

---

# Get Rid of Task Isolation: A Continuous Multi-task Spatio-Temporal Learning Framework

---

Zhongchao Yi<sup>1</sup>, Zhengyang Zhou<sup>1,2,3,\*</sup>, Qihe Huang<sup>1</sup>, Yanjiang Chen<sup>1</sup>,  
Liheng Yu<sup>1</sup>, Xu Wang<sup>1,2</sup>, Yang Wang<sup>1,2,\*</sup>

<sup>1</sup>University of Science and Technology of China (USTC), Hefei, China

<sup>2</sup>Suzhou Institute for Advanced Research, USTC, Suzhou, China

<sup>3</sup>State Key Laboratory of Resources and Environmental Information System, Beijing, China

{zhongchaoyi, hqh, yjchen, yuliheng}@mail.ustc.edu.cn,

{wx309, zzy0929\*, angyan\*}@ustc.edu.cn

## Abstract

Spatiotemporal learning has become a pivotal technique to enable urban intelligence. Traditional spatiotemporal models mostly focus on a specific task by assuming a same distribution between training and testing sets. However, given that urban systems are usually dynamic, multi-sourced with imbalanced data distributions, current specific task-specific models fail to generalize to new urban conditions and adapt to new domains without explicitly modeling interdependencies across various dimensions and types of urban data. To this end, we argue that there is an essential to propose a Continuous Multi-task Spatio-Temporal learning framework (CMuST) to empower collective urban intelligence, which reforms the urban spatiotemporal learning from single-domain to cooperatively multi-dimensional and multi-task learning. Specifically, CMuST proposes a new multi-dimensional spatiotemporal interaction network (MSTI) to allow cross-interactions between context and main observations as well as self-interactions within spatial and temporal aspects to be exposed, which is also the core for capturing task-level commonality and personalization. To ensure continuous task learning, a novel Rolling Adaptation training scheme (RoAda) is devised, which not only preserves task uniqueness by constructing data summarization-driven task prompts, but also harnesses correlated patterns among tasks by iterative model behavior modeling. We further establish a benchmark of three cities for multi-task spatiotemporal learning, and empirically demonstrate the superiority of CMuST via extensive evaluations on these datasets. The impressive improvements on both few-shot streaming data and new domain tasks against existing SOAT methods are achieved. Code is available at <https://github.com/DILab-USTCSZ/CMuST>.

## 1 Introduction

Spatiotemporal learning has become a pivotal technique to enable smart and convenient urban lives, benefiting diverse urban applications from intra-city travelling, environment controlling to location-based POI recommendation, and injecting the vitality into urban economics. Existing spatiotemporal learning solutions [19, 9, 38, 40, 51, 42, 37, 10, 8] focus on improving performances of a task-specific model independently where these methods devise various spatial learning blocks [44, 8, 43] and temporal dependency extraction modules [40, 34, 9] to model the spatiotemporal heterogeneity.

Actually, urban spatiotemporal systems are usually highly dynamic with emerging new data modality, leading to serious generalization issue on both data pattern and task adaptation. As illustrated in

---

\*Yang Wang and Zhengyang Zhou are corresponding authors.

Figure 1, the traffic volume patterns can evolve with urban expansion and establishment of new POIs. Concurrently, with increasing attention on road safety, traffic accident prediction has become a new task in intelligent transportation that inevitably suffers from the cold-start issue. Unfortunately, traditional task-specific models usually assume that data on a single task follows independent and identical distribution and are intensively available where such assumption directly leads to failures on data sparsity scenarios and generalization to new tasks. In fact, given diverse datasets, separately training single task-specific spatiotemporal models is cost sensitive and will trap the models into isolation. To this end, we argue that a continuous multi-task spatiotemporal learning framework is highly desirable to facilitate the task-level cooperation. It is even more interesting and exciting to jointly model multi-domain datasets with multi-task learning, which empowers understanding spatiotemporal system in a holistic perspective and reinforce each individual task by exploiting the collective intelligence from diverse data domains.

The key towards exploiting task-wise correlations for mutual improvement is to capture the common interdependencies across data dimensions and domains. Current multi-task learning schemes either investigate the regularization effects between auxiliary and main tasks [31, 17], or devise loss objectives to constrain the consistency between each task [7, 32]. Actually, given a spatiotemporal domain, there must be common interdependencies across different data types and domains, which are valuable for cooperated learning. Even prosperity of multi-task learning and spatiotemporal forecasting, there are never a systematic solution on how various sourced data from different tasks reinforce a specific task with multi-task learning. More specifically, interrelations among observations in a system can be decomposed into multi-dimensional interactions, i.e., from contextual environment to respective spatial relations and temporal evolutions, spatial-temporal level interactions and relations across different data domains. Considering the multi-level semantic correlations, learning the system in a holistic perspective is required and further pose two challenges to a continuous multi-task spatiotemporal learning framework, i.e., 1) How to disentangle complex associations between data dimensions and domains, and capture such dependencies in an adaptive manner to improve spatiotemporal representation, hence facilitating the extraction of common patterns for mutual enhancement. 2) How to exploit task-level commonality and personality to jointly model the multi-task datasets, and exploit such extracted task-level commonality and diversity to reinforce respective task for getting rid of task isolation.

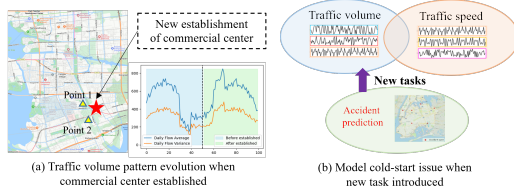


Figure 1: Illustration of evolution on data patterns and learning tasks.

In our work, a Continuous Multi-task SpatioTemporal learning framework, CMuST, is proposed to jointly model multiple datasets in an integrated urban system thus reinforcing respective learning tasks. Specifically, a Multi-dimensional Spatio-Temporal Interaction Network (MSTI) is first devised to dissect interactions across data dimensions, including context-spatial, context-temporal and self-interaction within spatial and temporal dimensions. MSTI enables improved spatiotemporal representation with interactions, and also provides disentangled patterns to support commonality extraction. After that, a Rolling Adaptation training scheme, RoAda, which iteratively captures the task-wise consistency and task-specific diversity, is proposed. In RoAda, to maintain task characterization, a task-specific prompt is constructed to preserve unique patterns distinguishing from other tasks by compressing data patterns via an AutoEncoder. To capture the commonality across tasks, we propose a weight behavior modeling strategy to iteratively highlight the minimized variations of learnable weights, i.e., stable interactions during continuous training, which encapsulates crucial task-level commonalities. This approach not only stabilizes learning through continuous task rolling, but alleviates the cold-start over new tasks with shared patterns. Finally, a task-specific refinement is devised to leverage commonality and fine-grained adaptation on specific tasks.

The contributions of this work can be three-fold. 1) The first continuous multi-task spatiotemporal learning framework, CMuST to jointly model learning tasks in a same spatiotemporal domain, which not only reinforces individual correlated learning task in collective perspective, but also help understand the cooperative mechanism of dynamic spatiotemporal systems. 2) Technically, two learning modules, MSTI and RoAda are proposed to dissect the impacts and interactions over multi-dimensions, and iteratively update the task-wise commonality and generate individual personalization to continuous task adaptation in multi-task learning. 3) We construct benchmark datasets in each of

three cities, where two of them consists of at least 3 types of observations within same spatiotemporal domain. The extensive experiments demonstrate the superiority on enhancement of each individual task with limited data and the interpretation of task-wise continuous learning.

## 2 Related Work

**Spatiotemporal forecasting** is an emerging technique to capture the dynamic spatial and temporal evolution for diverse urban predictions, where the methods can be divided into machine learning-based and deep learning-based. Conventional solutions rely on complex mathematical tools to simulate the dynamics including ARIMA [25], SVR [3] and matrix-factorization learning [28] for capturing spatial correlations. With deep learning solutions flourishing, Convolution Neural Networks (CNNs) [40, 11, 46] are exploited to imitate the temporal dependencies and GNNs [43, 13, 36] are utilized to imitate spatial propagation. Meanwhile, by taking the advantage of the flexibility and interpretability of attention, Spatial-Temporal Attention [9], and Vision Transformer (ViT) [6] are introduced to improve spatiotemporal representation. Also there are also many methods for temporal periodicity capturing [12, 20]. More specifically, DG2RNN [49] designs a dual-graph convolutional module to capture local spatial dependencies from both road distance and adaptive correlation perspectives. PDFormer [15] designs a spatial self-attention and introduces two graph masking matrices to highlight the spatial dependencies of short- and long-range views. TESTAM [18] uses time-enhanced ST attention by mixture-of-experts and modeling both static and dynamic graphs. Even so, most solutions focus on single-task intelligence, fail to deal with complex interactions between data dimensions and never extract task-level commonality patterns, resulting in inferior performances on exploiting collective intelligence over multiple tasks. In contrast, we merge the gap by disentangling learnable interaction patterns and exploring rolling task adaptations.

**Multi-task learning.** Plenty of efforts have been made on multi-task learning (MTL) and MTL can be elaborated by two-fold, i.e., feature-based and parameter-based. Feature-based MTL [2, 24] learns a common feature representation for different tasks, but it may be easily affected by outlier tasks. To this end, parameter-based MTL [30, 16] is devised to exploit model parameters to relate different tasks, which is expected to learn robust parameters. Majority of these MTL schemes either concentrate on the diversity design and regularization effects of auxiliary tasks to main task [30, 27], or construct loss functions to ensure task-wise consistency [47]. A pioneering work investigates a gradient-driven task grouping to realize multi-task learning [41] where the focuses are text and images using pre-trained CLIP model. For spatiotemporal learning, RiskOracle [52] and RiskSeq [53] are proposed to simultaneously predict multi-grained risks and auxiliary traffic elements. More recently, UniST [45] and UniTime [22] construct a unified model for spatiotemporal and time-series prediction. However, all researches on ST learning still never dissect task-wise correlations, especially capture explicit consistency and diversity among tasks and investigate how each task reinforce the core task, which is of great significance for performance and interpretability in MTL.

