FOURIER NEURAL NETWORK APPROXIMATION OF TRANSITION DENSITIES IN FINANCE

RONG DU * AND DUY-MINH DANG†

Abstract. This paper introduces FourNet, a novel single-layer feed-forward neural network (FFNN) method designed to approximate transition densities for which closed-form expressions of their Fourier transforms, i.e. characteristic functions, are available. A unique feature of FourNet lies in its use of a Gaussian activation function, enabling exact Fourier and inverse Fourier transformations and drawing analogies with the Gaussian mixture model. We mathematically establish FourNet's capacity to approximate transition densities in the L_2-sense arbitrarily well with finite number of neurons. The parameters of FourNet are learned by minimizing a loss function derived from the known characteristic function and the Fourier transform of the FFNN, complemented by a strategic sampling approach to enhance training. We derive practical bounds for the L_2 estimation error and the potential pointwise loss of nonnegativity in FourNet for d-dimensions, $d \geq 1$, highlighting its robustness and applicability in practical settings. FourNet's accuracy and versatility are demonstrated through a wide range of dynamics common in quantitative finance, including Lévy processes and the Heston stochastic volatility models-including those augmented with the self-exciting Queue-Hawkes jump process.

Key words. transition densities, neural networks, Gaussian activation functions, Fourier transforms, characteristic functions

MSC codes. 62M45, 91-08, 60E10, 62P05

1 Introduction The application of machine learning, especially deep learning, in quantitative finance has garnered considerable interest. Recent breakthroughs in computational resources, data availability, and algorithmic enhancements have encouraged the adoption of machine learning techniques in various quantitative finance domains. These include, but are not limited to, portfolio optimization [33, 55], asset pricing [57, 4], model calibration and option pricing [52, 34, 24], solution of high-dimensional partial differential equations [22, 26, 56, 50], valuation adjustments [16, 19, 20], as well as aspects of stochastic control and arbitrage-free analysis [27, 46, 9].

Transition (probability) density functions, which are crucial in quantitative finance due to their primary role in governing the dynamics of stochastic processes, often do not admit a closed-form expression. Consequently, the utilization of numerical methods for estimating these density functions becomes necessary. Classical methods include kernel density estimation, as referenced in [43, 48, 18]. Yet, surprisingly, the development of neural network (NN) methods for estimating these transition probability density functions is significantly underdeveloped. While some existing NN strategies tackle the associated high-dimensional Kolmogorov partial differential equations (PDEs) using deep NNs, these are primarily black-box in nature. Such methodologies have seen applications in option pricing ([52, 51]) and general Itô diffusions ([21]). While these methods are generally effective and versatile, they come with a major limitation: their model-dependent nature necessitates a constant reformulation of the Kolmogorov PDEs for different stochastic models. In addition, the inherent complexity associated with deploying NNs to solve PDEs might deter their practical application. Furthermore, a notable gap in the NN literature, particularly regarding transition density function estimation, is the limited analysis of estimation

*School of Mathematics and Physics, The University of Queensland, St Lucia, Brisbane 4072, Australia, (rong.du1@uq.net.au).

†School of Mathematics and Physics, The University of Queensland, St Lucia, Brisbane 4072, Australia, (duyminh.dang@uq.edu.au).

error and potential compromise of non-negativity.

In quantitative finance, many popular stochastic models have unknown transition densities; however, their Fourier transforms, i.e. characteristic functions, are often explicitly available via the Lévy-Khintchine formula [28]. This property has been extensively utilized in option pricing through various numerical methods. Prominent among these are the Carr-Madan approach [7], the Convolution (CONV) technique [35], Fourier Cosine (COS) method proposed by [13], Shannon-wavelet methods [45, 11], with the COS method being particularly noteworthy. Specifically, the COS method is known for achieving high-order convergence for piecewise smooth problems. However, within the broader framework of stochastic optimal control, where problems often exhibit complex and non-smooth characteristics, this high-order convergence is unattainable, as noted in [38, 15]. A notable drawback of the COS method is its lack of a mechanism to control the potential loss of non-negativity in estimated transition densities. This issue–more pronounced with short maturities–may lead to violations of the no-arbitrage principle in numerical value functions, posing significant challenges in stochastic optimal control where the accuracy of these values is crucial for making optimal decisions [15]. In the same vein of research, recent works on ϵ-monotone Fourier methods for control problems in finance merit attention [15, 37, 38, 36].

In response to the noted observations, this paper sets out to achieve three primary objectives. Firstly, we present a single-layer feed-forward (FF) NN approach to approximate transition densities with closed-form Fourier transforms, facilitating training in the Fourier domain. This approach simplifies the implementation considerably when compared to deeper NN structures. Second, we conduct a rigorous and comprehensive analysis of the L_2 estimation error between the exact and the estimated transition densities obtained through the proposed approach. This methodology, dubbed the Fourier-trained Neural Network method or "FourNet", showcases the benefits of using the Fourier transform in FFNN models. Lastly, we validate FourNet's accuracy and versatility across a spectrum of stochastic financial models. The main contributions of this paper are as follows.

- We establish two key results for FourNet: (i) transition densities can be approximated arbitrarily well in the L_2-sense using a single-layer FFNN with a Gaussian activation function and a finite number of neurons; and (ii) the L_2-error in this approximation remains invariant under the Fourier transform map. Here, $L_2(\mathbb{R})$ denotes the space of square-integrable functions.
  FourNet's methodology underscores the potential and efficacy of shallow NN architectures for complex approximation tasks. The inherent invariance under Fourier transformation opens opportunities for training and error analysis in the Fourier domain, rather than the conventional spatial domain. This unique capability allows us to utilize the the known closed-form expression of the characteristic function and the Fourier transform of the FFNN for an in-depth analysis of the L_2 estimation error.
- Using FourNet, we formulate an approximation for transition densities using a single-layer FFNN equipped with a Gaussian activation function. FourNet's parameters are fine-tuned by minimizing a mean-squared-error (MSE) loss, supplemented with a mean-absolute-error (MAE) regularization. Both the loss function and regularization term stem from the known characteristic function and the Fourier transform of the FFNN. A strategic sampling approach is proposed, maximizing the benefits of MAE regularization.
  We establish practical bounds for the L_2 estimation error and potential loss of nonnegativity in FourNet for the general case of d-dimensions ($d \geq 1$),

which are attributed to truncation, training, and sampling errors. These bounds highlight FourNet's advantages over existing Fourier-based and NN-based density estimation methods, offering valuable insights into its reliability and robustness in practical applications.

- We showcase FourNet's accuracy and versatility across a broad spectrum of financial models, with a particular focus on option pricing. Our analysis encompasses the class of exponential Lévy processes, such as the CGMY model [6], Merton jump-diffusion model [42], and Kou asymmetric double exponential model [31], along with their multi-dimensional extensions. We also explore the Heston model [23] and its recent adaptations that incorporate the self-exciting Queue-Hawkes jump process [12, 2]. Notably, FourNet demonstrates exceptional robustness in handling ultra-short maturities and asymmetric heavy-tailed distributions, scenarios that pose significant challenges for traditional Fourier-based methods.

This paper introduces the FourNet method and its initial applications, setting the stage for subsequent works. Primarily focusing on European options, it also demonstrates FourNet's capabilities in pricing Bermudan options within Lévy process-based models. Future work will extend FourNet's application to complex stochastic control problems, including portfolio optimization, thereby broadening its scope and utility. Although this work centers on transition densities, FourNet's methodology and its comprehensive error analysis are also relevant to the study of Green's functions for parabolic integro-differential equations [17], due to their foundational relationship.

The remainder of the paper is organized as follows. Section 2 outlines the structure of single-layer FFNNs with non-sigmoid activation functions and introduces a related universal approximation theorem. Section 3 presents FourNet, detailing key mathematical results and the MSE loss function. An error analysis of FourNet is detailed in Section 4. Section 5 discusses sampling strategies, training considerations, and algorithms. Section 6 demonstrates FourNet's accuracy and versatility through extensive numerical experiments. The paper concludes in Section 7 with a discussion of potential future work.

2 Background on single-layer FFNNs

2.1 Non-sigmoid activation functions Feed-forward neural networks can be perceived as function approximators comprising of several inputs, hidden layers composed of neurons/nodes, an activation function, and several outputs. This study primarily concentrates on shallow NNs characterized by a single input, a single output, and a number of nodes within the hidden layer. We also consider only the case that the input is one-dimensional. Figure 2.1 depicts a single-layer FFNN having a total of N nodes in the hidden layer.

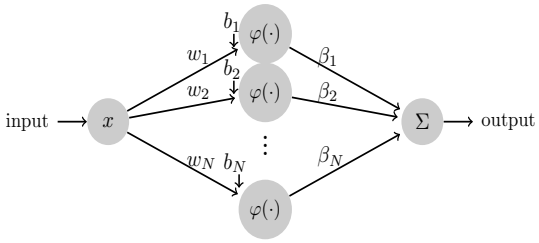


Fig. 2.1: Single-layer FFNN with one-dimensional input $x \in \mathbb{R}$, featuring N neurons with weights w_n and β_n, biases b_n, and activation function $\varphi(\cdot)$.

We now start with FFNNs with (Borel measurable) non-sigmoid activation functions, and the associated Universal Approximation Theorem [40][Theorem 2.1]. This class of of FFNNs is defined below.

DEFINITION 2.1 ($\Sigma^\dagger(\varphi)$ - activation function φ). *Let $\Sigma^\dagger(\varphi)$ be the class of single-layer FFNNs having arbitrary Borel measurable activation functions φ defined by*

$$\Sigma^\dagger(\varphi) = \left\{ \widehat{g} : \mathbb{R} \to \mathbb{R} \middle| \widehat{g}(x;\theta) = \sum_{n=1}^{N} \beta_n \varphi\left(w_n x + b_n\right),\ \beta_n, w_n, b_n \in \mathbb{R},\ N \in \mathbb{N} \right\}. \tag{2.1}$$

Here, $x \in \mathbb{R}$ is the input; for a fixed N, the parameter $\theta \in \mathbb{R}^{3N}$ is constituted by the weights w_n and β_n, and the bias terms b_n, $n = 1, \ldots, N$.

For subsequent use, for $1 \leq p < \infty$, we define the sets of p-integrable and p-locally-integrable functions, respectively denoted by $L_p(\mathbb{R})$ and $L_p(\mathbb{R}, loc)$, as follows

$$\begin{aligned} L_p(\mathbb{R}) &= \left\{ f \in \mathcal{M} \mid \|f\|_p \equiv \left[\int |f(x)|^p dx \right]^{1/p} < \infty \right\}, \\ L_p(\mathbb{R}, loc) &= \left\{ f \in \mathcal{M} \mid f\, \mathbb{I}_{[-A,A]} \in L_p(\mathbb{R}), \forall A \in \{1, 2, 3, \ldots\} \right\}. \end{aligned} \tag{2.2}$$

Here, $\mathcal{M}$ is the space of all Borel measurable functions $f : \mathbb{R} \to \mathbb{R}$.[1]

Closeness of two elements f_1 and f_2 of $L_p(\mathbb{R}, loc)$ is measured by a metric $\rho_{p,loc}(f_1, f_2)$ defined as follows [40]

$$\rho_{p,loc}(f_1, f_2) = \sum_{A=1}^{\infty} \left(2^{-A}\right) \min\left(\left\| (f_1 - f_2)\, \mathbb{I}_{[-A,A]} \right\|_p, 1 \right), \quad f_1, f_2 \in L_p(\mathbb{R}, loc). \tag{2.3}$$

Here, the indicator function $\mathbb{I}_D(\cdot)$ is defined as follows: $\mathbb{I}_D(x) = 1$ if $x \in D$ and zero otherwise. We now introduce the notion of $\rho_{p,loc}$-denseness for $L_p(\mathbb{R}, loc)$ [40].

DEFINITION 2.2 (ρ_p-denseness, $1 \leq p < \infty$). *A subset $\mathcal{S}$ of $L_p(\mathbb{R}, loc)$ is $\rho_{p,loc}$-dense in $L_p(\mathbb{R}, loc)$ if, for any f_1 in $L_p(\mathbb{R}, loc)$ and any $\varepsilon > 0$, there is a f_2 in $\mathcal{S}$ such that $\rho_{p,loc}(f_1, f_2) < \varepsilon$, where $\rho_{p,loc}(f_1, f_2)$ is defined in (2.3).*

2.2 Universal Approximation Theorem The Universal Approximation Theorem proposed in [25] for sigmoid activation functions play a key theoretical foundation. However, sigmoid activation functions are not necessary for universal approximation as highlighted in [40][Theorem 2.1] - therein, an identical universal approximation theorem to the one in [25] was obtained. The key finding of [40] is that, for sufficiently complex single-layer FFNNs with an arbitrary (Borel measurable) activation function at the hidden layer can approximate an arbitrary target function $f(\cdot) \in L_p(\mathbb{R}, loc)$, $1 \leq p < \infty$, arbitrary well, provided that the activation function, denoted by $\varphi(x)$, belong to $L_1(\mathbb{R}) \cap L_p(\mathbb{R})$ and $\int_{\mathbb{R}} \varphi(x) dx$ does not vanish. Formally, we state the Universal Approximation Theorem for non-sigmoid activation functions below.

THEOREM 2.3 (Universal Approximation Theorem [40] Theorem 2.1). *Let φ be the (Borel measurable) activation function that belongs to $L_1(\mathbb{R}) \cap L_p(\mathbb{R})$, $1 \leq p < \infty$. If $\int_{\mathbb{R}} \varphi(x)\, dx \neq 0$, then $\Sigma^\dagger(\varphi)$ is $\rho_{p,loc}$-dense in $L_p(\mathbb{R}, loc)$. Here, $\Sigma^\dagger(\varphi)$ and $\rho_{p,loc}$ are respectively defined in Definitions 2.1 and 2.2.*

3 A Fourier-trained network (FourNet) We denote by $T > 0$ a finite horizon, and let t and Δt be fixed such that $0 \leq t < t + \Delta t \leq T$. For the sake of exposition, we focus on estimating a time and spatially homogeneous transition density, denoted by $g(\cdot)$ and is represented as $g(x, t + \Delta t; y, t) = g(x - y; \Delta t)$. Such transition densities are characteristic of Lévy processes.

[1] The set $\mathcal{M}$ essentially contains all functions relevant to practical applications.

Our methodology also applies to models like the Heston and Heston-Queue-Hawkes, which exhibit non-homogeneity in variance. For these models, pricing European options needs only a single training session; however, control problems typically require multiple independent sessions to address time-stepping. Each session uses a dataset derived from characteristic functions conditioned on different starting and ending variance values. These sessions can be conducted in parallel, thereby enhancing computational efficiency. This aligns with existing Fourier methods (e.g. [14, 15, 49]) which similarly utilize characteristic functions conditioned on variance values at each timestep. We plan to explore Heston-type models in control problems in future work.

For notational simplicity, we momentarily suppress the explicit dependence of the transition density on Δt, denoting $g(\cdot) \equiv g(\cdot; \Delta t) : \mathbb{R} \to \mathbb{R}$ as the transition density we seek to approximate using single-layer FFNNs. The importance of Δt will be highlighted in our applications detailed in Section 6.

