

Appendix

TABLE OF CONTENTS

A	Derivations of the Stationary Merton Jump Diffusion Model	17
A.1	MJD and Lévy Process	17
A.2	Explicit Solution to MJD	17
A.3	Likelihood Function of MJD	18
B	Derivations of the Non-stationary Merton Jump Diffusion Model	19
B.1	Non-stationary MJD and Additive Process	19
B.2	Explicit Solution to Non-stationary MJD	19
B.3	Likelihood Function of Non-stationary MJD	20
C	Proofs of Theorem and Proposition	22
C.1	Proof of Theorem 4.1	22
C.2	Proof of Proposition 4.2	24
D	Experiment Details	25
D.1	Baseline, Model Architecture, and Experiment Settings	25
D.2	Datasets Details	25
D.3	Additional Deterministic Time-Series Baselines (Third-Party Implementations) . .	27
D.4	Limitations	27
D.5	Vanilla Euler Solver	27
E	Impact Statement	28

A Derivations of the Stationary Merton Jump Diffusion Model

In this section, we briefly review the mathematical derivations from classical textbooks to ensure the paper is self-contained. Our primary focus is on the case where the state variable S is scalar, as is common in many studies. However, in Sec. 4, we extend our analysis to the more general \mathbb{R}^d setting. Notably, in our framework, we do not account for correlations among higher-dimensional variables. For instance, the covariance matrix of the Brownian motion is assumed to be isotropic, meaning all components have the same variance. To maintain clarity and consistency with standard textbook conventions, we adopt scalar notations throughout this section for simplicity.

A.1 MJD and Lévy Process

Definition A.1. Lévy process [65, Definition 3.1] A càdlàg (right-continuous with left limits) stochastic process $(X_t)_{t \geq 0}$ on $(\Omega, \mathcal{F}, \mathbb{P})$ with values in \mathbb{R}^d such that $X_0 = 0$ is called a Lévy process if it possesses the following properties:

1. Independent increments: For every increasing sequence of times t_0, t_1, \dots, t_n , the random variables $X_{t_0}, X_{t_1} - X_{t_0}, \dots, X_{t_n} - X_{t_{n-1}}$ are independent.
2. Stationary increments: The law of $X_{t+h} - X_t$ does not depend on t .
3. Stochastic continuity: For all $\varepsilon > 0$, $\lim_{h \rightarrow 0} \mathbb{P}(|X_{t+h} - X_t| \geq \varepsilon) = 0$.

A Lévy process $(X_t)_{t \geq 0}$ is a stochastic process that generalizes jump-diffusion dynamics, incorporating both continuous Brownian motion and discontinuous jumps. The Merton Jump Diffusion (MJD) model given by,

$$dS_t = S_t((\mu - \lambda k)dt + \sigma dW_t + dQ_t), \quad (15)$$

is a specific example of a Lévy process, as it comprises both a continuous diffusion component and a jump component. According to the Lévy–Itô decomposition [65, Proposition 3.7], any Lévy process can be expressed as the sum of a deterministic drift term, a Brownian motion component, and a pure jump process, which is represented as a stochastic integral with respect to a Poisson random measure.

A.2 Explicit Solution to MJD

To derive the solution to MJD in Eq. (2), based on [65, Proposition 8.14], we first apply Itô's formula to the SDE:

$$\begin{aligned} df(S_t, t) &= \frac{\partial f(S_t, t)}{\partial t} dt + b_t \frac{\partial f(S_t, t)}{\partial S_t} dt + \frac{\omega_t^2}{2} \frac{\partial^2 f(S_t, t)}{\partial S_t^2} dt + \omega_t \frac{\partial f(S_t, t)}{\partial S_t} dW_t \\ &\quad + [f(S_t) - f(S_{t-})], \end{aligned} \quad (16)$$

where $b_t = (\mu - \lambda k)S_t$, $\omega_t = \sigma S_t$, and S_{t-} represents the value of S before the jump at time t .

By setting the function $f(S_t, t) = \ln S_t$, the formula can be rearranged as:

$$\begin{aligned} d \ln S_t &= \frac{\partial \ln S_t}{\partial t} dt + (\mu - \lambda k)S_t \frac{\partial \ln S_t}{\partial S_t} dt + \frac{\sigma^2 S_t^2}{2} \frac{\partial^2 \ln S_t}{\partial S_t^2} dt + \sigma S_t \frac{\partial \ln S_t}{\partial S_t} dW_t \\ &\quad + [\ln(S_t) - \ln(S_{t-})] \\ &= (\mu - \lambda k)S_t \frac{1}{S_t} dt + \frac{\sigma^2 S_t^2}{2} \left(-\frac{1}{S_t^2} \right) dt + \sigma S_t \left(\frac{1}{S_t} \right) dW_t + [\ln(S_t) - \ln(S_{t-})] \\ &= (\mu - \lambda k)dt - \frac{\sigma^2}{2} dt + \sigma dW_t + [\ln(S_t) - \ln(S_{t-})] \end{aligned} \quad (17)$$

From the definition of the Compound Poisson process, we have that $S_t = Y_i S_{t-}$, such that $\ln(S_t) - \ln(S_{t-}) = \ln Y_i$. Here, Y_i is the magnitude of the multiplicative jump. Therefore, integrating both sides of Eq. (17), we get the final explicit solution for MJD model:

$$\ln S_t - \ln S_0 = (\mu - \lambda k - \frac{1}{2}\sigma^2)t + \sigma W_t + \sum_{i=1}^{N_t} \ln Y_i. \quad (18)$$

We can reorganize the explicit solution as:

$$S_t = S_0 \exp \left(\left(\mu - \lambda k - \frac{\sigma^2}{2} \right) t + \sigma W_t + \sum_{i=1}^{N_t} \ln Y_i \right), \quad (19)$$

since the drift term, diffusion term and jump term are independent, we can derive the mean of S_t conditional on S_0 :

$$\begin{aligned} \mathbb{E}[S_t|S_0] &= S_0 \mathbb{E} \left[\exp \left(\left(\mu - \lambda k - \frac{\sigma^2}{2} \right) t + \sigma W_t + \sum_{i=1}^{N_t} \ln Y_i \right) \right] \\ &= S_0 \mathbb{E} \left[\exp \left(\left(\mu - \lambda k - \frac{\sigma^2}{2} \right) t \right) \right] \cdot \mathbb{E} [\exp (\sigma W_t)] \cdot \mathbb{E} \left[\exp \left(\sum_{i=1}^{N_t} \ln Y_i \right) \right] \\ &= S_0 \exp \left(\left(\mu - \lambda k - \frac{\sigma^2}{2} \right) t \right) \cdot \exp \left(\frac{\sigma^2}{2} t \right) \cdot \exp (\lambda k t) \\ &= S_0 \exp (\mu t) \end{aligned} \quad (20)$$

