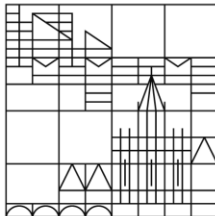


# Unifying Change Towards a Framework for Detecting the Unexpected

Iris Adä and Michael R. Berthold  
Universität Konstanz

Universität  
Konstanz



NYCOMED CHAIR FOR BIOINFORMATICS  
AND INFORMATION MINING

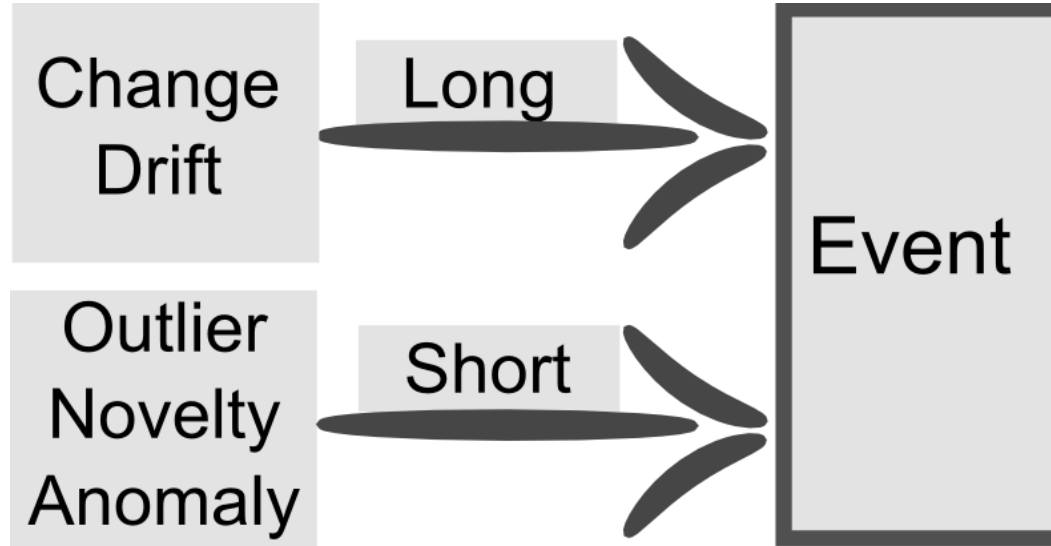


1. Introduction
2. Event type overview
3. Framework
4. Example



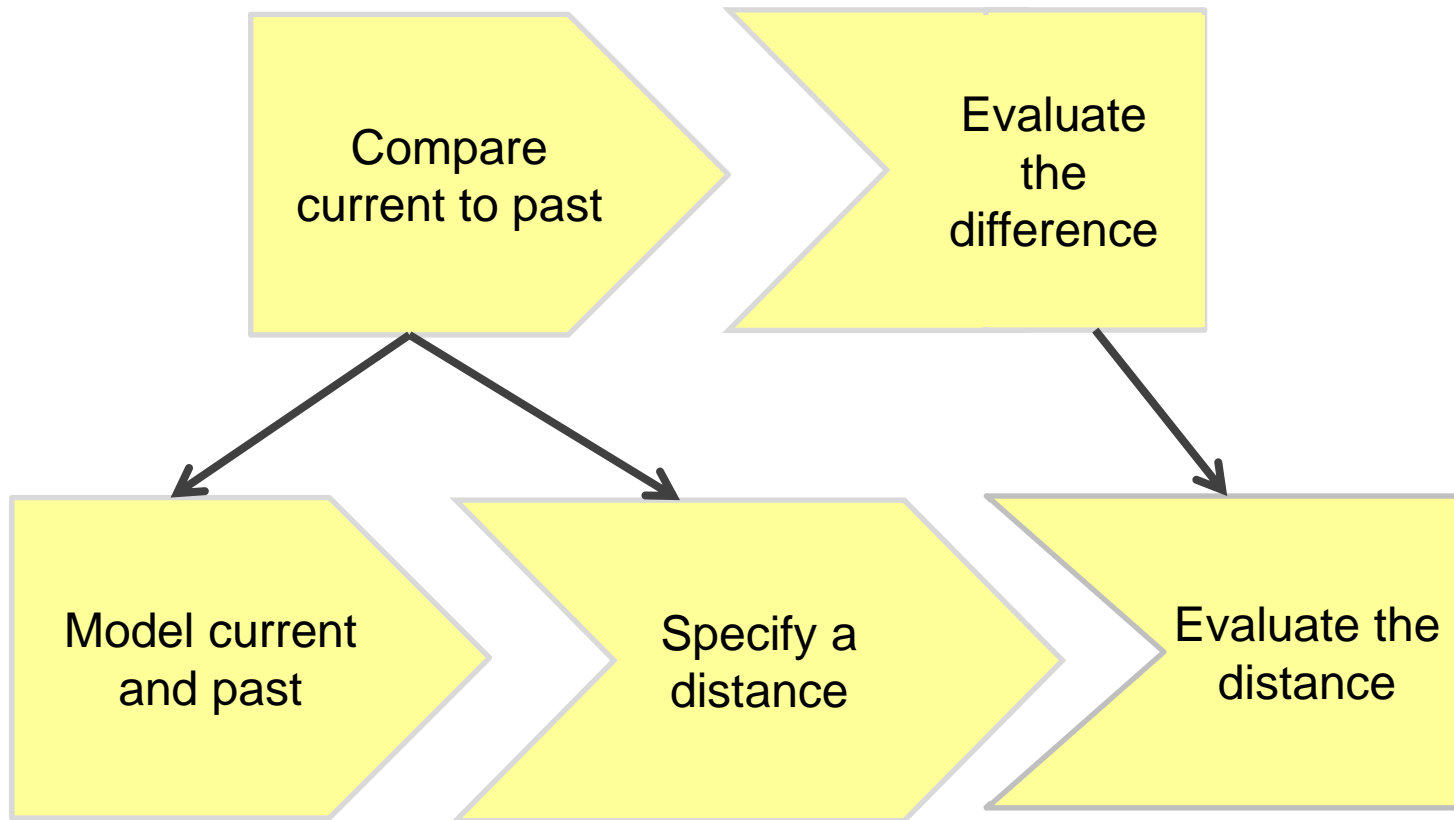
# Introduction

- Event detection is interested in finding *unexpected* behavior
  - The data changes
  - The data shows an unknown behavior
  - There is a single unknown pattern
  - There are a collection of unknown patterns



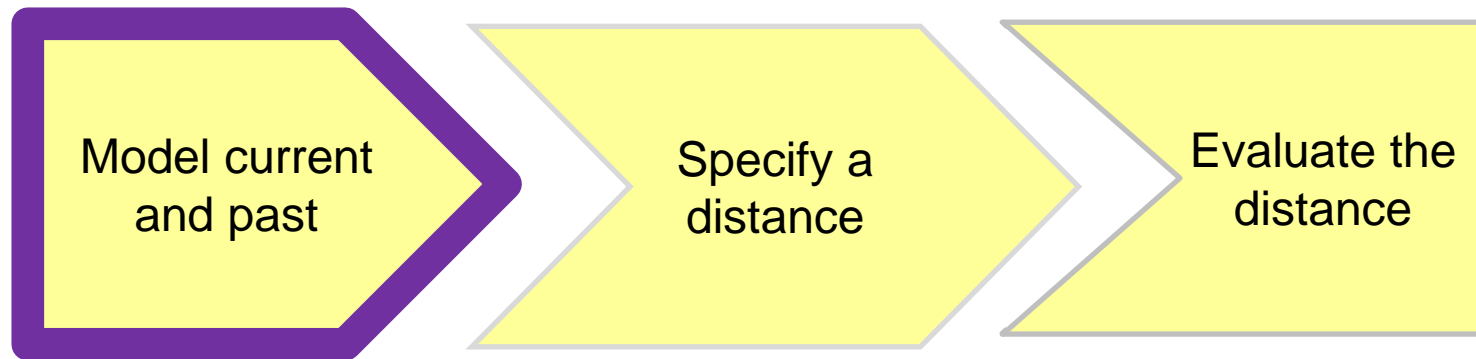


# Framework overview





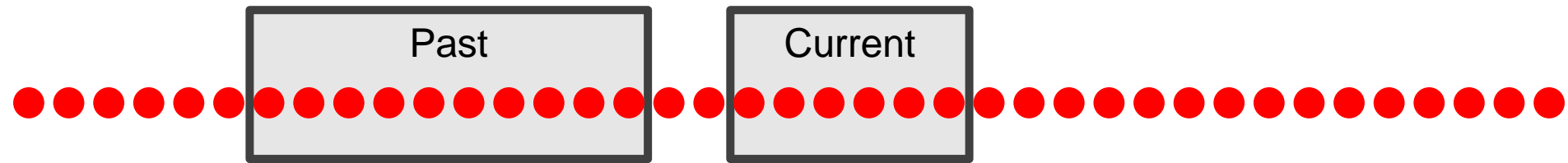
# Framework overview





# Compare current to past

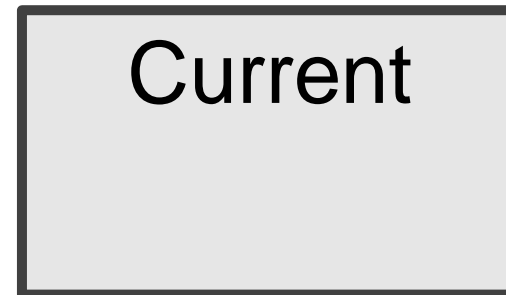
- Current : Includes *recent* data points
- Past : Includes *older* data points





