Automated Experiment Design for Data-Efficient Verification of Parametric Markov Decision Processes

E. Polgreen[1], V. Wijesuriya[1], S. Haesaert[2], A. Abate[1]

[1]Department of Computer Science, University of Oxford

[2]Department of Electrical Engineering, TU Eindhoven

QEST 2017

Introduction

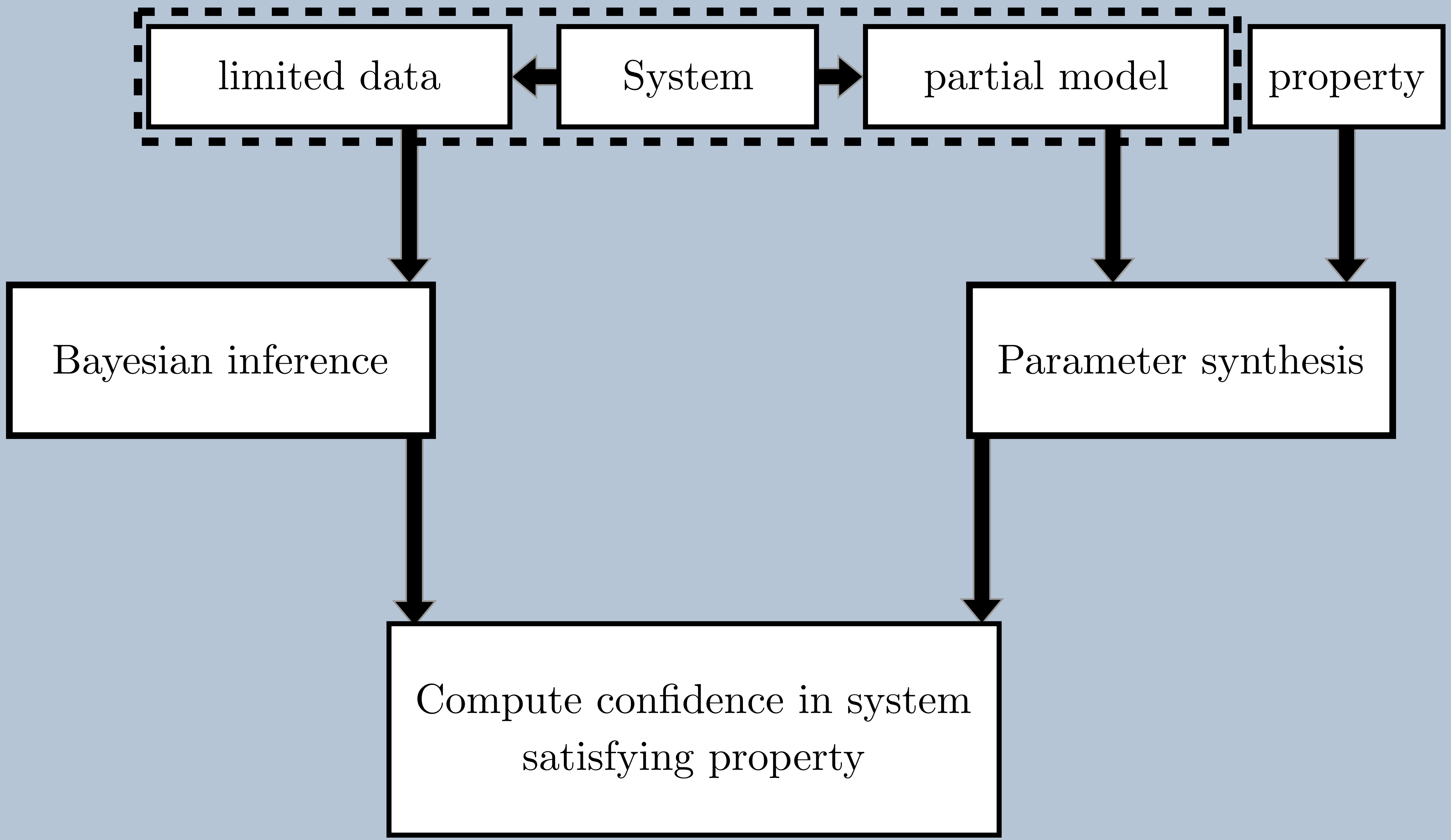
- Verifying real systems is hard; full models are difficult to obtain

- Data-based verification requires a lot of data

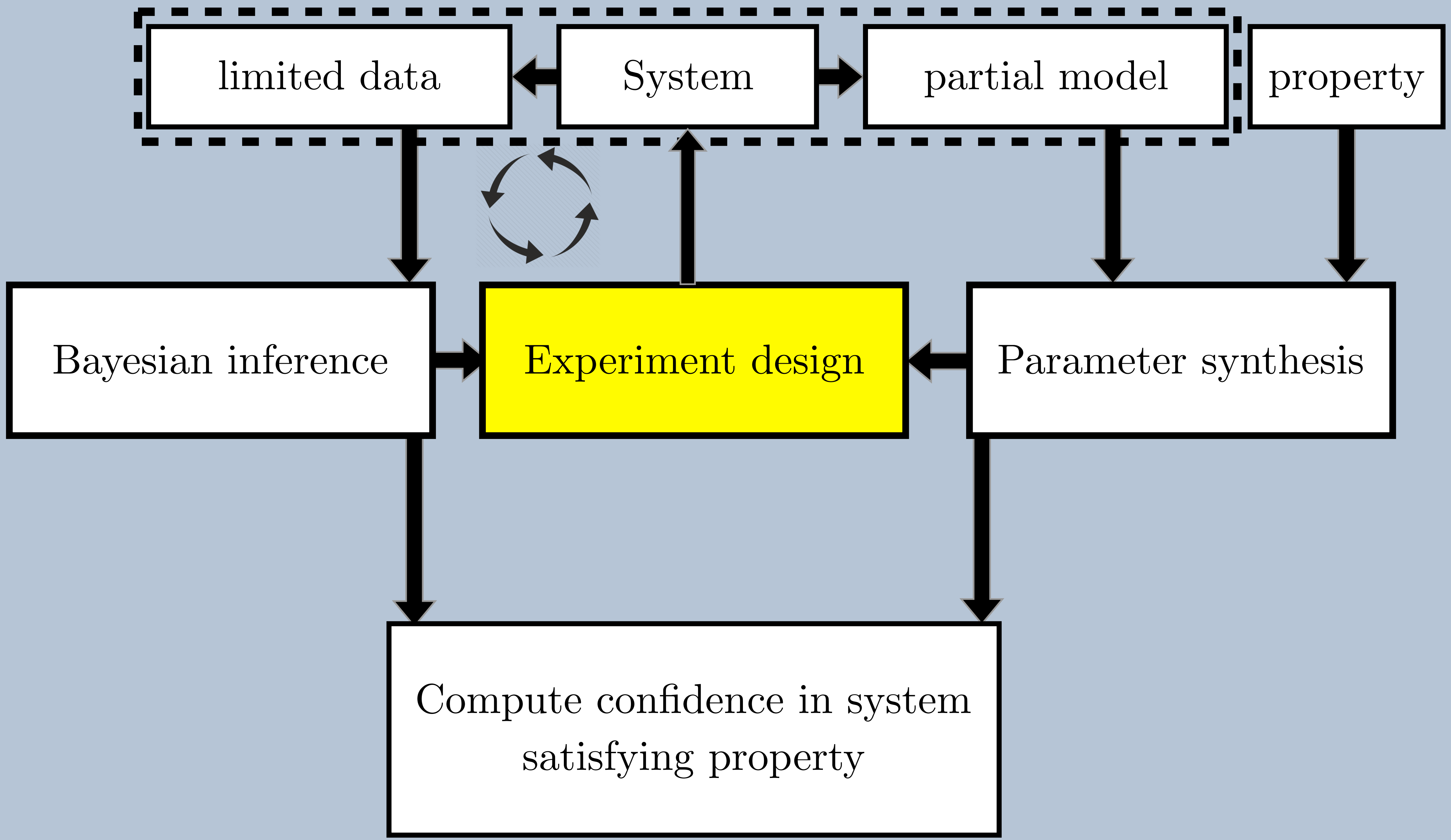
- 2016: Bayesian verification framework for Markov chains

- Now: Markov Decision Processes, using automated experiment design

Overview - Bayesian verification

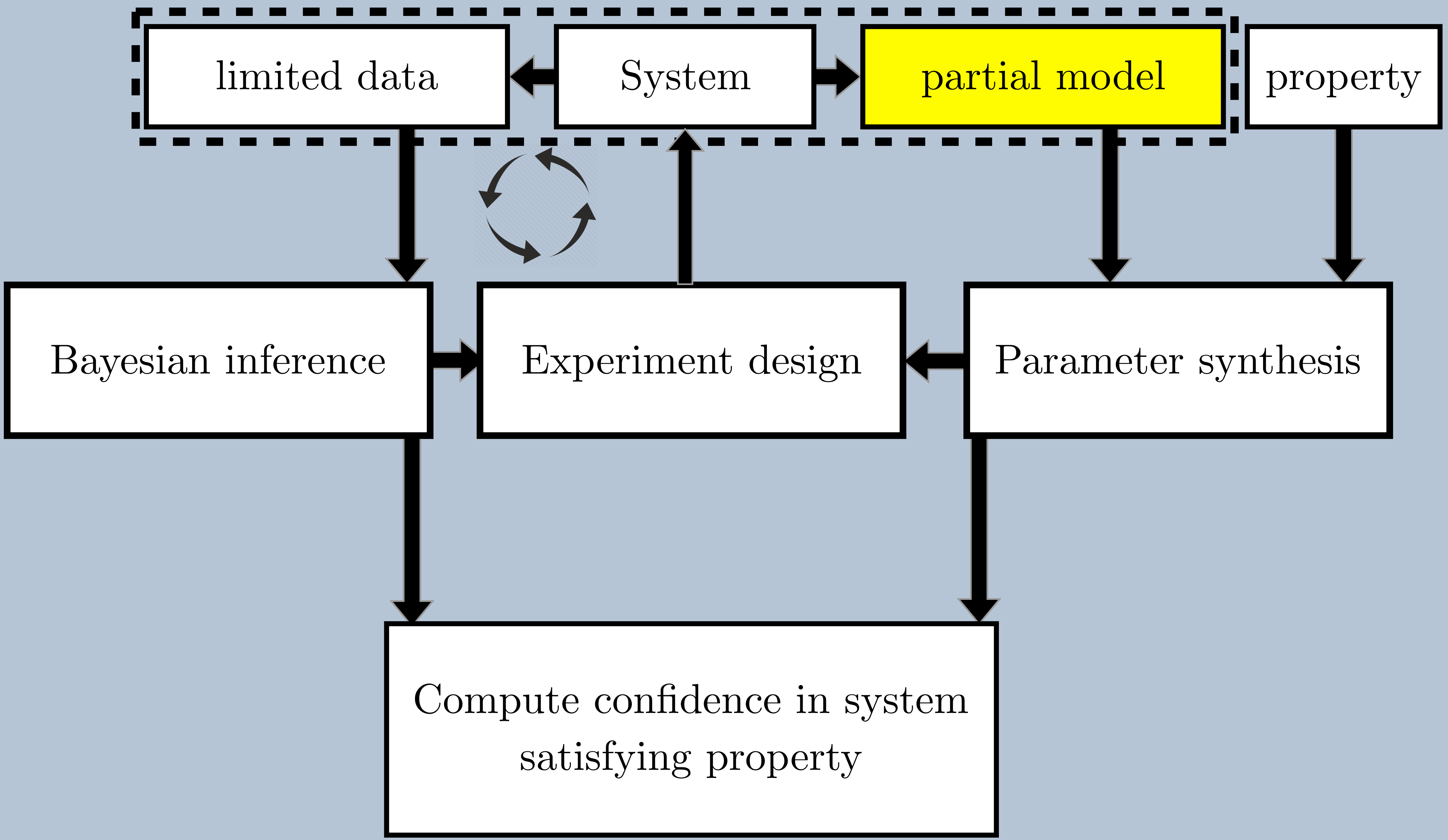


Overview



limited data
System
partial model
property
Bayesian inference
Experiment design
Parameter synthesis
Compute confidence in system satisfying property

