ViSTA: A Simple Bridge Extends Visual Alignment to Clinical Time-Series Understanding in Multimodal LLMs

Anonymous authors
Paper under double-blind review

Abstract

Clinical prediction models estimate risk from patient measurements, while large language models support medical text understanding and question answering. Yet their language capabilities do not ensure accurate prediction from structured, high-dimensional clinical time series. Improving this ability would connect risk estimation with flexible questions about a patient's evolving condition. We introduce ViSTA, a compact adapter that incorporates irregular numerical measurements into a pretrained vision-language model's chart representations. It learns corrections to visual tokens while leaving all pretrained parameters unchanged. On MIMIC-IV, ViSTA has the highest mean scores among the compared adaptations on all four metrics for acute kidney injury and mortality prediction across models with 2-9 billion parameters. With 0.516 million trainable parameters, the 2-billion-parameter model reaches an area under the ROC curve of 0.7376 for acute kidney injury, compared with GPT-5.6 Sol's 0.7380 with text input and high reasoning effort. Training for temporal question answering yields 69.27% accuracy at 4 billion parameters with over 90% fewer trainable parameters than low-rank adaptation using charts or numerical text, at a 2.82-4.88 percentage-point accuracy gap. ViSTA extends pretrained language models to numerical prediction and temporal questions.

1 Introduction

Clinical artificial intelligence supports outcome prediction and patient-record interpretation. Statistical and deep learning models use patient measurements to estimate mortality risk, anticipate deterioration, and predict length of stay (Harutyunyan et al., 2019; Gao et al., 2020; Ma et al., 2020). Large language models (LLMs) support clinical summarization and medical question answering (Van Veen et al., 2024; Singhal et al., 2023), offering flexible queries that complement predefined prediction targets.

Extending this flexibility to numerical records remains difficult. Vital signs, laboratory tests, and interventions form high-dimensional time series with irregular sampling, missing values, and dependencies across variables. Their numerical and temporal relationships differ from information in clinical narratives. Recent work finds that even state-of-the-art closed-source LLMs remain behind conventional models on non-generative clinical prediction from structured records, with larger gaps for smaller open-source LLMs (Zhu et al., 2026a). Improving this capability would let a language-based system use the same measurements to estimate risk and answer questions about threshold crossings or changes across trajectories, which conventional prediction models cannot do (Zhu et al., 2026b). This motivates adaptation that strengthens numerical prediction while retaining a language interface for patient-record questions.

Existing approaches differ in how they present numerical records and which components they adapt. Text serialization exposes exact measurements but spreads temporal relationships across long sequences of names, timestamps, and values. Learned numerical encoders provide compact representations, but these must be aligned with the language model. Charts offer an established visual-language pathway, yet access to precise measurements depends on what survives rendering and visual encoding (Zhao et al., 2025; Liu et al., 2025). Fine-tuning and low-rank adaptation (LoRA)

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

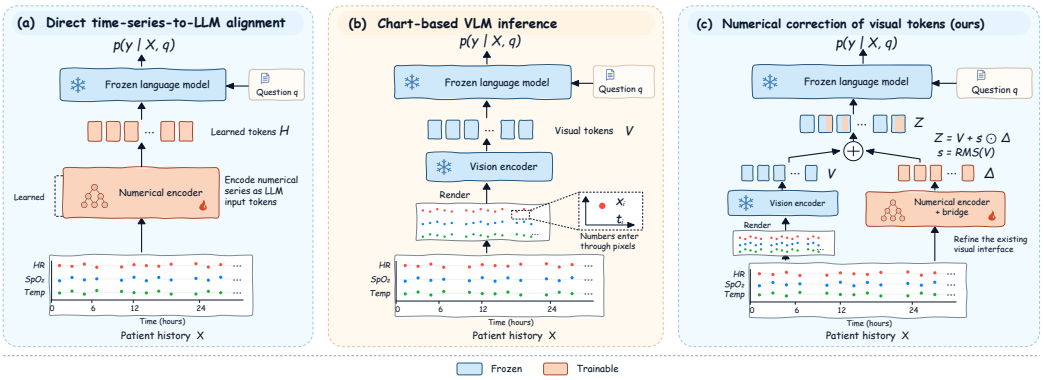

Figure 1: **Numerical correction of an established visual interface.** Direct numerical interfaces learn how to present a new continuous representation to a language model. Chart inputs use its pretrained visual pathway. ViSTA adds numerical corrections to original visual tokens at their existing positions, preserving token order and positional indices. The language model and its output head remain frozen.

require adjustment to language-model weights (Hu et al., 2021; Kalajdzievski, 2024). These approaches leave an opportunity to combine direct numerical access with the visual-language alignment already learned during pretraining. Can a small adapter enrich the model's existing chart representations with precise measurements while keeping its pretrained components and language output interface unchanged?

Our design starts from chart tokens that already pass through the pretrained visual-language interface. We add corrections derived from the underlying measurements at those same positions, learning from an established input pathway. Figure 1 summarizes the three input paradigms.

We introduce ViSTA (**Vi**sually aligned **S**eries-to-language **T**emporal **A**dapter), comprising an irregular-series encoder and a residual bridge. The encoder summarizes values, times, observation gaps, and variable identity; queries at visual positions read these summaries to correct visual tokens. Zero initialization recovers the original chart model. We train separate adapters for outcome prediction and temporal question answering (QA), and also evaluate QA after training only on outcomes.

Our contributions are threefold. First, we introduce a numerical adapter with fewer than one million trainable parameters that enriches the visual representations of a fully frozen vision-language model. Second, at 4 billion parameters, ViSTA reaches average precision of 0.4650 for acute kidney injury and 0.2651 for mortality, representing relative AP improvements of 3.15% and 7.59% over the strongest competing adaptation for each outcome, and 66.37% and 77.56% over the unadapted chart model, respectively. The 2-billion-parameter model's AKI AUROC is 0.7376, compared with GPT-5.6 Sol's 0.7380. Third, outcome-only training yields 41.73% temporal QA accuracy at 4 billion parameters, 2.33 percentage points above native text (39.40%). QA training yields 69.27%, 29.87 percentage points above the same reference and 2.82-4.88 percentage points below the LoRA baselines, with more than 90% fewer trainable parameters. We further observe consistent improvements in clinical prediction as the pretrained backbone scales from 2 to 9 billion parameters.

2 Related Work

Visual and multimodal time-series analysis. Charts expose temporal patterns through pretrained visual features, but recovering individual measurements depends on the rendering and visual representation (Zhao et al., 2025). CaTS-Bench finds that even strong vision-language models struggle with numerical nuances in time-series descriptions (Zhou et al., 2025). MLLM4TS aligns temporal patches with plot features for prediction (Liu et al., 2025), but its task-specific output heads do not directly support a shared language interface for prediction and QA. ViSTA supplies numerical evidence through residuals at the original visual positions, allowing both tasks to use the unchanged language decoder and output head.

Connecting time series to language models. Time-LLM reprograms numerical patches using text prototypes and projects language-model representations into forecasts (Jin et al., 2023). ChatTS learns continuous representations using synthetic descriptions and questions; Time-MQA trains on contextualized numerical sequences for temporal QA (Xie et al., 2024; Kong et al., 2025). ITFormer connects temporal encoders to frozen language models through instruction-aware interaction, while OpenTSLM explores soft prompts and gated cross-attention for medical signals (Wang et al., 2025a; Langer et al., 2025). Text requires sequences of values and timestamps; learned representations require alignment with the language model, even when its decoder is frozen. ViSTA starts from the existing chart representation, using a zero-initialized residual to learn how numerical evidence should modify this established input.

Parameter-efficient adaptation and clinical evaluation. Representation adaptation has supported knowledge transfer in biomedical prediction (Shi et al., 2026). LoRA adapts pretrained weights through low-rank updates (Hu et al., 2021); BLIP-2 and Flamingo learn interfaces between pretrained components (Li et al., 2023; Alayrac et al., 2022). Parameter reduction alone does not determine access to numerical evidence. We study where to adapt: a small numerical module augments an existing visual interface while all pretrained components remain frozen. Clinical prediction remains challenging for general-purpose language models (Zhu et al., 2026a), and temporal QA requires more than recognizing global trends (Nie et al., 2026). We assess both tasks on the same backbone, including QA after outcome-only training, to examine task specialization and behavior beyond the training objective.

