Event-Time Geometry: Motivation

Modern computing architectures—especially **asynchronous** and **neuromorphic systems**—operate by processing discrete *events* that occur at various spatial locations and times. In such systems each *event* (for example, a spike from a sensor or neuron) carries a timestamp and a location tag. Reasoning about system behavior thus requires tracking these event timestamps and their causal relations. In distributed systems this is classically captured by Lamport's *happened-before* relation, a partial order of events based on causality [1] [2]. However, that framework lacks a quantitative notion of "distance" or interval between events. By analogy with spacetime in relativity, we introduce an **event-time geometry** in which each event is a point with coordinates (location, time) and a metric distance between events. This provides a unified way to compare events across different execution frames (schedules or clock domains) while preserving causal structure.

In neuromorphic vision and sensor systems, the need for such a formalism is especially clear. Event-based sensors (like Dynamic Vision Sensors) output asynchronous streams of *address-events* whenever pixels detect changes [3]. Each such event encodes its **time**, **location**, and a payload (e.g. brightness change) [4]. For example, DVS cameras output "a stream of events that encode the time, location, and sign of the brightness changes" [4]. These events propagate through spiking neural networks (SNNs), where neurons fire asynchronously. To ensure that computations are causal and consistent across different hardware or schedules, we need a rigorous event-time calculus: a geometry of events and time. We will develop this formalism in parallel with its analogy to spacetime physics, culminating in an **Event-Time Invariance Theorem** akin to the invariance of the spacetime interval.

Core Mathematical Structures

We begin by defining the basic elements of event-time geometry:

- **Locations (`L`)**: The set of all possible *locations* where events can occur. A location `l ∈ L` might represent a sensor pixel, a neuron index, a processor ID, or any spatial identifier. We assume a distance function `d: L×L → ℝ_{\ge0}` on locations (for example, Euclidean distance or graph distance), capturing communication latency or "spatial" separation between locations.

- **Timestamps (`T`)**: The time domain, typically `T = ℝ` (real time) or a subset thereof. Each event carries a timestamp `t ∈ T` indicating its occurrence time on some clock.

- **Effects/Payloads (`Φ`)**: A set of possible *effects* or data associated with an event (e.g. a sensor reading, spike amplitude, or logical value). This is the content of the event, but we will focus on the causal aspects of time and location; the effect value is carried along but does not affect geometry.

- **Events (`E`)**: An event `e ∈ E` is a tuple `(l, t, φ) ∈ L × T × Φ`. We denote `e.l` its location coordinate, `e.t` its timestamp, and `e.φ` its effect. For example, "a neuron at position `l= (x,y)` fired at time `t` with output value `φ`." In neuromorphic systems, one says the event is asynchronous: neurons or pixels generate events independently and immediately when their threshold is crossed [3].

These structures allow us to formalize causality and delays:

- **Causal Relation ($\to$)**: We say event $e1$ can **cause** event $e2$ (write $e1 \to e2$) if there exists a causal chain from $e1$ to $e2$ (e.g. a signal or influence propagates). Concretely, if $e1$ is at location $l1$ and time $t1$, and it can influence $e2$ at $l2, t2$, then necessarily $t2 \geq t1 + d(l1,l2)/c$, where c is a maximal signal speed (e.g. the speed of communication between nodes).

- **Causal Delay (δ)**: Given two causally related events $e1 \to e2$, the **causal delay** is the time separation $\delta(e1,e2) = e2.t - e1.t$. This reflects the latency from cause to effect. If $e2$ is not causally influenced by $e1$, one may say $\delta(e1,e2)$ is undefined or infinite.

- **Event-Time Interval (Δs)**: We define an interval (squared) between two events $e1, e2$ mirroring the spacetime interval. Let

  ```
  Δt = e2.t - e1.t,      Δx = d(e1.l, e2.l).
  ```

  Then the *event-time interval squared* is

  ```
  Δs^2 = c^2 (Δt)^2 - (Δx)^2,
  ```

  where c is a normalization constant (analogous to signal speed or light speed in the system). If $\Delta s^2 > 0$ and $\Delta t > 0$, the separation is **timelike** and $e1$ can potentially influence $e2$. If $\Delta s^2 = 0$ it is **lightlike** (signal speed limit exactly), and if $\Delta s^2 < 0$ it is **spacelike**: then the events cannot causally affect each other. This partitions event pairs exactly as in relativity: events within each other's "causal cones" vs. outside them.

