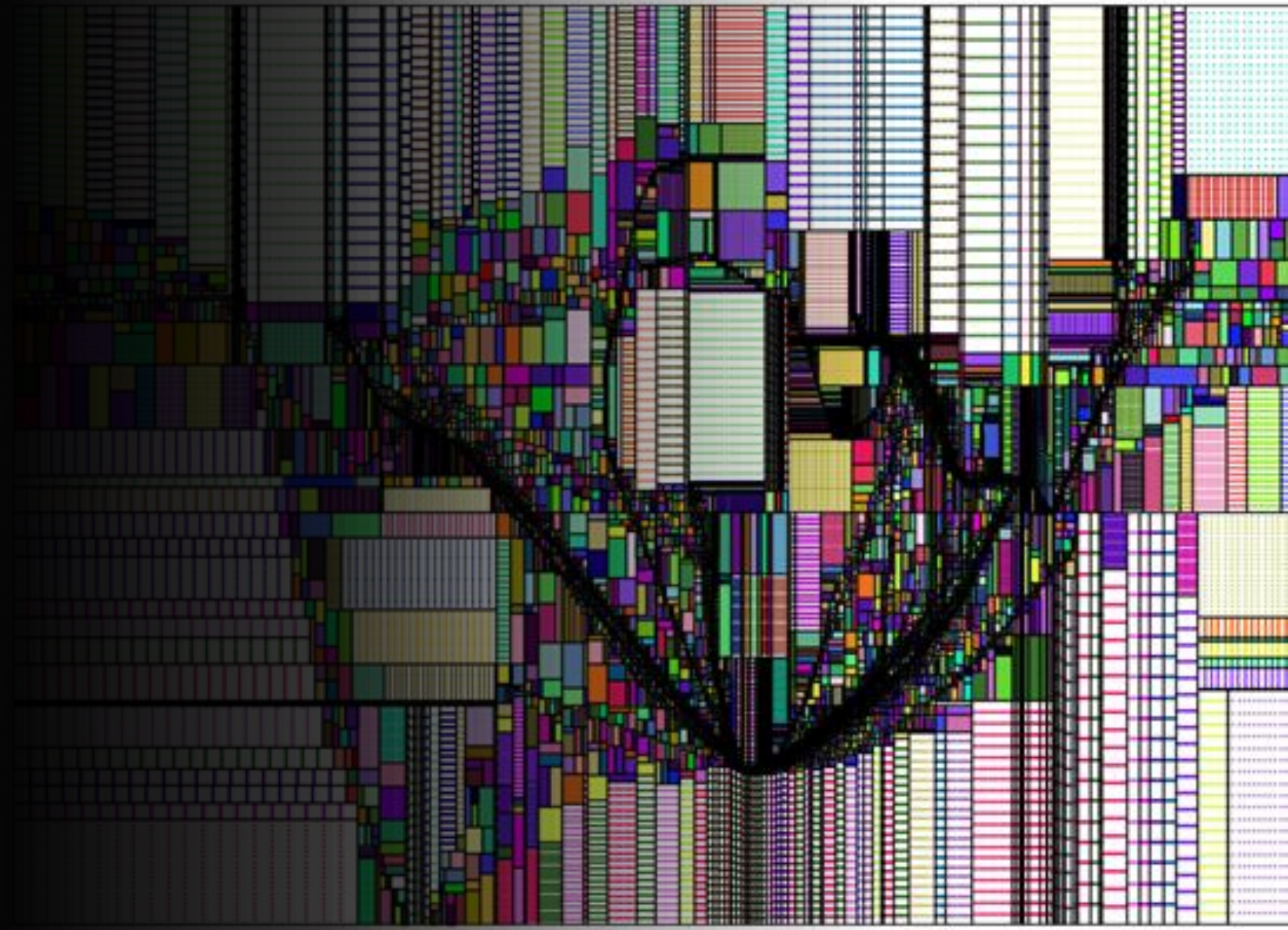
