EXPLICIT AND EFFECTIVELY SYMMETRIC SCHEMES FOR NEURAL SDES

Anonymous authors
Paper under double-blind review

ABSTRACT

Backpropagation through (neural) SDE solvers is traditionally approached in two ways: discretise-then-optimise, which offers accurate gradients but incurs prohibitive memory costs due to storing the full computational graph (even when mitigated by checkpointing); and optimise-then-discretise, which achieves constant memory cost by solving an auxiliary backward SDE, but suffers from slower evaluation and gradient approximation errors. Algebraically reversible solvers promise both memory efficiency and gradient accuracy, yet existing methods such as the Reversible Heun scheme are often unstable under complex models and large step sizes. We address these limitations by introducing a novel class of stable, near-reversible Runge–Kutta schemes for neural SDEs. These *Explicit and Effectively Symmetric (EES)* schemes retain the benefits of reversible solvers while overcoming their instability, enabling memory-efficient training without severe restrictions on step size or model complexity. Through numerical experiments, we demonstrate the superior stability and reliability of our schemes, establishing them as a practical foundation for scalable and accurate training of neural SDEs.

1 INTRODUCTION

Neural stochastic differential equations (SDEs) have recently emerged as a flexible tool for modelling stochastic dynamics, with training typically cast as a distribution-matching problem between generated and observed trajectories. Several approaches have been proposed in the literature, differing mainly in the choice of discriminating divergence. SDE-GANs (Kidger et al., 2021a) use the 1-Wasserstein distance, while Latent SDEs (Li et al., 2020) optimize with respect to the KL divergence via variational inference, and can be viewed as variational autoencoders. Another alternative proposed by Issa et al. (2023) trains neural SDEs non-adversarially using maximum mean discrepancies (MMD) with *signature kernels* (Király & Oberhauser, 2019; Salvi et al., 2021a; Lemercier et al., 2024), a recently introduced family of kernels on path space which received significant attention due to their efficiency in handling path-dependent problems (Salvi et al., 2021b; Lemercier et al., 2021; Pannier & Salvi, 2024; Muça Cirone & Salvi, 2025b).

While effective, the stochastic calculus underpinning SDEs can be technically cumbersome, especially when developing higher order solvers, deriving and analysing backpropagation algorithms, or extending to rougher noises than Brownian motion. A natural way to bypass these limitations is to view SDEs through the lens of *rough paths*. Rough path theory (Lyons, 1998; Gubinelli, 2004; 2010) provides a deterministic calculus on path space that extends classical Itô integration beyond semimartingales, enabling one to treat SDEs as a special case of *rough differential equations (RDEs)*

$$dy(t) \;=\; f_0(t, y_t)\, dt \;+\; f(t, y_t)\, d\mathbf{X}(t), \qquad y_{t_0} = y_0 \in \mathbb{R}^m, \tag{1}$$

where $\mathbf{X}$ is a *rough path* above a driving signal $X = (X^1, ..., X^d) : [0, T] \to \mathbb{R}^d$, $f = (f_1, ..., f_d)$ and $f_i : \mathbb{R}^m \to \mathbb{R}^m$ are vector fields for $i = 0, ..., d$ which determine the system dynamics. In fact, there is a large class of stochastic processes which can be "naturally lifted" to rough paths, including Gaussian processes such as fractional Brownian motion with Hurst parameter $H > 1/4$ and Volterra processes (Friz & Victoir, 2010). Viewing SDEs through the lens of RDEs provides a significant conceptual simplification. In the RDE framework, the driving signal $\mathbf{X}$ is treated as a rough path, which abstracts away the stochastic integral and allows one to work with deterministic calculus on path space. Rough path theory has become a powerful mathematical framework for analyzing modern machine learning models. It has provided the foundation for proving theoretical properties of neural

differential equations (Morrill et al., 2021; Arribas et al., 2020; Cirone et al., 2023; Holberg & Salvi, 2024), sequence-to-sequence architectures (Kidger et al., 2019), and more recently for deep selective state-space models (SSMs) (Muça Cirone et al., 2024; Muça Cirone & Salvi, 2025a; Walker et al., 2025) and score-based diffusion models (Barancikova et al., 2024). for a survey of recent applications of rough path theory to machine learning, we refer the reader to (Fermanian et al., 2023).

From an algorithmic perspective, the RDE formulation also plays a central role in training neural SDEs (NSDEs), where one must perform backpropagation through an SDE solver. For example, the derivation of the *adjoint method*—a key tool for backpropagation through differential equation solvers—becomes far more transparent when formulated in terms of RDEs, avoiding much of the technical machinery required in the stochastic setting; see for instance (Cass & Salvi, 2024). More generally, the RDE viewpoint unifies backpropagation across ODEs, SDEs, and controlled differential equations (CDEs) (Kidger et al., 2020), offering a clean pathwise calculus that is both mathematically rigorous and practically aligned with autodiff frameworks. Several strategies have been proposed. A first approach, known as *discretise-then-optimise*, directly backpropagates through the solver's internal operations. This yields accurate gradients and is computationally efficient, but requires storing all intermediate states, making it memory-intensive. A second approach, *optimise-then-discretise*, instead derives a backwards-in-time adjoint equation and solves it numerically using another call to the solver. This eliminates the need to store intermediate quantities, resulting in constant memory cost with respect to solver depth. However, it typically produces less accurate gradients and is slower due to the need to recompute forward trajectories during the backward pass.

A third option leverages *algebraically reversible solvers*, which enable the exact reconstruction of the solution trajectory of a differential equation from its terminal point, allowing for accurate and memory-efficient backpropagation. In the setting of an autonomous ODE

$$dy_t = f(y_t)dt, \tag{2}$$

a one step method $y_{n+1} = y_n + \Phi_h(y_n)$ is said to be *reversible* or *symmetric* if a step of the method starting from y_1 with a negative step size exactly recovers the initial condition y_0, that is, $\Phi_{-h} = \Phi_h^{-1}$. Whilst reversible schemes offer an efficient approach to backpropagation through differential equations, such schemes are difficult to construct. It is well known that Runge–Kutta schemes are reversible only if they are implicit, making them unsuitable for applications to Neural ODEs. More generally, symmetric parasitism-free general linear methods cannot be explicit (Butcher et al., 2016). To overcome this problem, existing reversible methods proposed in literature, such as the asynchronous leapfrog integrator (ALF) (Zhuang et al., 2021) and the Reversible Heun method (Kidger et al., 2021b), track auxiliary states as part of the integration. Whilst this technique allows for explicit reversible schemes, it comes at the price of low stability. Both ALF and Reversible Heun are well-known to be unstable, and fail when integrating complex equations. For example, Zhang & Chen (2021) find that Reversible Heun is too unstable for their applications, stating:

> *"Regardless the accuracy and memory advantages of Reversible Heun claimed by torchsde, we found this integration approach is less stable compared with simple Euler integration without adjoint and results in numerical issues occasionally. We empirically found that methods without adjoint are more stable and lower loss compared with adjoint ones, even in low dimensional data" (Zhang & Chen, 2021)*

A potential solution to the difficulties of reversible solvers comes with the class of *Explicit and Effectively Symmetric (EES)* Runge–Kutta schemes introduced in Shmelev et al. (2025). Instead of considering exactly reversible schemes, the authors propose schemes which are almost reversible, up to an acceptable tolerance level. Such schemes offer stable, explicit integration methods which are virtually indistinguishable from truly symmetric schemes in practice. The authors demonstrate the efficacy of EES schemes on classical ODEs, and show that EES schemes are capable of producing similar results to classical schemes such as RK3 and RK4. In Shmelev et al. (2025), the formulation of EES schemes is limited to ODEs, without an explicit generalisation to the case of SDEs or RDEs.

Derivative-free *Runge–Kutta (RK) methods* for RDEs were introduced by Redmann & Riedel (2022). Similarly to classical RK schemes for ODEs, the study of these methods is conducted through the formalism of B-series (Hairer et al., 2006; McLachlan et al., 2015; Butcher, 2021). A B-series is an infinite series representation of a method, indexed by (labelled) non-planar rooted trees. A brief overview of B-series is given in Appendix A. A natural consequence of this analysis is that a general

Runge–Kutta method for RDEs is given in terms of tree-iterated integrals of the underlying driving process. In practice, these tree-iterated integrals cannot be simulated directly as their distributions are often intractable. Following (Deya et al., 2012), Redmann & Riedel (2022) replaced these tree-iterated integrals with products of increments of the driving path. This substitution simplifies the derivation of Runge–Kutta coefficients and makes it feasible to establish order conditions up to any desired order.