3 ViSTA: Adding Numerical Information to Visual Tokens

Our design idea is simple and intuitive: charts provide an established visual input to a pretrained model, and their underlying measurements supply direct numerical evidence. ViSTA learns how this evidence should modify the existing visual tokens (Figure 2).

3.1 Problem formulation and the pretrained visual pathway

A patient history is a collection of irregular observations $X = \{(t_i, c_i, x_i)\}_{i=1}^{M}$, where t_i is the measurement time, $c_i \in \{1, \dots, C\}$ identifies a clinical variable, and x_i is its value. Each question specifies a visible interval and, for prospective tasks, an information-availability cutoff. We denote visible intervention context by a and the question by q. The model represents $p(y \mid X, a, q)$ through its original language output head.

We render each variable as a separate chart image, with visible timestamps and values, and use the vision encoder and projector of the pretrained multi-modal LLM model, such as Qwen3.5, to obtain

$$V = f_{\text{vis}}(\text{Render}(X)) \in \mathbb{R}^{N \times D}. \tag{1}$$

Here N is the visual token count and D the language model's hidden dimension. Retaining chart order, image delimiters, and positional indices lets numerical adaptation operate within the decoder's expected input structure.

3.2 Encoding irregular numerical observations

Encoding individual events exposes both measurements and observation gaps. For observation i, $\psi_i \in \mathbb{R}^{d_{\text{in}}}$ contains its normalized value, relative time within the visible interval, time since the preceding measurement of that variable, and a first-observation indicator. A multilayer perceptron (MLP) and variable embedding produce

$$e_i = f_\theta(\psi_i) + b_{c_i} \in \mathbb{R}^d. \tag{2}$$

Unobserved variables receive a learned missing-observation token plus their variable embedding, explicitly representing absence. Stacking all tokens gives memory $E \in \mathbb{R}^{M_e \times d}$.

To summarize variable-length histories into a fixed-size memory, learned queries $U \in \mathbb{R}^{K \times d}$, indexed by variable, pool the events:

$$R = \text{LN}(U + \text{MHA}(U, E, E)), \qquad S = R + \text{FFN}(R). \tag{3}$$

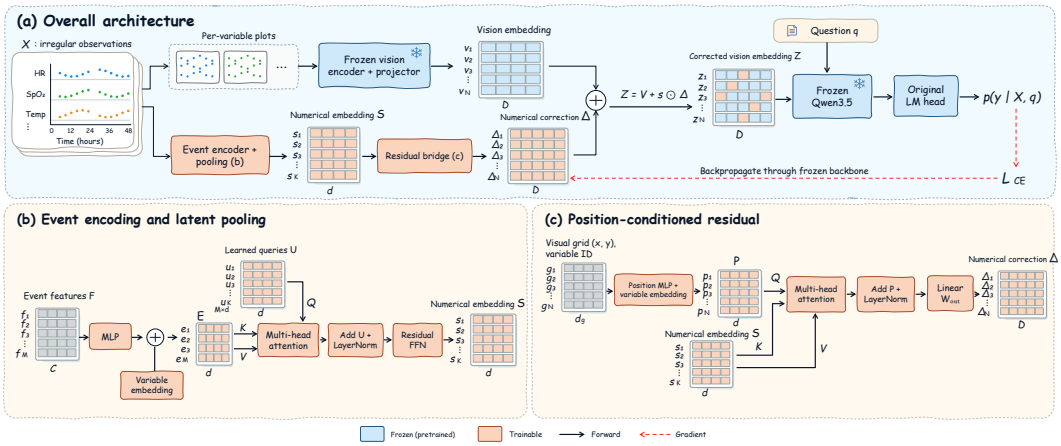

Figure 2: **ViSTA architecture.** An event encoder pools irregular numerical observations into latent summaries. Queries derived from visual grid positions and variable identity read these summaries and produce residuals with the same shape as the original visual tokens. The residual output projection starts at zero. Answer supervision propagates through the frozen language model to the event encoder and bridge.

MHA denotes multi-head attention, LN layer normalization, and FFN a feedforward network. Each summary can draw on events across time. **Joint** encoding lets queries attend to all variables to model their relationships during pooling. **Separate** encoding restricts query groups to their own variable, keeping evidence separate until the bridge. Both share attention and feedforward weights, have identical parameter counts, and expose all K summaries to the bridge.

3.3 Correcting Original Visual Tokens

To tailor each correction to its chart location, we condition retrieval on token j's normalized grid coordinate r_j and chart identity $c(j)$, using $p_j = h_\theta(r_j) + b_{c(j)} + W_v v_j$. Stacking queries into $P \in \mathbb{R}^{N\times d}$, the bridge retrieves numerical evidence in the visual dimension:

$$\Delta = \mathrm{Linear}_{\mathrm{out}}[\mathrm{LN}(P + \mathrm{MHA}(P, S, S))] \in \mathbb{R}^{N\times D}. \tag{4}$$

Residual addition makes the visual feature the starting point for adaptation. We scale the correction relative to each token's magnitude:

$$z_j = v_j + s_j \Delta_j, \qquad s_j = \sqrt{D^{-1}\textstyle\sum_{k=1}^{D} v_{jk}^2}. \tag{5}$$

The scale s_j is the root mean square (RMS) of the frozen visual token. For nonzero v_j, $\|s_j\Delta_j\|_2/\|v_j\|_2 = \mathrm{RMS}(\Delta_j)$, so the adapter learns a relative correction magnitude. We initialize the output projection's weights and bias to zero, giving $Z = V$. The frozen decoder thus initially receives its native chart representation, and supervision subsequently determines the numerical correction.

Position-conditioned queries can retrieve across times and variables. The K summaries remain internal, so the decoder still receives N visual tokens. For fixed widths, interface attention scales with $M_eK + NK$.

3.4 Learning through the Original Language Interface

Expressing predictions and QA responses as language answers gives both tasks the same output interface. With pretrained parameters ϕ frozen, we optimize the event encoder and bridge parameters θ using answer-token cross-entropy:

$$\mathcal{L}(\theta) = -\sum_{\ell=1}^{|y|} \log p_\phi(y_\ell \mid y_{<\ell}, q, a, Z_\theta(X)). \tag{6}$$

The vision encoder, pretrained projector, language decoder, embeddings, and language head remain frozen. Gradients through the decoder teach the adapter which numerical corrections are useful for the requested answer.

Binary outcomes use the original vocabulary tokens for the two labels, with $p(y = 1) = \exp(\ell_1)/[\exp(\ell_0) + \exp(\ell_1)]$ from their logits. Multiple-choice QA normalizes over its answer labels in the same way. The question guides the decoder's use of the corrected record, allowing different questions about an unchanged visible interval to reuse its representation. The original output head also leaves autoregressive generation available; our experiments evaluate outcome prediction and temporal multiple-choice QA.

4 Experimental Setup

Tasks and data. We use MIMIC-IV v3.1 (Johnson et al., 2023; 2024), with eleven irregularly observed clinical variables and visible intervention records. Patient splits are disjoint. Observations from the first 24 hours in the intensive care unit (ICU) predict newly documented creatinine-defined acute kidney injury (AKI) during hours $(24, 48]$ and in-hospital mortality. These outcomes have 5,000 training examples for each task, with 500 validation and 1,000 test examples each. Temporal QA contains 22,000 training, 1,100 validation, and 2,200 test questions across eleven categories following CLIR-Bench (Nie et al., 2026), including event localization, cross-variable retrieval, trends, missingness, numerical summaries, and forecasting. Prospective questions use only information available at their cutoff. Appendices A and B define the cohorts and tasks.

Prediction and question-answering training. We evaluate the multi-modal Qwen3.5 models with 2, 4, and 9 billion parameters, denoted 2B, 4B, and 9B (Qwen Team, 2026). For prediction, one adapter is trained on both clinical outcomes and selected by validation average precision. We also evaluate this adapter on QA, without using QA examples or labels for training or model selection. For supervised QA, a separate adapter is trained on question-answer pairs and selected by validation accuracy. We evaluate both pooling designs defined in Section 3: Joint combines observations across variables, and Separate first summarizes each variable independently.