# Find a window definition

1. The relation of the windows
  1. Are the windows overlapping?
  2. What is the distance between the windows
2. The internal configuration for the windows
  1. Size of the window
  2. Is the window evolving over time?





# Window definition

Simplified to :

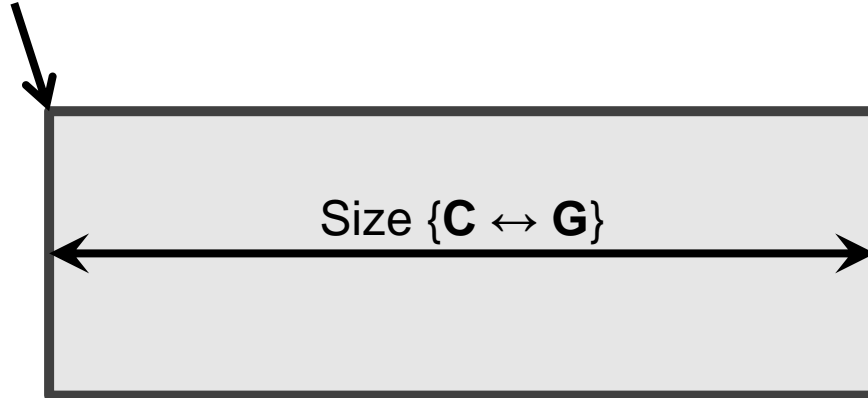
First point of the window

- **Sliding**  $\leftrightarrow$  **Fixed**

Size of the window

- **Constant**  $\leftrightarrow$  **Growing**

Start {**S**  $\leftrightarrow$  **F**}



Start {**S**  $\leftrightarrow$  **F**}



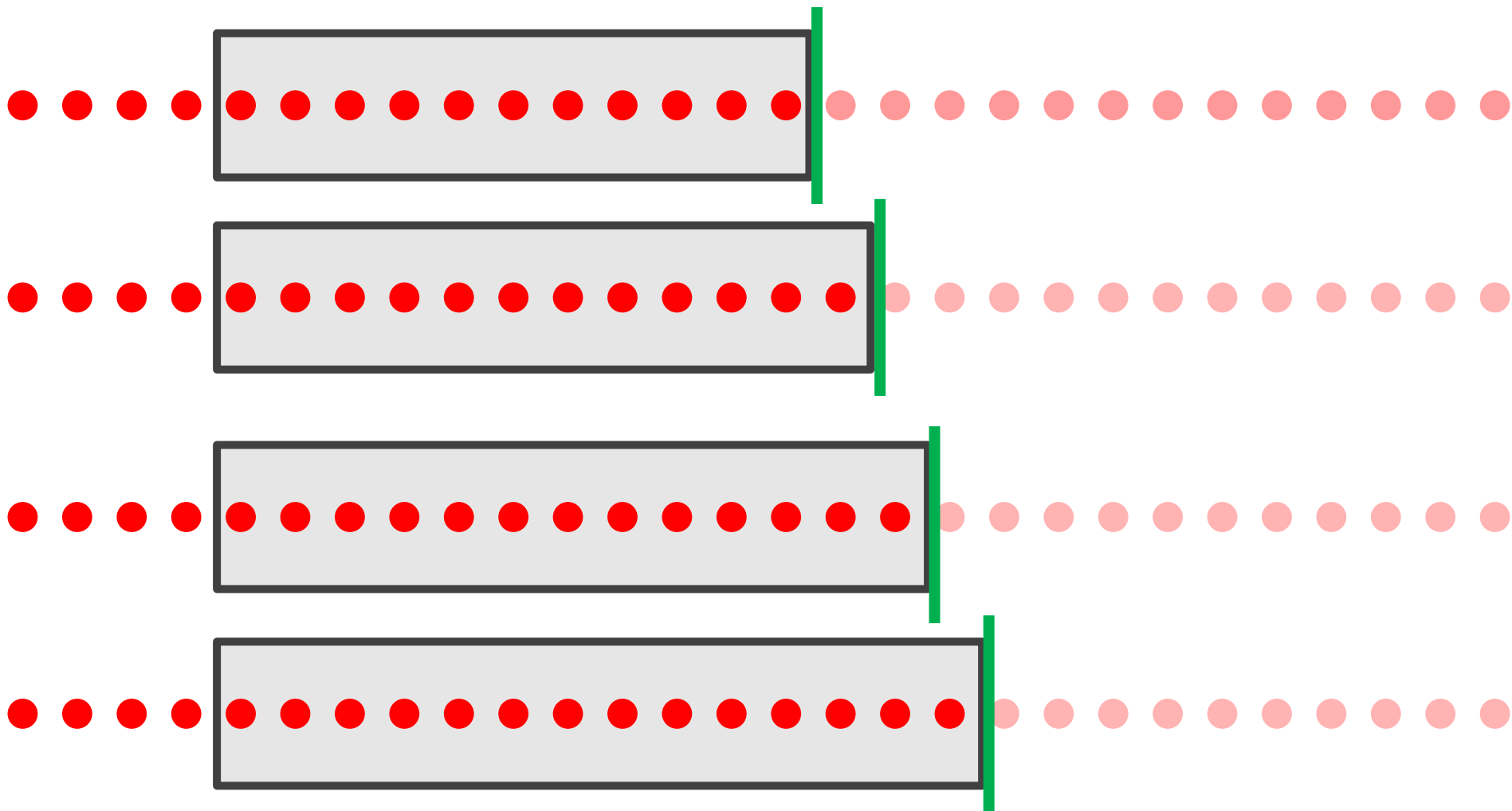




# Window types : Fixed Growing

FG

- Constant start position over the data stream
- Growing size (e.g. incorporates one point each time step)



