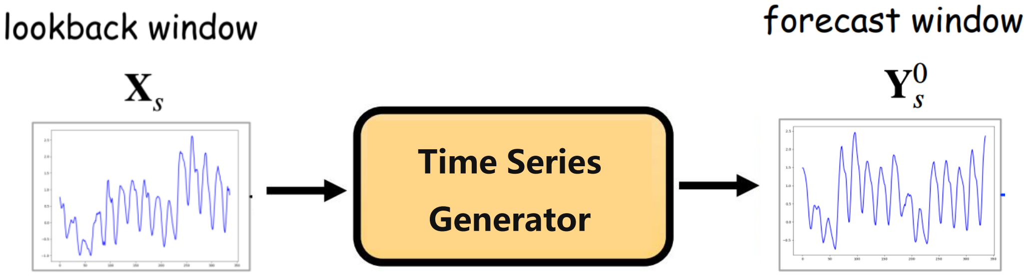
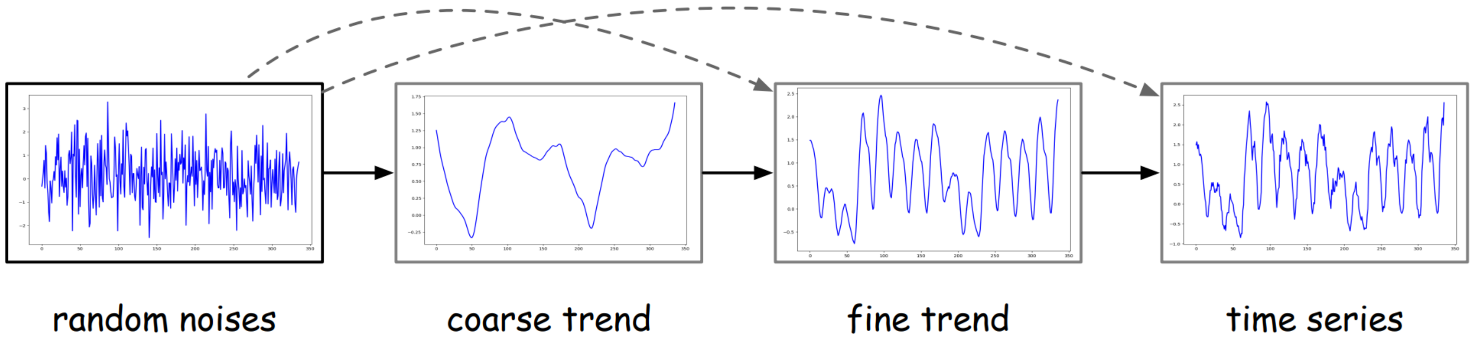
2024 ICLR CCF-A

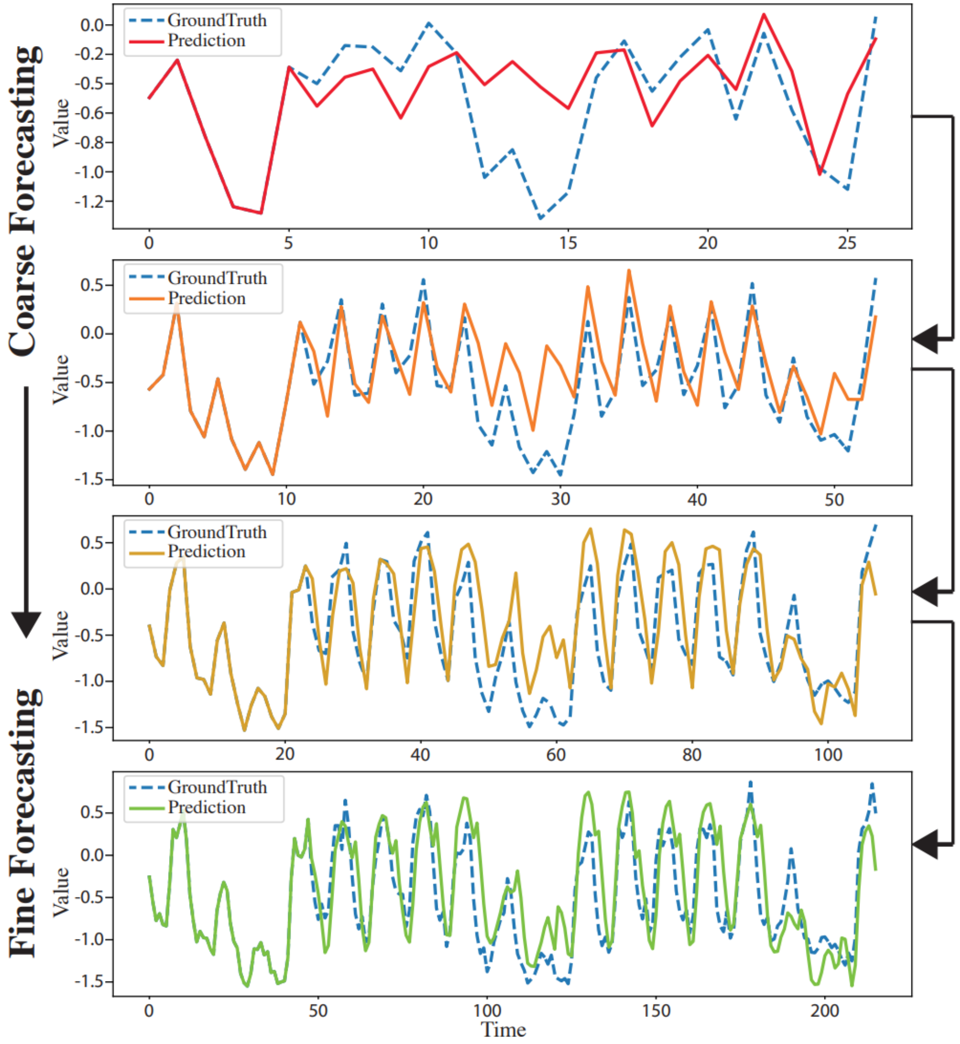
Multi-Resolution Diffusion Models for Timeseries Forecasting





01 创新点

- ➢ Multi-Resolution 多分辨率
  - 季节趋势分解
  - 趋势由粗→细（去噪由易→难）
- ➢ Diffusion Model 扩散模型
  - 正向扩散（加噪）
  - 反向去噪



02

背景知识：扩散模型

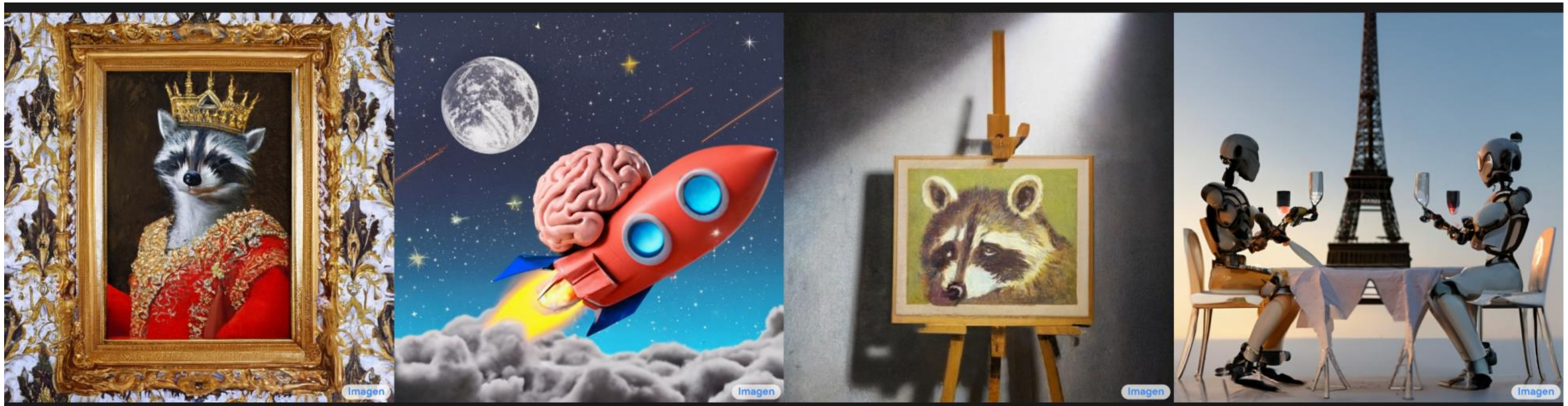
- Denoising Diffusion Probabilistic Models (DDPM)
  - 生成模型：GAN、VAE
  - Text-to-Image：文生图
  - Stable Diffusion（StabilityAI）、DALL-E series（OpenAI）、Imagen（Google）

[1] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in neural information processing systems, 2020, 33: 6840-6851.
[2] Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 10684-10695. Stable Diffusion：https://stability.ai/news/stable-diffusion-v2-release
[3] Ramesh A, Pavlov M, Goh G, et al. Zero-shot text-to-image generation[C]//International Conference on Machine Learning. PMLR, 2021: 8821-8831. DALL-E 2：https://openai.com/dall-e-2
[4] Saharia C, Chan W, Saxena S, et al. Photorealistic text-to-image diffusion models with deep language understanding[J]. Advances in Neural Information Processing Systems, 2022, 35: 36479-36494. Imagen：https://imagen.research.google/

02

背景知识：扩散模型

- Denoising Diffusion Probabilistic Models (DDPM)
  - 生成模型：GAN、VAE
  - Text-to-Image：文生图
  - Stable Diffusion（StabilityAI）、DALL-E series（OpenAI）、Imagen（Google）



A majestic oil painting of a raccoon Queen wearing red French royal gown. The painting is hanging on an ornate wall decorated with wallpaper.

A brain riding a rocketship heading towards the moon.

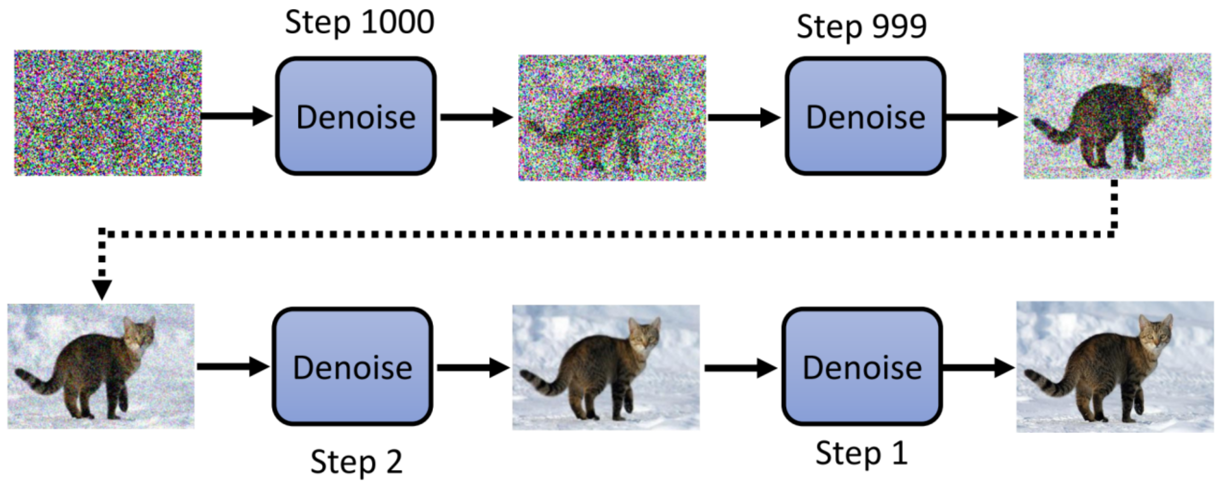
A single beam of light enter the room from the ceiling. The beam of light is illuminating an easel. On the easel there is a Rembrandt painting of a raccoon.