In this paper, we present a formulation of EES Runge–Kutta schemes for RDEs, and demonstrate their efficacy as integrators for Neural SDEs. The paper is structured as follows. Section 2 gives an overview of existing reversible methods for neural differential equations. Section 3 recounts the framework of Runge–Kutta methods for RDEs introduced in Redmann & Riedel (2020). Section 3 begins by introducing EES schemes for ODEs. Using the framework of Redmann & Riedel (2020), we extend EES schemes to the case of RDEs, and derive results regarding their orders of convergence. Through *mean-square stability* analysis, we show that EES schemes possess a similar stability domain to classical RK3 and RK4 schemes when applied to SDEs, and are significantly more stable than existing reversible schemes designed for Neural SDEs. We end Section 3 by outlining a backpropagation algorithm for explicit RDE Runge–Kutta solvers such as EES schemes. In Section 4 we demonstrate the efficacy of EES schemes as integrators for Neural SDEs with two experiments concerned with the learning of extreme stochastic dynamics where existing reversible solvers fail due to instability, and show that EES successfully overcomes this issue. Section 5 closes this work with a summary of the results, limitations and potential avenues for future work.

2 Existing reversible SDE solvers

The major drawback of classical reversible schemes is their low efficiency. It is well known that Runge–Kutta schemes are reversible only if they are implicit. More generally, symmetric parasitism-free general linear methods cannot be explicit (Butcher et al., 2016). A limited number of efficient reversible solvers have been proposed in the literature. The asynchronous leapfrog integrator (ALF) (Zhuang et al., 2021) for Neural ODEs overcomes the barrier of implicit schemes by tracking an additional state v as part of the integration. Applied to the ODE equation 2, the update rule of the ALF scheme can be written as

$$y_{n+2} = y_n + hf\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}v_n\right),$$
$$v_{n+2} = 2f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}v_n\right) - v_n.$$

A similar approach is taken for SDEs of the form

$$dy_t = g(t, y_t)dt + f(t, y_t)dW_t \tag{3}$$

by the Reversible Heun method (Kidger et al., 2021b), where the integration step reads

$$y_{n+1} = y_n + \frac{1}{2}(g(t_n, v_n) + g(t_{n+1}, v_{n+1}))\Delta t + \frac{1}{2}(f(t_n, v_n) + f(t_{n+1}, v_{n+1}))\Delta W_n,$$
$$v_{n+1} = 2y_n - v_n + g(t_n, v_n)\Delta t + f(t_n, v_n)\Delta W_n.$$

The Reversible Heun method is efficient, requiring only one evaluation of the drift g and the diffusion f per step. However, the method is inherently unstable.

Theorem 2.1. (Kidger et al., 2021b, Theorem D.19) Suppose that the Reversible Heun method is used to obtain a solution $\{y_n, v_n\}_{n\geq 0}$ to the linear test ODE $dy = \lambda y\, dt$, where $\lambda \in \mathbb{C}$ and $y_0 \neq 0$. Then $\{y_n, v_n\}_{n\geq 0}$ is bounded if and only if $\lambda h \in [-i, i]$.

As remarked in Kidger et al. (2021b), this domain is also the absolute stability region for the reversible asynchronous leapfrog integrator (Zhuang et al., 2021). This instability has proven to be a significant bottleneck in certain practical applications. In McCallum & Foster (2024), a method was proposed for transforming any ODE integration method $y_{n+1} = y_n + \Psi_h(t, y_n)$ into one which is reversible, by taking the update

$$y_{n+1} = \lambda y_n + (1-\lambda)v_n + \Psi_h(t, v_n),$$
$$v_{n+1} = v_n - \Psi_{-h}(t_{n+1}, y_{n+1}),$$

for a given coupling parameter $\lambda \in (0, 1]$. The method offers a way of constructing reversible schemes with larger stability domains than those of the ALF and Reversible Heun integrators (McCallum & Foster, 2024, Theorem 2.3). However, the resulting stability domain of the transformed method is typically much smaller than that of the underlying method Ψ, and depends additionally on the coupling parameter λ.

3 Explicit and Effectively Symmetric (EES) schemes for RDEs

We will adopt the general framework of RDEs for our analysis of reversible SDE solvers, following the work of Redmann & Riedel (2020). As discussed in the introduction, a generalised Runge–Kutta scheme for RDEs can be formulated in terms of tree-iterated integrals of the underlying driving rough path. For a brief introduction to these generalised methods, we refer the reader to Appendix B. Computation of such tree-iterated integrals is usually not tractable in practice, and so we adopt the simplified scheme given instead in terms of products of increments of the driving rough path.

3.1 Simplified Runge–Kutta methods for RDEs

Throughout, we will consider rough differential equations (RDEs) of the form

$$dy_t = f(y_t)d\mathbf{X}_t, \tag{4}$$

where $\mathbf{X}$ is an α-Hölder branched rough path for some $\alpha \in (0, 1]$ and f is sufficiently smooth and bounded with bounded derivatives. For an introduction to branched rough paths, we refer the reader to Appendix A.3. Following Redmann & Riedel (2022), we assume that there exist smooth paths $\{X^h\}_{h>0}$ whose natural lifts to branched rough paths $\{\mathbf{X}^h\}_{h>0}$ converge (almost surely) to $\mathbf{X}$ under the metric for α-Hölder rough paths ϱ_α (see Appendix A.3) as $h \to 0$. That is, $\mathbf{X}$ is a geometric rough path. Assume that this Wong-Zakai-type approximation converges at a rate $r_0 > 0$ with respect to the inhomogemeous rough path metric for geometric rough paths ϱ_α^g, such that $\varrho_\alpha^g(\mathbf{X}^h, \mathbf{X}) = \mathcal{O}(h^{r_0})$. For examples of such convergence rates for Gaussian processes, see Friz & Riedel (2014). Let y^h denote the solution associated with the driver $\mathbf{X}^h$,

$$dy_t^h = f(y_t^h)d\mathbf{X}_t^h. \tag{5}$$

A simplified Runge-Kutta scheme for y^h is defined by

$$\begin{aligned} y_{n+1}^h &= y_n^h + \sum_{m=1}^{d}\sum_{i=1}^{s} b_i f_m(k_i) X_{t_n,t_{n+1}}^{(m)}, \\ k_i &= y_n^h + \sum_{m=1}^{d}\sum_{j=1}^{s} a_{ij} f_m(k_j) X_{t_n,t_{n+1}}^{(m)}, \end{aligned} \tag{6}$$

where $X_{t_n,t_{n+1}}$ denotes the increment of X^h over $[t_n, t_{n+1}]$. For simplicity, we will assume equidistant grid points $t_{n+1} - t_n = h$ for all $n \geq 0$. The convergence rates derived in Redmann & Riedel (2020) for schemes of the form in equation 6 are given in Appendix B.1.

Notation 3.1. To avoid confusion, we will generally use Φ to refer to a classical ODE Runge–Kutta method, and Υ to refer to an RDE method of the form given in equation 6. Given an ODE Runge-Kutta scheme Φ, we will write $\mathcal{R}(\Phi)$ to denote the RDE scheme of the form in equation 6 with the same coefficients $\{a_{ij}\}_{1\leq i,j\leq s}$ and $\{b_i\}_{1\leq i\leq s}$ as the ODE scheme.

3.2 EES schemes for ODEs

EES schemes (Shmelev et al., 2025) are a class of Runge–Kutta methods which offer an efficient approach to reversible integration without compromising on stability. Given positive integers $m \geq n$, an explicit Runge–Kutta scheme Φ_h is said to be an $\mathrm{EES}(n, m)$ scheme if Φ_h is of order n and applying the scheme $\Phi_{-h} \circ \Phi_h$ to an ODE recovers the initial condition up to order m. When m is large, such schemes offer near-reversible behaviour, which is often sufficient for practical applications. Butcher tableaux for 3-stage $\mathrm{EES}(2, 5)$ and 4-stage $\mathrm{EES}(2, 7)$ schemes are derived in Shmelev et al. (2025). In Shmelev et al. (2025), the authors focus mostly on $\mathrm{EES}(2, 7)$ schemes, as these

significantly outperform $\mathrm{EES}(2,5)$ schemes as integrators for ODEs. For our applications to Neural SDEs, we will instead restrict ourselves to $\mathrm{EES}(2,5)$, as we do not expect the extra accuracy of $\mathrm{EES}(2,7)$ justifies the extra stage required in this case. Proposition 3.1 gives the general form of the Butcher tableau of a scheme belonging to $\mathrm{EES}(2,5)$ in terms of a parameter x, which we will denote by $\mathrm{EES}(2,5;x)$.

Proposition 3.1 ((Shmelev et al., 2025, Proposition 8.4)). 3-stage Runge–Kutta schemes belonging to $\mathrm{EES}(2,5)$ have a Butcher tableau of the form:

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1+2x}{4(1-x)} & \frac{1+2x}{4(1-x)} & & \\ \frac{3}{4(1-x)} & \frac{(4x-1)^2}{4(x-1)(1-4x^2)} & \frac{1-x}{(1-4x^2)} & \\ \hline & x & \frac{1}{2} & \frac{1}{2}-x \end{array} \tag{7}$$

for some $x \in \mathbb{R}$, $x \neq 1, \pm\frac{1}{2}$.

