ICLR 2024

Multi-Resolution Diffusion Models For Time Series Forecasting

mr-Diff

- Conditioning Network（condition）

- Conditional Diffusion Network（Mutli-Resolution）

Extracting Fine-To-Coarse Trends

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

$$\{\mathbf{Y}_s\}_{s=1,\ldots,S-1}$$

Embedding

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

Forward Diffusion

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

Backward Denoising

Conditioning Network

$$\mathbf{X}_s \longrightarrow \mathbf{z}_{\texttt{history}}$$

$$\mathbf{z}_{\texttt{mix}} = \mathbf{m} \odot \mathbf{z}_{\texttt{history}} + (1-\mathbf{m}) \odot \mathbf{Y}_s^0 \qquad \mathbf{m} \in [0,1)$$

$$\mathbf{Y}_{s+1} + \mathbf{z}_{\texttt{mix}} = \mathbf{c}_s$$

Denoising Network

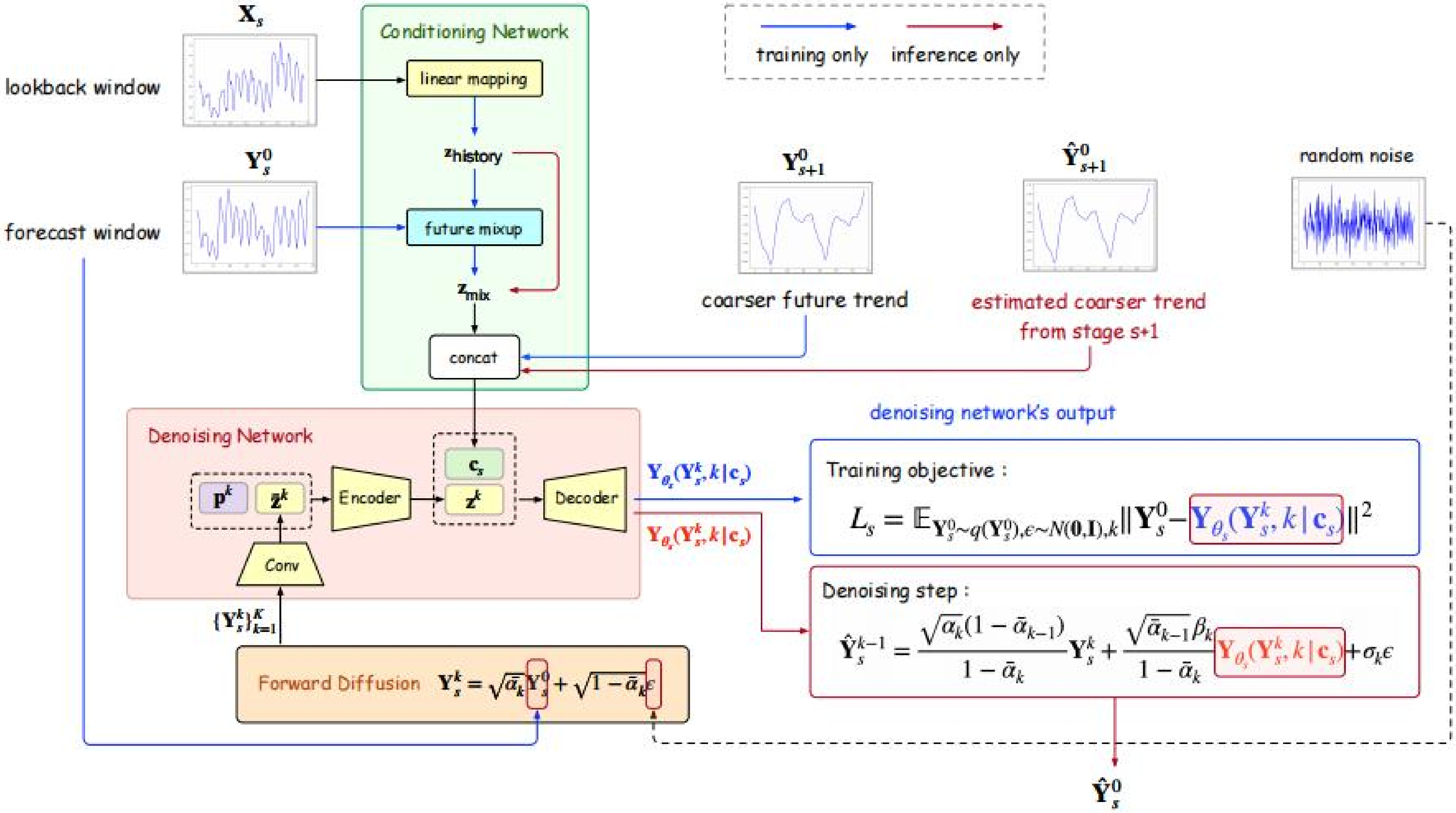
$$p_{\theta_s}(\mathbf{Y}_s^{k-1}|\mathbf{Y}_s^k, \mathbf{c}_s) = \mathcal{N}(\mathbf{Y}_s^{k-1}; \mu_{\theta_s}(\mathbf{Y}_s^k, k|\mathbf{c}_s, \sigma_k^2\mathbf{I})), \quad k = K, \dots, 1,$$

