Arbitrage-Free Implied Volatility Surface Generation with Variational Autoencoders*

Brian (Xin) Ning†, Sebastian Jaimungal†, Xiaorong Zhang†, and Maxime Bergeron‡

Abstract. We propose a hybrid method for generating arbitrage-free implied volatility (IV) surfaces consistent with historical data by combining model-free Variational Autoencoders (VAEs) with continuous time stochastic differential equation (SDE) driven models. We focus on two classes of SDE models: regime switching models and Lévy additive processes. By projecting historical surfaces onto the space of SDE model parameters, we obtain a distribution on the parameter subspace faithful to the data on which we then train a VAE. Arbitrage-free IV surfaces are then generated by sampling from the posterior distribution on the latent space, decoding to obtain SDE model parameters, and finally mapping those parameters to IV surfaces. We further refine the VAE model by including conditional features and demonstrate its superior generative out-of-sample performance.

1. Introduction.

Modelling implied volatility (IV) surfaces in a manner that reflects historical dynamics while remaining arbitrage-free is a challenging open problem in finance. There are numerous approaches driven by stochastic differential equations (SDEs) that aim to do just so, including local volatility models [13], stochastic volatility models [18, 17], stochastic local volatility models [29], jump-diffusion models [12], and regime switching models [7], among many others. Such approaches make specific assumptions on the dynamics of the underlying asset and a choice of an equivalent martingale measure in order to avoid arbitrage. While these assumptions are not necessarily dynamically consistent with historical data, they do allow, e.g., pricing exotic derivatives via Monte Carlo or PDE methods.

An alternative to the SDE approach is to use non-parametric models to approximate IV surfaces directly without making assumptions on the underlying dynamics. For example, ML models such as support vector machines (SVMs) have been used to model such surfaces [33]. The issue of ensuring arbitrage-free surfaces is often tackled jointly during model fitting [3] either through penalisation of arbitrage constraints [1] or by directly encoding them into the network architecture [34]. These approaches, however, typically do not provide any guarantees and may not be arbitrage-free across the entire surface. A recent intriguing approach [11] is to reduce surfaces to arbitrage-free 'factors' – learned, e.g., through principal component analysis (PCA) – which can then be modeled using neural SDEs [26]. This approach, while very promising, relies on the quality of the 'factors' which are often complicated to compute. Another recent approach is that of [10] where the authors use Gaussian processes under shape constraints to generate surfaces and illustrate good fits to S&P data. Here, however, we are interested in the setting of sparse FX data and in generating the distribution over surfaces in a manner that is consistent with the historical data. The construction of arbitrage-free models based on ML approaches for stochastic interest rates has been tackled in [24]. In contrast, our

*The authors thank Ivan Sergienko for his comments on earlier versions of this work. S.J. acknowledges the support of the Natural Sciences & Engineering Research council of Canada [ALLRP 550308 - 20].

†Department of Statistical Sciences, University of Toronto (brian.ning@mail.utoronto.ca, sebastian.jaimungal@utoronto.ca; http://sebastian.statistics.utoronto.ca, xiaorong.zhang@mail.utoronto.ca)

‡Riskfuel Analytics (mb@riskfuel.com; http://riskfuel.com)

focus is on European options and, more specifically, our application setting is to FX options.

In this paper, we develop a hybrid approach to resolve these issues by using SDE models that are by construction arbitrage-free yet flexible enough to fit arbitrary IV surfaces. One immediate dividend of this approach lies in its ability to produce realistic synthetic training data that can be used to leverage deep learning pricing methods in downstream tasks [15, 20]. The class of SDE models we consider include time-varying regime switching models and Lévy additive processes detailed in Section 3. We avoid overfitting by incorporating a Wasserstein penalty to keep the SDE model's risk-neutral density from deviating too far from the candidate one. The SDE model parameters, once fitted to data, represent a parameter subspace reflecting the features embedded in the data. The distribution on the subspace depends on the characteristics of the underlying asset and can be complex. We "learn" this distribution by using Variational Autoencoders (VAEs) which also allows for disentanglement of the subspace in an interpretable manner. SDE model parameters may be generated from the VAE model and used to create IV surfaces that are both faithful to the historical data but also strictly risk-neutral. This is similar in spirit, but distinct from, the tangent Lévy model approach introduced in [8] where a Lévy density is used to generate arbitrage-free prices while, here, the VAE generates parameters of the SDE model.

The overall approach may be summarised as: (i) fit a rich arbitrage-free SDE model to historical market data to obtain a collection of parameters, (ii) train a generative model, in particular a VAE model, on the collection of SDE model parameters, (iii) sample from the latent space of the generative VAE model, (iv) decode the samples to obtain a collection of SDE model parameters, and (v) use said SDE model and parameters to obtain arbitrage-free surfaces faithful to the historical data. A flow-chart of the process is presented in Figure 1. We further refine the VAE model by including conditioning features into the encoding and decoding architectures. This results in a conditional VAE (CVAE) model, first introduced in [30] in a very different setting, for the arbitrage-free model parameter embeddings. We find that the CVAE model outperforms all others when comparing out-of-sample performance.

The remainder of this article is organised as follows. Section 2 describes a generic method of fitting SDE models to a limited data set. Section 3 defines the financial models we use in calibration. Section 4.1 details the structure of the VAE and its generative process. Section 4.4 extends the VAE framework by conditioning prespecified features. Finally, Section 5 presents the results of our algorithm applied to 1,900 days of foreign exchange (FX) data for three currency pairs[1]: AUD-USD, EUR-USD, and CAD-USD.

2. Model Setup and Estimation Procedure.

We work with a completed filtered probability space $(\Omega, \mathbb{Q}, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0})$ where the filtration is the natural one generated by a stochastic driver $X := (X_t)_{t\geq 0}$. We explore several choices of models for X in Section 3. Here, $\mathbb{Q}$ represents the risk-neutral probability measure and we assume that the market prices options using this measure and model the FX rate process $S = (S_t)_{t\geq 0}$ as follows:

$$S_t = S_0\, e^{\int_0^t (r_s^d - r_s^f)\, ds + \int_0^t l(s) ds + X_t}, \qquad \forall\, t \geq 0, \tag{2.1}$$

[1] AUD = Australian Dollar, USD = US Dollar, and CAD = Canadian Dollar.

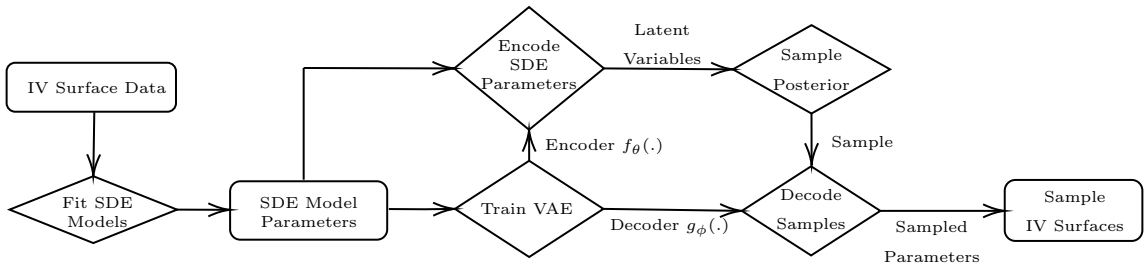


Figure 1: Flow chart of algorithm starting from raw IV surface data to generated surfaces.

where $r^d := (r^d_t)_{t\geq 0}$ and $r^f := (r^f_t)_{t\geq 0}$ are the domestic and foreign short rate processes, and l is a deterministic function of time that ensures $(e^{-\int_0^t (r^d_s - r^f_s)\,ds} S_t)_{t\geq 0}$ is a $\mathbb{Q}$-martingale. We assume interest rates are deterministic since there is no conceptual difficulty in generalising to the stochastic case.

From, e.g., Theorem 3.2 in [25], we may write the undiscounted option price as

$$\mathbb{E}^{\mathbb{Q}}\Big[(S_\tau - K)_+\Big] = \tilde{S}_0 - \frac{\sqrt{K\,\tilde{S}_0}}{\pi}\int_0^\infty \Re\left(e^{i\,z\,\log\frac{\tilde{S}_0}{K}}\,\phi_\theta\left(z - \tfrac{1}{2}i\right)\right)\frac{dz}{z^2 + \frac{1}{4}}, \tag{2.2}$$

where $\tilde{S}_0 := S_0\, e^{\int_0^\tau (r^d_s - r^f_s)\,ds + \int_0^\tau l(s)ds}$, $\phi_\theta(z) := \mathbb{E}^{\mathbb{Q}}[e^{izX_\tau}]$ is the characteristic function of X_τ, θ encodes the parameters of the stochastic driver X, and $\Re(\cdot)$ denotes the real component of its argument. For the class of SDE models considered here, the characteristic function is known in closed form and the above formula allows for efficient calibration to market data.

A naive approach to parameter estimation is to minimise the squared error between the model and data prices. Such parameter estimation is prone to overfitting when data is sparse as is often the case in FX markets. To address this issue, we add a regularisation term to our objective. Specifically, we use the 1-Wasserstein distance between the model price's risk-neutral probability distribution function (pdf), denoted $F^m_\theta(dx)$, and the implied pdf derived from option data, denoted $F^d(dx)$. Wasserstein distances provide a natural metric on the space of probability measures and have seen wide application across many fields [32]. Other choices include divergences, such as the Kullback-Liebler divergence, or metric variations such as Jensen-Shannon divergence. We chose the Wasserstein distance in particular because it is the most ubiquitous and robust.