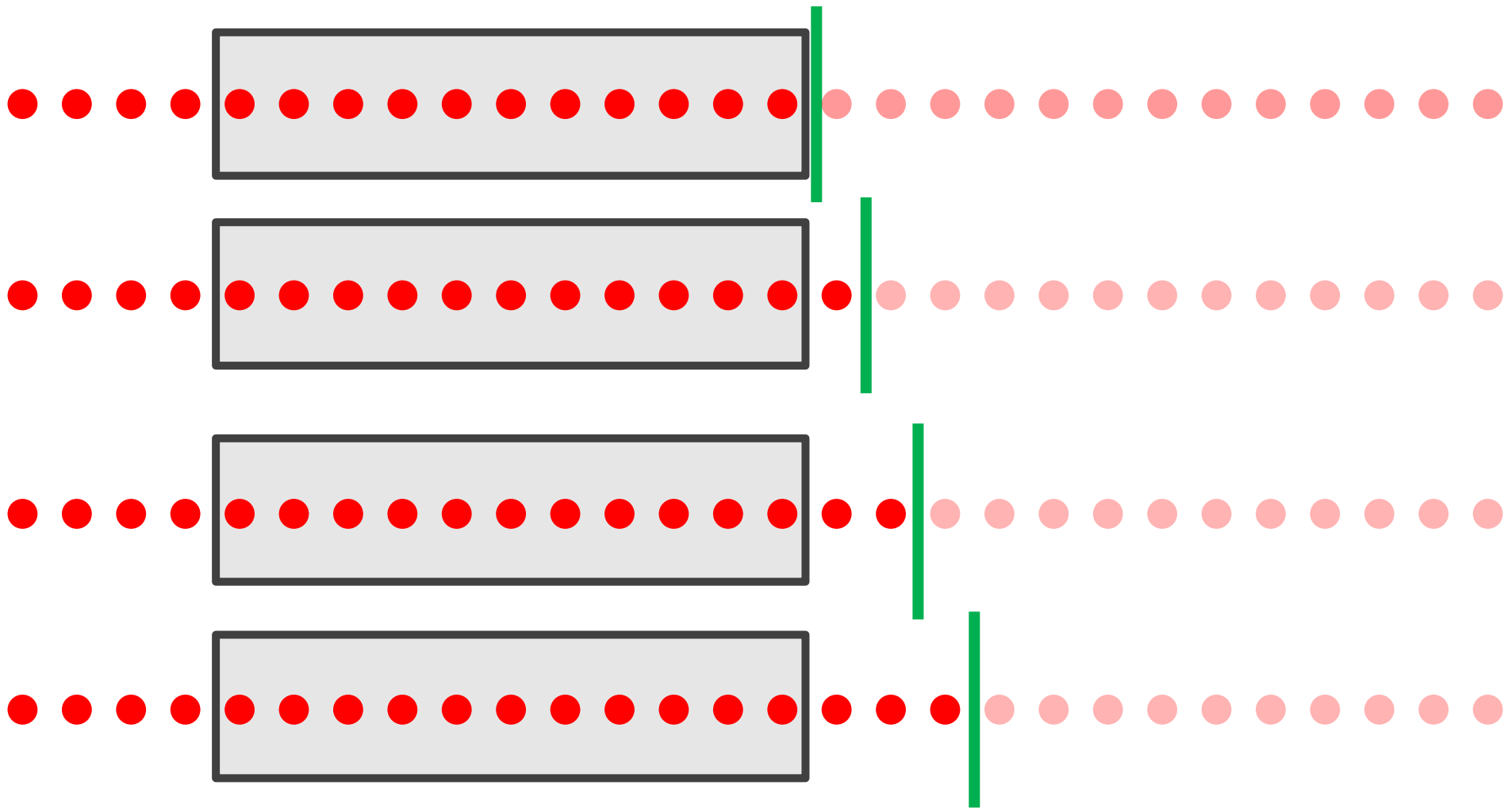


# Window types : Fixed Constant

FC

Start position and size are not changing

→ Window is not changing

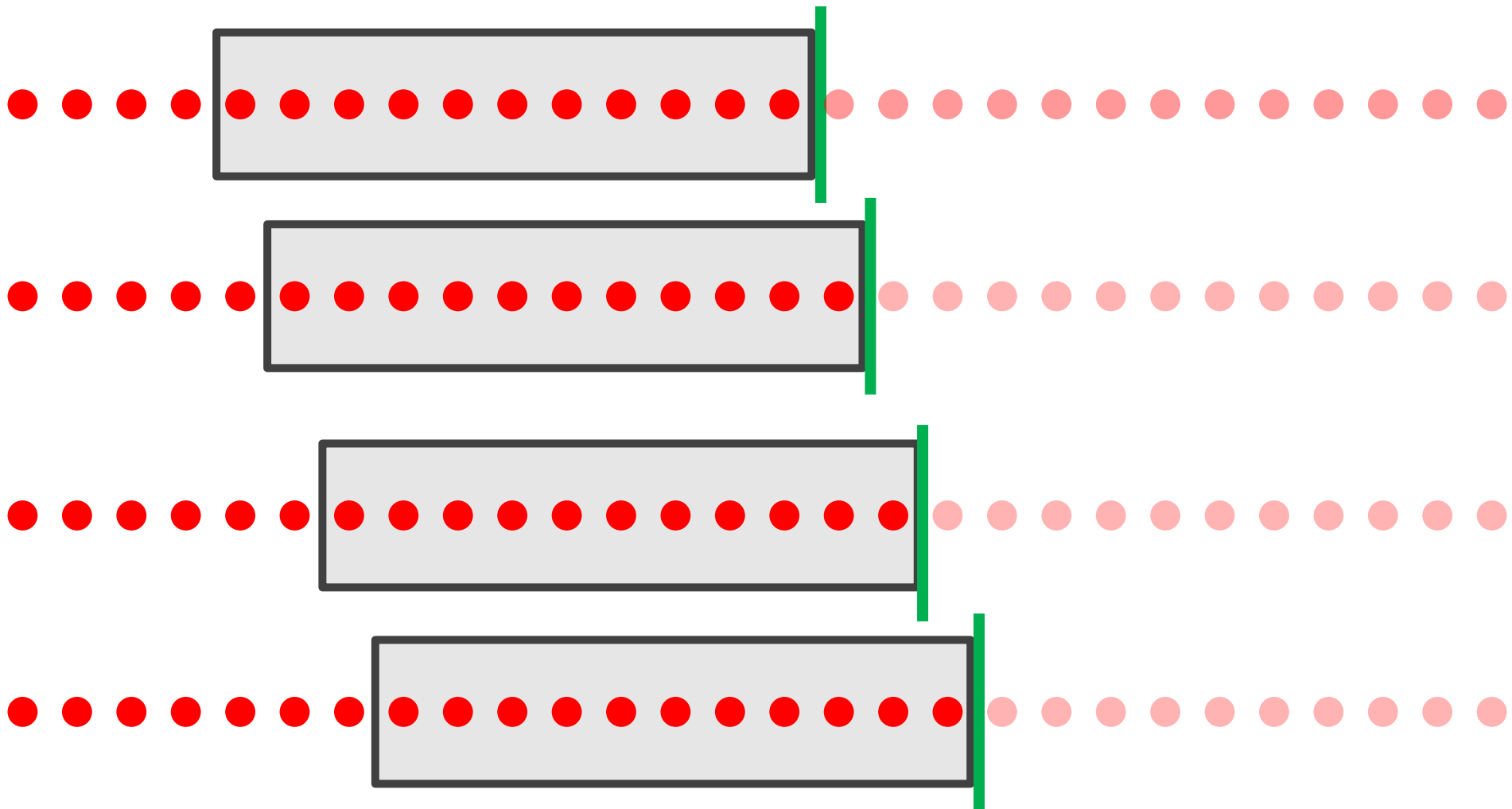




# Window types : Sliding Constant

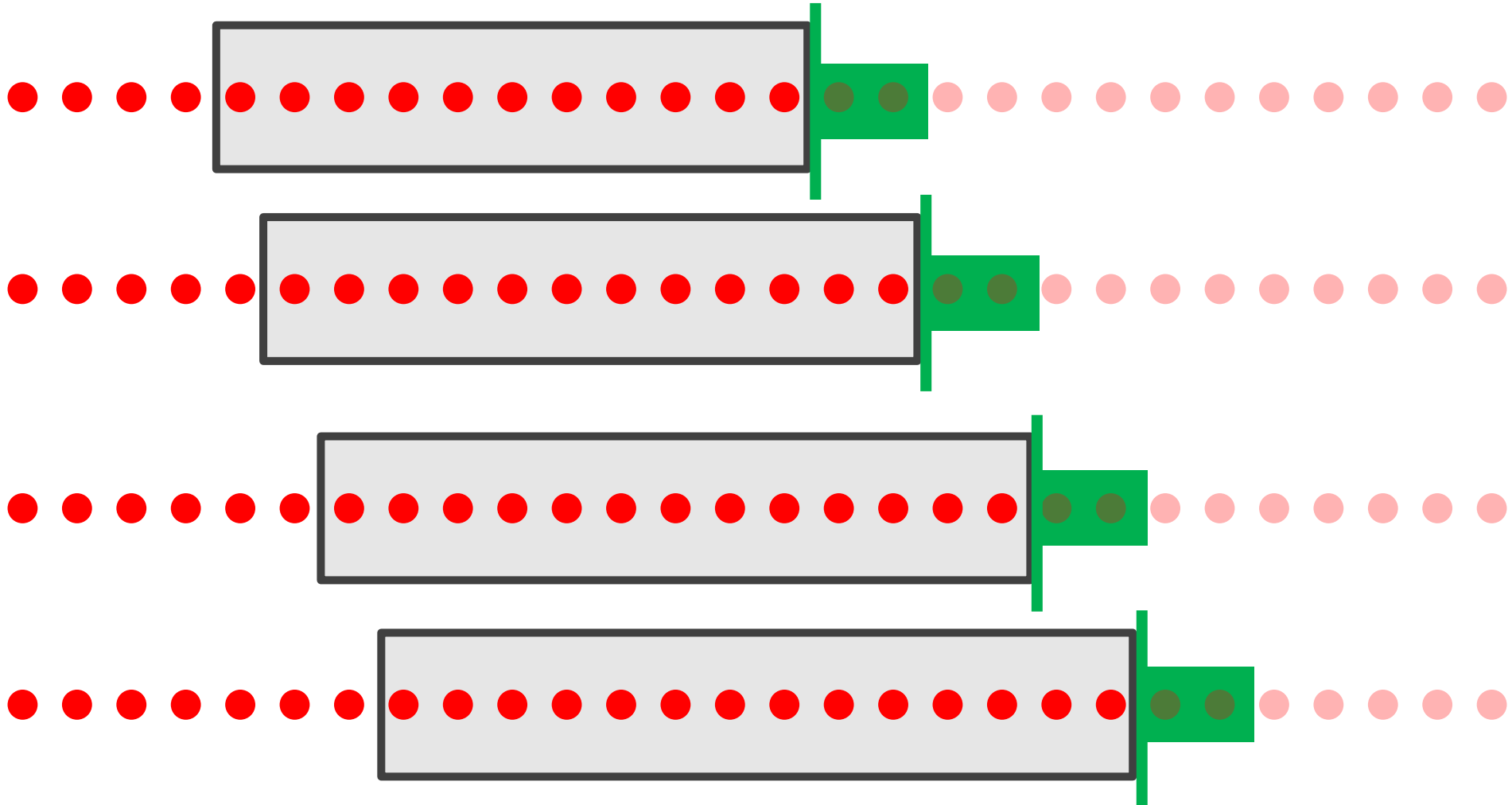
SC

