

θ

Actionable Bayesian Workflow

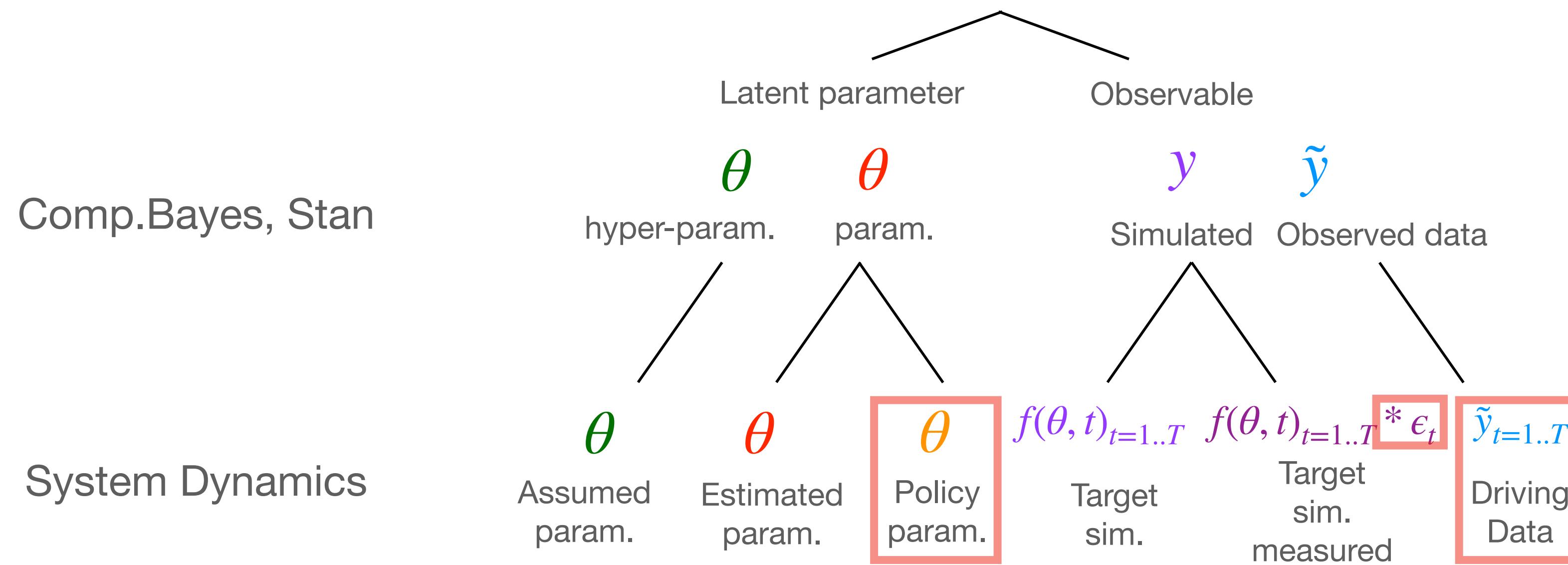
Orchestrated with
Simulation-based Calibration

Angie.H.Moon (MIT Sloan)
09.2022 MIT Probabilistic Computing Team Lab Seminar
amoon@mit.edu

As a Stan developer (Computational Bayes)
striving to learn
System dynamics

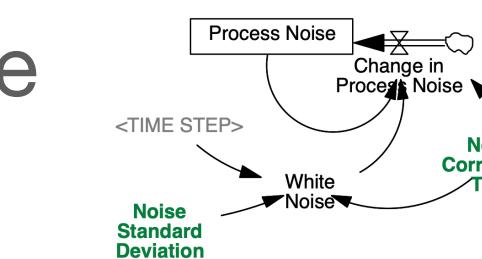
**different language is the most
difficult part**

Mapping Variables of Computational Bayes and Dynamics



: variable types in system dynamics but haven't seen in Stan

$* \epsilon_t$: multiplicative process noise

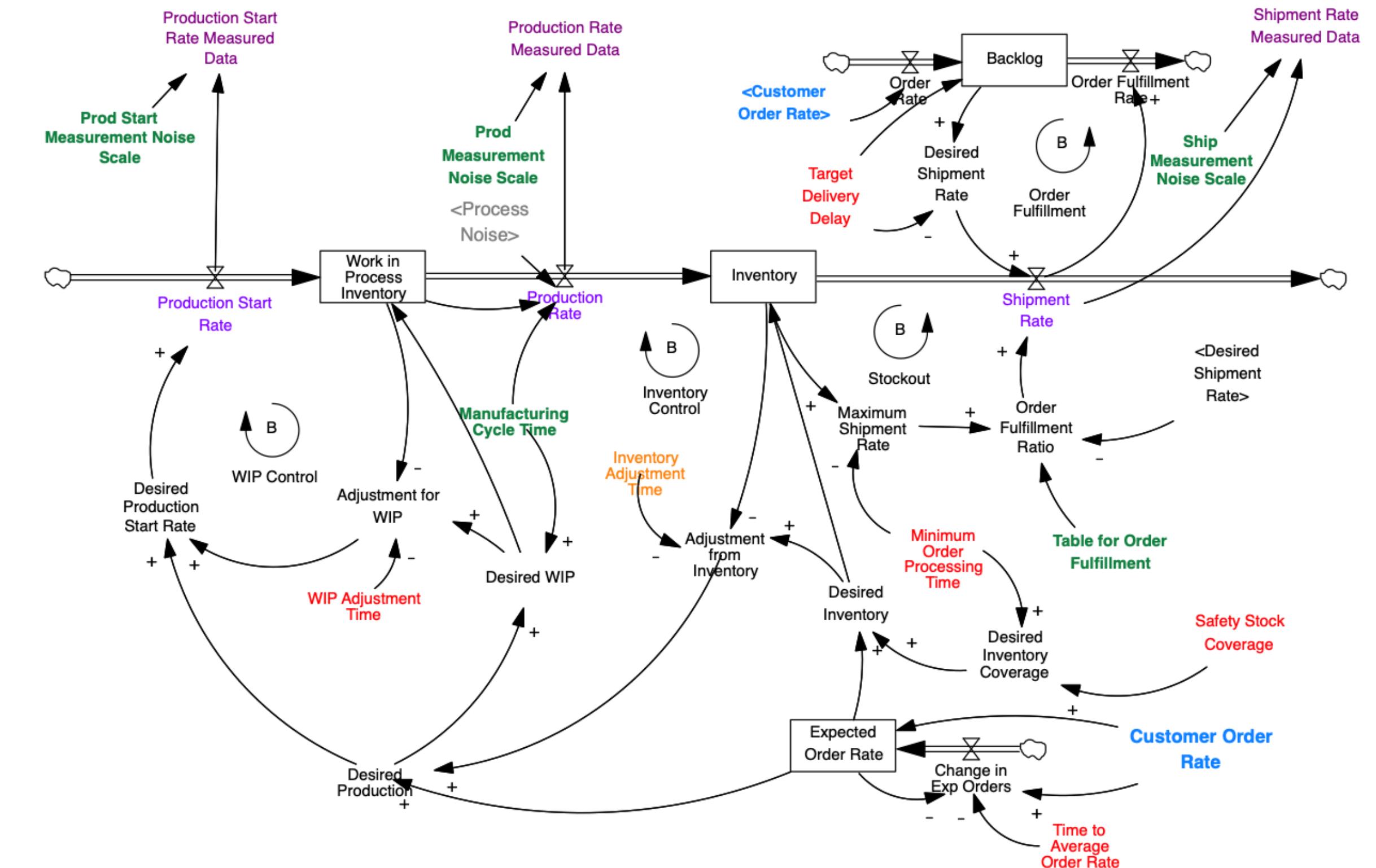
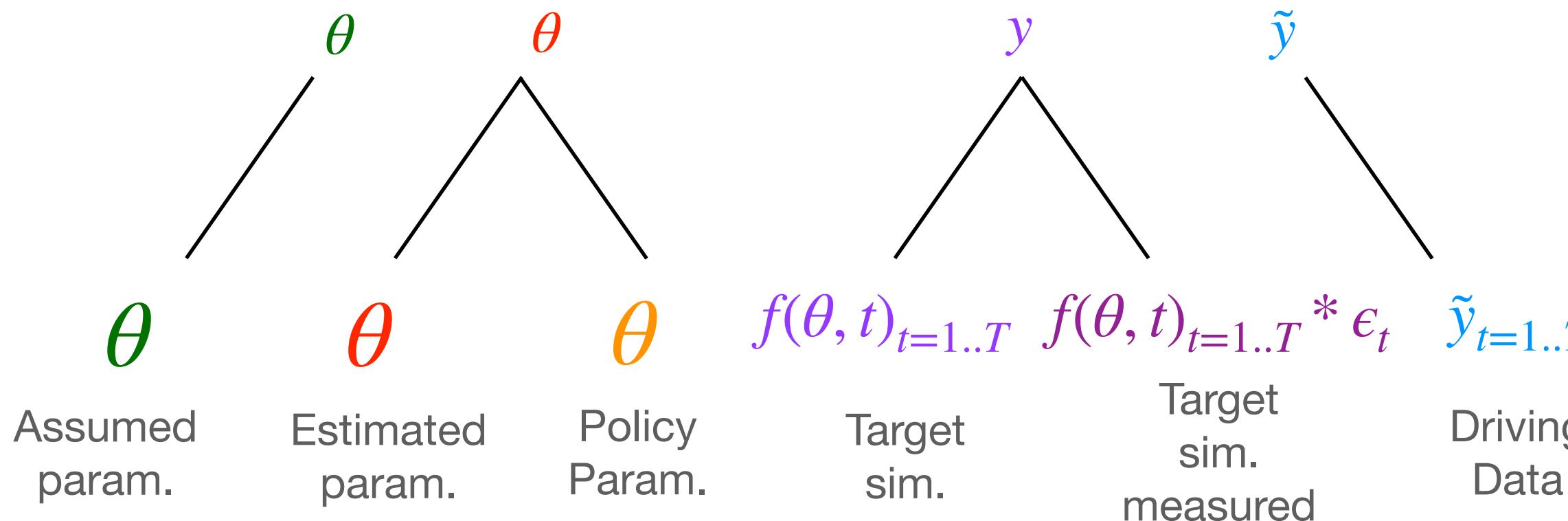


Variable Semantics in Dynamic Model

Inventory management example

Inventory manager asks:

What is the optimal **inventory adjustment time**
to maximize my profit, as a function of states?
i.e. profit = f (Work in Process Inventory, Inventory, Backlog)



Variable Types in Dynamic Model

- **Scalar**
- **Vector**
- **Differential equations**

Scalar and Vector variable

Inventory management example

Computer estimates θ conditional on fixed $\theta_{assumed}$ and \tilde{y}

θ

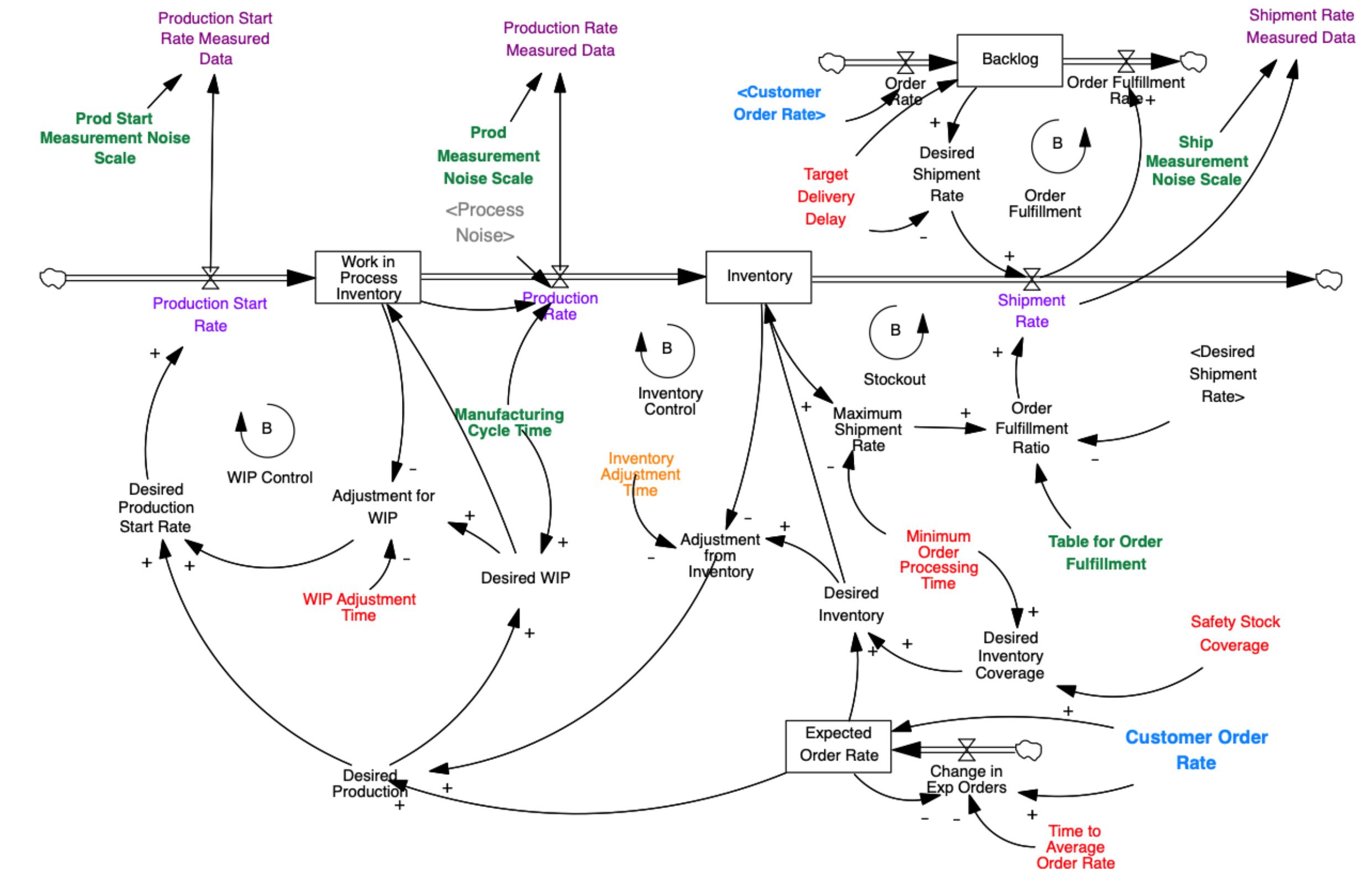
- Time to average order rate (1~3 days)
- Target delivery delay (2~4 days)

$\theta_{assumed}$

- Manufacturing cycle time (8 days)
- Measurement Noise Scale (0.1)

\tilde{y}

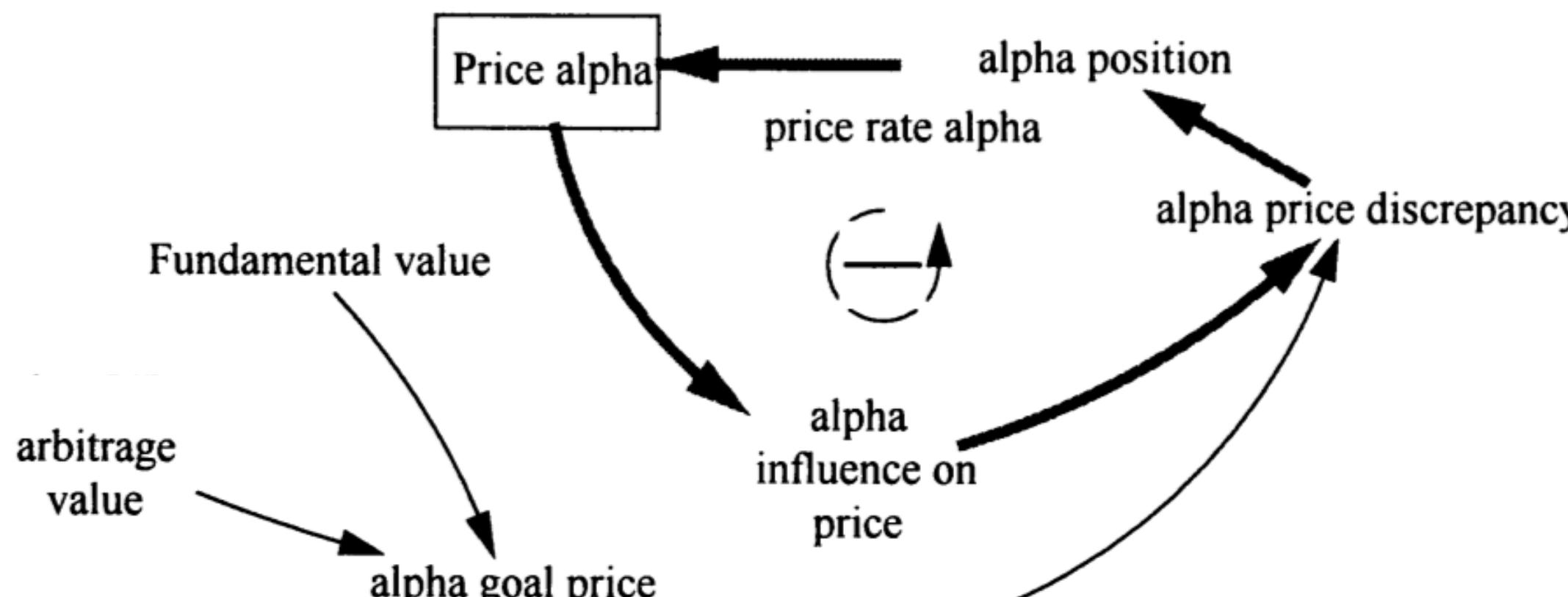
- Customer order rate (observed time series, 5,4,7,...)