Overview


limited data
System
partial model
property
Bayesian inference
Experiment design
Parameter synthesis
Compute confidence in system satisfying property

Parametric Markov Decision Process

initial state

atomic propositions

set of actions

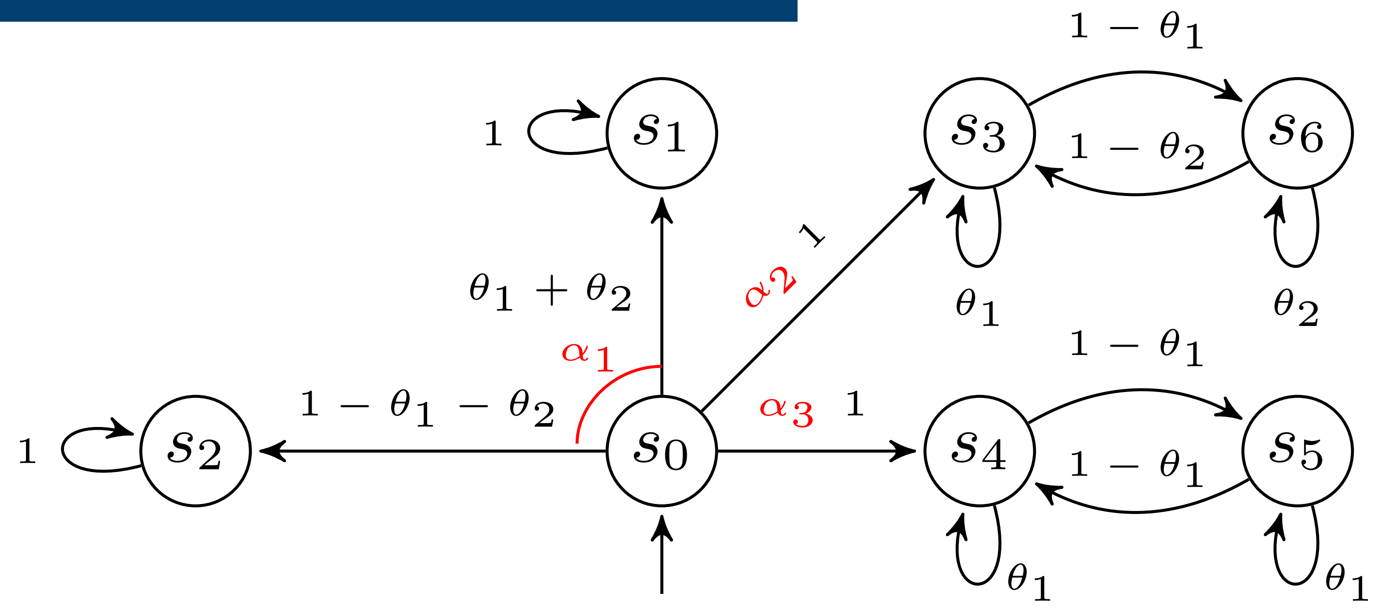
labelling function

finite, non-empty set of states

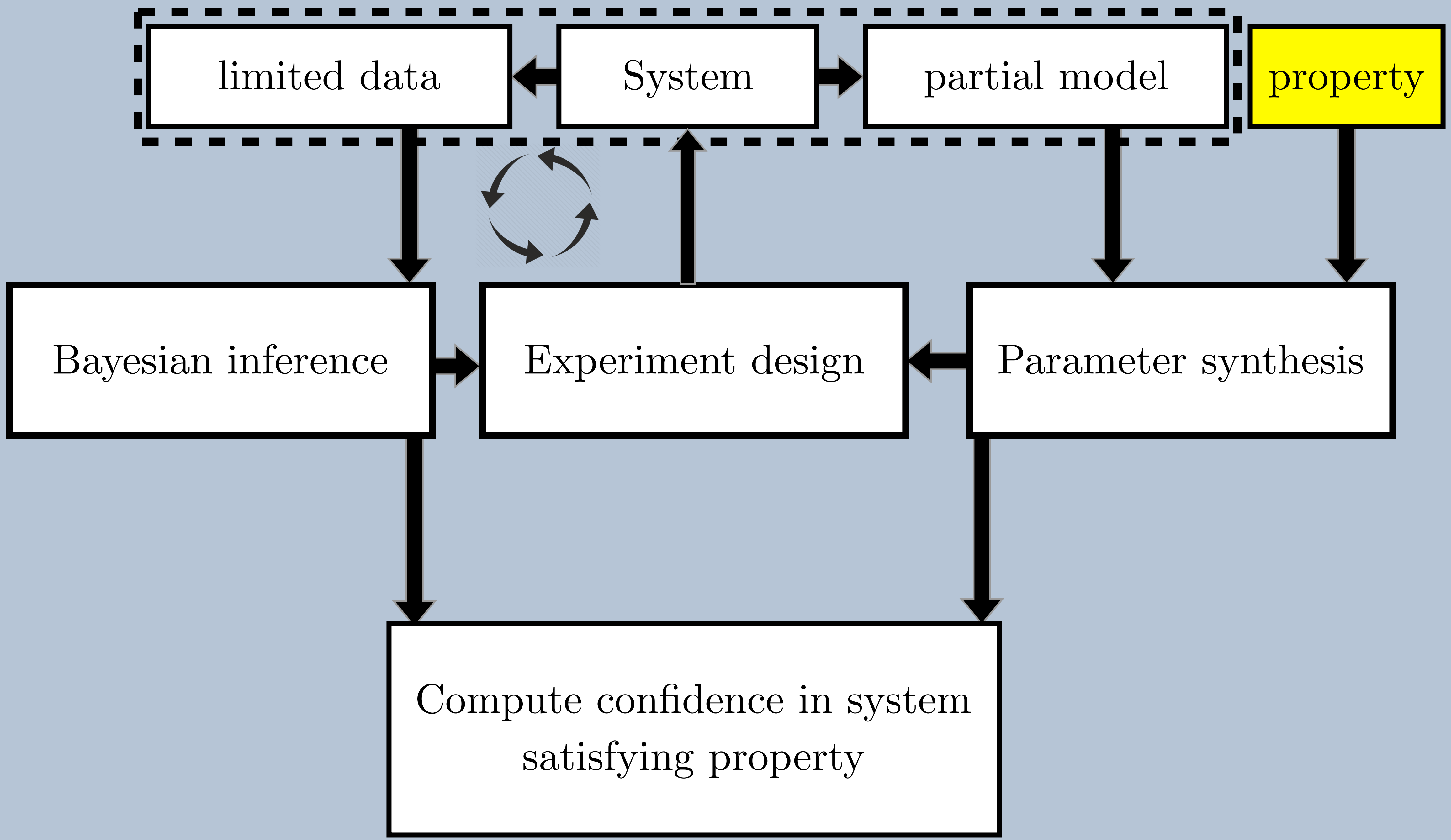
$$\mathbf{M}_{\Theta} = (S, \mathrm{Act}, \mathbb{T}_{\theta}, \iota_{init}, \mathrm{AP}, L, \Theta)$$

$\mathbb{T}_{\theta} =$ *transition probability function*

$\Theta =$ *set of all possible evaluations of* θ



Overview



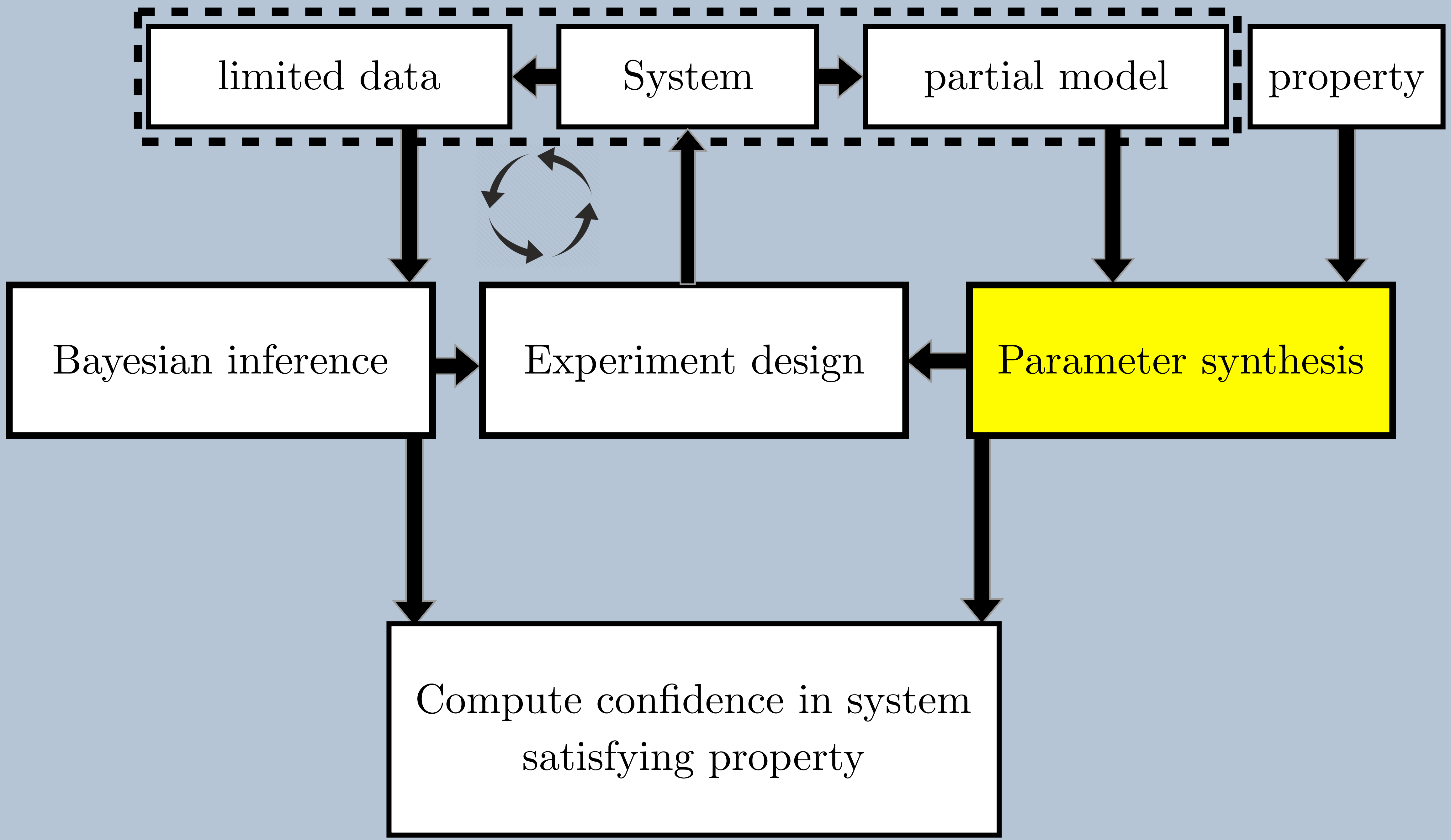
limited data
System
partial model
property
Bayesian inference
Experiment design
Parameter synthesis
Compute confidence in system satisfying property

