Causality
and
Simulation-Based Inference

Demian Wassermann 2022, Inria Saclay *île-de-France*

Causality

based on the presentation by I. Guyon et al.

Why Causality

AI / ML

• Underspecified Goals	➡Big Data Cures Everything
• Underspecified Limitations	➡Big Data Can Do Everything
• Underspecified Caveats	➡Big Data & Big Brother

Goals in AI

• Fair	➡Biases
• Accountable	➡explainability
• Transparent	➡Decision making can be supported
• Robust	➡attacks / manipulations

Why Causality —What's the Issue with pure AI

- Biases in data, lots of them
- Leads to biased learnt models
- Robustness
- Scope becomes very important

References

- C. O'Neill, Weapons of Math Destruction, 2016
- Zeynep Tufekci, We're building a dystopia just to make people click on ads, Ted Talks, Oct 2017.

Why Causality —Some Issues with “Data is Everything”

- Biases in data, lots of them
- Leads to biased learnt models
- Robustness
- Scope becomes very important

References

- C. O’Neill, Weapons of Math Destruction, 2016
- Zeynep Tufekci, We’re building a dystopia just to make people click on ads, Ted Talks, Oct 2017.

ML Approach to Explainable Models

Discriminative or Generative modelling

- Given

$$D = \{(x_i, y_i), x_i \in \mathbb{R}^d, i \in 1 \ldots N\}, \text{ iid samples } P(X,Y)$$

- Supervised learning $\hat{h} : X \to Y, \text{ i.e. } \hat{P}(Y|X)$
- Generative modelling $\hat{q} : X \times Y \to \mathbb{R}_+, \text{ i.e. } \hat{P}(X,Y)$

Lead to Predictive Modelling which will reproduce data biases

e.g.: If there are lots of umbrellas, then it rains

Caillebotte, 1877

ML Approach to Explainable Models

But Not All Biases are Bad

Seurat, 1884

The Implicit Big Data Promise

- If you can predict, can you control?

Knowledge -> Prediction -> Control

So How can this be Tested? Interventions

- Think about nutrition
- Think about healthcare
- Economy
- Climate

Pearl's "Do" operator: $do(X = a)$ means that we intervene a system on event X to make "a" true (Pearl 2009).

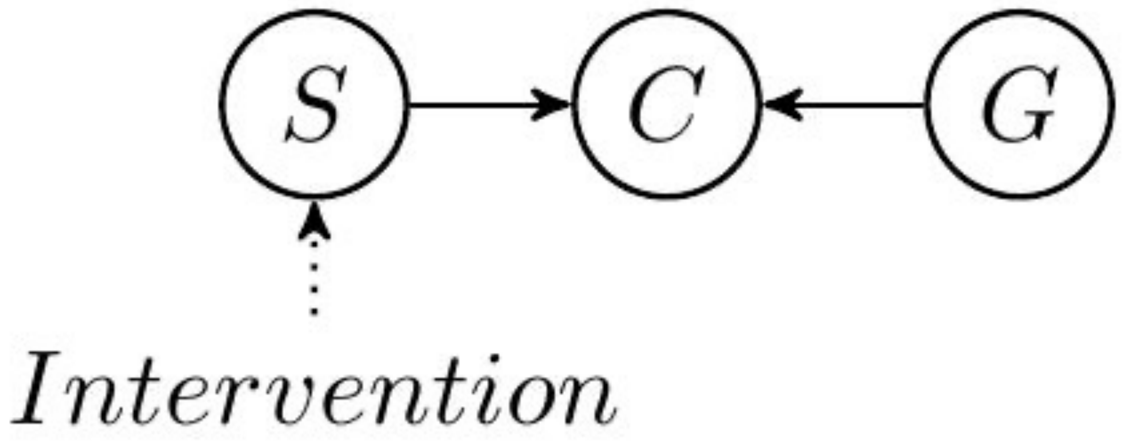
The Implicit Big Data Promise

X is a direct cause of Y if when we intervene it Y's law changes

$$X \to Y \quad \text{iif}$$

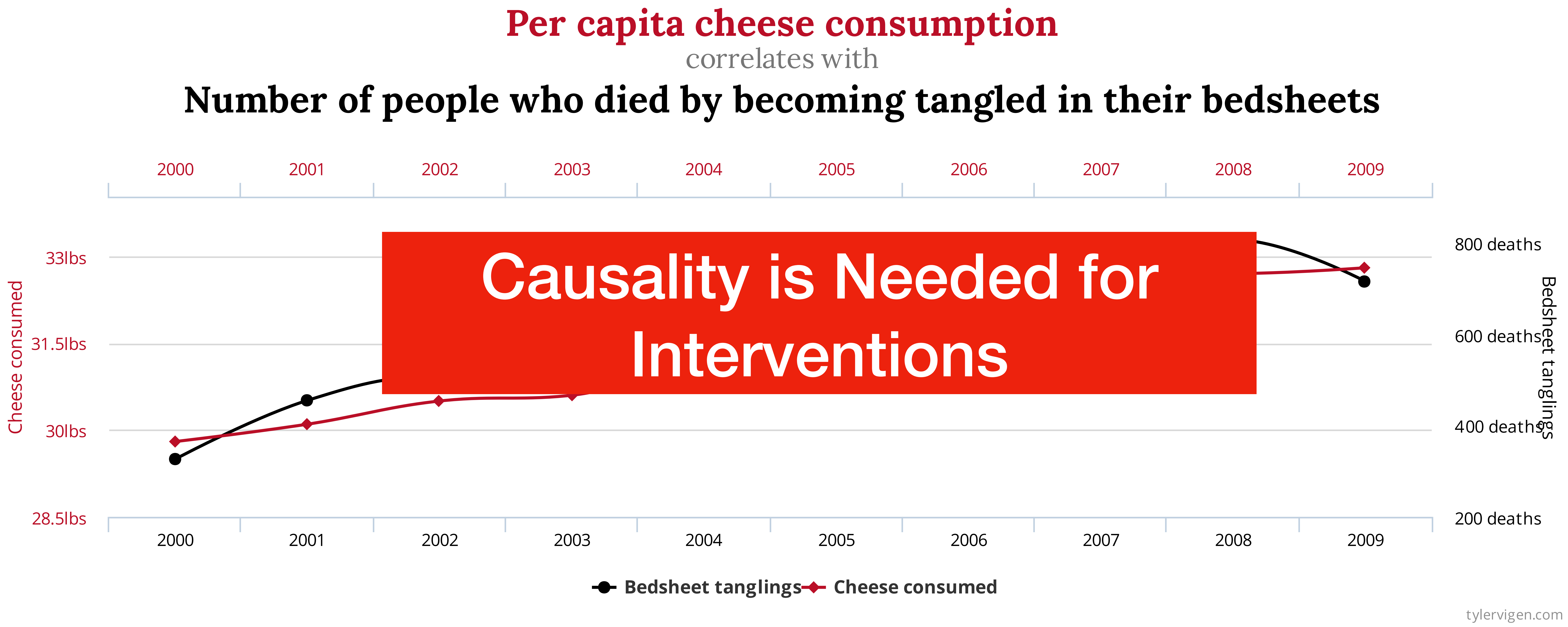
$$P_{Y|do(X=a,Z=c)} \neq P_{Y|do(X=b,Z=c)}$$

Example: Cancer, Smoking, and Genetic Factors



$$P_{C|do(S=1,G=0)} \neq P_{C|do(S=0,G=0)}$$

Correlation does not Imply Causation



Causality is Needed for Interventions

https://www.tylervigen.com/spurious-correlations

Prediction is not Causation

- Consider

$$\begin{aligned} X &\sim \text{Uniform}(0,1) \\ E_Y, E_Z &\sim \mathcal{N}(0,1) \\ Y &\leftarrow 0.5X + E_Y \\ Z &\leftarrow Y + E_Z \end{aligned}$$

- Prediction

$$\hat{Y} = 0.25X + 0.5Z$$

as a causal model suggests that Y depends on Z

Direction of prediction often indistinguishable

Correlation does not Imply Causation: A Serious Case

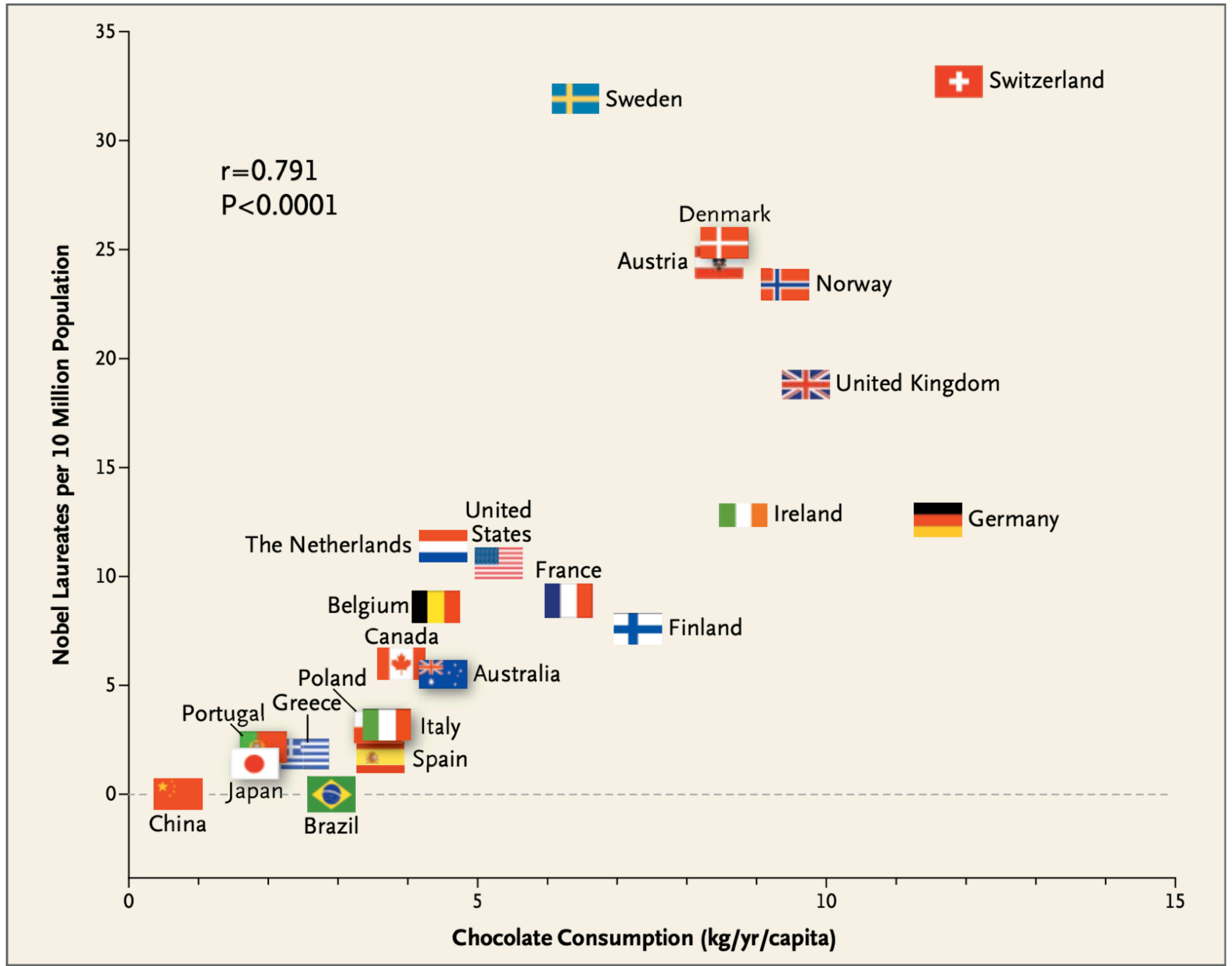
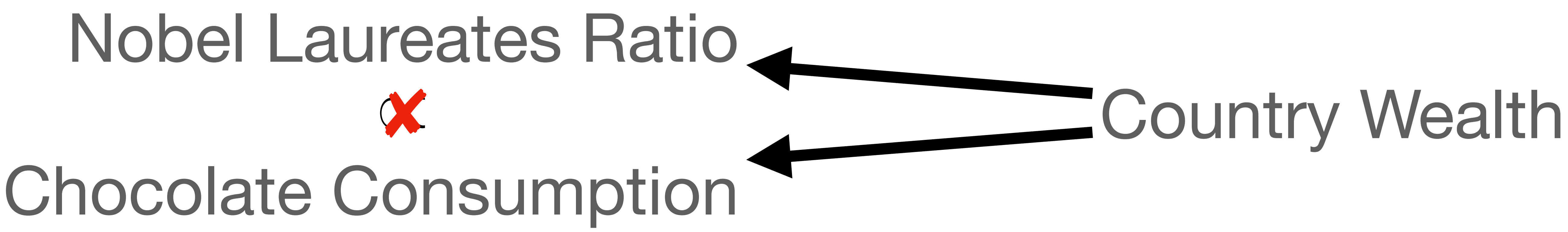


Figure 1. Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.



This means Confounders:
Variables are not Independent

$$\text{chocolate consumption} \not\perp\!\!\!\perp \text{nobel laurate ration}$$

Probable Explanation:
Variables are Independent Conditionally to Another Event

$$\text{chocolate consumption} \perp\!\!\!\perp \text{nobel laurate ration} \mid \text{country wealth}$$

Causality and Paradoxes

- If mother smokes, child is small
- Tiny child, implies health issues
- However, P(tiny child, mother smokes)>P(tiny child)

So smoking is beneficial to child's health?

Explain issues away:

- Multi-causality of children weight
- These causes *also* affect health
- Compared to these mother smoking is not that bad, but frequency of smoking?
- Conclusions Contain Social Biases: mother is always responsible (autism, etc)

Causality and Paradoxes

- If mother smokes, child is small
- Tiny child, implies health issues
- However, P(tiny child, mother smokes)>P(tiny child)

So smoking is beneficial to child's health?

Explain issues away:

  - Multi-causality of children weight
  - These causes *also* affect health
  - Compared to these mother smoking is not that bad, but frequency of smoking?
  - Conclusions Contain Social Biases: mother is always responsible (autism, etc)

Why Causality

Goals in AI

- Fair ➡Biases
- Accountable ➡explainability
- Transparent ➡Decision making can be supported
- Robust ➡attacks / manipulations

Causality Argued Advantages

- Decreased sensitivity wrt to Data
- Simulation of Interventions ➡variable clamping
- Hopes for explanation / bias detection
- Robust

Causal Discovery

How

- Gold Standard ➡Randomised Controlled Experiments
- Feasibility ➡Low in many cases, especially human
- The AI/ML Setting ➡discovery: infer model from data

What For?

- Understandable, interpretable models
- Prioritise confirmatory experiments: enable some control
- Generate new data: for simulation, privacy, medical training

Applications

- Physics
- Neuroscience
- Epidemiology
- Economy
- Climate

How do we do it?

Causal Modelling

Setting

- Assume we have the random variables

$$X_1, \ldots, X_d$$

- with a sample joint distribution

$$\mathcal{D} = \{x_i \in \Omega^d, i = 1 \ldots n\}$$

Formal Background

- Key concept
- Framework
- Approaches

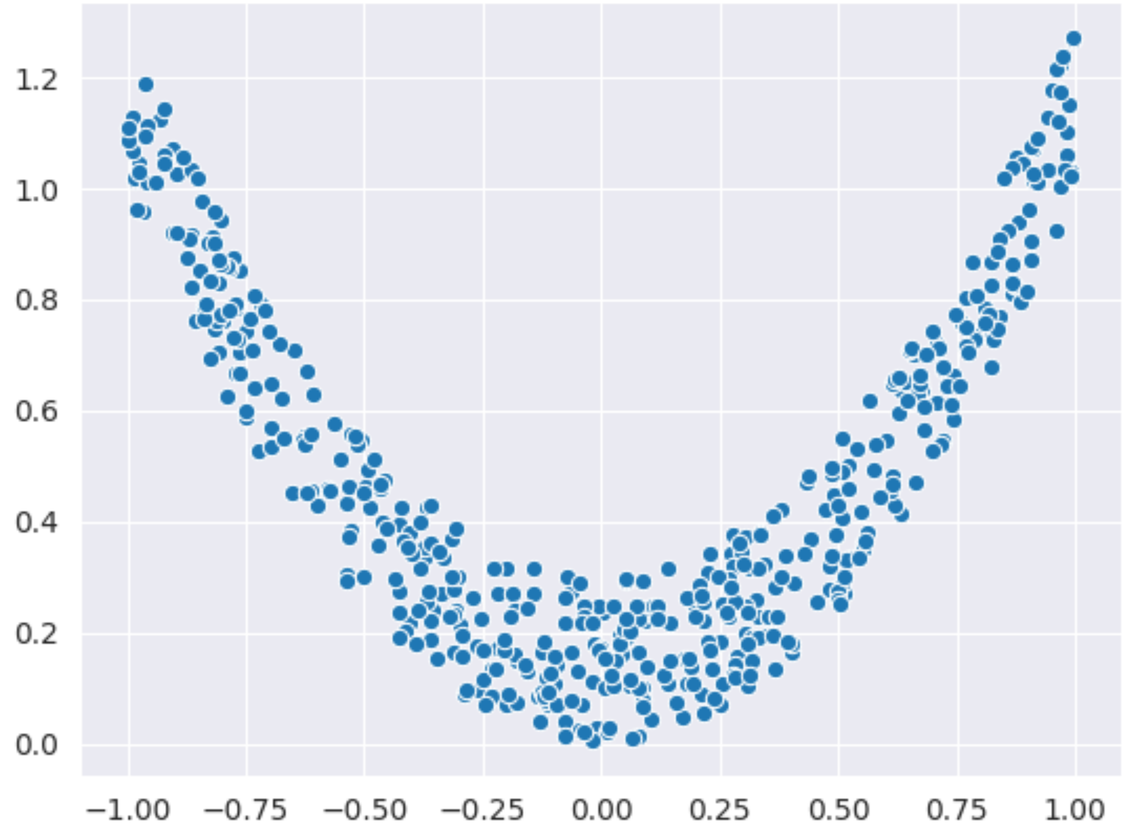
Key Concept 1: Variable (in)Dependency

- Definition of Independency

$$X \perp\!\!\!\perp Y \leftrightarrow P(X,Y) = P(X)P(Y)$$

- How do we test for independency?
  Correlation? It only works for first order linear dependencies

$$Y = X^2 + \epsilon \rightarrow \text{correlation}(X,Y) \simeq 0$$



Key Concept 1: Variable (in)Dependency

- Definition of Independency

$$X \perp\!\!\!\perp Y \leftrightarrow P(X,Y) = P(X)P(Y)$$

- How do we test for independency?
  Different tests:
  - Correlation $Y = X^2 + \epsilon \rightarrow \mathrm{correlation}(X,Y) \simeq 0$
  - HSIC, Hilbert-Schmitt Independence Criterion (Gretton et al 05)

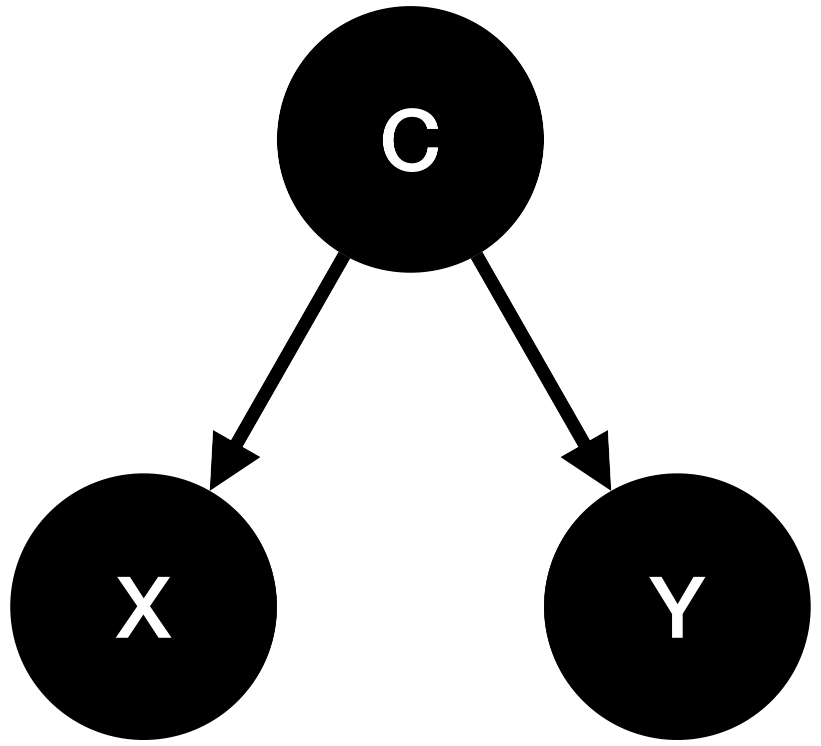
    $$\mathrm{HSIC}(Pr_{XY}), \mathcal{F}, \mathcal{G}) \triangleq \|C_{XY}\|^2_{HS}$$

    where $\|C_{XY}\|^2_{HS}$ is the Hilbert-Schmitt norm of the kernel correlation matrix and $\mathcal{F}, \mathcal{G}$ are two kernels: i.e. it’s the kernel trick for correlation.