$$\mu_{\theta_s}(\mathbf{Y}_s^k, k|\mathbf{c}_s, \sigma_k^2\mathbf{I}) = \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\mathbf{Y}_s^k + \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\mathbf{Y}_{\theta_s}(\mathbf{Y}_s^k, k|\mathbf{c}_s)$$

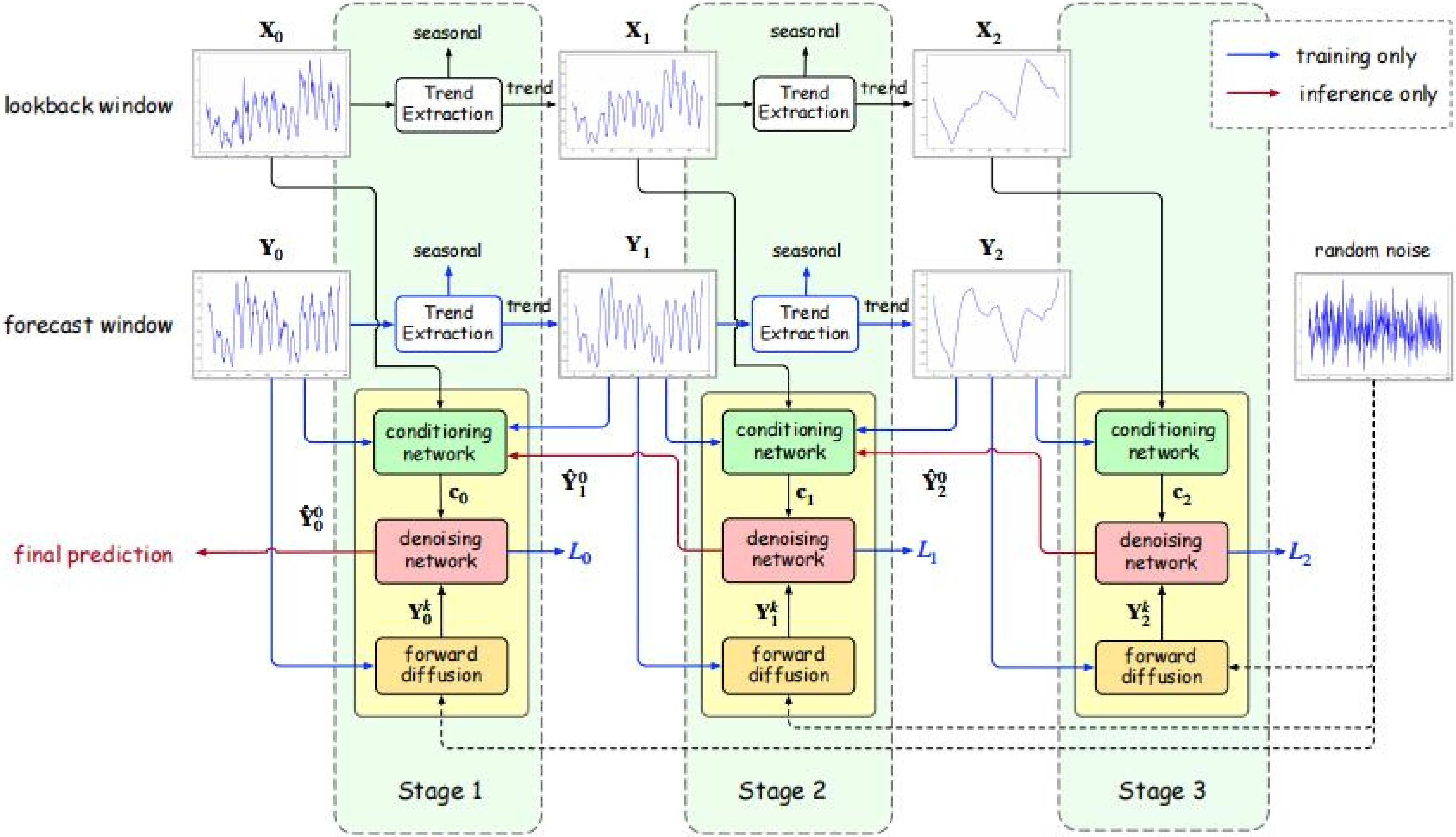
$$\mathbf{Y}_s^k \xrightarrow{\text{Convolutional layers}} \bar{\mathbf{z}}^k + \mathbf{p}^k \xrightarrow{\text{Encoder}} \mathbf{z}^k + \mathbf{c}_s \xrightarrow{\text{Decoder}} \mathbf{Y}_{\theta_s}$$

