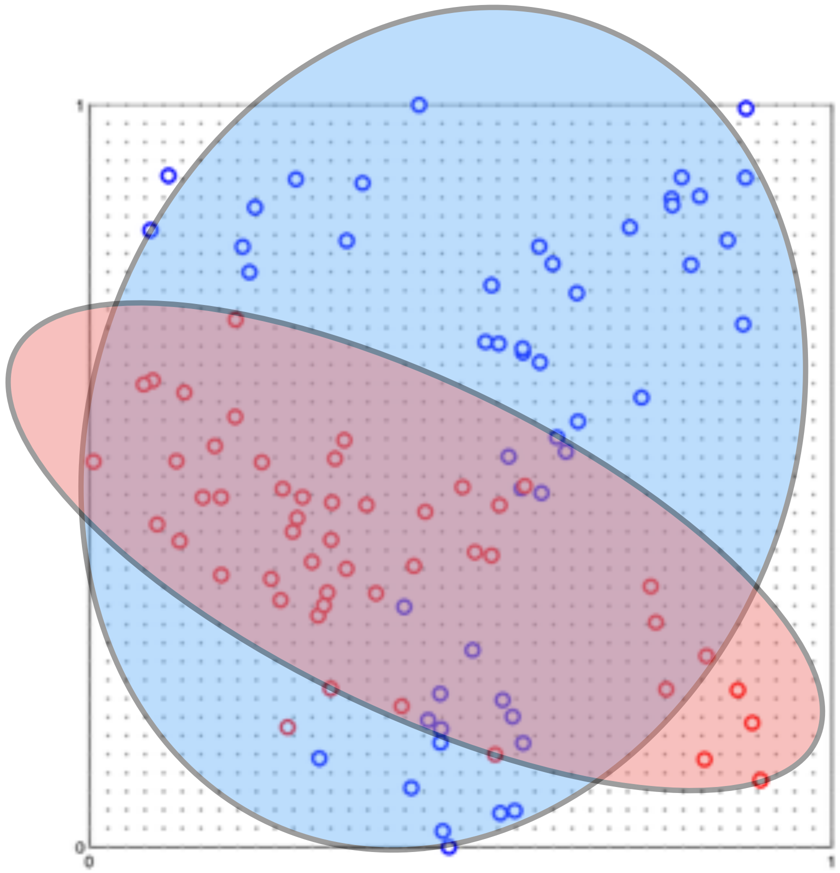
Semi-supervised learning

Computational Cognitive Science 2014

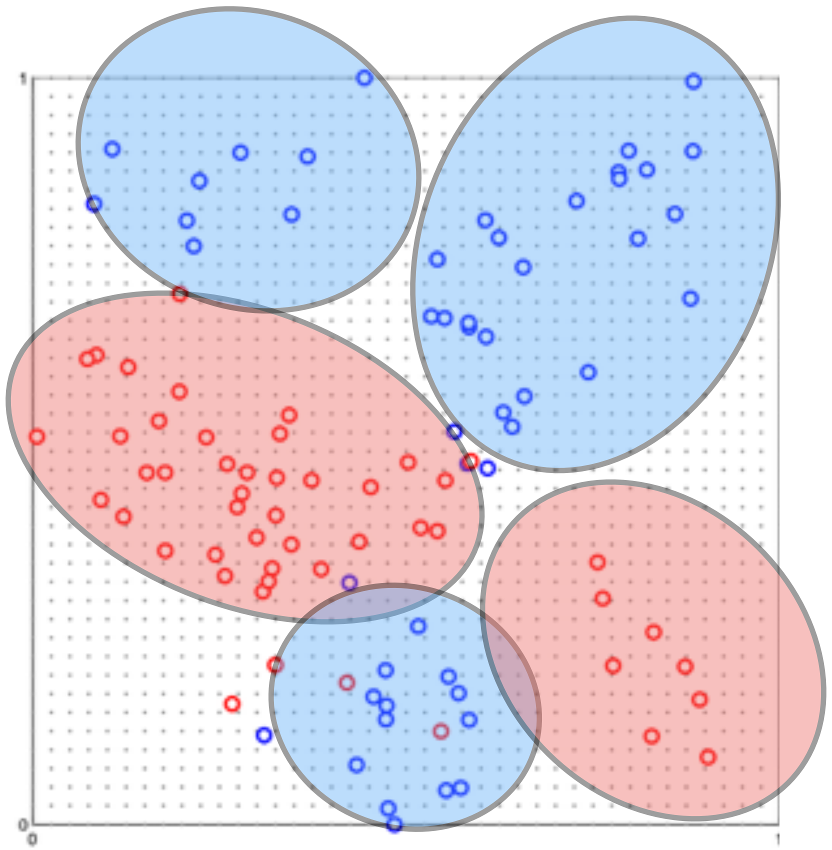
Dan Navarro

From last time… cluster learning



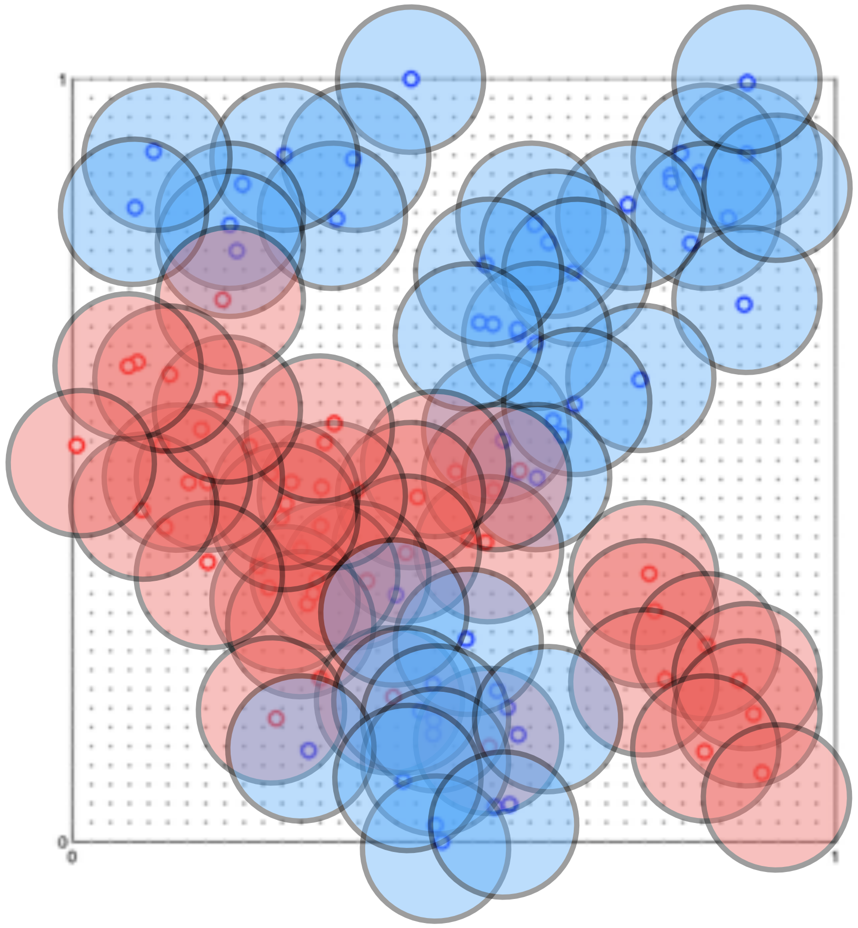
Prototype

Too few



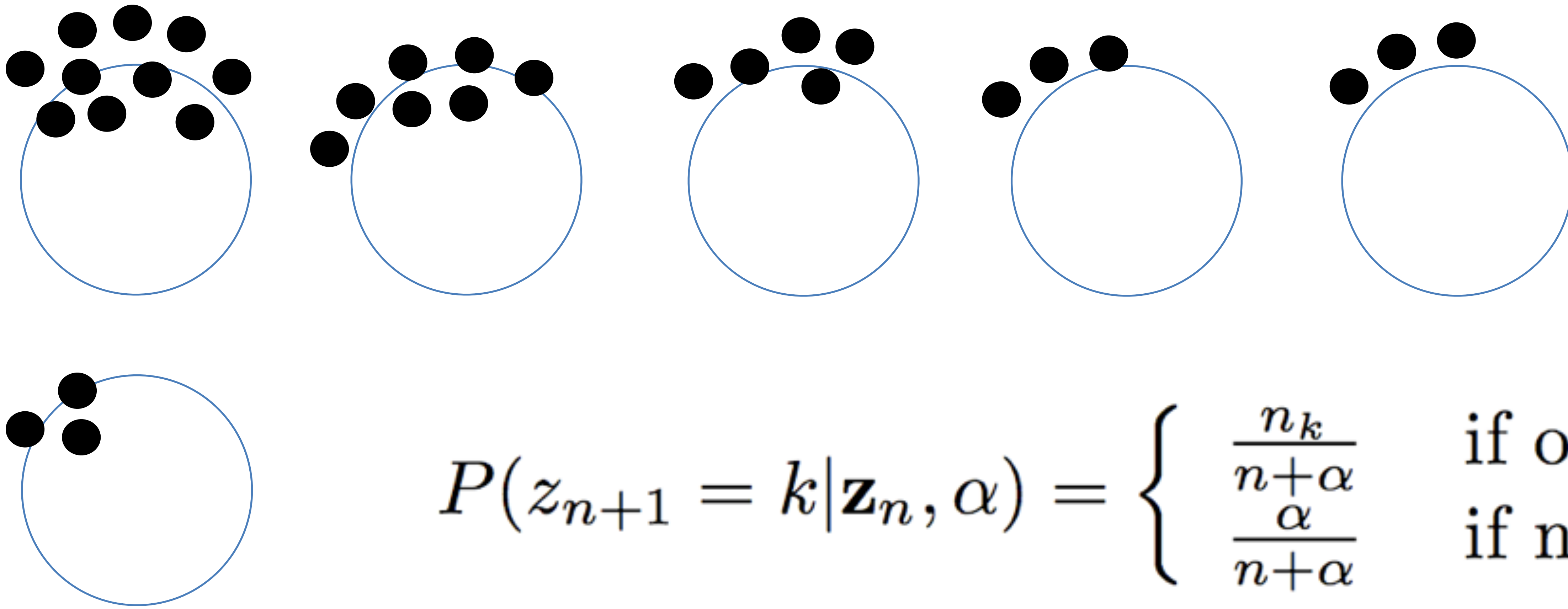
Cluster

Just right?



Exemplar

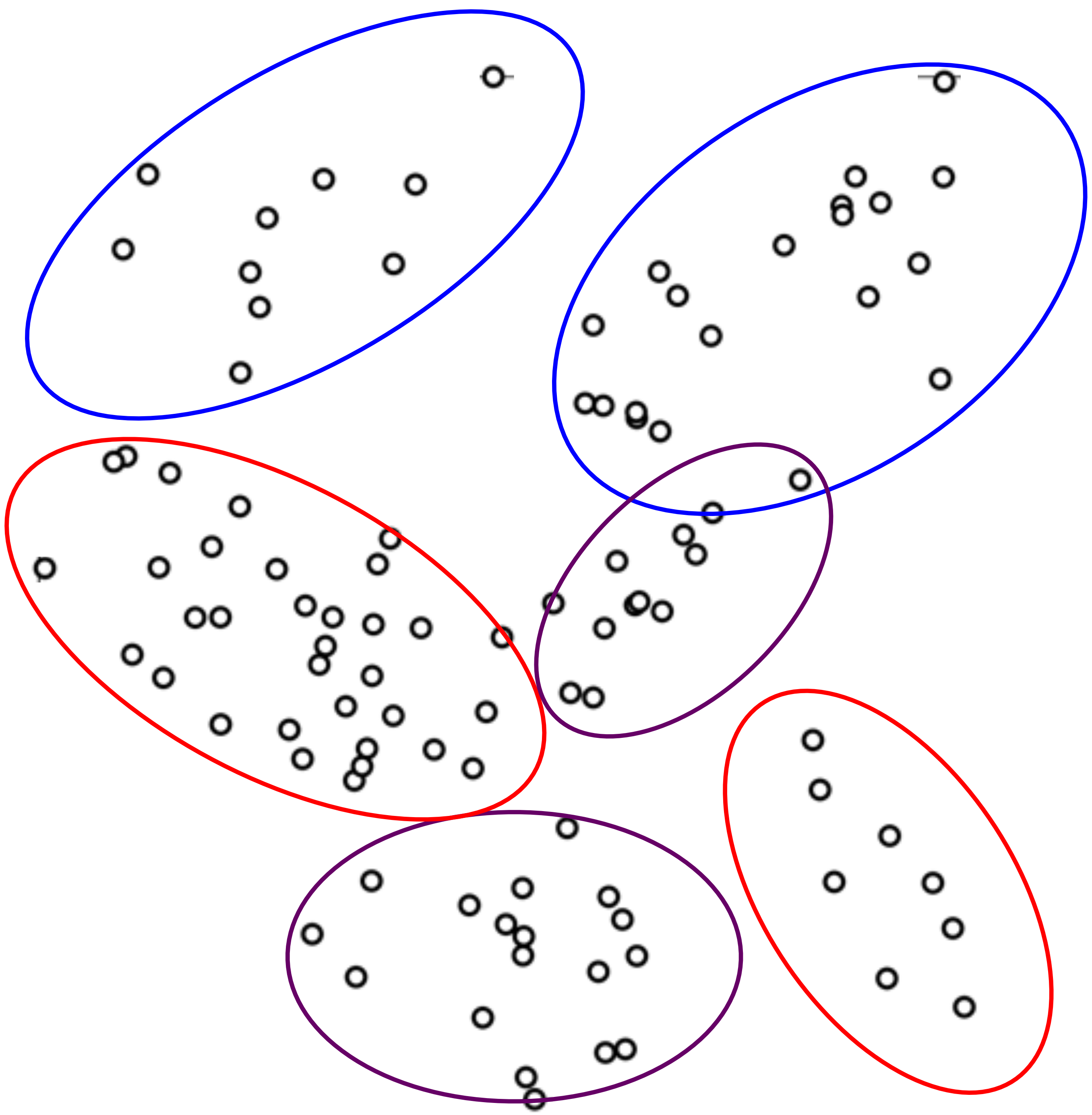
Too many



$$P(z_{n+1} = k | \mathbf{z}_n, \alpha) = \begin{cases} \frac{n_k}{n+\alpha} & \text{if old} \\ \frac{\alpha}{n+\alpha} & \text{if new} \end{cases}$$

