ICLR | 2024
Twelfth International Conference
on Learning Representations

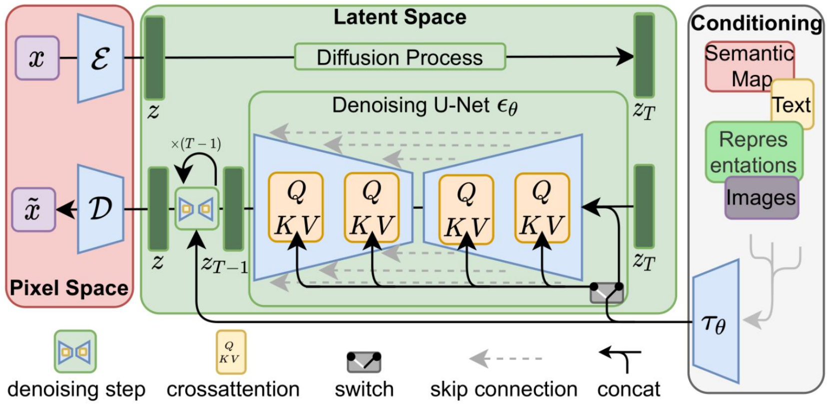
香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

# Multi-Resolution Diffusion Models for Time Series Forecasting

Lifeng Shen, Weiyu Chen, James T. Kwok

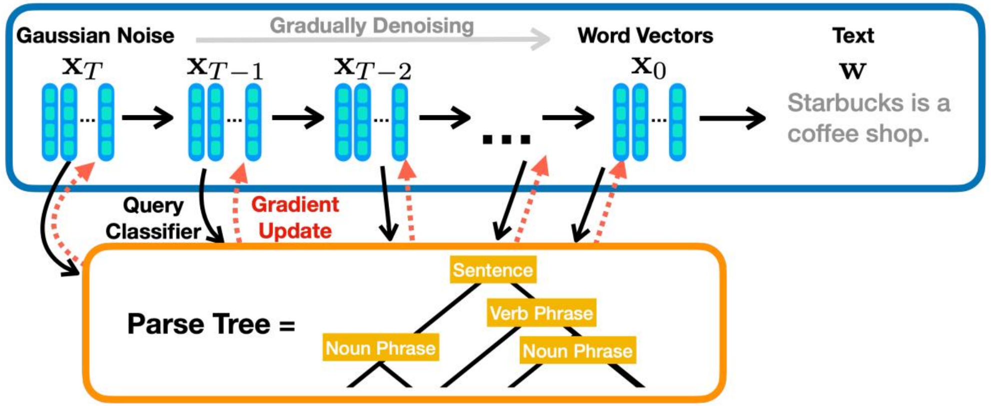
Hong Kong University of Science and Technology

