SUPERVISED QUADTREES:

a new metalearner for local data science

LEVI JOHN WOLF levi.john.wolf@bristol.ac.uk

CLUSTERING REGRESSION

GEOGRAPHICAL CLUSTER-REG

UNDERSTANDING QUADTREES

APPLYING QUADTREE REGRESSION

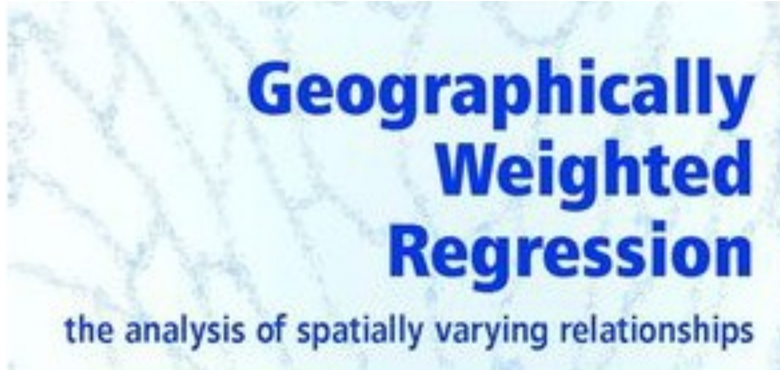
CLUSTERING REGRESSION

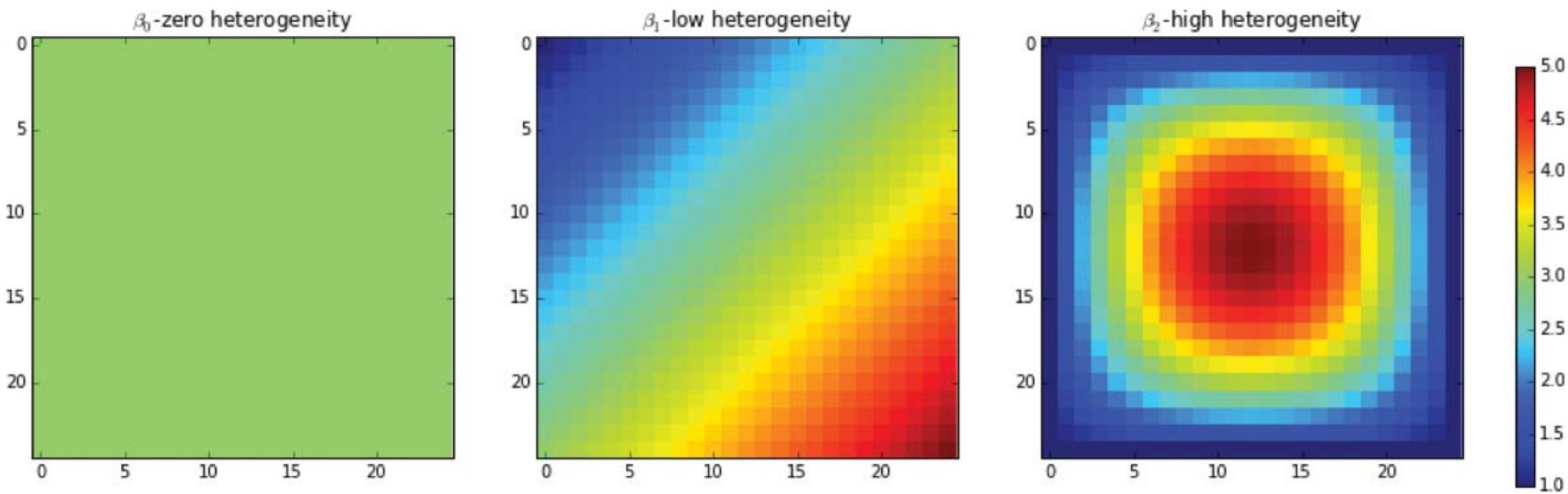
jointly solving clustering and regression

GEOGRAPHICAL CLUSTER-REG

UNDERSTANDING QUADTREES

APPLYING QUADTREE REGRESSION





WHAT IS CLUSTERING REGRESSION

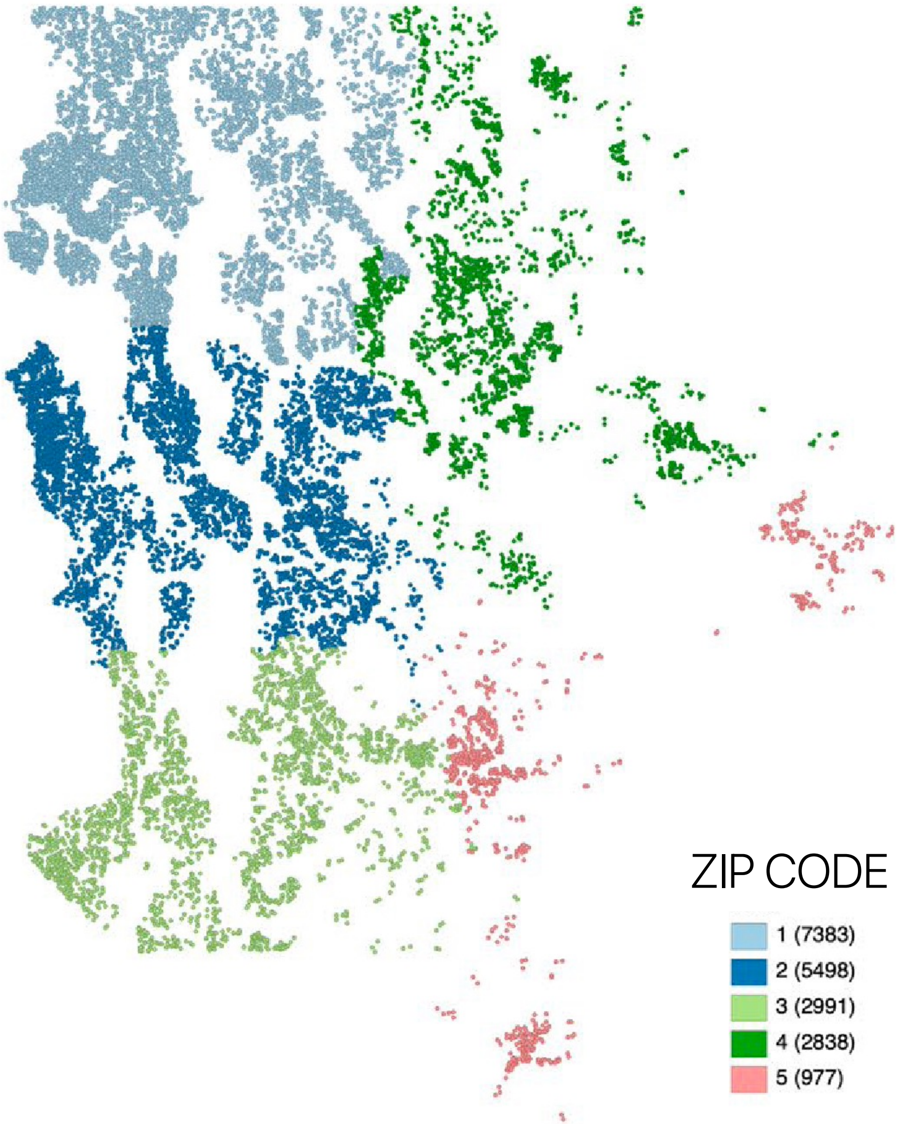
JOURNAL OF REGIONAL SCIENCE, VOL. 30, NO. 2, 1990, pp. 185–207

SPATIAL DEPENDENCE AND SPATIAL STRUCTURAL INSTABILITY IN APPLIED REGRESSION ANALYSIS*

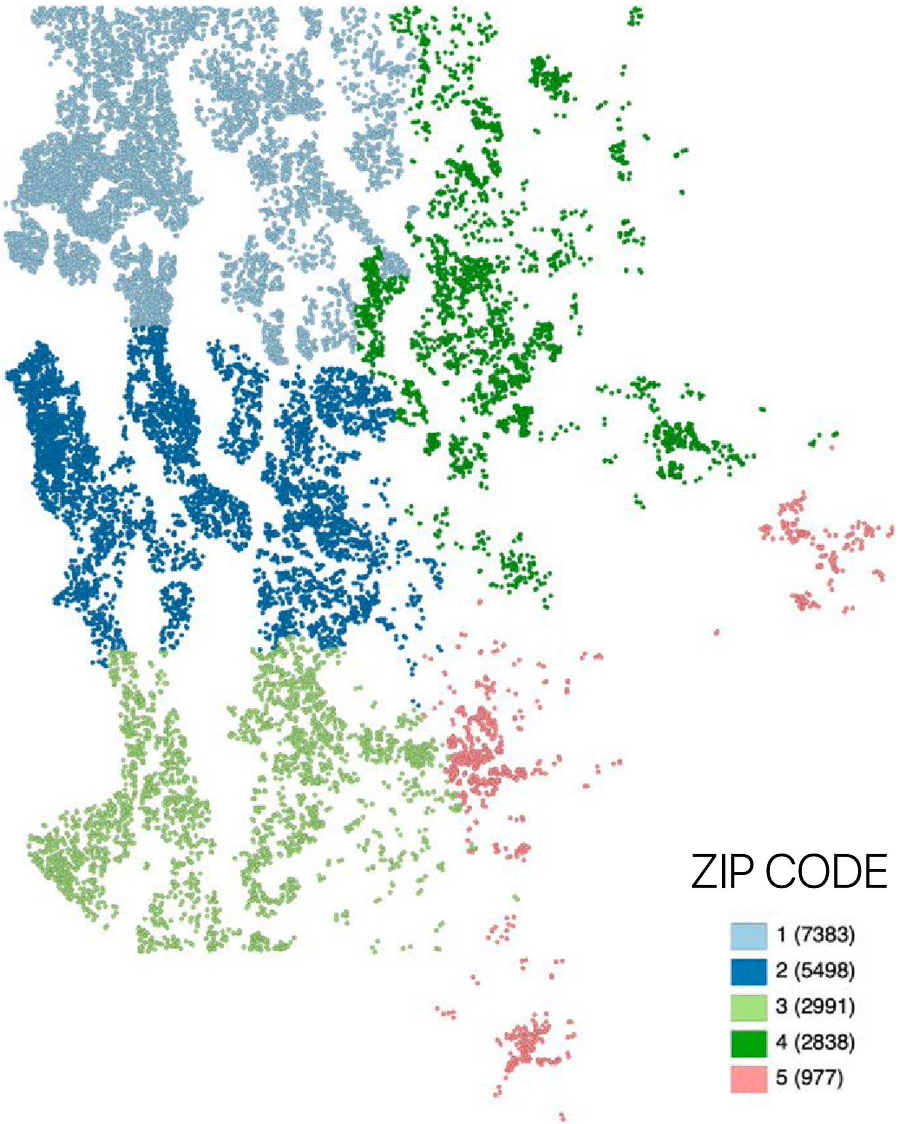
Luc Anselin†

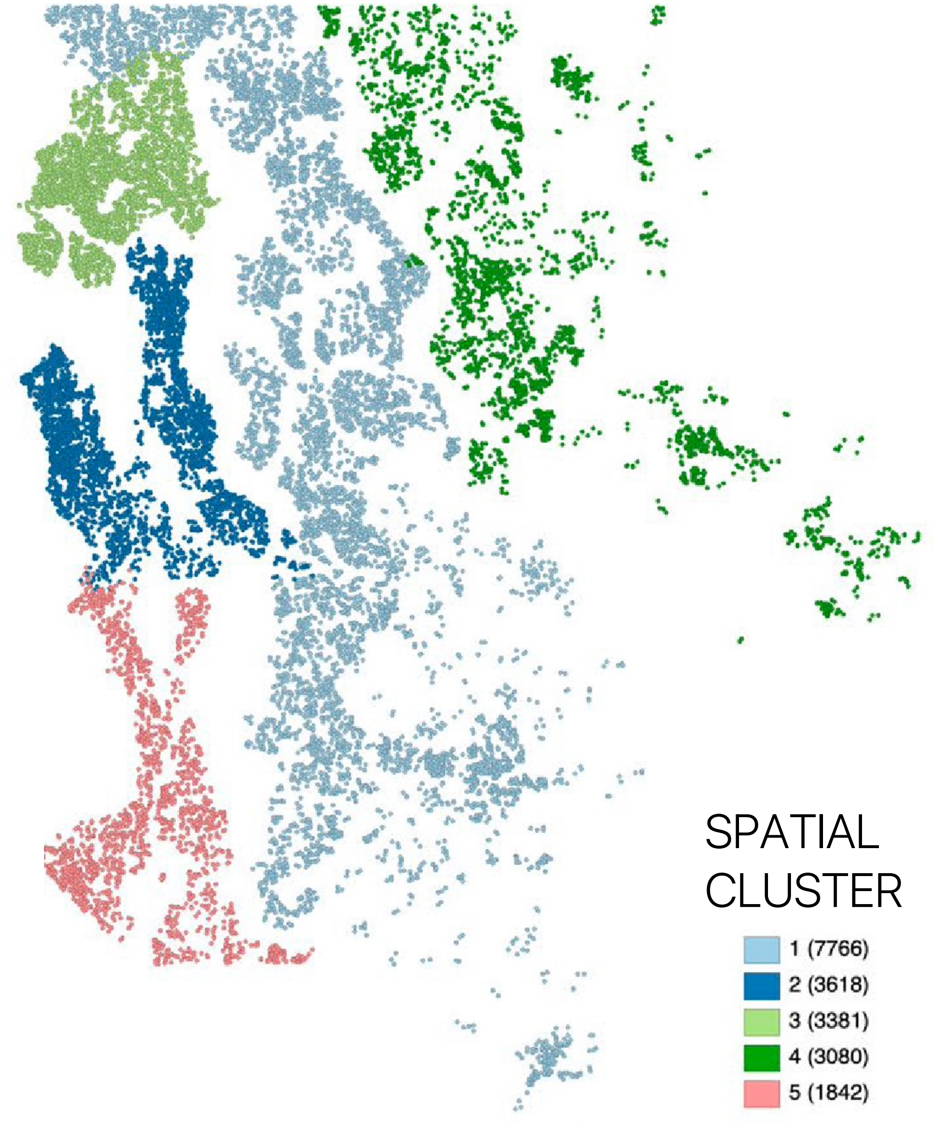
ABSTRACT. The stability of regression coefficients over the observation set ("regional homogeneity") is typically assessed by means of a Chow test or within a seemingly unrelated regression (SUR) framework. When spatial error autocorrelation is present in cross-sectional equations the traditional tests are no longer applicable. I evaluate this both in formal terms as well as empirically. I introduce a taxonomy of spatial effects in models for structural instability, and discuss its implication for testing. I compare the performance of traditional tests, robust approaches, maximum-likelihood procedures and pretest techniques by means of a series of simple Monte Carlo experiments.