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

$$\hat{\mathbf{Y}}_s^{k-1} = \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\hat{\mathbf{Y}}_s^k + \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\mathbf{Y}_{\theta_s}(\hat{\mathbf{Y}}_s^k, k|\mathbf{c}_s) + \sigma_k\epsilon$$



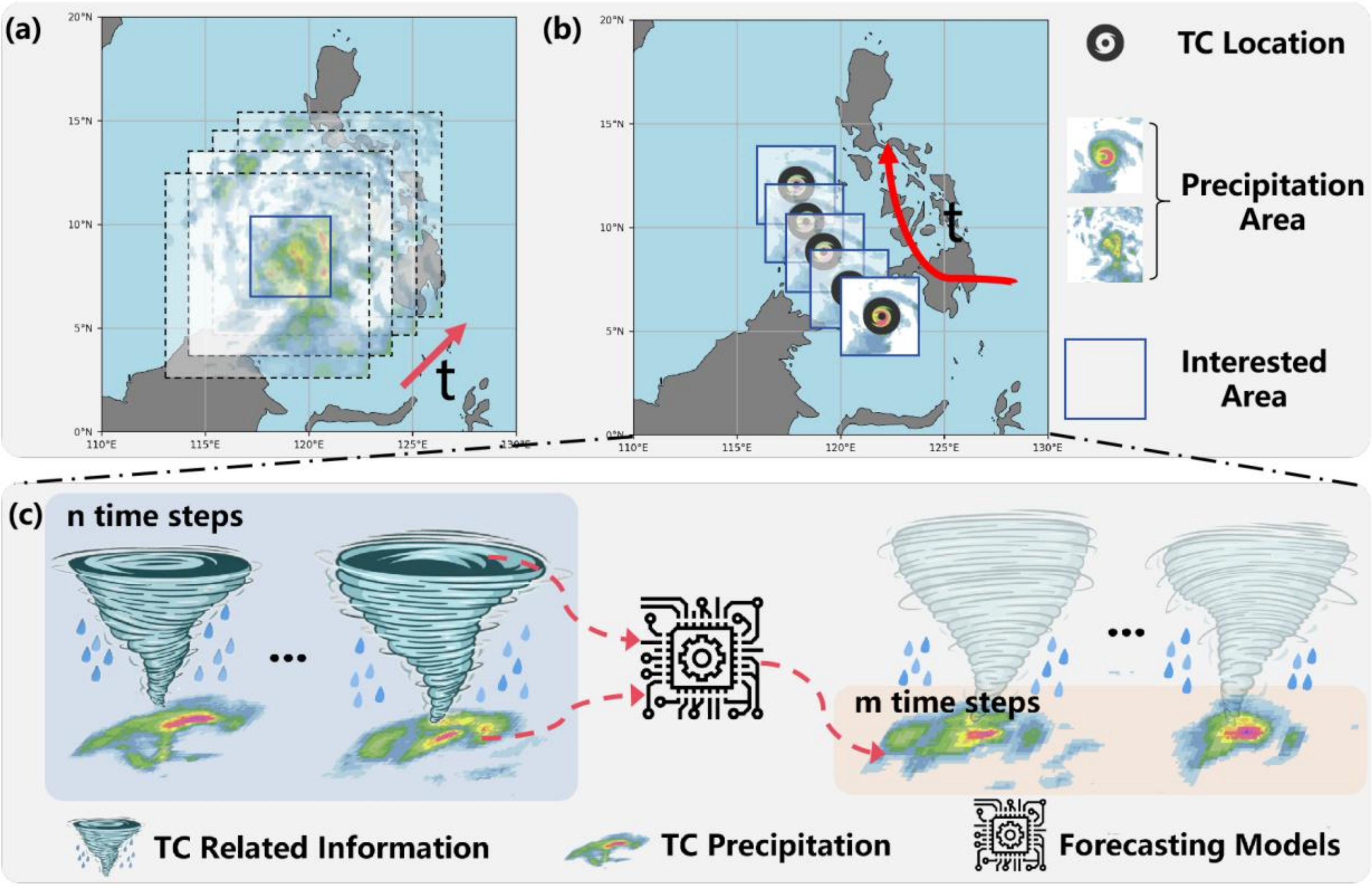
$\mathbf{X}_s$
lookback window
Conditioning Network
linear mapping
$z_{history}$
$\mathbf{Y}_s^0$
forecast window
future mixup
z_{mix}
concat
training only
inference only
$\mathbf{Y}_{s+1}^0$
coarser future trend
$\hat{\mathbf{Y}}_{s+1}^0$
estimated coarser trend from stage s+1
random noise
denoising network's output
Denoising Network
$\mathbf{p}^k$
$\mathbf{z}^k$
Encoder
$\mathbf{c}_s$
$\mathbf{z}^k$
Decoder
