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

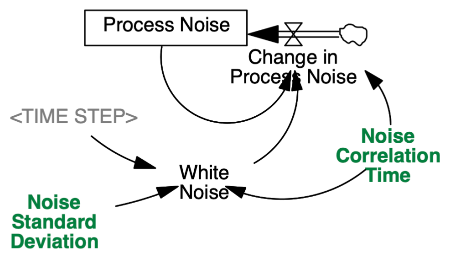
different language is the most difficult part

Mapping Variables of Computational Bayes and Dynamics

	Latent parameter			Observable			
Comp.Bayes, Stan	θ hyper-param.	θ param.		y Simulated		$\tilde{y}$ Observed data	
System Dynamics	θ Assumed param.	θ Estimated param.	$\boxed{\theta}$ Policy param.	$f(\theta, t)_{t=1..T}$ Target sim.	$f(\theta, t)_{t=1..T}$ Target sim. measured	$\boxed{* \epsilon_t}$	$\boxed{\tilde{y}_{t=1..T}}$ Driving Data

$\boxed{}$: variable types in system dynamics but haven't seen in Stan

$* \epsilon_t$: multiplicative process noise



Variable Semantics in Dynamic Model

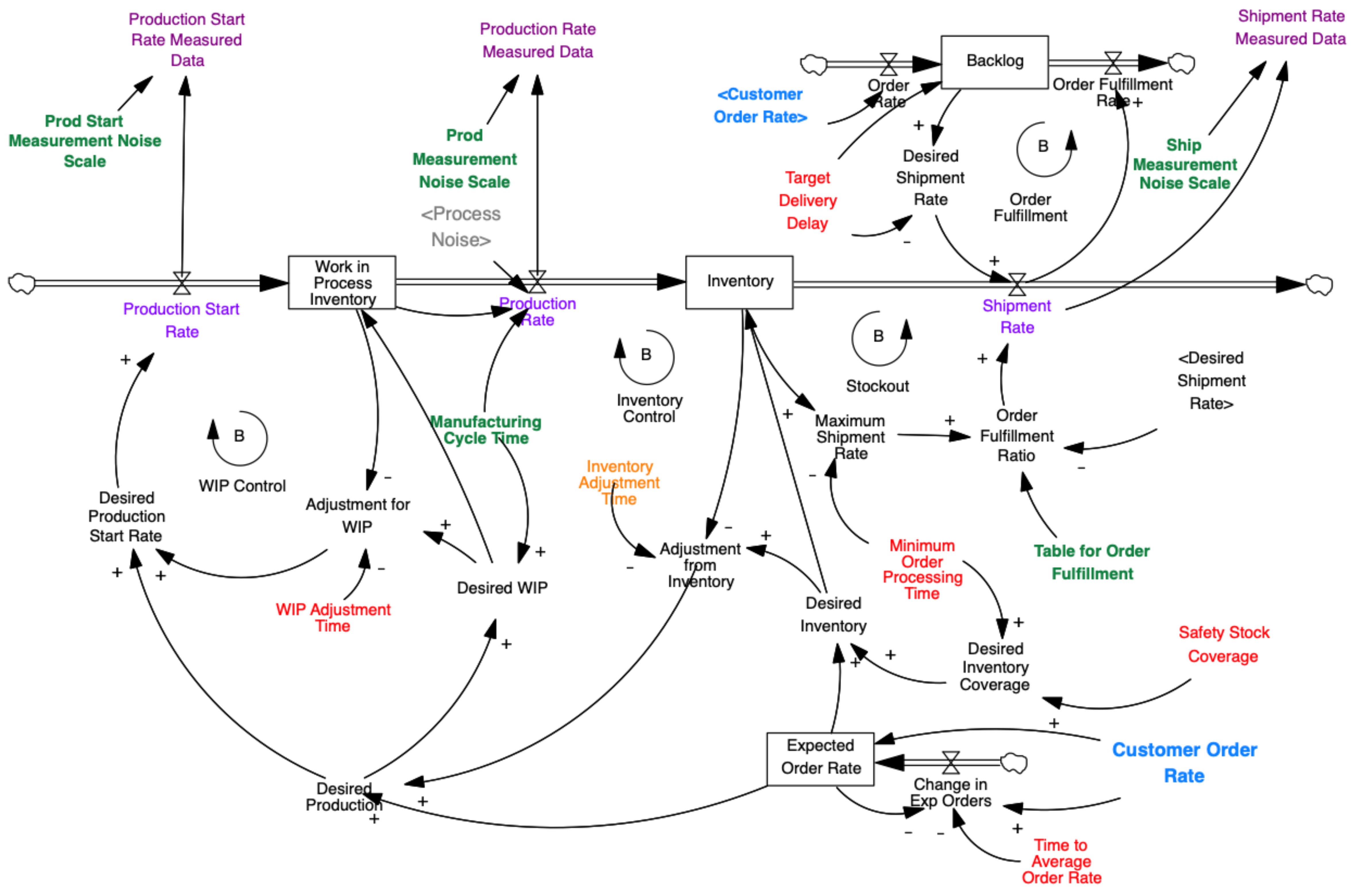
Inventory management example

Inventory manager asks:

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

θ $\quad\quad$ θ $\quad\quad\quad\quad$ y $\quad\quad$ $\tilde{y}$

θ	θ	θ	$f(\theta, t)_{t=1..T}$	$f(\theta, t)_{t=1..T} * \epsilon_t$	$\tilde{y}_{t=1..T}$
Assumed param.	Estimated param.	Policy Param.	Target sim.	Target sim. measured	Driving Data



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

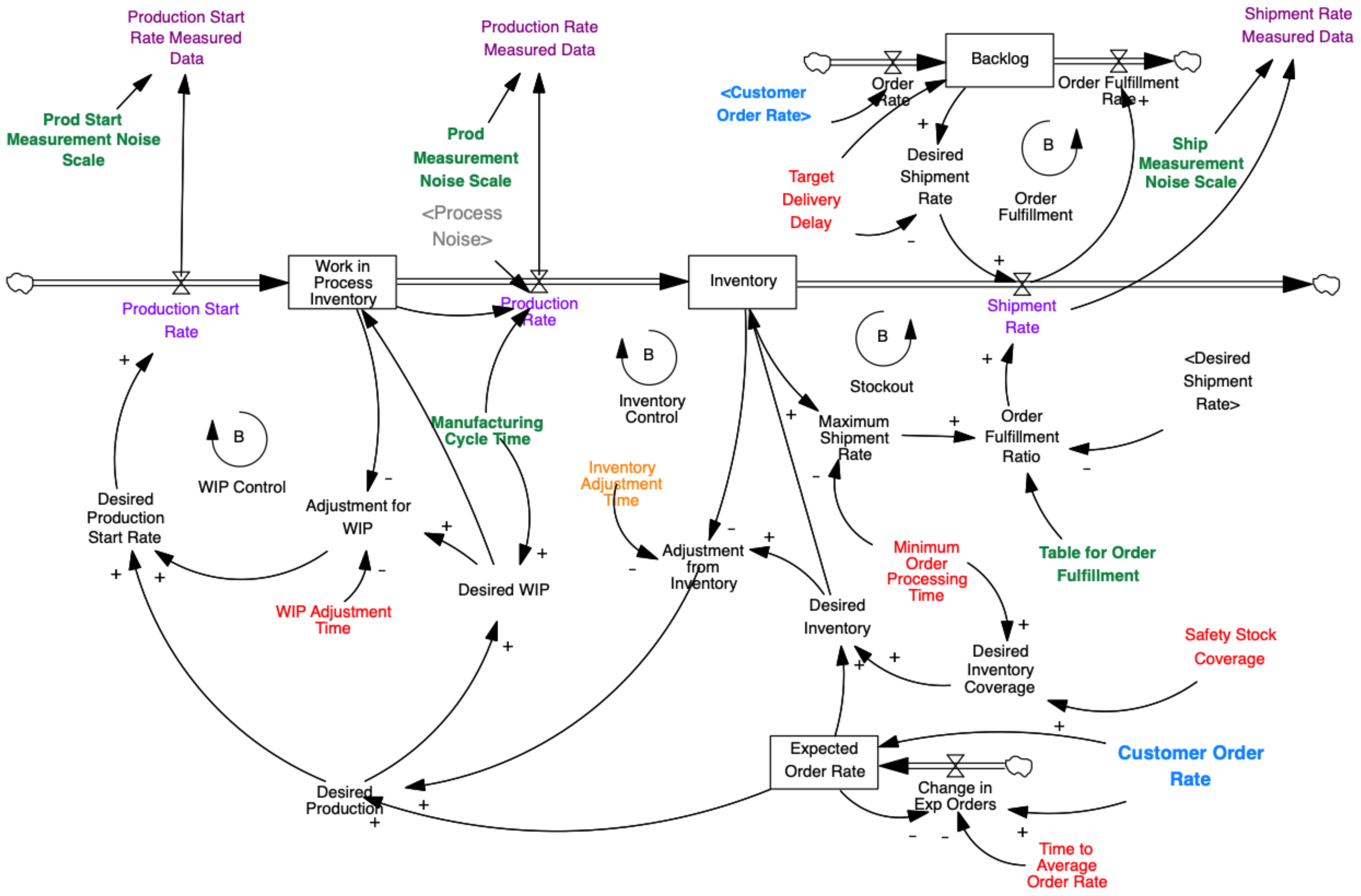
- Time to average order rate (1~3 days)
- Target delivery delay (2~4 days)

$\theta_{assumed}$

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

$\tilde{y}$

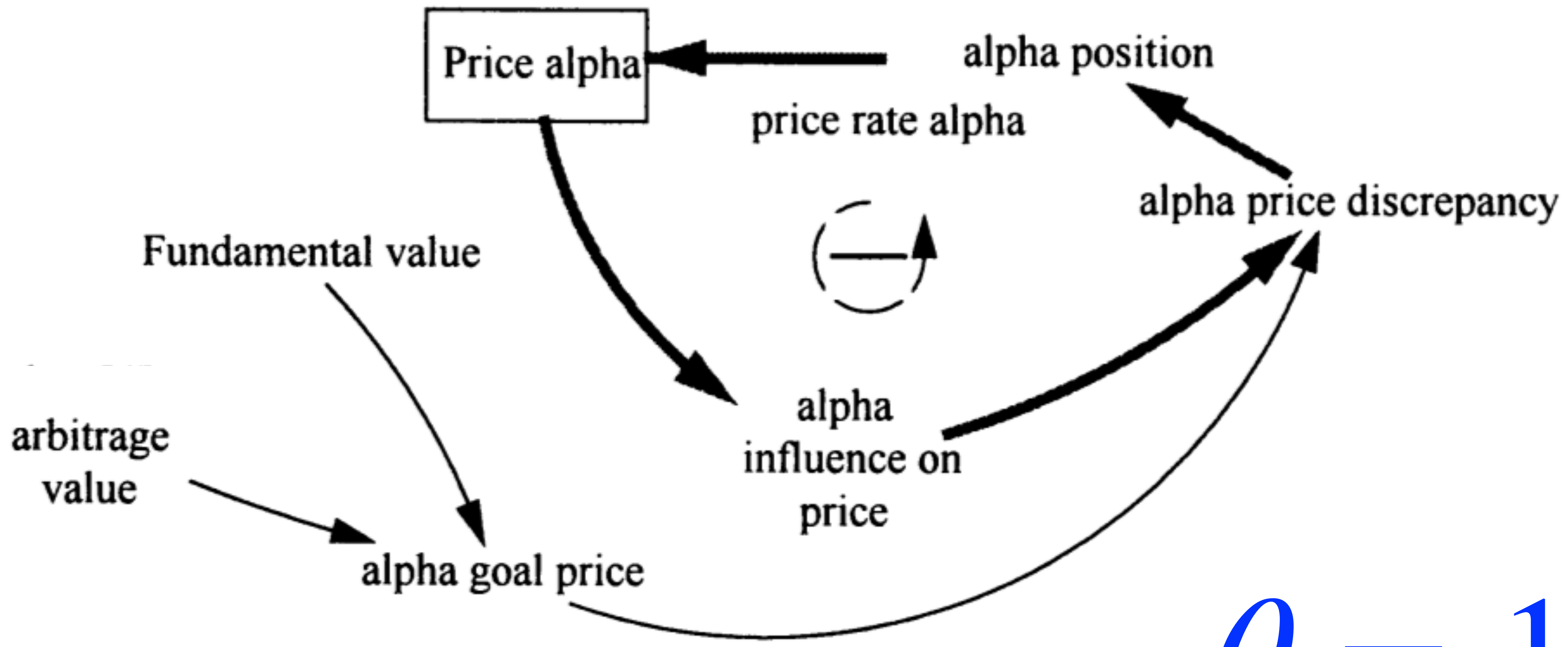
- Customer order rate (observed time series, 5,4,7,..)