WHAT IS CLUSTERING REGRESSION



WHAT IS CLUSTERING REGRESSION



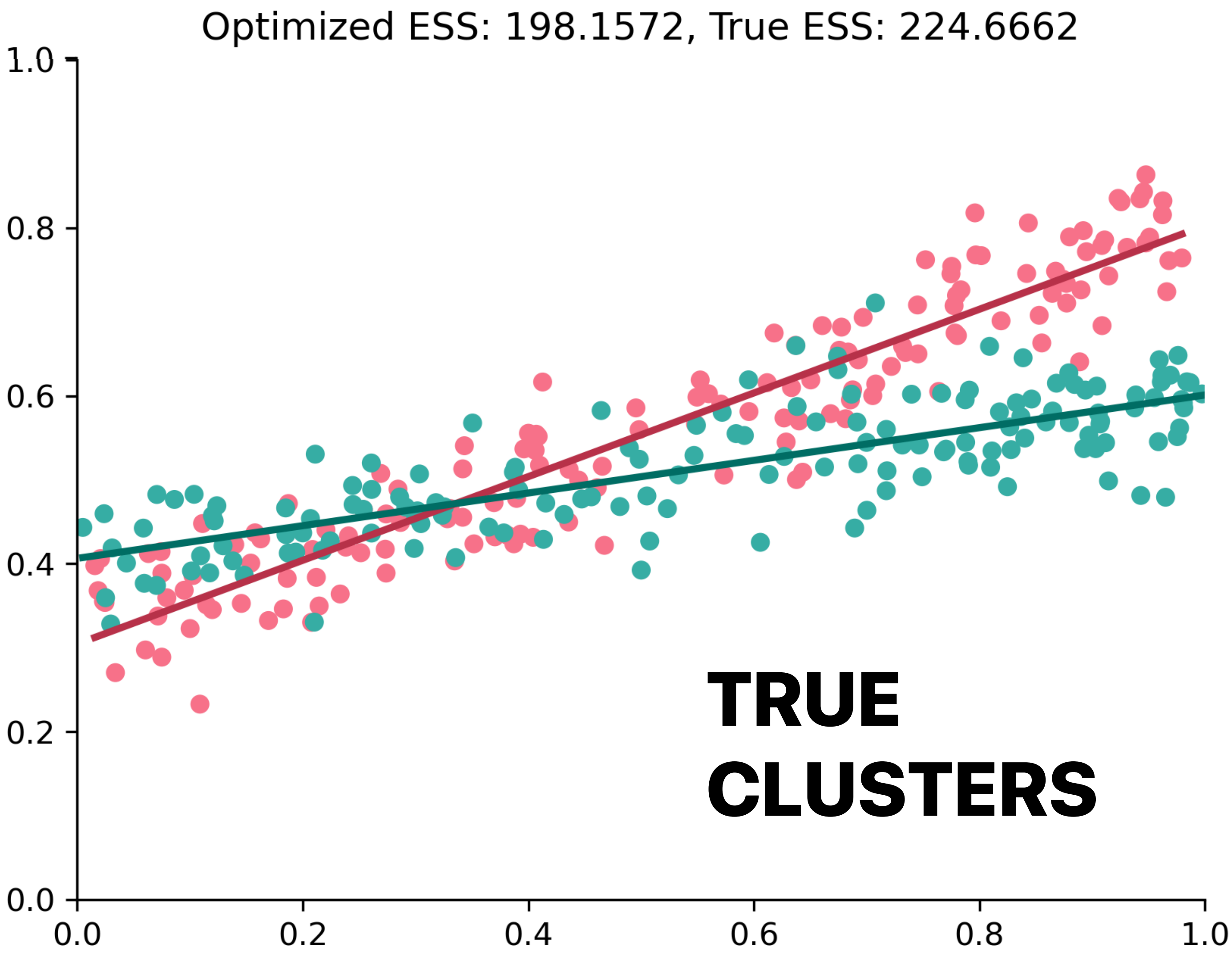


WHAT IS CLUSTERING REGRESSION

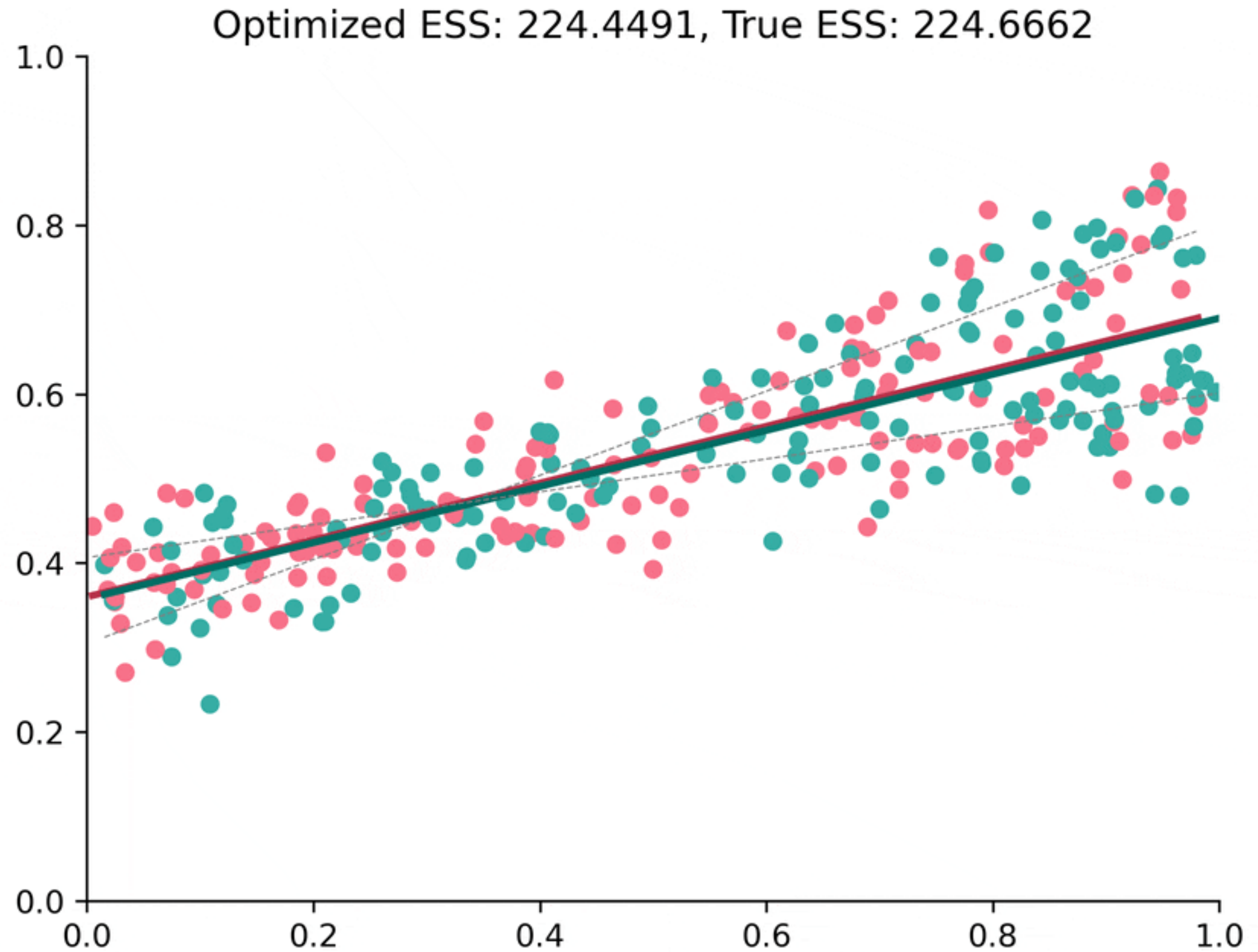
Späth (1979)'s three-stage process
- **CLUSTER:** find clusters in the data
- **ESTIMATE:** fit the model using clusters as a feature
- **REFINE:** flip poorly-fit observations to a different cluster

Heuristic (i.e. not optimal), slow+computationally intensive, likelihood-dependent, even in the seriously improved spatial variant provided by Sugasawa and Murakami (2021)
No guarantee of recovering the original classes

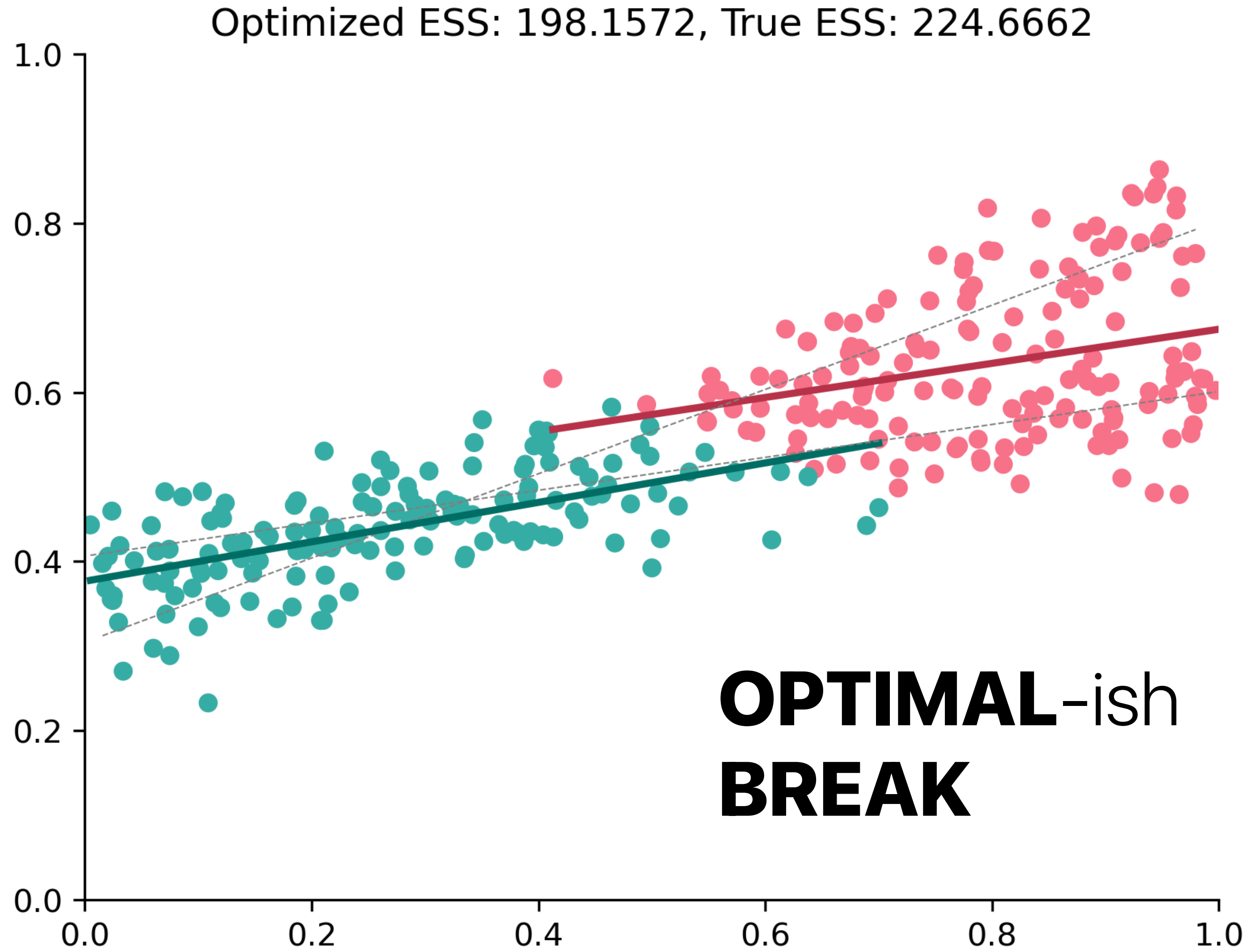
BEFORE THE SUPERVISED QUADTREE



SPÄTH (1979) ITERATIVE CLUSTERING



SPÄTH (1979) ITERATIVE CLUSTERING



SPÄTH (1979) ITERATIVE CLUSTERING

CLUSTERING REGRESSION

jointly solving clustering and regression

GEOGRAPHICAL CLUSTER-REG

jointly solving clustering and regression

UNDERSTANDING QUADTREES

APPLYING QUADTREE REGRESSION

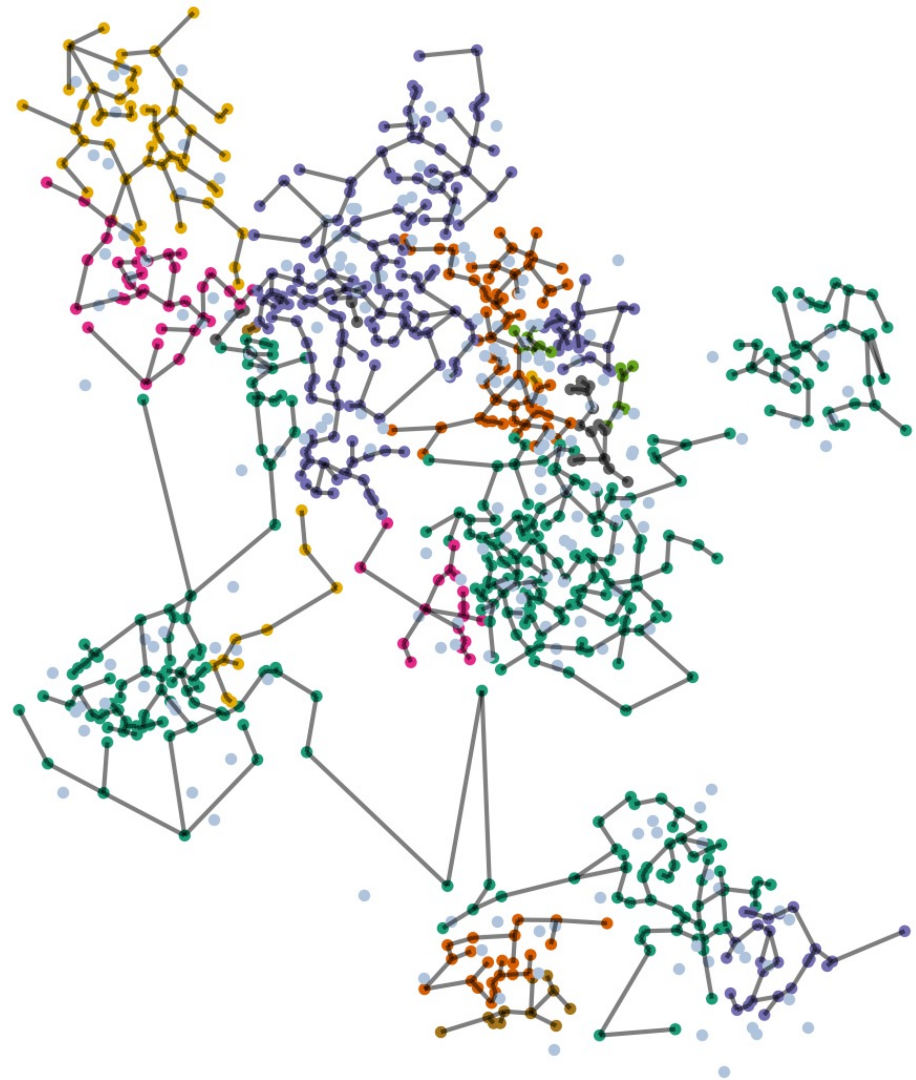
Späth (1979)'s three-stage process
- **CLUSTER:** find clusters in the data
- **ESTIMATE:** fit the model using clusters as a feature
- **REFINE:** flip poorly-fit observations to a different cluster

Heuristic (i.e. not optimal), slow+computationally intensive, likelihood-dependent, even in the seriously improved spatial variant provided by Sugasawa and Murakami (2021)
No guarantee of recovering the original classes

Anselin & Amaral (2023)
- **COMBINE:** Combine spatial and feature similarity
- **SPAN:** compute a MST over the combined affinity matrix
- **PRUNE:** remove MST links to create clusters according to $(SSR - (SSR_a + SSR_b))$

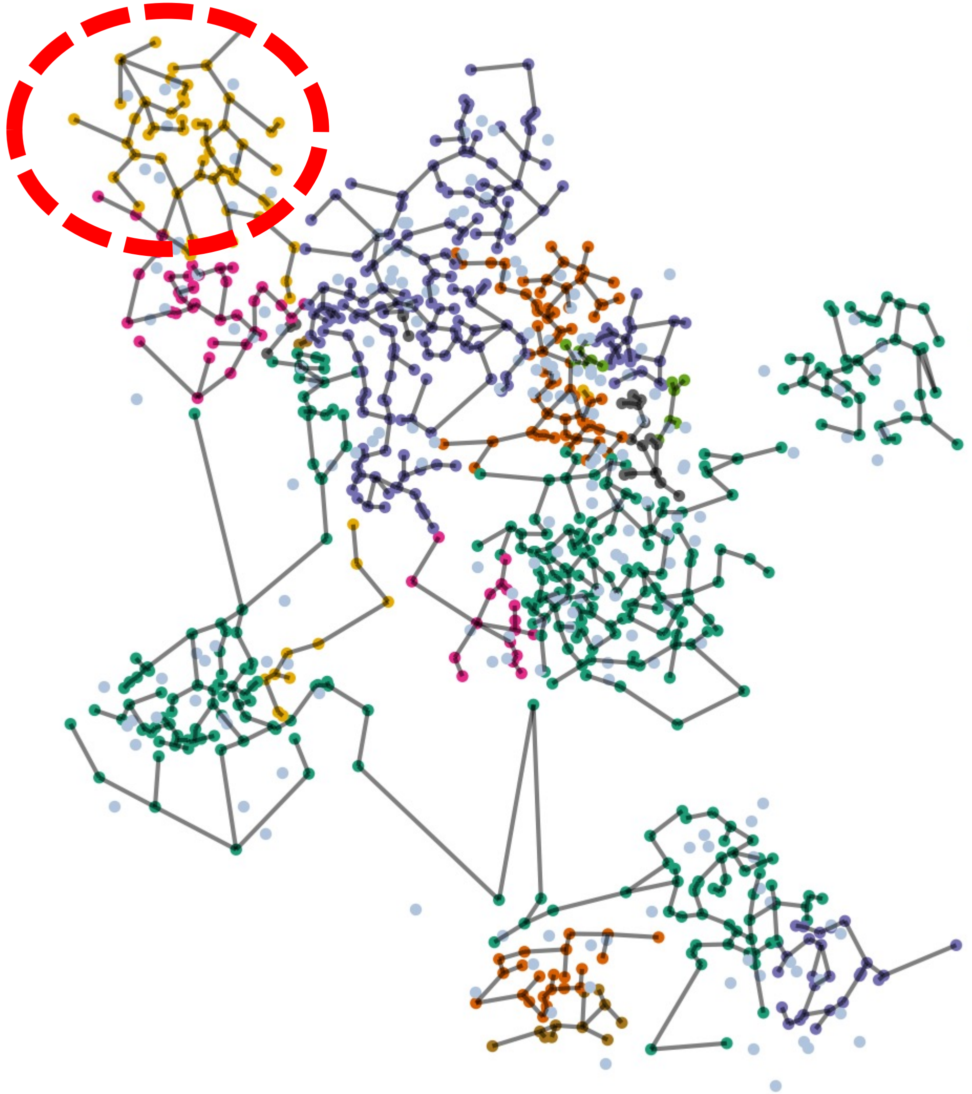
Spends degrees of freedom judiciously; n-clusters x n-features interactions
Remains fully within the standard Wald/LRT/Chow-based testing framework
MSTs can be quite unstable over Cross-validation
Requires prior knowledge of the deepest number of clusters
Avoids overfitting, since splits could be made below *j*

