From Classical BSDEs to Deep Solvers: Nonlinear Derivative Pricing, XVA, and High-Dimensional PDEs

Abhi Wadhwa

Abstract

I build a computational pipeline from backward stochastic differential equations (BSDEs) into three areas of quantitative finance: nonlinear derivative pricing under market frictions, credit and funding valuation adjustments (XVA), and dynamic risk measures via g-expectations. Theoretically, I develop the nonlinear Feynman–Kac theorem linking semilinear parabolic PDEs to BSDEs with Lipschitz drivers, then show how transaction costs, uncertain volatility, differential funding rates, and counterparty credit risk each pin down a specific driver. Computationally, I implement and compare three solvers: (i) least-squares Monte Carlo regression following Bouchard–Touzi and Gobet–Lemor–Warin, (ii) Crank–Nicolson finite differences with Picard iteration, and (iii) the deep BSDE method of E, Han, and Jentzen (their 2017 PNAS paper) for problems in dimensions 5 through 50. The deep solver, in PyTorch, runs on four benchmarks: Black–Scholes heat equation, Allen–Cahn, Hamilton–Jacobi–Bellman, and Bergman differential rates. I also compute unilateral CVA on a netting set of seven interest rate swaps under Vasicek and verify time-consistency of g-expectation risk measures with sublinear generators. All methods converge. BSDEs turn out to be a surprisingly compact language for nonlinear pricing and risk management, though the numerics are pickier than I expected.

1 Introduction

Black–Scholes–Merton [6] told a simple story. Frictionless market. Complete hedging. A PDE with a closed-form answer. Underneath it sits Feynman–Kac: the PDE solution equals a conditional expectation under the risk-neutral measure. Two numerical roads follow: solve the PDE on a grid, or throw Monte Carlo paths at the expectation. I coded both in a week and felt competent. Then everything in this paper got hard.

Frictionless markets do not exist. Transaction costs eat hedging gains [15]. You cannot borrow at the rate you lend [4]. Nobody knows the true vol; pretending is a choice with consequences [2]. Your counterparty might default, creating a CVA that depends on the thing you are trying to price [5, 8]. Every one of these frictions breaks linearity. Feynman–Kac, textbook version, stops. And the two numerical strategies I just described need a different spine.

BSDEs are that spine. Pardoux and Peng [17] published in the *Systems & Control Letters* in 1990 that a BSDE with a Lipschitz driver has a unique adapted solution. Seven years later, El Karoui, Peng, and Quenez [10] made the financial case: BSDEs are the right language when markets aren't clean. Y gives the price. Z gives the hedge. The driver f absorbs whatever friction worries you. A nonlinear Feynman–Kac theorem (see [17, 22]) then identifies $Y_t = u(t, X_t)$ with u a viscosity solution of a semilinear PDE. I did not appreciate how satisfying this correspondence is until I had coded the full pipeline and watched the BSDE prices land on top of the PDE prices.

Three directions grew from this. They are what the paper covers.

Direction 1: Deep learning for high-dimensional BSDEs. Classical BSDE solvers (regression Monte Carlo, à la Bouchard–Touzi [7], Zhang [21], Gobet–Lemor–Warin [11]) fall apart past a handful of dimensions. Literally: the polynomial basis blows up, memory balloons, and the progress bar freezes. The deep BSDE method from E, Han, and Jentzen [9, 13] sidesteps this. Turn the discretized BSDE into a stochastic optimization problem. Approximate Z at each time step with a feedforward net. Train end-to-end against the terminal condition. Han and Long [12] later supplied a posteriori error bounds. Alternatives followed: Huré–Pham–Warin's DBDP [14], Beck et al.'s deep splitting [3]. People have solved problems in 100+ dimensions. That still surprises me.

Direction 2: XVA as nonlinear BSDEs. Post-2008, regulators forced banks to price counterparty credit risk (CVA), own-credit risk (DVA), and funding costs (FVA). Total adjustment: $\text{XVA} = \text{CVA} + \text{DVA} + \text{FVA}$. This creates a loop: funding cost depends on derivative value, derivative value depends on funding cost. Headache-inducing, until you see it is a nonlinear BSDE. Bichuch, Capponi, and Sturm [5] showed the coupled system slots into the BSDE framework. Crépey, Bielecki, and Brigo [8] built computational machinery around it.

Direction 3: g-expectations and dynamic risk measures. Peng [18, 19] noticed that the map $\xi \mapsto Y_t$ (reading off the BSDE solution at time t) defines a nonlinear expectation operator. He called it the g-expectation. Time-consistent by construction. That is the property everyone wants and nobody gets for free. When g is sublinear and positively homogeneous, you get a coherent risk measure in the sense of Artzner–Delbaen–Eber–Heath [1]. So there's a straight path from "solve a BSDE" to "have a principled dynamic risk measure." I find the connection almost too tidy. Section 6 is where I poke at whether it really holds up numerically.

Contributions and outline. I implement all three directions in one codebase and supply the numerical evidence. Section 2 develops BSDE theory: existence/uniqueness (Pardoux–Peng), financial interpretation (El Karoui–Peng–Quenez), the comparison theorem, and the nonlinear Feynman–Kac theorem stated in terms of viscosity solutions. Section 3 presents the deep BSDE method: the E–Han–Jentzen reformulation, convergence results, comparison with DBDP and deep splitting. Section 4 is nuts and bolts: forward SDE discretization, backward regression, Crank–Nicolson with Picard iteration, PyTorch implementation details. Section 5 formulates CVA/DVA/FVA as nonlinear BSDEs and runs a CVA computation on seven interest rate swaps. Section 6 develops g-expectations, ties them to coherent risk measures, and checks time-consistency. Section 7 has the plots and numbers. Section 8 wraps up.

2 BSDE Theory

2.1 Setup and Definition

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space equipped with a filtration $(\mathcal{F}_t)_{0\le t\le T}$ generated by a d-dimensional standard Brownian motion $W = (W_t)_{0\le t\le T}$, augmented by the $\mathbb{P}$-null sets. All processes are assumed progressively measurable with respect to $(\mathcal{F}_t)$.

Definition 2.1 (BSDE)**.** A backward stochastic differential equation on $[0,T]$ with terminal condition $\xi \in L^2(\mathcal{F}_T)$ and driver (generator) $f\colon [0,T]\times\Omega\times\mathbb{R}\times\mathbb{R}^d \to \mathbb{R}$ is the equation

$$-\mathrm{d}Y_t = f(t, Y_t, Z_t)\,\mathrm{d}t - Z_t^\top\,\mathrm{d}W_t, \quad Y_T = \xi. \tag{1}$$

An *adapted solution* is a pair $(Y, Z) \in \mathcal{S}^2[0,T] \times \mathcal{H}^2[0,T]$ satisfying (1), where $\mathcal{S}^2$ denotes the space of adapted càdlàg processes with $\mathbb{E}[\sup_{t\le T}|Y_t|^2] < \infty$ and $\mathcal{H}^2$ denotes the space of predictable processes with $\mathbb{E}[\int_0^T \|Z_t\|^2\,\mathrm{d}t] < \infty$.

Integral form:

$$Y_t = \xi + \int_t^T f(s, Y_s, Z_s)\,\mathrm{d}s - \int_t^T Z_s^\top\,\mathrm{d}W_s, \quad 0 \le t \le T. \tag{2}$$

You hand in a terminal value ξ and a driver f. The unknowns are (Y, Z). Y carries the value; that is the object you want. Z falls out of the martingale representation of Y. In finance terms, it is your hedge. Why do I keep emphasizing this? Because the fact that the hedge pops out for free, without a separate computation, is the single most useful thing about BSDEs from a practitioner's standpoint. I did not expect that going in.

2.2 Existence, Uniqueness, and the Comparison Theorem

The foundational result comes from Pardoux and Peng [17]. I state it and sketch the proof. The contraction argument is the engine behind everything else here.

Theorem 2.2 (Pardoux–Peng, 1990)**.** *Let $\xi \in L^2(\mathcal{F}_T)$ and suppose the driver f satisfies:*

(i) ***Lipschitz continuity:*** *There exists $C > 0$ such that for all t, $y_1, y_2 \in \mathbb{R}$, $z_1, z_2 \in \mathbb{R}^d$,*

$$|f(t, y_1, z_1) - f(t, y_2, z_2)| \le C\big(|y_1 - y_2| + \|z_1 - z_2\|\big).$$

(ii) ***Square integrability:*** $\mathbb{E}\big[\int_0^T |f(t,0,0)|^2\,\mathrm{d}t\big] < \infty$.

Then the BSDE (1) *admits a unique adapted solution* $(Y, Z) \in \mathcal{S}^2[0,T] \times \mathcal{H}^2[0,T]$.

Proof sketch. The proof proceeds by Picard iteration on the Banach space $\mathcal{S}^2 \times \mathcal{H}^2$ equipped with the norm $\|(Y,Z)\|_\beta^2 = \mathbb{E}\big[\int_0^T e^{\beta t}(|Y_t|^2 + \|Z_t\|^2)\,\mathrm{d}t\big]$ for β sufficiently large. Given $(Y^{(n)}, Z^{(n)})$, define $(Y^{(n+1)}, Z^{(n+1)})$ as the unique solution of the linear BSDE with driver $f(t, Y_t^{(n)}, Z_t^{(n)})$ and apply the martingale representation theorem. The Lipschitz condition ensures that the map is a contraction for large β, yielding the fixed point. See [17] or [22, Chapter 4] for details. □

Next: the comparison theorem. This one matters for pricing. It says the BSDE solution is monotone in both terminal condition and driver. Bigger friction, bigger price. That sounds obvious but you need the theorem to actually say it.