- Sliding start (moves one point each time step)
- Constant size





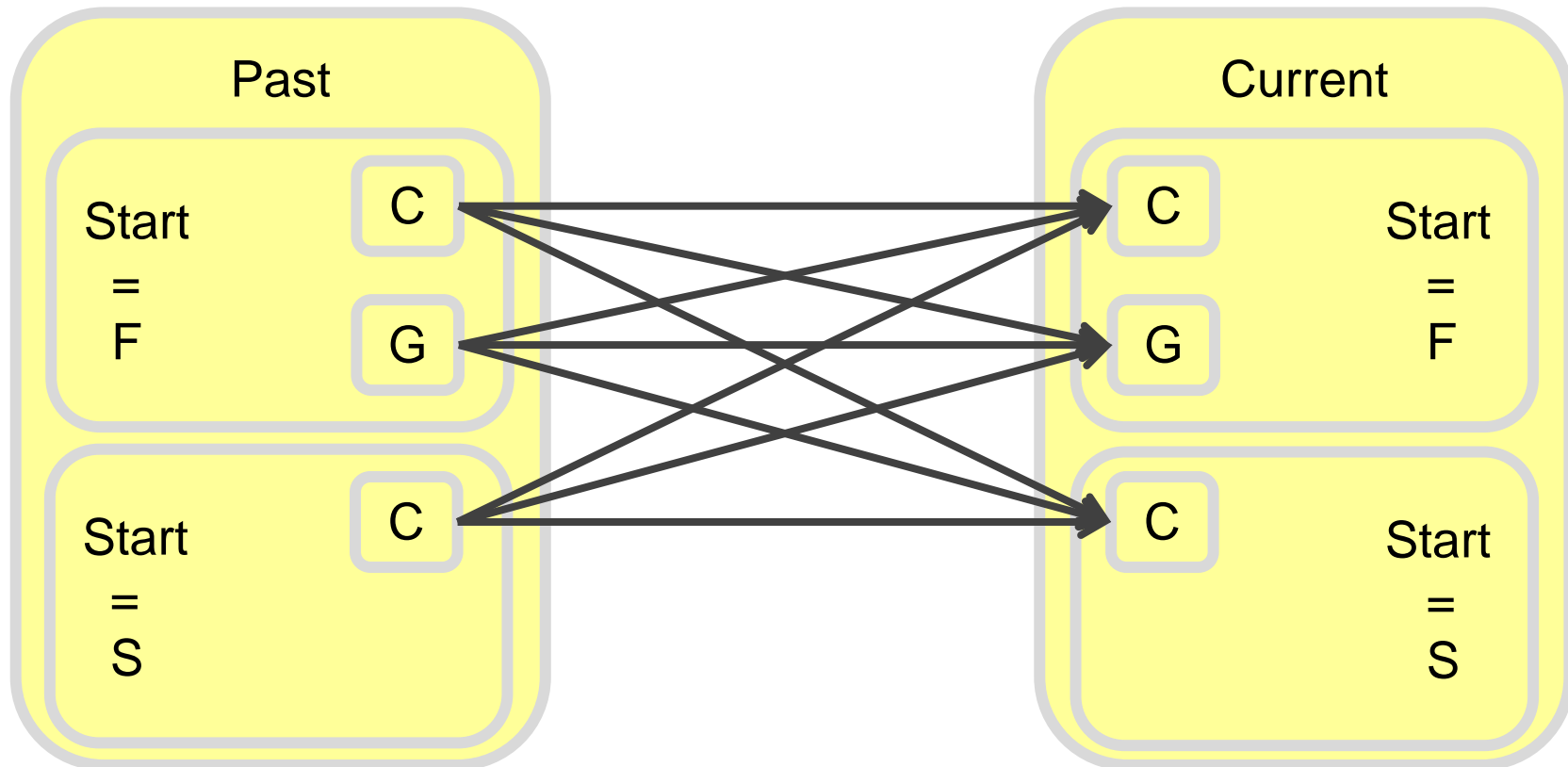
- Grows and slides at the same time
  - Open:





# Window combination

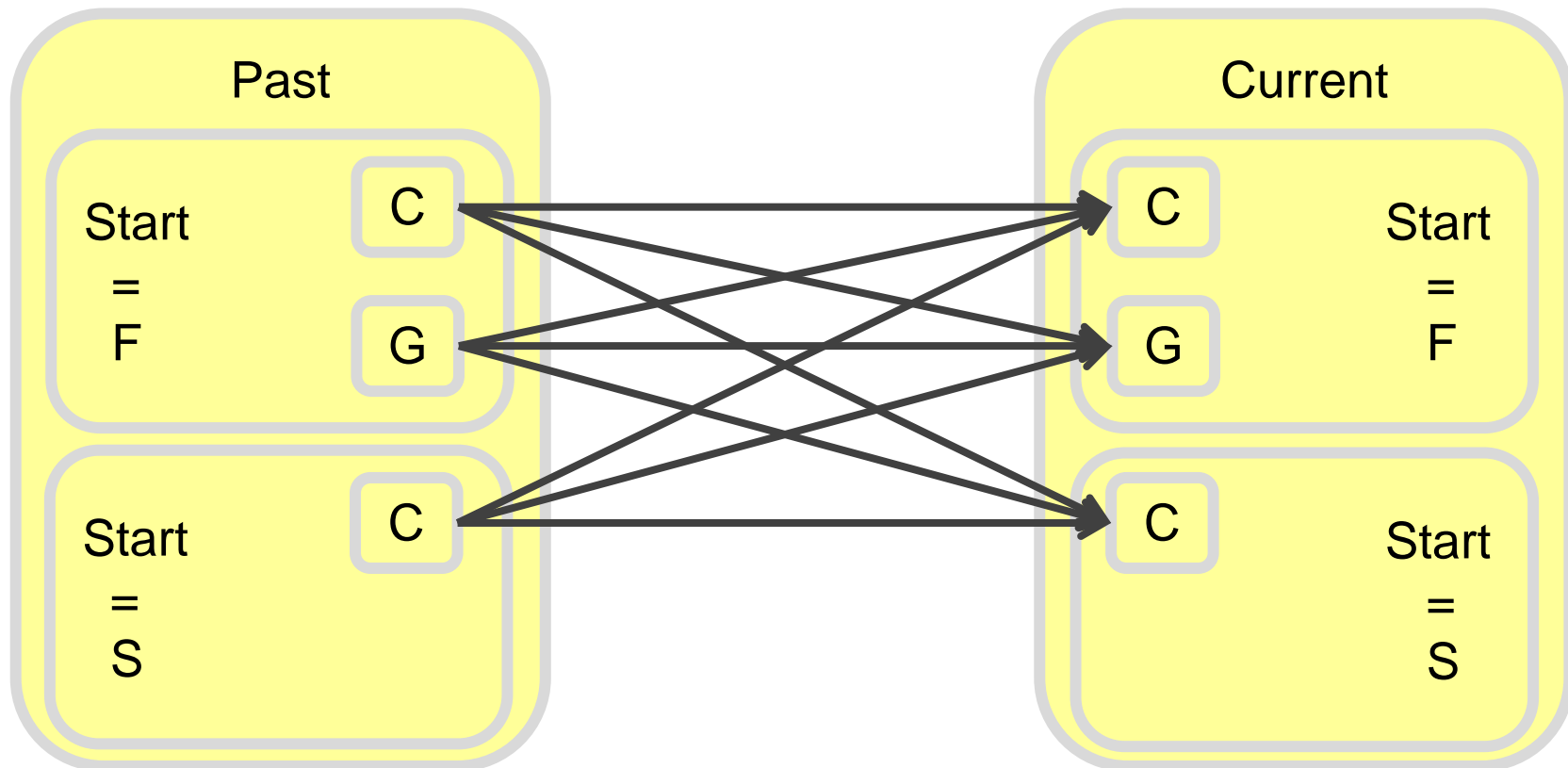
- 9 possible combinations
- Sliding and Growing is not considered





