CVA sensitivities, hedging and risk

Stéphane Crépey, Bouazza Saadeddine, Botao Li and Hoang Nguyen present a framework for computing credit valuation adjustment (CVA) sensitivities, hedging the CVA and assessing CVA risk, using probabilistic machine learning as a refined regression tool applied to simulated data, which can be validated by low-cost companion Monte Carlo procedures. They identify the sensitivities representing the best practical trade-offs in downstream tasks, including CVA hedging and risk assessment.

This work illustrates the potential of probabilistic machine learning for pricing and Greeking applications, in the challenging context of credit valuation adjustment (CVA) computations.[1] By probabilistic machine learning we mean machine learning used as a refined regression tool applied to simulated data. Probabilistic machine learning for CVA pricing was introduced in Abbas-Turki *et al* (2023). Here we extend the approach to encompass CVA sensitivities and risk. The fact that probabilistic machine learning is performed on simulated data, which can be augmented at will, does not mean that there are no related data issues. As always with machine learning, the quality of the data is the first driver of the success of the approach. The variance of the training loss may be high and jeopardise the potential of a learning approach. This was first encountered in the granular CVA default pricing setup of Abbas-Turki *et al* (2023) due to the scarcity of the default events compared with the diffusive scale of the risk factors in the model. Switching from prices to sensitivities in the current paper is another case of increased variance. But with probabilistic machine learning we can also develop suitable variance-reduction tools: namely, oversimulation of defaults in Abbas-Turki *et al* (2023) and common random numbers in this work. Another distinguishing feature of probabilistic machine learning, key for regulated banking applications, is the possibility of assessing the quality of a predictor by means of low-cost companion Monte Carlo procedures.

All the equations in this paper are written using the risk-free asset as a numeraire and stated under the financial risk insurance probability measure advocated in Albanese *et al* (2021, remark 2.3) for valuation adjustment (XVA) computations, with the related expectation operator denoted below by $\mathbb{E}$. Unless explicitly stated, we include a ridge-regularisation term to stabilise trainings and regressions.

Fast bump sensitivities

Consider a time-0 option price $\Pi_0(\rho) = \mathbb{E}\xi(\rho)$, where the payout $\xi(\rho) \equiv \xi(\rho;\omega)$ depends on constant model parameters ρ and (implicitly in the shorthand notation $\xi(\rho)$) on the randomness ω of the stochastic drivers of the model risk factors with respect to which the expectation is taken. The model parameters ρ encompass the initial values of the risk factors of the pricing model, as well as all the exogenous (constant, in principle) model parameters (eg, the value of the volatility in a Black-Scholes model). For each constant ρ, the price $\Pi_0(\rho)$ can be estimated by Monte Carlo simulation. Our problem in this section is the estimation of the corresponding sensitivities $\partial_\rho \Pi_0(\rho_0)$, at a baseline value $\rho = \rho_0$ (which, in practice, would be calibrated) of the model parameters. Such sensitivities lie at the core of any related hedging scheme for the option. They are also key in many regulatory capital formulas.

Monte Carlo estimation of sensitivities in finance is divided into three main streams:

(i) differentiation of the density of the underlying process via integration by parts or more general Malliavin calculus techniques, assuming some regularity of this process;

(ii) cashflow differentiation, assuming their differentiability; and

(iii) Monte Carlo finite differences, biased but generic, which are the Monte Carlo version of the industry-standard bump sensitivities.

Approach (i), however, suffers from intrinsic variance issues. With contemporary technology, approach (ii) can take advantage of adjoint algorithmic differentiation (AAD). But AAD can soon lead to significant implementation and memory costs when used for complex pricing problems at the portfolio level, such as CVA computations (see Capriotti *et al* 2017). Such an AAD Greeking approach becomes nearly unfeasible in the case of pricing problems that embed numerical optimisation subroutines, as with the training of the conditional risk measures embedded in the refined CVA and higher-order XVA metrics of Albanese *et al* (2021).

■ **Common random numbers.** Under approach (iii), bump sensitivities are computed by relaunching the Monte Carlo pricing engine with common random numbers ω for values bumped typically by $\pm 1\%$ (in relative terms) of each risk factor and/or model parameter of interest, and then taking the accordingly normalised difference between the corresponding $\Pi_0(\rho)$ and $\Pi_0(\bar\rho)$, where the $\bar{\cdot}$ denotes symmetrisation with respect to ρ_0, so:

$$\frac{\rho + \bar\rho}{2} = \rho_0, \quad \text{ie, } \bar\rho = 2\rho_0 - \rho \tag{1}$$

This approach requires two Monte Carlo simulation runs (with common drivers ω) of m paths per sensitivity, making it a robust but heavy procedure, dubbed the 'benchmark' bump sensitivity approach. We also compute 'smart' bump sensitivities, which are similar but use only m/p paths each, where $p = |\rho|$. Hence the computing time for all the smart bump sensitivities is of the same order of magnitude as the time for pricing $\Pi_0(\rho_0)$ by Monte Carlo simulation with m paths. More precisely, each smart bump sensitivity uses m/p paths of a Monte Carlo simulation run with m paths overall. This is significantly more efficient than doing p Monte Carlo runs of size m/p each, especially in the graphics processing unit simulation environment of our CVA computations below.

