# Event2Vec: A Geometric Approach to Learning Composable Representations of Event Sequences


## Antonin Sulc

Lawrence Berkeley National Laboratory, Berkeley, CA, U.S.A. ASULC@LBL.GOV


BERKELEY LAB

U.S. DEPARTMENT OF ENERGY
Office of Science

## 1. Motivation: Word2Vec for Events

Neural activity and real-world tasks often unfold on low-dimensional manifolds. We aim to translate the success of Word2Vec to discrete event sequences.

- **The Goal:** Learn **composable** embeddings where *History* $= \sum$ *Events*.
- **The Nuance:** While Euclidean addition is commutative ($A + B = B + A$), our training data is strictly sequential ($A \rightarrow B$).
- **Inference:** Ideally, the learned embedding space should effectively ignore the order of summation during inference, allowing us to treat a trajectory as the "sum of its parts" for vector arithmetic ($A - B + C = D$).

## 2. Geometric Intuition

The model learns to align event vectors into coherent trajectories. Unlike standard RNNs, we enforce a geometry where displacement equates to meaning.

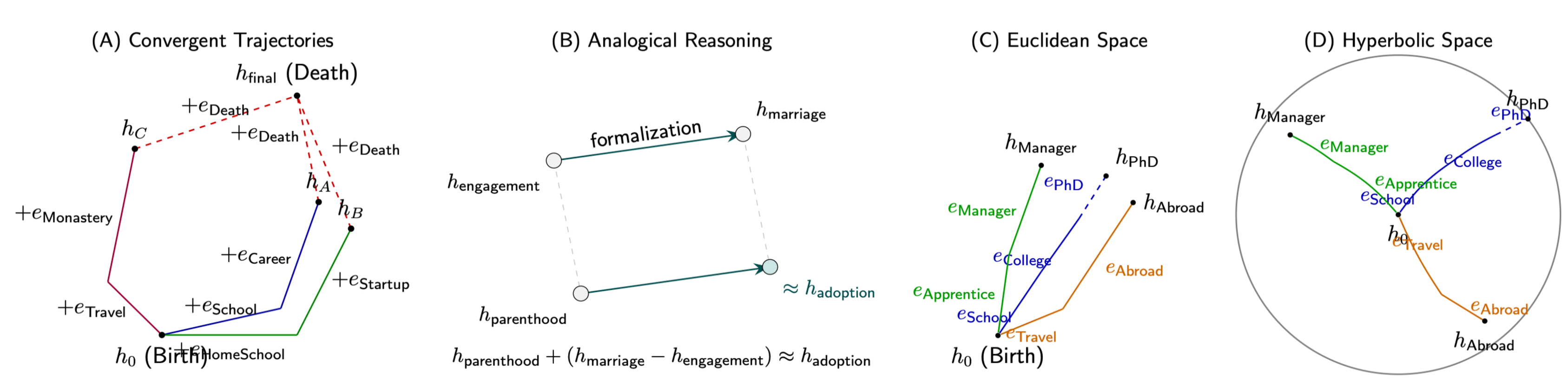



Figure: **(A)** Probable sequences form aligned trajectories (e.g., School → Career). **(B)** The additive structure enables analogical reasoning: applying the "formalization" vector (Engagement → Marriage) to Parenthood yields Adoption.

## 3. Two Geometric Variants

We propose an additive recurrent framework $h_t = h_{t-1} \oplus e_{s_t}$:

### A. Euclidean Model (Flat)

$$h_t = h_{t-1} + e_{s_t} \quad \text{(with norm clipping)}$$

Simple vector addition. Interpretable, but distinct paths eventually crowd together in finite space.

### B. Hyperbolic Model (Curved)

$$h_t = h_{t-1} \oplus_c e_{s_t}$$

Uses **Möbius addition** in the Poincaré ball. Ideally suited for hierarchical branching.

Hyperbolic space is a Gyrovector space. Addition is **Gyrocommutative**:

$$a \oplus b = \mathrm{gyr}[a, b](b \oplus a)$$

*(Intuitively: The order of operations affects the final orientation due to the curvature of the space, preserving the sequence history in the vector structure.)*

## 4. Theoretical Guarantees

We minimize a composite loss $\mathcal{L} = \mathcal{L}_{pred} + \mathcal{L}_{recon}$.

### Theorem 1: Justification for Additivity

Minimizing reconstruction loss $\mathcal{L}_{recon} = \sum \|(h_t - e_{s_t}) - h_{t-1}\|^2$ enforces strict reversibility: $f(f(h, e), -e) = h$.

