Neural Spatio-Temporal Point Processes

Ricky T. Q. Chen[1], Brandon Amos[2], Maximilian Nickel[2]

[1]University of Toronto; Vector Institute. [2]Facebook AI Research.

FACEBOOK AI

Spatio-Temporal Event Modeling

Towards building a **generative model** of *discrete events* that are localized in continuous time and space. Each sample is a sequence of **variable** length (*e.g.* all events within $t_i \in [0, T]$):

$$\mathcal{H} = \{(t_1, x_1), (t_2, x_2), \dots\}$$

Applications include spatial propagation of neurons, epidemic outbreaks, ride-hailing customers, earthquakes, etc.

Events can propagate along complex routes, requiring *high-fidelity conditional spatial distributions*.

Point Processes

Characterized by a conditional intensity function $\lambda(t, x \mid \mathcal{H}_t) \triangleq$

$$\lim_{\Delta t \downarrow 0, \Delta x \downarrow 0} \frac{\mathbb{P}\left(\text{One event occurs in } [t, t+\Delta t], B(x, \Delta x) \mid \mathcal{H}_t\right)}{|B(x, \Delta x)| \Delta t}$$

where $\mathcal{H}_t$ denotes history before time t, and $B(x, \Delta x)$ denotes a ball centered at $x \in \mathbb{R}^d$ and with radius Δx.

Maximum likelihood training requires solving an integral in x,

$$\log p\left(\mathcal{H}\right) = \sum_{i=1}^{n} \log \lambda^*(t_i, x_i) - \int_0^T \int_{\mathbb{R}^d} \lambda^*(\tau, x) \, dx d\tau.$$

Continuous Normalizing Flows (CNF)

Describes a **continuum of distributions** by tracking *infinitesimal* changes. Given $\frac{dx_t}{dt} = f_\theta(t, x)$, the time-dependent distribution follows

$$\log p_t(x_t) = \log p_0(x_0) - \int_0^t \nabla \cdot f \; dt$$

The resulting probability densities p_t are tractable to compute (with an ODE solver) and **always normalized**.

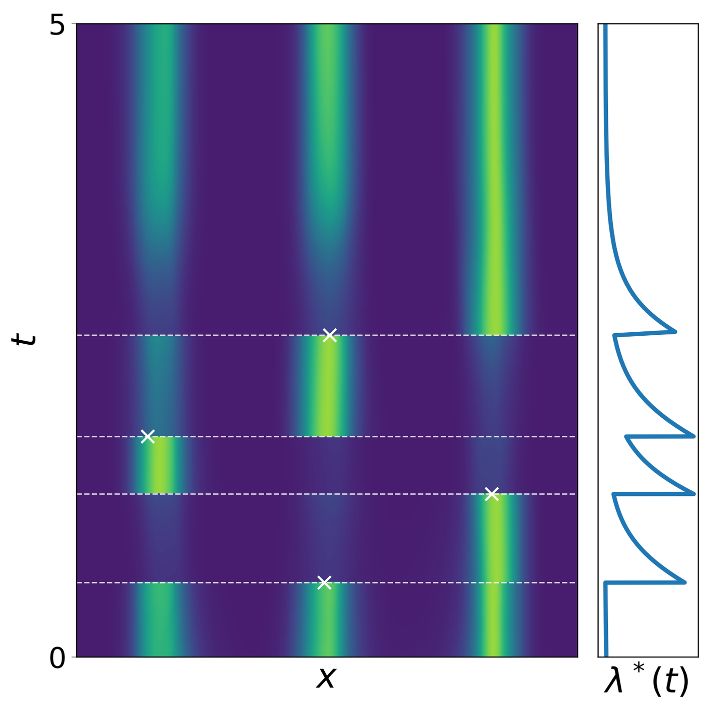
Neural Spatio-Temporal Point Process

Parameterize intensity with density of a CNF.

$$\lambda^*(t, x) = \lambda^*(t) \; p^*(x \mid t)$$

where * is shorthand for dependence on history $\mathcal{H}_t$.

(Effectively replaces $\int_{\mathbb{R}^d}$ with $n \int_0^{t_i}$)



$$\log p(\mathcal{H}) = \sum_{i=1}^{n} \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau) \, d\tau + \sum_{i=1}^{n} \log p^*(x_{t_i} | t_i)$$

How can we **condition a CNF** on $\mathcal{H}_t$ for parameterizing p^*?

