MESHED GPs

FOR EFFICIENT BAYESIAN INFERENCE OF **BIG DATA** SPATIAL REGRESSION MODELS

Michele Peruzzi

Duke University

BIG SPATIAL DATA

SOURCES

- Satellite images
- Remote sensing
- Crowd-sourced
- Other sources

DATA

- Vegetation/soil indices
- Air quality/climate
- Species distribution
- Others

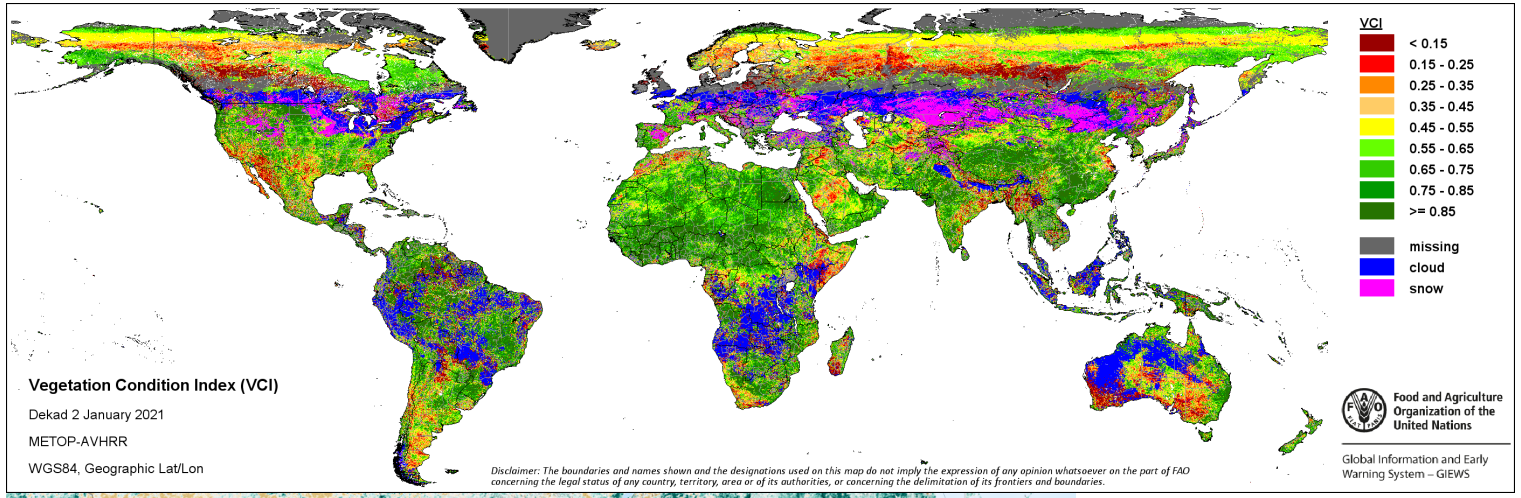
MEASUREMENT

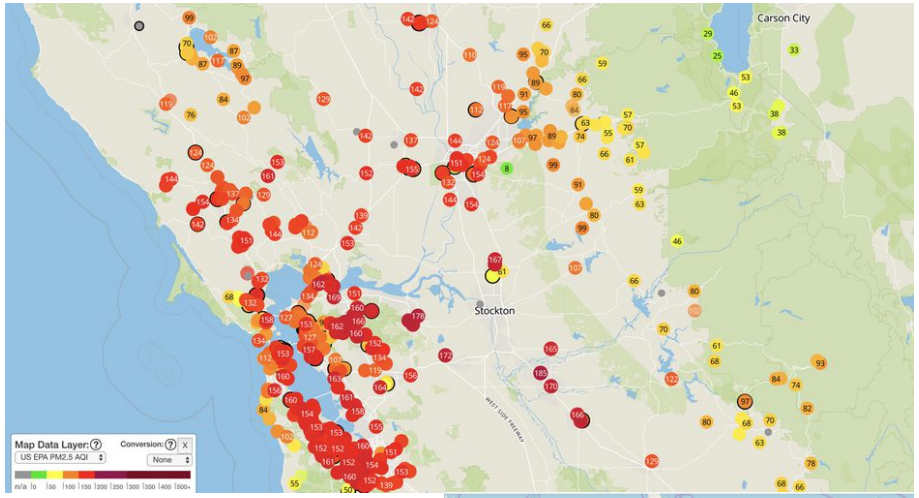
- Indexed by spatial coordinates (longitude, latitude)
- Possibly repeated in time

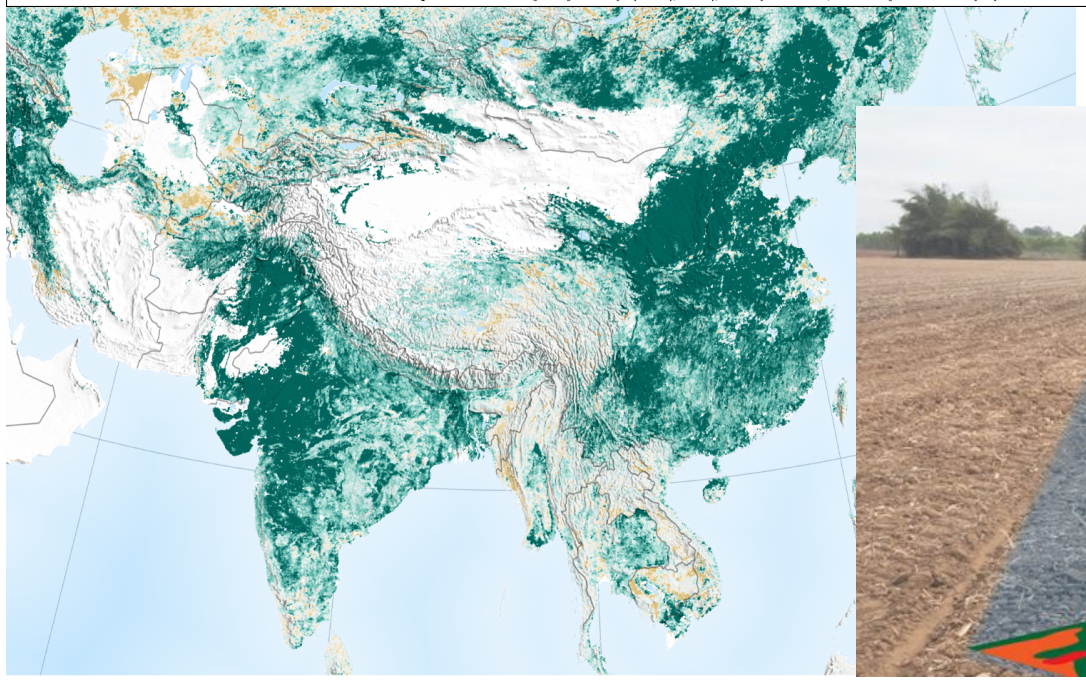
GOALS

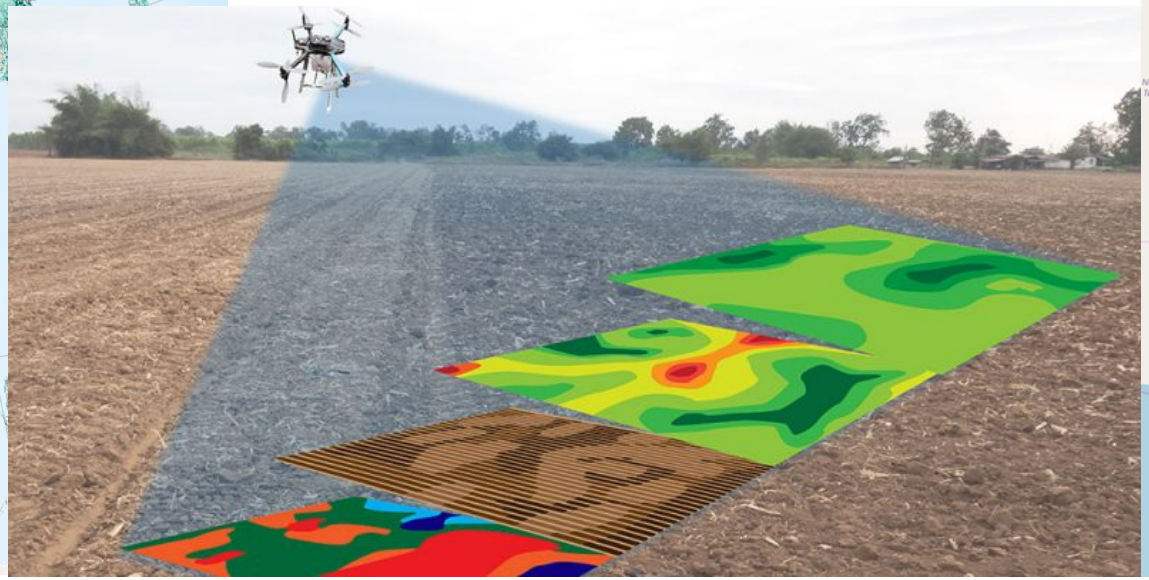
- Model estimation
- Prediction at new locations
- Uncertainty quantification

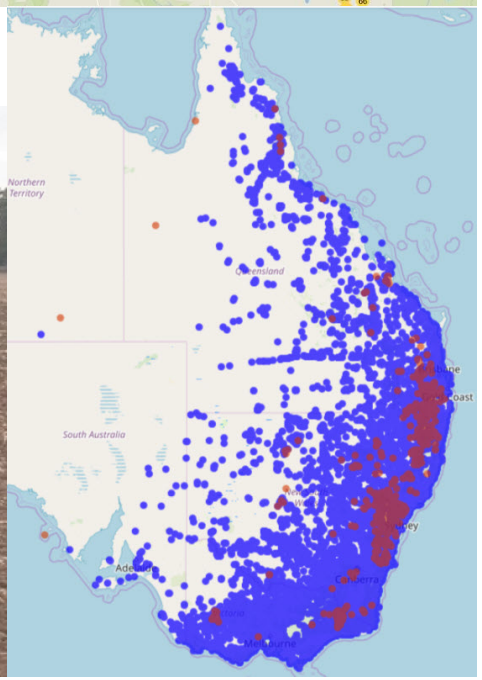
BIG SPATIAL DATA











SPATIAL REGRESSIONS

Basic univariate linear model with spatial random effects.

$$y(\ell_i) = x(\ell_i)^\top \beta + w(\ell_i) + \varepsilon(\ell_i), \qquad \varepsilon(\ell_i) \sim N(0, \tau^2)$$

Spatial locations: $\ell_i \in D \subset \mathbb{R}^d \qquad i = 1, \ldots, n$ $n > 10^5$

Spatial random effects: $w \sim GP\big(0, C(\cdot,\cdot;\theta)\big)$ encode spatial dependence

Covariance function/Kernel $C(\cdot,\cdot;\theta) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ parametrized by θ

SPATIAL REGRESSIONS

Basic univariate linear model with spatial random effects.

$$y = X\beta + w + \varepsilon, \qquad \varepsilon \sim N(0, \tau^2 I_n)$$

Where now we stack into vectors, e.g. $y = \left(y(\ell_1), \ldots y(\ell_n) \right)^{\mathsf{T}}$

Outcomes are conditionally independent *given* the spatial random effects, but integrating out we get:

$$y = X\beta + u, \qquad u \sim N(0, C_\theta + \tau^2 I_n)$$

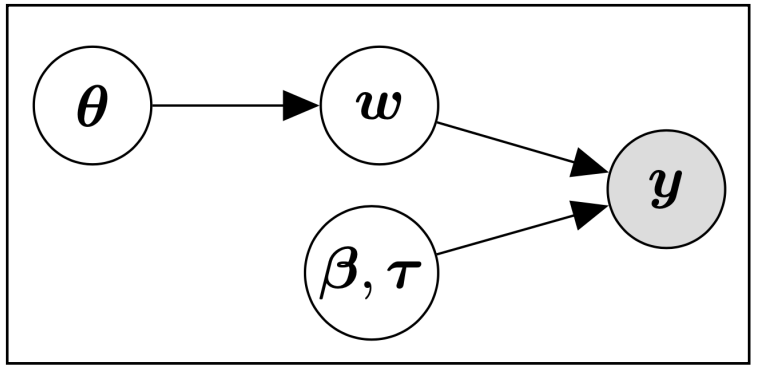
BAYESIAN HIERARCHICAL MODEL

Basic univariate linear model with spatial random effects.

$$y = X\beta + w + \varepsilon, \qquad \varepsilon \sim N(0, \tau^2 I_n)$$

$$w \sim GP(0, C_\theta)$$

- Assign prior distributions $\pi(\cdot)$ for β, τ^2, θ
- Represent the model as a directed acyclic graph (DAG):



✓ The posterior distribution is

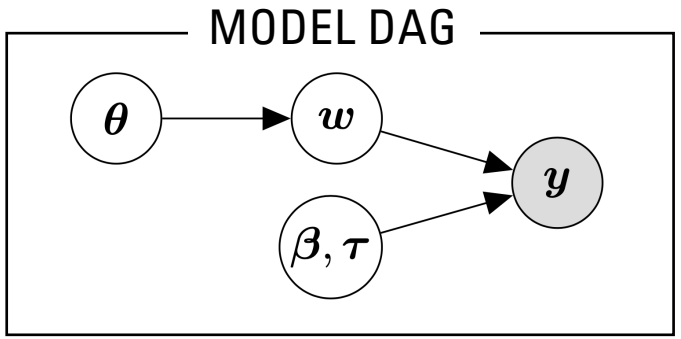
$$\pi(w, \beta, \tau^2, \theta \mid y) \propto p(y \mid \ldots)\pi(w \mid \theta)\pi(\theta)\pi(\beta, \tau^2)$$

COMPUTING THE POSTERIOR

Basic univariate linear model with spatial random effects.

$$y = X\beta + w + \varepsilon, \qquad \varepsilon \sim N(0, \tau^2 I_n)$$

$$w \sim GP(0, C_\theta)$$



✓ The posterior distribution is

$$\pi(w, \beta, \tau^2, \theta \mid y) \propto p(y \mid w, \beta, \tau^2)\pi(w \mid \theta)\pi(\theta)\pi(\beta, \tau^2)$$

Our goal is to recover w *a posteriori* along with all other unknowns

- Alternatively:

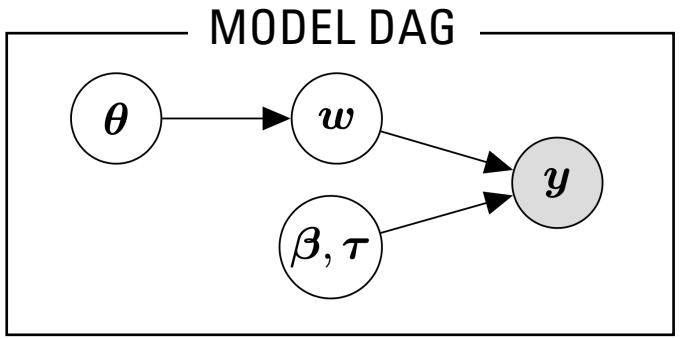
$$\pi(\beta, \tau^2, \theta \mid y) \propto \int p(y \mid w, \beta, \tau^2)\pi(w \mid \theta)\pi(\theta)\pi(\beta, \tau^2)dw$$

COMPUTING THE POSTERIOR

Basic univariate linear model with spatial random effects.

$$y = X\beta + w + \varepsilon, \qquad \varepsilon \sim N(0, \tau^2 I_n)$$

$$w \sim GP(0, C_\theta)$$



✓ The posterior distribution is

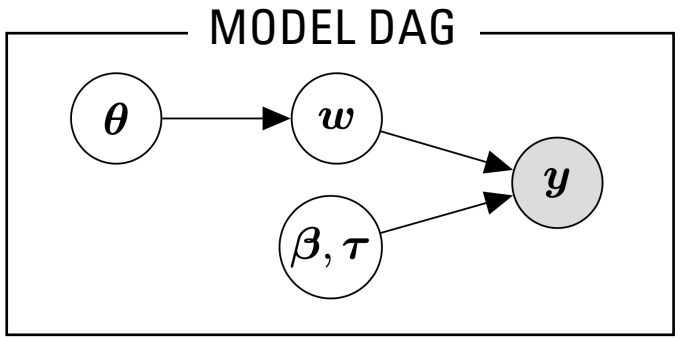
$$\pi(w, \beta, \tau^2, \theta \mid y) \propto p(y \mid w, \beta, \tau^2)\pi(w \mid \theta)\pi(\theta)\pi(\beta, \tau^2)$$