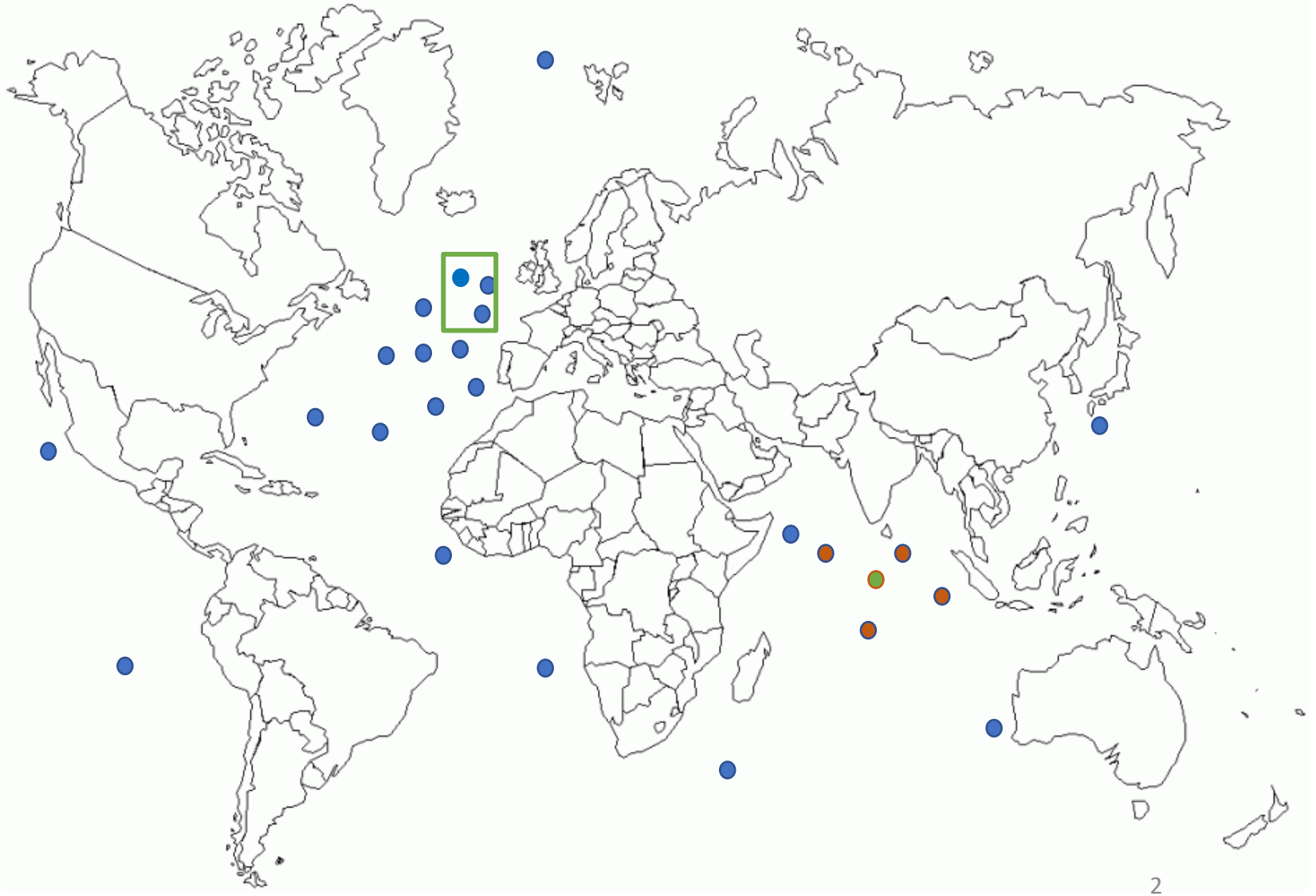
# Learned Spatial Indexes