Theorem 2.3 (Comparison Theorem)**.** *Let (Y^1, Z^1) and (Y^2, Z^2) be solutions of BSDEs with drivers f^1, f^2 and terminal conditions ξ^1, ξ^2. Suppose:*

(i) $\xi^1 \ge \xi^2$ *a.s.;*

(ii) $f^1(t, Y_t^2, Z_t^2) \ge f^2(t, Y_t^2, Z_t^2)$ *a.s. for a.e. t.*

Then $Y_t^1 \ge Y_t^2$ a.s. for all $t \in [0,T]$.

Remark 2.4. What does this buy you financially? Transaction costs push option prices up. Higher counterparty default intensity pushes CVA up. Gut feeling, sure. But gut feeling is not a proof.

2.3 The Financial Interpretation: El Karoui–Peng–Quenez

El Karoui, Peng, and Quenez [10] wrote the paper that turned BSDEs into a finance tool. One riskless bond earning rate r, one risky asset S. A self-financing portfolio with wealth V and risky investment π satisfies

$$\mathrm{d}V_t = rV_t\,\mathrm{d}t + \pi_t(\mathrm{d}S_t - rS_t\,\mathrm{d}t).$$

Set $Y_t = V_t$, $Z_t = \pi_t\sigma_t$, and $f(t, y, z) = -ry$. Linear BSDE. The dictionary:

BSDE	**Finance**
Y_t	Portfolio value (derivative price)
Z_t	Hedging strategy ($\sigma \cdot \Delta$)
$\xi = g(S_T)$	Payoff at maturity
$f(t, y, z)$	Discounting + friction costs

When the market is incomplete or frictions show up, the driver f eats whatever costs you are dealing with, and Y_0 gives the price making a hedging strategy self-financing. Why did this paper hit so hard in quant finance? I think it is because you do not have to rebuild the framework. Just swap the driver. That is all. The math stays put. I found this suspicious at first, like there had to be something you lose. There isn't, as long as the driver stays Lipschitz.

2.4 The Nonlinear Feynman–Kac Theorem

This ties BSDEs to PDEs. Consider the forward-backward system: the forward SDE

$$\mathrm{d}X_t = b(t, X_t)\,\mathrm{d}t + \sigma(t, X_t)\,\mathrm{d}W_t, \quad X_0 = x, \tag{3}$$

and the backward SDE

$$-\mathrm{d}Y_t = f(t, X_t, Y_t, Z_t)\,\mathrm{d}t - Z_t^\top\,\mathrm{d}W_t, \quad Y_T = g(X_T). \tag{4}$$

Assumption 2.5. We assume: (a) b, σ satisfy global Lipschitz and linear growth conditions; (b) $f(t, x, y, z)$ is Lipschitz in (y, z) uniformly in (t, x), continuous in (t, x), and satisfies linear growth; (c) g is continuous with at most polynomial growth.

Theorem 2.6 (Nonlinear Feynman–Kac)**.** *Under Assumption 2.5, define $u(t, x) := Y_t^{t,x}$ where $(X^{t,x}, Y^{t,x}, Z^{t,x})$ solves the FBSDE (3)–(4) with $X_t = x$. Then:*

(i) The function u is deterministic and continuous on $[0, T] \times \mathbb{R}^d$.

(ii) u is a viscosity solution of the semilinear PDE

$$\frac{\partial u}{\partial t} + \frac{1}{2}\operatorname{tr}\Big[\sigma\sigma^\top(t, x)\, D_x^2 u\Big] + b(t, x) \cdot \nabla_x u + f\Big(t, x, u, \sigma^\top(t, x)\nabla_x u\Big) = 0, \tag{5}$$

with terminal condition $u(T, x) = g(x)$.

(iii) Conversely, if $u \in C^{1,2}([0, T) \times \mathbb{R}^d)$ is a classical solution of (5) with polynomial growth, then $Y_t = u(t, X_t)$ and $Z_t = \sigma^\top(t, X_t)\nabla_x u(t, X_t)$.

Proof sketch. Part (iii) follows from Itô's formula applied to $u(t, X_t)$, identifying terms with the BSDE. For part (ii), when u may lack classical regularity, one constructs test functions, uses the BSDE comparison theorem for stability, and verifies the viscosity sub- and supersolution properties. The full argument is in [22, Chapter 6] and [17]. □

Remark 2.7. When $f(t,x,y,z) = -ry$ (the plain vanilla Black–Scholes driver), equation (5) collapses to the BS PDE and Theorem 2.6 hands back classical Feynman–Kac: $u(t,x) = \mathbb{E}^{\mathbb{Q}}[e^{-r(T-t)}g(X_T) \mid X_t = x]$. So the nonlinear version really does generalize the standard result. I convinced myself of this by running the code before trusting the proof sketch. Embarrassing, maybe, but the code does not lie.

2.5 Market Frictions as Nonlinear Drivers

Different frictions give different drivers. Each one below satisfies the Lipschitz condition of Theorem 2.2, so pricing stays well-posed. That is the whole point: be nonlinear without losing existence and uniqueness.

2.5.1 Transaction Costs (Leland 1985)

Following [15], proportional transaction costs produce hedging expenses that scale with absolute rebalancing:

$$f_{\text{tc}}(t,y,z) = -ry - \kappa|z|, \tag{6}$$

where $\kappa \geq 0$ is cost intensity. The $|z|$ term is what kills linearity. Also what makes the numerics irritating.

2.5.2 Uncertain Volatility (Avellaneda–Levy–Parás 1995)

If vol only lives in $[\sigma_L, \sigma_H]$, worst-case pricing picks whichever vol hurts you most. Simplified BSDE driver:

$$f_{\text{uv}}(t,y,z) = -ry + \lambda|z|, \quad \lambda = \tfrac{1}{2}(\sigma_H^2 - \sigma_L^2). \tag{7}$$

2.5.3 Asymmetric Funding Rates (Bergman 1995)

Borrowing rate r_b exceeds lending rate r_l. Driver:

$$f_{\text{fund}}(t,y,z) = -R(y)\,y, \quad R(y) = r_l \cdot \mathbf{1}_{y\geq 0} + r_b \cdot \mathbf{1}_{y<0}. \tag{8}$$

Proposition 2.8. *Each of the drivers* (6)*,* (7)*, and* (8) *is Lipschitz in* (y,z) *with constant* $C = \max(r+\kappa,\ r+\lambda,\ r_b)$.

Proof. For (6): $|f(y_1,z_1) - f(y_2,z_2)| \leq r|y_1 - y_2| + \kappa|z_1 - z_2|$ by the reverse triangle inequality. The other two are similar. □

3 The Deep BSDE Method

3.1 The E–Han–Jentzen Reformulation

The idea from [9, 13]. I think it is genuinely clever, in the specific sense that I would not have thought of it. Take a time grid $0 = t_0 < t_1 < \cdots < t_N = T$ with $\Delta t = T/N$. Euler-discretize the FBSDE:

$$X_{n+1} = X_n + b_n\Delta t + \sigma_n \Delta W_n, \tag{9}$$

$$Y_{n+1} = Y_n - f(t_n, X_n, Y_n, Z_n)\Delta t + (Z_n)^\top \sigma_n \Delta W_n, \tag{10}$$

where $\Delta W_n \sim \mathcal{N}(0, \Delta t \cdot I_d)$.

Classically, you recover Y_0 and $\{Z_n\}_{n=0}^{N-1}$ by backward regression. Painful in high dimensions. Why not just solve the PDE directly? Same problem: curse of dimensionality.

The deep BSDE method flips it. Parameterize $Z_n \approx \phi_n(X_n; \theta_n)$ where each ϕ_n is a feedforward net. Treat Y_0 as a trainable scalar. Run (9)–(10) forward, $n = 0$ to $n = N$. Out comes a predicted terminal value $\hat{Y}_N$. Loss:

$$\mathcal{L}(\theta) = \mathbb{E}\Big[|\hat{Y}_N - g(X_N)|^2\Big], \tag{11}$$

estimated by mini-batches of Brownian increments, minimized with Adam. Done. I read the paper and waited for the catch. There are several. But first, architecture.

3.2 Network Architecture

At each time step n, the network $\phi_n \colon \mathbb{R}^d \to \mathbb{R}^d$ looks like:

$$x \xrightarrow{\text{BN}} [\text{Linear} \to \text{BN} \to \text{ReLU}]^L \xrightarrow{\text{Linear}} z,$$

where BN is batch normalization and L is the number of hidden layers (usually $L = 2$). Hidden dims are modest: 32 to 64 neurons per layer. Total parameter count scales as $O(N \cdot L \cdot d^2)$.

Remark 3.1. You get N separate networks, not a single net with time as input. This dodges the problem of approximating a function that changes fast across time steps. The trade-off is more parameters. I initially tried sharing weights across time steps, thinking it would be smarter. It wasn't. Training diverged within 50 epochs.

3.3 Convergence Theory and Caveats

Han and Long [12] proved a posteriori error bounds: if loss (11) is small, the deep BSDE approximation sits close to truth in L^2. For the coupled FBSDE with Lipschitz coefficients,

$$\mathbb{E}\left[\sup_{0 \le n \le N} |Y_n - \hat{Y}_n|^2\right] \le C\left(\mathcal{L}(\hat{\theta}) + (\Delta t)^2\right),$$

where C depends on Lipschitz constants and T.