Comparisons. The unadapted models receive numerical text or time-series charts, labeled *Unadapted text* and *Unadapted plots*. Chart LoRA (Zhou et al., 2025) and Text LoRA (Kong et al., 2025) follow the cited approaches to adapt the decoder using charts and serialized measurements, respectively. We also compare ChatTS, MLLM4TS, ITFormer, and OpenTSLM. Each implementation follows the original method, with input formats, interface dimensions, and task-specific output heads adjusted for the shared Qwen models and clinical tasks. All follow a common training and selection protocol. GPT-5.6 Sol-high (OpenAI, 2026b) and Luna-high (OpenAI, 2026a) provide zero-shot references; specialized RNN- and Transformer-based clinical predictors (Ma et al., 2020; Kim et al., 2024; Wang et al., 2025b) appear in Appendix I.

Table 1 summarizes decoder updates and trainable parameter counts. Among the trained adaptations, `ViSTA` uses the fewest trainable parameters: 90.61% fewer than Chart LoRA and 91.73% fewer than Text LoRA at 4B.

Training and model selection. For each backbone size, all compared methods use the same pretrained backbone, patient splits, training data, and input information. We use a consistent validation frequency, early-stopping rule, and best-checkpoint selection criterion across methods. Hyperparameters are selected by grid search for each method, and all configurations are determined exclusively on the validation set.

Metrics. Prediction uses area under the receiver operating characteristic curve (AUROC) and average precision (AP); QA uses answer accuracy. We assess predicted probabilities with Brier score and expected calibration error (ECE; ten equal-width bins; Appendix J). Tables report standard deviations (SD) from patient-level bootstrap resampling, keeping each patient's questions together. QA differences are expressed in percentage points; AUROC and AP differences are absolute changes on the 0-1 scale.

Table 1: **Comparison methods and adaptation size.** Update describes adaptation within the language decoder. Parameters are trainable counts at 4B. ViSTA uses 0.516M/0.582M/0.780M at 2B/4B/9B.

Method	Update	Trainable params (M)
Unadapted text	None	0
Unadapted plots	None	0
Chart LoRA	LoRA	6.193
Text LoRA	LoRA	7.032
ChatTS	LoRA	6.692
MLLM4TS	Layer norm	0.884
ITFormer	Frozen	0.975
OpenTSLM	Cross-attention	3.074
ViSTA-Joint	Frozen	0.582
ViSTA-Separate	Frozen	0.582

5 Results and Analysis

We first compare outcome prediction and supervised temporal QA, then examine the effects of model size and question category.

5.1 Compact adaptation improves outcome prediction

Prediction through a frozen backbone. ViSTA-Joint has the highest mean scores on all four prediction metrics at every Qwen scale (Table 2). At 4B, AKI AP is 0.4650 versus Chart LoRA's 0.4508, an absolute difference of 0.0142; mortality AP is 0.2651 versus Text LoRA's 0.2464, a difference of 0.0187. Against native charts, the corresponding absolute differences are 0.1855 and 0.1158. These gains use the smallest trainable interface.

Probability quality also improves. Native models must predict future outcomes from irregular histories without endpoint-specific training. Their weak ranking, including sub-chance AKI AUROC at 2B and 9B, illustrates the difficulty of applying untuned language models to structured clinical prediction (Zhu et al., 2026a). Their normalized label probabilities can also reflect answer preferences (Zhao et al., 2021), so accurate risk calibration is not inherent to the language interface. After adaptation, Brier score improves over native charts in all twelve combinations of size, outcome, and encoding, and ECE in ten, without post-hoc calibration (Appendix J). At 9B, mortality ECE falls from 0.7261 to 0.0666 with ViSTA-Joint and 0.0284 with ViSTA-Separate.

Prediction gains do not automatically improve downstream QA. At 9B, all six alternatives have lower mean QA accuracy than native charts. Chart LoRA is lower by 1.87 percentage points against native charts and Text LoRA by 3.33 against native text. Outcome supervision may favor features or answer-token preferences for binary prediction over temporal questions. Frozen-decoder ITFormer also scores below both references, suggesting that input adaptation itself can affect QA. ViSTA-Joint's mean accuracy is higher than native text by 0.42, 2.33, and 4.00 percentage points across scales. At 4B, Text LoRA scores 42.05% versus ViSTA-Joint's 41.73%, a difference of 0.32 percentage points.

Comparison with GPT-5.6 Sol. The 2B ViSTA-Joint model's AKI AUROC is 0.7376 versus 0.7380 for text-input Sol; its AP is 0.4360 versus 0.4212. At 9B, ViSTA-Joint has higher mean scores across prediction metrics and both Sol input formats. Clinical adaptation brings compact models to this prediction level.

Table 2: **Prediction and QA performance after outcome training.** Values are mean ± patient-bootstrap SD; bold and underline mark the best and second-best Qwen means per scale and metric. Adapted Qwen models use outcome supervision only. GPT models receive no task-specific training, and their QA scores appear in Table 3.

	AKI		Mortality		QA
Method / input	AUROC	AP	AUROC	AP	Acc. (%)
Proprietary references					
GPT 5.6 Sol-high / text	0.7380 ±0.0293	0.4212 ±0.0246	0.7515 ±0.0231	0.2935 ±0.0254	-
GPT 5.6 Sol-high / plots	0.7283 ±0.0292	0.4126 ±0.0330	0.7221 ±0.0291	0.2733 ±0.0285	-
GPT 5.6 Luna-high / text	0.7422 ±0.0297	0.4382 ±0.0312	0.6870 ±0.0306	0.2780 ±0.0324	-
GPT 5.6 Luna-high / plots	0.6689 ±0.0284	0.3358 ±0.0342	0.6684 ±0.0306	0.2605 ±0.0368	-
Qwen3.5-2B					
Unadapted text	0.4249 ±0.0217	0.1774 ±0.0131	0.5247 ±0.0327	0.1264 ±0.0181	32.85 ±0.99
Unadapted plots	0.5297 ±0.0223	0.2301 ±0.0196	0.5235 ±0.0297	0.1172 ±0.0154	30.27 ±0.98
Chart LoRA	0.7117 ±0.0209	0.3988 ±0.0336	0.6703 ±0.0273	0.1906 ±0.0263	31.28 ±0.93
Text LoRA	0.7052 ±0.0195	0.3957 ±0.0366	0.6663 ±0.0286	0.1956 ±0.0310	32.16 ±1.04
ChatTS	0.6297 ±0.0220	0.3193 ±0.0299	0.6673 ±0.0257	0.1890 ±0.0277	29.87 ±0.93
MLLM4TS	0.6606 ±0.0217	0.3139 ±0.0256	0.6793 ±0.0245	0.1935 ±0.0260	27.67 ±0.96
ITFormer	0.5958 ±0.0220	0.2869 ±0.0287	0.6141 ±0.0301	0.1767 ±0.0244	28.58 ±0.89
OpenTSLM	0.5405 ±0.0225	0.2911 ±0.0264	0.5801 ±0.0319	0.1616 ±0.0258	29.50 ±0.93
ViSTA-Joint	**0.7376** ±0.0202	**0.4360** ±0.0245	**0.6989** ±0.0279	**0.2391** ±0.0309	**33.27** ±0.96
ViSTA-Separate	<u>0.7292</u> ±0.0202	<u>0.4261</u> ±0.0246	<u>0.6816</u> ±0.0275	<u>0.2047</u> ±0.0295	<u>33.18</u> ±0.98
Qwen3.5-4B					
Unadapted text	0.5124 ±0.0213	0.2078 ±0.0173	0.6352 ±0.0301	0.1962 ±0.0321	39.40 ±1.08
Unadapted plots	0.6250 ±0.0212	0.2795 ±0.0232	0.5872 ±0.0288	0.1493 ±0.0211	35.50 ±0.95
Chart LoRA	0.7292 ±0.0203	<u>0.4508</u> ±0.0354	0.7156 ±0.0259	0.2285 ±0.0335	40.95 ±0.95
Text LoRA	0.7279 ±0.0203	0.4412 ±0.0386	<u>0.7256</u> ±0.0270	<u>0.2464</u> ±0.0372	**42.05** ±1.12
ChatTS	0.6801 ±0.0217	0.3624 ±0.0314	0.7076 ±0.0243	0.2347 ±0.0350	34.68 ±0.97
MLLM4TS	0.6952 ±0.0208	0.3468 ±0.0283	0.7161 ±0.0235	0.2372 ±0.0339	30.91 ±0.98
ITFormer	0.6591 ±0.0219	0.3347 ±0.0298	0.6660 ±0.0268	0.2029 ±0.0316	31.05 ±0.96
OpenTSLM	0.6324 ±0.0232	0.3291 ±0.0285	0.6367 ±0.0291	0.1908 ±0.0307	30.82 ±0.98
ViSTA-Joint	**0.7440** ±0.0202	**0.4650** ±0.0262	**0.7350** ±0.0258	**0.2651** ±0.0384	<u>41.73</u> ±1.00
ViSTA-Separate	<u>0.7378</u> ±0.0208	0.4432 ±0.0262	0.7238 ±0.0264	0.2392 ±0.0226	41.64 ±1.01
Qwen3.5-9B					
Unadapted text	0.4046 ±0.0206	0.1313 ±0.0170	0.6104 ±0.0295	0.1950 ±0.0374	47.00 ±1.05
Unadapted plots	0.5204 ±0.0220	0.2115 ±0.0186	0.5643 ±0.0290	0.1430 ±0.0204	45.09 ±0.99
Chart LoRA	0.7377 ±0.0206	<u>0.4652</u> ±0.0349	0.7451 ±0.0256	0.2582 ±0.0378	43.22 ±0.97
Text LoRA	0.7343 ±0.0203	0.4521 ±0.0363	0.7430 ±0.0258	0.2777 ±0.0421	43.67 ±1.07
ChatTS	0.6476 ±0.0225	0.3329 ±0.0302	0.7416 ±0.0231	0.2726 ±0.0380	44.28 ±0.97
MLLM4TS	0.6709 ±0.0211	0.3030 ±0.0299	<u>0.7468</u> ±0.0231	<u>0.2796</u> ±0.0386	42.58 ±0.90
ITFormer	0.5935 ±0.0223	0.3001 ±0.0304	0.6811 ±0.0267	0.2313 ±0.0364	42.62 ±0.92
OpenTSLM	0.5374 ±0.0245	0.2679 ±0.0296	0.6300 ±0.0283	0.2060 ±0.0346	42.46 ±0.95
ViSTA-Joint	**0.7502** ±0.0203	**0.4840** ±0.0266	**0.7559** ±0.0256	**0.3021** ±0.0327	<u>51.00</u> ±0.97
ViSTA-Separate	<u>0.7423</u> ±0.0201	0.4627 ±0.0259	0.7440 ±0.0249	0.2646 ±0.0276	**51.73** ±0.96

