Expectation Truncation

Jörg Lücke

FIAS, Goethe-Universität Frankfurt, Germany

Frankfurt, 2011

Expectation Truncation

Jörg Lücke

FIAS, Goethe-Universität Frankfurt, Germany

Frankfurt, 2011

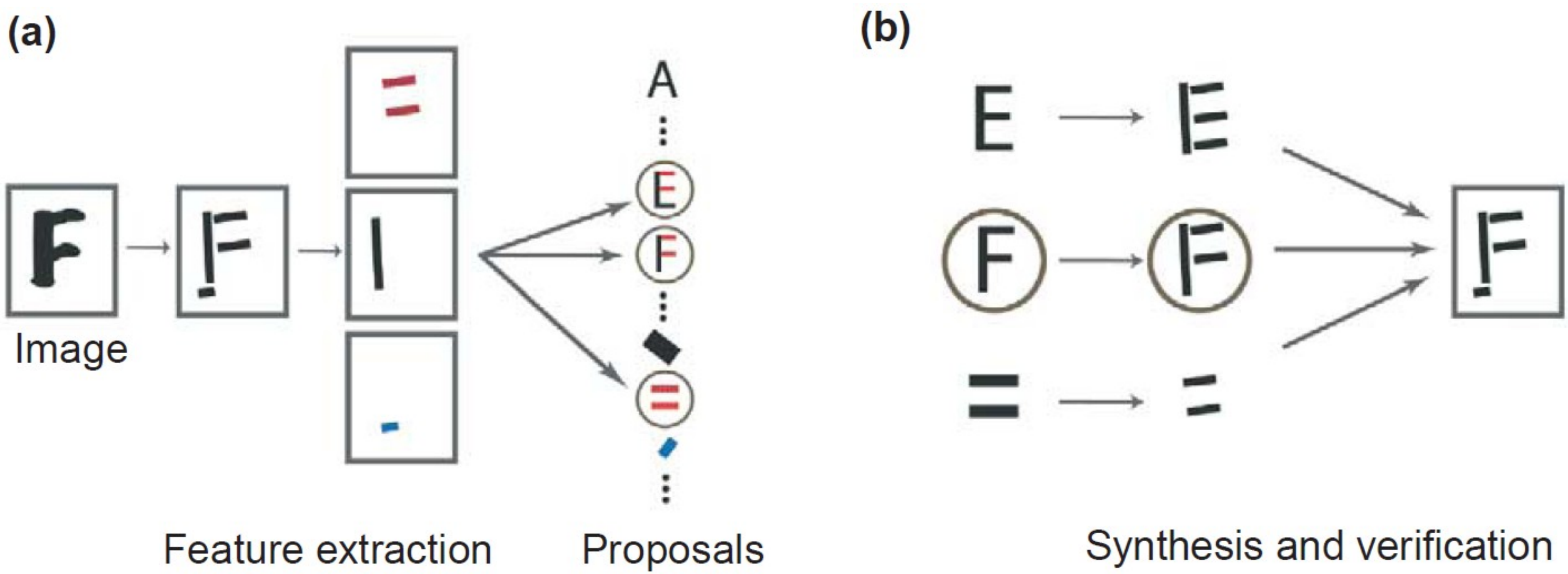
This talk is about the paper:
"Expectation Truncation and the Benefits of Preselection", Lücke & Eggert, JMLR 2010.

Text that explains the slides in the absence of a speaker is provided in grey.
Additional material such as animations are available on fias.uni-frankfurt.de/cnml → Selected Publications

Motivation

Example Motivation:
"We propose an 'analysis by synthesis' strategy where lowlevel cues, combined with spatial grouping rules (similar to Gestalt laws), make bottom-up proposals which activate hypotheses about objects and scene structures. "

Text and Fig. From: A. Yuille & D. Kersten, *TICS* 2006
Vision as Bayesian inference: analysis by synthesis?



Preselection

Recurrent Recognition

… this strategy is kind of well established.

