arXiv:2301.01315v1 [stat.ML] 3 Jan 2023

用于条件时间序列生成的神经随机微分方程与签名 Wasserstein-1 度量

Pere Díaz Lozano
数学系 巴塞罗那自治大学
pere.diaz@uab.cat

Toni Lozano Bagén
数学系 巴塞罗那自治大学
tonilb@mat.uab.cat

Josep Vives
巴塞罗那大学数学与计算机科学系
josep.vives@ub.edu

2023年1月5日

摘要

（条件）生成对抗网络（GAN）近年来取得了巨大成功，这得益于其能够逼近极高维空间上的（条件）分布。然而，这类网络训练过程极不稳定且计算成本高昂，尤其是在时间序列场景中。近期，有研究提出使用粗糙路径理论中的一个关键对象——路径签名，将（条件）GAN框架中的极小极大形式转化为经典的最小化问题。然而，该方法的内存开销极高，有时甚至难以承受。为解决这一问题，我们提出使用条件神经随机微分方程，其内存开销不随网络深度变化，因此比传统深度学习架构更加节省内存。我们通过实证测试发现，与其他经典方法相比，该模型在内存开销和计算时间方面效率更高，而且通常也能取得更好的性能。

关键词： 条件生成建模、神经网络、期望签名、粗糙路径理论、Wasserstein 生成对抗网络、神经随机微分方程

1 引言

进行时间序列预测的最简单方法，是只关注确定性响应，通常将其视为各种可能结果的均值。因此，模型无法报告任何内在的不确定性，这对大多数现实应用而言是一个重大缺陷。

近期的研究尝试通过将注意力转向生成对抗网络（GAN）（Arjovsky 等人，2017；Goodfellow 等人，2014）来解决这一问题。其基本思想是训练两个神经网络，使其相互对抗：

- **生成器：** 通过将随机噪声输入参数化模型，从某个分布中生成样本。
- **对抗网络：** 其任务是近似真实数据分布与生成器产生的分布之间的损失函数。

训练算法包括先对对抗网络进行若干步训练；当我们关注的损失函数达到足够好的近似后，再对生成器的参数执行一步更新。

原始的生成对抗网络 formulation 被设定为使对抗网络近似詹森-香农散度（Goodfellow 等人，2014）。然而，在我们的情形中，我们将重点关注 Wasserstein 生成对抗网络 formulation，其中对抗网络（称为 判别器）旨在近似 Wasserstein-1 距离（Arjovsky 等人，2017）。

尽管生成对抗网络在极高维空间中逼近概率测度方面取得了巨大成功，尤其是在计算机视觉领域，但其训练过程非常不稳定。此外，在生成器执行一步更新之前，需要先训练一个对抗网络，这意味着训练时间会显著增加。

1.1 签名 Wasserstein-1 度量

一种更高效的方法是以闭式形式近似损失函数，而无需使用迭代方法。沿着这一思路，Ni 等，2021 提出使用签名变换。这是粗糙路径理论中的一个基本对象，能够捕捉给定路径的许多重要解析性质和几何性质[1]。借助它，我们可以推导出一个表达式，以闭式非参数方式均匀且精确地近似路径空间上两个分布之间的 Wasserstein-1 距离。这样一来，（Wasserstein）生成对抗网络方法固有的所有缺点都得以克服，同时仍保留其强大的理论保证。这个新框架称为*Signature-Wasserstein* 生成对抗网络（SigWGAN）。

[1]尽管在实践中我们通常得到的是一组离散数据流，但我们始终可以通过执行某种插值方案，将其嵌入连续路径的集合中。因此，我们实际上不会严格区分这两个概念。

在 Ni 等，2020 中，作者提出了条件签名-瓦瑟斯坦生成对抗网络（SigCW-GAN），这是对 SigWGAN 的一种改进，用于学习条件分布；在非确定性时间序列预测中，我们关注的正是这类分布。然而，由于该算法需要针对每个数据样本执行一次蒙特卡洛过程，内存开销会大幅增加，有时甚至变得难以承受。为解决这一问题，他们假设观测到的时间序列的分布具有自回归结构，即下一个观测值仅取决于前一个 $q > 0$，且 q 相对较小。

1.2 贡献

为克服内存成本的增加，我们提出使用神经随机微分方程 （神经随机微分方程）（参见 Kidger、Foster、Li、Oberhauser 等，2021，Li 等，2020）。这类方程本质上是由经典随机微分方程构成的生成模型，其向量场由神经网络参数化。与传统深度学习架构相比，其主要优势在于，无需存储前向传播过程中产生的中间量，即可通过神经随机微分方程进行反向传播，因此比传统深度学习模型更加节省内存。

通过对我们希望作为条件依据的路径进行编码，并使神经随机微分方程的初始条件取决于该编码，我们引入了所谓的条件神经随机微分方程 （CNSDEs），它们能够在路径空间中生成条件分布。

综合运用上述方法，我们能够构建一个模型和训练算法，用于近似路径空间上的条件分布，同时兼具坚实的数学基础和更高的实践效率。此外，我们还将其与其他更传统的基线方法进行比较，结果表明，根据多个度量指标，在大多数情况下，该方法的表现都优于这些基线方法。

2 背景

2.1 签名 Wasserstein-1 度量

令 $sig(x)$、 $sig^N(x)$ 分别表示连续路径 x 的签名和 N 阶截断签名，二者均经过基点和时间增强（参见 Morrill 等，2020）。

设 μ、 ν 是路径空间中的两个分布。Kantorovich-Rubinstein 对偶指出，Wasserstein-1 距离可以计算为

$$W_1(\mu,\nu) = \sup_{\|F\|_L \leq 1} \left\{ \mathbb{E}_{x\sim\mu}[F(x)] - \mathbb{E}_{x\sim\nu}[F(x)] \right\}$$

其中，上确界取遍所有 $1-$Lipschitz 函数。

在 Ni 等，2021、Ni 等，2020 中，作者利用签名变换及其普适非线性，推导出一个能够一致逼近 Wasserstein-1 距离的闭式表达式，

$$sig^N W_1(\mu,\nu) = \left\| \mathbb{E}_{x\sim\mu}[sig^N(x)] - \mathbb{E}_{x\sim\nu}[sig^N(x)] \right\|_2 \tag{1}$$

其中$\|\cdot\|_2$ 表示 L^2 范数。该近似称为 N阶截断签名- Wasserstein-1 度量。

补充材料 A 节和 B 节对公式 (1) 进行了深入推导 .

2.2 条件签名-瓦瑟斯坦生成对抗网络算法

在无条件情形下，每当我们希望根据一组给定数据计算公式 (1) 时，只需通过执行蒙特卡洛模拟，用经验平均值分别近似两个期望签名。

然而，如果我们感兴趣的是近似条件分布，那么由于大多数时候我们手中只有一个样本，因此无法对数据给出的条件分布的期望签名执行蒙特卡洛模拟来进行近似。

利用签名的诸多性质，可以证明，条件期望截断签名能够通过将一个线性映射作用于条件路径 x 的截断签名来任意精确地近似，

$$\hat{\mathbb{E}}_{y\sim Y|x}[sig^M(y)] = \hat{\ell}(sig^N(x)) \tag{2}$$

而该映射可以通过标准线性回归技术从数据中估计。式 (2) 的详细推导见 C 节。

由此得到的训练算法称为条件签名 Wasserstein 生成对抗网络（SigCWGAN）（见 Ni 等，2020，算法 2）。

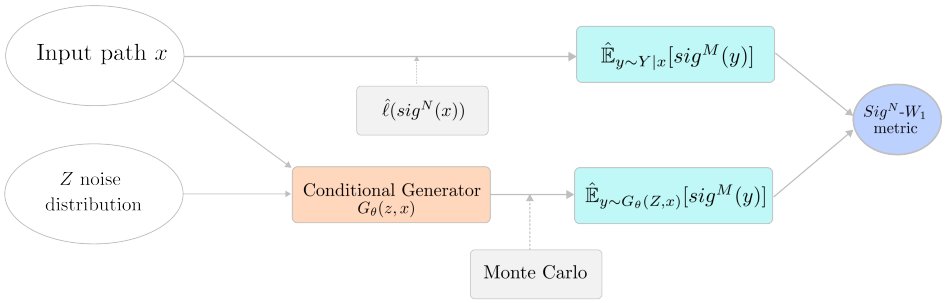

图 1：SigCWGAN算法的流程图。

2.2.1 SigCWGAN算法的缺点

SigCWGAN算法的主要缺点之一是，对于小批量中的每个样本，都需要通过蒙特卡洛过程估计条件生成器的期望签名，这显著增加了内存开销。这与一般的条件生成对抗网络设定形成对比，在后一种设定中，生成器和判别器只需将相应的样本作为输入 x （Mirza & Osindero，2014 年）。因此，条件 Wasserstein 生成对抗网络（ CWGAN ）算法的内存开销与无条件 WGAN方法相同。