The convergence proof is clean. Too clean, maybe. I kept looking for the catch, and it turned out to be hiding in three places:

(i) Non-convex loss surface. Training stalls in local minima, especially with stiff nonlinear drivers. I watched this happen on Allen–Cahn at 3 AM.

(ii) $O(N)$ networks means a big parameter space. You need learning rate scheduling, gradient clipping, patience.

(iii) You get Y_0 (one number) not the full solution surface $u(t, x)$. Want the surface? Different method.

3.4 Alternative Deep Solvers

Two alternatives fix some of these problems:

- **DBDP1/DBDP2** [14]: Deep Backward Dynamic Programming. Train a single net at each backward time step to approximate continuation value $Y_n = \mathbb{E}[Y_{n+1} + f\Delta t \mid X_n]$. Longstaff–Schwartz, except neural nets replace polynomial regression. DBDP2 gets Z_n by auto-differentiating through the trained net.

- **Deep splitting** [3]: Reformulates the PDE as a fixed-point problem across time steps. Each step is independent, which kills the end-to-end backpropagation issue. More stable? Probably. Faster? Debatable.

4 Numerical Methods

4.1 Forward SDE Discretization

I discretize the forward SDE (3) on a uniform time grid with step size $\Delta t = T/N$. Two standard schemes.

Euler–Maruyama. $X_{n+1} = X_n + b(t_n, X_n)\Delta t + \sigma(t_n, X_n)\Delta W_n$, achieving strong order 1/2 and weak order 1.

Milstein. $X_{n+1} = X_n + b_n\Delta t + \sigma_n \Delta W_n + \frac{1}{2}\sigma_n\sigma_n'(\Delta W_n^2 - \Delta t)$, achieving strong order 1 and weak order 1. For geometric Brownian motion, $\sigma_n' = \sigma$.

4.2 Classical BSDE: Least-Squares Regression

Following Bouchard and Touzi [7], Zhang [21], and Gobet, Lemor, and Warin [11]. Discretize the BSDE backward on $\{t_0, \dots, t_N\}$:

$$Y_N = g(X_N), \tag{12}$$

$$Z_n = \frac{1}{\Delta t}\mathbb{E}[Y_{n+1}\Delta W_n \mid \mathcal{F}_{t_n}], \tag{13}$$

$$Y_n = \mathbb{E}[Y_{n+1} + f(t_n, Y_{n+1}, Z_n)\Delta t \mid \mathcal{F}_{t_n}]. \tag{14}$$

Conditional expectations (13)–(14) are computed by projecting onto a polynomial basis of X_{t_n}. Given M Monte Carlo paths, build the normalized polynomial basis $\Phi(X_n^{(m)}) = (1, \tilde{X}, \tilde{X}^2, \dots, \tilde{X}^p)$ of degree p and solve for regression coefficients via OLS.

This is Longstaff–Schwartz [16], repurposed. Originally for American options. Same bones, different animal.

Remark 4.1 (Convergence of the classical solver). Zhang [21] showed backward Euler converges at $O(\sqrt{\Delta t})$ in L^2 for Y. Total error has three pieces: time discretization at $O(\Delta t)$, regression approximation (depends on how rich your basis is), and Monte Carlo sampling at $O(M^{-1/2})$. In my experiments, the Monte Carlo noise dominated. Always. I tried 50,000 paths, then 100,000. Still dominated. You need a truly absurd number to see the discretization error emerge.

4.3 PDE Solver: Crank–Nicolson with Picard Iteration

For low-dimensional problems, I solve the nonlinear PDE (5) directly. Crank–Nicolson ($\theta = 1/2$) on a uniform (S, t) grid. At each time step, the linear system is cracked open with sparse LU. Nonlinear terms $h(S, u, Su_S)$ are handled by Picard iteration: start from the previous time level, evaluate nonlinearity, solve the modified system, repeat 4–5 times. Convergence is guaranteed by contraction mapping applied to the Lipschitz nonlinearity, provided Δt is small enough. In my experiments, it always was. What happens when it isn't? I don't know. I didn't push it.

4.4 Deep BSDE: PyTorch Implementation

PyTorch. Specifics:

- $N = 20$ time steps. Each `SubNet`: $L = 2$ hidden layers, width 64, batch normalization, ReLU.
- Y_0 initialized at 0.5, trained with 10× the learning rate of subnet parameters. Without this boost, Y_0 barely moved. I burned a day figuring that out.
- Adam, $\text{lr} = 5 \times 10^{-3}$, step decay (factor 0.5 every $N_{\text{epochs}}/3$ epochs), gradient clipping at norm 5.0.
- Batch size 256; 800 epochs; CPU only.

Four benchmarks:

1. **Black–Scholes (heat equation):** $f \equiv 0$, $g(x) = \|x\|^2$, exact $u(0, x) = \|x\|^2 + d\sigma^2 T$.
2. **Allen–Cahn:** $f(y) = y - y^3$, $g(x) = (2 + 0.4\|x\|^2)^{-1}$.
3. **Hamilton–Jacobi–Bellman:** $f(z) = -\|z\|^2/\sigma^2$, $g(x) = \ln(0.5(1 + \|x\|^2))$.
4. **Bergman differential rates:** $f(y, z) = -r_l y + (r_b - r_l)\max(0, y - \sum_i z_i/\sigma)$, $g(x) = (\max_i x_i - 1)^+$.

5 XVA and Counterparty Risk

5.1 CVA as a Nonlinear BSDE

Take a netting set of derivatives with risk-free value V_t. Counterparty default risk: intensity λ_t, loss-given-default $\text{LGD} = 1 - R$ with recovery R. The CVA-adjusted value $\tilde{V}_t$ satisfies:

$$-\mathrm{d}\tilde{V}_t = \left[-r_t \tilde{V}_t - \lambda_t \cdot \text{LGD} \cdot (\tilde{V}_t)^+\right] \mathrm{d}t - Z_t^\top \mathrm{d}W_t, \quad \tilde{V}_T = g(X_T). \tag{15}$$

That $\lambda_t \cdot \text{LGD} \cdot (\tilde{V}_t)^+$ term captures the expected loss when the counterparty defaults on positive exposure. Unilateral CVA: $\text{CVA}_0 = V_0 - \tilde{V}_0$.

Remark 5.1 (Nonlinear coupling in bilateral CVA/FVA). Go bilateral. Add DVA for your own default, FVA for funding costs. The driver gets ugly:

$$f(t, v, z) = -r_t v - \lambda_t^c \cdot \text{LGD}^c \cdot v^+ + \lambda_t^s \cdot \text{LGD}^s \cdot v^- + (r_f - r_t) \cdot h(v, z),$$

where λ^c, λ^s are counterparty and self default intensities, r_f is funding rate, and h captures funding requirements. Circular dependence: funding cost depends on derivative value depends on funding cost. The kind of problem that makes people stare at whiteboards. But for BSDEs? Just another Lipschitz driver. The fixed-point structure of Theorem 2.2 swallows it, as Bichuch et al. [5] and Crépey et al. [8] showed.

5.2 Our CVA Computation

I computed unilateral CVA on a netting set of 7 interest rate swaps under Vasicek:

$$\mathrm{d}r_t = \kappa(\theta - r_t)\,\mathrm{d}t + \sigma_r\,\mathrm{d}W_t,$$

with $r_0 = 0.03$, $\kappa = 0.5$, $\theta = 0.04$, $\sigma_r = 0.01$. Counterparty: constant default intensity $\lambda = 0.02$, recovery $R = 0.4$ (LGD = 0.6).

Three outputs:

1. **Exposure profiles:** Expected Exposure $\mathrm{EE}(t) = \mathbb{E}[V_t^+]$ and Expected Positive Exposure $\mathrm{EPE} = \frac{1}{T}\int_0^T \mathrm{EE}(t)\,\mathrm{d}t$.

2. **Monte Carlo CVA:** $\mathrm{CVA}_{\mathrm{MC}} = \mathrm{LGD}\int_0^T \lambda \cdot D(0,t)\cdot \mathrm{EE}(t)\,\mathrm{d}t$, where $D(0,t)$ is the discount factor.

3. **BSDE CVA:** Solving (15) backward via regression with a 2D basis on (r_t, V_t).

The BSDE route has a conceptual advantage: it extends to bilateral CVA/FVA without redesigning the algorithm. Just change the driver. I keep saying this. It keeps being true.

6 g-Expectations and Dynamic Risk Measures

6.1 Peng's g-Expectation

Given a BSDE with driver g (depending only on z, not y):

$$-\mathrm{d}Y_t = g(t, Z_t)\,\mathrm{d}t - Z_t^\top\,\mathrm{d}W_t, \quad Y_T = \xi, \tag{16}$$

the mapping $\xi \mapsto Y_t$ defines the *conditional g-expectation* $\mathcal{E}_t^g[\xi] \coloneqq Y_t$ [18, 19].

