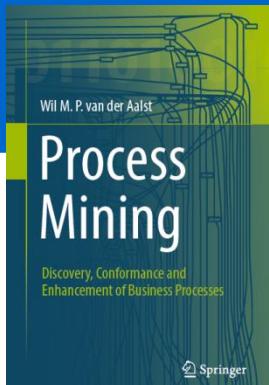


Process Mining: Data Science in Action

On the Representational Bias of Process Mining

prof.dr.ir. Wil van der Aalst
www.processmining.org



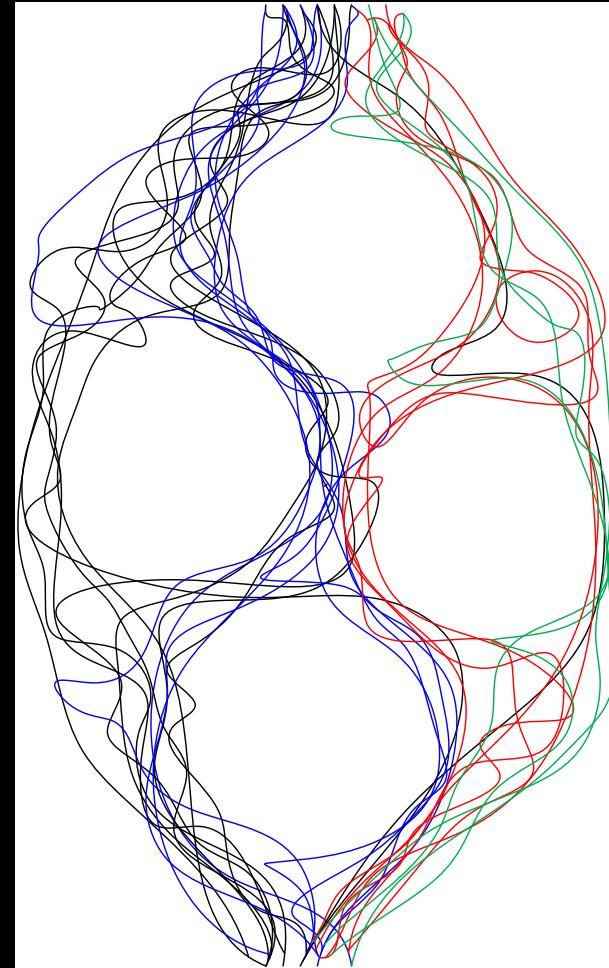
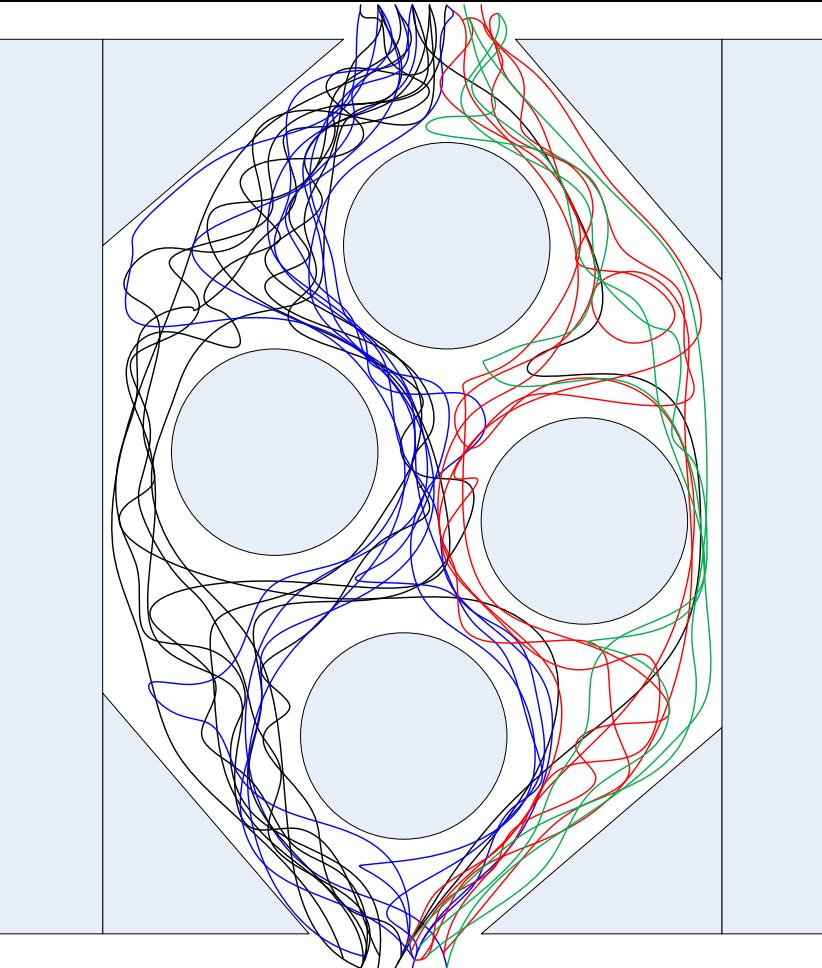
Technische Universiteit
Eindhoven
University of Technology

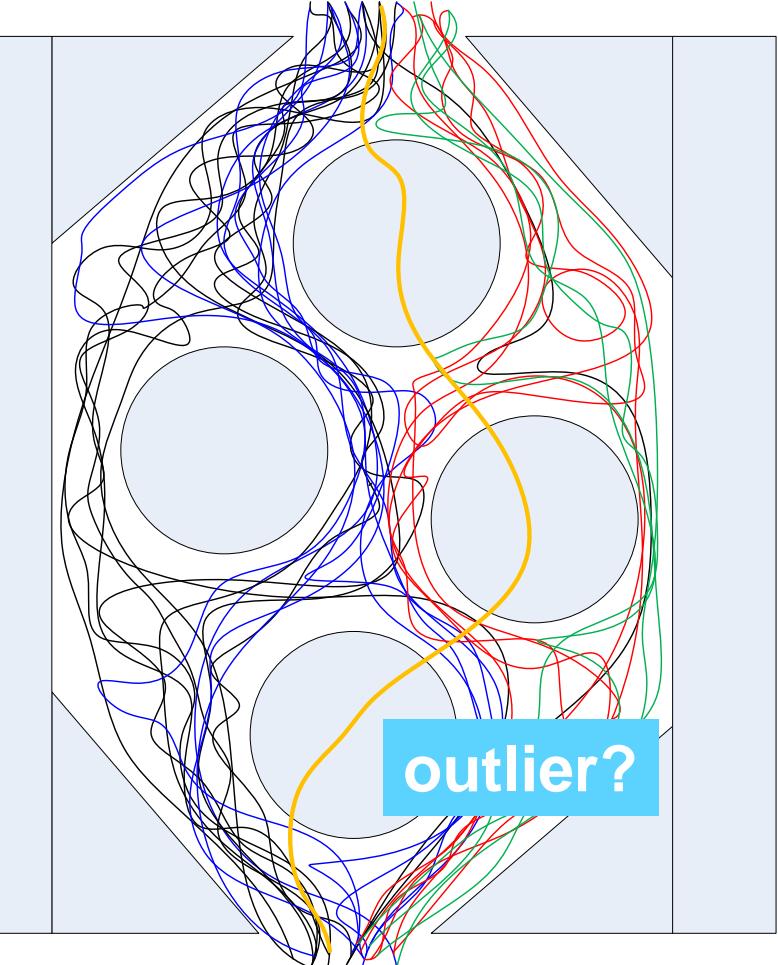
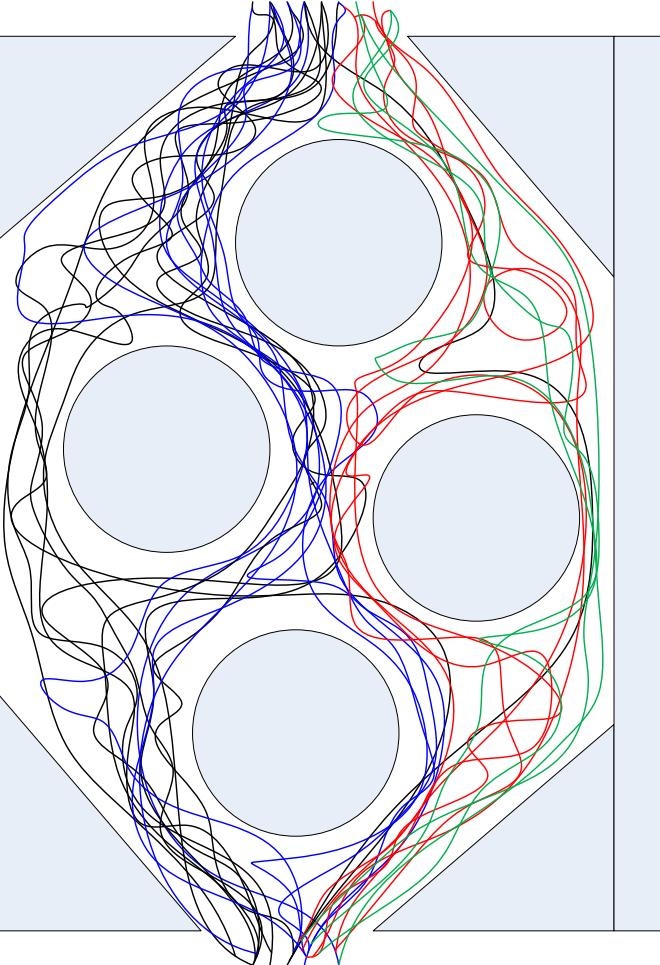
Where innovation starts