Differential equations variable

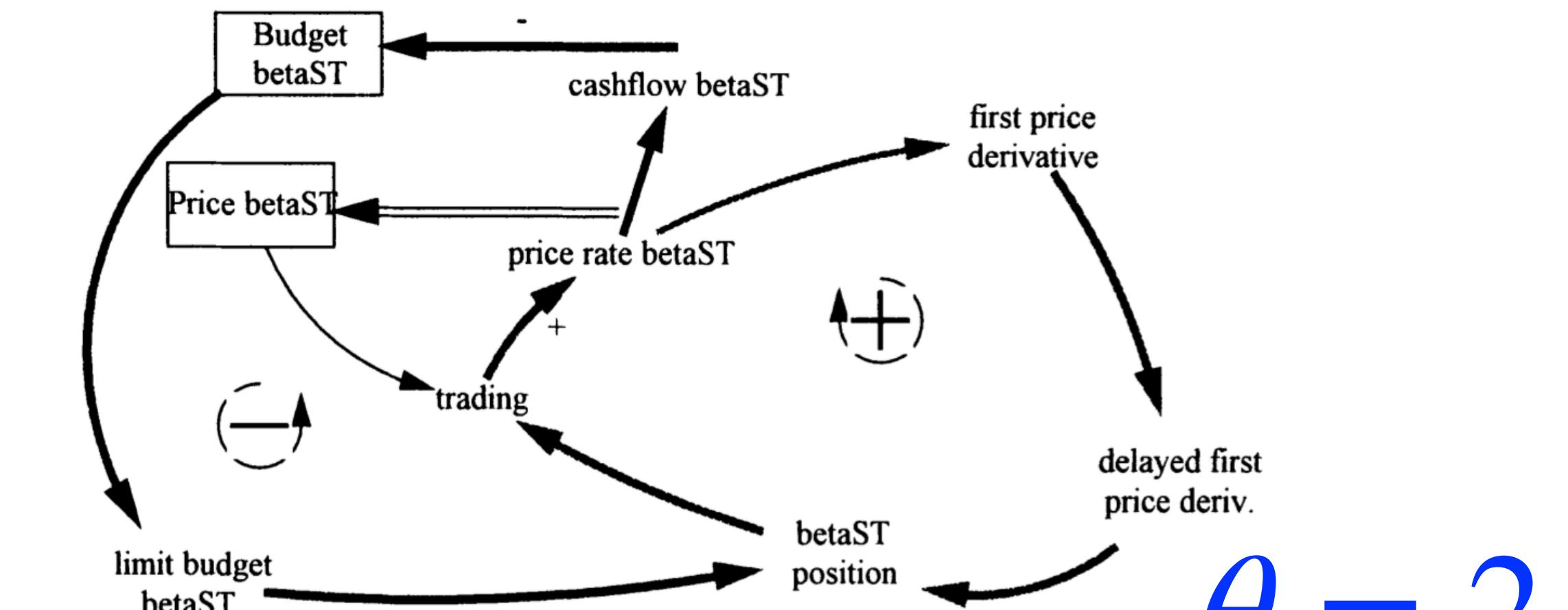
Behavioral finance example

Kunsch et al classified investor behaviors to explain market price as the mixture of three behavior families



$$\theta = 1$$

Rational goal seeking behavior
“Stabilize price”



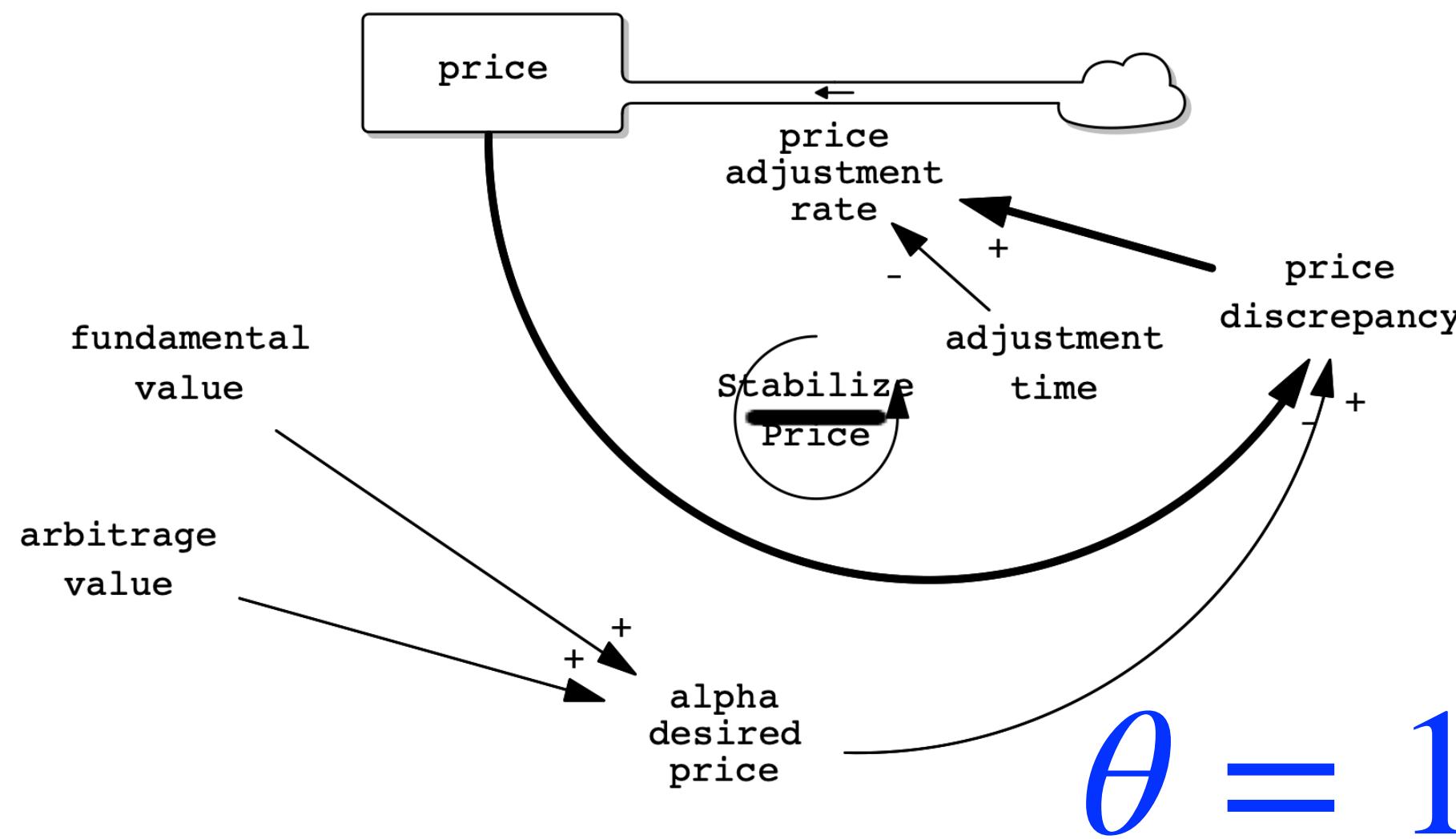
$$\theta = 2$$

Heuristic
“In phase with price trend creating noise”

$\theta = 3..$ Category expands but can assumed to be fixed for short time horizon

Differential equations variable

Introducing Vensim to Stan auto translator



Vensim (system dynamic software)

translated to

```
vector vensim_ode_func(real time, vector outcome){  
    vector[1] dydt; // Return vector of the ODE function  
  
    // State variables  
    real price = outcome[1];  
  
    real adjustment_time = 3;  
    real fundamental_value = 1;  
    real arbitrage_value = 1;  
    real alpha_desired_price = fundamental_value + arbitrage_value;  
    real price_discrepancy = alpha_desired_price - price;  
    real price_adjustment_rate = price_discrepancy / adjustment_time;  
    real price_dydt = price_adjustment_rate;  
  
    dydt[1] = price_dydt;  
  
    return dydt;  
}
```

Stan (Computational Bayes open-source)

Weights of behavioral categories can be estimated with mixture models

Actionable Workflow with Sequential Bayesian Update

- a. Check $\Theta_t, \theta\theta$ passes simulation-based calibration
- b. Update Θ_t to Θ_{t+1} with real data
- c. Prescribe policy based on $E[f(\Theta_{t+1}, \theta, \tilde{y}) | \theta, \theta, \theta]_{\theta=1..P}$

-> Example next pg.

a: generate then estimate, c: generate only

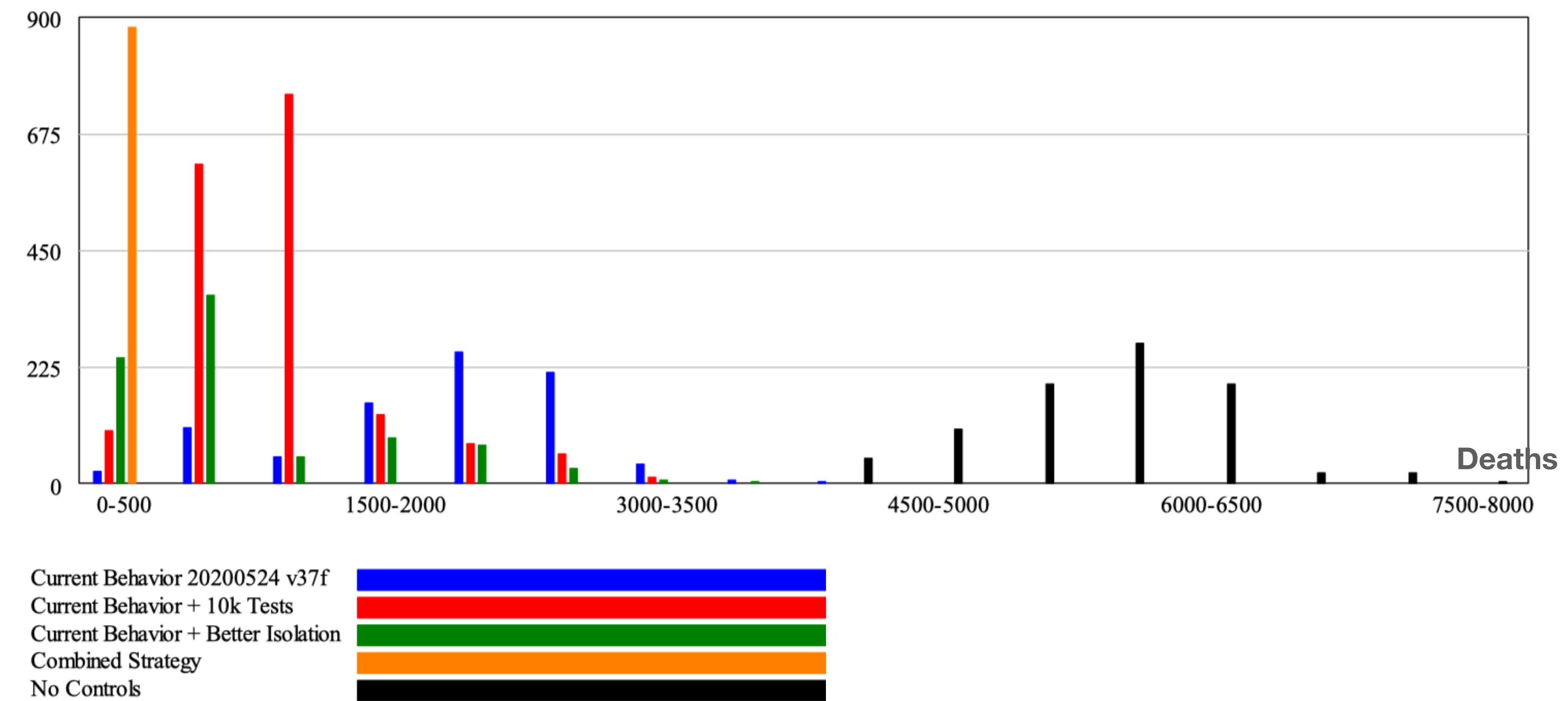
Final Result of Actionable Bayesian Workflow

Example from Epidemiology

Public health manager asks:

Which **policy** performs the best under combined uncertainties?

Deaths distribution conditional on each policies



What is Simulation-based Calibration?

Skip pg. 11-34 if you know SBC

SBC is characterized as

Rehearsal to orchestrate (P, A, D)