Differential equations variable

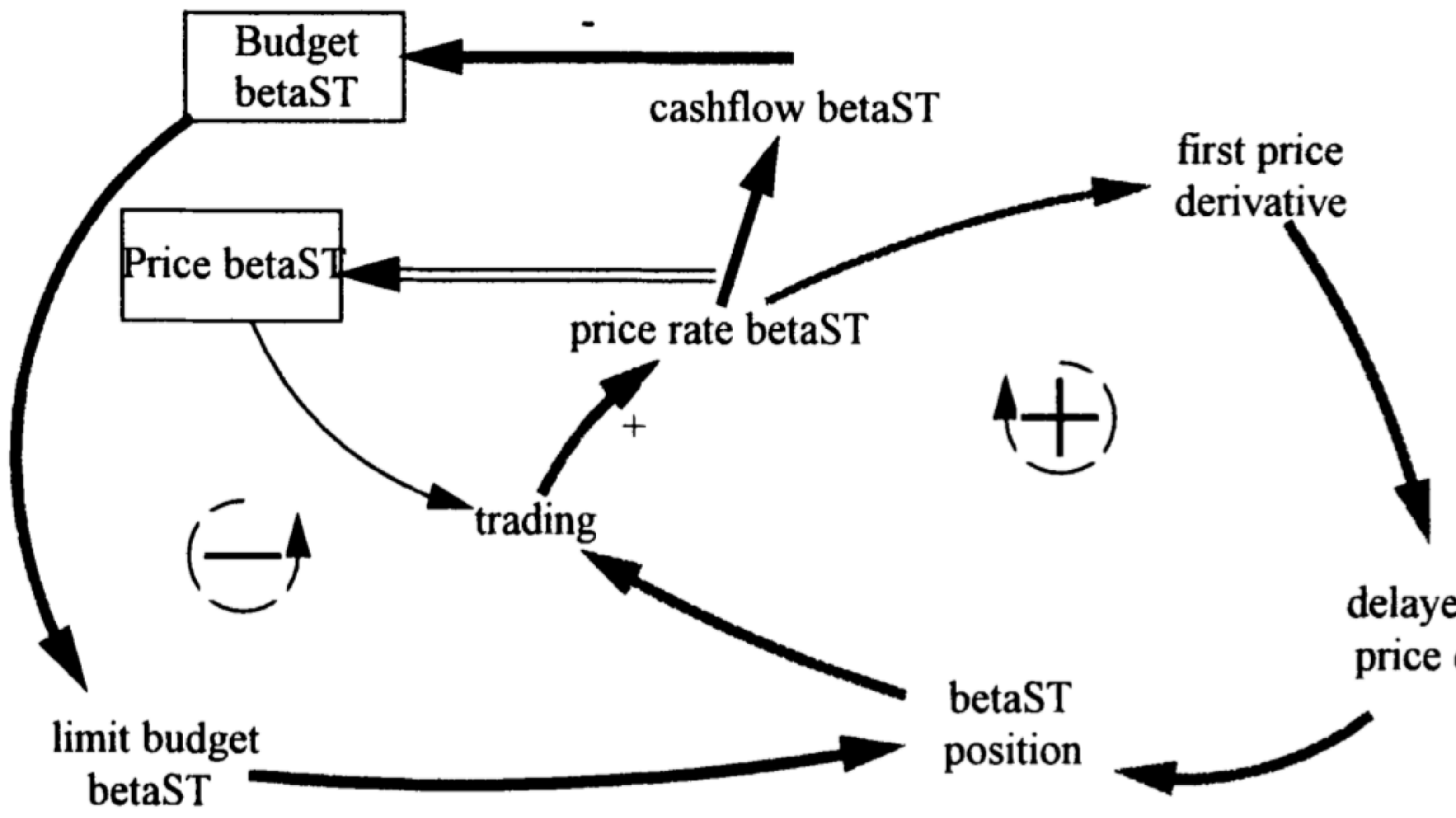
Behavioral finance example

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”



$\theta = 2$

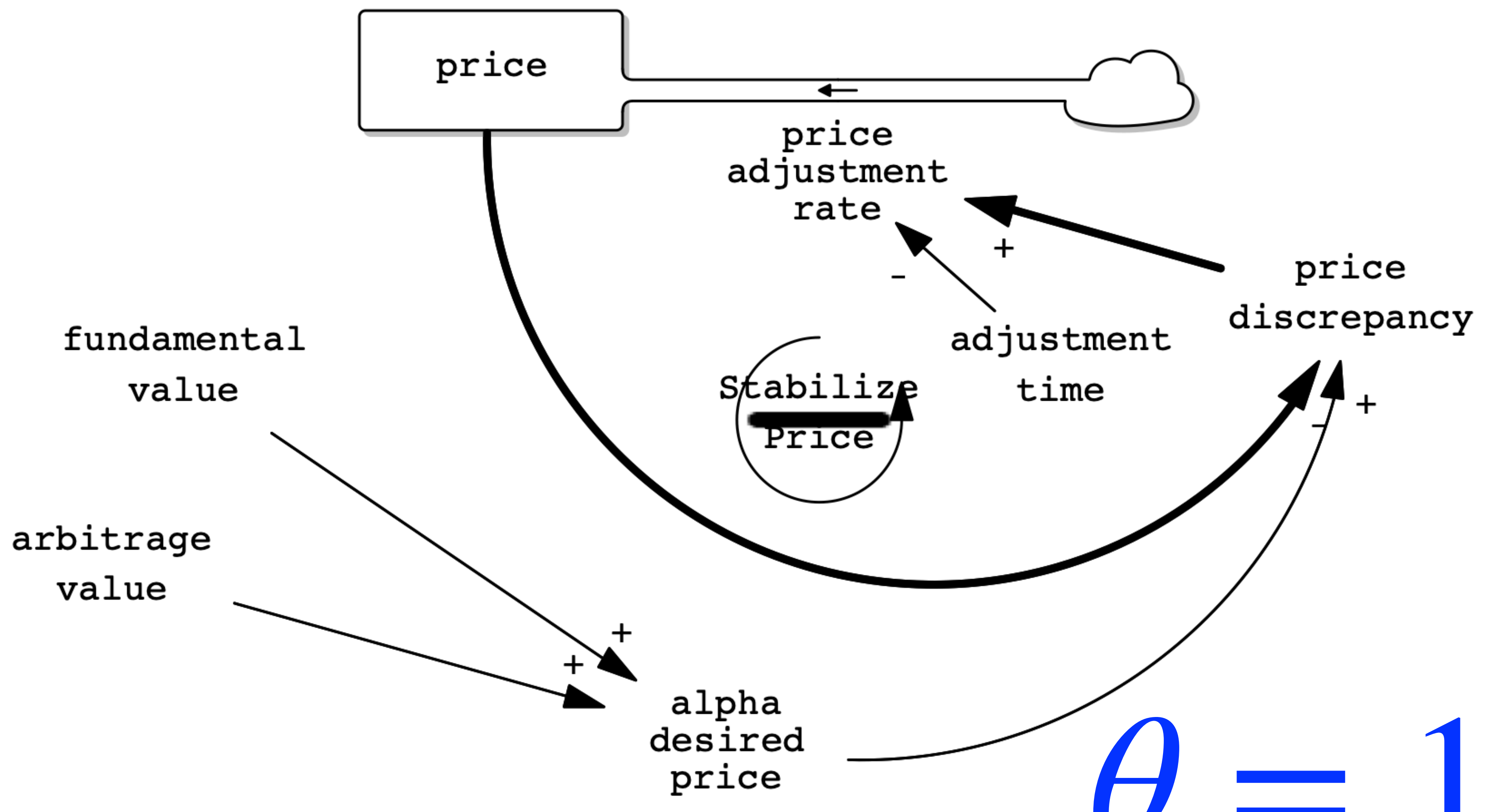
Heuristic
“In phase with price trend creating noise”

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

Kunsch et al 2000, *A System Dynamics Model of Stock Price Movements*

Differential equations variable

Introducing Vensim to Stan auto translator



$\theta = 1$

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 Θ_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(Θ_{t+1},θ,θ,$\tilde{y}$)|θ,θ,θ]$_{\theta=1..P}$

-> Example next pg.

a: generate then estimate, c: generate only

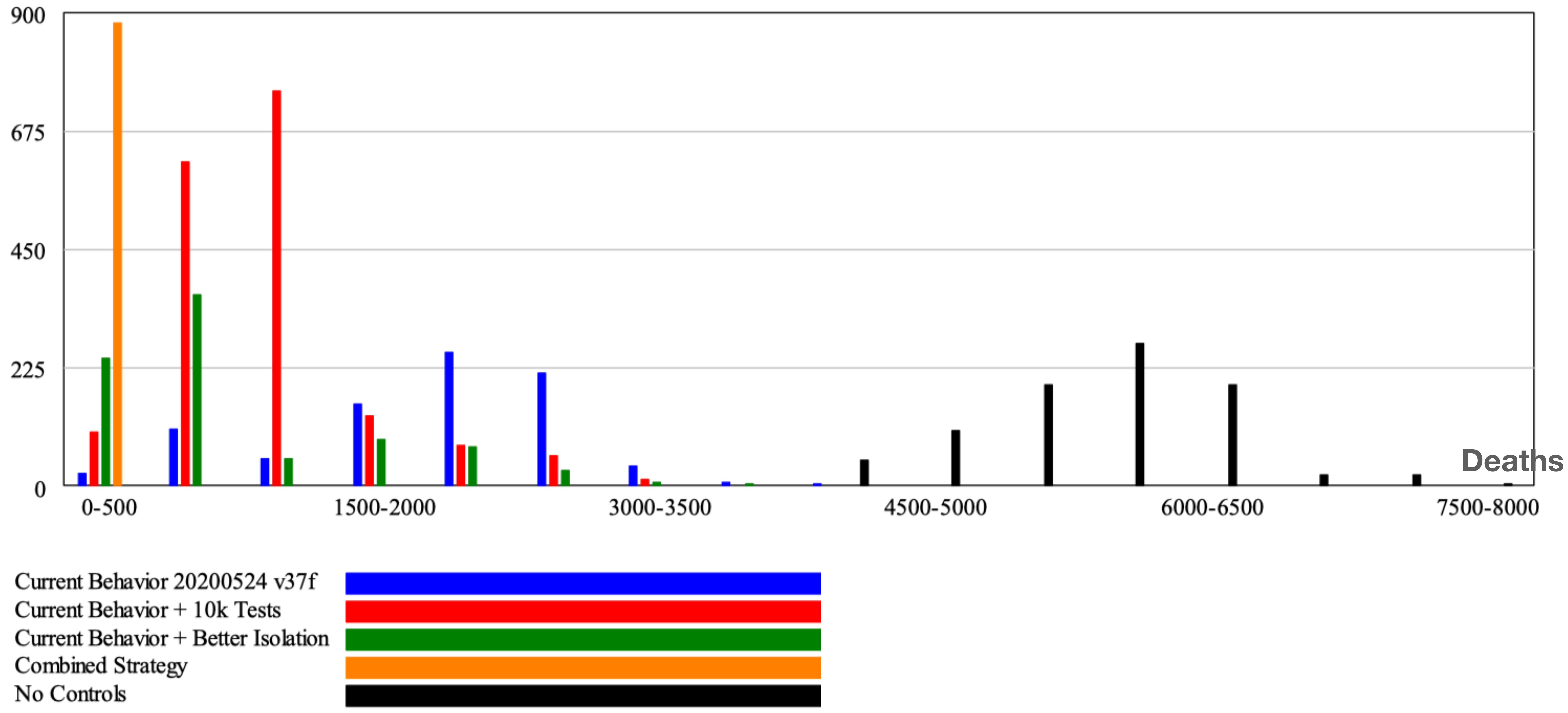
Final Result of Actionable Bayesian Workflow

