Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 3734557, 22 pages
https://doi.org/10.1155/2023/3734557

Research Article

Meta-Learning-Based Spatial-Temporal Adaption for Coldstart Air Pollution Prediction

Zhiyuan Wu,[1] Ning Liu,[1] Guodong Li,[2] Xinyu Liu,[3] Yue Wang,[1] and Lin Zhang[3]

[1] *Department of Electronic Engineering, Tsinghua University, Beijing, China*
[2] *Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen, China*
[3] *Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China*

Correspondence should be addressed to Lin Zhang; linzhang@tsinghua.edu.cn

Received 13 October 2022; Revised 27 May 2023; Accepted 29 May 2023; Published 16 June 2023

Academic Editor: Alexander Hošovský

Air pollution is a significant public concern worldwide, and accurate data-driven air pollution prediction is crucial for developing alerting systems and making urban decisions. As more and more cities establish their monitoring networks, there is a pressing need for coldstart model training with limited data accumulation in new cities. However, traditional spatial-temporal modeling and transfer learning schemes have been challenged under this scenario because of insufficient usage of available source data and suboptimal transferring strategy. To address these issues, we propose a meta-learning-based spatial-temporal adaptation solution for coldstart air pollution prediction. Our approach is a model-agnostic framework that enables a given backbone predictor with adaption ability across different space and time locations. Specifically, it learns a factorization of the available source data distribution and recognizes the target city as one of its components, greatly reducing the data accumulation requirement and providing coldstart capability. Furthermore, we design a novel bidirectional meta-learner that can simultaneously leverage task embeddings learned from data and features constructed based on prior knowledge. We conduct comprehensive experiments on both synthetic and real-world air pollution datasets of four distinct pollutants. The results demonstrate that our proposed method achieves a 5.2% lower 24-hour prediction mean absolute error (MAE) than pretraining and fine-tuning solutions when facing a new city with only 200 hours of data, which empirically verifies the effectiveness of our approach as a coldstart training solution.

1. Introduction

1.1. Background and Contribution. Air pollution has been a pressing global issue in recent years. The heavy air pollution not only puts public health at risk but also to a great extent constrains the city development [1]. Under this circumstance, accurate air pollution prediction has become an increasingly urgent necessity to mitigate its adverse effects. Timely government warnings and emergency actions such as traffic restrictions and road sprinkling can be undertaken to minimize the damage caused while citizens can also plan their outdoor activities accordingly. Recent advances in deep learning have sparked increasing research interest in data-driven air pollution prediction to achieve this goal [2–4].

However, most existing deep learning models require large amounts of data accumulation for model training. Consequently, their applicability is severely limited in some practical scenarios with only a few valid samples available. For instance, a newly established monitoring station in a new city may have only a few days or weeks of data, or a mobile sensing system may be unable to gather sufficient data in a constrained spatial-temporal (ST) window. In such cases, it is notoriously hard or computationally expensive to develop a customized prediction model using limited data, either from scratch or fine-tuning from pretrained one. This issue is referred to as the *coldstart* model training problem.

Take transferring to a new city as an example; traditional transfer learning solutions typically involve pretraining a deep network using a mixed dataset from source cities and fine-tuning it on the target city's data. However, this approach may suffer from overfitting, catastrophic forgetting, and negative transfer, particularly when the target data are extremely limited [5–7], thereby leaving the coldstart problem quite challenging [8, 9]. Nevertheless, pretraining and fine-tuning schemes are usually designed for general and unspecified case, without considering possible scenario-specific knowledge. For coldstart air pollution prediction problem, data inefficiency can be attributed to an insufficient usage of available source data and suboptimal transferring strategy.

> *Insufficient usage of available source data.* The pretraining stage of the general transfer learning scheme usually takes independent and identically distributed (i.i.d.) assumptions over the source data distribution and neglects potential finer distribution structures. However, it is widely acknowledged that air pollutant patterns can differ depending on the underlying physical dynamics [10], such as temporally different dominant pollution processes (e.g., diffusion of local sources, transition from remote sources, or transformation from other pollutants) and spatially different environments (e.g., industrial areas, residential areas, or park areas). Therefore, constructing samples from this consecutive spatial-temporal process can yield highly correlated samples that belong to different pattern-specific sub-distributions [11, 12]. Mixing up all samples and shuffling datasets under i.i.d. assumptions compromise the valuable source data structures and lead to performance degradation [13].

> *Sub-optimal transferring strategy.* The general transfer learning scheme assumes arbitrary source and target distribution relationships without considering possible biases. However, as a physical system, the target air pollution distribution is likely to be recognized as an existing sub-distribution in the source dataset [14]. In this case, the data accumulation requirement may be significantly reduced compared to training the target distribution directly. Furthermore, the fine-tuning result under limited target data can suffer from overfitting and be unreliable [13, 15].

> In order to address these challenges, we propose a meta-learning-based [16] spatial-temporal adaption solution for the coldstart air pollution prediction task. Instead of assuming the source dataset as i.i.d. samples from a fixed distribution, we assume that the samples within a small spatiotemporal window follow a pattern-specific sub-distribution. Accordingly, we learn a meta-model that can utilize a few nearby samples to identify and adapt to the corresponding data distribution with explicitly optimized performance across different spatial-temporal locations. The learned meta-model can then address the coldstart problem by adapting to the target spatial-temporal location even with very limited data. This formulation is illustrated in Figure 1. To more effectively learn the correlation between provided samples and target patterns, we further propose a bidirectional meta-learner that simultaneously utilizes both end-to-end learned features [17] and prior constructed features [18]. Our framework is model-architecture-agnostic and can provide spatial-temporal adaptation features and coldstart capabilities to various customized backbone networks and prediction settings. We validate our method through comprehensive experiments on both synthetic and real-world air pollution datasets.

Our contribution can be summarized as follows:

(i) We propose a novel formulation of the spatiotemporal adaptation problem for air pollution prediction, which enables the meta-model to capture diverse data patterns and provide dynamic and adaptive predictions across time and space.

(ii) We address the coldstart air pollution prediction problem by leveraging the inference procedure of the meta-model at a new spatiotemporal location. The adapted model can achieve better fit to the target pattern even with very limited available samples.

(iii) We design a bidirectional meta-learner that incorporates both learned task representation and iteratively constructed support features. This structure is empirically a simple but effective meta-learner that works well in our experiments.

This paper is organized as follows. We first briefly review the related works in Section 1.2. Then, we formalize the problem formulation and explain the proposed method in detail in Section 2. Our main experiment results and the corresponding discussion are presented in Section 3 and Section 4. We conclude this paper in Section 5.

1.2. Difference to Related Work.

Our work is closely related to two research topics, i.e., data-driven air pollution prediction models and meta-learning models.

1.2.1. Air Pollution Prediction.

Data-driven air pollution prediction problem [2] has been long discussed as a time series forecasting problem, has been considered as spatial-temporal forecasting problem when spatial typology is available [3, 19–22]. Remarkable efforts have been devoted to better temporal sequential modeling as well as spatial feature construction. On the one hand, for temporal sequential modeling, traditional statistical methods like ARIMA [23, 24], HMM [25], and linear models [21] have been considered but their performance tends to be less satisfactory due to over-simplified probabilistic assumptions over temporal models. With the fast development of machine learning techniques, especially deep learning models, automatic feature extraction learned from big data shows great potential. Deep neural networks based on sequence-to-sequence framework [26] have been widely adopted with various temporal models, including recurrent neural networks [3, 10, 27, 28], full convolutional structures [20, 29], and transformer architectures [30]. There also exists a branch of work that tries to enhance learned features with signal processing techniques like Fourier or wavelet

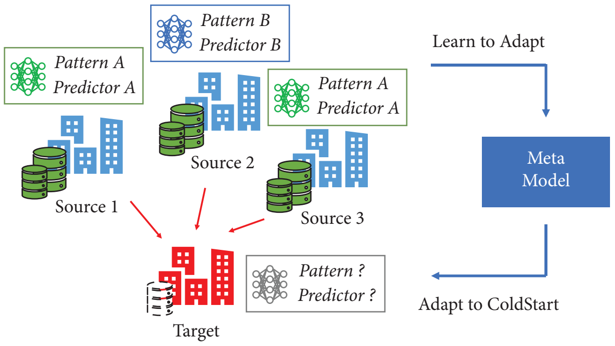


Figure 1: A spatiotemporal adaption approach for coldstart air pollution prediction.

transform [31]. On the other hand, for spatial correlation modeling, handcrafted spatial features have been widely considered [3, 21, 32]. Beyond these heuristically and carefully designed features, deep spatial features given by embedded spatial modules like spatial convolutions [28, 33, 34], graph convolution networks [20, 35–38], and transformers [39] have shown better performance. Being parallel to network architecture design, learning framework research focusing on mining fine-grained data structures and improving adaptive abilities has also been investigated [40, 41]. Among these works, transfer learning techniques have been used to handle limited data problem [42], and meta-learning-based models are designed to better process different environment conditions and/or human activities [11, 12]. Using deep learning methods to time series modeling, either with or without causal constraints as required in prediction task, has significant implications and advantages for air pollution analysis applications such as missing-value imputation and data preprocessing [43, 44].

Our work is complementary to most existing air pollution model structure designs. Our approach does not involve the design of a specific spatial-temporal neural network. Instead, our aim is to create a model-agnostic framework that facilitates the adaptation of a given backbone network across space and time. On the other hand, we focus on a more challenging scenario of coldstart model training at new city under extremely limited data and propose a new coldstart algorithm that outperforms traditional transfer learning solutions.

1.2.2. Meta-Learning. Meta-learning [16, 45], also known as learning to learn, is a learning scheme under deep learning that aims to mimic the human learner's abilities of being able to fast adapt to a new task based on a few samples/demonstrations/hints after experiencing a tremendous number of other tasks. Based on different settings on what task refers to, recently the meta-learning techniques have been successfully applied to few-shot learning [13, 18], model adaption [11, 12, 41], transfer learning [46, 47], and data compression [48]. On the one hand, the meta-learning model can be roughly categorized into three classes based on the task inference procedure. The model-based meta-learning, also known as black-box meta-learning, uses a learnable meta-model to directly generate model for target tasks [17, 49, 50]. The metric-based meta-learning, mostly designed for classification tasks, learns an embedding space and corresponding learner that can generalize across tasks [51–53]. The optimization-based meta-learning constructs the target model by explicitly solving a learnable optimization problem or learning an updating rule [18, 54–57]. On the other hand, the Bayesian meta-learning algorithms set up a probabilistic generative model to explicitly describe prior assumptions over support set and query set [58–60]. Then, specific meta-learner and their variants can be derived by choosing different distribution models and inference methods [61, 62]. The meta-learning algorithms have also been used in spatial-temporal prediction tasks, not limited to air pollution prediction, to better learn hybrid patterns across environment [12] or POI [63].

Different from previous approaches that apply meta-learning to air pollution prediction task for learning dynamic environments, our research takes a step further by utilizing this adaptation capability to address the challenging coldstart model training problem. In addition, we introduce a novel bidirectional meta-learner that can effectively leverage both black-box task embedding and optimization-based feature construction, resulting in improved adaptation performance for spatial-temporal data structures.

2. Dataset and Method

We will first describe the problem formulation and the used notations in Section 2.1. Then, we introduce the proposed spatial-temporal adaption (STA) solution in Section 2.2. In Section 2.3, we introduce the bidirectional meta-learner we designed for STA problem. In Section 2.4, we describe a novel algorithm based on STA formulation and bidirectional meta-learner for the challenging coldstart air pollution prediction task. In Section 2.5, we introduce the dataset and experiment setting that we used to validate the effectiveness of the proposed method.

2.1. Problem Formulation. The air pollution data can be sensed and stored as a series simultaneously indexed by time and space coordinates. Let the coordinate tuple (t, p) be a *spatiotemporal location* at a specific time stamp t and position p. With $c_{t,p}$ denoting the scalar concentration of some air pollutants reported by the sensor, a *sample* at (t, p) can then be defined as a regression variable pair $(x_{t,p}, y_{t,p})$:

$$\begin{aligned} x_{t,p} &:= \left\{c_{t',p}\right\}_{t-L_o \le t' \le t}, \\ y_{t,p} &:= \left\{c_{t',p}\right\}_{t<t' \le t+L_p}, \end{aligned} \tag{1}$$

where the regression input (covariate) $x_{t,p}$ is a vector of history observations and the regression output (target) $y_{t,p}$ is a vector of future trends. L_o and L_p are correspondingly observation length and required prediction length. Then the *air pollution prediction task* is the regression problem to find a *predictor* f such that $y_{t,p} = f(x_{t,p})$.