Recall from Theorem 2.1 that the stability domain of Reversible Heun is the interval $[-i, i]$, which is also the stability domain of the ALF integrator.The stability region for $\mathrm{EES}(2,5)$ schemes is significantly larger and is comparable to classical methods such as Kutta's RK4, as shown in Figure 1. Theorem 3.1 gives the exact form of this region for $\mathrm{EES}(2,5;x)$.

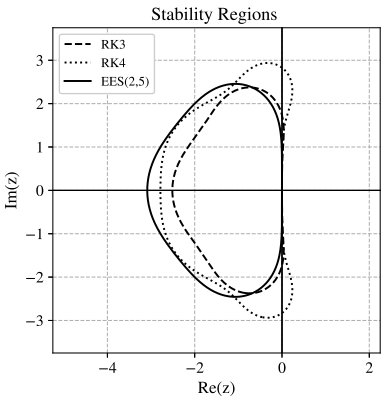

Figure 1: *Stability domain for* $\mathrm{EES}(2,5;1/10)$ *compared to Kutta's RK3 and RK4.*

Theorem 3.1. Suppose that, for some $x \neq 1, \pm\frac{1}{2}$, $\mathrm{EES}(2,5;x)$ is used to obtain a solution $\{y_n\}_{n\geq 0}$ to the linear test equation $dy = \lambda y\, dt$, where $\lambda \in \mathbb{C}$ and $y_0 \neq 0$. Then $y_n \to 0$ as $n \to \infty$ if and only if

$$\left|1 + \rho + \frac{1}{2}\rho^2 + \frac{1}{8}\rho^3\right| < 1,$$

where $\rho = \lambda h$.

The result follows from the fact that Runge–Kutta methods applied to linear test equations admit a linear update rule, $y_{n+1} = R(\rho)y_n$, where $R(\rho)$ is the stability function associated with the scheme. As such, the proof of Theorem 3.1 is simply a direct computation of the stability function $R(\rho) = 1 + \rho + \frac{1}{2}\rho^2 + \frac{1}{8}\rho^3$ for $\mathrm{EES}(2,5;x)$, which we omit here. We note that the stability region of $\mathrm{EES}(2,5;x)$ is completely independent of the choice of the parameter x. Motivated by the discussion in Shmelev et al. (2025, Section 8.1), we choose to fix the parameter $x = 1/10$ and refer to $\mathrm{EES}(2,5;1/10)$ as *the* $\mathrm{EES}(2,5)$ scheme from now on.

3.3 EES SCHEMES FOR RDES

We aim to define an analogue of EES schemes for RDEs and give convergence rates for the local and global errors of these schemes. To reason about the reversibility of schemes, it will be convenient to adopt the following notation.

Notation 3.2. Let $\{y_n^h\}_{i=0,\dots,N}$ denote the solution of a Runge–Kutta scheme $y_{n+1}^h = y_n^h + \Upsilon(y_n^h, X_{t_n,t_{n+1}})$ of the form given in equation 6 applied to equation 5. Let $\overleftarrow{\Upsilon}$ denote the reverse-time scheme, $\overleftarrow{\Upsilon}(\cdot, X_{t_n,t_{n+1}}) := \Upsilon(\cdot, -X_{t_n,t_{n+1}})$, and let $\overleftarrow{y}_n^h$ denote the result of an n-fold application of $\overleftarrow{\Upsilon}$ to y_n^h, such that $\{\overleftarrow{y}_n^h\}_{i=0,\dots,N}$ is a sequence of approximations to the initial condition y_0.

We may now define a class of EES schemes for RDEs, which we will denote by $\mathrm{EES}_{\mathcal{R}}$. In a similar fashion to Shmelev et al. (2025), we will define the schemes in terms of both the local error of the solution and the local error of the recovered initial condition when the scheme is run in reverse.

Definition 3.1. When applied to equation 5, a Runge–Kutta scheme $y^h_{n+1} = y^h_n + \Upsilon(y^h_n, \mathbf{X}_{t_n,t_{n+1}})$ of the form equation 6 is said to belong to $\mathrm{EES}_{\mathcal{R}}(n,m)$ for $m \geq n$ if

$$y^h(t_1) - y^h_1 = \mathcal{O}(h^{(n+1)\alpha}), \quad y_0 - \overleftarrow{y}^h_1 = \mathcal{O}(h^{(m+1)\alpha}).$$

As a consequence of Redmann & Riedel (2020, Theorem 3.3), $\mathrm{EES}_{\mathcal{R}}$ schemes can be constructed directly from EES schemes by using the same coefficients. A global error rate for the schemes is given below as a corollary of (Redmann & Riedel, 2020, Theorem 4.2), noting that the error in the recovery of y_0 from $\overleftarrow{y}^h_n$ is independent of the chosen Wong-Zakai approximation, and as such independent of r_0. Numerical experiments verifying the global rates on a test RDE driven by a fractional Brownian motion are presented in Appendix C.

Theorem 3.2. Let $\Phi \in \mathrm{EES}(n,m)$ for some $m \geq n$. Then $\mathcal{R}(\Phi) \in \mathrm{EES}_{\mathcal{R}}(n,m)$.

Proof. Since Φ is of order n, it follows from (Redmann & Riedel, 2020, Theorem 3.3) that $\mathcal{R}(\Phi)$ has a local error of order $(n+1)\alpha$. Consider the scheme $\mathcal{R}(\overleftarrow{\Phi}) \circ \mathcal{R}(\Phi) = \mathcal{R}(\overleftarrow{\Phi} \circ \Phi)$. By definition of Φ, the scheme $\overleftarrow{\Phi} \circ \Phi$ admits a B-series expansion $y_0 + \mathcal{O}(h^{m+1})$ (Shmelev et al., 2025). It follows from Redmann & Riedel (2020) that the corresponding B-series expansion for $\mathcal{R}(\overleftarrow{\Phi} \circ \Phi)$ is $y_0 + \mathcal{O}(h^{(m+1)\alpha})$. □

Theorem 3.3. Let $\Upsilon \in \mathrm{EES}_{\mathcal{R}}(n,m)$ for some $m \geq n$. Then

$$\max_{i=0,\dots,N} |y(t_i) - y_i| = \mathcal{O}(h^{\eta_1}), \quad \max_{i=0,\dots,N} |y_0 - \overleftarrow{y}_i| = \mathcal{O}(h^{\eta_2}),$$

where $\eta = \min\{r_0, (n+1)\alpha - 1\}$ and $\eta_2 = (m+1)\alpha - 1$.

Proof. Since Φ is of order n, it follows from (Redmann & Riedel, 2020, Theorem 4.2) that $\mathcal{R}(\Phi)$ has a global error of order $\eta_1 = \min\{r_0, (n+1)\alpha - 1\}$. The global rate for the recovery of the initial condition y_0 from $\overleftarrow{y}_i$ follows immediately from Redmann & Riedel (2020, Proposition 4.1). □

3.4 Stability of EES schemes for SDEs

As discussed in the introduction and subsequently in Section 3, EES schemes offer a stable alternative to reversible integration. Whilst the stability of EES in the case of ODEs has been studied in Shmelev et al. (2025) and Section 3, we are interested in the stability of $\mathrm{EES}_{\mathcal{R}}$ when applied to stochastic drivers for our applications to Neural SDEs. To evaluate the stability in the context of SDEs, we consider the *mean-square stability*, which has been widely used for the analysis of stochastic integration methods in the literature (Higham, 2000; Drummond & Mortimer, 1991; Hernandez & Spigler, 1993; Komori & Mitsui, 1995; Komori et al., 1994; Petersen, 1998; Saito & Mitsui, 1993; 1996; Schurz, 1996). Given the test equations

$$dy_t = \lambda y_t dt + \mu y_t dW_t, \tag{8}$$

where $\lambda, \mu \in \mathbb{C}$ and $y_0 \neq 0$ almost surely, a solution $\{y_n\}_{n\geq 0}$ derived from a numerical integrator is said to be *mean-square stable* if $\lim_{n\to\infty} \mathbb{E}(|y_n|^2) = 0$. It follows in a similar fashion to Theorem 3.1 that $\mathrm{EES}(2,5;x)$ applied to 8 is mean-square stable if and only if

$$\mathbb{E}\left[\left|1 + \rho + \frac{1}{2}\rho^2 + \frac{1}{8}\rho^3\right|^2\right] < 1,$$

where $\rho = \lambda dt + \mu dW_t \sim N(\lambda dt, \mu^2 dt)$. Figure 2 shows 4 cross-sections of the stability domain. For comparison, we take the RDE analogues of RK3 and RK4 of the form in equation 6, $\mathcal{R}(\mathrm{RK3})$ and $\mathcal{R}(\mathrm{RK4})$. Along most cross-sections, $\mathrm{EES}_{\mathcal{R}}(2,5)$ achieves a similar or greater stability to $\mathcal{R}(\mathrm{RK3})$ and $\mathcal{R}(\mathrm{RK4})$.