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

Dynamics

Rehearsal

Data-based

Elements

Desired effect

arrange
elements of situation
to surreptitiously produce
desired effect

Five Cores of Simulation-based Calibration

Dynamics

Rehearsal

Data-based

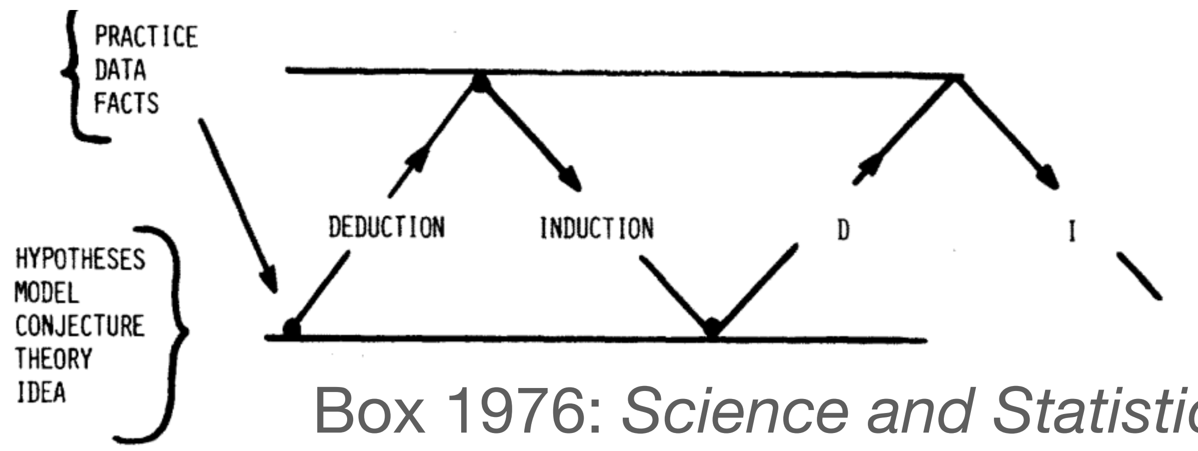
Elements

Desired effect

1. Orchestrate in Rehearsal

arrange
elements of situation
to surreptitiously produce
desired effect

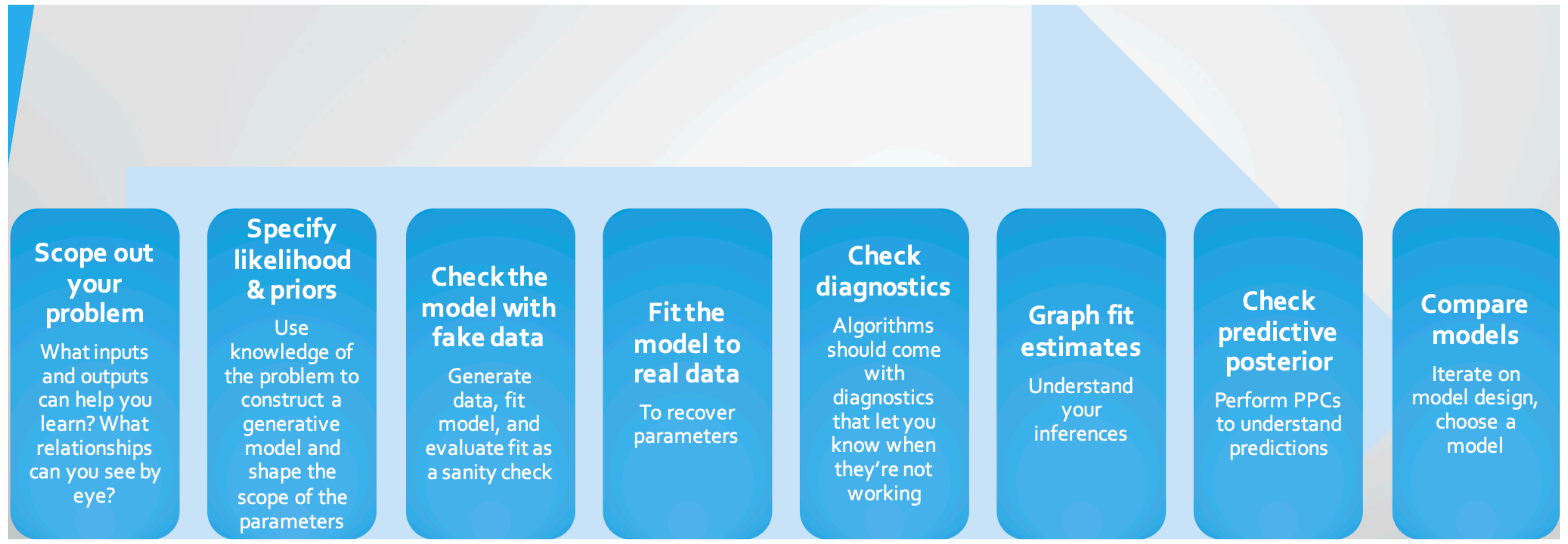
Bayesian Workflow Includes Rehearsal



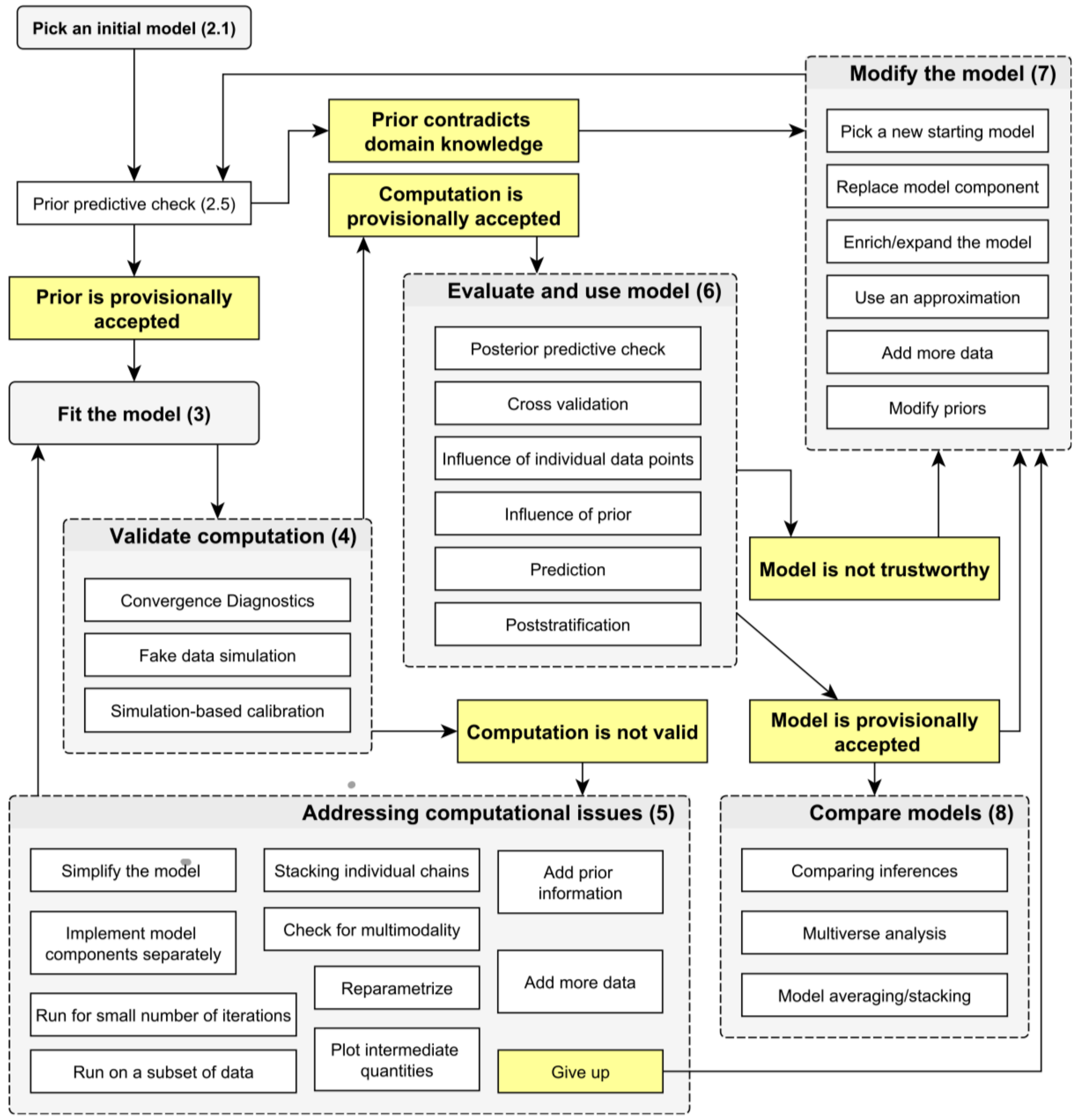
Box 1976: *Science and Statistics*

"theory and practice iteration with **feedback** loop"

Evolved to..



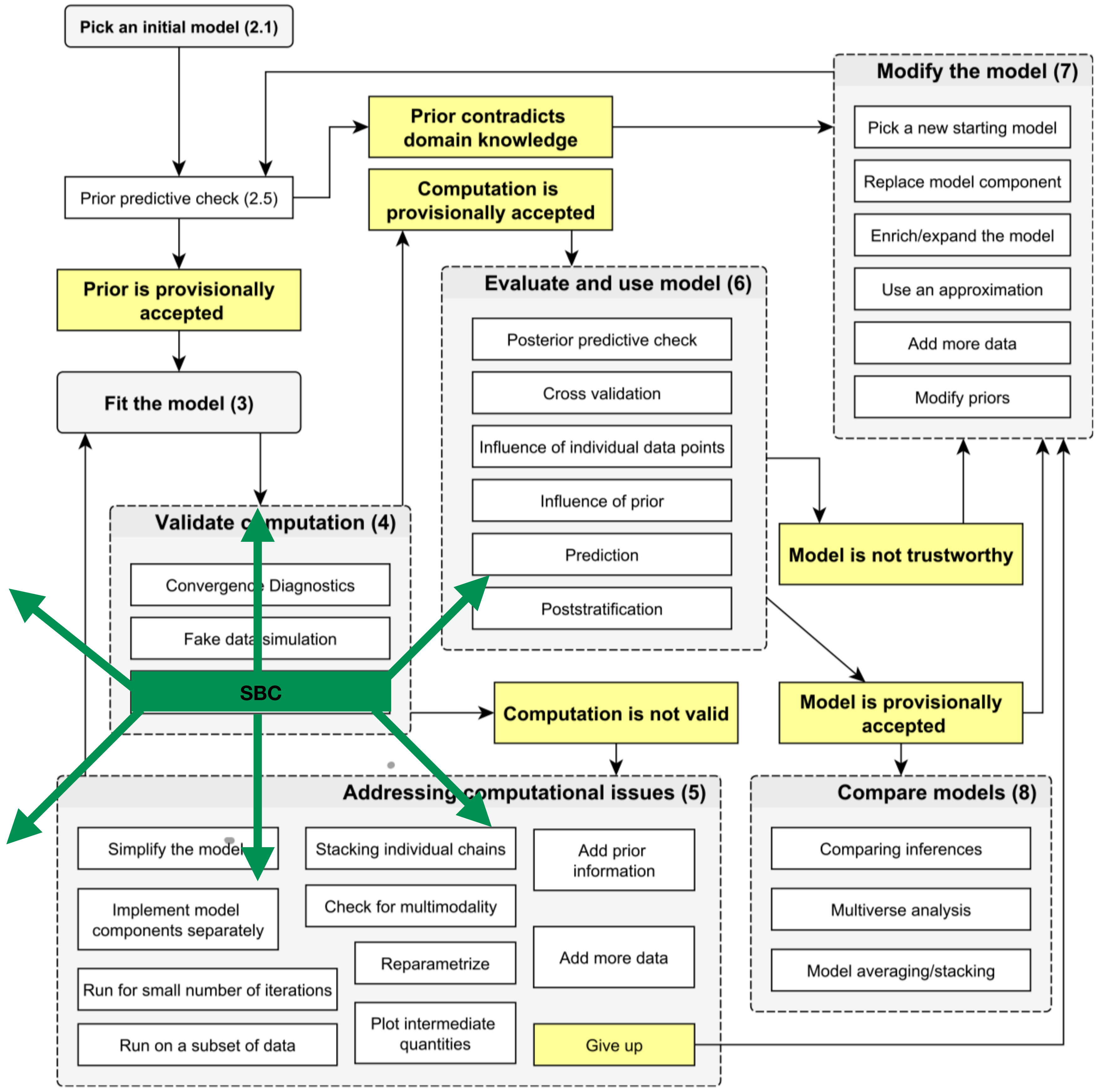
Talts 2018: *StanCon Intro*



Gelman et al 2020: *Bayesian workflow*

Simulation-based Calibration is Rehearsal

Not limited to computation validation!



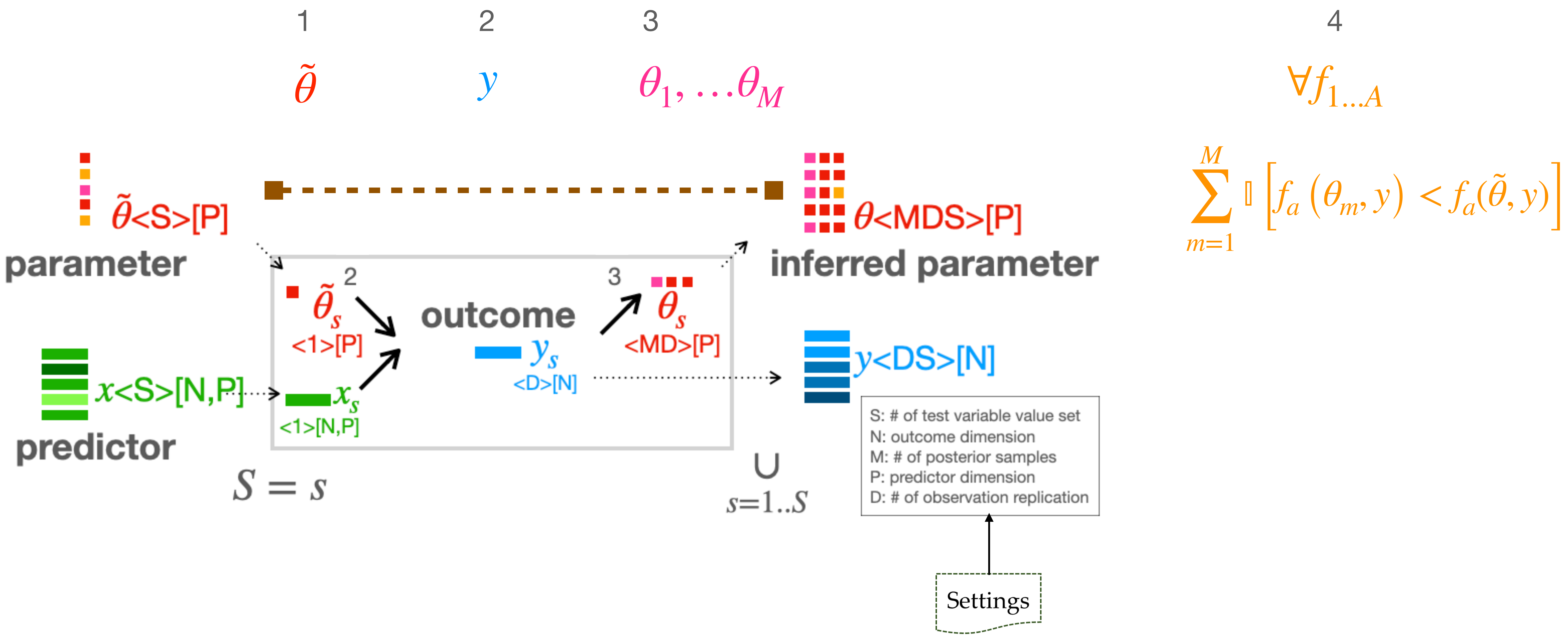
Bayesian time machine allowing Cause-Effect switching **experiment**

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

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

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

Simulation-based Calibration (SBC)



Procedural Elements of SBC

1. Draws2Data

$\tilde{\theta} \sim \pi_{\text{prior}}(\theta)$

$y \sim \pi_{\text{obs}}(y \mid \tilde{\theta})$

2. Data2Draws

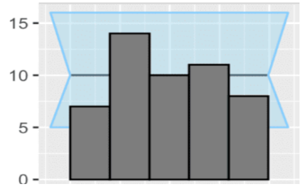
$\theta_1, \ldots \theta_M \sim \pi_{post}(\theta \mid y)$

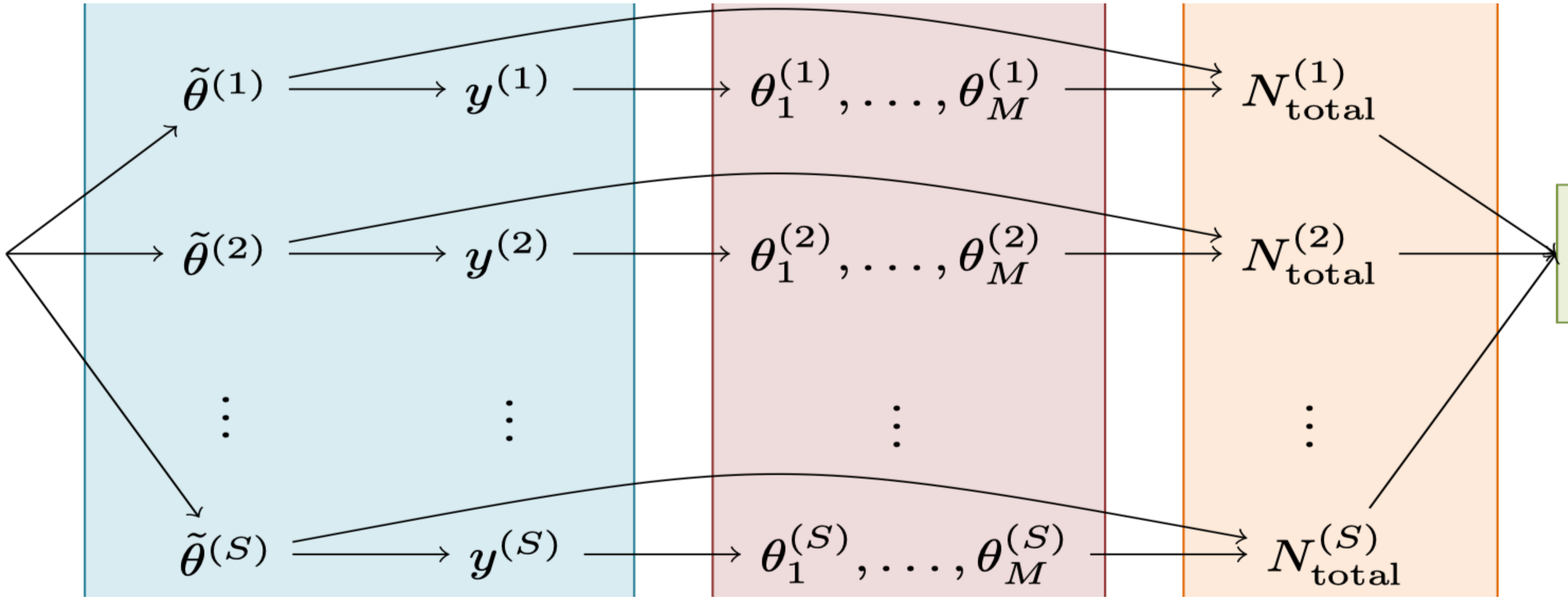
3. Construct Tests
Action/Adversarial

$\sum_{m=1}^{M} \mathbb{I}\left[f_a(\theta_m, y) < f_a(\tilde{\theta}, y)\right] \sim Unif(0, M)$

4. Diagnose
Stats./Graph

$\sim \chi^2$



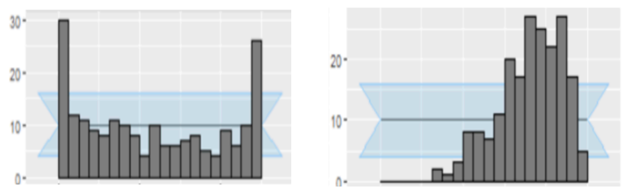


Generator

Probabilistic Program

Test Quantity

Informative deviation from uniformity



2. Elements to Orchestrate?

arrange
elements of situation
to surreptitiously produce
desired effect

Elements of Computational Bayes

Complicated, not consistent

Statistical model

Generator

$e \sim \pi$

$+ - \times /$

Computational model

Probabilistic program

0 1

Unobserved parameter

Assumed θ

Estimated θ θ

Data

Simulated y

Evolving to..

