

On the representation for stochastic graph delay propagation

Shibo Zeng, Weiguo Gao, Yingzhou Li*

School of Mathematical Sciences, Fudan University, Shanghai, China
23110180055@m.fudan.edu.cn, wggao@fudan.edu.cn, yingzhouli@fudan.edu.cn

Abstract

In this work, we utilize the Hermite expansion to approximate the distributions of the sum and maximum of independent random variables. We model distributions with a three-segment representation, where the left and right tails are respectively modeled as combinations of Hermite functions, and the intermediate segment is approximated by piecewise polynomials. This approximation admits rigorous L^2 - and pointwise convergence properties supported by classical results. We develop an algorithmic framework for applying our model to the graph delay propagation problem, where sum and max operations are performed on the proposed model structure. Numerical experiments demonstrate that our model can capture the quantile values with high accuracy compared to Monte Carlo simulation results, significantly outperforming classical Gaussian-based models.

Keywords: graph delay propagation, the Hermite expansion, tail approximation, Gaussian decay, non-linear least-squares

1 Introduction

1.1 Background

This paper studies the problem of estimating delay propagation on a directed acyclic graph (DAG), which has long been a topic of interest in static timing analysis (STA). Suppose $G = (V, E)$ is a DAG, where V denotes the node set and E denotes the edge set. Each node and edge is associated with a *delay*, representing the time it takes for a signal to pass through. The delay propagation problem aims to determine the worst-case signal arrival times from the source node(s) to the sink node(s). These arrival times are then checked against the clocking constraints to identify potential timing violations. These violations may cause registers to capture incorrect data, potentially resulting in function failure of the entire circuit.

There are two general approaches to the delay propagation problem in STA: *graph-based analysis (GBA)* and *path-based analysis (PBA)*. A detailed introduction to these two methods can be found in [1]. GBA is often used in the first stage of timing analysis, where delays are propagated iteratively along the graph. For each node v , the arrival time $D(v)$ is computed in topological order according to

$$D(v) = \max_{\substack{u \in V \\ (u,v) \in E}} \{D(u) + d(u,v)\} + d(v), \quad (1)$$

where $d(u,v)$ denotes the incremental delay on edge (u,v) , and $d(v)$ denotes the incremental delay incurred when the signal passes through node v .

After completing the GBA stage, paths that potentially violate the timing constraints will be identified. The PBA method is then applied to perform a more detailed and accurate examination, though at a higher computational cost. In this paper, we focus on the GBA framework, within which statistical static timing analysis (SSTA) is conducted.

In SSTA, each delay is modeled as a random variable due to on-chip variation (OCV) arising from spatial and temporal fluctuations, or a sum of random variables, each capturing the uncertainty contributed by a different source of variation. Under this formulation, our objective is to efficiently compute, for every node v in the graph, the accumulated delay (i.e., the arrival time) from the source node(s), based on Eq. (1).

*Corresponding author.

Modeling node delays and edge delays as random variables reduces the computation of $D(v)$ to a sequence of binary operations over random variables, where each operation is either a max operation or a sum operation. To be precise, we consider the following basic operations on independent random variables X_1 and X_2 :

$$X = X_1 + X_2, \quad \text{or} \quad X = \max\{X_1, X_2\}.$$

Other binary operations that may arise in practice, such as subtraction and minimum,

$$X = X_1 - X_2, \quad \text{or} \quad X = \min\{X_1, X_2\},$$

can be equivalently transformed into the above forms. Therefore, without loss of generality, we restrict our discussion to the sum and max operations.

Although X_1 and X_2 might be statistically dependent in practical circuits, we ignore the correlation between X_1 and X_2 throughout the calculation, as is commonly assumed in STA [2]. These two binary operations put forward two crucial problems. One is how to appropriately model the random variables, and the other is how to efficiently and accurately evaluate the max and sum operations.

A widely adopted simplification is to model all delay random variables as Gaussian. The probabilistic density function (PDF) of a Gaussian random variable $\mathcal{N}(\mu, \sigma^2)$ is given by

$$\phi(x; \mu, \sigma) := \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (2)$$

When $\mu = 0$ and $\sigma = 1$, this expression reduces to the PDF of the standard normal distribution $\mathcal{N}(0, 1)$, which we denote by $\phi(x)$ for brevity.

The cumulative distribution function (CDF) of a Gaussian variable $\mathcal{N}(\mu, \sigma^2)$ is given by

$$\Phi(x; \mu, \sigma) := \Phi\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right) \right], \quad (3)$$

where $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ is the standard error function. However, Gaussian random variables fail to capture the characteristics of delay, since the advancement of manufacturing technology has made the distribution of gate delay and interconnect delay increasingly complex. Moreover, the maximum of independent Gaussian random variables is not necessarily Gaussian. When more factors are taken into account, such as the skewness of the random variables, more sophisticated models become necessary.

1.2 Related Work

There have been massive studies concerning the delay propagation problem. The earliest methods, as mentioned above, modeled each node delay and interconnect delay as a Gaussian random variable. Their algorithms were based on the idea of the classic Clark's algorithm [3]. For example, Berkelaar [4] proposed a linear-time algorithm for Gaussian delay propagation under the assumption that all delays are independent, while Tsukiyama et al. [5] considered the correlations between delays.

A more advanced Gaussian framework was proposed by Kang et al. [6], who introduced Levelized Covariance Propagation (LCP), in which covariance information is maintained and propagated level-by-level. Using Clark's algorithm, LCP computes the mean μ_{\max} and standard deviation σ_{\max} of the maximum of correlated Gaussian random variables and then projects the result back to $\mathcal{N}(\mu_{\max}, \sigma_{\max}^2)$ for further propagation. This can be viewed as a *moment-matching* approach, which is also useful for sums of distributions that are not closed under addition.

Chopra et al. [7] took skewness into consideration by modeling the distributions to propagate as skew-normal random variables. They proposed a statistical max operation method that captures the skewness of the delay distributions, which shows improved accuracy results compared to the classical Clark's approach.

Hcine and Bouallegue [8] made a similar effort, where the sum of independent lognormal random variables was approximated by a log-skew-normal distribution. However, instead of using the moment-matching method as proposed by Fenton [9], the authors chose a tail-matching method to capture the tail asymptotics of the desired distribution. These log-type distributions are suitable for modeling delay since they naturally preserve positivity during propagation.

Recently, Jin et al. [10] proposed a statistical delay model based on the log-extended-skew-normal (LESN) distribution, which extends the conventional log-skew-normal model by incorporating kurtosis control. They also applied the moment-matching method. In particular, the LESN model has a precise prediction of the 3σ delay, which is crucial for the worst-case analysis.

A different model has been formulated by Cheng et al. [11] where all delays to propagate were modeled as quadratic polynomials of several independent random variables. While this model supports closed-form delay summation, it also provides an approximation method for computing the maximum of independent random variables by second-degree polynomial fitting.

Apart from the method using moment-matching and polynomial fitting, Azuma et al. [12] introduced a novel framework for approximating the maximum of Gaussian random variables by a Gaussian mixture model (GMM). Utilizing the fact that the convolution of PDFs of Gaussian random variables is still a Gaussian PDF, computing the sum in this model is straightforward. To address the maximum problem, the authors also used a moment-matching method to fit the first two moments of the maximum of GMMs. Lately, a series of works [13–15] have been proposed to further apply this method to the large-scale graph propagation problem. Furthermore, Bosák et al. [2] used a histogram-based approach and optimization methods to propagate delay distributions on a graph.

Our work is inspired not only by the works mentioned above, but also by the classical Hermite polynomial expansion theory. The Gram–Charlier series, for example, is a well-known method for approximating a probability distribution by a series of Hermite polynomials (for probabilists) multiplied by a Gaussian kernel through moment-matching, as discussed in [16]. In recent years, there have also been several works that apply this method to problems in different fields. For example, Capodaglio et al. [17] applied the Gram–Charlier series to approximate the PDF of solutions to PDEs with random parameters, and Dufresne and Li [18] used it in Asian option pricing. However, all these methods face the poor pointwise convergence property of the Gram–Charlier series, as was discussed in [19].

1.3 Contributions

We present a general framework for representing and propagating random variables on DAGs under binary operations such as max and sum. Each distribution is approximated by a three-segment formulation consisting of polynomial-Gaussian tail models and an interpolation approach across the intermediate segment. During the propagation process, this structure is preserved, whose convergence properties are proved base on the theory of the Hermite function expansion. Moreover, we presented an algorithmic framework to evaluate the binary operations and reproject the input and resulting distributions back into our model family through CDF sampling and a nonlinear least-squares tail-fitting procedure. This yields a complete and operational representation that remains stable under propagation. Numerical experiments are conducted to demonstrate that our model and algorithm can capture the quantile values with high accuracy with respect to Monte Carlo simulation results, which improves over classical Gaussian-based models.

The rest of this paper is organized as follows. In Sec. 2, we introduce briefly the mathematical preliminaries that will be used later in our model. In Sec. 3, we present the mathematical formulation of the delay propagation model. In Sec. 4, we illustrate the completeness of the model by proving the L^2 -convergence and pointwise convergence of our model. In Sec. 5, we introduce algorithms for propagating the proposed distributions through sum and max operations. In Sec. 6, we present numerical results of our model and our proposed algorithms.

2 Preliminaries

In this section, we summarize the mathematical preliminaries needed for the analysis in subsequent sections. To support the convergence results established later, we review the standard and parametric Hermite polynomials and derive their associated orthogonality relations.

2.1 Standard Hermite polynomials

Unless otherwise stated, we adopt the physicists’ convention for the Hermite polynomials, defined by

$$H_k(x) := (-1)^k e^{x^2} \frac{d^k}{dx^k} e^{-x^2}, \quad k = 0, 1, 2, \dots$$

We refer to these as the *standard* Hermite polynomials to distinguish them from the *parametric* Hermite polynomials, which will be discussed in the following subsection.

Function $H_k(x)$ is a polynomial of degree k in x . The Hermite polynomials satisfy the following three-term recurrence relation:

$$H_{k+1}(x) = 2xH_k(x) - 2kH_{k-1}(x), \quad k = 1, 2, \dots,$$

with $H_0(x) = 1$ and $H_1(x) = 2x$. They are orthogonal with respect to the weight function $\omega(x) := e^{-x^2}$:

$$\int_{-\infty}^{+\infty} H_m(x)H_n(x)e^{-x^2} dx = \gamma_n \delta_{mn},$$

where the normalization constant γ_n is given by $\gamma_n := 2^n n! \sqrt{\pi}$.

2.2 Parametric Hermite polynomials and Hermite functions

For better approximation capability, we use the parametric Hermite polynomials/functions, where we modify the weight function as $\omega(x; \mu, \sigma) := e^{-(x-\mu)^2/\sigma^2}$ for $\sigma > 0$ and μ being real parameters. Also, we denote by $H_k(x; \mu, \sigma) := H_k\left(\frac{x-\mu}{\sigma}\right)$ the parametric Hermite polynomials.

The orthogonality of the parametric Hermite polynomials can be established through a straightforward computation:

$$\int_{-\infty}^{+\infty} H_m(x; \mu, \sigma)H_n(x; \mu, \sigma)e^{-\frac{(x-\mu)^2}{\sigma^2}} dx = \sigma \gamma_n \delta_{mn}.$$

The parametric Hermite functions $\psi_k(x; \mu, \sigma)$ are defined by

$$\psi_k(x; \mu, \sigma) := \frac{1}{\sqrt{\sigma \gamma_k}} H_k(x; \mu, \sigma) e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

The L^2 -completeness of $\{\psi_k(x; \mu, \sigma)\}_{k=0}^{\infty}$ is guaranteed by the following lemma, which is a direct generalization of the standard case in [20].