In most previous data-driven air pollution prediction solutions, a supervised training dataset $D = \left\{(x_{t_i,p_i}, y_{t_i,p_i}) \mid i = 1, \ldots, |D|\right\}$ can be constructed from history records by randomly sampling possible spatiotemporal locations (t_i, p_i). Here $|D|$ denotes the size of dataset. Then, the empirical estimation over D of some expected prediction loss l (like mean squared error) is optimized within some predictor family $\mathbb{F}$:

$$\begin{aligned} &\min_\theta \sum_{x_i, y_i \in D} [l(y_i, f_\theta(x_i))] \\ &\text{s.t. } f_\theta \in \mathbb{F}: \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_y}, \end{aligned} \tag{2}$$

where we use (x_i, y_i) to denote the sample $(x_{t_i,p_i}, y_{t_i,p_i})$ for succinctness and d_x and d_y are correspondingly the vector dimensions. In practice, $\mathbb{F}$ is usually parameterized by θ, a carefully designed deep neural network. The learned model f_{θ^*} is expected to generalize to unseen samples and have good prediction ability in the future.

Note that equation 1 is for a temporal-only predictor. Though it is usually the case that the new city does not have monitoring networks and only supports temporal prediction, equation 1 and the following derivation over it can be easily extended to more specific settings when additional information is available. For example, one can introduce readings from surrounding sensors into input vector as $x_{t,p} := \{c_{t',p'}\}_{t-L_o \leq t' \leq t, p' \in \mathcal{N}(p)}$ to include spatial correlations and build spatial-temporal predictors. $\mathcal{N}(p)$ is a set of positions that are considered as a neighborhood of p. It is also a common practice to append some other auxiliary features like weather observation, weather forecast, or traffic status as a part of $x_{t,p}$ for better prediction accuracy. For these cases, the structure of predictor family $\mathbb{F}$ has to be correspondingly modified to fit the input. However, in this paper, we will focus on developing a *model-agnostic* algorithm, i.e., a framework that can enable, in principle, any given backbone predictor the coldstart ability. Therefore, without losing generality, we will use the equation 1 through this section.

We further consider the *coldstart* prediction task. It is a common situation in practice that we need to set up a prediction model for a *target city* p_* with very limited data accumulation. In such case, we may only have days to weeks data accumulation, and the number of samples at p_* is very small. However, there exists a set of *source cities* $\{p_i\}$ where abundant samples at these locations are available. Then, the problem is how to derive a good predictor for target city by efficiently using source city data.

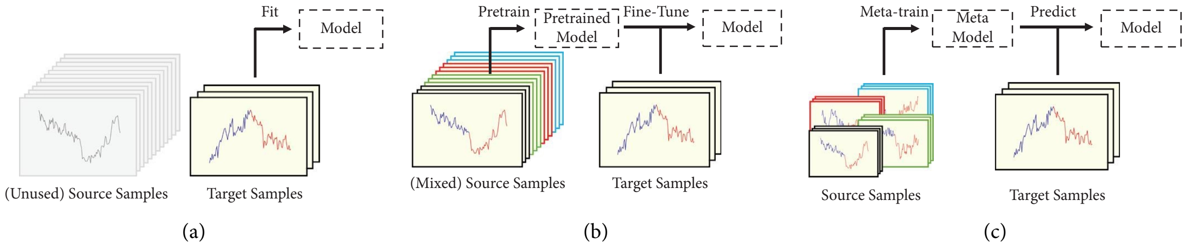
2.2. The Spatiotemporal Adaption Problem. As the samples at target city p_* are very limited, training the model from scratch on a few samples as equation (2) can be impracticable, and fine-tuning the model pretrained on source cities also suffers from issues like overfitting. In this section, we propose to treat the coldstart prediction problem as a spatiotemporal adaption (STA) problem. Instead of directly training or fine-tuning a new model for target city, we seek a meta-model on source data that learns to adjust the given backbone model to make it adapt across space and time. Consequently, the resulting meta-model can be used to generate an adaptive model for the target city as the required coldstart result. The difference between several learning schemes is illustrated in Figure 2.

We start by providing a probabilistic view of the traditional workflow in equation (2). Suppose that all samples (x_i, y_i) are drawn from an underlying true *joint distribution* $p_d(x, y)$. We will use the subscript d to denote true data distribution through this paper. Optimization objective 2 can then be considered as the expected KL divergence between the true *predictive distribution* $p_d(y \mid x)$ and an energy model surrogate defined by predictor and loss function.

This means in problem (2) the predictor is asked to directly approximate an atomic predictive distribution $p_d(y \mid x)$ across time and space. However, this may lead to data inefficiency and prediction accuracy sacrifice for the air pollution prediction task because it neglects the *fine-grained structure* of the target distribution. The fine-grained distribution structure means that the predictive distribution can have inherent spatial-temporal structure as a mixture of several time-space-dependent subdistribution. For specific underlying physical dynamics around a spatiotemporal coordinate, it is possible to give a better prediction using a localized or adaptive predictive distribution instead of the global one. Moreover, samples that are close in spatiotemporal coordinates can be strongly correlated to identify such adaptive predictive distribution and should not be set to i.i.d. as equation (2) did.

In order to address the above issues, we propose to explicitly learn the fine-grained structure of the predictive distribution to obtain an adaptive prediction model. With the assumption that the local predictive distribution is stable in a local spatial-temporal (ST) window, we learn a meta-model g that can generate a ST-specific predictor optimized for this window. More specifically, a ST window at (t, p) is defined as a set of ST locations centered at (t, p), i.e., $W_{t,p} := \{(t', p') \mid d[(t', p'), (t, p)] < d_0\}$, where we use symbols $d(\cdot,\cdot)$ and d_0 to denote a selected distance measure between two ST locations and corresponding ST window radius. It is convenient to represent the ST window by the samples over it, i.e., let $D_{t,p} = \{(x_{t',p'}, y_{t',p'}) \mid t', p' \in W_{t,p}\}$. Note that we are considering a window at the sample level (with sample (x, y) as elements) instead of at the sensory data level (with concentration reading c as elements). As result, we decompose the source dataset D into a group of mini-datasets $D = \cup D_i$, where D_i is short for D_{t_i,p_i}.

Then, we can formally define the learning objective of the adaptive predictor g. Instead of directly learning one model for the overall dataset as most previous works did, we learn a meta-model g that first generates an adapted model for a specific ST location based on some provided samples from the corresponding mini-dataset. The generated model can then be used to make predictions for new samples from the same mini-dataset. Following popular meta-learning terminology convenience, we split each D_i into two disjoint parts. Labeled samples from the first part, called support set D_i^s, will be used to identify local task characteristics and generate an adapted prediction model. Samples from the second part, called query set D_i^q, will be used to evaluate prediction loss and update the adaption strategy. Note that typically there are



(a)

(b)

(c)

Figure 2: Illustration of different learning schemes for coldstart prediction task: (a) train from scratch, (b) pretrain and fine-tune transfer, and (c) STA. Each curve stands for a sample (blue part as input and red part as label, auxiliary input omitted). The color of the curve box denotes samples from different ST windows.

only a few samples in the support set and samples in the support set are temporally anterior to the query set for causality. We can then optimize the average performance of the adapted prediction model with respect to the original meta-model within some parameterized function family $\mathbb{G}$:

$$\begin{aligned} &\min_{\theta} \sum_{D_i} \sum_{x,y\in D_i^q} \left[l\left(y, f_i(x)\right)\right] \\ &\text{s.t. } f_i = g_\theta\left(D_i^s\right), \\ &g_\theta \in \mathbb{G}:\mathbb{R}^{\left(d_x+d_y\right)\left|D_i^s\right|} \mapsto \mathbb{F}. \end{aligned} \tag{3}$$

We use a deep network as g_θ to learn the complex spatial-temporal correlations between provided samples D_i^s and optimized local predictor f_i. We will detail the design of the learnable meta-model g_θ in Section 2.3. The learned model g_{θ^*} is then likewise expected to generalize to unseen ST locations, i.e., have the ability to use a few samples at a new ST location and generate a good adaptive predictor, which can be further used to derive adapted predictions within target ST window. Once the g_{θ^*} is learned, we can complete the second step of coldstart task by feeding the available samples at p_* to g_{θ^*} and use the returned f_* as the final predictor which has explicitly optimized adaption performance.

We also provide a probabilistic interpretation of the proposed formulation following basic Bayesian meta-learning settings similar to [59]. As we will see, the proposed formulation reveals the fine-grained ST structure in predictive distribution by learning a latent factorization of it.

Assuming that there exists a latent state ϕ encoding the ongoing physical air pollution process for a specific ST location, then the samples around it can be considered as i.i.d samples from a parameterized distribution $p_\theta(y \mid x, \phi)$. As the data generation process, the latent states ϕ are first independently sampled from a prior distribution $p_\theta(\phi)$ for all spatiotemporal windows. Then, the samples from each window, including both support set D_i^s and query set D_i^q, can be considered as independent samples from this local distribution when given a specific ϕ. This model allows different data distribution under different latent variable ϕ and therefore is able to provide distinct and adapted prediction patterns for different ST locations. Note that this is a conditional independence assumption and will naturally lead to marginal correlation if the latent variable is integrated out. The probabilistic graph of mentioned variables is illustrated in Figure 3.

The likelihood of observed data under such model can therefore be written as a marginal distribution by integrating over the latent variable, or equivalently we obtain a learnable factorization of the original data distribution, namely,

$$p_\theta\left(D_i\right) = \int p_\theta(\phi) \prod_{x,y\sim D_i^s,D_i^q} p_d(x) p_\theta(y \mid x, \phi) d\phi, \tag{4}$$

where $p_\theta(\phi)$ is a parameterized prior for the latent variable. Learning such a factorization model across different ST locations instead of a merged model explicitly describes the fine-grained ST structures and the sample-level correlations mentioned above. Therefore, it enables us to infer the best posterior (adapted) prediction for the query set when given support set as observations:

$$p_\theta\left(y \mid x, D_i^s\right) = \int p_\theta\left(\phi \mid D_i^s\right) p_\theta(y \mid x, \phi) \mathrm{d}\phi, \tag{5}$$

where we make use of conditional independency between query sample (x, y) and support set D_i^s. In equation (5), $p_\theta(\phi \mid D_i^s)$ is a posterior distribution over a latent variable that is implicitly defined by equation (4) with Bayes rule. The exact evaluation of $p_\theta(\phi \mid D_i^s)$ is usually intractable and needs appropriate approximation. For example, as in [59], we can use a Dirac delta distribution centered at the mode of $p_\theta(\phi \mid D_i^s)$ as a proxy, and this requires us to learn a model that can give a good MAP estimation of ϕ given D_i^s:

$$\max_{g:\phi=g\left(D_i^s\right)} \mathbb{E}_{D_i} \log p_\theta\left(D_i^s \mid \phi\right) + \log p_\theta(\phi), \tag{6}$$

where we omit the intractable normalizer $p_\theta(D_i^s)$ because it is independent of optimization variable g.

In the case that $p_\theta(\phi \mid D_i^s)$ is estimated with a delta function, the integral for $p_\theta(y \mid x, D_i^s)$ in equation (5) then can also be easily computed. We therefore can learn this parameterized probabilistic model by simple maximizing (conditional) log-likelihood estimation criterion:

$$\max_\theta \mathbb{E}_{D_i} \log p_\theta\left(D_i^q \mid D_i^s\right). \tag{7}$$

We find that the proposed learning objective equation (3) can be derived from equation (7) by clarifying specific choice of distribution form. Specifically, by following settings,

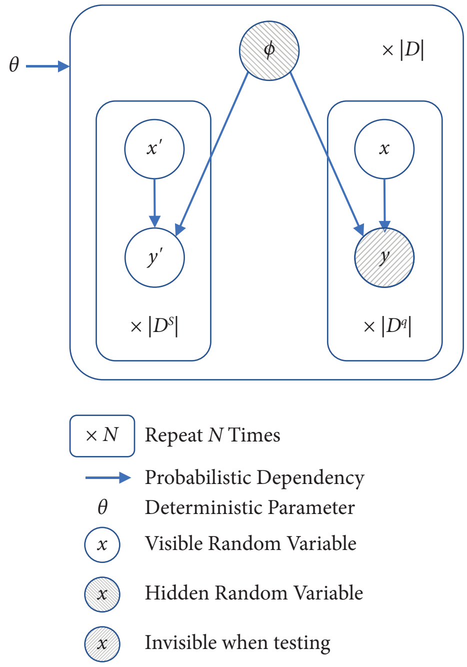


Figure 3: Illustration of the probabilistic graph.

(i) Set latent prior $p_\theta(\phi)$ to a fixed normal distribution without learnable parameters.

(ii) Set conditional predictive distribution $p(y \mid x, \phi)$ into a probabilistic model determined by backbone network b and loss function l. For instance, use Gaussian distribution with model prediction as mean and a constant covariance when using MSE loss.

(iii) Set the meta-learner g in equation (3) as an approximated optimizer for equation (6). We will discuss later in detail why our choice of meta-learner can be considered as such an approximation (Section 2.3).