PCTL properties

We are able to consider any property that is compatible with the PRISM parameter synthesis tool. We focus on non-nested PCTL:

$$\mathbf{P}_{\geq 0.5}(\text{true}\ \mathcal{U}\ \text{complete})$$

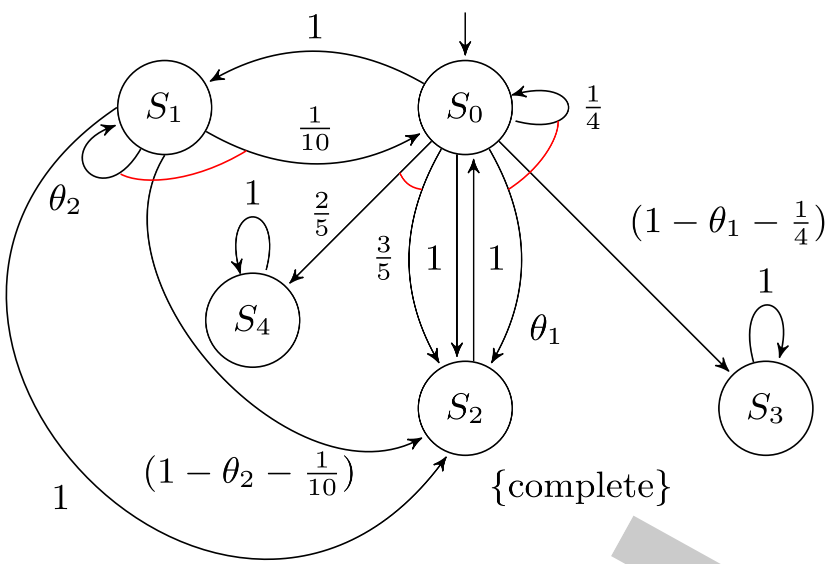
Overview



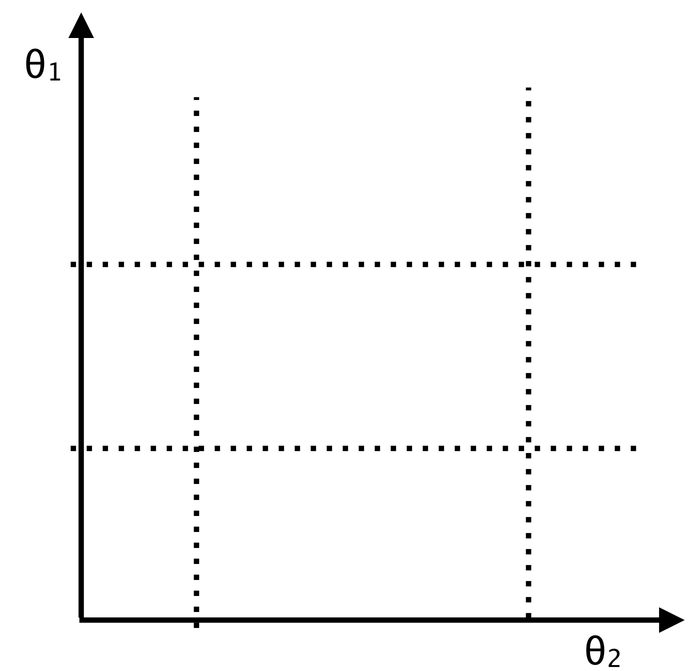
limited data
System
partial model
property
Bayesian inference
Experiment design
Parameter synthesis
Compute confidence in system satisfying property

Parameter Synthesis

We use PRISM to synthesise the feasible set of parameters, for which the model satisfies the property:

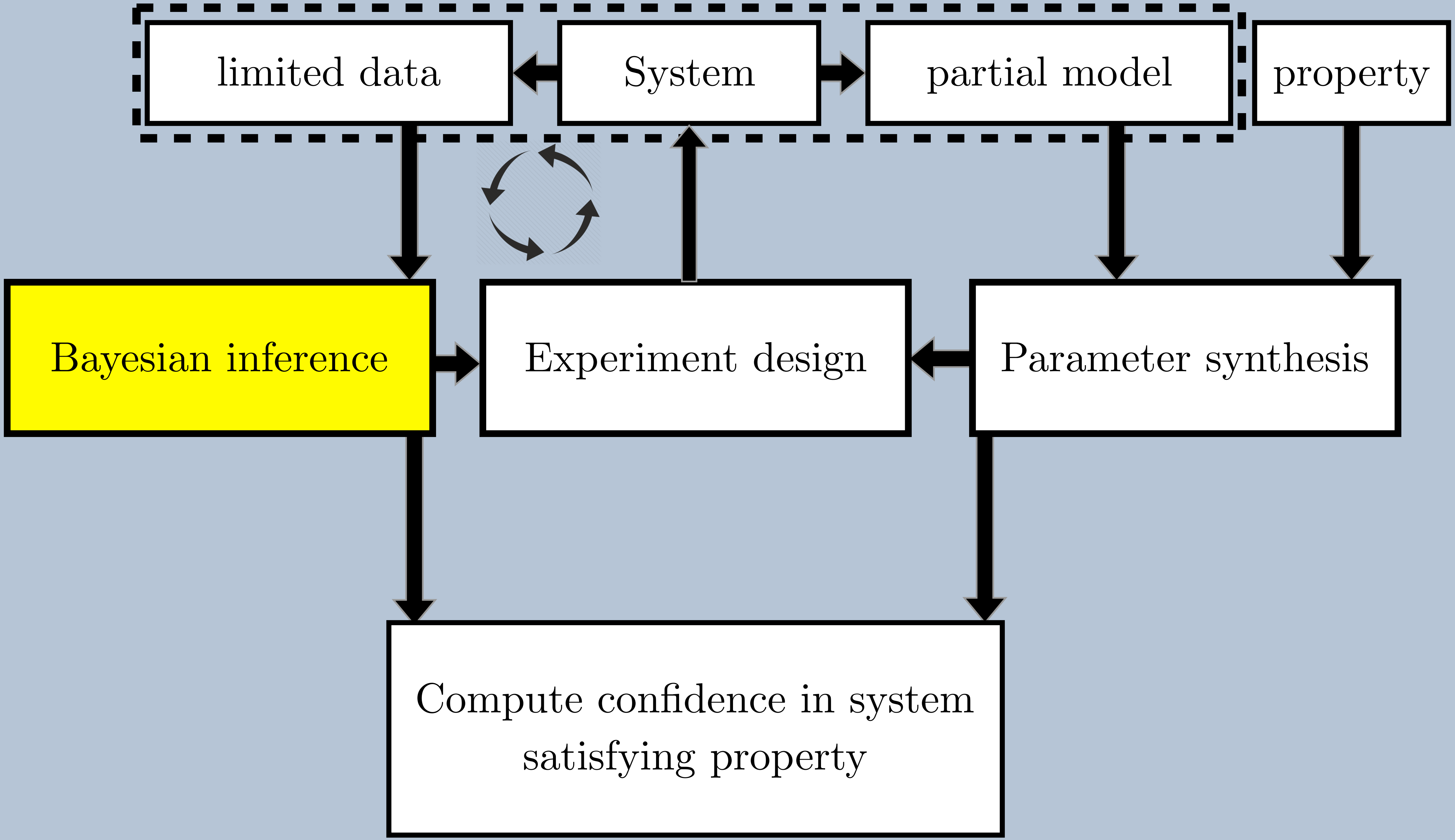


$\mathbf{P}_{\geq 0.5}(\text{true } \mathcal{U} \text{ complete})$



$$\Theta_\phi = \{\theta \in \Theta : \mathbf{M}(\theta) \models \phi\}$$

Overview


limited data
System
partial model
property
Bayesian inference
Experiment design
Parameter synthesis
Compute confidence in system satisfying property

Data

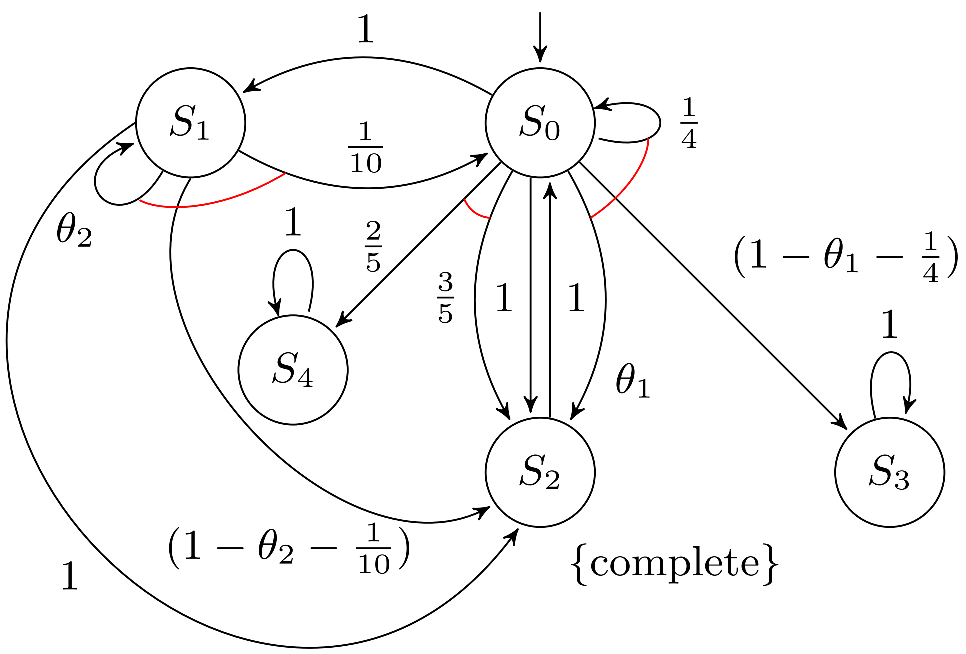
- We collect data from the underlying system in the form of finite number of finite traces
- We turn this into transition counts, group by parameter

$$\boxed{D_{\theta_j,\neg\theta_j} = D_{\theta_j}, D_{\neg\theta_j}}$$

$$D_{\theta_j} = \sum_{s_i \in S, s_l \in S, \alpha_k \in Act} D_{s_i,\alpha_k,s_l} \text{ for } \mathbb{T}(s_i, \alpha_k, s_l) = \theta_j$$

$$D_{\neg\theta_j} = \sum_{s_i \in S, s_l \in S, \alpha_k \in Act} D_{s_i,\alpha_k,s_l} \text{ for } \mathbb{T}(s_i, \alpha_k, s_l) \neq \theta_j \wedge \exists s_m \in S : \mathbb{T}(s_i, \alpha_k, s_m) = \theta_j$$

Bayesian Inference



observed data *prior*

$$p(\theta_j \mid D) = \frac{\mathbb{P}(D \mid \theta_j)p(\theta_j)}{\mathbb{P}(D)}$$