Since the transition density $g(\cdot)$ is almost everywhere bounded on $\mathbb{R}$, together with the fact that $g \in L_1(\mathbb{R})$, we have $g \in L_2(\mathbb{R})$. Therefore, we consider approximating a transition density $g \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$.

3.1 A universal approximation result in $L_2(\mathbb{R})$ We now provide a refined version of the Universal Approximation Theorem 2.3 for target functions in $L_2(\mathbb{R})$. Specifically, by invoking Hölder's inequality, we have that $L_2(\mathbb{R}) \subset L_2(\mathbb{R}, \mathrm{loc})$. A natural question thus emerges: if the function we aim to approximate, f, belongs to $L_2(\mathbb{R})$ rather than $L_2(\mathbb{R}, loc)$, can we identify an FFNN in $\Sigma^\dagger(\varphi) \cap L_2(\mathbb{R})$ that approximates f arbitrarily well, in the sense of the Universal Approximation Theorem 2.3? In the forthcoming lemma, we affirmatively address this question.

LEMMA 3.1 ($\rho_{2,loc}$-denseness of $\Sigma^\dagger(\varphi) \cap L_2(\mathbb{R})$). *Let φ be a continuous activation function that belongs to $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$. Assume that $f(\cdot)$ is in $L_2(\mathbb{R})$. For any $\epsilon > 0$, there exists a neural network $f'(\cdot; \theta') \in \Sigma^\dagger(\varphi) \cap L_2(\mathbb{R})$ such that $\rho_{2,loc}(f, f') < \epsilon$, where $\rho_{2,loc}(\cdot)$ is defined in Definition 2.3.*

Proof of Lemma 3.1. Since $\varphi \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$, it satisfies the conditions of Theorem 2.3 for $p = 2$. Therefore, $\exists f'(\cdot; \theta') \in \Sigma^\dagger(\varphi)$ such that $\rho_{2,loc}(f, f') < \epsilon$. Here, $f'(x; \theta') = \sum_{n=1}^{N'} \beta'_n \varphi(w'_n x + b'_n)$, where N' is the finite number of neurons. As $f'(\cdot; \theta') \in L_2(\mathbb{R}, loc)$, each $\beta'_n \varphi(w'_n x + b'_n)$ is square-integrable on any compact set. Since $\varphi \in L_2(\mathbb{R})$, the square-integrability of $\beta'_n \varphi(w'_n x + b'_n)$ implies that $|\beta'_n| < \infty$ for all $n \leq N'$. Finally, as $f'(\cdot; \theta')$ is a finite sum of functions in $L_2(\mathbb{R})$, it follows that $f'(\cdot; \theta') \in L_2(\mathbb{R})$. This concludes the proof. ∎

3.2 Gaussian activation function e^{-x^2} Building upon Lemma 3.1, we present a corollary focusing on the Gaussian activation function $\varphi(x) \equiv \phi(x) = e^{-x^2}$.

COROLLARY 3.2. *For a target function $f(\cdot)$ in $L_2(\mathbb{R})$ and any $\epsilon > 0$, there exists an FFNN $f'(\cdot; \theta') \in \Sigma^\dagger(\phi) \cap L_2(\mathbb{R})$ with $\phi(x) = e^{-x^2}$, where f' approximates f such that $\rho_{2,loc}(f, f') < \epsilon$. The FFNN $f'(\cdot; \theta')$ has bounded parameters: $0 < |\beta'_n| < \infty$, $0 < |w'_n| < \infty$, and $|b'_n| < \infty$, $\forall n = 1, \ldots, N'$.*

Proof. Through integration, we verify that $\phi(x) = e^{-x^2} \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$. Thus, by Lemma 3.1, $\exists\ f'(\cdot; \theta') \in \Sigma^\dagger(\phi) \cap L_2(\mathbb{R})$ such that $\rho_{2,loc}(f, f') < \epsilon$. Here, $f'(x; \theta') = \sum_{n=1}^{N'} \beta'_n \varphi(w'_n x + b'_n)$. By Lemma 3.1, $|\beta'_n| < \infty$, $\forall n \leq N'$. Additionally, it must be true that $\beta'_n \neq 0$, $\forall n \leq N$; otherwise the corresponding neuron output is zero. For the same reason, we also have $|w'_n|, |b'_n| < \infty$, $\forall n \leq N'$. Lastly, it is also the case

that $w'_n \neq 0$ for all $n \leq N'$. Otherwise, suppose that $w'_k = 0$ for $n = k \leq N'$, then $0 < |\beta'_k| \exp(-(w'_k x + b'_k)^2) = |\beta'_k| \exp(-(b'_k)^2) \leq c$, where $0 < c < \infty$, contradicting with $f'(\cdot;\theta') \in L_2$. This concludes the proof. ∎

Corollary 3.2 establishes that for any target function in $L_2(\mathbb{R})$, an FFNN with bounded parameters exists within $\Sigma^\dagger(\phi) \cap L_2(\mathbb{R})$, where $\phi(x) = e^{-x^2}$, capable of approximating the target function arbitrarily well as measured by $\rho_{2,loc}(\cdot,\cdot)$. This result allows us to narrow down our focus to $\Sigma(\phi)$, a more specific subset of $\Sigma^\dagger(\phi)$, that encapsulates FFNNs characterized by parameter bounds. We now formally define $\Sigma(\phi)$ and its associated bounded parameter space Θ, both of which are crucial for our subsequent theoretical analysis and practical application:

$$\Sigma(\phi) = \Big\{ \widehat{g} : \mathbb{R} \to \mathbb{R} \Big| \ \widehat{g}(x;\theta) = \sum_{n=1}^{N} \beta_n \phi(w_n x + b_n),\ \phi(x) = e^{-x^2},\ \theta \in \Theta \Big\}, \tag{3.1}$$

$$\Theta = \Big\{ \theta \in \mathbb{R}^{3N} \ \Big| \ 0 < |\beta_n| < \infty,\ 0 < |w_n| < \infty,\ |b_n| < \infty,\ n = 1, \ldots, N \Big\}. \tag{3.2}$$

Remark 3.3. All FFNNs in $\Sigma(\phi)$, where $\phi(x) = e^{-x^2}$, belong to $L_2(\mathbb{R})$ as per definitions (3.1) and (3.2). By Corollary 3.2, given any target function in $f(\cdot)$ in $L_2(\mathbb{R})$ and any $\epsilon > 0$, there exists an FFNN $f'(\cdot;\theta') \in \Sigma(\phi)$ such that $\rho_{2,loc}(f, f') < \epsilon$.

3.3 Existence and invariance of FourNet We now establish a key result demonstrating the existence of $\widehat{g}(\cdot;\theta^*_\epsilon) \in \Sigma(\phi)$, where $\Sigma(\phi)$ is defined in (3.1), that is capable of approximating the exact transition density $g(\cdot)$ arbitrarily well in the L_2-sense. We hereafter refer to $\widehat{g}(\cdot;\theta^*_\epsilon)$ as a theoretical FFNN approximation to the true transition density $g(\cdot)$. Furthermore, we also show that the associated theoretical approximation error in L_2 remains invariant under the Fourier transform map.

To this end, we recall that the transition density $g(\cdot)$ and the associated characteristic function $G(\eta)$ are a Fourier transform pair. They are defined as follows

$$\mathfrak{F}[g(\cdot)](\eta) \equiv G(\eta) = \int_{-\infty}^{\infty} e^{i\eta x}\, g(x)\, dx, \quad \mathfrak{F}^{-1}[G(\cdot)](x) \equiv g(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\eta x}\, G(\eta)\, d\eta.$$

For subsequent discussions, for a complex-valued function $f : \mathbb{R} \to \mathbb{C}$, we denote by $\mathrm{Re}_f(\cdot)$ and $\mathrm{Im}_f(\cdot)$ its real and imaginary parts. We also have $|f(\cdot)|^2 = f(\cdot)\overline{f(\cdot)}$, where $\overline{f(\cdot)}$ is the complex conjugate of $f(\cdot)$.

We will also utilize the Plancherel Theorem, which is sometimes also referred to as the Parseval-Plancherel identity [58, 1, 30]. For the sake of convenience, we reproduce it below. Let $f : \mathbb{R} \to \mathbb{R}$ be a function in $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$. The Plancherel Theorem states that its Fourier transform $\mathfrak{F}[f(\cdot)](\eta)$ is in $L_2(\mathbb{R})$, and

$$\int_{\mathbb{R}} |f(x)|^2\, dx = \frac{1}{2\pi} \int_{\mathbb{R}} |\mathfrak{F}[f(\cdot)](\eta)|^2\, d\eta. \tag{3.3}$$

THEOREM 3.4 (FourNet's existence result). *Given any $\epsilon > 0$, there exists an FFNN $\widehat{g}(\cdot;\theta^*_\epsilon) \in \Sigma(\phi)$, where $\Sigma(\phi)$ is defined in (3.1), that satisfies the following*

$$\int_{\mathbb{R}} |g(x) - \widehat{g}(x;\theta^*_\epsilon)|^2\ dx = \frac{1}{2\pi} \int_{\mathbb{R}} \left| G(\eta) - \widehat{G}(\eta;\theta^*_\epsilon) \right|^2\ d\eta\ <\ \epsilon. \tag{3.4}$$

Here, $\widehat{G}(\eta;\theta^*_\epsilon)$ is the Fourier transform of $\widehat{g}(\cdot;\theta^*_\epsilon)$, i.e. $\widehat{G}(\eta;\theta^*_\epsilon) = \mathfrak{F}\left[\widehat{g}(\cdot;\theta^*_\epsilon)\right](\eta)$.

Proof of Theorem 3.4. We first show $\int_{\mathbb{R}} |g(x) - \widehat{g}(x;\theta^*_\epsilon)|^2\ dx < \epsilon$, then the equality in (3.4). Since $g(\cdot)$ and $\widehat{g}(\cdot;\theta^*_\epsilon)$ are in $L_2(\mathbb{R})$, $\exists A'$ sufficiently large such that

$$\int_{\mathbb{R}\setminus[-A',A']} |g(x)|^2\ dx\ <\ \epsilon/8, \quad \int_{\mathbb{R}\setminus[-A',A']} |\widehat{g}(x;\theta^*_\epsilon)|^2\ dx\ <\ \epsilon/8. \tag{3.5}$$

By Remark 3.3, there exists $\widehat{g}(x;\theta_\epsilon^*) \in \Sigma(\phi)$ such that

$$\rho_{2,loc}\left(g, \widehat{g}\left(\cdot;\theta_\epsilon^*\right)\right) = \sum_{A=1}^{\infty} 2^{-A} \min\left(\left\|\left(g - \widehat{g}(\cdot;\theta_\epsilon^*)\right) \mathbb{I}_{[-A,A]}\right\|_2, 1\right) < \frac{\epsilon^{1/2}}{2^{1/2}} 2^{-A'}.$$

Therefore,

$$2^{-A'} \min\left(\left\|\left(g - \widehat{g}(\cdot;\theta_\epsilon^*)\right) \mathbb{I}_{[-A',A']}\right\|_2, 1\right) < \frac{\epsilon^{1/2}}{2^{1/2}} 2^{-A'},$$

from which, we have

$$\int_{[-A',A']} \left(g(x) - \widehat{g}(x;\theta_\epsilon^*)\right)^2 \, dx < \epsilon/2. \tag{3.6}$$

Using (3.5)-(3.6), we have $\int_{\mathbb{R}} (g(x) - \widehat{g}(x;\theta_\epsilon^*))^2 \, dx = \ldots$

$$\begin{aligned} \ldots &= \int_{[-A',A']} |g(x) - \widehat{g}(x;\theta_\epsilon^*)|^2 \, dx + \int_{\mathbb{R}\setminus[-A',A']} |g(x) - \widehat{g}(x;\theta_\epsilon^*)|^2 \, dx \\ &< \epsilon/2 + \int_{\mathbb{R}\setminus[-A',A']} 2\left|g(x)^2 + \widehat{g}(x;\theta_\epsilon^*)^2\right| \, dx < \epsilon, \end{aligned} \tag{3.7}$$

as wanted. Next, the equality in (3.4) follows directly from the Plancherel Theorem (3.3), noting $L_1(\mathbb{R})$ and $L_2(\mathbb{R})$ are closed under addition. This completes the proof.□

Remark 3.5. Theorem 3.4 presents a significant theoretical result, demonstrating that the FourNet can approximate the exact transition density $g(\cdot)$ within an error of any given magnitude in the L_2-sense.[2] Interestingly, this error is invariant under the Fourier transform, tying together FourNet's approximation capabilities in both spatial and Fourier domains. This invariance opens opportunities for training and error analysis in the Fourier domain instead of the spatial domain. In particular, it enables us to utilize the known closed-form expression of the characteristic function $G(\cdot)$, a process we elaborate on in subsequent sections.

3.4 Loss function Recall that $\widehat{g}(x;\theta)$ in $\Sigma(\phi)$ has the form

$$\widehat{g}(x;\theta) = \sum_{n=1}^{N} \beta_n \phi\left(w_n x + b_n\right), \quad \phi(x) = \exp(-x^2), \quad \theta \in \Theta. \tag{3.8}$$

We let $\widehat{G}(\cdot;\theta)$ be the Fourier transform of $\widehat{g}(\cdot;\theta)$, i.e. $\widehat{G}(\eta;\theta) = \mathfrak{F}\left[\widehat{g}(\cdot;\theta)\right](\eta)$. By substitution, we have

$$\begin{aligned} \widehat{G}(\eta;\theta) &= \int e^{i\eta x} \widehat{g}(x;\theta) \, dx = \sum_{n=1}^{N} \beta_n \int e^{i\eta x} \phi(w_n x + b_n) \, dx \\ &= \sum_{n=1}^{N} \beta_n \int_{\mathbb{R}} \cos(\eta x) \, \phi(w_n x + b_n) \, dx + i \sum_{n=1}^{N} \beta_n \int_{\mathbb{R}} \sin(\eta x) \, \phi(w_n x + b_n) \, dx \\ &= \mathrm{Re}_{\widehat{G}}(\eta;\theta) + i\mathrm{Im}_{\widehat{G}}(\eta;\theta). \end{aligned} \tag{3.9}$$

Here, by integrating the integral terms with $\phi(x) = \exp(-x^2)$, we obtain

$$\mathrm{Re}_{\widehat{G}}(\cdot) = \sum_{n=1}^{N} \frac{\beta_n \sqrt{\pi}}{w_n} \cos\left(\frac{\eta b_n}{w_n}\right) \exp\left(\frac{-\eta^2}{4w_n^2}\right), \; \mathrm{Im}_{\widehat{G}}(\cdot) = \sum_{n=1}^{N} \frac{\beta_n \sqrt{\pi}}{w_n} \sin\left(\frac{-b_n \eta}{w_n}\right) \exp\left(\frac{-\eta^2}{4w_n^2}\right).$$

[2] This result is proved in [5] when the variances are fixed.

Recall that our starting point is that $G(\cdot)$, the Fourier transform of the transition density $g(\cdot)$, is known in closed form. Therefore, motivated by Theorem 3.4, the key step of our methodology is to use the known data $\{(\eta, \mathrm{Re}_G(\eta;\theta))\}$ and $\{(\eta, \mathrm{Im}_G(\eta;\theta))\}$ to train $\widehat{G}(\eta;\theta)$ using the expressions in (3.9).