As another way to accelerate the bump sensitivities with common random numbers as in (iii), a useful trick is to introduce $\varsigma(\rho;\omega) := \xi(\rho;\omega) - \xi(\bar\rho;\omega)$ (cf (1)). We can then learn the sensitivity function (in the sense of finite differences) $\rho \mapsto \Sigma_0(\rho) := \mathbb{E}\varsigma(\rho)$, which, by linearity and the chain rule (as $\bar\rho = 2\rho_0 - \rho$), satisfies $\Sigma_0'(\rho) = \Pi_0'(\rho) + \Pi_0'(\bar\rho)$, and in particular

[1] *A longer, preprint version of this work is available on arXiv. The* Python *code for this paper is available from https://github.com/botaoli/CVA-sensitivities-hedging-risk.*

$\partial_\rho \Sigma_0(\rho_0) = 2\partial_\rho \Pi_0(\rho_0)$. For learning the function $\Sigma_0(\cdot)$ locally around ρ_0, with ϱ randomising ρ as above, we rely on the representation $\Sigma_0(\rho) = \mathbb{E}(\varsigma(\varrho) \mid \varrho = \rho)$. That is:

$$\Sigma_0(\cdot) = \arg\min_{\Phi \in \mathcal{B}} \mathbb{E}[(\varsigma(\varrho) - \Phi(\varrho))^2] \tag{2}$$

where $\mathcal{B}$ denotes the Borel-measurable functions of ρ. Then, in the optimisation problem (2), we replace $\mathcal{B}$ by a linear hypothesis space $\Sigma_0^\theta(\rho) = \theta^\top(\rho - \rho_0)$ (noting that $\Sigma_0(\rho_0) = 0$) and $\mathbb{E}$ by a simulated sample mean $\hat{\mathbb{E}}$, with each draw of ϱ followed by one draw of ω that is implicit in $\varsigma(\varrho)$. This results in a linear least-squares problem for the weights θ, solved by regression. The estimated weights $\theta/2$ are our 'linear' bump sensitivity estimate for:

$$\tfrac{1}{2}\partial_\rho \Sigma_0(\rho_0) = \partial_\rho \Pi_0(\rho_0)$$

For simple parametric distributions of ϱ, the covariance matrix that appears in the regression is known and invertible in closed form, which reduces this regression (implemented without ridge regularisation in this analytical case) to a standard Monte Carlo simulation endowed with the associated confidence intervals, for a computation time comparable to that of one of the smart bump sensitivities.

The sensitivities $\partial_\rho \Pi_0(\rho_0)$ are sensitivities to the model parameters. Practical hedging schemes require sensitivities to the calibrated prices of hedging instruments. Hence, for hedging purposes, our sensitivities must be mapped to hedging ratios in market instruments. This is done via the implicit function theorem, in the way explained in Antonov *et al* (2018).

CVA and its bump sensitivities

Since we are also interested in the risk of CVA fluctuations, we now consider the targeted price Π (CVA from now on) as a process. We let MtM^c denote the counterparty-risk-free valuation of the bank's portfolio with its client c; τ_c denote the client c's default time, with intensity process γ^c; X denote the vector of the default indicator processes of the clients of the bank, with $X_0 \equiv 0$ (component-wise); and Y denote a diffusive vector process of model risk factors, such that each MtM_t^c and γ_t^c are measurable functions of (t, Y_t) for $t \geqslant 0$ (in the case of credit derivatives with the client c, MtM_s^c would also depend on X_t, which can be accommodated at no harm in our setup). The exogenous model parameters are denoted by ϵ. Let the baseline $\rho_0 = (y_0, \epsilon_0)$ denote a calibrated value of (y, ϵ), where y is used to refer to the initial condition of Y, whenever it is assumed constant. Let ι denote an initial condition for Y randomised around its baseline y_0, ε likewise be a randomisation of ϵ around its baseline ϵ_0, and $\varrho_t = (Y_t, \varepsilon)$, $t \geqslant 0$. Starting from the (random) initial condition $(0, \iota)$, the model (X, Y) is supposed to evolve according to some Markovian dynamics (eg, those of our 'CVA lab' in the next section) parameterised by ε. This setup encompasses in a common formalism:

- the 'baseline mode' of Abbas-Turki *et al* (2023, section 4), where $\varrho_0 \equiv \rho_0$;
- the 'risk mode', where $Y_0 \equiv y_0$;
- the 'sensis mode', or general ϱ_0 case (used only for $t = 0$).

The CVA engine in the baseline mode $\varrho_0 \equiv \rho_0$ was introduced in Abbas-Turki *et al* (2023). The risk and sensis modes – which additionally incorporate a randomisation ε of the exogenous model parameters ϵ, and also a randomisation of Y_0 in the sensis mode – are the novelties of our work.

For notational simplicity, we restrict our analysis to an uncollateralised CVA. Given n pricing time steps of length h such that $nh = T$, the final maturity of the derivatives portfolio of the bank, at each $t = ih$ let:

$$\left.\begin{aligned} \mathrm{LGD}_t &= \sum_c \sum_{j=0}^{i-1} (\mathrm{MtM}_{jh}^c)^+ \mathbf{1}_{jh < \tau_c \leqslant (j+1)h} \\ \xi_{t,T} &= h \sum_c \sum_{j=i}^{n-1} (\mathrm{MtM}_{jh}^c)^+ (\mathrm{e}^{-\sum_{\iota=i}^{j-1} \gamma_\iota^c} - \mathrm{e}^{-\sum_{\iota=i}^{j} \gamma_\iota^c}) \mathbf{1}_{\{\tau_c > ih\}} \end{aligned}\right\} \tag{3}$$

