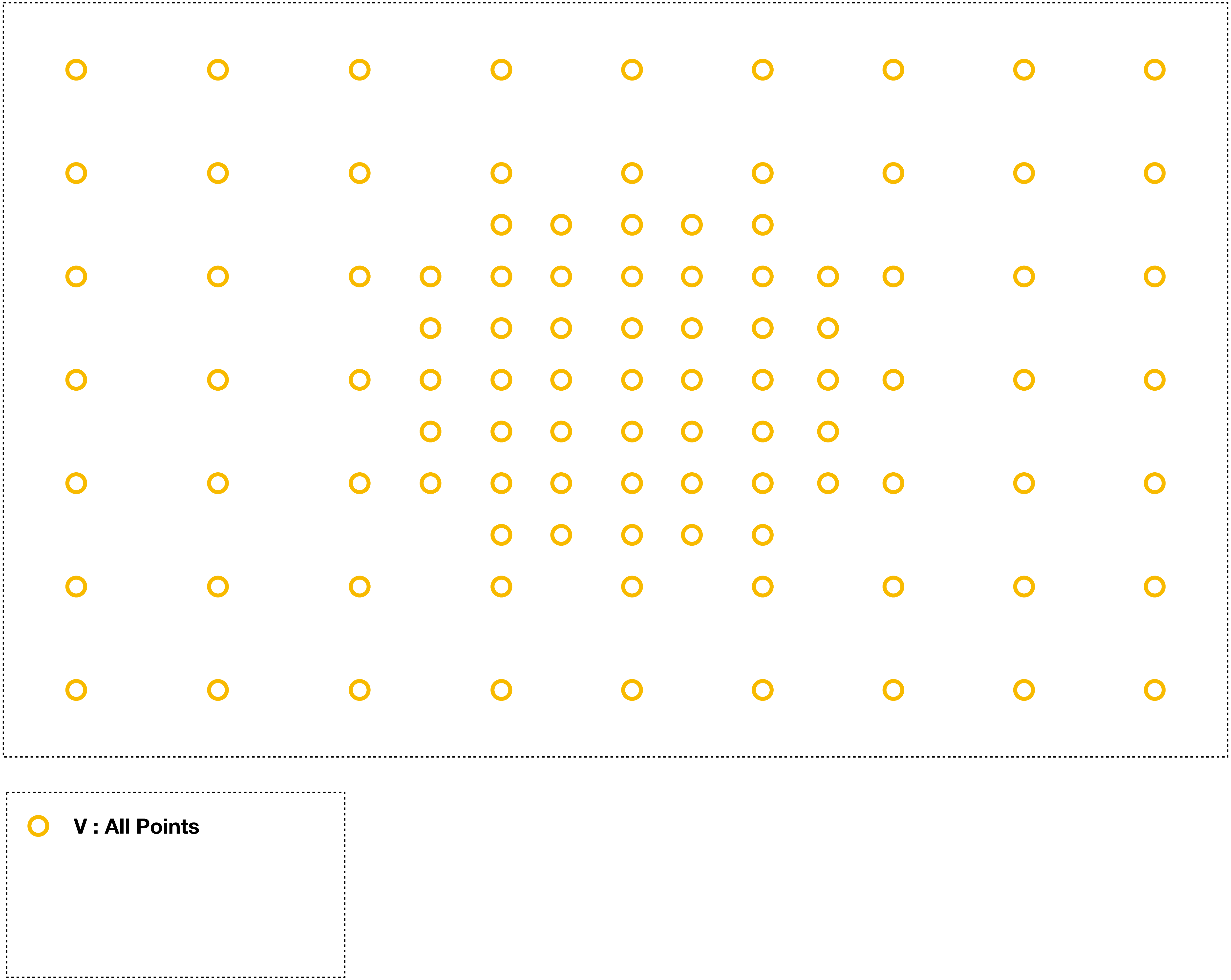