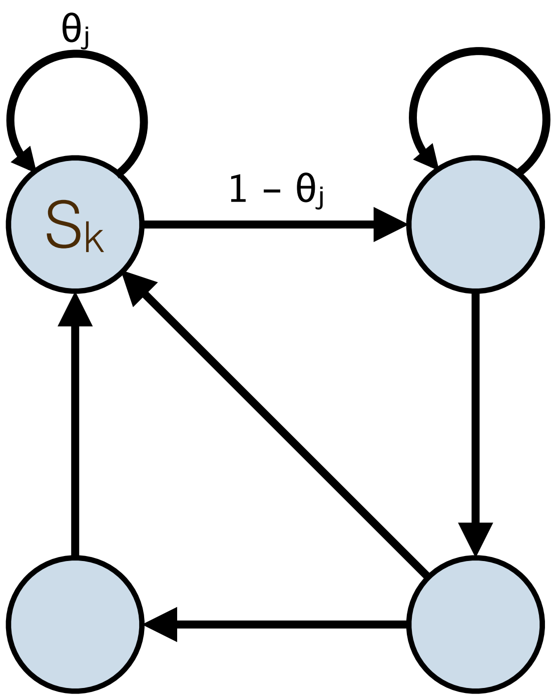
$$= \frac{p(\theta_j)\theta_j^{D_{\theta_j}}(1-\theta_j)^{D_{\neg\theta_j}}}{\mathbb{P}(D_{\theta_j,\neg\theta_j})}$$

binomial distribution

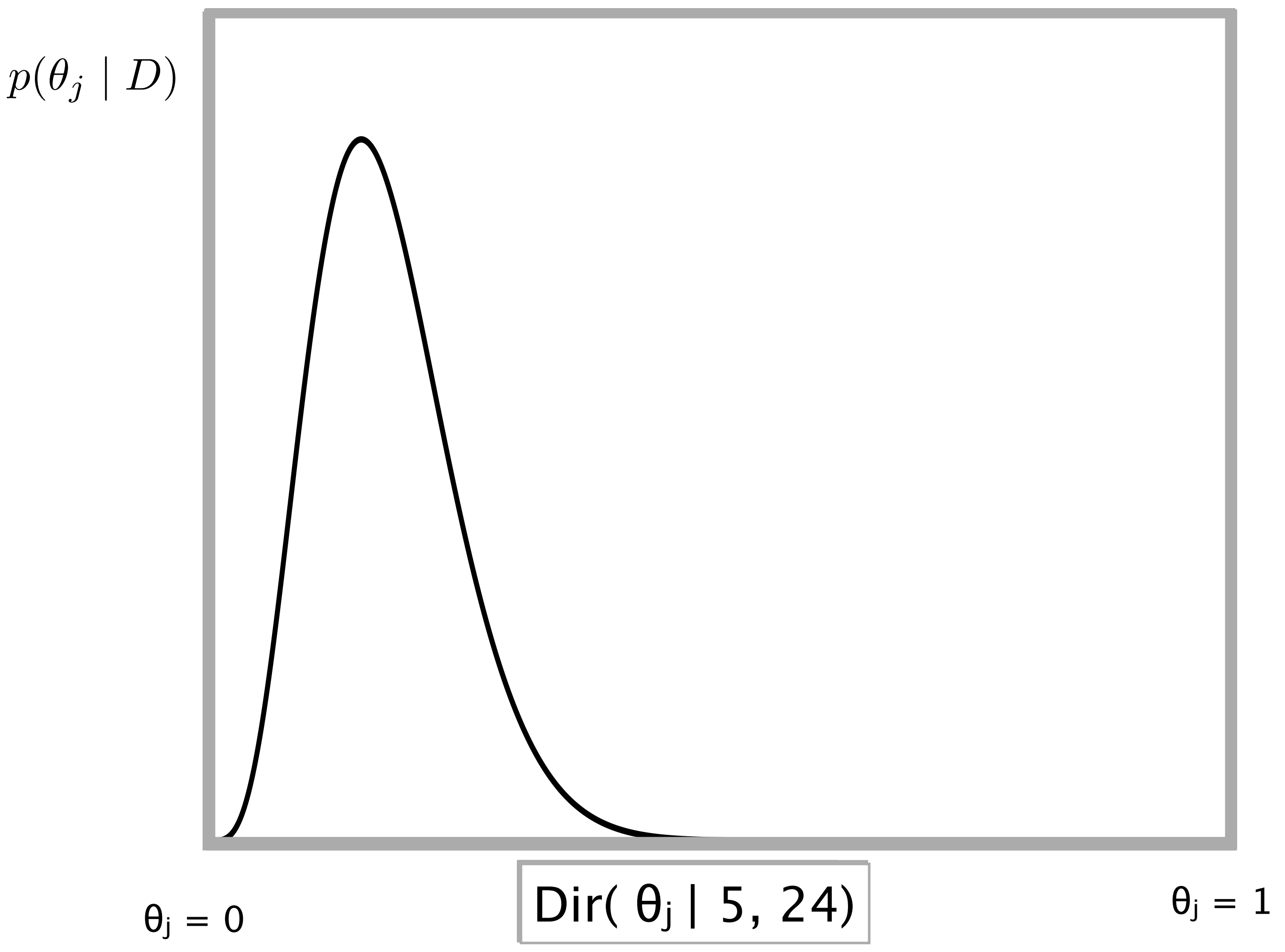
Conjugate prior $=$ Dirichlet

$$\mathrm{Dir}(\theta_j \mid \alpha) = \frac{1}{B(\alpha)}\theta_j^{\alpha_1-1}(1-\theta_j)^{\alpha_2-1}$$

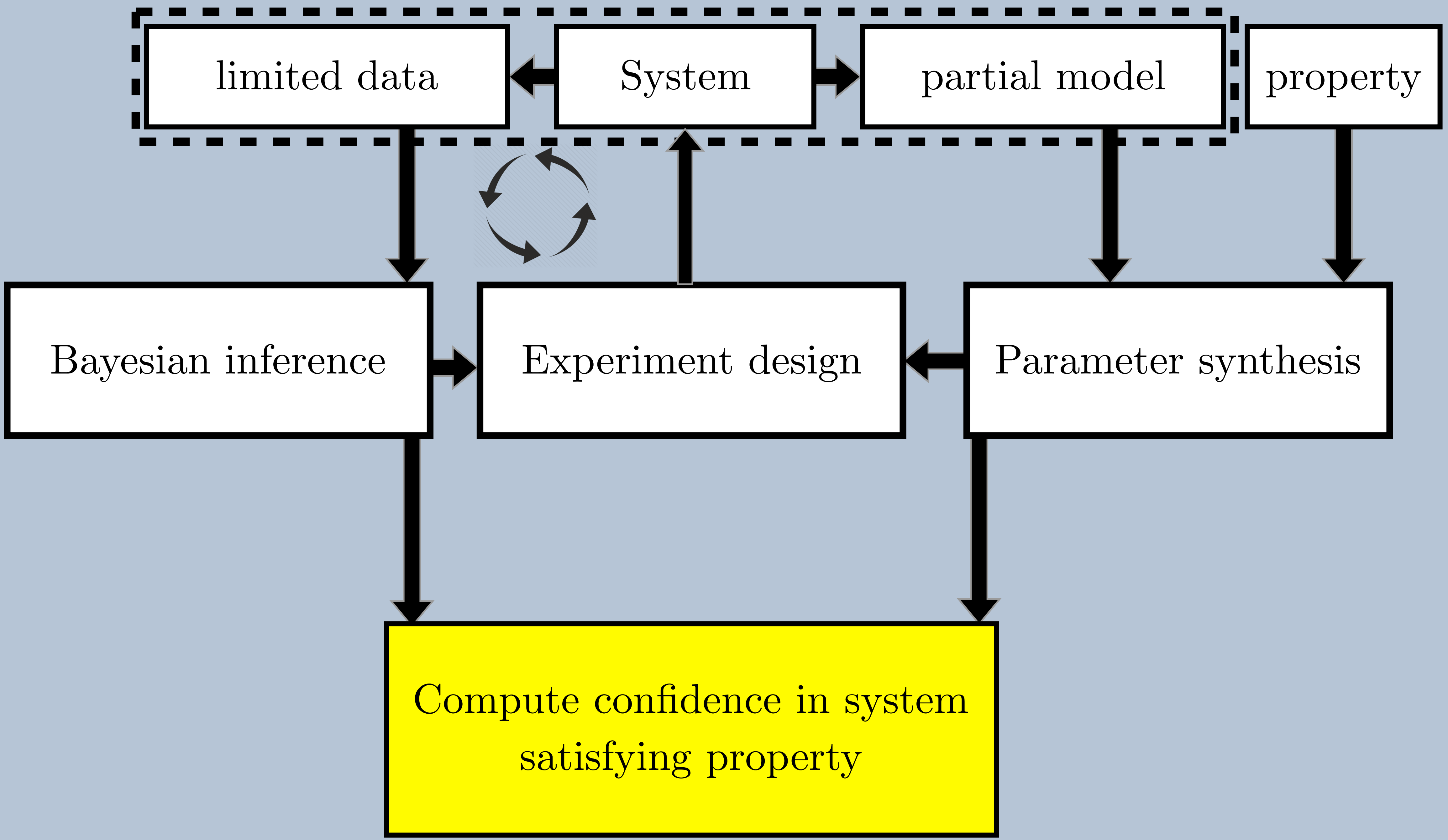
Bayesian Inference



COUNT [1 – θ_j]	23
COUNT [θ_j]	4



Overview

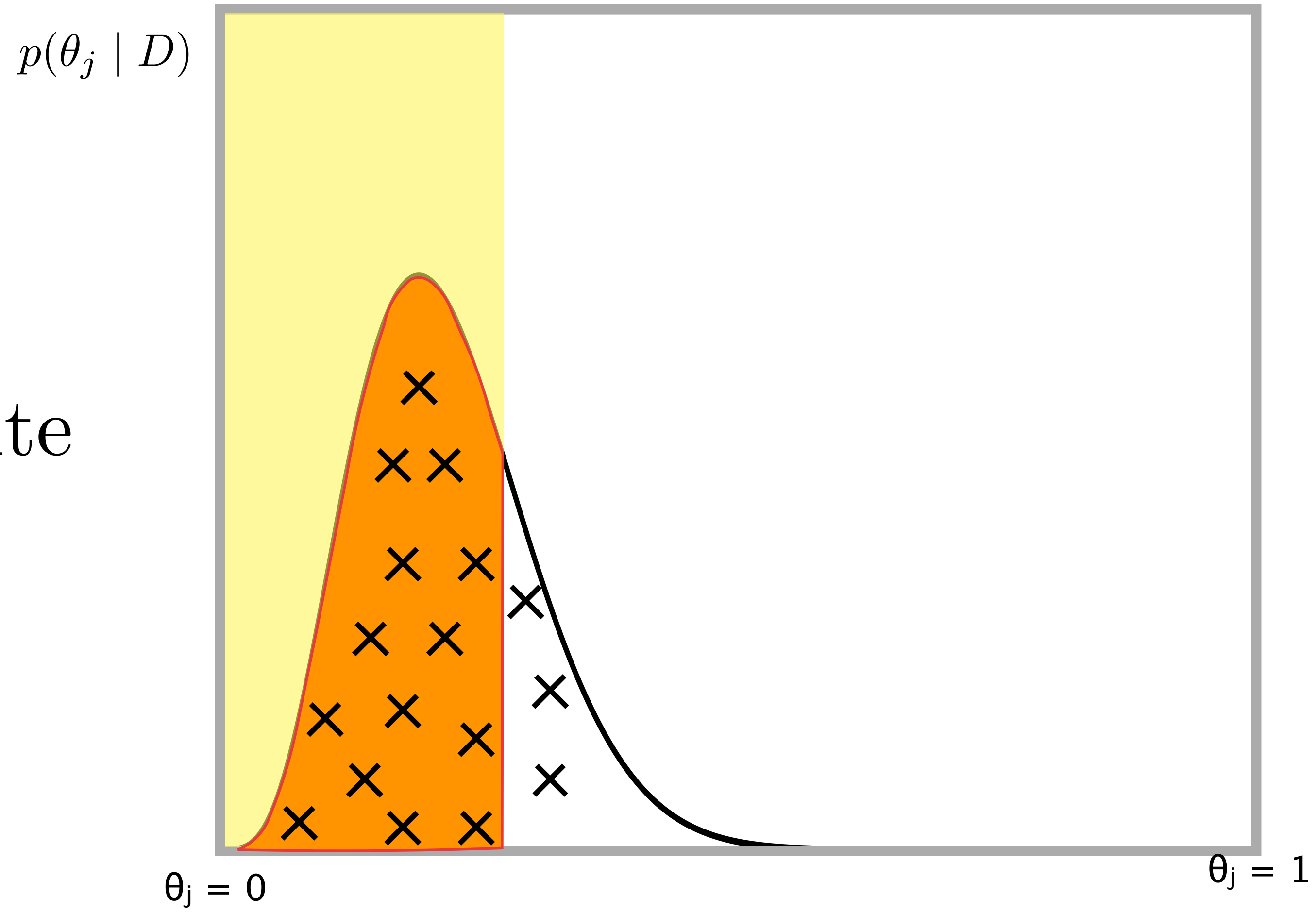

limited data
System
partial model
property
Bayesian inference
Experiment design
Parameter synthesis
Compute confidence in system satisfying property

