

GRAPH TRACKING
IN DYNAMIC PROBABILISTIC PROGRAMS
VIA SOURCE TRANSFORMATIONS

PHILIPP GABLER

2021-03-11

ABSTRACT

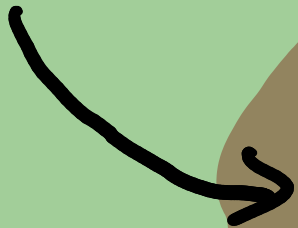
- Q: HOW TO ADD STRUCTURE TO TURING MODELS?
- OUTCOMES:
 - "EXTENDED WENGERT LIST" TRACKING TECHNIQUE
 - EXTRACT DEPENDENCY GRAPHS (SLICES) FROM TURING MODELS
 - AUTOMATIC GIBBS SAMPLER

ROADMAP

MODELS &
INFERENCE



PROBABILISTIC
PROGRAMMING



TRACKING
MODELS



DERIVING
SAMPLERS

GRAPH
TRACKING

EXTENDED
WENGERT
LISTS



DEPENDENCY
TRACKING
IN TURING

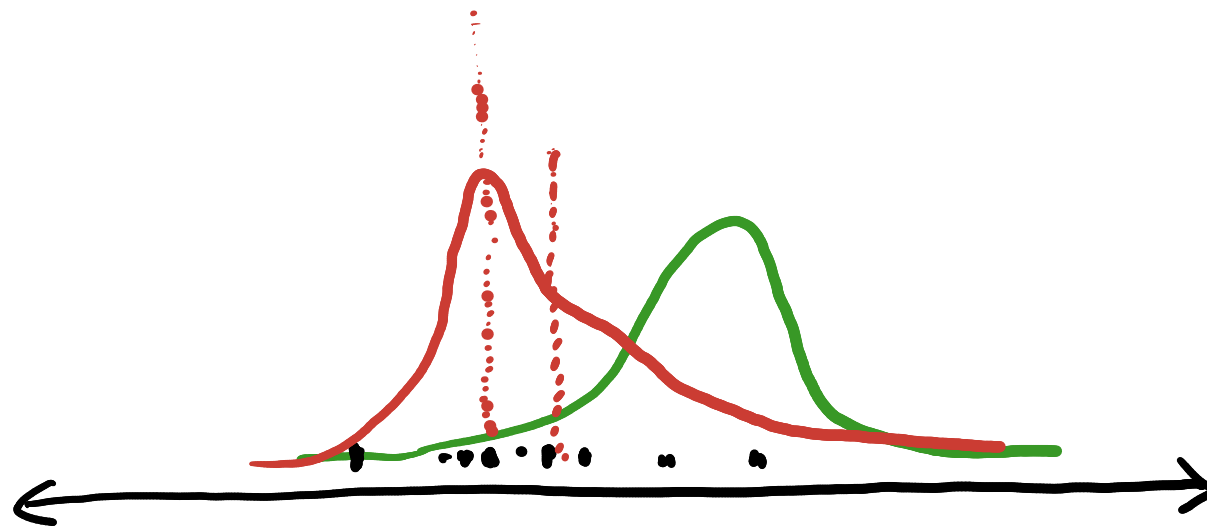


AUTOMATIC
GIBBS COND.S

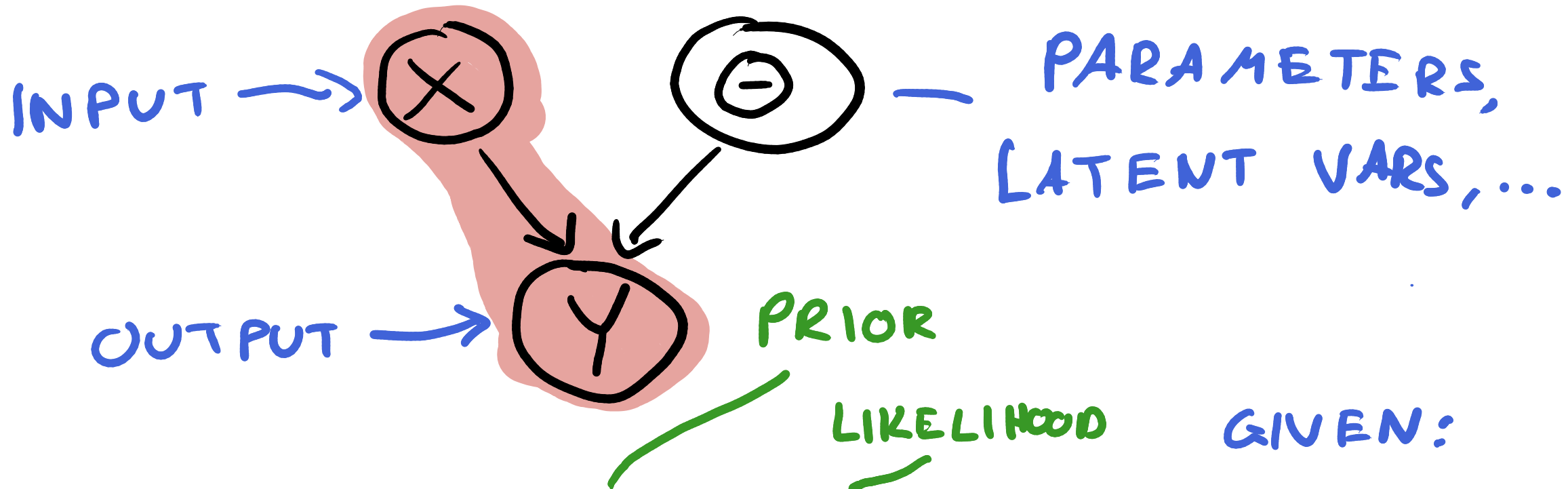


⑦

BAYES-ICS



GENERATIVE MODELS



$$P(Y, \Theta | X) = P(\Theta) P(Y | X, \Theta) \leftarrow \text{GENERATING PROCESS}$$

$$P(\Theta | \text{D}) = \frac{P(Y, \Theta | X)}{P(Y | X)}$$

OBSERVED DATA
 $D = \{(X_1, Y_1), \dots\}$

EVIDENCE

WANT: POSTERIOR

POSTERIOR EXPECTATION

- $E[f(\theta) | D]$ ^{DATA}
 $= \int f(\vartheta) p(\vartheta | D) d_{\nu}(\vartheta)$ ^{POSTERIOR}

- INTERESTING QUANTITIES (OFTEN)
ARE INTRACTABLE INTEGRALS!

