BRL Seminar

( 2024. 03. 05. Tue )

# Diffusion Model with Time Series Data 2

통합과정 8학기 이승한

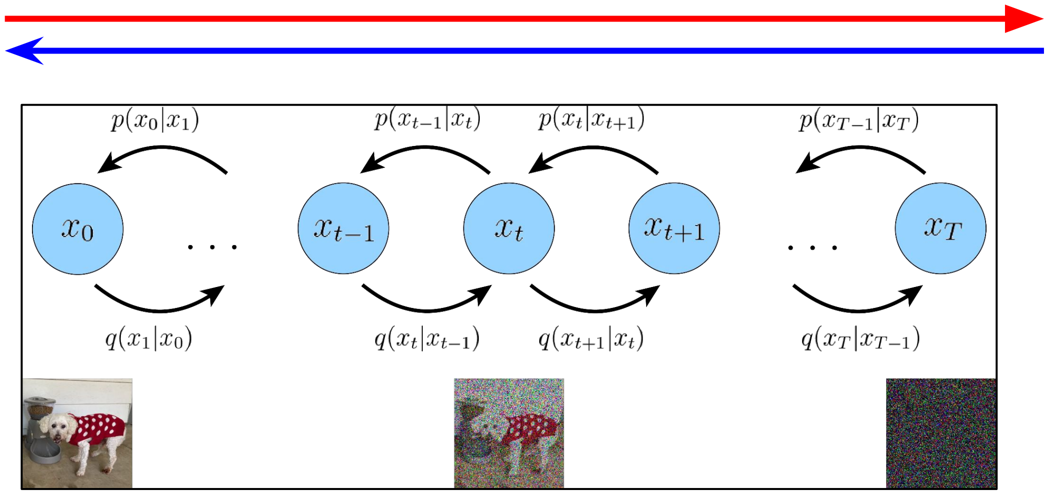
# Contents

# 1. Preliminaries: Time-series Diffusion Model

# 1. Preliminaries: Time-series Diffusion Model

## 1-1. Diffusion Model

- **Forward** Process: **Add** Noise
- **Backward** Process: **Remove** Noise



# 1. Preliminaries: Time-series Diffusion Model

## 1-1. Diffusion Model

- **Forward** Process: **Add** Noise

$$q\left(\mathrm{x}_t \mid \mathrm{x}_{t-1}\right) = N\left(\mathrm{x}_t; \sqrt{1-\beta_t}\mathrm{x}_{t-1}, \beta_t \mathrm{I}\right)$$

- **Backward** Process: **Remove** Noise

$$p_\theta\left(\mathrm{x}_{t-1} \mid \mathrm{x}_t\right) = N\left(\mathrm{x}_{t-1}; \mu_\theta\left(\mathrm{x}_t, t\right), \Sigma_\theta\left(\mathrm{x}_t, t\right)\right)$$

$\beta_t$ controls the strength of the noise → **Noise Scheduling**

ex) DDPM: Linear Scheduling $T = 1000, \beta_0 = 0.0001, \beta_{1000} = 0.02$

# 1. Preliminaries: Time-series Diffusion Model

## 1-2. Time-series Diffusion Model: TimeGrad (ICML 2021)

- Diffusion model for **TS forecasting**
- Conditional diffusion model
  - **"condition = past information"**

$$\Pi_{t=t_0}^{T} p_\theta\left(\mathbf{x}_t^0 \mid \boxed{\mathbf{h}_{t-1}}\right)$$

=

$$\boxed{\mathbf{h}_t = \mathrm{RNN}_\theta\left(\mathrm{concat}\left(\mathbf{x}_t^0, \mathbf{c}_t\right), \mathbf{h}_{t-1}\right)}$$

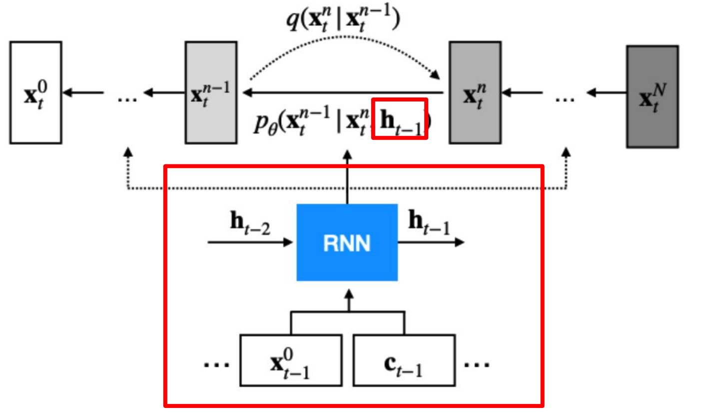
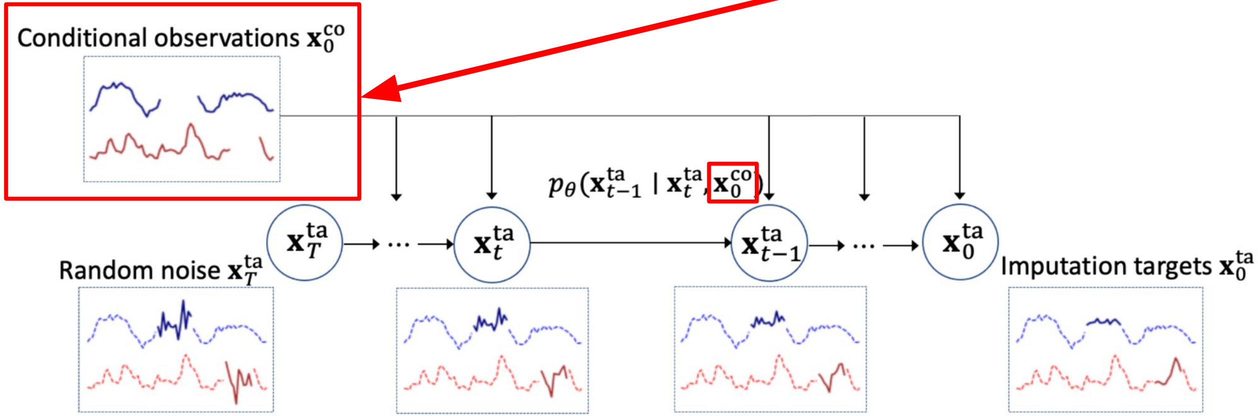



*Figure 1.* TimeGrad schematic: an RNN *conditioned* diffusion probabilistic model at some time $t-1$ depicting the fixed forward process that adds Gaussian noise and the learned reverse processes.

# 1. Preliminaries: Time-series Diffusion Model

## 1-2. Time-series Diffusion Model: CSDI (NeurIPS 2021)

- Diffusion model for **TS imputation**
- Conditional diffusion model, where **"condition = observed values"**



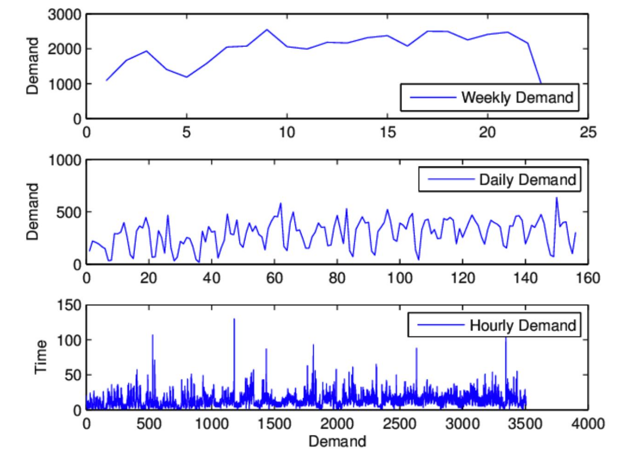
# 1. Preliminaries: Time-series Diffusion Model

## 1-3. Multi-granularity & Multi-resolution

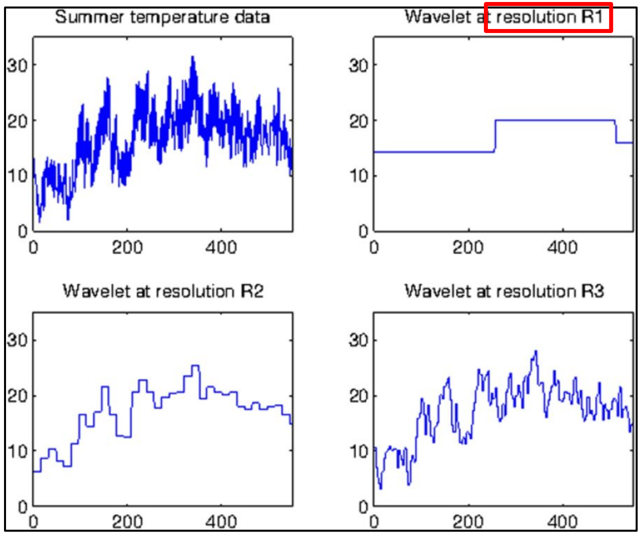
TS domain에서 **아래의 개념들은 비슷한 맥락**

- Granularity: 데이터 분할의 정도
- Resolution: 해상도

- Multi-granularity
- Multi-resolution
- Multi-level
- Hierarchy
- Global & Local
- (Decomposition)




Time series with different granularity. The top one is aggregated weekly, the middle on is aggregated daily, the bottom one is aggregated hourly.



# 1. Preliminaries: Time-series Diffusion Model

## 1-3. Multi-granularity & Multi-resolution

TS domain에서 **아래의 개념들은 비슷한 맥락**

- Multi-granularity
- Multi-resolution
- Multi-level
- Hierarchy
- Global & Local

***시계열 패턴을 coarse & fine 구분하여 포착***

- (Decomposition) ➡ trend를 거시적 & seasonality를 미시적 관점에서 보기도 함!

# 1. Preliminaries: Time-series Diffusion Model

## 1-3. Multi-granularity & Multi-resolution

ICLR 2024에 accept 된 4편의 **TS Diffusion Paper**

TS에서의 “Interpretable”
= 99% TS decomposition

| | | | | |
|---|---|---|---|---|
| ★ | 2024 | MG-TSD | MG-TSD: Multi-Granuality Time Series Diffusion Models with Guided Learning Process | ICLR |
| ★ | 2024 | mr-Diff | Multi-resolution Diffusion Models for Time-Series Forecasting | ICLR |
| | 2024 | Diffusion-TS | Diffusion-TS: Interpretable Diffusion for General Time Series Generation | ICLR |
| | 2024 | TMDM | **Transformer-Modulated Diffusion Models for Probabilistic Multivariate Time Series For** | ICLR |

**multi-resolution을 잡아내겠다는 Goal**은 동일.

이를 포착하는 방법은 알고리즘 별로 상이.

→ 현재 진행 중인 연구에서 제안하는 방법론 또한 **multi-resolution을 잡아내고자 함.**

# 2. MG-TSD: **Multi-Granularity** Time Series Diffusion Models with Guided Learning Process (ICLR 2024)

# 2. MG-TSD

이전까지의 TS Diffusion 연구에서는, **시계열을 분해/계층화** 하는 접근이 없었음!

MG-TSD (Multi-Granularity TS Diffusion)

- Model that considers **granularity level** within TS
- Utilize “**coarse-grained data**” across various granularity levels
- MG-TSD
    - (1) MG-TSD architecture
    - (2) Guided diffusion process module

TS Notation: $\boldsymbol{X}^{(1)} = [\boldsymbol{x}_1^1, \ldots, \boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_T^1]$, where $\boldsymbol{x}_t \in \mathbb{R}^D$

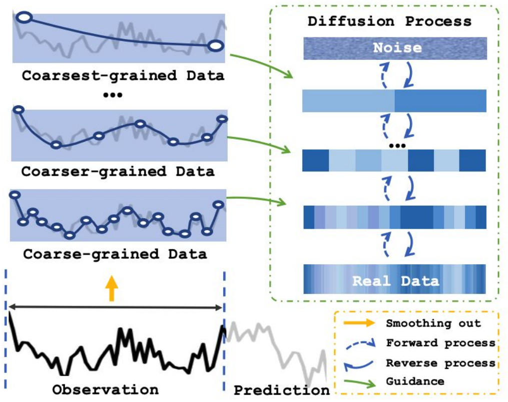



Figure 1: The process of smoothing data from finest-grained to coarsest-grained naturally aligns with the diffusion process.

# 2. MG-TSD

$\boxed{c = f(x_{0:t})}$

$\boxed{\hat{x_{t+1}} = g(c, n)}$

- $f$ : Encoder
- $g$ : Diffusion model
- Time step ($t$) = $1 \sim T$
- Diffusion step ($n$) = $1 \sim N$

Output = t+1 시점 (Autoregressive)

Input = 1~t시점

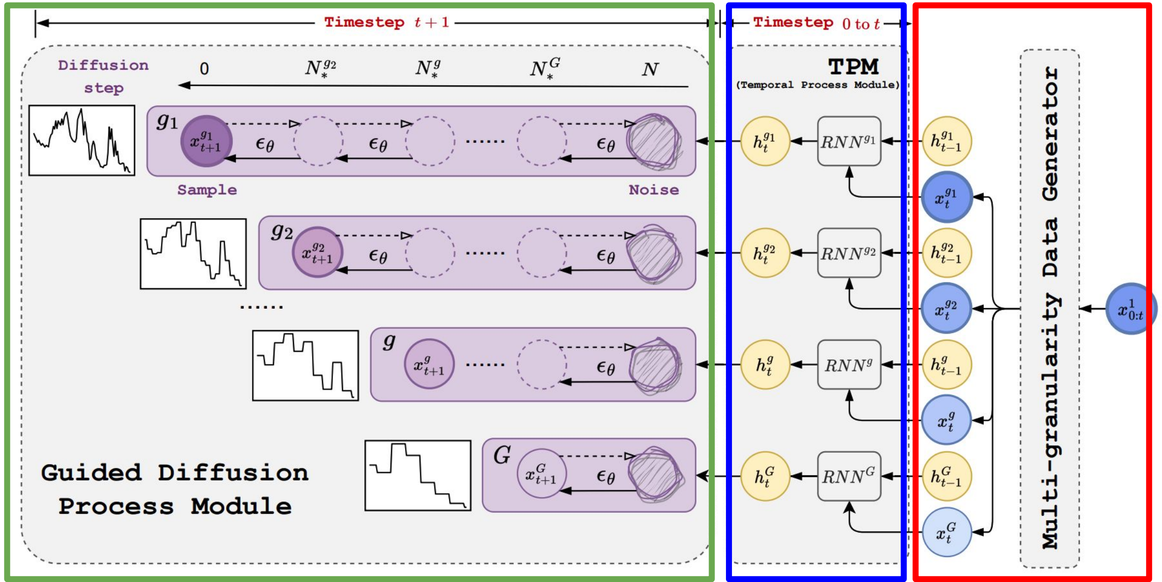



Figure 2: Overview of the Multi-Granularity Time Series Diffusion (MG-TSD) model, consisting of three key modules: **Multi-granularity Data Generator**, **Temporal Process Module (TPM)**, and **Guided Diffusion Process Module** for time series forecasting at a specific granularity level.