# Diffusion models

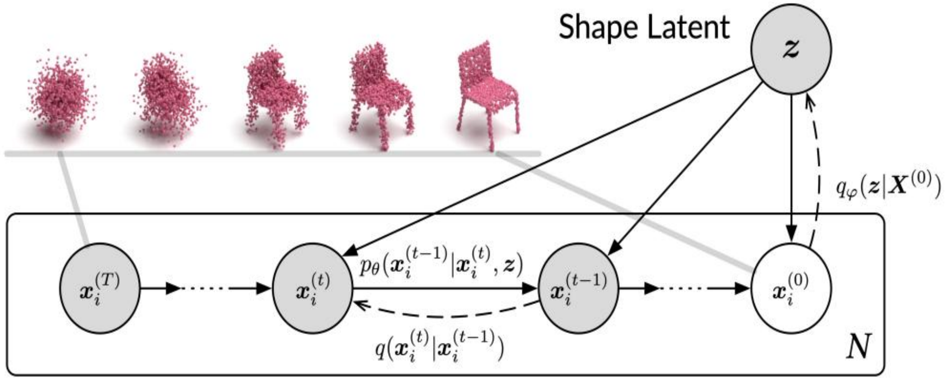



**images**: Stable Diffusion, Midjourney

**video**: Sora



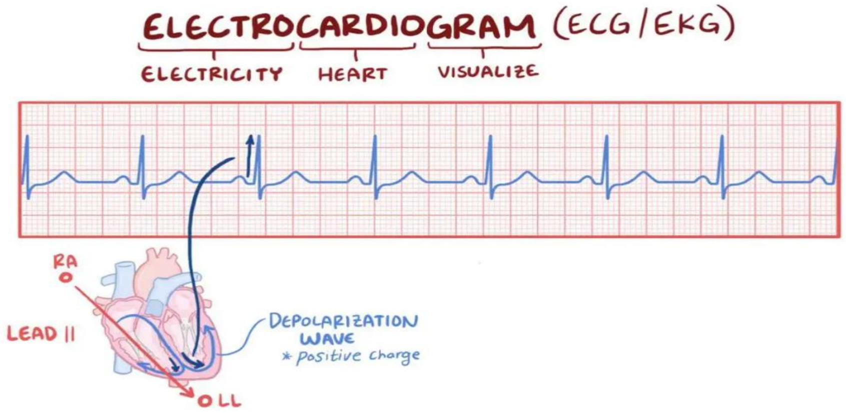

**text**




**3D data**

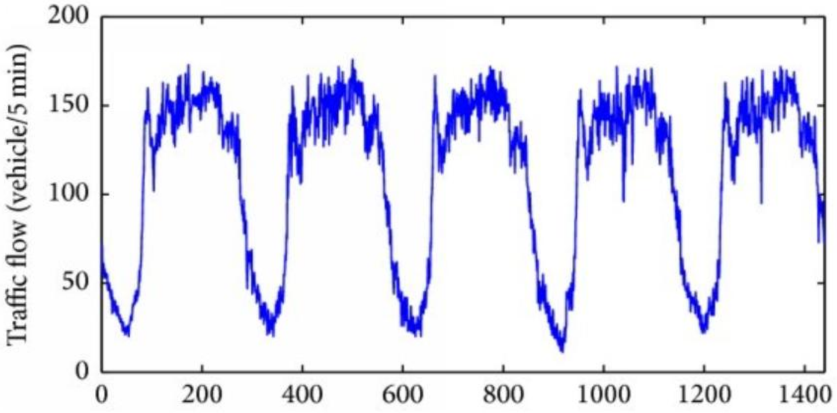
# Time series data are prevalent


**machine monitoring**



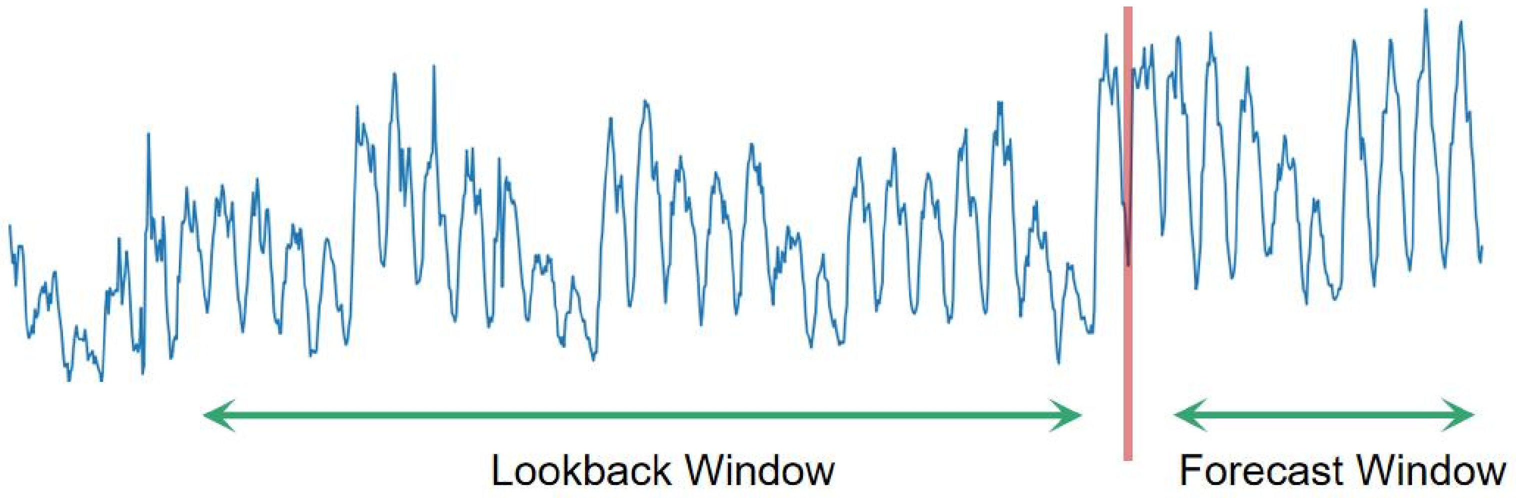

**patient health monitoring**


**stock price prediction**




**traffic flow optimization**

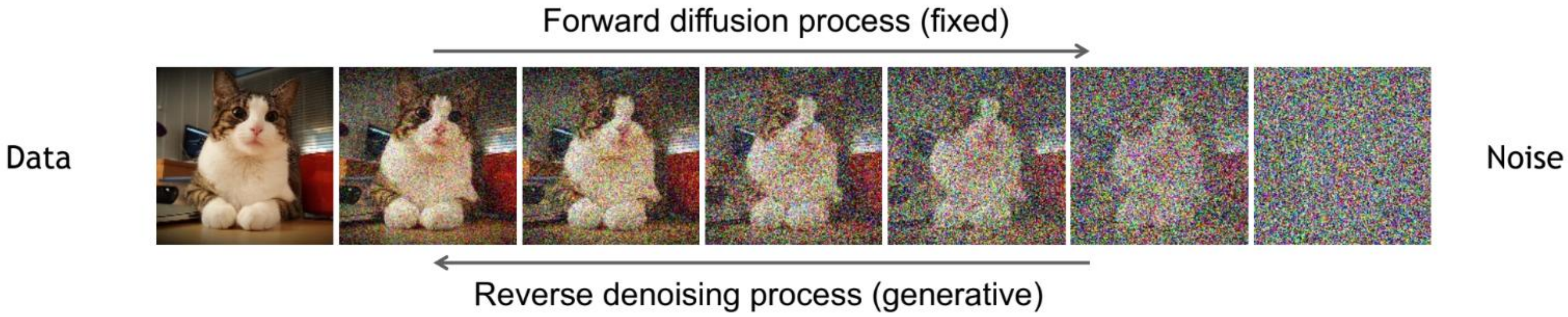
# Time series forecasting




predict forecast window $\mathbf{x}^0_{1:H} \in \mathbb{R}^{d\times H}$ given lookback window $\mathbf{x}^0_{-L+1:0} \in \mathbb{R}^{d\times H}$

- *d*: number of variables
- *H*: length of the forecast window
- *L*: length of the lookback window

# Denoising diffusion probabilistic models (DPPM)




forward diffusion

- input $x^0$ is gradually corrupted to a Gaussian noise vector

$$q(\mathbf{x}^k|\mathbf{x}^{k-1}) = \mathbb{N}(\mathbf{x}^k; \sqrt{1-\beta_k}\mathbf{x}^{k-1}, \beta_k\mathbf{I})$$

backward denoising

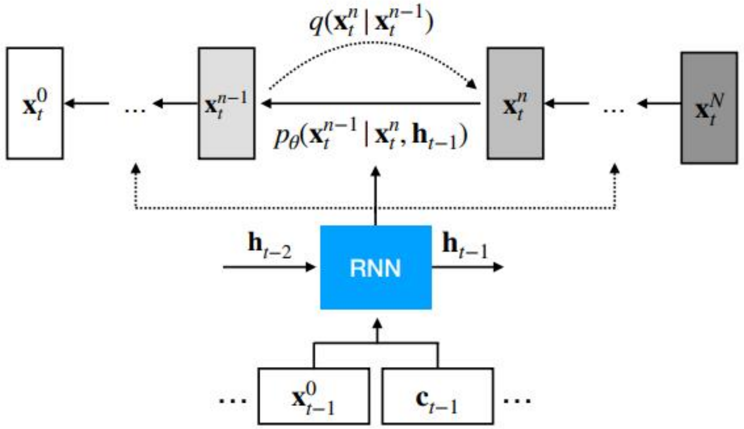
- learns to generate data by denoising

$$p_\theta(\mathbf{x}^{k-1}|\mathbf{x}^k) = \mathbb{N}(\mathbf{x}^{k-1}; \mu_\theta(\mathbf{x}^k, k), \sigma_k^2\mathbf{I})$$

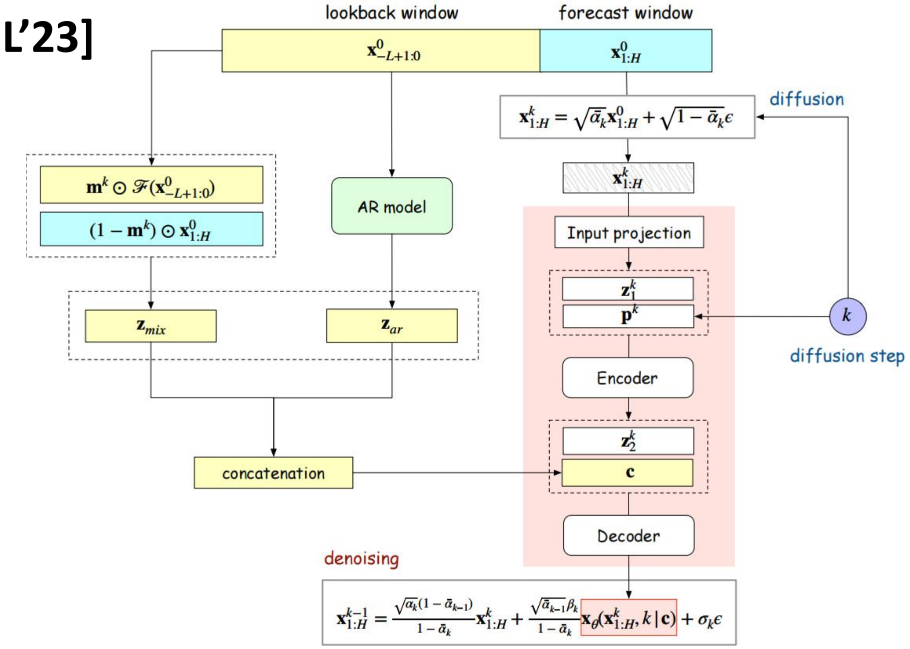
- $\mu_\theta(\mathbf{x}^k, k)$: defined by a neural network (with parameter $\theta$ ).

# Example diffusion models for time series

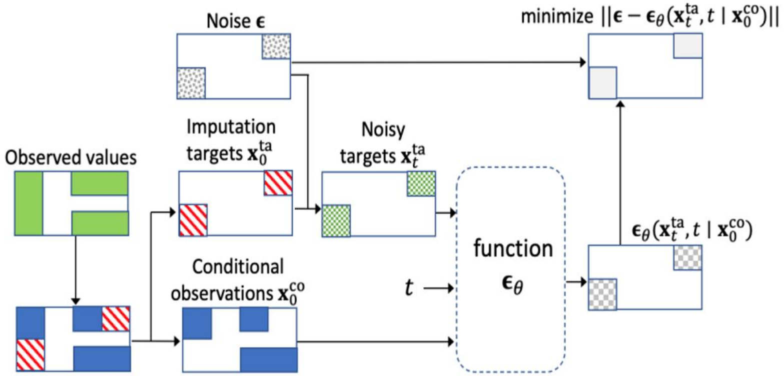
**TimeGrad [ICML'21]**



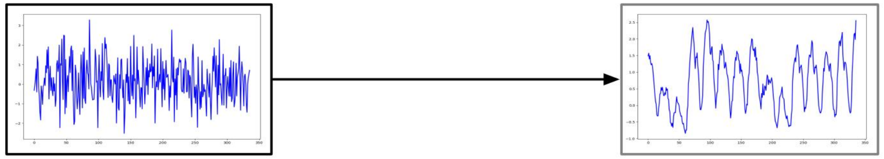

**TimeDiff [ICML'23]**




**CSDI [NeurIPS'21]**




generate time series directly from random vectors



random noises

time series

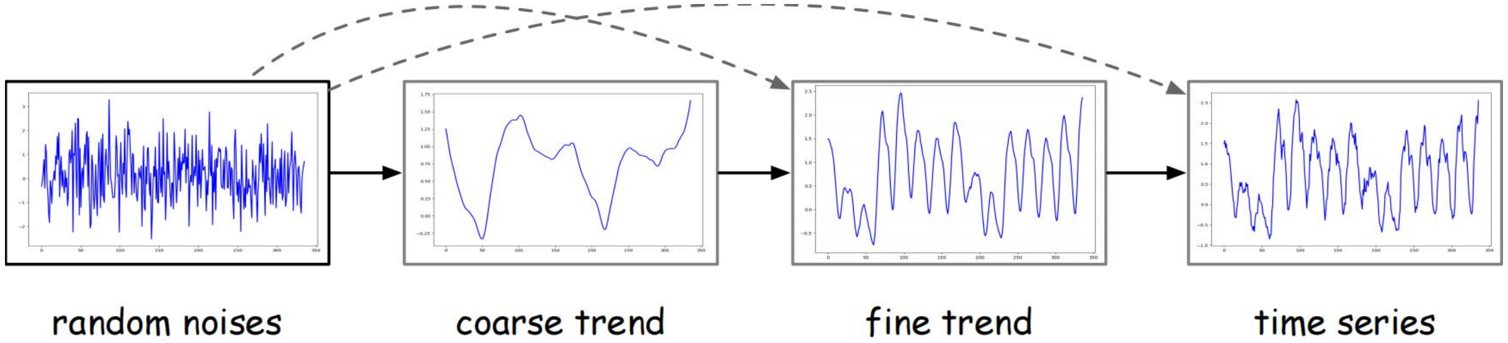
# Motivation

how to better utilize structural properties in time series?