Lemma 2.1 (*L^2 -convergence of the parametric Hermite function expansion*). *The set of parametric Hermite functions $\{\psi_k(x; \mu, \sigma)\}_{k=0}^{\infty}$ forms a complete orthogonal basis in $L^2(\mathbb{R})$. That is, for any function $f \in L^2(\mathbb{R})$, there exists a unique sequence of coefficients $\{c_k(\mu, \sigma)\}_{k=0}^{\infty}$ such that we can write $f(x)$ as*

$$f(x) = \sum_{k=0}^{\infty} c_k(\mu, \sigma) \psi_k(x; \mu, \sigma).$$

The coefficients $c_k(\mu, \sigma)$ are given by the inner product in $L^2(\mathbb{R})$:

$$c_k(\mu, \sigma) = \int_{-\infty}^{+\infty} f(x) \psi_k(x; \mu, \sigma) dx.$$

The projection of $f(x)$ onto the span of the first N parametric Hermite functions is given by

$$\mathcal{P}_N^f(x; \mu, \sigma) = \sum_{k=0}^N c_k(\mu, \sigma) \psi_k(x; \mu, \sigma).$$

Then, the sequence of $\mathcal{P}_N^f(x)$ converges to $f(x)$ in the L^2 -norm:

$$\lim_{N \rightarrow \infty} \int_{-\infty}^{\infty} |f(x) - \mathcal{P}_N^f(x; \mu, \sigma)|^2 dx = 0.$$

The pointwise convergence property of the parametric Hermite polynomial expansion is analogous to that of the standard case given in [21]. Indeed, the parametric result is obtained from the standard case by the affine transform $x \mapsto (x - \mu)/\sigma$. Moreover, the pointwise convergence of the parametric Hermite function expansion for $f \in L^2(\mathbb{R})$ reduces to the pointwise convergence of the parametric Hermite polynomial expansion for $g = f \sqrt{\omega(\cdot; \mu, \sigma)} \in L^2_{\omega(\cdot; \mu, \sigma)}(\mathbb{R})$.

We state the pointwise convergence property of the parametric Hermite function expansion in the lemma below, paralleling the main theorem of [21].

Lemma 2.2 (Pointwise convergence of the parametric Hermite function expansion). *Suppose $f(x)$ satisfies the following conditions:*

- (1) $f(x)$ is absolutely integrable in any finite interval,
- (2) $f(x)$ is of limited variation in a certain interval $(x - \delta, x + \delta)$ for some $\delta > 0$,
- (3) $f(x)$ belongs to $L^2(\mathbb{R})$.

Then the series expansion $\sum_{k=0}^{\infty} c_k(\mu, \sigma) \psi_k(x; \mu, \sigma)$ converges and satisfies

$$\sum_{k=0}^{\infty} c_k(\mu, \sigma) \psi_k(x; \mu, \sigma) = \frac{f(x^+) + f(x^-)}{2}.$$

3 Model formulation

To accurately capture the delay distributions during propagation, especially in the tail segments, we observed that Gaussian models tend to underestimate the probability of large deviations. Inspired by the observation that empirical delay distributions often exhibit longer or skewed tails, we adopt a polynomial-modulated Gaussian model, where the Gaussian kernel is multiplied by a low-degree polynomial. This allows more flexibility in shaping the PDF while maintaining a Gaussian-like functional form.

To further support efficient computation, we represent the intermediate segment PDF with piecewise polynomials, yielding a three-segment model for the PDF on the whole real line. The resulting form retains analytical tractability while significantly enhancing the approximation capacity.

3.1 Basic PDF model

We adopt a three-segment PDF model for propagating independent random variables on a graph. The PDF of a random variable X to be propagated along the graph is modeled as

$$f(t) = \begin{cases} \sum_{j=0}^N c_j^{(l)} t^j \phi(t; \mu^{(l)}, \sigma^{(l)}), & t \leq q_0, \\ P_i(t), & q_{i-1} \leq t < q_i, \quad i = 1, 2, \dots, n, \\ \sum_{j=0}^N c_j^{(r)} t^j \phi(t; \mu^{(r)}, \sigma^{(r)}), & t \geq q_n. \end{cases} \quad (4)$$

Here n denotes the number of partition points in the intermediate segment, and N is the degree of the Gaussian-tail polynomials, both user-specified. The PDF on each subinterval $[q_{i-1}, q_i]$ in the intermediate segment is modeled by a polynomial $P_i(t)$, which is constructed to ensure continuity and smoothness of the overall PDF. Unless otherwise stated, we use an equally spaced partition in the intermediate segment, i.e., $q_i - q_{i-1} = (q_n - q_0)/n$, $i = 1, 2, \dots, n$.

Suppose $F(x)$ is the CDF of X . We choose q_0 and q_n as the quantile values corresponding to two prescribed CDF levels of X for each random variable X on the entire graph. For example, taking the CDF levels 0.135% and 99.865% yields the $\pm 3\sigma$ quantiles in the Gaussian case.

Remark 1. (a) In our general model (4), the intermediate segment of the PDF are piecewise polynomials of arbitrary order to preserve the regularity of the PDF. However, in practice we typically assume the intermediate segment to be piecewise constant, i.e., $P_i(t) = (F(q_i) - F(q_{i-1})) / (q_i - q_{i-1})$, as this choice tends to maintain the nonnegativity of the PDF. Selecting the degree of the intermediate polynomials involves a trade-off between model complexity and approximation accuracy. A more detailed study of the convergence rate for higher-degree models is deferred to future work. For the sake of simplicity, we assume $P_i(t)$ to be piecewise constant throughout the remainder of this paper.

(b) Our PDF model does not inherently provide non-negativity, which is a fundamental requirement for any valid probability density function. To address this issue, one may impose constraints on the leading coefficients of the tail polynomials. In the subsequent analysis of convergence properties, we restrict our attention to the case where f is indeed a valid probability density function.

3.2 Basic CDF model

During delay propagation, our primary interest lies in the CDF of the propagated random variables, since its behavior determines the target quantiles used to approximate the worst-case delay.

Given a random variable X whose PDF takes the form (4), the CDF of X can be evaluated numerically at any point x . When $q_0 \leq x \leq q_n$, the CDF can be evaluated efficiently due to the specific structure of the intermediate segment. For $x < q_0$ or $x > q_n$, the CDF can be respectively determined by a linear combination of

$$I_i(x; \mu, \sigma) \triangleq \int_{-\infty}^x t^i \phi(t; \mu, \sigma) dt, \quad \text{or} \quad J_i(x; \mu, \sigma) \triangleq \int_x^{\infty} t^i \phi(t; \mu, \sigma) dt.$$

We first derive a recurrence relation for $I_i(x; \mu, \sigma)$ by induction. Based on the relation

$$\frac{d}{dt} \phi(t; \mu, \sigma) = -\frac{t - \mu}{\sigma^2} \phi(t; \mu, \sigma),$$

we obtain for arbitrary $i \geq 2$

$$\begin{aligned} I_i(x; \mu, \sigma) &= \int_{-\infty}^x t^i \phi(t; \mu, \sigma) dt \\ &= \mu I_{i-1}(x; \mu, \sigma) - \sigma^2 \int_{-\infty}^x t^{i-1} \frac{d}{dt} \phi(t; \mu, \sigma) dt \\ &= \mu I_{i-1}(x; \mu, \sigma) - x^{i-1} \sigma^2 \phi(x; \mu, \sigma) + (i-1) \sigma^2 I_{i-2}(x; \mu, \sigma). \end{aligned}$$

For $i = 1$, a similar derivation yields

$$I_1(x; \mu, \sigma) = \mu I_0(x; \mu, \sigma) - \sigma^2 \phi(x; \mu, \sigma).$$

Thus, with a linear combination of elements in $\{I_i(x; \mu, \sigma)\}_{i \leq N}$, we can achieve any element in

$$\text{span} \{ \Phi(x; \mu, \sigma), \phi(x; \mu, \sigma), x\phi(x; \mu, \sigma), \dots, x^{N-1}\phi(x; \mu, \sigma) \}.$$

Also, as shown in the derivation above, the transition matrix between the two sets of functions is upper-triangular with nonzero diagonal entries, and is therefore invertible.

By applying the same argument to J_i for the right tail (see Appendix A), we obtain the following lemma:

Lemma 3.1. *Fix $\mu \in \mathbb{R}$, $\sigma > 0$, and $N \geq 1$. Then*

(1) *The function spaces*

$$\text{span} \{ I_0(x; \mu, \sigma), I_1(x; \mu, \sigma), \dots, I_N(x; \mu, \sigma) \}$$

and

$$\text{span} \{ \Phi(x; \mu, \sigma), \phi(x; \mu, \sigma), x\phi(x; \mu, \sigma), \dots, x^{N-1}\phi(x; \mu, \sigma) \}$$

coincide.

(2) *The function spaces*

$$\text{span} \{ J_0(x; \mu, \sigma), J_1(x; \mu, \sigma), \dots, J_N(x; \mu, \sigma) \}$$

and

$$\text{span} \{ 1 - \Phi(x; \mu, \sigma), \phi(x; \mu, \sigma), x\phi(x; \mu, \sigma), \dots, x^{N-1}\phi(x; \mu, \sigma) \}$$

coincide.

During delay propagation, we need to compute the sum, subtraction, maximum and minimum of two independent random variables X_1 and X_2 with known PDFs and CDFs. Since

$$X_1 - X_2 = X_1 + (-X_2), \quad \text{and} \quad \min(X_1, X_2) = -\max(-X_1, -X_2),$$

the subtraction and minimum operations can be reduced to sum and max operations, respectively. These reductions only involves sign inversion and therefore preserves independence. Thus, we consider only the

sum and max operations in the following discussion. Let F_1 and F_2 denote the CDF of X_1 and X_2 . The CDF of $X_1 + X_2$ and $\max(X_1, X_2)$ can be respectively given by the following formulae:

$$F_{X_1+X_2}(x) = \int_{-\infty}^{+\infty} F_1(x-t)f_2(t) dt = \int_{-\infty}^{+\infty} f_1(t)F_2(x-t) dt,$$

$$F_{\max(X_1, X_2)}(x) = F_1(x)F_2(x).$$

Using the formulae above, we can numerically evaluate the CDF of $X := X_1 \odot X_2$ at any point on \mathbb{R} , where \odot denotes either the sum or the max operation.

In the intermediate segment, we can use interpolation methods to derive the piecewise polynomial approximation of the CDF. In the tail segments, we aim to approximate the exact CDF of X by a linear combination of $\{I_i(x; \mu, \sigma)\}_{i=0}^N$ or $\{J_i(x; \mu, \sigma)\}_{i=0}^N$. In particular, we focus on approximating the L^2 -error and the L^∞ -error between the target CDF and the approximation. We present the derivation for the left tail below, while the right tail can be treated similarly and is shown in Appendix A. Also, only the max case is presented, as the sum case can be handled similarly. The target functions can be respectively written as:

$$\int_{-\infty}^{q_0} \left(F_1(x)F_2(x) - \sum_{i=0}^N c_i^{(l)} I_i(x; \mu^{(l)}, \sigma^{(l)}) \right)^2 dx \quad \text{or} \quad F_1(x)F_2(x) - \sum_{i=0}^N c_i^{(l)} I_i(x; \mu^{(l)}, \sigma^{(l)}), \quad \forall x \leq q_0,$$

where $\{c_i^{(l)}\}_{i=0}^N, \mu^{(l)}, \sigma^{(l)}$ are real coefficients, and q_0 is a left quantile point for $\max(X_1, X_2)$.

Take the left-tail L^2 -error as an example, where a similar derivation holds for L^∞ -error. According to Lem. 3.1, we can rewrite the left-tail model CDF as

$$F(x) = \alpha^{(l)} \Phi(x; \mu^{(l)}, \sigma^{(l)}) + \sum_{i=0}^{N-1} \beta_i^{(l)} x^i \phi(x; \mu^{(l)}, \sigma^{(l)}), \quad (5)$$

where $\alpha^{(l)}, \{\beta_i^{(l)}\}_{i=0}^{N-1}, \mu^{(l)}, \sigma^{(l)}$ are the parameters to be decided. Thus, minimizing the L^2 -error is equivalent to minimizing

$$\int_{-\infty}^{q_0} \left(\left(F_1(x)F_2(x) - \alpha^{(l)} \Phi(x; \mu^{(l)}, \sigma^{(l)}) \right) - \sum_{i=0}^{N-1} \beta_i^{(l)} x^i \phi(x; \mu^{(l)}, \sigma^{(l)}) \right)^2 dx. \quad (6)$$

If the Gaussian parameters $\mu^{(l)}$ and $\sigma^{(l)}$ are fixed, the problem of minimizing L^2 -error can be seen as a linear least-squares problem concerning the coefficients $\alpha^{(l)}$ and $\{\beta_i^{(l)}\}_{i=0}^{N-1}$.

