UNDERSTANDING BEFORE FORECASTING: MULTI-CITY URBAN DYNAMICS PREDICTION WITH VISION-LANGUAGE MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-city urban dynamics prediction requires capturing fine-grained spatial-temporal patterns across heterogeneous cities while effectively transferring knowledge across diverse urban environments. Vision-language models (VLMs), with their large-scale pretraining, multimodal representation, and generalization capabilities, offer a potential opportunity for this task. However, VLMs lack domain-specific knowledge of spatial-temporal urban dynamics and an understanding of flexible prediction tasks that vary in target dynamics, spatial regions and scales, and prediction times. Moreover, heterogeneous city distributions can induce conflicting optimization objectives during joint learning, hindering effective cross-city knowledge sharing. In this paper, we propose **UrbanBind**, a unified visual-language framework for multi-city urban dynamics prediction with three tightly coupled components: (i) *QA-Guided Spatial-Temporal Knowledge Grounding*, which adapts a pretrained VLM to urban dynamics and flexible prediction-task semantics through structured question-answering supervision; (ii) *Multi-City Urban Dynamics Encoding*, which explicitly learns spatial-temporal representations across cities, complementing QA-guided grounding with spatial-temporal knowledge and task understanding; and (iii) *Nash-Balanced Multi-City Optimization*, which balances conflicting city-wise updates to shared parameters while preserving city-specific optimization. Extensive experiments on multiple real-world urban dynamics datasets demonstrate the effectiveness of UrbanBind for multi-city forecasting.

1 INTRODUCTION

Urban dynamics prediction aims to forecast the evolution of human activities across urban regions, such as taxi demand, bike usage, and population movement. Accurate prediction of these dynamics is fundamental to a wide range of urban applications, including transportation management, resource allocation, and urban planning Xie et al. (2020). To this end, extensive research Wang et al. (2018); Jin et al. (2022); Zhang et al. (2025); Hao et al. (2025) has developed urban dynamics prediction models that capture complex spatial dependencies among urban regions and temporal patterns in historical observations. While these models have achieved substantial progress both in individual cities and across multiple cities, they primarily learn urban dynamics from numerical spatial-temporal observations and often rely on task-specific representations and architectures.

Recent advances in vision-language models (VLMs) offer a new paradigm for urban dynamics prediction. By jointly modeling visual and textual information, VLMs provide powerful capabilities for extracting structured patterns from visual representations while leveraging rich semantic knowledge and reasoning capabilities acquired during large-scale pretraining. This creates an opportunity to represent urban dynamics as visual spatial-temporal images and leverage VLMs to understand and predict their evolution Zhong et al. (2025); Zhao et al. (2026). Such a paradigm may move beyond conventional sequence modeling Liu et al. (2024); Wang et al. (2024); Liu et al. (2025) by enabling prediction models to capture both the spatial-temporal structure of urban dynamics and the corresponding semantic context.

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.

More importantly, VLMs provide a potential opportunity for multi-city urban dynamics prediction. Cities differ substantially in spatial layouts, grid resolutions, mobility modalities, and flow distributions, while the amount and quality of available data can also vary considerably across cities. Such heterogeneity and data imbalance make it challenging for conventional forecasting models Zhang et al. (2024; 2025); Hao et al. (2025) to effectively transfer knowledge across cities, particularly to those with limited observations. By treating urban dynamics as images coupled with natural-language task descriptions, VLMs can potentially map heterogeneous urban dynamics into a shared representation space and identify transferable spatial-temporal patterns across cities. Their large-scale pretraining and multimodal representation capabilities further enable knowledge sharing between data-rich and data-scarce cities, providing a promising path toward more robust and generalizable urban dynamics prediction across diverse urban environments.

Despite this potential, effectively leveraging VLMs for multi-city urban dynamics prediction remains challenging: *(1) Lack of spatial-temporal knowledge of urban dynamics.* VLMs are primarily pretrained on general-purpose image-text data and thus lack domain-specific knowledge of spatial-temporal urban dynamics and an understanding of flexible multi-city prediction tasks, which may vary in target dynamics, spatial regions and scales, and prediction times. Existing studies have also demonstrated their limitations in interpreting domain-specific heatmaps and numerical spatial information Chen et al. (2025); Yang et al. (2025). *(2) Inter-city optimization conflicts.* Joint multi-city learning can introduce optimization conflicts, as heterogeneous city distributions may produce competing gradients and interfere with each other during training Zhang et al. (2025); Hao et al. (2025).

In this paper, we propose **UrbanBind**, a unified visual-language framework for multi-city urban dynamics prediction that addresses both challenges. The primary contributions of this paper are summarized as follows:

- We formulate multi-city urban dynamics prediction from a visual-language perspective, representing urban dynamics as images and prediction tasks as language descriptions, thereby enabling flexible prediction across diverse cities and tasks.
- We address the two challenges through three tightly coupled components: (i) *QA-Guided Spatial-Temporal Knowledge Grounding*, which adapts the VLM to urban dynamics and enhances its understanding of spatial-temporal patterns through pretraining on a set of carefully designed question-answering (QA) tasks; and (ii) *Multi-City Urban Dynamics Encoding*, which explicitly learns spatial-temporal representations across cities through a dedicated urban dynamics encoder, complementing QA-guided grounding to enhance spatial-temporal knowledge and prediction task understanding. Together, (i) and (ii) address Challenge (1). (iii) *Nash-Balanced Multi-City Optimization* mitigates inter-city optimization conflicts by balancing city-wise updates to shared parameters while preserving city-specific optimization, directly addressing Challenge (2).
- Extensive experiments on heterogeneous real-world urban dynamics datasets demonstrate the effectiveness of UrbanBind for multi-city prediction, while further analyses validate its spatial-temporal grounding capability and the contributions of its key components.

2 PRELIMINARIES

Urban Dynamics Data. Let $\mathcal{C}$ denote the set of cities. For each city $c \in \mathcal{C}$, the urban dynamics observed at time t are represented as $\mathbf{X}_c^t \in \mathbb{R}^{M \times H_c \times W_c}$, where M is the number of dynamic channels and $H_c \times W_c$ denotes the city-specific spatial grid. In this work, $M = 2$, corresponding to inflow and outflow. For each valid forecasting origin t_i, we construct a historical–future window by pairing the preceding L observations with the subsequent K observations. Specifically, the i-th historical input and forecasting target are

$$\mathbf{X}_c^i = [\mathbf{X}_c^{t_i-L+1}, \ldots, \mathbf{X}_c^{t_i}] \in \mathbb{R}^{L \times M \times H_c \times W_c}, \qquad \mathbf{Y}_c^i = [\mathbf{X}_c^{t_i+1}, \ldots, \mathbf{X}_c^{t_i+K}] \in \mathbb{R}^{K \times M \times H_c \times W_c}. \tag{1}$$

We refer to $(\mathbf{X}_c^i, \mathbf{Y}_c^i)$ as the i-th historical–future window and define the dataset of city c as

$$\mathcal{D}_c = \{(\mathbf{X}_c^i, \mathbf{Y}_c^i)\}_{i=1}^{N_c}, \tag{2}$$

where N_c is the number of valid windows constructed from city c.

Unified Multi-City Urban Dynamics Prediction. Given urban dynamics datasets from multiple cities, $\mathcal{D} = \{\mathcal{D}_c\}_{c \in \mathcal{C}}$, our goal is to learn a unified predictor F_Θ that captures transferable spatial-

temporal knowledge while preserving city-specific dynamics. For a joint multi-city historical input $\mathbb{X} = \{\mathbf{X}_c\}_{c\in\mathcal{C}}$, we render the corresponding urban dynamics maps as visual inputs

$$\mathcal{I} = \mathcal{R}(\mathbb{X}), \tag{3}$$

where $\mathcal{R}(\cdot)$ denotes the rendering operation that converts the historical numerical dynamics grids into temporally ordered heatmaps while preserving each city's native spatial structure Liu et al. (2023); Dai et al. (2023) (Appendix A.4.1). A task description ϕ_{pred} specifies the forecasting objective, such as the target dynamics and prediction horizon. UrbanBind jointly uses the original numerical dynamics $\mathbb{X}$, their visual representations $\mathcal{I}$, and the forecasting prompt ϕ_{pred} to predict

$$\{\hat{\mathbf{Y}}_c\}_{c\in\mathcal{C}} = F_\Theta\left(\mathbb{X}, \mathcal{I}, \phi_{\text{pred}}\right), \qquad \hat{\mathbf{Y}}_c \in \mathbb{R}^{K\times M\times H_c\times W_c}. \tag{4}$$

Each prediction retains the native spatial resolution of its corresponding city.

3 METHOD

3.1 OVERVIEW

In this section, we introduce **UrbanBind**, a unified visual-language framework for multi-city urban dynamics prediction. It consists of three components: (1) *QA-Guided Spatial-Temporal Knowledge Grounding*, which adapts a pretrained VLM to spatial-temporal urban dynamics data and flexible task semantics through structured QA supervision (Section 3.2); (2) *Multi-City Urban Dynamics Encoding*, which explicitly learns spatial-temporal representations across cities through a dedicated urban dynamics encoder, enabling cross-city information exchange. This complements QA-guided grounding to enhance spatial-temporal knowledge and prediction task understanding for VLM (Section 3.3); and (3) *Nash-Balanced Multi-City Optimization*, which balances conflicting city-wise gradients over shared parameters while preserving city-specific optimization (Section 3.4).

