Active Inference

Practical Part: Computational Phenotyping in Psychiatry

Philipp Schwartenbeck

Centre for Cognitive Neuroscience, Salzburg

Wellcome Trust Centre for Neuroimaging, London

Computational Psychiatry Course 2016, TNU, Zurich

CCNS

Active Inference Essentials

Belief-based (choice) behaviour

- Agents build **generative models** of their environment.
  - I.e. joint probability distributions over observations y and causes θ: $P(y, \theta) = P(y|\theta) \cdot P(\theta)$
- Agents invert these models to infer latent variables, including current states, policies and the precision of beliefs

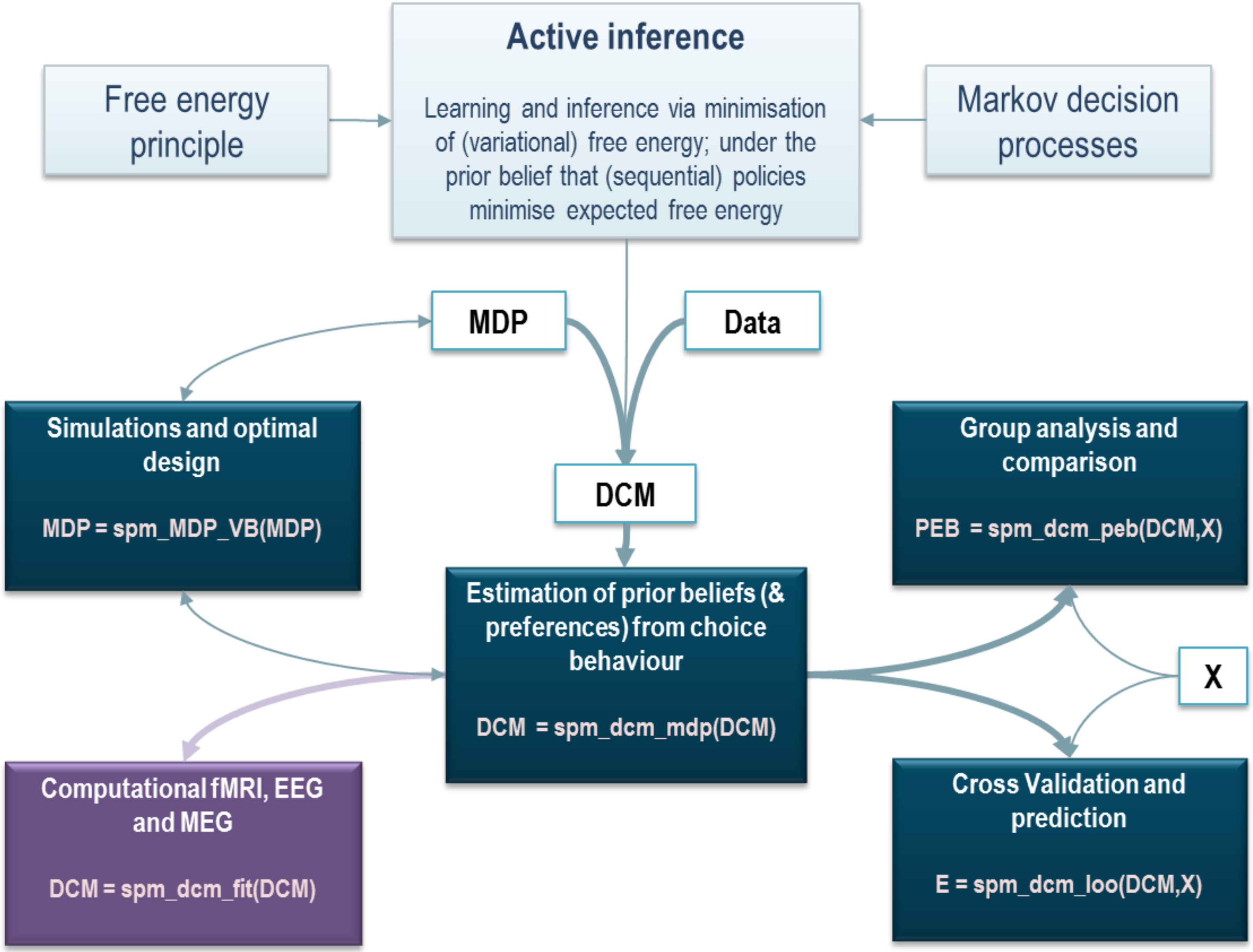
Preferences are cast as beliefs (expectations) about future states

- Agents try to fulfil their expectations
- Rewarding outcomes = expected outcomes

Objective function (cf., *computational level*, Marr): **minimise surprise**

- That means fulfilling expectations and maximising model evidence

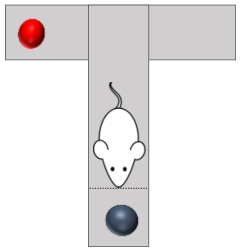
Computational Phenotyping: Overview



Schwartenbeck & Friston, 2016 *eNeuro*

Computational Phenotyping: Overview

I. ABC of choice behaviour: model a task that involves exploratory and exploitative behaviour



II. Simulate data and invert model to obtain subject-specific parameters

III. Simulate group effect and recover group-specific parameters

IV. Perform hierarchical Bayesian inference on group effects

Schwartenbeck & Friston, 2016 *eNeuro*

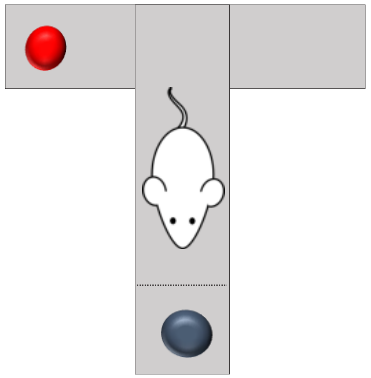
CCNS

UCL

Computational Phenotyping: Task

Two-step maze task

- A rat needs to obtain a reward in the left or right arm of a T-shaped maze
- It starts in the middle and can decide to go either left or right – or sample a cue at the bottom first
- The cue tells the rat with 95% validity where the reward is located
- The left and right arm are *absorbing states* (the rat cannot sample both)