P: joint distribution $p(\theta, y)$

A: approximator $P_A(\theta | y)$

D: observed data \tilde{y}

Conductor Orchestrates

:=



arrange
elements of situation
to surreptitiously produce
desired effect

Five Cores of Orchestrate

Rehearsal

Elements

Dynamics

Data-based

Desired effect

arrange
elements of situation
to surreptitiously produce
desired effect

Five Cores of Simulation-based Calibration

Rehearsal

Elements

Dynamics

Data-based

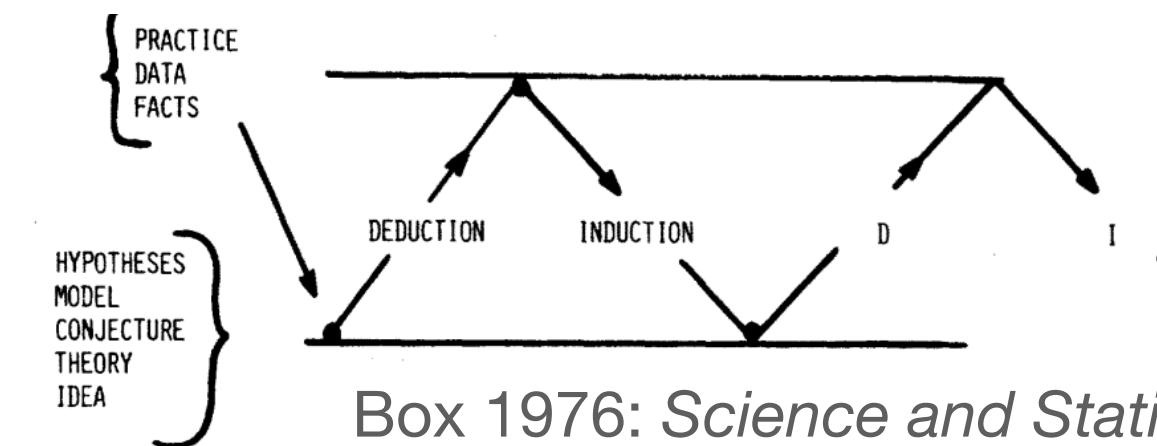
Desired effect

1. Orchestrate in Rehearsal

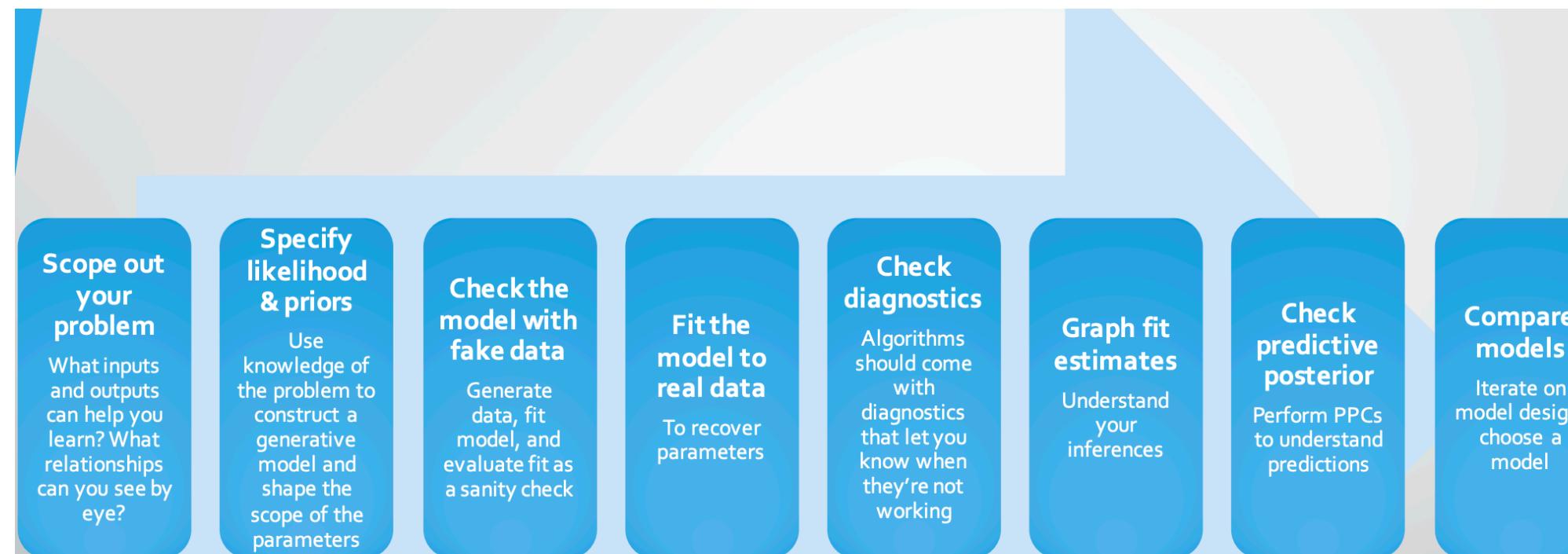


arrange
elements of situation
to **surreptitiously** produce
desired effect

Bayesian Workflow Includes Rehearsal

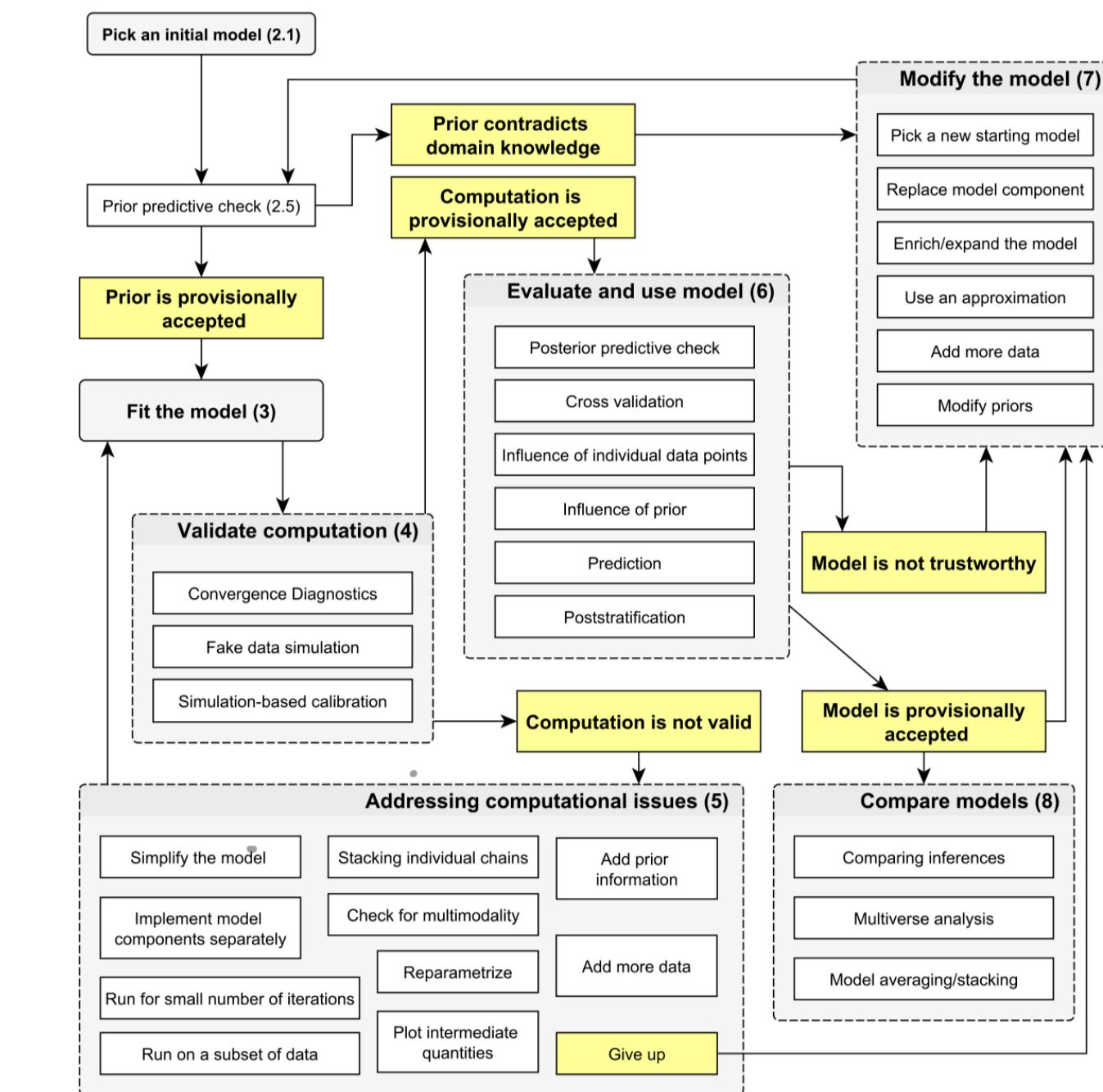


Box 1976: *Science and Statistics*
“theory and practice iteration with **feedback loop**”



Talts 2018: *StanCon Intro*

Evolved to...

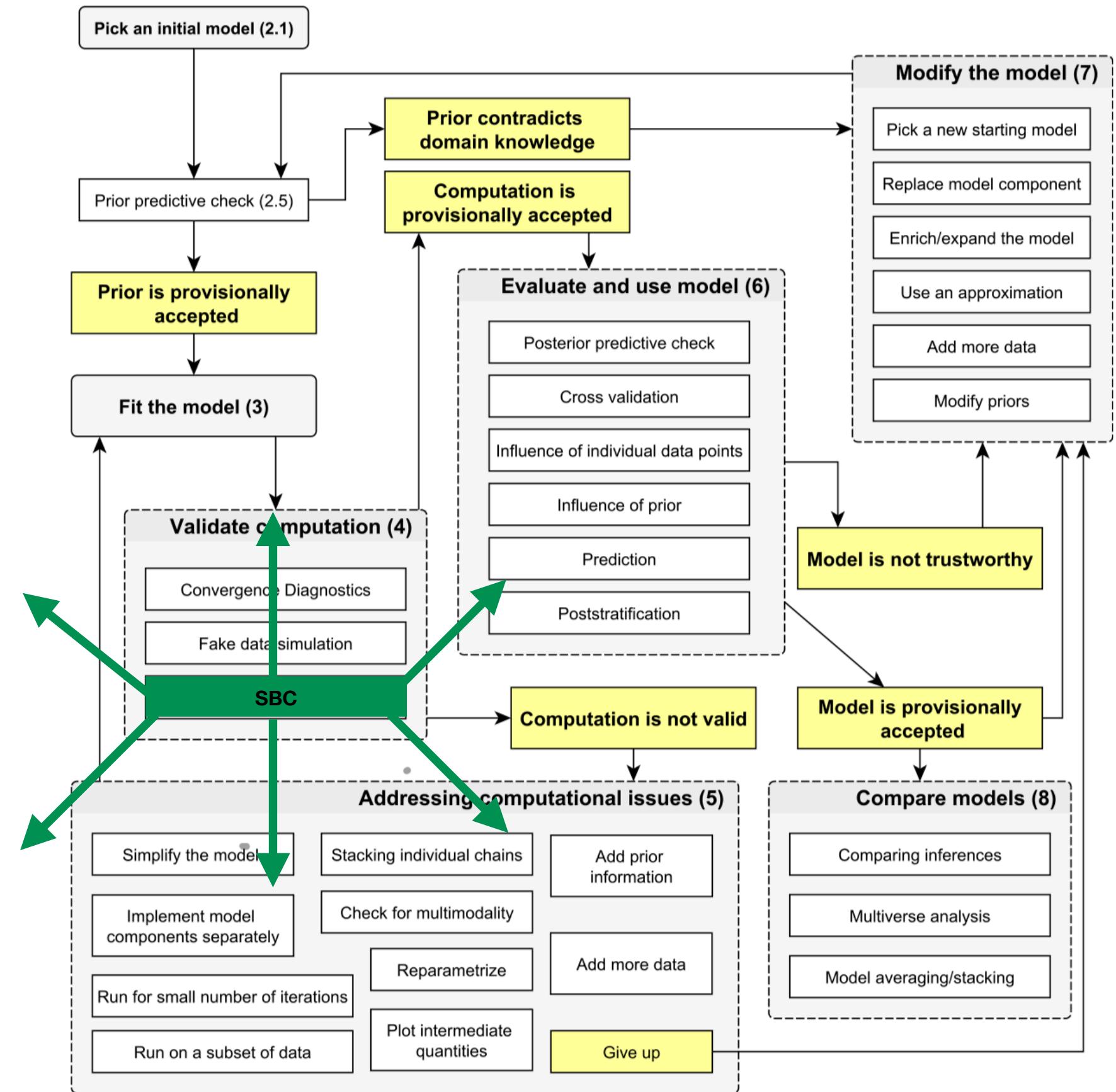


Gelman et al 2020: *Bayesian workflow*

Simulation-based Calibration is Rehearsal

Not limited to computation validation!

Bayesian time machine allowing Cause-Effect switching **experiment**

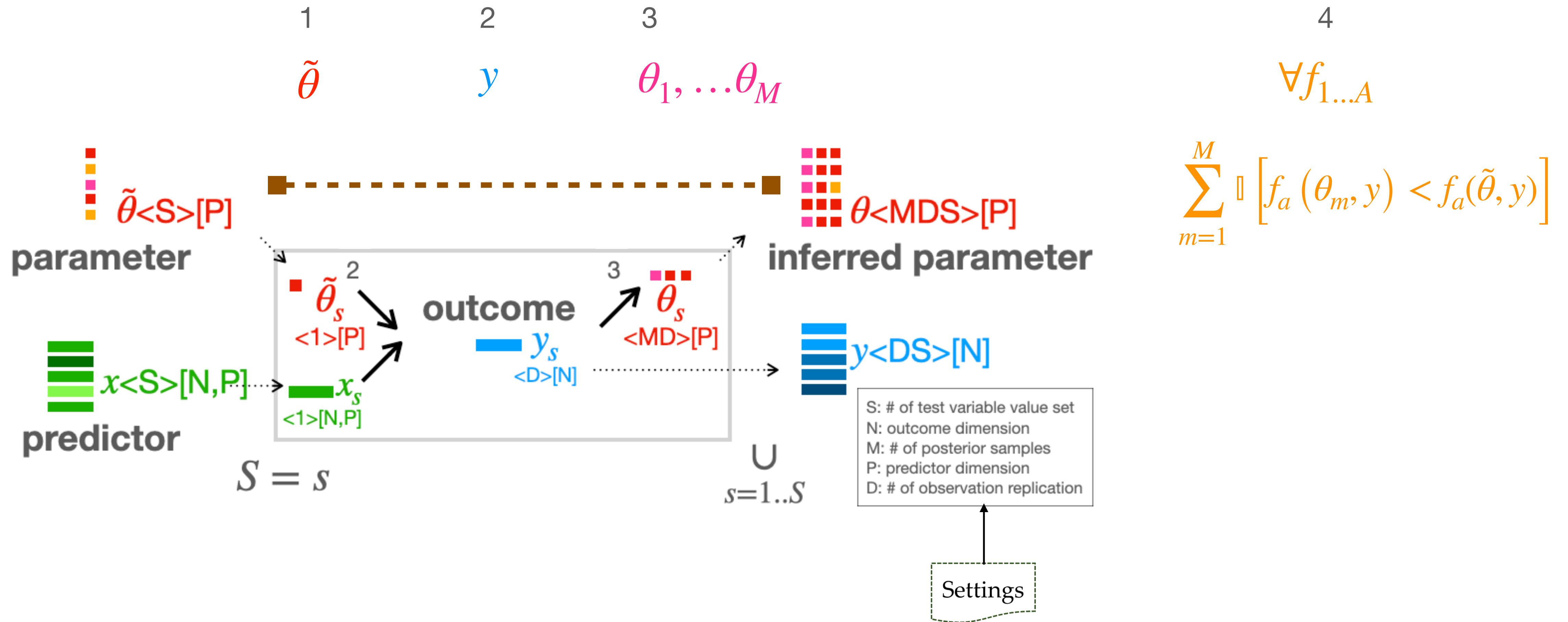


$$\pi_{\text{obs}}(y|\tilde{\theta})\pi_{\text{prior}}(\tilde{\theta}) = \pi_{\text{marg}}(y)\pi_{\text{post}}(\tilde{\theta}|y)$$

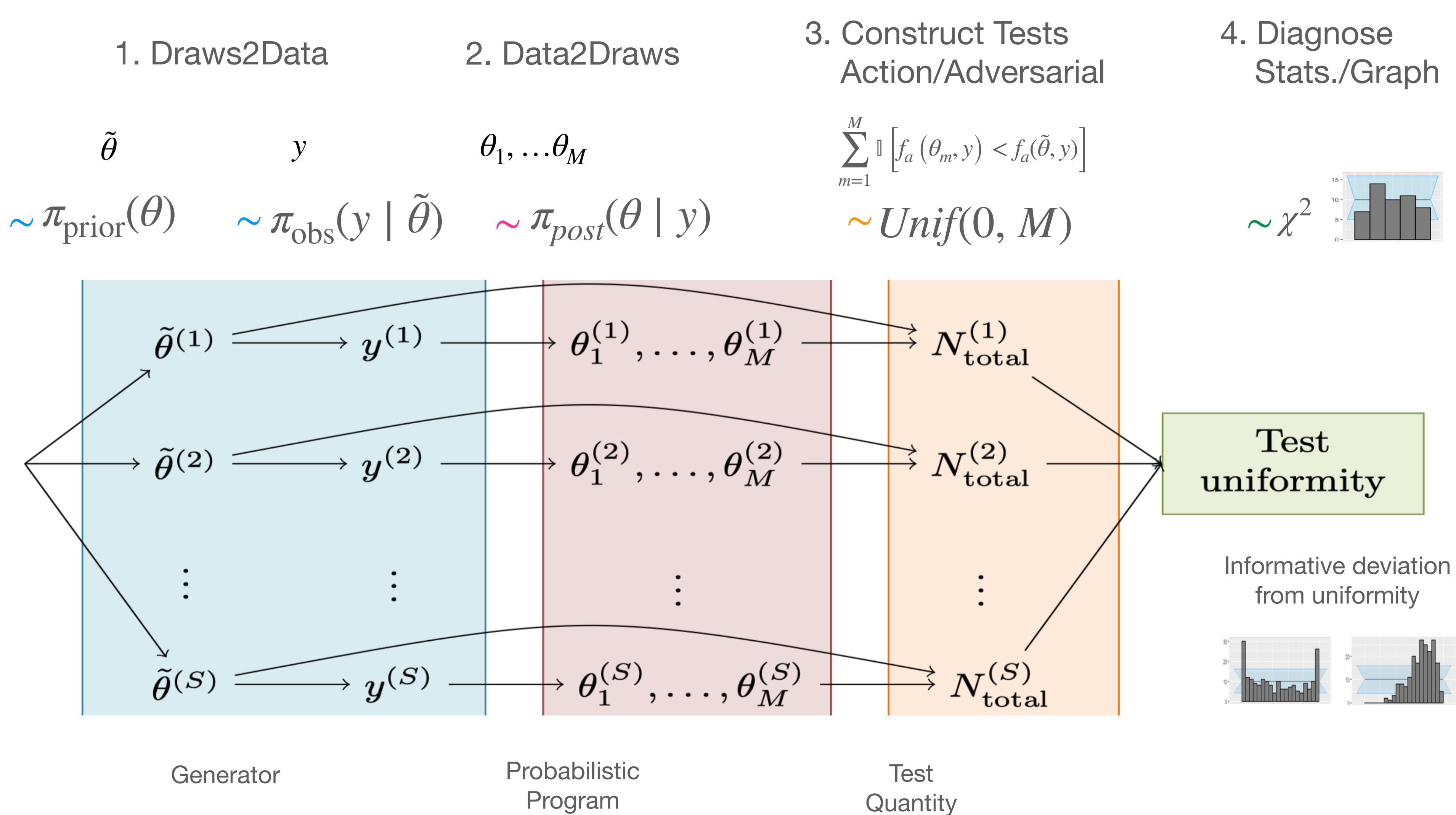
$$\pi_{\text{SBC}}(y, \theta, \tilde{\theta}) = \pi_{\text{prior}}(\tilde{\theta}) \pi_{\text{obs}}(y|\tilde{\theta}) \pi_{\text{post}}(\theta|y)$$

$$\pi_{\text{prior}}(\theta) = \int_Y dy \int_{\Theta} d\tilde{\theta} \pi_{\text{post}}(\theta|y) \pi_{\text{obs}}(y|\tilde{\theta}) \pi_{\text{prior}}(\tilde{\theta})$$

Simulation-based Calibration (SBC)



Procedural Elements of SBC



2. Elements to Orchestrate?

A musical score consisting of three staves. The top staff is for the Violin, the middle for the Viola, and the bottom for the Saxophone. The score is in common time (indicated by '4'). The Violin has a melody of eighth and sixteenth notes. The Viola provides harmonic support with sustained notes and eighth-note chords. The Saxophone plays a rhythmic pattern of eighth and sixteenth notes. The score is divided into measures by vertical bar lines.