**Continuous learning and task continuous learning.** Continuous learning (CL) usually keeps long-term and important information while updates model memories with newly arrived instances [29, 5, 35]. For spatiotemporal forecasting, Chen, et, al. [5] proposes a historical-data replay strategy, TrafficStream, to update the neural network with all nodes feeding, while PECPM [35] manages a pattern bank with conflict nodes, which reduces the memory storage burdens. Most existing CL solutions are designed for homogeneous sourced data. Then there are very few research investigating task-level continuous learning where task can be converted from one to another. A pioneering work, CLS-ER [1] realizes class-level and domain-level continuous learning with a dual memory to respectively store instances and build long-short term memories. Even so, the data input to CLS-ER is with same image-like property and scales, which is still far away from task diversity. We argue that learning continuously on task levels can increase the data tolerance over low-quality data and also generalization capacity on new tasks. In this work, by taking bonus of continuous learning, we propose to iteratively assimilate the commonality and extract personalization to reinforce each learning task and realize continuous multi-task spatiotemporal intelligence.

## 3 Preliminary

*Definition 1 (Spatiotemporal features.)* Spatiotemporal features refer to the data points collected by sensors deployed in urban environments, such as traffic dynamics on roads. We define a spatiotemporal

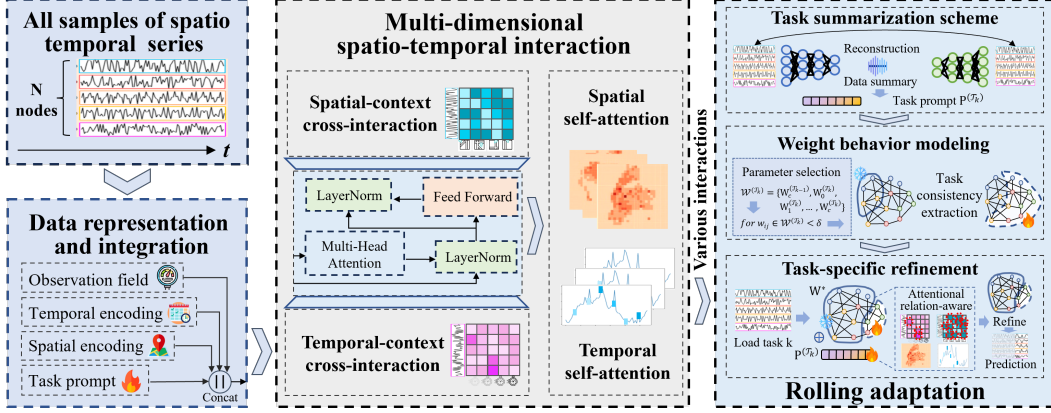


Figure 2: Framework overview of CMuST.

feature at a specific time and location as a vector  $X \in \mathbb{R}^C$ , where  $C$  represents the number of attributes (e.g., vehicle flow, speed) recorded by the sensor. We generalize this to define spatiotemporal data collected over a period as  $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$ , where  $T$  denotes the number of discrete time intervals, and  $N$  denotes the number of spatial nodes corresponding to sensor locations.

**Definition 2 (Spatiotemporal prediction.)** Spatiotemporal prediction involves forecasting future values of spatiotemporal data based on historical observations. Specifically, given a historical dataset  $\mathbf{X} = [X_{t-T}, \dots, X_{t-1}, X_t] \in \mathbb{R}^{T \times N \times C}$ , the objective is to predict future observations  $\hat{\mathbf{Y}} = [X_{t+1}, \dots, X_{t+T'}] \in \mathbb{R}^{T' \times N \times C}$ . Here,  $T'$  represents the prediction horizon, the number of future time steps we seek to forecast, which for simplicity is set equal to  $T$  in this paper.

**Definition 3 (Multi-task spatiotemporal learning.)** Given a set of spatiotemporal tasks in an integrated urban system  $\mathbb{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k\}$  where each task  $\mathcal{T}_i$  is associated with a dataset  $\mathcal{D}_i = \{\mathbf{X}_i, \mathbf{Y}_i\}$ . Here,  $\mathbf{X}_i \in \mathbb{R}^{T_i \times N_i \times C_i}$  represents the input features collected over  $T_i$  time steps, across  $N_i$  spatial locations, and  $C_i$  feature dimensions, and  $\mathbf{Y}_i \in \mathbb{R}^{T'_i \times N_i \times C_i}$  represents the corresponding targets. Multi-task spatiotemporal learning aims to learn a function  $f : \mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_{k-1} \rightarrow \hat{\mathbf{Y}}_0, \hat{\mathbf{Y}}_1, \dots, \hat{\mathbf{Y}}_{k-1}$  that optimally predicts all target  $\mathbf{Y}_i$  from their respective inputs  $\mathbf{X}_i$ , leveraging shared and unique patterns across tasks to improve generalization performance.

## 4 Methodology

### 4.1 Framework Overview

The CMuST is crafted to advance urban intelligence through a synergistic integration of three components in Figure 2. We process and standardize diverse urban data into a harmonized format, and propose the MSTI to intricately dissect the complex interactions within spatiotemporal data, and devise a RoAda to dynamically fine-tune the model via continuous and careful updating, ensuring robust adaptability and consistent performance across fluctuating urban environments.

### 4.2 Data Representation and Integration

To harness various data domains within urban spatiotemporal systems as well as data interactions between diverse dimensions, the first task is to appropriately process main ST observations, spatial indicator, and temporal indicator to create a comprehensive and integrated data representation tailored for multi-task ST learning, enabling further interactive modeling between them.

To be specific, the main observation data, namely target of interest of urban datasets, are denoted as  $\mathbf{X}_{obs} \in \mathbb{R}^{T \times N \times C}$ , and then mapped into a spatiotemporal representation  $\mathbf{E}_{obs} \in \mathbb{R}^{T \times N \times d_{obs}}$  via an MLP  $\text{ObsMLP}(\mathbf{X}_{obs}; \theta_{obs})$ . Similarly, the spatial indicator  $\mathbf{X}_s \in \mathbb{R}^{T \times N \times 2}$ , consisting of longitude and latitude coordinates, is applied with a linear layer  $\text{SpatialMLP}(\mathbf{X}_s; \theta_s)$  to produce the spatial representation  $\mathbf{E}_s \in \mathbb{R}^{T \times N \times d_s}$ . Temporal indicator comprises day-of-week, time-of-day  $\mathbf{X}_{dow}, \mathbf{X}_{tod} \in \mathbb{R}^{T \times N}$  and timestamps  $\mathbf{X}_{ts} \in \mathbb{R}^{T \times N \times 6}$ , which are further compressed into hidden

representation, i.e.,  $\mathbf{E}_t = \text{TemporalMLP}(\mathbf{E}_{ts} \parallel \mathbf{E}_{dow} \parallel \mathbf{E}_{tod}; \theta_t) \in \mathbb{R}^{T \times N \times d_t}$ , enabling the model to capture the periodicity and sequential features of temporal data.  $\mathbf{P}_\tau$  is a task-specific prompt<sup>2</sup> to ensure task-awareness, and is integrated into the final embedding. Given the task  $\mathcal{T}_k$ ,

$$\mathbf{H}^{(\mathcal{T}_k)} = \mathbf{E}_{obs} \parallel \mathbf{E}_s \parallel \mathbf{E}_t \parallel \mathbf{P}^{(\mathcal{T}_k)} \quad (1)$$

where  $\parallel$  denotes vector concatenation, combining spatial, temporal, and observational embeddings with the task prompt into a comprehensive representation  $\mathbf{H}^{(\mathcal{T}_k)} \in \mathbb{R}^{T \times N \times d_h}$ . We will use  $\mathbf{H}$  as representation for a specific task in following sections.

### 4.3 Multi-dimensional Spatio-Temporal Interaction

Spatiotemporal observations are usually complex with multiple-level interactions where such interactions and correlations play vital roles in enhancing commonality learning through different learning domains. To this end, we devise an MSTI, to intricately dissect and disentangle interactions within spatiotemporal data from spatial-temporal indicators to main observations by inheriting nice property of attention mechanisms, which all utilize transformed slices from the integrated representation  $\mathbf{H}$ .