잠재 공간 상 Diffusion

<Step 3>

임베딩 (RNN)

<Step 2>

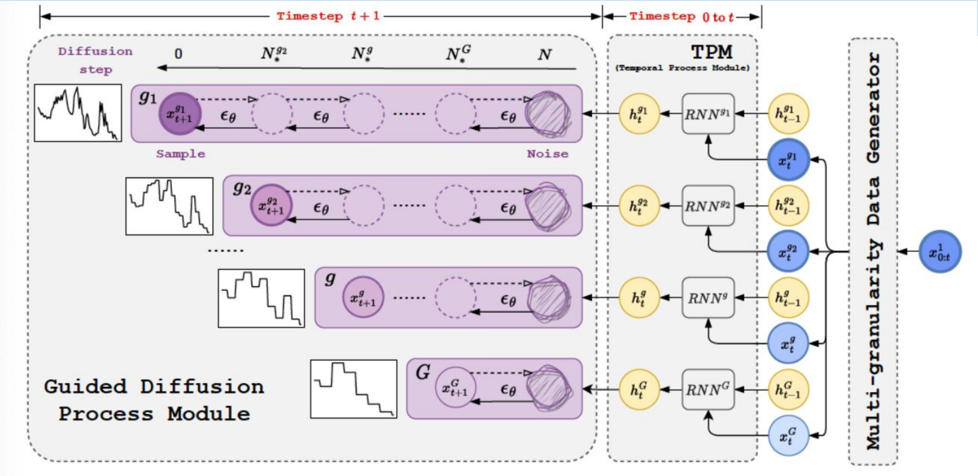
Multi-granularity 데이터 생성

<Step 1>

# 2. MG-TSD

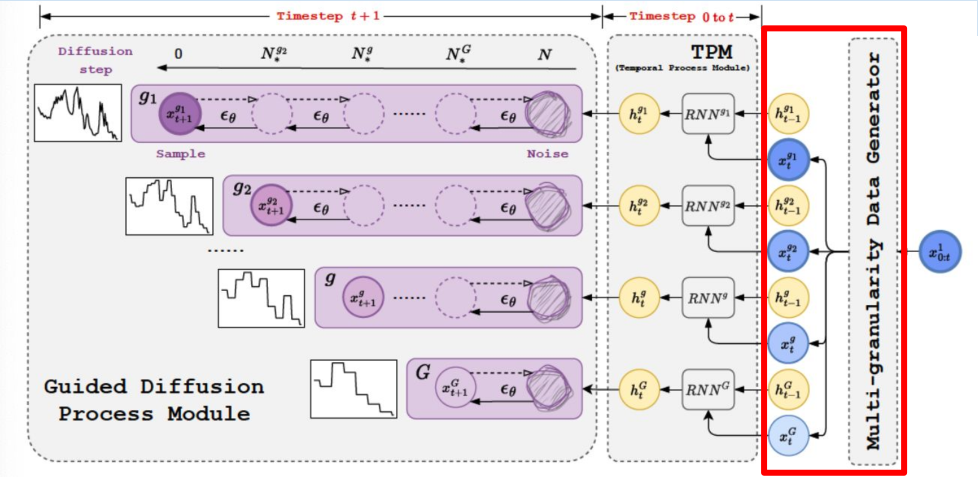
## (1) MG-TSD architecture

- Composed of three parts
  - a) Multi-granularity data generator
  - b) Temporal process module
  - c) Guided diffusion process module



# 2. MG-TSD

## (1) MG-TSD architecture



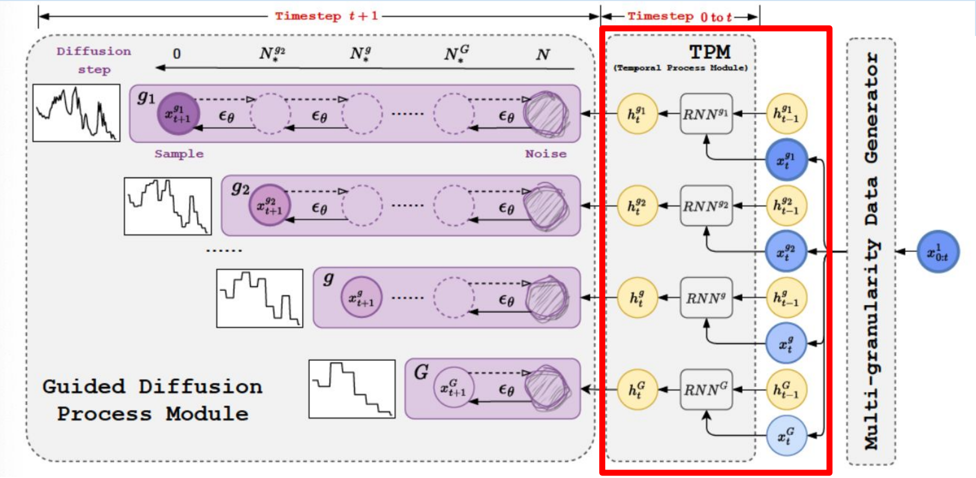

- Composed of three parts
  - **a) Multi-granularity data generator**
    - generate multi-granularity data
    - with sliding window + smoothing function (ex. MA)
    - w/o overlapping! -> replicate for same length
  - b) Temporal process module
  - c) Guided diffusion process module

$$\boldsymbol{X}^{(g)} = f\left(\boldsymbol{X}^{(1)}, s^{g}\right)$$

# 2. MG-TSD

## (1) MG-TSD architecture

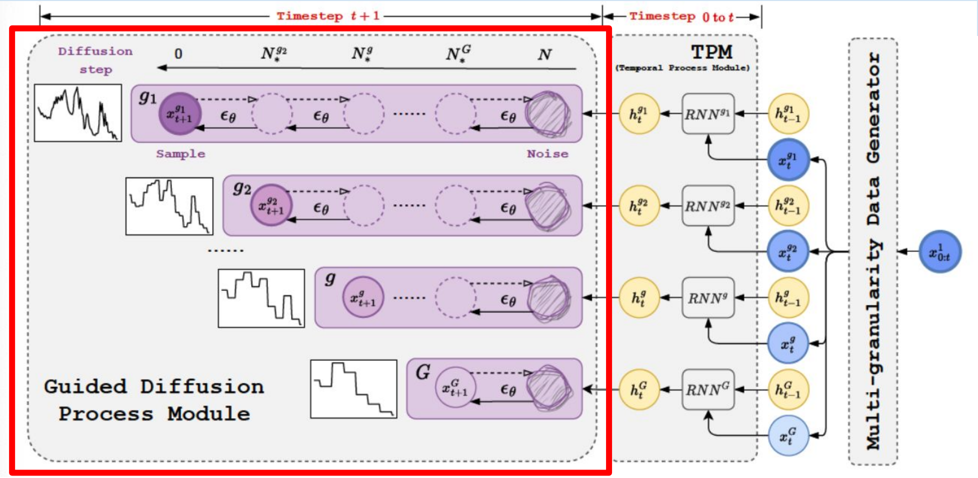
- Composed of three parts
    - a) Multi-granularity data generator
    - **b) Temporal process module**
        - capture temporal relationships
        - use RNN for “each granularity level”
    - c) Guided diffusion process module




Encoded hidden states : $\mathbf{h}_t^g$

# 2. MG-TSD

## (1) MG-TSD architecture

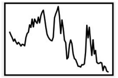


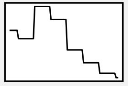

- Composed of three parts
  - a) Multi-granularity data generator
  - b) Temporal process module
  - **c) Guided diffusion process module**
    - Goal: make TS prediction
    - For guidance, use multi-granularity data

# 2. MG-TSD

## (2) Multi-granularity guided diffusion

Notation

- (**Finest**-grained) $\boldsymbol{x}_t^{g_1}(g_1 = 1) \ldots.$ from $\boldsymbol{X}^{(g_1)}$ 

  (**Coarse**-grained) $x_t^{g} \ldots\ldots\ldots\ldots\ldots\ldots\ldots.$ from $\boldsymbol{X}^{(g)}$ 
- Variance schedule: $\{\beta_n^1 = 1 - \alpha_n^1 \in (0,1)\}_{n=1}^N$

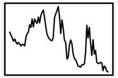
# 2. MG-TSD

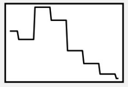
## (2) Multi-granularity guided diffusion

Granularity level ( 1(fine) ~ G(coarse) )

Time step (t 시점)

Notation

- (**Finest**-grained) $\boldsymbol{x}_t^{g_1}(g_1 = 1)\ldots.$ from $\boldsymbol{X}^{(g_1)}$ 

  (**Coarse**-grained) $x_t^g \ldots\ldots\ldots\ldots\ldots\ldots\ldots$ from $\boldsymbol{X}^{(g)}$ 

- Variance schedule: $\{\beta_n^1 = 1 - \alpha_n^1 \in (0,1)\}_{n=1}^N$

# 2. MG-TSD

## (2) Multi-granularity guided diffusion

Notation

Granularity level ( 1(fine) ~ G(coarse) )

Time step (t 시점)

1로 고정 (무시 가능)

Diffusion step ( 1~N )

- (**Finest**-grained) $\boldsymbol{x}_t^{g_1}(g_1 = 1) \ldots.$ from $\boldsymbol{X}^{(g_1)}$

  (**Coarse**-grained) $x_t^g \ldots\ldots\ldots\ldots\ldots\ldots\ldots$ from $\boldsymbol{X}^{(g)}$
- Variance schedule: $\{\beta_n^1 = 1 - \alpha_n^1 \in (0,1)\}_{n=1}^N$

# 2. MG-TSD

## (2) Multi-granularity guided diffusion

Notation

Granularity level ( 1(fine) ~ G(coarse) )

Time step (t 시점)

1로 고정 (무시 가능)

Diffusion step ( 1~N )

- (**Finest**-grained) $\boldsymbol{x}_{\boxed{t}}^{\boxed{g_1}}(g_1 = 1) \ldots .$ from $\boldsymbol{X}^{(g_1)}$

  (**Coarse**-grained) $x_t^g \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$ from $\boldsymbol{X}^{(g)}$

- Variance schedule: $\{\beta_{\boxed{n}}^{\boxed{1}} = 1 - \alpha_n^1 \in (0,1)\}_{n=1}^N$

Goal: approximate the distn $\underline{q(\boldsymbol{x}^{\boxed{g_1}})}$

가장 fine 한 (즉, original scale)의 data distn

# 2. MG-TSD

## (2) Multi-granularity guided diffusion

Notation

Granularity level ( 1(fine) ~ G(coarse) )

Time step (t 시점)

1로 고정 (무시 가능)

Diffusion step ( 1~N )

- (**Finest**-grained) $\boldsymbol{x}_t^{\boxed{g_1}} (g_1 = 1) \ldots.$ from $\boldsymbol{X}^{(g_1)}$

  (**Coarse**-grained) $x_t^g \ldots\ldots\ldots\ldots\ldots\ldots$ from $\boldsymbol{X}^{(g)}$
- Variance schedule: $\{\beta_n^1 = 1 - \alpha_n^1 \in (0,1)\}_{n=1}^N$

Goal: approximate the distn $\underline{q(\boldsymbol{x}^{\boxed{g_1}})}$

가장 **fine** 한 **(**즉**, original scale)**의 **data distn**

***Key Idea: Coarse한 정보를, Fine 한 정보에게 주입시키면서 denosing하면 도움이 될 것!***

# 2. MG-TSD

(논문 Notation 실수) 어쩔때는

- TS의 time step
- **Diffusion의 time step**

(2) Multi-granularity guided diffusion

$$x_0^{g_1} \sim q\left(x_{\boxed{0}}^{g_1}\right)$$ : Original Data

$$q\left(x_{0:N}^{g_1}\right)$$ : Forward

$$p_\theta\left(x_{0:N}^{g_1}\right)$$ : Backward

[Goal] Guide the generation of samples by ensuring that the **intermediate latent space retains the underlying time series structure**

[How] By introducing coarse-grained targets $x^g$ at intermediate diffusion step $N_*^g \in [1, N-1]$

# 2. MG-TSD

## (2) Multi-granularity guided diffusion

### Loss Function

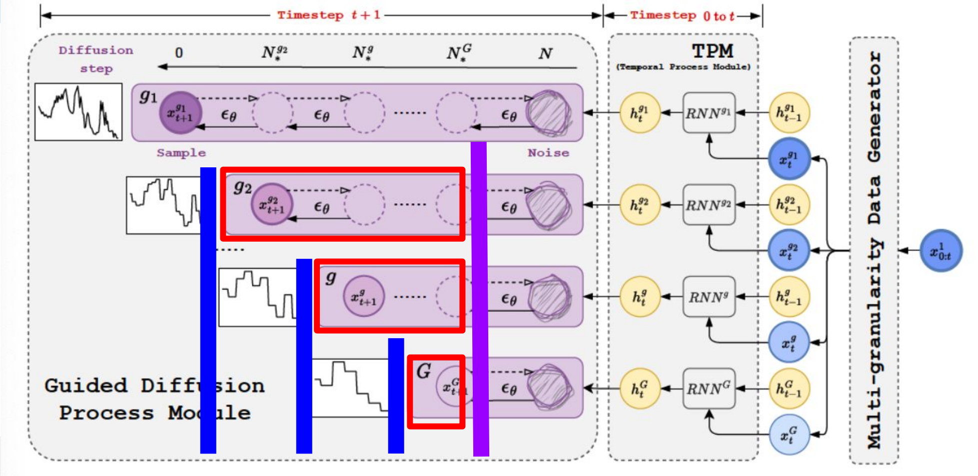
$$\mathbb{E}_{\epsilon,\boldsymbol{x}^g,n}\left[\|\ \epsilon-\epsilon_\theta\left(\boldsymbol{x}_n^g,n\right)\ \|^2\right].$$

- $\boxed{\boldsymbol{x}_n^g} = \left(\prod_{i=N_*^g}^{n}\alpha_i^1\right)x^g+\sqrt{1-\prod_{i=N_*^g}^{n}\alpha_i^1}\epsilon$ and $\epsilon\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$

n번째 diffusion step의 g번째 granularity

# 2. MG-TSD

## (2) Multi-granularity guided diffusion



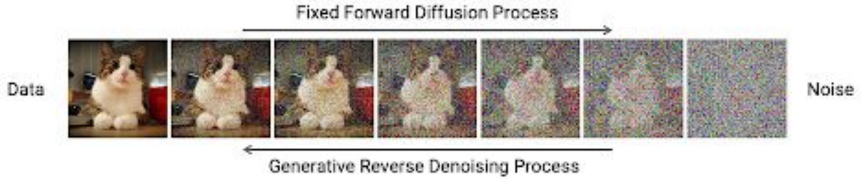

### Loss Function

$$\mathbb{E}_{\epsilon, \boldsymbol{x}^g, n}\left[\| \epsilon - \epsilon_\theta\left(\boldsymbol{x}_n^g, n\right) \|^2\right].$$

- $\boldsymbol{x}_n^g = \left(\prod_{i=N_*^g}^{n} \alpha_i^1\right) x^g + \sqrt{1 - \prod_{i=N_*^g}^{n} \alpha_i^1}\epsilon$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

Granularity level에 depend하는 noise schedule

# 2. MG-TSD




## (2) Multi-granularity guided diffusion

Shared Ratio $\square$ / $\square$

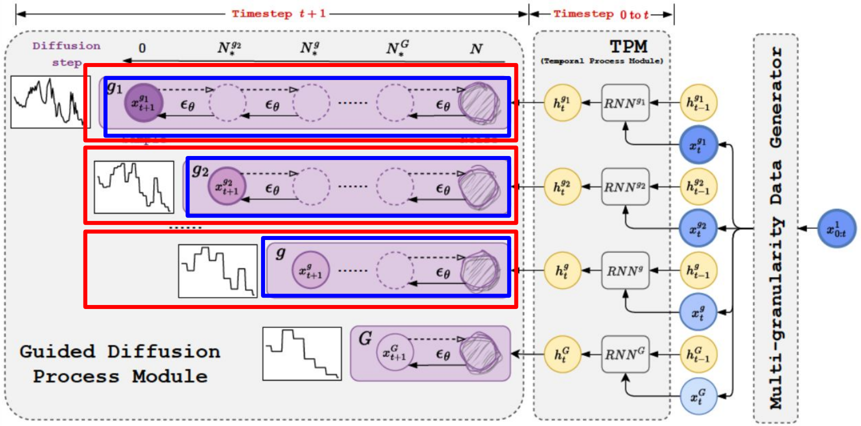
“Noise(Variance) schedule”을 share하는 비율

Share ratio: shared percentage of variance schedule between

- (1) the $\boxed{g \text{ th granularity}}$ data, where $g \in \{2, \ldots, G\}$
- (2) the $\boxed{\text{finest-grained}}$ data