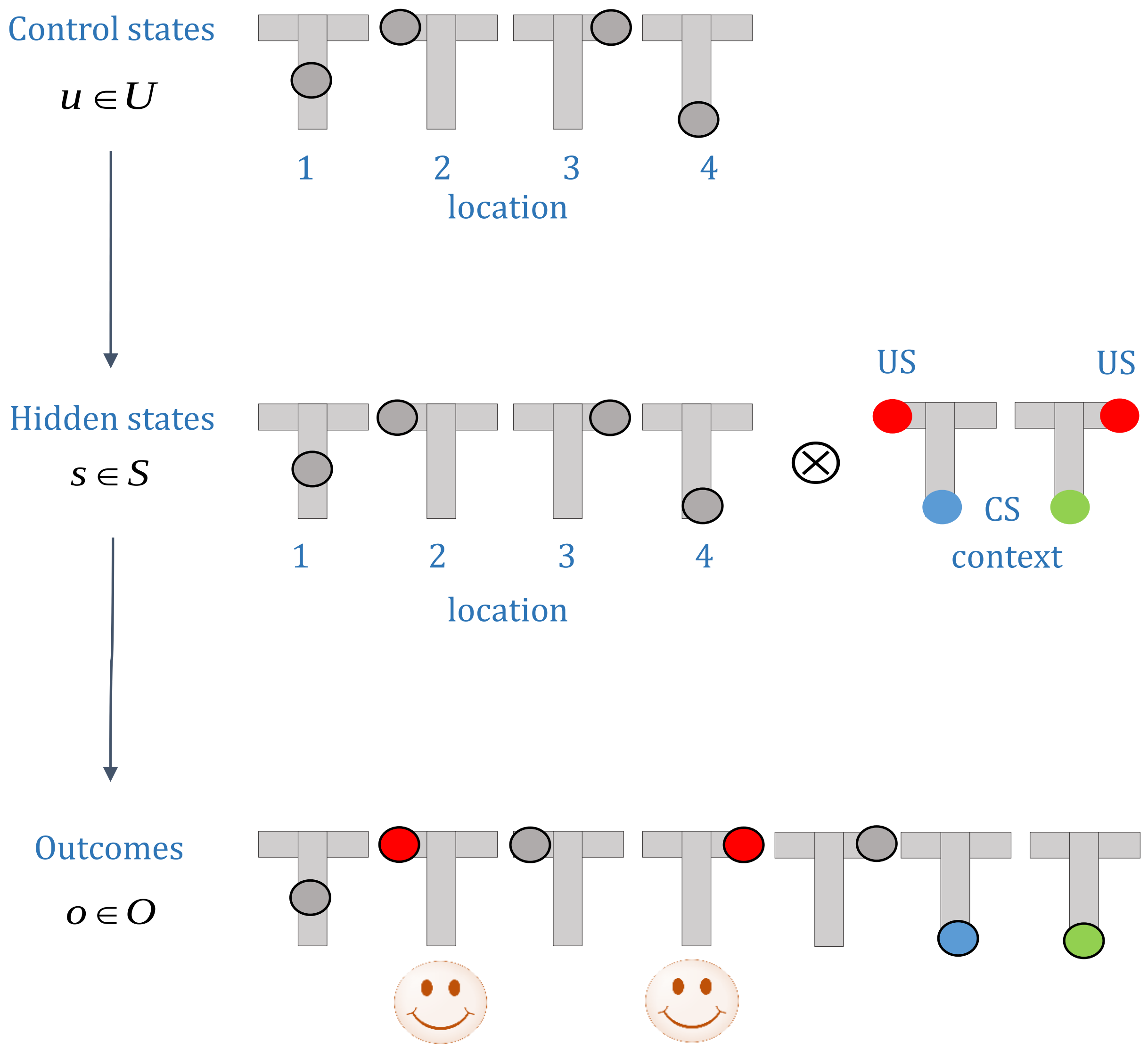


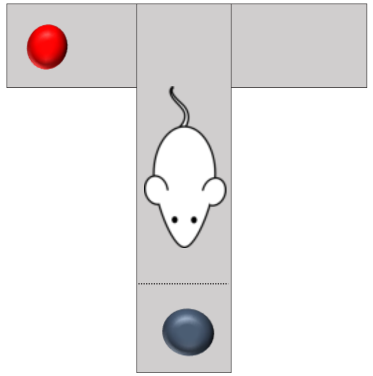
Thus, the rat needs to solve the exploitation-exploration dilemma

- In case of high uncertainty, it should sample the cue first

Friston et al., 2015 *Cognitive Neuoroscience*

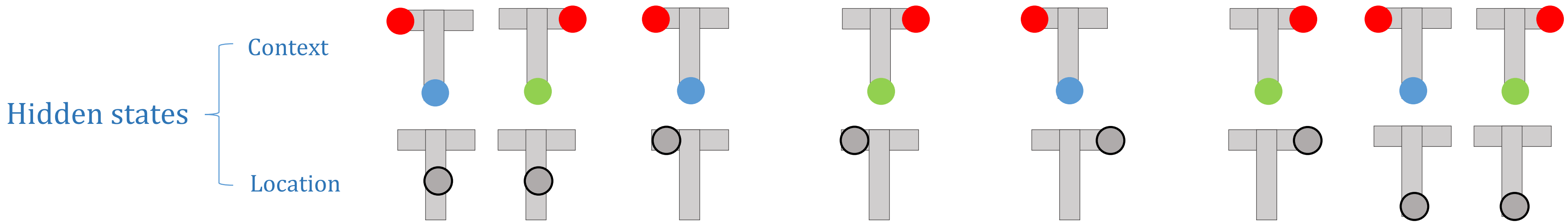
Subjective Generative Model



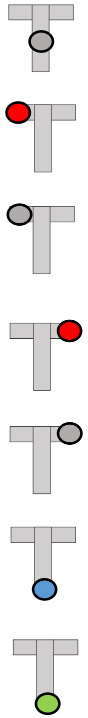


→**Toolbox**: ABC of choice behaviour

ABC of choice behaviour:
A – Mapping from hidden states to outcomes



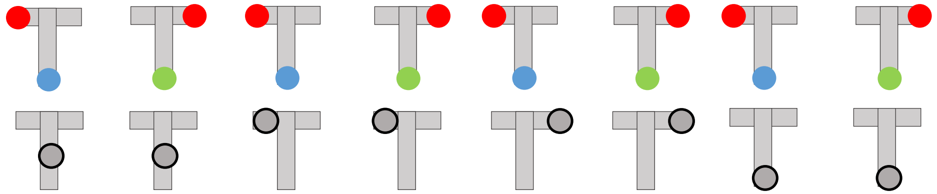
$$\mathcal{A} = \mathcal{P}(o_t|s_t) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.95 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0.95 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.05 & 0.95 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.95 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