Key Concept 2: Conditional (in)Dependency

- Definition of Conditional Independency

$$X \perp\!\!\!\perp Y | C \leftrightarrow P(X, Y | C) = P(X|C)P(Y|C)$$



- C=rains, X=wet sidewalk, Y=people with umbrellas

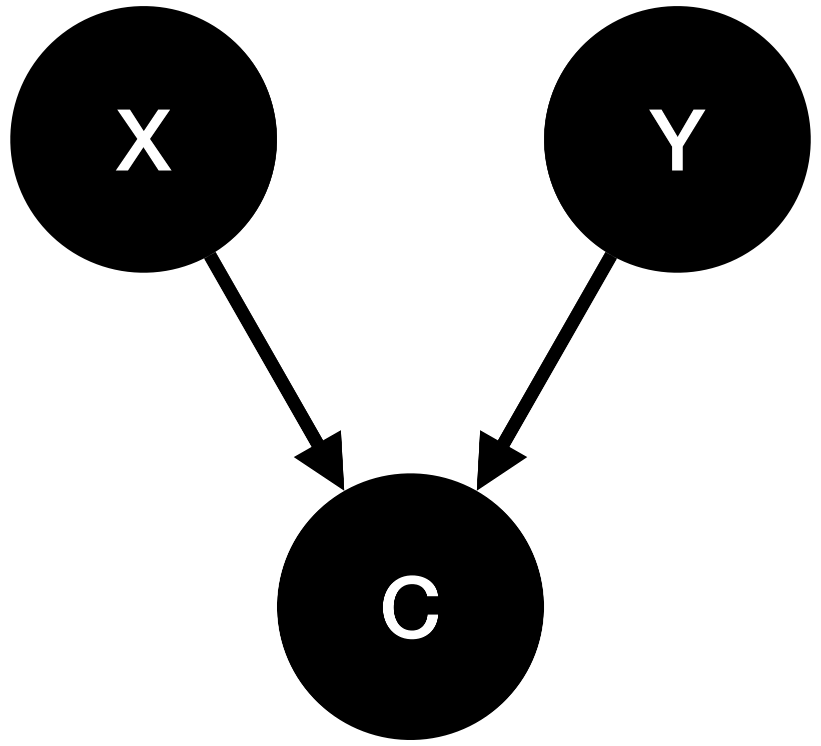
- Definition of Conditional Dependency

$$P(C|X, Y) \neq P(C|X)P(C|Y)$$

$$X \not\perp\!\!\!\perp Y | C = 1 \leftrightarrow$$

$$P(X, Y) = P(X)P(Y)$$

$$P(X, Y | C = 1) \neq P(X|C = 1)P(Y|C = 1)$$



- X=Complex Machine, Y=Inexperienced worker, C=Accident

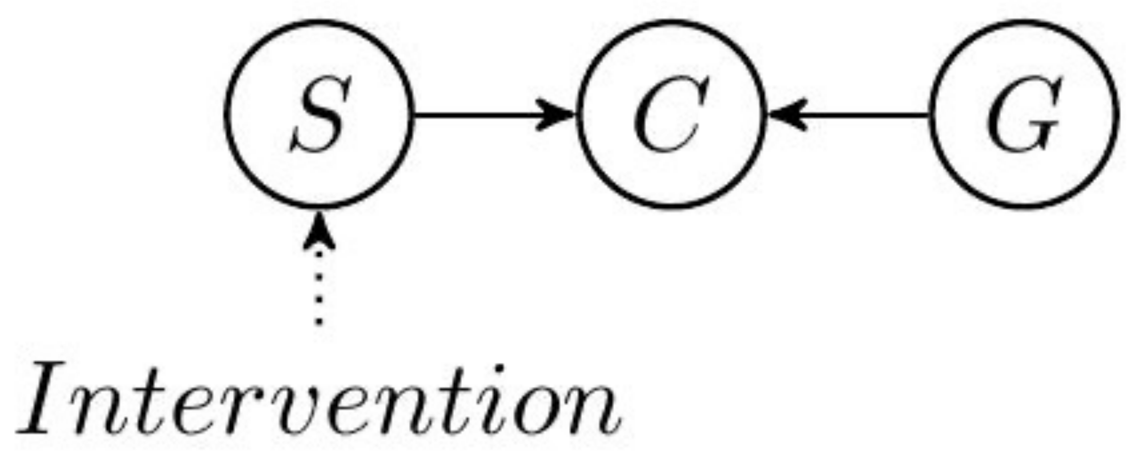
Definition of Causal Relationship

X is a direct cause of Y if when we intervene it Y's law changes

$$X \to Y \quad \text{iif}$$

$$P_{Y|do(X=a,Z=c)} \neq P_{Y|do(X=b,Z=c)}$$

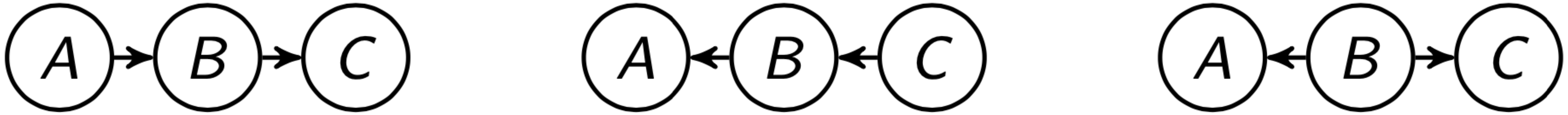
Example: Cancer, Smoking, and Genetic Factors



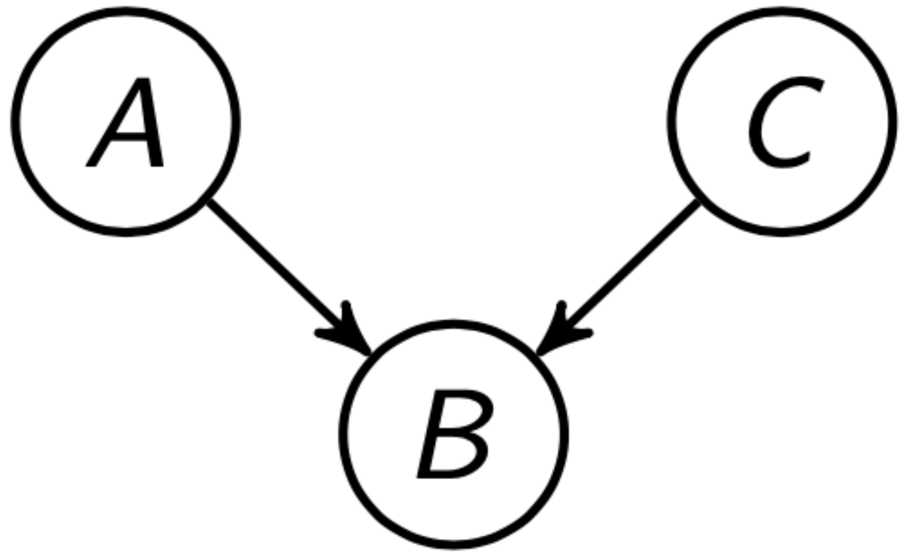
$$P_{C|do(S=1,G=0)} \neq P_{C|do(S=0,G=0)}$$

Markov Equivalences

Markov Equivalent Class: $A \perp\!\!\!\perp C | B \text{ and } A \not\perp\!\!\!\perp C$



V-Structure: $A \not\perp\!\!\!\perp C | B \text{ and } A \perp\!\!\!\perp C$

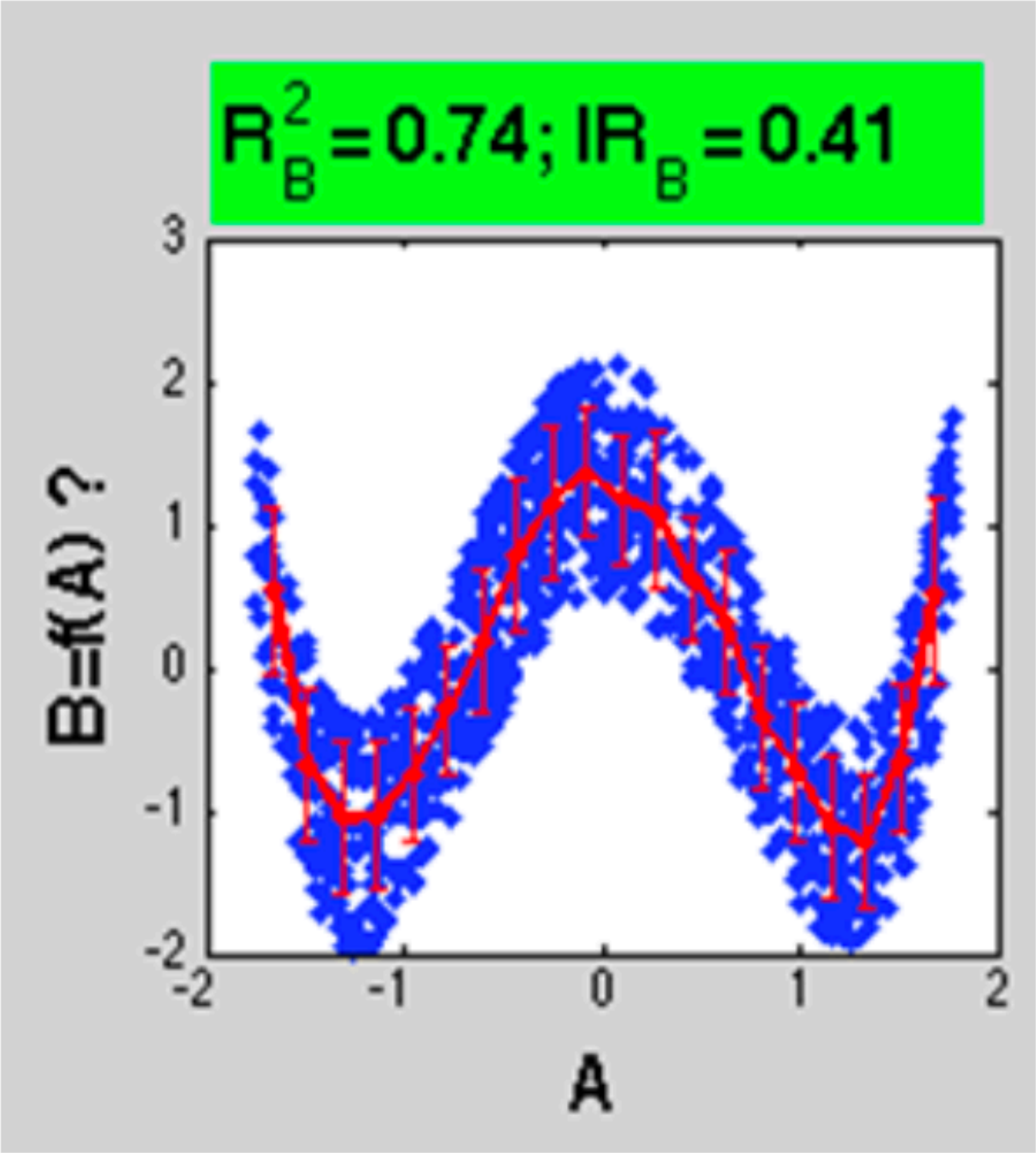


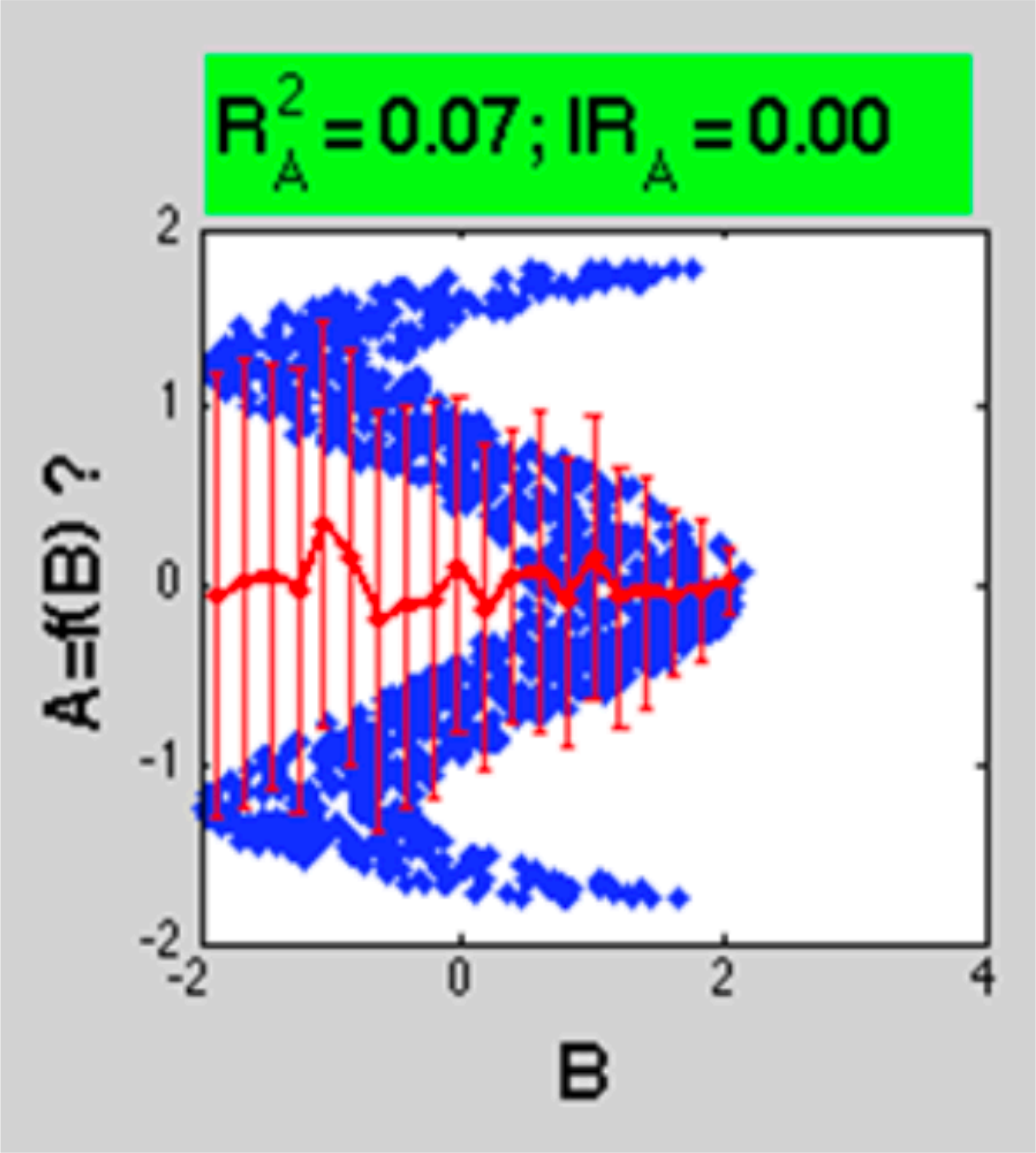
Sprites et al 00, 16

Key Concept 3: Causality with Distributional Assymetry

- Leverages Occam's Principle

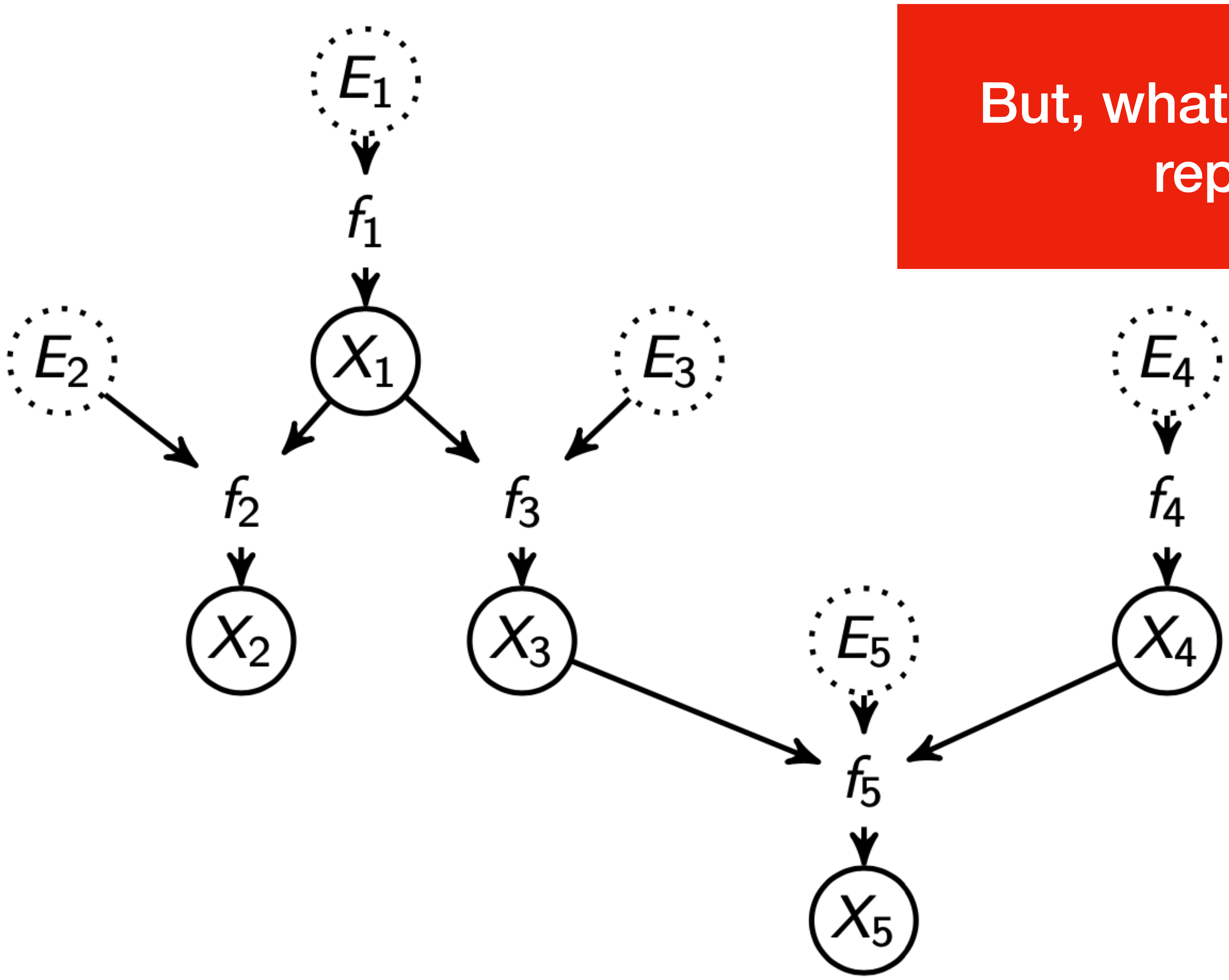
  The causal model as the simplest explaining the data (Janzig 19)





Framework: Functional Causal Models (FCMs)

- Given $X_1, \ldots, X_d$ where $X_i = f_i(X_{Pa(X_i)}, E_i)$, with $X_{Pa(X_i)}$ the parents or causes of X_i, a deterministic function f_i, and E_i an error representing independent random variable.



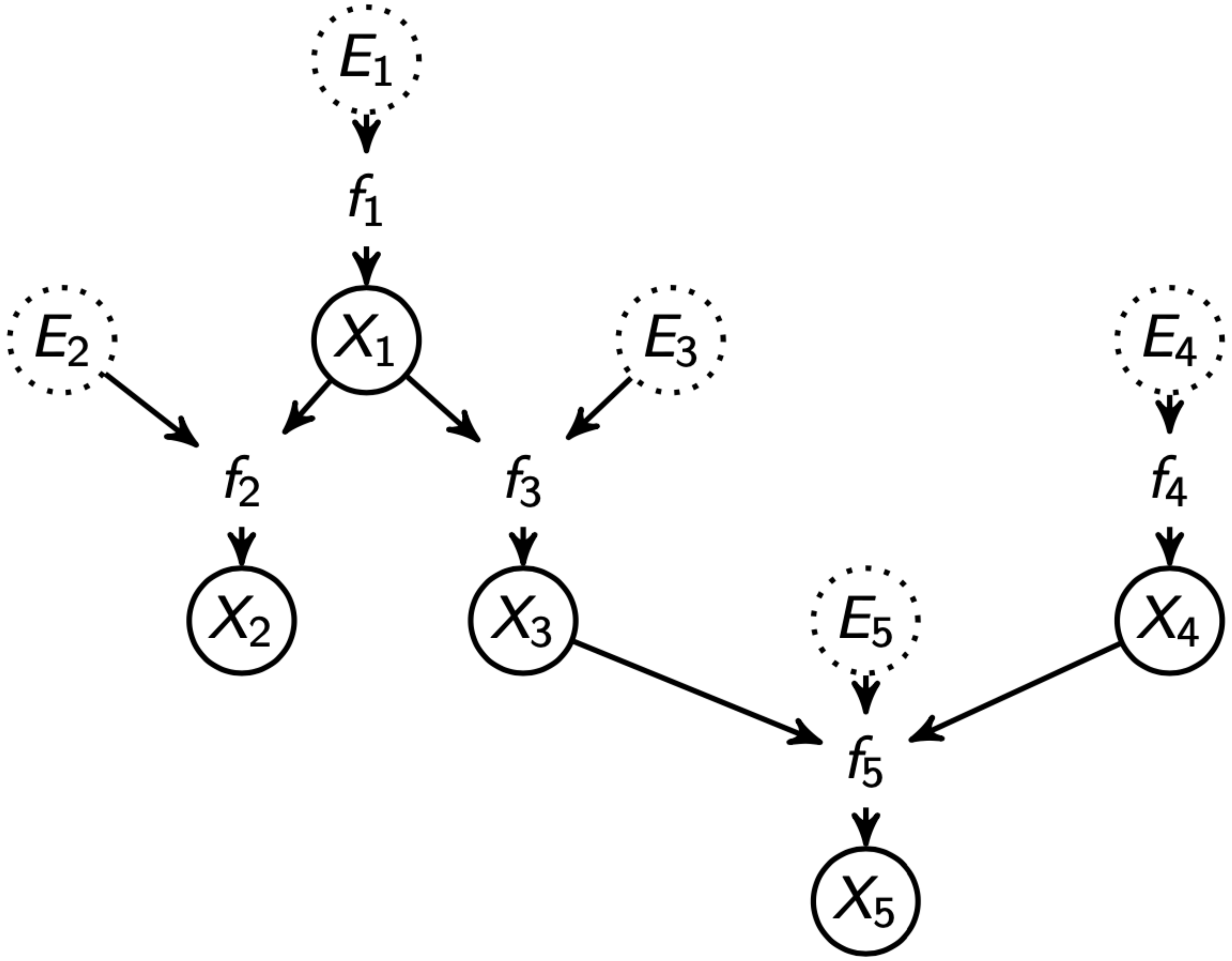
But, what do we need for this system to represent a causal model?

$$\begin{cases} X_1 = f_1(E_1) \\ X_2 = f_2(X_1, E_2) \\ X_3 = f_3(X_1, E_3) \\ X_4 = f_4(E_4) \\ X_5 = f_5(X_3, X_4, E_5) \end{cases}$$

$$P(X_1, \ldots, X_d) = \Pi P(X_i | X_{Pa(X_i)})$$

Conditions for Causal Model Representation

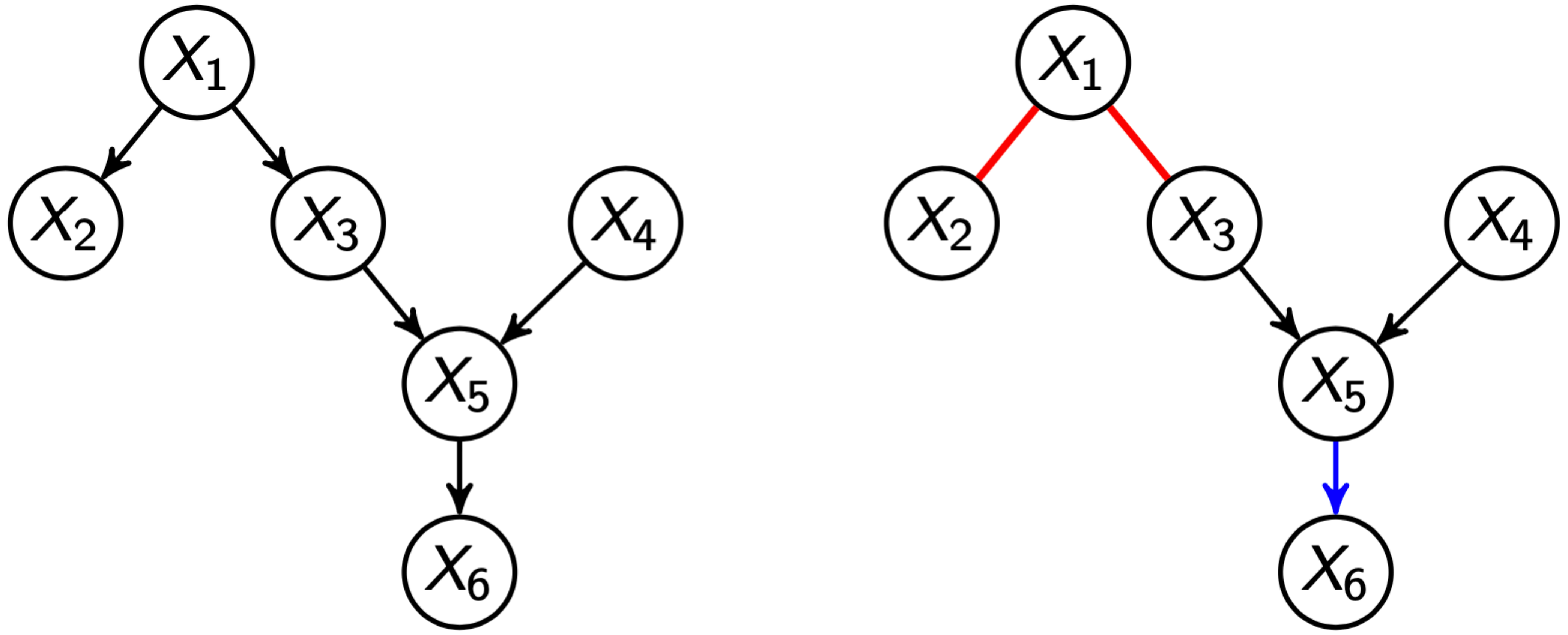
- Causal Sufficiency: no unobserved confounders
- Causal Markov: all d-separations in the causal graph G imply conditional independencies in the observational distribution P
- Causal Faithfulness: all conditional independencies in P imply d-separations in the causal graph G



How Do We Infer the Causal Model From Data?

Key Approach 1: Constraint-Based Methods

- Constraint-based methods, through V-Structures and constraint propagation, output a CPDAG (Completed Partially Directed Acyclic Graph).



(a) The exact DAG of $\mathcal{G}$.

(b) The CPDAG of $\mathcal{G}$.

- Examples: Peter-Clark Algorithm (PC) and it's extensions such as PC-Hist (Spires et al 00, Zhang et al 12)

Key Approach 2: Score-Based

- Use an objective function to optimise the graph. For instance the Bayesian information criterion

$$BIC(\mathcal{G}) = -2\ln(L) + k\ln(n)$$

- with L the likelihood of the model, k number of parameters, and n the number of samples
- We optimise the sample with operations such as:
    - Add an edge
    - remove an edge
    - revert and dee
- An algorithm for this are Greedy Equivalence Search (GES) by Chickering et al 02.