# Spatial queries

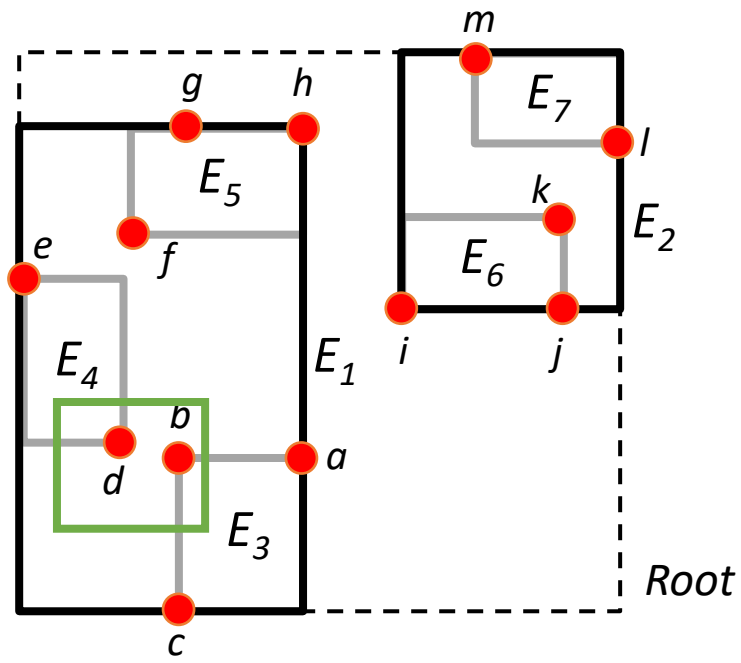
Given a database of objects with location information, spatial queries retrieve objects with respect to their location (and possibly other attributes)

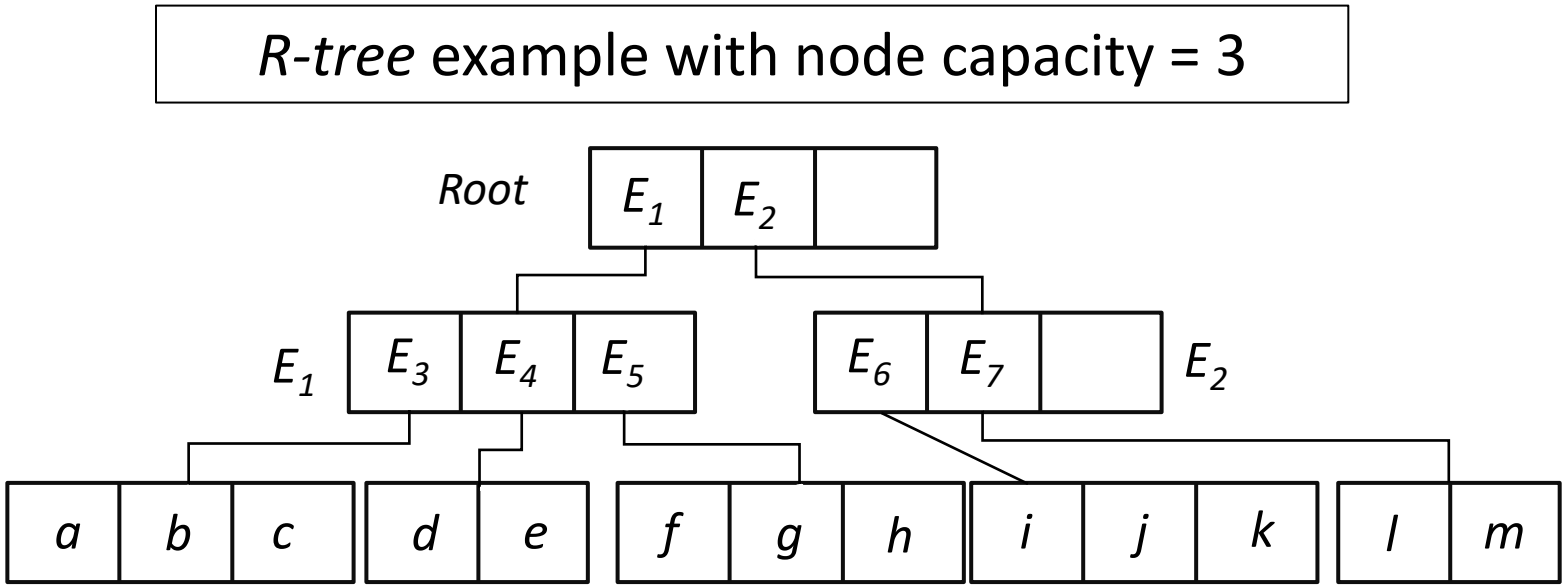
- *Range queries*
- *k nearest neighbor queries*