arrange
elements of situation
to surreptitiously produce
desired effect

Elements of Computational Bayes

Complicated, not consistent

Unified taxonomy

Statistical model

Generator

$$e \sim \pi$$
$$+ - \times /$$

Unobserved parameter

Assumed Estimated

$$\theta \quad \theta \theta$$

Computational model

Probabilistic program

0 1



Data



Simulated

y

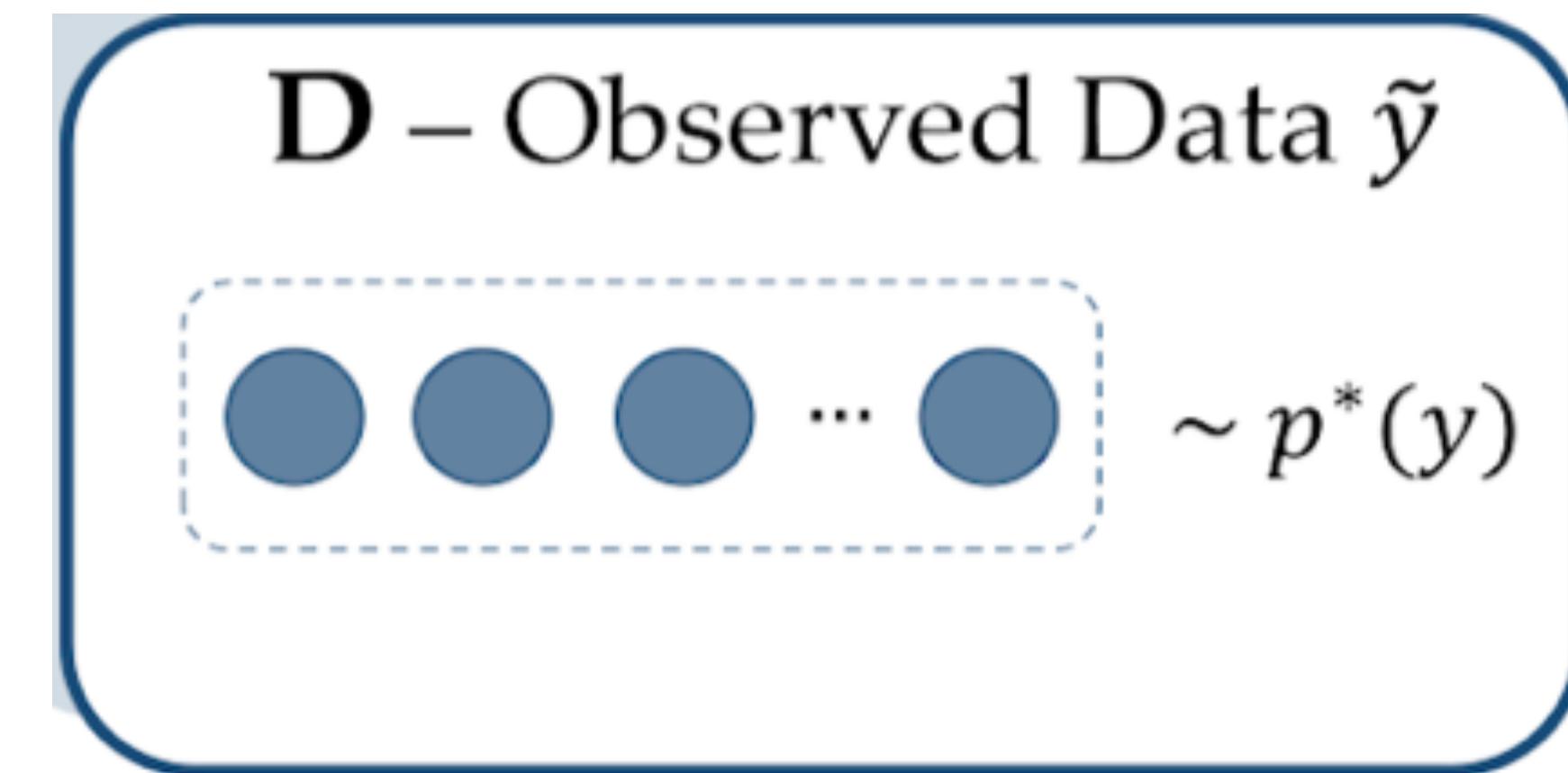
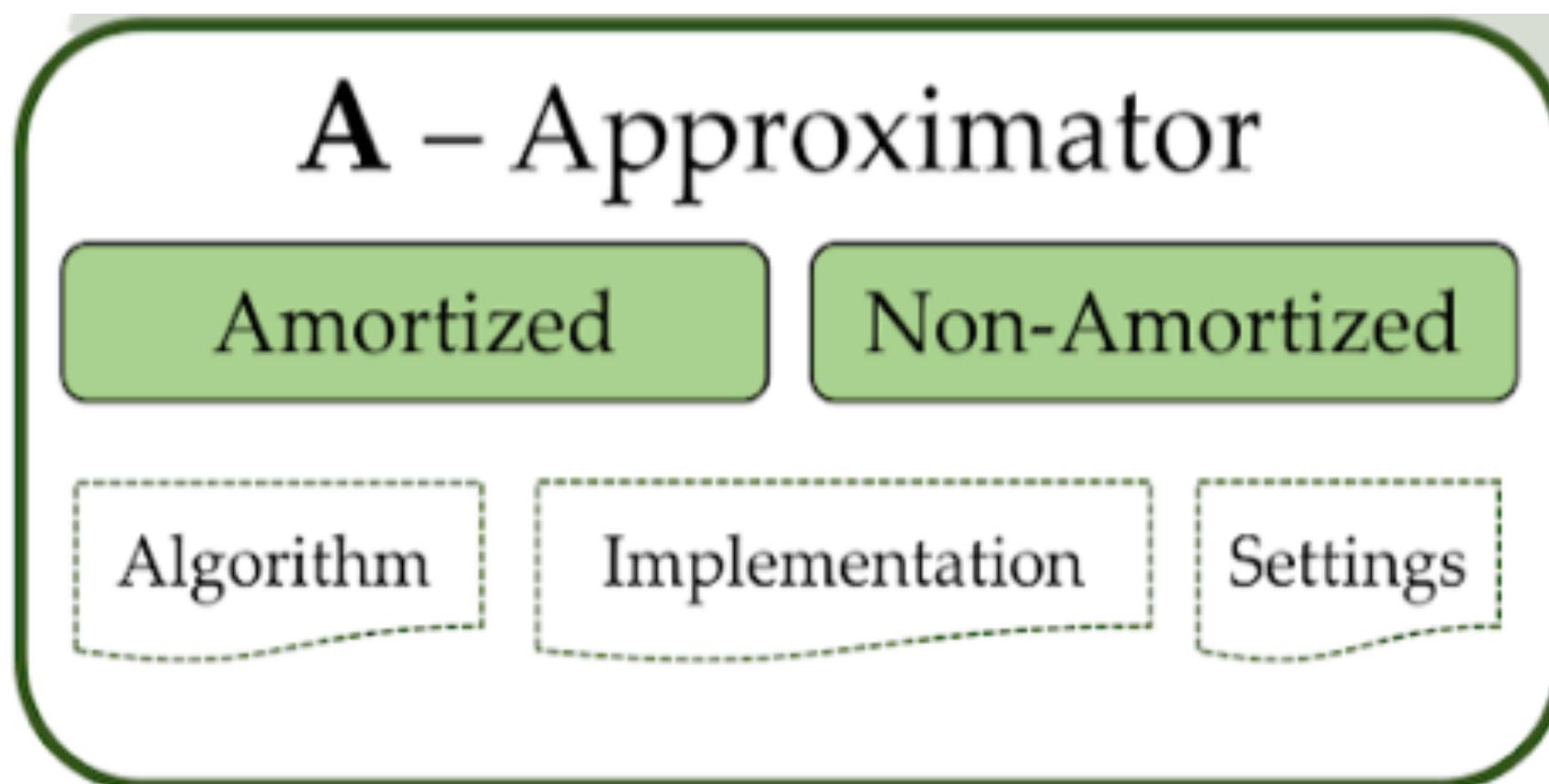
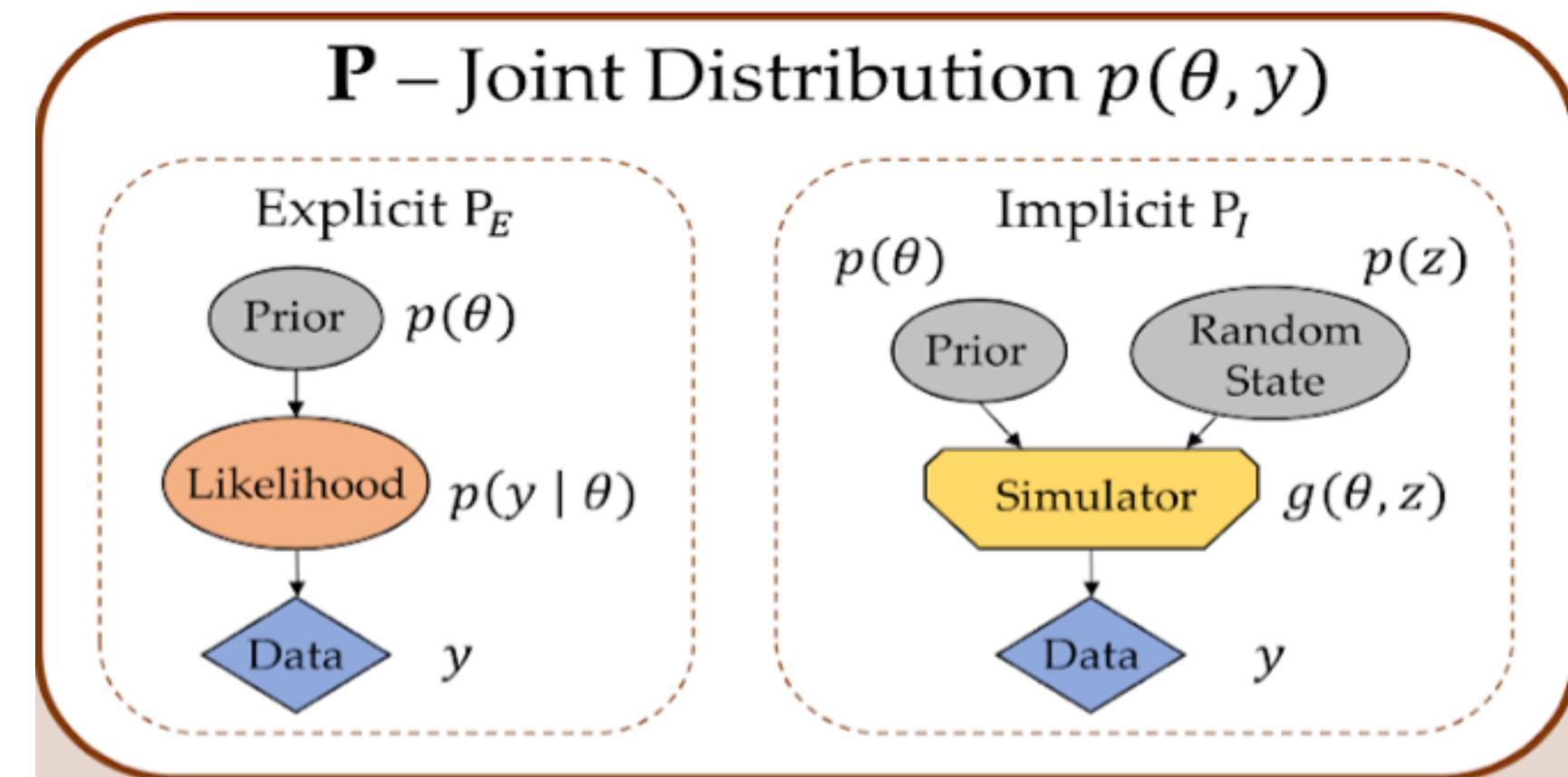
Evolving to..

P: joint distribution $p(\theta, y)$

A: approximator $P_A(\theta | y)$

D: observed data \tilde{y}

Elements of Computational Bayes



SBC Orchestrates Computational Bayes Elements

a. Identify P, A, D



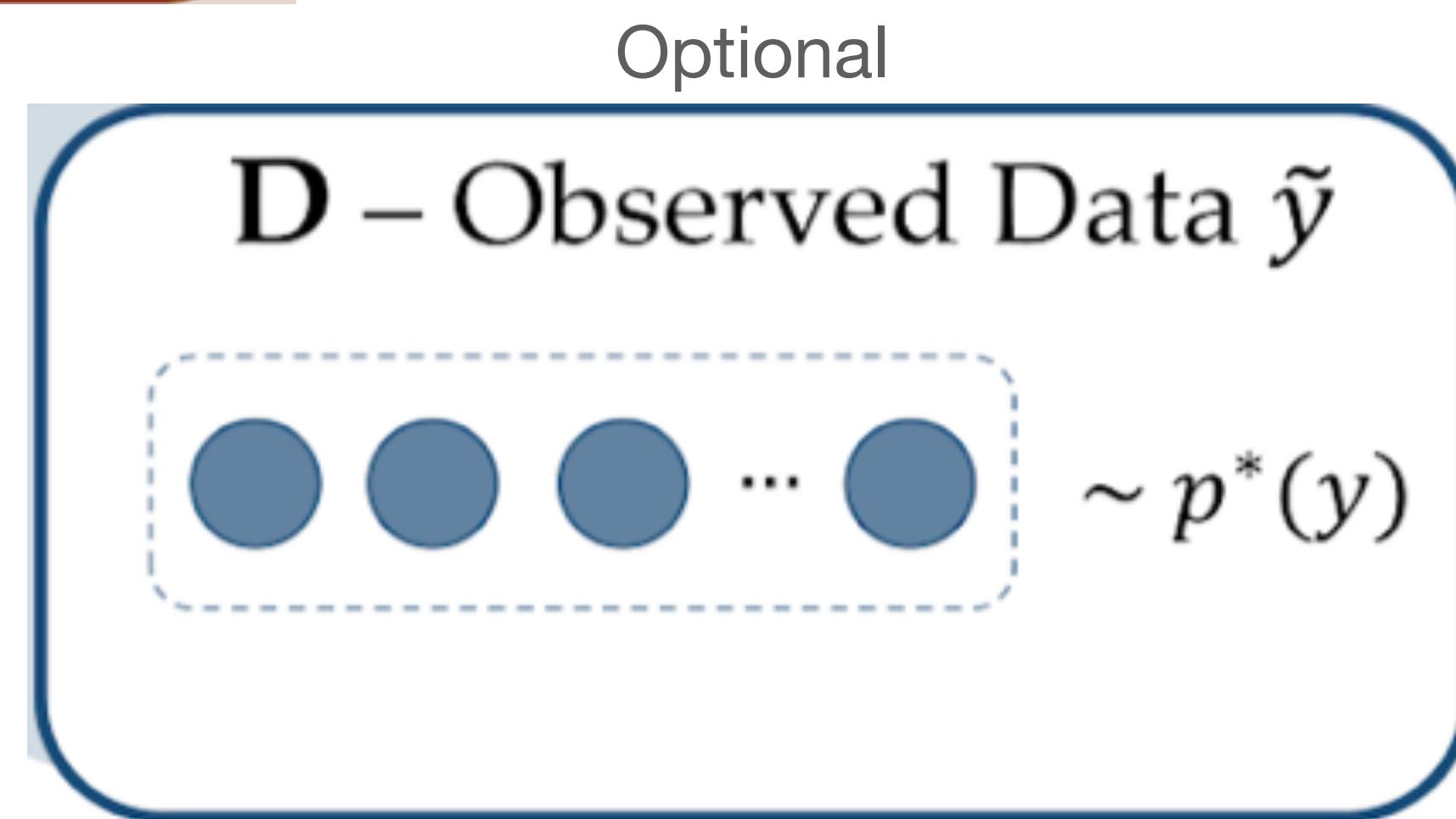
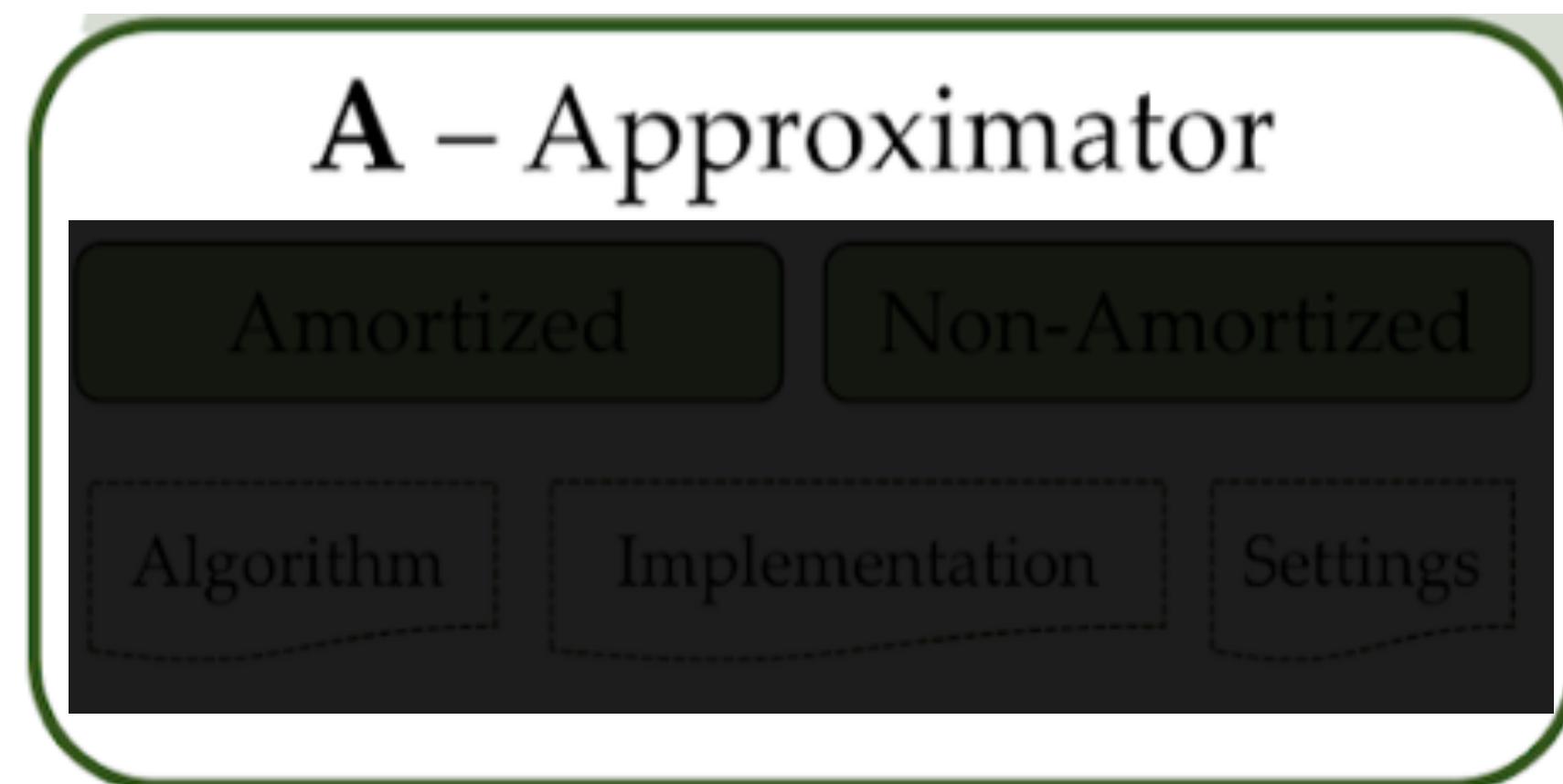
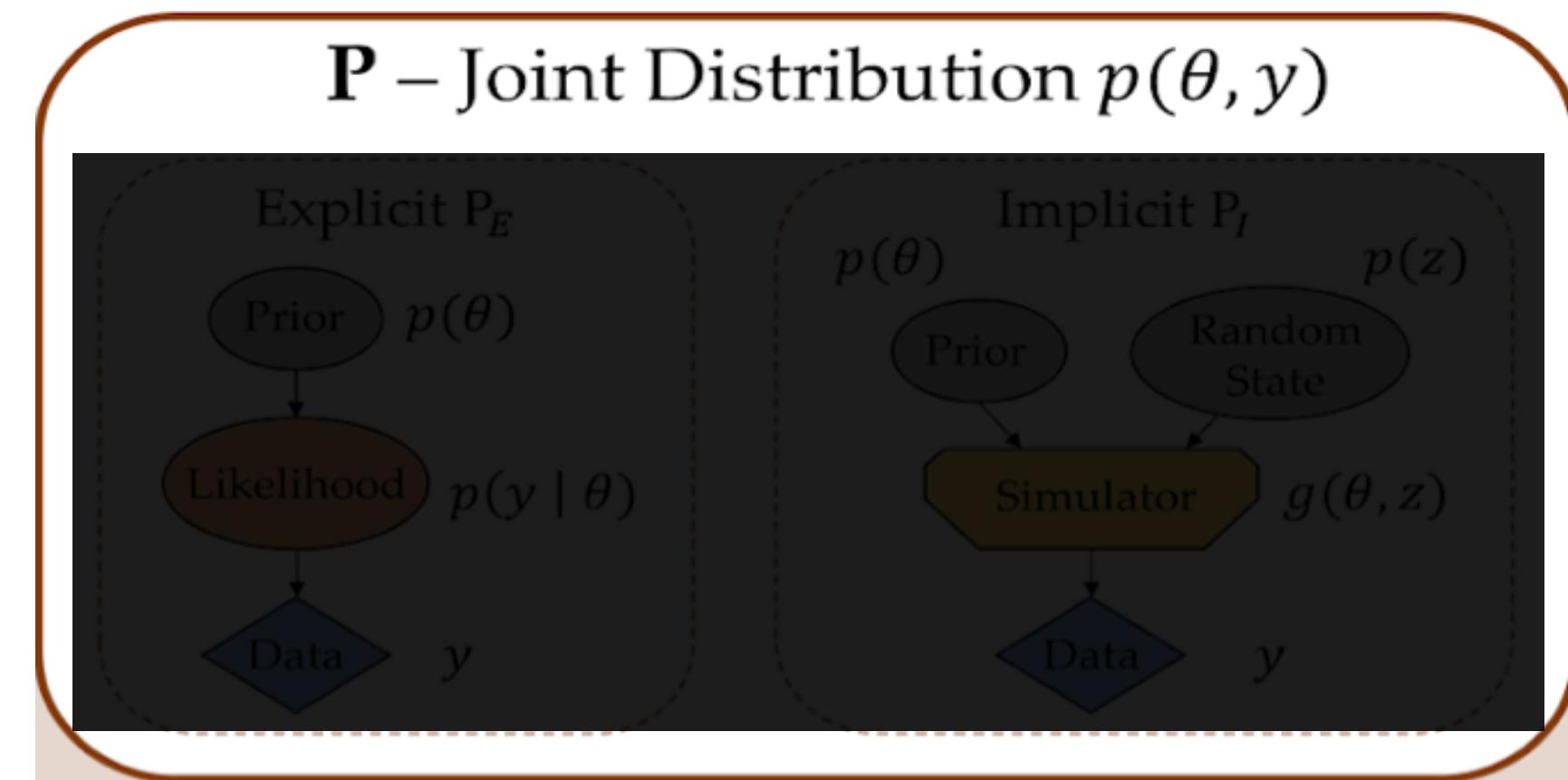
b. Orchestrate P, A, D