…the CRP prior

We built an extended metaphor for tables with locations (means), shapes (covariances) and colours (labels)



Now we need to turn this idea into a proper classifier…

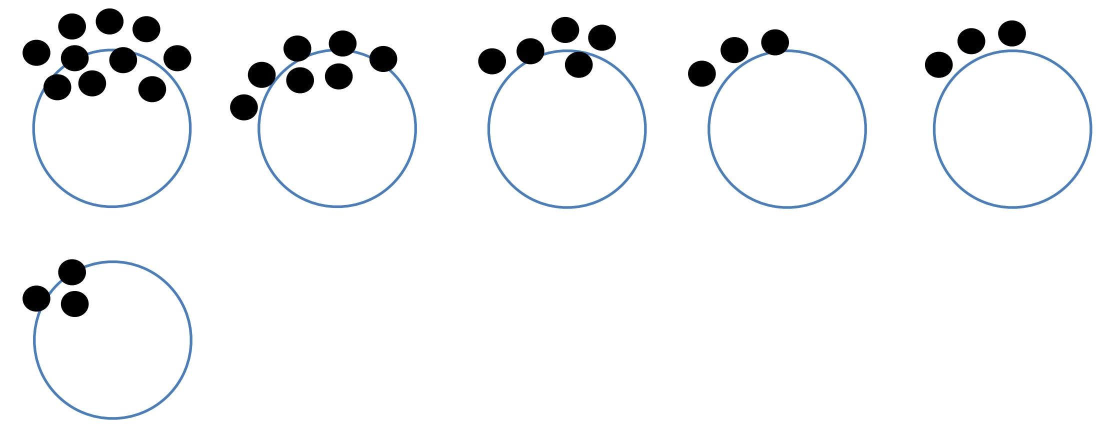
Formal statement of the model: all the ugly details

(not on the exam)

This is our story as a Bayesian model

$$
\begin{aligned}
\mathbf{z}|\alpha &\sim \mathrm{CRP}(\alpha) \\
\boldsymbol{\mu}_k &\sim \mathrm{Uniform} \\
\boldsymbol{\Sigma}_k &\sim \mathrm{Uniform} \\
\theta_k|\beta &\sim \mathrm{Beta}(\beta, \beta) \\
x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, z_i = k &\sim \mathrm{Normal}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
\ell_i|\theta_k, z_i = k &\sim \mathrm{Bernoulli}(\theta_k)
\end{aligned}
$$

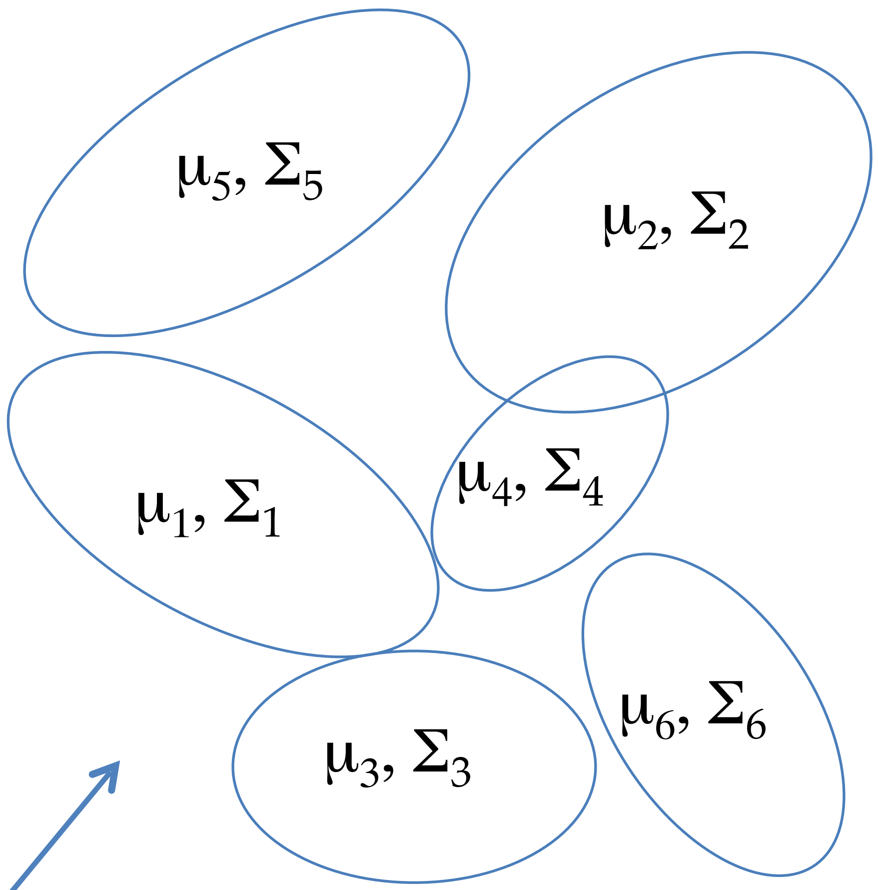
$$P(z_{n+1} = k | \mathbf{z}_n, \alpha) = \begin{cases} \frac{n_k}{n+\alpha} & \text{if old} \\ \frac{\alpha}{n+\alpha} & \text{if new} \end{cases}$$



$$\boxed{\mathbf{z}|\alpha \quad \sim \quad \mathrm{CRP}(\alpha)}$$

$$\begin{aligned} \boldsymbol{\mu}_k &\sim \mathrm{Uniform} \\ \boldsymbol{\Sigma}_k &\sim \mathrm{Uniform} \\ \theta_k|\beta &\sim \mathrm{Beta}(\beta, \beta) \\ x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, z_i = k &\sim \mathrm{Normal}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ \ell_i|\theta_k, z_i = k &\sim \mathrm{Bernoulli}(\theta_k) \end{aligned}$$

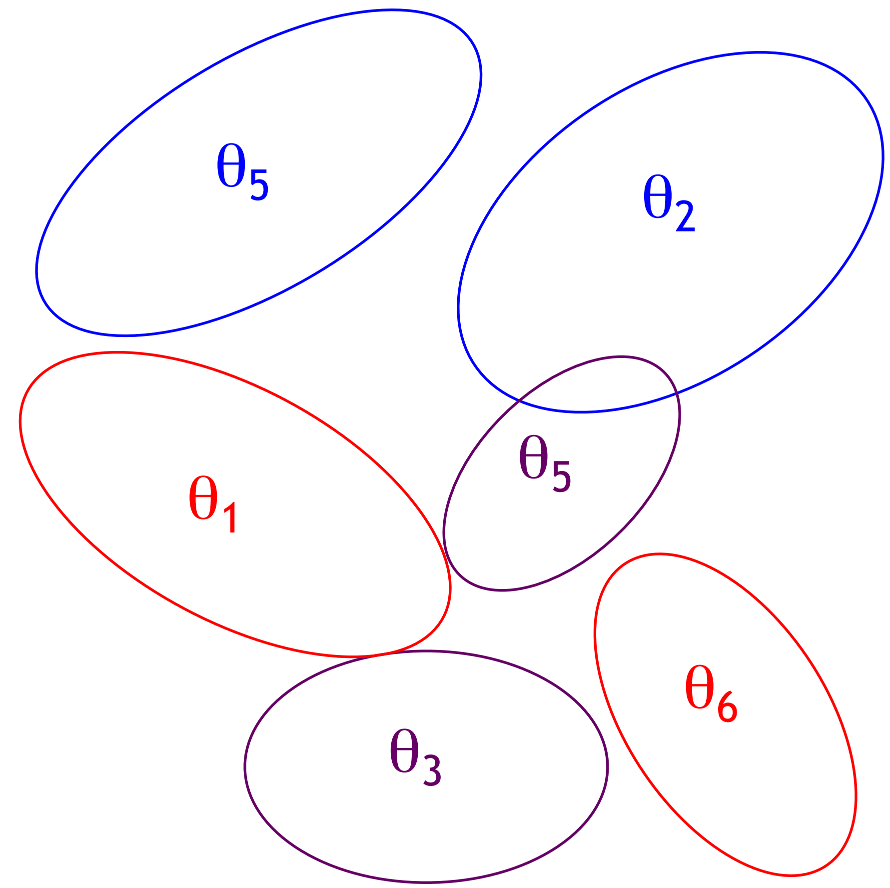
For simplicity, I'm going to assume that all possible cluster means μ and all possible (positive definite) covariance matrices Σ are equally likely. This is an extremely silly assumption, but I want to take some shortcuts later on, otherwise the maths gets tedious.



$$
\begin{aligned}
\mathbf{z}|\alpha &\sim \mathrm{CRP}(\alpha) \\
\boldsymbol{\mu}_k &\sim \mathrm{Uniform} \\
\boldsymbol{\Sigma}_k &\sim \mathrm{Uniform} \\
\theta_k|\beta &\sim \mathrm{Beta}(\beta, \beta) \\
x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, z_i = k &\sim \mathrm{Normal}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
\ell_i|\theta_k, z_i = k &\sim \mathrm{Bernoulli}(\theta_k)
\end{aligned}
$$

$$P(\theta|\beta) \propto \theta^{\beta-1}(1-\theta)^{\beta-1}$$

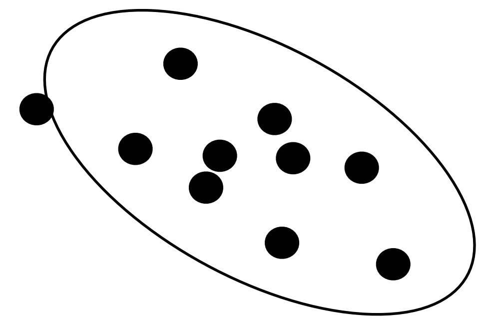
(Beta distribution)



$$\begin{aligned}
\mathbf{z}|\alpha &\sim \mathrm{CRP}(\alpha) \\
\boldsymbol{\mu}_k &\sim \mathrm{Uniform} \\
\boldsymbol{\Sigma}_k &\sim \mathrm{Uniform} \\
\boxed{\theta_k|\beta \sim \mathrm{Beta}(\beta,\beta)} \\
x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, z_i = k &\sim \mathrm{Normal}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
\ell_i|\theta_k, z_i = k &\sim \mathrm{Bernoulli}(\theta_k)
\end{aligned}$$

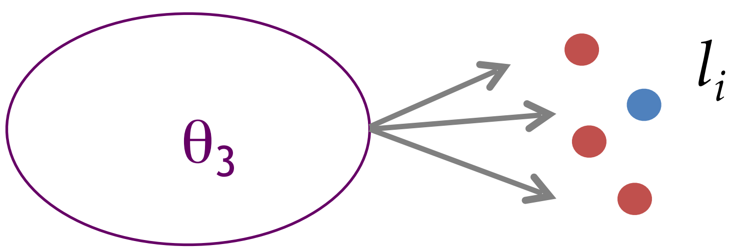
$$P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{k/2}\sqrt{\det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

(Multivariate normal)



$$\begin{aligned}
\mathbf{z}|\alpha &\sim \mathrm{CRP}(\alpha) \\
\boldsymbol{\mu}_k &\sim \mathrm{Uniform} \\
\boldsymbol{\Sigma}_k &\sim \mathrm{Uniform} \\
\theta_k|\beta &\sim \mathrm{Beta}(\beta, \beta) \\
x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, z_i = k &\sim \mathrm{Normal}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
\ell_i|\theta_k, z_i = k &\sim \mathrm{Bernoulli}(\theta_k)
\end{aligned}$$

$$P(\ell_i = 1) = \theta$$



$$\begin{aligned}
\mathbf{z}|\alpha &\sim \mathrm{CRP}(\alpha) \\
\boldsymbol{\mu}_k &\sim \mathrm{Uniform} \\
\boldsymbol{\Sigma}_k &\sim \mathrm{Uniform} \\
\theta_k|\beta &\sim \mathrm{Beta}(\beta, \beta) \\
x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, z_i = k &\sim \mathrm{Normal}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
\ell_i|\theta_k, z_i = k &\sim \mathrm{Bernoulli}(\theta_k)
\end{aligned}$$

How to work with this model

- There's the "right" way, which takes some effort to learn
  - Markov chain Monte Carlo (MCMC) and/or particle filtering
  - These are efficient algorithms that let you learn the complete posterior distribution over category labels, cluster assignments, cluster means and cluster covariances etc
  - We'll introduce these algorithms towards the end of the class
- For now, let's pick some good-enough methods:
  - A sequential assignment algorithm: "poor man's particle filtering"
  - An iterative "simulated annealing" method: similar to Bayesian MCMC, but we're cheating in a few places.

A sequential assignment algorithm

```
"base" mean set to the mean of all training observations
"base" covariance matrix set to the covariance of all training observations
"base" probability of all labels set to be equally likely

assign observation 1 to cluster 1
compute the current mean and covariance for cluster 1
compute the estimate the label probabilities for cluster 1

for( O in 2:N ) {

   compute the prior probability of all existing clusters using CRP
   compute the prior probability of a new cluster using the CRP

   compute the likelihood of observation O under all clusters (multivariate normal)
   compute the likelihood of observation O under a new cluster (using base distribution)
   compute the likelihood of label of O under all clusters, including new one

   convert prior + likelihood to a posterior distribution over clusters
   use posterior distribution to select a cluster (denoted K) for observation O
   assign observation O to cluster K

   update the mean and covariance for cluster K
   update the label probabilities for cluster K

}
```

```
"base" mean set to the mean of all training observations
"base" covariance matrix set to the covariance of all training observations
"base" probability of all labels set to be equally likely

assign observation 1 to cluster 1
compute the curr
compute the est

for( O in 2:N ) 

   compute the p
   compute the p

   compute the l                                                  ariate normal)
   compute the l                                                  base distribution)
   compute the likelihood of label of O under all clusters, including new one

   convert prior + likelihood to a posterior distribution over clusters
   use posterior distribution to select a cluster (denoted K) for observation O
   assign observation O to cluster K

   update the mean and covariance for cluster K
   update the label probabilities for cluster K

}
```

There are some “substantive” choices we need to make for these steps in particular. Read the demonstration code: the comments explain a lot about what the different choices imply

A sequential **re**assignment algorithm
(with the simulated annealing trick)

```
start with assignments from the last algorithm
set temperature T high

while( not bored yet) {

 lower temperature T a little bit

 for( O in 1:N ) {

   compute CRP priors
   compute Gaussian part of the likelihood
   compute the label probability part of the likelihood

   compute posterior
   select cluster K from posterior (at temperature T)
   assign observation O to cluster K

   update the mean and covariance for cluster K
   update the label probabilities for cluster K

  }
}
```

```
start with assignments from the
set temperature T high

while( not bored yet) {

 lower temperature T a little b

 for( O in 1:N ) {

   compute CRP priors
   compute Gaussian part of the
   compute the label probabilit

   compute posterior
   select cluster K from poster
   assign observation O to clus

   update the mean and covarian
   update the label probabiliti

  }
}
```

At **high temperature**, we impose a weak bias towards selecting high posterior clusters: encourages exploration of the space of possible clusterings

At **low temperature**, we impose a strong bias towards selecting high probability clusters: encourages the algorithm to settle on better possibilities

Grr! This is **so** close to being the right thing to do, you lazy bastard.

