Graph Tracing in Dynamic Probabilistic Programs via Source Transformations

NLP Version

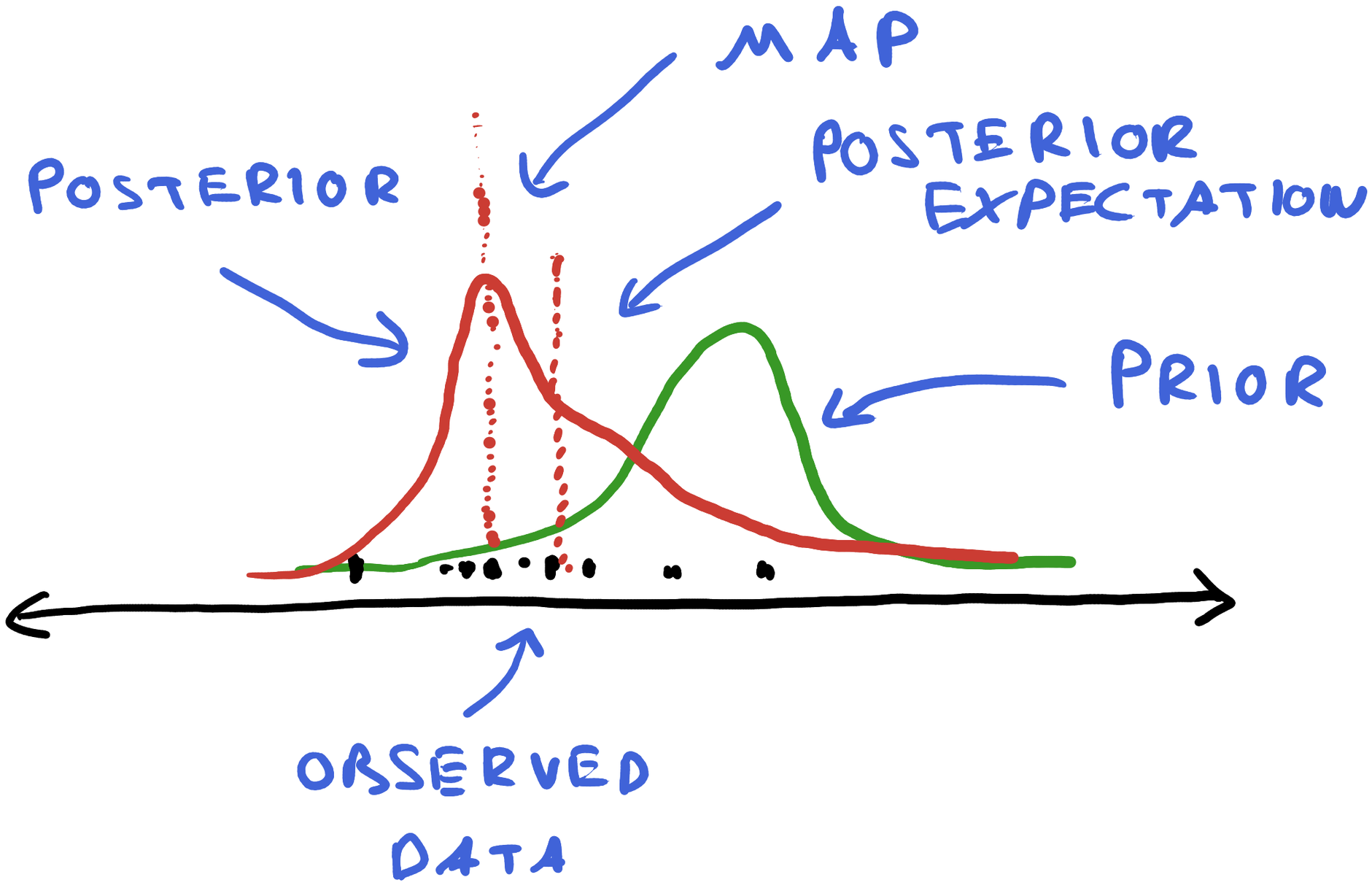
Philipp Gabler

2021-02-17

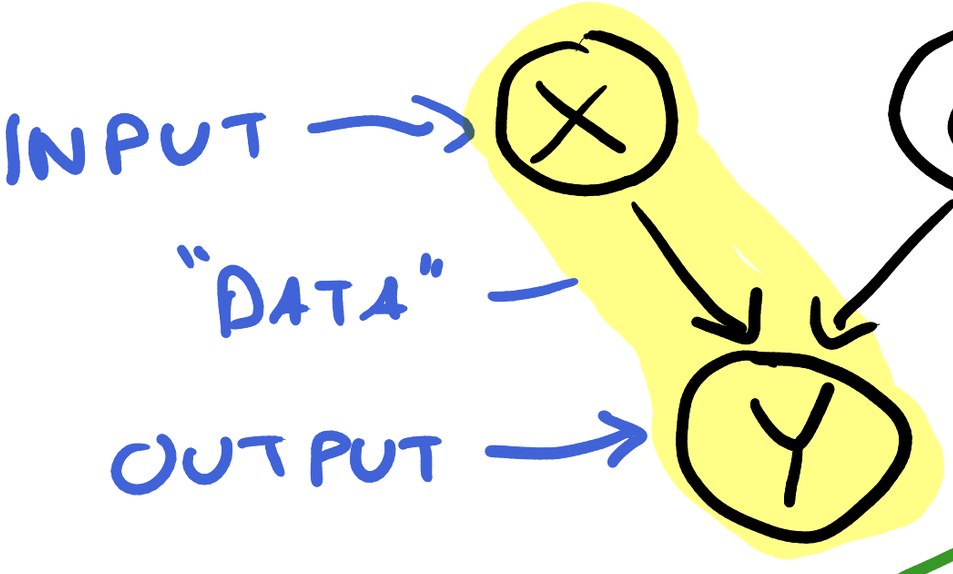
OVERVIEW

1. GENERATIVE MODELS, BAYESIAN INFERENCE
2. PROBABILISTIC PROGRAMMING
3. GRAPH TRACKING
4. TRACKING MODELS, DERIVING GIBBS SAMPLERS