E.G. PPD, CREDIBLE INTERVALS, MAP,
MODEL COMPARISON,...

APPROXIMATE INFERENCE

• VI: MINIMIZE

$$D(q_{\lambda}(D) \parallel p(\theta|D))$$

• SAMPLING:

USE LLN!

$$I^{(n)}(f) = \frac{1}{N} \sum_n f(Y^{(n)})$$

$$\xrightarrow{\text{a.s.}} \mathbb{E}[f(\theta) | D]$$

THE MARKOV CHAIN

o CONSTRUCT TRANSITION KERNEL:

$$\begin{aligned} P[Y^{(k+1)} \in d_Y \mid Y^{(k)} = y^{(k)}, \dots, Y^{(1)} = y^{(1)}] \\ = K(y^{(k)}, d_Y) \end{aligned}$$

$$\text{s.t. } \underbrace{p(\mathcal{Y} \mid D) K(\mathcal{Y}, S)}_{\triangleq \pi K} = P[\Theta \in S \mid D] \quad \forall S$$

o SCHEME:

- PROPOSE NEW $Y^{(i)}$
FROM $Y^{(i-1)}$

- ACCEPT/REJECT

LIVE
DEMO

KERNELS \approx SAMPLERS

- METROPOLIS - HASTINGS, HMC, ...
- OFTEN EASY: GIBBS CONDITIONALS!

$$p(\vartheta_1 | \vartheta_2, D) \quad p(\vartheta_2 | \vartheta_1, D)$$

NO ACCEPTANCE STEP!

- WITHIN-GIBBS: MIX SAMPLERS
PER VARIABLE!

②

PROBABILISTIC PROGRAMMING

$$X \sim \mathcal{N}(\mu, \sigma_1)$$

$$Y \sim \mathcal{N}(x, \sigma_2)$$

PROBABILISTIC PROGRAMS I

HOW TO SPECIFY MODELS?

PARAMETERS

OBSERVED
DATA

```
@model function hierarchical_gaussian(x)
  λ ~ Gamma(2.0, inv(3.0))
  m ~ Normal(0, sqrt(1 / λ))
  x ~ Normal(m, sqrt(1 / λ))
end
```

JULIA FUNCTIONS

PROBABILISTIC PROGRAMS II

```
@model function normal_mixture(x, K, m, s,  $\sigma$ )
```

PARAMETERS

```
  N = length(x)
```

DATA STRUCTURES

```
   $\mu$  = Vector{Float64}(undef, K)
```

```
  for k = 1:K
```

```
     $\mu[k] \sim \text{Normal}(m, s)$ 
```

MUTATION

```
  end
```

```
  z = Vector{Int}(undef, N)
```

```
  for n = 1:N
```

```
    z[n]  $\sim$  Categorical(K)
```

```
  end
```

```
  for n = 1:N
```

```
    x[n]  $\sim$  Normal( $\mu[z[n]]$ ,  $\sigma$ )
```

```
  end
```

```
  return x
```

```
end
```

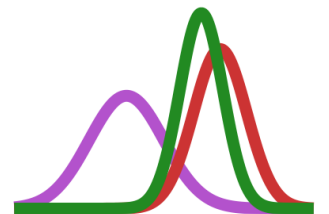
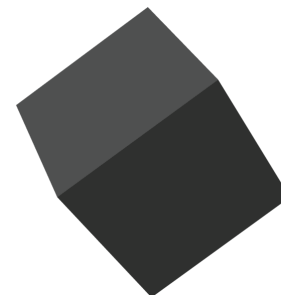
LOOPS / CONTROL FLOW

PPL CHARACTERISTICS

- DENSITY EVALUATION
- MODEL STRUCTURE
- AUTOMATIC DIFFERENTIATION
- SAMPLING, DATA GENERATION
- DIAGNOSTICS, EVALUATION
- PROGRAMMING & EXTERNAL LIBS
- COMPOSITION

