# TSROUTER: Dynamic Modality-Model Selection for Time Series Reasoning

**Fangxu Yu**[1], **Tao Feng**[2], **Dehai Min**[3], **Lu Cheng**[3], **Ge Liu**[2], **Tianyi Zhou**[4]
[1]University of Maryland, College Park, [2]University of Illinois Urbana-Champaign,
[3]University of Illinois Chicago, [4]MBZUAI

## Abstract

Time series reasoning is essential for real-world problem-solving. While both Large Language Models (LLMs) and Vision-Language Models (VLMs) can reason about time-series data, their capabilities are complementary: LLMs process time series as text sequences and thus preserve exact numerical understanding, but struggle with global patterns, whereas VLMs efficiently capture these patterns by visualizing time series but may lose fine-grained details. Moreover, models vary significantly in task-specific expertise and inference costs. Dynamically selecting the most suitable modality and model for each query is therefore crucial, yet challenging because it requires modeling the complex interactions among tasks, queries, modalities, and models, which carry rich contextual signals. To this end, we introduce TSROUTER, a graph-based dynamic routing framework. TSROUTER constructs a heterogeneous graph of task, query, modality, and model nodes to contextualize the interactions among query characteristics, modality attributes, and model capabilities. TSROUTER formulates routing as a candidate scoring problem, where each modality-model pair is evaluated based on user-defined performance-cost preferences to select the optimal candidate. Comprehensive evaluations on 4 distinct time series reasoning tasks reveal that TSROUTER substantially outperforms diverse baselines with 16% to 46% relative improvements. Furthermore, TSROUTER demonstrates robust zero-shot plug-and-play generalization to unseen models and novel tasks and preserves high performance while reducing computational overhead through cost-aware optimization. Our code is available at https://github.com/tianyi-lab/TSRouter.

## 1 Introduction

Time series plays a critical role in high-stakes domains where time series records the temporal evolution of real-world variables. Interpreting these sequential patterns is fundamental to understanding complex processes or guiding proactive decisions. For example, in education (Mao et al., 2024) , analyzing student trajectories such as engagement and attempt counts helps identify struggling learners and personalize study plans. In scientific discovery (Yu et al., 2025c), time series such as gravitational wave recordings encode astrophysical processes that may lead to discoveries of exoplanets and black-hole mergers. Analyzing time series and conducting reasoning enable automated systems to support high-impact applications.

Recently, Large Language Models (LLMs) and Vision-Language Models (VLMs) have emerged as powerful tools for time series analysis. LLM-based methods (Merrill et al., 2024; Wang et al., 2025) treat time series as raw text sequences, preserving numerical precision but suffering from context length bottlenecks, high computational costs, and poor global pattern recognition. Conversely, VLM-based approaches (Zhong et al., 2025; Liu et al., 2025) convert time series into visual plots, efficiently capturing global patterns in long sequences but losing fine-grained numerical detail due to image resolution limits. Beyond modality choice, individual models also vary in their ability to handle different types

| Method | Modality Routing | Interaction | | Efficiency | |
|---|---|---|---|---|---|
| | | Query-Query | Query-Model | Inference Speed | Memory Use |
| CausalLM (Ong et al., 2024) | ✗ | ✗ | ✗ | Slow | Medium |
| GraphRouter (Feng et al., 2024) | ✗ | ✗ | ✓ | Fast | Low |
| Hybrid LLM (Ding et al., 2024a) | ✗ | ✗ | ✗ | Medium | Medium |
| RouterDC (Chen et al., 2024b) | ✗ | ✓ | ✗ | Medium | Medium |
| Router-R1 (Zhang et al., 2025a) | ✗ | ✗ | ✗ | Medium | High |
| TSROUTER | ✓ | ✓ | ✓ | Fast | Low |

Table 1: Comparison of TSROUTER with existing routing methods. Unlike previous approaches that only select among models under a fixed modality, TSROUTER jointly performs modality and model routing while capturing both query-query and query-model interactions with fast inference and low memory.

of reasoning tasks and problems. Training a single model that unifies these complementary strengths is expensive, whereas routing among off-the-shelf LLMs and VLMs offers a lightweight and scalable alternative. Therefore, dynamically routing each query to the most appropriate modality and model is crucial for fully exploiting their complementary strengths across both dimensions. However, as summarized in Table 1, existing routing approaches focus solely on model selection, without jointly considering modality choice. Moreover, they capture at most one type of interaction, failing to jointly model the complex relationships among tasks, queries, modalities, and models that carry rich contextual signals for routing. These limitations make them suboptimal for time series reasoning, where both modality fitness and model expertise critically determine performance.

To address these limitations, we introduce TSROUTER, a graph-based router designed for dynamic modality and model selection. TSROUTER constructs a heterogeneous graph to explicitly model the contextual information and interactions among tasks, queries, modalities, and models, enabling context-aware routing that generalizes to new models and tasks without retraining. Specifically, the graph comprises 4 distinct node types: tasks, queries, modalities, and models, with edges encoding the different relationships among them. TSROUTER frames the model-modality routing problem as a candidate scoring task, where the optimal choice of time series modality-model relies on user-defined preferences regarding performance and cost. Once the candidates are scored, the system routes each query to the optimal candidate. To achieve zero-shot generalization to unseen tasks and models, we prompt an LLM such as GPT-5 to profile each node, outlining critical properties such as task requirements, model descriptions, and token costs. These profiles are then embedded using a pretrained text embedding model to initialize node features. Finally, a heterogeneous Graph Neural Network (GNN) aggregates cross-type neighborhood information, followed by a scorer to produce the final routing decision.

We comprehensively evaluate TSROUTER across 4 fundamental time series reasoning tasks. Experimental results indicate that TSROUTER greatly outperforms a wide range of baselines with 16% - 46% improvements. Furthermore, TSROUTER demonstrates remarkable generalization on unseen tasks and robustly handles new models without additional fine-tuning. Finally, by explicitly optimizing for cost-aware routing, TSROUTER establishes a superior Pareto frontier, preserving reasoning capabilities while minimizing computational overhead compared to current approaches.

## 2 TSROUTER: Modality-Model Selection for Time Series Reasoning

### 2.1 Motivating Analysis

We start by demonstrating the complementary benefits of different time series modalities and LLMs and VLMs for time series reasoning on the TSRBench test set (see Section 3) and identify two observations that motivate TSROUTER.

**Different time series modalities excel on different tasks.** Figure 1 (a) shows which modality achieves the highest average accuracy across models on each of the 15 subtasks, grouped by 4 task categories: Perception, Decision-Making, Prediction, and Reasoning. We

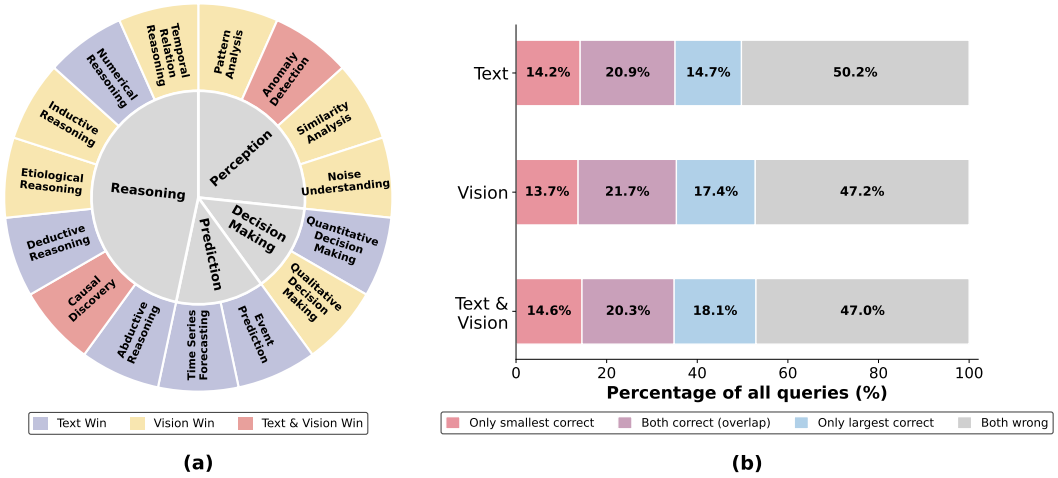


Figure 1: Motivation analysis results. (a) Distribution of best-performing modality across 15 subtasks grouped by 4 task categories. (b) Breakdown of queries correctly answered by only the smallest model, both models, and only the largest model within each modality.

observe that visual modality dominates Perception, winning 3 out of 4 subtasks, as visual plots efficiently encode the global shape and structural patterns that perception tasks rely on. In contrast, text modality excels at Prediction, where preserving exact numerical values is critical for forecasting. For Reasoning and Decision-Making, however, no single modality holds a consistent advantage. This diverse pattern confirms that different modalities complement rather than dominate others, motivating per-query modality selection rather than a fixed modality choice.