"BAYES-ICS"



GENERATIVE MODELS



PRIOR

LIKELIHOOD

$$P(Y, \Theta | X) = P(\Theta) P(Y | X, \Theta)$$

GIVEN: GENERATING PROCESS

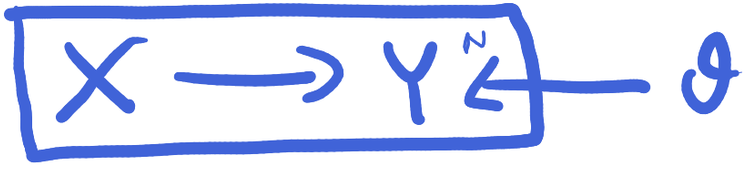
$$P(\Theta | X, Y) = \frac{P(Y, \Theta | X)}{P(Y | X)}$$

WANT: POSTERIOR

EVIDENCE

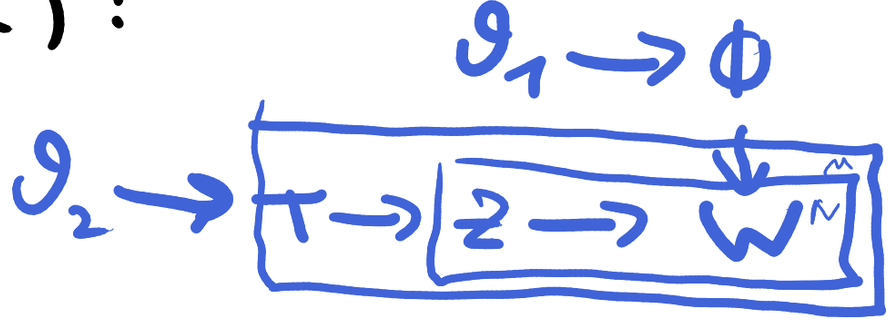
MODELS IN NLP

- REGRESSION:

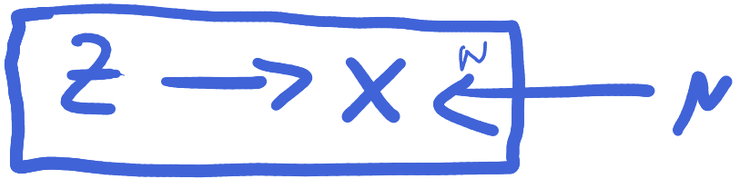


- LM:

- TOPIC MODEL (LDA):



- CLUSTERING (PCA, GMM, ...):



- PCFG:

Posterior Expectation

$$\mathbb{E}[f(\theta) \mid D]$$

DATA

$$= \int f(\vartheta)\, p(\vartheta \mid D)\, d\mu(\vartheta)$$

INTERESTING QUANTITIES ARE INTRACTABLE INTEGRALS!

Approximate Inference

- VI: Minimize

$$D(q_\lambda(\vartheta) \,\|\, p(\vartheta \mid D))$$

- Sampling:

USE LLN!

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

$$\xrightarrow{a.s.} \mathbb{E}[f(\Theta) \mid D]$$

THE MARKOV CHAIN

- CONSTRUCT TRANSITION KERNEL:

$$\mathbb{P}[Y^{(k+1)} \in dy \mid Y^{(k)} = y^{(k)}, \dots, Y^{(1)} = y^{(1)}]$$
$$= K(y^{(k)}, dy)$$
$$\text{s.t. } \underbrace{p(\vartheta \mid D) K(\vartheta, S) = \mathbb{P}[\Theta \in S \mid D] \ \forall S}_{\widehat{=}\ \pi K = \pi}$$

- SCHEME:
  - PROPOSE NEW $Y^{(i)}$ FROM $Y^{(i-1)}$
  - ACCEPT / REJECT

LIVE DEMO

KERNELS ≈ SAMPLERS

- METROPOLIS-HASTINGS, HMC, ...

- OFTEN EASY: GIBBS CONDITIONALS!

$$p(\vartheta_1 \mid \vartheta_2, D) \qquad p(\vartheta_2 \mid \vartheta_1, D)$$

NO ACCEPTANCE STEP!

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, σ)
    N = length(x)

    μ = Vector{Float64}(undef, K)
    for k = 1:K
        μ[k] ~ Normal(m, s)
    end

