DEMI Algorithm

Temporal Order

Step 1 of the DEMI Pipeline — Temporal Order of Variables

This video focuses on Step 1 only

Why Order Matters

Causes must precede effects

Causal inference depends on directionality.

Two Sources of Ordering Evidence

Hybrid approach: conceptual + empirical

Conceptual

Domain knowledge, clinical logic, prior literature

Empirical

Observed timing patterns in patient-level data

Age – Anchor Variable

Age exists at birth; ordered first

Fixed at birth → temporally first among all variables

Add Treatment

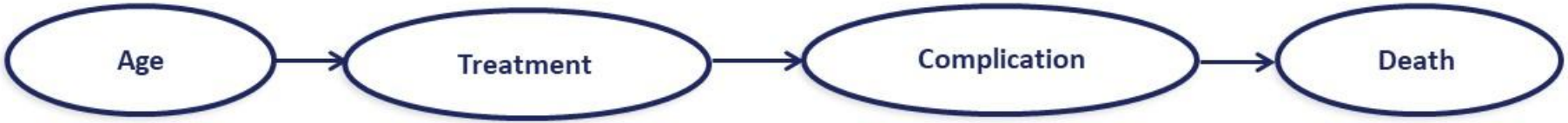
Occurs after age

Add Complication

Occurs after treatment

Add Outcome

Death is the terminal variable



Each variable added in the order it occurs in the clinical timeline.

Pairwise Scoring

Quantifying directional evidence for each variable pair

Pairwise Setup

Compare variables X and Y

For every pair (X, Y) we collect three counts from the data:

`n_before`	Patients where X occurs before Y
`n_after`	Patients where Y occurs before X
`n_absent`	Patients where X occurs but Y is absent

Evidence: X Before Y

n_before

$$C(X \prec Y) \rightarrow \text{count patients where } X \text{ timestamp} < Y \text{ timestamp}$$

```python
n_before = count(X before Y)
```

Evidence: Y Before X

n_after

$$C(Y \prec X) \rightarrow \text{count patients where } Y \text{ timestamp} < X \text{ timestamp}$$

```python
n_after = count(Y before X)
```

Evidence: X Occurs, Y Absent

n_absent – temporal implication

$$C(X \wedge \neg Y) \rightarrow X \text{ present, } Y \text{ never recorded for this patient}$$

```python
n_absent = count(X and not Y)
```

Pairwise Score

Net directional signal for X → Y

$$Score(X, Y) = n\_before + n\_absent - n\_after$$

Python

```
score_xy = (n_before + n_absent) - n_after
```

Aggregation & Sorting

From pairwise scores to global variable order

Aggregate Scores

Sum pairwise scores over all variable pairs

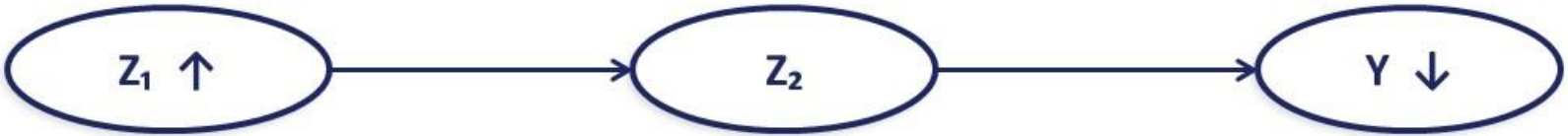
$$Score(Z_i) = \Sigma_j \; Score(Z_i, Z_j)$$

Python

```
scores[zi] = sum(score_z_ij for j in variables)
```

Sorting

Rank variables by descending score



$$Order = sort(scores, descending)$$

```python
sorted_scores = sorted(scores, reverse=True)
```

Final Temporal Order

Feeds directly into DEMI Step 2

$$Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow \ldots \rightarrow Y$$

Code Walkthrough

Step-by-step implementation

Initialize Scores

Initialize empty scores dict

```python
scores = {}
```

Loop Over Variables

Iterate over each variable Z_i

```python
for zi in variables:
```

Retrieve Pairs

Fetch pairwise data from KB

```python
pairs = get_pairs(zi)
```

Compute Differences

Core directional signal

```python
diff = pairs['before'] - pairs['after']
```

Aggregate

Sum over all j partners

```python
score = sum(diff_j for j in ...)
```

Store Result

Save into scores dictionary

```python
scores[zi] = score
```

Sort

Rank all variables

Python

```
sorted(scores, reverse=True)
```

Key Assumptions

Theoretical foundations

Transitivity

If $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$.

Cycles in the temporal order are disallowed. Any detected cycle is resolved by score magnitude.

Asymmetry

Variables can only precede each other in one direction.

The pairwise score resolves conflicting signals statistically.

Population

Ordering is derived from population-level data.

Individual patient trajectories may differ; the order reflects the dominant pattern.

Summary

1

Pairwise

Count n_before, n_after, n_absent for every variable pair

2

Score

Score = n_before + n_absent − n_after for each pair

3

Aggregate

Sum pairwise scores to get a global ranking score per variable

4

Sort

Sort variables by descending score → ordered DAG skeleton

Enables causal decomposition in Step 2