### Neural 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 \( t \in [0, T] \)):

\[ \mathcal{H} = \{(t_1, x_1), (t_2, x_2), \ldots \} \]

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 \( \lambda(t, x \mid \mathcal{H}_t) \)

\[
\lim_{\Delta t \to 0} \mathbb{P}(\text{One event occurs in } [t, t + \Delta t], B(x, \Delta x) \mid \mathcal{H}_t) = \lambda(t, x) d\Delta x
\]

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 \( \Delta x \).

Maximum likelihood training requires solving an integral in \( x \),

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

### Continuous Normalizing Flows (CNF)

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

\[
\log p_i(x_i) = \log p_0(x_0) - \int_0^t \nabla \cdot f \ dt
\]

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

### Neural Spatio-Temporal Process

Parameterize intensity with density of a CNF.

\[
\lambda^*(t, x) = \lambda^*(t) p^*(x \mid t)
\]

where \( \lambda^* \) is shorthand for dependence on history \( \mathcal{H}_t \).

(Effectively replaces \( \int_{\mathbb{R}^d} \) with \( \int_0^t \))

\[
\log p(\mathcal{H}) = \sum_{i=1}^n \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau) d\tau + \sum_{i=1}^n \log p^*(x_i \mid t_i)
\]

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 \( f \).

**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:

### Ablation Experiments

- **Figure**: Low-variance Estimator
- **Figure**: Runtime Comparison

### Applications across Multiple Domains

- **Pinwheel**
- **Earthquakes JP**
- **COVID-19 NJ**
- **BOLD5000**

<table>
<thead>
<tr>
<th>Model</th>
<th>Temporal</th>
<th>Spatial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pinwheel Process</td>
<td>-0.794</td>
<td>-0.111</td>
</tr>
<tr>
<td>Self-correcting Process</td>
<td>-2.117</td>
<td>-7.051</td>
</tr>
<tr>
<td>Hawkes Process</td>
<td>-0.276</td>
<td>-0.114</td>
</tr>
<tr>
<td>Neural Hawkes Process</td>
<td>-0.023</td>
<td>-2.292</td>
</tr>
</tbody>
</table>

| Conditional KDE | -2.996   | -2.259  |
| Time-varying CNF | -2.185   | -1.495  |
| Neural Jump SDE (GRU) | 0.006    | 2.077   |
| Jump CNF | 0.027    | -1.562  |
| Attentive CNF | 0.014    | 0.186   |

Table: Log-likelihood per event on hold-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](https://github.com/facebookresearch/neural_stpp)