# Filtering

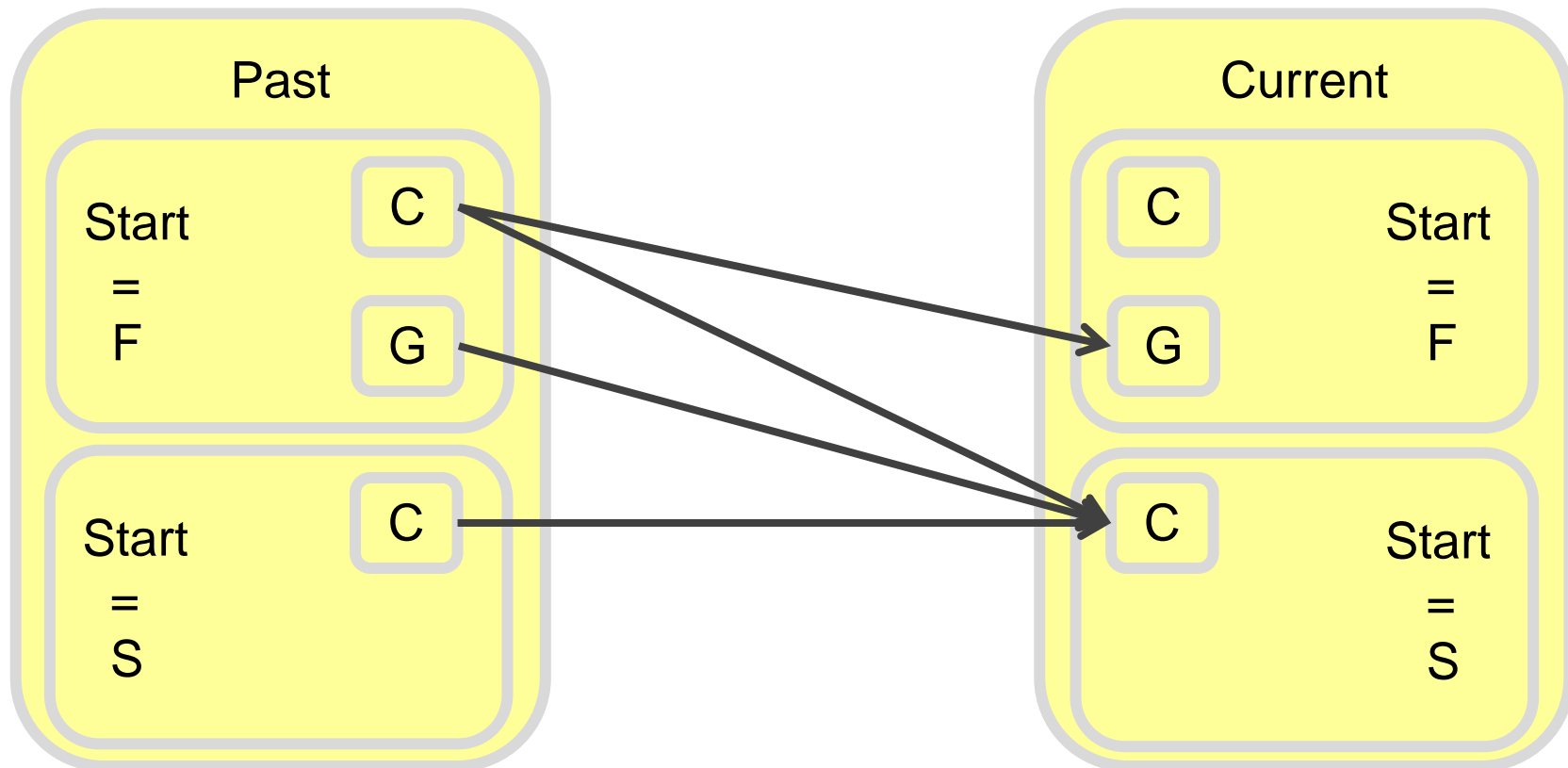
- Combination with no changes  $FC \rightarrow FC$
- Combinations in which the past window overtake the current
  - $\rightarrow FG \rightarrow FC$
  - $\rightarrow$  Similar :  $SC \rightarrow FC$





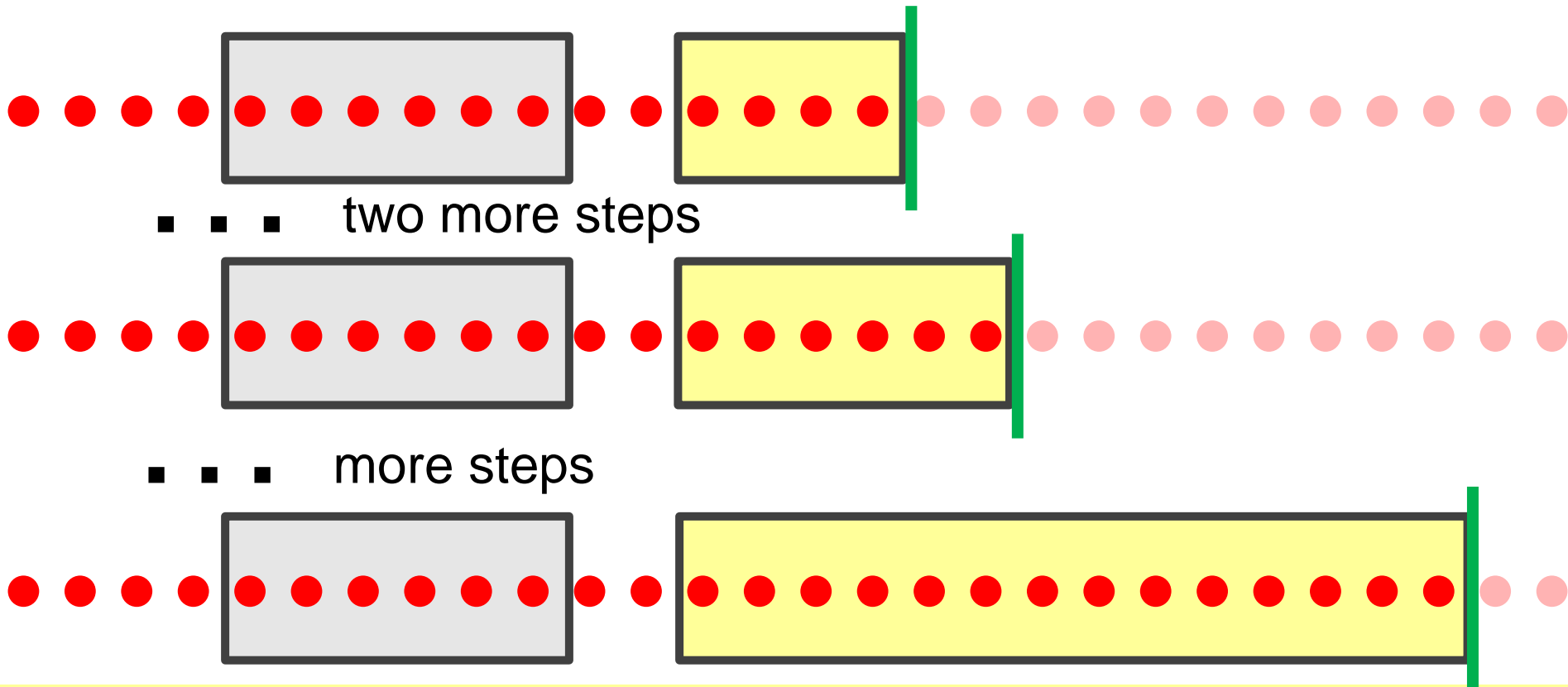
# Window combination

- Results in four combinations
  - For the current window only changing windows
  - The past can be modeled constant as well