Under this probabilistic model, we can interpret the proposed STA formulation as simultaneously learning three components. They are a latent representation of the underlying ST process, an inferencer that recognizes the posterior latent estimation through a few given samples, and an adaptive predictor conditioned on different latent status. The inferencer is capable of identifying ongoing patterns, e.g., different city characteristics or different temporal patterns, based on given support samples. The predictor can then use the pattern-specific conditioned prediction, which is trained to better fit local prediction distribution than the global one. Considering the ubiquitous existence of complicated mixed pattern data such as the air pollution process, explicit factorization modeling can potentially make improvements compared to one-model-for-all-sample learning strategy.

2.3. The Bidirectional Meta-Learner. One of the most critical challenges for designing meta-learner g is to better learn the sample level correlations between support and query set, consistent with additional prior knowledge or constraint in equation (6). For the STA problem discussed above, smaller model space and lower data requirement can be achieved by better utilizing the fact that the support set D_i^s is a set of samples rather than arbitrary objects. This leads to three key motivations for the design of support set processing module: Firstly, it should be a set function. It should be invariant to sample order permutation and capable of processing sets of various sizes (M1). Secondly, it should be able to optimize the selected metric. It needs sufficient capacity to automatically learn the task embedding that leads to better adaption performance on query set (M2). Thirdly, it should be able to incorporate the prior knowledge on the similarity between support set and query set. The adapted model that works well for query set should first be accurate for the support set (M3).

We introduce a specific meta-learner structure, called *the bidirectional meta-learner*, that simultaneously meets the above motivations. Our meta-learner produces a task-specific adapted model by conditioning a given backbone predictor with task embeddings extracted from the given support set. As illustrated in Figure 4, the task embeddings are generated by two mechanisms. On the one hand, we use a parametric set function to automatically extract a support set feature that is used to end-to-end optimize required adaption performance on query set (meeting M2). On the other hand, we update the learned embeddings by explicitly requiring its improvement on support set as well (meeting M3). The resulting support set embedding is naturally a valid set function since both involved operators are permutation invariant (meeting M1).

Specifically, for a given parameterized backbone network $b_0 \in \mathbb{F}$ that is designed for air pollution prediction task, we first augment it to accept an additional input for task embeddings ϕ as

$$\hat{y} = b(x, \phi \mid \theta) = \text{MLP}\left(\left[b_0(x), \phi\right] \mid \theta\right), \tag{8}$$

where $[\cdot]$ is the vector concatenate operation and $\text{MLP}(\cdot \mid \theta)$ is a dense layer parameterized by θ. Without losing generality, we simply concatenate the output of b_0 and provided ϕ and pass them through another dense layer to get a fusion result. Note that there is a little assumption we made on the detailed structure of backbone b_0 to make it a model-agnostic framework. Most favorable ST deep network design can therefore be incorporated into our framework and obtain improvement as long as it can be abstracted as $\hat{y} = b_0(x \mid \theta)$. However, it is still possible for users to customize the way to introduce ϕ according to case-by-case network architecture and further increase performance.

The task embedding is composed of two parts $\phi = [\phi_F, \phi_B]$ and is a function of the support set. The first part ϕ_F is obtained by *the forward path* which is a black-box learnable network with support set as input. The second part ϕ_B is obtained by *the backward path* which solves an optimization problem conditioned on the support set.

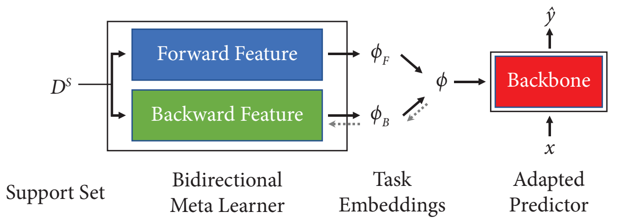


Figure 4: Illustration of the proposed bidirectional meta-learner.

Then, our meta-learner g_θ that gives the ST-location-specific model f_i given the local support set D_i^s can be defined as

$$f_i(x) = g_\theta(D_i^s)(x) = b(x, [\phi_F(D_i^s), \phi_B(D_i^s, x)]), \tag{9}$$

where we omit dependency on θ for b, ϕ_F, and ϕ_B for succinctness. The detailed structures for two paths ϕ_F and ϕ_B are described as follows.

2.3.1. The Forward Path. We learn a forward feature that can optimize future query prediction. In order to enable the model to automatically learn the best task embedding without heuristic restriction, we directly use a deep set network [64] to map the entire support set to the forward feature ϕ_F:

$$\phi_F(D_i^s) = o\left(\frac{1}{|D_i^s|}\sum_{(x,y)\in D_i^s} h(x, y \mid \theta) \middle| \theta\right), \tag{10}$$

where $h:\mathbb{R}^{d_x+d_y}\mapsto\mathbb{R}^{d_h}$ and $o:\mathbb{R}^{d_h}\mapsto\mathbb{R}^{d_{\phi_F}}$ are two different learnable networks with parameter θ. We use θ to denote all learnable parameters in our model for succinctness, where it does not mean o and h share the parameter. This network first independently uses h to map items in support set into a feature space and then puts the mean of all features into another postmapping layer o to derive the final output.

It is well acknowledged that the forward path defined in equation (10) is not only a valid set function due to the permutation invariance of the mean pooling operator but also a universal approximator and thus is capable of learning complex feature from input support set in a black-box fashion. Therefore it satisfies M1 and M2. Similar design has also been adopted in recent works [17, 65].

However, the forward path alone is not enough to provide a satisfying result for the STA problem. The forward path treats the input as general set elements and lacks sufficient usage of important prior knowledge on ST structure and sample level correlation between the support set and target query set. For example, we expect the extracted feature should first best explain the available support set before being used to predict query samples (to satisfy the Motivation M3). But features extracted by black-box learner may not provide such properties. Though there exist various improvements on the deep set structure to enable usage of additional meta-input as conditioned model [64, 66], it will be more efficient to directly embed the knowledge as a soft or hard constraint to model learning. Therefore, we additionally introduce the backward path in supplement to the forward path to leverage such prior knowledge.

2.3.2. The Backward Path. We find a backward feature that can best explain the available support set. Specifically, besides the forward feature that seeks the best query loss, the backward feature is selected as the one that can complete the forward feature to achieve optimal adaptive performance on support set as well. This can be considered as explicit consideration of consistency between the support set and query set. We also let the backward feature as a function of query input. It can be described as the following optimization problem:

$$\phi_B(D_i^s, x) = \operatorname{argmin}_{\phi:\phi(x)}\mathbb{E}_{x,y\sim D_i^s}[l(y, f_i(x))], \tag{11}$$

where $f_i = g_\theta(D_i^s)$ is defined by equation (9) by replacing required $\phi_B(D_i^s, x)$ with optimization variable $\phi(x)$. ϕ_B is then also a valid set function due to the permutation invariance of the expectation operator. Note that the task embedding derived from equation (11) also makes it consistent with the requirement of being an approximated posterior estimator shown in equation (6).

However, directly solving equation (11) as a part of optimization equation (3) will lead to a complex bilevel optimization problem. On the one hand, since the optimal ϕ_B is dependent on D_i^s, it will be computationally expensive to run a solver ϕ_B for each possible D_i^s over which an expectation is required in equation (3). On the other hand, the implicit and non-analytical dependency between ϕ_B and θ in equation (11) will make it hard to estimate the gradient of ϕ_B with respect to θ to enable gradient descent-based optimization procedure of equation (3). Therefore, we need to find a surrogate to ϕ_B that can have a similar effect but an easier optimization procedure.

In order to address the above problem, we resort to a truncated gradient descent estimator as an amortized approximation to equation (11) similar to some previous works [18, 59, 61]. More specifically, we use several gradient descent steps on the objective in equation (11) starting from a learnable initial point to derive an approximated solution. In other words, we learn an initial point $\phi_{B,0}(x \mid \psi)$ that can achieve as good as possible $\phi_B(x)$ for each given D_i^s in several gradient descent steps.

$$\begin{aligned}\phi_B(D_i^s, x) &= \phi_{B,0}\left(x \mid \psi'\right),\\ \psi' &= \psi - \alpha\nabla_\psi\mathbb{E}_{x,y\sim D_i^s}[l(y, f_i(x))],\end{aligned} \tag{12}$$

where α is a predefined hyperparameter that controls update step length and ψ is a learnable parameter and will be considered as a part of θ in the rest of this paper. Here we describe a special case of one gradient step, and the generalization to more steps is straightforward. We emphasize that ϕ_B given by equation (12) removes bilevel optimization issues of equation (11) since it is analytically defined (thus can be conveniently evaluated for given D_i^s) and is differentiable (thus being compatible to backpropagation for equation (3)). The evaluation of equation (12) includes a gradient term that can also be directly computed by backpropagation algorithm, and that is why we name it the backward feature. Though the approximated ϕ_B in equation

(12) is not exactly the solution to equation (11) but a solution to a regularized surrogate according to [59, 67], it is still able to carry similar prior knowledge.

We empirically find that the combination and synergism of the forward path and the backward path lead to a simple but effective meta-learner for the STA problem. Intuitively, we attribute such improvement to the following reasons. The feature ϕ_F in the forward path can be considered as a black-box meta-learner. It is capable of mining underlying task patterns from data in an end-to-end fashion. Besides its generality, this procedure can be less efficient not only because of large assumption space caused by ignorance of specific ST data structure but also because of deficiency in finding a feature with the desired property. However, the feature ϕ_B in the backward path can be considered as a heuristically designed meta-learner. It is designed to explicitly describe the prior knowledge on the ST consistency of the support set and query set. It generates a feature with a specific property but to some extent may miss underlying statistical information and degrade by prior bias. Therefore, the two meta-learners are functionally complementary to each other to leverage learning information simultaneously from prior knowledge and available dataset. A detailed ablation study can be found in Section 4.

The dataflow of our model is illustrated in Figure 5. For a support set that consists of a few labeled samples at a given ST location, the forward feature is first computed by the forward path. The initial backward feature is appended to this forward feature to get an initial value for task embedding. Then, the task-specific model conditioned by initial task embedding is evaluated on the entire support set, where the adaption gradient for backward feature that improves support set performance can be further calculated. Then, a few gradient steps are applied to the initial value to get a finalized estimate of the backward feature. The task-specific model conditioned on the updated task embedding can finally be used to evaluate adaption performance on query set samples. The gradient of query set performance with respect to all (initial) model parameters can be calculated to update all model parameters.

2.4. A New Coldstart Algorithm.

We summarize and instantiate our novel learning algorithm based on the aforementioned STA formulation and bidirectional meta-learner for the challenging coldstart air pollution prediction scenario.

Our algorithm first learns a meta-model that has STA ability from the source city dataset, which can learn and identify specific ST patterns within a ST window and make adaptive predictions. Then, with an assumption that the prediction pattern at target city can be found in source history records (i.e., we assume invariant distribution at task level instead of data sample level), we can naturally recognize target city with a few labeled samples as a new ST window. Therefore, an adapted prediction model for the target city can be generated by the trained meta-model and can be directly used for future usage.

The main workflow is illustrated in Figure 6. The algorithm can be decomposed into three phases, i.e., meta-dataset construction from raw source sensing data, training a meta-model to enable given backbone network STA ability, and generating an adapted model for the target city. The detailed algorithm can be found in Algorithm 1.

Compared to the traditional pretraining and fine-tuning scheme, the proposed STA formulation and the corresponding bidirectional meta-learner are favorable for addressing the coldstart problem for the following reasons. Firstly, the proposed model does not learn a predictive distribution directly from the limited target data but instead recognizes it as a component of the source dataset. This approach intuitively reduces the requirement for target data accumulation and can lead to better coldstart performance. It also avoids model parameter updates that can be unstable when dealing with limited samples. Secondly, compared to the heuristic effectiveness of the fine-tuning procedure, the output of the meta-model is explicitly optimized over the source dataset to give better spatial-temporal adaptation performance.

(1) We first sample several ST windows of length $2d_0 = L_s + L_q$ from source cities as meta-datasets, from which several samples (x, y) can be further sampled by sliding windows to construct a local mini-dataset D_i. D_i is further split into the support set and query set. In other words, we are constructing some simulated coldstart tasks in the available source data.

(2) The meta-model, i.e., the aforementioned bidirectional meta-learner, is explicitly trained on the constructed meta-dataset. For a given ST window and corresponding support and query set, the support set is sent into the bidirectional meta-learner to get a task-specific prediction model. Model performance is evaluated on the query set and optimized with respect to meta-model parameters until convergence.

(3) The adapted prediction model for the target city can be given by the trained meta-model with provided samples as support set. The trained meta-model can also provide consecutive adaption results if further labeled samples can be provided.