multi-resolution temporal structure in time series

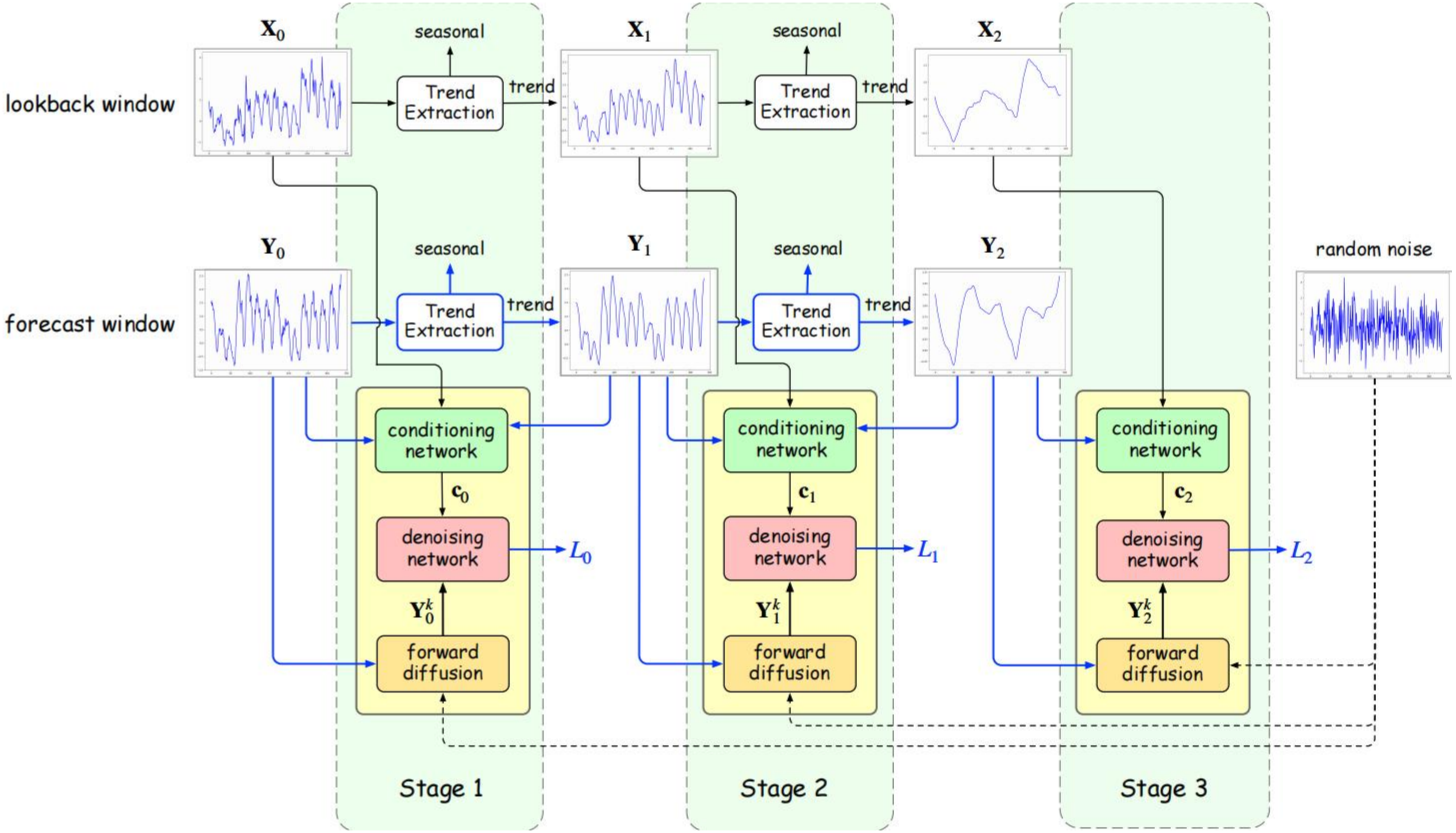
- **seasonal-trend decomposition**: extract the seasonal and trend components
- **the coarser temporal patterns can be used to help modeling the finer patterns**

how to use multi-resolution analysis in time series diffusion models?



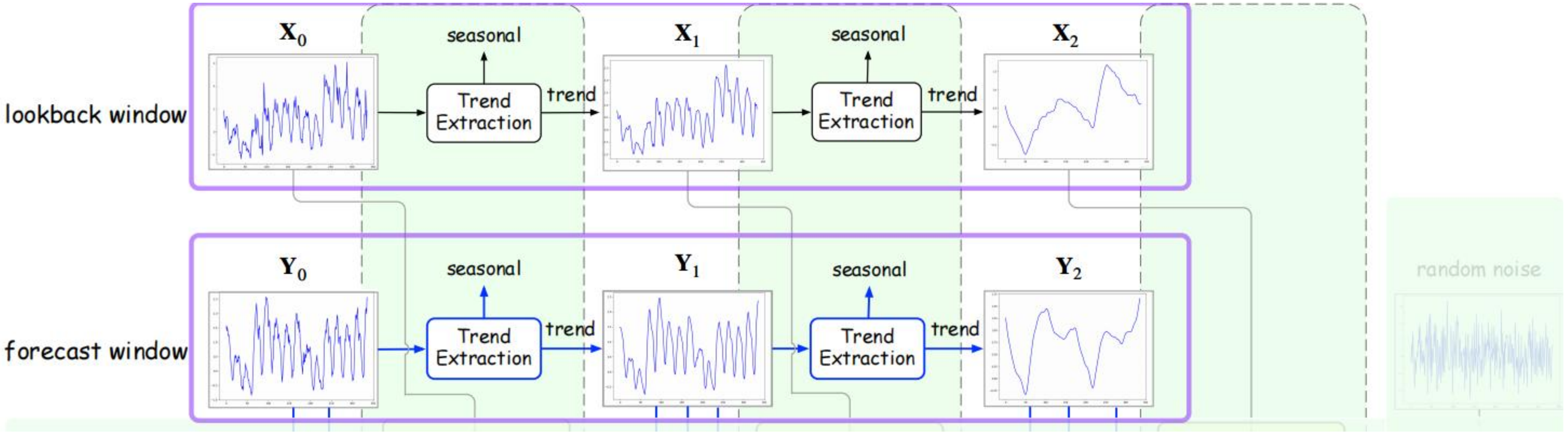

- coarser trends are generated first and the finer details are then progressively added.
- decompose the denoising objective into several sub-objectives, each corresponds to a particular resolution.