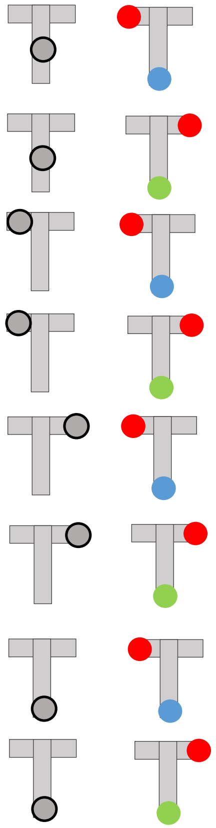
Outcomes

ABC of choice behaviour:
B – Transition Probabilities

Hidden states: Context, Location



$$\mathcal{B}\{\ \} = \mathcal{P}(s_{t+1} | s_t, \ ) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



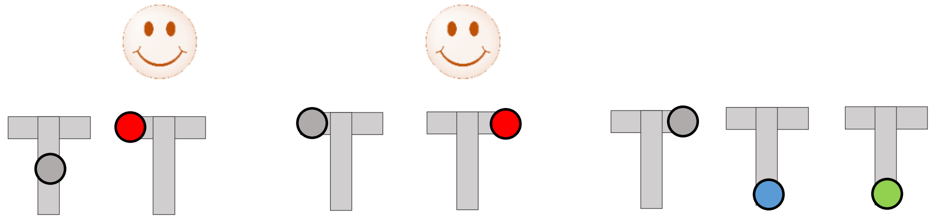
Hidden states

Also: $\mathcal{B}\{\ \}$, $\mathcal{B}\{\ \}$, $\mathcal{B}\{\ \}$ ...

CCNS

UCL

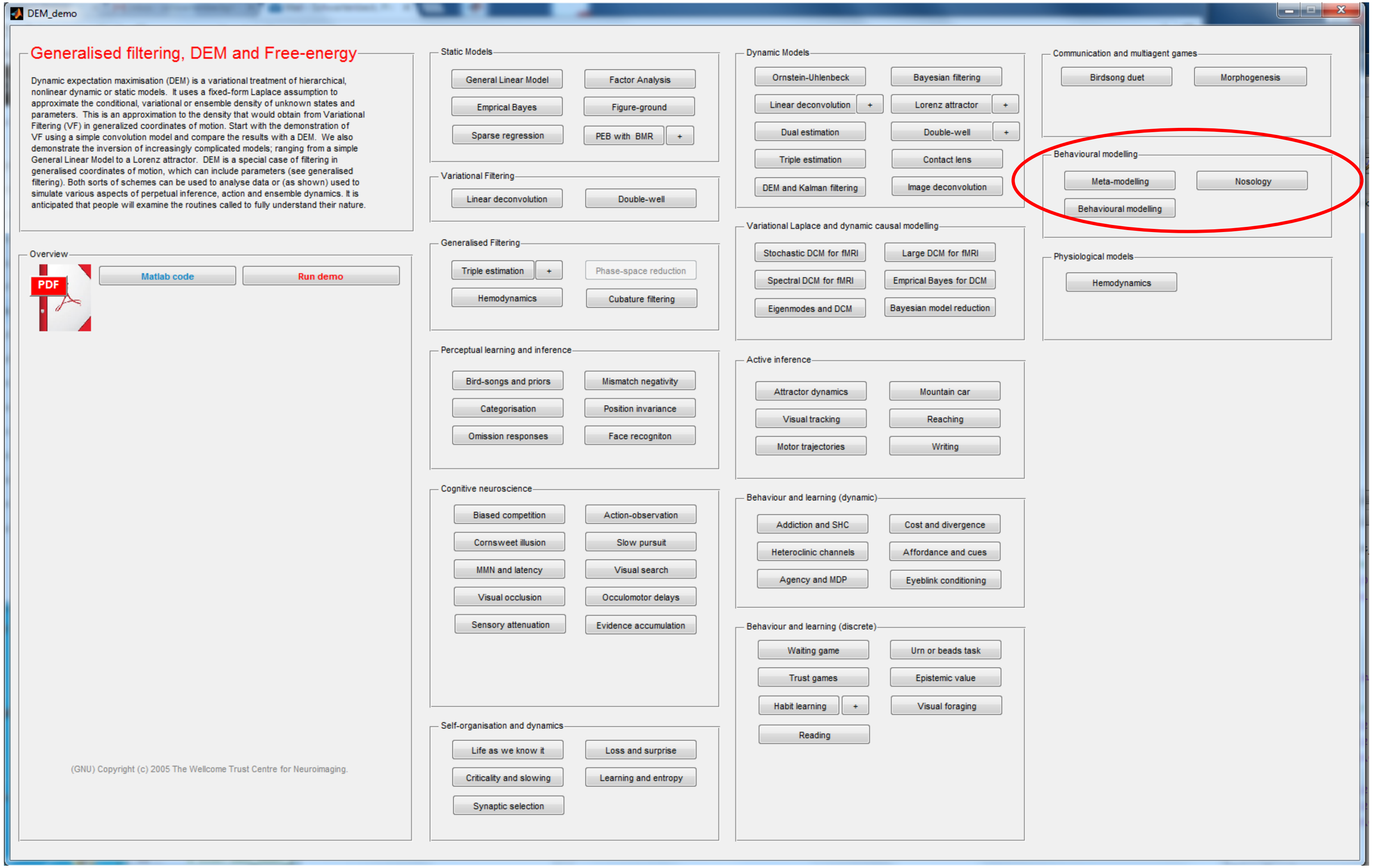
ABC of choice behaviour:
C – Preferences over Outcomes



