Insurance risk control with reinsurance

Amos Anderson & Indu Sai Atla

Department of Applied Mathematics and Statistics
Stony Brook University

December 4, 2025

Abstract

We study optimal investment and reinsurance for an insurer with jump–diffusion surplus dynamics under exponential utility. A Physics-Informed Neural Network (PINN) is developed to solve the resulting HJB equation, incorporating the nonlocal jump term through Gaussian quadrature. Numerical experiments examine (i) the effect of different claim-size distributions under proportional reinsurance and (ii) the impact of replacing proportional reinsurance with an excess-of-loss (XoL) contract. Heavier-tailed claims produce more conservative investment and greater reinsurance use, while XoL reinsurance yields qualitatively different behaviour: the optimal investment includes mild hedging regions, and the optimal deductible settles around an interior level of roughly five units. The results show how actuarial tail risk and reinsurance structure jointly shape optimal insurer decisions and demonstrate the effectiveness of PINNs for solving stochastic control problems with jumps.

Contents

1 Introduction

Insurance companies operate in environments characterized by both continuous fluctuations in financial markets and discontinuous jump risks arising from insurance claims. Optimal risk management therefore requires the joint selection of (i) an investment strategy in financial assets and (ii) a reinsurance policy that transfers part of the claim risk to another firm. When the insurer seeks to maximize the expected utility of terminal surplus under a jump-diffusion surplus model, the resulting control problem is governed by a nonlinear Hamilton-Jacobi-Bellman (HJB) equation containing both local (differential) and nonlocal (integral) operators.

Classical numerical methods struggle with this problem because the jump integral induces nonlocality, the domain is two-dimensional (t, x), and the control variables enter the HJB operator in nonlinear form. Physics-Informed Neural Networks (PINNs), introduced by Raissi et al. (2019) [1], provide a mesh-free and differentiable framework for solving high-dimensional PDEs and stochastic control problems.
This project applies the PINN methodology to learn simultaneously the value function $V(t, x)$ and the optimal controls $(\pi(t, x), \theta(t, x))$ or $(\pi(t, x), M(t, x))$ for proportional and excess-of-loss reinsurance structures, respectively.

The contributions of this project are threefold:

1. We formulate the insurer's surplus dynamics as a jump-diffusion model and derive the associated nonlinear HJB equation.
2. We design a PINN architecture capable of handling the nonlocal jump term through Gaussian–Legendre quadrature and train the network to satisfy the HJB equation, terminal condition, and boundary behaviour.
3. We conduct numerical experiments comparing (i) three claim-size distributions under proportional reinsurance and (ii) the impact of proportional and excess-of-loss reinsurance structures on optimal investment and reinsurance policies.

The results demonstrate how actuarial assumptions and reinsurance structures affect the insurer's optimal investment policy, optimal reinsurance decision, and overall certainty-equivalent utility. The PINN framework captures these behaviours reliably and provides smooth, interpretable policy surfaces.

The remainder of this report is structured as follows. Section 2 introduces the surplus model and control objectives. Section 3 presents the HJB equation. Section 4 details the PINN architecture and training procedure. Section 5 reports numerical experiments and comparative analyses. Section 6 concludes.

2 Model Setup

We model an insurance company that invests in financial markets while being exposed to random claim arrivals. Let X_t denote the insurer's surplus at time t, defined as the difference

between assets and liabilities. The company invests in a risk-free asset with rate r and a risky asset with drift μ and volatility σ. The control $\pi(t,x)$ represents the amount invested in the risky asset when the current surplus is x.

2.1 Surplus Dynamics

Between claim arrivals, the insurer earns deterministic interest on its surplus and bears the gains and losses from the risky investment. This contributes

$$\big(rX_t + \pi_t(\mu - r)\big)dt + \sigma\pi_t dW_t.$$

Claims arrive according to a Poisson process N_t with intensity λ, and each claim has an i.i.d. severity Z_i. Without reinsurance, the cumulative claim process is

$$C_t = \sum_{i=1}^{N_t} Z_i.$$

Thus, in the absence of reinsurance, the surplus evolves as

$$dX_t = rX_t dt + \pi_t(\mu - r)dt + \sigma\pi_t dW_t - dC_t. \tag{1}$$

2.2 Retained Losses Under Reinsurance

Reinsurance modifies the loss the insurer must pay upon a claim. Let $L(Z, \rho_t)$ denote the retained loss, where ρ_t is the reinsurance control. The reinsurance control ρ is interpreted as

$$\rho(t,x) = \begin{cases} \theta(t,x), & \text{proportional reinsurance control} \\ M(t,x), & \text{excess-of-loss reinsurance control} \end{cases}$$

Two forms of $L(Z, \rho_t)$ are examined in later sections:

- **Proportional reinsurance:** The insurer retains $(1 - \theta_t)Z$, where $\theta_t \in [0, 1]$ is the ceded proportion.
- **Excess-of-Loss reinsurance:** The insurer retains $\min(Z, M_t)$, where M_t is a deductible.

In particular, $L(Z, \rho_t)$ denotes the retained loss under the chosen reinsurance contract and is defined as

$$L(Z, \rho_t) := \begin{cases} (1 - \theta)Z, & \text{retained loss from proportional reinsurance} \\ \min(M, Z), & \text{retained loss from excess-of-loss reinsurance} \end{cases}$$

Let C_t^{retained} denote the cumulative retained-claim process:

$$C_t^{\text{retained}} = \sum_{i=1}^{N_t} L(Z_i, \rho_t).$$

The general surplus SDE under reinsurance is therefore

$$dX_t = rX_t dt + \pi_t(\mu - r)dt + \sigma\pi_t dW_t - dC_t^{\text{retained}}. \tag{2}$$

This is the model used throughout the project, with the particular form of $L(Z, \rho_t)$ depending on the reinsurance structure considered.

2.3 Terminal Surplus

Integrating (2) over $[0, T]$ yields the terminal surplus

$$X_T = X_0 + \int_0^T \big(rX_s + \pi_s(\mu - r)\big)ds + \int_0^T \sigma\pi_s dW_s - \sum_{i=1}^{N_T} L(Z_i, \rho_{T_i}),$$

where T_i are the claim arrival times. The insurer ultimately cares only about the terminal surplus X_T as the best achievable utility at time T is the surplus at time T.

3 Control Objective

The insurer dynamically controls two decision variables:

- the amount $\pi(t, x)$ invested in the risky asset,
- the reinsurance decision $\rho(t, x)$, which represents either the ceded proportion $\theta(t, x)$ (proportional reinsurance) or the deductible $M(t, x)$ (excess-of-loss reinsurance).

Given the surplus dynamics in (2), the insurer seeks to maximize the expected utility of terminal wealth at time T. The value function is defined by

$$V(t, x) = \sup_{\pi,\rho} \mathbb{E}[\, U(X_T^{\pi,\rho}) \mid X_t = x \,], \tag{3}$$

where the supremum is taken over all admissible controls. The optimal strategies (π^*, ρ^*) are those that attain the above value.
In the numerical experiments that follow,

$$\rho(t, x) = \theta(t, x) \quad \text{(Experiments 1 and 2A)}, \qquad \rho(t, x) = M(t, x) \quad \text{(Experiment 2B)}.$$

4 Utility Function

We adopt the exponential utility

$$U(x) = -e^{-x} \tag{4}$$

which exhibits constant absolute risk aversion (CARA).