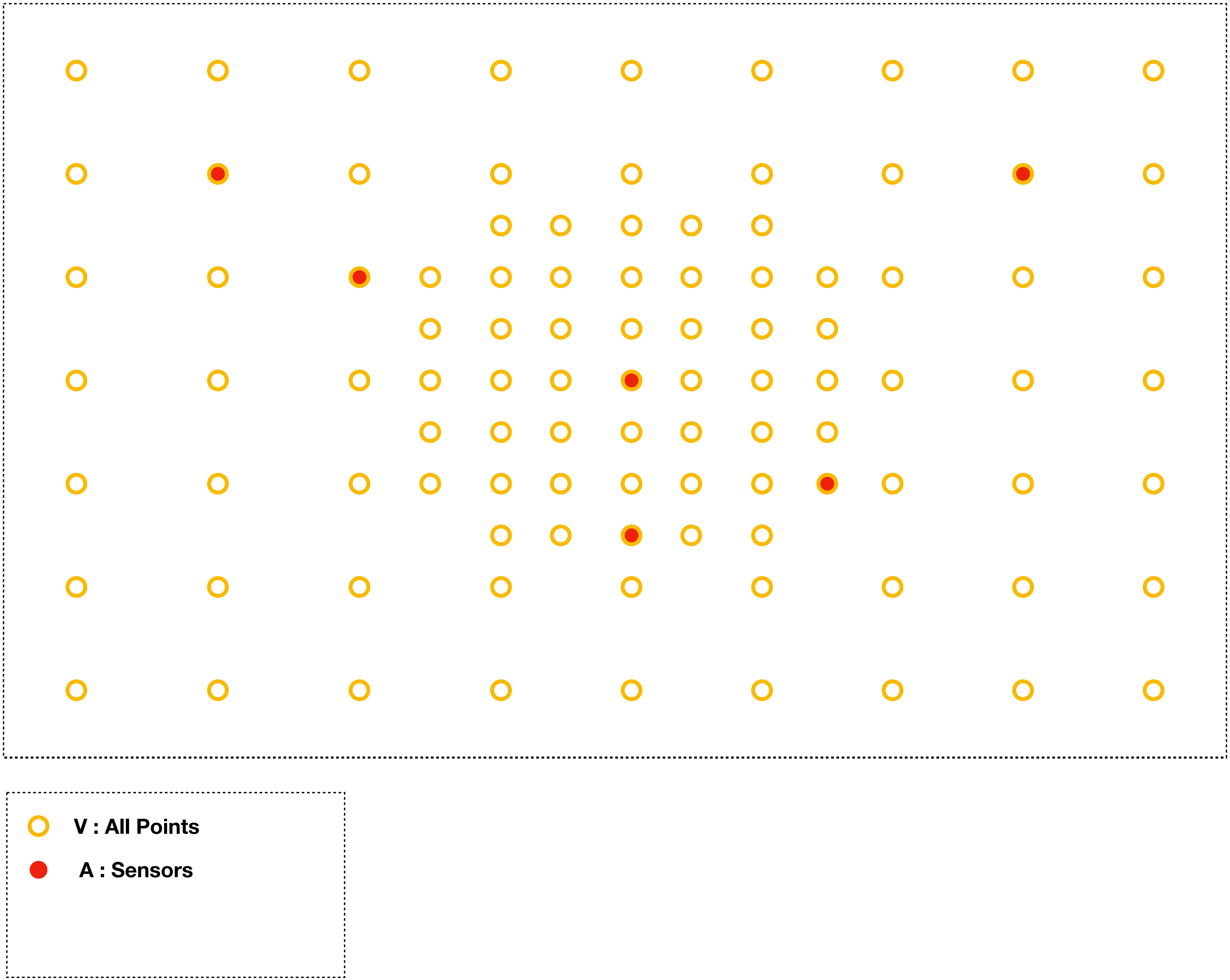
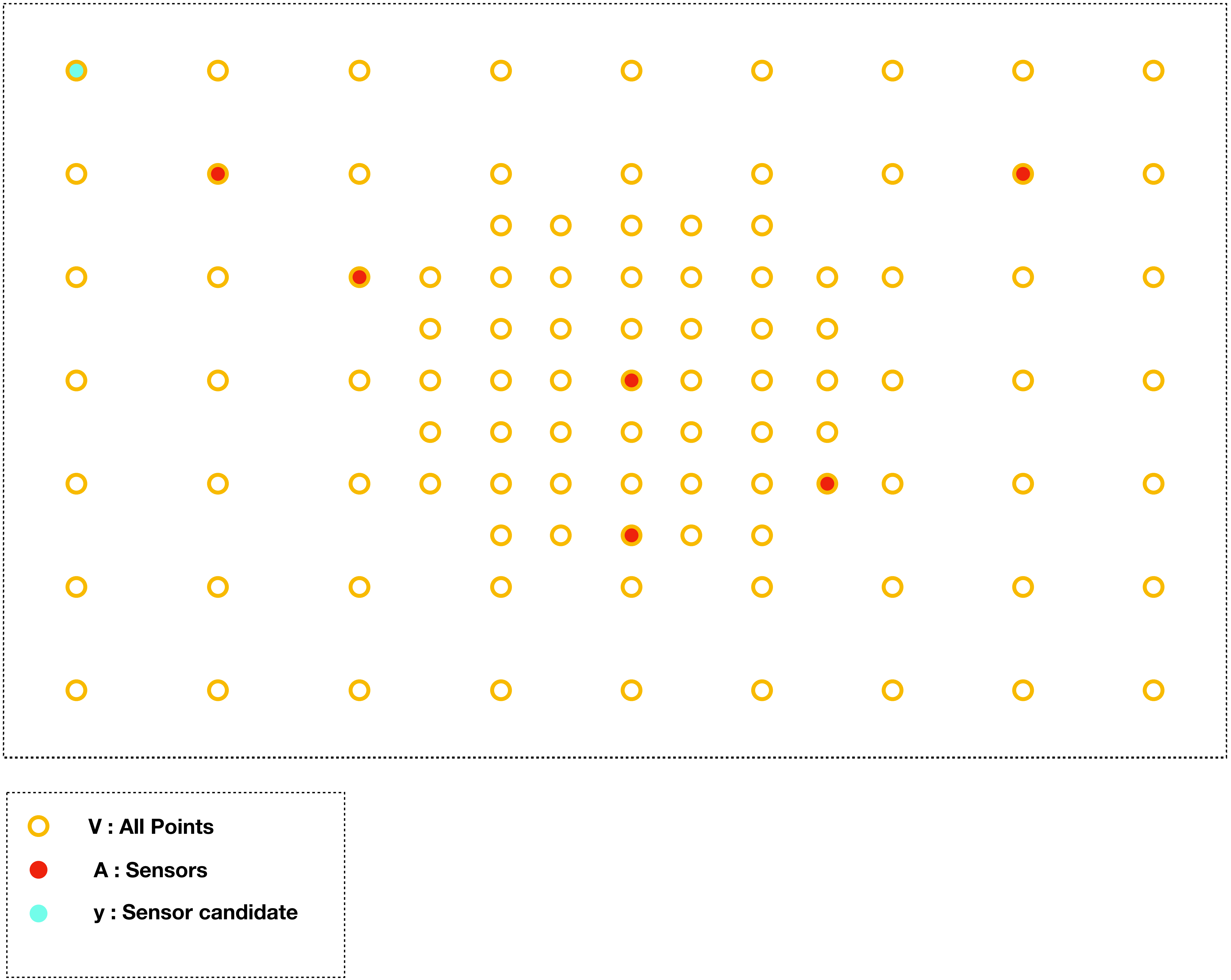