这意味着，与传统的 CWGAN 方法相比，SigCWGAN算法会消耗更多内存，因此有时我们需要减小批量大小或降低模型的复杂度。正如我们将在后文看到的，使用神经随机微分方程作为生成器可以解决这一问题，因为与传统的神经网络架构相比，神经随机微分方程的训练内存效率要高得多。

2.3 神经随机微分方程

神经随机微分方程 （神经随机微分方程）是由随机微分方程和神经网络这两种数学建模中最重要的范式相结合而形成的模型，进而产生在路径空间中生成分布的生成模型。

神经随机微分方程的基本结构为

$$
\begin{aligned}
Z_0 &\sim \xi_\theta(V) \\
dZ_t &= f_\theta(t, Z_t)dt + g_\theta(t, Z_t) \circ dW_t \\
Y_t &= \alpha_\theta Z_t + \beta_\theta
\end{aligned}
$$

其中， $V \sim \mathcal{N}(\mathbf{0}, I_{d_v})$服从 d_v–维标准正态分布， ξ_θ、 f_θ、 g_θ为神经网络， $\alpha_\theta, \beta_\theta$ 为可学习权重矩阵。 W_t 表示维纳过程，而$\circ$ 表示积分采用 Stratonovich 意义。

Z 表示模型的隐藏状态。如果它是输出，那么得到的随机过程就会满足马尔可夫性质，但一般而言这并非必然。因此，最后需要采用读出线性。

正如我们之前所说，随机微分方程的解是路径空间上的一个分布。然而，和常微分方程一样，几乎总是无法以解析形式求出它。不过，我们可以从中采样。这正是数值随机微分方程求解器所做的：返回一条在一组离散时间点上求得的采样路径。

因此，神经随机微分方程符合生成对抗网络设定，因为无法计算其概率密度，而且它能够将随机噪声（以初始条件和维纳过程的形式）输入参数化模型来生成样本。

2.3.1 神经随机微分方程的反向传播

如果我们打算用某些数据拟合神经随机微分方程，就需要一种算法来计算相对于其参数的梯度。下面简要概述两种实现方法：

先离散后优化： 最直接的方法是执行常规的反向传播，对微分方程求解器的内部运算求导，因此需要在前向传播时将这些运算存储起来。在文献中，这通常称为先离散后优化，因为我们直接对实际使用的离散化进行优化。

如果用 H 表示记录一个求解器步数的运算所需的内存开销，用N 表示步数，那么先离散后优化方法大约需要$\mathcal{O}(HN)$的内存。请注意，这与传统深度学习模型（如循环神经网络）的内存开销相同。

可逆求解器： 另一种方法是使用可逆求解器，从而大幅降低内存开销。考虑一个微分方程求解器，它在前向传播中迭代地根据上一步计算出下一步，

$$(z_{t_i}, \alpha_{t_i}) \mapsto (z_{t_{i+1}}, \alpha_{t_{i+1}}), \tag{3}$$

其中$\{z_{t_i}\}_{i=1}^n$ 是某个微分方程解的数值近似，而 α_{t_i} 表示求解器在步数 i 需要存储的中间量，以便计算下一步 $i+1$的解。

那么，如果求解器能够通过闭式表达式从下一步计算出上一步，就称其为代数可逆的，

$$(z_{t_{i+1}}, \alpha_{t_{i+1}}) \mapsto (z_{t_i}, \alpha_{t_i}), \tag{4}$$

。请注意，在这种情况下，中间值（z_{t_i}， α_{t_i}）无需存储在内存中，并且可以在反向传播时重新计算。在这种情况下，内存成本仅为$\mathcal{O}(H)$。

完整算法以及目前唯一已知的此类随机微分方程求解器（称为可逆 Heun 方法）见 Kidger、Foster、Li 和 Lyons（2021）。

3 条件 神经随机微分方程

在我们的研究中，我们感兴趣的是生成以某条输入路径 x 为条件的分布。我们提出，通过让随机微分方程的初始条件取决于我们希望作为条件的路径，采用编码器-解码器结构来构造条件神经随机微分方程。

定义1（条件神经随机微分方程）。 设

$$\begin{aligned}
\xi_\theta &: \mathbb{R}^{d_h} \to \mathbb{R}^{d_z} \\
f_\theta &: [0,T] \times \mathbb{R}^{d_z} \to \mathbb{R}^{d_z} \\
g_\theta &: [0,T] \times \mathbb{R}^{d_z} \to \mathbb{R}^{d_z \times d_w}
\end{aligned}$$

为前馈神经网络。设 $\phi : T S([0,\tau];\mathbb{R}^{d_x}) \to \mathbb{R}^{d_h}$ 是一个连续映射（可包含或不包含可学习参数），其中 $T S([0,\tau];\mathbb{R}^{d_x})$ 表示时间戳位于 $[0,\tau]$ 且观测为 d_x-维的时间序列空间。于是，我们将条件神经随机微分方程（*CNSDE*）定义为

$$
\begin{aligned}
h &= \phi(x) \\
Z_0 &= \xi_\theta(h) \\
dZ_t &= f_\theta(t, Z_t)dt + g_\theta(t, Z_t) \circ dW_t \\
Y_t &= \alpha_\theta Z_t + \beta_\theta
\end{aligned}
$$

其中 $\alpha_\theta \in \mathbb{R}^{d_y \times d_z}$ 和 $\beta_\theta \in \mathbb{R}^{d_y}$ 是可学习权重矩阵。

以下是 CNSDE 的概览。输入路径 x （红色）被送入模型，模型据此确定随机微分方程 z_0的初始条件。随后，从维纳过程（蓝色）中采样得到一条轨迹 w 。所有这些内容都会被送入微分方程求解器，从而得到解 z （绿色）。之后，我们应用最终的读出线性层，生成模型的输出 y （紫色）。总而言之， y 是路径空间中某个分布的样本，该分布以给定样本 x为条件。每当我们改变 x时，这个分布也会随之改变。

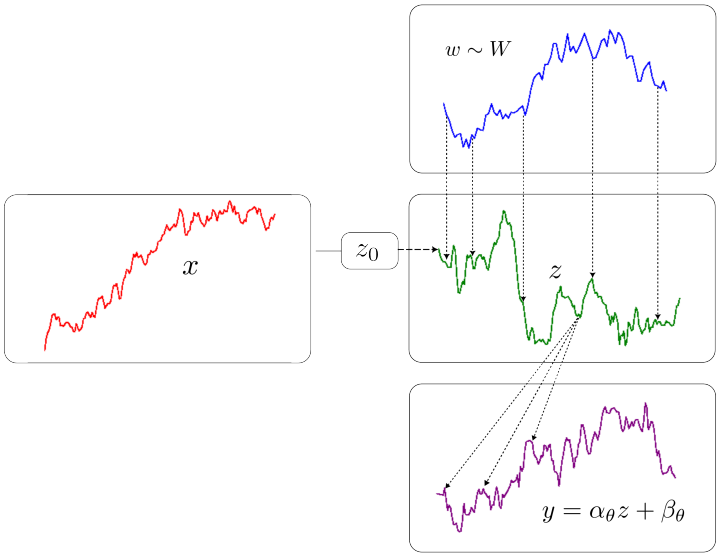

图 2：条件神经随机微分方程概览。

请注意，随机微分方程的初始条件完全由输入路径 x决定（参数固定后）。另一种方法是在初始条件 Z_0中加入随机分量，这通常能增强多样性并改善学习效果。只需将初始条件设为

$$
h = \phi(x) \tag{5}
$$

$$
Z_0 \sim [\xi_\theta^1(h), \xi_\theta^2(V)] \tag{6}
$$

其中 $\xi_\theta^1 \colon \mathbb{R}^{d_h} \to \mathbb{R}^{d_z - k}$和 $\xi_\theta^2 \colon \mathbb{R}^{d_v} \to \mathbb{R}^k$为神经网络， $V \sim \mathcal{N}(0, I_{d_v})$。相对于 d_z， k 的规模越大，我们对多样性的强调就越强。

4 实证分析

本节旨在将 SigCWGAN算法和 CNSDE 的性能与一些更传统的方法进行比较，并以这些方法作为基准。所有代码均可在 GitHub 仓库中获取：https://github.com/pere98diaz/Neural-SDEs-for-Conditional-Time-Series-Generation-and-the-Signature-Wasserstein-1-metric。