4 Model justification and main results

In this section, we present L^2 - and pointwise convergence properties of the proposed model. Throughout the following discussion, we focus on the theoretical results concerning approximating $X = X_1 \odot X_2$ with \tilde{X} , where $X_k, k = 1, 2$ follow PDFs f_k of the form (4):

$$f_k(t) = \begin{cases} \sum_{j=0}^{N_k} c_{k,j}^{(l)} t^j \phi(t; \mu_k^{(l)}, \sigma_k^{(l)}), & t \leq q_{k,0}, \\ P_{k,i}(t), & q_{k,i-1} \leq t < q_{k,i}, \quad i = 1, 2, \dots, n_k, \\ \sum_{j=0}^{N_k} c_{k,j}^{(r)} t^j \phi(t; \mu_k^{(r)}, \sigma_k^{(r)}), & t \geq q_{k,n_k}, \end{cases} \quad (7)$$

and \tilde{X} also follows a PDF of the form (4). In what follows, we restrict our attention to the case where the intermediate pieces $P_{k,i}$ are constants on each subinterval. Before proceeding to justify our model, we introduce the class of random variables \mathcal{X} :

Definition 1. We denote by \mathcal{X} the class of real-valued continuous random variables X with CDF F such that there exist $\varepsilon > 0, \delta > 0$, and points $x_1, x_2 \in \mathbb{R}$ satisfying

$$F(x_1) = \varepsilon, \quad F(x_2) = 1 - \varepsilon,$$

and

$$F \in L^2(-\infty, x_1 + \delta) \quad \text{and} \quad 1 - F \in L^2(x_2 - \delta, +\infty).$$

We refer to \mathcal{X} as the *class of random variables with L^2 -integrable tails*.

4.1 Expressive power of the model

We first consider any random variable in class \mathcal{X} , and prove that it can be approximated by our model. This result characterizes the expressive power of our model: before performing delay propagation on the DAG, every input random variable must be projected onto the model space, and the approximation ensures that this projection is well-defined.

Theorem 4.1 (Expressive power of the model). *Suppose $X \in \mathcal{X}$ is a real-valued continuous random variable with cumulative distribution function F . Then there exists $\{f_{N,n}\}$, each element of which follows a PDF of the form (4), such that*

$$\lim_{n,N \rightarrow \infty} \left\| F(x) - \int_{-\infty}^x f_{N,n}(t) dt \right\|_2 = 0, \quad (8)$$

$$\lim_{n,N \rightarrow \infty} F(x) - \int_{-\infty}^x f_{N,n}(t) dt = 0, \quad \forall x \in \mathbb{R}. \quad (9)$$

Here q_0 and q_n in (4) are two fixed parameters, and we take $x_1 = q_0$, $x_2 = q_n$ in Def. 1.

Proof. We first use the Hermite polynomial theory to approximate the left tail and the right tail in a constructive manner. Without loss of generality, we assume that $\mu^{(l)} = 0$, $\sigma^{(l)} = 1$, $\mu^{(r)} = 0$, and $\sigma^{(r)} = 1$.

By Lem. 2.1 and Lem. 2.2, together with the preservation of L^2 -integrability under finite sums, we obtain:

$$\begin{cases} F(x) - \alpha^{(l)}\Phi(x) \in L^2(-\infty, q_0 + \delta), & \forall \alpha^{(l)} \in \mathbb{R}, \\ 1 - F(x) - \alpha^{(r)}(1 - \Phi(x)) \in L^2(q_n - \delta, +\infty), & \forall \alpha^{(r)} \in \mathbb{R}. \end{cases}$$

Thus, we can approximate $F(x) - \alpha^{(l)}\Phi(x)$ on the left tail and $1 - F(x) - \alpha^{(r)}(1 - \Phi(x))$ on the right tail using the Hermite function expansion.

Consider the left-tail case. Due to the Hermite function expansion convergence results in Lem. 2.1 and Lem. 2.2, there exist coefficients $\{\beta_i^{(l)}\}_{i=0}^{+\infty}$ such that

$$\begin{cases} \lim_{N \rightarrow +\infty} \sum_{i=0}^{N-1} \beta_i^{(l)} x^i \phi(x) = F(x) - \alpha^{(l)}\Phi(x), & \forall x \leq q_0, \\ \lim_{N \rightarrow +\infty} \|F(x) - \alpha^{(l)}\Phi(x) - \sum_{i=0}^{N-1} \beta_i^{(l)} x^i \phi(x)\|_{L^2(-\infty, q_0)} = 0. \end{cases}$$

We define $F_N^{(l)}(x) := \alpha^{(l)}\Phi(x) + \sum_{i=0}^{N-1} \beta_i^{(l)} x^i \phi(x)$ for $x \leq q_0$. Due to the isomorphism in Lem. 3.1 between

$$\text{span} \left\{ \int_{-\infty}^x \phi(t) dt, \int_{-\infty}^x t\phi(t) dt, \dots, \int_{-\infty}^x t^N \phi(t) dt \right\}$$

and

$$\text{span} \{ \Phi(x), \phi(x), \dots, x^{N-1}\phi(x) \},$$

we can map each $F_N^{(l)}(x)$ back to $f_N^{(l)}(x)$ that takes the left-tail form of (4) and satisfies $F_N^{(l)}(x) = \int_{-\infty}^x f_N^{(l)}(t) dt$. We can also get the right-tail expression for $f_N^{(r)}(x)$ in the same manner.

Thus, by defining

$$f_N(x) := \begin{cases} f_N^{(l)}(x), & x \in (-\infty, q_0], \\ f_N^{(r)}(x), & x \in [q_n, +\infty), \end{cases}$$

and

$$F_N(x) := \begin{cases} \int_{-\infty}^x f_N^{(l)}(t) dt, & x \in (-\infty, q_0], \\ 1 - \int_x^{+\infty} f_N^{(r)}(t) dt, & x \in [q_n, +\infty), \end{cases} \quad (10)$$

we obtain $f_N(x)$ and $F_N(x)$ on $(-\infty, q_0] \cup [q_n, +\infty)$ that satisfy

$$\begin{cases} \lim_{N \rightarrow \infty} \left\| F(x) - F_N(x) \right\|_{L^2(-\infty, q_0] \cup [q_n, +\infty)} = 0, \\ \lim_{N \rightarrow \infty} F(x) - F_N(x) = 0, & \forall x \in (-\infty, q_0] \cup [q_n, +\infty). \end{cases}$$

Next, we extend f_N onto (q_0, q_n) by defining $f_{N,n}$ for each $n > 1$ as

$$f_{N,n}(x) = \begin{cases} f_N(x), & x \in (-\infty, q_0] \cup [q_n, +\infty), \\ \frac{F(q_1) - F_N(q_0)}{q_1 - q_0}, & x \in (q_0, q_1), \\ \frac{F(q_{i+1}) - F(q_i)}{q_{i+1} - q_i}, & x \in [q_i, q_{i+1}), \quad i = 1, \dots, n-2, \\ \frac{F_N(q_n) - F(q_{n-1})}{q_n - q_{n-1}}, & x \in [q_{n-1}, q_n). \end{cases}$$

Now we can finally define $F_{N,n}$ as a CDF-form function:

$$F_{N,n}(x) := \int_{-\infty}^x f_{N,n}(t) dt, \quad x \in \mathbb{R}. \quad (11)$$

Here we note that although the representations of Eq. (10) and Eq. (11) on the right tail are different, we still have

$$F_{N,n}(x) = 1 - \int_x^{+\infty} f_{N,n}(t) dt, \quad x \in [q_n, +\infty),$$

since

$$\int_{-\infty}^{q_n} f_{N,n}(t) dt + \int_{q_n}^{+\infty} f_{N,n}(t) dt = F_N(q_n) + (1 - F_N(q_n)) = 1.$$

Thus, the convergence results of (8) and (9) restricted to $(-\infty, q_0] \cup [q_n, +\infty)$ are guaranteed, as

$$F_{N,n}(x) = F_N(x), \quad x \in (-\infty, q_0] \cup [q_n, +\infty).$$

For the intermediate segment, we define the auxiliary functions

$$h_n(t) = \frac{F(q_i) - F(q_{i-1})}{q_i - q_{i-1}}, \quad q_{i-1} \leq t < q_i, \quad i = 1, 2, \dots, n, \\ H_n(x) = F(q_0) + \int_{q_0}^x h_n(t) dt, \quad x \in [q_0, q_n].$$

For the L^2 -convergence in the intermediate segment, by the triangular inequality of the L^2 -norm we have

$$\|F_{N,n} - F\|_{L^2(q_0, q_n)} \leq \|F_{N,n} - H_n\|_{L^2(q_0, q_n)} + \|H_n - F\|_{L^2(q_0, q_n)}.$$

To estimate $\|F_{N,n} - H_n\|_{L^2(q_0, q_n)}$, we utilize the fact that $F_{N,n}$ and H_n differ only on the first and the last subintervals. Direct computation yields

$$\begin{aligned} \|F_{N,n} - H_n\|_{L^2(q_0, q_n)}^2 &= \frac{1}{3}(q_1 - q_0)(F(q_0) - F_N(q_0))^2 + \frac{1}{3}(q_n - q_{n-1})(F(q_n) - F_N(q_n))^2 \\ &= \frac{1}{3n}(q_n - q_0)(F(q_0) - F_N(q_0))^2 + \frac{1}{3n}(q_n - q_0)(F(q_n) - F_N(q_n))^2. \end{aligned}$$

Thus, we have

$$\lim_{n \rightarrow \infty} \|F_{N,n} - H_n\|_{L^2(q_0, q_n)} = 0.$$

The term $\|H_n - F\|_{L^2(q_0, q_n)}$ represents the L^2 -error of approximating F on (q_0, q_n) by linear interpolation. Given that F is an actual CDF, it is uniformly continuous on $[q_0, q_n]$. Accordingly, for arbitrary $\varepsilon > 0$, we can find $\delta > 0$, such that for all $x, y \in [q_0, q_n]$ satisfying $|x - y| < \delta$, we have $|F(x) - F(y)| < \varepsilon$. Choosing n sufficiently large so that $\max_i(q_i - q_{i-1}) < \delta$, we obtain

$$\|H_n - F\|_{L^2(q_0, q_n)} = \sum_{i=1}^n \int_{q_{i-1}}^{q_i} |H_n(x) - F(x)|^2 dx < \sum_{i=1}^n \int_{q_{i-1}}^{q_i} \varepsilon^2 dx = (q_n - q_0)\varepsilon^2,$$

which implies that

$$\lim_{n \rightarrow +\infty} \|H_n - F\|_{L^2(q_0, q_n)} = 0.$$

Consequently, we obtain

$$\lim_{n, N \rightarrow \infty} \|F - F_{N,n}\|_{L^2(q_0, q_n)} = 0,$$

For the pointwise convergence property, we again invoke the auxiliary function H_n and apply the triangle inequality:

$$|F_{N,n}(x) - F(x)| \leq |F_{N,n}(x) - H_n(x)| + |H_n(x) - F(x)|, \quad \forall x \in (q_0, q_n).$$

Here we can also utilize the absolute continuity of F to guarantee that

$$\lim_{n \rightarrow +\infty} |H_n(x) - F(x)| = 0, \quad \forall x \in (q_0, q_n).$$

The pointwise convergence of $F_{N,n}$ to H_n also follows from the fact that the discrepancy between $F_{N,n}(x)$ and $H_n(x)$ occurs only on the first and last subintervals, whose lengths vanish as $n \rightarrow +\infty$. Thus, for any fixed $x \in (q_0, q_n)$, we may choose n sufficiently large so that x lies in one of the interior subintervals. Putting these together, we obtain the pointwise convergence of $F_{N,n}$ to F on (q_0, q_n) . \square

Notably, most commonly used distributions fall within the class \mathcal{X} . Typical examples include the Gaussian distribution, the exponential and gamma families, the lognormal distribution and the log-skew-normal distribution.

4.2 Propagation capability of the model

Our model is designed not only to approximate individual random variables, but also to preserve their essential characteristics under propagation through a DAG. In particular, the model should remain stable under max and sum operations.