To this end, we restrict the domain of η from $\mathbb{R}$ to a fixed finite interval $[-\eta', \eta']$, where $0 < \eta' < \infty$ and is sufficiently large. We denote the total number of training data points by P, and we consider a deterministic, potentially non-uniform, partition $\{\eta_p\}_{p=1}^P$ of the interval $[-\eta', \eta']$. With $\delta_p = \eta_{p+1} - \eta_p$, $p = 1, \ldots, P-1$, we assume

$$\delta_{\min} = C_0/P, \quad \delta_{\max} = C_1/P, \quad \text{with } \delta_{\min} = \min_p \delta_p \text{ and } \delta_{\max} = \max_p \delta_p, \tag{3.10}$$

where the constants $C_0, C_1 > 0$ are finite and independent of P. Letting $\widehat{\Theta} \subseteq \Theta$ be the empirical parameter space, we introduce an empirical loss function $\mathrm{Loss}_P(\theta)$, $\theta \in \widehat{\Theta}$, below

$$\mathrm{Loss}_P(\theta) = \frac{1}{P}\sum_{p=1}^{P} \left| G(\eta_p) - \widehat{G}(\eta_p;\theta) \right|^2 + R_P(\theta), \quad \{\eta_p\}_{p=1}^P \text{ satisfying (3.10)}. \tag{3.11}$$

Here, $\widehat{G}(\eta_p;\theta)$ is defined in (3.9), with $R_P(\theta)$ as the MAE regularization term

$$R_P(\theta) = \frac{1}{P}\sum_{p=1}^{P} \left(|\mathrm{Re}_G(\eta_p) - \mathrm{Re}_{\widehat{G}}(\eta_p;\theta)| + |\mathrm{Im}_G(\eta_p) - \mathrm{Im}_{\widehat{G}}(\eta_p;\theta)| \right). \tag{3.12}$$

By training $\mathrm{Loss}_P(\cdot)$, we aim to find the empirical minimizer $\widehat{\theta}^* \in \widehat{\Theta}$, where

$$\widehat{\theta}^* = \arg\min_{\theta \in \widehat{\Theta}} \mathrm{Loss}_P(\theta). \tag{3.13}$$

Remark 3.6 (MAE regularization). The incorporation of the MAE regularization term in the loss function (3.11) is strategically motivated by its ability to significantly enhance FourNet's robustness and accuracy through two key mechanisms listed below.

- Control over non-negativity: As detailed in Remark 4.2, the MAE component enables direct control over the upper bound of the potential pointwise loss of non-negativity in $\widehat{g}(\cdot;\theta)$. This control over non-negativity though training of the loss function not only enhances the mathematical integrity of our density estimates but also underscores FourNet's significant practical advantages over traditional Fourier-based and NN-based density estimation methods.
- Enhanced training accuracy in critical regions: The MAE component improves accuracy specifically at critical regions identified through deterministic partition points $\eta_{p_{p=1}}^P$. These points target critical areas, such as those with convexity changes and peaks, in both the real ($\mathrm{Re}_G(\cdot)$) and imaginary ($\mathrm{Im}_G(\cdot)$) components of $G(\cdot)$. This focus ensures precise local fits, thereby complementing L_2-error minimization. This strategy not only aims for an optimal overall fit across the entire domain of $G(\cdot)$ but also prioritizes precision at points critical to FourNet's effectiveness. It aligns with our overarching goal of minimizing L_2-errors, crucial for subsequent L_2-error analysis. Further details on the selection of $\{\eta_p\}_{p=1}^P$ are discussed in Subsection 5.2

We conclude that, for deep NNs, the function $\widehat{g}(\cdot;\theta)$ is expressed as a composition of functions. However, computing its Fourier transform can be very complex, as noted by [3]. Yet, our extensive numerical experiments have demonstrated that a single-layer FFNN possesses remarkable estimation capabilities.

Remark 3.7 (Truncation error in the Fourier domain). Note that, given the boundedness of the parameter space Θ, both $|\text{Re}_{\widehat{G}}(\cdot;\theta)|$ and $|\text{Im}_{\widehat{G}}(\cdot;\theta)|$, $\theta \in \Theta$, are in $L_1(\mathbb{R})$. We also recall that both $|\text{Re}_G(\cdot)|$ and $|\text{Im}_G(\cdot)| \in L_1(\mathbb{R})$. Furthermore, since both $g(\cdot)$ and $\widehat{g}(\cdot;\theta) \in \Sigma(\phi)$, for any $\theta \in \Theta$, are in $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$, by the Plancherel Theorem (3.3), both $|G(\cdot)|$ and $|\widehat{G}(\cdot;\theta)|$ are in $L_2(\mathbb{R})$. Therefore, for any given $\epsilon > 0$, $\exists \eta' > 0$ such that, with $f \in \{\text{Re}_G, \text{Im}_G, \text{Re}_{\widehat{G}}(\cdot;\theta), \text{Im}_{\widehat{G}}(\cdot;\theta)\}$ and $h \in \{G(\cdot), \widehat{G}(\cdot;\theta)\}$,

$$\int_{\mathbb{R}\setminus[-\eta',\eta']} |f(\eta)|\, d\eta < \epsilon, \quad \int_{\mathbb{R}\setminus[-\eta',\eta']} |h(\eta)|^2\, d\eta < \epsilon, \quad \forall \theta \in \Theta. \tag{3.14}$$

That is, the truncation error in the Fourier domain can be made arbitrarily small by choosing $\eta' > 0$ sufficiently large. In practice, given a closed-form expression for $G(\cdot)$, η' can be determined numerically, as illustrated in Subsection 6.1.

Remark 3.8 (Gaussian mixtures). There are two potential interpretations of our methodology. The first sees $\widehat{g}(x;\theta)$ in (3.8) as an FFNN approximation of the exact transition density $g(\cdot)$, and its parameters are learned by minimizing the loss function $\text{Loss}_P(\theta)$ (defined in (3.11)). Alternatively, $\widehat{g}(x;\theta)$ in (3.8) can be written as

$$\widehat{g}(x;\theta) = \sum_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(x-\mu_n)^2}{2\sigma_n^2}\right), \quad \mu_n = -\frac{b_n}{w_n},\ \sigma_n^2 = \frac{1}{2w_n^2}. \tag{3.15}$$

This can be essentially viewed as a Gaussian mixture with N components [41], where the n-th Gaussian component has mean $\mu_n = -\frac{b_n}{w_n}$ and variance $\frac{1}{2w_n^2}$. Unlike traditional Gaussian mixtures, the centers of the component distributions are not predetermined but are also learned through training. Finally, it is worth noting that the set of all normal mixture densities is dense in the set of all density functions under the L_1-metric (see [32]), hence a mixture of Gaussian like in (3.15) can be used to estimate any unknown density function.

4 Error analysis We denote by $\widehat{\theta}$ the parameter learned from training the loss function $\text{Loss}_P(\theta)$, and refer to $\widehat{g}(\cdot;\widehat{\theta})$ as the corresponding estimated transition density. We aim to derive an upper bound for the L_2 estimation error $\int_{\mathbb{R}} |g(x) - \widehat{g}(x;\widehat{\theta})|^2\, dx$. By the Plancherel Theorem (3.3), we have $\int_{\mathbb{R}} |g(x) - \widehat{g}(x;\widehat{\theta})|^2\, dx = \int_{\mathbb{R}} \left|G(\eta) - \widehat{G}(\eta;\widehat{\theta})\right|^2 d\eta$. This underscores the unique advantages of the proposed approach: error analysis is better suited to the Fourier domain than to the spatial domain, as we can directly benefit from the loss function $\text{Loss}_P(\theta)$, which is designed specifically for this domain.

In our error analysis, we require $C' := \sup_{\eta,\theta} |\partial |G(\eta) - \widehat{G}(\eta;\theta)|^2 / \partial \eta| < \infty$, for all $\eta \in [-\eta', \eta']$ and $\theta \in \Theta$. Given that Θ is bounded and thus $\widehat{G}(\eta;\theta)$ possesses a bounded first derivative, the requirement for $C' < \infty$ is that $G(\eta)$ also has a bounded first derivative. This leads us to the assumption that the random variable associated with the density $g(\cdot)$ is absolutely integrable. We also recall C_0 and C_1 from (3.10). We now present an error analysis of the FourNet method in Lemma 4.1 below.

LEMMA 4.1. *As per Remark 3.7, for a given $\epsilon_1 > 0$, let the truncated Fourier domain $[-\eta', \eta']$ be such that, with $f \in \{Re_G(\cdot), Im_G(\cdot), Re_{\widehat{G}}(\cdot;\theta), Im_{\widehat{G}}(\cdot;\theta)\}$ and $h \in \{G(\cdot), \widehat{G}(\cdot;\theta)\}$,*

$$\int_{\mathbb{R}\setminus[-\eta',\eta']} |f(\eta)|\, d\eta < \epsilon_1, \quad \int_{\mathbb{R}\setminus[-\eta',\eta']} |h(\eta)|^2\, d\eta < \epsilon_1,\ \forall \theta \in \Theta. \tag{4.1}$$

Suppose that the parameter $\widehat{\theta}$ learned by training the empirical loss function $Loss_P(\theta)$,

$\theta \in \widehat{\Theta}$, defined in (3.11), is such that

$$\left| \frac{1}{P} \sum_{p=1}^{P} \left| G(\eta_p) - \widehat{G}(\eta_p; \widehat{\theta}) \right|^2 + R_P(\widehat{\theta}) \right| < \epsilon_2, \tag{4.2}$$

and the regularization term $R_P(\widehat{\theta}) < \epsilon_3$, where $\epsilon_2, \epsilon_3 > 0$. Then, we have

$$\int_{\mathbb{R}} \left| g(x) - \widehat{g}(x; \widehat{\theta}) \right|^2 \, dx < \frac{1}{2\pi} \left(4\epsilon_1 + C_1(\epsilon_2 + \epsilon_3) + \frac{C' C_1^2}{P} \right). \tag{4.3}$$

Proof of Lemma 4.1. Applying the error bound for the composite left-hand-side quadrature rule on a non-uniform partition gives

$$\Big| \sum_{p=1}^{P} \delta_p \big| G(\eta_p) - \hat{G}(\eta_p; \theta) \big|^2 - \int_{[-\eta', \eta']} \big| G(\eta) - \hat{G}(\eta; \theta) \big|^2 \, d\eta \Big| \leq C' P (\delta_{\max})^2 = C/P. \tag{4.4}$$

noting $\delta_{\max} = C_1/P$, as in (3.10), where $C = C' C_1^2$. Therefore,

$$\begin{aligned} \int_{\mathbb{R}} \left| G(\eta) - \widehat{G}(\eta; \widehat{\theta}) \right|^2 d\eta &\overset{\text{(i)}}{=} \int_{\mathbb{R} \setminus [-\eta', \eta']} \left| G(\eta) - \widehat{G}(\eta; \widehat{\theta}) \right|^2 d\eta + \int_{[-\eta', \eta']} \left| G(\eta) - \widehat{G}(\eta; \widehat{\theta}) \right|^2 d\eta \\ &\overset{\text{(ii)}}{<} 4\epsilon_1 + \frac{C_1}{P} \sum_{p=1}^{P} \left| G(\eta_p) - \widehat{G}(\eta_p; \widehat{\theta}) \right|^2 + C/P \\ &\overset{\text{(iii)}}{<} 4\epsilon_1 + C_1 \big(\epsilon_2 + \epsilon_3 \big) + C/P. \end{aligned} \tag{4.5}$$

Here, from (i) to (ii), we respectively bound the first and the second terms in (i) by $4\epsilon_1$, using (4.1), and by $C/P + \sum_{p=1}^{P} \delta_{\max} \left| G(\eta_p) - \widehat{G}(\eta_p; \widehat{\theta}) \right|^2$, using (4.4) with $\theta = \widehat{\theta}$, noting that $\delta_p \leq \delta_{\max} = C_1/P \ \forall p$. In (iii), we use (4.2) and $R_P(\widehat{\theta}) < \epsilon_3$ to bound the second term in (ii) by $C_1(\epsilon_2 + \epsilon_3)$. Using the Plancherel Theorem (3.3) and (4.5) gives

$$2\pi \int_{\mathbb{R}} \left| g(x) - \widehat{g}(x; \widehat{\theta}) \right|^2 \, dx < 4\epsilon_1 + C_1(\epsilon_2 + \epsilon_3) + \frac{C' C_1^2}{P},$$

noting $C = C' C_1^2$. Rearrange the above gives (4.3). This completes the proof. ∎

Lemma 4.1[Eqn. (4.3)] decomposes the upper bound for the L_2 estimation error into several error components listed below.

- Truncation error (Fourier domain): This arises from truncating the sampling domain from $\mathbb{R}$ to $[-\eta', \eta']$. It is bounded by ϵ_1 (see (4.1)), contributing $4\epsilon_1/(2\pi)$ to the derived error bound in (4.3).
- Training error: This error results from the deviation of the learned parameters $\widehat{\theta}$ from minimizing the empirical loss function, which includes the MSE error and MAE regularization. Its total contribution to the error bound in (4.3) is $C_1(\epsilon_2 + \epsilon_3)/(2\pi)$.
- Sampling error: This is caused by the use of a finite set of P data points in training. This error, captured as numerical integration error, is represented as $\frac{C' C_1^2}{2\pi P}$ in the error bound in (4.3).

Remark 4.2 (Nonnegativity of $\widehat{g}(\cdot; \widehat{\theta})$.). We now investigate the potential loss of nonnegativity in $\widehat{g}(\cdot; \widehat{\theta})$, where $\widehat{\theta}$ is learned as per Lemma 4.1. To this end, we use $|\min(\widehat{g}(x; \widehat{\theta}), 0)|$, for an arbitrary $x \in \mathbb{R}$, as a measure of this potential pointwise loss. Following similar steps (i)-(ii) of (4.4), noting $R_P(\widehat{\theta}) < \epsilon_3$, we have

$$\int_{\mathbb{R}} \left(|\mathrm{Re}_G(\eta) - \mathrm{Re}_{\widehat{G}}(\eta; \widehat{\theta})| + |\mathrm{Im}_G(\eta) - \mathrm{Im}_{\widehat{G}}(\eta; \widehat{\theta})| \right) d\eta < 4\epsilon_1 + C_1 \epsilon_3 + C' C_1^2 / P.$$

Hence, $|\min(\widehat{g}(x;\widehat{\theta}),0)| \le |g(x)-\widehat{g}(x;\widehat{\theta})| = \frac{1}{2\pi}\big|\int_{\mathbb{R}} e^{-i\eta x}(G(\eta)-\widehat{G}(\eta;\widehat{\theta}))d\eta\big| = \dots$

$$\dots \le \frac{1}{2\pi}\int_{\mathbb{R}} |\mathrm{Re}_G(\eta)-\mathrm{Re}_{\widehat{G}}(\eta;\widehat{\theta})| + |\mathrm{Im}_G(\eta)-\mathrm{Im}_{\widehat{G}}(\eta;\widehat{\theta})|\, d\eta < \frac{1}{2\pi}\big(4\epsilon_1 + C_1\epsilon_3 + \frac{C'C_1^2}{P}\big).$$

As demonstrated above, the derived upper bound for $|\min(\widehat{g}(x;\widehat{\theta}),0)|$ can be decomposed into several error components: truncation error ($4\epsilon_1/(2\pi)$), the MAE regularization term ($C_1\epsilon_3/(2\pi)$), and sampling error ($C'C_1^2/(2\pi P)$).