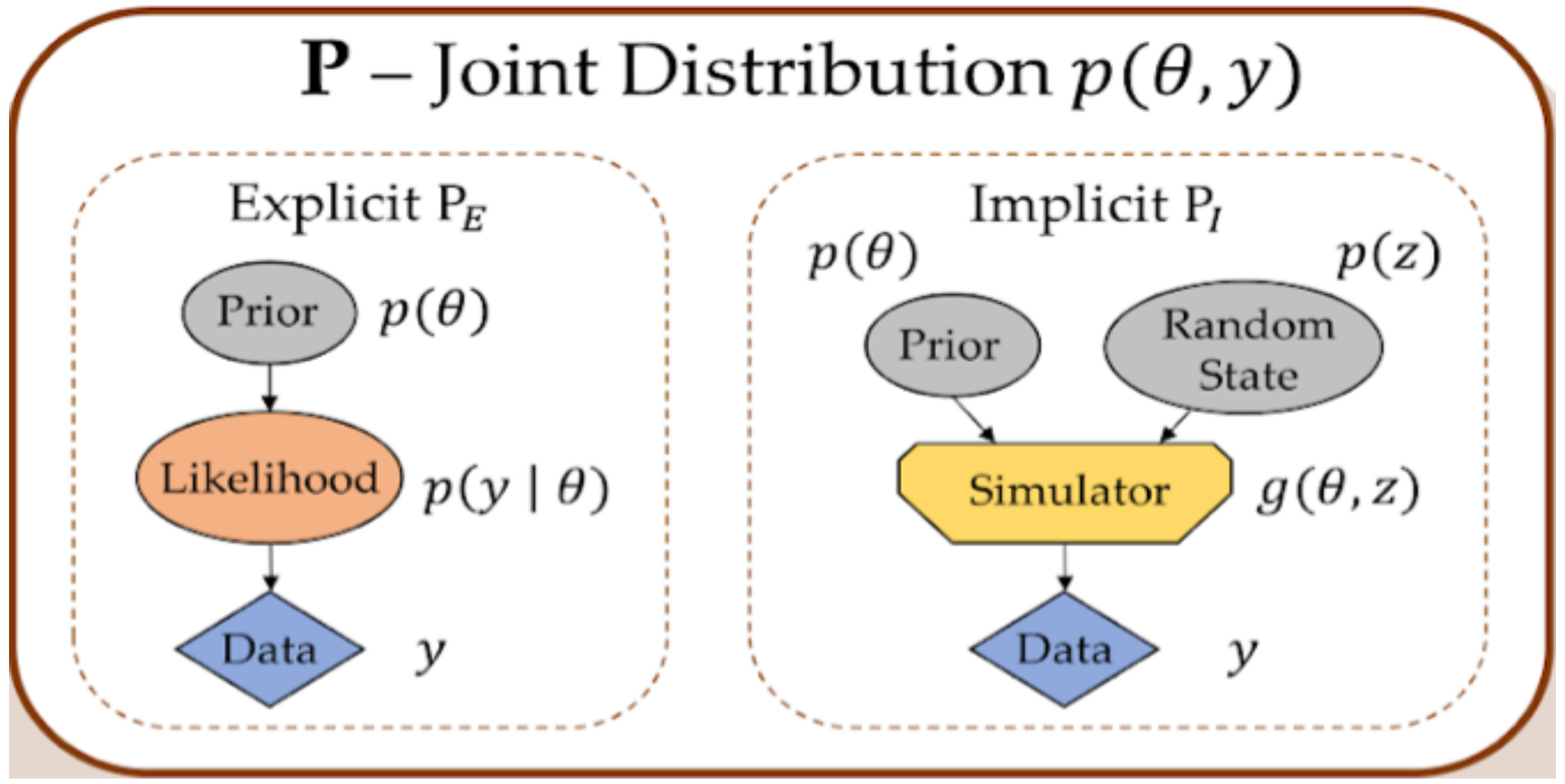
Unified taxonomy

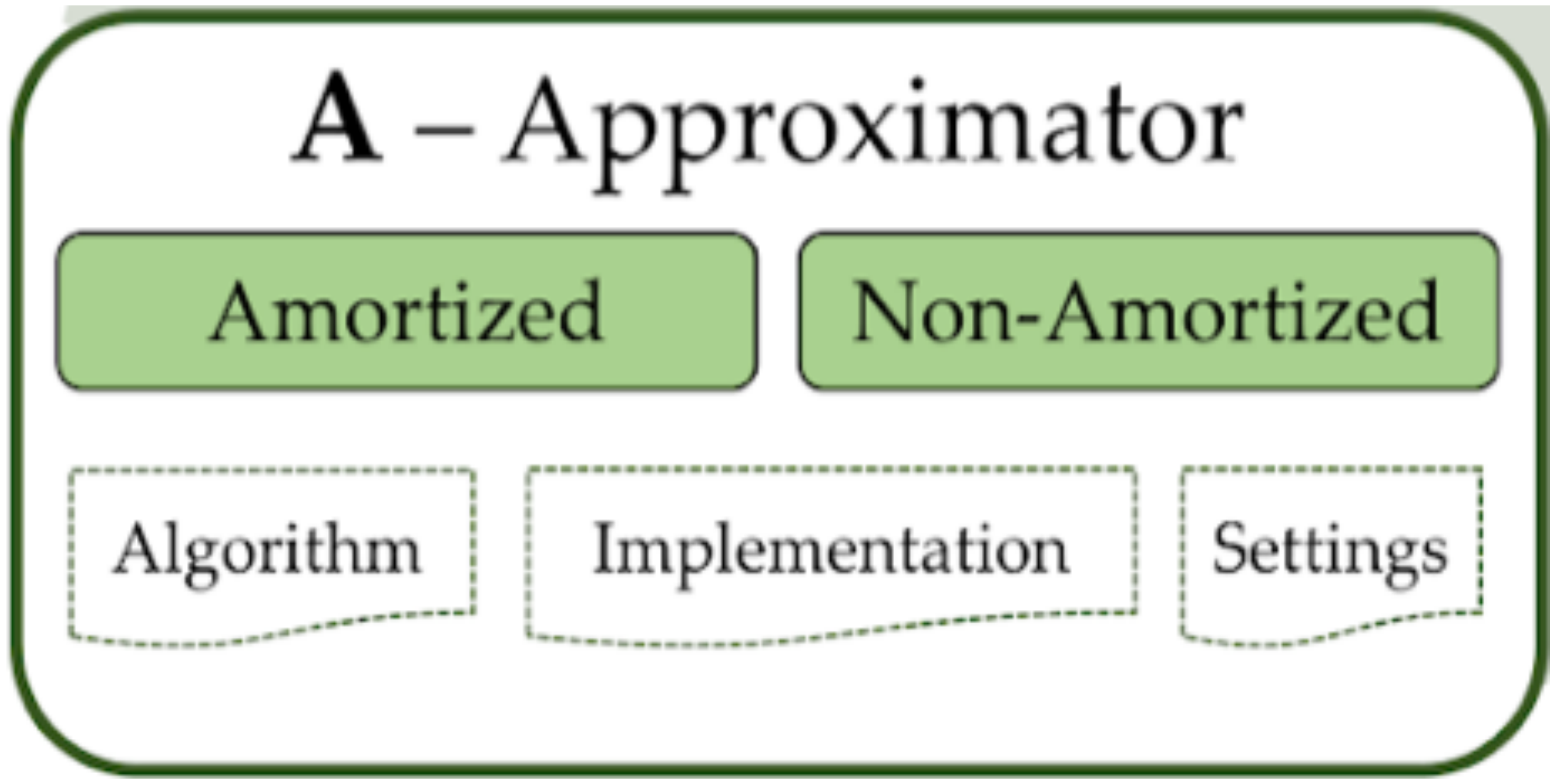
P: joint distribution $p(\theta, y)$

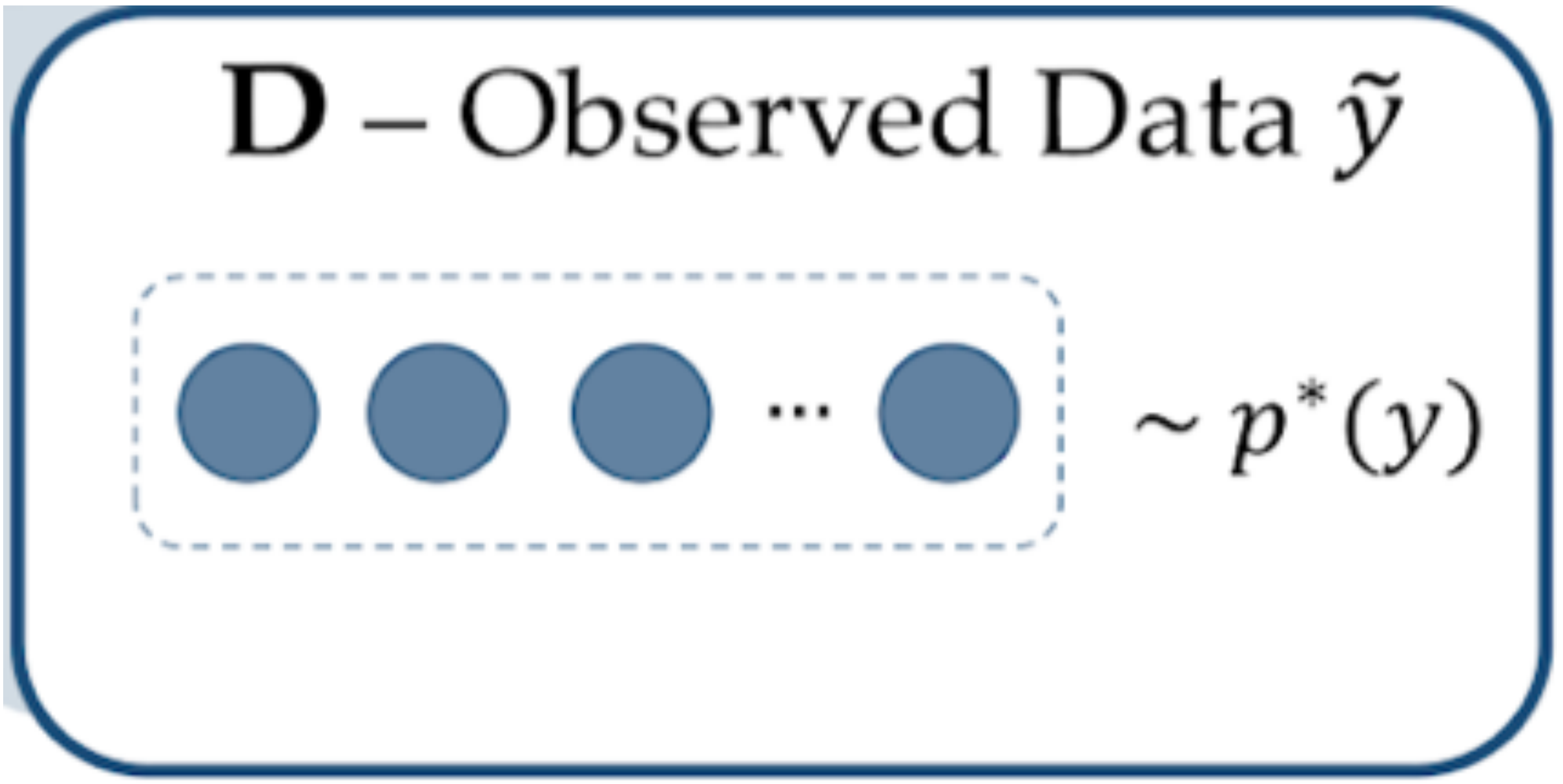
A: approximator $P_A(\theta | y)$

D: observed data $\tilde{y}$

Elements of Computational Bayes







SBC Orchestrates Computational Bayes Elements

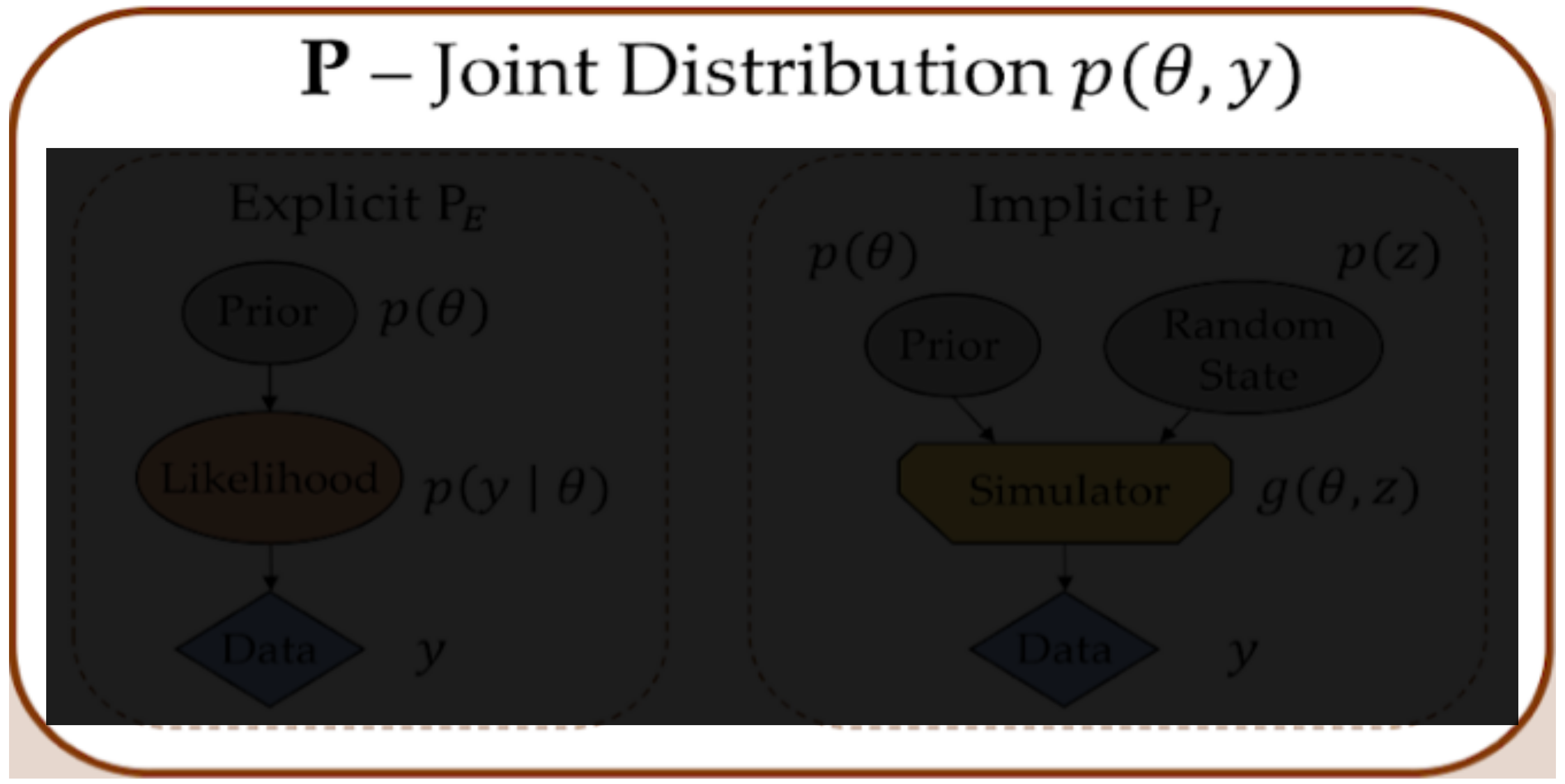
a. Identify P, A, D

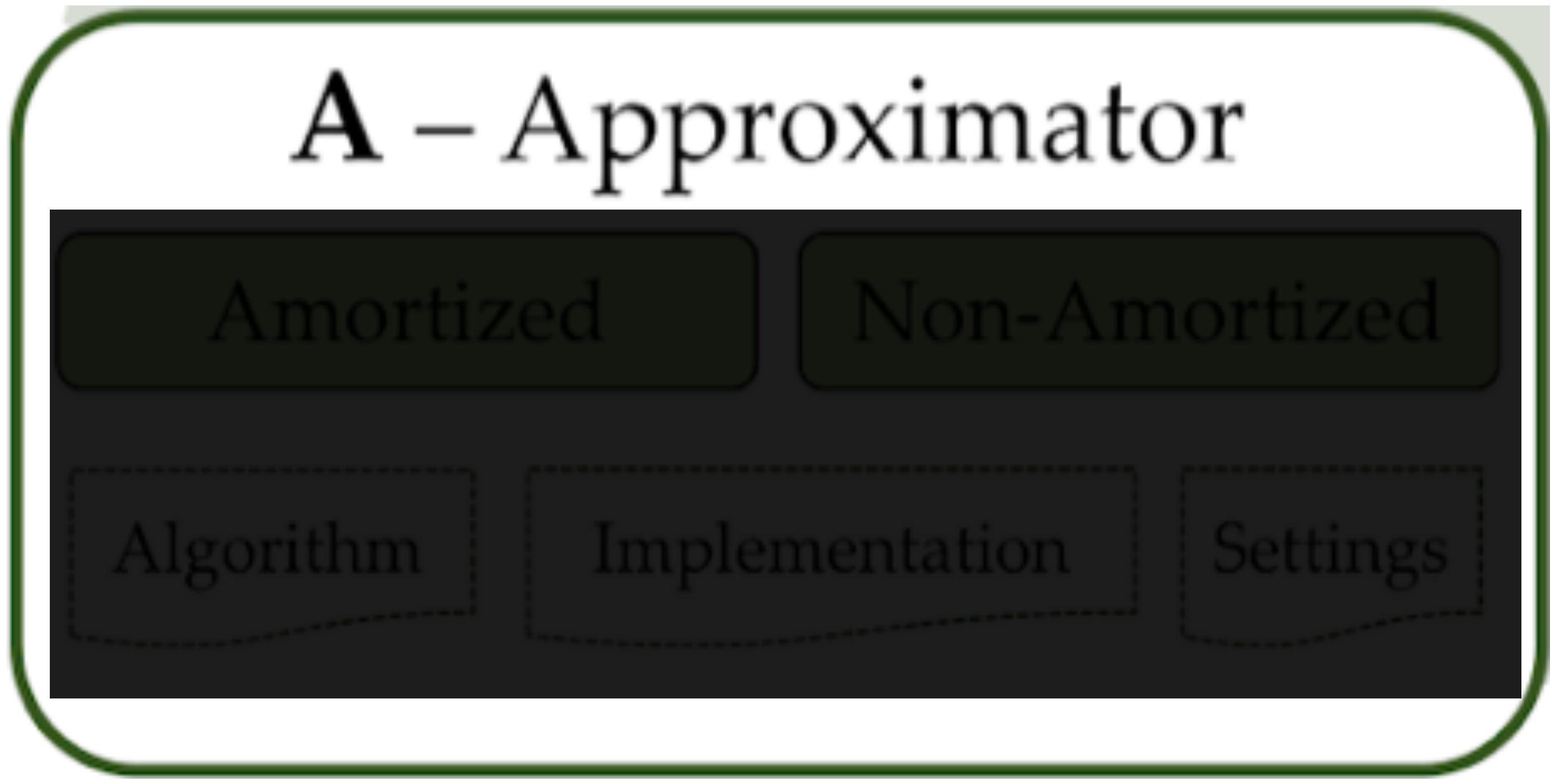