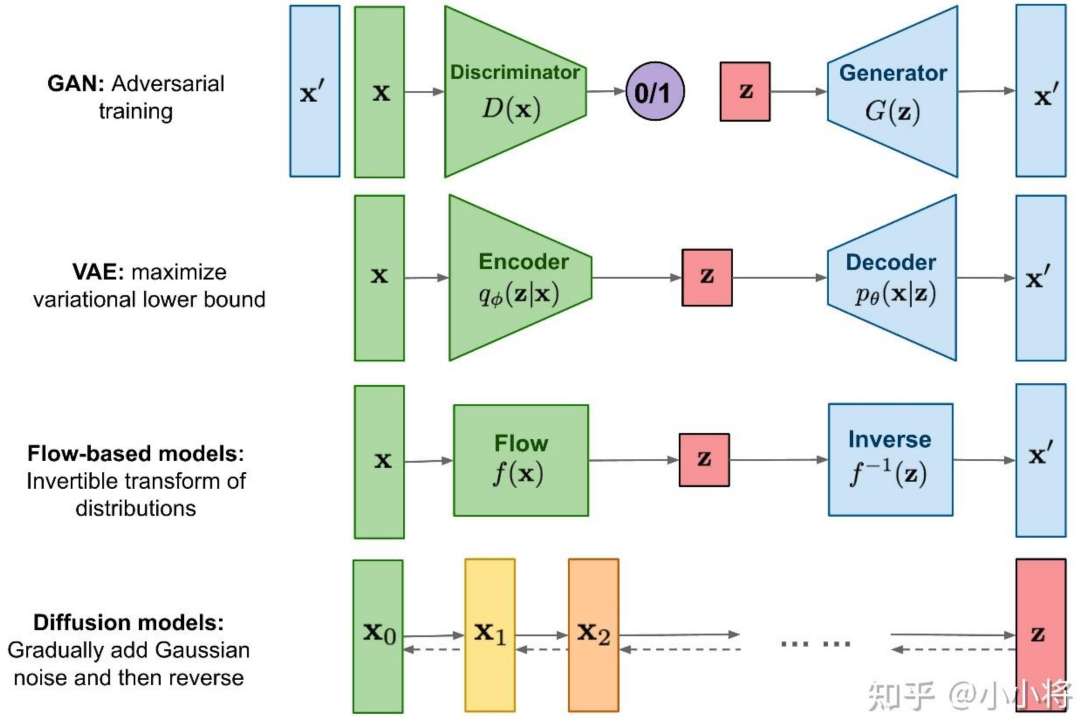
A robot couple fine dining with Eiffel Tower in the background.

02

背景知识：扩散模型

➢ Diffusion Model

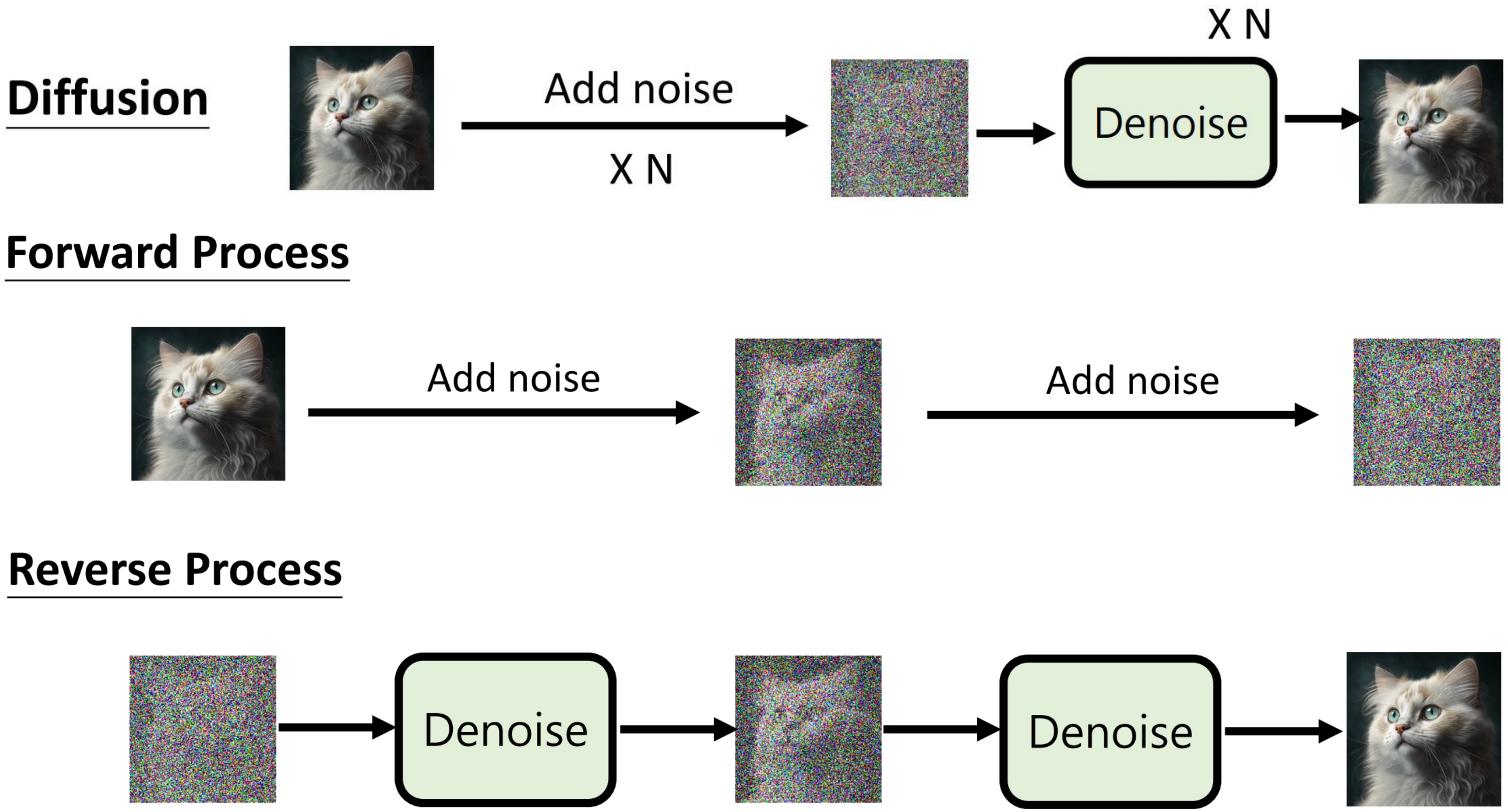




*李宏毅教授讲diffusion model：https://www.youtube.com/watch?v=ifCDXFdeaaM&t=210s

02 背景知识：扩散模型

➢ Diffusion Model
- 前向扩散（加噪）
- 反向去噪

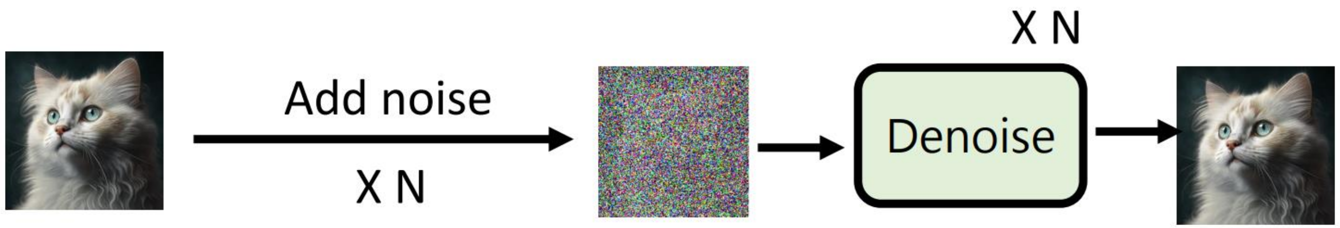


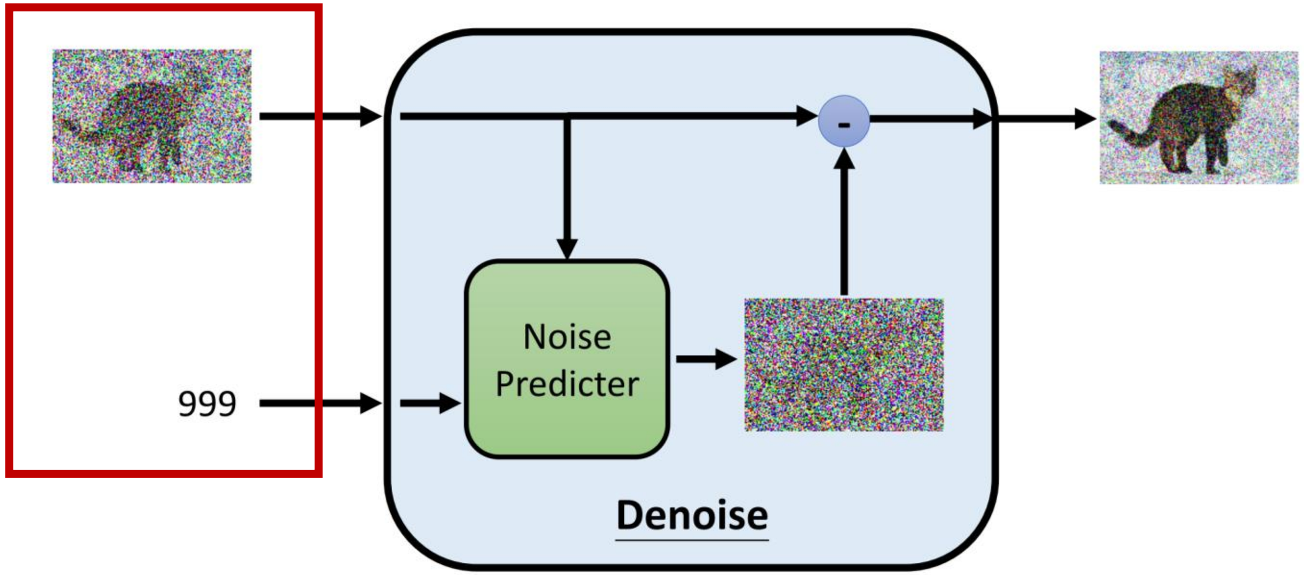
*李宏毅教授讲diffusion model：https://www.youtube.com/watch?v=ifCDXFdeaaM&t=210s

02 背景知识：扩散模型

➢ Diffusion Model

- 前向扩散（加噪） **Diffusion**
- 反向去噪

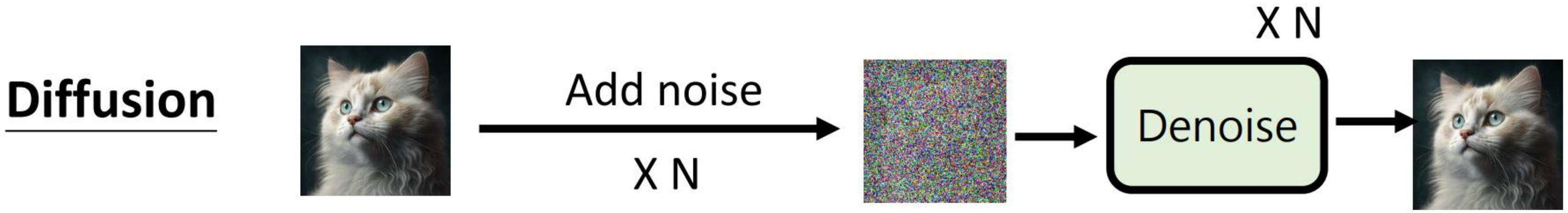




*李宏毅教授讲diffusion model：https://www.youtube.com/watch?v=ifCDXFdeaaM&t=210s

02 背景知识：扩散模型

- Diffusion Model
  - 前向扩散（加噪）
  - 反向去噪
  - 数学原理：最大似然估计、马尔科夫过程



Algorithm 1 Training

```
repeat
    x_0 ~ q(x_0)
    t ~ Uniform({1, ..., T})
    ε ~ N(0, I)
    Take gradient descent step on
        ∇_θ ‖ε − ε_θ(√ᾱ_t x_0 + √(1 − ᾱ_t) ε, t)‖²
until converged
```

repeat

$\mathbf{x}_0 \sim q(\mathbf{x}_0)$

$t \sim \mathrm{Uniform}(\{1, \ldots, T\})$

$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$

until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t = T, \ldots, 1$ **do**