→ Define it as $r_g := 1 - \left(N_*^g - 1\right)/N.$

- ex) For the finest-grained data, $N_*^1 = 1$ and $r^1 = 1.$



# 2. MG-TSD

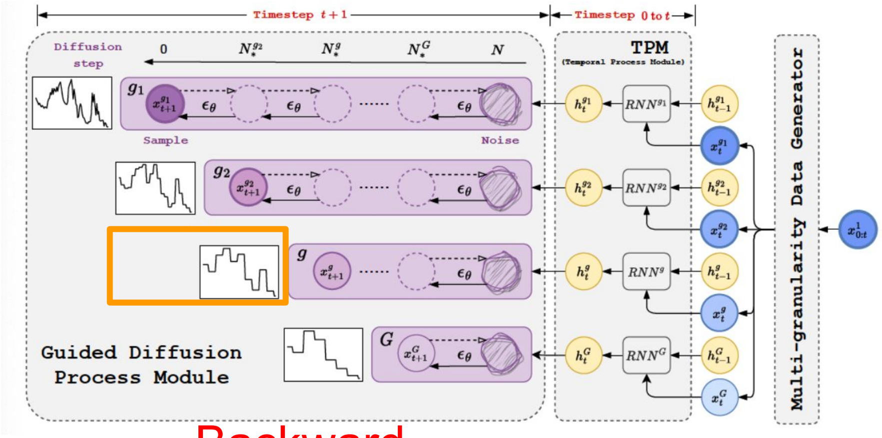
(2) Multi-granularity guided diffusion

Variance Schedule

Variance schedule for granularity $g$

$$\alpha_n^g\left(N_*^g\right) = \begin{cases} \boxed{1 \qquad \text{if } n = 1, \ldots, N_*^g} \\ \boxed{\alpha_n^1 \qquad \text{if } n = N_*^g + 1, \ldots, N} \end{cases}.$$

and $\boxed{\{\beta_n^g\}_{n=1}^N = \{1 - \alpha_n^g\}_{n=1}^N.}$




Backward

Forward

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0.1 | 0.2 | 0.3 | 0.4 | .. | 0.6 | 0.7 |
| 0 | 0 | 0 | 0 | 0.2 | 0.3 | 0.4 | .. | 0.6 | 0.7 |
| 0 | 0 | 0 | 0 | 0 | 0.3 | 0.4 | .. | 0.6 | 0.7 |

# 2. MG-TSD

## (2) Multi-granularity guided diffusion

Guidance Loss Function:

Guidance (X) : g=1 (finest)

Guidance (O) : g=1 (finest) + **g=2~G (coarse)**

Guidance loss function $L^{(g)}(\theta)$

- for $g$ th-granularity $\boldsymbol{x}_{n,t}^{g}$ at timestep $t$ and diffusion step $n$,
- $L^{(g)}(\theta) = \mathbb{E}_{\epsilon, \boldsymbol{x}_{0,t}^{g}, n} \; || \left( \epsilon - \epsilon_\theta \left( \sqrt{a_n^g} \boldsymbol{x}_{0,t}^{g} + \sqrt{b_n^g} \epsilon, n, \mathbf{h}_{t-1}^{g} \right) \right) ||_2^2.$
  - where $\mathbf{h}_t^g = \mathrm{RNN}_\theta \left( \boldsymbol{x}_t^g, \mathbf{h}_{t-1}^g \right)$

**Total Guidance loss function**

( with $G - 1$ granularity levels of data )

- $L^{\text{guidance}} = \sum_{g=2}^{G} \omega^g L^{(g)}(\theta),$

# 2. MG-TSD

(2) Multi-granularity guided diffusion

Guidance (X) : g=1 (finest)

Guidance (O) : **g=1 (finest)** + **g=2~G (coarse)**

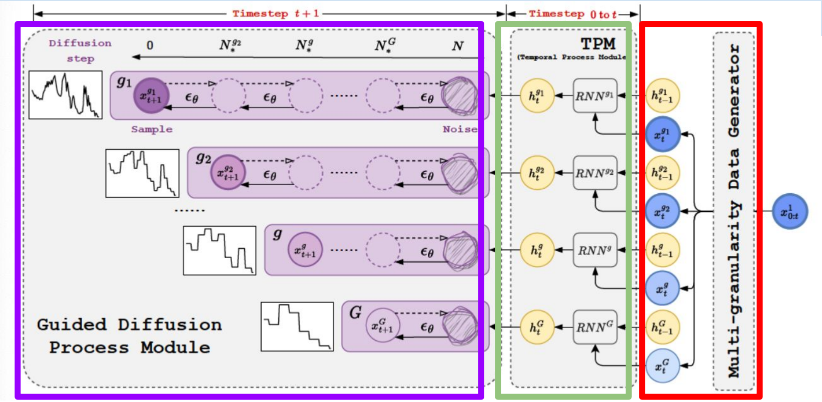
Guidance Loss Function:

$$L^{\text{final}} = \boxed{\omega^1 L^{(1)}(\theta)} + \boxed{L^{\text{guidance}}(\theta)}$$

$$= \sum_{g=1}^{G} \omega^g \mathbb{E}_{\boldsymbol{\epsilon}, \boldsymbol{x}_{0,t}^g, n} [\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{n,t}^g, n, \mathbf{h}_{t-1}^g) \|^2]$$

# 2. MG-TSD

## Training & Sampling




x Time step 수 (1~T까지)

**Algorithm 1** Training procedure

**Input:** Context interval $[1, t_0)$; prediction interval $[t_0, T]$; number of diffusion step $N$; a set of share ratio for $g$ granularity (or equivalently $\{N_*^g, g \in \{1, \ldots, G\}\}$); generated multi-granularity data $[\boldsymbol{x}_1^g, \ldots, \boldsymbol{x}_{t_0}^g, \ldots, \boldsymbol{x}_T^g], g \in \{1, \ldots, G\}$; initial hidden states $\mathbf{h}_0^g, g \in \{1, \ldots, G\}$

**repeat**

Sample the multi-granularity time series $[\boldsymbol{x}_1^g, \ldots, \boldsymbol{x}_T^g], g \in \{1, \ldots, G\}$

Obtain $\mathbf{h}_t^g = \mathrm{RNN}^g(\boldsymbol{x}_t^g, \mathbf{h}_{t-1}^g), g \in \{1, \ldots, G\}, t \in [1, \ldots, T]$

**for** $t = t_0$ to $T$ **do**

  Initialize $n \sim \mathrm{Uniform}(1, \ldots, N)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

  Reset the variance schedule $\{\beta_n^g = 1 - \alpha_n^g(N_*^g)\}_{n=1}^N, g \in \{1, \ldots, G\}$.

  Compute loss $L^{\text{final}}$ according to Equation 10

  Take the gradient $\nabla_\theta L^{\text{final}}$

**end for**

**until** converged

# 2. MG-TSD

## Training & Sampling

Reverse process

**Algorithm 2** Inference procedure for each timestep $t \in [t_0, T]$

**Input:** Noise $\boldsymbol{x}_t^N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and hidden states $\mathbf{h}_{t-1}^g$, $g \in \{1, \ldots, G\}$

**for** $\boxed{n = N \text{ to } 1}$ **do**
  **if** $n > 1$ **then**
    Sample $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  **else**
    $\boldsymbol{z} = \mathbf{0}$
  **end if**
  **for** $g = 1$ to $G$ **do**
    $\boldsymbol{x}_{n-1,t}^g = \frac{1}{\sqrt{\alpha_n^g}}\left(\boldsymbol{x}_{n,t}^g - \frac{\beta_n^g}{\sqrt{1-a_n^g}}\epsilon_\theta(\boldsymbol{x}_{n,t}^g, n, \mathbf{h}_{t-1}^g)\right) + \sqrt{\sigma_n^g}\boldsymbol{z}$, where $\sigma_n^g = \frac{1-a_{n-1}^g}{1-a_n^g}\beta_n^g$.
  **end for**
**end for**

**Return** $\boxed{\boldsymbol{x}_{0,t}^g, g = 1 \text{(finest-grained data)}}$; (Optional: $\boldsymbol{x}_{0,t}^g$, $g \in \{2, \ldots, G\}$)

Fine ~ Coarse 모두 생성 (순서 상관 X)

최종적으로 얻길 원하는 fine-granularity의 데이터

# 2. MG-TSD

## Experiments

### (1) Dataset

| Name | Frequency | Number of series | Context length | Prediction length | Multi-granularity dictionary |
|---|---|---|---|---|---|
| Solar | 1 hour | 137 | 24 | 24 | [1 hour, 4 hour, 12 hour, 24hour, 48 hour] |
| Electricity | 1 hour | 370 | 24 | 24 | [1 hour, 4 hour, 12 hour, 24 hour, 48 hour] |
| Traffic | 1 hour | 963 | 24 | 24 | [1 hour, 4 hour, 12 hour, 24 hour, 48 hour] |
| Taxi | 30 min | 1214 | 24 | 24 | [30 min , 2 hour, 6 hour, 12 hour, 24 hour] |
| KDD-cup | 1 hour | 270 | 48 | 48 | [1 hour, 4 hour, 12 hour, 24hour, 48 hour] |
| Wikipedia | 1 day | 2000 | 30 | 30 | [1 day, 4 day, 7 day, 14 day] |

Table 6: Detailed information of the datasets used in our benchmark including data frequency and number of times series (dimension), including the information about context length and prediction length and the multi-granularity dictionary utilized in the multivariate time series forecasting task.

# 2. MG-TSD

## Experiments

### (2) MTS forecasting

**Baselines.** We assess the predictive performance of the proposed **MG-TSD** model in comparison with multivariate time series forecasting models, including Vec-LSTM-ind-scaling (Salinas et al., 2019), GP-scaling (Salinas et al., 2019), GP-Copula (Salinas et al., 2019), Transformer-MAF (Rasul et al., 2020), LSTM-MAF (Rasul et al., 2020), $\boxed{\text{TimeGrad (Rasul et al., 2021),}}$ and TACTiS (Drouin et al., 2022). The MG-Input ensemble model serves as the baseline with multi-granularity inputs. It

- **TS Diffusion Model**은 하나 뿐!
- 비교적 오래된 알고리즘과의 비교

  ( 다른 2023,2024 TS Diffusion 논문들도 마찬가지 경향성 O.

  가장 대표적인 **TimeGrad (2021), CSDI (2021)**과만 비교하는 경우가 종종 있음

Table 1: Comparison of $\text{CRPS}_{\text{sum}}$ (smaller is better) of models on six real-world datasets. The reported mean and standard error are obtained from 10 re-training and evaluation independent runs.

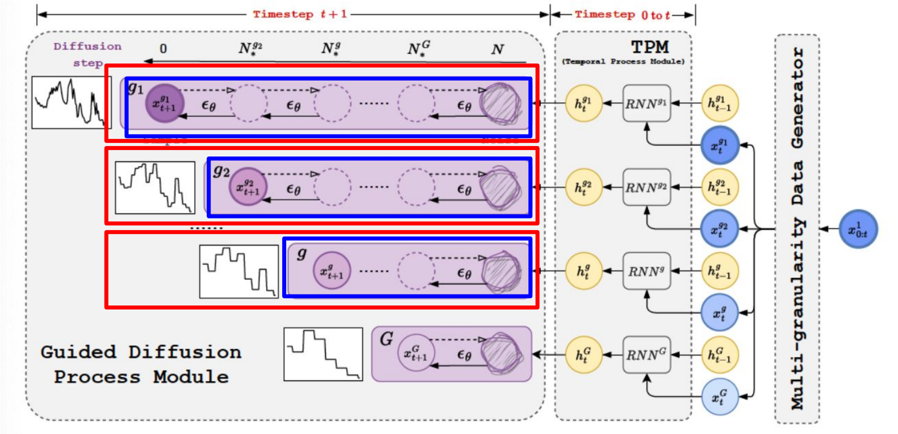
| Method | Solar | Electricity | Traffic | KDD-cup | Taxi | Wikipedia |
|---|---|---|---|---|---|---|
| Vec-LSTM ind-scaling | $0.4825_{\pm 0.0027}$ | $0.0949_{\pm 0.0175}$ | $0.0915_{\pm 0.0197}$ | $0.3560_{\pm 0.1667}$ | $0.4794_{\pm 0.0343}$ | $0.1254_{\pm 0.0174}$ |
| GP-Scaling | $0.3802_{\pm 0.0052}$ | $0.0499_{\pm 0.0031}$ | $0.0753_{\pm 0.0152}$ | $0.2983_{\pm 0.0448}$ | $0.2265_{\pm 0.0210}$ | $0.1351_{\pm 0.0612}$ |
| GP-Copula | $0.3612_{\pm 0.0035}$ | $0.0287_{\pm 0.0005}$ | $0.0618_{\pm 0.0018}$ | $0.3157_{\pm 0.0462}$ | $0.1894_{\pm 0.0087}$ | $0.0669_{\pm 0.0009}$ |
| LSTM-MAF | $0.3427_{\pm 0.0082}$ | $0.0312_{\pm 0.0046}$ | $0.0526_{\pm 0.0021}$ | $0.2919_{\pm 0.1486}$ | $0.2295_{\pm 0.0082}$ | $0.0763_{\pm 0.0051}$ |
| Transformer-MAF | $0.3532_{\pm 0.0053}$ | $0.0272_{\pm 0.0017}$ | $0.0499_{\pm 0.0011}$ | $0.2951_{\pm 0.0504}$ | $0.1531_{\pm 0.0038}$ | $0.0644_{\pm 0.0037}$ |
| TimeGrad | $0.3335_{\pm 0.0653}$ | $0.0232_{\pm 0.0035}$ | $0.0414_{\pm 0.0112}$ | $0.2902_{\pm 0.2178}$ | $0.1255_{\pm 0.0207}$ | $0.0555_{\pm 0.0088}$ |
| TACTiS | $0.4209_{\pm 0.0330}$ | $0.0259_{\pm 0.0019}$ | $0.1093_{\pm 0.0076}$ | $0.5406_{\pm 0.1584}$ | $0.2070_{\pm 0.0159}$ | – |
| MG-Input | $0.3239_{\pm 0.0427}$ | $0.0238_{\pm 0.0035}$ | $0.0658_{\pm 0.0065}$ | $0.2977_{\pm 0.1163}$ | $0.1592_{\pm 0.0087}$ | $0.0567_{\pm 0.0091}$ |
| MG-TSD | $\mathbf{0.3081}_{\pm \mathbf{0.0099}}$ | $\mathbf{0.0149}_{\pm \mathbf{0.0017}}$ | $\mathbf{0.0323}_{\pm \mathbf{0.0125}}$ | $\mathbf{0.1837}_{\pm \mathbf{0.0865}}$ | $\mathbf{0.1159}_{\pm \mathbf{0.0132}}$ | $\mathbf{0.0529}_{\pm 0.0054}$ |