To this end, the 1-Wasserstein distance between F^d and F^m_θ is given by

$$W_1(F^d, F^m_\theta) = \inf_{\pi\in\Pi(F^d, F^m_\theta)} \int_{\mathbb{R}^2} |x-y|\,d\pi(x,y) = \int_{\mathbb{R}} |F^d(x) - F^m_\theta(x)|\,dx, \tag{2.3}$$

where $\Pi(F^d, F^m_\theta)$ denotes the set of all probability distributions on $\mathbb{R}^2$ with marginals F^d and F^m_θ and the second equality holds in dimension one [31]. As data is observed at discrete strikes, we approximate the candidate density by interpolating IVs at each fixed maturity using B-splines. It is well known [6] that $\partial_{KK}C(T,K)$ corresponds to the risk-neutral density of the underlying asset price evaluated at K. The derivation of the spline implied density for the case of call options can be found in Appendix A. Since we are using a B-spline, the corresponding density is not necessarily risk-neutral. This is not a problem, however, as we

merely use them as a regularisation term and ultimately use risk-neutral models to derive option prices.

We use a combination of the pricing error and the Wasserstein distance (2.3) as the loss function in model estimation. That is we seek to obtain model parameters

$$\theta^* := \underset{\theta}{\operatorname{argmin}} \left(\left\| \boldsymbol{C}^{\boldsymbol{m}}_{\boldsymbol{\theta}} - \boldsymbol{C}^{\boldsymbol{d}} \right\|^2 + \alpha \sum_{n=1}^{N} W_1(F^{d,(n)}, F_\theta^{m,(n)}) \right), \tag{2.4}$$

where $\boldsymbol{C}^{\boldsymbol{d}}$ and $\boldsymbol{C}^{\boldsymbol{m}}_{\boldsymbol{\theta}}$ denotes the flattened vector of data and model prices, respectively, at each strike-maturity pair, and $F^{d,(n)}$ and $F_\theta^{m,(n)}$ represent the model and data implied distribution functions at each maturity, respectively. The regularisation parameter $\alpha \geq 0$ controls the importance placed on being close to the spline implied densities. The effect this hyperparameter has on model accuracy is explored in detail in Appendix B.

3. Class of Stochastic Drivers.

In this section, we describe the class of models over which we perform estimation. Throughout, we denote the sequence of dates on which we have option implied volatility data by $\{\tau_1, \tau_2, \ldots, \tau\}$ and define $\tau_0 = 0$.

3.1. CTMC.

The continuous time Markov-Chain model (CTMC) is a multi-regime model that assumes the underlying asset follows a Geometric Brownian motion (GBM) modulated by a continuous time Markov-Chain representing the current market regime. They were first introduced into financial modelling in [7]. Here, however, we generalise the model to allow for time-varying parameters and use transform methods in [21] to solve for the characteristic function. We use this model as a non-parametric approach to modelling the sequence of risk-neutral densities. The potential of overfitting of such models is mitigated by the Wasserstein distance penalty in (2.3).

Let $(Z_t)_{t\geq 0}$ denote a continuous time Markov chain taking on values in $\mathfrak{K} := \{1 \ldots K\}$. Suppose, moreover, that the generator matrix A driving the CTMC, the regime specific vector of drifts μ, and the regime specific vector of volatilities σ are all constant on the sequence of maturity intervals $[\tau_{i-1}, \tau_i)$, but may vary across maturity periods. We can then consider a driving process X satisfying the SDE

$$dX_t = \left(\mu^{(n)}_{Z_t} - \tfrac{1}{2}(\sigma^{(n)}_{Z_t})^2 \right) dt + \sigma^{(n)}_{Z_t} \, dW_t, \qquad \forall t \in [\tau_{n-1}, \tau), \tag{3.1}$$

where $\mu^{(n)}_k \in \mathbb{R}$ and $\sigma^{(n)}_k \in \mathbb{R}_+$ denote the expected return and instantaneous volatility in period n when $Z_t = k$, $k \in \mathfrak{K}$. Similarly, we let $A^{(n)}$ denote the transition rate matrix for period n, satisfying $A^{(n)}_{ij} \in \mathbb{R}_+$, $i \neq j$, and $\sum_{j=1}^{K} A_{ij} = 0$.

Proposition 3.1. *If X satisfies (3.1), then the characteristic function $\phi_{X_\tau}(\omega) := \mathbb{E}^{\mathbb{Q}}[e^{i\omega X_\tau}]$ is given by $\phi_X(\omega) = \boldsymbol{\pi}^\intercal e^{\tau_1 \Psi^{(1)}(\omega)} e^{(\tau_2 - \tau_1)\Psi^{(2)}(\omega)} \ldots e^{(\tau - \tau_{N-1})\Psi^{(N)}(\omega)} \, \mathbf{1}$, where $\pi_k = \mathbb{Q}(Z_0 = k)$ is the prior probability of the latent state,*

$$[\Psi^{(n)}(\omega)]_{jk} := \left(A^{(n)}_{kk} + i \left(\mu^n_k - \tfrac{1}{2}(\sigma^n_k)^2 \right) \omega - \tfrac{1}{2}(\sigma^n_k)^2 \omega^2 \right) \delta_{jk} + A^{(n)}_{jk} \, (1 - \delta_{jk}), \tag{3.2}$$

and δ_{jk} is the Kroencker delta, which equals 1 if $j = k$ and 0 otherwise.

Proof. See Appendix ∎

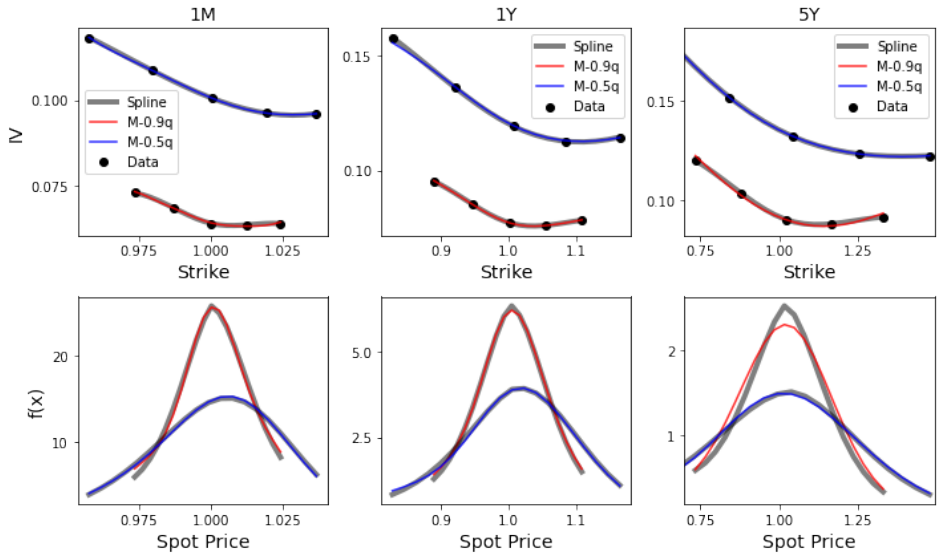


Figure 2: Typical fits of the CTMC model to the IVs (top) and densities (bottom) of AUD-USD data with root mean squared error in the 50th (blue) and 90th (red) quantiles. Strike ranges differ on different days.

To assist with identifiability when estimating the CTMC model parameters, we introduce a cyclic structure on the transition rate matrices. More precisely, we require that $A^{(n)}_{ij} = \frac{1}{2}\lambda^{(n)}_i(\delta_{j=i+1} + \delta_{j=i-1})$, for all $K > i > 1$, $A^{(n)}_{1,2} = A^{(n)}_{1,K} = \frac{1}{2}\lambda^{(n)}_1$, $A^{(n)}_{K,1} = A^{(n)}_{K,K-1} = \frac{1}{2}\lambda^{(n)}_K$, and $\lambda^{(n)}_i > 0$, together with the usual constraint that $\sum_{j=1}^K A_{ij} = 0$, for all $i \in \mathfrak{K}$.

Figure 2 shows the CTMC model fitted to two days of data including the corresponding implied densities. The two specific days are chosen as they correspond to a root mean squared error (rmse) that lie in the 50th and 90th quantiles of rmse across all days.

3.2. Lévy Additive Processes. For comparison, we also study a class of Lévy additive processes to allow jumps in FX rates. To this end, we model X as

$$X_t = \int_0^t \int_{\mathbb{R}} y\, [\mu(dy, ds) - \nu(dy, ds)] + \int_0^t \sigma_s\, dW_s, \tag{3.3}$$

where $(\sigma_t)_{t\geq 0}$ is piecewise deterministic, $\sigma_t := \sigma^{(n)} \mathbb{1}_{t\in[\tau_{n-1},\tau)}$, μ is a Poisson random measure with compensator ν, and where $\nu(dy, ds) = \mathbb{1}_{t\in[\tau_{n-1},\tau)}\, \nu^{(n)}(dy)\, ds$ with $\nu^{(n)}(dy)$ being Lévy measures. This allows the structure of the Lévy measure to differ between maturity periods and allows for both finite and infinite activity processes. For example, we may have $\nu^{(n)}(dy) = \lambda^{(n)}\, F^{(n)}(dy)$, in which case X is an additive compound Poisson process with jump measure $F^{(n)}(dy)$ and intensity $\lambda^{(n)}$, or $\nu^{(n)}(dy) = C\left(\frac{e^{-G|y|}}{|y|^{1+Y}}\mathbb{1}_{x<0} + \frac{e^{-M\,y}}{|y|^{1+Y}}\mathbb{1}_{x>0}\right) dy$ in which case X is additive version of a tempered stable Lévy measure (also known as the KoBol [5, 4] or CGMY [9] models). There are a slew of alternate models as well, however, in the sake of brevity we restrict to these two classes. For the additive compound Poisson process, we include two jump measure types: mixture of normals $F^{(n)}(dy) = \sum_{j=1}^K \tilde{\pi}^{(n)}_k\, \phi\left(\frac{y-\tilde{\mu}^{(n)}_k}{\tilde{\sigma}^{(n)}_k}\right) dy$,

