VLM4ST: REPURPOSING VISION-LANGUAGE MODELS FOR SPATIOTEMPORAL FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatiotemporal (ST) forecasting requires modelling spatial dependencies and temporal dynamics in high-dimensional fields. Adapting pretrained large language models offers transferable sequence representations, but tokenising spatial fields into 1D sequences can obscure spatial topology. Recent vision-language model (VLM) approaches encode images of hand-crafted statistics alongside fixed textual templates, leaving the correspondence between VLM architecture and ST structure largely unexploited. In this work, we propose **VLM4ST**, a framework that aligns the dual-path architecture of VLMs with spatial and temporal modelling. A 2D visual encoder processes spatial fields, while a 1D sequential encoder captures temporal dynamics. Multi-layer cross-attention couples both encoders, preserving spatial topology while leveraging pretrained sequence representations. To bridge the gap between natural image–text pretraining and ST data, we introduce **Prompt Forge**, a hierarchical pattern-memory module that learns latent ST prototypes and generates task-adaptive continuous prompts for cross-modal fusion. These prompts encode temporal periodicity, spatial correlations, and cross-dimensional dependencies. Extensive experiments show that VLM4ST outperforms task-specific, LLM-based, and recent VLM-based predictors, reducing short-term MAE by up to 16.5% (12.3% on average) relative to the strongest baseline on each dataset, with substantial gains in few-shot and zero-shot transfer. Code is available at https://anonymous.4open.science/r/VLM4ST/.

1 INTRODUCTION

Urban spatiotemporal (ST) data underpins a wide range of high-stakes applications in transportation, mobility, and urban risk management. Accurately forecasting from historical ST observations is therefore essential for decision-making, resource allocation, and risk mitigation. However, ST forecasting remains challenging because ST observations are neither simple time series nor static spatial data: they jointly encode high-dimensional spatial fields and long-range temporal dynamics (Chen et al., 2023). This dual nature requires models to capture spatial layouts and temporal evolution with equal fidelity.

Limitations of Existing Works. Traditional ST forecasting relies on task-specific graph, transformer, and convolutional-recurrent architectures (Yu et al., 2018; Wu et al., 2019; Shi et al., 2015; Wang et al., 2017; Li et al., 2024a), whose generalization can be limited under domain shift and data scarcity. Recent approaches adapt pretrained large language models (LLMs) through prompting, reprogramming, or tokenization to improve transferability (Jin et al., 2023; Liu et al., 2025; Li et al., 2024b). Although pretrained sequence representations support temporal modeling, the resulting 1D token interface does not explicitly preserve the native 2D spatial layout, requiring additional mechanisms to capture spatial structure. Recent VLM-based methods incorporate correlation images or rendered frames alongside historical observations and fixed textual prompts (Zhao et al., 2026; Yang et al., 2025). However, these designs do not explicitly align the VLM's visual and sequential pathways with spatial field encoding and temporal dynamics, respectively. This leaves a key question: *how can a pretrained VLM's architecture be aligned with the complementary spatial and temporal requirements of ST forecasting?*

Motivation. Our key observation is that the visual and sequential pathways of VLMs offer a natural architectural match to the spatial and temporal structure of ST data. VLMs combine visual and

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.

textual representations (Radford et al., 2021; Li et al., 2021a; 2022), and fusion-based models such as ALBEF and BLIP further enable cross-modal interaction through cross-attention. As illustrated in Figure 1, the visual pathway can encode 2D spatial layouts, while the textual pathway can be repurposed to model temporal dependencies. Cross-modal attention then offers a mechanism for coupling spatial patterns with temporal evolution. Beyond this architectural correspondence, VLM pretraining provides transferable representations for ST forecasting.

Challenges. Despite this affinity, adapting VLMs to ST forecasting is non-trivial. First, ST coupling is tighter and more structured than vision-text alignment, requiring explicit mechanisms to preserve cross-dimensional dependencies. Second, natural image-text pretraining differs substantially from continuous ST tensors, creating a domain gap that calls for ST inductive biases such as periodicity and spatial correlation, while preserving the pretrained visual-semantic knowledge through parameter-efficient adaptation. Third, VLM input interfaces are incompatible with raw ST observations: VLMs expect RGB images and discrete text tokens, whereas ST data arrives as continuous tensors. Naive channel replication or sequence flattening can destroy the spatial and temporal structures that motivate using VLMs in the first place, making careful format transformation essential.

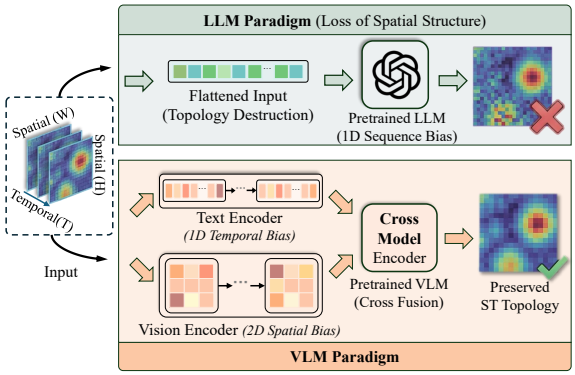


Figure 1: Comparison of LLMs and VLMs architectures for spatiotemporal forecasting.

Our Solution. To address these challenges, we propose VLM4ST, a framework that repurposes **V**ision-**L**anguage **M**odels **for S**patio**T**emporal forecasting by aligning their architecture with ST data. The visual encoder encodes a spatially structured representation of the ST field, while the text encoder is repurposed as a temporal sequence encoder without natural-language input. The two pathways interact through multi-layer cross-attention. VLM4ST combines three tailored modules with LoRA-based adaptation. First, we design a Disentangled Cross-Attention (DiX-Attention) module that separately extracts temporal features, spatial features, and explicit ST coupling through dedicated attention pathways. Second, we introduce Prompt Forge, a hierarchical pattern-memory module that constructs learnable prototype banks at multiple granularities and generates task-adaptive continuous prompts to inject temporal, spatial, and cross-dimensional ST patterns into the VLM. Third, we propose a Dual-Path Format Transformer that converts ST tensors into VLM-compatible inputs through two complementary pathways: a visual path that preserves spatial resolution by compressing temporal information into pseudo-RGB channels, and a sequence path that preserves temporal resolution through spatial pooling and sequentialization. Finally, we adopt LoRA-based adaptation by freezing most VLM parameters and injecting low-rank matrices into attention layers, enabling parameter-efficient adaptation to ST data. The contributions of this work can be summarized as follows:

- We identify an architecture-level correspondence between dual-stream VLMs and ST data and instantiate it in **VLM4ST**. Unlike prior VLM-based ST predictors that encode hand-crafted images with fixed text prompts, VLM4ST operates directly on the native spatial layout and requires no natural-language input.
- We develop a structured adaptation pipeline for ST data, integrating disentangled spatiotemporal encoding, dual-path format transformation, and **Prompt Forge**, which generates task-adaptive continuous prompts to bridge domain gap between pretrained VLMs and urban ST observations.
- Extensive experiments on heterogeneous urban benchmarks demonstrate consistent gains over task-specific, LLM-based, and recent VLM-based predictors across short-term, long-term, few-shot, and zero-shot settings, with ablations confirming the effectiveness of dual-path modeling and pattern-aware prompting.

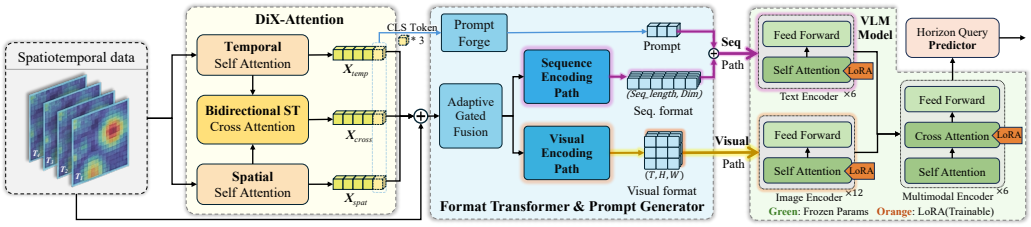


Figure 2: Overview of the VLM4ST framework architecture. The pipeline transforms raw spatiotemporal observations through disentangled feature extraction, dual-path format transformation, vision-language fusion, and horizon query-based forecasting.

2 METHOD

Problem Setup & Framework Overview. Given a spatiotemporal observation tensor $\mathbf{X} \in \mathbb{R}^{T \times H \times W}$ over T historical steps and a spatial grid of height H and width W, the goal is to predict future states $\mathbf{Y} \in \mathbb{R}^{T' \times H \times W}$ for the next T' steps. This formulation covers diverse applications, where the spatial grid captures geographic structure and temporal sequences encode dynamic evolution. As shown in Fig. 2, VLM4ST framework consists of four stages. First, a Disentangled Cross Attention (DiX-Attention) module models spatial, temporal, and cross-dimensional dependencies through parallel attention pathways. Second, Format Transformer and Prompt Generator fuse multi-dimensional features and convert them into VLM-compatible visual and sequential representations, while generating task-adaptive prompts from learned pattern memories. Third, a pretrained VLM backbone performs multimodal fusion between spatial and temporal representations, with parameter-efficient adaptation enabled via LoRA. Finally, a Horizon Query Predictor decodes future states using horizon-aware query tokens, producing adaptive spatiotemporal forecasts.

2.1 DIX-ATTENTION ARCHITECTURE

Traditional spatiotemporal models jointly model temporal and spatial dependencies via unified operations such as 3D convolutions or spatiotemporal attention, which can obscure dimension-specific patterns by entangling temporal dynamics with spatial heterogeneity. To overcome this limitation, we propose DiX-Attention module, which explicitly decouples temporal modeling, spatial modeling, and their cross-dimensional interactions into three pathways.

Temporal Self-Attention. The temporal encoder models how each spatial location evolves over time by treating the temporal sequence at each grid position as an independent sequence. Given a mini-batch $\mathbf{X} \in \mathbb{R}^{B \times T \times H \times W}$ (where B denotes the batch size), we first map each spatiotemporal position to a d-dimensional embedding, yielding $\mathbf{X}_{input} \in \mathbb{R}^{B \times T \times H \times W \times d}$. For each spatial location (h, w), we treat its temporal trace as a sequence $\mathbf{x}_{h,w} \in \mathbb{R}^{B \times T \times d}$ and apply multi-head self-attention with learnable positional encodings; the same temporal encoder is shared across all locations. This produces temporally-enhanced features $\mathbf{X}_{\text{temp}} \in \mathbb{R}^{B \times T \times H \times W \times d}$ where each position's representation incorporates context from its temporal neighbors. A learnable classification (**CLS**) token aggregates the global temporal pattern for each spatial location, and we average across spatial positions to obtain a sample-level temporal summary $\mathbf{CLS}_{\text{temp}} \in \mathbb{R}^{B \times d}$.