Outcomes

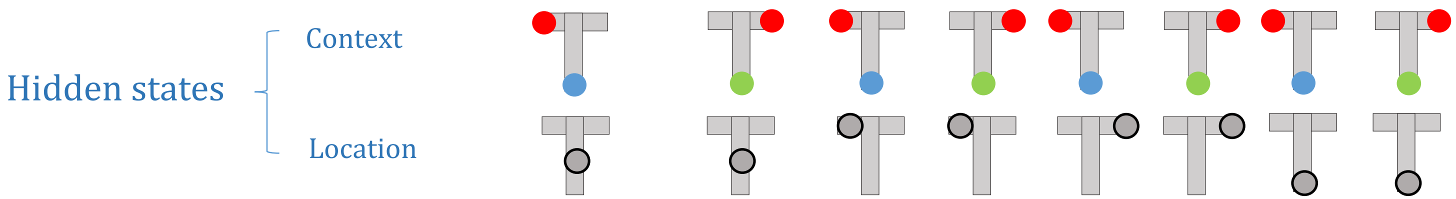
$$c = ln\mathcal{P}(o_t) = \begin{bmatrix} 0 & 2 & -2 & 2 & -2 & 0 & 0 \end{bmatrix}^T$$

DEM Toolbox



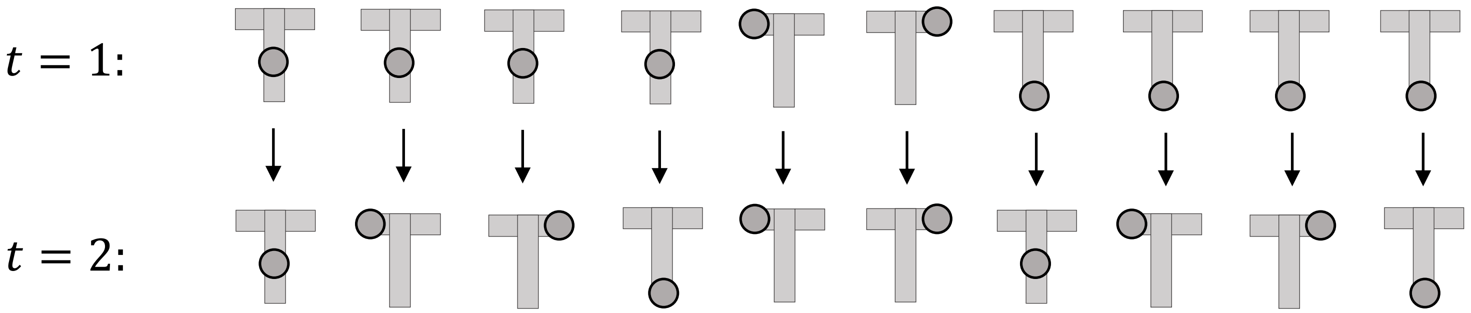
Schwartenbeck & Friston, 2016 *eNeuro*

ABC of choice behaviour:
Prior over initial states and allowable policies



$$d = \mathcal{P}(s_0) = \begin{bmatrix}0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T$$

$$\mathcal{V} = \begin{bmatrix}1 & 1 & 1 & 1 & 2 & 3 & 4 & 4 & 4 & 4 \\ 1 & 2 & 3 & 4 & 2 & 3 & 1 & 2 & 3 & 4\end{bmatrix}$$



I. **ABC of choice behaviour**

II. Simulate data and invert model to obtain subject-specific parameters

III. Simulate group effect and recover group-specific parameters

IV. Perform hierarchical Bayesian inference on group effects


```
%% set up and preliminaries: first generate synthetic (single subject) data
%==========================================================================
rng('default')

% outcome probabilities: A
%--------------------------------------------------------------------------
% We start by specifying the probabilistic mapping from hidden states
% to outcomes.
%--------------------------------------------------------------------------
a      = .95;
b      = 1 - a;
A      = [1 1 0 0 0 0 0 0;    % ambiguous starting position (centre)
          0 0 a b 0 0 0 0;    % left arm selected and rewarded
          0 0 b a 0 0 0 0;    % left arm selected and not rewarded
          0 0 0 0 b a 0 0;    % right arm selected and rewarded
          0 0 0 0 a b 0 0;    % right arm selected and not rewarded
          0 0 0 0 0 0 1 0;    % informative cue - reward on right
          0 0 0 0 0 0 0 1];   % informative cue - reward on left

% controlled transitions: B{u}
%--------------------------------------------------------------------------
% Next, we have to specify the probabilistic transitions of hidden states
% under each action or control state. Here, there are four actions taking the
% agent directly to each of the four locations.
%--------------------------------------------------------------------------
B{1}  = [1 0 0 1; 0 1 0 0;0 0 1 0;0 0 0 0];     % move to the middle
B{2}  = [0 0 0 0; 1 1 0 1;0 0 1 0;0 0 0 0];     % move up left  (and check for reward)
B{3}  = [0 0 0 0; 0 1 0 0;1 0 1 1;0 0 0 0];     % move up right (and check for reward)
B{4}  = [0 0 0 0; 0 1 0 0;0 0 1 0;1 0 0 1];     % move down     (check cue)
for i = 1:4
    B{i} = kron(B{i},eye(2));
end
```

```
% priors: (utility) C
%--------------------------------------------------------------------------
% Finally, we have to specify the prior preferences in terms of log
% probabilities. Here, the agent prefers rewarding outcomes
%--------------------------------------------------------------------------
c  = 2;
C  = [0 c -c c -c 0 0]';

% now specify prior beliefs about initial state, in terms of counts
%--------------------------------------------------------------------------
d  = kron([1 0 0 0],[1 1])';

% allowable policies (of depth T).  These are just sequences of actions
%--------------------------------------------------------------------------
V  = [1  1  1  1  2  3  4  4  4  4
      1  2  3  4  2  3  1  2  3  4];
```

```
%% MDP Structure - this will be used to generate arrays for multiple trials
%==========================================================================
mdp.V = V;                    % allowable policies
mdp.A = A;                    % observation model
mdp.B = B;                    % transition probabilities
mdp.C = C;                    % preferred states
mdp.D = d;                    % prior over initial states
mdp.s = 1;                    % initial state

mdp.alpha = 2;                % precision of action selection
% mdp.alpha = 8;                % precision of action selection
mdp.beta  = 1;                % inverse precision of policy selection

% true parameters
%--------------------------------------------------------------------------
n          = 128;             % number of trials
i          = rand(1,n) > 1/2; % randomise hidden states over trials
P.beta     = log(2);
P.C        = log(2);

MDP        = mdp;
% MDP.C      = mdp.C;
MDP.C      = mdp.C*exp(P.C);
MDP.beta   = mdp.beta*exp(P.beta);


[MDP(1:n)] = deal(MDP);
[MDP(i).s] = deal(2);
```

