IMPERIAL

Imperial College London

Department of Mathematics

Deep Learning Accelerator for CVA Calculations with Wrong-Way Risk

Author: Jingtong Xu (CID: 06022693)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2024-2025

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Acknowledgements

I am deeply grateful to my supervisor, Prof. Damiano Brigo, for his continued support and thoughtful guidance throughout this project. His clarity of insight, high standards, and constructive feedback were instrumental in shaping both the direction and quality of this dissertation.I would also like to thank all the academic and administrative staff of the Mathematics and Finance programme at Imperial College London for their outstanding teaching and ongoing support. The knowledge and perspective I have gained over the course of the year have not only deepened my understanding of quantitative finance but also proved essential to the successful completion of this thesis.I am thankful to my classmates and friends for their companionship, encouragement, and the many moments of shared effort and growth. Finally, I extend my heartfelt thanks to my family for their unwavering support and constant belief in me throughout this journey.

Abstract

Credit Valuation Adjustment (CVA) quantifies the present value of expected losses arising from counterparty default and is particularly sensitive to Wrong-Way Risk (WWR) during periods of market stress. In modern financial environments, accurate and timely CVA computation is critical for effective CVA pricing, trading, hedging, risk management and regulatory compliance. Although Monte Carlo (MC) simulation remains a conventional and widely used method, it becomes computationally burdensome when applied to large parameter grids or frequent scenario analyses. To address this challenge, this dissertation proposes a neural network surrogate in a tractable setting with geometric Brownian exposure, CIR default intensity, and correlated Brownian shocks under the risk-neutral measure. A labeled Monte Carlo dataset is constructed, and fully connected surrogates with one to four hidden layers are trained following a protocol that begins with depth-specific initial configurations, which are subsequently optimized through extensive hyperparameter search and targeted fine-tuning. The results show that a single-hidden-layer model (FCNN-1) offers the best trade-off between accuracy, efficiency, and robustness: it achieves test errors below 1% (around 0.88% of average CVA) and yields empirical speedups of order 10^4 compared to MC. Under distributional shift, FCNN-1 extrapolates smoothly and remains consistent with financial intuition, while deeper models exhibit occasional instability. Interpretability is addressed via SHAP (DeepSHAP), identifying exposure volatility, initial intensity, and correlation as key drivers, with the one-layer model replicating the attribution structure of its deeper counterpart. Overall, a carefully regularized shallow network can retain analytical structure, generalize reliably, and deliver production-grade acceleration without compromising transparency.

Contents

1 Data Generation **9**

1.1 Model Setup . . . 9

1.1.1 Mathematical Framework . . . 9

1.1.2 CVA Definition . . . 9

1.1.3 Intensity and Default Time Modeling (CIR Model) . . . 10

1.1.4 Monte Carlo Approaches . . . 10

1.1.5 Numerical Schemes . . . 11

1.1.6 Analytic CVA for $\rho = 0$ . . . 12

1.2 Simulation Parameters . . . 12

1.2.1 Comparison of τ-based vs. Immersion CVA Estimators . . . 13

1.2.2 Impact of Weekly vs. Daily discretization . . . 14

1.2.3 Choice of $N_{MC} = 10^5$ as a Practical Working Point . . . 15

1.2.4 Final Parameter Grid for Simulations . . . 15

1.3 Dataset Output and Storage . . . 16

1.4 Summary and Next Steps . . . 17

2 Neural Network Construction **18**

2.1 Initial Setup . . . 18

2.1.1 Overview of the Architectures . . . 18

2.1.2 Data Split and Preprocessing . . . 18

2.1.3 Training Protocol . . . 19

2.1.4 Initial Hyperparameter Selection (1–3 Layers) . . . 19

2.1.5 Four-Layer Initial Search . . . 20

2.1.6 Systematic Evaluation of Initial Sets . . . 21

2.2 Extensive Hyperparameter Search . . . 22

2.2.1 Methodology . . . 22

2.2.2 Search Outcome and Selected Configuration . . . 22

2.3 Hyperparameter Fine-Tuning . . . 23

2.3.1 Methodology . . . 23

2.3.2 Model Size and Trainability . . . 23

2.3.3 Results . . . 24

2.4 Depth-wise Search and Fine-Tuning . . . 24

2.4.1 1 Layer (FCNN-1) . . . 25

2.4.2 2 Layers (FCNN-2) . . . 25

2.4.3 3 Layers (FCNN-3) . . . 25

2.4.4 4 Layers (FCNN-4) . . . 26

2.4.5 Cross-Depth Comparison . . . 26

3 One vs. Three Layers: Performance, Efficiency, and Robustness **29**
3.1 Evaluation Protocol and Setup . . . 29
3.2 Test Accuracy and Inference Throughput . . . 29
3.3 OOD Stress Tests under Structural Constraints . . . 30
3.4 Local Sensitivity Analysis . . . 31
3.4.1 Protocol . . . 31
3.4.2 Default–intensity Level λ_0 . . . 31
3.4.3 Mean–reversion Speed k . . . 31
3.4.4 Long–run Intensity θ . . . 32
3.4.5 Intensity Volatility v . . . 33
3.4.6 Asset Volatility σ . . . 34
3.4.7 Correlation ρ (WWR) . . . 34
3.5 ρ=0 Sanity Check: Closed Form vs. Monte Carlo vs. Surrogates . . . 34
3.6 Summary and Implications . . . 36

4 Model Interpretability via SHAP **37**
4.1 Motivation and Setup . . . 37
4.2 Global Importance . . . 38
4.3 Local Effects and Directionality . . . 39
4.4 Economic Interpretation . . . 40
4.4.1 Comparative Interpretation: 1-layer vs. 3-layer . . . 40
4.4.2 Dominant Short-term Drivers . . . 40
4.5 Summary and Implications . . . 43

A Technical Proofs **47**
A.1 Equivalence of the Compact and Intensity Representations of CVA . . . 47

B Dataset Files **49**
B.1 The CSV File `cva_grid.csv` . . . 49

Bibliography **53**

List of Figures

3.1 Sensitivity of CVA to λ_0 (FCNN-1 vs. FCNN-3). 32
3.2 Sensitivity of CVA to k (FCNN-1 vs. FCNN-3). 32
3.3 Sensitivity of CVA to θ (FCNN-1 vs. FCNN-3). 33
3.4 Sensitivity of CVA to v (FCNN-1 vs. FCNN-3). 33
3.5 Sensitivity of CVA to σ (FCNN-1 vs. FCNN-3). 34
3.6 WWR sensitivity: CVA vs. ρ (FCNN-1 & FCNN-3) with MC baseline. . . . 35

4.1 Global SHAP importance (mean absolute SHAP) for the two surrogates. Left: 1-layer network; Right: 3-layer network. 39
4.2 SHAP beeswarm plots (local effects, coloured by feature values). Top: 1-layer surrogate; bottom: 3-layer surrogate. The horizontal axis shows the SHAP value (impact on the model output relative to the baseline), while colour encodes the actual feature value. 41

List of Tables

1.1 Baseline parameter configuration for CIR intensity and GBM exposure dynamics. . . . 13
1.2 Comparison of τ-based and immersion-based CVA estimators at $T = 1$ year and $\rho = 0$, reporting Monte Carlo estimates with standard errors (SE), relative error (Rel. Err), and runtime. . . . 13
1.3 Impact of correlation ρ on CVA estimates WWR. Simulations use $T = 1$ year, $N_{MC} = 10{,}000$, and 52 steps per year. Relative deviations Δ are expressed with respect to the analytic benchmark CVA at $\rho = 0$, equal to 0.3116. . . . 14
1.4 Impact of discretization frequency and working point on CVA estimation ($T = 1$ year, $\rho = 0$, immersion-based estimator) . . . 14

2.1 Data split used in model development. . . . 18
2.2 Common training protocol across architectures. . . . 19
2.3 Initial seeds for 1–3 hidden layer networks. . . . 19
2.4 Promising seeds identified for 4-layer networks via Hyperband. . . . 20
2.5 Parameter growth from 1 to 4 hidden layers (input dimension = 6, output = 1). . . . 21
2.6 Test performance of the initial sets under a unified training protocol. . . . 21
2.7 Random-search space for the 3-layer architecture (500 trials). . . . 22
2.8 Best 3-layer configuration returned by RandomSearch (validation objective). 22
2.9 Fine-tuning setup (architecture fixed, optimization refined). . . . 23
2.10 Parameter accounting for the fixed $(832, 160, 16)$ model with BN. . . . 24
2.11 Held-out test performance before and after fine-tuning (architecture fixed). 24
2.12 Best 1-layer search configuration. . . . 25
2.13 Best 2-layer search configuration. . . . 25
2.14 Best 3-layer search configuration. . . . 25
2.15 Best 4-layer configuration (Seed Family I). . . . 26
2.16 Best 4-layer configuration (Seed Family II). . . . 26
2.17 Search vs. fine-tuned test performance across depths (percentages w.r.t. average CVA). The 4-layer row reports the better of the two seed families at each stage. . . . 26
2.18 FCNN-3: interaction of capacity and fine-tuning regime (percentages w.r.t. average CVA). . . . 28

3.1 Speed/accuracy trade-off on the held-out test set. . . . 29
3.2 Mean absolute deviations on the ρ=0 grid. Both surrogates are closer on average to the closed-form benchmark than to the MC labels. . . . 35

4.1 Mean absolute SHAP values (global importance). Percent shares equal each entry divided by the column-wise sum (0.3507 for 1-layer; 0.3304 for 3-layer). 39

B.1 Schema of `cva_grid.csv` (field names, meanings, and types). . . . 49

B.2 Preview of `cva_grid.csv` (first and last rows; illustrative). 50

Introduction

Counterparty credit risk has become a central concern in modern finance, especially since the 2008 financial crisis. This has led to the widespread adoption of Credit Value Adjustment (CVA) as a key metric incorporating counterparty default risk into derivative pricing [21, 5]. CVA is the present value of expected losses due to a counterparty's potential default over the life of a portfolio [12]. In practice, institutions compute CVA and related quantities on a regular basis for regulatory capital and internal risk management, which underscores the need for both accuracy and computational efficiency [24]. A critical driver is Wrong-Way Risk (WWR), the adverse dependence between exposure and counterparty credit quality: exposures tend to be elevated precisely in scenarios where the counterparty's credit state deteriorates. Such dependence can materially increase CVA and, when neglected, as was common before the crisis, lead to understated risk. Accounting for WWR is therefore essential; yet doing so introduces both modeling and computational challenges [16]. In particular, Brigo and Pallavicini [14, 15] provide evidence that the dependence between market factors and credit risk—such as the correlation between interest rates and default intensities—has a significant influence on CVA valuation.

Accurately evaluating CVA under WWR typically requires complex models and intensive simulation [13]. Unlike simple derivatives that admit closed forms in special cases, CVA with adverse exposure–credit dependence rarely has analytical solutions and must be computed numerically. Monte Carlo (MC) remains the workhorse [23], especially when modeling the correlation between exposure and default [14]. Despite its conceptual simplicity, MC is computationally demanding: stabilizing estimators of expectations over exposure-at-default outcomes often necessitates 10^5–10^6 simulated paths, even with variance reduction and careful time discretization [27]. The resulting runtime can hinder intraday monitoring, portfolio-scale aggregation, and broad scenario analysis. These limitations motivate faster surrogate methods that retain fidelity to WWR effects.

In recent years, machine learning—particularly deep learning—has emerged as a powerful tool for accelerating complex computations in quantitative finance. Neural networks can act as surrogate models [25, 20]: once trained on Monte Carlo–generated data, they may learn to approximate the CVA as a function of underlying parameters, delivering near-instant predictions for new inputs without re-running costly simulations. This enables substantial speed-ups, making tasks such as real-time monitoring, rapid scenario analysis, and regulatory reporting computationally feasible. Prior research has further demonstrated the efficacy of neural surrogates in replicating sophisticated pricing models—such as those involving stochastic or rough volatility—while maintaining high accuracy and structural coherence [29]. Nevertheless, the practical adoption of such models hinges on interpretability: in risk management contexts, where transparency and auditability are paramount, black-box models are typically viewed with caution. As such, the development of interpretable surrogates is essential for building trust, supporting model validation, and satisfying regulatory requirements [9, 32]. To this end, we perform an interpretability analysis using DeepSHAP, which approximates Shapley values by combining the game-theoretic attribution framework with DeepLIFT-style backpropagation rules [34, 30, 18]. Global feature importance is assessed via mean absolute SHAP values, while

local effects are visualized through beeswarm plots that convey the distribution, sign, and magnitude of individual contributions. This methodology, recently applied in financial modeling, provides a principled foundation for verifying economic coherence, diagnosing potential inconsistencies, and understanding the structural dependencies learned by the model [36, 9].

This dissertation investigates a neural-network-based surrogate approach for efficient CVA estimation under Wrong-Way Risk. The proposed architecture is a feedforward network specifically designed to reflect key structural and qualitative features of the pricing problem—such as directional sensitivity to exposure volatility and credit intensity parameters—consistent with established financial intuition and theoretical benchmarks. Trained on Monte Carlo–generated data, the network learns a nonparametric approximation of the CVA mapping across a broad input domain. The objective is to combine rapid inference with high numerical accuracy and transparent interpretability, thereby supporting practical applications such as real-time risk monitoring, scenario analysis, and regulatory reporting, without compromising model fidelity.

The dissertation is organized as follows. Chapter 1 (*Data Generation*) introduces the modeling framework for CVA under Wrong-Way Risk, combining geometric Brownian motion (GBM) for the exposure with a CIR process for default intensity under the risk-neutral measure $\mathbb{Q}$. It reviews the Monte Carlo estimation techniques and outlines the associated numerical schemes and discretization choices. An analytic benchmark for the special case $\rho = 0$ is also provided. It then finalizes the simulation setup by selecting a practical working sample size $N_{\text{MC}} = 10^5$ and constructing a suitable parameter grid. The resulting dataset includes CVA estimates with standard errors, and its generation time serves as the baseline for evaluating acceleration in subsequent chapters. Chapter 2 (*Neural Network Construction*) sets out the development of the surrogate modeling framework. It introduces feedforward architectures with one to four hidden layers, describes the data split and preprocessing steps, and details the training procedure, including standardization, early stopping with weight restoration, Huber loss refinement, and cosine learning rate (LR) scheduling. The chapter then presents the initial hyperparameter selection, followed by an extensive search and targeted fine-tuning across all depths. A depth-wise comparison ultimately identifies FCNN-1 and FCNN-3 as the most promising candidates for evaluation in subsequent chapters. Chapter 3 (*One vs. Three Layers*) provides the main empirical assessment. It compares the one- and three-layer surrogates in terms of test-set accuracy and inference speed, examines out-of-distribution robustness through stress tests, and analyses local sensitivities across the parameter space $(\lambda_0, k, \theta, v, \sigma, \rho)$. A dedicated $\rho = 0$ sanity check benchmarks both models against the analytical solution and Monte Carlo estimates, providing a comparative view of predictive performance, computational efficiency, and generalization robustness. Chapter 4 (*Model Interpretability via SHAP*) focuses on interpretability. Using DeepSHAP, we compute global importance scores and visualize local directional effects, providing an economic interpretation of the learned CVA response surfaces. It further compares the attribution patterns of the one- and three-layer surrogates, showing that a simpler architecture can retain interpretability while improving computational efficiency.

Chapter 1

Data Generation

1.1 Model Setup

1.1.1 Mathematical Framework

In modeling Credit Valuation Adjustment, we use a stochastic framework combining asset price dynamics modeled by Geometric Brownian Motion (GBM) and default intensity dynamics modeled by a Cox–Ingersoll–Ross (CIR) process [19] under the risk–neutral measure $\mathbb{Q}$. This setup allows us to capture essential financial risks, including Wrong-Way Risk, where default risk correlates with underlying asset values [15].

Specifically, the dynamics for the underlying asset price S_t and default intensity λ_t are given by the stochastic differential equations (SDEs):

$$dS_t = rS_t\,dt + \sigma S_t\,dW_t^{(2)}, \tag{1.1.1}$$

$$d\lambda_t = k(\theta - \lambda_t)\,dt + v\sqrt{\lambda_t}\,dW_t^{(1)}, \tag{1.1.2}$$

with correlation between the Brownian motions given by

$$dW_t^{(1)} dW_t^{(2)} = \rho\,dt, \tag{1.1.3}$$

where $W_t^{(1)}$ and $W_t^{(2)}$ are correlated standard Brownian motions.

1.1.2 CVA Definition

CVA represents the expected discounted loss arising from the default of a counterparty. In this work, we consider the CVA of a single European call option with maturity T and strike K. Under the risk-neutral measure $\mathbb{Q}$, the general CVA formula takes the form:

$$\text{CVA} = \text{LGD} \cdot \mathbb{E}^{\mathbb{Q}}\left[D(0,\tau)\cdot \mathbf{1}_{\{\tau<T\}}\cdot\left(\mathbb{E}_{\tau}^{\mathbb{Q}}\left[D(\tau,T)\cdot(S_T-K)^{+}\right]\right)^{+}\right], \tag{1.1.4}$$

where LGD is the loss given default, $D(t,T) = e^{-r(T-t)}$ is the discount factor, and τ is the counterparty's default time [10, 12].