我们将比较三种模型的表现：

LSTM条件Wasserstein生成对抗网络：作为纯基线的模型基于长短期记忆（LSTM）神经网络（Hochreiter与Schmidhuber，1997）。该模型使用条件Wasserstein生成对抗网络算法进行训练，判别器也由LSTM构成。

LSTM条件Signature-Wasserstein生成对抗网络：第二个模型采用与LSTM CWGAN模型相同的条件生成器，但使用截断签名 Wasserstein-1 度量来近似Wasserstein-1 距离。我们使用SigCWGAN算法训练该模型。

神经SDE条件Signature-Wasserstein生成对抗网络：第三个模型采用条件神经随机微分方程作为生成器，并使用截断签名 Wasserstein-1 度量来近似Wasserstein-1 距离。我们同样使用SigCWGAN算法进行训练。

4.1 资源开销

本节将从两项关键资源的角度比较前文定义的三个模型：图形处理器上分配的最大内存（左侧）以及生成器执行一步所需的计算时间（右侧）。三个模型的架构经过选择，使其参数数量相近。

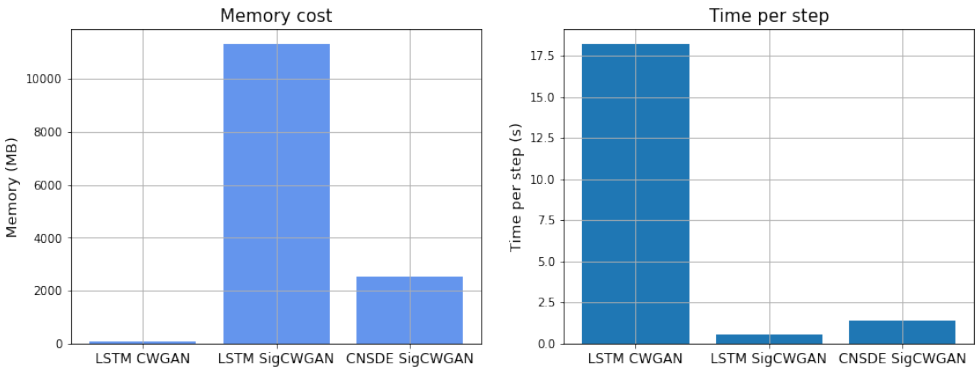

图 3：三个模型的内存消耗（MB）和每步所需时间（秒）。

可以看出，就内存开销而言，使用 SigCWGAN算法训练的长短期记忆网络远远是开销最高的。这主要是因为我们需要对小批量中的每个样本执行蒙特卡洛过程。不过请注意，当使用

CNSDE 作为条件生成器时，由于使用可逆求解器进行反向传播，内存开销会大幅降低。不出所料，使用 CWGAN算法训练的长短期记忆网络开销最低。

就每个生成器步所需的时间而言，最耗时的模型是使用 CWGAN算法训练的 LSTM。这主要是因为，在对生成器参数执行一步更新之前，我们需要先对判别器参数执行 k 步更新（本例中，我们采用标准默认值，即 $k = 10$）。使用 SigCWGAN算法训练的模型以闭式非参数方式近似 Wasserstein-1损失，因此执行一步生成器更新所需的时间要少得多。

总而言之，三个模型中最为均衡的是使用 SigCWGAN算法训练的 CNSDE。

4.2 实验

我们在四个不同的一维数据集上进行了实验，接下来将对这些数据集进行介绍。三个模型各训练了三次，并在时间外测试集上评估以下统计量，以衡量它们的性能。

分类误差： 我们训练了一个长短期记忆网络，其任务是根据一对输入/输出流，判断输出流是真实的还是生成的。所有模型使用的分类器架构相同。该分类器的表现越差，生成模型的效果就越好。

高阶Signature-Wasserstein-1度量： 与 SigCWGAN 算法类似，我们计算预测的期望签名与生成的期望签名之间的 L^2-距离。但在这种情况下，我们将签名截断在比训练期间使用的更高层级：输入路径的深度为 6，输出路径的深度为 5。

无序Wasserstein-1度量： 我们比较以下方式得到的真实一维分布之间的 Wasserstein-1 距离：a) 从输出数据流中提取数据点 y_t，不考虑它们在时间上的顺序。b)计算输出数据流中的数据点与输入数据流的最后一个值 $y_t - x_T$ 之间的差值，同样不考虑它们在时间上的顺序。c) 对每个数据流，取通过 b) 中的方法得到的最大差值。d) 对每个数据流，取通过 b) 中的方法得到的最小差值。

极值度量： 考虑通过“无序Wasserstein-1度量”c)中详述的方法得到的经验分布。令 q 表示其中一个非常高的百分位数值，例如 95%。然后，我们估计在给定输入路径 x 的条件下，生成满足 $\max_t y_t - x_T$ 等于或大于 q 的路径 y 的概率。在某些情况下，我们会改为考察百分比增幅，例如 $\max_t(y_t - x_T)/x_T$。

就像我们定义了一种方法来衡量模型检测到出现极大增量的高概率的能力一样，我们也可以用同样的方法衡量极大幅度下降的情况，具体做法是

改为考虑 $\min_t y_t - x_T$（或 $\min_t (y_t - x_T)/x_T$），并将百分位数设得很低，例如 5%。

需要指出的是，评估条件生成模型的性能，尤其是在时间序列领域，是一项极具挑战性的工作，因此上述统计量均不应被视为真实值指标。此外，我们感兴趣的度量指标大多取决于所要解决的具体问题，以及我们希望赋予该模型的用途和目标。

4.2.1 AR(5) 数据

我们使用的第一个数据集由一个 $p = 5$ 阶自回归模型模拟生成。我们考虑的问题是：根据前 60 步预测接下来的 40 步。训练时间最长设为 2 小时。不过，对于使用 SigCWGAN算法的模型，可以设置提前停止标准，即连续 1,000 步没有改进时停止。此外，训练结束后，只需保留在验证集上取得最佳损失的参数即可。

下表列出了一些训练相关的基本信息，例如分配的最大内存、训练时间以及生成器执行的步数。最后一列列出了在测试集上评估的 SigCWGAN 模型训练时所用的目标函数。为求完整，我们也列出了 LSTM CWGAN 的目标函数，尽管该模型并未被明确设置为优化这一目标函数。

	内存（MB）	时间（小时）	步数	Sig-W_1 损失
LSTM CWGAN	1945	2.000 ± 0.000	566 ± 15	5.578 ± 2.297
LSTM SigCWGAN	9931	1.712 ± 0.499	8789 ± 949	2.804 ± 0.048
NSDE SigCWGAN	3764	1.555 ± 0.386	16839 ± 1673	1.855 ± 0.095

表1：AR(5) 数据集训练过程的一些一般信息。

接下来，我们展示通过训练一个长短期记忆网络来区分真实数据和伪造数据所得到的曲线下面积（AUC）和准确率。我们可以清楚地看到，LSTM CWGAN 的表现优于其他方法。这完全不足为奇，因为在生成对抗网络框架中，生成器直接与另一个神经网络竞争，而后者的任务正是区分真实数据和生成数据。

根据各项度量指标得出的最佳模型始终以粗体标出。

	AUC	准确率
LSTM CWGAN	$\mathbf{0.747 \pm 0.144}$	$\mathbf{0.685 \pm 0.121}$
LSTM SigCWGAN	0.866 ± 0.053	0.775 ± 0.043
NSDE SigCWGAN	0.959 ± 0.021	0.873 ± 0.031

表2：分类误差，AR(5) 数据集。

	HO Sig-W_1 度量	EV+ 曲线下面积	EV_ 曲线下面积
LSTM CWGAN	5.412 ± 0.361	0.831 ± 0.098	0.834 ± 0.130
LSTM SigCWGAN	5.092 ± 0.005	0.958 ± 0.039	0.960 ± 0.035
NSDE SigCWGAN	$\mathbf{1.074 \pm 0.083}$	$\mathbf{0.988 \pm 0.004}$	$\mathbf{0.982 \pm 0.013}$

表3：AR(5) 数据集的高阶Signature-Wasserstein-1度量和两种极值度量。EV+ 和EV_ 选取的百分位数分别为99%和1%。

	W_1-a	W_1-b	W_1-c	W_1-d
LSTM CWGAN	0.092 ± 0.127	0.061 ± 0.075	0.216 ± 0.296	0.236 ± 0.301
LSTM SigCWGAN	0.086 ± 0.016	0.056 ± 0.004	$\mathbf{0.155 \pm 0.089}$	$\mathbf{0.135 \pm 0.059}$
NSDE SigCWGAN	$\mathbf{0.036 \pm 0.015}$	$\mathbf{0.017 \pm 0.005}$	0.156 ± 0.035	0.165 ± 0.061

表4：AR(5) 数据集的四种无序Wasserstein-1度量。

总之，从表1可以看出，与两种长短期记忆网络相比，神经随机微分方程在最小化 Sig-W_1 损失函数方面表现更佳。但这并未转化为表2中分类误差方面的更好表现，这意味着签名 Wasserstein-1 度量可能未能充分近似条件随机过程的编码。不过，在其余度量指标上，神经随机微分方程优于其他模型，这表明它能够编码许多最重要的几何性质。

4.2.2 西雅图天气

第二个数据集由 1948 年至 2017[2] 年西雅图每日最高气温观测值构成。我们的任务是根据最近 60 天的数据预测接下来的 30 天。

训练时间最长设为 3 小时。对于使用 SigCWGAN 的模型，我们将早停标准设为连续 1,000 步没有改进。

	内存（MB）	时间（小时）	步数	Sig-W_1 损失
LSTM CWGAN	2055	3.000 ± 0.000	965 ± 6	6.944 ± 0.360
LSTM SigCWGAN	8881	0.937 ± 0.201	7501 ± 1639	3.119 ± 1.009
NSDE SigCWGAN	4696	1.317 ± 0.460	4834 ± 1626	1.410 ± 0.067

表5：一　些关于训练过程的一般信息，西雅图天气数据集　　.

