and description. The model is asked

to forecast the next-multi-day interaction volume, conditioning on the historical daily interactions, the cover image, and the metadata text.

Finance (FinMultiTime). The source is the FinMultiTime dataset [31], instantiated on the S&P 500 universe. Each instance is one asset paired with a half-year history window. The visual stream is the candlestick chart of the same window, which encodes the open, high, low, and close prices; the temporal stream is the daily closing-price series, so the two share only the close price; the text modality is a small set of recent news snippets selected strictly from before the prediction horizon. The model forecasts closing prices over the next 12 trading days, combining numeric trajectories with semantic drivers (events, sentiment) and visual technical patterns.

Climate (Terra). The source is the Terra dataset [32], defined on a global 1° latitude–longitude grid. Each instance is one grid cell. The visual stream is a multi-image stack of remote-sensing and geophysical raster products (4 to 7 images per sample, including satellite imagery, geoid, free-air anomaly, magnetic anomaly, land–sea mask, relief, and vertical gravity gradient); the temporal stream is seven climate variables over the historical window (precipitation, temperature, surface pressure, relative humidity, specific humidity, wind speed, downward shortwave or longwave radiation); the text modality is a geographic description of the cell synthesized by an LLM from the dataset's per-cell metadata (coordinates, city, vegetation, climate, elevation). The model forecasts the next 20 daily values for all variables.

3.3 Data Quality

Two components of MMTA are constructed by our pipeline rather than inherited from upstream annotations: the rule-derived CoVLA brake-reason labels and the LLM-generated Terra descriptions. AgiBotWorld, Future Factories, and HoloAssist keep the upstream label semantics through deterministic aggregation or label-space normalization. For each of the two constructed components, 100 samples were checked independently by two annotators, and disagreements were resolved by adjudication. The CoVLA labels reach 96% accuracy (Cohen's $\kappa = 0.910$) and the Terra descriptions 93% ($\kappa = 0.687$). All 25,057 video+TS samples passed a mechanical audit of sample identity, time boundaries, and time-series coverage: 4 HoloAssist samples (0.016%) do not fully cover the target window, and 1.42% of the released frames lie outside it because clips are cut at keyframes. Image+TS domains are joined on deterministic keys. The audit protocol, error types, and per-domain results are given in Appendix B.

4 TimeOmni-v for Multimodal Temporal Analysis

To improve the efficiency of a multimodal LLM on long time series and to tighten its fusion with vision and text, we develop **TimeOmni-v** as an illustrative framework built on Qwen2.5-Omni [21]. As illustrated in Figure 2, TimeOmni-v comprises four components. A Chronos-2 [33] encoder branch compresses the raw time-series stream into a compact token sequence. A TS-specific multimodal RoPE gives time series a 2D positional code aligned with the video timeline. A modality-aware token layout interleaves time-series tokens with video frames or blocks them next to still images. A lightweight forecast head emits multi-step numerical predictions in a single forward pass.

4.1 Time-series modality integration

The time-series modality enters the LLM through a Chronos-2 [33] encoder branch parallel to the vision encoder. Chronos-2 is a pretrained time-series foundation model, trained on large-scale corpora via a forecasting objective. Given a multivariate input $\mathbf{X} \in \mathbb{R}^{T \times C}$ with T time steps over C channels, the encoder applies channel-independent patch tokenization along the time axis with patch length $P = 16$. It produces a patch embedding sequence $\mathbf{H} \in \mathbb{R}^{N \times D_{\text{enc}}}$ with $N = C \cdot \lceil T/P \rceil$ output tokens, where D_{enc} is the Chronos-2 hidden dimension. A lightweight adapter then maps each patch embedding from D_{enc} to the LLM hidden dimension D_{llm}, producing the TS token sequence $\mathbf{X}_{\text{ts}} \in \mathbb{R}^{N \times D_{\text{llm}}}$. $\mathbf{X}_{\text{ts}}$ is finally scattered into the LLM input embeddings.

To align time series with the visual stream both positionally and sequentially, TS tokens carry a TS-specific multimodal RoPE and follow a modality-aware layout. The RoPE uses layout (t, c, c), with the temporal axis on the first channel and the channel index duplicated into the remaining two,

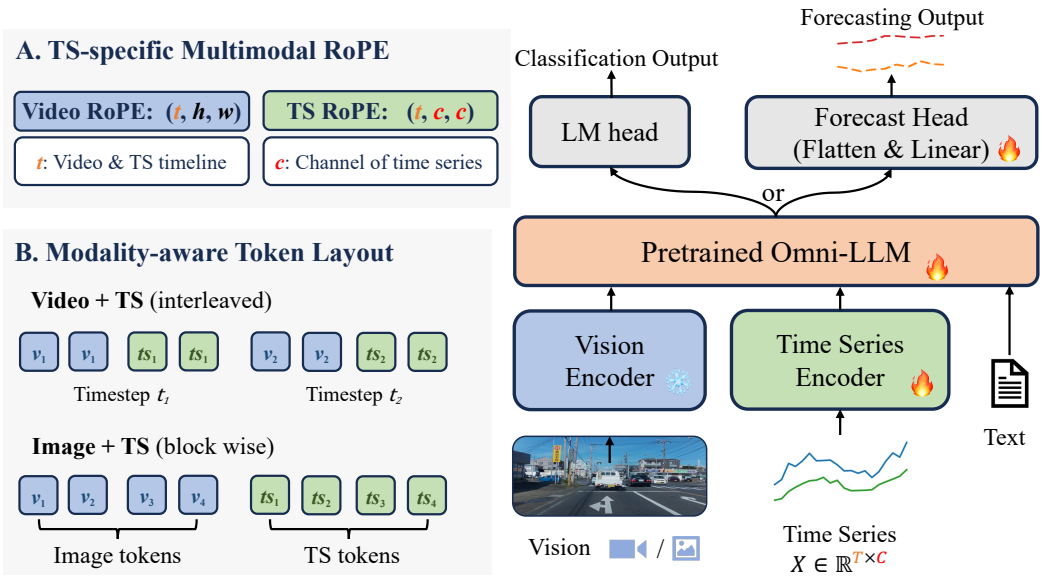


Figure 2: Overview of TimeOmni-v. Time series enters the multimodal LLM through a dedicated Chronos-2 encoder branch parallel to the vision encoder. TS tokens carry a TS-specific multimodal RoPE and are arranged relative to the visual stream via a modality-aware layout (interleaved with video, blocked against still images). A forecast head is attached for numerical prediction; the standard LM head is used for classification.

replacing the (t, h, w) axes used for video. When TS coexists with video, t is rescaled onto the video timeline. A TS patch and the co-occurring video frame then share the same RoPE temporal value, which yields a zero relative-time offset under attention. The same TS tokens are time-interleaved with video frames when both are present, or grouped as a contiguous block next to still images. In the interleaved case, the video stream is split into per-frame token spans, and after each frame the C TS tokens of the patch covering that frame's time interval are spliced in, so each frame is immediately followed by the cross-channel readings of the same instant.

4.2 Forecast head for numerical prediction

To produce multi-step numerical predictions in a single forward pass, TimeOmni-v attaches a lightweight forecast head on top of the LLM in place of autoregressive digit decoding. The head takes the last-layer hidden states $\mathbf{Z} \in \mathbb{R}^{W \times D_{\text{llm}}}$ over a fixed window of W tokens anchored at the first TS token of the sample. A linear projection then maps $\mathbf{Z}$ to $\hat{\mathbf{Y}} \in \mathbb{R}^{L \times C}$ in normalized space, where L is the prediction length and C is the channel count. Per-sample input statistics are used to denormalize $\hat{\mathbf{Y}}$ to the original scale. The forecast head is attached only on forecasting tasks; classification reuses the standard LM head, so the same model serves both formats.

5 Experimental Setup

5.1 Evaluated Models

Both open-source and proprietary MLLMs are evaluated. The open-source set covers two families. The vision-language family contains InternVL3.5-4B [34], InternVL3.5-8B [34], Qwen3-VL-8B-Instruct [35], and Qwen3-VL-30B-A3B-Instruct [35]. The omni-modal family contains Qwen2.5-Omni-7B [21] and Qwen3-Omni-30B-A3B-Instruct [22]. The proprietary set contains Gemini-3-

Table 1: Main results on MMTA. swF1 (↑) for classification, swMAPE (↓) for forecasting. Best per column in **bold**, second best underlined. "—" marks the cell where the model produced no parsable answer on any sample (Qwen3-VL-30B-A3B-Inst. runs out of memory on every AgiBot sample); per-row averages aggregate only over defined cells.

	Classification (swF1 ↑)						Forecasting (swMAPE ↓)			Overall	
Model	AgiBot	CoVLA	CUHK-X	FuFa	HoloA	MIMIC	PixelRec	SP500	Terra	Avg-F1 ↑	Avg-MAPE ↓
Open-source vision-language MLLMs											
InternVL3.5-4B	0.034	0.495	0.218	0.049	0.340	0.528	150.0	4.77	318.4	0.277	157.7
InternVL3.5-8B	0.057	0.514	0.253	0.050	0.299	0.565	98.0	5.44	156.1	0.290	86.5
Qwen3-VL-8B-Instruct	0.108	0.592	0.084	0.049	0.269	**0.655**	145.5	5.49	77.4	0.293	76.1
Qwen3-VL-30B-A3B-Inst.	—	0.577	0.084	0.129	0.000	0.585	129.2	4.75	75.6	0.275	69.8
Open-source omni-modal MLLMs											
Qwen2.5-Omni-7B	0.074	0.424	0.099	0.198	0.257	0.555	104.4	7.73	177.8	0.268	96.6
Qwen3-Omni-30B-A3B-Inst.	0.052	0.491	0.244	0.129	0.234	0.642	98.0	6.72	**65.4**	0.299	**56.7**
Proprietary models											
GPT-5-mini	0.116	0.586	0.251	0.093	0.569	0.602	**91.3**	4.94	104.0	0.370	66.7
Gemini-3-flash	0.126	0.560	**0.463**	0.315	0.561	0.628	91.7	**4.45**	96.1	0.442	64.1
TimeOmni-v (Ours)	**0.844**	**0.759**	0.343	**0.362**	**0.694**	0.602	108.4	4.45	159.6	**0.601**	90.8

Flash and GPT-5-Mini. Official checkpoints or APIs are used with default inference settings unless stated otherwise.[4]

In addition to the eight reference baselines, the proposed **TimeOmni-v** (Section 4) is evaluated under the same protocol. Its multimodal LLM half is initialized from Qwen2.5-Omni-7B [21] and its time-series encoder branch from Chronos-2 [33]; the multimodal LLM is fine-tuned with LoRA [36] while the time-series branch, the modality adapter, and the forecast head are fine-tuned. The remaining training details and per-dataset hyperparameters are deferred to Appendix C.

Three controlled baselines separate the contribution of the time-series integration from that of fine-tuning and of the LLM. The *adapted TS-as-text baseline* fine-tunes the same Qwen2.5-Omni-7B backbone with the identical LoRA configuration and per-dataset budget as TimeOmni-v, with the time series passed as lossless inline text. The *non-LLM baselines* reuse the feature extractors of TimeOmni-v: Chronos-2 for time series, the frozen Qwen2.5-Omni vision encoder under the same preprocessing for vision, and a BERT encoder [37] for text. On these features, single-modality linear probes and a tri-modal late-fusion classifier over the concatenated features are trained for classification, and an MLP regression head over the same concatenated features is trained for forecasting. Finally, the TimeOmni-v time-series integration (Chronos-2 encoder branch, modality-aware token layout, and scatter injection) is ported to InternVL3.5-8B, a backbone with a different vision stack, and evaluated on CoVLA, AgiBotWorld, MIMIC, and FinMultiTime.

5.2 Tasks and Metrics

MMTA contains two evaluation formats, each paired with one headline metric.

Generation models may occasionally fail to return a valid, parsable answer (*e.g.*, when the serialized time series exceeds the model's context window or the model deviates from the requested output format).[5] To penalize such failures alongside ordinary errors, the *success rate* – the fraction of samples that produce a parsable answer – is folded into both headline metrics.

Classification: swF1. A classification task asks the model to select one option from a candidate set. The base metric is Macro-F1, computed by averaging per-class F1 scores (one-vs-rest) to balance performance across classes. Unparsable outputs are treated as misclassifications, so the headline metric is success-weighted Macro-F1, swF1 = Macro-F1 × SuccessRate, which penalizes both class confusion and frequent parsing failures.

