NPDiff: A General Noise Prior Framework for Diffusion Model-based Mobile Traffic Prediction

Anonymous Author(s)*

Abstract

Accurate prediction of mobile traffic, which refers to the network traffic from cellular base stations, is crucial for optimizing network performance and enhancing user experience on web-based services. However, mobile traffic data is highly non-stationary with fluctuations driven by human activity and environmental changes, resulting in both regular patterns and abrupt variations. Diffusion models have shown great promise in modeling such complex temporal dynamics due to their ability to capture the inherent uncertainties. Most existing approaches focus on designing novel denoising networks with conditions but overlook the critical role of noise itself, which may result in sub-optimal performance. In this paper, we introduce a novel perspective by emphasizing the role of noise in the denoising process. Through an in-depth analysis of the noise, we demonstrate that it fundamentally shapes the prediction outcomes of mobile traffic, exhibiting clear and consistent patterns. To leverage this, we propose NPDiff, a general framework that incorporates the intrinsic dynamics of mobile traffic data as noise priors. By decomposing the noise into *prior* and *residual* parts, NPDiff enhances the model's ability to capture both regular patterns and complex, abrupt variations. Our approach is a general framework that can seamlessly integrate with various state-of-the-art diffusion-based prediction models, boosting prediction accuracy and accelerating training convergence. Extensive experiments demonstrate that NPDiff achieves superior performance with an improvement over 30%, offering a new perspective on leveraging diffusion models in this domain. Code and data are available at https://anonymous.4open.science/r/WWW25-submission-1670.

1 Introduction

The widespread adoption of web services and mobile devices has led to a substantial increase in mobile data usage, which constitutes a significant portion of overall web traffic [34, 49, 56]. As mobile services and web-based applications become ubiquitous in everyday activities, managing the resulting data traffic has become increasingly complex. Mobile traffic prediction [12, 40, 43] aims to forecast the traffic volume of cellular-based stations, which is essential for optimizing network performance and improving user experience [13, 44]. This capability is crucial for both service providers aiming to align network capacity with user demand and for the development of smart city infrastructure, where accurate prediction can support dynamic network management, improve energy efficiency, and enhance overall connectivity.

Mobile traffic prediction involves forecasting temporal data, similar to time series and spatio-temporal data prediction. Traditional deep learning methods—such as CNNs [24, 27, 60], RNNs [26, 46, 47], GCNs [1, 15, 64], and transformers [7, 8, 22, 57]—have achieved notable success in this area, often employing a deterministic approach that directly maps inputs to outputs. Recently, diffusion models [18, 30] have emerged as a promising alternative. Unlike traditional models, they offer a probabilistic framework that captures the intricate distributions inherent in temporal data, effectively handling uncertainties and complexities. For time series and spatio-temporal data, most diffusion model-based solutions [37, 42, 55, 58] naively adopt the noise modeling in the image domain and design specialized denoising networks. This noise is typically modeled as independent and identically distributed (i.i.d.) Gaussian noise, which guides the denoising process step-by-step.

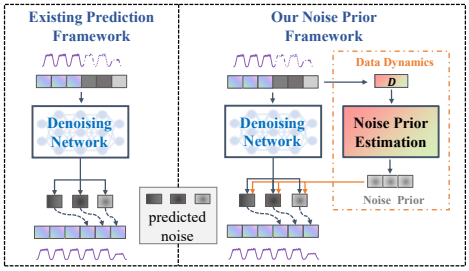


Figure 1: Comparison of existing diffusion-based prediction framework with our noise prior framework.

However, mobile traffic data exhibits unique dynamics that differ significantly from those in the image domain. Originating from human movements and activities, it naturally reflects human rhythms, with temporal dynamics often showing periodicity and abrupt changes [38, 53, 54]. Existing diffusion model-based solutions mainly focus on designing innovative denoising networks or conditioning mechanisms to incorporate various features [37, 42, 58]. While conditioning features for the denoising network can provide some context for predictions, the noise learned in the denoising process step by step plays a more critical role [14, 61]—it directly shapes the model's generative process, significantly influencing the quality and accuracy of predictions. Despite its importance, the impact of noise in the denoising process for temporal remains largely unexplored.

Therefore, we are motivated to address a simple but essential question: can we reconstruct noise in the denoising process that preserves inherent characteristics of mobile traffic data? However, it is not a trivial task. First, the assumption that the noise to be predicted is Gaussian provides essential properties that ensure the stability and convergence of diffusion models. Manipulating this noise requires preserving these characteristics to maintain the model's reliability. Second, mobile traffic data is highly non-stationary, exhibiting fluctuations driven by diverse factors like human activity patterns and environmental changes. As a result, the model must capture regular, predictable patterns while remaining sensitive to sudden changes.

To address these challenges, we introduce NPDiff, a novel **N**oise **P**rior framework for **Diff**usion models in mobile traffic prediction to leverage the patterns of data dynamics as priors to reconstruct noise. The key idea is to decompose the noise into two components.

The first component, ***noise prior***, captures the intrinsic dynamics of mobile traffic data, including periodic patterns and local variations, which provides a basic reference for the noise estimation in each step. The second component, ***noise residual***, accounts for the additional variations that the noise prior cannot fully represent, enabling the model to adapt to more complex and unpredictable aspects of the data. Based on theoretical derivations, we can successfully reconstruct mobile traffic data through the ~~a one-step~~ backward diffusion process using the noise prior and the noisy data. As a general solution, NPDiff can be seamlessly integrated into existing diffusion-based models, boosting their prediction performance and adaptability to the unique characteristics of mobile traffic data.

In summary, our contributions are as follows:

- To our best knowledge, we are the first to emphasize the role of noise in diffusion models for mobile traffic prediction, offering new insights for applying diffusion models in this domain.
- We propose NPDiff, a general framework for diffusion-based mobile traffic prediction that leverages the intrinsic dynamics of mobile traffic data as noise priors. It can be seamlessly integrated into existing state-of-the-art diffusion-based models.
- Extensive experiments across various prediction tasks and different denoising networks demonstrate that NPDiff significantly improves prediction accuracy by over 30%. Additionally, NPDiff enhances training efficiency with faster convergence, improves robustness, and reduces prediction uncertainty, positioning it as an efficient and versatile enhancement to current methodologies.

2 Related Work

2.1 Mobile Traffic Prediction

Mobile traffic prediction is crucial for managing modern networks, attracting extensive research [21]. Traditional time series methods like ARIMA [9] leverage temporal correlations but struggle with non-linear traffic patterns. Recent advances in deep learning, such as RNNs and LSTMs [10, 43], have improved temporal modeling but still focus mainly on sequence patterns. CNN-based models [25, 36] address spatial aspects by treating traffic data as images, but they often overlook distant spatial dependencies. To tackle non-Euclidean spatial relationships, multiple graph-based approaches [16, 17, 50, 65] have been proposed, enabling a more accurate representation of complex spatial interactions. In addition, Transformer-based models [20, 28] have emerged as a powerful tool for mobile traffic prediction due to their ability to capture long-range dependencies and parallelized computation. Moreover, diffusion-based methods, which have already shown remarkable success in traffic data generation, have also attracted attention in the traffic prediction domain [3, 4, 42]. By treating the prediction task as a generative process, diffusion-based methods [3] offer a novel approach to improve prediction accuracy, especially when carefully designed to adapt to the specific requirements of forecasting tasks.

2.2 Diffusion Models for Temporal Data

Diffusion models [18, 30] have recently become popular for time series [37, 42] and spatio-temporal data [51, 58, 59] due to their strong generative capabilities. They work through a two-step process: noise injection and noise removal, allowing them to model complex data distributions. Unconditioned models like DDPMs [18] operate without extra information, while conditioned models utilize additional data, such as historical observations [37] or knowledge graphs [66], to guide the denoising process. Diffusion models offer a powerful alternative for capturing the intricate dynamics of time series data, surpassing traditional methods in flexibility and generative quality. Currently, diffusion models are primarily applied to two tasks: generation and prediction [3, 42, 51]. The generation task aims to create new data, filling in regions with scarce data, while the prediction task focuses on forecasting future values based on historical data. The data generated can also be used to train prediction models, enhancing their performance by providing additional training samples. In temporal data prediction, most diffusion-based approaches [37, 42, 51, 55, 58, 59] focus on conditioning to guide the generative denoising process, with limited attention to optimizing the noise component itself, despite its critical role in shaping model performance. Our work addresses this gap by investigating the impact of noise and introducing a general noise prior framework, leading to more accurate temporal data prediction.

2.3 Noise Manipulation in Video Models

In video generation tasks, diffusion-based methods [2, 19, 32, 45] have made significant progress, but generating high-quality videos with frame-to-frame consistency remains a core challenge [5, 52]. Recently, many studies [5, 14, 33, 61] have enhanced the content coherence of video generation by manipulating noise during the diffusion process. Specifically, some studies sample similar noise for each video frame in the early stages of diffusion [14, 33]. Additionally, other research introduces new forms of noise representation [5]. Moreover, some ~~Some~~ researchers use historical static images as references to provide static noise prior for the diffusion process [61]. Compared with video data that requires temporal consistency, mobile data are much more dynamic with complex spatio-temporal correlations. How to design effective noise priors for such data remains unexplored yet is a promising direction. This paper is the first to pioneer this exploration in the context of mobile traffic data.