BEFORE THE SUPERVISED QUADTREE



SKATER REGRESSION

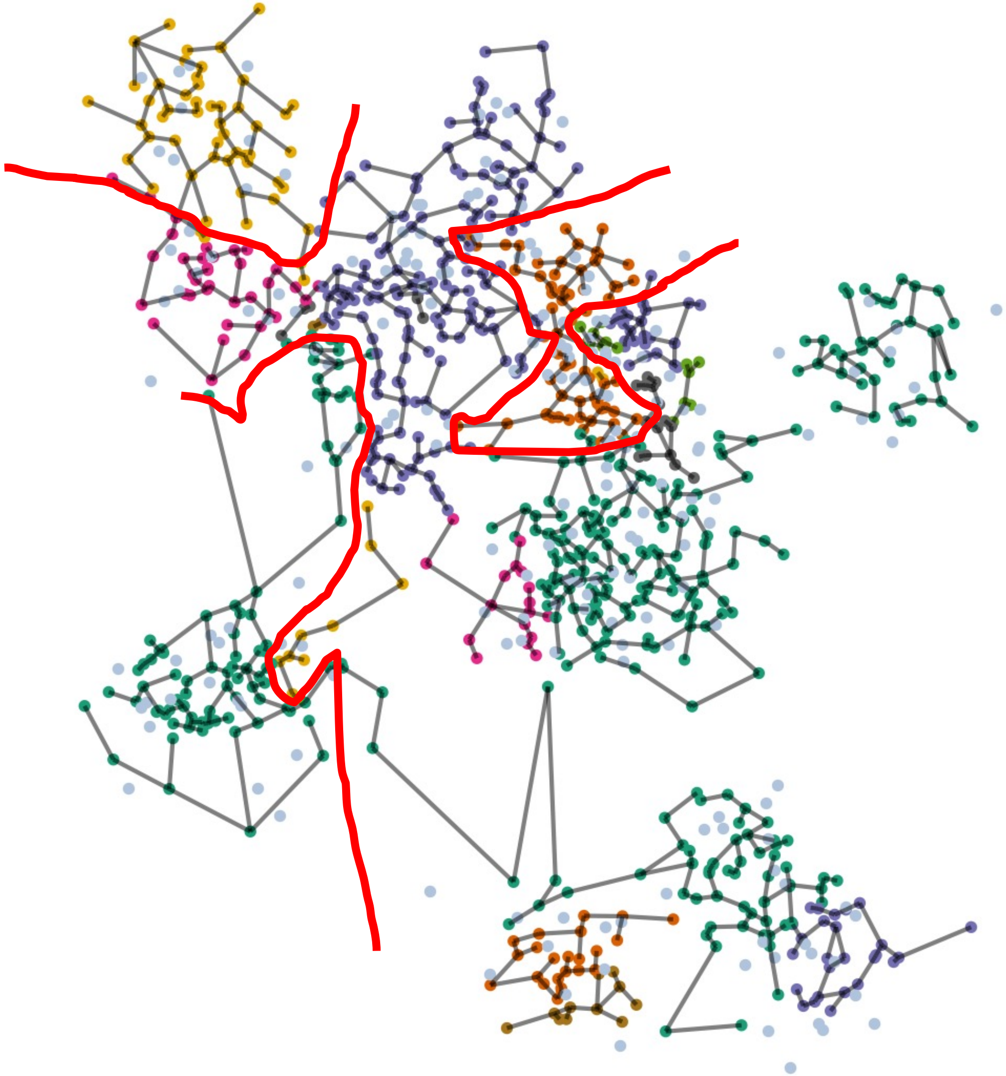
each cluster gets
one regression



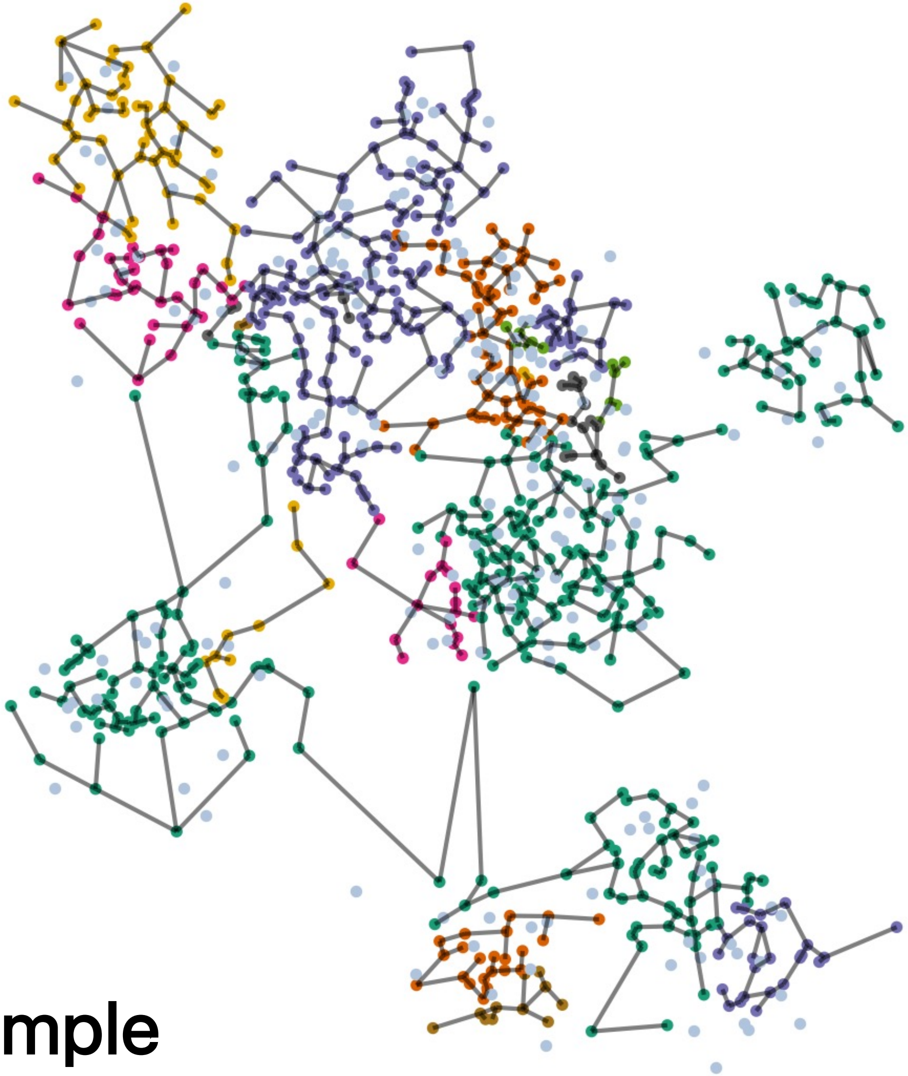
SKATER REGRESSION

each cluster gets
one regression

clusters are regions
that improve model fit

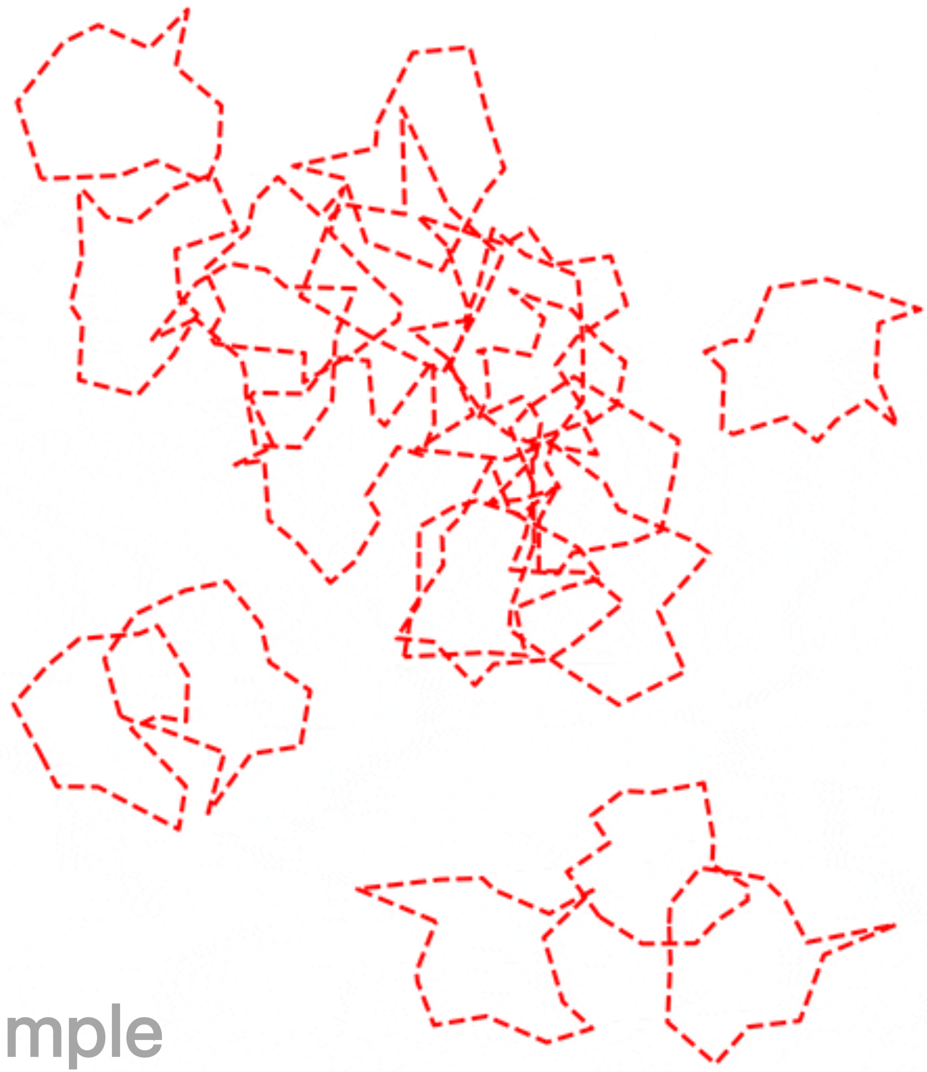


SKATER REGRESSION



MST Blues
- finnicky in CV
- no total order
- can't do out of sample

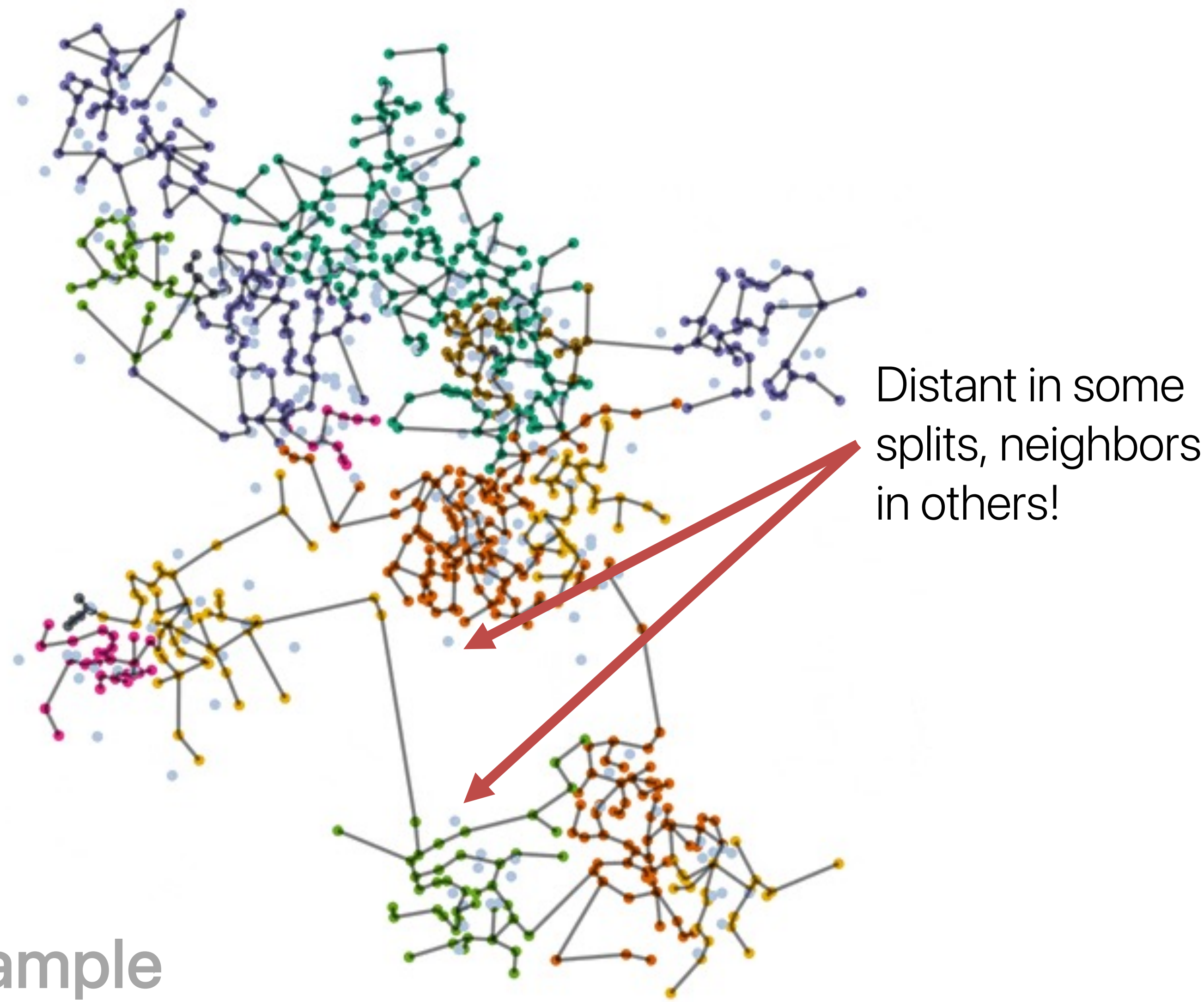
SKATER REGRESSION



MST Blues

- finnicky in CV
- no total order
- can’t do out of sample

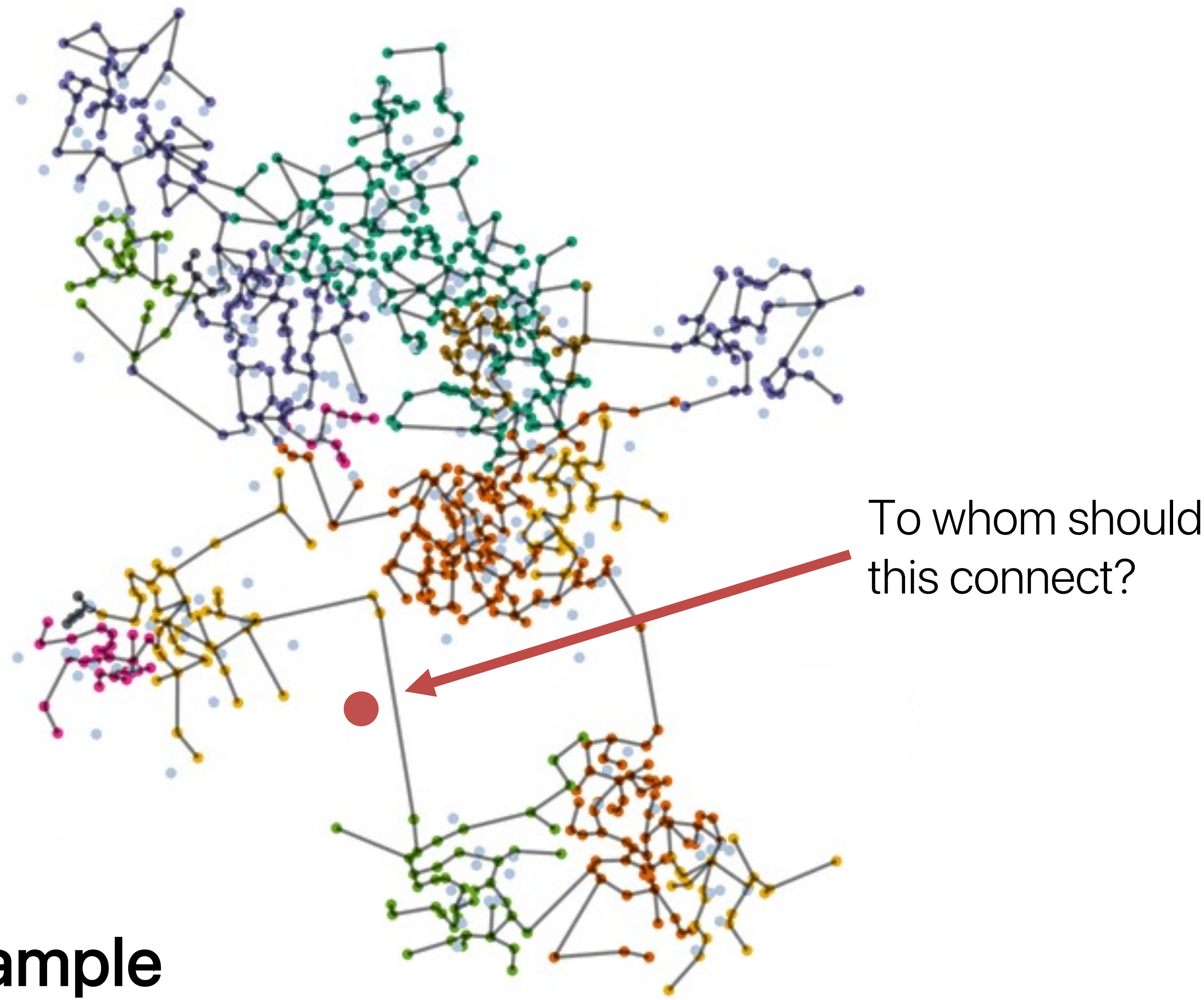
MST CHANGES DRAMATICALLY UNDER CV



MST Blues

- finnicky in CV
- no total order
- can't do out of sample

TREE ADJACENCY IS NOT FULL SIMILARITY



MST Blues

- finnicky in CV
- no total order
- can't do out of sample

ITERATIVE MST IS NOT FULL MST

CLUSTERING REGRESSION

jointly solving clustering and regression

GEOGRAPHICAL CLUSTER-REG

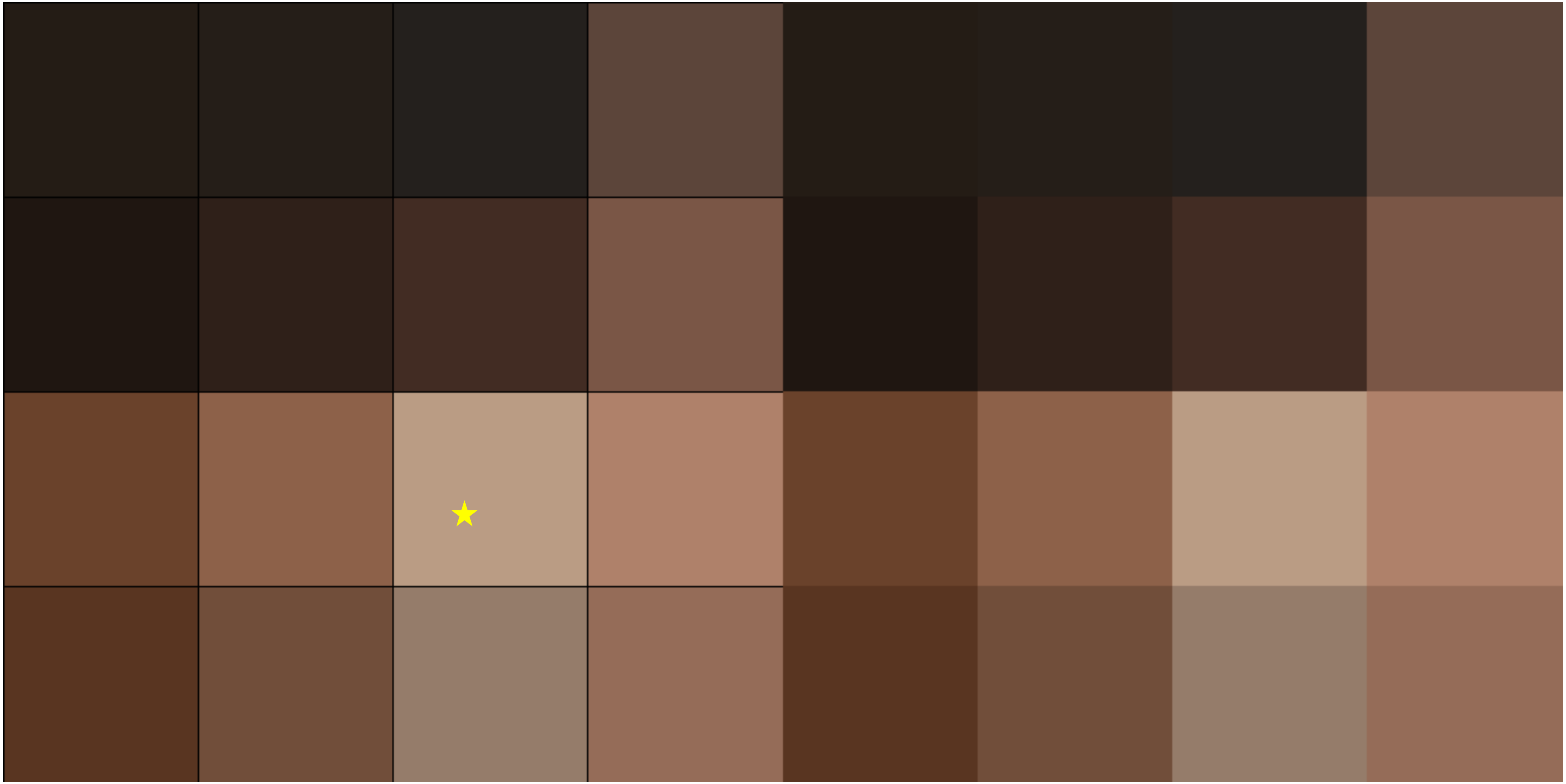
jointly solving clustering and regression