c. Identify P, A, D's elements

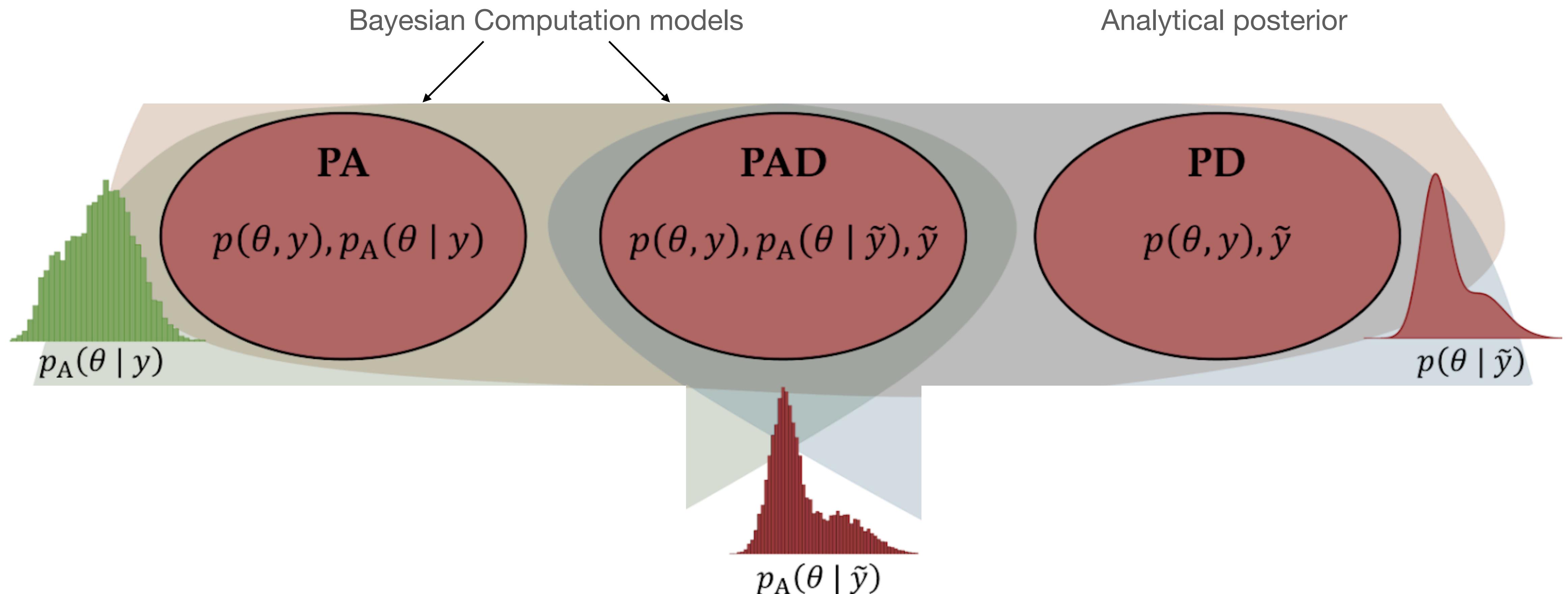


d. Orchestrate P, A, D's elements

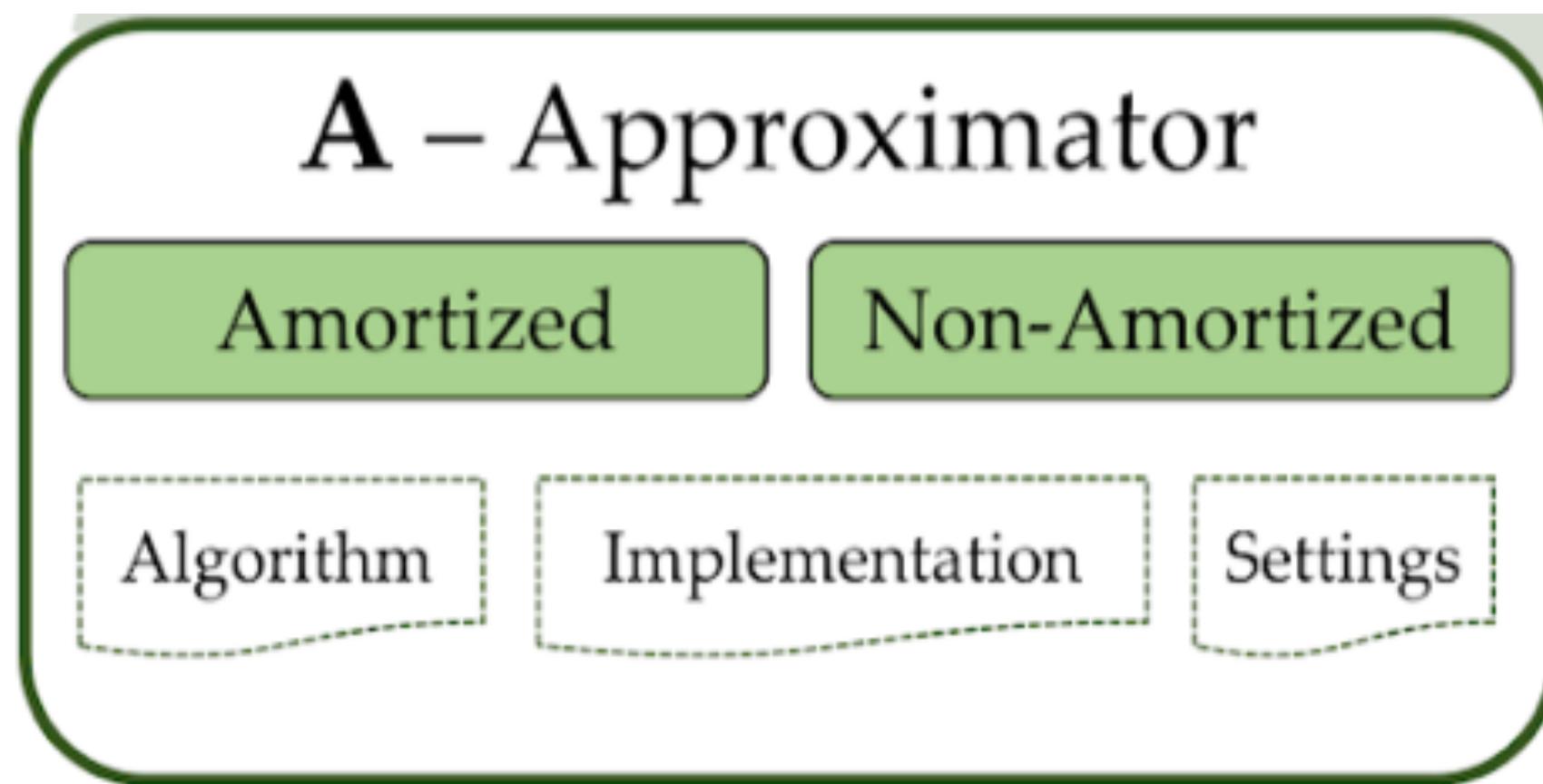
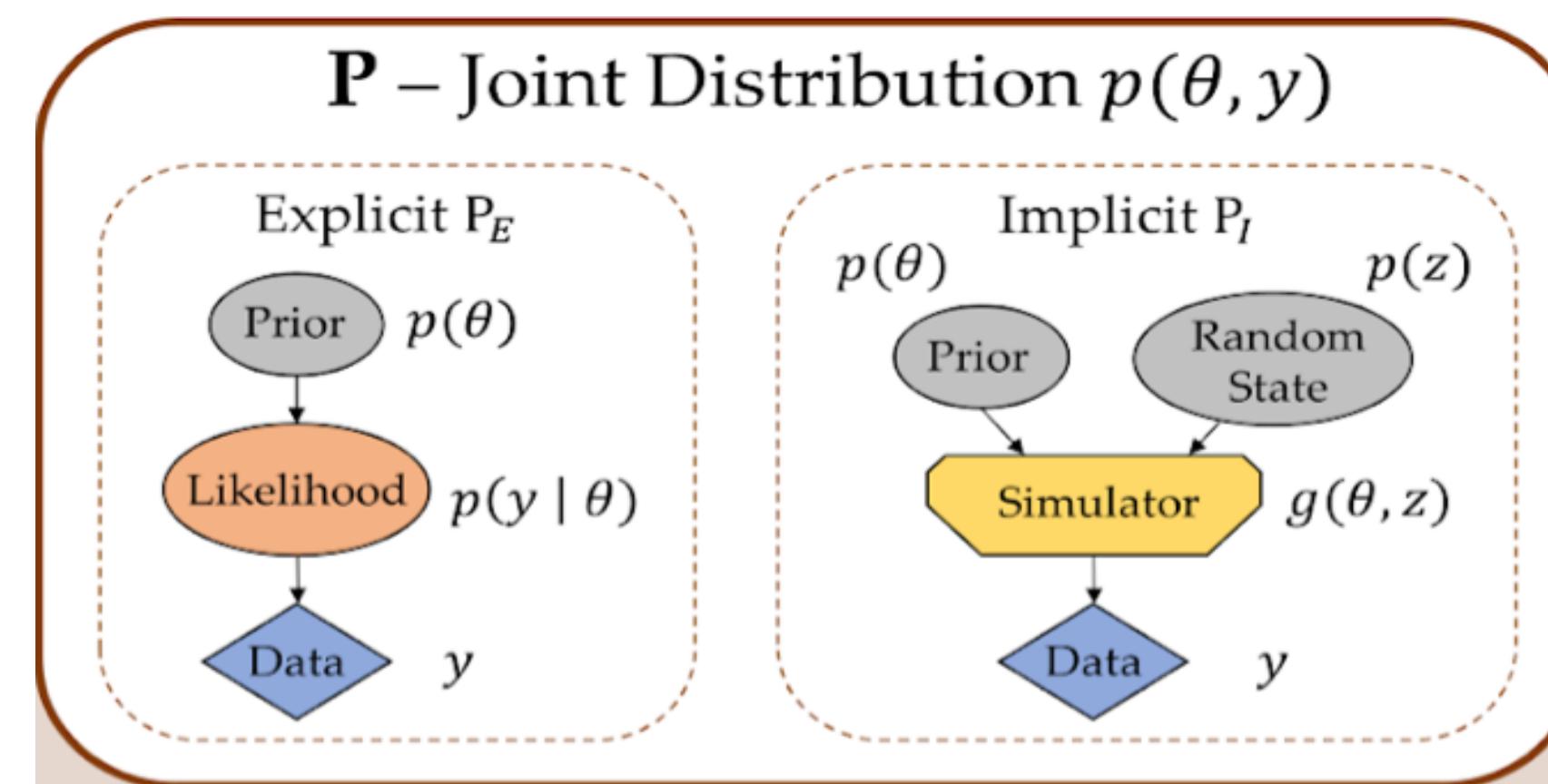
a. Identify P, A, D