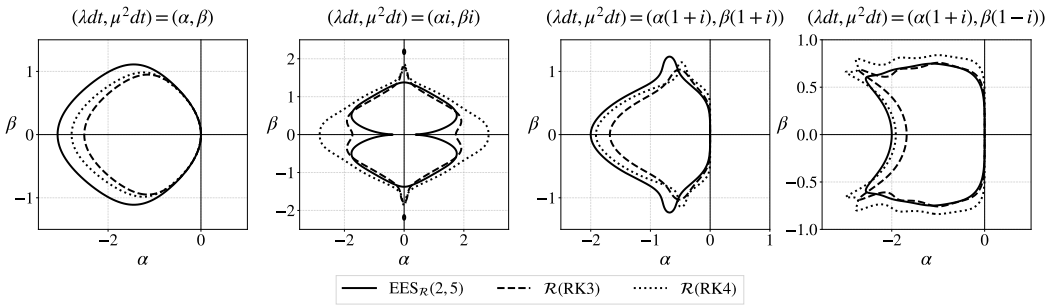

Figure 2: *Cross sections of the mean-square stability domains of* $\mathrm{EES}_{\mathcal{R}}(2,5)$*,* $\mathcal{R}$*(RK3) and* $\mathcal{R}$*(RK4).*

3.5 Backpropagation through explicit Runge–Kutta methods

The algorithm for backpropagation through an explicit Runge–Kutta scheme Υ of the form in equation 6 is given in Algorithm 1. We assume the solver is applied to a (neural) RDE of the form

$$dy_t^h = f(y_t^h; \theta) d\mathbf{X}_t^h, \tag{9}$$

where θ are learnable parameters requiring backpropagation, trained with respect to a loss $L(\{y_n^h\}_{n=0}^N)$. As with all reversible schemes, a reverse step $\overleftarrow{\Upsilon}$ is used to recover y_n from y_{n+1}, followed by a backpropagation through the internal operations of the solver Υ. The latter step is achieved by defining $z_i = f(k_i; \theta)$ and computing the derivatives $\partial L/\partial z_i$ and $\partial L/\partial k_i$ in reverse through the stages $i = s, s-1, \ldots, 1$. At each stage, a backpropagation algorithm is called to backpropagate the derivative $\partial L/\partial z_i$ through f, resulting in the derivative $\partial L/\partial k_i$ and a local derivative with respect to θ, d_θ.

Algorithm 1 Backpropagation through Explicit Runge–Kutta Schemes

Input: $y_{n+1}, \partial_{y_{n+1}} L$
Input: Running derivative with respect to θ, $\partial_\theta L$
Input: Explicit RK method Υ of the form in equation 6 with coefficients $\{a_{ij}\}_{1 \le i,j \le s}$ and $\{b_i\}_{1 \le i \le s}$.

$y_n = \overleftarrow{\Upsilon}(y_{n+1}, dX)$
for $i = s, \ldots 1$ **do**
  $\partial_{z_i} L = b_i dX \cdot \partial_{y_{n+1}} L + \sum_{j=i+1}^{s} a_{ji} dX \cdot \partial_{k_j} L$
  $d_\theta, \partial_{k_i} L = \mathrm{backprop}_f(\partial_{z_i} L)$
  $\partial_\theta L \mathrel{+}= d_\theta$
end for
$\partial_{y_n} L = \partial_{y_{n+1}} L + \sum_{i=1}^{s} \partial_{k_i} L$
return $y_n, \partial_{y_n} L, \partial_\theta L$

4 Experiments

We evaluate the performance of $\mathrm{EES}_{\mathcal{R}}(2,5)$ on examples of Neural SDEs with challenging dynamics. We note that there are limited solvers which can be used as baselines in our experiments, with Reversible Heun being the only widely adopted explicit reversible SDE solver at the time of writing. In order to expand our baselines, we take the approach of McCallum-Foster for constructing reversible ODE solvers and apply it to the RDE versions of the Euler and Explicit Midpoint schemes, defined by the form in equation 6. Algorithm 1 can then be used in conjunction with the backpropagation algorithm found in McCallum & Foster (2024) to backpropagate efficiently through the resulting schemes.

We present the results of training Neural SDEs on OU and GBM dynamics. In both cases, we pick the step sizes of the solvers to fix the total number of function evaluations of the drift and diffusion, leading to comparable runtimes for all of the solvers.

4.1 HIGH VOLATILITY ORNSTEIN–UHLENBECK PROCESS

Consider learning the Ornstein–Uhlenbeck (OU) dynamics

$$dy_t = \nu(\mu - y_t)dt + \sigma dW_t, \quad y_0 \in \mathbb{R},$$

under a high-volatility regime $\sigma \gg 0$. Specifically, we take $\nu = 0.2, \mu = 0.1$ and $\sigma = 2$. Motivated by Oh et al. (2024), we take a Neural Langevin SDE (LSDE) defined by

$$dz_t = g(z_t; \theta_g)dt + f(t; \theta_f) \circ dW_t, \quad z_0 = h(\mathbf{x}, \theta_h) \in \mathbb{R}^{d_z},$$

where h is a learnable affine function of the input data $\mathbf{x} = \{x_n\}_{n \geq 0}$, $x_n \in \mathbb{R}^2$, sampled from the true OU dynamics, and g, f are neural networks parametrised by θ_g, θ_f respectively. We choose the dimension of the latent representation $d_z = 32$, and parametrise f, g as 2-layer neural networks of width 32 with LipSwish activations. The SDEs are integrated over $t \in [0, 10]$ and the LSDE is trained for 250 epochs using the Adam optimiser with a fixed learning rate of 10^{-3}. At each epoch, 50,000 realisations of the trained dynamics are sampled and the MSE loss is computed against the true OU dynamics.

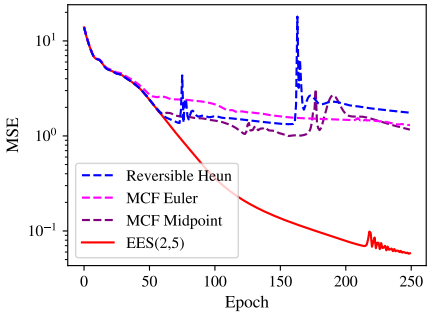

Figure 3: *Training MSE for OU dynamics with a fixed number of evaluations of f, g.*

Figure 3 shows the training loss using Reversible Heun, McCallum-Foster (MCF) methods and $\text{EES}_{\mathcal{R}}(2, 5)$, with the step size chosen such that the number of evaluations of f, g is fixed between solvers. Such a choice results in comparable runtimes for all of the solvers, allowing for a fair comparison. Table 1 gives the number of evaluations of f, g per step of the solvers, the chosen step size, the terminal MSE and the total runtime of each solver. From Figure 3, we see that for the initial ~ 50 epochs, the methods perform similarly. After this, $\text{EES}_{\mathcal{R}}(2, 5)$ significantly outperforms the other methods, suggesting the model has begun to learn high-volatility dynamics which cause instability in the Reversible Heun and McCallum-Foster methods.

Method	#Eval. / Step	Step Size	Terminal MSE	Runtime (s)
Reversible Heun	1	1/12	1.0190	368.2
MCF Euler	2	1/6	1.3048	307.5
MCF Midpoint	4	1/3	1.1651	279.5
$\text{EES}_{\mathcal{R}}(2, 5)$	3	1/4	0.0582	261.3

Table 1: Metrics for OU dynamics. The step size is chosen such that the total number of evaluations of f, g per integration is fixed.

4.2 HIGH-DIMENSIONAL GBM WITH STIFF DRIFT

Consider learning the dynamics of a high-dimensional geometric Brownian motion (GBM)

$$dy_t = Ay_t dt + \sigma y_t dW_t, \quad y_0 \in \mathbb{R}^d,$$

where $A \in \mathbb{R}^{d \times d}$ and $\sigma \in \mathbb{R}$. We introduce a stiff drift component by choosing $A = QDQ^T$, where $D = \text{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{d-1})$, $\lambda_i = -20(1 + \frac{i}{d})$, and Q is a randomly generated orthogonal matrix, and take $d = 25$ and $\sigma = 0.1$. We choose to learn the dynamics using a Neural SDE of the form

$$dz_t = g(z_t; \theta_g)dt + f(z_t; \theta_f) \circ dW_t, \quad z_0 = h(\mathbf{x}, \theta_h) \in \mathbb{R}^{d_z},$$

where f, g are neural networks with the same architecture as Section 4.1. We integrate the SDEs over $t \in [0, 1]$ for 1,000 epochs, sampling 10,000 realisations of the dynamics at every epoch. The Adam optimiser is used with a fixed learning rate of 2×10^{-1}.