- **Event-Time Frame**: An event-time *frame* F is a choice of coordinate system or reference for measuring event coordinates. Concretely, a frame specifies for each event e a coordinate pair $(x_F(e), t_F(e))$. Different frames might correspond to different clock synchronizations or reference clocks (for example, one frame could be a master clock, another a local core clock). For simplicity we focus on frames moving at a constant "velocity" relative to each other, just as in inertial frames in special relativity. In a one-dimensional analogy (one spatial dimension of locations), a standard Lorentz-like transformation between two frames F and F' moving at relative speed v along the x-axis is:

  ```
  ```

  t' = γ (t - v x / c^2),
  x' = γ (x - v t),

```

```

where `γ = 1/√(1 - v^2/c^2)`. Here `(x,t)` are the coordinates of an event in frame `F`, and `(x',t')` in frame `F'`.

With these definitions, we can formulate how event-time intervals behave under frame changes, leading to our main theorem.

Formal Definitions

1. **Event**: An element `e ∈ E = L × T × Φ` with `e = (l, t, φ)`. We often write `e = (l,t)` when the effect `φ` is irrelevant to the discussion. Each event has a location `l ∈ L` and timestamp `t ∈ T`.

2. **Causal Delay**: For two events `e1, e2`, if `e2` can be caused by `e1`, define the *causal delay*

   ```
   δ(e1,e2) = e2.t - e1.t,
   ```

   provided `e2.t ≥ e1.t`. If `e1` cannot causally influence `e2` (for example, if they are spacelike separated), one may set δ undefined or note that no causal relation exists.

3. **Event-Time Interval**: For events `e1 = (l1, t1)` and `e2 = (l2, t2)`, define

   ```
   Δt = t2 - t1,  Δx = d(l1,l2),
   ```

   and the *event-time interval*

   ```
   (Δs)^2 = c^2 (Δt)^2 - (Δx)^2.
   ```

   This scalar quantity is frame-independent (as we will show). If `(Δs)^2 > 0` with `Δt > 0`, the events are **timelike separated**; if `(Δs)^2 < 0` they are **spacelike separated**.

4. **Event-Time Frames**: An **event-time frame** `F` is a linear coordinate system for events. In one dimension, each frame assigns to an event `e` coordinates `(x_F(e), t_F(e))` by linear maps. We consider transformations between inertial frames of the form

   ```
   t' = γ (t - v x / c^2),
   x' = γ (x - v t),
   ```

   as above, which preserve the form of the interval. In higher dimensions one includes rotations or boosts accordingly.

With these, we can state and prove the invariance theorem.

Event-Time Invariance Theorem

Theorem (Event-Time Interval Invariance). *For any two events* `e1, e2`*, the squared event-time interval* `(Δs)^2 = c^2 (Δt)^2 - (Δx)^2` *is the same in all inertial event-time frames. In particular, under any Lorentz-like transformation between frames, the interval is invariant.*

Proof: Without loss of generality consider a one-dimensional spatial coordinate `x` and an event-time frame transformation with relative velocity `v` as above. Let `Δt = t2 - t1` and `Δx = x2 - x1` be the time and location differences of `e1, e2` in frame `F`. In frame `F'` moving at speed `v` relative to `F`, their coordinates become `t'_i = γ (t_i - v x_i / c^2)` and `x'_i = γ (x_i - v t_i)` for `i=1,2`. Then the differences are `Δt' = t'_2 - t'_1 = γ (Δt - v Δx / c^2)` and `Δx' = x'_2 - x'_1 = γ (Δx - v Δt)`. Compute the new interval:

```
c^2 (Δt')^2 - (Δx')^2
= c^2 [γ (Δt - v Δx / c^2)]^2 - [γ (Δx - v Δt)]^2
= γ^2 [ c^2 (Δt - v Δx / c^2)^2 - (Δx - v Δt)^2 ].
```

Expanding inside the brackets yields (after algebraic simplification using `γ^2 = 1/(1-v^2/c^2)`):

```
γ^2 [ c^2 Δt^2 - 2 v Δt Δx + (v^2/c^2) Δx^2 - (Δx^2 - 2v Δx Δt + v^2 Δt^2) ]
= γ^2 [ c^2 Δt^2 - v^2 Δt^2 - Δx^2 + (v^2/c^2) Δx^2 ]
= γ^2 [ (c^2 - v^2) Δt^2 - (1 - v^2/c^2) Δx^2 ]
= (Δt^2 (c^2 - v^2) - Δx^2 (c^2 - v^2) )
= c^2 Δt^2 - Δx^2.
```

Thus one finds

```
c^2 (Δt')^2 - (Δx')^2 = c^2 (Δt)^2 - (Δx)^2,
```

which shows `(Δs')^2 = (Δs)^2`. This proves the interval is invariant under the transformation. Since any composition of such Lorentz boosts (and spatial rotations) also preserves the interval, the event-time interval is frame-invariant.