[2]该数据集可　　在 https://www.kaggle.com/datasets/rtatman/did-it-rain-in-seattle-19482017 找到

	AUC	准确率
LSTM CWGAN	$\mathbf{0.617 \pm 0.060}$	$\mathbf{0.573 \pm 0.033}$
LSTM SigCWGAN	0.715 ± 0.114	0.652 ± 0.087
NSDE SigCWGAN	0.732 ± 0.038	0.660 ± 0.031

表 6：西雅图天气数据集的分类误差。

	HO Sig-W_1	EV+ 曲线下面积	EV_ 曲线下面积
LSTM CWGAN	6.014 ± 0.026	0.514 ± 0.025	0.577 ± 0.047
LSTM SigCWGAN	5.465 ± 0.265	0.808 ± 0.099	0.888 ± 0.046
NSDE SigCWGAN	$\mathbf{1.032 \pm 0.082}$	$\mathbf{0.905 \pm 0.001}$	$\mathbf{0.940 \pm 0.003}$

表 7：签名 Wasserstein-1 度量和两种极值度量，西雅图天气数据集。EV+ 和 EV_ 的选定百分位数分别为 95% 和 5%。

	W_1-a	W_1-b	W_1-c	W_1-d
LSTM CWGAN	0.061 ± 0.007	0.155 ± 0.018	0.251 ± 0.018	0.175 ± 0.018
LSTM SigCWGAN	0.060 ± 0.011	0.035 ± 0.006	0.094 ± 0.030	0.144 ± 0.072
NSDE SigCWGAN	$\mathbf{0.047 \pm 0.002}$	$\mathbf{0.028 \pm 0.001}$	$\mathbf{0.062 \pm 0.012}$	$\mathbf{0.055 \pm 0.008}$

Table 8：四种无序Wasserstein-1度量，西雅图天气数据集 .

结论与我们针对 AR(5) 数据集得出的结论基本相似。从表5可以看出，神经随机微分方程在最小化 CSig-W损失函数方面远胜于长短期记忆网络。由于这一点并未反映在表 6 所示的分类误差表现上，这意味着签名 Wasserstein-1 度量未能很好地编码条件随机过程。不过，鉴于神经随机微分方程在其余度量指标上的表现优于其他模型，我们也可以得出结论：它成功捕捉到了许多重要的几何性质。

我们特别强调表 7 中展示的结果。从中可以看出，使用 WGAN算法训练的长短期记忆网络在检测极端值方面表现完全不佳。相比之下，使用 SigCWGAN方法训练的模型则表现出色。

4.2.3 外汇

第三个数据集是一个外汇时间序列，由欧元兑美元买入价的观测值构成。具体而言，每个时间戳表示买方愿意支付多少美元来购买一欧元的最高价格。

这些观测数据跨越2020年的第19周和第20周，且采样时间不规则。NSDE SigCWGAN 模型能够处理不规则采样数据，因为条件器和损失函数都基于签名变换。不过，其他两个模型并非如此

。为了能够比较这两个模型，我们对数据进行了聚合，计算每个30秒区间内的均值。

我们的任务是根据最近的80个观测值预测接下来的80个观测值。训练时间最长设为4小时。对于使用SigCWGAN算法的模型，我们将提前停止标准设为连续1,000步没有改进。

	内存（MB）	时间（小时）	步数	Sig-W_1 损失
LSTM CWGAN	5818	4.000 ± 0.000	718 ± 6	4.860 ± 1.947
LSTM SigCWGAN	9209	1.555 ± 0.252	10417 ± 1664	2.491 ± 0.002
NSDE SigCWGAN	8697	1.653 ± 0.236	3001 ± 433	2.049 ± 0.019

Ta表 9：关于训练过程、外汇数据集的一些一般信息 .

	AUC	准确率
LSTM CWGAN	0.962 ± 0.087	0.932 ± 0.097
LSTM SigCWGAN	0.838 ± 0.104	0.768 ± 0.118
NSDE SigCWGAN	$\mathbf{0.630 \pm 0.093}$	$\mathbf{0.587 \pm 0.060}$

表10：外汇数据集的分类误差。

	HO Sig-W_1	EV$_+$ 曲线下面积	EV$_-$ 曲线下面积
LSTM CWGAN	4.614 ± 1.346	0.500 ± 0.004	0.510 ± 0.017
LSTM SigCWGAN	3.010 ± 0.004	0.488 ± 0.011	0.499 ± 0.005
NSDE SigCWGAN	$\mathbf{2.704 \pm 0.077}$	$\mathbf{0.596 \pm 0.024}$	$\mathbf{0.544 \pm 0.013}$

表 11：签名 Wasserstein-1 度量和两种极值度量，外汇数据集。EV$_+$ 和 EV$_-$ 选定的百分位数分别为 90% 和 10%。

	W_1-a	W_1-b	W_1-c	W_1-d
LSTM CWGAN	0.103 ± 0.048	0.107 ± 0.046	0.053 ± 0.014	0.146 ± 0.052
LSTM SigCWGAN	$\mathbf{0.012 \pm 0.001}$	0.020 ± 0.003	0.029 ± 0.003	0.037 ± 0.007
NSDE SigCWGAN	0.013 ± 0.000	$\mathbf{0.019 \pm 0.002}$	$\mathbf{0.028 \pm 0.003}$	$\mathbf{0.015 \pm 0.002}$

表12：外汇数据集的四种无序Wasserstein-1度量。

请注意，与前两个实验相比，NSDE SigCWGAN 模型在分类误差方面明显优于其他模型。我们推测，这是由于输入和输出序列的长度增加所致。不过，还需要开展进一步实验，以检验这是否确实是原因。

在这个问题中，我们或许尤其想知道，给定已知路径 x 后，在固定时间段内是否会出现较大的增量或减量。我们通过极值度量对此进行测试，各模型的结果见表 11。可以看出，在这些度量指标方面，基于神经随机微分方程的模型也优于其他模型。

4.2.4 IBEX 35

最终数据集由 IBEX 35（伊比利亚指数）在 1993 年至 2022 年间的日收益率构成。IBEX 35 是马德里证券交易所（西班牙主要证券交易所）的基准股票市场指数。

我们的任务是根据最近 30 个观测值预测接下来的 15 个。训练时间最长设为 2 小时。对于使用 SigCWGAN算法的模型，我们将提前停止标准设为连续 1,000 步没有改善。

	内存（MB）	时间（h）	步数	Sig-W_1 损失
LSTM CWGAN	1786	2 ± 0.000	1283 ± 3	2.837 ± 0.297
LSTM SigCWGAN	9201	1.442 ± 0.295	1334 ± 2919	1.788 ± 0.020
NSDE SigCWGAN	5825	1.647 ± 0.423	6917 ± 1773	1.811 ± 0.016

表13：IBEX35 数据集训练过程的一些一般信息。

	AUC	准确率
LSTM CWGAN	0.844 ± 0.115	0.765 ± 0.104
LSTM SigCWGAN	0.661 ± 0.066	0.618 ± 0.052
NSDE SigCWGAN	$\mathbf{0.588 \pm 0.074}$	$\mathbf{0.562 \pm 0.058}$

表14：IBEX35 数据集的分类误差。

	HO Sig-W_1 度量	EV+ 曲线下面积	EV− 曲线下面积
LSTM CWGAN	4.910 ± 0.231	0.776 ± 0.041	0.507 ± 0.118
LSTM SigCWGAN	4.606 ± 0.043	0.867 ± 0.042	0.642 ± 0.048
NSDE SigCWGAN	$\mathbf{1.301 \pm 0.367}$	$\mathbf{0.886 \pm 0.005}$	$\mathbf{0.687 \pm 0.012}$