Skeleton of a Gibbs sampler – repeat these steps:

1. Sample new β *given* y and current τ^2, w, ~~θ~~ — easy
2. Sample new τ^2 *given* y and current β, w, ~~θ~~ — easy
3. Sample new w *given* y and current β, τ^2, θ — eh
4. Sample new θ *given* ~~y~~ and current ~~β~~, ~~τ^2~~, w — hard

COMPUTING THE POSTERIOR: BOTTLENECKS

4. Sample new θ *given* w **hard**



Why is this challenging? The posterior distribution is

$$\pi(w, \beta, \tau^2, \theta \mid y) \propto p(y \mid w, \beta, \tau^2)\boldsymbol{\pi(w \mid \theta)\pi(\theta)}\pi(\beta, \tau^2)$$

Bottleneck: $$\boldsymbol{\pi(w \mid \theta)} = N(0, C_\theta) \propto \boldsymbol{|C_\theta|^{-\frac{1}{2}}} \exp\left\{-\frac{1}{2}\, w^\top \boldsymbol{C_\theta^{-1}} w\right\}$$

which for a sample of size n:

- C_θ is of size $n \times n$ with storage cost $O(n^2)$
- C_θ^{-1} and $|C_\theta|$ computed with cost $O(n^3)$
- evaluated at each Gibbs iteration to sample from posterior of θ

SCALING COMPUTATIONS – STEP 1: "PREDICTIVE" GP

First step à la *Gaussian Predictive Process* (Banerjee et al 2008 JRSSB):

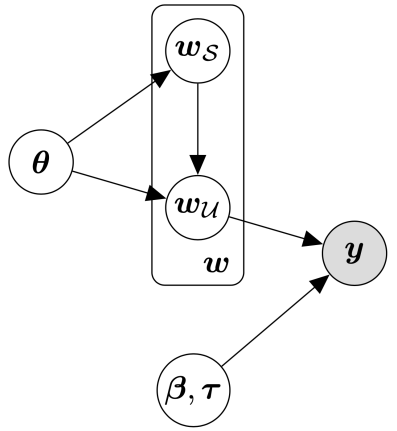
- restrict spatial dependence to a small set S of $k \ll n$ knots
- assume conditional independence in w itself at other locations $u \in U$
- at generic location u the regression model becomes

$$y(u) = x(u)^\top \beta + w(u) + \varepsilon(u), \qquad \varepsilon(u) \sim N(0, \tau^2)$$

$$w(u) = H_u w_S + v, \qquad v \sim N(0, R_u)$$

$$H_u = C_{u,S} C_S^{-1} \text{ and } R_u = C_u - C_{u,S} C_S^{-1} C_{S,u}.$$

PREDICTIVE PROCESS DAG



Skeleton of a Gibbs sampler – repeat these steps:

1. Sample new β *given* y, τ^2, w
2. Sample new τ^2 *given* y, β, w
3. Sample new w_S *given* $w_U, \beta, \tau^2, \theta$ small = easy
4. Sample new w_U *given* $y, w_S, \beta, \tau^2, \theta$ c. indep. = easy
5. Sample new θ *given* w_U *and* w_S not so hard

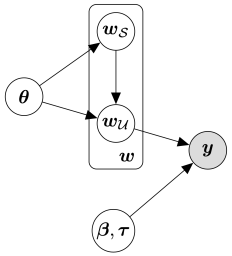
$$\pi(w_U, w_S) = \pi(w_S)\pi(w_U \mid w_S)$$
$$= N(w_S; 0, C_S) \prod_{u \in U} N(w(u); H_u w_S, R_u)$$

↓

👍 now a $k \times k$ matrix
- storage $O(k^2)$
- compute $O(k^3)$

SCALING COMPUTATIONS – STEP 1: "PREDICTIVE" GP

PRED. PROC. DAG

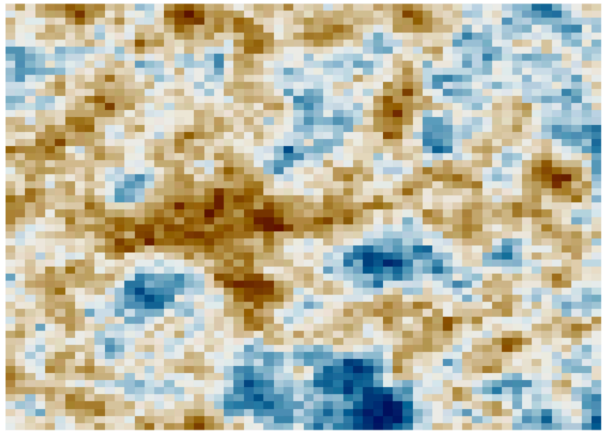


First step à la *Gaussian Predictive Process* (Banerjee et al 2008 JRSSB):

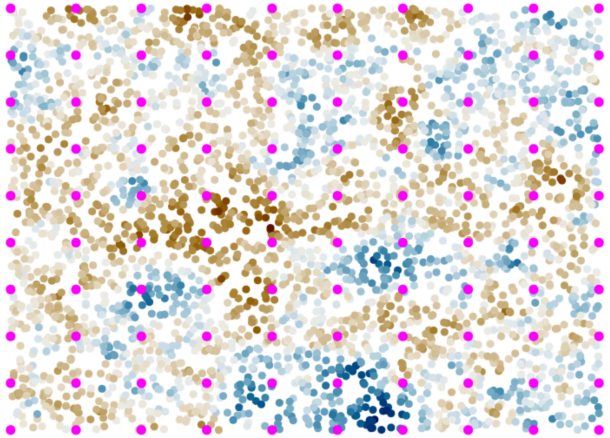
- $k \ll n$ knots may oversmooth the spatial surface
- choice of knot location has no effect on scalability
- "low rank", "inducing points", "sensors"

Example: simulated data with $n = 3600, k = 100$, knots on a 10x10 grid

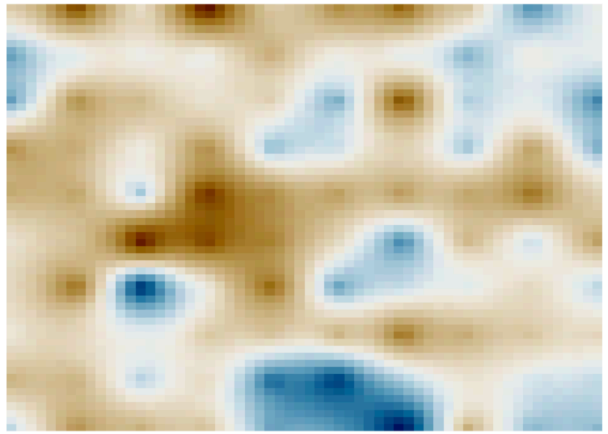
TARGET



DATA & KNOTS



RECOVERED



$C(\ell, \ell') = \sigma^2 \exp(-\varphi|\ \ell - \ \ell'|), \ell \in [0,1]^2$, and setting $\sigma^2 = 1$, $\varphi = 10$, measurement error variance $\tau^2 = 10^{-4}$

SCALING COMPUTATIONS – STEP 1: "PREDICTIVE" GP

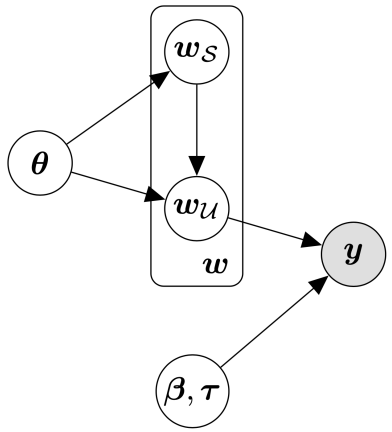
First step à la *Gaussian Predictive Process* (Banerjee et al 2008 JRSSB):

- reduce oversmoothing by increasing k… back to the "same" problem as in the full GP 🤔
- this is because dependence within the set of knot locations S is unrestricted

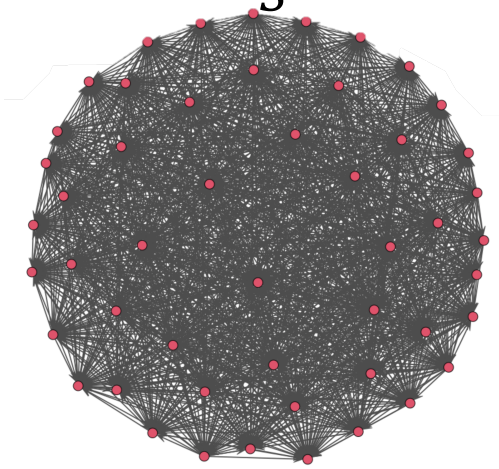
In fact, choose an order within S, so that $w_S = \{w_{s_1}, \ldots, w_{s_k}\}$. Then

$$\pi(w_S) = \pi(w_{s_1})\pi(w_{s_2} \mid w_{s_1})\pi(w_{s_3} \mid w_{s_1}, w_{s_2}) \cdots \pi(w_{s_k} \mid \overbrace{w_{s_1}, \ldots, w_{s_{k-1}}}^{k-1 \text{ parents}})$$

GPP MODEL DAG



GPP w_S DAG



SCALING COMPUTATIONS – STEP 2: NEAREST-NEIGHBOR GP

GPP $$\pi(w_S) = \pi(w_{s_1})\pi(w_{s_2} \mid w_{s_1})\pi(w_{s_3} \mid w_{s_1}, w_{s_2}) \cdots \pi(w_{s_k} \mid \overbrace{w_{s_1}, \ldots, w_{s_{k-1}}}^{k-1 \text{ parents}})$$

Nearest-neighbor GP (Datta et al 2016 JASA):

- limits dependence within w_S
- parent sets reduced to at most m "neighbors"
- DAG determined by knot ordering and m
- DAG is sparsely connected

NNGP $$\pi(w_S) = \pi(w_{s_1})\pi(w_{s_2} \mid w_{s_1})\, \pi(w_{s_3} \mid w_{s_1}, w_{s_2}) \cdots \pi(w_{s_k} \mid \cancel{w_{s_1}, \ldots, w_{s_{k-m-1}}}, \overbrace{w_{s_{k-m}}, \ldots, w_{s_{k-1}}}^{m \text{ parents}})$$

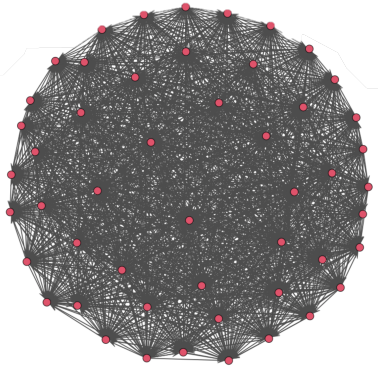
$$= \pi(w_{s_1})\pi(w_{s_2} \mid w_{[s_2]})\pi(w_{s_3} \mid w_{[s_3]}) \cdots \pi(w_{s_k} \mid w_{[s_k]})$$

$[s_i]$ set of parent locations for s_i in the DAG

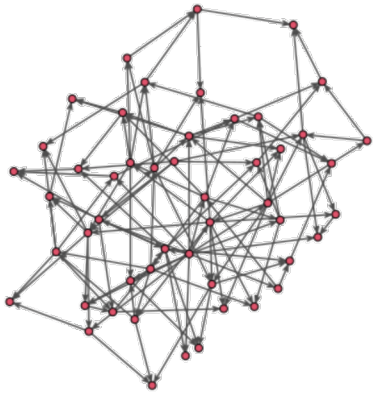
SCALING COMPUTATIONS – STEP 2: NEAREST-NEIGHBOR GP

✓ *sparse* DAG implies we can increase size of S. Set $S = T$ i.e. knots at observed locations so $k = n$

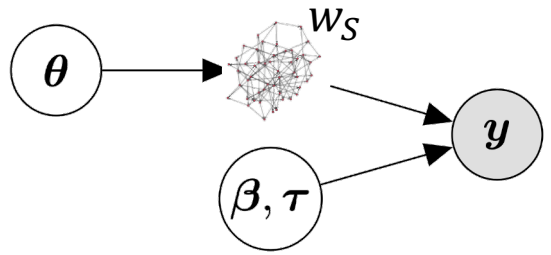
GPP w_S DAG



NNGP w_S DAG



NNGP MODEL DAG



NNGP

$$\pi(w_S) = \pi(w_{s_1})\pi(w_{s_2} \mid w_{s_1})\,\pi(w_{s_3} \mid w_{s_1}, w_{s_2}) \cdots \pi(w_{s_k} \mid w_{s_1}, \ldots, w_{s_{k-m-1}}, \overbrace{w_{s_{k-m}}, \ldots, w_{s_{k-1}}}^{m \text{ parents}})$$

$$= \pi(w_{s_1})\pi(w_{s_2} \mid w_{[s_2]})\pi(w_{s_3} \mid w_{[s_3]}) \cdots \pi(w_{s_k} \mid w_{[s_k]})$$

$[s_i]$ set of parent locations for s_i in the DAG

SCALING COMPUTATIONS – STEP 2: NEAREST-NEIGHBOR GP

If knots aka *reference* locations overlap with observed locations: $y(s) = x(s)^\top \beta + w(s) + \varepsilon(s),\ \varepsilon(s) \sim N(0, \tau^2)$,

$$w(s) = H_s w_{[s]} + v, \qquad v \sim N(0, R_s)$$

$$\pi(w_S) = \pi(w_{s_1})\pi(w_{s_2} \mid \overbrace{w_{[s_2]}}^{\text{size } m \text{ or less}})\pi(w_{s_3} \mid \overbrace{w_{[s_3]}}^{\text{size } m \text{ or less}}) \cdots \pi(w_{s_n} \mid \overbrace{w_{[s_n]}}^{\text{size } m \text{ or less}})$$

$$= \Pi_{s \in S} N(w(s); H_s w_{[s]}, R_s)$$

$$H_s = C_{s,[s]} \underbrace{C_{[s]}^{-1}}_{\text{size } m \times m \text{ or less}} \qquad R_s = C_s - C_{s,[s]} \underbrace{C_{[s]}^{-1}}_{\text{size } m \times m \text{ or less}} C_{[s],s}$$

Skeleton of a Gibbs sampler – repeat these steps:

1. Sample new β *given* y, τ^2, w
2. Sample new τ^2 *given* y, β, w
3. Sample new $w(s)$ *given* $y, w_{[s]}, \beta, \tau^2, \theta$
4. Sample new θ *given* current w_S

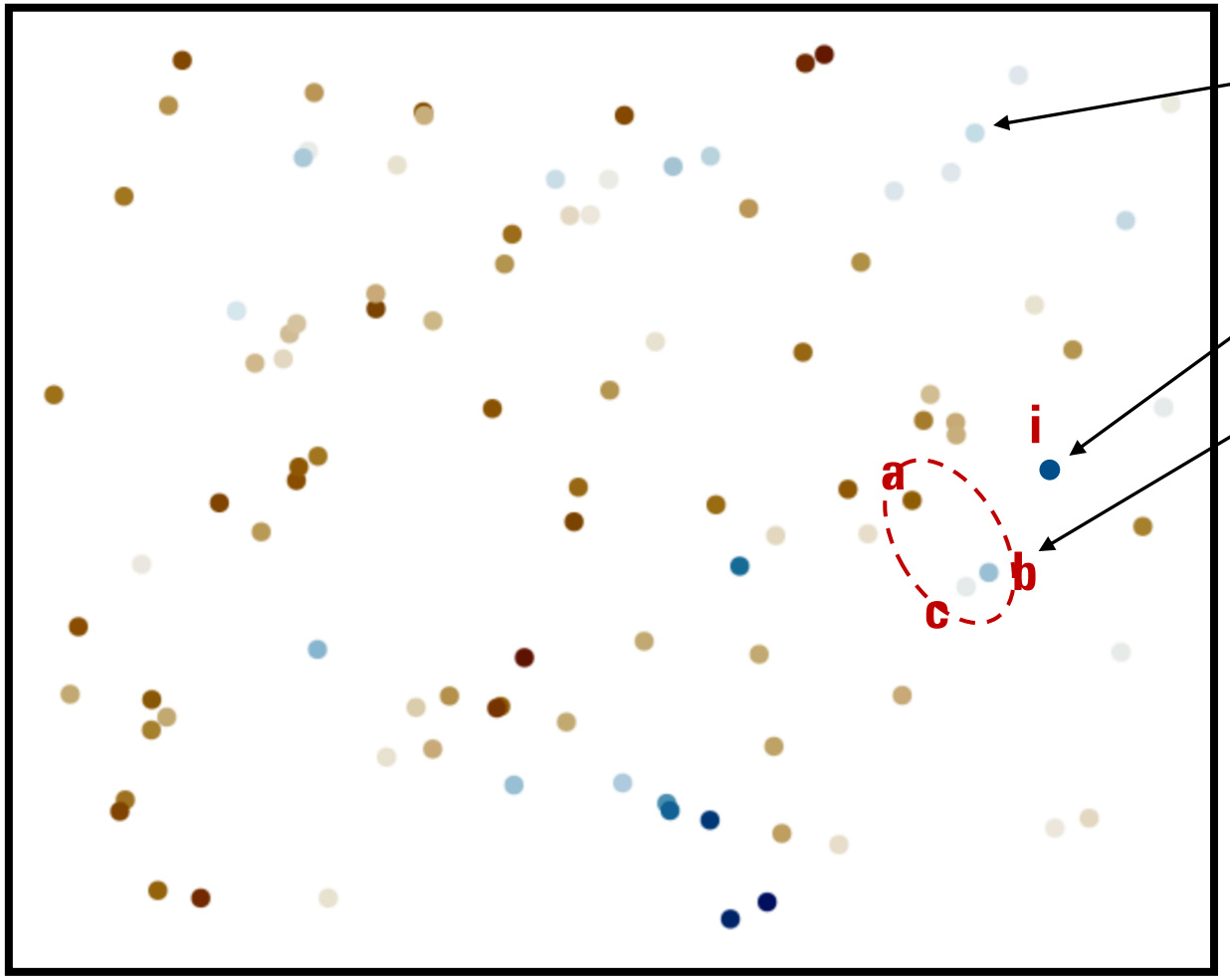
For each $s \in S$, we need $C_{[s]}^{-1}$. In total:

- 👍 n matrices of size $m \times m$
- storage $O(nm^2)$
- compute $O(nm^3)$

$m < 20$? Scales to big n

SCALING COMPUTATIONS — STEP 2: NEAREST-NEIGHBOR GP

Spatial domain D



Observed locations

Example location i

Parent set for location i is the set of locations {a, b, c}

We need

$$H_i = C_{i,[i]} C_{[i]}^{-1}$$

$$R_i = C_i - C_{i,[i]} C_{[i]}^{-1} C_{[i],s}$$

$$C_{[i]} = \begin{bmatrix} C(a,a) & C(a,b) & C(a,c) \\ C(b,a) & C(b,b) & C(b,c) \\ C(c,a) & C(c,b) & C(c,c) \end{bmatrix}$$

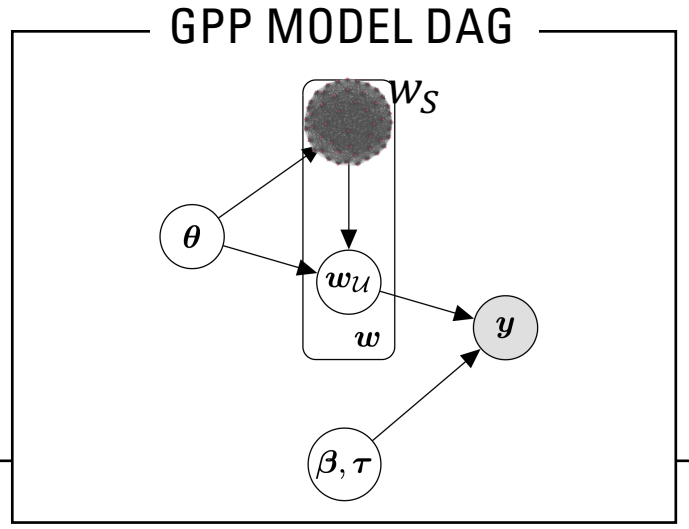
SCALING COMPUTATIONS — STEP 2: NEAREST-NEIGHBOR GP

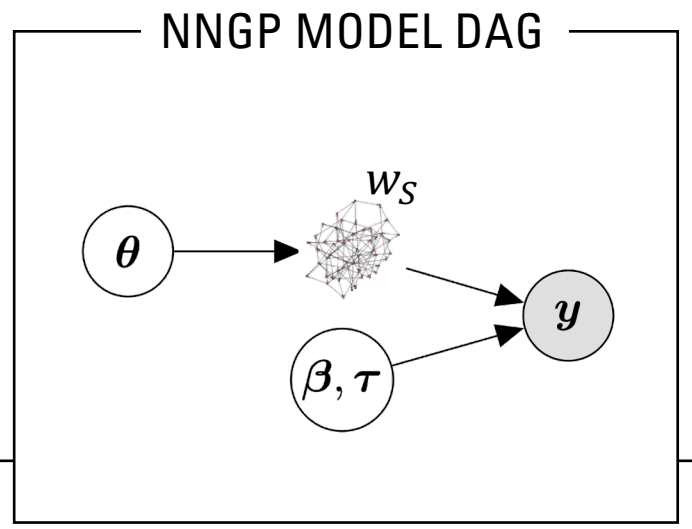
Nearest-neighbor GP (Datta et al 2016 JASA):

- defines valid process based on DAG for w
- DAG for w embedded in Bayesian model DAG
- sparse DAG built "automatically" from neighbors
- DAG knots 1:1 to elements of S
- Gibbs sampling for $w(s)$ using Gaussian full conditionals
- no need for reference set (the knots) being separate from the set of observed locations, i.e. $U = \emptyset$

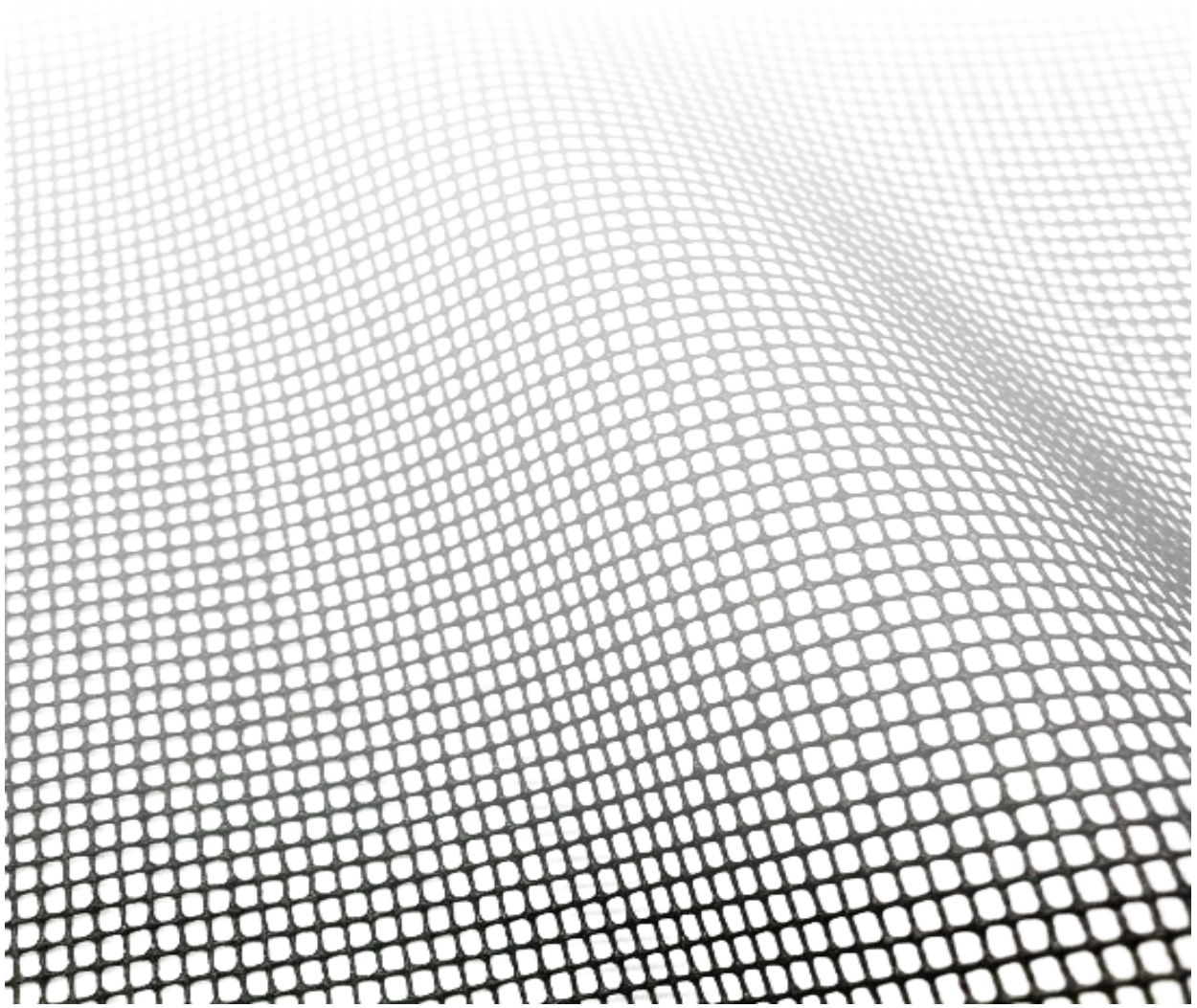
- scalability limited to $m \lesssim 20$
- not immediate extension to space-time or $d > 3$
- must reduce m in multivariate regression

  (if q outcomes, size of parent sets is mq)
- Gibbs sampler proceeds sequentially for w
- MCMC efficiency/convergence concerns





MESHED GAUSSIAN PROCESSES

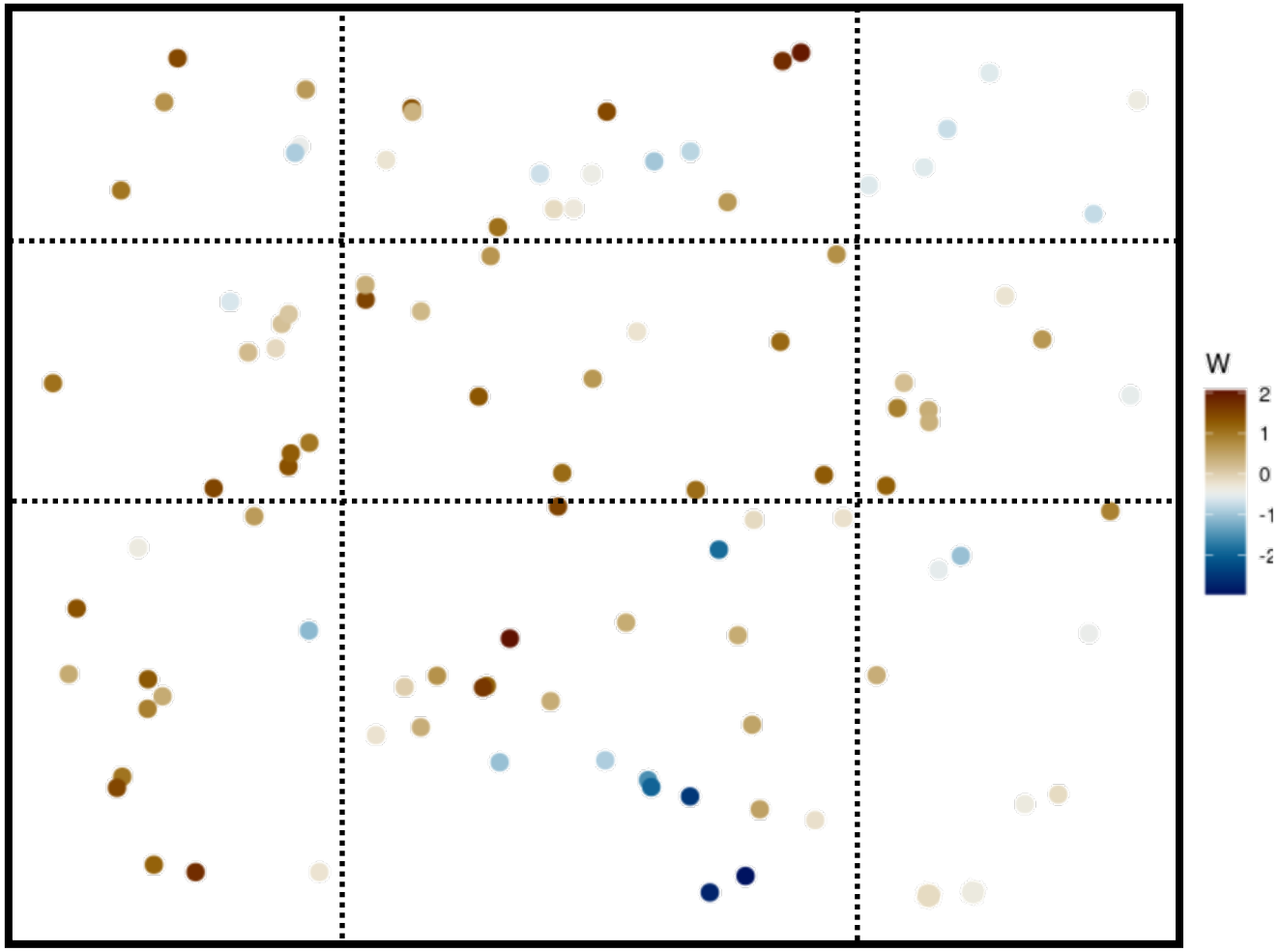


Key ingredients:

1. Domain partitioning
2. Fixed, “nice” DAGs
3. Nodes 1:1 knot partitions
4. Gridded/patterned knots

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP [1/4]

Spatial domain D

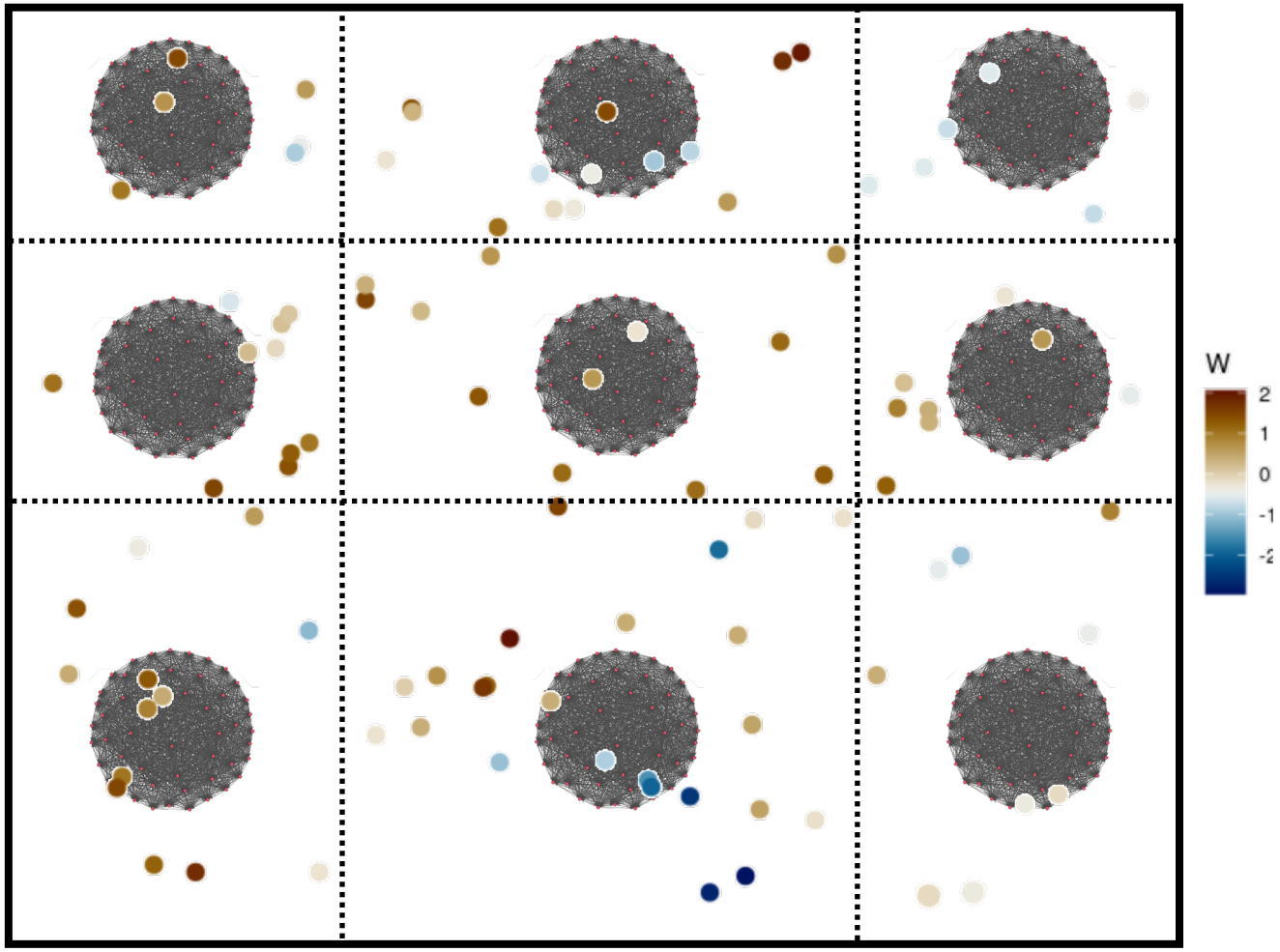


1. Domain partitioning

- Induces partition on observation/knots
- Axis-parallel is convenient

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP [1/4]