Scope and generality. Focusing on a European call provides a canonical benchmark for analysis, but the modeling setup and methodology extend naturally to broader classes of instruments and portfolios, especially simple contingent claims. In particular, for equity-driven exposures, one may replace the call payoff with a general simple claim payoff $E_T = f(S_T)$, such as those arising in long convex portfolios or netting sets, while retaining the same reduced-form intensity framework. The immersion formula may have to be generalized if f is not always positive, but this is feasible, and a full simulation approach

without immersion is also an option. With path-dependent or early-exercise features in the exposure, the approach would have to be adapted, although, overall, only the exposure profile changes, and the CVA computation proceeds analogously.

1.1.3 Intensity and Default Time Modeling (CIR Model)

Default intensity models assume a strictly positive stochastic default intensity (or hazard rate) λ_t. The CIR model [19] for intensity is described as:

$$d\lambda_t = k(\theta - \lambda_t)\, dt + v\sqrt{\lambda_t}\, dW_t, \tag{1.1.5}$$

with mean–reversion speed k, long–term mean θ, volatility parameter v, and initial value $\lambda_0 > 0$. Strict positivity holds under the Feller condition $2k\theta \geq v^2$ [22]; otherwise, numerical schemes such as the full truncation Euler method are applied to preserve non–negativity. The default time τ is defined by the inverse of the cumulative intensity [21]:

$$\tau = \inf\{t \geq 0 : \Lambda_t \geq \xi\}, \tag{1.1.6}$$

where $\Lambda_t = \int_0^t \lambda_s\, ds$ and ξ is an exponential random variable independent of λ_t.

1.1.4 Monte Carlo Approaches

Default Time Simulation (τ Simulation). Under the risk–neutral measure $\mathbb{Q}$, Monte Carlo simulation explicitly simulates the default time τ via an exponential random variable $\xi \sim \text{Exp}(1)$ [11]:

$$\tau = \inf\{t \geq 0 : \Lambda_t \geq \xi\}, \qquad \Lambda_t = \int_0^t \lambda_s\, ds. \tag{1.1.7}$$

The CVA is then approximated by

$$\text{CVA}_{\text{MC}}^{(\tau)} = \frac{\text{LGD}}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} D(0,\tau_i)\, \mathbf{1}_{\{\tau_i < T\}} \cdot \left(\mathbb{E}_{\tau_i}^{\mathbb{Q}}\left[D(\tau_i, T)(S_T - K)^+\right]\right)^+. \tag{1.1.8}$$

Immersion Formula (Without τ Simulation). Under the immersion hypothesis (H–hypothesis) [12], avoiding explicit simulation of τ, the CVA admits the compact representation

$$\text{CVA} = \text{LGD}\, \mathbb{E}^{\mathbb{Q}}\left[\left(1 - e^{-\int_0^T \lambda_s\, ds}\right) e^{-rT}(S_T - K)^+\right]. \tag{1.1.9}$$

Brief derivation.

Let $\mathbb{F} = (\mathcal{F}_t)_{t\geq 0}$ be the market filtration generated by (S, λ), and let $\mathbb{G} = (\mathcal{G}_t)_{t\geq 0}$ be its progressive enlargement with the default process $H_t = \mathbf{1}_{\{\tau \leq t\}}$, so that $\mathcal{G}_t = \mathcal{F}_t \vee \sigma(H_s : s \leq t)$. Assume λ is non–negative and progressively measurable with $\int_0^T \lambda_s\, ds < \infty$ a.s., and $\mathbb{E}^{\mathbb{Q}}[(S_T - K)^+] < \infty$. Define $D(s,t) = \exp\big(-\int_s^t r_u\, du\big)$. τ is a $\mathbb{G}$–stopping time with $\mathbb{Q}(\tau = T) = 0$ in the Cox setup.

We start from the general CVA definition:

$$\text{CVA} = \text{LGD}\, \mathbb{E}^{\mathbb{Q}}\Big[D(0,\tau)\, \mathbf{1}_{\{\tau<T\}}\, \big(\mathbb{E}^{\mathbb{Q}}[D(\tau,T)(S_T - K)^+ \mid \mathcal{G}_\tau]\big)^+\Big].$$

Since $D(\tau,T)(S_T - K)^+ \geq 0$, the inner conditional expectation is non–negative and the outer positive part can be dropped:

$$\text{CVA} = \text{LGD}\, \mathbb{E}^{\mathbb{Q}}\Big[D(0,\tau)\, \mathbf{1}_{\{\tau<T\}}\, \mathbb{E}^{\mathbb{Q}}[D(\tau,T)(S_T - K)^+ \mid \mathcal{G}_\tau]\Big].$$

Because $D(0,\tau)$ and $\mathbf{1}_{\{\tau<T\}}$ are $\mathcal{G}_\tau$–measurable, they can be brought inside the inner conditional expectation (linearity and measurability):

$$\text{CVA} = \text{LGD}\ \mathbb{E}^{\mathbb{Q}}\Big[\mathbb{E}^{\mathbb{Q}}\big(D(0,\tau)\,\mathbf{1}_{\{\tau<T\}}\,D(\tau,T)(S_T-K)^+ \mid \mathcal{G}_\tau\big)\Big].$$

Using the multiplicative property of discount factors $D(0,\tau)D(\tau,T) = D(0,T)$ (pathwise, even for stochastic r), and then the tower property $\mathbb{E}[\mathbb{E}(X \mid \mathcal{G}_\tau)] = \mathbb{E}[X]$, we obtain

$$\text{CVA} = \text{LGD}\ \mathbb{E}^{\mathbb{Q}}\big[\mathbf{1}_{\{\tau<T\}}\,D(0,T)(S_T-K)^+\big].$$

Now apply the tower property in the opposite direction, conditioning further on the default–free σ–algebra $\mathcal{F}_T$ (law of iterated expectations). Since $D(0,T)$ and $(S_T-K)^+$ are $\mathcal{F}_T$–measurable, they factor out:

$$\text{CVA} = \text{LGD}\ \mathbb{E}^{\mathbb{Q}}\Big[D(0,T)(S_T-K)^+\,\mathbb{E}^{\mathbb{Q}}(\mathbf{1}_{\{\tau<T\}} \mid \mathcal{F}_T)\Big].$$

At this point we compute the conditional default probability. Let $\Lambda_t := \int_0^t \lambda_s\,ds$. By the Cox construction, $\tau = \inf\{t \geq 0 : \Lambda_t \geq \xi\}$ with $\xi \sim \text{Exp}(1)$ independent of $\mathcal{F}_\infty$. Therefore,

$$\mathbb{E}^{\mathbb{Q}}(\mathbf{1}_{\{\tau<T\}} \mid \mathcal{F}_T) = \mathbb{Q}(\xi < \Lambda_T \mid \mathcal{F}_T) = 1 - e^{-\Lambda_T}.$$

Substituting this into the previous display yields the immersion representation

$$\text{CVA} = \text{LGD}\ \mathbb{E}^{\mathbb{Q}}\left[\left(1 - e^{-\int_0^T \lambda_s\,ds}\right) D(0,T)(S_T-K)^+\right].$$

If r is constant, then $D(0,T) = e^{-rT}$ and

$$\text{CVA} = \text{LGD}\ \mathbb{E}^{\mathbb{Q}}\left[\left(1 - e^{-\int_0^T \lambda_s\,ds}\right) e^{-rT}(S_T-K)^+\right].$$

Since S and λ may be correlated, the expectation cannot, in general, be factorized into separate exposure and default components.

1.1.5 Numerical Schemes

CIR Process (Full Truncation Euler Scheme): For the CIR intensity, we employ the Full Truncation Euler scheme, proposed by Lord, Koekkoek, and van Dijk [27], to ensure positivity and reduce numerical bias.

$$\begin{cases} \widetilde{\lambda}_{n+1} = \lambda_n + k(\theta - \max(\lambda_n,0))\Delta t + v\sqrt{\max(\lambda_n,0)}\sqrt{\Delta t}W_n^{(1)}, \\ \lambda_{n+1} = \max(\widetilde{\lambda}_{n+1},0), \end{cases} \qquad W_n^{(1)} \sim \mathcal{N}(0,1). \tag{1.1.10}$$

GBM Process (Exact Scheme): Asset dynamics under GBM can be exactly simulated as [23]:

$$S_{t+\Delta t} = S_t \exp\left[\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}W^{(2)}\right], \quad W^{(2)} \sim \mathcal{N}(0,1). \tag{1.1.11}$$

Correlation between Brownian Motions: To simulate correlated Brownian motions $W_t^{(1)}$ and $W_t^{(2)}$, we generate two independent standard normal random variables $Z^{(1)}$ and $Z^{(2)}$ and construct [23]:

$$W^{(1)} = Z^{(1)}, \tag{1.1.12}$$

$$W^{(2)} = \rho Z^{(1)} + \sqrt{1-\rho^2}\,Z^{(2)}, \quad Z^{(1)}, Z^{(2)} \sim \mathcal{N}(0,1) \text{ independently.} \tag{1.1.13}$$

1.1.6 Analytic CVA for $\rho = 0$

Starting from the immersion representation (1.1.9),

$$\mathrm{CVA} = \mathrm{LGD}\,\mathbb{E}^{\mathbb{Q}}\Big[\Big(1 - e^{-\int_0^T \lambda_s\,ds}\Big)e^{-rT}(S_T - K)^+\Big],$$

the uncorrelated case $\rho = 0$ implies that the Brownian motions driving the CIR intensity λ and the GBM asset S are independent. Consequently, $\Lambda_T := \int_0^T \lambda_s ds$ is independent of S_T, so the expectation factorizes (for constant short rate r):

$$\mathrm{CVA}_{\rho=0} = \mathrm{LGD}\ \underbrace{\mathbb{E}^{\mathbb{Q}}\big[1 - e^{-\Lambda_T}\big]}_{\mathbb{Q}(\tau<T)} \cdot \underbrace{e^{-rT}\,\mathbb{E}^{\mathbb{Q}}\big[(S_T - K)^+\big]}_{C_{\mathrm{BS}}(S_0,K,r,\sigma,T)}.$$

Brief derivation. In the Cox setup, conditional survival is $\exp(-\Lambda_T)$, $\mathbb{Q}(\tau > T) = \mathbb{E}^{\mathbb{Q}}\big[e^{-\Lambda_T}\big]$. For the CIR process $d\lambda_t = k(\theta - \lambda_t)\,dt + v\sqrt{\lambda_t}\,dW_t$, this Laplace transform is affine [19].:

$$\mathbb{E}^{\mathbb{Q}}\big[e^{-\Lambda_T}\big] = A(T)\,e^{-B(T)\lambda_0},$$

with

$$\begin{aligned}
\gamma &= \sqrt{k^2 + 2v^2},\\
B(T) &= \frac{2\,(e^{\gamma T} - 1)}{(k+\gamma)(e^{\gamma T} - 1) + 2\gamma},\\
A(T) &= \left(\frac{2\gamma\, e^{(k+\gamma)T/2}}{(k+\gamma)(e^{\gamma T} - 1) + 2\gamma}\right)^{\frac{2k\theta}{v^2}}.
\end{aligned}$$

Thus $\mathbb{Q}(\tau < T) = 1 - A(T)e^{-B(T)\lambda_0}$. Under risk–neutral GBM, the exposure is the Black–Scholes call price [3]:

$$C_{\mathrm{BS}}(S_0, K, r, \sigma, T) = S_0\,\Phi(d_1) - K\,e^{-rT}\,\Phi(d_2), \tag{1.1.14}$$

$$d_1 = \frac{\ln(S_0/K) + \big(r + \frac{1}{2}\sigma^2\big)T}{\sigma\sqrt{T}}, \tag{1.1.15}$$

$$d_2 = d_1 - \sigma\sqrt{T}. \tag{1.1.16}$$

Substituting these components gives the closed–form benchmark:

$$\mathrm{CVA}_{\rho=0} = \mathrm{LGD}\,\big(1 - A(T)e^{-B(T)\lambda_0}\big)\,C_{\mathrm{BS}}(S_0, K, r, \sigma, T). \tag{1.1.17}$$

This analytic expression serves as a benchmark for validating our Monte Carlo CVA estimates when $\rho = 0$, and is particularly useful in evaluating numerical errors introduced by simulation.

1.2 Simulation Parameters

This section details the parameter configuration and methodological choices adopted to ensure robustness, accuracy, and computational efficiency in the CVA simulations. Unless otherwise specified, all numerical experiments reported in Sections 1.2.1, 1.2.2, and 1.2.3 are performed under the model assumptions and baseline parameter values summarized in Table 1.1.

These values were selected to ensure sufficient volatility and default activity for meaningful CVA exposure, while remaining within the range of realistic financial parameters. All simulation results involving the comparison of methods (tau-based vs. immersion),

Baseline Model Parameters	
Risk-free rate	$r = 0.02$
Loss-Given-Default	LGD $= 1.0$
Initial stock price	$S_0 = 100.0$
Strike price	$K = S_0$ (ATM call)
Time horizon	$T = 1.0$ year
CIR intensity parameters	
Mean reversion speed	$k = 0.1$
Long-run mean	$\theta = 0.1$
Volatility of intensity	$v = 0.2$
Initial intensity	$\lambda_0 = 0.02$
GBM asset parameters	
Volatility of stock	$\sigma = 0.3$

Table 1.1: Baseline parameter configuration for CIR intensity and GBM exposure dynamics.

discretization mesh (weekly vs. daily), and scenario count (e.g., $N_{MC} = 10^4$ to 10^6) are computed with the above baseline unless otherwise specified.

In the following subsections, we systematically examine how modeling choices—such as the representation of default time, time-step granularity, and number of Monte Carlo scenarios—influence CVA estimation under these fixed input parameters.

1.2.1 Comparison of τ-based vs. Immersion CVA Estimators

We assessed two Monte Carlo estimation methods for calculating CVA: the explicit simulation of the default time τ (the "τ-based" method) and the method employing the immersion hypothesis ("no-τ" method). Results were validated against the analytic CVA available at zero correlation ($\rho = 0$). The standard error (SE) and relative error are computed as:

$$\text{SE} = \frac{\text{Stdev(simulated CVA scenarios)}}{\sqrt{N_{MC}}}, \quad \text{Relative Error} = \frac{\text{CVA}_{\text{MC}} - \text{CVA}_{\text{analytic}}}{\text{CVA}_{\text{analytic}}}.$$

Table 1.2 summarizes the results across different time steps and sample sizes.

N_{MC}	Steps	**With τ**		**No τ**	
		CVA (SE)	Rel. Err / Time	CVA (SE)	Rel. Err / Time
10^4	52	0.2938 (0.0265)	−5.71% / 0.10s	0.3088 (0.0067)	−0.87% / 0.06s
	252	0.2773 (0.0288)	−10.99% / 0.39s	0.3194 (0.0068)	+2.50% / 0.31s
10^6	52	0.3137 (0.0031)	+0.70% / 11.09s	0.3110 (0.0007)	−0.19% / 8.93s
	252	0.3069 (0.0030)	−1.50% / 82.20s	0.3126 (0.0007)	+0.33% / 64.33s

Table 1.2: Comparison of τ-based and immersion-based CVA estimators at $T = 1$ year and $\rho = 0$, reporting Monte Carlo estimates with standard errors (SE), relative error (Rel. Err), and runtime.

The Wrong-Way Risk pattern remains clearly visible across all simulated scenarios. Table 1.3 illustrates one representative case to demonstrate how CVA responds to changes in the correlation parameter ρ under both estimation methods.

From these results, we observe the following:

ρ	**With τ**			**No τ**		
	CVA	SE	Δ	CVA	SE	Δ
0.00	0.2938	0.0265	0%	0.3088	0.0067	0%
0.25	0.4422	0.0395	+50.53%	0.3768	0.0081	+22.00%
0.50	0.4624	0.0395	+57.39%	0.4396	0.0096	+42.34%
0.75	0.4999	0.0464	+70.18%	0.5066	0.0120	+64.03%
1.00	0.6107	0.0479	+107.88%	0.5914	0.0138	+91.47%

Table 1.3: Impact of correlation ρ on CVA estimates WWR. Simulations use $T = 1$ year, $N_{MC} = 10{,}000$, and 52 steps per year. Relative deviations Δ are expressed with respect to the analytic benchmark CVA at $\rho = 0$, equal to 0.3116.

- **Reduced Relative Error via Immersion:** At low sample size ($N_{MC} = 10^4$), the immersion method exhibits significantly smaller bias. This is because the τ-based method relies on the indicator $\mathbf{1}_{\tau<T}$, which frequently evaluates to zero, leading to high variance and underestimation of CVA. In contrast, the immersion method bypasses this sparsity and directly estimates:

$$\text{CVA} = \text{LGD} \cdot \mathbb{E}^{\mathbb{Q}} \left[\left(1 - e^{-\int_0^T \lambda_s ds} \right) (S_T - K)^+ e^{-rT} \right],$$

  where λ and S are simulated jointly but τ and auxiliary exponential variable ξ are not needed.

- **Superior Computational Efficiency:** The no-τ estimator avoids both τ and ξ simulation. This streamlines the computation, leading to consistent runtime reductions of approximately 20% at $N_{MC} = 10^6$.

- **Numerical Stability:** Across both weekly and daily steps, the immersion method yields much lower SE, demonstrating faster convergence and greater robustness with respect to pathwise variance.