**Spatial-context cross-interaction.** To quantitatively investigate how spatial indicator interact with main observations, we devise a multi-head cross-attention [14] architecture (MHCA) where spatial and observational components are alternately used as queries ( $\mathbf{Q}$ ) and key-value ( $\mathbf{KV}$ ) pairs:

$$\text{MHCA}^{(a,b)}(\mathbf{X}) = \left\|_{h=1}^{\text{head}} \text{CrossAttention}^{(a,b)}(h) \mathbf{W}^O \quad (2)$$

$$\text{CrossAttention}^{(a,b)}(h) = \text{Attention} \left( \mathbf{Q}_h^{(a)}, \mathbf{K}_h^{(b)}, \mathbf{V}_h^{(b)} \right) \quad (3)$$

$$\text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) = \text{softmax} \left( \frac{\mathbf{Q}_h \mathbf{K}_h^T}{\sqrt{D}} \right) \mathbf{V}_h \quad (4)$$

The symbol  $\parallel$  denotes the concatenation of multiple attention heads, and  $\mathbf{W}^O$  is the projection matrix that aligns the output dimensions with those of  $\mathbf{H}$ . Here, we let variables  $a$  and  $b$  indicate spatial ( $s$ ) indicator and main observation ( $o$ ), i.e.,  $a = s$  and  $b = o$ . Then the queries, keys, and values are generated through following transformations,

$$\mathbf{Q}_h^{(s)} = \mathbf{H}[\dots, \text{slice}^{(s)}] \mathbf{W}_h^{(Q_s)}, \quad \mathbf{K}_h^{(o)} = \mathbf{H}[\dots, \text{slice}^{(o)}] \mathbf{W}_h^{(K_o)}, \quad \mathbf{V}_h^{(o)} = \mathbf{H}[\dots, \text{slice}^{(o)}] \mathbf{W}_h^{(V_o)} \quad (5)$$

where  $\mathbf{W}$  transforms input data into dimension  $D$  of attention space,  $\text{slice}^{(s)}$  and  $\text{slice}^{(o)}$  denote the respective slices of  $\mathbf{H}$  for spatial and observational features. After computing attention scores, the embeddings are taken into a feed-forward network (FFN) to enhance the learning capabilities, where  $\text{FFN}(\mathbf{X}) = \max(0, \mathbf{X} \mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2$ . The final attention outputs are then normalized by,

$$\begin{aligned} \tilde{\mathbf{H}}^{(a,b)}[\dots, \text{slice}^{(b)}] &= \text{LN}(\text{FFN}(\text{LN}(\text{MHCA}^{(a,b)}(\mathbf{H}) + \mathbf{H}[\dots, \text{slice}^{(b)}]))) \\ &+ \text{LN}(\text{MHCA}^{(a,b)}(\mathbf{H}) + \mathbf{H}[\dots, \text{slice}^{(b)}])), \end{aligned} \quad (6)$$

where LN is layer normalization, the resulting matrices  $\tilde{\mathbf{H}}^{(s,o)}$  and  $\tilde{\mathbf{H}}^{(o,s)}$  are then concatenated back to  $\mathbf{H}$  at their respective dimensions as  $\tilde{\mathbf{H}}^{(SCCI)} \in \mathbb{R}^{T \times N \times d_h}$ , enriching the original representation with refined features that encapsulate intricate cross-dimensional relationships.

**Temporal-context cross-interaction.** To facilitate the attention computation with respect to temporal dimension, the representation is first transposed as  $\tilde{\mathbf{H}}^{(SCCI)} \in \mathbb{R}^{N \times T \times d_h}$  by denoting  $T$  as the sequence length for subsequent attention calculations. Then the step-wise positional encoding (refer to Appendix B.2) is introduced to allow our attention aware of specific temporal evolution.

The subsequent steps closely follow those of the spatial-context cross-attention mechanism, so the temporal-context cross-interaction (TCCI) can be performed as  $\hat{\mathbf{H}}^{(CI)} = \text{TCCI}(\tilde{\mathbf{H}}^{(SCCI)})$ , and final representation  $\hat{\mathbf{H}}^{(CI)} \in \mathbb{R}^{N \times T \times d_h}$  becomes the outcome of cross interactions between spatial and temporal dimensions.

**Self-interactions within Spatial and Temporal Aspects.** We proceed to apply self-interaction across different dimensions of the representations using self-attentions [21]. Specifically, we begin with

<sup>2</sup>We introduce a task summarization as the prompt (refer Section 4.4) to generate the distinct prompts.

temporal dimension, as the sequence length of representation dimension has naturally been aligned as  $T$ . This setup allows direct computation of the required self-attention,

$$\text{MHA}(\mathbf{X}) = \left\| \right\|_{h=1}^{\text{head}} \text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) \mathbf{W}^O \quad (7)$$

In this multi-head attention (MHA) configuration, the attention calculation (referred to in Equation 4, 5) involves queries, keys, and values, which is derived from the entire representation  $\tilde{\mathbf{H}}^{(CT)}$ , rather than individual slices. Then the output undergoes further processing through a FFN and non-linear transformations and LN to stabilize and enrich the feature representations similar Equation 6. The resulting  $\tilde{\mathbf{H}}^{(TSI)}$  in  $\mathbb{R}^{N \times T \times d_h}$  signifies the outcome of temporal self-interactions (TSI). Then the tensor is transposed to  $\tilde{\mathbf{H}}'^{(TSI)} \in \mathbb{R}^{T \times N \times d_h}$ . The final spatial self-interaction (SSI) computation is analogous to the temporal version, which refines spatial interactions and aggregates features across spatial nodes, as  $\tilde{\mathbf{H}} = \text{SSI}(\tilde{\mathbf{H}}'^{(TSI)})$ . The resulting tensor from this computation is  $\tilde{\mathbf{H}} \in \mathbb{R}^{T \times N \times d_h}$ , which represents the outcome of comprehensive multi-dimensional interactions.

We then adaptively integrate the interactions by a fusion strategy and Huber loss is adopted to ensure the robustness to outliers in spatiotemporal samples. Details can be found in Appendix B.3. Our MSTI allows extracting diverse interactions, including spatial-temporal domain interactions via designing cross attentions on respective indicators, and self interactions within respective dimensions, enhancing the data relation learning and supporting commonality extraction across task domains.

#### 4.4 Rolling Adaptation over Continuous Multi-task Spatio-Temporal Learning

To ensure continuous task learning, we propose a Rolling adaptation scheme, RoAda, to model the distinction and commonality among task domains. Our RoAda is composed of two stages with a warm-up for commonality extraction and a task-specific refinement. Before the task rolling, we construct prompts to distinguish personalization of each task. Thus, the commonality and diversity can be leveraged to boost individual task adaptation.

**Task summarization as prompts.** To capture the task distinction, we devise a task summarization by a Sampling-AutoEncoding scheme from each task. Consider task  $\mathcal{T}_k$ , main observation becomes  $\mathbf{X}^{(\mathcal{T}_k)} \in \mathbb{R}^{T_{all}^{(\mathcal{T}_k)} \times N \times C}$ . Such data is sampled by averaging observations over equivalent times of day, yielding a periodic sample representation  $\mathbf{X}_{samp}^{(\mathcal{T}_k)} \in \mathbb{R}^{L_t^{(\mathcal{T}_k)} \times N \times C}$ , where  $L_t^{(\mathcal{T}_k)}$  denotes the number of time slots for task  $k$  within a day. Since neural networks tend to fit any data regularity, the sampled features are fed into an autoencoder for extracting the compressed and distinguished data patterns. Given the encoding and decoding processes, i.e.,  $\phi : \mathcal{X}_{samp} \rightarrow \mathcal{S}$  and  $\psi : \mathcal{S} \rightarrow \mathcal{X}'_{samp}$ ,  $\mathbf{S}$  encapsulates the summarized core characteristics of the task. The decoding phase maps  $\mathbf{S}$  back to a reconstructed  $\mathbf{X}'_{samp}$ , by minimizing the mean squared reconstruction error,

$$\phi : \mathbf{S} = \text{sigmoid}(\mathbf{W}_s \mathbf{X}_{samp} + \mathbf{b}_s), \quad \phi, \psi = \arg \min_{\phi, \psi} \|\mathbf{X}_{samp} - (\psi \circ \phi) \mathbf{X}_{samp}\|^2 \quad (8)$$

where  $\mathbf{W}_s$  is the weight, and  $\mathbf{b}_s$  is the bias. Following the encoder, the summary features  $\mathbf{S}^{(\mathcal{T}_k)}$  are transformed into the  $k$ -th task prompt  $\mathbf{P}^{(\mathcal{T}_k)} = \text{FC}_p(\mathbf{S}^{(\mathcal{T}_k)}; \theta_p)$  with dimension alignment.

**Weight behavior modeling.** The first warm-up stage is designed via weight behavior modeling, which assimilates the regularity from task to task. This process not only adapts the model to new tasks but also solidifies its ability in generalizing across scenarios by capturing task-wise common relations with modeling of model weight behaviors.

We begin with the task  $\mathcal{T}_1$  via independently training the model until its performance stabilizes. By denoting  $\mathcal{M}$  as the model learned by MSTI, the training phase can be formally described as,

$$\mathbf{P}^{(\mathcal{T}_1)} \xrightarrow[\text{load}]{\text{prompt}} \mathcal{M}, \quad \text{Train}(\mathcal{M}(\mathcal{D}_{train}^{(\mathcal{T}_1)}; \mathbf{W}_{init})) \xrightarrow[\text{convergence}]{\text{until}} \mathbf{W}_c^{(\mathcal{T}_1)} \quad (9)$$

where  $\mathbf{W}_{init}$  are the initialized weights,  $\mathcal{D}_{train}^{(\mathcal{T}_1)}$  is the training dataset of task  $\mathcal{T}_1$ , and  $\mathbf{W}_c^{(\mathcal{T}_1)}$  are the weights when model converges. After that, our model transitions learning task from  $\mathcal{T}_1$  to  $\mathcal{T}_2$  by loading the corresponding task prompt  $\mathbf{P}^{(\mathcal{T}_2)}$  and dataset  $\mathcal{D}_{train}^{(\mathcal{T}_2)}$ . This transition involves a critical step where the evolution behavior of model weights  $\mathbf{W}$  are carefully stored,

$$\text{Train}(\mathcal{M}(\mathcal{D}_{train}^{(\mathcal{T}_2)}; \mathbf{W}_c^{(\mathcal{T}_1)})) \xrightarrow[\text{each epoch}]{\text{for}} \mathbf{W}_0^{(\mathcal{T}_2)}, \mathbf{W}_1^{(\mathcal{T}_2)}, \dots, \mathbf{W}_c^{(\mathcal{T}_2)} \quad (10)$$