Confidence Calculation

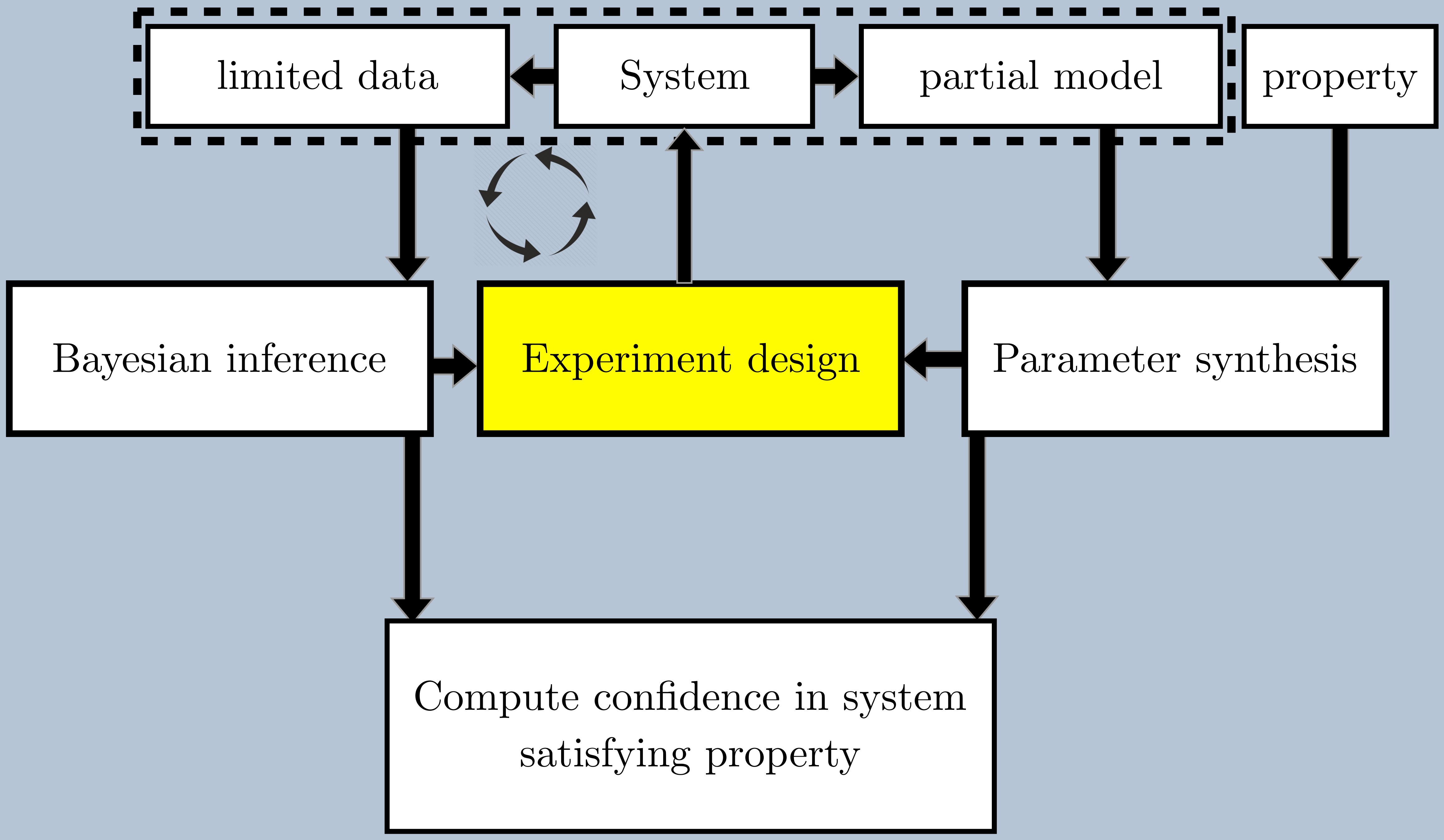
$$\mathbb{P}(\mathbf{S} \models \phi \mid D) = \int_{\Theta_\phi} p(\theta \mid D) d\theta$$

$$\Theta_\phi = \{\theta \in \Theta : \mathbf{M}(\theta) \models \phi\}$$

Note: we use simple Monte Carlo to compute the integral.



Overview

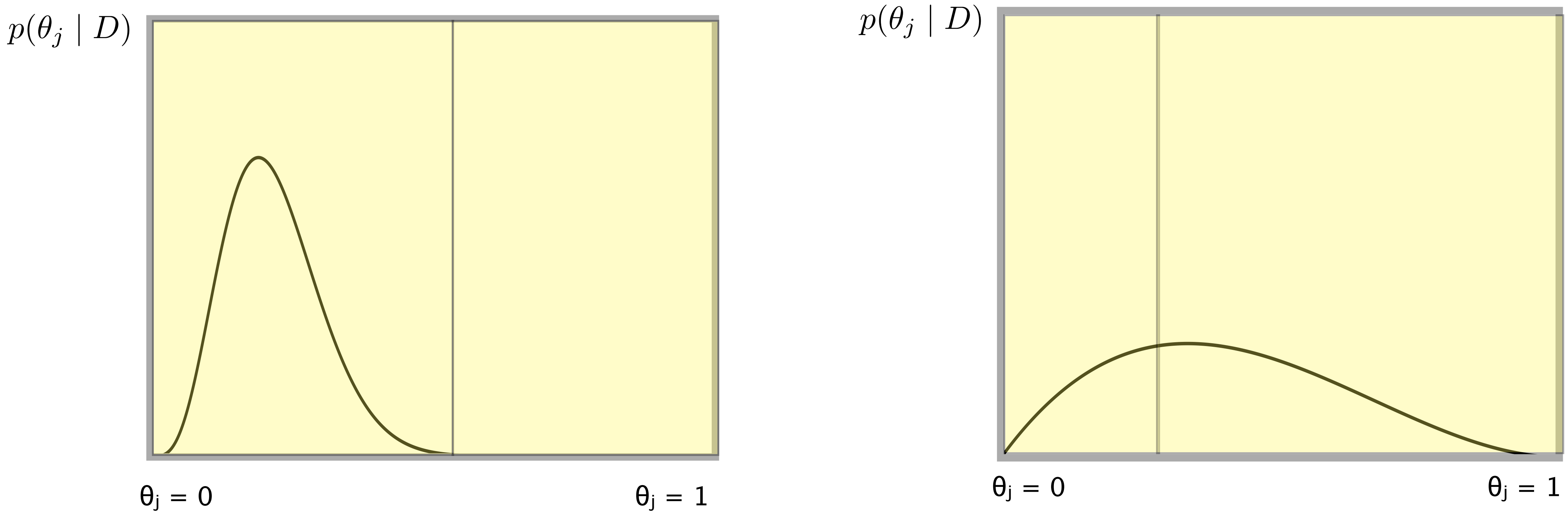

limited data
System
partial model
property
Bayesian inference
Experiment design
Parameter synthesis
Compute confidence in system satisfying property

Experiment design

Some data is more useful than other data, i.e., it tells us more about whether the property is satisfied by the system. e.g.,

- traces with no parameterised transitions are useless
- knowledge about some parameters is more useful than others.

The "utility" of data containing a parameter is a function of the posterior distribution for a parameter, and the feasible set:



Experiment design

- We estimate the “utility” of picking an action by predicting the confidence after we take the action:

$$\mathcal{C}^{\text{pred}}_{s,\alpha} = \int_{\Theta_\phi} \prod_{\theta_i \in \theta} p(\theta_i \mid \mathbb{E}_{s,\alpha}\left(D_{\theta_i,\neg\theta_i}\right))d\theta,$$

- We can then estimate “information gain” and assign it to a state-action pair as a reward:

$$\mathbb{G}_{s,\alpha} = |0.5 - \mathcal{C}^{\text{pred}}_{s,\alpha}| - |0.5 - \mathcal{C}|$$

Experiment design

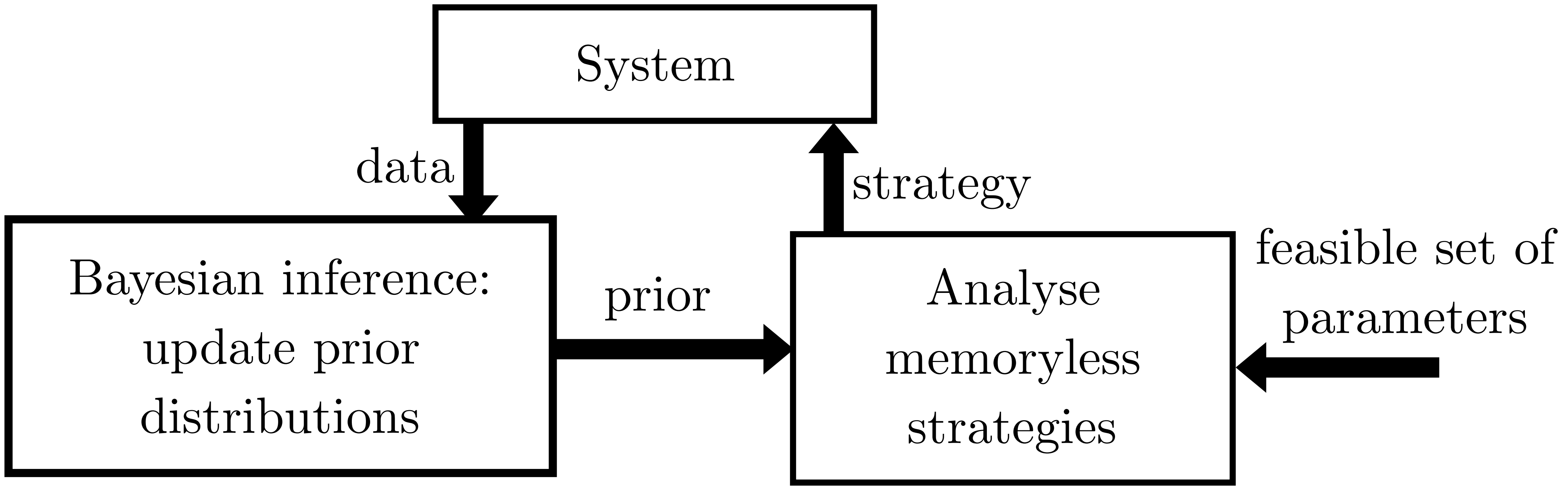
- We compute the information gain for every state-action pair in the MDP
- For a trace of length N, the optimal information gain is then:

$$x_s^t = \begin{cases} \max_{\alpha \in Act(s)}(\mathbb{G}_{s,\alpha} + \sum(\mathbb{T}(s,\alpha,s').\, x_{s'}^{t+1})) & \text{if } 0 < t < N \\ 0 & \text{if } t \geq N. \end{cases}$$