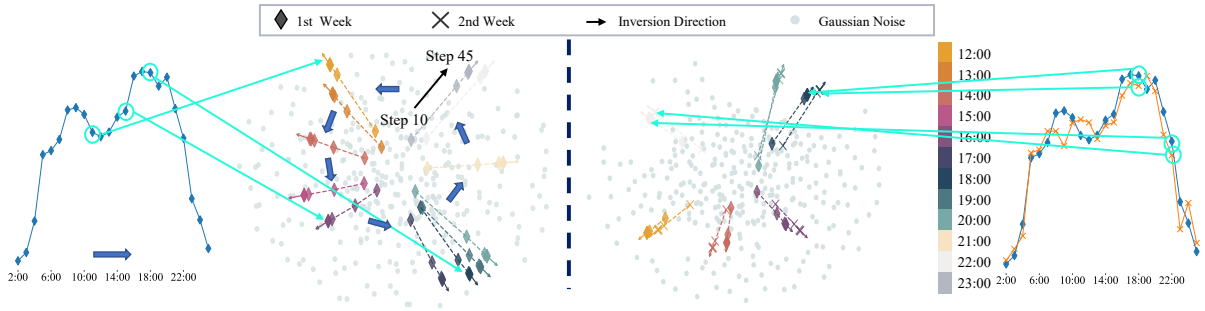
3 Preliminaries

3.1 Problem Formulation

Mobile traffic prediction utilizes historical data to predict future traffic flows. The format of mobile traffic data is typically characterized as a three-dimensional tensor $X \in \mathbb{R}^{T\times K\times C}$, where T represents the temporal length of the time series, K represents the number of spatial locations, and C represents traffic flow features. Specifically, our task is to learn a predictive model $\mathcal{F}$, which, given historical context data $X^{co} = X^{t-H+1:t}$ of length H, predicts the traffic flow $X^{ta} = X^{t+1:t+M}$ for the next M steps, which can be represented as $X^{ta} = \mathcal{F}(X^{co})$.

3.2 Diffusion-based Spatio-Temporal Prediction

For spatio-temporal prediction tasks, the diffusion-based models predict target values conditioned on historical data x_0^{co}. During the forward process, noise is progressively added to the target data with the noise schedule $\{\beta_n\}_{t=1}^{N}$. The noised target at any diffusion



(a) Visualization of Noise Maps

(b) Comparison of the Same Time Points across Two Weeks

Figure 2: Visualization of the noise on the MobileNJ dataset. Figure (a) shows the noise in 12 consecutive time points during the denoising process, encompassing five distinct diffusion steps from step 10 to step 45. Figure (b) compares the noise of the same timestamps across two consecutive weeks.

step can be calculated using the formula below:

$$x_n^{ta} = \sqrt{\bar{\alpha}_n} x_0^{ta} + \sqrt{1-\bar{\alpha}_n}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

where $\alpha_n = 1 - \beta_n$ and $\bar{\alpha}_n = \prod_{i=1}^{n} \alpha_i$. Reversely, the model starts by sampling x_N^{ta} from the standard noise distribution, conditioned on historical data, and progressively denoises the data over N steps to produce the final prediction. This reverse process is defined by a Markov chain:

$$\begin{aligned} p_\theta(x_{0:N}^{ta}) &:= p(x_N^{ta}) \prod_{n=1}^{N} p_\theta(x_{n-1}^{ta} | x_n^{ta}), \\ p_\theta(x_{n-1}^{ta} | x_n^{ta}) &:= \mathcal{N}(x_{n-1}^{ta}; \mu_\theta(x_n^{ta}, n | x_0^{co}), \Sigma_\theta(x_n^{ta}, n)), \\ \mu_\theta(x_n^{ta}, n | x_0^{co}) &= \frac{1}{\sqrt{\bar{\alpha}_n}} \left(x_n^{ta} - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}} \epsilon_\theta(x_n^{ta}, n | x_0^{co}) \right). \end{aligned} \tag{2}$$

where the variance $\Sigma_\theta(x_n^{ta}, n)$ is a fixed constant. Therefore, during the denoising process, the model only need to compute $\mu_\theta(x_n^{ta}, n | x_0^{co})$, the only unknown term is $\epsilon_\theta(x_n^{ta}, n | x_0^{co})$. The model is trained to estimate the noise vector $\epsilon - \theta$, which is added to the noised data $x_{0:N}^{ta}$. The model's parameters θ are optimized by solving the optimization problem below:

$$\min_\theta \mathcal{L}(\theta) = \min_\theta \mathbb{E}_{n, x_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(x_n^{ta}, n | x_0^{co}) \right\|_2^2 \right]. \tag{3}$$

4 Observations and Analysis

To better understand and demonstrate the role of noise in the denoising process, we visualize the estimated noise for different diffusion steps and timestamps using MobileNJ dataset (see Section 6.1 for more details about the dataset). Specifically, we employ t-distributed Stochastic Neighbor Embedding (t-SNE) and we take the inversion process of Denoising Diffusion Implicit Models (DDIM) as an example. We plot the results in Figure 2, which denote the noise estimated by the model during this process. The small dots in the background represent Gaussian noise, serving as a reference noise. Figure 2(a) illustrates that, for the same time point, the noise across different diffusion steps exhibits high continuity, approximately following a linear trend. This phenomenon suggests that noise at every diffusion step is constrained by certain inherent conditions. Additionally, the noise shows a counterclockwise rotation trend, aligning with the temporal dimension.

In Figure 2(b), we further observe that the noise from the same time point one week later highly overlaps with those from the current time point, revealing clear periodic characteristics. This periodicity is highly consistent with the inherent dynamics of mobile traffic data, indicating a strong connection between temporal dynamics and noise within diffusion process.

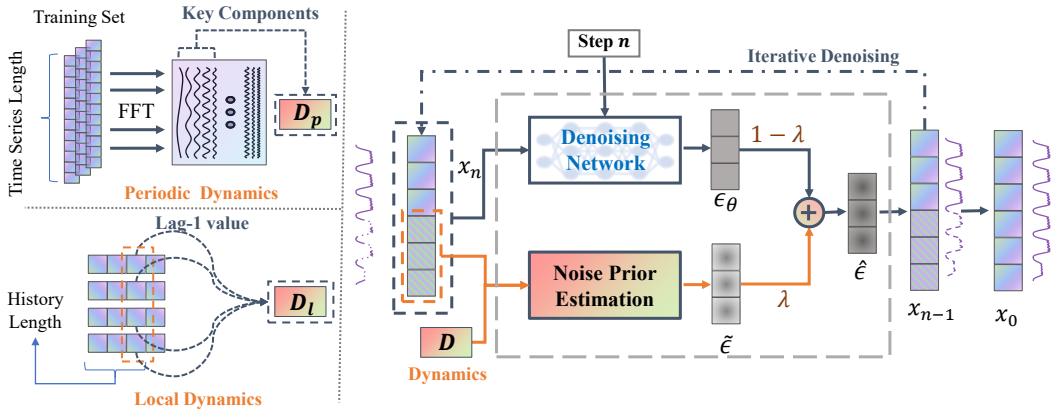
These findings underscore the critical role of noise in the denoising process, as the noise exhibits clear and consistent patterns. Consequently, integrating the intrinsic dynamics of temporal data into the modeling of noise is well-motivated and promising for enhancing the model's effectiveness.

5 Method

Recognizing the crucial role of the noise, we introduce the NPDiff framework, which incorporates the mobile traffic dynamics as noise prior to the diffusion process. Figure 3 illustrates an overview of our approach. Figure 3(a) shows the extraction of two key dynamics inherent in mobile traffic data: periodic dynamics and local dynamics, while Figure 3(b) illustrates the denoising process with noise priors. In summary, our approach computes the final noise $\widehat{\epsilon}$ through two pathways. The first path involves calculating the noise prior, which serves as a basic reference noise. The second path predicts the residual noise using the denoising network. The final noise is then obtained by performing a weighted fusion of these two components.

5.1 Dynamics of Mobile Traffic Data

By analyzing the dynamics of mobile traffic data, we select two representative dynamics to calculate noise priors, which we define as periodic dynamics and local dynamics, respectively. Periodic dynamics capture regular and predictable patterns, while local dynamics reflect subtle yet abrupt changes. We calculated the cosine similarities between these dynamics and actual values. As shown in Table 1, they both achieve high cosine similarities of around 0.8 across all datasets.



(a) Extraction of Data Dynamics (b) The Denoising Process

Figure 3: Overall architecture of the NPDiff. Figure (a) shows the process of extracting data dynamics, and Figure (b) illustrates the denoising process with noise priors.