**Different models address varied queries.** Figure 1 (b) decomposes the correctly answered queries into three disjoint groups for each modality type: those answered correctly by only the smallest model, by both models, and by only the largest model. Although the largest models achieve higher overall accuracy in all three categories, the overlap ratio, which is the fraction of queries answered correctly by both the smallest and largest models, is low. This indicates that larger and smaller models largely succeed on different subsets of queries: nearly half of the queries correctly answered by the large model are not the same ones correctly answered by the small model, and vice versa. Consequently, always selecting the largest model leaves a substantial portion of queries unserved that a smaller model could have handled correctly, motivating the need for per-query routing to different models. See a qualitative analysis example in Appendix D.

These findings indicate that performance varies significantly across tasks, queries, modalities, and models, suggesting that an effective router should account for task differences, discern individual query characteristics, and jointly select the most suitable modality and model. Since these elements exhibit complex interactions that carry rich information for routing, there is a pressing need for a framework that explicitly models the interplay among them to make context-aware decisions.

### 2.2 Problem Formulation

Then, we formulate the time series routing problem. Let $\mathcal{Q} = \{q_1, \ldots, q_N\}$ denote a set of *time series queries*, where each query $q_i = (\mathbf{x}_i, u_i)$ consists of a set of time series $\mathbf{x}_i = \{x_i^{(1)}, \ldots, x_i^{(K_i)}\}$ with $x_i^{(k)} \in \mathbb{R}^{T_i^{(k)}}$ and a natural language question $u_i$. We assume access to a set of foundation models $\mathcal{L} = \{\ell_1, \ldots, \ell_L\}$ containing both LLMs and VLMs, and a set of input modalities $\mathcal{M} = \{m_1, \ldots, m_M\}$, where each modality specifies how the time series is presented to a model: as numerical text, as a visual chart, or as a mixture of both. LLMs can only process textual time series as inputs, and VLMs can process visual and textual inputs. We denote the set of valid modality-model combinations as $\mathcal{C} \subseteq \mathcal{M} \times \mathcal{L}$, so that

each candidate $c = (m, \ell) \in \mathcal{C}$ specifies both how the time series is presented and which model processes it.

Given a query $q_i$ and a candidate $c \in \mathcal{C}$, let $I(q_i, c) \in \{0, 1\}$ denote the correctness indicator of whether $c$ produces a correct answer, and let $\text{cost}(q_i, c) \geqslant 0$ denote the inference cost (e.g., API pricing). We define the *effectiveness score* as a convex combination of correctness and (normalized) cost:

$$e(q_i, c) = \alpha \cdot I(q_i, c) - (1 - \alpha) \cdot \widetilde{\text{cost}}(q_i, c), \tag{1}$$

where $\widetilde{\text{cost}}$ denotes the min-max normalized cost across all candidates, and $\alpha \in [0, 1]$ controls the user's preference over the performance–cost trade-off. When $\alpha = 1$, the objective reduces to pure accuracy maximization; smaller values of $\alpha$ increasingly penalize expensive candidates. The goal of routing is to select the candidate with the highest effectiveness:

$$c_i^* = \arg\max_{c \in \mathcal{C}} e(q_i, c) \tag{2}$$

where $c_i^*$ is the optimal candidate for solving query $q_i$.

### 2.3 TSROUTER Framework

**Overview.** As shown in Figure 2, TSROUTER first transforms the interaction data among tasks, queries, modalities, and models into a graph, and models them as task nodes, query nodes, modality nodes, and model nodes, while the relationships derived from the interaction data are represented by edges. We apply the GNN to embed the node features and use them for routing. We exploit its compositional structure by constructing a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

**Node set.** The node set $\mathcal{V} = \mathcal{T} \cup \mathcal{Q} \cup \mathcal{M} \cup \mathcal{L}$ comprises 4 disjoint types. All node features are initialized by encoding natural-language descriptions using a pretrained text-embedding model $\phi$, such as Qwen3-embedding (Zhang et al., 2025b). Specifically, (i) **task nodes** embed a category-level description of scope and requirements, (ii) **query nodes** embed the concatenation of the question text $u_i$ and a textual statistical summary of its time series (e.g., channel count, trend direction), allowing the router to consider modality preferences from data characteristics without accessing the raw series (iii) **modality nodes** embed a description of modality properties, and (iv) **model nodes** embed a capability-aware profile summarizing architecture, parameter scale, and cost. See all profiles in Appendix B.

**Edge set.** The heterogeneous edge set $\mathcal{E} = \mathcal{E}_{\text{TQ}} \cup \mathcal{E}_{\text{QM}} \cup \mathcal{E}_{\text{QL}} \cup \mathcal{E}_{\text{MM}} \cup \mathcal{E}_{\text{QQ}}$ consists of five relation types, each encoding a distinct relationship: (i) **task–query** edges ($\mathcal{E}_{\text{TQ}}$) link each query $q_i$ to its task category $\tau_q(q_i)$ so that queries belonging to the same task share task-level context; (ii) **query–modality** edges ($\mathcal{E}_{\text{QM}}$) link each query $q_i$ to every modality $m_j$; (iii) **modality–model** edges ($\mathcal{E}_{\text{MM}}$) connect each modality $m$ to every compatible model $\ell$, reflecting the valid candidates in $\mathcal{C}$; (iv) **query–model** edges ($\mathcal{E}_{\text{QL}}$) link each query $q_i$ to every model $\ell_k$. The fifth type captures query similarity: (v) **query–query** edges ($\mathcal{E}_{\text{QQ}}$) connect each query to its $k$-nearest neighbors based on query embeddings' cosine similarity, enabling similar queries to share information during message passing.

**Routing via Heterogeneous GNN.** After constructing the interaction histories into a heterogeneous graph, we perform representation learning over $\mathcal{G}$ using a stack of Heterogeneous Graph Transformer (HGT) layers (Hu et al., 2020) as our backbone due to its outstanding ability to maintain dedicated representations for different types of nodes.

To enable cross-type message passing, we encode each node's natural language description with $\phi$ to a embedding space, then apply a linear projection to map them into a shared latent space as the initial features: $\mathbf{h}_v^{(0)} = \mathbf{W}\phi(\mathbf{x}_v) + \mathbf{b}$. Starting from the features $\mathbf{h}_v^{(0)}$, HGT updates node embeddings by attending to type-specific neighbors, thereby capturing structured interaction patterns among different types of nodes. Formally, at each layer $l$, the embedding of a node $v$ is updated by aggregating messages from its neighbors based on relation-aware multi-head attention. We apply residual connections to preserve the initial