b. Orchestrate P, A, D

c. Identify P, A, D’s elements

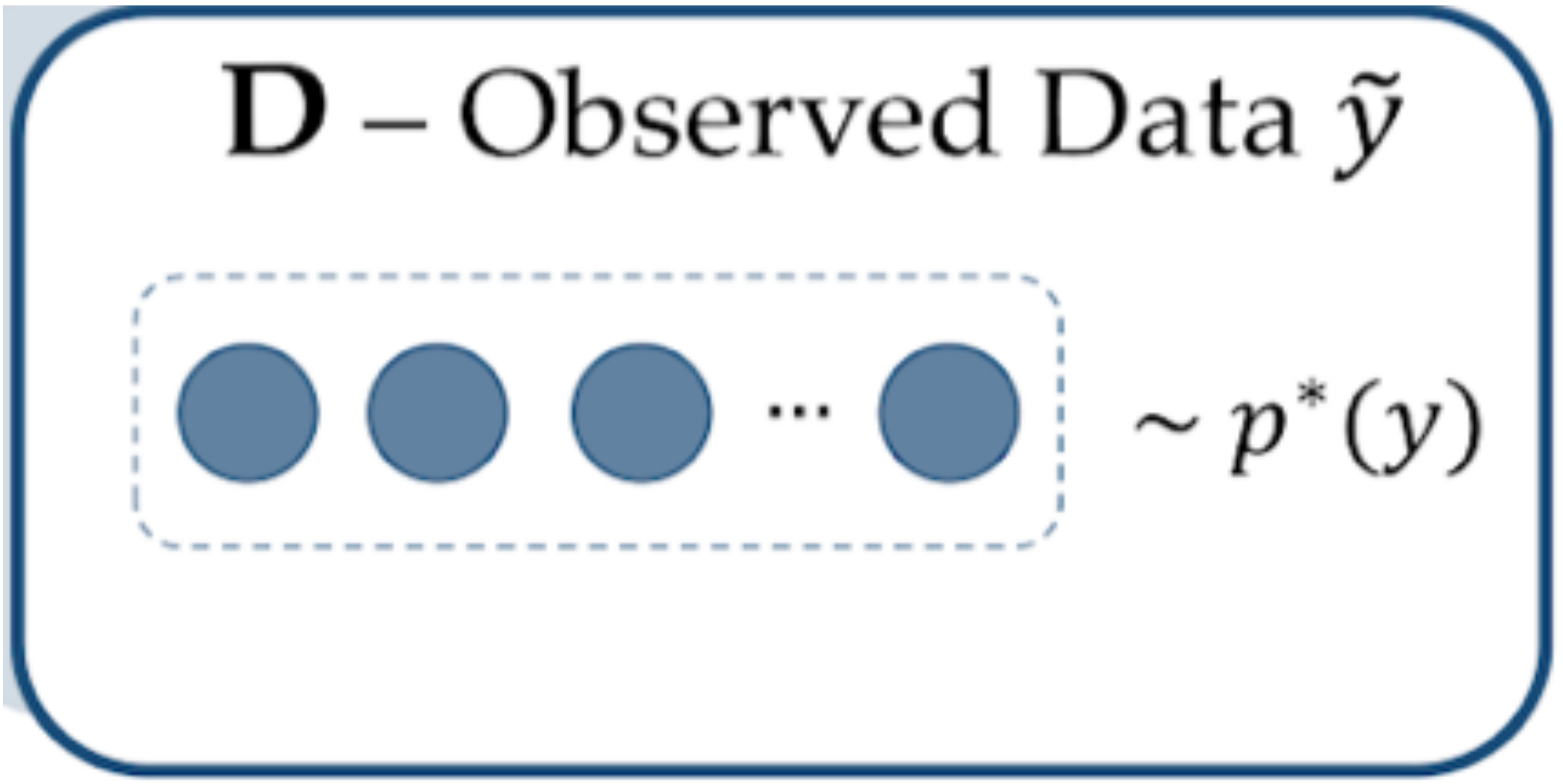
d. Orchestrate P, A, D’s elements

a. Identify P, A, D

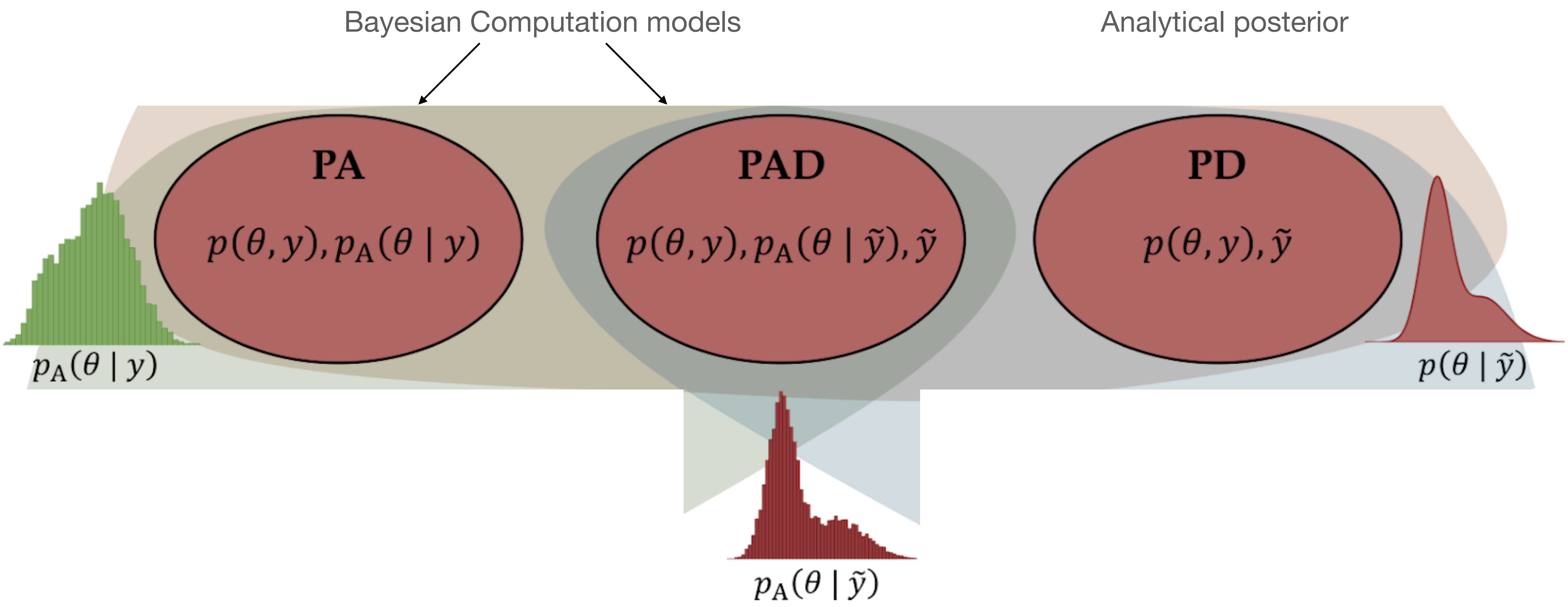




Optional

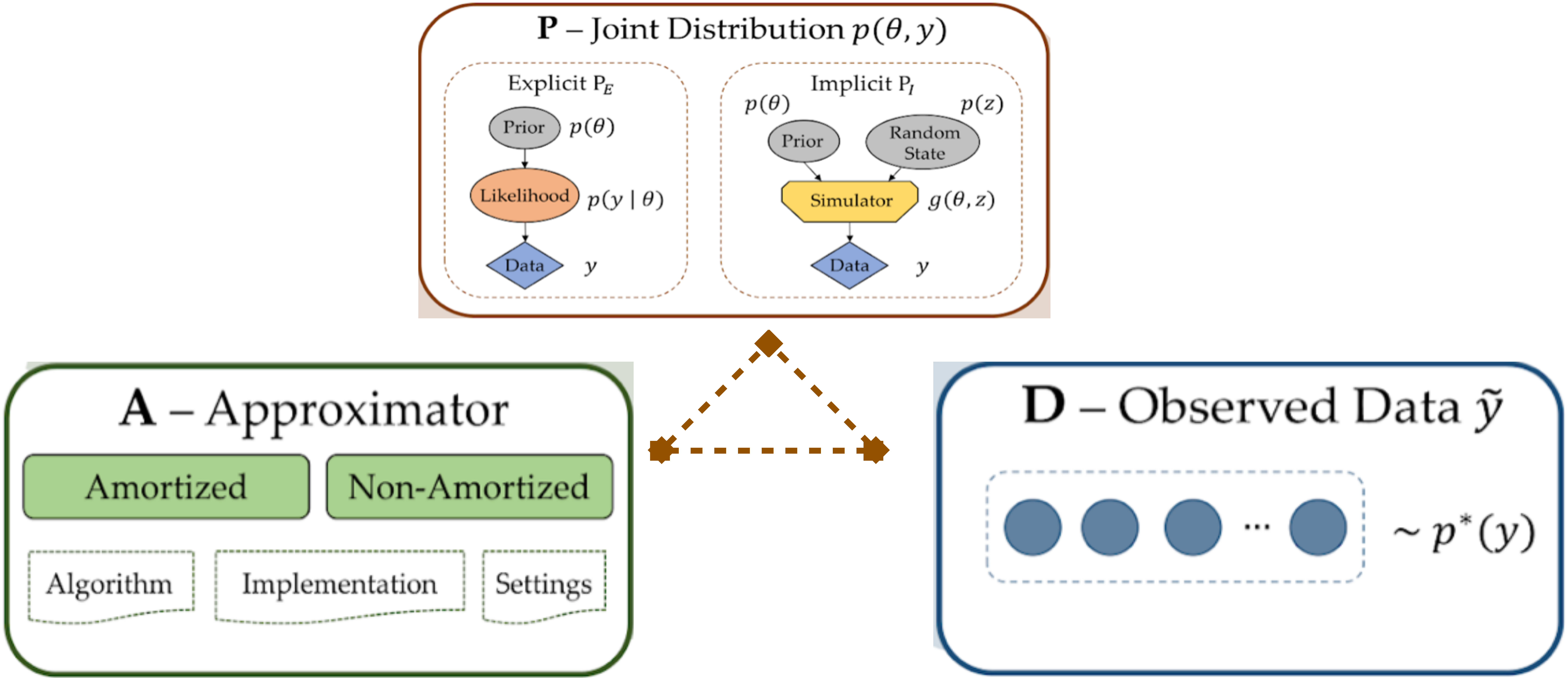


b. Orchestrate P, A, D as PA, PAD, PD



Plot from Bürkner et al 2022, *Some models are useful, but how do we know which ones? Towards a unified Bayesian model taxonomy*