Periodic Dynamics. To capture periodic dynamics, we leverage Fast Fourier Transform (FFT), which is widely used in time series analysis. Specifically, we select several key components obtained by using FFT and convert them back to the time domain. This process can be formulated as follows:

$$A_{(k)} = |\text{FFT}(x)_k|, \quad \Phi_{(k)} = \phi\left(\text{FFT}(x)_k\right),$$
$$\kappa^{(1)}, \cdots, \kappa^{(K)} = \underset{k \in \{1, \cdots, \lfloor \frac{L}{2} \rfloor + 1\}}{\text{arg-TopK}} \left\{A_{(k)}\right\},$$
$$S[i] = \sum_{k=1}^{N_K} A_{\kappa^{(k)}} \left[\cos\left(2\pi f_{\kappa^{(k)}} i + \Phi_{\kappa^{(k)}}\right) + \cos\left(2\pi \bar{f}_{\kappa^{(k)}} i + \bar{\Phi}_{\kappa^{(k)}}\right)\right], \tag{4}$$

where $A_{(k)}$ is the amplitude of the $k-$th frequency component obtained by applying FFT to the training data x, and $\Phi_{(k)}$ represents the phase of the $k-$th frequency component. L represents the length of the input sequence, which is the length of the time series from the training set. $\kappa^{(1)}, \cdots, \kappa^{(K)}$ are the the top K amplitudes selected by arg-TopK. N_K is the number of selected frequency components used for computing the periodic dynamics. $f_{\kappa^{(k)}}$ refers to the frequency corresponding to the k-th component, and $\bar{f}_{\kappa^{(k)}}, \bar{\Phi}_{\kappa^{(k)}}$ represent the corresponding conjugates.

We then compute the average values of the time-domain signal at corresponding time points across different cycles within a specified period length P to extract the final dynamics. This processing can be described by the following mathematical expression:

$$D_p[t] = \frac{1}{N_P} \sum_{k=0}^{N_P - 1} S[t + kP], \tag{5}$$

where N_P denotes the total number of complete periods in the training data, while $t + kP$ represents the sampled value at time t within each period P. we extend periodic dynamics $D_p[t]$ within period P to the entire dataset to obtain the dynamics corresponding to each time point. The period P is defined as one week in most datasets, while P is set to one day for MobileBJ dataset due to the limited data duration. If the dataset is sufficiently large, P can be chosen to represent longer periods, such as a month or a year, which correspond to the behavioral patterns of human activity.

Table 1: Cosine similarities between extracted dynamics and target values.

	MobileBJ	MobileNJ	MobileSH14	MobileSH16
Periodic Dynamics	0.889	0.876	0.778	0.763
Local Dynamics	**0.923**	**0.905**	**0.826**	**0.825**

Local Dynamics. Periodic dynamics reflect the periodic patterns, but they may not be well-suited for capturing short-term variations. Mobile traffic data usually depend on consecutive time steps, the current values are strongly correlated with the nearby last step. Therefore, we incorporate local dynamics into consideration. Specifically, we utilize the lag-1 values to depict local variations. Mathematically, this can be described as:

$$D_l[t] = x[t-1]. \tag{6}$$

5.2 Derivation and Fusion of Noise Priors

As analyzed in Section 4, there is a strong correlation between noise and data dynamics. Therefore, our objective is to enhance the model's performance by leveraging these correlations. In this section, we describe the method for calculating noise prior and explain how it is integrated into the denoising process. For a detailed explanation of the diffusion process and the use of noise priors in the diffusion process, please refer to Figure 10.

Derivations of Noise Priors. Given the extracted periodic and local dynamics, a basic assumption is that they are closely aligned with the corresponding target values. Based on this assumption [61], we can reformulate the mobile traffic data as follows:

$$x_0^{ta} = D + \Delta x, \tag{7}$$

where Δx represents a small residual term that quantifies the difference between the target value and the extracted dynamics. We

then rearrange Eq. (1) to derive the formula below:

$$\epsilon = \frac{x_n^{ta} - \sqrt{\bar{\alpha}_n} x_0^{ta}}{\sqrt{1-\bar{\alpha}_n}}. \tag{8}$$

This noise ϵ serves as the target noise we need to predict during the training process. Next, we substitute Eq. (7) into Eq. (8), and obtain:

$$\begin{aligned} \epsilon &= \frac{x_n^{ta} - \sqrt{\bar{\alpha}_n}(D + \Delta x)}{\sqrt{1-\bar{\alpha}_n}} \\ &= \frac{x_n^{ta} - \sqrt{\bar{\alpha}_n} D}{\sqrt{1-\bar{\alpha}_n}} - \frac{\sqrt{\bar{\alpha}_n}}{\sqrt{1-\bar{\alpha}_n}} \Delta x. \end{aligned} \tag{9}$$

Thus, the noise prior can be defined as follows:

$$\widetilde{\epsilon} = \begin{cases} \frac{x_n^{ta} - \sqrt{\bar{\alpha}_n} D_l}{\sqrt{1-\bar{\alpha}_n}}, & \text{for single-step prediction} \\ \frac{x_n^{ta} - \sqrt{\bar{\alpha}_n} D_p}{\sqrt{1-\bar{\alpha}_n}}, & \text{for multi-step prediction} \end{cases}. \tag{10}$$

And the target noise can be written as:

$$\begin{aligned} \epsilon &= \widetilde{\epsilon} - \frac{\sqrt{\bar{\alpha}_n}}{\sqrt{1-\bar{\alpha}_n}} \Delta x \\ &= \widetilde{\epsilon} + \Delta\epsilon, \end{aligned} \tag{11}$$

where $\Delta\epsilon$ is the residual noise estimated by the denoising network.

Fusion of Noise Priors. According to Eq. (7) and Eq. (9), when the dynamics are close to the target values (i.e., Δx is small), the noise prior $\widetilde{\epsilon}$ can serve as a reference noise to ϵ. Considering that the extracted prior dynamics represent the regular patterns and may not always be accurate enough when facing situations with irregular changes, we still need the denoising network to estimate the residual noise. So we define the final noise $\widehat{\epsilon}$ as a weighted combination of the reference prior noise $\widetilde{\epsilon}$ and the model-predicted residual noise $\epsilon_\theta(x_n^{ta}, n|x_0^{co})$. This combination is controlled by an adjustable coefficient that balances the contributions of the noise prior and residual noise:

$$\widehat{\epsilon} = \lambda\widetilde{\epsilon} + (1-\lambda)\epsilon_\theta(x_n^{ta}, n|x_0^{co}), \tag{12}$$

where λ is the adjustable hyperparameter. During the diffusion process, the noise prior can be computed at any diffusion step by incorporating prior dynamics. By applying Eq. (12), the reference noise prior is combined with the model's output, allowing the integration of data dynamics by directly manipulating noise at any step of the diffusion process.

5.3 Training and Inference

Training. In the training process, we sample a random noise ϵ from a standard normal distribution and use it to transform the target data x_0^{ta} into a noisy version x_n^{ta} according to Eq. (1) and calculate the noise prior $\widetilde{\epsilon}$ based on the prior dynamics D. The model parameters θ are optimized by minimizing the loss function:

$$\min_\theta \mathcal{L}(\theta) = \min_\theta \mathbb{E}_{n,x_0,\epsilon}\left[\left\|\epsilon - (1-\lambda)\epsilon_\theta(x_n^{ta}, n|x_0^{co}) - \lambda\widetilde{\epsilon}\right\|_2^2\right]. \tag{13}$$

The steps of the training process are presented in Appendix Algorithm (1).

Inference. Algorithm (2) details the inference process of the model. First, the noisy data x_N^{ta} are randomly sampled from a standard normal distribution. Then, we estimate the noise $\widetilde{\epsilon}$ using Eq. (10), and perform denoising x_N^{ta} step by step as follows:

$$\begin{aligned} &\mu_\theta(x_n^{ta}, n|x_0^{co}) = \frac{1}{\sqrt{\bar{\alpha}_n}}\left(x_n^{ta} - \frac{(1-\lambda)\beta_n}{\sqrt{1-\bar{\alpha}_n}}\epsilon_\theta(x_n^{ta}, n|x_0^{co}) - \frac{\lambda\beta_n}{\sqrt{1-\bar{\alpha}_n}}\widetilde{\epsilon}\right), \\ &p_\theta(x_{n-1}^{ta}|x_n^{ta}) := \mathcal{N}(x_{n-1}^{ta}; \mu_\theta(x_n^{ta}, n|x_0^{co}), \Sigma_\theta(x_n^{ta}, n)). \end{aligned} \tag{14}$$

5.4 Implementation of the Denoising Network

We employ three representative models from the spatio-temporal domain as the denoising network: CSDI [42], ConvLSTM [39], and STID [35]. CSDI employs transformers as the denoising network, ConvLSTM is based on the Convolutional LSTM, and STID utilizes the multi-layer perceptron (MLP) as its backbone. These models cover the main network modules in deep learning, such as CNN, LSTM, MLP, and transformer. Utilizing these models enables a comprehensive evaluation of our method's effectiveness.

6 Evaluations

6.1 Experimental Settings

Datasets. We evaluate our method on four real-world mobile traffic datasets: MobileBJ, MobileNJ, MobileSH14, and MobileSH16, which are sourced from three major cities in China: Beijing, Nanjing, and Shanghai. Detailed descriptions of these datasets are provided in Appendix Table 6.

Baselines. We select 13 state-of-the-art models as our baselines, which are divided into three categories:

- **Classic models** include HA and ARIMA, which can be applied to time series prediction without extensive training.
- **Time series prediction models** include PatchTST [31], iTransformer [29], and Time-LLM [23], which are state-of-the-art models for multivariate time series forecasting.
- **Deep spatio-temporal prediction models** consist of two groups: *(i) urban spatio-temporal models*, including STResNet [60], ATFM [27], STNorm [11], STGSP [63], TAU [41], and PromptST [62]; *(ii) video prediction models*, including MAU [6] and MIM [48], considering that the format of mobile traffic data is similar to video data.