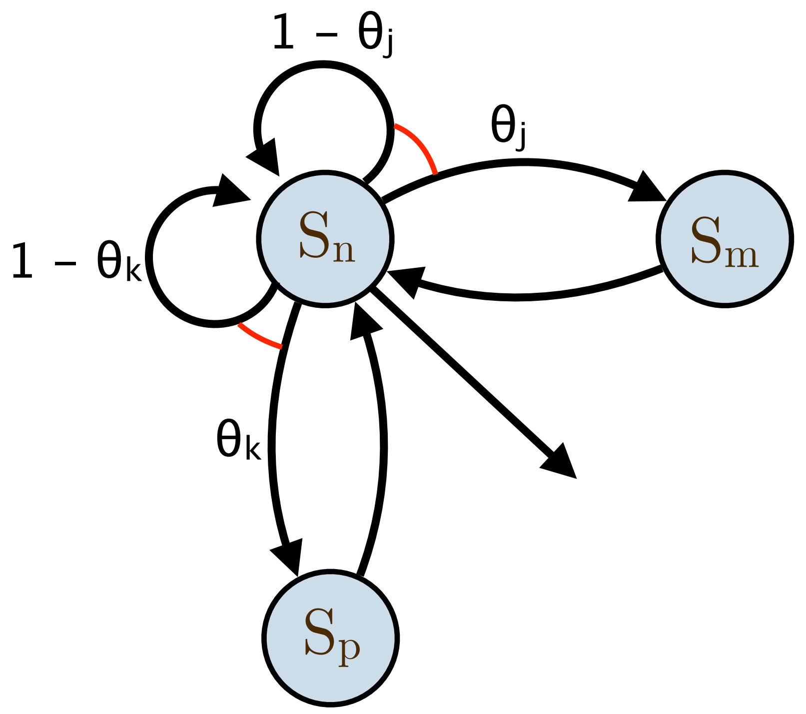
$\mathbb{G}_{s,\alpha}$ depends on the distribution of the parameters at time t

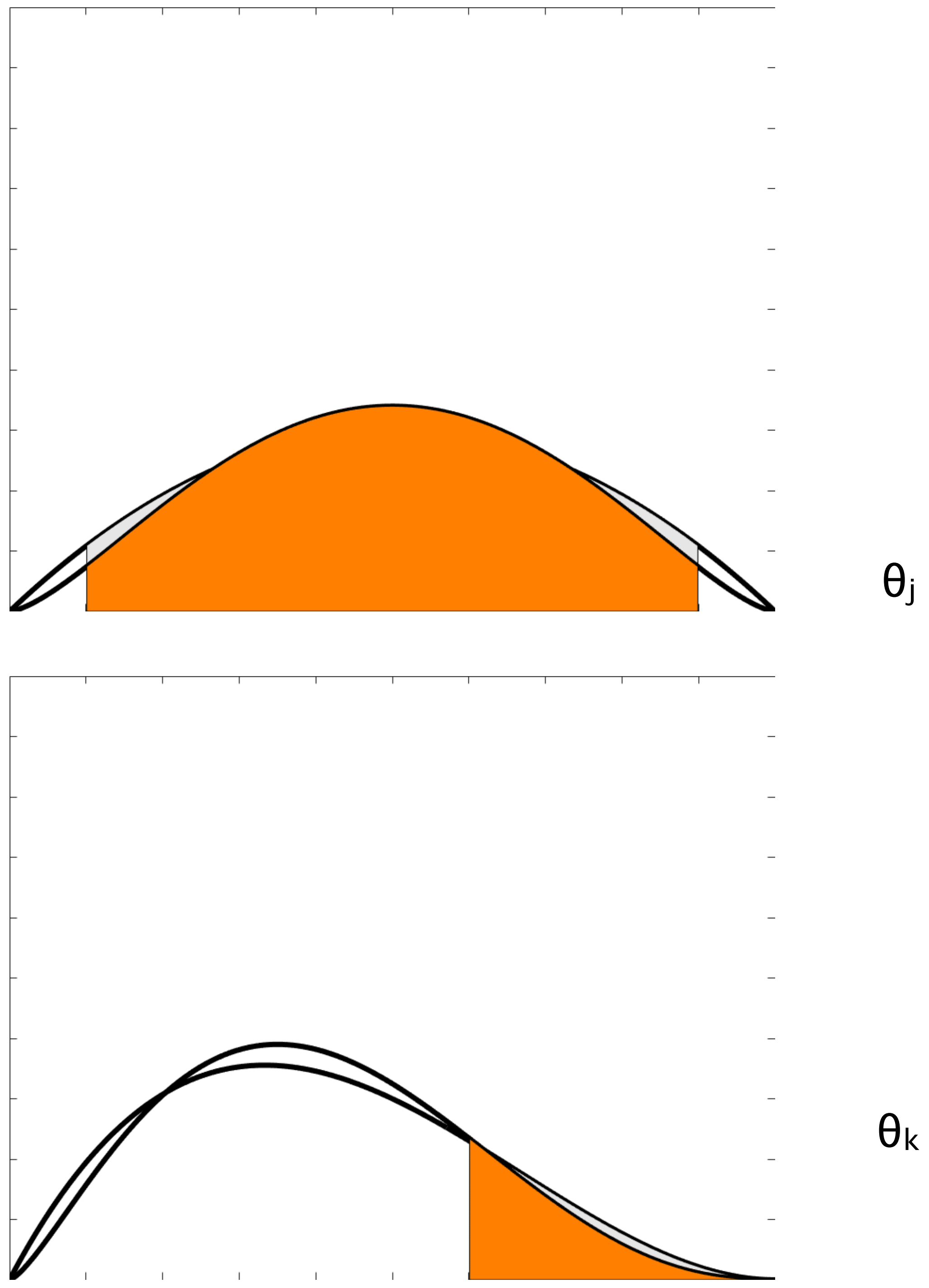
Experiment design

- The memory dependence of the information gain makes finding the optimal strategy hard
- To simplify the problem we consider only memoryless strategies



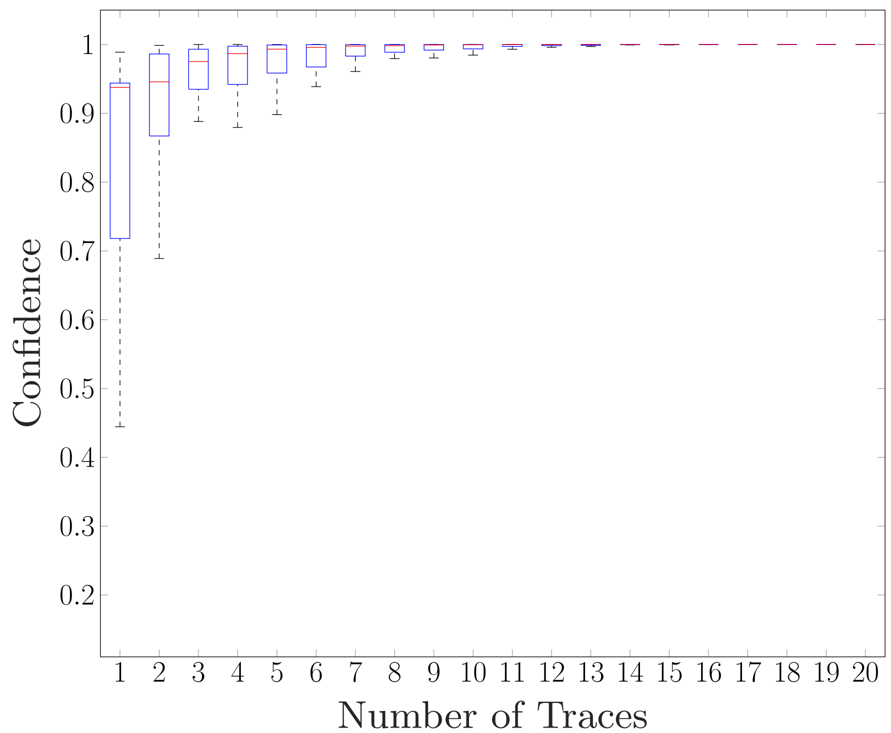
Experiment design



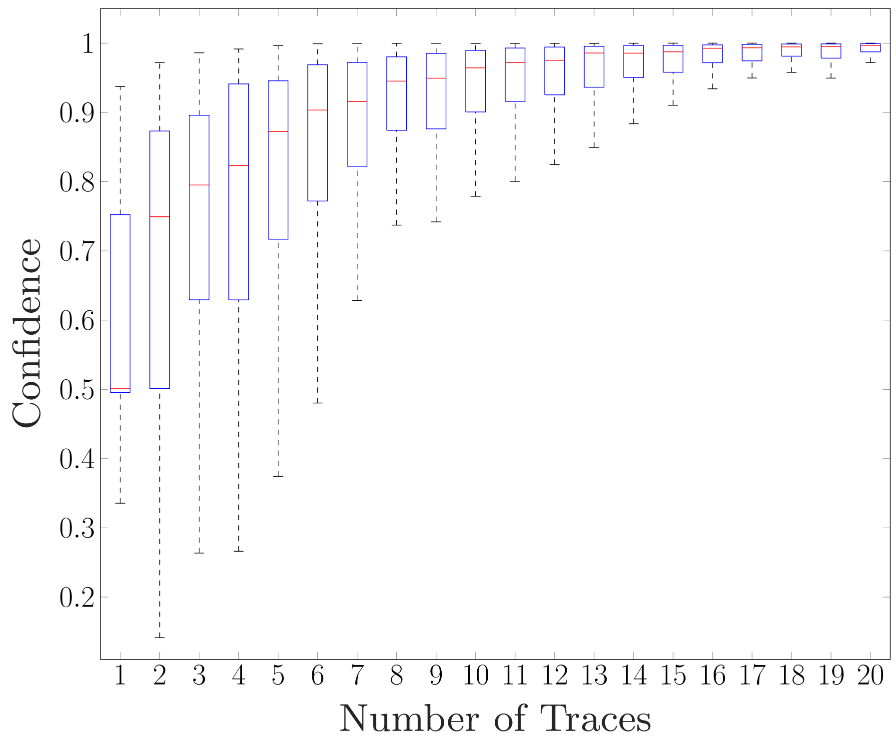


Evaluation

- Previous work has shown that the Bayesian verification framework uses data more efficiently than other statistical methods
- We compare our automated experiment design with the basic Bayesian verification framework using no strategy (randomly selecting actions)