Spatial Self-Attention. Complementarily, the spatial encoder captures how spatial patterns organize at each time step by treating each temporal snapshot as an independent spatial grid. For each time step t, we flatten the spatial grid to obtain $\mathbf{x}_t \in \mathbb{R}^{B \times (H \cdot W) \times d}$ and apply multi-head self-attention, allowing each spatial location to attend to all others within the same time step. The spatial encoder similarly produces spatially-enhanced features $\mathbf{X}_{\text{spat}} \in \mathbb{R}^{B \times T \times H \times W \times d}$ and a global spatial summary $\mathbf{CLS}_{\text{spat}} \in \mathbb{R}^{B \times d}$ via **CLS** token aggregation across time steps.

Bidirectional Cross-Attention. While the temporal and spatial encoders capture within-dimension dependencies, real spatiotemporal phenomena exhibit rich cross-dimensional interactions. We model these interactions through bidirectional cross-attention comprising two complementary mechanisms. **(1)** In the temporal-to-spatial branch, temporal representations query spatial features to capture temporally-driven spatial variations: we form global temporal queries by averaging $\mathbf{X}_{\text{temp}}$ over spatial dimensions, obtaining $\mathbf{q}_{t2s} \in \mathbb{R}^{B \times T \times d}$, which attend to spatially pooled key-value pairs derived

from $\mathbf{X}_{\text{spat}}$ (pooling is applied to reduce computational complexity). **(2)** In the spatial-to-temporal branch, spatial patterns query temporal features to model location-specific temporal behavior: we compute spatial queries $\mathbf{q}_{s2t} \in \mathbb{R}^{B\times H\times W\times d}$ by averaging $\mathbf{X}_{\text{spat}}$ across time, and let them attend to temporal key-value pairs from $\mathbf{X}_{\text{temp}}$ to relate each location's dynamics to the overall temporal context. The two cross-attention outputs are concatenated and projected to obtain cross-dimensional feature $\mathbf{X}_{\text{cross}} \in \mathbb{R}^{B\times T\times H\times W\times d}$, with an aggregated **CLS** representation $\mathbf{CLS}_{\text{cross}} \in \mathbb{R}^{B\times d}$.

Overall, DiX-Attention outputs three complementary feature sets ($\mathbf{X}_{\text{temp}}$, $\mathbf{X}_{\text{spat}}$, $\mathbf{X}_{\text{cross}}$) along with three global pattern summaries ($\mathbf{CLS}_{\text{temp}}$, $\mathbf{CLS}_{\text{spat}}$, $\mathbf{CLS}_{\text{cross}}$) that serve as inputs to the pattern memory module for prompt generation. These representations are then adaptively fused into a unified spatiotemporal embedding $\mathbf{X}_{\text{fused}}$ via a gated fusion module that balances temporal, spatial, and cross-dimensional cues. Details are provided in Appendix D.1.

2.2 Prompt Forge

A fundamental challenge in adapting VLMs to spatiotemporal forecasting lies in the domain gap between natural vision-language data and structured sensor observations. While VLMs are trained on natural images paired with semantic text, spatiotemporal data consists of numerical tensors without explicit semantic annotations. Bridging this gap requires injecting domain-specific knowledge about spatiotemporal patterns into the VLM processing pipeline. We address this through Prompt Forge module, which constructs learnable prototype banks encoding common spatiotemporal patterns and generates dynamic prompts that guide VLM's interpretation of the input, as shown in Figure 3.

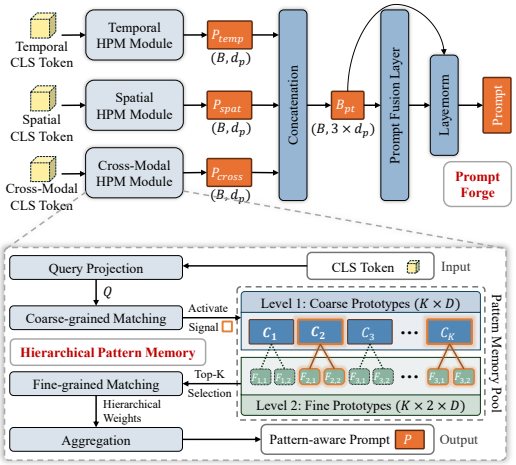


Figure 3: Architecture of Prompt Forge (upper) and Hierarchical Pattern Memory (HPM, Lower) Module. The Prompt Forge generates task-adaptive prompts by fusing outputs from three HPM modules. Each HPM performs two-level coarse-to-fine prototype matching to capture pattern-aware representations from the corresponding CLS tokens.

Hierarchical Pattern Memory. Spatiotemporal patterns exhibit natural hierarchical structure. Temporal patterns decompose into coarse categories (e.g., periodic vs. trending vs. stationary) and finer subcategories (e.g., daily vs. weekly periodicity, linear vs. exponential trends). Spatial patterns similarly range from global distributions to localized anomalies. This hierarchical organization motivates our design of the Hierarchical Pattern Memory, which maintains a two-level prototype structure for each dimension. For each of the three feature dimensions (temporal, spatial, and cross-modal), we construct an independent memory pool containing K coarse-grained prototype vectors $\{\mathbf{p}_k^c\}_{k=1}^K$, each representing a major pattern category. Within each coarse category k, we further define M fine-grained sub-prototypes $\{\mathbf{p}_{k,m}^f\}_{m=1}^M$ that capture variations within the category, as shown in Figure 3. All prototypes are learnable parameters initialized using orthogonal initialization and optimized end-to-end during training.

Memory-Based Prompt Generation. Given the CLS tokens extracted from the DiX-Attention module, the pattern memory generates task-adaptive prompts through hierarchical similarity matching. Taking temporal dimension as an example, we first compute similarity between temporal CLS token and each coarse prototype. The coarse weights $\{w_k^c\}$ identify which major pattern categories are relevant to the current input, where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and τ is a temperature hyperparameter. Within the selected coarse categories (those with cumulative weight exceeding a threshold), we perform fine-grained matching weights $\{w_{k,m}^f\}$ to identify specific sub-patterns:

$$w_k^c = \frac{\exp(\text{sim}(\mathbf{CLS}_{\text{temp}}, \mathbf{p}_k^c)/\tau)}{\sum_{j=1}^K \exp(\text{sim}(\mathbf{CLS}_{\text{temp}}, \mathbf{p}_j^c)/\tau)}, \quad w_{k,m}^f = \frac{\exp(\text{sim}(\mathbf{CLS}_{\text{temp}}, \mathbf{p}_{k,m}^f)/\tau)}{\sum_{j=1}^M \exp(\text{sim}(\mathbf{CLS}_{\text{temp}}, \mathbf{p}_{k,j}^f)/\tau)}$$

The final prompt for temporal dimension is computed as a weighted combination of all fine-grained prototypes, where combined weight reflects both category membership and sub-pattern relevance:

$$\mathbf{P}_{\text{temp}} = \sum_{k=1}^{K} \sum_{m=1}^{M} w_k^c \cdot w_{k,m}^f \cdot \phi(\mathbf{p}_{k,m}^f)$$

where $\phi(\cdot)$ is a learnable projection to the prompt dimension.

The three dimension-specific prompts—$\mathbf{P}_{\text{temp}}$, $\mathbf{P}_{\text{spat}}$, and $\mathbf{P}_{\text{cross}}$—are concatenated and passed through a fusion layer to produce the unified pattern prompt $\mathbf{B}_{\text{prompt}}$, which encodes a holistic description of the current spatiotemporal characteristics.

2.3 Dual-Path Format Transformation

Converting spatiotemporal tensors to VLM-compatible formats presents a fundamental trade-off: image representations naturally preserve two-dimensional spatial structure but require temporal compression, while sequence representations naturally preserve sequential ordering but require spatial flattening. Existing single-path approaches inevitably sacrifice information in one dimension, see Appendix E for theoretical analysis. Our dual-path design resolves this trade-off by generating both formats simultaneously, ensuring comprehensive information preservation.

Visual Path. The visual path transforms the fused spatiotemporal features $\mathbf{X}_{\text{fused}} \in \mathbb{R}^{T\times H\times W\times d}$ into a three-channel pseudo-image $\mathbf{D} \in \mathbb{R}^{3\times H\times W}$ suitable for the visual encoder, without catastrophically losing temporal information. We employ a temporal attention mechanism that learns to intelligently weight the contribution of each time step to each output channel. First, we reduce the feature dimension through a convolutional projection to obtain $\mathbf{X}_{\text{reduced}} \in \mathbb{R}^{T\times H\times W}$. We then compute attention logits through a temporal convolution layer:

$$\mathbf{A} = \text{SoftMax}_T(\text{Conv}(\mathbf{X}_{\text{reduced}})) \in \mathbb{R}^{3\times T\times H\times W}$$

where the softmax is applied along the temporal dimension, ensuring that each output channel attends to a normalized mixture of input time steps. The attended output is computed as:

$$\mathbf{D}_{\text{attn}} = \sum_{t=1}^{T} \mathbf{A}_{:,t,:,:} \odot \mathbf{X}_{\text{reduced}}^{(t)}$$

To preserve temporal statistics that may be lost through soft attention, we augment the representation with explicit statistical features: the temporal mean (capturing the average state), standard deviation (capturing variability), and trend (capturing directional change between first and last time steps). These statistics are fused with the attention output through a learned combination:

$$\mathbf{D} = \lambda_1 \mathbf{D}_{\text{attn}} + \lambda_2 \mathbf{D}_{\text{proj}} + \lambda_3 \mathbf{D}_{\text{stat}}$$

where $\mathbf{D}_{\text{proj}}$ is a direct linear projection and $\mathbf{D}_{\text{stat}}$ encodes the statistical features. The resulting output is resized to match the VLM's expected resolution.

Sequence Path. The sequence path converts spatiotemporal features into a sequence format that preserves the complete temporal dimension while accepting controlled spatial compression. The strategy proceeds through spatial pooling followed by Perceiver-style compression. Then compressed sequence is projected to the VLM's hidden dimension through a linear layer with layer normalization. Finally, we prepend a learnable CLS token to the sequence and insert the pattern prompt (projected from $\mathbf{B}_{\text{prompt}}$) as a special token, yielding the final sequence representation $\mathbf{C}$ ready for the text encoder. Additional details are provided in Appendix D.2.

2.4 Cross-Modal Fusion via VLM

We adopt ALBEF (Li et al., 2021a) as our backbone due to its demonstrated effectiveness in multimodal alignment and fusion. ALBEF comprises a visual encoder based on Vision Transformer (ViT) (Dosovitskiy, 2020) and a text encoder based on BERT (Devlin et al., 2019), with cross-attention layers enabling interaction between modalities.