- **Result:** The simplest function satisfying this is linear addition. The model *must* learn to sum vectors to satisfy the loss.

### Theorem 2: Semantic Grounding

The prediction loss $\mathcal{L}_{pred}$ (log-likelihood) acts like Skip-Gram Negative Sampling.

- **Result:** It forces the inner product $h_t^T e_{s_{t+1}}$ to approximate the Pointwise Mutual Information (PMI).
- **Synthesis:** $\mathcal{L}_{pred}$ aligns likely events; $\mathcal{L}_{recon}$ ensures the path to them is additive.

## 6. Analogical Reasoning ($A - B + C \approx D$)

The additive structure allows us to reason about abstract concepts via vector arithmetic.

### 1. The "Life Transition" Vector

$$\text{Death} - \text{Retirement} + \text{Graduation} = \textbf{Internship}$$

The model learns the abstract concept of "ending a major phase." Removing 'Retirement' from 'Death' isolates this transition vector; adding it to 'Graduation' (ending school) correctly predicts 'Internship' (starting work).

### 2. The "Formalization" Vector

$$\text{Marriage} - \text{Engagement} + \text{Parenthood} = \textbf{Adoption}$$

The vector difference between Engagement and Marriage represents legal/formal solidification. Applied to Parenthood, the model identifies Adoption as the formalized equivalent.

### 3. The "Financial Outcome" Vector

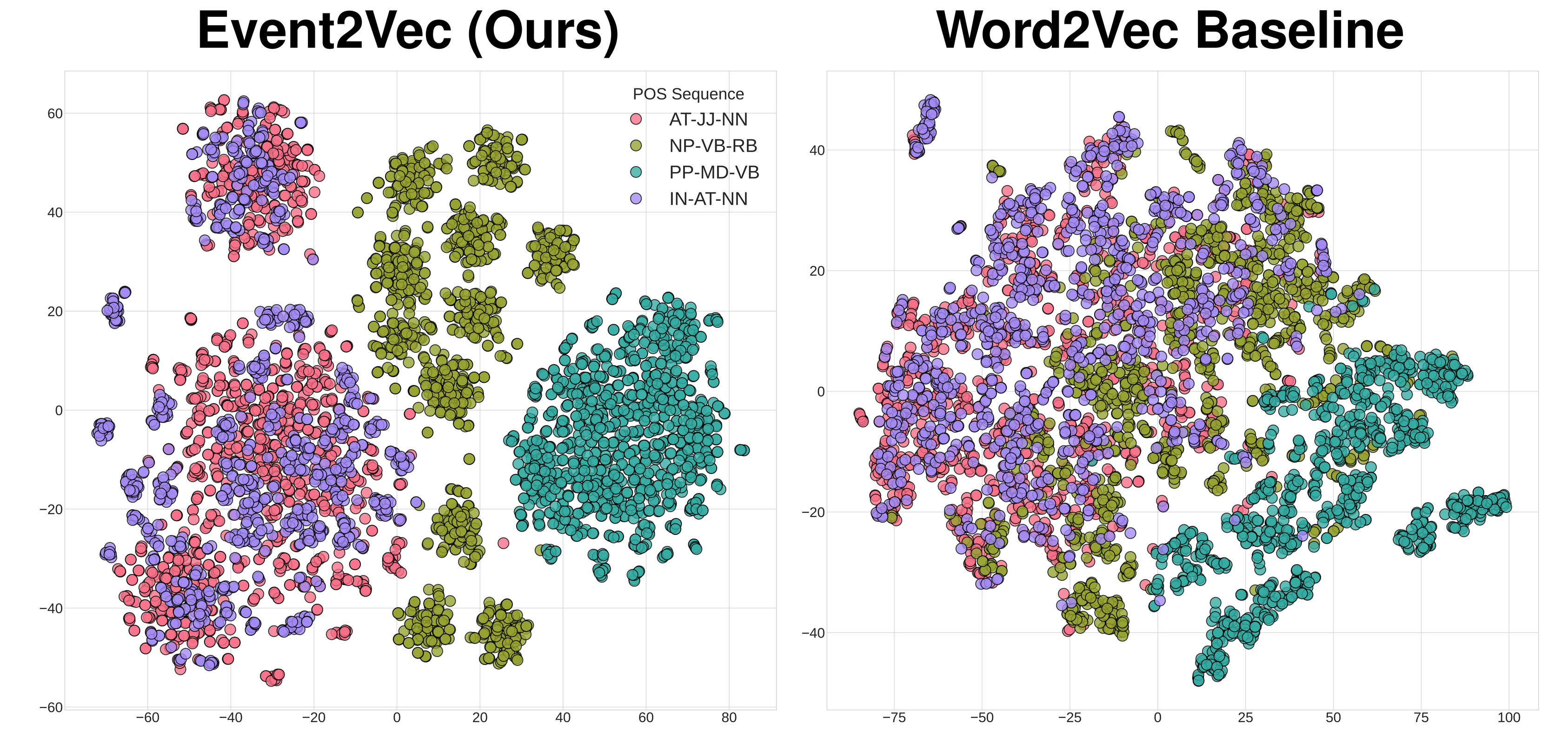
$$\text{Business Success} - \text{Entrepreneurship} + \text{Investment} = \textbf{Inheritance}$$