CCNS

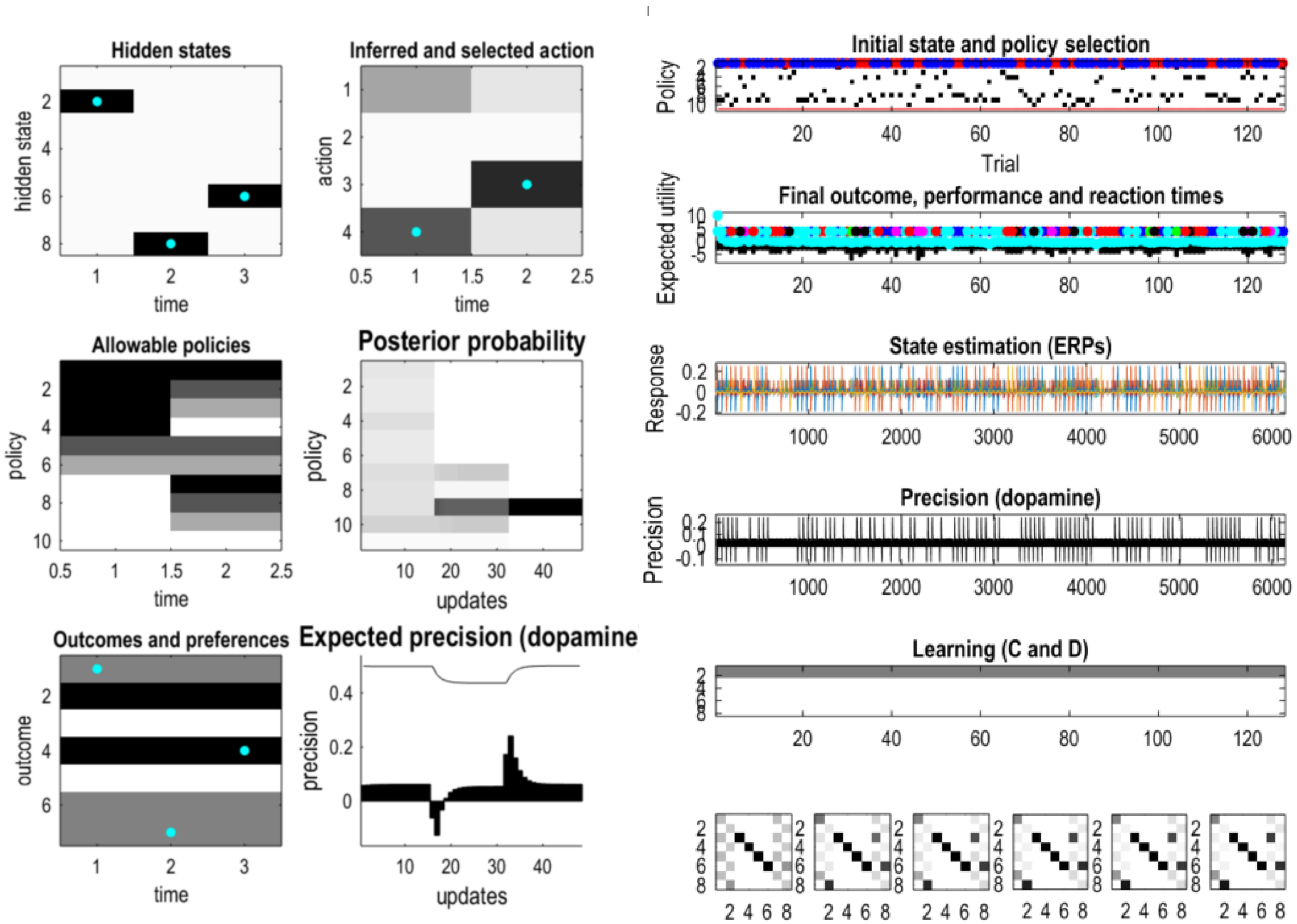
I. ABC of choice behaviour

II. Simulate data and invert model to obtain subject-specific parameters **(spm_MDP_VB)**

III. Simulate group effect and recover group-specific parameters

IV. Perform hierarchical Bayesian inference on group effects

```
%% Solve to generate data
%==========================================================================
MDP  = spm_MDP_VB(MDP);

% illustrate behavioural responses - single trial
%--------------------------------------------------------------------------
spm_figure('GetWin','Figure 1a'); clf
spm_MDP_VB_trial(MDP(1));

% illustrate behavioural responses and neuronal correlates over trials
%--------------------------------------------------------------------------
spm_figure('GetWin','Figure 1b'); clf
spm_MDP_VB_game(MDP);

%--------------------------------------------------------------------------
% This completes the generation of data. We now turn to the estimation of
% subject specific preferences and precision encoded by the parameters
% beta and C. Model parameters here are log scaling parameters that allow
% for increases or decreases in the default prior values.
%--------------------------------------------------------------------------
```