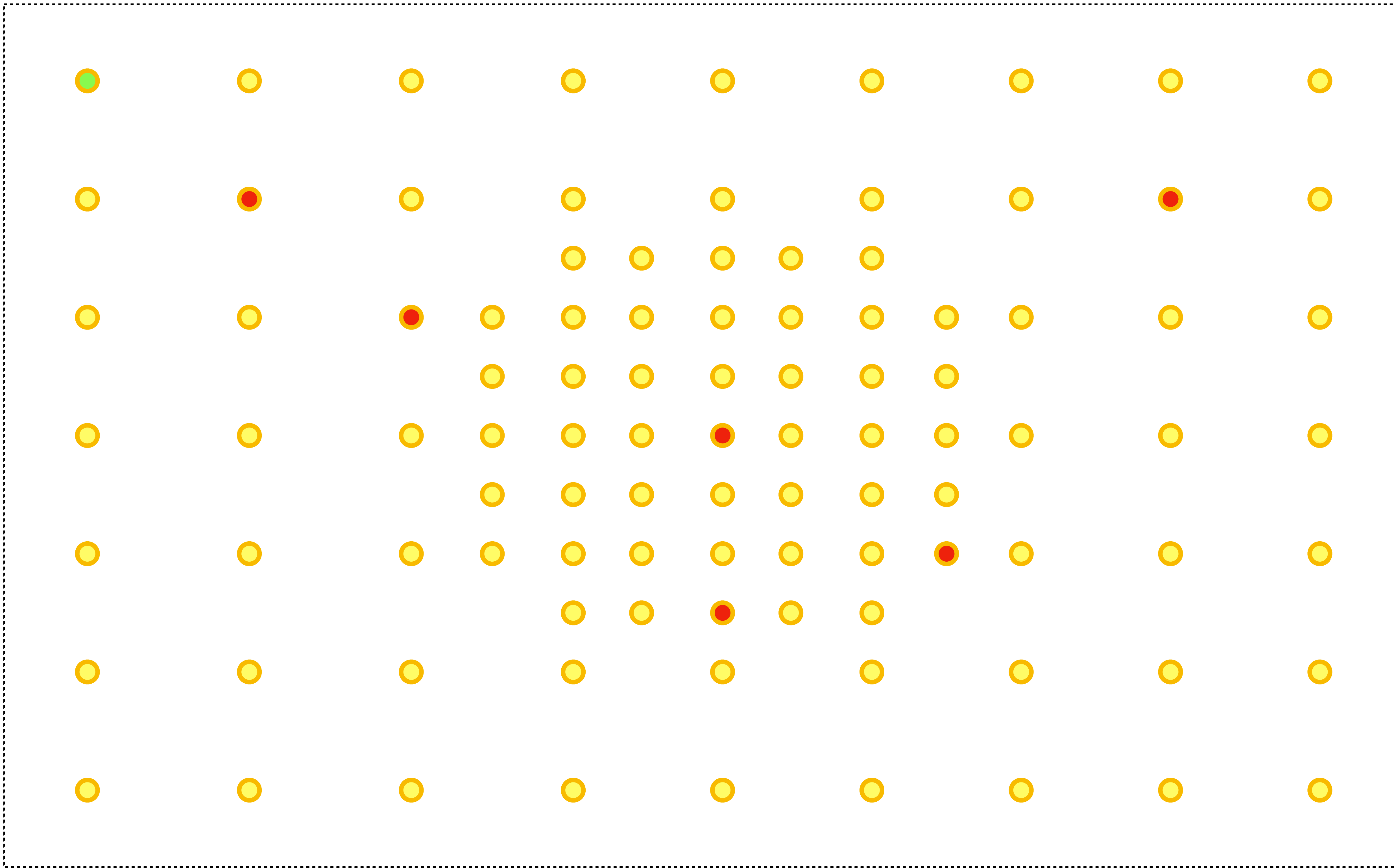
Sensor Location Optimisation using Gaussian Processes