3.2 STAGE 1: QA-GUIDED SPATIAL-TEMPORAL KNOWLEDGE GROUNDING

Urban dynamics exhibit complex spatial and temporal patterns, including regional interactions, short-term fluctuations, periodic patterns, and long-term evolution Li et al. (2024a). Pretrained VLMs provide transferable visual-semantic knowledge from large-scale image–text pretraining Radford et al. (2021), with demonstrated potential in urban understanding and mobility-related reasoning Huang et al. (2024); Sima et al. (2024); Xu et al. (2025). However, generic VLMs are not explicitly grounded in the quantitative spatial-temporal patterns of urban dynamics or the semantics of flexible prediction tasks. We therefore introduce *QA-Guided Spatial-Temporal Knowledge Grounding* to adapt the VLM through structured question–answering (QA) supervision. Two types of QA tasks are designed. *Understanding-oriented QA tasks* cover value reading, numerical computation, spatial and temporal reasoning, pattern recognition, and cross-city comparison, aiming to enhance the VLM's understanding of spatial-temporal patterns in urban dynamics. *Prediction-oriented QA tasks* cover future trend prediction, dominant spatial unit prediction, numerical forecasting, and dense-map prediction, aiming to adapt the VLM to diverse and flexible urban dynamics prediction tasks. The complete set of 18 understanding-oriented and 8 prediction-oriented QA tasks is detailed in Appendix A.4.

Let q index a QA task from the predefined QA set. For the q-th QA task, we convert the corresponding historical urban dynamics $\mathbf{X}_c$ into visual input $\mathcal{I}_q$, and derive a natural-language task question ϕ_q and its target answer a_q. As shown in Figure 1 (Stage 1), the VLM is composed of VLM encoder, VLM backbone and decoder. We first extract a hidden-state readout tokens $\mathbf{s}_q$ (*i.e.*, intermediate output tokens from the VLM backbone), which summarizes the historical visual context and task question. In our QA tasks, some questions require the VLM to produce a specific value or vector as the expected answer, while others require it to generate a dynamics map for a given region (see Appendix A.4). To enable the VLM to effectively handle these different QA tasks, we attach two auxiliary prediction heads to the readout tokens $\mathbf{s}_q$:

$$\hat{\mathbf{y}}_q = h_{\text{val}}(\mathbf{s}_q), \qquad \hat{\mathbf{Y}}_q = h_{\text{map}}(\mathbf{s}_q). \tag{5}$$

The value head h_{val} predicts scalar or vector-valued numerical targets, while the map head h_{map} predicts dense future dynamics maps at the native resolution of city c. These auxiliary heads are

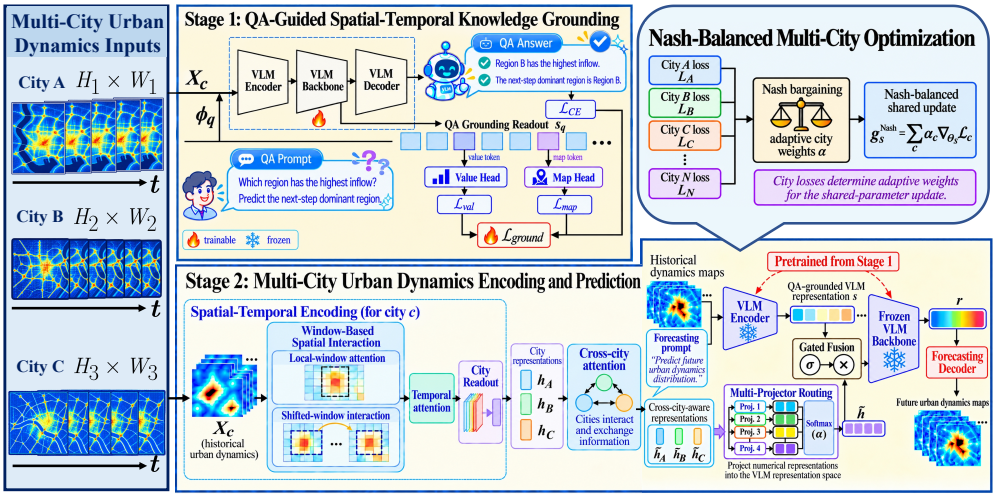


Figure 1: Overview of UrbanBind. Stage 1 performs QA-Guided Spatial-Temporal Knowledge Grounding to adapt a pretrained VLM to urban dynamics and forecasting tasks through structured QA supervision. Stage 2 integrates the grounded VLM with the multi-city spatial-temporal encoder for forecasting. The VLM encoder and backbone are frozen, while the remaining modules are jointly trained using Nash-Balanced Multi-City Optimization.

used only during QA-guided grounding to facilitate the VLM's understanding of diverse spatial-temporal patterns and prediction tasks.

For QA tasks requiring numerical supervision, let $\mathcal{Q}_v$ denote the set of value-supervised QA tasks. For each $q \in \mathcal{Q}_v$, let $\mathbf{y}_q \in \mathbb{R}^{d_q}$ denote its scalar- or vector-valued numerical target, where d_q is the target dimension determined by the corresponding QA task. The value loss is defined as

$$\mathcal{L}_{\text{val}} = \frac{1}{|\mathcal{Q}_v|} \sum_{q \in \mathcal{Q}_v} \frac{1}{d_q} \|\hat{\mathbf{y}}_q - \mathbf{y}_q\|_2^2 . \tag{6}$$

For QA tasks requiring dense-map supervision, let $\mathcal{Q}_m$ denote the set of map-supervised QA tasks. For each $q \in \mathcal{Q}_m$, let $\mathcal{C}_q$ denote the set of cities with map targets, and let $\mathbf{Y}_{q,c}$ denote the normalized future dynamics map of city c. The map loss is defined using element-wise mean squared error (MSE):

$$\mathcal{L}_{\text{map}} = \frac{1}{|\mathcal{Q}_m|} \sum_{q \in \mathcal{Q}_m} \frac{1}{|\mathcal{C}_q|} \sum_{c \in \mathcal{C}_q} \text{MSE}\left(\hat{\mathbf{Y}}_{q,c}, \mathbf{Y}_{q,c}\right) . \tag{7}$$

In addition to the numerical and map supervision described above, each QA task is also supervised through its textual target answer. Let $\mathcal{A}_q$ denote the positions of the target answer tokens for task q. We denote p_ψ as the conditional token distribution produced by the VLM, where ψ represents its trainable grounding parameters. The autoregressive cross-entropy loss is defined as

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N_{\text{tok}}} \sum_{q \in \mathcal{Q}} \sum_{\text{pos} \in \mathcal{A}_q} \log p_\psi \left(a_{q,\text{pos}} \mid \mathcal{I}_q, \phi_q, a_{q,<\text{pos}}\right) , \tag{8}$$

where $a_{q,\text{pos}}$ denotes the target token at position pos, $a_{q,<\text{pos}}$ denotes the preceding target tokens, and $N_{\text{tok}} = \sum_{q \in \mathcal{Q}} |\mathcal{A}_q|$ is the total number of supervised answer tokens.

The overall QA grounding objective is

$$\mathcal{L}_{\text{ground}} = \mathcal{L}_{\text{CE}} + \lambda_v \mathcal{L}_{\text{val}} + \lambda_m \mathcal{L}_{\text{map}}. \tag{9}$$

During grounding, VLM encoder, backbone, decoder and auxiliary prediction heads are optimized. After grounding, the auxiliary heads are discarded, and the resulting QA-grounded VLM encoder is retained for downstream forecasting in Section 3.3.

3.3 Stage 2: Multi-City Urban Dynamics Encoding and Prediction

QA-guided grounding equips the VLM with general spatial-temporal knowledge of urban dynamics, such as comparing flow magnitudes, identifying high- and low-demand regions, recognizing increasing or decreasing trends, detecting periodic patterns, and reasoning about spatial and temporal variations (see Appendix A.4). However, complex region-wise spatial dependencies, fine-grained temporal patterns, and cross-city relationships remain difficult to fully capture through QA tasks alone. We therefore introduce a dedicated multi-city urban dynamics encoder to explicitly learn spatial-temporal representations across cities and facilitate cross-city information exchange. Recent multimodal and language-based forecasting methods have similarly highlighted the complementary role of explicit numerical modeling Zhong et al. (2025); Jin et al. (2024). The resulting representations complement QA-guided grounding by providing fine-grained and cross-city spatial-temporal information, further enhancing the VLM's understanding of urban dynamics and flexible prediction tasks.

3.3.1 Multi-City Urban Dynamics Encoder

Urban dynamics exhibit complex spatial-temporal dependencies, where regional activities interact across space and evolve continuously over time Zhang et al. (2017); Yuan et al. (2026b). As shown in Figure 1 (Stage 2), given a specific historical urban dynamics window $\mathbf{X}_c^i = [\mathbf{X}_c^{t_i-L+1}, \ldots, \mathbf{X}_c^{t_i}] \in \mathbb{R}^{L\times M\times H_c\times W_c}$, we first model spatial patterns within each dynamics map $\mathbf{X}_c^\tau$, where $\tau \in \{t_i - L + 1, \ldots, t_i\}$, and then capture their temporal evolution across the L maps within the i-th window. Each city first goes through an independent dynamics encoder to preserve its native spatial structure and city-specific dynamics.