CCNS

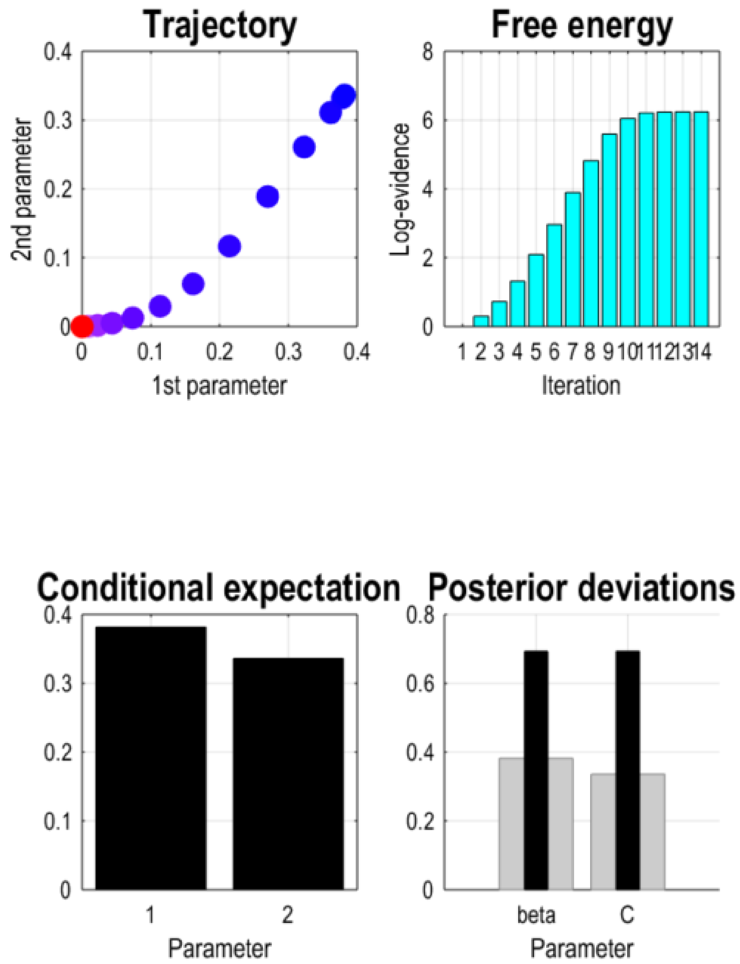
I. ABC of choice behaviour

II. Simulate data and **invert model to obtain subject-specific parameters (spm_dcm_mdp)**

III. Simulate group effect and recover group-specific parameters

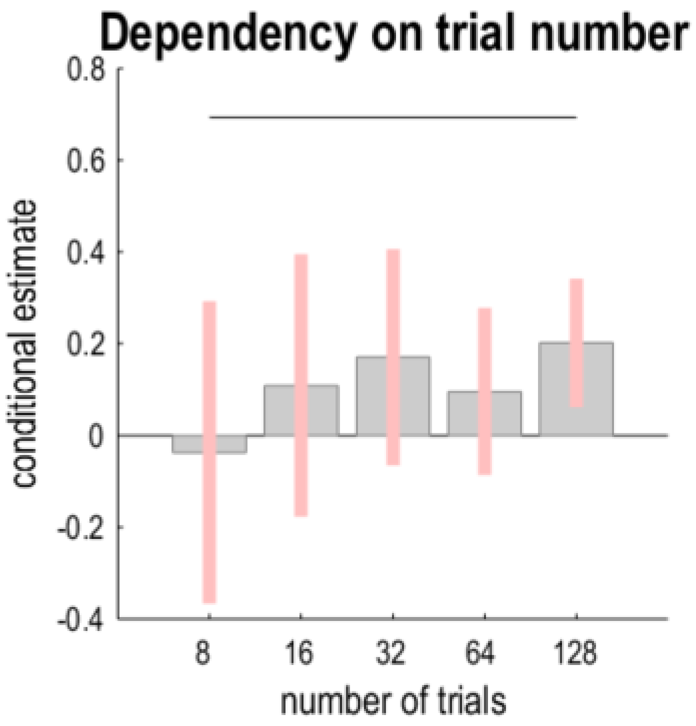
IV. Perform hierarchical Bayesian inference on group effects

```
%% Invert to recover parameters (preferences and precision)
%==========================================================================
DCM.MDP   = mdp;                  % MDP model
DCM.field = {'beta','C'};         % parameter (field) names to optimise
DCM.U     = {MDP.o};              % trial specification (stimuli)
DCM.Y     = {MDP.u};              % responses (action)

DCM       = spm_dcm_mdp(DCM);

% compare true values with posterior estimates
%--------------------------------------------------------------------------
subplot(2,2,4),hold on
bar(spm_vec(P),1/4)
set(gca,'XTickLabel',DCM.field)
set(gcf,'Name','Figure 2','Tag','Figure 2')
```



Optimise design parameters (trial number)

```
%% now repeat using subsets of trials to illustrate effects on estimators - design optimisation!
%==========================================================================
DCM.field = {'beta'};
n         = [8 16 32 64 128];
for i = 1:length(n)
    DCM.U = {MDP(1:n(i)).o};
    DCM.Y = {MDP(1:n(i)).u};
    DCM   = spm_dcm_mdp(DCM);
    Ep(i,1) = DCM.Ep.beta;
    Cp(i,1) = DCM.Cp;
end

% plus results
%--------------------------------------------------------------------------
spm_figure('GetWin','Figure 3'); clf
subplot(2,1,1), spm_plot_ci(Ep(:),Cp(:)), hold on
plot(1:length(n),(n - n) + P.beta,'k'),       hold off
set(gca,'XTickLabel',n)
xlabel('number of trials','FontSize',12)
ylabel('conditional estimate','FontSize',12)
title('Dependency on trial number','FontSize',16)
axis square
```



CCNS

UCL

I. ABC of choice behaviour

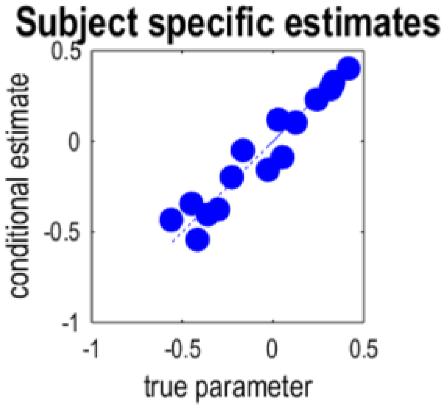
II. Simulate data and invert model to obtain subject-specific parameters