To capture common patterns, we reflect the task-level stability and variations by the evolution behavior of weights in  $\mathcal{M}$ , i.e.,  $\mathcal{W}^{(\mathcal{T}_2)} = \{\mathbf{W}_c^{(\mathcal{T}_1)}, \mathbf{W}_0^{(\mathcal{T}_2)}, \mathbf{W}_1^{(\mathcal{T}_2)}, \dots, \mathbf{W}_c^{(\mathcal{T}_2)}\}$ . The weight set  $\mathcal{W}^{(\mathcal{T}_2)}$  is deliberately incorporated with finalized weight of task  $\mathcal{T}_1$  and evolution of  $\mathcal{T}_2$ , which explicitly captures the weight transition between tasks. We then introduce a collective variance  $\sigma$  to capture such stability, and employ a threshold  $\delta$  to disentangle the stable and dynamic weights along the learning process, i.e.,

$$\mathbf{W}_{\text{stable}}^{(\mathcal{T}_2)}, \mathbf{W}_{\text{dynamic}}^{(\mathcal{T}_2)} = \{w_{ij} \in \mathcal{W}^{(\mathcal{T}_2)} : \text{Var}(w_{ij}) < \delta\}, \{w_{ij} \in \mathcal{W}^{(\mathcal{T}_2)} : \text{Var}(w_{ij}) \geq \delta\} \quad (11)$$

where  $\text{Var}(w_{ij})$  represents the element-wise variance of the across different training iterations from  $\mathbf{W}_c^{(\mathcal{T}_1)}$  to  $\mathbf{W}_c^{(\mathcal{T}_2)}$ , indicating the fluctuation degree of the weight values. A lower variance indicates higher stability, suggesting minimal change in weights across updates. After that, stable weights  $\mathbf{W}_{\text{stable}}^{(\mathcal{T}_2)}$  are then frozen, and the model transitions to the next task,  $\mathcal{T}_3$ , using the stabilized weights as the initialization for further training,

$$\text{Train}(\mathcal{M}(\mathcal{D}_{\text{train}}^{(\mathcal{T}_3)}; \mathbf{W}_c^{(\mathcal{T}_2)}.\text{frozen}(\mathbf{W}_{\text{stable}}^{(\mathcal{T}_2)}))) \xrightarrow[\text{each epoch}]{\text{for}} \mathbf{W}_0^{(\mathcal{T}_3)}, \mathbf{W}_1^{(\mathcal{T}_3)}, \dots, \mathbf{W}_c^{(\mathcal{T}_3)} \quad (12)$$

Similar to Equation 12, this process is repeated until the completion of task  $\mathcal{T}_k$ , resulting in the collection  $\mathcal{W}^{(\mathcal{T}_k)} = \{\mathbf{W}_c^{(\mathcal{T}_{k-1})}, \mathbf{W}_0^{(\mathcal{T}_k)}, \mathbf{W}_1^{(\mathcal{T}_k)}, \dots, \mathbf{W}_c^{(\mathcal{T}_k)}\}$ . The Equation 11 is similarly implemented to derive  $\mathbf{W}^{(\mathcal{T}_k)} = \mathbf{W}_{\text{stable}}^{(\mathcal{T}_k)} \parallel \mathbf{W}_{\text{dynamic}}^{(\mathcal{T}_k)}$ . Since  $\mathcal{T}_1$  is not involved with the commonality extraction, we subsequently load  $\mathcal{T}_1$  with  $\mathbf{W}^{(\mathcal{T}_k)}.\text{frozen}(\mathbf{W}_{\text{stable}}^{(\mathcal{T}_k)})$  to achieve the complete rolling process. The stabilized weights across continuous tasks  $\mathbf{W}'^{(\mathcal{T}_1)} = \mathbf{W}'_{\text{stable}}^{(\mathcal{T}_1)} \parallel \mathbf{W}'_{\text{dynamic}}^{(\mathcal{T}_1)}$ , serves as  $\mathbf{W}^*$  can ultimately result in a robust multi-task learning parameters encapsulated with well-extracted common patterns via iterative stable weight selection, and it can also be served a collective intelligence by exploiting multiple tasks, thus enhancing the generalization for subsequent learning.

**Task-specific refinement phase.** This phase aims to merge the gap between task-level commonality and specificity, where the process continuously train the model from the prior phase to the next one, as  $\mathbf{W}_{\text{stable}}^*$  is frozen to maintain overall model stability and remaining weights are iteratively update with task-specific prompts. This process allows CMuST to insert the individual intelligence into integrated model and adjust itself to better suit the unique pattern of each task, i.e.,

$$\mathcal{M}^{(\mathcal{T}_i)} = \text{FineTuning}(\mathcal{M}, \mathbf{W}^*, P^{(\mathcal{T}_i)}) \quad (13)$$

where  $\mathcal{M}^{(\mathcal{T}_i)}$  is tuned to maximize performance on the task at hand, which represents the task-specific submodel after refinement. The FineTuning denotes adjustment process of model parameters.

**Summary.** Our RoAda not only ensures the preservation of commonalities across tasks, but each model is also optimally tuned for its respective task with compressed task-level patterns, providing CMuST with opportunity to achieve peak performance with both collective and individual intelligence. More details of our methodology can be found in Appendix.

## 5 Experiment

### 5.1 Datasets and Experiment Setup

**Data description.** Given the emerging multi-task ST learning, we collect and process three real-world datasets for evaluation: 1) **NYC**<sup>3</sup>: Includes three months of crowd flow and taxi hailing from Manhattan and its surrounding areas in New York City, encompassing four tasks: *Crowd In*, *Crowd Out*, *Taxi Pick*, and *Taxi Drop*. 2) **SIP**: Contains records of *Traffic Flow* and *Traffic Speed* within Suzhou Industrial Park over a period of three months. 3) **Chicago**<sup>4</sup>: Comprises of traffic data collected in the second half of 2023 from Chicago, including three tasks: *Taxi Pick*, *Taxi Drop*, and *Risk*.

**Baselines and metrics.** Our CMuST model is evaluated on a widely-used baselines spatiotemporal prediction, including RNN-based models (**DCRNN** [19], **AGCRN** [4]), STGNNs (**GWNET** [40], **STGCN** [44]) *for single task*, and **MTGNN** [39], **STEP** [33] **PromptST** [48] *for multiple tasks*.

<sup>3</sup><https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

<sup>4</sup><https://data.cityofchicago.org/browse>

Table 1: Performance comparison on three datasets. Best results are **bold** and the second best are underlined.

Datasets		NYC				SIP		Chicago		
Methods	Metrics	Crowd In	Crowd Out	Taxi Pick	Taxi Drop	Traffic Flow	Traffic Speed	Taxi Pick	Taxi Drop	Risk
DCRNN	MAE	17.5289	19.5667	10.8188	9.6142	12.5326	0.7044	3.0624	2.5793	1.1174
	MAPE	0.5939	0.5695	0.4330	0.4818	0.2455	0.2686	0.4237	0.4816	0.2504
AGCRN	MAE	11.5135	13.1569	7.0675	6.0066	15.8319	0.6924	2.3542	2.0884	1.1183
	MAPE	0.5094	0.4773	0.3753	0.3665	0.2926	0.2744	0.4092	0.4046	0.2505
GWNEN	MAE	11.4420	13.2992	7.0701	6.1171	13.0529	0.6900	2.3671	2.0434	1.1197
	MAPE	0.4778	0.6171	0.3713	0.3514	0.2483	0.2655	0.3912	0.4044	0.2514
STGCN	MAE	11.3766	13.3522	7.1259	5.9268	15.3501	0.7111	2.3781	2.1427	1.1184
	MAPE	0.5018	0.4318	0.3234	<b>0.3339</b>	0.3041	0.2660	0.4074	0.4331	0.2507
GMAN	MAE	11.3414	<u>13.1923</u>	<u>7.0662</u>	6.0912	13.0368	0.6952	2.3663	2.0316	1.1182
	MAPE	0.4782	0.6065	0.3652	0.3468	0.2464	0.2678	0.3953	0.4036	0.2516
ASTGCN	MAE	14.2847	17.1582	9.1430	7.7063	16.4896	0.6980	2.5091	2.1520	1.1175
	MAPE	0.6396	0.5922	0.4607	0.4524	0.3104	0.2682	0.4593	0.4413	<b>0.2502</b>
STTN	MAE	12.1994	14.1966	7.6716	6.3816	15.1751	0.6939	<b>2.2996</b>	2.0355	1.1214
	MAPE	0.4757	0.4744	0.3600	0.3763	0.2881	0.2625	<u>0.3893</u>	0.4133	0.2518
MTGNN	MAE	11.4350	13.3072	7.0736	6.1162	13.0486	0.6989	2.3692	2.0361	1.1201
	MAPE	0.4785	0.6185	0.3782	0.3502	0.2475	0.2687	0.3979	0.4073	0.2578
STEP	MAE	11.2328	13.1043	6.9619	5.9101	12.0032	0.6970	2.3592	2.0168	1.1190
	MAPE	0.4537	0.4361	0.3248	0.3379	0.2391	0.2638	0.3914	0.4019	0.2507
PromptST	MAE	<b>11.0036</b>	<u>13.0237</u>	<u>6.8711</u>	<u>5.8797</u>	<u>11.8620</u>	0.6921	2.3576	2.0065	1.1186
	MAPE	<u>0.4465</u>	0.4358	0.3265	0.3382	<u>0.2375</u>	0.2632	0.3913	<u>0.4012</u>	0.2511
CMuST	MAE	<u>11.1533</u>	<b>12.9088</b>	<b>6.7581</b>	<b>5.8546</b>	<b>11.5811</b>	<b>0.6843</b>	<u>2.3264</u>	<b>2.0034</b>	<b>1.1172</b>
	MAPE	<b>0.4384</b>	<b>0.4265</b>	<b>0.3118</b>	<u>0.3375</u>	<b>0.2279</b>	<b>0.2585</b>	<b>0.3872</b>	<b>0.4009</b>	0.2503

as well as *attention-based models* of (GMAN [50], ASTGCN [9], STTN [42]). The performance metrics are mean absolute error (MAE), and mean absolute percentage error (MAPE), where lower values indicate higher predictive performance.