As in the previous example, Figure 4 and Table 2 show the results of training using various reversible methods, with the step size chosen such that the number of evaluations of f, g is fixed. We see that the

instability caused by the stiff drift results in diverging MSE for all solvers except $\mathrm{EES}_{\mathcal{R}}(2,5)$, which manages to retain moderate stability for the entire 1,000 epochs of training. Figure 5 shows the MSE of the gradient of the loss during training, where the true gradient is computed by autodifferentiation through a discretise-then-optimise solution, using the same solver and step size. Despite its near-reversibility, $\mathrm{EES}_{\mathcal{R}}(2,5)$ achieves a lower gradient MSE compared to other solvers. This effect is likely the result of the superior stability of $\mathrm{EES}_{\mathcal{R}}(2,5)$, in combination with the linear nature of the target SDE.

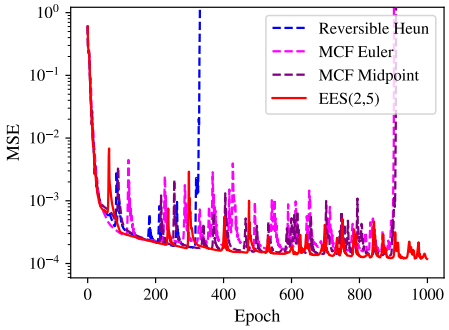

Figure 4: *Training MSE for GBM dynamics with a fixed number of evaluations of f, g.*

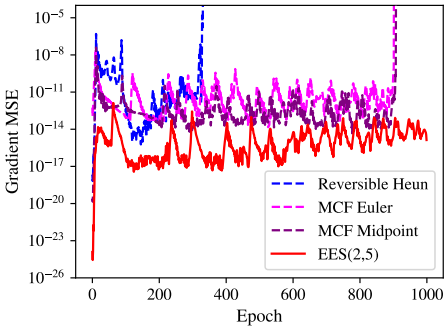

Figure 5: *Gradient MSE for GBM dynamics with a fixed number of evaluations of f, g.*

Method	#Eval. / Step	Step Size	Terminal MSE	Runtime (s)
Reversible Heun	1	1/60	-	1283.6
MCF Euler	2	1/30	-	1119.9
MCF Midpoint	4	1/15	-	1270.1
$\mathrm{EES}_{\mathcal{R}}(2,5)$	3	1/20	1.1803E-4	1050.0

Table 2: Metrics for stiff GBM dynamics. The step size is chosen such that the total number of evaluations of f, g per integration is fixed.

5 Conclusions, Limitations and Future Work

In this paper, we have introduced the use of Explicit and Effectively Symmetric (EES) schemes as stable integrators for Neural SDEs, and more generally, RDEs. Using the framework of Redmann & Riedel (2020), we have adapted existing EES schemes introduced by Shmelev et al. (2025) for ODEs to the more general setting of RDEs. Through *mean-square stability* analysis, we have shown that EES schemes possess similar stability domains to classical RK3 and RK4 schemes when applied to SDEs. We discussed an efficient algorithm for backpropagation through explicit Runge–Kutta schemes for RDEs, and presented two experiments involving the training of Neural ODEs on extreme SDE dynamics. In both experiments, our $\mathrm{EES}_{\mathcal{R}}(2,5)$ scheme demonstrated superior stability to the Reversible Heun scheme, resulting in faster training and a lower terminal loss.

EES schemes provide a stable alternative to existing methods such as Reversible Heun, resulting in fast and accurate training when applied to complex dynamics. However, when stability is not an issue, Reversible Heun offers faster integration by requiring only one evaluation of the drift and diffusion per step, as opposed to 3 evaluations required by $\mathrm{EES}_{\mathcal{R}}$. Addressing this limitation, either through algorithmic changes or the derivation of new EES-type schemes which employ auxiliary variables, is left for future work.

There are several potential extensions to this paper which are left for future research. Applications of EES schemes to more complicated models, including but not limited to Neural Jump SDEs (Jia & Benson, 2019; Herrera et al., 2020), Neural CDEs (Kidger et al., 2020) and Neural RDEs (Morrill et al., 2021), may be of interest. An extension of EES schemes to include partitioned or adaptive step-size schemes would prove valuable for the training of stiff neural differential equations.

REFERENCES

Imanol Perez Arribas, Cristopher Salvi, and Lukasz Szpruch. Sig-sdes model for quantitative finance. In *ACM International Conference on AI in Finance*, 2020.

Barbora Barancikova, Zhuoyue Huang, and Cristopher Salvi. Sigdiffusions: Score-based diffusion models for time series via log-signature embeddings. In *The Thirteenth International Conference on Learning Representations*, 2024.

Kevin Burrage and Pamela M Burrage. Order conditions of stochastic runge–kutta methods by b-series. *SIAM Journal on Numerical Analysis*, 38(5):1626–1646, 2000.

Kevin Burrage and Pamela Marion Burrage. High strong order explicit runge-kutta methods for stochastic ordinary differential equations. *Applied Numerical Mathematics*, 22(1-3):81–101, 1996.

Kevin Burrage and PM Burrage. General order conditions for stochastic runge-kutta methods for both commuting and non-commuting stochastic ordinary differential equation systems. *Applied Numerical Mathematics*, 28(2-4):161–177, 1998.

JC Butcher, AT Hill, and TJT Norton. Symmetric general linear methods. *BIT Numerical Mathematics*, 56:1189–1212, 2016.

John C Butcher. *B-series: algebraic analysis of numerical methods*, volume 55. Springer, 2021.

John C Butcher, Taketomo Mitsui, Yuto Miyatake, and Shun Sato. On the B-series composition theorem. *arXiv preprint arXiv:2409.08533*, 2024.

John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

Thomas Cass and Cristopher Salvi. Lecture Notes on Rough Paths and Applications to Machine Learning, 2024.

Nicola Muca Cirone, Maud Lemercier, and Cristopher Salvi. Neural signature kernels as infinite-width-depth-limits of controlled resnets. In *International Conference on Machine Learning*, pp. 25358–25425. PMLR, 2023.

Alain Connes and Dirk Kreimer. Hopf algebras, renormalization and noncommutative geometry. In *Quantum field theory: perspective and prospective*, pp. 59–109. Springer, 1999.

Aurélien Deya, Andreas Neuenkirch, and Samy Tindel. A milstein-type scheme without lévy area terms for sdes driven by fractional brownian motion. In *Annales de l'IHP Probabilités et statistiques*, volume 48, pp. 518–550, 2012.

PD Drummond and IK Mortimer. Computer simulations of multiplicative stochastic differential equations. *Journal of computational physics*, 93(1):144–170, 1991.

Adeline Fermanian, Terry Lyons, James Morrill, and Cristopher Salvi. New directions in the applications of rough path theory. *IEEE BITS the Information Theory Magazine*, 3(2):41–53, 2023.

Peter Friz and Sebastian Riedel. Convergence rates for the full gaussian rough paths. In *Annales de l'IHP Probabilités et statistiques*, volume 50, pp. 154–194, 2014.

Peter K Friz and Nicolas B Victoir. *Multidimensional stochastic processes as rough paths: theory and applications*, volume 120. Cambridge University Press, 2010.

Massimiliano Gubinelli. Controlling rough paths. *Journal of Functional Analysis*, 216(1):86–140, 2004.

Massimiliano Gubinelli. Ramification of rough paths. *Journal of Differential Equations*, 248(4): 693–721, 2010.

E. Hairer and G. Wanner. On the Butcher group and general multi-value methods. *Computing (Arch. Elektron. Rechnen)*, 13(1):1–15, 1974. ISSN 0010-485X,1436-5057. doi: 10.1007/bf02268387. URL https://doi.org/10.1007/bf02268387.

Ernst Hairer, Marlis Hochbruck, Arieh Iserles, and Christian Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

Martin Hairer and David Kelly. Geometric versus non-geometric rough paths. In *Annales de l'IHP Probabilités et statistiques*, volume 51, pp. 207–251, 2015.

Diego Bricio Hernandez and Renato Spigler. Convergence and stability of implicit runge-kutta methods for systems with multiplicative noise. *BIT Numerical Mathematics*, 33(4):654–669, 1993.

Calypso Herrera, Florian Krach, and Josef Teichmann. Neural jump ordinary differential equations: Consistent continuous-time prediction and filtering. *arXiv preprint arXiv:2006.04727*, 2020.

Desmond J Higham. Mean-square and asymptotic stability of the stochastic theta method. *SIAM journal on numerical analysis*, 38(3):753–769, 2000.