    z = Vector{Int}(undef, N)
    for n = 1:N
        z[n] ~ Categorical(K)
    end

    for n = 1:N
        x[n] ~ Normal(μ[z[n]], σ)
    end

    return x
end
```

PARAMETERS

DATA STRUCTURES

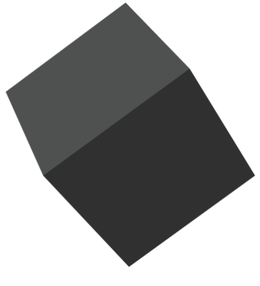
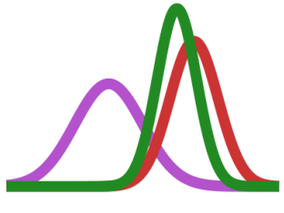
MUTATION

LOOPS / CONTROL FLOW

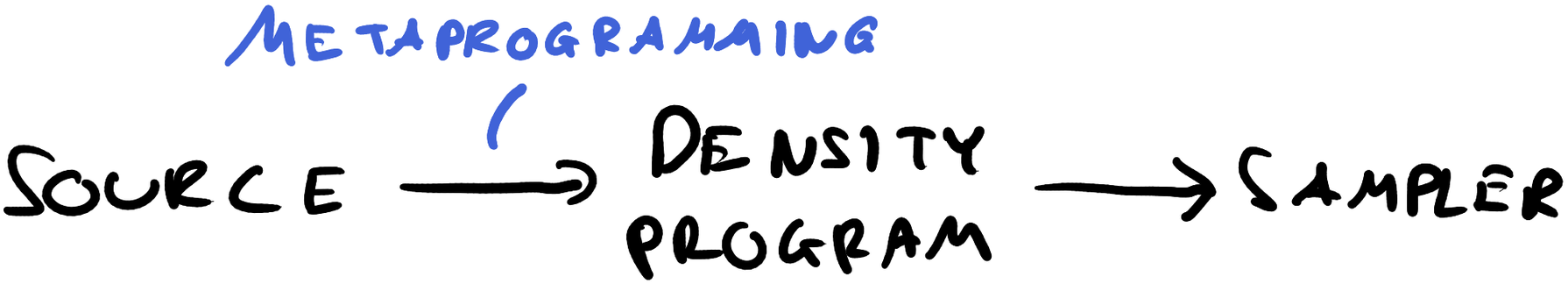
PPL CHARACTERISTICS

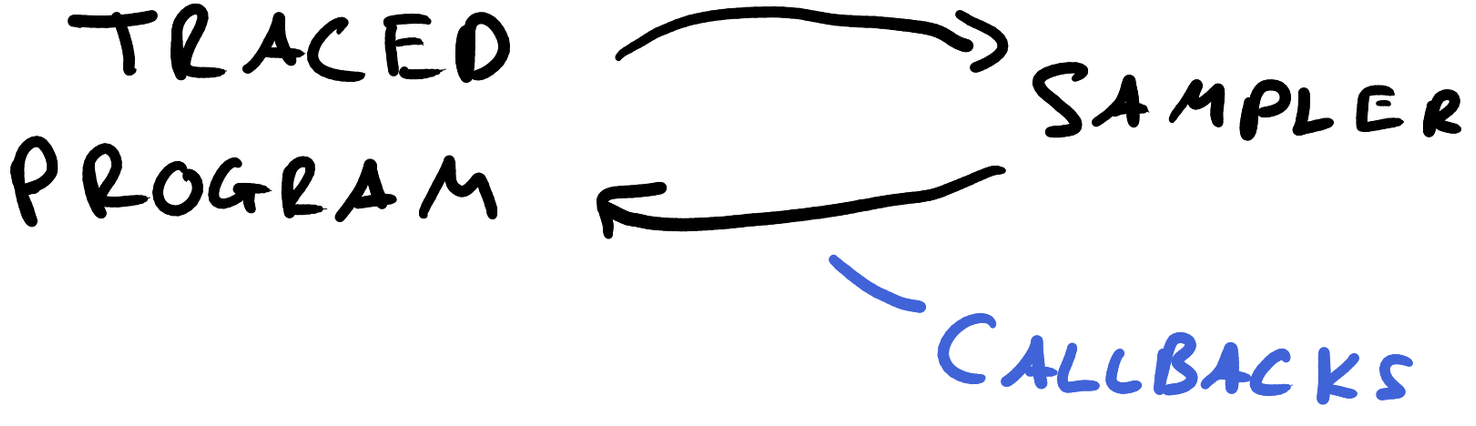
- DENSITY EVALUATION
- MODEL STRUCTURE
- AUTOMATIC DIFFERENTIATION
- SAMPLING, DATA GENERATION
- DIAGNOSTICS, EVALUATION
- PROGRAMMING & EXTERNAL LIBS
- COMPOSITION



PPL Implementations





Computation Graphs

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