Model	Characteristic Function
CTMC	$\boldsymbol{\pi}^\intercal e^{\tau_1 \Psi^{(1)}(\omega)} e^{(\tau_2-\tau_1)\Psi^{(2)}(\omega)} \cdots e^{(\tau-\tau_{N-1})\Psi^{(N)}(\omega)} \mathbf{1}$
Double Exponential JD	$-\frac{\sigma^2\omega^2}{2} + \lambda\left(p\frac{a_+}{a_+-i\omega} + (1-p)\frac{a_-}{a_-+i\omega} - 1\right)$
Gaussian Mixture JD	$-\frac{\sigma^2\omega^2}{2} + \lambda\left(\sum_{i=1}^K \tilde{\pi}_i (e^{i\bar{\mu}_i\omega - \tilde{\sigma}_i{}^2\omega^2/2}) - 1\right)$
CGMY/KoBoL	$C\Gamma(-Y)[(M-i\omega)^Y - M^Y + (G+i\omega)^Y - G^Y]$

Table 1: Summary of characteristic functions within a given maturity period.

where ϕ is the standard normal pdf – to mimick a non-parametric estimation of the jump distribution implied by the data, and a double exponential model (see [23]), in which case $F^{(n)}(dy) = \left((1-p^{(n)})\, e^{-a_-^{(n)}|x|}\mathbb{1}_{x<0} + p^{(n)}\, e^{-a_+^{(n)}x}\mathbb{1}_{x>0}\right) dy$.

Proposition 3.2. *If X satisfies (3.3), then the characteristic function $\phi_X(\omega) := \mathbb{E}^{\mathbb{Q}}[e^{i\omega X_\tau}]$ is given by $\phi_X(\omega) = e^{\sum_{n=1}^N \Psi^{(n)}(\omega)(\tau-\tau_{n-1})}$, where*

$$\Psi^{(n)}(\omega) = -\tfrac{1}{2}(\sigma^{(n)})^2\omega^2 + \int_{\mathbb{R}} \left(e^{i\omega y} - 1 - i\omega y \mathbb{1}_{|y|\le 1}\right) \nu^{(n)}(dy)\,. \tag{3.4}$$

Proof. Apply the Lévy-Khintchine formula [2, Chap 1.2.4] within each period. ∎

The specific form of the Lévy characteristic function appearing in (3.4) for the models we employ in the numerical analysis appear in Table 1. The parameters for the appropriate period should be inserted into these expression when computing the full characteristic function. We also record the characteristic function for the CTMC model in the same table.

4. Model Parameter Generation.

In the previous section, we described two classes of stochastic models that can be calibrated to option prices. The SDE model parameters $\boldsymbol{\theta}$ are estimated by minimising a weighted average of the mean-squared error in option prices and the Wasserstein distance between the model implied risk-neutral densities and the densities implied by a B-spline fit of the IV smiles. Once the SDE model parameters are estimated on market data, our goal is to generate new synthetic parameters $\boldsymbol{\theta}$ that are consistent with the historical data. This allows us to produce synthetic IV surfaces that are guaranteed to be both arbitrage-free and representative of real surfaces.

4.1. Variational Autoencoder (VAE).

Variational Autoencoders [22] are generative models that aim to train a multivariate latent representation, known as an encoding, from a collection of data. A key advantage of VAEs over conventional dimensionality reduction techniques such as vanilla autoencoders (AEs), PCA, or KPCA is their ability to generalize the latent feature space. A well-known issue of AEs is that the latent space may not be continuous and may not exhibit any well defined structure. This makes it difficult to interpolate between training data points and poses difficulties in generating new data, as demonstrated by [27]. In [27], attempts to rectify such issues are made by regularizing the training procedure to ensure the latent manifold is smooth and locally convex. VAEs, however, place a prior on the latent feature and are thus able to easily generate out-of-sample data by sampling from either the prior or the posterior, depending on the specific application.

Given a set of samples $\{\mathbf{x_i} \,|\, \mathbf{x_i} \in \mathbb{R}^D\}_{i\geq 1}$ from a distribution parameterized by ground truth latent factors $\{\mathbf{z_i} \,|\, \mathbf{z_i} \in \mathbb{R}^{D'}\}_{i\geq 1}$ where $D \gg D'$ and a generative model $p_\psi(\mathbf{x})$, we seek to maximise the log-likelihood $\log p_\psi(\mathbf{x})$. The log-likelihood is, however, intractable as it involves integration over the posterior $p_\psi(\mathbf{z}|\mathbf{x})$, which, even in setups where $p(\mathbf{x}|\mathbf{z})$ is specified, is itself intractable. The VAE circumvents this issue by introducing an approximation of the true posterior with a neural network $q_\phi(\mathbf{z}|\mathbf{x})$ parameterized by ϕ. Indeed, for any distribution $q_\phi(\mathbf{z}|\mathbf{x})$, we have that the log-likelihood satisfies the inequality

$$\log p_\psi(\mathbf{x}) = \log \int p_\psi(\mathbf{x}|\mathbf{z})\, p_\psi(\mathbf{z})\, d\mathbf{z} = \log \int \frac{p_\psi(\mathbf{x}|\mathbf{z})\, p_\psi(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\, q_\phi(\mathbf{z}|\mathbf{x})\, d\mathbf{z}$$
$$\geq \int \log\left(\frac{p_\psi(\mathbf{x}|\mathbf{z})\, p_\psi(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right) q_\phi(\mathbf{z}|\mathbf{x})\, d\mathbf{z} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\psi(\mathbf{x}|\mathbf{z})\right] - KL\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p_\psi(\mathbf{z})\right].$$

The right most expression of this inequality is known as the evidence lower bound (ELBO) of the log-likelihood. It can be shown that the precise gap between the left and right hand sides of this inequality equals the KLD (KL-Divergence) between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$, i.e.,

$$\log\, p_\psi(\mathbf{x}) = ELBO + KL[\, q_\phi(\mathbf{z}|\mathbf{x}) \| p_\phi(\mathbf{z}|\mathbf{x})\,]. \tag{4.2}$$

VAEs then view the negative ELBO as a loss and, rather than maximising the intractable log-likelihood, aim to minimize

$$-\, ELBO(\phi, \psi) = KL[\, q_\phi(\mathbf{z}|\mathbf{x}) \| p_\psi(\mathbf{z})\,] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\,\log(p_\psi(\mathbf{x}|\mathbf{z}))\,], \tag{4.3}$$

where the first term is known as the KLD loss and the second term the reconstruction loss. In principle, the prior, posterior, and generator can be arbitrary distributions. In practice, however, this typically leads to an intractable ELBO. Thus, we assume they are from the family of Gaussian distributions with diagonal covariance matrices, and that the prior is the isotropic unit Gaussian. There is no real loss of generality in light of the universal approximation theorem. Specifically, we assume $q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu_\phi}(\mathbf{x}), \boldsymbol{\sigma_\phi}(\mathbf{x}))$, $p_\psi(\mathbf{x}|\mathbf{z}) \sim \mathcal{N}(\boldsymbol{\mu_\psi}(\mathbf{x}), \boldsymbol{\sigma_\psi}(\mathbf{x}))$, and $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, where $\boldsymbol{\sigma_\phi}(\mathbf{x})$ and $\boldsymbol{\sigma_\psi}(\mathbf{x})$ are diagonal matrices.

The neural network that parametrises $q_\phi(\mathbf{z}|\mathbf{x})$ is called the encoder as it "encodes" the data into its latent representation, while the neural net that parameterises $p_\psi(\mathbf{x}|\mathbf{z})$ is called the decoder as it "decodes" latent representation to recover the original data. Figure 3 shows the typical structure of a VAE. We train the encoding ϕ and decoding ψ networks simultaneously but only use the decoder at inference time once a sample of latent features have been chosen. As detailed in Section 4.3, however, the latent sampling procedure makes use of the encoder.

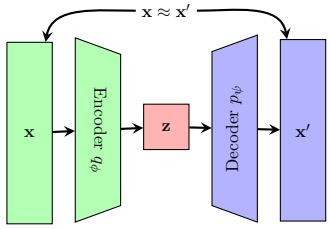


Figure 3: VAE architecture.

4.2. β-VAE. The β-VAE [19] is a modification of the traditional VAE objective that introduces an adjustable hyperparameter $\beta > 0$, precisely the ELBO is modified to

$$\underset{\phi,\psi}{\text{argmax}}\ \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log(p(\mathbf{x}|\mathbf{z}))] - \beta\, KL(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})). \tag{4.4}$$

Larger values of β result in more disentangled latent representations $\mathbf{z}$, while smaller values result in more faithful reconstructions. β is often chosen to be greater than one to encourage disentanglement; however, this constrains latent information $\mathbf{z}$ and can lead to poorer reconstructions [19]. Instead, we may choose smaller values of β to improve reconstructions and rely on directly sampling from the posterior to accurately sample from the less structured latent space.

4.3. Latent Sampling. Typically, samples from a VAE are generated by sampling from the latent prior $p(\mathbf{z})$ and decoding the result. From a Bayesian perspective, however, conditional on a set of observed data $\boldsymbol{X}$, it is more appropriate to sample from the posterior $p_\psi(\mathbf{z}|\boldsymbol{X}) \approx q_\phi(\mathbf{z}|\boldsymbol{X}) = \int q_\phi(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$. While this integral is typically intractable, it is possible to sample from. This can be done by first uniformly sampling from the data $\mathbf{x}_0 \sim \boldsymbol{X}$, encoding using the approximate posterior $q_\phi(\mathbf{x}_0)$ to obtain the Gaussian parameters $\mu_\phi(\mathbf{x}_0)$ and $\sigma_\phi(\mathbf{x}_0)$, and finally sampling latent states from said Gaussian $\mathbf{z}_0 \sim \mathcal{N}(\mu_\phi(\mathbf{x}_0), \sigma_\phi(\mathbf{x}_0))$. This latent sample $\mathbf{z}_0$ can then be decoded to obtain model parameters which may then be used to construct the IV surface.