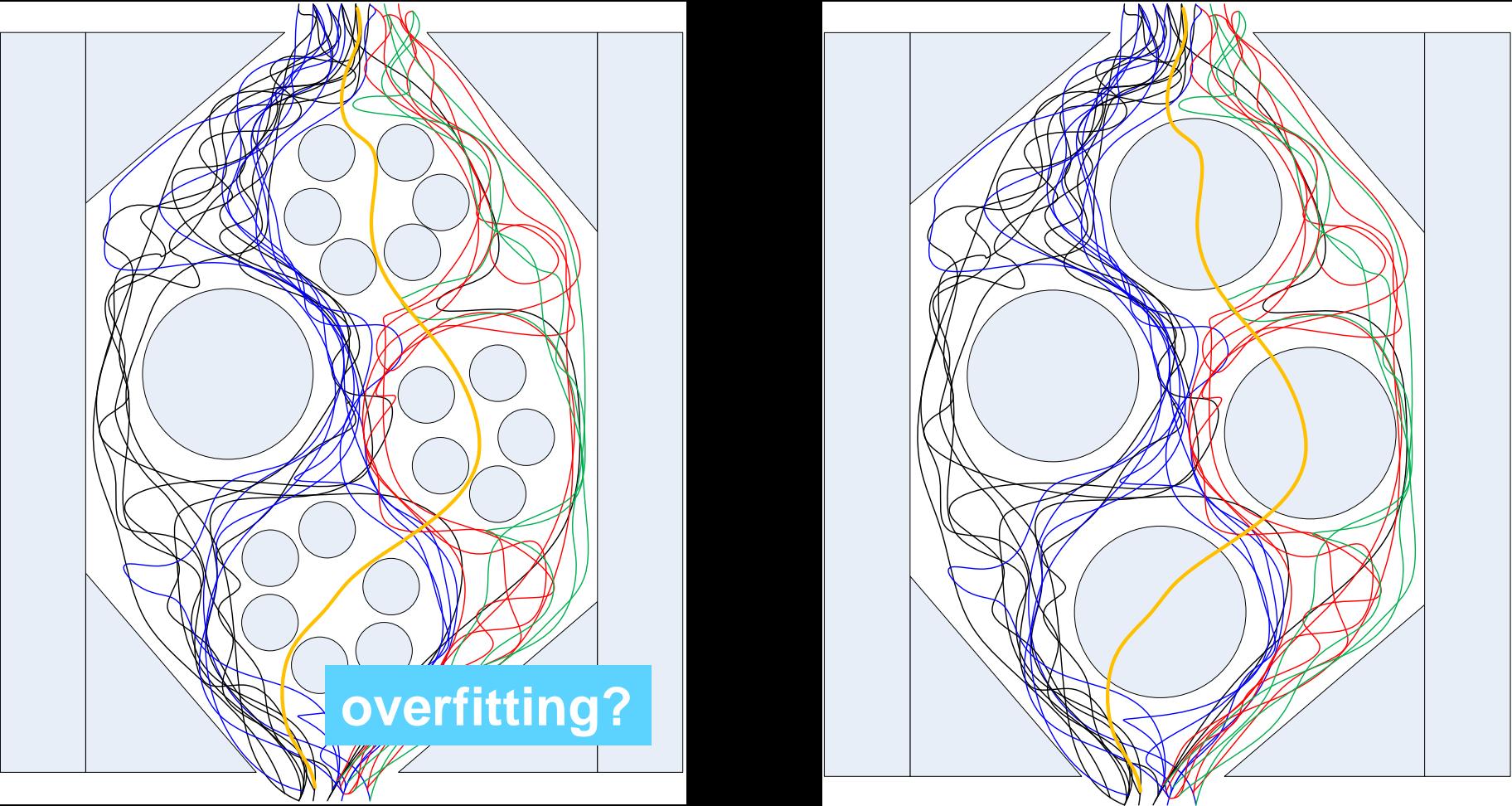


北京大学
小草对您感激万分
请您走路绕一绕
KEEP OFF GRASS



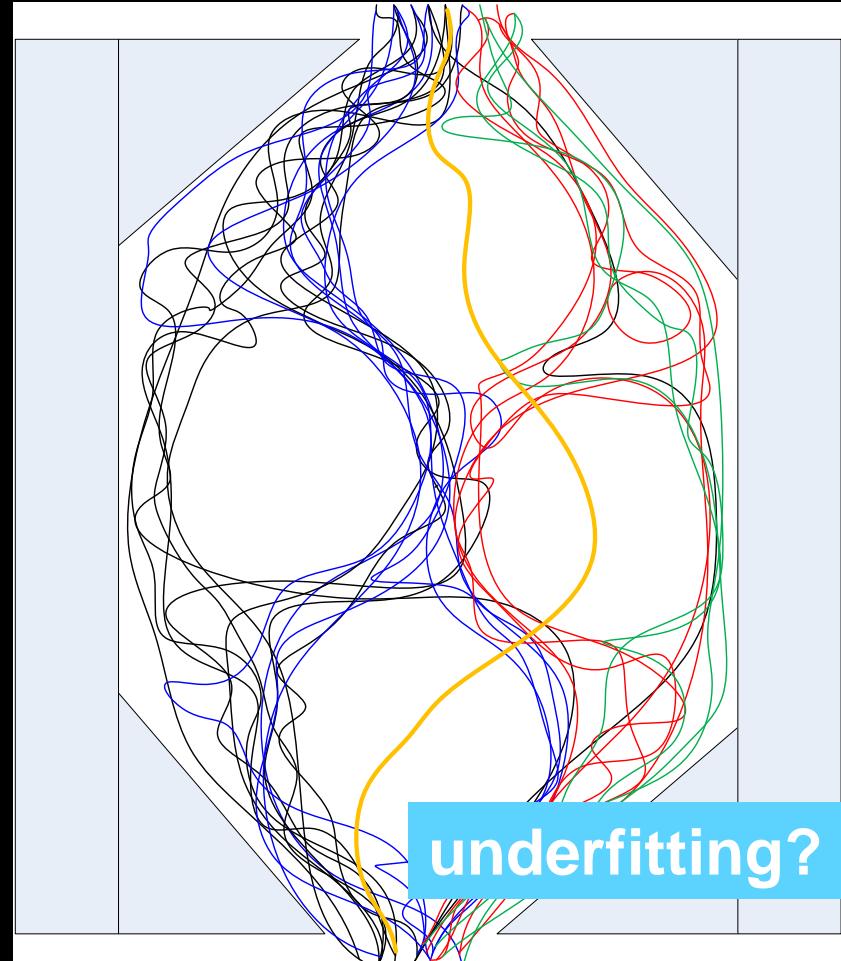