EXPRESSION / PROGRAM

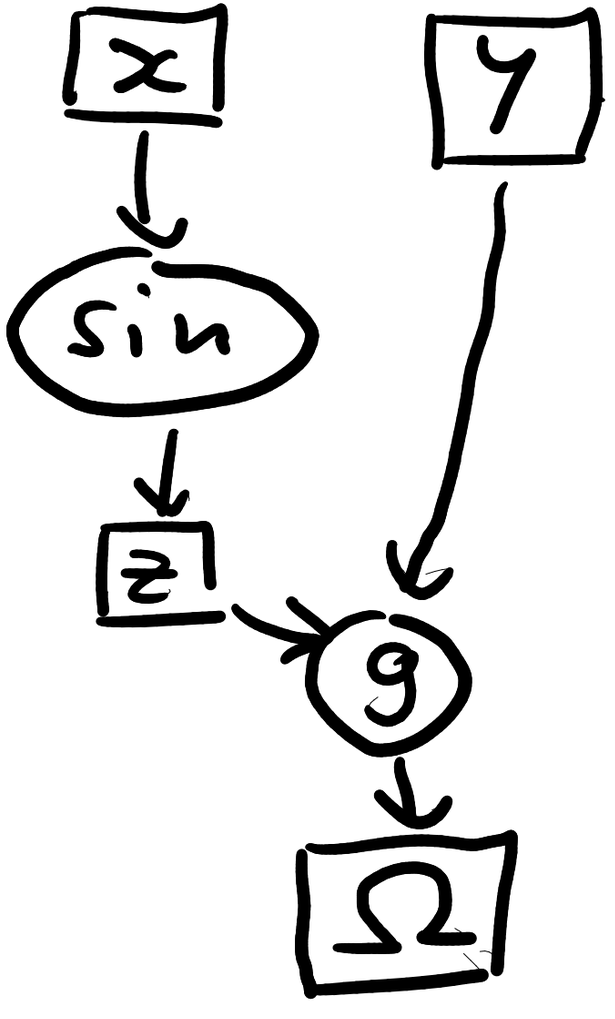
$x = ?$

$y = ?$

$z = \sin(x)$

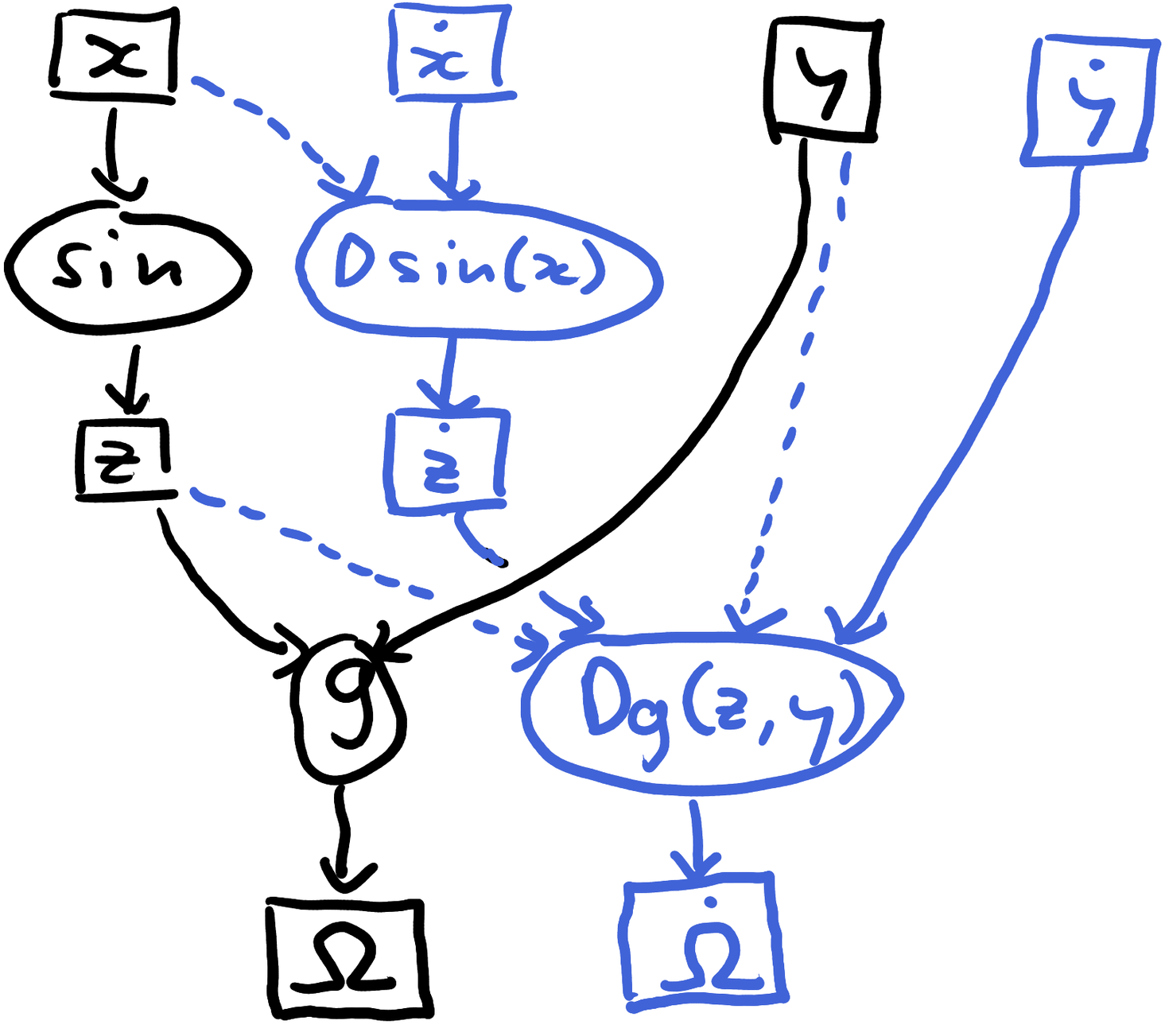
$\Omega = g(z, y)$

IR



GRAPH

Graphs in AD



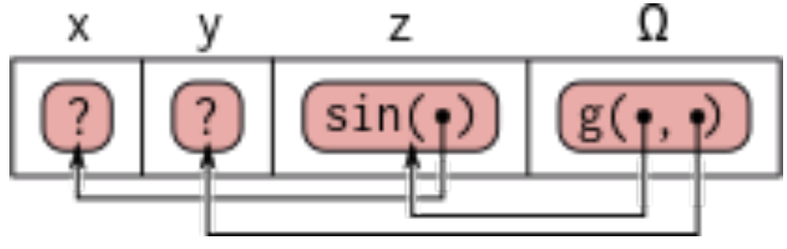
Implementaton Techniques

- Operator overloading:

x = TRACKED(?)

y = TRACKED(?)

⋮



WENGERT LIST, AKA TAPE

- Source transformation:

$x = ?$

$\dot{x} = \dot{x}_1$

$y = ?$

$\dot{y} = \dot{x}_2$

$z = \sin(x)$

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

$\Omega = g(z, y)$

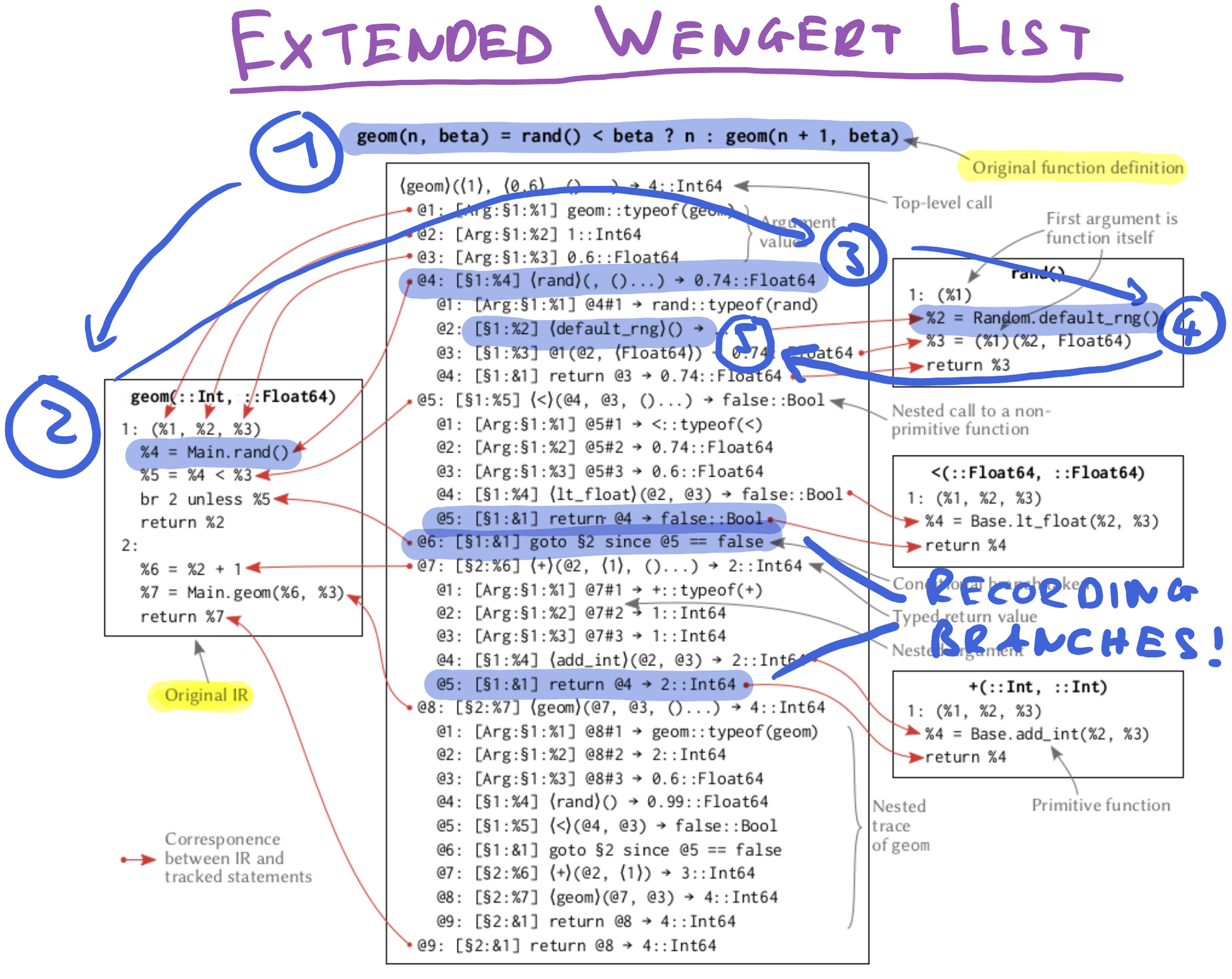
$\dot{\Omega} = Dg(z, y)(\dot{z}, \dot{y})$

JULIA IR

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