**Implementation details.** We partitioned datasets into training, validation, and testing sets with 7:1:2 ratio. CMuST forecasts observations of next 12 time steps based on previous 12 steps, as  $T = T' = 12$ . All data were normalized to zero mean and unit variance, and predictions were denormalized to normal values for evaluation. For the MSTI, embedding dimensions were  $d_{obs} = 24$ ,  $d_s = 12$ ,  $d_t = 60$ , and the prompt dimension was 72. Dimensions for self-attention and cross-attention respectively were 168 and 24, with each attention having 4 heads and FFN’s hidden dimension was 256. The Adam optimizer is adopted with an initialized learning rate of  $1 \times 10^{-3}$ , and weight decay of  $3 \times 10^{-4}$ , where the early-stop was applied. For RoAda, the threshold  $\delta = 10^{-6}$ . Our model was implemented with PyTorch on a Linux system equipped with Tesla V100 16GB.

## 5.2 Performance Comparison

1) **Performance comparison among baselines.** In Table 1, we compared the predictive performance of our CMuST with other methods across various city datasets. Obviously, our CMuST significantly outperforms other baselines on all datasets across most of metrics and the multitask-based methods improve performance by an average of 8.51% over the singletask-based methods. This result underscores the effectiveness of the cross-attention mechanism for decoupling multidimensional dependencies, which not only enhances spatiotemporal representational capacity, but also enables easy extraction of common correlations among tasks and thus empowering each individual task

Table 2: Performance against data sparsity.

model	NYC for Crowd In							
	25% nodes		50% nodes		2 times interval		4 times interval	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
GWNEN	13.7648	0.4825	12.4637	0.4731	20.2547	0.4465	20.6487	0.4958
STEP	13.1827	0.4772	12.2393	0.4612	20.1936	0.4436	20.1465	0.4915
PromptST	12.8362	0.4719	12.0361	0.4607	19.8465	0.4384	19.5238	0.4872
CMuST	<b>12.1611</b>	<b>0.4506</b>	<b>11.2864</b>	<b>0.4470</b>	<b>18.2925</b>	<b>0.4279</b>	<b>18.4084</b>	<b>0.4797</b>

to benefit from well-extracted common patterns. 2) **Robustness in data-scarce scenarios within multi-task framework.** We constructed scenarios of data scarcity for specific tasks, i.e., we reduce part of the spatial nodes in prediction of *Crowd In*, and also reduce number of samples in temporal dimension by expanding time intervals on NYC, to study the robustness of CMuST under the challenging scenario of limited (reduced) data. Results shown in Table 2 indicate that assimilating common information from other tasks can help better prediction in a single task even though it is with limited data on either spatial and temporal dimension. This demonstrates that multi-task prediction relax the requirement of individual tasks on both data volumes and distributions, where the shared commonality effectively captures and delivers the consistency and diversity among tasks.



### 5.3 Ablation Study

To assess the effectiveness of each module in CMuST and its capability for multi-task learning, we designed a set of variants as 1) **w/o context-data interaction**: remove the spatial-context and temporal-context cross-interactions in the MSTI module, 2) **w/o consistency maintainer**: omit the separation and freezing of stable weights during the RoAda phase, instead using all weights for rolling training, 3) **w/o task-specific preserver**: eliminate the task-specific prompts, thus removing the task-specific diversity preservation. Figure 3 supports our hypothesis that CMuST is a cohesive and integral system, where the results of "w/o context-data interaction" deteriorate, potentially indicating multidimensional interactions from context environments to spatial relations and temporal evolution exactly make sense for prediction. The results on "w/o consistency maintainer" highlights the validity of capturing task consistency and commonality to facilitate task learning. Lastly, performances of "w/o task-specific preserver" show that removing unique patterns distinctive from other tasks makes inferior results.

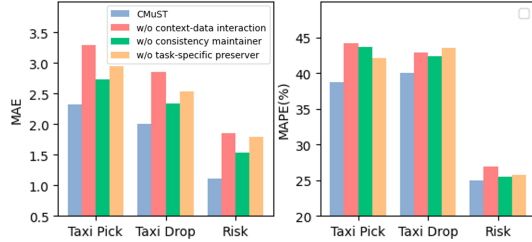
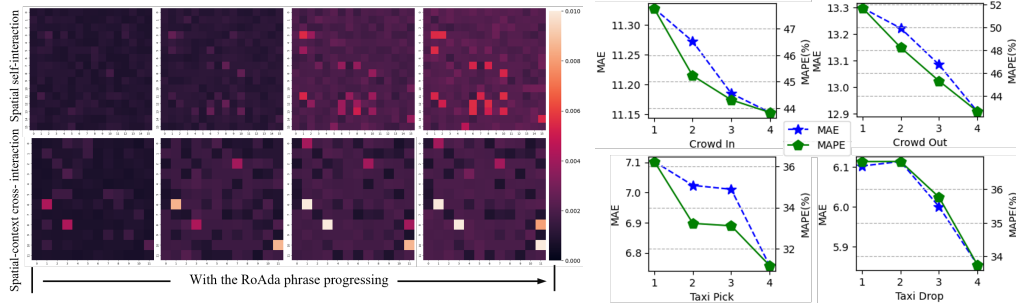


Figure 3: Ablation studies of CMuST on Chicago.

### 5.4 Case Study

1) **Visualizing attention across training phases.** In Figure 4(a), we visualize the changes of attention weights of CMuST during the stage of RoAda. It is observed that as tasks are learning continuously, the relationships and interactions across various dimensions is becoming distinctive and going stable, demonstrating the consolidation process of dimension-level relations. By modeling weight behavior, such consolidated relations and interactions between context and observations can further enable the extraction of consistency in spatiotemporal interactions across tasks. 2) **Performance variation along with task increasing.** Figure 4(b) shows the performance of individual tasks on NYC as the number of tasks increases. The performance of each task is improving with the addition of more tasks, which indicates that tasks are no longer isolated but gain the collective intelligence via assimilating common representations and interactive information.



(a) Visualizing attention across training phases.

(b) Performance variation along with task increasing.

Figure 4: Case studies of our proposed CMuST on NYC.

### 5.5 Parameter sensitivity analysis

We varied the dimension of the task prompt  $d_p$  as  $\{18, 36, 72, 144\}$ , the number of attention heads in MSTI as  $\{1, 2, 4, 8, 16\}$ , and the threshold  $\delta$  for RoAda among  $\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$ . Results shown in Figure 6 indicate that the optimal settings were  $d_p = 72$ ,  $head = 4$ , and  $\delta = 10^{-6}$ . Details of parameter analysis and fine-tuning can be found in our Appendix C.6.

## 6 Conclusion

In this work, we enable spatiotemporal learning to get rid of isolation with proposed CMuST, which consists of two major components. In CMuST, the MSTI is devised to dissect complex multi-dimension data correlations, to reveal disentangled patterns. To extract the task-wise consistency

and task-specific diversity, we propose a rolling learning scheme RoAda, where it simultaneously models weight behavior to enable collective intelligence, and constructs task-specific prompts by compressing domain data with AutoEncoding to empower task-specific refinement for enhancement. We believe our CMuST can not only help better understand the collective regularity and intelligence in urban systems, but significantly reduce repeated training and improve data exploitation, which is progressively approaching green computing in future cities. For future work, we will further investigate the collective intelligence in open urban systems, which can potentially generalize to wider domains such as energy and environment for human-centered computing.

## 7 Acknowledgement

This paper is partially supported by the National Natural Science Foundation of China (No.12227901, No.62072427), Natural Science Foundation of Jiangsu Province (BK.20240460), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No.YSBR-005), and the grant from State Key Laboratory of Resources and Environmental Information System.

## References

- [1] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. *arXiv preprint arXiv:2201.12604*, 2022.
- [2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine learning*, 73:243–272, 2008.
- [3] Muhammad Tayyab Asif, Justin Dauwels, Chong Yang Goh, Ali Oran, Esmail Fathi, Muye Xu, Menoth Mohan Dhanya, Nikola Mitrovic, and Patrick Jaillet. Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):794–804, 2013.
- [4] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting, 2020.
- [5] Xu Chen, Junshan Wang, and Kunqing Xie. Trafficstream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning. *arXiv preprint arXiv:2106.06273*, 2021.
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [7] Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz H Elibol. A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, 7:141627–141632, 2019.
- [8] Kan Guo, Yongli Hu, Yanfeng Sun, Sean Qian, Junbin Gao, and Baocai Yin. Hierarchical graph convolution network for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 151–159, 2021.
- [9] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929, 2019.
- [10] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5415–5428, 2021.
- [11] Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, and Yonghuai Liu. Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24:16453–16482, 2020.