UNDERSTANDING QUADTREES

spatial splits for spatial fits

APPLYING QUADTREE REGRESSION

QUADTREES IN IMAGE COMPRESSION



SPEND DETAIL WHERE ITS NEEDED MOST

0. INITIALIZE

Fit a global model across all observations

1. SPLIT

Recursively find quadrants where local is improves on global

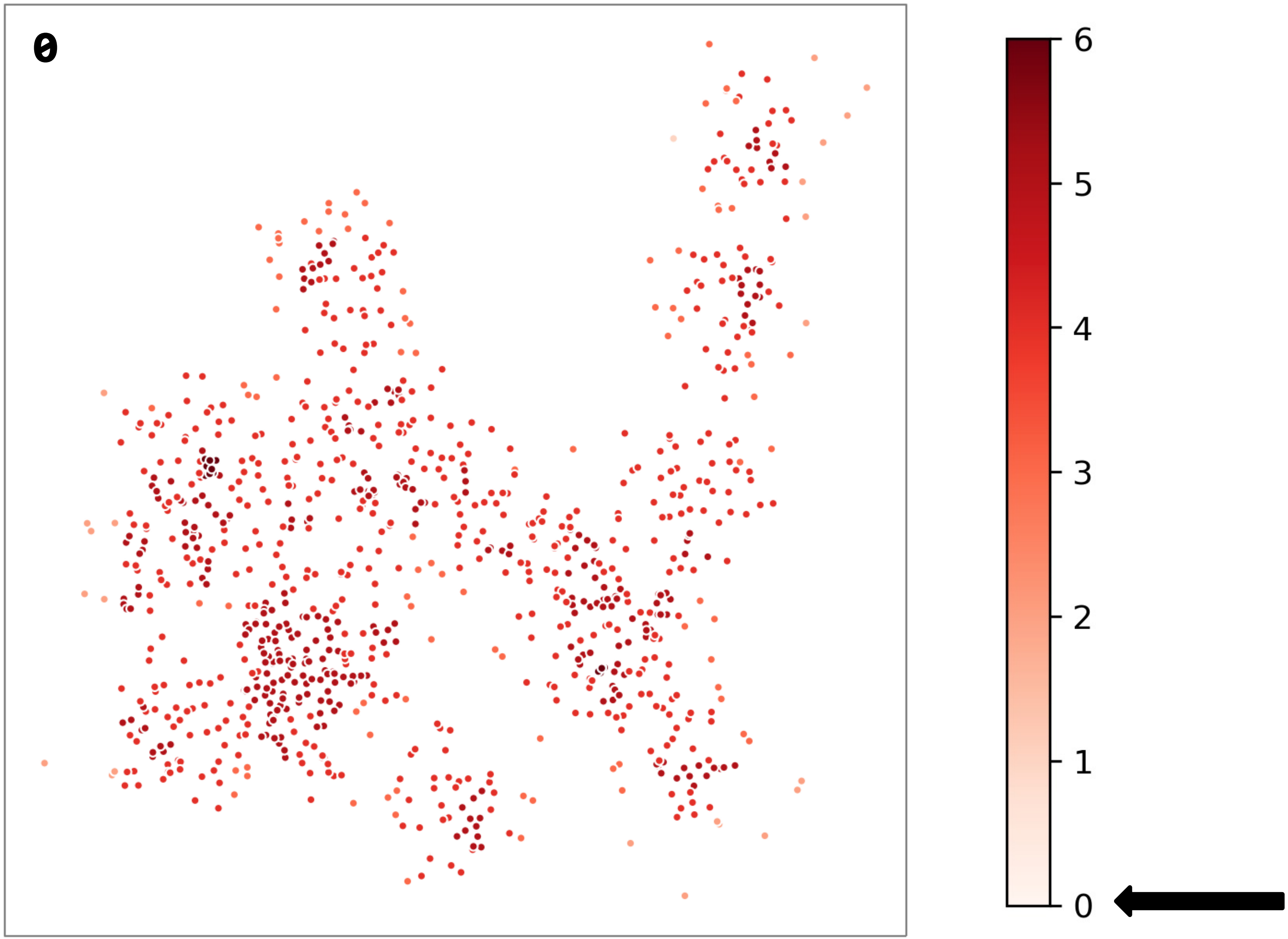
2. PRUNE

After, “roll up” useless leaf:feature interactions depth-first

3. FINALIZE

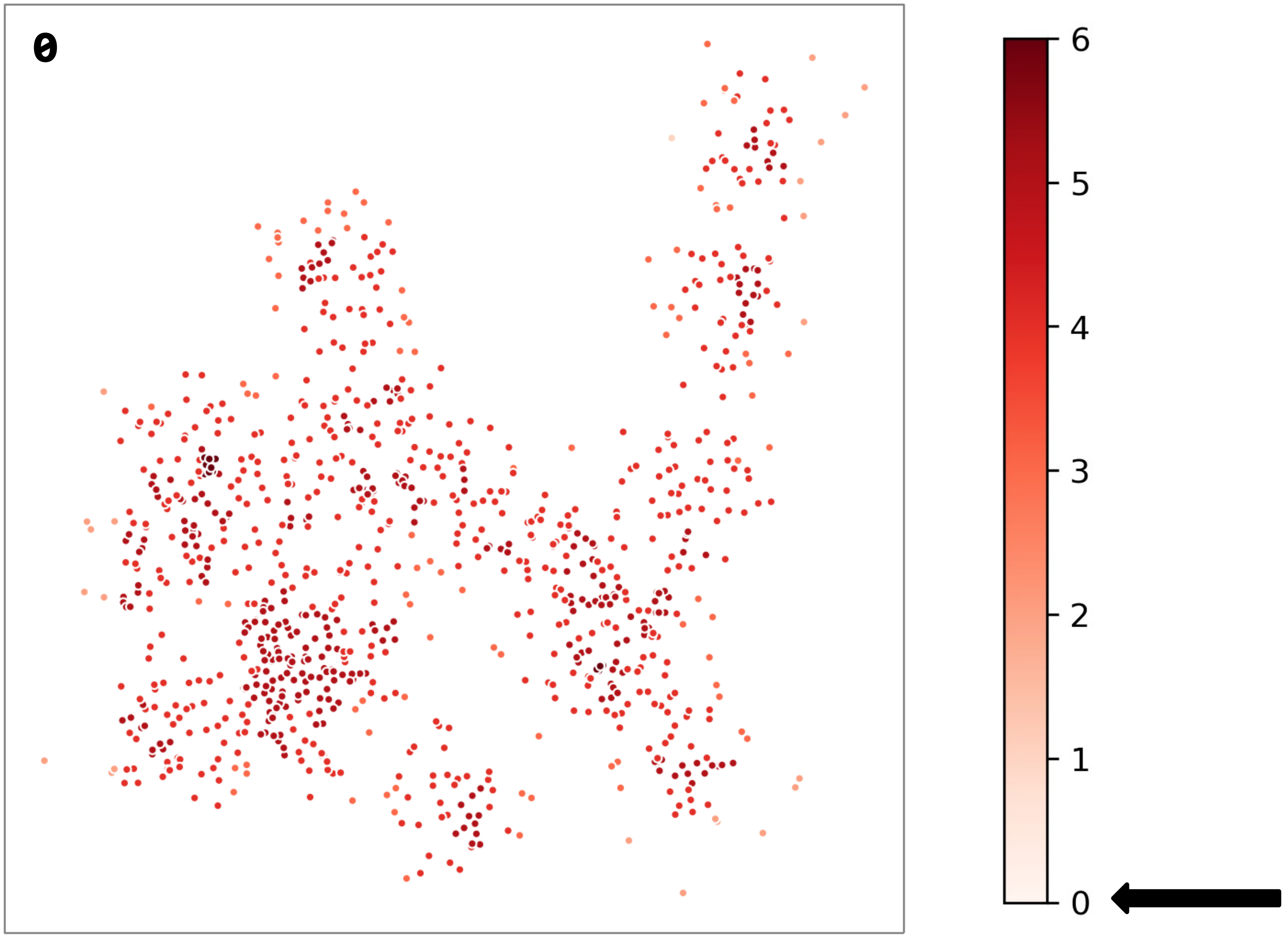
Interpret the pruned global model as you need

THE SUPERVISED QUADTREE

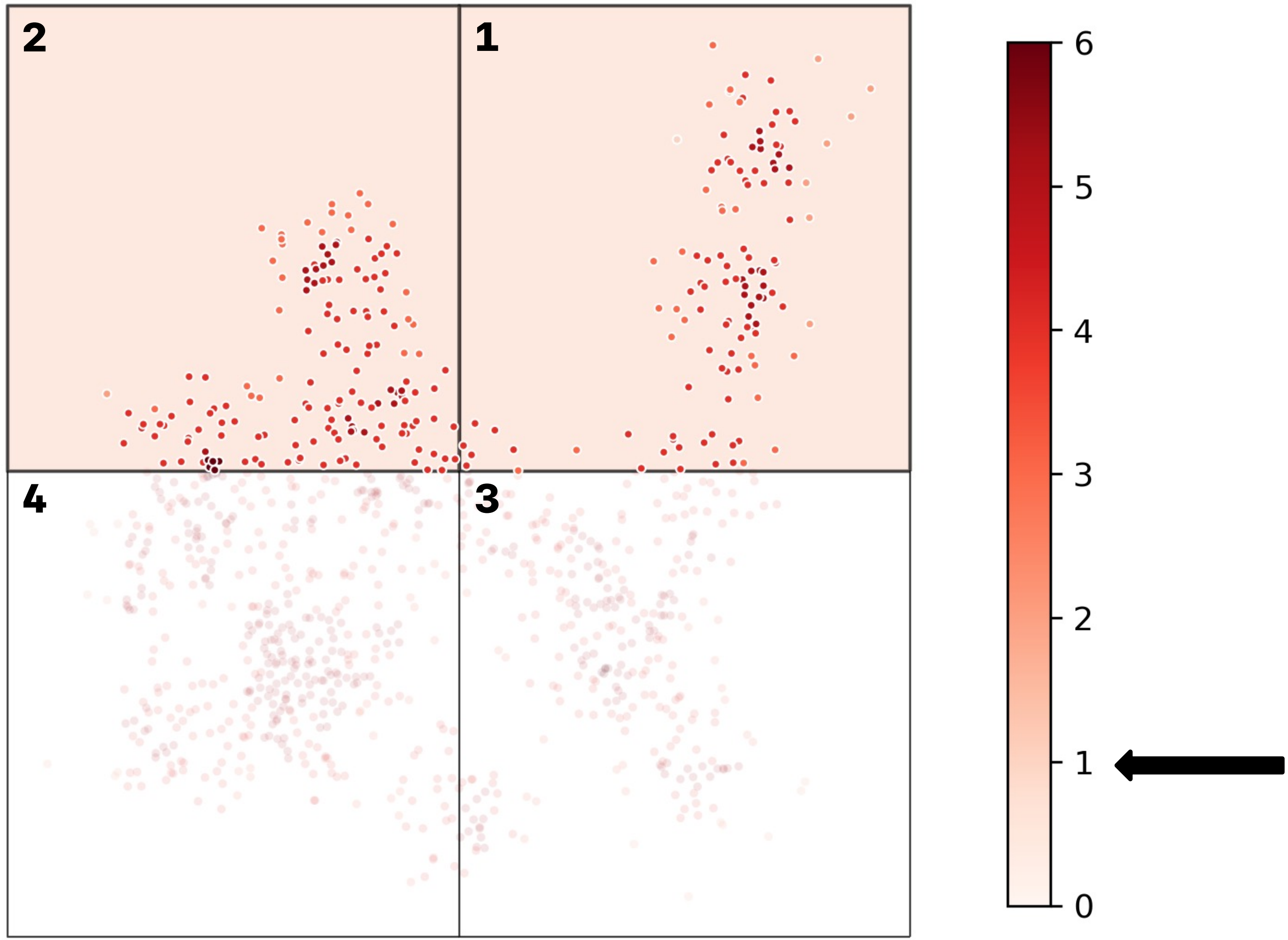


WALK DOWN THE DEPTH LADDER

c	x_1
1	x_{11}
1	x_{12}
1	x_{13}
1	x_{14}
1	x_{15}
…	…
1	x_{1n}



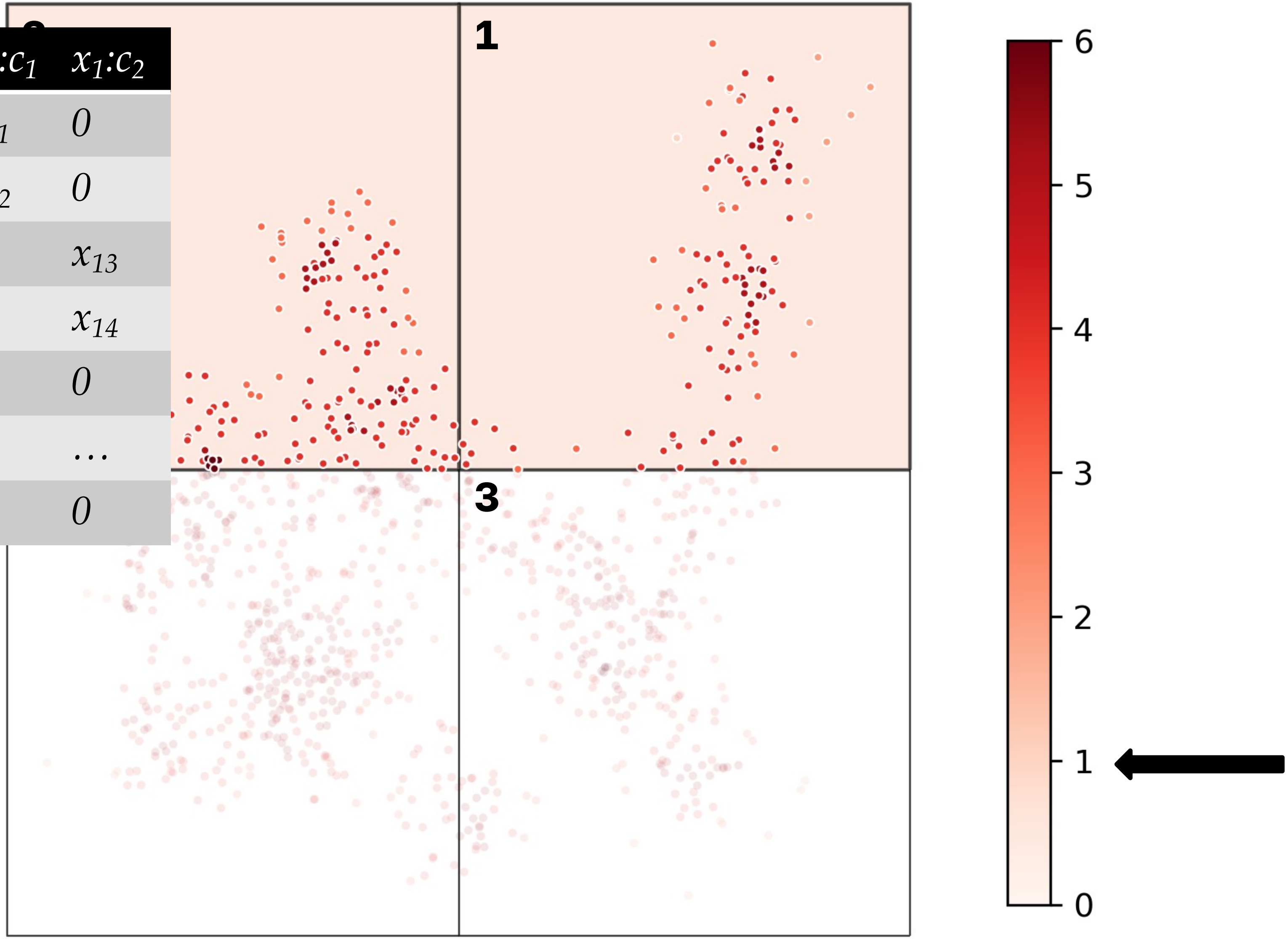
WALK DOWN THE DEPTH LADDER



WALK DOWN THE DEPTH LADDER

c	c_1	c_2
0	1	0
0	1	0
0	0	1
0	0	1
1	0	0
…	…	…
1	0	0

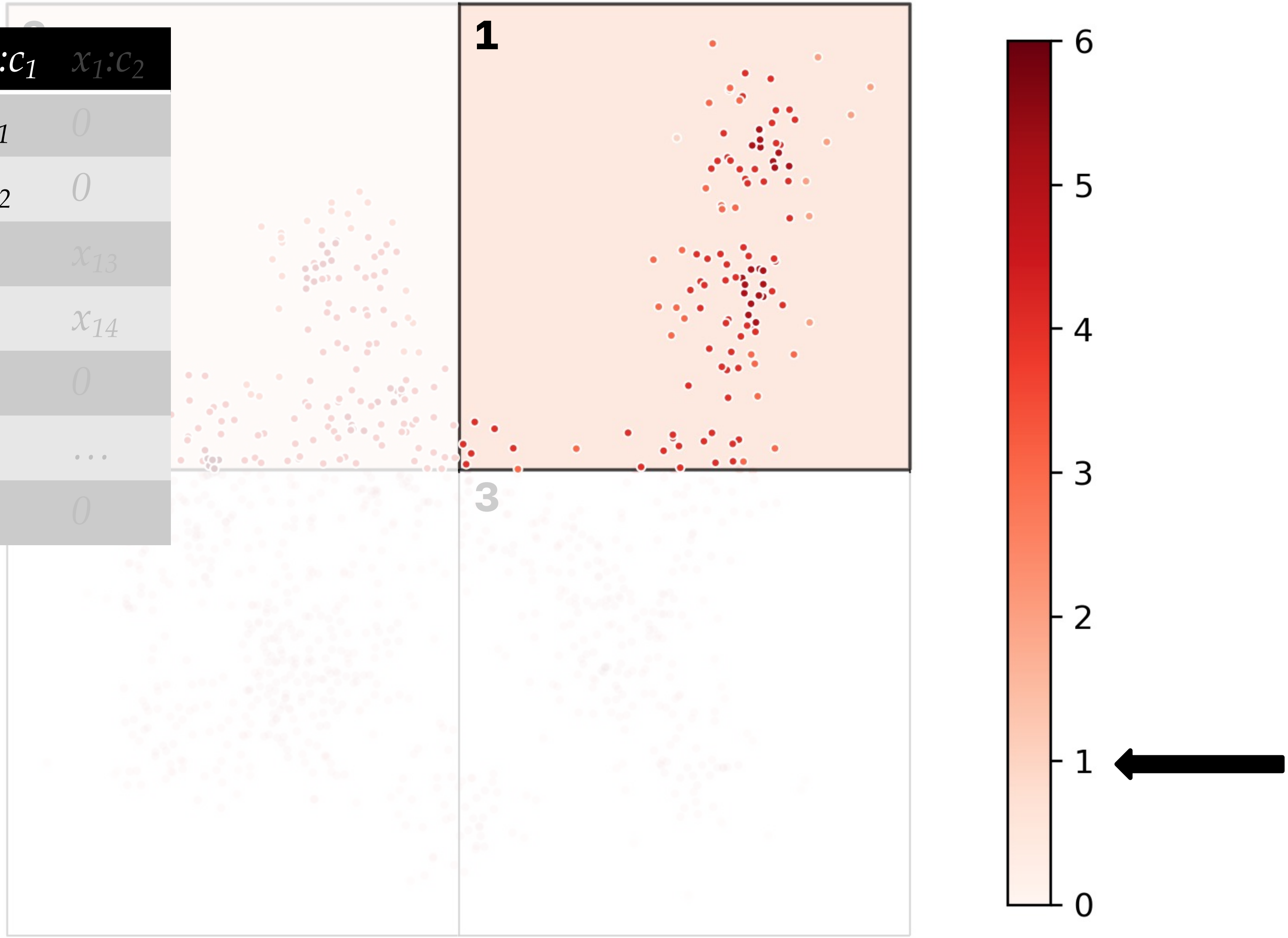
$x_1{:}c$	$x_1{:}c_1$	$x_1{:}c_2$
0	x_{11}	0
0	x_{12}	0
0	0	x_{13}
0	0	x_{14}
x_{15}	0	0
…	…	…
x_{1n}	0	0