Remark 4.3 (Multi-dimensional). We now extend Lemma 4.1 and Remark 4.2 to the general d-dimensional case, where $d \ge 1$. In this context, the target density is $g(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^d$, and $\widehat{g}(\boldsymbol{x};\widehat{\theta})$, with $\widehat{\theta} \in \widehat{\Theta} \subseteq \Theta \subset \mathbb{R}^{(d+2)N}$, is the estimated density.

We denote by $\boldsymbol{\eta} = [\eta^{(1)}, \eta^{(2)}, \dots, \eta^{(d)}]$ the d-dimensional Fourier domain vector, restricted to $[-\eta', \eta']^d$, where $0 < \eta' < \infty$ and is sufficiently large. For ease of exposition, we assume there are $P^{1/d}$ partition points in each dimension, resulting in P total data points for training. The same (potentially non-uniform) partition is used across all dimensions, with partition intervals satisfying $\delta_{\min} = C_0/P^{1/d}$ and $\delta_{\max} = C_1/P^{1/d}$, where $C_0, C_1 > 0$ are finite constants independent of P and d. Additionally, we assume

$$C' := \sup_{\boldsymbol{\eta},\theta} \Big|\nabla_{\boldsymbol{\eta}} \Big(|G(\boldsymbol{\eta}) - \widehat{G}(\boldsymbol{\eta};\theta)|^2\Big)\Big| < \infty,$$

where $\nabla_{\boldsymbol{\eta}}(\cdot)$ is the gradient with respect to $\boldsymbol{\eta}$. In this case, the loss function becomes

$$\mathrm{Loss}_P(\theta) = \frac{1}{P}\sum_{p=0}^{P} \Big|G(\boldsymbol{\eta}_p) - \widehat{G}(\boldsymbol{\eta}_p;\theta)\Big|^2 + R_P(\theta), \tag{4.6}$$

where $R_P(\theta) = \frac{1}{P}\sum_{p=1}^{P}\big(\big|\mathrm{Re}_G(\boldsymbol{\eta}_p) - \mathrm{Re}_{\widehat{G}}(\boldsymbol{\eta}_p;\theta)\big| + \big|\mathrm{Im}_G(\boldsymbol{\eta}_p) - \mathrm{Im}_{\widehat{G}}(\boldsymbol{\eta}_p;\theta)\big|\big)$.

Remark 3.7 also extends to the general multi-dimensional case: for a given $\epsilon_1 > 0$, we can find $[-\eta', \eta']^d$ such that the error from truncating the sampling domain from $\mathbb{R}^d$ to $[-\eta', \eta']^d$ is bounded by ϵ_1. Suppose that the parameter $\widehat{\theta}$ learned by training $\mathrm{Loss}_P(\theta)$, $\theta \in \widehat{\Theta}$, defined in (4.6), is such that $\mathrm{Loss}_P(\widehat{\theta}) < \epsilon_2$, with the regularization term $R_P(\widehat{\theta}) < \epsilon_3$, where $\epsilon_2, \epsilon_3 > 0$. Then, we have

$$\int_{\mathbb{R}^d} \big|g(\boldsymbol{x}) - \widehat{g}(\boldsymbol{x};\widehat{\theta})\big|^2\, d\boldsymbol{x} < \frac{1}{2\pi}\big(4\epsilon_1 + C_1(\epsilon_2+\epsilon_3) + \frac{C'C_1^{d+1}}{2P^{1/d}}\big). \tag{4.7}$$

The potential loss of nonnegativity in $\widehat{g}(\cdot;\widehat{\theta})$, i.e. $|\min(\widehat{g}(\boldsymbol{x};\widehat{\theta}),0)|$, is bounded by

$$|\min(\widehat{g}(\boldsymbol{x};\widehat{\theta}),0)| \le \frac{1}{2\pi}\big(4\epsilon_1 + C_1\epsilon_3 + \frac{C'C_1^{d+1}}{2P^{1/d}}\big). \tag{4.8}$$

When $d = 1$, we recover the bounds presented in Lemma 4.1 and Remark 4.2. The main distinction between the bounds for the one- and multi-dimensional cases arises from the error introduced by the composite left-hand quadrature rule.

We emphasize that the explicit quantification of bounds for L_2 estimation error and the potential loss of nonnegativity in the estimated transition density, identified and controlled through truncation, training, and sampling errors, highlights FourNet's significant practical advantages. The rigorous analysis of these practical bounds validates our methodology's robustness and ensures its applicability in real-world settings. In addition, the derived bounds reflect the impact of the dimensionality d, as seen with the term $\frac{C_1^{d+1}}{P^{1/d}}$ in (4.7)-(4.8).

The presented analysis offers an in-depth insight into factors influencing the quality of FourNet's approximation for the transition density $g(\cdot)$. Crucially, it also draws attention to the significance of the coefficients preceding each error component. These coefficients act as markers for the worst-case amplification of individual error components, either in the overall L_2 estimation error or in potential loss of nonnegativity. Consequently, they serve as signposts, guiding us towards areas where we should focus our efforts for more efficient training and, consequently, reduced error.

Of all components, C' attracts special attention. This value is directly related to the oscillatory behavior of $G(\cdot)$, highlighting challenges in approximation. Therefore, curtailing C' is an important step toward improving FourNet approximation's quality. In the subsequent section, we discuss a straightforward a linear transformation on the input domain which can effectively temper the oscillatory nature of $G(\cdot)$, thereby resulting in significantly improved approximation's quality.

5 Training

We now discuss FourNet's data sampling and training algorithms.

5.1 Linear transformation

As a starting point for subsequent discussions, we consider a random variable X and denote the characteristic function of the random variable X by $G_X(\eta) = \mathbb{E}[e^{i\eta X}]$, assumed to be available in closed-form. We analyze $\mathrm{Re}_{G_X}(\eta)$ and $\mathrm{Im}_{G_X}(\eta)$ under two scenarios: the Heston model [23] ($X = \ln(S_T)$) and the Kou model [31] ($X = \ln(S_T/S_0)$), where S_t represents the underlying asset price at time $t \in [0, T]$. Given that $\mathrm{Im}_{G_X}(\eta)$ exhibits similar behaviors across both models, our detailed analysis will focus on $\mathrm{Re}_{G_X}(\eta)$.

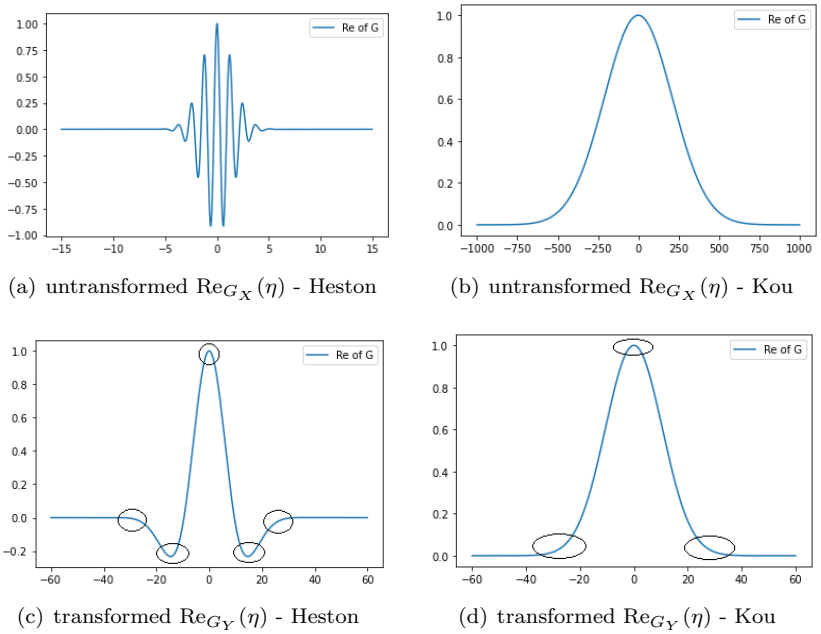
Figures 5.1-(a) and (b) display $\mathrm{Re}_{G_X}(\eta)$ for these cases, with data from Tables 6.6 and 6.18. The Heston model exhibits rapid oscillations within a small domain (about $[-10, 10]$), whereas the Kou model, with a simple oscillation pattern, presents a very large domain (approximately $[-1000, 1000]$). Both situations pose significant challenges during neural network training: rapid oscillations can lead to numerous local minima and erratic gradients, while large domains can compromise training efficiency.

Motivated by these challenges, we propose a linear transformation $Y = aX + c$ as a mechanism for adjusting oscillations and controlling domain sizes, resulting in $G_Y(\eta) = e^{i\eta c} G_X(a\eta)$. To maintain positive direction and scale in the transformation, a is constrained to be greater than zero. It modulates oscillation frequency and domain size: $a < 1$ reduces frequency and expands the domain, while $a > 1$ compresses it. The parameter c, a small real number, adjusts the phase of $G_X(a\eta)$: positive c shifts the phase forward, negative c backward. We recommend a c range from $[-1, 1]$, allowing for significant yet manageable phase shifts across typical η values [44].

Since the overall effectiveness of the transformation heavily depends on the interaction between a and c and the properties of $G_X(\eta)$–which are highly model-dependent–empirical testing is essential to determine suitable parameter settings.

To illustrate, for the Heston model, $a = 0.15$ and $c = -0.6$ reduce oscillation frequency, making $\mathrm{Re}_{G_Y}(\cdot)$ more amenable to NN learning despite a slightly expanded domain $[-60, 60]$ (Figure 5.1-(c)). In the Kou model, $a = 20$ (and $c = 0$) significantly compresses the domain to $[-50, 50]$, maintaining the same oscillation pattern, which enhances training efficiency (Figure 5.1-(d)). The critical regions of $\mathrm{Re}_{G_Y}(\cdot)$ for both models are highlighted, with similar behaviors observed for $\mathrm{Im}_{G_Y}(\cdot)$.

To improve training efficiency further, judicious allocation of sampling data points in crucial areas of both the $\mathrm{Re}_{G_Y}(\cdot)$ and $\mathrm{Im}_{G_Y}(\cdot)$ is essential, a strategy to be elaborated in the following subsection.



(a) untransformed $\mathrm{Re}_{G_X}(\eta)$ - Heston (b) untransformed $\mathrm{Re}_{G_X}(\eta)$ - Kou

(c) transformed $\mathrm{Re}_{G_Y}(\eta)$ - Heston (d) transformed $\mathrm{Re}_{G_Y}(\eta)$ - Kou

Fig. 5.1: Comparisons between $G_X(\eta)$ and $G_Y(\eta)$, where $Y = aX + c$; Heston model - (a) and (c): $a = 0.15$ and $c = -0.6$; Kou model - (b) and (d): $a = 20$ and $c = 0$; critical regions of Re_{G_Y} are highlighted; the behaviour of $\mathrm{Im}_{G_X}(\eta)$ is similar (not shown).

Remark 5.1. Unless otherwise stated, throughout our discussion, the characteristic function $G(\cdot)$ employed for the loss function $\mathrm{Loss}_P(\theta)$ (as defined in (3.11)) corresponds to potentially linearly transformed characteristic function. Specifically, $G(\cdot) = G_Y(\cdot)$, where $Y = aX + c$, where a and c are known real constants. Let $\widehat{g}_Y(y;\widehat{\theta}) = \sum_{n=1}^{N} \widehat{\beta}_n \phi\left(\widehat{w}_n y + \widehat{b}_n\right)$, where $\phi = e^{-x^2}$, be a Fourier-trained FFNN transition density. We can recover the estimated transition density for the random variable X by simply using $\widehat{g}_X(x;\widehat{\theta}) = \widehat{g}_Y(ax + c;\widehat{\theta})\ \left|\frac{d}{dx}(ax + c)\right| = |a|\ \widehat{g}_Y(ax + c;\widehat{\theta})$.

5.2 Sampling data and MAE regularization Given our prior knowledge of the (potentially linearly transformed) characteristic function $G(\eta)$ in its closed-form, we strategically concentrate spatial sampling points $\{\eta_p\}_{p=1}^{P}$ towards critical regions of $G(\eta)$. To identify critical regions, we use symbolic computation to derive the first and second partial derivatives, as well as Hessians for multi-dimensional scenarios, of both the real and imaginary parts of $G(\eta)$. Critical points and inflection points are determined through these derivatives, with numerical methods applied when closed-form solutions are infeasible.[3] For visualization, refer to Fig. 5.1 (c) and (d), which highlight critical regions (in circles) for the Heston and Kou models. This methodology allows us to strategically focus our sampling on areas of convexity change, peaks, and other significant features of $\mathrm{Re}_G(\eta)$ and $\mathrm{Im}_G(\eta)$. Such partitioning of the truncated sampling domain $[-\eta', \eta']$ is achieved through a mapping function, such as the `sinh`$(\cdot)$-based function, which transforms uniform grids into non-uniform ones

[3]For overly complex forms of $G(\eta)$, we utilize Python to approximate its real/imaginary parts and visually verify the locations of critical regions.

with a concentration of points in these critical regions. It is noteworthy that similar methodologies for point construction have found successful applications as evidenced in [53, 8, 10]. A partitioning scheme that addresses such scenarios with multiple concentration points is presented in Appendix A. We emphasize that a randomly sampled dataset of $\{\eta_p\}_{p=1}^P$ might inadequately cover these crucial regions, often requiring a significantly larger dataset for the same precision.

With our strategically defined set $\{\eta_p\}_{p=1}^P$ in place, we emphasize the role of the MAE regularization $R_P(\theta)$ in our optimization process. It allows the optimization to focus on concentrating efforts to reduce discrepancies specifically in critical regions while potentially allowing for some discrepancies in less essential areas. Through this, we aim to strike a balance between precision and generalization, thereby curbing potential over-fitting. Our comprehensive numerical tests, presented in Section 6, suggest that this combined approach - strategic sampling based on $G(\cdot)$ characteristics and employing MAE regularization (3.12) - is efficient and robust.

5.3 Training considerations We briefly describe key considerations in FourNet's training the $\text{Loss}_P(\cdot)$ to obtain the empirical minimizer $\widehat{\theta}^*$. The training of FFNNs is divided into two main stages: the rapid exploration phase and the refinement phase. The initial phase seeks to find a good set of initial weights for the FFNN and fine-tune the baseline learning rate, as these initial weights significantly impact the convergence and accuracy of the training. The refinement phase focuses on further perfecting these weights, often necessitating reduced learning rates to achieve meticulous updates. Due to different focuses of the two stages, choosing the right optimizer for each phase is essential. The Adaptive Moment Estimation (Adam) [29] and AMSGrad with a Modified Stochastic Gradient [54] are standout candidates.