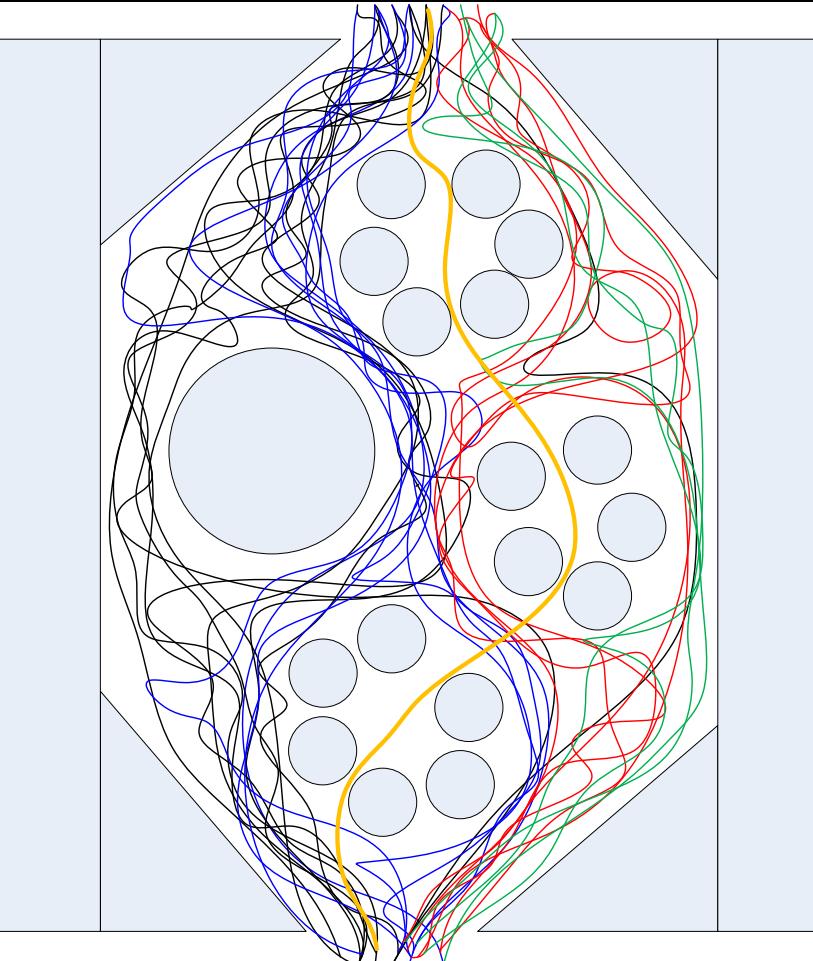




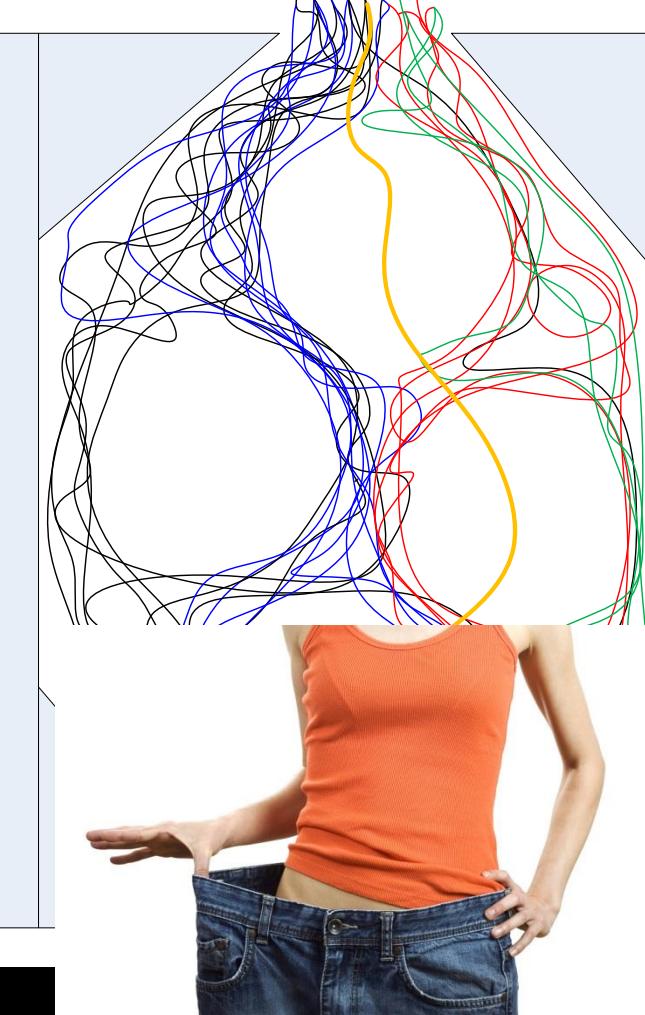
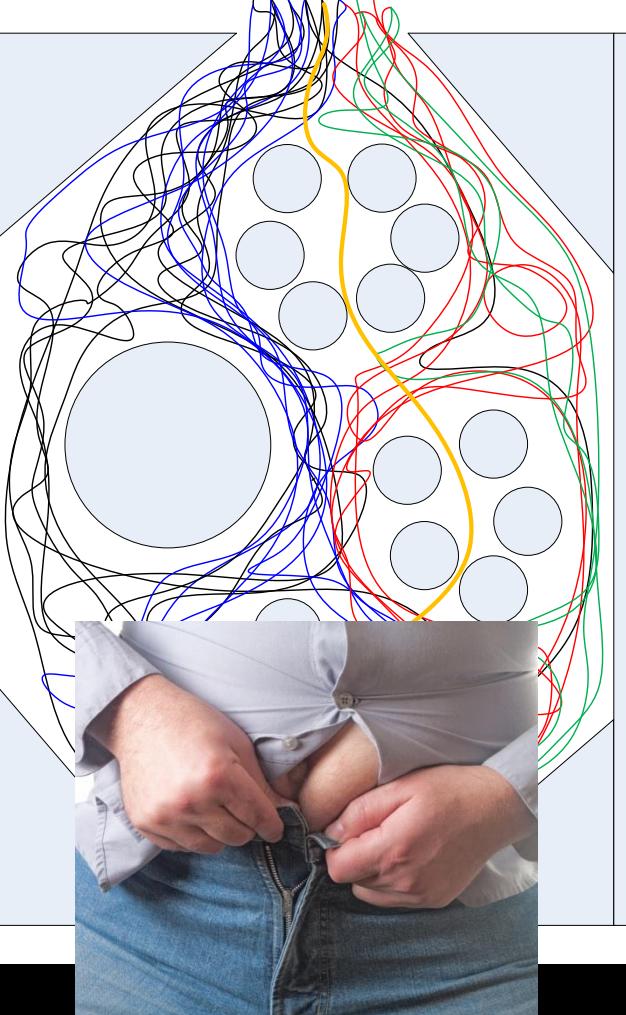