- [12] Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12608–12616, 2024.
- [13] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossggn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems*, 36:46885–46902, 2023.
- [14] Qihe Huang, Zhengyang Zhou, Kuo Yang, Gengyu Lin, Zhongchao Yi, and Yang Wang. Leret: Language-empowered retentive network for time series forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 2024.
- [15] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4365–4373, 2023.
- [16] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 521–528, 2011.
- [17] Po-Nien Kung, Sheng-Siang Yin, Yi-Cheng Chen, Tse-Hsuan Yang, and Yun-Nung Chen. Efficient multi-task auxiliary learning: selecting auxiliary data by feature similarity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 416–428, 2021.
- [18] Hyunwook Lee and Sungahn Ko. Testam: a time-enhanced spatio-temporal attention model with mixture of experts. *arXiv preprint arXiv:2403.02600*, 2024.
- [19] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.
- [20] Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. In *Forty-first International Conference on Machine Learning*, 2024.
- [21] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4125–4129, 2023.
- [22] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. *arXiv preprint arXiv:2310.09751*, 2023.
- [23] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. In *Advances in Neural Information Processing Systems*, 2023.
- [24] Aurelie C Lozano and Grzegorz Swirszcz. Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Conference on Machine Learning*, pages 595–602, 2012.
- [25] Spyros Makridakis and Michele Hibon. Arma models and the box-jenkins methodology. *Journal of forecasting*, 16(3):147–163, 1997.
- [26] Gregory P Meyer. An alternative probabilistic interpretation of the huber loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5261–5269, 2021.
- [27] Guillaume Obozinski, Ben Taskar, and Michael I Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20:231–252, 2010.

- [28] Zheyi Pan, Zhaoyuan Wang, Weifeng Wang, Yong Yu, Junbo Zhang, and Yu Zheng. Matrix factorization for spatio-temporal neural networks with applications to urban flow prediction. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2683–2691, 2019.
- [29] German I Parisi, Jun Tani, Cornelius Weber, and Stefan Wermter. Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization. *Frontiers in neurorobotics*, 12:78, 2018.
- [30] Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465–3489, 2010.
- [31] Fynn Schröder and Chris Biemann. Estimating the influence of auxiliary tasks for multi-task learning of sequence tagging tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2971–2985, 2020.
- [32] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.
- [33] 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*, pages 1567–1577, 2022.
- [34] Xiaoming Shi, Heng Qi, Yanming Shen, Genze Wu, and Baocai Yin. A spatial-temporal attention approach for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(8):4909–4918, 2020.
- [35] Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2223–2232, 2023.
- [36] Kun Wang, Guohao Li, Shilong Wang, Guibin Zhang, Kai Wang, Yang You, Xiaojiang Peng, Yuxuan Liang, and Yang Wang. The snowflake hypothesis: Training deep gnn with one node one receptive field. *arXiv preprint arXiv:2308.10051*, 2023.
- [37] Kun Wang, Hao Wu, Yifan Duan, Guibin Zhang, Kai Wang, Xiaojiang Peng, Yu Zheng, Yuxuan Liang, and Yang Wang. Nuwadynamics: Discovering and updating in causal spatio-temporal modeling. In *The Twelfth International Conference on Learning Representations*, 2024.
- [38] Kun Wang, Hao Wu, Guibin Zhang, Junfeng Fang, Yuxuan Liang, Yuankai Wu, Roger Zimmermann, and Yang Wang. Modeling spatio-temporal dynamical systems with neural discrete learning and levels-of-experts. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [39] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Xiaojun Chang. Connecting the dots: Multivariate time series forecasting with graph neural networks, 2020.
- [40] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.
- [41] Yi Xin, Junlong Du, Qiang Wang, Ke Yan, and Shouhong Ding. Mmap: Multi-modal alignment prompt for cross-domain multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16076–16084, 2024.
- [42] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.
- [43] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, and Hui Xiong. Coupled layer-wise graph convolution for transportation demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4617–4625, 2021.

- [44] 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*. International Joint Conferences on Artificial Intelligence Organization, 2018.
- [45] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. Unist: A prompt-empowered universal model for urban spatio-temporal prediction. *arXiv preprint arXiv:2402.11838*, 2024.
- [46] Mingyang Zhang, Yong Li, Funing Sun, Diansheng Guo, and Pan Hui. Adaptive spatio-temporal convolutional network for traffic prediction. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1475–1480. IEEE, 2021.
- [47] Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012.
- [48] Zijian Zhang, Xiangyu Zhao, Qidong Liu, Chunxu Zhang, and Qian Ma. Promptst: Prompt-enhanced spatio-temporal multi-attribute prediction, 2023.
- [49] Jie Zhao, Chao Chen, Chengwu Liao, Hongyu Huang, Jie Ma, Huayan Pu, Jun Luo, Tao Zhu, and Shilong Wang. 2f-tp: Learning flexible spatiotemporal dependency for flexible traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):15379–15391, 2022.
- [50] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction, 2019.
- [51] Zheng yang Zhou, Hao Liu, Kun Wang, Peng kun Wang, Xu Wang, and Yang Wang. A teacher-student spatiotemporal semi-supervised method for urban event forecasting (in chinese). *Acta Electronica Sinica*, 51(12):3557–3571, 2023.
- [52] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Hengchang Liu. Riskoracle: A minute-level citywide traffic accident forecasting framework. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1258–1265, 2020.
- [53] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Chaochao Zhu. Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3786–3799, 2020.

## A Explanation of Relevant Concepts

### A.1 The Concept, Definition and Scope of Multi-task Learning

Actually, in our study, various domains correspond to different urban elements collected with different manners in a given city. For instance, in an integrated urban system, it includes taxi demands, crowd flow, traffic speed and accidents. We collect and organize various domain data (urban elements) in a city into one integrated dataset. The goal of our work is to explore the integrated intelligence from various domains and enhance learning of each individual urban element. To this end, the concept of multi-task here is to forecast various elements from different domains in an integrated model. Therefore, our work does not target at unifying regression or classification problems, but proposes an integrated model to iteratively establish the common intelligence among different elements and improve generalization for each element learning in succession, thus getting rid of task isolation. Noted that our experiments are performed with regression tasks, but it can easily generalize to classification task with shared representations.

### A.2 Continuous & Continual learning

In this work, 'continuous' is equivalent to 'continual'. The uniqueness of our work refers to a novel continuous task learning in ST community, which collects the integrated intelligence and benefits each individual learning task.

## B Methodology Details

### B.1 Illustration of Data Representation and Integration

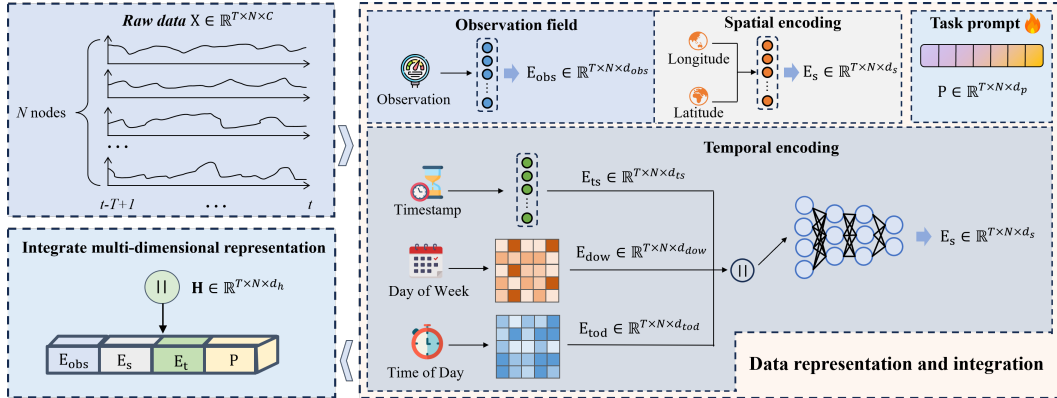


Figure 5: This figure illustrates the process of Data Representation and Integration within the CMuST framework.  $T$  and  $N$  denote the length of the time series and the number of geographic nodes, respectively. The diagram on the right details the encoding of information across different dimensions, with the bottom right corner aggregating the results of all encodings into a unified multidimensional representation.

### B.2 Positional Encoding for Temporal-context Cross-interaction

Given the  $D$  dimensions in total, we define the positional encoding as,

$$\begin{cases} \text{pos}_{(t,2d)} = \sin(t/10000^{2d/D}) \\ \text{pos}_{(t,2d+1)} = \cos(t/10000^{2d/D}) \end{cases} \quad \text{for } d = 0, \dots, D/2 - 1 \quad (14)$$

where  $t$  and  $d$  are the indexes of time slots and feature dimensions, which alternately uses sine and cosine to provide a unique representation for each time slot.

### B.3 Fusion & Regression

Predictions for future steps are influenced by representations affected by multi-dimensional interactions, where the impact of each dimension may vary. Therefore, we integrate information across various aspects effectively by a parameter matrix-based fusion strategy,

$$\mathbf{Z} = \mathbf{W}_o * \tilde{\mathbf{H}}[\dots, \text{slice}^{(o)}] + \mathbf{W}_s * \tilde{\mathbf{H}}[\dots, \text{slice}^{(s)}] + \mathbf{W}_t * \tilde{\mathbf{H}}[\dots, \text{slice}^{(t)}] \quad (15)$$

where  $*$  denotes the convolution operation, employing  $1 \times 1$  convolution kernels.  $\mathbf{W}_o$ ,  $\mathbf{W}_s$ , and  $\mathbf{W}_t$  are the parameters within these kernels, tailored to adjust the influence of observational data, spatial locations, and temporal information on the prediction targets, respectively. We further integrate the task-specific prompt to culminate in the final prediction  $\hat{\mathbf{Y}} = \text{FC}_y(\mathbf{Z}\mathbf{W}_z + \tilde{\mathbf{H}}[\dots, \text{slice}_p]\mathbf{W}_p; \theta_y)$ , the Huber Loss [26] is utilized as the optimization function, which is less sensitive to outliers compared to squared error loss, defined as:

$$\mathcal{L}_y = \begin{cases} \frac{1}{2}(\hat{\mathbf{Y}} - \mathbf{Y})^2, & \text{if } |\hat{\mathbf{Y}} - \mathbf{Y}| < \delta \\ \delta(|\hat{\mathbf{Y}} - \mathbf{Y}| - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \quad (16)$$

where  $\mathbf{Y}$  is the ground truth,  $\delta$  controls the sensitivity to outliers, which enhances the accuracy by integrating diverse data dimensions and improves robustness against data variability and anomalies.