4.4. Conditional Variational Autoencoder (CVAE). A natural extension of the variational modeling framework is the inclusion of observable market features, such as indices or spot rates. We may then generate surfaces conditional on the state of these features through Conditional Variational Autoencoders (CVAEs) [30]. Such an approach may be used in risk calculations in which scenarios for the conditional features are generated through other means, and our approach used to generate IV surfaces conditioned on those simulated features.

In brief, a CVAE is constructed as follows. Given ground truth latent factors $\mathbf{z}$, observations $\mathbf{x}$, and conditional features $\mathbf{y}$, we define the generative model $p_\psi(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ where $p(\mathbf{y})$ is the prior on the conditional features. Following the same argument as in section 4.1 we approximate the posterior $p_\psi(\mathbf{z}|\mathbf{x},\mathbf{y})$ by a neural network $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ and minimize the conditional negative ELBO:

$$-ELBO(\phi,\psi) = KL[\, q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \| p_\psi(\mathbf{z}|\mathbf{y})\,] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log(p_\psi(\mathbf{x}|\mathbf{z},\mathbf{y}))\,]. \tag{4.5}$$

This allows the conditioning feature $\mathbf{y}$ to modulate how data $\mathbf{x}$ gets encoded and how a latent factor $\mathbf{z}$ gets decoded. In the results section, we include the VIX as a conditioning feature and find that it improves the out of sample performance of the VAE model. In implementation, the CVAE model is identical to the VAE model, except that the conditional features are added into the encoding and decoding networks.

5. Results.

5.1. Market Data. We apply our hybrid method to IV data for three currency pairs provided by Exchange Data International: AUD-USD, EUR-USD, and CAD-USD for the 1,900 days between September 18th, 2012 to December 30, 2019. The data is divided equally into the training set (September 18th, 2012 to May 09, 2016) and the testing set (May 10, 2016 to December 30, 2019.) The data includes option prices with five strikes at each of eight different maturities (1M, 2M, 3M, 6M, 9M, 1Y, 3Y, 5Y). Foreign exchange option prices are quoted in terms of deltas rather than strikes and in terms of at-the-money call, risk reversal,

	AUD-USD	EUR-USD	CAD-USD
CTMC	8.1	5.0	6.0
DE JD	62.0	42.9	64.2
GM JD	10.2	17.0	24.3
CGMY/KoBoL	140.2	151.3	136.2

Table 2: Median rmse ($\times 10^{-5}$) for the collection of models and FX pairs.

and butterfly spread options, rather than simple calls. We use standard formulas [28] to convert raw quotes into IVs at deltas of 0.1, 0.25, 0.5, 0.75, and 0.9 for each maturity.

5.2. SDE Model Specifics. The CTMC model assumes three regimes with a Wasserstein's distance penalty[2] of 0.3. To reduce the number of parameters to fit, initial regime probabilities $\boldsymbol{\pi}$ are set to $\frac{1}{3}$. To take advantage of the label invariance of the transition matrix, the mean μ of each regime is assumed to be in ascending order, i.e., $\mu_1^n \leq \mu_2^n \leq \mu_3^n$. The structure of the transition matrix is detailed in Appendix D. We fit the CTMC model iteratively by first optimizing for the parameters of the first maturity, then iteratively optimizing the parameters of the i'th maturity by holding the parameters of the first $(i-1)$ maturities fixed. For the Lévy additive processes, we fit the parameters subject to a Wasserstein's distance penalty of 0.1 for time to maturity (TTM) less than 1 year and 0.3 otherwise to increase the regularising power at larger TTM. Moreover, we apply a penalty of 10^{-8} on day-to-day parameter percentage changes to stabilise model parameters without sacrificing fit quality. For the Gaussian Mixture JD model, we assume a mixture of two Gaussians, as adding more factors did not increase the fit quality. The penalty varies with maturity as long maturities tend to have stable pdfs but less stable IVs, while shorter maturities tend to have less stable pdfs. Table 2 show the median rmse across days, where the rmse on a given day is computed across all Delta/maturity pairs, for the collection of models and FX pairs we study.

5.3. VAE Model Specifics. The encoder and decoder of the VAE have four fully connected hidden layers, with 64, 128, 256, and 512 nodes each, and a single output layer mapping to the appropriate dimensions. The network structure was selected using a validation set, however, we found that any network exceeding four layers with a minimum of 64 node in each layer is sufficient to produce satisfactory results. We used ADAM with weight decay (AdamW) with a fixed learning rate of 0.001. Appropriate transformations (normalizations, log-transforms) are performed to ensure standardized inputs. Details of the transformations used can be found in Table 6 in Appendix E. We perform a grid search over β values and number of latent dimensions summarized in Table 3. The table reports an evaluation metric described in the next subsection. Training is carried out with batches of 200 randomly sampled days from the training set. We set a fixed training duration of $2,000$ epochs as we find that is usually sufficient to train the β-VAE.

5.4. Benchmarks. We introduce three benchmarks to assess the performance of our approach. These benchmark models all generate distributions of IV surfaces (using only the

[2]The Wasserstein penalty for the various models are chosen from a grid search and balancing goodness of fit to IV smiles with goodness of fit to the implied risk-neutral densities.

training data) that we use to assess how close they are to the testing data. Details on the benchmarks themselves will be given below while the metric we use is described in the next subsection.

The first is a β-VAE that is fit directly to the set of IVs on the fixed grid of delta and time to maturity (as defined in Section 5.5) without the addition of any arbitrage constraints. This technique is inspired by [3] (see also [34]), although they favor a flexible point-based method where the inputs to the VAE are arbitrary strikes and times to maturity. Typically, these point-based approaches are complemented by either penalizing deviations from (static) arbitrage constraints during training or smoothing the resulting surfaces after generation. However, [3] shows that arbitrage constraints do not enhance the fit and excluding them introduces only negligible amounts of static arbitrage. In our context, the grid-based method is more natural as our data is already structured in this fashion. For comparison purposes, we present the result from a grid of latent dimensions and β's consistent with those used for our other approaches. We label this approach as VAE-IV.

As a second benchmark model, we perform a PCA on the trained CTMC model parameters and sample from the dimensionally reduced latent submanifold using a kernel density estimator (KDE). Several choices for the number of latent dimensions are explored in Table 3. We choose a Gaussian kernel with a bandwidth selected through 20-fold cross-validation. Gaussian kernels are generally quite flexible, but other choices (such as Epanechnikov, Triweight, and Triangular) are possible. For discussions on kernel and bandwidth selection more generally see, e.g., [16]. This PCA approach serves as a simplification of our CTMC-VAE model where the VAE sampler is replaced with a dimensionality reduction technique combined with a KDE sampler.

As a final benchmark, we use the empirical distribution of the training data.

5.5. Evaluation Metric. Our goal is to generate arbitrage-free IV surfaces that are faithful to the historical dataset. Here, we describe a natural metric that allows us to assess how well we meet this goal. Let G and F denote the probability distribution over IV surfaces for the trained model using the our algorithm and the true distribution, respectively. As Wasserstein distances provide a natural metric on the space of probability measures, we use the 1-Wasserstein distance between the trained and true distribution as our performance metric. While the true distribution F is unknown, the data provides a finite sample from it at a set of discrete 2-dimensional grid points $\mathfrak{Z} := \{z_i\}_{i\in\mathfrak{G}}$ (the collection of Delta/TTM pairs which are observed). Specifically, we look at the collection: $\mathfrak{Z} = \mathfrak{D} \times \mathfrak{M}$ where $\mathfrak{D} := \{0.1, 0.25, 0.50.75, 0.9\}$ and $\mathfrak{M} := \{1M, 2M, 3M, 6M, 9M, 1Y, 3Y, 5Y\}$. Further, the trained model's distribution may be estimated by sampling from the posterior distribution in latent space (using the method described in Section 4.3) and decoding to produce SDE model parameters, which can be mapped to a sample of IVs at the set of grid points $\mathfrak{Z}$. The 1-Wasserstein distance between the true and the model's distribution may be estimated by the 1-Wasserstein distance between the multi-variate distribution of IVs at grid points $\mathfrak{Z}$ for the test data and the model generated ones. We refer to this quantity as the Wasserstein metric.