We chose the exponential utility function because it ensures concavity, fits insurance risk theory and is easy to differentiate during the training process in the neural network.

The insurer's objective is therefore

$$\max_{\pi,\rho} \mathbb{E}\big[-e^{-X_T}\big], \tag{5}$$

which corresponds to maximizing the certainty-equivalent terminal wealth under jump-diffusion risk.

5 The Hamilton-Jacobi-Bellman (HJB) Equation

Given the surplus dynamics in (2), and by the dynamic programming principle, the value function $V(t,x)$ satisfies the Hamilton-Jacobi-Bellman (HJB) equation

$$0 = \sup_{\pi,\rho} \Big\{ V_t + (rx + \pi(\mu - r))V_x + \frac{1}{2}\sigma^2\pi^2 V_{xx} + \lambda\big(\mathbb{E}[V(t, x - L(Z,\rho_t))] - V(t,x)\big)\Big\}, \tag{6}$$

with terminal condition

$$V(T,x) = U(x)$$

Interpretation of the Jump Term

The jump component of the HJB equation,

$$\lambda\big(\mathbb{E}[V(t, x - L(Z,\rho_t))] - V(t,x)\big),$$

captures the effect of claim arrivals on the surplus process. Here Z is a continuous claim-size random variable with density $f_Z(z)$ and $L(Z,\rho_t)$ denotes the retained loss under the chosen reinsurance contract.

Since claims arrive according to a Poisson process with intensity λ, the expected change in the value function over a small interval dt is given by

$$\lambda\, dt \times \big(\mathbb{E}[V(t, x - L(Z,\rho_t))] - V(t,x)\big).$$

For continuous claim distributions, the expectation can be written explicitly as

$$\mathbb{E}[V(t, x - L(Z,\rho_t))] = \int_0^\infty V(t, x - L(z,\rho_t))\, f_Z(z)\, dz. \tag{7}$$

Thus, the jump term introduces a nonlocal operator in the HJB equation, whose form depends entirely on the chosen reinsurance structure. This nonlocality is the primary numerical challenge addressed by the PINN methodology.

6 PINN Implementation

To numerically solve the HJB equation under both proportional and excess-of-loss reinsurance, we implement a Physics-Informed Neural Network (PINN) using PyTorch. The PINN jointly approximates the value function $V(t,x)$ and the optimal controls $\pi(t,x)$ (investment) and $\rho(t,x)$ (reinsurance).

6.1 Network Architecture

Figure 1 depicts the forward propagation of the network. The model consists of:

- a fully connected feed-forward neural network (MLP),
- an automatic differentiation module computing V_t, V_x, and V_{xx},
- a quadrature module approximating the nonlocal jump expectation,
- a composite loss enforcing the HJB equation, terminal condition, boundary conditions, and control regularization.

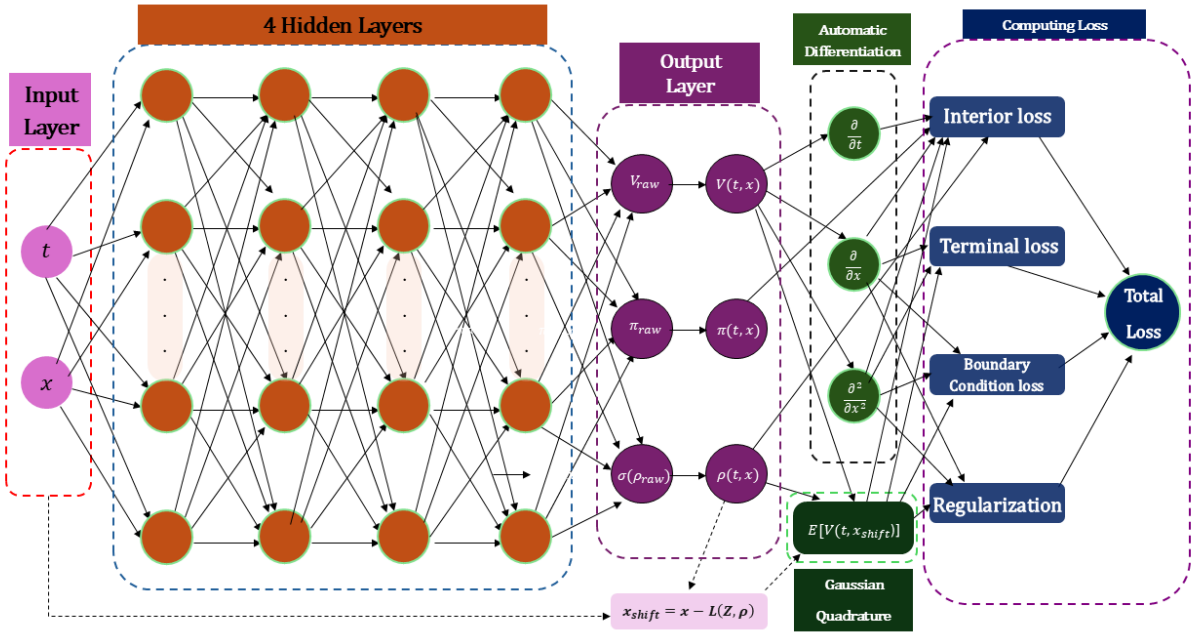


Figure 1: Forward Propagation of a Fully Interconnected feed-forward MLP

Figure 1: Forward propagation of the PINN

6.2 Inputs and Outputs

Inputs

The HJB equation is defined on the domain

$$t \in [0, T], \qquad x \in [x_{\min}, x_{\max}],$$

and the network input is simply the pair (t, x).

Outputs

The network outputs three quantities:

$$\Theta_{NN}(t,x) \;\mapsto\; \big(V_\Theta(t,x),\, \pi_\Theta(t,x),\, \rho_\Theta(t,x)\big),$$

where:

- $V_\Theta(t,x)$ approximates the value function,
- $\pi_\Theta(t,x)$ is the investment control,
- $\rho_\Theta(t,x)$ is the reinsurance control.

These raw outputs are post-processed to enforce control constraints:

$$V_\Theta(t,x) = V_{\text{raw}}(t,x), \quad \pi_\Theta(t,x) = \pi_{\text{raw}}(t,x), \quad \rho_\Theta(t,x) = \sigma(\rho_{\text{raw}}(t,x))$$

The output layer contains three neurons producing

$$(V_{\text{raw}}(t,x), \pi_{\text{raw}}(t,x), \rho_{\text{raw}}(t,x))$$

For proportional reinsurance

$$\theta(t,x) = \sigma(\theta_{\text{raw}}(t,x)) \in (0,1)$$

Likewise, for XoL reinsurance,

$$M(t,x) = M_{\max}\, \sigma(M_{\text{raw}}(t,x))$$

where $\sigma(\cdot)$ is the sigmoid function.