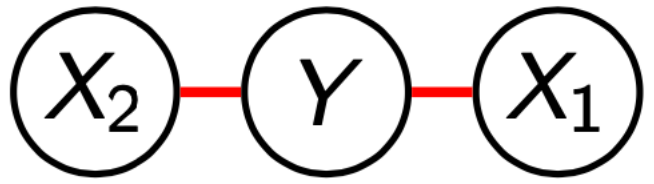
Key Approaches 1 and 2

- Limitations
  - Computational cost depending on the test/scoring/loss
  - Data hungry
  - Identifiability issues
  - Example:

$$X_1, E_{X_1}, E_{X_2} \sim U(0,1) X_1 \perp\!\!\!\perp E_{X_1}, Y \perp\!\!\!\perp E_{X_2}$$

$$Y \leftarrow 0.5X_1 + E_{X_1}$$

$$X_2 \leftarrow Y + E_{X_2}$$



$X_1 \perp\!\!\!\perp X_2|Y$. No V-struture

Key Approach 3: Global Optimisation

- Assuming linear causal mechanisms, the system can be formulated in terms of linear equations

  $$X = B^T X + E$$

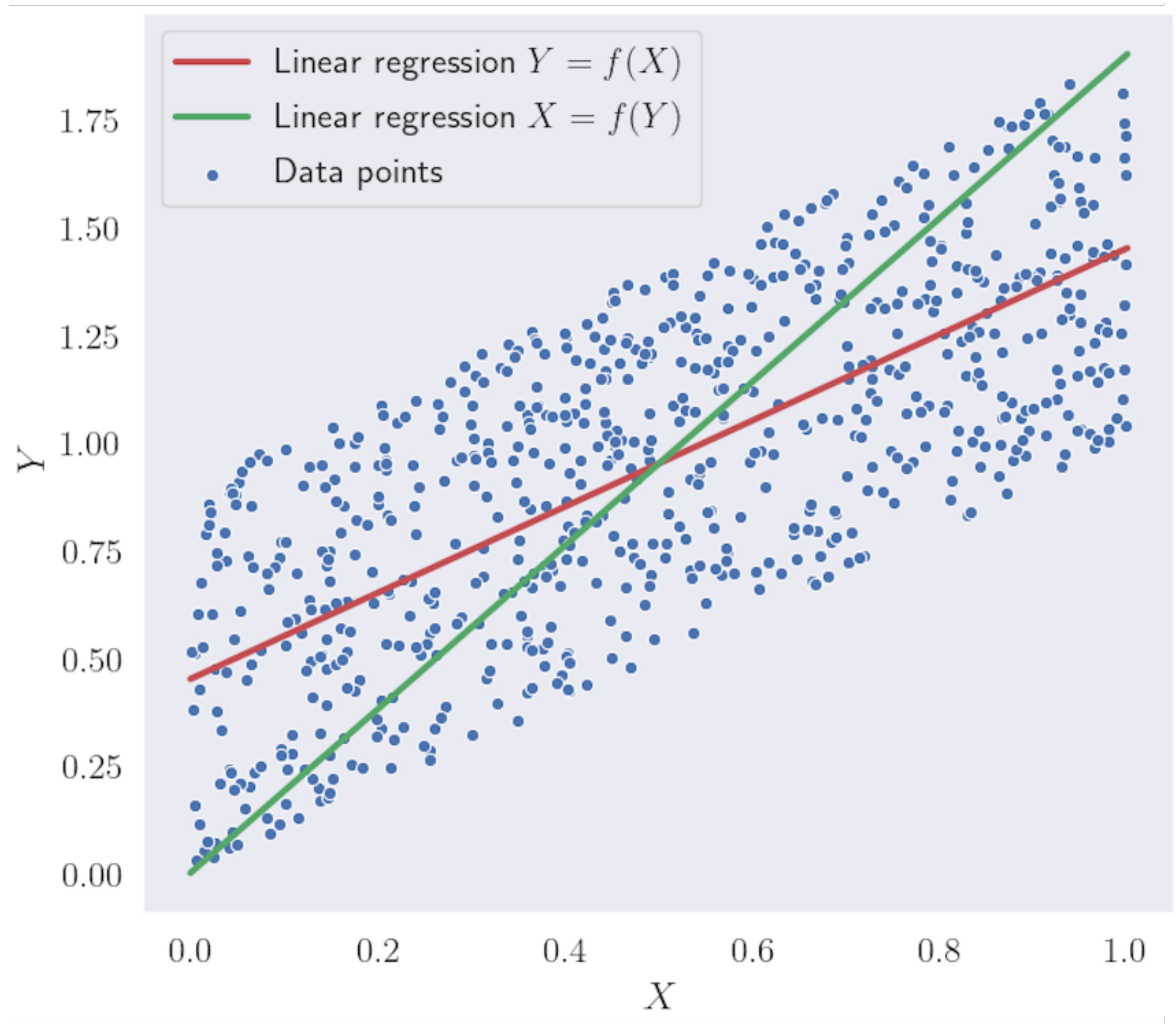
  where the triangular B matrix can be estimated through ICA for LinGAM (Shimizu 06, Hyvarien 99)

- This also can be done in terms of graphical models (Pearl 09, Friedman 08)

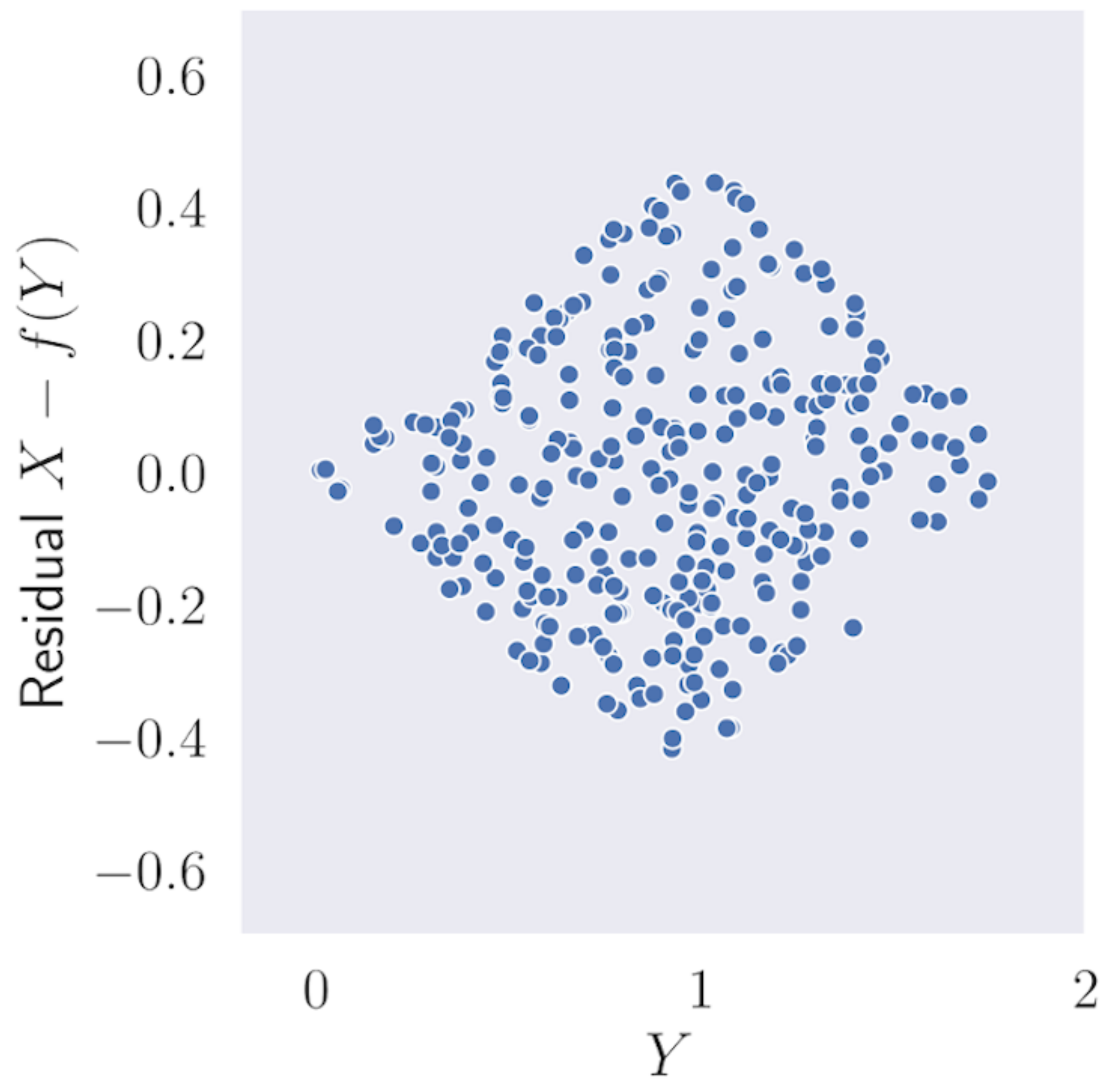
  For instance with Max-Min Hill-Climbing (MMHC) by Tsamardinos (06) and concave penalised Descent (CCDr) by Aragam (15)

Key Approach 4: Exploiting Asymmetries

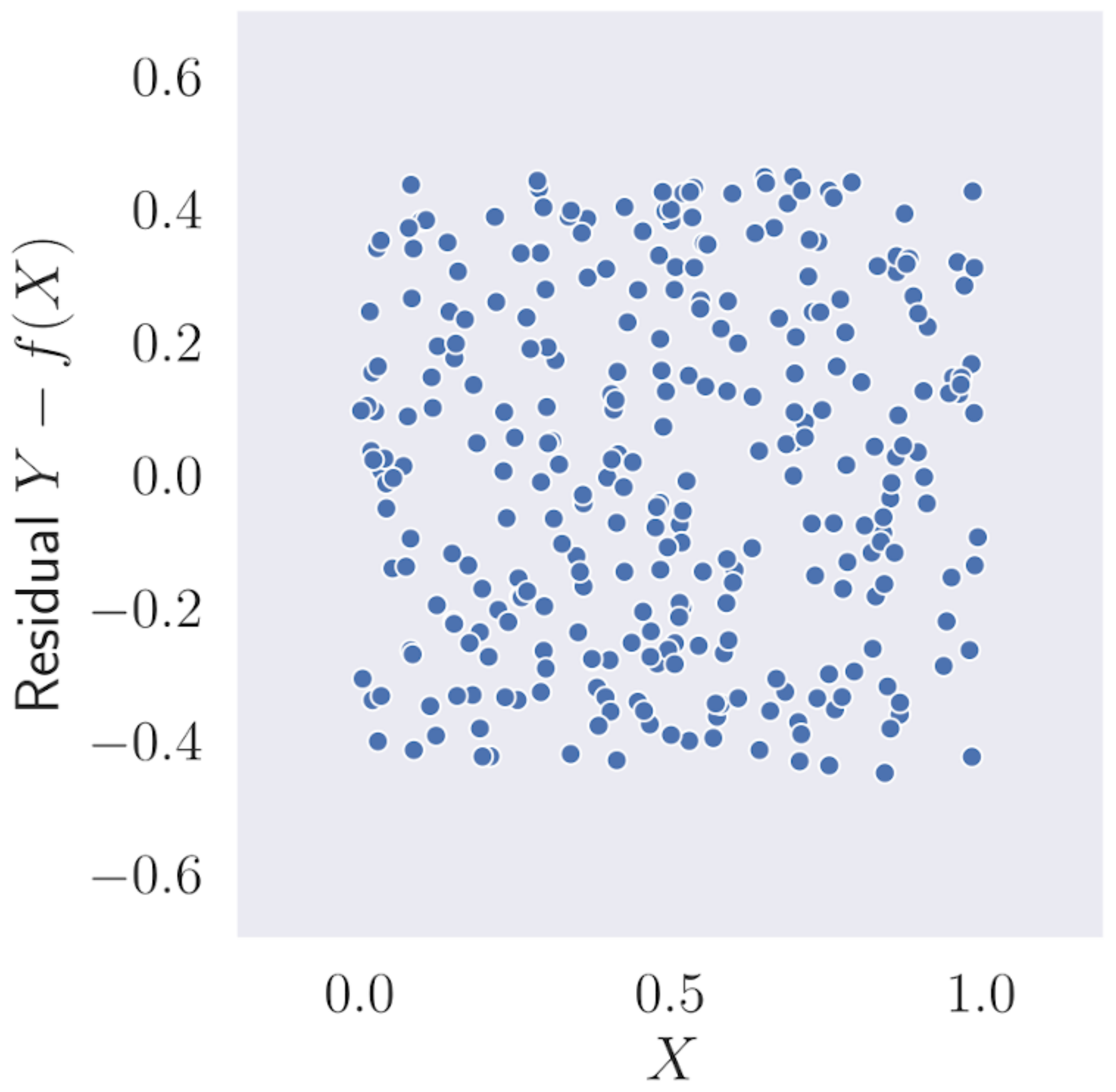
- If no v-structure is available and causal discovery with 2 variables is hard, we can leverage asymmetries in the distributions . For instance with the Additive Noise Model (ANM) of Hoyer (09)



Original data



Residuals of X=g(Y)



Residuals of Y=f(X)

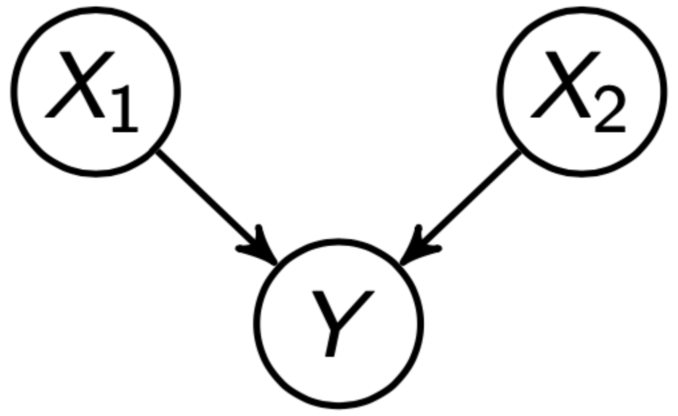
Key Approach 4: Exploiting Asymmetries

Limitations

- Restrictive assumptions on the type of causal mechanisms
- Conditional independence is not taken into account

$$X_1, E_{X_1}, X_2 \sim \mathcal{N}(0,1) X_1 \perp\!\!\!\perp E_{X_1}, Y \perp\!\!\!\perp E_{X_1}$$

$$Y \leftarrow 0.5 X_1 + X_2 + E_1$$

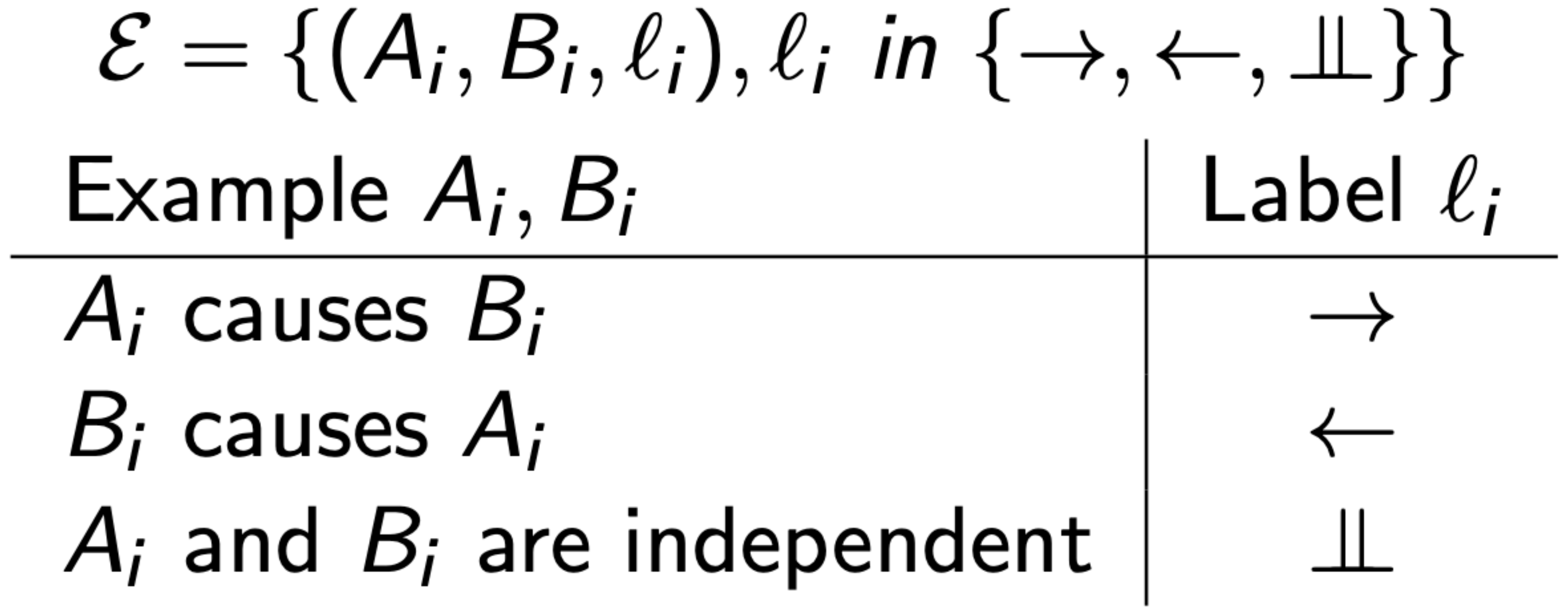


(X1,Y) and (X2,Y) are a perfectly symmetric pairwise distribution after rescaling. However, $X_1 \not\perp\!\!\!\perp X_2 | Y$ a v-structure is at the origin of the data.

Key Approach 5: Machine Learning Base

Guyon et al 2014—2015

- Pair Cause-Effect Challenges
  - Gather data: a sample is a pair of variables (Ai,Bi)
  - Its label ℓ_i is the “true” causal relation (e.g. age “causes” salary)
- Input



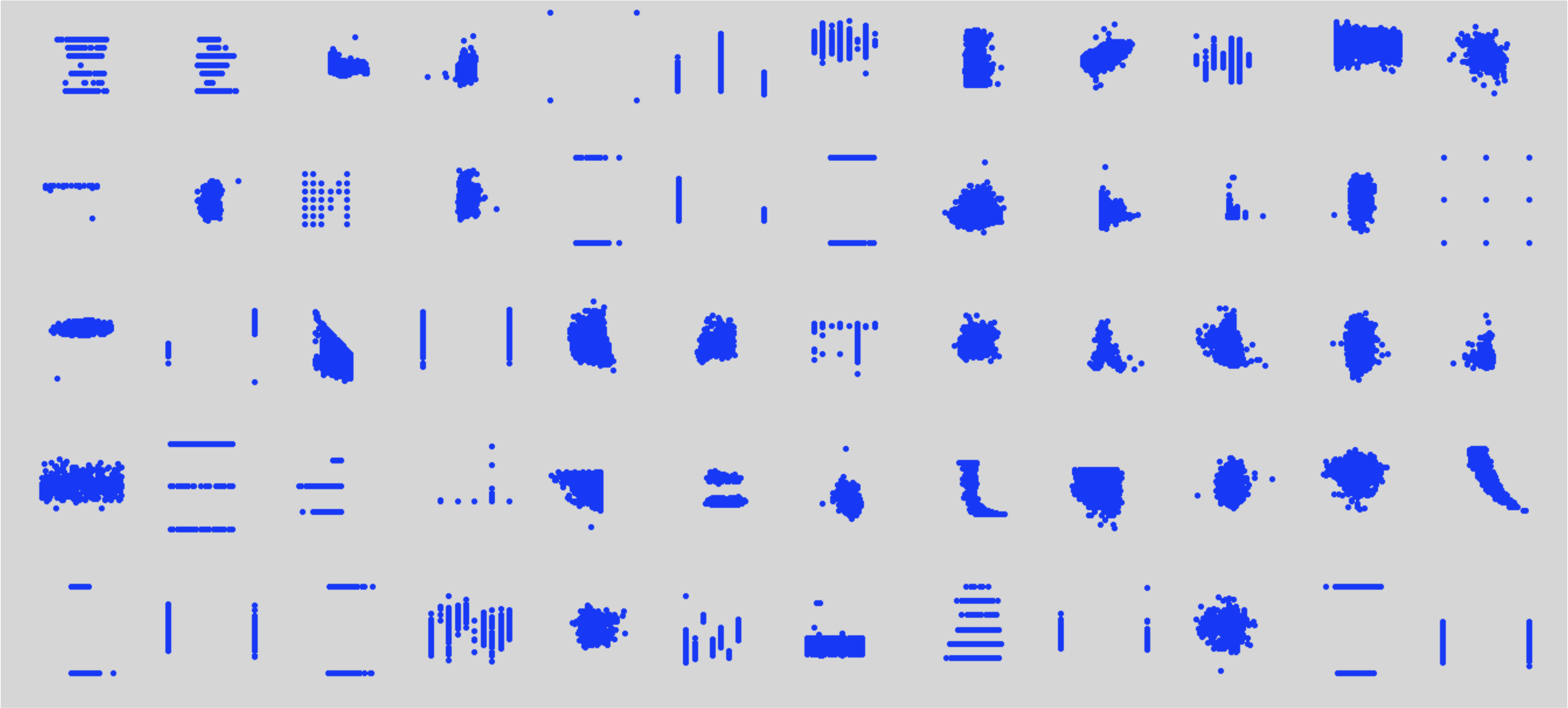
$$\mathcal{E} = \{(A_i, B_i, \ell_i), \ell_i \ in \ \{\rightarrow, \leftarrow, \perp\!\!\!\perp\}\}$$

Example A_i, B_i	Label ℓ_i
A_i causes B_i	$\rightarrow$
B_i causes A_i	$\leftarrow$
A_i and B_i are independent	$\perp\!\!\!\perp$

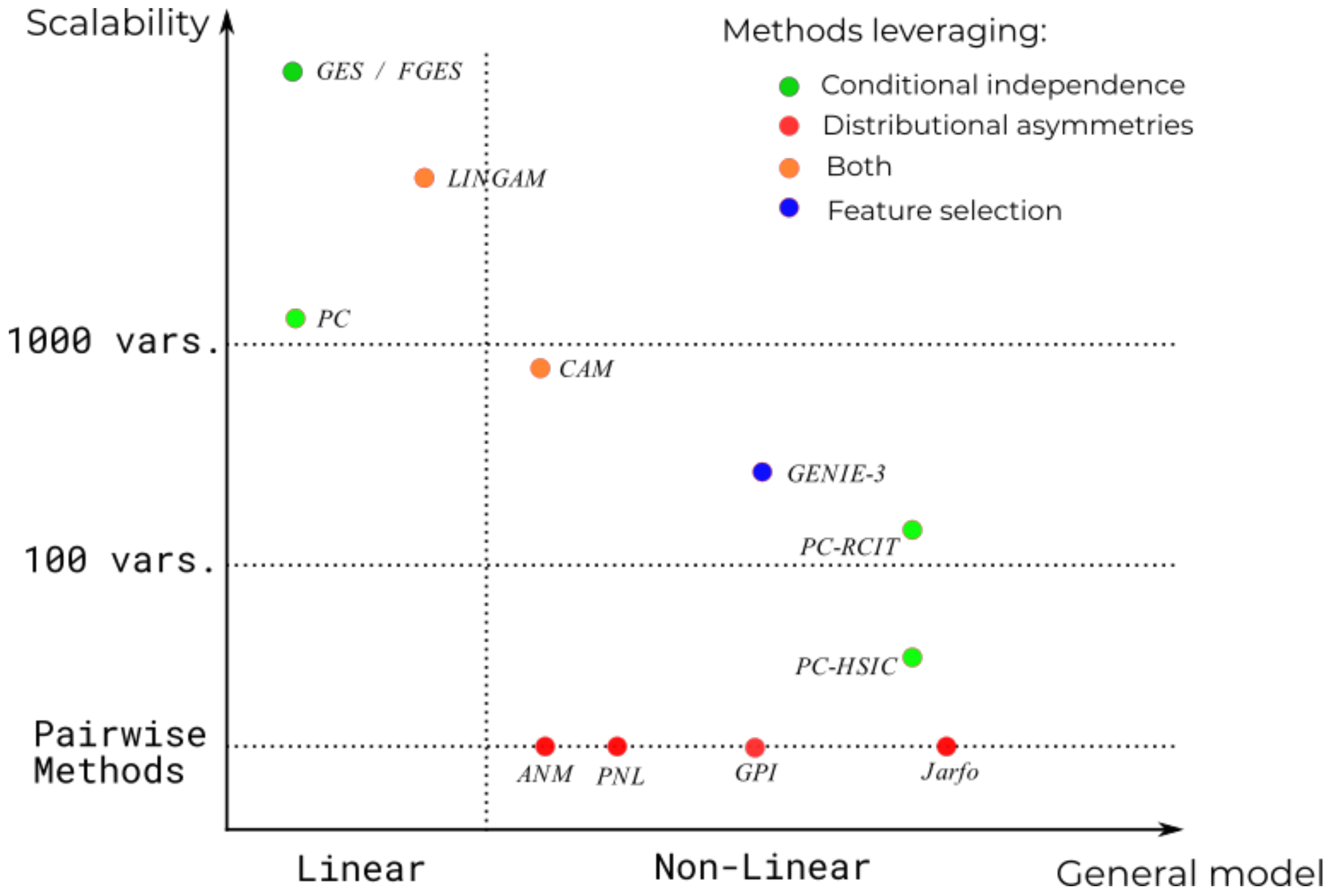
- Output: $(A, B) \rightarrow \ell$

Key Approach 5: Machine Learning Base

Guyon et al 2014—2015



Summary for “Key Approaches”



A Python Package for Causal Discovery

All the presented framework is available on GitHub at :

https://github.com/Diviyan-Kalainathan/CausalDiscoveryToolbox

It includes multiple algorithms as well as tools for graph structure.