Spatial domain D

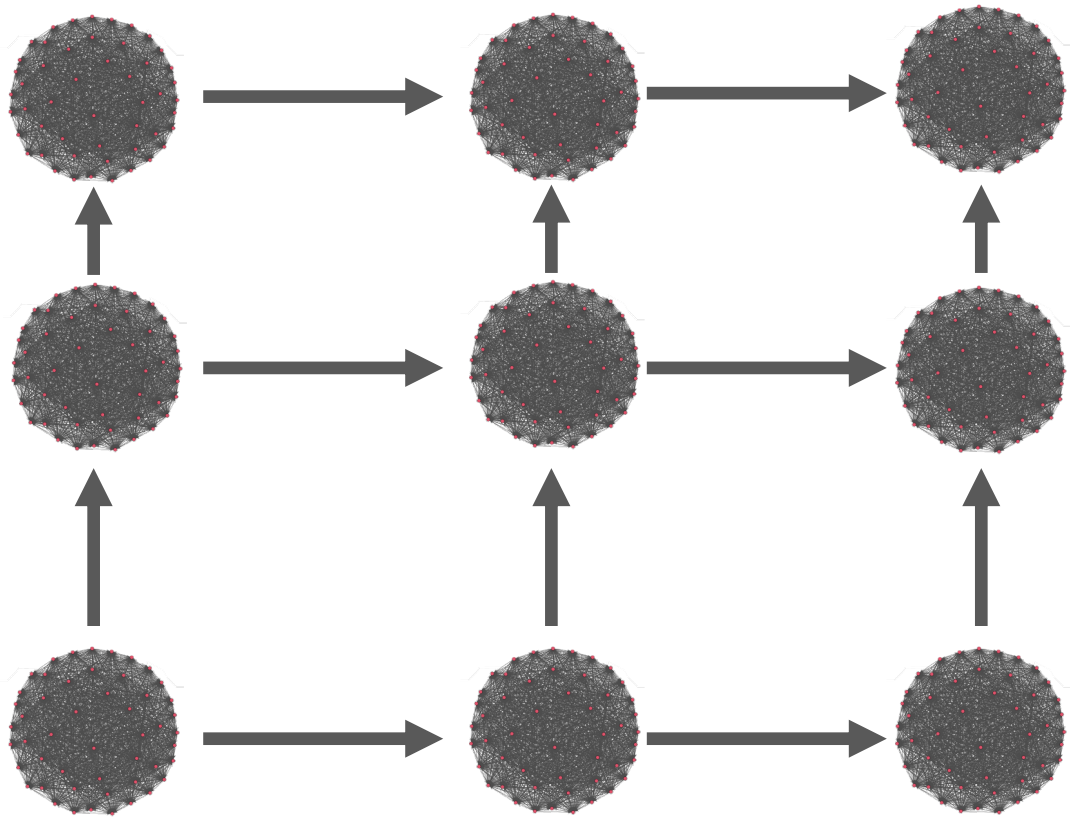


1. Domain partitioning

- Induces partition on observation/knots
- Axis-parallel is convenient
- Full dependence allowed in each partition

(Independent partition models would stop here.)

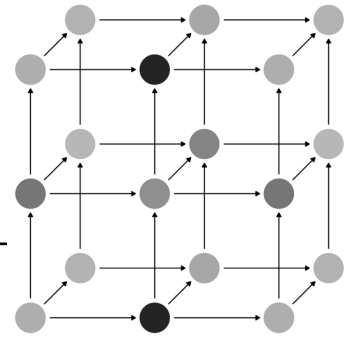
MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP [2/4]



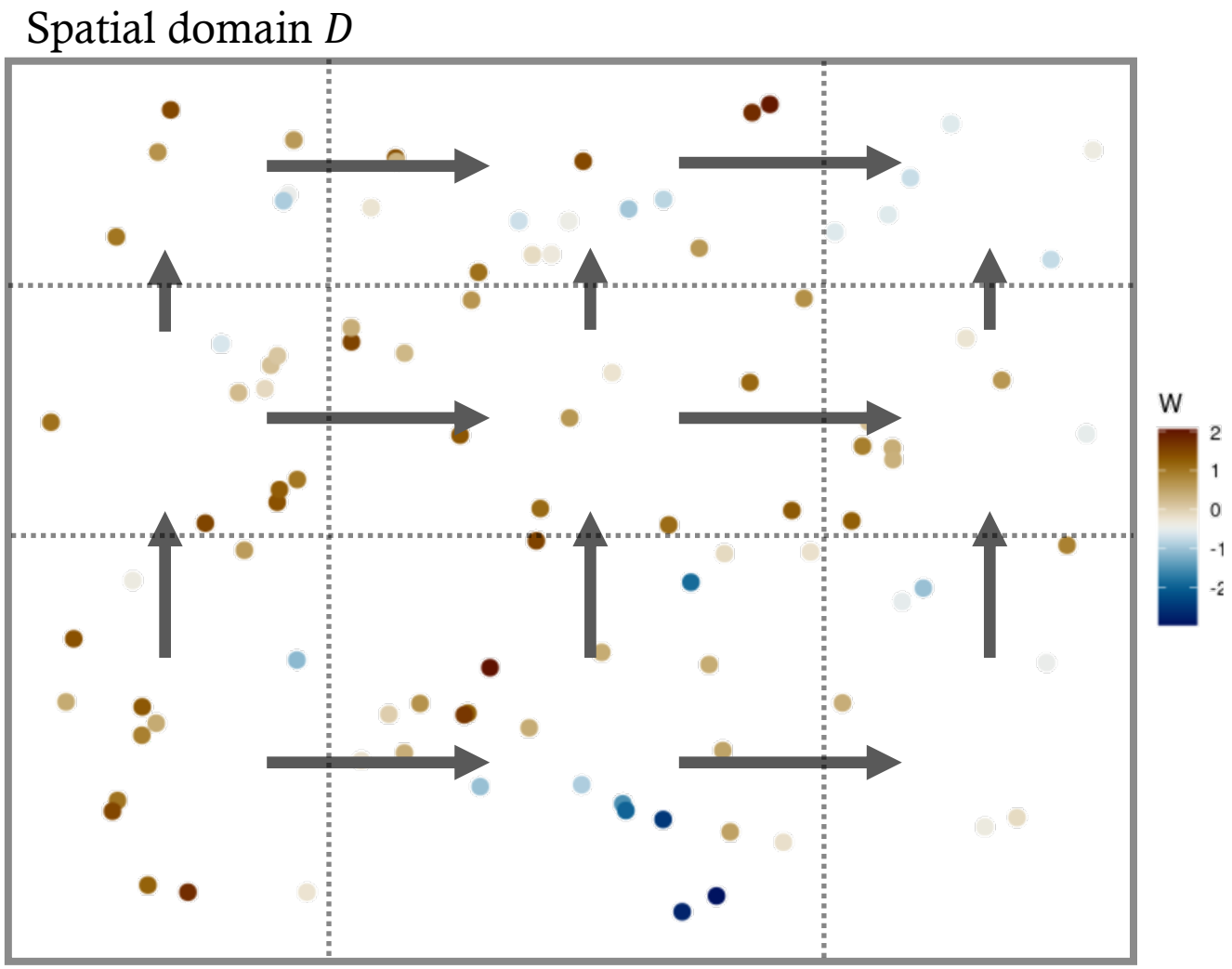
2. "Nice" DAG

- Abstracts from observations
- Agnostic of actual locations
- *Chosen* because it induces good properties

A "cubic" DAG extends to space-time domains:



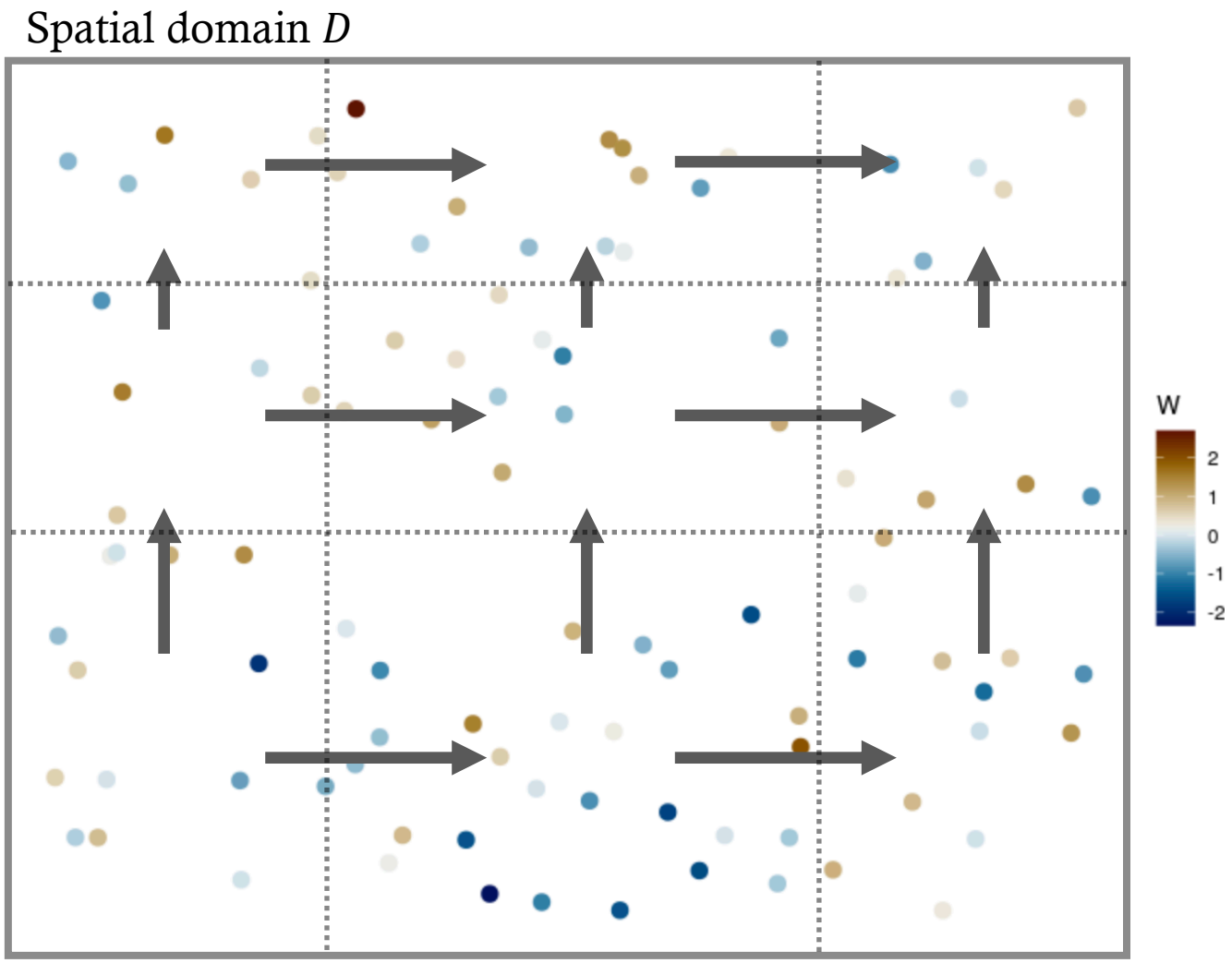
MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP [3/4]



3. Nodes 1:1 partitions

- Groups of knots collectively act as a node
- Partition membership matters

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP [3/4]

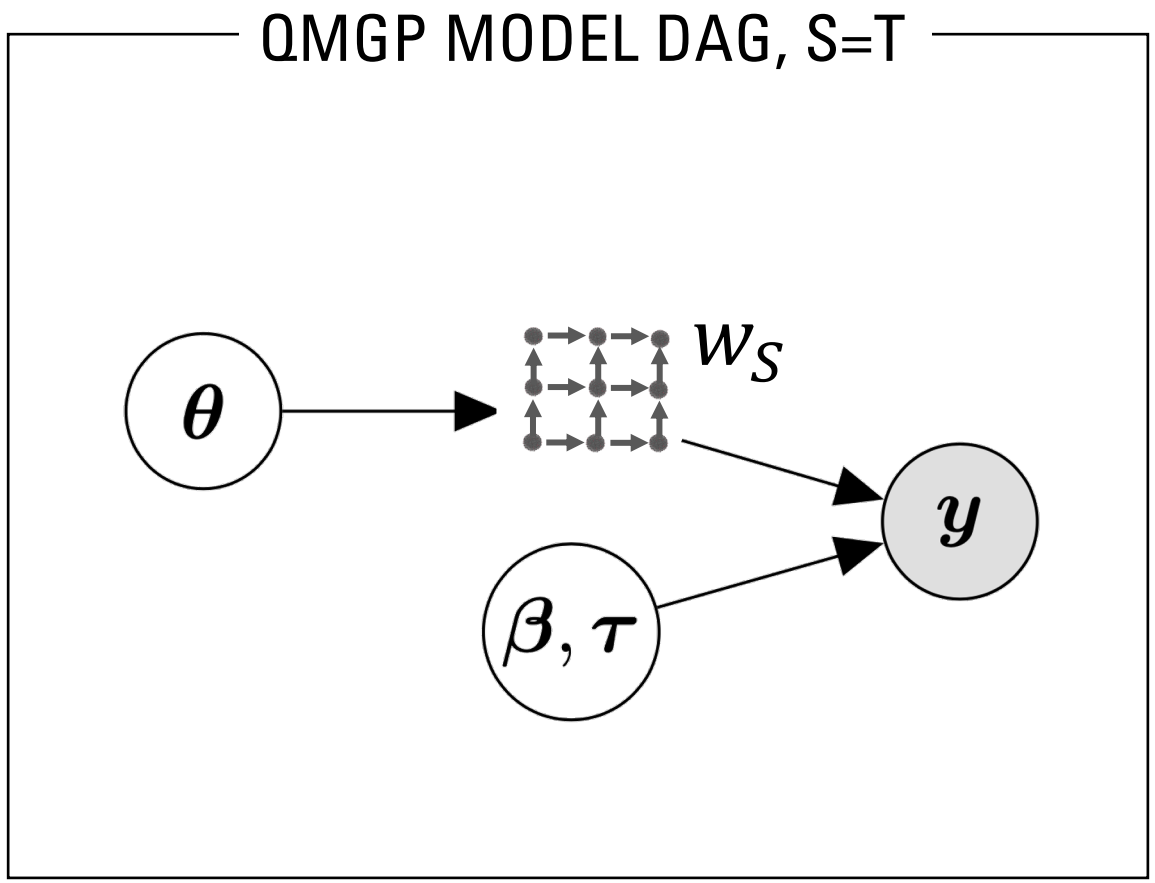


3. Nodes 1:1 partitions

- Groups of knots collectively act as a node
- Partition membership matters
- Same abstract DAG for any sample

  = same properties always

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP



Choice of reference/knot set S

- We can overlap knots S with observations T
- Not necessarily the smartest thing

MESHED GAUSSIAN PROCESSES – EXAMPLE: CUBIC MGP

Back to computations…

- We now have M partitions indexed as $j_1, \ldots, j_M$
- Number of parents is 2 or less
- Size of parent sets depends on number of locations in each partition

$$\pi(w_S) = \pi(w_{j_1})\pi(w_{j_2} \mid w_{[j_2]})\pi(w_{j_3} \mid w_{[j_3]}) \cdots \pi(w_{j_M} \mid w_{[j_M]})$$

$$= \Pi_{i=1}^{M} N(w_{j_i}; H_{j_i} w_{[j_i]}, R_{j_i}) \qquad H_{j_i} = C_{j_i,[j_i]} C_{[j_i]}^{-1} \qquad R_{j_i} = C_{j_i} - C_{j_i,[j_i]} C_{[j_i]}^{-1} C_{[j_i],j_i}$$