A.3 Likelihood Function of MJD

For the log-likelihood derivation, given the conditional probability in Eq. (4), the log-likelihood of the MJD model can be expressed as:

$$\begin{aligned} \log P(\ln S_t | S_0) &= \log \sum_{n=0}^{\infty} P(N_t = n) P(\ln S_t | S_0, N_t = n) \\ &= \log \sum_{n=0}^{\infty} \exp(-\lambda t) \frac{(\lambda t)^n}{n!} \frac{1}{\sqrt{2\pi b_n^2}} \exp \left(-\frac{(\ln S_t - a_n)^2}{2b_n^2} \right) \\ &= \log \sum_{n=0}^{\infty} \exp \left(-\lambda t + \log \frac{(\lambda t)^n}{n!} + \log \frac{1}{\sqrt{2\pi b_n^2}} + \left(-\frac{(\ln S_t - a_n)^2}{2b_n^2} \right) \right) \\ &= \log \sum_{n=0}^{\infty} \exp \left(-\lambda t + n \log(\lambda t) - \log n! \sqrt{2\pi} - \frac{\log b_n^2}{2} - \frac{(\ln S_t - a_n)^2}{2b_n^2} \right), \end{aligned} \quad (21)$$

where $a_n = \ln S_0 + \left(\mu - \lambda k - \frac{\sigma^2}{2} \right) t + n\nu$ and $b_n^2 = \sigma^2 t + \gamma^2 n$. In maximum likelihood estimation (MLE), the initial asset price S_0 is assumed to be constant (non-learnable) and can therefore be excluded from optimization. The objective of MLE is to estimate the parameter set $\Theta = \{\mu, \sigma, \lambda, \gamma, \nu\}$ by maximizing the likelihood of the observed data under the estimated parameters. For the MJD model, the MLE objective is to determine the optimal parameters $\hat{\Theta}$. By omitting constant terms and expanding s_n and a_n , the final expression of the MLE objective can be simplified as:

$$\begin{aligned} \hat{\Theta} &= \arg \max_{\Theta} \log P(\ln S_t | S_0) \\ &= \arg \max_{\Theta} \log \sum_{n=0}^{\infty} \exp \left(-\lambda t + n \log(\lambda t) - \frac{\log(\sigma^2 t + n\gamma^2)}{2} \right. \\ &\quad \left. - \frac{(\ln S_t - \ln S_0 - \left(\mu - \frac{\sigma^2}{2} - \lambda k \right) t - n\nu)^2}{2(\sigma^2 t + n\gamma^2)} \right) \end{aligned} \quad (22)$$

According to [90], the Fourier transform can be applied to the Merton Jump Diffusion log-return density function. The characteristic function is then given by:

$$\begin{aligned} \phi_c(\omega) &= \int_{-\infty}^{\infty} \exp(i\omega x) P(x) dx \\ &= \exp \left[\lambda t \left\{ \exp \left(i\omega \nu + \frac{\gamma^2 \omega^2}{2} \right) - 1 \right\} + i\omega \left(\left(\mu - \frac{\sigma^2}{2} - \lambda k \right) t \right) - \frac{\sigma^2 \omega^2}{2} t \right], \end{aligned} \quad (23)$$

where $x = \ln \frac{S_t}{S_0}$.

With simplification $\phi_c(\omega) = \exp[tg(\omega)]$, we can find the characteristic exponent, namely, the cumulant generating function (CGF):

$$g(\omega) = \lambda \left\{ \exp \left(i\omega\nu + \frac{\gamma^2\omega^2}{2} \right) - 1 \right\} + i\omega \left(\mu - \frac{\sigma^2}{2} - \lambda k \right) - \frac{\sigma^2\omega^2}{2}, \quad (24)$$

where $k = \exp(\nu + \gamma^2/2) - 1$.

The series expansion of CFG is:

$$g(\omega) = i\omega k_1 - \frac{\omega^2 k_2}{2} + \frac{\omega^3 k_3}{6} \dots \quad (25)$$

According to [65, Proposition 3.13], the cumulants of the Lévy distribution increase linearly with t . Therefore, the first cumulant k_1 is the mean of the standard MJD:

$$k_1(t) = \mathbb{E} \left[\ln \frac{S_t}{S_0} \right] = (\mu - \lambda k - \sigma^2/2 + \lambda\nu)t \quad (26)$$

The second cumulant k_2 is variance of the standard MJD, which is:

$$k_2(t) = \text{Var} \left[\ln \frac{S_t}{S_0} \right] = (\sigma^2 + \lambda(\gamma^2 + \nu^2))t \quad (27)$$

The corresponding higher moments can also be calculated as:

$$\text{Skewness} = k_3(t) = \lambda(3\gamma^2\nu + \nu^3)t \quad (28)$$

$$\text{Excess Kurtosis} = k_4(t) = \lambda(3\gamma^4 + 6\nu^2\gamma^2 + \nu^4)t \quad (29)$$

B Derivations of the Non-stationary Merton Jump Diffusion Model

B.1 Non-stationary MJD and Additive Process

Definition B.1. Additive process [65, Definition 14.1] A stochastic process $(X_t)_{t \geq 0}$ on \mathbb{R}^d is called an additive process if it is càdlàg, satisfies $X_0 = 0$, and has the following properties:

1. Independent increments: For every increasing sequence of times t_0, t_1, \dots, t_n , the random variables $X_{t_0}, X_{t_1} - X_{t_0}, \dots, X_{t_n} - X_{t_{n-1}}$ are independent.
2. Stochastic continuity: For all $\varepsilon > 0$, $\lim_{h \rightarrow 0} \mathbb{P}(|X_{t+h} - X_t| \geq \varepsilon) = 0$.