5.6. Results Summary. Table 3 shows a complete summary of the Wasserstein metric computed for each currency pair on a range of β values and latent dimensions. Increasing the number of latent dimensions does not generally increase performance. This suggests that

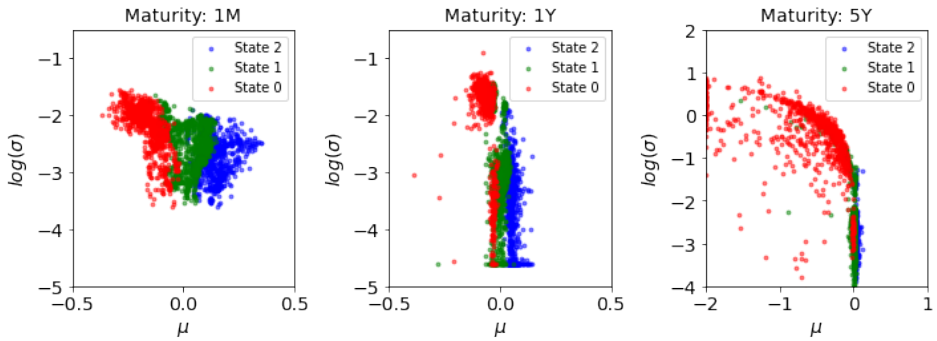
Model			Latent Dimension											
			AUD-USD				EUR-USD				CAD-USD			
		β	3	5	10	15	3	5	10	15	3	5	10	15
VAE	CTMC	0.01	4.56	5.35	4.82	4.40	3.88	3.63	3.61	3.97	1.54	2.12	1.78	1.63
		0.1	4.32	5.83	4.54	4.62	3.18	3.29	3.62	**2.77**	1.81	1.53	1.40	**1.34**
		1	4.67	4.43	4.13	**3.69**	3.64	3.94	3.12	3.62	1.72	1.55	1.48	1.70
		10	6.10	5.92	6.40	5.63	4.00	3.59	3.86	3.90	2.98	2.96	3.17	3.06
	DE	0.01	5.13	5.37	5.30	4.66	3.52	3.79	3.88	3.69	1.61	1.76	1.96	2.09
		0.1	4.61	5.29	5.04	**4.41**	4.58	3.96	3.69	3.87	1.51	1.95	1.50	**1.47**
		1	5.84	4.92	4.61	4.81	3.70	3.46	4.12	4.01	1.65	1.76	1.94	1.51
		10	6.01	5.25	5.40	5.08	**3.38**	3.73	3.89	3.69	1.84	2.28	1.82	2.20
	GM	0.01	5.49	5.14	5.61	5.48	3.65	3.67	3.56	4.55	1.86	1.73	1.88	1.63
		0.1	5.94	5.04	5.02	5.13	3.49	3.36	3.59	3.54	**1.54**	1.70	**1.54**	1.91
		1	5.15	5.27	5.00	5.83	4.01	3.49	3.96	3.30	2.11	2.28	1.93	1.62
		10	5.72	5.56	**4.83**	5.26	3.22	**3.15**	3.73	3.27	1.81	1.70	2.03	2.11
	CGMY/ KoBoL	0.01	5.86	6.11	5.80	**5.53**	3.58	**3.36**	3.94	4.02	2.33	2.08	**1.85**	1.94
		0.1	6.12	6.37	5.99	5.65	3.68	4.17	3.92	3.56	2.19	**1.85**	2.06	2.08
		1	6.54	6.43	6.17	6.22	3.58	3.79	3.94	3.88	2.09	2.34	1.97	2.28
		10	5.96	5.97	5.90	6.20	4.01	3.84	4.31	4.21	2.16	2.24	2.21	2.61
	IV	0.01	6.07	6.16	6.06	6.26	3.55	4.17	4.05	3.65	1.65	1.61	1.62	1.64
		0.1	6.22	6.34	5.87	5.86	3.94	3.61	**3.44**	4.04	1.68	**1.44**	1.59	1.86
		1	6.43	6.55	6.05	**5.60**	3.88	4.14	4.11	4.08	2.05	1.97	1.83	1.79
		10	6.23	6.21	6.16	6.11	4.37	4.14	3.88	4.28	1.82	1.76	2.32	1.75
PCA			**5.25**	5.34	7.15	8.38	**2.51**	3.00	3.5	6.15	2.22	**1.73**	2.40	2.76
Empirical			5.82	5.82	5.82	5.82	3.83	3.83	3.83	3.83	1.67	1.67	1.67	1.67

Table 3: Wasserstein metrics ($\times 10^{-2}$) for varying levels of β, latent dimensionality, and currency pair. Three benchmarks are shown here for reference: the Wasserstein metric ($\times 10^{-2}$) between the test set and the VAE-IV, PCA, and training empirical models as described in Section 5.4. Bold numbers are the smallest metric within each subgrid.

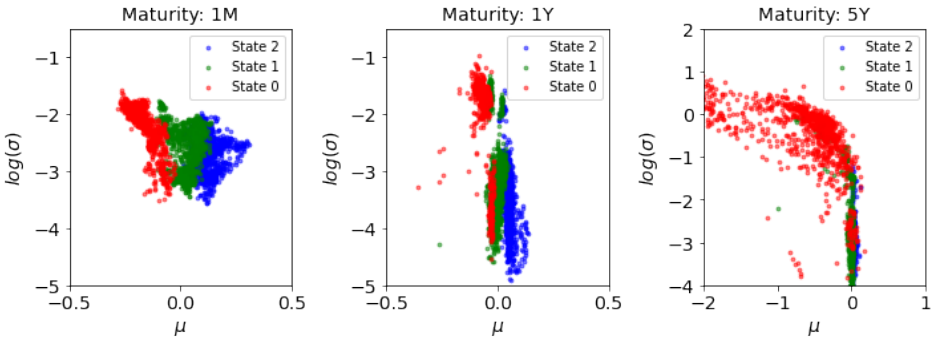
for these currency pairs, most surfaces can be captured with as few as three factors when viewed holistically. However, it does appear that the optimal hyperparameter pair are often at somewhat higher latent dimensions ($\geq$10) which is natural when there are no penalizations on dimensionality. Interestingly, for both classes of SDE models, decreasing β does not lead to significantly worse performance. This is an indication that the posterior sampling from Section 4.3 performed admirably in highly unstructured latent spaces, which is a by-product of low β values. For the three Lévy additive processes explored here, the double exponential model performs the best but it is generally worse performing than the CTMC model. Moreover, the CTMC is able to significantly outperform most of the benchmark methods.

Overall, the results show that our generated surfaces are as close to the testing data's distribution as the training set itself! This suggests that to improve our model's performance we need to include additional explanatory features (Section 5.7) or perhaps a temporal structure.

Figure 4 shows a comparison between (a) the CTMC parameters obtained by fitting to test data, and (b) random samples from the corresponding VAE model. We focus on the two most important parameters μ and σ (which are state and maturity specific) of the CTMC model. We select three maturities 1M, 1Y, and 5Y to succinctly illustrate the results. Ideally, this comparison would be made using the IV surfaces directly implied by the the parameters; however, no good visualization is available for such comparisons. Instead, we illustrate the similarity of the generated and test data distributions using the model parameters. The figure showcases the VAE's ability to capture the complex structures of the CTMC model parameters



(a) Testing Parameters



(b) Generated Parameters

Figure 4: Scatter plots of the three regime specific μ and σ from (a) fitting to test data, and (b) random sample from the generative model, using the CTMC model on AUD-USD data. Colours represent three different regimes.

which in turn is used to generate the final implied volatility surfaces. As the figure shows, the VAE successfully captures the various complex structures that are inherent in the test data across all maturities and states.

Next, we investigate the impact the training window has on our results. To this end, Figure 5 summarizes how the length of the training set affects the Wasserstein metric between the testing data and the CTMC-VAE model. The score is computed as the average Wasserstein metric, using a randomly generated sample of 500 surfaces and a predefined testing set, across sixteen different sets of hyperparameters as described in Table 3. The testing data is fixed to be the period from January 15th, 2019 to December 30th, 2019. Here, we reduced the size of the test set in order to illustrate how the size of the training set may both improve and worsen the model's performance. The ending date of the training data is fixed to be January 14th, 2019, while the starting date ranges from September 18th, 2012 to December 5th, 2018. The figure suggests that approximately 350 days of data is sufficient to fully train the network. The average score increases as the training set size is reduced beyond this point. In contrast, larger time horizon training sets typically produce worse scores as the training set becomes less representative of the current state of the market. This leads to a generative model that

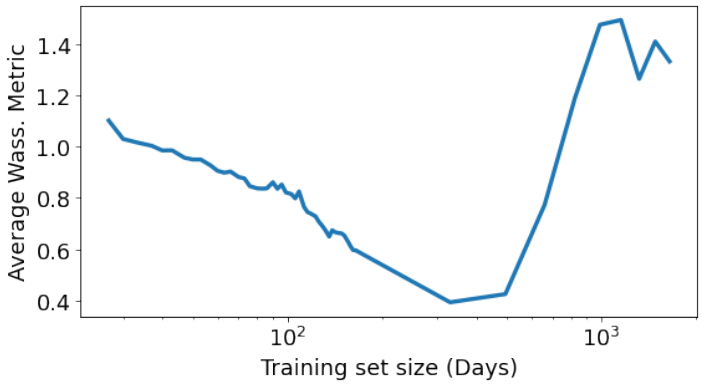


Figure 5: Average scores of the CTMC model when trained on a range of different sized training sets and tested on the period from January 15th, 2019 to December 30, 2019.

may be historically accurate but does not reflect the data in the near future.

While we show the sampling of model parameters in Figure 4, it is informative to generate surfaces themselves. For this purpose, we show three randomly generated surfaces from each of the three currency pairs in Figure 6. There are clear differences between the surfaces for a specific currency pair, however, the general characteristics (skew, level and smile) are similar for a fixed pair. This demonstrates the model's ability to capture the innate characteristics of each currency pair but still faithfully respect the observed variation between random samples.

5.7. CVAE Results. To test the efficacy of the CVAE approach, we focus on using the daily closing CBOE Volatility Index (VIX) as a predictor for generating IV surfaces in the testing set, courtesy of Wharton Data Services [14]. Specifically, we train the CVAE on several currency pairs conditional on the end-of-day VIX index value. We then condition on the value of the VIX index for each day in the testing set to generate surfaces by randomly sampling from the latent space as specified in section 4.3. Table 5 in the Appendix details the results while Table 4 summarizes the comparison between the CVAE, CTMC-VAE, and the benchmarks described in Section 5.4. We train our model using the same training set as in Section 5. The results in the table show that the VIX index has significant predictive power on IV surfaces, as the generated surfaces conditional on the VIX index produces significantly smaller metric compared to both the unconditional CTMC-VAE and all benchmark approaches.