WALK DOWN THE DEPTH LADDER

c	c_1	c_2
0	1	0
0	1	0
0	0	1
0	0	1
1	0	0
...	...	...
1	0	0

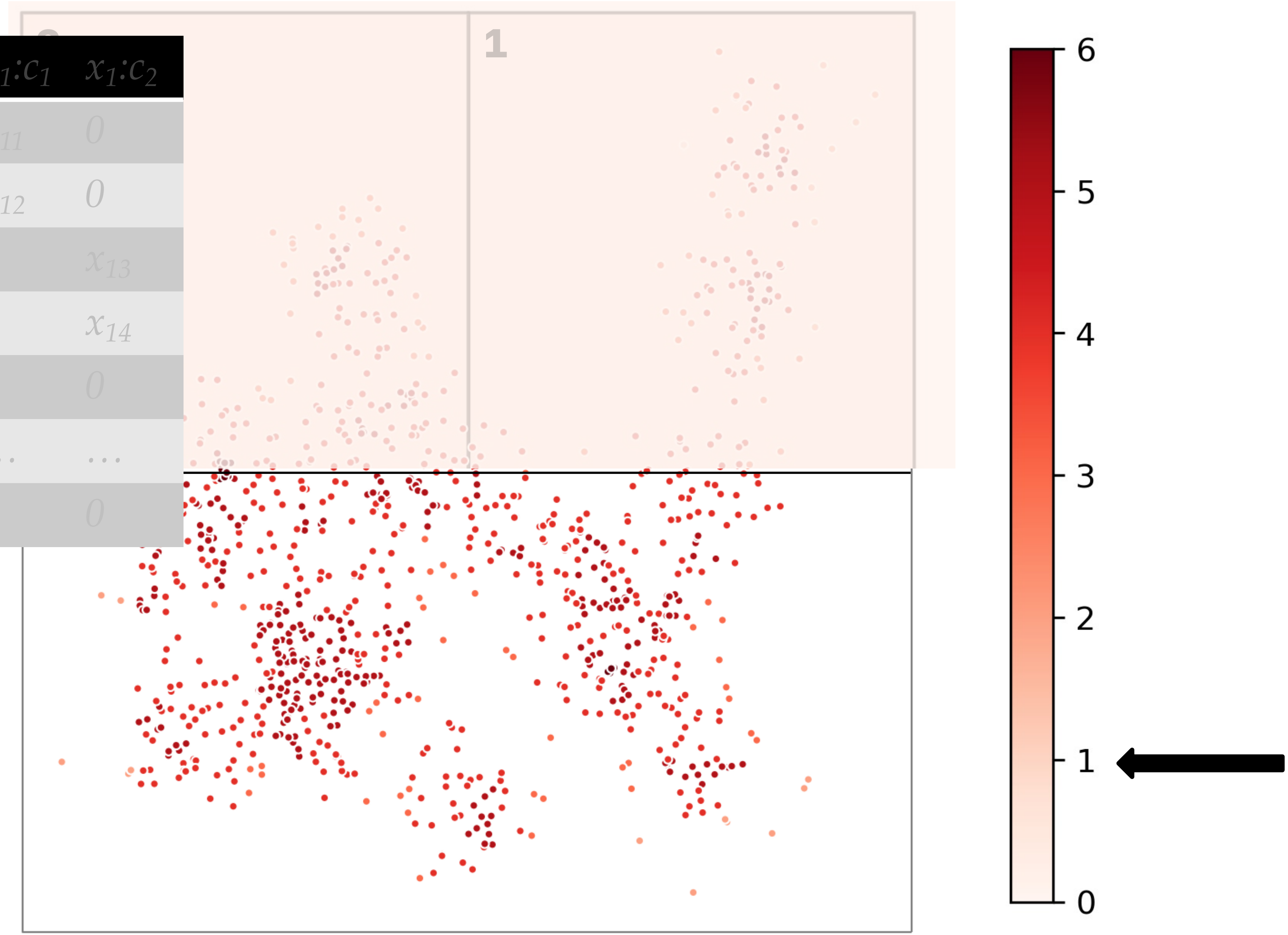
$x_1{:}c$	$x_1{:}c_1$	$x_1{:}c_2$
0	x_{11}	0
0	x_{12}	0
0	0	x_{13}
0	0	x_{14}
x_{15}	0	0
...	...	...
x_{1n}	0	0



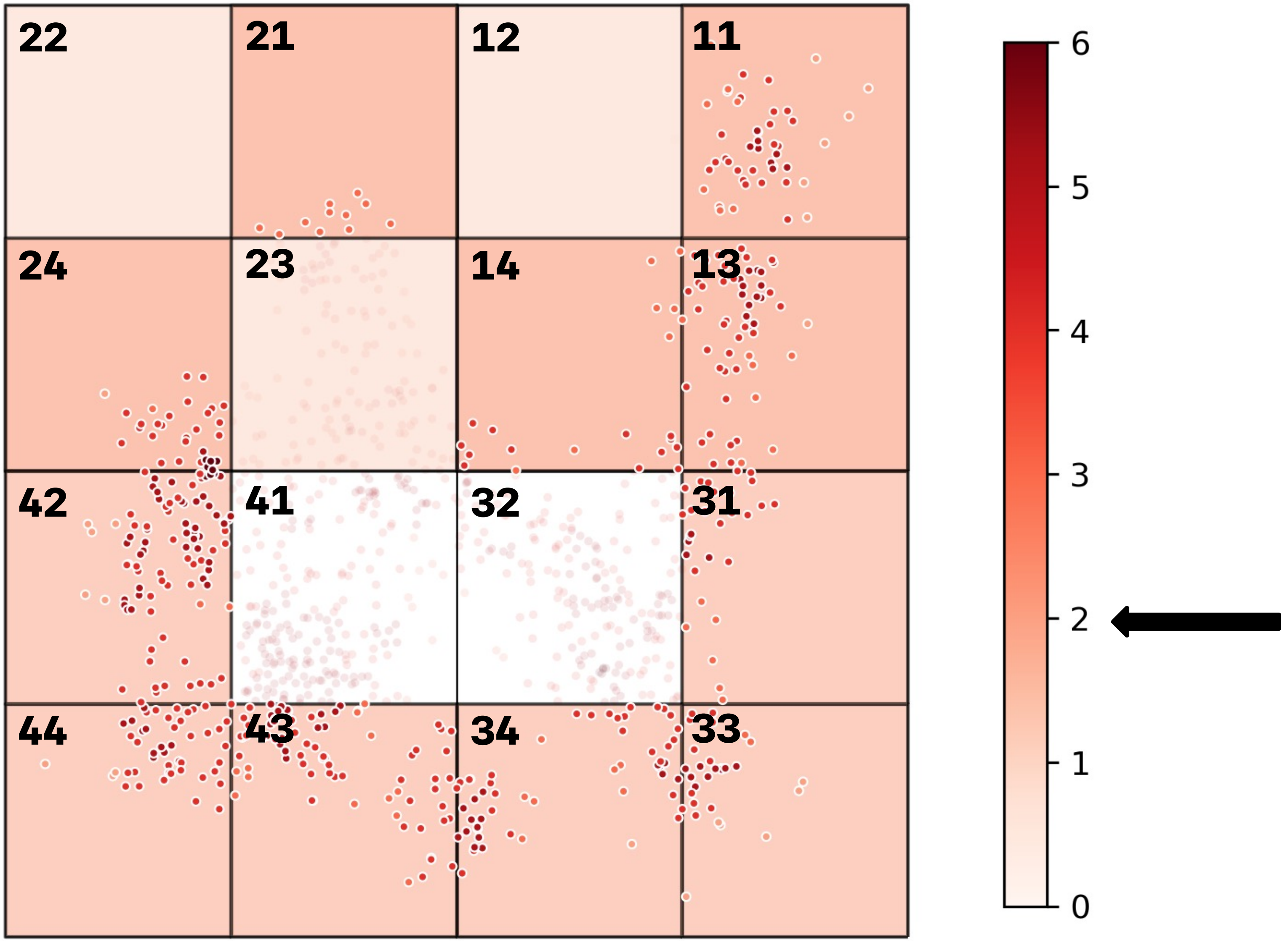
WALK DOWN THE DEPTH LADDER

c	c_1	c_2
0	1	0
0	1	0
0	0	1
0	0	1
1	0	0
…	…	…
1	0	0

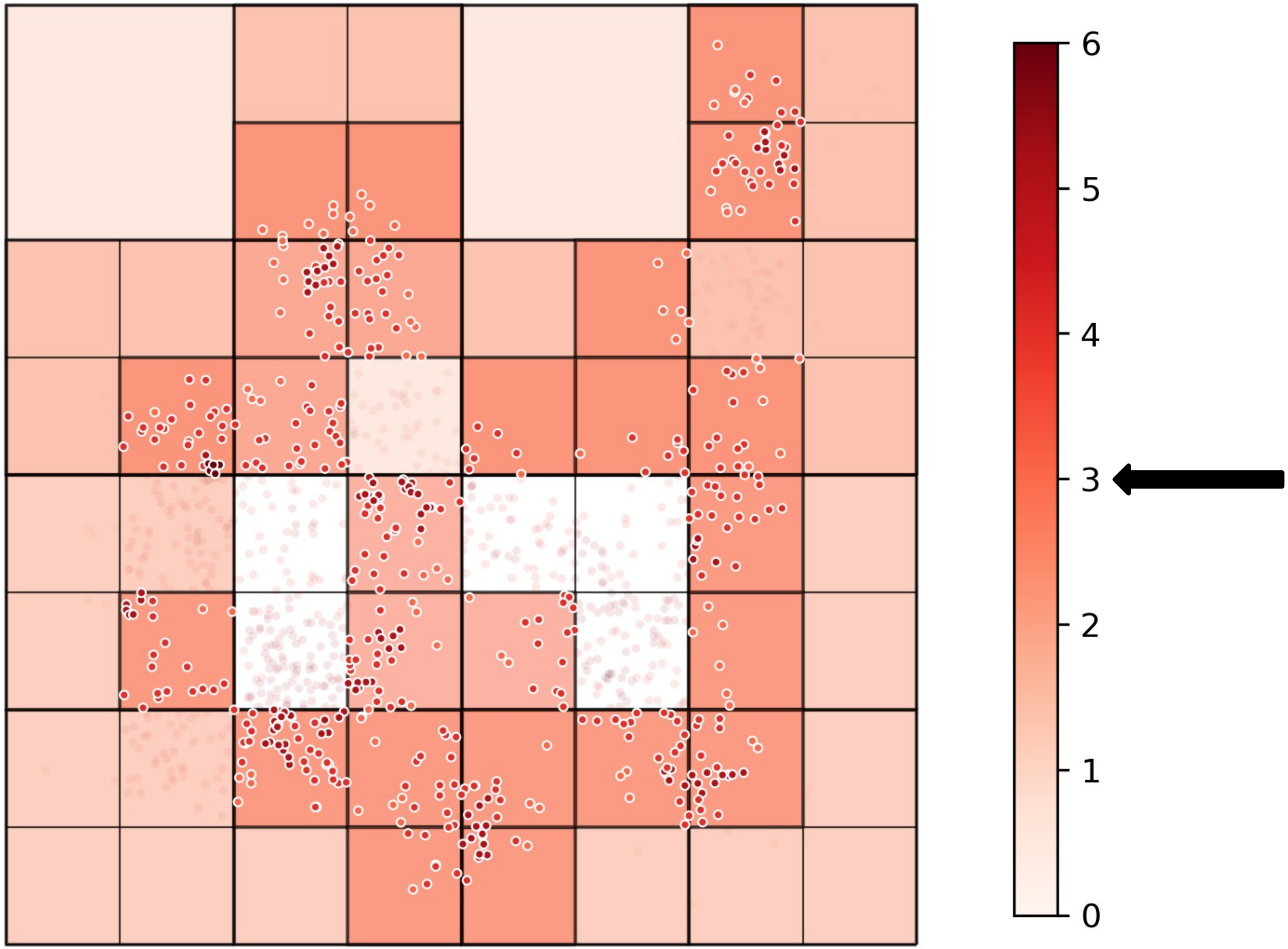
$x_1{:}c$	$x_1{:}c_1$	$x_1{:}c_2$
0	x_{11}	0
0	x_{12}	0
0	0	x_{13}
0	0	x_{14}
x_{15}	0	0
…	…	…
x_{1n}	0	0