Experiment design



No experiment design

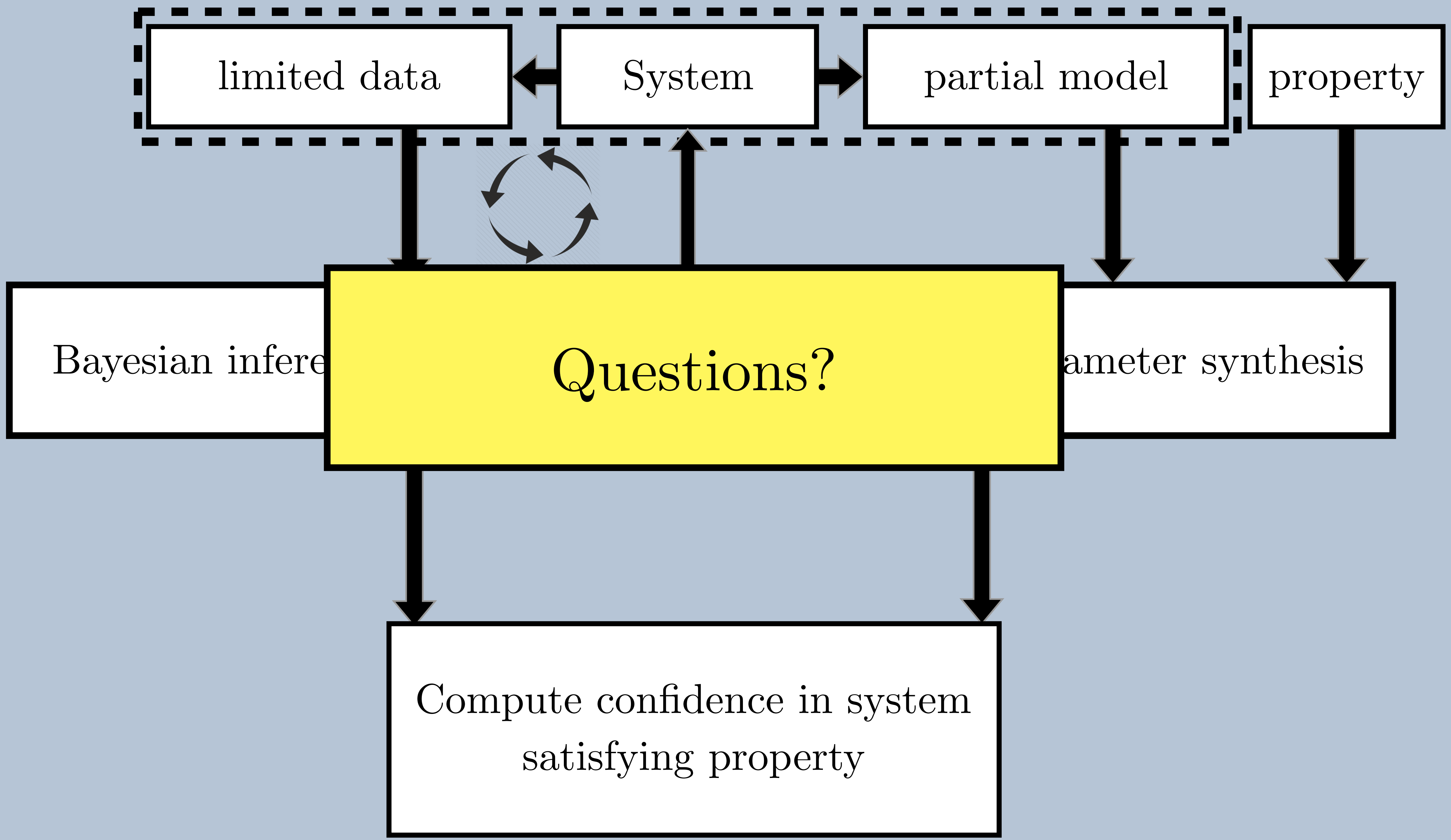
Conclusions

- We have extended the Bayesian verification framework to systems with external non-determinism
- We have shown that automated experiment design reduces the amount of data needed

Future work

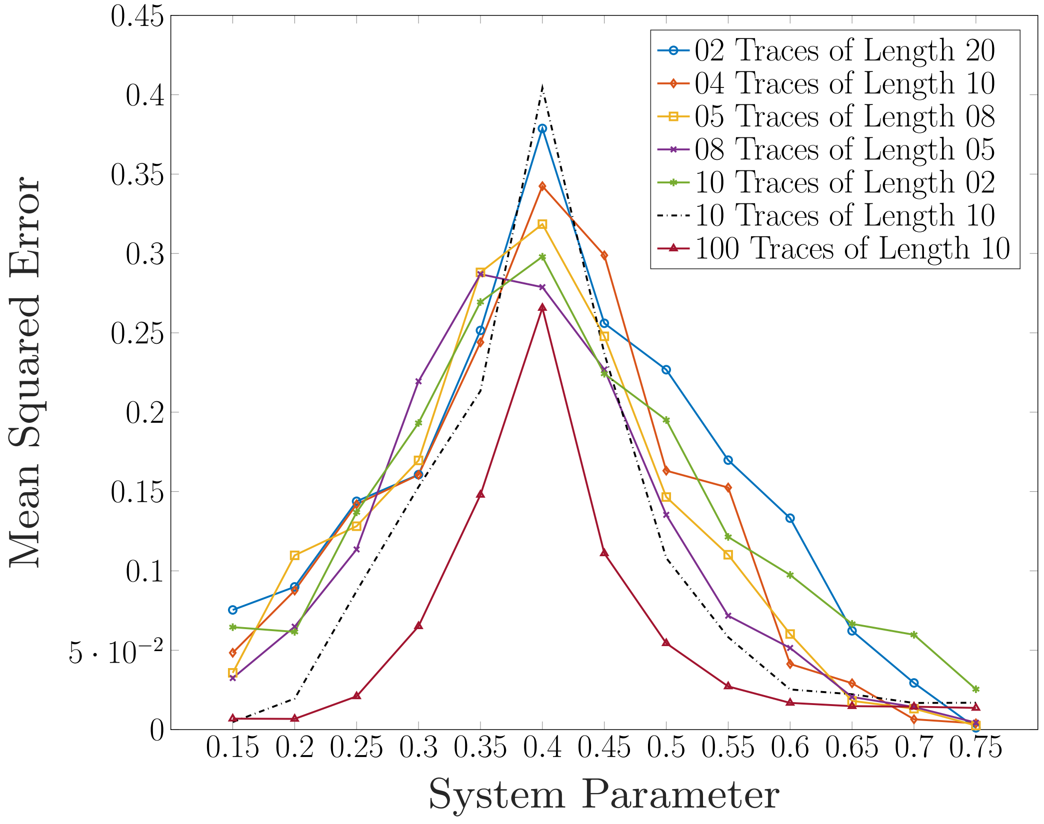
- Improvements to the experiment design
- Other frameworks: continuous time

Overview


limited data
System
partial model
property
Bayesian infere
ameter synthesis
Questions?
Compute confidence in system satisfying property

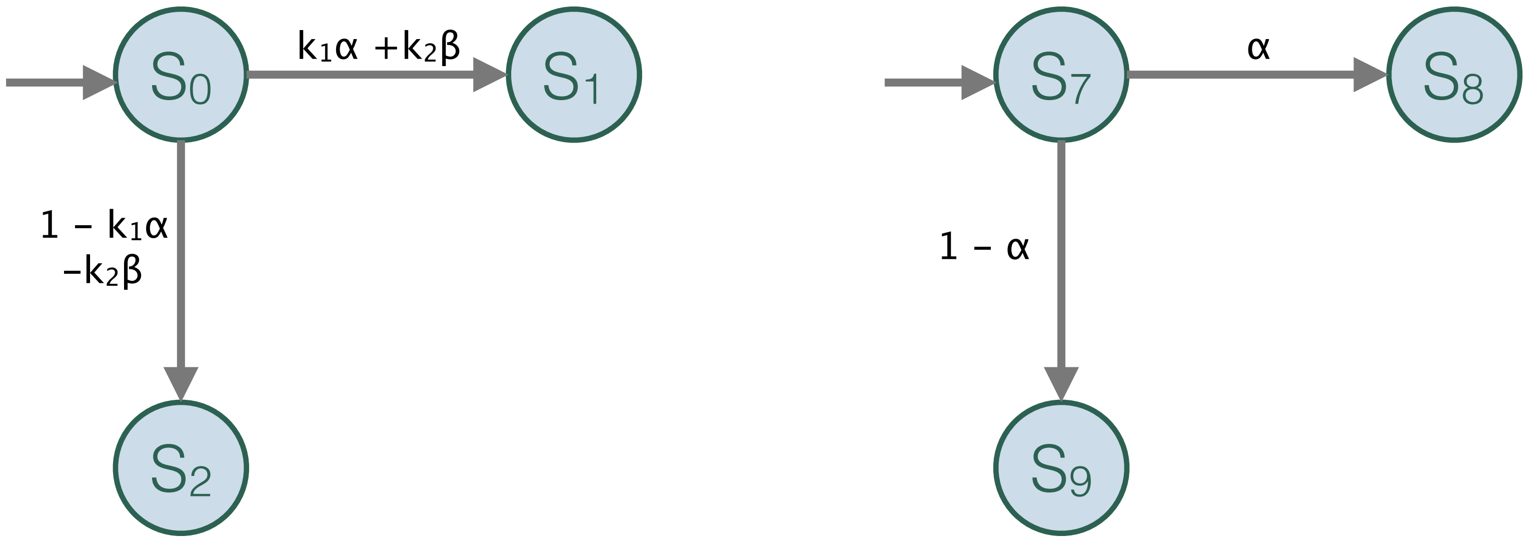
Evaluation

- Previous work has shown that the Bayesian verification framework uses data more efficiently than other statistical methods
- We compare our automated experiment design with the basic Bayesian verification framework using no strategy (randomly selecting actions)



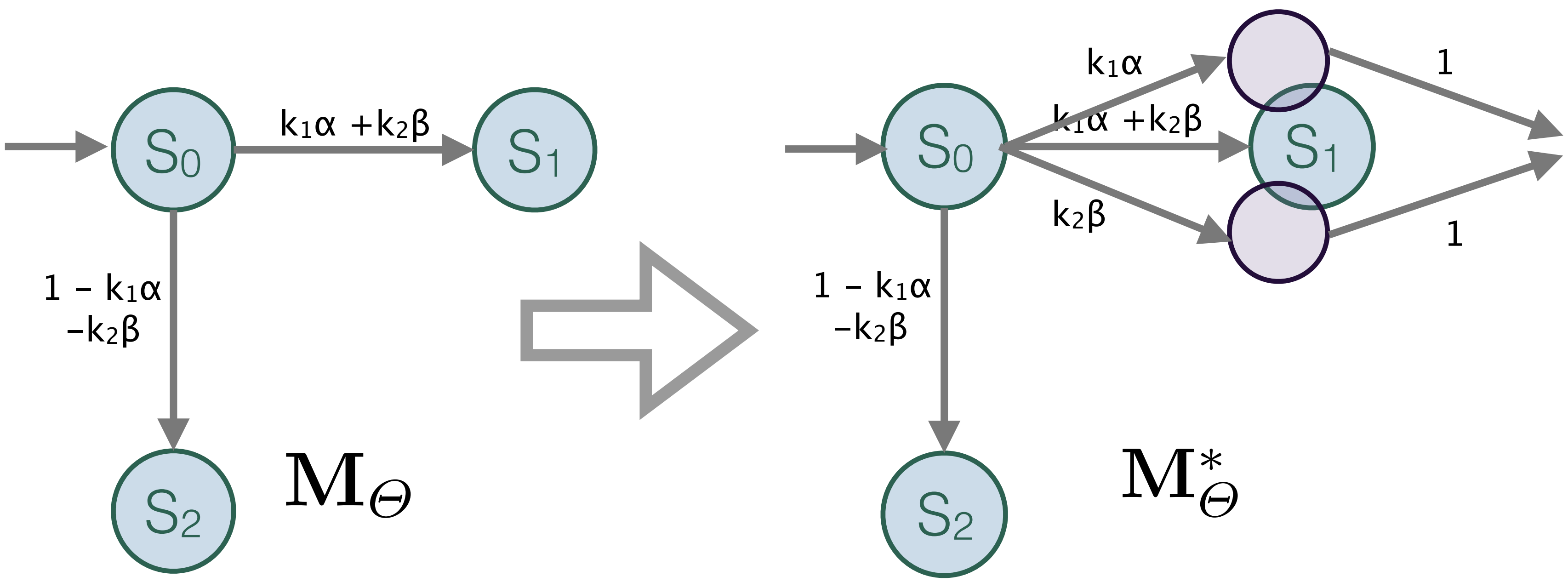
Markov chain expansion

What if a parameter appears multiple times in a linear pMC, in different linear equations? How do we combine the posterior distributions?