表15：IBEX35 数据集的签名 Wasserstein-1 度量和两种极值度量。EV+ 和 EV− 的选定百分位数分别为 95% 和 5%。

值得一提的是，就极值度量而言，模型在 COVID-19 年间的表现显著下降。例如，对于神经随机微分方程，2015–2019 年期间 EV− 曲线下面积的结果为 0.758 ± 0.033，而 2020–2022 年期间为 0.633 ± 0.004。

	W_1-a	W_1-b	W_1-c	W_1-d
LSTM CWGAN	0.025 ± 0.009	0.021 ± 0.004	0.041 ± 0.019	0.023 ± 0.013
LSTM SigCWGAN	0.014 ± 0.002	0.022 ± 0.001	$\mathbf{0.027 \pm 0.004}$	0.025 ± 0.001
NSDE SigCWGAN	$\mathbf{0.014 \pm 0.001}$	$\mathbf{0.017 \pm 0.001}$	0.029 ± 0.001	$\mathbf{0.016 \pm 0.002}$

表16：IBEX35 数据集的四种无序Wasserstein-1度量。

由此可见，在这种情况下，NSDE SigCWGAN 在所有度量指标方面都明显优于其他模型。

5 结论

本文提出在 SigCWGAN算法中使用条件神经随机微分方程作为条件生成器，以抵消小批量中每个样本所需的蒙特卡洛过程带来的内存成本大幅增加。

随后，我们通过实践检验了先用 SigCWGAN 方法替代传统 WGAN算法，再用神经随机微分方程替代传统长短期记忆网络生成器，在计算时间和内存开销方面分别带来的收益与代价。我们明确表明，采用 SigCWGAN算法训练的神经随机微分方程在资源开销方面最为均衡。最后，我们在四个不同的数据集上进行了四项实验，对它们的性能进行比较。实验结果表明，在大多数情况下，神经随机微分方程对真实数据集某些特性的捕捉效果更好。

致谢

Josep Vives 的研究部分得到了西班牙资助项目 PID2020-118339GB-100（2021—2024年）的支持。

利益冲突声明

所有作者均声明不存在利益冲突。本文的内容和撰写由作者独自负责。

参考文献

Arjovsky、M.、Chintala、S. 和 Bottou、L.（2017）。Wasserstein生成对抗网络。载于 D. Precup 和 Y. W. Teh（编），第*34*届国际机器学习会议论文集 （第 214–223 页）。PMLR。

Arribas，I. P.（2018）。使用签名收益进行衍生品定价。*arXiv* 电子预印本，arXiv:1809.0946 6.

Chevyrev，I. 和 Kormilitzin，A.（2016）。机器学习中的签名方法入门。*arXiv* 电子预印本，arXiv:1603.03788。

Fawcett, T. (2003年)。随机分析中的问题：粗糙路径与非交换调和分析之间的联系 （博士学位论文）。牛津大学。牛津大学。

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014年)。生成对抗网络。载于 Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (编)，神经信息处理系统进展（第 2672–2680 页）。Curran Associates, Inc. Hambly, B., & Lyons, T. (2010年)。有界变差路径签名的唯一性及约化路径群。数学年刊，*171.1*，109–167。Hochreiter, S., & Schmidhuber, J. (1997年)。长短期记忆。神经计算，*9*(8)，1735–1780。Kidger, P., Foster, J., Li, X., & Lyons, T. (2021)。神经随机微分方程的高效且准确的梯度。载于 M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (编)，神经信息处理系统进展（第 18747–18761 页）。Curran Associates, Inc. Kidger, P., Foster, J., Li, X., Oberhauser, H., & Lyons, T. (2021)。作为无限维生成对抗网络的神经随机微分方程。国际机器学习会议。Kidger, P., & Lyons, T. (2021)。Signatory：在中央处理器和图形处理器上计算签名与对数签名变换的可微运算 [https://github.com/patrick-kidger/signatory]。国际学习表征会议。Li, X. (2020年)。Torchesde。*GitHub* 仓库。https://github.com/google-research/torchsde Li, X., Wong, T.-K. L., Chen, R. T. Q., & Duvenaud, D. (2020年)。随机微分方程的可扩展梯度。载于 S. Chiappa & R. Calandra (编)，第二十三届人工智能与统计国际会议论文集（第 3870–3882 页）。PMLR。Lyons, T., Caruana, M., & Lévy, T. (2004年)。粗糙路径驱动的微分方程。圣弗卢尔第三十四届夏季概率学学校。Mirza, M., & Osindero, S. (2014年)。条件生成对抗网络。*arXiv* 电子预印本，arXiv:1411.1784。Morrill, J., Fermanian, A., Kidger, P., & Lyons, T. J. (2020年)。用于时间序列的广义签名方法。*arXiv* 电子预印本，arXiv:2006.00873。Ni, H., Szpruch, L., Sabate-Vidales, M., Xiao, B., Wiese, M., & Liao, S. (2021)。用于时间序列生成的签名-瓦瑟斯坦生成对抗网络。*arXiv* 电子预印本，arXiv:2006.05421。Ni, H., Szpruch, L., Wiese, M., Liao, S., & Xiao, B. (2020年)。用于时间序列生成的条件签名-瓦瑟斯坦生成对抗网络。*SSRN*。Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., . . . Chintala, S. (2019年)。PyTorch：一种命令式风格的高性能深度学习库。载于 H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, & R. Garnett (编)，神经信息处理系统进展第*32*卷（第 8024–8035 页）。Curran Associates, Inc.

Villani, C.（2009）。最优传输：旧与新。数学科学基础丛书。Springer，柏林。

补充材料

A 签名变换

为求完整，我们将对签名变换作一个基础介绍，并简要了解使其成为机器学习中一种便捷变换的诸多性质。

A.1 有界变差路径

定义 2 ($p-$变差)**.** 设 $p \geq 1$ 为一个实数，且 $\|\cdot\|$ 为 $\mathbb{R}^d$上的任意范数。令$x\colon [0,T] \to \mathbb{R}^d$ 为一条连续路径。 x 的 $p-$变差定义为

$$\|x\|_p = \left[\sup_{\mathcal{D}} \sum_{i=0}^{n-1} ||x_{t_{i+1}} - x_{t_i}||^p \right]^{1/p} \tag{7}$$

其中，上确界是在 $[0,T]$ 的分割集合上取的，该集合记为 D。有限 $p-$变差的 $d-$维连续路径空间记为 $V^p([0,T];\, \mathbb{R}^d)$。

我们将为 $\mathcal{V}^p([0,T];\, \mathbb{R}^d)$ 配备如下范数。

定义 3 ($p-$变差范数)。 路径 $x \in V^p([0,T];\ \mathbb{R}^d)$ 的 $p-$变差范数定义为

$$\|x\|_{p-var} = \|x\|_p + \sup_{t\in[0,T]} \|x_t\| \tag{8}$$

其中 $\|\cdot\|$ 是 $\mathbb{R}^d$ 上的任意范数。

为简单起见，我们只研究有限 $1-$变差的路径，这类路径称为有界变差路径。这样做的原因是，在实践中，我们总是会得到离散数据流，通过某种插值方案，可以将其转换为连续路径。最常见的方法是线性插值（由此得到的路径具有有界变差），因为这样一来，签名就很容易计算。Signatory 软件包（Kidger & Lyons，2021）正是采用这种方法，提供了在图形处理器上计算签名的可微运算。

此外，要定义一般有限 $p-$变差路径的签名，我们需要引入粗糙路径理论中的许多新概念，而这超出了本文的范围。

A.2 张量代数

定义 4 （ $\mathbb{R}^d$的张量代数）。令（$\mathbb{R}^d$）$^{\otimes k}$ 表示 $\mathbb{R}^d$ 的 k 阶张量幂。按照惯例，（$\mathbb{R}^d$）$^{\otimes 0} = \mathbb{R}$。我们将 $\mathbb{R}^d$ 的扩张张量代数定义为

$$T((\mathbb{R}^d)) = \{\boldsymbol{a} = (a_0, a_1, ...) \mid \forall k \geq 0, a_k \in (\mathbb{R}^d)^{\otimes k}\}$$

我们还将 $\mathbb{R}^d$ 的 N 阶截断张量代数定义为一个线性子空间，属于T（（$\mathbb{R}^d$））， 其定义如下

$$T^{(N)}(\mathbb{R}^d) = \{\boldsymbol{a} = (a_0, a_1, ..., a_N) \mid \forall N \geq k \geq 0, a_k \in (\mathbb{R}^d)^{\otimes k}\}$$