In the non-stationary MJD model, given by,

$$dS_t = S_t((\mu_t - \lambda_t k_t)dt + \sigma_t dW_t + \int_{\mathbb{R}^d} (y - 1)N(dt, dy)), \quad (30)$$

the parameters governing drift, volatility, and jump intensity evolve over time, resulting in non-stationary increments. This violates the key stationarity property required for Lévy processes, as discussed in App. A. Consequently, the non-stationary MJD no longer falls within the Lévy process framework. Instead, according to the definition above, a stochastic process with independent increments that follow a non-stationary distribution is classified as an additive process. Similar to the relationship between the stationary MJD and the Lévy process, the non-stationary MJD can be viewed as a specific instance of an additive process. Thus, we can apply corresponding mathematical tools for additive processes to study the non-stationary MJD.

B.2 Explicit Solution to Non-stationary MJD

To derive the explicit solution to the non-stationary MJD, according to [65, Proposition 8.19], we have the Itô formula for semi-martingales:

$$\begin{aligned} f(t, X_t) - f(0, X_0) &= \int_0^t \frac{\partial f(s, X_s)}{\partial s} ds + \int_0^t \frac{\partial f(s, X_{s-})}{\partial x} dX_s + \frac{1}{2} \int_0^t \frac{\partial^2 f(s, X_{s-})}{\partial x^2} d[X, X]_s^c \\ &\quad + \sum_{0 \leq s \leq t, \Delta X_s \neq 0} [f(s, X_s) - f(s, X_{s-}) - \Delta X_s \frac{\partial f(s, X_{s-})}{\partial x}]. \end{aligned} \quad (31)$$

According to [65, Remark 8.3], for a function independent of time (*i.e.*, $f(t, X_t) = f(X_t)$), when we have finite number of jumps, we can rewrite the above equation as:

$$f(X_t) - f(X_0) = \int_0^t f'(X_{s-}) dX_s^c + \frac{1}{2} \int_0^t f''(X_{s-}) d[X, X]_s^c + \sum_{0 \leq s \leq t, \Delta X_s \neq 0} [f(X_s) - f(X_{s-})],$$

where X_s^c is the continuous part of X_s , and $[X, X]_s^c$ is the continuous quadratic variation of X over the interval $[0, s]$.

In our case, let $X_t = S_t$, and define $f(t, X_t) = \ln S_t$, the corresponding derivatives are $\frac{\partial f(t, X_t)}{\partial X_t} = \frac{\partial \ln S_t}{\partial S_t} = \frac{1}{S_t}$, and $\frac{\partial^2 f(t, X_t)}{\partial X_t^2} = \frac{1}{-S_t^2}$. The dynamics of non-stationary MJD is defined by:

$$dS_t = S_t \left(\mu_t dt - \lambda_t k_t dt + \sigma_t dW_t + \int_{\mathbb{R}^d} (y - 1) N(dt, dy) \right).$$

The continuous part of the quadratic variation of S_t is $d[S, S]_s^c = S_s^2 \sigma_s^2 ds$. A jump at time s is modeled as a multiplicative change $S_s = y S_{s-}$. Thus, the jump contribution is $\sum_{0 \leq s \leq t, \Delta X_s \neq 0} [f(X_s) - f(X_{s-})] = \sum_{0 \leq s \leq t, \Delta X_s \neq 0} [\ln(y S_{s-}) - \ln(S_{s-})] = \sum_{0 \leq s \leq t, \Delta X_s \neq 0} [\ln y]$. Since the jump process is driven by a Poisson random measure $N(dt, dy)$ on $[0, t] \times \mathbb{R}^d$, we can rewrite the sum over all jump times as an integral with respect to this measure. When there are finitely many jumps on $[0, t]$, we have $\sum_{0 \leq s \leq t, \Delta X_s \neq 0} [\ln y] = \int_0^t \int_{\mathbb{R}^d} \ln y N(ds, dy)$.

Based on [65, Ch. 14], even when the parameters (drift, volatility, jump intensity, etc.) are time-dependent, the non-stationary MJD remains a semi-martingale. Therefore, we can simplify the equation as follows for time t :

$$\begin{aligned} \ln S_t - \ln S_0 &= \int_0^t \frac{1}{S_s} S_s (\mu_s - \lambda_s k_s) ds - \frac{1}{2} \int_0^t \frac{1}{S_s^2} S_s^2 \sigma_s^2 ds + \int_0^t \sigma_s dW_s \\ &\quad + \int_{[0, t] \times \mathbb{R}^d} \ln y N(ds, dy) \end{aligned} \quad (32)$$

Therefore, the explicit solution is:

$$\ln \frac{S_t}{S_0} = \int_0^t (\mu_s - \lambda_s k_s - \frac{\sigma_s^2}{2}) ds + \int_0^t \sigma_s dW_s + \int_0^t \int_{\mathbb{R}^d} \ln y N(ds, dy). \quad (33)$$

The only assumption needed for the derivation is the finite variation condition: $\int_0^t \int_{\mathbb{R}^d} |y| N(ds, dy) < \infty$. Based on the explicit solution for S_t , we can easily compute the conditional expectations as,

$$\mathbb{E}[\ln(S_t/S_0)] = \int_0^t (\mu_s - \lambda_s k_s - \frac{\sigma_s^2}{2}) ds + \int_0^t \int_{\mathbb{R}^d} \ln y \lambda_s f_Y(s, y) dy ds. \quad (34)$$

and

$$\mathbb{E}[S_t | S_0] = S_0 \exp\left(\int_0^t \mu_s ds\right), \quad (35)$$

The variance can also be calculated as,

$$\text{Var}[\ln(S_t/S_0)] = \int_0^t \sigma_s^2 ds + \int_0^t \int_{\mathbb{R}^d} (\ln y)^2 \lambda_s f_Y(s, y) dy ds. \quad (36)$$

Given the results for the general time-inhomogeneous system, one can directly substitute the coefficients into the discrete formulation implemented in Sec. 4.2 to obtain the corresponding results.

B.3 Likelihood Function of Non-stationary MJD

Let $X_t = \ln S_t / \ln S_0$, $t \geq 0$ be the log-return of the asset price S_t . Under the non-stationary MJD settings, X_t is an additive process, therefore by the general property of additive process [65, Ch 14], the law of X_t is infinitely divisible and its characteristic function is given by the Lévy–Khintchine formula:

$$\mathbb{E}[\exp(iu \cdot X_t)] = \exp \psi_t(u),$$

where

$$\psi_t(u) = -\frac{1}{2}u \cdot A_t u + iu \cdot \Gamma_t + \int_{\mathbb{R}^d} \eta(dy) (e^{iu \cdot x} - 1 - iu \cdot x), \quad (37)$$

where we have the integrated volatility term $A_t = \int_0^t \sigma_s ds$, the integrated drift term $\Gamma_t = \int_0^t (\mu_s - \lambda_s k_s) ds$, and the Lévy measure $\eta(dy) = \lambda_t f_Y(t, y)$.