[4] Hugging Face checkpoints: `OpenGVLab/InternVL3_5-4B`, `OpenGVLab/InternVL3_5-8B`, `Qwen/Qwen3-VL-8B-Instruct`, `Qwen/Qwen3-VL-30B-A3B-Instruct`, `Qwen/Qwen2.5-Omni-7B`, and `Qwen/Qwen3-Omni-30B-A3B-Instruct`; APIs: `gemini-3-flash-preview` and `gpt-5-mini`.

[5] All open-source baselines are evaluated with the standard `transformers` inference pipeline at the model's default context length and precision. When the tokenized prompt exceeds the model's context window, or when decoding triggers a CUDA out-of-memory error on the available hardware, the sample is recorded as an inference failure and counted against the success rate.

Table 2: Inference cost of TimeOmni-v and its base model on AgiBot test (131 samples, the longest median TS in MMTA; single H200, max_new_tokens=8, identical dynamic-batching budgets). TS tokens count the time-series portion of the prompt only. *Completed* counts samples that finish without CUDA OOM in this run. AgiBot is reported as a worst-case TS-length study; per-sample latency and VRAM scale with TS length, the token compression ratio does not.

Model	TS tokens ↓	Per sample (s) ↓	Peak VRAM (GB) ↓	Completed ↑
Qwen2.5-Omni-7B (TS-as-text)	167,041	27.70	108.6	113 / 131
TimeOmni-v	**722**	**1.20**	**35.0**	**131 / 131**
Ratio (baseline / TimeOmni-v)	232.8×	23.0×	3.1×	—

Forecasting: swMAPE. A forecasting task asks the model to output multi-step numerical predictions over a fixed horizon. The base metric is MAPE $= (100\%/n)\sum_{i=1}^{n}|(y_i - \hat{y}_i)/y_i|$, and the headline metric is success-weighted MAPE, swMAPE $=$ MAPE$/$SuccessRate, which penalizes both large errors and frequent parsing failures.

Decomposed metrics. For every model and test set, Appendix D.1 also reports the base metric on the parsed subset, the success rate, and a failure taxonomy (CUDA OOM, other inference failures such as API refusals, and unparsable outputs). The modality-subset ablation and the controlled baselines are reported as relative changes in Macro-F1 for classification and in MAE for forecasting.

6 Results and Discussion

6.1 Main Results

Headline. TimeOmni-v attains the highest Avg-swF1 (0.601 versus 0.442 for the next model, Gemini-3-flash) and the largest single-column gains on AgiBot (0.844), CoVLA (0.759), and HoloAssist (0.694). On Avg-swMAPE TimeOmni-v is mid-pack (90.8); the three forecasting domains remain comparable across recent baselines, and TimeOmni-v does not lead them.

Where the gap concentrates. Results show that TimeOmni-v's classification advantage concentrates on long-video, long-TS samples, and is markedly smaller where one or both modalities are short or fixed-length. On the long subset (AgiBot, CoVLA, and HoloAssist), every baseline stays below 0.6 swF1, while TimeOmni-v reports the highest swF1 on each. Table 3 shows what these three share: each pairs multi-second to multi-minute video with hundreds-to-thousands of TS steps. AgiBot reaches 155.7 s of egocentric video and 4,627-step state traces; HoloAssist spans 20–80 s with 600–2,401-step sensor windows; CoVLA samples extend up to 30 s. On the short-or-fixed subset, Gemini-3-flash leads on CUHK-X (sub-second clips) by 12.0 pp, and TimeOmni-v leads on Future Factories (fixed 16-step TS) by 4.7 pp, a margin within the uncertainty of its 46 test samples (Appendix D.2). A plausible reading of this split is that when the TS is short or fixed-length, its textual serialization fits inside the LLM context with room to spare, and a strong general visual backbone is enough to read it alongside the video. When the TS becomes long, serialization instead consumes the alignment budget and crowds out the co-occurring video frames; on the two longest-series domains, where TS-as-text fine-tuning runs out of memory (Section 6.3), a dedicated TS encoder is what makes training possible.

Scale does not close the long-TS gap. On AgiBot, the 30B-class Qwen3-VL-30B-A3B-Instruct runs out of memory on every sample, and every other baseline, including the 30B-class Qwen3-Omni-30B-A3B-Instruct and both proprietary models, stays under 0.13 swF1. Increasing parameter count alone does not move either family beyond the failure mode named in Section 1: a long serialized time series consumes the alignment budget that the LLM would otherwise spend on the co-occurring video frames.

6.2 Inference Efficiency on the Worst-Case TS Domain

Token compression: dataset-independent. For an input time series of shape $T \times C$, TimeOmni-v represents the TS slice as $P \cdot C + 2$ tokens, where $P = \lceil T/p \rceil$ is the number of Chronos-2 patches at

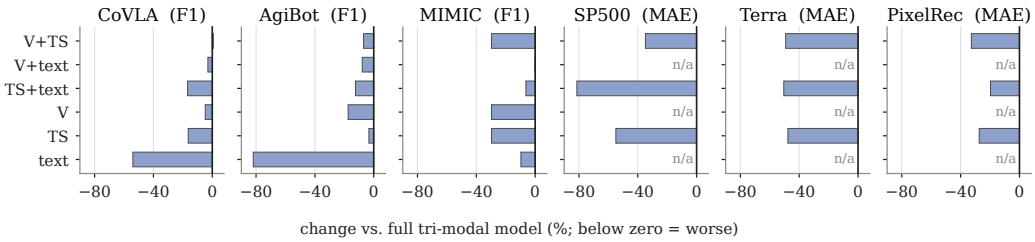


Figure 3: Training-time modality ablation. Each bar is TimeOmni-v trained and tested with the kept modalities only (V = vision), as the relative change (%) against the full tri-modal model in Macro-F1 for classification and in MAE for forecasting; MAE changes are sign-flipped so that a bar below zero is always worse. "n/a": forecasting without time series is undefined, since the forecast head requires time-series input.

patch size $p=16$ and the $+2$ are boundary tokens. Text serialization, by contrast, produces roughly $T \cdot C \cdot k$ sub-word tokens, where k counts the sub-word tokens spent on each numeric value together with the surrounding formatting (timestamps, channel names, separators, decimal digits) and is typically larger than 10 in practice. The compression ratio is therefore $TCk/(TC/p+2) \approx p \cdot k$ in the large-T, C limit, fixed by the patch size and the serialization verbosity rather than by dataset content. On AgiBot the median TS portion drops from 167,041 sub-word tokens to 722 ($232.8\times$), consistent with $p=16$ and an effective $k \approx 14.5$.

Per-sample speedup: dataset-dependent. Token compression and end-to-end speedup are not the same factor. On AgiBot per-sample latency drops $23.0\times$ and peak VRAM $3.1\times$. Video tokens, system prompt, and decoding overhead are unchanged; only the TS slice shrinks, and its share of the total prompt determines how much of the cost actually moves. The implication is that this speedup is largest on long-TS domains (AgiBot is the worst case) and shrinks toward unity on short-TS image+TS domains.

Reliability. Eighteen of the 131 AgiBot samples (14%) cause Qwen2.5-Omni-7B to exceed the 140 GB H200 activation budget in this run, whereas TimeOmni-v finishes all 131 samples. The reliability gap shares its origin with the swF1 gap: a long serialized TS is not only expensive but, on a non-trivial fraction of samples, infeasible at all.

6.3 Ablations and Controlled Comparisons

Figure 3 retrains TimeOmni-v on every single- and two-modality subset and tests it on the same subset, so each change measures what the removed modalities contribute; removing a modality only at test time measures reliance instead and is reported for all nine domains in Appendix D.3. Figure 4 compares TimeOmni-v with the controlled baselines of Section 5.1.

The dominant modality differs across datasets, but every reduced input loses somewhere. No subset improves on the full model by more than 0.7%, and each single- or two-modality subset is worse on at least two of the datasets where it is defined. Where the losses vanish, they locate the dominant modality. Dropping text costs nothing on CoVLA, whose text only states the task and the channels, so CoVLA is vision-led; dropping time series costs nothing on MIMIC, where the report and the X-ray carry more of the prognosis signal than the vital signs, so MIMIC is text-led. On AgiBot every subset loses, and on the three forecasting datasets every subset raises MAE by 19.7–81.4%. The time series alone trails the full model by 16.4% on CoVLA, so the partial overlap between its rule-derived labels and the input channels (Section 3.1) does not make the task solvable from the series.

Fine-tuning explains much of the short-series gap; the encoder makes long series trainable. The TS-as-text baseline (Figure 4a) shares the backbone, the LoRA configuration, and the training budget with TimeOmni-v, so it isolates how the time series enters the model. On MIMIC it matches TimeOmni-v (-0.4%), and on CoVLA it is ahead ($+11.0\%$), where the rule-derived labels reward

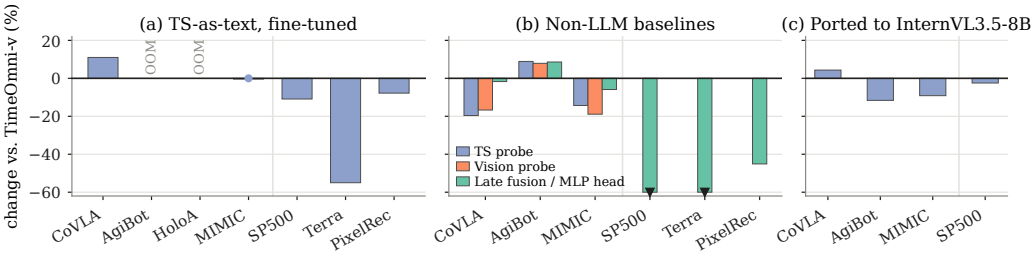


Figure 4: Controlled comparisons, as relative changes (%) in Macro-F1 for classification and in MAE for forecasting (sign-flipped, so a bar below zero is always worse than the reference). (a) Qwen2.5-Omni-7B fine-tuned with time series as text under the same LoRA configuration and budget, against TimeOmni-v; OOM: training runs out of memory on an H200 140 GB GPU at batch size one. (b) Non-LLM baselines: linear probes and late fusion on the frozen TimeOmni-v features, against TimeOmni-v; triangles mark bars that extend beyond the axis. (c) The TimeOmni-v integration ported to InternVL3.5-8B, against the Qwen2.5-Omni-7B version (swF1 and swMAPE; both parse every sample).

exact numeric readout, which lossless text keeps and patch compression does not. On AgiBot and HoloAssist the textualized prompts run out of memory, so the encoder is what makes fine-tuning possible. On forecasting the TS-as-text baseline has 7.8–55.0% higher MAE, a gap that also includes the change from digit decoding to the forecast head.

Non-LLM baselines solve some classification tasks but fail at forecasting. On AgiBot, the TS probe, the vision probe, and late fusion exceed TimeOmni-v by 7.9–8.9% (Figure 4b), whereas on CoVLA and MIMIC they trail it by 1.7–19.6%. The AgiBot labels are therefore learnable without an LLM, and the zero-shot MLLMs in Table 1 stay far below what a linear probe extracts from the same inputs. The same tri-modal features with an MLP regression head collapse on forecasting, raising MAE by 45.1% on PixelRec and by more than an order of magnitude on SP500 and Terra.

The time-series integration is not tied to one backbone. Ported to InternVL3.5-8B, a backbone with a different vision stack, the integration (Figure 4c) parses every sample and stays within 12% of the Qwen2.5-Omni-7B version on all four tested datasets, ahead on CoVLA and behind on AgiBot and MIMIC. On AgiBot its 0.746 swF1 still exceeds every zero-shot baseline in Table 1 by more than 0.6.

7 Conclusion

We introduce **MMTA**, a collection of tri-modal tasks for multimodal temporal analysis with diverse cross-modal complementarity patterns, spanning nine real-world domains, two cross-modal layouts (video+TS and image+TS), and a 3,524-sample test split that covers both classification and forecasting. We evaluate eight representative open-source and proprietary MLLMs on MMTA. Two key insights emerge. First, the dominant modality differs across domains, yet every single- and two-modality variant of the same model falls behind the tri-modal one on part of the benchmark. Second, the evaluated MLLMs share a failure mode in which serializing time series as text inflates the prompt, causes out-of-memory failures and unparsable outputs on long sequences, and leaves accuracy low even when every sample is answered; neither model scale nor proprietary tuning closes this long-sequence gap. To this end, we present **TimeOmni-v**, a framework that integrates time series into multimodal LLMs as a first-class modality while remaining compatible with the existing visual and language pretraining. On MMTA, TimeOmni-v lifts Avg-swF1 over the strongest baseline and can be fine-tuned on long-sequence domains where the same backbone with textualized time series runs out of memory. By providing a tri-modal benchmark and a preliminary integration recipe, this work takes a step toward Multimodal LLMs that read time series as faithfully as they already read vision and language, enabling end-to-end decision-making over real-world dynamical systems where evidence is distributed across modalities.