Quiet, you. Wait until the last few lectures of the class, okay? This subject is theory-heavy enough as it is.

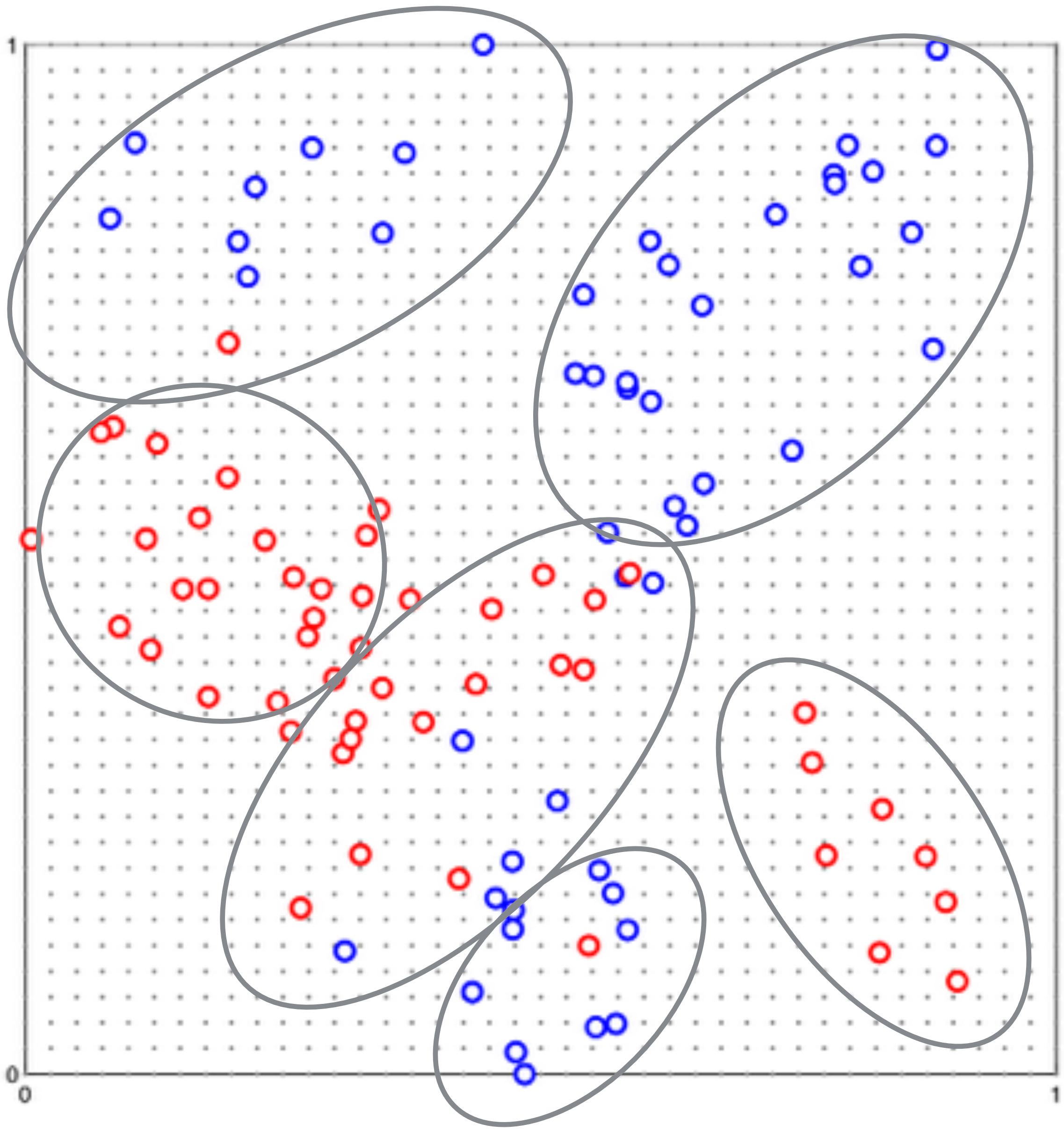
Application to our running example

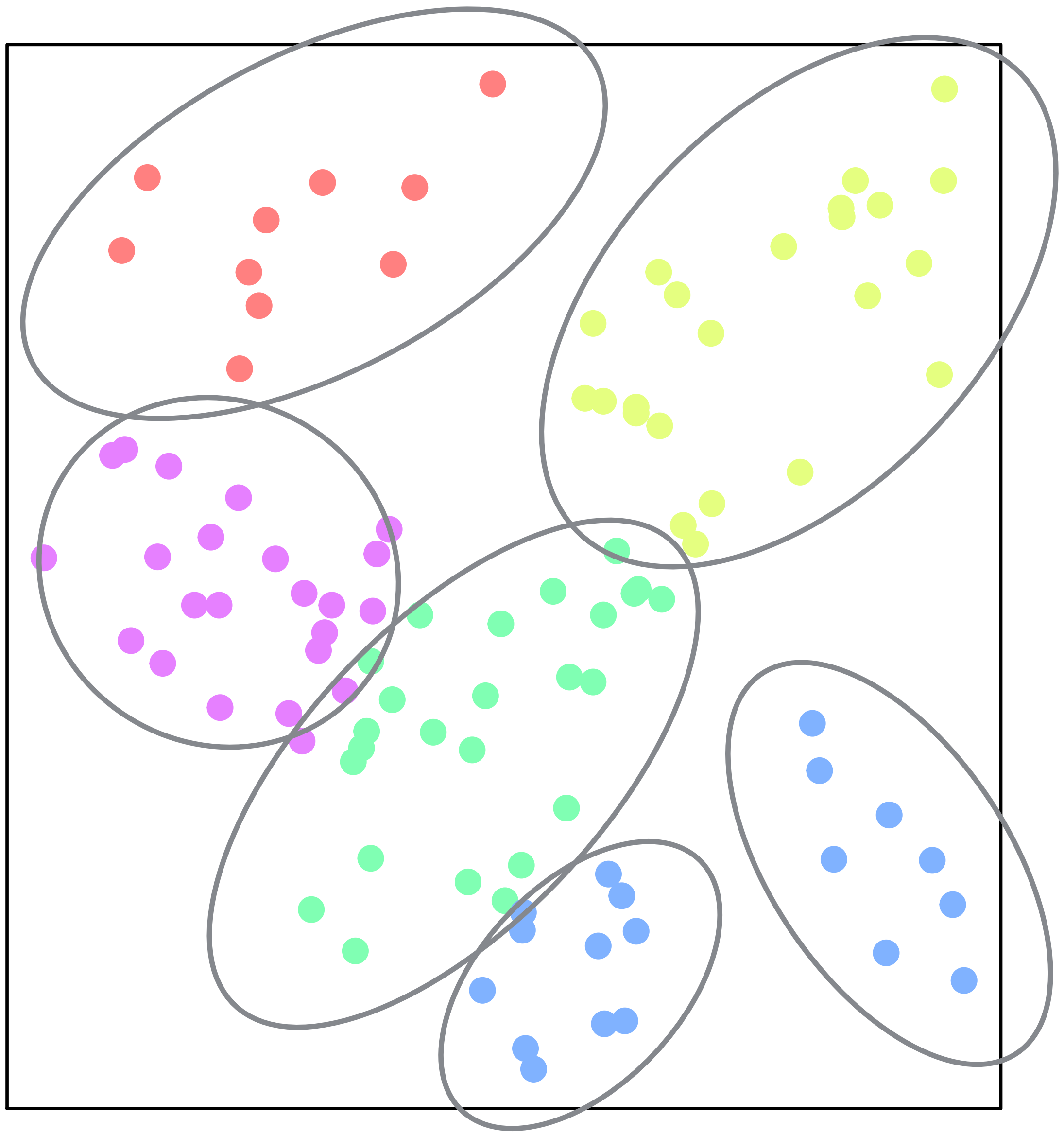
(classifiers.R, RMC function)

Application of the model to a problem in cognitive science

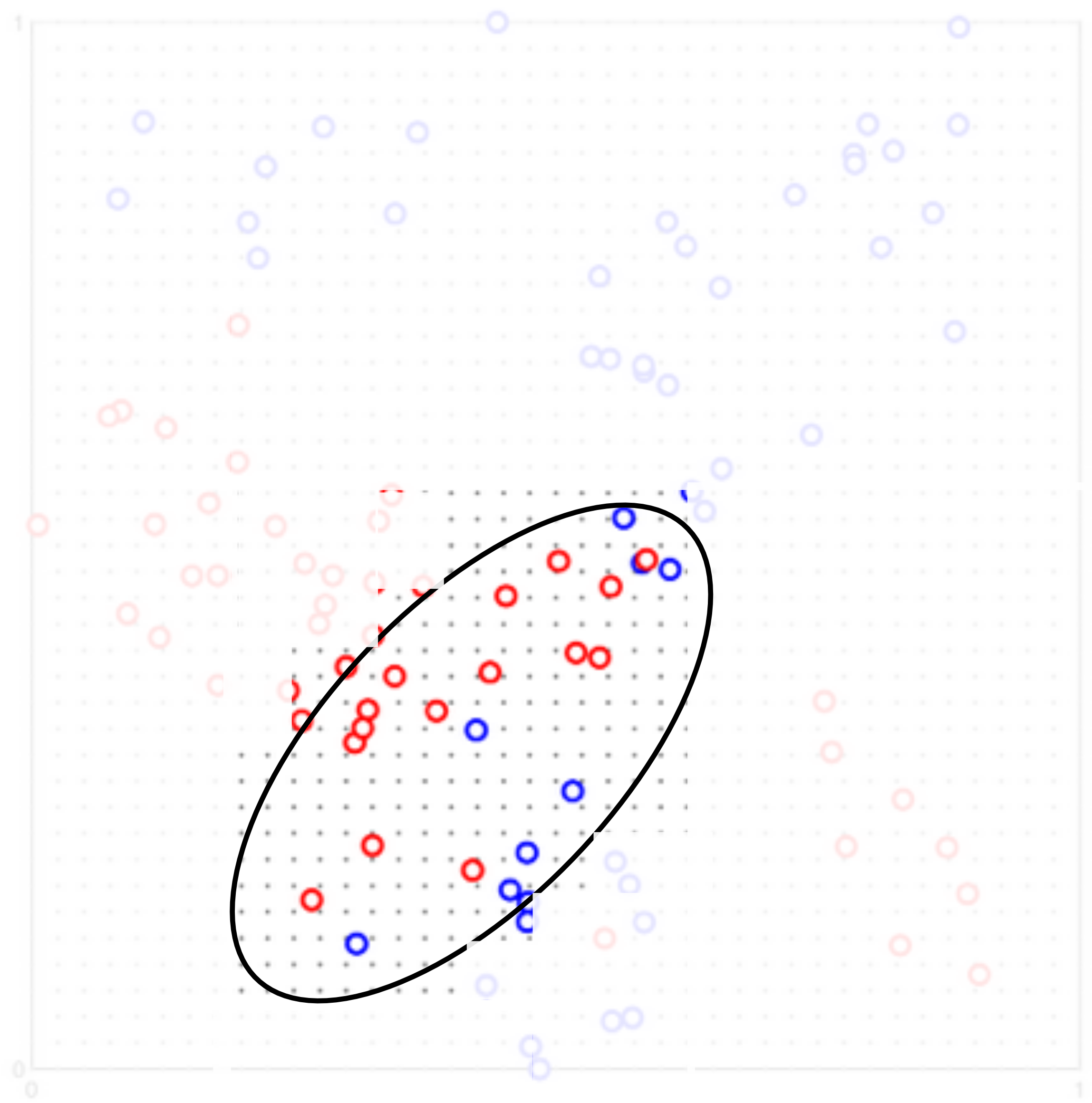
(Vong, Perfors & Navarro, under review)

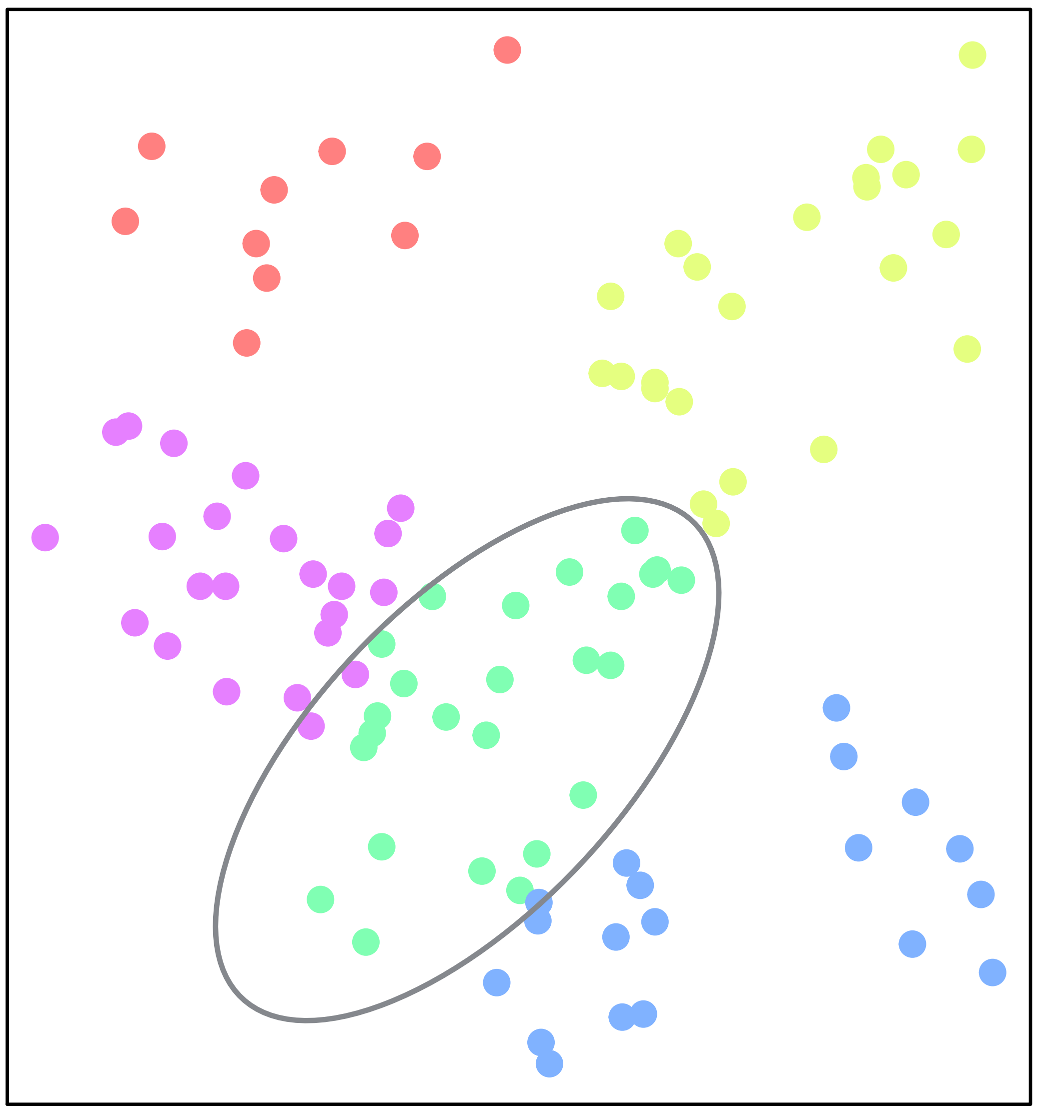
k-means works okay without labels!