Spatial-Temporal Encoding. To first capture the spatial patterns of maps in the i-th window $\mathbf{X}_c^i$, at each time step τ, every grid cell in the map $\mathbf{X}_c^\tau$ is retained as one spatial token. Let P_c denote a city-specific 1×1 projector that maps the M dynamics channels to a d-dimensional feature space, a map $\mathbf{X}_c^\tau$ can be represented with a general spatial representation: $\mathbf{E}_c^\tau = P_c(\mathbf{X}_c^\tau) \in \mathbb{R}^{H_c\times W_c\times d}$. We further model more detailed spatial patterns and interactions in two successive steps based on the general $\mathbf{E}_c^\tau$. First, to capture the local spatial patterns within a window, we apply the city-specific self-attention $\mathrm{Attn}_{\mathrm{win}}^c$ within each non-overlapping window, yielding the local spatial representation $\mathbf{A}_c^\tau = \mathrm{Attn}_{\mathrm{win}}^c(\mathbf{E}_c^\tau) \in \mathbb{R}^{H_c\times W_c\times d}$. Second, we shift all the maps within the i-th window by δ grid cells spatially to get a new $(i+1)$-th window, where δ denotes the shift size, so that the maps in i-th window are overlapped with the ones in the new window. We then apply self-attention to the shifted new window: $\mathbf{U}_c^\tau = \mathrm{Attn}_{\mathrm{shift}}^c(\mathbf{A}_c^\tau) \in \mathbb{R}^{H_c\times W_c\times d}$. $\mathbf{U}_c^\tau$ captures both local and cross-window spatial interactions.

Next, to model temporal evolution within the i-th window, we apply self-attention $\mathrm{Attn}_{\mathrm{tmp}}^c$ to the representations $\mathbf{U}_c^\tau$ over the L historical steps within the window : $\mathbf{V}_c = \mathrm{Attn}_{\mathrm{tmp}}^c(\mathbf{U}_c^\tau) \in \mathbb{R}^{L\times H_c\times W_c\times d}$.

Cross-City Interaction. To enable cross-city interaction, we first aggregate the spatial-temporal representation $\mathbf{V}_c$ into a fixed-size city-level representation $\mathbf{h}_c = \mathrm{Attn}_{\mathrm{read}}^c(\mathbf{V}_c) \in \mathbb{R}^d$, where $\mathrm{Attn}_{\mathrm{read}}^c$ is the city-specific readout attention that aggregates information across spatial locations and historical steps. The resulting city representations provide a common interface for cross-city interaction. We stack them as $\mathbf{H} = [\mathbf{h}_c]_{c\in\mathcal{C}} \in \mathbb{R}^{|\mathcal{C}|\times d}$. Let $\mathrm{Attn}_{\mathrm{city}}$ denote self-attention applied along the city dimension. The cross-city-aware representations are obtained as $\widetilde{\mathbf{H}} = \mathrm{Attn}_{\mathrm{city}}(\mathbf{H}) \in \mathbb{R}^{|\mathcal{C}|\times d}$, where $\widetilde{\mathbf{H}} = \{\widetilde{\mathbf{h}}_c\}_{c\in\mathcal{C}}$ is the set of city-specific representations.

3.3.2 Combining the Encoder with VLM for Final Prediction

To perform multi-city urban dynamics prediction, we integrate the QA-grounded VLM encoder learned in Section 3.2, the spatial-temporal encoder introduced in Section 3.3.1, and the QA-grounded VLM backbone as follows.

First, with the QA-grounded VLM encoder $\mathcal{E}_{\mathrm{QA}}$ frozen, we feed the historical dynamics maps $\mathcal{I}$ and forecasting prompt ϕ_{pred} into $\mathcal{E}_{\mathrm{QA}}$ to obtain a task-aware urban dynamics representation: $\mathbf{s} = \mathcal{E}_{\mathrm{QA}}(\mathcal{I}, \phi_{\mathrm{pred}}) \in \mathbb{R}^{d_v}$. The representation $\mathbf{s}$ captures task-aware spatial-temporal knowledge relevant to the current multi-city forecasting task.

In parallel, the cross-city-aware representations $\{\widetilde{\mathbf{h}}_c\}_{c\in\mathcal{C}}$ learned from spatial-temporal encoder in Section 3.3.1 explicitly capture spatial-temporal and cross-city patterns, complimenting the representation $\mathbf{s}$ produced by $\mathcal{E}_{\mathrm{QA}}$. To ensure $\{\widetilde{\mathbf{h}}_c\}_{c\in\mathcal{C}}$ can be aligned with $\mathbf{s}$ in the VLM representation space successfully, we introduce a VLM-conditioned multi-projector mechanism: The cross-city representations $\{\widetilde{\mathbf{h}}_c\}_{c\in\mathcal{C}}$ are first concatenated into a joint multi-city representation. A linear projection then maps the concatenated representation from $\mathbb{R}^{|\mathcal{C}|d}$ to the VLM feature dimension d_v, yielding $\mathbf{h}_0 \in \mathbb{R}^{d_v}$. We subsequently apply a set of projectors $\{R_j\}_{j=1}^{J}$ to $\mathbf{h}_0$. Each projector produces a candidate representation $\mathbf{p}_j = \mathbf{h}_0 + R_j(\mathbf{h}_0)$, where $j = 1, \ldots, J$. Let f_π denote a lightweight routing network that takes the QA-grounded VLM representation $\mathbf{s}$ and $\mathbf{h}_0$ as inputs, and let $\boldsymbol{\pi} \in \mathbb{R}^J$ denote the resulting normalized projector weights, the routing and aggregation are defined as $\boldsymbol{\pi} = \mathrm{softmax}\left(f_\pi(\mathbf{s}, \mathbf{h}_0)\right), \widetilde{\mathbf{h}} = \sum_{j=1}^{J} \pi_j \mathbf{p}_j$.

We further integrate $\widetilde{\mathbf{h}}$ with the QA-grounded VLM representation $\mathbf{s}$ through feature-wise gated fusion. Let $\mathbf{g} \in (0,1)^{d_v}$ denote the feature-wise gate produced by a lightweight sigmoid gating network conditioned on $\mathbf{s}$ and $\widetilde{\mathbf{h}}$, and let $\odot$ denote element-wise multiplication. The fused representation is $\mathbf{z} = (1-\mathbf{g}) \odot \mathbf{s} + \mathbf{g} \odot \widetilde{\mathbf{h}}$. The gate adaptively balances the QA-grounded VLM representation and projected $\widetilde{\mathbf{h}}$ at each feature dimension. Next, $\mathbf{z}$ goes through the frozen VLM backbone producing $\mathbf{r} = \overline{\mathcal{T}}(\mathbf{z}) \in \mathbb{R}^{d_v}$.

For each city c, let F^c_{dec} denote the city-specific forecasting decoder. Given $\mathbf{r}$ produced by the VLM backbone, the final prediction is $\hat{\mathbf{Y}}_c = F^c_{\mathrm{dec}}(\mathbf{r})$, preserving the native spatial resolution of city c. The city-specific forecasting loss is defined as $\mathcal{L}_c = \mathrm{MSE}(\hat{\mathbf{Y}}_c, \mathbf{Y}_c) + \lambda_1 \mathcal{L}_1(\hat{\mathbf{Y}}_c, \mathbf{Y}_c)$, where λ_1 controls the contribution of the L1 term.

3.4 Nash-Balanced Multi-City Optimization

Joint multi-city training exposes the shared modules to city-specific objectives arising from heterogeneous data distributions and spatial-temporal patterns. These objectives may induce conflicting gradients over the shared parameters, making effective balancing important for stable multi-city optimization Yu et al. (2020); Navon et al. (2022). We therefore formulate the optimization of shared parameters as a Nash bargaining problem and determine city weights from the joint geometry of their gradients.

Let $\mathcal{L}_c$ denote the forecasting loss for city c, and let Θ_s denote the trainable parameters shared across cities. We denote the gradient of $\mathcal{L}_c$ with respect to Θ_s by $\mathbf{g}_c$, stack the city-wise gradients as columns of $\mathbf{G}$, and denote their Gram matrix by $\boldsymbol{\Gamma}$:

$$\mathbf{g}_c = \nabla_{\Theta_s} \mathcal{L}_c, \qquad \mathbf{G} = [\mathbf{g}_c]_{c\in\mathcal{C}}, \qquad \boldsymbol{\Gamma} = \mathbf{G}^\top \mathbf{G}. \tag{10}$$

The Gram matrix $\boldsymbol{\Gamma}$ captures the pairwise geometry of city-wise gradients in the shared parameter space. We seek a common gradient direction $\mathbf{d}$ to balance the city-specific objectives over the shared parameters, which are updated as $\Theta_s \leftarrow \Theta_s - \eta\mathbf{d}$, where $\eta > 0$ is the learning rate. We determine the common direction using a Nash bargaining objective Navon et al. (2022). Let $\mathcal{B}_\epsilon = \{\mathbf{d} : \|\mathbf{d}\|_2 \le \epsilon\}$ denote the feasible set with $\epsilon > 0$. The balanced direction is obtained by

$$\mathbf{d}^* = \arg\max_{\mathbf{d}\in\mathcal{B}_\epsilon} \sum_{c\in\mathcal{C}} \log\left(\mathbf{g}_c^\top \mathbf{d}\right), \qquad \mathbf{g}_c^\top \mathbf{d} > 0. \tag{11}$$

This objective balances the competing city-wise gradients while maintaining a positive improvement direction for each city. The resulting direction can be expressed as a weighted combination of the city-wise gradients, $\mathbf{d}^* \propto \mathbf{G}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{C}|}_+$ denotes the positive city-weight vector. These weights are determined by the gradient geometry encoded in $\boldsymbol{\Gamma}$. Let $\boldsymbol{\alpha}^{-1}$ denote the element-wise reciprocal of $\boldsymbol{\alpha}$. The city weights satisfy $\boldsymbol{\Gamma}\boldsymbol{\alpha} = \boldsymbol{\alpha}^{-1}$. The complete derivation is provided in Appendix A.5.

Since the Nash solution determines the common descent direction up to a positive scale, we use $\mathbf{G}\boldsymbol{\alpha}$ as its representative for optimizing the shared parameters. Let $\mathbf{g}_s^{\mathrm{Nash}}$ denote this Nash-balanced shared gradient, and let $\mathrm{sg}(\cdot)$ denote the stop-gradient operation, which treats the Nash weights as constants during parameter optimization. We define

$$\mathbf{g}_s^{\mathrm{Nash}} = \mathbf{G}\boldsymbol{\alpha} = \sum_{c\in\mathcal{C}} \mathrm{sg}(\alpha_c)\mathbf{g}_c. \tag{12}$$

Table 1: Unified multi-city urban dynamics forecasting results on three cities.