Acknowledgments and Disclosure of Funding

This work was supported by Tsinghua University and the Shanghai Artificial Intelligence Laboratory.

References

[1] Yilin Wang, Peixuan Lei, Jie Song, Yuzhe Hao, Tao Chen, Yuxuan Zhang, Lei Jia, Yuanxiang Li, and Zhongyu Wei. ITFormer: Bridging Time Series and Natural Language for Multi-Modal QA with Large-Scale Multitask Dataset. In *Proc. ICML*, Vancouver, 2025.

[2] Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. MTBench: A Multimodal Time Series Benchmark for Temporal Reasoning and Question Answering. *arXiv preprint arXiv:2503.16858*, 2025.

[3] Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, et al. Context is Key: A Benchmark for Forecasting with Essential Textual Information. In *Proc. ICML*, Vancouver, 2025.

[4] Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-MMD: Multi-Domain Multimodal Dataset for Time Series Analysis. In *Proc. NeurIPS*, Vancouver, 2024.

[5] Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-MQA: Time Series Multi-Task Question Answering with Context Enhancement. *arXiv preprint arXiv:2503.01875*, 2025.

[6] Xin Zhou, Weiqing Wang, Francisco J Baldán, Wray Buntine, and Christoph Bergmeir. MoTime: A Dataset Suite for Multimodal Time Series Forecasting. *arXiv preprint arXiv:2505.15072*, 2025.

[7] Wen Wu, Ziyang Zhang, Liwei Liu, Xuenan Xu, Junlin Liu, Ke Fan, Qitan Lv, Jimin Zhuang, Chen Zhang, Zheqi Yuan, et al. SciTS: Scientific Time Series Understanding and Generation with LLMs. *arXiv preprint arXiv:2510.03255*, 2025.

[8] AgiBot World Team. AgiBot World 2026. `https://huggingface.co/datasets/agibot-world/AgiBotWorld2026`, 2026.

[9] Geon Lee, Wenchao Yu, Kijung Shin, Wei Cheng, and Haifeng Chen. TimeCAP: Learning to Contextualize, Augment, and Predict Time Series Events with Large Language Model Agents. In *Proc. AAAI*, Philadelphia, 2025.

[10] Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From News to Forecast: Integrating Event Analysis in LLM-Based Time Series Forecasting with Reflection. In *Proc. NeurIPS*, Vancouver, 2024.

[11] Haoxin Liu, Yichen Zhou, Rajat Sen, B. Aditya Prakash, and Abhimanyu Das. Rethinking Multimodal Time-Series Forecasting Evaluation. In *1st ICLR Workshop on Time Series in the Age of Large Models*, 2026. URL `https://openreview.net/forum?id=P37UYgBjSB`.

[12] Ching Chang, Jeehyun Hwang, Yidan Shi, Haixin Wang, Wei Wang, Wen-Chih Peng, and Tien-Fu Chen. Time-IMM: A Dataset and Benchmark for Irregular Multimodal Multivariate Time Series. In *Proc. NeurIPS Datasets and Benchmarks Track*, 2025.

[13] Yiyuan Yang, Zichuan Liu, Lei Song, Kai Ying, Zhiguang Wang, Tom Bamford, Svitlana Vyetrenko, Jiang Bian, and Qingsong Wen. Time-RA: Towards Time Series Reasoning for Anomaly Diagnosis with LLM Feedback. *arXiv preprint arXiv:2507.15066*, 2025.

[14] Haoxin Liu, Chenghao Liu, and B Aditya Prakash. A Picture is Worth a Thousand Numbers: Enabling LLMs Reason about Time Series via Visualization. In *Proc. NAACL*, Albuquerque, 2025.

[15] Peng Chen, Siyuan Wang, Shiyan Hu, Xingjian Wu, Yang Shu, Zhongwen Rao, Meng Wang, Yijie Li, Bin Yang, and Chenjuan Guo. Empowering Time Series Analysis with Large-Scale Multimodal Pretraining. *arXiv preprint arXiv:2602.05646*, 2026.

[16] Yao Yin, Zhenyu Xiao, Musheng Li, Yiwen Liu, Sutong Nan, Yiting He, Ruiqi Wang, Zhenwei Zhang, Qingmin Liao, and Yuantao Gu. MMTS-Bench: A Comprehensive Benchmark for Time Series Understanding and Reasoning. *arXiv preprint arXiv:2602.08588*, 2026.

[17] Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. ChatTS: Aligning Time Series with LLMs via Synthetic Data for Enhanced Understanding and Reasoning. *arXiv preprint arXiv:2412.03104*, 2025.

[18] 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. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *Proc. ICLR*, Vienna, 2024.

[19] Paul Quinlan, Qingguo Li, and Xiaodan Zhu. Chat-TS: Enhancing Multi-Modal Reasoning Over Time-Series and Natural Language Data. *arXiv preprint arXiv:2503.10883*, 2025.

[20] Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. GPT4MTS: Prompt-based Large Language Model for Multimodal Time-series Forecasting. In *Proc. AAAI*, Vancouver, 2024.

[21] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni Technical Report. *arXiv preprint arXiv:2503.20215*, 2025.

[22] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-Omni Technical Report. *arXiv preprint arXiv:2509.17765*, 2025.

[23] Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. CoVLA: Comprehensive Vision-Language-Action Dataset for Autonomous Driving. In *Proc. WACV*, Tucson, 2025.

[24] Siyang Jiang, Mu Yuan, Xiang Ji, Bufang Yang, Zeyu Liu, Lilin Xu, Yang Li, Yuting He, Liran Dong, Wenrui Lu, et al. A Large-Scale Multimodal Dataset and Benchmarks for Human Activity Scene Understanding and Reasoning. In *Proc. MobiSys*, pages 352–370, Cambridge, 2026. doi: 10.1145/3745756.3809209.

[25] Ramy Harik, Fadi El Kalach, Jad Samaha, Devon Clark, Drew Sander, Philip Samaha, Liam Burns, Ibrahim Yousif, Victor Gadow, Theodros Tarekegne, et al. Analog and Multi-modal Manufacturing Datasets Acquired on the Future Factories Platform. *arXiv preprint arXiv:2401.15544*, 2024.

[26] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. HoloAssist: An Egocentric Human Interaction Dataset for Interactive AI Assistants in the Real World. In *Proc. ICCV*, pages 20270–20281, Paris, 2023.

[27] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.

[28] Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

[29] Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV-Note: Deidentified free-text clinical notes. *PhysioNet*, January 2023. doi: 10.13026/1n74-ne17. URL `https://doi.org/10.13026/1n74-ne17`. Version 2.2.

[30] Yu Cheng, Yunzhu Pan, Jiaqi Zhang, Yongxin Ni, Aixin Sun, and Fajie Yuan. An Image Dataset for Benchmarking Recommender Systems with Raw Pixels. In *Proc. SDM*, Houston, 2024.

[31] Wenyan Xu, Dawei Xiang, Yue Liu, Xiyu Wang, Yanxiang Ma, Liang Zhang, Shu Hu, Chang Xu, and Jiaheng Zhang. FinMultiTime: A Four-Modal Bilingual Dataset for Financial Time-Series Analysis. *arXiv preprint arXiv:2506.05019*, 2025.

[32] Wei Chen, Xixuan Hao, Yuxuan Liang, et al. Terra: A Multimodal Spatio-Temporal Dataset Spanning the Earth. In *Proc. NeurIPS*, Vancouver, 2024.

[33] Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From Univariate to Universal Forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

[34] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[35] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025.

[36] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *Proc. ICLR*, Virtual, 2022.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. NAACL-HLT*, Minneapolis, 2019.

Table 3: Standardized dataset statistics of MMTA. *Layout* indicates the cross-modal layout (V+TS = video+TS; I+TS = image+TS). *Train* and *Test* are the split sizes. *Duration* is the per-sample video length in seconds (min–max), reported only for V+TS domains. *Seq. Len.* is the time-series length per sample, *Dim.* is the number of time-series channels, and *Visual* summarizes the per-sample visual content. The MIMIC entry merges the mortality and discharge subtasks (75 test samples each).

Dataset	Layout	Train	Test	Duration (s)	Seq. Len.	Dim.	Visual	Task
AgiBotWorld	V+TS	1,177	131	7.9–155.7	194–4,627	10	egocentric clip	Classification (4-way)
CoVLA	V+TS	13,437	1,344	0.3–30.0	6–600	5	dashcam clip	Classification (3-way)
CUHK-X	V+TS	2,629	308	0.05–16.05	1–816	7	skeleton clip	Classification (40-way)
Future Factories	V+TS	49	46	56.8–68.4	16 (fixed)	36	16-frame surveillance	Classification (4-way)
HoloAssist	V+TS	4,991	945	20.0–80.4	600–2,401	9	egocentric clip	Classification (binary)
MIMIC	I+TS	150	150	—	4–39	6	1–5 chest X-rays	Classification (binary)
PixelRec	I+TS	931	200	—	14–38	1	1 cover image	Forecasting
FinMultiTime	I+TS	3,368	200	—	82–124	1	1 candlestick chart	Forecasting
Terra	I+TS	2,000	200	—	100 (fixed)	7	4–7 remote-sensing tiles	Forecasting

A Real-World Multimodal Temporal Datasets in MMTA

This appendix complements Section 3 with the dataset-level details that do not fit into the main text. For each of the nine MMTA domains, the corresponding subsection describes (i) the source corpus and the modalities it supplies, (ii) the construction pipeline that turns raw releases into one MMTA instance under the unified schema, and (iii) a representative prompt example that shows what the model actually sees at evaluation time. Three modalities recur across the corpus: **time series** (physiological signals, joint and gripper traces, vehicle states, IMU streams, gaze and head trajectories, daily interaction counts, stock prices, climate variables); **vision** (egocentric RGB clips, dashcam segments, skeleton renders, on-site surveillance video, chest X-rays, item thumbnails, candlestick charts, remote-sensing imagery); and **text** (task instructions, fine-grained action sequences, cycle reports, radiology reports, item metadata, news snippets, geographic descriptions). Domains are grouped by cross-modal layout: five *video+TS* domains first, then four *image+TS* domains. Standardized statistics over the 3,524-sample MMTA test split are summarized in Table 3.

A.1 Robotics: AgiBotWorld

Source and modalities. AgiBotWorld2026 [8] is a robotics dataset for embodied AI and imitation learning, focused on real-world robotic manipulation. The MMTA subset is built from selected task groups with complete episode metadata and synchronized visual/state streams. Each episode contains an explicit task instruction, a stepwise execution trace, and synchronized state and action trajectories. The visual modality is the top-head camera clip (`observation.images.top_head`) over the full Task Frame, providing context around each error moment. The temporal modality contains 10 channels extracted from `observation.state`: six arm-position channels (`arm_pos_0`–`arm_pos_5`) and four end-effector velocity channels (`end_vel_0`–`end_vel_3`), with timestamps and frame indices retained as alignment metadata. The text modality combines the task instruction with the ordered fine-grained step sequence. **Data source:** `https://huggingface.co/datasets/agibot-world/AgiBotWorld2026`.

Atomic instance and label. An atomic instance is one Task Frame within an episode. The model classifies the dominant anomaly cause among four labels: *paused* (A), *meaningless operation* (B), *fall* (C), or *other* (D). Task Frames containing multiple anomaly types are dropped to keep the label space single-valued. The final export contains 1,308 instances (1,177 train / 131 test).

Processing pipeline. (1) Read structured episode metadata for each usable segment and collect task-frame and instruction annotations. (2) Identify Task Frame candidates from annotations whose frame type indicates a Task Frame. (3) Collect error frames whose midpoints fall inside each Task Frame and map raw error descriptions onto the four anomaly categories. (4) Load the corresponding episode trajectory, slice the state tensor over the Task Frame interval, and align the 10-channel state stream with the cropped video clip on the same timeline. (5) Assemble the prompt from the task instruction, ordered step sequence, cropped video, and aligned state traces.