Motivation

Example Motivation:
"We propose an 'analysis by synthesis' strategy where lowlevel cues, combined with spatial grouping rules (similar to Gestalt laws), make bottom-up proposals which activate hypotheses about objects and scene structures. "

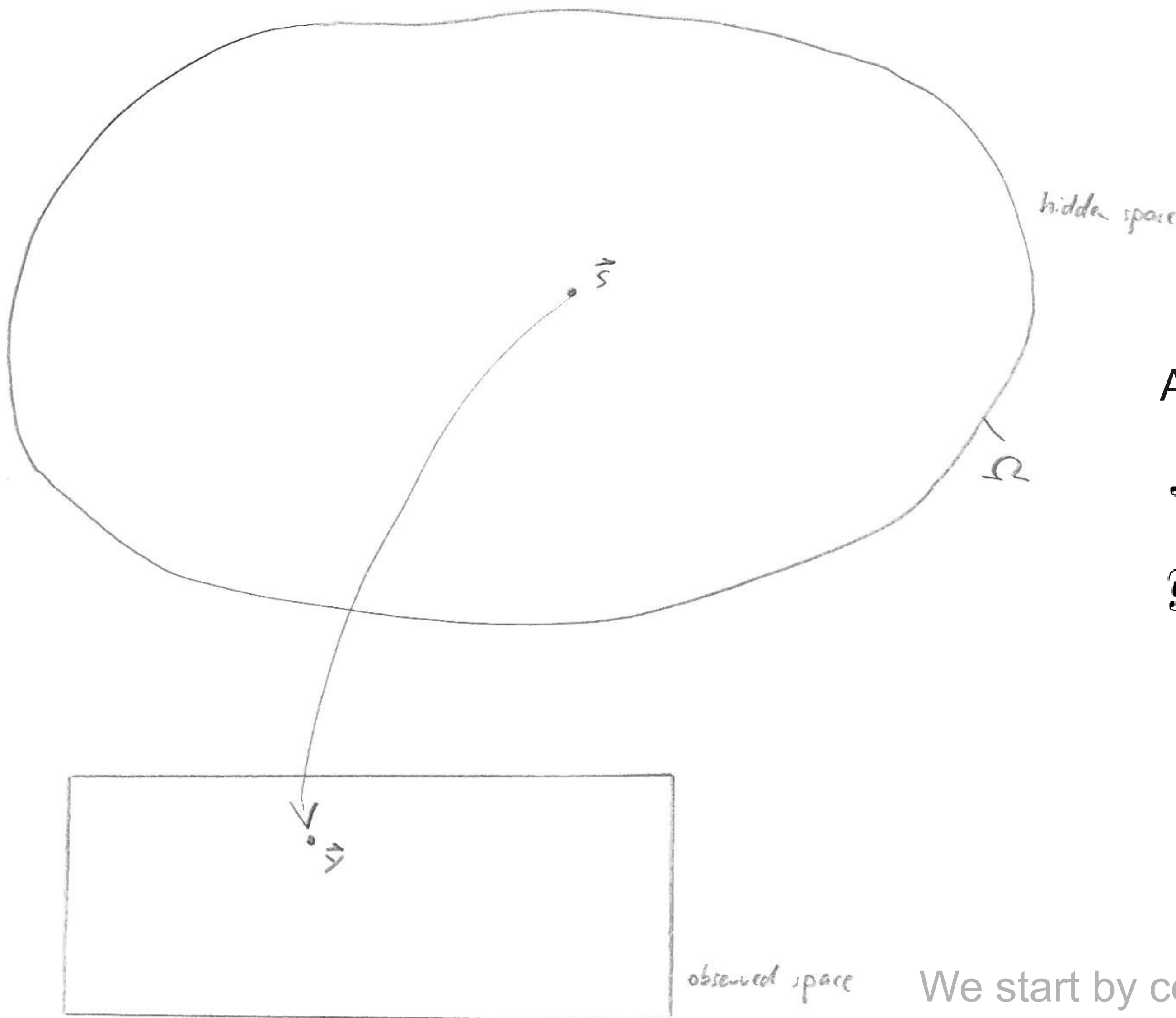
Text and Fig. From: A. Yuille & D. Kersten, *TICS* 2006
Vision as Bayesian inference: analysis by synthesis?

Further examples:
"[The anatomy of the cortex provides] a large-scale computational hypothesis on visual recognition, which includes both, rapid parallel forward recognition, independent of any feedback prediction, and a feedback controlled refinement system."
Körner et al., *Neural Networks* 1999
A model of computation in neocortical architecture

or
Lee & Mumford, J Opt Soc Am A, 2003
Wolfrum et al., Journal of Vision 2008
Westphal und Würtz, Neural Comp 2009
and many more ...