Since the jumps follow a time-inhomogeneous Poisson random measure and the process is additive, we can denote the integrated intensity of jumps by $\Lambda(t) = \int_0^t \lambda_s ds$, then the number of jumps N_t in the time range $[0, t]$ is a Poisson distribution with this integrated jump intensity $\Lambda(t)$. When conditioning on N_t , we will have:

$$P(N_t = n) = \frac{\exp(-\Lambda(t)) \Lambda(t)^n}{n!} \quad (38)$$

We now derive the conditional density $P(\ln S_t | N_t = n, S_0)$, and here we can start with the case of one jump. When there is exactly one jump in $[0, t]$, the jump time s_1 is random. Given jump time s_1 , in a time-inhomogeneous setting, the instantaneous probability of a jump at time s_1 is proportional to λ_{s_1} . According to the dynamics of non-stationary MJD, the continuous part of the log-return leads to a normal distribution with mean being $a_1 = \ln S_0 + \int_0^{s_1} \left(\mu_s - \lambda_s k_s - \frac{\sigma_s^2}{2} \right) ds + \nu_{s_1}$, and variance being $b_1^2 = \int_0^{s_1} \sigma_s^2 ds + \gamma_{s_1}^2$. Thus, the conditional density of $\ln S_t$ given one jump at time s_1 is $\phi(\ln S_t; a_1, b_1^2)$, where $\phi(\cdot; a_1, b_1^2)$ denotes the Gaussian density with mean a_1 and variance b_1^2 . Since the jump could have occurred at any time in $[0, t]$, we must integrate over the possible jump time s_1 . Therefore, the conditional density given $N_t = 1$ is:

$$P(\ln S_t | N_t = 1, S_0) = \frac{1}{\Lambda(t)} \int_0^t \lambda_{s_1} \phi(\ln S_t; a_1, b_1^2) ds_1, \quad (39)$$

where $\frac{1}{\Lambda(t)}$ normalizes the density.

When generalizing to the case of $N_t = n$, the conditional density $P(\ln S_t | N_t = n, S_0)$ is defined via an integration over the n jump times, with the jump times denoted by $0 \leq s_1, \dots, s_n \leq t$.

Because the process is time-inhomogeneous, the probability density that a jump occurs at a specific time s_i is given by the instantaneous rate λ_{s_i} , therefore for a given set of jump times, the joint density for the jumps is proportional to $\prod_{i=1}^n \lambda_{s_i}$. The conditional density can be written as:

$$P(\ln S_t | N_t = n, S_0) = \frac{1}{\Lambda(t)^n} \int \cdots \int_{[0, t]} \prod_{i=1}^n \lambda_{s_i} \phi(\ln S_t; a_n, b_n^2) ds_1 \cdots ds_n \quad (40)$$

Here $\phi(\ln S_t; a_n, b_n^2)$ is the density of a normal distribution with mean a_n and variance b_n^2 , which are defined by:

$$a_n = \ln S_0 + \int_0^t \left(\mu_s - \lambda_s k_s - \frac{\sigma_s^2}{2} \right) ds + \sum_{i=1}^n \nu_{s_i}$$

$$b_n^2 = \int_0^t \sigma_s^2 dt + \sum_{i=1}^n \gamma_{s_i}^2$$

For convenience, we may write the mixture term as

$$\Phi_n = \int \cdots \int_{[0, t]} \prod_{i=1}^n \lambda_{s_i} \phi(\ln S_t; a_n, b_n^2) ds_1 \cdots ds_n \quad (41)$$

Therefore, for the time-varying SDEs, the conditional probability of $\ln S_t$ is given by,

$$P(\ln S_t | S_0) = \sum_{n=0}^{\infty} \frac{\exp(-\Lambda(t))}{n!} \Phi_n. \quad (42)$$

C Proofs of Theorem and Proposition

C.1 Proof of Theorem 4.1

Theorem 4.1. *Let the likelihood approximation error in Eq. (12), truncated to at most κ jumps, be*

$$\Psi_\kappa(t, \delta) := \sum_{n=\kappa+1}^{\infty} P(\Delta N = n) P(\ln S_{t+\delta} \mid S_t, \mathcal{C}, \Delta N = n).$$

Then, $\Psi_\kappa(t, \delta)$ decays at least super-exponentially as $\kappa \rightarrow \infty$, with a convergence rate of $O(\kappa^{-\kappa})$.

Before diving into the proof, we first introduce two important lemmas.

Lemma C.1 (Theorem 2 in [2]). *Let $Y \sim \text{Pois}(m)$ be a Poisson-distributed random variable with mean m . Its distribution function is defined as $P(Y \leq k) := \exp(-m) \sum_{i=0}^k \frac{m^i}{i!}$, with integer support $k \in \{0, 1, \dots, \infty\}$. For $k = 0$ and $k = \infty$, one has $P(Y \leq 0) = \exp(-m)$, $P(Y \leq \infty) = 1$. For every other $k \in \{1, 2, 3, \dots\}$, the following inequalities hold:*

$$\Phi\left(\text{sign}(k - m)\sqrt{2H(m, k)}\right) < P(Y \leq k) < \Phi\left(\text{sign}(k + 1 - m)\sqrt{2H(m, k + 1)}\right),$$

where $H(m, k)$ is the Kullback-Leibler (KL) divergence between two Poisson-distributed random variables with respective means m and k :

$$H(m, k) = D_{KL}(\text{Pois}(m) \parallel \text{Pois}(k)) = m - k + k \ln\left(\frac{k}{m}\right).$$

And $\Phi(x)$ is the cumulative distribution function (CDF) of the standard normal distribution and $\text{sign}(\cdot)$ is the signum function.

Lemma C.1 is particularly helpful in our proof below. We also acknowledge its foundation in an earlier work [91], which provides many insights and a profound amount of valuable knowledge on its own.

Lemma C.2 (Bounds on the Standard Normal CDF). *The following upper bound for $\Phi(\cdot)$ holds when $x < 0$:*

$$\Phi(x) < \frac{\phi(x)}{|x|},$$

where $\phi(x) = \frac{\exp(-x^2/2)}{\sqrt{2\pi}}$ is the probability density function of the standard normal distribution.