$\square$

This invariance implies that **timelike** (causally connected) versus **spacelike** (causally independent) classification of two events is agreed upon in all frames. Moreover, the numerical value of the interval `(Δs)^2` is a frame-independent measure of separation, providing an intrinsic metric on the set of events.

Analogy with Spacetime Physics

The above construction closely parallels the geometry of spacetime in special relativity. In relativity, each "event" is a point `(x,t)` in Minkowski space, with the invariant interval `c^2Δt^2 - Δx^2` preserved by Lorentz transformations [5]. Likewise, in event-time geometry each event `(l,t)` has coordinates and an invariant interval.

- **Causal Cones:** As in relativity one can draw *causal cones* around each event. The **future light (event) cone** of `e1` consists of all events `e2` such that `(Δs)^2 ≥ 0` and `e2.t > e1.t` (timelike or lightlike future). Analogously the **past cone** is events that can influence `e1`. Events outside these cones have `(Δs)^2 < 0` (spacelike) and cannot be causally connected. In physics language: only within the cones can information travel between events [6] [7]. Thus event-time geometry imposes a strict causal structure: as the Stanford Encyclopedia notes, *"anything outside of the light cone of p cannot causally interact with p"* [6] [7]. In computing terms, if two events are too far apart in location relative to their time difference, they are causally independent.

- **Relativity of Simultaneity:** Just as in relativity two observers moving relative to each other disagree on whether spatially separated events are simultaneous [8], different event-time frames (different clock domains or schedules) may also order independent events differently. For example, consider two events `A` and `B` occurring at equal times in some frame (simultaneous in that frame). In a frame moving relative to the first, the transformed times `t'_A, t'_B` will generally differ (unless the motion is perpendicular to the line connecting them) [8]. Concretely, two events in distant locations that appear simultaneous in one execution clock may have different timestamps in another clock domain. However, the invariant interval ensures that **causally ordered** events cannot flip order: if `A → B` (i.e. `(Δs)^2 ≥ 0` and `t_B > t_A`), all frames will agree `t'_B > t'_A`. This exactly mirrors how Lorentz transformations preserve the temporal order of timelike-separated events. In event-driven computation this means true causality (dependence) is preserved across scheduling: only independent events may "swap" order in different frames.

In summary, the formal similarity is strong: **event-time geometry** treats asynchronous computation much like spacetime treats physical events. We have lightlike cones of influence, invariant intervals, and frame-dependent simultaneity, all in service of a causal model.

Implications for Multi-Frame Execution Models

In practical computing, an "event-time frame" can correspond to a particular execution context or clock domain. For instance, a multi-core CPU or distributed system might have local clocks or pipelined stages—each a different frame with its own view of time. The event-time formalism provides a coherent way to reason across these frames:

- **Unified Causality:** By assigning absolute coordinates `(l,t)` to events in some agreed reference frame (or transforming as needed), one can ensure that causally dependent events are correctly ordered regardless of where they occur. The happened-before partial order [1] [2] is embedded in this geometry: if `e1 → e2`, then `(Δs)^2 ≥ 0` and `e2` lies in the future cone of `e1`. No transformation will invert their order, so all frames respect the causality.

- **Frame Transformations:** Different execution frames may run at different speeds (analogous to relative velocity `v`). The Lorentz-like transformations of event-time allow mapping timestamps from one frame to another. For example, if two processors have clocks running at slightly different rates or with offset, we can model this as a boost and compute how one processor's timestamp corresponds to the other's. This can help synchronize logs or replay events consistently.

- **Partition and Concurrency:** Events that are **spacelike separated** (`(Δs)^2 < 0`) do not causally affect each other, so they can be executed or recorded in any order without affecting correctness. This is analogous to how in relativity the order of spacelike events is frame-dependent. Recognizing spacelike separations allows parallel execution and optimizations: concurrent tasks can proceed independently.