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$$

end for

return $\mathbf{x}_0$

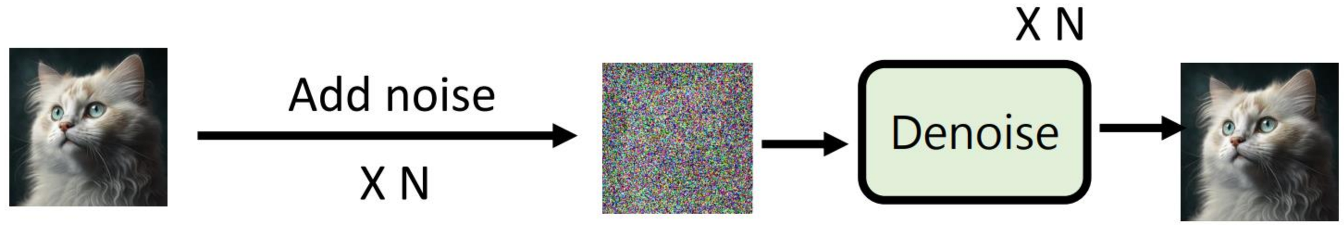
*李宏毅教授讲diffusion model：https://www.youtube.com/watch?v=ifCDXFdeaaM&t=210s

02 背景知识：扩散模型

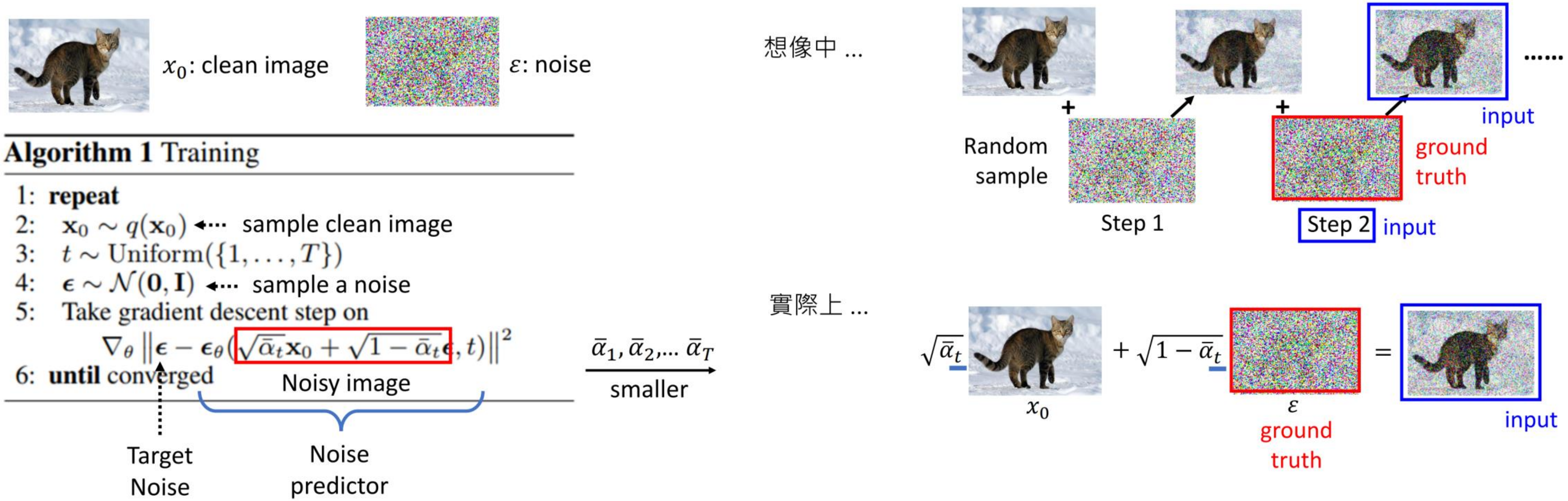
➢ Diffusion Model

- 前向扩散（加噪）
- 反向去噪

Diffusion



Training



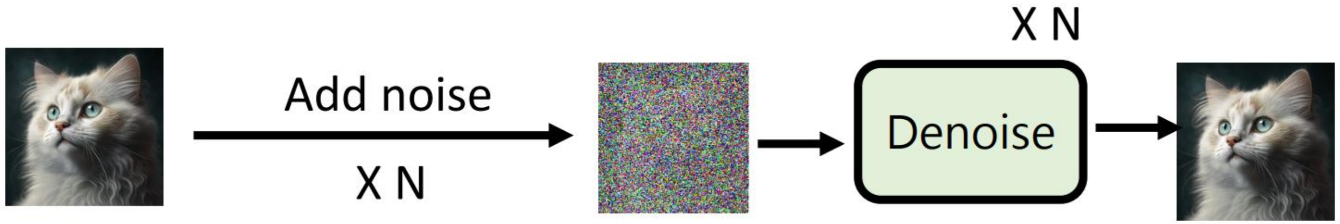
*李宏毅教授讲diffusion model：https://www.youtube.com/watch?v=ifCDXFdeaaM&t=210s

02 背景知识：扩散模型

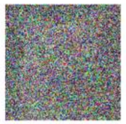
➢ Diffusion Model

- 前向扩散（加噪）
- 反向去噪

Diffusion



Inference



x_T

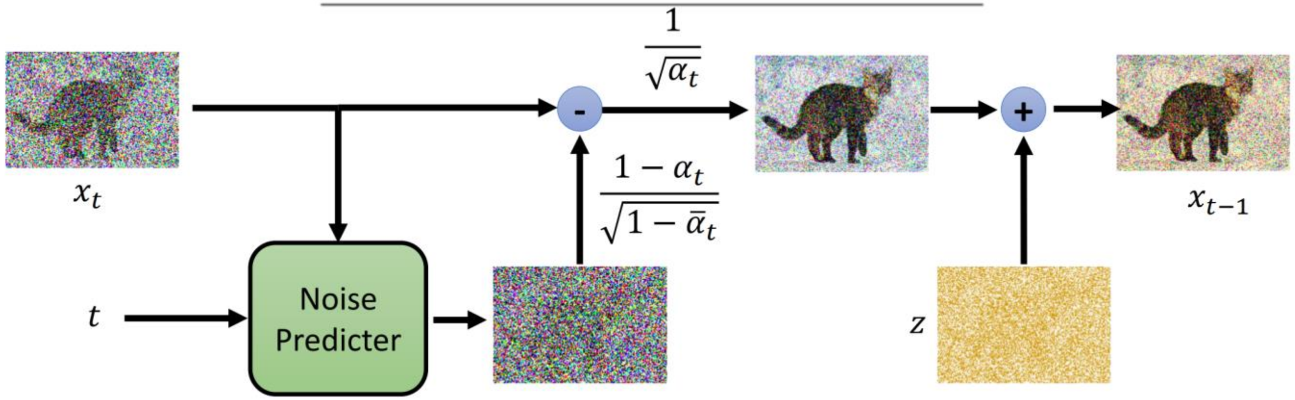
Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\boxed{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

sample a noise?!

