OPTIMIZING VISUALIZATION FOR TIME SERIES ANALYSIS VIA VISION-EMBEDDED BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series analysis (TSA) is fundamental across many real-world domains, supporting forecasting, anomaly detection, and decision making. Recent vision-based approaches convert time-series data into images and apply models for TSA. However, these methods typically use fixed or manually selected visualization settings, even though different visual presentations may lead to different model performance. Therefore, we introduce Visual Presentation Optimization (VPO), a new formulation that casts time-series visualization design as an optimization problem to improve vision-based TSA performance for a given task and model. To efficiently explore the large combinatorial configuration space, we propose Vision-embedded Bayesian Optimization (VeBO), which incorporates visual embeddings into a deep kernel learning-based Bayesian Optimization surrogate to predict performance from visualization settings and guide the search. Extensive experiments show that optimizing visual presentations without additional training improves performance efficiently across multiple TSA tasks and models, highlighting VPO as an effective, model-agnostic approach for vision-based time-series understanding. *Code will be published upon acceptance.*

1 INTRODUCTION

Time-series Analysis (TSA) aims to understand temporal patterns and extract meaningful information from sequential data. Accurate TSA is critical across diverse real-world applications, including industrial monitoring (Liu et al., 2020), healthcare (Piccialli et al., 2021), and finance (Zhu & Shasha, 2002), where identifying temporal structures supports forecasting, anomaly detection (Liu et al., 2024b), and decision-making (Mahalakshmi et al., 2016). Given its importance, a wide range of approaches have been developed for TSA, ranging from classical statistical models (Pegels, 1969; Box et al., 2015) to modern deep learning methods (Zhou et al., 2021; Wu et al., 2021).

More recently, Large Language Models (LLMs) have emerged as a promising option for TSA (Gruver et al., 2023; Zhou et al., 2023; Zhang et al., 2024), as they have demonstrated generality across a wide range of tasks and flexibility with language understanding. Yet, directly feeding raw numerical time-series data into LLMs has shown limited effectiveness (Zeng et al., 2023; Tan et al., 2024; Merrill et al., 2024), since numerical signals often do not align well with the representations learned from structured perceptual modalities, such as language (Jin et al., 2024).[1] Inspired by how humans interpret time-series data (Pettersson, 1993; Dondis, 1974; Shneiderman, 2003), recent studies instead convert time-series signals into visual representations and analyze them using vision-based TSA models (Li et al., 2023; Chen et al., 2024; Naiman et al., 2024; He et al., 2025; Xu et al., 2025). Visualization can make temporal structures, such as trends, seasonality, and irregular patterns, more perceptible than raw numerical sequences.

While existing vision-based TSA work focuses on developing models that operate on visualized time-series data for downstream tasks, the visualization design itself has received little attention. In most existing approaches, visualization design choices such as axis scaling, window length, resolution, or plotting style, are fixed or manually selected. Meanwhile, recent studies reveal that modern foundation models can be highly sensitive to visual presentation (Mahbub et al., 2025; Guo et al., 2024; Rosenfeld et al., 2025; Verma et al., 2025; Jung-Mok et al., 2025), suggesting that how time-series data are visualized may significantly influence model performance. These observations raise

[1] LLMs often produce incorrect comparisons, such as $9.11 > 9.9$ due to token design (Yang et al., 2025).

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.

an important question: *can the visual presentation of time-series data be systematically optimized to improve model understanding?*

To this end, we introduce Visual Presentation Optimization (VPO), a new paradigm that formulates visualization as a configuration optimization problem for improving vision-based TSA. This paradigm shifts the focus from model-centric improvement to input-centric optimization, opening a new direction for vision-based TSA. Rather than modifying model architectures or training procedures, VPO optimizes visualization configurations to identify task- and model-specific representations of time-series data that facilitate interpreting temporal patterns.

However, solving the VPO problem is challenging because the visualization design space can be large and combinatorial. It contains many configurable options, and the number of possible configurations grows exponentially as additional parameters are introduced, making exhaustive exploration intractable. To efficiently identify effective visualization designs, we propose Vision-embedded Bayesian Optimization (VeBO), a framework that leverages Bayesian Optimization (BO) to guide the search. Specifically, VeBO adopts BO with a Deep Kernel Learning (DKL)-based surrogate model as a principled search strategy (Ranković & Schwaller, 2025; 2023). VeBO models the relationship between plot hyperparameters and task performance by taking the plot hyperparameters as input and predicting the performance of the task model under the corresponding plot. To fully exploit the visual nature induced by plot hyperparameters, VeBO utilizes visual embeddings from a pretrained Multimodal Large Language Model (MLLM), enabling it to leverage its visual representation capability when predicting task performance. Guided by these predictions, VeBO iteratively selects promising visualization configurations to evaluate, efficiently exploring the space and identifying effective visual presentations without any model modification or fine-tuning.

Through extensive experiments, we demonstrate that optimizing the visual presentation of time-series data alone can substantially improve performance across multiple downstream tasks, without retraining the task model. We also show that VeBO effectively searches the visual presentation space to identify high-performing configurations, which vary across different tasks and models. Since VeBO does not depend on a specific task or model architecture, it can be broadly applied to a wide range of vision-based TSA systems.

- **Visual Presentation Optimization.** We introduce Visual Presentation Optimization (VPO), a new formulation that casts time-series visualization design as an optimization problem. For this, we define the objective function that maximizes task model performance over plot configuration parameters, enabling VPO to discover visual presentations optimal to the task and the model.
- **Vision-embedded Bayesian Optimization.** We propose Vision-embedded Bayesian Optimization (VeBO), a framework that incorporates MLLM-derived visual embeddings into Bayesian Optimization for efficient exploration of visualization configurations.
- **Task- and Model-Agnostic Framework.** We demonstrate that optimizing visualization alone, without retraining the model, significantly improves performance across multiple TSA tasks and models. This shows that VeBO is task- and model-agnostic, making it applicable to diverse vision-based TSA tasks and models.

2 Related Work

Numerical data-based time-series analysis. Time-series analysis (TSA) has been extensively studied for tasks such as anomaly detection, classification, forecasting, and question answering. Time-series data are fundamentally numerical. Existing approaches address TSA by directly modeling raw values, ranging from classical statistical methods (Pegels, 1969; Box et al., 2015; Hamzaçebi, 2008; Liu et al., 2008) to deep learning models (Sen et al., 2019), including autoencoders (Audibert et al., 2020; Wong et al., 2022) and transformer architectures (Tuli et al., 2022; Xu et al., 2022; Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022). More recently, researchers have explored leveraging the autoregressive behavior of LLMs for time-series tasks (Gruver et al., 2023; Zhou et al., 2023; Zhang et al., 2024; Liu et al., 2024c;a; Alnegheimish et al., 2024), often by transforming numerical sequences into structured textual representations. For example, LLMTime (Gruver et al., 2023) reformulates forecasting as next-token prediction by tokenizing numerical values as text and using LLMs in a zero-shot manner. Commonly, such approaches remain value-centric in that they process raw numerical sequences directly, where concerns have been raised about the suitability of LLM

mechanisms for numerical time-series, such as potential information loss caused by the self-attention mechanism (Zeng et al., 2023) and limited effectiveness on raw numerical inputs (Merrill et al., 2024; Tan et al., 2024). These limitations motivate us to incorporate additional modalities to better capture temporal structure (Liu et al., 2025a; Chen et al., 2024; Liu et al., 2025c; Hyeon-Woo et al., 2024; Jung-Mok et al., 2025).

Vision-based approach in time-series analysis. Motivated by findings from cognitive science suggesting that humans often perceive temporal patterns more effectively through visual representations than raw numbers (Pettersson, 1993; Dondis, 1974; Shneiderman, 2003), recent studies in TSA have increasingly adopted vision-based approaches that transform time-series into images (Li et al., 2023; Liu et al., 2025a; Daswani et al., 2024; Liu et al., 2025b; Yang et al., 2024; Shen et al., 2025; Zhong et al., 2025; Kong et al., 2025b; Hao et al., 2024; Ortigossa et al., 2025). In particular, several studies (Li et al., 2023; Liu et al., 2025a; Daswani et al., 2024; Yang et al., 2024; Shen et al., 2025; Kong et al., 2025b; Aigner et al., 2007) show that visualization helps models better capture patterns in time-series data. A wide range of visual representations have been explored for time-series data (Ortigossa et al., 2025; Fang et al., 2020; Havre et al., 2000; Van Wijk & Van Selow, 1999) because different visualization designs emphasize different temporal characteristics. Conventional representations such as line and bar charts provide intuitive views of temporal trends and value changes (Van Wijk & Van Selow, 1999; Ortigossa et al., 2025), while more specialized representations have also been developed to reveal particular temporal structures (Aigner et al., 2007; Havre et al., 2000). However, most prior work has largely focused on human perception and visual exploration (Javed et al., 2010; Heer et al., 2009), leaving visual presentation underexplored in vision-based time-series analysis. To address this gap, our work formulates visual presentation selection as an optimization problem.

Bayesian optimization. Bayesian Optimization (BO) is an efficient framework for optimizing black-box functions that are expensive to evaluate. Rather than exhaustively searching the entire space, BO fits a probabilistic surrogate model (*e.g.*, Gaussian Process (GP)) and uses an acquisition function to select the next candidate to evaluate (Jones et al., 1998; Shahriari et al., 2015). This enables sample-efficient optimization with a small number of iterations. As optimization problems have become more complex and high-dimensional, Deep Kernel Learning (DKL) has emerged as a way to increase the expressiveness of GP surrogates (Wilson et al., 2016). In DKL, neural networks such as variational autoencoders (VAEs) or vision transformers (ViTs) commonly serve as feature extractors, and a GP is applied to the learned embeddings for BO (Gómez-Bombarelli et al., 2018; Maus et al., 2022; Picard et al., 2024). More recently, the strong representational capabilities of LLMs have motivated their use as feature extractors within BO (Ranković & Schwaller, 2023; Nguyen et al., 2024; Ranković & Schwaller, 2025). However, these methods typically rely on text templates to describe candidate solutions and may be less suitable for optimizing visual presentations. In contrast, we use visual embeddings extracted from an MLLM within a DKL-based BO framework. By deriving embeddings directly from rendered time-series images, our surrogate can capture similarity in the visual space, yielding a more appropriate approach for visual presentation optimization.