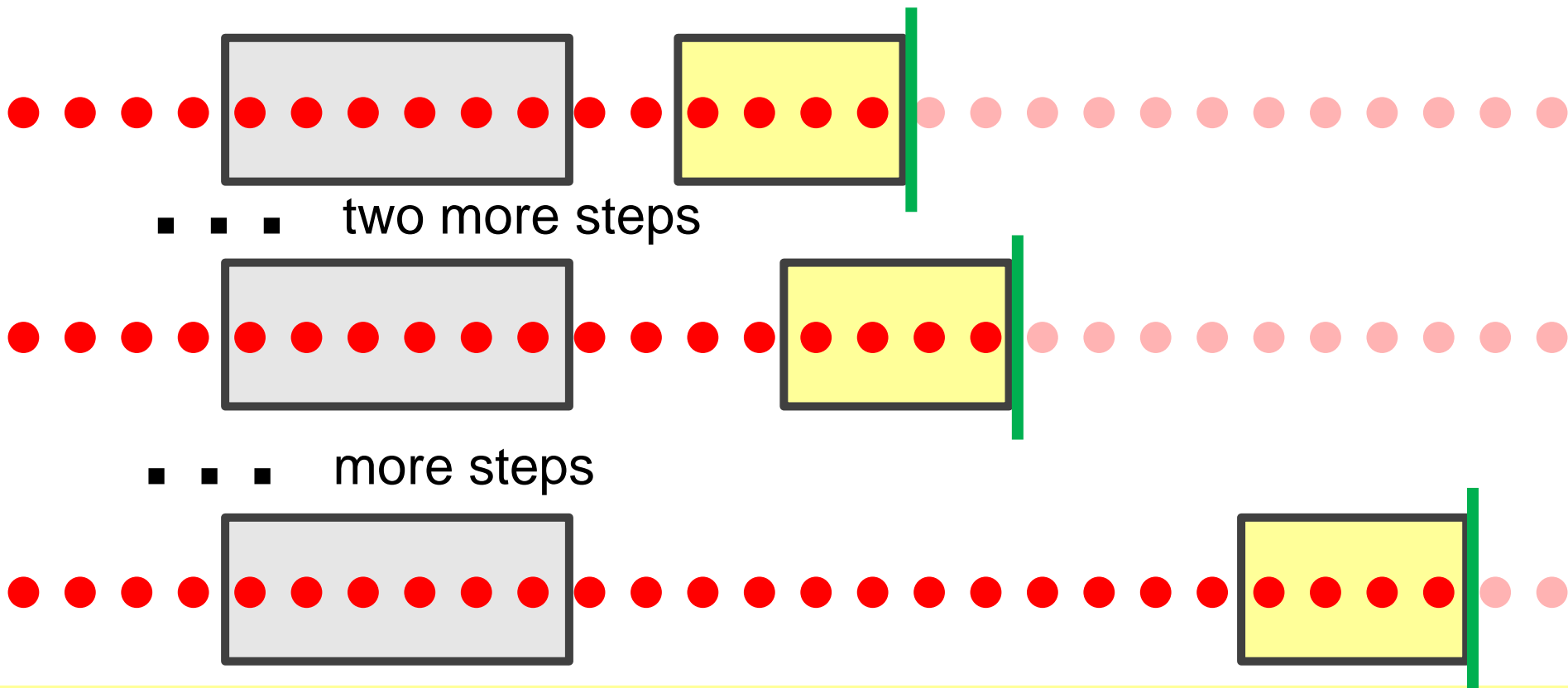
- Simulates a constant past behavior
- Incorporates all new data points into the current behavior
- E.g. change detection





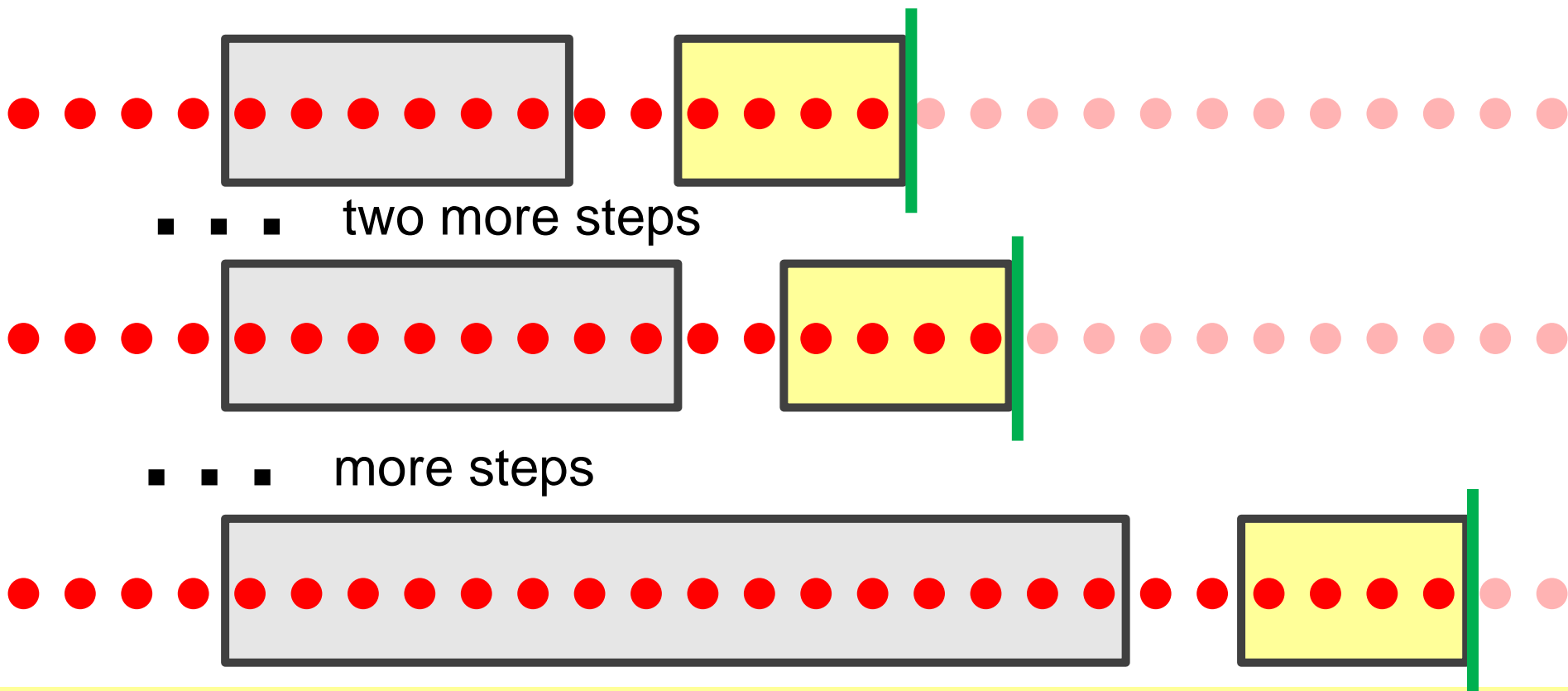


- Simulates a constant past behavior
- Slides a window at the end of the stream for current
- E.g. outlier detection without acceptance of changes