6.3 Hidden Layers

The inputs are passed through a fully connected MLP with $L = 4$ hidden layers, each containing 64 neurons and using the smooth tanh activation:

$$z^{[l]} = \tanh\big(W^{[l]} z^{[l-1]} + b^{[l]}\big), \qquad l = 1, \ldots, L.$$

The final layer produces $(V_{\text{raw}}, \pi_{\text{raw}}, \rho_{\text{raw}})$, which are transformed into admissible outputs as described above.

6.4 The HJB Residual

From the HJB equation in 6, the optimal $\pi(t,x)$ and $\theta(t,x)$ satisfy:

$$0 = V_t + (rx + \pi(\mu - r))V_x + \frac{1}{2}\sigma^2\pi^2 V_{xx} + \lambda\big(\mathbb{E}[V(t, x - L(Z,\rho))] - V(t,x)\big).$$

At a collocation point (t_i, x_i), the PINN residual is therefore:

$$\mathcal{R}(t_i, x_i; \Theta_{NN}) = V_t(t_i,x_i) + \big(rx_i + \pi_i(\mu - r)\big)V_x(t_i,x_i) + \frac{1}{2}\sigma^2\pi_i^2 V_{xx}(t_i,x_i) + \lambda\big(\widehat{\mathbb{E}}_i - V_i\big),$$

where the expectation is approximated via quadrature (next subsection).

6.5 Quadrature Approximation of The Jump Term

For each collocation point and its control ρ_i, we approximate:

$$\mathbb{E}[V(t_i, x_i - L(Z, \rho_i))] = \int_{S_Z} V(t_i, x_i - L(z, \rho_i))\, f_Z(z)\, dz.$$

We truncate the claim domain to $[a, b]$ (for exponential claims, typically $[0, 10/\lambda]$), map it to the reference interval $[-1, 1]$, and apply K-point Gaussian quadrature:

$$\int_a^b h(z)\, dz \approx \sum_{k=1}^{K} \tilde{w}_k\, h(z_k).$$

Thus, the expectation becomes:

$$\widehat{\mathbb{E}}_i = \sum_{k=1}^{K} \tilde{w}_k\, V_\Theta(t_i,\ x_i - L(z_k, \rho_i))\, f_Z(z_k).$$

6.6 Loss Function

The total loss consists of four components:

$$\mathcal{L} = w_{PDE}\mathcal{L}_{PDE} + w_T\mathcal{L}_{\text{terminal}} + w_{BC}\mathcal{L}_{BC} + \mathcal{L}_{\text{reg}}.$$

PDE Residual Loss

$$\mathcal{L}_{PDE} = \frac{1}{N_{int}} \sum_{i=1}^{N_{int}} \mathcal{R}(t_i, x_i; \Theta_{NN})^2.$$

Terminal Loss

$$\mathcal{L}_{\text{terminal}} = \frac{1}{N_T} \sum_{j=1}^{N_T} \big(V_\Theta(T, x_j) - U(x_j)\big)^2.$$

Boundary Loss

We enforce soft Dirichlet conditions:

$$V(t, x_{\min}) = U(x_{\min}), \qquad V(t, x_{\max}) = U(x_{\max}),$$

leading to:

$$\mathcal{L}_{BC} = \frac{1}{N_{BC}} \sum_{k=1}^{N_{BC}} \Big[(V(t_k, x_{\min}) - U(x_{\min}))^2 + (V(t_k, x_{\max}) - U(x_{\max}))^2 \Big].$$

Regularization of Controls

Since controls are learned directly (no pointwise maximization), we penalize excessively large values:

$$\mathcal{L}_{\text{reg}} = \eta_\pi \frac{1}{N_{int}} \sum_i \pi(t_i, x_i)^2 + \eta_\rho \frac{1}{N_{int}} \sum_i \big(\rho(t_i, x_i) - \rho_0\big)^2.$$

We use:

- $\rho_0 = 0.5$ for proportional reinsurance. This keeps θ away from boundaries early in training.
- $\rho_0 = M_{\max}/2$ for XoL to encourages exploration of the interior range.

This stabilizes training [2] and is standard in PINN-based stochastic control [3].

7 PINN Training: Backward Propagation

Backward propagation computes the gradients of the total loss with respect to all network parameters so that the optimizer (Adam or L–BFGS) can update the weights. Because the HJB loss involves derivatives of V, nonlinear dependence on the controls, and a nonlocal jump expectation, the computational graph is substantially richer than in standard supervised learning.

7.1 Computational Graph for the HJB PINN

The complete differentiable graph consists of:

1. the neural network parameters $\{W^{[l]}, b^{[l]}\}$,
2. the network outputs $(V(t,x), \pi(t,x), \rho(t,x))$,
3. the post-processed controls
4. the automatic differentiation nodes producing V_t, V_x, V_{xx},
5. the quadrature module computing $\widehat{\mathbb{E}}[V(t, x - L(Z,\rho))]$,
6. the HJB residual $\mathcal{R}(t,x)$,
7. the loss components: PDE, terminal, boundary, and regularization.

Every operation in this chain is differentiable, so the total loss is a smooth function of all network parameters.

7.2 Gradient of the Total Loss

For each layer l, backward propagation computes

$$\frac{\partial \mathcal{L}}{\partial W^{[l]}}, \qquad \frac{\partial \mathcal{L}}{\partial b^{[l]}},$$

where the dependence of $\mathcal{L}$ flows through:

- the outputs (V, π, ρ),
- the derivatives (V_t, V_x, V_{xx}),
- the shifted evaluation points $x - L(z_k, \rho)$,
- the quadrature approximation of the jump expectation.

Thus, the gradient flow incorporates every mathematical component appearing in the HJB operator.

7.3 Backpropagation Through Automatic Differentiation

Automatic differentiation (AD) or `autograd` in using PyTorch, computes the spatial and temporal derivatives:

$$V_t(t, x), \qquad V_x(t, x), \qquad V_{xx}(t, x),$$

by constructing a subgraph of elementary operations. During the backward pass, AD evaluates

$$\frac{\partial V_t}{\partial W^{[l]}}, \qquad \frac{\partial V_x}{\partial W^{[l]}}, \qquad \frac{\partial V_{xx}}{\partial W^{[l]}},$$

which feed directly into

$$\frac{\partial \mathcal{R}}{\partial W^{[l]}}.$$

This allows the HJB derivative terms to exert influence on all network weights even deep inside the architecture.