Michael E. Hoffman. Combinatorics of rooted trees and Hopf algebras. *Trans. Amer. Math. Soc.*, 355 (9):3795–3811, 2003. ISSN 0002-9947,1088-6850.

Christian Holberg and Cristopher Salvi. Exact gradients for stochastic spiking neural networks driven by rough signals. *Advances in Neural Information Processing Systems*, 37:31907–31939, 2024.

Zacharia Issa, Blanka Horvath, Maud Lemercier, and Cristopher Salvi. Non-adversarial training of neural sdes with signature kernel scores. *Advances in Neural Information Processing Systems*, 36: 11102–11126, 2023.

Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.

Patrick Kidger, Patric Bonnier, Imanol Perez Arribas, Cristopher Salvi, and Terry Lyons. Deep signature transforms. *Advances in neural information processing systems*, 32, 2019.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020.

Patrick Kidger, James Foster, Xuechen Li, and Terry J Lyons. Neural sdes as infinite-dimensional gans. In *International conference on machine learning*, pp. 5453–5463. PMLR, 2021a.

Patrick Kidger, James Foster, Xuechen Chen Li, and Terry Lyons. Efficient and accurate gradients for neural sdes. *Advances in Neural Information Processing Systems*, 34:18747–18761, 2021b.

Franz J Király and Harald Oberhauser. Kernels for sequentially ordered data. *Journal of Machine Learning Research*, 20(31):1–45, 2019.

Y Komori, Y Saito, and T Mitsui. Some issues in discrete approximate solution for stochastic differential equations. *Computers & Mathematics with Applications*, 28(10-12):269–278, 1994.

Yoshio Komori and Taketomo Mitsui. Stable row-type weak scheme for stochastic differential equations. *Monte Carlo Methods and Applications*, 1:279–300, 01 1995.

Maud Lemercier, Cristopher Salvi, Theodoros Damoulas, Edwin Bonilla, and Terry Lyons. Distribution regression for sequential data. In *International Conference on Artificial Intelligence and Statistics*, pp. 3754–3762. PMLR, 2021.

Maud Lemercier, Terry Lyons, and Cristopher Salvi. Log-pde methods for rough signature kernels. *arXiv preprint arXiv:2404.02926*, 2024.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David K Duvenaud. Scalable gradients and variational inference for stochastic differential equations. In *Symposium on Advances in Approximate Bayesian Inference*, pp. 1–28. PMLR, 2020.

Terry J Lyons. Differential equations driven by rough signals. *Revista Matemática Iberoamericana*, 14(2):215–310, 1998.

Dominique Manchon. Hopf algebras, from basics to applications to renormalization. *arXiv preprint math/0408405*, 2004.

Sam McCallum and James Foster. Efficient, accurate and stable gradients for neural odes. *arXiv preprint arXiv:2410.11648*, 2024.

Robert I McLachlan, Klas Modin, Hans Munthe-Kaas, and Olivier Verdier. Butcher series: a story of rooted trees and numerical methods for evolution equations. *arXiv preprint arXiv:1512.00906*, 2015.

James Morrill, Cristopher Salvi, Patrick Kidger, and James Foster. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pp. 7829–7838. PMLR, 2021.

Nicola Muça Cirone and Cristopher Salvi. Parallelflow: Parallelizing linear transformers via flow discretization, 2025a.

Nicola Muça Cirone and Cristopher Salvi. Rough kernel hedging, 2025b.

Nicola Muça Cirone, Antonio Orvieto, Benjamin Walker, Cristopher Salvi, and Terry Lyons. Theoretical foundations of deep selective state-space models, 2024.

YongKyung Oh, Dong-Young Lim, and Sungil Kim. Stable neural stochastic differential equations in analyzing irregular time series data. *arXiv preprint arXiv:2402.14989*, 2024.

Alexandre Pannier and Cristopher Salvi. A path-dependent pde solver based on signature kernels, 2024.

WP Petersen. A general implicit splitting for stabilizing numerical simulations of itô stochastic differential equations. *SIAM journal on numerical analysis*, 35(4):1439–1451, 1998.

Martin Redmann and Sebastian Riedel. Runge-kutta methods for rough differential equations. *arXiv preprint arXiv:2003.12626*, 2020.

Martin Redmann and Sebastian Riedel. Runge-kutta methods for rough differential equations. *Journal of Stochastic Analysis*, 3(4):6, 2022.

Yoshihiro Saito and Taketomo Mitsui. T-stability of numerial scheme for stochastic differential equations. In *Contributions in numerical mathematics*, pp. 333–344. World Scientific, 1993.

Yoshihiro Saito and Taketomo Mitsui. Stability analysis of numerical schemes for stochastic differential equations. *SIAM Journal on Numerical Analysis*, 33(6):2254–2267, 1996.

Cristopher Salvi, Thomas Cass, James Foster, Terry Lyons, and Weixin Yang. The signature kernel is the solution of a goursat pde. *SIAM Journal on Mathematics of Data Science*, 3(3):873–899, 2021a.

Cristopher Salvi, Maud Lemercier, Chong Liu, Blanka Horvath, Theodoros Damoulas, and Terry Lyons. Higher order kernel mean embeddings to capture filtrations of stochastic processes. *Advances in Neural Information Processing Systems*, 34:16635–16647, 2021b.

Henri Schurz. Asymptotical mean square stability of an equilibrium point of some linear numerical solutions with multiplicative noise. *Stochastic Analysis and Applications*, 14(3):313–353, 1996.

Daniil Shmelev, Kurusch Ebrahimi-Fard, Nikolas Tapia, and Cristopher Salvi. Explicit and effectively symmetric runge-kutta methods. *arXiv preprint arXiv:2507.21006*, 2025.

Benjamin Walker, Lingyi Yang, Nicola Muca Cirone, Cristopher Salvi, and Terry Lyons. Structured linear cdes: Maximally expressive and parallel-in-time sequence models. *arXiv preprint arXiv:2505.17761*, 2025.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. *arXiv preprint arXiv:2111.15141*, 2021.

Juntang Zhuang, Nicha C Dvornek, Sekhar Tatikonda, and James S Duncan. Mali: A memory efficient and reverse accurate integrator for neural odes. *arXiv preprint arXiv:2102.04668*, 2021.

A ROOTED TREES AND B-SERIES

A.1 THE CONNES-KREIMER HOPF ALGEBRA

We give a brief account of non-planar (labelled) rooted trees and the Connes-Kreimer Hopf algebra. We refer the reader to Hoffman (2003) for a comprehensive presentation. A non-planar labelled rooted tree is defined as a graph $\tau = (V, E, r)$ with vertex set V, edge set E and a root vertex $r \in V$, together with a set of vertex decorations drawn from $\{1, \dots, d\}$. We denote the empty tree by $\emptyset$. Given trees $\tau_1, \dots, \tau_m$, we write $[\tau_1, \dots, \tau_m]_a$ to denote the tree formed by connecting the root vertices of $\tau_1, \dots, \tau_m$ to a new root, which receives the label $a \in \{1, \dots, d\}$. Non-planarity means that the order of the trees in $[\tau_1, \dots, \tau_m]_a$ is irrelevant. Repeated trees will be denoted using power notation, for instance

$$[\tau_1, \tau_1, \tau_2, \tau_3, \tau_3, \tau_3]_a = [\tau_1^2, \tau_2, \tau_3^3]_a.$$

We write $|\tau|$ to denote the number of vertices in a tree. Additionally, we define the following combinatorial quantities, defined on unlabelled trees:

$$\emptyset! = 1, \quad \bullet! = 1, \quad [\tau_1, \dots, \tau_m]! = |[\tau_1, \dots, \tau_m]| \prod_{i=1}^{m} \tau_i!,$$

$$\sigma(\emptyset) = 1, \quad \sigma(\bullet) = 1, \quad \sigma([\tau_1^{k_1}, \dots, \tau_m^{k_m}]) = \prod_{i=1}^{m} k_i! \sigma(\tau_i)^{k_i},$$

$$\beta(\emptyset) = 1, \quad \beta(\bullet) = 1, \quad \beta([\tau_1^{k_1}, \dots, \tau_m^{k_m}]) = \binom{[\tau_1^{k_1}, \dots, \tau_m^{k_m}]}{|\tau_1|, \dots, |\tau_m|} \prod_{i=1}^{m} \frac{1}{k_i!} \beta(\tau_i)^{k_i}.$$