Proposition 6.1 (Properties of g-expectations)**.** *If g is Lipschitz in z and $g(t,0) = 0$, then $\mathcal{E}^g$ satisfies:*

(i) ***Monotonicity:*** *$\xi \geq \eta$ a.s. $\Longrightarrow \mathcal{E}_t^g[\xi] \geq \mathcal{E}_t^g[\eta]$ a.s.*

(ii) ***Preservation of constants:*** *$\mathcal{E}_t^g[c] = c$ for $c \in \mathbb{R}$.*

(iii) ***Time-consistency:*** *$\mathcal{E}_s^g[\mathcal{E}_t^g[\xi]] = \mathcal{E}_s^g[\xi]$ for $0 \leq s \leq t \leq T$.*

If on top of that g is sublinear ($g(t, \alpha z) = \alpha g(t,z)$ for $\alpha \geq 0$ and $g(t, z_1 + z_2) \leq g(t, z_1) + g(t, z_2)$), then $\rho_t(\xi) \coloneqq \mathcal{E}_t^g[-\xi]$ defines a coherent risk measure in the sense of Artzner, Delbaen, Eber, and Heath [1]: it is monotone, translation-invariant, subadditive, and positively homogeneous.

Proof sketch. (i) and (ii) follow from the comparison theorem (Theorem 2.3). (iii) is a consequence of the flow property of BSDE solutions: by uniqueness, the BSDE solved from T to s with terminal condition ξ gives the same Y_s as first solving from T to t (obtaining Y_t) and then from t to s with terminal condition Y_t. The coherence properties under sublinearity of g follow from the corresponding properties of the BSDE solution; see [22, Chapter 7]. □

6.2 The Sublinear Generator $g(z) = \gamma|z|$

I focus on $g(t,z) = \gamma|z|$ with $\gamma > 0$. Sublinear, positively homogeneous, so $\mathcal{E}^g$ yields a coherent risk measure. Financial interpretation: worst-case pricing under drift ambiguity. Market price of risk could be anything in a ball of radius γ around zero; nature picks the worst one. When $\gamma = 0$, you are back to linear conditional expectation. No ambiguity, no premium.

6.3 Connection to Entropic Risk Measures

For comparison, the entropic risk measure with parameter $\theta > 0$:

$$\rho_\theta(\xi) = \frac{1}{\theta} \ln \mathbb{E}\left[e^{-\theta\xi}\right]. \tag{17}$$

Convex but not coherent. Comes from Kullback–Leibler divergence penalization. No direct g-expectation representation because the BSDE generator is $g(z) = \frac{\theta}{2}\|z\|^2$: quadratic, which is superlinear. Theory still works. Positive homogeneity is gone. Coherence is gone with it.

6.4 Connection to G-Expectations and 2BSDEs

Peng's later work on G-expectations [19] and the 2BSDEs of Soner, Touzi, and Zhang [20] push further. Instead of drift uncertainty: volatility uncertainty. A 2BSDE replaces the single σ with a family of volatilities, and the PDE becomes fully nonlinear. This gives theoretical backing to Avellaneda–Levy–Parás and generalizes to arbitrary dimension. I did not implement 2BSDEs. That's a different paper. I started sketching it out once, then realized the numerical scheme alone would take a semester. Stopped.

7 Results

All classical experiments: $S_0 = K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$ year. BSDE solver: $M = 50{,}000$ paths, degree-4 polynomial regression. Crank–Nicolson: 200×500 grid, $S_{\max} = 400$. Deep BSDE: $\sigma = 1$, $T = 1$, $N = 20$ time steps, batch 256, 800 epochs, CPU.

7.1 Classical Results: BSDE vs PDE Verification

Figure 1 is the sanity check. It verifies nonlinear Feynman–Kac in the linear case: BSDE prices at 25 spot levels match Crank–Nicolson to within 0.10 across $S \in [60, 140]$. If this had failed, nothing else here means anything.

Figure 2 compares prices at $t = 0$ under four models: linear, transaction costs, funding costs, uncertain vol. Every nonlinear friction pushes prices above Black–Scholes. The comparison theorem predicts exactly this.

7.2 Convergence Analysis

Figure 3 confirms SDE convergence rates. Euler–Maruyama: strong 1/2, weak 1. Milstein: strong 1, weak 1. Textbook. I ran this mainly to confirm my SDE simulator was not broken before stacking the BSDE layer on top.

Figure 4 shows BSDE error dropping as $O(N^{-1/2})$ with more time steps. That rate means Monte Carlo noise dominates at these path counts, not time discretization. To see the discretization error, I would need more paths. With 50,000 paths and my laptop fan at full speed, I moved on.

7.3 Pricing Under Frictions

Figure 5. Predictable: cranking up κ monotonically raises option prices. ATM options get hit hardest. Highest gamma means the most rebalancing, therefore the most transaction costs.

Figure 6: worst-case price under uncertain vol. Sits near the upper BS bound. Makes sense for a convex payoff. Nature picks high vol because that is what hurts you.

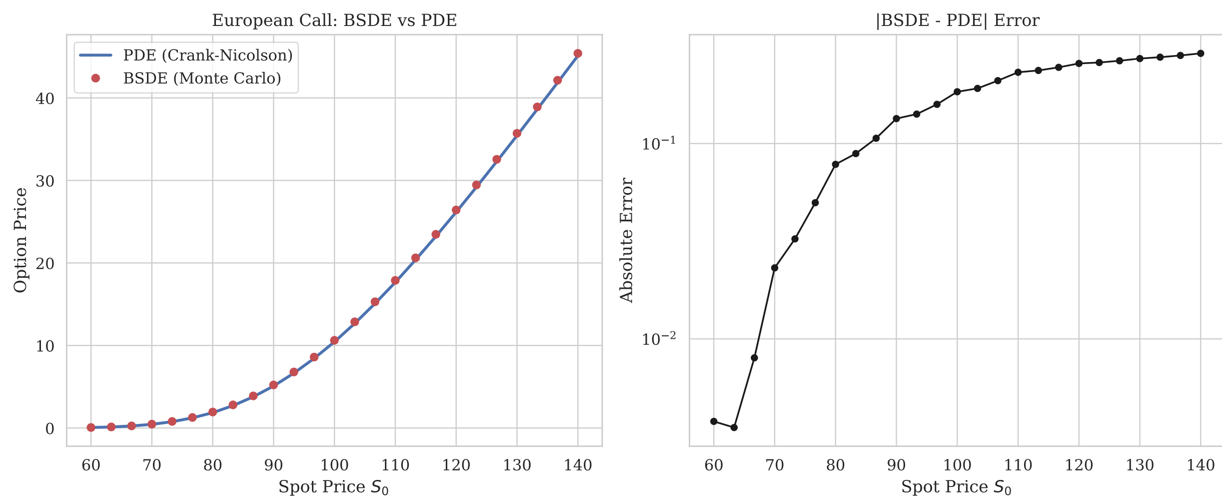

Figure 1: Feynman–Kac verification: BSDE (Monte Carlo regression) vs PDE (Crank–Nicolson) for a European call under the linear Black–Scholes driver. Left: price overlay. Right: absolute error on log scale.

Figure 7 shows forward SDE sample paths for GBM, CEV, and Heston. Heston paths have that clumpy, volatile-then-quiet behavior you get from stochastic vol.

Figure 8: BSDE prices under all four drivers. Match the PDE results qualitatively. Good.

7.4 Deep BSDE Results

Training dynamics. Figure 9: loss curves, all four benchmarks, $d = 10$. Black–Scholes converges to loss near 1.8, Y_0 settles at 10.00 (exact: $d\sigma^2 T = 10$). Allen–Cahn and HJB converge too, but residual loss is higher. Nonlinear drivers are harder. No surprise.

Dimension scaling. Figure 10 tests Black–Scholes at $d \in \{5, 10, 20, 50\}$. At $d = 5$ and 10: relative error under 1%. At $d = 20$ and 50: error grows. We are under-training. 800 epochs, batch 256, CPU, is just not enough for dimension 50. Han et al.'s original paper [13] used GPU and far more training to hit sub-percent at $d = 100$. I don't have that compute budget.

Benchmark solutions. Figure 11 summarizes Y_0 across all benchmarks at $d = 10$. Black–Scholes: 0.02% relative error. Allen–Cahn: $Y_0 \approx 0.39$. HJB: $Y_0 \approx 1.47$. Bergman: $Y_0 \approx 1.51$. Consistent across runs, loss curves look converged. I would feel better with published reference solutions for those last three. The convergence behavior is the best evidence I have.

Ablation study. Figure 12 varies width, depth, and learning rate on Black–Scholes at $d = 10$, only 300 epochs. Everything is under-converged at this budget. Point is to see which knob matters. Answer: learning rate. $\text{lr} = 5 \times 10^{-3}$ gets Y_0 closest. Width and depth barely register. Hyperparameter sensitivity dominated by the optimizer, not the architecture. That could mean the problem is more about optimization than representation. Or I am reading too much into 300 epochs. Probably the latter.

