A Mathematical Derivations

A.1 Derivation of Multi-Resolution Diffusion Transitions

Given the diffusion trajectory modified by cross-scale correlation, which is defined as Eq. (1), we now derive the multi-resolution diffusion transitions:

$$q_\phi(\mathbf{x}_k^r|\mathbf{x}_{k-1}^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1}) = \mathcal{N}(\sqrt{\alpha_k}\mathbf{x}_{k-1}^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) - \sqrt{\alpha_k}\gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1), \beta_k\mathbf{I}). \tag{9}$$

We can achieve this by proving that Lemma. 1.

lemma 1 *Given the forward process defined as* $q(\mathbf{x}_1^r,\mathbf{x}_2^r,...,\mathbf{x}_k^r|\mathbf{x}_0^r) = \prod_{t=1}^T q(\mathbf{x}_k^r|\mathbf{x}_{k-1}^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1})$*, where the diffusion transitions* $q(\mathbf{x}_k^r|\mathbf{x}_{k-1}^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1})$ *are defined as:*

$$q_\phi(\mathbf{x}_k^r|\mathbf{x}_0^r,\mathbf{x}_0^{r+1}) = \mathcal{N}(\mathbf{x}_k^r, \sqrt{\bar\alpha_k}\mathbf{x}_0^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k), (1-\bar\alpha_k)\mathbf{I}). \tag{10}$$

Proof 1 *We can prove this lemma by induction. Assume that at time k, both* $q(\mathbf{x}_k^r|\mathbf{x}_{k-1}^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1})$ *and* $q(\mathbf{x}_{k-1}^r|\mathbf{x}_0^r,\mathbf{x}_0^{r+1})$ *adhere to their respective distributions as in Eq.(3) and Eq. (1). We need to prove that* $q(\mathbf{x}_k^r|\mathbf{x}_0^r,\mathbf{x}_0^{r+1}) = \mathcal{N}(\mathbf{x}_k^r; \sqrt{\bar\alpha_k}\mathbf{x}_0^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k), (1-\bar\alpha_k)\boldsymbol{I})$.

We can rewrite $q(\mathbf{x}_k^r|\mathbf{x}_{k-1}^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1})$ *and* $q(\mathbf{x}_{k-1}^r|\mathbf{x}_0^r,\mathbf{x}_0^{r+1})$ *as follows:*

$$\mathbf{x}_k^r = \sqrt{\alpha_k}\mathbf{x}_{k-1}^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) - \sqrt{\alpha_k}\gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1) + \sqrt{\beta_k}\epsilon_1, \tag{11}$$

$$\mathbf{x}_{k-1}^r = \sqrt{\bar\alpha_{t-1}}\mathbf{x}_0^r + \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1) + \sqrt{1-\bar\alpha_{t-1}}\epsilon_2, \tag{12}$$

where ϵ_1 *and* ϵ_2 *denote independent standard Gaussian variables. Substituting* $\mathbf{x}_{k-1}^r$ *from the latter equation into the former, we obtain:*

$$\begin{aligned}\mathbf{x}_k^r &= \sqrt{\bar\alpha_k}\mathbf{x}_0^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) + \sqrt{\beta_k}\epsilon_1 + \sqrt{\alpha_k*(1-\bar\alpha_{k-1})}*\epsilon_2 \\ &= \sqrt{\bar\alpha_k}\mathbf{x}_0^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) + \sqrt{\alpha_t(1-\bar\alpha k-1)+\beta_k}\epsilon,\end{aligned} \tag{13}$$

where ϵ *is a Gaussian noise resulting from a linear combination of* ϵ_1 *and* ϵ_2*. To this end,* $q_\phi(\mathbf{x}_k^r|\mathbf{x}_{k-1}^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1})$ *with mean* $\sqrt{\bar\alpha_k}\mathbf{x}_0^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k)$ *and variance* ϵ *admits the expected distribution.*