7.4 Backpropagation Through the Nonlocal Jump Term

The jump expectation is approximated by Gaussian quadrature:

$$\widehat{\mathbb{E}}[V] = \sum_{k=1}^{K} \tilde{w}_k \, V\big(t, x - L(z_k, \rho)\big) \, f_Z(z_k),$$

where

$$L(z, \rho) = \begin{cases} (1-\theta)z, & \text{proportional reinsurance,} \\ \min(z, M), & \text{excess-of-loss reinsurance.} \end{cases}$$

During backpropagation:

- gradients flow from the loss $\mathcal{L}$ into each term $V(t, x - L(z_k, \rho))$,
- each shifted value is a forward call of the *same network*, so all such paths contribute gradients to every parameter,
- gradients also flow through the shift:

$$x_k = x - L(z_k, \rho),$$

which means

$$\frac{\partial x_k}{\partial \rho} \neq 0.$$

Thus, the network learns *both* the value function shape and the reinsurance control that best reduces the HJB residual.

7.5 Backpropagation Through the Control Outputs

The controls appear in three distinct parts of the HJB:

- the drift term: $(rx + \pi(\mu - r))V_x$,
- the diffusion term: $\frac{1}{2}\sigma^2\pi^2 V_{xx}$,
- the jump shift: $x - L(z_k, \rho)$.

Therefore,

$$\frac{\partial \mathcal{L}}{\partial \pi(t,x)} \neq 0, \qquad \frac{\partial \mathcal{L}}{\partial \rho(t,x)} \neq 0.$$

These propagate through the output activations. For proportional reinsurance:

$$\theta = \sigma(\rho_{\text{raw}}), \qquad \frac{\partial \theta}{\partial \rho_{\text{raw}}} = \sigma(\rho_{\text{raw}})\big(1 - \sigma(\rho_{\text{raw}})\big),$$

while for XoL,

$$M = M_{\max}\sigma(\rho_{\text{raw}}), \qquad \frac{\partial M}{\partial \rho_{\text{raw}}} = M_{\max}\sigma(\rho_{\text{raw}})\big(1 - \sigma(\rho_{\text{raw}})\big).$$

This is where the regularization term plays a stabilizing role, preventing the nonlinear diffusion term and jump-shift derivatives from producing exploding gradients early in training.

7.6 Parameter Update

After computing all gradients, the optimizer (Adam in our implementation) updates each layer:

$$W^{[l]} \leftarrow W^{[l]} - \eta \frac{\partial \mathcal{L}}{\partial W^{[l]}}, \qquad b^{[l]} \leftarrow b^{[l]} - \eta \frac{\partial \mathcal{L}}{\partial b^{[l]}},$$

with learning rate η.
This iterative process trains the network to simultaneously satisfy:

- the nonlinear HJB PDE,
- the terminal utility condition,
- boundary conditions,
- admissibility of the controls.

The resulting (V^*, π^*, ρ^*) represent the numerically learned optimal value function and optimal joint investment–reinsurance strategies.

8 Numerical Experiments

We organize our numerical study into two main experiment groups.

- Experiment 1 investigates the impact of the claim size distribution on optimal investment and proportional reinsurance strategies, with all other modeling choices fixed.
- Experiment 2 keeps the claim distribution fixed (exponential) and compares different reinsurance structures (proportional vs excess-of-loss).

This design isolates the effect of each modeling component and yields interpretable comparative statics.

8.1 Experiment 1: Impact of Claim Distribution On Optimal Investment-Reinsurance Policies

8.1.1 Problem Setup

The goal of this experiment is to investigate how different actuarial assumptions about the claim size distribution Z affect the optimal investment policy $\pi(t, x)$, the optimal proportional reinsurance strategy $\theta(t, x)$, and the value function $V(t, x)$ derived from the HJB equation. We hold all other components of the model fixed, including market parameters, utility structure, and reinsurance type.

We consider three claim distributions namely; exponential, pareto and lognormal, each considered as a sub-experiment.

Model Setup.

The insurer's surplus follows the jump-diffusion SDE:

$$dX_t = rX_t dt + \pi_t(\mu - r)dt - (1 - \theta)dC_t + \sigma\pi_t dW_t. \tag{8}$$

Let the risk aversion level be $\gamma = 1$. We adopt exponential utility

$$U(x) = -e^{-x},$$

and the terminal condition is $V(T,x) = U(x)$.

We use **proportional reinsurance** throughout experiment 1. The retained loss is

$$L(Z,\theta_t) = (1-\theta_t)Z,$$

From 6, this gives the proportional HJB

$$0 = \sup_{\pi,\theta}\Big\{V_t + (rx + \pi(\mu - r))V_x + \frac{1}{2}\sigma^2\pi^2 V_{xx} + \lambda\big(\mathbb{E}[V(t,x-(1-\theta)Z)] - V(t,x)\big)\Big\} \quad (9)$$

We seek to solve the proportional HJB to determine optimal $\pi(t,x)$ and $\theta(t,x)$ using the PINN algorithm.

Common Parameter Values.

For all sub-experiments in Experiment 1, we fix

$$r = 0.03, \quad \mu = 0.08, \quad \sigma = 0.20, \quad \lambda = 1.0, \quad T = 1,$$

and define the surplus domain as $x \in [0, 10]$.

Sub-Experiments.

- **Experiment 1A: Exponential Claims.**

$$Z \sim \text{Exponential}(\beta), \qquad f_Z(z) = \beta e^{-\beta z}, \quad \beta = 1.$$

- **Experiment 1B: Pareto Claims.**

$$Z \sim \text{Pareto}(\alpha, k), \qquad f_Z(z) = \frac{\alpha k^\alpha}{(z+k)^{\alpha+1}}, \quad \alpha = 3,\ k = 1.$$

- **Experiment 1C: Lognormal Claims.**

$$Z \sim \text{Lognormal}(\mu_Z, \sigma_Z^2), \qquad f_Z(z) = \frac{1}{z\sigma_Z\sqrt{2\pi}}\exp\left(-\frac{(\ln z - \mu_Z)^2}{2\sigma_Z^2}\right),$$

  with $\mu_Z = 0$ and $\sigma_Z = 0.5$.

Quadrature Approximation.

For all three claim distributions we approximate

$$\mathbb{E}[V(t,x-(1-\theta)Z)] = \int_0^\infty V(t,x-(1-\theta)z) f_Z(z)\,dz$$

by Gaussian quadrature:

$$\widehat{\mathbb{E}}[V] = \sum_{k=1}^{K} \tilde{w}_k\, V(t,x-(1-\theta)z_k)\, f_Z(z_k).$$

Experimental Procedure.

For each distribution we train the PINN, record the losses, and generate

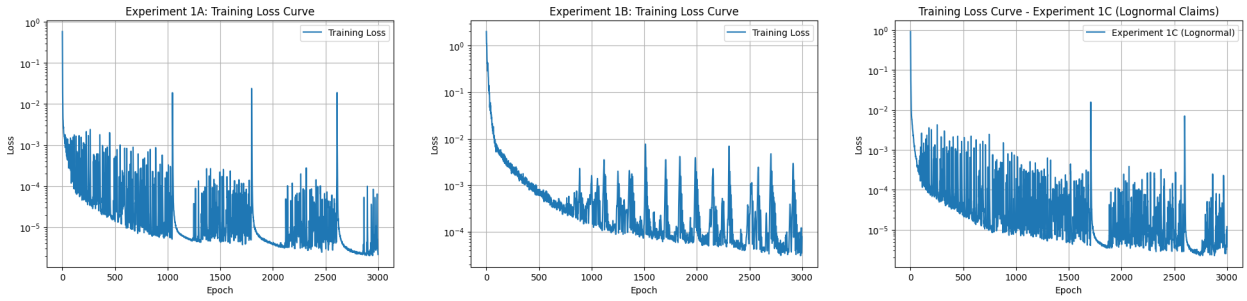
- Convergence plots to track the PINN loss.
- 3D plots of $V(t,x)$, $\pi(t,x)$ and $\theta(t,x)$ respectively.
- For a chosen surplus x and time t, we explore the optimal $V(t,x)$, $\pi(t,x)$ and $\theta(t,x)$.