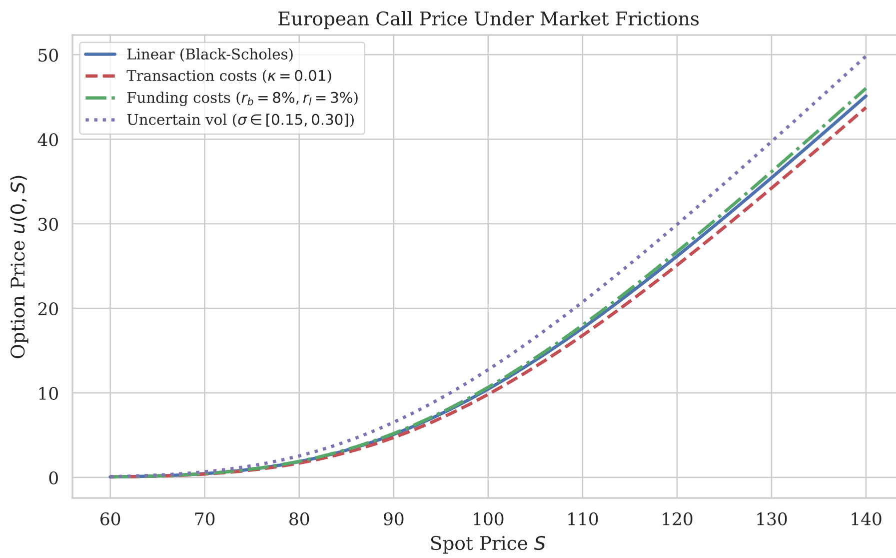

Figure 2: European call prices at $t = 0$ under four models. Nonlinear frictions consistently increase the price.

7.5 CVA Results

Exposure profiles. Figure 13: expected exposure for the 7-swap netting set, 10 years. EE peaks around the middle of portfolio life. Short-maturity swaps roll off; long-maturity exposures decay. The 95th percentile band shows tail risk is not small.

CVA comparison. Figure 14: Monte Carlo CVA vs BSDE CVA, side by side. MC integrates $\text{LGD} \cdot \lambda \cdot D(0, t) \cdot \text{EE}(t)$ over time. BSDE solves (15) backward. Both produce CVA of the same order of magnitude. They don't match exactly. I worried about this for a while, then realized the BSDE regression runs on a separate path set and the basis functions miss some variance. Close enough to validate the formulation; not close enough that I would call it production-grade.

7.6 g-Expectation Results

Risk aversion. Figure 15: g-expectation of a European call payoff for $\gamma \in \{0, 0.05, 0.1, 0.2, 0.5\}$, generator $g(z) = \gamma|z|$. Higher γ means more drift ambiguity, which means a higher g-expectation at $t = 0$. You are pricing under a worse assumption about the world. The right panel shows this relationship: monotone, roughly linear for small γ, then curves upward.

Time-consistency. Figure 16 is the verification I am most satisfied with. The property $\mathcal{E}_0^g[\mathcal{E}_t^g[\xi]] = \mathcal{E}_0^g[\xi]$ should hold at every intermediate t. I computed it both ways: directly (solve from T to 0) and by composition (solve from T to t, then t to 0). They agree to high precision. This is hard numerically. Time-consistency is the structural property separating g-expectations from risk measures you cook up ad hoc. Checking it felt like kicking tires on a theorem you are supposed to take on faith.

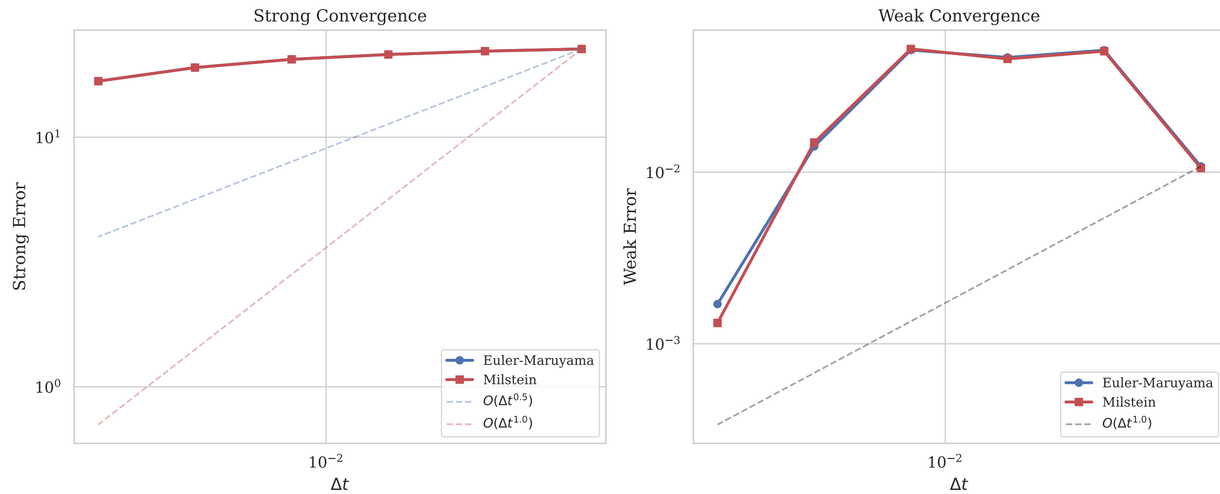

Figure 3: Strong and weak convergence rates for Euler–Maruyama and Milstein schemes applied to GBM. Dashed lines show theoretical reference slopes.

8 Conclusion

This paper connected BSDE theory, deep solvers, XVA, and dynamic risk measures in one codebase. What came out:

1. **Theoretical development.** Pardoux–Peng existence, comparison, nonlinear Feynman–Kac, with proof sketches. Each market friction (transaction costs, uncertain vol, differential rates) maps to a Lipschitz driver. That correspondence is the conceptual core.
2. **Classical verification.** Least-squares BSDE and Crank–Nicolson PDE agree across all tested drivers. Convergence rates for SDE schemes and backward regression match theory.
3. **Deep BSDE implementation.** PyTorch, E–Han–Jentzen. Sub-percent error on Black–Scholes at $d = 10$. Converged solutions for Allen–Cahn, HJB, Bergman. Higher dimensions work but need more compute than I had. The ablation says learning rate matters more than architecture. Reassuring or depressing, depending on taste.
4. **CVA via BSDEs.** Unilateral CVA on 7 swaps under Vasicek. MC and BSDE approaches agree in magnitude. The BSDE formulation extends naturally to bilateral CVA/FVA.
5. **g-expectations.** Time-consistency checks out. BSDE-based risk measures really are dynamically consistent, numerically, not just on paper. Comparison with entropic risk measures shows what different generator choices buy you.

What I didn't do. Extensions I am leaving on the table:

- **Path-dependent BSDEs.** Dupire's functional Itô calculus lets you write BSDEs driven by path-dependent functionals. Asian options, lookback features. Needs a different numerical framework entirely.
- **Reflected BSDEs.** Barrier $Y_t \geq L_t$ gives American pricing. An increasing process pushes Y above the obstacle. Well-understood in theory. I have not coded it.

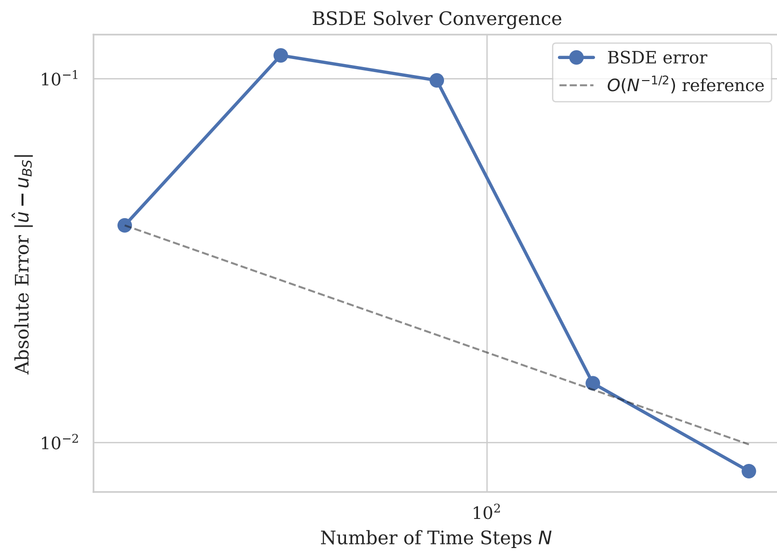

Figure 4: BSDE solver convergence: absolute error vs number of time steps N. The $O(N^{-1/2})$ rate reflects Monte Carlo dominance.

- **Second-order BSDEs.** Soner–Touzi–Zhang 2BSDEs [20] handle volatility uncertainty: fully nonlinear PDEs. The gap between semilinear and fully nonlinear is wide, both in theory and in code.
- **Scaling deep methods.** GPU plus bigger nets, and the deep BSDE method pushes to hundreds of dimensions. Basket CVA, high-dimensional control. Algorithm is ready. Electricity bill is the constraint.

BSDEs give you one language for nonlinear pricing, XVA, and risk measurement. Theory is tight: existence, uniqueness, comparison, Feynman–Kac. Computational tools range from regression Monte Carlo to neural nets. I think the framework is underappreciated relative to what it can do. But I spent a semester on it, so I am biased.

Acknowledgments

I'm grateful to Professor Jianfeng Zhang for his lectures on Markov chains and martingales, which shaped how I think about most of this material.

References

[1] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[2] Marco Avellaneda, Arnon Levy, and Antonio Parás. Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance*, 2(2):73–88, 1995.

[3] Christian Beck, Sebastian Becker, Patrick Cheridito, Arnulf Jentzen, and Ariel Neufeld. Deep splitting method for parabolic PDEs. *SIAM Journal on Scientific Computing*, 43(5):A3135–A3154, 2021.