Skeleton of a Gibbs sampler – repeat these steps:

1. Sample β *given* y, τ^2, w
2. Sample τ^2 *given* y, β, w
3. Sample $w(s)$ *given* $y, w_{[s]}, \beta, \tau^2, \theta$
4. Sample θ *given* w_S

For each partition, we need $C_{[j_i]}^{-1}$. In total:

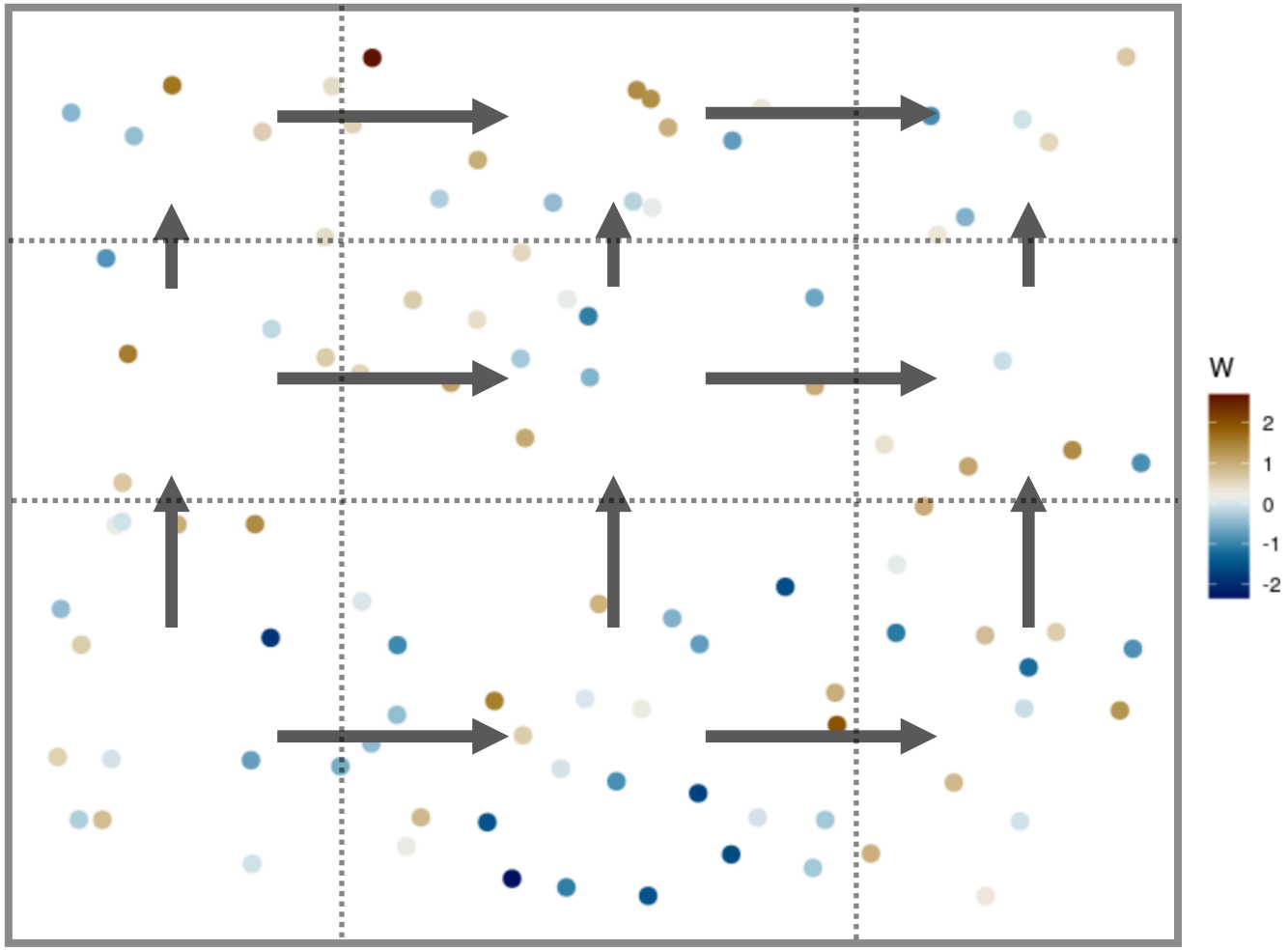
👍 M matrices of size J or less

- storage $O(MJ^2)$
- compute $O(MJ^3)$

J small? Scales to big n

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP

Spatial domain D

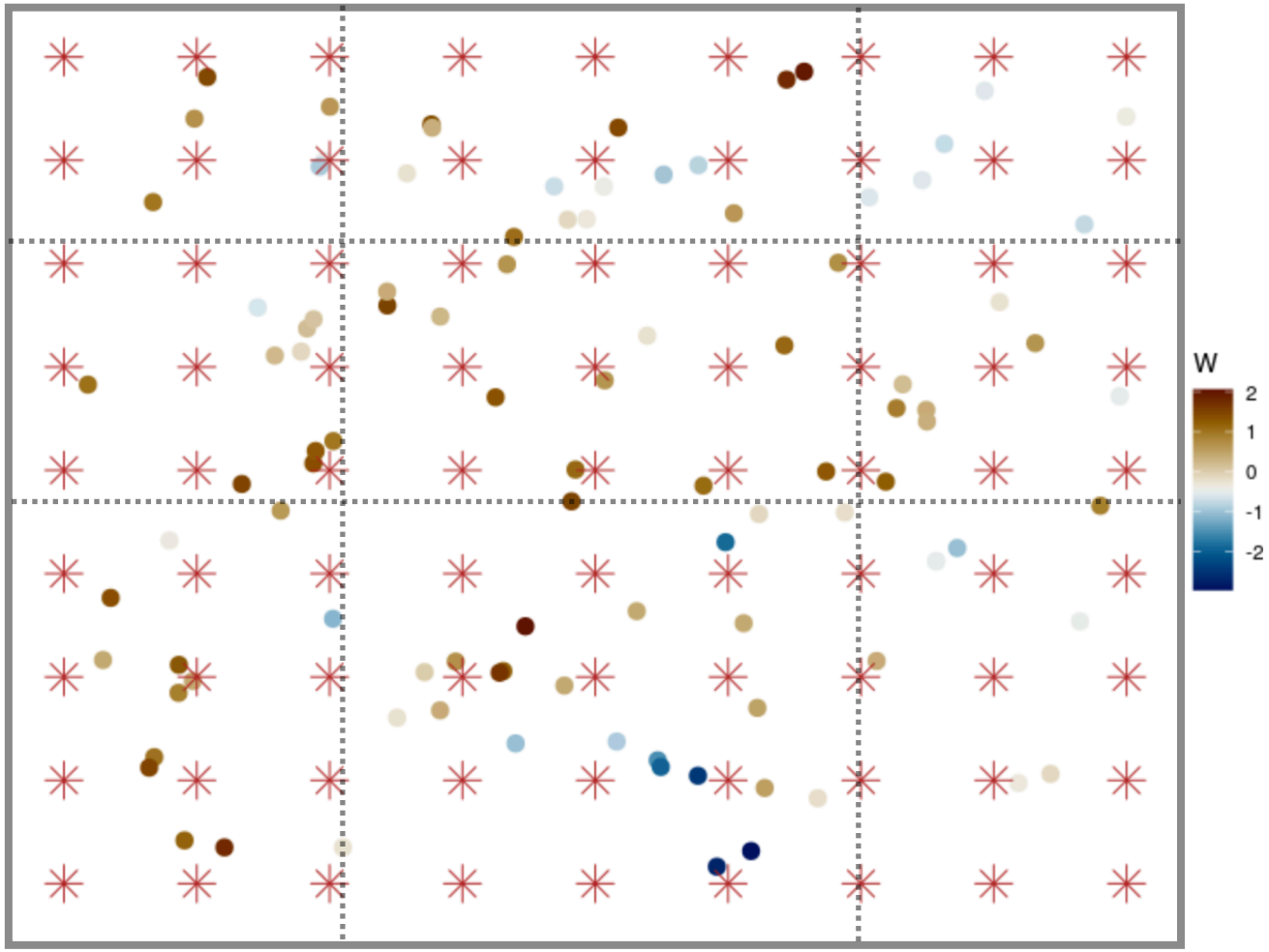


Can we do better?

- $M \ll n$ means we can choose relatively large partitions, but
- We are limited by the size of the largest partitions
- Which depends on observed locations…

MESHED GAUSSIAN PROCESSES – EXAMPLE: CUBIC MGP [4/4]

Spatial domain D



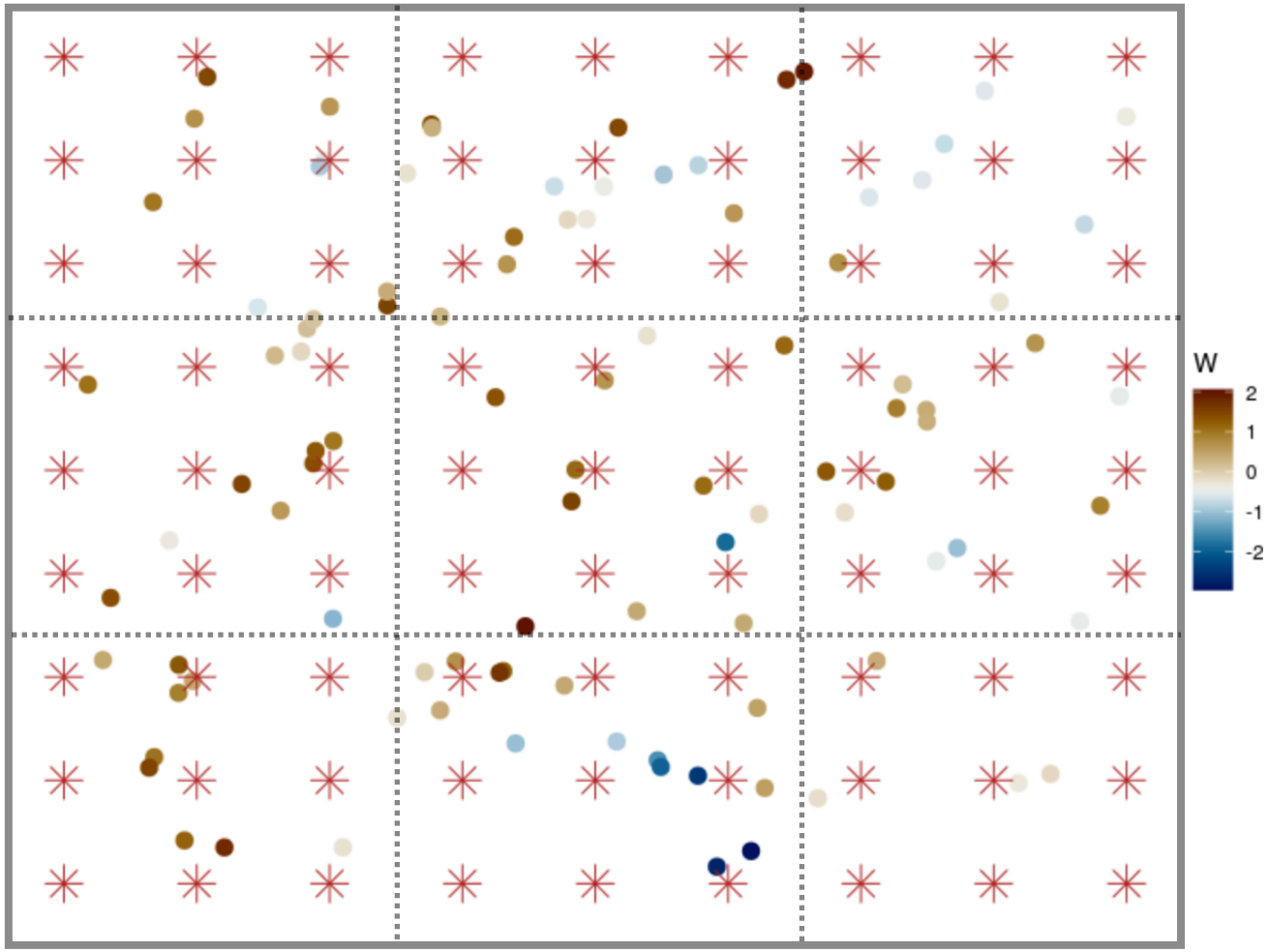
Yes, we can!

4. Gridded/patterned knots

- Choose the reference set on a pattern
- Simplest: regular grid
- DAG remains the same!
- Place all observed locations outside the reference set S

MESHED GAUSSIAN PROCESSES – EXAMPLE: CUBIC MGP [4/4]

Spatial domain D



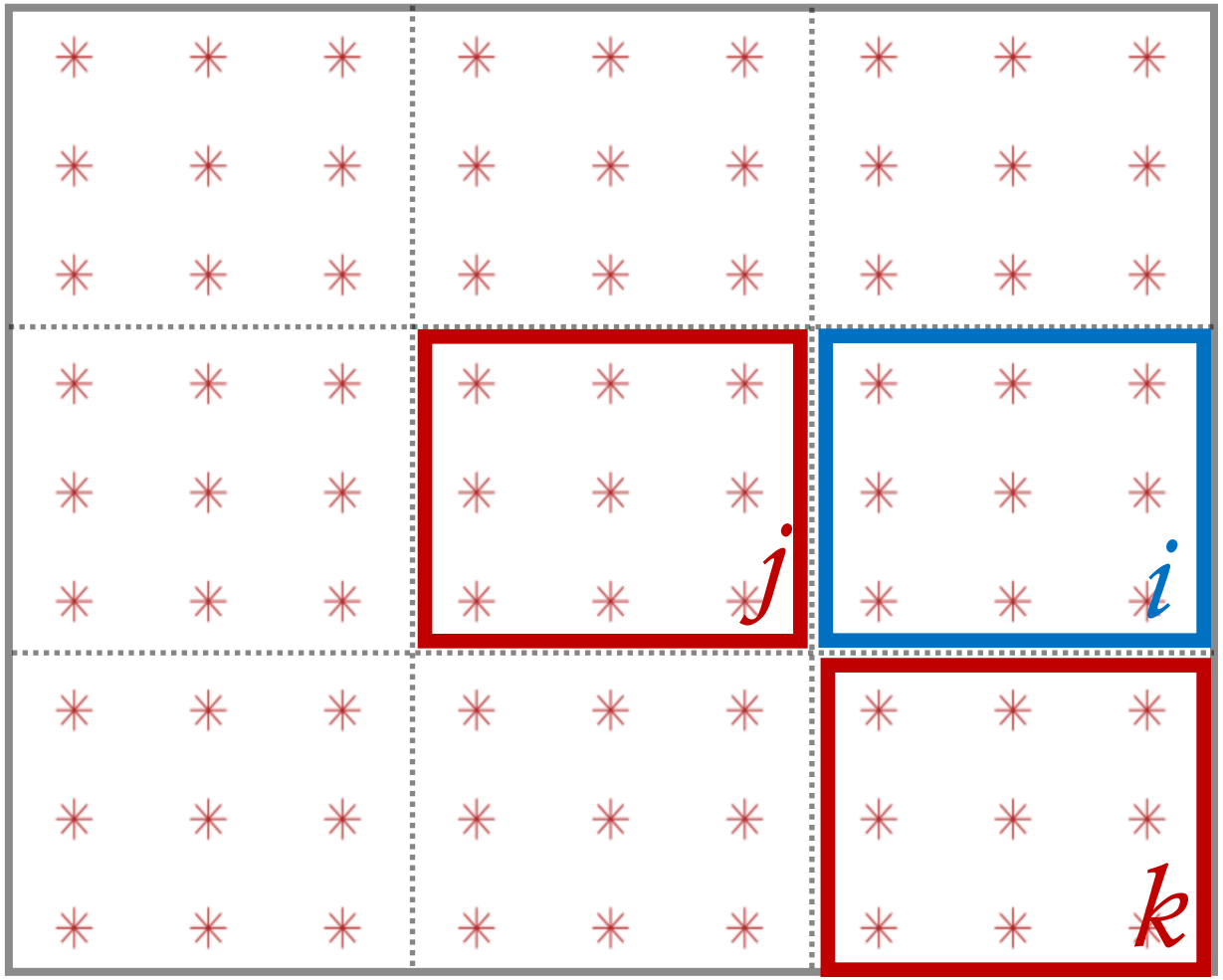
Yes, we can!