```
1: (%1, %2, %3)          (f, x, y)
   %4 = %3 > 0
   br 3 unless (%4)
2:
   %5 = sin(%2)          ← z = sin(x)
   %6 = g(%5, %3)        ← Ω = g(z, y)
   return %6
3:
   return %3
```

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

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

Special extra block for return values

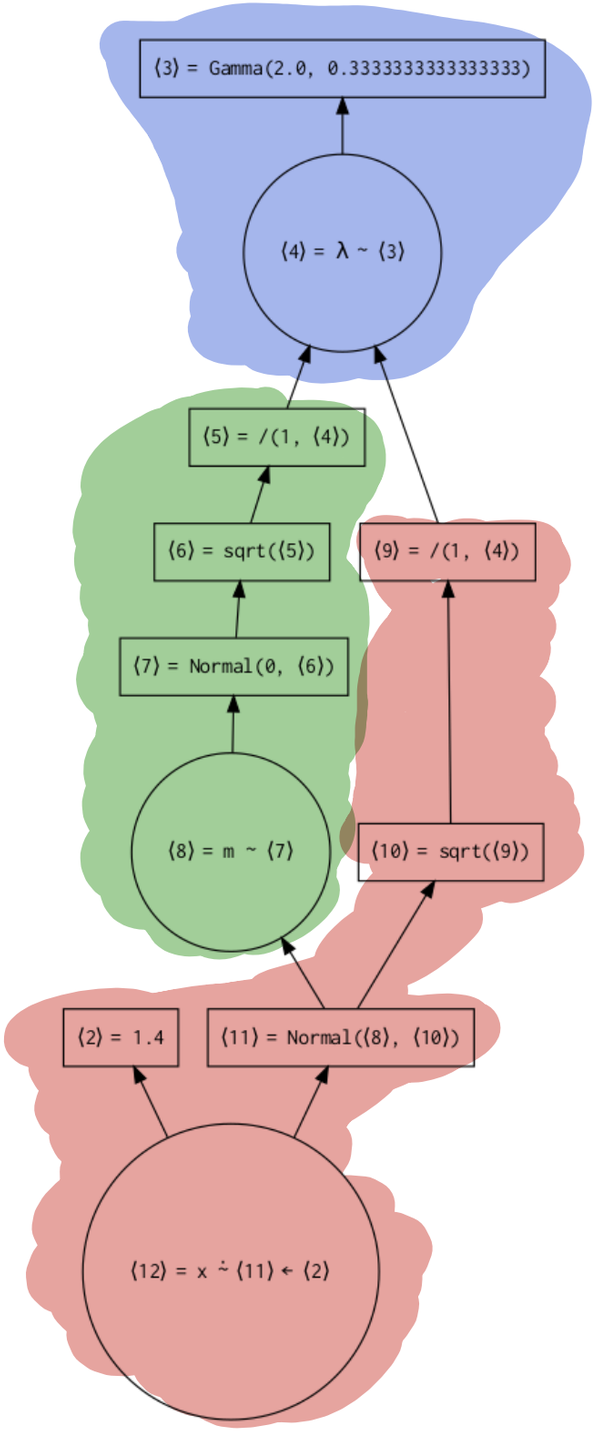
Original IR

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

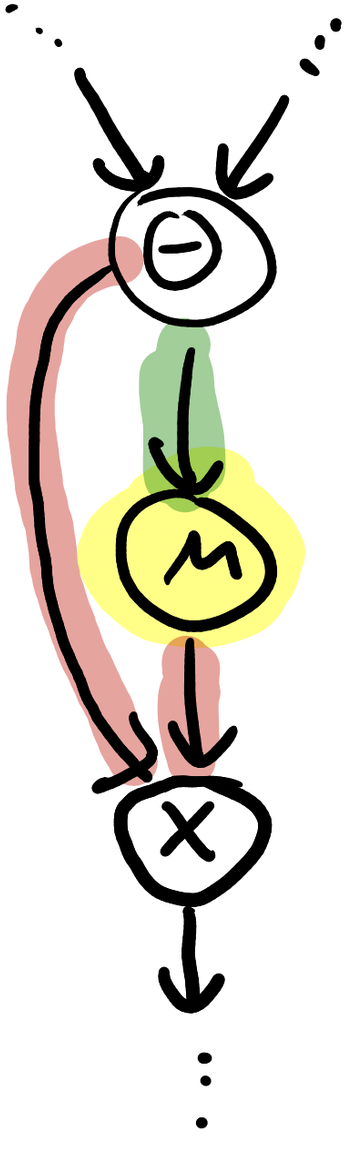
Jumps and returns are passed down to the next block

Dependency Extraction