The diagram illustrates a neural network architecture with two input layers (represented by grey rectangles) and one output layer (represented by a grey triangle). The first input layer has three neurons, and the second has four. The output layer has one neuron. Colored lines represent weights between neurons: blue lines connect the first input layer to the first output neuron; red lines connect the second input layer to the first output neuron; and green lines connect the third input layer to the first output neuron. A yellow line represents the bias weight for the output neuron. The diagram shows that the network has learned to map all input patterns to the same output value, indicated by the single grey circle in the output layer. This results in a very complex, non-linear decision boundary (indicated by the many overlapping lines) that separates the different input classes. A blue callout box at the bottom left contains the text "overfitting?", suggesting that this complex model may be overfitting the training data.

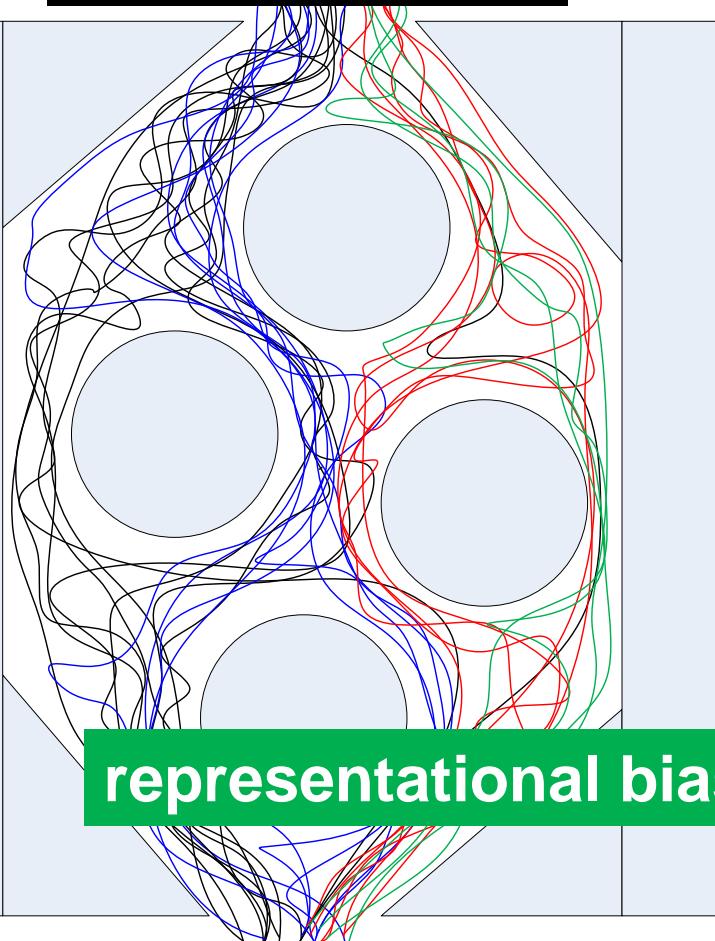
overfitting?



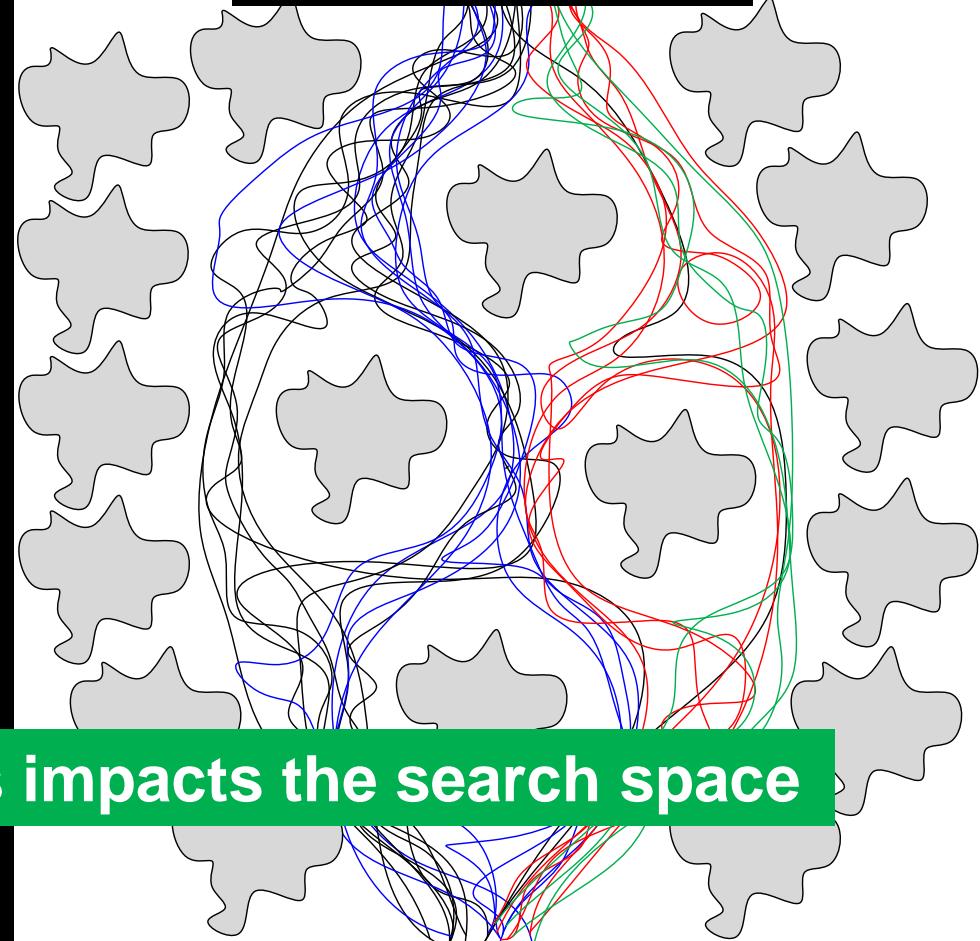
underfitting?



modeling language 1

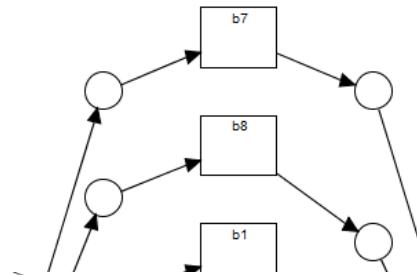


modeling language 2



representational bias impacts the search space

Representation



<10% of transition system
having the same behavior

Plethora of notations

Petri Nets
(many variants)