# 2. MG-TSD

## Experiments

### (3) Ablation Studies

### 3-1) Share-ratio




| Ratio | 4 hour | | | 6 hour | | |
|---|---|---|---|---|---|---|
| | $\text{CRPS}_{\text{sum}}$ | NMAE | NRMSE | $\text{CRPS}_{\text{sum}}$ | NMAE | NRMSE |
| **20%** | $0.3489_{\pm 0.0190}$ | $0.3826_{\pm 0.0200}$ | $0.7177_{\pm 0.0445}$ | $0.3378_{\pm 0.0305}$ | $0.3703_{\pm 0.0368}$ | $0.6916_{\pm 0.0536}$ |
| **40%** | $0.3405_{\pm 0.0415}$ | $0.3792_{\pm 0.0386}$ | $0.6870_{\pm 0.0870}$ | $0.3275_{\pm 0.0250}$ | $0.3608_{\pm 0.0267}$ | $0.6650_{\pm 0.0374}$ |
| **60%** | $0.3268_{\pm 0.0475}$ | $0.3604_{\pm 0.0463}$ | $0.6579_{\pm 0.0919}$ | $\mathbf{0.3166}_{\pm 0.0376}$ | $\mathbf{0.3491}_{\pm 0.0368}$ | $\mathbf{0.6478}_{\pm 0.0696}$ |
| **80%** | $\mathbf{0.3172}_{\pm \mathbf{0.0249}}$ | $\mathbf{0.3510}_{\pm \mathbf{0.0240}}$ | $\mathbf{0.6515}_{\pm \mathbf{0.051}}$ | $0.3221_{\pm 0.0425}$ | $0.3555_{\pm 0.0443}$ | $0.6542_{\pm 0.0747}$ |
| **100%** | $0.3178_{\pm 0.0342}$ | $0.3480_{\pm 0.0356}$ | $0.6591_{\pm 0.0503}$ | $0.3232_{\pm 0.0396}$ | $0.3548_{\pm 0.0417}$ | $0.6550_{\pm 0.0660}$ |
| **Ratio** | **12 hour** | | | **24 hour** | | |
| | $\text{CRPS}_{\text{sum}}$ | NMAE | NRMSE | $\text{CRPS}_{\text{sum}}$ | NMAE | NRMSE |
| **20%** | $0.3440_{\pm 0.0391}$ | $0.3767_{\pm 0.0450}$ | $0.6999_{\pm 0.0772}$ | $0.3315_{\pm 0.0266}$ | $0.3693_{\pm 0.0298}$ | $0.6801_{\pm 0.0354}$ |
| **40%** | $0.3374_{\pm 0.0370}$ | $0.3713_{\pm 0.0346}$ | $0.6837_{\pm 0.0641}$ | $\mathbf{0.3276}_{\pm \mathbf{0.0358}}$ | $\mathbf{0.3612}_{\pm \mathbf{0.0361}}$ | $\mathbf{0.6722}_{\pm \mathbf{0.0552}}$ |
| **60%** | $\mathbf{0.3240}_{\pm 0.0382}$ | $\mathbf{0.3597}_{\pm 0.0388}$ | $\mathbf{0.6694}_{\pm 0.0746}$ | $0.3382_{\pm 0.0343}$ | $0.3737_{\pm 0.0365}$ | $0.6878_{\pm 0.0655}$ |
| **80%** | $0.3391_{\pm 0.0390}$ | $0.3719_{\pm 0.0403}$ | $0.6953_{\pm 0.0691}$ | $0.3288_{\pm 0.0460}$ | $0.3639_{\pm 0.0476}$ | $0.6741_{\pm 0.0929}$ |
| **100%** | $0.3284_{\pm 0.0323}$ | $0.3538_{\pm 0.0450}$ | $0.6609_{\pm 0.0917}$ | $0.3407_{\pm 0.0248}$ | $0.3692_{\pm 0.0244}$ | $0.6933_{\pm 0.0528}$ |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0.1 | 0.2 | 0.3 | 0.4 | .. | 0.6 | 0.7 |
| 0 | 0 | 0 | 0 | 0.2 | 0.3 | 0.4 | .. | 0.6 | 0.7 |
| 0 | 0 | 0 | 0 | 0 | 0.3 | 0.4 | .. | 0.6 | 0.7 |

Table 2: Influence of share ratios for different granularities on Solar dataset. The reported mean and standard error are obtained from 10 re-training and evaluation independent runs.

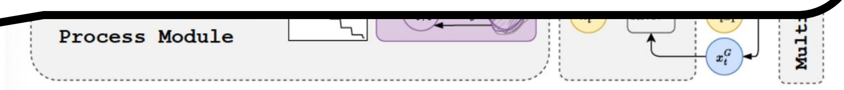
# 2. MG-TSD

## Experiments

### (3) Ablation Studie

For **COARSER** granularities, the model performs better with a **SMALLER** share ratio.

-> Model achieves optimal performance when the share ratio is chosen at the step where the coarse-grained samples **most closely resemble intermediate states.**




## 3-1) Share-ratio

Fine

| Ratio | 4 hour | | | 6 hour | | |
|---|---|---|---|---|---|---|
| | CRPS$_{sum}$ | NMAE | NRMSE | CRPS$_{sum}$ | NMAE | NRMSE |
| **20%** | $0.3489_{\pm 0.0190}$ | $0.3826_{\pm 0.0200}$ | $0.7177_{\pm 0.0445}$ | $0.3378_{\pm 0.0305}$ | $0.3703_{\pm 0.0368}$ | $0.6916_{\pm 0.0536}$ |
| **40%** | $0.3405_{\pm 0.0415}$ | $0.3792_{\pm 0.0386}$ | $0.6870_{\pm 0.0870}$ | $0.3275_{\pm 0.0250}$ | $0.3608_{\pm 0.0267}$ | $0.6650_{\pm 0.0374}$ |
| **60%** | $0.3268_{\pm 0.0475}$ | $0.3604_{\pm 0.0463}$ | $0.6579_{\pm 0.0919}$ | $\mathbf{0.3166}_{\pm 0.0376}$ | $\mathbf{0.3491}_{\pm 0.0368}$ | $\mathbf{0.6478}_{\pm 0.0696}$ |
| **80%** | $\mathbf{0.3172}_{\pm \mathbf{0.0249}}$ | $\mathbf{0.3510}_{\pm \mathbf{0.0240}}$ | $\mathbf{0.6515}_{\pm \mathbf{0.051}}$ | $0.3221_{\pm 0.0425}$ | $0.3555_{\pm 0.0443}$ | $0.6542_{\pm 0.0747}$ |
| **100%** | $0.3178_{\pm 0.0342}$ | $0.3480_{\pm 0.0356}$ | $0.6591_{\pm 0.0503}$ | $0.3232_{\pm 0.0396}$ | $0.3548_{\pm 0.0417}$ | $0.6550_{\pm 0.0660}$ |

Coarse

| Ratio | 12 hour | | | 24 hour | | |
|---|---|---|---|---|---|---|
| | CRPS$_{sum}$ | NMAE | NRMSE | CRPS$_{sum}$ | NMAE | NRMSE |
| **20%** | $0.3440_{\pm 0.0391}$ | $0.3767_{\pm 0.0450}$ | $0.6999_{\pm 0.0772}$ | $0.3315_{\pm 0.0266}$ | $0.3693_{\pm 0.0298}$ | $0.6801_{\pm 0.0354}$ |
| **40%** | $0.3374_{\pm 0.0370}$ | $0.3713_{\pm 0.0346}$ | $0.6837_{\pm 0.0641}$ | $\mathbf{0.3276}_{\pm \mathbf{0.0358}}$ | $\mathbf{0.3612}_{\pm \mathbf{0.0361}}$ | $\mathbf{0.6722}_{\pm \mathbf{0.0552}}$ |
| **60%** | $\mathbf{0.3240}_{\pm 0.0382}$ | $\mathbf{0.3597}_{\pm 0.0388}$ | $\mathbf{0.6694}_{\pm 0.0746}$ | $0.3382_{\pm 0.0343}$ | $0.3737_{\pm 0.0365}$ | $0.6878_{\pm 0.0655}$ |
| **80%** | $0.3391_{\pm 0.0390}$ | $0.3719_{\pm 0.0403}$ | $0.6953_{\pm 0.0691}$ | $0.3288_{\pm 0.0460}$ | $0.3639_{\pm 0.0476}$ | $0.6741_{\pm 0.0929}$ |
| **100%** | $0.3284_{\pm 0.0323}$ | $0.3538_{\pm 0.0450}$ | $0.6609_{\pm 0.0917}$ | $0.3407_{\pm 0.0248}$ | $0.3692_{\pm 0.0244}$ | $0.6933_{\pm 0.0528}$ |

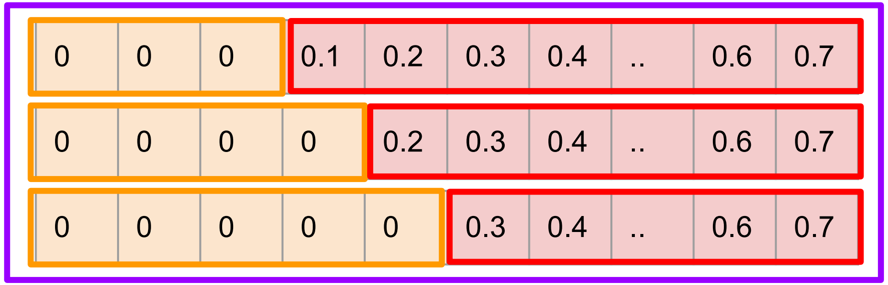
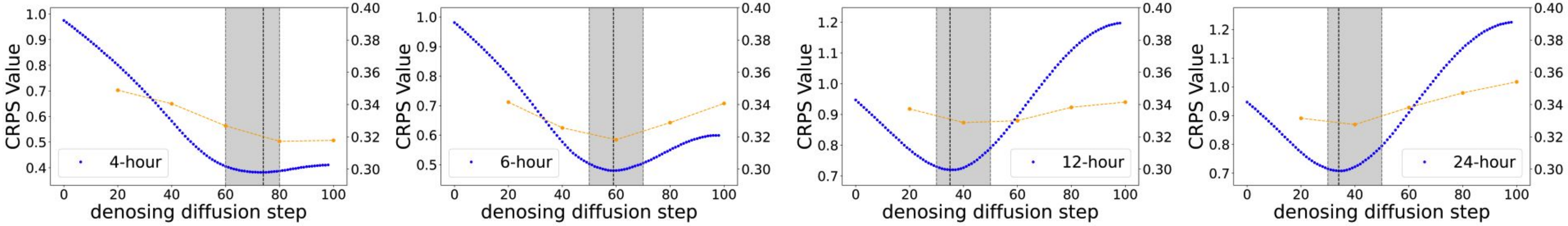



Table 2: Influence of share ratios for different granularities on Solar dataset. The reported mean and standard error are obtained from 10 re-training and evaluation independent runs.

# 2. MG-TSD

For **coarser** granularities, the model performs better with a **smaller** share ratio.




(a) $\text{CRPS}_{\text{sum}}$ values between 4-hour targets and samples
(b) $\text{CRPS}_{\text{sum}}$ values between 6-hour targets and samples
(c) $\text{CRPS}_{\text{sum}}$ values between 12-hour targets and samples
(d) $\text{CRPS}_{\text{sum}}$ values between 24h-granularity targets and samples

**Fine** ⟷ **Coarse**

Figure 3: Selection of share ratio for **MG-TSD** models

| | | | | | | |
|---|---|---|---|---|---|---|
| **20%** | $0.3440_{\pm 0.0391}$ | $0.3767_{\pm 0.0450}$ | $0.6999_{\pm 0.0772}$ | $0.3315_{\pm 0.0266}$ | $0.3693_{\pm 0.0298}$ | $0.6801_{\pm 0.0554}$ |
| **40%** | $0.3374_{\pm 0.0370}$ | $0.3713_{\pm 0.0340}$ | $0.6837_{\pm 0.0641}$ | $\mathbf{0.3276}_{\pm 0.0358}$ | $\mathbf{0.3612}_{\pm 0.0361}$ | $\mathbf{0.6722}_{\pm 0.0552}$ |
| **60%** | $\mathbf{0.3240}_{\pm 0.0382}$ | $\mathbf{0.3597}_{\pm 0.0388}$ | $\mathbf{0.6694}_{\pm 0.0746}$ | $0.3382_{\pm 0.0343}$ | $0.3737_{\pm 0.0365}$ | $0.6878_{\pm 0.0655}$ |
| **80%** | $0.3391_{\pm 0.0390}$ | $0.3719_{\pm 0.0403}$ | $0.6953_{\pm 0.0691}$ | $0.3288_{\pm 0.0460}$ | $0.3639_{\pm 0.0476}$ | $0.6741_{\pm 0.0929}$ |
| **100%** | $0.3284_{\pm 0.0323}$ | $0.3538_{\pm 0.0450}$ | $0.6609_{\pm 0.0917}$ | $0.3407_{\pm 0.0248}$ | $0.3692_{\pm 0.0244}$ | $0.6933_{\pm 0.0528}$ |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0.3 | 0.4 | .. | 0.6 | 0.7 |

Table 2: Influence of share ratios for different granularities on `Solar` dataset. The reported mean and standard error are obtained from 10 re-training and evaluation independent runs.

# 2. MG-TSD

## Experiments

### (3) Ablation Studies

## 3-1) Number of Granularity

Utilizing four to five granularity levels generally suffices

Table 3: Influence of the number of granularities on **MG-TSD** performance for `Solar` and `Electricity` Dataset.

| Num of gran | Solar | | | Electricity | | |
|---|---|---|---|---|---|---|
| | $\text{CRPS}_{\text{sum}}$ | NMAE | NRMSE | $\text{CRPS}_{\text{sum}}$ | NMAE | NRMSE |
| 2 | $0.3172_{\pm 0.0249}$ | $0.3510_{\pm 0.0240}$ | $0.6515_{\pm 0.0571}$ | $0.0174_{\pm 0.0042}$ | $0.0226_{\pm 0.0071}$ | $0.0296_{\pm 0.0086}$ |
| 3 | $0.3110_{\pm 0.0329}$ | $0.3494_{\pm 0.0378}$ | $0.6452_{\pm 0.0632}$ | $0.0160_{\pm 0.0020}$ | $0.0198_{\pm 0.0029}$ | $0.0262_{\pm 0.0039}$ |
| 4 | $\mathbf{0.3081}_{\pm 0.0099}$ | $\mathbf{0.3445}_{\pm 0.0102}$ | $\mathbf{0.6245}_{\pm 0.0268}$ | $\mathbf{0.0149}_{\pm \mathbf{0.0017}}$ | $\mathbf{0.0178}_{\pm \mathbf{0.0018}}$ | $\mathbf{0.0241}_{\pm \mathbf{0.0030}}$ |
| 5 | $0.3093_{\pm 0.0411}$ | $0.3430_{\pm 0.0451}$ | $0.6117_{\pm 0.0746}$ | $0.0153_{\pm 0.0027}$ | $0.0181_{\pm 0.0043}$ | $0.0254_{\pm 0.0058}$ |
| **Num of gran** | **Traffic** | | | **KDD-cup** | | |
| | $\text{CRPS}_{\text{sum}}$ | NMAE | NRMSE | $\text{CRPS}_{\text{sum}}$ | NMAE | NRMSE |
| 2 | $0.0347_{\pm 0.0020}$ | $0.0396_{\pm 0.0022}$ | $0.0593_{\pm 0.0043}$ | $0.2427_{\pm 0.1167}$ | $0.3171_{\pm 0.1557}$ | $0.3745_{\pm 0.1652}$ |
| 3 | $0.0334_{\pm 0.0034}$ | $0.0382_{\pm 0.0035}$ | $0.0574_{\pm 0.0066}$ | $0.2414_{\pm 0.1619}$ | $0.3030_{\pm 0.1789}$ | $0.3808_{\pm 0.2168}$ |
| 4 | $0.0326_{\pm 0.0041}$ | $0.0374_{\pm 0.0048}$ | $0.0573_{\pm 0.0050}$ | $0.2198_{\pm 0.1162}$ | $0.2893_{\pm 0.1554}$ | $0.3315_{\pm 0.1882}$ |
| 5 | $\mathbf{0.0323}_{\pm 0.0125}$ | $\mathbf{0.0370}_{\pm 0.0140}$ | $\mathbf{0.0563}_{\pm 0.0230}$ | $\mathbf{0.1837}_{\pm 0.0636}$ | $\mathbf{0.2463}_{\pm 0.0865}$ | $\mathbf{0.3001}_{\pm 0.0997}$ |

# 2. MG-TSD

## Experiments

### (4) Case Study

## Visualization (vs. TimeGrad)

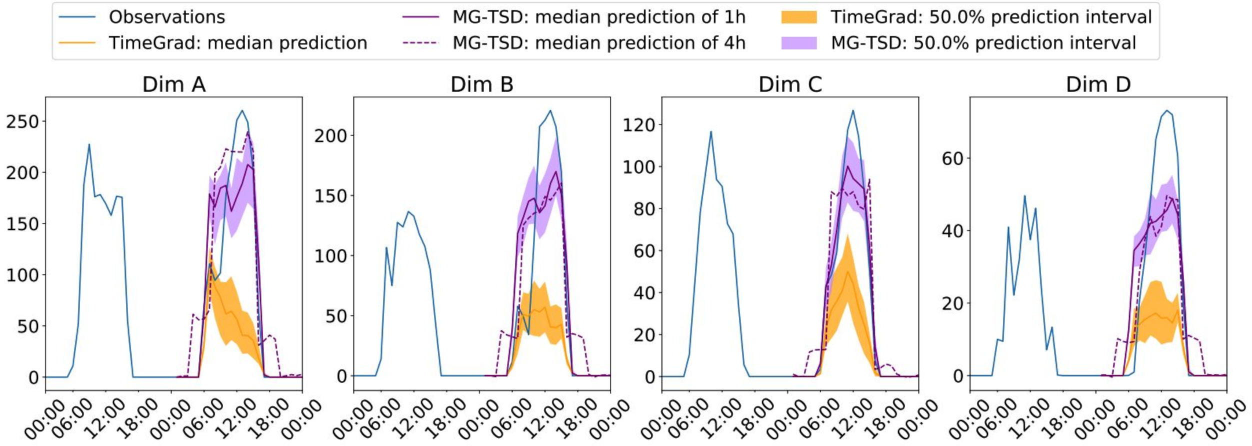
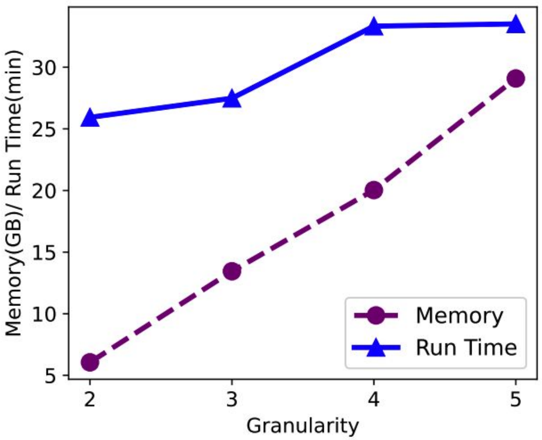



Figure 4: `Solar` dataset: visualization of the ground-truth, **MG-TSD** predicted mean for 4-hour and 1-hour time series, and TimeGrad predicted mean for the 1-hour time series. Additionally, the 50% prediction intervals for the 1-hour data are also included. These plots represent some illustrative dimensions out of 370 dimensions from the first 24-hour rolling-window.