Our computations rely on the following default- and intensity-based formulations of the (time-discretised) CVA of a bank with client c, at the pricing time $t = ih$ (cf Abbas-Turki *et al* 2023, equations (25)–(27)):

$$\begin{aligned} &\mathrm{CVA}_t(x, \rho) \\ &= \mathbb{E}\Bigg[\underbrace{\sum_c \sum_{j=i}^{n-1} (\mathrm{MtM}_{jh}^c)^+ \mathbf{1}_{jh < \tau_c \leqslant (j+1)h}}_{\mathrm{LGD}_T - \mathrm{LGD}_t} \,\Bigg|\, X_{ih} = x,\ \varrho_{ih} = \rho \Bigg] \\ &= \mathbb{E}\Bigg[\underbrace{h \sum_c \sum_{j=i}^{n-1} (\mathrm{MtM}_{jh}^c)^+ (\mathrm{e}^{-\sum_{\iota=i}^{j-1} \gamma_\iota^c} - \mathrm{e}^{-\sum_{\iota=i}^{j} \gamma_\iota^c}) \mathbf{1}_{\{\tau_c > ih\}}}_{\xi_{t,T}} \\ &\qquad\qquad \Bigg|\, X_{ih} = x,\ \varrho_{ih} = \rho \Bigg] \end{aligned} \tag{4}$$

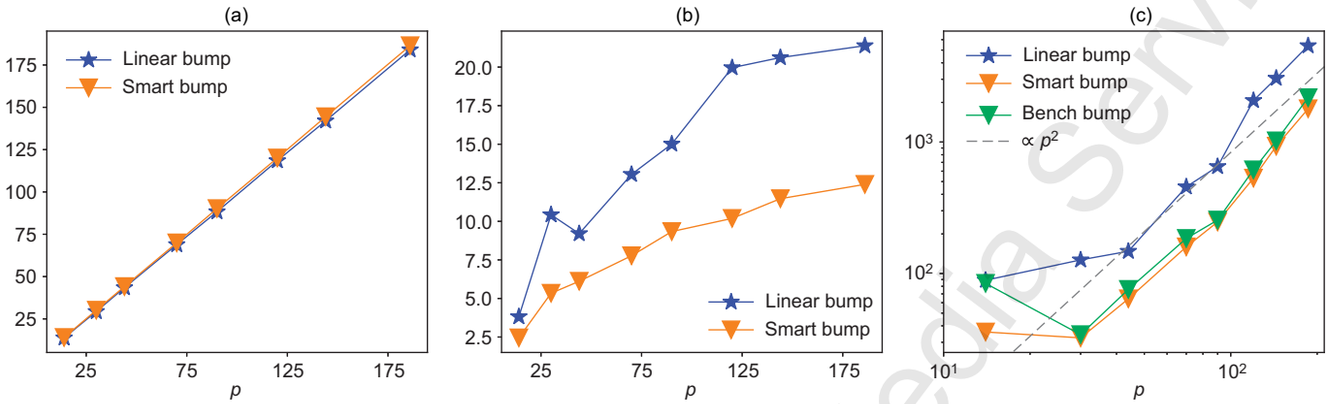
where each coordinate of x is 0 or 1. From a numerical viewpoint, the second equality of (4) entails less variance than the first one (see figure 5 in Abbas-Turki *et al* (2023)). Hence we rely for our CVA computations on this second equality. At the initial time (time 0), as $X_0 \equiv 0$, we can restrict our attention to the origin $x = 0$ and skip the argument x as well as the conditioning by $X_{ih} = x$ in (4). In the sensis mode, we are in setup (2) for $\xi = \xi_{0,T}$, which depends implicitly on $\varrho_0 = (\iota, \varepsilon)$; that is, $\mathrm{CVA}_0(\varrho_0) = \mathbb{E}(\xi(\varrho_0) \mid \varrho_0)$. CVA fast bump sensitivities can thus be computed in the ways described in the first section of this paper, with CVA and $(y = Y_0, \epsilon)$ here taking the roles of Π and ρ therein. In the baseline mode, where $\varrho_0 \equiv \rho_0$, $\mathrm{CVA}_0(\varrho_0)$ is constant, equal to the corresponding:

$$\mathrm{CVA}_0(\rho_0) = \mathbb{E}\,\mathrm{LGD}_{nh} = \mathbb{E}\xi_{0,T} \tag{5}$$

which is computed by Monte Carlo simulation based on the second equality of (4) with $t = 0$, as a sample mean of $\xi_{0,T}$, along with the corresponding 95% confidence interval.

CVA lab. In our numerics, we have 10 economies. For each of them we have a short-term interest rate driven by a Vasicek diffusion and, except for the reference economy, a Black-Scholes exchange rate with respect to the currency of the reference economy. The reference bank has eight counterparties with corresponding default intensity processes driven by Cox-Ingersoll-Ross diffusions. We thus have eight default indicator processes X for the counterparties and $10 + 9 + 8 = 27$ diffusive risk factors Y. This results in a Markovian model (X, Y) of dimension 35, entailing $p = 90$ parameters corresponding to the 27 initial conditions of the Y processes plus their 63 exogenous parameters.