Figure 5.2 presents a visual comparative analysis of the performance of these optimizers is compared in terms of reducing the empirical loss function $\text{Loss}_P(\cdot)$ over a series of training epochs for the case of the Heston model. As illustrated therein, AMSGrad achieves a smoother and steeper reduction in the $\text{Loss}_P(\cdot)$ compared to Adam, especially at higher initial learning rates. However, as the epochs progress, Adam tends to surpass AMSGrad. Our proposed methodology suggests employing AMSGrad during the rapid exploration phase and switching to Adam during the refinement phase.

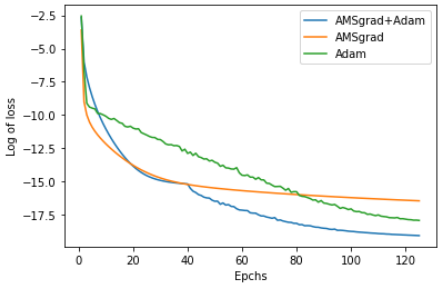


Fig. 5.2: Comparisons among AMSgrad+Adam, Adam, and AMSgrad for the loss function $\text{Loss}_P(\cdot)$ corresponding to Figures 5.1 (c) and (d).

Putting everything together, a single-layer FFNN algorithm for estimating the transition density by learning its Fourier transform is given in Algorithm 5.1.

The computational complexity of Algorithm 5.1 is primarily determined by the training of the single-layer FFNN, involving both forward and backward passes. During the forward pass, the NN computes outputs using operations like matrix-vector multiplications and activation function evaluations, with complexity proportional to $N(d+3)$ (flops), assuming a fixed batch size. The backward pass, crucial for gradient computation and parameter updates using AMSGrad and Adam optimizers, mirrors this complexity. Thus, for P data points and H epochs in both the forward and backward passes, the total complexity is $\mathcal{O}\big(2PHN(d+3)\big)$ (flops) or $\mathcal{O}(PHNd)$ (flops).

Algorithm 5.1 Algorithm for approximating the transition density function $g(\cdot)$ using an FFNN trained in the Fourier domain, given a closed-form expression of the Fourier transform $G(\cdot)$

1: using a closed-form expression of $G(\cdot)$ and numerical integration to find sufficiently large η' as per (3.14);
2: initialize N (number of neurons), P (number of samples); generate $\{\eta_p\}_{p=1}^{P}$ on $[-\eta', \eta']$ using a non-uniform partitioning algorithm (see Algorithm A.1);
3: use AMSgrad optimizer in first training stage to find a good set of initial weights and fine-tune the baseline learning rate;
4: use Adam optimizer in the second training stage
5: construct $\widehat{g}(\cdot, \widehat{\theta})$ with $\widehat{\theta} \in \arg\min_{\theta \in \widehat{\Theta}} \text{Loss}_N(\theta)$, where $\text{Loss}_N(\theta)$ is defined in (3.11);

6 Numerical experiments In this section, we demonstrate FourNet's accuracy and versatility through extensive examples. To measure the accuracy of FourNet, we define several (empirical) metrics. Specifically, the closeness of two elements f_1 and f_2 of $L_p(\mathbb{R})$, $p \in \{1, 2\}$, is measured by $L_p(f_1, f_1) = \int_{[-A,A]} |(f_1(x) - f_2(x))|^p \, dx$, for $A > 0$ sufficiently large. In addition, the Maximum Pointwise Error (MPE) is defined by $\text{MPE}(f_1, f_2) = \max_{1 \le k \le K} |f_1(x_k) - f_2(x_h)|$, where $\{x_k\}_{k=1}^{K}$ is the set of evaluation points. Among these, L_2-error stands out as the principal metric, underscored by the L_2 error analysis presented in Section 4.

In our experiments, unless otherwise stated, all integrals, including those appear in pricing an option, are computed using adaptive Gauss quadrature rule (based on QUADPACK library in Fortran 77 library, `quad` function in Python).

6.1 Setup and preliminary observations Informed by Remarks 3.7 and 4.2, for all numerical experiments carried out in this paper, the sampling domain $[-\eta', \eta']$ (in the Fourier space), the number of samples P are chosen sufficiently large. Specifically, in computing a sufficiently large η', given a closed-form expression for $G(\cdot)$, we perform numerical integration to estimate η' such that (3.14) corresponding to $G(\cdot)$ is satisfied for a tolerance $\epsilon_1 = 10^{-7}$. That is, with $D = \mathbb{R} \setminus [-\eta', \eta']$, we have

(6.1)
$$\int_D |\text{Re}_G(\eta)| \, d\eta < \epsilon_1, \quad \int_D |\text{Im}_G(\eta)| \, d\eta < \epsilon_1, \quad \int_D |G(\eta)|^2 \, d\eta < \epsilon_1.$$

This typically results in $[-\eta', \eta'] = [-60, 60]$ for all models considered hereafter.

	1-D ($d = 1$)	2-D ($d = 2$)
N	45	45
P	10^6	10^6
# $epochs_1$	5	6
# $epochs_2$	100	40
l_1	0.0015	0.04
l_2	0.0012	0.00025
$batchsize$	1024	1024
time (mins)	3.2	22.5

Table 6.1: Hyperparameters for NN training with typical training times.

The training setup utilizes a system equipped with an Intel Core i7-13705H Processor, operating on Windows 11 with 32GB of memory and 1TB of storage. The environment runs Python 3.10 and TensorFlow 2.8. Detailed hyperparameters for all experiments are outlined in Table 6.1, which specifies the number of epochs and learning rates for both the rapid exploration phase (# $epochs_1$ and l_1) and the refinement phase (# $epochs_2$ and l_2). All datasets feature one-dimensional inputs ($d = 1$), except for the two-dimensional Merton jump-diffusion process discussed in

Section 6.3. The table also presents typical total training times for both phases, aggregated over 30 runs.

The parameters $\widehat{\theta}$ is learned through training, satisfying $\text{Loss}_P(\widehat{\theta}) \leq 10^{-6}$. This implies that the MAE regularization term $R_P(\theta)$, as defined in (3.12), is less than 10^{-6}. We observe that the measure for loss of non-negativity $|\min(\widehat{g}(\cdot;\widehat{\theta}), 0)|$ is about 10^{-7}, negligible for all practical purposes. For comparison, we evaluate the bound $\varepsilon = \frac{1}{2\pi}\big(4\epsilon_1 + C_1\epsilon_3 + \frac{C'C_1^2}{P}\big)$, as presented in Remark 4.2. We take $\epsilon_1 = 10^{-7}$ (in accordance with (6.1)), and $\epsilon_3 = 10^{-6}$, given that $\text{Loss}_P(\widehat{\theta}) \leq 10^{-6}$). Considering a uniform partition, we have $\delta_p = 120/P$ for all p. Consequently, $C_1 \geq 120$. Using a conservative estimate, we take $C_1 = 120$, yielding $C_1\epsilon_3 \approx 10^{-4}$ and $\frac{C'C_1^2}{P} \approx C'10^{-2}$. Our numerical findings suggest a notable reduction in the loss of non-negativity when the linear transformation highlighted in Subsection 5.1 is used. This transformation diminishes C', suggesting that $\frac{C'C_1^2}{P} \approx C'10^{-2}$ is the primary contributing term.

We now explore FourNet's accuracy in estimating transition densities a broad array of dynamics commonly encountered in quantitative finance. Subsequently, we will focus on its application for pricing both European and Bermudan options.

6.2 Transition densities

6.2.1 Exponential Lévy processes We select models that are well-known within the domain of exponential Lévy processes, where the Lévy-Khintchine formula provides a clear representation for the characteristic function $G(\cdot)$ as detailed in [28]. As example, we focus on the Merton jump-diffusion model, introduced by [42], and the CGMY model as proposed by [6]. It's worth noting that the CGMY model can be seen as an extension of the Variance-Gamma model, originally presented in [39]. Additionally, while we conducted tests on the Variance-Gamma model and the Kou jump-diffusion model [31], FourNet consistently proved to be very accurate. In fact, the outcomes from these tests align so closely with those of the highlighted models that we have chosen not to detail them in this subsection for the sake of brevity.

In exponential Lévy processes, with $\{S_t\}_{t=0}^T$ being the price process, the process $\{X_t\}_{t=0}^T$, where $X_t = \ln(S_t/S_0)$, is a Lévy process. Relevant to our discussions is the fact that the characteristic function of the random variable X_t is $G_X(\eta) = \exp(t\psi(\eta))$ [28]. As in all numerical examples on transition densities presented in this section, we take $t = T$ which is specified below. The characteristic exponent $\psi(\eta)$ for various exponential Lévy processes considered in this paper are given subsequently.

Merton jump-diffusion dynamics [42] In this case, the characteristic exponent $\psi(\eta)$ is given by $\psi(\eta) = i\left(\mu - \frac{\sigma^2}{2}\right)\eta - \frac{\sigma^2\eta^2}{2} + \lambda\left(e^{i\tilde{\mu}\eta - \tilde{\sigma}^2\eta^2/2} - 1\right)$. In this case, a semi-explicit formula for $g(x;T)$ is given by (see [59][Corollary 3.1])

$$g(x;T) = \sum_{k=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^k}{k!} g_{\text{norm}}\big(x; \big(\mu - \frac{\sigma^2}{2} - \lambda\kappa\big)T + k\tilde{\mu}, \sigma^2 T + k\tilde{\sigma}^2\big). \tag{6.2}$$

Here, $\kappa = e^{\tilde{\mu}+\tilde{\sigma}^2/2} - 1$, and $g_{\text{norm}}(x;\mu',(\sigma')^2)$ denotes the probability density function of a normal random variable with mean μ' and variance $(\sigma')^2$. The semi-explicit formula given by (6.2) serves as our reference density against which we validate the estimated transition density produced by FourNet. Computationally, we truncate the infinite series in (6.2) to 15 terms. The approximation error resulting from this truncation is approximately 10^{-20}, which is sufficiently small for all practical intents and purposes.

Parameters	Values
T (maturity in years)	1
S_0 (initial asset price)	100
r (risk free rate)	0.05
σ (volatility)	0.15
λ (jump intensity)	0.1
$\tilde{\mu}$ (mean of jump size)	-1.08
$\tilde{\sigma}$ (std of jump size)	0.4

Table 6.2: Parameters for the Merton jump diffusion dynamics; values are taken from [15][Table 4].

N (# of neurons)	45
$L_1\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	2.4×10^{-04}
$L_2\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	1.4×10^{-09}
$\mathrm{MPE}\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	1.0×10^{-05}
$L_1\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	5.8×10^{-04}
$L_2\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	1.8×10^{-09}
$\mathrm{MPE}\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	2.4×10^{-05}
$L_2\left(g, \widehat{g}\right)$	1.6×10^{-09}

Table 6.3: Estimation errors for the Merton model; parameters from Table 6.2; linear transform in Remark 5.1 used with $(a, c) = (0.6, 0.08)$.

The parameters used for this test case are given in Table (6.2). The linear transform in Remark 5.1 is used with $(a, c) = (0.6, 0.08)$. The number of neurons (N) and L_p/MPE estimation errors by FourNet are presented in Table 6.3, with the principal metric L_2-error highlighted. As evident, FourNet is very accurate with negligible L_2 estimation error (of order 10^{-9}). We note that, without a linear transform, the resulting L_p/MPE estimation errors are much larger. For example, $L_2\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right) = 3.3 \times 10^{-7}$ instead of 1.4×10^{-9} and $L_2\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right) = 1.8 \times 10^{-7}$ vs 1.8×10^{-9}.

CGMY model [6] In this case, the characteristic exponent $\psi(\eta)$ is given by $\psi(\eta) = CG(\eta) + i\eta(r + \varpi)$, where $CG(\eta) = C\Gamma(-Y)\left[(M - i\eta)^Y - M^Y + (G + i\eta)^Y - G^Y\right]$ and $\varpi = -CG(-i)$. Here, $\Gamma(\cdot)$ represents the gamma function. In the CGMY model, the parameter should satisfy $C \geq 0$, $G \geq 0$, $M \geq 0$ and $Y < 2$.

The parameters used for this test case are given in Table (6.4). The linear transform in Remark 5.1 is used with $(a, c) = (0.5, 0.0)$. The number of neurons (N) and L_p/MPE estimation errors by FourNet are presented in Table 6.5, with the principal metric L_2-error highlighted. Again, it is clear that FourNet is very accurate, with negligible L_2 estimation error (of order 10^{-8}).

Parameters	Values
T (maturity in years)	1
S_0 (initial asset price)	100
r (risk free rate)	0.1
C (overall activity)	1
G (exp. decay on right)	5
M (exp. decay on left)	5
Y (finite/infinite activity)	0.5

Table 6.4: Parameters for the CGMY dynamics; values taken from [13][Equation 56].

N (# of neurons)	45
$L_1\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	7.8×10^{-4}
$L_2\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	1.7×10^{-8}
$\mathrm{MPE}\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	3.1×10^{-5}
$L_1\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	2.2×10^{-4}
$L_2\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	1.3×10^{-9}
$\mathrm{MPE}\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	9.2×10^{-6}

Table 6.5: Estimation errors for the CGMY model; parameters from Table 6.4; linear transform in Remark 5.1 is employed with $(a, c) = (0.5, 0.0)$.

6.2.2 Heston and Heston Queue-Hawkes Moving beyond exponential Lévy processes, we first evaluate the applicability of FourNet to the Heston model [23], followed by an investigation of the Heston Queue-Hawkes model, as presented in [2]. For these models, the characteristic functions of the log-asset price, $\ln(S_t)$, over $t \in [0, T]$, are available in closed-form. Despite the non-homogeneous variance features, our focus here is on estimating the transition density of the process $\{\ln(S_t)\}_{t\in[0,T]}$ for Eu-

ropean option pricing. Since these options do not require time-stepping for valuation, we can efficiently estimate transition densities using a single FFNN training session.

Heston model [23] The log-price $\ln(S_t)$ and its variance V_t follow the dynamics

$$d\ln(S_t) = \big(r - \frac{V_t}{2}\big)dt + \sqrt{V_t}\, dW_t^{(1)}, \quad dV_t = \kappa(\bar{V} - V_t)\, dt + \sigma\sqrt{V_t}\, dW_t^{(2)},$$

with $S_0 > 0$ and $V_0 > 0$ given. Here, $\kappa, \bar{V} > 0$, and $\sigma > 0$ are constants representing the mean-reversion rate, the long-term mean level of the variance, and the instantaneous volatility of the variance; $\{W_t^{(1)}\}$ and $\{W_t^{(2)}\}$ are assumed to be correlated with correlation coefficient $\rho \in [-1, 1]$. As presented in [47][Equation 5], the characteristic function of $X_T = \ln(S_T)$ is given by

$$\begin{aligned} G_X^{\text{Heston}}(\eta) = \exp^{ir\eta T} \exp\{i\ln(S_0)\eta\} \left(\frac{e^{\kappa T/2}}{\cosh(dT/2) + \xi\sinh(dT/2)/d}\right)^{2\kappa\bar{V}/\sigma^2} \\ \cdot \exp\left\{-V_0 \frac{(i\eta + \eta^2)\sinh(dT/2)/d}{\cosh(dT/2) + \xi\sinh(dT/2)/d}\right\}, \end{aligned} \tag{6.3}$$

where $d = d(\eta) = \sqrt{(\kappa - \sigma\rho i\eta)^2 + \sigma^2(i\eta + \eta^2)}$ and $\xi = \xi(\eta) = \kappa - \sigma\rho\eta i$.