# 2. MG-TSD

## Appendix




(1) Runtime & Memory Efficiency

| Dataset | Num gran | Gran dict | Share ratio | Loss weight |
|---|---|---|---|---|
| Solar<br>Electricity<br>Traffic<br>KDD-cup | 2 | [1h,4h]<br>[1h,12h] | [1,0.9]<br>[1,0.8]<br>[1,0.6] | [0.9,0.1] |
| | 3 | [1h,4h,12h]<br>[1h,4h,24h] | [1,0.9,0.8]<br>[1,0.8,0.8]<br>[1,0.8,0.6] | [0.8, 0.1, 0.1]<br>[0.9, 0.05, 0.05]<br>[0.85, 0.10, 0.05] |
| | 4 | [1h,4h,12h,24h]<br>[1h,4h,12h,48h] | [1,0.9,0.8,0.8]<br>[1,0.9,0.8,0.6]<br>[1,0.8,0.6,0.6]<br>[1,0.8,0.6,0.4] | [0.8, 0.1, 0.05, 0.05]<br>[0.7,0.1,0.1,0.1] |
| | 5 | [1h,4h,8h,12h,24h]<br>[1h,4h,12h,24h,48h] | [1,0.9,0.8,0.6,0.6]<br>[1,0.9,0.8,0.6,0.4]<br>[1,0.8,0.6,0.6,0.6]<br>[1,0.8,0.6,0.6,0.4]<br>[1,0.8,0.6,0.4,0.4] | [0.8,0.1,0.05,0.04,0.01]<br>[0.8,0.05,0.05,0.05,0.05]<br>[0.6,0.1,0.1,0.1,0.1] |
| Taxi | 2 | [30m,2h]<br>[30m,6h]<br>[30m,12h]<br>[30m,24h] | [1,0.8]<br>[1,0.6] | [0.9,0.1] |
| Wikipedia | 3 | [1d,4d]<br>[1d,7d]<br>[1d,14d] | [1,0.8]<br>[1,0.6] | [0.9,0.1] |

Table 7: Tested hyper-parameter values for the **MG-TSD** Model. The reported results in the paper are based on a parameter search within these choices.

(2) Hyperparameter Tuning

# 2. MG-TSD

## Summary

- TS를 **multi-granularity로 나눠서** diffuse한다는 점에서 novel (최초)
- But (상대적으로) **빈약한 실험 및 분석**
    - 적은 수의 실험 및 분석
    - 비교하는 baseline이 단순 ( 2021 TimeGrad, 2022 CSDI 위주 비교 )

      ( But 대다수의 TS Diffusion 논문들도 마찬가지. 데이터셋/세팅들이 상이해서 (?) )
- Openreview Rating: [6,6,6]

# 3. **Multi-resolution** Diffusion Model

# for Time-series Forecasting

# 3. mr-Diff: Multi-Resolution Diffusion Model

Motivation: TS data = different patterns at “**multiple scales**”

=> Why not utilize this “**multi-resolution**” temporal structure?

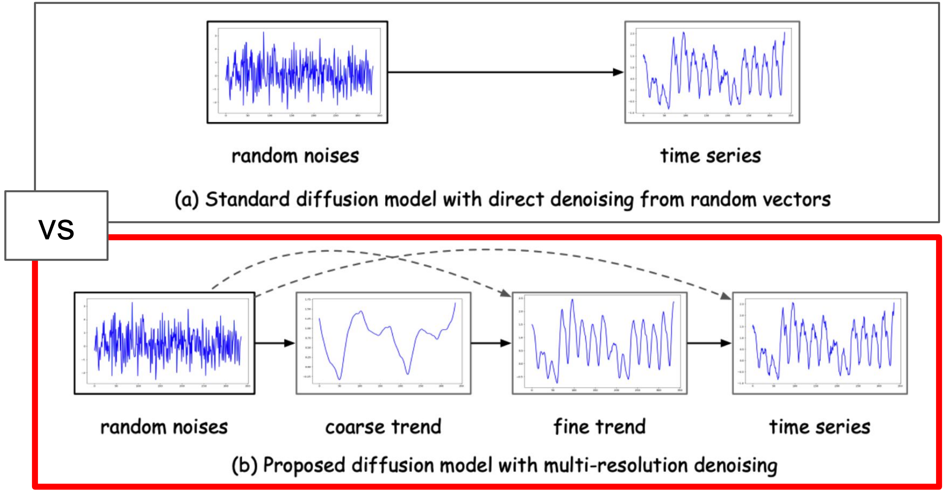
Propose mr-Diff

- (1) **Multi-resolution** diffusion model
- (2) **Seasonal-trend** decomposition ( Importance: **Trend** >> Seasonality )
- (3) Sequentially extract “**fine-to-coarse**” trends
    - 3-1) Coarsest trend first
    - 3-2) Finer details later! ( with coarse trends as condition )
- (4) **Non-autoregressive**

# 3. mr-Diff: Multi-Resolution Diffusion Model

Details & Contributions of mr-Diff

- 1) Decompose the denoising objective into **several sub-objectives**
- 2) First work to integrate "**seasonal-trend decomposition**" based **multi-resolution** analysis
- 3) **Progressive** denoising ( coarse -> fine )



# 3. mr-Diff: Multi-Resol

**Notation**

- $\mathbf{X} = \mathbf{x}_{-L+1:0}$ and $\mathbf{Y} = \mathbf{x}_{1:H}$
- Trend component of the lookback/forecast segment at stage $s + 1$ be $\mathbf{X}_s$ / $\mathbf{Y}_s$
  - **Trend gets coarser** as $s$ increases     granularity level와 같은 개념
- $\mathbf{X}_0 = \mathbf{X}$ and $\mathbf{Y}_0 = \mathbf{Y}$.

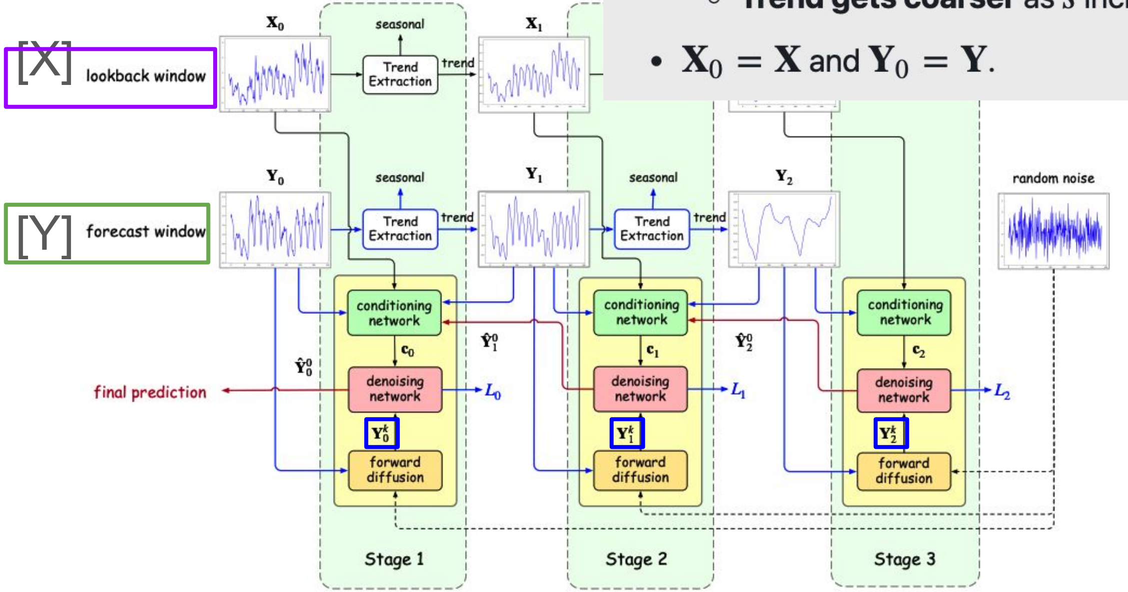



Figure 2: The proposed multi-resolution diffusion model mr-Diff. For simplicity of illustration, we use $S = 3$ stages. At stage $s + 1$, $\mathbf{Y}_s$ denotes the corresponding trend component extracted from the segment in the forecast window, $\mathbf{Y}_s^k$ is the diffusion sample at diffusion step $k$, and $\hat{\mathbf{Y}}_s^0$ is the denoised output.

# 3. mr-Diff: Multi-Re

In each stage $s+1$ ...

a conditional diffusion model is learned to **reconstruct the “trend $\mathbf{Y}_s$ extracted from the forecast window”**

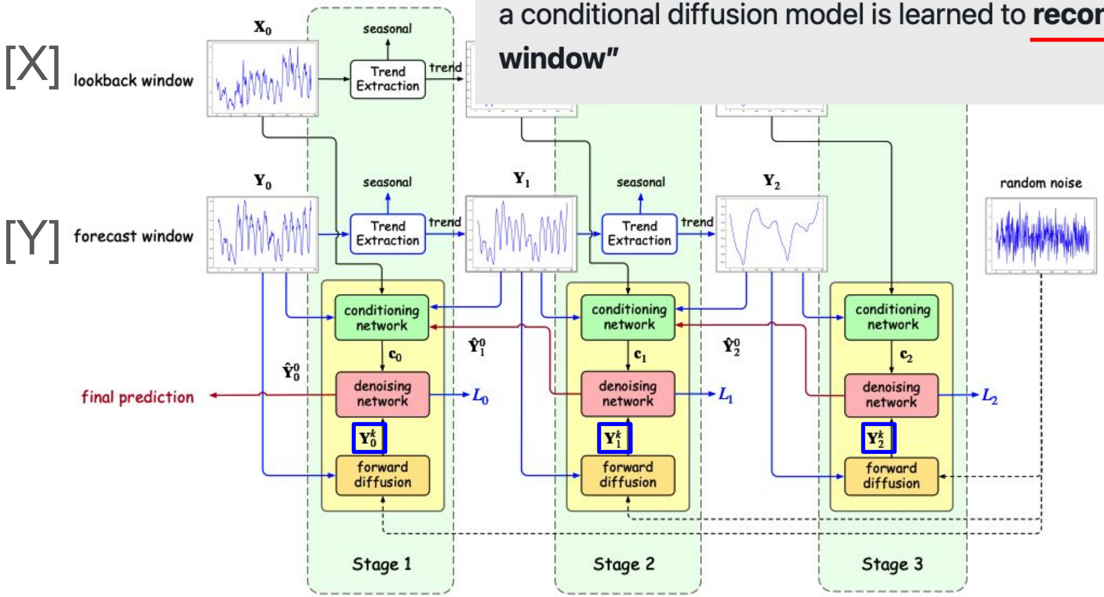



Figure 2: The proposed multi-resolution diffusion model mr-Diff. For simplicity of illustration, we use $S = 3$ stages. At stage $s+1$, $\mathbf{Y}_s$ denotes the corresponding trend component extracted from the segment in the forecast window, $\mathbf{Y}_s^k$ is the diffusion sample at diffusion step $k$, and $\boxed{\hat{\mathbf{Y}}_s^0}$ is the denoised output.

# 3. mr-Diff

Stage 1

In each stage $s + 1$ ...

a conditional diffusion model is learned to **reconstruct the "trend $\mathbf{Y}_s$ extracted from the forecast window"**

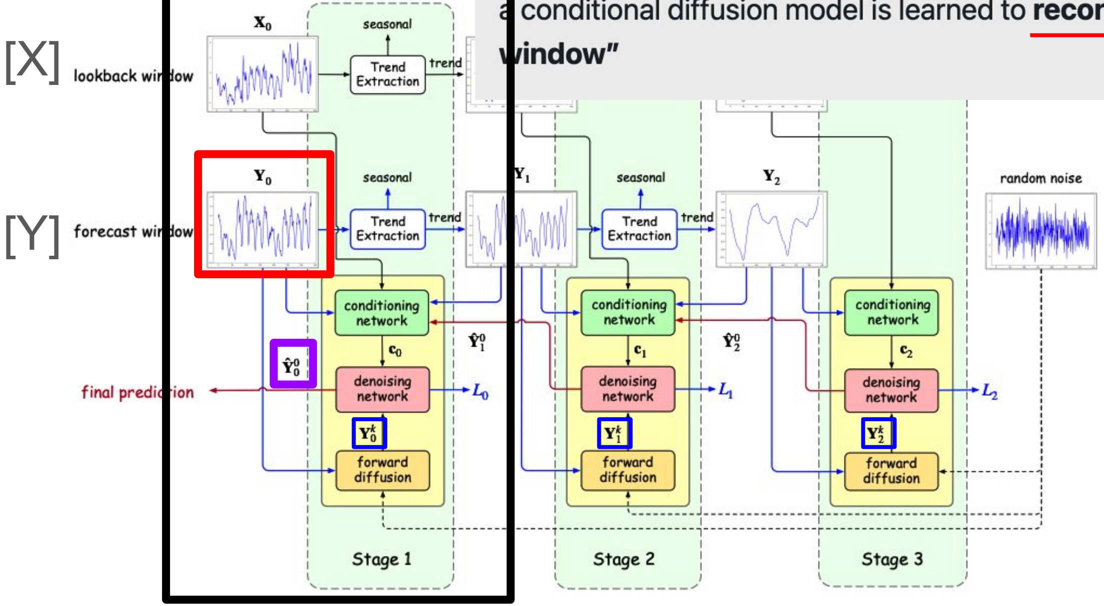



Figure 2: The proposed multi-resolution diffusion model mr-Diff. For simplicity of illustration, we use $S = 3$ stages. At stage $s + 1$, $\mathbf{Y}_s$ denotes the corresponding trend component extracted from the segment in the forecast window, $\mathbf{Y}_s^k$ is the diffusion sample at diffusion step $k$, and $\hat{\mathbf{Y}}_s^0$ is the denoised output.