```
@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
```



(DISCRETE) GIBBS CONDITIONALS



$$p(m, \vartheta, x) = p(\vartheta)\, p(m \mid \vartheta)\, p(x \mid m, \vartheta)$$

MARKOV BLANKET

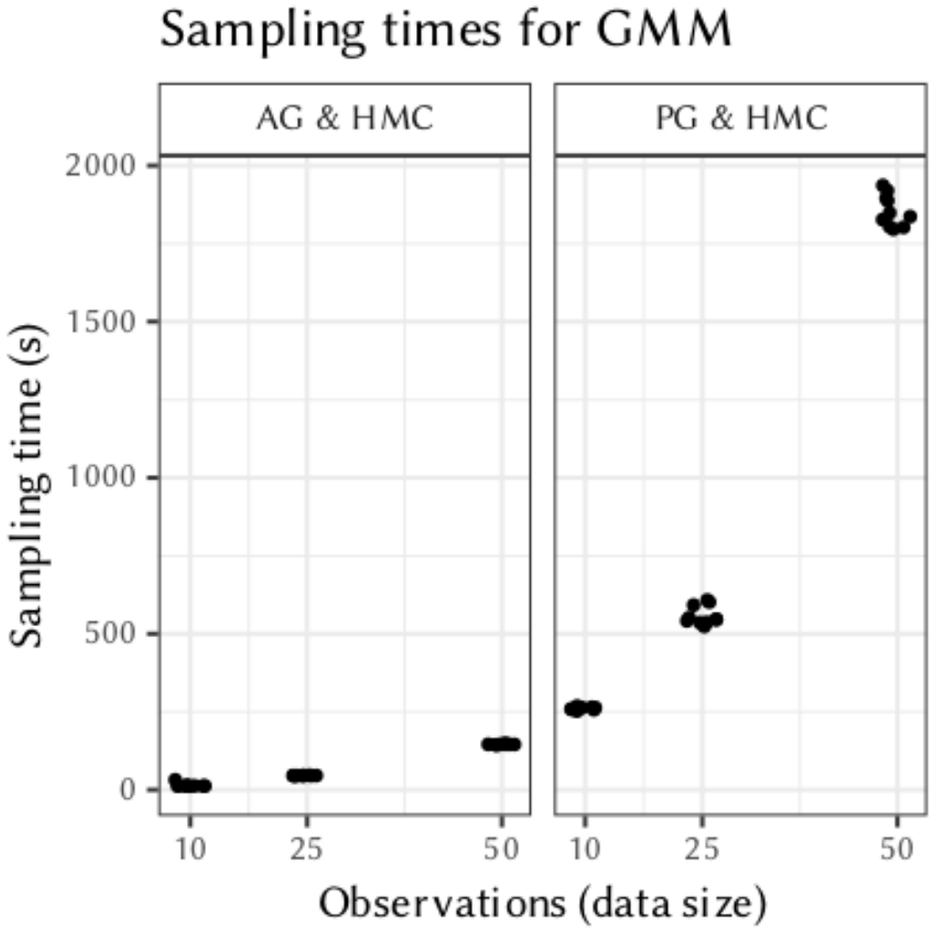
$$p(m \mid \vartheta, x) = \frac{p(m \mid \vartheta)\, p(x \mid m, \vartheta)}{Z}$$

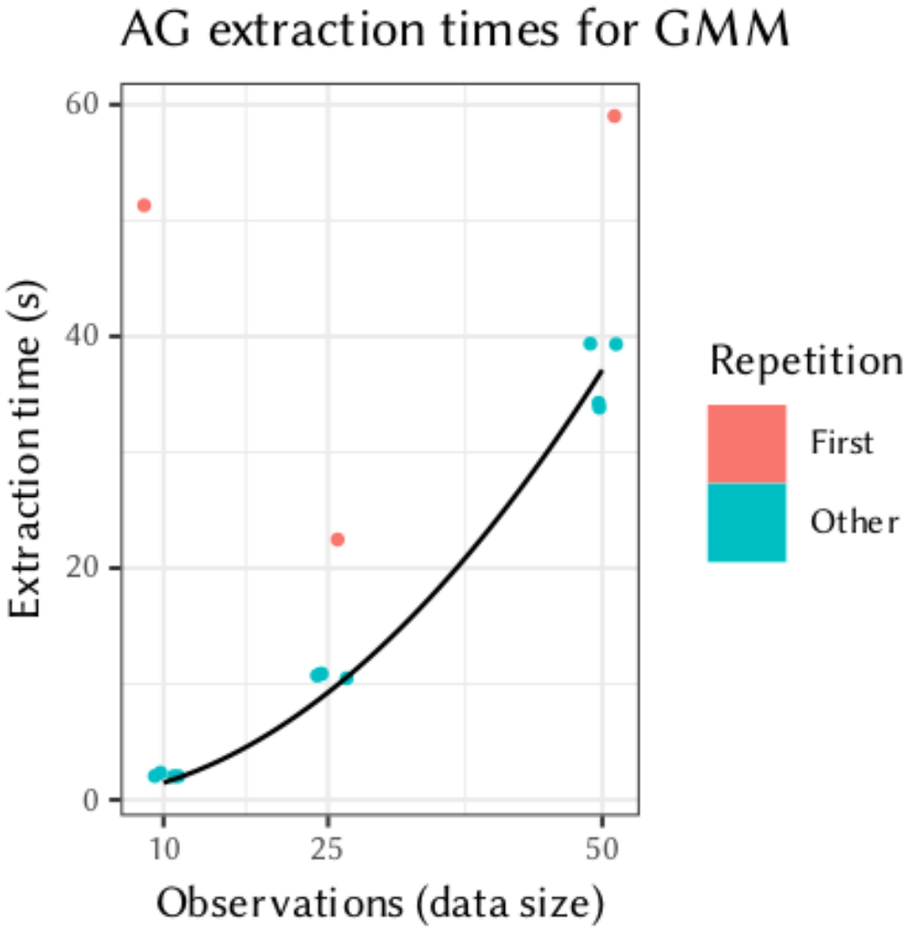