# Multi-resolution diffusion model (mr-Diff)




- in each stage, diffusion is interleaved with seasonal-trend decomposition

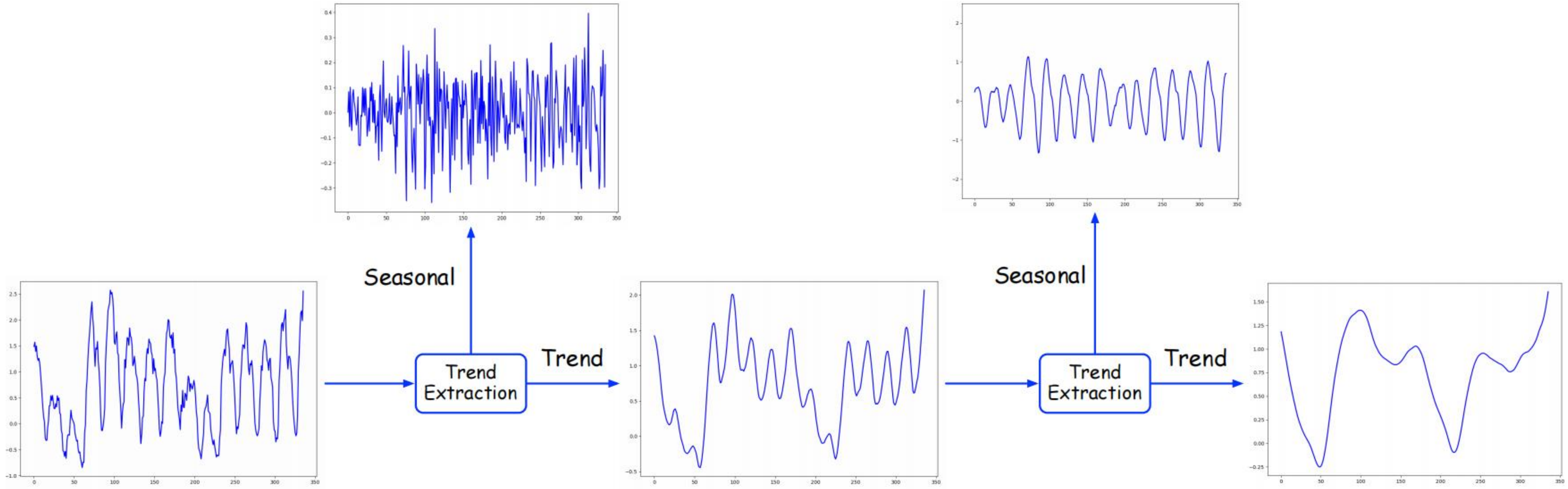
# Extract trend components successively




$$\mathbf{X}_s = \texttt{AvgPool}(\texttt{Padding}(\mathbf{X}_{s-1}), \tau_s)$$

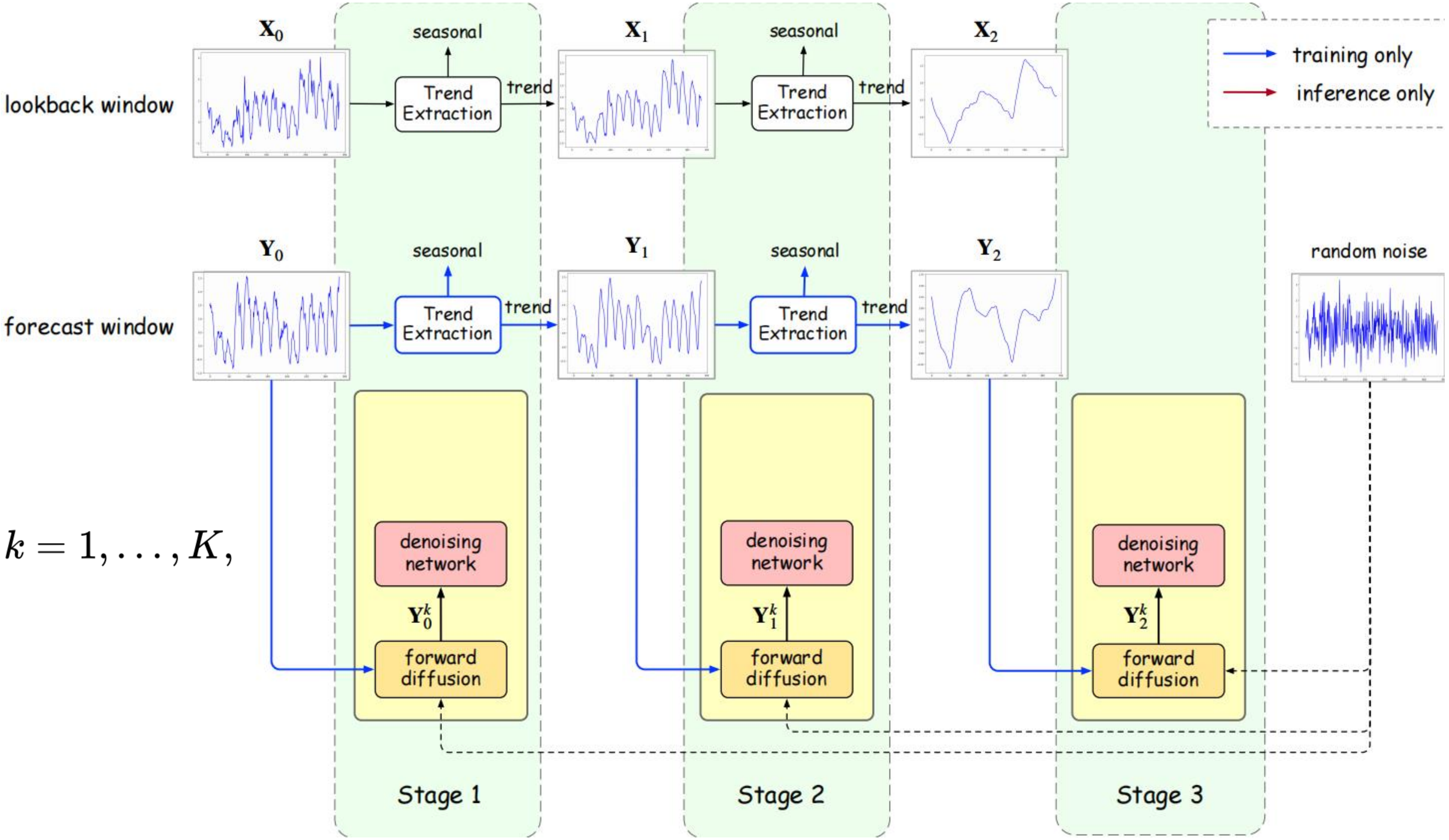
- **X**s : trend component at stage s + 1 ($\mathbf{X}_0 = \mathbf{X}$)
  - AvgPool: average pooling
  - Padding: keeps the lengths of Xs−1 and Xs the same
  - $\tau_s$ : smoothing kernel size
    - increases with s ( trend gets coarser as s increases
- similar processing for the segment $\mathbf{Y}_0$ in the forecast window
  extract trend components $\{\mathbf{Y}_s\}_{s=1,\ldots,S-1}$

# Extract trend components successively




- focus here is on the trend component
- for time series, it is easier to predict a finer trend from a coarser trend
- stage s + 1: learns to reconstruct the trend component Ys from Ys+1

# Forward diffusion

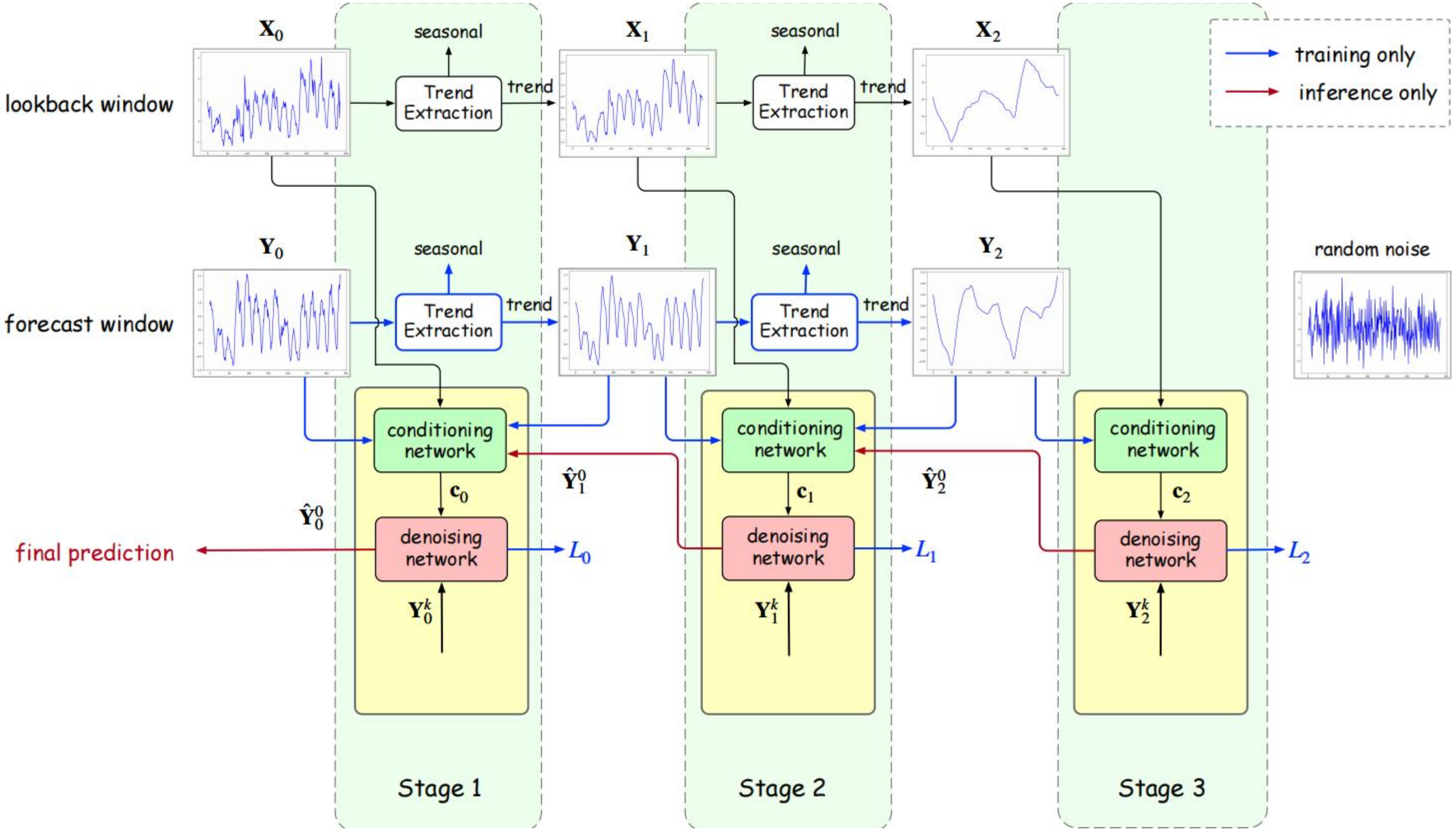



- same as in DDPM

$$\mathbf{Y}_s^k = \sqrt{\bar{\alpha}_k}\mathbf{Y}_s^0 + \sqrt{1-\bar{\alpha}_k}\epsilon, \quad k = 1, \ldots, K,$$