1.2.2 Impact of Weekly vs. Daily discretization

We next examine how discretization frequency affects CVA estimation accuracy using the immersion-based estimator, which we previously identified as optimal. Table 1.4 compares the accuracy, precision, and computational run-times for weekly (52 steps/year) and daily (252 steps/year) discretization:

Time-step	N_{MC}	CVA	SE	Rel. Error	Runtime
Weekly (52)	10^5	0.30314	0.00208	-0.80%	4.32s
Daily (252)	10^4	0.31936	0.00681	$+2.50\%$	0.31s
Weekly (52)	10^6	0.31098	0.00065	-0.19%	8.93s
Daily (252)	10^6	0.31258	0.00066	$+0.33\%$	64.33s

Table 1.4: Impact of discretization frequency and working point on CVA estimation ($T = 1$ year, $\rho = 0$, immersion-based estimator)

The analysis of Table 1.2 and Table 1.4 indicates that although daily discretization with 252 steps per year may in principle reduce the relative error and standard error compared to weekly discretization, the improvements are minor, or in some practical cases entirely absent, and are disproportionate to the substantial increase in computational effort. For instance, increasing the discretization frequency from weekly to daily at $N_{MC} = 10^6$ only

changes the relative error a little (from -0.19% to $+0.33\%$), yet it inflates computation time nearly sevenfold—from 8.93 seconds to 64.33 seconds.

This disproportionate increase in runtime highlights that discretization frequency is not the primary source of bias or error; instead, the full-truncation Euler scheme applied to the CIR intensity model plays a more dominant role. Thus, given the marginal benefits and significantly higher computational costs, weekly discretization emerges as the practical choice, delivering reliable accuracy with manageable computational overhead.

1.2.3 Choice of $N_{MC} = 10^5$ as a Practical Working Point

In determining a practical working point, we focus on a balance among accuracy, computational efficiency, and robustness across varied correlation scenarios. Table 1.4 highlights that employing $N_{MC} = 10^5$ scenarios with weekly discretization results in a relative error of -0.80% and a standard error around 2 basis points (0.00208). Such accuracy is entirely sufficient, particularly given that observed CVA variations induced by correlation effects typically exceed tens of percentage points.

Indeed, the weekly discretization mesh combined with the immersion-based estimator at $N_{MC} = 10^5$ presents a 1% bias and a standard error on the order of 2 basis points—far smaller than typical correlation-driven variations. This choice therefore provides a highly suitable accuracy-to-speed trade-off, allowing for efficient and practical scenario analyses.

Thus, we conclude that a weekly discretization schedule paired with the immersion-based estimator and $N_{MC} = 10^5$ represents the optimal practical configuration for our subsequent comprehensive CVA simulations. This choice ensures computational tractability without sacrificing the critical precision required for accurate risk assessment, robust dataset generation, and efficient neural network model training.

1.2.4 Final Parameter Grid for Simulations

Building on detailed numerical analysis above and in line with good practices from the literature [9], we have carefully constructed the final parameter grid for comprehensive CVA simulations. Our primary goal was to ensure numerical stability, realistic financial scenarios, and effective coverage of the relevant volatility spectrum.

A critical consideration was adherence to the Feller condition, which guarantees the positivity of default intensity λ_t in the CIR model, defined as [22]:

$$2k\theta \geq v^2. \tag{1.2.1}$$

Initial simulations indicated substantial numerical instability and unacceptable relative errors (exceeding 10%) for large values of volatility parameter v (notably $v = 0.3, 0.4, 0.5$). Such high v values generate excessive equivalent lognormal volatilities for λ, given by:

$$V_{\text{logn}} = \frac{v}{\sqrt{\lambda_0}}. \tag{1.2.2}$$

For instance, with initial intensities $\lambda_0 \in [0.01, 0.05]$, a moderate volatility parameter $v = 0.2$ produces lognormal volatilities from 89% to 200%, resulting in unrealistic and numerically unstable conditions. Additionally, high volatility scenarios frequently violated the Feller condition, causing simulated intensity paths to be artificially trapped at zero due to the Full Truncation Euler scheme, thus inflating Monte Carlo estimation errors.

Consequently, we refined the volatility grid to more moderate values:

$$v \in \{0.025, 0.05, 0.075, 0.1, 0.125\},$$

which yield manageable lognormal volatility ranges (for example, approximately 11%–25% when $v = 0.025$).

Building on the methodological insights of Brigo et al. [9], we explicitly removed parameter sets violating the Feller condition. An exhaustive analysis of the parameter grid indicated that out of 1250 combinations, 135 cases violated the Feller condition (violation ratio of 10.8%). Crucially, among the 1115 remaining compliant scenarios, 575 retained significant lognormal volatilities exceeding 40%, ensuring the preservation of high-volatility dynamics essential for robust CVA modeling.

The final parameter grid, ensuring numerical stability and financial realism, is thus defined as:

- **Initial default intensity (λ_0):** $\{0.01, 0.02, 0.03, 0.04, 0.05\}$
- **Mean reversion speed (k):** $\{0.1, 0.2, \ldots, 1.0\}$
- **Long-term mean (θ):** $\{0.01, 0.02, 0.03, 0.04, 0.05\}$
- **Volatility of intensity (v):** $\{0.025, 0.05, 0.075, 0.1, 0.125\}$
- **Asset volatility (σ):** $\{0.1, 0.2, 0.3, 0.4, 0.5\}$
- **Correlation (ρ):** $\{-1.0, -0.8, \ldots, 0.8, 1.0\}$

This selection results in a total of:

$$5 \times 10 \times 5 \times 5 \times 5 \times 11 \times (1 - 0.108) = 61{,}325$$

valid parameter combinations for data generation.

Moreover, throughout all simulations we adopt the baseline model parameters reported in Table 1.1: risk-free rate $r = 0.02$, loss-given-default LGD $= 1.0$, initial stock price $S_0 = 100.0$, strike price $K = S_0$ (ATM call), and time horizon $T = 1.0$ year. Unless otherwise stated, CVA values are computed using the immersion-based estimator with weekly discretization (52 steps per year) and a Monte Carlo sample size of $N_{MC} = 10^5$.

In summary, this refined and meticulously structured parameter grid ensures the generation of robust, precise, and computationally efficient datasets, optimally suited for training neural networks and enabling sophisticated financial risk assessments, especially under Wrong-Way Risk conditions.

1.3 Dataset Output and Storage

Results from the simulations are stored in a structured CSV file, `cva_grid.csv`. The final dataset contains 61,325 rows. The fields are:

- **Parameters**: λ_0, k, θ, v, σ, ρ.
- **Monte Carlo outputs**: CVA estimate (`CVA_MC`) and its standard error (`SE_MC`).
- **Analytic benchmark**: closed–form CVA for $\rho = 0$ (`CVA_analytic`) and the relative error `RelError_%`.

A full schema and sample rows are reported in Appendix B.1, see Tables B.1–B.2.

The dataset comprises 61,325 valid parameter tuples. Each Monte Carlo valuation uses $N_{\mathrm{MC}} = 10^5$ paths with fixed random seeds. On the $\rho = 0$ analytic slice, no entry exhibits $|\texttt{RelError\_\%}| > 5\%$.

Generating the full grid required approximately 18.7 hours of wall–clock time with the baseline engine. This figure is used as the reference runtime for the acceleration comparisons in Chapter 3.

1.4 Summary and Next Steps

The parameter grid is constructed as follows: $\lambda_0 \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$, $k \in \{0.1, 0.2, \ldots, 1.0\}$, $\theta \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$, $v \in \{0.025, 0.050, 0.075, 0.100, 0.125\}$, $\sigma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, and $\rho \in \{-1.0, -0.8, \ldots, 0.8, 1.0\}$. This yields an evenly spaced mesh across the parameter ranges. Parameter combinations that violate fundamental feasibility or stability conditions are excluded.

This dataset provides the foundation for training a neural surrogate to accelerate CVA computations under WWR. In addition to broad parametric coverage, the inclusion of the ρ=0 analytic slice offers an internal benchmark for sanity checks and error calibration, supporting systematic bias–variance diagnostics across the grid. The baseline generation time of 18.7 hours for 61,325 grid points serves as the reference against which the surrogate's inference speedups are quantified.

Subsequent chapters detail the neural network architecture, training methodology, performance evaluation, and interpretability analysis.

Chapter 2

Neural Network Construction

In order to construct a surrogate model for Monte Carlo–based CVA estimation, we design and compare four fully-connected feedforward neural network (FNN) architectures with 1 to 4 hidden layers. Unless otherwise stated, we take the 3-hidden-layer network as the running example to illustrate the construction and tuning pipeline. All implementations are based on TensorFlow/Keras [1, 17], and the dataset is the simulated grid `cva_grid.csv` described in Chapter 1.

2.1 Initial Setup

2.1.1 Overview of the Architectures

The four candidate network structures are defined as follows:

1. **1-layer network:** One hidden layer with u_1 units, denoted by Dense(u_1, ReLU).
2. **2-layer network:** Two hidden layers with (u_1, u_2) units respectively.
3. **3-layer network:** Three hidden layers with (u_1, u_2, u_3) units respectively.
4. **4-layer network:** Four hidden layers with (u_1, u_2, u_3, u_4) units respectively.

In all cases, the input layer has dimension 6, corresponding to the parameters $(\lambda_0, k, \theta, v, \sigma, \rho)$, and the output layer is a single linear neuron producing the CVA estimate. Hidden layers use ReLU activations.

2.1.2 Data Split and Preprocessing

We randomly split the dataset into an 85% training pool and a 15% test set, with a fixed `random_state` of 42 used throughout unless otherwise specified. Within the training pool, 10% is reserved as a validation subset via `validation_split=0.1` during training. As a result, the effective proportions with respect to the full dataset are as summarized in Table 2.1.

Split	Proportion	Purpose
Train (updates)	76.5%	Parameter learning
Validation	8.5%	Early stopping / model selection
Test	15%	Final generalization check

Table 2.1: Data split used in model development.

All six input features are standardized by a `StandardScaler` fitted on the training pool and applied to validation and test sets. The target (`CVA_MC`) is left in its natural scale so that RMSE can be interpreted directly as a percentage of the average CVA.

2.1.3 Training Protocol

All networks are trained using the mean squared error (MSE) loss and optimized via Adam with tuned learning rates. We report root mean squared error (RMSE) on the validation and test sets as the primary performance metric. To prevent overfitting and to enable efficient model selection, early stopping is employed. Training terminates early if no improvement is observed, and the best model weights are restored.

To ensure robustness against batch-size sensitivity, we conduct a small discrete sweep over batch sizes $\{128, 256, 512\}$, especially after finalizing the architecture and core hyperparameters. A batch size of 256 is used as the default throughout unless otherwise specified. All models are trained for at most 200 epochs. A detailed summary of the protocol is given in Table 2.2.

Setting	Value	Notes
Loss	MSE	Regression objective
Metric	RMSE	Reported on validation/test sets
Optimizer	Adam	With tuned learning rate
Activation	ReLU	Applied to all hidden layers
Epochs (max)	200	Early stopping enabled
Early stopping	Patience = 5 / 10	Restore best weights
Batch size	128 / 256 / 512	Discrete sweep; default = 256

Table 2.2: Common training protocol across architectures.

2.1.4 Initial Hyperparameter Selection (1–3 Layers)

Before a full-scale search, we set robust seeds from targeted exploratory tuning. They were stable across random initializations and gave a good bias–variance trade-off (Table 2.3):

Initial Set	Units per hidden layer	Learning rate
SetD (1-layer)	[128]	0.0010000
SetE (2-layer)	[128, 64]	0.0010000
SetC (3-layer)	[128, 64, 32]	0.0010000

Table 2.3: Initial seeds for 1–3 hidden layer networks.

Why these seeds are reasonable.

- *Function complexity.* The surrogate $f : \mathbb{R}^6 \to \mathbb{R}$ is smooth on the calibrated, bounded domain; the dominant nonlinearities come from the ρ-driven WWR coupling and the CIR-intensity tails. By universal approximation on compact sets, a single hidden layer with $\mathcal{O}(10^2)$ ReLU units (`SetD`) already provides a sufficiently rich piecewise-linear basis to fit such targets with low bias.

- *Capacity scaling.* Moving from one to two and three layers (`SetE`, `SetC`) increases representational efficiency: modest depth captures compositional structure with controlled parameter growth. The tapered widths ($128 \to 64 \to 32$) follow a "funnel" design that concentrates capacity near the input (where interactions are strongest), progressively reduces dimensionality, and improves conditioning without sacrificing expressiveness.

- *Optimization stability.* With standardized inputs, an Adam learning rate of 10^{-3} lies in a well-behaved regime—large enough to avoid slow drift, yet small enough to

prevent divergence or extended plateaus. Empirically, these seeds reach stable validation plateaus with low sensitivity to random initialization, making them reliable centers for localized hyperparameter refinement.

Taken together, the seeds are expressive enough to avoid underfitting, compact enough to provide implicit regularization, and stable enough to support efficient fine-tuning.

2.1.5 Four-Layer Initial Search

A 4-hidden-layer family introduces substantially more degrees of freedom than its 1–3 layer counterparts. First, parameter count grows superlinearly with width, so modest increases in per-layer units translate into large jumps in total capacity. Second, deeper compositions alter the optimization landscape: they tend to accentuate training pathologies—extended loss plateaus, higher sensitivity to sharp minima, and stronger learning rate-width coupling that makes a single, hand-picked step size suboptimal across widths. In such regimes, manual seeding is brittle: small changes in widths or learning rate can qualitatively flip training dynamics (e.g., delayed descent, premature overfitting, oscillatory validation curves). To remove hand-tuned bias and ensure a fair capacity comparison to the 3-layer model, we therefore learn the initial seeds for the 4-layer family via an automated, resource-aware search.

We adopt Keras-Tuner Hyperband [33], which allocates epochs adaptively, promoting promising trials while pruning weak ones, thus covering a wide design space under a fixed compute budget. The search spans:

- hidden-layer count $n \in \{3, 4\}$ (depth control under identical budgets),
- per-layer units in $[64, 1024]$ (step 64), chosen to traverse under- to over-parameterized regimes while remaining GPU-friendly,
- learning rate in $[10^{-4}, 10^{-2}]$ (log-scale), covering the stable range for Adam with standardized inputs.

Inputs are standardized; optimization uses Adam with the sampled learning rate; early stopping (patience = 5) and validation RMSE define the objective. After Hyperband selects the best configuration, we run a small batch-size sweep $\{128, 256, 512\}$ and retrain with early stopping on the full training pool (same validation split). This two-stage procedure (global search $\rightarrow$ local refinement) mitigates depth-induced instability while preserving efficiency.

This process yields two strong 4-layer seeds (Table 2.4):

Initial Set (4-layer)	Units per hidden layer	Learning rate
SetA	[384, 384, 128, 192]	0.0017755
SetB	[768, 320, 192, 512]	0.0014500

Table 2.4: Promising seeds identified for 4-layer networks via Hyperband.

To quantify the capacity jump that motivates automation, we report approximate parameter counts (including biases). For a Dense layer with input dimension n_{in} and output width n_{out}, the parameter count is

$$\#\text{params} \;=\; (n_{\text{in}} \times n_{\text{out}}) + (n_{\text{out}}) \;=\; (n_{\text{in}} + 1)\, n_{\text{out}}.$$

For an L-hidden-layer MLP with input dimension d, output dimension $y = 1$, and hidden widths $[u_1, \ldots, u_L]$, the total becomes

$$\#\text{params} \;=\; (d+1)\, u_1 \;+\; \sum_{i=1}^{L-1} (u_i + 1)\, u_{i+1} \;+\; (u_L + 1)\, y$$

(we use $d = 6$ and $y = 1$ throughout). Applying this formula (Table 2.5):

Model	Hidden units	#Params (approx.)
SetD	[128]	1,025
SetE	[128, 64]	9,217
SetC	[128, 64, 32]	11,265
SetA	[384, 384, 128, 192]	224,769
SetB	[768, 320, 192, 512]	412,417

Table 2.5: Parameter growth from 1 to 4 hidden layers (input dimension = 6, output = 1).

The jump from `SetC` to 4-layer seeds is over an order of magnitude(about 20–37×), which explains why automated, resource-aware initialization (Hyperband) is both necessary (to stabilize width–LR interactions at depth) and effective (to discover operating points that are simultaneously expressive and trainable without hand-crafted seeding).

2.1.6 Systematic Evaluation of Initial Sets

To compare depth and capacity on equal footing, each initial set is trained under an identical protocol: at most 200 epochs with early stopping (patience = 10), fixed batch size = 256, the data split described above (Train 76.5%, Validation 8.5%, Test 15%), standardized inputs, Adam optimization, and MSE loss. We report the held-out test RMSE and, for scale-aware interpretation, the same RMSE as a percentage of the average CVA (Table 2.6).

Set	RMSE	% of avg CVA
SetA_384_384_128_192	0.003829	1.00%
SetB_768_320_192_512	0.004157	1.08%
SetC_128_64_32	0.003815	0.99%
SetD_128	0.004378	1.14%
SetE_128_64	0.003592	0.93%

Table 2.6: Test performance of the initial sets under a unified training protocol.

Discussion. All five seeds achieve sub-1.15% relative error on the held-out test set, indicating that even comparatively compact architectures already approximate the CVA surface well on the simulated domain. The spread across configurations is modest: the best (`SetE`, 0.93%) and worst (`SetD`, 1.14%) differ by only 0.21 percentage points of the average CVA (i.e., 0.000786 in absolute RMSE). This narrow band suggests diminishing returns from additional depth once basic capacity is in place. Notably, the 2-layer `SetE` attains the lowest RMSE with far fewer parameters than the 4-layer seeds, underscoring strong parameter efficiency; the 3-layer `SetC` is statistically comparable (0.99%) and offers a balanced trade-off between expressiveness and model size, which motivates its use as the running example in subsequent fine-tuning. The 4-layer seeds (`SetA`, `SetB`) remain competitive but at a substantially higher parameter cost, justifying the automated initialization strategy used for that family. Overall, these results validate the initial search ranges and establish a high-performance baseline for the later, localized hyperparameter refinement.