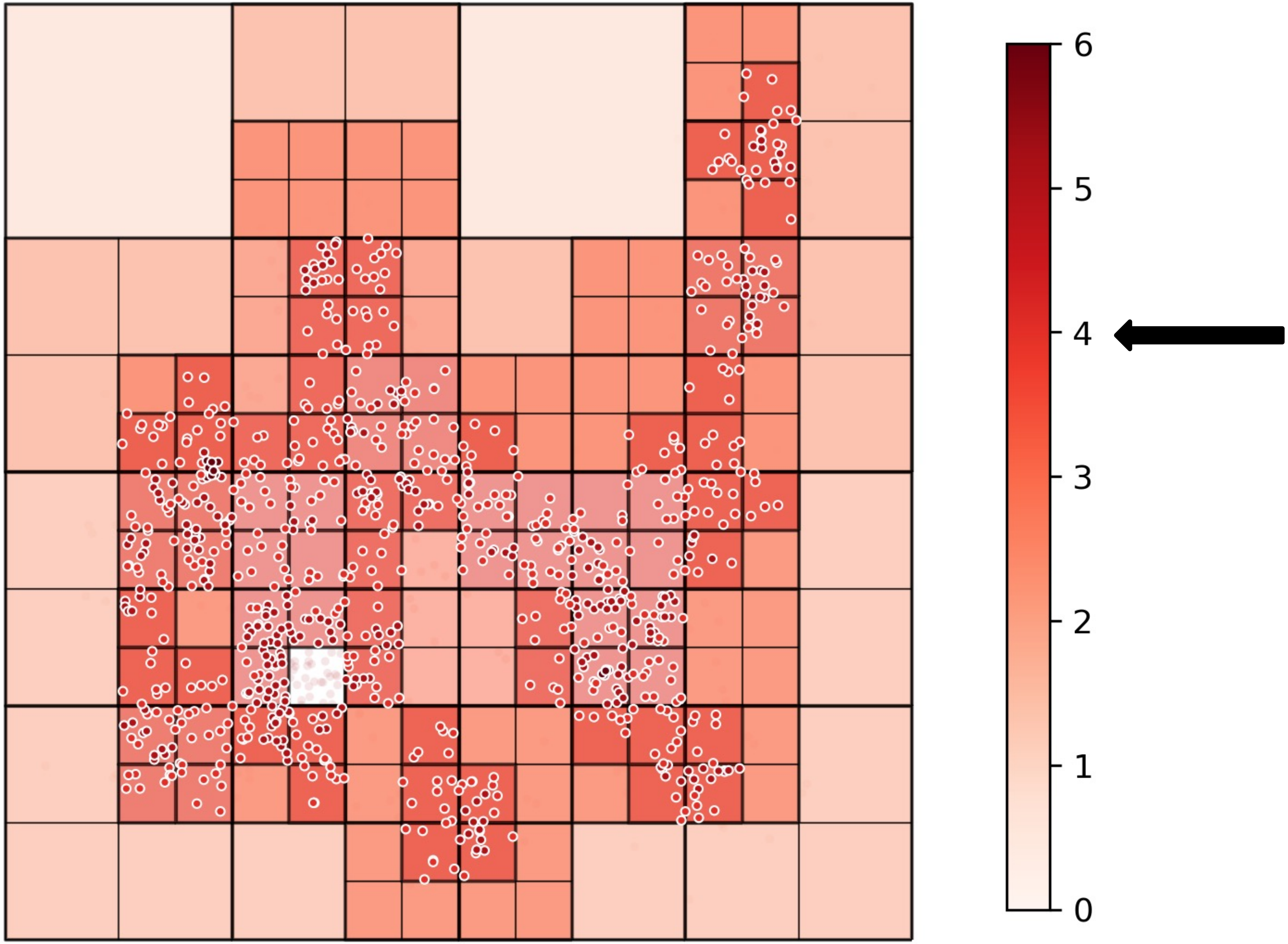
WALK DOWN THE DEPTH LADDER



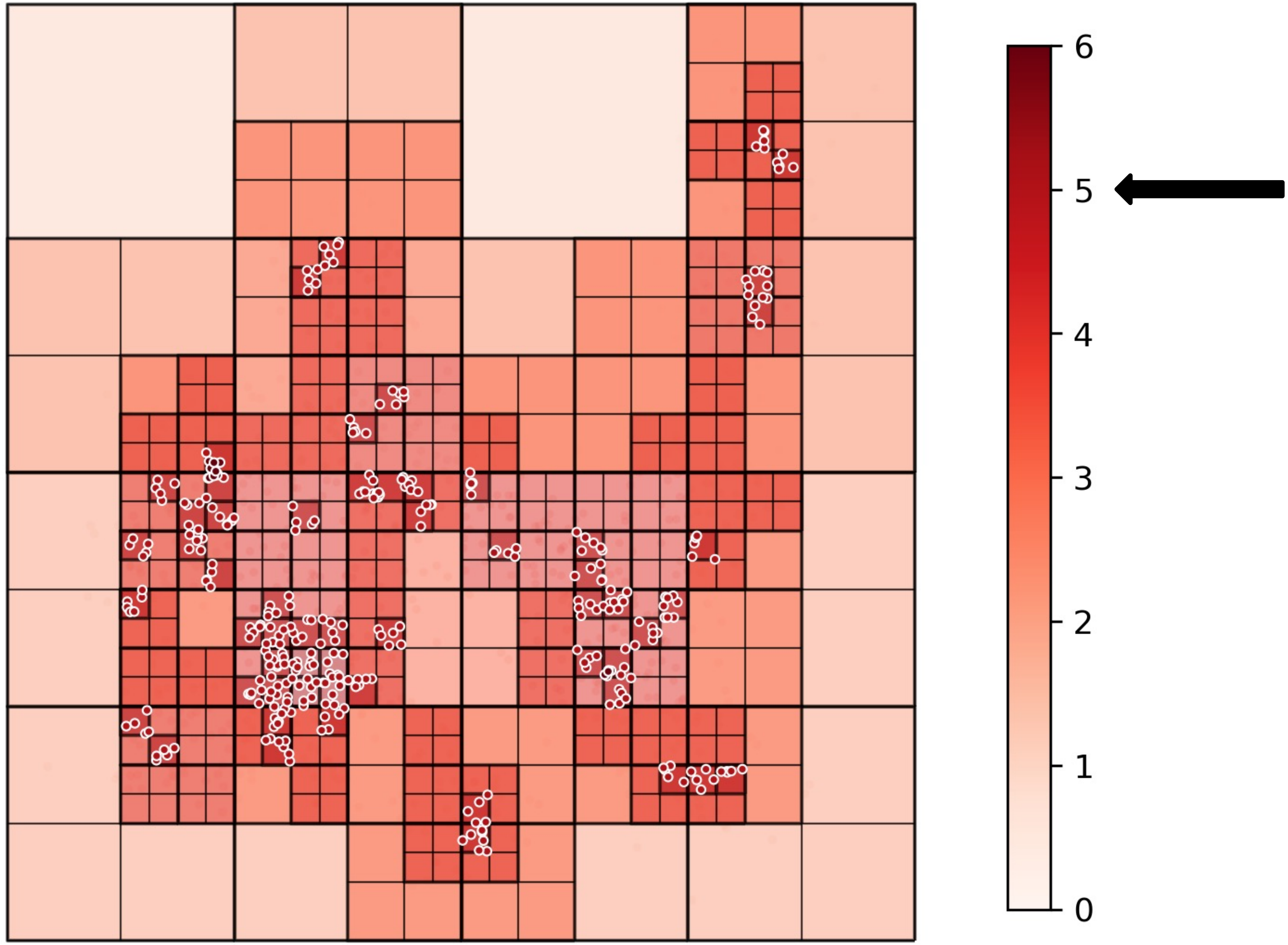
WALK DOWN THE DEPTH LADDER



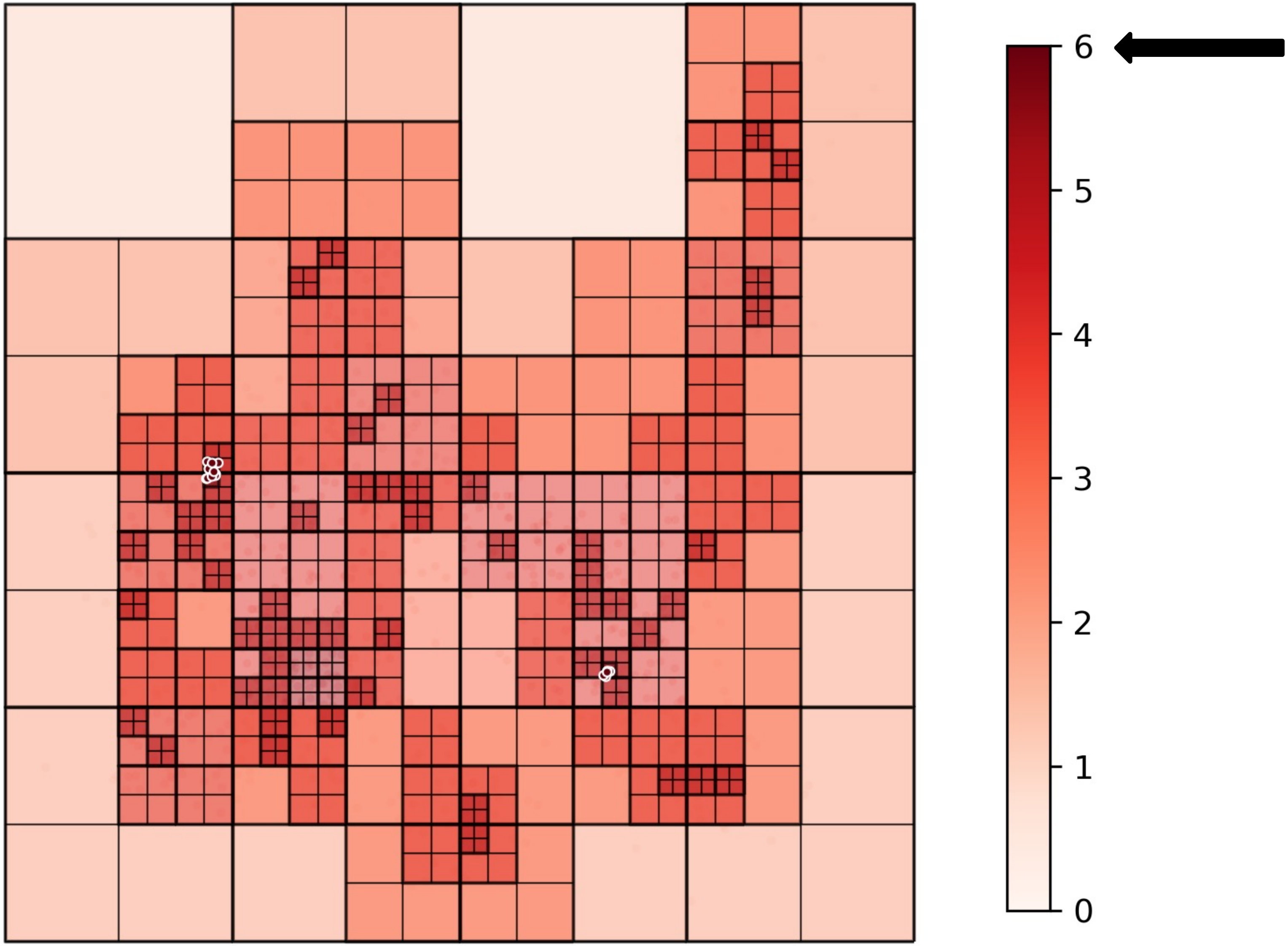
WALK DOWN THE DEPTH LADDER



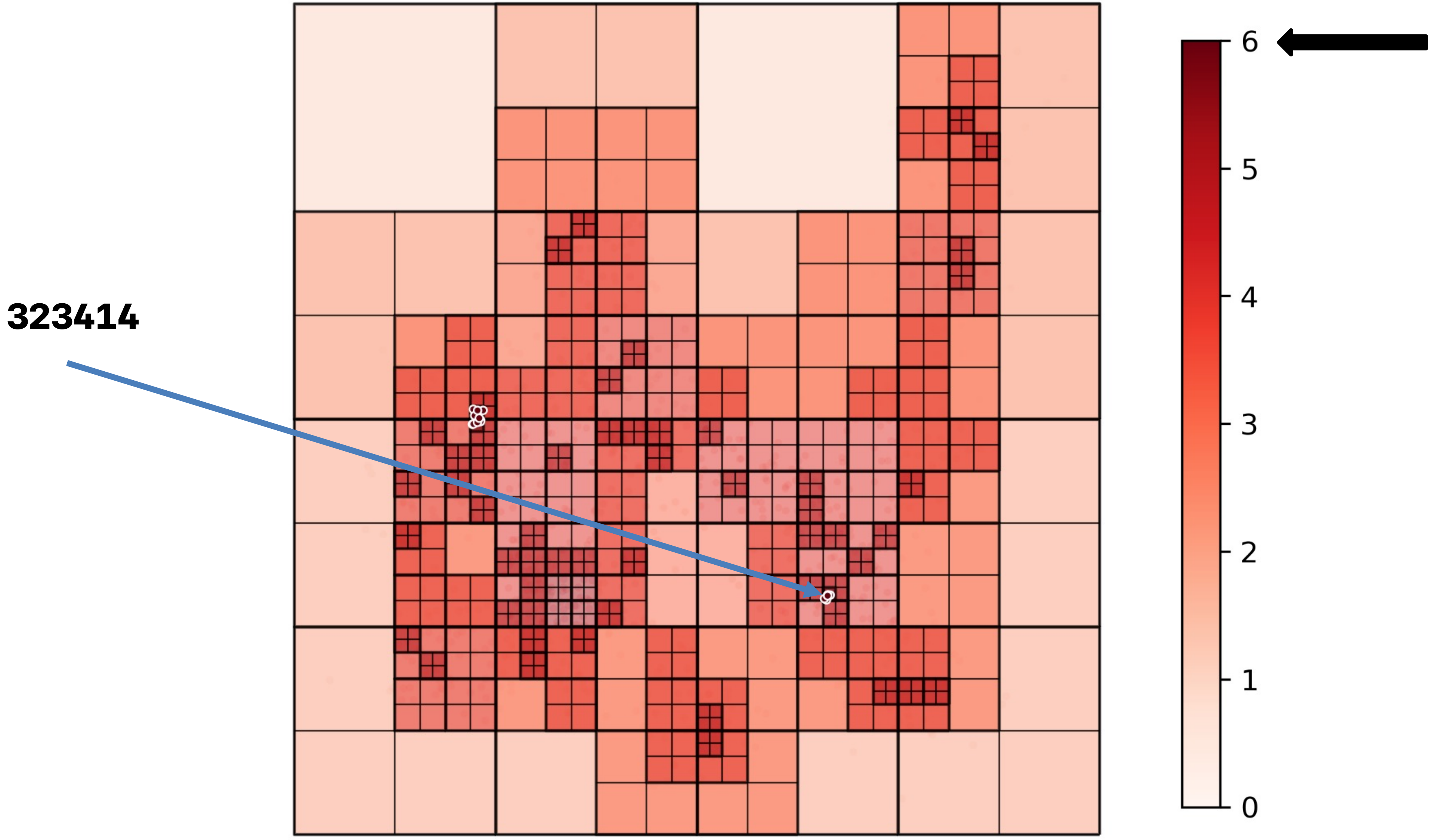
WALK DOWN THE DEPTH LADDER



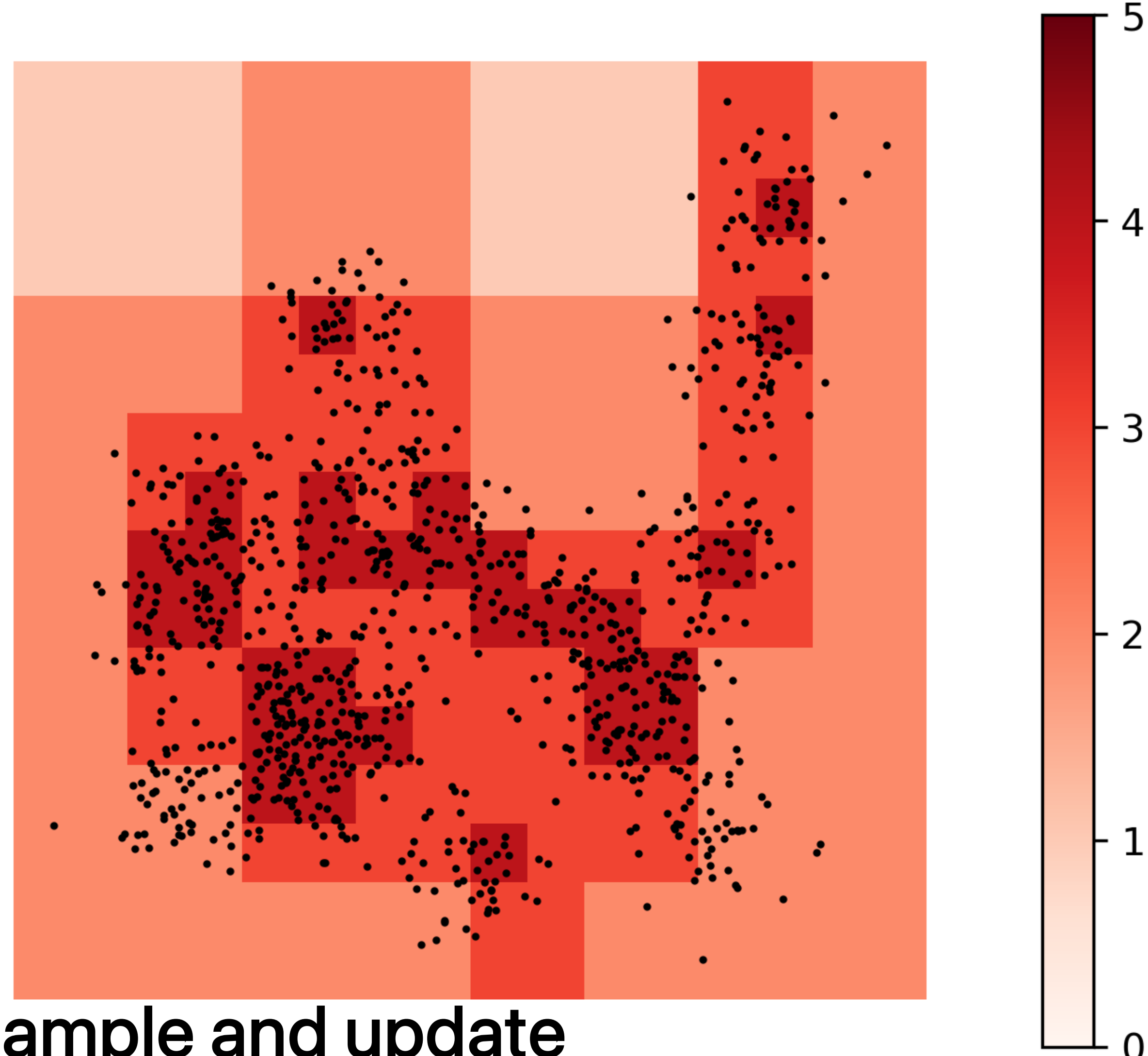
WALK DOWN THE DEPTH LADDER



WALK DOWN THE DEPTH LADDER



WALK DOWN THE DEPTH LADDER

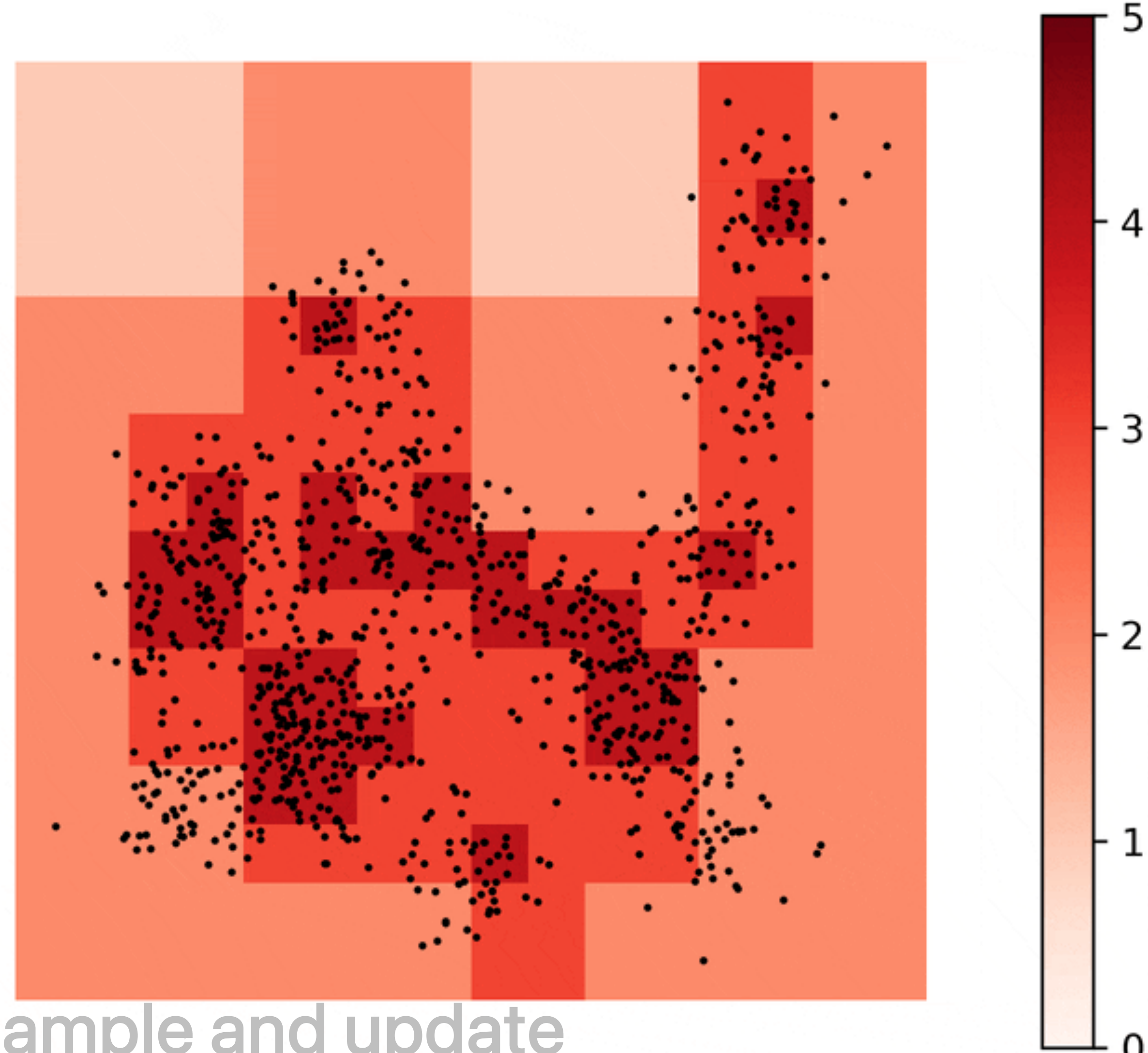


SuperQT

- stable in CV
- totally ordered
- simple out of sample and update

QUADTREES ARE FAIRLY STABLE IN CV

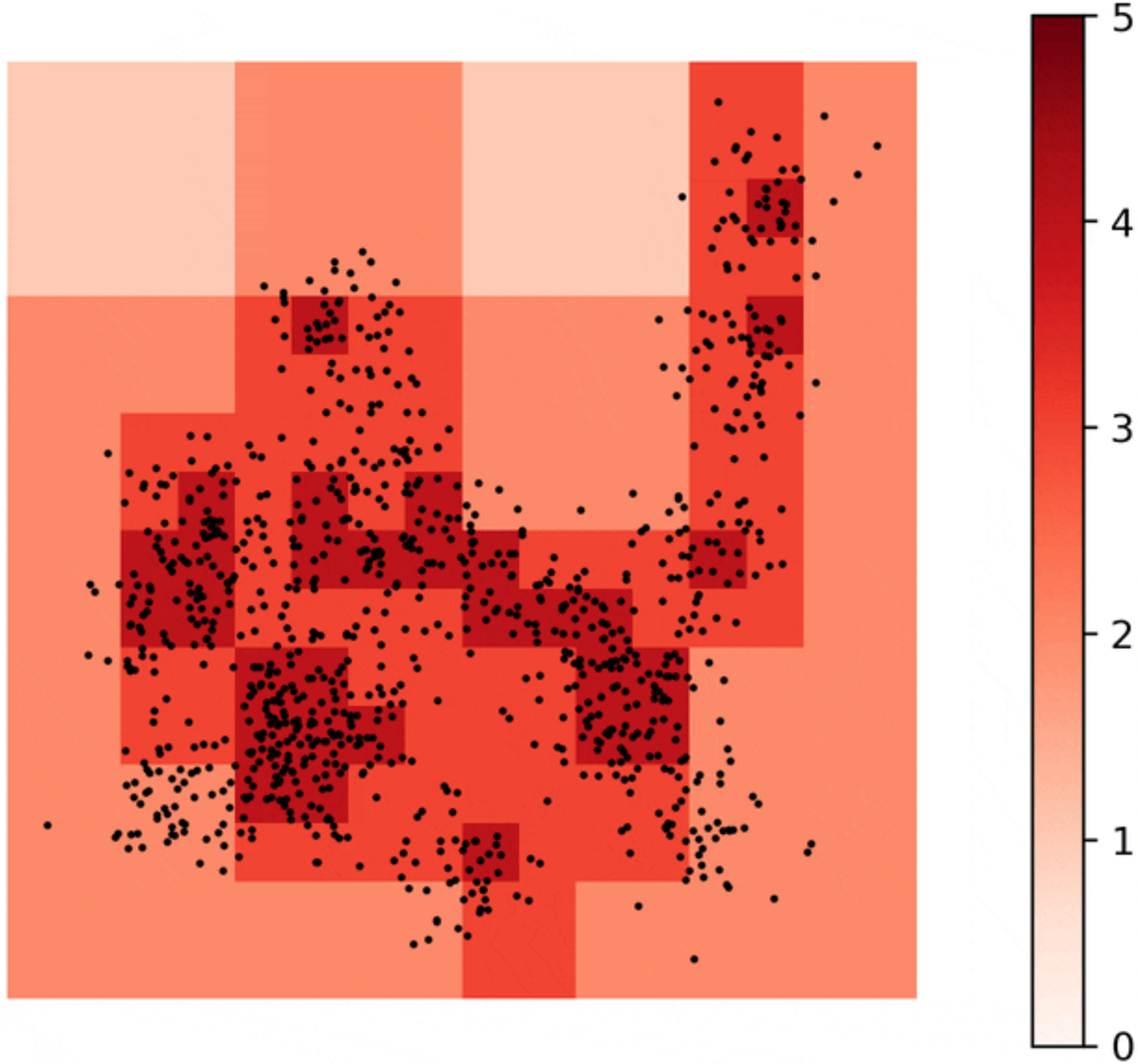
SuperQT
- stable in CV
- totally ordered
- simple out of sample and update



QUADTREES ARE FAIRLY STABLE IN CV

TOTALLY ORDERED