1 Comparison of the performance of linear and smart bump sensitivities as a function of the number of model parameters p, with m fixed at 2^{17} and benchmark bump sensitivities used as references



The speed-ups displayed in part (a) are obtained by dividing the execution times for benchmark sensitivities by the execution times for linear or smart bump sensitivities. For each value of p, the error ratio in the middle panel denotes the median of the p ratios between the Monte Carlo standard errors of the linear (respectively smart) bump sensitivity and of the benchmark sensitivity. The complexities in the right panels are defined as the projected times of reaching a relative standard error of 1% , building on a $1/\sqrt{|\text{sample size}|}$ scaling of the standard errors justified by the central limit theorem, with $|\text{sample size}| = m$ for linear and benchmark bumps and m/p for smart bumps

A 'reasonable' but arbitrary baseline ρ_0 plays the role of calibrated model parameters in our numerics. The portfolio consists of 500 interest rate swaps with random characteristics (maturity up to $T = 10$ years, notional, currency and counterparty) and strikes such that the swaps have zero worth in the baseline model (ie, for $\varrho_t \equiv \rho_0$) at time 0. The swaps have analytic counterparty-risk-free valuation in our pricing model. Their price processes are converted into the reference currency and aggregated into the corresponding client's MtM^c processes.

We simulate by an Euler scheme $m = 2^{17} \approx 1.3 \times 10^5$ paths of the pricing model (X, Y), with $n = 100$ mark-to-market pricing time steps of length $h = 0.1$ and 25 Euler simulation sub-steps per pricing time step. A Monte Carlo computation of (5) in the baseline mode yields $\mathrm{CVA}_0(\rho_0) \in 5{,}027 \pm 18$ with 95% probability (computed in about 30 seconds). A randomisation of Y_0 and ϵ in the sensis mode is used for deriving CVA linear bump sensitivities in the way explained below (2). In figure 1, the increasing curves in part (a) highlight the almost linear growth in the speed-up of the linear and smart bump sensitivities with respect to the benchmark ones when the number p of pricing model parameters increases, but part (b) also shows that this comes with increasing errors. The smart bump sensitivities have smaller errors than the linear bump sensitivities for all tested p. The superiority of the smart bump sensitivities over the linear ones is also observed in figure 1(c), where for all tested p the complexity of the linear bump sensitivities is higher than that of the smart bump sensitivities, itself very close (as expected) to that of the benchmark bump sensitivities.

We specify $q = 256$ market instruments corresponding to zero-coupons, credit defaults swaps and foreign exchange forwards of various maturities in the different economies and with the different counterparties of the bank in the model. The model sensitivities are then converted into market sensitivities by the implicit function theorem.

■ **Learning the future CVA.** Equivalently to the second equality in (4):

$$\mathrm{CVA}_t(\cdot) = \operatorname*{arg\,min}_{\Phi \in \mathcal{B}} \mathbb{E}[(\xi_{t,T} - \Phi(X_t, \varrho_t))^2] \tag{6}$$

where $\mathcal{B}$ is the set of the Borel-measurable functions of (x, ρ). We denote by $\mathrm{CVA}_t^\theta(X_t, \varrho_t)$ the conditional CVA at time $t = ih > 0$ learned by a neural network with parameters θ on the basis of simulated pairs (X_t, ϱ_t) and cashflows $\xi_{t,T}$. The conditional CVA pricing function $\mathrm{CVA}_t^\theta(x, \rho)$ is obtained by replacing $\mathbb{E}$ by a simulated sample mean $\hat{\mathbb{E}}$ and $\mathcal{B}$ by a neural net search space (or linear search space as a special case) in the optimisation problem (6). The latter is then addressed numerically by Adam gradient descent on the basis of simulated pairs (X_t, ϱ_t) as features and $\xi_{t,T}$ as labels.

A key asset of probabilistic machine learning procedures for any conditional expectation, such as $\Pi_0(\varrho)$ in the first section of this paper (or $\mathrm{CVA}_t(X_t, \varrho_t)$ above), is the availability of the companion 'twin Monte Carlo validation procedure' of Abbas-Turki *et al* (2023, section 2.4), which allows the accuracy of a predictor to be assessed. Let $\xi^{(1)}(\varrho)$ and $\xi^{(2)}(\varrho)$ denote two independent copies of $\xi(\varrho)$ given ϱ, such that:

$$\mathbb{E}[f(\xi^{(1)}(\varrho))g(\xi^{(2)}(\varrho)) \mid \varrho] = \mathbb{E}[f(\xi^{(1)}(\varrho)) \mid \varrho]\mathbb{E}[g(\xi^{(2)}(\varrho)) \mid \varrho]$$

holds for any Borel-bounded functions f and g. The twin Monte Carlo validation procedure for a predictor $\Phi(\varrho)$ of $\Pi_0(\varrho) = \mathbb{E}(\xi(\varrho) \mid \varrho)$ consists in computing the Monte Carlo estimates' twin statistic (twin-stat) for:

$$\mathbb{E}[\Phi(\varrho)^2 - (\xi^{(1)}(\varrho) + \xi^{(2)}(\varrho))\Phi(\varrho) + \xi^{(1)}(\varrho)\xi^{(2)}(\varrho)] = \mathbb{E}[(\Phi(\varrho) - \mathbb{E}(\xi(\varrho) \mid \varrho))^2] \tag{7}$$

(as follows from the tower rule by conditional independence) and twin variance (twin-var) for:

$$\mathrm{Var}[\Phi(\varrho)^2 - (\xi^{(1)}(\varrho) + \xi^{(2)}(\varrho))\Phi(\varrho) + \xi^{(1)}(\varrho)\xi^{(2)}(\varrho)]$$