5.2 Strong supervised QA with a compact numerical interface

Chart LoRA and Text LoRA achieve the highest mean supervised QA accuracy among Qwen adaptations at every scale (Table 3). At 4B, they reach 72.09% and 74.15%, versus 69.27% for ViSTA-Separate. LoRA directly adapts the decoder to questions and answers, giving it greater task-specific flexibility. ViSTA learns how to present numerical evidence to a fixed decoder.

At 4B, ViSTA-Separate gains 29.87 percentage points over native text and 33.77 over native charts, coming within 2.82-4.88 points of LoRA with more than 90% fewer trainable parameters. Both encodings outperform ChatTS, MLLM4TS, ITFormer, and OpenTSLM across scales. The contribution is strong QA through a compact interface, using the same frozen-backbone design that leads the prediction comparisons.

Table 3: **Large QA gains with a compact frozen interface.** Accuracy (%) is mean ± patient-bootstrap SD; Δ is the gain over the unadapted chart model in percentage points. GPT models receive no task-specific training. Qwen adaptations are trained on QA; bold and underline mark the best and second-best Qwen means per scale.

GPT reference / input	QA without task-specific training (%)					
GPT 5.6 Sol-high / text	90.51 ±0.70					
GPT 5.6 Sol-high / plots	84.24 ±0.79					
GPT 5.6 Luna-high / text	80.30 ±0.93					
GPT 5.6 Luna-high / plots	74.05 ±1.13					
Method	2B		4B		9B	
	Acc. (%)	Δ (pp)	Acc. (%)	Δ (pp)	Acc. (%)	Δ (pp)
Unadapted text	32.85 ±0.99	+2.58	39.40 ±1.08	+3.90	47.00 ±1.05	+1.91
Unadapted plots	30.27 ±0.98	+0.00	35.50 ±0.95	+0.00	45.09 ±0.99	+0.00
Chart LoRA	<u>67.42</u> ±0.87	+37.15	<u>72.09</u> ±0.92	+36.59	<u>73.00</u> ±0.88	+27.91
Text LoRA	**67.78** ±0.94	+37.51	**74.15** ±0.94	+38.65	**75.04** ±0.93	+29.95
ChatTS	54.12 ±1.07	+23.85	59.18 ±1.02	+23.68	62.81 ±1.01	+17.72
MLLM4TS	52.66 ±1.00	+22.39	58.50 ±1.02	+23.00	62.05 ±0.99	+16.96
ITFormer	52.31 ±1.01	+22.04	57.36 ±1.05	+21.86	62.00 ±0.97	+16.91
OpenTSLM	51.72 ±0.99	+21.45	57.41 ±1.03	+21.91	62.05 ±0.99	+16.96
ViSTA-Joint	63.05 ±1.00	+32.78	68.27 ±0.99	+32.77	70.55 ±0.95	+25.46
ViSTA-Separate	63.55 ±0.97	+33.28	69.27 ±0.96	+33.77	69.32 ±0.93	+24.23

GPT-5.6 Sol remains substantially stronger: 90.51% with text and 84.24% with charts, without task-specific training. This highlights the temporal reasoning capabilities of stronger proprietary models. Numerical adaptation narrows the gap for compact open models, motivating its evaluation with stronger pretrained backbones.

5.3 Larger backbones further improve numerical adaptation

Prediction improves through 9B. ViSTA-Joint AKI AP rises from 0.4360 to 0.4840 and mortality AP from 0.2391 to 0.3021 across 2B-9B (Figure 3). Native prediction lacks this consistent trend; adapted 2B even exceeds native 9B charts on both AP measures (0.2115 and 0.1430). Capacity benefits prediction when coupled with numerical adaptation.

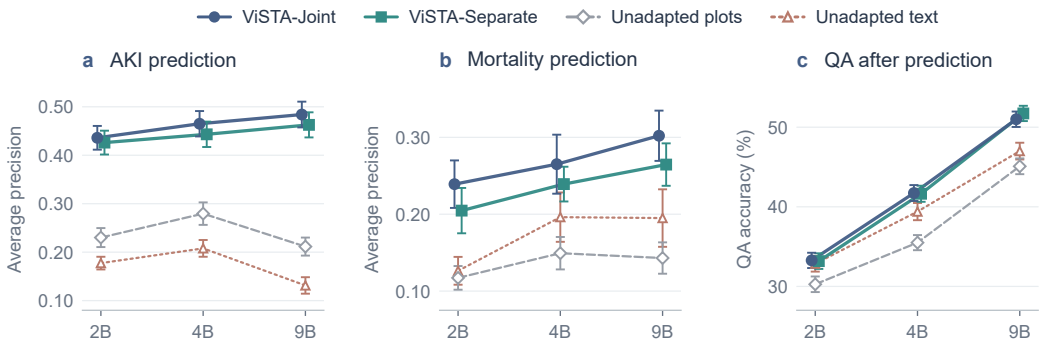

Figure 3: **Prediction gains persist as the backbone grows.** Adapted models use outcome supervision only; the right panel evaluates QA without QA training. Error bars show patient-bootstrap SD. Both encodings have identical parameter counts at each scale.

Early cross-variable pooling benefits prediction more than QA. ViSTA-Joint has higher mean scores than ViSTA-Separate on every prediction metric, while outcome-trained QA differs by at

most 0.73 percentage points. Both bridges access all variables; joint pooling additionally mixes them before compression. This suggests early interaction helps retain combinations relevant to patient risk. Supervised QA favors ViSTA-Separate at 2B and 4B, ViSTA-Joint at 9B. ViSTA-Separate's stronger 4B missingness counting (63.0% versus 54.0%) is consistent with preserving variable-specific detail. Pooling preference thus depends on the task (Appendix E.1).

Design ablations. Removing zero initialization or RMS scaling reduces all four prediction metrics and QA performance (Table 8). Keeping only visual tokens or only numerical residuals at inference lowers both prediction and QA scores. Appendix F discusses their interpretation.