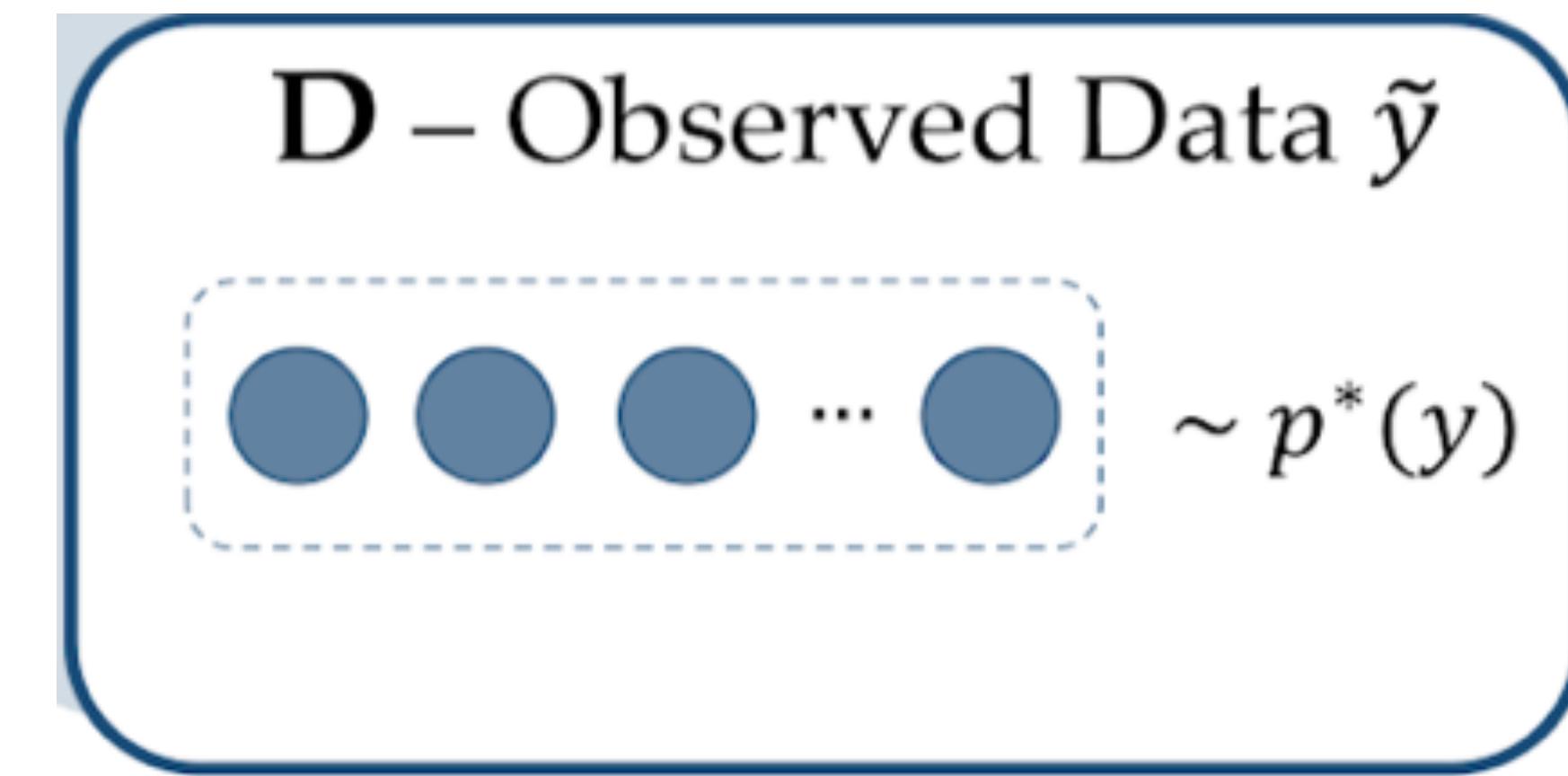
b. Orchestrate P, A, D as PA, PAD, PD



c. Identify P, A's Elements + d. Orchestrate



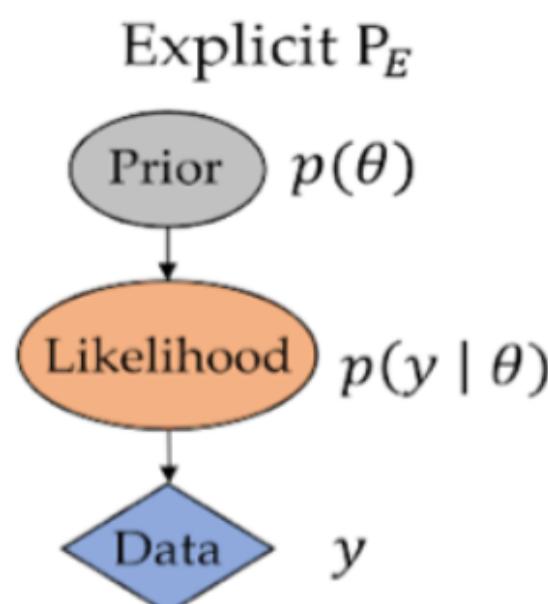
Settings e.g. are metric, stepsize, S, N, M



Example of normal model (P) and Stan MCMC (A)

a. Identify P, A

P joint distribution



```

function (N)
{
  mu <- rnorm(1, 0, 1)
  sigma <- abs(rnorm(1, 0, 1))
  y <- rnorm(N, mu, sigma)
  list(variables = list(mu = mu,
                        sigma = sigma),
       generated = list(N = N, y = y))
}
  
```

code forms $p(\theta, y)$

A approximator

MCMC

Non-Amortized

Implementation Stan file:

Settings Stan default

```

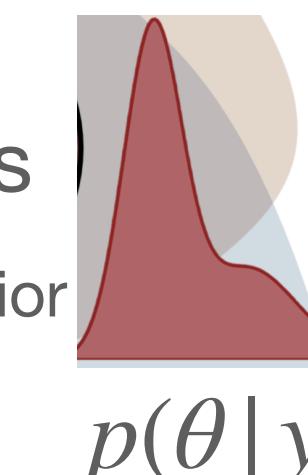
data {
  int<lower=0> N;
  vector[N] y;
}

parameters {
  real mu;
  real<lower=0> sigma;
}

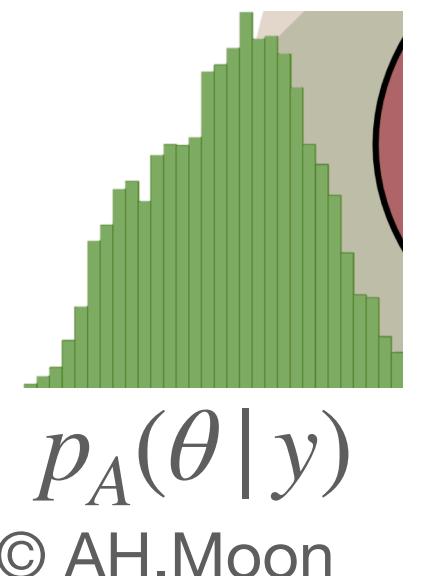
model {
  y ~ normal(mu, sigma);
  mu ~ normal(0, 1);
  sigma ~ normal(0, 1);
}
  
```

Stan file forms

Analytic posterior

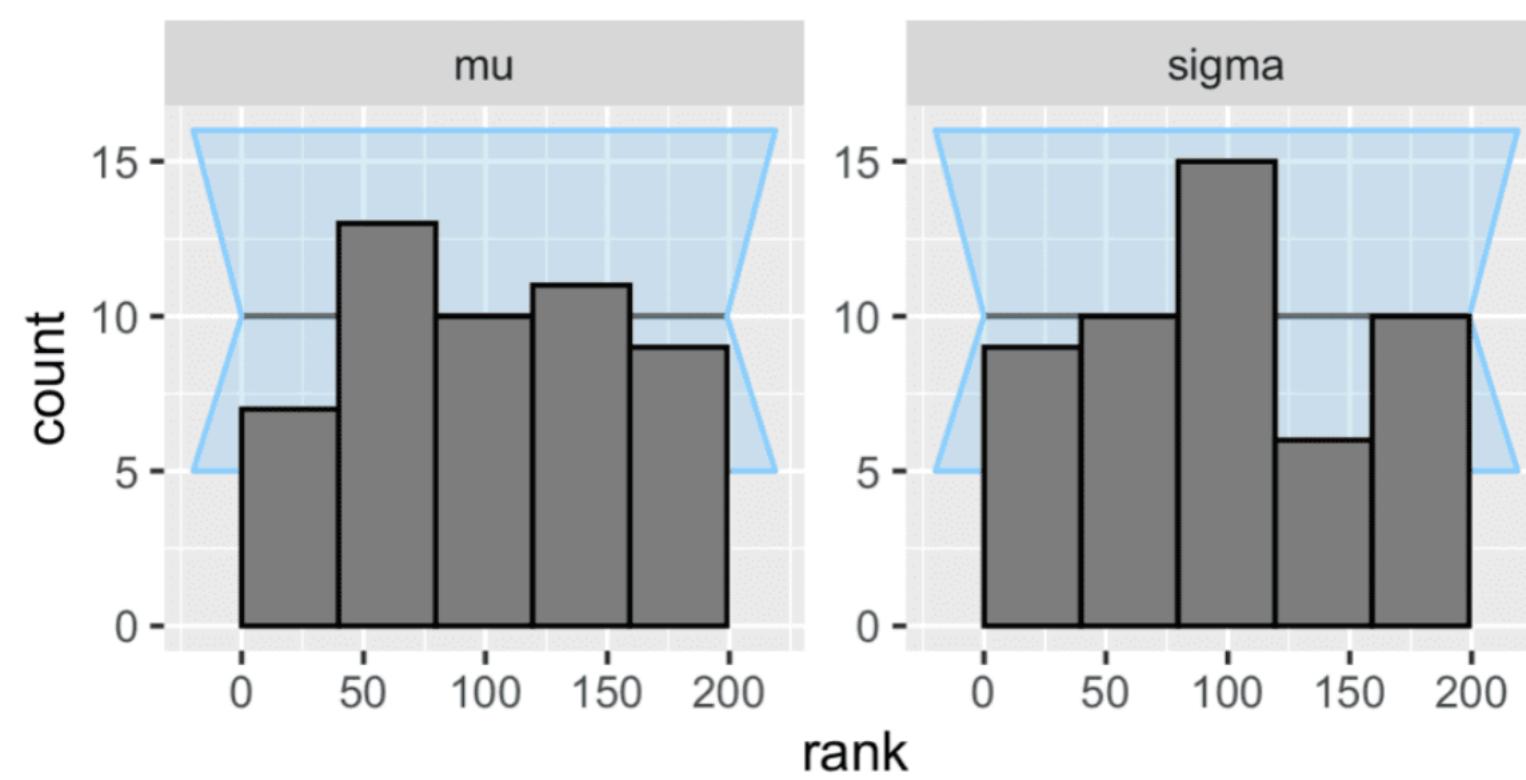


MCMC transforms to

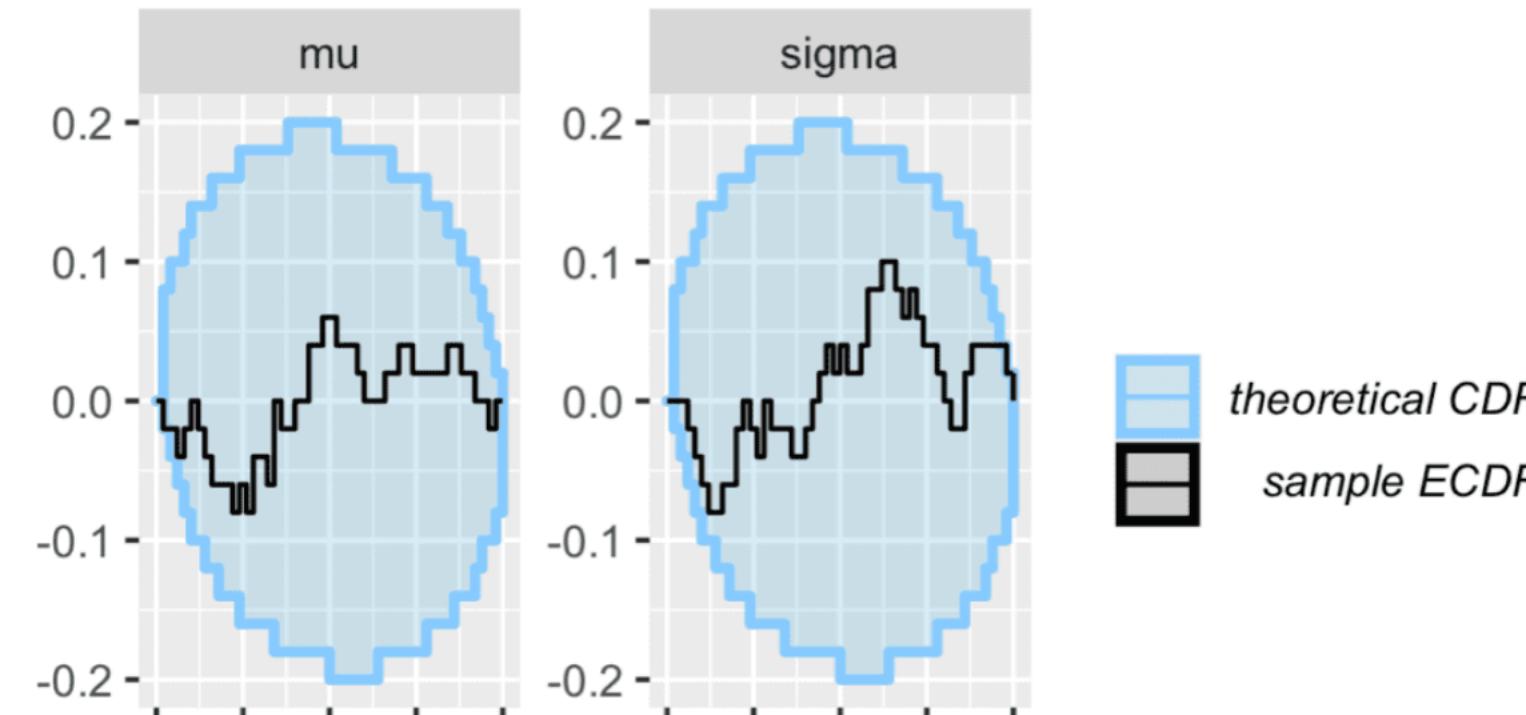


Example of normal model (P) and Stan MCMC (A)