$\bar{\alpha}_1, \bar{\alpha}_2, \ldots \bar{\alpha}_T$

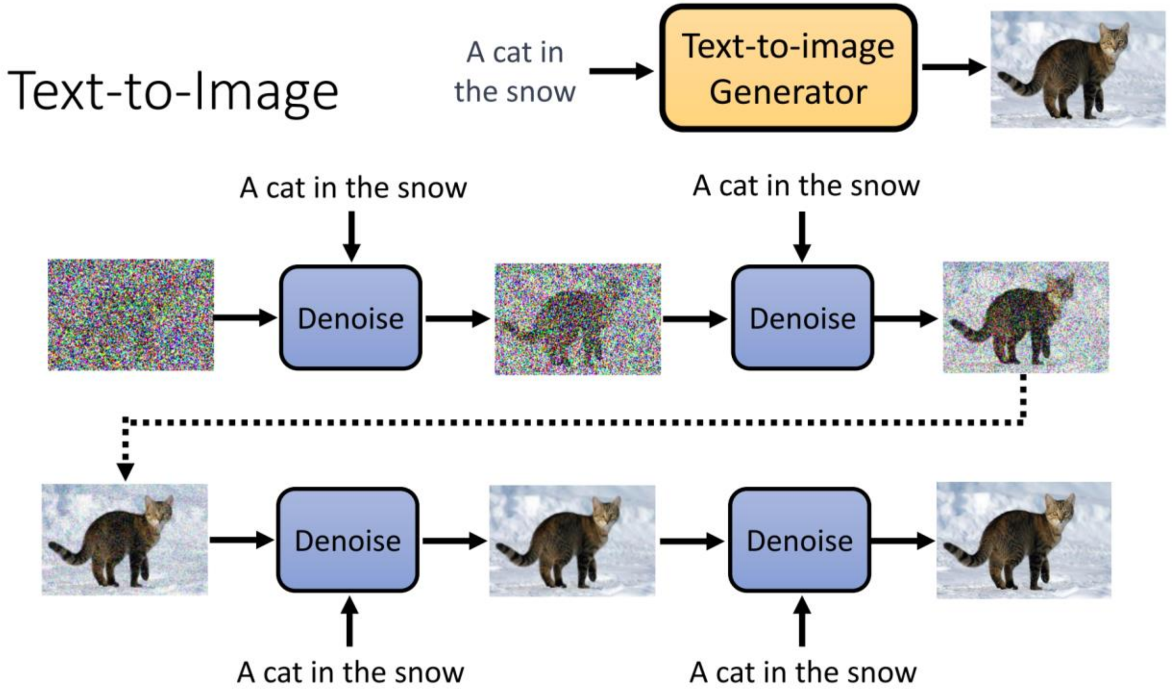
$\alpha_1, \alpha_2, \ldots \alpha_T$

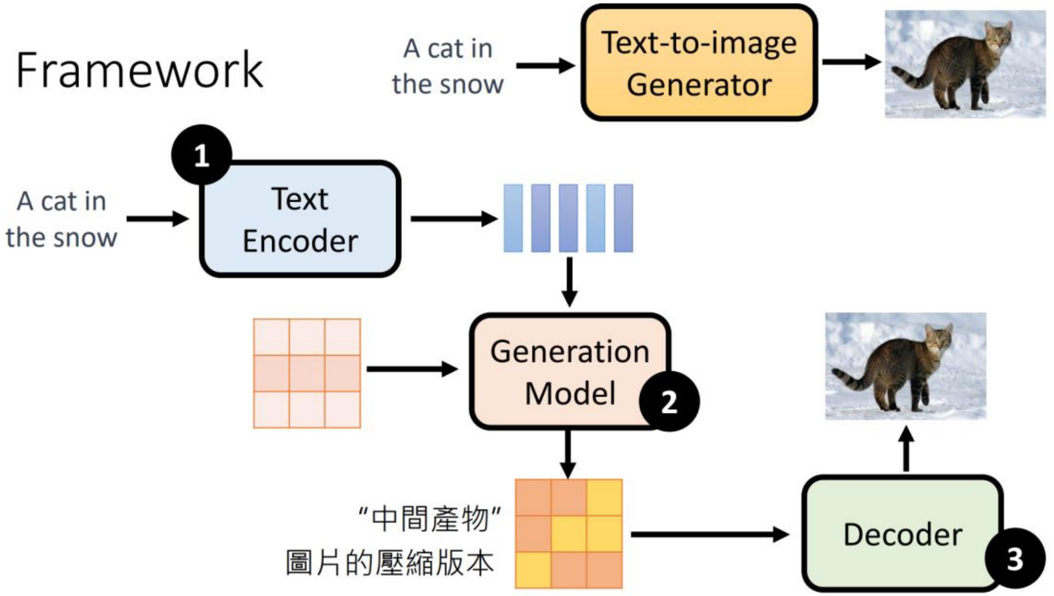


*李宏毅教授讲diffusion model：https://www.youtube.com/watch?v=ifCDXFdeaaM&t=210s

02 背景知识：条件扩散模型

➢ Conditional Diffusion Model

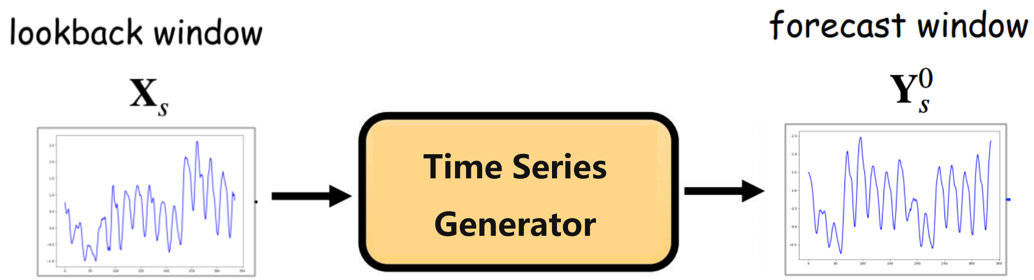




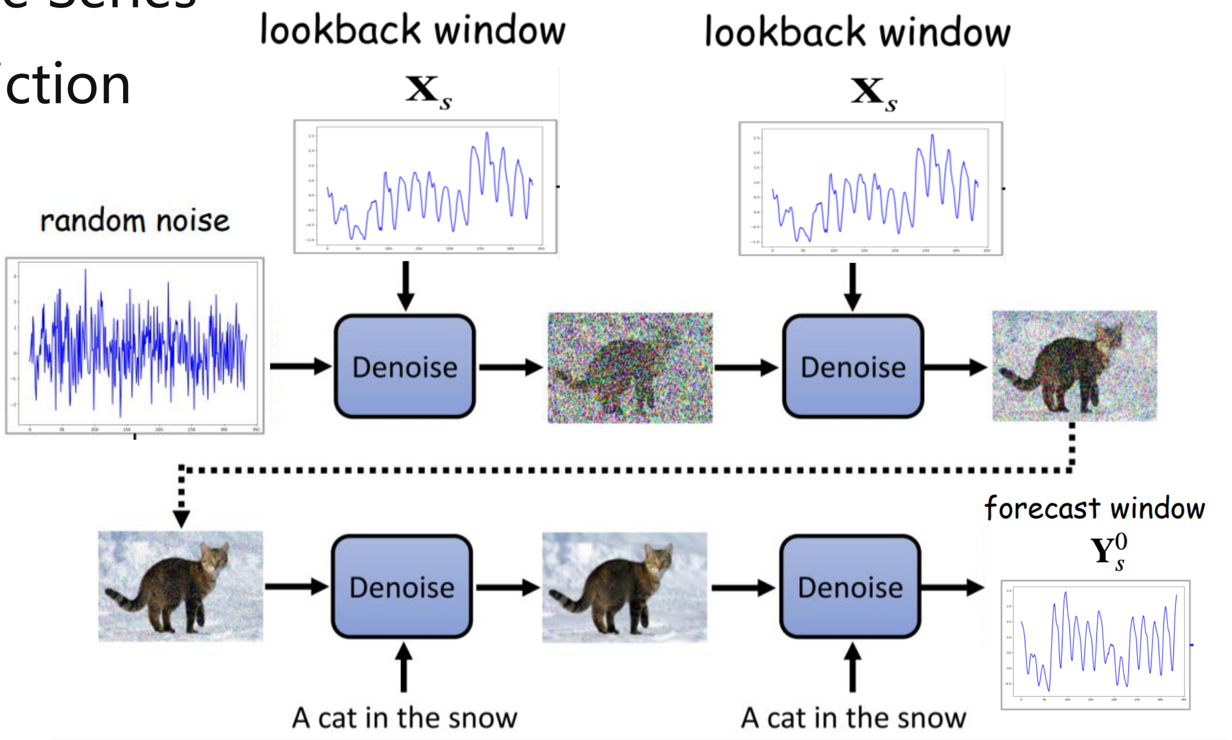
*李宏毅教授讲diffusion model：https://www.youtube.com/watch?v=ifCDXFdeaaM&t=210s

02 背景知识：条件扩散模型

- Conditional Diffusion Model

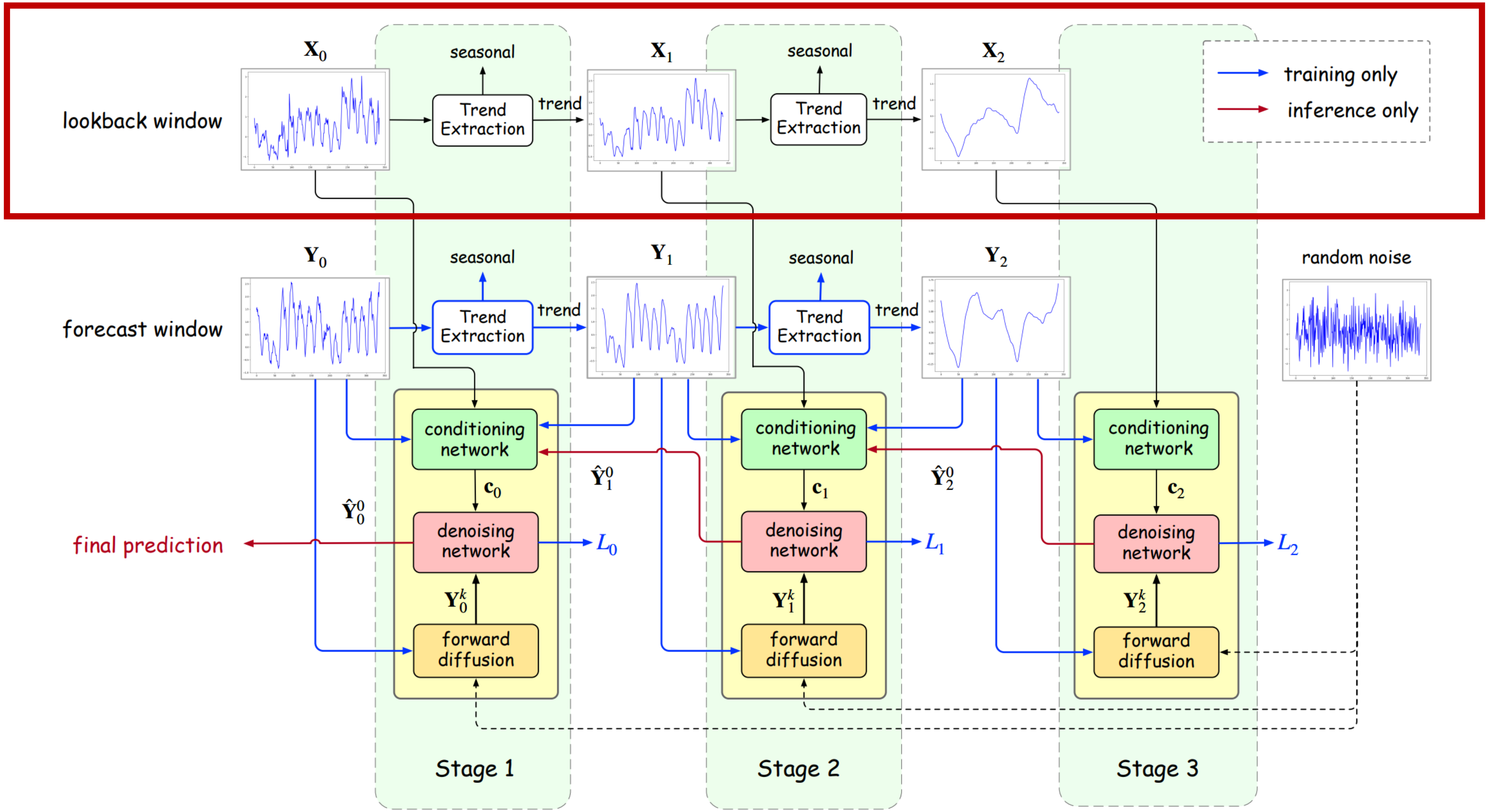


For Time Series Prediction



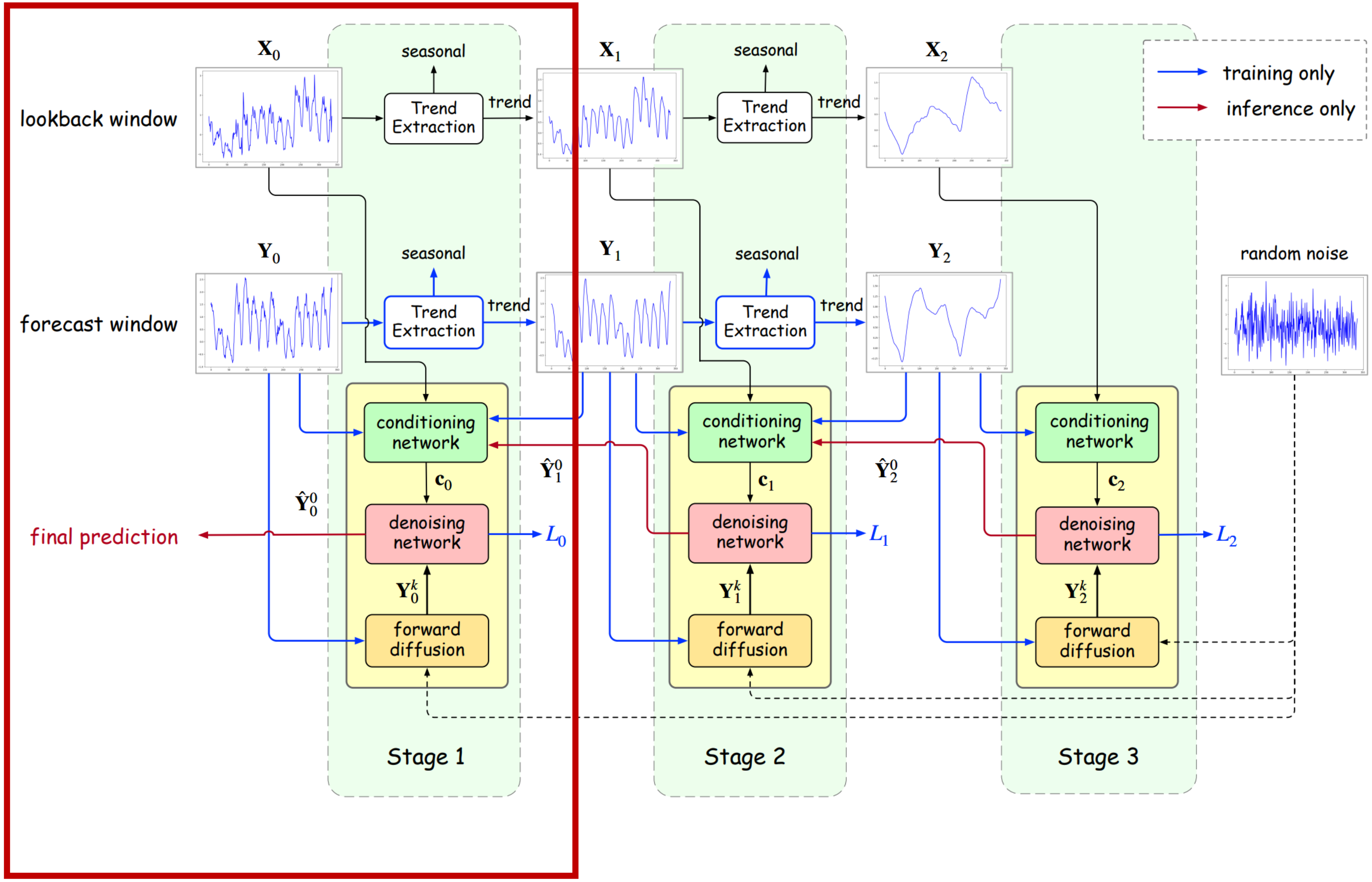
$$p_\theta(\mathbf{x}_{1:H}^{k-1} | \mathbf{x}_{1:H}^{k}, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{1:H}^{k-1}; \mu_\theta(\mathbf{x}_{1:H}^{k}, k|\mathbf{c}), \sigma_k^2 \mathbf{I}), \quad k = K, K-1, \ldots, 1.$$