A.2 Derivation of Posterior Distributions of Multi-Resolution Diffusion Process

Given the modified diffusion trajectories in Eq. (1) and the diffusion transitions in Eq. (2), we now derive the posterior distributions of multi-resolution diffusion process:

$$\begin{aligned}q_\phi(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1}) =& \mathcal{N}(\frac{\sqrt{\bar\alpha_{k-1}}\beta_k}{1-\bar\alpha_k}\mathbf{x}_0^r + \frac{\sqrt{\alpha_k}(1-\bar\alpha_{k-1})}{1-\bar\alpha_k}(\mathbf{x}_k^r - \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k)) \\ &+ \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1), \frac{(1-\bar\alpha_{k-1})\beta_k}{1-\bar\alpha_k}\mathbf{I}).\end{aligned} \tag{14}$$

Proof 2 *By Bayes's rule, we have:*

$$q(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1}) = \frac{q(\mathbf{x}_{k-1}^r|\mathbf{x}_0^r,\mathbf{x}_0^{r+1})q(\mathbf{x}_k^r|\mathbf{x}_{k-1}^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1})}{q(\mathbf{x}_k^r|\mathbf{x}_0^r,\mathbf{x}_0^{r+1})}. \tag{15}$$

Given that the numerator and denominator are both Gaussian, the posterior distribution is also Gaussian, and we can proceed to calculate its mean and variance:

$$\begin{aligned}q(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^r,\mathbf{x}_0^r) =& \frac{\mathcal{N}(\mathbf{x}_{k-1}^r, \sqrt{\bar\alpha_{k-1}}\mathbf{x}_0^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k), (1-\bar\alpha_{t-1})\boldsymbol{I})}{\mathcal{N}(\mathbf{x}_k^r, \sqrt{\bar\alpha_k}\mathbf{x}_0^r + \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1), (1-\bar\alpha_{t-1})\boldsymbol{I})} * \\ & \frac{\mathcal{N}(\boldsymbol{x}_t, \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) - \sqrt{\alpha_t}\gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1), \beta_t\boldsymbol{I})}{\mathcal{N}(\mathbf{x}_k^r, \sqrt{\bar\alpha_k}\mathbf{x}_0^r + \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1), (1-\bar\alpha_{t-1})\boldsymbol{I})},\end{aligned} \tag{16}$$

Dropping the constants that are unrelated to $\mathbf{x}_0^r$, $\mathbf{x}_k^r$, $\mathbf{x}_{k-1}^r$, and $\mathbf{x}_0^{r+1}$, we have:

$$
\begin{aligned}
q(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1}) \propto & \exp\{-\frac{(\mathbf{x}_{k-1}^r-\sqrt{\bar{\alpha}_{k-1}}\mathbf{x}_0^r-\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k))^2}{2(1-\bar{\alpha}_{k-1})} \\
& + \frac{(\mathbf{x}_k^r-\sqrt{\bar{\alpha}_k}\mathbf{x}_0^r-\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k))^2}{2(1-\bar{\alpha}_k)} \\
& - \frac{(\mathbf{x}_k^r-\sqrt{\alpha_k}\mathbf{x}_{k-1}^r-\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k)+\sqrt{\alpha_k}\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k))^2}{2\beta_k}\} \\
= & \exp\{C(k,r)-\frac{1}{2}(\frac{1}{1-\bar{\alpha}_{k-1}}+\frac{\alpha_k}{\beta_k})*{\mathbf{x}_{k-1}^r}^2+\mathbf{x}_{k-1}^r* \\
[\frac{(\sqrt{\bar{\alpha}_{k-1}}\mathbf{x}_0^r+\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k))}{1-\bar{\alpha}_{k-1}} & + \sqrt{\alpha_k}\frac{(\mathbf{x}_k-\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k)+\sqrt{\alpha_k}\gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1))}{\beta_k}]\} \\
= & \exp\{C(k,r)-\frac{1}{2}(\frac{1}{1-\bar{\alpha}_{k-1}}+\frac{\alpha_k}{\beta_k})*{\mathbf{x}_{k-1}^r}^2+\mathbf{x}_{k-1}* \\
[\frac{(\sqrt{\bar{\alpha}_{k-1}}}{1-\bar{\alpha}_{k-1}}\mathbf{x}_0^r+\frac{\sqrt{\alpha_k}}{\beta_k}(\mathbf{x}_k^r & -\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k))+(\frac{1}{1-\bar{\alpha}_{k-1}}+\frac{\alpha_k}{\beta_k})*\gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1)]\},
\end{aligned}
\tag{17}
$$