注意， $T^{(N)}(\mathbb{R}^d)$ 是一个维数为

$$\sum_{k=0}^{N} d^k = \begin{cases} N+1 & if\ d = 1 \\ \frac{d^{N+1}-1}{d-1} & if\ d > 1 \end{cases} \tag{9}$$

我们将在 $T((\mathbb{R}^d))$ 上赋予一个容许范数，定义如下。

定义 5. 若扩张张量代数 $T((\mathbb{R}^d))$ 赋有一个容许范数 $\|\cdot\|$，且满足以下条件，则称其具有容许范数:

1. 对于每个 $k \geq 1$，对称群 S_k 通过等距变换作用于 $(\mathbb{R}^d)^{\otimes k}$，即

$$\|\sigma v\| = \|v\|, \quad \forall v \in (\mathbb{R}^d)^{\otimes k},\ \forall \sigma \in \mathcal{S}_k \tag{10}$$

2. 张量积的范数为1，即 $\forall n, m \geq 1$,

$$\|v \otimes w\| \leq \|v\| \|w\|, \quad \forall v \in (\mathbb{R}^d)^{\otimes n},\ w \in (\mathbb{R}^d)^{\otimes m} \tag{11}$$

A.3 路径签名

签名变换提供了一种将任意连续路径编码为无限序列统计量的方法。它具有许多特性，使其成为机器学习中非常理想的变换（Chevyrev & Kormilitzin，2016）。

定义 6（路径签名）。 设 $x = (x^1, ..., x^d)$： $[0, T] \to \mathbb{R}^d$ 是一条有界变差连续路径。路径 x 的签名定义为由迭代积分构成的无限集合

$$sig(x) = \left(\int \cdots \int_{0<t_1<\cdots<t_k<T} dx_{t_1} \otimes \cdots \otimes dx_{t_k} \right)_{k \geq 0} \in T((\mathbb{R}^d))$$

其中积分是在黎曼–斯蒂尔切斯意义下进行的。按照惯例， $k = 0$ 项取为 $1 \in \mathbb{R}$。对于每个固定的 $k \geq 0$，相应的项称为签名的第 k 层。

路径签名最重要的性质之一，是除时间重参数化和平移外具有唯一性。具体而言，有如下结果。

引理1 (Hambly 和 Lyons，2010)。 设 A 是 $V^1([0, T];\ \mathbb{R}^d)$ 的一个子集，且 A 中所有路径具有相同的初始值，并且至少有一个单调坐标。那么，路径签名能够完全确定 A 中的路径。

换言之，对原始路径进行几种简单的变换（即基点增强和时间增强，见 Morrill 等，2020）后，签名便能为任意有界变差的连续路径提供完美的编码。我们将这样的空间记作 $\mathcal{A}^1([0,T];\mathbb{R}^d)$。

实际上，我们通常无法计算签名的所有项，因此只能计算其中有限个。

定义 7 (路径的截断签名**)**。设 $x=(x^1,...,x^d):[0,T]\to\mathbb{R}^d$ 是一条有界变差的连续路径。x 的 N 阶截断签名定义为由以下迭代积分构成的有限集合

$$sig^N(x)=\left(\int\limits_{0<t_1<\cdots<t_k<T}\cdots\int dx_{t_1}\otimes\cdots\otimes dx_{t_k}\right)_{N\geq k\geq 0}\in T^{(N)}(\mathbb{R}^d)$$

其中，$T^{(N)}(\mathbb{R}^d)$ 表示 $\mathbb{R}^d$ 的 N 阶截断张量代数。

事实证明，选取签名的前几层是很好的近似，因为其各层的大小会按阶乘衰减。

定理 1 (阶乘衰减，Lyons 等人，2004，命题 2.2**)**。设 $x:[0,T]\to\mathbb{R}^d$ 是一条有界变差的连续路径。令 $sig_k(x)$ 表示 x 的签名的第 k 层。那么，对任意 $k\in\mathbb{N}$，都有

$$\|sig_k(x)\|\leq\frac{\|x\|_{1-var}^k}{k!}\tag{12}$$

由此可直接得出，可以通过截断签名序列，以任意精度近似签名。

推论 2. 设 $x:\ [0,T]\to\mathbb{R}^d$ 是有界变差的连续路径。那么，对每个$\epsilon>0$，都存在一个 $N\in\mathbb{N}$ 使得

$$\left\|sig(x)-sig^N(x)\right\|\leq\epsilon\tag{13}$$

此外，若将范围限制在紧子集 $K\subset V^1([0,T];\ \mathbb{R}^d)$上，则收敛是一致的。

签名的另一个非常重要的性质是通用非线性性质，它指出签名为路径空间上的连续函数空间提供了一组基。

定理 3（通用非线性，Arribas，2018，定理 4.2）。设 $K\subset\mathcal{A}^1([0,T];\mathbb{R}^d)$是一个紧子集。那么

$$\bigcup_{N\in\mathbb{N}}\left\{x\mapsto\ell(sig^N(x))\mid\ell:T^{(N)}(\mathbb{R}^{d+1})\to\mathbb{R}^v\ is\ linear\right\}\tag{14}$$

是一致的　在从 K 到 $\mathbb{R}^v$ 的连续映射空间中稠密，记作 $C(K;\mathbb{R}^v$).

A.4 随机过程的签名

我们将看到的最后一个性质指出，具有紧支撑的随机过程的期望签名能够完全确定其分布。

定理4 (Fawcett, 2003)**.** 设 $\{X_t\}_{t\in[0,T]}$ 和 $\{Y_t\}_{t\in[0,T]}$ 是两个具有紧支撑 $K \subset A^1([0,T];\mathbb{R}^d)$ 的随机过程。那么，我们有：

1. 期望签名是定义良好的，$\mathbb{E}_{x\sim X}[sig(x)]$, $\mathbb{E}_{y\sim Y}[sig(y)] < \infty$。

2. 如果 $\mathbb{E}_{x\sim X}[sig(x)] = \mathbb{E}_{y\sim Y}[sig(y)]$,，那么 X 和 Y 在分布意义下相等。换言之，它们具有相同的分布。

B 签名 Wasserstein-1 度量

我们首先回顾定义在同一可测空间上的分布之间的 Wasserstein-1 距离的定义。

定义 8（Wasserstein-1 距离）。令 $\Pi(\mu,\nu)$ 表示可测空间 $(X \times X, F \otimes F)$ 上所有联合分布的集合，其边缘分布分别为 μ 和 ν。那么，*Wasserstein-1* 距离定义为

$$W_1(\mu,\nu) := \inf_{\gamma\in\Pi(\mu,\nu)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||_1] \tag{15}$$

在实践中，式 (15) 中的下确界极难计算。幸运的是，Kantorovich-Rubinstein 对偶性（Villani，2009）指出，Wasserstein-1 距离可以计算为

$$W_1(\mu,\nu) = \sup_{\|F\|_L\le 1} \Big\{\mathbb{E}_{x\sim\mu}[F(x)] - \mathbb{E}_{x\sim\nu}[F(x)]\Big\} \tag{16}$$

其中，上确界是在所有 $1-$Lipschitz 函数 F 上取的：$\mathcal{X} \to \mathbb{R}$。最常用的方法是用一族参数化的 $1-$Lipschitz 函数来近似 F，如 Wasserstein-GAN 方法（Arjovsky 等人，2017）所采用的方法。

现在，令 Z 为取值于空间 $\mathcal{Z}$ 上的随机变量。令 $G_\theta : \mathcal{Z} \to \mathcal{X}$ 为由 θ 参数化的映射。考虑 $\mathcal{X}$ 上由 $G_\theta(Z)$ 给出的推前条件分布，记为 $\mathbb{P}_\theta$，并令 $\mathbb{P}_r$ 表示数据的分布。那么，我们的目标是最小化 $\mathbb{P}_\theta$ 与 $\mathbb{P}_r$ 之间 Wasserstein-1 距离的一个近似值。

在瓦瑟　　斯坦生成对抗网络方法中，试图求解的优化问题是

$$\min_{\theta\in\Theta}\max_{\psi\in\Psi} \Big\{\mathbb{E}_{x\sim\mathbb{P}_r}[F_\psi(x)] - \mathbb{E}_{z\sim Z}[F_\psi(G_\theta(z))]\Big\} \tag{17}$$

其中 $\{F_\psi\}_{\psi\in\Psi}$ 是一族 $1-$Lipschitz 函数。训练算法包括对判别器 F_ψ 的参数执行 k 步训练；当 Wasserstein-1 距离的近似足够准确时，