The visual path output $\mathbf{D}$ is processed by the visual encoder to produce patch-level visual embeddings:

$$\mathbf{E}_{\text{visual}} = \text{ViT}(\mathbf{D}) \in \mathbb{R}^{(N_p+1)\times D_{\text{vlm}}}$$

Table 1: Short-term forecasting results. The best and second-best results are highlighted in bold and underline, respectively.

Model	BikeNYC$_{in}$		BikeNYC$_{out}$		TaxiNYC$_{in}$		TaxiNYC$_{out}$		TrafficHZ		TrafficTJ		TrafficZZ	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Heuristic approaches														
HA	3.42	11.48	7.22	15.63	18.56	57.03	15.74	52.84	0.346	0.767	0.825	1.619	0.864	1.472
ARIMA	3.90	12.30	13.63	25.02	20.93	55.48	29.54	62.89	0.373	0.793	1.588	2.011	1.004	1.786
Deep spatiotemporal approaches														
PredRNN	1.814	5.109	3.082	6.461	5.799	16.59	7.239	19.86	0.289	0.594	0.538	0.968	0.518	0.843
MIM	1.627	4.328	2.970	6.335	6.821	18.87	6.627	18.04	1.229	1.916	2.504	3.449	1.557	2.059
TAU	1.838	4.232	2.993	5.991	6.776	16.91	6.753	15.28	0.299	0.536	0.597	1.002	0.529	0.845
MC-STL	2.101	4.065	3.376	6.220	9.512	18.36	8.422	16.81	0.594	0.843	0.860	1.221	0.803	1.145
STGSP	1.718	4.993	7.449	14.18	10.22	28.09	10.10	29.03	0.340	0.606	0.561	1.001	0.511	0.825
PromptST	1.613	4.378	2.494	5.574	5.859	16.08	5.580	14.06	<u>0.232</u>	<u>0.526</u>	0.534	0.970	0.512	0.831
ST-ResNet	4.986	8.268	3.959	7.186	17.94	29.45	12.05	22.18	0.402	0.673	0.716	1.121	0.697	1.032
STGCN	2.738	5.514	3.544	6.999	6.983	20.05	7.554	19.13	0.350	0.751	0.829	1.169	0.759	1.130
UniST	1.530	3.919	2.617	5.564	5.868	16.12	5.483	14.46	0.250	0.534	0.510	0.949	0.490	0.845
LLM-based approaches														
ST-LLM	1.957	4.044	2.991	6.951	7.042	17.93	6.764	19.45	0.409	0.731	0.826	1.189	0.835	1.548
ST-LLM+	1.403	3.747	2.685	6.453	6.188	17.11	5.265	13.50	0.382	0.695	0.712	1.140	0.790	1.154
ST-LINK	1.333	3.741	3.238	7.276	7.722	18.24	6.536	17.56	0.365	0.644	0.742	1.195	0.771	1.162
UrbanGPT	1.548	3.801	2.927	7.002	5.654	15.48	6.661	13.51	0.352	0.545	0.685	1.015	0.778	1.197
VLM-based approaches														
ST-Vision-LLM	1.216	3.402	2.492	5.606	5.351	14.55	4.950	12.75	0.243	0.529	<u>0.492</u>	<u>0.861</u>	<u>0.468</u>	<u>0.813</u>
ST-VLM	<u>1.172</u>	<u>3.188</u>	<u>2.374</u>	<u>5.152</u>	<u>5.173</u>	<u>13.65</u>	<u>4.628</u>	<u>11.50</u>	0.317	0.530	0.602	0.865	0.554	0.829
VLM4ST (ours)	**1.018**	**2.772**	**2.063**	**4.384**	**4.637**	**12.10**	**3.866**	**9.918**	**0.217**	**0.461**	**0.437**	**0.756**	**0.397**	**0.719**

where N_p is the number of image patches, D_{vlm} denotes the hidden dimension of the VLM.

The sequence path features $\mathbf{C}$, augmented with the pattern prompt $\mathbf{B}_{\text{prompt}}$, are fed directly into the text encoder as continuous embeddings, bypassing the standard tokenization layer. This direct embedding injection is crucial: our spatiotemporal representations are dense continuous vectors rather than discrete tokens, and forcing them through a tokenizer designed for natural language would destroy semantic content. The text encoder operates in fusion mode, where the initial layers process the sequence independently and subsequent layers perform cross-attention with visual embeddings:

$$\mathbf{E}_{\text{fused}} = \text{BERT}_{\text{fusion}}([\text{CLS}; \mathbf{B}_{\text{prompt}}; \mathbf{C}], \mathbf{E}_{\text{visual}}) \in \mathbb{R}^{L' \times D_{\text{vlm}}}$$

The resulting fused embeddings integrate information from both the visual and sequential pathways, capturing spatiotemporal patterns through cross-modal attention. These fused representations are subsequently fed into a horizon-query predictor, where learnable future-step queries attend to the VLM embeddings and decode multi-step spatial predictions. See Appendix D.3 for predictor details.

3 Experiments

3.1 Experimental Setup

Datasets and Metrics. We evaluate VLM4ST on representative spatiotemporal benchmarks widely used in prior studies (Yuan et al., 2024), covering heterogeneous urban sensing scenarios such as traffic dynamics, taxi mobility, and bike sharing across multiple cities. These datasets vary in spatial structure, temporal resolution, and scale, enabling evaluation under diverse real-world conditions; statistics are provided in Appendix F. All datasets are split into 70%/15%/15% for training, validation, and testing. Inputs are normalized to $[-1, 1]$ during training, while MAE and RMSE are computed on denormalized predictions to measure average forecasting error and large-deviation sensitivity, respectively.

Baselines and Implementation Details. We compare VLM4ST against representative baselines spanning heuristic/statistical methods (HA and ARIMA (Box et al., 2015)), RNN-based models

(PredRNN (Wang et al., 2017) and MIM (Wang et al., 2019)), CNN-based models (ST-ResNet (Zhang et al., 2017)), attention-based models (TAU (Tan et al., 2023), STGSP (Zhao et al., 2022), UniST (Yuan et al., 2024), and PromptST (Zhang et al., 2023b)), GNN-based models (STGCN (Yu et al., 2018) and MC-STL (Zhang et al., 2023a)), LLM-based models (ST-LLM (Liu et al., 2024), ST-LLM+ (Liu et al., 2025), UrbanGPT (Li et al., 2024b), and ST-LINK (Jeon et al., 2025)), and VLM-based models (ST-Vision-LLM (STV.-LLM) (Yang et al., 2025) and ST-VLM (Zhao et al., 2026)). See Appendix G for baseline details. For graph-based baselines, the original $H \times W$ grids are flattened into $S = H \times W$ nodes. A four-neighbor adjacency matrix is constructed, and temporal features are appended to node representations. For LLM-based baselines, we follow their original settings: ST-LLM and ST-LLM+ use GPT-2 and LLaMA-2, UrbanGPT uses Vicuna v1.5, and ST-LINK uses GPT-2. The VLM-based baselines are reimplemented following their published designs and evaluated with the same splits, horizons, and metrics. For our method, the BERT and ViT components are initialized with pretrained weights from ALBEF-14M. The temperature parameter in Prompt Forge is set to 0.1. AdamW with a cosine learning rate scheduler is used for optimization. All experiments are conducted on NVIDIA GH200 GPUs with 96 GB memory. All reported results are averaged over three runs with different random seeds.

3.2 Evaluation against State-of-the-art Baselines

To comprehensively evaluate the forecasting capability of VLM4ST, we compare it against a broad spectrum of baselines under both short-term and long-term forecasting scenarios.

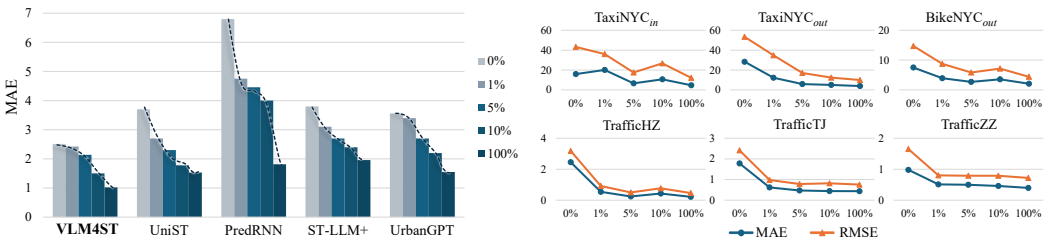
Table 2: Long-term forecasting results.

Model	BikeNyc$_{in}$		BikeNyc$_{out}$		TaxiNyc$_{in}$		TaxiNyc$_{out}$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Heuristic approaches								
HA	3.73	10.95	8.08	15.92	21.22	60.96	19.52	57.32
ARIMA	4.02	11.61	7.22	15.37	28.67	68.07	26.88	63.52
Deep spatiotemporal approaches								
PredRNN	2.06	5.96	4.38	9.64	7.21	21.05	7.02	20.08
MIM	2.79	7.57	4.64	10.06	29.84	63.27	26.39	59.36
TAU	1.68	3.88	6.10	11.49	11.04	24.87	9.30	21.27
MC-STL	2.54	5.06	7.94	12.71	17.28	29.32	15.20	27.78
STGSP	1.51	4.82	7.61	14.61	10.46	28.55	8.99	26.90
PromptST	1.54	3.69	4.14	9.71	6.90	19.91	7.11	19.21
ST-ResNet	3.80	7.13	7.10	12.80	14.40	29.47	12.71	26.87
STGCN	3.05	10.27	7.30	13.92	18.29	37.56	17.14	32.42
UniST	1.69	3.64	5.08	11.83	6.80	19.95	6.21	19.54
LLM-based approaches								
ST-LLM	2.72	5.91	5.22	11.34	8.36	25.56	7.40	20.01
ST-LLM+	2.27	5.08	4.83	10.73	7.82	24.00	6.87	16.32
ST-LINK	1.43	3.58	4.68	9.47	7.89	26.07	6.81	18.50
UrbanGPT	1.86	4.10	5.21	10.78	6.60	19.13	7.01	19.49
VLM-based approaches								
STV.-LLM	1.49	3.84	4.32	9.78	6.63	19.75	6.42	17.45
ST-VLM	1.41	3.55	4.15	9.52	6.42	19.07	6.14	16.30
VLM4ST (ours)	**1.31**	**3.16**	**3.67**	**8.14**	**5.70**	**16.95**	**5.61**	**14.91**

Results of Short-term Forecasting. Table 1 presents the short-term forecasting performance across seven benchmark datasets, where models are tasked with predicting the next 6 time steps given 6 historical observations. VLM4ST consistently achieves the lowest MAE and RMSE across all datasets and both metrics, reducing MAE by 12.3% and RMSE by 12.7% on average relative to the strongest baseline on each dataset. On BikeNYC, VLM4ST attains MAE values of 1.018 and 2.063 for inflow and outflow, both 13.1% lower than the strongest competitor, ST-VLM. On the traffic speed tasks TrafficTJ and TrafficZZ, it obtains MAE values of 0.437 and 0.397, outperforming the strongest competitor, ST-Vision-LLM, by 11.2% and 15.2%.