An observation will nest exactly into one cell at any given depth, and fit depth is a function of both fit and density, and big jumps in depth are rare.



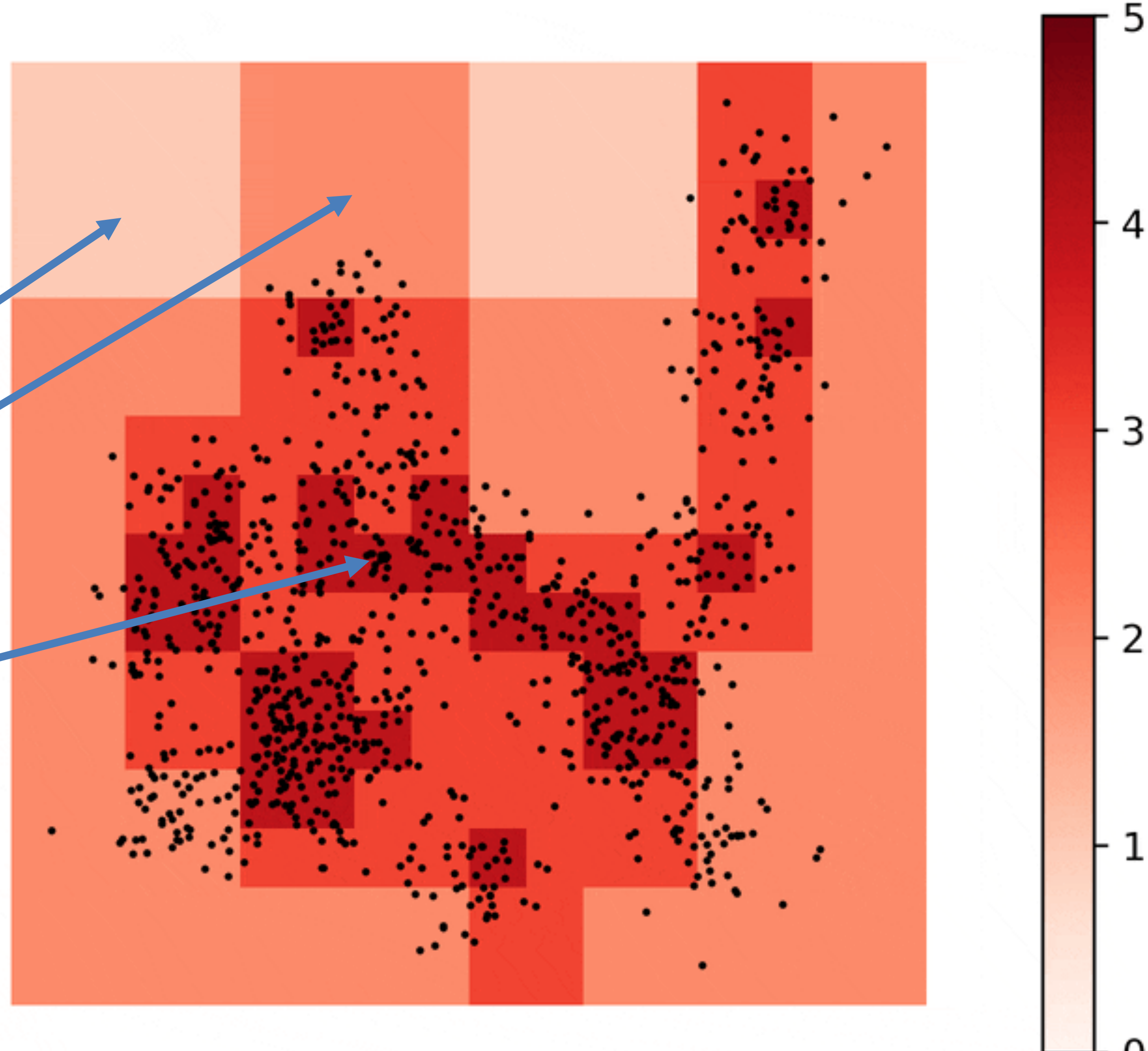
QUADTREES ARE FAIRLY STABLE IN CV

OUT-OF-BAG
For interpolating predictions *within* frame, we always have a model

Always **2**

2 or **21**

4112
or
411

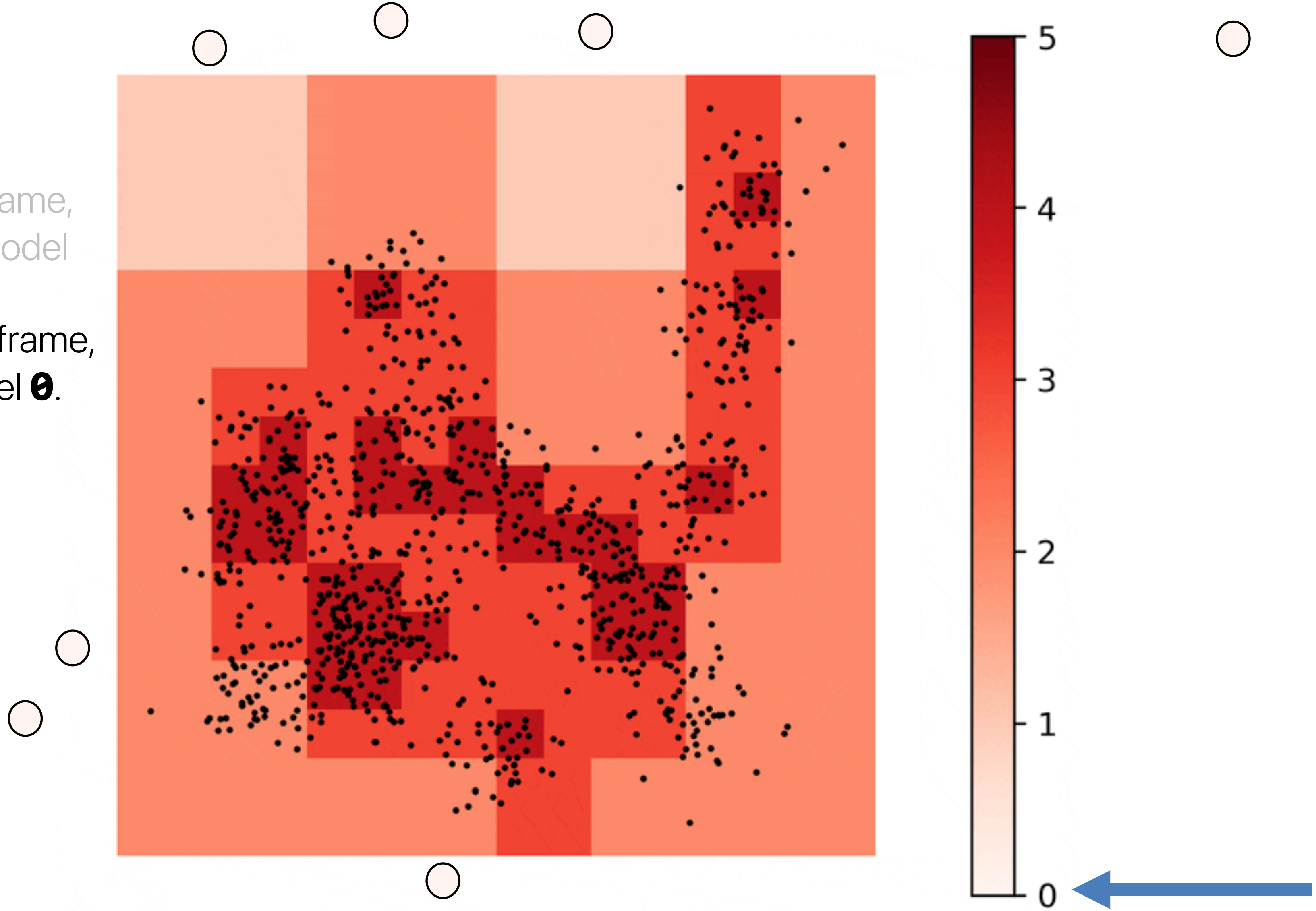


QUADTREES ARE FAIRLY STABLE IN CV

OUT-OF-BAG

For interpolating predictions *within* frame, we always have a model

For *extrapolating* predictions outside frame, we always use model **0**.



QUADTREES ARE FAIRLY STABLE IN CV

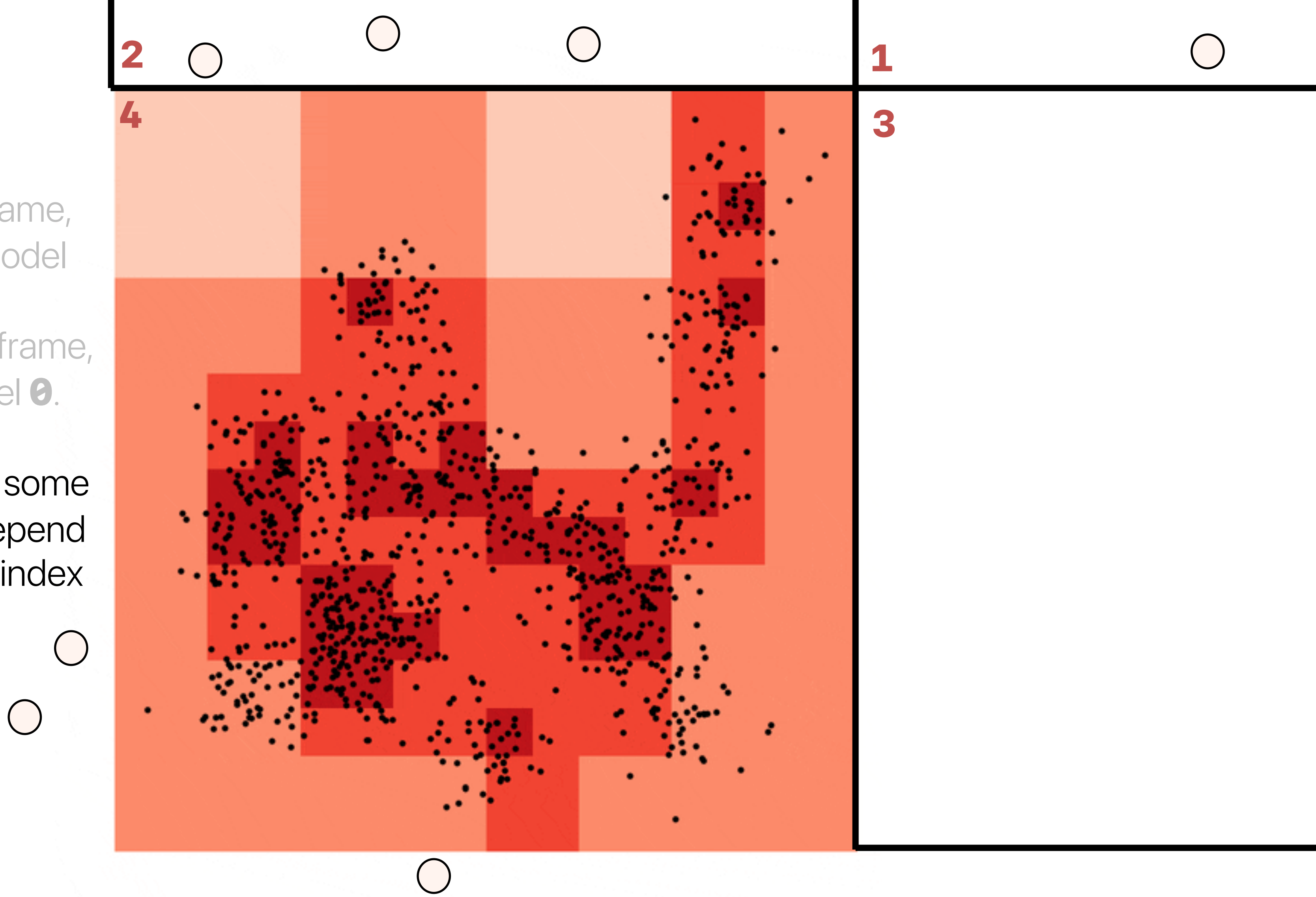
OUT-OF-BAG

For interpolating predictions *within* frame, we always have a model

For *extrapolating* predictions outside frame, we always use model **0**.