Transition
Systems

Workflow Nets

Flow Charts



French

Event-driven
Process Chains
(EPCs)



Dutch

State Charts



Italian

UML Activity
Diagrams



Spanish

Declare



German

Business Process
Modeling Notation
(BPMN)

Markov
Chains

Message
Sequence
Charts

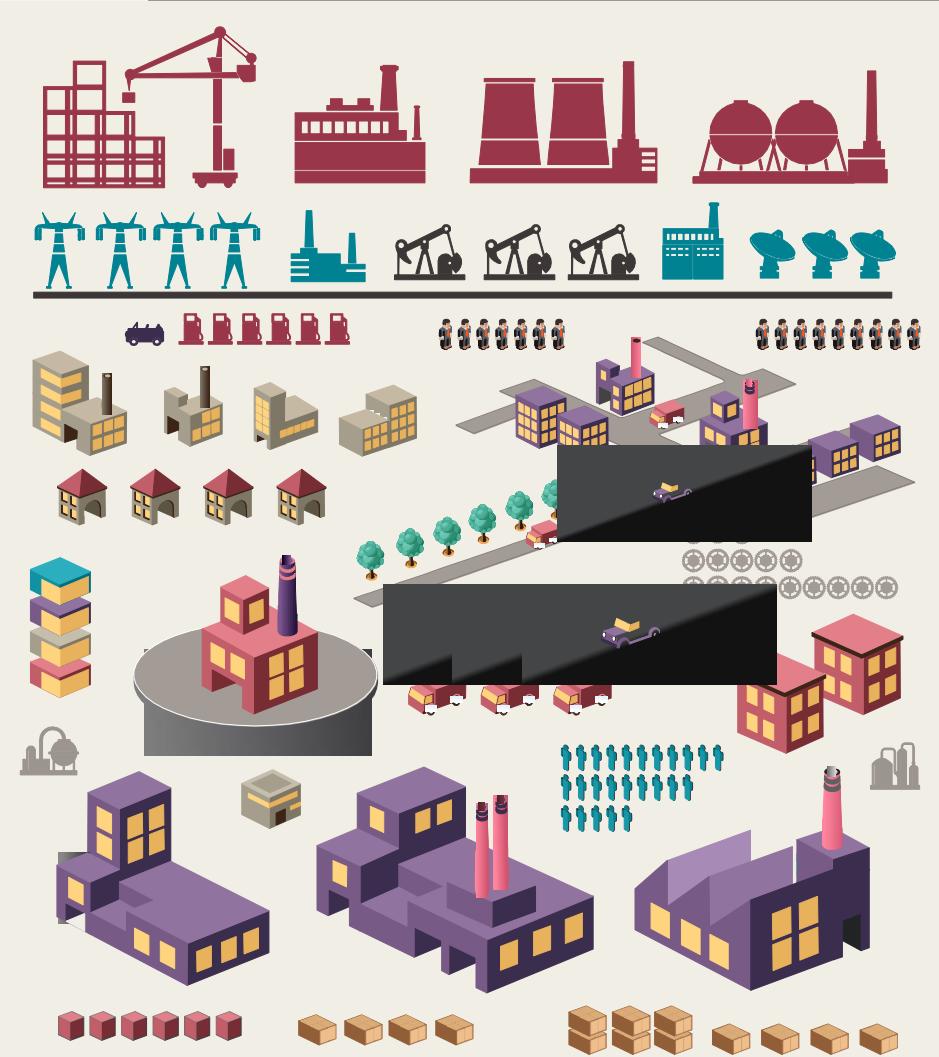
Process
Algebras

Business Process
Execution Language
(BPEL)

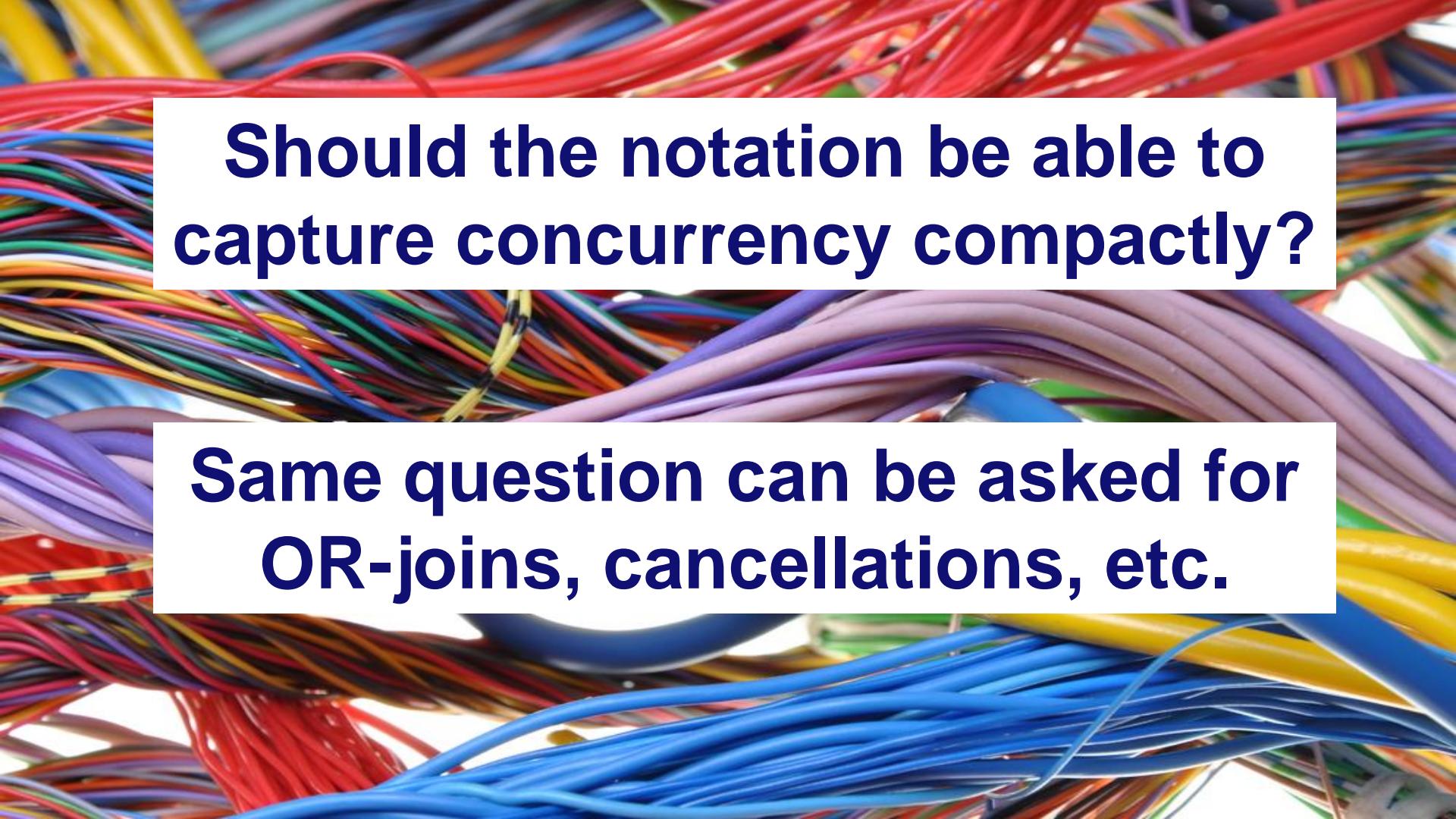
YAWL

let us consider concurrency in more detail ...