Results of Long-term Forecasting. Table 2 reports the long-term forecasting results, where models must forecast 64 future time steps based on 64 historical observations. Despite the increased difficulty, VLM4ST maintains its superiority across all evaluation metrics, reducing MAE by 9.6% and RMSE by 11.2% on average relative to the strongest baseline on each dataset. On BikeNYC, our method achieves MAE values of 1.31 and 3.67 for inflow and outflow forecasting, representing improvements of 7.1% and 11.4% over ST-VLM and PromptST. For TaxiNYC, VLM4ST obtains MAE values of 5.70 and 5.61, outperforming the strongest competitor, ST-VLM, by 11.2% and 8.6%.

Analysis. First, heuristic methods perform worst in most settings. Second, deep spatiotemporal architectures are competitive on certain datasets but not consistently across benchmarks, suggesting dataset-specific inductive biases. Third, LLM-based approaches show mixed performance: the mismatch between sequential language modeling and 2D spatial structure limits fine-grained spatial reasoning. Fourth, VLM-based predictors generally outperform LLM-based counterparts, suggesting the benefit of incorporating visual representations. Relative to the strongest VLM-based baseline on each dataset, VLM4ST further reduces MAE by 10.4%–16.5% in short-term and 7.1%–11.6% in long-term forecasting. These gains suggest that effective VLM adaptation depends on how ST structure is represented and coupled, consistent with the benefits of spatially structured field inputs, explicit temporal encoding, and sample-adaptive prompts.



(a) Few/zero-shot performance (MAE) comparison on BikeNYC$_{in}$. Each bar group represents a method, with bars indicating different training data ratios. Dashed lines show the performance trend for each method.

(b) VLM4ST few/zero-shot performance across six benchmark datasets. MAE (blue circles) and RMSE (orange triangles) are reported for each dataset from zero-shot (0%) to full supervision (100%).

Figure 4: Few/zero-shot performance

3.3 Evaluation under Few-shot and Zero-shot Protocols

To evaluate VLM4ST's generalization under data-scarce conditions, we perform experiments with few-shot and zero-shot transfer protocols, simulating practical settings where labeled target-domain data is limited or unavailable. In the few-shot setting, models are trained with 1%, 5%, or 10% of the target data after pretraining on auxiliary datasets. In the zero-shot setting (0%), models are trained only on auxiliary datasets, which exclude the target and its same-source counterpart (Appendix F), and evaluated directly on the held-out target.

Results of Few/Zero-Shot Forecasting. Figure 4a compares VLM4ST with representative baselines on BikeNYC$_{in}$ under varying data availability settings. VLM4ST consistently achieves the lowest MAE, indicating superior sample efficiency. In the zero-shot setting (0%), it reaches an MAE of 2.51, substantially outperforming all competitors and even surpassing PredRNN trained with 5% or 10% target data. With 5% data, VLM4ST attains 2.14 MAE, approaching its full-data result (1.018) markedly faster than other methods. The trend lines in Figure 4a further show that, although all methods benefit from more data, VLM4ST exhibits the smallest gap between zero-shot and full-data performance. Figure 4b summarizes VLM4ST's few-shot results on the remaining six datasets. Performance generally improves with more target data, with the largest gains from 0% to 1%, suggesting that minimal supervision enables effective adaptation. On traffic speed datasets, VLM4ST is near-optimal with only 1% data, implying strong transferability from pretraining. Notably, VLM4ST with 5% data consistently achieves performance comparable to baseline methods trained with substantially larger proportions across all datasets.

Analysis. Baseline performance deteriorates as training data decreases, with several methods approaching heuristic-level errors in the zero-shot setting. VLM4ST maintains stronger performance under severe data scarcity, suggesting more effective transfer of pretrained knowledge. With auxiliary training data held fixed, replacing the pretrained backbone with random initialization increases zero-shot MAE on BikeNYC$_{in}$ from 2.51 to 4.37 (Appendix A), demonstrating a substantial contribution from vision-language pretraining. Prompt Forge's learned prototypes and the structure-preserving dual-path transformation may further support transfer by providing reusable ST patterns and preserving complementary spatial and temporal information.

3.4 Model Analysis

3.4.1 *Ablation Study*

We retrain VLM4ST without: (1) DiX-Attention, using only input projection; (2) Prompt Forge, feeding a zero prompt token; (3) the sequence path; and (4) the visual path. To isolate the VLM's contribution from the ST-specific modules, (5) *w.o. VLM backbone* removes ALBEF together with LoRA and Prompt Forge (whose prompts feed the text encoder), letting horizon queries attend directly to the format-transformer tokens and keeping all other modules (Appendix A).

As shown in Table 3, removing DiX-Attention causes the most severe degradation on every dataset (62%–301% MAE increase), confirming the importance of disentangled ST encoding before VLM processing. Removing Prompt Forge increases MAE by 6%–81%, most on traffic speed (50%–81%), indicating its role in bridging pretrained VLM knowledge with domain-specific patterns.

Table 3: Ablation study of VLM4ST components across seven benchmarks.

Model	BikeNYC$_{in}$		BikeNYC$_{out}$		TaxiNYC$_{in}$		TaxiNYC$_{out}$		TrafficHZ		TrafficTJ		TrafficZZ	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w.o. DiX-Attention	2.138	6.627	4.596	10.57	14.13	40.635	15.52	49.444	0.617	1.143	0.856	1.301	0.642	0.967
w.o. Prompt Forge	1.331	3.159	2.184	4.551	6.092	14.863	5.388	13.090	0.392	0.804	0.677	1.239	0.594	1.036
w.o. Seq path	1.770	3.743	2.495	4.969	7.673	16.344	7.820	17.673	0.304	0.608	0.588	0.877	0.480	0.826
w.o. Visual path	1.386	2.904	2.167	4.658	5.342	14.251	4.588	12.040	0.320	0.641	0.517	0.834	0.499	0.886
w.o. VLM backbone	1.632	3.767	2.923	5.714	6.685	15.949	5.649	13.398	0.335	0.673	0.624	1.030	0.551	0.932
VLM4ST (ours)	**1.018**	**2.772**	**2.063**	**4.384**	**4.637**	**12.103**	**3.866**	**9.918**	**0.217**	**0.461**	**0.437**	**0.756**	**0.397**	**0.719**

Removing the sequence path hurts the mobility datasets more, whereas removing the visual path hurts TrafficHZ/ZZ more; neither single-path variant approaches the full model, consistent with the complementary information losses analyzed in Appendix E. Without the VLM backbone, MAE rises by 39%–60% and the variant trails the strongest baseline in Table 1 on every dataset, showing that the gains of VLM4ST cannot be attributed to its ST-specific modules alone.

3.4.2 *Hierarchical Pattern Weight Analysis*

To understand how the Prompt Forge module encodes and retrieves spatiotemporal patterns, we analyze the hierarchical prototype activation patterns across three representative datasets. Figure 5 displays nine heatmaps organized as a 3×3 grid, where columns correspond to pattern modalities and rows correspond to different datasets.

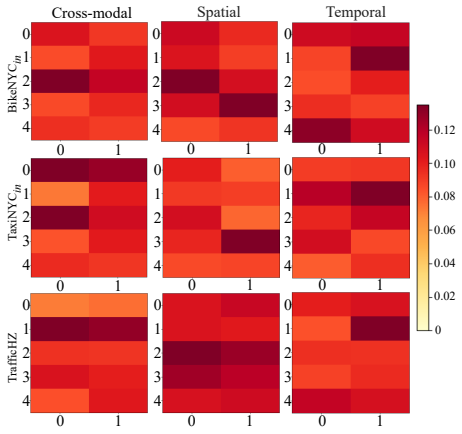


Figure 5: Coarse-fine prototype activation heatmaps for the Prompt Forge module. Each cell shows combined prototype weights where vertical axes denote coarse-grained categories (0-4), horizontal axes denote fine-grained sub-patterns (0-1).

The **temporal** pattern heatmaps show notable similarities across all three datasets, with consistent coarse-prototype activations irrespective of task. This matches the intuition that temporal dynamics (e.g., daily/weekly periodicity and trends) are broadly shared in urban sensing, and the learned temporal prototypes capture these regularities to support knowledge transfer across heterogeneous spatiotemporal datasets. In contrast, **spatial** activations are more domain- and location-dependent. The two New York City datasets (BikeNYC and TaxiNYC) exhibit similar spatial prototype profiles, reflecting shared urban geography and mobility structure, whereas TrafficHZ shows a distinct pattern, consistent with different road topology and traffic characteristics. This suggests the spatial memory learns location-aware representations while retaining transferable high-level spatial concepts. **Cross-modal** heatmaps exhibit the largest inter-dataset variability, with each dataset favoring different prototype combinations. This reflects task-dependent spatiotemporal coupling: the where–when interactions differ across bike usage, taxi demand, and traffic speed. Overall, these analyses indicate that VLM4ST learns structured, interpretable representations that separate shared temporal foundations from domain-specific spatial and cross-modal adaptations.

4 Conclusion

We present VLM4ST, a framework that aligns the architecture of dual-stream VLMs with the spatial and temporal structure of urban ST data. Disentangled encoding and dual-path format transformation preserve complementary spatial and temporal information, while Prompt Forge generates pattern-aware continuous prompts to support parameter-efficient adaptation. Experiments demonstrate consistent improvements over task-specific, LLM-based, and VLM-based predictors across short-term, long-term, few-shot, and zero-shot settings. Ablation studies support the contributions of both processing pathways and adaptive prompting, while backbone comparisons demonstrate the value of vision-language pretraining under data scarcity. VLM4ST provides a foundation for exploring structure-aware adaptation of pretrained multimodal models to broader numerical forecasting tasks.

AI USE STATEMENT

We used generative AI tools (LLM-based writing assistants) for language editing of the manuscript and for checking consistency between the numbers reported in the text and in the tables. Generative AI tools were not used to generate research ideas, design the method, implement the experiments, or produce experimental results. All AI-assisted text was reviewed and revised by the authors, who take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

Code is available at the anonymous repository linked in the abstract. Architectural details are given in Section 2 and Appendix D.1–D.3; dataset statistics and preprocessing in Appendix F; baseline configurations, including the reimplementation of VLM-based baselines, in Appendix G; and training settings in Section 3.1.

REFERENCES

Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation models defining a new era in vision: a survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

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

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, 16(3):1–20, 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.

Xinyu Chen, Chengyuan Zhang, Xiaoxu Chen, Nicolas Saunier, and Lijun Sun. Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression. *IEEE Transactions on Knowledge and Data Engineering*, 36(2):504–517, 2023.

Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, pp. 3074–3082, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Lan Feng, Mohammadhossein Bahari, Kaouther Messaoud Ben Amor, Éloi Zablocki, Matthieu Cord, and Alexandre Alahi. Unitraj: A unified framework for scalable vehicle trajectory prediction. In *European Conference on Computer Vision*, pp. 106–123. Springer, 2024.

Jiho Jang, Chaerin Kong, Donghyeon Jeon, Seonhoon Kim, and Nojun Kwak. Unifying vision-language representation space with single-tower transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 980–988, 2023.

Hyotaek Jeon, Hyunwook Lee, Juwon Kim, and Sungahn Ko. St-link: Spatially-aware large language models for spatio-temporal forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1057–1067, 2025.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Zhe Jiang. A survey on spatial prediction methods. *IEEE transactions on knowledge and Data Engineering*, 31(9):1645–1664, 2018.

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.

Yoonhwa Jung, Jaehoon Jung, Byungil Kim, and SangUk Han. Long short-term memory recurrent neural network for modeling temporal patterns in long-term power forecasting for solar pv facilities: Case study of south korea. *Journal of Cleaner Production*, 250:119476, 2020.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021a.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Lecheng Li, Xinzhuo Chai, Fei Dai, Shuai Wang, Yueyu Dong, and Bi Huang. Stformer: Spatio-temporal transformer network for traffic flow prediction. In *2024 IEEE Cyber Science and Technology Congress (CyberSciTech)*, pp. 250–257. IEEE, 2024a.

Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 4189–4196, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 4582–4597, 2021.

Yangfan Li, Kenli Li, Cen Chen, Xu Zhou, Zeng Zeng, and Keqin Li. Modeling temporal patterns with dilated convolutions for time-series forecasting. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(1):1–22, 2021b.

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.

Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5351–5362, 2024b.

Yuxuan Liang, Haomin Wen, Yutong Xia, Ming Jin, Bin Yang, Flora Salim, Qingsong Wen, Shirui Pan, and Gao Cong. Foundation models for spatio-temporal data science: A tutorial and survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6063–6073, 2025.

Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. Spatial-temporal large language model for traffic prediction. In *2024 25th IEEE International Conference on Mobile Data Management (MDM)*, pp. 31–40. IEEE, 2024.

Chenxi Liu, Kethmi Hirushini Hettige, Qianxiong Xu, Cheng Long, Shili Xiang, Gao Cong, Ziyue Li, and Rui Zhao. St-llm+: Graph enhanced spatio-temporal large language models for traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1567–1577, 2022.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8):1421–1441, 2019.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15638–15650, 2022.

Feiyan Sun, Wenning Hao, Ao Zou, and Qianyan Shen. A survey on spatio-temporal series prediction with deep learning: taxonomy, applications, and future directions. *Neural Computing and Applications*, 36(17):9919–9943, 2024.

Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18770–18782, 2023.

Michael Tschannen, Basil Mustafa, and Neil Houlsby. Clippo: Image-and-language understanding from pixels only. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11006–11017, 2023.

Hao Wang, Jindong Han, Wei Fan, Leilei Sun, and Hao Liu. Repst: Language model empowered spatio-temporal forecasting via semantic-oriented reprogramming. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3362–3370, 2025.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

Lijing Wang, Aniruddha Adiga, Jiangzhuo Chen, Adam Sadilek, Srinivasan Venkatramanan, and Madhav Marathe. Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 12191–12199, 2022.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017.

Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9154–9162, 2019.

Z Wu, S Pan, G Long, J Jiang, and C Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *The 28th International Joint Conference on Artificial Intelligence (IJCAI)*. International Joint Conferences on Artificial Intelligence Organization, 2019.

Yiming Xu, Xilei Zhao, Xiaojian Zhang, and Mudit Paliwal. Real-time forecasting of dockless scooter-sharing demand: A spatio-temporal multi-graph transformer approach. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):8507–8518, 2023.

Ning Yang, Hengyu Zhong, Haijun Zhang, and Randall Berry. Vision-llms for spatiotemporal traffic forecasting. *arXiv preprint arXiv:2510.11282*, 2025.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3634–3640. International Joint Conferences on Artificial Intelligence Organization, 2018.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. Unist: A prompt-empowered universal model for urban spatio-temporal prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4095–4106, 2024.

Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. Dnn-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 1–4, 2016.

Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Weijia Zhang, Jindong Han, Zhao Xu, Hang Ni, Hao Liu, and Hui Xiong. Urban foundation models: A survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6633–6643, 2024.

Xu Zhang, Yongshun Gong, Xinxin Zhang, Xiaoming Wu, Chengqi Zhang, and Xiangjun Dong. Mask-and contrast-enhanced spatio-temporal learning for urban flow prediction. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 3298–3307, 2023a.

Zijian Zhang, Xiangyu Zhao, Qidong Liu, Chunxu Zhang, Qian Ma, Wanyu Wang, Hongwei Zhao, Yiqi Wang, and Zitao Liu. Promptst: Prompt-enhanced spatio-temporal multi-attribute prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3195–3205, 2023b.

Liang Zhao, Min Gao, and Zongwei Wang. St-gsp: Spatial-temporal global semantic representation learning for urban flow prediction. In *Proceedings of the Fifteenth ACM international conference on web search and data mining*, pp. 1443–1451, 2022.

Tong Zhao, Junping Du, Zhe Xue, Meiyu Liang, Aijing Li, Xiaolong Meng, and Dandan Liu. St-vlm: A spatial-to-image multimodal spatial-temporal prediction framework with vision-language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16441–16449, 2026.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1234–1241, 2020.

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.

Table 4: Comparison of VLM backbones on BikeNYC$_{in}$, evaluating ALBEF, CLIP, and ALBEF without pretrained weights.

Backbone Model	Backbone Params	MAE	RMSE
ALBEF (ViT-B/16)	209M	**1.018**	**2.772**
CLIP (ViT-B/16)	149M	1.258	3.047
CLIP (ViT-L/14)	428M	1.194	3.171
ALBEF (ViT-B/16 w/o pretrain)	209M	1.451	3.596

A Impact of the VLM Backbone

To investigate the contribution of the VLM backbone, we compare ALBEF with CLIP variants and further evaluate the effect of pretraining by initializing ALBEF from scratch. As shown in Table 4, pretrained ALBEF achieves the best performance, outperforming CLIP-ViT-L by 14.7% in MAE and 12.6% in RMSE. This advantage suggests that performance is not primarily determined by backbone size: although a mild scaling trend is observed within CLIP, where ViT-L improves over ViT-B in MAE, the larger CLIP-ViT-L still underperforms ALBEF despite having substantially more parameters. This indicates that simply scaling the VLM does not yield proportional gains and is insufficient to outperform a better-matched backbone. Instead, the backbone architecture and pretraining objective are more critical. In particular, ALBEF enables deep cross-modal interaction through its multimodal encoder, whereas CLIP relies on separate visual and textual encoders whose interaction is mainly enforced at the contrastive alignment level. Moreover, removing pretrained weights from ALBEF leads to a substantial degradation, increasing MAE from 1.018 to 1.451 and RMSE from 2.772 to 3.596. These results confirm that vision-language pretraining provides transferable representations for spatiotemporal forecasting and plays a crucial role in the effectiveness of VLM4ST. Replacing ALBEF with BLIP (Li et al., 2022), another encoder-fusion VLM, yields a comparable MAE/RMSE of 1.041/2.83 on BikeNYC$_{in}$, indicating that the pipeline is not tied to a single checkpoint.

Table 5: Backbone ablation on additional datasets (MAE). Relative degradation w.r.t. pretrained ALBEF is shown in parentheses.

Backbone	BikeNYC$_{out}$	TaxiNYC$_{in}$	TrafficHZ
ALBEF (pretrained)	**2.063**	**4.637**	**0.217**
CLIP (ViT-B/16)	2.4102	5.4810	0.2512
ALBEF (w/o pretrain)	2.8603 (+38.6%)	6.2896 (+35.6%)	0.3021 (+39.2%)

Consistency across datasets. We extend the backbone comparison to three additional datasets covering mobility demand and traffic speed (Table 5). Consistent with the results on BikeNYC$_{in}$, removing ALBEF pretraining increases MAE by 35.6%–39.2%, while pretrained ALBEF outperforms CLIP on all three datasets. These results show that the benefits of pretraining and the observed backbone ranking extend beyond a single benchmark.

Removing the VLM backbone. To separate the contribution of the VLM from that of the ST-specific modules, the *w.o. VLM backbone* variant in Table 3 retains DiX-Attention with adaptive gated fusion (Appendix D.1), the Dual-Path Format Transformer, and the Horizon Query Predictor including its $\mathrm{Proj}(\mathrm{Pool}(\mathbf{X}_{\text{fused}}))$ residual (Appendix D.3), while removing the ALBEF backbone and its LoRA adapters. Prompt Forge is removed as well, since its prompt $\mathbf{B}_{\text{prompt}}$ is injected into the VLM text encoder and has no counterpart without the backbone; the horizon queries instead cross-attend directly to the output tokens of the Format Transformer. Like every variant in Table 3, it is retrained with the same initialization, hyperparameters, and random seeds as the full model rather than derived from the trained model at test time. This variant increases MAE by 38.8%–60.3% and RMSE by 29.6%–46.0% relative to VLM4ST and trails the strongest baseline in Table 1 on every dataset (e.g., 1.632 vs. 1.172 for ST-VLM on BikeNYC$_{in}$), so the ST-specific modules alone do not account for the improvements of VLM4ST. Conversely, retaining the VLM but removing DiX-Attention degrades performance even more, indicating that the gains arise from combining the two. This larger drop is expected, as DiX-Attention also supplies the CLS summaries that query Prompt Forge and the $\mathbf{X}_{\text{fused}}$ residual of the predictor, and without it the Format Transformer compresses point-wise embeddings before any spatial or cross-dimensional context is exchanged.

Backbone capacity vs. pretraining. Comparing this variant with the randomly initialized ALBEF (Tables 4 and 5), MAE decreases monotonically from no backbone to randomly initialized ALBEF to pretrained ALBEF on all four datasets with random-initialization results (BikeNYC$_{in}$, BikeNYC$_{out}$, TaxiNYC$_{in}$, and TrafficHZ). Loading pretrained weights accounts for 70.5%–92.7% of the total MAE reduction from the no-backbone variant to the full model, whereas the randomly initialized backbone, which has the same architecture and parameter count as the pretrained one, recovers only 7.3%–29.5%. The contribution of the VLM therefore stems mainly from pretrained representations rather than from added model capacity.

