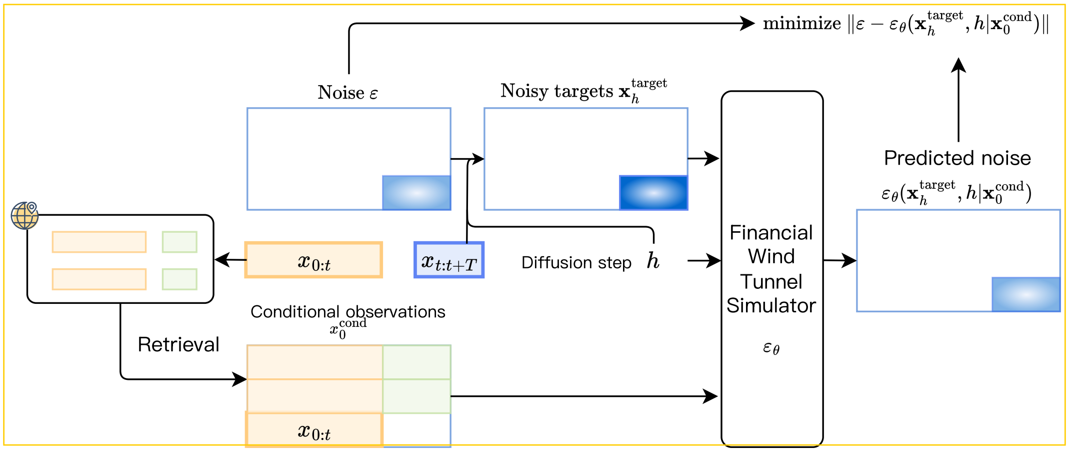
minimize $\|\varepsilon - \varepsilon_\theta(\mathbf{x}_h^{\text{target}}, h|\mathbf{x}_0^{\text{cond}})\|$
Noise ε
Noisy targets $\mathbf{x}_h^{\text{target}}$
Predicted noise
$\varepsilon_\theta(\mathbf{x}_h^{\text{target}}, h|\mathbf{x}_0^{\text{cond}})$
$x_{0:t}$
$x_{t:t+T}$
Diffusion step h
Financial Wind Tunnel Simulator
ε_θ
Conditional observations
x_0^{cond}
Retrieval
$x_{0:t}$

Figure 2: The self-supervised training procedure of FWT. FWT synthesizes $x_{t:t+T}$ when given a history time series $x_{0:t}$, which is used to query top-k most relevant stocks to construct conditional observations. FWT predicts noise from noisy target $\mathbf{x}_h^{\text{target}}$ at the diffusion step h to recover $x_{t:t+T}$.

the same market or across different markets. By concatenation, the value of the observed multivariate time series is:

$$\mathbf{X} = \{r_{0:K+1,0:t+T}\} \in \mathbb{R}^{(k+1)\times(t+T)} \tag{3}$$

We also denote an observation mask as $\mathbf{M} = \{m_{0:K+1,0:t+T}\} \in \{0,1\}^{(k+1)\times(t+T)}$, where $m_{0,t:t+T} = 0$ and $m_{\text{else}} = 1$. The mask filters the future part Y_{S_0} of stock S_0. Our goal is to synthesize a time series Y'_{S_0} to approximate the real Y_{S_0}. To achieve the goal, we utilize the retrieval results as condition and deploy a diffusion method as is described in the next section3.2.

We emphasize that the retrieval process can run on all frequencies of stock dynamics. This setting is named Multi-frequency Generation, whose result will be shown in Section 4.2. Besides, we can apply retrieval filtering rules to identify different forms of conditions to control the generation. in order to retrieve-generate data from various sources of time-series for downstream tasks. In this paper, we explore two more cases: 1) Cross-market generation, which retrieves from other markets; and 2) What-if generation, which recalls from the provided time series database. The results will be presented in Section 4.2.