**Should the notation be able to
capture concurrency compactly?**

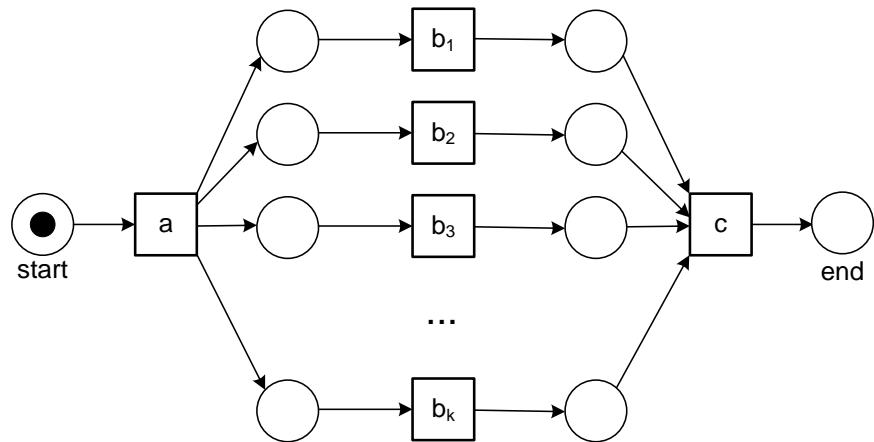
**Same question can be asked for
OR-joins, cancellations, etc.**

Question

How many traces in case of concurrency?

- Consider a process model with a start activity and end activity and in the middle k parallel activities.
- How many traces are possible (say $k = 10$)?
- Does the Alpha algorithm need to see all of these to rediscover the original model?

Model with k parallel activities



k	number of different traces: $k!$
1	1
2	2
5	120
10	3628800
20	2432902008176640000

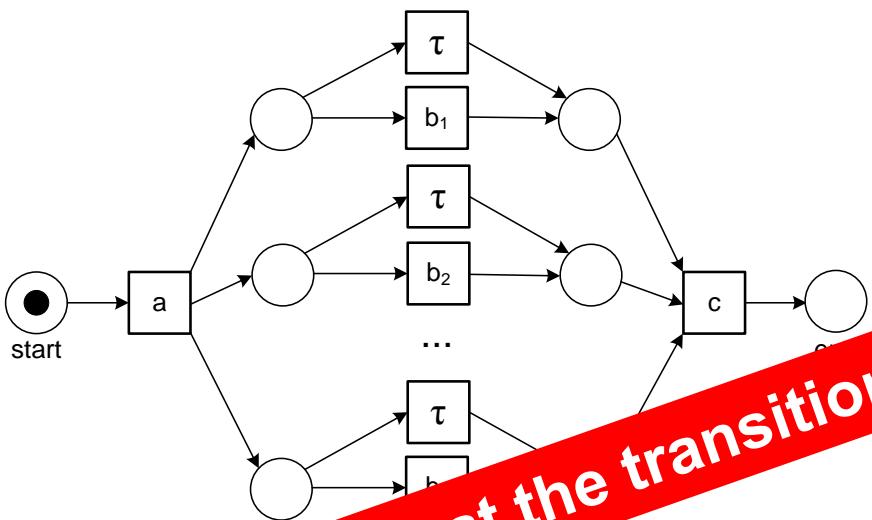
Alpha algorithm only needs to see $k(k-1)$ direct successions.

Question

How many traces in case of an OR-join?

- Suppose now that the k parallel activities are all **optional**.
- Is the Alpha algorithm able to discover such constructs?
- How many traces are possible?

Model with k optional activities



imagine what the transition system looks like

Alpha algorithm will not discover this model.

k	number of different traces
1	1
2	5
3	326
10	9864101
20	6613313319248080001

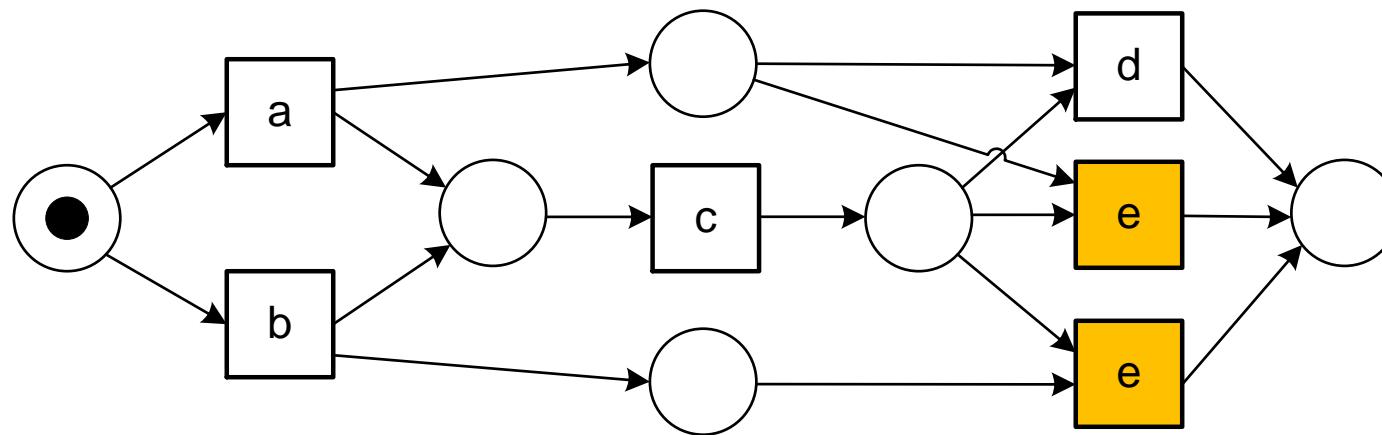
$$\sum_{i=0}^k \binom{k}{i} i!$$



- Highly unlikely to discover concurrency or OR-joins if such behaviors cannot be represented easily.
- Often the discovery process is guided by the representational bias!

Assumed Bias: WF-nets with unique labels

$$L = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}, \langle a, c, e \rangle^{20}]$$