### B.4 Details of Rolling Adaptation

The detailed algorithmic procedure for the RoAda phase can be found in Algorithm 1.

## C Additional Experiment Details

### C.1 Dataset Details

We provide detailed information about the dataset, including the number of records in the original data, the number of regions into which the data was divided, and the time intervals. These details are presented in Table 3. The specific data preprocessing is as follows:

**NYC:** We collect yellow taxi trip data from January to March 2016 from the NYC Open Data website. Each trip record includes information such as pickup and dropout times, locations, and the number of passengers. We filter out records with abnormal longitude and latitude values or missing data. Then we select data within Manhattan and surrounding areas, divided into  $30 \times 15$  grids, and counted trips per grid, selecting those with total trips greater than or equal to 1000, resulting in 206 grids. Each grid’s data is aggregated into 30-minute intervals, yielding taxi pickup counts, taxi dropout counts, and crowd in/out flows. We also include time of day (tod) and day of week (dow) as context, resulting in four tasks with input features [value, tod, dow].

**SIP:** We collect traffic data from Suzhou Industrial Park from January to March 2017, comprising tens of thousands of records. The area is divided into nodes, and data is aggregated into 5-minute intervals. After filtering out grids with sparse data, we obtain 108 nodes, each containing traffic speed and traffic flow. We include time of day and day of week as input context, resulting in two tasks: traffic flow and traffic speed, with input [value, tod, dow].

**Chicago:** We collect taxi trip and accident data from the Chicago Open Data platform for June to December 2023. The taxi data includes trip start, end times and locations. We divide the area into  $30 \times 20$  grids and select grids with total trips greater than 100, resulting in 220 grids. Similar to the NYC dataset, data is aggregated into 30-minute intervals, yielding taxi pickup and dropout counts, resulting in two tasks with input features [value, tod, dow]. The accident data includes incident locations, times, casualty numbers, and injury severity of each casualty. We then obtain the risk score by weighting it according to each casualty and injury, mapped it to the 220 grids, and aggregated the risk score over time intervals, resulting in a risk task with input features [risk score, tod, dow].

### C.2 Implementation Details and Fairness-aware Experimental Evaluation

To verify whether the multi-task learning in urban systems can compete single-task learning scheme and further show the superiority of our continuous multi-task learning, our experiments are designed

---

**Algorithm 1** Rolling Adaptation Process

---

**Require:** Model  $\mathcal{M}$ , tasks  $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k\}$ , prompts  $\{\mathbf{P}^{(\mathcal{T}_1)}, \mathbf{P}^{(\mathcal{T}_2)}, \dots, \mathbf{P}^{(\mathcal{T}_k)}\}$ , train data  $\{\mathcal{D}_{\text{train}}^{(\mathcal{T}_1)}, \mathcal{D}_{\text{train}}^{(\mathcal{T}_2)}, \dots, \mathcal{D}_{\text{train}}^{(\mathcal{T}_k)}\}$ , learning rate  $\gamma$ , threshold  $\delta$

- 1: **Warm-up Phase:**
- 2:  $\triangleright$  For all parameters in  $\mathcal{M}$ , the parameters corresponding to their respective names  $\triangleleft$
- 3: **Initialize:**  $\mathcal{W}_{\text{histories}} \leftarrow \{\}$   $\triangleright$  Dictionary with keys as names and values as parameters list
- 4: **for** each task  $\mathcal{T}_i$  in  $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k\}$  **do**
- 5:     **if**  $i == 1$  **then**
- 6:          $\triangleright$  Train task 1 for initializing model  $\triangleleft$
- 7:         Load prompt for task  $\mathcal{T}_1$ :  $\mathcal{M} \leftarrow \mathbf{P}^{(\mathcal{T}_1)}$
- 8:         Set learning rate as  $\gamma$
- 9:         Train  $\mathcal{M}$  on  $\mathcal{D}_{\text{train}}^{(\mathcal{T}_1)}$  until convergence to obtain  $\mathbf{W}_c^{(\mathcal{T}_1)}$
- 10:         Store weights:  $\mathcal{W}_{\text{histories}} \xleftarrow[\text{name:parameters}]{\text{append}} \mathbf{W}_c^{(\mathcal{T}_1)}$
- 11:     **else**
- 12:          $\triangleright$  Rolling training for task 2 to k  $\triangleleft$
- 13:         Load prompt for task  $\mathcal{T}_i$ :  $\mathcal{M} \leftarrow \mathbf{P}^{(\mathcal{T}_i)}$
- 14:         Set learning rate as  $\gamma \times 0.01$   $\triangleright$  Prevent catastrophic forgetting during rolling
- 15:         Train  $\mathcal{M}$  on  $\mathcal{D}_{\text{train}}^{(\mathcal{T}_i)}$  with initial weights  $\mathbf{W}_c^{(\mathcal{T}_{i-1})}$
- 16:         **for** each epoch  $j$  **do**
- 17:             Record weights:  $\mathcal{W}_{\text{histories}} \xleftarrow[\text{name:parameters}]{\text{append}} \mathbf{W}_j^{(\mathcal{T}_i)}$
- 18:         After training, obtain the weights of  $\mathcal{M}$  as  $\mathbf{W}_c^{(\mathcal{T}_i)}$
- 19:          $\triangleright$  Calculate variances and freeze stable weights  $\triangleleft$
- 20:         **for** each parameters' name in  $\mathcal{M}$  **do**
- 21:              $v \leftarrow \text{CALCULATE\_VARIANCE}(\mathcal{W}_{\text{histories}}[\text{name}])$
- 22:             **if**  $v < \delta$  **then**
- 23:                 Freeze the parameter corresponding to the name
- 24:         Reset  $\mathcal{W}_{\text{histories}}$  with  $\mathbf{W}_c^{(\mathcal{T}_i)}$
- 25:          $\triangleright$  Train task 1 again since it's not involved with the commonality extraction  $\triangleleft$
- 26:          $\mathcal{M} \leftarrow \mathbf{P}^{(\mathcal{T}_1)}$
- 27:         The remaining steps are similar to lines 14-22
- 28:         Finally, the weights of  $\mathcal{M}$  are saved as  $\mathbf{W}^*$
- 29:     **Task-Specific Refinement Phase:**
- 30:     **for** each task  $\mathcal{T}_i$  in  $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k\}$  **do**
- 31:         Load weights and prompt  $\mathcal{M} \leftarrow \mathbf{W}^*$ ,  $\mathcal{M} \leftarrow \mathbf{P}^{(\mathcal{T}_i)}$
- 32:         Set learning rate as  $\gamma$
- 33:         Fine-tune  $\mathcal{M}$  on  $\mathcal{D}_{\text{train}}^{(\mathcal{T}_i)}$  and save model

---

into both single-task learning and multi-task learning. We will further clarify how to ensure the fair evaluation and comparisons across different models.

For single-task learning, we take different datasets as individual ones to respectively train and test the model [23], where compared models and our CMuST are respectively and repeatedly trained for 5 times with different random seeds. The average results are reported in our Table. 1.

For multi-task learning, since existing models have not designed the task-level continuous learning scheme, we align the features of different types of data in the same urban system into a same urban graph, where data features are concatenated for training the model. The learning objectives are followed by corresponding statement of each paper. For our CMuST, we implement the continuous and iterative model update described in Sec. 4.4. The fairness is well-incorporated by ensuring the all the data input to each model is equivalent.

The experimental results find that almost all multi-task learning can compete against single-task learning, and our continuous multi-task learning is also superior to other multi-task learning without explicitly capturing the task commonality and diversity.



Table 3: The detailed information of the three datasets.

City	Task	#Records	Time Span	#Regions	#Time Steps	Time Interval
NYC	Taxi Drop	30,245k	01/01/2016- 03/31/2016	206	4368	30mins
	Taxi Pick					
	Crowd In					
	Crowd Out					
SIP	Traffic Flow	1,237k	01/01/2017-	108	25920	5mins
	Traffic Speed	307k	03/31/2017			
Chicago	Taxi Drop	3,291k	06/01/2023-	220	10272	30mins
	Taxi Pick					
	Risk					

### C.3 Avoiding Catastrophic Forgetting

To avoid catastrophic forgetting, we implement several strategies during the RoAda phase. Firstly, we set the learning rate for each task to  $1e-5$ . This helps to retain more knowledge from previous tasks and prevents the model from over-adjusting to new tasks. By maintaining a low learning rate, the model can incrementally learn new information while preserving the stability of previously learned tasks. Additionally, we use task-specific weights for each task, such as task prompts. This method allows us to absorb the common features across all tasks while independently preserving and updating task-specific parameters. This approach ensures that when learning new tasks, the model does not forget the knowledge gained from previous tasks, thereby preventing catastrophic forgetting and avoiding overfitting to new tasks.