2.5. Experiment Setting

2.5.1. Dataset.

In order to analyze model performance under a clear and controllable data distribution, we construct a synthetic sinusoidal dataset similar to [18]. The dataset is designed to have a similar spatiotemporal data structure to real air pollution data but has much simpler dependency between covariant x and target y. More specifically, we consider $y = A\sin(\omega x + \varphi)$ as ground-truth prediction function for each space/time stamp, where x is a scalar independently and uniformly sampled from [0,1]. The parameters of the sinusoidal function A, ω and φ, are used to mimic the different underlying physical processes. These parameters will be identical for all samples within

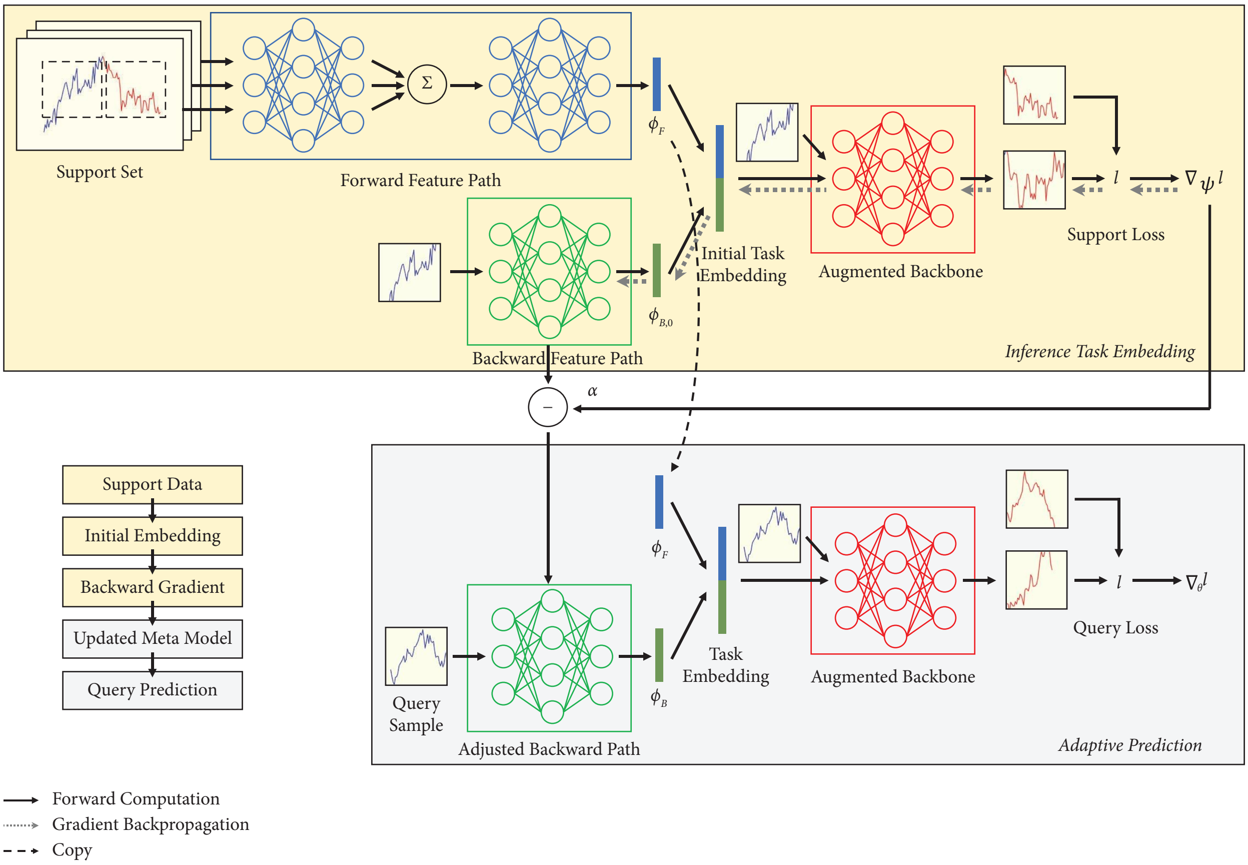


FIGURE 5: Illustration of model dataflow. The text flowchart on the left summarizes the main steps of the computation, while the rest shows the dataflow of corresponding steps in detail. We use neural networks in different colors to denote learnable parts in different modules, i.e., blue for the forward feature path o and h, green for the backward feature path ψ, and red for augmented backbone network b.

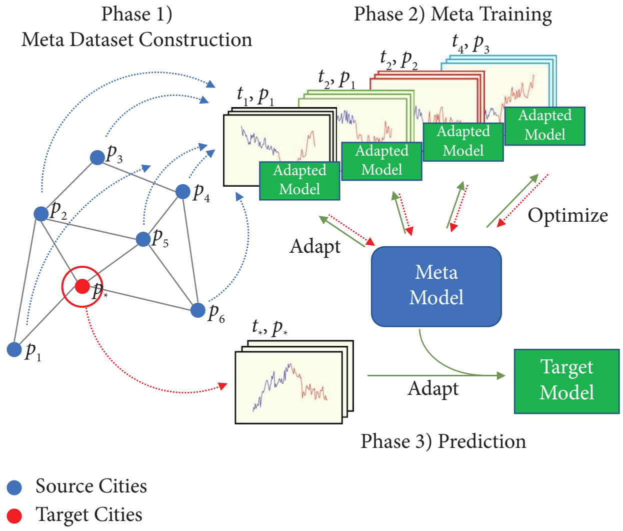


FIGURE 6: Illustration of the workflow of Algorithm 1. The ST location on the top of each D_i aims to emphasize that the meta-dataset is constructed across different space and time on source history data.

a spatial-temporal window. Parameters for different ST windows will be different and will be independently sampled from the Gaussian distributions $A \sim \mathcal{N}(10,4)$, $\omega \sim \mathcal{N}(10,4)$, and $\varphi \sim \mathcal{N}(\pi/2,\pi^2/4)$. For a specific ST window, we set the size of the support set to 5 and the query set to 100, i.e., the model is required to use 5 recent samples

```
Input: Source city data $\{c_{t,p_i}\}$, target city data $\{c_{t,p_*}\}$, backbone network $b_0$
Output: Prediction model for target city $f_*$
* Construct Meta-Dataset
Init $D = \{\}$
for $\tau$ in range of target meta-dataset size do
    Sample a source city $p_i$
    Sample a window slice $[t_0 - L_s, t_0 + L_q], L_s + L_q \gg L_o + L_p$
    Build $D_\tau^s = \{(x_{t,p_i}, y_{t,p_i}) | t \in [t_0 - L_s, t_0]\}$ as support set using equation (1)
    Build $D_\tau^q = \{(x_{t,p_i}, y_{t,p_i}) | t \in [t_0, t_0 + L_q]\}$ as query set using equation (1)
    Append $D_\tau = (D_\tau^s, D_\tau^q)$ into $D$
end for
)
* Meta-Training
Random initialize $\theta$
while not convergence do
    Sample a batch of STA tasks $B = \{(D_i^s, D_i^q)\} \subset D$
    for $(D_i^s, D_i^q) \in B$ do
        Compute $\phi_F$ using 10
        Compute $f_i$ with $\phi = [\phi_F, \phi_{B,0}]$ using 9
        Compute $\phi_B$ using 12
        Update $f_i$ with $\phi = [\phi_F, \phi_B]$ using 9
        Evaluate objective 3 and corresponding gradients with respect to $\theta$
    end for
    Apply a gradient update on $\theta$ using average gradient within batch $B$.
end while

* Generate Target Model
Build $D_*^s = \{(x_{t,p_*}, y_{t,p_*})\}$ as target support set using equation (1)
Compute $\phi_F$ using 10
Compute $f_*$ with $\phi = [\phi_F, \phi_{B,0}]$ using 9
Compute $\phi_B$ using 12
Update $f_*$ with $\phi = [\phi_F, \phi_B]$ using 9
Return $f_*$
```

Algorithm 1: Spatiotemporal adaption for coldstart prediction.

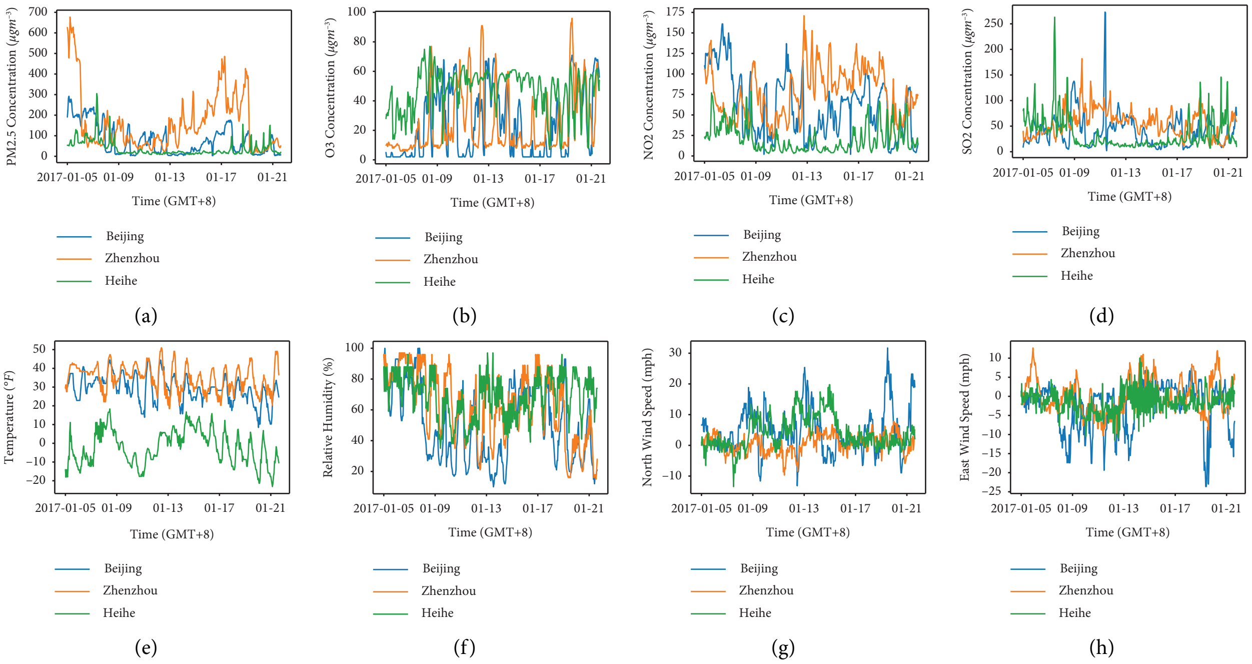
to recognize the current task and will be evaluated on future 100 samples. Note that a traditional training scheme that mixes up all samples and ignores ST structure can only learn an averaged curve (expectations over parameters) and fail to give a satisfying result. 500 independent tasks are generated on fly for each training epoch. Other 1000 and 10000 independent tasks are independently generated as the validation set and test set.

We also evaluate our model on a real-world air pollution dataset, where we collect air pollution data in 50 main cities in China for 4 distinct pollutants. The air pollution concentration is hourly reported by national air quality monitoring stations, and we choose one monitoring station that has the longest available history record at each city. We also collect regional same-period meteorological data as auxiliary features. The meteorological data including temperature, humidity, wind speed, and wind direction are reported in three-hour intervals, and thus we linearly upsample them into hourly series in alignment with air pollution data. We also apply some common preprocessing on raw data for machine learning convenience, e.g., complete a few missing values by linear interpolation across time and normalize different physical values to 0-1 with their global statistics (1% and 99% percentile values in consideration of extreme values). Detailed dataset statistics are summarized in Table 1. Data samples are illustrated in Figure 7. We can see that data pattern significantly differs across space (cities) and time and adaptive predictions are required. The geographical distribution of the involved cities is presented in Figure 8. The dataset exhibits a comprehensive coverage of diverse locations, where spatial-temporal adaption ability can be better learned.

The dataset is first spilt along time at a ratio of 50%:25%:25% as training, validation, and testing period. We also exclusively and randomly choose 30, 10, and 10 cities out of total 50 cities as training, validation, and testing cities. Data splits across time and space are correspondingly paired. More specifically, data from the first 50% time in the first 30 cities are used to train the model. Model selection and hyperparameter tuning are performed on the next 25% time and 10 cities. Then, the model is evaluated on the last 25% time and 10 cities. We adopt such disjoint separation to best test model performance under canonical coldstart prediction data settings. Without losing generality, this split is fixed for all experiments and we did not test other split. Then, we generate a meta-dataset from raw time series according to Algorithm 1.

Table 1: Statistics of the China air pollution dataset.

Item	Description
Number of cities	50
Latitude span	18.24°N–45.75°N
Longitude span	87.58°E–126.54°E
Time span (UTC+8)	2015/1/2 00:00–2020/10/30 23:00
Pollutants	$PM_{2.5}$, O_3, NO_2, SO_2
City avg. $PM_{2.5}$ span	14.35–75.52 ($\mu g/m^3$)
City avg. O_3 span	40.77–82.67 ($\mu g/m^3$)
City avg. NO_2 span	9.24–58.49 ($\mu g/m^3$)
City avg. SO_2 span	2.91–56.41 ($\mu g/m^3$)
Meteorologic	Temperature, humidity, wind speed, wind direction
Record interval	1 h (pollutants), 3 h (meteorologic)



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 7: Samples from the China air pollution dataset. Here we show $PM_{2.5}$ (a), O_3 (b), NO_2 (c), SO_2 (d), temperature (e), humidity (f), north wind speed (g), and east wind speed (h) data from three cities. Here pollutants are the target variables to be predicted and others are considered auxiliary information.