We will refer to the commutative juxtaposition of trees as a forest. We write $\mathcal{T}$ to denote the set of all non-planar labelled rooted trees, and $\mathcal{T}_N \subset \mathcal{T}$ to denote the trees τ with $|\tau| \leq N$. The free commutative $\mathbb{R}$-algebra generated by $\mathcal{T}$ will be denoted $\mathcal{H}$. The Connes-Kreimer (Connes & Kreimer, 1999) Hopf algebra on $\mathcal{H}$ is defined as follows. Multiplication $\mu : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H}$ is defined as the commutative juxtaposition of two forests, extended linearly to $\mathcal{H}$. The multiplicative unit is defined to be the empty forest $\emptyset$. The counit map $\varepsilon : \mathcal{H} \to \mathbb{R}$ is defined by $\varepsilon(\emptyset) = 1$ and $\varepsilon(\tau) = 0$ for all non-empty trees $\tau \in \mathcal{H}$. The coproduct map is defined recursively by

$$\Delta(\emptyset) = \emptyset \otimes \emptyset,$$
$$\Delta[\tau_1, \dots, \tau_m]_a = [\tau_1, \dots, \tau_m]_a \otimes \emptyset + (\mathrm{id} \otimes B_+^a)(\Delta\tau_1 \cdots \Delta\tau_m),$$

where $B_+^a(\tau_1 \cdots \tau_m) := [\tau_1 \cdots \tau_m]_a$ for a forest $\tau_1 \cdots \tau_m$. The definition is extended to a linear multiplicative map on $\mathcal{H}$. We will occasionally use Sweedler's notation

$$\Delta\tau = \sum_{(\tau)} \tau^{(1)} \otimes \tau^{(2)}$$

for the coproduct. We omit the definition of the antipode S here, and instead refer the reader to (Manchon, 2004; Hoffman, 2003). We denote the dual of the Connes-Kreimer Hopf algebra by $\mathcal{H}^*$. For $\varphi_1, \varphi_2 \in \mathcal{H}^*$, the convolution product is defined by

$$\varphi_1 * \varphi_2 = \mu_{\mathbb{R}} \circ (\varphi_1 \otimes \varphi_2) \circ \Delta,$$

with $\mu_{\mathbb{R}} : \mathbb{R} \otimes \mathbb{R} \to \mathbb{R}$ denoting multiplication in $\mathbb{R}$.

A.2 B-SERIES EXPANSIONS OF ODES

For any tree $\tau \in \mathcal{T}$, the so-called elementary differential $F(\tau)(y)$ (Butcher, 2016) is defined recursively by

$$F(\emptyset)(y) = y, \quad F(\bullet_i)(y) = f_i(y),$$
$$F([\tau_1, \tau_2, \dots, \tau_m]_i)(y) = f_i^{(m)}(y)(F(\tau_1)(y), F(\tau_2)(y), \dots, F(\tau_m)(y)).$$

Given a map $\varphi : \mathcal{T} \to \mathbb{R}$, the associated B-series is defined

$$B_h(\varphi, y_0) := \sum_{\tau \in \mathcal{T}} \frac{h^{|\tau|}}{\sigma(\tau)} \varphi(\tau) F(\tau)(y_0).$$

A key property of B-series is their closure under composition. One can show that for two B-series Hairer & Wanner (1974); Butcher et al. (2024),

$$B_h(\varphi_2, B_h(\varphi_1, y_0)) = B_h(\varphi_1 * \varphi_2, y_0),$$

where $\varphi_1 * \varphi_2$ is the convolution product defined above. It can be shown that the exact solution to the ODE in equation 2 admits a B-series representation $y(h) = B_h(e, y_0)$, where $e(\tau) = 1/\tau!$. Similarly, the solution given by a Runge–Kutta scheme with coefficients $\{a_{ij}\}_{1 \le i,j \le s}$, $\{b_i\}_{1 \le i \le s}$ admits the B-series representation $B_h(\varphi, y_0)$, where (Butcher, 2016, Lemma 312B)

$$\varphi(\tau) := \sum_{i_1, \dots, i_n} b_{i_1} \prod_{(k,\ell) \in E} a_{i_k, i_\ell}$$

for a tree $\tau = (V, E, r)$ with $|\tau| = n$.

We refer the reader to (Hairer et al., 2006; McLachlan et al., 2015; Butcher, 2021) for a detailed account of B-series and the Butcher group.

A.3 Branched Rough Paths

Let $\mathcal{H}$ be the Connes-Kreimer Hopf algebra of non-planar labelled rooted trees defined above.

Definition A.1. Let $\alpha \in (0, 1]$. An α-Hölder branched rough path is a map $\mathbf{X} : [0, T]^2 \to \mathcal{H}^*$ such that

1. for all $s, t \in [0, T]$ and $\tau_1, \tau_2 \in \mathcal{H}$,
$$\langle \mathbf{X}_{s,t}, \tau_1 \rangle \langle \mathbf{X}_{s,t}, \tau_2 \rangle = \langle \mathbf{X}_{s,t}, \tau_1 \tau_2 \rangle,$$
2. for all $\tau \in \mathcal{H}$,
$$\langle \mathbf{X}_{s,t}, \tau \rangle = \sum_{(\tau)} \langle \mathbf{X}_{s,u}, \tau^{(1)} \rangle \langle \mathbf{X}_{u,t}, \tau^{(2)} \rangle,$$
where $\Delta \tau = \sum_{(\tau)} \tau^{(1)} \otimes \tau^{(2)}$.
3. for all $\tau \in \mathcal{H}$,
$$\sup_{s \neq t} \frac{|\langle \mathbf{X}_{s,t}, \tau \rangle|}{|t - s|^{\alpha |\tau|}} < \infty.$$

Remark. As remarked in (Hairer & Kelly, 2015; Gubinelli, 2010), the components $\langle \mathbf{X}_{s,t}, \tau \rangle$ with $|\tau| > N$ are determined by those with $|\tau| \le N$, where N is the largest integer such that $N\alpha \le 1$.

The space of α-Hölder branched rough paths is a complete metric space under the metric

$$\varrho_\alpha(\mathbf{X}, \mathbf{Y}) := \sum_{\tau \in \mathcal{T}_N} \sup_{s \neq t} \frac{|\langle \mathbf{X}_{s,t} - \mathbf{Y}_{s,t}, \tau \rangle|}{|t - s|^{\alpha |\tau|}},$$

where $N = \lfloor 1/\alpha \rfloor$.

A.4 B-Series Expansions of RDEs

We recount the results of (Redmann & Riedel, 2020) regarding series expansions of the solutions to RDEs of the form equation 4. Recall the definition of the elementary differential $F(\tau)(y)$ from Appendix A.2.

Theorem A.1 ((Redmann & Riedel, 2020, Theorem 2.10)). Let $\mathbf{X}$ be an α-branched rough path and $h > 0$. Then equation 4 admits the series expansion

$$y(t_0 + h) = y_0 + \sum_{\tau \in \mathcal{T}_p} \frac{1}{\sigma(\tau)} F(\tau)(y_0) \, \langle \mathbf{X}_{t_0, t_0 + h}, \tau \rangle + \mathcal{O}(h^{(p+1)\alpha})$$

for all $p \ge \lfloor 1/\alpha \rfloor$.

B General Runge–Kutta methods for RDEs

Following Burrage & Burrage (1996; 1998; 2000); Redmann & Riedel (2020), consider the class of general Runge-Kutta methods defined by

$$y_{n+1} = y_n + \sum_{m=1}^{d}\sum_{i=1}^{s} z_i^{(m)} f_m(k_i), \tag{10}$$
$$k_i = y_n + \sum_{m=1}^{d}\sum_{j=1}^{s} Z_{ij}^{(m)} f_m(k_j),$$

where $Z^{(1)}, \dots, Z^{(d)} \in \mathbb{R}^{s\times s}$ and $z^{(1)}, \dots, z^{(d)} \in \mathbb{R}^s$. We briefly recount the local and global error rates for such schemes, as formulated in Redmann & Riedel (2020). The results are based on the adaptation of B-series to RDEs presented above.

Definition B.1. Given $h > 0$, define the maps a, φ recursively over non-planar labelled rooted trees τ by setting $\varphi(\emptyset)(h) := (1, \dots, 1)^T \in \mathbb{R}^s$, where $\emptyset$ denotes the empty tree, and for a tree $\tau = [\tau_1 \cdots \tau_n]_i$ formed by joining $\tau_1, \dots, \tau_n$ by a new root labelled i,

$$\varphi(\tau)(h) := \prod_{j=1}^{n} (Z^{(i)} \varphi(\tau_j)(h)),$$
$$a(\tau)(h) := \left\langle z^{(i)}, \prod_{j=1}^{n} \varphi(\tau_j)(h) \right\rangle.$$

Theorem B.1 ((Redmann & Riedel, 2020)). The general Runge-Kutta method given by equation 10 has a local error of order $(p+1)\alpha$ if and only if

$$\langle \mathbf{X}_{t_0, t_0+h}, \tau \rangle = a(\tau)(h)$$

for all non-planar labelled rooted trees τ with p or fewer nodes, i.e. $|\tau| \le p$.