where $C(k,r)$ is a constant term with respect to $\mathbf{x}_{t-1}^r$. With some algebraic derivation, this can be simplified to:

$$
\begin{aligned}
q_\phi(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1}) = & \mathcal{N}(\frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\mathbf{x}_0^r+\frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}(\mathbf{x}_k^r-\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k)) \\
& + \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1),\frac{(1-\bar{\alpha}_{k-1})\beta_k}{1-\bar{\alpha}_k}\mathbf{I}).
\end{aligned}
\tag{18}
$$

A.3 Derivation of Training Objective

According to Eq. (1), $\mathbf{x}_0^r$ can be rewritten as:

$$
\mathbf{x}_0^r=\frac{1}{\sqrt{\bar{\alpha}_k}}(\mathbf{x}_k^r-\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k)-\sqrt{1-\bar{\alpha}_k}\epsilon),
\tag{19}
$$

where ϵ denotes gaussian noise. Then we can obatin:

$$
\epsilon=\frac{\mathbf{x}_k^r-\sqrt{\bar{\alpha}_k}\mathbf{x}_0^r}{\sqrt{1-\bar{\alpha}_k}}-\frac{\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k)}{\sqrt{1-\bar{\alpha}_k}}.
\tag{20}
$$

Since the second term is available, we employ a denoising network $f_\theta(\mathbf{x}_k^r,k,r)$ to predict the first term for training. Then we can obtain the predicted posterior distributions:

$$
\begin{aligned}
p_\theta(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^{r+1}) = & \mathcal{N}(\frac{1}{\sqrt{\alpha_k}}\big[\mathbf{x}_k^r-\frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}f_\theta(\mathbf{x}_k^r,k,r)\big]-\frac{\sqrt{\alpha_k(1-\bar{\alpha}_{k-1})}}{1-\bar{\alpha_k}}\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) \\
& + \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1),\frac{(1-\bar{\alpha}_{k-1})\beta_k}{1-\bar{\alpha}_k}\mathbf{I}),
\end{aligned}
\tag{21}
$$

Revisiting the objective of diffusion model (Song et al., 2020), we insteadly minimize the KL-divergence $D_{KL}(q_\phi(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1})\|p_\theta(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^{r+1}))$.

With Eq. (3) and Eq. (4), we can obtain the training objective:

$$
\begin{aligned}
\mathcal{L}_{\theta,\phi,k,r} = & D_{KL}(q_\phi(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^r,\mathbf{x}_0^{r+1})\|p_\theta(\mathbf{x}_{k-1}^r|\mathbf{x}_k^r,\mathbf{x}_0^{r+1})) \\
= & \sum_{k=1}^{K}\eta_k\mathbb{E}_{\mathbf{x}_0^r,\epsilon,r}\big[\|\epsilon_\theta(\mathbf{x}_k^r,k,r)-\frac{\mathbf{x}_k^r-\sqrt{\bar{\alpha}_k}\mathbf{x}_0^r}{\sqrt{1-\bar{\alpha}_k}}\|^2\big],
\end{aligned}
\tag{22}
$$

where $\mathbf{x}_k^r=\sqrt{\bar{\alpha}_k}\mathbf{x}_0^r+\gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k)+\sqrt{1-\bar{\alpha}_k}\epsilon$ and $\eta_k=\frac{\beta_k}{2\alpha_k(1-\bar{\alpha}_{k-1})}$ is a loss weight.