2.2 Extensive Hyperparameter Search

We conduct an extensive, architecture-specific hyperparameter optimization using the three–hidden-layer family as a worked example. The objective is to refine the initial seed `SetC` ($[128, 64, 32]$, lr $= 10^{-3}$) by exploring a broad local neighbourhood of widths, learning rate, and light regularization while holding the data pipeline and objective fixed. We employ Keras-Tuner `RandomSearch` [33] with 500 trials under a fixed compute budget, using identical preprocessing (standardization), split (Train 76.5%, Validation 8.5%, Test 15%), loss (MSE), and optimizer (Adam) to ensure comparability across trials.

2.2.1 Methodology

The search space is centered on `SetC` but deliberately permissive to escape local optima, as detailed in Table 2.7: layer widths are allowed to expand/contract substantially, the learning rate is swept on a logarithmic scale, and light dropout is optionally applied after Batch Normalization (BN). BN follows each hidden layer to stabilize activations and reduce covariate shift, after which early stopping (patience = 15) and a Reduce-on-Plateau scheduler (factor 0.75, patience = 8, floor 10^{-5}) curb overfitting and unlock late-stage gains once validation loss stalls. The validation objective is RMSE. To isolate the effects of width, learning rate and dropout, we fix the batch size at 256 throughout.

Hyperparameter	Search range	Notes
u_0 (layer 1 units)	$[64, 1024]$ (step 32)	default 128
u_1 (layer 2 units)	$[32, 512]$ (step 16)	default 64
u_2 (layer 3 units)	$[16, 128]$ (step 16)	default 32
Dropout per layer	$\{0.00, 0.05, 0.10, 0.15, 0.20\}$	applied after BN
Learning rate (Adam)	$[10^{-4}, 3 \times 10^{-3}]$ (log)	default 10^{-3}

Table 2.7: Random-search space for the 3-layer architecture (500 trials).

Each trial trains for up to 100 epochs with early stopping and the scheduler active. After selecting the best hyperparameters on validation RMSE, we retrain the corresponding model for up to 300 epochs on the same Train/Validation split, again with early stopping and the scheduler, then perform a single evaluation on the test set. Reproducibility is ensured via `random_state` = 42 for the data split.

2.2.2 Search Outcome and Selected Configuration

Across 500 trials, the best validation RMSE reached 0.003787. The corresponding configuration (Table 2.8) concentrates capacity early and tapers aggressively, with essentially no explicit regularization beyond BN and a light first-layer dropout:

Parameter	Selected value	Notes
u_0, u_1, u_2	832, 160, 16	wide $\rightarrow$ medium $\rightarrow$ small taper
Dropout (d_0, d_1, d_2)	0.05, 0.00, 0.00	light regularization only at layer 1
Learning rate	1.793×10^{-3}	Adam (log-searched)

Table 2.8: Best 3-layer configuration returned by RandomSearch (validation objective).

For reference, the approximate parameter count (including biases) is 141,697, computed via

$$\#\text{params} = (d{+}1)u_1 + \sum_{i=1}^{L-1}(u_i{+}1)\,u_{i+1} + (u_L{+}1)\,y, \quad d = 6,\ y = 1.$$

Retraining this configuration with early stopping and the scheduler yields a held-out test RMSE of 0.007308 (i.e., 1.90% of average CVA).

2.3 Hyperparameter Fine-Tuning

Building on the 500-trial search, we execute a targeted fine-tuning stage that fixes the best three-layer architecture and refines only the optimization dynamics to improve generalization on the held-out test set. Concretely, we retain the widths and dropout learned previously and replace the training regime with (i) a cosine–annealing schedule with warm restarts and (ii) a robust regression loss, while keeping the data pipeline and split unchanged.

2.3.1 Methodology

The architecture is fixed to $(u_0, u_1, u_2) = (832, 160, 16)$ with per-layer dropout $(d_0, d_1, d_2) = (0.05, 0, 0)$ and Batch Normalization (BN) after each hidden layer. The optimization protocol is:

- **Learning rate schedule.** Adam with *CosineDecayRestarts* [28] (initial LR $= 1.793 \times 10^{-3}$; first decay window = 200 steps; $t_{\text{mul}} = 2.0$, $m_{\text{mul}} = 1.0$) to couple fast initial descent with periodic warm restarts that bias convergence toward flatter minima.

- **Loss.** *Huber* loss [26] (default δ) replaces MSE to suppress the influence of heavy-tailed Monte Carlo noise while remaining quadratic near zero error.

- **Early stopping.** *Patience* = 60 with best-weight restoration, allowing long annealing phases to complete before termination.

Inputs are standardized using the training-pool scaler; batch size is fixed at 256. Training runs for up to 600 epochs with a 10% validation split within the training pool; we then perform a single evaluation on the held-out test set. The fine-tuning setup is summarized in Table 2.9.

Component	Value	Notes
Widths / Dropout	$(832, 160, 16)$ / $(0.05, 0, 0)$	fixed from 500-trial search
Normalization	BN after each hidden	stabilizes activations
Loss	Huber	robust to outliers
Optimizer / LR	Adam / 1.793×10^{-3}	with CosineDecayRestarts
Schedule	first_decay_steps = 200	$t_{\text{mul}} = 2.0,\ m_{\text{mul}} = 1.0$
Batch size / Epochs	256 / 600 (max)	early stopping, patience = 60

Table 2.9: Fine-tuning setup (architecture fixed, optimization refined).

2.3.2 Model Size and Trainability

Including BN parameters, the model has 145,729 total parameters, of which 143,713 are trainable and 2,016 are non-trainable (BN running statistics). This matches the Dense-only accounting,

$$\#\text{params}_{\text{Dense}} = (d{+}1)u_0 + (u_0{+}1)u_1 + (u_1{+}1)u_2 + (u_2{+}1)y = 141{,}697,$$

with $d = 6$ and $y = 1$, plus BN contributions of 2 trainable (γ, β) and 2 non-trainable (running mean/var) parameters per BN unit:

$$\#\text{params}_{\text{BN}} = 4 \times (832 + 160 + 16) = 4{,}032,$$

yielding the reported totals (Table 2.10).

Quantity	Value	Explanation
Total params	145,729	Dense + BN (trainable & non-trainable)
Trainable params	143,713	Dense + BN (γ, β)
Non-trainable params	2,016	BN running stats

Table 2.10: Parameter accounting for the fixed $(832, 160, 16)$ model with BN.

2.3.3 Results

Training stopped early at epoch 197, restoring the best validation weights from epoch 137. The held-out test performance is:

$$\text{Test RMSE} \;=\; 0.003027 \quad (0.79\% \text{ of average CVA}) \;.$$

Compared with the pre-fine-tuning result from the extensive search (0.007308, i.e. 1.90%), this is a 58.6% reduction in RMSE without changing the architecture (Table 2.11).

Stage	Test RMSE	% of avg CVA
After 500-trial search (fixed LR/MSE)	0.007308	1.90%
Fine-tuned (cosine LR + Huber)	0.003027	0.79%

Table 2.11: Held-out test performance before and after fine-tuning (architecture fixed).

Why the error drops

- *Flatter minima via cosine annealing.* The cosine schedule with warm restarts lowers the effective step size in smooth phases and periodically re-expands it, improving exploration and biasing convergence toward flatter basins that generalize better than sharp minima.

- *Variance reduction via Huber loss.* Replacing MSE with Huber attenuates the gradient impact of rare, high-variance CVA targets (Monte Carlo tails), reducing estimator variance and preventing the optimizer from overfitting noisy outliers.

- *Isolation of optimization effects.* Keeping capacity (widths, dropout, BN) and the data split fixed removes confounders; the gain is therefore attributable to the optimization regime itself rather than added parameters or data leakage.

2.4 Depth-wise Search and Fine-Tuning

We apply the two-stage protocol introduced in the previous sections to each depth $L \in \{1, 2, 3, 4\}$: a 500-trial Keras-Tuner `RandomSearch` over widths, optional per-layer dropout, and Adam learning rate (Stage 1), followed by a fixed-architecture fine-tuning pass with Huber loss and cosine-annealed restarts (Stage 2). All experiments share the same pipeline:

standardized inputs; split (Train 76.5%, Val 8.5%, Test 15%); loss (MSE in Stage 1); callbacks (EarlyStopping, ReduceLROnPlateau in Stage 1; EarlyStopping in Stage 2); batch size 256.

Stage 1 discovers capacity placement under a simple, uniform regime; Stage 2 holds size fixed but improves optimization dynamics (robust loss + cyclic LR), converting validation gains into robust test improvements without increasing model size.

2.4.1 1 Layer (FCNN-1)

The best configuration identified from the 500-trial random search is summarized in Table 2.12:

Hyperparameter	Value	Notes
u_0	480	single hidden layer
Dropout (d_0)	0.10	after BN
Learning rate	6.20×10^{-4}	Adam (log-searched)

Table 2.12: Best 1-layer search configuration.

Search → Test. RMSE = 0.003833 (1.00% of avg CVA).
Fine-tuning (fixed $u_0 = 480$). Huber + CosineDecayRestarts (initial LR = 6.20×10^{-4}; first_decay_steps = 200; $t_{\text{mul}} = 2$, $m_{\text{mul}} = 1$); patience = 60.
Final Test. $\boxed{0.003377}$ (0.88%), ↓ 11.9% vs. search.

2.4.2 2 Layers (FCNN-2)

The best configuration identified from the 500-trial random search is summarized in Table 2.13:

Hyperparameter	Value	Notes
$[u_0, u_1]$	[512, 64]	funnel profile
Dropout (d_0, d_1)	(0.05, 0.20)	after BN per layer
Learning rate	1.529×10^{-3}	Adam (log-searched)

Table 2.13: Best 2-layer search configuration.

Search → Test. RMSE = 0.003717 (0.97%).
Fine-tuning (fixed $[512, 64]$). Huber + CosineDecayRestarts (initial LR = 1.529×10^{-3}; first_decay_steps = 200; $t_{\text{mul}} = 2$, $m_{\text{mul}} = 1$); patience = 60.
Final Test. $\boxed{0.003272}$ (0.85%), ↓ 12.0% vs. search.

2.4.3 3 Layers (FCNN-3)

The best configuration identified from the 500-trial random search is summarized in Table 2.14:

Hyperparameter	Value	Notes
$[u_0, u_1, u_2]$	[832, 160, 16]	wide→medium→small
Dropout (d_0, d_1, d_2)	(0.05, 0.00, 0.00)	after BN per layer
Learning rate	1.793×10^{-3}	Adam (log-searched)

Table 2.14: Best 3-layer search configuration.

Search → Test. RMSE = 0.007308 (1.90%).
Fine-tuning (fixed $[832, 160, 16]$). Huber + CosineDecayRestarts (initial LR = $1.793 \times$

10^{-3}; first_decay_steps = 200; $t_{\text{mul}} = 2$, $m_{\text{mul}} = 1$); patience = 60.
Final Test. $\boxed{0.003027}$ (0.79%), $\downarrow$ 58.6% vs. search.

2.4.4 4 Layers (FCNN-4)

Deeper models exhibit stronger LR–width coupling and a more multi-modal search landscape. To mitigate local-search bias we explore two complementary seed families as described before: a front-loaded family near $[768, 320, 192, 512]$ (`SetB`-like) and a balanced family near $[384, 384, 128, 192]$ (`SetA`-like). Each family undergoes an independent 500-trial search followed by the same fine-tuning recipe.

Seed Family I: around $[768, 320, 192, 512]$**.** (Tables 2.15)

Hyperparameter	Value	Notes
$[u_0, u_1, u_2, u_3]$	[832, 352, 192, 544]	front-loaded capacity
Dropout $(d_0, \ldots, d_3)$	(0.15, 0.00, 0.00, 0.25)	after BN per layer
Learning rate	5.81×10^{-4}	Adam (log-searched)

Table 2.15: Best 4-layer configuration (Seed Family I).

Search $\rightarrow$ *Test.* RMSE = 0.007738 (2.01%).
Fine-tuning (fixed $[832, 352, 192, 544]$*).* Huber + CosineDecayRestarts (initial LR = 5.81×10^{-4}; first_decay_steps = 200; $t_{\text{mul}} = 2$, $m_{\text{mul}} = 1$); patience = 60.
Final Test. $\boxed{0.003307}$ (0.86%).

Seed Family II: around $[384, 384, 128, 192]$**.** (Tables 2.16)

Hyperparameter	Value	Notes
$[u_0, u_1, u_2, u_3]$	[544, 640, 256, 64]	deeper middle, narrow tail
Dropout $(d_0, \ldots, d_3)$	(0.00, 0.05, 0.05, 0.10)	after BN per layer
Learning rate	1.157×10^{-3}	Adam (log-searched)

Table 2.16: Best 4-layer configuration (Seed Family II).

Search $\rightarrow$ *Test.* RMSE = 0.003858 (1.00%).
Fine-tuning (fixed $[544, 640, 256, 64]$*).* Huber + CosineDecayRestarts (initial LR = 1.157×10^{-3}; first_decay_steps = 200; $t_{\text{mul}} = 2$, $m_{\text{mul}} = 1$); patience = 60.
Final Test. $\boxed{0.003301}$ (0.86%).

2.4.5 Cross-Depth Comparison

Table 2.17 summarizes test RMSEs before and after fine-tuning across architectures of increasing depth.

Depth	Search Test RMSE	Fine-tuned Test RMSE	Relative drop
1 layer	0.003833 (1.00%)	0.003377 (0.88%)	11.9%
2 layers	0.003717 (0.97%)	0.003272 (0.85%)	12.0%
3 layers	0.007308 (1.90%)	0.003027 (0.79%)	58.6%
4 layers (best seed)	0.003858 (1.00%)	0.003301 (0.86%)	14.4%

Table 2.17: Search vs. fine-tuned test performance across depths (percentages w.r.t. average CVA). The 4-layer row reports the better of the two seed families at each stage.

Interpretive notes. Occasional cases where a Stage 1 configuration underperforms a handpicked baseline on the test set, yet the Stage 2 model surpasses both in all cases, are normal in deep learning and consistent with our design:

- *Inductive-bias shift.* Baselines in in Section 2.1.6 were evaluated under a simpler training regime: MSE loss, fixed learning rate, and no BN/Dropout. Stage 1, in contrast, augments the hypothesis space with BN and optional dropout, and uses longer, adaptive schedules (EarlyStopping + ReduceLROnPlateau). Such changes alter the inductive bias: architectures optimal under one regime may not transfer one-to-one to another. This explains why some baselines can outperform certain Stage 1 picks on the test set, even though those picks are validation-optimal under the Stage 1 regime.

- *Capacity–regularization–schedule alignment.* Generalization depends on aligning network width with both the type and strength of regularization and with LR dynamics. A compact 3-layer baseline (`SetC`, $[128, 64, 32]$) has low variance and is near its bias floor; adding BN and a long annealing schedule can over-regularize it, increasing its fine-tuned test RMSE to 0.004689 (1.22%). By contrast, a higher-capacity model ($[832, 160, 16]$) initially performs worse under MSE + fixed LR (test 0.007308; 1.90%) but benefits from the Stage 2 regime: Huber loss (tail robustness) damps sensitivity to noisy Monte Carlo targets, and cosine restarts encourage exploration of flatter minima, reducing error to 0.003027 (0.79%). This illustrates a key deep-learning principle: larger models often require stronger, structured regularization and exploratory schedules to realize their potential, whereas small models may saturate early.

- *Why fine-tuning the baseline itself may not win.* Applying Stage 2 directly to a handpicked, narrow baseline cannot overcome intrinsic capacity limits: with insufficient width, the network cannot capture the curvature and cross-feature interactions needed to match the expressivity of the best Stage 1 architecture. Thus, decisive gains require both (i) identifying an effective capacity allocation (Stage 1) and (ii) optimizing that capacity with a robust schedule (Stage 2).

- *Selection noise and consolidation.* A 500-trial search introduces multiple-comparisons noise: the top validation score is a noisy proxy for true generalization. Stage 2 mitigates this by "freezing" the architecture and re-optimizing with Huber + cosine restarts, which acts as a variance-reduction and signal-consolidation step. This explains the consistent Stage 2 gains across all depths, even when the Stage 1 result trails a simpler baseline.

Illustrative contrast (FCNN-3). While the example below is drawn from the 3-layer case, the same pattern holds across other depths. Table 2.18 contrasts two representative architectures: a high-capacity model that underperforms at Stage 1 but achieves the largest relative gain after fine-tuning, and a simpler baseline model that starts strong but degrades when over-regularized by the same schedule.

Brief summary across depths. Across depths, the two-stage protocol consistently reduces test RMSE without increasing model size. The best overall performance is achieved by the fine-tuned 3-layer network (0.003027, i.e., 0.79% of average CVA). Four-layer models also benefit from the same protocol and two-seed exploration, converging near 0.86% while mitigating local-search bias.