The ensuing 95% confidence level upper bound on:

$$\sqrt{\mathbb{E}[(\Phi(\varrho) - \mathbb{E}(\xi(\varrho) \mid \varrho))^2]}$$

(the root mean square error of the predictor $\Phi(\varrho)$) is then given by:

$$\text{twin-up} = \sqrt{\text{twin-stat} + \frac{2}{\sqrt{m}}\sqrt{\text{twin-var}}} \tag{8}$$

A. Risk measures of $\delta\text{CVA}_t^\theta$, LGD_t and their sum ($\text{CVA}_0(\rho_0) = 5{,}027$)

	$t = 0.01$			$t = 0.1$			$t = 1$		
	$\delta\text{CVA}_t^\theta$	LGD_t	$\delta\text{CVA}_t^\theta + \text{LGD}_t$	$\delta\text{CVA}_t^\theta$	LGD_t	$\delta\text{CVA}_t^\theta + \text{LGD}_t$	$\delta\text{CVA}_t^\theta$	LGD_t	$\delta\text{CVA}_t^\theta + \text{LGD}_t$
Expectation	−9	0.28	−8	56	7	63	75	502	578
VAR 99%	388	0	389	1,347	0	1,389	4,796	11,997	11,757
ES 97.5%	395	0.28	397	1,361	7	1,445	4,953	12,383	12,297

B. Risk measures of $\delta\text{CVA}_{(t)}$ computed by Monte Carlo simulation using $\delta\text{CVA}_{(t)}$ simulated by various predictors

		Non-parametric		Linear(-quadratic) in $\delta\varrho_{(t)}$			Linear(-quadratic) in $\delta Z_{(t)}$		
t	Risk measure	Nested $\delta\text{CVA}_{(t)}$	$\delta\text{CVA}_{(t)}^\theta$	Benchmark bump sensis w/ Γ	Smart bump sensis	LS sensis w/ Γ	Benchmark bump sensis	Smart bump sensis	LS sensis w/ Γ
0.01	twin-up	**11**	29	**13**	**15**	18	23	23	18
	VAR 99%	**510**	**431**	429	418	415	**435**	431	424
	ES 97.5%	**514**	**434**	432	421	417	**437**	433	427
0.1	twin-up	**59**	103	110	111	109	**101**	104	**94**
	VAR 99%	**1,686**	**1,618**	**1,562**	1,463	1,539	1,554	1,542	1,556
	ES 97.5%	**1,693**	**1,622**	**1,570**	1,467	1,544	1,559	1,545	1,561
1	twin-up	**325**	**693**	1,244	1,307	1,113	932	943	**743**
	VAR 99%	**10,333**	**9,887**	8,991	6,646	8,812	7,914	7,846	**9,493**
	ES 97.5%	**10,654**	**10,090**	9,309	6,715	9,032	8,075	7,988	**9,792**

The three lowest (ie, best) twin upper bounds (8) and the three highest (ie, most conservative) risk estimates on each row are emphasised in bold

CVA risk and hedging

An economic (or 'internal') view gained from simulating the movements of model or/and market risk factors and obtaining risk measures of CVA fluctuations is an important dimension of the CVA capital regulatory requirements for a bank in the context of its supervisory review and evaluation process (SREP): 'the risks the institution has identified and quantified will play an enhanced role in, for example, the determination of additional own funds requirements on a risk-by-risk basis'.[2]

■ **Run-off CVA risk.** We first consider:

$$\delta\text{CVA}_t^\theta + \text{LGD}_t, \quad \text{where } \delta\text{CVA}_t^\theta = \text{CVA}_t^\theta(X_t, \varrho_t) - \text{CVA}_0(\rho_0) \tag{9}$$

(cf (3)–(4)), assessed in the risk mode, where $\varrho_t = (Y_t(y_0, \varepsilon), \varepsilon)$ and ε follows $\mathcal{N}(\epsilon_0, \text{diag}(t(1\%)^2\epsilon_0 \odot \epsilon_0))$. The random variable $(\delta\text{CVA}_t^\theta + \text{LGD}_t)$ reflects a dynamic but also run-off view on CVA risk together with counterparty default risk, as opposed to the stationary run-on view of CVA risk below. We assess CVA risk, counterparty default risk and both risks combined on the basis of the value-at-risk and expected shortfall (ES) of $\delta\text{CVA}_t^\theta$ and LGD_t in the risk mode, reported in table A for $t = 0.01$ and 0.1 years and 1 year and quantile levels 99% for VAR and 97.5% for ES. The results in table A emphasise that CVA and counterparty default risks assessed on a run-off basis are primarily driven by client defaults, especially at higher quantile levels.

■ **Run-on CVA risk.** With $\varepsilon_{(t)} \sim \mathcal{N}(\epsilon_0, \text{diag}(t(1\%)^2\epsilon_0 \odot \epsilon_0))$, let $Y_t(y_0, \varepsilon_{(t)})$ denote the process Y at time t starting from y_0 at 0 for model parameters set to $\varepsilon_{(t)}$, let $\varrho_{(t)}$ denote $(Y_t(y_0, \varepsilon_{(t)}), \varepsilon_{(t)})$, let $Z_0(\varrho_{(t)})$ denote the time-0 price of market instruments corresponding to the model parameters $\varrho_{(t)}$ and let z_0 denote the baseline price of the market instruments at time 0. Then, let:

$$\left.\begin{aligned} \delta\varrho_{(t)} &= (Y_t(y_0, \varepsilon_{(t)}) - y_0, \varepsilon_{(t)} - \epsilon_0) = \varrho_{(t)} - \rho_0 \\ \delta Z_{(t)} &= Z_0(\varrho_{(t)}) - z_0 \\ \delta\text{CVA}_{(t)} &= \text{CVA}_0(\varrho_{(t)}) - \text{CVA}_0(\rho_0) \end{aligned}\right\} \tag{10}$$