AgiBotWorld (concrete instance)

task_type: QA **video:** top-head egocentric clip, 640×400, 30 fps, 16.67 s, 500 frames. **ts_shape:** 452×10.

scene description:

This clip is from AgiBot World in a real retail shelf-rearrangement scene. The robot performs dual-arm manipulation near shelves and shopping carts.

text (task instruction + step sequence):

Task instruction: *Adjust the orange sausage set to the correct position on the shelf.*
Step sequence:
1. [Straighten] Adjust products.
2. [Other] Adjust the position of the orange sausage set with the left arm.

time series (10-channel state, format `timestamp: arm_pos_0=..., ..., end_vel_3=...`):

12.400: arm_pos_0=0.356311, arm_pos_1=−0.029625, arm_pos_2=0.378649, . . . , end_vel_0=0.053887, end_vel_1=0.416501, end_vel_2=−0.360235, end_vel_3=1.185560
12.433: arm_pos_0=0.358105, arm_pos_1=−0.018230, arm_pos_2=0.391642, . . . , end_vel_0=0.057594, end_vel_1=0.436849, end_vel_2=−0.387070, end_vel_3=1.130598
. . .
27.433: arm_pos_0=0.381310, arm_pos_1=−0.083640, arm_pos_2=0.318216, . . . , end_vel_0=0.000105, end_vel_1=−0.000011, end_vel_2=−0.000281, end_vel_3=−0.000038

prompt (abridged):

```
You are a robot video anomaly-reason classification assistant.

<scene_description> scene description </scene_description>

<video_capture>
The video is captured by the robot top-head camera
(observation.images.top_head), approximately 30 fps. Important: this
clip contains a complete task-frame segment, not only the short error
moment.
</video_capture>

<task_instruction> instruction </task_instruction>
<step_sequence> steps </step_sequence>
```

```
<error_definitions>
- A paused: operation is paused or stalled
- B meaningless operation: ineffective or redundant operation behavior
- C fall: object drops or a falling-related error occurs
- D other: other error causes not covered above
</error_definitions>

<numeric_field_explanations>
arm_pos_{0..5}: end-effector / arm position channels;
end_vel_{0..3}: end-effector velocity sub-channels.
</numeric_field_explanations>

<timeseries> time series </timeseries>
```

question: Based on the video, task instruction, step sequence, and timeseries above, what is the most appropriate anomaly cause category for this task-frame segment?
options: A. paused. B. meaningless operation. C. fall. D. other.
expected answer: A.

A.2 Autonomous Driving: CoVLA

Source and modalities. The CoVLA Dataset [23] consists of driving videos recorded on Japanese public roads, together with annotations generated with the assistance of VideoLLaMA 2. In the MMTA processing corpus, 6,010 driving sequences provide complete `states` and `front_car` annotations for segment extraction. The visual stream is a temporally cropped braking segment from the original dashcam clip, focused on the event window where the braking decision occurs. The temporal stream contains five synchronized driving variables: `vEgo`, `aEgo`, `steeringAngleDeg`, `lead_speed_kmh`, and `lead_a`. The upstream brake and gas signals are used to locate the braking interval but are not exposed as model inputs. The text modality consists of the scenario description and a short scene caption. **Data source:** `https://huggingface.co/datasets/turing-motors/CoVLA-Dataset`.

Atomic instance and label. An atomic instance is one braking segment, defined as a continuous interval where `brakePressed` stays true for at least six frames. The model performs three-way brake-reason classification: *lead-vehicle deceleration* (A), *ego-vehicle turning* (B), or *other reasons* (C). The final export contains 14,781 instances (13,437 train / 1,344 test).

Processing pipeline. (1) Read the per-frame vehicle-state and front-vehicle annotations. (2) Detect continuous segments where `brakePressed` remains true for at least six frames. (3) Assign the brake-reason label using synchronized evidence: `A` if the lead vehicle shows sustained deceleration; `B` if the ego vehicle exhibits turning behavior; `C` otherwise. (4) Slice the aligned per-frame signals to the braking window, convert timestamps to seconds relative to the video start, and assemble the prompt from the cropped clip and the five-channel state stream.

CoVLA (concrete instance)

task_type: QA **video:** dashcam clip, 1928×1208, 20 fps, 4.65 s, 93 frames. **ts_shape:** 93×5.

variable definitions:

`vEgo`: ego speed (m/s). `aEgo`: longitudinal acceleration (m/s^2, negative $\Rightarrow$ deceleration). `steeringAngleDeg`: steering wheel angle in degrees (large $|\cdot| \Rightarrow$ turning). `lead_speed_kmh`: leading-vehicle speed (km/h, `null` when no lead detected). `lead_a`: leading-vehicle acceleration (m/s^2, `null` when no lead detected).

time series (per-frame, braking onset $\rightarrow$ offset, format `t(s): vEgo=, aEgo=, steeringAngleDeg=, lead_speed_kmh=, lead_a=`):

25.5: vEgo=6.517, aEgo=0.197, steeringAngleDeg=−16.5, lead_speed_kmh=null, lead_a=null
25.5: vEgo=6.505, aEgo=0.149, steeringAngleDeg=−17.6, lead_speed_kmh=null, lead_a=null
...
30.0: vEgo=1.561, aEgo=−0.691, steeringAngleDeg=−2.0, lead_speed_kmh=null, lead_a=null
30.0: vEgo=1.578, aEgo=−0.505, steeringAngleDeg=15.2, lead_speed_kmh=null, lead_a=null

prompt (abridged):

```
You are given a question-answering (QA) task for autonomous driving.
You are provided with a dashcam video segment from an autonomous driving
system (ego vehicle) traveling on public roads. This video is already
temporally cropped to only the braking event, and the per-frame sensor
data below is aligned to this same braking segment. Your task is to
determine the most likely reason for the sudden braking in this segment.
Do NOT output any analysis. Return only the option letter.

<video>
This video clip contains only the braking segment. The braking segment
contains 93 frames. No pre-brake or post-brake frames are included; the
sensor data below also contains only this exact braking segment.
</video>

<variables> variable definitions </variables>
<timeseries> time series </timeseries>
```

question: Based on the video content and the sensor data above, why did the ego vehicle suddenly brake during the segment?
options: A. The leading vehicle suddenly decelerated. B. The ego vehicle was making a turn. C. Other reason (e.g., pedestrian about to cross, traffic light turning red, entering a tunnel).
expected answer: C.

A.3 Human Action: CUHK-X

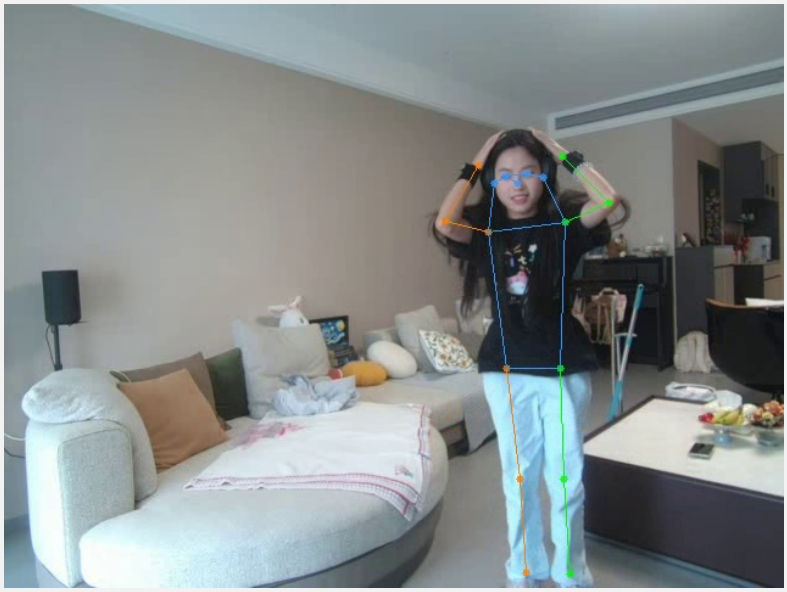
Source and modalities. CUHK-X [24] is a large-scale multimodal dataset for human action recognition, understanding, and reasoning, containing 64,342 samples collected from 30 participants across diverse environments. It supplies seven synchronized modalities including RGB, infrared, depth, thermal, IMU, and other sensor streams. The MMTA subset uses CUHK-S/HAR and constructs a pose- and inertial-sensor based action classification task. The visual stream is a synthetic skeleton-rendered clip from the HAR release, so body-pose evolution over time is encoded directly in the video. The temporal stream consists of five wearable IMU streams mapped to a common device space (`WTC`, `WTLA`, `WTLL`, `WTRA`, `WTRL`); each step contains `time_s`, `device_id`, three accelerometer channels (`acc_x_g`, `acc_y_g`, `acc_z_g`), and three gyroscope channels (`gyro_x_dps`, `gyro_y_dps`, `gyro_z_dps`). **Data source:** `https://github.com/openaiotlab/CUHK-X`.

Atomic instance and label. An atomic instance is one action trial. The model classifies the action over 40 categories (`action_id` 0–39). The final export contains 2,937 instances (2,629 train / 308 test).

Processing pipeline. (1) Match each skeleton-rendered clip with the corresponding action trial and IMU records using the dataset-provided hierarchy. (2) Merge the available IMU streams in each trial, clean column names, and sort rows by timestamp. (3) Normalize device names and derive a stable `device_id` mapping. (4) Downsample each second independently for each device, keeping up to five representative samples per second. (5) Use relative time from the first sampled IMU record and align the resulting sequence with the corresponding skeleton clip.

CUHK-X (concrete instance)

task_type: action classification (40-way) **video:** skeleton clip, 640×640, 20 fps, 0.75 s, 15 frames. **ts_shape:** 43×7.



device definitions:

`device_id=0`: WTC (chest). `device_id=1`: WTLA (left arm). `device_id=2`: WTLL (left leg). `device_id=3`: WTRA (right arm). `device_id=4`: WTRL (right leg).

action space (excerpt):

0: Wash_face, 1: Brush_teeth, 2: Comb_hair, ..., 28: Jog_in_place, **29: Do_squats**, 30: Do_jumping_jacks, ..., 39: Take_body_temperature.

time series (one row per device per timestamp, format `time_s: device_id=N, acc_x_g=, acc_y_g=, acc_z_g=, gyro_x_dps=, gyro_y_dps=, gyro_z_dps=`):

0.000: device_id=4, acc_x_g=−0.31, acc_y_g=0.63, acc_z_g=−0.75, gyro_x_dps=116, gyro_y_dps=90, gyro_z_dps=−4
0.004: device_id=1, acc_x_g=−0.13, acc_y_g=0.64, acc_z_g=0.72, gyro_x_dps=5, gyro_y_dps=112, gyro_z_dps=29
0.041: device_id=3, acc_x_g=−0.17, acc_y_g=−0.53, acc_z_g=0.83, gyro_x_dps=11, gyro_y_dps=34, gyro_z_dps=−2
...
1.380: device_id=1, acc_x_g=−0.58, acc_y_g=−0.08, acc_z_g=−0.11, gyro_x_dps=−3, gyro_y_dps=59, gyro_z_dps=−11

prompt (abridged):

```
You are solving an action-classification task for CUHK-X multimodal HAR.
The video segment and downsampled IMU samples belong to the same action
instance.

<field_definitions>
time_s: relative timestamp (s) from the first sampled IMU record;
device_id: numeric device identifier (see <device_definitions>);
acc_{x,y,z}_g: acceleration (g);
gyro_{x,y,z}_dps: angular velocity (deg/s).
</field_definitions>

<device_definitions> device defs </device_definitions>
<action_space> 40 action labels </action_space>
<timeseries> time series </timeseries>
```

question: Which action category (action_id) best matches this segment? Return only one integer action_id from the action_space.
expected answer: 29 (*Do_squats*).