Published in Kalainathan Goudet 2019 JMLR - Open Source Software

A Python Package for Causal Discovery

All the presented framework is available on GitHub at :

https://github.com/Diviyan-Kalainathan/CausalDiscoveryToolbox

It includes multiple algorithms as well as tools for graph structure.

Published in Kalainathan Goudet 2019 JMLR - Open Source Software

Simulation Based Inference

Simulation-Based Inference

The setting

- Assume that we have a generative (graphical and parametrical) model to produce the data. Can we train an inference system such that given a dataset we can obtain the parameters?

  More formally, given:

  $$\text{latent variables} \quad z \sim p(z|\theta)$$

  $$\text{simulated dataset} \quad x \sim p(x|\theta, z)$$

  Can we train a system to infer a density

  $$q(\theta|x)$$

Simulation-Based Inference

Current Approaches (Cranmer et al 2019)

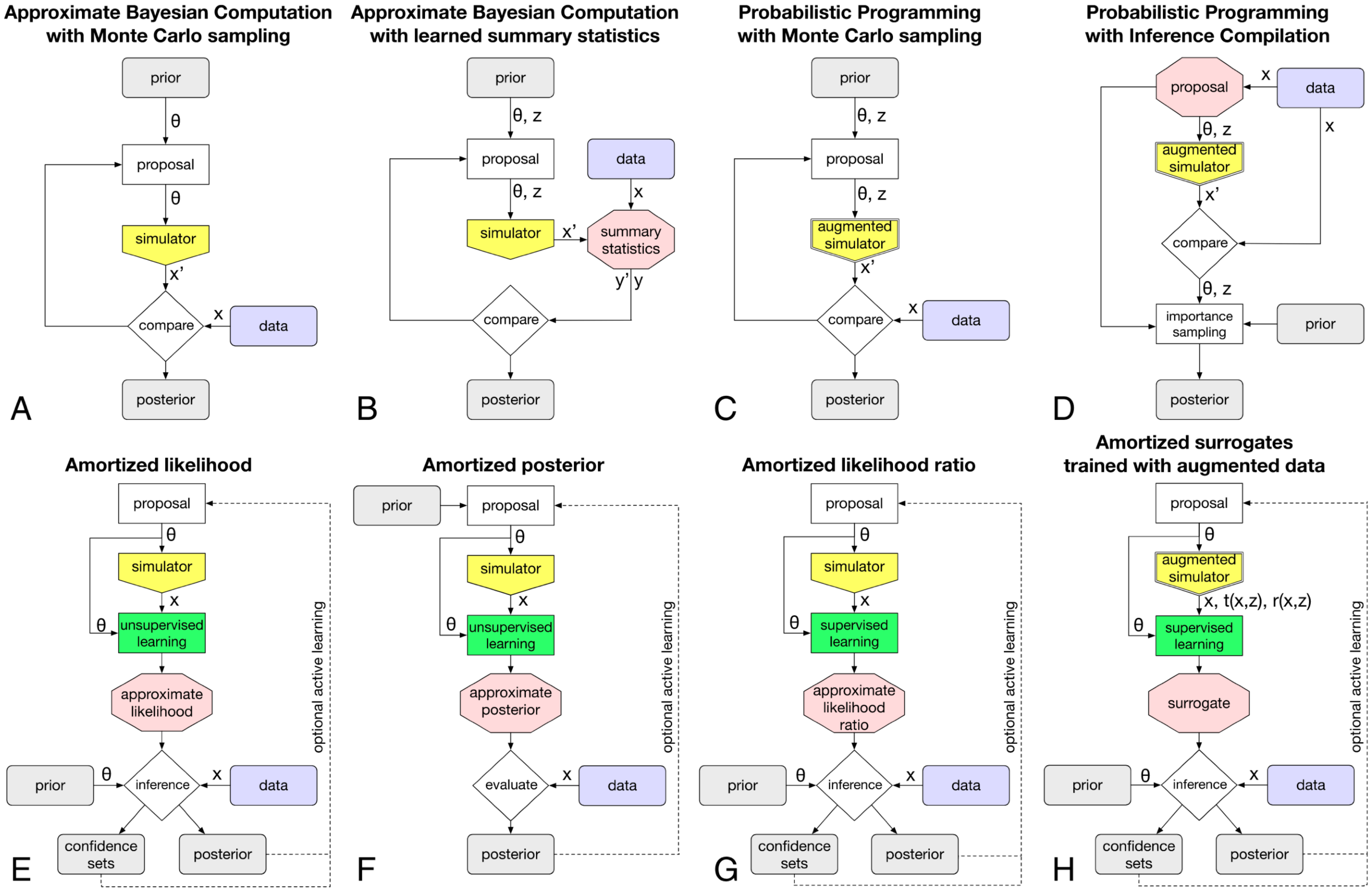


Fig. 1. (*A–H*) Overview of different approaches to simulation-based inference.

A Use Case Combining Graphical Models with Simulation-Based Inference in Neuroscience

Slides kindly provided by Louis Rouillard, Inria, Saclay Île-de-France
work to be published in ICLR 2022

Part 1

Problem statement

Experimental & Theoretical

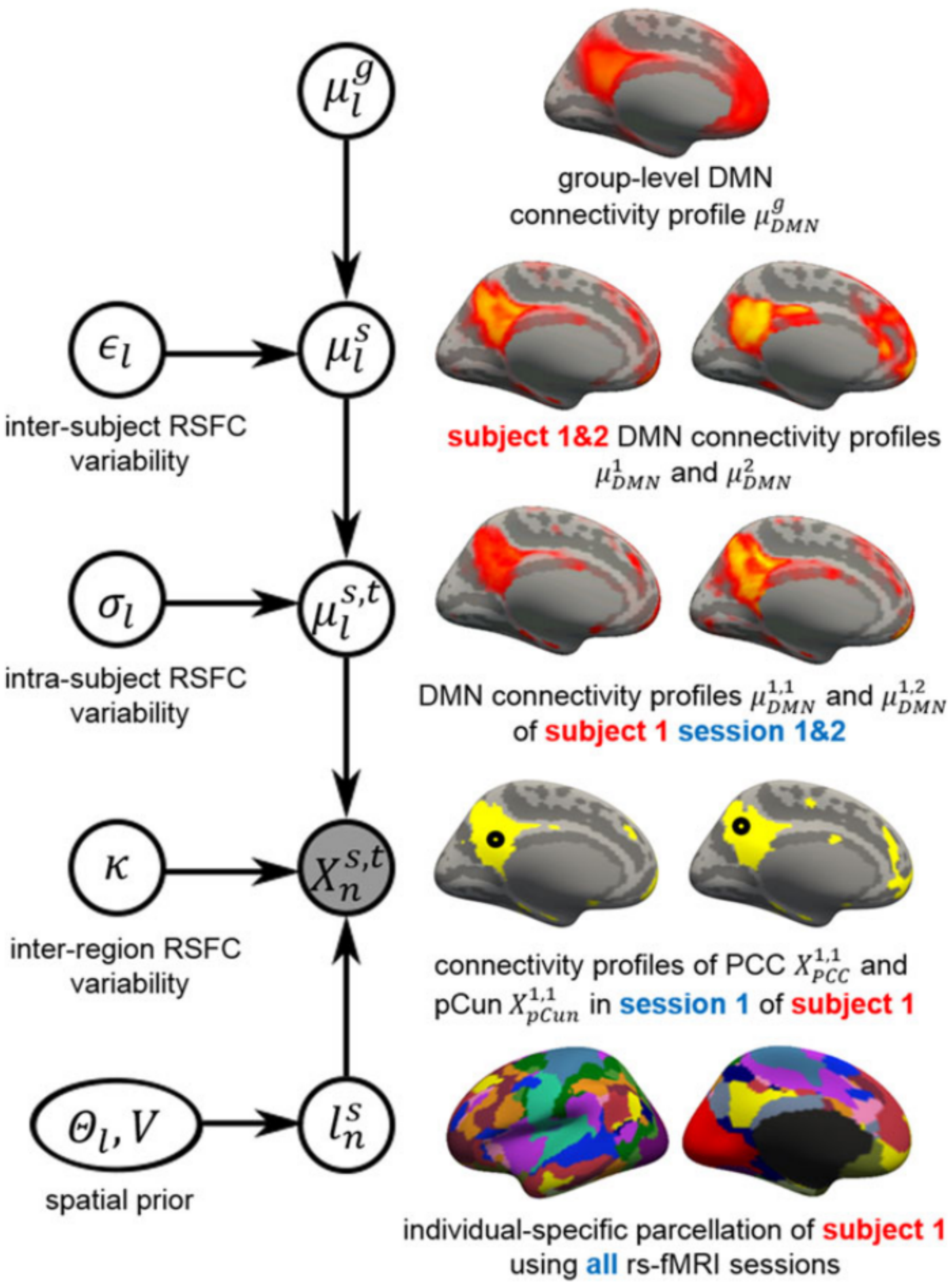
Pyramidal experimental setups

Kong et al. 2018 - MS-HBM

Functional connectivity modelled via a **Hierarchical Bayesian Model** (HBM)

Connectivity with **several scales** for variability:

- Multiple subjects
- Multiple measurement sessions per subject
- Multiple brain vertices per session



μ_l^g
group-level DMN connectivity profile μ_{DMN}^g
ϵ_l
inter-subject RSFC variability
μ_l^s
subject 1&2 DMN connectivity profiles μ_{DMN}^1 and μ_{DMN}^2
σ_l
intra-subject RSFC variability
$\mu_l^{s,t}$
DMN connectivity profiles $\mu_{DMN}^{1,1}$ and $\mu_{DMN}^{1,2}$ of subject 1 session 1&2
κ
inter-region RSFC variability
$X_n^{s,t}$
connectivity profiles of PCC $X_{PCC}^{1,1}$ and pCun $X_{pCun}^{1,1}$ in session 1 of subject 1
Θ_l, V
spatial prior
l_n^s
individual-specific parcellation of subject 1 using all rs-fMRI sessions

Kong et al. 2018

Inference in HBMs

- Latent parameters θ (for instance subject-level functional networks)
- Observed data X (for instance vertices connectivity in a given session)

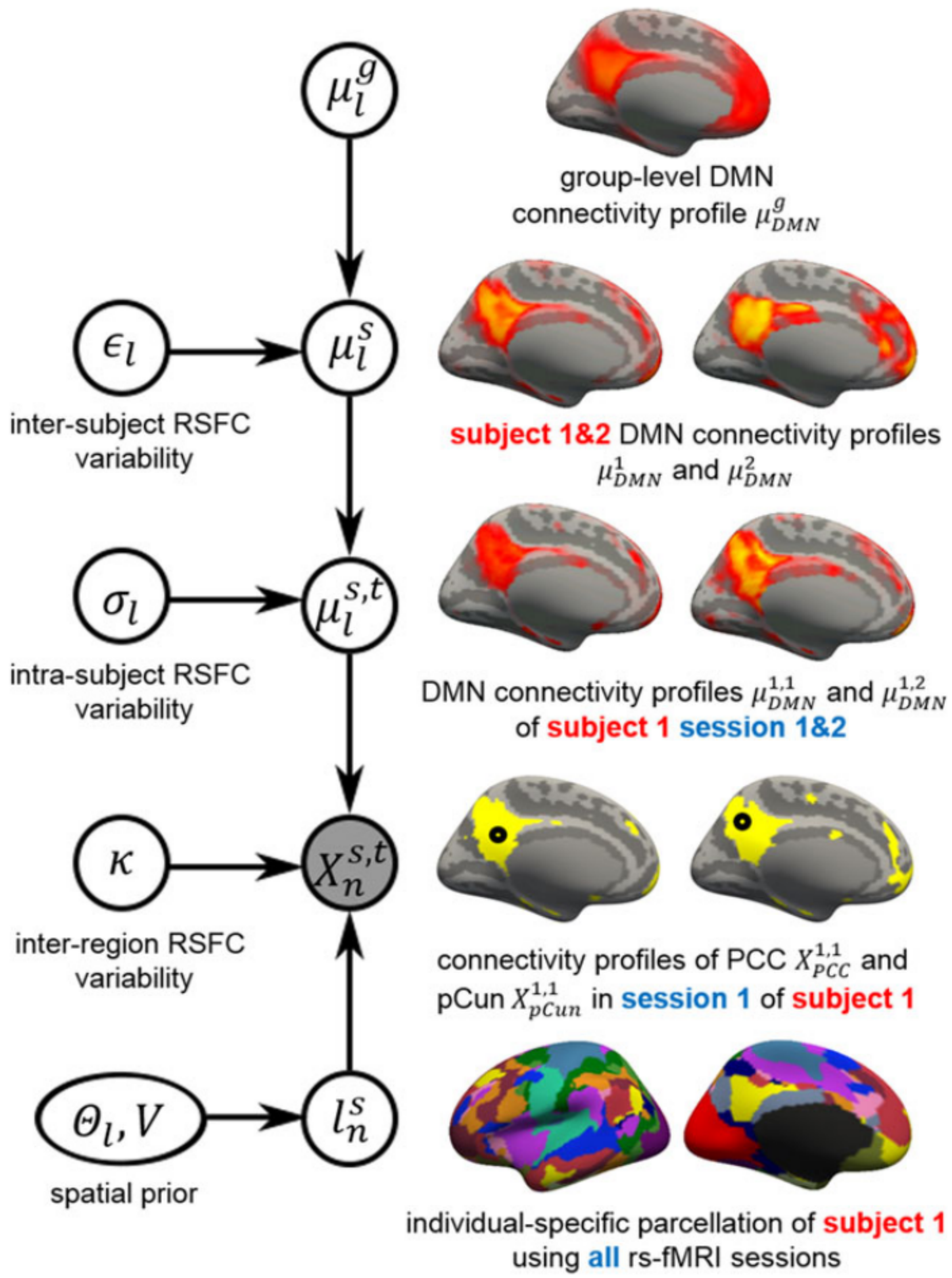
The generative Hierarchical Bayesian Model defines the joint probability:

$$p(X, \theta) = p(X \mid \theta) \times p(\theta)$$

Our goal is to obtain the **posterior distribution**:

$$p(\theta \mid X)$$

Inference can be **amortized**: once a training overhead has been paid for, we want to obtain the posterior distribution of θ given any data point X



μ_l^g
group-level DMN connectivity profile μ_{DMN}^g
ϵ_l
inter-subject RSFC variability
μ_l^s
subject 1&2 DMN connectivity profiles μ_{DMN}^1 and μ_{DMN}^2
σ_l
intra-subject RSFC variability
$\mu_l^{s,t}$
DMN connectivity profiles $\mu_{DMN}^{1,1}$ and $\mu_{DMN}^{1,2}$ of subject 1 session 1&2
κ
inter-region RSFC variability
$X_n^{s,t}$
connectivity profiles of PCC $X_{PCC}^{1,1}$ and pCun $X_{pCun}^{1,1}$ in session 1 of subject 1
Θ_l, V
spatial prior
l_n^s
individual-specific parcellation of subject 1 using all rs-fMRI sessions

Kong et al. 2018

Variational Inference (VI)

A popular inference framework (*Blei et al. 2017*)

Posits the inference problem as an **optimization**: we consider a **variational family** and look in this family for the function “closest” to our target:

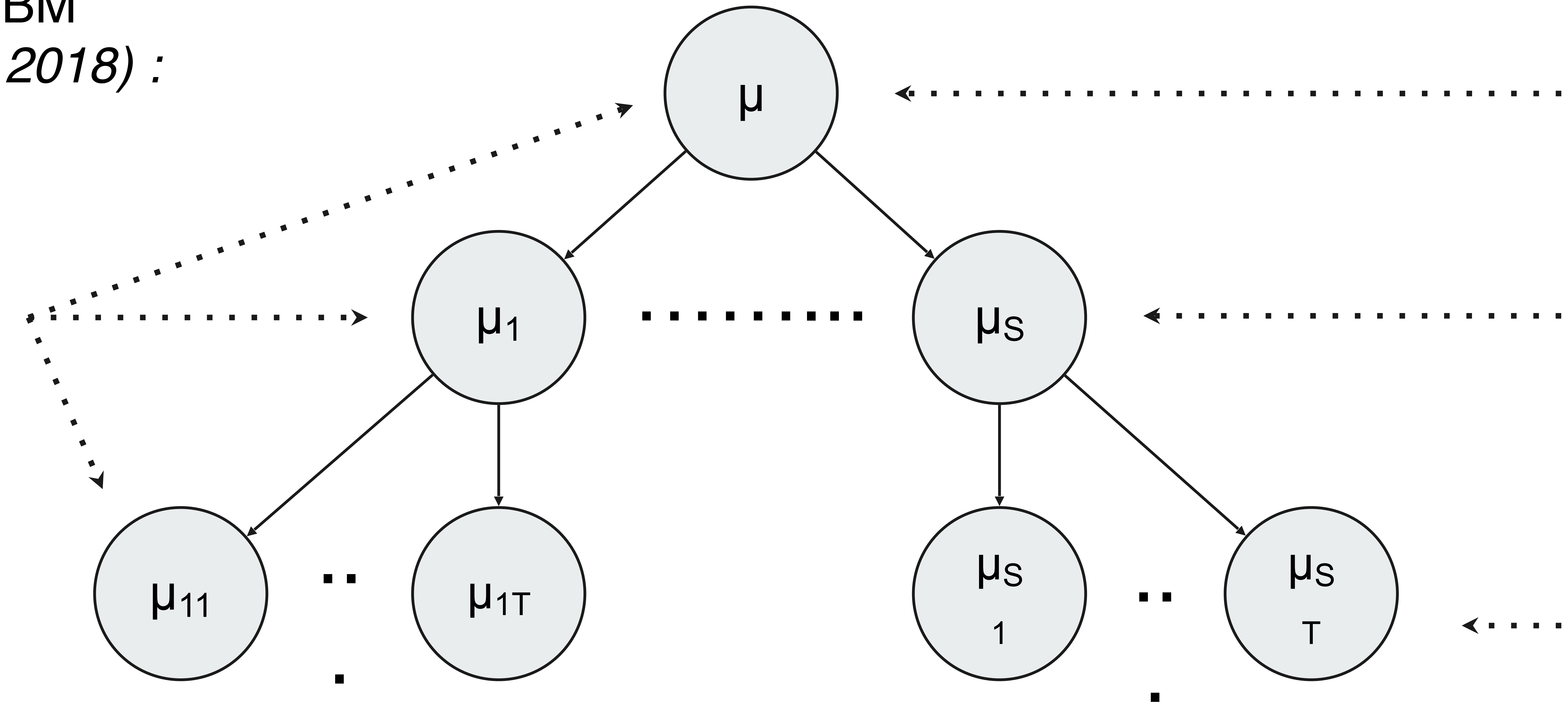
$$q \in \mathcal{Q} \ / \ q(\theta) \approx p(\theta \mid X)$$

VI now leverages **automatic differentiation** in modern ML frameworks to look for the optimal function (*ADVI Kucukelbir et al. 2016*)

Structured VI aims at exploiting the forward model’s structure to improve even further the variational family (*ASVI Ambrogioni et al. 2021, Weilbach et al. 2020, CF Ambrogioni et al. 2021*)

A massive dimensionality for the ground HBM

In the MS-HBM
(*Kong et al. 2018*) :



connectivity D
= 1483

population network

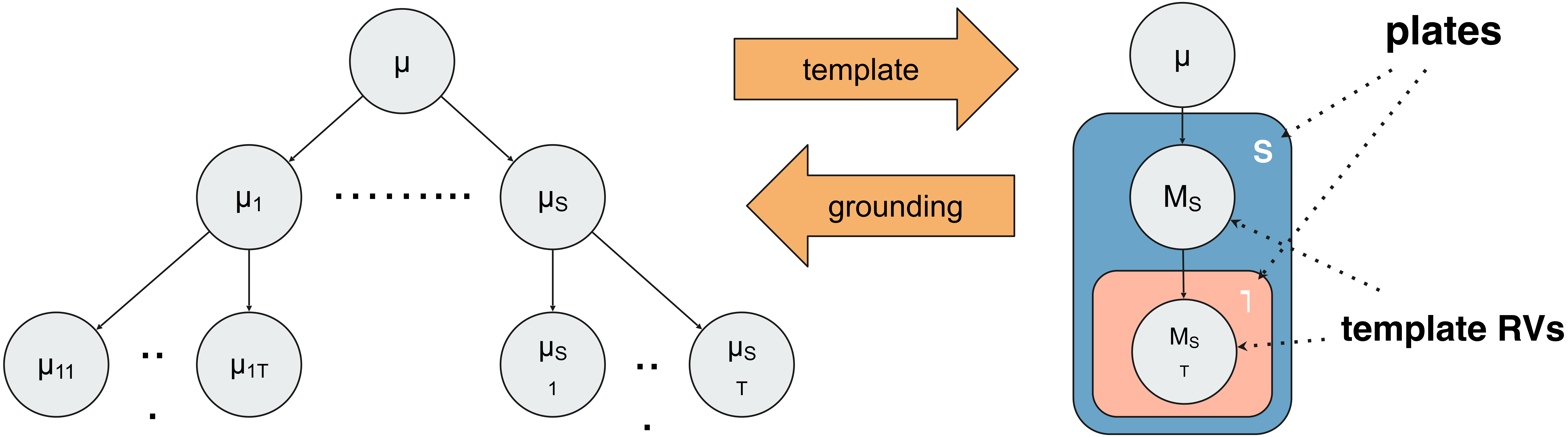
subject network
S=40

session network
T=4
(for every subject)

Total number of parameters: $\mathcal{O}(STD)$
→ ~ 5 millions !
→ prohibits traditional methods