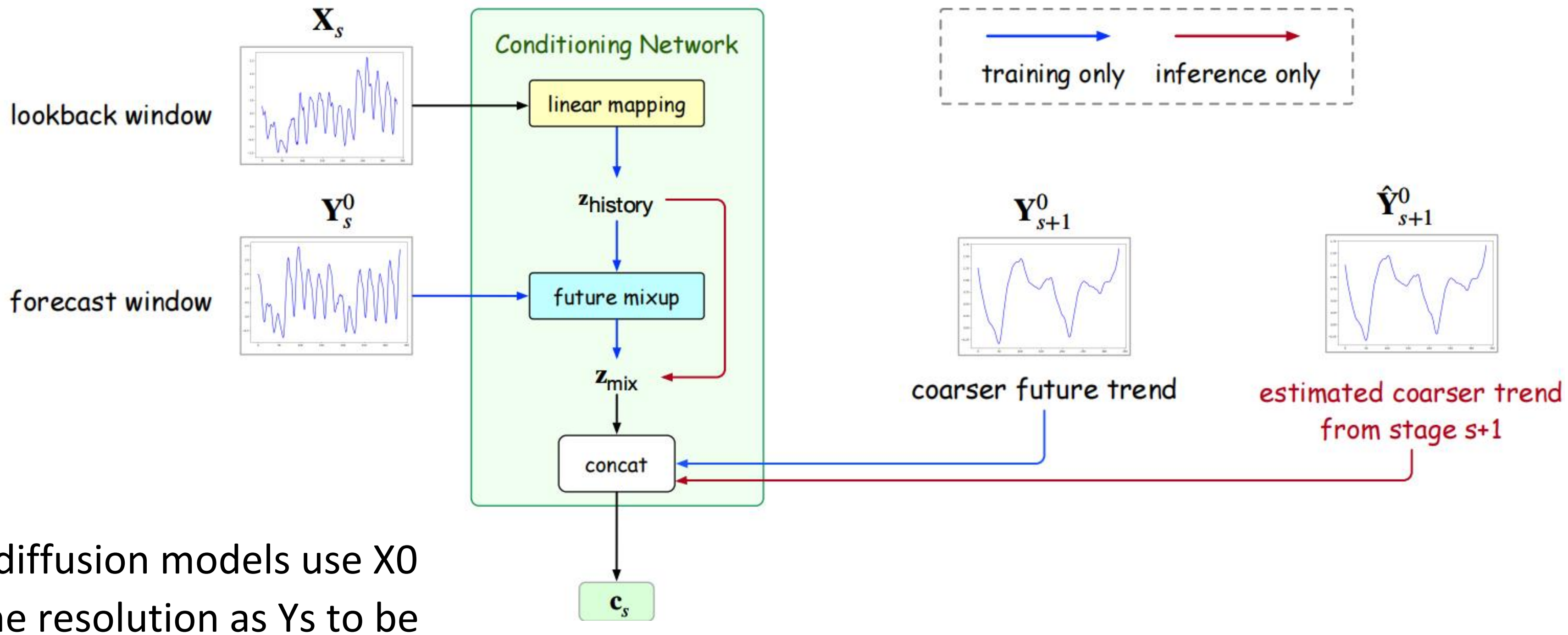
- no learnable parameters

# Backward denosing




- we perform progressive denoising in **an easy-to-hard manner**, generating coarser signals first and then finer details. This allows a more accurate prediction of time series.

# Conditioning network

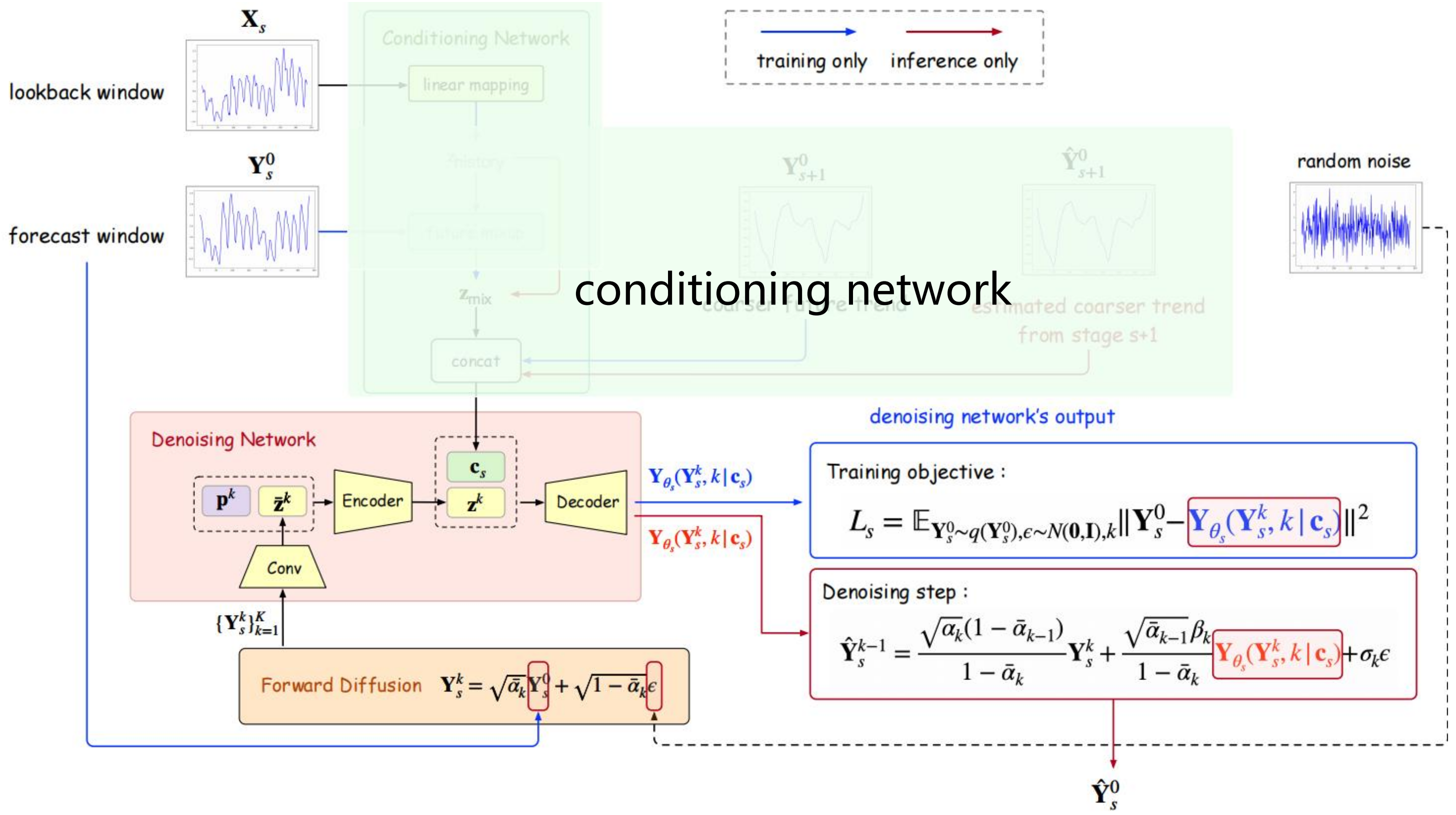



**Condition inputs:**

- **X**s (lookback segment)
  - ➢ existing time series diffusion models use X0
  - ➢ ours: Xs has the same resolution as Ys to be reconstructed
- **Y**s+1: (ground-truth) coarser trend
  - provides an overall picture of the finer Ys
- $\mathbf{Y}_S^0$: (ground-truth) future observation
  - future-mixup with $z_{history}$ (a linear mapping on Xs )
  - s = S (last stage): no coarser trend and cs is simply $z_{mix}$

**Future mixup:** combine past and future time series information $\mathbf{z}_{\texttt{mix}} = \mathbf{m} \odot \mathbf{z}_{\texttt{history}} + (1 - \mathbf{m}) \odot \mathbf{Y}_s^0$, where $\odot$ denotes the Hadamard product, and $\mathbf{m} \in [0, 1)^{d \times H}$.