### C.4 Comparison Experiments with Unified Models

We compare with unified spatiotemporal/time series learning, such as UniST [45] and UniTime [22], to better show the generalization ability of our model for multi-task learning in the same urban system. The results of these in Table 5.

Table 4: Explanation of table symbols of Table 5 and Table 6. The datasets are numbered by ①②③ respectively, and the specific tasks in the dataset are numbered by Roman numerals III and so on.

Dataset	Task	Notation(Dataset/task)
NYC	Crowd In	①/I
NYC	Crowd Out	①/II
NYC	Taxi Pick	①/III
NYC	Taxi Drop	①/IV
SIP	Traffic Flow	②/I
SIP	Traffic Speed	②/II
Chicago	Taxi Pick	③/I
Chicago	Taxi Drop	③/II
Chicago	Risk	③/III

Table 5: The results of unified model.

Model/Dataset	①/I	①/II	①/III	①/IV	②/I	②/II	③/I	③/II	③/III
UniST/MAE	11.3865	13.0762	6.8942	5.8804	11.7461	0.6985	2.3551	2.0134	1.1186
UniST/MAPE	0.4610	0.4478	0.3261	0.3490	0.2465	0.2661	0.3916	0.4162	0.2508
UniTime/MAE	12.2874	14.9120	7.4723	6.4641	13.9172	0.6993	2.4564	2.0341	1.1292
UniTime/MAPE	0.4721	0.4760	0.3671	0.3719	0.2965	0.2713	0.3987	0.4254	0.2511

## C.5 Experiments for Cold-start

We have designed the experiment of cold start. Specifically, for NYC dataset, we selected three of the four tasks of *Crowd In*, *Crowd Out*, *Taxi Pick* and *Taxi Drop* in turn for training, and calculated the adaptation time and results for the remaining one task on this basis, comparing with training a single task alone. A similar design is applied for SIP and Chicago datasets. The results are shown in Table 6, which show that, both in terms of effect and time, it performs better than single task, indicating that our model adapts to the newly arrived task more quickly and well, which is conducive to solving the problem of cold start of urban prediction.

Table 6: Experiments for cold start.

Type/Metric/Dataset	①/I	①/II	①/III	①/IV	②/I	②/II	③/I	③/II	③/III
Single Task/MAE	11.2457	13.1284	6.9357	6.0122	11.8684	0.6912	2.3317	2.0223	1.1175
Single Task/MAPE	0.4623	0.4782	0.3453	0.3531	0.2758	0.2613	0.3983	0.4023	0.2506
Adaptation Time (s)	2,132	1,344	1540	1,545	4,272	4,525	3,301	4,504	4,313
Cold Start/MAE	<b>11.1681</b>	<b>13.0027</b>	<b>6.8032</b>	<b>5.9834</b>	<b>11.7832</b>	<b>0.6901</b>	<b>2.3278</b>	<b>2.0089</b>	<b>1.1173</b>
Cold Start/MAPE	<b>0.4407</b>	<b>0.4323</b>	<b>0.3225</b>	<b>0.3463</b>	<b>0.2469</b>	<b>0.2598</b>	<b>0.3906</b>	<b>0.4013</b>	<b>0.2504</b>
Adaptation Time (s)	<b>1,571</b>	<b>1,150</b>	<b>1,322</b>	<b>1,441</b>	<b>3,019</b>	<b>4,053</b>	<b>2,886</b>	<b>2,736</b>	<b>4,198</b>

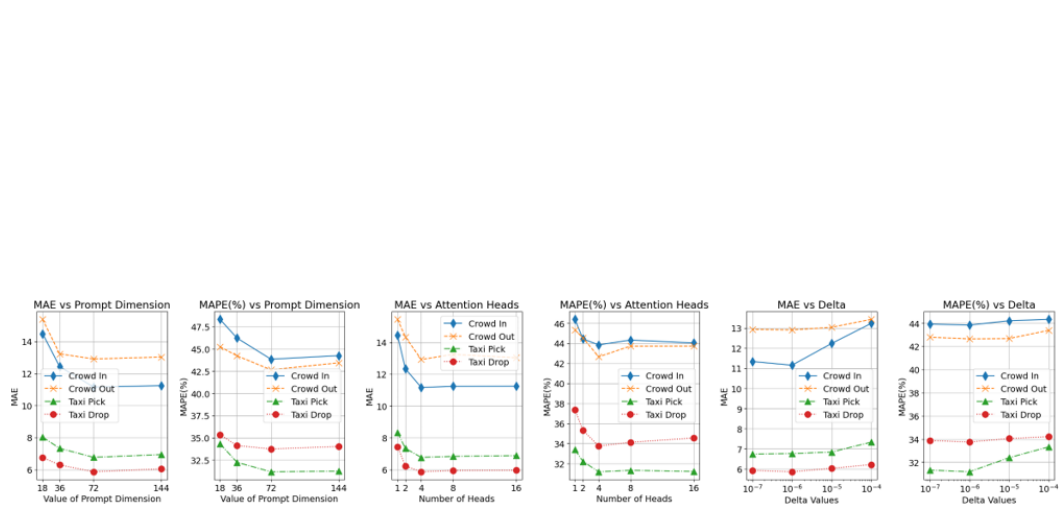
## C.6 Parameter Sensitivity Analysis

To study the impact of hyperparameters on model performance, we varied the dimension of the task prompt  $d_p$  as  $\{18, 36, 72, 144\}$ , the number of attention heads in MSTI as  $\{1, 2, 4, 8, 16\}$ , and the threshold  $\delta$  for RoAda among  $\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$ . The experimental results, displayed in Figure 6, are based on analysis of three datasets. 1) Initially, as the dimension of the prompt increases, both MAE and MAPE decrease, indicating that within a lower range, increasing  $d_p$  benefits the model by capturing more task-specific information. However, performance significantly deteriorates after  $d_p$  exceeds 72, suggesting that additional dimensions may carry redundant and non-informative elements that introduce noise and prevent the model from effectively learning useful information. 2) On the other hand, as the number of heads increases, model performance initially improves, allowing the attention mechanism to capture more diverse and dimensional information through different heads. Yet, this positive effect reverses after exceeding four heads, indicating that too many heads can lead to information overlap, reducing efficiency and adding unnecessary complexity to the model, thereby degrading performance. 3) During the rolling adaptation phase, both excessively high and low values for the  $\delta$  threshold lead to performance degradation. When  $\delta$  is set too low, it results in fewer stable weights being retained, which may fail to preserve essential information critical for learning stability. Conversely, setting  $\delta$  too high results in more weights being frozen, which restricts the model’s flexibility and adaptability, locking in potentially dynamic information and preventing the model from adapting to novel and evolving data patterns effectively.

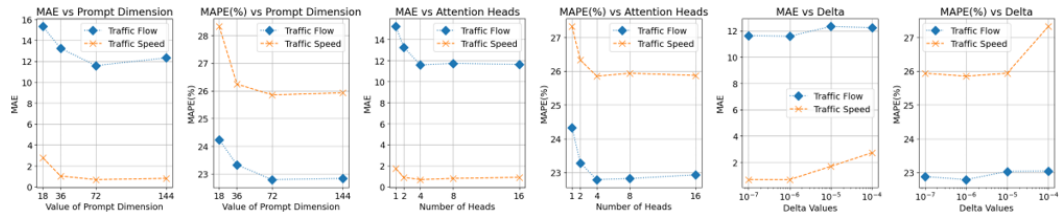
## D Others

### D.1 Limitation

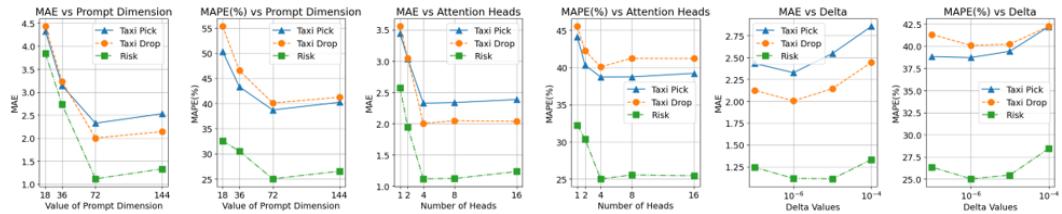
Our current work is confined to a single urban system and our experiments are solely within the transportation sector. The exploration into other domains within the same city, such as electricity, transportation, and pollution multi-task learning is still limited. Addressing these limitations will form the basis of our future research.



(a) Parameter sensitivity on NYC.



(b) Parameter sensitivity on SIP.



(c) Parameter sensitivity on Chicago.

Figure 6: Parameter sensitivity analysis of CMuST.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: In the introduction (Section 1), we summarized the contributions of this paper, which include the technical development of two modules: MSTI and RoAda, and the construction of benchmark datasets for three cities. Through extensive experimentation, we demonstrated the advantages of our CMuST framework in multi-task learning under conditions of limited data.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discussed the limitations of our paper and the areas requiring further improvement in Appendix D.1.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical assumptions and results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provided a detailed description of our model architecture, datasets, and implementation methods. At the same time, in order to facilitate reproduction, we also provided source code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We added the source code to an anonymous repository and provided the source of the dataset.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimental settings are presented in Section 5.1. Full details for fairness-aware experimental evaluation are presented in Appendix C.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For all experimental results in this paper we provided the mean of the numerous experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provided basic information about our deployment platform and computing power in Section 5.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We make sure our research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discussed the potential positive societal impacts of our paper in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: There are no such risks of the datasets in this paper.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

#### 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly cited the original papers and web pages that produced the code package and dataset.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.



- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Our code has been provided in anonymous link.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.