Experimental Configuration. For the model training, we set the maximum number of epochs to 100, accompanied by an early stopping strategy. For the validation, we sample 3 times and calculate the average result for early stopping. In the testing stage, we sample 50 times and report the average result. For baseline models, we set the maximum number of epochs to 200 for training. We provide the details of datasets, evaluation metrics, and experimental settings in Appendix A. Details of baselines can be found in Appendix B.

6.2 Overall Performance

Our experiments include both multi-step and one-step prediction tasks. The multi-step prediction evaluates capabilities of long-term prediction, while the one-step prediction task assesses the model's ability in tackling abrupt changes.

Table 2: Results of 12-12 multi-step prediction on four datasets evaluated using MAE and RMSE. The results presented in the table are obtained by averaging prediction errors across all prediction steps. Bold denotes the best results, and underline indicates the second-best results.

Model	MobileBJ		MobileNJ		MobileSH14		MobileSH16	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	0.232	0.343	0.454	0.881	0.100	0.165	13.44	38.92
ARIMA	0.236	0.404	0.327	0.692	0.112	0.197	9.15	26.70
PatchTST	0.189	0.291	0.213	0.447	0.062	0.108	10.69	28.17
iTransformer	0.154	0.249	0.205	0.436	0.045	0.072	10.19	25.91
Time-LLM	0.115	0.195	0.200	0.394	0.060	0.102	10.57	28.19
STResNet	0.546	0.751	0.439	0.624	0.102	0.138	45.63	59.82
ATFM	0.141	0.200	0.309	0.492	0.055	0.078	24.95	46.92
STNorm	0.132	0.198	0.194	0.316	0.042	0.065	11.88	28.46
STGSP	0.157	0.229	0.214	0.323	0.040	0.057	17.54	38.77
TAU	0.135	0.196	0.268	0.389	0.044	0.063	15.22	26.04
PromptST	0.099	**0.171**	0.157	0.303	0.043	0.069	9.30	23.01
MAU	0.166	0.256	0.387	0.662	0.081	0.125	21.38	45.04
MIM	0.214	0.298	0.270	0.447	0.079	0.126	22.49	47.29
CSDI	0.596	3.178	0.173	0.316	0.045	0.072	10.35	41.78
CSDI+Prior	**0.094**	0.173	**0.096**	**0.158**	**0.037**	**0.057**	**8.28**	**19.96**

Table 3: Results of 12-12 multi-step prediction on four datasets evaluated using MAE and RMSE for ConvLSTM and STID models. The results presented in the table are obtained by averaging prediction errors across all prediction steps.

Model	MobileBJ		MobileNJ		MobileSH14		MobileSH16	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
ConvLSTM	0.123	0.205	0.216	0.489	0.050	0.081	15.56	45.83
ConvLSTM+Prior	0.096	0.176	0.096	0.157	0.037	0.057	8.30	19.78
STID	0.116	0.183	0.168	0.319	0.046	0.073	9.91	27.80
STID+Prior	0.098	0.176	0.098	0.156	0.037	0.057	8.93	20.41

Table 4: Results of 12-6 multi-step prediction on four datasets evaluated using MAE and RMSE. The results presented in the table are obtained by averaging prediction errors across all prediction steps.

Model	MobileBJ		MobileNJ		MobileSH14		MobileSH16	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CSDI	0.153	0.483	0.202	0.638	0.068	0.112	9.74	29.61
CSDI+Prior	0.093	0.173	0.092	0.154	0.037	0.057	8.73	20.97
ConvLSTM	0.139	0.227	0.172	0.364	0.044	0.069	10.90	32.29
ConvLSTM+Prior	0.093	0.168	0.107	0.164	0.037	0.057	8.32	20.39
STID	0.102	0.168	0.132	0.257	0.045	0.071	8.94	21.71
STID+Prior	0.096	0.171	0.109	0.168	0.037	0.057	8.84	20.45

Multi-step Prediction. For the main experiment, we use 12 historical steps to predict the next 12 steps, which is one of the most typical setups in mobile traffic prediction. It is worth noting that different datasets have different temporal resolutions, so the time length corresponding to 12 steps differs. Table 2 shows the prediction results of NPDiff with CSDI (results for ConvLSTM and STID are shown in Appendix Table 3). Specifically, we have the following observations:

Table 5: Results of 12-1 one-step prediction on four datasets evaluated using MAE and RMSE.

Model	MobileBJ		MobileNJ		MobileSH14		MobileSH16	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CSDI	0.251	0.824	0.202	0.612	0.043	0.068	11.20	45.98
CSDI+Periodic Prior	0.090	0.167	0.099	0.160	0.037	0.057	6.78	17.25
CSDI+Local Prior	0.059	0.094	0.097	0.196	0.034	0.056	5.36	9.74
ConvLSTM	0.080	0.127	0.117	0.234	0.042	0.065	7.23	18.93
ConvLSTM+Periodic Prior	0.085	0.157	0.098	0.157	0.036	0.054	8.21	19.34
ConvLSTM+Local Prior	0.063	0.101	0.109	0.217	0.034	0.056	5.36	9.95
STID	0.077	0.125	0.096	0.168	0.040	0.062	7.02	13.73
STID+Periodic Prior	0.089	0.159	0.098	0.166	0.036	0.055	7.24	17.30
STID+Local Prior	0.061	0.095	0.088	0.163	0.033	0.055	5.48	10.27

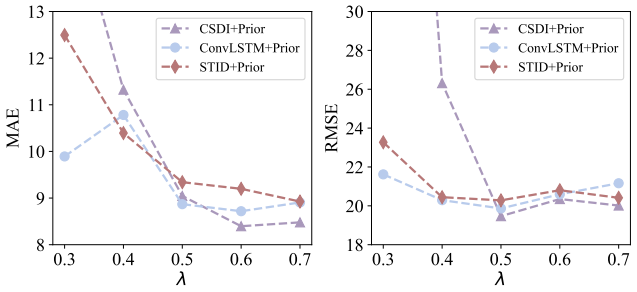


Figure 4: Ablation studies of λ on the MobileSH16 dataset.

- **Video prediction model performs poorly on mobile traffic data.** MAU and MIM, show relatively poor performance compared to other models, indicating that the dynamics of video data differ from those of mobile traffic data. Although video data and mobile traffic data share structural similarities, directly utilizing video prediction models delivers unsatisfying performance.
- **Our proposed noise prior framework achieves superior performance.** Our proposed approach achieves the best performance compared to competitive baselines across almost all datasets. Compared with CSDI without noise prior, it has significant improvements in most cases, with an average reduction of **41.6%** in MAE and **54.4%** in RMSE. The remarkable performance improvement highlights the great potential of manipulating noise in diffusion models.
- **More accurate prior dynamics boost more performance gain.** Specifically, on the CellularBJ and CellularNJ datasets, the diffusion models' average performance improves by 40.6% and 47.2% in terms of MAE when using the noise prior. Meanwhile, on the CellularSH and MobileSH datasets, although the improvements are slightly lower, there is still an average performance increase of 21.1% and 25.5% in MAE. The observations align with the cosine similarity results shown in Table 1, where the similarity scores for CellularBJ and CellularNJ datasets are higher.

We also conducted an experiment predicting 6 steps from 12 historical steps, with results shown in Table 4. The experiment demonstrates that by incorporating noise priors, the performance of all models is significantly improved, with the average MAE reduced by 25.2% and the RMSE decreased by 34.3%.

One-step Prediction. For this task, we set the context length to 12 and limit the prediction horizon to 1. Specifically, we use

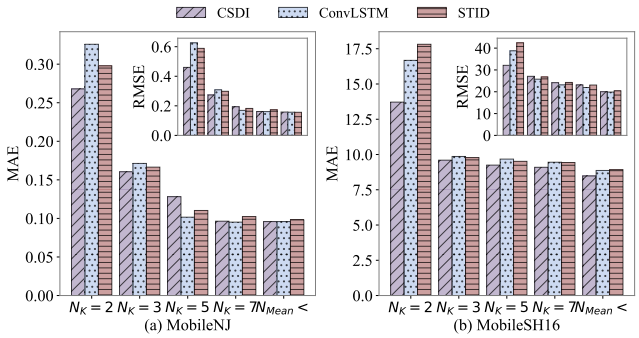


Figure 5: Ablation studies on the number of FFT frequency components in periodic dynamics.

periodic dynamics and local dynamics, separately, to investigate these contributions to the final performance.

Table 5 illustrates the results of one-step prediction. When applying periodic dynamics as noise prior, significant performance gains are observed in most datasets, with an average MAE reduction of 30.3%. However, in a few datasets, this approach does not yield the expected performance. In contrast, using local dynamics proves effective across all datasets and further reduces the MAE by 35.1% on average. It indicates that local dynamics are more suitable for one-step prediction, due to their ability to reflect short-term abrupt and sudden changes. Also as shown in Table 1, the similarity between local dynamics and the real data is higher than that of periodic dynamics. This further highlights the importance of adopting suitable dynamics as noise prior.

6.3 Ablation Studies

Noise prior is the most essential design of our method. In this section, we explore the impact of two key hyperparameters associated with it: N_k and λ.