5.4 Supervised QA gains span numerical and temporal operations

At 4B, ViSTA-Separate scores higher than the category-wise best of ChatTS, MLLM4TS, ITFormer, and OpenTSLM in eight categories, ties in one, and trails in two (Figure 4). Gains span numerical summaries (+26.5 percentage points), threshold forecasting (+19.5), segmented trends (+13.5), and cross-variable retrieval (+11.5). These gains span numerical access and temporal operations under QA supervision.

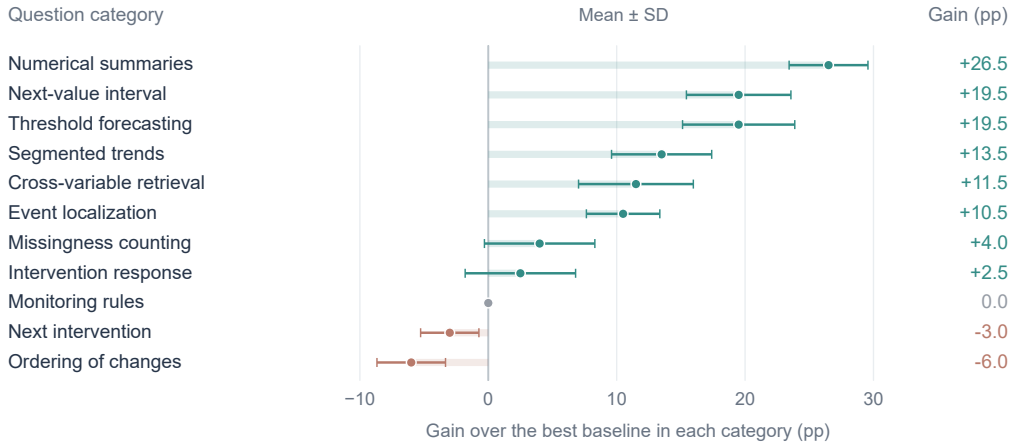

Figure 4: **Supervised QA gains across most categories.** At 4B, ViSTA-Separate is compared with the category-wise best of ChatTS, MLLM4TS, ITFormer, and OpenTSLM. Error bars show paired patient-bootstrap SD with baseline selection repeated per replicate. Absolute scores appear in Appendix G.

The advantage persists without numerical summaries. Without numerical summaries, ViSTA-Separate averages 66.45% across ten categories, versus 59.25% for the category-wise best baseline. The 7.20 percentage-point advantage also holds over every individual baseline. Ordering of changes and next-intervention prediction remain weaker by 6.0 and 3.0 percentage points, respectively.

6 Discussion and Conclusion

ViSTA improves clinical prediction and supervised temporal QA with fewer than one million trainable parameters while keeping the pretrained model fixed. The 2B model approaches GPT-5.6 Sol on AKI prediction, prediction improves through 9B, and QA gains extend beyond numerical summaries. These results motivate applying the adapter to more capable pretrained models.

The evaluation uses one hospital system, one training seed, and automatically generated multiple-choice questions. Specialized clinical predictors and LoRA trained on QA remain stronger in their respective tasks. Matched training controls are needed to attribute QA gains to outcome supervision and isolate the contribution of visual features. Further evaluation should include external cohorts, open-ended interaction, and systems combining separate prediction and QA models (Shi et al., 2025).

AI USE STATEMENT

Generative AI tools assist with language editing, manuscript organization, and numerical consistency checks. Methods and experiments are designed and executed by humans.

ETHICS STATEMENT

This study uses deidentified clinical records from MIMIC-IV under its controlled-access framework. The performance of OpenAI models on all datasets was processed using the secure Azure OpenAI API, with human review of the data waived. The experiments evaluate retrospective prediction and record understanding.

REPRODUCIBILITY STATEMENT

Section 3 specifies the model, and Section 4 describes the comparison methods. Appendices A-J document cohort construction, question definitions, prompt templates, input visibility, implementation settings, and metric computation. Methods share a common validation and checkpoint-selection protocol, with hyperparameters chosen by grid search on the validation set. Reproducing the clinical data extraction requires authorized access to MIMIC-IV under its data-use agreement.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning, 2022. URL https://arxiv.org/abs/2204.14198.

Junyi Gao, Cao Xiao, Lucas M Glass, and Jimeng Sun. Dr. agent: Clinical predictive model via mimicked second opinions. *Journal of the American Medical Informatics Association*, 27(7): 1084–1091, 2020.

Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6:96, 2019. doi: 10.1038/s41597-019-0103-9.

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, 2021. URL https://arxiv.org/abs/2106.09685.

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, 2023. URL https://arxiv.org/abs/2310.01728.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV (version 3.1). PhysioNet, 2024. URL https://physionet.org/content/mimiciv/3.1/.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, et al. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10:1, 2023. doi: 10.1038/s41597-022-01899-x.

Damjan Kalajdzievski. Scaling Laws for Forgetting When Fine-Tuning Large Language Models, 2024. URL https://arxiv.org/abs/2401.05605.

Taeri Kim, Jiho Heo, Hongil Kim, Kijung Shin, and Sang-Wook Kim. Vita:'carefully chosen and weighted less' is better in medication recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 8600–8607, 2024.

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, 2025. URL https://arxiv.org/abs/2503.01875.

Patrick Langer, Thomas Kaar, Max Rosenblattl, Maxwell A. Xu, Winnie Chow, Martin Maritsch, Robert Jakob, Ning Wang, Juncheng Liu, Aradhana Verma, Brian Han, Daniel Seung Kim, Henry Chubb, Scott Ceresnak, Aydin Zahedivash, Alexander Tarlochan Singh Sandhu, Fatima Rodriguez, Daniel McDuff, Elgar Fleisch, Oliver Aalami, Filipe Barata, and Paul Schmiedmayer. OpenTSLM: Time-Series Language Models for Reasoning over Multivariate Medical Text- and Time-Series Data, 2025. URL https://arxiv.org/abs/2510.02410.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, 2023. URL https://arxiv.org/abs/2301.12597.

Qinghua Liu, Sam Heshmati, Zheda Mai, Zubin Abraham, John Paparrizos, and Liu Ren. MLLM4TS: Leveraging Vision and Multimodal Language Models for General Time-Series Analysis, 2025. URL https://arxiv.org/abs/2510.07513.

Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 833–840, 2020.

Frank Nie, Ethan B. Liu, Yuan Zhu, Loe Yan, Wei Fan, and Jindong Han. CLIR-Bench: Benchmarking Multimodal Question Answering over Irregular Clinical Time Series, 2026. URL https://arxiv.org/abs/2607.09880.

OpenAI. GPT-5.6 Luna. OpenAI API Documentation, 2026a. URL https://developers.openai.com/api/docs/models/gpt-5.6-luna. Accessed: 2026-09-25.

OpenAI. GPT-5.6 Sol. OpenAI API Documentation, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol. Accessed: 2026-09-25.

Qwen Team. Qwen3.5: Official model cards. Hugging Face, 2026. URL https://huggingface.co/Qwen/Qwen3.5-4B. 2B, 4B, and 9B pretrained checkpoints.

Yu Shi, Wei Xu, and Pingzhao Hu. Out of distribution learning in bioinformatics: advancements and challenges. *Briefings in Bioinformatics*, 26(3):bbaf294, 2025.

Yu Shi, Wei Xu, and Pingzhao Hu. Deep unsupervised domain adaptation for translating cancer dependency maps from cell lines to breast cancer tumor genomics. *Genetic Epidemiology*, 50(5): e70044, 2026.

Karan Singhal, Shekoofeh Azizi, Tao Tu, et al. Large language models encode clinical knowledge. *Nature*, 620:172-180, 2023. doi: 10.1038/s41586-023-06291-2.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, et al. Adapted large language models can outperform medical experts in clinical text summarization. *Nature Medicine*, 30:1134-1142, 2024. doi: 10.1038/s41591-024-02855-5.

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, 2025a. URL https://arxiv.org/abs/2506.20093.

Zixiang Wang, Yinghao Zhu, Huiya Zhao, Xiaochen Zheng, Dehao Sui, Tianlong Wang, Wen Tang, Yasha Wang, Ewen Harrison, Chengwei Pan, et al. Colacare: Enhancing electronic health record modeling through large language model-driven multi-agent collaboration. In *Proceedings of the ACM on Web Conference 2025*, pp. 2250–2261, 2025b.