$$Z = \sum_m p(m \mid \vartheta)\, p(x \mid m, \vartheta)$$

A Test Model

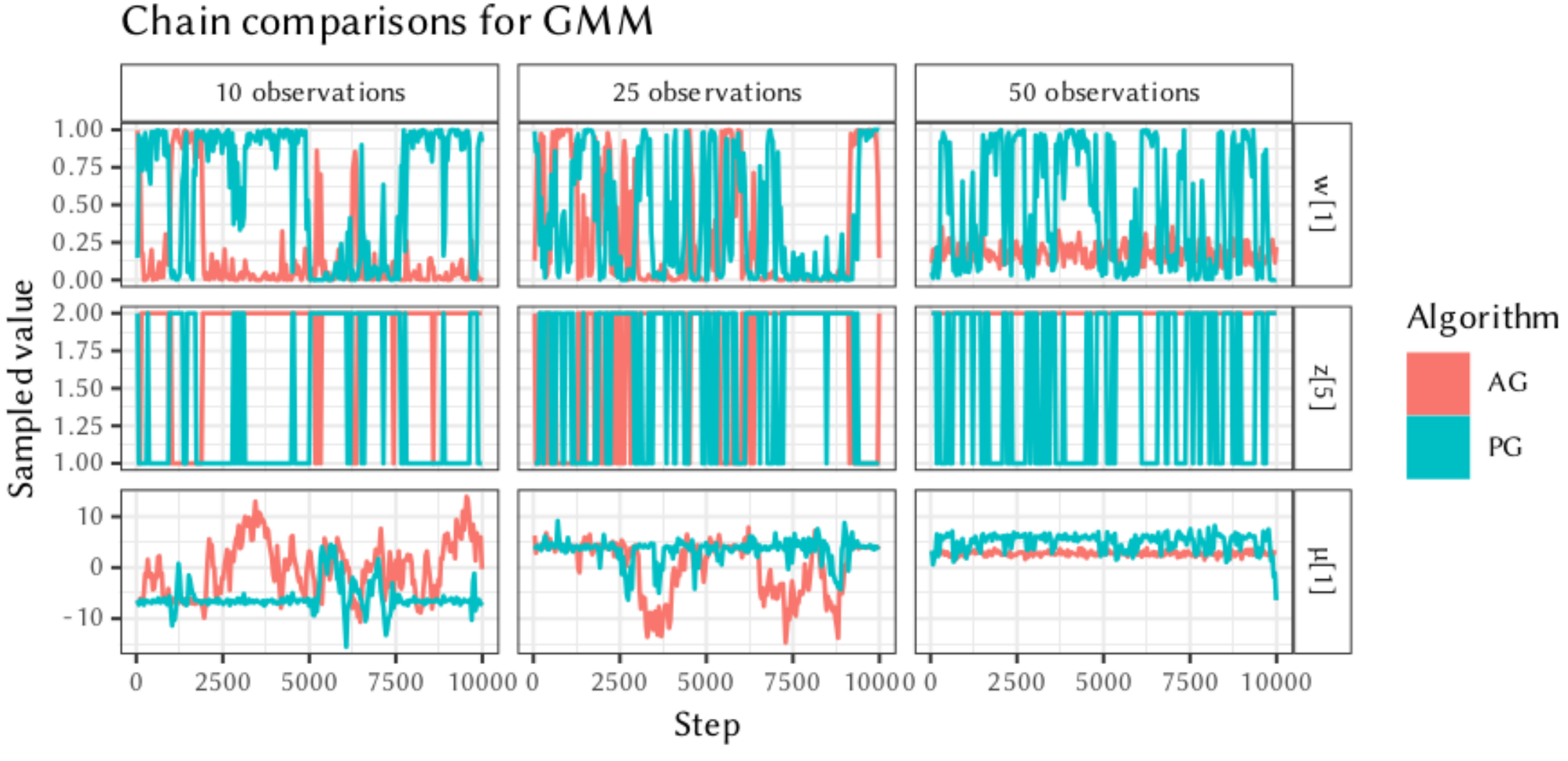
```
@model function gmm(x, K)
    N = length(x)
    w ~ Dirichlet(K, 1/K)  # Cluster association prior
    z ~ filldist(Categorical(w), N)  # Cluster assignments
    μ ~ filldist(Normal(0.0, s1_gmm), K)  # Cluster centers