3 METHOD

3.1 PRELIMINARY

Bayesian Optimization (BO) (Frazier, 2018) is designed to optimize an objective function f that is expensive to evaluate. Instead of evaluating f over the entire search space, BO constructs a surrogate model $\hat{f}$ to approximate f, enabling efficient exploration while minimizing the number of costly evaluations. A common choice for the surrogate model is a Gaussian Process (GP):

$$\hat{f} \sim \mathcal{GP}(\mu(\cdot), k(\cdot, \cdot)),$$

where $\mu(\cdot)$ denotes the mean function and $k(\cdot, \cdot)$ is the kernel function. To better model complex input representations, we adopt Deep Kernel Learning (DKL) (Wilson et al., 2016) with a feature transformation $\psi(\cdot)$, which maps an input v into a feature space before applying the GP kernel:

$$k_\psi(v, v') = k_{\text{base}}(\psi(v), \psi(v')),$$

where $k_{\text{base}}(\cdot, \cdot)$ denotes a base kernel (*e.g.*, Matérn kernel). This allows the GP to operate in a transformed feature space and capture richer similarity structures than those available in the raw input space.

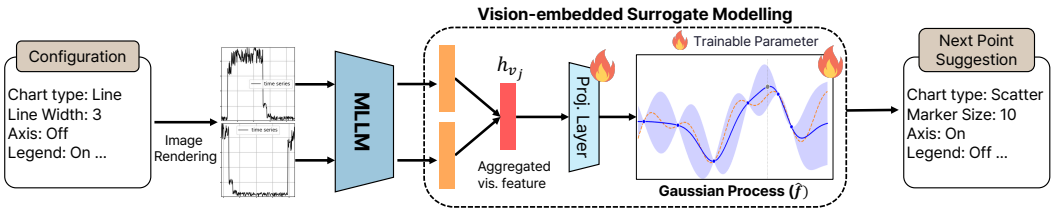

Figure 1: **Vision-embedded Bayesian Optimization (VeBO).** A candidate visual presentation (*e.g.*, line plot) is evaluated by a frozen vision-based task model to obtain performance. Simultaneously, presentation-level visual features are extracted via an MLLM and aggregated to construct a vision-embedded surrogate model. In this process, a projection layer enables Deep Kernel Learning, and BO iteratively suggests the next visualization point to evaluate.

Given the observed data $\mathcal{D} = \{(v_j, R_{v_j})\}_{j=1}^{N}$, where N denotes the number of evaluated samples and R_{v_j} denotes the observed function value of f at v_j, the surrogate model is trained by maximizing its marginal likelihood. The next candidate is then selected using an acquisition function that balances exploration and exploitation. The next candidate is inserted into $\mathcal{D}$, and GP is updated. This procedure is iteratively repeated until a satisfactory solution is found.

3.2 Problem Formulation

In this section, we introduce a new task, *Visual Presentation Optimization (VPO)*, and formulate this as an optimization problem. For this, We first define the design search space for visual presentations v and then introduce a black-box function f that maps each visual presentation to its corresponding performance R_v.

Search space of visual presentations. Given the time-series input space T and the rendered image space I, a *visual presentation* is defined as a rendering function:

$$g(\cdot; v) : T \to I,$$

where $g(\cdot; v)$ denotes a plot rendering program that renders time-series data into an image according to the visual presentation v. The rendered image I_v is provided as input to a task model $M(\cdot)$, which generates predictions. The performance score associated with the visual presentation v is then computed based on these predictions. To define the search space, we introduce a visual presentation hyperparameter, θ, where each θ^{ℓ} denotes the selected hyperparameter option for category $\ell = \{type, color, ..., label\}$. These hyperparameters include configuration options, such as plot type, color, and some visual elements (*e.g.*, legends, or labels). Each hyperparameter θ takes values from its corresponding domain Θ, defining the possible configurations of a visual presentation.[2] The search space S is then defined as the product of all parameter domains, representing all possible combinations of visual presentation.

$$v = (\theta^{type}, \theta^{color}, \theta^{width}, \ldots, \theta^{label}), \quad \theta^{\ell} \in \Theta^{\ell},$$
$$S = \Theta^{type} \times \Theta^{color} \times \Theta^{width} \times \cdots \times \Theta^{label}.$$

Thus, S forms a search space composed of diverse visual presentation attributes.

Visual Presentation Optimization. The goal of VPO is to find the best visual presentation v^* that maximizes the task model M's performance at the evaluation dataset $\{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the time-series instance and the y_i is the corresponding task ground truth. Accordingly, we define the objective function for visual presentation optimization as:

$$v^* = \arg\max_{v \in S} \frac{1}{N} \sum_{i=1}^{N} R\big(M(I_{v,i}), y_i\big) = \arg\max_{v \in S} R_v, \tag{1}$$

where $R(\cdot, \cdot)$ denotes the evaluation metric between the model prediction and the ground truth, R_v the model performance measured when using the visual presentation v, and $I_{v,i} = g(x_i; v)$ represents the rendered image of the time-series instance x_i with the visual presentation v.

[2]Detailed visual presentation options are presented in Sec. 4.

3.3 VISION-EMBEDDED BAYESIAN OPTIMIZATION

In this section, we propose *Vision-embedded Bayesian Optimization (VeBO)*, a Bayesian optimization (BO) framework for solving the VPO problem. Evaluating the VPO objective is expensive because each evaluation requires running model inference over the dataset. Moreover, since the objective can only be obtained through such evaluations, has no explicit analytical form, and is non-differentiable, we formulate VPO as a black-box optimization problem. VeBO addresses this problem by combining BO with visual features to efficiently explore the structured space of visual presentations.

Overall pipeline. VeBO aims to find the best visual presentation v^* by repeatedly exploring the defined search space S. For this, VeBO conducts 4 steps: (1) initialization, (2) model evaluation, (3) vision-embedded surrogate model update, and (4) next visual presentation selection. Figure 1 illustrates the overall pipeline of VeBO. VeBO begins by randomly selecting a visual presentation $v_0 \in S$. (1) For visual presentation v_0, each time-series instance x_i from the evaluation dataset is rendered to image $I_{v_0,i}$ via the plot rendering program $g(x_i; v_0)$. (2) The rendered images $I_{v_0} = \{I_{v_0,1}, I_{v_0,2}, \ldots, I_{v_0,N}\}$ are fed into the task model M to measure the performance R_{v_0}. (3) To update the surrogate model, visual embeddings are extracted from the rendered images I_{v_0} using a Multimodal Large Language Model (MLLM). The instance-level embeddings are aggregated and transformed through a projection layer P to obtain the surrogate input z_{v_0}. The surrogate model $\hat{f}$ is then updated using the observed dataset $\mathcal{D}$ with z_{v_0}. Afterward, (4) the acquisition function (*e.g.*, Expected Improvement) is computed over the search space, and the next candidate visual presentation v_1 is selected based on the acquisition value, balancing exploration and exploitation. VeBO repeats steps (2)–(4) for a predefined number of iterations K. The overall iterative procedure is described in Appendix (Algorithm 1).

Visual embedding extraction. To enable effective surrogate modeling at iteration j, VeBO extracts the visual embeddings from the rendered images I_{v_j}. In particular, it utilizes pretrained MLLM (*e.g.*, Qwen2.5-VL)'s hidden state h_{v_j} to encode the style of the visual presentation. Given the rendered image $I_{v_j,i}$ obtained by applying visual presentation v_j of time-series instance x_i, we extract the last layer's hidden state following (Jiang et al., 2024):

$$h_{v_j,i} = \texttt{MLLM}(I_{v_j,i}), \tag{2}$$

where `MLLM` denotes the MLLM used to extract visual embeddings from rendered images. Since the evaluation dataset contains N time-series instances $\{x_i\}_{i=1}^{N}$, we aggregate the embeddings extracted from each instance x_i to obtain a single representation that captures the visual style. Specifically, we compute the visual style embedding by averaging the instance-level embeddings:

$$h_{v_j} = \mathcal{A}\big(\{h_{v_j,i}\}_{i=1}^{N}\big), \tag{3}$$

where $\mathcal{A}(\cdot)$ denotes an averaging aggregation over the instance-level embeddings.

Vision-embedded surrogate modeling. Using the aggregated visual feature h_{v_j} for visual presentation v_j, we construct a surrogate model for BO over the visual presentation space S. To enable flexible modeling while preserving the semantic structure of MLLM, we introduce a lightweight feature transformation as a projection layer P. Then, we have an input z_{v_j} of the surrogate model as:

$$z_{v_j} = \psi(h_{v_j}) = \mathrm{P}(h_{v_j}), \tag{4}$$

To incorporate the projected visual representations into the surrogate model, we adopt a DKL formulation. Specifically, we define a deep kernel for the surrogate model:

$$k\big(z_{v_j}, z'_{v_j}\big) = k_{base}\big(\mathrm{P}(h_{v_j}), \mathrm{P}(h_{v'_j})\big), \tag{5}$$

where $k_{base}(\cdot,\cdot)$ denotes the base kernel function, and we adopt the Matérn 5/2 kernel. Such a surrogate model $\hat{f} \sim \mathcal{GP}(0, k(z_{v_j}, z'_{v_j}))$ estimates our visual presentation evaluation function by training on observed data $\mathcal{D} = \{(v_j, R_{v_j})\}_{j=0}^{K}$.