Proof. By the Mills' ratio inequality for the Gaussian distribution [92], we have $1 - \Phi(x) < \frac{\phi(x)}{x}$, $\forall x > 0$. Using the identity $\Phi(-x) = 1 - \Phi(x)$ for $x > 0$, we immediately obtain: $\Phi(-x) < \frac{\phi(x)}{x}$, $\forall x > 0$. For $x < 0$, substituting $-x$ into the previous bound and noting that $\phi(-x) = \phi(x)$, we obtain $\Phi(x) < \frac{\phi(x)}{|x|}$, $\forall x < 0$. \square

Proof of Theorem 4.1.

Proof. The original likelihood objective in Eq. (12) is as follows:

$$\begin{aligned} P(\ln S_{t+\delta} \mid S_t, \mathcal{C}) &= \sum_{n=0}^{\infty} P(\Delta N = n) P(\ln S_{t+\delta} \mid S_t, \mathcal{C}, \Delta N = n) \\ &= \sum_{n=0}^{\infty} \exp(-\lambda_{\rho_t} \delta) \frac{(\lambda_{\rho_t} \delta)^n}{n!} \phi(\ln S_{t+\delta}; a_{n,\delta}, b_{n,\delta}^2) \\ &= \sum_{n=0}^{\infty} \exp(-\lambda_{\rho_t} \delta) \frac{(\lambda_{\rho_t} \delta)^n}{n!} \frac{1}{\sqrt{2\pi b_{n,\delta}^2}} \exp\left(-\frac{(\ln S_{t+\delta} - a_{n,\delta})^2}{2b_{n,\delta}^2}\right) \end{aligned} \quad (43)$$

where δ is a small time change so that $\rho_t - 1 \leq t < t + \delta < \rho_t$, $a_{n,\delta} = \ln S_t + (\mu_{\rho_t} - \lambda_{\rho_t} k_{\rho_t} - \sigma_{\rho_t}^2/2)\delta + n\nu_{\rho_t}$ and $s_{n,\delta}^2 = \sigma_{\rho_t}^2 \delta + \gamma_{\rho_t}^2 n$.

We define the truncation error with a threshold κ as:

$$\Psi_{\kappa}(t, \delta) := \sum_{n=\kappa+1}^{\infty} P(\Delta N = n) P(\ln S_{t+\delta} \mid S_t, \mathcal{C}, \Delta N = n). \quad (44)$$

The second term $P(\Delta N = n) P(\ln S_{t+\delta} \mid S_t, \mathcal{C}, \Delta N = n)$ is a Gaussian density function and upper bounded by $\frac{1}{\sqrt{2\pi b_{n,\delta}^2}}$, so the truncation error $\Psi_{\kappa}(t, \delta)$ is bounded by:

$$\begin{aligned} \Psi_{\kappa}(t, \delta) &\leq \sum_{n=\kappa+1}^{\infty} \frac{1}{\sqrt{2\pi b_{n,\delta}^2}} P(\Delta N = n) && \text{(Gaussian density bound)} \\ &\leq \frac{1}{\sqrt{2\pi b_{\kappa+1,\delta}^2}} \sum_{n=\kappa+1}^{\infty} P(\Delta N = n) && (b_{\kappa,\delta} \text{ increases as } \kappa \text{ goes up}) \\ &= \frac{1}{\sqrt{2\pi b_{\kappa+1,\delta}^2}} (1 - \sum_{n=0}^{\kappa} P(\Delta N = n)) && \text{(property of Poisson CDF)} \\ &< \frac{1}{\sqrt{2\pi b_{\kappa+1,\delta}^2}} \left(1 - \Phi(\text{sign}(\kappa - \lambda_{\rho_t} \delta) \sqrt{2D_{\text{KL}}(\text{Pois}(\lambda_{\rho_t} \delta) \parallel \text{Pois}(\kappa))}) \right) && \text{(Lemma C.1)} \\ &= \frac{1}{\sqrt{2\pi b_{\kappa+1,\delta}^2}} \Phi(\text{sign}(\lambda_t \delta - \kappa) \sqrt{2D_{\text{KL}}(\text{Pois}(\lambda_{\rho_t} \delta) \parallel \text{Pois}(\kappa))}) && \text{(Gaussian CDF)} \end{aligned}$$

As stated above, the KL divergence between two Poisson distributions follows

$$D_{\text{KL}}(\text{Pois}(a) \parallel \text{Pois}(b)) = a - b + b \ln\left(\frac{b}{a}\right)$$

Therefore,

$$\begin{aligned} \Psi_{\kappa}(t, \delta) &< \frac{1}{\sqrt{2\pi b_{\kappa+1,\delta}^2}} \Phi\left(\text{sign}(\lambda_{\rho_t} \delta - \kappa) \sqrt{2D_{\text{KL}}(\text{Pois}(\lambda_{\rho_t} \delta) \parallel \text{Pois}(\kappa))}\right), \\ &= \frac{1}{\sqrt{2\pi(\sigma_{\rho_t}^2 \delta + \gamma_{\rho_t}^2 (\kappa + 1))}} \Phi\left(\text{sign}(\lambda_{\rho_t} \delta - \kappa) \sqrt{2(\lambda_{\rho_t} \delta - \kappa + \kappa \ln(\frac{\kappa}{\lambda_{\rho_t} \delta}))}\right) \\ &= \frac{1}{\sqrt{2\pi(\sigma_{\rho_t}^2 \delta + \gamma_{\rho_t}^2 (\kappa + 1))}} \Phi\left(\text{sign}\left(\frac{\lambda_{\rho_t} \delta}{\kappa} - 1\right) \sqrt{2(\lambda_{\rho_t} \delta - \kappa - \kappa \ln(\frac{\lambda_{\rho_t} \delta}{\kappa}))}\right) \end{aligned}$$