3.2 Generation

We utilize a conditional diffusion model to generate the future price trajectory of the target stock S_0. To begin with, in conventional DDPM [13], given data $\mathbf{x}_0 \sim p(\mathbf{x}_0)$, the forward process is a Markov-Gaussian process that gradually adds noise to obtain a perturbed sequence $\{x_1, x_2, \cdots, x_H\}$,

$$q(\mathbf{x}_1, \cdots, \mathbf{x}_H \mid \mathbf{x}_0) = \prod_{h=1}^{H} q(\mathbf{x}_h \mid \mathbf{x}_{h-1}), \tag{4}$$

where $q(\mathbf{x}_h \mid \mathbf{x}_{h-1}) = \mathcal{N}(\mathbf{x}_h; \sqrt{1-\beta_h}\mathbf{x}_{h-1}, \beta_h\mathbf{I})$, $\mathbf{I}$ is the identity matrix, β_h is a learnable parameter, H is the number of diffusion steps, q represents the forward process, and $\mathcal{N}$ denotes the Gaussian distribution parameterized by hyperparameters $\{\beta_h\}_{h\in[H]}$. Perturbed samples are sampled via $x_h = \sqrt{\bar{\alpha}_h}x_0 + \sqrt{1-\bar{\alpha}_h}\epsilon$, where $\epsilon \sim \mathcal{N}(0,\mathbf{I})$, $\alpha_h = 1 - \beta_h$ and $\bar{\alpha_h} = \prod_{s=1}^{h}\alpha_s$. For the reverse process, the model learns to predict noise ϵ from the noisy target $\mathbf{x}_h$ at the diffusion step h to recover the original data $\mathbf{x}_0$. It introduces a specific parameterization of $p_\theta(\mathbf{x}_{h-1} \mid \mathbf{x}_h)$:

$$\begin{aligned} \mu^{\text{DDPM}}(\mathbf{x}_h, h) &= \frac{1}{\alpha_h}(\mathbf{x}_h - \frac{\beta_h}{\sqrt{1-\alpha_h}}\epsilon_\theta(\mathbf{x}_h, h)) \\ \sigma^{\text{DDPM}}(\mathbf{x}_h, h) &= \sqrt{\bar{\beta}_h} \end{aligned} \tag{5}$$

The training loss [13] for diffusion models is to predict the normalized noise ϵ by solving the following optimization problem:

$$\min_\theta \mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0),\epsilon\sim\mathcal{N}(0,\mathbf{I}),h}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_h, h)\|_2^2\right], \tag{6}$$

The denoising function ϵ_θ estimates the noise ϵ that was added to the noisy target $\mathbf{x}_h$ at the diffusion step h.

Then, we consider extending the parameterization of DDPM to the conditional case. We define a conditional denoising function $\epsilon_\theta : (\mathcal{X}^{\text{target}} \times \mathbb{R} \mid \mathcal{X}^{\text{cond}}) \to \mathcal{X}^{\text{target}}$, which takes conditional observations $\mathbf{x}_0^{\text{cond}}$ as inputs. We consider the following parameterization with ϵ_θ:

$$\begin{aligned} \mu_\theta(\mathbf{x}_h^{\text{target}}, h \mid \mathbf{x}_0^{\text{cond}}) &= \mu^{\text{DDPM}}(\mathbf{x}_h^{\text{target}}, h, \epsilon_\theta(\mathbf{x}_h^{\text{target}}, h \mid \mathbf{x}_0^{\text{cond}})) \\ \sigma_\theta(\mathbf{x}_h^{\text{target}}, h \mid \mathbf{x}_0^{\text{cond}}) &= \sigma^{\text{DDPM}}(\mathbf{x}_h^{\text{target}}, h) \end{aligned} \tag{7}$$

For the sampling, we set all observed values $\mathbf{x}_0$ as conditional observations $\mathbf{x}_0^{\text{cond}}$ and all missing values as the target $\mathbf{x}_0^{\text{target}}$ to be generated. We sample noisy targets $\mathbf{x}_h^{\text{target}} = \sqrt{\bar{\alpha}_h}\mathbf{x}_0^{\text{target}} + \sqrt{1-\bar{\alpha}_h}\epsilon$ and train ϵ_θ by minimizing the loss:

$$\min_\theta \mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0),\epsilon\sim\mathcal{N}(0,\mathbf{I}),h}\left[\left\|\epsilon - \epsilon_\theta(\mathbf{x}_h^{\text{target}}, h \mid \mathbf{x}_0^{\text{cond}})\right\|_2^2\right]. \tag{8}$$

As is illustrated in Figure 2, given a sample $\mathbf{x}_0$, we separate it into conditional observations $\mathbf{x}_0^{\text{cond}}$ and the target $\mathbf{x}_0^{\text{target}}$. In market synthesization, the target $\mathbf{x}_0^{\text{target}} = \mathbf{M} * \mathbf{X}$ is the future price trajectory of the target stock S_0 over the interval $[t, t+T)$, and