A.4 Derivation of Accelerated Sampling

Given $\mathbf{x}_k^r$, we can obtain $\mathbf{x}_{k-1}^r$ via:

$$\begin{aligned}\mathbf{x}_{k-1}^r =& \sqrt{\bar{\alpha}_{k-1}}\hat{\mathbf{x}}_0^r + \sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2}\cdot\frac{\mathbf{x}_k^r-\sqrt{\bar{\alpha}_k}\hat{\mathbf{x}}_0^r}{\sqrt{1-\bar{\alpha}_k}} \\ &- \gamma_k\mathcal{E}_\phi(\hat{\mathbf{x}}_0^r,\hat{\mathbf{x}}_0^{r+1},k)\cdot\frac{1-\bar{\alpha}_{k-1}-\sigma_k^2}{\sqrt{1-\bar{\alpha}_k}} + \gamma_{k-1}\mathcal{E}_\phi(\hat{\mathbf{x}}_0^r,\hat{\mathbf{x}}_0^{r+1},k-1).\end{aligned} \tag{23}$$

Proof 3 *We can prove this by induction. Assume that at time k, the posterior and marginal distributions admit the expected distributions, then we need to prove that at time $k-1$, $q_\phi(\mathbf{x}_{k-1}^r|\mathbf{x}_0^r,\mathbf{x}_0^{r+1})$ also has the expected distribution. We can rewrite the posterior and marginal distribution:*

$$\begin{aligned}\mathbf{x}_{k-1}^r =& \sqrt{\bar{\alpha}_{k-1}}\mathbf{x}_0^r + \sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2} * \frac{\mathbf{x}_k^r-\sqrt{\bar{\alpha}_k}\mathbf{x}_0^r}{\sqrt{1-\bar{\alpha}_k}} \\ &- \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) * \frac{\sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2}}{\sqrt{1-\bar{\alpha}_k}} + \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1) + \sigma_k\epsilon_1,\end{aligned} \tag{24}$$

$$\mathbf{x}_k^r = \sqrt{\bar{\alpha}_k}\mathbf{x}_0^r + \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) + \sqrt{1-\bar{\alpha}_k}\epsilon_2, \tag{25}$$

where ϵ_1, ϵ_2 are standard gaussian noises. Plugging in $\mathbf{x}_k^r$, we have:

$$\begin{aligned}\mathbf{x}_{k-1}^r =& \sqrt{\bar{\alpha}_{k-1}}\mathbf{x}_0^r \\ &+ \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) * \frac{\sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2}}{\sqrt{1-\bar{\alpha}_k}} - \gamma_k\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k) * \frac{\sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2}}{\sqrt{1-\bar{\alpha}_k}} \\ &+ \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1) + \sigma_k\epsilon_1 + \sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2}\epsilon_2 \\ =& \sqrt{\bar{\alpha}_{k-1}}\mathbf{x}_0^r + \gamma_{k-1}\mathcal{E}_\phi(\mathbf{x}_0^r,\mathbf{x}_0^{r+1},k-1) + \sigma_k\epsilon_1 + \sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2}\epsilon_2.\end{aligned} \tag{26}$$

Since the variance of $(\sigma_t\epsilon_1 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\epsilon_2)^2 = (1-\bar{\alpha}_{t-1})\boldsymbol{I}$, we have the expected sampling.

B Experimental Details

B.1 Datasets

We use the following five datasets for anomaly detection experiments:

- **SMD (Server Machine Dataset)** (Su et al., 2019): The SMD dataset is collected from a large internet company and includes 5 weeks of data from 28 server machines with 38 sensors each. The initial 5 days consist solely of normal data, while anomalies are intermittently introduced over the last 5 days.
- **PSM (Pooled Server Metrics)** (Abdulaal et al., 2021): The PSM dataset is collected internally from multiple application server nodes at eBay. It consists of 13 weeks of training data and 8 weeks of testing data.
- **MSL (Mars Science Laboratory)** (Hundman et al., 2018) and **SMAP (Soil Moisture Active Passive satellite)** (Entekhabi et al., 2010): The MSL and SMAP datasets are publicly available datasets collected by NASA. They contain telemetry anomaly data derived from the Incident Surprise Anomaly (ISA) reports of spacecraft monitoring systems. The MSL dataset has 55 dimensions, while the SMAP dataset has 25 dimensions. The training sets for both datasets include unlabeled anomalies.
- **SWaT (Secure Water Treatment)** (Mathur & Tippenhauer, 2016): The SWaT dataset is collected over 11 days from a scaled-down water treatment testbed with 51 sensors. For the first 7 days, only normal data were generated. During the last 4 days, 41 anomalies were injected using various attack methods.