A synthetic *template* HBM

See *Koller et Friedman (2009)*



template

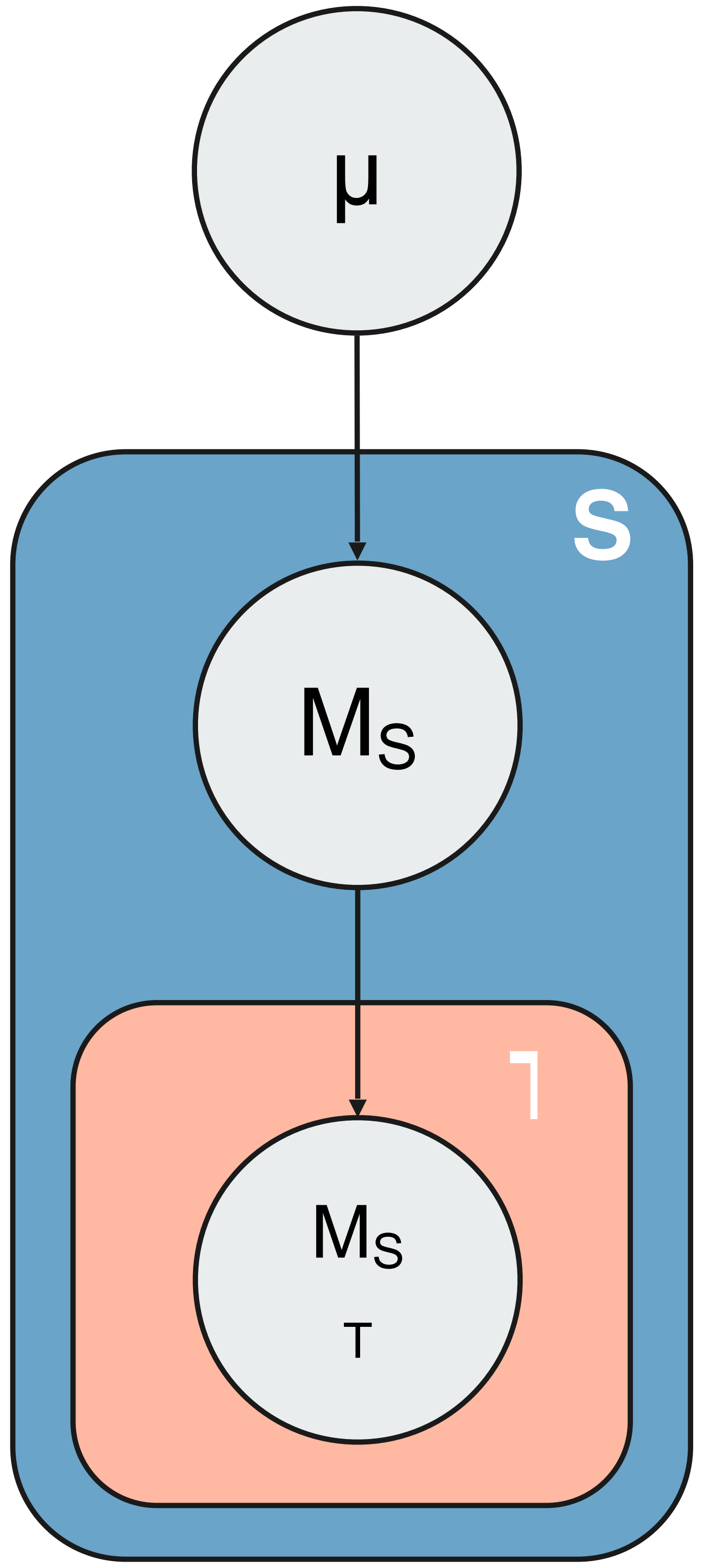
grounding

plates

template RVs

ADAVI: structured VI exploiting plates

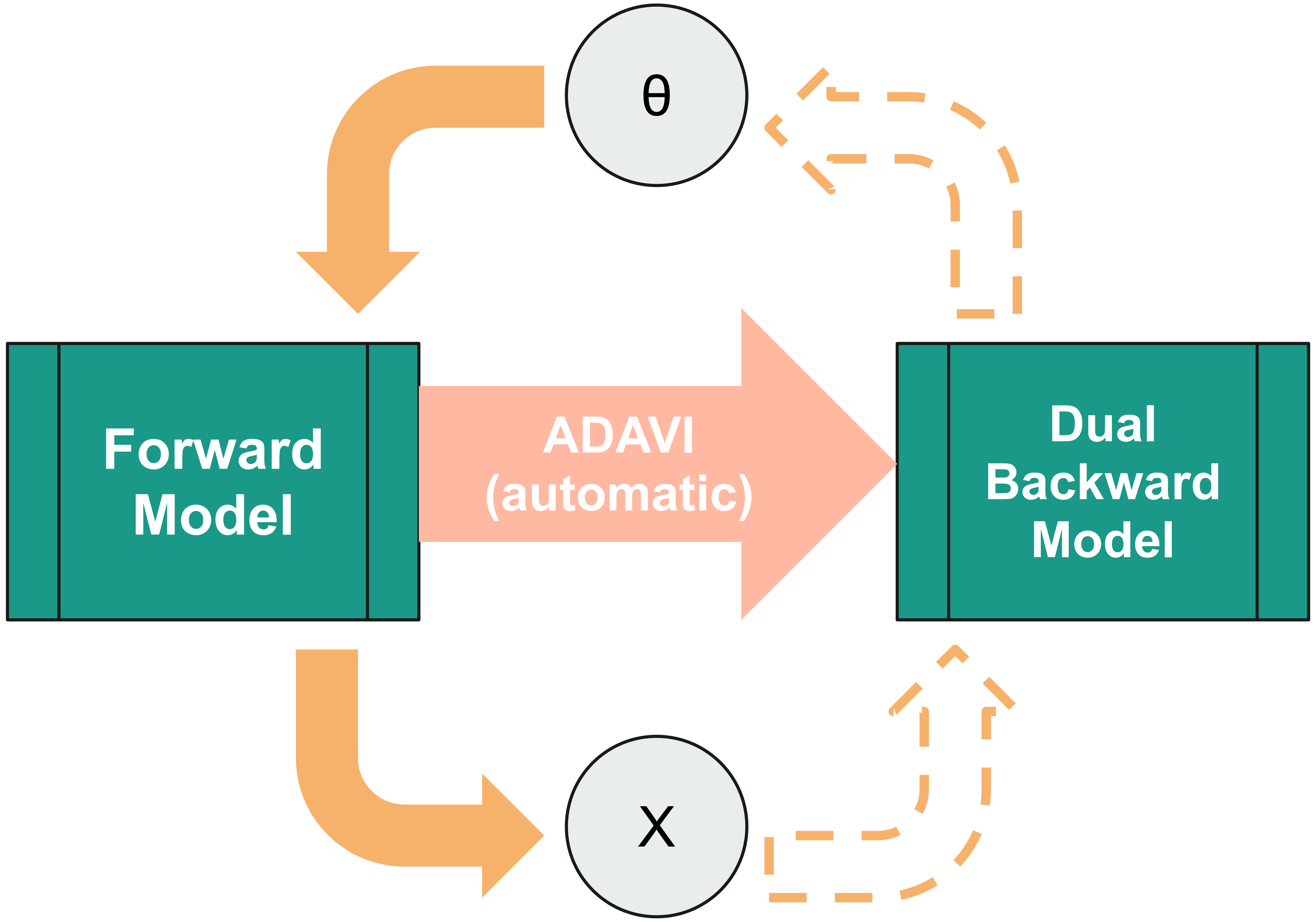
- Plates translate i.i.d sampling from a common distribution: there is a strong **symmetry** in the forward HBM (several identical sub-graphs in the ground graph)
- ADAVI's main idea is to **exploit that symmetry** to reduce the variational family's **number of parameters** (and improve its performance)
- We want to scale our parametrization over the dimensionality of the graph template and NOT the ground graph



Breaking down the acronym

ADAVI:

- **Automatic**: the variational family is derived directly from the forward HBM
- **Dual**: a backward model is constructed that goes from data X to parameters θ
- **Amortized**: once trained, the posterior is available for every data point X
- **VI**: we use optimization to derive the variational posterior



Part 2

Methodological overview

Subpart A: pyramidal HBMs

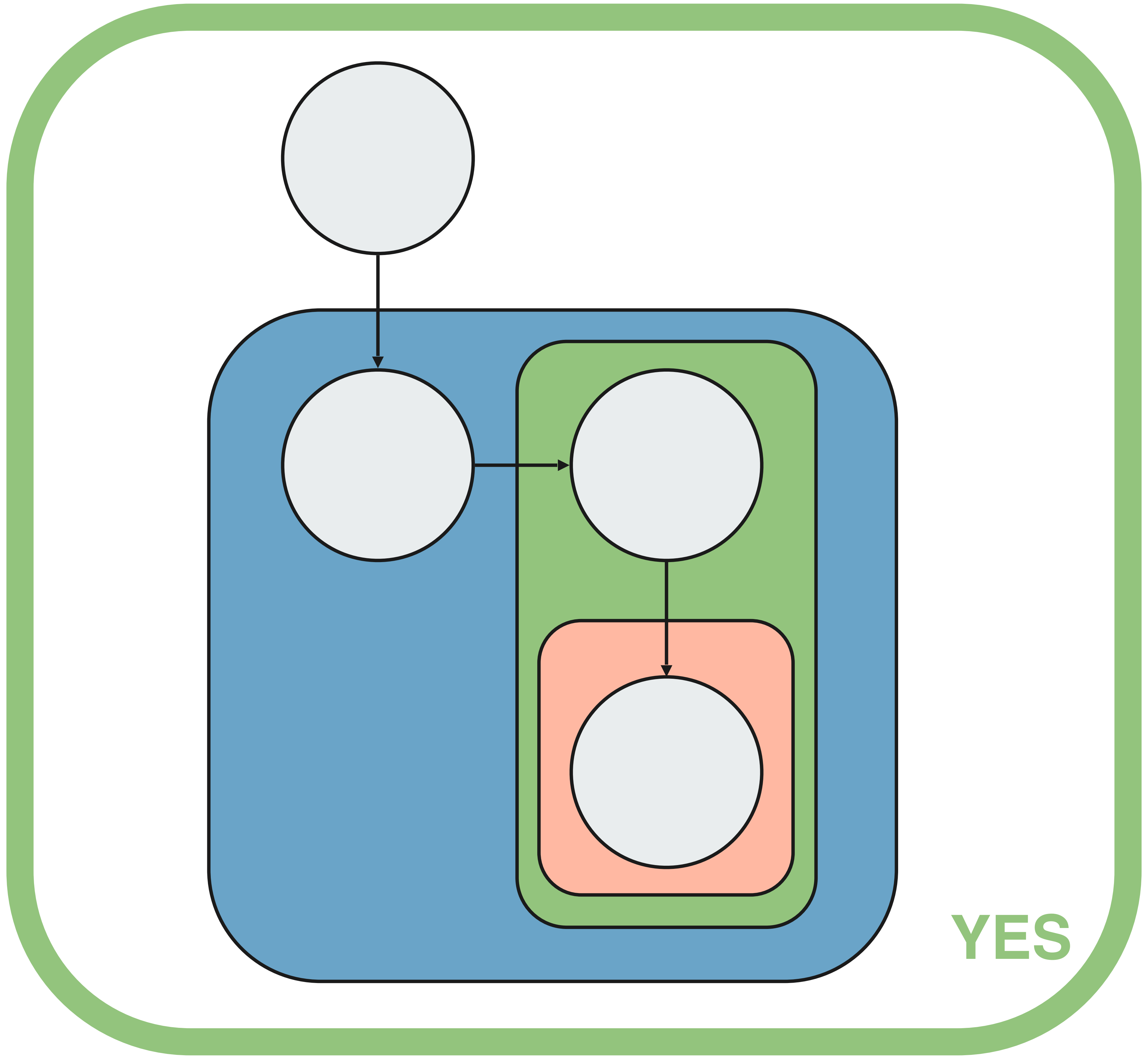
Definition of a pyramidal HBM

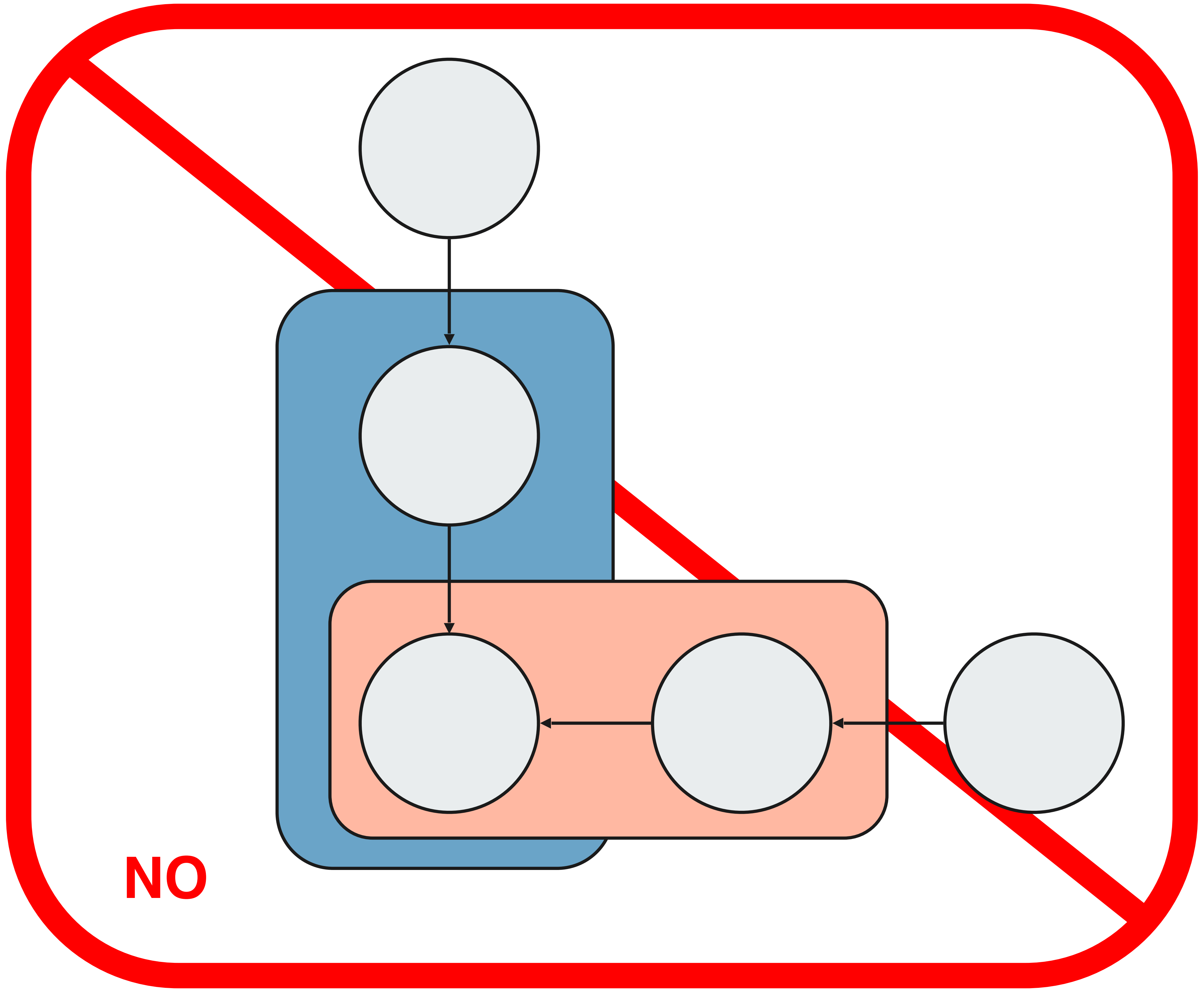
- A **simpler class** of problems to build our proof-of-concept architecture…
- ...yet **expressive** enough to encompass “real-life” models
- A subclass of **plate-enriched** Hierarchical Bayesian Models

Pyramidal HBM =

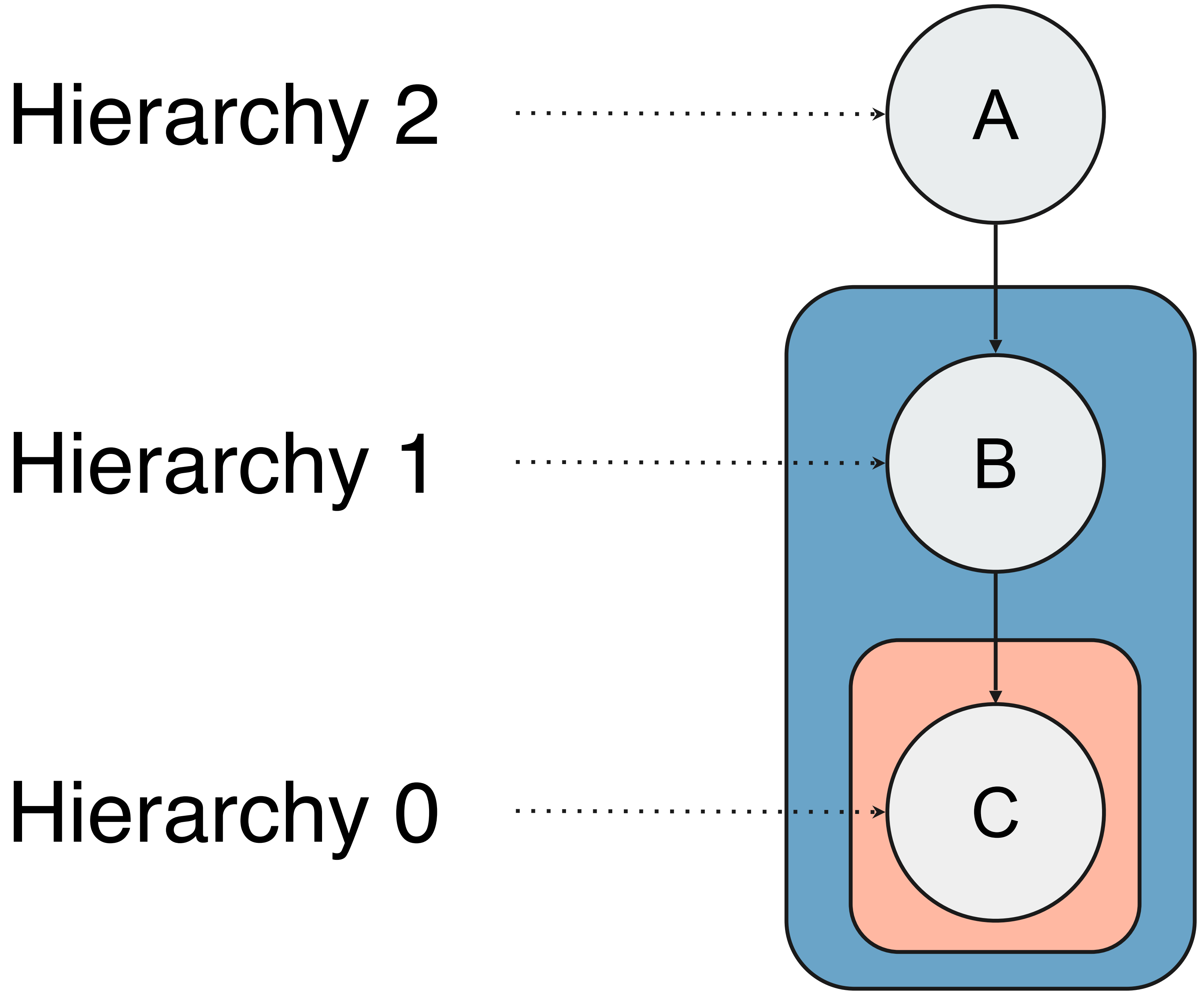
> “a single stack of plates with a single observed data at the bottom”

Graphical overview: no colliding plates





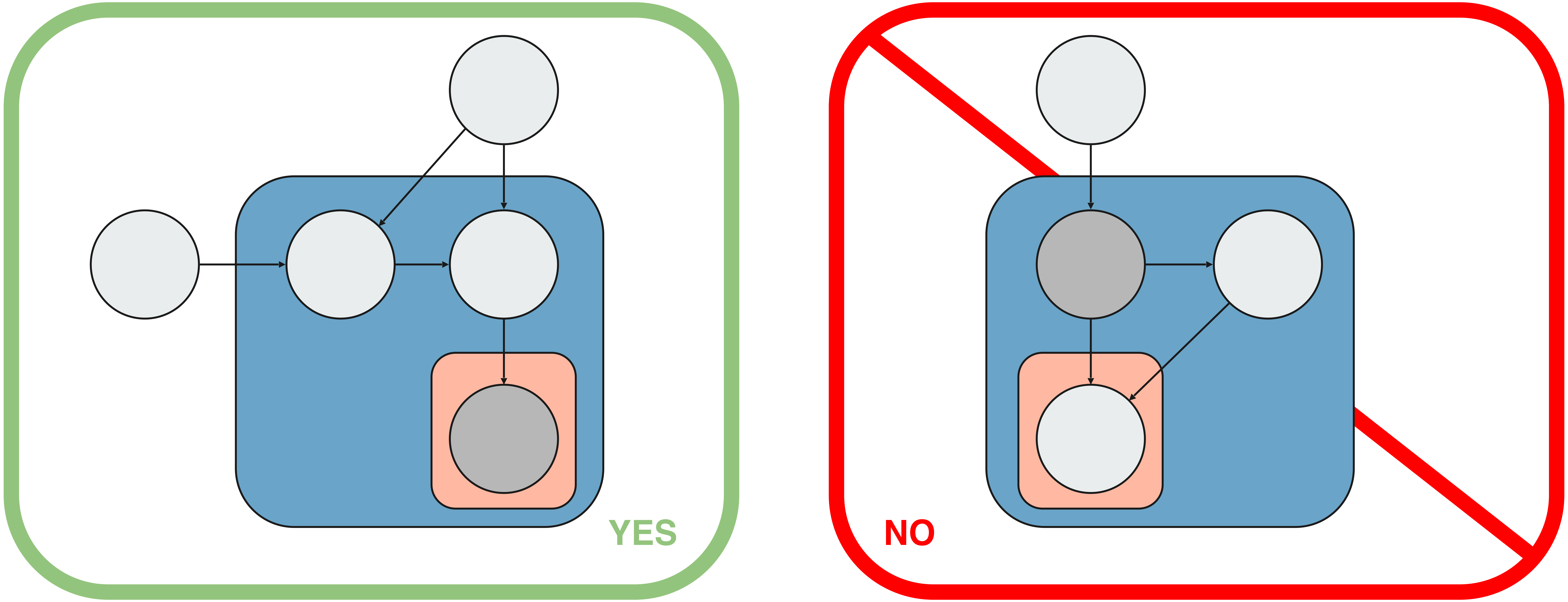
The notion of a RV's hierarchy



Hierarchy =
How "high" is a RV in the pyramid

○ = latent ● = observed

Graphical overview: unique observed data at last hierarchy



ADAVI: 2 main building blocks

- A **hierarchical encoder** (HE) that encodes the observed data X across multiple **hierarchies**
- A set of **conditional density estimators** that approximate the posterior distribution

We’ll review sequentially those items

Part 2

Methodological overview

Subpart B: Hierarchical Encoder

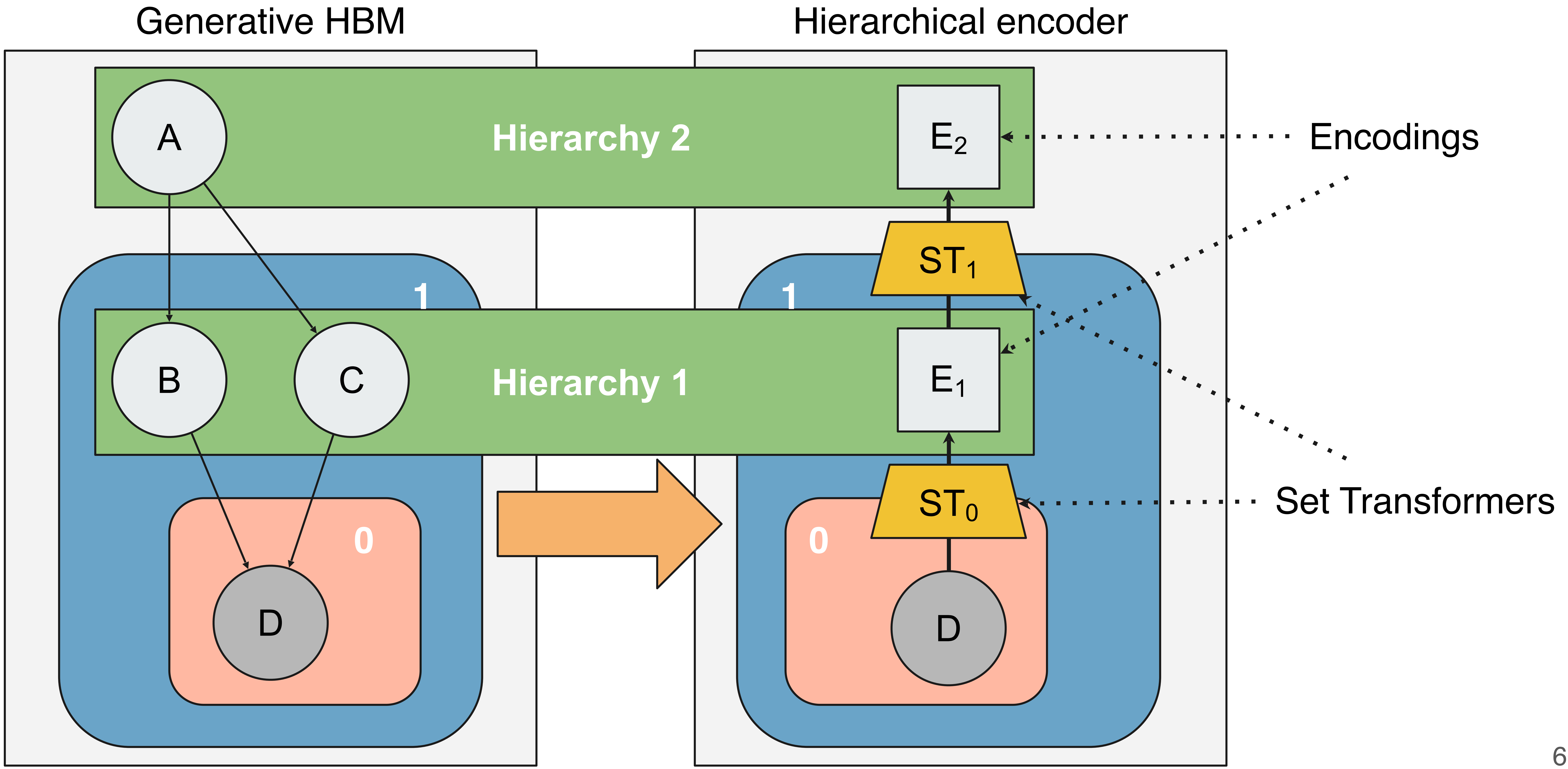
Hierarchical Encoder

- Sequentially contracts plates in the observed data X to produce multiple encodings
- **One encoding per hierarchy level** (later used for every RV that shares this hierarchy)
- Idea: exploit the i.i.d symmetry across a plate, using multiple stacked *Set Transformers (Lee et al. 2019)*