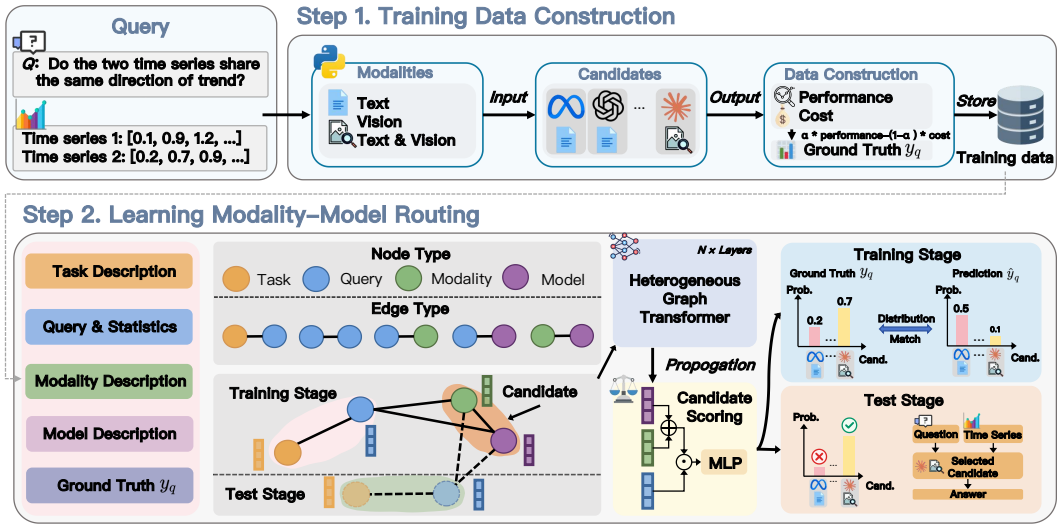


Figure 2: Overview of TSROUTER. Each query is paired with all modality–model combinations to collect performance and cost, forming the training data. Then, TSROUTER builds a heterogeneous graph from task, query, modality, and model descriptions, learns node representations, and scores candidates to route queries to the optimal modality–model pair.

features and stabilize optimization:

$$\mathbf{h}_v^{(\ell+1)} = \mathbf{h}_v^{(\ell)} + \mathrm{HGTConv}\left(\left\{\mathbf{h}_u^{(\ell)} : (u, r, v) \in \mathcal{E}\right\}\right), \tag{3}$$

where $\mathbf{h}_v^{(l)}$ denotes the embedding of node $v$ at layer $l$. After $L$ layers' propagation, we obtain the final node representations $\mathbf{h}_v^{(L)}$.

To score a candidate $(m, \ell)$ for query $q$, we first combine the modality and model representations additively. A candidate's capability is jointly determined by how the time series is presented and which model processes it. We then capture the interaction between the query and the candidate through element-wise multiplication, followed by a two-layer MLP:

$$s(q, m, \ell) = \mathrm{MLP}\left(\mathbf{h}_q^{(L)} \odot \left(\mathbf{h}_m^{(L)} + \mathbf{h}_\ell^{(L)}\right)\right), \tag{4}$$

Rather than hard labels that only identify the single best candidate, we train with *soft targets* that capture the full ranking over candidates. For each training query $q$, we compute a target distribution by applying softmax to the ground-truth effectiveness scores:

$$y_{q,c} = \frac{\exp\big(e(q, c)\big)}{\sum_{c' \in \mathcal{C}} \exp\big(e(q, c')\big)}, \tag{5}$$

Similarly, the predicted scores are converted into a distribution over candidates:

$$\hat{y}_{q,c} = \frac{\exp\big(s(q, m, \ell)\big)}{\sum_{c' = (m', \ell') \in \mathcal{C}} \exp\big(s(q, m', \ell')\big)}, \tag{6}$$

The model is trained by minimizing the KL divergence between the target distribution $y_q$ and the predicted distribution $\hat{y}_q$:

$$\mathcal{L} = \mathcal{D}_{\mathrm{KL}}\big(y_q \parallel \hat{y}_q\big), \tag{7}$$

This objective encourages the router to match the full ranking of candidates rather than merely identifying the top one, providing rich supervision when multiple candidates perform comparably. At test time, we select the candidate with the highest predicted score:

$$c^* = \arg\max_{(m,\ell) \in \mathcal{C}} s(q, m, \ell), \tag{8}$$

During training, the graph contains only training queries. At test time, each test query is inserted into the graph with its k-nearest-neighbor edges to existing nodes, and a single forward pass produces the routing score.

**TSROUTER for new tasks and models.** To incorporate a new task or model, we simply insert the corresponding node into the graph with its embedded feature and connect it following the same edge construction rules. For a new task, TSROUTER then scores all candidates of the existing modality-model against queries in that task. For a new model, TSROUTER leverages messages from neighboring nodes to estimate its per-query effectiveness and incorporate it into routing decisions.

# 3 Experiments

## 3.1 Experimental Setups

**Datasets.** We evaluate on *TSRBench* (Yu et al., 2026b), a comprehensive testbed covering 4 time series reasoning tasks including *Perception*, *Reasoning*, *Prediction*, and *Decision-Making*, with 15 different subtasks, whose statistics are summarized in Table 2. We use two additional tasks as unseen test sets to assess the generalization of TSROUTER, including *time series imputation* from *TSQA* dataset (Kong et al., 2025a), which predicts the missing value in the time series, and *Correlation Prediction* from *MTBench* (Chen et al., 2025), which requires predicting the correlation between the financial news sentiment and stock price time series.

**LLMs and VLMs.** For LLMs, we use Qwen3-8B, Qwen3-32B (Yang et al., 2025), LlaMa-3.3-70B-Turbo (Grattafiori et al., 2024), and Qwen3.5-397B-A17B (Qwen Team, 2026). For VLMs, we use Qwen3-VL-8B, Qwen3-VL-32B (Bai et al., 2025), GLM-4.5V (Hong et al., 2025), and Kimi-K2.5 (Team et al., 2026).

**Data Preprocessing;** Data are divided into train, validation, and test sets using a 7:1:2 ratio per task. Based on the training split, we construct the training data for TSROUTER by recording per-query correctness and cost (computed from token counts and per-token pricing; see Appendix B). Qwen3.5-397B-A17B and Kimi-K2.5 are selected as new LLM and VLM to assess generalizability.

| Task | #Subtask | #Cases |
|---|---|---|
| Perception | 4 | 700 |
| Reasoning | 7 | 1710 |
| Prediction | 2 | 1080 |
| Decision | 2 | 635 |

Table 2: Overview of Dataset.

**Baselines.** We compare against two groups of baselines. (i) **Rule-based: Largest LLM/VLM** always selects the largest available model. (ii) **Learning-based: Hybrid LLM** (Ding et al., 2024a) trains an MLP on frozen query embeddings; **RouterDC** (Chen et al., 2024b) ranks candidates by cosine similarity between query and model embeddings; **CausalLMRouter** (Ong et al., 2024) fine-tunes a decoder to generate the optimal candidate name; **EloRouter** (Ong et al., 2024) routes all queries to the highest Elo-rated model; **MFRouter** (Ong et al., 2024) uses matrix factorization to score query–model compatibility; and **KNNRouter** selects the best-performing candidate among $k$-nearest training queries. We adapt all baselines to our joint modality-model routing setting by treating each (modality, model) pair as an independent routing candidate.

**Evaluation Metrics.** We evaluate each routing method along two axes. *Accuracy*: following (Yu et al., 2025a; Xie et al., 2024), we use accuracy for time series reasoning and correlation prediction tasks; for time series imputation, we follow (Du et al., 2024; Yıldız et al., 2022) to report Mean Squared Error (MSE) and Mean Absolute Error (MAE). *Cost*: we report the total API cost in USD consumed on each task's test set, reflecting the expense of deploying each routing method.

**Implementation Details.** In the training stage, we search the hyperparameters on the validation set and use a two-layer GNN with an embedding dimension of 64 and $k = 60$ for nearest neighbor edges, and train the GNN using Adam optimizer (Kingma & Ba, 2014) with the learning rate as 1e-3. We implement our method using PyTorch Geometric (Fey & Lenssen, 2019) and conduct all experiments on a single NVIDIA RTX A6000 GPU. For LLMs and VLMs, we rely on API calling to obtain responses.