8.1.2 PINN Architecture and Training Setup

The PINN consists of four hidden layers with 64 neurons each and `tanh` activation. Inputs are (t,x) and outputs are (V,π,θ). The jump expectation uses Gauss-Legendre quadrature with 32 nodes. To stabilize training, the PDE residual is evaluated in mini-batches of 500 collocation points, and small regularization terms keep the control outputs within reasonable ranges during early training. Optimization uses Adam with learning rate 10^{-3}.

8.1.3 Training Convergence Across Claim Distributions

Figure 2 shows the total loss for each model over 3000 epochs.



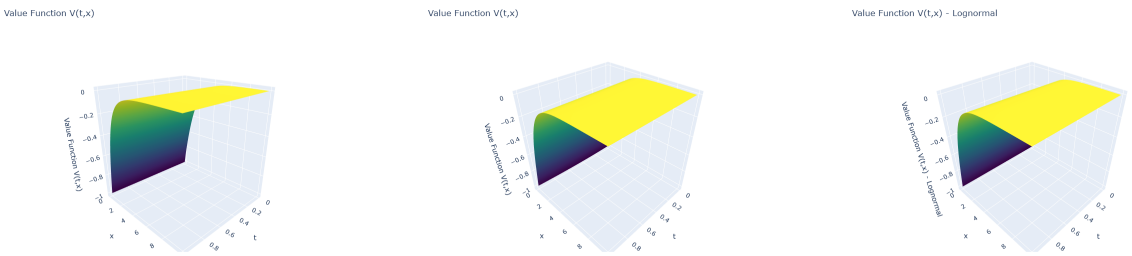
(a) Exponential Claims (b) Pareto Claims (c) Lognormal Claims

Figure 2: Training loss curves for the three claim distributions in Experiment 1.

All three models demonstrate stable convergence, with the loss decreasing by several orders of magnitude during training. The Pareto model shows more pronounced oscillations, reflecting the numerical difficulty of learning solutions under heavy-tailed jump distributions. Nonetheless, the loss curves indicate that the PINN successfully approximates the HJB solution across the three claim-size specifications.

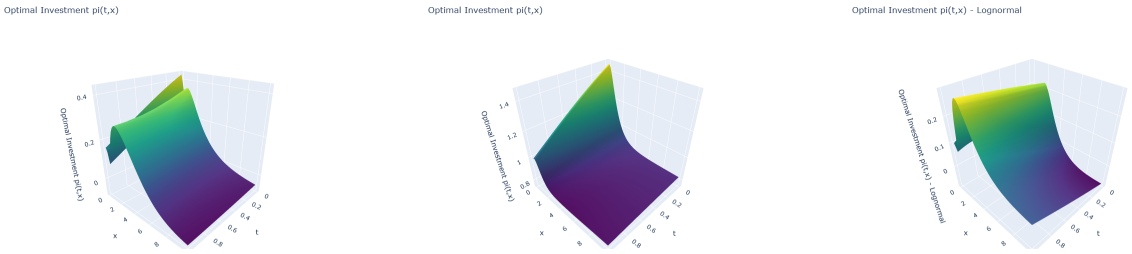
8.1.4 Learned Value Function and Optimal Policy Surfaces

Next, we present the learned value functions $V(t,x)$ and the corresponding optimal investment $\pi(t,x)$ and proportional reinsurance strategies $\theta(t,x)$ obtained from the PINN solutions of the HJB equation under the three different claim-size distributions considered in Experiment 1. All surfaces are plotted over the state domain $(t,x) \in [0,1] \times [0,10]$ to facilitate direct visual comparison across models.



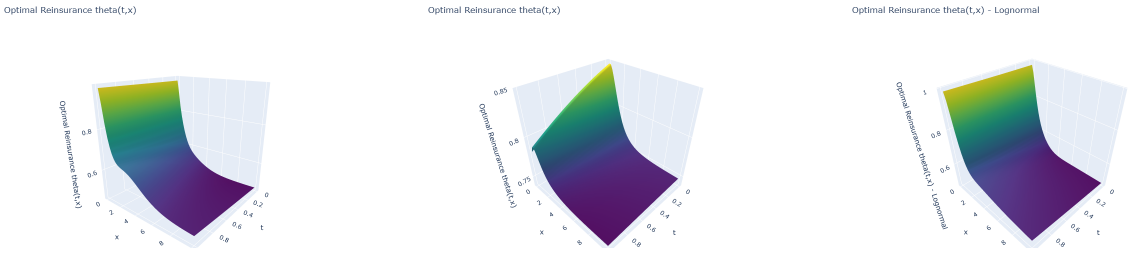
(a) Exponential (b) Pareto (c) Lognormal

Figure 3: Learned value function $V(t, x)$ for the three claim-size distributions.



(a) Exponential (b) Pareto (c) Lognormal

Figure 4: Optimal investment policy $\pi(t, x)$ under different claim-size distributions.



(a) Exponential (b) Pareto (c) Lognormal

Figure 5: Optimal reinsurance retention $\theta(t, x)$ across the three claim-size models.

The three claim-size distributions introduce fundamentally different jump-risk profiles into the HJB equation. The exponential model represents a light-tailed environment, the lognormal model introduces moderately heavy but still well-behaved tails, and the Pareto case generates genuinely heavy-tailed jumps with substantial probability of catastrophic losses. These statistical differences manifest clearly in the learned value functions and optimal controls.

Value Function.

Across all distributions the value function $V(t, x)$ is increasing in x and decreasing in t, reflecting the structure of the exponential utility and the terminal condition $V(T, x) = U(x)$. The key differences lie in curvature:

- **Exponential (light tail):** The surface is smooth with relatively mild curvature.
- **Lognormal (moderately heavy tail):** Curvature increases for small x, reflecting greater downside risk, but remains well-behaved because higher moments exist.
- **Pareto (heavy tail):** The value surface shows the steepest curvature in x, consistent with the insurer being highly sensitive to states where catastrophic losses dominate jump risk.

The worsening of V (more negative values) from exponential $\rightarrow$ lognormal $\rightarrow$ Pareto is fully consistent with the increasing tail risk in the claim distribution.

Investment Policy $\pi(t, x)$.