- **Time and Error Budgets:** The interval gives a quantitative measure of allowable time slack. If an effect must propagate from `e1` to `e2`, the condition `e2.t ≥ e1.t + d(e1.l,e2.l)/c` must hold. This yields an **error budget** for clock drift or processing delay: one can tolerate small frame discrepancies as long as the invariant interval remains timelike. In high-performance pipelines, this can guide how much latency can be introduced before violating causality.

Overall, the event-time model ensures **reproducibility** and **interoperability**: by encoding computation as events in a geometric space, one can replay or migrate workloads across different hardware or simulation environments while preserving the causal relationships. Any two conforming event-time frames will yield the same end-to-end behavior for a causal computation.

Applications to Neuromorphic Computing

The event-time formalism has direct applications to neuromorphic systems and spiking neural networks (SNNs):

- **ANN-to-SNN Translation:** Converting a conventional frame-based Artificial Neural Network (ANN) into an event-driven SNN is a major challenge. Using event-time geometry, one can map the continuous activations of an ANN to spike *events* with precise timestamps. For example, each neuron's firing can be assigned a time `t` proportional to its analog activation. The invariant interval then ensures that the timing differences encode the original computation order. In practice, ANN-to-SNN conversion techniques rely on aligning average firing rates with ANN activations [9], but an event-time view emphasizes exact causality: the *relative timing* of spikes conveys the same information as the original signals. By imposing an event-time metric, one can derive conversions that preserve the spatiotemporal ordering of neural activity, potentially yielding more faithful SNN emulations of ANNs.

- **Preservation of Causality Across Platforms:** Neuromorphic hardware (like IBM TrueNorth or Intel Loihi) and simulators may process spikes at different speeds or with different latencies. If each platform uses an event-time framework, then a spike occurring at time `t` on one platform can be translated to time `t'` on another via a frame transformation. The invariant interval guarantees that causally related spikes remain so. In other words, an SNN run on GPU vs. run on ASIC will produce outputs that differ only by a frame-dependent offset, not by changed causal order. This enables

interoperability: spike trains recorded on one device can be replayed on another without altering the computation, as long as timestamps are transformed consistently.

- **Error Budgeting and Latency Management:** Neuromorphic engineers can use the interval metric to budget delays and jitters. For instance, if a spike must drive another neuron, the causal delay `δ` must exceed the minimal propagation time `d/c`. Any hardware-induced jitter must be less than this window to avoid causal violation. This quantification aids in designing neuromorphic pipelines where exact timing matters.

- **Reproducibility and Debugging:** Since the event-time model encodes an absolute ordering structure, it can help reproduce experiments. Given the same stream of input events with their timestamps, a neuromorphic network will produce the same outputs irrespective of scheduler differences. This is crucial for debugging and benchmarking SNNs, where nondeterminism is a common issue.

In summary, framing neuromorphic computation in event-time geometry ensures that **causality is explicitly modeled and preserved**. This leads to practical benefits: one can systematically translate frame-based algorithms into spike-timing-based ones, cross-validate neuromorphic implementations, and guarantee that asynchronous event streams are processed consistently. As spiking networks and event-driven sensors grow in importance, the event-time formalism provides the mathematical underpinning for *timing correctness*, *interoperability*, and *efficient scheduling* in neuromorphic AI.

Sources: We have drawn on concepts from concurrency theory and neuromorphic computing to develop this exposition. The Lorentz-invariance analogy is standard in physics [5], and discussions of causality and light cones are found in physics literature [6] [7]. Event-based neuromorphic systems are described in recent works [3] [4]. Classical causal ordering in distributed systems is given by Lamport's happened-before relation [1] [2]. These sources inspire the formal event-time definitions and theorems presented above.

---

[1] [2] Happened-before - Wikipedia
https://en.wikipedia.org/wiki/Happened-before

[3] A spiking neural network model of 3D perception for event-based neuromorphic stereo vision systems | Scientific Reports
https://www.nature.com/articles/srep40703?error=cookies_not_supported&code=a8903b50-271c-4856-92ca-6f91a2aeddc7

[4] [9] Frontiers | Optimizing event-driven spiking neural network with regularization and cutoff
https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2025.1522788/full

[5] Lorentz transformation - Wikipedia
https://en.wikipedia.org/wiki/Lorentz_transformation

[6] [7] Singularities and Black Holes > Light Cones and Causal Structure (Stanford Encyclopedia of Philosophy)
https://plato.stanford.edu/entries/spacetime-singularities/lightcone.html

[8] Relativity of simultaneity - Wikipedia
https://en.wikipedia.org/wiki/Relativity_of_simultaneity