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, 2024. URL https://arxiv.org/abs/2412.03104.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697-12706. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

Ziming Zhao, ChengAo Shen, Hanghang Tong, Dongjin Song, Zhigang Deng, Qingsong Wen, and Jingchao Ni. From Images to Signals: Are Large Vision Models Useful for Time Series Analysis?, 2025. URL https://arxiv.org/abs/2505.24030.

Luca Zhou, Pratham Yashwante, Marshall Fisher, Alessio Sampieri, Zihao Zhou, Fabio Galasso, and Rose Yu. CaTS-Bench: Can Language Models Describe Time Series?, 2025. URL https://arxiv.org/abs/2509.20823.

Yinghao Zhu, Junyi Gao, Zixiang Wang, Weibin Liao, Xiaochen Zheng, Lifang Liang, Miguel O. Bernabeu, Yasha Wang, Lequan Yu, Chengwei Pan, Ewen M. Harrison, and Liantao Ma. ClinicRealm: Re-evaluating large language models with conventional machine learning for non-generative clinical prediction tasks. *npj Digital Medicine*, 9:319, 2026a. doi: 10.1038/s41746-026-02539-z.

Yinghao Zhu, Dehao Sui, Zixiang Wang, Xuning Hu, Lei Gu, Yifan Qi, Tianchen Wu, Ling Wang, Yuan Wei, Wen Tang, et al. Augmenting clinical decision-making with an interactive and interpretable ai copilot: A real-world user study with clinicians in nephrology and obstetrics. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pp. 1–26, 2026b.

A Cohort and Outcome Definitions

Study population. We include adults with an ICU stay of at least 24 hours in MIMIC-IV v3.1 and use the first eligible stay per patient. Training, validation, and test sets are disjoint at the patient level. A patient may contribute to both prediction endpoints or to multiple QA categories within the same split.

Clinical variables and availability. The eleven numerical variables are heart rate, mean arterial pressure, systolic pressure, diastolic pressure, oxygen saturation, respiratory rate, temperature, lactate, creatinine, white blood cell count, and urine output. Observations retain their values, variable identities, and timestamps. Prediction and prospective QA use only observations measured and available by the question cutoff. Visible intervention starts are provided as timestamped text; future observations and outcome labels remain hidden.

Acute kidney injury. The landmark is ICU hour 24 and the outcome window is $(24, 48]$ hours. AKI is defined by a creatinine increase of at least 0.3 mg/dL within 48 hours or at least 1.5 times the minimum qualifying prior measurement within seven days. Eligibility requires an ICU stay of at least 48 hours, two baseline creatinine measurements available by the landmark (including one at or after hour 12), and follow-up measurement at or after hour 36. Patients with AKI or renal replacement therapy before the landmark are ineligible. This is a recorded creatinine-defined endpoint; urine-output staging is not included. Test prevalence is 20.5%.

In-hospital mortality. The mortality cohort consists of patients alive and hospitalized at ICU hour 24. The outcome is death before discharge from the current hospitalization. Test prevalence is 10.9%.

B Temporal Question Construction

The question taxonomy follows CLIR-Bench (Nie et al., 2026). We could not reproduce the original benchmark inputs from the released materials. We therefore rebuilt questions, labels, and distractors on our MIMIC-IV cohort using an independent implementation of the published task definitions and answer-construction rules. The resulting evaluation uses our patient splits and newly generated questions. Each question presents four distinct answer options in randomized order. Table 4 defines the eleven categories. Appendix C gives prompt templates, and Appendix H reports category-wise majority-answer and random baselines.

Retrospective and prospective questions. Retrospective questions concern the supplied observation window. Forecasting and next-intervention questions expose only information available at their cutoff. Intervention prediction targets recorded care, while intervention-response questions describe observed changes around treatment starts.

Answer construction. Thresholds and stability tolerances are stated in each question. Missingness counts absent entries on the union of observation timestamps. Numerical-summary distractors perturb values from the correct summary. These operations produce reproducible labels for evaluating temporal and numerical understanding.

Table 4: **Temporal QA definitions.** Each question specifies its observation window, thresholds, and tolerances.

Category	Answer construction
TG: Event localization	Locate the start of the first consecutive pair of observed threshold breaches within one of four quarters of a time window.
ASR: Cross-variable retrieval	Find an extremum of one variable and retrieve the nearest observation of another; resolve time ties using the earlier reading.
TPR: Segmented trends	Split ordered observations into five nearly equal blocks, compare medians using a stated tolerance, and merge adjacent repeated directions.
MA: Missingness counting	Count a variable's observed and absent entries on the union of recorded timestamps in a specified window.
TSS: Numerical summaries	Select the correct first, last, minimum, and maximum values for two variables from summaries with numerical distractors.
CVR: Ordering of changes	Compare the first specified directional changes of two variables, including simultaneous and absent-change cases.
IR: Intervention response	Compare pre- and post-intervention medians within stated windows, requiring sufficient observations on both sides.
TF: Threshold forecasting	Predict a future observed threshold breach and identify the direction of the final two visible observations.
NIF: Next-value interval	Predict the interval containing the first subsequent observed value within a stated horizon.
IID: Next intervention	Predict the category combination at the first subsequent recorded intervention start.
MED: Monitoring rules	Apply a stated rule combining repeated blood-pressure and oxygen-saturation threshold breaches and observation sufficiency.

C PROMPT TEMPLATES

The following templates show the common Qwen instructions and six task questions. Braced fields are filled from the visible record or question parameters. Chart inputs and numerical text use the same task wording and answer labels. Charts and intervention context contain only information available within the question's visible interval.

C.1 SHARED INSTRUCTIONS AND INPUT FORMAT

System message

```
Answer the clinical record question using the supplied observations.
```

Chart input: user message

```
{variable name 1}
{chart image 1}
...
{variable name 11}
{chart image 11}
Intervention starts: {time}: {intervention}; ...
{task question and answer instruction}
```

Numerical text: user message

```
Visible ICU history: {start} to {end} hours.
{variable name} ({unit}), hour:value: {time}:{value}; ...
...
Intervention starts: {time}: {intervention}; ...
{task question and answer instruction}
```

The numerical-text block repeats for all eleven variables; a variable with no measurements is represented by `No observations.` The chart format shows the per-variable rendering used in the experiments.

C.2 CLINICAL PREDICTION

Acute kidney injury

```
Using only information available by ICU hour 24: Will a new recorded
creatinine-defined AKI event occur during ICU hours (24, 48]? An event
is a creatinine rise of at least 0.3 mg/dL over a prior value within 48
hours or at least 1.5 times a prior value within 7 days. Answer 0 for
no or 1 for yes.
Return only 0 or 1.
```

In-hospital mortality

```
Using only information available by ICU hour 24, will this patient die
before discharge from the current hospitalization? Answer 0 for
survival to discharge or 1 for in-hospital death.
Return only 0 or 1.
```

C.3 TEMPORAL QUESTION ANSWERING

Each QA question below is followed by the same option block. The four distinct answer texts are permuted before assigning letters; the correct letter depends on this permutation.

QA option block and answer instruction

```
Options:
A. {option A}
B. {option B}
C. {option C}
D. {option D}
Return only the answer letter A, B, C, or D.
```

Event localization

```
Within ICU hours 0-{end}, which quarter of the window contains the start
of the first pair of consecutive observed {variable} values both {below
or above} {threshold} {unit}?  Count observed pairs, without assuming
values between measurements.
```

The options identify the four quarters of the observation window, including their time boundaries.

Cross-variable retrieval

```
Within ICU hours 0-{end}, find the earliest {maximum or minimum} of
{variable 1}.  Which interval contains the closest measured {variable 2}
({unit})?  Resolve equal time distances using the earlier reading.
Intervals include the lower endpoint only.
```

The options are four numerical intervals for the second variable.

Numerical summaries

```
Which numerical handoff summary matches the observed measurements during
ICU hours 0-{end}?
```

Each option lists the first, last, minimum, and maximum measurements for two variables, with their units. Distractors perturb these summary values.

Threshold forecasting

```
Using only observations through ICU hour {cutoff}, predict whether any
observed {variable} will be {below or above} {threshold} {unit} during
the next {horizon} hours, and identify the direction of the last two
visible readings.
```

The options combine future threshold breach or absence with nondecreasing or decreasing last observed values. Only measurements available by the cutoff enter the input; subsequent observations determine the reference answer.