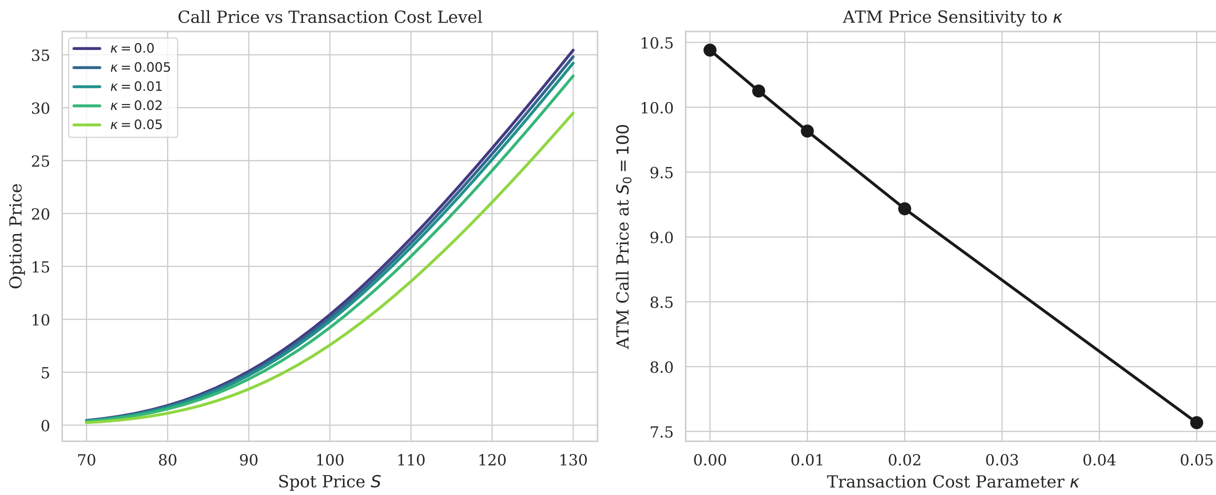

Figure 5: Transaction cost sensitivity. Left: price profiles for varying κ. Right: ATM price as a function of κ.

[4] Yaacov Z Bergman. Option pricing with differential interest rates. *The Review of Financial Studies*, 8(2):475–500, 1995.

[5] Maxim Bichuch, Agostino Capponi, and Stephan Sturm. Arbitrage-free XVA. *Mathematical Finance*, 28(2):582–620, 2018.

[6] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[7] Bruno Bouchard and Nizar Touzi. Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications*, 111(2): 175–206, 2004.

[8] Stéphane Crépey, Tomasz R Bielecki, and Damiano Brigo. *Counterparty Risk and Funding: A Tale of Two Puzzles*. Chapman and Hall/CRC, 2014.

[9] Weinan E, Jiequn Han, and Arnulf Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

[10] Nicole El Karoui, Shige Peng, and Marie-Claire Quenez. Backward stochastic differential equations in finance. *Mathematical Finance*, 7(1):1–71, 1997.

[11] Emmanuel Gobet, Jean-Philippe Lemor, and Xavier Warin. A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability*, 15(3):2172–2202, 2005.

[12] Jiequn Han and Jihao Long. Convergence of the deep BSDE method for coupled FBSDEs. *Probability, Uncertainty and Quantitative Risk*, 5(1):1–33, 2020.

[13] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34): 8505–8510, 2018.

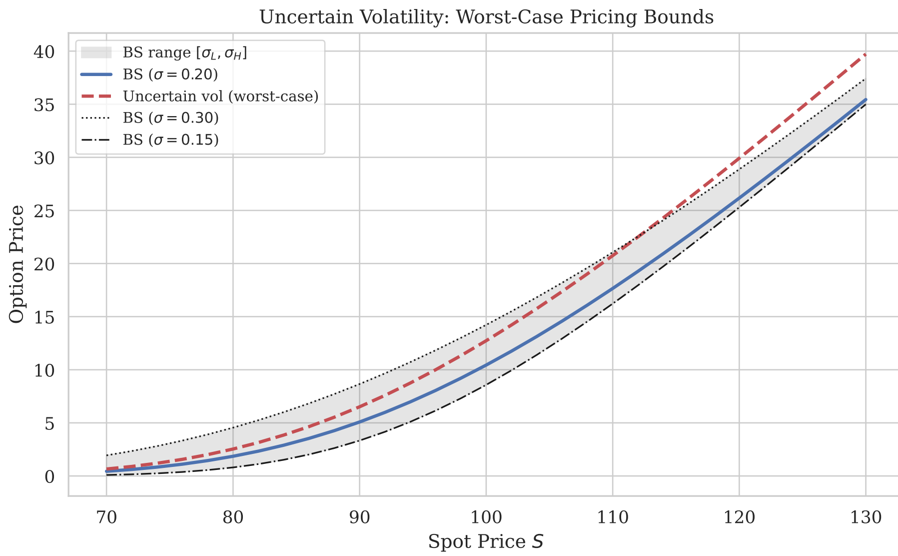

Figure 6: Uncertain volatility: the worst-case price lies near the upper BS bound, reflecting positive convexity of the call payoff.

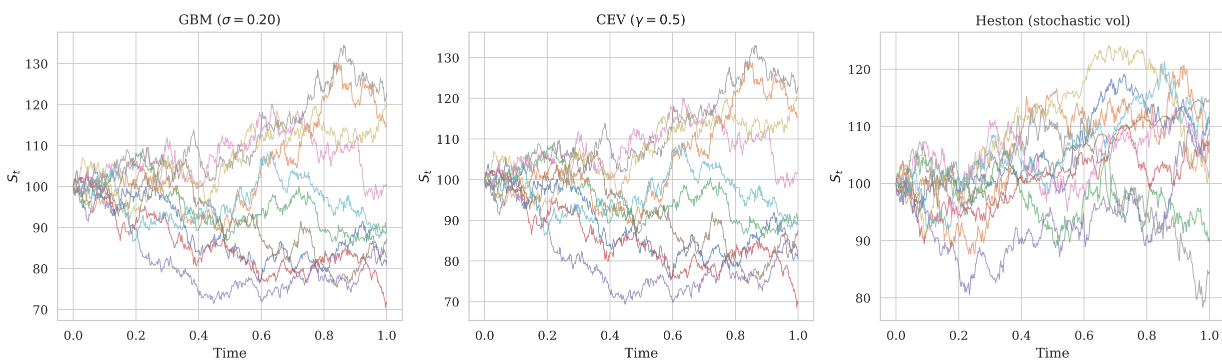

Figure 7: Forward SDE sample paths: GBM (left), CEV with $\gamma = 0.5$ (center), Heston (right).

[14] Cóme Huré, Huyên Pham, and Xavier Warin. Deep backward schemes for high-dimensional nonlinear PDEs. *Mathematics of Computation*, 89(324):1547–1580, 2020.

[15] Hayne E Leland. Option pricing and replication with transactions costs. *The Journal of Finance*, 40(5):1283–1301, 1985.

[16] Francis A Longstaff and Eduardo S Schwartz. Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.

[17] Étienne Pardoux and Shige Peng. Adapted solution of a backward stochastic differential equation. *Systems & Control Letters*, 14(1):55–61, 1990.

[18] Shige Peng. Nonlinear expectations, nonlinear evaluations and risk measures. *Stochastic Methods in Finance*, pages 165–253, 2004.

[19] Shige Peng. *Nonlinear Expectations and Stochastic Calculus under Uncertainty*, volume 95 of *Probability Theory and Stochastic Modelling*. Springer, 2019.

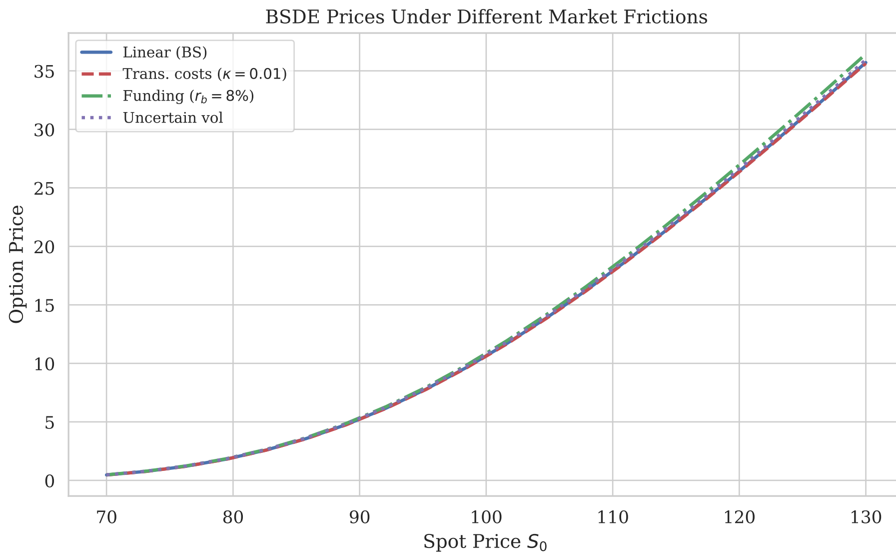

Figure 8: BSDE prices under four drivers, confirming consistency with PDE results (Figure 2).

[20] H Mete Soner, Nizar Touzi, and Jianfeng Zhang. Wellposedness of second order backward SDEs. *Probability Theory and Related Fields*, 153(1–2):149–190, 2012.

[21] Jianfeng Zhang. A numerical scheme for BSDEs. *The Annals of Applied Probability*, 14(1): 459–488, 2004.