**Ground Truth**

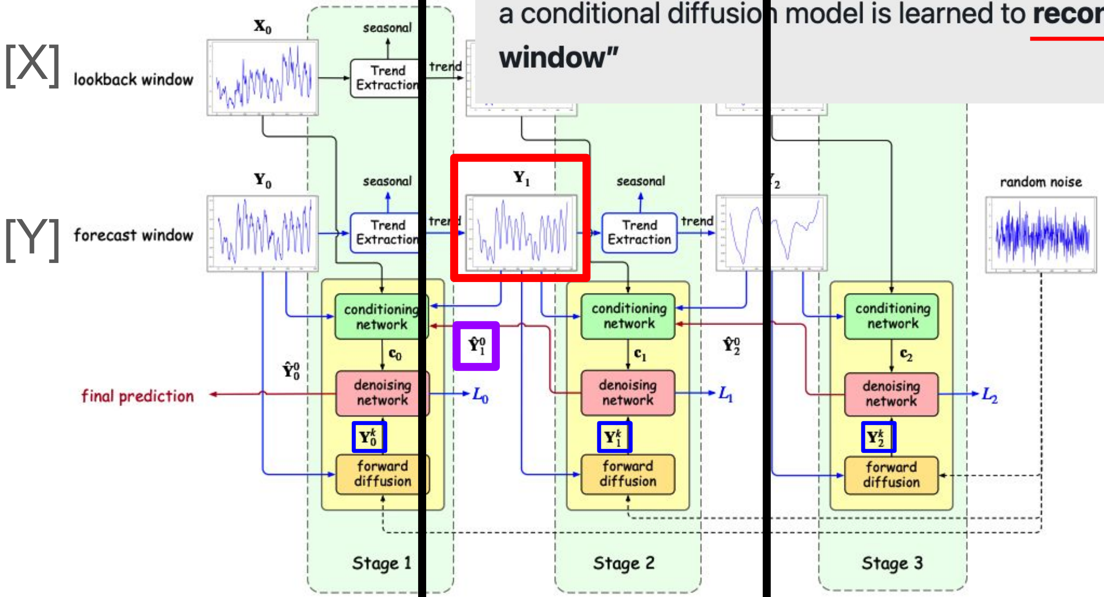
**Prediction**

# 3. mr-Diff: Multi-R

**Stage 2**

In each stage $s + 1$ :

a conditional diffusion model is learned to **reconstruct the "trend $\mathbf{Y}_s$ extracted from the forecast window"**




**Ground Truth**

**Prediction**

Figure 2: The proposed multi-resolution diffusion model `mr-Diff`. For simplicity of illustration, we use $S = 3$ stages. At stage $s + 1$, $\mathbf{Y}_s$ denotes the corresponding trend component extracted from the segment in the forecast window, $\mathbf{Y}_s^k$ is the diffusion sample at diffusion step $k$, and $\hat{\mathbf{Y}}_s^0$ is the denoised output.

# 3. mr-Diff: Multi-Re

In each stage $s+1$ ...

a conditional diffusion model is learned to **reconstruct the "trend $\mathbf{Y}_s$ extracted from the forecast window"**

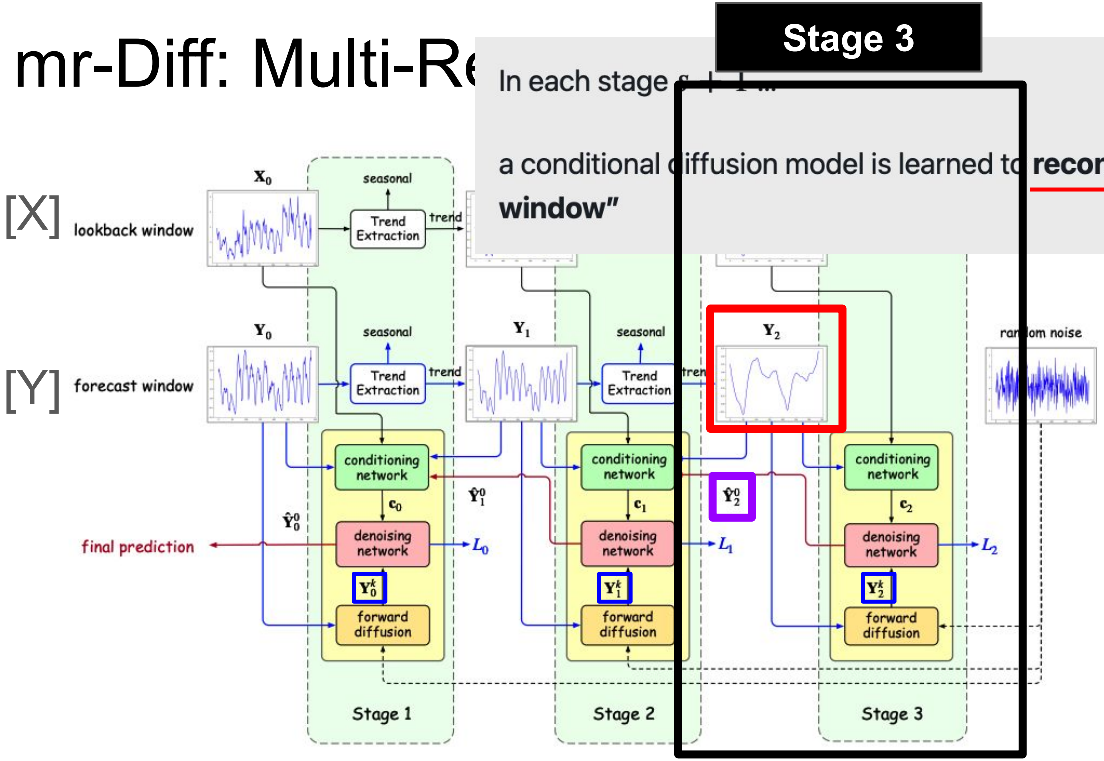



Figure 2: The proposed multi-resolution diffusion model mr-Diff. For simplicity of illustration, we use $S = 3$ stages. At stage $s+1$, $\mathbf{Y}_s$ denotes the corresponding trend component extracted from the segment in the forecast window, $\mathbf{Y}_s^k$ is the diffusion sample at diffusion step $k$, and $\hat{\mathbf{Y}}_s^0$ is the denoised output.

**Ground Truth**

**Prediction**

# 3. mr-Diff: Multi-Re

**Stage 3**

In each stage $s+1$ ...

extracted from the forecast

***How to predict the trend of the forecast window?***

[X]

[Y]

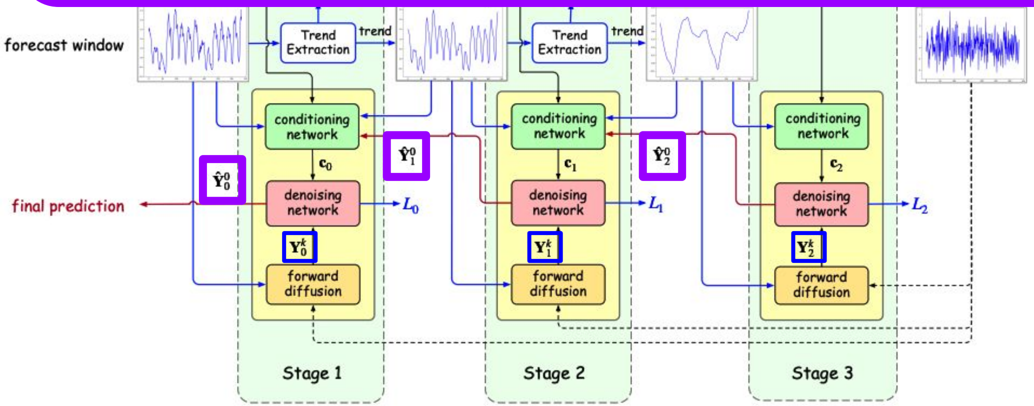
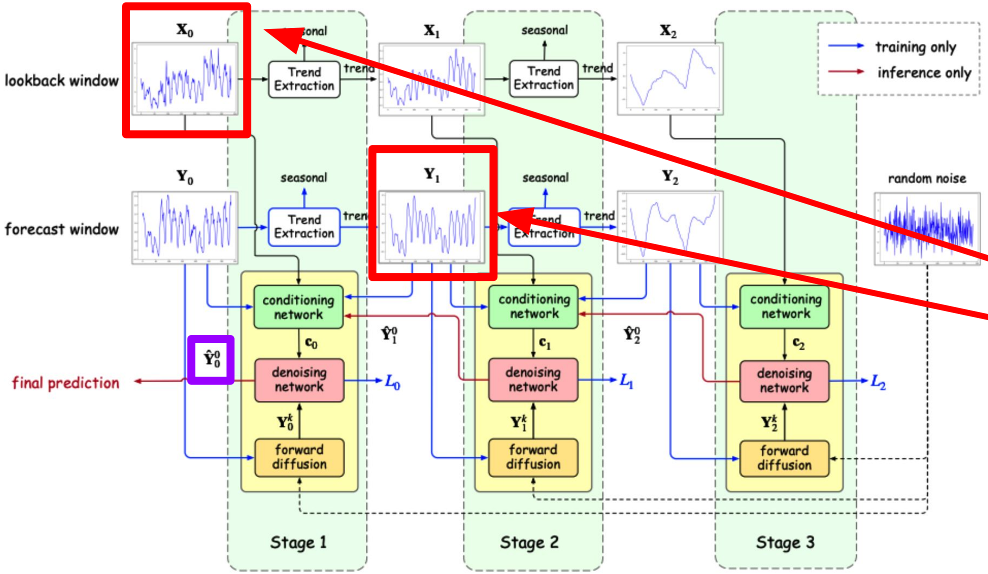



Figure 2: The proposed multi-resolution diffusion model mr-Diff. For simplicity of illustration, we use $S = 3$ stages. At stage $s+1$, $\mathbf{Y}_s$ denotes the corresponding trend component extracted from the segment in the forecast window, $\mathbf{Y}_s^k$ is the diffusion sample at diffusion step $k$, and $\boxed{\hat{\mathbf{Y}}_s^0}$ is the denoised output.

**Ground Truth**

**Prediction**

# 3. mr-Diff: Multi-Resolution Diffusion Model



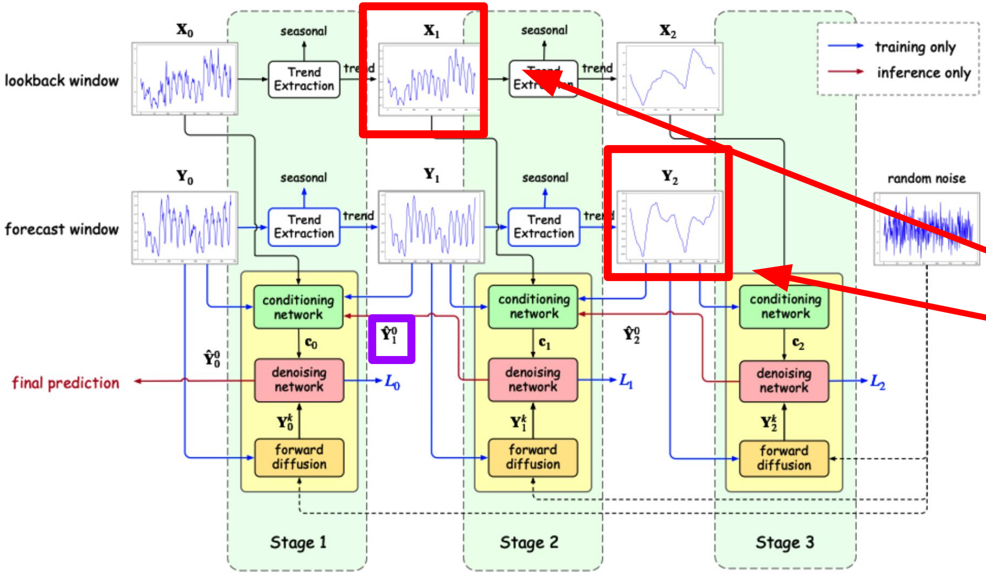

**[ Training & Inference ]**

**Training**

- Guide the **reconstruction of $\mathbf{Y}_s$**
- Condition:
  - Lookback segment $\mathbf{X}_s$
  - Coarser trend $\mathbf{Y}_{s+1}$

# [ Training ]

# 3. mr-Diff: Multi-Resolution Diffusion Model



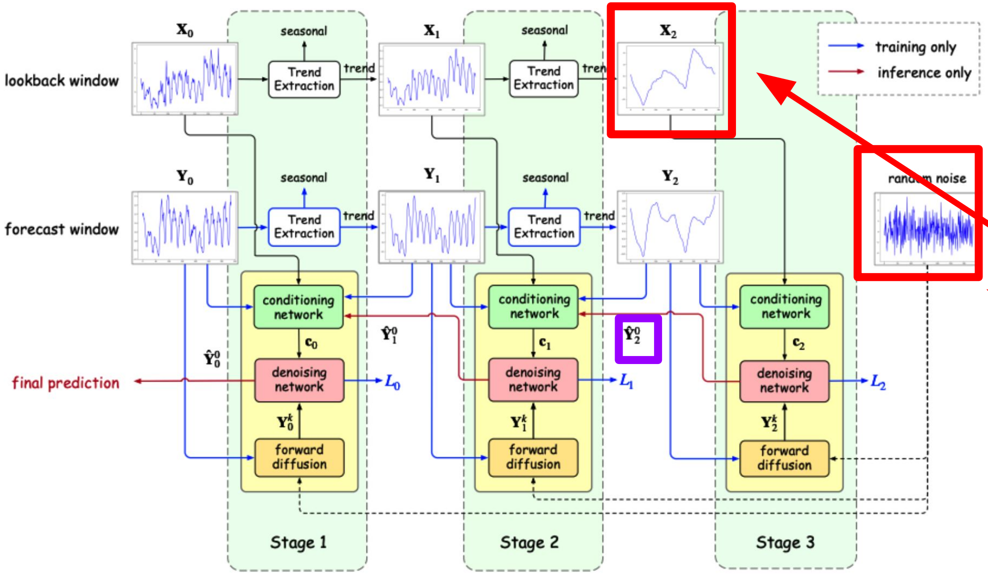

**[ Training & Inference ]**

**Training**

- Guide the **reconstruction of $\mathbf{Y}_s$**
- Condition:
  - Lookback segment $\mathbf{X}_s$
  - Coarser trend $\mathbf{Y}_{s+1}$

# [ Training ]

# 3. mr-Diff: Multi-Resolution Diffusion Model




**[ Training & Inference ]**

**Training**

- Guide the **reconstruction of $\mathbf{Y}_s$**
- Condition:
  - Lookback segment $\mathbf{X}_s$
  - Coarser trend $\mathbf{Y}_{s+1}$

**[ Training ]**

# 3. r

lookback window

forecast window

final prediction

**Y의 (s) 단계**의 trend를 예측하기 위해

- (1) **X의 (s) 단계**의 trend
- (2) **Y의 (s+1) 단계**의 trend

시점의 차이: X < Y

Granularity의 차이: (s) < (s+1)

**[ Training ]**

# 3. r

lookback window

forecast window

final prediction

**Y의 (s) 단계**의 trend를 예측하기 위해

- (1) **X의 (s) 단계**의 trend
- (2) **Y의 (s+1) 단계**의 trend

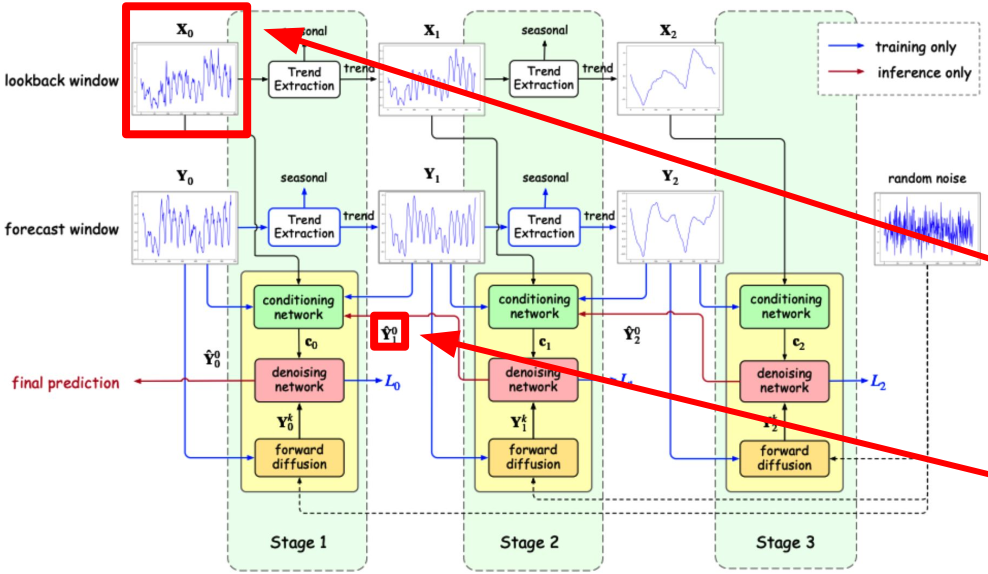
Q) Inference 시에는, ground truth를 모르지 않나?