It is noteworthy, however, that the 1-layer model already attains 0.88%, while the 2- and 4-layer variants achieve 0.85% and 0.86%, respectively—differences that are marginal

Architecture	Stage-1 Test RMSE	Stage-2 Test RMSE
$[832, 160, 16]$ $(d_0, d_1, d_2) = (0.05, 0, 0)$ $\text{LR} = 1.793 \times 10^{-3}$	0.007308 (1.90%)	0.003027 (0.79%)
$[128, 64, 32]$ $(0, 0, 0)$ $\text{LR} = 10^{-3}$	0.003815 (0.99%)	0.004689 (1.22%)

Table 2.18: FCNN-3: interaction of capacity and fine-tuning regime (percentages w.r.t. average CVA).

in absolute terms when all errors lie below 1%. This raises a practical question: is reducing the error from 0.88% to 0.79% worth the cost of adding two additional layers? Increasing depth inevitably introduces more nonlinearity, longer backpropagation paths, and stronger non-local interactions, which can complicate interpretability, debugging, and theoretical analysis.

Motivated by this trade-off between marginal accuracy gains and added complexity, the next chapter focuses on a detailed comparison between the 1-layer and 3-layer architectures, examining their respective properties, behaviour under varying conditions, and interpretability characteristics.

Chapter 3

One vs. Three Layers: Performance, Efficiency, and Robustness

This chapter presents a comparative evaluation of the best–performing one–layer (FCNN–1) and three–layer (FCNN–3) feedforward neural network surrogates identified in Section 2.4. Both models are fine–tuned on the same training/validation split and evaluated on a held–out 15% test set. Beyond standard in–distribution performance metrics, the analysis investigates out–of–distribution (OOD) stability in structurally constrained and low–density regions of the parameter space, conducts a detailed local sensitivity analysis—paying particular attention to Wrong–Way Risk—and examines the ρ=0 slice against both closed–form analytic values and Monte Carlo benchmarks. The chapter concludes with a synthesis of results, highlighting key insights on model robustness and their implications for high–stakes credit valuation adjustment estimation.

3.1 Evaluation Protocol and Setup

We reload the saved models and scalers, reconstruct the 15% test split, and standardize inputs using the model–specific scaler prior to inference. Inference throughput is measured on the full test set (9,199 points) with batch size 1024 under identical hardware/software conditions.

3.2 Test Accuracy and Inference Throughput

Table 3.1 summarizes speed and accuracy. FCNN–1 processes $\approx$ 34k samples/s, while FCNN–3 processes $\approx$ 25k samples/s. The 3–layer model reduces test RMSE from 0.003377 (0.88% of average CVA) to 0.003027 (0.79%), a relative improvement of $\approx$ 10.4%; both errors remain below 1% in absolute terms.

Model	Test RMSE (% of avg CVA)	Throughput (samples/s)
1-layer (FCNN-1)	0.003377 (0.88%)	34,161
3-layer (FCNN-3)	0.003027 (0.79%)	25,252

Table 3.1: Speed/accuracy trade-off on the held-out test set.

Computational efficiency and accuracy. As reported in Section 1.3, generating the Monte Carlo dataset of 61,325 grid points required 18.7 hours (67,320 s), corresponding

to a baseline throughput of only ≈ 0.911 samples/s. Against this reference, FCNN–1 and FCNN–3 achieve throughput multipliers of $\approx 3.75 \times 10^4$ and $\approx 2.77 \times 10^4$, reducing the evaluation of the full grid to ~ 1.80 s and ~ 2.43 s, respectively. Thus, the surrogate delivers orders–of–magnitude acceleration while preserving sub-1% test error.

The trade-off between accuracy and speed is moderate: the 3–layer network improves test RMSE from 0.003377 (0.88% of average CVA) to 0.003027 (0.79%), a relative gain of about 10%, at the cost of a $\sim 26\%$ reduction in throughput. If sub-1% error is the operational threshold, FCNN–1 achieves it with higher speed and a simpler hypothesis class; if maximum accuracy is required and latency is less critical (e.g., overnight CVA grids), FCNN–3 provides a measurable improvement. The choice therefore hinges on whether the application is primarily latency- or accuracy-constrained.

3.3 OOD Stress Tests under Structural Constraints

For robustness analysis, we evaluate twenty stress scenarios comprising ten cases with $\rho = 0$ and ten with $\rho \neq 0$, deliberately selected outside the training support. The baseline model parameters are fixed as in Table 1.1, while structural admissibility is enforced through the filters specified in Section 1.2.4, including in particular the CIR Feller condition $2k\theta \geq v^2$.

We compare each surrogate's prediction with a high-precision Monte Carlo estimate; for $\rho = 0$ we additionally report the analytic benchmark (CIR default probability $\times$ Black–Scholes exposure).

Results: $\rho = 0$

params (lambda0,k,theta,v,sigma)	MC	Analytic	1-layer	3-layer	err1(%)	err3(%)
[0.039, 0.104, 0.016, 0.040, 0.138]	0.242215	0.241744	0.243235	0.237155	0.42%	-2.09%
[0.031, 0.295, 0.046, 0.044, 0.421]	0.560212	0.563603	0.561308	0.565120	0.20%	0.88%
[0.019, 0.847, 0.048, 0.053, 0.314]	0.375120	0.374361	0.371325	0.371822	-1.01%	-0.88%
[0.026, 0.885, 0.036, 0.049, 0.456]	0.544528	0.548561	0.545986	0.551531	0.27%	1.29%
[0.030, 0.809, 0.033, 0.068, 0.248]	0.328892	0.329603	0.329458	0.327320	0.17%	-0.48%
[0.029, 0.660, 0.042, 0.067, 0.201]	0.289580	0.289713	0.289531	0.289991	-0.02%	0.14%
[0.046, 0.849, 0.038, 0.045, 0.119]	0.241592	0.242128	0.243908	0.219380	0.96%	-9.19%
[0.026, 0.202, 0.049, 0.040, 0.065]	0.102212	0.102877	0.102310	0.131174	0.10%	28.34%
[0.033, 0.479, 0.048, 0.038, 0.230]	0.355477	0.357285	0.355138	0.340378	-0.10%	-4.25%
[0.042, 0.343, 0.017, 0.045, 0.235]	0.383725	0.385797	0.383922	0.383602	0.05%	-0.03%

Results: $\rho \neq 0$

params (lambda0,k,theta,v,sigma,rho)	MC	1-layer	3-layer	err1(%)	err3(%)
[0.030, 0.710, 0.016, 0.044, 0.527, 0.30]	0.591581	0.590980	0.589652	-0.10%	-0.33%
[0.050, 0.511, 0.007, 0.055, 0.058, 0.70]	0.147259	0.143874	0.207132	-2.30%	40.66%
[0.050, 0.342, 0.011, 0.033, 0.430, 0.00]	0.774874	0.766628	0.770658	-1.06%	-0.54%
[0.043, 0.290, 0.026, 0.036, 0.376, 0.00]	0.630839	0.633436	0.632313	0.41%	0.23%
[0.044, 0.665, 0.053, 0.081, 0.160, 0.80]	0.383168	0.382762	0.411268	-0.11%	7.33%
[0.017, 0.133, 0.043, 0.049, 0.207, 0.50]	0.190420	0.192727	0.190437	1.21%	0.01%
[0.048, 0.518, 0.016, 0.027, 0.090, 0.10]	0.184951	0.188950	0.200724	2.16%	8.53%
[0.024, 0.512, 0.022, 0.060, 0.394, 0.10]	0.384354	0.386487	0.386525	0.55%	0.56%
[0.028, 1.003, 0.013, 0.043, 0.221, 0.80]	0.246736	0.246279	0.241414	-0.19%	-2.16%
[0.043, 1.031, 0.006, 0.032, 0.127, 0.40]	0.182881	0.177897	0.162209	-2.73%	-11.30%

Summary and Interpretation

When evaluated on parameter configurations far from the training distribution, both surrogates exhibit some degradation. However, the 1-layer network generally maintains greater stability in extreme cases:

- **Worst 3-layer error ($\rho = 0$):** +28.34% at [0.026, 0.202, 0.049, 0.040, 0.065].

- **Worst 1-layer error ($\rho = 0$):** -1.01% at [0.019, 0.847, 0.048, 0.053, 0.314].
- **Worst 3-layer error ($\rho \neq 0$):** $+40.66\%$ at [0.050, 0.511, 0.007, 0.055, 0.058, 0.70].
- **Worst 1-layer error ($\rho \neq 0$):** -2.73% at [0.043, 1.031, 0.006, 0.032, 0.127, 0.40].

The 1-layer model, although less expressive and therefore less accurate within the training distribution, induces a smoother hypothesis class with lower effective degrees of freedom. This simplicity promotes regularity, suppresses excessive curvature, and reduces extrapolation risk when evaluated under distributional shift.

By contrast, the 3-layer model leverages its greater representational capacity to achieve lower in-distribution error (0.79% of average CVA). However, this same capacity increases variance and amplifies sensitivity to model misspecification, particularly in structurally constrained or low-density regions of the input space. The most pronounced errors typically arise when effective option volatility is extremely low (rendering absolute deviations disproportionately large in relative terms), when Wrong-Way Risk is strong (inducing sharp, nonlocal exposure–default interactions), or when parameters approach admissibility boundaries (e.g., near-binding Feller condition), where the training data are sparse and local geometry becomes unstable.

In conclusion, while FCNN-3 excels in approximating the target function within the training domain, FCNN-1 exhibits greater robustness under distributional shift. The elevated out-of-distribution variance observed in FCNN-3 provides compelling evidence of mild overfitting to the training manifold.

3.4 Local Sensitivity Analysis

We assess ceteris paribus sensitivities by sweeping one input at a time over its empirical range while holding the remaining coordinates at their training–set medians. For the correlation parameter ρ we additionally overlay a Monte Carlo baseline evaluated at the exact grid points used to generate the dataset. All sweeps use the same preprocessing pipeline for both surrogates: the 1–layer (FCNN-1) and the 3–layer (FCNN-3) networks.

3.4.1 Protocol

Let $x = (\lambda_0, k, \theta, v, \sigma, \rho)$. For each coordinate x_i we form a 100–point grid spanning the empirical min–max of x_i and set x_{-i} to the componentwise medians. We evaluate both networks on the standardized inputs and report predicted CVA as a function of x_i. For the Wrong–Way Risk sweep we use $(\lambda_0, k, \theta, v, \sigma) = (0.03, 0.5, 0.03, 0.075, 0.3)$ and the discrete ρ grid from the data; MC values are read from the same grid.

3.4.2 Default–intensity Level λ_0

Both surrogates display an almost perfectly linear, strictly increasing response (Figure 3.1). This pattern is consistent with CIR dynamics: a higher initial intensity λ_0 raises the pathwise integrated hazard over the horizon, increasing the exposure–weighted default probability and thus the CVA. The two curves are virtually indistinguishable across the full range, indicating good calibration on this monotone dimension.

3.4.3 Mean–reversion Speed k

The sensitivity of CVA to the mean–reversion speed k is non-monotone (Figure 3.2). At low values of k, increasing mean reversion may initially reduce CVA: a stronger pull toward the long-run level θ could dampen fluctuations in the default intensity, effectively

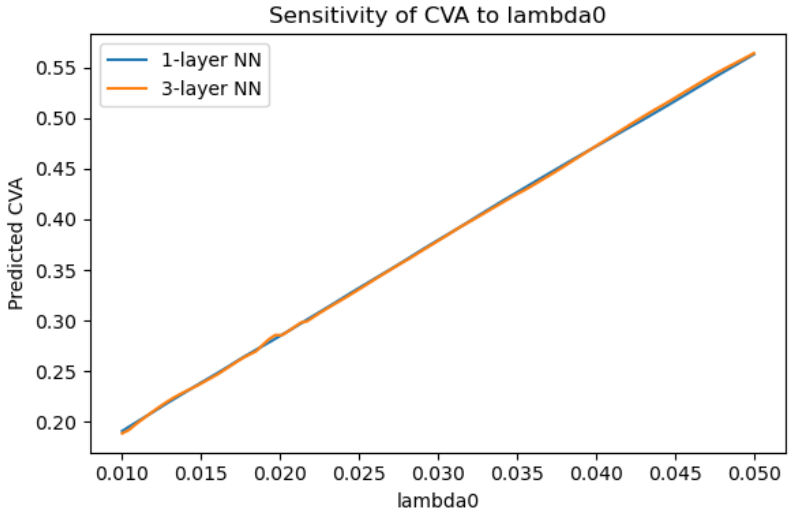

Figure 3.1: Sensitivity of CVA to λ_0 (FCNN-1 vs. FCNN-3).

smoothing the hazard process over time. This reduces the covariance between exposure and instantaneous default risk, thereby lowering the integrand in the CVA expression.

However, beyond a certain threshold, further increases in k may lead to a rise in CVA. As the intensity reverts more quickly, hazard mass become increasingly concentrated in the early stages of the contract, where exposure may remain significant. This temporal alignment could enhance the overlap between default likelihood and exposure, increasing overall CVA. Both surrogates capture the same qualitative timing-driven effect.

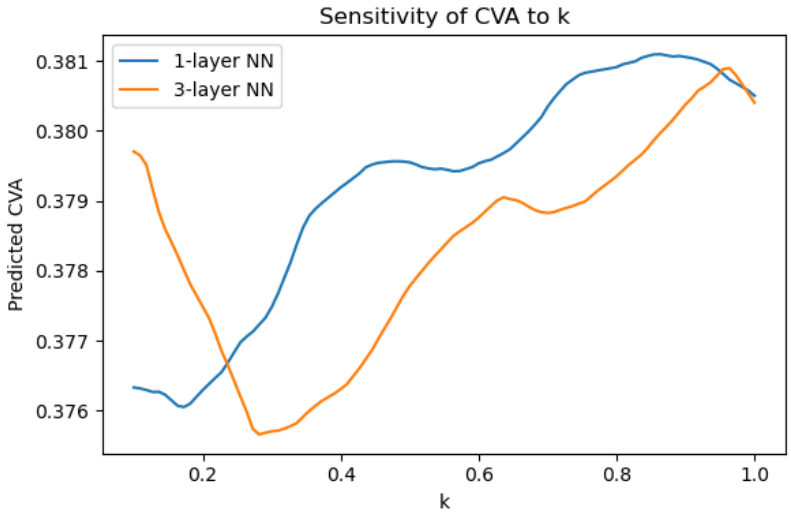

Figure 3.2: Sensitivity of CVA to k (FCNN-1 vs. FCNN-3).

3.4.4 Long–run Intensity θ

Responses are strictly increasing and nearly linear (Figure 3.3). Raising the long–run level θ shifts the intensity process upward on average, which increases the cumulative default probability and, in turn, the expected loss. FCNN-1 and FCNN-3 agree to within plotting resolution throughout, reflecting consistent learning of this effect.

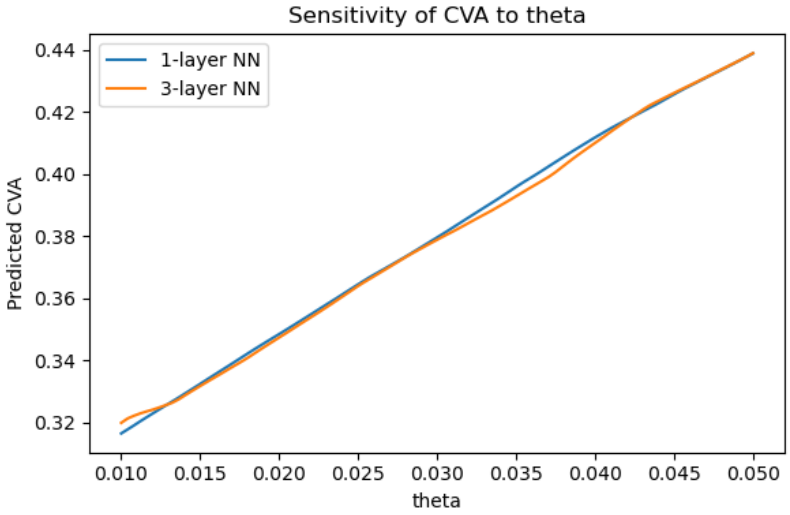

Figure 3.3: Sensitivity of CVA to θ (FCNN-1 vs. FCNN-3).

3.4.5 Intensity Volatility v

Both surrogates display a non-monotone dependence on v (Figure 3.4). CVA declines as v increases from approximately 0.03 to 0.06, reaches a minimum around $v \in [0.055, 0.065]$, and then exhibits a mild rebound over the range $v \approx 0.07$–0.10. Toward the upper end, FCNN-3 levels off, whereas FCNN-1 resumes a noticeable downward drift. Overall, FCNN-3 produces a smoother, lower-amplitude response, while FCNN-1 captures sharper curvature and larger excursions over the same interval.

These differences may reflect distinct inductive biases: FCNN-3, with higher capacity and stronger regularization, tends to smooth local irregularities—particularly in low-density regions of the training manifold—whereas the more constrained FCNN-1 may adhere more closely to the curvature suggested by the data.