III. Simulate group effect and **recover group-specific parameters (spm_dcm_fit)**

IV. Perform hierarchical Bayesian inference on group effects

```
%% now repeat but over multiple subjects with different beta
%==========================================================================

% generate data and a between subject model with two groups of eight
% subjects
%--------------------------------------------------------------------------
N     = 8;                            % numbers of subjects per group
X     = kron([1 1;1 -1],ones(N,1));   % design matrix
h     = 4;                            % between subject log precision
n     = 128;                          % number of trials
i     = rand(1,n) > 1/2;              % randomise hidden states

clear MDP
[MDP(1:n)] = deal(mdp);
[MDP(i).s] = deal(2);

reward = zeros(n,size(X,1));

for i = 1:size(X,1)

    % true parameters - with a group difference of one half
    %----------------------------------------------------------------------
    beta(i)     = X(i,:)*[0; 1/4] + exp(-h/2)*randn;          % add random Gaussian effects to group means -> BMR and PEB
%     beta(i)     = X(i,:)*[0; 0] + exp(-h/2)*randn;          % add random Gaussian effects to group means -> BMR and PEB
    [MDP.beta] = deal(exp(beta(i)));

    % solve to generate data
   %-----------------------------------------------------------------------
    DDP         = spm_MDP_VB(MDP);      % realisation for this subject
    DCM.U       = {DDP.o};              % trial specification (stimuli)
    DCM.Y       = {DDP.u};              % responses (action)
    GCM{i,1}    = DCM;

    for kk=1:length(DCM.U)
        if DCM.U{kk}(end)==2 || DCM.U{kk}(end)==4 % outcome 2 or 4 == reward
            reward(kk,i)=1;
        end
    end

    % plot behavioural responses
    %----------------------------------------------------------------------
    spm_figure('GetWin','Figure 4'); clf
    spm_MDP_VB_game(DDP);drawnow

end
```

```
%% Bayesian model inversion
%==========================================================================
GCM   = spm_dcm_fit(GCM);

% plot subject specific estimates and true values
%--------------------------------------------------------------------------
spm_figure('GetWin','Figure 4');
subplot(3,1,3)
for i = 1:length(GCM)
    qP(i) = GCM{i}.Ep.beta;
end
plot(beta,beta,':b',beta,qP,'.b','MarkerSize',32)
xlabel('true parameter','FontSize',12)
ylabel('conditional estimate','FontSize',12)
title('Subject specific estimates','FontSize',16)
axis square
```