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

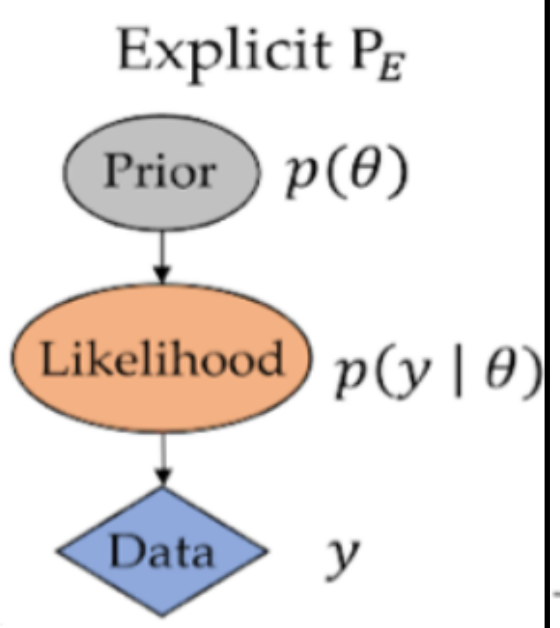


Settings e.g. are metric, stepsize, S, N, M

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

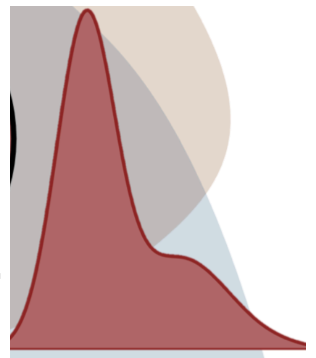
Implementation — Stan file:

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);
}
```

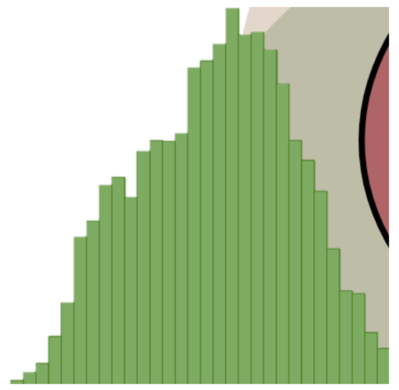
Stan file forms

Analytic posterior



$p(\theta | y)$

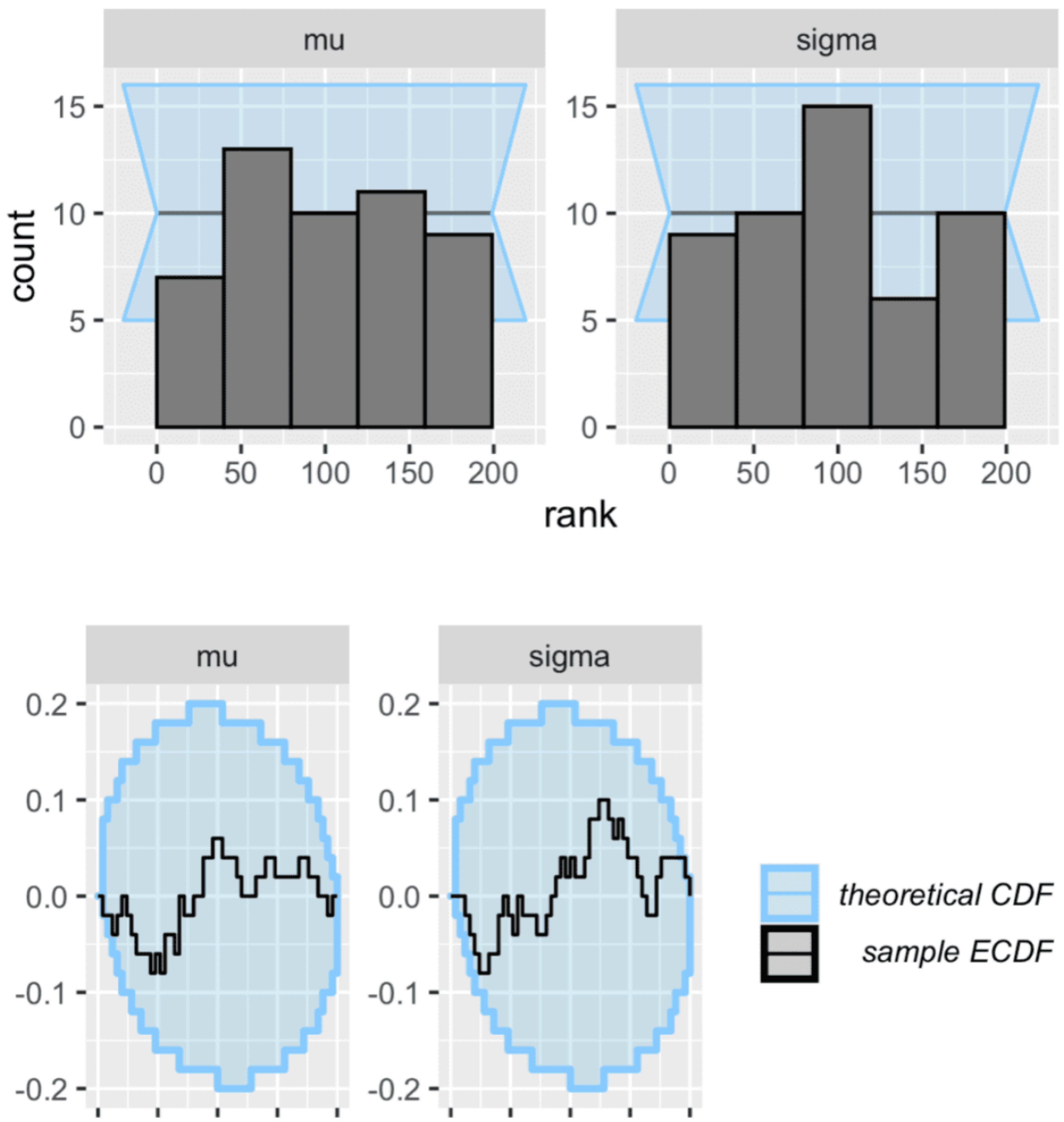
MCMC transforms to



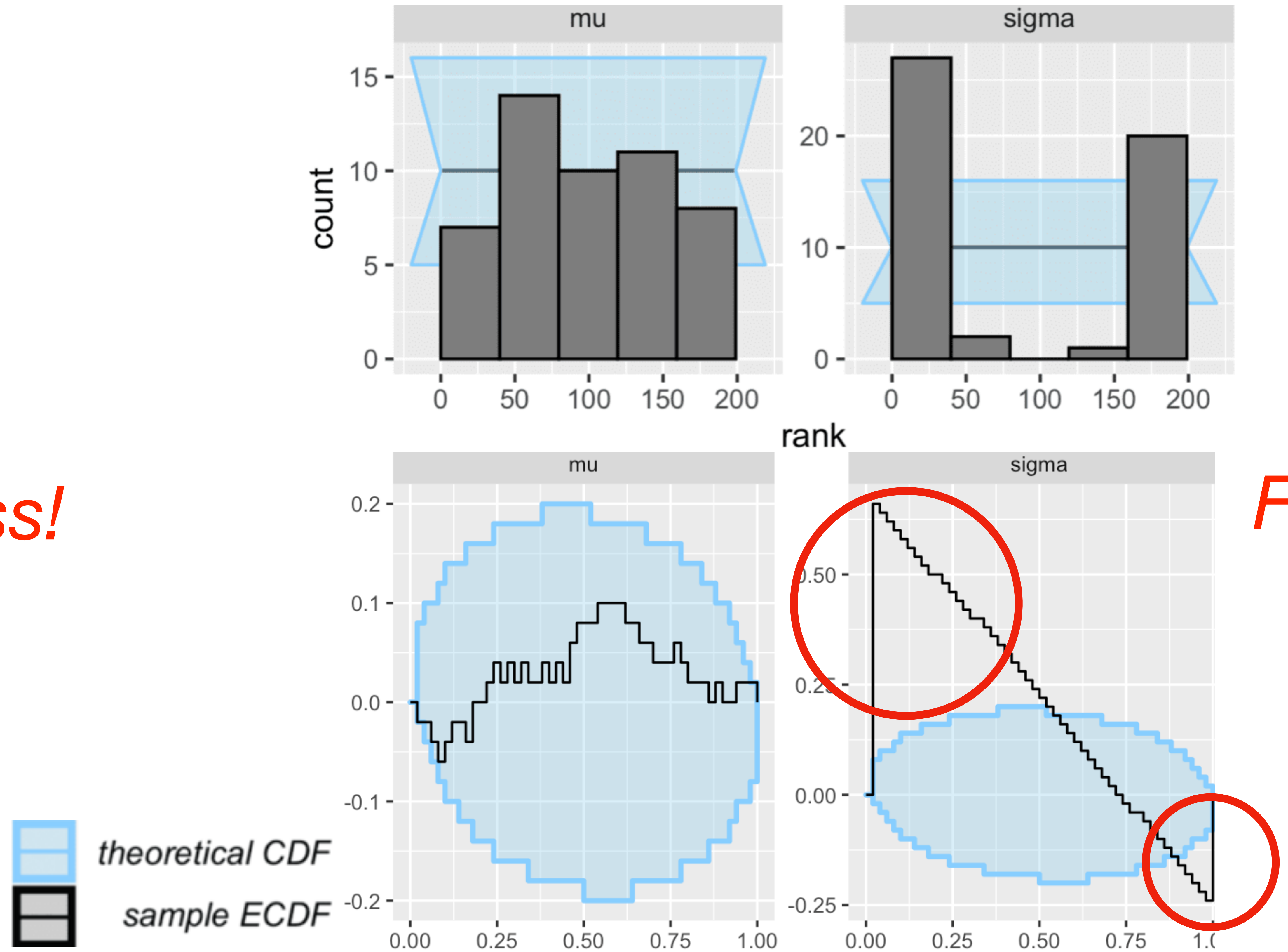
$p_A(\theta | y)$

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

b. Check P, A is well-orchestrated



Pass!



Fail!

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

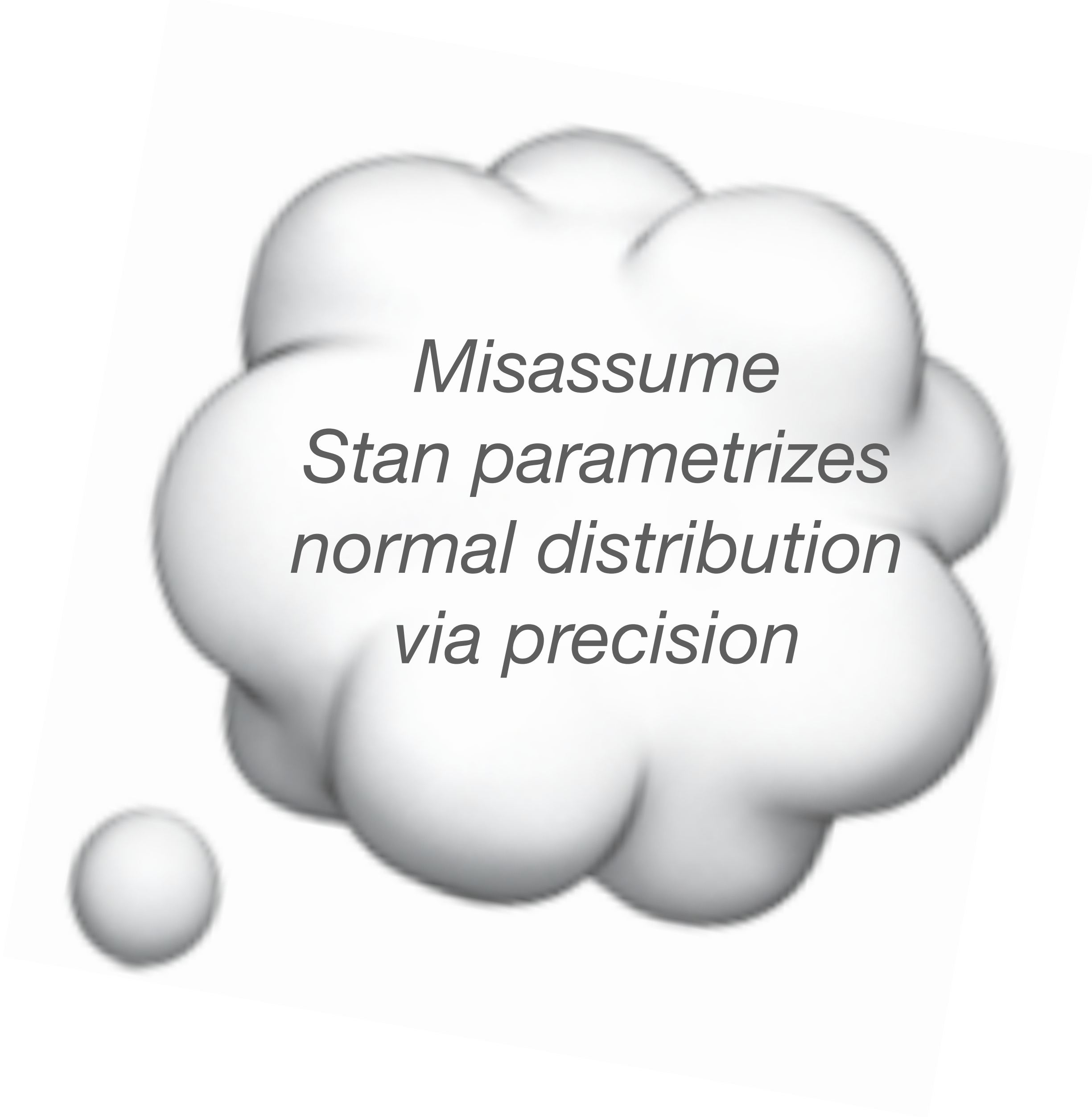
b. Cacophony

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

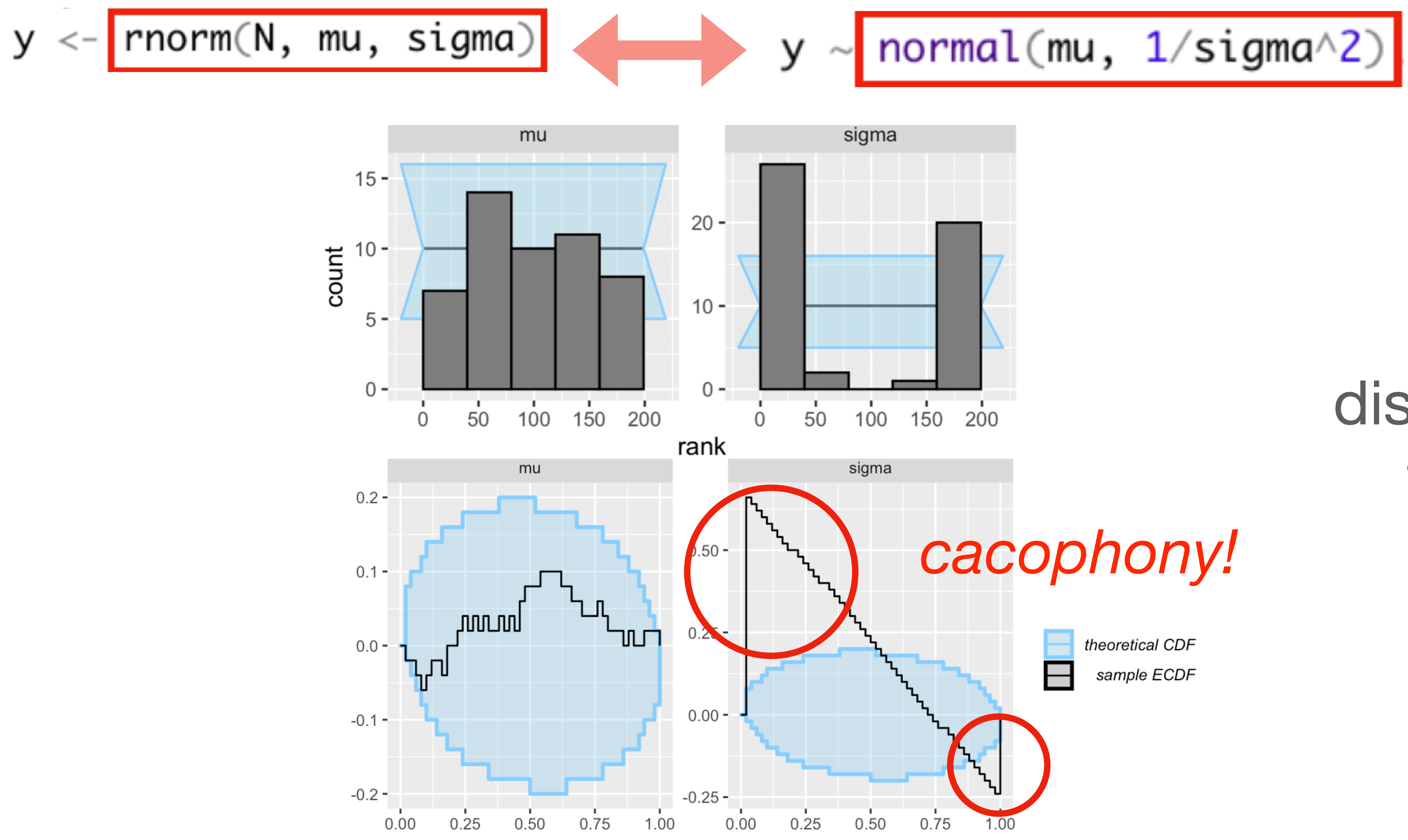
```
data {
  int<lower=0> N;
  vector[N] y;
}