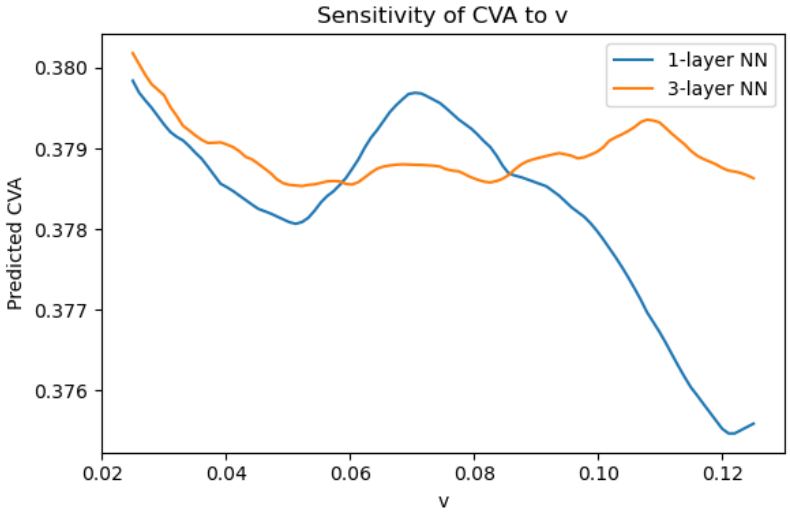

Figure 3.4: Sensitivity of CVA to v (FCNN-1 vs. FCNN-3).

3.4.6 Asset Volatility σ

CVA increases monotonically with σ (Figure 3.5), consistent with the convexity of option values and the amplification of exposure under higher underlying volatility. The two models track each other closely across the entire range; small slope differences at very low σ may be attributed to increased relative error when exposure is compressed, as marginal changes in σ have limited impact on the absolute exposure profile in this regime.

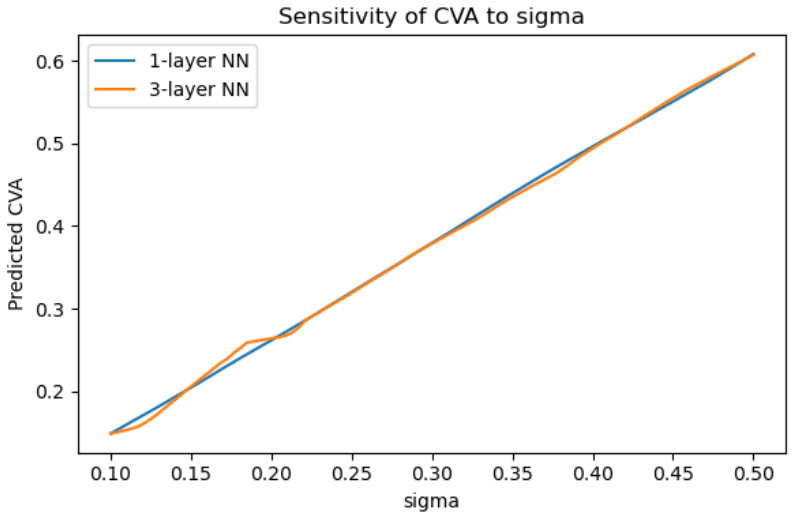

Figure 3.5: Sensitivity of CVA to σ (FCNN-1 vs. FCNN-3).

3.4.7 Correlation ρ (WWR)

Both surrogates accurately reproduce the Wrong-Way Risk effect: CVA increases monotonically and almost linearly with ρ (Figure 3.6). Using the grid point $(\lambda_0, k, \theta, v, \sigma) = (0.03, 0.5, 0.03, 0.075, 0.3)$ and sweeping the data's discrete ρ values, the 1– and 3–layer curves are essentially indistinguishable from the Monte Carlo baseline across the entire range, matching both level and slope. Small residual gaps near the extreme positive end are minor and do not alter the qualitative conclusion: the networks have learned the WWR mapping with high fidelity, capturing the strengthening exposure–hazard co-movement as ρ increases.

3.5 ρ=0 Sanity Check: Closed Form vs. Monte Carlo vs. Surrogates

In the uncorrelated case ρ=0, exposure and default are independent, and CVA admits a closed–form factorization:

$$\mathrm{CVA}_{\mathrm{analytic}}(T) = \mathrm{LGD} \times \underbrace{\mathbb{Q}(\tau \le T)}_{\text{CIR default probability}} \times \underbrace{C_{\mathrm{BS}}(S_0, K, r, \sigma, T)}_{\text{Black–Scholes call price}} .$$

This provides a natural benchmark against which to validate numerical results. Specifically, we use the ρ=0 slice of the dataset to cross–check Monte Carlo estimates and the outputs of the trained surrogates (FCNN–1 and FCNN–3). For each parameter configuration with ρ=0, we compare the surrogates' predictions against both the MC estimate $\mathrm{CVA}_{\mathrm{MC}}$ and the analytic benchmark (1.1.17), and report mean absolute errors (MAE).

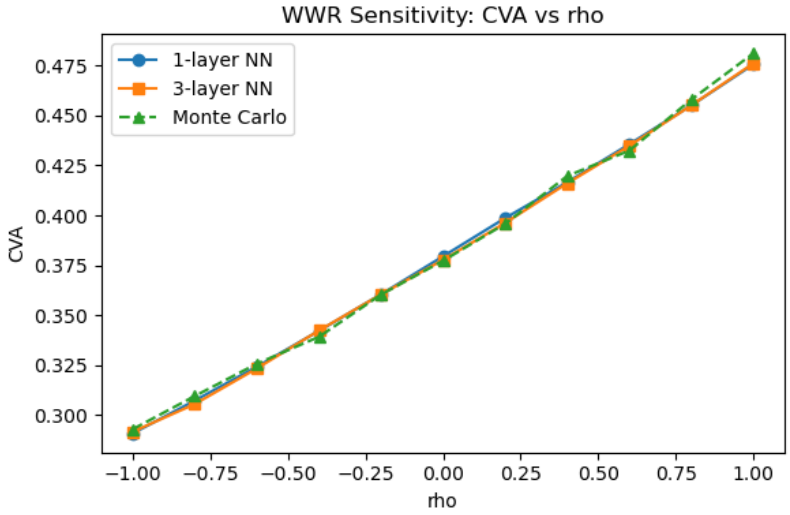

Figure 3.6: WWR sensitivity: CVA vs. ρ (FCNN-1 & FCNN-3) with MC baseline.

Model	MAE vs. MC	MAE vs. Analytic
1-layer (FCNN-1)	0.002238	0.001481
3-layer (FCNN-3)	0.001983	0.001175

Table 3.2: Mean absolute deviations on the ρ=0 grid. Both surrogates are closer on average to the closed-form benchmark than to the MC labels.

Result. Across all ρ=0 grid points, both surrogates align more closely with the analytic solution than with the MC baseline (Table 3.2). In particular, FCNN-3 attains the smallest deviation from the closed form (MAE $= 1.175 \times 10^{-3}$), despite being trained on $\mathrm{CVA}_{\mathrm{MC}}$ targets.

Interpretation. The finding that both surrogates are, on average, closer to the closed form than to the Monte Carlo labels on the ρ=0 slice is consistent with the networks attenuating label noise and small MC biases. Several mechanisms could plausibly contribute:

- *Sources of MC bias at* ρ=0. For computational efficiency, the MC engine employs a terminal–payoff proxy rather than simulating the default time explicitly. Although independence factorization holds exactly at ρ=0, residual bias can arise from discretization and sampling. In particular, simulating CIR intensities via the Full Truncation Euler scheme with a finite time step can introduce a nonzero discretization error in the path integral $\int_0^T \lambda_t \, dt$, especially when volatility is high or the Feller condition is nearly violated. This bias propagates directly into $\mathrm{CVA}_{\mathrm{MC}}$, and is compounded by the finite number of paths used.

- *Regularization as noise smoothing.* The surrogates are trained with standardization, early stopping, and, during fine-tuning, a robust Huber loss and cosine–annealed schedules. These design choices penalize fitting high–frequency label perturbations and encourage smoothness across neighboring grid points. Under standard empirical–risk–minimization arguments, when labels equal the true target plus zero–mean noise, a sufficiently regularized regressor will converge toward the conditional expectation and not replicate the noise. This can make the learned surface closer to the closed form than to any single noisy MC realization.

- *Inductive bias toward smooth structure.* On the ρ=0 manifold the map $(\lambda_0, k, \theta, v, \sigma) \mapsto$

CVA is smooth and admits a closed–form factorization. Architectural and optimization biases (weight decay, batch normalization, dropout, cosine restarts) may favor such low–curvature structure, thereby underweighting small, non–systematic MC idiosyncrasies that do not align with a globally smooth surface.

- *Cross–manifold continuity in ρ.* The networks are trained jointly over $\rho \in [-1, 1]$ and implicitly enforce continuity in ρ. The restriction of the learned map to ρ=0 therefore acts as a regularized projection of the full surface as $\rho \to 0$, naturally pulling predictions toward the analytic limit when correlation vanishes.

3.6 Summary and Implications

Computational efficiency. Relative to the Monte Carlo baseline, the surrogate models provide substantial acceleration. FCNN–1 and FCNN–3 sustain throughputs of approximately 34,161 and 25,252 samples/s, corresponding to speedups of about 3.75×10^4 and 2.77×10^4, respectively. while preserving test errors below 1%. This efficiency gain transforms overnight Monte Carlo runs into near real–time inference, making it feasible to conduct large–scale sensitivity analyses and WWR stress tests that would otherwise be computationally prohibitive.

Model robustness. Although the 3–layer (FCNN-3) architecture attains the lowest in-distribution RMSE, this advantage does not carry over uniformly under distributional shift. Stress tests in low-density and structurally constrained regions of the parameter space reveal materially weaker stability, with outliers exceeding 40%. This is a central result: for high-stakes tasks such as CVA estimation, superior in-distribution fit should not be the sole selection criterion. Rather, model choice must balance accuracy against generalization and stability under shift, reflecting a classical bias–variance trade-off—lower in-sample bias achieved by higher capacity may come at the cost of higher out-of-sample variance.

Implications. It is notable that both surrogates are, on average, closer to the closed-form analytic surface than to the Monte Carlo labels they were trained on along the ρ=0 slice. An interpretation is that the networks learn the underlying signal (the conditional expectation) while attenuating small MC noise and discretization bias. This behavior is plausibly induced by the interplay of regularization (standardization, early stopping, Huber fine-tuning, cosine annealing), architectural inductive biases toward smooth mappings, and continuity learned across ρ. Practically, this suggests that well-regularized neural surrogates can, in regimes with reliable theoretical structure, yield estimates more consistent with known limits than raw simulation outputs. At the same time, the documented OOD variability—especially for FCNN-3—argues for complementary safeguards (analytic cross-checks where available, OOD detection, and targeted augmentation) in any production deployment.

Chapter 4

Model Interpretability via SHAP

4.1 Motivation and Setup

To complement the preceding chapter's quantitative evaluation of the CVA surrogates, we now undertake a global–local interpretability analysis based on SHAP (Shapley Additive exPlanations). This methodology has been rigorously investigated in the financial deep learning literature [30, 32, 36], and is employed here to provide interpretability for the trained surrogates.

The analysis is conducted on the trained 1-layer and 3-layer neural networks, with input features (λ_0, k, θ, v, σ, ρ). Unless otherwise specified, and consistent with the experimental design adopted earlier, the option maturity is fixed at $T = 1$ year, the baseline model parameters in Table 1.1 are held constant, the dataset is split with a 15% test set, and the SHAP computation employs a background sample of $N_{\text{bg}} = 1000$ randomly selected, non-repeating observations from the test set.

Each surrogate model uses its own fitted scaler to standardize the features prior to explanation.

SHAP is grounded in the Shapley value concept from cooperative game theory, which provides a principled method for attributing to each "player" the average marginal contribution it makes when joining all possible coalitions. Let $P = \{p_1, \ldots, p_n\}$ denote the set of players and $f(G)$ the value generated by a coalition $G \subseteq P$. The Shapley value ϕ_k assigned to player p_k is given by [34]

$$\phi_k \;=\; \sum_{G \subseteq P \setminus \{p_k\}} \frac{|G|!\,(n - |G| - 1)!}{n!} \big(f(G \cup \{p_k\}) - f(G)\big), \tag{4.1.1}$$

where $|G|$ is the size of coalition G and $n = |P|$ is the total number of players. The weighting factor $\frac{|G|!\,(n-|G|-1)!}{n!}$ ensures that each possible coalition size is equally represented, and the term $f(G \cup \{p_k\}) - f(G)$ represents the marginal gain achieved when p_k joins coalition G. The sum thus yields the average marginal contribution of p_k across all possible coalitions.

In the machine learning setting, the "players" correspond to the input features of a single instance x, the "game" is the prediction task, and $f(G)$ denotes the model's expected output when only the features in G are fixed to the values in x while the others are marginalized over a reference (background) distribution. In this context, the Shapley values satisfy the additive decomposition [30]

$$f(x) \;=\; \mathbb{E}[f(X)] \;+\; \sum_{j=1}^{M} \phi_j(f, x), \tag{4.1.2}$$

where $\mathbb{E}[f(X)]$ is the baseline prediction over the background distribution, and $\phi_j(f, x)$ quantifies the signed contribution of feature j to the deviation of $f(x)$ from this baseline. A positive ϕ_j indicates that feature j increases the prediction relative to the baseline, whereas a negative value indicates a decreasing effect.

Computing exact Shapley values requires summation over all 2^{M-1} possible coalitions, which quickly becomes intractable as M increases. To address this, SHAP introduces efficient approximation schemes tailored to different model classes [18]. For deep neural networks, the DeepSHAP algorithm combines SHAP's game-theoretic attribution framework with DeepLIFT-style backpropagation rules [35, 2] to propagate contributions through network layers while using a chosen background set to estimate conditional expectations. In this study, we adopt DeepSHAP with the above-defined test-set background, as used by Yuan et al. [36], to measure global feature importance via the mean absolute SHAP value (averaged over all explained instances) and to visualize local effects through beeswarm plots that depict the distribution, sign, and magnitude of ϕ_j. This unified setup allows us to identify and rank the dominant economic drivers of CVA, assess their directional impact (e.g., the Wrong–Way Risk amplification induced by ρ), and compare the learned sensitivities between the simple and deep surrogate architectures in a methodologically consistent manner.

4.2 Global Importance

Figure 4.1 presents the mean absolute SHAP values for each input feature, with the corresponding numerical magnitudes reported in Table 4.1. Both surrogates exhibit an identical ordering of feature importance:

$$\sigma \;>\; \lambda_0 \;>\; \rho \;>\; \theta \;>\; v \gtrsim k,$$

indicating strong concordance in identifying the dominant drivers of the CVA surface at the T=1 year horizon.

The relative contribution (share) of each feature j is computed as

$$p_j = \frac{s_j}{\sum_m s_m},$$

where s_j denotes the mean absolute SHAP value for feature j. Percentages quoted in the text correspond to $p_j \times 100\%$. Under this normalization, the top three drivers $(\sigma, \lambda_0, \rho)$ jointly account for approximately 84.5% (1-layer) and 84.7% (3-layer) of the total attribution mass, with σ contributing roughly 39%, λ_0 about 32%, and ρ around 13%. The remaining parameters have comparatively limited influence: θ contributes about 10%, while v and k each contribute less than 3%.

Across architectures, the absolute importances are closely aligned. Transitioning from the 1-layer to the 3-layer surrogate yields slight decreases in the attributions for the main drivers, offset by marginal increases in v and k. These variations are numerically small and do not affect the ranking, confirming that both surrogates have effectively learned the same global sensitivity structure. For comparability, all features are standardized using model-specific scalers prior to SHAP computation, and global importance is consistently measured as the mean absolute SHAP value.

Summary of key observations. (i) Exposure volatility σ, initial intensity λ_0, and dependence ρ dominate (together $\approx 85\%$ of the total absolute attribution).
(ii) θ has a moderate but non-negligible contribution.
(iii) Intensity volatility v and mean-reversion speed k are minor at the one-year horizon.

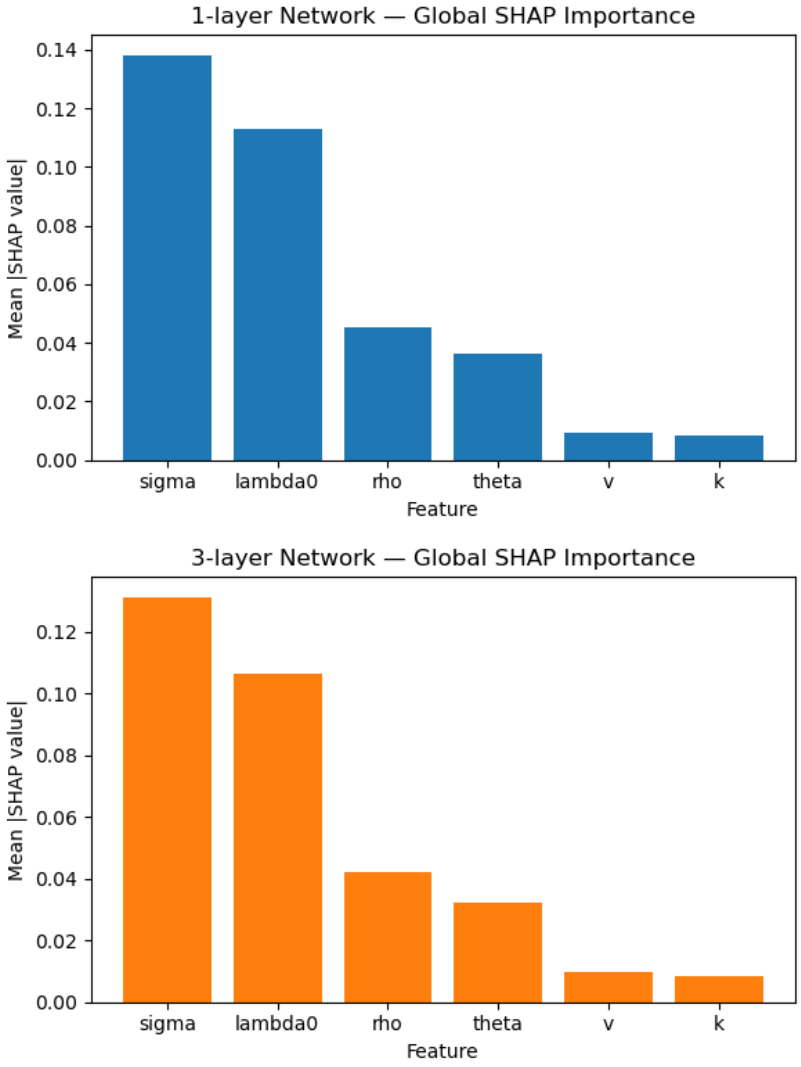

Figure 4.1: Global SHAP importance (mean absolute SHAP) for the two surrogates. Left: 1-layer network; Right: 3-layer network.