# Denoising network

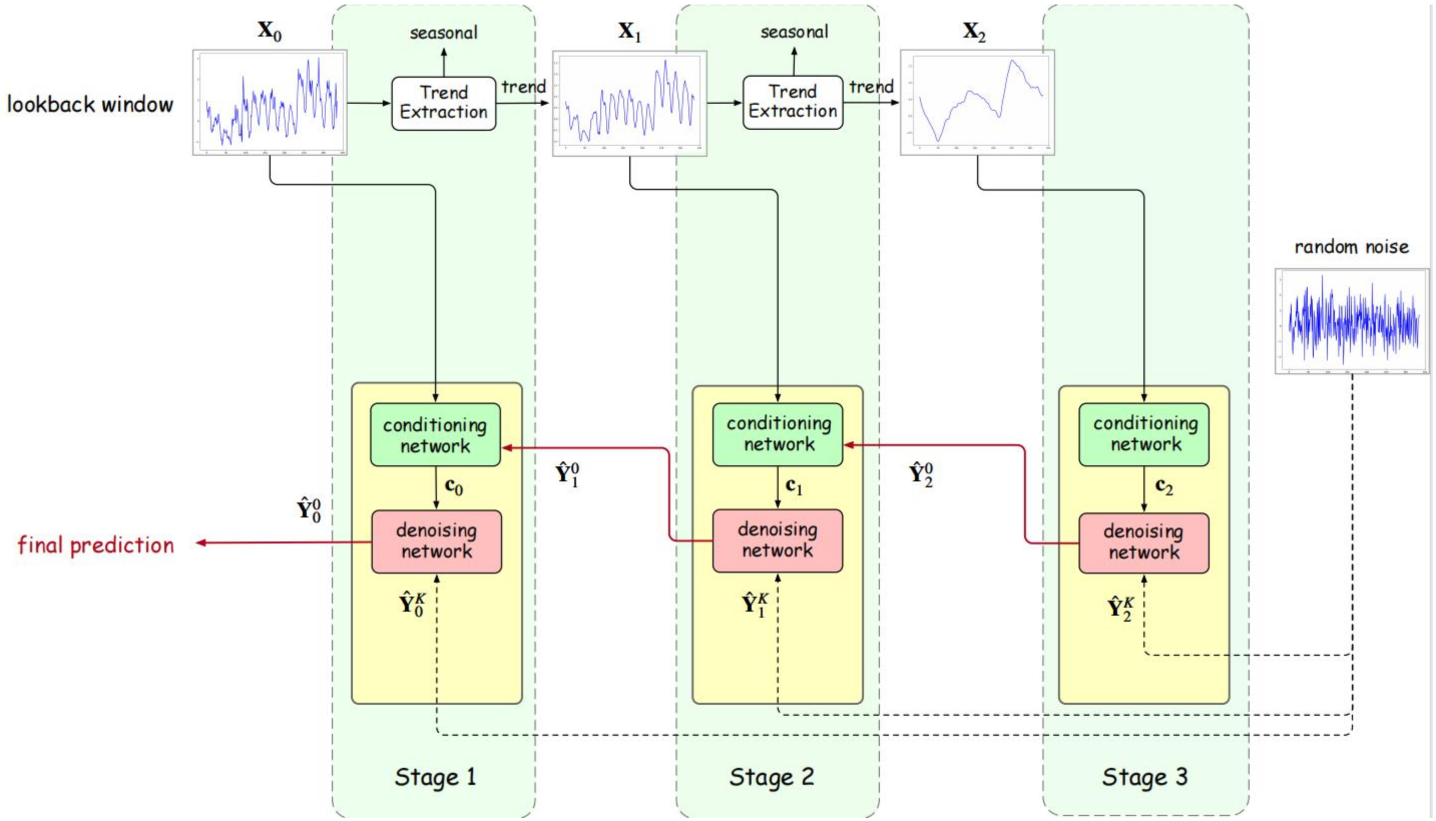



- map $\mathbf{Y}_S^k$ to the embedding by convolutional layers
- feed to an encoder
- concatenate with condition cs
- feed to a decoder, and output

- during training, one denoising objective one for each stage

$$\min_{\theta_s} \mathcal{L}_s(\theta_s) = \min_{\theta_s} \mathbb{E}_{\mathrm{Y}_s^0 \sim q(\mathrm{Y}_s), \epsilon \sim \mathcal{N}(0,\mathrm{I}), k} \left\| \mathrm{Y}_s^0 - \mathrm{Y}_{\theta_s}(\mathrm{Y}_s^k, k | \mathrm{c}_s) \right\|^2$$

# Inference



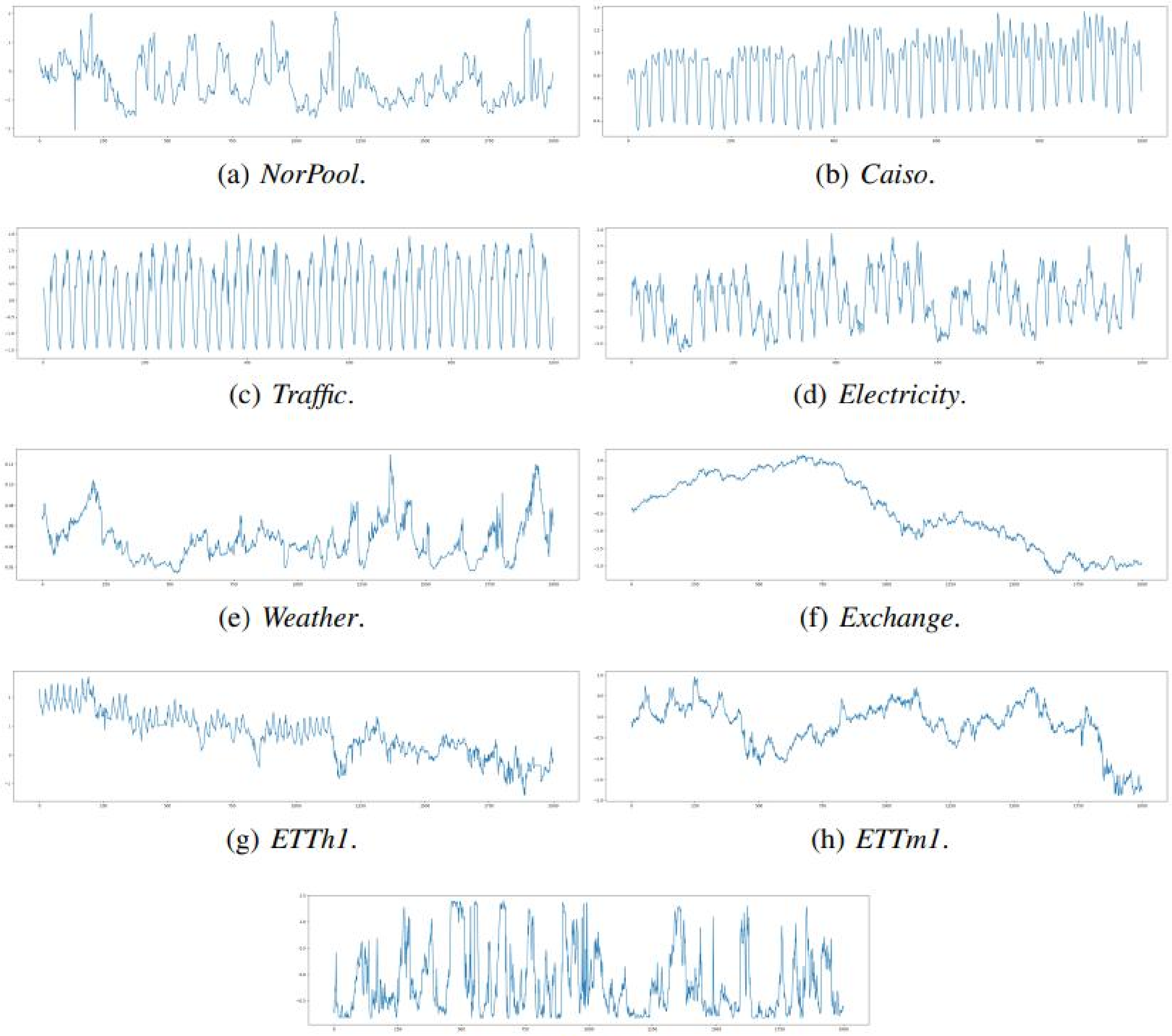

for each $s = S, \ldots, 1$, start from $\hat{\mathbf{Y}}_s^K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