시점의 차이: X < Y

Granularity의 차이: (s) < (s+1)

**[ Training ]**

# 3. mr-Diff: Multi-Resolution Diffusion Model




**[ Training & Inference ]**

**Training**

- Guide the **reconstruction of $\mathbf{Y}_s$**
- Condition:
  - Lookback segment $\mathbf{X}_s$
  - Coarser trend $\mathbf{Y}_{s+1}$

**Inference**

- Ground-truth $\mathbf{Y}_{s+1}$ is not available
- Replaced by its estimate $\hat{\mathbf{Y}}_{s+1}^0$ produced by the denoising process at stage $s+1$.

Use the "estimated" value!

# [ Inference ]

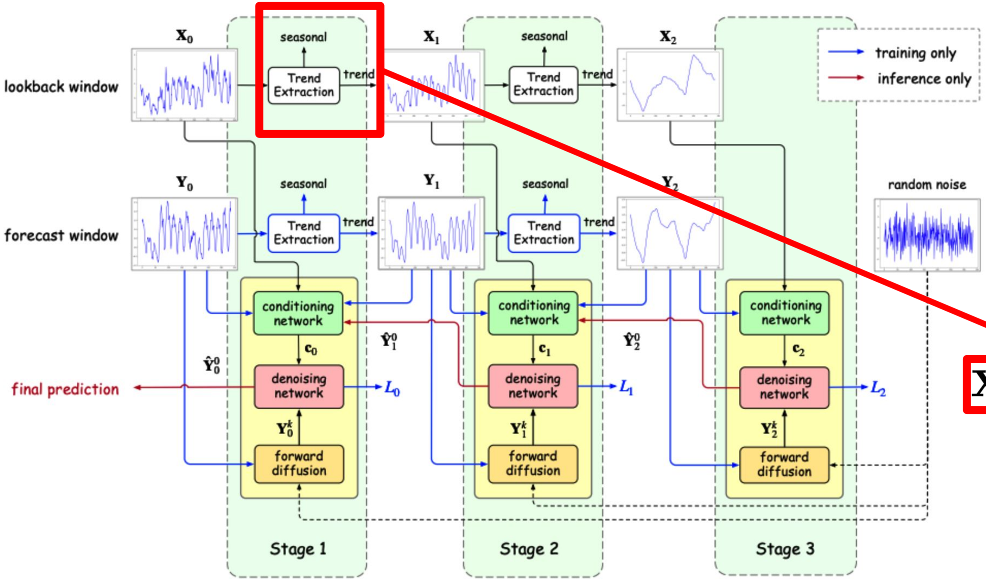
# 3. mr-Diff: Multi-Resolution Diffusion Model

1. **Extending Fine-to-Coarse Trends**
   - Seasonal & Trend Decomposition
2. **Temporal Multi-resolution Reconstruction**
   - Sinusoidal Temporal Embedding
   - Forward & Backward Process
   - Conditioning Network

# 3. mr-Diff: Multi-Resolution Diffusion Model

## 1. Extending Fine-to-Coarse Trends

- Seasonal & Trend Decomposition




- (Intuition) Easier to predict a finer trend **from a coarser trend**
- Finer seasonal component from a coarser seasonal component may be difficult

**Focus only on the **TREND**

$$\boxed{\mathbf{X}_s = \mathrm{AvgPool}(\mathrm{Padding}\left(\mathbf{X}_{s-1}\right), \tau_s), s = 1, \ldots, S-1}$$

# 3. mr-Diff: Multi-Resolution Diffusion Model

## 2. Temporal Multi-resolution Reconstruction

- **<u>Sinusoidal Temporal Embedding</u>**
- Forward & Backward Process
- Conditioning Network

$$k_{\text{embedding}} = \left[\sin\left(10^{\frac{0\times 4}{w-1}} t\right), \ldots, \sin\left(10^{\frac{w\times 4}{w-1}} t\right), \cos\left(10^{\frac{0\times 4}{w-1}} t\right), \ldots, \cos\left(10^{\frac{w\times 4}{w-1}} t\right)\right]$$

$$\mathbf{p}^k = \mathrm{SiLU}(\mathrm{FC}(\mathrm{SiLU}(\mathrm{FC}(k_{\text{embedding}}))))$$

# 3. mr-Diff: Multi-Resolution Diffusion Model

## 2. Temporal Multi-resolution Reconstruction

- Sinusoidal Temporal Embedding
- **<u>Forward</u>** & Backward Process
- Conditioning Network

Diffusion step

$$\mathbf{Y}_{\boxed{s}}^{\boxed{k}} = \sqrt{\bar{\alpha}_{\boxed{k}}} \mathbf{Y}_s^{\boxed{0}} + \sqrt{1 - \bar{\alpha}_{\boxed{k}}} \epsilon$$

Resolution

# 3. mr-Diff: Multi-Resolution Diffusion Model

## 2. Temporal Multi-resolution Reconstruction

- Sinusoidal Temporal Embedding
- Forward & **Backward** Process
- **Conditioning Network**

Decompose the **denoising objective into S sub-objectives**

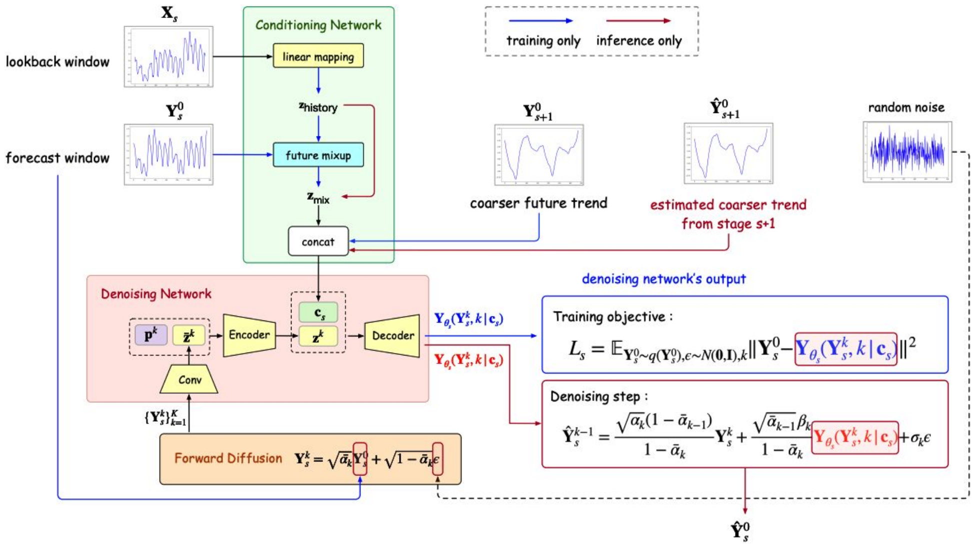
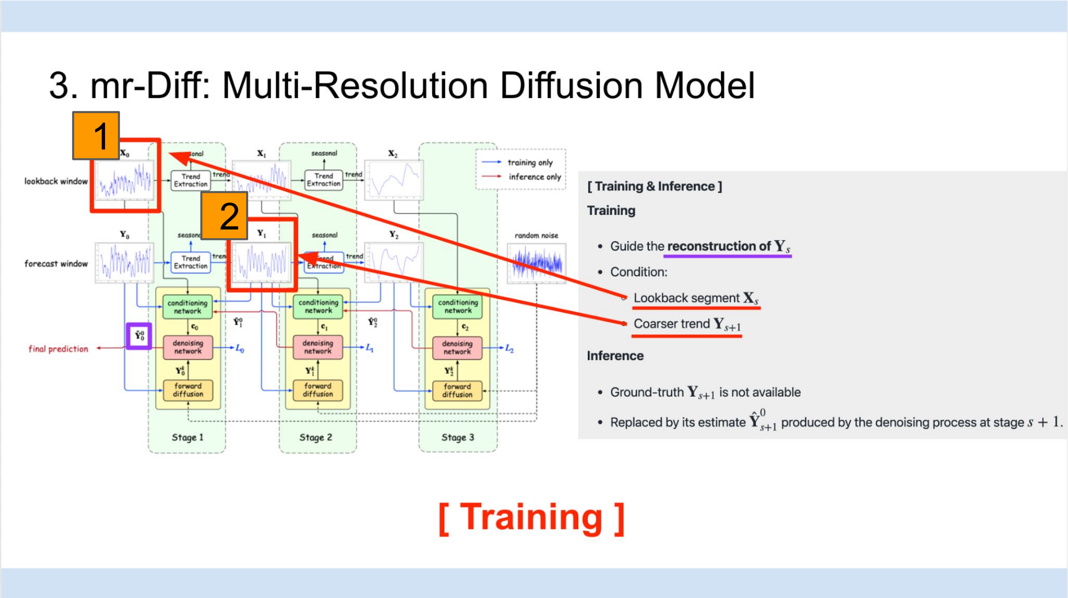


Figure 3: The conditioning network and denoising network.

# 3. mr-Diff: Multi-Resolution Diffusion Model

## 2. Temporal Multi-resolution Reconstruction




How to incorporate

1 and 2 as condition?

# 3. mr-Diff: Multi-Resolution Diffusion Model

## 2. Temporal Multi-resolution Reconstruction

Existing works

- Use the **original TS lookback** segment $\mathbf{X}_0$ as condition $\mathbf{c}$

vs

mr-Diff

- Use the lookback segment $\mathbf{X}_s$ at the same decomposition stage $s$.

→ Allows better and easier reconstruction

($\because$ $\mathbf{X}_s$ has the same resolution as $\mathbf{Y}_s$ to be reconstructed)

↔ When $\mathbf{X}_0$ is used as in existing TS diffusion models, the denoising network may overfit temporal details at the finer level.

1

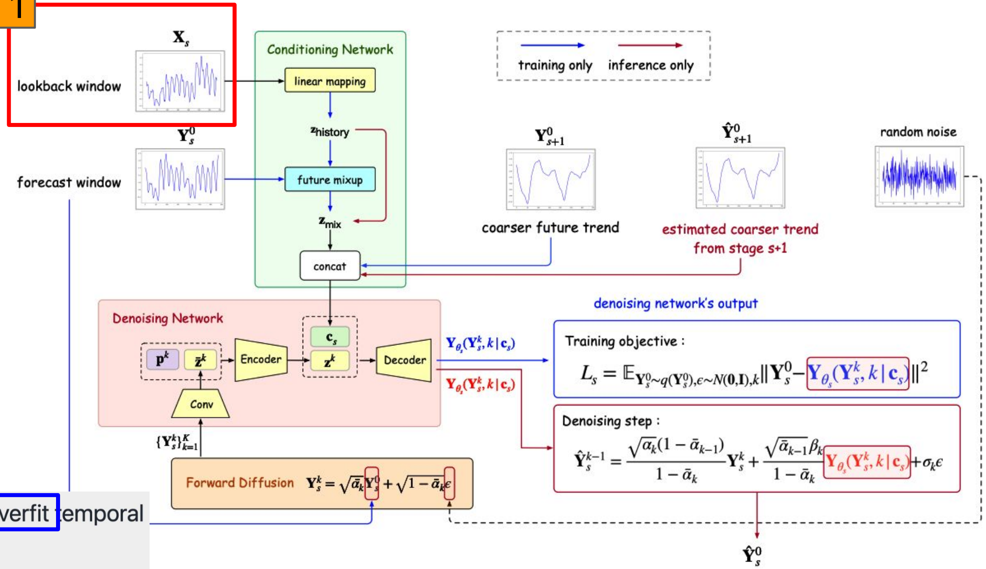



Figure 3: The conditioning network and denoising network.

# 3. mr-Diff: Multi-Resolution Diffusi

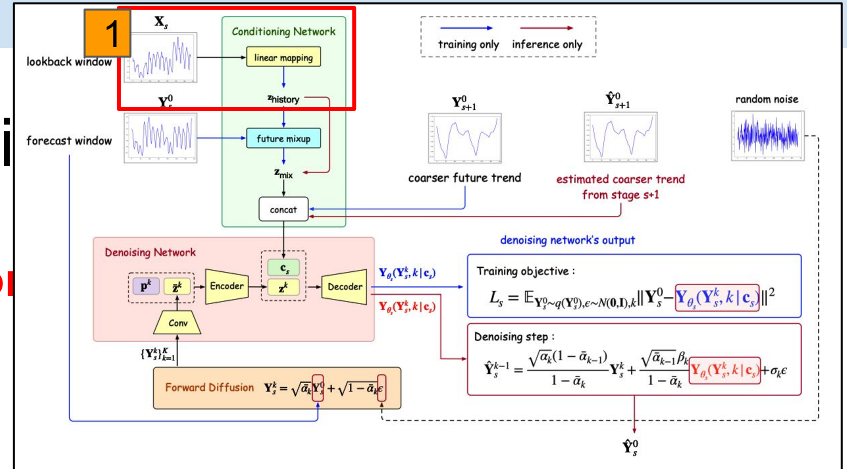



Figure 3: The conditioning network and denoising network.

## 2. Temporal Multi-resolution Reconstructio

Procedures

- Step 1) **Linear mapping** is applied on $\mathbf{X}_s$ to produce a $\mathbf{z}_{\text{history}} \in \mathbb{R}^{d \times H}$.
- Step 2) **Future-mixup**: to enhance $\mathbf{z}_{\text{history}}$ .
  - $\mathbf{z}_{\text{mix}} = \mathbf{m} \odot \mathbf{z}_{\text{history}} + (1 - \mathbf{m}) \odot \mathbf{Y}_s^0$.
  - Similar to teacher forcing, which mixes the ground truth with previous prediction output
- Step 3) **Coarser trend** $\mathbf{Y}_{s+1}$ $(= \mathbf{Y}_{s+1}^0)$ can also be useful for conditioning

$\rightarrow$ $\mathbf{z}_{\text{mix}}$ is concatenated with $\mathbf{Y}_{s+1}^0$ to produce the condition $\mathbf{c}_s$ (a $2d \times H$ tensor).

# 3. mr-Diff: Multi-Resolution Diffusi

## 2. Temporal Multi-resolution Reconstructio

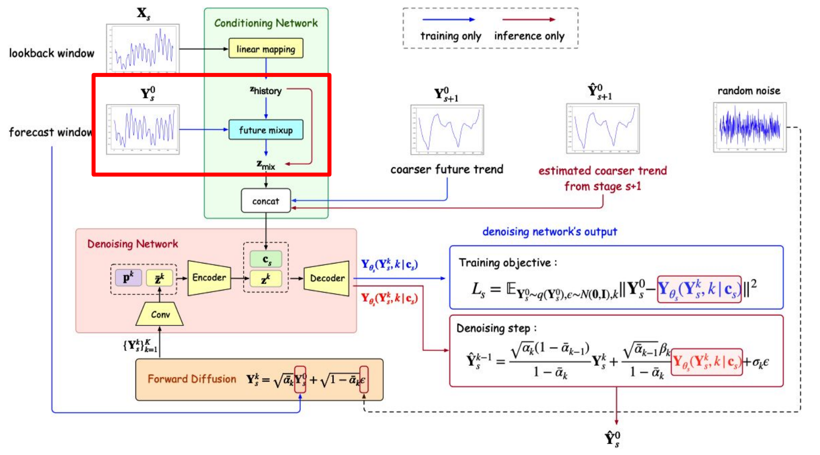



Figure 3: The conditioning network and denoising network.