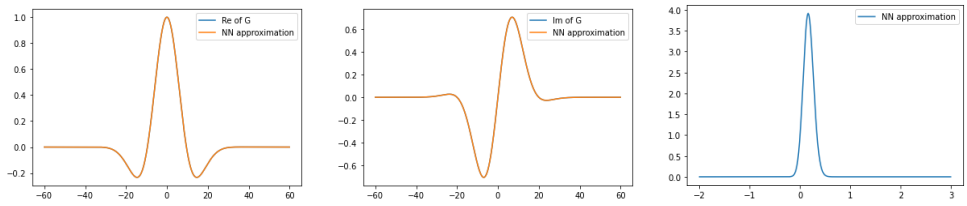
Numerical experiments for the Heston model utilize the parameters listed in Table 6.6 (the Feller condition is met). Estimation errors are documented in Table 6.7, noting the linear transformation Remark 5.1. In Figure 6.1, we display plots of the benchmark real part in (a) and the imaginary part in (b). Corresponding estimations by FourNet are also showcased, with the transition density estimated by FourNet depicted in (c). We emphasize FourNet's outstanding performance, particularly evident in the minimal L_2 estimation error.

Parameters	Values
T (maturity in years)	5
S_0 (initial asset price)	100
r (risk free rate)	0.15
σ (volatility of volatility)	0.3
κ (mean-reversion rate)	3
$\bar{V}$ (mean of volatility)	0.09
V_0 (initial volatility)	0.2
ρ (correlation)	0.4

Table 6.6: Parameters for the Heston model; values taken from [11][Table 1].

N (# of neurons)	45
$L_1\left(\text{Re}_G, \text{Re}_{\widehat{G}}\right)$	1.94×10^{-05}
$L_2\left(\text{Re}_G, \text{Re}_{\widehat{G}}\right)$	6.94×10^{-12}
$\text{MPE}\left(\text{Re}_G, \text{Re}_{\widehat{G}}\right)$	1.18×10^{-06}
$L_1\left(\text{Im}_G, \text{Im}_{\widehat{G}}\right)$	3.91×10^{-05}
$L_2\left(\text{Im}_G, \text{Im}_{\widehat{G}}\right)$	5.22×10^{-11}
$\text{MPE}\left(\text{Im}_G, \text{Im}_{\widehat{G}}\right)$	3.74×10^{-06}

Table 6.7: Estimation errors for the Heston model; parameters from Table 6.6; linear transform in Remark 5.1 is employed with $(a, c) = (0.15, -0.6)$.



(a) Re_G, $\text{Re}_{\widehat{G}}$ (b) Im_G, $\text{Im}_{\widehat{G}}$ (c) $\widehat{g}(\cdot; T)$

Fig. 6.1: Heston model corresponding to Table 6.6 and Table 6.7.

Heston Queue-Hawkes [12, 2] With $t \in [0,T]$, let $t^{\pm} = \lim_{\epsilon \searrow 0}(t \pm \epsilon)$. Informally, t^- (t^+) denotes the instant of time immediately before (after) calendar time t. The risk-neutral Heston Queue-Hawkes dynamics of the stock price are given by [2]:

$$d\left(\frac{S_t}{S_{t^-}}\right) = (r - \mu_Y \lambda_{t^-})\, dt + \sqrt{V_t} dW_t^{(1)} + (\exp(Y_t) - 1)\, d\pi_t,$$
$$dV_t = \kappa(\bar{V} - V_t)dt + \sigma\sqrt{V_t} dW_t^{(2)}.$$

Here, $\{V_t\}$ is the variance process; $\{W_t^{(1)}\}$, $\{W_t^{(2)}\}$ are correlated standard Brownian motions with the constant correlation $\rho \in [-1,1]$; $Y_t \sim \text{Normal}(\mu_Y, \sigma_Y^2)$; $\kappa > 0$, $\bar{V} > 0$ and $\sigma > 0$ are the variance's speed of mean reversion, long-term mean, and volatility of volatility parameters, respectively. Finally, $\{\pi_t\}$ is a counting process with stochastic intensity λ_t satisfying the Queue-Hawkes process: $d\lambda_t = \alpha(d\pi_t - d\pi_t^Q)$, where π_t^Q is a counting process with intensity βQ_t, the constants α, and β respectively are the clustering and expiration rates.

The characteristic function of $X_T = \ln(S_T)$ is given by [2][Equation 6]:

$$G_X^{\text{HQH}}(\eta) = G_X^{\text{Heston}}(\eta)\; G_M(\eta). \tag{6.4}$$

Here, $G_X^{\text{Heston}}(\eta)$ is given in (6.3), and $G_M(\eta)$ is defined as follows

$$G_M(\eta) = e^{\frac{\lambda^* T}{2\alpha}(\beta - \alpha - i\alpha\mu_Y\eta - f(\eta))} \cdot \left(\frac{2f(\eta)}{f(\eta) + g(\eta) + e^{-Tf(\eta)}(f(\eta) - g(\eta))}\right)^{\frac{\lambda^*}{\alpha}}$$
$$\cdot \left(\frac{\left(1 - e^{-Tf(\eta)}\right)(2\beta)}{f(\eta) + g(\eta) + e^{-Tf(\eta)}(f(\eta) - g(\eta))}\right)^{Q_0}.$$

Here, $f(\eta) = \sqrt{(\beta + \alpha(1 + i\eta\mu_Y))^2 - 4\alpha\beta\psi_Y(\eta)}$, $g(\eta) = \beta + \alpha(1 + i\eta\mu_Y)$, ψ_Y is the characteristic function of normal random variable with mean μ_Y and std σ_Y.

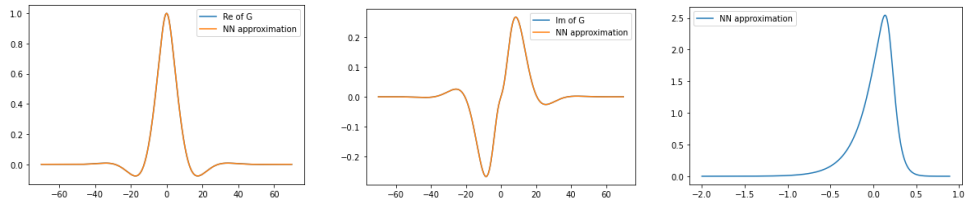
Parameters	Values
T (maturity in years)	1
S_0 (initial asset price)	9
V_0 (initial volatility)	0.0625
$\bar{V}$ (mean of volatility)	0.16
r (risk free rate)	0.1
σ (volatility of volatility)	0.9
Q_0 (initial value of Q_t)	2
α (clustering rate)	2
β (expiration rate)	3
λ^* (baseline jump intensity)	1.1
μ_Y (mean of jump size)	-0.3
σ_Y (std of jump size)	0.4
ρ (correlation)	0.1

Table 6.8: Parameters for the Heston Queue-Hawkes model. values are taken from [2][Table 1].

N (# of neurons)	45
$L_1\left(\text{Re}_G, \text{Re}_{\widehat{G}}\right)$	3.36×10^{-4}
$L_2\left(\text{Re}_G, \text{Re}_{\widehat{G}}\right)$	4.19×10^{-9}
$\text{MPE}\left(\text{Re}_G, \text{Re}_{\widehat{G}}\right)$	2.44×10^{-5}
$L_1\left(\text{Im}_G, \text{Im}_{\widehat{G}}\right)$	3.38×10^{-4}
$L_2\left(\text{Im}_G, \text{Im}_{\widehat{G}}\right)$	4.66×10^{-9}
$\text{MPE}\left(\text{Im}_G, \text{Im}_{\widehat{G}}\right)$	1.87×10^{-5}

Table 6.9: Estimation errors for the Heston Queue-Hawkes model; parameters from Table 6.8; linear transform in Remark 5.1 with $(a,c) = (0.18, -0.31)$ is applied to $G_X^{\text{HQH}}(\cdot)$ in (6.4).

We conduct numerical experiments using the parameters listed in Table 6.8, with estimation errors detailed in Table 6.9, noting the linear transformation Remark 5.1. Figure 6.1, we present several plots for the benchmark real/imaginary part in (a)/(b) and respective results obtained by FourNet, as well as the estimated transition density in (c). Impressively, FourNet demonstrates outstanding L_2 estimation accuracy.



(a) Re_G, $\mathrm{Re}_{\widehat{G}}$ (b) Im_G, $\mathrm{Im}_{\widehat{G}}$ (c) $\widehat{g}(\cdot;T)$

Fig. 6.2: Heston Queue-Hawkes model corresponding to Table 6.8 and Table 6.9.

6.3 Two-dimensional Merton jump-diffusion process We now demonstrate the capability of FourNet to two-dimensional Merton jump-diffusion process [49]. The stock prices follow the risk-neutral dynamics

$$dS_t^{(\ell)} = (r - \lambda\kappa^{(\ell)})S_t^{(\ell)}dt + \sigma^{(\ell)}S_t^{(\ell)}dW_t^{(\ell)} + \left(e^{Y^{(\ell)}} - 1\right)S_t^{(\ell)}d\mathcal{P}_t,\ \ell = 1,2, \tag{6.5}$$

with $S_0^{(\ell)} > 0$ given. Here, $r > 0$ is risk free rate and $\sigma^{(\ell)} > 0$, $\ell = 1,2$, are instantaneous volatility for the ℓ-underlying; $\{W_t^{(1)}\}$ and $\{W_t^{(2)}\}$ are standard Brown motions with correlation $\rho \in [-1,1]$; $\{\mathcal{P}_t\}$ is a Poisson process with a constant finite jump arrival rate $\lambda \geq 0$; $[Y^{(1)}, Y^{(2)}]$ are bivariate normally distributed jump sizes with mean $\tilde{\boldsymbol{\mu}} = [\tilde{\mu}^{(1)}, \tilde{\mu}^{(2)}]$ and covariance matrix for the jump components, denoted by $\widetilde{\boldsymbol{\Sigma}}$, where $\widetilde{\boldsymbol{\Sigma}}^{(\ell,k)} = \tilde{\sigma}^{(\ell)}\tilde{\sigma}^{(k)}\tilde{\rho}^{(\ell,k)}$, $\ell,k \in \{1,2\}$, with $\tilde{\rho}^{(1,2)} = \tilde{\rho}^{(2,1)} = \tilde{\rho} \in [-1,1]$; $\kappa^{(\ell)} = \mathbb{E}[e^{Y_{(\ell)}} - 1]$, $\ell = 1,2$.

For subsequent use, we define $\boldsymbol{\mu} = [\mu^{(1)}, \mu^{(2)}]$, where $\mu^{(\ell)} = (r - \lambda\kappa^{(\ell)} - (\sigma^{(\ell)})^2/2)T$, $\ell \in \{1,2\}$, and covariance matrix for the diffusion components, denoted by $\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}^{(\ell,k)} = \sigma^{(\ell)}\sigma^{(k)}\rho^{(\ell,k)}$, $\ell,k \in \{1,2\}$, with $\rho^{(1,2)} = \rho^{(2,1)} = \rho \in [-1,1]$.

Parameters	Values
T (maturity in years)	1
σ_1 (volatility)	0.12
σ_2	0.15
r (risk free rate)	0.05
K (strike price)	100
ρ (correlation)	0.3
$\tilde{\rho}$ (jump correlation)	-0.2
λ (jump intensity)	0.6
$\tilde{\mu}_1$ (jump size mean)	-0.1
$\tilde{\mu}_2$	0.1
$\tilde{\sigma}_1$ (jump size std)	0.17
$\tilde{\sigma}_2$	0.13

Table 6.10: Parameters for the 2D Merton jump diffusion. Values are taken from [49] [Parameter sets 2].

N (# of neurons)	45
$L_2\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	3.2×10^{-8}
$\mathrm{MPE}\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	8.3×10^{-5}
$L_2\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	2.2×10^{-8}
$\mathrm{MPE}\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	4.6×10^{-5}

Table 6.11: Estimation errors for the 2D Merton jump-diffusion model; parameters from Table 6.10;

The characteristic function of the random variable $\boldsymbol{X}_T = \left[\ln\left(S_T^{(\ell)}/S_0^{(\ell)}\right)\right]$, $\ell = 1,2$, is given by [49][Eqn (6.7)]

$$G_{\boldsymbol{X}}(\boldsymbol{\eta}) = \exp\left(i\boldsymbol{\mu}'\boldsymbol{\eta} - \frac{1}{2}\boldsymbol{\eta}'\boldsymbol{\Sigma}\boldsymbol{\eta}\right)\exp\left(\lambda T\left(\exp\left(i\tilde{\boldsymbol{\mu}}'\boldsymbol{\eta} - \frac{1}{2}\boldsymbol{\eta}'\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\eta}\right) - 1\right)\right). \tag{6.6}$$

In this case, it is convenient to write $\widehat{g}(\boldsymbol{x};\theta)$ in the following form:

$$\widehat{g}(\boldsymbol{x};\theta)=\sum_{n=1}^{N}\beta_n\frac{1}{(2\pi)|\widehat{\boldsymbol{\Sigma}}_n|^{1/2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\hat{\boldsymbol{\mu}}_n\right)\hat{\boldsymbol{\Sigma}}_n^{-1}(\boldsymbol{x}-\hat{\boldsymbol{\mu}}_n)\right).$$

Here, for $n\in N$, $\hat{\boldsymbol{\mu}}_n=[\hat{\mu}_n^{(1)},\hat{\mu}_n^{(2)}]$, $\widehat{\boldsymbol{\Sigma}}_n$ is the covariance matrix, where $\widehat{\boldsymbol{\Sigma}}_n^{(\ell,k)}=\hat{\sigma}_n^{(\ell)}\hat{\sigma}_n^{(k)}\hat{\rho}_n^{(\ell,k)}$, $\ell,k\in\{1,2\}$, with $\hat{\rho}_n^{(1,2)}=\hat{\rho}_n^{(2,1)}=\hat{\rho}_n\in[-1,1]$. The parameters to be learned are: $\{\beta_n,\hat{\mu}_n^{(1)},\hat{\mu}_n^{(2)},\hat{\rho}_n\}$, $n=1,\dots N$. The real and imaginary parts of the Fourier transform of $\widehat{g}(\boldsymbol{x};\theta)$ are given by

$$\mathrm{Re}_{\widehat{G}}(\boldsymbol{\eta})=\sum_{n=1}^{N}\beta_n\cos(\boldsymbol{\eta}'\hat{\boldsymbol{\mu}}_n)\exp\left(\frac{-\boldsymbol{\eta}'\widehat{\boldsymbol{\Sigma}}_n\boldsymbol{\eta}}{2}\right),\ \mathrm{Im}_{\widehat{G}}(\boldsymbol{\eta})=\sum_{n=1}^{N}\beta_n\sin(\boldsymbol{\eta}'\hat{\boldsymbol{\mu}}_n)\exp\left(\frac{-\boldsymbol{\eta}'\widehat{\boldsymbol{\Sigma}}_n\boldsymbol{\eta}}{2}\right).$$

We conduct numerical experiments using the parameters listed in Table 6.10, with estimation errors detailed in Table 6.11. As evident from Table 6.11, FourNet demonstrates impressive L_2 estimation accuracy. Here, 10^3 partition points per dimension are used, totaling 10^6 data points for training ($P=10^6$, as shown in Table 6.1). Comparing the L_2 estimation errors for the two-dimensional and one-dimensional cases (Table 6.11 and Table 6.3), and in view of the multi-dimensional error bound (4.7), it appears that FourNet is robust, accurate, and reliable even in higher dimensions.