Set Transformer = an **attention-based** neural network architecture that exploits the **permutation invariance across a plate**

The hierarchical encoder is responsible for the **amortization** of our variational family

Graphical overview

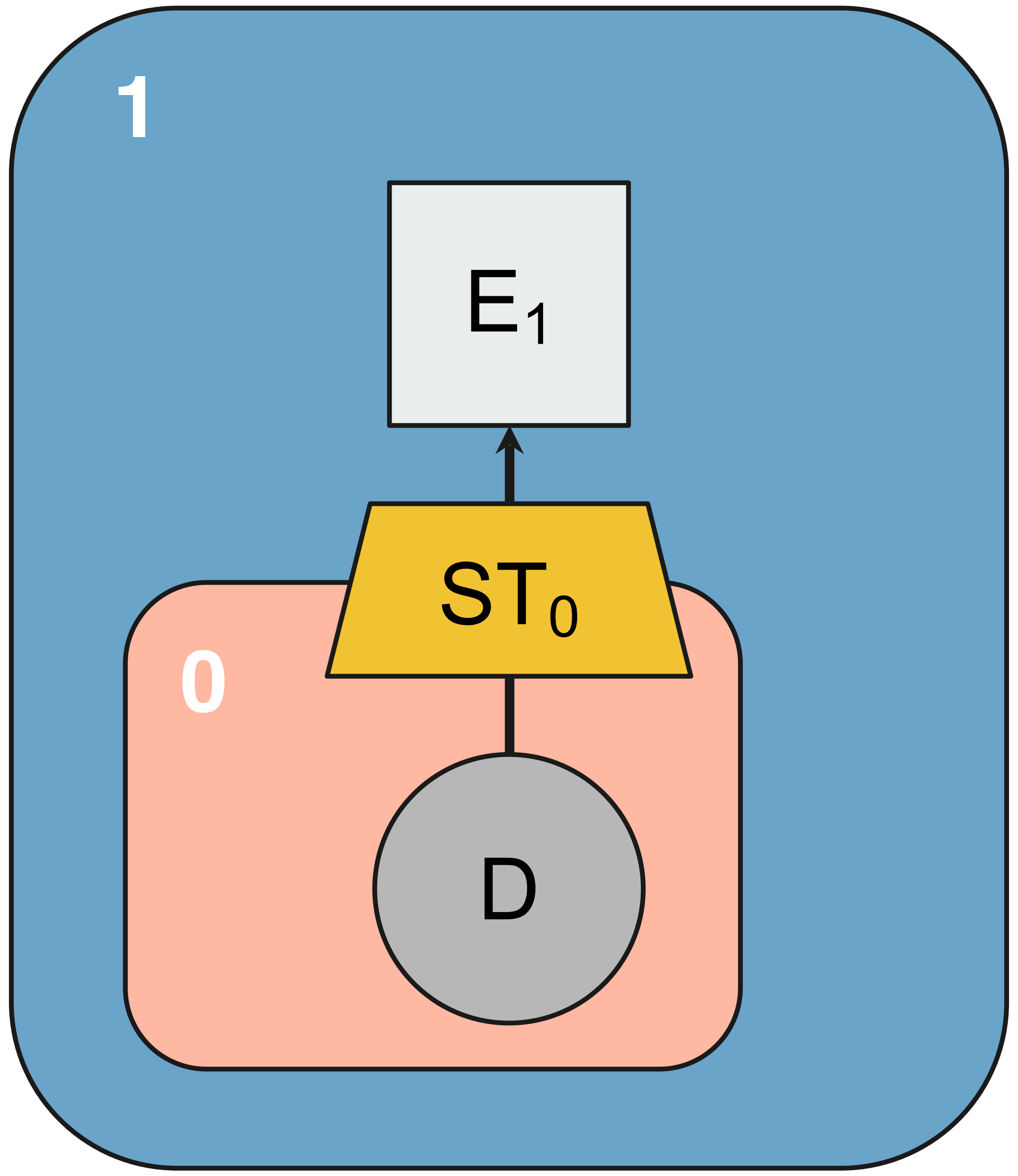


Function mapping for Set Transformers

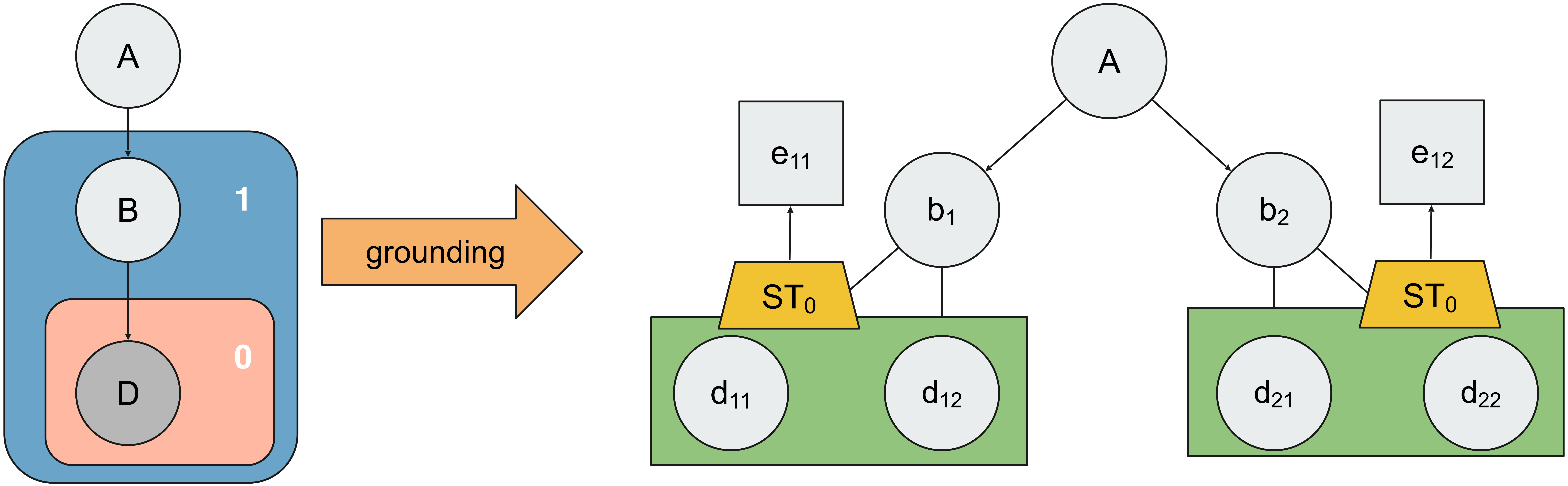
- The set transformer ST_0 **contracts** the plate P_0
- It does this operation in **parallel** across plate P_1

This means that **the parametrization of ST_0 is shared** for multiple operations: ST_0 produces as many encodings as the cardinality of P_1

This is an **essential feature** of our architecture: this is how we reduce our total number of parameters.



Overview over the ground graph (ignoring C)



One single function ST_0 produces the encoding $E_1 = \{ e_{11} ; e_{12} \} = \{ ST_0(d_{11} , d_{12}) ; ST_0 (d_{21} , d_{22}) \}$

e_{11} will be used to infer b_1 and e_{12} will be used to infer b_2

Part 2

Methodological overview

Subpart C: Conditional density estimators

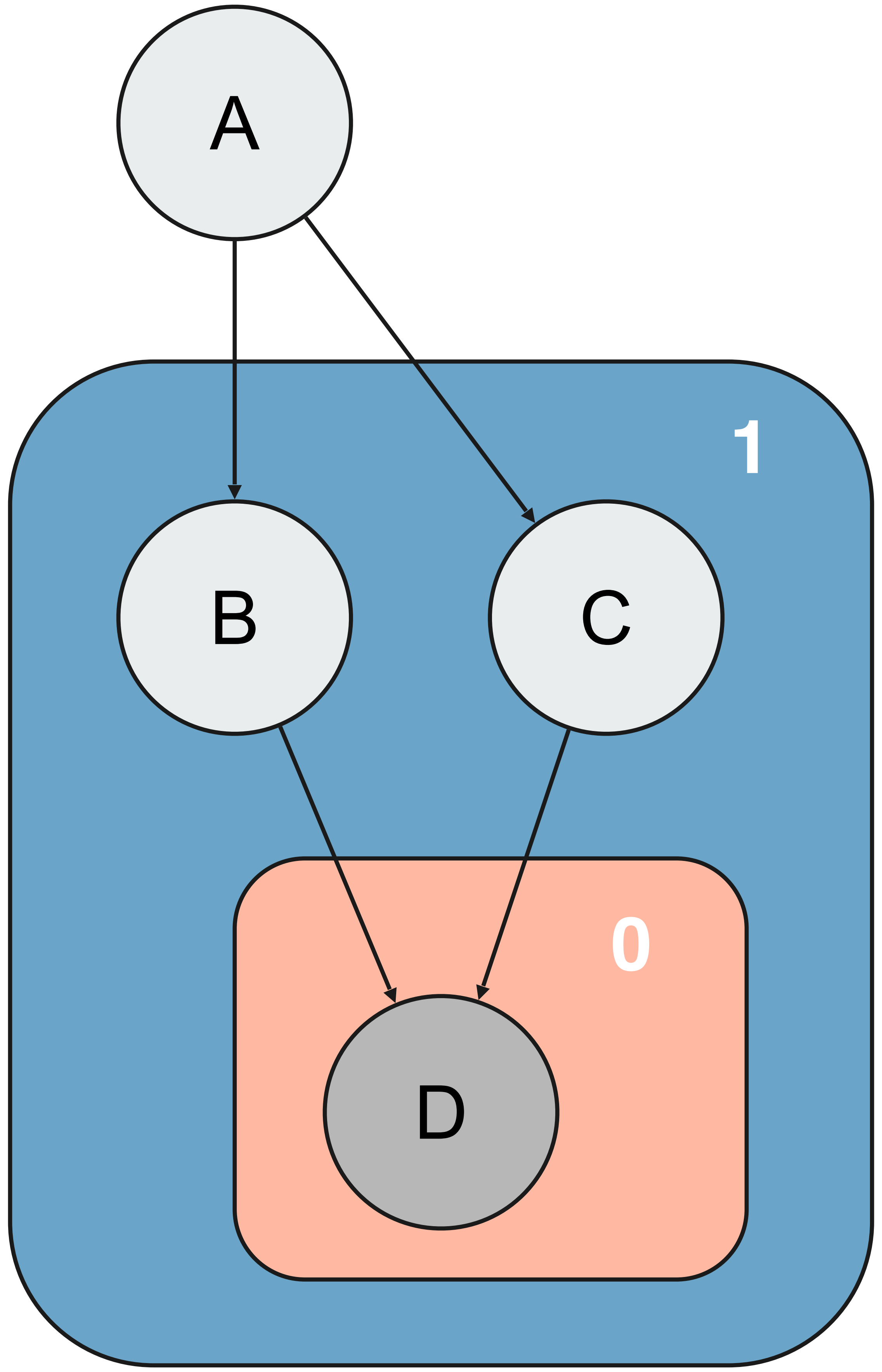
Conditional density estimators

We build **a density estimator for every latent RV template**. If for the generative HBM we have (D is observed):

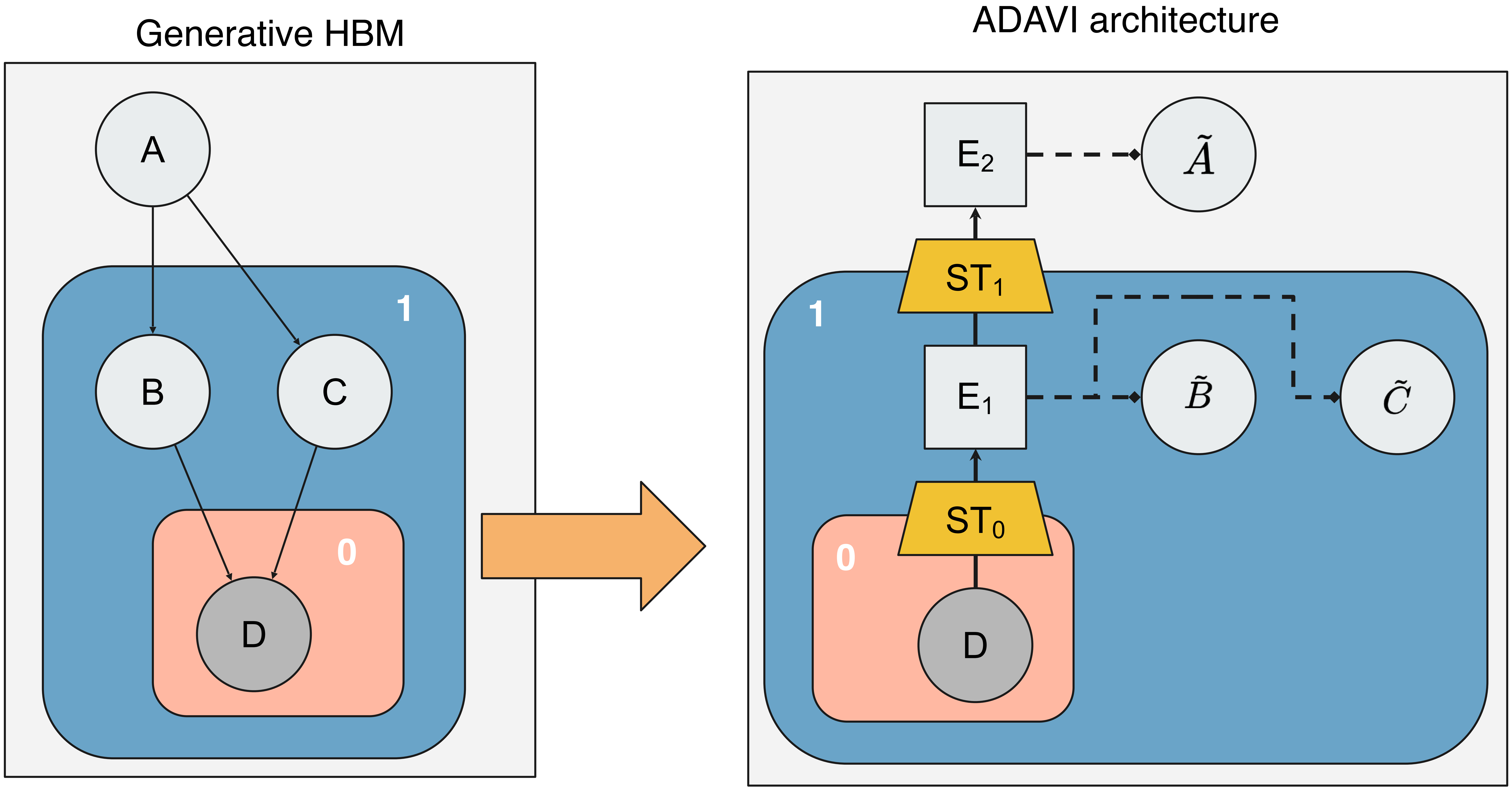
$$p(A,\, D,\, C,\, D) = p(A) \times p(B \mid A) \times p(C \mid A) \times p(D \mid B, C)$$

Then we will have 3 different density estimators:

$$q_A(A) \approx p(A \mid D)$$
$$q_B(B) \approx p(B \mid D)$$
$$q_C(C) \approx p(C \mid D)$$

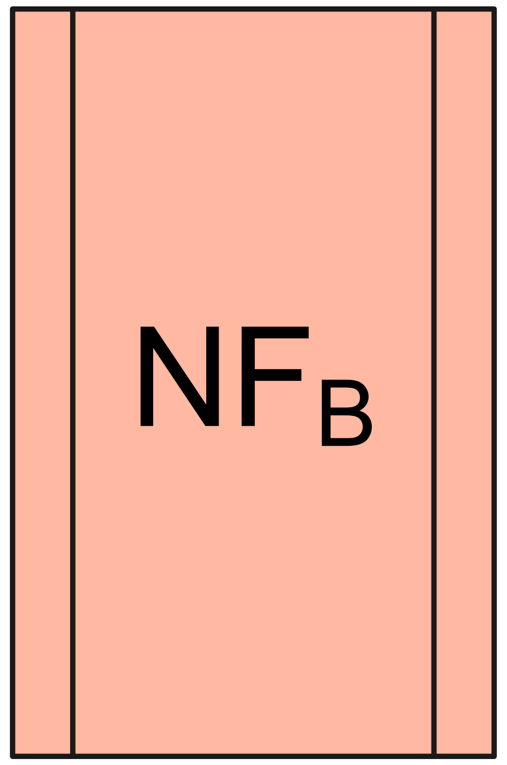


Graphical overview



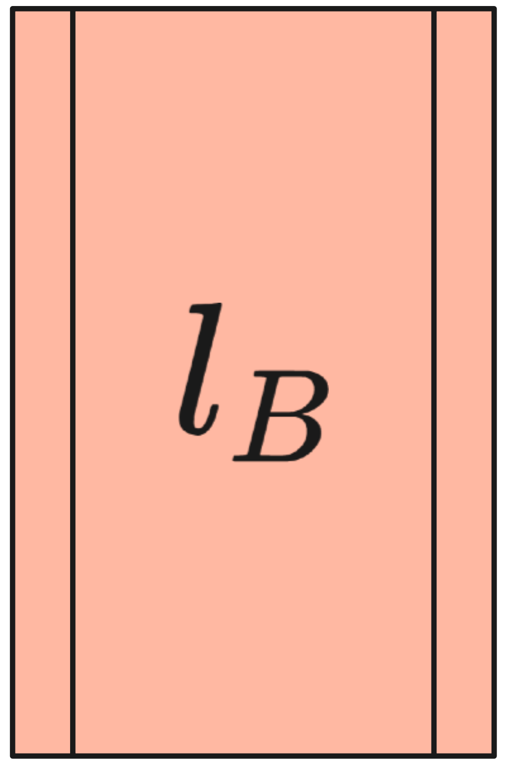
$$\tilde{A} \sim q_A$$
$$\tilde{B} \sim q_B$$
$$\tilde{C} \sim q_C$$

Architecture of a density estimator (1/2)



A single density estimator is the combination of 2 items:

- a “universal” density estimator in the real unbounded space: for this we use **Normalizing Flows** (*Rezende et al. 2016, Papamakarios et al. 2019)*
  - a normalizing flow re-parametrizes a standard normal distribution into a more complex distribution
  - leveraging the normalizing flow litterature, we can obtain very expressive density estimators



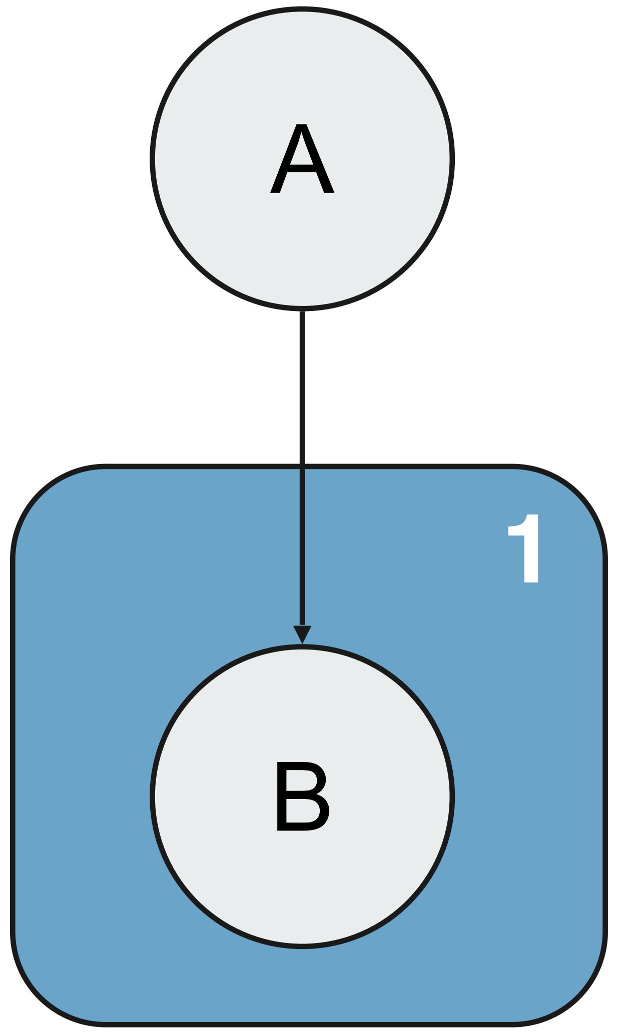
Architecture of a density estimator (2/2)

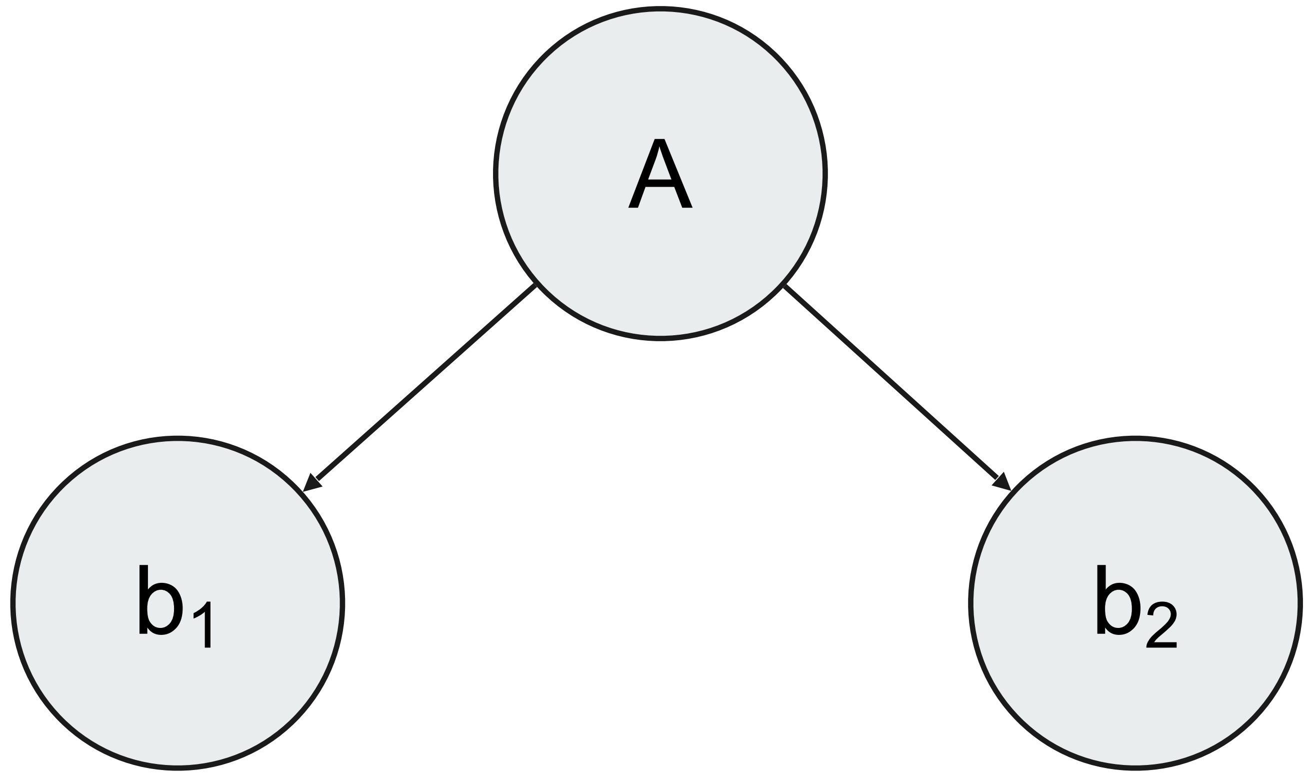
A single density estimator is the combination of 2 items:

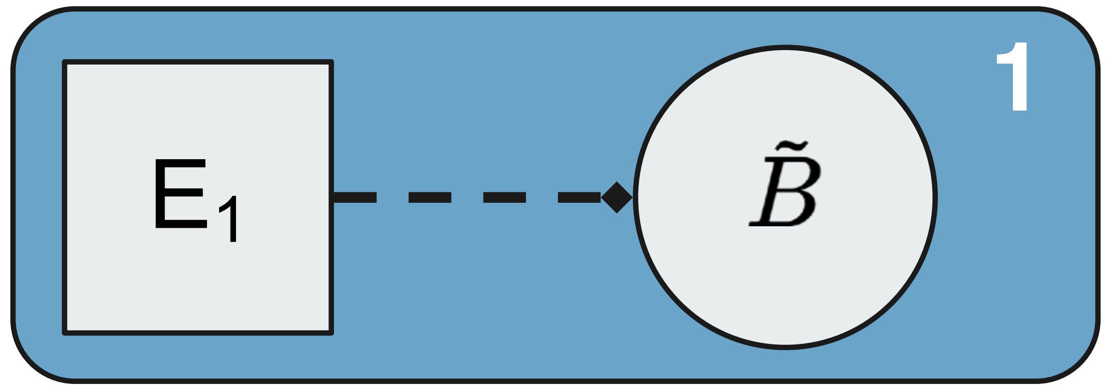
- a **link function** to project the real unbounded space to the constrained space in which the RV evolves:
    - for instance the space of real positive numbers for a variance
    - or the simplex for a mixture parameter
    - etc...