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

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



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

b. Cacophony

Mismatch of P, A detected via SBC plots



Feedback to
discover implementation bug
from wrong assumption

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

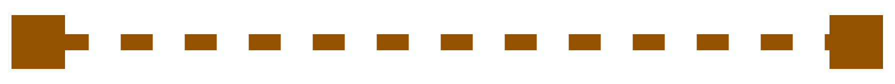
Additional use cases and advanced topics

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

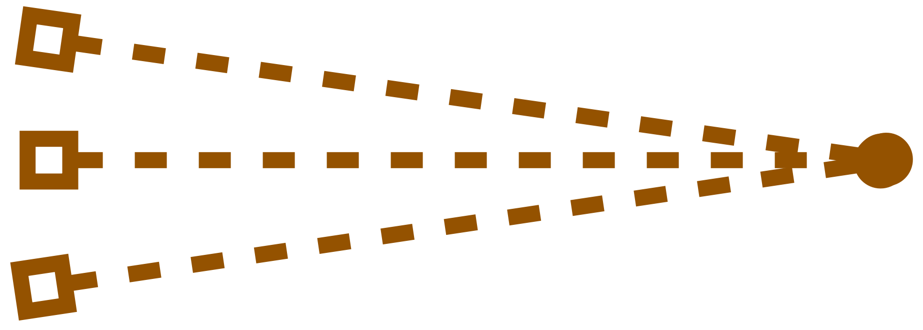
	P	**A**	
	$P(\theta, y)$	$P_A(\theta \mid y)$	

1. my P and A



e.g. debug model implementation

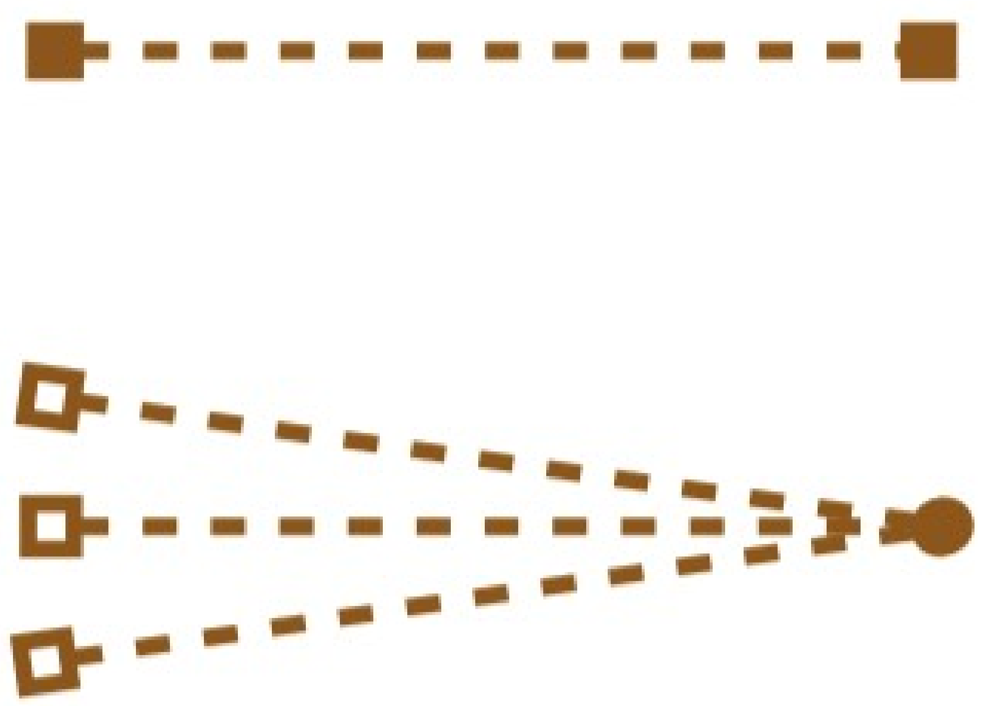
2. my A with canonical Ps



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

Hidden Known Problem of SBC

	P	**A**
	$P(\theta, y)$	$P_A(\theta \mid y)$
1. my P and A		
2. my A with canonical Ps		



Conditional on assumption

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

SBC Orchestrates Computational Bayes Elements

a. Identify P, A, D

b. Orchestrate P, A, D

c. Identify P, A, D’s elements

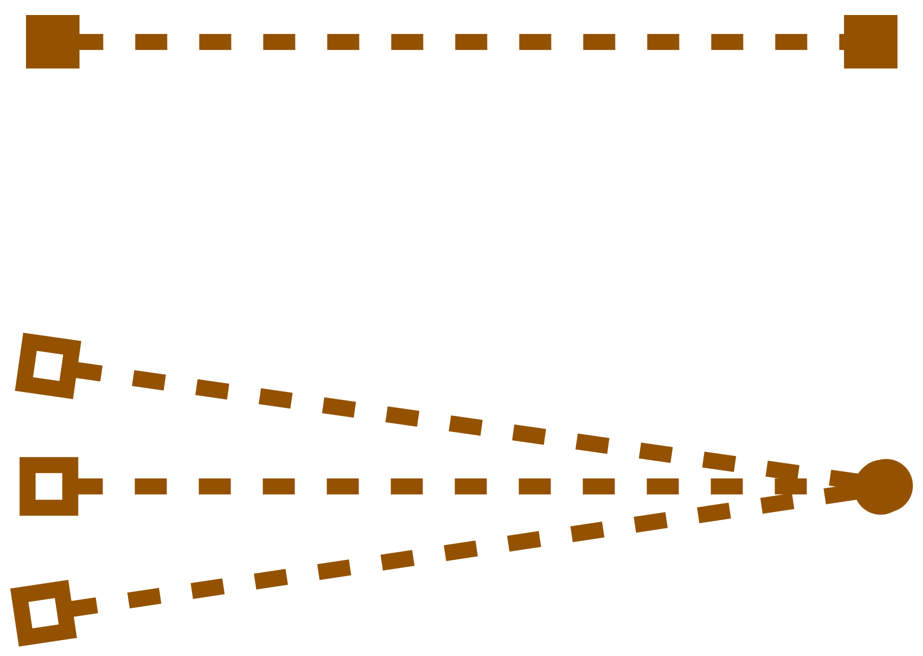
d. Orchestrate P, A, D’s elements

Hidden Known Problem of SBC

P $P(\theta, y)$

A $P_A(\theta \mid y)$

1. my P and A
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

Conditional on assumption

e.g. assumed parameter

+

assumed default

A (approximator)'s Settings

= 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 adher
  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 Θ_t $\theta\theta$ passes simulation-based calibration

b. Update Θ_t to Θ_{t+1} with real data

c. Prescribe policy based on $E[f(\Theta_{t+1},\theta,\theta,\tilde{y})|\theta,\theta,\theta]_{\theta=1..P}$

-> Example next pg.

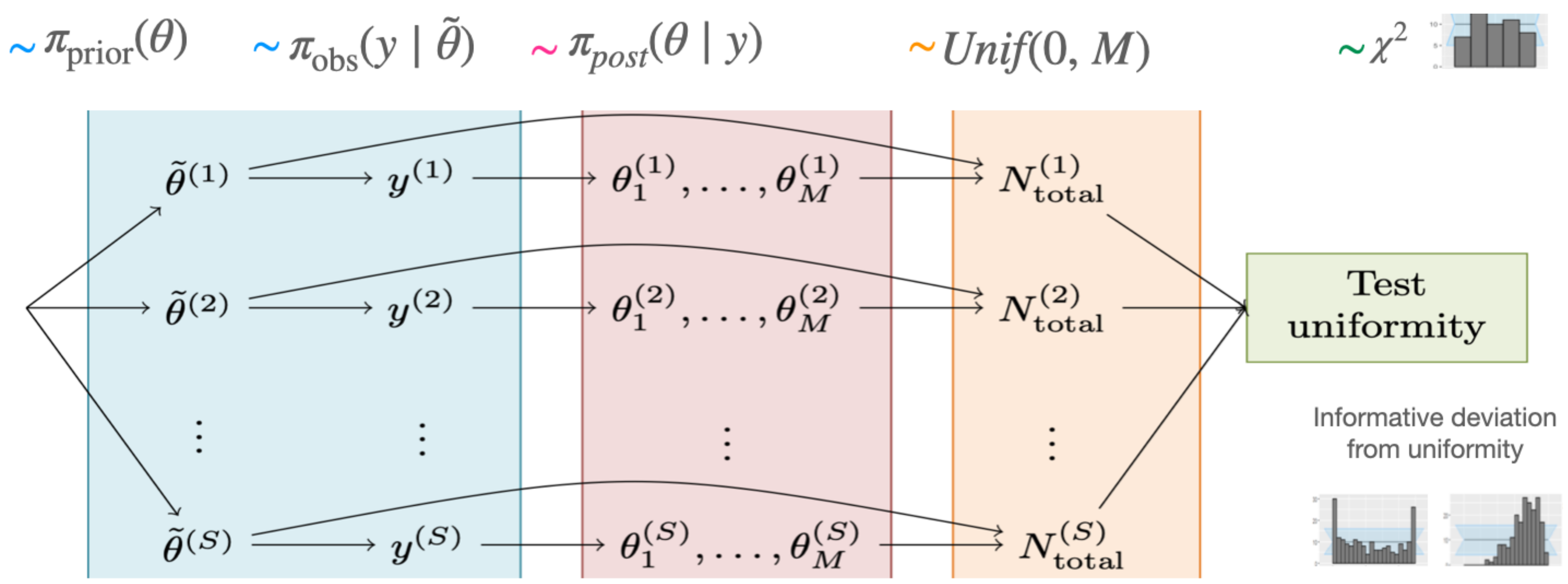
a: generate then estimate, c: generate only