Method	NYCTAXI		BIKECHI		NYC-BIKE	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
UniST	0.180	2.031	0.679	2.832	2.684	5.863
UrbanGPT	0.176	1.930	0.725	3.352	3.386	8.064
UrbanDiT	0.234	2.027	0.952	3.376	3.448	6.804
UniFlow	0.432	2.799	1.253	4.441	3.529	7.115
RegionTrans	2.405	4.198	3.291	6.546	4.441	9.220
CrossTReS	2.246	3.762	2.609	4.483	4.212	7.077
STBaT	0.552	2.440	1.582	5.048	4.990	10.646
PLM-CUP	1.471	4.420	2.327	5.371	4.378	7.141
QA-Grounded VLM	0.401	3.233	1.243	4.859	4.132	8.973
UrbanBind	**0.169**	**1.892**	**0.649**	**2.725**	**2.540**	**5.538**

Thus, α_c controls the contribution of city c to the shared-parameter update, while city-specific parameters are optimized using only their corresponding loss $\mathcal{L}_c$ without Nash weighting. The complete training procedure and implementation details are provided in Appendix A.3.

4 Experiment

We conduct extensive experiments to evaluate the multi-city urban dynamics prediction capability and spatial-temporal knowledge grounding effectiveness of UrbanBind. Specifically, we seek to answer the following research questions: (1) Can UrbanBind effectively perform urban dynamics forecasting across different cities? (2) Does QA-Guided Spatial-Temporal Knowledge Grounding improve the VLM's understanding of urban dynamics and produce more prediction-relevant representations? (3) How do the individual components of UrbanBind contribute to its overall forecasting performance? (4) How do different hyperparameter settings affect the predictive accuracy and robustness of UrbanBind?

4.1 Datasets and Experimental Setup

We evaluate UrbanBind on four hourly urban dynamics datasets from the LibCity benchmark Wang et al. (2021): NYCTAXI, NYC-BIKE, BIKECHI, and BIKEDC, each containing inflow and outflow on its native spatial grid. NYCTAXI, NYC-BIKE, and BIKECHI are used for joint multi-city training, while BIKEDC is held out for evaluating cross-city generalization. We use eight historical observations to predict the subsequent four steps and instantiate the VLM with Qwen2.5-VL-3B-Instruct Bai et al. (2025). Forecasting performance is evaluated using mean absolute error (MAE) and root mean squared error (RMSE), averaged over the four prediction horizons and two dynamics channels. Detailed evaluation metrics, dataset statistics, and experimental settings are provided in Appendix A.2, Appendix A.1, and Appendix A.3, respectively.

4.2 Baselines

We compare UrbanBind with eight representative baselines from two categories: *general-purpose and foundation urban models*, including **UniST** Yuan et al. (2024), **UrbanGPT** Li et al. (2024b), **UrbanDiT** Yuan et al. (2026b), and **UniFlow** Yuan et al. (2026a); and *cross-city and language-augmented urban dynamics models*, including **RegionTrans** Wang et al. (2018), **CrossTReS** Jin et al. (2022), **STBaT** Hao et al. (2025), and **PLM-CUP** Zhou et al. (2026). Implementation details are provided in Appendix A.6.

Results of Question (1) (Multi-City Forecasting Performance). To evaluate whether UrbanBind can consistently achieve strong forecasting performance under unified multi-city modeling, we report the overall results on the three training cities in Tab. 1. We further visualize the horizon-wise RMSE in Fig. 2 to examine prediction performance as the forecasting horizon increases. Tab. 1 shows that UrbanBind achieves the best overall performance across NYCTAXI, BIKECHI, and

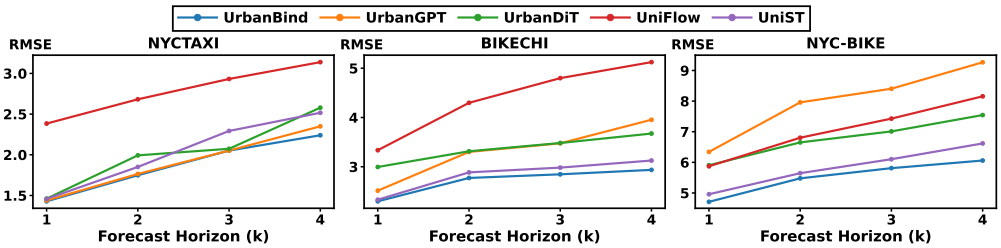


Figure 2: RMSE over forecast horizons(k) on NYCTAXI, BIKECHI, and NYC-BIKE.

Table 2: Analysis of spatial-temporal knowledge grounding and representation quality.

Evidence	Setting	CE ↓	PPL ↓	MAE ↓	RMSE ↓	R^2 ↑
QA Grounding	Base VLM	0.869	2.384	–	–	–
QA Grounding	QA-Grounded VLM	**0.126**	**1.134**	–	–	–
Visual Grounding	Correct Images	–	–	**2.238**	**6.491**	–
Visual Grounding	Shuffled Time	–	–	3.612	9.214	–
Visual Grounding	Permuted Cities	–	–	2.391	7.115	–
Visual Grounding	Blank Images	–	–	3.370	8.538	–
QA Representation Probe	Base VLM	–	–	0.158	0.252	0.112
QA Representation Probe	QA-Grounded VLM	–	–	**0.098**	**0.163**	**0.626**

Notes. **QA Grounding** compares the base and QA-grounded VLMs on the QA test set using assistant-token CE and PPL. For **Visual Grounding**, **Correct Images** uses the original P8 inputs; **Shuffled Time** permutes the eight historical observations within each city; **Permuted Cities** swaps visual inputs across cities; and **Blank Images** removes the visual dynamics content. For the **QA Representation Probe**, ridge regression is applied to frozen base and QA-grounded representations to predict the normalized future target.

NYC-BIKE, obtaining the lowest MAE and RMSE on all three datasets. Fig. 2 further shows that UrbanBind maintains consistently low forecasting errors across different horizons, with particularly clear advantages on BIKECHI and NYC-BIKE as the prediction horizon increases. These results demonstrate that UrbanBind can effectively model heterogeneous urban dynamics within a unified multi-city forecasting framework.

Results of Question (2) (Spatial-Temporal Knowledge Grounding). Tab. 2 evaluates the QA-grounded VLM from three complementary perspectives. Compared with the base VLM, QA grounding substantially reduces cross-entropy and perplexity on the QA test set, demonstrating improved understanding of urban dynamics. Performance also consistently degrades when the temporal order is shuffled, city inputs are permuted, or visual dynamics are removed, confirming that the grounded VLM effectively uses the historical visual and temporal context rather than relying solely on textual instructions. Finally, linear probing yields substantially lower forecasting errors and higher R^2, indicating that QA grounding produces more prediction-relevant representations. Together, these results show that QA-Guided Spatial-Temporal Knowledge Grounding improves urban dynamics understanding and representation quality. Further analysis of representation evolution and VLM–dynamics integration is provided in Appendix B.1.

Results of Question (3) (Ablation Studies). Tab. 3 shows that removing any major component consistently degrades forecasting performance across all three datasets. Removing QA-Guided Spatial-Temporal Knowledge Grounding confirms the benefit of adapting the VLM to urban dynamics and forecasting tasks, while ablating Multi-City Urban Dynamics Encoding—especially spatial-temporal encoding—causes the largest performance drop, highlighting the importance of explicit numerical dynamics modeling and cross-city interaction. Removing the VLM-conditioned multi-projector also worsens performance, demonstrating the benefit of projecting numerical dynamics before fusion with the QA-grounded VLM representation. Finally, removing Nash-Balanced Multi-City Optimization consistently increases forecasting error, confirming its contribution to balancing heterogeneous city-wise objectives during joint training.

Results of Question (4) (Hyperparameter Sensitivity Analysis). Fig. 3 evaluates the sensitivity of UrbanBind to the Stage 1 loss weights λ_v and λ_m, Stage 2 window shift size, and number of projectors J using final RMSE on NYCTAXI. Increasing λ_v generally improves performance, while varying λ_m or the window shift size produces only minor changes. Increasing J also gradually

Table 3: Ablation study of UrbanBind on three urban dynamics datasets.

Dataset	Metric	Full	w/o QG	UDE		w/o MP	w/o NBO
				w/o STE	w/o CCI		
NYCTAXI	MAE	**0.169**	0.180	0.198	0.194	0.184	0.183
	RMSE	**1.892**	1.921	2.270	2.111	2.054	1.955
BIKECHI	MAE	**0.649**	0.658	0.751	0.686	0.734	0.687
	RMSE	**2.725**	2.807	3.195	2.912	3.022	2.808
NYC-BIKE	MAE	**2.540**	2.684	3.071	2.939	2.883	2.867
	RMSE	**5.538**	5.581	6.910	6.231	6.250	6.150

Notes. **QG:** QA-Guided Spatial-Temporal Knowledge Grounding; **UDE:** Multi-City Urban Dynamics Encoding; **STE:** Spatial-Temporal Encoding; **CCI:** Cross-City Interaction; **MP:** VLM-Conditioned Multi-Projector; **NBO:** Nash-Balanced Multi-City Optimization.