对生成器 G_θ 的参数执行一步训练。然而，这种方法非常不稳定，而且对超参数的选择很敏感。

请记住，在我们的情形中，我们感兴趣的是近似路径空间中的分布，使得 $\mathcal{X} = \mathcal{A}^1([0,T];\mathbb{R}^d)$。令 $K \subseteq \mathcal{X}$ 表示 μ 和 ν 的支撑的并集。根据 (16) 的定义，存在一个函数序列 $F_n : K \to \mathbb{R}$，其中的函数均为 1–Lipschitz 函数，并且该序列达到上确界 $W_1(\mu,\nu)$。

若假设 $K \subset \mathcal{X}$ 是一个紧子集，根据签名的普适非线性性质，我们有 $\forall \epsilon > 0$，并且对每个 F_n，都存在一个线性泛函 $\ell_n : T((\mathbb{R}^{d+1})) \to \mathbb{R}$，使得

$$\sup_{x\in K} |F_n(x) - \ell_n(sig(x))| \leq \epsilon \tag{18}$$

这意味着，(16) 中的上确界可以由一列作用于签名的线性泛函达到，因此 (16) 等价于

$$sigW_1(\mu,\nu) = \sup_{\|\ell\|_L \leq 1} \Big\{ \mathbb{E}_{x\sim\mu}[\ell(sig(x))] - \mathbb{E}_{x\sim\nu}[\ell(sig(x))] \Big\} \tag{19}$$

$$= \sup_{\|\ell\|_L \leq 1} \ell\Big(\mathbb{E}_{x\sim\mu}[sig(x)] - \mathbb{E}_{x\sim\nu}[sig(x)] \Big) \tag{20}$$

其中期望签名是定义良好的，因为我们假设两个测度都有紧支撑。此外，利用截断签名和推论 2，我们可以用下式一致地逼近 (19)

$$sig^N W_1(\mu,\nu) = \sup_{\|\ell\|_L \leq 1} \ell\Big(\mathbb{E}_{x\sim\mu}[sig^N(x)] - \mathbb{E}_{x\sim\nu}[sig^N(x)] \Big) \tag{21}$$

此处，线性泛函 ℓ 定义在截断张量代数 $T^{(N)}(\mathbb{R}^{d+1})$ 中。由于定义域空间是有限维的，线性泛函 $\ell : T^{(N)}(\mathbb{R}^{d+1}) \to \mathbb{R}$ 的 Lipschitz 常数就是其系数作为欧几里得向量时的范数。

事实证明，当我们将其选为 L_2 范数时，优化问题 (21) 有一个闭式解（Ni 等人，2020 年，引理 A.3），其解为

$$sig^N W_1(\mu,\nu) = \Big\| \mathbb{E}_{x\sim\mu}[sig^N(x)] - \mathbb{E}_{x\sim\nu}[sig^N(x)] \Big\|_2 \tag{22}$$

这称为阶数为 N 的截断签名 Wasserstein-1 度量。在生成对抗网络设定中，我们可以通过进行蒙特卡洛模拟来近似模型和数据的期望签名，即计算签名的经验平均值。

C 条件签名-瓦瑟斯坦生成对抗网络方法

如果我们掌握了一些想要作为条件来约束该分布的信息，该怎么办？令 $X = \{X_t\}^{t\in[0,T_x]}$ 和 $Y = \{Y_t^{t\in[0,T_y]}$ 为两个取值于

$$\mathcal{X} = \mathcal{A}^1([0,T_x];\mathbb{R}^{d_x}), \qquad \mathcal{Y} = \mathcal{A}^1([0,T_y];\mathbb{R}^{d_y}) \tag{23}$$

分别。令 $p(X, Y)$ 表示它们的联合分布。那么，给定任意已知轨迹$x \in \mathcal{X}$，我们关注的是近似条件分布 $p(Y|X = x)$，简记为 $Y|x$。

实践中，我们无法通过蒙特卡洛过程近似 $Y|x$ 的期望签名，因为对于每个 x ，通常只给定一个样本。因此，我们必须采用另一种方法来估计 $\mathbb{E}_{y\sim Y|x}[sig(y)]$。

与监督学习[3], 一样，我们可以将这个问题表述为近似一个映射

$$\begin{array}{rccc} f\colon & \mathcal{X} & \to & \mathcal{P}(\mathcal{Y}) \\ & x & \mapsto & Y|x \end{array} \tag{24}$$

其中$\mathcal{P}(\mathcal{Y})$ 是所有取值于$\mathcal{Y}$的随机过程所构成的空间。如果我们将范围限定为具有紧支撑的随机过程族，那么根据定理4和引理1，存在唯一函数 $\hat{f}\colon T((\mathbb{R}^{d_x+1})) \to T((\mathbb{R}^{d_y+1}))$，使得

$$\hat{f}(sig(x)) = \mathbb{E}_{y\sim Y|x}[sig(y)] \tag{25}$$

其中我们记 $f(x) = Y|x$。如果假设 f 的定义域是一个紧子集，我们就可以利用截断签名的一致收敛性，将式（25）近似为

$$\begin{array}{rccc} \hat{f}_{N,M}\colon & K \subset T^{(N)}(\mathbb{R}^{d_x+1}) & \to & T^{(M)}(\mathbb{R}^{d_y+1}) \\ & sig^N(x) & \mapsto & E_{y\sim Y|x}[sig^M(y)] \end{array} \tag{26}$$

$\hat{f}_{N,\,M}$ 的像空间是实数域上的有限维空间，因此可以将其视为一个欧几里得空间。于是，我们可以利用签名的普适非线性，通过线性泛函 $\ell_{N,\,M}$来近似 $f_{N,\,M}$。

$$\begin{array}{rccc} \ell_{N,M}\colon & K \subset T^{(N)}(\mathbb{R}^{d_x+1}) & \to & T^{(M)}(\mathbb{R}^{d_y+1}) \\ & sig^N(x) & \mapsto & E_{y\sim Y|x}[sig^M(y)] \end{array} \tag{27}$$

总之，我们将式 (24) 中两个连续无限维空间之间的非线性建模问题，转化为了两个离散有限维空间之间的线性问题。此外，函数$\ell_{N,\,M}$ 表示一个条件期望，因此我们可以使用经典的线性回归框架来近似它。

也就是说，给定一些数据 $\{x^{(i)},\ y^{(i)}\}_{i=1}^n$，我们将假设

$$sig^M(y^{(i)}) = W \cdot sig^N(x^{(i)}) + \epsilon_i \tag{28}$$

其中

[3]与监督学习的区别在于，在这种情况下，我们关注的是条件分布，而不是它的期望。

- $sig^M(y_{(i)})$ 可视为一个 $(d_y+1)^{M-1}$ 维实向量。
- $sig^N(x_{(i)})$ 可以视为一个 $(d_x+1)^{N-1}$ 维实向量。
- W 是一个维度为 $(d_y+1)^{M-1} \times (d_x+1)^{N-1}$ 的实矩阵。
- $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$ 是独立同分布的高斯噪声，其中 I 表示单位矩阵 ix.

并使用任意标准库来计算权重矩阵 W。

在实践中，我们将按如下方式进行。设 Z 是一个取值于空间 $\mathcal{Z}$ 上的随机变量。设 $G_\theta : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$ 是一个由 θ 参数化的映射。那么，对于已知的 $x \in \mathcal{X}$，我们将考虑 $\mathcal{Y}$ 上的推前条件分布，其定义为 $G_\theta(Z, x)$。与无条件情形一样，$G_\theta(Z, x)$ 的期望签名通过蒙特卡洛过程进行估计。

为了根据给定的数据集估计 $Y|x$ 的期望签名，我们只需执行前文所述的线性回归，由此得到（27）的一个估计值，记作 $\hat{\ell}$。然后，我们可以通过以下方式得到 $\mathbb{E}_{y\sim Y|x}[sig(y)]$ 的近似值

$$\hat{\mathbb{E}}_{y\sim Y|x}[sig^M(y)] = \hat{\ell}(sig^N(x)) \tag{29}$$

D 实验细节

第4节中使用的实验细节，例如模型架构和选定的超参数，可在以下页面中找到。

D.1 资源成本分析架构

LSTM 条件 WGAN 条件生成器中的长短期记忆网络包含 5 个宽度为 32 的隐藏层，使用 5 维随机噪声，并包含 77,089 个可学习参数。判别器中的长短期记忆网络也包含 5 个宽度为 32 的隐藏层，共有 76,609 个可学习参数。

LSTM 条件 Sig-WGAN 生成器的架构与刚才描述的 LSTM 条件 WGAN 生成器相同。输入路径和输出路径的签名变换分别截断于深度 5 和 4。我们还应用了累积和变换[4]。总的来说，两个截断签名的维度分别为 363 和 120。