03 模型结构：mr-diff



03

模型结构：mr-diff



03

模型结构：提取从精细到粗糙的趋势

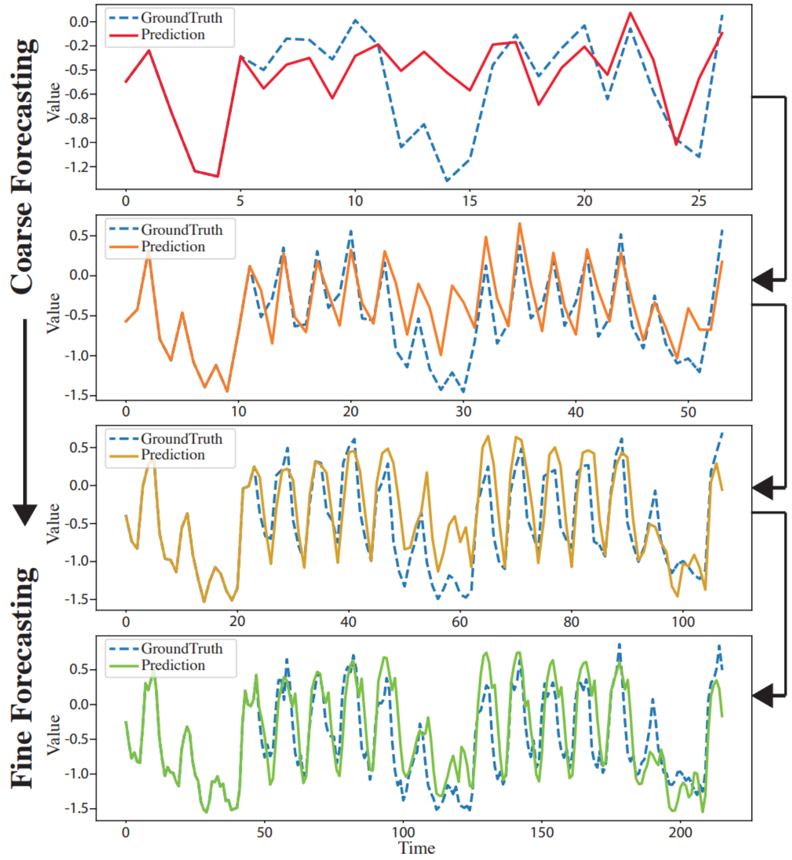
➢ **提取FINE-TO-COARSE的趋势：从精细到粗糙**

- Pooling窗口由小到大

$$\mathbf{X}_s = \mathtt{AvgPool}(\mathtt{Padding}(\mathbf{X}_{s-1}), \tau_s), \quad s = 1, \ldots, S-1,$$

➢ **只关注趋势分量**（不同于Autoformer和FEDformer）

- 使用扩散模型在不同的阶段/分辨率进行时间序列重建，从较粗的趋势中更容易预测更精细的趋势。
- 从较粗的季节成分重建更精细的季节成分可能很困难，特别是因为季节成分可能没有明确的模式。

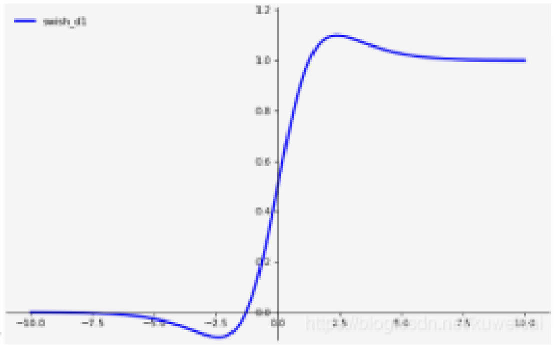


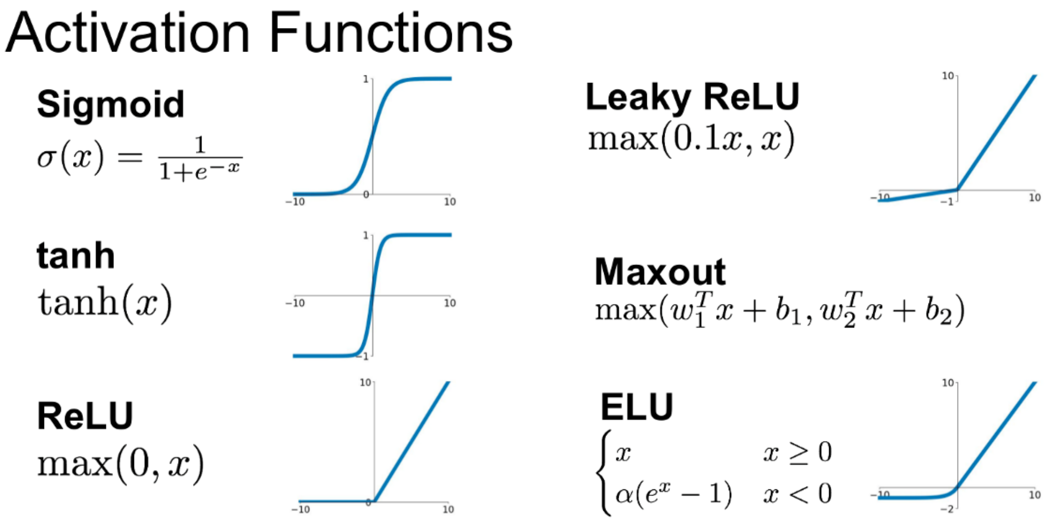
03 模型结构：时序多分辨率重建

➢ **维度嵌入：**Sinusoidal Position Embedding（Transformer）

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

- 两个全连接层
- SiLU激活函数：sigmoid 加权线性单元





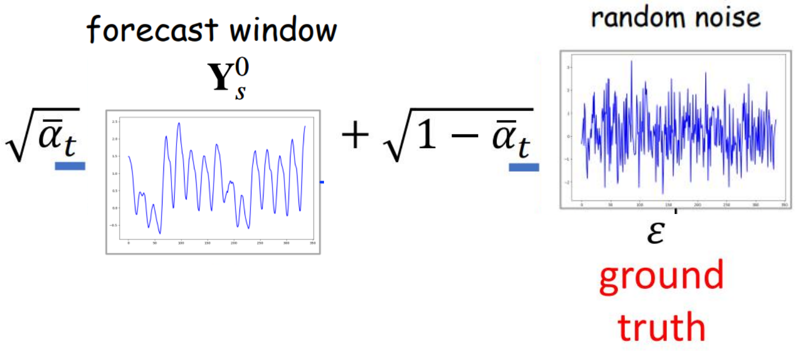
03 模型结构：时序多分辨率重建

- **前向扩散：**和原始DDPM的扩散过程相同
  - S阶段，扩散K步

Forward diffusion is straightforward. Analogous to (1), with $\mathbf{Y}_s^0 = \mathbf{Y}_s$, we obtain at step k,

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

where the noise matrix ϵ is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ with the same sizes as $\mathbf{Y}_s$.



03 模型结构：时序多分辨率重建

➢ **反向去噪：**将去噪目标分解为S个子目标，去噪过程从易到难

- Conditioning network
- Denoising network

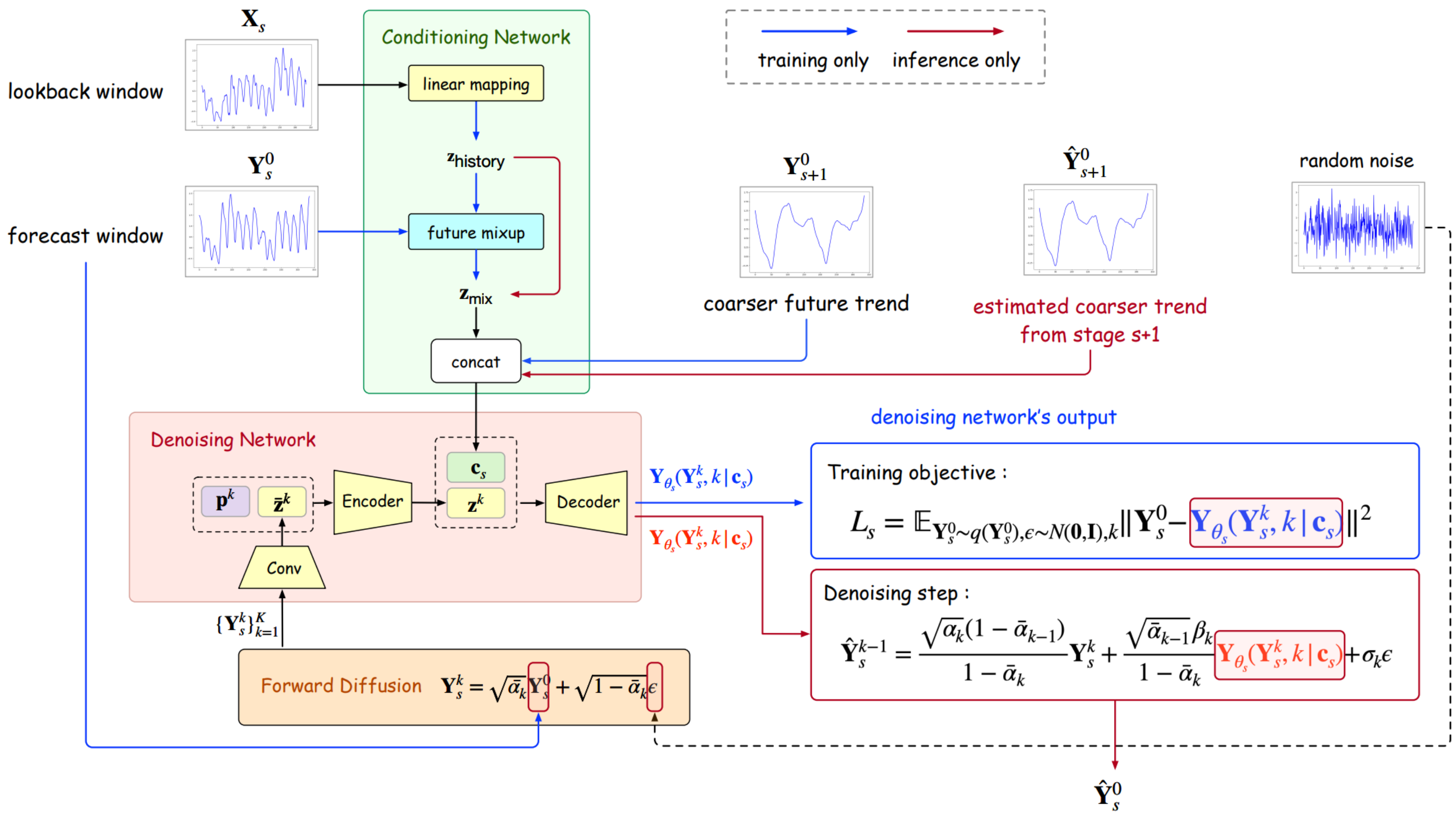


Figure 3: The conditioning network and denoising network.