We begin by establishing results that justify the propagation capability in the L^2 sense. To avoid ambiguity, we denote by $F_k^{(l)}$ and $F_k^{(r)}$ the left-tail and right-tail CDFs of X_k for $k = 1, 2$, respectively. Their expressions take the following form:

$$\begin{cases} F_k^{(l)}(x) = \alpha_k^{(l)} \Phi(x; \mu_k^{(l)}, \sigma_k^{(l)}) + \sum_{i=0}^{N_k-1} \beta_{k,i}^{(l)} x^i \phi(x; \mu_k^{(l)}, \sigma_k^{(l)}), \\ F_k^{(r)}(x) = 1 - \alpha_k^{(r)} \left(1 - \Phi(x; \mu_k^{(r)}, \sigma_k^{(r)})\right) - \sum_{i=0}^{N_k-1} \beta_{k,i}^{(r)} x^i \phi(x; \mu_k^{(r)}, \sigma_k^{(r)}), \end{cases} \quad (12)$$

where $\alpha_k^{(l)}$, $\alpha_k^{(r)}$, $\beta_{k,i}^{(l)}$, and $\beta_{k,i}^{(r)}$ are real coefficients for $k = 1, 2$ and $i = 0, 1, \dots, N_k - 1$.

Let $F^{(l)}$ and $F^{(r)}$ denote the left-tail and right-tail CDFs of $X = X_1 \odot X_2$, respectively. For the sum case, $F^{(l)}$ and $F^{(r)}$ are both computed via convolution on the real line and thus share the same expression. Therefore, we omit the superscript (l) and (r) whenever no confusion arises.

Since the family $\{e^{-x^2/2} x^n\}_{n \geq 0}$ forms a complete set in $L^2(\mathbb{R})$ [20, pp. 10, 108], and we have established a constructive L^2 -convergence result for the Hermite function expansion in Lem. 2.1, it suffices to establish the following lemma for L^2 -justification of our tail model.

Lemma 4.2. *Suppose X_1, X_2 are two independent real-valued continuous random variables and have PDFs of the form (7), and let $X = X_1 \odot X_2$ denote either their maximum or sum. Then for any $\delta > 0$, we have*

- (1) $F^{(l)} \in L^2(-\infty, q_0 + \delta)$, where $F^{(l)}$ denotes the left-tail CDF expression of X that is naturally expanded from $(-\infty, q_0)$ onto $[q_0, q_0 + \delta)$.
- (2) $1 - F^{(r)}(x) \in L^2(q_n - \delta, +\infty)$, where $F^{(r)}$ denotes the right-tail CDF expression of X that is naturally expanded from $(q_n, +\infty)$ onto $(q_n - \delta, q_n]$.

In particular, $X \in \mathcal{X}$.

Proof. (1) **Max case.** As $x \rightarrow -\infty$, $F^{(l)}(x) = F_1^{(l)}(x)F_2^{(l)}(x)$. In this case, we have

$$F_1^{(l)}(x)F_2^{(l)}(x) = \prod_{k=1}^2 \left[\alpha_k^{(l)} \Phi(x; \mu_k^{(l)}, \sigma_k^{(l)}) + \sum_{i=0}^{N_k-1} \beta_{k,i}^{(l)} x^i \phi(x; \mu_k^{(l)}, \sigma_k^{(l)}) \right],$$

where

$$\alpha_k^{(l)} \Phi(x; \mu_k^{(l)}, \sigma_k^{(l)}) + \sum_{i=0}^{N_1-1} \beta_{k,i}^{(l)} x^i \phi(x; \mu_k^{(l)}, \sigma_k^{(l)}) \sim \left(-\frac{\alpha_k^{(l)} \sigma_k^{(l)}}{x - \mu_k^{(l)}} + \sum_{i=0}^{N_1-1} \beta_{k,i}^{(l)} x^i \right) \phi(x; \mu_k^{(l)}, \sigma_k^{(l)}), \quad (13)$$

as $x \rightarrow -\infty$, $k = 1, 2$. As the decay of e^{-x^2} suppresses all rational contributions, the decay behavior of $F_1(x)F_2(x)$ is dominated by the Gaussian factor.

Sum case. We utilize the following union-bound-type inequalities:

$$\begin{cases} \mathbb{P}(X_1 + X_2 < x) \leq \mathbb{P}(X_1 < \frac{x}{2}) + \mathbb{P}(X_2 < \frac{x}{2}), \\ \mathbb{P}(X_1 + X_2 > x) \leq \mathbb{P}(X_1 > \frac{x}{2}) + \mathbb{P}(X_2 > \frac{x}{2}). \end{cases}$$

Note that $\mathbb{P}(X_i < \frac{x}{2})$ also shows Gaussian decay as $x \rightarrow -\infty$, $i = 1, 2$, it follows that $F_{X_1+X_2}(x)$ is L^2 -integrable on the left tail.

(2) For the max case, as $x \rightarrow +\infty$, $F^{(r)}(x) = F_1^{(r)}(x)F_2^{(r)}(x)$. In this case, we have

$$1 - F_1^{(r)}(x)F_2^{(r)}(x) = 1 - \prod_{k=1}^2 \left[1 - \alpha_k^{(r)} \left(1 - \Phi(x; \mu_k^{(r)}, \sigma_k^{(r)}) \right) - \sum_{i=0}^{N_k-1} \beta_{k,i}^{(r)} x^i \phi(x; \mu_k^{(r)}, \sigma_k^{(r)}) \right].$$

Similar to the left-tail case, as $\alpha_k^{(r)} \left(1 - \Phi(x, \mu_k^{(r)}, \sigma_k^{(r)}) \right) + \sum_{i=0}^{N_k-1} \beta_{k,i}^{(r)} x^i \phi(x, \mu_k^{(r)}, \sigma_k^{(r)})$ shows Gaussian decay for $k = 1, 2$ as $x \rightarrow +\infty$, their product also shows Gaussian-type decay behavior. Thus, $1 - F(x)$ is L^2 -integrable around $+\infty$.

The proof for the sum case on the right tail follows the same reasoning as in the left-tail analysis. \square

By combining the above lemma with Thm. 4.1, we obtain the following result.

Theorem 4.3 (Propagation capability of the model). *Suppose X_1, X_2 are independent real-valued continuous random variables and have PDFs of the form (7), and let $X = X_1 \odot X_2$ denote either their maximum or sum, and write F for the cumulative distribution function of X . Then there exists $\{f_{N,n}\}$, each element of which follows a PDF of the form (4), such that*

$$\begin{aligned} \lim_{n, N \rightarrow \infty} \left\| F(x) - \int_{-\infty}^x f_{N,n}(t) dt \right\|_{L^2(\mathbb{R})} &= 0, \\ \lim_{n, N \rightarrow \infty} \left(F(x) - \int_{-\infty}^x f_{N,n}(t) dt \right) &= 0, \quad \forall x \in \mathbb{R}. \end{aligned}$$

5 Algorithmic framework for delay graph propagation

In this section, we describe the algorithmic framework related to our delay model for propagating random delays on DAGs. As was discussed in Sec. 3, each random variable to be propagated, whose PDF follows the form (4) with a piecewise constant intermediate segment, can be represented by the following information:

- (1) $\{c_j^{(l)}\}_{j=0}^N, \mu^{(l)}, \sigma^{(l)}$: the left-tail parameters,
- (2) $\{(p_i, F(p_i))\}_{i=0}^n$: the sample points and their CDF values in the intermediate segment,
- (3) $\{c_j^{(r)}\}_{j=0}^N, \mu^{(r)}, \sigma^{(r)}$: the right-tail parameters.

Given a DAG $G = (V, E)$, its topological order, and the incremental delay distribution at each node and along each edge, we aim to compute the arrival time distribution at each node. We assume that N and n are prescribed and remain fixed throughout the propagation. The incremental delays at each node and along each edge are also assumed to have pre-given analytic PDFs and CDFs that can be evaluated on each point on the real line.

Two main algorithmic components are involved in the propagation process:

1. **CDF point sampling.** We need to evaluate the CDF values at arbitrary points on the real line as well as compute the quantile values q_0 and q_n . This is not only required when we obtain the CDF sample points in the intermediate segment, but also when approximating the tail parameters.

2. **Tail parameter fitting.** We employ the Levenberg–Marquardt (LM) algorithm [22] to fit the tail parameters based on the CDF values at several reference points.

In the delay propagation process, there are two scenarios where the two algorithms above are applied:

1. **Input projection.** When a random variable X is first introduced into the graph, it must be projected onto our model form. In this case, the analytic forms of its PDF and CDF are known a priori, allowing us to evaluate the CDF at any point on the real line directly.
2. **Binary operation.** During propagation, max and sum operations are applied to pairs of random variables. This is carried out in a *stage* manner. A *stage* in the STA process typically refers to the timing arc from the input pin of a cell to the input pin of one of its fanout cells, which includes a cell delay arc and an interconnect delay arc. In this work, we adopt the simplified model in which all input-to-output paths of a cell share the same delay, i.e., a single node delay, although cases with distinct cell-delay arcs can be handled in the same way. As our model supports only binary operations, we here index the nodes and process in algebraic order by default. In practical scenarios, one may proceed in the order that the inputs arrive.

Notably, different initialization strategies are used for LM-based tail fitting in the two scenarios described above; these strategies are detailed in Subsec. 5.2.

5.1 CDF point sampling

To determine the quantile values q_0 and q_n , we first apply the adaptive bracket expansion algorithm [23, Chapter 9]. Starting from an initial guess x_{init} , we evaluate the CDF value $F(x_{\text{init}})$ and determine whether the target CDF lies to its left or right. The interval is then expanded in the corresponding direction with an exponentially increasing step size until the function values at the two endpoints bracket the desired value. Once such a pair is identified, the standard bisection method is applied. After the interval $[l, r]$ becomes sufficiently small, linear interpolation is used to obtain the final solution. The full procedure is summarized in Alg. 1.

Algorithm 1 Adaptive bracket expansion and bisection for computing quantile values

Require: CDF $F(\cdot)$, target CDF value $y \in (0, 1)$, initial guess x_0 , initial step size $\Delta x > 0$, expansion factor η , tolerance $\varepsilon > 0$

Ensure: Approximate solution x satisfying $|F(x) - y| \leq \varepsilon$

```

1:  $F_0 \leftarrow F(x_0)$ 
2: if  $F_0 = y$  then
3:   return  $x_0$ 
4: end if
5:  $s \leftarrow \text{sign}(y - F_0)$ 
6: repeat
7:    $x_1 \leftarrow x_0 + s \Delta x$ ,  $F_1 \leftarrow F(x_1)$ 
8:   if  $(F_1 - y)(F_0 - y) > 0$  then
9:      $x_0 \leftarrow x_1$ ,  $F_0 \leftarrow F_1$ ,  $\Delta x \leftarrow \eta \Delta x$ 
10:  end if
11: until  $(F_1 - y)(F_0 - y) \leq 0$ 
12:  $l \leftarrow \min(x_0, x_1)$ ,  $r \leftarrow \max(x_0, x_1)$ 
13: while  $r - l > \varepsilon$  do
14:    $m \leftarrow (l + r)/2$ 
15:   if  $F(m) \leq y$  then
16:      $l \leftarrow m$ 
17:   else
18:      $r \leftarrow m$ 
19:   end if
20: end while
21: return  $\frac{(F(r) - y)l + (y - F(l))r}{F(r) - F(l)}$ 

```

After obtaining q_0 and q_n , we uniformly sample $n - 1$ more points in $[q_0, q_n]$ and evaluate the CDF values at these points. The full procedure is summarized in Alg. 2.

If a closed-form expression for the input distribution is available and enables direct evaluation of quantiles, this sampling step can be omitted. For example, in many practical settings, delays are modeled as Gaussian or lognormal random variables, for which quantiles can be computed analytically. Moreover, once a random variable has been projected onto our model, one may also recover a parametric distribution using moment matching based on numerically estimated moments.

When evaluating the CDF at the desired points, two different scenarios arise. If the distribution admits a known analytic expression, the values can be computed directly. Otherwise, numerical evaluation is required. In our framework, the max operation falls into the former category, whereas the sum operation belongs to the latter. For the numerical evaluation of the convolution integral in the sum case, the integral is decomposed into three regions: the left tail, the intermediate segment, and the right tail. In each region, rather than relying entirely on general-purpose numerical integrators such as GSL's `qagi` [24] or Boost's `sinh_sinh`, we exploit the polynomial-Gaussian tail structure and the piecewise linear approximation of the CDF in the intermediate segment to obtain closed-form expressions whenever possible, and resort to numerical quadrature methods only for the remaining parts.

Algorithm 2 CDF point sampling

Require: Target CDF values y_0, y_n , number of CDF points n , the CDF mapping F

Ensure: $\{(q_i, F(q_i))\}_{i=0}^n$

- 1: Apply Alg. 1 to find q_0 and q_n such that $F(q_0) = y_0, F(q_n) = y_n$
 - 2: **for** $i = 1$ to $n - 1$ **do**
 - 3: $q_i \leftarrow q_0 + \frac{i}{n}(q_n - q_0)$
 - 4: Compute $F(q_i)$ according to the analytic or numerical method
 - 5: **end for**
-

5.2 Tail parameter fitting

After obtaining the CDF values at the reference points, we proceed to fit the tail parameters. If the input distribution is Gaussian, we directly set the tail parameters accordingly and skip the fitting step.