The fact that we consider here the time-0 $\text{CVA}_0(\varrho_{(t)})$ (see the discussion below (4)) and likewise $Z_0(\varrho_{(t)})$ is in line with an assessment of risk on a run-on portfolio and customer basis and with a siloing of CVA versus counterparty default risk, which have both become standard in regulation and market practice. Various predictors of $\delta\text{CVA}_{(t)}$ can be learned directly from the simulated model parameters $\varrho_{(t)}$ and cashflows $\xi_{0,T}(\varrho_{(t)}) - \xi_{0,T}(\rho_0)$ (which is better than learning $\delta\text{CVA}_{(t)}$ via $\text{CVA}_0(\varrho_{(t)})$, as this would involve more variance): a nested Monte Carlo estimator; a neural net regressor $\delta\text{CVA}_{(t)}^\theta$; and a linear(-diagonal quadratic) regressor against $\delta\varrho_{(t)}$ or $\delta Z_{(t)}$, referred to as the LS (for 'least squares') regressor below. The neural network used for training $\delta\text{CVA}_{(t)}^\theta$ on the basis of simulated data $(\varrho_{(t)} - \rho_0, \xi_{0,T}(\varrho_{(t)}; \omega) - \xi_{0,T}(\rho_0; \omega))$ has one hidden layer with 200 hidden units and softplus activation functions.

Table B displays some twin upper bounds (8) and risk measures of $\delta\text{CVA}_{(t)}$ computed with these different approximations, as well as with linear(-diagonal quadratic) Taylor expansions in $\delta\varrho_{(t)}$ or $\delta Z_{(t)}$ with coefficients estimated as benchmark or smart bump sensitivities. In terms of the twin upper bounds, the nested CVA has the best accuracy, but (for a given risk horizon t) it takes about two hours (with 1,024 inner paths) versus about one minute of simulation time for generating the labels plus 30 seconds for training by neural networks and 2–3 seconds for LS regression. The neural network excels at large t, for which the nonlinearity becomes significant, while being outperformed by the linear methods at small t, for which $\delta\text{CVA}_{(t)}$ is approximately linear. With diagonal gamma (Γ) elements taken

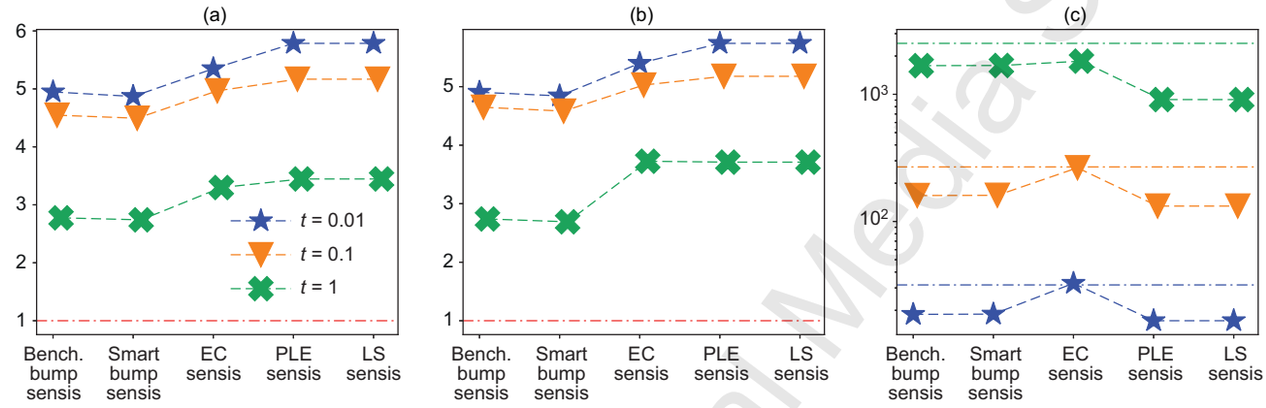
[2] *URL: https://bit.ly/4cL2DPh.*

C. Computation times for learning $\delta\mathrm{CVA}^{\theta}_{(t)}$ and the related sensitivities

[Smart] bump sensitivities		Optimised sensitivities					Speed-up		
					PnL regression				
Model sensis	Jacobian transforms	MtM simulation	LS	$\delta\mathrm{CVA}^{\theta}_{(t)}$ learning	EC	PLE	LS	EC	PLE
12m 48s [8.5s]	30s	27s	1s	6s	31s	1s	28.5 [1.4]	12.5 [0.6]	23.5 [1.1]

The speed-ups measure the ratios of the total time taken by the benchmark bump sensitivity approach to the total time taken by each of the sensitivities

2 Compression ratios of the standard error of (a) UPL and (b) EC, and (c) HVA trends c of various $\delta\mathrm{CVA}^{\theta}_{(t)}$ hedging approaches



Horizontal dash-dot lines correspond to the unhedged case with $\Delta = 0$ in (11)

into account, the performance of the LS regressor improves for large t. The linear quadratic Taylor expansions show relatively good twin upper bounds for small t, but they worsen for large t.