The model extracts the "financial windfall" component from the specific act of entrepreneurship and applies it to investment, identifying inheritance as a semantically similar major financial gain.

## 7. Validation: Unsupervised Grammar Induction

Trained on raw text (Brown Corpus) without labels. We composed vectors for Part-of-Speech sequences by summing word embeddings.

**Event2Vec (Ours)** | **Word2Vec Baseline**

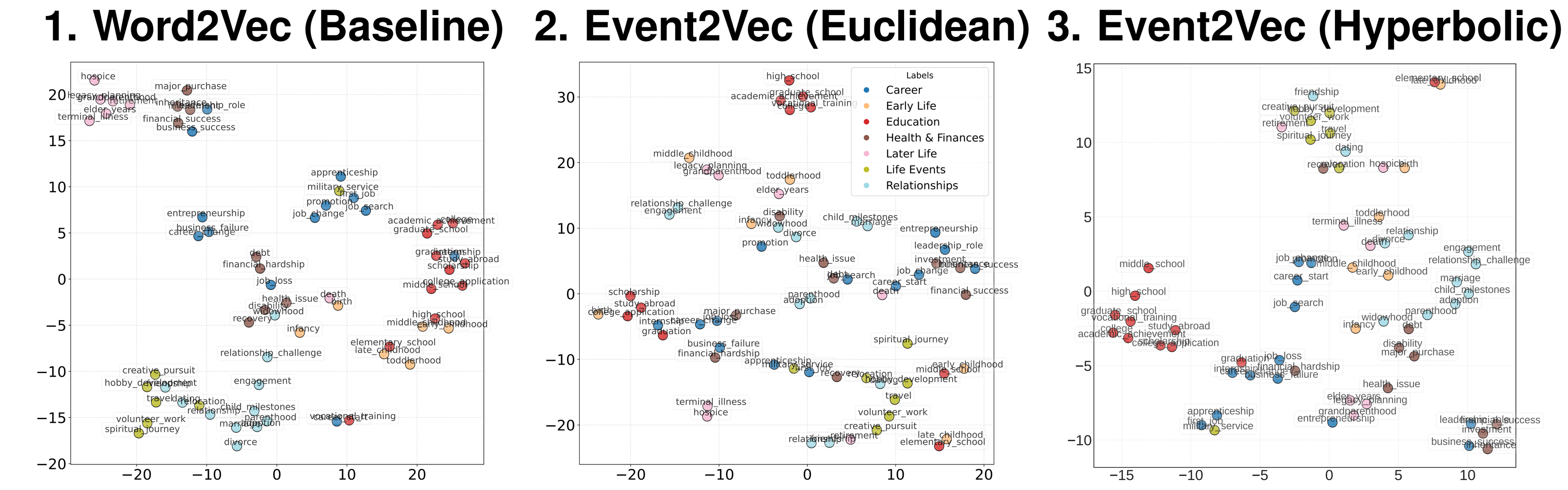



**Result:** Event2Vec automatically clusters distinct grammatical structures (e.g., 'Article-Adj-Noun') significantly better than the baseline.

- **Silhouette Score: 0.0564** (Ours) vs 0.0215 (Baseline).

## 5. Life Path Trajectories

**1. Word2Vec (Baseline)** | **2. Event2Vec (Euclidean)** | **3. Event2Vec (Hyperbolic)**



Clusters by semantic context. (Birth is near Death). | Captures chronological time. (Linear progression). | Captures hierarchical branching. (Radial expansion).

## 8. Summary

- We successfully translated the "Linear Additive Hypothesis" to event sequences.
- **Euclidean:** Good for linear timelines and simple arithmetic.
- **Hyperbolic:** Superior for hierarchical/branching processes due to Gyro-geometry.
- **Impact:** Provides a path toward geometric mechanistic interpretability for sequential data.