For the training set, a time window of length 350 hours (i.e., $L_s + L_q = 350$, roughly two weeks) for a single site is considered as a ST neighborhood, where the first 200 hours is used to build support set and the rest 150 hours is used to build query set. The covariate x of a sample is composed of $L_o = 8$-hour pollutant concentration observations along with corresponding weather observation and $L_p = 24$-hour weather forecast. The target y is then the future 24-hour pollutant concentration that follows x. At last, the number of different ST tasks in the training, validation, and testing set is 5040, 830, and 830, where in each task there are 168 support samples and 118 query samples.

2.5.2. Model Parameters. Considering that the proposed method is a model-agnostic algorithm and without loss of generality, we use an MLP as the backbone predictor b throughout the experiment. We also set o, h, and ψ to MLP as well. All involved MLPs have two 128-neural hidden layers and use the ReLU activation. The forward feature and backward feature are both set to 32 dimensions, and thus the final task embeddings have 64 dimensions. The number of truncated gradient steps in the backward feature is set to 1, and the update step length is set to $\alpha = 0.06$ for the synthetic dataset and $\alpha = 0.6$ for the air dataset. The loss function is set to MAE loss $l(y, \hat{y}) = |y - \hat{y}|$. Most hyperparameters are set by empirical values or grid-searched within a predefined set.

We implement the model with the TensorFlow package. The model is optimized for 300 epochs by the Adam optimization algorithm with the default settings in the TensorFlow package. The learning rate is fixed at 0.001 during the training process. We select the model with the best validation performance during training as the trained model. All experiments, if available, are repeated 5 times with

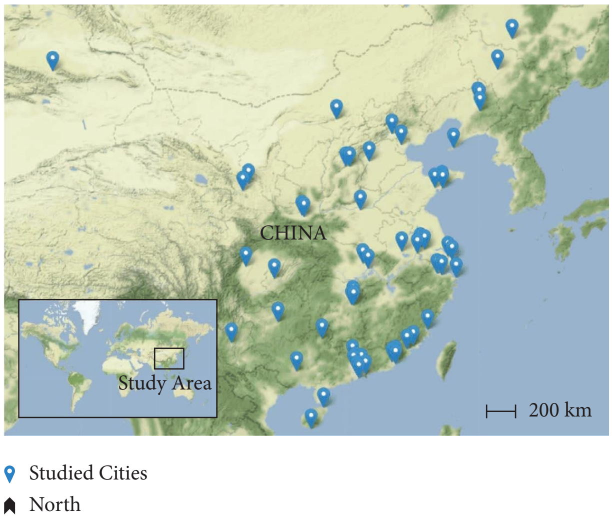


Figure 8: The geographical distribution of 50 cities in the China air pollution dataset.

different random seeds. We report the result as the mean and 95% confidence interval evaluated from 5 outcomes.

2.5.3. Baselines and Metrics. We compare the proposed method with the following baseline models. All baselines are implemented with the same backbone network as ours for a fair comparison. First, we list three approaches that are not trained under the proposed STA formulation:

(i) *Backbone Net*. We trained the backbone network as a conventional data-driven air pollution prediction model learning scheme, i.e., mix up all training samples and optimize average sample prediction MAE. The trained model is directly applied to new cities in the test set without further adjustment. We list this as a reference line under specific model capacity and data accumulations for the given backbone. This is a non-adaptive approach.

(ii) *Linear Regression*. We train a ridge regression model from scratch on the provided support samples for each test task. This approach ignores available source city data. We did not test more complex models for this case in consideration of overfitting to extremely limited support samples.

(iii) *Transfer Learning*. We report the performance of the popular pretraining and fine-tuning strategy. We use the trained model in backbone net as pretrained model and update model prediction error on the provided support samples independently for each test task by a fixed number of gradient steps.

We also include different meta-learning models as the meta-learner under the proposed STA formulation:

(i) *MAML* [18]. The model-agnostic meta-learning method is a meta-learner that adapts to the target task purely relying on gradient information on support set. The parameters of MAML are set similarly to the backward path in our model.

(ii) *CNP* [17]. The conditional neural process method is a meta-learner that adapts to the target task purely relying on a learnable feature computed from the support set. The parameters of CNP are set similarly to the forward path in our model.

(iii) *MetaST* [11]. As a variation of the MAML model, a learnable memory unit is added along the gradient information on the support set to learn both short-term and long-term temporal dependencies in data. Since the model is originally designed for the few-shot spatiotemporal traffic prediction task, we replace its backbone network and meta-learning parameters to be similar to our model for the air pollution prediction task.

(iv) *cSTML* [12]. The continuous spatial-temporal meta-learner is a variation of CNP models that use a learnable long short-term memory (LSTM) network to extract a time-dependent (rather than permutation invariant) feature on the support set. It is customized for traffic status prediction and we also replace its backbone network and meta-learning parameters similar to ours.

(v) *MetaFun* [56]. The iterative functional meta-learner considers a functional representation of a task instead of finite-dimension embeddings. In implementation, it introduces a learnable parameter update formula based on gradient information on the support set.

(vi) *Ours*. The proposed method.

We use the mean absolute error (MAE) between prediction and target as the evaluation metric in most of our experiments, which is averaged across different test tasks, different query samples within each task, and different dimensions of the target variable. More specifically, we have

$$\mathrm{MAE} = \frac{1}{Z} \sum_{\substack{D_i^q \in D \\ (x,y)\in D_i^q \\ j\in \left[1,d_y\right]}} \left|y_j - \hat{y}_j\right|, \tag{13}$$

where we use Z to denote the total number of averaged terms in the summation. The lower the MAE is, the better the model is. We also tested the root mean square error (RMSE) and the mean absolute percentage error (MAPE) for our main result, which are defined as

$$\begin{aligned}
\mathrm{RMSE} &= \sqrt{\frac{1}{Z} \sum_{D_i^q \in D\,(x,y)\in D_i^q\, j\in \left[1,d_y\right]} \left|y_j - \hat{y}_j\right|^2}, \\
\mathrm{MAPE} &= \frac{1}{Z} \sum_{D_i^q \in D\,(x,y)\in D_i^q\, j\in \left[1,d_y\right]} \frac{\left|y_j - \hat{y}_j\right|}{\left|y_j\right|}.
\end{aligned} \tag{14}$$

3. Result

With our main result presented as the proposed new coldstart algorithm (Algorithm 1), in this section, we report the main experimental result that validates its effectiveness under a typical coldstart scenario.

3.1. Results on the Synthetic Dataset

3.1.1. Prediction Samples. Can the proposed method adapt to the target with a few samples on synthetic dataset? We visualize model samples in Figure 9. The result shows that the model correctly learns the common knowledge that all tasks involved are sinusoidal waves and can give a reasonable adaptive prediction for each task with only a few support samples. This means that the STA formulation can enable the model to have adaption performance, i.e., to find fine-grained structure and learn common knowledge across tasks and adapt to target task with only a few available observations. We also find that task inference quality and adaption performance can be well improved by providing more support samples, though the model is only trained with 5 support samples. This implies that the model correctly learns to extract useful support set embeddings. Note that ignorance of the sample-level data structure will be equivalent to stacking all possible sinusoidal curves together and will lead to huge ambiguity (possible y for a specific x will be almost uniformly distributed instead of a clear curve).

3.1.2. Baseline Comparison. Can the proposed method effectively improve coldstart prediction precision on synthetic dataset? We evaluate the adaptive prediction MAE of our model and other baselines on the synthetic dataset in Table 2. The results reveal the following insights. (1) The model trained with the proposed STA formulation significantly outperforms those without it. This is expected since the synthetic dataset has clear data structures and sample-level correlations, i.e., there is an underlying process of varying sinusoidal parameters and nearby samples are more likely to share similar task parameters. Ignoring such structures and mixing up all samples, e.g., traditional transfer learning strategies, will result in an over-smoothed average pattern, which can be far worse than the optimal solution. Discarding source datasets and learning a low-complexity model using limited available data, like linear regression, is also suboptimal. (2) Our bidirectional meta-learner achieves the best performance among all tested meta-learning algorithms. We attribute this performance to the synergy of using both black-box learned support embeddings and optimization-based feature construction. On average, we can draw a similar conclusion to [68] that black-box meta-learners, like CNP, outperform gradient-based meta-optimizers, like MAML, and MetaST. In particular, we find that MetaFun, which is designed as a black-box meta-learner but also implicitly performs gradient-based update, can be also considered as a two-mechanism meta-learner like ours and shows good performance among all baselines. However, our model explicitly uses two mechanisms and achieves the best result. (3) Almost all models benefit from more support samples, i.e., the performance improves from 5-shot test to 20-shot test. Our model shows the largest relative improvement, and the previous two conclusions consistently hold for different numbers of support samples.

3.2. Results on the Air Pollution Dataset

3.2.1. Prediction Samples. Can the proposed method enable better adaptive air pollution prediction? We visualize in Figure 10 some of the samples of our model compared to the backbone network trained without STA modeling. We find that our model can better trace the true curve trend as well as better estimate the peak and valley value. As shown in Table 1, the average pollution level can be very different across cities. Therefore, it is necessary for a newly settled model to make a target-adapted prediction rather than using a fixed strategy that works averagely well in history. We attribute this improvement of our model to the usage of STA formulation and better target pattern estimation.

3.2.2. Baseline Comparison. To what extent the proposed method can reduce the coldstart prediction error? We compare our model with all baselines for four different air pollutants. We first report the result of the source test in Table 3, i.e., we train the model on history source city data and test the adaptive prediction performance on future source city data. This aims to reflect the model's adaption ability within source cities that have sufficient history training data. From the result, we can see that transfer learning-based solutions can only provide slight improvement (on average 2.4% relative error reduction over the backbone network). However, the proposed STA formulation, equipped with various meta-learner implementations, can give far better results with the same backbone network, same source dataset, and similar training resources. Especially, our bidirectional meta-learner achieves the best result that on average reduces 11.1% relative error over the backbone network. Among the different