3. Orchestrate with Dynamics, Tempo!

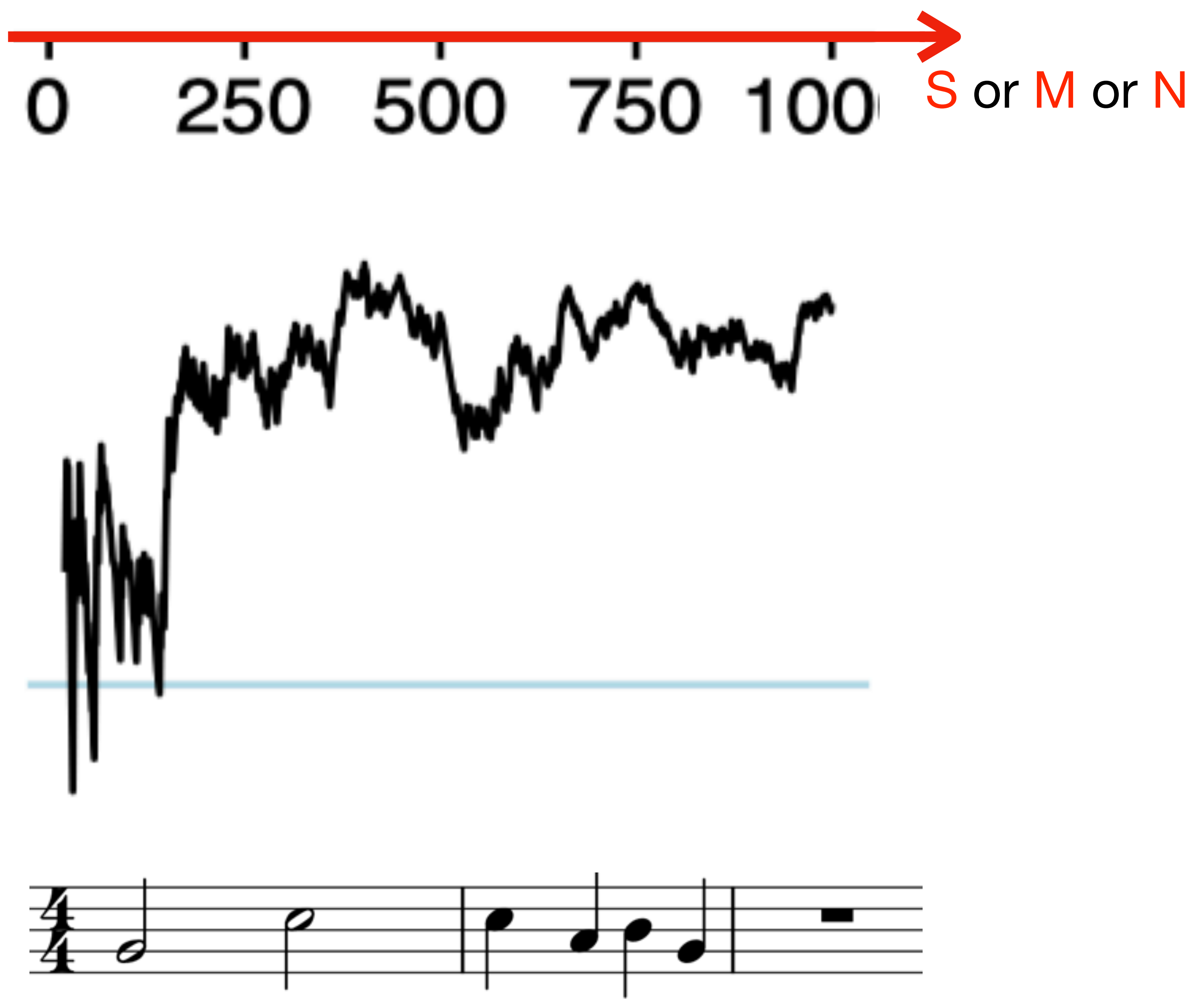
Grave = very slow, solemn
Largo = slow and broad
Adagio = slow
Andante = walking pace, medium speed
Moderato = moderate speed
Allegro = happy, upbeat, fast
Vivace = lively, fast
Presto = very fast

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

Don't wait till the final verdict!



Can evolve to

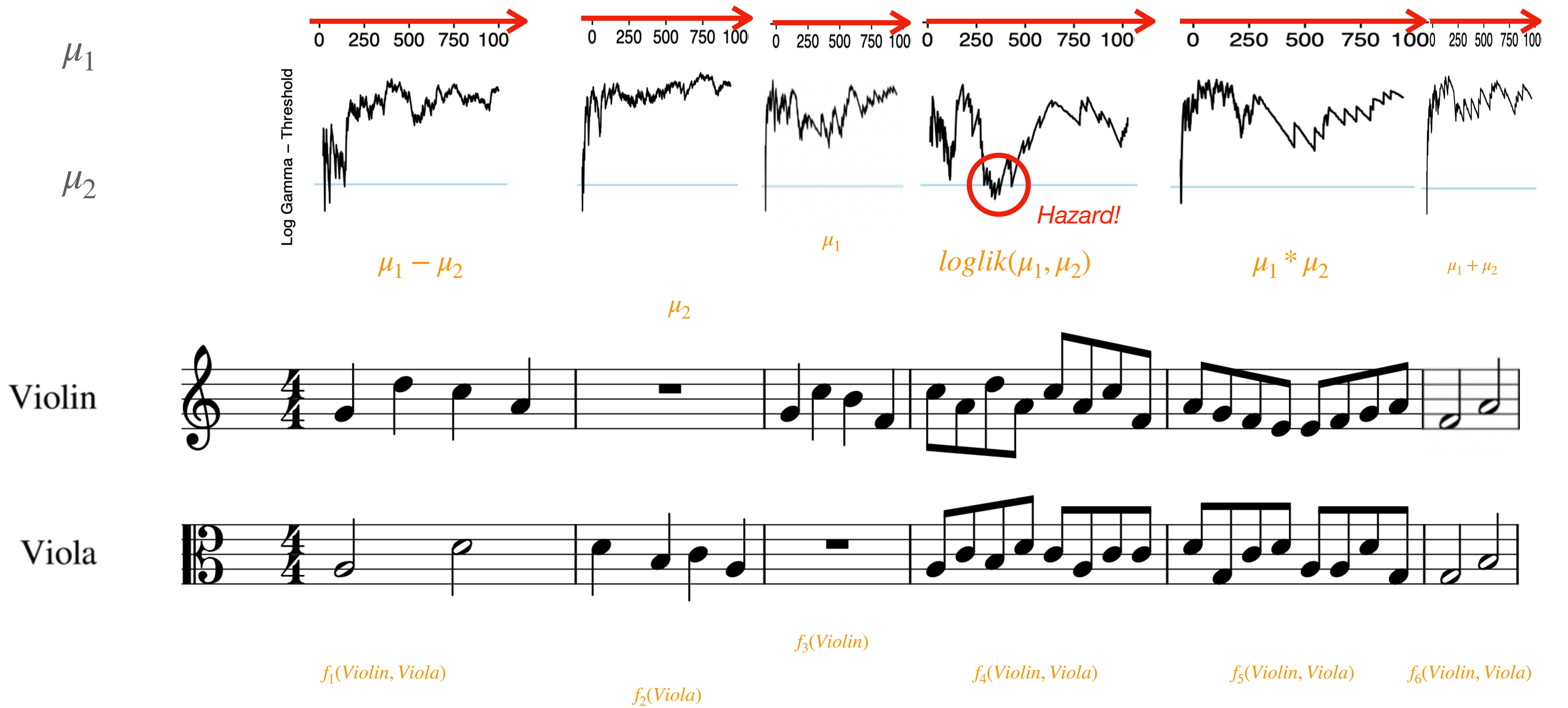


Plots from coauthoring paper: Modrak et al, *Good choice of test quantities substantially improves the sensitivity of simulation-based calibration for validating Bayesian computation*

4. Orchestrate for Desire

arrange
elements of situation
to surreptitiously produce
desired effect

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



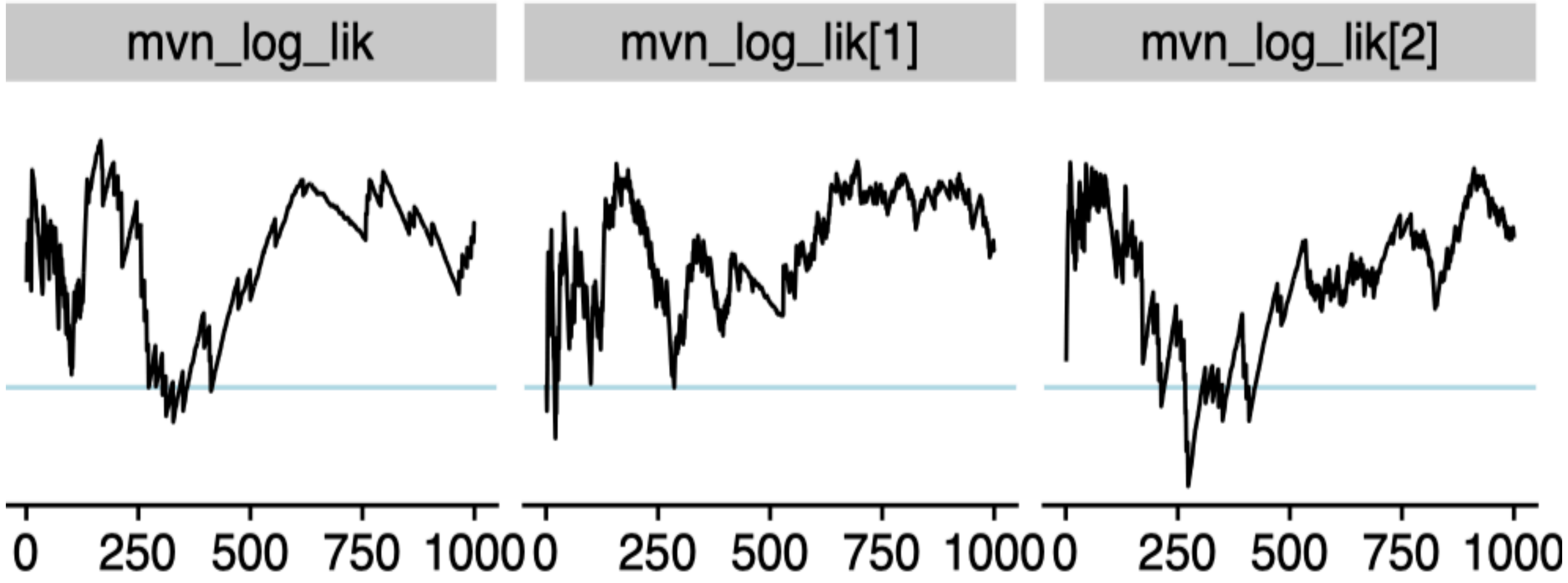
Plots from coauthoring paper: Modrak et al, *Good choice of test quantities substantially improves the sensitivity of simulation-based calibration for validating Bayesian computation*

5. Orchestrate based on Data

arrange
elements of situation
to surreptitiously produce
desired effect

Data-based Test Quantities

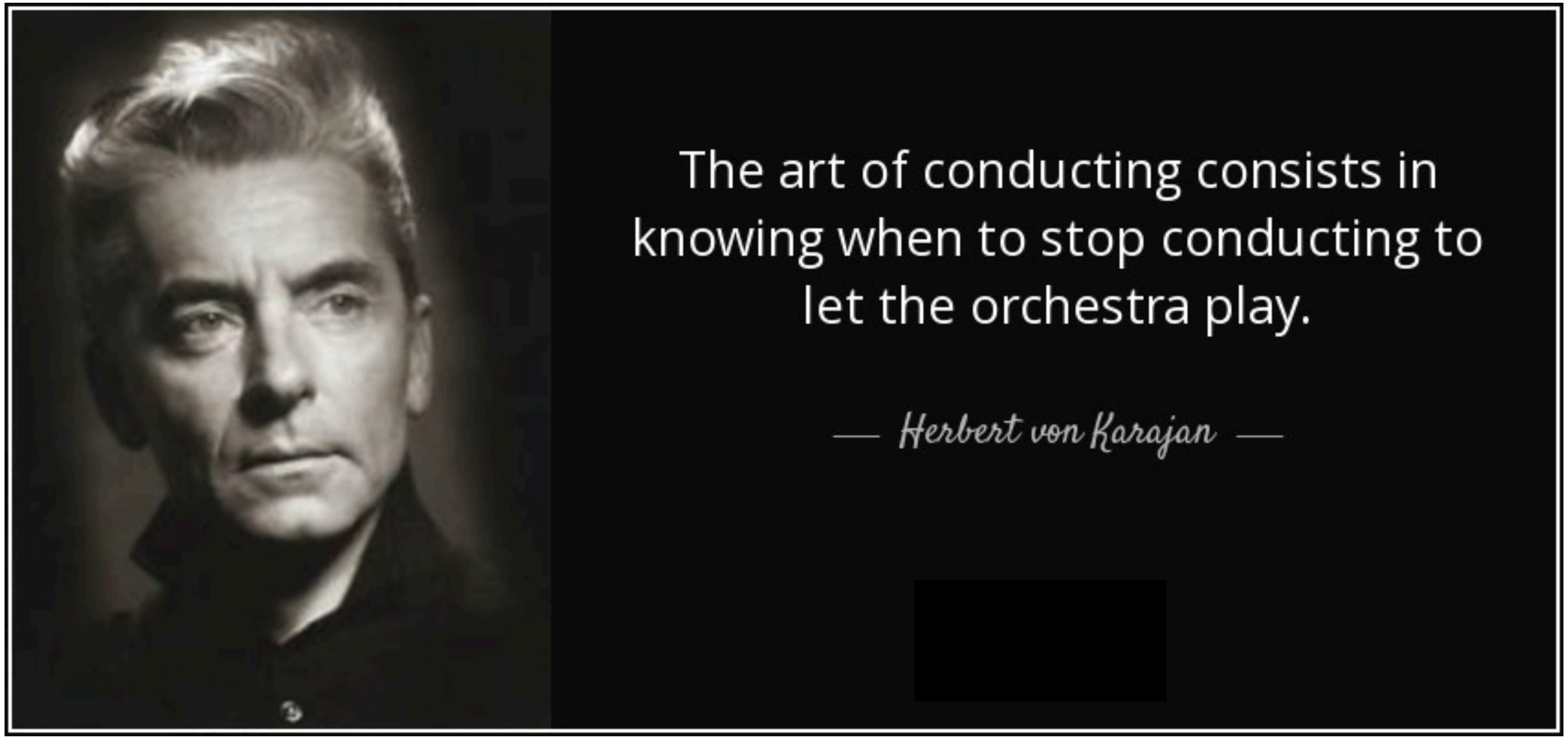
$$\begin{aligned} \mu_1, \mu_2 &\sim MVN(0, \boldsymbol{\Sigma}) \\ \mathbf{y}_1, \ldots, \mathbf{y}_N &\sim MVN(\mu, \boldsymbol{\Sigma}) \\ \boldsymbol{\Sigma} &= \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \end{aligned}$$



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

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



Colors in SBC

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