Regarding the risk measures, the nested $\delta\mathrm{CVA}_{(t)}$ and the neural network $\delta\mathrm{CVA}^{\theta}_{(t)}$ provide more conservative VAR and ES estimates than any linear(-quadratic) proxy in all cases. Surprisingly, even though $\mathrm{CVA}_0(\varrho_{(t)})$ in $\delta\mathrm{CVA}_{(t)}$ is a function of $\varrho_{(t)}$, for $t = 1$ the linear(-quadratic) proxies in $\delta Z_{(t)}$ outperform those in $\varrho_{(t)}$ in terms of both twin error and the consistency of the ensuing risk measures with those provided by the non-parametric (neural net and nested) references. Also note that, when compared again with the non-parametric approaches, the smart bump sensitivity proxy in $\delta Z_{(t)}$ yields results almost as good as those of the much slower benchmark bump sensitivity proxy in $\delta Z_{(t)}$.

■ **Run-on CVA hedging.** We define loss $L^{\theta}_{(t)}$ as the following hedged profit-and-loss (PnL) of the CVA desk over the risk horizon t (cf (10)):

$$L^{\theta}_{(t)} = \delta\mathrm{CVA}^{\theta}_{(t)} - (\delta Z_{(t)})^{\top}\Delta - c \tag{11}$$

where the hedging ratio $\Delta \in \mathbb{R}^q$ is treated as a free parameter, while the real constant c is deduced from Δ through the constraint that $\mathbb{E}L^{\theta}_{(t)} = 0$ (or $\hat{\mathbb{E}}L^{\theta}_{(t)} = 0$ in the numerics). The constant c, which is equal to 0 (modulo the numerical noise) in the baseline mode, where CVA + LGD and Z + CF are both martingales, can be interpreted in terms of a hedging valuation adjustment (HVA) in the spirit of Albanese *et al* (2023) (ie, as a provision for model risk). Albanese *et al* develop how, such a provision having been updated in a first stage, the loss $L^{\theta}_{(t)}$, thus centered via the 'HVA trend' c (ie, $c = -\mathbb{E}\delta\mathrm{HVA}_{(t)} = \mathrm{HVA}_0 - \mathbb{E}\mathrm{HVA}_{(t)}$, where $\mathrm{HVA}_{(t)}$ is the HVA analogue of $\mathrm{CVA}_{(t)}$), deserves an amount of economic capital, which we quantify below as an ES of $L^{\theta}_{(t)}$. We define economic capital (EC) and PnL explain (PLE) run-on sensitivities as (cf Rockafellar & Uryasev 2000):

$$\left.\begin{aligned} \Delta^{\mathrm{ec}} &= \operatorname*{arg\,min}_{\Delta\in\mathbb{R}^q} \mathbb{ES}(L^{\theta}_{(t)}) \\ (\Delta^{\mathrm{ec}}, \mathbb{VaR}(L^{\theta}_{(t)})) &= \operatorname*{arg\,min}_{\Delta\in\mathbb{R}^q, k\in\mathbb{R}} k + (1-\alpha)^{-1}\mathbb{E}[(L^{\theta}_{(t)} - k)^{+}] \\ \Delta^{\mathrm{ple}} &= \operatorname*{arg\,min}_{\Delta\in\mathbb{R}^q} \mathbb{E}[(L^{\theta}_{(t)})^2] \end{aligned}\right\} \tag{12}$$

where $\mathbb{VaR}$ and $\mathbb{ES}$ mean the 95% VAR and the corresponding ES. Once $\delta\mathrm{CVA}^{\theta}_{(t)}$ has been learned from simulated $\varrho_{(t)}$ and $\xi_{0,T}(\varrho_{(t)}) - \xi_{0,T}(\rho_0)$ in the way mentioned below (10), these sensitivities are computed as per (12), by training Δ and k in the second part of the first line (for Δ^{ec}) and by regression of Δ in the second equation (for Δ^{ple}). Simpler (but still optimised) LS run-on sensitivities are obtained without prior learning of $\delta\mathrm{CVA}_{(t)}$, just by linearly regressing $\xi_{0,T}(\varrho_{(t)}) - \xi_{0,T}(\rho_0)$ against $\delta Z_{(t)}$ (though here, due to the hedging focus of this section, this is a purely linear LS regression, as opposed to the linear-diagonal quadratic LS regression in table B). These LS sensitivities are thus obtained much like the linear bump sensitivities (see the discussion below (2)), but directly in the market (price) variables, without Jacobian transformations. The computation times of the LS, EC and PLE run-on sensitivities are reported in table C.

By unexplained PnL, UPL (respectively economic capital EC), we mean the standard error (respectively expected shortfall) of $L^{\theta}_{(t)}$. As performance metrics, we consider a backtesting, out-of-sample UPL (respectively EC with $\alpha = 95\%$) for $\Delta = 0$, divided by UPL (respectively EC with $\alpha = 95\%$) for each considered set of sensitivities: the higher the corresponding 'compression ratios', the better are the corresponding sensitivities. For each simulation run below, we use $m = 2^{17}$ paths to estimate the PLE, EC and LS sensitivities and we generate another $m = 2^{17}$ paths for our back-test.