Interaction with Prompt Forge. On the three traffic speed datasets, *w.o. VLM backbone* attains lower MAE and RMSE than *w.o. Prompt Forge* (e.g., MAE 0.335 vs. 0.392 on TrafficHZ), whereas the opposite holds on the four mobility datasets (Table 3). On traffic speed, the pretrained backbone thus underperforms having no backbone when pattern-aware prompts are absent, yet yields the best results once Prompt Forge is added, suggesting that Prompt Forge is important for bridging the gap between natural image–text pretraining and ST fields. Although the prompt is a single token, every sequence token attends to it in each layer of the frozen text encoder, so it acts as the main sample-adaptive signal for steering the backbone, similar to continuous prompts for frozen transformers (Li & Liang, 2021; Lester et al., 2021). Traffic speed is also where HA, which averages same-period observations over the full training history rather than a six-step window, is strongest: its MAE is only 59%–118% above that of VLM4ST, versus 236%–307% on mobility data, and on TrafficHZ it outperforms eight of the fifteen learning-based baselines in Table 1. Variants whose degradation exceeds this narrow margin, such as *w.o. Prompt Forge* on TrafficHZ, therefore fall behind HA.

Pretrained representations vs. weight statistics. To examine whether the pretraining gain can be explained by weight distribution and scale alone, we randomly permute ALBEF's pretrained weights within each layer, preserving the layer-wise weight values while disrupting their learned arrangement. On BikeNYC$_{in}$, the shuffled backbone achieves MAE/RMSE of 1.437/3.550, close to random initialization (1.451/3.596) and substantially worse than pretrained ALBEF (1.018/2.772). Preserving these weight statistics therefore does not retain the pretraining advantage, supporting the importance of learned representations beyond weight scale alone, this is consistent with the attention rollout analysis in Appendix B.

Contribution of pretraining to zero-shot transfer. To assess the contribution of VLM pretraining beyond auxiliary-data training, we compare pretrained and randomly initialized ALBEF backbones under the same zero-shot training protocol. Both variants use identical auxiliary datasets without target-domain training examples. On BikeNYC$_{in}$, replacing pretrained weights with random initialization increases zero-shot MAE from 2.51 to 4.37, a 74.1% increase. This comparison demonstrates that vision-language pretraining makes a substantial contribution to cross-dataset transfer under the evaluated protocol.

B Visualization Analysis: ViT Responses to Spatial Structures in ST Pseudo-Images

To analyze the mechanism of the visual branch on ST pseudo-images, we visualize the temporal compression weights and the ViT attention rollout, as shown in Figure 6 and Figure 7. These two visualizations illustrate how the input temporal sequence is compressed into a pseudo-RGB image and how the ViT in the VLM attends to the generated pseudo-image.

As shown in Figure 6, the temporal-to-image transformation module does not simply average the historical frames. Instead, it generates distinct R-like, G-like, and B-like channel weights according to different time steps and spatial locations. The average weights of different time steps vary clearly across the three channels, indicating that the pseudo-RGB image encodes temporally selective and spatially adaptive information. In Figure 7, the pseudo-RGB image on the left exhibits a clear urban spatial structure, where high-intensity regions are concentrated in a high-activity area resembling Manhattan. The ViT attention rollout in the middle shows that the high-response regions of the ViT are mainly located around the high-intensity area in the lower-left part of the pseudo-image and extend along the diagonal spatial structure. The overlay on the right further demonstrates that the attention hotspots are highly aligned with the major high-intensity regions in the pseudo-RGB image.

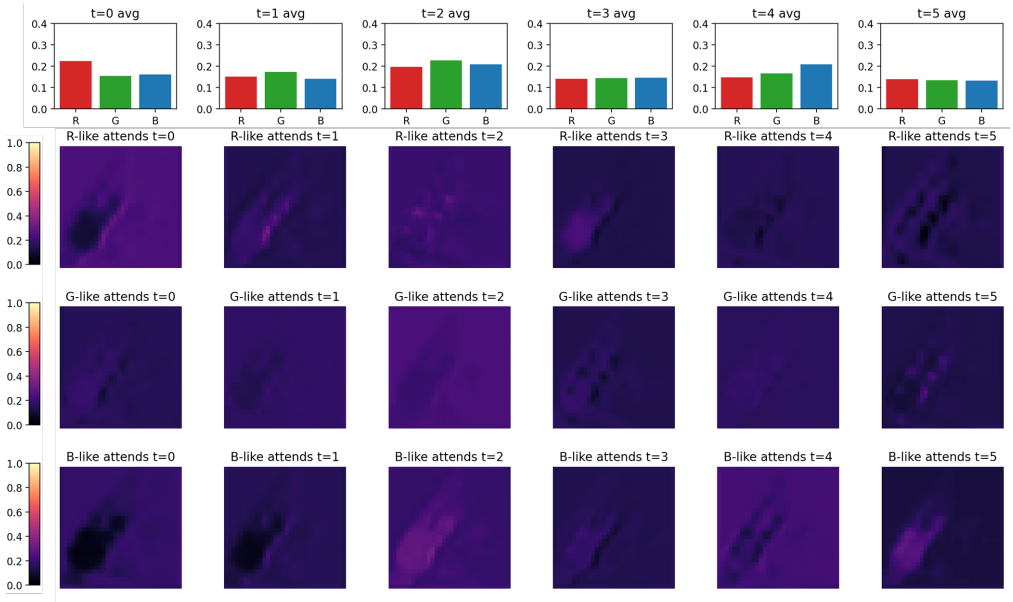


Figure 6: Visualization of temporal-to-image projection on TaxiNYC$_{in}$

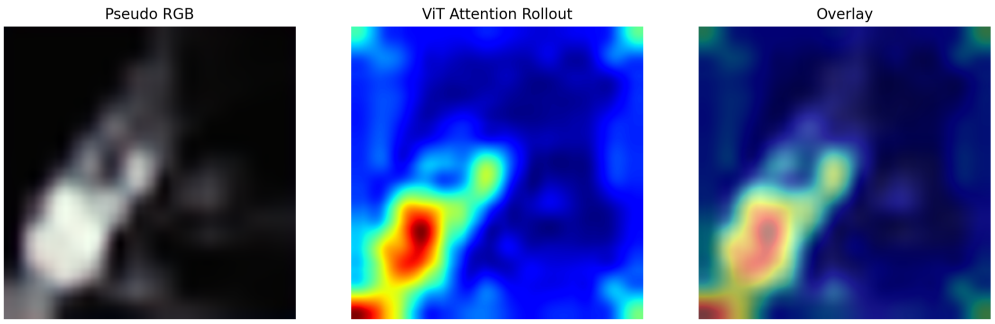
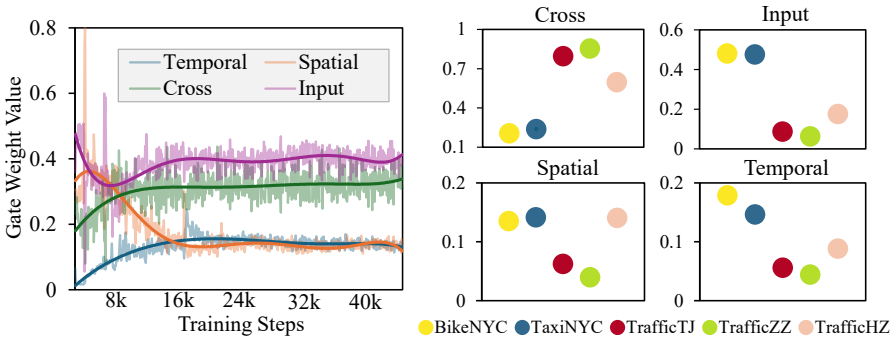


Figure 7: ViT attention rollout on TaxiNYC$_{in}$

These results indicate that the visual encoder can identify and attend to spatially salient structures in ST pseudo-images. They further suggest that, when processing pseudo-images generated from ST data, the pretrained ViT can leverage the locality, regional continuity, and salient-structure modeling capabilities learned from natural images, thereby providing effective visual representations for spatio-temporal prediction.

C Adaptive Gating Weight Analysis

To investigate how VLM4ST dynamically balances contributions from different feature pathways, we visualize the behavior of the Adaptive Gated Fusion module, which computes data-dependent weights for temporal, spatial, cross-modal, and input feature branches. Figure 8a illustrates the training dynamics of these gate weights across optimization steps. We observe that all four weight trajectories converge within 16k training steps and subsequently maintain stable values, the converged values reveal an interpretable hierarchy: the input branch receives the highest weight (≈ 0.4), followed by the cross-modal branch (≈ 0.3), while the temporal and spatial branches each contribute roughly 0.15. This distribution suggests that the model prioritizes preserving the original input information as a foundation, while the cross-modal features—which capture bidirectional spatiotemporal interactions—serve as the primary source of learned representations. The relatively lower but non-negligible weights assigned to the temporal and spatial branches indicate that these disentangled representations provide complementary refinements that enhance the fused features.



(a) Evolution of gating weights over training steps

(b) Dataset-specific gate weight distributions at convergence

Figure 8: Analysis of Adaptive Gated Fusion weights.

Figure 8b presents a dataset-wise comparison of the learned gate weights across diverse datasets. The visualization reveals that the fusion weights are not fixed but adapt to the inherent characteristics of each dataset, demonstrating that the Adaptive Gated Fusion mechanism learns to calibrate feature importance based on task-specific requirements, providing an interpretable window into how the model allocates representational capacity across different sensing scenarios.

D Additional Method Details

D.1 Feature Fusion with Adaptive Gating

The relative importance of temporal, spatial, and cross-modal features varies across different spatiotemporal scenarios. To accommodate this heterogeneity, we employ an adaptive gated fusion mechanism that dynamically weighs the contributions of different feature branches.

Given the four feature representations ($\mathbf{X}_{\text{temp}}$, $\mathbf{X}_{\text{spat}}$, $\mathbf{X}_{\text{cross}}$), and the original embedding $\mathbf{X}_{\text{input}}$, the gating network computes position-wise and time-step-wise weights through a multi-layer perceptron followed by softmax normalization:

$$\boldsymbol{\alpha} = \text{Softmax}(\text{MLP}([\mathbf{X}_{\text{temp}}; \mathbf{X}_{\text{spat}}; \mathbf{X}_{\text{cross}}; \mathbf{X}_{\text{input}}]))$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{T\times H\times W\times 4}$ contains the gate weights for each position and time step. The fused representation is then computed as the weighted sum:

$$\mathbf{X}_{\text{fused}} = \sum_{i=1}^{4} g_i \odot \mathbf{X}_i^{proj}$$

where $\odot$ denotes element-wise multiplication and $\mathbf{X}_i^{proj}$ represents the projected i-th branch. This formulation allows the network to emphasize different feature dimensions at different spatiotemporal locations: a sudden traffic incident might trigger high weights for cross-dimensional features at the affected location while maintaining temporal feature dominance elsewhere. The gating weights also provide interpretability, revealing which feature dimensions the model deems most informative for each prediction. The fused features $\mathbf{X}_{\text{fused}} \in \mathbb{R}^{B\times T\times H\times W\times d}$ proceed to the format transformation stage.