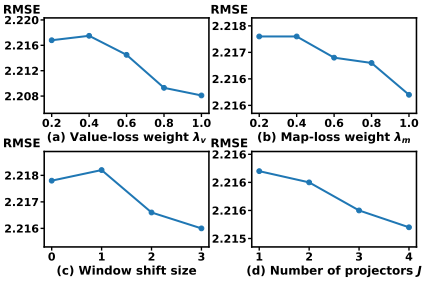


Figure 3: Hyperparameter sensitivity on NYCTAXI using final forecasting RMSE.

reduces RMSE, suggesting that additional projectors provide greater capacity for VLM–dynamics integration. Overall, UrbanBind remains stable across the tested hyperparameter ranges, indicating limited sensitivity to these configurations.

5 Related Work

Unified and Cross-City Urban Dynamics Prediction. Recent studies increasingly pursue unified spatial-temporal modeling across heterogeneous cities, data structures, and prediction tasks. UniST Yuan et al. (2024) learns transferable urban representations through large-scale pretraining and prompting, while OpenCity Li et al. (2025), UniFlow Yuan et al. (2026a), and UrbanDiT Yuan et al. (2026b) extend unified modeling across cities, spatial structures, and urban tasks. TransFM Jiang et al. (2026) and SQGEF Wang et al. (2025) further address cross-city heterogeneity and spatial-granularity differences. Cross-city transfer methods such as CrossTReS Jin et al. (2022) and STBaT Hao et al. (2025) improve knowledge transfer through selective source-region weighting and balanced transfer learning. Unlike these approaches, UrbanBind preserves city-specific native-grid dynamics while enabling explicit cross-city interaction and VLM–dynamics integration.

Language- and Vision-Augmented Spatial-Temporal Forecasting. Language and vision models have also been introduced to enhance numerical forecasting. UrbanGPT Li et al. (2024b) combines a spatial-temporal encoder with an instruction-tuned LLM, while FSTLLM Jiang et al. (2025) and PLM-CUP Zhou et al. (2026) leverage pretrained language-model knowledge for few-shot or cross-city prediction. Time-VLM Zhong et al. (2025) jointly models temporal, visual, and textual representations, and ST-Vision-LLM Yang et al. (2025) treats traffic matrices as structured visual inputs. UrbanBind differs by grounding task-aware spatial-temporal knowledge through structured QA supervision and explicitly integrating the resulting VLM representations with heterogeneous multi-city numerical dynamics.

6 Conclusion

In this work, we introduced **UrbanBind**, a unified visual-language framework for multi-city urban dynamics prediction. UrbanBind first employs QA-Guided Spatial-Temporal Knowledge Grounding to adapt a pretrained VLM to quantitative spatial-temporal patterns and flexible prediction tasks. It then combines the QA-grounded VLM with Multi-City Urban Dynamics Encoding, which preserves city-specific native grids while enabling cross-city information exchange and explicit numerical dynamics modeling. Nash-Balanced Multi-City Optimization further mitigates conflicting city-wise updates during joint training while preserving city-specific optimization. Extensive experiments on heterogeneous real-world urban dynamics datasets show that UrbanBind consistently achieves strong unified forecasting performance. Further grounding analyses demonstrate improved spatial-temporal knowledge and representation quality in the QA-grounded VLM, while ablation studies confirm the contributions of the grounding, multi-city dynamics encoding, VLM-conditioned projection, and Nash-balanced optimization. Overall, UrbanBind provides a unified framework for integrating task-aware VLM representations with explicit numerical dynamics modeling across heterogeneous cities.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, retrieval and discovery of relevant literature, and the refinement and debugging of research code. All AI-assisted edits, retrieved references, and code modifications were manually reviewed and verified by the authors. The authors take full responsibility for the final content and implementation of this work.

REPRODUCIBILITY STATEMENT

We provide detailed information for reproducing UrbanBind in Appendix A, including the dataset descriptions and public data sources, evaluation metrics, experimental settings, QA construction, optimization details, and baseline implementations. We also release an implementation of UrbanBind at `https://github.com/anonymousaabbcc/anony.git`.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. 2025. URL `https://arxiv.org/abs/2502.13923`.

Jian Chen, Peilin Zhou, Yining Hua, Dading Chong, Meng Cao, Yaowei Li, Wei Chen, Bing Zhu, Junwei Liang, and Zixuan Yuan. Climateiqa: A new dataset and benchmark to advance vision-language models in meteorology anomalies analysis. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5322–5333, 2025.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. volume 36, pp. 49250–49267, 2023.

Xinyan Hao, Huaiyu Wan, Shengnan Guo, and Youfang Lin. Balancing imbalance: Data-scarce urban flow prediction via spatio-temporal balanced transfer learning. In *IJCAI*, pp. 2865–2873, 2025.

Weiming Huang, Jing Wang, and Gao Cong. Zero-shot urban function inference with street view images through prompting a pretrained vision-language model. *International Journal of Geographical Information Science*, 38(7):1414–1442, 2024.

Yue Jiang, Yile Chen, Xiucheng Li, Qin Chao, Shuai Liu, and Gao Cong. Fstllm: Spatio-temporal llm for few shot time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

Zhihuan Jiang, Ailing Huang, Renhe Jiang, Geqi Qi, Qian Luo, and Yoshihide Sekimoto. Transfm: A foundation model for cross-city multimodal transportation demand prediction. *Transportation Research Part C: Emerging Technologies*, 186:105618, 2026.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James 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. In *International conference on learning representations*, volume 2024, pp. 23857–23880, 2024.

Yilun Jin, Kai Chen, and Qiang Yang. Selective cross-city transfer learning for traffic prediction via source city region re-weighting. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 731–741, 2022.

Xiaoyu Li, Yongshun Gong, Wei Liu, Yilong Yin, Yu Zheng, and Liqiang Nie. Dual-track spatio-temporal learning for urban flow prediction with adaptive normalization. *Artificial Intelligence*, 328:104065, 2024a.

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.

Zhonghang Li, Long Xia, Lei Shi, Yong Xu, Dawei Yin, and Chao Huang. Open spatio-temporal foundation models for traffic prediction. *ACM Transactions on Intelligent Systems and Technology*, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. volume 36, pp. 34892–34916, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 83982–84006, 2025.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. 2022.

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.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pp. 256–274. Springer, 2024.

Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. Libcity: An open library for traffic prediction. In *Proceedings of the 29th international conference on advances in geographic information systems*, pp. 145–148, 2021.

Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. Cross-city transfer learning for deep spatio-temporal prediction. 2018.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in neural information processing systems*, 37:469–498, 2024.

Zhigaoyuan Wang, Ying Sun, and Hengshu Zhu. Unifying knowledge from diverse datasets to enhance spatial-temporal modeling: A granularity-adaptive geographical embedding approach. In *ICML*, 2025.

Peng Xie, Tianrui Li, Jia Liu, Shengdong Du, Xin Yang, and Junbo Zhang. Urban flow prediction from spatiotemporal data using machine learning: A survey. *Information Fusion*, 59:1–12, 2020.

Zhenhua Xu, Yan Bai, Yujia Zhang, Zhuoling Li, Fei Xia, Kwan-Yee K Wong, Jianqiang Wang, and Hengshuang Zhao. Drivegpt4-v2: Harnessing large language model capabilities for enhanced closed-loop autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17261–17270. IEEE, 2025.

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

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. volume 33, pp. 5824–5836, 2020.

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.

Yuan Yuan, Jingtao Ding, Chonghua Han, Zhi Sheng, Depeng Jin, and Yong Li. Uniflow: a foundation model for unified urban spatio-temporal flow prediction. *IEEE Transactions on Mobile Computing*, 2026a.

Yuan Yuan, Chonghua Han, Jingtao Ding, Guozhen Zhang, Depeng Jin, and Yong Li. Diffusion transformers as open-world spatiotemporal foundation models. volume 38, pp. 30518–30545, 2026b.

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.

Yu Zhang, Hua Lu, Ning Liu, Yonghui Xu, Qingzhong Li, and Lizhen Cui. Personalized federated learning for cross-city traffic prediction. In *IJCAI*, pp. 5526–5534, 2024.

Yudong Zhang, Xu Wang, Xuan Yu, Zhaoyang Sun, Kai Wang, and Yang Wang. Drawing informative gradients from sources: A one-stage transfer learning framework for cross-city spatiotemporal forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1147–1155, 2025.

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.

Siru Zhong, Yuxuan Liang, Weilin Ruan, Ming Jin, Huan Li, and Qingsong Wen. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting. In *AI for Time Series*, pp. 173–203. CRC Press, 2025.

Qiang Zhou, Xudong Tong, Yuting Liu, Chuanxing Liu, and Jingjing Gu. Exploiting pre-trained language model for cross-city urban flow prediction guided by information-theoretic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16513–16521, 2026.

A APPENDIX FOR REPRODUCIBILITY

To support reproducibility, we release our code and provide detailed dataset and preprocessing specifications. [1]

A.1 DETAILED DESCRIPTION OF THE DATASET

We use four urban dynamics datasets from the LibCity benchmark Wang et al. (2021), collected from three cities: taxi flow in New York City (NYCTAXI), bike flow in New York City (NYCBike), bike flow in Chicago (BIKECHI), and bike flow in Washington, D.C. (BIKEDC). NYCTAXI, NYCBike, and BIKECHI are used in the main experiments, while BIKEDC is held out for evaluating cross-city generalization.

Data description. NYCTAXI spans January 1–March 31, 2014 (90 days), NYCBike spans April 1–September 30, 2014 (183 days), and BIKECHI/BIKEDC span July 1–September 30, 2020 (92 days). All datasets are organized as grid-based urban dynamics data at an hourly resolution, where inflow counts movements entering each grid cell and outflow counts movements leaving each grid cell. NYCTAXI uses a 10×20 spatial grid, NYCBike and BIKEDC use a 16×8 grid, and BIKECHI uses a 15×18 grid. Each observation contains two channels corresponding to inflow and outflow. Overall, NYCTAXI has shape $(2160, 2, 10, 20)$, NYCBike has shape $(4392, 2, 16, 8)$, BIKECHI has shape $(2208, 2, 15, 18)$, and BIKEDC has shape $(2208, 2, 16, 8)$.