| Scenario | Perception | | Reasoning | | Prediction | | Decision-Making | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | Cost | Accuracy | Cost | Accuracy | Cost | Accuracy | Cost | Accuracy | Cost |
| *Rule-based Routing Methods* | | | | | | | | | | |
| LLM (largest) | 48.92 | 0.02 | 33.15 | 0.34 | 31.56 | 0.16 | 34.91 | 0.21 | 35.59 | 0.74 |
| VLM (largest) | 64.03 | 0.03 | 38.76 | 0.29 | 24.44 | 0.22 | 38.68 | 0.05 | 39.10 | 0.59 |
| *Learning-based Routing Methods* | | | | | | | | | | |
| EloRouter | 50.36 | 0.03 | 30.90 | 0.38 | 32.00 | 0.16 | 35.85 | 0.16 | 35.11 | 0.75 |
| MFRouter | 50.36 | 0.03 | 32.58 | 0.38 | 32.00 | 0.16 | 39.62 | 0.17 | 36.32 | 0.77 |
| KNNRouter | 56.12 | 0.03 | 35.11 | 0.34 | 32.44 | 0.19 | 40.57 | 0.17 | 38.62 | 0.79 |
| GraphRouter | 62.58 | 0.03 | 42.42 | 0.38 | 30.22 | 0.20 | 39.62 | 0.16 | 42.13 | 0.77 |
| Hybrid LLM | 58.99 | 0.03 | 48.03 | 0.43 | 30.22 | 0.17 | 40.57 | 0.13 | 44.07 | 0.76 |
| RouterDC | 61.87 | 0.03 | 34.27 | 0.41 | 40.00 | 0.23 | 39.62 | 0.17 | 41.16 | 0.83 |
| CausalLM | 55.40 | 0.03 | 39.04 | 0.39 | 38.67 | 0.22 | 39.62 | 0.14 | 41.77 | 0.78 |
| Router-R1 | 58.99 | 0.03 | 38.48 | 0.40 | 33.33 | 0.17 | 35.85 | 0.15 | 40.55 | 0.75 |
| TSROUTER | **67.63** | 0.03 | **52.81** | 0.45 | **43.11** | 0.21 | **42.45** | 0.04 | **51.33** | 0.73 |

Table 3: Performance and cost comparison of various baselines on 4 time series reasoning tasks. Bold and underline denote the best and second-best results.

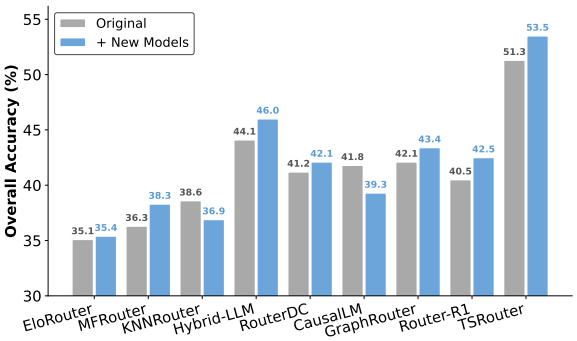


Figure 3: Results on generalization to new models.

| Scenario | Correlation Prediction | Imputation | |
|---|---|---|---|
| | Accuracy | MSE | MAE |
| LLM (largest) | 22.24 | 0.67 | 0.46 |
| VLM (largest) | 29.20 | 0.89 | 0.46 |
| EloRouter | 25.42 | 1.07 | 0.65 |
| MFRouter | 25.22 | 1.07 | 0.65 |
| KNNRouter | 24.93 | 0.99 | 0.59 |
| GraphRouter | 27.51 | 0.61 | 0.49 |
| Hybrid LLM | 29.59 | 0.60 | 0.46 |
| RouterDC | 28.60 | 0.74 | 0.47 |
| CausalLM | 27.81 | 0.89 | 0.47 |
| Router-R1 | 29.59 | 0.70 | 0.48 |
| TSROUTER | **31.38** | **0.56** | **0.43** |

Table 4: Results on generalization to new time series tasks.

### 3.2 Main Results

Table 3 compares TSROUTER against ten baselines across 4 task categories. TSROUTER achieves 51.33% overall accuracy, outperforming all routers by 16–46% relative at competitive cost. Two patterns stand out: (1) always selecting the largest LLM or VLM yields only mediocre accuracy, confirming that scale alone is insufficient; (2) routers with pretrained language model backbones (Hybrid LLM, RouterDC, CausalLM) outperform those without (EloRouter, MFRouter, KNNRouter), yet still lag TSROUTER by a substantial margin. This suggests that query-level semantics benefit routing but are insufficient on their own, and the structured relational signals captured by our heterogeneous graph are critical for joint modality-model selection.

### 3.3 Generalization Ability of TSROUTER

**Generalization to New Models.** A critical requirement for any practical routing framework is the ability to incorporate newly released models efficiently. We expand the candidate pool with two models (one LLM, one VLM) *unseen* during training and compare all routers under both the original and augmented pools. As illustrated in Figure 3, TSROUTER achieves the largest gain, increasing from 51.3% to 53.5%. In contrast, KNNRouter and CausalLM exhibit performance degradation upon incorporating new models, indicating that these methods fail to leverage the capabilities of unseen models and instead disrupt their existing routing decisions. These results confirm that TSROUTER can effectively integrate new models without retraining.

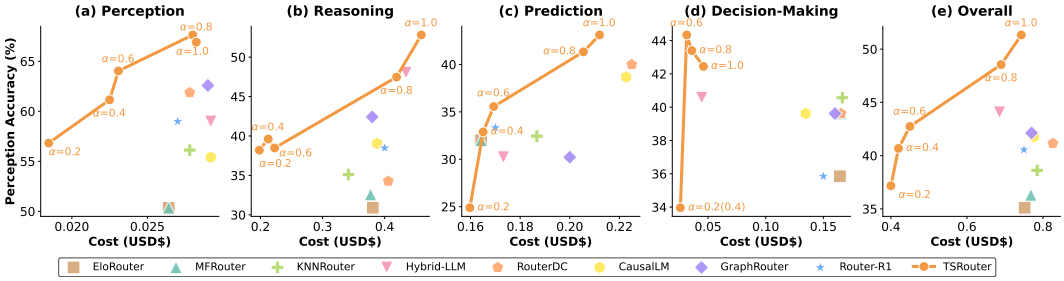


Figure 5: Accuracy–cost trade-off under varying cost preference $\alpha$. Each subplot corresponds to a task and overall accuracy. TSROUTER traces a clear Pareto front, consistently dominating all baselines across the full range of cost budgets.

**Generalization to New Tasks.** Beyond the 4 time series reasoning tasks, we evaluate TSROUTER on two held-out time series tasks: *correlation prediction*, which requires jointly analyzing news sentiment and stock price time series to assess their correlation, and *time series imputation*, which requires predicting missing numerical values in a sequence. As shown in Table 4, TSROUTER consistently outperforms all baselines on both tasks. Notably, the two tasks exhibit opposite modality preferences: the largest VLM leads on correlation prediction, where visual patterns aid alignment, while the largest LLM leads on imputation, where numerical precision is essential. Most baseline routers perform poorly on both tasks, indicating weak generalization to unseen task types. These results confirm that TSROUTER effectively identifies the characteristics of novel tasks and routes queries accordingly.

### 3.4 Modality Routing Analysis

To understand how TSROUTER allocates queries for different time series modalities, we analyze the proportion of instances routed to each modality per task category. As shown in Figure 4, TSROUTER learns routing strategies that align closely with the modality distribution identified in §2.1 and results in Table 3. Perception and reasoning queries are predominantly routed to the vision modality. Prediction queries, by contrast, are overwhelmingly routed to textual modality, reflecting the need for precise numerical extrapolation that textual modality better preserves. Decision-making exhibits the most balanced distribution across all three modality types, reflecting its need for both quantitative precision and global pattern understanding. These results demonstrate that TSROUTER effectively captures the modality demands of different queries and tasks.

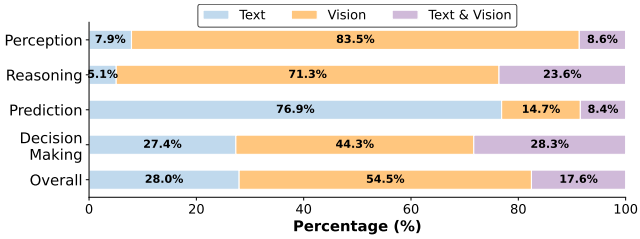


Figure 4: Modality routing distribution across tasks of TSROUTER.