D Implementation and Evaluation Details

Numerical features. Variable values are centered by the training-set median and scaled by the training interquartile range, floored at 10^{-3}. For normalized value $\tilde{x}_i$, interval $[t_s, t_e]$, and within-variable gap g_i, the event features are

$$\psi_i = \Bigg[\operatorname{sign}(\tilde{x}_i)\log(1+|\tilde{x}_i|),\ \frac{t_i - t_s}{\max(t_e - t_s, 1)},\ \frac{\log(1+\max(t_i, 0))}{10},\ \frac{\log(1+g_i)}{10},\ \mathbf{1}_{\text{first},i},\ \frac{\log(1+\max(t_e - t_s, 1))}{10}\Bigg]. \tag{7}$$

Gaps and first-observation flags are recomputed using only visible observations. The event MLP has two linear layers with a SiLU activation. Missing-variable tokens are appended to the event memory, which accounts for the distinction between M and M_e.

Table 5: **Selected `ViSTA` interface and training configuration.** Settings are selected on the validation set.

Setting	Value
Clinical variables C; slots per variable	11; 12
Total internal slots K	132
Visual tokens per chart; total N	192; 2,112
Numerical optimizer	AdamW, learning rate 10^{-4}
Weight decay; gradient norm ceiling	0.01; 1.0
Numerical precision	BF16 backbone
Hyperparameter selection	Grid search on the validation set

Positional queries and frozen components. Position features are normalized horizontal and vertical coordinates of the visual grid. A two-layer MLP maps coordinates to numerical width, and the chart's variable embedding is added. Residuals preserve image delimiters and visual positions. The final linear layer, including its bias, starts at zero. Pretrained parameters remain frozen, while gradients propagate through the language decoder to the interface.

Hyperparameter and model selection. All methods follow the protocol in Section 4. Hyperparameters are selected by grid search on the validation set. Prediction models are selected by validation AP and QA models by validation accuracy. The selected prediction model is also evaluated on QA without further training or QA-based selection. Test results do not influence configuration choices.

Input representations. Numerical text lists variable names, units, and timestamp-value observations. Visual inputs render one chart per variable. Both representations describe the history available at the question cutoff, together with visible intervention context.

Scoring and uncertainty. QA selects the most probable answer label. Prediction metrics use continuous positive-label probabilities obtained by normalizing the original language-head logits over the two binary labels. Training uses answer-token cross-entropy.

Standard deviations use 2,000 patient-level bootstrap resamples, keeping all questions from the same patient together. QA accuracy is averaged across categories. For paired differences, both methods use the same resampled patients. Comparisons with the best baseline in each category reselect that baseline within every resample. Bootstrap SD measures test-patient uncertainty for one trained seed.

E Scaling and Variable Encoding

Figure 3 traces prediction and QA performance after outcome training across scales, alongside both unadapted input baselines. Table 6 compares `ViSTA`-Joint's QA accuracy with the unadapted text baseline and the unadapted chart baseline. The respective differences are 0.42/2.33/4.00 and

3.00/6.23/5.91 percentage points at 2B, 4B, and 9B. The table separately reports supervised QA gains over the unadapted chart baseline. Interface size is 0.516M at 2B, 0.582M at 4B, and 0.780M at 9B.

Table 6: **Prediction and QA gains over the unadapted models.** Prediction AUROC changes use the unadapted chart baseline as the reference. QA after prediction training is compared with both unadapted input baselines. These columns use ViSTA-Joint; supervised QA gains use their respective encodings and are relative to the unadapted chart baseline.

	Prediction ΔAUROC		QA after prediction (pp)		Supervised QA Δ (pp)	
Scale	AKI	Mortality	vs. plots	vs. text	ViSTA-Joint	ViSTA-Separate
2B	+0.2079	+0.1754	+3.00	+0.42	+32.78	+33.28
4B	+0.1190	+0.1478	+6.23	+2.33	+32.77	+33.77
9B	+0.2298	+0.1916	+5.91	+4.00	+25.46	+24.23

E.1 Encoding comparison

ViSTA-Joint and ViSTA-Separate differ in the attention mask during the first numerical pooling stage. Parameter sharing, latent width, slot count, and the downstream bridge are unchanged. ViSTA-Separate pools observations within each variable; subsequent visual queries can read all pooled variables. Table 7 calculates ViSTA-Joint minus ViSTA-Separate from the final prediction and QA scores. ViSTA-Joint has higher mean prediction AP at every size. QA after prediction training differs by 0.09 percentage points at 2B and 4B, and favors ViSTA-Separate by 0.73 percentage points at 9B.

Table 7: **ViSTA-Joint minus ViSTA-Separate encoding.** Positive entries favor ViSTA-Joint. Prediction and supervised QA use separately trained adapters. The first QA column evaluates the prediction-trained adapter without further training.

Scale	AKI AP	Mortality AP	QA after prediction (pp)	Supervised QA (pp)
2B	+0.0099	+0.0344	+0.09	-0.50
4B	+0.0218	+0.0259	+0.09	-1.00
9B	+0.0213	+0.0375	-0.73	+1.23

F Input Interventions and Design Ablations

Table 8 separates changes to the input of a trained model from changes to adapter training. All results use the 4B backbone and outcome supervision; QA uses no question-answer training examples.

Inference-time input interventions. We supply either the original visual tokens V or the scaled numerical residual $R = s \odot \Delta$ at the visual positions. The trained adapter is unchanged. The residual-only input retains s computed from V, as well as the original positional structure, so it still depends on visual information. Both interventions reduce prediction and QA scores relative to the full input. These results measure sensitivity to removing an input component; the altered inputs differ from those seen during training and do not isolate semantic use of chart content.

Zero initialization and RMS scaling. Removing zero initialization lowers AKI and mortality AP by 0.0198 and 0.0192; removing RMS scaling lowers them by 0.0148 and 0.0148. These are absolute AP differences. Both changes also reduce AUROC and transfer QA accuracy. This pattern supports starting adaptation from the native visual representation and expressing corrections relative to its feature scale for outcome prediction and QA.

Table 8: **Input interventions and design ablations at 4B.** The full model is ViSTA-Joint from Table 2. Here $R = s \odot \Delta$ denotes the scaled numerical residual. Visual-only and residual-only inputs are evaluated at inference without retraining. The last two rows remove the indicated component during training. QA is evaluated without QA supervision; higher is better for every metric.

	AKI		Mortality		Transfer QA
Method / input	AUROC	AP	AUROC	AP	Acc. (%)
ViSTA: full $V + R$	**0.7440**	**0.4650**	**0.7350**	**0.2651**	**41.73**
Inference-time input interventions					
Visual only: V	0.6250	0.2795	0.5872	0.1493	35.50
Residual only: R	0.5418	0.2216	0.5690	0.1332	29.95
Training ablations					
Without zero initialization	0.7303	0.4452	0.7205	0.2459	40.15
Without RMS scaling	0.7357	0.4502	0.7250	0.2503	39.85

G CATEGORY-LEVEL SUPERVISED QA

Table 9 gives the scores and differences underlying Figure 4. All methods are trained on QA. We select the best of ChatTS, MLLM4TS, ITFormer, and OpenTSLM in each category, making the comparison at least as strong as any single baseline. ViSTA-Separate has higher mean accuracy than this reference in eight categories, ties in monitoring rules, and scores lower in ordering of changes and next-intervention prediction.

Table 9: **Supervised QA gains over the best baseline in each category at 4B.** All methods are trained on QA. The reference is the maximum of ChatTS, MLLM4TS, ITFormer, and OpenTSLM in each category. Δ is ViSTA-Separate minus this reference, in percentage points. Values include patient-bootstrap SD.