神经微分方程条件 Sig-WGAN 条件神经随机微分方程的超参数数量略多一些。由于很难逐一追踪所有超参数，我们会在介绍定义1中使用的相应符号时提及它们。

[4]尽管截断签名可以任意精确地近似签名变换，但这并不意味着不计算更高层级就不会丢失某些关键信息。为缓解这一问题，通常会对给定的数据流应用一些变换，参见 Morrill 等，2020。

神经随机微分方程的规模 d_z 为 92。初始随机噪声的规模 d_v 为 16，神经网络 ξ_θ^2 由一个宽度为 32 的隐藏层构成。神经网络 ξ_θ^1 是一个简单的线性函数。随机微分方程的初始条件由一个规模为 8 的随机部分 k 和一个规模为 84 的确定性部分 $d_z - k$ 构成。扩散项 g_θ 采用一般类型。维纳过程 d_w 的维度为 10。漂移项 f_θ 和扩散项 g_θ 均包含一个宽度为 32 的隐藏层。我们唯一使用的激活函数是双曲正切函数 tanh。可学习参数总数为 79,273。

对于神经随机微分方程，CSig-WGAN 算法中的签名与 LSTM 条件 Sig-WGAN 模型使用相同的阶数和变换。输入路径 x 的变换 $\phi(x)$ 被设为签名变换，其截断阶数和变换与 CSig-WGAN 算法中使用的相同。

D.2 A实验的架构和超参数 s

本节将详细介绍第 4.2 节各项实验所使用的架构和超参数，这些参数是通过非正式的网格搜索选定的。
首先列出所有数据集通用的参数。

D.2.1 通用超参数

所使用的优化器如下：对于 LSTM CWGAN，我们遵循 Arjovsky 等人（2017）的做法，使用 RMSprop。对于采用 CSig-WGAN 算法训练的模型，我们使用 Adam。学习率始终设为 10^{-3}。

在所有情况下，输入路径和输出路径的签名变换分别截断至深度 5 和 4。我们还应用了累积和变换，并对每个维度进行归一化，使其均值为零、方差为单位方差。总体而言，两个截断签名的维度分别为 363 和 120。神经随机微分方程的变换 $\phi(x)$设为签名变换，其截断阶数和变换与 CSig-WGAN 算法中使用的相同。

D.2.2 AR(p) 数据集

训练集由 14,900 对时间序列数据（$x,\ y$）组成。验证集由 2,400 个样本组成。测试集由 12,400 个样本组成。我们使用训练集中输入流的均值和标准差对数据进行归一化。

各模型的架构和其他超参数如下。
LSTM WGAN生成器中的长短期记忆网络有 5 层，隐藏层大小为 32，并使用 5 维随机噪声。判别器中的长短期记忆网络有 4 层，隐藏层大小为 32。生成器的参数数量为 77,089，而判别器的参数数量为 59,713。

LSTM Sig-WGAN 如前所述，生成器与 LSTM WGAN 模型中的生成器相同。

Neural SDE Sig-WGAN 神经随机微分方程的大小 d_z 为 48。初始随机噪声的大小 d_v 为 16，神经网络 ξ_θ^2 由一个大小为 32 的隐藏层构成。神经网络 ξ_θ^1是一个简单的线性函数。

SDE 的初始条件由一个大小为 16 的随机部分 k 和一个大小为 32 的确定性部分 $d_z - k$ 构成。扩散项 g_θ 采用对角类型，因此维纳过程 d_w 的维度与解的大小相同，为 48。漂移项 f_θ 和扩散项 g_θ 均包含一个宽度为 84 的隐藏层。我们使用的唯一激活函数是双曲正切 tanh。按照 Kidger 、 Foster 、 Li 、 Oberhauser 等，2021 的做法，我们还对向量场应用了最终的 tanh。可学习参数总数为 29,161。

对于 LSTM WGAN 和 Neural SDE Sig-WGAN 模型，批量大小设为 528。由于内存限制，LSTM Sig-WGAN 的批量大小设为 228。

D.2.3 西雅图天气数据集

训练集由 15,122 对时间序列数据组成（x, y）。验证集由 3,781 个样本组成。测试集由 6,468 个样本组成。测试集提取自与训练集和验证集不同的时间段（时间外样本）。我们使用训练集中输入流的均值和标准差对数据进行了归一化。

各模型的架构和其他超参数如下。

LSTM WGAN生成器中的长短期记忆网络有 5 层，隐藏层大小为 32，并使用 5 维随机噪声。判别器中的长短期记忆网络有 4 层，隐藏层大小为 36。生成器的参数数量为 77,089，判别器的参数数量为 75,241。

LSTM Sig-WGAN 生成器与 LSTM WGAN 模型中的生成器相同。

Neural SDE Sig-WGAN 其架构与 D.2.2 节中详细介绍的架构相同，唯一不同的是我们将随机微分方程的隐藏层大小设为 $d_z = 64$。与前一个实验一样，我们在随机微分方程的向量场上应用了最终的 tanh。参数总数为 35,113。

对于 Neural SDE Sig-WGAN 模型，批量大小设为 724。对于 LSTM WGAN，批量大小设为 528。由于内存限制，LSTM Sig-WGAN 的批量大小设为 228。

D.2.4 外汇数据集

训练集由 15,000 对时间序列数据组成（$x,\ y$）。验证集由 5,000 个样本组成。测试集由 10,000 个样本组成。测试集

取自与训练集和验证集不同的时间段（时间外样本）。按照这类数据集的通常做法，我们对数据应用了对数变换。随后，我们使用训练集中输入流的均值和标准差对数据进行了归一化。

Th各模型的架构及其他超参数如下 .

LSTM WGAN 生成器的架构与我们在西雅图天气数据集实验中详细介绍的相同。不过，在本例中，判别器的两个长短期记忆网络的架构均设置为隐藏层大小 32、层数 5。判别器的参数数量为 234,113。

LSTM Sig-WGAN 生成器与 LSTM WGAN 模型中的相同。

Neural SDE Sig-WGAN 其架构与我们在西雅图天气数据集实验中详细介绍的相同。

LSTM WGAN 和 Neural SDE Sig-WGAN 的批量大小设为 528。由于内存限制，LSTM Sig-WGAN 的批量大小设为 128。

D.2.5 IBEX35 数据集

训练集由 4,296 对时间序列数据组成（x，y）。验证集由 1,074 个样本组成。测试集由 1,944 个样本组成。测试集取自与训练集和验证集不同的时间段（时间外样本）。我们使用训练集中输入流的均值和标准差对数据进行了归一化。

各模型的架构和其他超参数如下。

LSTM WGAN生成器中的长短期记忆网络有 2 层，隐藏层大小为 64，使用 5 维随机噪声。判别器中的长短期记忆网络有 4 层，隐藏层大小为 32。生成器的参数数量为 101,953，判别器的参数数量为 59,713。

LSTM Sig-WGAN 生成器与 LSTM WGAN 模式中的生成器相同 l.

Neural SDE Sig-WGAN 神经随机微分方程的大小 d_z 为 120。初始随机噪声大小 d_v 为 16，神经网络 ξ_θ^2 由一个大小为 48 的隐藏层组成。神经网络 ξ_θ^1 是一个简单的线性函数。

随机微分方程的初始条件由大小为 56 的随机部分 k 和大小为 64 的确定性部分 $d_z - k$ 构成。扩散项 g_θ 采用对角类型，因此维纳过程 d_w 的维度与解的大小相同，为 64。漂移项 f_θ 和扩散项 g_θ 均包含一个宽度为 96 的隐藏层。我们使用的唯一激活函数是双曲正切函数 tanh。与之前的实验一样，我们对随机微分方程的向量场应用了最终的 tanh 函数。可学习参数总数为 73,489。

对于 LSTM WGAN 和 Neural SDE Sig-WGAN，批量大小设置为 1024。由于内存限制，LSTM Sig-WGAN 的批量大小设置为 528。

E 计算基础设施

所有实验均在一台搭载 Windows 11 家庭版操作系统的计算机上运行，该计算机配备 AMD Ryzen 7 5800X 八核处理器、16GB 随机存取存储器，以及一块拥有 12GB 内存的 Nvidia GeForce RTX 3060 显卡。
以下列出了所使用的主要 Python 库：

- Pytorch 1.9.0+cu111 是主要的深度学习框架（Paszke 等，2019）。
- Signatory 1.2.6，可在图形处理器上对签名进行可微计算（Kidger 和 Lyons，2021）。
- torchsde 0.2.5，提供支持图形处理器的随机微分方程（SDE）求解器，并支持高效的反向传播（Li，2020）。

我们特别指出，要使用 Signatory 软件包，所需的 Pytorch 版本不得低于 1.9.0。