Feature	**1-layer**	**3-layer**
σ	0.1381	0.1312
λ_0	0.1132	0.1063
ρ	0.0451	0.0424
θ	0.0364	0.0324
v	0.0095	0.0096
k	0.0084	0.0085

Table 4.1: Mean absolute SHAP values (global importance). Percent shares equal each entry divided by the column-wise sum (0.3507 for 1-layer; 0.3304 for 3-layer).

4.3 Local Effects and Directionality

Figure 4.2 displays the SHAP beeswarm plots for the 1-layer and 3-layer surrogates. In each plot, every point corresponds to a single explained instance, with the horizontal axis representing the SHAP value (i.e., the signed contribution of the feature to the prediction relative to the baseline) and the colour indicating the actual feature value (from low in blue to high in red). The vertical ordering follows the global ranking of features, enabling

direct comparison between global and local patterns.

The beeswarm visualizations corroborate the global importance ordering reported in Section 4.2, while also providing directional insights into how feature values influence CVA. A pronounced directional asymmetry is observed for ρ: high values of ρ (in red) are predominantly associated with positive SHAP values, indicating that increased dependence between exposure and default intensity raises CVA — a manifestation of the Wrong–Way Risk effect. Conversely, low ρ values (in blue) tend to have negative SHAP contributions, consistent with right–way risk scenarios.

For σ and λ_0, the local effects are also strongly monotonic and of substantial magnitude. Higher σ broadens the exposure distribution, increasing the likelihood of large positive exposures coinciding with default events, thereby amplifying CVA. Similarly, a larger λ_0 raises the near-term default intensity, directly increasing expected losses and thus CVA. Both features exhibit a wide spread of SHAP values, reflecting their high global importance and strong influence across diverse parameter combinations in this one-year maturity setting.

In contrast, θ, v, and k exhibit more narrowly clustered SHAP distributions centered around zero, with v and k in particular displaying a more symmetric spread. This indicates weaker and less directional influence in the one-year maturity setting. It aligns with the economic interpretation that, over short horizons, the mean-reversion parameters of the CIR intensity process and its volatility play a comparatively less important role in determining CVA.

4.4 Economic Interpretation

This section examines the economic implications of the SHAP results presented in Figure 4.1 and Figure 4.2, with numerical values reported in Table 4.1. The results are highly consistent across architectures and align closely with financial intuition for short-maturity CVA.

Throughout this section, all expectations and probabilities are understood to be taken the risk–neutral measure $\mathbb{Q}$. When writing $\mathbb{E}[\cdot]$, we therefore refer to risk–neutral expectations unless explicitly stated otherwise.

4.4.1 Comparative Interpretation: 1-layer vs. 3-layer

The SHAP patterns in Figures 4.1–4.2 are highly consistent across architectures, with identical feature rankings and close absolute magnitudes. Over the calibrated grid, this suggests that the CVA map is largely linear or only mildly nonlinear in its inputs. The deeper 3-layer surrogate captures slightly steeper local gradients but does not materially change the attribution structure or the economic reading; the simpler 1-layer network already recovers the dominant drivers effectively.

4.4.2 Dominant Short-term Drivers

As shown in the above sections, CVA sensitivity at the one-year horizon is concentrated in three parameters: exposure volatility σ, initial intensity λ_0, and correlation ρ. The remaining parameters—long-run mean θ, intensity volatility v, and mean-reversion speed k—exhibit comparatively modest contributions. The discussion below focuses on the economic mechanisms underpinning these sensitivities.

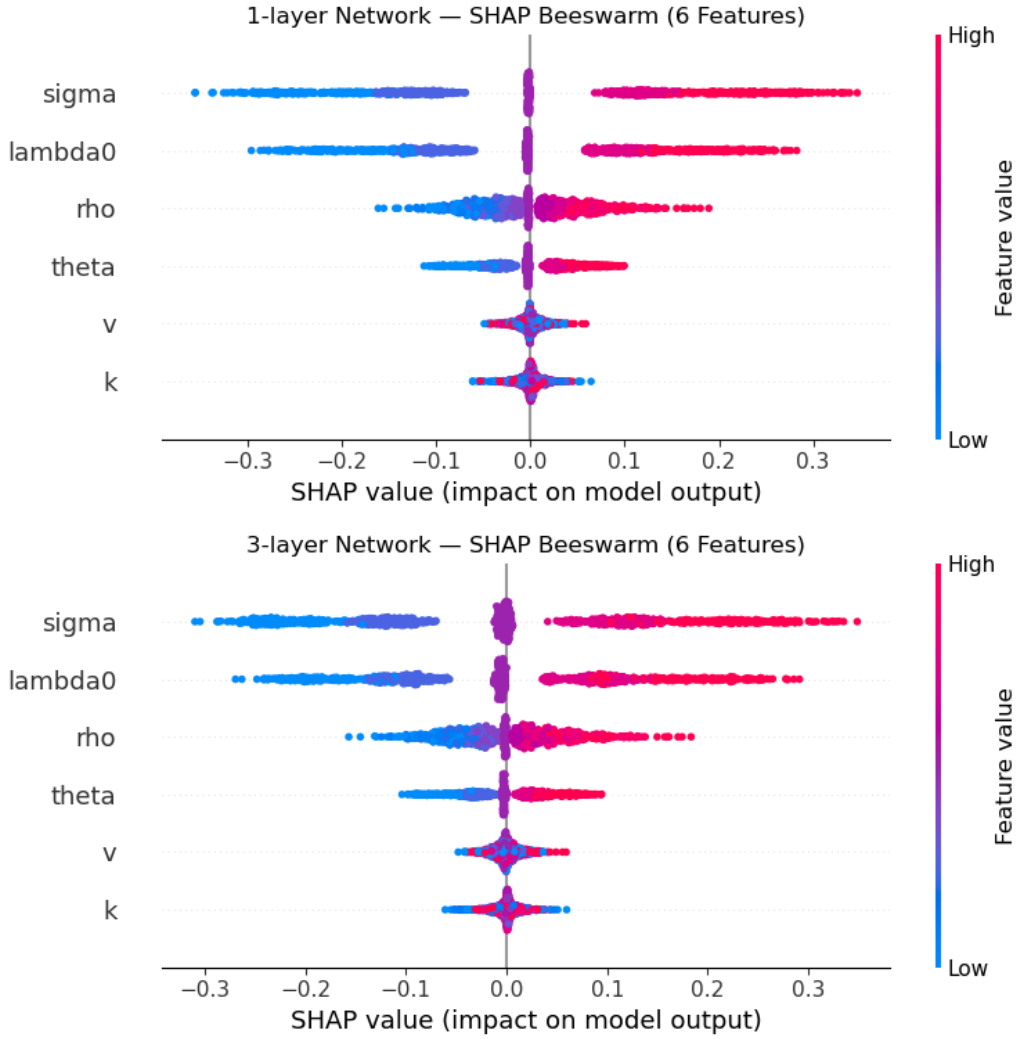

Figure 4.2: SHAP beeswarm plots (local effects, coloured by feature values). Top: 1-layer surrogate; bottom: 3-layer surrogate. The horizontal axis shows the SHAP value (impact on the model output relative to the baseline), while colour encodes the actual feature value.

Exposure Volatility σ

From the definition of CVA in (1.1.4), the valuation can be interpreted as that of a call option on the exposure, which is exercised only in the event of counterparty default prior to maturity, at a random time τ. In this option-like structure, the volatility parameter σ plays a role analogous to its function in classical option pricing. Since European call options exhibit positive vega—their value increases with volatility—greater σ broadens the distribution of potential exposures and, in expectation, increases their magnitude. It follows that CVA tends to rise monotonically with volatility.

This theoretical interpretation aligns with the SHAP analysis, which attributes a consistently positive and monotonic effect to σ. The convexity embedded in the exposure profile ensures that volatility enhances the potential loss, not only under independence (when the impact arises solely through the exposure component), but even more so under Wrong–Way Risk, where positive correlation between exposure and default intensity introduces an additional amplification mechanism.

Short-maturity Horizon Effects λ_0

In our CVA setup the option maturity is fixed at $T = 1$ year. This relatively short horizon is central to the observed SHAP sensitivities—most notably the prominence of the initial intensity λ_0 compared with the mean–reversion parameters (θ, k). Under the CIR specification, the risk–neutral mean intensity takes the closed form [19]

$$m(t) := \mathbb{E}[\lambda_t] = \theta + (\lambda_0 - \theta)\, e^{-kt},$$

so that deviations from the long–run mean decay exponentially as $|m(t)-\theta| = |\lambda_0-\theta|\, e^{-kt}$. The characteristic time scale of mean reversion is therefore $1/k$ (with half–life $(\ln 2)/k$), a standard interpretation in intensity–based and short–rate modeling [11].

Over a one–year horizon, unless k is sufficiently large, the intensity remains close to its initial level and has limited opportunity to migrate toward θ. Accordingly, the accumulated default risk over $[0, T]$ is driven primarily by the starting value λ_0, while θ and k enter only as secondary, horizon–dependent corrections. The volatility parameter v influences the dispersion of default risk, but within short horizons its effect remains comparatively modest unless it is large enough to induce substantial variability in the survival curve.

This horizon effect accounts for the SHAP rankings in Table 4.1 and the local patterns in Figure 4.2: λ_0 receives the largest attribution among the intensity parameters, whereas θ and k play a more modest role. At longer maturities—or under very rapid mean reversion (large k)—the influence of (θ, k) would be expected to strengthen, as the intensity path has more time to converge toward its long–run mean.

Role of Correlation and WWR

From the CVA definition in (1.1.4), losses are realized only at the random default time τ and depend on the prevailing exposure at that time. The correlation parameter ρ couples the Brownian shocks driving the asset price and the default intensity in (1.1.1)–(1.1.2), and therefore governs the joint occurrence of large exposure and elevated default risk.

Let V_t denote the undiscounted portfolio value at time t under $\mathbb{Q}$ and define the positive exposure $E_t := V_t^+$. For a European call option maturing at T with strike K,

$$V_t \;=\; \mathbb{E}^{\mathbb{Q}}\big[D(t,T)\,(S_T - K)^+ \mid \mathcal{F}_t\big].$$

Define the default–weight process

$$W_t := D(0,t)\,\lambda_t \exp\Big(-\int_0^t \lambda_u\, du\Big) \;\geq 0.$$

By Appendix A.1, under the immersion hypothesis the compact representation (1.1.9) is equivalent to the intensity–based formulation

$$\text{CVA} \;=\; \text{LGD} \int_0^T \mathbb{E}^{\mathbb{Q}}\big[E_t\, W_t\big]\, dt.$$

This makes explicit that the dependence on ρ enters solely through the covariance term

$$\mathbb{E}[E_t W_t] \;=\; \mathbb{E}[E_t]\mathbb{E}[W_t] \;+\; \text{Cov}(E_t, W_t).$$

Within the GBM–CIR framework, the marginals of E_t and W_t are determined by the asset volatility σ and the intensity parameters $(\lambda_0, \theta, k, v)$, and are unaffected by ρ. What ρ controls is their joint dependence: for the long–convex exposure $E_t = f(S_t)$, upward shocks to the underlying raise E_t, while conditional on the past intensity path, higher λ_t increases W_t. A positive ρ therefore increases the likelihood that both E_t and W_t are

simultaneously large, strengthening $\mathrm{Cov}(E_t, W_t)$ and amplifying CVA (Wrong–Way Risk) [12, 14]. Conversely, negative ρ aligns high exposures with lower default risk, mitigating CVA.

The strength of this effect scales with the volatilities of both components: larger σ widens the dispersion of E_t, while larger v widens that of W_t. When both are significant, the amplification under $\rho > 0$ is most pronounced. These mechanisms are reflected in the SHAP diagnostics: increasing ρ shifts SHAP contributions to the right (higher CVA), while low or negative ρ shifts them left, consistent with the global rankings in Table 4.1.

Modest Influence of v and k in a Short-maturity Setting

Consistent with the short-horizon analysis in Short-maturity Horizon Effects λ_0, a one–year horizon provides little scope for mean reversion. Unless k is very large (so that the characteristic scale $1/k$ is much shorter than one year), the intensity remains close to its initial level, and variations in k induce minor shifts in the accumulated default mass. Consequently, CVA's sensitivity to k is weak in a short horizon.

The intensity volatility v contributes primarily by introducing dispersion into the intensity path and the survival curve. In our calibration, v is deliberately confined to the range $[0.025, 0.125]$ to ensure numerical stability and preserve the Feller condition. This narrow band limits the stochasticity of λ_t, making the marginal effect of v on CVA comparatively small relative to dominant drivers such as λ_0 and the exposure volatility σ.

Importantly, a small v does not nullify the effect of correlation. Through the covariance channel, even limited randomness in λ_t can interact with a volatile exposure process to generate a meaningful Wrong–Way Risk effect when $\rho > 0$. Correlation requires variability on both sides to exert influence: if $v = 0$, ρ becomes irrelevant; but as soon as $v > 0$, a sufficiently large σ can amplify the effect of ρ to an economically significant level.

At longer maturities—or with substantially larger k or v—the influence of mean reversion and intensity dispersion is expected to increase, enhancing the overall impact of these parameters on CVA.

4.5 Summary and Implications

Model interpretability. SHAP diagnostics provide a transparent, model-agnostic decomposition of the surrogate CVA maps. Across all architectures, the global importance ranking remains stable, $\sigma > \lambda_0 > \rho > \theta > v \gtrsim k$, with the top three drivers capturing almost the entire attribution mass. Local beeswarm plots further reveal monotone effects of σ and λ_0, and the skewed directional effect of ρ, reflecting Wrong–Way Risk. These patterns align with the T=1 setting: λ_0 governs the leading level of integrated intensity over short horizons; mean reversion parameters (k, θ) have limited time to act, and the calibrated band for v restricts intensity dispersion. Correlation ρ remains impactful because even modest intensity randomness interacts with a volatile exposure (through σ) to raise covariation under $\rho > 0$.

Notably, the 1-layer surrogate reproduces both the global and local SHAP structures of the 3-layer model, demonstrating that interpretability is preserved under reduced architectural depth.

Implications. Two conclusions follow. First, calibration and stress testing should prioritize σ, λ_0, and ρ, as these parameters jointly govern both predictive accuracy and economic significance; in particular, careful modeling of correlation is essential, since WWR remains economically material even when v is modest. Second, the 1-layer network achieves explanatory structure and predictive performance that are effectively indistinguishable from

the deeper alternative, while being substantially more efficient in inference. It therefore emerges as an effective, high-throughput, and interpretable baseline for short-term CVA, illustrating that architectural parsimony is compatible with both accuracy and transparency in this regime.

Conclusion

This thesis developed an end-to-end deep learning framework to accelerate Credit Valuation Adjustment (CVA) computations under Wrong–Way Risk (WWR), based on a reduced-form framework in which the exposure follows a geometric Brownian motion and the default intensity evolves according to a Cox–Ingersoll–Ross (CIR) process, both defined under the risk-neutral measure $\mathbb{Q}$ and driven by correlated Brownian motions. Within this stochastic environment, a comprehensive dataset of 61,325 parameter configurations $(\lambda_0, k, \theta, v, \sigma, \rho)$ was constructed, with each tuple associated with a Monte Carlo (MC) estimate of CVA, the corresponding standard error, and—for the uncorrelated case $\rho = 0$—a closed-form analytic benchmark. The generation of this dataset required approximately 18.7 hours of computation time and serves as a baseline for quantifying the speedup achieved by the proposed surrogate models.

Building on this dataset, the study designed, trained, and systematically evaluated fully connected feedforward neural networks (FCNNs) with varying depths, ranging from one to four hidden layers. All models were trained under a unified methodology that included input standardization, early stopping with best-weight restoration, Huber loss fine-tuning, and a cosine annealing schedule for the learning rate. A consistent train/validation/test split was maintained across experiments to ensure comparability and statistical rigour in performance assessment.

The central empirical finding is that a single-hidden-layer network (FCNN–1) offers the best speed–accuracy–robustness trade-off at the one-year maturity horizon. In-distribution, FCNN–1 attains sub-1% test error (about 0.88% of average CVA), while a deeper 3-layer model (FCNN–3) achieves the numerically lowest RMSE (about 0.79%) but with only marginal absolute gain and a $\sim 26\%$ drop in throughput. Under distributional shift, the ranking reverses: systematic stresses in low-density or structurally constrained regions of the parameter space show FCNN–1 preserving smoothness, directionally consistent responses, and economically coherent behaviour for OOD parameters (across all OOD tests with $\rho \neq 0$, the worst FCNN–1 relative error was about -2.73%), whereas FCNN–3 exhibits occasional extrapolation artefacts, with worst-case deviations above 40% in selected OOD slices. On the $\rho = 0$ manifold, both surrogates align closely with the analytic surface and—on average—are closer to the closed form than to the MC labels they were trained on, indicating that the networks distilled the conditional expectation while attenuating residual MC noise and discretization bias.