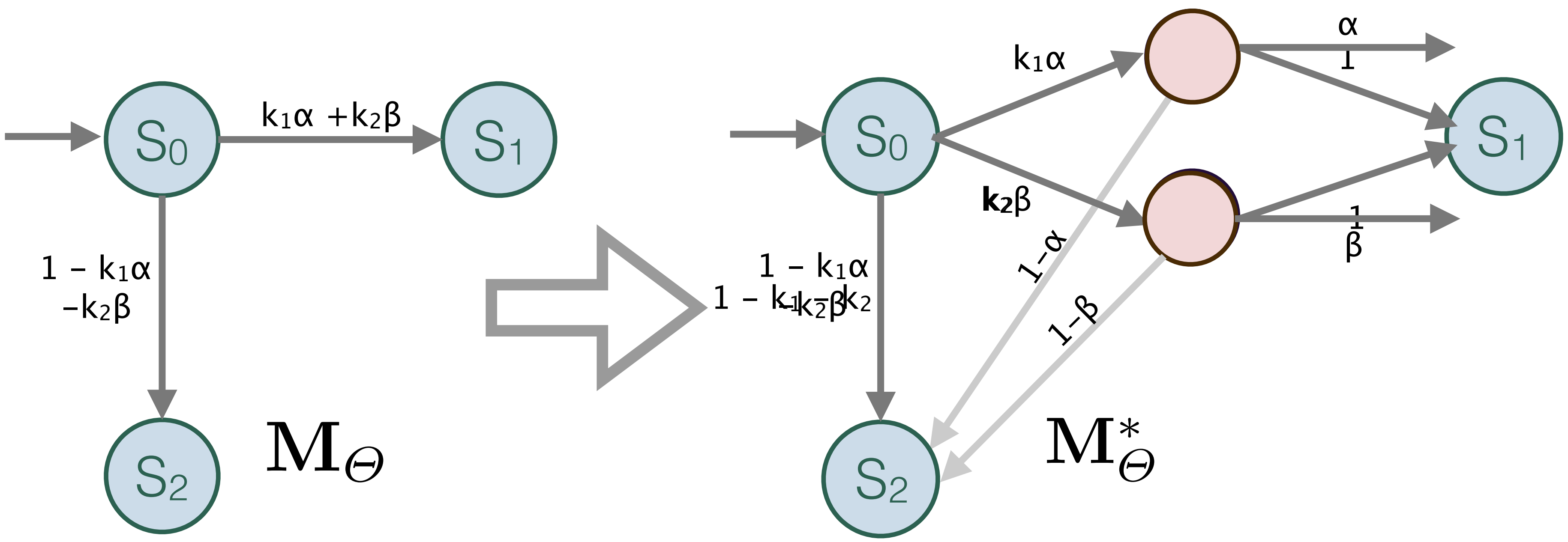
Markov chain expansion

We "expand" the transitions with linear parameterisation, to turn the MC into a basic pMC. i.e., transitions have only one parameter.



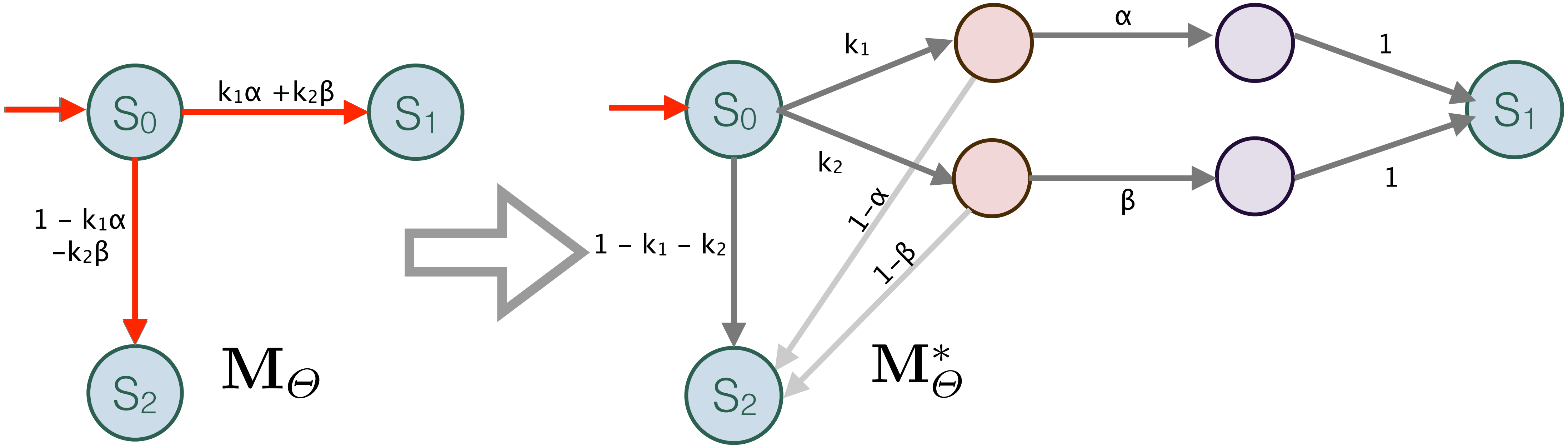
Markov chain expansion

We “expand” the transitions with linear parameterisation, to turn the MC into a basic pMC. i.e., transitions have only one parameter.

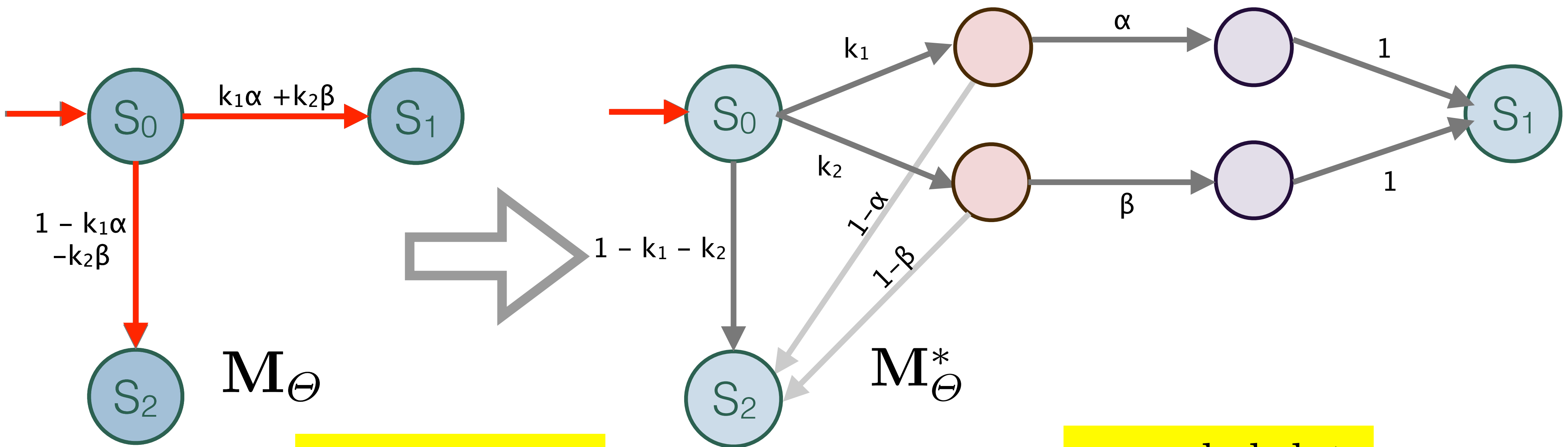


Hidden Data

We now have a data set with gaps in. We know the transitions counts only for the original transitions.



We apply Bayes' rule



original data

expanded data

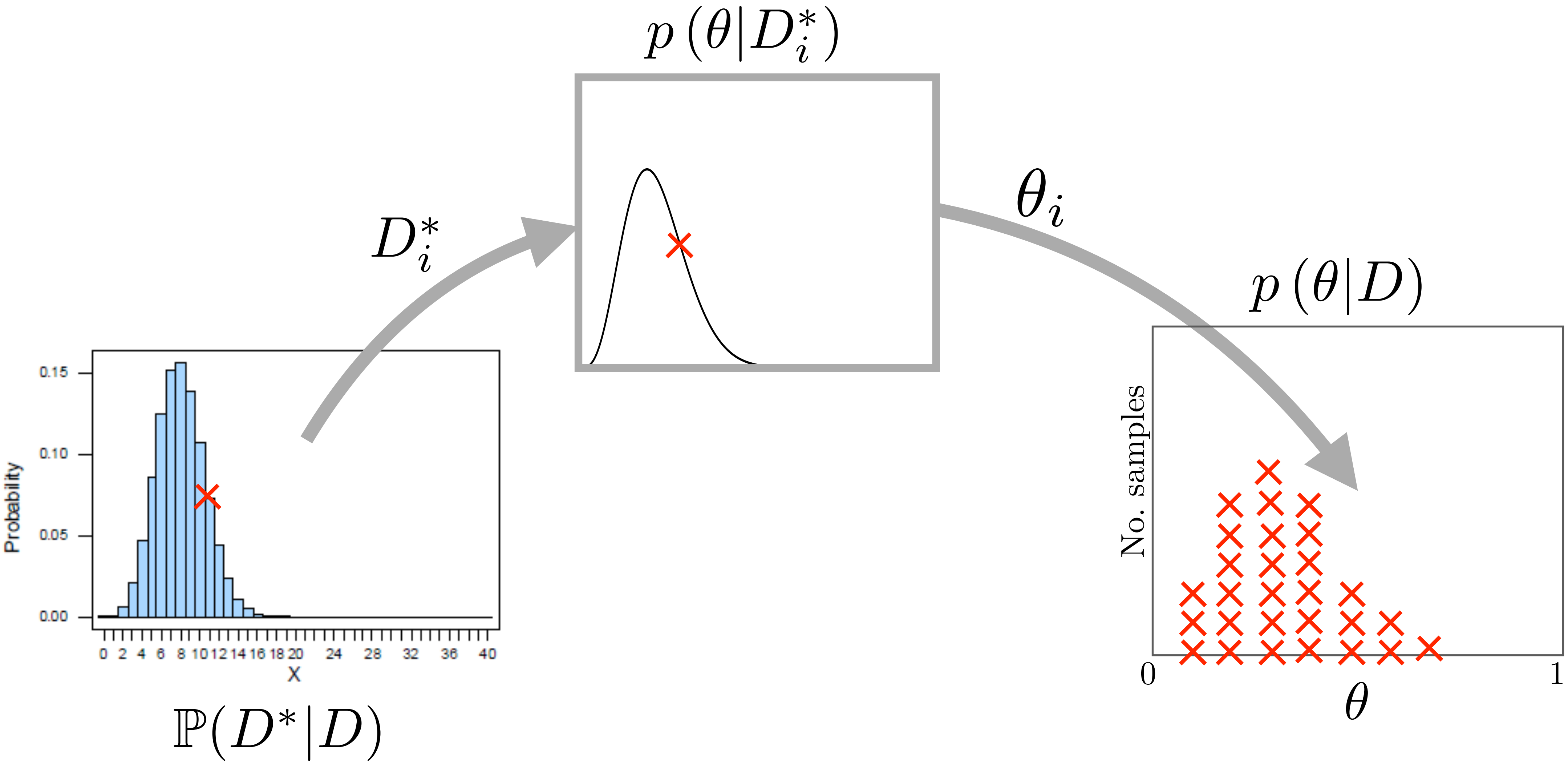
$$p\left(\theta \middle| D\right) = \sum_{D^* \in \mathcal{D}^*} p\left(\theta \middle| D^*\right) \mathbb{P}(D^* | D)$$

set of all possible completions of the expanded data

Hidden Data

We use sampling to obtain a realisation of the posterior distribution, without evaluating the integral

$$p\left(\theta|D\right) = \sum_{D^* \in \mathcal{D}^*} p\left(\theta|D^*\right) \mathbb{P}(D^*|D)$$



Confidence Calculation

$$\boxed{\mathbb{P}(\mathbf{S} \models \phi \mid D)} = \int_{\boxed{\Theta_\phi}} \boxed{p(\theta \mid D)} d\theta$$

$$\Theta_\phi = \{\theta \in \Theta : \mathbf{M}(\theta) \models \phi\}$$