### 3.5 Routing under Cost Constraints

In real-world deployment, a practical routing system must balance performance against inference cost. To investigate the impact of costs in routing, we train TSROUTER at varying cost coefficients $\alpha$ in Equation 1 and compare against various baselines. The Pareto-style plots in Figure 5 illustrate a clear trade-off between the answer accuracy and computational cost. When $\alpha = 1.0$, TSROUTER achieves the highest accuracy across all tasks, reflecting its performance-oriented routing preference. As $\alpha$ decreases, TSROUTER gradually shifts toward more cost-efficient decisions, substantially reducing the cost but with a moderate decline in performance. Notably, at a cost budget comparable to the best-performing baseline (Hybrid LLM), TSROUTER still significantly outperforms all baselines in accuracy,

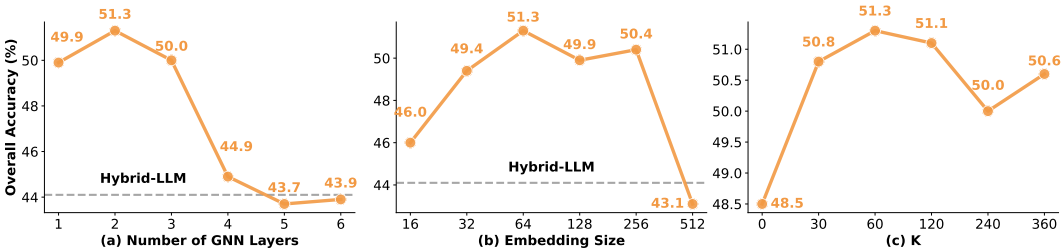


Figure 7: Hyperparameter sensitivity analysis. (a) Number of HGT layers; (b) Embedding dimension; (c) Number of nearest neighbors $k$. The dashed line indicates the best-performing baseline.

indicating that TSROUTER achieves both high performance and cost efficiency, providing practitioners with flexible control over the accuracy-cost operating point.

### 3.6 Ablation Studies

**Effect of the Number of GNN Layers.** The number of GNN layers has a significant impact on the expressiveness of the GNN. A shallow GNN struggles to learn deep contextual information, whereas an overly deep GNN results in performance degradation issues such as oversmoothing (Kipf & Welling, 2016). We conduct an exploration with the number of GNN layers for TSROUTER, ranging from 1 to 6. As shown in Figure 7 (a), the performance initially improves with more layers but then declines significantly, achieving its highest performance when the number of layers is 2.

**Effect of the Amount of Training Data.** We further examine how TSROUTER performs under varying amounts of training data, comparing against the strongest baseline. As shown in Figure 6, TSROUTER trained on only 10% of the data already surpasses the best baseline trained on the full dataset by a large margin, demonstrating the strong data efficiency of our approach.

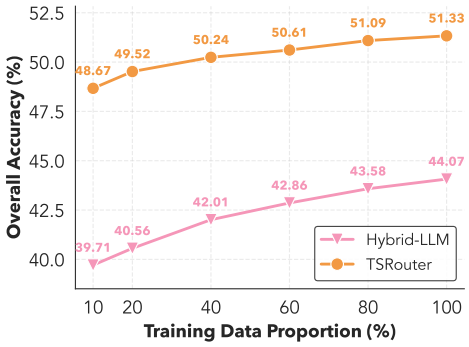


Figure 6: Performance of TSROUTER using different amounts of training data.

**Effect of the Embedding Size of GNN.** The size of a GNN is an important factor that affects both the performance and computational overhead. Figure 7 shows that the performance of TSROUTER initially improves as the embedding size increases, reaching its peak at a size of 64, after which it starts to decline.

**Effect of Components in TSROUTER.** We conduct a series of ablation studies on TSROUTER along the following aspects: (i) **Homogeneous Graph**: setting all edge types identical to reduce to a homogeneous graph, evaluating the benefit of heterogeneous message passing. (ii) **Query-Query Connections**: removing the nearest query–query edges ($\mathcal{E}_{QQ}$) to assess the value of inter-query information sharing; (iii) **Dot-Product Prediction Head**: replacing the MLP-based prediction head (Eq. 4) with a simple dot-product; (iv) **Interactions within Modality-Models**: we merge each modality–model nodes into a single candidate node, eliminating the separate additive composition and the

| Components | Perc. | Reas. | Pred. | Deci. | Over. |
|---|---|---|---|---|---|
| TSROUTER (Full) | **67.63** | **52.81** | **43.11** | **42.45** | **51.33** |
| - Hetero Graph | 64.03 | 49.44 | 41.78 | 37.74 | 46.25 |
| - Query-Query Edges | 65.47 | 49.72 | 39.11 | 42.45 | 48.55 |
| - MLP Scoring Head | 64.03 | 44.10 | 40.44 | 41.51 | 46.52 |
| - Modality-Model Edges | 66.91 | 49.44 | 36.44 | 37.77 | 47.33 |

Table 5: Ablation study of TSROUTER components.

modality–model structural edges. The results shown in Table 5 highlight the criticality of each element in TSROUTER.

## 4 Related Work

**LLMs/VLMs for Time Series.** Beyond traditional statistical methods (RB, 1990; Liu et al., 2021; Lim et al., 2021; Wu et al., 2021; Li et al., 2023; 2024), recent work adapts LLMs for time series via prompting (Cao et al., 2023; Chuang* et al., 2024; Guan et al., 2025) and reprogramming (Jin et al., 2023; Pillai et al., 2025), while a parallel line leverages VLMs by converting time series into visual plots (Chen et al., 2024a; Zhong et al., 2025; Liu et al., 2024; Jiang et al., 2025; Guan et al., 2026). LLMs and VLMs exhibit complementary strengths (Yu et al., 2026b; Kong et al., 2025b): raw numbers excel at precise numerical understanding, while visualizations better capture global patterns. Moreover, individual models vary in expertise, solving different subsets of problems even within the same task. We exploit this complementarity through dynamic, query-level routing that jointly selects the optimal modality and model.

**LLM Router.** LLM routers balance efficiency and accuracy by selecting among multiple LLMs per query, typically via query embeddings or trained classifiers, such as RouterKNN (Shnitzer et al., 2023), RouterMLP (Shnitzer et al., 2023), RouterSVM (Hu et al., 2024), RouterDC (Chen et al., 2024b), RouteLLM (Ong et al., 2024), GraphPlanner (Feng et al., 2026b), GraphRouter (Feng et al., 2024; 2026a), and Hybrid LLM (Ding et al., 2024a). However, these methods focus solely on model selection under a fixed input modality. For instance, GraphRouter formulates routing as edge prediction between query and model nodes, which does not naturally extend to joint modality-model selection. We go beyond by jointly optimizing both dimensions, routing each query to the best modality-model combination.

**LLM/VLM Reasoning.** Recent advances in LLMs and VLMs have substantially improved their capacity for deliberate problem solving. Prompting approaches elicit intermediate reasoning steps (Wei et al., 2022; Zhou et al., 2022; Ho et al., 2025; Yu et al., 2025b; Min et al., 2026; Yu et al., 2026a), but their effectiveness is bounded by the model's intrinsic abilities. Subsequent work therefore turns to test-time compute to increase reasoning depth and broaden exploration of the reasoning space, including sampling strategies (Brown et al., 2024; Karan & Du, 2025; Aggarwal et al., 2023) and structured reasoning frameworks (Yao et al., 2023; Besta et al., 2024; Hao et al., 2023; Ding et al., 2024b). More recently, a growing line of research internalizes reasoning capabilities through reinforcement learning (Yu et al., 2024; 2026d; Cheng et al., 2026; Guo et al., 2025) and on-policy distillation (Zhao et al., 2026; Yang et al., 2026; Shenfeld et al., 2026; Li et al., 2026; Yu et al., 2026c), but at the cost of substantial training effort and computational resources. TSROUTER sits at the intersection of LLM and VLM reasoning, which offers a lightweight solution that leverages the strengths of both for time series reasoning with only minimal training cost.

## 5 Conclusion

We introduce TSROUTER, a dynamic routing framework that optimizes both modality and model selection for time series reasoning. Through a heterogeneous graph that captures interactions among tasks, queries, modalities, and models, TSROUTER outperforms diverse baselines across 4 reasoning tasks in both accuracy and cost-efficiency, while generalizing to unseen tasks and models without retraining.

## References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, et al. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. *arXiv preprint arXiv:2305.11860*, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

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*, 2024a.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328, 2024b.

Jorge Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Nilabjo Dey, Yonghao Zhuang, Yuheng Zha, et al. Revisiting reinforcement learning for llm reasoning from a cross-domain perspective. *Advances in Neural Information Processing Systems*, 38, 2026.