03

模型结构：时序多分辨率重建

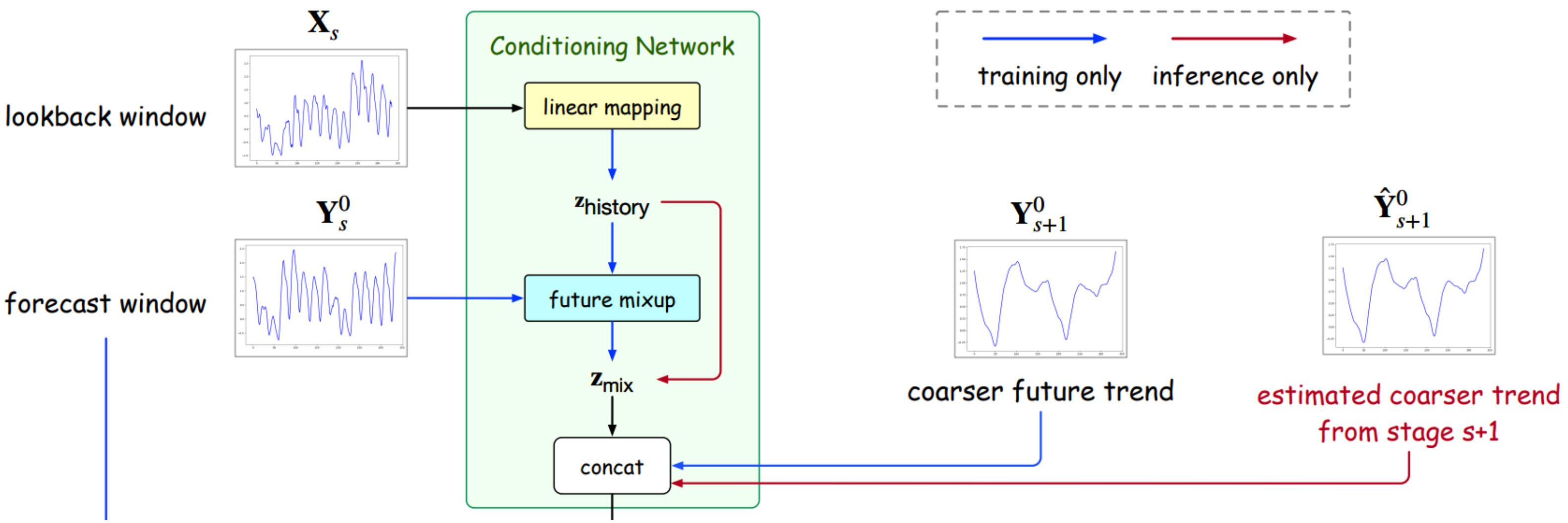
➢ Conditioning network：构造条件指导去噪网络

- 现有方法：将原始时间序列的回看窗口（历史序列）X_0作为条件c
- 多分辨率分解：在同一分解阶段S使用回看窗口X_S，$X_S \rightarrow z_{\text{history}}$

① future-mixup ($\rightarrow z_{mix}$)： $\mathbf{z}_{\text{mix}} = \mathbf{m} \odot \mathbf{z}_{\text{history}} + (1 - \mathbf{m}) \odot \mathbf{Y}_s^0,$

② concat $Y_{S+1}(\rightarrow c_S)$

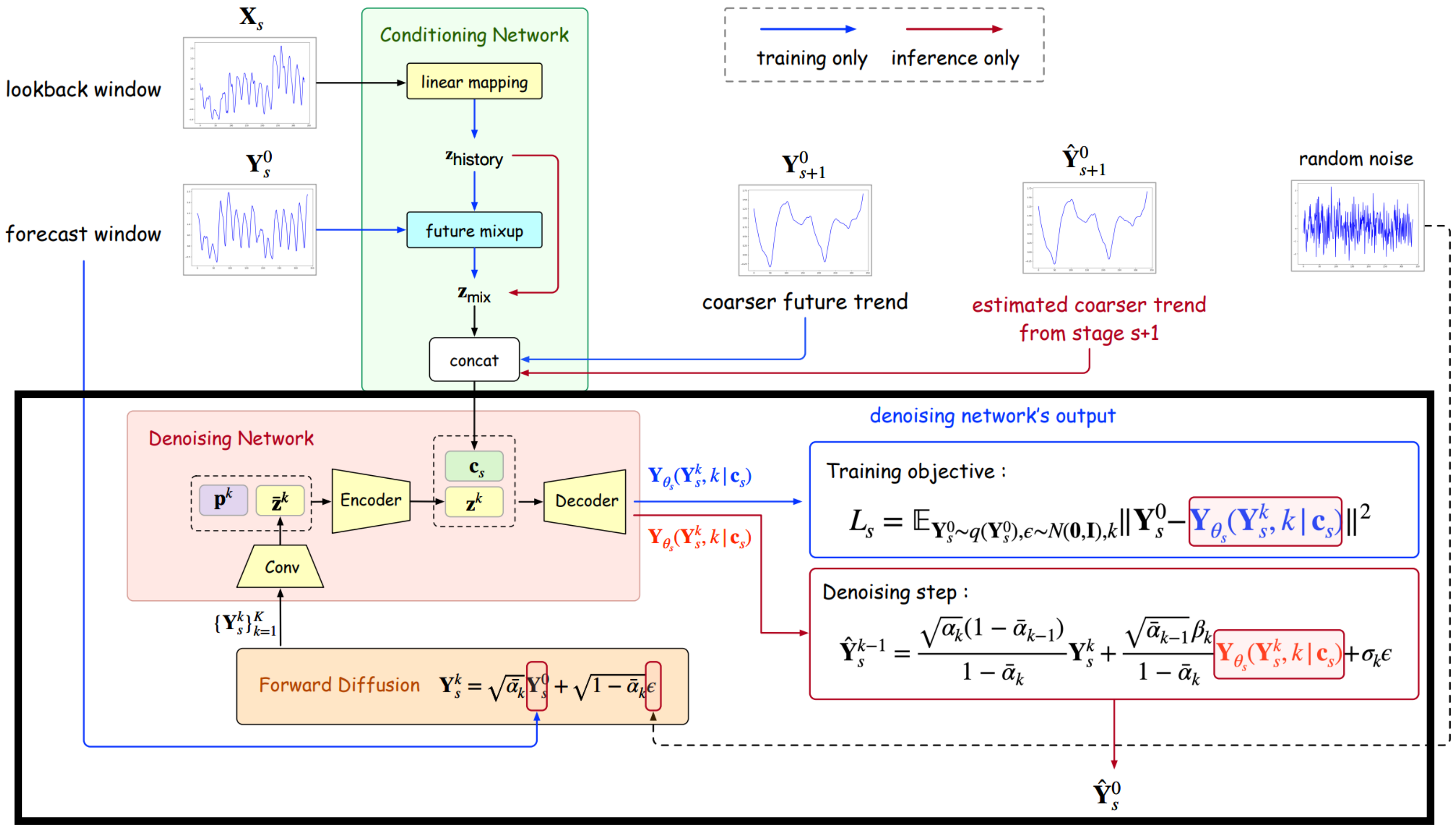
在推理阶段，①不用，②将ground truth替换为上一阶段（S+1）的预测结果