6. Conclusions. Overall, the results show that our generated surfaces are as close to the testing data's distribution as the training set itself! This suggests that to improve our model's performance we need to include additional explanatory features (Section 5.7) or perhaps a temporal structure.

To summarise, we propose a hybrid approach for generating synthetic IV surfaces by first calibrating SDE model parameters to historical data – using a Wasserstein penalty between the implied model risk-neutral distribution and that induced by option data as a regularisation term – and then training a rich VAE model to learn the distribution on the space of SDE model parameters. We show that the distribution of IV surfaces from the VAE model is capable of generating surfaces as close to the testing data's distribution as the training set itself, and performs well in comparison with several benchmarks, while ensuring the generated

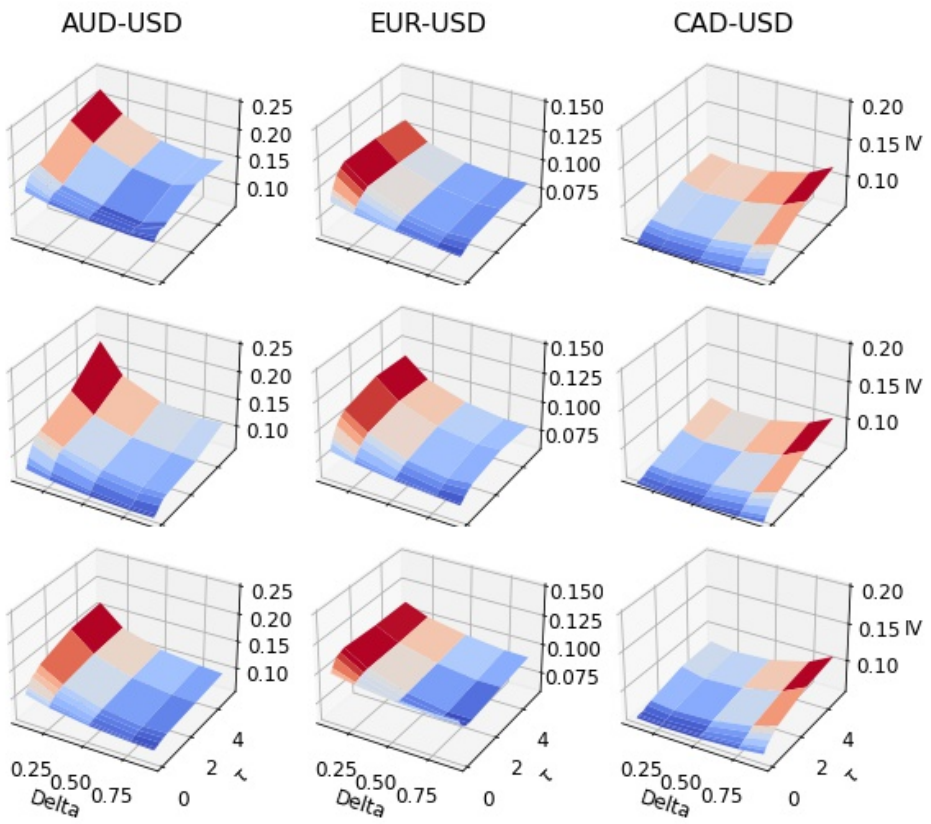


Figure 6: Sample of three randomly generated surfaces using the CTMC-VAE for each of the three currency pairs.

	Latent Dimension											
	AUD-USD				EUR-USD				CAD-USD			
Model	3	5	10	15	3	5	10	15	3	5	10	15
CTMC-CVAE	3.24	3.65	3.35	3.70	2.64	2.72	2.75	2.76	1.47	1.38	1.26	1.37
CTMC-VAE	4.91	5.38	4.97	4.59	3.68	3.61	3.55	3.57	2.01	2.04	1.96	1.93
IV-VAE	6.24	6.32	6.04	5.96	3.94	4.01	3.87	4.02	1.80	1.69	1.84	1.76
PCA	5.25	5.34	7.15	8.38	2.51	3.00	3.50	6.15	2.22	1.73	2.40	2.76
Empirical	5.82	5.82	5.82	5.82	3.83	3.83	3.83	3.83	1.67	1.67	1.67	1.67

Table 4: Average Wasserstein's metric ($\times 10^{-2}$) across all β values for different number of latent dimensions used for the CTMC-CVAE, CTMC-VAE, and benchmarks models IV-VAE, PCA, and Empirical for three currency pairs.

surfaces are arbitrage-free.

A short demo of the CTMC model and VAE model fitting procedure are available[3] at: https://github.com/BrianNingUT/ArbFreeIV-VAE.

[3]Please note that some notebooks require significant run time due to the adaption of C++ to Python.

REFERENCES

[1] D. Ackerer, N. Tagasovska, and T. Vatter, *Deep smoothing of the implied volatility surface*, in Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), 2020.

[2] D. Applebaum, *Lévy processes and stochastic calculus*, Cambridge university press, 2009.

[3] M. Bergeron, N. Fung, Z. Poulos, J. C. Hull, and A. Veneris, *Variational autoencoders: A hands-off approach to volatility*, Available at SSRN 3827447, (2021).

[4] S. Boyarchenko and S. Z. Levendorskii, *Non-Gaussian Merton-Black-Scholes Theory*, vol. 9, World Scientific, 2002.

[5] S. I. Boyarchenko and S. Z. Levendorskiĭ, *Option pricing for truncated lévy processes*, International journal of theoretical and applied finance, 3 (2000), pp. 549–552.

[6] D. T. Breeden and R. H. Litzenberger, *Prices of state-contingent claims implicit in option prices*, Journal of business, (1978), pp. 621–651.

[7] J. Buffington and R. J. Elliott, *Regime switching and european options*, in Stochastic Theory and Control, Springer, 2002, pp. 73–82.

[8] R. Carmona and S. Nadtochiy, *Tangent lévy market models*, Finance and Stochastics, 16 (2012), pp. 63–104.

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

[10] M. Chataigner, A. Cousin, S. Crépey, M. Dixon, and D. Gueye, *Short communication: Beyond surrogate modeling: Learning the local volatility via shape constraints*, SIAM Journal on Financial Mathematics, 12 (2021), pp. SC58–SC69, https://doi.org/10.1137/20M1381538, https://doi.org/10.1137/20M1381538, https://arxiv.org/abs/https://doi.org/10.1137/20M1381538.

[11] S. N. Cohen, C. Reisinger, and S. Wang, *Arbitrage-free neural-sde market models*, arXiv e-prints, (2021), pp. arXiv–2105.

[12] R. Cont and P. Tankov, *Calibration of jump-diffusion option pricing models: a robust non-parametric approach*, https://ssrn.com/abstract=332400, (2002).

[13] B. Dupire et al., *Pricing with a smile*, Risk, 7 (1994), pp. 18–20.

[14] C. B. O. Exchange, *Measure market expectations of near-term volatility conveyed by s&p 500 stock index option prices.*, tech. report, Wharton Research Data Services, 2020.

[15] R. Ferguson and A. Green, *Deeply learning derivatives*, arXiv preprint arXiv:1809.02233, (2018).

[16] A. Gramacki, *Nonparametric kernel density estimation and its computational aspects*, Springer, 2018.

[17] P. S. Hagan, D. Kumar, A. S. Lesniewski, and D. E. Woodward, *Managing smile risk*, Wilmott Magazine, 1 (2002), pp. 249–296.

[18] S. L. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, The review of financial studies, 6 (1993), pp. 327–343.

[19] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, *Beta-VAE: Learning basic visual concepts with a constrained variational framework*, 5th International Conference on Learning Representations, ICLR, (2016).

[20] B. Horvath, A. Muguruza, and M. Tomas, *Deep learning volatility*, arXiv preprint arXiv:1901.09647, (2019).

[21] K. R. Jackson, S. Jaimungal, and V. Surkov, *Fourier space time-stepping for option pricing with lévy models*, Journal of Computational Finance, 12 (2008), p. 1.

[22] D. P. Kingma and M. Welling, *Auto-encoding variational Bayes*, arXiv preprint arXiv:1312.6114, (2013).

[23] S. G. Kou and H. Wang, *Option pricing under a double exponential jump diffusion model*, Management science, 50 (2004), pp. 1178–1192.

[24] A. Kratsios and C. Hyndman, *Deep arbitrage-free learning in a generalized hjm framework via arbitrage-regularization*, Risks, 8 (2020), p. 40.

[25] A. L. Lewis, *A simple option formula for general jump-diffusion and other exponential lévy processes*, Available at SSRN 282110, (2001).

[26] X. Li, T.-K. L. Wong, R. T. Chen, and D. Duvenaud, *Scalable gradients for stochastic differential equations*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 3870–3882.

[27] A. Oring, Z. Yakhini, and Y. Hel-Or, *Autoencoder image interpolation by shaping the latent space*, arXiv preprint arXiv:2008.01487, (2020).
[28] D. Reiswich and W. Uwe, *Fx volatility smile construction*, Wilmott, 2012 (2012), pp. 58–69.
[29] K. Said, *Pricing exotics under the smile*, RISK, 12 (1999), pp. 72–75.
[30] K. Sohn, H. Lee, and X. Yan, *Learning structured output representation using deep conditional generative models*, Advances in neural information processing systems, 28 (2015), pp. 3483–3491.
[31] S. Vallender, *Calculation of the wasserstein distance between probability distributions on the line*, Theory of Probability & Its Applications, 18 (1974), pp. 784–786.
[32] C. Villani, *Optimal transport: old and new*, vol. 338, Springer, 2009.
[33] Y. Zeng and D. Klabjan, *Online adaptive machine learning based algorithm for implied volatility surface modeling*, Knowledge-Based Systems, 163 (2019), pp. 376–391.
[34] Y. Zheng, Y. Yang, and B. Chen, *Gated deep neural networks for implied volatility surfaces*, arXiv preprint arXiv:1904.12834, (2019).