Table 7: Datasets used for anomaly detection experiments.

Dataset	Entities	Dimensions	Train #	Test #	Anomaly Rate (%)
SMD	28	38	708405	708420	4.16
PSM	1	25	132481	87841	27.76
MSL	27	55	58317	73729	10.48
SMAP	55	25	140825	444035	12.83
SWaT	1	51	495000	449919	12.14

We present the statistics of these datasets in Table. 7. Train # and Test # denote the number of training and testing data, respectively. Anomaly Rate is the ratio between the sum of all anomaly points and sum of all test points.

We use the following three non-stationary datasets for forecasting and imputation experiments to assess the generalization ability of MODEM.

- **Traffic** (Tashiro et al., 2021): The Traffic dataset records the hourly road occupancy rates generated by sensors in the San Francisco Bay area freeways.
- **Exchange** (Shen & Kwok, 2023): The Exchange dataset describes the daily exchange rates of eight countries (Australia, British, Canada, Switzerland, China, Japan, New Zealand, and Singapore).
- **KDDCup** (Kollovieh et al., 2024): The KDDCup is a dataset of the air quality indices (AQIs) of Beijing and London used in the KDD Cup 2018.

For imputation performance evaluation, we examine three scenarios following (Kollovieh et al., 2024): (1) random missing, where values are missing sporadically, (2) blackout missing at the beginning of the context window, involving a sequence of consecutive missing values, and (3) blackout missing at the end of the context window. We report the average performance of three conditions.

B.2 Baselines

We introduce the following state-of-the-art time series anomaly detection methods for extensive comparisons:

- **Isolation Forest** (Li et al., 2022): constructs 3D features (text, reviewer behavior, deceptive ratings) and integrates feature selection to detect fake reviews.
- **LSTM-AD** (Malhotra et al., 2015): possesses long-term memory capabilities, and for the first time, hierarchical recurrent processing layers have been combined to detect anomalies in univariate time series without using labels for training.
- **MSCRED** (Zhang et al., 2019): designs an attention-based ConvLSTM network to capture temporal trends, and a convolutional autoencoder is used to encode and reconstruct the signature matrix instead of relying on the time series explicitly.
- **OmniAnomaly** (Su et al., 2019): is a Variational Autoencoder that performs anomaly detection by computing the reconstruction probability and quantifying interpretability based on the reconstruction probability of each feature.
- **InterFusion** (Li et al., 2021): uses a hierarchical variational autoencoder with two random latent variables to learn metrics and temporal representations and by relying on a ”reconstruction input” to compress the MTS.
- **GDN** (Deng & Hooi, 2021): utilizes the nodes and edges of the GNN to capture sensor features and spatial information, respectively. It then leverages this data to predict sensor behavior based on the attention function of adjacent sensors.
- **MST-GAT** (Ding et al., 2023): uses a multimodal graph attention network and a temporal convolutional network to capture spatiotemporal correlations in multimodal time series.
- **BeatGAN** (Zhou et al., 2019): uses a group of autoencoders and GANs in cases where tags are not available, which accurately detect anomalies in both ECG and sensor data.

- **MAD-GAN** (Li et al., 2019): uses LSTM-RNN as a generator and discriminator to capture temporal relationships in Gans, while using reconstruction and discriminant losses to detect anomalies.
- **Anomaly Transformer** (Xu et al., 2022): captures association differences by modeling prior and sequential associations for each timestamp, making rare exceptions easier to distinguish.
- **TFAD** (Zhang et al., 2022): introduces time series decomposition and data augmentation mechanisms into the designed time-frequency architecture, enhancing both performance and interpretability.
- **TranAD** (Tuli et al., 2022): adopts a two-step reconstruction method, introduces the attention mechanism into Transformer model, and integrates adversarial training.
- **NPSR** (Lai et al., 2024): proposes a framework for unsupervised time series anomaly detection using point-based and sequence-based reconstruction models.
- **DiffAD** (Xiao et al., 2023): designs a novel denoising diffusion-based imputation method to improve the imputation performance of missing values with conditional weight-incremental diffusion.
- **ImDiffusion** (Chen et al., 2023): combines time series imputation and diffusion models to achieve accurate and robust anomaly detection.
- **D^{3}R** (Wang et al., 2024): tackles the drift via decomposition and reconstruction, overcoming the limitation of the local sliding window.