Noise Fusion Coefficient. Figure 4 shows the impact of the coefficient λ in Eq. (12). We can observe that as λ increases, the model's performance gradually improves, demonstrating the effectiveness of noise prior. This also indicates that increasing the contribution of the noise prior allows for better utilization of prior dynamics. The model achieves optimal performance when λ is around 0.5. However, as λ continues to increase, a slight performance decline occurs. This can be attributed to the fact that the extracted prior dynamics only provide regular dynamics and cannot fully represent all variations in the real data. Over-reliance on the noise prior may result in suboptimal performance.

Components of FFT. Figure 5 illustrates the impact of the number of FFT frequency components in periodic dynamics on the model's performance. The experimental results show that as the number of components increases, the performance of models improves accordingly. When the number of Fourier components reaches 5, allowing for sufficiently accurate capture of the data dynamics, the model's performance metrics tend to stabilize.

6.4 Training Efficiency and Robustness

Training Efficiency. Figure 7 illustrates the trend of performance on the validation set over the first five epochs. We can observe that the model with noise prior exhibits significantly faster convergence in the early stages of training. This rapid convergence can be attributed to the model's ability to directly estimate part of the noise, enabling it to reach the optimal solution more quickly during training. This efficiency improvement is important for real-time applications that require high efficiency.

Robustness Analysis. We examine the robustness of our method from two aspects: first, we evaluate it on different prediction tasks; second, we evaluate it under noise perturbations.

- **Different Prediction Tasks.** We extend the evaluation task to a series of scenarios with different historical horizons and prediction steps, including 12-6, 24-24, 24-12, and 24-1. The detailed results can be found in Appendix Tables 4 and 7 to 9. The results show that our method improves model performance across all prediction tasks, demonstrating its robustness in different tasks.
- **Noise Perturbations.** We evaluate the performance of our method under noise perturbations by introducing Gaussian noise with a mean of 0 into the mobile traffic data. The variance levels are set to range from 10% to 50% of the mean value. Figure 8 shows the performance of CSDI under different levels of noise perturbation. The results show that as the noise level gradually increases, the models incorporating noise prior exhibit higher stability, with only a minor performance decrease, proving its robustness.

6.5 Case Study

To further investigate the impact of noise prior, we visualize the prediction results for the MobileNJ dataset. We use CSDI as an example, and Figure 6 illustrates the results (the prediction results with ConvLSTM and STID can be found in Appendix Figure 11).

We observe that, compared to CSDI (indicated in gray), incorporating the noise prior (shown in purple) significantly reduces the uncertainty in predictions. This is especially important for probabilistic forecasting, where uncertainty reflects the reliability of the model's predictions. Additionally, as shown in Figure 6(a, b, and h), when dealing with uncommon circumstances, where sudden changes occur, our approach can well handle them due to the incorporation of prior dynamics.

7 Conclusion

In this paper, we propose NPDiff, a general noise prior framework for diffusion-based mobile traffic prediction, pioneering the manipulation of noise in diffusion models for temporal data prediction. NPDiff offers significant advancements over existing solutions in several key areas, including enhanced prediction accuracy, improved training efficiency, increased robustness, and reduced uncertainty. These improvements provide a more efficient and reliable approach to real-world mobile traffic management, demonstrating the framework's potential to enhance both performance and practical applicability in this critical domain.

Our work is highly extensible, offering multiple avenues for future exploration. On one hand, further research could focus on refining the fusion of the noise prior and residual noise to better

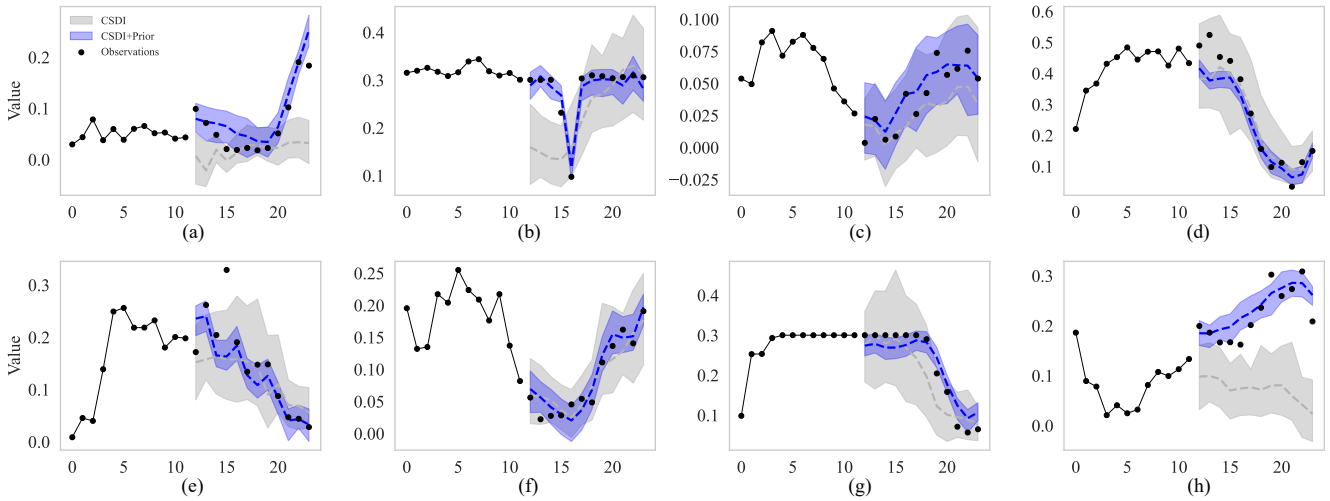


Figure 6: Visualizations of prediction uncertainties using CSDI and CSDI with noise prior on the MobileNJ dataset. The shaded areas represent prediction uncertainty, illustrating the 90% confidence interval based on 50 independent runs. The dashed lines indicate the median of the prediction results for each model.

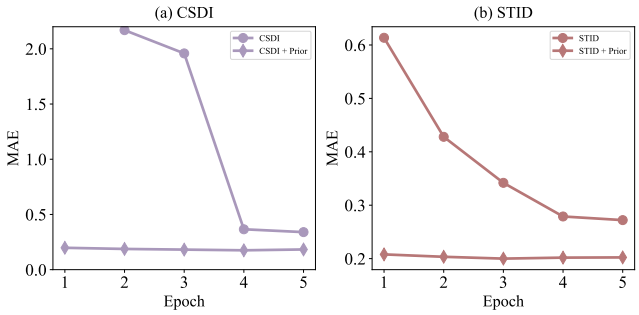


Figure 7: Results of training efficiency in terms of MAE on the validation set.

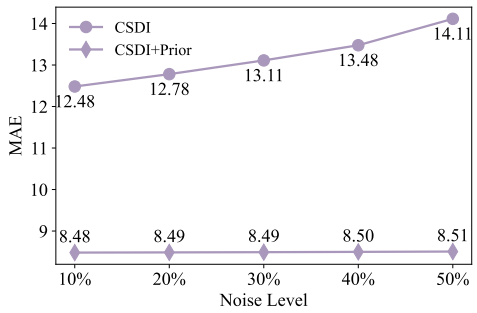


Figure 8: Results of noise perturbation on the MobileSH16 dataset.

handle complex data scenarios. On the other hand, designing more advanced noise priors is another promising direction. This could involve developing priors that adapt to the unique dynamics of various spatio-temporal scenarios or creating a more general prior that can generalize effectively across diverse scenarios.

References

[1] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. Advances in neural information processing systems 33 (2020), 17804–17815.

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023).

[3] Haoye Chai, Tao Jiang, and Li Yu. 2024. Diffusion Model-based Mobile Traffic Generation with Open Data for Network Planning and Optimization. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 4828–4838.

[4] Haoye Chai, Tong Li, Fenyu Jiang, Shiyuan Zhang, and Yong Li. 2024. Knowledge Guided Conditional Diffusion Model for Controllable Mobile Traffic Generation. In Companion Proceedings of the ACM on Web Conference 2024. 851–854.

[5] Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C Azevedo. 2024. How I Warped Your Noise: a Temporally-Correlated Noise Prior for Diffusion Models. In The Twelfth International Conference on Learning Representations.

[6] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. 2021. Mau: A motion-aware unit for video prediction and beyond. Advances in Neural Information Processing Systems 34 (2021), 26950–26962.

[7] Changlu Chen, Yanbin Liu, Ling Chen, and Chengqi Zhang. 2022. Bidirectional spatial-temporal adaptive transformer for Urban traffic flow forecasting. IEEE Transactions on Neural Networks and Learning Systems (2022).

[8] Weihuang Chen, Fangfang Wang, and Hongbin Sun. 2021. S2tnet: Spatio-temporal transformer networks for trajectory prediction in autonomous driving. In Asian Conference on Machine Learning. PMLR, 454–469.

[9] Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. 2003. ARIMA models to predict next-day electricity prices. IEEE transactions on power systems 18, 3 (2003), 1014–1020.

[10] Anestis Dalgkitsis, Malamati Louta, and George T Karetsos. 2018. Traffic forecasting in cellular networks using the LSTM RNN. In Proceedings of the 22nd Pan-Hellenic conference on informatics. 28–33.

[11] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. 2021. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 269–278.

[12] Jie Feng, Xinlei Chen, Rundong Gao, Ming Zeng, and Yong Li. 2018. Deeptp: An end-to-end neural network for mobile cellular traffic prediction. IEEE Network 32, 6 (2018), 108–115.