Next visual presentation selection. To select the next visual presentation, we use Expected Improvement (EI) as the acquisition function α. At iteration j, the next presentation is selected as

$$v_{j+1} = \arg\max_{v \in \mathcal{S}} \alpha(v). \tag{6}$$

Table 1: **Quantitative Result.** To show the effectiveness of our proposed method (VeBO), we present results on the TimeMQA (Kong et al., 2025a) dataset. NMSE_{v1} and NMSE_{v2} denote the NMSE measured on different versions of the forecasting dataset. T/F and MCQ refer to true-false and multiple-choice questions, respectively, while open-ended QA is evaluated using accuracy. VeBO increases the performance across multiple tasks and models without any fine-tuning.

Model	VeBO	TimeMQA										
		Anomaly Detection			Classification			Forecasting		Open-Ended QA		
		Precision(↑)	Recall(↑)	F1 (↑)	Precision(↑)	Recall(↑)	F1(↑)	NMSE_{v1}(↓)	NMSE_{v2}(↓)	T/F	MCQ	Overall
GPT-5-mini	✗	0.512	0.512	0.508	0.305	0.277	0.258	10.947	5.523	0.711	0.673	0.690
	✓	**0.636**	**0.616**	**0.604**	**0.387**	**0.351**	**0.348**	**3.647**	**3.338**	**0.778**	**0.691**	**0.730**
Gemini-2.5-Flash-Lite	✗	0.288	0.446	0.319	0.296	0.289	0.234	12.44	9.77	0.731	0.594	0.646
	✓	**0.513**	**0.511**	**0.487**	**0.316**	**0.303**	**0.255**	**5.35**	**8.07**	**0.742**	0.594	**0.651**
LLaMA3.2-11B-Vision	✗	0.422	0.461	0.384	0.053	**0.249**	0.084	18.65	394.81	0.623	0.477	0.533
	✓	**0.525**	**0.525**	**0.523**	**0.162**	0.248	**0.182**	**18.11**	**392.72**	**0.642**	**0.487**	**0.546**
Qwen2.5-VL-7B	✗	0.335	0.471	0.334	0.160	0.250	0.188	19.73	35.95	**0.735**	0.584	0.642
	✓	**0.535**	**0.535**	**0.534**	**0.210**	**0.294**	**0.237**	**17.85**	**26.21**	0.731	**0.603**	**0.652**

To efficiently search the acquisition space, we employ a Branch-and-Bound (BnB) strategy over the discrete configuration space (Lizotte, 2008; Wang et al., 2014). Each node N contains a subset $\mathcal{S}_N \subseteq \mathcal{S}$ of visual presentation candidates, and is further divided based on categorical attributes (*e.g.*, plot type, color). For each node, we sample at most m candidates and evaluate its optimistic estimate:

$$\widehat{U}_j(N) = \max_{v \in \widetilde{\mathcal{S}}_N} \alpha(v) + \lambda \max\left(\left| \max_{v \in \widetilde{\mathcal{S}}_N} \alpha(v) \right|, \epsilon_{\text{mach}} \right), \tag{7}$$

where $\widetilde{\mathcal{S}}_N \subseteq \mathcal{S}_N$ is the sampled subset and λ controls the margin. During the search, we maintain the best observed acquisition value $\alpha^\star$ and prune a node when $\widehat{U}_j(N) < \alpha^\star$. This progressively eliminates less promising regions, reducing acquisition evaluations without exhaustively searching the full visual presentation space.

4 Experiments

We evaluate the proposed VPO across diverse time-series analysis tasks, including anomaly detection, classification, open-ended QA, and forecasting. We also conduct ablation studies on VeBO, including comparisons to text-embedded BO (Ranković & Schwaller, 2025).

4.1 Experiment Setup

Search space construction. We defined a VPO's search space with six parameters: plot type, color, width, axis visibility, legend visibility, and canvas size. The space includes 10 plot types (*i.e.*, bar, boxplot, heatmap, histogram, line, pie, polar-area, scatter, spiral, and streamgraph), selected based on diverse time-series visualization studies (Havre et al., 2000; Ortigossa et al., 2025; Fang et al., 2020). 1,000 colors sampled from the color spectrum, three width values (1, 2, 3), two axis visibility settings, two legend visibility settings, and four canvas sizes, resulting in 480,000 configurations. Detailed search space explanation is in Appendix A.1.

Dataset and baselines. To show the effectiveness of the proposed task (VPO), we evaluated our method on TimeMQA (Kong et al., 2025a), which includes diverse time-series tasks: anomaly detection, classification, forecasting, and open-ended QA. We conducted experiments using diverse closed and open-source MLLMs: GPT-5-mini (Singh et al., 2025), Gemini-2.5-Flash-Lite (Comanici et al., 2025), Qwen2.5-VL-7B (Bai et al., 2025) and LLaMA3.2-11B-Vision (Grattafiori et al., 2024). For hyperparameter optimization baselines, we adopted 4 methods: Grid search (Larochelle et al., 2007), Random Search (Bergstra & Bengio, 2012), conventional Bayesian Optimization (BO) (Frazier, 2018), GoLLuM (Ranković & Schwaller, 2025). For GoLLuM, we utilized Qwen2.5-3B to extract text features, whereas VeBO uses Qwen2.5-VL-3B to extract visual features.

Metrics. For the anomaly detection and classification task, we report the Precision, Recall, and F1 scores. For the forecasting task, we report Normalized Mean Squared Error (NMSE), and for open-ended QA, we report the accuracy of both true/false questions and multiple-choice questions. Same metrics are used as task-specific optimization objective during optimization.

Table 2: **Comparison with baselines.** We report results for each VLM on TimeMQA. indicates the best performance for each model on each metric, while indicates the second-best performance.

Baselines	Model	TimeMQA										
		Anomaly Detection			Classification			Forecasting		OpenQA		
		Precision(↑)	Recall(↑)	F1(↑)	Precision(↑)	Recall(↑)	F1(↑)	N_{v1}(↓)	N_{v2}(↓)	T/F(↑)	MCQ(↑)	Overall(↑)
Grid	LLaMA3.2-11B-Vision	0.483	0.490	0.433	**0.172**	**0.261**	**0.193**	23.40	394.70	0.623	**0.485**	**0.537**
	Qwen2.5-VL-7B	0.499	0.499	0.460	0.198	0.277	0.221	18.25	29.64	**0.723**	0.584	0.637
Random	LLaMA3.2-11B-Vision	0.509	0.509	0.498	0.160	0.240	0.157	**21.91**	395.52	0.608	0.470	0.523
	Qwen2.5-VL-7B	0.501	0.501	0.473	0.192	**0.312**	**0.233**	18.11	89.60	0.719	**0.589**	**0.639**
BO	LLaMA3.2-11B-Vision	0.453	0.476	0.403	**0.170**	**0.267**	0.171	23.14	479.90	**0.677**	0.451	**0.537**
	Qwen2.5-VL-7B	0.498	0.499	0.441	0.201	0.271	0.218	18.01	30.92	0.700	0.582	0.627
GoLLuM	LLaMA3.2-11B-Vision	**0.510**	**0.515**	**0.514**	0.162	0.246	0.176	22.68	**393.82**	0.638	0.451	0.523
	Qwen2.5-VL-7B	**0.559**	**0.559**	**0.559**	**0.218**	0.272	0.223	**17.74**	**28.98**	0.712	**0.589**	0.636
VeBO (Ours)	LLaMA3.2-11B-Vision	**0.525**	**0.525**	**0.523**	0.162	0.248	**0.182**	**18.11**	**392.72**	**0.642**	**0.487**	**0.546**
	Qwen2.5-VL-7B	**0.535**	**0.535**	**0.534**	**0.210**	**0.294**	**0.237**	**17.85**	**26.21**	**0.731**	**0.603**	**0.652**

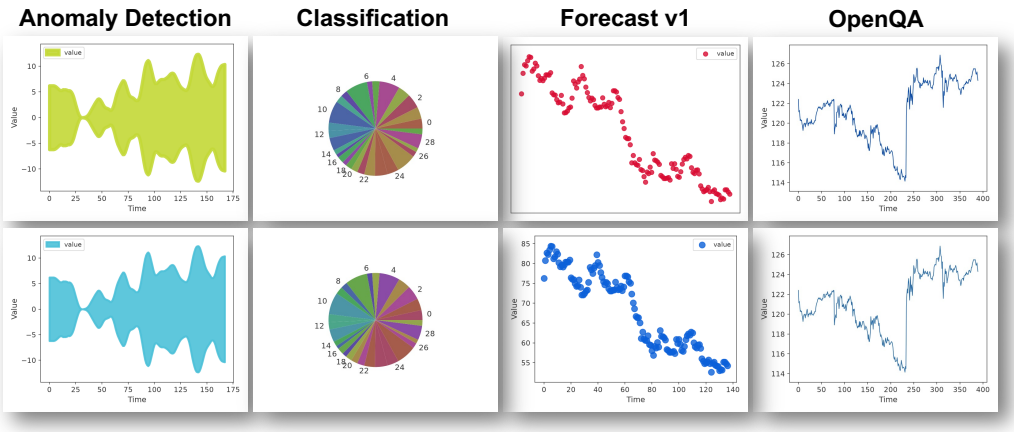

Figure 2: **Qualitative results.** We present two representative presentations from the top-3 visual presentations found by VeBO on TimeMQA (Kong et al., 2025a) using Qwen2.5-VL-7B. Different tasks favor distinct visualization designs, showing the task-specific preferences in visual presentation.

4.2 Experiment Result

Quantitative result. We evaluate VeBO across diverse time-series tasks in TimeMQA. Table 1 compares performance using the default time-series visualizations with performance using visual presentations selected by VeBO under the VPO objective. For all models, optimized visual presentations outperform the default lineplot visualization (*e.g.*, the default plot setting of `Matplotlib.pyplot.plot`) across most tasks. These results suggest that model performance strongly depends on how time-series data are visually presented and provided as inputs. In summary, input-level optimization alone improves performance in a zero-shot setting without modifying or fine-tuning the model, and the gains generalize across diverse time-series tasks.