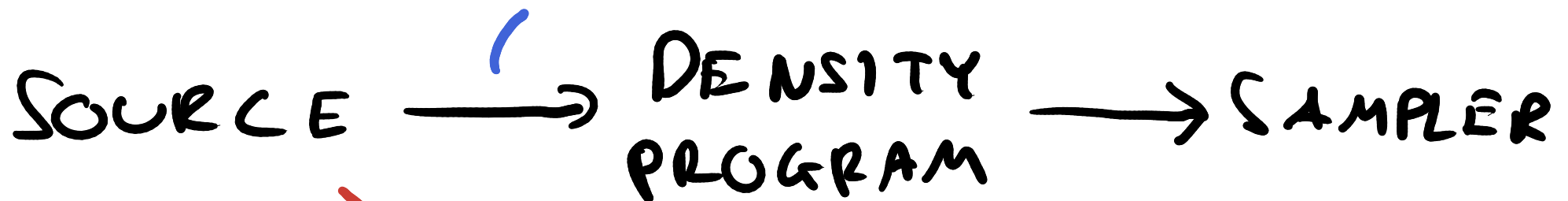


PYMC3



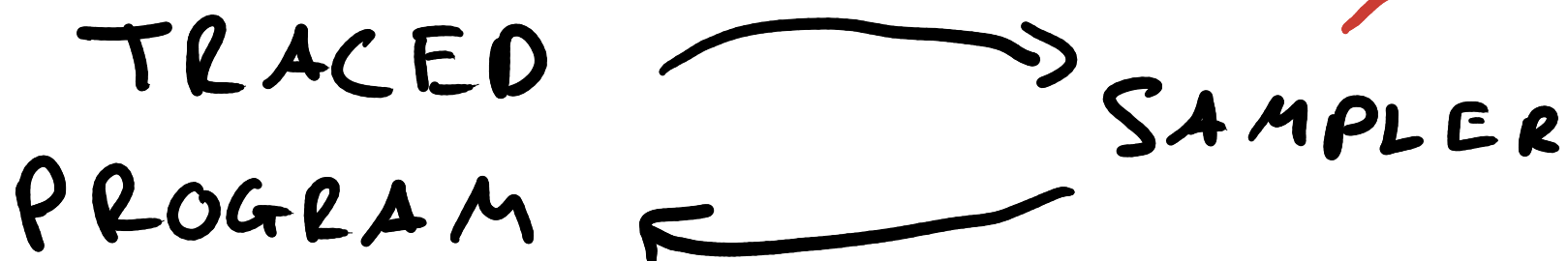
PPL IMPLEMENTATIONS

METAPROGRAMMING



RESTRICTED
DYNAMICS

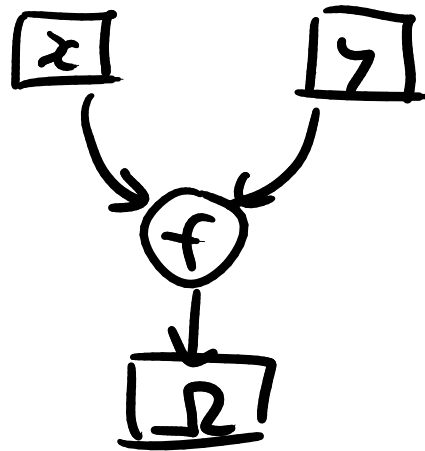
NO STRUCTURE



CALLBACKS

3

GRAPH TRACKING



COMPUTATION GRAPHS

$g(\sin(x), y)$

EXPRESSION /
PROGRAM

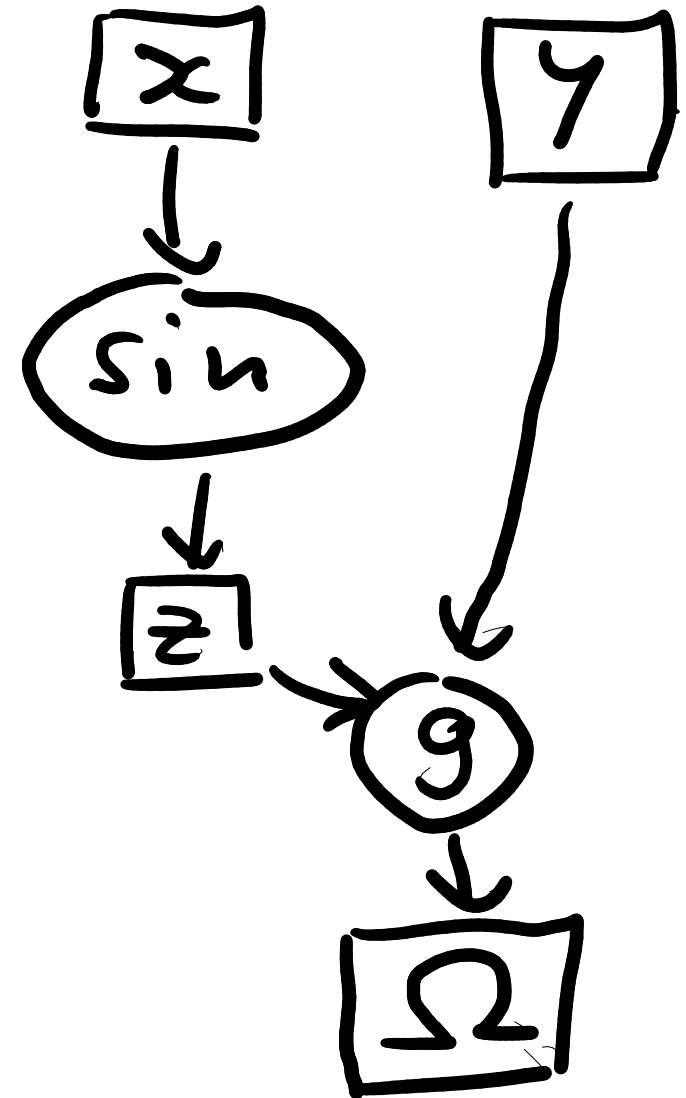
$x = ?$

$y = ?$

$z = \sin(x)$

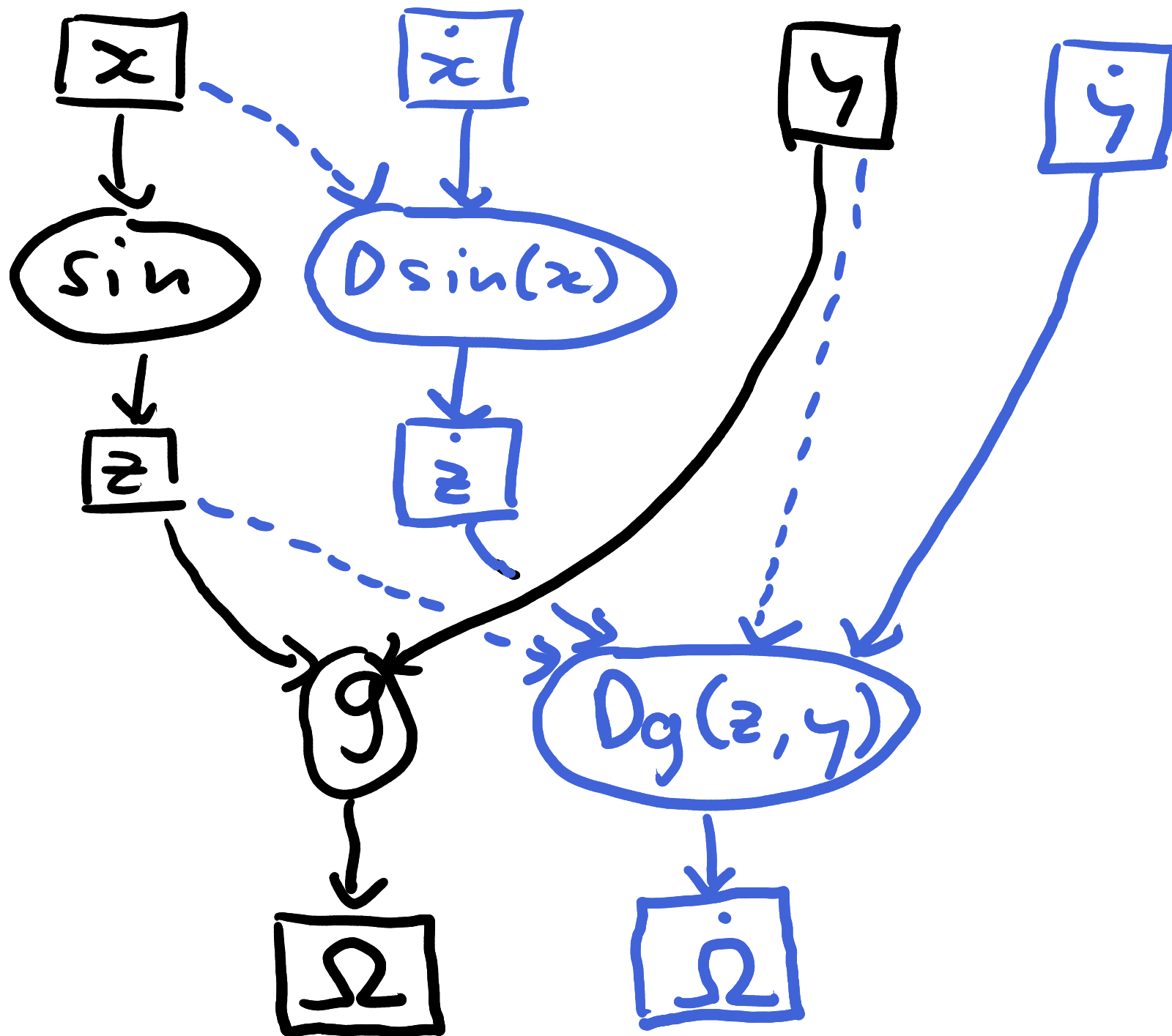
$\Omega = g(z, y)$

SSA FORM



GRAPH

GRAPHS IN AD



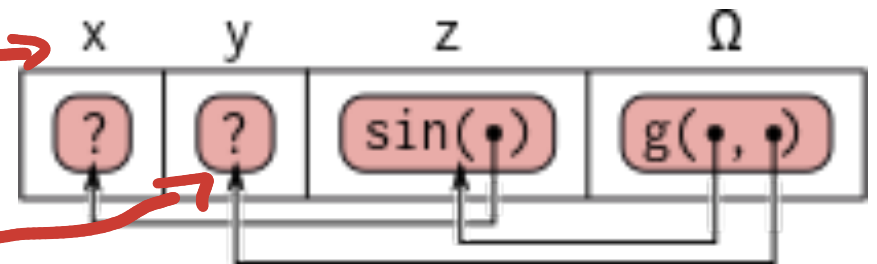