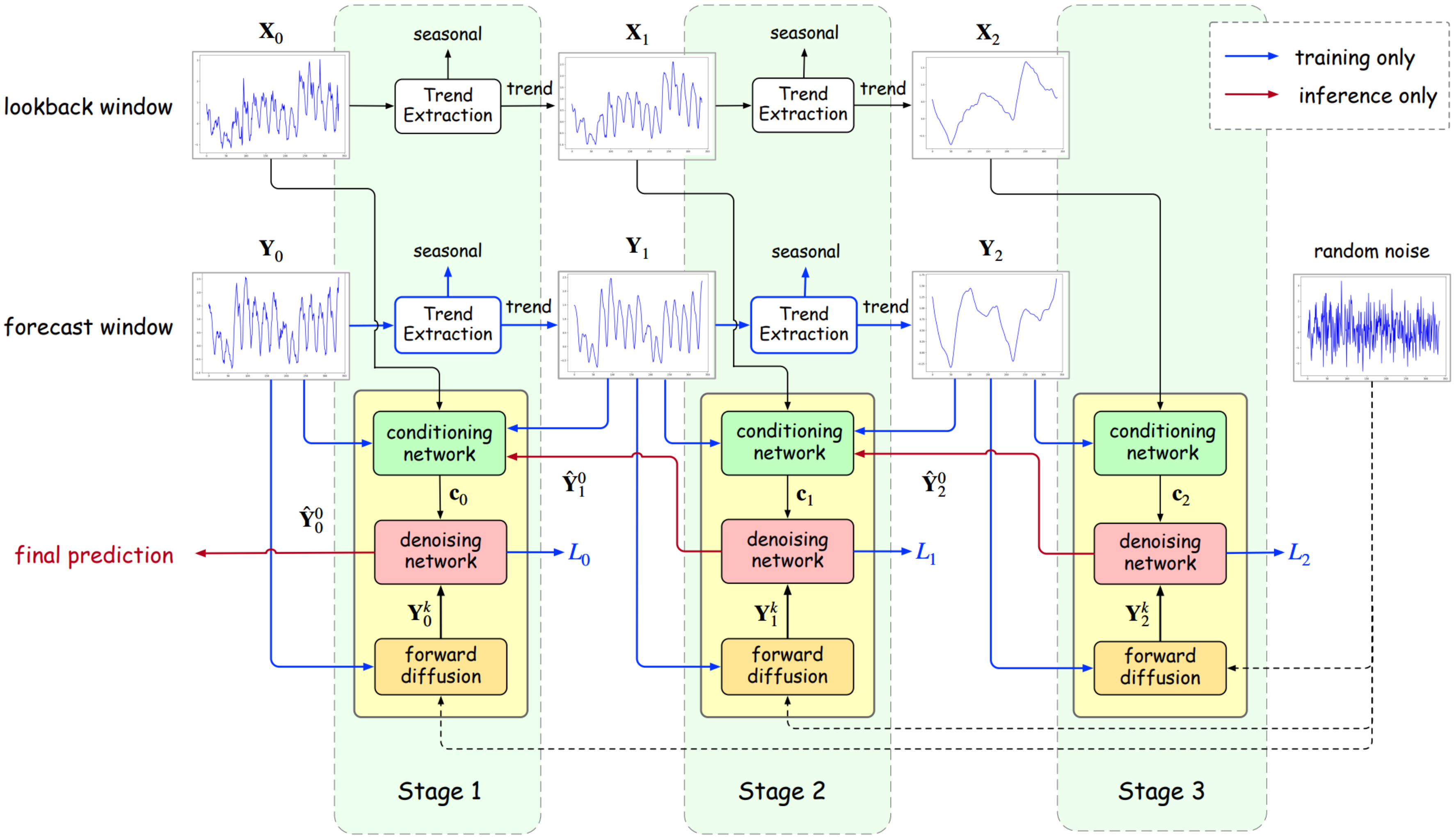
03 模型结构：时序多分辨率重建

➢ Denoising network：去噪网络

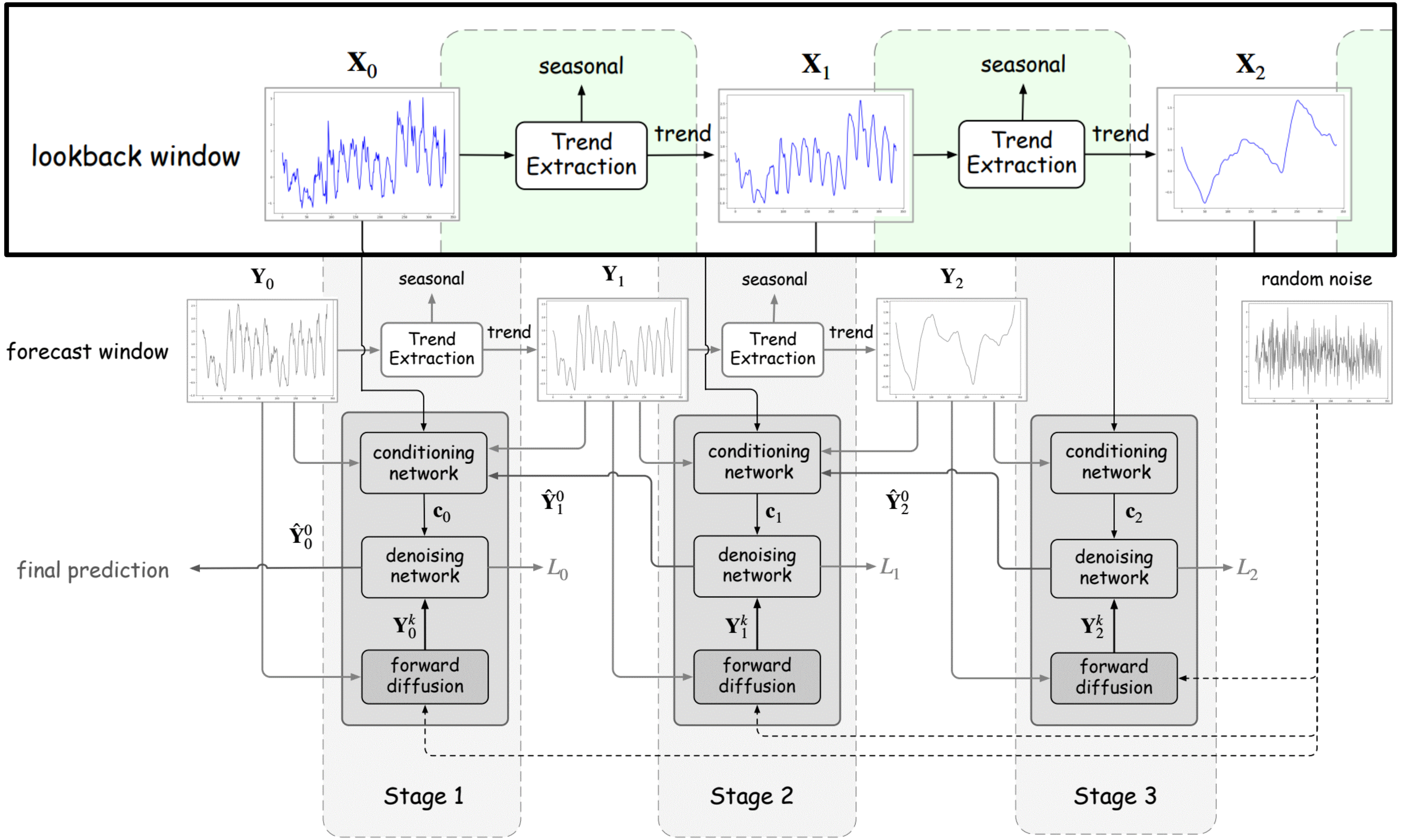
- 将数据嵌入、前向扩散的结果、条件作为输入，训练去噪网络



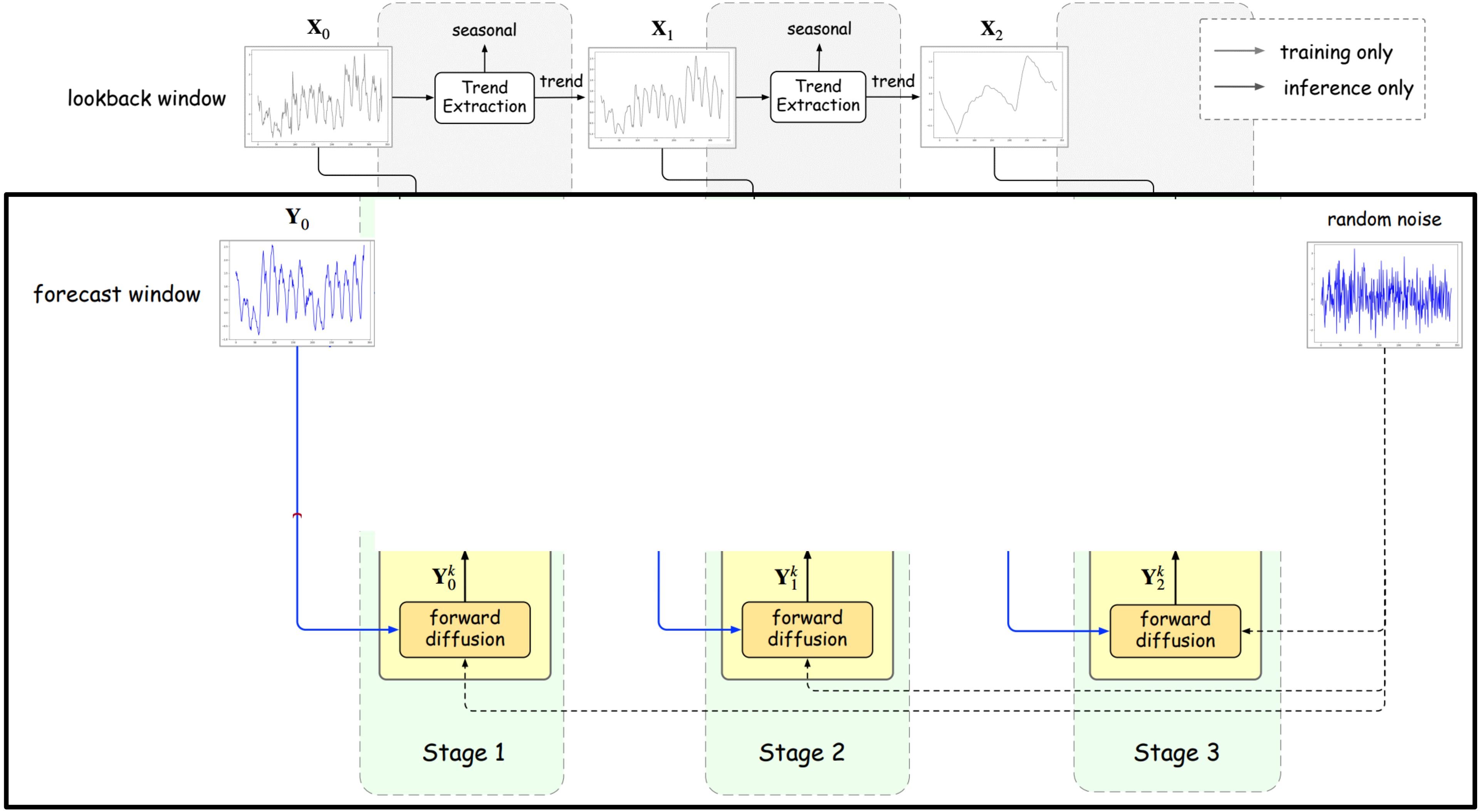
03 模型结构： mr-diff



03 模型结构： mr-diff

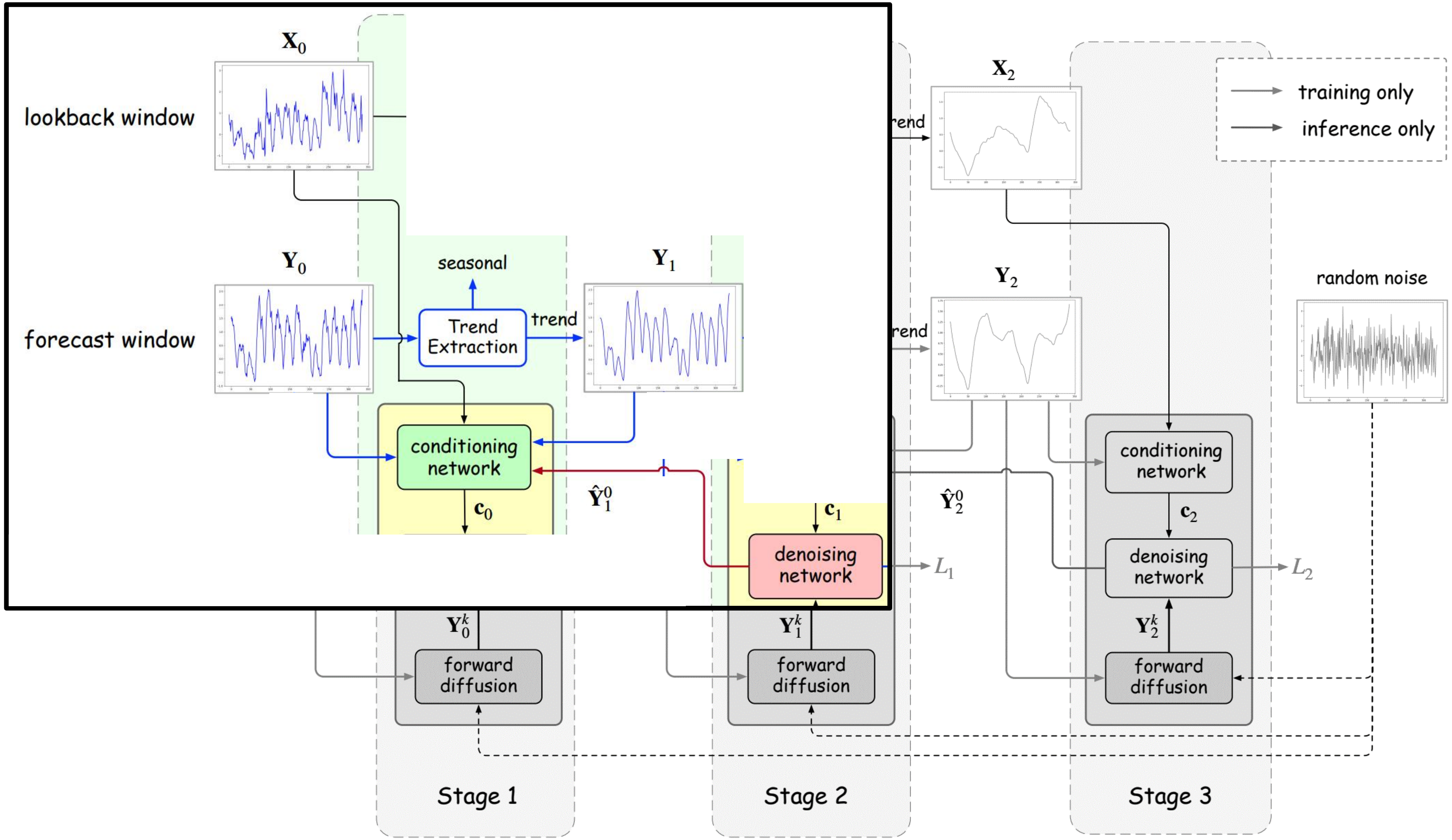


03 模型结构： mr-diff

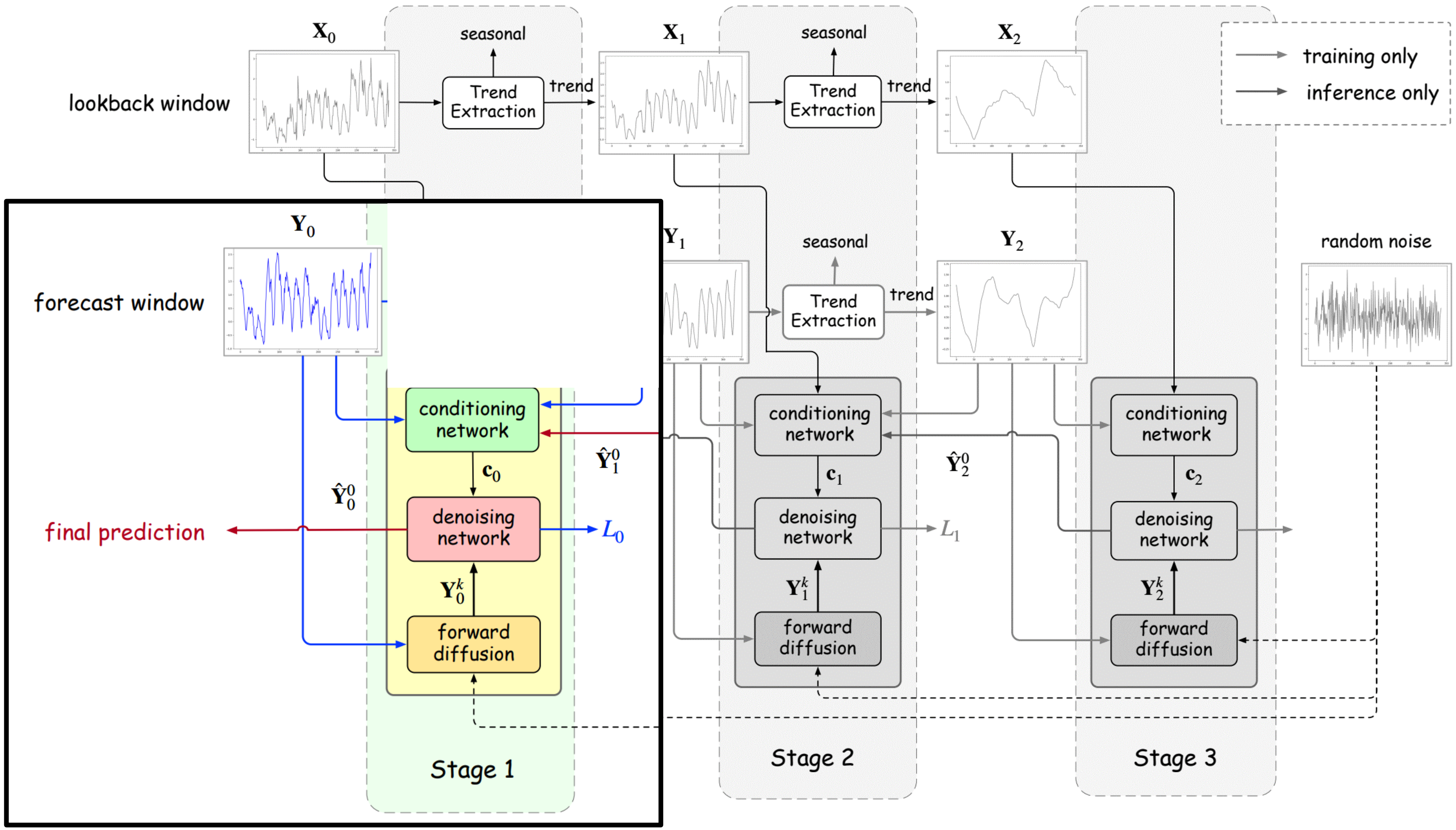


03

模型结构： mr-diff



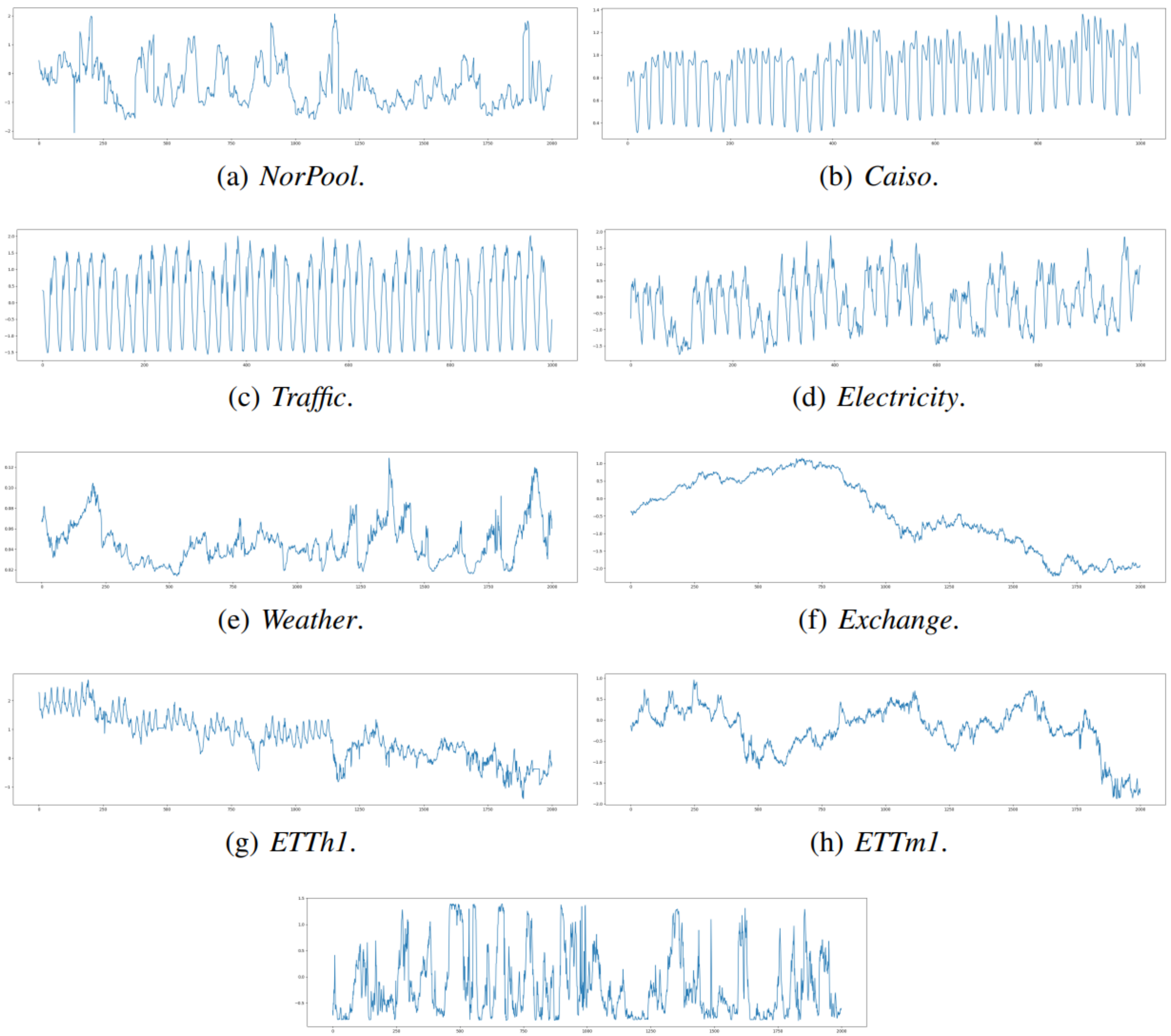
03 模型结构： mr-diff



04 实验：数据集

	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)

NorPool：多个欧洲国家的每小时能源生产量
Caiso：加州不同地区每小时的实际电力负荷序列
Traffic：旧金山高速公路车道的每小时交通数据
Electricity：用户的用户量
Weather：天气
Exchange：汇率
ETTh1/ETTm1：变压器油温
Wind：风电记录



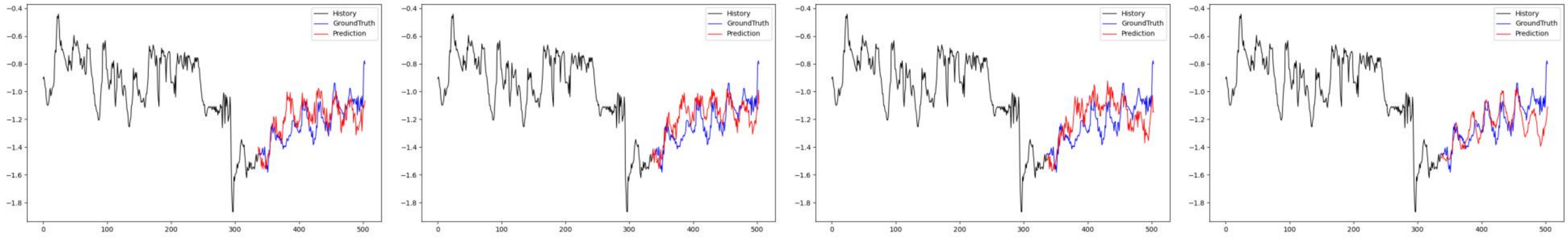
(a) *NorPool*. (b) *Caiso*. (c) *Traffic*. (d) *Electricity*. (e) *Weather*. (f) *Exchange*. (g) *ETTh1*. (h) *ETTm1*. (i) *Wind*.

Figure 5: Visualization of the time series datasets.

04 实验：对比实验—单变量MAE实验

	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^3VAE	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

04 实验：对比实验—单变量



(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`.

04

实验：对比实验—多变量MAE实验

	NorPool	*Caiso*	*Traffic*	*Electricity*	*Weather*	*Exchange*	*ETTh1*	*ETTm1*	*Wind*	avg rank
mr-Diff	0.604(2)	0.219(3)	0.320(5)	0.252(3)	0.324(2)	0.082(3)	0.422(2)	0.373(2)	**0.675**(1)	2.6
TimeDiff	0.611(3)	0.234(6)	0.384(8)	0.305(6)	**0.312**(1)	0.091(7)	0.430(3)	**0.372**(1)	0.687(3)	4.2
TimeGrad	0.821(19)	0.339(18)	0.849(22)	0.630(21)	0.381(14)	0.193(19)	0.719(22)	0.605(22)	0.793(21)	19.8
CSDI	0.777(17)	0.345(19)	-	-	0.374(12)	0.194(20)	0.438(5)	0.442(15)	0.741(10)	14.0
SSSD	0.753(13)	0.295(10)	0.398(13)	0.363(12)	0.350(8)	0.127(12)	0.561(17)	0.406(10)	0.778(18)	12.6
D^3VAE	0.692(9)	0.331(16)	0.483(17)	0.372(14)	0.380(13)	0.301(22)	0.502(14)	0.391(8)	0.779(19)	14.7
CPF	0.889(21)	0.424(21)	0.714(21)	0.643(22)	0.781(23)	0.082(3)	0.597(20)	0.472(16)	0.757(15)	18.0
PSA-GAN	0.890(22)	0.477(22)	0.697(20)	0.533(20)	0.578(22)	0.087(6)	0.546(16)	0.488(18)	0.756(13)	17.7
N-Hits	0.643(7)	0.221(4)	0.268(2)	0.245(2)	0.335(4)	0.085(5)	0.480(8)	0.388(6)	0.734(8)	5.1
FiLM	0.646(8)	0.278(8)	0.398(13)	0.320(8)	0.336(5)	**0.079**(1)	0.436(4)	0.374(3)	0.717(5)	6.1