Comparison with baselines. To demonstrate the effectiveness of VeBO, we compare VeBO with Grid search, Random search, standard BO, and GoLLuM on two open-source VLMs. Table 2 shows that VeBO achieves competitive performance across both models and diverse TSA tasks. For Qwen2.5-VL-7B, VeBO achieves strong performance on classification, Forecasting V2, and OpenQA, while remaining competitive on anomaly detection and Forecasting V1. For LLaMA3.2-11B-Vision, VeBO performs strongly on anomaly detection, forecasting, and OpenQA, although other baselines outperform it on some metrics, showing that VeBO remains competitive across heterogeneous tasks and models and supports the use of visual representations as a useful prior for navigating the visual presentation space.

Qualitative result. Figure 2 shows representative visual presentations identified by VeBO for different TSA tasks. As shown, the preferred visual presentations vary across tasks: anomaly detection

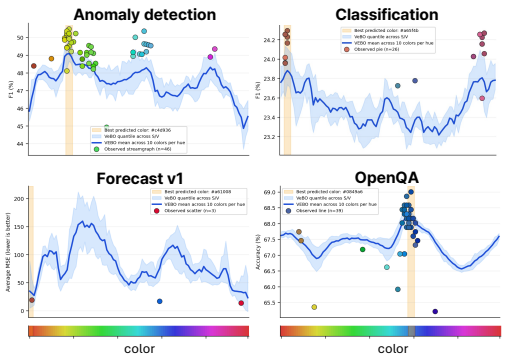

Figure 3: **Predicted performance across colors.** We report how the performance varies along the color per discovered best plot, using the VeBO's prediction.

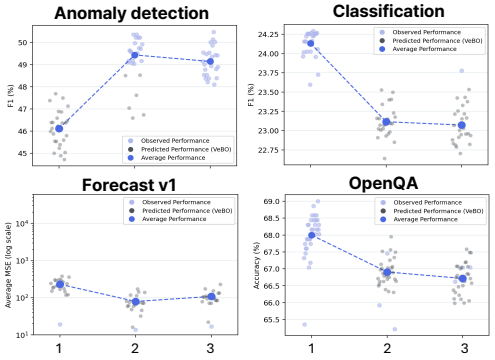

Figure 4: **Performance across width values.** For each task, we report performance as the width varies for the selected plot type using Qwen2.5-VL-7B.

favors a streamgraph, classification favors a polar-area plot, forecasting favors a scatter plot, and OpenQA favors a line plot. For anomaly detection, area-based representations visually emphasize local changes in the temporal signal, which may provide salient cues for distinguishing anomalous regions from normal patterns. Classification shows a distinct preference for circular representations such as polar-area and pie plots. Rather than directly encoding the sequential (x, y, z) structure through line or scatterplot, VPO reorganizes the observations into alternative spatial layouts that may expose useful visual cues for classification. Forecasting favors scatter plots that directly expose the temporal distribution, whereas OpenQA, which requires answering diverse questions about the time series, favors conventional line plots that preserve the overall temporal trajectory. Interestingly, top-performing presentations within the same task often share the same plot structure while differing in other visual attributes such as color. These results suggest that there is no single visual presentation that is universally optimal across TSA tasks, showing the importance of task-specific visual presentation optimization.

4.3 Analysis and Ablation Studies

In this section, we analyze the visual presentation configurations and conduct ablation studies on the embedding modalities used in our framework. First, we analyze the sensitivity of model performance to visual presentation attributes. We then compare text-based and visual embeddings in terms of optimization performance and analyze the structure of their representation spaces to understand how different embeddings influence surrogate modeling and search behavior.

Visual preference analysis. We analyze how model performance varies across visual presentation attributes, including plot type, color, and width at Qwen2.5-VL-7B. VeBO's deep kernel-based surrogate model provides a dense estimate of model performance over the visual presentation space, enabling analyses beyond the sparsely evaluated configurations. We leverage these predictions to characterize the model's visual preferences for individual attributes, including color and width.

Figure 3 shows the predicted performance over color for each task, where the line and shaded region indicate the mean and 10th-90th percentile predictions, respectively, and the points denote observed evaluations. We observe distinct task-dependent patterns rather than a universally preferred color. For anomaly detection, VeBO predicts the highest performance in the yellow and green region, while additional local peaks appear around cyan and magenta. Classification and forecasting exhibit different preferences, while at OpenQA VeBO predicts a pronounce performance peak in the blue region. This suggests that the role of color depends on the visual information required by each task, rather than reflecting a universal preference of the VLM. Figure 4 shows that width preferences also vary across tasks. Anomaly detection and forecasting favor width 2, whereas classification and OpenQA favor width 1. This suggests that even a low-level visual attribute such as width can meaningfully affect performance, with no single width being optimal across tasks.

Analysis of embedding geometry. We examine whether visual representations provide a stronger prior for modeling visualization performance than text representations. For a controlled compar-

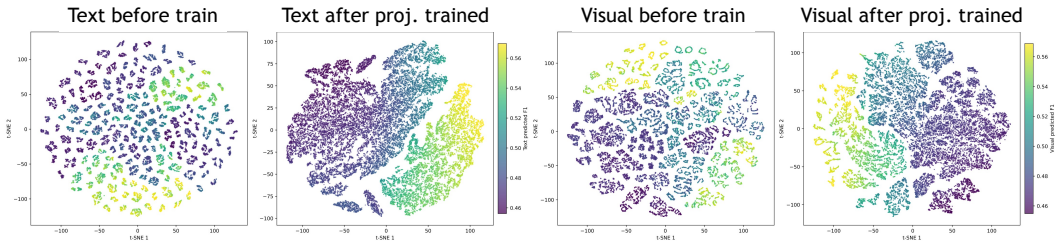

Figure 5: **Visualization of embedding spaces.** We visualize embedding spaces with each surrogate model's top-100 performing configurations highlighted. High-performing configurations are scattered in both the raw embedding space before projection and the textual embedding space. In contrast, visual embeddings form a compact cluster, suggesting a more suitable space.

ison, both text, and visual surrogates are trained using the same 50 randomly sampled configurations, and evaluated on an independent set of 50 configurations with no training overlap. As shown in Figure 5, the representations before projection are primarily organized by their original modality-dependent structure. After learning the projection, configurations with similar surrogate-predicted F1 scores become more smoothly organized in both spaces, indicating that the projection encodes performance-relevant structure. The quantitative results in Table 3 further show that the visual representation provides a stronger performance prior: it achieves a higher held-out R^2, higher Pearson and Spearman correlations, and lower MAE than the text representation. It shows that visual embeddings better preserve performance-relevant characteristics of visual presentation, while projection learning further encodes into performance-aware embedding spaces.

Ablation study of Branch and Bound. We additionally examine how effective the branch-and-bound method is for the acquisition function, at Table 4. Without BnB, VeBO exhaustively evaluates the acquisition values of all 480K candidates at each iteration, which leads to high inefficiency. In contrast, BnB avoids unnecessary acquisition evaluations by pruning candidates that cannot improve the current solution. As a result, BnB substantially reduces acquisition-value evaluations while maintaining comparable performance across all tasks.

Table 3: **Surrogate model performance.**

Embedding	R^2 (↑)	MAE (↓)	Pearson (↑)	Spearman (↑)
Text	0.757	0.0146	0.881	0.770
Visual	**0.803**	**0.0124**	**0.902**	**0.860**

Table 4: **Ablation study on Branch & Bound.**

Ours	Anomaly Detection				Classification			
	P (↑)	R (↑)	F1 (↑)	# Acq.	P (↑)	R (↑)	F1 (↑)	# Acq.
✗	0.498	0.498	0.498	480k	0.211	0.304	0.242	480k
✓	0.535	0.535	0.534	156.9k	0.210	0.294	0.237	59.1k

Ours	Forecasting			Open-Ended QA			
	NMSE_{v1} (↓)	NMSE_{v2} (↓)	# Acq.	T/F (↑)	MCQ (↑)	Overall (↑)	# Acq.
✗	16.64	27.72	480k	0.731	0.594	0.646	480k
✓	17.85	26.21	2.3k	0.731	0.603	0.652	7.4k

5 Conclusion

In this work, we introduce **Visual Presentation Optimization (VPO)**, a new perspective of time-series analysis that focuses on optimizing the visual presentation of time-series data rather than modifying the task-specific model. While prior works mainly focus on improving model architectures or model training, we show that optimizing visual presentation alone can significantly improve model performance without modifying the model itself. To systematically frame this problem, we formulate VPO as a black-box optimization problem and propose **Vision-embedded Bayesian Optimization (VeBO)**, which utilizes visual embeddings extracted from a pretrained MLLM for surrogate modeling, enabling the surrogate model to capture performance-relevant similarities between different visual presentations. Our results show that substantial performance gains can be achieved without fine-tuning the task model, highlighting the effectiveness of VPO across diverse time-series tasks. We also show that VeBO opens a broader direction for leveraging pretrained visual representations to guide BO's surrogate modeling, showing the effectiveness compared to the text representation for VPO.

Limitations. In this work, we evaluate the proposed method over a predefined visual presentation space, and thus its effectiveness may depend on the diversity of the included visual attributes. Additionally, although VeBO reduces the need for exhaustive search, optimization still requires repeated VLM evaluations, which may become costly for larger presentation spaces.