# Spatial indexes

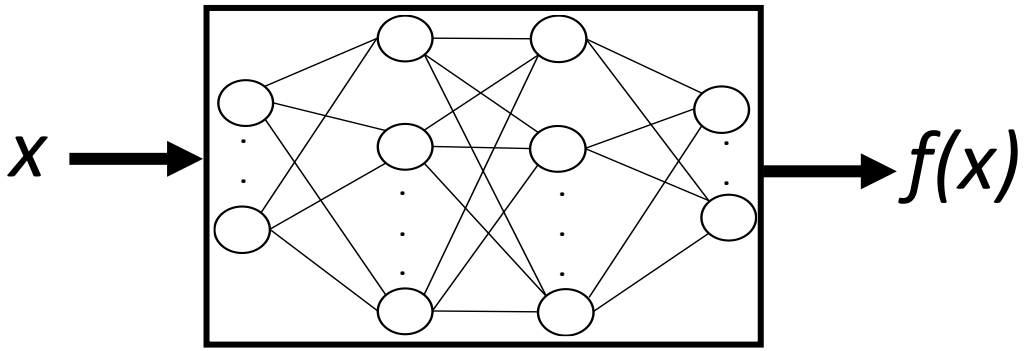
- Spatial indexes are used to organize database objects into nodes based on their location (so that objects nearby in space are stored in the same node)
- Most spatial indexes are balanced trees.
- Each intermediate node is associated with a bounding box, which contains all the child nodes bounding boxes.
- The leaf nodes contain pointers to the actual data.



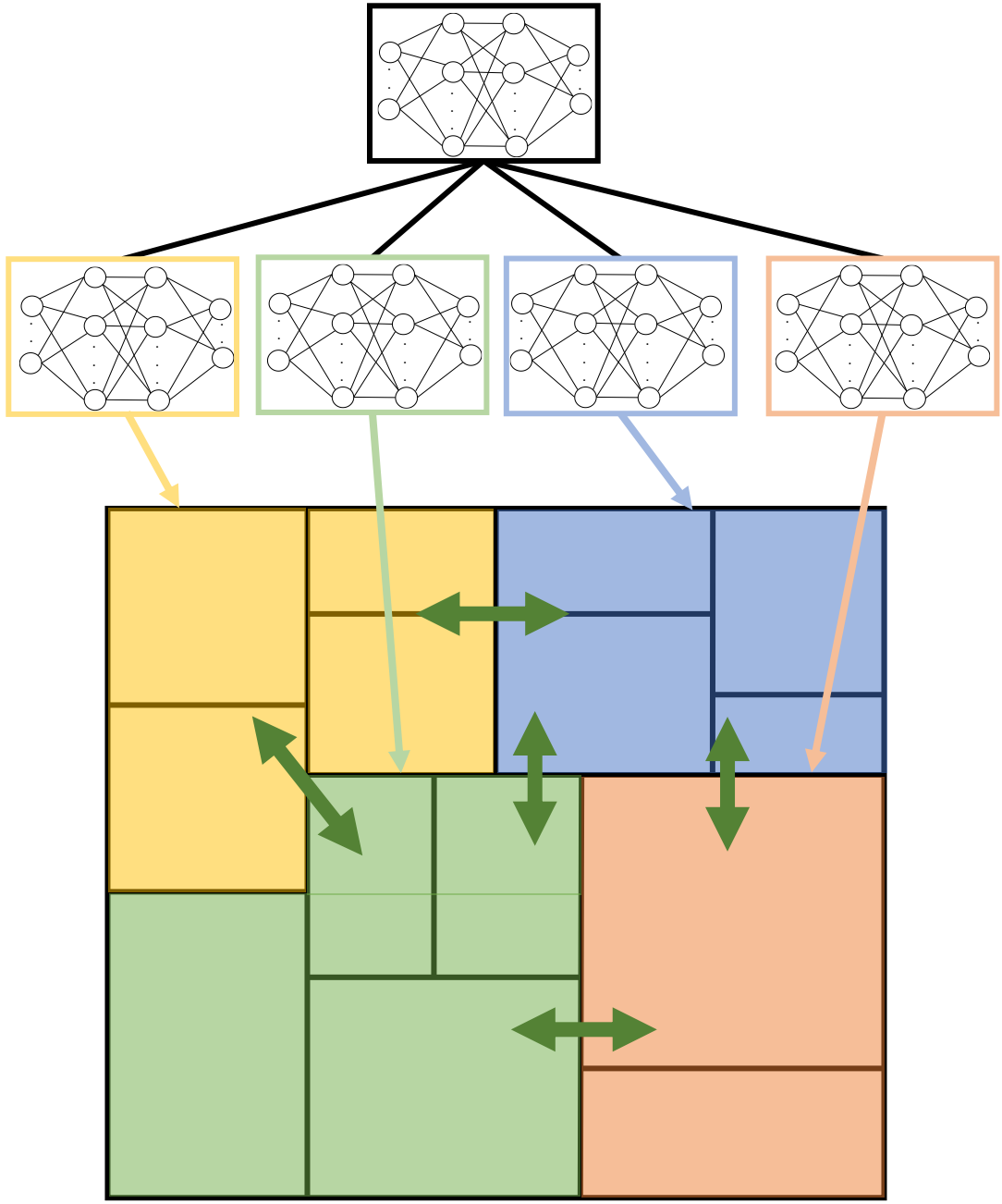
# Artificial Neural Networks (ANNs)