Conv
$\{\mathbf{Y}_s^k\}_{k=1}^K$
$\mathbf{Y}_{\theta_s}(\mathbf{Y}_s^k, k|\mathbf{c}_s)$
$\mathbf{Y}_{\theta_s}(\mathbf{Y}_s^k, k|\mathbf{c}_s)$
Training objective :
$L_s = \mathbb{E}_{\mathbf{Y}_s^0 \sim q(\mathbf{Y}_s^0), \epsilon \sim N(\mathbf{0},\mathbf{I}), k} \| \mathbf{Y}_s^0 - \boxed{\mathbf{Y}_{\theta_s}(\mathbf{Y}_s^k, k|\mathbf{c}_s)} \|^2$
Denoising step :
$\hat{\mathbf{Y}}_s^{k-1} = \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\mathbf{Y}_s^k + \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\boxed{\mathbf{Y}_{\theta_s}(\mathbf{Y}_s^k, k|\mathbf{c}_s)} + \sigma_k\epsilon$
Forward Diffusion
$\mathbf{Y}_s^k = \sqrt{\bar{\alpha}_k}\boxed{\mathbf{Y}_s^0} + \sqrt{1-\bar{\alpha}_k}\boxed{\epsilon}$
$\hat{\mathbf{Y}}_s^0$