[13] Zhiying Feng, Qiong Wu, and Xu Chen. 2024. Communication-efficient Multi-service Mobile Traffic Prediction by Leveraging Cross-service Correlations. In

Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 794–805.
[14] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. 2023. Preserve your own correlation: A noise prior for video diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 22930–22941.
[15] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In Proceedings of the AAAI conference on artificial intelligence, Vol. 33. 3656–3663.
[16] Jiahui Gong, Tong Li, Huandong Wang, Yu Liu, Xing Wang, Zhendong Wang, Chao Deng, Junlan Feng, Depeng Jin, and Yong Li. 2024. KGDA: A Knowledge Graph-based Decomposition Approach for Cellular Traffic Prediction. ACM Transactions on Intelligent Systems and Technology (2024).
[17] Jiahui Gong, Yu Liu, Tong Li, Haoye Chai, Xing Wang, Junlan Feng, Chao Deng, Depeng Jin, and Yong Li. 2023. Empowering Spatial Knowledge Graph for Mobile Traffic Prediction. In Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems. 1–11.
[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. Advances in neural information processing systems 33 (2020), 6840–6851.
[19] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. 2022. Video diffusion models. Advances in Neural Information Processing Systems 35 (2022), 8633–8646.
[20] Yahui Hu, Yujiang Zhou, Junping Song, Luyang Xu, and Xu Zhou. 2022. Citywide mobile traffic forecasting using spatial-temporal downsampling transformer neural networks. IEEE Transactions on Network and Service Management 20, 1 (2022), 152–165.
[21] Chih-Wei Huang, Chiu-Ti Chiang, and Qiuhui Li. 2017. A study of deep learning networks on mobile traffic forecasting. In 2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC). IEEE, 1–6.
[22] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In Proceedings of the AAAI conference on artificial intelligence, Vol. 37. 4365–4373.
[23] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728 (2023).
[24] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In International Conference on Learning Representations.
[25] Dongtian Liang, Jiaxin Zhang, Shuai Jiang, Xing Zhang, Jie Wu, and Qi Sun. 2019. Mobile traffic prediction based on densely connected CNN for cellular networks in highway scenarios. In 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 1–5.
[26] Zhihui Lin, Maomao Li, Zhuobin Zheng, Yangyang Cheng, and Chun Yuan. 2020. Self-attention convlstm for spatiotemporal prediction. In Proceedings of the AAAI conference on artificial intelligence, Vol. 34. 11531–11538.
[27] Lingbo Liu, Ruimao Zhang, Jiefeng Peng, Guanbin Li, Bowen Du, and Liang Lin. 2018. Attentive crowd flow machines. In Proceedings of the 26th ACM international conference on Multimedia. 1553–1561.
[28] Qingyao Liu, Jianwu Li, and Zhaoming Lu. 2021. ST-Tran: Spatial-temporal transformer for cellular traffic prediction. IEEE Communications Letters 25, 10 (2021), 3325–3329.
[29] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310.06625 (2023).
[30] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In International conference on machine learning. PMLR, 8162–8171.
[31] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. arXiv preprint arXiv:2211.14730 (2022).
[32] William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 4195–4205.
[33] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. 2023. Freenoise: Tuning-free longer video diffusion via noise rescheduling. arXiv preprint arXiv:2310.15169 (2023).
[34] Iqbal H Sarker, Mohammed Moshiul Hoque, Md Kafil Uddin, and Tawfeeq Alsanoosy. 2021. Mobile data science and intelligent apps: concepts, AI-based modeling and research directions. Mobile Networks and Applications 26, 1 (2021), 285–303.
[35] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. 2022. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 4454–4458.
[36] Bethelhem S Shawel, Endale Mare, Tsegamlak T Debella, Sofie Pollin, and Dereje H Woldegebreal. 2022. A multivariate approach for spatiotemporal mobile data traffic prediction. Engineering Proceedings 18, 1 (2022), 10.
[37] Lifeng Shen and James Kwok. 2023. Non-autoregressive conditional diffusion models for time series prediction. In International Conference on Machine Learning. PMLR, 31016–31029.
[38] Hongzhi Shi and Yong Li. 2018. Discovering periodic patterns for large scale mobile traffic data: Method and applications. IEEE Transactions on Mobile Computing 17, 10 (2018), 2266–2278.
[39] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems 28 (2015).
[40] Feiyang Sun, Pinghui Wang, Junzhou Zhao, Nuo Xu, Juxiang Zeng, Jing Tao, Kaikai Song, Chao Deng, John CS Lui, and Xiaohong Guan. 2021. Mobile data traffic prediction by exploiting time-evolving user mobility patterns. IEEE Transactions on mobile computing 21, 12 (2021), 4456–4470.
[41] Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z Li. 2023. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 18770–18782.
[42] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. Advances in Neural Information Processing Systems 34 (2021), 24804–24816.
[43] Hoang Duy Trinh, Lorenza Giupponi, and Paolo Dini. 2018. Mobile traffic prediction from raw data using LSTM networks. In 2018 IEEE 29th annual international symposium on personal, indoor and mobile radio communications (PIMRC). IEEE, 1827–1832.
[44] Sebastian Troia, Rodolfo Alvizu, Youduo Zhou, Guido Maier, and Achille Pattavina. 2018. Deep learning-based traffic prediction for network optimization. In 2018 20th International Conference on Transparent Optical Networks (ICTON). IEEE, 1–4.
[45] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. 2022. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. Advances in neural information processing systems 35 (2022), 23371–23385.
[46] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. 2018. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In International Conference on Machine Learning. PMLR, 5123–5132.
[47] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. 2017. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. Advances in neural information processing systems 30 (2017).
[48] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. 2019. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 9154–9162.
[49] Zhenzhen Wang, Sylvia Y He, and Yee Leung. 2018. Applying mobile phone data to travel behaviour research: A literature review. Travel Behaviour and Society 11 (2018), 141–155.
[50] Zi Wang, Jia Hu, Geyong Min, Zhiwei Zhao, Zheng Chang, and Zhe Wang. 2022. Spatial-temporal cellular traffic prediction for 5G and beyond: A graph neural networks-based approach. IEEE Transactions on Industrial Informatics 19, 4 (2022), 5722–5731.
[51] Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. 2023. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems. 1–12.
[52] Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. 2023. A survey on video diffusion models. Comput. Surveys (2023).
[53] Fengli Xu, Yong Li, Huandong Wang, Pengyu Zhang, and Depeng Jin. 2016. Understanding mobile traffic patterns of large scale cellular towers in urban environment. IEEE/ACM transactions on networking 25, 2 (2016), 1147–1161.
[54] Fengli Xu, Yuyun Lin, Jiaxin Huang, Di Wu, Hongzhi Shi, Jeungeun Song, and Yong Li. 2016. Big data driven mobile traffic understanding and forecasting: A time series approach. IEEE transactions on services computing 9, 5 (2016), 796–805.
[55] Yiyuan Yang, Ming Jin, Haomin Wen, Chaoli Zhang, Yuxuan Liang, Lintao Ma, Yi Wang, Chenghao Liu, Bin Yang, Zenglin Xu, et al. 2024. A survey on diffusion models for time series and spatio-temporal data. arXiv preprint arXiv:2404.18886 (2024).
[56] Xiaohu You, Cheng-Xiang Wang, Jie Huang, Xiqi Gao, Zaichen Zhang, Mao Wang, Yongming Huang, Chuan Zhang, Yanxiang Jiang, Jiaheng Wang, et al. 2021. Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts. Science China Information Sciences 64 (2021), 1–74.

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

[58] Yuan Yuan, Jingtao Ding, Chenyang Shao, Depeng Jin, and Yong Li. 2023. Spatio-temporal diffusion point processes. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 3173–3184.

[59] Yuan Yuan, Chenyang Shao, Jingtao Ding, Depeng Jin, and Yong Li. 2024. Spatio-Temporal Few-Shot Learning via Diffusive Neural Network Generation. In The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=QyFm3D3Tzi

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

[61] Zhongwei Zhang, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Ting Yao, Yang Cao, and Tao Mei. 2024. TRIP: Temporal Residual Learning with Image Noise Prior for Image-to-Video Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8671–8681.

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

[63] Liang Zhao, Min Gao, and Zongwei Wang. 2022. St-gsp: Spatial-temporal global semantic representation learning for urban flow prediction. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. 1443–1451.

[64] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. IEEE transactions on intelligent transportation systems 21, 9 (2019), 3848–3858.

[65] Shuai Zhao, Xiaopeng Jiang, Guy Jacobson, Rittwik Jana, Wen-Ling Hsu, Raif Rustamov, Manoop Talasila, Syed Anwar Aftab, Yi Chen, and Cristian Borcea. 2020. Cellular network traffic prediction incorporating handover: A graph convolutional approach. In 2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 1–9.

[66] Zhilun Zhou, Jingtao Ding, Yu Liu, Depeng Jin, and Yong Li. 2023. Towards Generative Modeling of Urban Flow through Knowledge-enhanced Denoising Diffusion. In Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems. 1–12.

APPENDIX

A Implementation Details

A.1 Datasets

Table 6 provides basic information about the datasets. For our experiments, we divide each dataset into three parts with a ratio of 6:2:2. The first 60% is used as the training set, the subsequent 20% as the validation set, and the remaining 20% as the test set. All datasets are standardized to a standard normal distribution. It is necessary to specify that the periodic dynamics utilized in our study are solely obtained through the analysis of the training set.