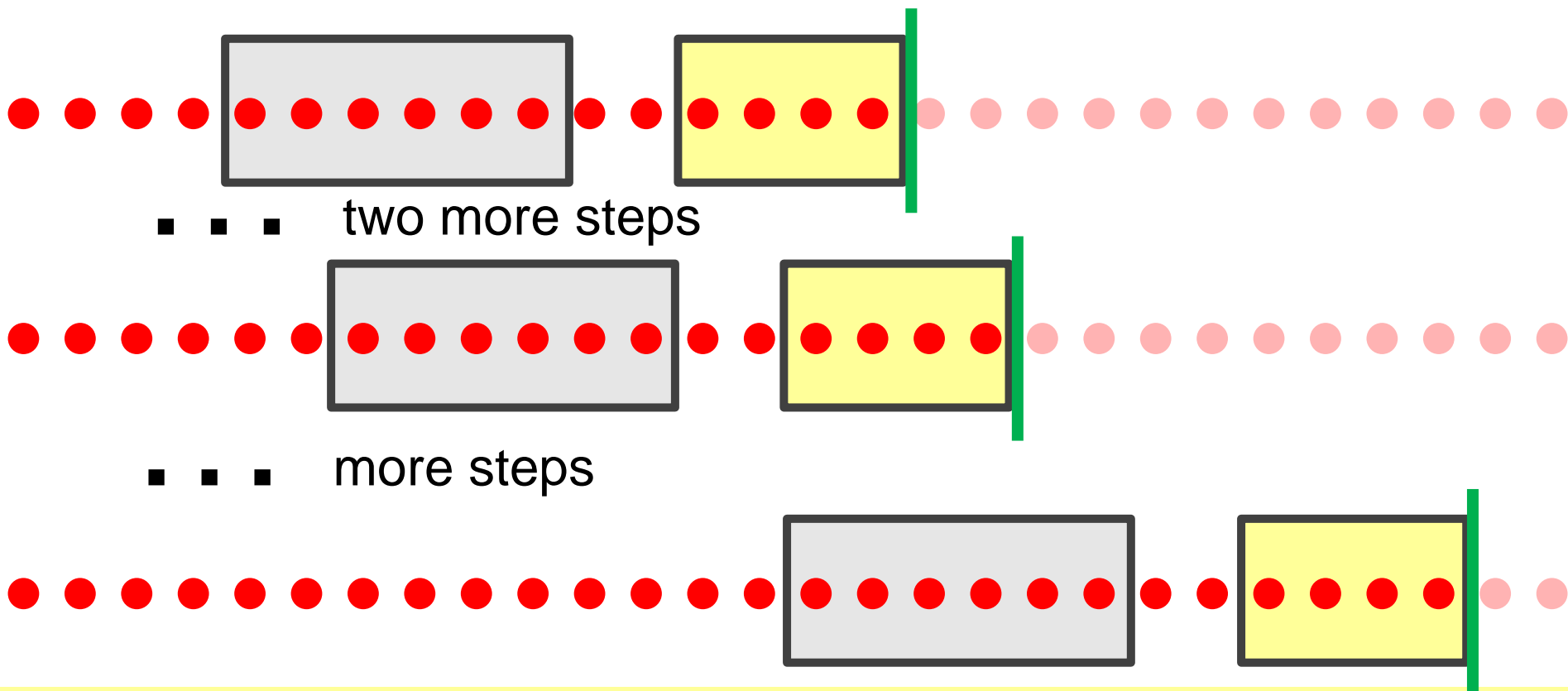


- Growth the past at each time step
- Slides a window at the end of the stream for current



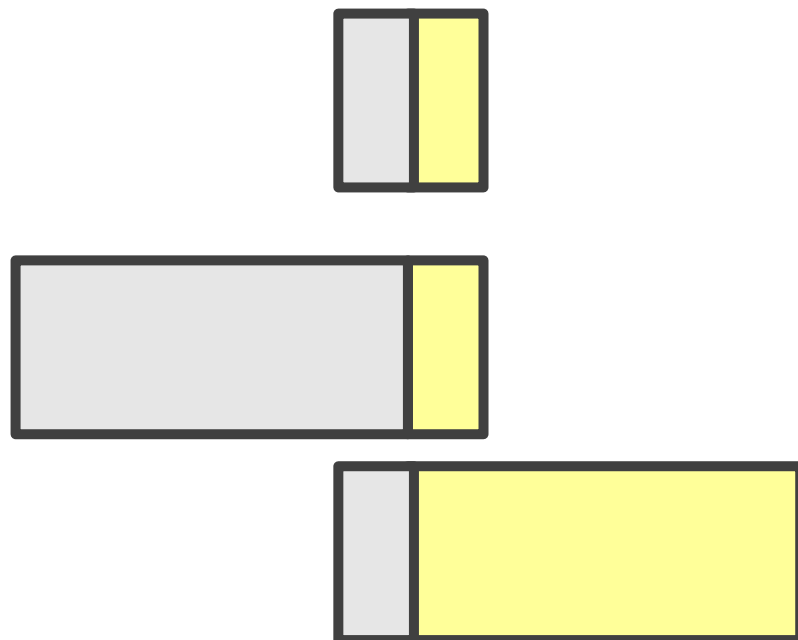
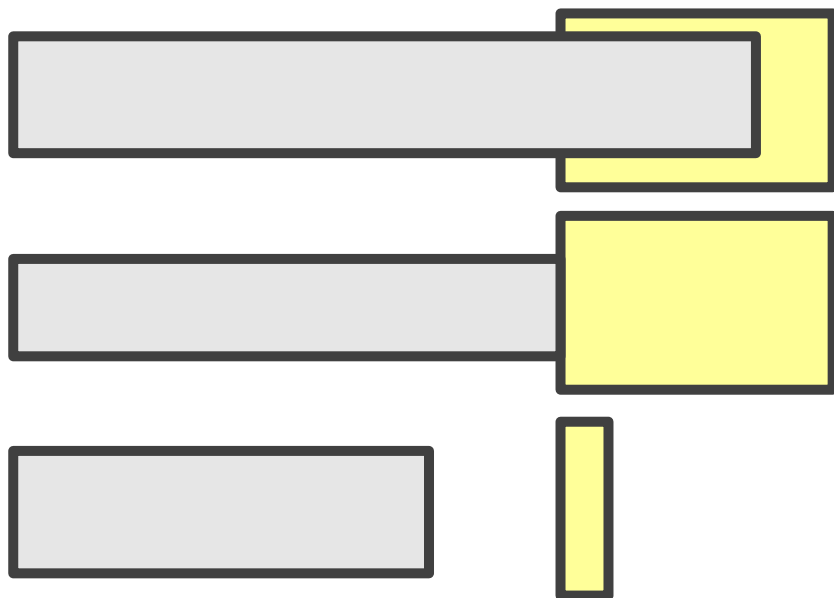


- Growth the past at each time step
- Slides a window at the end of the stream for current
- Adjusts to changes over time



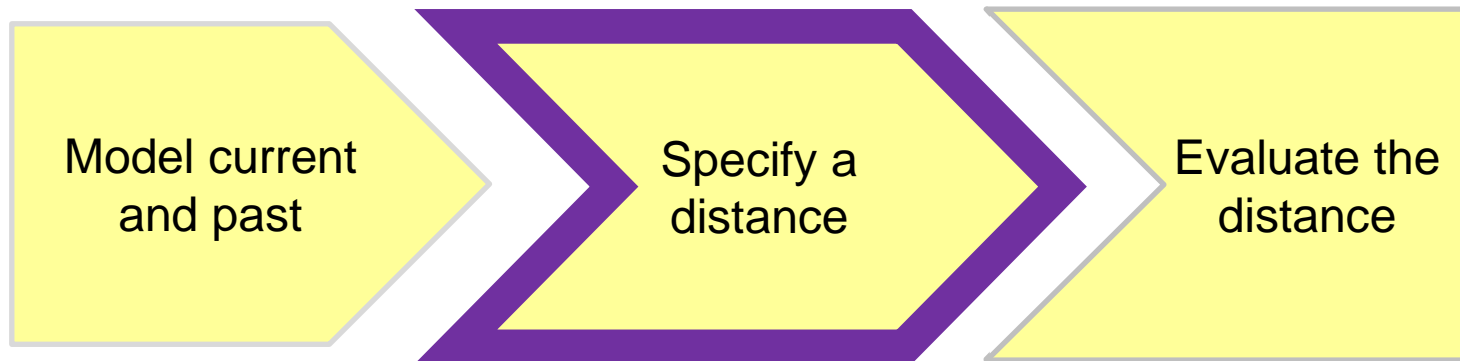


- window sizes,
- distances between windows,
- if windows are overlapping
  - is based on initialization