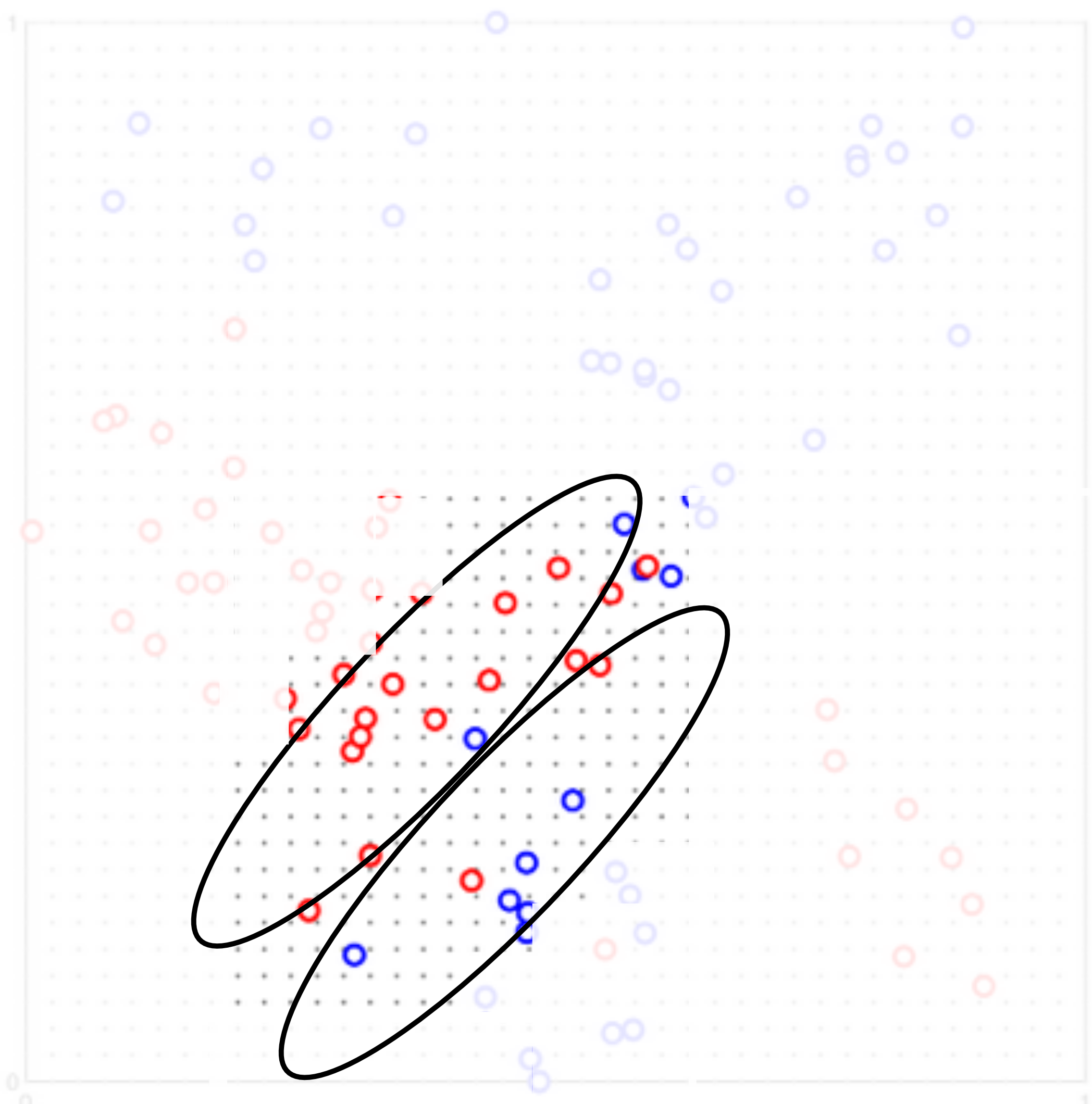


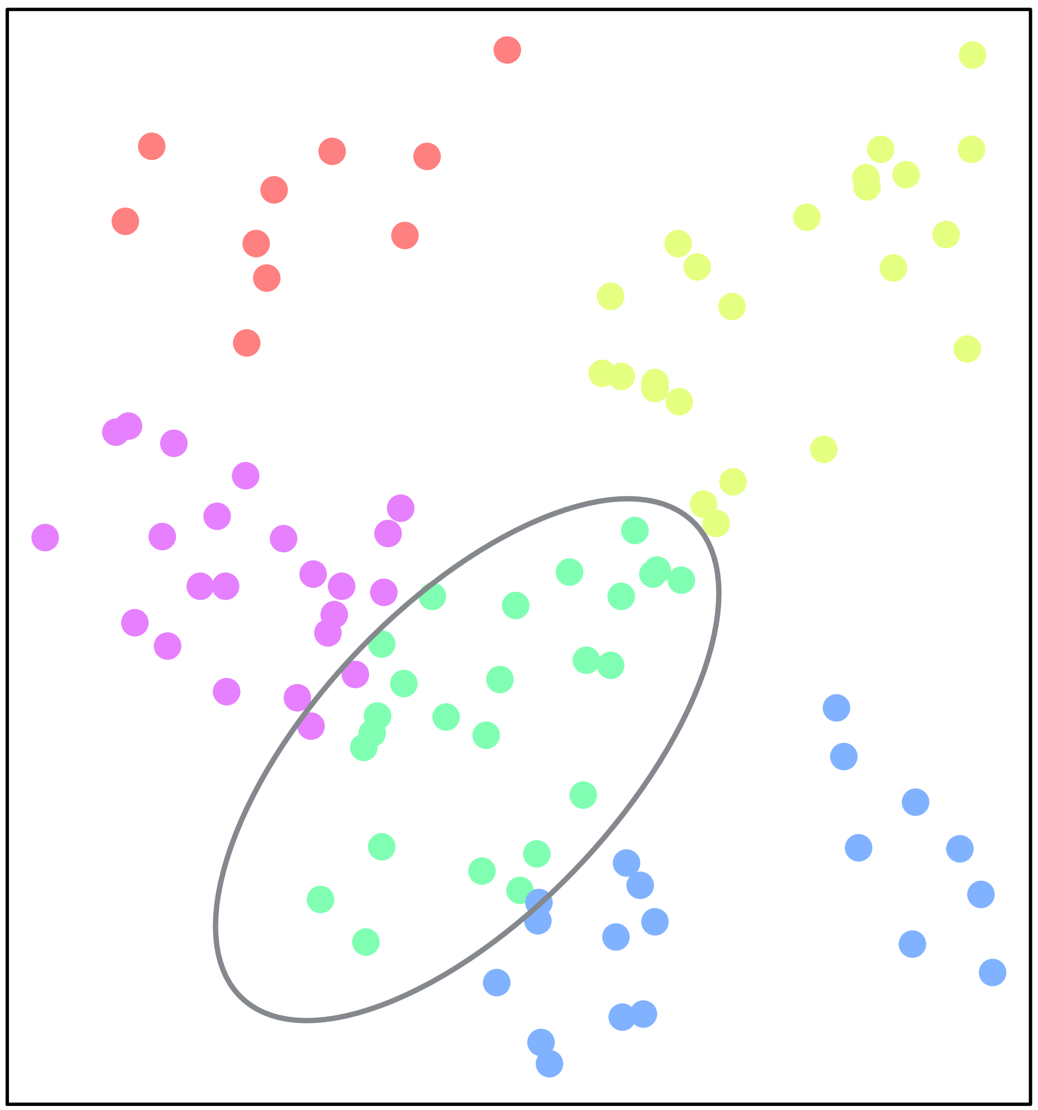
But this feels like a mistake. Labels <u>should</u> tell you not to group these items?





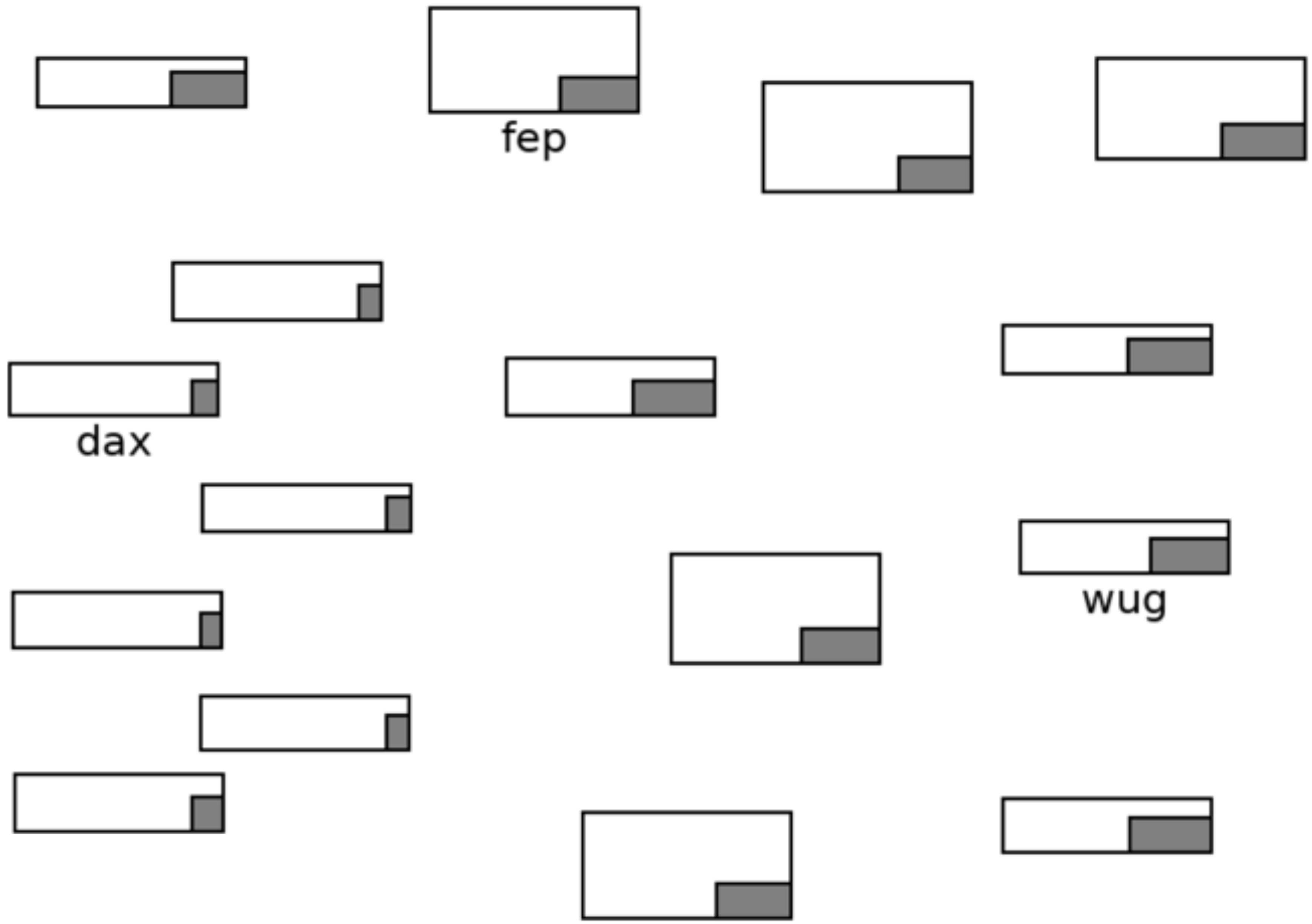
The category boundary runs right through the middle, so this should be split?