Although we have already illustrated the convergence properties of the Hermite function expansion for our model in Sec. 4, we do not adopt this expansion in practice. The reasons are twofold. First, computing Hermite coefficients is numerically expensive, especially for the sum operation, which requires evaluating integrals over the entire real line. Second, the intermediate segment lacks sufficient smoothness to guarantee any convergence rate estimate, potentially leading to slow convergence in practice.

For a general tail-fitting problem, the determination of the tail parameters is formulated as a nonlinear least-squares (NLS) problem and we solve it by the Levenberg–Marquardt (LM) algorithm. In the following discussion, we focus on the left-tail case; the corresponding right-tail procedure is presented in Appendix A. Let $\{x_i, y_i\}_{i=1}^k$ denote the reference CDF points, where $y_i = F(x_i)$. These points do not necessarily satisfy $x_i < q_0$ for all $i = 1, \dots, k$. The CDF values y_i at these points can be numerically computed. The selection of reference points is flexible in practice. In our implementation, we choose k equally spaced points in the interval $[F^{-1}(q_0/4), F^{-1}(2q_0)]$ for the left tail, where F^{-1} denotes the quantile function of the target distribution. The right tail reference points are chosen in a similar manner. The target function in the LM algorithm is set as

$$\min \sum_{i=1}^k \left[\left(\tilde{F}(x_i; \{c_j^{(l)}\}_{j=0}^N, \mu^{(l)}, \sigma^{(l)}) - y_i \right) / y_i \right]^2 + \lambda (c_0^{(l)} - 1)^2 + \lambda \left(\sum_{j=1}^N (c_j^{(l)})^2 \right), \quad (14)$$

where the function \tilde{F} is the approximate left-tail CDF that can be directly computed from the parameters $\{c_j^{(l)}\}_{j=0}^N, \mu^{(l)}$, and $\sigma^{(l)}$.

To balance the contributions of reference points across the tail region, we adopt the relative error formulation in Eq. (14), which assigns comparable importance to points near the tail and those moderately farther away. A regularization term is introduced to prevent overfitting. Without it, the optimization may produce excessively large polynomial coefficients, which in turn can make the propagation process unstable.

Since the Jacobian matrix J can be directly derived and evaluated in our model, we can utilize the LM algorithm, which solves

$$(J^T J + \lambda_{\text{LM}} I) \delta x = -J^T r,$$

where λ_{LM} is a damping factor, I is the identity matrix and r is the residual vector defined by

$$r_i = \begin{cases} \frac{y_i - \tilde{F}(x_i; \{c_j^{(l)}\}_{j=0}^N, \mu^{(l)}, \sigma^{(l)})}{y_i}, & i = 1, \dots, k, \\ \sqrt{\lambda} c_{i-k-1}^{(l)}, & i = k+1, \dots, k+N+1. \end{cases}$$

To impose the constraint that $\sigma^{(l)} > 0$, we reparameterize the scale as $p_\sigma = \log(\sigma^{(l)})$. This transformation keeps the objective and residual structure intact but alters the Jacobian accordingly.

As an illustrative example, consider the case $N = 2$. For simplicity, superscripts are omitted and the left-tail PDF is written as

$$\tilde{f}(x) = (\alpha + \beta x + \gamma x^2) \phi(x; \mu, \sigma). \quad (15)$$

The corresponding left-tail CDF is given by

$$\tilde{F}(x) = (\beta\mu + \alpha + \gamma(\mu^2 + \sigma^2)) \Phi(x; \mu, \sigma) - (\beta + \gamma x + \gamma\mu)\sigma^2 \phi(x; \mu, \sigma).$$

The scale parameter is reparameterized as $\sigma = \exp(p_\sigma)$ to ensure positivity during optimization. Correspondingly, the optimization variable is $p_\sigma = \log(\sigma)$.

Let $\Phi_i = \Phi(x_i; \mu, \sigma)$, $\phi_i = \phi(x_i; \mu, \sigma)$ and $z_i = (x_i - \mu)/\sigma$, then the i -th row of the Jacobian matrix J is given by

$$J(i, :) = -\frac{1}{y_i} \begin{bmatrix} \mu \Phi_i - \sigma^2 \phi_i \\ \Phi_i \\ (\mu^2 + \sigma^2) \Phi_i - (x_i + \mu) \sigma \phi_i \\ (\beta + 2\gamma\mu) \Phi_i - \left(\alpha + \beta\mu + \gamma(\mu^2 + \sigma^2) + \gamma\sigma^2 + \frac{(x_i - \mu)(\beta + \gamma(x_i + \mu))}{\sigma} \right) \sigma \phi_i \\ \sigma \left[2\gamma \Phi_i + (-(\beta + \gamma(x_i + \mu))(z_i^2 + 1) - \frac{z_i}{\sigma} (\beta\mu + \alpha + \gamma(\mu^2 + \sigma^2))) \sigma \phi_i \right] \end{bmatrix}^\top,$$

for $i = 1, \dots, k$.

The regularization term contributes $N + 1$ additional rows to the Jacobian matrix. For $N = 2$, the last three rows are

$$\begin{aligned} J(k+1, :) &= [\sqrt{\lambda}, 0, 0, 0, 0], \\ J(k+2, :) &= [0, \sqrt{\lambda}, 0, 0, 0], \\ J(k+3, :) &= [0, 0, \sqrt{\lambda}, 0, 0]. \end{aligned}$$

This ensures that minimizing the total squared residual norm corresponds exactly to the regularized objective function in Eq. (14).

Putting everything together, we obtain Alg. 3 for fitting the tail parameters.

Algorithm 3 Tail parameter fitting via the Levenberg–Marquardt algorithm

Require: Reference points $\{(x_i, y_i)\}_{i=1}^k$, polynomial degree N , regularization weight λ

Ensure: Fitted parameters $x = (c_0, c_1, \dots, c_N, \mu, \sigma)^\top$

- 1: Initialize μ, σ depending on the operation type and tail side
 - 2: Reparameterize σ as $p_\sigma = \log(\sigma)$
 - 3: Initialize coefficients c_0, c_1, \dots, c_N .
 - 4: **while** not converged **do**
 - 5: **for** $i = 1$ to k **do**
 - 6: Compute residual $r_i = \frac{\tilde{F}(x_i) - y_i}{y_i}$
 - 7: **end for**
 - 8: **for** $j = 0$ to N **do**
 - 9: Add regularization residual $r_{k+j+1} = \sqrt{\lambda} \cdot c_j$
 - 10: **end for**
 - 11: Compute the Jacobian matrix J
 - 12: Solve LM step: $(J^\top J + \lambda_{\text{LM}} I) \delta x = -J^\top r$
 - 13: Update $x \leftarrow x + \delta x$, with $\sigma = \exp(p_\sigma)$ and λ_{LM}
 - 14: **end while**
-

It is noteworthy that we do not impose the constraint that the CDF values at the quantile points q_0 and q_n must remain fixed when formulating the NLS problem. Since our primary objective is to achieve an accurate overall fit across the tail region, allowing small deviations at q_0 and q_n has negligible impact on the resulting approximation.

As the LM algorithm only ensures local convergence, we aim to provide a good initialization to achieve reliable and stable optimization performance.

During input projection, the tail parameters are initialized by fitting either the left or right tail to a Gaussian via least squares. Given sample points $\{(x_i, y_i)\}_{i=1}^k$, we compute $z_i = \Phi^{-1}(y_i)$ and then fit $x_i \approx \mu + \sigma z_i$ by solving

$$\min_{\mu, \sigma} \sum_{i=1}^k (x_i - \mu - \sigma z_i)^2,$$

which admits the following closed-form solution:

$$\begin{cases} \sigma = \frac{\sum_{i=1}^k (z_i - \bar{z})(x_i - \bar{x})}{\sum_{i=1}^k (z_i - \bar{z})^2}, \\ \mu = \bar{x} - \sigma \bar{z}, \end{cases} \quad \text{where} \quad \bar{x} = \frac{1}{k} \sum_{i=1}^k x_i, \quad \bar{z} = \frac{1}{k} \sum_{i=1}^k z_i.$$

The initialization of the tail parameters in binary operations differs from that used in input projection. Since the target distribution is now obtained through either a max or a sum operation applied to random variables of a prescribed form, additional information about the asymptotic behavior of the tails becomes available. This allows us to devise more informed and effective initialization strategies.

When initializing for the sum operation, we utilize the closure of Gaussian random variables under sum operations. For the left tail, we initialize μ and σ in Eq. (15) as

$$\mu = \mu_1^{(l)} + \mu_2^{(l)}, \quad \sigma^2 = (\sigma_1^{(l)})^2 + (\sigma_2^{(l)})^2.$$

The same initialization strategy is used for the right tail.

When initializing for the max operation, it is important to observe that the condition $F_1 F_2 \rightarrow 0$ does not necessarily imply that both F_1 and F_2 are simultaneously small. Hence, the initialization of the left tail should distinguish between the following two cases:

1. If the values of F_1 and F_2 differ significantly at the chosen reference points, one of them may remain of non-negligible magnitude even when the other approaches zero. In this case, the left-tail initialization is taken from the random variable whose CDF values are smaller at those points.
2. If the values of F_1 and F_2 are comparable at the reference points, we assume that both F_1 and F_2 show asymptotic behaviors. The left-tail exact PDF of the sum should satisfy:

$$f^{(l)}(x) = F_1^{(l)}(x)f_2^{(l)}(x) + f_1^{(l)}(x)F_2^{(l)}(x).$$

Suppose that the input PDFs are of the form $f_i^{(l)}(x) = p_i(x)\phi(x; \mu_i^{(l)}, \sigma_i^{(l)})$, where $i = 1, 2$, and $p_i(x)$ are polynomials. As $x \rightarrow -\infty$, we can see from Eq. (13) that $F_1^{(l)}$ and $F_2^{(l)}$ can both be seen as the similar form, since the fraction term in the bracket shows a decay behavior as $x \rightarrow -\infty$ and can be omitted. Thus we write

$$F_i^{(l)}(x) \sim q_i(x)\phi(x; \mu_i^{(l)}, \sigma_i^{(l)}), \quad i = 1, 2, \quad \text{as } x \rightarrow -\infty,$$

for some polynomials $q_i(x)$. Completing the square in the quadratic exponent of the product $\phi(x; \mu_1^{(l)}, \sigma_1^{(l)})\phi(x; \mu_2^{(l)}, \sigma_2^{(l)})$ yields the initialization

$$\begin{cases} \mu^{(l)} = \left(\frac{\mu_1^{(l)}}{(\sigma_1^{(l)})^2} + \frac{\mu_2^{(l)}}{(\sigma_2^{(l)})^2} \right) (\sigma^{(l)})^2, \\ \sigma^{(l)} = \left((\sigma_1^{(l)})^{-2} + (\sigma_2^{(l)})^{-2} \right)^{-1/2}. \end{cases}$$

We adopt the following heuristic criterion for determining when the values of F_1 and F_2 “differ significantly” as:

$$F_1(q_{1,0}) > F_2(q_{2,n}) \quad \text{or} \quad F_2(q_{2,0}) > F_1(q_{1,n}).$$

Right-tail initialization is comparatively simpler, because both F_1 and F_2 are close to 1 when $F_1 \cdot F_2 \rightarrow 1$. The right-tail PDF behavior of $\max\{X_1, X_2\}$ is dominated by the component of the larger $\sigma^{(r)}$ value, whose decay rate is slower. In practice, however, the reference points used for tail fitting do not lie sufficiently far into the extreme tail. Consequently, the decaying behavior among these points may not fully reflect the asymptotic characteristics.