lookback window
forecast window
$\mathbf{X}_0$
$\mathbf{X}_1$
$\mathbf{X}_2$
$\mathbf{Y}_0$
$\mathbf{Y}_1$
$\mathbf{Y}_2$
seasonal
trend
Trend Extraction
training only
inference only
random noise
conditioning network
denoising network
forward diffusion
$\mathbf{c}_0$
$\mathbf{c}_1$
$\mathbf{c}_2$
$\hat{\mathbf{Y}}_0^0$
$\hat{\mathbf{Y}}_1^0$
$\hat{\mathbf{Y}}_2^0$
$\mathbf{Y}_0^k$
$\mathbf{Y}_1^k$
$\mathbf{Y}_2^k$
L_0
L_1
L_2
final prediction
Stage 1
Stage 2
Stage 3

ICML2025

TCP-Diffusion: A Multi-modal Diffusion Model for Global Tropical Cyclone Precipitation Forecasting with Change Awareness



Improving TC rainfall prediction

- **Changing the training goal**

$$Rainfall_{Future} = Rainfall_{Current} + \Delta Rainfall_{Future}$$

- **Extracting richer meteorological information**

- **Integrating with NWP models**

Numerical weather prediction (NWP)

Model Design

$$X_{historical} = \{X_1^h, X_2^h, \ldots, X_t^h, \ldots, X_n^h\}$$

$$X_{future} = \{X_1^f, X_2^f, \ldots, X_t^f, \ldots, X_m^f\}$$

$$X = \{X_{historical}, X_{future}\}$$

$$\downarrow$$

$$\hat{Y} = \{\hat{y}_{n+1}, \hat{y}_{n+2}, \ldots, \hat{y}_{n+t}, \ldots, \hat{y}_{n+m}\}$$

Adjacent Residual Prediction (ARP)

$$\Delta_x^t = X_{rain}^t - X_{rain}^{t-1}$$

$$D_x = \{\Delta_x^1, \Delta_x^2, \ldots, \Delta_x^t, \ldots, \Delta_x^n\}$$

$$\hat{D}_y = \{\hat{\Delta}_y^{n+1}, \hat{\Delta}_y^{n+2}, \ldots, \hat{\Delta}_y^{n+t}, \ldots, \hat{\Delta}_y^{n+m}\}$$

$$\hat{y}_{n+t} = X_{rain}^n + \sum_{z=1}^{t} \hat{\Delta}_y^{n+z}$$

Diffusion Model

- Diffusion Process

$$D_y^s = \sqrt{\bar{\alpha}_s} D_y^0 + \sqrt{1 - \bar{\alpha}_s} r_s$$

- Denoising Process（Environmentally-Aware 3DUNet，*EA3DUNet*）

$$\hat{r}_s = EA3DUNet(X_{history}, X_{future}, D_y^s, s, r)$$

$$L(\theta) = \|r_s - \hat{r}_s\|_2$$

$$D_y^{s-1} = \frac{1}{\sqrt{\alpha_s}}(D_y^s - \frac{\beta_s}{\sqrt{1 - \bar{\alpha}_s}} \hat{r}_s) + \sigma_s \epsilon$$

Environmentally-Aware 3DUNet (EA-3DUNet)