Intuitively, the truncation error decreases to zero as κ approaches infinity. Below, we analyze the convergence rate. When κ is sufficiently large, the term $\text{sign}\left(\frac{\lambda_{\rho_t} \delta}{\kappa} - 1\right)$ is negative. Consequently, the upper bound becomes:

$$\begin{aligned} \Psi_{\kappa}(t, \delta) &< \frac{1}{\sqrt{2\pi(\sigma_{\rho_t}^2 \delta + \gamma_{\rho_t}^2 (\kappa + 1))}} \Phi\left(-\sqrt{2(\lambda_{\rho_t} \delta - \kappa - \kappa \ln(\frac{\lambda_{\rho_t} \delta}{\kappa}))}\right) \\ &< \frac{1}{\sqrt{2\pi(\sigma_{\rho_t}^2 \delta + \gamma_{\rho_t}^2 (\kappa + 1))}} \frac{\phi\left(-\sqrt{2(\lambda_{\rho_t} \delta - \kappa - \kappa \ln(\frac{\lambda_{\rho_t} \delta}{\kappa}))}\right)}{\sqrt{2(\lambda_{\rho_t} \delta - \kappa - \kappa \ln(\frac{\lambda_{\rho_t} \delta}{\kappa}))}} && \text{(Lemma C.2)} \\ &= \frac{\exp(-\lambda_{\rho_t} \delta + \kappa + \kappa \ln(\frac{\lambda_{\rho_t} \delta}{\kappa}))}{2\pi \sqrt{2(\sigma_{\rho_t}^2 \delta + \gamma_{\rho_t}^2 (\kappa + 1))(\lambda_{\rho_t} \delta - \kappa - \kappa \ln(\frac{\lambda_{\rho_t} \delta}{\kappa}))}} \end{aligned}$$

As $\kappa \rightarrow \infty$, the numerator is dominated by $\exp(-\kappa \ln \kappa)$, which decays super-exponentially (faster than any polynomial or exponential decay). The denominator consists of two components:

- The first term, $\sqrt{\sigma_{\rho_t}^2 \delta + \gamma_{\rho_t}^2 (\kappa + 1)}$, scales asymptotically as $\gamma_{\rho_t} \sqrt{\kappa}$.
- The second term, $\sqrt{\left(\lambda_{\rho_t} \delta - \kappa - \kappa \ln \left(\frac{\lambda_{\rho_t} \delta}{\kappa}\right)\right)}$, scales as $\sqrt{\kappa \ln \kappa}$.

Combining all terms, the upper bound scales as:

$$\frac{1}{2\pi\sqrt{2}\gamma_{\rho_t}} \cdot \frac{\exp(-\kappa \ln \kappa)}{\kappa\sqrt{\ln \kappa}} \sim \frac{\kappa^{-\kappa}}{\kappa\sqrt{\ln \kappa}}.$$

The term $\kappa^{-\kappa}$ decays super-exponentially, while the denominator grows algebraically (as $\kappa\sqrt{\ln \kappa}$). The rapid decay of $\kappa^{-\kappa}$ dominates the polynomial growth in the denominator. The overall convergence rate is super-exponentially fast, at the rate of $O(\exp(-\kappa \ln \kappa))$ or equivalently $O(\kappa^{-\kappa})$.

Since the upper bound of $\Psi_\kappa(t, \delta)$ decays at the rate of $O(\kappa^{-\kappa})$ and $\Psi_\kappa(t, \delta)$ is strictly positive, this implies that the original quantity $\Psi_\kappa(t, \delta)$ must decay at least as fast as the upper bound. This completes the proof. \square

C.2 Proof of Proposition 4.2

Proposition C.3. *Let $1/M$ be the step size. Both standard EM and our solver exhibit a weak convergence rate of $O(1/M)$. Specifically, the vanilla EM has a weak error of $\epsilon_t^E \leq K \exp(Lt)/M$ for some constant $L > 0$, while ours achieves a tighter weak error of $\epsilon_t^R \leq K \exp(L(t - \lfloor t \rfloor))/M$.*

Proof. Here, we prove that this restart strategy has a tighter weak-convergence error than the standard EM solver. Recall that we let $\epsilon_t := |\mathbb{E}[g(\bar{S}_t)] - \mathbb{E}[g(S_t)]|$ be the standard weak convergence error [75], where S_t is the ground truth state, \bar{S}_t is the estimated one using certain sampling scheme and g is a K -Lipschitz continuous function. We denote the weak convergence errors of our restarted solver and the standard EM solver by ϵ_t^R and ϵ_t^E , respectively.

Step 1: Standard EM Results on Time-Homogeneous MJD SDEs

Early works on jump-diffusion SDE simulations explored the weak error bounds, which we summarize as follows. For time-homogeneous MJD SDEs, the error term ϵ_t^E of the standard EM method is dominated by $\epsilon_t^E \leq K \exp(Lt)/M$. This is supported by the following: (a) Theorem 2.2 in [76] establishes the $O(1/M)$ rate; (b) Sec. 4-5 of [76] and Theorem 2.1 of [93] shows that ϵ_t^E grows exponentially regarding time with a big- O factor $O(e^{K_p(t)})$. In particular, the time-dependent term in the error bound $e^{K_p(t)}$ used in the proof of [76] is rooted in their Lemma 4.1, which can be proven in a more general setting in [93]; e.g., Eq. (2.16) in [93] discusses concrete forms of $K_p(t)$, which can be absorbed into $O(e^{Lt})$ for some constant $L > 0$. Lastly, the K -Lipschitz condition of the function g provides the coefficient K in the bound. For a detailed proof—which is more involved and not central to the design and uniqueness of our algorithm—we refer the reader to [75, 94]. When combining the above existing results from the literature, we can derive the error bound of $\epsilon_t^E \leq K \exp(Lt)/M$.

Step 2: Standard EM Results on Time-Inhomogeneous MJD SDEs

Our paper considers time-inhomogeneous MJD SDEs, with parameters fixed within each interval $[\tau - 1, \tau)$ ($\tau \in \mathbb{N}$, $\tau \geq 1$). This happens to align with the Euler-Peano scheme for general time-inhomogeneous SDEs approximation. As a specific case of time-varying Lévy processes, our MJD SDEs retain the same big- O bounds as the time-homogeneous case. Namely, the standard EM solver has the same weak convergence error $\epsilon_t^E \leq K \exp(Lt)/M$, as in the time-homogeneous MJD SDEs. This can be justified by extending Section 5 of [76] that originally proves the EM's weak convergence for time-homogeneous Lévy processes. Specifically, the core technique lies in the Lemma 4.1 of [76], which, based on [93], is applicable to both time-homogeneous and Euler-Peano-style inhomogeneous settings (see Remark 3.3.3 in [93]). Therefore, equivalent weak convergence bounds could be attained by extending Lemma 4.1 of [76] with proofs from [93] thanks to the Euler-Peano formulation.