We introduce the following state-of-the-art time series forecasting and imputation methods for comparison to assess the generalization ability of MODEM:

- **CSDI** (Tashiro et al., 2021): is a self-supervised method that uses the observed value as conditional information to imputation the masked time series.
- **TSDiff** (Kollovieh et al., 2024): is an unconditionally trained diffusion model for time series and a mechanism to condition TSDiff during inference for arbitrary forecasting tasks (observation self-guidance).

To ensure fair comparisons, we first reference the best-reported values from the original papers or other publications. If these values are unavailable, we report the results reproduced on our machine using the publicly available codes. Moreover, we assume that the thresholds of these SOTA methods have been optimized for optimal performance.

B.3 Evaluation Metrics

We use the following metrics for evaluating the performance of time series anomaly detection methods:

- **Precision**: It measures the proportion of correctly detected anomalies among all time points flagged as anomalies:
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{27}$$
where TP denotes the number of true anomalies correctly detected, and FP denotes the number of normal time points incorrectly identified as anomalies.
- **Recall**: This metric is also named as sensitivity or true positive rate. It measures the proportion of actual anomalies that are correctly detected by the algorithm:
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{28}$$
where FN denotes the number of true anomalies that are not detected.
- **F1-score**: F1-score is the harmonic mean of **Precision** and **Recall**, which can balance both false positives and false negatives:
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{29}$$

- **Average Sequence Detection Delay (ADD)** (Tuli et al., 2022): It is used for evaluating the speed and timeliness of time series anomaly detection algorithm:

$$\text{ADD} = \frac{1}{S}\sum_{i=1}^{S}(\mathcal{T}_i - \rho_i), \tag{30}$$

where ρ_i denotes the initial time of anomalous span i, $\mathcal{T}_i \geq \rho_i$ denotes the corresponding detection delay time by the anomaly detection algorithm. S is the total number of anomalous spans. A small ADD signifies a more timely detection of anomalies.

B.4 Network Structure

The detailed architecture of Time-Invariant Encoder is presented in Figure. 6, it comprises multiple residual blocks. The residual block first uses a 1×1 convolution kernel to encode the time-invariant components obtained from frequency decomposition. The diffusion step k and resolution scale r are fed into simple MLPs to obtain corresponding embeddings, which are added to the convolution result as supplementary information. Subsequently, it employs hierarchical transformers to further explore the intra-series temporal features and inter-series dependencies among various variables.

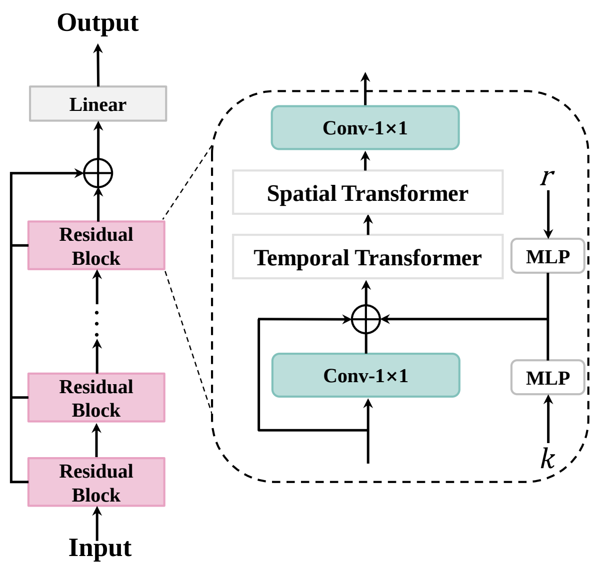

Figure 6: Architecture of Time-Invariant Encoder, consisting of multiple stacked Residual Block.