A.4 Manufacturing: Future Factories

Source and modalities. The Future Factories Platform [25] is a multimodal dataset family for industrial monitoring, capturing manufacturing production cycles with synchronized PLC sensor streams, on-site camera footage, and textual cycle reports. The MMTA subset is collected from an industry-grade assembly line at the Future Factories Lab, University of South Carolina; the processed source stream is grouped into 95 production cycles. The physical process performs collaborative assembly of a Lego-style modular rocket comprising four components: a base, two body segments (Body1, Body2), and a nose cone. Robot R01 introduces the tray; R02 retrieves the body segments; R03 mounts them onto the base before adding the nose cone; R04 then performs disassembly to reset the cycle. The visual stream is the on-site surveillance footage captured by two synchronized cameras monitoring the assembly station. The temporal stream contains 36 PLC channels: 12 gripper-related signals and 24 robot joint-angle channels. The text modality provides a cycle report with process overview, signal definitions, and time range. **Data source:** `https://www.kaggle.com/datasets/ramyharik/ff-2024-08-13-multi-modal-dataset-33`.

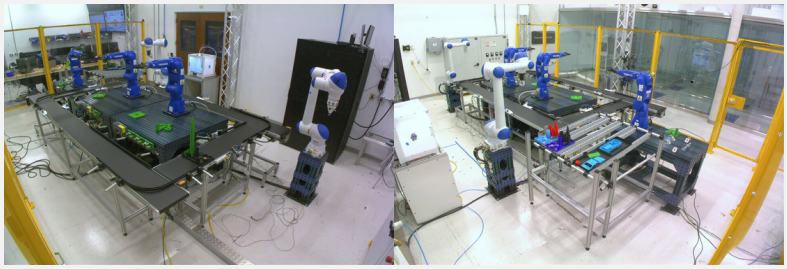
Atomic instance and label. An atomic instance is one production cycle. Defects are introduced by manually removing required parts before the cycle starts; the model classifies the cycle into one of four labels: *Correct* (A; 10 samples, 10.53%), *NoNose* (B; 33, 34.74%), *NoNose,NoBody2* (C; 11, 11.58%), *NoNose,NoBody2,NoBody1* (D; 41, 43.16%). The final export contains 95 instances (49 train / 46 test).

Processing pipeline. (1) Parse the timestamped source records and sort them by timestamp. (2) Segment the stream into 95 production cycles using the cycle delimiters in the source. (3) Downsample each cycle by keeping the last record in every 2-second bucket. (4) Sample 16 timestamps uniformly across the cycle; at each sampled timestamp, the two camera views are arranged side-by-side and annotated with the timestamp, producing a compact visual summary of the cycle. (5) Align the 36-channel PLC trace to the same 16 sampled timestamps, and build the prompt from the cycle report text, the visual summary, and the aligned time series.

Future Factories Platform

task_type: QA

visual: 16 paired surveillance frames sampled uniformly across the cycle.



text:

Time range (UTC): xxx Cycle ID: xxx Records: xxx
Process overview: xxx
Selected signals and meanings: xxx
Image description (cycle-specific sampled frames): xxx

time series (16 sampled timestamps, 36 channels):

timestamp:
I_R0X_Gripper_Pot=..., I_R0X_Gripper_Load=..., M_R0X_SJointAngle_Degree=...,
M_R0X_LJointAngle_Degree=..., M_R0X_UJointAngle_Degree=...,
M_R0X_RJointAngle_Degree=..., M_R0X_BJointAngle_Degree=..., M_R0X_TJointAngle_Degree=...
.....

prompt:

```
You are given a question-answering (QA) task. Scenario: this cycle
comes from the Future Factories testbed. A tray is introduced at the
material handling station (MHS), transported by a closed-loop conveyor
between workcells, assembled by robots (R01-R03), and later disassembled
by R04 before the tray returns to MHS. The goal is to judge whether the
cycle proceeds normally or if a missing-part anomaly occurs. Use the
sampled surveillance frames, the cycle report, and the aligned PLC
sensor signals as evidence. Return only the option letter.

Note: the timestamps and time-series rows below are aligned to the
sampled image frames above (one row per sampled frame, time-ordered).

<report>
Cycle report text:
text
</report>

<image>
image
</image>

<timeseries>
Aligned time series (each line is one timestamp; missing values empty):
```

```
time series
</timeseries>
```

question: Did anything go wrong during this production cycle? Select the single best option that describes the observed issue (or choose *correct* if no issue is present).
options: A. correct B. NoNose C. NoNose,NoBody2 D. NoNose,NoBody2,NoBody1

A.5 AR Assistance: HoloAssist

Source and modalities. HoloAssist [26] is a large-scale egocentric human interaction dataset for studying interactive AI assistants on real-world physical tasks. The annotation dump used in MMTA processing contains 1,758 videos, 12,542 coarse-grained action annotations, and 149,253 fine-grained action annotations. The dataset is suitable for temporal benchmarking because each coarse action may contain multiple fine-grained actions, some of which are annotated as incorrect and may later be corrected by the instructor or the performer. The visual stream is the egocentric RGB clip cropped to one coarse-grained action segment, so unrelated pre- and post-segment content is excluded. The temporal stream is built from synchronized HoloLens streams aligned to the RGB timeline: gaze direction (`gaze_dir_x`, `gaze_dir_y`, `gaze_dir_z`), head translation (`tx`, `ty`, `tz`), and gyroscope measurements (`gyro_x`, `gyro_y`, `gyro_z`). The text modality is the ordered fine-grained action sequence inside the coarse segment. **Data source:** `https://holoassist.github.io/`.

Atomic instance and label. An atomic instance is one coarse-grained action segment with duration between 20 and 80 seconds. The model performs binary correctness judgment: *all contained fine-grained actions are correct* (A) versus *at least one contained fine-grained action is incorrect* (B). The final export contains 5,936 instances (4,991 train / 945 test).

Processing pipeline. (1) Load the released annotations and enumerate all coarse-grained action segments. (2) Keep only coarse segments with duration between 20 and 80 seconds. (3) Retain segments containing at least one fine-grained action midpoint inside the coarse interval. (4) Assign the binary label by checking whether any contained fine-grained action has `Action Correctness` different from `Correct Action`. (5) Slice the synchronized eye, head-pose, and gyroscope streams over the same time window, then trim them to a common length for one-to-one temporal alignment. (6) Build the prompt from the cropped video, ordered fine-grained action sequence, and aligned sensor traces.

HoloAssist (concrete instance)

task_type: QA (binary correctness) **video:** egocentric RGB clip, 454×256, 24.46 fps, 25.55 s, 625 frames. **ts_shape:** 755×9.

coarse action:

Action sentence: *The student assembles the mounting_peg.*
Verb: `assemble`. Adjective: `none`. Noun: `mounting_peg`.

fine-grained action sequence (10 ordered actions):

1. rotate gopro 2. open leg 3. approach mounting_peg 4. grab mounting_peg 5. insert mounting_peg 6. grab screw 7. insert screw 8. screw screw 9. rotate gopro 10. slide mounting_peg.

variable definitions:

`gaze_dir_{x,y,z}`: components of the performer's gaze direction vector. `tx, ty, tz`: head translation along $x/y/z$ in space. `gyro_{x,y,z}`: gyroscope angular velocity around each axis (headset rotational motion).

time series (per-frame, format `t(s): gaze_dir_x=, gaze_dir_y=, gaze_dir_z=, tx=, ty=, tz=, gyro_x=, gyro_y=, gyro_z=`):

326.600: gaze_dir_x=0.490672, gaze_dir_y=0.326931, gaze_dir_z=−0.807686, tx=0.156120, ty=2.915354, tz=−0.046323, gyro_x=−0.009175, gyro_y=−0.012515, gyro_z=0.056830
326.633: gaze_dir_x=0.490563, gaze_dir_y=0.327070, gaze_dir_z=−0.807696, tx=0.156341, ty=2.915125, tz=−0.046315, gyro_x=−0.032264, gyro_y=−0.004942, gyro_z=0.033517
...
351.733: gaze_dir_x=0.401040, gaze_dir_y=0.370614, gaze_dir_z=−0.837743, tx=0.321765, ty=2.821166, tz=−0.070192, gyro_x=0.067685, gyro_y=0.025116, gyro_z=−0.006590

prompt (abridged):

```
You are given a question-answering (QA) task for egocentric human
activity understanding. You are provided with a temporally cropped video
segment from the HoloAssist dataset. The video segment corresponds to
one coarse-grained action performed by the wearer. This coarse-grained
action may contain multiple fine-grained actions. You are also provided
with synchronized sensor data aligned to this same coarse-grained action
segment. Your task is to judge whether all contained fine-grained
actions are correct, or whether at least one contained fine-grained
action is incorrect. Do NOT output any analysis. Return only the option
letter.

<video>
This video clip contains only the target coarse-grained action segment.
</video>

<coarse_action> coarse action </coarse_action>
<fine_actions> fine-grained sequence </fine_actions>
<variables> variable defs </variables>
<timeseries> time series </timeseries>
```

question: Based on the video content, the fine-grained action sequence, and the sensor data above, does this coarse-grained action segment contain any incorrect fine-grained action?
options: A. All contained fine-grained actions are correct. B. At least one contained fine-grained action is incorrect.
expected answer: A.

A.6 Healthcare: MIMIC

Source and modalities. The MIMIC collection is a suite of large-scale, de-identified critical-care datasets released via PhysioNet, spanning complementary clinical modalities collected in routine hospital practice. **MIMIC-IV** [27] provides longitudinal, patient- and encounter-level structured EHR data from ICU and hospital stays, including demographics, admissions and transfers, diagnoses and procedures, medication prescriptions and administrations, laboratory measure-

ments, and detailed time-stamped event records. **MIMIC-IV-Note** [29] releases free-text clinical narratives, such as discharge summaries and other note types, aligned to hospital encounters. **MIMIC-CXR-JPG** [28] provides chest radiographs in JPG format together with metadata, enabling joint study of imaging content and patient episodes when used with the structured and textual components. **Data sources:** MIMIC-IV (`https://physionet.org/content/mimiciv/3.1/`), MIMIC-IV-Note (`https://physionet.org/content/mimic-iv-note/2.2/`), MIMIC-CXR-JPG (`https://physionet.org/content/mimic-cxr-jpg/2.0.0/`).

Atomic instance and label. An atomic instance is anchored at one of two clinically meaningful time points: an imaging-study event (e.g., a chest X-ray acquisition) or a fixed 48-hour mark after hospital admission. The visual stream is the chest X-ray at the anchor; the temporal stream is six routinely recorded vital signs (heart rate, respiratory rate, SpO_2, temperature, NIBP systolic, NIBP diastolic) over the observation window; the text modality is the radiology report aligned to the same imaging study. Two prognosis-style binary tasks are evaluated: *in-hospital mortality prediction* (using the first 48 hours after admission, predict whether the patient ultimately dies during the hospital stay) and *discharge readiness prediction* (using a given 48-hour window, predict whether the patient should be discharged within the next 48 hours). The mortality task contains 20 mortality and 130 survival cases (150 samples); the discharge task contains 50 discharge and 100 non-discharge cases (150 samples).

Processing pipeline. For each anchor, a fixed observation window of vital-sign time series is extracted prior to the anchor time to provide structured physiologic context. The chest X-ray and the corresponding radiology report associated with the anchor event supply complementary visual and textual evidence aligned to the same encounter. Both tasks are formulated as QA-style multiple-choice prompts requiring integration of physiological measurements with radiology and text documentation.

Mimic

task_type: QA

image:

text:

Radiology report(s) (RR notes) associated with the imaging evidence:
[RR Note 1 | charttime=xxx]:
[RR Note 2 | charttime=xxx]:
.....

time series:

timestamp:
heart_rate_bpm=, resp_rate_insp_per_min=, spo2_percent=, temp_c=, nbp_systolic_mmhg=, nbp_diastolic_mmhg=

prompt_death:

`You are given a question-answering (QA) task. Use the chest X-ray`

```
image(s), radiology report(s), and vital signs as evidence to answer
the multiple-choice question. Do NOT output any analysis. Return only
the option letter.

<image>
MIMIC-CXR-JPG images: each image is a JPG converted from an original
DICOM chest X-ray.
image
</image>

<report>
Radiology report(s) (RR notes):
text
</report>

<variables>
Vital sign variables: an introduction to each variable.
</variables>

<timeseries>
Aligned time series (each line is one timestamp; missing values empty):
time series
</timeseries>
```

question_death: Based on the chest X-ray image(s), radiology report(s), and the ICU first-48-hour vital signs, will the patient die during this hospital admission?
question_disch: Using the chest X-ray image(s), radiology report(s), and the vital signs observed in the 48-hour lookback window ending at the anchor time, will the patient be discharged within the next 48 hours after the anchor time?
options: A. Yes B. No

A.7 Recommendation: PixelRec

Source and modalities. PixelRec [30] is a real-world recommendation dataset that exposes raw item-side cover images as first-class signals for modeling. Alongside covers, it ships standard recommender-system records, including user–item interaction logs and item metadata (titles, tags, descriptions when available), which together support studying how visual presentation and textual context correlate with user feedback. Unlike benchmarks that only release pre-extracted visual embeddings, PixelRec makes the original cover imagery directly accessible. **Data source:** `https://github.com/westlake-repl/PixelRec`.