AI USE STATEMENT

In this work, we used generative AI tools for assisting on experiments (e.g., summarizing the results), assist with language refinement. We have not used generative AI tools to generate synthetic data sets, develop theoretical models or conceptual frameworks, formulate mathematical claims, or making core algorithms. All AI-assisted text was reviewed and revised by the authors, and AI-assisted code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the overall VPO pipeline with VeBO in Sec. 3, including visual embedding extraction, surrogate modeling, and acquisition. Sec. 4 provides the experimental setup, including the visual presentation search space, datasets, and competing methods. For reproducibility, Appendix A.1 further specifies the construction of the visual presentation space and all attribute combinations Additional implementation details, including the feature extractor for deep kernel learning, embedding extraction and projection, task evaluation models, prompts, and the overall algorithm, are provided in Appendix A.2 and Appendix A.3.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visualizing time-oriented data—a systematic view. *Computers & Graphics*, 31(3):401–409, 2007.

Sarah Alnegheimish, Linh Nguyen, Laure Berti-Equille, and Kalyan Veeramachaneni. Can large language models be anomaly detectors for time series? In *2024 IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA)*. IEEE, 2024.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3395–3404, 2020.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

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.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters. *arXiv preprint arXiv:2408.17253*, 2024.

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

Mayank Daswani, Mathias MJ Bellaiche, Marc Wilson, Desislav Ivanov, Mikhail Papkov, Eva Schnider, Jing Tang, Kay Lamerigts, Gabriela Botea, Michael A Sanchez, et al. Plots unlock time-series understanding in multimodal models. *arXiv preprint arXiv:2410.02637*, 2024.

Donis A Dondis. *A primer of visual literacy*. Mit Press, 1974.

Yujie Fang, Hui Xu, and Jie Jiang. A survey of time series data visualization research. In *IOP Conference Series: Materials Science and Engineering*, volume 782, pp. 022013. IOP Publishing, 2020.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/27e8e17134dd7083b050476733207ea1-Paper.pdf.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

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

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in neural information processing systems*, 36:19622–19635, 2023.

Grace Guo, Jenna Jiayi Kang, Raj Sanjay Shah, Hanspeter Pfister, and Sashank Varma. Understanding graphical perception in data visualization through zero-shot prompting of vision-language models. *arXiv preprint arXiv:2411.00257*, 2024.

Coşkun Hamzaçebi. Improving artificial neural networks' performance in seasonal time series forecasting. *Information Sciences*, 178(23):4550–4559, 2008.

Jianing Hao, Zhuowen Liang, Chunting Li, Yuyu Luo, Jie Li, and Wei Zeng. Vistr: Visualizations as representations for time-series table reasoning. *arXiv preprint arXiv:2406.03753*, 2024.

Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: Visualizing theme changes over time. In *IEEE symposium on information visualization 2000. INFOVIS 2000. proceedings*, pp. 115–123. IEEE, 2000.

Zelin He, Sarah Alnegheimish, and Matthew Reimherr. Harnessing vision-language models for time series anomaly detection. *arXiv preprint arXiv:2506.06836*, 2025.

Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1303–1312, 2009.

Nam Hyeon-Woo, Moon Ye-Bin, Wonseok Choi, Lee Hyun, and Tae-Hyun Oh. Vlm's eye examination: Instruct and inspect visual competency of vision language models. *arXiv preprint arXiv:2409.14759*, 2024.

Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. Graphical perception of multiple time series. *IEEE transactions on visualization and computer graphics*, 16(6):927–934, 2010.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3182–3196, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.181. URL https://aclanthology.org/2024.findings-emnlp.181/.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *ICLR*, 2024.

Baoyu Jing, Sanhorn Chen, Lecheng Zheng, Boyu Liu, Zihao Li, Jiaru Zou, Tianxin Wei, Zhining Liu, Zhichen Zeng, Ruizhong Qiu, et al. Tsaqa: Time series analysis question and answering benchmark. In *Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM)*, pp. 944–979, 2026.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

Lee Jung-Mok, Nam Hyeon-Woo, Moon Ye-Bin, Junhyun Nam, and Tae-Hyun Oh. Automated model discovery via multi-modal & multi-step pipeline. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=qGFvTIMS3W.

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. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, 2025a.

Yaxuan Kong, Yiyuan Yang, Shiyu Wang, Chenghao Liu, Yuxuan Liang, Ming Jin, Stefan Zohren, Dan Pei, Yan Liu, and Qingsong Wen. Achieving time series reasoning requires rethinking model design, tasks formulation, and evaluation. *arXiv preprint arXiv:2502.01477*, 2025b.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A sober look at llms for material discovery: Are they actually good for bayesian optimization over molecules? *arXiv preprint arXiv:2402.05015*, 2024.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480, 2007.

Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series. *Advances in Neural Information Processing Systems*, 36:49187–49204, 2023.

Chen Liu, Shibo He, Qihang Zhou, Shizhong Li, and Wenchao Meng. Large language model guided knowledge distillation for time series anomaly detection. *arXiv preprint arXiv:2401.15123*, 2024a.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE, 2008.

Haoxin Liu, Harshavardhan Kamarthi, Zhiyuan Zhao, Shangqing Xu, Shiyu Wang, Qingsong Wen, Tom Hartvigsen, Fei Wang, and B Aditya Prakash. How can time series analysis benefit from multiple modalities? a survey and outlook. *arXiv preprint arXiv:2503.11835*, 2025a.

Haoxin Liu, Chenghao Liu, and B Aditya Prakash. A picture is worth a thousand numbers: Enabling llms reason about time series via visualization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7486–7518, 2025b.

Qinghua Liu, Paul Boniol, Themis Palpanas, and John Paparrizos. Time-series anomaly detection: Overview and new trends. *Proceedings of the VLDB Endowment (PVLDB)*, 17(12):4229–4232, 2024b.

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. *arXiv preprint arXiv:2510.07513*, 2025c.

Shuo Liu, Di Yao, Lanting Fang, Zhetao Li, Wenbin Li, Kaiyu Feng, Xiaowen Ji, and Jingping Bi. Anomalyllm: Few-shot anomaly edge detection for dynamic graphs using large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pp. 785–790. IEEE, 2024c.

Tianhong Liu, Haikun Wei, Sixing Liu, and Kanjian Zhang. Industrial time series forecasting based on improved gaussian process regression: T. liu et al. *Soft Computing*, 24(20):15853–15869, 2020.

Daniel James Lizotte. Practical bayesian optimization. 2008.

Ganapathy Mahalakshmi, Subbiah Sridevi, and Shyamsundar Rajaram. A survey on forecasting of time series data. In *2016 international conference on computing technologies and intelligent data engineering (ICCTIDE'16)*, pp. 1–8. IEEE, 2016.

Ridwan Mahbub, Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Mizanur Rahman, Mir Tafseer Nayeem, and Enamul Hoque. The perils of chart deception: How misleading visualizations affect vision-language models. In *2025 IEEE Visualization and Visual Analytics (VIS)*, pp. 6–10. IEEE, 2025.

Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. *Advances in neural information processing systems*, 35:34505–34518, 2022.

Mike A Merrill, Mingtian Tan, Vinayak Gupta, Thomas Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3512–3533, 2024.

Ilan Naiman, Nimrod Berman, Itai Pemper, Idan Arbiv, Gal Fadlon, and Omri Azencot. Utilizing image transforms and diffusion models for generative modeling of short and long time series. *Advances in Neural Information Processing Systems*, 37:121699–121730, 2024.

Tung Nguyen, Qiuyi Zhang, Bangding Yang, Chansoo Lee, Jorg Bornschein, Sagi Perel, Yutian Chen, and Xingyou Song. Predicting from strings: Language model embeddings for bayesian optimization. 2024.

Evandro S Ortigossa, Fábio F Dias, Diego C Nascimento, and Luis Gustavo Nonato. Time series information visualization—a review of approaches and tools. *IEEE Access*, 2025.

C Carl Pegels. Exponential forecasting: some new variations. *Management Science*, pp. 311–315, 1969.

Rune Pettersson. *Visual information*. Educational Technology, 1993.

Cyril Picard, Faez Ahmed, et al. Fast and accurate bayesian optimization with pre-trained transformers for constrained engineering problems. *arXiv preprint arXiv:2404.04495*, 2024.

Francesco Piccialli, Fabio Giampaolo, Edoardo Prezioso, David Camacho, and Giovanni Acampora. Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion. *Information Fusion*, 74:1–16, 2021.

Bojana Ranković and Philippe Schwaller. Bochemian: Large language model embeddings for bayesian optimization of chemical reactions. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023.

Bojana Ranković and Philippe Schwaller. Gollum: Gaussian process optimized llms—reframing llm finetuning through bayesian optimization. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025.

Amir Rosenfeld, Neta Glazer, and Ethan Fetaya. Questioning the stability of visual question answering. *arXiv preprint arXiv:2511.11206*, 2025.

Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.

Baek Seong-Eun, Lee Jung-Mok, Kim Sung-Bin, and Tae-Hyun Oh. Efficient hyper-parameter search for lora via language-aided bayesian optimization. *arXiv preprint arXiv:2602.11171*, 2026.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

ChengAo Shen, Wenchao Yu, Ziming Zhao, Dongjin Song, Wei Cheng, Haifeng Chen, and Jingchao Ni. Multi-modal view enhanced large vision models for long-term time series forecasting. In *NeurIPS*, 2025.

Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pp. 364–371. Elsevier, 2003.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Mingtian Tan, Mike Merrill, Vinayak Gupta, Tim Althoff, and Tom Hartvigsen. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 37:60162–60191, 2024.

Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proceedings of VLDB*, 15(6):1201–1214, 2022.

Jarke J Van Wijk and Edward R Van Selow. Cluster and calendar based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, pp. 4–9. IEEE, 1999.

Arnav Verma, Kushin Mukherjee, Christopher Potts, Elisa Kreiss, and Judith E Fan. Chart-6: human-centered evaluation of data visualization understanding in vision-language models. *arXiv preprint arXiv:2505.17202*, 2025.