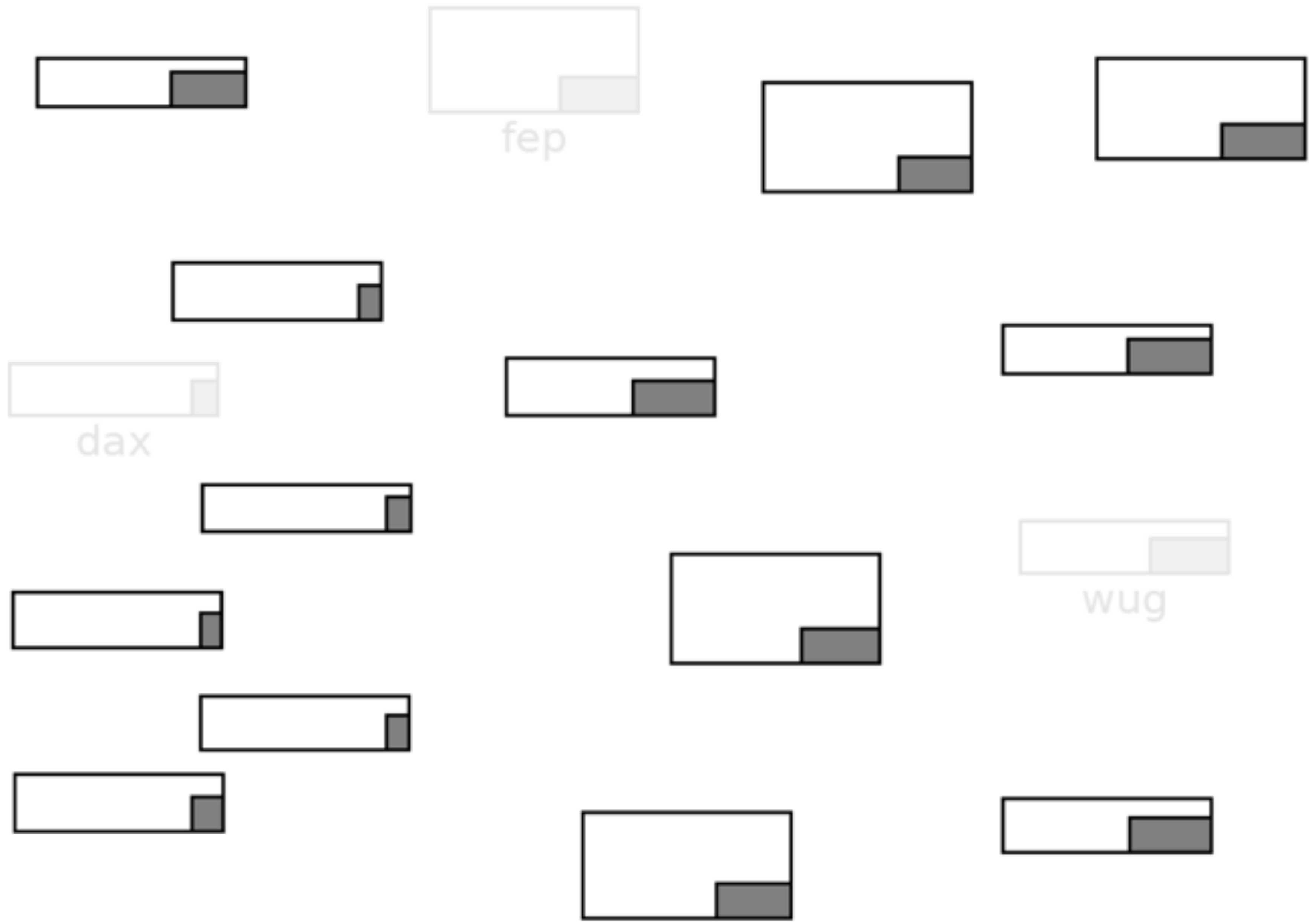
RMC tends not to make this error

Task: sort these into categories



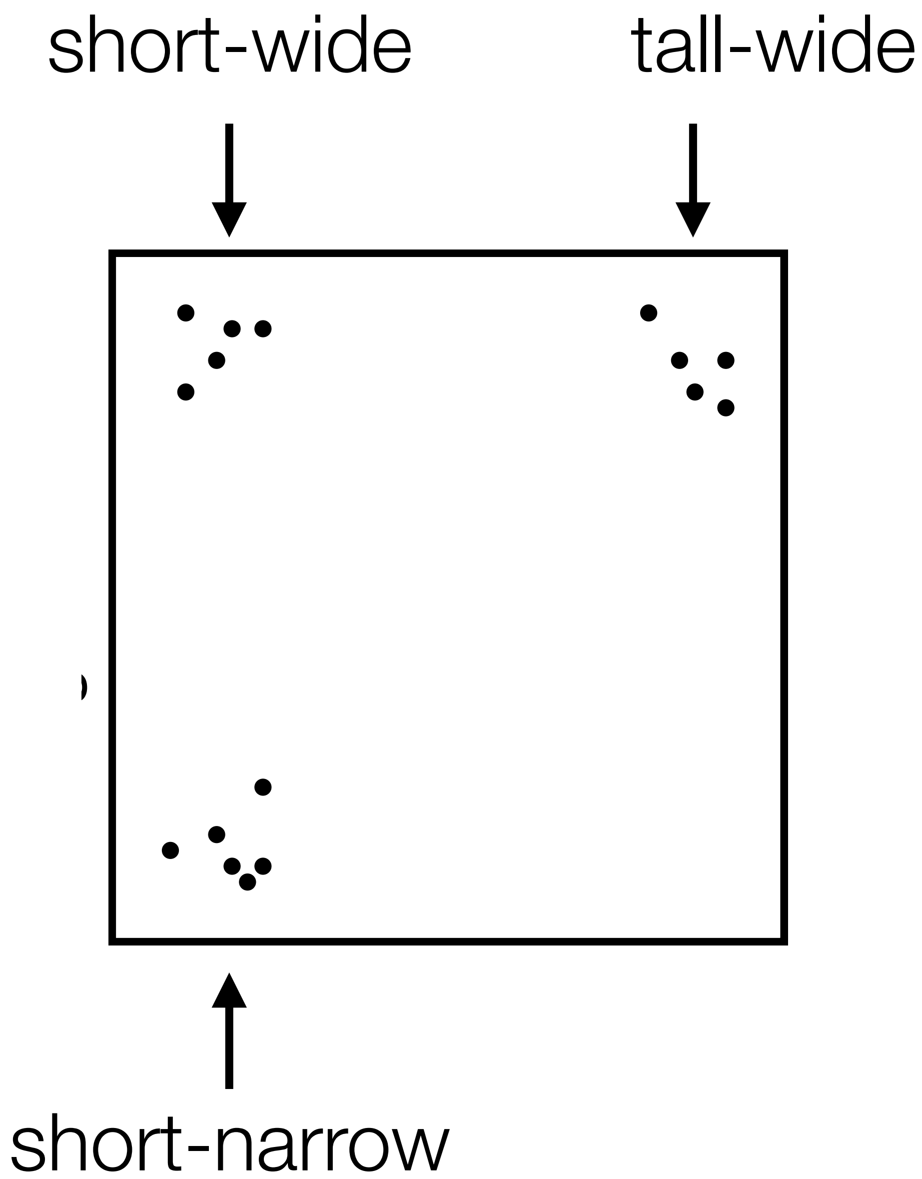
A few are labelled

Most are not

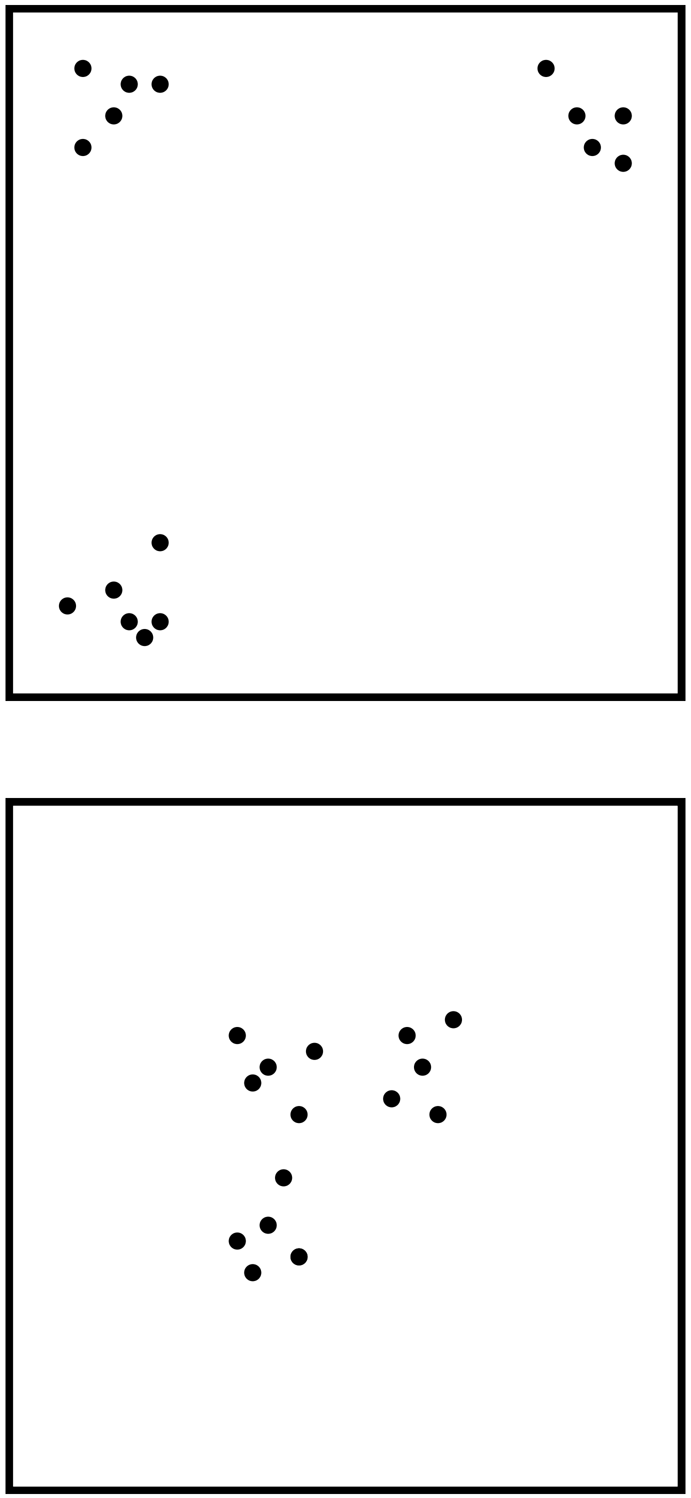


Stimuli vary in outer-height

Stimuli vary in inner-width

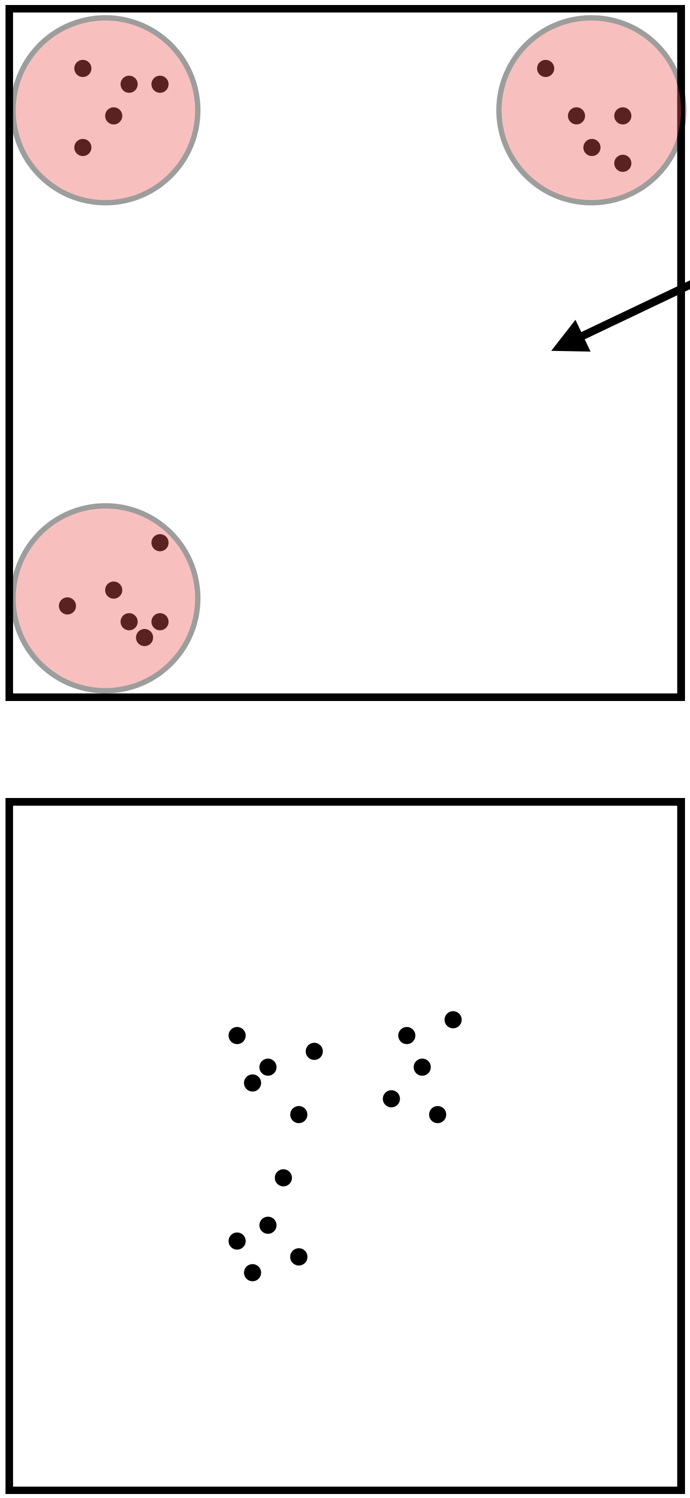


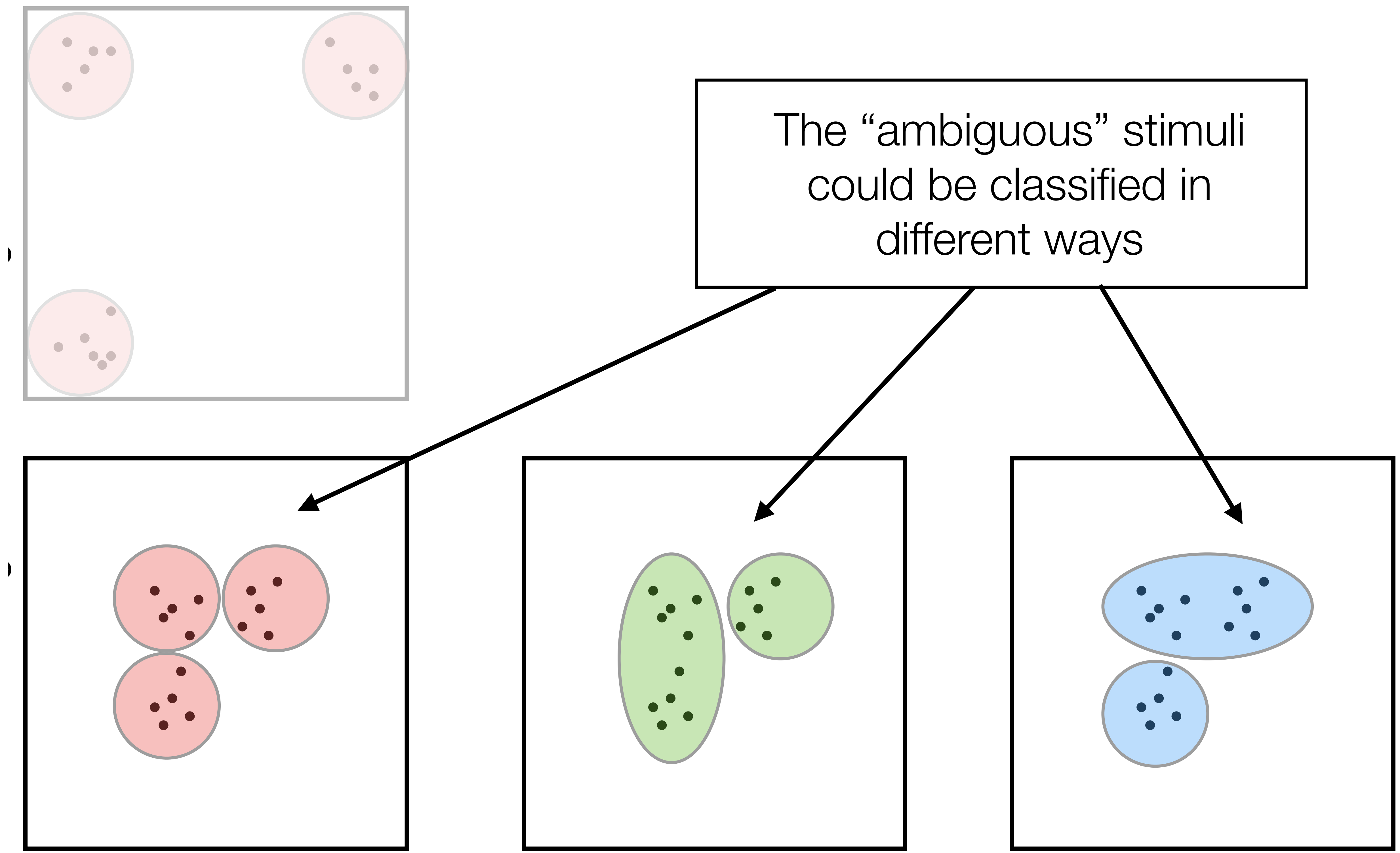
You don't really need anyone to tell you the labels to figure out what categories these should go into!