[1] UrbanBind code: `https://github.com/anonymousaabbcc/anony.git`

A.2 Evaluation Metrics

We use different evaluation metrics according to the corresponding experimental objective. For the main forecasting, cross-city transfer, ablation, and hyperparameter experiments, we report mean absolute error (MAE) and root mean squared error (RMSE). For QA grounding, we additionally report cross-entropy (CE) and perplexity (PPL), while representation probing is evaluated using MAE, RMSE, and the coefficient of determination (R^2). Centered kernel alignment (CKA) is further used to measure structural correspondence between learned representations.

MAE and RMSE. Given N target values $\{y_i\}_{i=1}^N$ and predictions $\{\hat{y}_i\}_{i=1}^N$, MAE and RMSE are defined as

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|, \tag{13}$$

and

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}. \tag{14}$$

Lower values indicate better predictive accuracy. For urban dynamics forecasting, the reported scores are aggregated over the four prediction horizons and the two dynamics channels, inflow and outflow.

Cross-Entropy and Perplexity. For the QA grounding analysis, CE is computed only over the target answer tokens:

$$\mathrm{CE} = -\frac{1}{N_{\mathrm{tok}}}\sum_{q}\sum_{t\in\mathcal{A}_q} \log p_\psi\left(a_{q,t} \mid \mathcal{I}_q, \phi_q, a_{q,<t}\right), \tag{15}$$

where $\mathcal{A}_q$ denotes the target answer-token positions and N_{tok} is the total number of evaluated answer tokens. Perplexity is derived from CE as

$$\mathrm{PPL} = \exp(\mathrm{CE}). \tag{16}$$

Lower CE and PPL indicate higher likelihood assigned to the correct target answers.

Coefficient of Determination. For the representation probing experiments, R^2 measures how much of the variation in the prediction target can be explained by a linear probe:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{17}$$

where $\bar{y}$ denotes the mean target value. Higher R^2 indicates that the learned representation makes more prediction-relevant information accessible to the linear probe.

Centered Kernel Alignment. We use linear CKA to quantify structural correspondence between two representations. Given centered representation matrices $\mathbf{X} \in \mathbb{R}^{N\times d_x}$ and $\mathbf{Y} \in \mathbb{R}^{N\times d_y}$, linear CKA is defined as

$$\mathrm{CKA}(\mathbf{X}, \mathbf{Y}) = \frac{\left\|\mathbf{X}^\top\mathbf{Y}\right\|_F^2}{\left\|\mathbf{X}^\top\mathbf{X}\right\|_F \left\|\mathbf{Y}^\top\mathbf{Y}\right\|_F}. \tag{18}$$

Higher CKA indicates stronger structural correspondence between the two representation spaces. In our analyses, CKA is used both to compare learned representations with future urban dynamics targets and to examine correspondence between VLM and numerical dynamics representations.

A.3 Experimental Setup

We use eight historical observations to predict the subsequent four steps and train UrbanBind across NYCTAXI, BIKECHI, and NYC-BIKE while preserving their native grids. Qwen2.5-VL-3B-Instruct Bai et al. (2025) is used as the base VLM.

Training Procedure. UrbanBind is trained in two stages. In Stage 1, *QA-Guided Spatial-Temporal Knowledge Grounding*, we optimize $\mathcal{L}_{\mathrm{ground}}$ to adapt the VLM to spatial-temporal urban dynamics

and flexible prediction tasks. In Stage 2, *Multi-City Urban Dynamics Encoding and Prediction*, the QA-grounded VLM encoder $\mathcal{E}_{\mathrm{QA}}$ and VLM backbone $\overline{\mathcal{T}}$ are frozen, while all remaining modules are trainable. The shared parameters are optimized using the Nash-balanced gradient $\mathbf{g}_s^{\mathrm{Nash}}$, while city-specific parameters are optimized using their corresponding forecasting losses $\mathcal{L}_c$.

Stage 1: QA-Guided Spatial-Temporal Knowledge Grounding. Stage 1 is trained on the QA data constructed from the three training cities, with BIKEDC excluded. The vision tower and base language-model parameters are frozen, while the vision-language merger, language-side LoRA modules, and auxiliary value and map prediction heads are optimized. LoRA is applied to the language attention and MLP projections with rank $r = 16$, scaling factor $\alpha = 32$, and dropout 0.05. We use AdamW with learning rates of 1.0×10^{-4}, 2.0×10^{-5}, and 5.0×10^{-4} for the LoRA parameters, vision-language merger, and auxiliary heads, respectively, with weight decay 0.01 and a cosine schedule with 3% warmup. The value and map losses are weighted equally, i.e., $\lambda_v = \lambda_m = 1$. Training uses four GPUs with a micro-batch size of 1 and two-step gradient accumulation, yielding an effective batch size of 8, for up to 5 epochs with validation-based early stopping. We use BF16 mixed precision, gradient clipping with maximum norm 1.0, and random seed 42.

Stage 2: Multi-City Urban Dynamics Encoding and Prediction. The QA-grounded VLM encoder and backbone are frozen during Stage 2. We use AdamW with learning rates of 1.5×10^{-4}, 1.0×10^{-4}, and 7.5×10^{-4} for the dynamics encoders, projection and fusion modules, and forecasting heads, respectively, together with a cosine schedule and 2% warmup. Training uses four GPUs with a micro-batch size of 1 and three-step gradient accumulation, yielding an effective batch size of 12, for up to 50 epochs with validation-based early stopping.

A.4 Structured Urban-Dynamics QA Design

We construct a structured urban-dynamics QA corpus to ground the VLM in quantitative spatial-temporal patterns of urban dynamics and cross-city relationships before downstream forecasting. The corpus contains 18 understanding-oriented QA families and 8 prediction-oriented families. Rather than treating forecasting as a single end-to-end mapping, these tasks decompose urban dynamics reasoning into temporal, spatial, joint spatial-temporal, and cross-city capabilities.

A.4.1 Spatial Granularity and Visual Input

Each QA example is constructed from the historical numerical dynamics tensor $\mathbf{X}_c \in \mathbb{R}^{L\times M\times H_c\times W_c}$. For each city and dynamics channel, the native $H_c \times W_c$ grid at each historical step is rendered as a heatmap without spatial resizing. The visualization range is determined by the 1st and 99th percentiles computed from the corresponding city's training split, while the underlying numerical values remain unchanged. The $L = 8$ historical maps are arranged in chronological order to form the visual input. Future observations are never included in the rendered input.

For understanding-oriented QA, the queried dynamics channel (inflow or outflow) is rendered, whereas prediction-oriented QA displays both channels. The spatial granularity determines the spatial unit queried by the QA task rather than the resolution of the rendered heatmap. We consider four granularities:

- **Whole (W):** the complete native city grid, with the whole-city mean used as the corresponding aggregate value;
- **Coarse (C):** regions defined by a 2×2 partition of the native grid;
- **Local (L):** regions defined by a 4×4 partition of the native grid;
- **Fine-grained (F):** individual native-grid cells.

For coarse and local QA, region boundaries and region-level mean values are overlaid on the native-grid heatmaps. For fine-grained QA, native row–column coordinates are shown and queried cells are explicitly highlighted. For whole-city QA, the complete native-grid heatmap is retained and its whole-city mean is provided. Thus, all spatial granularities preserve the original city geometry while providing task-specific spatial references for the VLM.

Table 4: Urban dynamics QA taxonomy. W/C/L/F denote whole-city, coarse-region, local-region, and native-grid granularities.

ID	Scope	Task	Levels
U1	Single	Temporal sequence reading	W/C/L/F
U2	Single	Temporal difference computation	W/C/L/F
U3	Single	Temporal pattern classification	W/C/L/F
U4–U6	Multi	Multi-city temporal reading, computation, and comparison	W/C/L/F
U7	Single	Spatial value reading	C/L/F
U8	Single	Spatial difference computation	C/L/F
U9	Single	Dominant spatial-unit identification	C/L/F
U10–U12	Multi	Multi-city spatial reading, concentration, and comparison	C/L/F
U13	Single	Space–time value reading	C/L/F
U14	Single	Space–time difference computation	C/L/F
U15	Single	Hotspot-movement reasoning	C/L/F
U16–U18	Multi	Multi-city space–time reading, computation, and comparison	C/L/F
P1	Single	Future trend classification	W/C/L/F
P2	Single	Future dominant-location prediction	C/L/F
P3	Single	Multi-horizon numerical prediction	W/C/L/F
P4	Single	Dense native-grid prediction	F
P5–P8	Multi	Multi-city counterparts of P1–P4	Corresponding

For multi-city QA, each city is rendered independently at its own native resolution and presented to the VLM together with the task prompt. No spatial resizing, cell-level correspondence, or common grid is imposed across cities.

A.4.2 QA Taxonomy

Table 4 summarizes the QA families. Understanding-oriented tasks cover value reading, numerical computation, pattern reasoning, and cross-city comparison across temporal, spatial, and joint spatial–temporal settings. Prediction-oriented tasks extend these grounded capabilities to future trend, dominant-location, numerical, and dense-map forecasting.

Whole-city granularity is omitted from spatial and joint spatial–temporal tasks when a single spatial unit would make the associated reasoning degenerate.

A.4.3 Target Construction

Reading tasks derive target values from the underlying historical dynamic arrays. Computation tasks capture temporal, spatial, or joint space–time differences, as well as spatial concentration statistics when appropriate. Pattern-oriented tasks characterize temporal evolution, dominant spatial activity, or hotspot movement. Cross-city tasks extend these reasoning operations across multiple cities and compare the resulting quantities.