Yu-Neng Chuang*, Songchen Li*, Jiayi Yuan*, Guanchu Wang*, Kwei-Herng Lai*, Leisheng Yu, Sirui Ding, Chia-Yuan Chang, Qiaoyu Tan, Daochen Zha, and Xia Hu. Understanding different design choices in training large time series models. *arXiv preprint arXiv:2406.14045*, 2024.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*, 2024a.

Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Wei Zhang, Si Qin, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. Everything of thoughts: Defying the law of penrose triangle for thought generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1638–1662, 2024b.

Wenjie Du, Jun Wang, Linglong Qian, Yiyuan Yang, Zina Ibrahim, Fanxing Liu, Zepu Wang, Haoxin Liu, Zhiyuan Zhao, Yingjie Zhou, et al. Tsi-bench: Benchmarking time series imputation. *arXiv preprint arXiv:2406.12747*, 2024.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. *arXiv preprint arXiv:2410.03834*, 2024.

Tao Feng, Fangxu Yu, Haozhen Zhang, Zhongjie Dai, Liangqi Yuan, Zijie Lei, Weizhi Zhang, Kunlun Zhu, Haodong Yue, Keyang Xuan, et al. Llmrouter: Unified infrastructure for developing, evaluating, and deploying llm routers. *arXiv preprint arXiv:2608.06867*, 2026a.

Tao Feng, Haozhen Zhang, Zijie Lei, Peixuan Han, and Jiaxuan You. Graphplanner: Graph memory-augmented agentic routing for multi-agent llms. *arXiv preprint arXiv:2604.23626*, 2026b.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

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.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Marco Siniscalchi, Ming Jin, and Shirui Pan. Timeomni-1: Incentivizing complex reasoning with time series in large language models. *arXiv preprint arXiv:2509.24803*, 2025.

Tong Guan, Sheng Pan, Johan Barthelemy, Zhao Li, Yujun Cai, Cesare Alippi, Ming Jin, and Shirui Pan. Timeomni-vl: Unified models for time series understanding and generation. *arXiv preprint arXiv:2602.17149*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Matthew Ho, Chen Si, Zhaoxiang Feng, Fangxu Yu, Zhijian Liu, Zhiting Hu, and Lianhui Qin. Arcmemo: Abstract reasoning composition with lifelong llm memory. *arXiv preprint arXiv:2509.04439*, 2025.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pp. 2704–2710, 2020.

Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. Timexl: Explainable multi-modal time series prediction with llm-in-the-loop. *arXiv preprint arXiv:2503.01013*, 2025.

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. *arXiv preprint arXiv:2310.01728*, 2023.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-mqa: Time series multi-task question answering with context enhancement. *arXiv preprint arXiv:2503.01875*, 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.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

Yuxin Li, Wenchao Chen, Bo Chen, Dongsheng Wang, Long Tian, and Mingyuan Zhou. Prototype-oriented unsupervised anomaly detection for multivariate time series. In *International Conference on Machine Learning*, pp. 19407–19424. PMLR, 2023.

Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Mingyuan Zhou, et al. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

Haoxin Liu, Chenghao Liu, B. Aditya Prakash, Harshavardhan Kamarthi, Lingkai Kong, Zhiyuan Zhao, Chao Zhang, Aditya Prakash, Shangqing Xu, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Jindong Wang, Harshavardhan Kamarthi, Lst-prompt, Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Haoyu Lu, Wen Liu, Bo Zhang, Bing-Yi Wang, Kai Dong, Bo Liu, M. Mathis, Alexander E Webber, Tomás M León, Erin L Murray, Monica Sun, Lauren A White, Logan Brooks, Alden Green, and Addison J. Hu. A picture is worth a thousand numbers: Enabling llms reason about time series via visualization. In *North American Chapter of the Association for Computational Linguistics*, 2024. URL https://api.semanticscholar.org/CorpusID:273963297.

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*, 2025.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2021.

Shengzhong Mao, Chaoli Zhang, Yichi Song, Jindong Wang, Xiao-Jun Zeng, Zenglin Xu, and Qingsong Wen. Time series analysis for education: Methods, applications, and future directions. *arXiv preprint arXiv:2408.13960*, 2024.

Mike A Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. *arXiv preprint arXiv:2404.11757*, 2024.

Dehai Min, Giovanni Vaccarino, Huiyi Chen, Yongliang Wu, Gal Yona, and Lu Cheng. Stop when reasoning converges: Semantic-preserving early exit for reasoning models. *arXiv preprint arXiv:2605.17672*, 2026.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

Arvind Pillai, Dimitris Spathis, Subigya Nepal, Amanda C Collins, Daniel M Mackin, Michael V Heinz, Tess Z Griffin, Nicholas C Jacobson, and Andrew Campbell. Time2lang: Bridging time-series foundation models and large language models for health sensing beyond prompting. *arXiv preprint arXiv:2502.07608*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

CLEVELAND RB. Stl: A seasonal-trend decomposition procedure based on loess. *J Off Stat*, 6:3–73, 1990.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12694–12702, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

A Yarkın Yıldız, Emirhan Koç, and Aykut Koç. Multivariate time series imputation with transformers. *IEEE Signal Processing Letters*, 29:2517–2521, 2022.

Fangxu Yu, Lai Jiang, Haoqiang Kang, Shibo Hao, and Lianhui Qin. Flow of reasoning: Training llms for divergent reasoning with minimal examples. *arXiv preprint arXiv:2406.05673*, 2024.

Fangxu Yu, Hongyu Zhao, and Tianyi Zhou. Ts-reasoner: Aligning time series foundation models with llm reasoning. *arXiv preprint arXiv:2510.03519*, 2025a.

Fangxu Yu, Tao Feng, Dehai Min, Zinan Lin, Weijia Xu, Michael Xu, Philip S Yu, Ge Liu, and Tianyi Zhou. Reinforcement learning with evolving rubrics as rewards for audio reasoning. *arXiv preprint arXiv:2608.02831*, 2026a.

Fangxu Yu, Xingang Guo, Lingzhi Yuan, Haoqiang Kang, Hongyu Zhao, Lianhui Qin, Furong Huang, Bin Hu, and Tianyi Zhou. Tsrbench: A comprehensive multi-task multi-modal time series reasoning benchmark for generalist models. *arXiv preprint arXiv:2601.18744*, 2026b.

Fangxu Yu, Zinan Lin, Xiaodong Liu, Weijia Xu, Michael Xu, Tianyi Zhou, and Jianfeng Gao. Weak-to-strong on-policy distillation. *arXiv preprint arXiv:2607.26246*, 2026c.

Fangxu Yu, Ziyao Lu, Liqiang Niu, Fandong Meng, and Jie Zhou. Arrowgev: Grounding events in video via learning the arrow of time. *arXiv preprint arXiv:2601.06559*, 2026d.

Le Yu, Kaishen Wang, Jianlong Xiong, Yue Cao, Lei Zhang, and Zhang Yi Tao He. Hallurnn: Mitigating hallucinations via recurrent cross-layer reasoning in large vision-language models. *arXiv preprint arXiv:2506.17587*, 2025b.

Runlong Yu, Chonghao Qiu, Robert Ladwig, Paul Hanson, Yiqun Xie, and Xiaowei Jia. Physics-guided foundation model for scientific discovery: An application to aquatic science. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 28548–28556, 2025c.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

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. *arXiv preprint arXiv:2502.04395*, 2025.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

| Variant | Perc. | Reas. | Pred. | Deci. | Over. |
|---|---|---|---|---|---|
| HGT | 67.63 | 52.81 | 43.11 | 42.45 | 51.33 |
| HAN | 71.22 | 51.12 | 43.11 | 42.45 | 51.21 |
| HeteroGAT | 69.06 | 47.75 | 44.89 | 43.40 | 50.00 |

Table 6: Results with different GNN backbone architectures for TSROUTER.

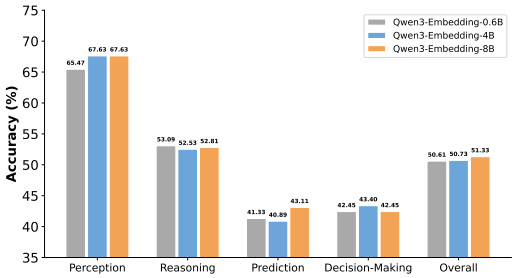


Figure 8: Performance with different embedding models for TSROUTER.