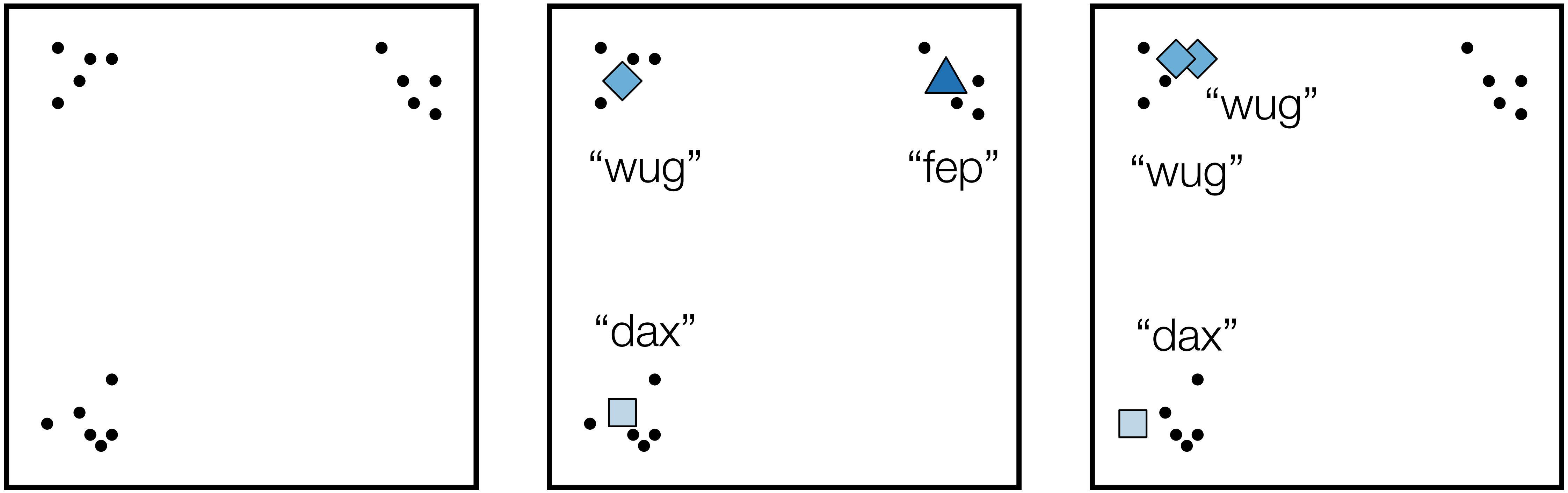


This is a lot more ambiguous

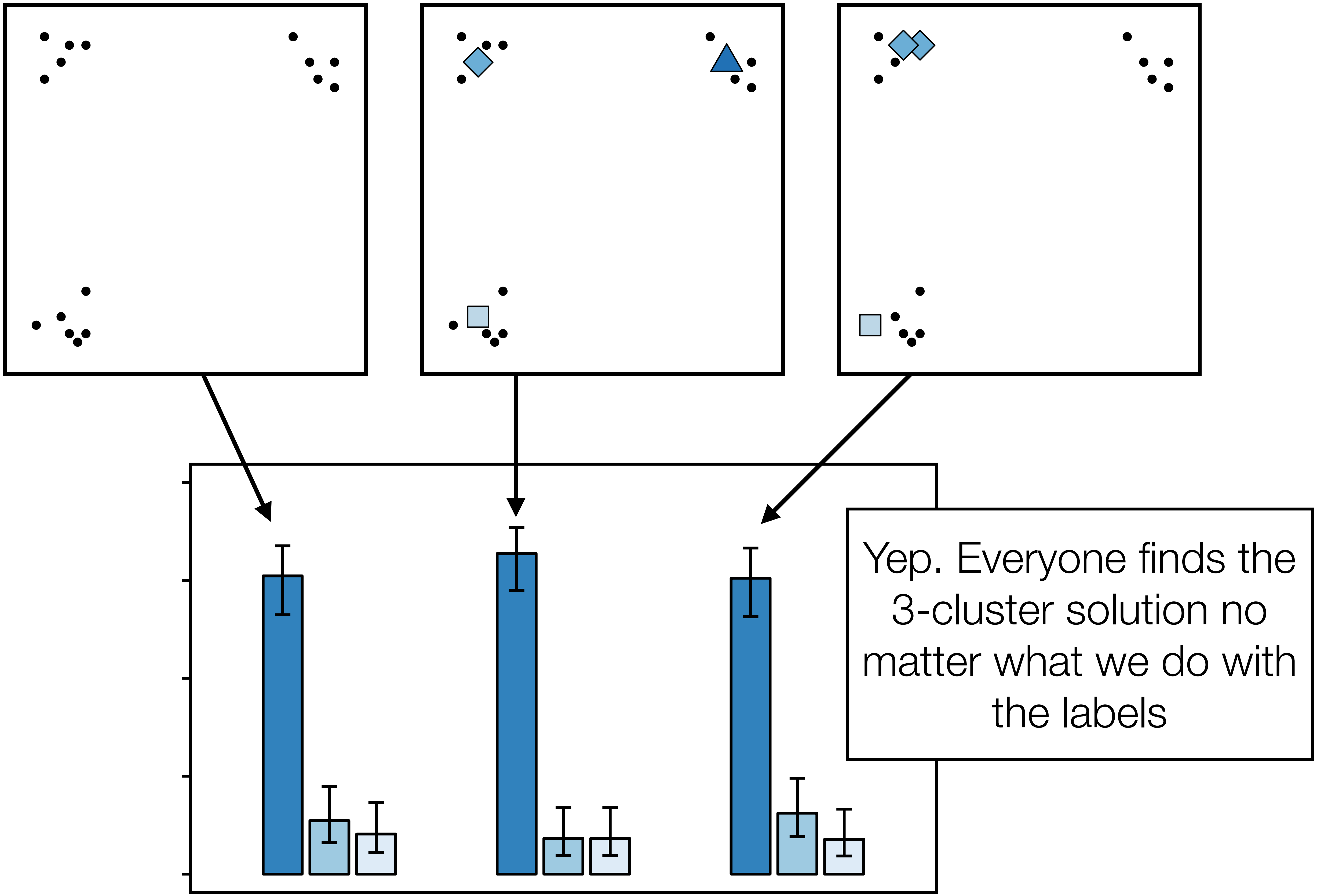
The “distinct” stimuli are
“obviously” three categories

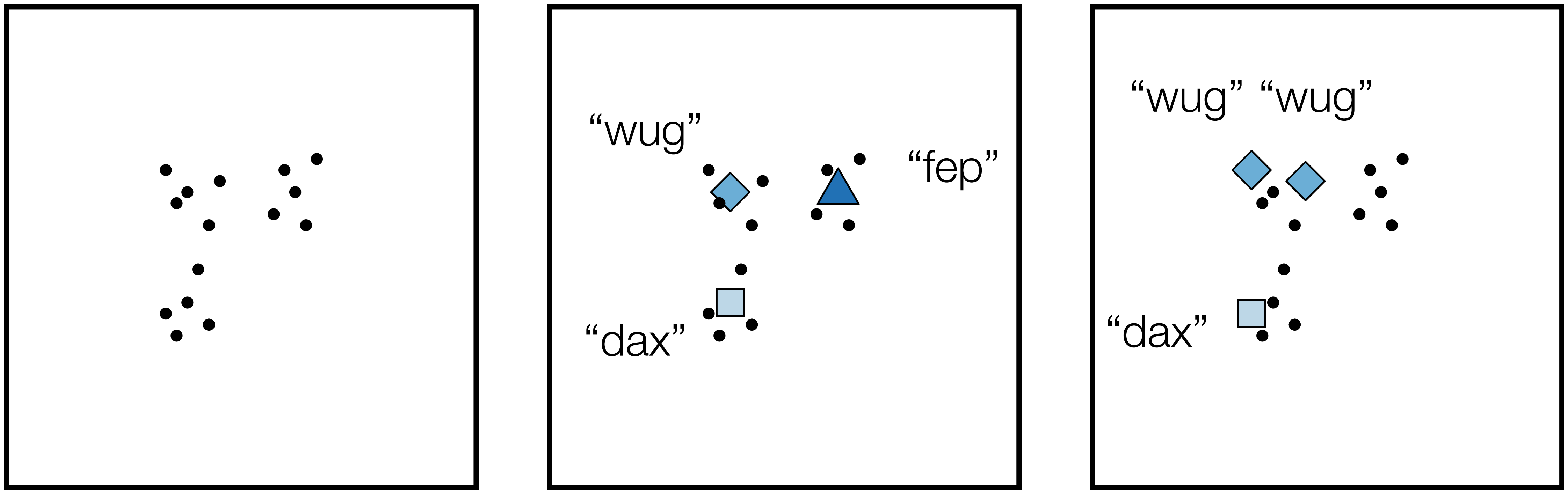



The "ambiguous" stimuli could be classified in different ways

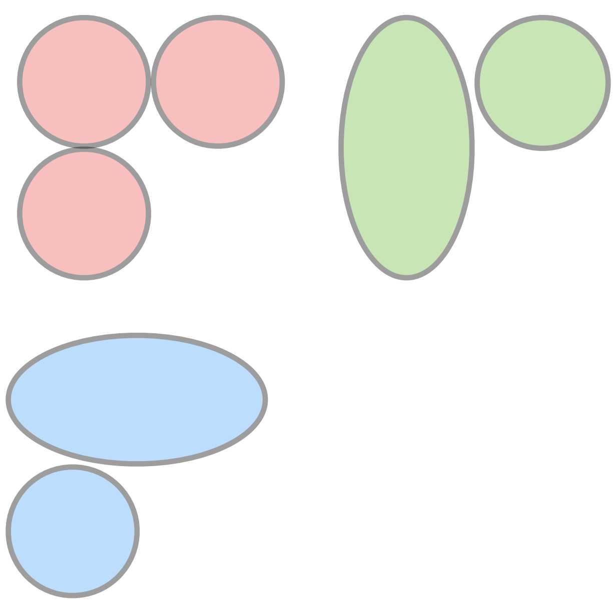


It doesn’t matter which labels you reveal, any intelligent learner is going to figure out what’s going on, right?