All three models exhibit the economically expected decline of $\pi(t, x)$ with increasing surplus x, reflecting diminishing marginal utility. However, the distributions differ substantially in the shape and magnitude of the surface:

- **Exponential:** Produces a stable, monotone, moderate investment policy.
- **Lognormal:** Slightly more conservative across most of the domain, consistent with increased downside exposure.
- **Pareto:** The global shape of the surface is *more conservative* for large x, but a localized spike occurs around $(t, x) \approx (0.5, 5)$. This spike does *not* represent an economic phenomenon but rather a numerical artifact: the heavy-tailed jump term sharply increases curvature in V, making the residual minimization highly stiff in that region. The PINN occasionally responds by increasing π locally to rebalance the PDE terms, but the global investment surface remains qualitatively conservative.

Thus, Pareto claims lead to overall lower investment levels, except for isolated local oscillations caused by PDE stiffness.

Reinsurance Policy $\theta(t, x)$.

The proportional reinsurance policy behaves consistently with classical insurance theory:

- **Exponential:** Moderate reinsurance demand.
- **Lognormal:** Nearly identical to the exponential case except for slightly higher retention near $x = 0$.
- **Pareto:** Significantly higher reinsurance across almost the entire domain.

Under heavy-tailed claims, reinsurance becomes the primary risk-transfer mechanism, and the optimal policy shifts visibly upward to mitigate catastrophic jump exposure.

8.1.5 Pointwise Comparison of Value Function and Optimal Controls

Next, we compare the three models at a chosen representative state

$$(t_0, x_0) = (0.5, 5)$$

corresponding to mid-horizon and mid-surplus.

The PINN outputs the value function optimal results of $V(t,x)$, $\pi(t,x)$ and $\theta(t,x)$.

These values are summarized in Table 1, enabling direct comparison across the three claim-size distributions.

Model	$\mathbf{V(t_0, x_0)}$	$\pi(\mathbf{t_0, x_0})$	$\theta(\mathbf{t_0, x_0})$
Exponential	-0.00954	0.0490	0.513
Pareto	-0.01003	0.8065	0.748
Lognormal	-0.00848	0.0264	0.510

Table 1: Pointwise comparison of value function and optimal controls at $(t_0, x_0) = (0.5, 5)$ for the three claim-size distributions.

At the representative state $(t_0, x_0) = (0.5, 5)$, the PINN yields the values shown in Table 1. The ordering of the value functions

$$V_{\text{Pareto}} < V_{\text{Exponential}} < V_{\text{Lognormal}}$$

matches the ordering of tail risk. The reinsurance levels follow the same ranking.

The investment result for the Pareto case exhibits a local increase at this specific point, but the 3D policy surface confirms that the overall strategy is more conservative when x is large. This localized deviation is a training artifact arising from the stiffness of the HJB operator under heavy-tailed distributions and does not contradict the global financial intuition.

8.2 Experiment 2: Impact of Reinsurance Structure on Optimal Policies

The purpose of this experiment is to investigate how the *form* of reinsurance policy influences the insurer's optimal investment policy $\pi(t,x)$, the optimal reinsurance decision, and the value function $V(t,x)$ arising from the HJB equation.

In Experiment 1 we studied proportional reinsurance and established how the claim-size distribution affects the optimal controls. However, insurers frequently employ other forms of risk transfer, and in particular, *excess-of-loss* (XoL) arrangements are among the most common in practice.

To isolate the effect of the reinsurance structure, we now fix the claim-size distribution to be exponential and compare two settings:

1. proportional reinsurance (Experiment 2A), and
2. excess-of-loss reinsurance (Experiment 2B).

The central question is:

> **How do optimal investment and reinsurance decisions change when the insurer switches from proportional to excess-of-loss reinsurance, holding all financial and actuarial parameters fixed?**

8.2.1 Experiment 2A: Proportional Reinsurance (Baseline)

Model Setup

Experiment 2A coincides exactly with the proportional reinsurance model studied in Experiment 1A. Thus, the surplus dynamics, exponential utility specification, proportional HJB equation, and the full PINN architecture (network structure, quadrature treatment, training procedure, and loss design) are the same as those already introduced in the set-up for Experiment 1A.

The claim-size distribution is again exponential,

$$Z \sim \text{Exponential}(1), \qquad f_Z(z) = e^{-z}.$$

Therefore, Experiment 2A serves as the *benchmark* proportional model against which the excess-of-loss results in Experiment 2B will be compared.

Results

Experiment 1A provides the proportional reinsurance benchmark. At the representative state $(t_0, x_0) = (0.5, 5)$, the learned optimal policies were:

$$V(t_0, x_0) = -0.00954, \qquad \pi(t_0, x_0) = 0.04899, \qquad \theta(t_0, x_0) = 0.51267.$$

The optimal investment level is moderate and increases with surplus, while the optimal reinsurance proportion $\theta(t, x)$ tends to be near 0.5, reflecting a balance between retaining premium income and reducing claim volatility.

These results serve as the baseline against which the excess-of-loss policies will be compared.

8.3 Experiment 2B: Excess-of-Loss Reinsurance

8.3.1 Model Setup

Under excess-of-loss reinsurance, the insurer retains only the first $M(t, x)$ units of each claim and cedes the remainder. The retained loss is

$$L(Z, M_t) = \min(Z, M_t),$$

The insurer surplus follows the SDE:

$$dX_t = rX_t dt + \pi_t(\mu - r)dt - \min(Z, M_t)dC_t + \sigma\pi_t dW_t. \tag{10}$$

From 6, this yields the XoL HJB

$$0 = \sup_{\pi,M}\Big\{V_t + (rx + \pi(\mu - r))V_x + \frac{1}{2}\sigma^2\pi^2 V_{xx} + \lambda\big(\mathbb{E}[V(t, x - \min(Z, M))] - V(t, x)\big)\Big\}.$$

Quadrature Approximation.

The non-local jump expectation term

$$\mathbb{E}[V(t, x - \min(Z, M))] = \int_0^\infty V(t, x - \min(Z, M))f_Z(z)\, dz$$

is approximated by the Gaussian quadrature:

$$\widehat{\mathbb{E}}[V] = \sum_{k=1}^{K} \tilde{w}_k\, V(t, x - \min(Z, M)z_k)\, f_Z(z_k).$$

8.3.2 Control Parameterization

We treat $M(t, x)$ as a continuous decision variable learned by the PINN. To enforce the constraint $0 < M(t, x) < M_{\max}$, we use the sigmoid transform:

$$M(t, x) = M_{\max}\,\sigma(M_{\text{raw}}(t, x)), \qquad M_{\max} = 10,$$

ensuring smoothness and compatibility with automatic differentiation.

8.3.3 PINN Architecture

The PINN used for Experiment 2B follows the same architectural design adopted in Experiment 1 and in Experiment 2A. In particular, the network remains a fully connected feed–forward neural network with:

- two input features (t, x),
- four hidden layers of 64 neurons each,
- tanh activation in all hidden layers,
- and three outputs corresponding to the value function and the control variables.

Modification for Excess-of-Loss Reinsurance: The only architectural change arises from replacing the proportional reinsurance control $\theta(t, x)$ with the excess-of-loss deductible $M(t, x)$.