Ziyu Wang, Babak Shakibi, Lin Jin, and Nando Freitas. Bayesian multi-scale optimistic optimization. In *Artificial Intelligence and Statistics*, pp. 1005–1014. PMLR, 2014.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pp. 370–378. PMLR, 2016.

Lawrence Wong, Dongyu Liu, Laure Berti-Equille, Sarah Alnegheimish, and Kalyan Veeramachaneni. Aer: Auto-encoder with regression for time series anomaly detection. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 1152–1161. IEEE, 2022.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

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*, 2024.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Xiongxiao Xu, Haoran Wang, Yueqing Liang, Philip S Yu, Yue Zhao, and Kai Shu. Can multimodal llms perform time series anomaly detection? *arXiv preprint arXiv:2502.17812*, 2025.

Haotong Yang, Yi Hu, Shijia Kang, Zhouchen Lin, and Muhan Zhang. Number cookbook: Number understanding of language models and how to improve it. *ICLR*, 2025.

Luoxiao Yang, Yun Wang, Xinqi Fan, Israel Cohen, Jingdong Chen, and Zijun Zhang. Vitime: Foundation model for time series forecasting powered by vision intelligence. *arXiv preprint arXiv:2407.07311*, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*, 2024.

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Yunyue Zhu and Dennis Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pp. 358–369. Elsevier, 2002.

APPENDIX

A IMPLEMENTATION DETAILS

We report the implementation details of the proposed Visual Presentation Optimization (VPO) and Vision-embedded Bayesian Optimization (VeBO) framework. We implement VeBO based on Ranković & Schwaller (2025), where a visual feature extractor precedes a projection layer with a Gaussian Process (GP) surrogate model. During training, the visual feature extractor is kept frozen, while a projection layer and the GP are jointly optimized to fit the surrogate model. The projection layer maps the extracted visual embeddings from 2048 dimensions to a 16-dimensional representation. The GP surrogate is implemented using GPyTorch (Gardner et al., 2018), with a constant mean function and a Matérn kernel as the covariance function. For optimization, we run Bayesian Optimization for 50 iterations, starting from a randomly initialized visual presentation.

A.1 DESIGN SPACE CONSTRUCTION

As mentioned in the experiments section of the main paper, we construct the design space using six visual presentation configurations: plot type, color, canvas size, width, axis, and legend. Table 5 summarizes the search space for each configuration. Specifically, we consider ten plot types: Line, Bar, Scatter, Histogram, BoxPlot, Pie, HeatMap, Spiral, PolarArea, and StreamGraph. The color space consists of 1,000 discrete hex colors, while the canvas size is selected from {800×600, 1200×800, 1280×720, 1600×900}. We use three width levels, 1, 2, 3, where each level maps to a plot-type-specific value (*e.g.*, line width for line plots, marker size for scatter plots, ...), and treat both axis and legend as binary on/off configurations.

Table 5: **Search space of visual presentations.**

Visual Hyperparameters	Search Space
Plot Type	{Line, Bar, Scatter, Histogram, BoxPlot, Pie, HeatMap, Spiral, PolarArea, StreamGraph}
Color	1,000 hex colors
Canvas Size	{800×600, 1200×800, 1280×720, 1600×900}
Width	{Line:1, 3, 5, Scatter:10, 50, 90, ⋯ }
Axis	{On, Off}
Legend	{On, Off}

A.2 SURROGATE MODEL IMPLEMENTATION

Visual embedding extraction. To encode the visual presentation into an embedding, we extract the visual embeddings from the rendered images using Qwen2.5-VL-3B. We adapt Explicit-One-Word (EOL) prompting Jiang et al. (2024) to generate a compact representation of the given image, explicitly instructing the model to describe the image in a single word. Especially, we use the prompt: *"Describe the configurations of the plots in one word"*. Given this EOL prompt, we use the final layer's last hidden state from the model as the image representation for each rendered image. In the main experiment, we extract these embeddings from a single rendered images, which serves as the representation of the corresponding visual presentation.

Vision-embedded surrogate modeling. Calibrating the embedding space extracted from the LLM can improve BO performance compared to using fixed embeddings Kristiadi et al. (2024); Ranković & Schwaller (2025); Seong-Eun et al. (2026). Motivated by this, after obtaining a visual embedding for the visual presentation, we pass it through a projection layer P that reduces its dimensionality from 2048 to 16. We then use the projected visual embedding as input to a Gaussian Process (GP) surrogate model. For the GP, we used the Matern kernel as the covariance function. With the visual embedding extraction, projection layer, and GP, the surrogate model is iteratively updated with observed data D during Bayesian Optimization to guide the search for better visual presentations.

Algorithm of VeBO. We provide the overall VeBO procedure in the form of pseudo-code in Algorithm 1. VeBO proceeds in four main stages. First, during initialization, an initial visual presentation is selected from the search space. Second, the selected presentation is evaluated on the target model to obtain its actual function value. Third, the surrogate model is updated using the newly observed evaluation result together with the corresponding visual embedding. Finally, based

Algorithm 1 Vision-embedded Bayesian Optimization (VeBO)

Init Search space S, evaluation dataset $\{(x_i, y_i)\}_{i=1}^{N}$, image renderer $g(\cdot; v)$, model $M(\cdot)$, visual embedding extractor $\texttt{MLLM}(\cdot)$, aggregation function $\mathcal{A}(\cdot)$, projection layer P, acquisition function $\alpha(\cdot)$, and iteration K. Output: Best visual presentation v^*.
(1) Initialization: sample $v_0 \sim S$, set $\mathcal{D} \leftarrow \emptyset$
For $j = 0$ to $K - 1$
(2) Model Evaluation:
$I_{v_j,i} = g(x_i; v_j) \leftarrow$ make image using v_j
$R_{v_j} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{R}\big(M(I_{v_j,i}), y_i\big) \leftarrow$ evaluate performance
$\mathcal{D} \leftarrow \mathcal{D} \cup \{(v_j, R_{v_j})\}$
Exclude v_j from S
(3) Vision-Embedded Surrogate Model Update:
$h_{v_j} = \mathcal{A}\big(\{\texttt{MLLM}(I_{v_j,i})\}_{i=1}^{N}\big) \leftarrow$ extract visual embeddings of v_j
$z_{v_j} = \mathrm{P}(h_{v_j})$
Update the surrogate model using $\mathcal{D}$, z_{v_j}
(4) Next Visual Presentation Selection:
Use Branch-and-Bound for the acquisition function $v_{j+1} \leftarrow \arg\max_{v \in S} \alpha(v)$
return $v^* = \arg\max_{(v,R_v) \in D} R_v$

Table 6: **Prompts for task model evaluation.** We provide the prompts for MLLM time-series task evaluation at TimeMQA: anomaly detection, classification, forecasting, and open-ended QA.

Task	Prompt
Anomaly Detection	{Question} If there is no anomaly, answer "Normal Point". If there is an anomaly, answer "Anomaly Point". Answer with only the label, nothing else.
Classification	{Question} Answer with only the class label, nothing else.
Forecasting	{Question} Based on the time series shown in the image and the context above, predict the next values. Return your answer as a Python list only, *e.g.*, `[v1, v2, ..., vN]`. Do not include any other text.
Open-Ended QA (T/F)	{Question} Answer with only "True" or "False", nothing else.
Open-Ended QA (MCQ)	{Question} Answer with only the correct option, including its label and text (*e.g.*, "A) option text"), nothing else.

on the surrogate model's predictions, the acquisition function proposes the next visual presentation to evaluate. These steps are repeated iteratively to search for the best-performing visual presentation.

A.3 Task Model Evaluation

In this stage, we evaluate the visual presentation v by its performance on the target task. Given v, we first render the corresponding plot and provide it as input to the task model for solving the time-series task. The task model then produces an output based on this plot, and we compute a task performance score from the output. This score is used as the evaluation score of v. To obtain consistent outputs, we append a short prompt to each question instructing the MLLM to respond with a single answer. The full prompts are shown in Table 6.

B Preliminary Experiment

As a proof of concept, we conduct a preliminary experiment in a smaller search space to check whether the proposed VPO and VeBO are effective. Below we report the experiment setup, quantitative results of VPO, comparison with HPO baselines, and computational costs. Since the search space is relatively small compared to the experiments in the main paper and it is possible to conduct an exhaustive search, we compared with the exhaustive search, random, BO, and GoLLuM. The main experiments over a substantially larger search space are presented in Sec. 4.

Table 7: **Preliminary experiment results.** We report quantitative results on closed models(*e.g.*, GPT-4o-mini, Gemeni-2.5-Flash-Lite). As shown, the models' performance consistently improves with VPO, without model fine-tuning.

Model	Ours	TimeMQA										
		Anomaly Detection			Classification			Forecasting		Open-Ended QA		
		Precision(↑)	Recall(↑)	F1 (↑)	Precision(↑)	Recall(↑)	F1(↑)	$NMSE_{v1}$(↓)	$NMSE_{v2}$(↓)	T/F (↑)	MCQ (↑)	Overall (↑)
GPT-4o-mini	✗	0.6098	0.2300	0.3340	0.2689	0.2867	0.1677	12.7281	62.6933	0.7192	0.6461	0.6740
	✓	0.5728	0.7700	0.6569	0.2535	0.3156	0.2662	9.1926	38.6880	0.7231	0.6508	0.6784
Gemini-2.5-Flash-Lite	✗	0.4995	0.6771	0.5749	0.2194	0.2671	0.2073	5.0867	7.1709	0.7192	0.5986	0.6446
	✓	0.5046	0.7129	0.5909	0.3858	0.2750	0.2342	4.8330	6.1644	0.7269	0.6223	0.6623
Qwen2.5-VL-7B	✗	0.4658	0.1071	0.1742	0.1604	0.2500	0.1884	95.0252	31.2699	0.6731	0.5273	0.5830
	✓	0.5004	0.9986	0.6666	0.2140	0.3080	0.2455	25.1929	29.9471	0.6692	0.5344	0.5859

Table 8: **Preliminary comparison results with baselines at a smaller search space.** We report that the proposed VeBO outperforms other baselines and shows competitive performance with exhaustive search.