Instantaneous vs. Continuous Updates

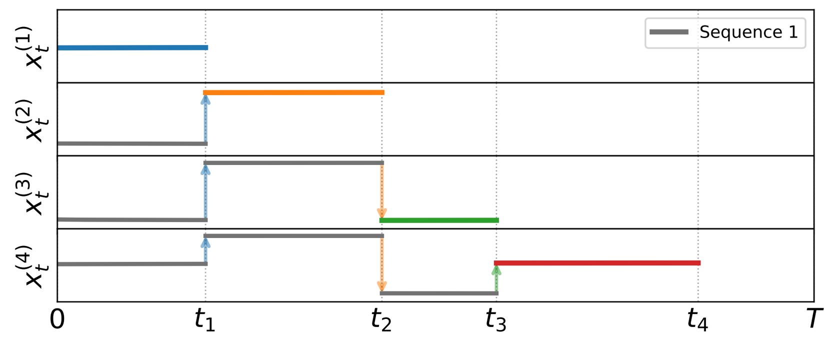
Jump CNF: Models conditioning with instantaneous *jumps* using standard normalizing flows.
Slow: requires sequentially updating for each event in $\mathcal{H}_t$.

Attentive CNF: Models conditioning with *continuous attention* on the sample paths within the drift function f.

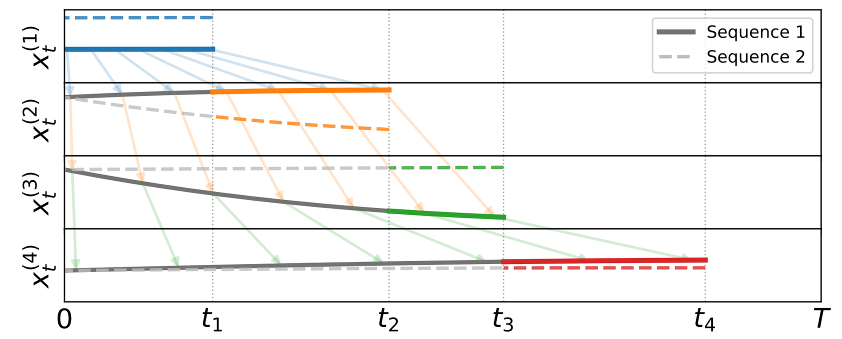
$$\frac{d\{x_i\}_{i=0}^n}{dt} = f(x_1, \dots, x_n) = \text{MaskedMultiheadAttention}$$

Fast: all ODEs can be solved in parallel.
Low-variance: Structure within MultiheadAttention allows efficient low-variance estimator of $\nabla \cdot f$.

Sample paths for a sequence of events:



Jump CNF



Attentive CNF

Ablation Experiments

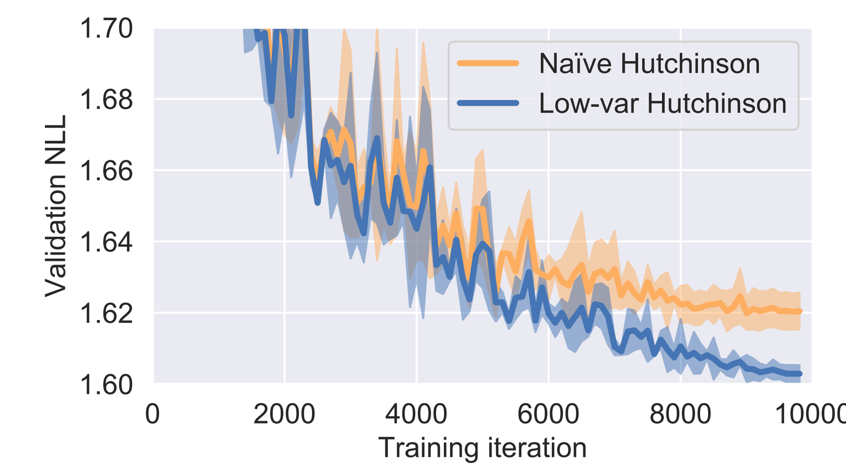


Figure: Low-variance Estimator

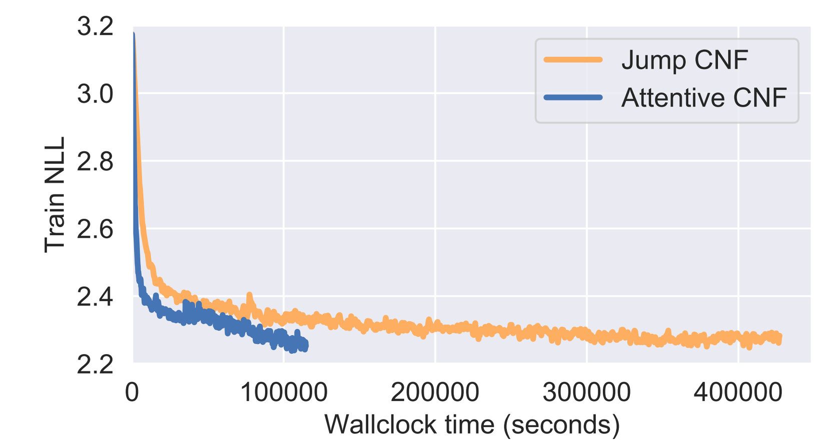


Figure: Runtime Comparison

Applications across Multiple Domains

Model	Pinwheel		Earthquakes JP		COVID-19 NJ		BOLD5000	
	Temporal	Spatial	Temporal	Spatial	Temporal	Spatial	Temporal	Spatial
Poisson Process	$-0.784_{\pm 0.001}$	–	$-0.111_{\pm 0.001}$	–	$0.878_{\pm 0.016}$	–	$0.862_{\pm 0.018}$	–
Self-correcting Process	$-2.117_{\pm 0.222}$	–	$-7.051_{\pm 0.780}$	–	$-10.053_{\pm 1.150}$	–	$-6.470_{\pm 0.827}$	–
Hawkes Process	$-0.276_{\pm 0.033}$	–	$0.114_{\pm 0.005}$	–	$2.092_{\pm 0.023}$	–	$2.860_{\pm 0.050}$	–
Neural Hawkes Process	$-0.023_{\pm 0.001}$	–	$0.198_{\pm 0.001}$	–	$2.229_{\pm 0.013}$	–	$3.080_{\pm 0.019}$	–
Conditional KDE	–	$-2.958_{\pm 0.000}$	–	$-2.259_{\pm 0.001}$	–	$-2.583_{\pm 0.000}$	–	$-3.467_{\pm 0.000}$
Time-varying CNF	–	$-2.185_{\pm 0.003}$	–	$-1.459_{\pm 0.016}$	–	$-2.002_{\pm 0.002}$	–	$-1.846_{\pm 0.019}$
Neural Jump SDE (GRU)	$-0.006_{\pm 0.042}$	$-2.077_{\pm 0.026}$	$0.186_{\pm 0.005}$	$-1.652_{\pm 0.012}$	$2.251_{\pm 0.004}$	$-2.214_{\pm 0.005}$	$5.675_{\pm 0.003}$	$0.743_{\pm 0.089}$
Jump CNF	$0.027_{\pm 0.002}$	$-1.562_{\pm 0.015}$	$0.166_{\pm 0.001}$	$-1.007_{\pm 0.050}$	$2.242_{\pm 0.002}$	$-1.904_{\pm 0.004}$	$5.536_{\pm 0.016}$	$1.246_{\pm 0.185}$
Attentive CNF	$0.034_{\pm 0.001}$	$-1.572_{\pm 0.002}$	$0.204_{\pm 0.001}$	$-1.237_{\pm 0.075}$	$2.258_{\pm 0.002}$	$-1.864_{\pm 0.001}$	$5.842_{\pm 0.005}$	$1.252_{\pm 0.026}$

Table: Log-likelihood per event on held-out test data (higher is better).

Adapts Spatial Densities On-the-fly

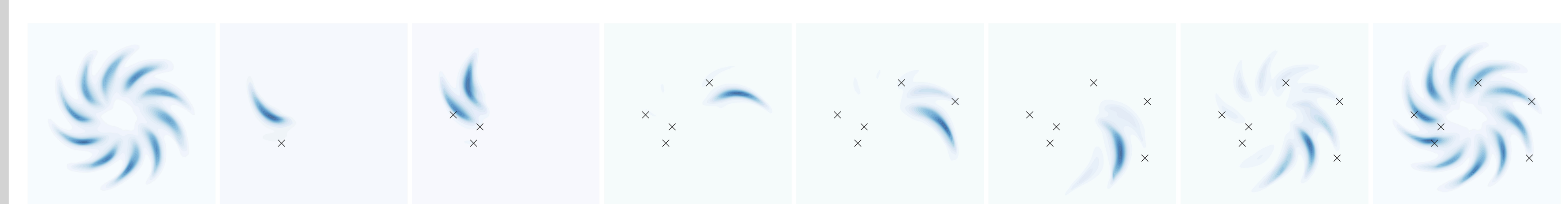


Figure: Evolution of spatial densities before returning back to marginal density.

Useful References and Links

[1] "Neural Ordinary Differential Equations" Chen et al. (2018)
[2] "FFJORD: <abbreviated>" Grathwohl & Chen et al. (2019)
[3] "Neural Jump SDEs" Jia & Benson (2019)
[4] "The Lipschitz Constant of Self-Attention" Kim et al. (2020)

Code: `https://github.com/facebookresearch/neural_stpp`