Appendix A. Candidate Risk Neutral Density.

Foreign exchange data is often quoted only at specific strikes/deltas. In order to reduce the possibility of overfitting, we introduce a candidate risk neutral density which we attempt to minimize the 1-Wasserstein distance to. We can approximate this candidate density by interpolating implied volatilities at each fixed maturity using B-splines. It is important to note that as this is simply a candidate density derived from a spline interpolation of the IV surface, it provides no guarantee on risk-neutrality.

Let us define such an interpolated surface by $\sigma(K,\tau)$. The price using this implied volatility may be written in terms of the Black-Scholes price of a call option with spot price S_0, strike K, maturity τ, and risk-neutral interest rate r as

$$C(S_0,K,\tau,r) = S_0\,\Phi(d^+(K,\tau)) - e^{-r\tau}\,K\,\Phi(d^-(K,\tau)), \tag{A.1}$$

$$d^\pm(K,\tau) = \frac{1}{\sigma(K,\tau)\sqrt{\tau}}\left(\log\left(\tfrac{S_0}{K}\right) + (r \pm \tfrac{1}{2}\sigma^2(K,\tau))\tau\right). \tag{A.2}$$

It is well known [6] that, in an arbitrage-free model, $\partial_{KK}C(S_0,K,\tau)$ corresponds to the risk-neutral density of the underlying asset price evaluated at K. Thus, we can simply take the second derivative of A.1 to find the implied density.

To compute this density, for a fixed τ, we may consider all but K fixed and constant. As we use a spline representation of the implied volatility we may evaluate the candidate risk-neutral density at any point within the range of available values of K. For simplicity, we set $r=0$, and drop the dependence of $d^\pm(K,\tau)$ on τ. Thus,

$$\frac{\partial}{\partial^2 K}C(S_0,K,\tau) = S_0\frac{\partial}{\partial K}\left\{\underbrace{\phi\left(\bar{d}^+(K)\right)\bar{d}^{+\prime}(K)}_{:=t_1(K)} - \underbrace{\left(\Phi\left(\bar{d}^-(K)\right) + K\,\phi\left(\bar{d}^-(K)\right)\bar{d}^{-\prime}(K)\right)}_{:=t_2(K)}\right\} \tag{A.3}$$

Focusing on the remaining derivatives, we have

$$\frac{\partial}{\partial K}t_1(K) = -\bar{d}^+(K)\phi\left(\bar{d}^+(K)\right)\bar{d}^{+\prime}(K)^2 + \phi\left(\bar{d}^+(K)\right)\bar{d}^{+\prime\prime}(K) \tag{A.4a}$$

$$\frac{\partial}{\partial K}t_2(K) = 2\,\phi\left(\bar{d}^-(K)\right)\bar{d}^{-\prime}(K) - K\phi\left(\bar{d}^-(K)\right)\left[\bar{d}^-(K)\,\bar{d}^{-\prime}(K)^2 - \bar{d}^{-\prime\prime}(K)\right] \tag{A.4b}$$

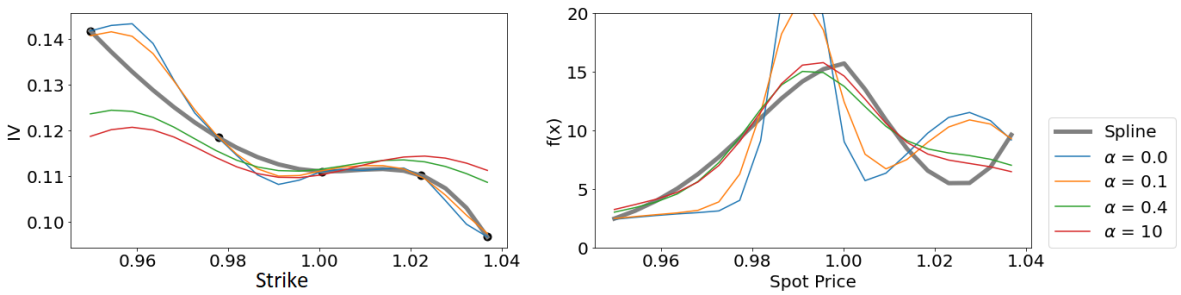


Figure 7: Fits of the CTMC model to the first maturity (1M) of a particular day (2012-05-16) at varying level of the regularizer α. The left panel showcases the fits to the IV surface and the right panel the corresponding risk-neutral density.

and

$$\text{(A.4c)} \quad \bar{d}^{+\prime}(K) = \frac{-1}{K\sigma(K)\sqrt{\tau}} + \frac{\log(K)\,\sigma'(K)}{\sigma^2(K)\sqrt{\tau}} + \frac{1}{2}\sigma'(K)\sqrt{\tau}$$

$$\text{(A.4d)} \quad \bar{d}^{-\prime}(K) = \bar{d}^{+\prime}(K) - \sigma'(K)\sqrt{\tau}$$

$$\text{(A.4e)} \quad \bar{d}^{+\prime\prime}(K) = \frac{\sigma(K) + 2K\sigma'(K)}{(K)^2\,\sigma^2(K)\sqrt{\tau}} + \frac{\log(K)\,\sigma(K)\,\sigma''(K) + 2\,\log(K)\,\sigma'(K)}{\sigma^3(K)\sqrt{\tau}} + \frac{1}{2}\sigma''(K)\sqrt{\tau}$$

$$\text{(A.4f)} \quad \bar{d}^{-\prime\prime}(K) = \bar{d}^{+\prime\prime}(K) - \sigma''(K)\sqrt{\tau}\,.$$

As we use splines for $\sigma(K)$, putting these computations together with (A.3) provides us with the candidate risk-neutral density which we use to regularise the implied volatility fits to.

Appendix B. Effect of α. The parameter α serves as a regularising term to prevent overfitting when the data points in the delta axis are sparse. Figure 7 shows its effects. We have chosen a day that is particularly difficult to fit to exhibit the effects of α. Large values of α (green) correspond to smoother risk-neutral density curves that only deviate slightly from the non-risk-neutral density implied by the spline interpolation at the cost of significantly poor fits to the IV surfaces. In contrast, smaller values of α (blue) typically correspond to rougher risk-neutral densities that often lead to rougher IV surfaces that over-fit to the small number of IV points available. Often, the middle ground (orange), which correctly balances both accuracy and smoothness, is required. In practice, candidate back-testing can determine the optimal choice. We employ a grid search and balance goodness of fit to IV smiles with goodness of fit to the implied risk-neutral densities. For different models, different optimal α are obtained, details are found in Section 5.2. It is important to note that the spline implied density (grey) is not guaranteed to be risk-neutral and, thus, a perfect fit to such densities is sometimes impossible despite choosing a large value of α.

Appendix C. Proofs.

Proof of Proposition 3.1. Denote $v_t := \mathbb{E}^{\mathbb{Q}}[e^{izX_\tau}|\mathcal{F}_t]$. As (X, Z) is Markov, there exists a function $v : \mathfrak{K} \times \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}$, such that $v_t = v^{Z_t}(t, X_t)$. Moreover, applying the Feynman-Kac

theorem, we have that the function $v^k(t,x)$ satisfies the coupled system of PDEs

$$\text{(C.1)}\quad \left(\partial_t + (A^{(n)}_{kk} + \mathcal{L}^{k(n)})\right) v^k(x,t) + \sum_{j\neq k} A^{(n)}_{kj} v^j(x,t) = 0, \quad \forall\, t\in[\tau_{n-1},\tau),\ n\in\mathfrak{N},\ k\in\mathfrak{K},$$

subject to the terminal condition (t.c.) $v^k(\tau,x) = e^{iz\,x}$, and $\mathcal{L}^k$ denotes the infinitesimal generator, given $Z_t = k$, which acts upon twice differentiable functions as follows

$$\text{(C.2)}\qquad \mathcal{L}^{k(n)} f(x) = \left(\mu^n_k - \tfrac{1}{2}(\sigma^n_k)^2\right)\partial_x f(x) + \tfrac{1}{2}(\sigma^n_k)^2 \partial_{xx} f(x)\,.$$

To solve the coupled system of PDEs (C.1), we apply the Fourier transform defined as $\hat{f}(\omega) = \mathcal{F}[f](\omega) := \int_{-\infty}^{\infty} e^{i\omega x} f(x)\,dx$ to both sides of (C.1). Recall that $\mathcal{F}[\partial^n_x f](\omega) = i\,\omega\,\mathcal{F}[\partial^{n-1}_x](\omega) = \cdots = (i\omega)^n\,\mathcal{F}[f](\omega)$. Thus,

$$\text{(C.3)}\qquad \mathcal{F}[\mathcal{L}^{k(n)} v^k](\omega,t) = \left(i\left(\mu^n_k - \tfrac{1}{2}(\sigma^n_k)^2\right)\omega - \tfrac{1}{2}(\sigma^n_k)^2\omega^2\right)\mathcal{F}[v^k](\omega,t).$$