Category	Best baseline (%)	ViSTA-Joint (%)	ViSTA-Separate (%)	Δ (pp)
ASR: Cross-variable retrieval	41.50 ± 3.29	52.00 ± 3.49	53.00 ± 3.47	11.50 ± 4.47
CVR: Ordering of changes	68.50 ± 3.25	63.00 ± 3.42	62.50 ± 3.42	-6.00 ± 2.67
IID: Next intervention	83.50 ± 2.44	78.00 ± 2.96	80.50 ± 2.87	-3.00 ± 2.27
IR: Intervention response	59.50 ± 3.40	64.50 ± 3.39	62.00 ± 3.40	2.50 ± 4.30
MA: Missingness counting	59.00 ± 3.19	54.00 ± 3.52	63.00 ± 3.41	4.00 ± 4.30
MED: Monitoring rules	75.50 ± 3.07	76.00 ± 3.04	75.50 ± 3.07	0.00 ± 0.00
NIF: Next-value interval	47.00 ± 2.95	66.00 ± 3.33	66.50 ± 3.30	19.50 ± 4.07
TF: Threshold forecasting	33.50 ± 2.93	53.50 ± 3.60	53.00 ± 3.56	19.50 ± 4.37
TG: Event localization	66.50 ± 3.28	77.00 ± 2.82	77.00 ± 2.98	10.50 ± 2.86
TPR: Segmented trends	58.00 ± 3.39	70.50 ± 3.28	71.50 ± 3.17	13.50 ± 3.90
TSS: Numerical summaries	71.00 ± 2.91	96.50 ± 1.33	97.50 ± 1.10	26.50 ± 3.07

Sensitivity to the numerical-summary category. Numerical summaries yield the highest accuracy and largest gain for ViSTA. Excluding this category leaves ViSTA-Separate at 66.45%, compared with 59.25% for the best baseline selected in each of the other ten categories. This 7.20 percentage-point difference therefore also holds against every individual baseline in the comparison.

H MAJORITY-ANSWER AND RANDOM QA BASELINES

We evaluate two baselines that use no patient information on the same 2,200 test questions, with 200 questions per category. The *majority-answer* baseline selects the most frequent answer letter (A, B, C, or D) among the 2,000 training questions in each category and always predicts that letter for the corresponding test category. Ties are resolved in alphabetical order. This measures answer-position preferences after option randomization. The *uniform random* baseline selects each of the four options with probability $1/4$, giving an exact expected accuracy of 25% in every category.

Table 10: **QA baselines without patient information.** Majority selects each category's most frequent training answer letter and reports test accuracy. Random is the exact expectation of uniform four-option guessing. All accuracies are percentages.

Category	Test n	Majority letter	Majority (%)	Random (%)
Event localization	200	A	27.00	25.00
Cross-variable retrieval	200	A	28.50	25.00
Segmented trends	200	A	21.00	25.00
Missingness counting	200	A	29.50	25.00
Numerical summaries	200	D	25.50	25.00
Ordering of changes	200	C	30.50	25.00
Intervention response	200	B	22.50	25.00
Threshold forecasting	200	D	25.00	25.00
Next-value interval	200	C	22.50	25.00
Next intervention	200	A	26.00	25.00
Monitoring rules	200	A	23.00	25.00
Macro-average	2200	-	25.55	25.00

The majority-answer baseline has macro-average accuracy of 25.55%, ranging from 21.00% to 30.50% across categories. Uniform random guessing has expected macro-average accuracy of 25.00%. These references complement the model comparisons in Table 9; the majority letter is chosen using training answers only.

I SPECIALIZED CLINICAL PREDICTION REFERENCES

The specialized prediction systems in Table 11 contextualize the clinical endpoint scores. LSTM, GRU, ConCare, VITA, and ColaCare exceed the 9B `ViSTA`-Joint interface on both endpoints. Their task-specific output heads produce endpoint scores; answering new temporal questions requires an additional language component. `ViSTA` retains the pretrained language head, allowing the prediction-trained interface to be evaluated on QA. At 9B, `ViSTA`-Joint also has higher mean scores than the specialized Transformer reference on all four prediction metrics.

Table 11: **Specialized clinical prediction references.** Reported AUROC/AP ± SD. These systems have no QA evaluation in this study.

	AKI		Mortality	
Method	AUROC	AP	AUROC	AP
LSTM	0.7793 ± 0.0178	0.5244 ± 0.0354	0.7780 ± 0.0212	0.3357 ± 0.0473
GRU	0.7761 ± 0.0183	0.5170 ± 0.0341	0.7728 ± 0.0233	0.3248 ± 0.0474
Transformer	0.7378 ± 0.0183	0.4400 ± 0.0340	0.7288 ± 0.0256	0.2997 ± 0.0465
ConCare	0.7946 ± 0.0175	0.5367 ± 0.0331	0.7881 ± 0.0218	0.3505 ± 0.0484
VITA	0.8006 ± 0.0170	0.5429 ± 0.0354	0.7969 ± 0.0211	0.3569 ± 0.0485
ColaCare	0.8074 ± 0.0170	0.5506 ± 0.0328	0.8020 ± 0.0210	0.3615 ± 0.0491

J PREDICTION PROBABILITY CALIBRATION

We evaluate the model's positive-event probabilities directly, without post-hoc recalibration. For outcome $y_i \in \{0, 1\}$ and predicted probability p_i, the Brier score is

$$\text{Brier} = \frac{1}{n} \sum_{i=1}^{n} (p_i - y_i)^2. \tag{8}$$

ECE partitions $[0, 1]$ into ten equal-width bins B_b and compares each bin's mean probability with its event frequency:

$$\text{ECE} = \sum_{b=1}^{10} \frac{|B_b|}{n} \left| \frac{1}{|B_b|} \sum_{i \in B_b} p_i - \frac{1}{|B_b|} \sum_{i \in B_b} y_i \right|. \tag{9}$$

Empty bins contribute zero. This measures calibration of the predicted event probability. Brier score evaluates overall probability error and reflects both calibration and discrimination. Lower values are better for both metrics. Standard deviations use 2,000 patient-level bootstrap resamples with fixed bin boundaries.

Interpreting native probabilities. All Qwen methods use the same label convention: 1 denotes the event and 0 its absence, and p_i is the normalized probability of label 1. For native models, these answer probabilities have not been fitted to clinical outcomes or event prevalence. Answer preferences can therefore produce extreme probabilities and large calibration errors (Zhao et al., 2021). The large 9B native-chart mortality errors characterize this zero-shot use of label probabilities; they are distinct from the weak patient ranking measured by AUROC. The main prediction comparison also includes clinically trained Chart LoRA and Text LoRA baselines.

Table 13 reports results for the unadapted models with text and chart inputs. Relative to the unadapted chart baseline, both `ViSTA` encodings lower Brier score at every model size and for both outcomes (Table 12). ECE also decreases except for 4B AKI, where the unadapted chart model already has low ECE. Reporting both metrics distinguishes improvements in probability error from uniform gains in calibration.

Table 12: **Numerical adaptation improves prediction probability quality.** Probabilities are evaluated without recalibration; lower is better. Bold marks the minimum for each model size, outcome, and metric. Table 13 reports both unadapted input baselines, including patient-bootstrap SD.

	AKI			Mortality		
Scale	Unadapted plots	ViSTA-Joint	ViSTA-Separate	Unadapted plots	ViSTA-Joint	ViSTA-Separate
ECE ↓						
2B	0.1965	0.0468	**0.0444**	0.4962	**0.0328**	0.0338
4B	**0.0378**	0.0437	0.0614	0.3860	0.0383	**0.0373**
9B	0.2418	0.0820	**0.0511**	0.7261	0.0666	**0.0284**
Brier score ↓						
2B	0.2026	0.1461	**0.1454**	0.3482	**0.0937**	0.0939
4B	0.1590	**0.1395**	0.1474	0.2537	**0.0911**	0.0935
9B	0.2364	0.1437	**0.1412**	0.6257	0.0922	**0.0915**

Table 13: **Calibration of unadapted models with text and chart inputs.** Values are mean ± patient-bootstrap SD. ECE uses ten equal-width probability bins. Bold marks the smallest mean within each scale and metric.

		AKI		Mortality	
Scale	Method	ECE ↓	Brier ↓	ECE ↓	Brier ↓
2B	Unadapted text	0.3462 ±0.0133	0.2895 ±0.0021	0.5665 ±0.0096	0.4214 ±0.0044
	Unadapted plots	**0.1965** ±0.0130	**0.2026** ±0.0030	**0.4962** ±0.0097	**0.3482** ±0.0035
4B	Unadapted text	0.0861 ±0.0124	0.1780 ±0.0059	**0.1300** ±0.0095	**0.1127** ±0.0048
	Unadapted plots	**0.0378** ±0.0111	**0.1590** ±0.0080	0.3860 ±0.0100	0.2537 ±0.0040
9B	Unadapted text	0.4843 ±0.0145	0.2990 ±0.0033	**0.4502** ±0.0092	**0.4586** ±0.0088
	Unadapted plots	**0.2418** ±0.0136	**0.2364** ±0.0046	0.7261 ±0.0095	0.6257 ±0.0071