Therefore, rather than relying completely on the asymptotic behavior, we adopt a heuristic Gaussian-based initialization strategy as follows:

$$\begin{cases} \mu^{(r)} = \max(\mu_1^{(r)}, \mu_2^{(r)}), \\ \sigma^{(r)} = \frac{1}{3} \left(\max\{\mu_1^{(r)} + 3\sigma_1^{(r)}, \mu_2^{(r)} + 3\sigma_2^{(r)}\} - \mu^{(r)} \right). \end{cases}$$

After determining the initial values of μ and σ , the polynomial coefficients are initialized via a least-squares fitting method. Evaluating the PDF at the reference points gives the values $\{p(x_i)\}_{i=0}^k$, and the coefficients are obtained by applying `polyfit` to solve

$$\min_{p \in P_N} \sum_{i=1}^k \left[p(x_i) - \frac{\tilde{f}(x_i)}{\phi(x_i; \mu, \sigma)} \right]^2. \quad (16)$$

After each binary operation, the values of \tilde{F} at q_0 and q_n may deviate from the target values $F(q_0)$ and $F(q_n)$. But we keep these values unchanged, as such deviations has limited impact in the propagation process and are consistent with our theoretical results.

5.3 Overall workflow

We summarize below the overall workflow that integrates all components discussed in this section into a complete delay propagation procedure on a DAG.

Given a DAG and all the random variables representing node delays and edge delays, we first perform *input projection* for each input random variable, converting its distribution into left-tail parameters, right-tail parameters, and CDF sample points in the intermediate segment. We then topologically sort the graph to determine the propagation order. Starting from the source node(s), propagation proceeds stage by stage along this order, during which the required binary operations are carried out.

At each node, we first obtain its arrival time by combining the outputs of all its predecessor nodes through successive max operations. The node delay is then added via a single sum operation to produce the node's output delay. Finally, the corresponding edge delays are added when passing this output delay for further propagation.

To complement this workflow, Fig. 1 illustrates a complete propagation stage on a simple DAG consisting of two source nodes and one sink node. For each incoming edge $(i, 3)$, the edge delay $d_{i,3}$ is first projected onto our model space, yielding $\theta_{i,3}$. The predecessor node delays d_1 and d_2 are also projected to θ_1 and θ_2 . Each pair $(\theta_i, \theta_{i,3})$ is combined through a sum operation to produce the arrival distributions $\theta_{3,in1}$ and $\theta_{3,in2}$. A max operation merges these two candidates into $\theta_{3,in}$, after which the node delay d_3 is projected to θ_3 and added, giving the output distribution $\theta_{3,out}$.

6 Numerical results

In this section, we present numerical experiments to validate our proposed model and algorithms. The experiments are designed to assess both the approximation accuracy of the three-segment representation and its stability under repeated sum and max operations on DAGs. In particular, we focus on the accuracy of tail quantiles, which are of primary interest in STA.

6.1 Experiment setup

To evaluate the performance of our proposed algorithm in Sec. 5 and to assess the feasibility of our proposed model in Sec. 3, we conducted numerical experiments under different scenarios. Throughout all experiments, we set the number of CDF sample points $n = 100$, the degree of the tail polynomials $N = 2$. On each tail, $k = 21$ points are used for tail fitting, and the left and right quantiles are respectively set to 0.135% and 99.865%, corresponding to the $\pm 3\sigma$ points of a standard Gaussian distribution. To focus on the behavior of the propagation algorithm itself, we adopt a simplified setting. Instead of

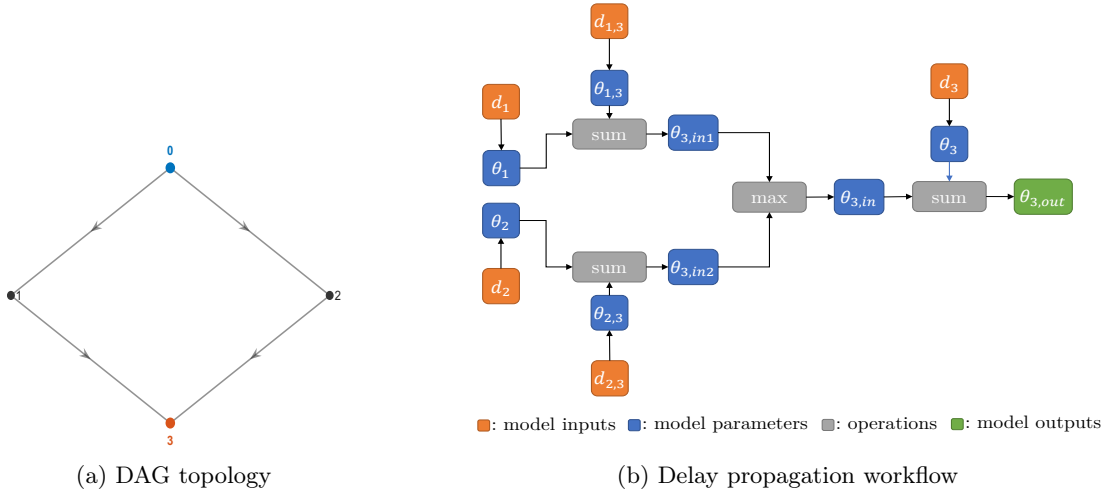


Figure 1: Illustration of delay propagation with a three-node DAG

letting delays at later stages depend on those of preceding stages, we assume that all node and edge delays in the graph are predetermined random variables. In other words, all delay information required for propagation is assumed to be available from the beginning.

We consider three experimental scenarios, each associated with a target distribution used as the primary reference for comparison. In graph propagation experiments, the distribution of the sink node is taken as the distribution of interest. To quantitatively assess the accuracy of different methods, we compare the key quantiles of our desired distributions, including the 0.135%, 1%, 99% and 99.865% quantiles, among which the 0.135% and 99.865% quantiles are of particular importance. The quantile errors are defined as their relative error compared to the ground truth, i.e.,

$$\text{Err}(q) = \frac{|q - q_{\text{true}}|}{q_{\text{true}}}, \quad (17)$$

where q_{true} denotes the ground-truth value.

For each scenario, we compare some of, if not all, the ground truth, the output of our propagation model, the Gaussian moment-matching results whose moments are derived from the ground truth, and an additional Gaussian-based approximation described below.

The additional baseline is a simple quantile-preserving method that provides a fast but crude delay-propagation approximation. For the sum operation, it exploits the closure of independent Gaussian random variables:

$$\begin{cases} \mu_{\text{sum}} &= \mu_1 + \mu_2, \\ \sigma_{\text{sum}} &= \sqrt{\sigma_1^2 + \sigma_2^2}. \end{cases}$$

For the max operation, this method computes

$$\begin{cases} \mu_{\text{max}} &= \max\{\mu_1, \mu_2\}, \\ \sigma_{\text{max}} &= \frac{1}{3}(\max\{\mu_1 + 3\sigma_1, \mu_2 + 3\sigma_2\} - \mu_{\text{max}}). \end{cases}$$

We refer to this method as the *Gaussian quantile-preserving method*.

For each case, the ground truth is obtained either by Monte Carlo simulation or directly from the analytical form of the underlying distribution class. Although our model is inherently CDF-oriented, we compare both the PDF and the CDF of the methods to provide a more comprehensive evaluation.

To ensure the effectiveness of our proposed model, we compare it against two alternative tail construction strategies, while retaining the same representation in the intermediate segment. The first strategy defines the left-tail PDF as

$$f(x) = \begin{cases} 0, & x < \frac{F(q_1)q_0 - F(q_0)q_1}{F(q_1) - F(q_0)}, \\ \frac{F(q_1) - F(q_0)}{q_1 - q_0}, & \frac{F(q_1)q_0 - F(q_0)q_1}{F(q_1) - F(q_0)} \leq x < q_0. \end{cases}$$

This construction corresponds to linearly extrapolating the CDF on $[q_0, q_1]$ towards $-\infty$ until the extrapolated line reaches 0. Similarly, the right tail is constructed by linearly extending the CDF on $[q_{n-1}, q_n]$ towards $+\infty$ until it reaches 1. We refer to this approach as the *extrapolation method*.

The second strategy adopts a different idea by introducing two fixed sentinel points q_{-1} and q_{n+1} , where $F(q_{-1}) = 0$ and $F(q_{n+1}) = 1$. These points serve as global boundaries for all distributions in the graph. The left-tail CDF is obtained by linearly interpolating between $(q_{-1}, 0)$ and $(q_0, F(q_0))$, while the right-tail CDF is obtained by interpolating between $(q_n, F(q_n))$ and $(q_{n+1}, 1)$. We refer to this strategy as the *sentinel method*.

6.2 Lognormal projection

In this scenario, we evaluate the performance of our projection algorithm on a lognormal random variable $X \sim \text{LN}(0, 1)$. The PDF of $\text{LN}(0, 1)$ is given by

$$f_{\text{LN}}(x) = \frac{1}{x\sqrt{2\pi}} \exp\left(-\frac{(\ln x)^2}{2}\right), \quad x > 0.$$

The ground truth is set as the exact PDF and CDF of $\text{LN}(0, 1)$, and our model is used to fit the corresponding tail parameters. The Gaussian moment-matching approximation $\mathcal{N}(\mu_{\text{LN}}, \sigma_{\text{LN}}^2)$ is given by

$$\mu_{\text{LN}} = \exp\left(\mu + \frac{\sigma^2}{2}\right), \quad \sigma_{\text{LN}}^2 = [\exp(\sigma^2) - 1] \exp(2\mu + \sigma^2),$$

The results are displayed in Fig. 2. The results demonstrate that our model provides an excellent

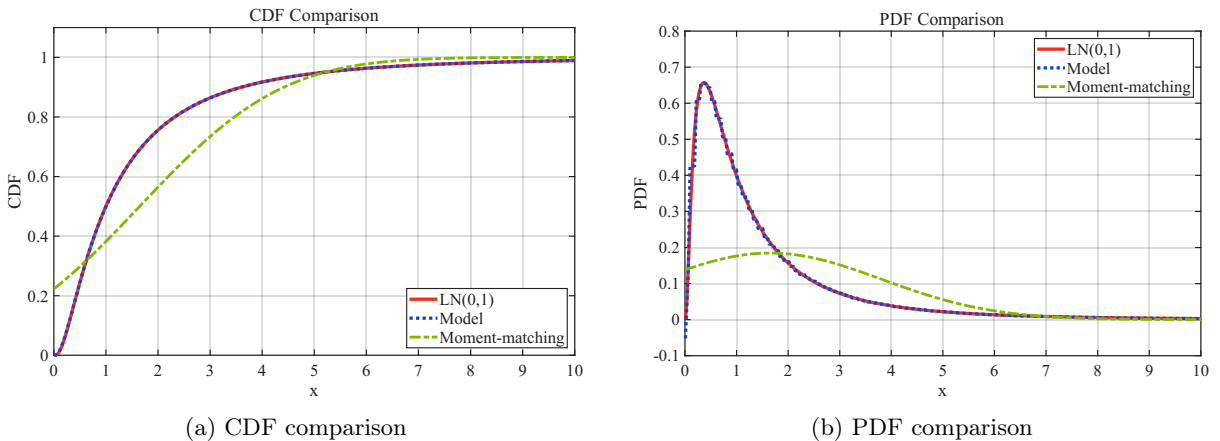


Figure 2: Comparison of (a) CDF and (b) PDF between $\text{LN}(0, 1)$, our model results, and Gaussian results using the moment-matching method.

approximation to the ground truth across the entire distribution. Although some deviations appear in the PDF comparison, these differences have minimal impact on quantile propagation and are therefore acceptable. In contrast, the Gaussian moment-matching approximation exhibits substantial errors, as expected from the strong skewness of the lognormal distribution. This demonstrates that our model effectively captures higher-order moment information and accurately projects lognormal inputs into our model space.

6.3 Inverted binary tree propagation

To evaluate the performance of our algorithm in delay propagation, we construct an inverted binary tree of 5 levels. The node 0 at level 0 is the source of the graph, where no input delay is imposed. It distributes delays to the nodes at level 1. At each node starting from level 2, the output distribution is obtained by taking the maximum of the two incoming distributions and then adding its own incremental delay. This construction guarantees that, at every merge point in the graph, the incoming delays are independent. The structure of the graph is shown in Fig. 3.

In this experiment, each edge delay is modeled as a Gaussian random variable, whose mean μ is uniformly sampled from $[3.0, 5.0]$ and standard deviation σ is uniformly sampled from $[1.5, 2.0]$.