Proposition B.1 ((Redmann & Riedel, 2020, Proposition 4.1)). Let $y(t, y_0)$ denote the solution to equation 4 at time t starting at y_0. Suppose the Runge–Kutta method equation 10 has a local error of order $(p+1)\alpha$, and there exists a constant $C_1 > 0$ such that

$$|y(h, y_0) - y(h, \widetilde{y}_0)| \le C_1 |y_0 - \widetilde{y}_0|,$$

for h sufficiently small. Then there exists $C > 0$ such that

$$\max_{n=0,\dots,N} |y(t_n) - y_n| \le C h^{(p+1)\alpha - 1}.$$

B.1 Convergence of simplified Runge–Kutta methods

Theorem B.2 ((Redmann & Riedel, 2020, Theorem 3.3)). Let Φ be an ODE Runge–Kutta scheme. The Runge-Kutta method $\mathcal{R}(\Phi)$ approximating y^h has a local error of order $(p+1)\alpha$, i.e.

$$y^h(t_0 + h) - y_1^h = \mathcal{O}(h^{(p+1)\alpha}),$$

if and only if the ODE Runge–Kutta method Φ is of order p.

Theorem B.3 ((Redmann & Riedel, 2020, Theorem 4.2)). Let Φ be an ODE Runge–Kutta method of order p. Suppose that f is Lip_b^γ for some $\gamma > 1/\alpha$. Then $\mathcal{R}(\Phi)$ has a global error rate of $\eta = \min\{r_0, (p+1)\alpha - 1\}$, where r_0 is the convergence rate of the Wong-Zakai approximation. That is,

$$\max_{n=0,\dots N} |y(t_n) - y_n^h| = \mathcal{O}(h^\eta).$$

C Convergence of $\mathrm{EES}_{\mathcal{R}}$ schemes

We verify the global error rates given in Theorem 3.3 experimentally by reproducing the example given in (Redmann & Riedel, 2020; Deya et al., 2012) for $\mathrm{EES}_{\mathcal{R}}(2, 5; 1/10)$ and $\mathrm{EES}_{\mathcal{R}}(2, 5; (5 - 3\sqrt{2})/14)$. We take the RDE

$$dy_t = \cos(y_t) d\mathbf{X}_t^{(1)} + \sin(y_t) d\mathbf{X}_t^{(2)}, \quad y_0 = 1$$

for $t \in [0, 1]$, where $\mathbf{X}$ is the geometric lift pf a 2-dimensional fractional Brownian motion (fBm) with hurst index H. We compute the average of the maximal discretization error over $M = 10$ realisations of the RDE,

$$\mathcal{E}(h) := \frac{1}{M} \sum_{i=1}^{M} \max_{n=0,\dots,N} |y_i(t_n) - y_{i,n}|,$$

where $y_i(t)$ denotes the solution to the i^{th} realisation of the RDE and $y_{i,n}$ denotes the discretisation of the i^{th} solution using an $\text{EES}_{\mathcal{R}}$ scheme. Additionally, we evaluate the average error when recovering the initial condition,

$$\overleftarrow{\mathcal{E}}(h) := \frac{1}{M} \sum_{i=1}^{M} |y_0 - \overleftarrow{y}_{i,n}|.$$

From (Friz & Riedel, 2014), the rate r_0 can be chosen arbitrarily close to $2H - 1/2$ for a fractional Brownian motion with Hurst parameter H. It follows that we expect $\eta_1 = 2H - 1/2$ in Theorem 3.3 for both $\text{EES}(2, 5)$ and $\text{EES}(2, 7)$, and $\eta_2 = 6H - 1$ for $\text{EES}(2, 5)$ and $\eta_2 = 8H - 1$ for $\text{EES}(2, 7)$. These rates are shown in Figures 6, 7 and 8 for $H = 0.4, 0.5$ and 0.6 respectively.

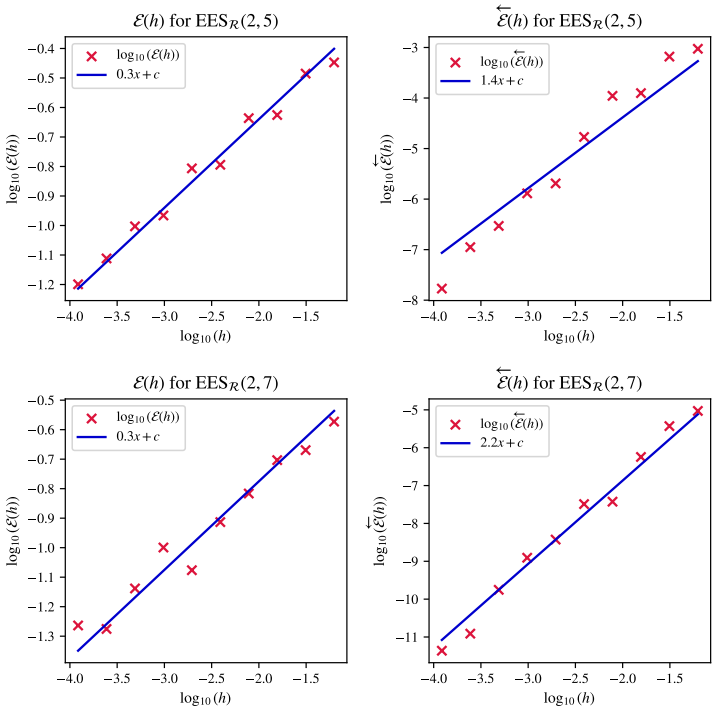

Figure 6: *Convergence rates for* $H = 0.4$

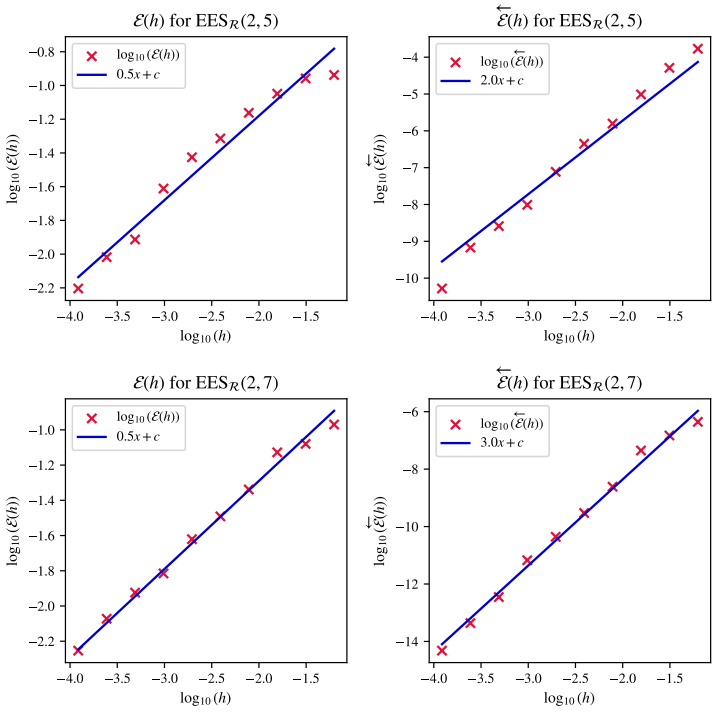

Figure 7: *Convergence rates for* $H = 0.5$

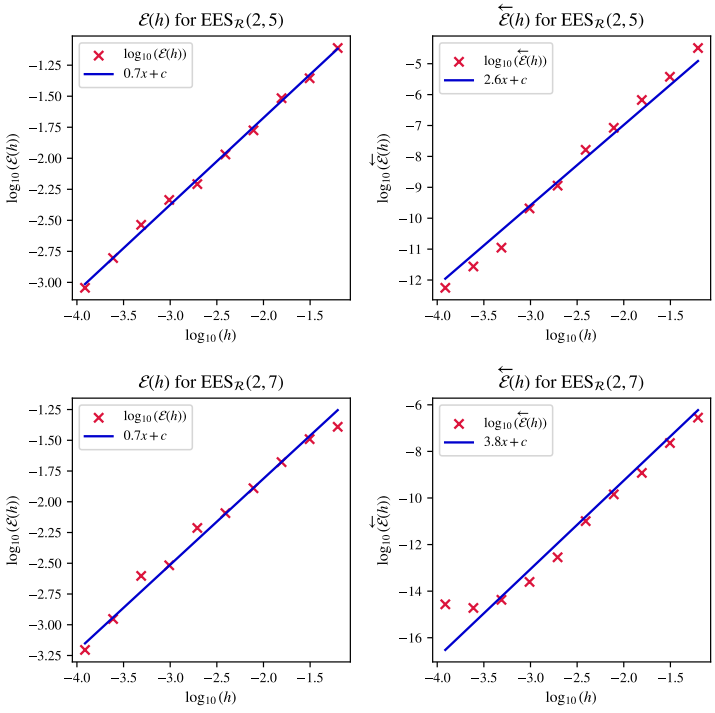

Figure 8: *Convergence rates for* $H = 0.6$