- encourages denoising to proceed in an easy-to-hard manner
- coarser trends are generated first, finer details are progressively added
- reconstruction at stage 1 corresponds to the target time series forecast

# Datasets

Summary of dataset statistics, including dimension, total observations, sampling frequency, and prediction length.

| | dimension | #observations | frequency | steps ($H$) |
|---|---|---|---|---|
| *NorPool* | 18 | 70,128 | 1 hour | 720 (1 month) |
| *Caiso* | 10 | 74,472 | 1 hour | 720 (1 month) |
| *Traffic* | 862 | 17,544 | 1 hour | 168 (1 week) |
| *Electricity* | 321 | 26,304 | 1 hour | 168 (1 week) |
| *Weather* | 21 | 52,696 | 10 mins | 672 (1 week) |
| *Exchange* | 8 | 7,588 | 1 day | 14 (2 weeks) |
| *ETTh1* | 7 | 17,420 | 1 hour | 168 (1 week) |
| *ETTm1* | 7 | 69,680 | 15 mins | 192 (2 days) |
| *Wind* | 7 | 48,673 | 15 mins | 192 (2 days) |



(a) *NorPool*. (b) *Caiso*. (c) *Traffic*. (d) *Electricity*. (e) *Weather*. (f) *Exchange*. (g) *ETTh1*. (h) *ETTm1*. (i) *Wind*.

# Baselines

(i) recent time series diffusion models: non-autoregressive diffusion model **TimeDiff** (Shen & Kwok, 2023), **TimeGrad** (Rasul et al., 2021), conditional score-based diffusion model for imputation (**CSDI**) (Tashiro et al., 2021), structured state space model-based diffusion (**SSSD**) (Alcaraz & Strodthoff, 2022);

(ii) recent generative models for time series prediction: variational autoencoder with diffusion, denoise and disentanglement (**D3VAE**) (Li et al., 2022), coherent probabilistic forecasting (**CPF**) (Rangapuram et al., 2023), and **PSA-GAN** (Jeha et al., 2022);

(iii) recent prediction models based on basis expansion: **NHits** (Challu et al., 2023), frequency improved Legendre memory model (**FiLM**) (Zhou et al.,2022a), **Depts** (Fan et al., 2022) and **NBeats** (Oreshkin et al., 2019);

(iv) time series transformers: **Scaleformer** (Shabani et al., 2023), **PatchTST** (Nie et al., 2022), **Fedformer** (Zhou et al., 2022b), **Autoformer** (Wu et al., 2021), **Pyraformer** (Liu et al., 2021), **Informer** (Zhou et al., 2021) and the standard **Transformer** (Vaswani et al., 2017); and

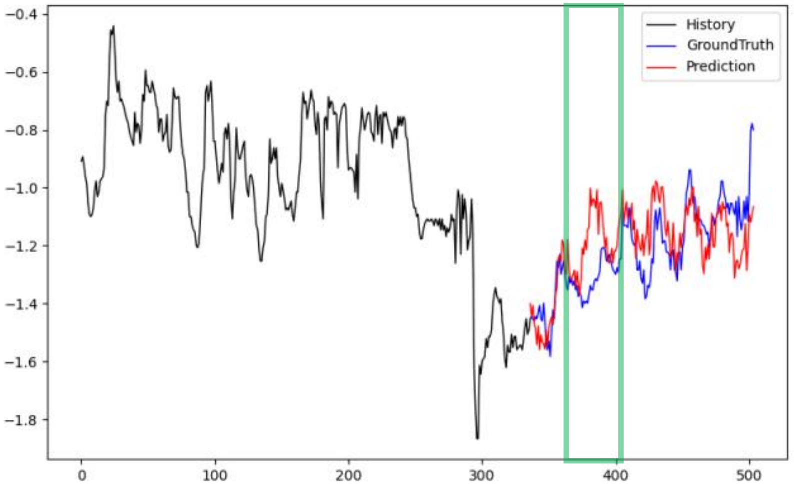
(v) other competitive baselines: **SCINet** (Liu et al., 2022) that introduces sample convolution and interaction for time series prediction, **NLinear** (Zeng et al., 2023), **DLinear** (Zeng et al., 2023) **LSTMa** (Bahdanau et al., 2015), and an attention-based **LSTM** (Hochreiter & Schmidhuber, 1997).