Atomic instance and task. An atomic instance is one item. Item-level forecasting instances are constructed by aggregating raw user–item interactions into a daily time series per item (daily interaction counts). To reduce irregularity from sparsity, a long consecutive segment of daily observations is retained per item, overly long histories are truncated to a fixed maximum length, and the last several days are reserved as forecast targets. Each instance combines three modalities: time series (historical daily counts as the primary signal), text (concatenation of title, tag, and description), and a single cover image. The model is asked to forecast the next 6 days of interaction counts.

Processing pipeline. A forecasting-oriented prompt is used that conditions on (i) item metadata, (ii) the cover image, (iii) the historical daily interaction series as ordered `(timestamp, value)` pairs, and (iv) a list of future timestamps to be predicted.

PixelRec

task_type: prediction
image:

text:

title: xxx tag: xxx description: xxx

time series (time, interaction volume):

(2021-11-24, 41)
(2021-11-25, 26)
(2021-11-26, 11)
(2021-11-27, 5)
(2021-11-28, 5)
.....

prompt:

```
I have a time series forecasting task for you.
Here is some context about the item (title/tag/description).
<context>
text
</context>

Here is the cover of the item.
image

Here is a historical time series. Timestamps are daily,
values are daily interaction counts.
<history>
time series
</history>
Now please predict the value at the following timestamps:
Return the forecast in (timestamp, value) format in between
<forecast> and </forecast> tags.
Do not include any other information (e.g., comments) in the forecast.
```

task: Predict the next 6 time points of the interaction volume.

A.8 Finance: FinMultiTime

Source and modalities. FinMultiTime [31] is a multimodal financial forecasting dataset centered on market time series as prediction targets (e.g., stock price/return movements). It pairs the core numerical signals with heterogeneous auxiliary evidence, including textual financial news and visual K-line/candlestick charts, enabling research on how narrative events and visualized technical patterns complement numerical dynamics. The corpus covers equities from the China A-share market and the S&P 500 universe, supporting cross-market evaluation under diverse regimes. **Data source:** `https://huggingface.co/datasets/Wenyan0110/Multimodal-Dataset-Image_Text_Table_TimeSeries-for-Financial-Time-Series-Forecasting`.

Atomic instance and task. The MMTA subset is built on the S&P 500 universe. An atomic instance is one asset paired with a half-year history window. Given a fixed history interval (e.g., the first half of a year), the model predicts the closing prices for the subsequent horizon. Each instance includes three modalities: the historical close-price sequence within the half-year window (numeric),

a candlestick chart summarizing the same period (visual), and a small set of recent news snippets selected strictly from before the prediction timestamps (text).

Processing pipeline. Forecasting-style prompts are used; they condition on the aligned multimodal evidence (price history, candlestick image, and pre-horizon news) and ask the model to predict closing prices at specified future dates. The model outputs 12 trading-day forecasts per instance.

FinMultiTime

task_type: prediction

image:

text:

Recent news summaries (most recent last):
.....

time series:

(2024-01-02, 168.8)
(2024-01-03, 165.2)
(2024-01-04, 165.1)
.....

prompt:

```
I have a time series forecasting task for you.
Here is some context and a candlestick chart image about the task.
Make sure to factor in any background knowledge, satisfy any
constraints, and respect any scenarios.
<context>
text
</context>

<image>
Candlestick image: this image shows the historical candlestick chart
for the asset.
image
</image>

<history>
Here is a historical time series in (timestamp, value) format:
time series
</history>

Now please predict the value at the following timestamps: xxx
Return the forecast in (timestamp, value) format in between
<forecast> and </forecast> tags.
Do not include any other information (e.g., comments) in the forecast.
```

task: Predict the next 12 time points of the closing price.

A.9 Climate: Terra

Source and modalities. Terra [32] is a global-scale multimodal spatio-temporal dataset constructed on a 1° latitude–longitude grid (64,800 spatial cells). It provides long-horizon climate time series from 1979-01-01 to 2024-01-01 at three temporal resolutions (Monthly, Daily, 3-Hourly). Time-series variables include precipitation, temperature, surface pressure, relative humidity, specific humidity, wind speed, and downward shortwave and longwave radiation. Beyond numerical climate fields, Terra provides aligned image modalities for each spatial cell, including satellite imagery and geophysical map layers such as geoid, free-air anomaly (FAA), magnetic anomaly (MA), land–sea mask, relief, and vertical gravity gradient (VGG). It also exposes two levels of text: (i) per-cell metadata (coordinates, city, vegetation, climate, elevation), and (ii) LLM-generated descriptions that expand the metadata into richer semantic narratives. **Data source:** `https://github.com/CityMind-Lab/NeurIPS24-Terra?tab=readme-ov-file`. **Paper:** `https://openreview.net/pdf?id=IOzpivKOAO`.

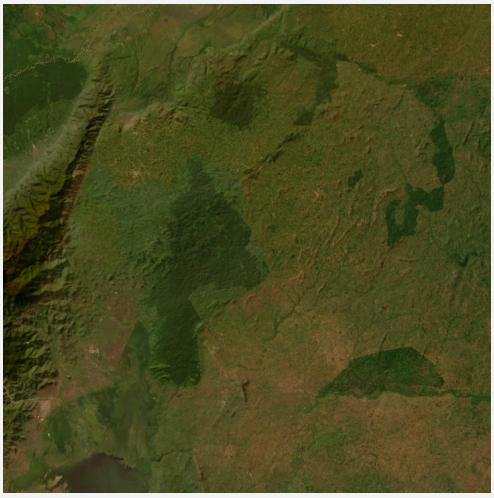
Atomic instance and task. An atomic instance is one grid cell. At each location, a daily time series is extracted and a 120-day window is formed; the first 100 days are used as historical context and the next 20 days are forecast. To construct a tractable subset, valid grid locations are randomly sampled over the globe under basic geographic constraints on latitude and longitude. Each sampled location yields one multimodal instance, which encourages models to learn patterns that generalize across regions. The remote-sensing imagery and related geospatial raster products provide static or slowly varying spatial context. The location-conditioned LLM-generated descriptions provide a compact semantic prior that complements numeric history and imagery.

Processing pipeline. Prediction-style prompts are used: given the aligned multimodal context (historical daily series, associated imagery, location text), the model outputs the next 20 daily values for all variables in the same order.

Terra

task_type: prediction

image:



text:

LLM-generated description of the grid cell, expanded from the dataset's per-cell metadata (coordinates, city, vegetation, climate, elevation).
.....

time series (example):

timestamp: lwd, precipitation, pres, relhumidity, spechumidity, temp, wind
.....

prompt:

```
You are given a geospatial multivariate time series forecasting task on
the Terra 1*1 global grid. We focus on a single grid cell identified by:
- Grid label: xxx

<image>
Each image is aligned to the same grid cell "{label}" and provides
complementary static geophysical context.
image
</image>

You are given multiple historical variables for the same grid cell and
timestamps.
A description of each variable
All variables are aligned on the same timestamps and must be forecast
in the SAME order.

<text>
The following description is generated from metadata and provides
semantic/geographic context for the same grid cell:
text
</text>

Given the historical multivariate time series, predict the future
values at the next timestamps xxx. You must output forecasts for ALL
variables listed below, in the SAME order.

Variable order:
xxx

Historical time series:
<history>
time series
</history>

Forecast timestamps:
xxx

## Output format
Return ONLY the forecast lines between <forecast> and </forecast> tags.
Each forecast line must be:
(timestamp: value_1, value_2, value_3, ...)
- Use the SAME variable order as specified above.
- Output exactly xxx lines.
- Do NOT include any extra text, explanations, or comments.
```

task: Predict the next 20 time points of all variables.

A.10 Dataset Licenses and Terms of Use

Table 4 lists the upstream license or access terms governing each of the nine MMTA source datasets, together with the access URL used in this work and the use restrictions that apply to MMTA's reuse. The benchmark is built strictly within the bounds of these terms: MIMIC remains behind PhysioNet credentialed access, no raw data from CoVLA, AgiBotWorld, CUHK-X, or PixelRec is redistributed (only construction scripts that operate on the user's own copy of the upstream release), and the released MMTA splits inherit the most restrictive applicable term per source. Where an upstream source carries a share-alike clause (Future Factories, AgiBotWorld), the corresponding subset of

Table 4: Upstream license / access terms for the nine MMTA source datasets, as of 2026-05-07. The "Use restriction" column summarizes the most stringent term that applies to MMTA's reuse; readers should consult the original license text before any further redistribution or commercial use.

Dataset	Source	License / access	Use restriction
AgiBotWorld [8]	Hugging Face: `agibot-world/AgiBotWorld2026`	CC BY-NC-SA 4.0	Non-commercial only; attribution and share-alike required.
CoVLA [23]	Hugging Face (gated): `turing-motors/CoVLA-Dataset`	Custom CoVLA terms; text descriptions also under VideoLLaMA 2 license	Academic / non-commercial only; non-transferable, non-redistributable; confidentiality required.
CUHK-X [24]	GitHub: `openaiotlab/CUHK-X`	Code MIT; data under DUA (non-commercial research, no redistribution); derived annotations CC BY 4.0	Raw data is *not* MIT; follow the DUA and do not redistribute the source data.
Future Factories [25]	Kaggle: `ramyharik/ff-2024-08-13-multi-modal-dataset-33`	CC BY-SA 4.0	Attribution and share-alike required for derived material.
HoloAssist [26]	`https://holoassist.github.io/`	CDLA-Permissive 2.0	Permissive data license; ship the CDLA Permissive 2.0 text alongside any redistribution.
MIMIC (IV / CXR-JPG / IV-Note) [27–29]	PhysioNet	PhysioNet Credentialed Health Data License 1.5.0	Credentialed access only; no re-identification; access not transferable; HIPAA training and code-sharing obligations apply.
PixelRec [30]	GitHub: `westlake-repl/PixelRec`	Code MIT; data under custom non-commercial research and education license	Data is *not* MIT; no commercial use; no rights to copy, modify, publish, or redistribute the dataset itself.
FinMultiTime [31]	Hugging Face: `Wenyan0110/Multimodal-Dataset-...`	MIT (per dataset metadata)	Preserve copyright and license notice on substantial redistribution.
Terra [32]	GitHub: `CityMind-Lab/NeurIPS24-Terra`	Apache-2.0 (project); upstream raster sources may carry additional terms	Verify the terms of any upstream raster source separately before redistribution.

MMTA is released under the same license; the TimeOmni-v code itself, which is independent of the data, is released under a permissive license in the public repository.

B Modality Coverage and Data Quality

This appendix supports Section 3.3. It states which domains carry sample-level text, how the labels are derived, how the two self-constructed components were audited, how video and time series were checked for alignment, and how the domains whose modalities share a source should be read.

B.1 Sample-Level Text

Text is counted as a sample-level modality only when it varies across samples and carries scene, event, or background information about the sample. Fixed task instructions, field definitions, and label lists define the task and are not counted. A field-level check of the training data shows that seven domains carry such text in every sample (Table 5). In CoVLA and CUHK-X the prompt contains only the fixed task description, the variable or sensor definitions, and the label space, so these two domains are video+TS tasks with respect to sample-level evidence.

B.2 Label Derivation

Only the CoVLA brake-reason label is generated by our pipeline; the other three relabeled domains keep the upstream label semantics through deterministic, traceable steps.

Table 5: Sample-level text in each MMTA domain.

Domain	Sample-level text	Per-sample text content
AgiBotWorld	yes	Upstream task instruction and the time-ordered execution steps inside the Task Frame
CoVLA	no	Fixed driving-task description, variable definitions, question, and three-way label space
CUHK-X	no	Fixed HAR task description, sensor definitions, and 40-way action space
Future Factories	yes	Production-cycle report (cycle ID, UTC time range, record count, process, signal meaning, camera notes)
HoloAssist	yes	Coarse-grained action description and the time-ordered fine-grained actions in the segment
MIMIC	yes	Radiology report of the imaging study (findings, and possibly clinical history and prior-exam comparisons)
PixelRec	yes	Item title, tags, and description
FinMultiTime	yes	News text about the asset
Terra	yes	Geographic and climatic description generated from the grid-cell metadata

AgiBotWorld. The upstream `Error Frame` annotation provides an `error_cause`. Error Frames whose temporal midpoint falls inside a Task Frame are collected, and the raw descriptions are normalized to *paused*, *meaningless operation*, *fall*, or *other*. Task Frames with more than one normalized anomaly type are excluded rather than collapsed into one label.