4. Gridded/patterned knots

- Choose the reference set on a pattern
- Simplest: regular grid
- DAG remains the same!
- Place all observed locations outside the reference set S
- Partition the gridded set

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP [4/4]

Spatial domain D



DAG + Partition + Grid = Magic!

Suppose our covariance function is **stationary**

Partition i with parents $[i] = \{j, k\}$

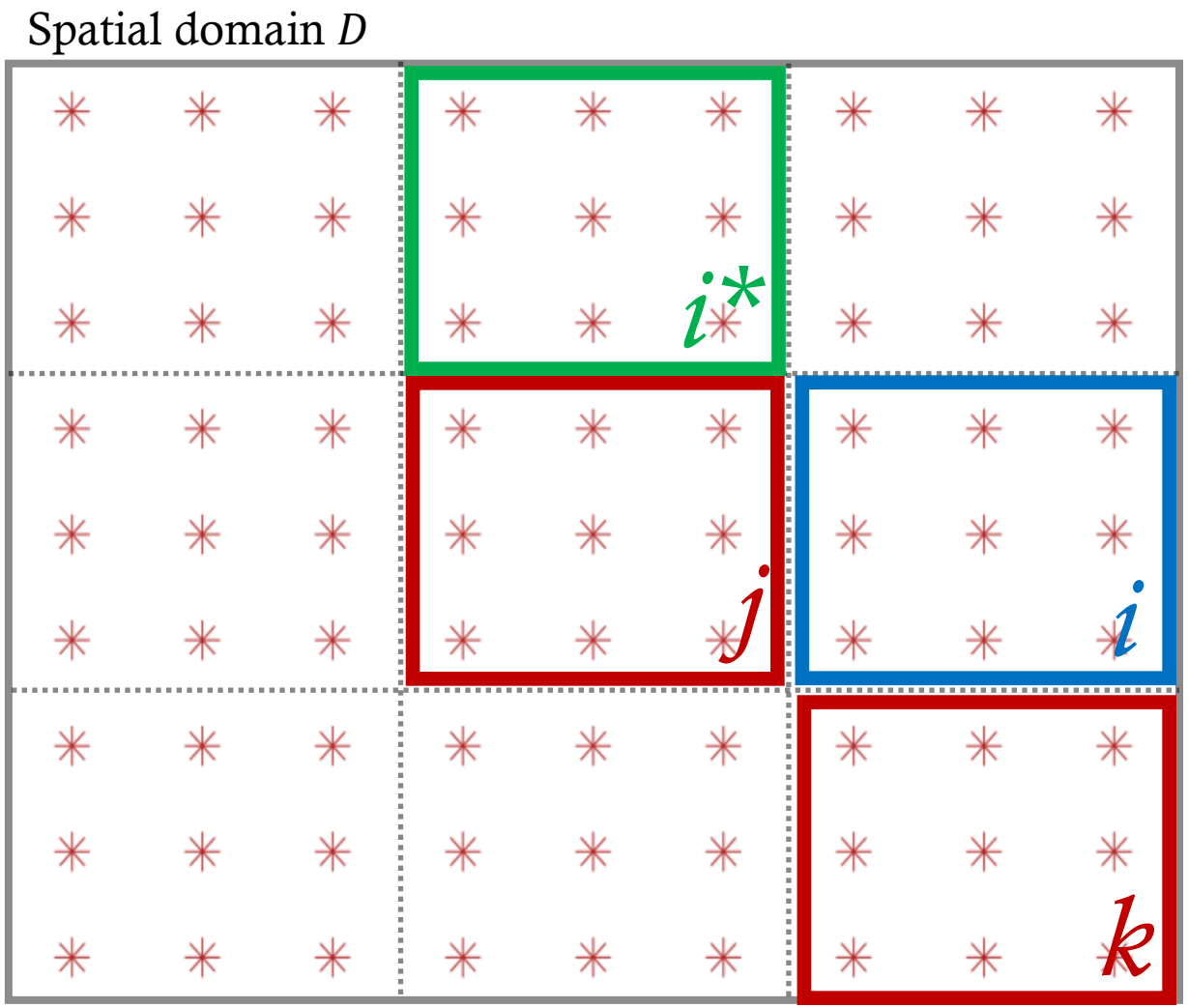
We need

$$H_i = C_{i,[i]} C_{[i]}^{-1}$$
$$R_i = C_i - C_{i,[i]} C_{[i]}^{-1} C_{[i],i}$$

H_i and R_i only depend on locations in i

relative to $\{j,k\}$

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP [4/4]



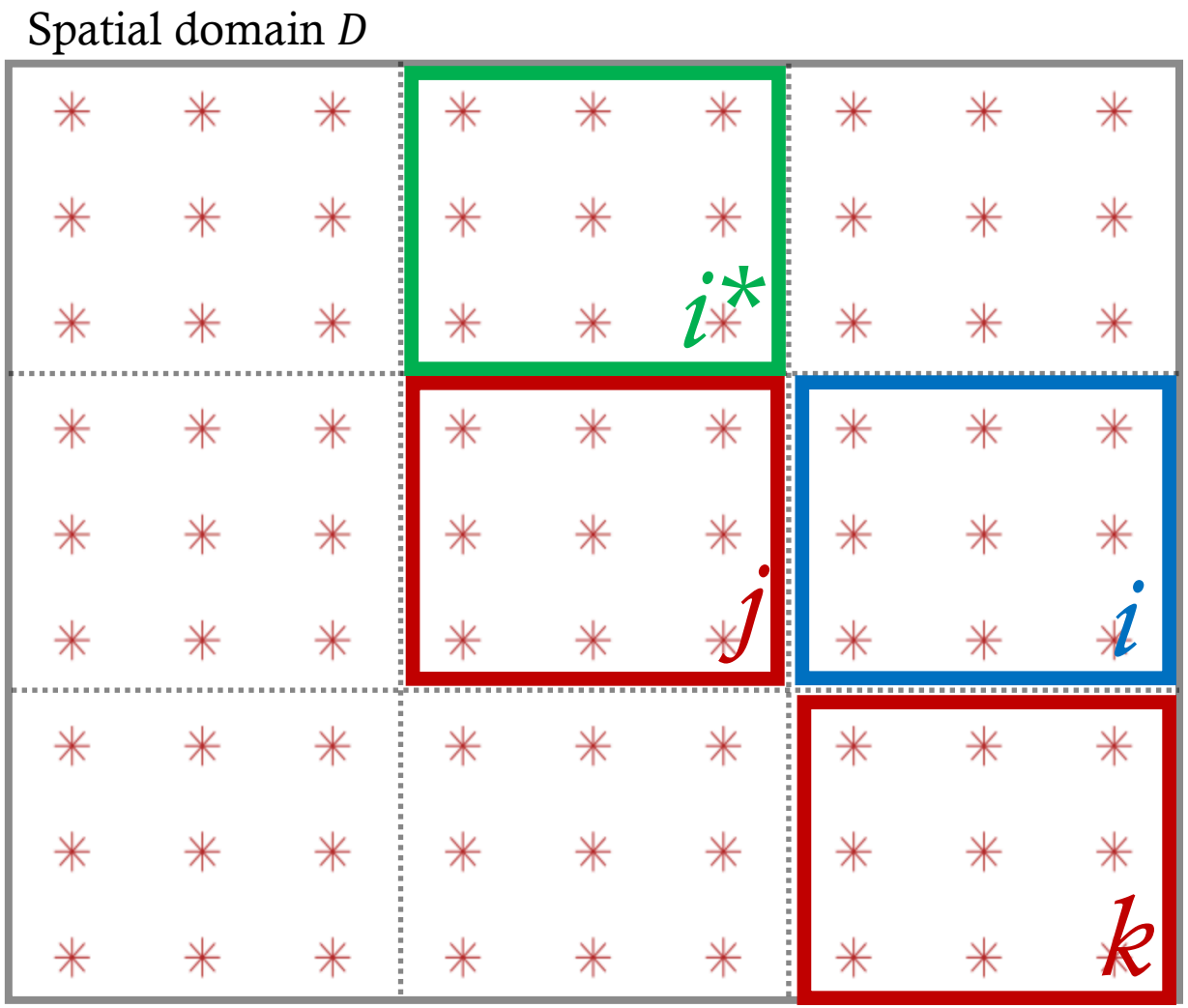
DAG + Partition + Grid = Magic!

Suppose our covariance function is **stationary**

$$H_i = H_{i*}$$

$$R_i = R_{i*}$$

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP [4/4]



DAG + Partition + Grid = Magic!

Suppose our covariance function is **stationary**

9 DAG nodes and 9 partitions

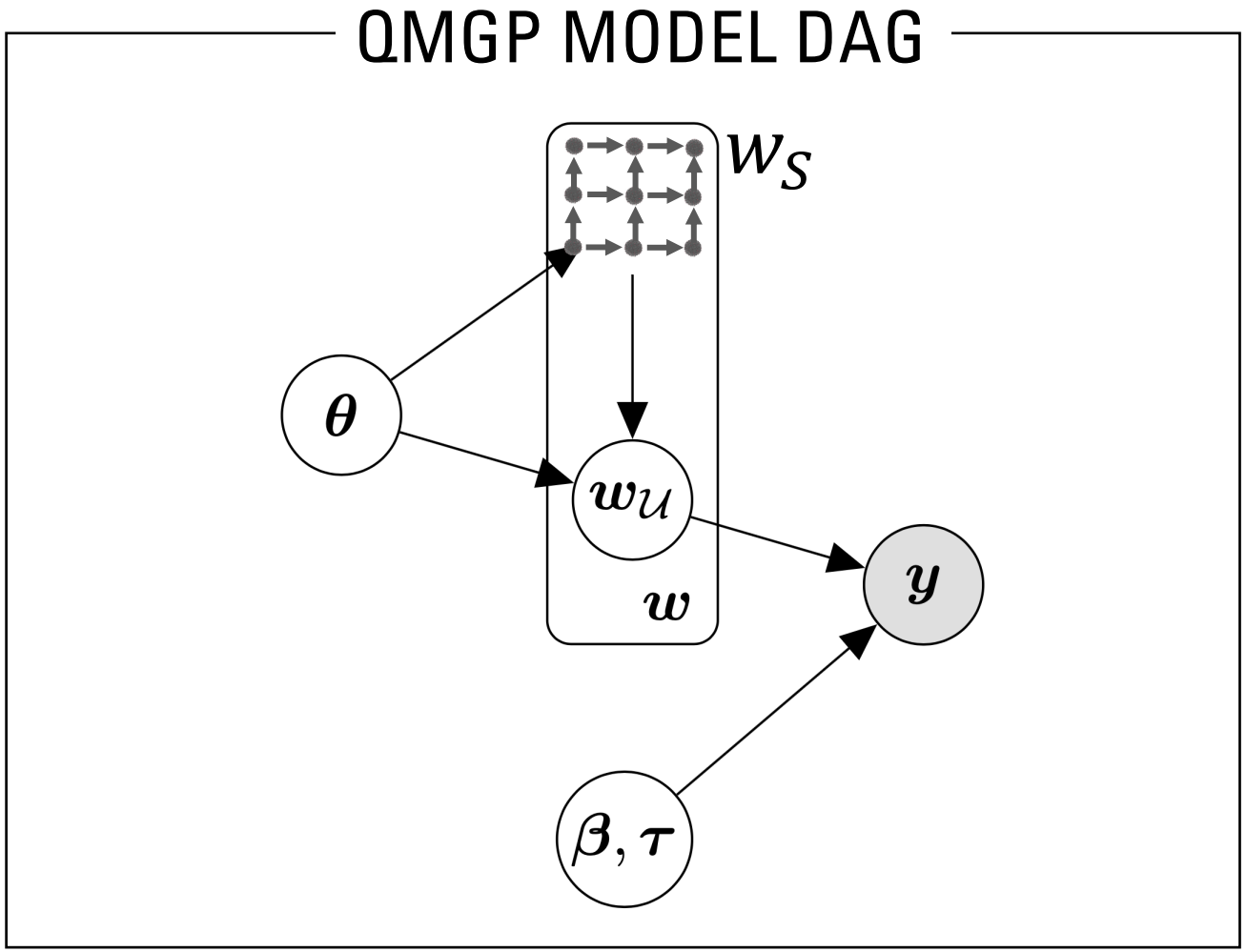
…but only ***3*** R_i*'s and* ***4*** H_i*'s to compute!*

More data, bigger domain, larger grid?

[many] DAG nodes and [many] partitions

…but only ***3*** R_i*'s and* ***4*** H_i*'s to compute!*

MESHED GAUSSIAN PROCESSES — EXAMPLE: CUBIC MGP



MESHED GAUSSIAN PROCESSES – EXAMPLE: CUBIC MGP

Back to computations…

- We now have M partitions indexed as $j_1, \ldots, j_M$
- Number of parents is 2 or less
- Size of parent sets is **fixed** as twice the size of grid partitions

CACHING

1. First, find the 4 "prototypes"
2. Then compute:

$$\boldsymbol{H_x} = C_{x,[x]} C_{[x]}^{-1}$$

$$\boldsymbol{R_x} = C_x - C_{x,[x]} C_{[x]}^{-1} C_{[x],x}$$

$$\pi(w_S) = \pi(w_{j_1})\pi(w_{j_2} \mid w_{[j_2]})\pi(w_{j_3} \mid w_{[j_3]}) \cdots \pi(w_{j_M} \mid w_{[j_M]})$$

$$= \Pi_{i=1}^{M} N(w_{j_i}; H_{j_i} w_{[j_i]}, R_{j_i}) \qquad H_{j_i} = \boldsymbol{H_{x^*}} \qquad R_{j_i} = \boldsymbol{R_{x^*}}$$

with x^* being the prototype for j_i

Skeleton of a Gibbs sampler – repeat these steps:

1. Sample β *given* y, τ^2, w
2. Sample τ^2 *given* y, β, w
3. Sample w_U *given* $y, w_S, \beta, \tau^2, \theta$
4. Sample w_{j_i} *given* $w_U, w_{[j_i]}, \theta$
5. Sample θ *given* w_U, w_S

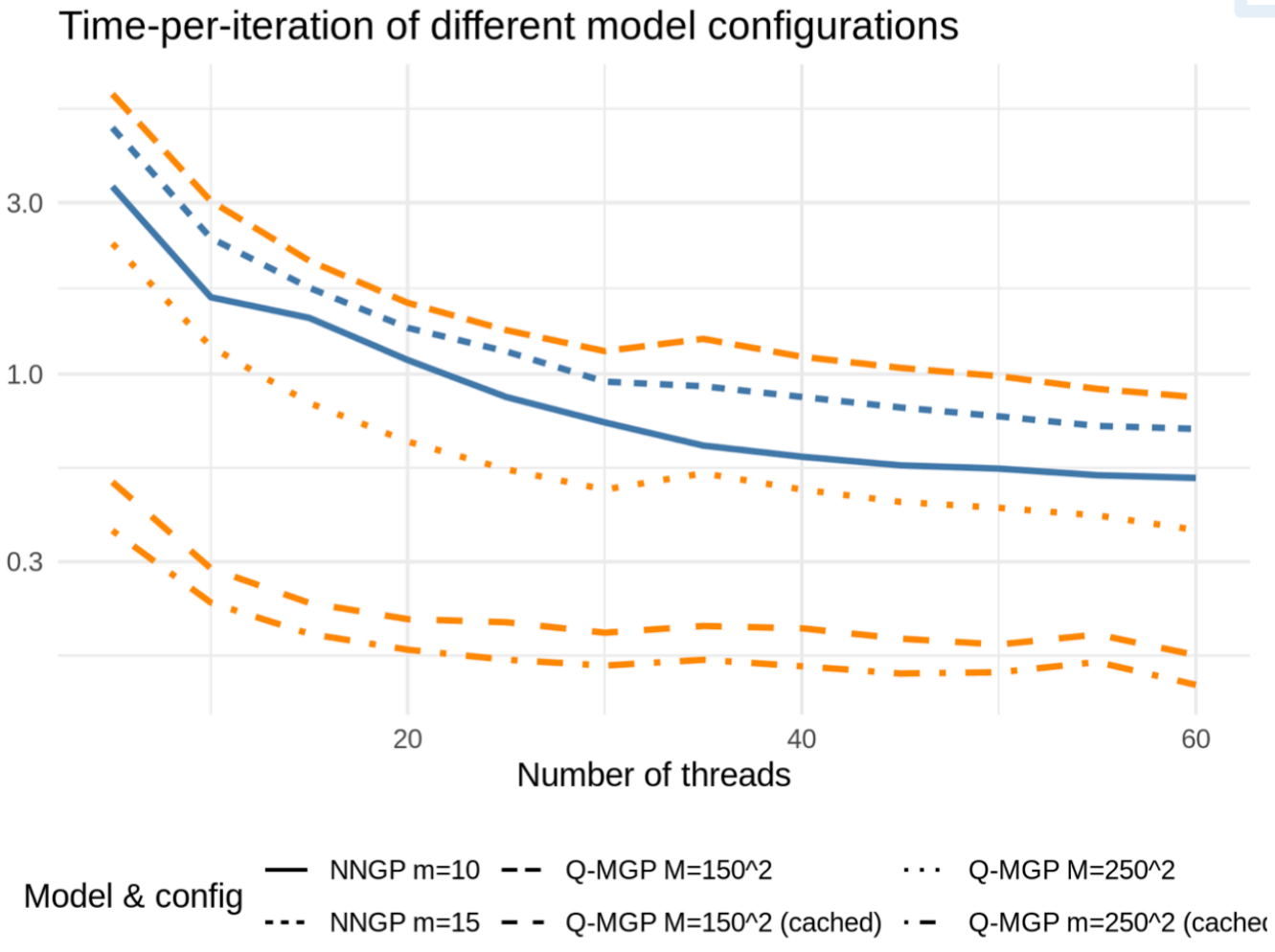
For each partition, we need $C_{[x]}^{-1}$. In total:

👍 **4** matrices of size J or less

- storage $O(J^2)$
- compute $O(J^3)$

Does not depend on data size

MESHED GAUSSIAN PROCESSES – EXAMPLE: CUBIC MGP



CACHING

1. First, find the 4 "prototypes"
2. Then compute:

$$\boldsymbol{H_x} = C_{x,[x]}C_{[x]}^{-1}$$

$$\boldsymbol{R_x} = C_x - C_{x,[x]}C_{[x]}^{-1}C_{[x],x}$$

MESHED GAUSSIAN PROCESSES – EXAMPLE: CUBIC MGP

- With caching, we can massively increase partition size!
- OR, we make density evaluation ultra cheap
- Total cost ≈ independent partitions model… without independence!
- If data are gridded, $S = T$ **with caching** 
- Otherwise, sample at $k + n$ locations, with $k \approx n$. Crazy? (Yes)

CACHING

1. First, find the 4 "prototypes"
2. Then compute:

$$\boldsymbol{H_x} = C_{x,[x]}C_{[x]}^{-1}$$

$$\boldsymbol{R_x} = C_x - C_{x,[x]}C_{[x]}^{-1}C_{[x],x}$$

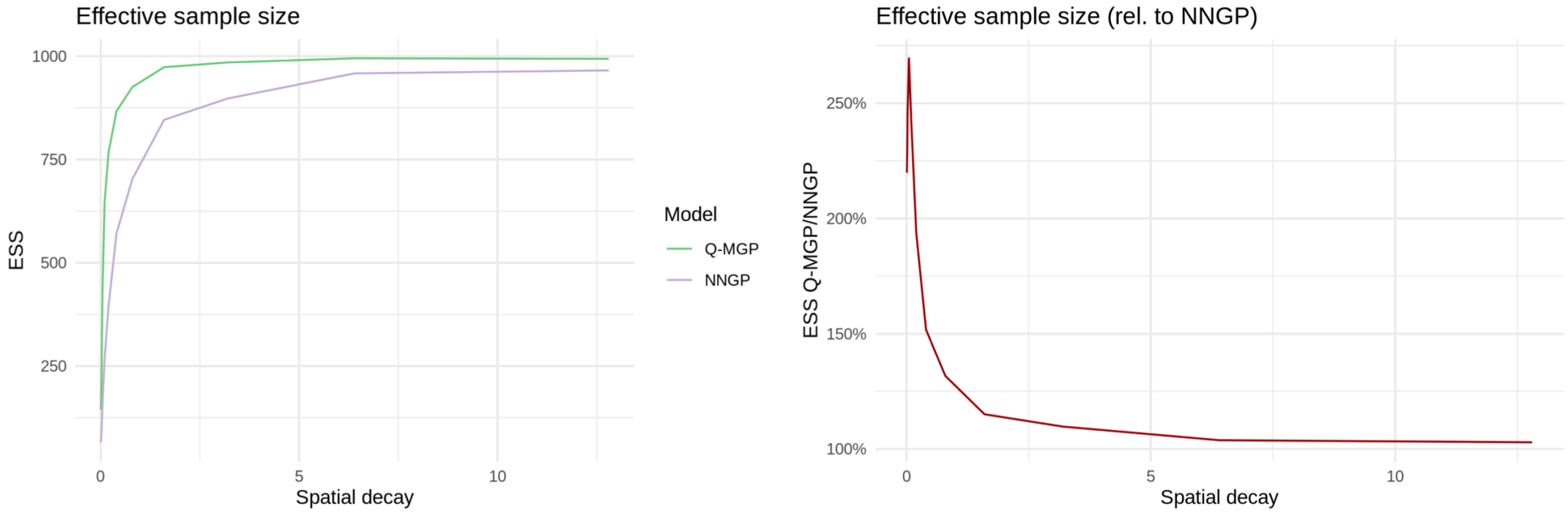
- **Sampling** w (steps 3 and 4) takes larger portion of time

Skeleton of a Gibbs sampler – repeat these steps:

1. Sample β *given* y, τ^2, w
2. Sample τ^2 *given* y, β, w
3. Sample w_U *given* $y, w_S, \beta, \tau^2, \theta$
4. Sample w_{j_i} *given* $w_U, w_{[j_i]}, \theta$
5. Sample θ *given* w_U, w_S

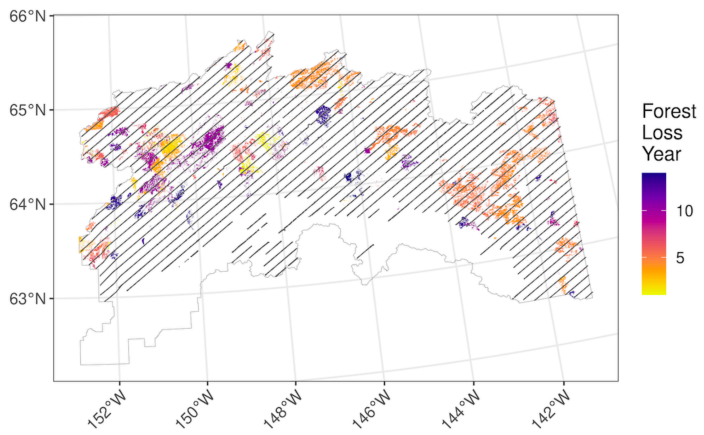
MESHED GAUSSIAN PROCESSES – EXAMPLE: CUBIC MGP

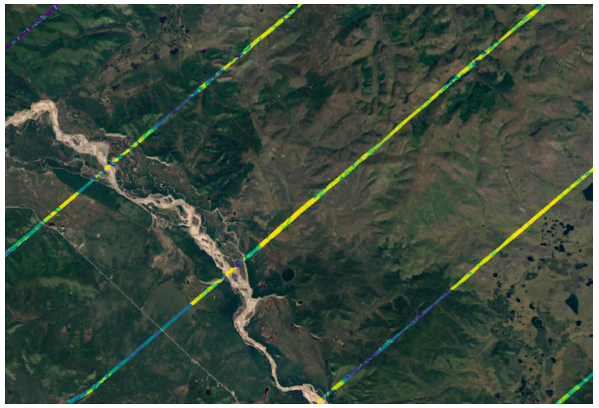
- **Sampling** w (steps 3 and 4) takes larger portion of time
- Luckily, QMGPs enable **parallel** sampling of w
- Blocking of the DAG improves ESS for posterior samples of w



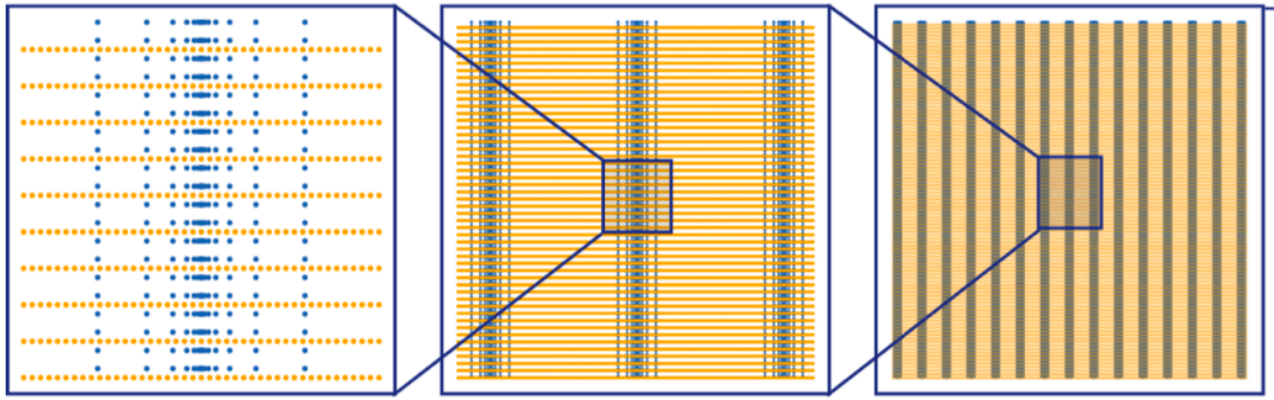
MESHED GAUSSIAN PROCESSES – EXAMPLE: CUBIC MGP

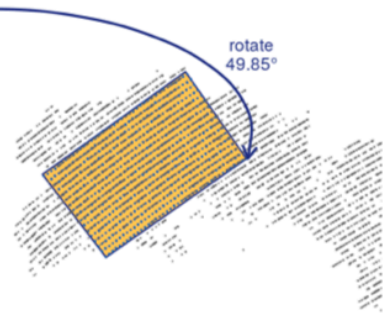
- Caching only requires a *patterned* grid, not necessarily regular (equally spaced points)
- Example: Tanana forest, Alaska. Data (5M pixels) on super narrow strips, 8km apart





Reference set as 2 overlapping grids. Domain partitioning following same pattern.





GRIPS: GRID-PARAMETRIZE-SPLIT

So far, we equated scalability with **speed**.

What about MCMC **efficiency** with big n and spatial multivariate models?

1. Avoid sampling $w(u)$

$$y(u) = x(u)^\top \beta + w(u) + \varepsilon(u),\ \varepsilon(u) \sim N(0, \tau^2),$$

location u: $w(u) = H_u w_j + v,\quad v \sim N(0, R_u)$

reference *block* j: $w_j = H_j w_{[j]} + v,\quad v \sim N(0, R_j)$

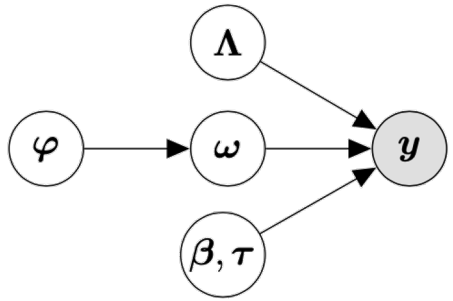
$$y(u) = x(u)^\top \beta + H_u w_j + \varepsilon(u),\ \varepsilon(u) \sim N(0, R_u + \tau^2),$$

reference *block* j: $w_j = H_j w_{[j]} + v,\quad v \sim N(0, R_j)$

2. Reparametrize the Matérn cross-correlation using micro-ergodic parameter

$$\widetilde{C}_j(\ell, \ell') = \frac{\rho(\ell, \ell'; \{\phi_j, \nu_j\})}{\phi_j^{2\nu_j}}$$

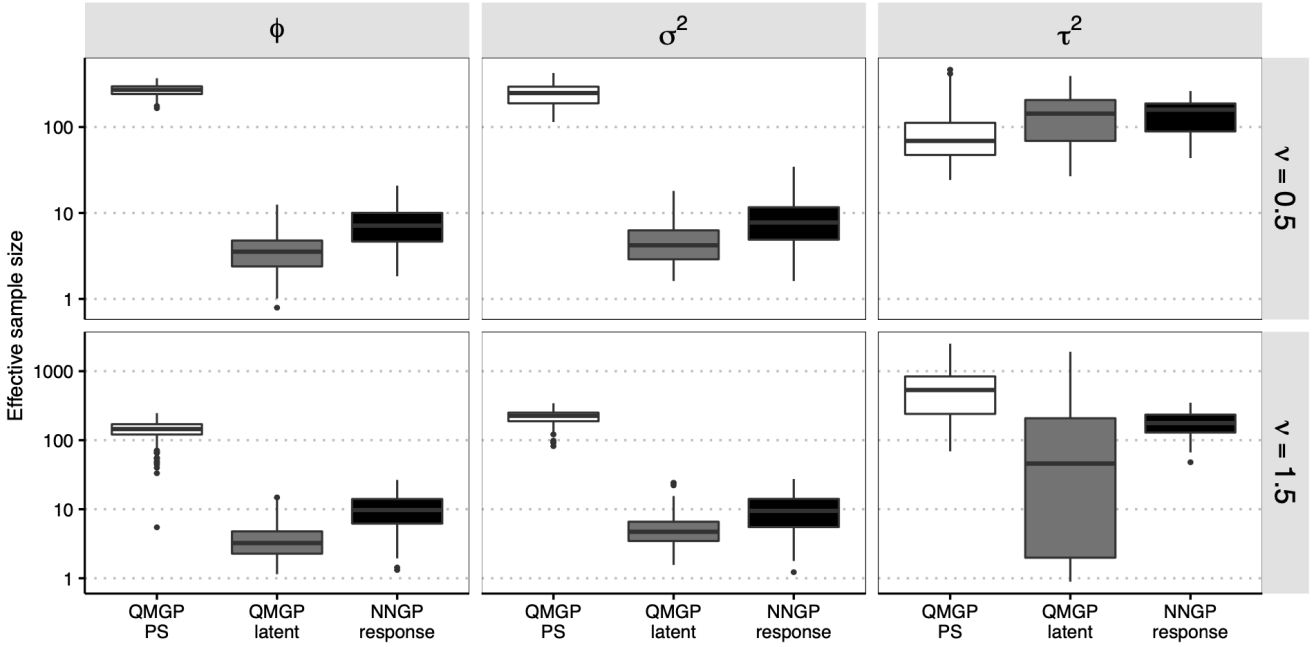
3. Split (expand) the process variance



GRIPS: GRID-PARAMETRIZE-SPLIT

Target: improved MCMC **efficiency** with big n and spatial multivariate models

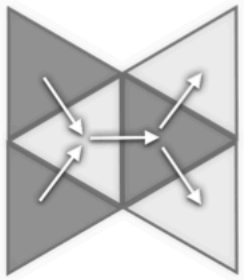
Comparing Effective Sample Size (ESS) and ESS/ second in estimating covariance parameters:



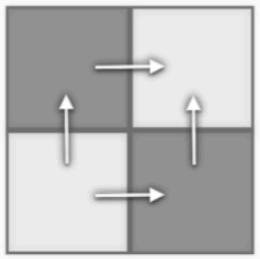
Efficiency		$\nu = 0.5$ ESS	$\nu = 0.5$ ESS/s	$\nu = 0.5$ Relative	$\nu = 1.5$ ESS	$\nu = 1.5$ ESS/s	$\nu = 1.5$ Relative
QMGP -PS	ϕ	**270.13**	**24.89**	**148.61**	**142.59**	**13.15**	**222.56**
	σ^2	**242.91**	**22.39**	**117.32**	**219.82**	**20.27**	**341.84**
	τ^2	101.94	9.36	3.04	**618.25**	**56.96**	**53.44**
QMGP latent	ϕ	3.96	0.28	1.69	3.87	0.28	4.70
	σ^2	4.97	0.36	1.87	5.58	0.40	6.75
	τ^2	**149.04**	**10.67**	**3.46**	215.96	15.48	14.52
NNGP response	ϕ	7.84	0.17	1.00	10.45	0.06	1.00
	σ^2	8.93	0.19	1.00	10.48	0.06	1.00
	τ^2	144.05	3.08	1.00	187.57	1.07	1.00

OTHER MGPS

So far, we only considered QMGPs. What about other DAGs and partitioning strategies?