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

Evaluation I

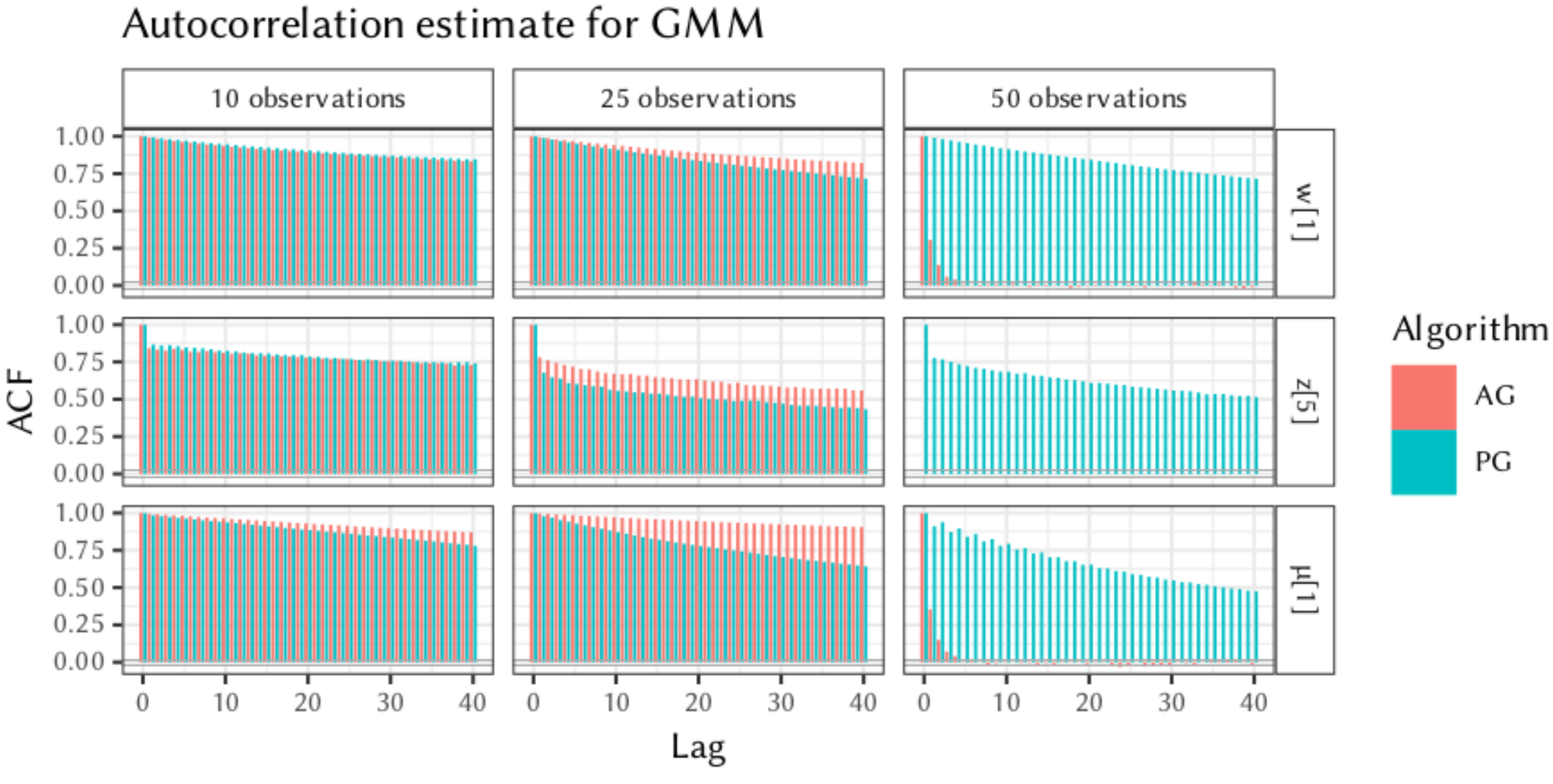




Evaluation II



Evaluation III



Conclusions

- Static Dependencies, finite conditionals: 
- Slicing Dynamic Models: 
- Recovering Dynamic Structure: 
- Future Proof: 

scratch

PREDICTION

- PLUGIN ESTIMATOR:

arg max!

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

- POSTERIOR PREDICTIVE:

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

POSTERIOR EXPECTATION!