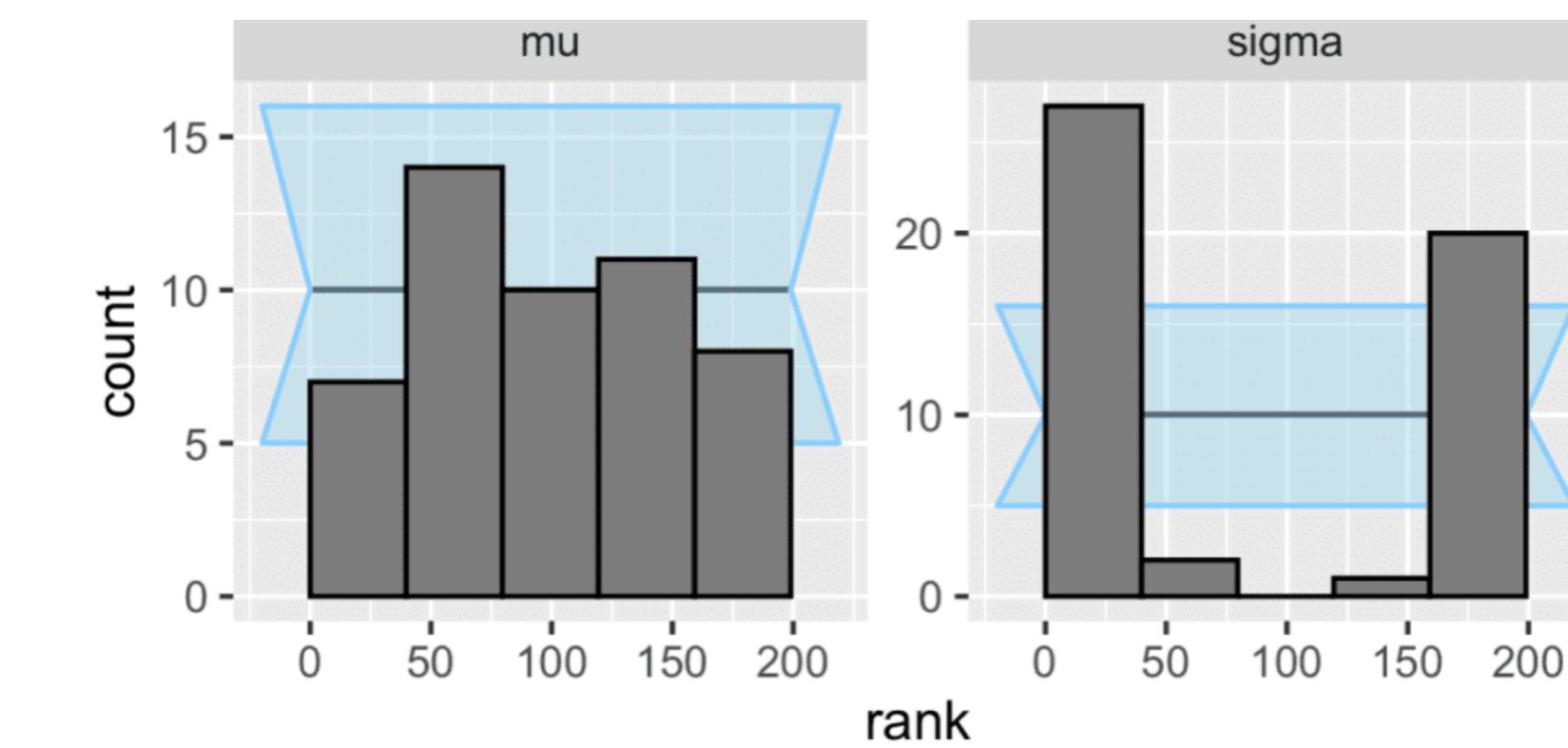
b. Check P, A is well-orchestrated



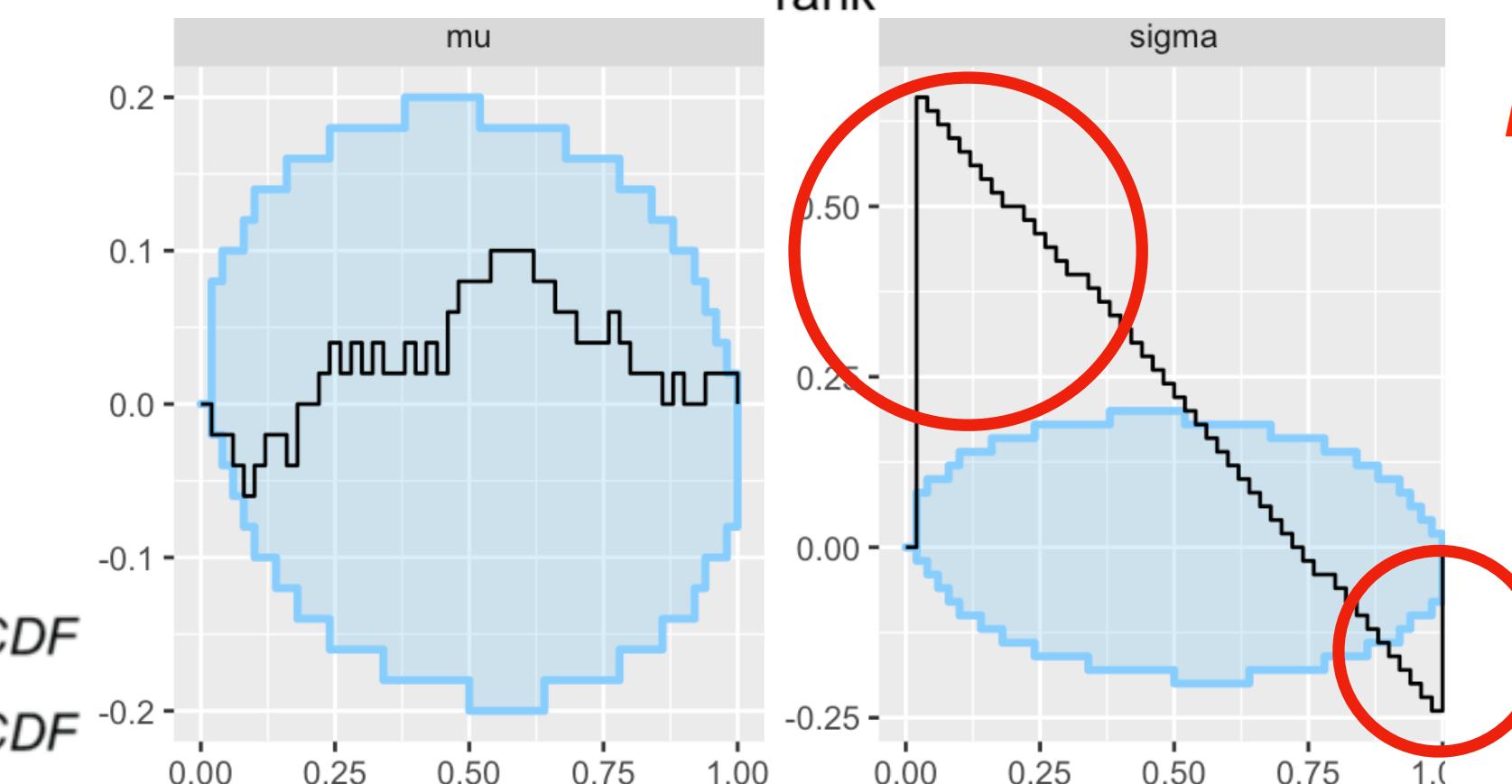
Pass!



theoretical CDF
sample ECDF



Fail!



Example of normal model (P) and Stan MCMC (A)

b. Cacophony

P joint distribution

```
function (N)
{
  mu <- rnorm(1, 0, 1)
  sigma <- abs(rnorm(1, 0, 1))
  y <- rnorm(N, mu, sigma)
  list(variables = list(mu = mu,
                        sigma = sigma),
       generated = list(N = N, y = y))
}
```

A approximator

```
data {
  int<lower=0> N;
  vector[N] y;
}

parameters {
  real mu;
  real<lower=0> sigma;
}

model {
  y ~ normal(mu, 1/sigma^2)
  mu ~ normal(0, 1);
  sigma ~ normal(0, 1);
}
```



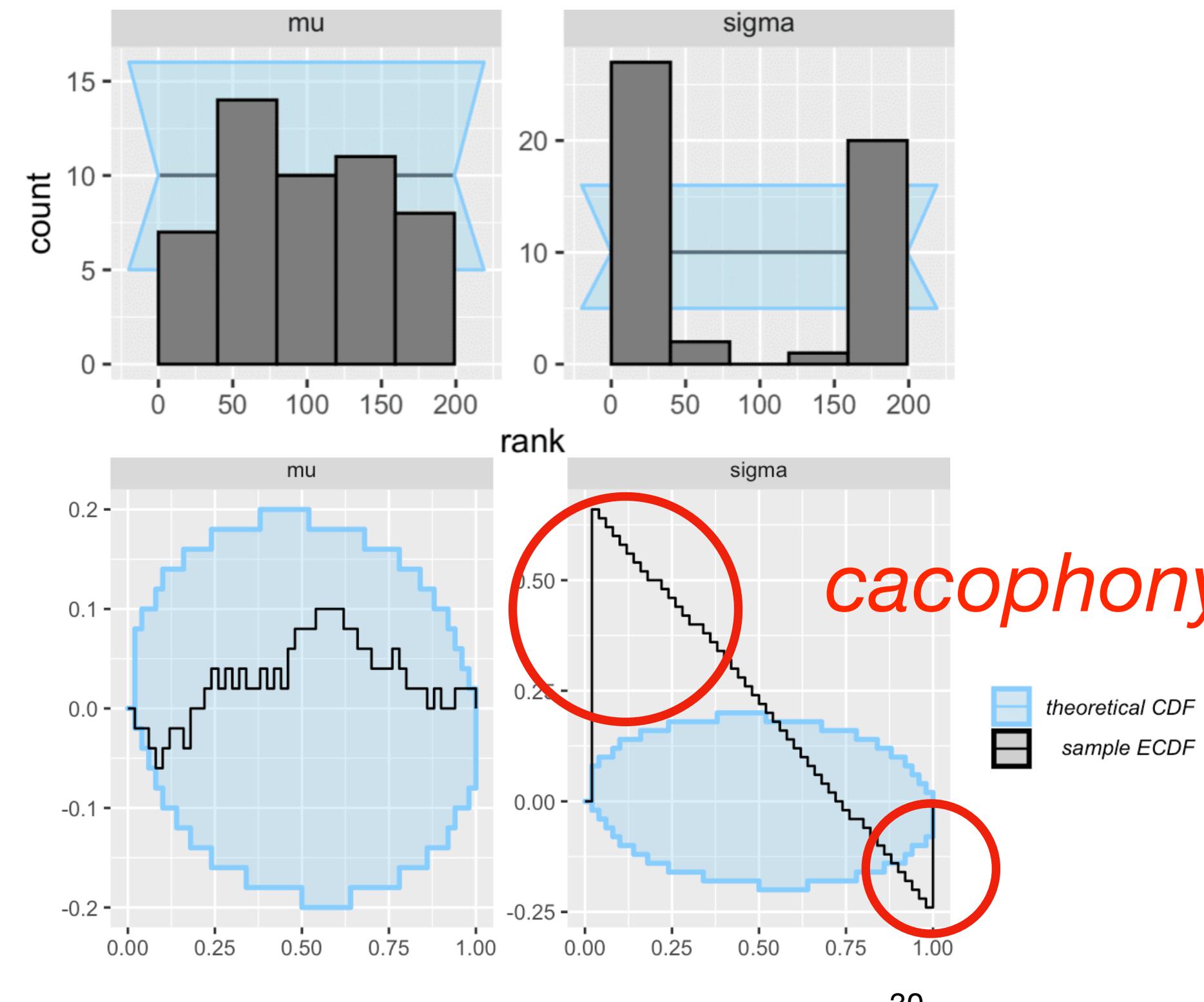
Misassume
Stan parametrizes
normal distribution
via precision

Example of normal model (P) and Stan MCMC (A)

b. Cacophony

Mismatch of P, A detected via SBC plots

`y <- rnorm(N, mu, sigma)` \longleftrightarrow `y ~ normal(mu, 1/sigma^2)`



Feedback to
discover implementation bug
from wrong assumption

cacophony!

Example of normal model (P) and Stan MCMC (A)

c. Identify A's elements: MCMC's settings

A approximator

MCMC

Non-Amortized

Implementation

Stan file

Settings

Stan default:

```
.sample(  
    chains = 4,  
    iter_sampling = 1000,  
    thin = 4,  
    max_treedepth = 20,  
    adapt_delta = 0.9,  
    step_size = NULL,  
)
```

SBC results

can be **Settings**-sensitive
(verified below)

SBC for ADVI and optimizing in Stan includes

- inability of the approximation family to capture the true posterior
- Potential bug in convergence metric of Stan ADVI approximator
- SBC plot sensitivity analysis on tolerance of ADVI Mean-field approximator
- SBC can justify using algorithm parameters chose to be computationally efficient

Link to Package Vignettes for other Examples

Using SBC for debugging/validating Stan models

Case studies showing how problems in a Stan model can be discovered with SBC.

[Discovering bad parametrization with SBC](#)

[Discovering indexing errors with SBC](#)

[Small model implementation workflow](#)

Additional use cases and advanced topics

[Limits of SBC](#)

[SBC for ADVI and optimizing in Stan \(+HMMs\)](#)

[Implementing a new backend \(algorithm\) + frequentist approximations](#)

[SBC for brms models](#)

[SBC with discrete parameters in Stan and JAGS](#)

[Rejection sampling in simulations](#)

Two SBC Use Cases for Orchestrating Elements

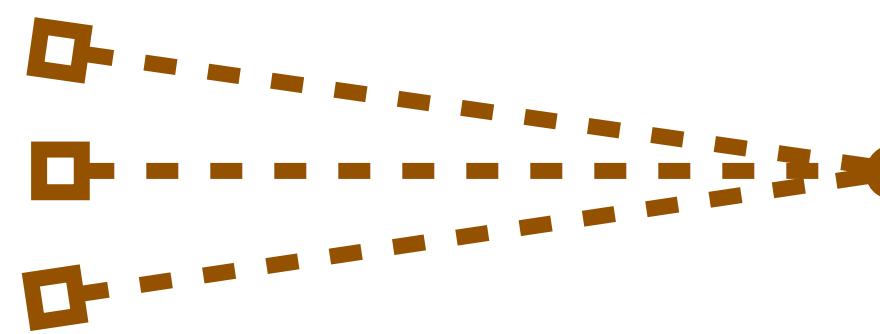
1. my P and A

$$\begin{array}{cc} \textbf{P} & \textbf{A} \\ P(\theta, y) & P_A(\theta | y) \end{array}$$



e.g. debug model implementation

2. my A with canonical Ps



e.g. sequential Monte Carlo sampler with
eight school model (hierarchical Bayesian),
Birthday model (Time series)

So far, SBC was characterized as

Rehearsal to orchestrate (P, A, D)

SBC Orchestrates Computational Bayes Elements

a. Identify P, A, D



b. Orchestrate P, A, D

c. Identify P, A, D's elements



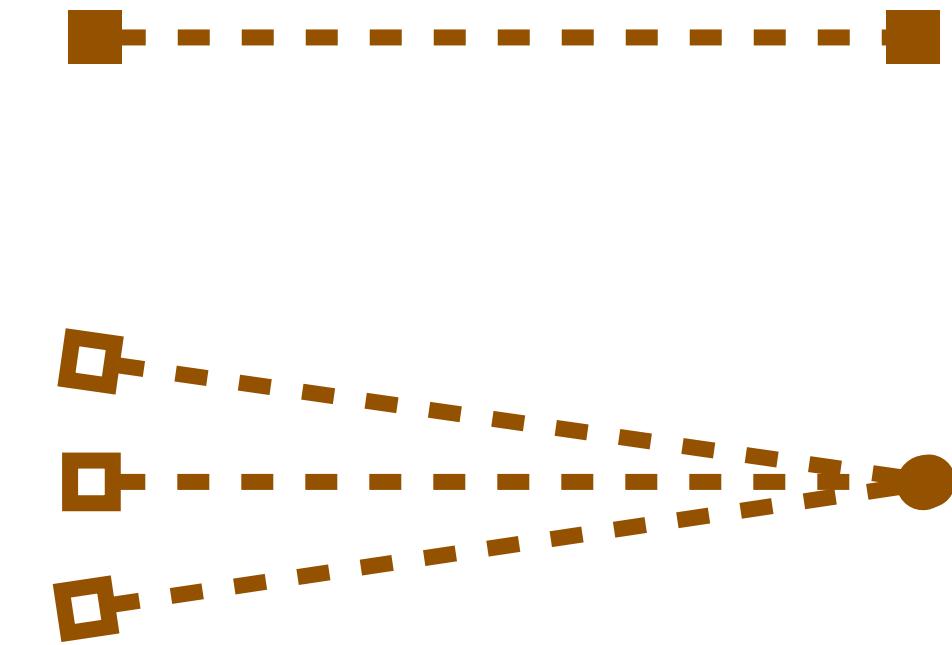
d. Orchestrate P, A, D's elements

Hidden Known Problem of SBC

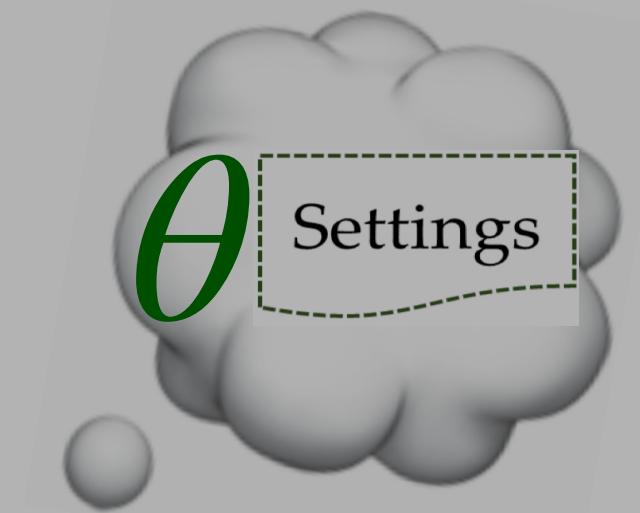
1. my P and A

$$\begin{array}{cc} \textbf{P} & \textbf{A} \\ P(\theta, y) & P_A(\theta | y) \end{array}$$

2. my A with canonical Ps



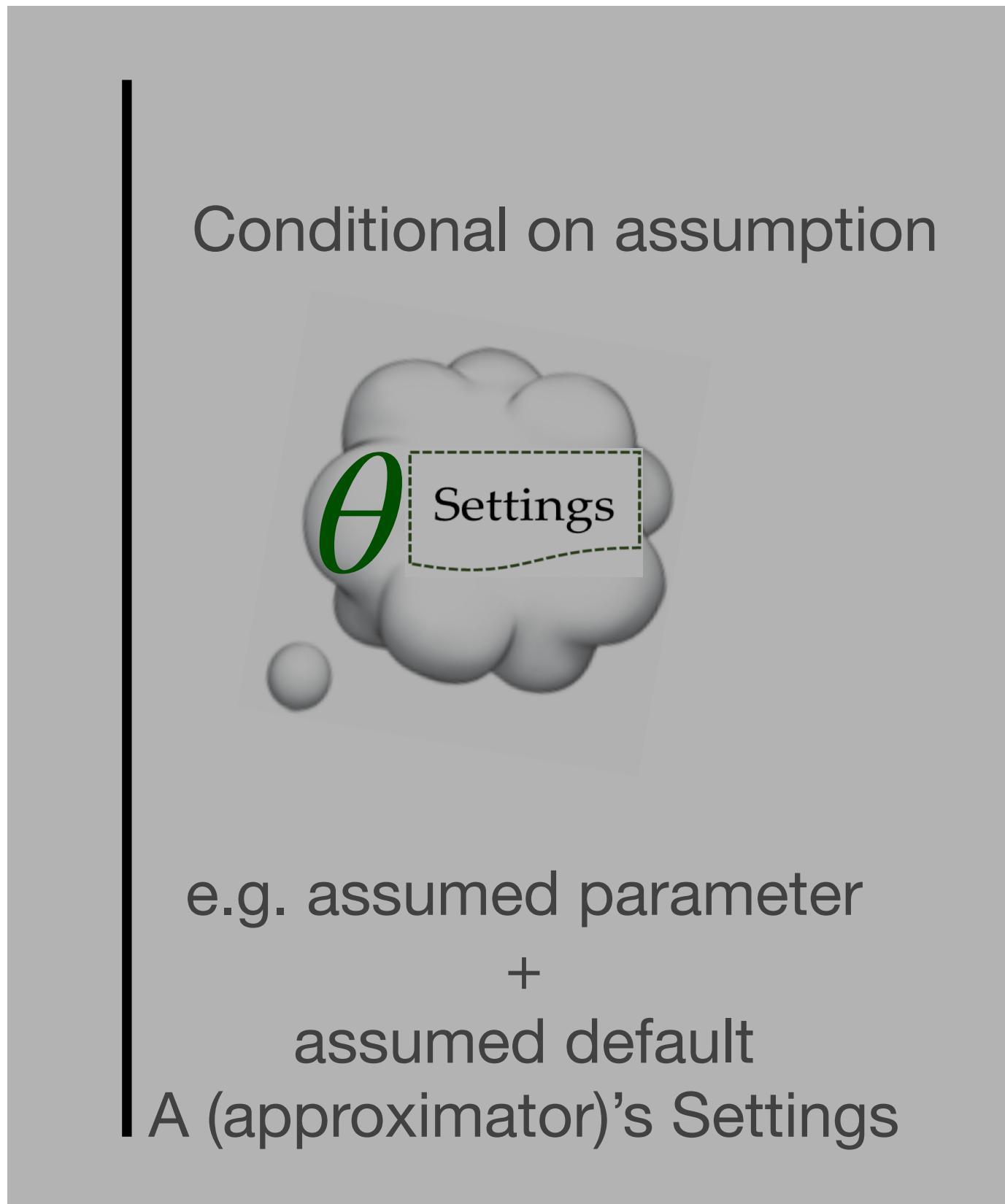
Conditional on assumption



e.g. assumed parameter
+
assumed default
A (approximator)'s Settings

Hidden Known Problem of SBC

e.g. assumed parameter



= FOR EACH VALUE OF a, b:
RUN SEPARATE STAN FILE (*ab_SBC.stan*)

```
data {  
    int<lower = 1> N;  
    real<lower = 0> a;  
    real<lower = 0> b;  
}  
transformed data { // these adhere to  
    real pi_ = beta_rng(a, b);  
    int y = binomial_rng(N, pi_);  
}  
parameters {  
    real<lower = 0, upper = 1> pi;  
}  
model {  
    target += beta_lpdf(pi | a, b);  
    target += binomial_lpmf(y | N, pi);  
}  
generated quantities { // these adhere to  
    int y_ = y;  
    vector[1] pars_;  
    int ranks_[1] = {pi > pi_};
```

ab_SBC.stan

a, b
Assumed or
Estimated value?