Procedures

- Step 1) **Linear mapping** is applied on $\mathbf{X}_s$ to produce a $\mathbf{z}_{\text{history}} \in \mathbb{R}^{d \times H}$.
- Step 2) **Future-mixup**: to enhance $\mathbf{z}_{\text{history}}$ . (feat. TimeDiff, ICML 2023)
  - $\mathbf{z}_{\text{mix}} = \mathbf{m} \odot \mathbf{z}_{\text{history}} + (1 - \mathbf{m}) \odot \mathbf{Y}_s^0$.
  - Similar to teacher forcing, which mixes the ground truth with previous prediction output
- Step 3) **Coarser trend** $\mathbf{Y}_{s+1}$ $(= \mathbf{Y}_{s+1}^0)$ can also be useful for conditioning

$\rightarrow$ $\mathbf{z}_{\text{mix}}$ is concatenated with $\mathbf{Y}_{s+1}^0$ to produce the condition $\mathbf{c}_s$ (a $2d \times H$ tensor).

# 3. mr-Diff: Multi-Resolution Diffusi

## 2. Temporal Multi-resolution Reconstructio

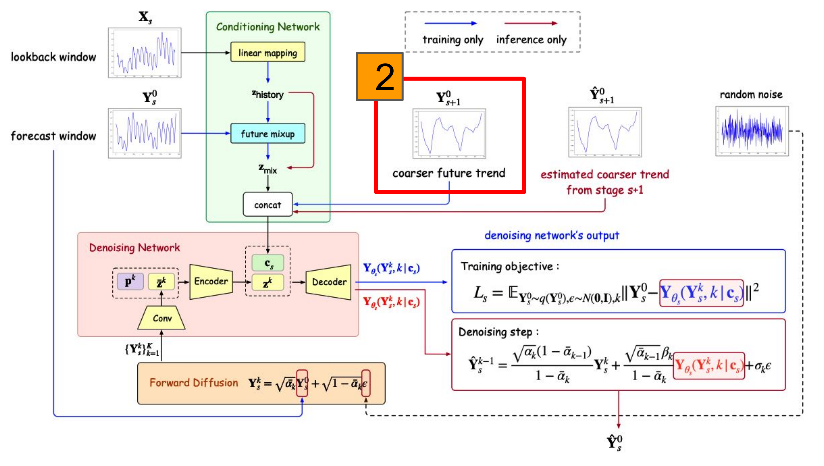


Figure 3: The conditioning network and denoising network.

Procedures

- Step 1) **Linear mapping** is applied on $\mathbf{X}_s$ to produce a $\mathbf{z}_{\text{history}} \in \mathbb{R}^{d \times H}$.
- Step 2) **Future-mixup**: to enhance $\mathbf{z}_{\text{history}}$ .
  - $\mathbf{z}_{\text{mix}} = \mathbf{m} \odot \mathbf{z}_{\text{history}} + (1 - \mathbf{m}) \odot \mathbf{Y}_s^0$.
  - Similar to teacher forcing, which mixes the ground truth with previous prediction output
- Step 3) **Coarser trend** $\mathbf{Y}_{s+1}$ $(= \mathbf{Y}_{s+1}^0)$ can also be useful for conditioning

$\rightarrow$ $\mathbf{z}_{\text{mix}}$ is concatenated with $\mathbf{Y}_{s+1}^0$ to produce the condition $\mathbf{c}_s$ (a $2d \times H$ tensor).

# 3. mr-Diff: Multi-Resolution Diffusi

## 2. Temporal Multi-resolution Reconstructio

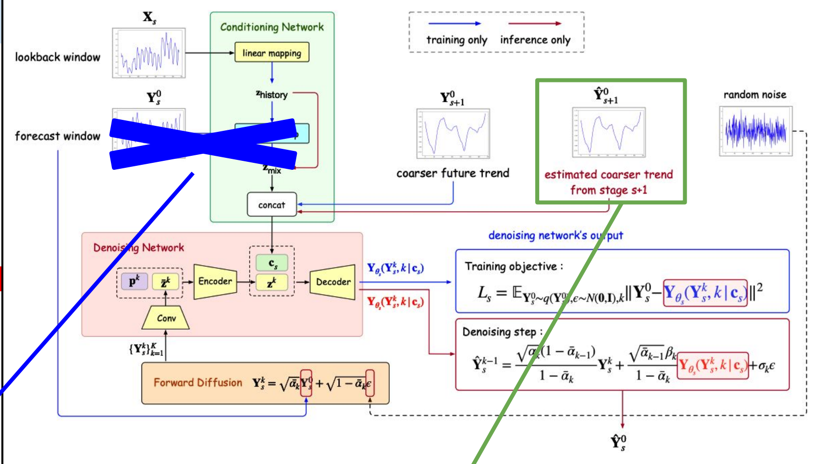



Figure 3: The conditioning network and denoising network.

### At test phase…

- Ground-truth $\mathbf{Y}_s^0$ is no longer available

  $\rightarrow$ No future-mixup … simply set $\mathbf{z}_{\text{mix}} = \mathbf{z}_{\text{history}}$ .

- Coarser trend $\mathbf{Y}_{s+1}$ is also not available

  $\rightarrow$ Concatenate $\mathbf{z}_{\text{mix}}$ with the estimate $\hat{\mathbf{Y}}_{s+1}^0$ generated from stage $s+2$ .

# 3. mr-Diff: Multi-Resolution Diffusi

## 2. Temporal Multi-resolution Reconstruction

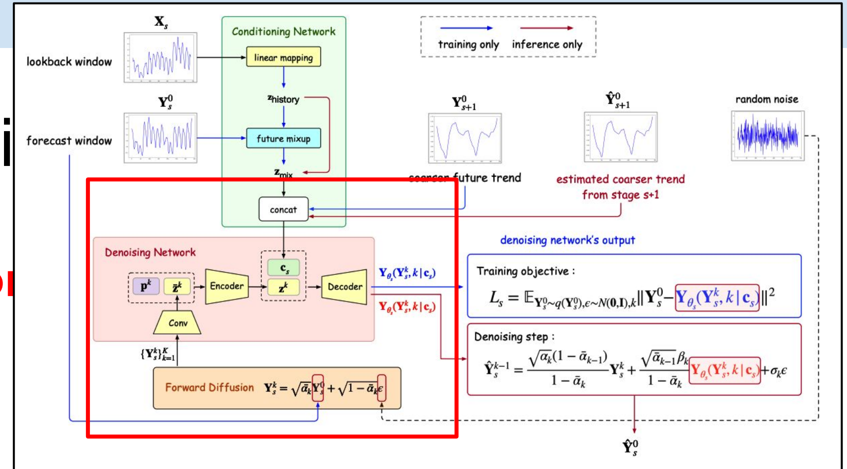


Figure 3: The conditioning network and denoising network.

### Denoising Network

Outputs a $\mathbf{Y}_{\theta_s}\left(\mathbf{Y}_s^k, k \mid \mathbf{c}_s\right)$ with guidance from the condition $\mathbf{c}_s$

Denoising process at step $k$ of stage $s+1$:

- $p_{\theta_s}\left(\mathbf{Y}_s^{k-1} \mid \mathbf{Y}_s^k, \mathbf{c}_s\right) = \mathcal{N}\left(\mathbf{Y}_s^{k-1}; \mu_{\theta_s}\left(\mathbf{Y}_s^k, k \mid \mathbf{c}_s, \sigma_k^2 \mathbf{I}\right)\right), k = K, \ldots, 1.$
- $\mathbf{Y}_{\theta_s}\left(\mathbf{Y}_s^k, k \mid \mathbf{c}_s\right)$ is an estimate of $\mathbf{Y}_s^0$.

# 3. mr-Diff: Multi-Resolution Diffusion Model

## Experiments

### (1) Dataset

Table 4: Summary of dataset statistics, including the dimension, total number of observations, sampling frequency, and prediction length $H$ used in the experiments.

| | 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) |

# 3. mr-Diff: Multi-Resolution Diffusion Model

## Experiments

### (2) TS forecasting

(2021,22) TS Diffusion models

(2023) TimeDiff

TimeDiff

Table 1: Univariate prediction MAEs on the real-world time series datasets (subscript is the rank). Results of all baselines (except NLinear, SCINet, and N-Hits) are from (Shen & Kwok, 2023).

| | *NorPool* | *Caiso* | *Traffic* | *Electricity* | *Weather* | *Exchange* | *[illegible]Th1* | *ETTm1* | *Wind* | avg rank |
|---|---|---|---|---|---|---|---|---|---|---|
| **mr-Diff** | 0.609 $_{(2)}$ | 0.212 $_{(4)}$ | 0.197 $_{(2)}$ | 0.332 $_{(1)}$ | **0.032** $_{(1)}$ | **0.094** [illegible] | [illegible]**96** $_{(1)}$ | **0.149** $_{(1)}$ | 1.168 $_{(2)}$ | 1.7 |
| TimeDiff | 0.613 $_{(3)}$ | 0.209 $_{(3)}$ | 0.207 $_{(3)}$ | 0.341 $_{(3)}$ | 0.035 $_{(4)}$ | 0.1[illegible] | 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 $_{(6)}$ | [illegible]1) | 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 [illegible] | [illegible] $_{(23)}$ | 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 [illegible] | [illegible]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[illegible] | 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 $_{(21)}$ | [illegible] $_{(4)}$ | 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 [illegible] | [illegible]9 $_{(22)}$ | 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[illegible] | 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[illegible] | 0.069 $_{(14)}$ | 0.104 $_{(10)}$ | 0.210 $_{(6)}$ | **0.149** $_{(1)}$ | 1.189 $_{(3)}$ | 8.6 |
| [illegible] | [illegible] | [illegible] | .241 $_{(10)}$ | [illegible] $_{(12)}$ | 0.102 $_{(19)}$ | 0.106 $_{(12)}$ | 0.202 $_{(2)}$ | 0.165 $_{(10)}$ | 1.472 $_{(23)}$ | 10.3 |
| [illegible] | [illegible] | [illegible] | .225 $_{(5)}$ | 0.439 $_{(13)}$ | 0.130 $_{(21)}$ | 0.096 $_{(3)}$ | 0.242 $_{(18)}$ | 0.165 $_{(10)}$ | 1.236 $_{(9)}$ | 10.4 |
| [illegible] | [illegible] | [illegible] | .375 $_{(18)}$ | 0.430 $_{(10)}$ | 0.083 $_{(17)}$ | 0.148 $_{(19)}$ | 0.302 $_{(22)}$ | 0.210 $_{(22)}$ | 1.348 $_{(21)}$ | 17.4 |
| [illegible] | [illegible] | [illegible] | .269 $_{(11)}$ | 0.478 $_{(17)}$ | 0.098 $_{(18)}$ | 0.111 $_{(15)}$ | 0.260 $_{(21)}$ | 0.174 $_{(16)}$ | 1.338 $_{(20)}$ | 14.4 |
| [illegible] | [illegible] | [illegible] | .278 $_{(12)}$ | 0.453 $_{(14)}$ | 0.057 $_{(13)}$ | 0.168 $_{(22)}$ | 0.212 $_{(8)}$ | 0.195 $_{(20)}$ | 1.271 $_{(14)}$ | 14.3 |
| [illegible] | [illegible] | [illegible] | .495 $_{(21)}$ | 0.623 $_{(22)}$ | 0.040 $_{(10)}$ | 0.152 $_{(20)}$ | 0.220 $_{(11)}$ | 0.174 $_{(16)}$ | 1.319 $_{(18)}$ | 17.3 |
| [illegible] | [illegible] | [illegible] | .215 $_{(4)}$ | 0.455 $_{(15)}$ | 0.107 $_{(20)}$ | 0.104 $_{(10)}$ | 0.211 $_{(7)}$ | 0.179 $_{(19)}$ | 1.284 $_{(15)}$ | 13.1 |
| [illegible] | [illegible] | [illegible] | .308 $_{(13)}$ | 0.433 $_{(11)}$ | 0.069 $_{(14)}$ | 0.118 $_{(16)}$ | 0.212 $_{(8)}$ | 0.172 $_{(15)}$ | 1.236 $_{(9)}$ | 13.2 |
| [illegible] | [illegible] | [illegible] | .336 $_{(16)}$ | 0.469 $_{(16)}$ | 0.071 $_{(16)}$ | 0.103 $_{(9)}$ | 0.247 $_{(19)}$ | 0.196 $_{(21)}$ | 1.212 $_{(6)}$ | 15.4 |
| [illegible] | [illegible] | [illegible] | .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 |

**TimeDiff**

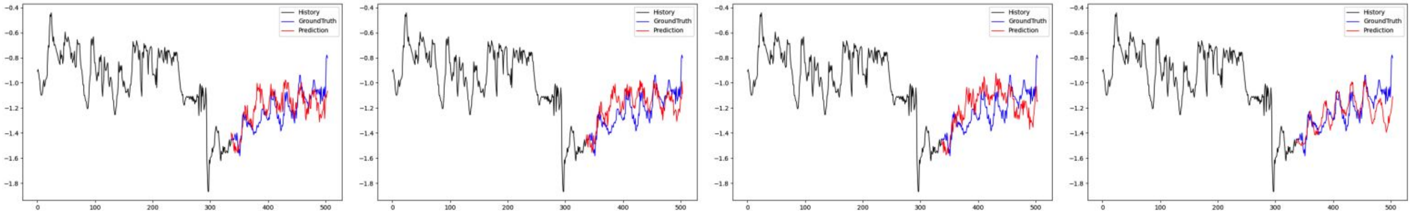
- No code
- Different result (use MSE, not MAE)

# 3. mr-Diff: Multi-Resolution Diffusion Model

## Experiments

### (3) Analysis

**Visualization** ( TimeDiff, mr-Diff: no code implementation )




(a) SCINet. (b) NLinear. (c) TimeDiff. (d) mr-Diff.

Figure 4: Visualizations on *ETTh1* by (a) SCINet (Liu et al., 2022), (b) NLinear (Zeng et al., 2023), (c) TimeDiff (Shen & Kwok, 2023) and (d) the proposed `mr-Diff`.

# 3. mr-Diff: Multi-Resolution Diffusion Model

## Experiments

### (3) Analysis

**Efficiency analysis**

Table 3: Inference time (in ms) of various time series diffusion models with different prediction horizons ($H$) on the univariate *ETTh1*.

| | $H=96$ | $H=168$ | $H=192$ | $H=336$ | $H=720$ |
|---|---|---|---|---|---|
| mr-Diff ($S$=2) | **8.3** | **9.5** | **9.8** | **11.9** | **21.6** |
| mr-Diff ($S$=3) | 12.5 | 14.3 | 14.9 | 16.8 | 27.5 |
| mr-Diff ($S$=4) | 16.7 | 19.1 | 19.7 | 28.5 | 36.4 |
| mr-Diff ($S$=5) | 30.0 | 30.2 | 30.2 | 35.0 | 43.6 |
| TimeDiff | 16.2 | 17.3 | 17.6 | 26.5 | 34.6 |
| TimeGrad | 870.2 | 1620.9 | 1854.5 | 3119.7 | 6724.1 |
| CSDI | 90.4 | 128.3 | 142.8 | 398.9 | 513.1 |
| SSSD | 418.6 | 590.2 | 645.4 | 1054.2 | 2516.9 |

# 3. mr-Diff: Multi-Resolution Diffusion Model

Summary

- TS를 multi-granularity로 나눠서 diffuse한다는 점에서 novel
- 마찬가지로, 빈약한 실험 및 분석
- Openreview Rating: [8,6,6,6]

# 4. Conclusion

- Time-Series Diffusion Model의 **연구 초창기**
- 다른 도메인의 모델을 Time Series에 접목할 때, **1차원적으로 쉽게 적용할 수 있는 접근**:
    - **Time Series Decomposition**
    - **Hierarchical Approach**
    - **Seasonality Analysis ( feat. Fourier Transform )**
    - …
- 이러한 연구들이 올해 서서히 나오고 있는 추세.
- 향후 연구 방향) 새로운 아이디어로 위의 **TS domain-specific** 특징을 포착하는 알고리즘 고안