Future Factories. The upstream cycle-management records give the missing-part condition that was set by hand in each experiment. Records are joined to production cycles by cycle ID and UTC time range, and the upstream descriptions are normalized to the four labels. The label is therefore a mapping of the experimental condition, not an inference from the input time series.

HoloAssist. The upstream annotation gives an `Action Correctness` value for every fine-grained action. Fine-grained actions are assigned to coarse-grained segments by temporal containment; a segment is labeled A if all its actions are `Correct Action` and B otherwise.

CoVLA. Braking intervals are located from the upstream `brakePressed` signal (at least six consecutive frames). Inside an interval, label A is assigned if `lead_a`< 0 holds for at least two consecutive frames, label B if a left or right turn signal is on for at least two consecutive frames, and label C otherwise; intervals that trigger both A and B are excluded.

B.3 Human Audit of the Constructed Components

The CoVLA rule-derived labels and the Terra LLM-generated descriptions were audited because they are the two components that need semantic verification. For each, 100 samples were drawn at random and checked independently by two annotators; every disagreement was resolved by adjudication, and the adjudicated decision was used to judge the generated content. The two annotators are doctoral students with at least one year of research experience in time-series processing, forecasting, or multimodal time-series analysis. Table 6 reports the results. The identified items are scheduled for correction in the next dataset release; no sample was deleted, because the underlying video, images, and time series remain usable and aligned.

Table 6: Double-annotated audit of the two self-constructed components. Agreement and Cohen's κ are computed before adjudication; accuracy is computed after adjudication, with a 95% Wilson interval.

Component	N	Agreement	κ	Adjudicated	Accuracy (95% CI)	Items to correct
CoVLA brake-reason labels	100	94%	0.910	6	96% (90.2–98.4%)	4
Terra generated text	100	95%	0.687	5	93% (86.3–96.7%)	7

Error types. The CoVLA errors concern fine-grained driving semantics: at road crossings with a lead vehicle present, the rule can attribute the braking to lead-vehicle deceleration, and some lane changes are labeled as turns. The Terra errors concentrate on ocean cells, where the text can contain internal geographic contradictions or read a negative elevation (sea-floor depth) as a land altitude.

B.4 Cross-Modal Alignment Audit

The mechanical audit checks source-sample identity, timestamps or frame boundaries, time-series coverage, and file existence for all 25,057 video+TS samples. Because clips are cut with keyframe-level stream copy, an exact frame-count match cannot separate a lost target event from a few extra context frames, so two rates are reported. The *coverage failure rate* is the fraction of samples whose released clip does not fully contain the target annotation window. The *boundary overhang rate* is the fraction of released frames that lie outside the target window.

Table 7: Mechanical alignment audit of the video+TS domains.

Domain	Samples	Coverage failure	Boundary overhang	Max. overhang
AgiBotWorld	1,308	0 (0.000%)	2.35%	66 frames / 2.20 s
CoVLA	14,781	0 (0.000%)	4.80%	11 frames / 0.55 s
CUHK-X	2,937	0 (0.000%)	0.00%	0
Future Factories	95	0 (0.000%)	0.00%	0
HoloAssist	5,936	4 (0.067%)	0.55%	13 frames / ≈0.5 s
Total	25,057	4 (0.016%)	1.42%	—

The four coverage failures are HoloAssist samples whose annotated end lies 3–4 frames past the end of the source video, so no existing content was lost in cutting. The video+TS data are therefore aligned at the level of the annotated segment, with a small amount of boundary context set by the video encoding. The image+TS domains are joined on deterministic keys: patient, imaging study, and admission time for MIMIC; item ID for PixelRec; asset and date for FinMultiTime; and grid cell and date for Terra.

B.5 CoVLA Labels and the Input Time Series

The CoVLA label rule and the released time series overlap only partially. The variables that locate the braking interval (`brakePressed`) and that define class B (the turn signals) are not among the five released channels (`vEgo`, `aEgo`, `steeringAngleDeg`, `lead_speed_kmh`, `lead_a`); only `lead_a` enters the class-A rule. The rule fires on a local trigger of two consecutive frames, whereas the released series covers the whole braking interval of 6–600 steps. A local deceleration also does not separate road crossings, traffic signals, and lead-vehicle behavior, which the video resolves. Consistent with this, the TS-only variant of TimeOmni-v trails the full model by 16.4% on CoVLA (Figure 3).

B.6 Domains with Shared Sources

Non-redundancy is read here as a property of the information each modality carries, not of how the modalities were acquired. Three domains combine modalities that share a source.

FinMultiTime. The candlestick chart and the price series come from the same market data. The released series carries only the close price, whereas the chart also encodes the open, high, and low prices and thus shows range, gaps, and volatility; the two share the close price. The news text adds company events that the price series does not contain.

Terra. The text is generated from the grid-cell metadata (coordinates, country or city, land cover, climate type, elevation) and discusses terrain, ocean influence, monsoons, prevailing winds, vegetation, and human activity. These static descriptors are not encoded in the seven forecast channels, and the images add satellite appearance and geophysical fields. The three modalities are linked through the same grid cell but do not substitute for one another.

MIMIC. The chest X-ray and the radiology report come from the same imaging study. The image keeps the raw morphology and spatial detail, while the report records the radiologist's interpretation and may add clinical history and comparisons with prior exams, which cannot be read from the pixels. The vital signs of the first 48 hours form a separate physiological stream, and the target is a later hospital outcome. The training-time ablation shows that removing the vital signs does not lower Macro-F1 on MIMIC (Figure 3), so MIMIC is a boundary case of complementarity rather than strong evidence of it.

C TimeOmni-v Training Details

This appendix documents the training recipe used to fine-tune **TimeOmni-v** (Section 4) on each MMTA dataset. A single recipe is shared across the nine MMTA datasets, with only two hyperparameters varying per dataset. All experiments are run from the same base model and the same set of trainable modules; the resulting checkpoints are evaluated under the protocol of Section 5.1.

C.1 Initialization and trainable modules

TimeOmni-v is initialized from two pretrained backbones. The Omni-LLM half is initialized from Qwen2.5-Omni-7B [21] (Thinker half), and the time-series encoder branch is initialized from Chronos-2 [33]. The vision and audio towers of the Thinker are kept frozen, since the visual evidence in MMTA is already covered by the pretrained vision encoder and the audio modality is not used. Four sets of parameters are fine-tuned in full precision: (i) the Chronos-2 encoder; (ii) the lightweight TS-to-LLM adapter that maps Chronos-2 patch embeddings to the LLM hidden size; (iii) the embedding rows of the three new TS special tokens (`<|ts_start|>`, `<|ts_placeholder|>`, `<|ts_end|>`); and (iv) the forecast head, attached only on forecasting datasets. The Omni-LLM itself is fine-tuned with LoRA [36] of rank 8, $\alpha = 32$, and dropout 0.05, applied to all attention projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and dense MLP projections (`gate_proj`, `up_proj`, `down_proj`) of every transformer block.

For each MMTA dataset, an independent TimeOmni-v adapter set is trained on top of the same base model. An adapter set comprises the LoRA weights, the Chronos-2 encoder, the TS-to-LLM adapter, the new TS-token embedding rows, and a forecast head where applicable.

C.2 Optimization

The model is optimized with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, weight decay 0). A linear warmup is applied for the first 10% of the training steps, followed by a cosine learning-rate decay schedule. Gradients are clipped at max-norm 1.0. Training is run in bf16 with gradient checkpointing enabled, on top of DeepSpeed ZeRO stage 2.

To keep per-step token counts balanced across DDP ranks under variable sequence length, batches are formed by a token-budget dynamic sampler that greedily packs each rank's batch up to $\sim 40\,000$ padded tokens, with a hard cap of 32 samples per rank-batch. The gradient-accumulation is 1. All experiments are run on NVIDIA H200 140 GB GPUs, with the GPU count per run scaled to the size of the dataset (1–8 GPUs per dataset).

C.3 Per-dataset hyperparameters

Only the learning rate and the number of training epochs vary across MMTA datasets; all other hyperparameters follow Sections C.1 and C.2. Table 8 summarizes the per-dataset settings.

Learning rate. A single rule is used to pick the per-dataset learning rate. The learning rate is initialized at 1×10^{-5}. A short pilot run is then launched, and the training loss is monitored. If the loss diverges to NaN within the first epoch, the learning rate is halved and the pilot is restarted; this halve-and-retry is repeated until the loss stays finite throughout the pilot. The smallest stable rate found this way is then used for the full training run. Forecasting variants tend to settle at a smaller rate than classification ones, since the freshly initialized forecast head amplifies the early gradient signal flowing back into the LoRA-side parameters.

Table 8: Per-dataset TimeOmni-v training hyperparameters. All other settings (initialization, frozen/-trainable modules, optimizer, schedule shape, batching, hardware) are shared across datasets and described in Sections C.1 and C.2.

Dataset	Modalities	Task	Learning rate	Epochs
`covla`	video + TS	classification	1×10^{-5}	10
`agibot`	video + TS	classification	1×10^{-5}	20
`cuhk_x_har`	video + TS	classification	1×10^{-5}	10
`holo_assist`	video + TS	classification	2×10^{-6}	10
`future_factories`	video + TS	classification	2×10^{-6}	100
`mimic_death`	image + TS	classification	2×10^{-6}	30
`mimic_disch`	image + TS	classification	2×10^{-6}	30
`terra_perch`	image + TS	forecasting	5×10^{-6}	5
`sp500_perch`	image + TS	forecasting	1×10^{-5}	5
`pixelrec_perch`	image + TS	forecasting	2×10^{-6}	30

Number of epochs. The epoch count is set to keep each dataset's optimizer-step budget comparable. Concretely, the number of epochs is chosen so that the total number of optimizer steps per dataset stays above 400, and is otherwise increased only as needed to compensate for very small training sets. As a result, datasets with large training sets (`covla`, `holo_assist`) use short schedules of 10 epochs, while datasets with small training sets (`future_factories`, `mimic_death`, `mimic_disch`, `pixelrec_perch`) extend to 30–100 epochs to clear the same step floor.

D Additional Results

D.1 Decomposed Metrics and Failure Taxonomy

Tables 9–11 decompose every cell of Table 1 into the base metric on the parsed subset, the success rate (SR), and the causes of failure. Context overflow appears as CUDA OOM, because an over-length prompt aborts in the first forward pass; semantic errors are the complement of the base metric on the parsed subset. The two MIMIC subtasks are reported separately.

Table 9: Classification: base Macro-F1 on the parsed subset, with SR (%) in parentheses. "—": no sample was parsable. Best base Macro-F1 per column in **bold**.

Model	AgiBot (131)	CoVLA (1,344)	CUHK-X (308)	FuFa (46)	HoloA (945)	MIMIC-death (75)	MIMIC-disch (75)
InternVL3.5-4B	0.070 (28.2)	0.495 (100.0)	0.218 (100.0)	0.049 (100.0)	0.514 (49.3)	0.547 (100.0)	0.508 (100.0)
InternVL3.5-8B	0.138 (19.1)	0.515 (99.8)	0.260 (94.5)	0.050 (97.8)	0.483 (43.7)	0.558 (100.0)	0.571 (100.0)
Qwen3-VL-8B-Instruct	0.139 (77.9)	0.592 (100.0)	0.084 (100.0)	0.049 (100.0)	0.273 (94.7)	**0.769** (100.0)	0.542 (100.0)
Qwen3-VL-30B-A3B-Inst.	— (0.0)	0.577 (100.0)	0.086 (99.4)	0.129 (100.0)	— (0.0)	0.638 (100.0)	0.532 (100.0)
Qwen2.5-Omni-7B	0.108 (56.5)	0.424 (100.0)	0.099 (100.0)	0.198 (100.0)	0.303 (74.6)	0.559 (100.0)	0.551 (100.0)
Qwen3-Omni-30B-A3B-Inst.	0.125 (28.2)	0.491 (100.0)	0.244 (100.0)	0.129 (100.0)	0.331 (63.5)	0.741 (100.0)	0.544 (100.0)
GPT-5-mini	0.118 (98.5)	0.586 (100.0)	0.253 (99.0)	0.093 (100.0)	0.569 (100.0)	0.638 (100.0)	0.567 (100.0)
Gemini-3-flash	0.126 (100.0)	0.560 (100.0)	**0.463** (100.0)	0.315 (100.0)	0.561 (100.0)	0.693 (100.0)	0.563 (100.0)
TimeOmni-v	**0.844** (100.0)	**0.759** (100.0)	0.343 (100.0)	**0.362** (100.0)	**0.694** (100.0)	0.606 (100.0)	**0.598** (100.0)

Table 10: Forecasting: base MAPE on the parsed subset, with SR (%) in parentheses. Best per column in **bold**; on SP500, Gemini-3-flash (4.4503) is marginally below TimeOmni-v (4.4512).