Function mapping for density estimators

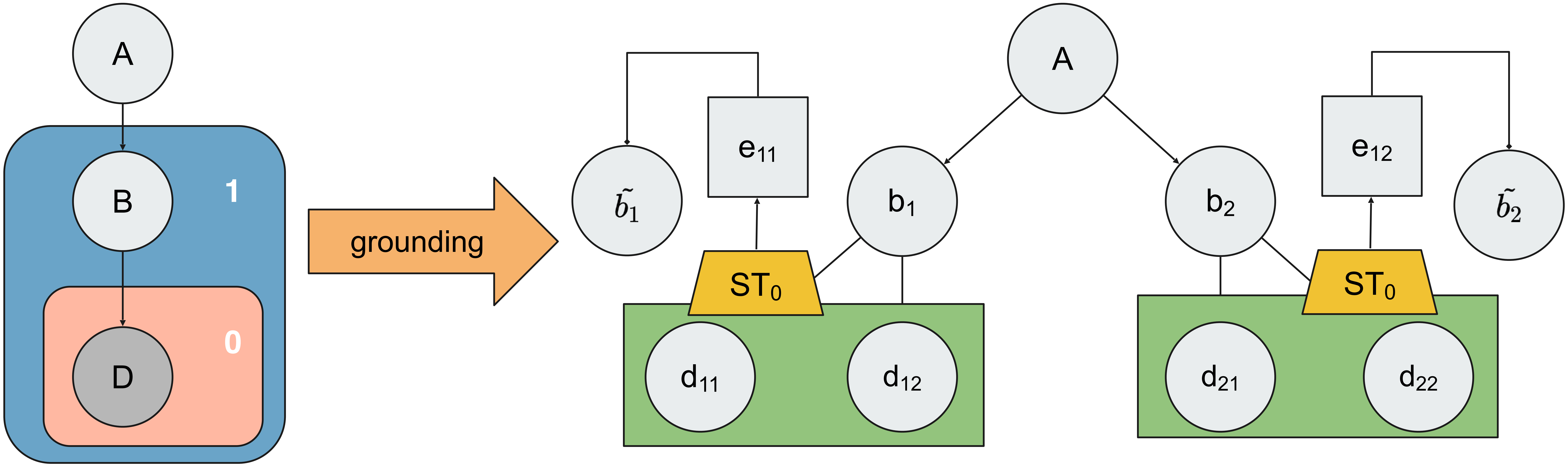
- Similar to Set Transformers, density estimators are applied **in parallel across plates**
- For instance, the density estimator q_B for the RV template B is applied in parallel across plate P_1, **sharing its parametrization** for the inference of both b_1 and b_2
- We therefore infer b_1 and b_2 independently
- For amortization purposes, the density estimation from q_B is **conditioned by the encoding** E_1:







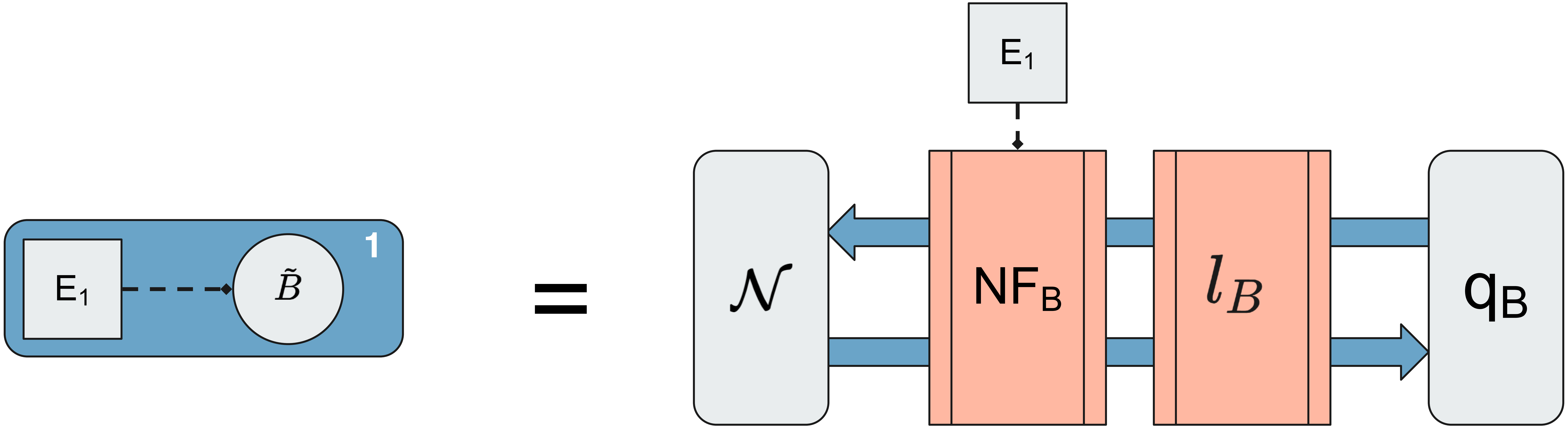
Overview over the ground graph (ignoring C)



One single function ST_0 produces the encoding $E_1 = \{ e_{11} ; e_{12} \} = \{ \mathrm{ST}_0(d_{11}, d_{12}) ; \mathrm{ST}_0 (d_{21}, d_{22}) \}$

One single density estimator q_b estimates both b_1 and b_2 $q_B(B) = q_B(B; E_1) = q_b(b_1; e_{11}) \times q_b(b_2; e_{12})$

Overview of a density estimator



Both the normalizing flow and the link function are diffeomorphisms, allowing for density computation using the change-of-variable formula (*Papamakarios et al. 2019*)

Putting estimators together

We combine the individual density estimators using a **mean field approximation**:

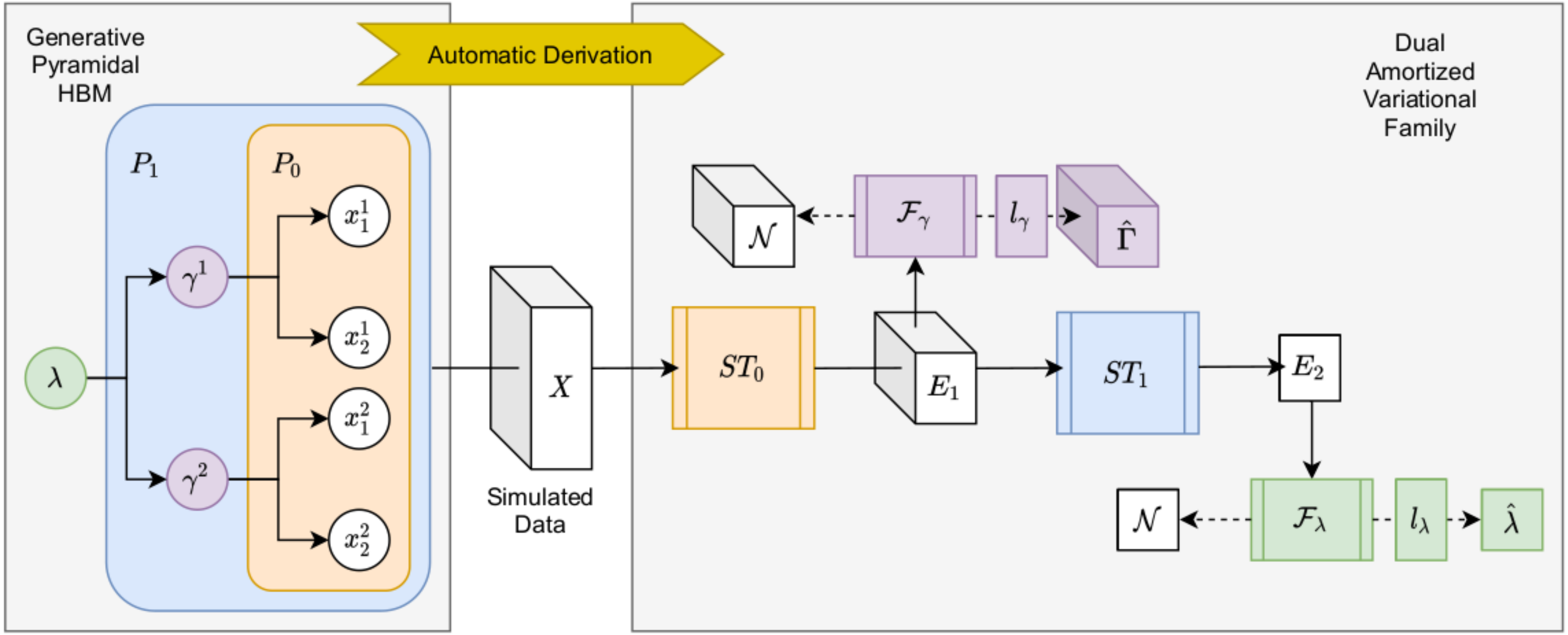
$$q(A,\ B,\ C) = q_A(A) \times q_B(B) \times q_C(C)$$

This means that we **don’t model statistical dependencies** in the posterior between different RV templates. This is an implementation choice, not a necessity for our architecture.

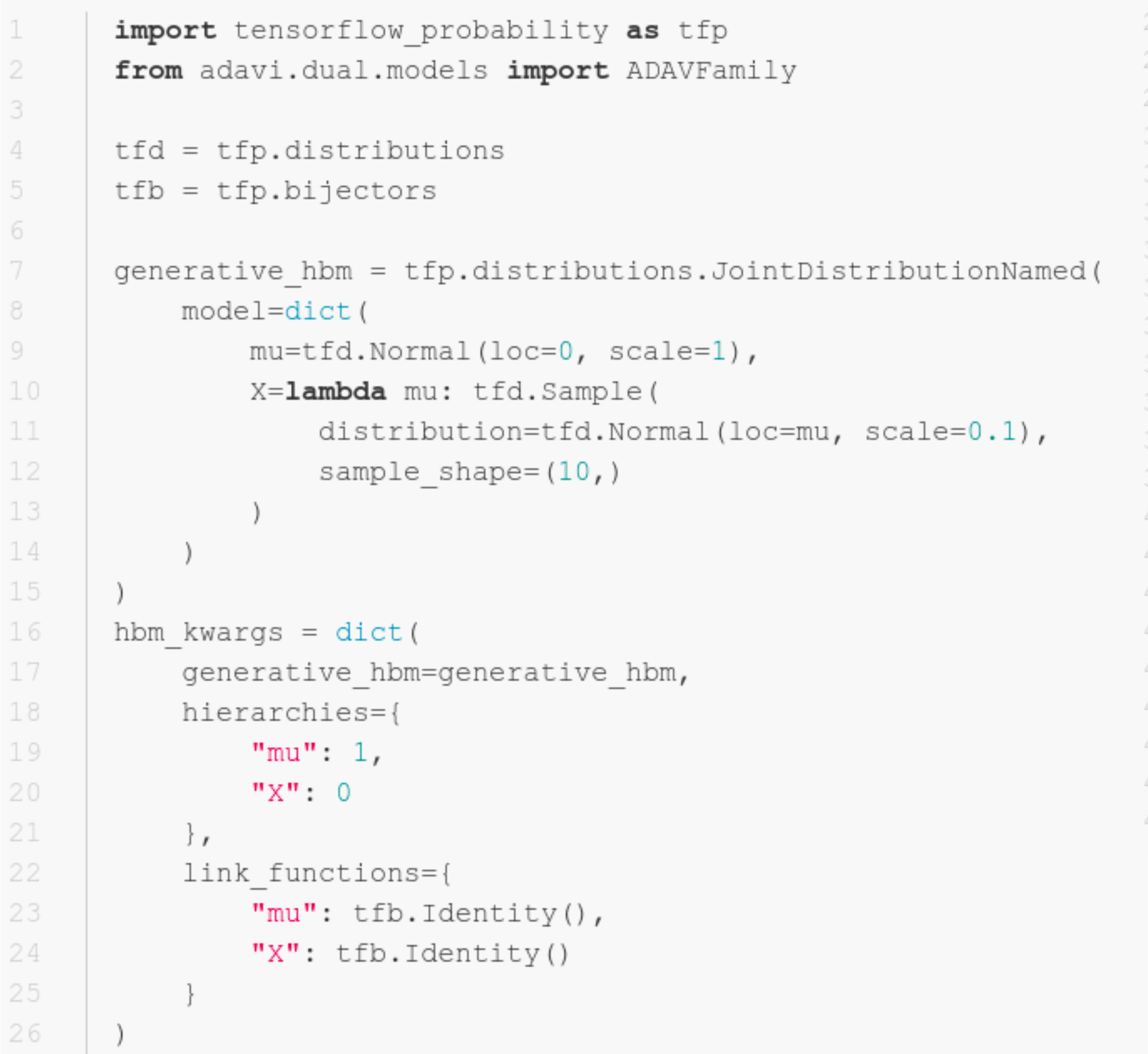
Inside the resulting variational family, we then optimize for q:

$$\arg\min_q \ KL(q(A, B, C) \,||\, p(A, B, C \mid D))$$

General overview of the ADAVI architecture



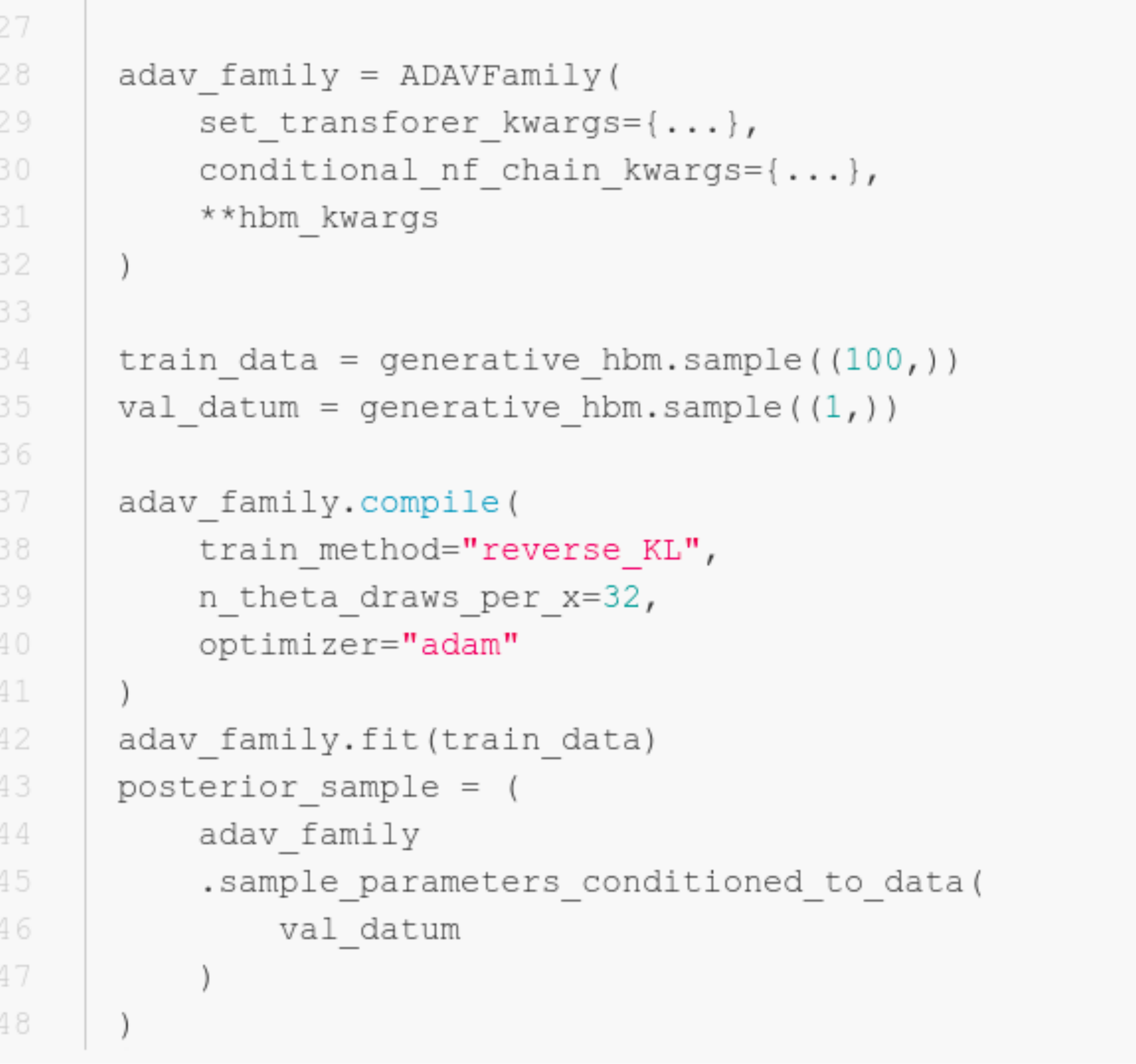
Rouillard et al. 2021



```python
import tensorflow_probability as tfp
from adavi.dual.models import ADAVFamily

tfd = tfp.distributions
tfb = tfp.bijectors

generative_hbm = tfp.distributions.JointDistributionNamed(
    model=dict(
        mu=tfd.Normal(loc=0, scale=1),
        X=lambda mu: tfd.Sample(
            distribution=tfd.Normal(loc=mu, scale=0.1),
            sample_shape=(10,)
        )
    )
)
hbm_kwargs = dict(
    generative_hbm=generative_hbm,
    hierarchies={
        "mu": 1,
        "X": 0
    },
    link_functions={
        "mu": tfb.Identity(),
        "X": tfb.Identity()
    }
)
```



```python

adav_family = ADAVFamily(
    set_transforer_kwargs={...},
    conditional_nf_chain_kwargs={...},
    **hbm_kwargs
)

train_data = generative_hbm.sample((100,))
val_datum = generative_hbm.sample((1,))

adav_family.compile(
    train_method="reverse_KL",
    n_theta_draws_per_x=32,
    optimizer="adam"
)
adav_family.fit(train_data)
posterior_sample = (
    adav_family
    .sample_parameters_conditioned_to_data(
        val_datum
    )
)
```

see https://github.com/NeuroLang/adavi
and *TFP Dillon et al. (2017)*

Part 3

Experimental results

Subpart A: Gaussian random effects

Baseline of comparison

Exploiting the structure of the forward HBM, we **factorize the parameter space** into multiple sub-spaces, corresponding to multiple NF blocks.

We furthermore solve in **parallel** multiple similar inference tasks (across a plate) using a common conditional density estimator.

Our point of comparison is **a single "big" NF** that wouldn't exploit this structure and simply model the joint distribution for θ:

- For instance (S)NPE-C (*Greenberg et al. 2019*)

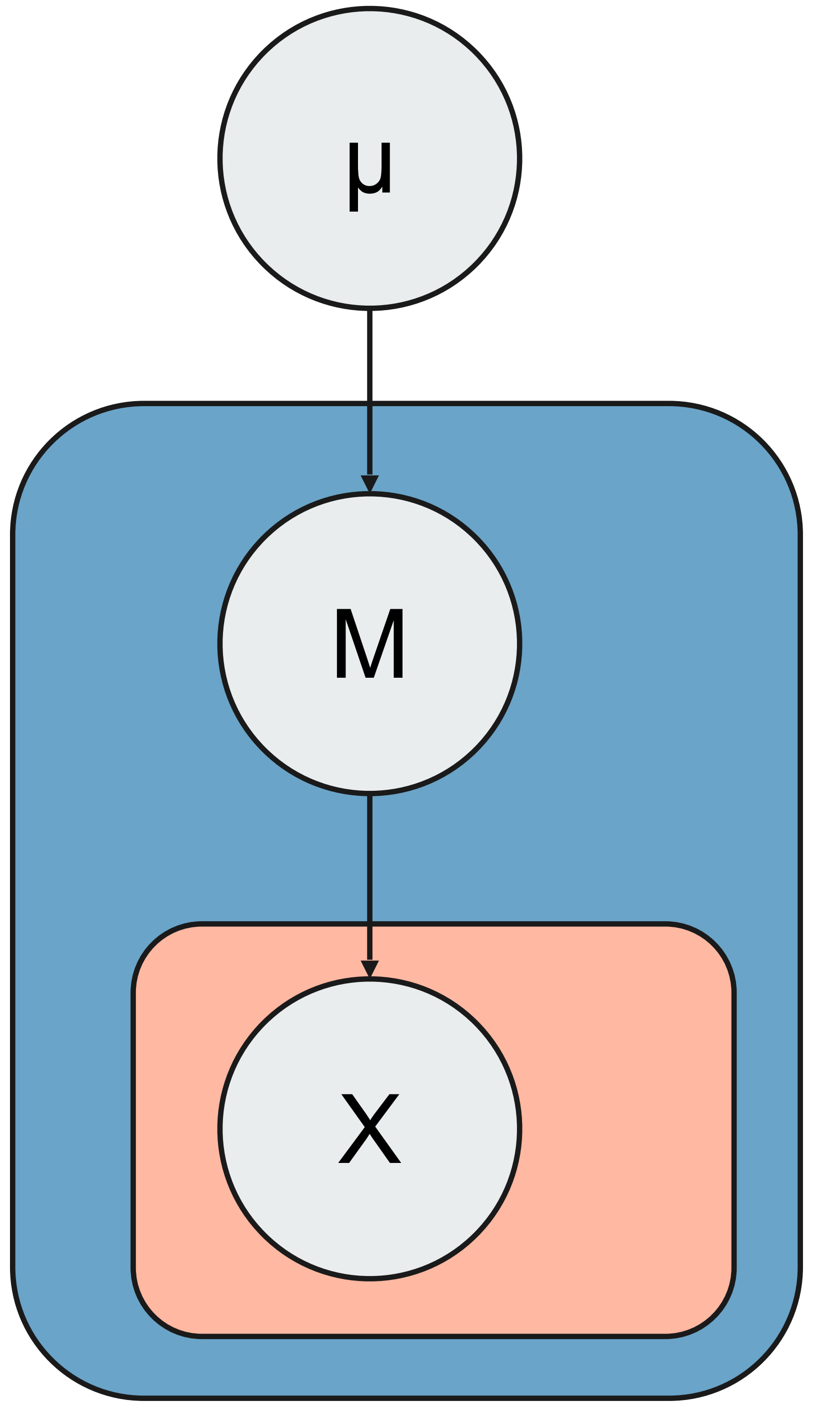
The forward HBM: Gaussian random effects

- We consider a **population mean** μ in dimension D=2
- From a Gaussian distribution centred on μ, we draw G=3 **group means** μ_1, μ_2 and μ_3
- For every group 1, 2, 3, we draw N=50 points from a gaussian centered on the group mean μ_1, μ_2, μ_3 to obtain the observed data X

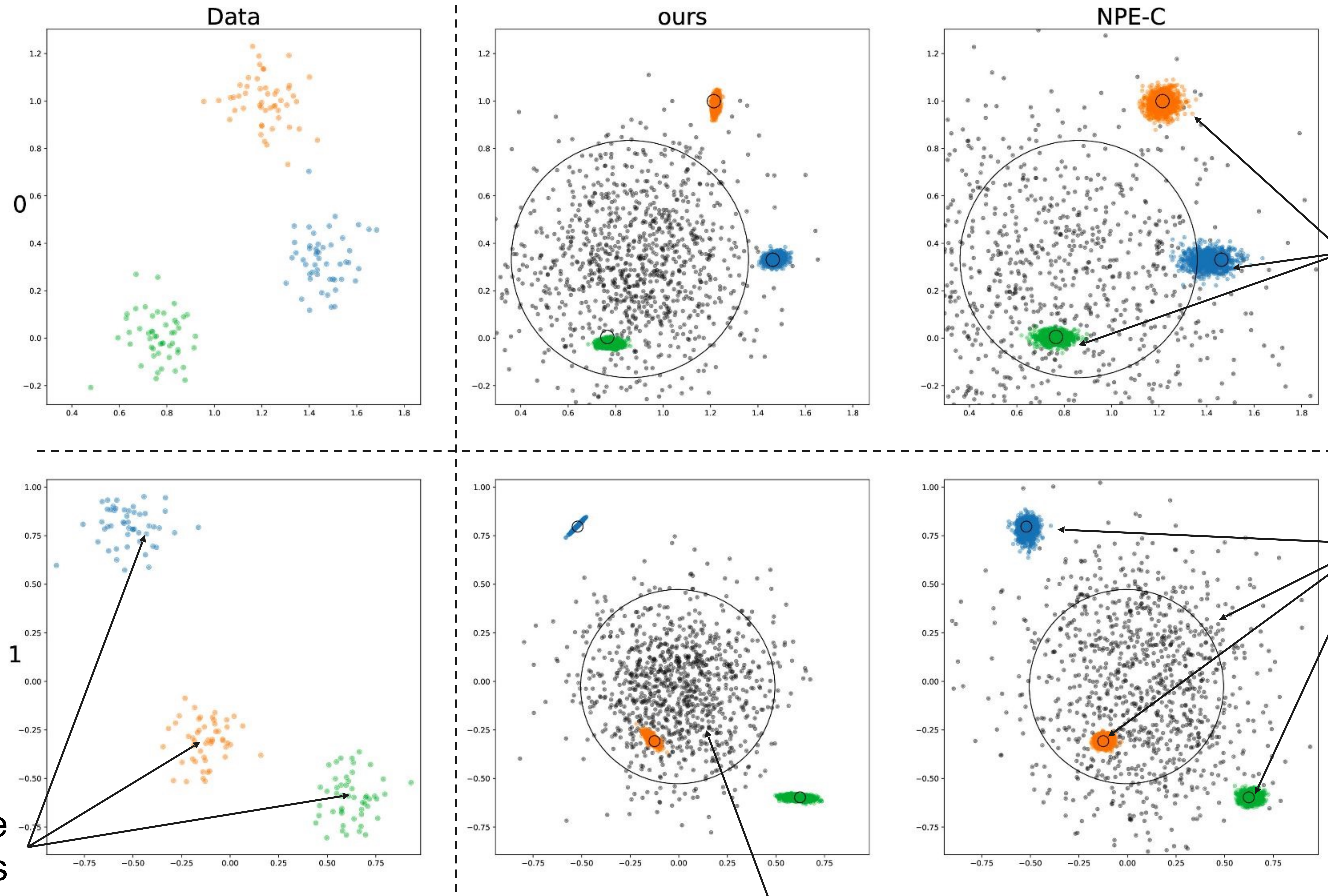
The goal:

infer the posterior distribution of μ_1, μ_2, μ_3 and μ given X

There are 2 plates and 3 levels of hierarchy in this problem.