B.5 Hyperparameter Settings

The detailed hyperparameter settings of our MODEM are presented in Tab. 8. Due to the large number of hyperparameters and resource constraints, we use empirical tuning combined with Bayesian selection to determine the parameter combination. The diffusion step K is set to 50, while the resolution scale R is 4, indicating the number of steps for the forward process. The sampling step L is set to 20 for the denoising process. 20% of the frequency components are used for decomposition, represented by m. Our model incorporates 2 residual blocks and 4 DMTBs. Additionally, a dilation factor of 2 is applied, and our model is equipped with 5 ModernTCN modules, which are temporal convolutional networks that help handle non-stationary time series. These hyperparameter values collectively define our model's structure and its ability to process non-stationarity.

B.6 Additional Effectiveness Analysis

Multi-Resolution Modeling We present several case studies in Fig. 7, which quantitatively validate the effectiveness of multi-resolution modeling for anomaly detection. In the figure, the pink and yellow areas represent true anomalies and false anomalies (normal data), respectively. The green

Table 8: Detailed hyperparameter settings of MODEM.

Hyperparameter	Value
Diffusion step K	50
Sampling step L	20
Resolution scale R	4
Percentage of frequency decomposition m	20
Number of residual blocks	2
Number of DMTBs	4
Dilated Factor	2
Number of ModernTCN modules	5

and purple lines correspond to the input and reconstructed time series, respectively, with the red bold frame highlighting anomalies detected by the model. These examples demonstrate that models operating at a single resolution struggle to detect true anomalies and frequently trigger false alarms. Benefiting from an expansion to multi-resolution settings, our model learns normal temporal patterns more effectively, providing a more reliable reconstructed baseline that significantly reduces false alarms while accurately detecting true anomalies.

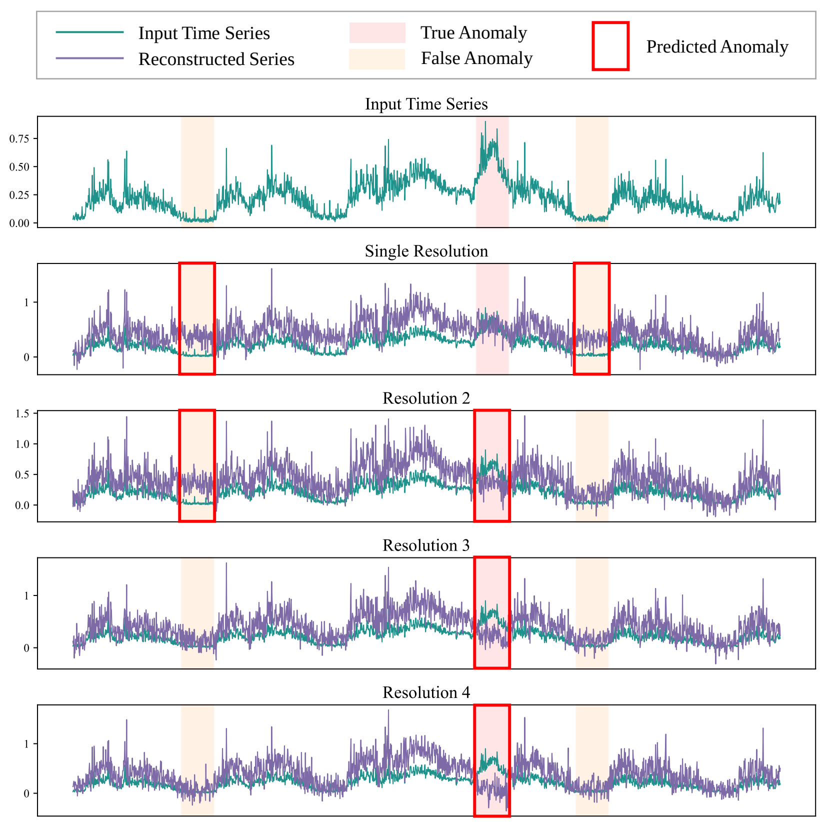

Figure 7: Comparison of detection performance across different resolution settings originates from the SMD dataset, where the pink and yellow areas represent true anomalies and false anomalies (normal data), respectively. The green and purple lines indicate the input time series and the reconstructed time series, respectively, while the red bold frame highlights anomalies detected by the model.