Baselines	TimeMQA										
	Anomaly Detection			Classification			Forecasting		OpenQA		
	Precision(↑)	Recall(↑)	F1(↑)	Precision(↑)	Recall(↑)	F1(↑)	$NMSE_{v1}$(↓)	$NMSE_{v2}$(↓)	T/F(↑)	MCQ(↑)	Overall(↑)
Random	0.418	0.474	0.365	0.200	0.264	0.210	107.8	15.57	0.573	0.534	0.549
GoLLuM [46]	0.439	0.476	0.383	0.213	0.290	0.235	66.45	16.54	0.669	0.530	0.583
BO	0.435	0.480	0.371	0.269	0.301	0.255	97.29	17.19	0.673	0.542	0.592
Exhaustive	0.524	0.524	0.519	0.241	0.316	0.255	66.17	14.39	0.677	0.539	0.592
VeBO (Ours)	0.444	0.476	0.388	0.249	0.320	0.262	63.32	15.50	0.669	0.534	0.586

Experiment setup. We conduct a preliminary experiment using a small search space consisting of eight visual presentation configurations: (1) plot type, (2) color, (3) thickness/marker size, (4) grid, (5) axis, (6) legend, (7) trend, and (8) label. We consider only two plot types: line and scatter, and restrict each RGB color channel to either 0 or 255, resulting in eight possible colors. For thickness/marker size, we use $\{1, 2, 3\}$ for line thickness and $\{10, 20, 30\}$ for marker size, depending on the plot type. The remaining configurations, such as grid, axis, legend, trend, and label, are binary on/off choices. This results in a total of $2 \times 8 \times 3 \times 2^5 = 1{,}536$ possible visual presentation configurations. Anomaly-F1 score is used for anomaly detection, average F1 score for classification, NMSE for forecasting, and accuracy for open-ended QA as an optimization objective. We extracted the visual embedding from 500 rendered images.

Quantitative results. We report results for diverse models: GPT-4o-mini Achiam et al. (2023) and Gemini-2.5-Flash-Lite Comanici et al. (2025)), and Qwen2.5-VL-7B Bai et al. (2025). As shown in Table 7, the proposed VPO consistently improves model performance without any model fine-tuning. This result indicates that optimizing the visual presentation of time-series inputs alone can significantly enhance model performance, even without modifying the model itself. These findings highlight the effectiveness of VPO, particularly in closed-model settings where internal parameters are inaccessible. As many modern models are deployed as API-based services, direct modification or fine-tuning is often infeasible; in such scenarios, VPO offers a practical alternative by improving performance through input-level optimization.

Comparison with baselines. We compare VeBO with optimization baselines in Table 8 [3]. VeBO achieves the best or second-best performance across tasks, notably outperforming Random search, BO, and GoLLuM. Importantly, it achieves performance close to exhaustive Grid search, which serves as an approximate upper bound by evaluating the entire search space, while requiring far fewer evaluations, which shows that VeBO can approach the best attainable visual presentation in a sample-efficient manner.

Computational costs. To quantify practical efficiency, we report the runtime of each method on an A100 in Table 9[3]. Exhaustive grid search evaluates all 1,536 combinations and requires more than 2 days to complete. In contrast, VeBO completes in only 2–3 hours with only marginal additional cost over other baselines, while achieving performance comparable

Table 9: **Compuational costs.**

Method	TimeMQA (hours)			
	A.D.	CLS.	F.C.	QA
Exhaustive	58.95	56.79	22.21	32.24
Random	2.81	2.49	0.79	1.30
BO	3.29	2.61	1.04	1.20
GoLLuM	2.98	2.81	2.26	1.85
VeBO (Ours)	3.30	3.16	2.28	2.24

[3]A.D: Anomaly Detection, CLS.: Classification, F.C.: Forecasting, QA: OpenQA, P: Precision, R: Recall, N: NMSE, : best, : second-best.

Table 10: **Ablation study of VPO across model size.** We analyze how the effectiveness of VPO varies with model size, showing larger performance gains for bigger models.

Model	Ours	TimeMQA										
		Anomaly Detection			Classification			Forecasting		Open-Ended QA		
		Precision(↑)	Recall(↑)	F1 (↑)	Precision(↑)	Recall(↑)	F1(↑)	$NMSE_{v1}$(↓)	$NMSE_{v2}$(↓)	T/F (↑)	MCQ (↑)	Overall (↑)
Qwen2.5-VL (**3B**)	✗	0.6667	0.0086	0.0170	0.1111	0.2707	0.1312	3814.8069	30.0963	0.5846	0.4584	0.5066
	✓	0.3889	0.0101	0.0197	0.2410	0.2742	0.1502	3748.2325	32.0578	0.5385	0.4669	0.4949
Qwen2.5-VL (**7B**)	✗	0.4658	0.1071	0.1742	0.1604	0.2500	0.1884	95.0252	31.2699	0.6731	0.5273	0.5830
	✓	0.5004	0.9986	0.6666	0.2140	0.3080	0.2455	25.1929	29.9471	0.6692	0.5344	0.5859

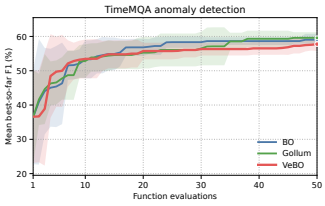

Figure 6: **Performance over Anomaly Detection.**

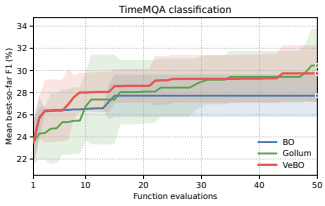

Figure 7: **Performance over Classification.**

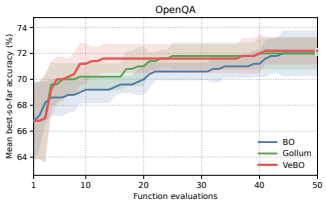

Figure 8: **Performance over OpenQA.**

to exhaustive search (Table 8). These results demonstrate that VeBO offers a favorable balance of computational efficiency and strong performance.

Ablation study on task model size. Table 10 shows an ablation study of VPO across different model sizes using Qwen2.5-VL-3B and Qwen2.5-VL-7B. As shown, both Qwen2.5-VL-3B and Qwen2.5-VL-7B benefit from optimized visual presentations, improving overall performance. While the 3B model shows modest improvements across tasks, the 7B model exhibits substantially larger gains, indicating that larger models benefit more from solving VPO. This suggests that larger models with greater representation capacity can better leverage VPO, yielding greater performance improvements. In particular, the optimized visual presentations discovered by VeBO appear to be more effectively utilized by larger models.

C Additional Results at Larger Space

Based on the PoC results, we enlarged the visual presentation search space into more larger, as we reported in main paper. The experiments here follows the implementation details mentioned at Sec. A. In here, we report the budget curve or the 5-time repeated experiments, and additional results at other benchmarks, and discovered visual presentations at different models.

C.1 Additional Quantitative Results

Convergence analysis. We conducted 5-time repeated experiments to validate our BO-based baselines and VeBO. For repeated experiments, we checked the baselines' convergence through a subset of original validation set ($N = 100$). Figures 6, 7, 8 show the convergence behavior during the visual presentation optimization process. While VeBO performs moderately among the baselines, it shows superior performance in classification and OpenQA. We propose that VeBO shows faster or simlar convergence on all three tasks, showing its effectiveness in guiding the model's performance using the visual prior from MLLM.

VPO Generalizes Across Datasets. To further evaluate the generalizability of VPO across different benchmarks, we provide additional results on TSAQA (Jing et al., 2026), MTBench (Chen et al., 2025), and ChatTS (Xie et al., 2024), which cover diverse datasets and TSA tasks. For this experiment, VPO is conducted over a smaller search space of 40K visual presentation configurations. As shown in Table 11[3], the optimized visual presentations consistently improve performance across these benchmarks. These results suggest that the benefit of visual presentation optimization is not limited to a single benchmark and further hint that visual presentations discovered on one dataset can capture model-specific preferences that transfer across datasets.

Table 11: **Results on TSAQA, MTBench, and ChatTS.** We report the result on other benchmarks. As shown, optimized visual presentation consistently improves performance accross diverse benchmarks.

Dataset	**TSAQA**						**MTBench**			**ChatTS**	
Task	**A.D.**		**CLS.**		**QA**		**F.C.**	**QA**		**QA**	
Metric	Precision(↑)	Recall(↑)	F1(↑)	Acc.(↑)	T/F(↑)	MCQ(↑)	MAE(↓)	7 days(↑)	30 days(↑)	Cate.(↑)	Num.(↑)
Default	0.254	0.500	0.337	0.407	0.543	0.472	4.627	0.495	0.436	0.594	0.520
VeBO (Ours)	0.582	0.544	0.482	0.539	0.548	0.481	2.356	0.499	0.458	0.603	0.520

Table 12: **Visual representation details in TimeMQA.** We report the top-1 visual presentation results obtained with VeBO. In the "Task" column, *A.D.* denotes anomaly detection, *Cls.* denotes classification, *F.C.* denotes forecasting, and *QA* denotes open-ended question answering. "Width" indicates the marker size for scatter plots, the linewidth for line plots, and the element width for the remaining plot types.