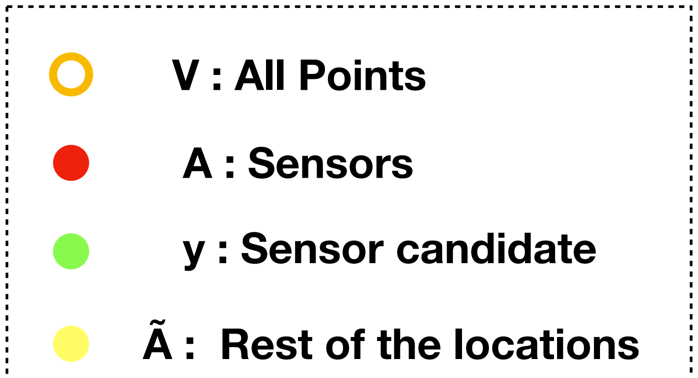
A Graphical Overview


V : All Points


V : All Points
A : Sensors


V : All Points
A : Sensors
y : Sensor candidate



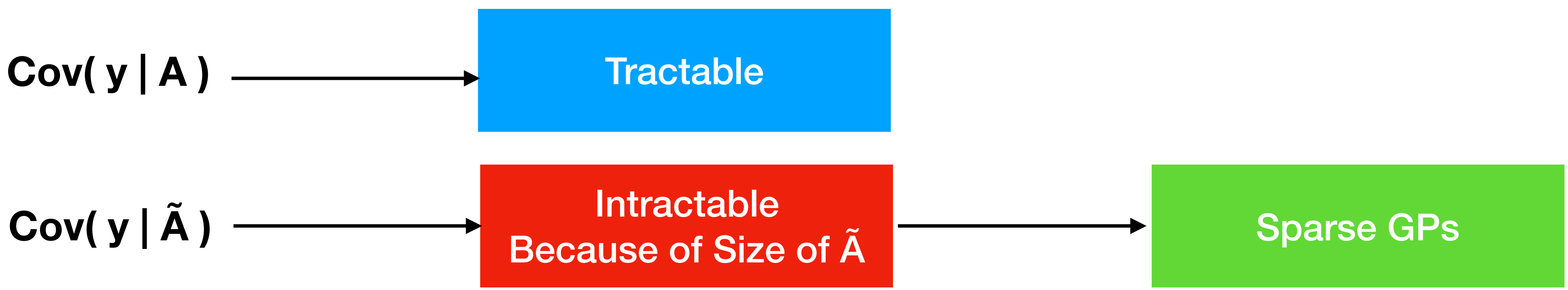


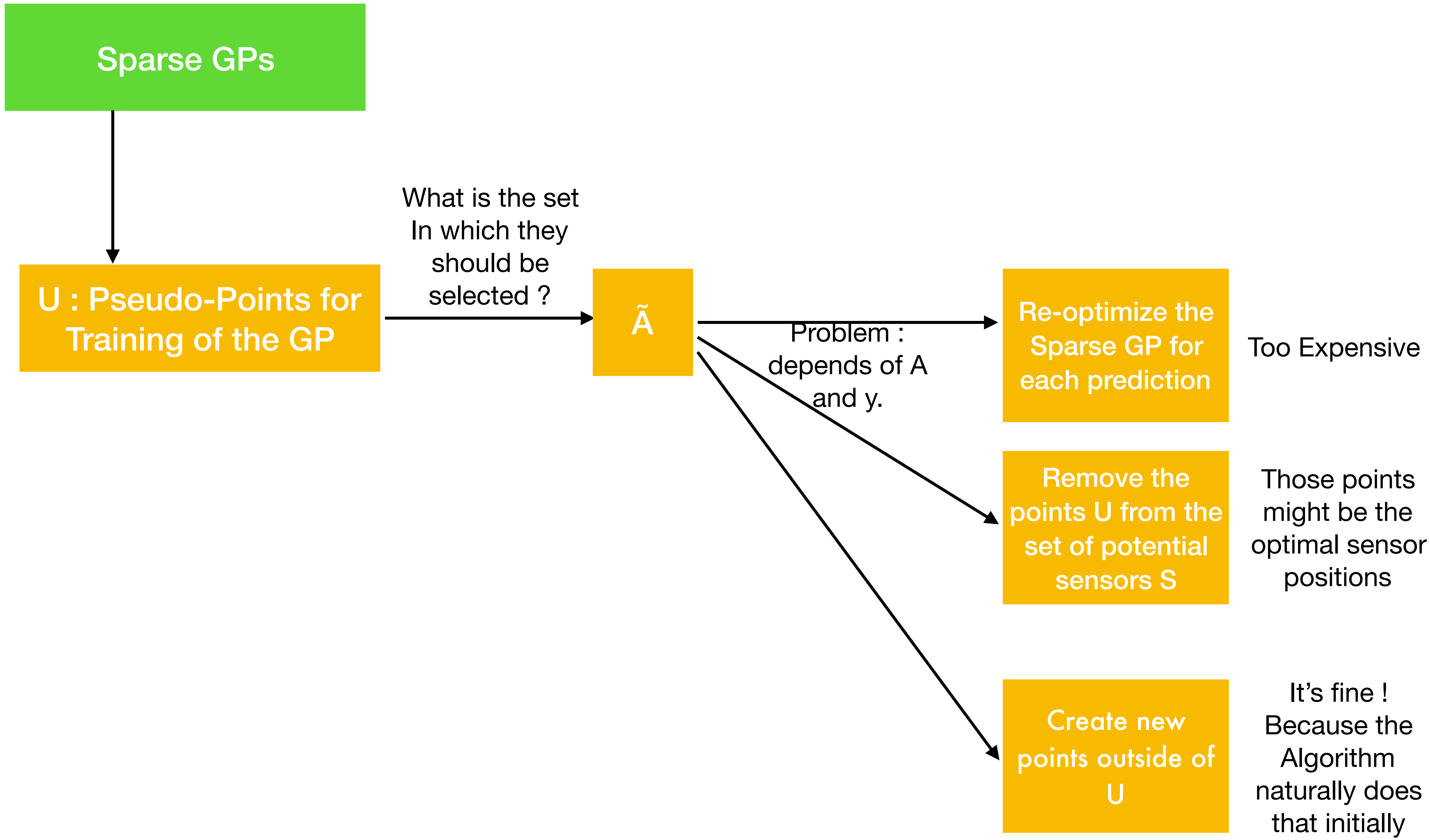
Gain in Mutual Information :