IMPLEMENTATION TECHNIQUES

o OPERATOR OVERLOADING:

$x = \text{TRACKED}(?)$

$y = \text{TRACKED}(?)$

\vdots



WENGERT LIST,
AKA TAPE

o SOURCE TRANSFORMATION:

$$x = ?$$

$$\dot{x} = D_1$$

$$y = ?$$

$$\dot{y} = D_2$$

$$z = \sin(x)$$

$$\dot{z} = D\sin(z)(\dot{x})$$

$$\Omega = g(z, y)$$

$$\dot{\Omega} = Dg(z, y)(\dot{z}, \dot{y})$$

JULIA IR

$$f(x, y) = y > 0 ? g(\sin(x), y) : y$$



1: ^f(^x%1, ^y%2, %3)

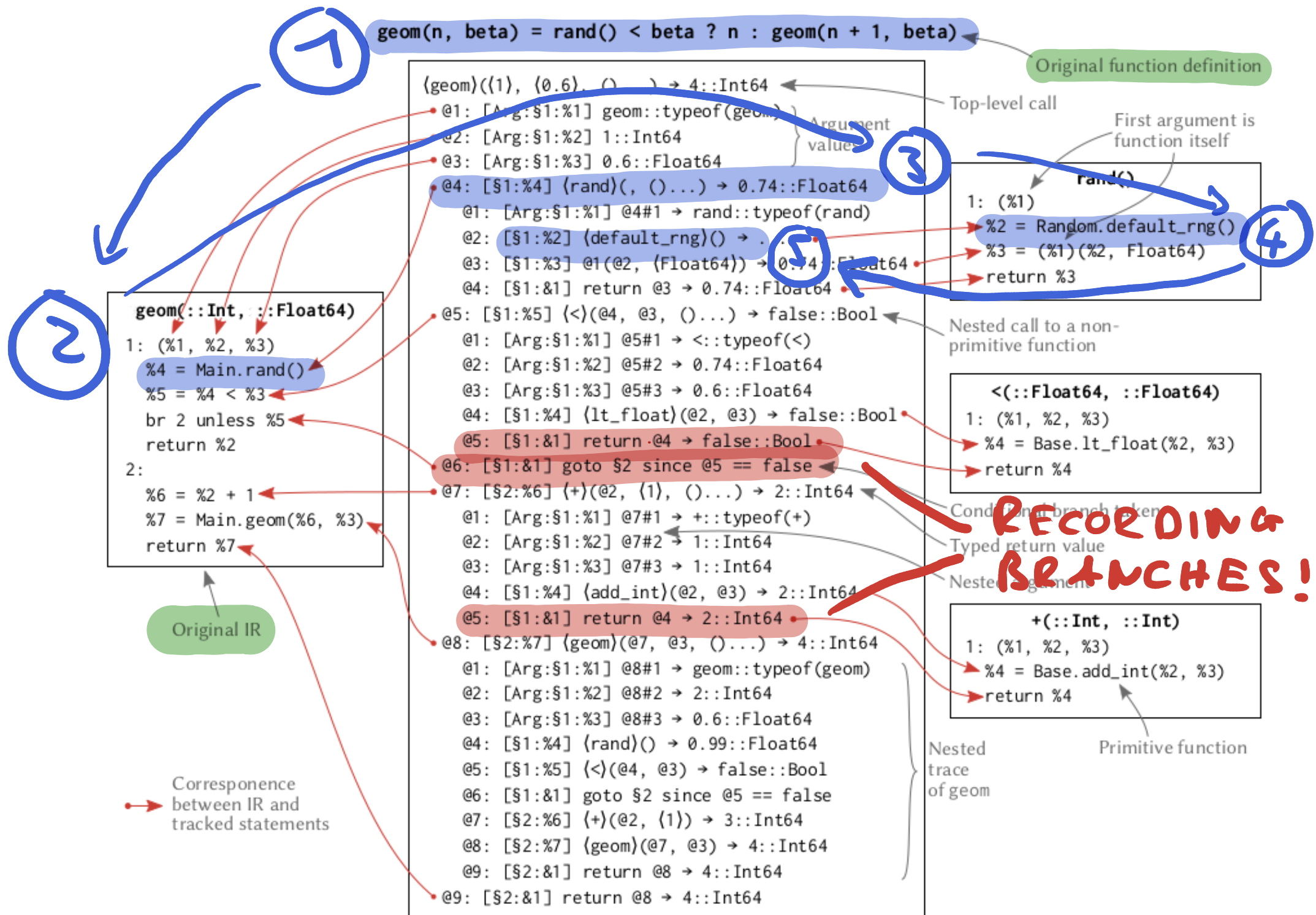
%4 = %3 > 0
br 3 unless(%4)

3: return %3

2:

%5 = sin(%2) ← z = sin(x)
%6 = g(%5, %3) ← Ω = g(z, y)
return %6

EXTENDED WENGERT LIST



IR TRANSFORMATION

← TRACKED IR

```
%15 = record!(%5, %14)
%16 = TapeConstant(Main.rand)
%17 = Base.tuple()
%18 = trackedcall(%5, %16, %17, $(QuoteNode($1:%4)))
%19 = record!(%5, %18)
%20 = TapeConstant(Main.:<)
%21 = trackedvariable(%5, $(QuoteNode(%4)), %19)
%22 = trackedvariable(%5, $(QuoteNode(%3)), %3)
%23 = Base.tuple(%21, %22)
%24 = trackedcall(%5, %20, %23, $(QuoteNode($1:%5)))
%25 = record!(%5, %24)
%26 = Base.tuple()
%27 = trackedvariable(%5, $(QuoteNode(%5)), %25)
%28 = trackedjump(%5, 2, %26, %27, $(QuoteNode($1:&1)))
%29 = trackedvariable(%5, $(QuoteNode(%2)), %2)
%30 = trackedreturn(%5, %29, $(QuoteNode($1:&2)))
br 2 (%28) unless %25
br 3 (%2, %30)
```

Actual jump is recorded

↓

```
1: (%1, %2, %3)
%4 = Main.rand()
%5 = %4 < %3
br 2 unless %5
return %2
```

ORIGINAL
← IR

Jumps and returns are passed down to the next block

↓

```
3: (%46, %47)
%48 = record!(%5, %47)
return %46
```

Special extra block
for return values

④

TRACKING

TURING MODELS

$$\langle 1 \rangle = \{1, \langle 2 \rangle\}$$



$$\langle 3 \rangle = \text{sqvt}(\langle 1 \rangle)$$



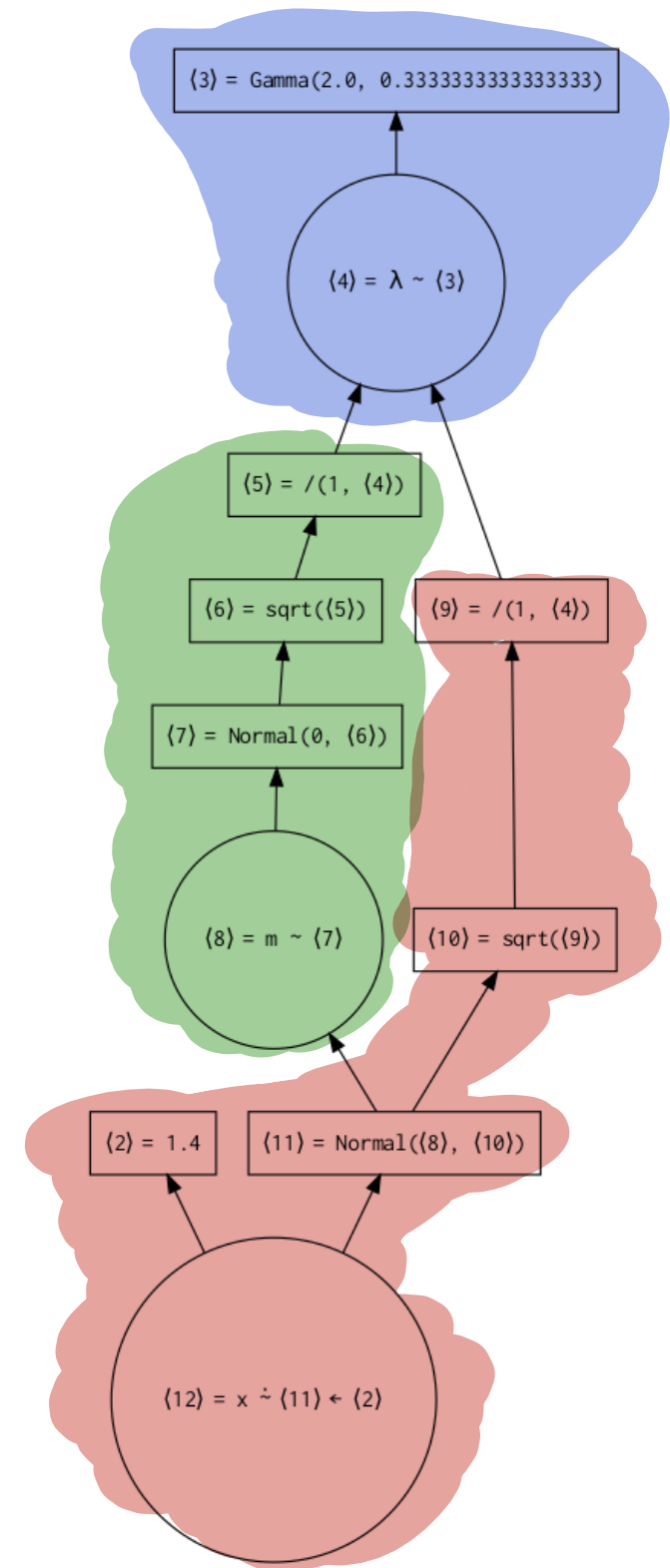
$$\langle 4 \rangle = \mathcal{V}(0, \langle 3 \rangle)$$



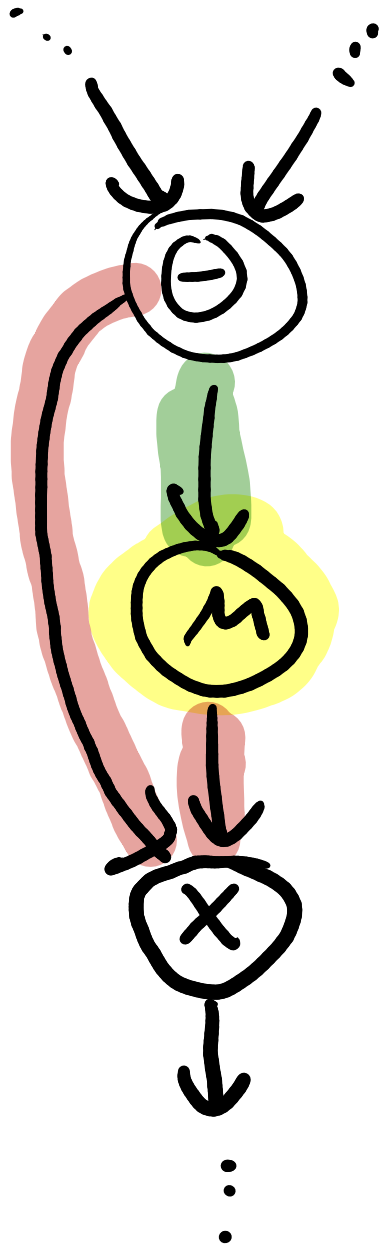
$$\langle 5 \rangle = m \sim \langle 4 \rangle$$