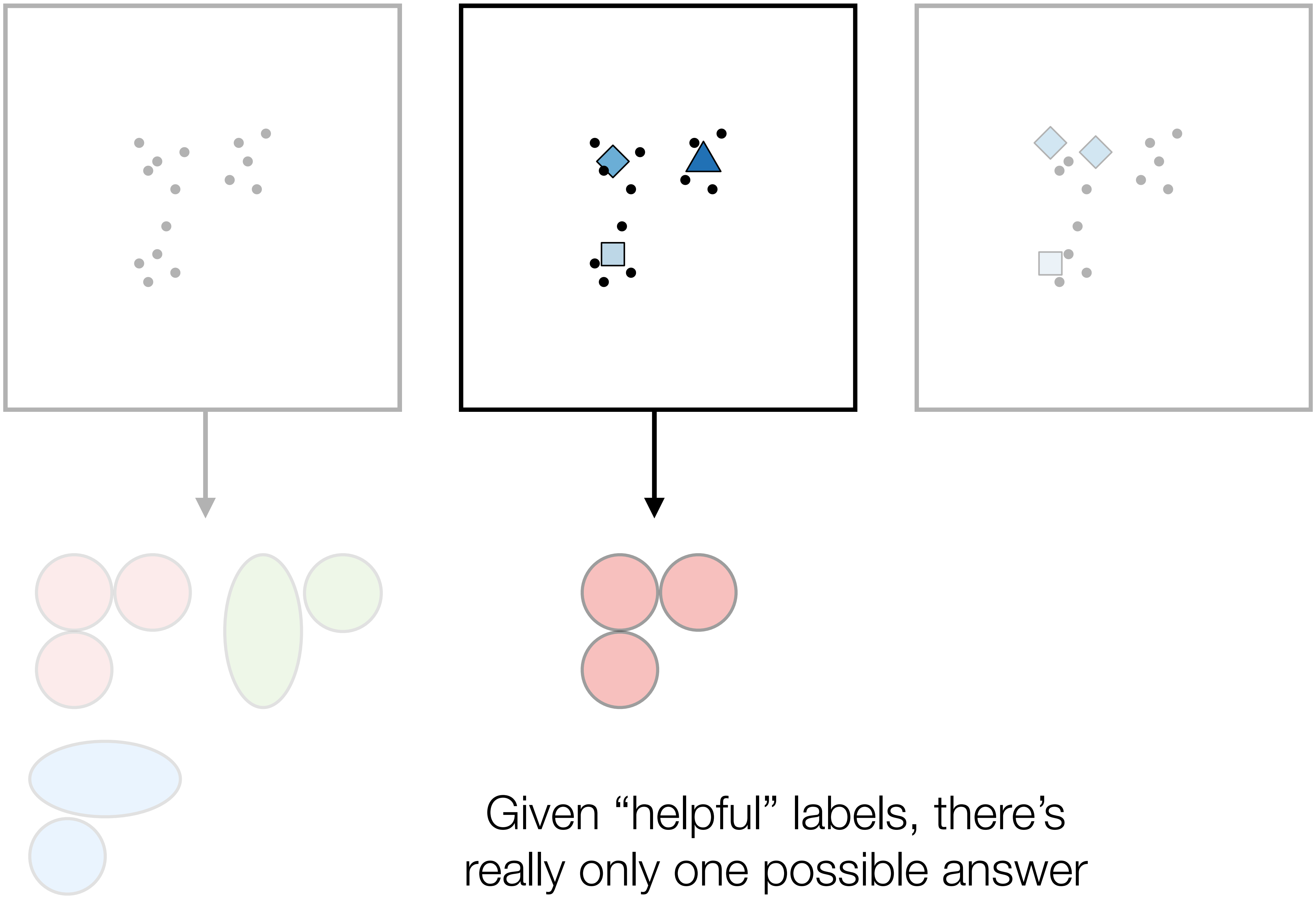

Yep. Everyone finds the 3-cluster solution no matter what we do with the labels



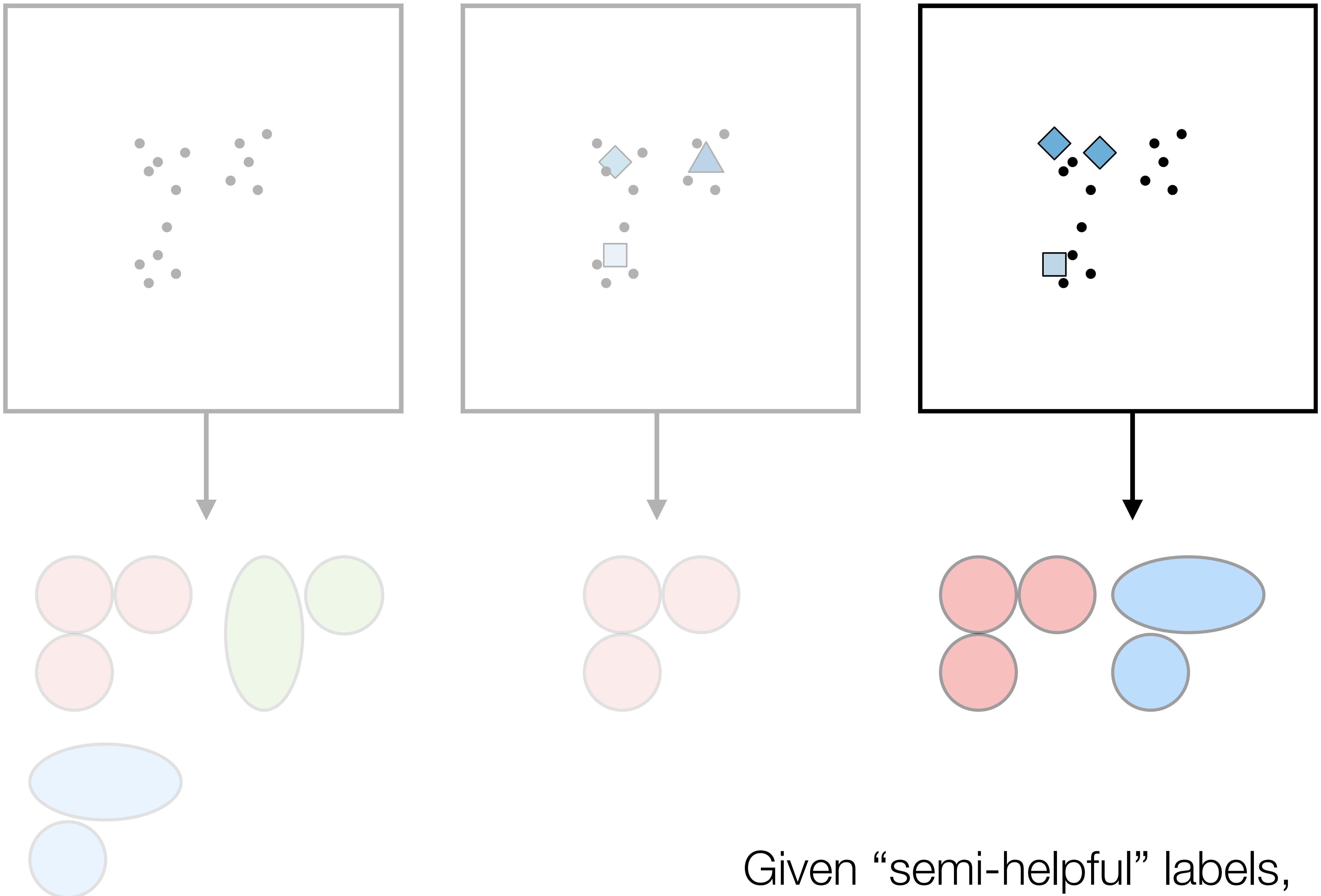
With the ambiguous stimuli, the labelling might matter?



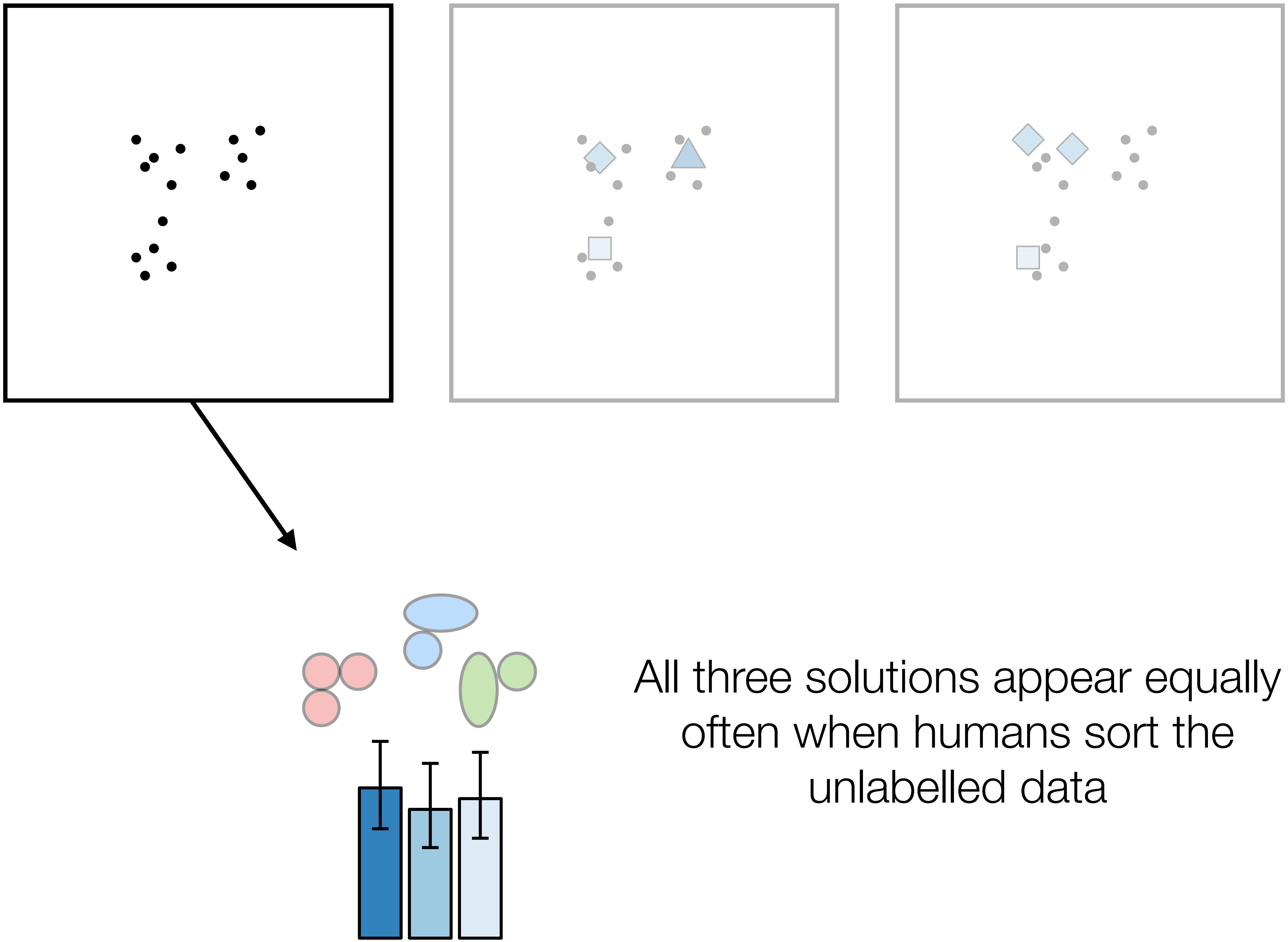
Without labels, all three of these make sense



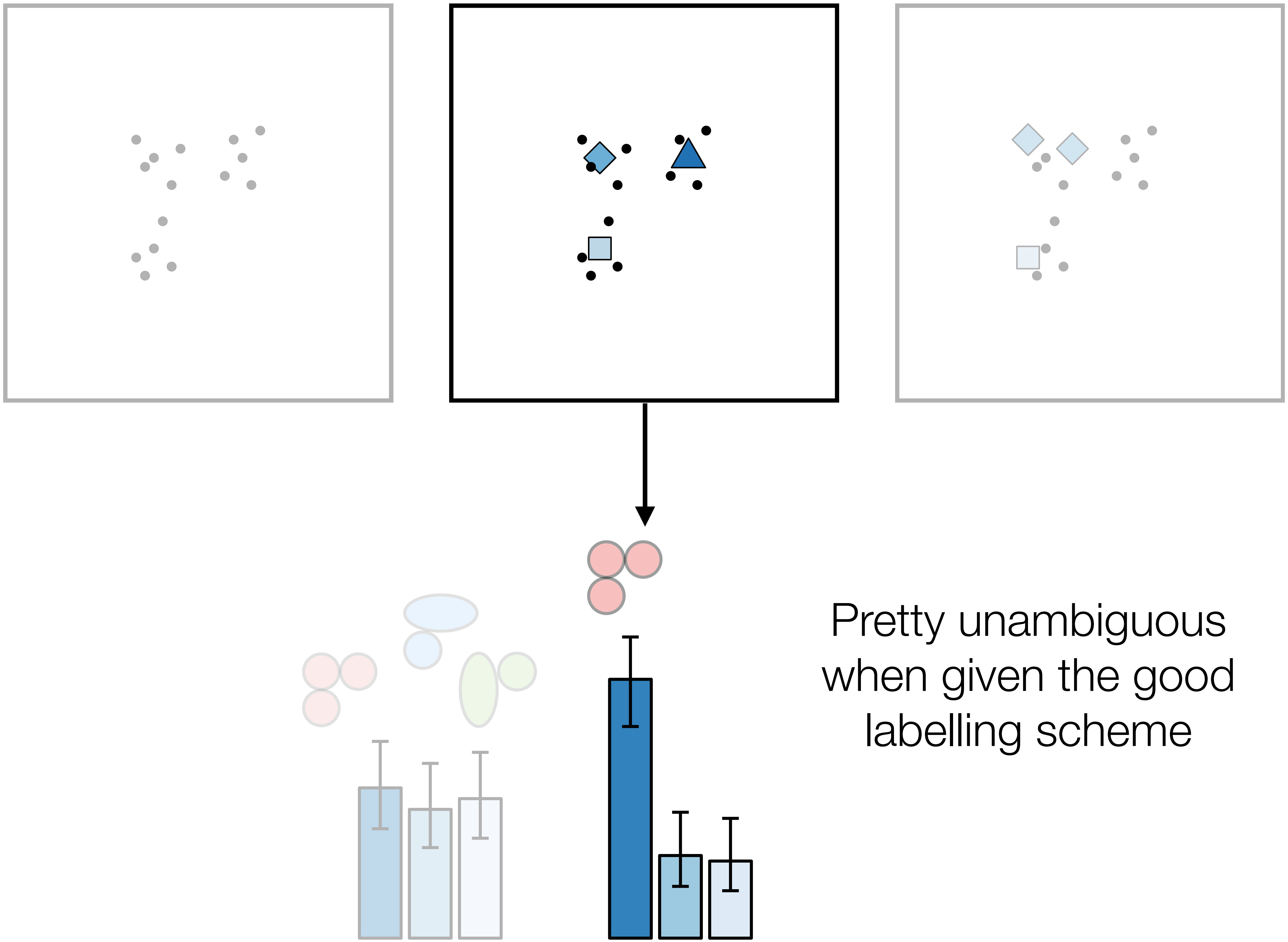
Given “helpful” labels, there’s really only one possible answer