All other components of the PINN, including the computation of spatial and temporal derivatives through automatic differentiation, the loss composition (PDE residual, terminal condition, boundary constraints, and regularization), and the Gaussian–quadrature approximation of jump integrals, remain unchanged.

8.3.4 Training Convergence.

Figure 6 shows the training loss curve for Experiment 2B. The PINN demonstrates stable convergence, with the loss reaching values below 10^{-5} after approximately 3000 epochs. Although oscillations occur, the long-run monotonic decay confirms that the network successfully learns the HJB dynamics under XoL reinsurance.

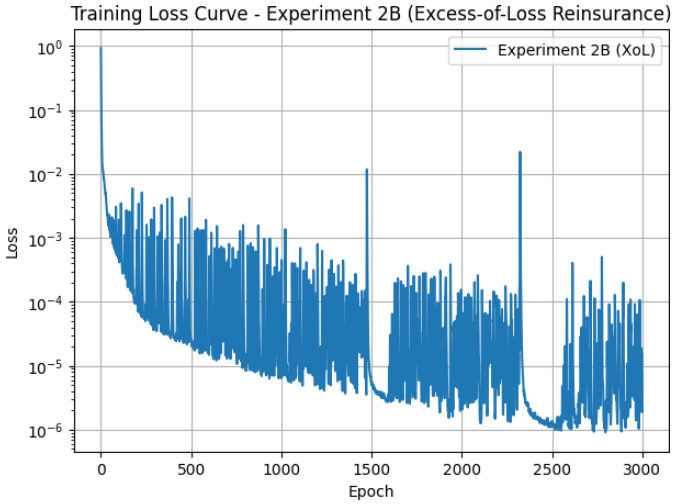
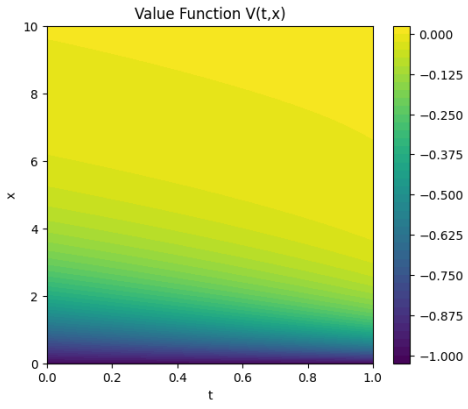
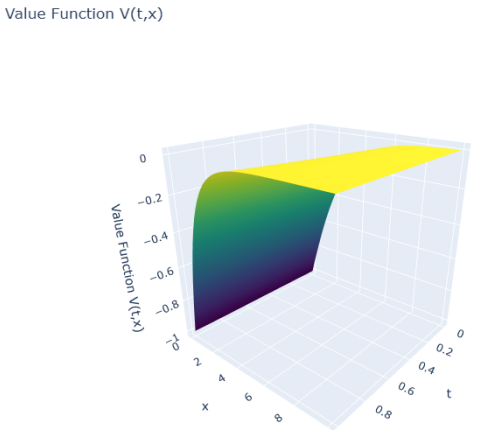


Figure 6: Training loss curve for Experiment 2B (Excess-of-Loss reinsurance).

8.3.5 Value Function $V(t, x)$ under XoL Reinsurance.



Heatmap: $V(t, x)$ under XoL



3D Surface: $V(t, x)$ under XoL

Figure 7: Value function under excess-of-loss reinsurance.

The value function $V(t, x)$ under excess-of-loss reinsurance preserves the expected qualitative features: it is increasing in the surplus level x and varies smoothly in time t, reflecting higher certainty-equivalent utility for better-capitalized states.

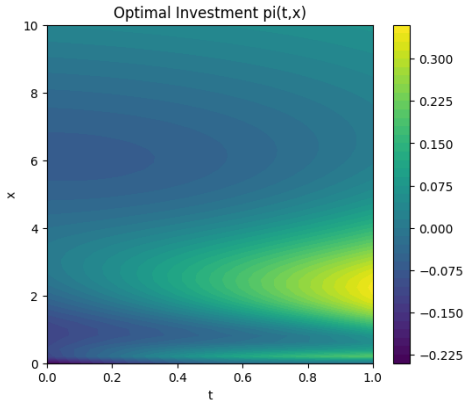
Compared with the proportional reinsurance benchmark (Experiment 2A and 1A), the XoL configuration produces a systematically lower value function at representative interior points. For example, at $(t_0, x_0) = (0.5, 5)$ we have

$$V_{\text{prop}}(t_0, x_0) \approx -0.00954, \qquad V_{\text{XoL}}(t_0, x_0) \approx -0.02709,$$

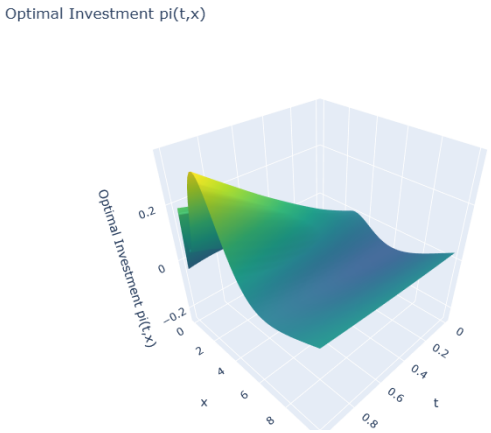
This indicates that, for the same surplus level, the insurer is less well off under XoL than under proportional reinsurance. This is because XoL coverage only activates beyond the deductible and can be relatively expensive when evaluated under exponential utility, so the overall certainty equivalent of terminal wealth is reduced.

The heatmap and 3D surface in 7 together show that this effect is not confined to a single point, but persists over a nontrivial region of the state space, even though the differences in levels remain moderate in absolute terms.

8.3.6 Optimal Investment Policy $\pi(t, x)$ under XoL Reinsurance.



Heatmap: $\pi(t, x)$ under XoL



3D Surface: $\pi(t, x)$ under XoL

Figure 8: Optimal investment under excess-of-loss reinsurance.

The change from proportional to excess-of-loss reinsurance also affects the optimal investment policy.