From a computational perspective, the surrogate overcomes the prohibitive runtime of the Monte Carlo baseline, which required many hours to generate the full dataset. Relative to the reference throughput of approximately 0.91 samples/s under direct simulation, FCNN–1 delivers real-time inference with an empirical acceleration factor on the order of 10^4 (about 3.7×10^4). This substantial gain in efficiency enables the instantaneous evaluation of extensive parameter grids and facilitates interactive, scenario-based risk analysis that would be infeasible with raw Monte Carlo methods.

SHAP diagnostics provide further support for these results. At the one-year maturity horizon, the principal global drivers of CVA are exposure volatility, the initial intensity, and correlation, with an approximate ordering of importance given by $\sigma > \lambda_0 > \rho > \theta \gtrsim$

$v \gtrsim k$. Exposure volatility σ, initial intensity λ_0, and dependence ρ dominate the overall attribution, together accounting for roughly 85% of the total absolute SHAP importance. Higher σ consistently elevates CVA, reflecting the convexity of the exposure profile. The initial intensity λ_0 ranks second, increasing the probability of default over short horizons. The correlation parameter ρ follows closely, exerting a significant upward influence on CVA and capturing the essence of Wrong–Way Risk whereby positive dependence between exposure and intensity amplifies counterparty credit risk. By contrast, the long-run mean θ and the mean-reversion speed k contribute only modestly at short maturities, in line with the limited horizon over which mean-reverting dynamics can act. The intensity volatility v likewise plays a minor role at the one-year horizon, as there is insufficient time for stochastic fluctuations in the default intensity to accumulate. Importantly, FCNN–1 reproduces these ordered and economically interpretable relationships with the same fidelity as the deeper network architecture.

Taken together and consistent with the cumulative evidence across Chapter 1 (data and benchmarks), Chapter 2 (architecture design and tuning), Chapter 3 (performance, efficiency, and robustness under stress), and Chapter 4 (interpretability)—the one-layer surrogate, when equipped with appropriate inductive bias, respects analytic structure and benchmarks, generalizes more reliably under distributional shift, and delivers production-grade acceleration without compromising interpretability.

Building on these findings, a natural direction is to move beyond the one-year horizon and undertake a maturity-wise (term-structure) analysis of CVA under WWR. As maturity increases, parameters governing the level and dynamics of the intensity are expected to gain prominence, so a central question is how the relative roles of θ, k, and v evolve with t and interact with ρ and σ. Two complementary strategies present themselves. One is to train a family of matched one-layer surrogates at selected maturities, enabling controlled cross-sectional comparisons. The other is to learn a single time-conditioned mapping $(\lambda_0, k, \theta, v, \sigma, \rho, t) \mapsto \mathrm{CVA}(t)$ that provides a unified representation of the term structure. The time-conditioned approach allows mild cross-maturity regularity and partial shape constraints (e.g., monotonicity in σ, ρ) to stabilize extrapolation, whereas the per-maturity approach provides a clear basis for attributing how feature salience evolves with maturity.

Beyond the term-structure agenda, two further directions offer extensions of the present work. First, develop a market-consistent calibration and validation framework [11, 4, 24]: calibrate $(\lambda_0, k, \theta, v, \sigma, \rho)$ to liquid CDS and option data [6, 7], refine or regularize the surrogate around the calibrated prior, and include simple uncertainty diagnostics with an auditable Monte Carlo fallback to safeguard accuracy in sparsely sampled regions. Second, extend the modeling of dependence and credit dynamics in a controlled manner—allowing gently time- or state-varying correlation and introducing jump or level-shift components in the intensity (e.g., CIR^{++}, JCIR^{++}) [14, 8, 31]— to test the robustness of WWR asymmetries and to assess how departures from pure diffusion affect CVA sensitivities and exposure–default interactions.

Appendix A

Technical Proofs

A.1 Equivalence of the Compact and Intensity Representations of CVA

Work on a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in[0,T]}, \mathbb{Q})$ supporting the GBM–CIR factors. Let λ be a non-negative, progressively measurable intensity with $\int_0^T \lambda_s\, ds < \infty$ a.s., and let $D(t,u) = e^{-r(u-t)}$ with $r \geq 0$. Assume $\mathbb{E}^{\mathbb{Q}}[(S_T - K)^+] < \infty$. Under the immersion (H-) hypothesis, conditioning on the market filtration $\{\mathcal{F}_t\}$ coincides with conditioning in the filtration enlarged by default [12].

Claim. The compact "no–τ" formula

$$\mathrm{CVA} = \mathrm{LGD}\, \mathbb{E}^{\mathbb{Q}}\Big[\big(1 - e^{-\int_0^T \lambda_s\, ds}\big)\, D(0,T)\, (S_T - K)^+\Big] \tag{A.1.1}$$

is equivalent to the intensity–based representation

$$\mathrm{CVA} = \mathrm{LGD} \int_0^T \mathbb{E}^{\mathbb{Q}}\big[E_t\, W_t\big]\, dt, \qquad E_t := \mathbb{E}^{\mathbb{Q}}\big[D(t,T)\,(S_T - K)^+ \mid \mathcal{F}_t\big], \quad W_t := D(0,t)\,\lambda_t\, e^{-\int_0^t \lambda_u\, du}. \tag{A.1.2}$$

Proof. Let $S(t) := \exp\big(-\int_0^t \lambda_s\, ds\big)$ denote the survival process. Then S is a.s. absolutely continuous and satisfies $S'(t) = -\lambda_t S(t)$. Hence, pathwise,

$$1 - S(T) = \int_0^T \lambda_t S(t)\, dt = \int_0^T \lambda_t \exp\Big(-\int_0^t \lambda_s\, ds\Big)\, dt. \tag{A.1.3}$$

Substituting (A.1.3) into (A.1.1) yields

$$\mathrm{CVA} = \mathrm{LGD}\, \mathbb{E}^{\mathbb{Q}}\left[\int_0^T \lambda_t e^{-\int_0^t \lambda_s ds}\, D(0,T)\, (S_T - K)^+\, dt\right].$$

The integrand is non-negative (all factors are ≥ 0), so Tonelli's theorem permits exchanging expectation and integration:

$$\mathrm{CVA} = \mathrm{LGD} \int_0^T \mathbb{E}^{\mathbb{Q}}\Big[\lambda_t e^{-\int_0^t \lambda_s ds}\, D(0,T)\, (S_T - K)^+\Big]\, dt. \tag{A.1.4}$$

Use the multiplicative property of discounting $D(0,T) = D(0,t)\, D(t,T)$ to rewrite (A.1.4) as

$$\mathrm{CVA} = \mathrm{LGD} \int_0^T \mathbb{E}^{\mathbb{Q}}\left[\underbrace{D(0,t)\,\lambda_t e^{-\int_0^t \lambda_s ds}}_{W_t \text{ is } \mathcal{F}_t\text{-measurable}}\, D(t,T)\, (S_T - K)^+\right] dt.$$

Define $E_t := \mathbb{E}^{\mathbb{Q}}\big[D(t,T)\,(S_T - K)^+ \mid \mathcal{F}_t\big]$. Since W_t is $\mathcal{F}_t$–measurable and integrable, the tower property (law of iterated expectations) gives, for each fixed t,

$$\mathbb{E}^{\mathbb{Q}}\big[W_t\,D(t,T)\,(S_T - K)^+\big] = \mathbb{E}^{\mathbb{Q}}\Big[\,W_t\,\mathbb{E}^{\mathbb{Q}}\big[D(t,T)\,(S_T - K)^+ \mid \mathcal{F}_t\big]\Big] = \mathbb{E}^{\mathbb{Q}}\big[W_t\,E_t\big].$$

Substituting back into the time integral gives (A.1.2). □

Appendix B

Dataset Files

B.1 The CSV File `cva_grid.csv`

As discussed in Chapter 1, results from the simulations are stored in a structured CSV file, `cva_grid.csv`. The final dataset contains 61,325 rows. The fields are summarized below.

We adopt the following conventions:

- All intensities and rates are annualized and expressed in absolute units (e.g. 0.05 denotes 5%);
- The exposure follows a GBM under $\mathbb{Q}$ with volatility parameter σ;
- The default intensity λ follows CIR dynamics with parameters (k, θ, v);
- The correlation ρ couples the Brownian motions driving intensity and exposure.

The Monte Carlo estimator `CVA_MC` is computed under common random numbers for variance control. The column `SE_MC` reports its standard error. When available, `CVA_analytic` provides a closed–form benchmark computed on the $\rho = 0$ slice; for other configurations, it is set to `NA`. The column `RelError_%` gives a diagnostic of Monte Carlo bias and variance relative to the analytic benchmark.

Field	Meaning	Type
λ_0	Initial default intensity	float
k	Mean-reversion speed (CIR)	float
θ	Long-run mean of intensity (CIR)	float
v	Volatility of intensity (CIR)	float
σ	Exposure volatility (GBM)	float
ρ	Correlation between W_1 and W_2	float
`CVA_MC`	Monte Carlo CVA estimate	float
`SE_MC`	MC standard error	float
`CVA_analytic`	Analytic CVA (available on ρ=0 slice)	float/NA
`RelError_%`	$100 \times$ (`CVA_MC` − `CVA_analytic`)/`CVA_analytic`	float

Table B.1: Schema of `cva_grid.csv` (field names, meanings, and types).

Rows are ordered lexicographically by $(\lambda_0, k, \theta, v, \sigma, \rho)$. Numeric values use a dot as decimal separator. The file can be loaded directly using `pandas.read_csv` (Python) and retains the field structure shown in Table B.1.

Illustrative preview. To provide a visual sense of the dataset structure, we include below a truncated preview of the file, consisting of the first and last few rows. These show the numerical encoding of parameters and CVA outputs. Note that `CVA_analytic` and `RelError_%` are only populated for the $\rho = 0$ slice.

idx	λ_0	k	θ	v	σ	ρ	`CVA_MC`	`SE_MC`	`CVA_analytic`	`RelError_%`
0	0.01	0.1	0.01	0.025	0.1	−1.0	0.043264	0.000175	NaN	NaN
1	0.01	0.1	0.01	0.025	0.1	−0.8	0.044811	0.000185	NaN	NaN
2	0.01	0.1	0.01	0.025	0.1	−0.6	0.045711	0.000192	NaN	NaN
3	0.01	0.1	0.01	0.025	0.1	−0.4	0.047094	0.000200	NaN	NaN
4	0.01	0.1	0.01	0.025	0.1	−0.2	0.048731	0.000210	NaN	NaN
5	0.01	0.1	0.01	0.025	0.1	0	0.049893	0.000219	0.049915	-0.04
...										
61320	0.05	1.0	0.05	0.125	0.5	0.2	1.055518	0.006941	NaN	NaN
61321	0.05	1.0	0.05	0.125	0.5	0.4	1.115955	0.007476	NaN	NaN
61322	0.05	1.0	0.05	0.125	0.5	0.6	1.189597	0.008133	NaN	NaN
61323	0.05	1.0	0.05	0.125	0.5	0.8	1.256012	0.008673	NaN	NaN
61324	0.05	1.0	0.05	0.125	0.5	1.0	1.318221	0.009107	NaN	NaN

Table B.2: Preview of `cva_grid.csv` (first and last rows; illustrative).
Note: 61,325 rows × 10 columns (illustrative preview).

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., *Tensorflow: Large-scale machine learning on heterogeneous systems*, 2015.

[2] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, *Towards better understanding of gradient-based attribution methods for deep neural networks*, arXiv preprint arXiv:1711.06104, (2017).

[3] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, Journal of political economy, 81 (1973), pp. 637–654.

[4] D. Brigo and A. Alfonsi, *Credit default swap calibration and derivatives pricing with the ssrd stochastic intensity model*, Finance and stochastics, 9 (2005), pp. 29–42.

[5] D. Brigo and I. Bakkar, *Accurate counterparty risk valuation for energy-commodities swaps*, Energy Risk, (2009).

[6] D. Brigo, A. Capponi, and A. Pallavicini, *Arbitrage-free bilateral counterparty risk valuation under collateralization and application to credit default swaps*, Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics, 24 (2014), pp. 125–146.

[7] D. Brigo and K. Chourdakis, *Counterparty risk for credit default swaps: Impact of spread volatility and default correlation*, International Journal of Theoretical and Applied Finance, 12 (2009), pp. 1007–1026.

[8] D. Brigo and N. El-Bachir, *An exact formula for default swaptions'pricing in the ssrjd stochastic intensity model*, Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics, 20 (2010), pp. 365–382.

[9] D. Brigo, X. Huang, A. Pallavicini, and H. S. de Ocariz Borde, *Interpretability in deep learning for finance: a case study for the heston model*, Risk Sciences, (2025), p. 100030.

[10] D. Brigo and M. Masetti, *Risk neutral pricing of counterparty risk*, in Counterparty Credit Risk Modelling: Risk Management, Pricing and Regulation, M. Pykhtin, ed., Risk Books, London, 2006, ch. 11.

[11] D. Brigo and F. Mercurio, *Interest rate models—Theory and practice: With smile, inflation and credit*, Springer, 2006.

[12] D. Brigo, M. Morini, and A. Pallavicini, *Counterparty credit risk, collateral and funding: with pricing cases for all asset classes*, John Wiley & Sons, 2013.

[13] D. Brigo, M. Morini, M. Tarenghi, T. Bielecki, and F. Patras, *Credit calibration with structural models and equity return swap valuation under counterparty*

risk, Credit risk frontiers: Subprime crisis, pricing and hedging, CVA, MBS, ratings, and liquidity, (2011), pp. 457–484.

[14] D. Brigo and A. Pallavicini, *Counterparty risk pricing under correlation between default and interest rates*, in Numerical methods for finance, Chapman and Hall/CRC, 2007, pp. 63–82.

[15] ——, *Counterparty risk and contingent cds valuation under correlation between interest-rates and default*, ssrn working paper, Fitch Solutions; Imperial College London; Banca Leonardo, 2008.

[16] D. Brigo and F. Vrins, *Disentangling wrong-way risk: pricing credit valuation adjustment via change of measures*, European Journal of Operational Research, 269 (2018), pp. 1154–1164.

[17] F. Chollet et al., *Keras*. `https://keras.io`, 2015. Version 2.0 and onwards available at https://keras.io.

[18] I. Covert, S. Lundberg, and S.-I. Lee, *Explaining by removing: A unified framework for model explanation*, Journal of Machine Learning Research, 22 (2021), pp. 1–90.

[19] J. C. Cox, J. E. Ingersoll, and S. A. Ross, *A theory of the term structure of interest rates*, Econometrica, 53 (1985), pp. 385–407.

[20] S. Crépey and M. Dixon, *Gaussian process regression for derivative portfolio modeling and application to cva computations*, arXiv preprint arXiv:1901.11081, (2019).

[21] D. Duffie and K. J. Singleton, *Credit risk: pricing, measurement, and management*, Princeton university press, 2003.

[22] W. Feller, *Two singular diffusion problems*, Annals of mathematics, 54 (1951), pp. 173–182.

[23] P. Glasserman, *Monte Carlo methods in financial engineering*, vol. 53, Springer, 2004.

[24] J. Gregory, *Counterparty Credit Risk and Credit Value Adjustment: A Continuing Challenge for Global Financial Markets*, John Wiley & Sons, 2012.

[25] B. Horvath, A. Muguruza, and M. Tomas, *Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models*, Quantitative Finance, 21 (2021), pp. 11–27.

[26] P. J. Huber, *Robust estimation of a location parameter*, in Breakthroughs in statistics: Methodology and distribution, Springer, 1992, pp. 492–518.

[27] R. Lord, R. Koekkoek, and D. V. Dijk, *A comparison of biased simulation schemes for stochastic volatility models*, Quantitative Finance, 10 (2010), pp. 177–194.

[28] I. Loshchilov and F. Hutter, *Sgdr: Stochastic gradient descent with warm restarts*, arXiv preprint arXiv:1608.03983, (2016).

[29] M. Ludkovski, *Statistical machine learning for quantitative finance*, Annual Review of Statistics and Its Application, 10 (2023), pp. 271–295.

[30] S. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, in Advances in Neural Information Processing Systems, vol. 30, 2017.

[31] C. Mbaye and F. Vrins, *A subordinated cir intensity model with application to wrong-way risk cva*, International Journal of Theoretical and Applied Finance, 21 (2018), p. 1850045.

[32] C. Molnar, *Interpretable machine learning*, Lulu. com, 2020.

[33] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, et al., *Kerastuner*. `https://github.com/keras-team/keras-tuner`, 2019.

[34] L. S. Shapley, *A value for n-person games*, Contributions to the Theory of Games, 2 (1953), pp. 307–317.

[35] A. Shrikumar, P. Greenside, and A. Kundaje, *Learning important features through propagating activation differences*, in International conference on machine learning, PMlR, 2017, pp. 3145–3153.

[36] B. Yuan, D. Brigo, A. Jacquier, and N. Pede, *Deep learning interpretability for rough volatility*, arXiv preprint arXiv:2411.19317, (2024).