D.2 Sequence Path

We first apply adaptive spatial pooling to reduce the spatial dimensions from $H \times W$ to a smaller grid, yielding $\mathbf{X}_{pooled} \in \mathbb{R}^{B\times T\times h\times w\times d}$ where $h, w \ll H, W$. This pooled tensor is flattened to a sequence of length $T \cdot h \cdot w$, with each element representing a spatiotemporal patch. Learnable spatiotemporal positional embeddings are added to preserve location information.

The resulting sequence, potentially containing hundreds of tokens, may exceed practical length limits for the text encoder. We address this through a Perceiver-based compression mechanism. The

Perceiver uses a fixed number of learnable latent queries $\mathbf{Q} \in \mathbb{R}^{L \times d}$ (where L is the target sequence length) that attend to the long input sequence through cross-attention, effectively distilling the essential information into a compact representation. Multiple Perceiver layers with alternating cross-attention and self-attention refine the latent representation. The compressed sequence is projected to the VLM's hidden dimension through a linear layer with layer normalization.

Finally, we prepend a learnable CLS token to the sequence and insert the pattern prompt (projected from $\mathbf{B}_{\text{prompt}}$) as a special token, yielding the final sequence representation $\mathbf{C} \in \mathbb{R}^{B \times (L+2) \times D_{\text{vlm}}}$ ready for the text encoder. This sequence preserves complete temporal information—all T time steps contribute to the Perceiver's cross-attention—while the spatial information is encoded in the learned spatiotemporal positions and the Perceiver's compressed representation.

D.3 Predictor

The Horizon Query Predictor decodes future states from VLM embeddings using learnable query tokens that represent future time steps. This design draws inspiration from query-based detection architectures DETR (Carion et al., 2020) where learnable queries attend to encoded features to produce structured outputs. Specifically, we initialize a set of T' learnable horizon query tokens $\{\mathbf{q}_t\}_{t=1}^{T'} \in \mathbb{R}^{D_{\text{vlm}}}$, where each token corresponds to a specific future time step. These queries interact with the fused VLM embeddings $\mathbf{E}_{\text{fused}}$ through cross-attention:

$$\mathbf{h}_t = \text{CrossAttn}(\mathbf{q}_t, \mathbf{E}_{\text{fused}}, \mathbf{E}_{\text{fused}}) + \mathbf{q}_t$$

where the residual connection stabilizes optimization and preserves horizon-specific identity. To retain fine-grained spatiotemporal cues that may be attenuated by deep VLM processing, we further incorporate a lightweight residual enhancement from the DiX-Attention output $\mathbf{X}_{\text{fused}}$. The shallow features are projected to the forecasting space and combined with the horizon-aware representations:

$$\mathbf{h}_t^{'} = \mathbf{h}_t + \text{Proj}(\text{Pool}(\mathbf{X}_{\text{fused}}))$$

The enhanced horizon features $\{\mathbf{h}_t^{'}\}_{t=1}^{T'}$ are then jointly decoded by a spatial decoder to produce the final multi-step predictions $\hat{\mathbf{Y}} \in \mathbb{R}^{T' \times H \times W}$.

E Theoretical Analysis of Dual-Path Representation

In this section, we first define the representation bottleneck inherent in VLM format conversion, then analyze how the visual and sequence pathways incur structurally different dimensional compressions. We establish that their information losses are complementary rather than redundant and explain why single-path approaches face fundamental limitations. Finally, we show how VLM cross-modal fusion integrates the complementary information, with theoretical predictions aligning with our ablation results.

The Representation Bottleneck. Spatiotemporal forecasting requires converting a tensor $\mathbf{X} \in \mathbb{R}^{T \times H \times W}$ into VLM-compatible formats. VLM visual encoders expect inputs of shape $(3, H', W')$, while text encoders expect sequences of length L. Since VLMs are pretrained with fixed specifications (e.g., ALBEF uses $H' = W' = 224$ and $L \leq 512$), any format transformation must compress the original tensor. A format transformation $\phi : \mathbb{R}^{T \times H \times W} \to \mathcal{Z}$ maps spatiotemporal data to a VLM-compatible space, and different strategies incur structurally different information losses.

Visual Path Compression. The visual pathway produces a pseudo-image $\mathbf{D} \in \mathbb{R}^{3 \times H' \times W'}$ via $\phi_{\text{visual}} : \mathbb{R}^{T \times H \times W} \to \mathbb{R}^{3 \times H' \times W'}$, preserving full spatial resolution but collapsing T time steps into 3 channels—a $T/3$ compression ratio. By the data processing inequality, no downstream processing can recover more than 3 independent temporal statistics per spatial location. Complex temporal patterns requiring more parameters (e.g., multi-frequency periodicity, weekday/weekend distinctions) cannot be faithfully preserved. Our attention-weighted aggregation and statistical features (mean, std, trend) partially mitigate but cannot eliminate this limitation.

Sequence Path Compression. The sequence pathway produces tokens $\mathbf{C} \in \mathbb{R}^{L \times D}$ via $\phi_{\text{seq}} : \mathbb{R}^{T \times H \times W} \to \mathbb{R}^{L \times D}$, first applying spatial pooling from $H \times W$ to 10×10, then Perceiver-style compression to $L = 300$ tokens. This preserves full temporal ordering but reduces spatial resolution

Table 6: Statistics of spatiotemporal benchmark datasets used in experiments.

Dataset	Type	City	Frequency	Spatial partition	Mean	Standard deviation
BikeNYC$_{in}$	Bike Inflow	New York City	1 hour	10*20	8.4	17.9
BikeNYC$_{out}$	Bike Outflow	New York City	1 hour	10*20	5.5	16.1
TaxiNYC$_{in}$	Taxi Inflow	New York City	30 min	20*10	44	103
TaxiNYC$_{out}$	Taxi Outflow	New York City	30 min	20*10	37	77
TrafficHZ	Traffic Speed	Hangzhou	5 min	28×24	3.81	4.38
TrafficTJ	Traffic Speed	Tianjin	5 min	24×30	6.32	5.05
TrafficZZ	Traffic Speed	Zhengzhou	5 min	26×26	6.67	4.32

by factor $(H \times W)/100$. Spatial pooling averages local neighborhoods, smoothing fine-grained structure; features smaller than the pooling kernel become unrecoverable.

Complementarity of Information Loss. The two pathways lose different information: the visual path preserves spatial structure while compressing temporal dynamics; the sequence path preserves temporal ordering while compressing spatial resolution. For spatiotemporal patterns depending on both dimensions (e.g., a traffic jam at a specific intersection occurring during rush hours), the joint representation $(\phi_{\text{visual}}(\mathbf{X}), \phi_{\text{seq}}(\mathbf{X}))$ recovers information that neither pathway preserves alone. This is consistent with our ablation results: removing the sequence path causes larger degradation on mobility datasets (BikeNYC, TaxiNYC) where temporal patterns are critical, while removing the visual path causes larger degradation on TrafficHZ and TrafficZZ, where fine-grained spatial structure matters more. Neither single-path variant approaches full model performance, confirming complementary rather than redundant information losses.

Limitations of Single-Path Approaches. Prior LLM-based approaches adopt sequence-only strategies, flattening spatiotemporal tensors into 1D sequences. This must either flatten the spatial grid (destroying locality and requiring $O(H \cdot W)$ context length) or apply aggressive spatial compression (losing fine-grained patterns). Context length constraints force the latter, explaining why LLM-based baselines underperform VLM4ST on spatially-important datasets.

Cross-Modal Fusion. The dual-path design's effectiveness depends on VLM cross-modal fusion. In ALBEF, the text encoder performs cross-attention with visual embeddings:

$$\text{CrossAttn}(\mathbf{E}_{\text{seq}}, \mathbf{E}_{\text{vis}}) = \text{softmax}\left(\frac{\mathbf{E}_{\text{seq}}\mathbf{W}_Q(\mathbf{E}_{\text{vis}}\mathbf{W}_K)^\top}{\sqrt{D}}\right)\mathbf{E}_{\text{vis}}\mathbf{W}_V \tag{1}$$

This allows each sequence position (carrying temporal information) to attend to all visual positions (carrying spatial information). Attention weights learn to correlate temporally-significant moments with spatially-significant locations, reconstructing spatiotemporal dependencies distributed across the two pathways. This mechanism provides a principled foundation for the dual-path design grounded in concrete operations rather than abstract information-theoretic quantities.

F Dataset Statistics

We adopt the standard data format from prior work, where spatio-temporal observations are organized as 4D tensors (N, T, H, W). All data values are normalized to [-1, 1] via min-max scaling. For multi-dataset training, spatial dimensions are unified through bilinear interpolation. Table 6 summarizes the key statistics of each dataset, including the application domain, city coverage, temporal frequency, spatial partition resolution, and distributional characteristics (mean and standard deviation).

Auxiliary datasets for few-shot and zero-shot transfer. For each target dataset, the auxiliary datasets exclude both the target itself and its counterpart collected from the same source, since paired inflow/outflow data share the same city, spatial regions, and time period and would otherwise leak target-domain information. For example, when BikeNYC$_{in}$ is the target, both BikeNYC$_{in}$ and BikeNYC$_{out}$ are excluded, and the remaining five datasets (TaxiNYC$_{in}$, TaxiNYC$_{out}$, TrafficHZ, TrafficTJ, and TrafficZZ) serve as auxiliary data.