Under proportional reinsurance (Experiment 2A), the learned $\pi(t, x)$ surface was strictly positive at the reference point $(t_0, x_0) = (0.5, 5)$, with

$$\pi_{\text{prop}}(t_0, x_0) \approx 0.0490,$$

whereas under XoL we obtain

$$\pi_{\text{XoL}}(t_0, x_0) \approx -0.0283.$$

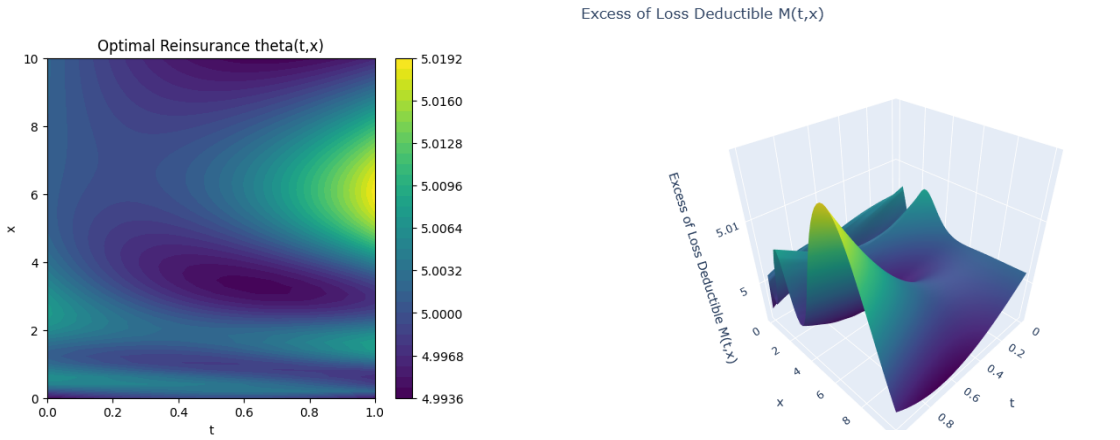
Thus, at this state the optimal exposure to the risky asset switches from a small long position to a small short position.

A negative value of $\pi(t, x)$ corresponds to a hedging position in the risky asset, which reduces the insurer's exposure to market risk.

The heatmap and surface plotnin 8 indicate that, overall, the magnitudes of $\pi(t, x)$ under XoL remain relatively small (close to zero in most of the domain), but there are regions where the policy becomes slightly negative. This behaviour is consistent with the structure of the HJB equation: when large claims are partially retained up to a deductible, the curvature of V with respect to x increases in regions where jump risk is most influential, and the optimal response is to reduce or reverse the risky investment to hedge that residual risk.

In contrast, under proportional reinsurance the risk is more evenly spread across all claim sizes, leading to a smoother and uniformly positive investment policy.

8.3.7 Optimal Deductible $M(t, x)$: Reinsurance Policy under XoL.



Heatmap: $M(t, x)$ under XoL

3D Surface: $M(t, x)$ under XoL

Figure 9: Optimal deductible under excess-of-loss reinsurance.

The deductible surface $M(t, x)$ is the key control variable that distinguishes XoL from proportional reinsurance.

The heatmap scale shows that

$$M(t, x) \in [4.99,\ 5.02]$$

across the entire state space. That is, the learned policy keeps the deductible extremely close to an interior constant level around $M^\star \approx 5$, far below the upper bound $M_{\max} = 10$.

In other words, the PINN does *not* push the deductible to the maximum admissible level; instead, it converges to a nearly flat surface representing an interior optimum.

Economically, this indicates that the insurer finds it optimal to retain moderate-sized losses up to approximately five monetary units while transferring only the extreme tail beyond that level to the reinsurer.

Retaining a much higher deductible would expose the insurer to excessive tail risk, while a much lower deductible would make XoL coverage too costly relative to the protection gained. The fact that $M(t, x)$ is almost constant in both t and x suggests that, under the chosen parameterization (short time horizon, exponential claims with unit mean, and the given market parameters), the risk–premium trade-off that determines the deductible is largely insensitive to the current surplus level and time to maturity.

This behaviour contrasts with the proportional case, where the reinsurance rate $\theta(t, x)$ varies more visibly across the state space.

9 Conclusion

This project applied Physics-Informed Neural Networks (PINNs) to the joint investment - reinsurance optimization problem for an insurance company operating under jump–diffusion surplus dynamics. The presence of nonlocal jump operators makes the associated HJB equation challenging for classical numerical methods, but the PINN framework, augmented with Gaussian quadrature for the jump expectation, successfully solved the problem for both proportional and excess-of-loss reinsurance structures.

Across the three claim-size distributions considered under proportional reinsurance, we observed that tail heaviness plays a central role in shaping optimal policies. Heavy-tailed Pareto claims lead to higher reinsurance levels and lower value functions, reflecting the greater importance of transferring catastrophic losses. The exponential and lognormal models produce smoother and more moderate policies.

In comparing reinsurance structures, excess-of-loss arrangements exhibited qualitatively different behaviour from proportional reinsurance. The optimal deductible converged to a nearly constant interior value, while the optimal investment policy featured small regions of hedging (negative exposure), highlighting how concentrated tail risk interacts with the curvature of the value function. These differences illustrate that the form of risk transfer is at least as important as the distribution of claim sizes.

Overall, the PINN framework produced stable convergence, smooth control surfaces, and interpretable comparative statics, demonstrating its utility for stochastic control problems

with nonlocal jump terms. Future work may include longer horizons, alternative utility specifications, calibration to real insurance data, or extending the approach to multidimensional surplus models.

References

[1] M. Raissi, P. Perdikaris, G. Karniadakis. *Physics-Informed Neural Networks*. Journal of Computational Physics, 2019.

[2] J. Han, A. Jentzen, W. E. *Overcoming the Curse of Dimensionality: Deep Learning for Forward–Backward SDEs*. PNAS, 2017.

[3] J. Sirignano, K. Spiliopoulos. *DGM: A Deep Learning Algorithm for Solving Partial Differential Equations*. Journal of Computational Physics, 2018.

A PINN Algorithm for the Jump–Diffusion HJB Equation

Algorithm 1 Short PINN Algorithm for the Jump–Diffusion HJB Equation

Inputs: Time horizon T, domain $[x_{\min}, x_{\max}]$, claim pdf f_Z, quadrature nodes $\{z_k\}$ and weights $\{\tilde{w}_k\}$.
Initialize: Neural network parameters Θ.
repeat
  Sample interior points (t_i, x_i), terminal points x_j, and boundary points $(t_k, x_{\min\,/max})$.
  for each interior point (t_i, x_i) **do**
    Evaluate NN: (V_i, π_i, θ_i).
    Compute derivatives $V_{t,i}, V_{x,i}, V_{xx,i}$ via autograd.
    Compute shifted points $x_{i,k} = x_i - (1 - \theta_i) z_k$.
    Evaluate $V_{i,k} = V(t_i, x_{i,k})$.
    Approximate jump expectation:

$$\widehat{\mathbb{E}}_i[V] = \sum_k \tilde{w}_k \, V_{i,k} \, f_Z(z_k).$$

    Form HJB residual $\mathcal{R}_i$.
  end for
  Compute losses:

$$\mathcal{L}_{\text{PDE}}, \quad \mathcal{L}_T, \quad \mathcal{L}_{BC}, \quad \mathcal{L}_{\text{reg}}.$$

  Form total loss:

$$\mathcal{L} = w_{\text{PDE}} \mathcal{L}_{\text{PDE}} + w_T \mathcal{L}_T + w_{BC} \mathcal{L}_{BC} + \mathcal{L}_{\text{reg}}.$$

  Update parameters:

$$\Theta \leftarrow \Theta - \eta \nabla_\Theta \mathcal{L}.$$

until convergence
Output: PINN approximations $V(t, x)$, $\pi(t, x)$, $\theta(t, x)$.