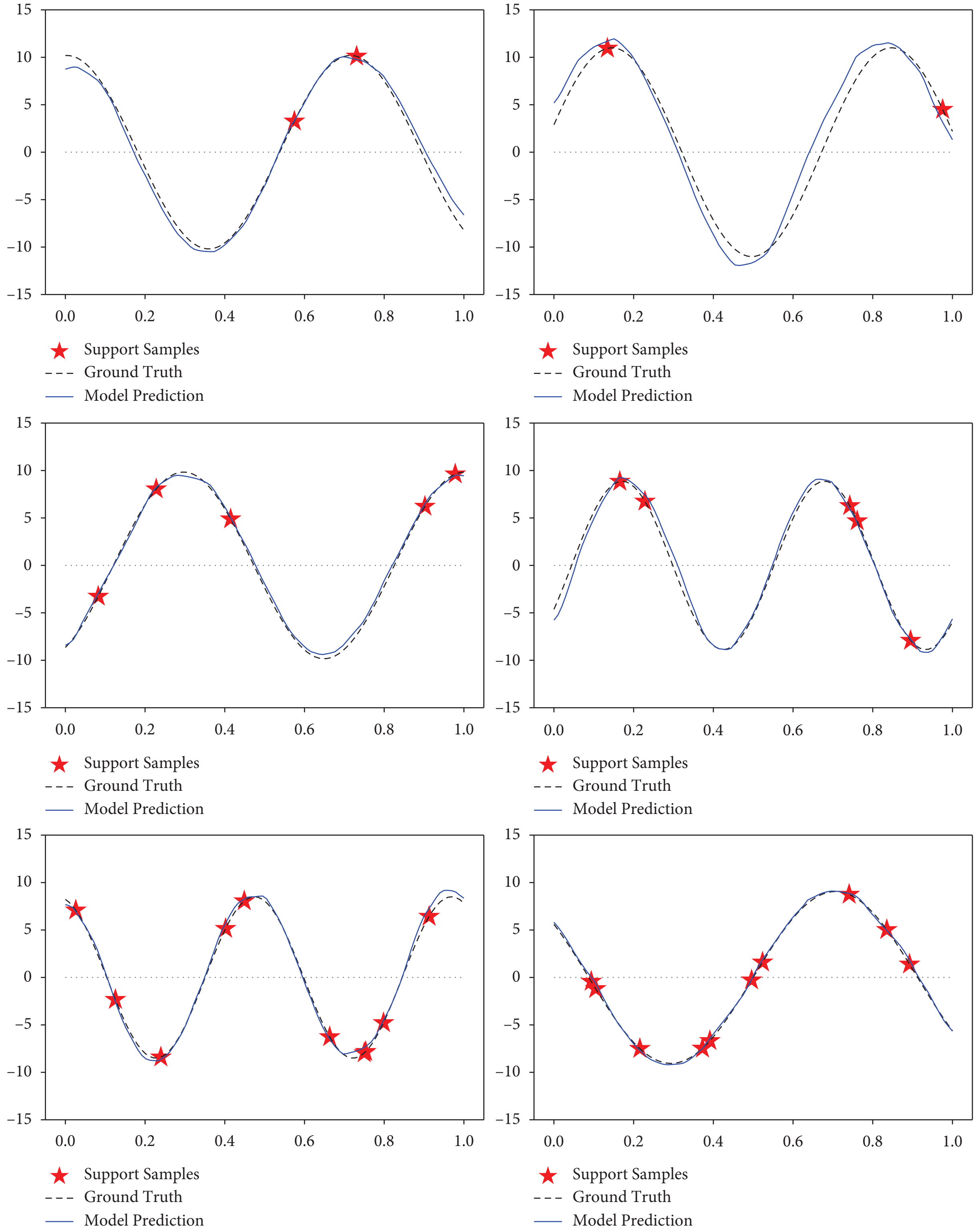
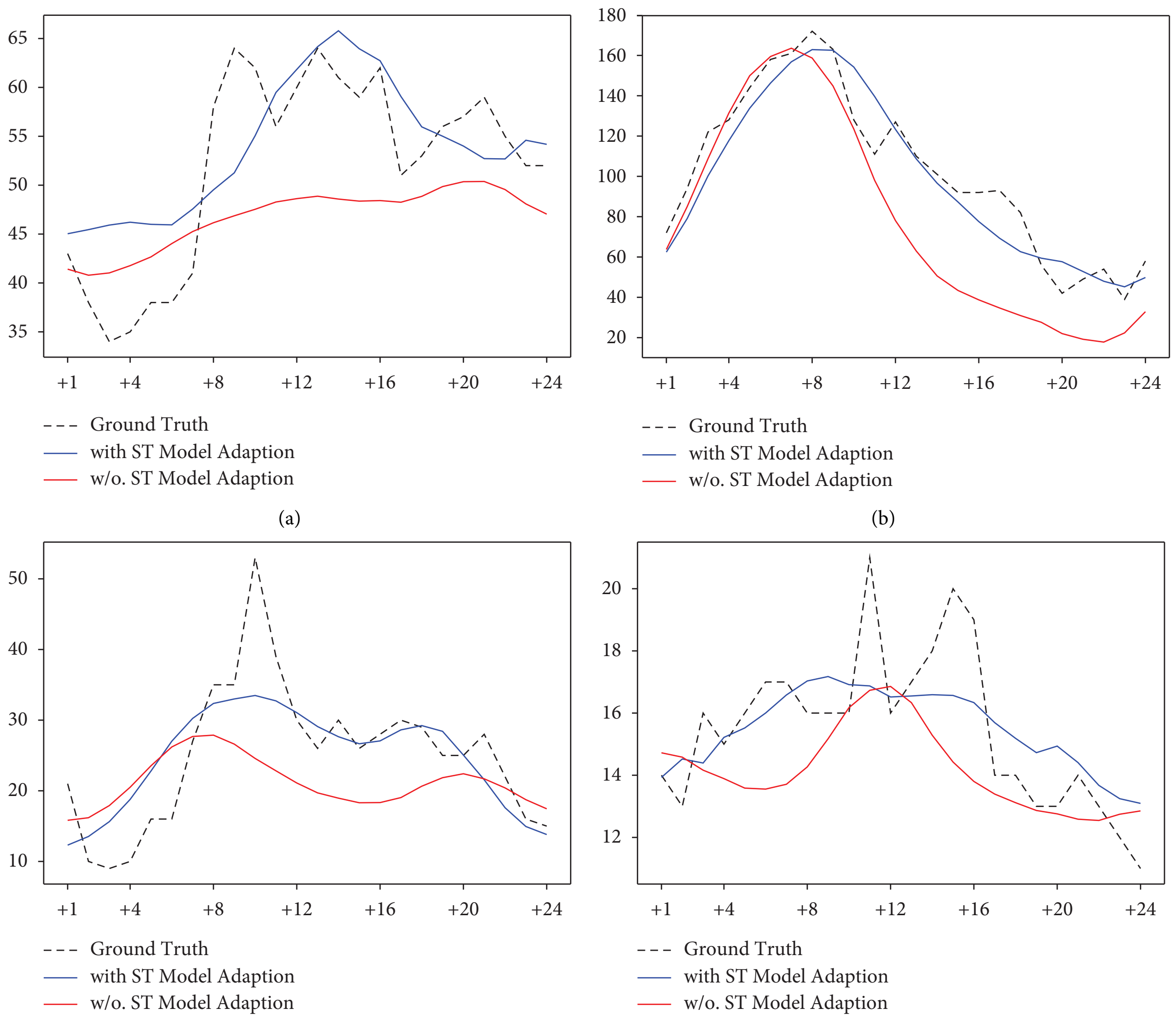


Figure 9: Prediction samples on the sinusoidal synthetic dataset. Each subfigure is an independent task. The model is given only a few support samples (marked in star), to adapt to the underlying sinusoidal curve (black dashed). The blue curve shows all predictions by varying possible inputs. The three rows, from up to down, are, respectively, tasks with 2, 5, and 10 independent support samples.

Table 2: MAE comparison on synthetic dataset.

Method	MAE	
	5-shot	20-shot
W/o STA		
Linear regression	6.389 ± 0.014	6.286 ± 0.014
Backbone net	6.164 ± 0.010	6.164 ± 0.010
Transfer learning	5.416 ± 0.069	5.079 ± 0.082
With STA		
cSTML [12]	4.011 ± 0.476	4.293 ± 0.742
MAML [18]	2.256 ± 0.089	0.928 ± 0.031
CNP [17]	1.638 ± 0.053	0.665 ± 0.082
MetaST [11]	2.093 ± 0.050	0.744 ± 0.056
MetaFun [56]	2.026 ± 0.051	0.333 ± 0.011
Ours	**1.275 ± 0.060**	**0.232 ± 0.008**

Bold values indicate the best performance.



(a)

(b)

(c)

(d)

Figure 10: 24-hour samples on air pollution dataset: (a) $PM_{2.5}$, (b) O_3, (c) NO_2, and (d) SO_2. The X-axis is the prediction time stamp in hour, i.e., from +1 to +24 for 24 h prediction. Y-axis is the concentration value in $\mu g/m^3$.

choices of meta-learners, we find similar phenomena observed in the synthetic dataset under high dimensional input structure, i.e., gradient-based meta-learner (like MAML) gives better results than learnable support set feature extractor (like CNP). The proposed bidirectional meta-learner shows its compatibility towards such data characteristic and still provides reliable adaption ability, not only outperforming both MAML and CNP but also providing slight improvement.

We then report results on coldstart test in Table 4, i.e., we train the model on history source city data and test the adaptive prediction performance on unseen target cities. Though being a much harder scenario and with generally larger error, we still find similar results to the above source test case. Transfer learning-based solution failed to provide reasonable transfer benefits on average and even led to negative transfer for some pollutants (e.g., −3.9% relative error reduction on SO_2 dataset), which can be considered as the result of overfitting to the limited target data. However, our STA formulation still shows satisfying adaption results and our bidirectional meta-learner achieves the best 5.2% average relative error reduction. It is important to note that such improvement is obtained under same backbone network, same dataset (without additional auxiliary data), and similar optimization procedure.

In order to more comprehensively report the statistics of prediction error, the coldstart test performance under different error metrics is listed in Table 5 for $PM_{2.5}$ as an example. Besides the MAE ($\mu g/m^3$) mentioned above, we also list the root mean square error (RMSE, in $\mu g/m^3$) and mean absolute percentage error (MAPE). We find that their results are similar to MAE case and our previous analysis between baselines also holds. Our method gives better result than both non-adaptive solutions and adaptive solutions implemented by other meta-learners.

4. Discussion

We present the discussion about the proposed coldstart algorithm through comprehensive experiments.

4.1. Discussion on the Synthetic Dataset

4.1.1. Ablation Study. How do submodules of the proposed bidirectional meta-learner contribute to improvement? We further demonstrate the effectiveness of the proposed bidirectional meta-learner by conducting the module ablation study. In Table 6, we list the MAE performance of our model under different module combinations. We, respectively, remove (indicated by ×) the STA formulation (MA, result in backbone alone), the forward feature (FF), or the backward feature (BF) from our model (correspondingly named as Ours-MA, Ours-FF, and Ours-BF). AUG is an augmented module for the forward path and will be discussed later. We find that when only using the forward feature alone (row 2), our model degenerates to a special case similar to CNP. When only using the backward feature alone (row 3), our model degenerates to another special case similar to MAML. Though the two features alone can to some extent provide adaption performance, their combination (row 4) offers additional improvement and achieves the best MAE. We intuitively attribute this to the fact that the two mechanisms can functionally complete each other.

How does the proposed bidirectional meta-learner fit different input structures? The two mechanisms involved in the proposed bidirectional meta-learner can, respectively, contribute more to the final result when the data characteristic changes. We empirically verify it through two additional cases. On the one hand, we first consider a case where it is possible to access extra task information beyond the target support set. We may not have precise prior knowledge to design corresponding optimization problems as the backward feature, but it is compatible to include it in forward feature. To investigate this, we modify the task generation process to let the parameters of different tasks on one time slice be correlated with each other (A, ω, φ is sampled from a joint Gaussian with non-diagonal covariance). The forward feature is then augmented with a multi-head attention block [69] (named AUG module) to fuse other task's support set as additional input. In row 5 and row 6 of Table 6, we can see that the AUG module can bring additional error reduction by making good use of this information. On the other hand, the backward feature that builds with prior knowledge may be more efficient when the input data structure is of high complexity or high dimension, where it may be difficult to automatically learn effective task embeddings from limited data resources. We simulate complex sample structure by appending additional dummy dimensions of noise onto input x (but target y remains unchanged), thus turning the scalar-scalar regression problem into a vector-scalar regression problem. When setting task variance $\epsilon = 0.2$ and setting additional input dimensions to 0, 32, and 64, Ours-FF model reports robust MAE of 0.604, 0.636, and 0.644. But Ours-BF degrades fast when the dimension increases, reporting MAE of 0.408, 0.560, and 0.767, respectively. Our full model consistently shows the best performance, with MAE of 0.259, 0.504, and 0.604, respectively. These results imply that our bidirectional meta-learner, as the combination of two mechanisms, can fit wider data characteristics and therefore have boarder application potential.

4.1.2. Parameter Study. How does result change with model hyperparameters? We draw the prediction MAE under different backward feature update step numbers and step lengths in Figures 11(a) and 11(b). We find that the best update step length is around 0.06. Lower or larger step length deteriorates the performance. In particular, the step length 0 will lead to a dummy backward path (the backward feature is directly computed and not adjusted by support data), and model performance will degrade to a level similar to forward path only case in Table 6. Step number 1 can provide good performance and results are similar for 1–4 steps.

How does result change with task properties? The synthetic dataset allows us to control the properties of the generated dataset and allows us to study how data structure influences model performance. We vary the variance of task

Table 3: MAE comparison on air dataset: source test.

Method	MAE ($\mu g/m^3$)				
	$PM_{2.5}$	O_3	NO_2	SO_2	Average
W/o STA					
Linear regression	15.745 ± 0.000	25.274 ± 0.000	11.458 ± 0.000	3.785 ± 0.000	14.065
Backbone net	9.983 ± 0.094	18.347 ± 0.176	8.313 ± 0.070	2.542 ± 0.042	9.796
Transfer learning	9.880 ± 0.105	18.096 ± 0.101	7.874 ± 0.042	2.384 ± 0.026	9.558
With STA					
cSTML [12]	10.594 ± 0.322	18.827 ± 0.223	8.677 ± 0.081	3.139 ± 0.170	10.309
MAML [18]	**9.413 ± 0.080**	16.147 ± 0.089	7.451 ± 0.065	2.295 ± 0.020	8.827
CNP [17]	9.503 ± 0.090	16.764 ± 0.212	7.834 ± 0.072	2.329 ± 0.031	9.108
MetaST [11]	**9.413 ± 0.057**	16.088 ± 0.257	7.470 ± 0.071	2.327 ± 0.035	8.825
MetaFun [56]	9.447 ± 0.072	16.566 ± 0.398	7.551 ± 0.056	2.265 ± 0.028	8.958
Ours	**9.437 ± 0.052**	**15.817 ± 0.076**	**7.350 ± 0.051**	**2.235 ± 0.042**	**8.710**

Bold values indicate the best performance.

Table 4: MAE comparison on air dataset: coldstart test.