Frequency-Enhanced Decomposable Network Previous works (Yang & Hong, 2022; He et al., 2023) have demonstrated that spectral responses can more robustly capture underlying temporal patterns compared to time-domain representations. Despite this foundation, our approach introduces several distinct contributions and innovations for non-stationary time series anomaly detection.

Firstly, we explore different frequency computation strategies and utilize the Short-Term Fourier Transform (STFT) to extract the spectral information. STFT not only provides a detailed view of how the frequency content of the signal changes over time but also captures the transient behavior and dynamics of time series through overlapping windows. Compared to the conventional Fast Fourier Transform (FFT), STFT is more suitable for decomposing non-stationary time series, which is quantitatively verified through experiments on PSM and SMD datasets, as shown in Tab. 9.

Secondly, the proposed Frequency-Enhanced Decomposable Network differs significantly in its network structure from the MLP-based block like Koopa (Liu et al., 2024). Specifically, it incorporates hierarchical transformers in the time-variant and time-invariant encoders to capture both intra-series temporal dependencies and inter-series correlations among variables. Additionally, it designs a novel ModernTCN block enhanced by dilated convolution (named DMTBs) to uncover complex periodic patterns across multiple time scales. These unique designs, compared to Koopa's all-MLP architecture, are more suitable for long-series reconstruction, thus providing reliable reconstruction signals for anomaly detection, as demonstrated by the quantitative validations in Tab. 10.

Table 9: Comparison of different frequency statistics approaches on PSM and SMD datasets.

Dataset	Approach	P	R	F1
PSM	FFT	96.64	97.98	97.30
	STFT	96.97	98.35	97.65
SMD	FFT	95.46	96.07	95.76
	STFT	95.70	96.32	96.01

Table 10: Comparison of different network types on PSM and SMD datasets.

Dataset	Network	P	R	F1
PSM	Koopa (MLP)	95.68	97.53	96.59
	MODEM	96.97	98.35	97.65
SMD	Koopa (MLP)	94.42	95.43	94.92
	MODEM	95.70	96.32	96.01

In our method, we select frequency components with the top m percent of amplitudes as stationary factors based on spectral statistics, with the remaining frequency components treated as non-stationary factors. As mentioned in Appendix B.5, we set the percentage m to 20, because the frequencies corresponding to the top 20% of amplitudes account for over 90% of all frequency components, which aligns with the reality where stationary factors are dominant. To investigate the impact of percentage m on the detection performance further, we conduct a sensitivity analysis on the SMD dataset regarding m, as illustrated in Tab. B.6. Variations within a reasonable range of m do not cause drastic changes in performance, demonstrating that this frequency-based selection approach is robust. If m is set too low (see $m = 2$) or too high (see $m = 50$), the model's performance significantly decreases because the time-variant and time-invariant variables are not effectively separated.

Table 11: Impact of different percentages m on detection performance.

Percentage m	Frequency Ratio	P	R	F1
2	66.89%	95.34	95.67	95.50
10	85.33%	95.68	96.10	96.88
15	88.62%	95.74	96.21	95.97
20	91.45%	95.70	96.32	96.01
25	93.56%	95.66	96.25	95.95
50	98.73%	95.40	95.76	95.57

C LIMITATIONS AND FUTURE WORK

We extend the diffusion model into a multi-resolution paradigm, leading directly to an R-fold increase in the number of sampling steps, which results in longer training convergence times and prolonged inference durations. While we introduce accelerated sampling strategies for each resolution R, these are predicated on the acceptance of loss in precision. In the future, it may prove beneficial to investigate sampling along a deterministic trajectory within mixed resolutions to improve both accuracy and efficiency. Furthermore, the hyperparameters R and m in our proposed frequency-enhanced decomposition network require customization for different datasets, thereby adding to the complexity of parameter tuning. Additionally, in the future, it may be advantageous to consider employing dynamic smoothing strategies instead of fixed pooling sizes to better reveal the periodic and varying characteristics of non-stationary time series.

D Potential Negative Societal Impacts

The diffusion models, like other generative technologies, have inherent risks. Our model, as a case in point, could potentially have negative societal effects. For instance, it might memorize private data and be exploited to fabricate misleading or false information.