A.2 Evaluation Metrics

We apply two widely used metrics in spatio-temporal prediction, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to assess the models' performance. These metrics measure the differences between predicted outcomes and actual values, where lower scores reflect better performance. The formulas for MAE and RMSE are as follows:

$$\text{MAE} = \frac{1}{N}\frac{1}{F}\sum_{i=1}^{N}\sum_{p=t+1}^{t+F}|\hat{X}_{i,j} - X_{i,j}|, \tag{15}$$

$$\text{RMSE} = \frac{1}{N}\frac{1}{F}\sum_{i=1}^{N}\sum_{j=t+1}^{t+F}(\hat{X}_{i,j} - X_{i,j})^2. \tag{16}$$

In the two formulas, $\hat{X}_{i,j}$ represents the predicted value, while $X_{i,j}$ corresponds to the target values. N denotes the total number of samples in the dataset, and F signifies the prediction horizon.

A.3 Experimental Configuration

For denoising networks, we set CSDI's residual layers to 4, with 64 residual channels and 8 attention heads. For ConvLSTM, we configure 3 LSTM layers with a hidden size of 64. For STID, we set encoder to 4 layers with an embedding dimension of 64. During training, we employ a quadratic noise schedule in our diffusion model, starting with a noise level of $\beta_1 = 0.0001$, which increases to a maximum of $\beta_N = 0.5$ over $N = 50$ steps. We use a batch size of 8 and an initial learning rate of 1e-3, which is reduced to 4e-4 after 40 epochs. We apply the Adam optimizer with a weight decay of 1e-6. For our method's hyperparameters, we set the number of periodic dynamic components N_K to 5, or N_m, representing the number of components with amplitudes above the average. We select the weighted coefficient λ from the range {0.3, 0.4, 0.5, 0.6, 0.7}. In the experiments, we report the best performance selected from all combinations of these two parameter settings.

A.4 Denoising Networks

A brief introduction of the three denoising networks used in our experiments is provided below:

- CSDI [42]: CSDI is one of the most classic diffusion-based models in the spatio-temporal domain. Its denoising network is designed based on a transformer architecture and has demonstrated outstanding performance in time series data imputation tasks.
- ConvLSTM [39]: ConvLSTM is a seminal model in spatio-temporal forecasting, utilizing Convolutional LSTM units to effectively capture spatio-temporal dynamics. It has demonstrated great performance in applications such as weather forecasting and traffic flow prediction. In our work, we utilize ConvLSTM as one of the denoising networks.
- STID [35]: STID is an efficient and straightforward spatio-temporal forecasting model that employs MLP to capture dependencies in multivariate time series, thereby avoiding the complexities inherent in deeper models. It effectively captures spatio-temporal features, demonstrating superior performance across various tasks. In our experiments, STID is utilized as a denoising network to develop a new diffusion-based model and to evaluate the performance of our approach.

Algorithm 1 Training of NPDiff

1: **Input:** Distribution of training data $q(x_0)$, noise schedule β_n, data dynamics D
2: **Output:** Trained denoising function ϵ_θ
3: **repeat**
4: Separate the values of x_0 into context data x_0^{co} and target data x_0^{ta}
5: Initialize $n \sim \text{Uniform}(1, \dots, N)$ and $\epsilon \sim \mathcal{N}(0, I)$
6: Calculate noisy targets $x_n^{ta} = \sqrt{\bar{\alpha}_n}x_0^{ta} + \sqrt{1-\bar{\alpha}_n}\epsilon$
7: Calculate noise prior $\widetilde{\epsilon} = \frac{x_n^{ta} - \sqrt{\bar{\alpha}_n}D}{\sqrt{1-\bar{\alpha}_n}}$
8: Take a gradient step on:

$$\nabla_\theta \left[\left\| \epsilon - (1-\lambda)\epsilon_\theta(x_n^{ta}, n | x_0^{co}) - \lambda\widetilde{\epsilon} \right\|_2^2 \right]$$

9: **until** converged

Algorithm 2 Sampling of NPDiff

1: **Input:** Context data x_0^{co}, data dynamics D, trained denoising function ϵ_θ
2: **Output:** Target data x_0^{ta}
3: Sample x_N^{ta} from $\epsilon \sim \mathcal{N}(0, I)$
4: **for** $n = N$ to 1 **do**
5: Calculate noise prior $\widetilde{\epsilon} = \frac{x_n^{ta} - \sqrt{\bar{\alpha}_n}D}{\sqrt{1-\bar{\alpha}_n}}$
6: Predict noise $\epsilon_\theta(x_n^{ta}, n | x_0^{co})$
7: Reverse diffusion to get x_{n-1}^{ta} using Eq. (14)
8: **end for**
9: **Return:** x_0^{ta}

A.5 Training Algorithm

The training process of NPDiff is shown in Algorithm 1, and sampling process is shown in Algorithm 2.

B Baselines

- **HA**: The History Average approach relies on the mean of data from previous time intervals to forecast target values.
- **ARIMA**: The ARIMA model is a frequently applied statistical approach for time series forecasting. This method effectively predicts time series data obtained at consistent time intervals.

Table 6: The basic information of four mobile traffic datasets.

Dataset	City	Type	Temporal Period	Spatial partition	Interval	Mean	Std
MobileBJ	Beijing	Crowd flow	2021/10/25 - 2021/11/21	28×24	One hour	0.367	0.411
MobileNJ	Nanjing	Crowd flow	2021/02/02 - 2021/02/22	20×28	One hour	0.842	1.299
MobileSH14	Shanghai	Dynamic population	2014/08/01 - 2014/08/21	32×28	One hour	0.175	0.212
MobileSH16	Shanghai	Mobility flow	2016/04/25 - 2016/05/01	20×20	15min	31.935	137.926

- **PatchTST** [31]: It introduces patching and self-supervised learning for multivariate time series forecasting. The time series is divided into segments to capture long-term dependencies. Different channels are processed independently using a shared network.
- **iTransformer** [29]: This advanced model for multivariate time series prediction leverages attention and feed-forward processes on an inverted dimension. It focuses on capturing the relationships between different variables.
- **Time-LLM** [23]: TIME-LLM represents a leading-edge approach for applying large language models to time series forecasting. It employs a reprogramming framework that adapts LLMs for general time series predictions while maintaining the original structure of the language models.
- **STResNet** [60]: It makes use of residual neural networks to model dynamics in the data. These networks effectively capture temporal closeness, periodic behaviors, and long-term trends.
- **ATFM** [27]: The Attentive Crowd Flow Machine model forecasts crowd movement by using an attention mechanism. This mechanism adaptively combines both sequential and periodic dynamics to improve prediction accuracy.
- **STNorm** [11]: It introduces two distinct normalization modules: spatial normalization and temporal normalization. These modules work independently, with spatial normalization managing high-frequency components and temporal normalization addressing local variations.
- **STGSP** [63]: This model underscores the importance of both global and positional information within the temporal dimension when performing spatio-temporal predictions. By employing a semantic flow encoder, it captures temporal positional signals, while an attention mechanism is used to handle multi-scale temporal dependencies.
- **TAU** [41]: The Temporal Attention Unit breaks down temporal attention into intra-frame and inter-frame components. It introduces differential divergence regularization to effectively manage variations between frames.
- **PromptST** [62]: It is a state-of-the-art model that combines pre-training and prompt-tuning, specifically tailored for spatio-temporal prediction.
- **MIM** [48]: This model utilizes differential information between adjacent recurrent states to address non-stationary characteristics. By stacking multiple MIM blocks, it enables the modeling of higher-order non-stationarity.
- **MAU** [6]: The Motion-aware Unit improves the detection of motion correlations between frames by extending the temporal coverage of the prediction units. It employs both an attention mechanism and a fusion module to enhance video forecasting.

Table 7: Results of 24-24 multi-step prediction on four datasets evaluated using MAE and RMSE. The results presented in the table are obtained by averaging prediction errors across all prediction steps.

Model	MobileBJ		MobileNJ		MobileSH14		MobileSH16	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CSDI	0.184	0.522	0.130	0.262	0.047	0.076	9.95	29.77
CSDI+Prior	0.089	0.168	0.097	0.168	0.037	0.057	8.98	21.02
ConvLSTM	0.084	0.144	0.159	0.339	0.049	0.080	19.52	53.24
ConvLSTM+Prior	0.090	0.165	0.098	0.158	0.037	0.057	10.53	23.36
STID	0.107	0.168	0.148	0.307	0.054	0.084	13.96	37.44
STID+Prior	0.089	0.166	0.102	0.163	0.037	0.057	11.56	26.76

Table 8: Results of 24-12 multi-step prediction on four datasets evaluated using MAE and RMSE. The results presented in the table are obtained by averaging prediction errors across all prediction steps.

Model	MobileBJ		MobileNJ		MobileSH14		MobileSH16	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CSDI	0.156	0.364	0.139	0.301	0.049	0.078	14.62	66.96
CSDI+Prior ~~+Prior~~	0.092	0.172	0.095	0.157	0.037	0.057	9.38	21.65
ConvLSTM	0.105	0.172	0.381	0.803	0.051	0.082	13.72	39.45
ConvLSTM+Prior	0.092	0.169	0.097	0.161	0.037	0.057	10.37	24.07
STID	0.084	0.131	0.133	0.268	0.050	0.079	10.70	30.99
STID+Prior	0.091	0.168	0.108	0.161	0.037	0.057	10.28	23.99

Table 9: Results of 24-1 one-step prediction on four datasets evaluated using MAE and RMSE.