D. Conclusions regarding sensitivities and hedging

	Sensitivities	Speed	Stability	Local accuracy	CVA run-on hedge
	Benchmark bump	**Very slow**	**Very stable**	**Benchmark**	**Good**
Fast bump sensis	Linear bump	Fast	Average	Good	Good*
	Smart bump	Fast	Stable	Good	Good
Optimised sensis	EC sensis	Fast	Average	N/A	Very good
	PLE sensis	Very fast	Stable	N/A	Excellent
	LS w/o Γ	Very fast	Stable	N/A	Excellent

By the local accuracy of a bump sensitivity we mean the accuracy of the approximation it provides for the corresponding partial derivative. The asterisk denotes the result of a test not reported in the tables and figures in this paper. N/A: not applicable

E. Conclusions regarding run-on CVA risk

	δCVA$_{(t)}$ learners	Speed	Stability	Twin accuracy	δCVA$_{(t)}$ VAR and ES
Non-parametric	Nested MC	Very slow	Stable	Very good	Very conservative
	Neural net	Fast	Average	Good	Conservative
Linear (-quadratic) in market bumps	**Benchmark bump**	Very slow	Very stable	Good/average for small/large t	Aggressive
	LS w/ Γ in δZ_t	Very fast	Stable	Good	Conservative

Figure 2 shows the run-on CVA hedging performance of different candidate sensitivities. All the risk compression ratios decrease with the risk horizon t. All sensitivities reduce both unexplained PnL and EC by at least 2.5 times for $t = 0.01$ and 4.5 times for $t = 0.1$ or 1. Since client defaults are skipped in the run-on mode, the efficiency of bump sensitivity hedges is understandable. For each risk horizon and performance metric, the optimised sensitivities always have better results than (benchmark or smart) bump sensitivities. Among those, the PLE and LS sensitivity hedges display the highest risk compression ratios and the lowest HVA trend c. Most sensitivities (except for EC sensitivities when $t = 0.01$ or 0.1) also compress the HVA trend c compared with the unhedged case.

Conclusion

Table D synthesises our findings on CVA sensitivities regarding the quality of their approximation of corresponding partial derivatives (for bump sensitivities) and their hedging abilities (for both bump and optimised sensitivities). The winner that emerges as the best trade-off for each downstream task shown in blue or red in the column headings is identified by the same colour in the list of sensitivities. The 'benchmark bump' plays the role of market standard. Sensitivities that are novelties of this work are emphasised in yellow (smart bump sensitivities essentially mean standard bump sensitivities with fewer paths, but with the important implementation caveat mentioned below (1); PLE sensitivities were already introduced in the different context of Simm computations in Albanese *et al* (2017); EC sensitivities were introduced in Rockafellar & Uryasev (2000)).

Regarding the assessment of CVA risk, in the run-on CVA case (see table E, which uses the same colour code as table D), we found that neural net regression of conditional CVA results in VAR and ES estimates that are likely to be more reliable (judging by the twin scores of the associated CVA learners) and also faster than those obtained from CVA Taylor expansions based on bump sensitivities (such as the ones that inspire certain regulatory CVA capital charge formulas). But an LS proxy, linear-diagonal quadratic in market bumps, provides an even quicker (as it is regressed without training) and almost equally reliable view on CVA risk as the neural net CVA. In the run-off CVA case (not represented in the table), the neural net learner of CVA$_t$ in the risk mode (or equally the nested CVA learner, though taking a much longer time) allows us to get a consistent and dynamic view on CVA risk together with counterparty default risk. ■

Stéphane Crépey is a distinguished professor of applied mathematics at Université Paris Cité/Laboratoire de Probabilités, Statistique et Modélisation (LPSM), Team Mathematical Finance and Numerical Probabilities; Bouazza Saadeddine is a quantitative researcher at Crédit Agricole CIB; Botao Li is a postdoctoral student; and Hoang Nguyen is a PhD student. The research of Botao Li was funded by the chair of Capital Markets Tomorrow: Modeling and Computational Issues under the aegis of the Institut Europlace de Finance, a joint initiative of LPSM/Université Paris Cité and Crédit Agricole CIB, with the support of Labex FCD (ANR-11-LABX-0019-01). The research of Hoang NGuyen is funded by a CIFRE grant from Natixis. The authors are grateful to Moez Mrad, head of XVA, counterparty risk, collateral and credit derivatives quantitative research at Crédit Agricole CIB, and to an anonymous referee for inspiring exchanges.
Email: stephane.crepey@lpsm.paris,
bouazza.saadeddine2@ca-cib.com,
botaoli@lpsm.paris,
hdnguyen@lpsm.paris.

REFERENCES

Abbas-Turki L, S Crépey and B Saadeddine, 2023
Pathwise CVA regressions with oversimulated defaults
Mathematical Finance 33(2), pages 274–307

Albanese C, C Benezet and S Crépey, 2023
Hedging valuation adjustment and model risk
Preprint, arXiv:2205.11834v2

Albanese C, S Caenazzo and M Syrkin, 2017
Optimising VAR and terminating Arnie-VAR
Risk October, www.risk.net/5342056

Albanese C, S Crépey, R Hoskinson and B Saadeddine, 2021
XVA analysis from the balance sheet
Quantitative Finance 21(1), pages 99–123

Antonov A, S Issakov and A McClelland, 2018
Efficient SIMM-MVA calculations for callable exotics
Risk August, www.risk.net/5902496

Capriotti L, Y Jiang and A Macrina, 2017
AAD and least-square Monte Carlo: fast Bermudan-style options and XVA Greeks
Algorithmic Finance 6(1–2), pages 35–49

Rockafellar RT and S Uryasev, 2000
Optimization of conditional value-at-risk
Journal of Risk 2, pages 21–42