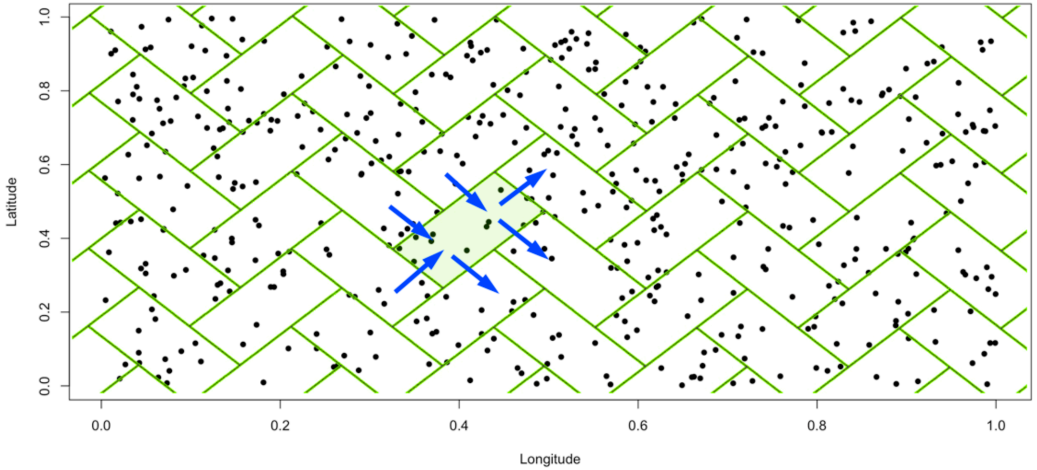


TriMGP ?



QMGP

HexMGP ?



FloorTileMGP ?

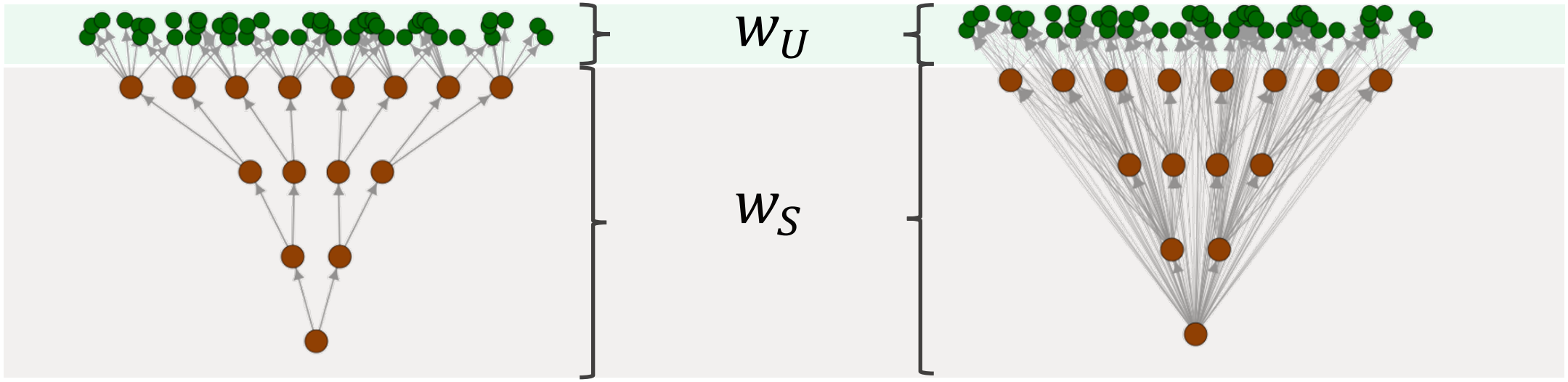
All of these lead to essentially the same properties (caching + parallel sampling).

SPAMTREES: SPATIAL MULTIVARIATE TREES

So far, we only considered QMGPs. What about other DAGs and partitioning strategies?

Recursive partitioning + Treed DAG + Multivariate outcomes = SPAMTREES

- Cost of computing is *less than* the cube of the size of parent sets
- "Automatic caching" without adding a grid (but less powerful)
- Favor longer range dependence



SPAMTREES: SPATIAL MULTIVARIATE TREES

Recursive partitioning + Treed DAG + Multivariate outcomes = SPAMTREES

- Cost of computing is *less than* the cube of the size of parent sets

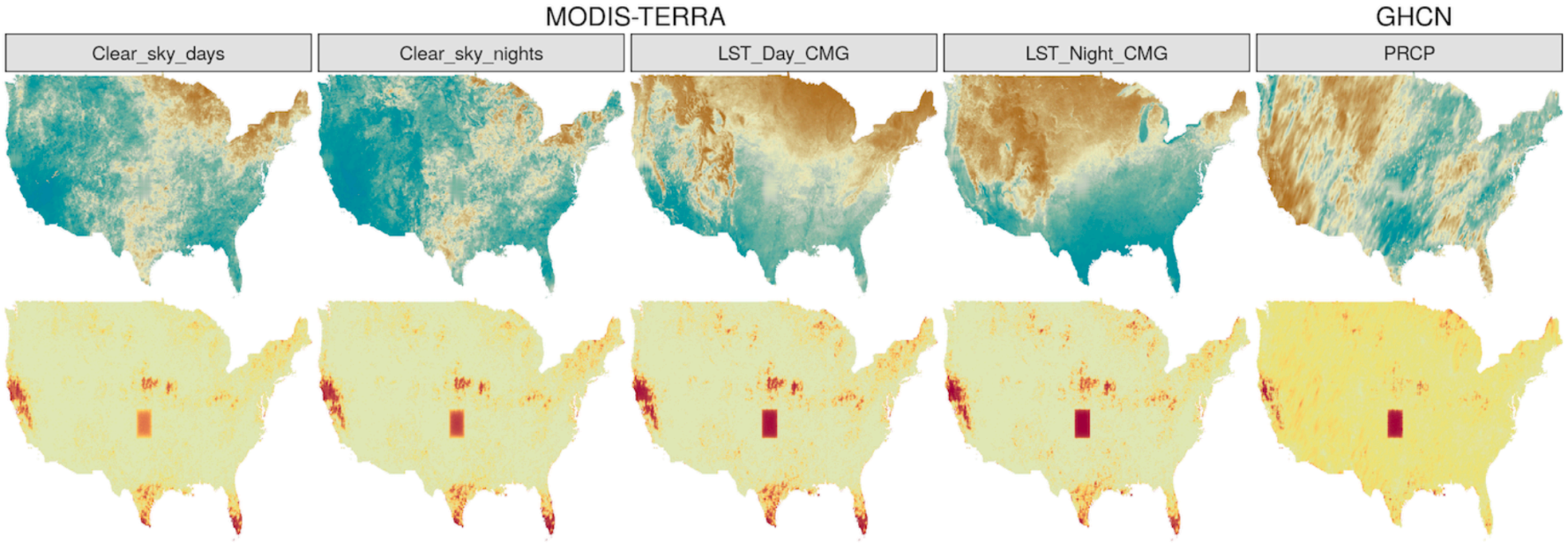
This is due to the following recurrent relation

$$\boldsymbol{C}_{[j]}^{-1} = \begin{bmatrix} \boldsymbol{C}_{[i]}^{-1} + \boldsymbol{H}_i^{\top}\boldsymbol{R}_i^{-1}\boldsymbol{H}_i & -\boldsymbol{H}_i^{\top}\boldsymbol{R}_i^{-1} \\ -\boldsymbol{R}_i^{-1}\boldsymbol{H}_i & \boldsymbol{R}_i^{-1} \end{bmatrix}$$

SPAMTREES: SPATIAL MULTIVARIATE TREES

Recursive partitioning + Treed DAG + Multivariate outcomes = SPAMTREES

Applied on misaligned multivariate data with large missing areas



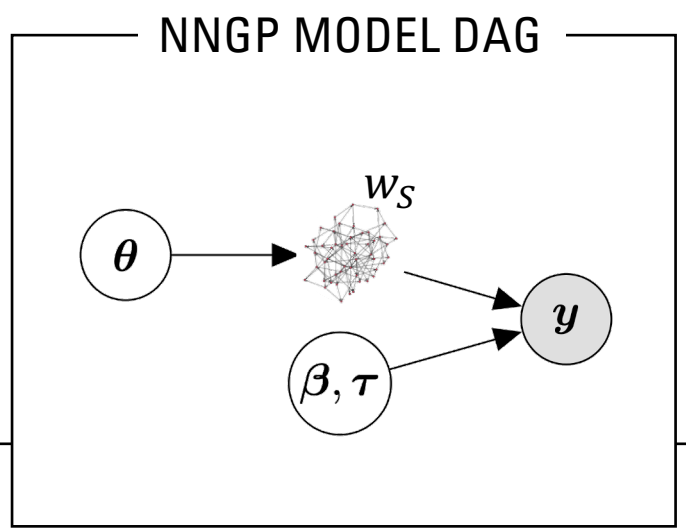
COMPARED

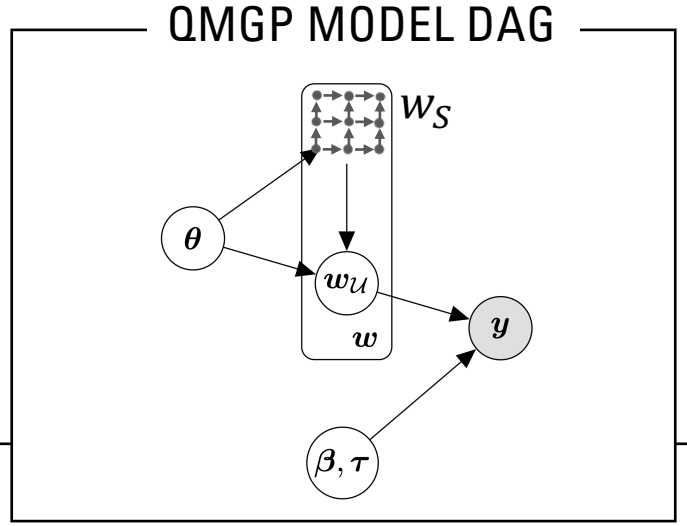
Nearest-neighbor GP (Datta et al 2016 JASA):

- defines valid process based on DAG for w
- DAG for w embedded in Bayesian model DAG
- sparse DAG built "automatically" from neighbors
- DAG knots 1:1 to elements of S
- Gibbs sampling for $w(s)$ using Gaussian full conditionals
- no need for reference set (the knots) being separate from the set of observed locations, i.e. $U = \emptyset$

Meshed GP (P et al 2020 JASA):

- defines valid process based on DAG for w
- DAG for w embedded in Bayesian model DAG
- sparse DAG **fixed beforehand**
- DAG knots 1:1 to **partitions of** S
- **Parallel** Gibbs sampling for **blocks of** w_j
- **advantages** arise with gridded knots
- **more advantages** with gridded data





BUT WHY SAMPLE?

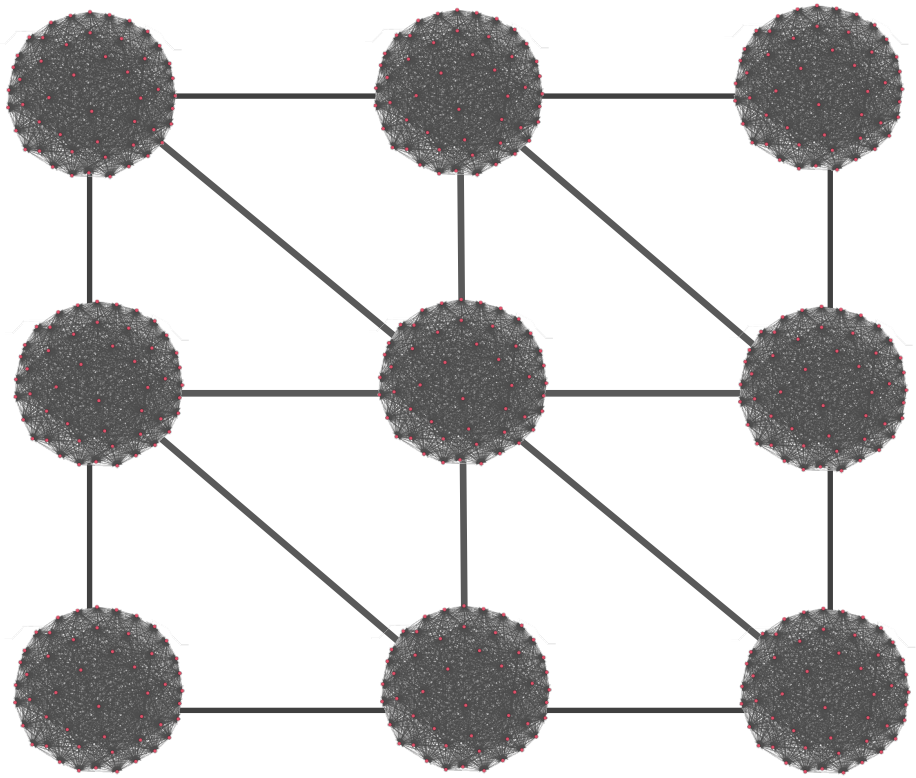
- "Vecchia" or DAG-based models (e.g. NNGPs) correspond to sparse Gaussian precision matrices
- MGP also leads to sparse precision matrices

Computing big data models without sampling latent effects:

- MGP model of the response
- MGP "response-conjugate" or "latent-conjugate" models

Unlike when sampling w, there are **no substantial computing advantages** compared to NNGPs.

BUT WHY SAMPLE?



Moral graph for QMGPs

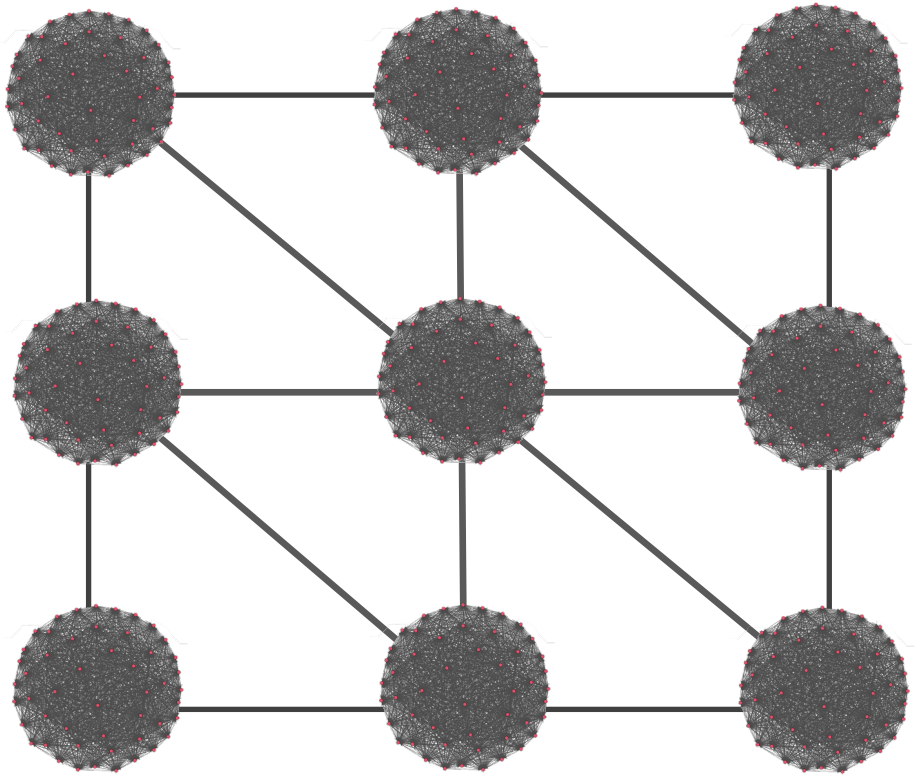
We focus on "nice" DAGs:

- known properties
- predictable algorithm behavior

With **Gaussian** outcomes:

- each node has a Gaussian full conditional
- **Gibbs** steps to update w
- posterior for w "looks like the prior"

BUT WHY SAMPLE?



Moral graph for QMGPs

We focus on "nice" DAGs:

- known properties
- predictable algorithm behavior

With **non-Gaussian** outcomes:

- each node has *a* full conditional 🤓
- **HMC** steps to update *w*
- posterior for *w* "looks like the prior"
- *spatially-meshed* MCMC

SPATIALLY MESHED MCMC

Simplified Riemannian Manifold MALA steps to update the full conditionals

- Very minor overhead in computation speed compared to the Gaussian case
- Theoretical connection with the Gaussian Gibbs sampler
- Works with multivariate outcomes
- Works with multi-type outcomes
- Works with misaligned outcomes
- Works with non-Gaussian latent process (e.g. T-process)
- …