Posterior samples for the 2 methods



2 different data points

samples for the G=3 groups

colored points = samples from μ_1, μ_2, μ_3 posterior

black circles = theoretical ground truth

black points = samples from μ posterior

Parameterization with respect to plate dimensionality

The **total number of parameters** to estimate grows with the plate size G: adding more groups means more group means to infer.

A NF’s parameterization **scales quadratically** with the size of the parameter space (e.g. *Real NVP Dinh et al. 2017, FFJORD Grathwohl et al. 2018, MAF Papamakarios et al. 2018*)

In this example, the parametrization of a “single big NF” will be

$$\mathcal{O}(G^2 D^2)$$

In comparison, our parameterization is

In the general case with M plates, we have $\mathcal{O}(D^2)$ parameters vs

$\mathcal{O}(MD^2)$ $\mathcal{O}(\mathrm{Card} P_1^2 \times \ldots \times \mathrm{Card} P_M^2 \times D^2)$

		NPE-C	ADAVI
G = 3	C2ST mean (std)	1.00 (0.00)	0.70 (0.10)
	# Parameters	42k	13k
	Computing time (CPU)	1d	20 m (1m on GPU)
G = 15	C2ST mean (std)	1.00 (0.00)	0.70 (0.17)
	# Parameters	85k	13k
	Computing time (CPU)	4.9d	99m
G = 30	C2ST mean (std)	1.00 (0.00)	0.85 (0.17)
	# Parameters	138k	13k
	Computing time (CPU)	7.6d	166m

See benchmark from *Lueckmann et al. (2021)* for Classifier 2-Sample Test (C2ST) metric

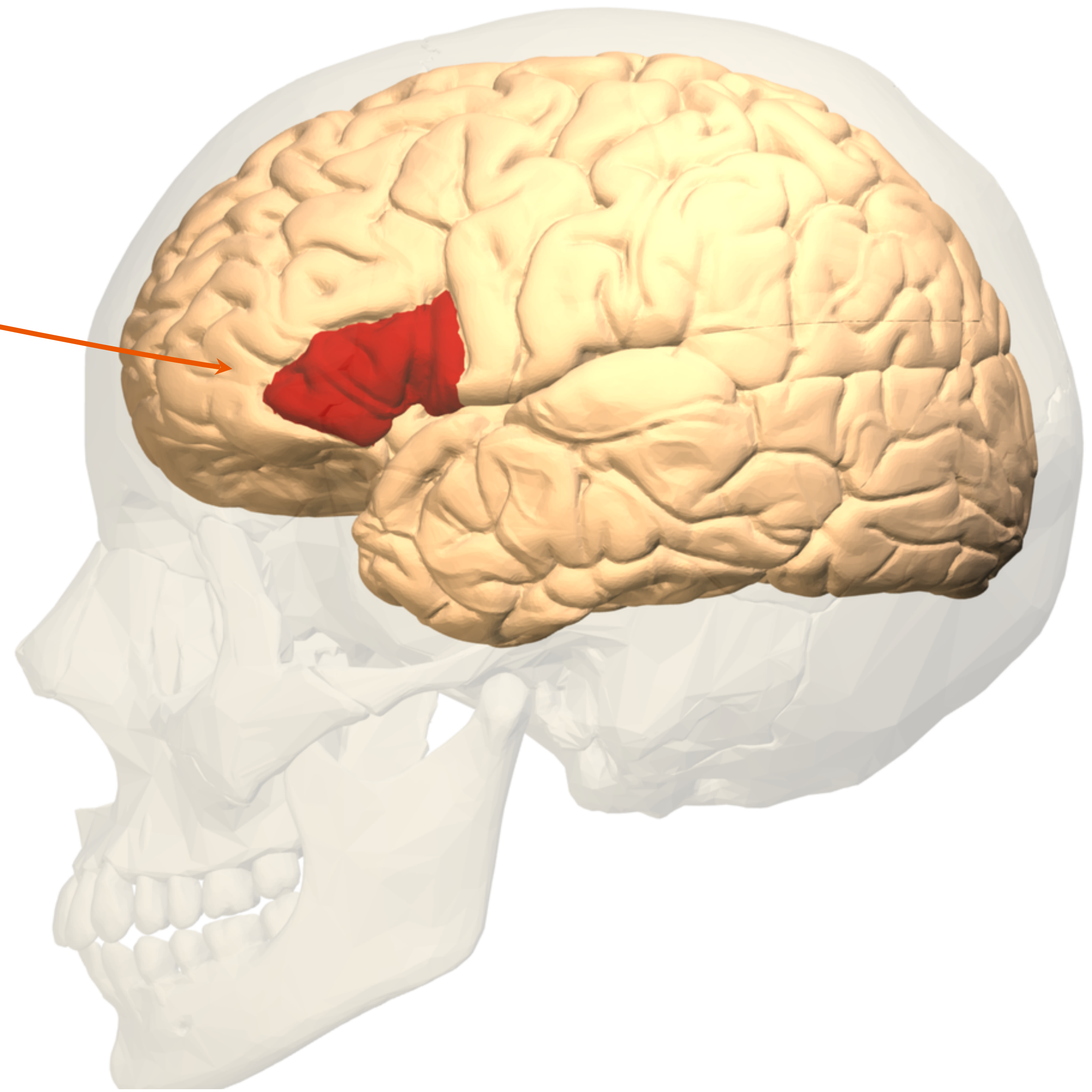
Part 3

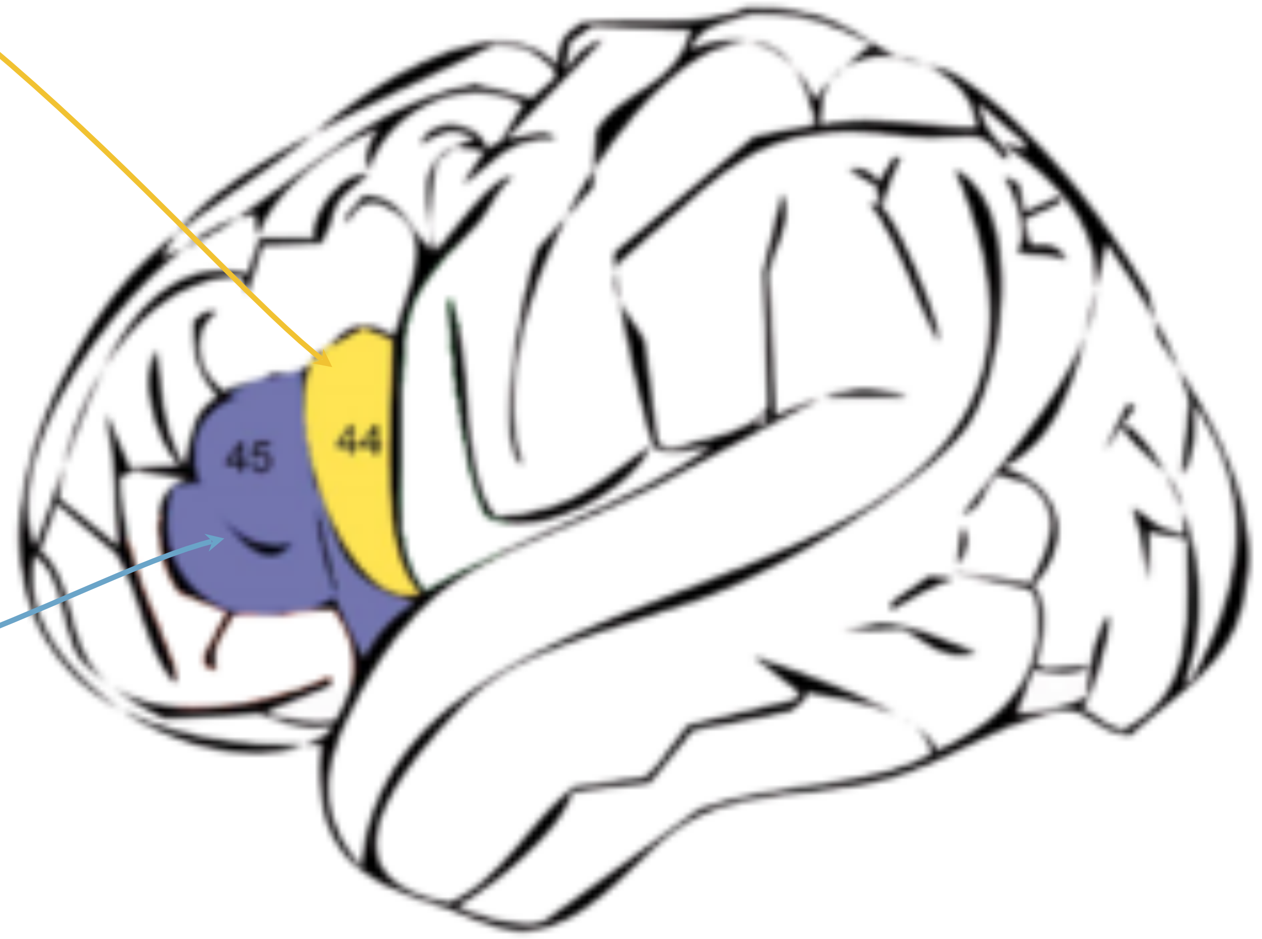
Experimental results

Subpart B: Neuroimaging experiment

Broca’s area functional parcellation

- We consider **Broca’s area** in the Inferior Frontal Gyrus, traditionally associated to language
- Broca’s area can be anatomically split into **2 parts** (pars triangularis and pars opercularis). Our goal is to recover that binary split using a **functional parcellation** based on f-MRI data
- We consider **connectivity vectors** = how is a given brain vertex “wired” to the rest of the brain (functional definition)
- Data from the Human Connectome Project (HCP) (*Van Essen et al. 2012*) preprocessed with the help of Dr. Thomas Yeo and Dr. Ru Kong (CBIG)





wikipedia

Multiple scales of variability

We adapt the MS-HBM from *Kong et al. (2018)*:

○ we consider 2 distinct **population** connectivity networks μ_1^g and μ_2^g

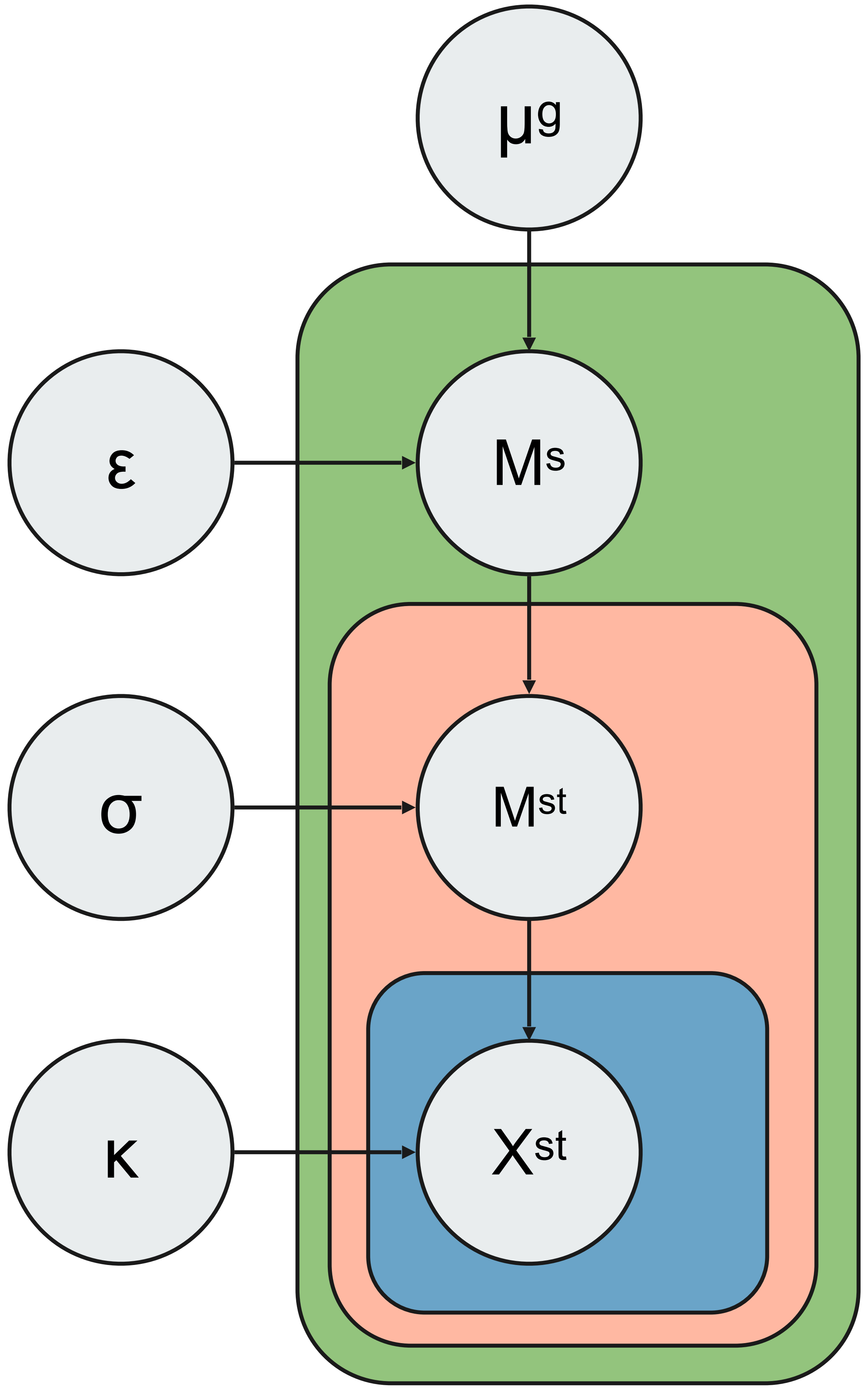
○ each **subject**'s connectivity networks μ_1^s and μ_2^s vary from the population networks

○ the connectivity networks of an individual can vary across time, resulting in **session** connectivity networks μ_1^{st} and μ_2^{st}

○ for a given subject and session, a given **brain vertex** can express a connectivity X^{st} as a variation of one of the 2 connectivity networks (mixture model)

○ a given vertex therefore has a **label** corresponding to the network it belongs to (1 or 2)

All this variability is encompassed into a single hierarchical model, with a probabilistic treatment: this showcases the **strength of the Bayesian approach**.



Total: 300k parameters !

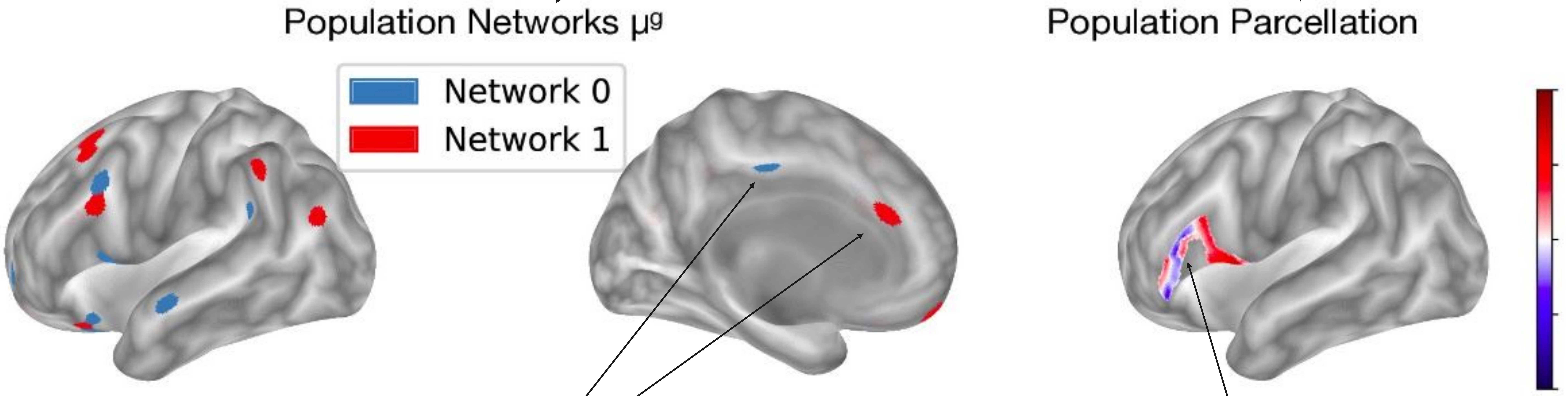
Barriers to entry for experimenters

- Though Bayesian methods are appealing, inference usually requires a **lot of work**, and strong methodological knowledge: analytical derivations, lengthy method building and tuning, etc...
- In the original implementation, *Kong et al.* use a **manually-derived EM** procedure (with pages of equations)
- Furthermore, the **very high dimensionality** of the parameter space prohibits any naive approach, doubling down on the methodological knowledge required

With ADAVI, we place ourselves in the line of **automatic VI**, seamless to use for experimenters once the forward model has been expressed in a modern probabilistic framework (*TFP Dillon et al. 2017*).

Our exploitation of plates allows us to perform inference efficiently in a data regime where existing methods would quickly become intractable “wiring” to the rest of the cortex

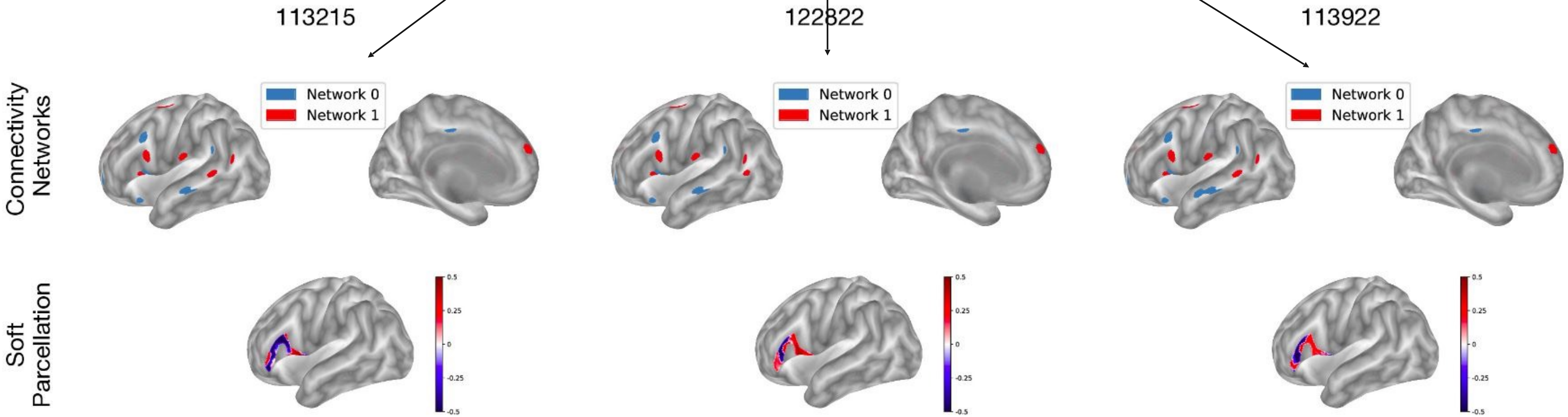
“functional cartography” for the cortex



colored spots mark the top 99% of connectivity for both networks (red and blue)

- “red-ish” and “blue-ish” parts represent posterior probability for the vertex’ network label
- “white-ish” means uncertainty

3 different pair of networks for 3 different subjects



3 different parcellations for 3 different subjects

Part 4
Conclusive remarks

Methodological extensions

- ADAVI leverages a simple principle: **the i.i.d symmetry introduced by plates is translated into a shared parametrization both for encoding and density estimation**
- Many limiting implementation details (not tied to the method in itself) can be relaxed:
  - the **pyramidal** class of models
  - the **mean-field** approximation
  - the **non-sequentiality** of inference (see *SBI Cranmer et al. (2020)*)

Insights into inference

- ADAVI is an example of the gains from **exploiting structure** in an inference problem. It does so to reduce its parametrization rather than boosting its performance.
- More generally, the idea of ADAVI is to derive an Structured Variational family from a graph *template*, to exploit symmetries that exist in a *ground* graph
- That general line of thinking (shared in structured VI) is a promising road to **more and more effective (automatic) Variational Inference**

We tackled a complex real-life neuroimaging experiment with a fully Bayesian treatment, advancing the capabilities of Bayesian methods and making them more experimenter-friendly.

Thank you for your attention !

Bibliography

Kong et al., *Spatial Topography of Individual-Specific Cortical Networks Predicts Human Cognition, Personality, and Emotion*, 2018

Blei et al., *Variational Inference: A Review for Statisticians*, 2017

Kucukelbir et al., *Automatic Differentiation Variational Inference*, 2016

Ambrogioni et al., Automatic structured variational inference, 2021

Weilbach et al., *Structured Conditional Continuous Normalizing Flows for Efficient Amortized Inference in Graphical Models*, 2020

Bibliography

Ambrogioni et al., Automatic Variational Inference with Cascading Flows, 2021

Koller et Friedman, *Probabilistic graphical models: principles and techniques*, 2009

Lee et al., *Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks*, 2019

Rezende et al., *Variational Inference with Normalizing Flows*, 2016

Papamakarios et al., *Normalizing Flows for Probabilistic Modeling and Inference*, 2019

Rouillard et al., *ADAVI: Automatic Dual Amortized Variational Inference Applied To Pyramidal Bayesian Models*, 2021

Bibliography

Greenberg et al., *Automatic Posterior Transformation for Likelihood-Free Inference*, 2019

Dinh et al., *Density estimation using Real NVP*, 2017

Grathwohl et al., *FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models*, 2018

Papamakarios et al., *Masked Autoregressive Flow for Density Estimation*, 2018

Lueckmann et al., *Benchmarking Simulation-Based Inference*, 2021

Van Essen et al., *The Human Connectome Project: a data acquisition perspective*, 2012

Bibliography

Dillon et al., *TensorFlow Distributions*, 2017

Cranmer et al., *The frontier of simulation-based inference*, 2020