DEPENDENCY EXTRACTION

```
@model function hierarchical_gaussian(x)
   $\lambda \sim \text{Gamma}(2.0, \text{inv}(3.0))$ 
   $m \sim \text{Normal}(0, \text{sqrt}(1 / \lambda))$ 
   $x \sim \text{Normal}(m, \text{sqrt}(1 / \lambda))$ 
end
```



(DISCRETE) GIBBS CONDITIONALS



$$p(m, \theta, x) \\ = p(\theta) p(m | \theta) p(x | m, \theta)$$

MARKOV BLANKET /

$$p(m | \theta, x) = \frac{p(m | \theta) p(x | m, \theta)}{Z}$$

$$Z = \sum_m p(m | \theta) p(x | m, \theta)$$

A TEST MODEL

```
@model function gmm(x, K)
  N = length(x)
  w ~ Dirichlet(K, 1/K) # Cluster association prior
  z ~ filldist(Categorical(w), N) # Cluster assignments
   $\mu$  ~ filldist(Normal(0.0, s1_gmm), K) # Cluster centers

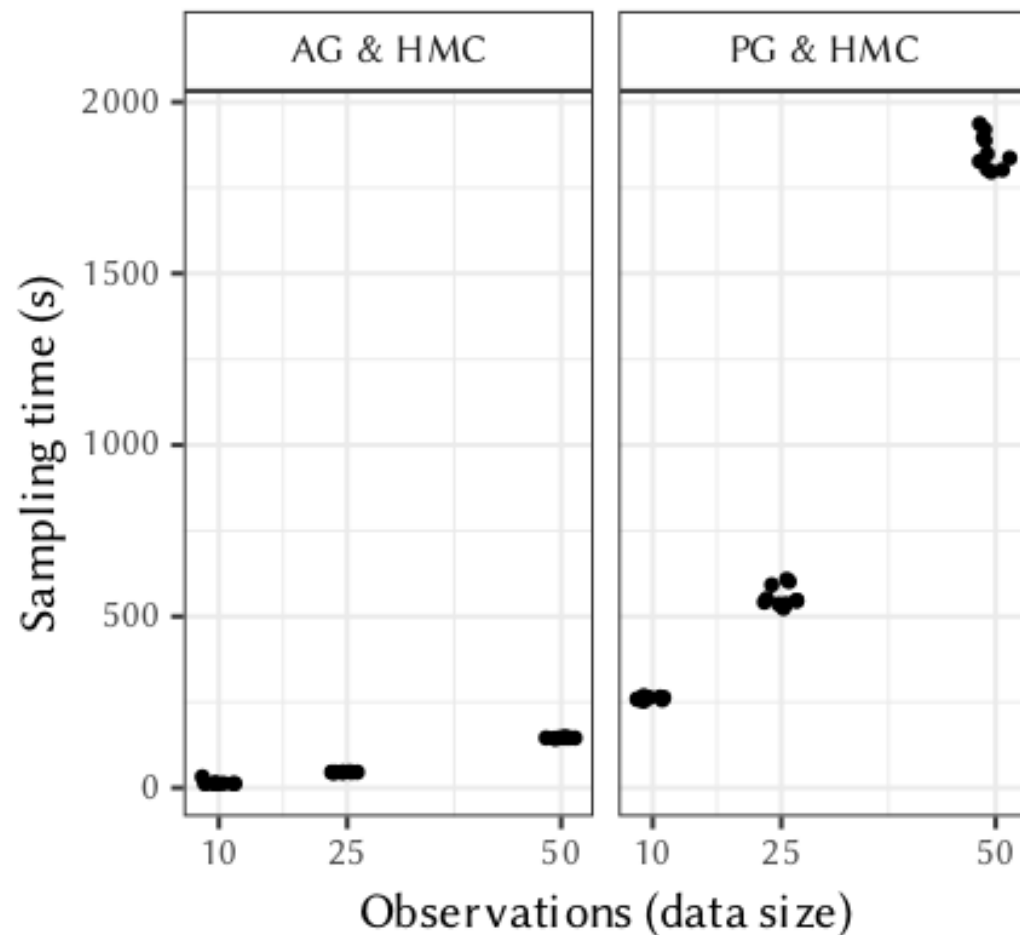
  for n = 1:N
    x[n] ~ Normal( $\mu$ [z[n]], s2_gmm) # Observations
  end
end
```