Depts	0.611(3)	0.204(2)	0.568(19)	0.401(17)	0.394(16)	0.100(9)	0.491(12)	0.412(12)	0.751(12)	11.3
NBeats	0.832(20)	0.235(7)	**0.265**(1)	0.370(13)	0.420(17)	0.081(2)	0.521(15)	0.409(11)	0.741(10.5)	10.7
Scaleformer	0.769(16)	0.310(12)	0.379(7)	0.304(5)	0.438(18)	0.138(14)	0.579(19)	0.475(17)	0.864(22)	14.4
PatchTST	0.710(10)	0.293(9)	0.411(16)	0.348(11)	0.555(21)	0.147(15)	0.489(11)	0.392(9)	0.720(6)	12.0
FedFormer	0.744(11)	0.317(13)	0.385(9)	0.341(10)	0.347(7)	0.233(21)	0.484(9)	0.413(13)	0.762(16)	12.1
Autoformer	0.751(12)	0.321(14)	0.392(12)	0.313(7)	0.354(9)	0.167(16)	0.484(9)	0.496(19)	0.756(13)	12.3
Pyraformer	0.781(18)	0.371(20)	0.390(10)	0.379(15)	0.385(15)	0.112(11)	0.493(13)	0.435(14)	0.735(9)	13.9
Informer	0.757(14)	0.336(17)	0.391(11)	0.383(16)	0.364(10)	0.192(18)	0.605(21)	0.542(20)	0.772(17)	16.0
Transformer	0.765(15)	0.321(14)	0.410(15)	0.405(18)	0.370(11)	0.178(17)	0.567(18)	0.592(21)	0.785(20)	16.6
SCINet	**0.601**(1)	**0.193**(1)	0.335(6)	0.280(4)	0.344(6)	0.137(13)	0.463(7)	0.389(7)	0.732(7)	5.8
NLinear	0.636(5)	0.223(5)	0.293(4)	**0.239**(1)	0.328(3)	0.091(7)	**0.418**(1)	0.375(4)	0.706(4)	3.8
DLinear	0.640(6)	0.497(23)	0.268(2)	0.336(9)	0.444(19)	0.102(10)	0.442(6)	0.378(5)	0.686(2)	9.1
LSTMa	0.974(23)	0.305(11)	0.510(18)	0.444(19)	0.501(20)	0.534(20)	0.782(23)	0.699(23)	0.897(23)	20.3

实验：效率实验

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

04 实验：消融实验

Table 6: Prediction MAE versus lookback window length L.

L	*Electricity*	*ETTh1*	*ETTm1*
96	0.449	0.210	0.165
192	0.376	0.202	0.163
336	0.362	**0.196**	0.157
720	**0.332**	0.196	0.152
1,440	0.346	0.199	**0.149**

Rebuttle回复：主要限制是回溯窗口需要足够长且信息丰富。当回溯窗口太短时，提取多分辨率趋势变得具有挑战性，并且性能改进可能不那么明显。

Table 8: Prediction MAE versus number of stages S. L is set to the optimal setting in Table 6 (i.e., *Electricity*: 720, *ETTh1*: 336, *ETTm1*: 1440.

S	smoothing kernel size τ_i's	*Electricity*	*ETTh1*	*ETTm1*
1	without seasonal-trend decomposition	0.403	0.208	0.1573
2	5	0.389	0.200	0.1529
3	5, 25	0.363	**0.196**	0.1525
4	5, 25, 51	0.346	0.197	0.1509
5	5, 25, 51, 201	**0.332**	0.197	**0.1496**

谢谢观看

MANY THANKS !

24.2.6