- ANNs are function estimators: Given an input *x*, they output *f(x)*.
- By given enough examples, ANNs are trained to discover patterns in the data by themselves.
- In our case, *x* is a location (co-ordinates) and *f(x)* is the node covering *x*.



# Learned Spatial Index

- Replace the intermediate nodes of a conventional spatial index with a hierarchy of models (ANNs).
- Each intermediate node/model is responsible for a sub-partition of space.
- Given a location *x*, the top layer of the learned index picks another model of the next level. The process continues until the lowest level.



# Advantages and Challenges of LSI

Advantages

- Fast query answering.
  - Replace costly intersection checks with cheaper ANN operations
- Small space consumption.
  - Instead of storing intermediate node extents, store ANN weights

Challenges

- ANNs are inherently inaccurate
- No 1D ordering of multi-dimensional data
- Training of the ANNs and handling of data distribution changes

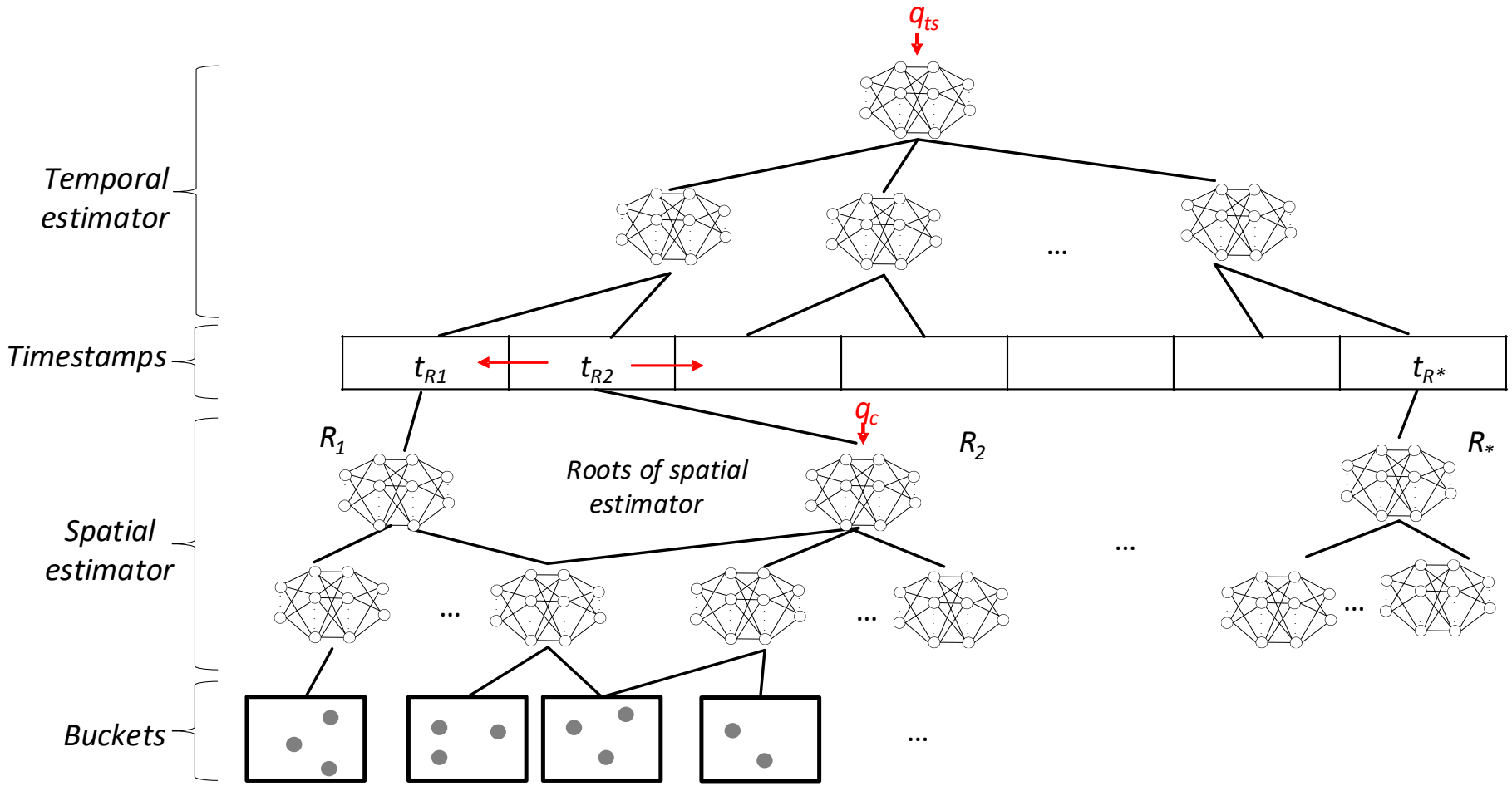
# LSI Components

1. *Partition scheme*: It divides the space in buckets necessary to construct the index.
2. *Estimator*: A hierarchy of ANNs or other hybrid models that returns a bucket covering the given query.
3. *Error fixing framework*: *Hop and Expand (HOPE).*
    - Hop enables the LSI to discover a bucket covering the query in the presence of ANN inaccuracy.
    - Expand discovers all buckets containing objects satisfying the query.
    - HOPE utilizes neighboring information at the leaf level.

# Extension to Historical LSI

- Conventional indexes store information and answer queries about the current time.
- Historical Indexes store all locations (trajectories) of moving objects in history.
- Queries refer to the past or the present:
  - Find all ships in Hong Kong post during the last 24 hours.
  - Which was the 5 nearest ships to a given ship on 1/1/2021.
- Each query $q$ has a temporal $q_{ts}$ and a spatial extent $q_c$.

# Architecture of Historical LSI



# Research Topics on LSI

- Partitioning schemes
  - Can we effectively replace conventional (e.g., geometric) partitioning with AI (e.g., reinforcement learning) methods?
- ANN architectures
  - Type of ANNs, e.g., Regression vs. Classification.
- Methods for handling inaccuracy
  - Trade-off of accuracy and query efficiency/index size
- Fast and effective retraining in the presence of changing distributions
- Extension to queries about the future given the current and past data
- Handling of location errors and missing information.