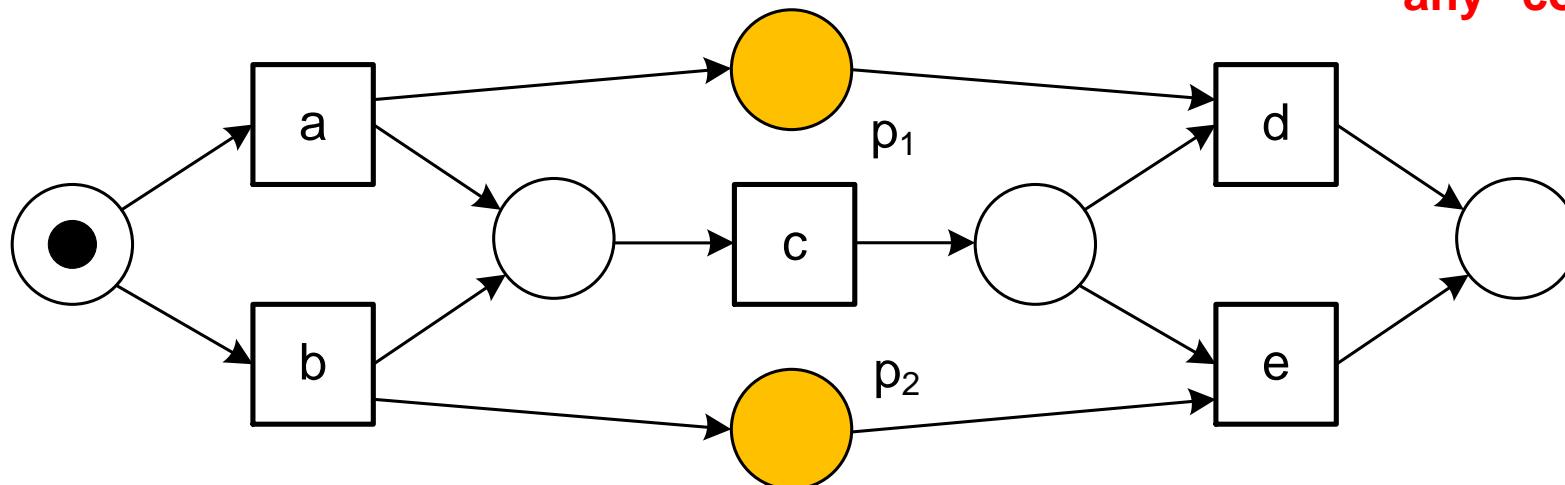


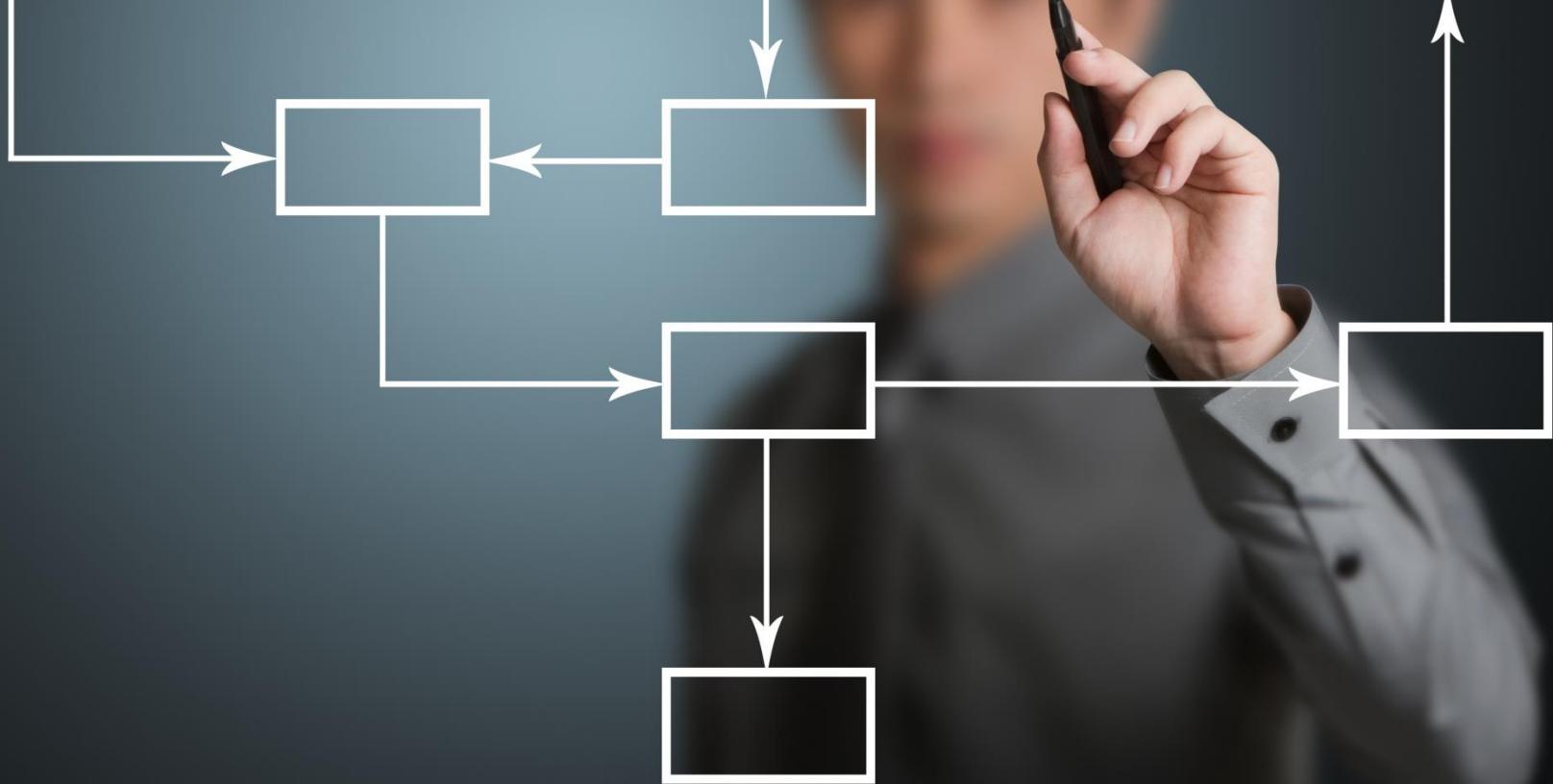
No WF-net with unique labels can be discovered that has precisely this behavior!

Assumed Bias: No indirect dependencies

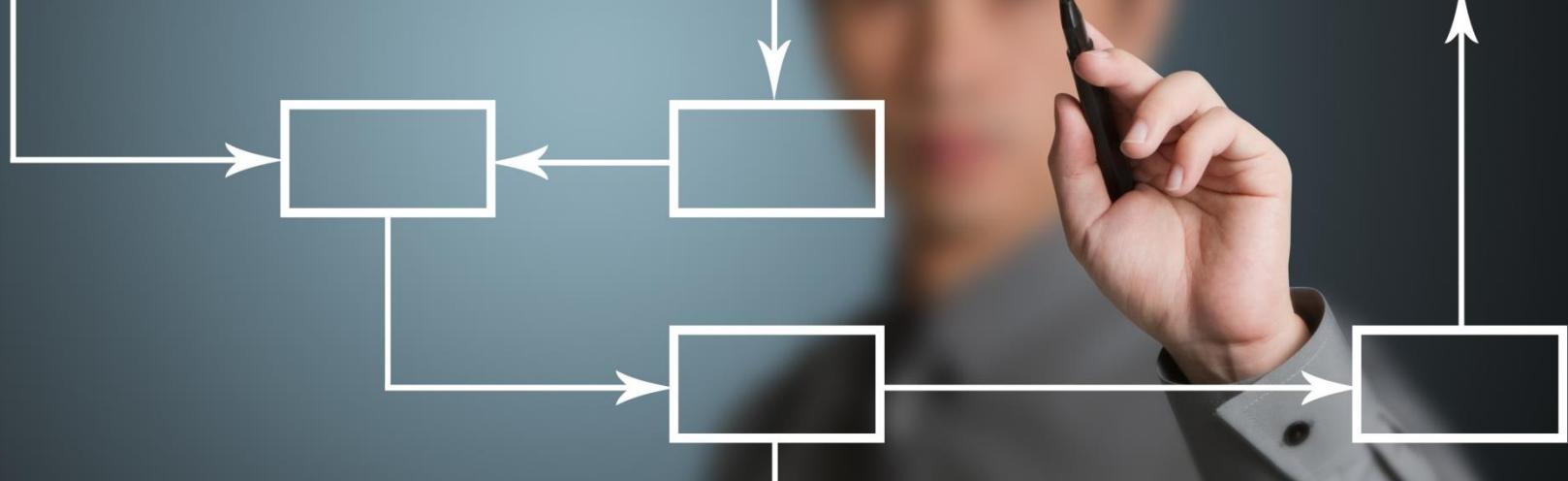
$$L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}]$$

Activities a and b influence the choice for d and e but there is never any "contact"!