Using the above, and denoting $\hat{v}^k = \mathcal{F}[v^k]$, (C.1) may be written in Fourier space as

$$\text{(C.4)}\quad \left[\partial_t + A^{(n)}_{kk} + \gamma^n_k(\omega)\right]\hat{v}^k(t,\omega) + \sum_j A^{(n)}_{kj}\hat{v}^j(t,\omega) = 0\,, \quad \forall\, t\in[\tau_{n-1},\tau),\ n\in\mathfrak{N},\ k\in\mathfrak{K},$$

s.t. the t.c. $v^k(\tau,x) = \mathcal{D}(z-\omega)$, where $\gamma^n_k(\omega) := i\left(\mu^n_k - \tfrac{1}{2}(\sigma^n_k)^2\right)\omega - \tfrac{1}{2}(\sigma^n_k)^2\omega^2$ and $\mathcal{D}$ is the Dirac delta function. We may further rewrite this system of equations in matrix notation by (i) defining the matrix $\Psi^n(\omega)$ whose entries are $[\Psi(\omega)]_{jk} = \left(A^{(n)}_{kk} + \gamma^n_k(\omega)\right)\delta_{jk} + A^{(n)}_{jk}\,(1-\delta_{jk})$ where δ_{jk} is the Kroencker delta, and (ii) defining the vector of transformed prices $\hat{\boldsymbol{v}}(t,\omega) = (\hat{v}^1(t,\omega),\ldots,\hat{v}^K(t,\omega))^\intercal$. Thus, (C.4) may be written as a vector-valued ODE

$$\text{(C.5)}\qquad (\partial_t + \Psi(\omega))\hat{\boldsymbol{v}}(t,\omega) = \mathbf{0}\,, \qquad\qquad \forall\, t\in[\tau_{n-1},\tau),\ n\in\mathfrak{N},$$

s.t. the t.c. $\hat{\boldsymbol{v}}(\tau,\omega) = \mathcal{D}(\omega - z)\,\mathbf{1}$. This system may be solved explicitly by backward induction. For $t\in[\tau_{N-1},\tau)$ (the last period), the matrix ODE admits the solution $\hat{\boldsymbol{v}}(t,\omega) = e^{(\tau-t)\Psi(\omega)}\mathbf{1}\ \hat{\varphi}(\omega)$. Next, due to continuity, we have that $\hat{\boldsymbol{v}}(\tau_{N-1},\omega) = \lim_{t\downarrow\tau_{N-1}}\hat{\boldsymbol{v}}(t,\omega) = e^{(\tau-\tau_{N-1})\Psi(\omega)}\mathbf{1}\ \hat{\varphi}(\omega)$. Using this limit as the t.c. at $t=\tau_{N-1}$, for $t\in(\tau_{N-2},\tau_{N-1}]$, we solve $(\partial_t + \Psi_{N-1}(\omega))\hat{\boldsymbol{v}}(t,\omega) = \mathbf{0}$, which admits the solution

$$\hat{\boldsymbol{v}}(t,\omega) = e^{(\tau_{N-1}-t)\Psi_{N-1}(\omega)}e^{(\tau-\tau_{N-1})\Psi(\omega)}\mathbf{1}\ \mathcal{D}(\omega-z).$$

Continuing iteratively, we obtain Continuing iteratively we arrive at

$$\text{(C.6)}\qquad \hat{\boldsymbol{v}}(0,\omega) = e^{\tau_1\Psi_1(\omega)}e^{(\tau_2-\tau_1)\Psi_2(\omega)}\ldots e^{(\tau-\tau_{N-1})\Psi(\omega)}\mathbf{1}\ \mathcal{D}(\omega-z)\,.$$

Averaging over the prior π on Z_0, and taking the Fourier inverse (which is trivial due to the Dirac delta function), we obtain the stated result. ∎

Appendix D. Structure of the A matrix. We assume each state k has its corresponding rate parameter λ_k which determines the rate at which the chain will move out of the state. For any state $1\le k\le K$, the chain has an equal chance of moving into the state below $(k-1)$

		Latent Dimension											
		AUD-USD				EUR-USD				CAD-USD			
Model	β	3	5	10	15	3	5	10	15	3	5	10	15
CTMC-CVAE	0.01	3.47	4.11	3.41	3.94	3.19	3.17	3.34	3.16	1.63	1.34	1.12	1.38
	0.1	3.54	3.47	3.46	4.18	2.90	3.58	3.11	3.12	1.08	1.13	1.08	1.22
	1	3.99	3.90	3.27	3.49	2.48	2.23	2.55	2.81	1.18	1.06	1.13	0.91
	10	2.97	3.14	3.26	3.18	1.96	1.89	2.00	1.96	2.00	2.00	1.72	1.96
CTMC-VAE	0.01	4.56	5.35	4.82	4.40	3.88	3.63	3.61	3.97	1.54	2.12	1.78	1.63
	0.1	4.32	5.83	4.54	4.62	3.18	3.29	3.62	2.77	1.81	1.53	1.40	1.34
	1	4.67	4.43	4.13	3.69	3.64	3.94	3.12	3.62	1.72	1.55	1.48	1.70
	10	6.10	5.92	6.40	5.63	4.00	3.59	3.86	3.90	2.98	2.96	3.17	3.06
IV-VAE(B)	0.01	6.07	6.16	6.06	6.26	3.55	4.17	4.05	3.65	1.65	1.61	1.62	1.64
	0.1	6.22	6.34	5.87	5.86	3.94	3.61	3.44	4.04	1.68	1.44	1.59	1.86
	1	6.43	6.55	6.05	5.60	3.88	4.14	4.11	4.08	2.05	1.97	1.83	1.79
	10	6.23	6.21	6.16	6.11	4.37	4.14	3.88	4.28	1.82	1.76	2.32	1.75
CTMC-PCA(B)		5.25	5.34	7.15	8.38	2.51	3.00	3.50	6.15	2.22	1.73	2.40	2.76
Empirical (B)		5.82	5.82	5.82	5.82	3.83	3.83	3.83	3.83	1.67	1.67	1.67	1.67

Table 5: Wasserstein metrics ($\times 10^{-2}$) for varying levels of β, latent dimensionality, and currency pair using the CVAE with the CTMC model with VIX as predictor compared with the vanilla CTMC-VAE and benchmark approaches (B) IV-VAE and CTMC-PCA trained on the period from September 18th, 2012 to May 09, 2016. The Wasserstein metric ($\times 10^{-2}$) between the training and tests sets (Empirical) are presented here for reference.

or above $(k+1)$. We further assume that states form a cyclical graph so that state 1 may transition to state K, and vice versa. Any other transitions will have probability 0. This restricts the process to only being able to move through one state at a time without jumping.

All together, this creates a generator matrix of the form:

$$A^n = \begin{bmatrix} -\lambda_1^n & \frac{\lambda_1^n}{2} & 0 & \dots & \frac{\lambda_1^n}{2} \\ \frac{\lambda_2^n}{2} & -\lambda_2^n & \frac{\lambda_2^n}{2} & 0 & \dots \\ \vdots & \ddots & & & \\ \dots & 0 & \frac{\lambda_{K-1}^n}{2} & -\lambda_{K-1}^n & \frac{\lambda_{K-1}^n}{2} \\ \frac{\lambda_K^n}{2} & \dots & 0 & \frac{\lambda_K^n}{2} & -\lambda_K^n \end{bmatrix} \tag{D.1}$$

which reduces the number of parameters needed to be estimated at each given maturity to $3K$ (K from σ, K from μ, and K from λ) excluding the vector of initial probabilities, where K is the number of possible states of the system.

Appendix E. Additional Tables and Figures.

Model	Parameter	Transforms
CTMC	$\pi_1 \dots \pi_3$	None [1]
	$\mu_1 \dots \mu_3$	$\frac{\mu_i - \bar{\mu}(\mu_i)}{s(\mu_i)}$
	$\sigma_1 \dots \sigma_3$	$\frac{log(\sigma_i) - \bar{\mu}(log(\sigma_i))}{s(log(\sigma_i))}$
	$\lambda_1, \dots \lambda_3$	$\frac{log(\lambda_i) - \bar{\mu}(log(\lambda_i))}{s(log(\lambda_i))}$
DE-JD	σ	$\frac{log(\sigma) - \bar{\mu}(log(\sigma))}{s(log(\sigma)}$
	λ	$\frac{\lambda_i - \bar{\mu}(\lambda_i)}{s(\lambda_i)}$
	p	$\frac{\tilde{p} - \bar{\mu}(\tilde{p})}{s(\tilde{p})}, \quad \tilde{p} = log\left(\frac{p}{1-p}\right)$
	a_1	$\frac{a_1 - \bar{\mu}(a_1)}{s(a_1)}$
	a_2	$\frac{a_2 - \bar{\mu}(a_2)}{s(a_2)}$
GM-JD	σ	$\frac{log(\sigma) - \bar{\mu}(log(\sigma))}{s(log(\sigma)}$
	λ	$\frac{\lambda_i - \bar{\mu}(\lambda_i)}{s(\lambda_i)}$
	$\tilde{\pi}_1, \tilde{\pi}_2$	$\frac{log(\eta_i) - \bar{\mu}(log(\eta_i))}{s(log(\eta_i)}, \quad \eta_1 = 1, \eta_2 = \frac{\tilde{\pi}_2}{\tilde{\pi}_1} \dots$
	$\tilde{\mu}_1, \tilde{\mu}_2$	$\frac{\tilde{\mu}_i - \bar{\mu}(\tilde{\mu}_i)}{s(\tilde{\mu}_i)}$
	$\tilde{\sigma}_1, \tilde{\sigma}_2$	$\frac{log(\tilde{\sigma}_i) - \bar{\mu}(log(\tilde{\sigma}_i))}{s(log(\tilde{\sigma}_i))}$
CGMY/KoBoL	C	$\frac{log(C) - \bar{\mu}(log(C)))}{s(log(C))}$
	G	$\frac{log(G) - \bar{\mu}(log(G)))}{s(log(G))}$
	M	$\frac{log(M) - \bar{\mu}(log(M)))}{s(log(M))}$
	Y	$\frac{log(Y) - \bar{\mu}(log(Y)))}{s(log(Y))}$

Table 6: Normalizations performed on model parameters before input into VAE. $\bar{\mu}(\cdot)$ and $s(\cdot)$ are the empirical mean and standard deviation respectively.

[1] $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$