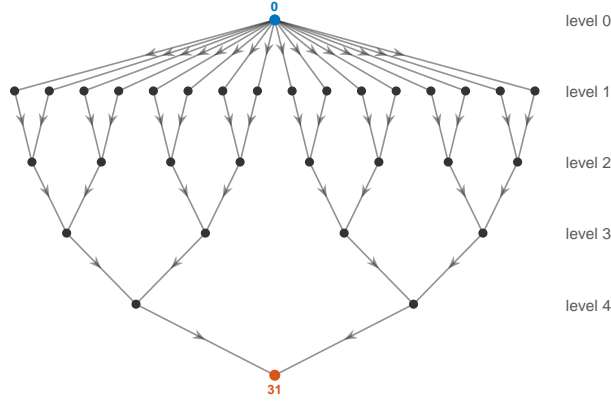


Figure 3: Inverted binary tree structure

Our primary interest is the distribution at the sink node, i.e., node 31. The ground truth is obtained by a Monte Carlo simulation with 10^7 samples.

The resulting CDF and PDF comparisons are presented in Fig. 4, and the corresponding quantile errors are listed in Table 1.

Table 1: Quantile comparison for the inverted binary tree case.

Method	0.135%	1%	99%	99.865%
Model	-0.028%	-0.008%	0.009%	-0.019%
Extrapolation	0.228%	0.023%	-0.484%	-1.489%
Gaussian moment-matching	-2.066%	-1.158%	-0.991%	-1.798%
Gaussian quantile-preserving	-31.928%	-26.540%	-5.240%	-4.010%

It can be observed from the CDF graph that the Gaussian quantile-preserving method exhibits substantial errors. This is expected, as each max operation within the Gaussian framework would introduce significant model bias, and these errors accumulate during the propagation process. Even when a Gaussian approximation is obtained directly from the Monte Carlo distribution via moment matching, the resulting error remains noticeably larger than that of our model.

Regarding the tail models, the sentinel method fails to produce a reasonable approximation, whereas the extrapolation method performs comparably to our model, though with noticeably larger tail errors. Examination of the PDF error curves further reveals that our model exhibits better approximation behavior, particularly in the tails. This indicates our model's better capability in capturing tail characteristics than linear tail models.

The high-frequency fluctuations in the PDF error curves are mainly due to the numerical noise introduced by Monte Carlo simulations and the model error we introduced in approximating the intermediate segment PDF as piecewise constants. If higher-degree approximations are employed in the intermediate segment, the magnitude of these oscillations is expected to decrease accordingly.

6.4 Ladder graph propagation

To further evaluate the performance of our algorithm on deeper graph structures, we consider a ladder graph with 20 levels. This setup is analogous to that in [14], where this graph was used to assess the Gaussian RBF model. In the Gaussian RBF model, the PDF of each delay distribution is approximated by a mixture of Gaussian basis functions as

$$f(x) = \sum_{i=1}^M \omega_i \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right).$$

On the i -th node, where $i = 1, \dots, 20$, we compute the output of this node as

$$X_i = \max\{X_{i-1} + d_{i-1,i}, \xi_i\} + d_i,$$

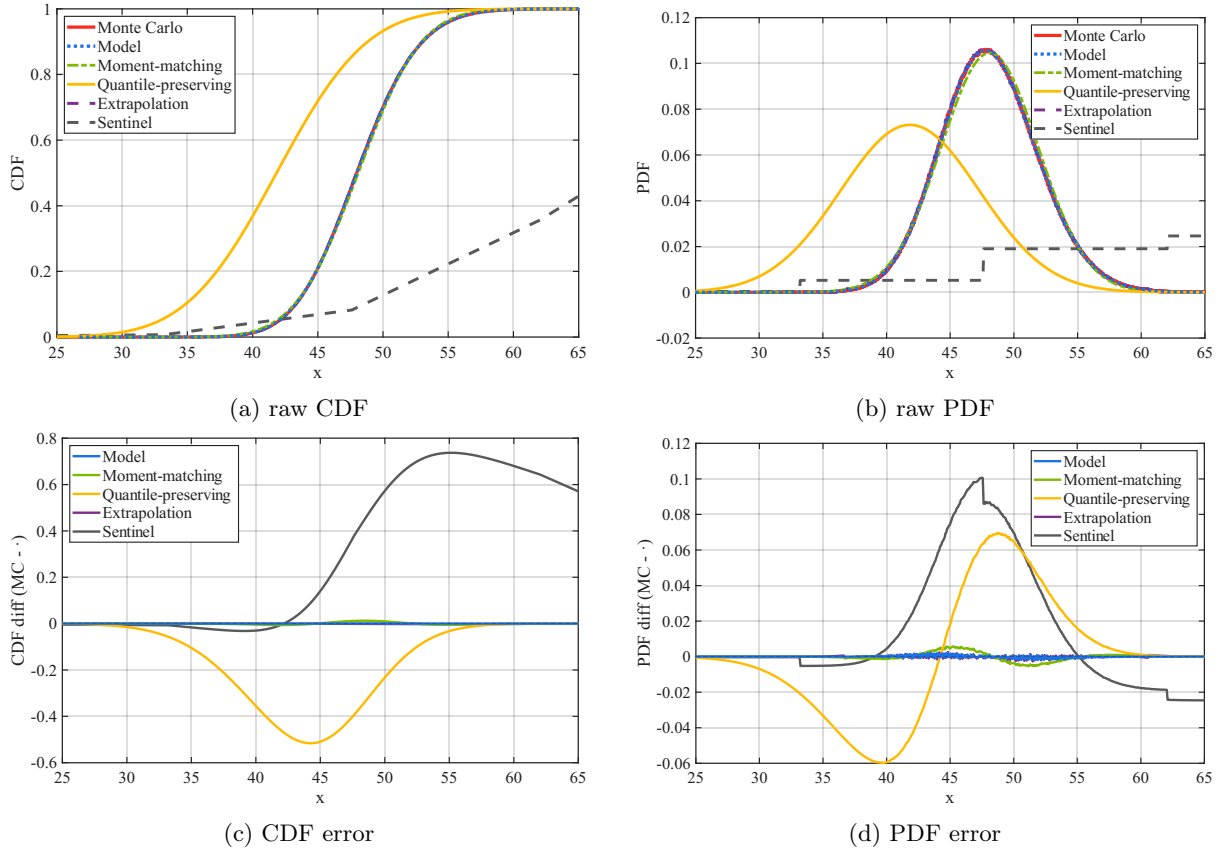


Figure 4: Delay propagation on an inverted binary tree structure: comparison between Monte Carlo simulation, our model results, the Gaussian moment-matching method, the Gaussian quantile-preserving method, the extrapolation method, and the sentinel method.

where X_i denotes the output delay at node i , $d_{i-1,i}$ denotes the delay from node $i-1$ to node i , d_i denotes the incremental delay on node i , and ξ_i represents the input delay of node i , which can be seen as the delay on the edge $(0, i)$. The ladder graph structure is displayed in Fig. 5.

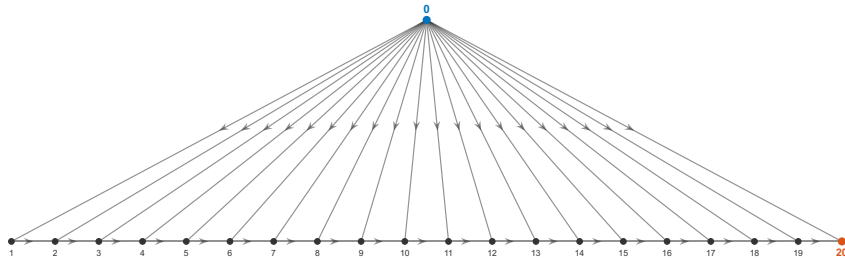


Figure 5: Ladder structure

To introduce more variability in this scenario, the input delay on the i -th node ξ_i is set as a Gaussian random variable whose mean and standard deviation are drawn from $\mathcal{U}(i, 3i)$ and $\mathcal{U}(1.5\sqrt{i}, 2.5\sqrt{i})$, respectively. Each edge delay of $\{d_{i-1,i}\}$ and each incremental node delay $\{d_i\}$ is also modeled as a Gaussian random variable with mean sampled from $\mathcal{U}(1.0, 3.0)$ and standard deviation sampled from $\mathcal{U}(1.5, 2.5)$. Our primary focus is the output delay at the final node, i.e., X_{20} . The ground truth is obtained by a Monte Carlo simulation with 10^7 samples.

The resulting PDF and CDF comparisons are presented in Fig. 6, and the corresponding quantile errors are summarized in Table 2.

The results for the ladder graph exhibit trends similar to those observed in the inverted binary tree, though the greater number of propagation stages in the ladder graph leads to additional error accumulation. While our model does not outperform the extrapolation method at the 1% quantile,

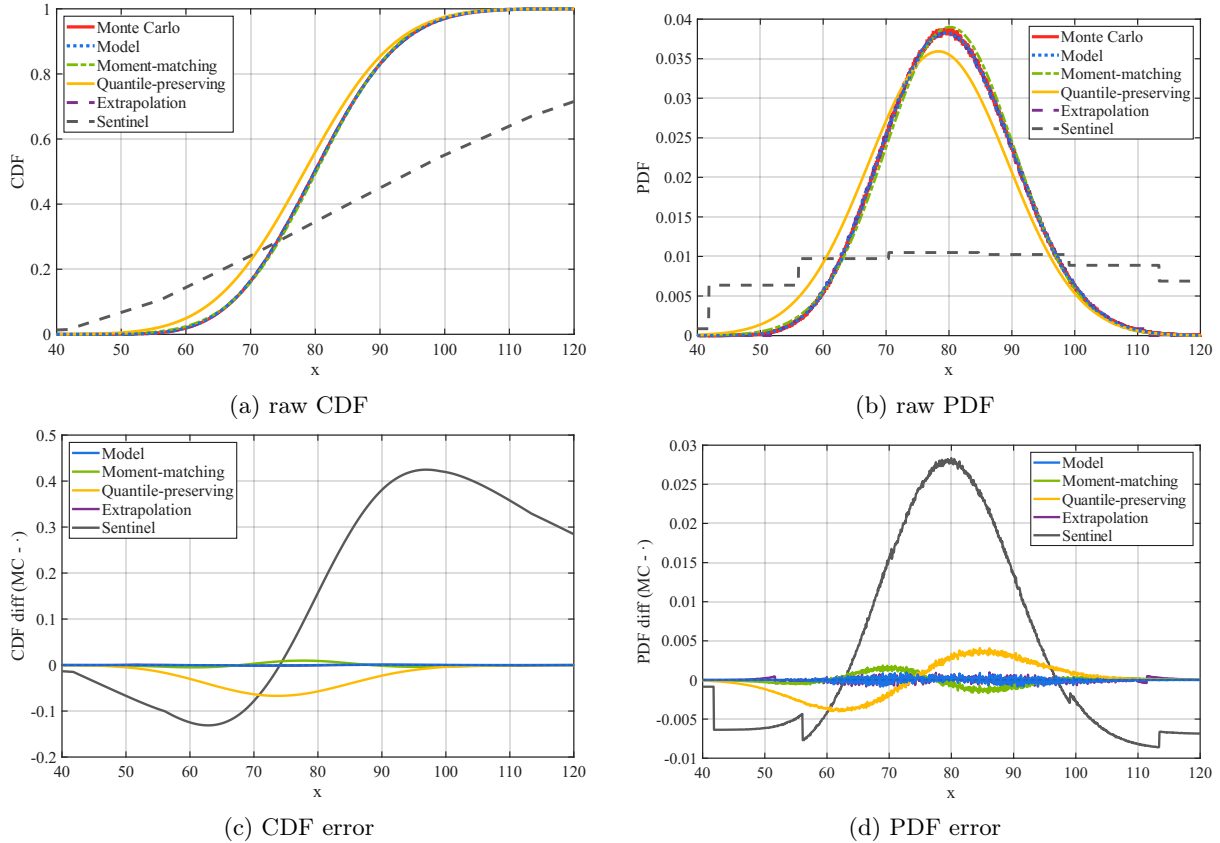


Figure 6: Delay propagation on a ladder structure: comparison between Monte Carlo simulation, our model results, the Gaussian moment-matching method, the Gaussian quantile-preserving method, the extrapolation method, and the sentinel method.

Table 2: Quantile comparison for the ladder case.

Method	0.135%	1%	99%	99.865%
Model	-0.124%	-0.112%	0.042%	0.008%
Extrapolation	1.421%	0.025%	-0.321%	-1.985%
Gaussian moment-matching	-5.680%	-2.510%	-0.764%	-1.116%
Gaussian quantile-preserving	-14.180%	-9.202%	-0.576%	-0.415%

it remains more robust overall, particularly at the 0.135% and 99.865% quantiles, which are the most critical quantiles.