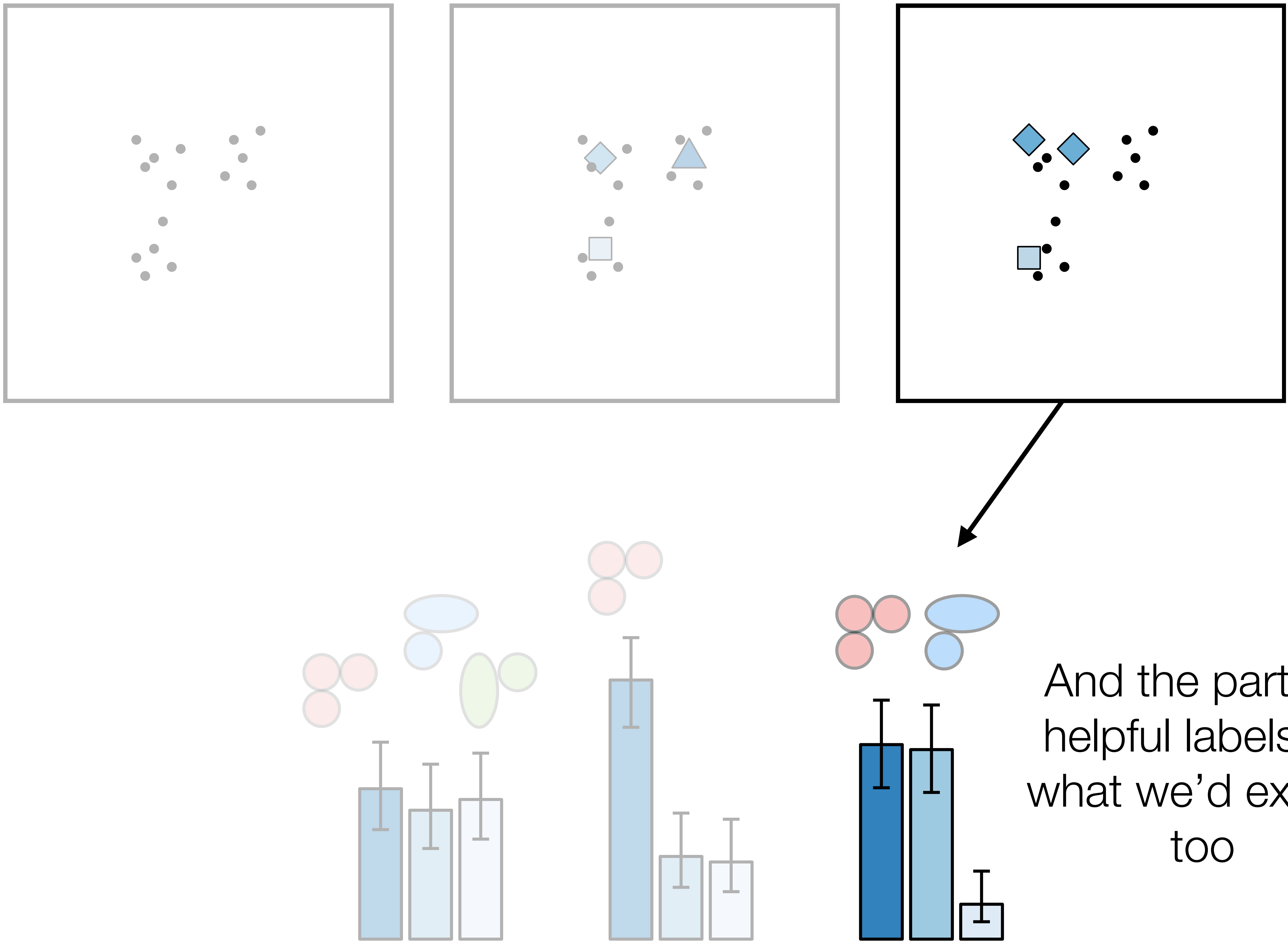
Given "semi-helpful" labels,
there's still two possibilities



All three solutions appear equally often when humans sort the unlabelled data

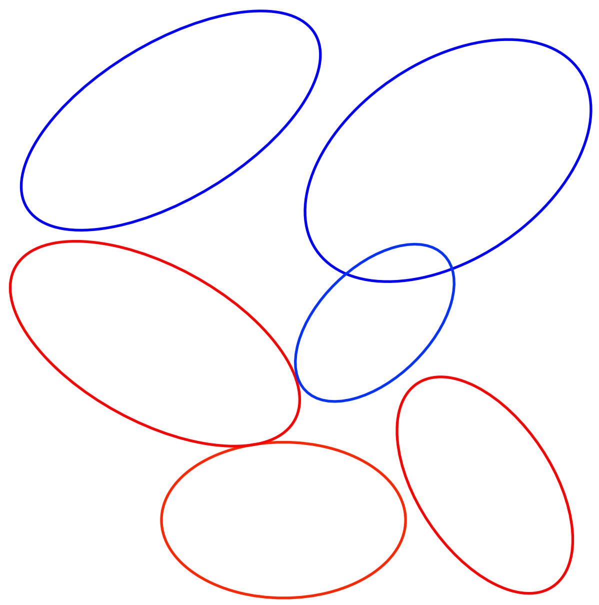


Pretty unambiguous when given the good labelling scheme



And the partially helpful labels do what we'd expect too

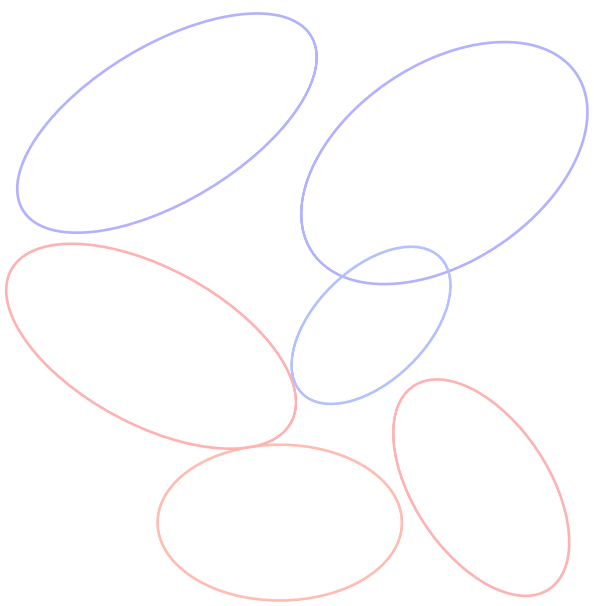
The usual RMC



Number of clusters is **unknown** so we use a **CRP prior** to learn it from the data

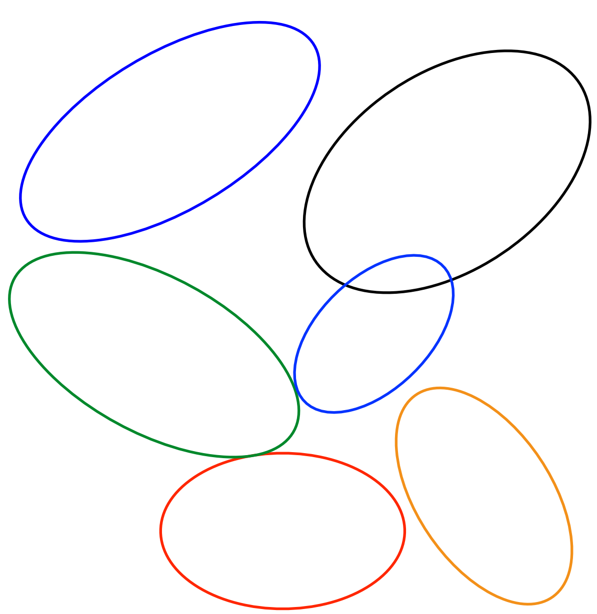
Number of possible labels (red, blue) is known so we don't need a CRP

An extended RMC



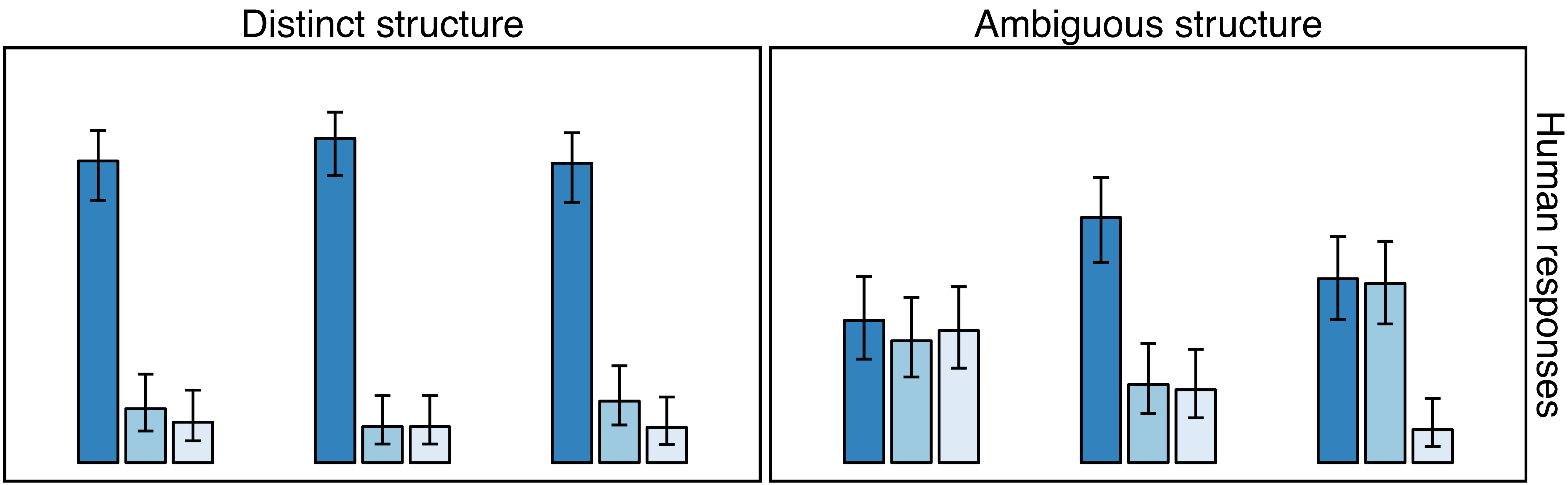
Number of clusters is **unknown** so we use a **CRP prior** to learn it from the data

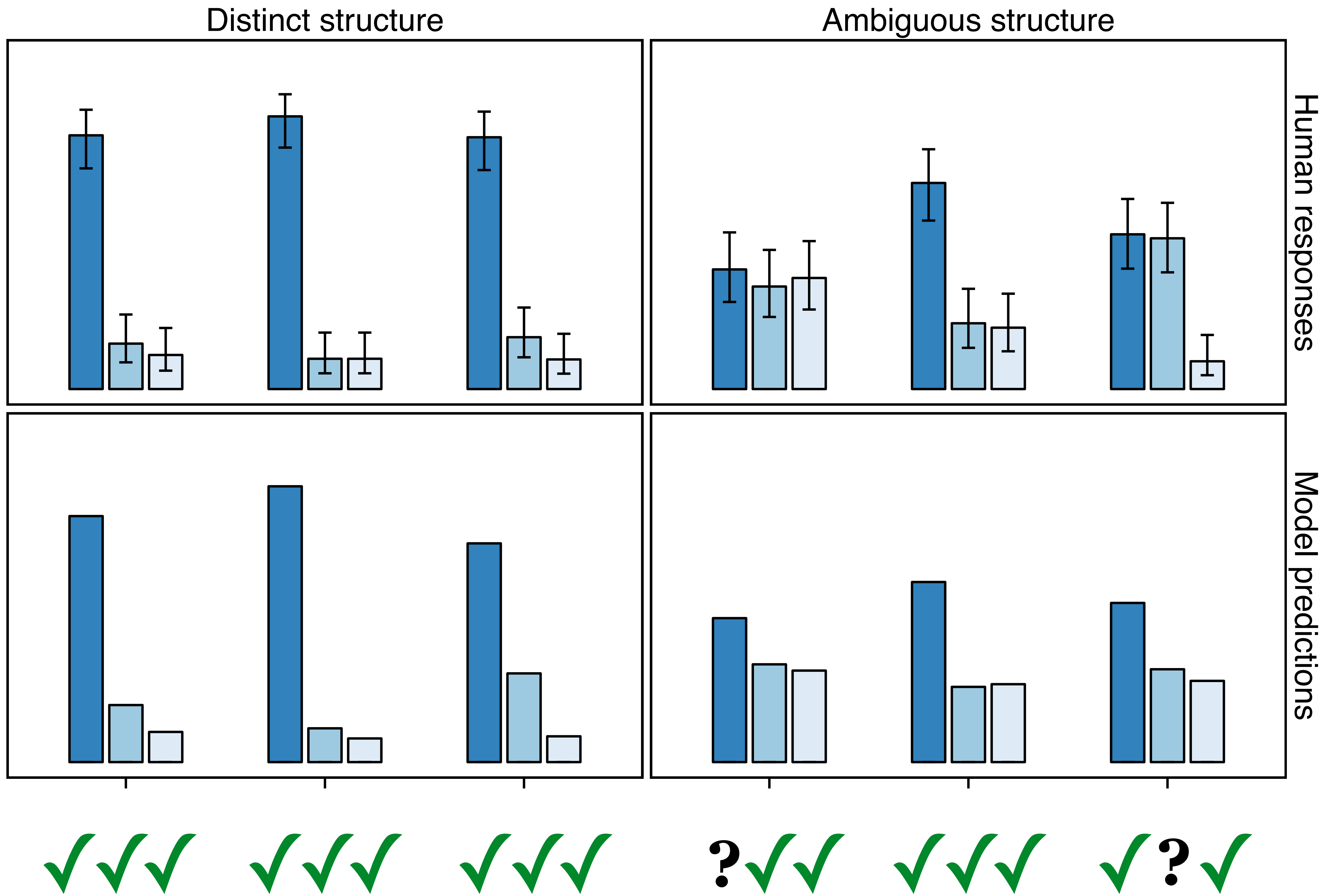
Number of possible labels (red, blue) is known so we don't need a CRP



Number of clusters is **unknown** so we use a **CRP prior** to learn it from the data

Number of possible labels (red, blue, green?) is unknown so we use a **CRP prior** to learn it from the data


Distinct structure
Ambiguous structure
Human responses


Distinct structure
Ambiguous structure
Human responses
Model predictions
?
?

A statistical answer to the question

- Why do category labels seem useful sometimes?
  - Because they’re sometimes actually helpful.
- Why are category labels seem useless sometimes?
  - Because they’re sometimes ambiguous
  - Because they’re sometimes entirely unnecessary

Summary

- Supervised learning:
  - prototypes and exemplars
  - Gaussian classifiers, k-NN, kernel methods
- Unsupervised learning:
  - k-means classfiers, mixtures of Gaussians
  - example from phonetic learning
- Semi-supervised learning
  - a simple heuristic, the CRP prior and the RMC
  - example from human concept learning

- Next… learning richer structure!