# MAE on univariate time series

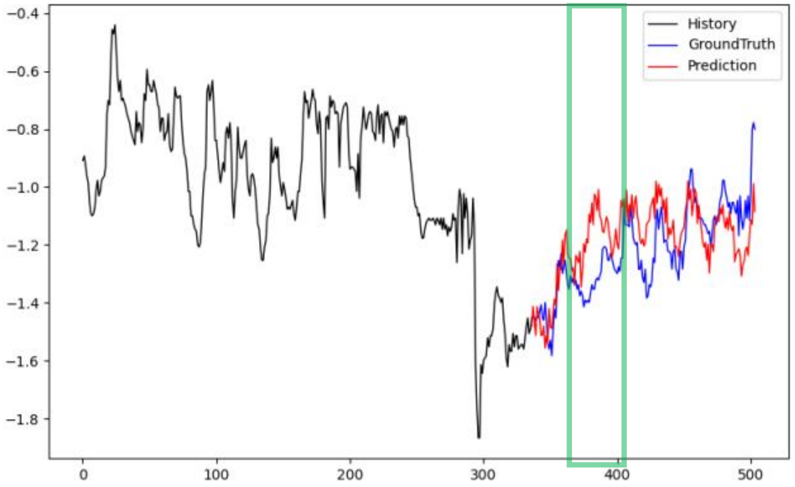
| | *NorPool* | *Caiso* | *Traffic* | *Electricity* | *Weather* | *Exchange* | *ETTh1* | *ETTm1* | *Wind* | avg rank |
|---|---|---|---|---|---|---|---|---|---|---|
| **mr-Diff** | 0.609$_{(2)}$ | 0.212$_{(4)}$ | 0.197$_{(2)}$ | **0.332**$_{(1)}$ | **0.032**$_{(1)}$ | **0.094**$_{(1)}$ | **0.196**$_{(1)}$ | **0.149**$_{(1)}$ | 1.168$_{(2)}$ | 1.7 |
| TimeDiff | 0.613$_{(3)}$ | 0.209$_{(3)}$ | 0.207$_{(3)}$ | 0.341$_{(3)}$ | 0.035$_{(4)}$ | 0.102$_{(7)}$ | 0.202$_{(2)}$ | 0.154$_{(6)}$ | 1.209$_{(5)}$ | 4.0 |
| TimeGrad | 0.841$_{(23)}$ | 0.386$_{(22)}$ | 0.894$_{(23)}$ | 0.898$_{(23)}$ | 0.036$_{(6)}$ | 0.155$_{(21)}$ | 0.212$_{(8)}$ | 0.167$_{(12)}$ | 1.239$_{(11)}$ | 16.6 |
| CSDI | 0.763$_{(20)}$ | 0.282$_{(14)}$ | 0.468$_{(20)}$ | 0.540$_{(19)}$ | 0.037$_{(7)}$ | 0.200$_{(23)}$ | 0.221$_{(12)}$ | 0.170$_{(14)}$ | 1.218$_{(7)}$ | 15.1 |
| SSSD | 0.770$_{(21)}$ | 0.263$_{(12)}$ | 0.226$_{(6)}$ | 0.403$_{(8)}$ | 0.041$_{(11)}$ | 0.118$_{(16)}$ | 0.250$_{(20)}$ | 0.169$_{(13)}$ | 1.356$_{(22)}$ | 14.3 |
| $D^3$VAE | 0.774$_{(22)}$ | 0.613$_{(23)}$ | 0.237$_{(9)}$ | 0.539$_{(18)}$ | 0.039$_{(9)}$ | 0.107$_{(13)}$ | 0.221$_{(12)}$ | 0.160$_{(9)}$ | 1.321$_{(19)}$ | 14.9 |
| CPF | 0.710$_{(15)}$ | 0.338$_{(18)}$ | 0.385$_{(19)}$ | 0.592$_{(21)}$ | 0.035$_{(4)}$ | 0.094$_{(1)}$ | 0.221$_{(12)}$ | 0.153$_{(5)}$ | 1.256$_{(12)}$ | 11.9 |
| PSA-GAN | 0.623$_{(5)}$ | 0.250$_{(8)}$ | 0.355$_{(17)}$ | 0.373$_{(6)}$ | 0.139$_{(22)}$ | 0.109$_{(14)}$ | 0.225$_{(16)}$ | 0.174$_{(16)}$ | 1.287$_{(16)}$ | 13.3 |
| N-Hits | 0.646$_{(7)}$ | 0.276$_{(13)}$ | 0.232$_{(7)}$ | 0.419$_{(9)}$ | 0.033$_{(2)}$ | 0.100$_{(5)}$ | 0.228$_{(17)}$ | 0.157$_{(8)}$ | 1.256$_{(12)}$ | 8.9 |
| FiLM | 0.654$_{(9)}$ | 0.290$_{(15)}$ | 0.315$_{(14)}$ | 0.362$_{(5)}$ | 0.069$_{(14)}$ | 0.104$_{(10)}$ | 0.210$_{(6)}$ | **0.149**$_{(1)}$ | 1.189$_{(3)}$ | 8.6 |
| Depts | 0.616$_{(4)}$ | **0.205**$_{(1)}$ | 0.241$_{(10)}$ | 0.434$_{(12)}$ | 0.102$_{(19)}$ | 0.106$_{(12)}$ | 0.202$_{(2)}$ | 0.165$_{(10)}$ | 1.472$_{(23)}$ | 10.3 |
| NBeats | 0.671$_{(10)}$ | 0.228$_{(5)}$ | 0.225$_{(5)}$ | 0.439$_{(13)}$ | 0.130$_{(21)}$ | 0.096$_{(3)}$ | 0.242$_{(18)}$ | 0.165$_{(10)}$ | 1.236$_{(9)}$ | 10.4 |
| Scaleformer | 0.687$_{(12)}$ | 0.320$_{(16)}$ | 0.375$_{(18)}$ | 0.430$_{(10)}$ | 0.083$_{(17)}$ | 0.148$_{(19)}$ | 0.302$_{(22)}$ | 0.210$_{(22)}$ | 1.348$_{(21)}$ | 17.4 |
| PatchTST | **0.590**$_{(1)}$ | 0.260$_{(11)}$ | 0.269$_{(11)}$ | 0.478$_{(17)}$ | 0.098$_{(18)}$ | 0.111$_{(15)}$ | 0.260$_{(21)}$ | 0.174$_{(16)}$ | 1.338$_{(20)}$ | 14.4 |
| FedFormer | 0.725$_{(17)}$ | 0.254$_{(9)}$ | 0.278$_{(12)}$ | 0.453$_{(14)}$ | 0.057$_{(13)}$ | 0.168$_{(22)}$ | 0.212$_{(8)}$ | 0.195$_{(20)}$ | 1.271$_{(14)}$ | 14.3 |
| Autoformer | 0.755$_{(19)}$ | 0.339$_{(19)}$ | 0.495$_{(21)}$ | 0.623$_{(22)}$ | 0.040$_{(10)}$ | 0.152$_{(20)}$ | 0.220$_{(11)}$ | 0.174$_{(16)}$ | 1.319$_{(18)}$ | 17.3 |
| Pyraformer | 0.747$_{(18)}$ | 0.257$_{(10)}$ | 0.215$_{(4)}$ | 0.455$_{(15)}$ | 0.107$_{(20)}$ | 0.104$_{(10)}$ | 0.211$_{(7)}$ | 0.179$_{(19)}$ | 1.284$_{(15)}$ | 13.1 |
| Informer | 0.698$_{(13)}$ | 0.345$_{(20)}$ | 0.308$_{(13)}$ | 0.433$_{(11)}$ | 0.069$_{(14)}$ | 0.118$_{(16)}$ | 0.212$_{(8)}$ | 0.172$_{(15)}$ | 1.236$_{(9)}$ | 13.2 |
| Transformer | 0.723$_{(16)}$ | 0.345$_{(20)}$ | 0.336$_{(16)}$ | 0.469$_{(16)}$ | 0.071$_{(16)}$ | 0.103$_{(9)}$ | 0.247$_{(19)}$ | 0.196$_{(21)}$ | 1.212$_{(6)}$ | 15.4 |
| SCINet | 0.653$_{(8)}$ | 0.244$_{(7)}$ | 0.322$_{(15)}$ | 0.377$_{(7)}$ | 0.037$_{(7)}$ | 0.101$_{(6)}$ | 0.205$_{(5)}$ | 0.150$_{(4)}$ | **1.167**$_{(1)}$ | 6.7 |
| NLinear | 0.637$_{(6)}$ | 0.238$_{(6)}$ | **0.192**$_{(1)}$ | 0.334$_{(2)}$ | 0.033$_{(2)}$ | 0.097$_{(4)}$ | 0.203$_{(4)}$ | **0.149**$_{(1)}$ | 1.197$_{(4)}$ | 3.3 |
| DLinear | 0.671$_{(10)}$ | 0.206$_{(2)}$ | 0.236$_{(8)}$ | 0.348$_{(4)}$ | 0.310$_{(23)}$ | 0.102$_{(7)}$ | 0.222$_{(15)}$ | 0.155$_{(7)}$ | 1.221$_{(8)}$ | 9.3 |
| LSTMa | 0.707$_{(14)}$ | 0.333$_{(17)}$ | 0.757$_{(22)}$ | 0.557$_{(20)}$ | 0.053$_{(12)}$ | 0.136$_{(18)}$ | 0.332$_{(23)}$ | 0.239$_{(23)}$ | 1.298$_{(17)}$ | 18.4 |

- mr-Diff is the best in 5 of the 9 datasets on remaining 4 datasets,
- mr-Diff ranks second in 3 of them

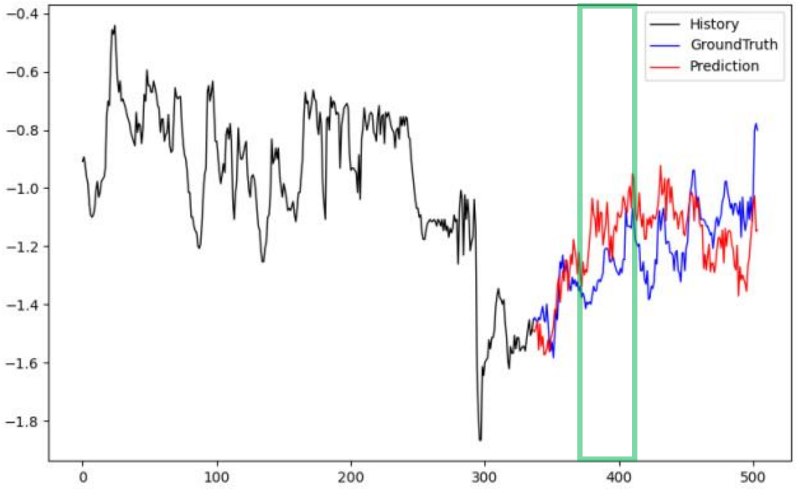
# Example prediction results on ETTh1



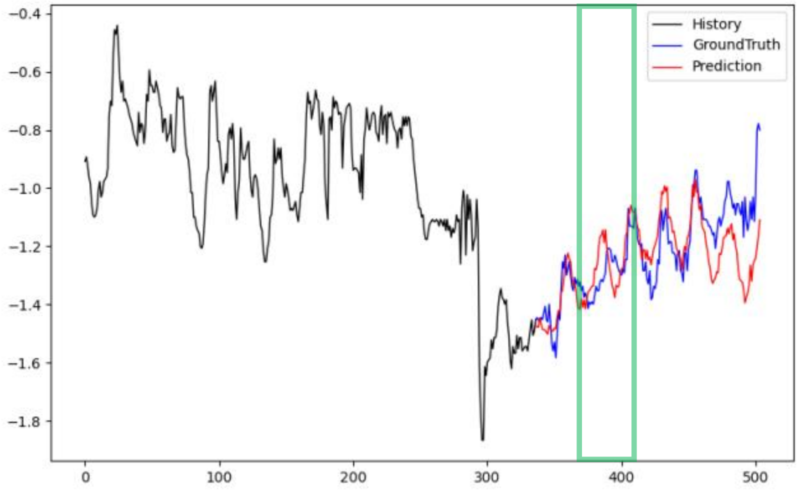
(a) SCINet.



(b) NLinear.



(c) TimeDiff.



(d) mr-Diff.

By progressively denoising the time series in a coarse-to-fine manner, mr-Diff produces higher-quality predictions than the others.

# Conclusions

- mutli-resolution diffusion (**mr-Diff**), a new cascaded diffusion model for time series
- incorporates seasonal-trend decomposition and uses multiple temporal resolutions in both the diffusion and denoising processes
- progressive denoising the time series in a coarse-to-fine manner

  → more reliable predictions
- experiments show that mr-Diff outperforms the state-of-the-art time series diffusion models