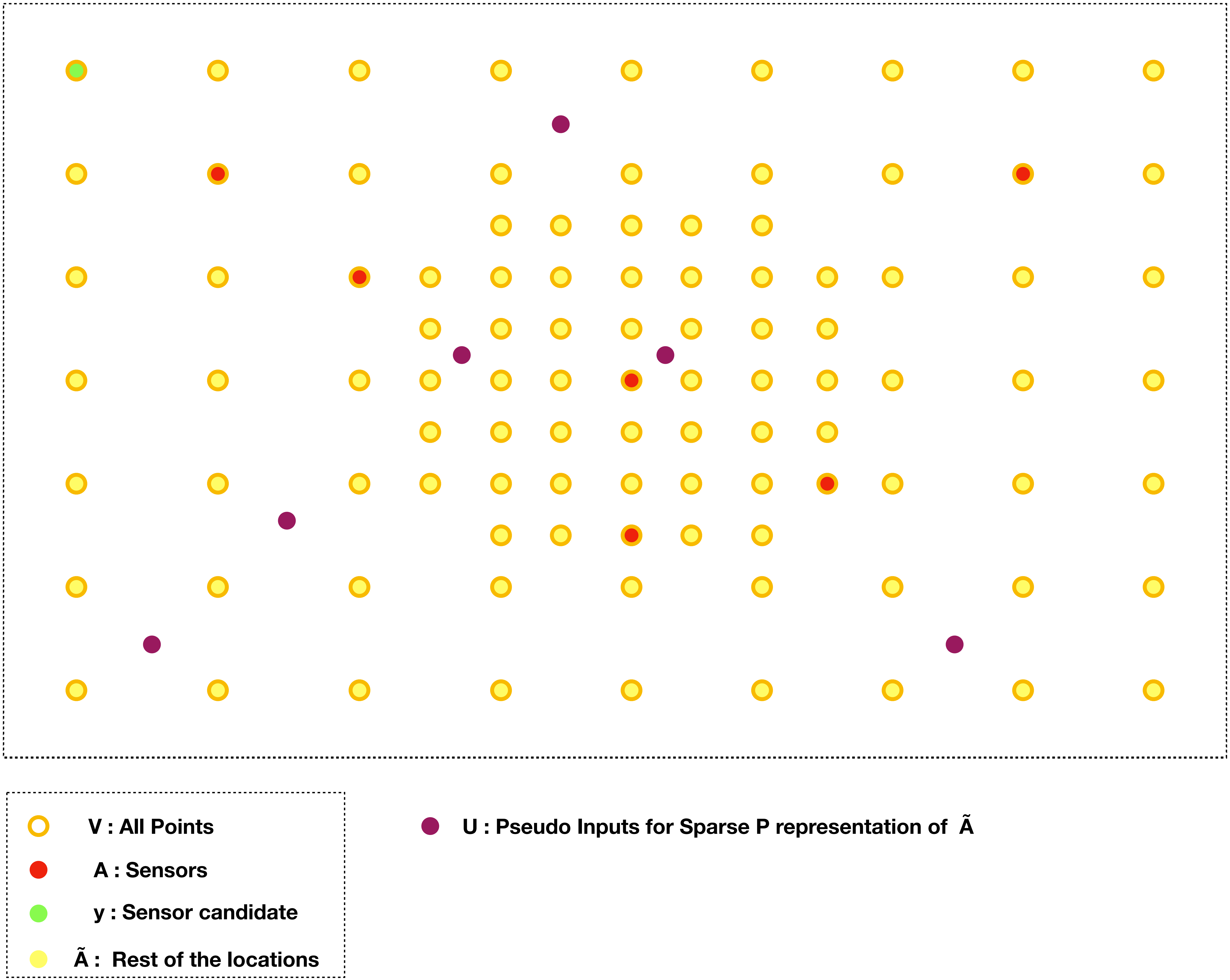
$$MI(A + y) - MI(A) = H(y \mid A) - H(y \mid \tilde{A})$$

$$\Rightarrow \; Cov(y \mid A) / Cov(y \mid \tilde{A})$$

Cov(y | A) and Cov(y | Ã) are obtained via GPs




Sparse GPs
U : Pseudo-Points for Training of the GP
What is the set In which they should be selected ?
Ã
Problem : depends of A and y.
Re-optimize the Sparse GP for each prediction
Too Expensive
Remove the points U from the set of potential sensors S
Those points might be the optimal sensor positions
Create new points outside of U
It's fine ! Because the Algorithm naturally does that initially


V : All Points
A : Sensors
y : Sensor candidate
Ã : Rest of the locations
U : Pseudo Inputs for Sparse P representation of Ã