Method	MAE (μg/m^3)				
	$PM_{2.5}$	O_3	NO_2	SO_2	Average
W/o STA					
Linear regression	19.272 ± 0.000	30.136 ± 0.000	14.440 ± 0.000	10.975 ± 0.000	18.706
Backbone net	11.420 ± 0.168	18.572 ± 0.103	9.476 ± 0.100	6.204 ± 0.041	11.418
Transfer learning	11.374 ± 0.049	18.672 ± 0.196	9.131 ± 0.094	6.452 ± 0.064	11.407
With STA					
cSTML [12]	11.997 ± 0.141	18.772 ± 0.112	9.708 ± 0.165	6.692 ± 0.034	11.792
MAML [18]	11.145 ± 0.051	17.635 ± 0.120	8.779 ± 0.049	6.060 ± 0.044	10.905
CNP [17]	11.205 ± 0.123	18.371 ± 0.241	8.878 ± 0.035	5.998 ± 0.095	11.113
MetaST [11]	11.132 ± 0.103	17.648 ± 0.054	8.732 ± 0.035	6.071 ± 0.031	10.896
MetaFun [56]	11.115 ± 0.091	17.849 ± 0.075	8.856 ± 0.090	6.009 ± 0.086	10.957
Ours	**11.043 ± 0.092**	**17.595 ± 0.069**	**8.671 ± 0.057**	**5.961 ± 0.015**	**10.817**

Bold values indicate the best performance.

Table 5: Coldstart test for different error metrics.

Method	$PM_{2.5}$ prediction error		
	MAE	RMSE	MAPE
W/o STA			
Linear regression	19.272 ± 0.000	31.558 ± 0.000	1.174 ± 0.000
Backbone net	11.420 ± 0.168	18.278 ± 0.135	0.673 ± 0.043
Transfer learning	11.374 ± 0.049	18.395 ± 0.093	0.670 ± 0.016
With STA			
cSTML [12]	11.997 ± 0.141	18.757 ± 0.166	0.769 ± 0.062
MAML [18]	11.145 ± 0.051	18.215 ± 0.167	0.631 ± 0.015
CNP [17]	11.205 ± 0.123	18.050 ± 0.028	0.661 ± 0.053
MetaST [11]	11.132 ± 0.103	18.273 ± 0.056	0.633 ± 0.032
MetaFun [56]	11.115 ± 0.091	18.019 ± 0.116	0.650 ± 0.033
Ours	**11.043 ± 0.092**	**18.006 ± 0.192**	**0.621 ± 0.020**

Bold values indicate the best performance.

parameters and therefore control the significance of inner data structures. Specifically, we time the standard deviation of task parameters A, ω, φ with an additional parameter $\epsilon \in [0.1, 1]$, e.g., sample $A \sim \mathcal{N}(10, 4\epsilon^2)$. Therefore, $\epsilon = 0$ will be identical to traditional i.i.d. settings and $\epsilon = 1$ recovers previous experiments. The result shown in Figure 11(c) indicates that our model can consistently provide adaption ability when task data structure exists. When increasing the task discrepancy, both the absolute improvement of adopting the STA, i.e., the difference between backbone network and meta-learning-based method group, and relative advantages of using the bidirectional meta-learner, i.e., ours compared to other meta-learners, are getting larger.

Table 6: Module ablation study on synthetic dataset.

MA	FF	BF	AUG	MAE
×	×	×	×	6.164 ± 0.010
√	√	×	×	1.610 ± 0.059
√	×	√	×	2.789 ± 0.027
√	√	√	×	**1.275 ± 0.060**
√	√	×	√	1.358 ± 0.068
√	√	√	√	**1.051 ± 0.059**

Bold values indicate the best performance.

4.2. Discussion on the Air Pollution Dataset

4.2.1. Ablation Study. How do submodules of the proposed bidirectional meta-learner contribute to improvement on real-world coldstart data? We test the prediction MAE of different ablation variants of our model on the air pollution dataset to verify the effectiveness of the bidirectional meta-learner. As the result shown in Table 7, our full model achieves the best coldstart prediction MAE on all 4 pollutants compared to Ours-FF and Ours-BF. The result is consistent with our findings on the synthetic dataset that both the forward feature and backward feature alone can have a certain adaption ability (compared to MAE of backbone network trained without STA formulation listed in Table 4), while their combination as the proposed bidirectional meta-learner can further provide improvement.

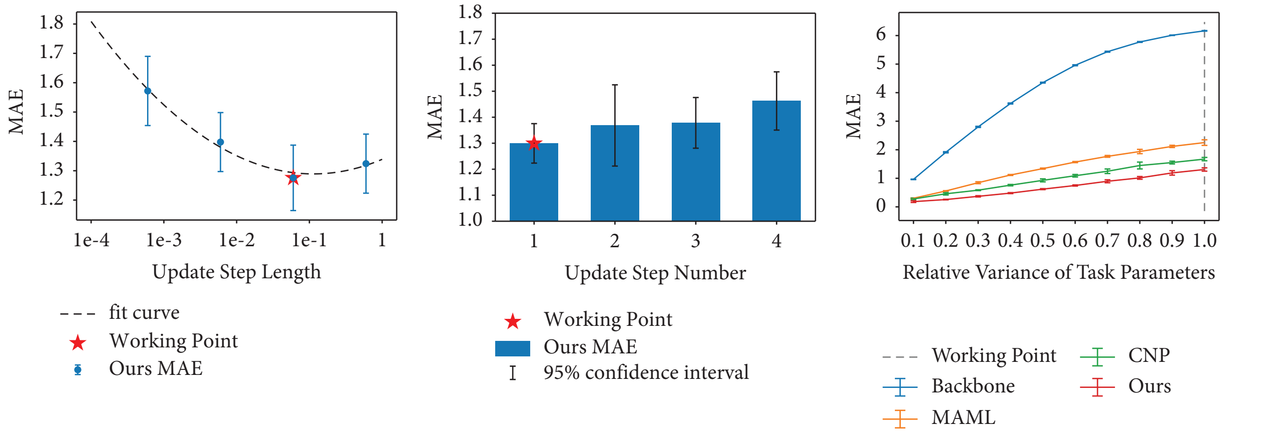
Can the model benefit different backbone networks? The proposed method is a model-agnostic framework that enables a given backbone predictor the STA and coldstart ability. We conduct experiments to evaluate the performance of the proposed method using different backbone networks (b_0), and the results are presented in Table 8. In addition to the MLP structure, we employ three commonly used architectures for temporal modeling, namely, 1D convolutional network (CNN), long short-term memory (LSTM), and transformer decoder. These networks were used to extract temporal features from historical observations and weather forecasts independently, and the resulting outputs were concatenated to form the backbone output. The results in Table 8 demonstrate that by incorporating the proposed spatial-temporal adaptation (STA) formulation, all backbone networks achieved improved coldstart performance. Different architectures show similar performance, and transformer decoder slightly outperforms others.

4.2.2. Parameter Study. How does result change under different prediction ranges? It is intuitive that adaptive prediction can better fit mid-term or long-term target patterns and thus bring larger benefits when a longer prediction length is required. We empirically verify it by testing our model at different prediction lengths between 1 and 24 hours. In Figure 12(a), we draw the prediction MAE of our model compared to the backbone network with respect to different settings of L_p for all four pollutants. We find that our model consistently outperforms the baseline backbone network. The performance gap gets larger at longer prediction interval lengths like 24 hours.

How much data do we need at target city? We are also interested in how much data we need to have good coldstart prediction performance, in sense of both source city data used for meta-training and target city data used for inference prompting. In Figure 12(b), we report the relative performance change of our model under different lengths of available target city data. We test the same model listed in Table 4 (trained with 200 hours support length) but only input with fewer target support data, varying from 1 day to roughly 9 days (specifically, 40, 50, 60, 70, 80, 90, 100, 125, 150, 175, and 200 hours). We calculate and plot the relative improvement (MAE reduction normalized by baseline) compared to the non-adaptively pretrained backbone network. Note that this figure is in logarithmic axis. While pretraining and fine-tuning have shown limited transfer improvement even with 9 days target data (refer to Table 4), the proposed model can provide positive transfer improvements with only 3 days target data (2 days for NO_2). When target data are at extremely low level, our model may also get worse than pretrained backbone network. In such case, the available target city data are too scarce to support accurate target city pattern inference, and directly using pretrained network in a zero-shot style is recommended.

How does source city coverage influence coldstart result? In Figure 12(c), we report the prediction MAE of our model, compared to the non-adaptive backbone network and two representative meta-learners MAML and CNP, when only using a subset of available source cities in the meta-training phase. We train the model with 10, 20, and 30 source cities on the SO_2 dataset while keeping all other settings unchanged (e.g., still test model on the same 10 leave-out target cities). We find that our model successfully enables backbone network's adaptive prediction ability and improves its MAE by 5.2%, 4.0%, and 3.9%, respectively, for 10, 20, and 30 source cities. The absolute error decreases as more source cities are available, i.e., coldstart performance benefits from larger coverage of source cities. We attribute the phenomenon that our model has slightly better relative performance at fewer source cities to larger distribution difference caused by limited source dataset coverage. Our STA formulation can better handle the data distribution shift than pretrained models that do not explicitly consider distribution structures and differences. The result also shows that our model can consistently outperform two meta-learner baselines.

4.3. Limitations. The proposed coldstart algorithm has certain limitations. Firstly, this method assumes that the pattern distribution during training and testing is the same. However, if the target city pattern differs greatly from the training source and has not occurred before, the algorithm's performance may be poor. Note that this same pattern



(a)

(b)

(c)

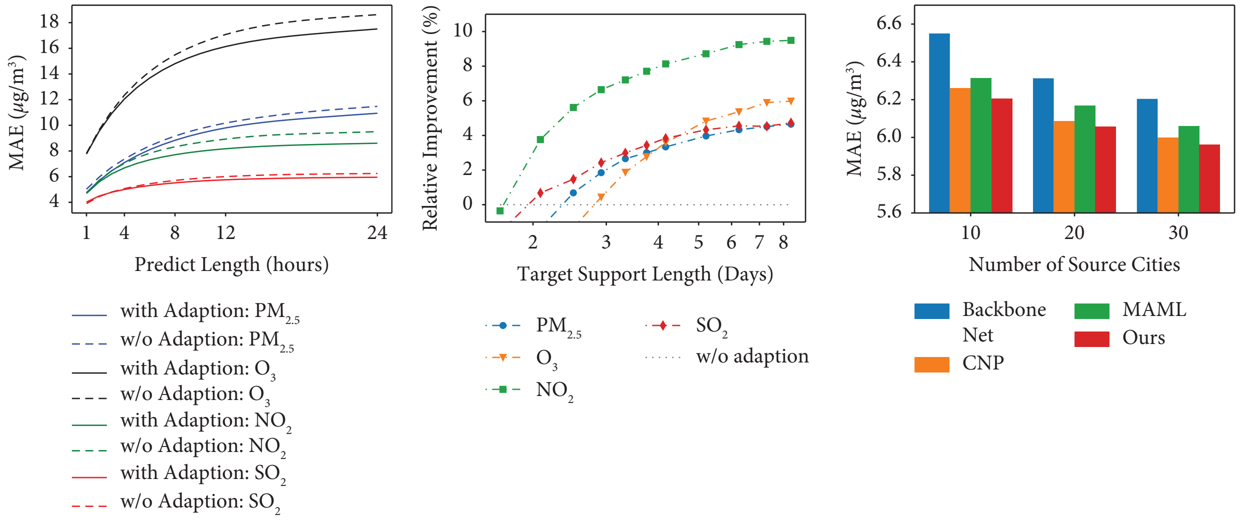
Figure 11: (a) Parameter study on the update step length of the backward feature. We fit the mean of the result to a second-order polynomial for better reference (shown in the dashed curve). (b) Parameter study on the update step number of the backward feature. (c) Parameter study on task parameter variance. The working point denotes the default parameter setting of all other experiments.

Table 7: Module ablation study on air dataset.

Pollutants	MAE (μg/m^3)		
	Ours-FF	Ours-BF	Ours
$PM_{2.5}$	11.199 ± 0.188	11.214 ± 0.100	11.043 ± 0.092
O_3	17.636 ± 0.127	18.276 ± 0.218	17.595 ± 0.069
NO_2	8.768 ± 0.087	8.907 ± 0.083	8.671 ± 0.057
SO_2	6.029 ± 0.048	5.984 ± 0.052	5.961 ± 0.015

Table 8: Backbone ablation study on air dataset.

Backbone network	MAE (μg/m^3)	
	W/o STA	With STA (ours)
MLP	11.420 ± 0.168	11.043 ± 0.092
CNN	11.447 ± 0.124	11.146 ± 0.144
LSTM	11.554 ± 0.185	11.030 ± 0.122
Transformer	11.567 ± 0.170	11.008 ± 0.129



(a)

(b)

(c)

Figure 12: Task parameter studies on air pollution dataset. (a) Prediction MAE for different prediction lengths. (b) Relative MAE improvement of our method compared to non-adaptively pretrained backbone, under different lengths of target support data. (c) Prediction MAE on SO_2 dataset for a different number of source cities.

distribution assumption between source and target is independent from the non-i.i.d modeling of source data distribution. Secondly, while the algorithm does not require direct training of a target distribution, but instead recognizes it as a source component, there appears to be a trade-off between the resolution of the finer structure of the predictive distribution and the size of the spatiotemporal window, as well as the number of support samples. In our experiments, we preset these values as hyperparameters empirically, but further theoretical analysis remains an interesting topic in future work. Thirdly, as the proposed model is a model-agnostic framework, we only test temporal predictor as the backbone network in our experiment because there is only one fixed monitoring site for each city in our dataset. While this aligns with the common coldstart scenario that the target city may only have one newly built monitoring station [8], we acknowledge that the test over spatiotemporal backbone networks can be further investigated when sensory network data for individual cities become available.