CCNS

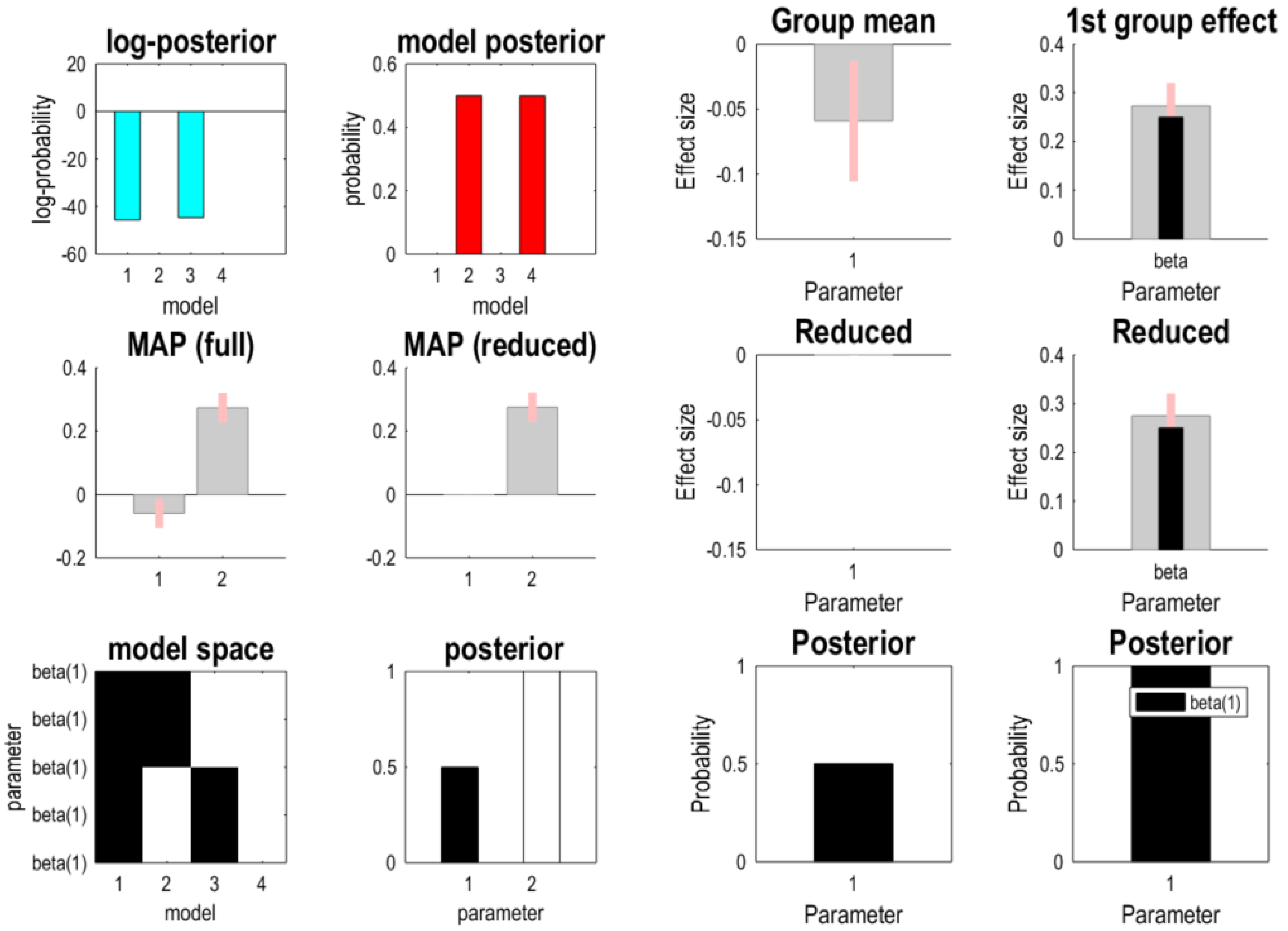
I. ABC of choice behaviour

II. Simulate data and invert model to obtain subject-specific parameters

III. Simulate group effect and recover group-specific parameters

IV. Perform hierarchical Bayesian inference on group effects (spm_dcm_peb)

```
%% hierarchical (empirical) Bayes
%==========================================================================

% second level model
%--------------------------------------------------------------------------
M    = struct('X',X);

% BMA - (second level)
%--------------------------------------------------------------------------
PEB  = spm_dcm_peb(GCM,M);
BMA  = spm_dcm_peb_bmc(PEB);

subplot(3,2,4),hold on, bar(1,1/4,1/4), set(gca,'XTickLabel',DCM.field)
subplot(3,2,2),hold on, bar(1,1/4,1/4), set(gca,'XTickLabel',DCM.field)
```



CCNS

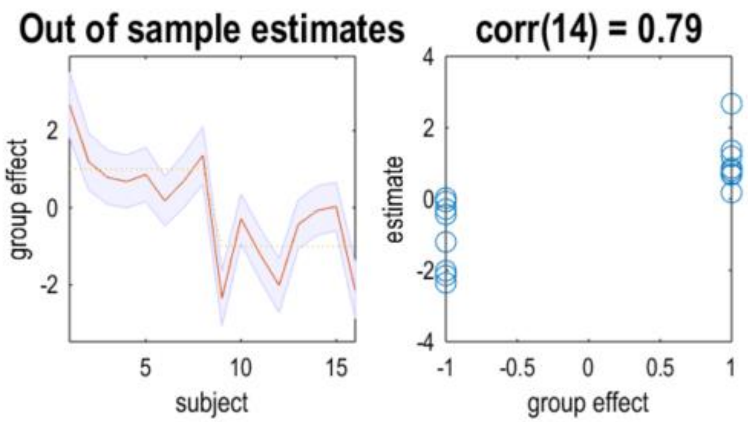
UCL

I. ABC of choice behaviour

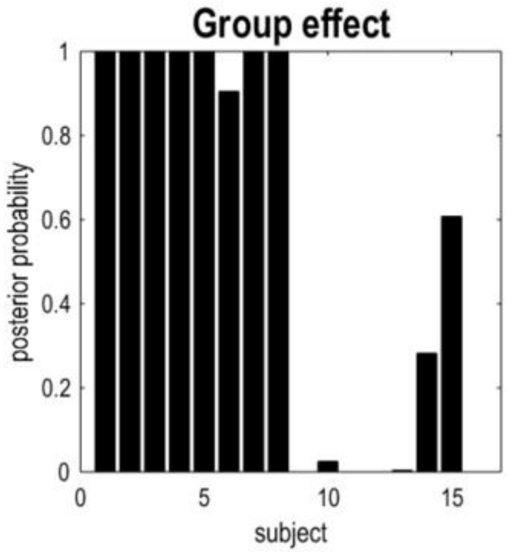
II. Simulate data and invert model to obtain subject-specific parameters

III. Simulate group effect and recover group-specific parameters

IV. Perform hierarchical Bayesian inference on group effects and **leave-one-out cross-validation**
(spm_dcm_loo)




```
%% posterior predictive density and cross validation
%==========================================================================
spm_dcm_loo(GCM,M,DCM.field);
```



CCNS UCL

Active Inference and MDP toolbox - Summary

Active inference introduces concept of surprise minimisation to modelling (choice) behaviour

- Central role of *subjective* and *objective* generative models

Using the toolbox essentially requires defining the state space of a task

- ABC of choice behaviour

Computational phenotyping in Psychiatry:

- Routines for simulating data, inverting models and inferring group effects freely available in SPM (DEM toolbox)
- Test empirical predictions, such as role of precision, expected utility vs. surprise minimisation or Bayesian model reduction

CCNS UCL

Active Inference and MDP toolbox - Summary

1. Why is this model useful?
   - (Choice) behaviour as probabilistic inference
   - Distinction between inference and learning

2. Where can you use it?
   - Any Inference or planning problem, particularly choice behaviour
   - Any learning problem (more recent version)

3. Where can't you use it?
   - Continuous problems (cf., predictive coding)

4. What do we like about it?
   - Applies normative theory of belief-updating (*Free Energy Principle*) to choice behaviour
   - Highlights importance of prior beliefs to understand (suboptimal) behaviour
   - Highlights necessity to build and update generative models of the world

5. What are the most common mistakes?
   - Definition of state space

CCNS UCL

Active Inference and MDP - Overview

Theory

- ‘Anatomy of choice’: MDP as surprise minimisation (*Friston et al., 2013*)
- Epistemic value: exploration versus exploitation (*Friston et al., 2015*)
- Active inference and learning (*Friston et al., 2016*)
- Artificial insight (*Friston et al., in prep*)
- \+ Worked example for computational phenotyping (*Schwartenbeck & Friston, 2016*) and hierarchical Bayesian inference and Bayesian model reduction (*Friston et al., 2015*)

Experimental work

- Role of dopamine, model-based fMRI (*Cerebral Cortex, 2014*)
- Interpersonal inference & trust games (*Front Comp Neurosci, 2014*)
- Evidence accumulation (*Neural Computations, 2015*)
- Economic decision theory (*Scientific Reports, 2015*)
- Scene construction & visual foraging (*Front Comp Neurosci, 2016*)
- Model averaging and structure learning, model-based fMRI (in prep)
- …

Thanks to

Karl Friston (UCL)

Thomas FitzGerald (UEA)

Christoph Mathys (UCL)

Martin Kronbichler (CCNS)

Ray Dolan (UCL)

Thank you for listening!