Model	MobileBJ		MobileNJ		MobileSH14		MobileSH16	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CSDI	0.223	0.809	0.177	0.662	0.066	0.160	8.49	28.99
CSDI+Periodic Prior	0.087	0.157	0.106	0.162	0.037	0.057	8.12	19.92
CSDI+Local Prior	0.058	0.092	0.098	0.178	0.033	0.053	5.27	9.65
ConvLSTM	0.085	0.139	0.113	0.213	0.040	0.063	7.22	18.26
ConvLSTM+Periodic Prior	0.083	0.155	0.097	0.159	0.036	0.055	8.09	20.16
ConvLSTM+Local Prior	0.064	0.101	0.117	0.220	0.034	0.055	5.26	9.81
STID	0.131	0.171	0.105	0.180	0.039	0.058	6.52	13.73
STID+Periodic Prior	0.087	0.153	0.095	0.161	0.037	0.055	7.88	18.06
STID+Local Prior	0.051	0.079	0.094	0.174	0.031	0.050	5.47	10.31

C Additional Results

Table 3 reports the results of two additional diffusion models for the 12-12 multi-step prediction task, both showing notable performance improvements across four datasets. Tables 4 and 7 to 9 detail the performance of our method across different prediction tasks, demonstrating strong outcomes in both multi-step and one-step predictions. Figure 9 presents additional results for the ablation

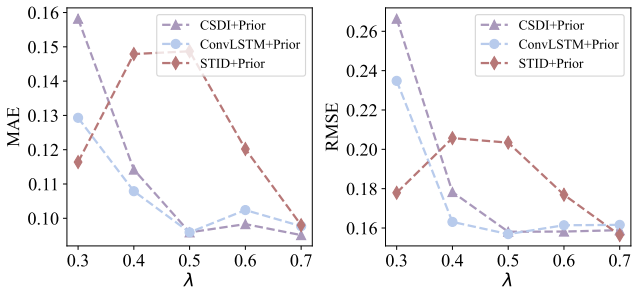
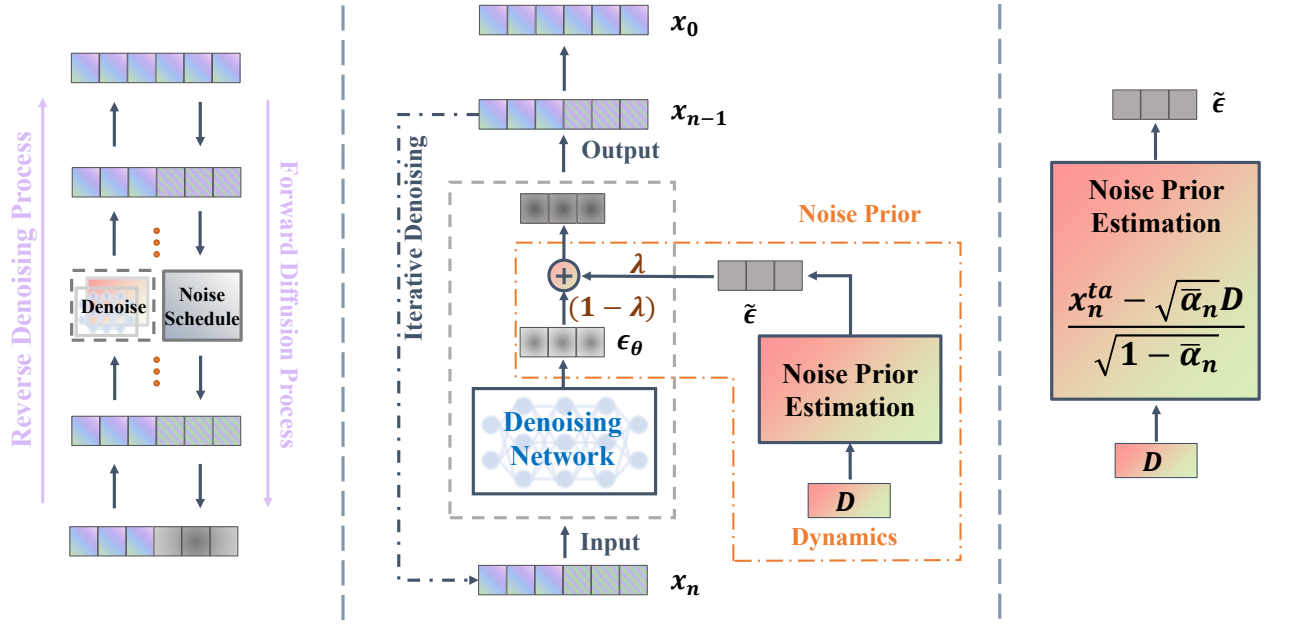


Figure 9: Ablation studies of λ coefficient on MobileNJ dataset.

study on λ using the MobileNJ dataset. Figure 11 visualizes the prediction results with ConvLSTM and STID on MobileNJ dataset.



(a) Diffusion Process (b) Noise Prior Framework (c) Noise Prior Caculation

Figure 10: Detailed explanation of noise prior in the diffusion process.

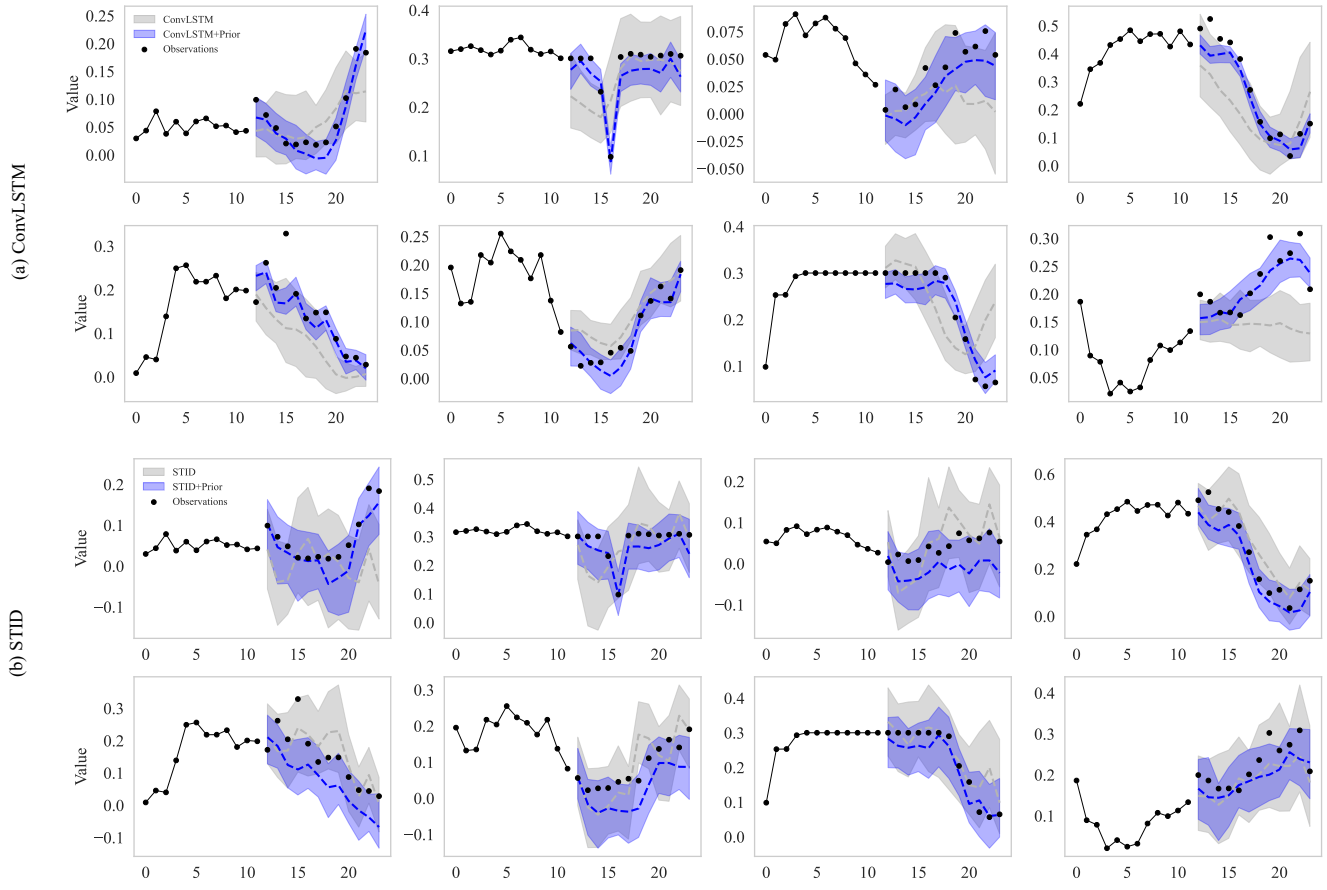


Figure 11: Prediction visualization comparing the noise prior-enhanced models and two baseline models (ConvLSTM and STID) on the MobileNJ dataset. The shaded areas represent prediction uncertainty, illustrating the 90% confidence interval based on 50 independent runs. The dashed lines indicate the median of the predictions for each model.