Model	PixelRec (200)	SP500 (200)	Terra (200)
InternVL3.5-4B	150.0 (100.0)	4.76 (100.0)	296.1 (93.0)
InternVL3.5-8B	98.0 (100.0)	5.44 (100.0)	156.1 (100.0)
Qwen3-VL-8B-Instruct	145.5 (100.0)	5.49 (100.0)	77.4 (100.0)
Qwen3-VL-30B-A3B-Inst.	129.2 (100.0)	4.75 (100.0)	75.2 (99.5)
Qwen2.5-Omni-7B	104.4 (100.0)	7.73 (100.0)	176.9 (99.5)
Qwen3-Omni-30B-A3B-Inst.	98.0 (100.0)	6.72 (100.0)	**65.4** (100.0)
GPT-5-mini	91.3 (100.0)	4.84 (98.0)	100.4 (96.5)
Gemini-3-flash	**90.8** (99.0)	**4.45** (100.0)	95.6 (99.5)
TimeOmni-v	108.4 (100.0)	4.45 (100.0)	159.6 (100.0)

Table 11: Failure taxonomy: % CUDA OOM / % other inference failures (e.g., API refusals) / % unparsable outputs, over all samples. In each cell, SR $= 100-$ the sum of the three.

Model	AgiBot	CoVLA	CUHK-X	FuFa	HoloA	MIMIC-death	MIMIC-disch	PixelRec	SP500	Terra
InternVL3.5-4B	0.8/0/71.0	0/0/0	0/0/0	0/0/0	0/0/50.7	0/0/0	0/0/0	0/0/0	0/0/0	0/0/7.0
InternVL3.5-8B	1.5/0/79.4	0/0/0.2	0/0/5.5	0/0/2.2	0/0/56.3	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
Qwen3-VL-8B-Instruct	22.1/0/0	0/0/0	0/0/0	0/0/0	0/0/5.3	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
Qwen3-VL-30B-A3B-Inst.	100.0/0/0	0/0/0	0/0/0.6	0/0/0	100.0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0.5
Qwen2.5-Omni-7B	11.5/0/32.1	0/0/0	0/0/0	0/0/0	4.0/0/21.4	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0.5
Qwen3-Omni-30B-A3B-Inst.	42.7/0/29.0	0/0/0	0/0/0	0/0/0	21.8/0/14.7	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
GPT-5-mini	0/1.5/0	0/0/0	0/0.3/0.6	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/2.0	0/0/3.5
Gemini-3-flash	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/1.0	0/0/0	0/0/0.5
TimeOmni-v	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0

The taxonomy shows three failure patterns. The open-source Qwen models run out of memory on the long-TS domains (on AgiBot, 100% for Qwen3-VL-30B-A3B-Instruct, 42.7% for Qwen3-Omni-30B-A3B-Instruct, 22.1% for Qwen3-VL-8B-Instruct, and 11.5% for Qwen2.5-Omni-7B; on HoloAssist, up to 100%). The InternVL3.5 models stop following the output format on the same domains (71–79% unparsable on AgiBot, 51–56% on HoloAssist). The proprietary models rarely fail, but their base accuracy on AgiBot stays low (0.118 and 0.126 Macro-F1). TimeOmni-v has no failure on any test set.

D.2 Paired Bootstrap Tests

TimeOmni-v is compared with the strongest baseline of each dataset in Table 1 under a paired sample-level bootstrap with 1,000 resamples. Δ is the swF1 difference for classification (positive favors TimeOmni-v) and the swMAPE difference for forecasting (negative favors TimeOmni-v); p is the one-sided resampling probability that TimeOmni-v is not better, so $p < 0.05$ means TimeOmni-v leads significantly and $p > 0.95$ means the baseline leads significantly. For MIMIC, Δ is the mean over the two subtasks, which are resampled separately.

Table 12: Paired bootstrap comparison of TimeOmni-v with the strongest baseline per dataset, with two-sided 95% intervals for Δ.

	CoVLA	HoloA	CUHK-X	MIMIC	AgiBot	FuFa	PixelRec	SP500	Terra
Baseline	Qwen3-VL-8B	GPT-5-mini	Gemini-3-flash	Qwen3-VL-8B	Gemini-3-flash	Gemini-3-flash	GPT-5-mini	Gemini-3-flash	Qwen3-Omni-30B
Test n	1,344	945	308	150	131	46	200	200	200
Δ	+0.167	+0.125	−0.120	−0.053	+0.717	+0.047	+17.1	+0.00	+93.8
95% CI	[+0.134, +0.199]	[+0.084, +0.168]	[−0.180, −0.042]	[−0.157, +0.049]	[+0.630, +0.795]	[−0.101, +0.207]	[+3.8, +31.0]	[−0.76, +0.84]	[+41.6, +169.2]
p	<0.001	<0.001	0.999	0.844	<0.001	0.246	0.995	0.522	>0.999

TimeOmni-v leads significantly on CoVLA, HoloAssist, and AgiBot, the latter already at $n = 131$. The strongest baseline leads significantly on CUHK-X, PixelRec, and Terra. On SP500 the narrow interval around zero marks a tie. Future Factories ($n = 46$) and MIMIC ($n = 150$) remain unresolved, with intervals that span zero, so no ranking conclusion is drawn on them.

D.3 Inference-Time Modality Removal

Figure 5 removes one modality at test time from models whose training or evaluation otherwise uses all modalities. Because TimeOmni-v is trained with every modality present, a drop here measures how much the trained model relies on the removed stream, not whether the task needs it; Figure 3 answers the second question.

For TimeOmni-v, removing vision lowers Avg-swF1 from 0.601 to 0.349 and removing time series lowers it to 0.406. The three domains outside the training-time ablation show distinct reliance patterns. CUHK-X relies on vision (0.343 to 0.006 without vision, 0.327 without time series), Future Factories relies on time series (0.362 to 0.049 without time series), and HoloAssist changes little under either removal (0.650 without vision, 0.654 without time series). On MIMIC, removing time series at test time raises swF1 from 0.602 to 0.615, in line with the training-time result that the vital signs add little there. Qwen2.5-Omni-7B, which is evaluated zero-shot, often scores higher after a removal (for example, 0.257 to 0.481 on HoloAssist without time series), which indicates that it does not use the removed stream effectively.

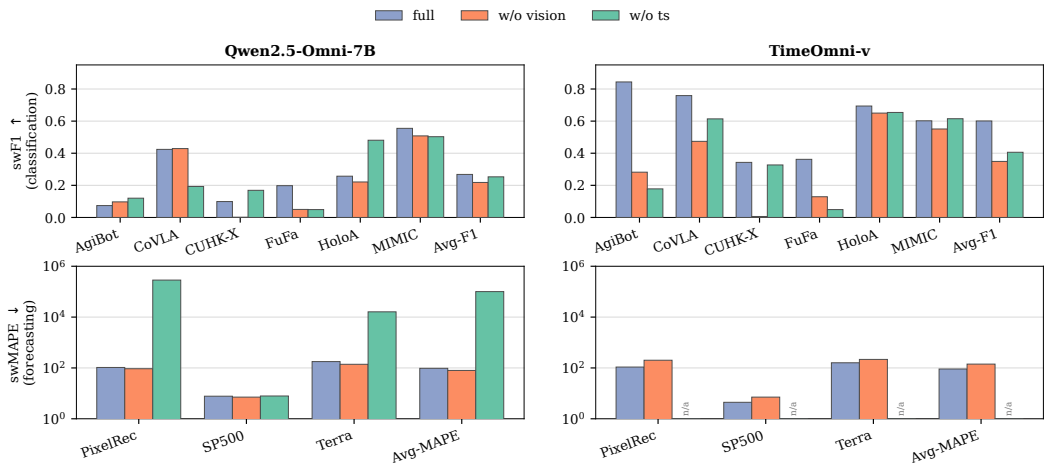


Figure 5: Inference-time modality removal under three input conditions: *full* (all modalities provided), *w/o vision* (image/video stream removed), and *w/o ts* (time-series stream removed). Top row: classification swF1 (↑); bottom row: forecasting swMAPE (↓, log axis); each column is one model. "n/a" marks undefined entries (the TimeOmni-v forecast head requires TS input). Avg-F1 and Avg-MAPE aggregate only over defined entries.

E Broader Impacts

This appendix lists the positive and negative societal impacts that follow from the released artifacts (the MMTA benchmark and the TimeOmni-v codebase). The discussion is scoped to the use cases that MMTA's nine domains directly support; impacts that apply to MLLMs in general but are not specific to this work are not repeated.

Positive impacts. MMTA targets domains where decisions are made under joint visual, textual, and temporal evidence – healthcare, autonomous driving, manufacturing, AR assistance, and climate. A public tri-modal benchmark surfaces failure modes (notably the long-sequence textualized-TS failure mode named in Section 6.1) before such systems are placed near deployment, and supports apples-to-apples comparison across model families. Beyond evaluation, TimeOmni-v' compression of the time-series modality lowers the activation-memory and inference-latency barrier to training tri-modal models on commodity GPUs (Section 6.2), which makes the modality-aware recipe more accessible to academic and clinical research groups whose hardware budget cannot match those of large industry labs.

Negative impacts and mitigations. Three negative impacts are visible. First, the surveillance-style streams in CUHK-X (skeleton-rendered human action), Future Factories (assembly-station cameras), and HoloAssist (egocentric AR video) could in principle be repurposed for unconstrained behaviour monitoring; this risk is bounded by the upstream releases, which already published these streams under their own access policies, and MMTA does not republish raw video beyond what those sources already make available. Second, the MIMIC-derived prognosis tasks (mortality and discharge readiness) inherit demographic biases from the underlying ICU population and are not validated for clinical decision-making; the credentialed-access requirement of PhysioNet remains the primary control, and the benchmark is positioned as a model-evaluation instrument rather than a screening tool. Third, the FinMultiTime-derived forecasting task uses delayed news snippets and historical S&P 500 closes; the headline numbers are not validated for live trading and should not be read as production-ready signals. Across all three, no fine-tuned model checkpoints from this work are released for sensitive domains (MIMIC, FinMultiTime); only the benchmark splits and the training code are public, which keeps deployment a deliberate downstream choice rather than a default.

F Limitations

On the benchmark side, MMTA is a collection of tri-modal tasks with diverse cross-modal complementarity patterns, not a benchmark in which every domain strictly satisfies tri-modal non-redundant evidence: CoVLA and CUHK-X carry no sample-level text, FinMultiTime, Terra, and MIMIC combine modalities that share a source (Appendix B), and modality necessity is measured by training-time ablation on six of the nine domains (Figure 3). Human validation covers the two self-constructed components, and no per-sample check was run that a sample needs at least two modalities, because a reliable judgment requires domain experts such as clinicians and driving experts. Several test splits are small, and the paired bootstrap leaves the comparisons on Future Factories (46 samples) and MIMIC (150 samples) unresolved.

On the evaluation side, the baselines in Table 1 are evaluated zero-shot, whereas TimeOmni-v is fine-tuned per dataset. The adapted TS-as-text baseline removes this confound only on short-series domains, because TS-as-text fine-tuning runs out of memory on AgiBotWorld and HoloAssist and truncated textual baselines were not trained; on forecasting, the comparison also changes the output from digit decoding to a forecast head. The success-weighted metrics mix semantic errors with parsing and hardware failures, which Appendix D.1 separates. The time-series integration has been ported to one other backbone on four datasets, and each domain is trained and evaluated separately, so transfer across backbones and across domains remains untested.