6.4 Option pricing We now turn our attention to the application of estimated transition densities produced by FourNet utilized for European and Bermudan option pricing. Recall that $0\le t<t+\Delta t\le T$, where t and Δt are fixed. Typically, in option pricing, we need to approximate a generic convolution integral of the form

$$\begin{aligned}v(x,t)&=e^{-r\Delta t}\int_{\mathbb{R}}v((x'-c)/a,t+\Delta t)g(x-x';\Delta t)\,dx'\\&\approx e^{-r\Delta t}\int_{x_{\min}}^{x_{\max}}v((x'-c)/a,t+\Delta t)\widehat{g}(x-x';\widehat{\theta},\Delta t)\,dx.\end{aligned}\tag{6.7}$$

Here, $v(\cdot,t+\Delta t)$ is the time-$(t+\Delta t)$ condition; $\widehat{g}(x;\widehat{\theta},\Delta t)$ is the estimated transition density obtained through FourNet. As we pointed out in Remark 5.1, $\widehat{g}(x;\widehat{\theta},\Delta t)$ reflects a linear transformation applied to the original density. In light of this, the time-$(t+\Delta t)$ terminal condition must be adjusted correspondingly in the convolution integral, as depicted in (6.7), through $(x'-c)/a$. As noted earlier, this integral is evaluated using adaptive Gauss quadrature rule (based on QUADPACK library in Fortran 77 library, `quad` function in Python). The range $[x_{\min},x_{\max}]$ for numerical integration will be provided for each test case subsequently.

6.4.1 European options For European options, we set $t=0$ and $\Delta t=T$, and $v(x',t+\Delta t)=v(x',T)$ as the payoff function. The strike of the option is given by $E>0$. In the context of exponential Lêvy processes examined in this study, which include the Merton and CGMY dynamics, the European call option payoff function is defined as $v\big((x'-c)/a,T\big)\equiv\big(s_0e^{(x'-c)/a}-E\big)^+$, where E is strike price of the option. For the Heston and Heston Queue-Hawkes models, the European call option payoff is $v\big((s'-c)/a,T\big)=\big(e^{(s'-c)/a}-E\big)^+$.

We provide numerically computed European option prices for the models discussed in the previous section. These are presented in Tables 6.12 (Merton), 6.13 (CGMY), 6.4.1 (Heston), and 6.4.1 (Heston Queue-Hawkes). Option prices are derived using the FourNet-estimated transition density $\widehat{g}(s;\widehat{\theta},\Delta t)$, combined with an

adaptive Gauss quadrature rule (the `quad` function in Python) to evaluate the corresponding convolution integral. These results are displayed under the "FourNet-`quad`" column.

Strike (E)	Ref. [42]	FourNet `quad`	Rel. error
96	14.83787	14.83790	2×10^{-6}
98	13.43922	13.43925	3×10^{-6}
100	12.10782	12.10785	3×10^{-6}
102	10.84925	10.84928	3×10^{-6}
104	9.66805	9.66808	3×10^{-6}

Table 6.12: European call option prices under the Merton model corresponding to Tables 6.2 and 6.3; $[x_{\min}, x_{\max}] = [-4, 1]$.

Strike (E)	Ref. [13] (COS)	FourNet `quad`	Rel. error
96	21.78472	21.78466	3×10^{-6}
98	20.77826	20.77819	3×10^{-6}
100	19.81294	19.81288	3×10^{-6}
102	18.88821	18.88815	3×10^{-6}
104	18.00334	18.00328	3×10^{-6}

Table 6.13: European call option prices under CGMY dynamics corresponding to data from Tables 6.4 and 6.5; $[x_{\min}, x_{\max}] = [-4, 2]$.

Strike (E)	Ref. [23]	FourNet `quad`	Rel. error
96	57.35019	57.35014	1×10^{-6}
98	56.61132	56.61127	1×10^{-6}
100	55.88119	55.88114	1×10^{-6}
102	55.15980	55.15975	1×10^{-6}
104	54.44716	54.44711	1×10^{-6}

Table 6.14: European call option prices under Heston dynamics corresponding to data from Tables 6.6 and 6.7; $[x_{\min}, x_{\max}] = [-4, 1]$.

Strike (E)	Ref. [13] (COS)	FourNet `quad`	Rel. error
7	4.27369	4.27373	1×10^{-5}
8	3.81734	3.81738	1×10^{-5}
9	3.40704	3.40708	1×10^{-5}
10	3.04018	3.04022	1×10^{-5}
11	2.71399	2.71403	1×10^{-5}

Table 6.15: European call option prices under Heston Queue-Hawkes dynamics; corresponding to data from Tables 6.8 and 6.9; $[x_{\min}, x_{\max}] = [-3, 1]$

Benchmark prices are detailed under the "Ref." column. For the Merton and Heston models, these benchmark prices are determined using the analytical solutions from [42] and the method by [23], respectively. For CGMY, and Heston Queue-Hawkes models, reference European option prices are derived from our implementation of the Fourier Cosine (COS) method [13]. The associated relative errors of these approximations are indicated in the "Rel. error" column. Clearly, the FourNet-`quad` method proves highly accurate, showcasing a negligible error (on the order of 10^{-5}).

6.4.2 Bermudan options We present a Bermudan put option written on the underlying following the Merton jump-diffusion model [15]. Unlike European options which can only be exercised at maturity, a Bermudan put option can be exercised at any fixed dates t_m^-, $t_m \in \mathcal{T}$, where $\mathcal{T} \equiv \{t_m\}_{m=1}^M$ is a discrete set of pre-determined early exercise dates. We adopt the convention that no early exercise at time t_0. In this example, the early exercise dates are annually apart, that is, $t_{m+1} - t_m = \delta t = 1$ (year), In addition, the underlying asset pays a fixed dividend amount D at t_m^-. Importantly, given that the transition density needed for pricing is time and spatially homogeneous, and with $\delta t = 1$ year for all intervals, we can efficiently train the FFNN

only once, and apply it across all intervals $[t_m, t_{m+1}]$.

Over each $[t_m, t_{m+1}]$, the pricing algorithm for a Bermudan put option consists of two steps. In Step 1 (time-advancement), we need to approximate the convolution integral (6.7): $v(x, t_m^+) = e^{-r\Delta t}\int_{\mathbb{R}} v((x'-c)/a, t_{m+1})\widehat{g}(x-x';\widehat{\theta},\delta t)\,dx$, for $x \in [x_{\min}, x_{\max}]$, where $x_{\min} < 0 < x_{\max}$, $|x_{\min}|$ and $x_{\max}$ are sufficiently large. In Step 2 (intervention), we impose the condition

$$v(x, t_m) = \max\big(v\big(\ln(\max(e^x - D, e^{x_{\min}})), t_m^+\big), \max(E - e^x, 0)\big). \tag{6.8}$$

Here, E is the strike price, and the expression $\ln(\max(e^x - D, e^{x_{\min}}))$ in (6.8) ensures that the no-arbitrage condition holds, i.e. the dividend paid can not be larger than the stock price at that time, taking into account the localized grid.

Adopting annual early exercise dates, where $\delta t = 1$ year, the transition density $\widehat{g}(\cdot; T = 1)$ as obtained from FourNet (as detailed in Table 6.3 and based on parameters from Table 6.2) is used in Step 1 (time-advancement) above. These parameters and those pertaining to the Bermudan put are given in Table 6.16.

Letting $\{x_q\}_{q=0}^{Q}$ be a partition of $[x_{\min}, x_{\max}]$, we denote by v_q^m a numerical approximation to the exact value $v(x_q, t_m)$, where $t_m \in \mathcal{T} \cup \{t_0\}$. Intermediate value $\{v_q^{m+}\}$, $q = 0, \ldots, Q$, is computed by evaluating the convolution integral in Step 1 via an adaptive Gauss quadrature rule, specifically the `quad` function in Python. The time t_{m+1}-condition $v(\cdot, t_{m+1})$ is given by a linear combination of discrete solutions $\{v_q^{m+1}\}$. For condition (6.8), linear interpolation is then used on $\{v_q^{m+}\}$ to determine the option value $\{v_q^m\}$. Convergence results for the FourNet-quad approach are displayed in Table 6.17, showcasing evident agreement with an accurate benchmark option price taken from [15][Table 5] (finest grid). Here, to estimate the convergence rate of FourNet-quad, we calculate the "change" as the difference in values from coarser to finer partitions (i.e. transitioning from smaller to larger values of Q) and the "ratio" as the quotient of these changes between successive partition.

Parameters	Values
S_0 (initial asset price)	100
r (risk free rate)	0.05
σ (volatility)	0.15
λ (jump intensity)	0.1
$\tilde{\mu}$ (mean of jump size)	-1.08
$\tilde{\sigma}$ (std of jump size)	0.4
T (maturity in years)	10
δt (frequency in years)	1
E (strike)	100
D (dividend)	1

Table 6.16: Parameters for the Bermudan put option; values taken from [15][Table 4].

Q	FourNet-quad	change	ratio
200	24.8323		
400	24.7903	4.2×10^{-2}	
800	24.7838	6.5×10^{-3}	6.7
1600	24.7812	2.6×10^{-3}	2.5
3200	24.7806	6.0×10^{-4}	4.3

Table 6.17: Bermudan put option; parameters from Table 6.16; benchmark price: 24.7807, taken from [15][Table 5, finest grid]; $x_{\min} = \ln(S_0) - 10$, $x_{\max} = \ln(S_0) + 10$.

6.5 Robustness tests This section evaluates FourNet's robustness against the COS method [13] in challenging scenarios, particularly when the transition density approaches a Dirac's delta function as $T \to 0$. Fourier-based methods often struggle in such cases, requiring a very large number of terms to accurately estimate the transition density. To further assess robustness, we introduce an asymmetric heavy-tailed distribution. We test with a very short maturity $T = 0.001$ years, or about 0.26 business days, using the Kou jump-diffusion model [31], known for effectively modeling

the leptokurtic nature of market returns.

Parameters	Values
T (maturity in years)	0.001
S_0 (initial asset price)	100
q_1 (jump-up probability)	0.3445
r (risk free rate)	0.05
σ (volatility)	0.15
λ (jump intensity)	0.1
ξ_1 (jump-up param.)	3.0465
ξ_2 (jump-down param.)	3.0775
E (strike price)	100

Table 6.18: Parameters for the Kou jump diffusion dynamics; values are taken from [15][Table 1].

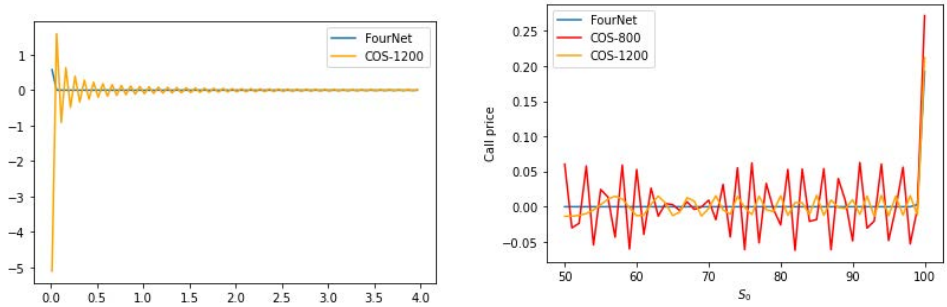
N (# of neurons)	45
$L_1\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	1.06×10^{-03}
$L_2\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	1.86×10^{-08}
$\mathrm{MPE}\left(\mathrm{Re}_G, \mathrm{Re}_{\widehat{G}}\right)$	7.00×10^{-05}
$L_1\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	1.10×10^{-03}
$L_2\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	2.11×10^{-08}
$\mathrm{MPE}\left(\mathrm{Im}_G, \mathrm{Im}_{\widehat{G}}\right)$	3.16×10^{-05}

Table 6.19: Estimation errors for the Kou model; parameters from Table 6.18; linear transform in Remark 5.1 is employed with $(a, c) = (20, 0)$.

For this model, the characteristic function of the random variable $X_t = \ln(S_t/S_0)$ is $G_X(\eta) = \exp(t\psi(\eta))$, where $\psi(\eta) = i\left(\mu - \frac{\sigma^2}{2}\right)\eta - \frac{\sigma^2\eta^2}{2} + \lambda\left(\frac{q_1}{1-i\eta\xi_1} + \frac{q_2}{1+i\eta\xi_2} - 1\right)$, with $q_1 \in (0,1)$ and $q_1 + q_2 = 1$, $\xi_1 > 1$ and $\xi_2 > 0$.

The parameters for this experiment are detailed in Table 6.18, with the parameters for the linear transformation set to $(a, c) = (20, 0.0)$. FourNet's training results, shown in Table 6.19, reveal an L_2-estimation error around 10^{-8}, demonstrating significant accuracy in this challenging scenario.

In comparison, the COS method, using 800 and 1200 terms (COS-800 and COS-1200) as implemented per [13], exhibited significant oscillations and losses of non-negativity in estimated transition densities, especially near $x = 0$, with these issues being particularly pronounced in the right tail (Figure 6.3(a)). Similar issues were noted in the left tail but are not shown. Figure 6.3(b) further illustrates how these oscillations and non-negativity issues in the COS method can lead to highly fluctuating and sometimes negative European call option prices, violating the no-arbitrage principle. In contrast, FourNet displayed minimal non-negativity loss, demonstrating its robustness and precision for financial applications. Notably, compared to European option prices calculated using an analytical formula from [31], FourNet achieved maximum relative errors in option prices of about 10^{-3}, showcasing superior accuracy.



(a) Estimated transition density

(b) Option prices

Fig. 6.3: Comparison between FourNet and COS-800/COS-1200, corresponding to parameters/data from Table 6.18 and Table 6.19.

7 Conclusion and future work This paper has introduced and rigorously analyzed FourNet, a novel single-layer FFNN developed to approximate transition densities with known closed-form Fourier transforms. Leveraging the unique Gaussian activation function, FourNet not only facilitates exact Fourier and inverse Fourier operations, which is crucial for training, but also draws parallels with the Gaussian mixture model, demonstrating its power in approximating sufficiently well a vast array of transition density functions. The hybrid loss function, integrating MSE with MAE regularization, coupled with a strategic sampling approach, has significantly enhanced the training process.

Through a comprehensive mathematical analysis, we demonstrate FourNet's capability to approximate transition densities in the L_2-sense arbitrarily well. We derive practical bounds for the L_2 estimation error and the potential (pointwise) loss of nonnegativity in the estimated densities for the general case of d-dimensions ($d \geq 1$), underscoring the robustness and applicability of our methodology in practical settings. We illustrate FourNet's accuracy and versatility through a broad range of models in quantitative finance, including (multi-dimensional) exponential Lévy processes and the Heston stochastic volatility models-even those augmented with the self-exciting Queue-Hawkes jump process. European and Bermudan option prices computed using estimated transition densities obtained through FourNet exhibit impressive accuracy.