# A Additional Experimental Results

## A.1 Performance under Different GNNs.

To investigate the performance with different GNN backbones, we replace the HGT with two widely used alternatives, HAN and HeteroGAT, while keeping all other components unchanged. As shown in Table 6, all three variants achieve highly competitive overall accuracy, ranging from 50.00% to 51.33%, with a gap of merely 1.33 points between the best and worst configurations. Notably, no single backbone uniformly dominates across all subtasks: HGT excels at reasoning, HAN achieves the highest perception score, and HeteroGAT leads on both prediction and decision-making. This complementary pattern suggests that the routing performance of TSROUTER is primarily driven by the overall heterogeneous graph formulation, which explicitly model the interaction between contextual information, rather than by the specific message-passing mechanism employed.

## A.2 Performance under Different Embedding Models

To investigate the robustness of TSROUTER with different embedding model backbones for node feature initialization, we evaluate TSROUTER across three distinct sizes of the Qwen3-Embedding model: 0.6B, 4B, and 8B. The results shown in Figure 8 confirm that TSROUTER is both highly effective and robust. We observe a positive correlation between overall accuracy and the model size, indicating that the embeddings from larger embedding models are more informative for the following contextual information interactions. Performance remains consistent across all tasks, demonstrating the robustness of the embedding model and representation pipeline. This suggests that the downstream GNN is resilient to moderate variations in input features.

## A.3 Performance when Removing LLM-generated Descriptions

To assess the robustness of TSROUTER to the quality of node descriptions, we remove the LLM-generated descriptions and initialize node features using only the task, modality, and model names. We report results averaged over 3 independent runs in Table 7. Even without the descriptions, TSROUTER exhibits only a minor performance drop and still clearly outperforms the best baseline. This is because the descriptions serve merely as an initial semantic prior for node features, whereas the routing signal is primarily produced by structural message passing over the heterogeneous graph. Consequently, TSROUTER remains effective and highly competitive without relying on LLM-generated descriptions.

| Variant | Perc. | Reas. | Pred. | Deci. | Over. |
|---|---|---|---|---|---|
| TSROUTER w/o description | 66.20$\pm$0.37 | 51.70$\pm$0.17 | 43.40$\pm$0.25 | 40.90$\pm$0.52 | 50.50$\pm$0.34 |
| TSROUTER w/ description | 68.35$\pm$0.72 | 52.34$\pm$0.59 | 42.81$\pm$0.26 | 42.77$\pm$0.55 | 51.21$\pm$0.21 |

Table 7: Results on variants of node text descriptions for TSROUTER.

| Model | Type | Input ($/1M Token) | Output ($/1M Token) | Provider |
|---|---|---|---|---|
| Qwen3-8B | LLM | 0.12 | 0.45 | Alibaba Cloud Int. |
| Qwen3-32B | LLM | 0.16 | 0.64 | Alibaba Cloud Int. |
| LLaMA-3.3-70B-Turbo | LLM | 0.25 | 0.75 | Together AI |
| Qwen3-VL-8B | VLM | 0.12 | 0.45 | Alibaba Cloud Int. |
| Qwen3-VL-32B | VLM | 0.16 | 0.64 | Alibaba Cloud Int. |
| GLM-4.5V | VLM | 0.20 | 1.10 | Together AI |
| Qwen3.5-397B-A17B | LLM | 0.65 | 3.60 | Together AI |
| Kimi-K2.5 | VLM | 0.50 | 2.80 | Together AI |

Table 9: LLMs and VLMs used in TSROUTER with costs per 1M tokens.

### A.4 Efficiency Analysis

In real-world deployment, low routing latency and memory use are essential for responsive systems. We compare TSROUTER with the five strongest learning-based baselines (RouterDC, Hybrid LLM, CausalLM, GraphRouter, and Router-R1) in terms of per-query routing time and memory usage (Table 8). TSROUTER achieves the lowest latency at 3.21 ms per query, approximately 6× faster than RouterDC, 9× faster than Hybrid LLM, and 284× faster than CausalLM. CausalLM incurs the highest cost as it relies on an LLM itself for routing decisions. More broadly, all three baselines need to process each query independently, limiting parallelism. In contrast, the graph-based formulation of TSROUTER naturally supports batch inference, where multiple queries are incorporated into the graph and the GNN operates over the entire graph simultaneously, amortizing overhead across queries and significantly improving the computational efficiency. The memory advantage stems from TSROUTER's lightweight GNN architecture, whereas most baselines load pretrained language models as their routing backbone.

| Method | Inference Time (ms) | Memory Use (MiB) |
|---|---|---|
| CausalLM | 910.78 | 33400 |
| RouterDC | 18.64 | 29358 |
| HybridLLM | 27.65 | 29299 |
| GraphRouter | 4.02 | 1433 |
| Router-R1 | 66.80 | 41177 |
| TSROUTER | **3.21** | **576** |

Table 8: Per-query routing latency comparison.

## B Additional Setup Details

**Description Prompts.** Using descriptions generated by LLM and embeddings derived from a pretrained embedding model as initial node features for GNNs enhances their expressiveness and generalization capability. Here, we list descriptions of different tasks, modalities, and models. The profiles contain only the public attributes without the information of the time series benchmark, which ensures there is no data leakage. The detailed descriptions are shown in the following figures.

**Cost Details.** We report the cost for 1M token input and output. For VLMs, image tokens are billed according to each provider's pricing schedule. We detail the cost information of all the models in Table 9.

**Time Series Visualization.** Each time series is rendered as a line chart image before being passed to VLMs. Given a query with $N$ channels, we plot each channel as a separate subplot, with a shared x-axis (time/index) and individual y-axes. For univariate series, a single plot is generated; for multivariate series, we stack $N$ subplots vertically, each labeled with its channel name. We apply grid lines and bold axis labels to preserve visual clarity. All charts have 100 PPI resolution.

**Case Study**

**Question:** The given time series has multiple trends that follow each other. What is the correct ordering of the trend components?

**Answer Choices:**
(A) Log
(B) Linear→Exponential→Log
(C) Linear→Exponential
(D) Exponential→Linear→Log
**Correct Answer:** (D)

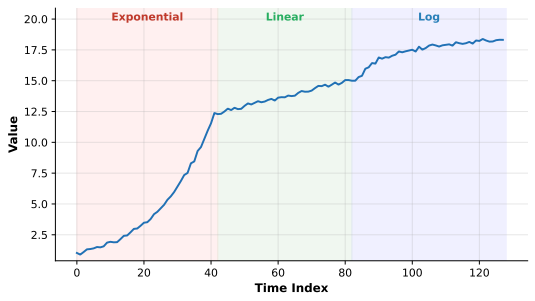


Figure 9: Case study from the Perception task.

| Model (Modality) | Reasoning Process | |
|---|---|---|
| **Qwen3-8B** (Text) | *Ans: (B)* — The time series data shows an initial steady increase (linear trend), followed by a more rapid growth (exponential trend), and eventually a leveling off or slower growth (logarithmic trend). The data starts with a gradual rise, then accelerates, and later appears to stabilize... | ✗ |
| **Qwen3-32B** (Text) | *Ans: (B)* — Observing the given data, it first increases at a steady rate (linear trend), then starts growing faster (exponential trend), and finally the growth slows and stabilizes (log trend). | ✗ |
| **LlaMa-3.3-70B** (Text) | *Ans: (B)* — The time series starts with a steady increase (linear), followed by a rapid growth (exponential), and then a slowing down with minor fluctuations (logarithmic). | ✗ |
| **Qwen3-VL-8B** (Vision) | *Ans: (D)* — The plot shows three distinct phases. Initially (time 0 to ∼40), the series exhibits rapid, accelerating growth — characteristic of an exponential trend. After time 40, the growth rate slows significantly... | ✓ |
| **Qwen3-VL-32B** (Vision) | *Ans: (D)* — The plot shows three distinct phases: 1. Initial phase (0–40): rapid increase with accelerating rate — exponential growth. 2. Middle phase (40–80): growth becomes constant — linear trend. 3. Final phase: curve flattens — logarithmic trend. | ✓ |
| **GLM-4.5V** (Vision) | *Ans: (D)* — Three distinct segments: 1) Initial steep upward curve (exponential), 2) consistent upward slope (linear), 3) gradually decreasing slope (logarithmic). | ✓ |
| **Qwen3-VL-8B** (Text+Vision) | *Ans: (D)* — The data shows a clear progression through multiple trend phases. Initially (to ∼40), rapid accelerating growth characteristic of an exponential trend. After index 40, growth slows down significantly... | ✓ |
| **Qwen3-VL-32B** (Text+Vision) | *Ans: (B)* — The data starts with a slow increase, then accelerates rapidly around index 40–50, and finally stabilizes. The initial phase shows exponential-like growth, but the text data biases toward interpreting it as linear... | ✗ |
| **GLM-4.5V** (Text+Vision) | *Ans: (A)* — Analyzing the data and visualization, I identify three phases. The initial phase (0–30) shows a linear growth pattern with a steady upward slope. The middle phase (30–80) exhibits exponential growth. The final phase (80–130) displays logarithmic growth... | ✗ |