**My desired character of
Simulation-based Calibration is**

**Dynamic, Data, Desire-based
Orchestration**

Actionable Workflow with Sequential Bayesian Update

- a. Check $\Theta_t, \theta\theta$ passes simulation-based calibration
- b. Update Θ_t to Θ_{t+1} with real data
- c. Prescribe policy based on $E[f(\Theta_{t+1}, \theta, \tilde{y}) | \theta, \theta, \theta]_{\theta=1..P}$

-> Example next pg.

a: generate then estimate, c: generate only

3. Orchestrate with Dynamics, Tempo!



Andante

Violin

Viola

Saxophone

The musical score consists of three staves. The top staff is for the Violin, the middle for the Viola, and the bottom for the Saxophone. All staves are in common time (indicated by a '4'). The Violin and Viola staves begin with a quarter note followed by a series of eighth-note pairs. The Saxophone staff begins with a quarter note followed by a series of sixteenth-note pairs. The notation is in black ink on white paper.

Grave = very slow, solemn

Largo = slow and broad

Adagio = slow

Andante = walking pace, medium speed

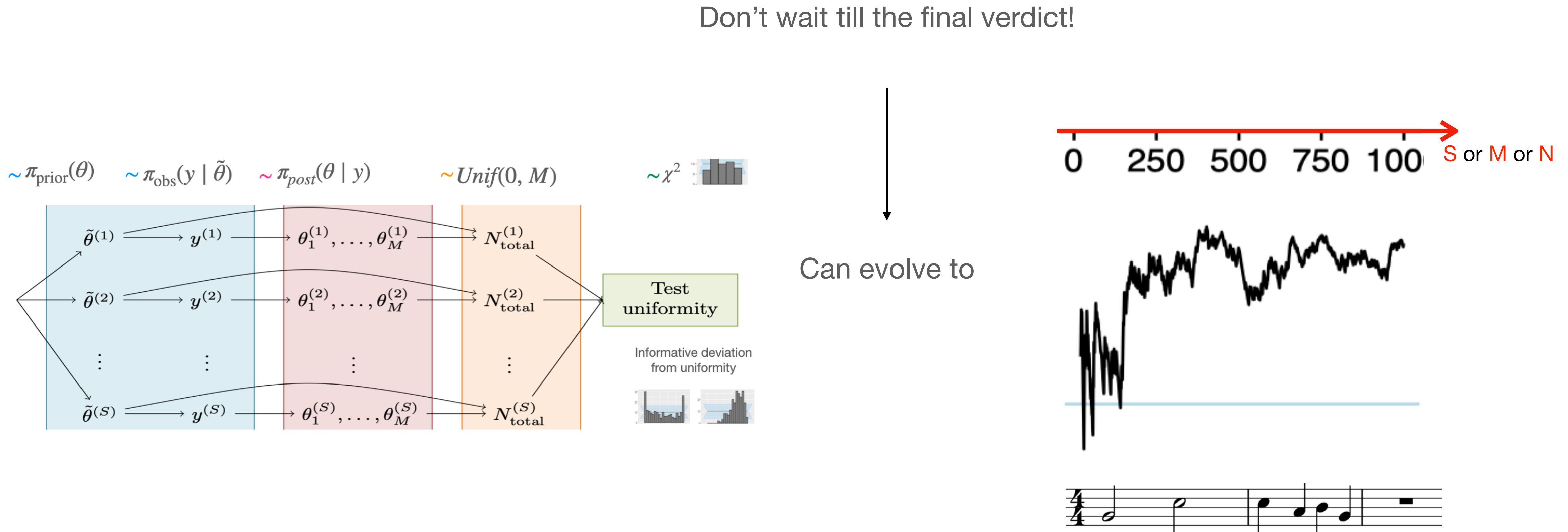
Moderato = moderate speed

Allegro = happy, upbeat, fast

Vivace = lively, fast

Presto = very fast

Static to Dynamic SBC Diagnostic (P, A, D)_t



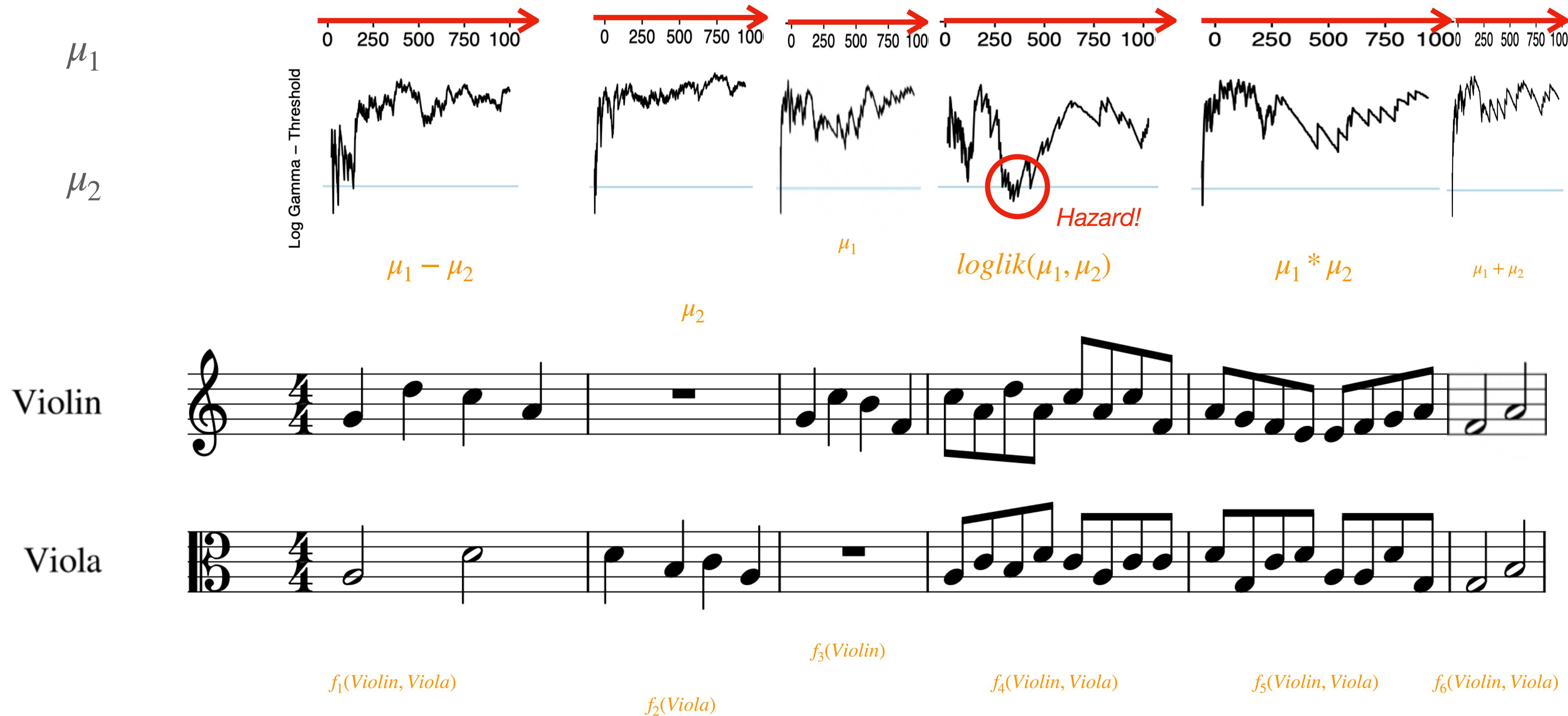
Plots from coauthoring paper: Modrak et al, *Good choice of test quantities substantially improves the sensitivity of simulation-based calibration for validating Bayesian computation*

4. Orchestrate for Desire



arrange
elements of situation
to surreptitiously produce
desired effect

Dynamic Diagnostic $(P, A, D)_t$ + Test Quantities



Plots from coauthoring paper: Modrak et al, *Good choice of test quantities substantially improves the sensitivity of simulation-based calibration for validating Bayesian computation*

5. Orchestrate based on Data



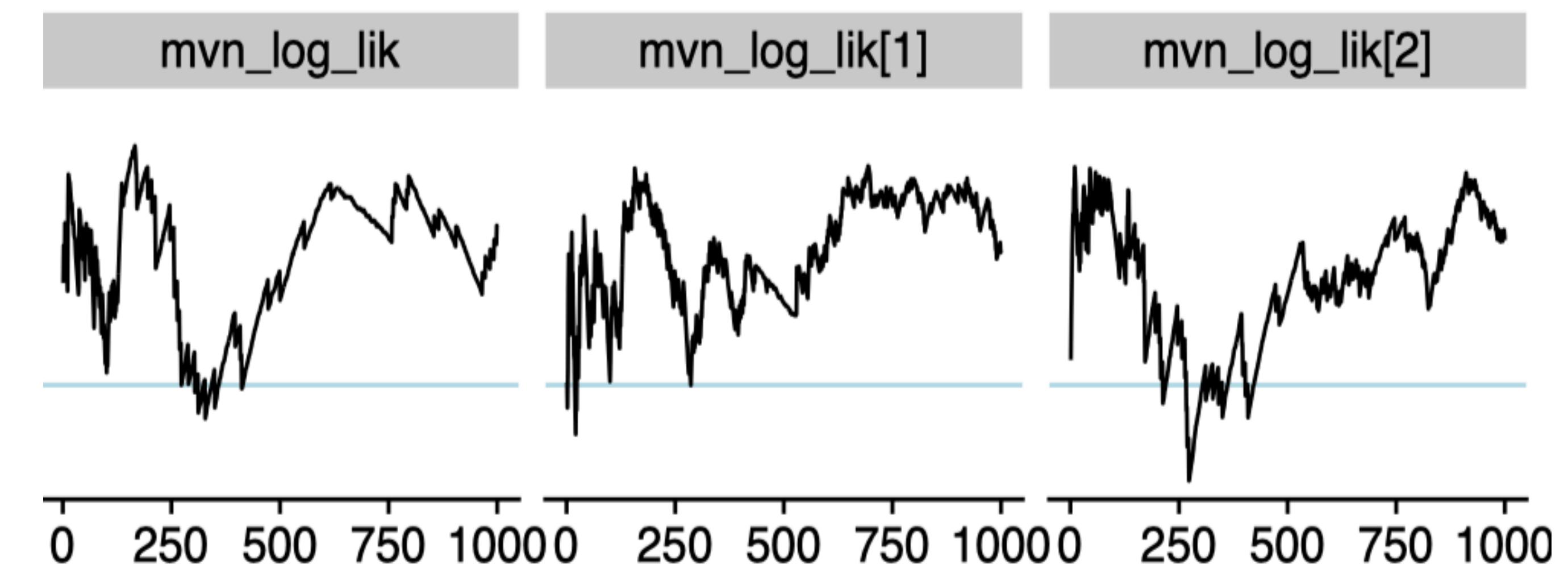
arrange
elements of situation
to surreptitiously produce
desired effect

Data-based Test Quantities

$$\mu_1, \mu_2 \sim MVN(0, \Sigma)$$

$$\mathbf{y}_1, \dots, \mathbf{y}_N \sim MVN(\mu, \Sigma)$$

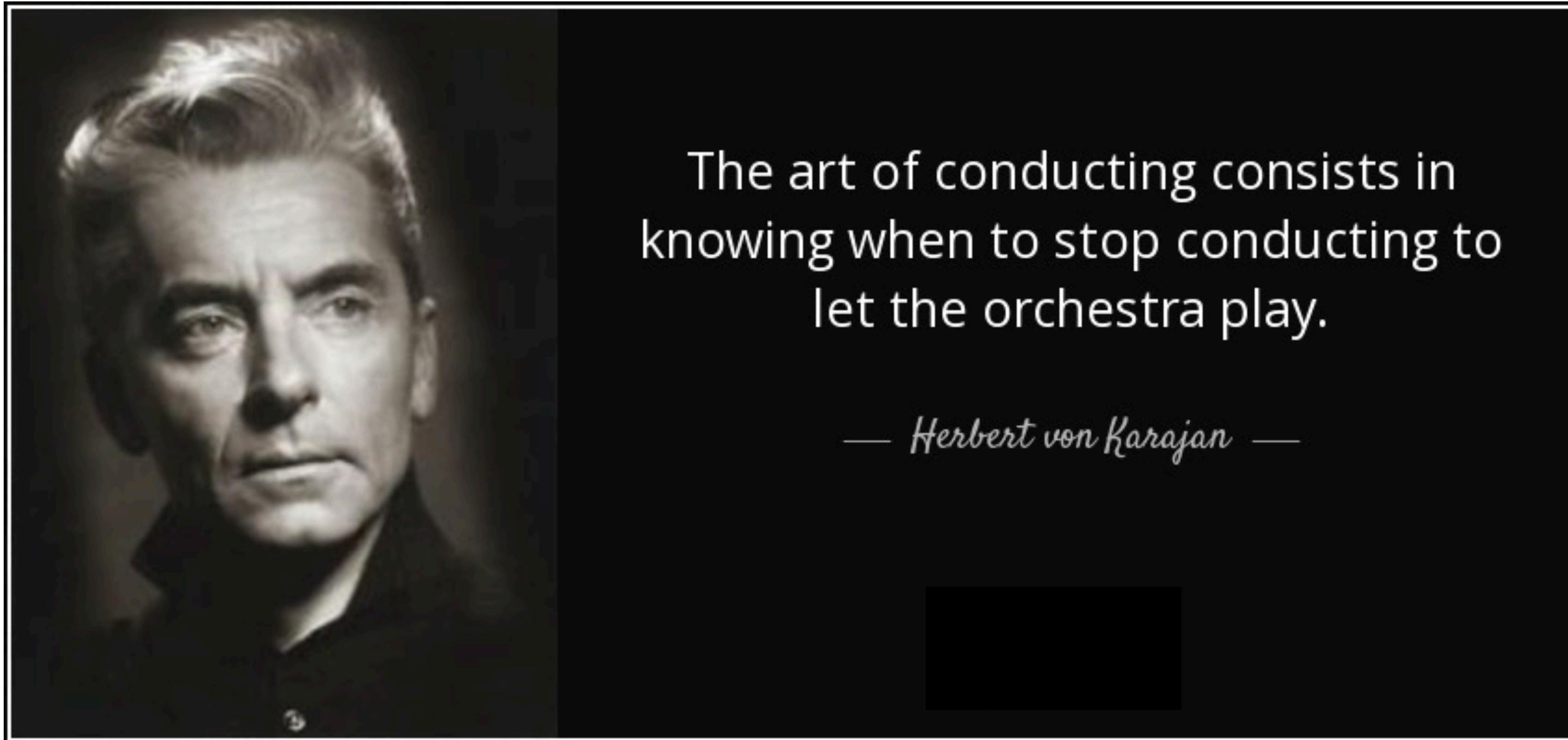
$$\Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$



mvn_log_lik is joint likelihood,
the right two are pointwise likelihoods $\pi(y_1 | \mu_1, \mu_2)$ and $\pi(y_2 | \mu_1, \mu_2)$

Plots from coauthoring paper: Modrak et al, *Good choice of test quantities substantially improves the sensitivity of simulation-based calibration for validating Bayesian computation*

6. Orchestrate based on Customized Precision



The art of conducting consists in
knowing when to stop conducting to
let the orchestra play.

— *Herbert von Karajan* —

Colors in SBC

Assumed parameter	Moss #2D834A
Driving data	Aqua #2F86FA
Estimated parameter	Red #DE351B
Computed Estimated parameter	Strawberry #F13C87
Target simulated	Grape #832CF9
Target simulated measured	Plum #822385
Policy parameter	Tangerine #F08000