In future work, we aim to extend FourNet to tackle more complex stochastic control problems, potentially involving higher dimensionality and model nonhomogeneity. This expansion is expected to broaden FourNet's applicability and enhance its utility in sophisticated financial modeling. We plan to explore various approaches to improve its performance in high-dimensional settings, assessing a range of enhancements to optimize its architecture and training processes. In addition, FourNet's simplicity and ease of implementation position it well for realistic models previously deemed challenging within existing frameworks. One particular area of interest includes investigating the impact of self-exciting jumps on optimal investment decisions in Defined Contribution superannuation–a topic of heightened relevance in a climate marked by rising inflation and economic volatility.

REFERENCES

[1] J.-P. Anker and B. Orsted, *Lie Theory: Harmonic Analysis on Symmetric Spaces–General Plancherel Theorems*, vol. 230, Springer Science & Business Media, 2006.

[2] L. A. S. Arias, P. Cirillo, and C. W. Oosterlee, *A new self-exciting jump-diffusion process for option pricing*, arXiv preprint arXiv:2205.13321, (2022).

[3] S. Bergner, T. Moller, D. Weiskopf, and D. J. Muraki, *A spectral analysis of function composition and its implications for sampling in direct volume visualization*, IEEE transactions on visualization and computer graphics, 12 (2006), pp. 1353–1360.

[4] A. Borovykh, S. Bohte, and C. W. Oosterlee, *Conditional time series forecasting with convolutional neural networks*, arXiv preprint arXiv:1703.04691, (2017).

[5] C. Calcaterra and A. Boldt, *Approximating with gaussians*, arXiv preprint arXiv:0805.3795, (2008).

[6] P. Carr, H. Geman, D. B. Madan, and M. Yor, *The fine structure of asset returns: An empirical investigation*, The Journal of Business, 75 (2002), pp. 305–332.

[7] P. Carr and D. Madan, *Option valuation using the fast Fourier transform*, Journal of Computational Finance, 2 (1999), pp. 61–73.

[8] C. Christara and D. Dang, *Adaptive and high-order methods for valuing American options*, Journal of Computational Finance, 14(4) (2011), pp. 73–113.

[9] S. N. Cohen, C. Reisinger, and S. Wang, *Arbitrage-free neural-sde market models*, arXiv preprint arXiv:2105.11053, (2021).

[10] D.-M. Dang, C. Christara, K. R. Jackson, and A. Lakhany, *An efficient numerical PDE approach for pricing foreign exchange interest rate hybrid derivatives*, Journal of Compu-

tational Finance, 18.
[11] D.-M. Dang and L. Ortiz-Gracia, *A dimension reduction Shannon-wavelet based method for option pricing*, Journal of Scientific Computing, 75 (2018), pp. 733–761.
[12] A. Daw and J. Pender, *An ephemerally self-exciting point process*, Advances in Applied Probability, 54 (2022), pp. 340–403.
[13] F. Fang and C. Oosterlee, *A novel pricing method for European options based on Fourier-Cosine series expansions*, SIAM Journal on Scientific Computing, 31 (2008), pp. 826–848.
[14] F. Fang and C. W. Oosterlee, *A Fourier-based valuation method for Bermudan and barrier options under Heston's model*, SIAM Journal on Financial Mathematics, 2 (2011), pp. 439–463.
[15] P. A. Forsyth and G. Labahn, *An ϵ-monotone Fourier methods for optimal stochastic control in finance*, Journal of Computational Finance, 22(4) (2019), pp. 25–71.
[16] I. Frodé, V. Sambergs, and S. Zhu, *Neural networks for credit risk and xVA in a front office pricing environment*, Available at SSRN 4136123, (2022).
[17] M. G. Garroni and J. L. Menaldi, *Green functions for second order parabolic integro-differential problems*, no. 275 in Pitman Research Notes in Mathematics, Longman Scientific and Technical, Harlow, Essex, UK, 1992.
[18] M. B. Giles, T. Nagapetyan, and K. Ritter, *Multilevel Monte Carlo approximation of distribution functions and densities*, SIAM/ASA journal on Uncertainty Quantification, 3 (2015), pp. 267–295.
[19] A. Gnoatto, A. Picarelli, and C. Reisinger, *Deep xva solver: A neural network–based counterparty credit risk management framework*, SIAM Journal on Financial Mathematics, 14 (2023), pp. 314–352.
[20] L. Goudenege, A. Molent, and A. Zanette, *Computing xVA for American basket derivatives by machine learning techniques*, arXiv preprint arXiv:2209.06485, (2022).
[21] Y. Gu, J. Harlim, S. Liang, and H. Yang, *Stationary density estimation of itô diffusions using deep learning*, SIAM Journal on Numerical Analysis, 61 (2023), pp. 45–82.
[22] J. Han, A. Jentzen, and W. E, *Solving high-dimensional partial differential equations using deep learning*, Proceedings of the National Academy of Sciences, 115 (2018), pp. 8505–8510.
[23] S. Heston, *A closed form solution for options with stochastic volatility with applications to bond and currency options*, Review of Financial Studies, 6 (1993), pp. 327–343.
[24] P. Hinds and M. Tretyakov, *Neural variance reduction for stochastic differential equations*, arXiv preprint arXiv:2209.12885, (2022).
[25] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, Neural networks, 2 (1989), pp. 359–366.
[26] C. Huré, H. Pham, and X. Warin, *Deep backward schemes for high-dimensional nonlinear PDEs*, Mathematics of Computation, 89 (2020), pp. 1547–1579.
[27] K. Ito, C. Reisinger, and Y. Zhang, *A neural network-based policy iteration algorithm with global h^2-superlinear convergence for stochastic games on domains*, Foundations of Computational Mathematics, 21 (2021), pp. 331–374.
[28] S. Ken-Iti, *Lévy processes and infinitely divisible distributions*, Cambridge University Press, 1999.
[29] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).
[30] T. Kobayashi, J. Adams, B. Lian, and S. Sahi, *Representation theory and mathematical physics*, Contemporary Mathematics, 557 (2011), pp. 23–40.
[31] S. G. Kou, *A jump-diffusion model for option pricing*, Management science, 48 (2002), pp. 1086–1101.
[32] J. Li and A. Barron, *Mixture density estimation*, Advances in neural information processing systems, 12 (1999).
[33] Y. Li and P. A. Forsyth, *A data-driven neural network approach to optimal asset allocation for target based defined contribution pension plans*, Insurance: Mathematics and Economics, 86 (2019), pp. 189–204.
[34] S. Liu, A. Borovykh, L. A. Grzelak, and C. W. Oosterlee, *A neural network-based framework for financial model calibration*, Journal of Mathematics in Industry, 9 (2019), pp. 1–28.
[35] R. Lord, F. Fang, F. Bervoets, and C. Oosterlee, *A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1678–1705.
[36] Y. Lu and D. Dang, *A pointwise convergent numerical integration method for Guaranteed Lifelong Withdrawal Benefits under stochastic volatility.* https://people.smp.uq.edu.au/DuyMinhDang/papers/epsilon_GLWB.pdf, 1 2023. Submitted.
[37] Y. Lu, D. Dang, P. Forsyth, and G. Labahn, *An ϵ-monotone Fourier method for Guar-*

anteed Minimum Withdrawal Benefit (GMWB) as a continuous impulse control problem. https://people.smp.uq.edu.au/Duy-MinhDang/papers/epsilon_GMWB.pdf, 06 2022. Submitted.

[38] Y. Lu and D.-M. Dang, *A semi-Lagrangian ε ε-monotone Fourier method for continuous withdrawal GMWBs under jump-diffusion with stochastic interest rate*, Numerical Methods for Partial Differential Equations, 40 (2024), p. e23075.

[39] D. B. Madan and E. Seneta, *The variance gamma (VG) model for share market returns*, Journal of business, (1990), pp. 511–524.

[40] S. Maxwell and W. Halbert, *Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions*, in International Joint Conference on Neural Networks, vol. 1, 1989, pp. 613–617, https://doi.org/10.1109/IJCNN.1989.118640.

[41] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, *Finite mixture models*, Annual review of statistics and its application, 6 (2019), pp. 355–378.

[42] R. C. Merton, *Option pricing when underlying stock returns are discontinuous*, Journal of financial economics, 3 (1976), pp. 125–144.

[43] G. N. Milstein, J. G. Schoenmakers, and V. Spokoiny, *Transition density estimation for stochastic differential equations via forward-reverse representations*, Bernoulli, 10 (2004), pp. 281–312.

[44] A. V. Oppenheim and G. C. Verghese, *Signals, systems & inference*, Pearson London, 2017.

[45] L. Ortiz-Gracia and C. W. Oosterlee, *A highly efficient Shannon wavelet inverse Fourier technique for pricing European options*, SIAM Journal on Scientific Computing, 38 (2016), pp. B118–B143.

[46] C. Reisinger and Y. Zhang, *Rectified deep neural networks overcome the curse of dimensionality for nonsmooth value functions in zero-sum games of nonlinear stiff systems*, Analysis and Applications, 18 (2020), pp. 951–999.

[47] S. d. B. Rollin, A. Ferreiro-Castilla, and F. Utzet, *A new look at the Heston characteristic function*, arXiv preprint arXiv:0902.2154, (2009).

[48] M. Rosenblatt, *Remarks on some nonparametric estimates of a density function*, The annals of mathematical statistics, (1956), pp. 832–837.

[49] M. J. Ruijter and C. W. Oosterlee, *Two-dimensional fourier cosine series expansion method for pricing financial options*, SIAM Journal on Scientific Computing, 34 (2012), pp. B642–B671.

[50] J. Sirignano and K. Spiliopoulos, *DGM: A deep learning algorithm for solving partial differential equations*, Journal of computational physics, 375 (2018), pp. 1339–1364.

[51] H. Su and D. P. Newton, *Widening the range of underlyings for derivatives pricing with quad by using finite difference to calculate transition densities-demonstrated for the no-arbitrage SABR model*, The Journal of Derivatives, 28 (2020), pp. 22–46.

[52] H. Su, M. V. Tretyakov, and D. P. Newton, *Option valuation through deep learning of transition probability density*, arXiv preprint arXiv:2105.10467, (2021).

[53] D. Tavella and C. Randall, *Pricing financial instruments: The finite difference method*, vol. 13, John Wiley & Sons, 2000.

[54] P. T. Tran et al., *On the convergence proof of AMSGrad and a new version*, IEEE Access, 7 (2019), pp. 61706–61716.

[55] P. M. van Staden, P. A. Forsyth, and Y. Li, *A parsimonious neural network approach to solve portfolio optimization problems without using dynamic programming*, arXiv preprint arXiv:2303.08968, (2023).

[56] E. Weinan, J. Han, and A. Jentzen, *Algorithms for solving high dimensional PDEs: from nonlinear Monte Carlo to machine learning*, Nonlinearity, 35 (2021), p. 278.

[57] H. Yan and H. Ouyang, *Financial time series prediction based on deep learning*, Wireless Personal Communications, 102 (2018), pp. 683–700.

[58] K. Yosida, *Functional analysis, xii+ 465*, 1968.

[59] H. Zhang and D.-M. Dang, *A monotone numerical integration method for mean–variance portfolio optimization under jump-diffusion models*, Mathematics and Computers in Simulation, 219 (2024), pp. 112–140.

Appendices Appendix A. Constructing non-uniform partitions with multiple peaks.

Algorithm A.1 provides a detailed procedure for constructing non-uniform, yet fixed, partitions of the interval $[\eta_l, \eta_u]$, comprised of M sub-intervals. These partitions feature denser points around a chosen point, $\eta_c \in [\eta_l, \eta_u]$. The parameters d_l and d_u determine

the point densities in $[\eta_l, \eta_c]$ and $[\eta_c, \eta_u]$, respectively, represented as $\frac{1}{d_l}$ and $\frac{1}{d_u}$.

Algorithm A.1 Algorithm for constructing a non-uniform partition of an interval $[\eta_l, \eta_u]$ into M sub-intervals, having a single concentration point, η_c, which is the m-partition point, $m \in \{0, \ldots, M\}$, is fixed.

PartitionOne$(\eta_l, \eta_u, \eta_c, M, m, d_l, d_u)$

1: compute $\alpha_l = \sinh^{-1}\left(\frac{\eta_l - \eta_c}{d_l}\right)$ and $\alpha_u = \sinh^{-1}\left(\frac{\eta_u - \eta_c}{d_u}\right)$;

2: compute $\eta_0 = \eta_l; \eta_j = \eta_c + d_l \sinh(\alpha_l(1 - k_j))$, where $k_j = \dfrac{j}{m}$, $j = 1, \ldots, m$;

3: compute $\eta_j = \eta_c + d_u \sinh(\alpha_u k_j)$, where $k_j = \dfrac{j}{M - m}$, $j = 1, \ldots, (M - m)$;

4: return $Q \equiv \{\eta_j\}_{j=0}^{m} \cup \{\eta_j\}_{j=1}^{M-m}$;

Algorithm A.2 Algorithm for constructing a non-uniform partition of an interval with multiple concentration points.

PartitionMulti$(\eta_{\min}, \eta_{\max}, \{\eta_j\}_{j=1}^{J}, \{P_j\}_{j=1}^{v}, \{q_j\}_{j=1}^{J}, \{\eta_l^j\}_{j=1}^{J}, \{\eta_u^j\}_{j=1}^{J})$

1: $Q_1 \leftarrow PartitionOne\Big(\eta_{\min}, \dfrac{\eta_1 + \eta_2}{2}, \eta_1, P_1, q_1, \eta_l^1, \eta_u^1\Big)$;

2: $Q_j \leftarrow PartitionOne(\dfrac{\eta_{j-1} + \eta_j}{2}, \dfrac{\eta_j + \eta_{j+1}}{2}, \eta_j, P_j, q_j, \eta_l^j, \eta_u^j)$, $j = 2, \ldots, J - 1$;

3: $Q_J \leftarrow PartitionOne(\dfrac{\eta_{J-1} + \eta_J}{2}, \eta_{\max}, \eta_J, P_J, q_J, \eta_l^J, \eta_u^J)$;

4: return $Q \equiv \cup_{j=1}^{J} Q_j$;

We use Algorithm A.1 in Algorithm A.2 to generate a non-uniform partition having P sub-intervals for the region $[\eta_{\min}, \eta_{\max}] \equiv [-\eta', \eta']$ with concentration points η_j, $j = 1, \ldots, J$, satisfying $\eta_{\min} \leq \eta_1 < \eta_2 < \ldots < \eta_J \leq \eta_{\max}$. Here, P_j is the number of sub-intervals for the j-th sub-region containing η_j, $j = 1, \ldots, J$, with $\sum_{j=1}^{J} P_j = P$; q_j is the local index of the gridpoint in the j-th sub-region that is equal to η_j; η_l^j and η_u^j are the upper and lower density parameters, respectively, associated with the j-th sub-region containing η_j.