5. Conclusions

The development of accurate data-driven air pollution prediction systems is crucial to mitigate the harmful effects of air pollution. However, predicting air pollution in a new city with extremely limited data accumulation remains a significant challenge. Traditional transfer learning solutions may not be satisfying due to the insufficient usage of available source data and suboptimal transferring strategy. To address this problem, we propose formulating the air pollution prediction task as a STA problem. This involves decomposing the source dataset into a mixture of spatial-temporal-specific subdistributions and learning to adapt across space and time. By doing so, it is possible to significantly reduce the data accumulation requirement for the new city and improve coldstart prediction performance. We further propose an effective bidirectional meta-learner and derive a coldstart training algorithm based on them. Results from both synthetic and real-world air pollution datasets demonstrate the effectiveness of our approach. Our proposed method outperforms pretraining and fine-tuning solutions by 5.2% in 24-hour prediction mean absolute error (MAE) when only 200 hours of data are available for a new city.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the Shenzhen Science and Technology Program under grant no. KQTD20170810150821146 and the Tsinghua-Toyota Joint Research Institute Cross-Discipline Program.

References

[1] W. H. Organisation, *Ambient Air Pollution: A Global Assessment of Exposure and burden of Disease*, World Health Organization, Geneva, Switzerland, 2016.

[2] A. Masood and K. Ahmad, "A review on emerging artificial intelligence (ai) techniques for air pollution forecasting: fundamentals, application and performance," *Journal of Cleaner Production*, vol. 322, Article ID 129072, 2021.

[3] Z. Luo, J. Huang, K. Hu, X. Li, and P. Zhang, "Accuair: winning solution to air quality prediction for kdd cup 2018," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1842–1850, Anchorage, AK, USA, June 2019.

[4] B. Zhang, Y. Rong, R. Yong et al., "Deep learning for air pollutant concentration prediction: a review," *Atmospheric Environment*, vol. 45, Article ID 119347, 2022.

[5] M. Wang and W. Deng, "Deep visual domain adaptation: a survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[6] M. De Lange, R. Aljundi, M. Masana et al., "A continual learning survey: defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2022.

[7] J. Wang, C. Lan, C. Liu et al., "Generalizing to Unseen Domains: A Survey on Domain Generalization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 45, 2022.

[8] J. Ma, Z. Li, J. C. Cheng, Y. Ding, C. Lin, and Z. Xu, "Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network," *The Science of the Total Environment*, vol. 705, Article ID 135771, 2020.

[9] I. H. Fong, T. Li, S. Fong, R. K. Wong, and A. J. Tallon-Ballesteros, "Predicting concentration levels of air pollutants by transfer learning and recurrent neural network," *Knowledge-Based Systems*, vol. 192, Article ID 105622, 2020.

[10] Y. Zhang, Q. Lv, D. Gao et al., "Multi-group encoder-decoder networks to fuse heterogeneous data for next-day air quality prediction," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}*, pp. 4341–4347, Macao, China, July 2019.

[11] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: a meta-learning approach for spatial-temporal prediction," in *Proceedings of the The World Wide Web Conference*, pp. 2181–2191, New York, NY, USA, August 2019.

[12] Y. Zhang, Y. Li, X. Zhou, J. Luo, and Z.-L. Zhang, "Urban traffic dynamics prediction—a continuous spatial-temporal meta-learning approach," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 2, pp. 1–19, 2022.

[13] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: a survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[14] Y. Du, J. Wang, W. Feng et al., "Adarnn: adaptive learning and forecasting of time series," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 402–411, Queensland, Australia, November 2021.

[15] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018, https://arxiv.org/abs/1801.06146.

[16] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks: a survey," ieee transactions on pattern analysis and machine intelligence," 2021, https://arxiv.org/abs/2004.05439.

[17] M. Garnelo, D. Rosenbaum, C. Maddison et al., "Conditional neural processes," in *Proceedings of the International*

Conference on Machine Learning, pp. 1704–1713, PMLR, Baltimore, MA, USA, June 2018.

[18] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference On Machine Learning, Ser. Proceedings of Machine Learning Research*, D. Precup and Y. W. Teh, Eds., pp. 1126–1135, Baltimore, MA, USA, August 2017.

[19] T.-D. Hoang, N. M. Ky, N. T. N. Thuong, H. Q. Nhan, and N. V. C. Ngan, “Artificial intelligence in pollution control and management: status and future prospects,” *Artificial Intelligence and Environmental Sustainability*, vol. 34, pp. 23–43, 2022.

[20] Z. Wu, Y. Wang, and L. Zhang, “Msstn: multi-scale spatial temporal network for air pollution prediction,” in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, June 2019.

[21] Y. Zheng, X. Yi, M. Li et al., “Forecasting fine-grained air quality based on big data,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2267–2276, Singapore, December 2015.

[22] P. Zhao and K. Zettsu, “Mastgn: multi-attention spatio-temporal graph networks for air pollution prediction,” in *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, pp. 1442–1448, Manhattan, New York, USA, July 2020.

[23] G. T. Wilson, “Time series analysis: forecasting and control,” in *Journal of Time Series Analysis*, john wiley and sons inc, Hoboken, NY, USA, 5 edition, 2016.

[24] L. Zhang, J. Lin, R. Qiu et al., “Trend analysis and forecast of pm2. 5 in fuzhou, China using the arima model,” *Ecological Indicators*, vol. 95, pp. 702–710, 2018.

[25] M. Dong, D. Yang, Y. Kuang, D. He, S. Erdal, and D. Kenski, “Pm2. 5 concentration prediction using hidden semi-markov model-based times series data mining,” *Expert Systems with Applications*, vol. 36, no. 5, pp. 9046–9055, 2009.

[26] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 3104–3112, 2014.

[27] A. Tiwari, R. Gupta, and R. Chandra, “Delhi Air Quality Prediction Using Lstm Deep Learning Models with a Focus on Covid-19 Lockdown,” 2021, https://arxiv.org/abs/2102.10551.

[28] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, “Geoman: multi-level attention networks for geo-sensory time series prediction,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3428–3434, Stockholm, Sweden, June 2018.

[29] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *Proceedings of the International Conference on Machine Learning*, pp. 1243–1252, Baltimore, MA, USA, June 2017.

[30] H. Zhou, S. Zhang, J. Peng et al., “Informer: beyond efficient transformer for long sequence time-series forecasting,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11106–11115, 2021.

[31] B. Liu, X. Yu, J. Chen, and Q. Wang, “Air pollution concentration forecasting based on wavelet transform and combined weighting forecasting model,” *Atmospheric Pollution Research*, vol. 12, no. 8, Article ID 101144, 2021.

[32] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, “Deep distributed fusion network for air quality prediction,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 965–973, New York, NY, USA, August 2018.

[33] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, August 2018.

[34] A. Alléon, G. Jauvion, B. Quennehen, and D. Lissmyr, “Plumenet: Large-Scale Air Quality Forecasting Using a Convolutional Lstm Network,” 2020, https://arxiv.org/abs/2006.09204.

[35] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[36] J. Toutouh, “Conditional generative adversarial networks to model urban outdoor air pollution,” in *Proceedings of the Ibero-American Congress of Smart Cities*, pp. 90–105, Berlin, Germany, May 2020.

[37] Y. Lin, N. Mago, Y. Gao et al., “Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning,” in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 359–368, Seattle, WA, USA, November 2018.

[38] J. Xu, L. Chen, M. Lv, C. Zhan, S. Chen, and J. Chang, “Highair: a hierarchical graph neural network-based air quality forecasting method,” 2021, https://arxiv.org/abs/2101.04264.

[39] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, “Spatio-temporal graph transformer networks for pedestrian trajectory prediction,” in *Proceedings of the European Conference on Computer Vision*, pp. 507–523, Berlin, Germany, May 2020.

[40] S. P. Arango, F. Heinrich, K. Madhusudhanan, and L. Schmidt-Thieme, “Multimodal meta-learning for time series regression,” in *Proceedings of the International Workshop on Advanced Analytics and Learning on Temporal Data*, pp. 123–138, Berlin, Germany, August 2021.

[41] T. Li, X. Su, W. Liu et al., “Memory-augmented meta-learning on meta-path for fast adaptation cold-start recommendation,” *Connection Science*, vol. 34, no. 1, pp. 301–318, 2022.

[42] S. Tariq, J. Loy-Benitez, K. Nam et al., “Transfer learning driven sequential forecasting and ventilation control of pm2. 5 associated health risk levels in underground public facilities,” *Journal of Hazardous Materials*, vol. 406, Article ID 124753, 2021.

[43] Z. Wu, C. Ma, X. Shi et al., “Brnn-gan: generative adversarial networks with bi-directional recurrent neural networks for multivariate time series imputation,” in *Proceedings of the 2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 217–224, Beijing, China, December 2021.

[44] Z. Wu, C. Ma, X. Shi, L. Wu, Y. Dong, and M. Stojmenovic, “Imputing missing indoor air quality data with inverse mapping generative adversarial network,” *Building and Environment*, vol. 215, Article ID 108896, 2022.

[45] S. Thrun and L. Pratt, “Learning to learn: introduction and overview,” *Learning to Learn*, vol. 78, pp. 120–193, 1998.

[46] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, “Online meta-learning,” in *Proceedings of the International Conference on Machine Learning*, pp. 1920–1930, Baltimore, MA, USA, September 2019.

[47] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, “Learning to generalize: meta-learning for domain generalization,”

Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.

[48] J. R. Schwarz and Y. W. Teh, "Meta-learning sparse compression networks," 2022, https://arxiv.org/abs/2205.08957.

[49] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 1842–1850, Baltimore, MA, USA, May 2016.

[50] H. Kim, A. Mnih, J. Schwarz et al., "Attentive neural processes," 2019, https://arxiv.org/abs/1901.05761.

[51] C. H. Kao, W.-C. Chiu, and P.-Y. Chen, "Maml is a noisy contrastive learner in classification," in *Proceedings of the International Conference on Learning Representations*, Addis Ababa, Ethiopia, August 2021.

[52] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017, https://arxiv.org/abs/1703.05175.

[53] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, Long Beach, CA, USA, June 2019.

[54] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, https://arxiv.org/abs/1803.02999.

[55] A. A. Rusu, D. Rao, J. Sygnowski et al., "Meta-learning with latent embedding optimization," 2018, https://arxiv.org/abs/1807.05960.

[56] J. Xu, J.-F. Ton, H. Kim, A. Kosiorek, and Y. W. Teh, "Metafun: meta-learning with iterative functional updates," in *Proceedings of the International Conference on Machine Learning*, Beijing, China, November 2020.

[57] T. Deleu, D. Kanaa, L. Feng et al., "Continuous-time meta-learning with forward mode differentiation," 2022, https://arxiv.org/abs/2203.01443.

[58] H. Edwards and A. Storkey, "Towards a neural statistician," 2016, https://arxiv.org/abs/1606.02185.

[59] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," 2018, https://arxiv.org/abs/1801.08930.

[60] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," in *Proceedings of the 32nd International Conference On Neural Information Processing Systems*, pp. 9537–9548, Red Hook, NY, USA, June 2018.

[61] S. Ravi and A. Beatson, "Amortized bayesian meta-learning," in *Proceedings of the International Conference on Learning Representations*, Addis Ababa, Ethiopia, May 2018.

[62] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7343–7353, Red Hook, NY, USA, December 2018.

[63] Z. Pan, W. Zhang, Y. Liang et al., "Spatio-temporal meta learning for urban traffic prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, 2020.

[64] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola, "Deep sets," 2017, https://arxiv.org/abs/1703.06114.

[65] X. Zhang, Y. Li, X. Zhou et al., "Dac-ml: domain adaptable continuous meta-learning for urban dynamics prediction," in *Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM)*, pp. 906–915, Shanghai, China, June 2021.

[66] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: a framework for attention-based permutation-invariant neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 3744–3753, Long Beach, CA, USA, June 2019.

[67] R. J. Santos, "Equivalence of regularization and truncated iteration for general ill-posed problems," *Linear Algebra and Its Applications*, vol. 236, pp. 25–33, 1996.

[68] N. Gao, H. Ziesche, N. A. Vien, M. Volpp, and G. Neumann, "What matters for meta-learning vision regression tasks?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14776–14786, Washington, DC, USA, November 2022.

[69] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.