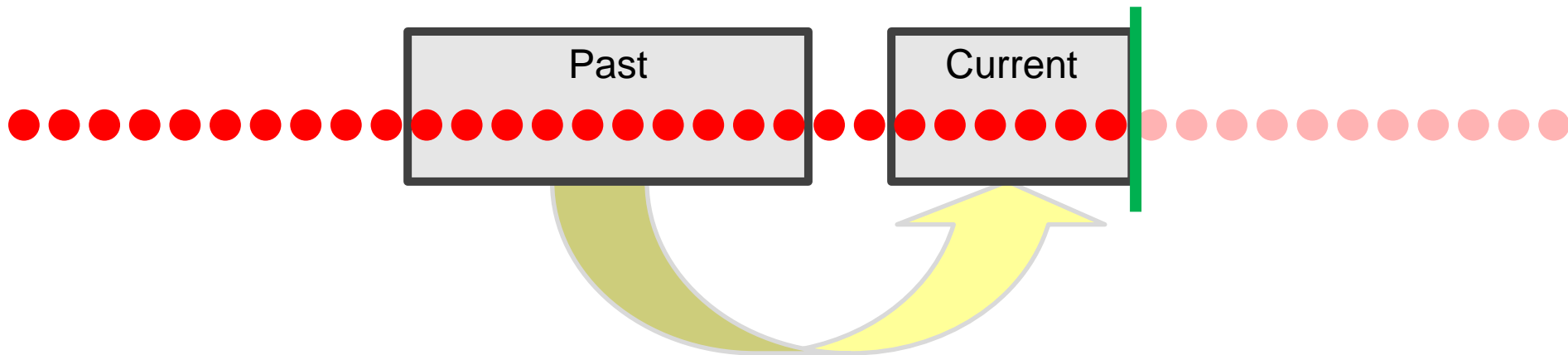
# Framework overview



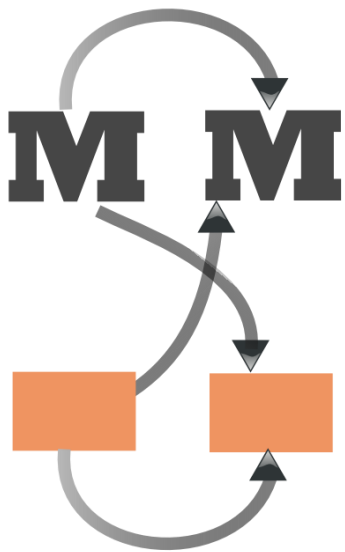


# Distance selection

- How can dissimilarity be measured ?



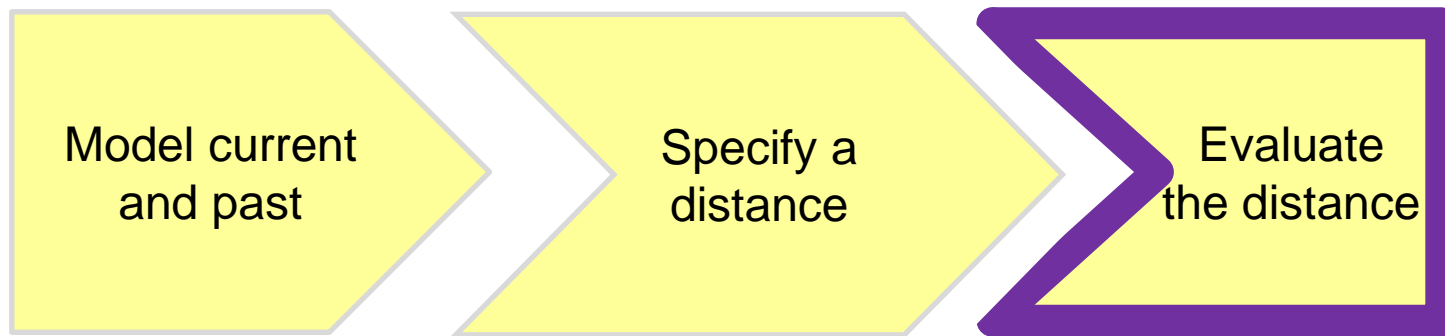
- How can dissimilarity be measured ?



- Compare model with model
- Compare model and data
- Compare data with data



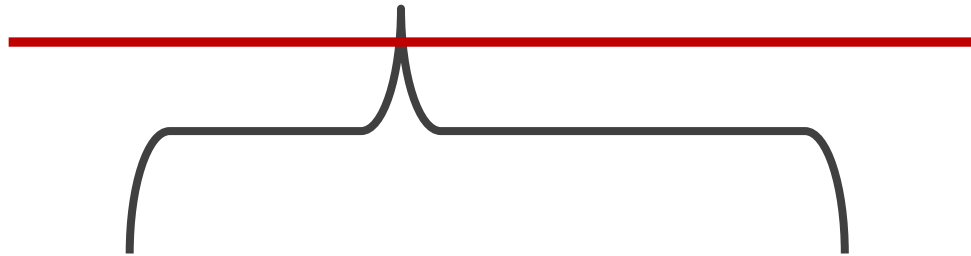
# Framework overview





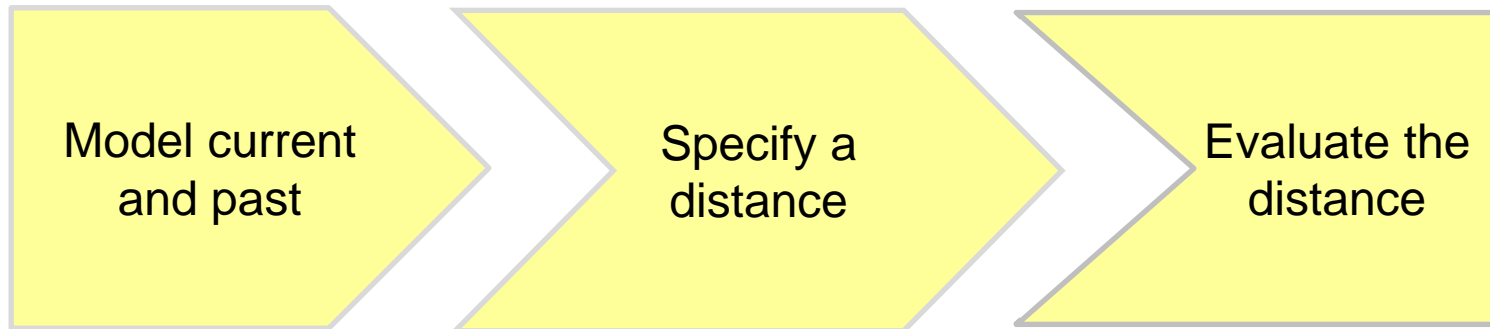
- Until now :  
(high dimensional) stream  $\Rightarrow$  one dimensional error measure

e.g.



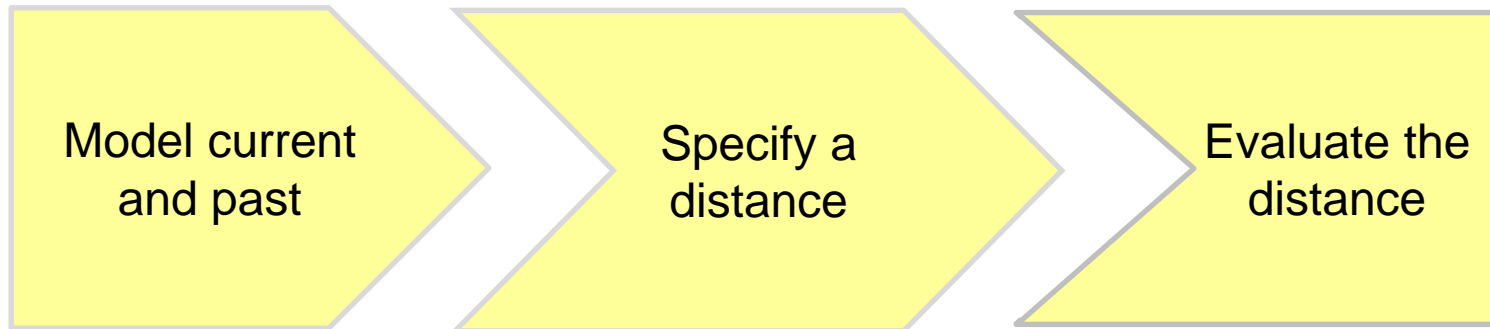
- How is a potential event reported to the user?
  - most : use a threshold or adaptations of it
  - Other : adapt drift detection methods for more flexibility





## Anomaly detection

1. FG -> SC
2. Model to Data  
(Cluster model, evaluation on nearest representative)
3. Threshold evaluation



## Change detection

1. SC -> SC
2. Model to Model  
(Mixture of Gaussians, Kullback-Leibler)
3. Derivation evaluation



- Event detection can be seen as the combination of multiple detection methods
- A framework for
  - Summarizing many existing approaches
  - Motivating new approaches
- The framework consists of three steps
  - Formulate window configuration
  - Model a distance function between windows
  - Define the evaluation