[22] Jianfeng Zhang. *Backward Stochastic Differential Equations: From Linear to Fully Nonlinear Theory*, volume 86 of *Probability Theory and Stochastic Modelling*. Springer, 2017.

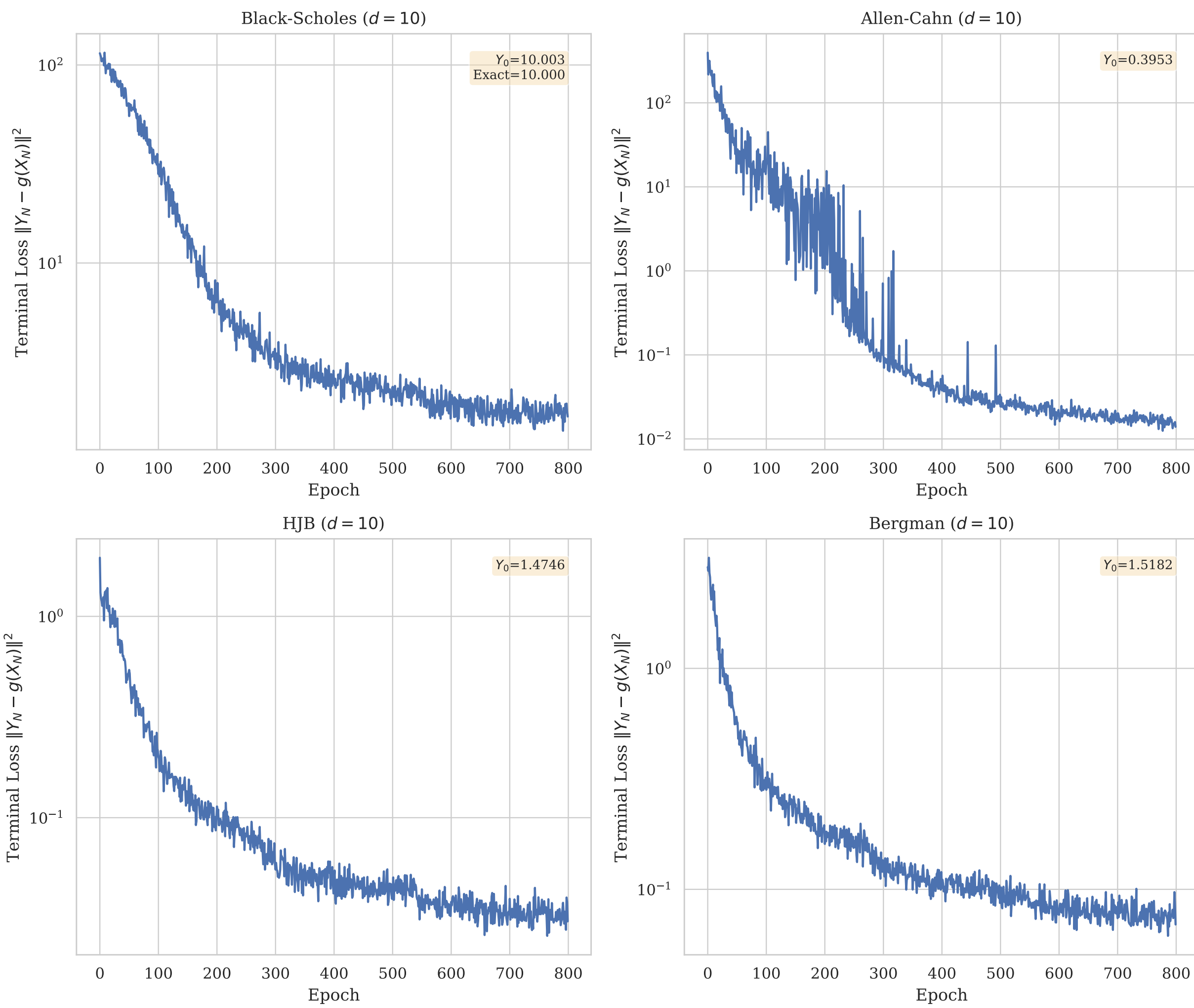

Figure 9: Deep BSDE training loss curves for four benchmarks at $d = 10$. All problems show monotone loss decrease. Annotations show final Y_0 and exact values where available.

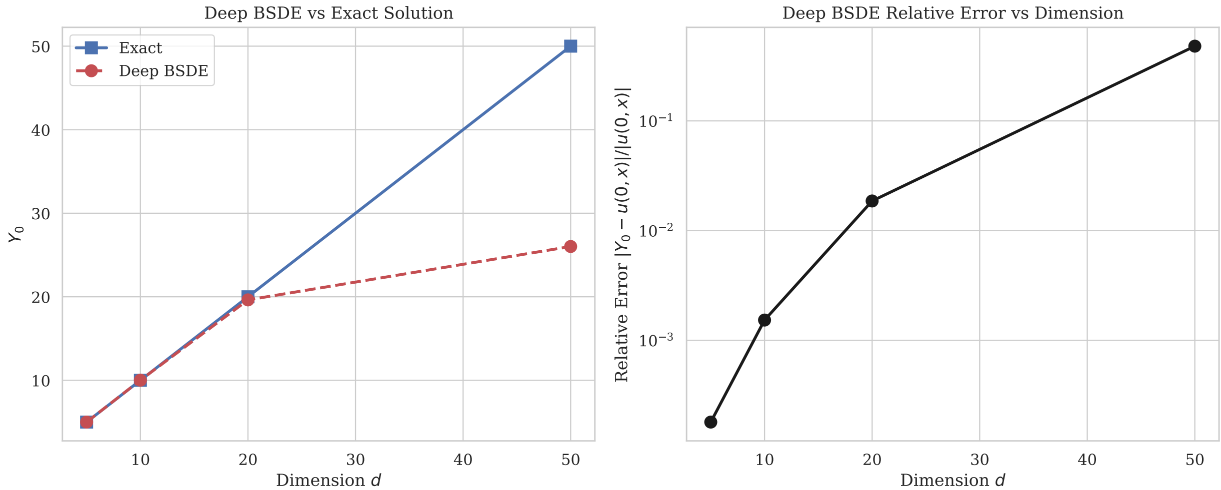

Figure 10: Dimension scaling for the Black–Scholes benchmark. Left: Y_0 vs exact value. Right: relative error on log scale. Sub-percent accuracy at $d \leq 10$; higher dimensions require more training.

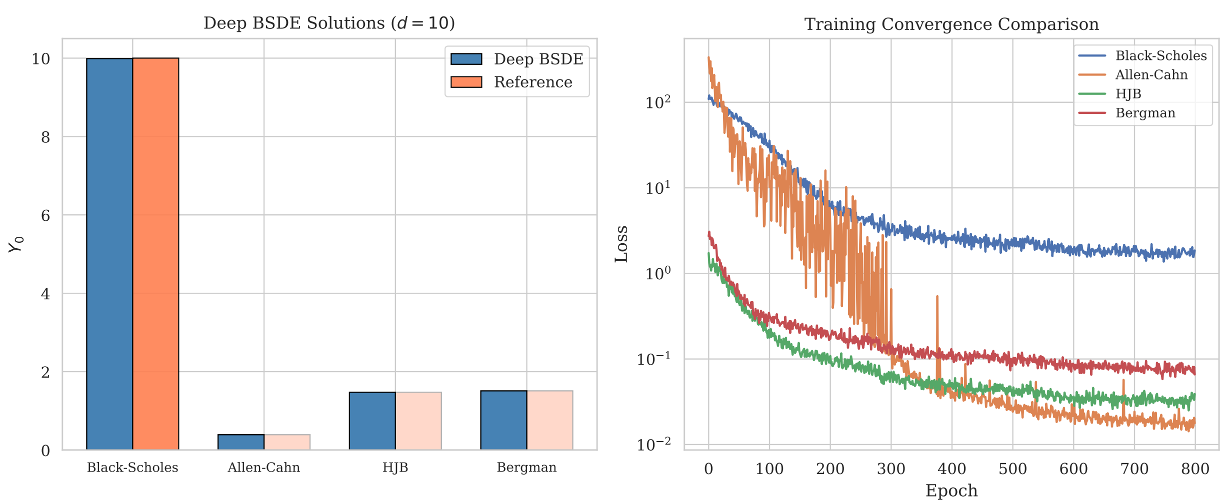

Figure 11: Deep BSDE solutions at $d = 10$. Left: Y_0 values (blue: deep BSDE; coral: reference where available). Right: training convergence for all benchmarks.

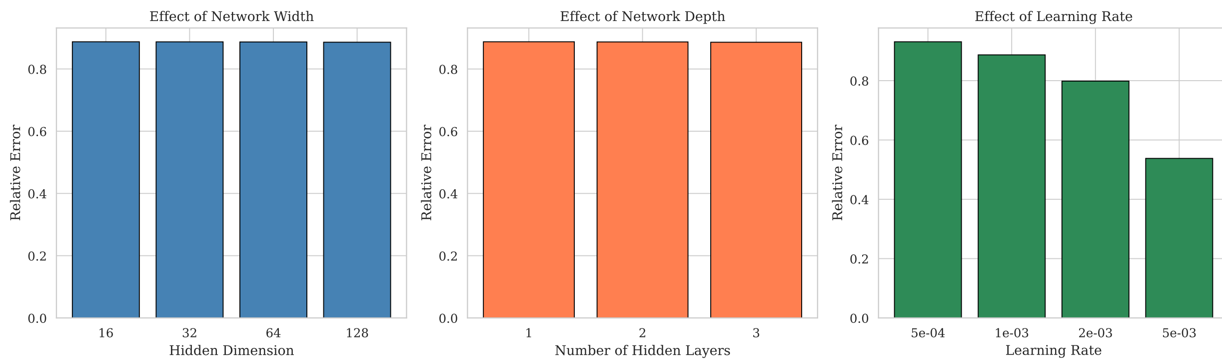

Figure 12: Ablation on the Black–Scholes benchmark ($d = 10$, 300 epochs). Left: effect of hidden dimension. Center: effect of depth. Right: effect of learning rate. All configurations are under-converged; the learning rate effect dominates.