UPDATING

Treat 0 as a child of some larger frame and prepend all cells with its new index



QUADTREES ARE FAIRLY STABLE IN CV

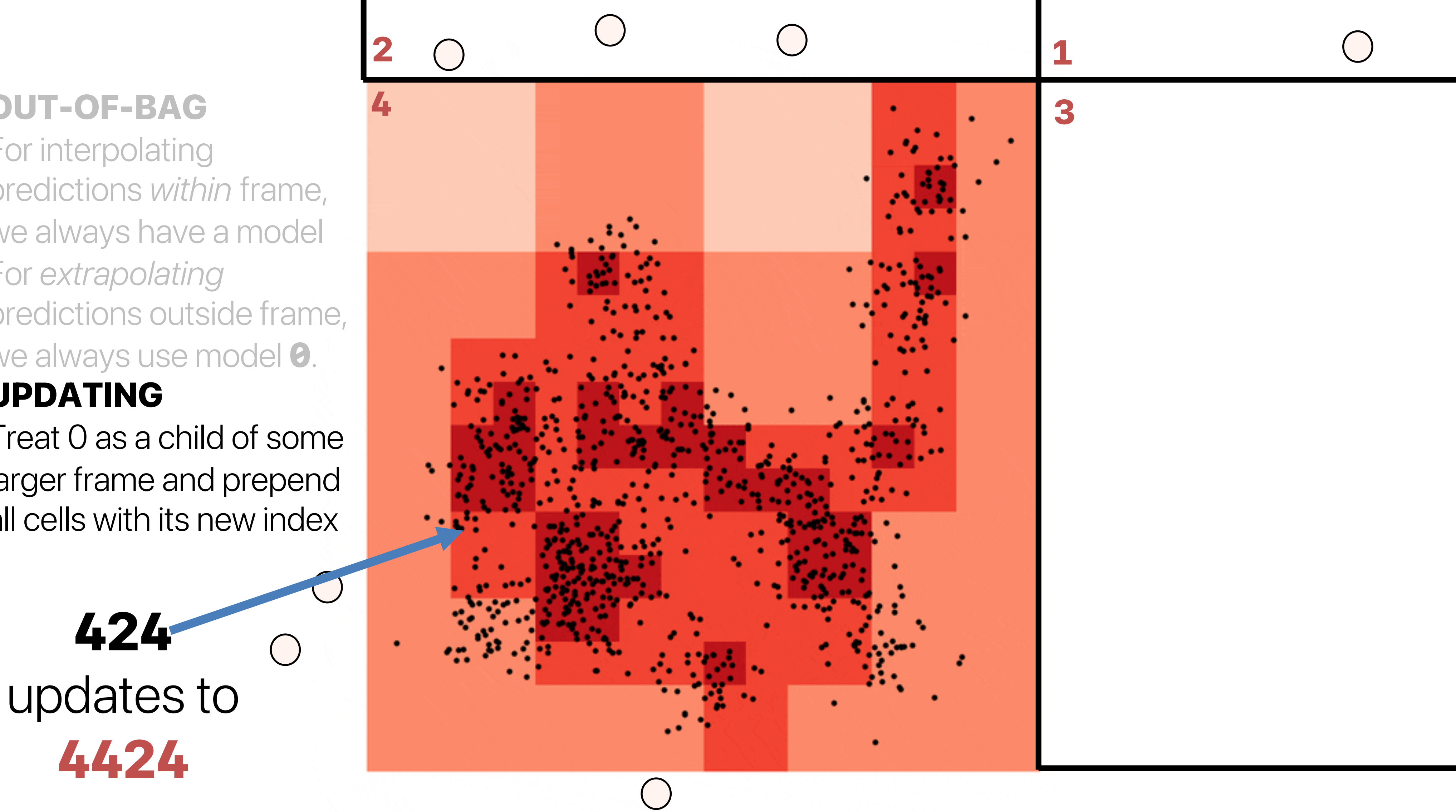
OUT-OF-BAG

For interpolating predictions *within* frame, we always have a model

For *extrapolating* predictions outside frame, we always use model **0**.

UPDATING

Treat 0 as a child of some larger frame and prepend all cells with its new index



QUADTREES ARE FAIRLY STABLE IN CV

0. INITIALIZE: enqueue a root cell that predicts y ~ X, the “global” model

1. SPLIT: pop the first cell from our queue and split it into four. For each split:

1-1. fit a submodel predicting only outcomes in that cell using data in that cell.

1-2. update predictions for the whole map

1-3. if predictions are **significantly better**, return the split to the queue & go to 1,
otherwise assign all observations in that split the parent’s label & discard the split.

2. PRUNE: fit a model with all current feature X label interactions (one hot).

2-1. if all terms **are significant**, go to 3,
otherwise find the deepest non-significant term, *d**

2-2. for all non-significant terms at *d**

2-2-1. merge the non-significant term into its parent leaf’s term

2-2-2. remove the non-significant term itself and go to 2-1.

3. FINALIZE: Fit a final model on the final design matrix from 2.

THE SUPERVISED QUADTREE

0. INITIALIZE: enqueue a root cell that predicts y ~ X, the "global" model
1. SPLIT: pop the first cell from our queue and split it into four. For each split:
1-1. fit a submodel predicting only outcomes in that cell using data in that cell.
1-2. update predictions for the whole map
1-3. if predictions are **significantly better**, return the split to the queue & go to 1,
otherwise assign all observations in that split the parent's label & discard the split.
2. PRUNE: fit a model with all current feature X label interactions (one hot).
2-1. if all terms **are significant**, go to 3,
otherwise find the deepest non-significant term, *d**
2-2. for all non-significant terms at *d**
2-2-1. merge the non-significant term into its parent leaf's term
2-2-2. remove the non-significant term itself and go to 2-1.
3. FINALIZE: Fit a final model on the final design matrix from 2.

Note that this is *model agnostic*: can use any learner, classification or regression.
- GLMs can split using LRT (χ^2) prune using Wald (t) tests:
- Always can split using score improvement ($\Delta MSE > \varepsilon$)
prune using permutation feature importance ($X_p * I(leaf)$ in best q% of unpruned features)

THE SUPERVISED QUADTREE

CLUSTERING REGRESSION

jointly solving clustering and regression

GEOGRAPHICAL CLUSTER-REG

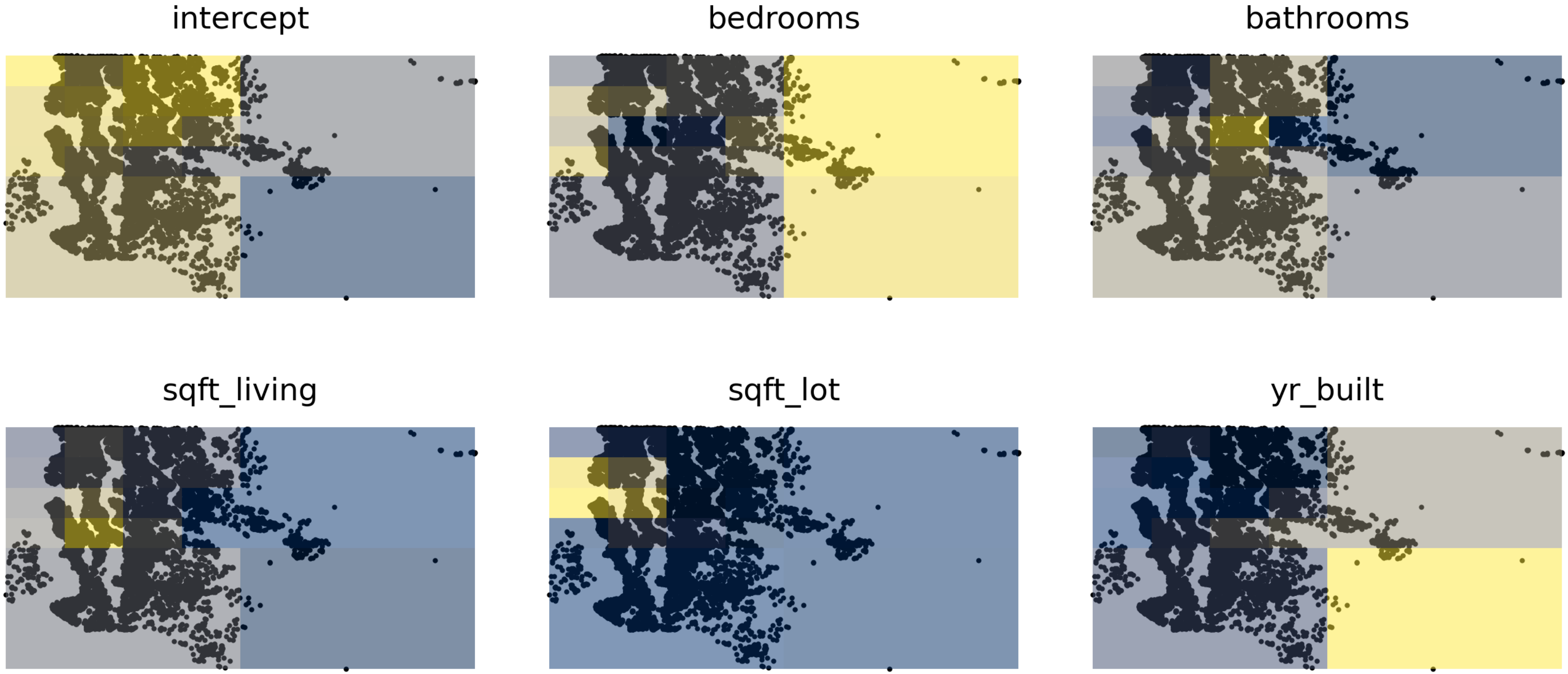
jointly solving clustering and regression

UNDERSTANDING QUADTREES

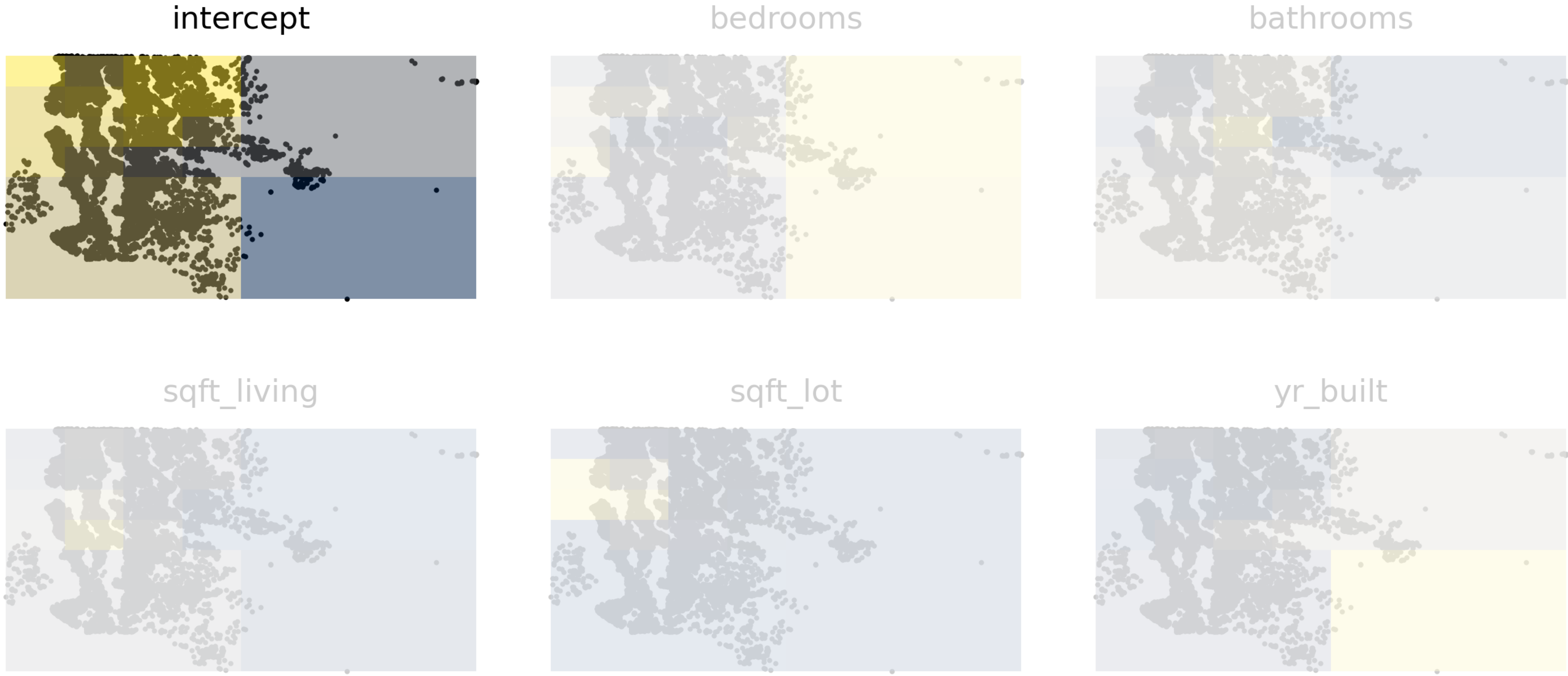
spatial splits for spatial fits

APPLYING QUADTREE REGRESSION

model comparison & basic metrics

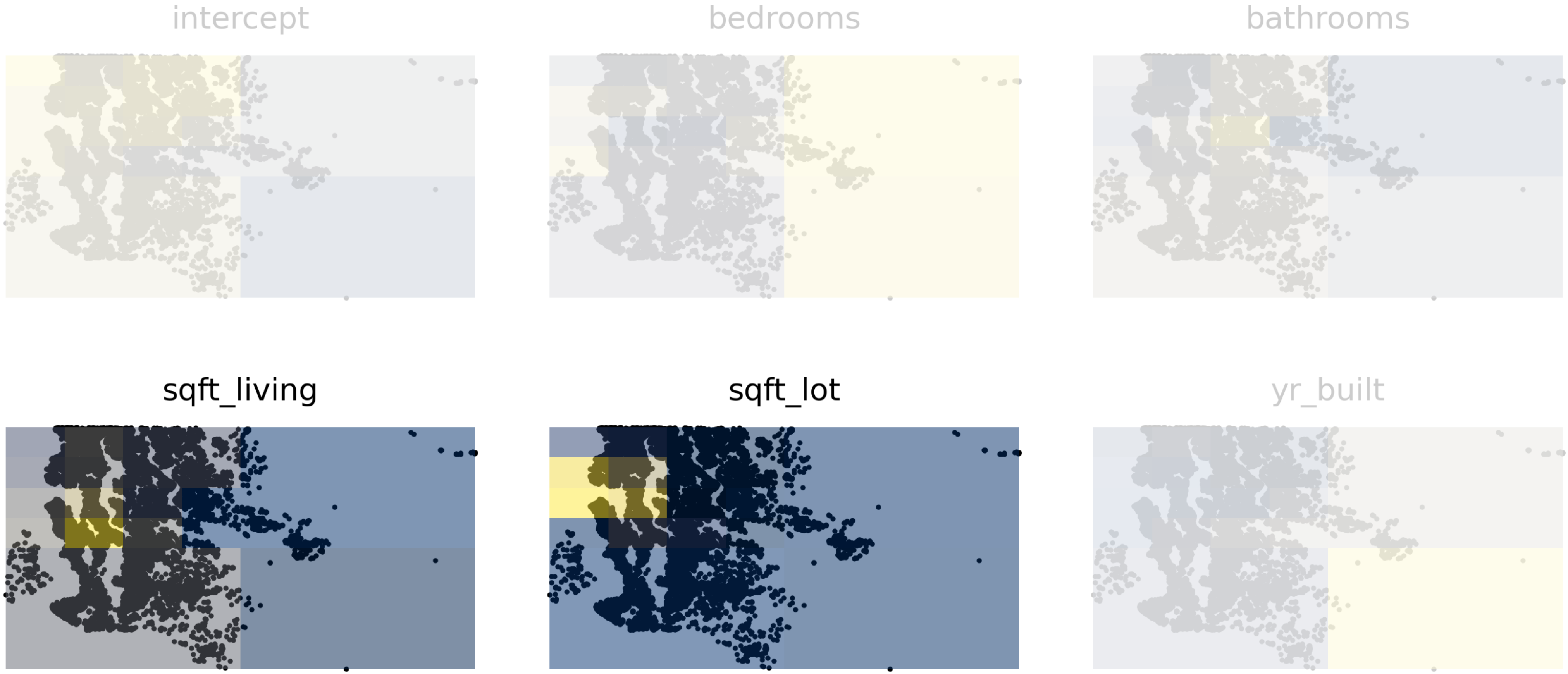


COEFFICIENT SURFACES LOOK LIKE THIS



Prices in general are expensive towards the coast, also in the start-upish areas in the north east.

COEFFICIENT SURFACES LOOK LIKE THIS



Space is at a premium, with living space slightly more expensive south-center.

COEFFICIENT SURFACES LOOK LIKE THIS

PLACE	METHOD	MSE	R^2	TIME
Baltimore	OLS	.16	.49	Sub-seconds
n=211	Skater 15 Regimes	.02	.93	Seconds
	MGWR	.01	.95	Seconds
	SuperQT (no prune)	.04 (.02)	.89 (.94)	Seconds
King's Co.	OLS	.13	.53	Seconds
n=31k	Skater 15 Regimes	.05	.811	Hours
	MGWR	DNF	DNF	DNF
	SuperQT (no prune)	.06 (.06)	.78 (.78)	Minutes

COMPARISONS

PLACE	METHOD	MSE	R^2	TIME
Baltimore	OLS	.16	.49	Sub-seconds
n=211	Skater 15 Regimes	.02	.93	Seconds
	MGWR	.01	.95	Seconds
	SuperQT (no prune)	.04 (.02)	.89 (.94)	Seconds
King's Co.	OLS	.13	.53	Seconds
n=31k	Skater 15 Regimes	.05	.811	Hours
	MGWR	DNF	DNF	DNF
	SuperQT (no prune)	.06 (.06)	.78 (.78)	Minutes

Small n: usually need eps+perm (lrt+wald dof issues)

Big n: both lrt+wald and eps+perm are useful

COMPARISONS

PLACE	METHOD	MSE	R^2	TIME
Baltimore	OLS	.16	.49	Sub-seconds
n=211	Skater 15 Regimes	.02	.93	Seconds
	MGWR	.01	.95	Seconds
	SuperQT (no prune)	.04 (.02)	.89 (.94)	Seconds
King's Co.	OLS	.13	.53	Seconds
n=31k	Skater 15 Regimes	.05	.811	Hours
	MGWR	DNF	DNF	DNF
	SuperQT (no prune)	.06 (.06)	.78 (.78)	Minutes

COMPETITIVE EVEN AFTER PRUNING

PLACE	METHOD	MSE	R^2	TIME
Baltimore	OLS	.16	.49	Seconds
n=211	Skater 15 Regimes	.02	.93	Seconds
	SuperQT (no prune)	.04 (.02)	.89 (.94)	Seconds
King's Co.	OLS	.13	.53	Seconds
n=31k	Skater 15 Regimes	.05	.811	Hours
	MGWR	DNF	DNF	DNF
	SuperQT (no prune)	.06 (.06)	.78 (.78)	Minute

Pruning is especially powerful in big data:
~90 feature x leaf interactions in Skater & SuperQT
~60 post pruning w/ very small change to fit (for both)

COMPETITIVE EVEN AFTER PRUNING

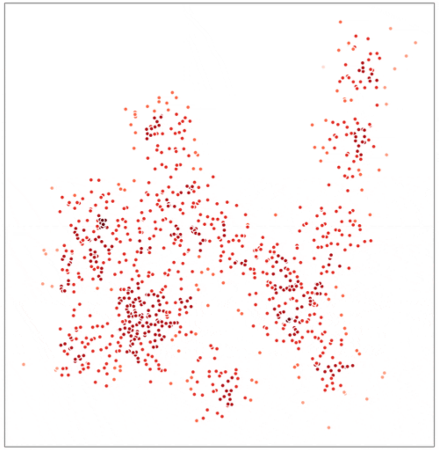
MIT-Licensed

High performance w/ `scipy.sparse`

Available in PySAL by December 2023

SUPERVISED QUADTREES:

a new metalearner for local data science



LEVI JOHN WOLF levi.john.wolf@bristol.ac.uk