To make hotspot concentration comparable across granularities with different numbers of spatial units, we use the normalized peak-share score

$$C = \frac{N \frac{\max_i x_i}{\sum_j x_j} - 1}{N - 1}, \tag{19}$$

where N is the number of spatial units. We set $C = 0$ when the total flow is zero. Under this definition, uniform activity corresponds to $C = 0$, whereas activity concentrated entirely in one unit corresponds to $C = 1$.

For hotspot-movement tasks, tied maximum-flow units are represented by their centroid rather than by an arbitrarily selected winner. This avoids introducing artificial motion caused by exact ties. Other dominant-location tasks use deterministic tie handling to produce unique targets.

Prediction-oriented QA likewise uses historical-only visual inputs but derives supervision from the next $K = 4$ observations. P1/P5 classify future trends, P2/P6 predict future dominant spatial

units, P3/P7 supervise multi-horizon inflow and outflow values, and P4/P8 supervise complete future native-grid tensors.

A.4.4 FINE-GRAINED AND MULTI-CITY CONSTRUCTION

Fine-grained sampling is designed to preserve naturally inactive cells while also covering cells with different levels of activity and temporal variation. Reference sampling statistics are estimated from training histories, while cell-selection decisions for prediction-oriented QA depend only on the available historical window and never on future targets.

Multi-city QA preserves the native geometry of every city. Coarse and local regions are defined within each city's own native grid, while fine-grained cells are selected in their respective native coordinate systems. No cell-level spatial correspondence is assumed across cities. Multi-city windows are also not required to share the same absolute timestamp. The resulting construction therefore supports transferable urban reasoning without requiring spatial resizing, cell-level alignment, or synchronized events.

A.4.5 SUPERVISION

Each QA family contains multiple paraphrased question variants. Every example contributes token-level language supervision. Continuous scalar or vector targets additionally activate an auxiliary value loss, while dense P4/P8 targets activate a native-grid tensor loss.

Flow-valued numerical targets are normalized using statistics estimated exclusively from the corresponding city's training split, while quantities that are already dimensionless retain their native scale. The exact target routing and normalization procedures are provided in the released implementation.

The auxiliary numerical heads are used only during QA-guided grounding. After grounding, downstream forecasting uses the resulting VLM representation, and the QA-specific output heads are no longer required. Detailed template definitions, sampling rules, target construction, and rendering procedures are provided in the released implementation.

A.5 DERIVATION OF NASH-BALANCED CITY WEIGHTS

For completeness, we derive the city-weight equation used in Section 3.4. We assume that the feasible set is non-empty, i.e., there exists a direction $\mathbf{d} \in \mathcal{B}_\epsilon$ such that $\mathbf{g}_c^\top \mathbf{d} > 0$ for all $c \in \mathcal{C}$. Define

$$F(\mathbf{d}) = \sum_{c \in \mathcal{C}} \log\left(\mathbf{g}_c^\top \mathbf{d}\right), \tag{20}$$

where

$$\mathcal{B}_\epsilon = \{\mathbf{d} : \|\mathbf{d}\|_2 \le \epsilon\}. \tag{21}$$

The gradient of F is

$$\nabla_{\mathbf{d}} F(\mathbf{d}) = \sum_{c \in \mathcal{C}} \frac{\mathbf{g}_c}{\mathbf{g}_c^\top \mathbf{d}}. \tag{22}$$

For any feasible $\mathbf{d}$ and $\kappa > 1$ such that $\kappa \mathbf{d} \in \mathcal{B}_\epsilon$,

$$F(\kappa \mathbf{d}) = F(\mathbf{d}) + |\mathcal{C}| \log \kappa. \tag{23}$$

Hence, the objective increases under positive radial scaling, and the optimum $\mathbf{d}^*$ lies on the boundary $\|\mathbf{d}^*\|_2 = \epsilon$.

Applying the KKT condition using the equivalent squared-norm constraint $\frac{1}{2}\|\mathbf{d}\|_2^2 \le \frac{1}{2}\epsilon^2$ gives

$$\sum_{c \in \mathcal{C}} \frac{\mathbf{g}_c}{\mathbf{g}_c^\top \mathbf{d}^*} = \lambda \mathbf{d}^*, \qquad \lambda > 0. \tag{24}$$

Therefore,

$$\mathbf{d}^* = \sum_{c \in \mathcal{C}} \frac{1}{\lambda \mathbf{g}_c^\top \mathbf{d}^*} \mathbf{g}_c. \tag{25}$$

Define

$$\beta_c = \frac{1}{\lambda \mathbf{g}_c^\top \mathbf{d}^*} > 0. \tag{26}$$

Then

$$\mathbf{d}^* = \mathbf{G}\boldsymbol{\beta} = \sum_{c \in \mathcal{C}} \beta_c \mathbf{g}_c. \tag{27}$$

From Eq. 26,

$$\mathbf{g}_c^\top \mathbf{d}^* = \frac{1}{\lambda \beta_c}. \tag{28}$$

Substituting $\mathbf{d}^* = \mathbf{G}\boldsymbol{\beta}$ and stacking the resulting conditions over all cities gives

$$\mathbf{G}^\top \mathbf{G}\boldsymbol{\beta} = \frac{1}{\lambda}\boldsymbol{\beta}^{-1}. \tag{29}$$

Using $\boldsymbol{\Gamma} = \mathbf{G}^\top \mathbf{G}$,

$$\boldsymbol{\Gamma}\boldsymbol{\beta} = \frac{1}{\lambda}\boldsymbol{\beta}^{-1}. \tag{30}$$

We absorb the common positive scale by defining

$$\boldsymbol{\alpha} = \sqrt{\lambda}\,\boldsymbol{\beta}. \tag{31}$$

It follows that

$$\boldsymbol{\Gamma}\boldsymbol{\alpha} = \boldsymbol{\alpha}^{-1}, \tag{32}$$

which is the city-weight equation used in the main text. Moreover,

$$\mathbf{G}\boldsymbol{\alpha} = \sqrt{\lambda}\,\mathbf{G}\boldsymbol{\beta} = \sqrt{\lambda}\,\mathbf{d}^*. \tag{33}$$

Thus, $\mathbf{G}\boldsymbol{\alpha}$ is proportional to the Nash bargaining direction $\mathbf{d}^*$. Since this common positive scale can be absorbed into the learning rate, we use $\mathbf{G}\boldsymbol{\alpha}$ as the Nash-balanced shared gradient in Eq. 12.

A.6 BASELINES

Overview of Baselines. We compare **UrbanBind** with two categories of representative baselines: (1) *general-purpose and foundation urban models* designed to learn transferable spatial-temporal representations across heterogeneous urban scenarios, and (2) *cross-city and language-augmented urban dynamics models* designed to improve knowledge transfer across cities or incorporate pre-trained language-model knowledge into urban dynamics prediction. All baselines are evaluated under the same urban dynamics forecasting setting for consistent comparison. The *general-purpose and foundation urban models* include **UniST** Yuan et al. (2024), **UrbanGPT** Li et al. (2024b), **UrbanDiT** Yuan et al. (2026b), and **UniFlow** Yuan et al. (2026a). The *cross-city and language-augmented urban dynamics models* include **RegionTrans** Wang et al. (2018), **CrossTReS** Jin et al. (2022), **STBaT** Hao et al. (2025), and **PLM-CUP** Zhou et al. (2026).

Details of Baselines. For all baselines, we use an 8-step historical window to predict the subsequent 4 steps. NYCTAXI, NYC-BIKE, and BIKECHI are used as source datasets, while BIKEDC is held out from source training and used only for cross-city adaptation and evaluation. We preserve the core architecture and learning mechanism of each method and make only the necessary modifications to support the common forecasting horizon, heterogeneous grid sizes, and multi-source setting.

- **CrossTReS Jin et al. (2022)**: CrossTReS performs selective cross-city transfer by learning target-aware source-region weights from domain-adapted regional representations and joint meta-learning. We retain its ST-Net predictor, consisting of three 3×3 residual blocks with 64 channels and a single-layer LSTM with 128 hidden units, together with its multi-view regional feature module based on two-layer, two-head GATs with 64-dimensional embeddings, node-level MMD alignment, and edge-level relation alignment. To extend the original single-source formulation, we sequentially train a shared prediction and weighting model over the three source datasets while preserving model parameters and optimizer states across datasets. The learned weighting mechanism is subsequently used during few-shot adaptation on BIKEDC.