Small Scale

- Numpy and GPy give consistent results

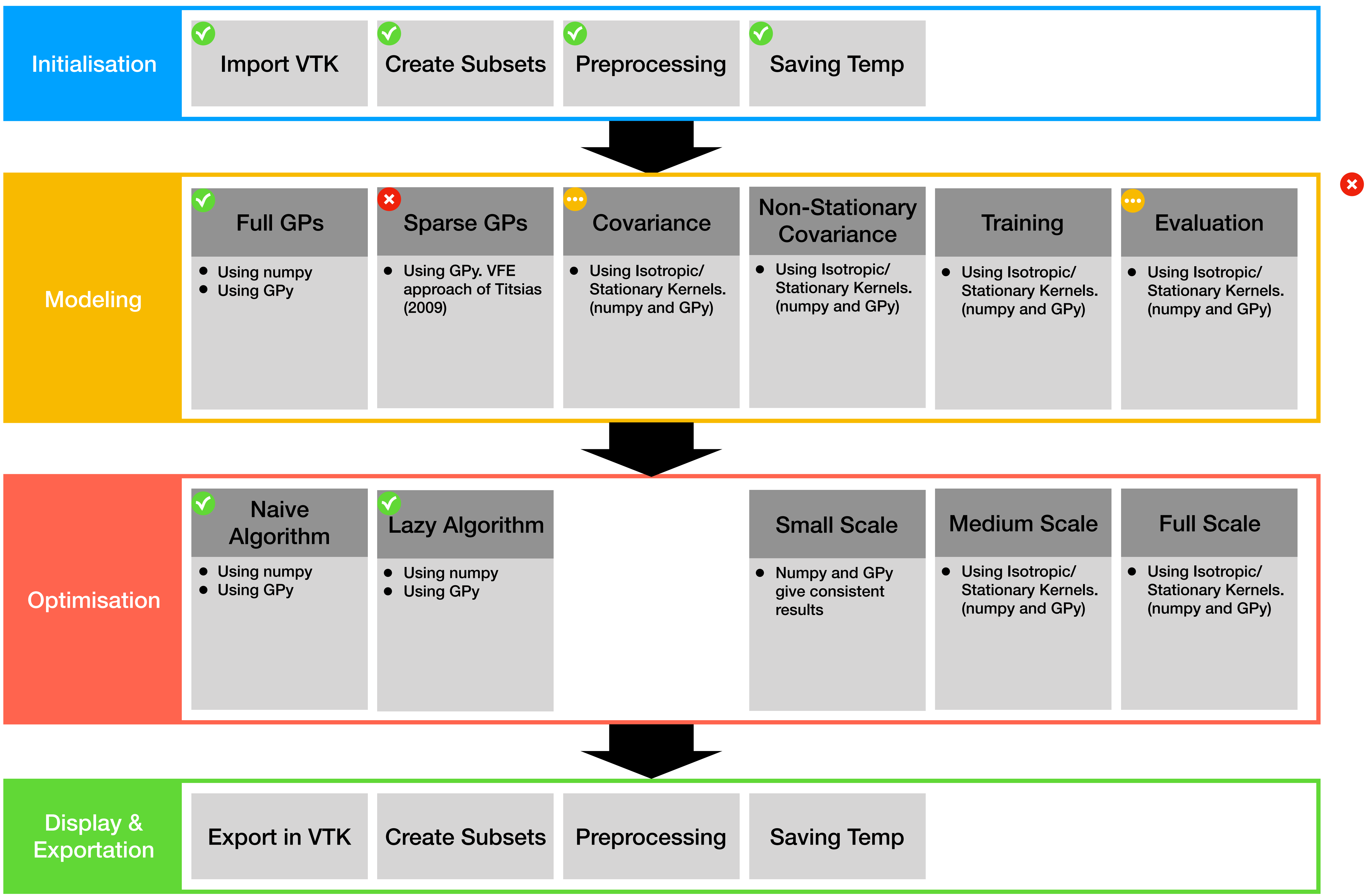
Medium Scale

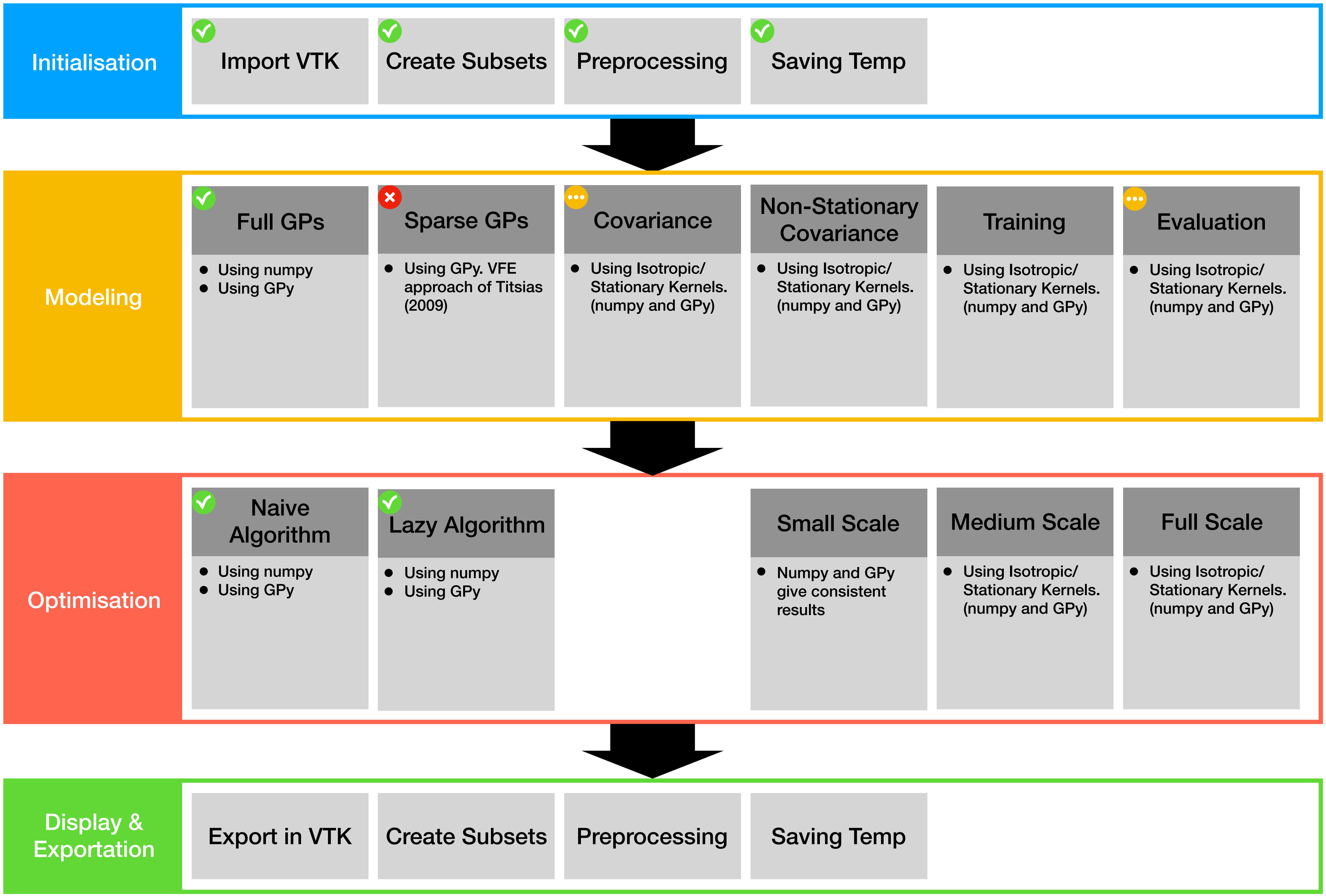
- Using Isotropic/ Stationary Kernels. (numpy and GPy)

Full Scale

- Using Isotropic/ Stationary Kernels. (numpy and GPy)

Roadmap Sensors GP Project




Initialisation
Import VTK
Create Subsets
Preprocessing
Saving Temp
Modeling
Full GPs
Using numpy
Using GPy
Sparse GPs
Using GPy. VFE approach of Titsias (2009)
Covariance
Using Isotropic/ Stationary Kernels. (numpy and GPy)
Non-Stationary Covariance
Using Isotropic/ Stationary Kernels. (numpy and GPy)
Training
Using Isotropic/ Stationary Kernels. (numpy and GPy)
Evaluation
Using Isotropic/ Stationary Kernels. (numpy and GPy)
Optimisation
Naive Algorithm
Using numpy
Using GPy
Lazy Algorithm
Using numpy
Using GPy
Small Scale
Numpy and GPy give consistent results
Medium Scale
Using Isotropic/ Stationary Kernels. (numpy and GPy)
Full Scale
Using Isotropic/ Stationary Kernels. (numpy and GPy)
Display & Exportation
Export in VTK
Create Subsets
Preprocessing
Saving Temp

Focus on Covariance Estimation ?
Or Scalable GPs with simpler covariance