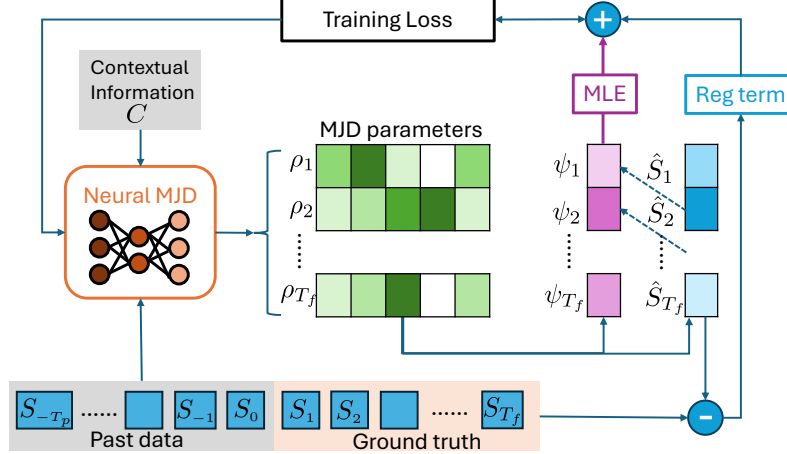


Figure 4: Neural MJD training pipeline. The symbol ρ represents the MJD parameters $\{\mu_\tau, \sigma_\tau, \lambda_\tau, \nu_\tau, \gamma_\tau\}$ in our model.

Step 3: Our Restarted EM Solver Error Bound

We now discuss the error bound for the restarted EM solver, ϵ_t^R . Thanks to explicit solutions for future states $\{S_1, S_2, \dots, S_{T_f}\}$, we can analytically compute their mean $\mathbb{E}[S_\tau | \mathcal{C}]$, $\tau \geq 1$, based on Eq. (13), which greatly simplifies the analysis. Using the restart mechanism in line 10 of Alg. 2, we ensure that $\mathbb{E}[\hat{S}_\tau | \mathcal{C}]$ from our restarted EM solver closely approximates the true $\mathbb{E}[S_\tau | \mathcal{C}]$ at restarting times. ϵ_t^R is significantly reduced when restart happens (when t is an integer in our context for simplicity), then it grows again at the same rate as the standard EM method until the next restart timestep. This explains the $O(e^{t-\lfloor t \rfloor})$ difference in the error bounds of ϵ_t^R and ϵ_t^E , where $\lfloor t \rfloor$ is the last restart time. Note that we could make the restart timing more flexible to potentially achieve a tighter bound in terms of weak convergence. However, this may affect the diversity of the simulation results, as the fidelity of path stochasticity could be impacted. \square

D Experiment Details

D.1 Baseline, Model Architecture, and Experiment Settings

For the statistical BS and MJD baselines, we assume a stationary process and estimate the parameters using a numerical MLE objective based on past sequences. For the other deep learning baselines, including DDPM, EDM, FM, Neural BS, and Neural MJD, we implement our network using the standard Transformer architecture [25]. All baseline methods are based on the open-source code released by their authors, with minor modifications to adapt to our datasets. Note that the technical term *diffusion* in the context of SDE modeling (e.g., Merton jump diffusion) should not be conflated with diffusion-based generative models [45]. While both involve SDE-based representations of data, their problem formulations and learning objectives differ significantly.

We illustrate the training loss computation pipeline for Neural MJD in Fig. 4. Notably, the loss computation can be processed in parallel across the future time-step horizon, eliminating the need for recursive steps during training. We normalize the raw data into the range of $[0, 1]$ for stability and use a regularization weight $\omega = 1.0$ during training. All experiments were run on NVIDIA A40 and A100 GPUs (48 GB and 80 GB VRAM, respectively).

D.2 Datasets Details

For all datasets, we normalize the input data using statistics computed from the training set. For non-denoising models, normalization maps the data to the range $[0, 1]$. In contrast, for denoising models (DDPM, EDM, FM), we scale the data to $[-1, 1]$ to align with standard settings used in image generation. Importantly, normalization coefficients are derived solely from the training set statistics. Further details on this process are provided below.

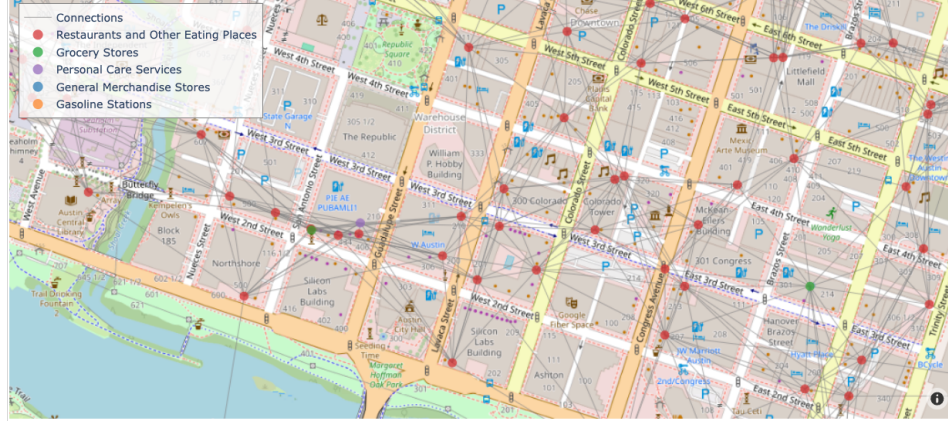


Figure 5: Visualization of Ego Graph Dataset Construction in Austin, Texas

Synthetic Data. We generate synthetic data using a scalar Merton Jump Diffusion model. The dataset consists of $N = 10,000$ paths over the interval $[0, 1]$, simulated using the Euler scheme with 100 time steps. To facilitate time-series forecasting, we employ a sliding window approach with a stride of 1, where the model predicts the next 10 frames based on the previous 10. The dataset is divided into 60% training, 20% validation, and 20% testing. For each simulation, model parameters are randomly sampled from uniform distributions: $\mu \sim U(0.1, 0.5)$, $\sigma \sim U(0.1, 0.5)$, $\lambda \sim U(3, 10)$, $\nu \sim U(-0.1, 0.1)$, and $\gamma \sim U(0.5, 1.0)$. These parameter choices ensure the presence of jumps, capturing the stochastic nature of the process.