The CDF error patterns further mirror those in the inverted binary tree experiment. Although the extrapolation method yields comparable CDF curves, its tail errors exhibit pronounced peaks, whereas our model maintains a smoother and more stable error profile.

From this case and the inverted binary tree case, it can be concluded that our model and algorithm possess strong propagation capability across graph structures, and that our tail model has better approximation power than those reference linear tail models.

7 Conclusion

In this work, we proposed a three-segment distribution model for stochastic delay propagation on directed acyclic graphs, combining polynomial-Gaussian tail representations with piecewise polynomial approximation in the intermediate segment. We established both L^2 -convergence and pointwise convergence results, and developed a propagation framework based on CDF sampling and Levenberg–Marquardt tail fitting. Numerical experiments on several graph structures demonstrated the accuracy and robustness of the proposed approach.

Several directions remain for future investigation. Although the numerical computation of the convolution in the sum operation has been optimized, it remains a computational bottleneck. Moreover, our current framework supports only binary operations, so extending it to multi-input settings is a natural next step. Finally, the independence assumption may not hold in practical scenarios; incorporating dependency information, for example through correlation-aware propagation in the spirit of [6], is another important direction.

References

- [1] Jayaram Bhasker and Rakesh Chadha. *Static timing analysis for nanometer designs: A practical approach*. Springer Science & Business Media, 2009.
- [2] Adam Bosák, Dmytro Mishagli, and Jakub Mareček. Statistical static timing analysis via modern optimization lens: I. histogram-based approach. *Optimization and Engineering*, 25(3):1405–1429, 2024.
- [3] Charles E Clark. The greatest of a finite set of random variables. *Operations Research*, 9(2):145–162, 1961.
- [4] M. R. C. M. Berkelaar. Statistical delay calculation, a linear time method. In *Proceedings of the 1997 ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 15–24, 1997.
- [5] Shuji Tsukiyama, Masakazu Tanaka, and Masahiro Fukui. A statistical static timing analysis considering correlations between delays. In *Proceedings of the 2001 Asia and South Pacific Design Automation Conference*, pages 353–358, 2001.
- [6] Kunhyuk Kang, Bipul C Paul, and Kaushik Roy. Statistical timing analysis using leveled covariance propagation. In *Design, Automation and Test in Europe*, pages 764–769. IEEE, 2005.
- [7] Kaviraj Chopra, Bo Zhai, David Blaauw, and Dennis Sylvester. A new statistical max operation for propagating skewness in statistical timing analysis. In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pages 237–243, 2006.
- [8] Marwane Ben Hcine and Ridha Bouallegue. Fitting the log skew normal to the sum of independent lognormals distribution. *arXiv preprint arXiv:1501.02344*, 2015.
- [9] Lawrence Fenton. The sum of log-normal probability distributions in scatter transmission systems. *IRE Transactions on communications systems*, 8(1):57–67, 1960.
- [10] Leilei Jin, Wenjie Fu, Hao Yan, and Longxing Shi. A statistical cell delay model for estimating the 3σ delay by matching kurtosis. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(6):2932–2936, 2022.
- [11] Lerong Cheng, Jinjun Xiong, and Lei He. Non-gaussian statistical timing analysis using second-order polynomial fitting. *2008 Asia and South Pacific Design Automation Conference*, pages 298–303, 2008.
- [12] Daiki Azuma, Shuji Tsukiyama, and Masahiro Fukui. Approximating the maximum of gaussians by a gaussian mixture model for statistical designs. *2017 European Conference on Circuit Theory and Design (ECCTD)*, pages 1–4, 2017.
- [13] Jennifer Freeley, Dmytro Mishagli, Tom Brazil, and Elena Blokhina. Statistical simulations of delay propagation in large scale circuits using graph traversal and kernel function decomposition. *2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pages 213–9, 2018.
- [14] Dmytro Mishagli and Elena Blokhina. Radial basis functions based algorithms for non-gaussian delay propagation in very large circuits. *Computational Science – ICCS 2020*, 12141:217 – 229, 2020.
- [15] Dmytro Mishagli, Eugene Koskin, and Elena Blokhina. Gate-level statistical timing analysis: Exact solutions, approximations and algorithms. *arXiv preprint arXiv:2401.03588*, 2024.

- [16] Harald Cramér. *Mathematical methods of statistics*, volume 9. Princeton university press, 1999.
- [17] Giacomo Capodaglio, Max D. Gunzburger, and Henry P. Wynn. Approximation of probability density functions for pdes with random parameters using truncated series expansions. *Vietnam Journal of Mathematics*, 49:685 – 711, 2018.
- [18] Daniel Dufresne and HanBo Li. Pricing asian options: Convergence of gram–charlier series. *Actuarial Research Clearing House*, 2016 Issue 2, 2016.
- [19] Sergei Blinnikov and Richhild Moessner. Expansions for nearly gaussian distributions. *Astronomy & Astrophysics Supplement Series*, 130:193–205, 1997.
- [20] Gabor Szeg. *Orthogonal polynomials*, volume 23. American Mathematical Soc., 1939.
- [21] James V Uspensky. On the development of arbitrary functions in series of hermite’s and laguerre’s polynomials. *Annals of Mathematics*, 28(1/4):593–619, 1926.
- [22] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of The Society for Industrial and Applied Mathematics*, 11:431–441, 1963.
- [23] William H Press. *Numerical recipes in Pascal: the art of scientific computing*, volume 1. Cambridge university press, 1989.
- [24] Brian Gough. *GNU scientific library reference manual*. Network Theory Ltd., 2009.

Appendix

A Right-tail model derivation

To complement the left-tail derivation in Sec. 3 and the tail-fitting algorithm proposed in Sec. 5, the analogous formulation for the right tail is presented here. Suppose the right-tail PDF is given by Eq. (4) as follows:

$$f(t; \{c_j^{(r)}\}_{j=0}^N, \mu^{(r)}, \sigma^{(r)}) = \sum_{j=0}^N c_j^{(r)} t^j \phi(t; \mu^{(r)}, \sigma^{(r)}), \quad t \geq q_n,$$

where $\phi(t; \mu, \sigma)$ is the Gaussian PDF for $\mathcal{N}(\mu, \sigma^2)$. The corresponding right-tail CDF is computed by

$$\tilde{F}(x; \{c_j^{(r)}\}_{j=0}^N, \mu^{(r)}, \sigma^{(r)}) = 1 - \int_x^{+\infty} f(t) dt.$$

We denote J_i as

$$J_i(x; \mu, \sigma) := \int_x^{+\infty} t^i \phi(t; \mu, \sigma) dt.$$

We derive the recurrence relations for $J_i(x; \mu, \sigma)$ inductively:

$$\begin{aligned} J_0(x; \mu, \sigma) &= 1 - \Phi(x; \mu, \sigma), \\ J_1(x; \mu, \sigma) &= \mu J_0(x; \mu, \sigma) + \sigma \phi(x; \mu, \sigma), \\ J_i(x; \mu, \sigma) &= \mu J_{i-1}(x; \mu, \sigma) + (i-1)\sigma^2 J_{i-2}(x; \mu, \sigma) + x^{i-1} \sigma \phi(x; \mu, \sigma), \quad i \geq 2. \end{aligned}$$

Thus, for $x \geq q_n$, the right-tail CDF lies in the span of

$$\text{span} \left\{ 1 - \Phi(x; \mu^{(r)}, \sigma^{(r)}), \phi(x; \mu^{(r)}, \sigma^{(r)}), x\phi(x; \mu^{(r)}, \sigma^{(r)}), \dots, x^{N-1}\phi(x; \mu^{(r)}, \sigma^{(r)}) \right\}.$$

For the max operator, we focus on the optimization of the following two expressions:

$$\begin{cases} \int_{q_n}^{+\infty} \left(1 - F_1(x)F_2(x) - \sum_{j=0}^N c_j^{(r)} J_j(x; \mu^{(r)}, \sigma^{(r)}) \right)^2 dx, \\ 1 - F_1(x)F_2(x) - \sum_{j=0}^N c_j^{(r)} J_j(x; \mu^{(r)}, \sigma^{(r)}), \quad \forall x \geq q_n. \end{cases}$$

The tail-fitting optimization problem for the right tail in our algorithm is analogous to that for the left tail. Suppose y_i is the exact CDF at x_i . We solve the following nonlinear least-squares problem:

$$\min \sum_{i=1}^k \left[\left(\tilde{F}(x_i; \{c_j^{(r)}\}_{j=0}^N, \mu^{(r)}, \sigma^{(r)}) - y_i \right) / (1 - y_i) \right]^2 + \lambda (c_0^{(r)} - 1)^2 + \lambda \left(\sum_{j=1}^N (c_j^{(r)})^2 \right),$$

The Jacobian matrix for this expression can be computed by differentiating each $J_j(x_i; \mu^{(r)}, \sigma^{(r)})$ for the parameters $\{c_j^{(r)}\}_{j=0}^N$, $\mu^{(r)}$, and $\sigma^{(r)}$. We use the same reparameterization $\sigma^{(r)} = \exp(p_\sigma)$ to ensure positivity.

B Model sensitivity to correlation

While the proposed model and propagation algorithm demonstrate strong expressive power and robust performance under the independence assumption, their accuracy may degrade in settings where correlations within the graph are non-negligible. To examine this limitation, we consider the simple graph shown in Fig. 7.

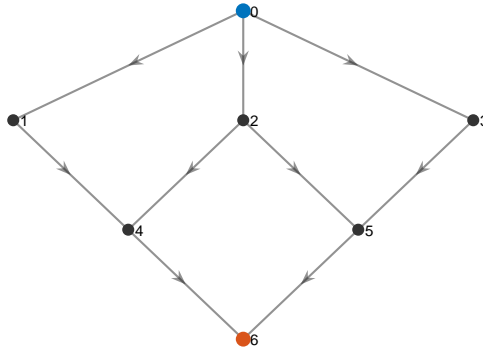


Figure 7: Diamond tree structure

The incremental delays on each node and edge are modeled as independent Gaussian random variables, whose μ and σ are drawn randomly from $\mathcal{U}(2.0, 5.0)$ and $\mathcal{U}(1.5, 2.0)$ respectively.

As illustrated in Fig. 7, the four nodes $\{2, 4, 5, 6\}$ form a diamond structure, which induces statistical dependence when the input delay of node 6 is computed via a max operation over two paths that share common ancestors.

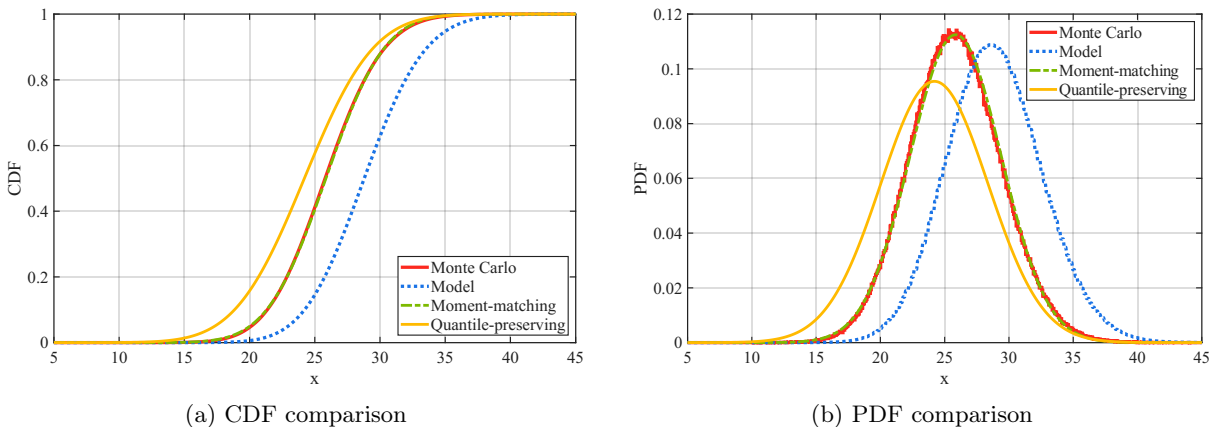


Figure 8: Comparison of (a) CDF and (b) PDF of the delay propagation on a diamond tree structure between Monte Carlo simulation, our model results, the moment-matching method, and the Gaussian quantile-preserving method.

As expected, Fig. 8 shows that our model exhibits noticeable deviations from the Monte Carlo ground truth. This behavior is a characteristic of propagation schemes that ignore correlations among random variables.