- **PLM-CUP Zhou et al. (2026)**: PLM-CUP bridges urban flow and language representations using a semantic bridge encoder, a LoRA-adapted pretrained language model, and a task-specific spatio-temporal adapter. We use its Qwen3-0.6B variant and retain the trend–periodicity decomposition, GDAConv-based spatial modeling, domain-shift regularization, and temporal calendar features. To support heterogeneous city grids, we maintain city-specific node embeddings while sharing the remaining transferable modules. We sequentially train the shared PLM-CUP components over the three source datasets with persistent model and optimizer states, and then transfer the learned model to BIKEDC for few-shot adaptation, freezing the decomposition modules following the original Phase-2 design.
- **RegionTrans Wang et al. (2018)**: RegionTrans transfers knowledge by matching target regions with source regions exhibiting similar flow patterns and aligning their region-level representations during target adaptation. We adopt its service-data-based matching (S-Match) variant and retain the original backbone with two 5×5 ConvLSTM layers with 32 hidden states, followed by BatchNorm and two 1×1 convolutional layers. Since the source and target datasets come from different calendar periods, region matching is based on 24-hour flow profiles computed from source training data and the observed target support set, with each target region matched to the source region having the highest normalized correlation. We sequentially train a shared backbone over the three source datasets and, during BIKEDC adaptation, average the matched-region representation losses from all source cities and jointly optimize them with the target prediction loss.
- **STBaT Hao et al. (2025)**: STBaT improves cross-city transfer by correcting regional distribution imbalance through a Regional Imbalance Acquisition Module (RIAM) and a Spatio-Temporal Balanced Learning Module (STBLM). We retain its ST-Net backbone with three 64-channel residual blocks, a single-layer LSTM with 128 hidden units, a 256-dimensional regional representation, and a feedforward predictor. RIAM estimates regional pattern density and inter-regional similarity, which guide the density-aware prediction reweighting and similarity-aware distribution contrastive learning in STBLM. We jointly train a single model on the three source datasets, placing sampled source regions in a shared contrastive space and optimizing their prediction losses together. The resulting spatio-temporal prediction network is then fine-tuned on the few-shot BIKEDC support set.
- **UniFlow Yuan et al. (2026a)**: UniFlow learns a unified urban flow model through spatio-temporal patching, Transformer-based sequence modeling, and Spatio-Temporal Memory Retrieval Augmentation (ST-MRA). Since all datasets in our setting are grid-based, we use its grid branch and retain the Conv3D patching module, encoder–decoder Transformer, and four ST-MRA memories for time-domain, frequency-domain, time-spatial, and frequency-spatial patterns. Inflow and outflow are treated as channel-independent instances with shared model parameters, while dataset-compatible spatial patching modules accommodate different grid resolutions. We jointly train one UniFlow model by interleaving batches from the three source datasets through the same backbone, memory modules, and optimizer, and then fine-tune the resulting model on BIKEDC for few-shot adaptation.
- **UniST Yuan et al. (2024)**: UniST learns transferable spatio-temporal representations through masked pre-training followed by knowledge-guided prompt learning. We retain its spatio-temporal patching and Transformer encoder–decoder architecture. In the first stage, a shared model is jointly pre-trained on the three source datasets using random, temporal, tube, and block masking. In the second stage, the pretrained model is augmented with the original prompt network, which models spatial closeness, spatial hierarchy, temporal closeness, and temporal periodicity through learnable spatial and temporal memory pools. Inflow and outflow are processed as channel-independent instances, and source-city batches are interleaved through a shared model and optimizer in both stages. The resulting model is subsequently adapted to BIKEDC using the same prompt-based forecasting objective.
- **UrbanDiT Yuan et al. (2026b)**: UrbanDiT formulates spatio-temporal forecasting as conditional generation with a diffusion Transformer and unified prompt learning. We use its grid-data pipeline with spatio-temporal patching and factorized temporal and spatial attention, and retain the data-driven temporal, frequency, and spatial key–value memory pools together with the mask-derived task prompt. We specialize the original multi-task framework to forward prediction and treat inflow and outflow as channel-independent instances

Table 5: Few-shot transfer results on the held-out BIKEDC dataset with $N_{\text{shot}} \in \{50, 100, 200\}$ target-city training samples (MAE/RMSE).

Method	$N_{\text{shot}} = 50$		$N_{\text{shot}} = 100$		$N_{\text{shot}} = 200$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
STBaT	2.41	12.67	2.43	11.87	2.04	10.60
CrossTReS	2.78	12.06	2.07	**11.37**	1.94	10.88
PLM-CUP	2.93	13.21	3.35	11.61	4.14	12.06
RegionTrans	9.49	19.38	7.62	22.28	6.68	22.89
UniST	3.54	13.66	2.94	12.26	2.32	10.62
UniFlow	3.38	13.06	3.00	12.84	2.85	11.82
UrbanDiT	2.14	11.76	2.88	11.90	1.97	11.02
UrbanGPT	2.67	**11.67**	2.66	11.83	2.00	**10.59**
UrbanBind	**2.13**	**11.67**	**2.05**	**11.37**	**1.92**	**10.59**

with shared model parameters. A single UrbanDiT model is jointly trained by interleaving batches from the three source datasets, using rectified-flow denoising on the masked future interval while keeping the historical observations fixed. The resulting model is then fine-tuned on the few-shot BIKEDC support set.

- **UrbanGPT Li et al. (2024b)**: UrbanGPT integrates a spatio-temporal dependency encoder with a language model by projecting urban dynamics into the LLM representation space. We retain its gated dilated temporal convolutional encoder, lightweight alignment module, spatio-temporal historical and forecasting embeddings, and regression-based prediction head. To keep the LLM-based baseline within our computational budget, we replace the original Vicuna-7B backbone with a frozen Qwen2.5-1.5B model while preserving the original alignment and regression mechanisms. We first pre-train one shared spatio-temporal encoder on the three source datasets. During subsequent instruction alignment, both the pretrained encoder and the LLM remain frozen, while the alignment projector, spatio-temporal embeddings, and regression head are optimized on the source datasets and subsequently updated using the few-shot BIKEDC support set.

B Additional Experiments

Few-Shot Generalization to an Unseen City. To further evaluate cross-city generalization, we conduct a held-out-city experiment on BIKEDC. All models are first trained on NYCTAXI, BIKECHI, and NYC-BIKE and are then adapted to the previously unseen BIKEDC dataset using a limited number of target-city training samples. We evaluate three few-shot settings, $N_{\text{shot}} \in \{50, 100, 200\}$. Table 5 reports the corresponding MAE and RMSE results.

Across all three settings, UrbanBind demonstrates strong transfer to the unseen city. It achieves the lowest MAE for $N_{\text{shot}} = 50$, 100, and 200, while obtaining the best or tied-best RMSE in every setting. As the number of target-city samples increases from 50 to 200, UrbanBind's MAE decreases from 2.13 to 1.92, showing that it can effectively leverage additional target-city supervision while remaining competitive under limited adaptation data.

These results further support the cross-city generalization capability of UrbanBind. QA-Guided Spatial-Temporal Knowledge Grounding provides transferable knowledge of urban dynamics and forecasting tasks, while Multi-City Urban Dynamics Encoding preserves city-specific native-grid structures and facilitates cross-city information exchange. Together with the VLM-conditioned projection and fusion mechanism, these components enable effective adaptation to an unseen city across different few-shot training budgets.

B.1 Additional Grounding and Representation Analysis

Task-Wise Spatial-Temporal Knowledge Grounding. Tab. 7 reports test-set cross-entropy (CE) and perplexity (PPL) across the major QA categories. The QA-grounded VLM consistently achieves

Table 6: Representation analysis across the two UrbanBind training stages and the VLM–dynamics integration pathway. For Stage 1 and Stage 2, lower probing errors and higher R^2 and Target CKA indicate more prediction-relevant representations. For VLM–dynamics integration, CKA measures structural correspondence with the QA-grounded VLM representation.

Analysis	Representation	MAE ↓	RMSE ↓	R^2 ↑	CKA ↑
Stage 1	Before grounding: $\mathbf{s}_q^{\text{base}}$	0.158	0.252	0.112	0.019
	After grounding: $\mathbf{s}_q^{\text{ground}}$	**0.098**	**0.163**	**0.626**	**0.597**
Stage 2	Before training: $\mathbf{r}^{\text{init}}$	0.090	0.149	0.702	0.429
	After training: $\mathbf{r}^{\text{trained}}$	**0.083**	**0.129**	**0.777**	**0.741**
Integration	Pre-integration: $\mathbf{s}$ vs. $\mathbf{h}_0$	–	–	–	0.228
	Post-fusion: $\mathbf{s}$ vs. $\mathbf{z}$	–	–	–	**0.994**

Notes. For Stage 1 and Stage 2, MAE, RMSE, and R^2 are obtained using the same ridge-probe protocol, while Target CKA measures correspondence between the learned representation and future urban dynamics. For the integration analysis, CKA is computed directly between the QA-grounded VLM representation $\mathbf{s}$ and the corresponding pre-integration or fused representation within the trained model.

Table 7: Task-wise analysis of spatial-temporal knowledge grounding on the QA test set.

QA Category	Base CE ↓	Grounded CE ↓	Base PPL ↓	Grounded PPL ↓
Temporal Understanding	0.560	**0.022**	1.750	**1.023**
Spatial Understanding	1.052	**0.151**	2.865	**1.163**
Spatial-Temporal Understanding	1.083	**0.048**	2.952	**1.049**
Forecasting QA	0.753	**0.256**	2.124	**1.292**

Notes. CE and PPL are computed over the target answer tokens on the QA test set. Forecasting QA aggregates P1–P3 and P5–P7; P4 and P8 are excluded because their textual responses describe tensor structure, while their dense-map targets are evaluated separately through numerical prediction metrics.

lower CE and PPL than the base VLM on temporal, spatial, spatial-temporal, and forecasting QA tasks. This consistent improvement indicates that QA-Guided Spatial-Temporal Knowledge Grounding enhances the VLM across diverse urban reasoning and forecasting capabilities rather than benefiting only a specific task type.

Representation Quality and VLM–Dynamics Integration. Tab. 6 analyzes representation quality across the two training stages and the VLM–dynamics integration pathway. For Stage 1, the QA-grounded representation $\mathbf{s}_q^{\text{ground}}$ contains more prediction-relevant information than the base representation $\mathbf{s}_q^{\text{base}}$, as reflected by improved probing performance and higher Target CKA. Stage 2 further improves the contextualized forecasting representation $\mathbf{r}$, showing that Multi-City Urban Dynamics Encoding and subsequent VLM–dynamics integration provide complementary information for forecasting. For the integration pathway, the relatively weak correspondence between the QA-grounded VLM representation $\mathbf{s}$ and numerical representation $\mathbf{h}_0$ becomes substantially stronger after the VLM-Conditioned Multi-Projector and feature-wise gated fusion. Together with the corresponding ablation results, this suggests that the integration mechanism incorporates useful numerical dynamics information while preserving strong correspondence with the QA-grounded VLM representation.