Table 10: Reasoning Trace of each modality-model for the case in Figure 9

## C Time Series Statistical Summary

Each query node embeds the concatenation of the question text $u_i$ and a textual statistical summary of its time series $\mathbf{x}_i$. This summary is appended to the question text before encoding with the pretrained embedding model $\phi$, allowing the router to infer modality preferences from data characteristics.

For each time series $\mathbf{x}_i \in \mathbb{R}^{T_i}$, we compute the following statistics:

- **Mean**: $\bar{x}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} x_{i,t}$.

- **Standard deviation**: $\sigma_i = \sqrt{\frac{1}{T_i}\sum_{t=1}^{T_i}(x_{i,t} - \bar{x}_i)^2}$.
- **Minimum and maximum**: $\min(\mathbf{x}_i)$ and $\max(\mathbf{x}_i)$.
- **Trend direction**: determined by the sign of the slope $\hat{\beta}_1$ from a least-squares linear fit $x_{i,t} \approx \beta_0 + \beta_1 t$, reported as *upward* ($\hat{\beta}_1 > 0$) or *downward* ($\hat{\beta}_1 \leqslant 0$).

These statistics are rendered into a natural language sentence and appended to the query text before embedding.

## D A Case on Reasoning with Different Modalities and Models

We provide a sample question in Figure 9, and report the reasoning trace of different candidates in Table 10. All three LLMs operating on textual time series arrive at the wrong answer, misidentifying the initial exponential phase as linear, likely because the early absolute increments appear small in raw numerical form. In contrast, all three VLMs correctly identify the trend ordering from the visual plot, where the curvature of each regime is directly observable. Interestingly, when both modalities are provided simultaneously, the textual numerical values partially interfere with visual reasoning. For example, Qwen3-VL-32B and GLM-4.5V are both correct under vision-only input, but flip to incorrect answers given the additional textual time series. This case highlights clear differences in reasoning across modalities and models. All text-only LLMs arrive at the wrong answer, while all vision-only VLMs succeed. Furthermore, the smaller Qwen3-VL-8B answers correctly under mixed input while the larger Qwen3-VL-32B does not, confirming that larger models are not always optimal. These observations motivate TSROUTER to jointly select the best modality and model per query.

**Perception Task Description**

Perception tasks focus on observing and describing what is directly visible in a time series. Models are expected to identify surface-level patterns such as trends, periodicity, and anomalies, requiring strong observational and descriptive abilities.

Figure 10: Description of the Perception task.

**Reasoning Task Description**

Reasoning tasks require models to go beyond observation and construct logical explanations or conclusions from time series data. This category broadly covers inferring causes, discovering relationships, applying rules, and performing quantitative analysis, demanding strong analytical and multi-step thinking capabilities.

Figure 11: Description of the Reasoning task.

**Prediction Task Description**

Prediction tasks ask models to forecast future values or outcomes based on historical time series. Models need to capture temporal dependencies and extrapolate patterns, requiring both quantitative estimation and event anticipation skills.

Figure 12: Description of the Prediction task.

**Decision Making Task Description**

Decision-making tasks require models to produce actionable judgments informed by time series analysis. This encompasses both qualitative domain assessments and quantitative decision strategies, calling for the integration of analytical reasoning with domain-specific knowledge.

Figure 13: Description of the Decision-Making task.

**Text Modality Description**

Text modality encodes time series as plain numerical sequences. It emphasizes precise arithmetic and statistical reasoning over raw numeric values, without relying on visual or spatial perception.

Figure 14: Description of the Text modality.

**Visual Modality Description**

Visual modality represents time series as rendered line chart images. It emphasizes spatial perception and visual pattern recognition, requiring models to interpret curves, axes, and chart structures without access to raw numbers.

Figure 15: Description of the Visual modality.

**Both Modality Description**

Mix modality provides both numerical text sequences and chart images simultaneously. It enables cross-modal reasoning by combining token-level arithmetic with pixel-level visual interpretation, allowing models to leverage complementary information from both formats.

Figure 16: Description of using both textual and visual modality.

**Qwen3-8B Description**

A small text-only language model with 8 billion parameters from Alibaba. It offers fast inference at low cost (input: \$0.12/M tokens, output: \$0.45/M tokens), making it suitable for tasks that require quick, cost-efficient responses. Its compact scale trades off depth of reasoning for speed and affordability.

Figure 17: Qwen3-8B description.

**Qwen3-32B Description**

A medium-sized text-only language model with 32 billion parameters from Alibaba. It provides a stronger capacity for analytical and mathematical reasoning than smaller models, at a moderate cost (input: \$0.16/M tokens, output: \$0.64/M tokens). It balances capability and efficiency for tasks requiring careful quantitative analysis.

Figure 18: Qwen3-32B description.

**LLaMA-3.3-70B-Turbo Description**

A large text-only language model with 70 billion parameters from Meta. Its scale provides advanced capabilities for complex multi-step logical inference and nuanced language understanding, at a higher cost (input: \$0.25/M tokens, output: \$0.75/M tokens). It is best suited for demanding tasks where smaller models fall short.

Figure 19: LLaMA-3.3-70B-Turbo description.

**Qwen3-VL-8B-Instruct Description**

A small vision-language model with 8 billion parameters from Alibaba, supporting both visual and multimodal inputs. It offers cost-effective visual understanding at low cost (input: $0.12/M tokens, output: $0.45/M tokens), suitable for straightforward visual perception tasks where computational budget is limited.

Figure 20: Qwen3-VL-8B-Instruct description.

**Qwen3-VL-32B-Instruct Description**

A powerful vision-language model with 32 billion parameters from Alibaba, supporting both visual and multimodal inputs. It combines strong visual perception with solid analytical reasoning capabilities, at a moderate cost (input: $0.16/M tokens, output: $0.64/M tokens). It is well-suited for tasks requiring sophisticated interpretation of visual inputs.

Figure 21: Qwen3-VL-32B-Instruct description.

**GLM-4.5V Description**

A large mixture-of-experts vision-language model with 106 billion parameters from Zhipu AI, supporting both visual and multimodal inputs. Its MoE architecture enables efficient scaling with strong visual and domain-specific capabilities, at a higher cost (input: $0.20/M tokens, output: $1.10/M tokens). It is designed for tasks demanding nuanced visual judgment and quantitative analysis.

Figure 22: GLM-4.5V description.

**Qwen3.5-397B-A17B Description**

A very large mixture-of-experts text-only language model with 397 billion parameters from Alibaba, with 397 billion total parameters and 17 billion active parameters per forward pass. Its MoE architecture delivers strong reasoning capabilities at a relatively moderate active compute cost, though pricing reflects its scale (input: $0.65/M tokens, output: $3.60/M tokens). It is designed for complex analytical tasks requiring deep multi-step thinking.

Figure 23: Qwen3.5-397B-A17B description.

**Kimi-K2.5 Description**

A large vision-language model from Moonshot AI with built-in chain-of-thought reasoning capabilities, supporting both visual and multimodal inputs. It combines strong visual perception with multi-step logical reasoning, at a moderate cost (input: $0.5/M tokens, output: $2.80/M tokens). It is suited for tasks requiring integrated visual understanding and structured inference.

Figure 24: Kimi-K2.5 description.