G BASELINES

- **History Average (HA)**: Uses the average of historical observations from the same time period as the future forecasting.
- **ARIMA** (Box et al., 2015): A classical time-series forecasting method that models future values using autoregressive terms, differencing to ensure stationarity, and moving-average components.
- **PredRNN** (Wang et al., 2017): A neural network for spatiotemporal forecasting that introduces a spatiotemporal LSTM to jointly model spatial appearances and temporal dynamics through a unified memory that flows across both layers and time steps.
- **MIM** (Wang et al., 2019): A spatiotemporal forecasting model that captures non-stationary dynamics by explicitly modeling differential signals between consecutive states using cascaded memory modules, enabling effective learning of both non-stationary and approximately stationary components.
- **ST-ResNet** (Zhang et al., 2017): A deep residual network for citywide crowd flow prediction that jointly models spatial correlations and multiple temporal patterns, while dynamically weighting these components and incorporating external factors such as weather and calendar effects.
- **TAU** (Tan et al., 2023): A spatiotemporal forecasting framework that replaces recurrent temporal modeling with a fully parallel Temporal Attention Unit, decomposing temporal attention into intra-frame static and inter-frame dynamic components while enhancing learning with differential divergence regularization.
- **STGSP** (Zhao et al., 2022): An urban flow forecasting model that learns global spatiotemporal semantics by encoding temporal positional information, modeling multi-scale temporal dependencies with self-attention, and leveraging masked pretraining to better capture long-range temporal context.
- **MCSTL** (Zhang et al., 2023a): A spatio-temporal learning framework for urban flow forecasting that captures time-varying regional heterogeneity through mask-enhanced and contrastive pretraining, combined with graph-based and global–local attention mechanisms.
- **PromptST** (Zhang et al., 2023b): A spatiotemporal multi-attribute forecasting framework that learns shared spatiotemporal patterns via a unified transformer backbone and adapts to specific attributes through prompt tuning for efficient and transferable modeling.
- **STGCN** (Yu et al., 2018): A graph-based spatiotemporal forecasting model that uses fully convolutional structures to jointly capture spatial dependencies and temporal dynamics in traffic networks, enabling efficient and accurate traffic forecasting.
- **UniST** (Yuan et al., 2024): A universal spatio-temporal forecasting model that leverages large-scale pretraining on diverse urban data and knowledge-guided prompts to generalize across multiple scenarios with strong few-shot and zero-shot performance.
- **ST-LLM** (Liu et al., 2024): A traffic forecasting framework that adapts large language models by treating spatiotemporal observations as tokens with spatial–temporal embeddings and partially frozen attention, enabling effective modeling of global dependencies and strong few-shot and zero-shot performance.
- **ST-LLM+** (Liu et al., 2025): A graph-enhanced spatiotemporal large language model that integrates traffic network structure via partially frozen graph attention and LoRA-based adaptation to capture both global and localized dependencies for traffic forecasting.
- **UrbanGPT** (Li et al., 2024b): An instruction-tuned spatiotemporal large language model that integrates a spatiotemporal dependency encoder to achieve strong generalization and zero-shot performance across diverse urban forecasting tasks.
- **ST-LINK** (Jeon et al., 2025): A spatiotemporal traffic forecasting framework that augments large language models with spatially enhanced attention and memory retrieval mechanisms to jointly capture spatial correlations and long-term temporal dependencies.
- **ST-Vision-LLM** (Yang et al., 2025): A vision-LLM-based traffic forecasting framework that processes grid frames together with per-cell history and a fixed instruction template in a decoder-only VLM, and decodes predictions cell by cell.

Table 7: Computational efficiency comparison across traditional, LLM-based, and our method.

Method	Trainable Params	Total Params	Trainable Ratio	Inference Speed	Memory Cost
STResNet	2.51M	2.51M	100%	0.011min	0.127GB
UniST	6.70M	6.70M	100%	0.034min	0.280GB
STLLM	1.49M	77M	1.94%	0.105min	0.914GB
STLLM+	1.75M	83M	2.11%	0.117min	0.973GB
UrbanGPT	1.36M	6.7B	0.02%	3.791min	15.83GB
VLM4ST	4.212M	232.74M	1.81%	0.133min	2.015GB

- **ST-VLM** (Zhao et al., 2026): A VLM-based spatiotemporal prediction framework that renders inter-node DTW, time-lagged cross-correlation, and spectral correlation matrices as a three-channel image, encodes node histories with an LSTM alongside a fixed task prompt, fuses modalities through an attention bias in the first VLM layer with LoRA fine-tuning, and decodes with an LSTM predictor.

Reimplementation of VLM-based baselines. Both VLM-based baselines are reimplemented following their published designs and evaluated with the same datasets, splits, horizons, normalization, and metrics as all other methods. For ST-VLM, which was designed for sensor graphs, each grid cell is treated as a node, so the correlation image for an $H \times W$ grid has size $HW \times HW$, and the textual task template is adapted to the forecasting horizon and the number of regions.

H Model Efficiency

To assess the computational efficiency of VLM4ST, we report the number of trainable parameters, total parameters, trainable ratio, inference speed, and peak GPU memory consumption in Table 7. All measurements are obtained on the BikeNYC$_{in}$ dataset, with inference speed computed as the total time required to process the entire test set and memory cost recorded as the peak GPU memory usage during inference. Traditional spatiotemporal models (ST-ResNet and UniST) train all parameters from scratch, achieving fast inference and low memory footprint, but their limited representational capacity constrains forecasting accuracy. LLM-based methods (ST-LLM, ST-LLM+, and UrbanGPT) freeze the majority of backbone parameters to leverage pretrained knowledge, resulting in trainable ratios below 2.2%; VLM4ST likewise trains less than 2% of its parameters. However, UrbanGPT, which employs a 6.7B-parameter LLM, incurs substantial computational overhead with 3.791 min inference time and 15.83 GB memory consumption, limiting its practical applicability. In contrast, VLM4ST achieves a favorable trade-off between model capacity and efficiency: with only 4.212M trainable parameters, our method maintains competitive inference speed and moderate memory cost while delivering superior forecasting performance. This efficiency stems from the parameter-efficient adaptation strategy that freezes the pretrained VLM backbone and introduces lightweight learnable modules (Prompt Forge and DiX-Attention), enabling effective knowledge transfer without the computational burden of full fine-tuning or excessively large language models.

I Related Work

I.1 Spatiotemporal forecasting

Spatiotemporal forecasting aims to forecast future states of dynamic systems evolving across both spatial and temporal dimensions (Jiang, 2018; Sun et al., 2024). Recent advances in deep learning have driven significant progress in this domain. Existing approaches can be broadly categorized into RNN-based (Shih et al., 2019; Jung et al., 2020; Li et al., 2021b), CNN-based (Li et al., 2021b; Zhang et al., 2016; Wang et al., 2023), transformer-based (Zheng et al., 2020; Xu et al., 2023), and GNN-based (Wang et al., 2022; Li & Zhu, 2021; Dai et al., 2020) families, each excelling within specific scenarios. However, most of these methods are designed and optimized for particular tasks and datasets, relying heavily on task-specific inductive biases and supervision signals. Although several recent efforts have started to explore cross-domain learning and more universal representations (Zhang

et al., 2023b; Yuan et al., 2024), their performance still degrades substantially in cross-dataset transfer, especially under zero-shot and few-shot settings. Our work addresses this limitation by leveraging the broad generalization capabilities of large-scale pretrained models, enabling robust cross-dataset and cross-domain spatiotemporal forecasting.

I.2 Spatiotemporal Foundation Models

Foundation models for spatiotemporal data seek to learn universal representations that transfer across diverse forecasting tasks and datasets (Awais et al., 2025). Current approaches bifurcate into two paradigms: models trained from scratch on spatiotemporal corpora, and those adapted from pretrained large language models (Liang et al., 2025; Zhang et al., 2024). (1) The former category constructs specialized architectures tailored to spatiotemporal structures, training them on collections of traffic, climate, or sensor datasets (Shao et al., 2022; Feng et al., 2024; Yuan et al., 2024). However, the available spatiotemporal data remains orders of magnitude smaller than text or image corpora used in NLP and computer vision, constraining the generalization capacity of scratch-trained models. (2) The latter paradigm repurposes language models by converting spatiotemporal sequences into tokenized formats amenable to autoregressive modeling (Chang et al., 2025; Liu et al., 2025; Wang et al., 2025). While these approaches inherit the zero-shot reasoning and in-context learning capabilities of LLMs, they face inherent structural mismatches. Spatiotemporal data exhibit tightly intertwined spatial and temporal dependencies—a grid location's future state depends simultaneously on its temporal history and spatial neighbors—whereas language models excel at sequential structures with primarily unidirectional dependencies. This mismatch forces spatiotemporal patterns into one-dimensional sequences, potentially discarding rich geometric structures. Moreover, language tokenizers designed for discrete symbols struggle to encode continuous-valued grids efficiently. These limitations motivate our exploration of vision-language models, which naturally accommodate both spatial and temporal modalities without enforcing sequential constraints.

I.3 Vision and VLM-based Forecasting

A growing line of work transfers pretrained vision models to temporal data by rendering series as images. ViTST Li et al. (2023) plots irregularly sampled multivariate series as line-graph images and classifies them with a pretrained ViT, while VisionTS Chen et al. (2024) reshapes univariate series into 2D periodic images and reformulates forecasting as masked-patch reconstruction with a pretrained visual MAE, achieving strong zero-shot performance. Time-VLM Zhong et al. (2025) further combines retrieval-augmented temporal features, rendered images, and generated text within a frozen VLM for time-series forecasting. These methods address 1D (possibly multivariate) series without a native spatial layout. For spatiotemporal data, ST-VLM Zhao et al. (2026) targets sensor graphs, converting pairwise DTW, lagged cross-correlation, and spectral similarity among nodes into an $N \times N$ three-channel image that is fused with LSTM-encoded node histories and a fixed textual prompt via a first-layer attention bias in a single-stream VLM; ST-Vision-LLM Yang et al. (2025) feeds grid frames and instruction templates to a decoder-only VLM. In both cases the VLM serves as a generic multimodal encoder for hand-crafted images and text. VLM4ST instead aligns the VLM's architecture with ST structure: the visual pathway operates on the ST field itself, the sequential pathway on temporally ordered representations without natural-language input, and the two are coupled through multi-layer cross-attention under learned, per-sample prompts. Our experiments (Tables 1 and 2) show that this alignment yields consistent gains over both VLM-based predictors.

I.4 Vision Language Models

Vision-language models achieve cross-modal understanding by aligning visual and textual representations through large-scale pretraining on image-text pairs (Radford et al., 2021; Li et al., 2021a; 2022). Architectural paradigms include dual-tower models that separately encode images and text before late fusion (Radford et al., 2021; Jia et al., 2021), dual-stream models that introduce early cross-attention between modalities (Yu et al., 2022; Singh et al., 2022), and unified models that process both modalities within a single transformer backbone (Tschannen et al., 2023; Jang et al., 2023). VLMs demonstrate remarkable zero-shot transfer and few-shot adaptation across vision-and-language tasks, attributed to the semantic richness of their learned representations and implicit world knowledge embedded during pretraining. Unlike language models that impose sequential ordering,

VLMs support parallel processing of spatial layouts, avoiding the loss of geometric information, enabling heterogeneous inputs to be projected into a shared semantic space. Motivated by these advantages, our VLM4ST framework leverages pretrained vision-language architectures to unify spatial and temporal reasoning, achieving strong cross-domain generalization without sacrificing the structural richness of spatiotemporal data.

J Limitations and Broader Impacts

Although VLM4ST demonstrates strong performance across diverse urban spatiotemporal forecasting benchmarks, it has several limitations. First, its dual-path design is most effective for grid-structured spatiotemporal data with meaningful 2D layouts, and may require topology-aware extensions for irregular graphs or sparse sensing systems. Second, although LoRA reduces trainable parameters, the pretrained VLM backbone still introduces higher computational cost than lightweight task-specific models. Third, our experiments mainly consider standard benchmarks; robustness under sensor noise, missing values, calibration shifts, and severe distribution shifts remains to be further studied.

VLM4ST may benefit urban forecasting applications such as traffic management, mobility planning, and resource allocation, but deployment in real-world decision systems should be cautious. Inaccurate predictions or biased historical data may affect downstream services unevenly across regions or populations. Practical use should therefore incorporate privacy protection, fairness checks, uncertainty estimation, and human oversight.