Model	**Task**	**Plot Type**	**Color**	**Width**	**Canvas**	**Axis**	**Legend**
GPT-5-mini	*A.D.*	Boxplot	[11,217,115]	0.55	1600×900	On	Off
	Cls.	Line	[166,41,65]	5	1600×900	On	On
	F.C.	Boxplot	[217,98,99]	0.55	800×600	Off	Off
	QA	Bar	[155,166,8]	0.60	800×600	Off	On
Gemini-2.5-Flash-Lite	*A.D.*	Boxplot	[173,54,217]	0.85	1200×800	Off	On
	Cls.	Line	[216,217,98]	5	1200×800	Off	Off
	F.C.	Scatter	[166,8,20]	10	800×600	Off	On
	QA	Boxplot	[166,85,41]	0.25	800×600	Off	Off
Qwen2.5-VL-7B	*A.D.*	Scatter	[217,176,76]	10	800×600	Off	On
	Cls.	Polar-area	[217,11,199]	0.70	800×600	Off	On
	F.C.	Scatter	[217,11,50]	50	800×600	Off	On
	QA	Line	[8,73,166]	1	800×600	On	Off
LLaMA3.2-11B-Vision	*A.D.*	Boxplot	[217,146,98]	0.55	800×600	Off	Off
	Cls.	Boxplot	[166,134,75]	0.85	800×600	On	On
	F.C.	Spiral	[217,33,90]	0.70	1200×800	Off	Off
	QA	Scatter	[166,83,58]	90	800×600	On	On

C.2 Additional Qualitative Results

We provide visualizations of the top-3 visual presentations discovered by VeBO for GPT-5-mini, Gemini-2.5-Flash-Lite, and LLaMA3.2-11B-Vision on the TimeMQA benchmark, as shown in Figures 9, 10, and 11, respectively. The discovered visual presentations are provided at Table 12. The results show that, even within the same model, the preferred visual presentations differ across tasks. They also vary across models, indicating that different models favor different visual presentations for the same benchmark. These qualitative results further highlight the task- and model-specific nature of visual presentation preferences.

D Additional Analysis

D.1 Additional results of VeBO surrogate model

Text and visual embedding. Figure 5 shows how the embeddings are aligned over the model's performance prediction. In here, we additionally visualize how the text and visual embedding is aligned with the visual presentation hyperparameters at Figure 12. For both text and visual, the original embeddings is primarily organized by visual presentation, especially plot type. After projection training, projection learning preserves the semantic organization of the original representation while reshaping it according to task performance. Compared to text embeddings, visual embeddings better preserve appearance-level attributes such as color, while still capturing structural differences across plot types. In contrast, text embeddings primarily reflect categorical configuration semantics and show weaker organization with respect to color. This appearance-aware representation helps explain the stronger surrogate performance of visual embeddings reported in the main results, as the surrogate can better capture performance variations induced by visual presentation attributes.

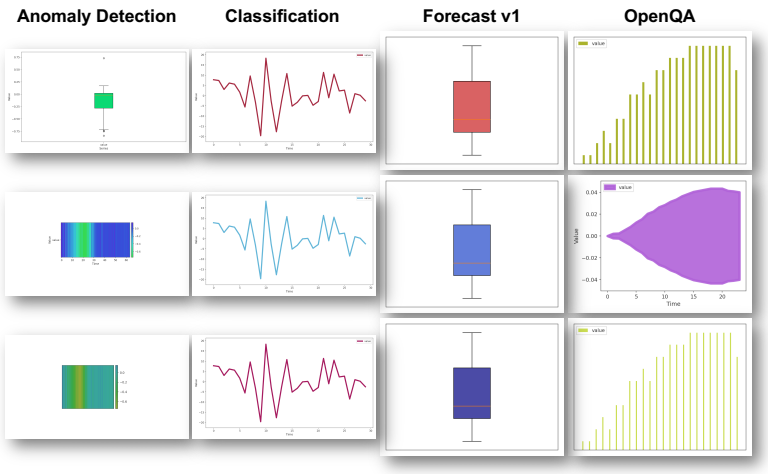

Figure 9: **Qualitative results of GPT-5-mini.**

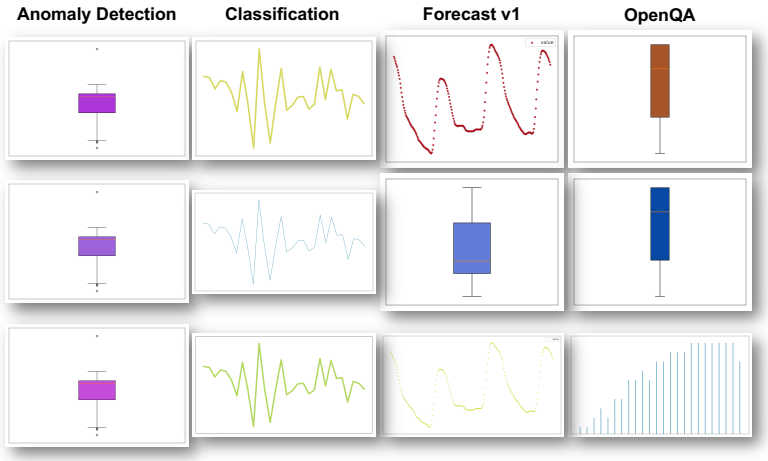

Figure 10: **Qualitative results of Gemini-2.5-Flash-Lite.**

Held-Out Performance Prediction. We further evaluate whether the performance-aware embedding spaces generalize to unseen visualization configurations. Both text- and visual-based surrogates are trained using the same 50 randomly sampled configurations and evaluated on an independent set of 50 configurations with no training overlap. Figure 13 compares the predicted and measured F1 scores, where the dashed line denotes ideal prediction ($y = x$). Both models exhibit a strong correspondence between predicted and actual performance, confirming that the learned projections capture performance-relevant structure. However, predictions derived from visual embeddings are more tightly concentrated around the identity line, yielding a higher R^2 (0.803 vs. 0.757), lower MAE (0.0124 vs. 0.0146), and stronger rank correlation (0.860 vs. 0.770), which indicates that visual embeddings provide a more informative prior for modeling visualization performance.

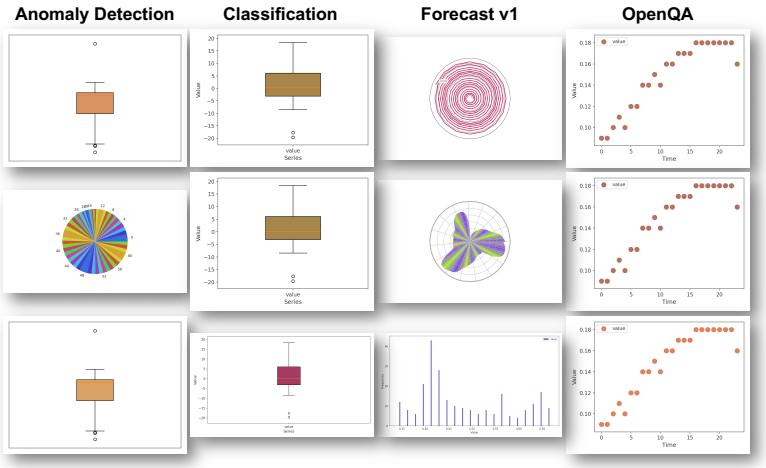

Figure 11: **Qualitative results of Llama-3.2-11B-Vision.**

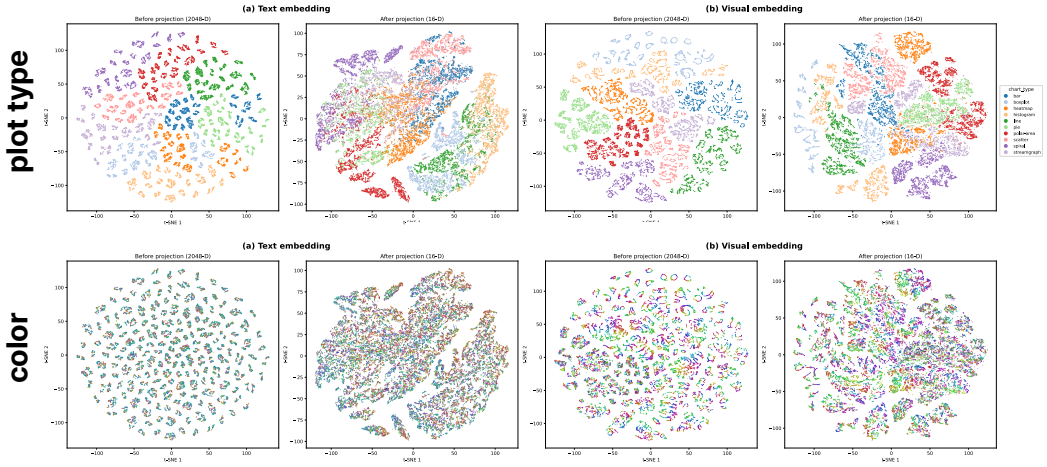

Figure 12: **t-SNE visualization of Qwen2.5-VL and Qwen2.5 per plot type and color.**

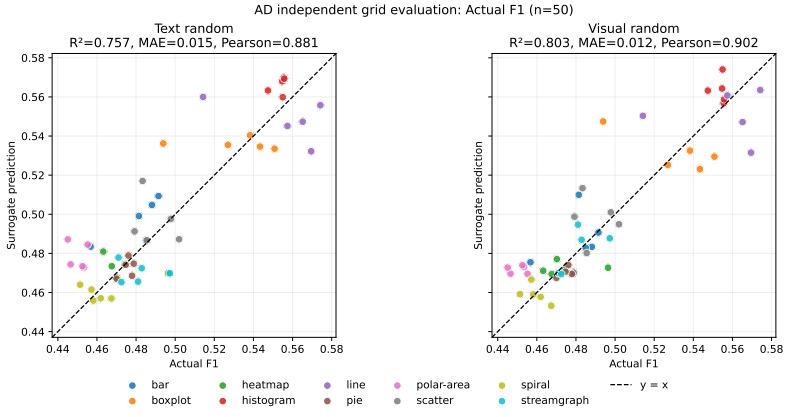

Figure 13: **Predicted F1 vs. measured F1 at anomaly detection on unseen visual presentations.**