z: AUTO GIBBS vs. PARTICLE GIBBS

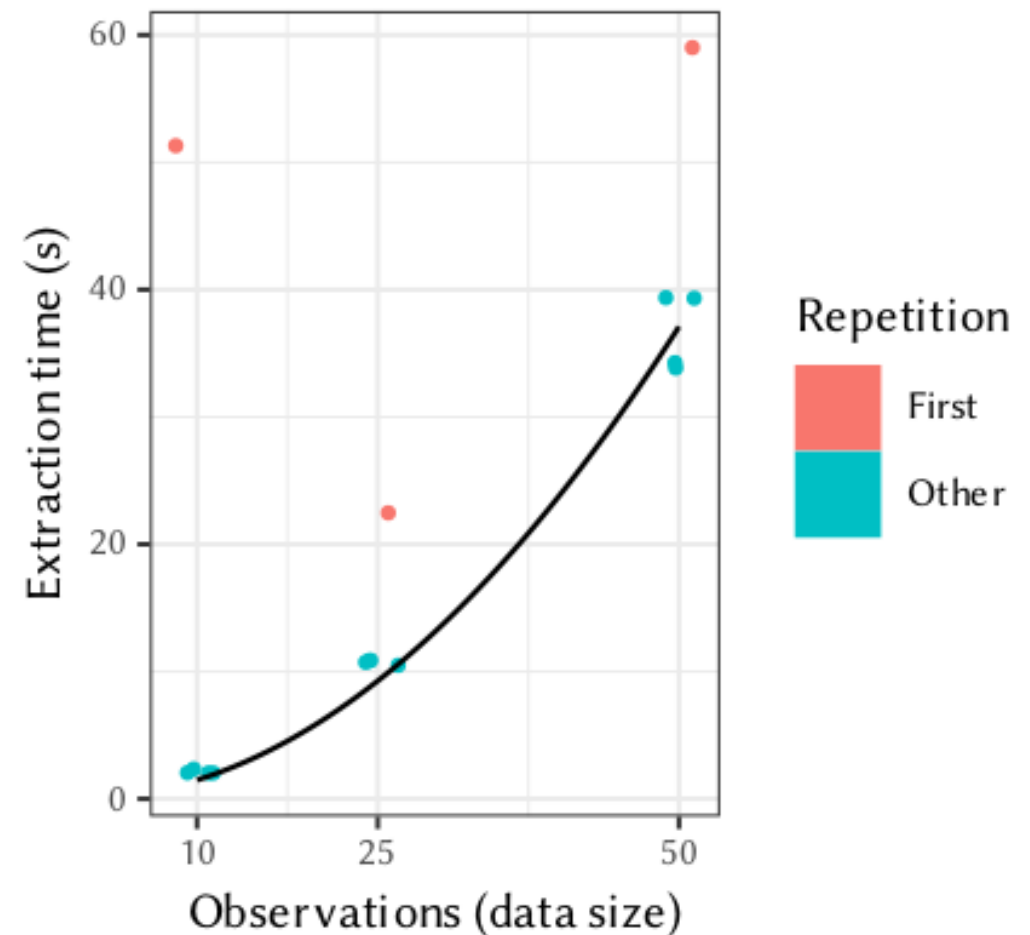
w, μ : HMC

EVALUATION: TIME

Sampling times for GMM

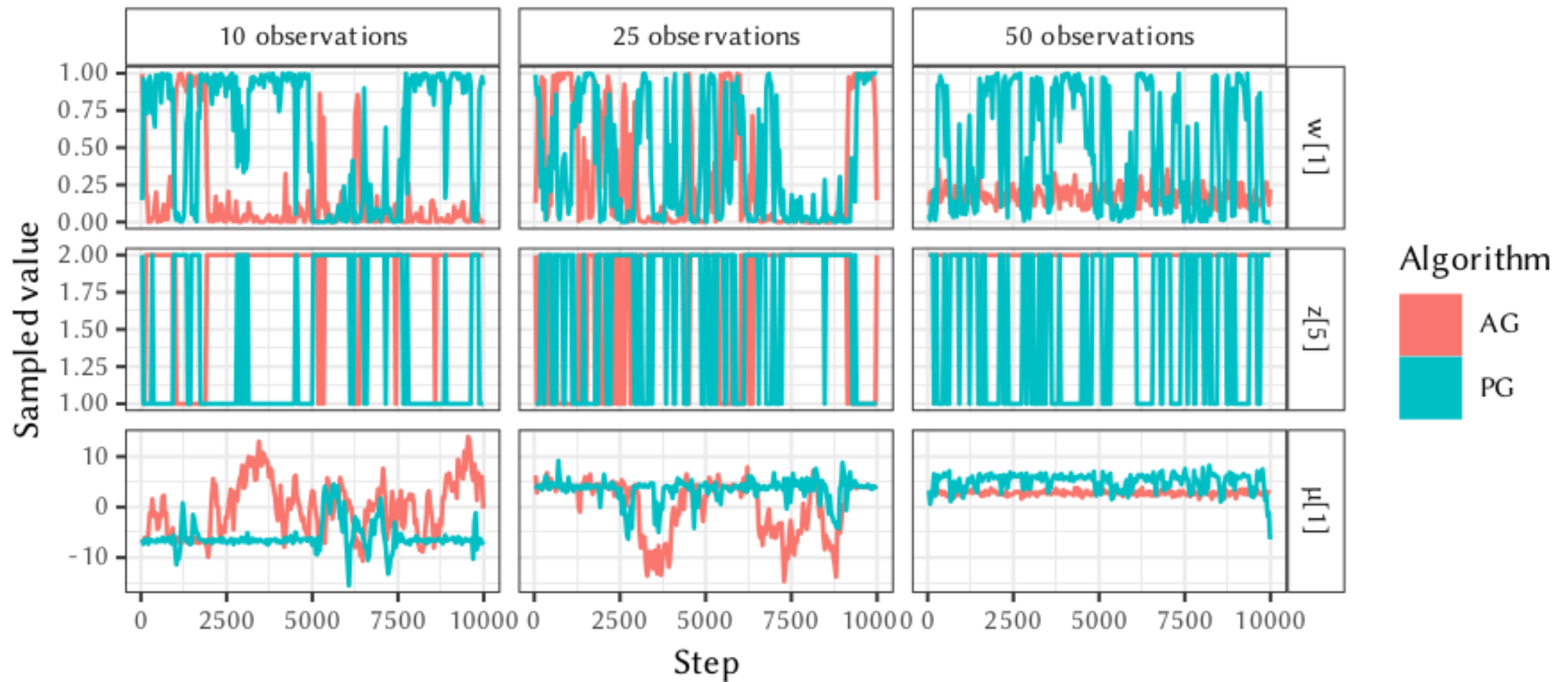


AG extraction times for GMM



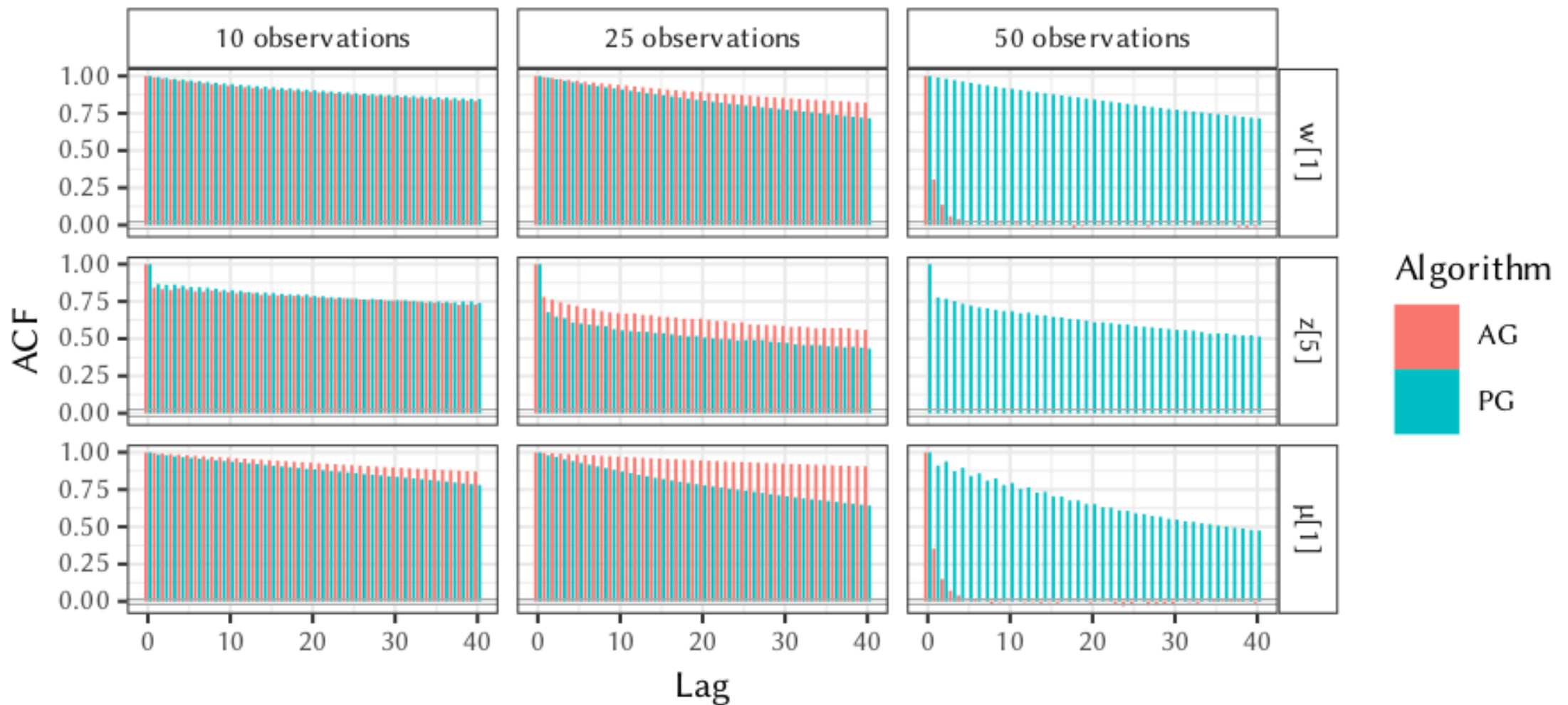
EVALUATION: CHAINS

Chain comparisons for GMM



EVALUATION: CONVERGENCE

Autocorrelation estimate for GMM

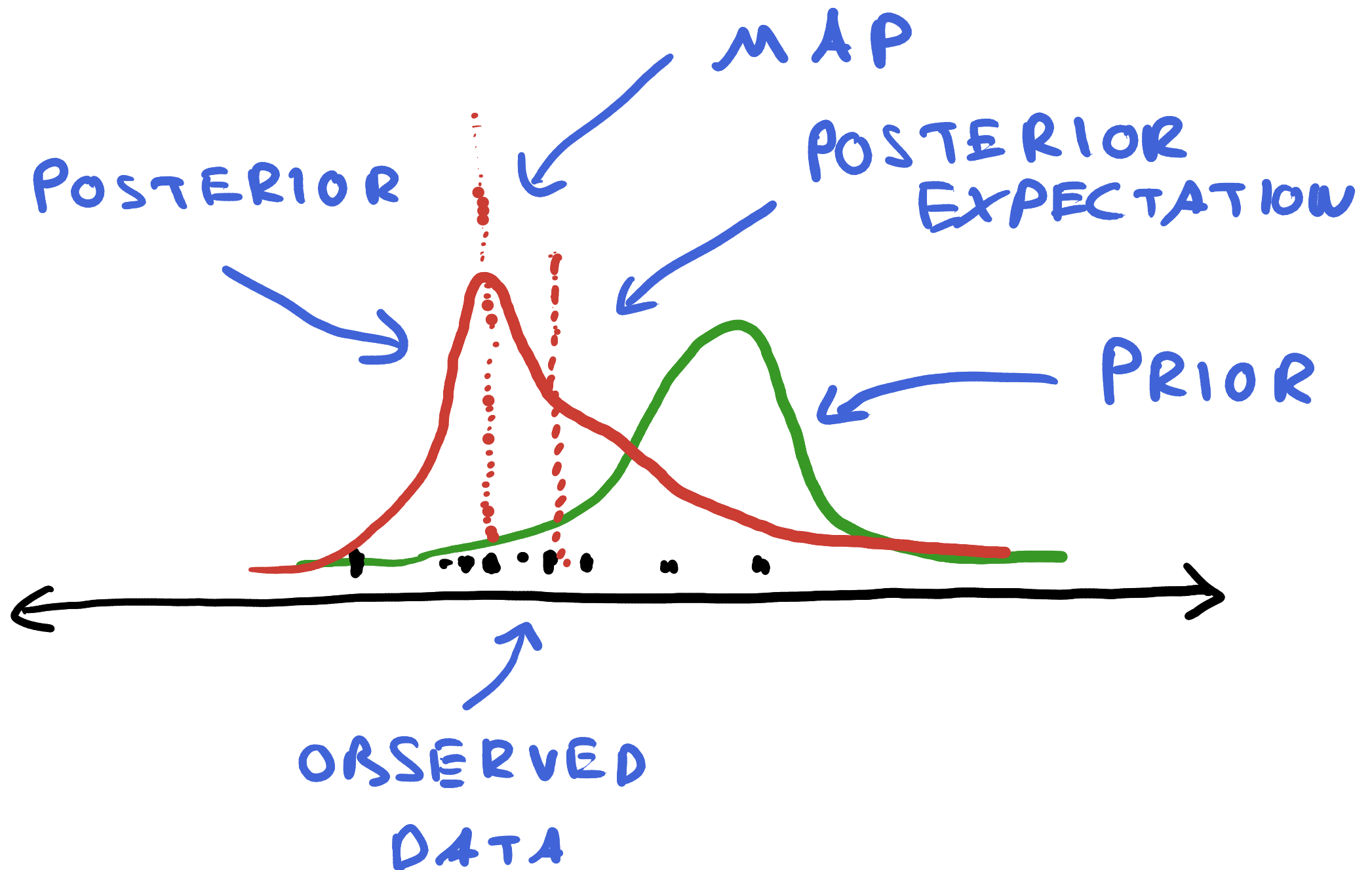


CONCLUSIONS

- STATIC DEPENDENCIES, FINITE CONDITIONALS: 😊
- SLICING DYNAMIC MODELS: 😊
- RECOVERING DYNAMIC STRUCTURE: 😞
- FUTURE PROOF: 😐?

* scratch *

"BAYES-ICS"



PREDICTION

- PLUG IN ESTIMATOR:

$$\hat{p}(y|x, D) = p(y|x, \vartheta^*(D))$$

arg max!

- POSTERIOR PREDICTIVE:

$$p(y|x, D) = \int p(y|x, \vartheta) p(\vartheta|D) d\nu(\vartheta)$$

POSTERIOR EXPECTATION!