SafeGraph&Advan Business Analytics Data. The SafeGraph&Advan business analytics dataset is a proprietary dataset created by integrating data from Advan [87] and SafeGraph [88] to forecast daily customer spending at points of interest (POIs) across Texas, USA. Both datasets are licensed through Dewey Data Partners under their proprietary commercial terms, and we comply fully with the terms. For each POI, the dataset includes time-series data with dynamic features and static attributes. Additionally, ego graphs are constructed based on geodesic distances, where each POI serves as a central node connected to its 10 nearest neighbors. An visualization is shown in Fig. 5. Specifically, we use POI area, brand name, city name, top and subcategories (based on commercial behavior), and parking lot availability as static features. The dynamic features include spending data, visiting data, weekday, opening hours, and closing hours. These features are constructed for both ego and neighboring nodes. Based on the top category, we determine the maximum spending in the training data and use it to normalize the input data for both training and evaluation, ensuring a regularized numerical range. For training stability, we clip the minimum spending value to 0.01 instead of 0 to enhance numerical stability for certain methods.

We adopt a sliding window approach with a stride of 1, using the past 14 days as input to predict spending for the next 7 days. The dataset spans multiple time periods: the training set covers January–December 2023, the validation set corresponds to January 2024, and the test set includes February–April 2024. This large-scale dataset consists of approximately 3.9 million sequences for training, 0.33 million for validation, and 0.96 million for testing.

S&P 500 Stock Price Data. The S&P 500 dataset [89] is a publicly available dataset from Kaggle that provides historical daily stock prices for 500 of the largest publicly traded companies in the U.S (CC0 1.0 Universal license). It primarily consists of time-series data with date information and lacks additional contextual attributes. We include all listed companies and construct a simple fully connected graph among them. Therefore, for models capable of handling graph data, such as GCN, our implemented denoising models, and Neural MJD, we make predictions for all companies (represented as nodes) simultaneously. This differs from the ego-graph processing used in the SafeGraph&Advan dataset, where predictions are made only for the central node, while neighbor nodes serve purely as contextual information. To normalize the data, we determine the maximum stock price for each company in the training data, ensuring that input values fall within the $[0, 1]$ range during training.

Following the approach used for the business analytics dataset, we apply a sliding window method with a stride of 1, using the past 14 days as input to predict stock prices for the next 7 days. The dataset is split into training (Jan.–Dec. 2016), validation (Jan. 2017), and testing (Feb.–Apr. 2017)

sets. In total, it contains approximately 62K sequences for training, 5K for validation, and 15K for testing. To better distinguish the effects of different methods on the S&P 500 dataset, we use an adjusted R^2 score $R^2 = 1 - (1 - R_{\text{reg}}^2) \cdot \frac{n-1}{n-p-1}$, where n is the sample size and we set the number of explanatory variables p to be $(k-1)(n-1)/k$, where $k = 70.0$.

D.3 Additional Deterministic Time-Series Baselines (Third-Party Implementations)

For completeness, we also report results from third-party implementations of Autoformer [95], TiDE [96], and N-HiTS [97], provided by the NeuralForecast library (NIXTLA). Results in the table were produced with the publicly available NeuralForecast package on the same train/validation/test splits and identical data input (*e.g.* exogenous stock ticker information) as our main experiments, using the package’s default training settings without modification.

Table 6: Quantitative results from NeuralForecast (NIXTLA) implementations on the **S&P 500** stock dataset.

Model	MAE ↓	MSE ↓	R^2 ↑
Autoformer	81.0	2.73e04	0.061
TiDE	27.7	7.28e03	0.750
N-HiTS	15.5	1.86e03	0.936

D.4 Limitations

Our approach explicitly models discontinuities (jumps) in the time series. Consequently, if the underlying data lack such jump behaviors—i.e., if they are extremely smooth and exhibit no abrupt shifts—our jump component may be inaccurately estimated or effectively unused. In these scenarios, the model can underperform compared to simpler or purely continuous alternatives that do not rely on capturing sudden changes. For applications where jumps are absent or extremely rare, users should first verify the presence (or likelihood) of discontinuities in their dataset before adopting our framework. Additionally, one potential extension is to design an adaptive mechanism that can automatically deactivate or regularize the jump component when the data do not exhibit significant jump behavior, thereby reducing unnecessary complexity and improving general performance on smooth series.

D.5 Vanilla Euler Solver

Algorithm 3 Vanilla Euler-Maruyama Method

Require: Total solver steps M

- 1: $\mathbf{C} \sim \mathcal{D}_{\text{test}}$, with $\mathbf{C} = [S_{-T_p:0}, C]$
 - 2: $\{\mu_\tau, \sigma_\tau, \lambda_\tau, \nu_\tau, \gamma_\tau\}_{\tau=1}^{T_f} \leftarrow f_\theta(\mathbf{C})$
 - 3: $\Delta \leftarrow \frac{T_f}{M}$ ▷ Solver time-step
 - 4: **for** $i = 0, \dots, M-1$ **do**
 - 5: $t_i \leftarrow i\Delta, t_{i+1} \leftarrow (i+1)\Delta, \rho_{t_i} \leftarrow \lfloor t_i \rfloor + 1$
 - 6: $\alpha_i \leftarrow (\mu_{\rho_{t_i}} - \lambda_{\rho_{t_i}} k_{\rho_{t_i}} - \sigma_{\rho_{t_i}}^2/2)\Delta$ ▷ Drift
 - 7: $\beta_i \leftarrow \sigma_{\rho_{t_i}} \sqrt{\Delta} z_1$, with $z_1 \sim \mathcal{N}(0, 1)$ ▷ Diffusion
 - 8: $\zeta_i \leftarrow \kappa \nu_{\rho_{t_i}} + \sqrt{\kappa} \gamma_{\rho_{t_i}} z_2$
 with $\kappa \sim \text{Pois}(\lambda_{\rho_{t_i}} \Delta), z_2 \sim \mathcal{N}(0, 1)$ ▷ Jump
 - 9: $\ln \bar{S}_{t_{i+1}} \leftarrow \ln \bar{S}_{t_i} + \alpha_i + \beta_i + \zeta_i$
 - 10: **return** $\{\bar{S}_{t_i}\}_{i=1}^M$
-

We present the standard Euler–Maruyama solver in Alg. 3, which is used in the ablation study for comparison with our restarted Euler solver.

E Impact Statement

This paper introduces Neural MJD, a learning-based time series modeling framework that integrates principled jump-diffusion-based SDE techniques. Our approach effectively captures volatile dynamics, particularly sudden discontinuous jumps that govern temporal data, making it broadly applicable to business analytics, financial modeling, network analysis, and climate simulation. While highly useful for forecasting, we acknowledge potential ethical concerns, including fairness and unintended biases in data or applications. We emphasize responsible deployment and continuous evaluation to mitigate inequalities and risks.