Historical Data Encoder

1) $$e_{his2D} = Conv3d(X_{his2D}, W_{his2D})$$

2) $$e_{mlp} = \phi(X_{Sc}, W_{mlp})$$

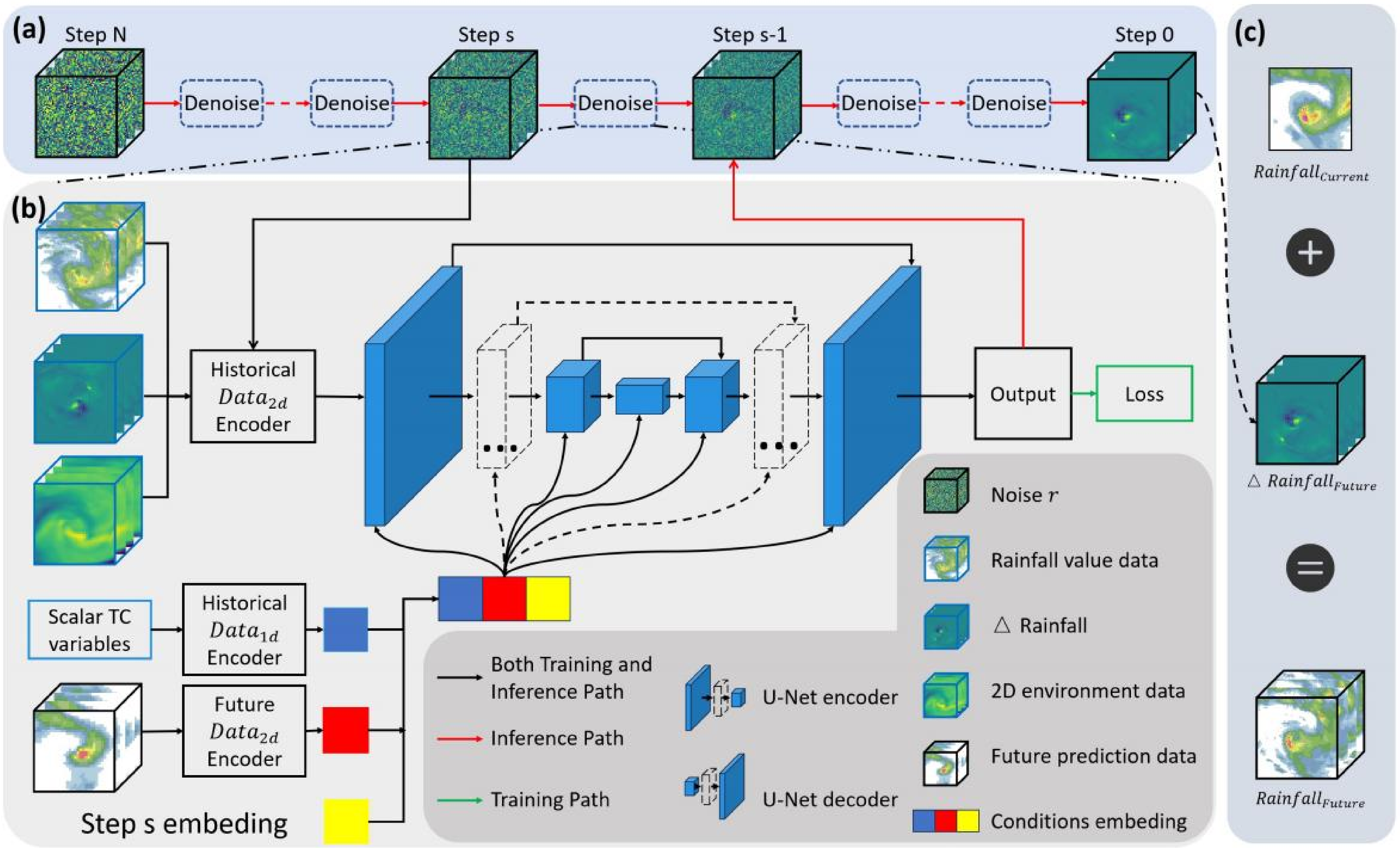
$$e_{his1D} = \text{Transformer}(e_{mlp}, W_{atten})$$

Future Prediction Data Encoder

$$e_{future} = \text{Resnet}(X_{future}, W_{res18})$$

3DUnet

1) U-Net encoder
2) U-Net decoder
3) bottleneck between encoder and decoder



(a)
Step N
Denoise
Denoise
Step s
Denoise
Step s-1
Denoise
Denoise
Step 0
(c)
$Rainfall_{Current}$
$\triangle\ Rainfall_{Future}$
$Rainfall_{Future}$
(b)
Historical $Data_{2d}$ Encoder
Output
Loss
Scalar TC variables
Historical $Data_{1d}$ Encoder
Future $Data_{2d}$ Encoder
Step s embeding
Both Training and Inference Path
Inference Path
Training Path
U-Net encoder
U-Net decoder
Noise r
Rainfall value data
△ Rainfall
2D environment data
Future prediction data
Conditions embeding