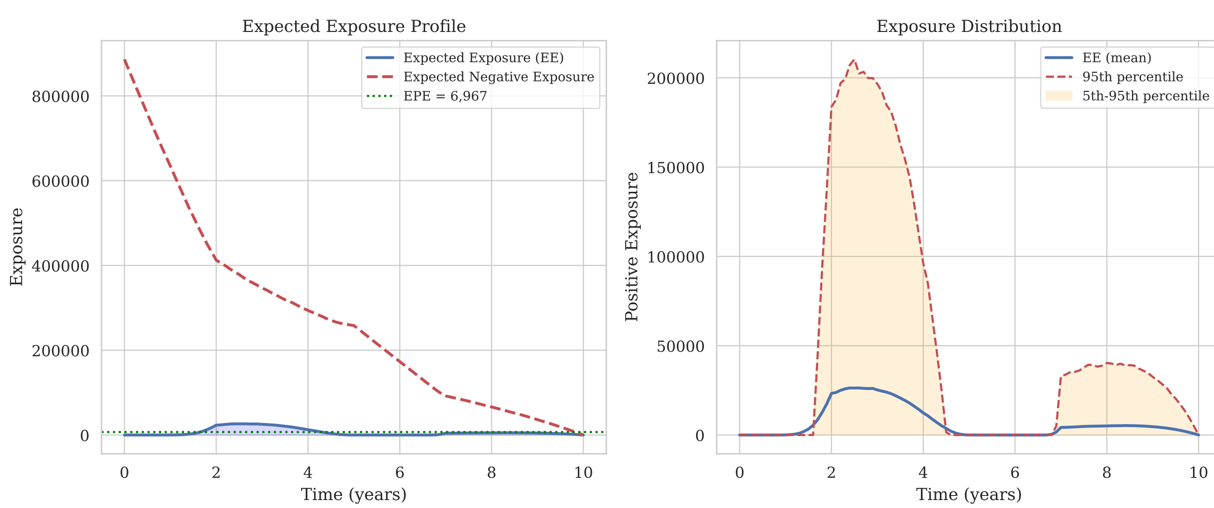

Figure 13: Exposure profiles for the 7-swap netting set. Left: expected exposure (EE) and expected negative exposure. Right: EE with 5th–95th percentile band.

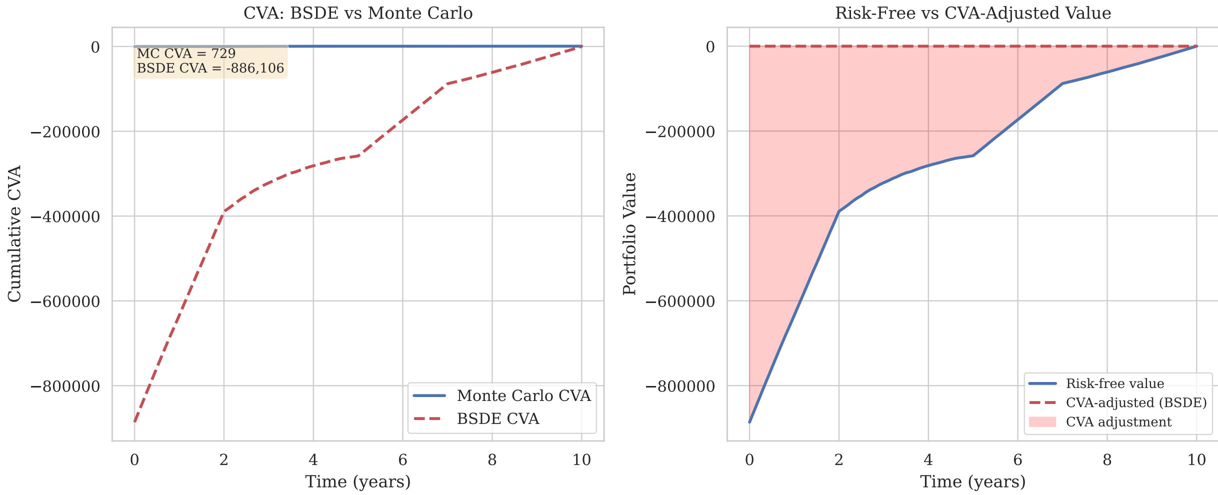

Figure 14: CVA: BSDE vs Monte Carlo. Left: cumulative CVA profiles. Right: risk-free vs CVA-adjusted portfolio value (mean paths).

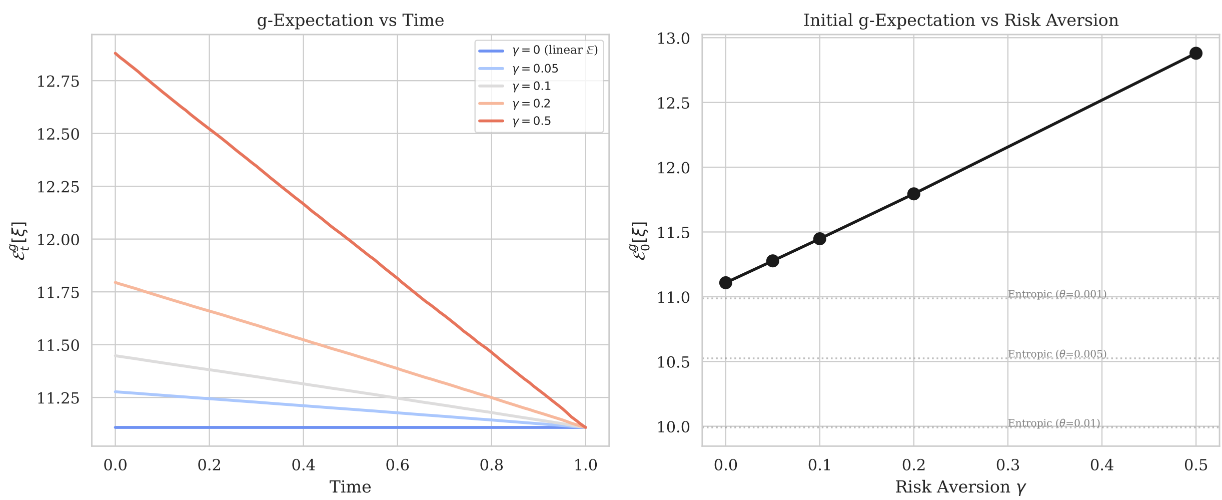

Figure 15: g-expectation analysis. Left: $\mathcal{E}_t^g[\xi]$ over time for different γ. Right: $\mathcal{E}_0^g[\xi]$ as a function of γ, with entropic risk measure comparisons (gray lines).

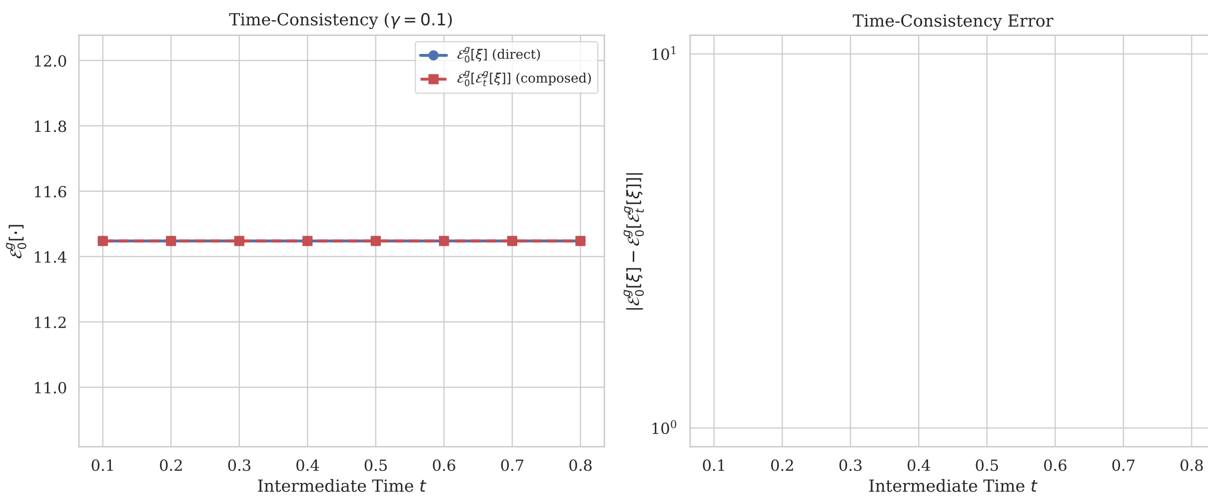

Figure 16: Time-consistency verification for $\gamma = 0.1$. Left: direct vs composed g-expectation at $t = 0$ for varying intermediate times. Right: absolute error between the two computations.