- **Representational bias matters!**
- Impacts search space of process discovery

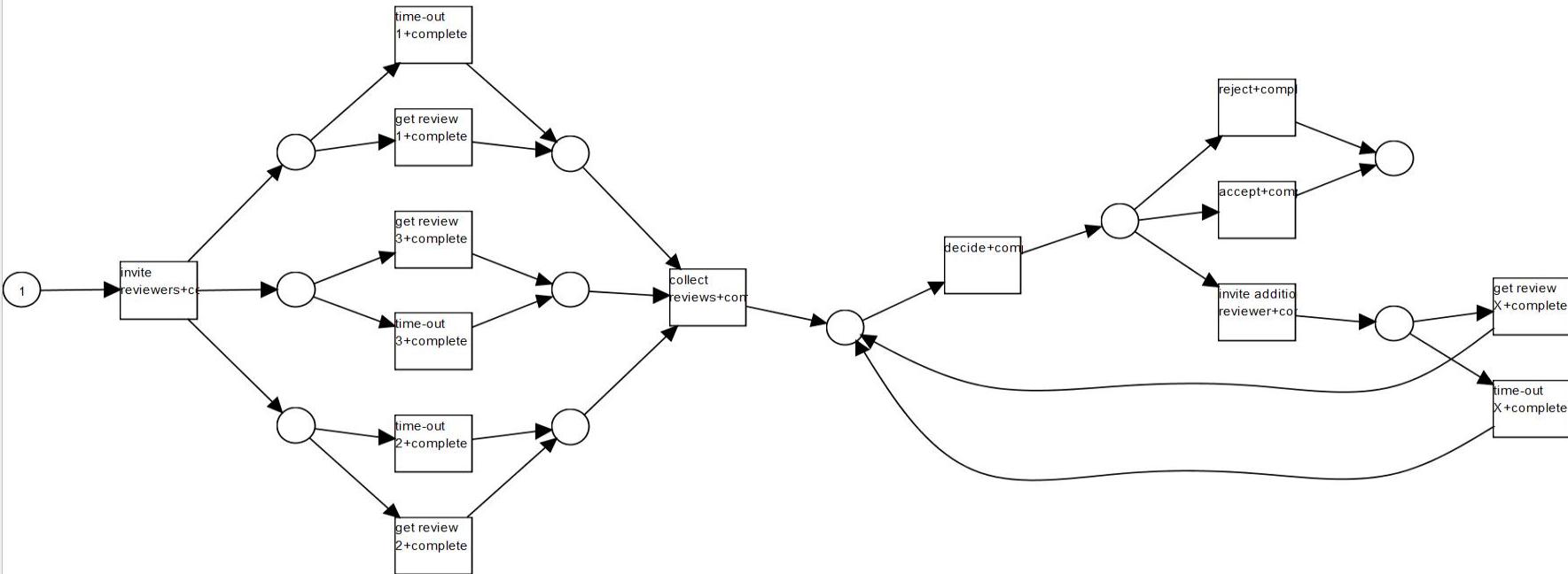


visualization of discovered model
≠
representational bias

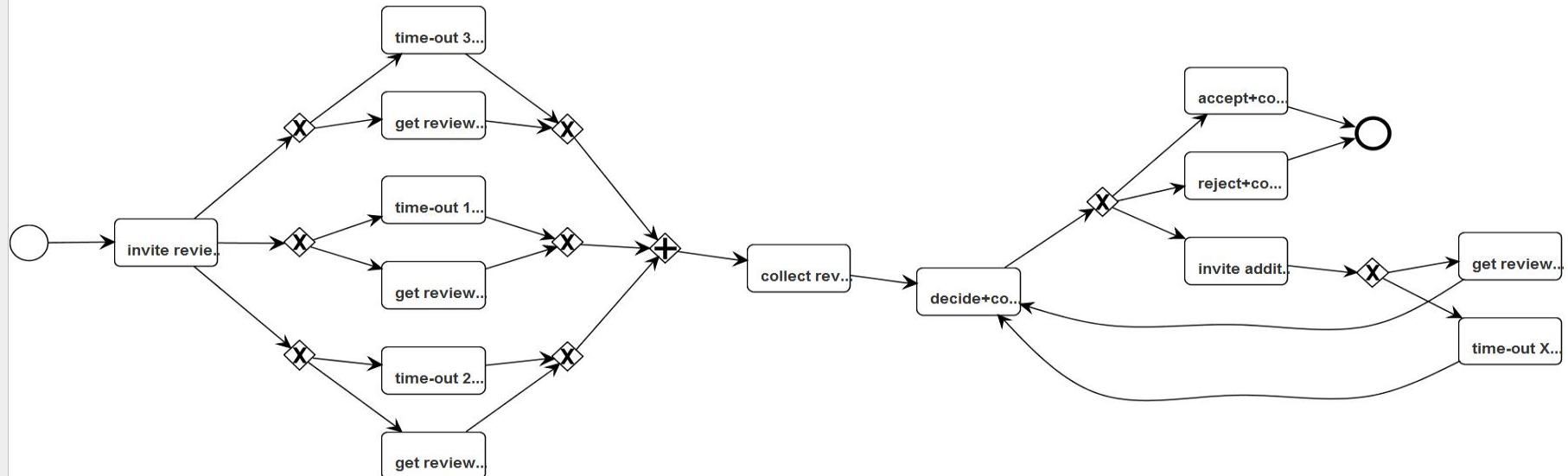
PIP

Zoom

Discovered using Alpha algorithm ...



... visualized as a BPMN diagram!



visualization of discovered model
≠
representational bias

More important than visualization ...

Why is process discovery so difficult?

- There are **no negative examples** (i.e., a log shows what has happened but does not show what could not happen).
- Due to concurrency, loops, and choices the **search space has a complex structure** and the log typically contains only a **fraction** of all possible behaviors.
- There is **no clear relation** between the size of a model and its behavior (i.e., a smaller model may generate more or less behavior although classical analysis and evaluation methods typically assume some monotonicity property).
- **Careful consideration of representational bias is needed !!**

Part I: Preliminaries

Chapter 1

Introduction

Chapter 2

Process Modeling and Analysis

Chapter 3

Data Mining

Part II: From Event Logs to Process Models

Chapter 4

Getting the Data

Chapter 5

Process Discovery: An Introduction

Chapter 6

Advanced Process Discovery Techniques



Part III: Beyond Process Discovery

Chapter 7

Conformance Checking

Chapter 8

Mining Additional Perspectives

Chapter 9

Operational Support

Part IV: Putting Process Mining to Work

Chapter 10

Tool Support

Chapter 11

Analyzing “Lasagna Processes”

Chapter 12

Analyzing “Spaghetti Processes”

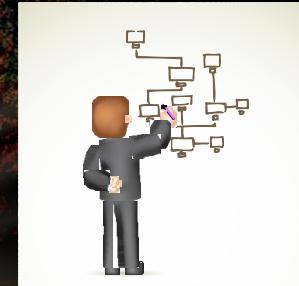
Part V: Reflection

Chapter 13

Cartography and Navigation

Chapter 14

Epilogue



Process Mining

Discovery, Conformance and
Enhancement of Business Processes

Springer