Preselection + Recurrent Recognition → faster inference

… sounds like an approximate inference scheme.

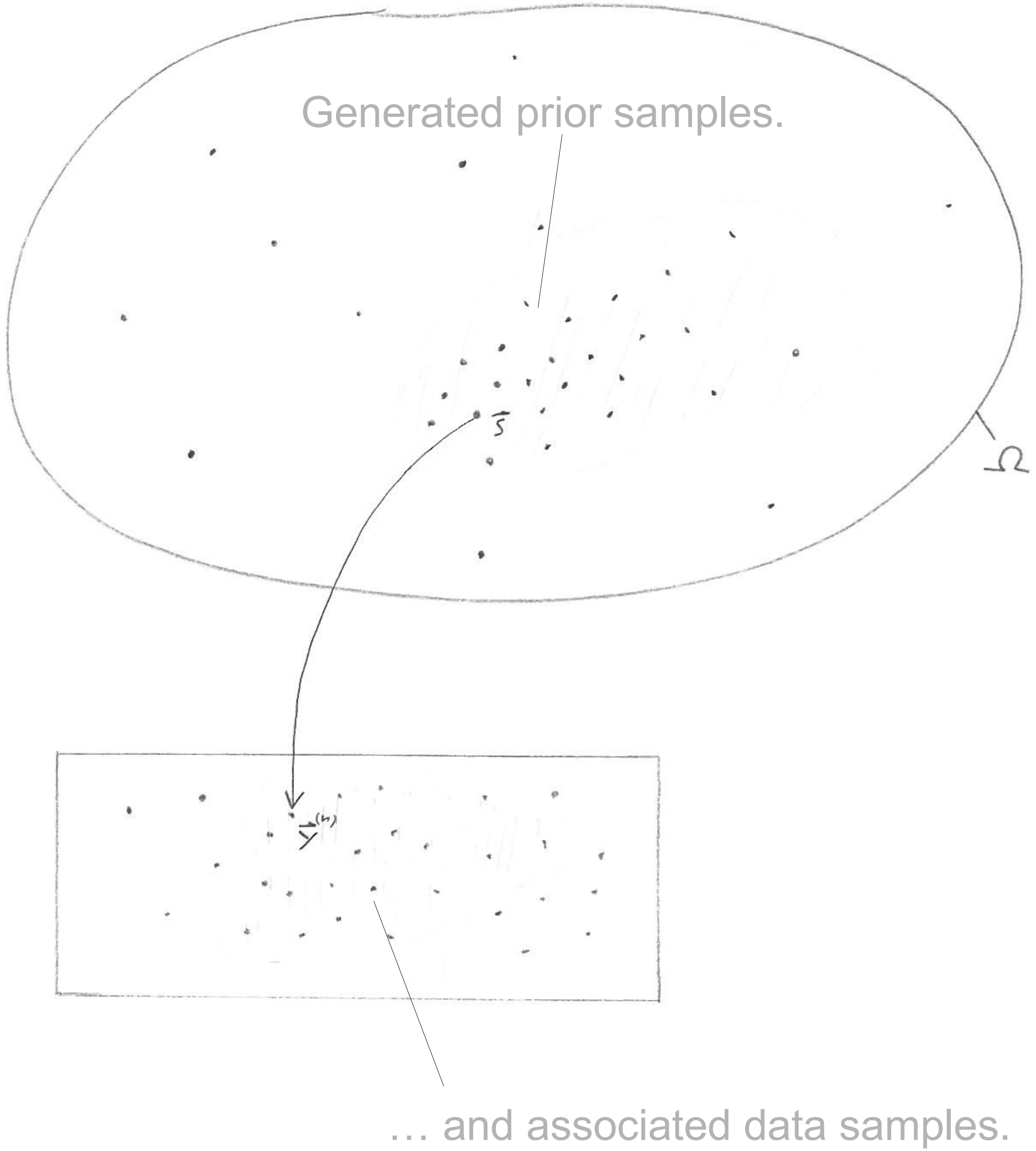


A generative model

$$\vec{s} \sim p(\vec{s}\,|\,\Theta)$$

$$\vec{y} \sim p(\vec{y}\,|\,\vec{s},\Theta)$$

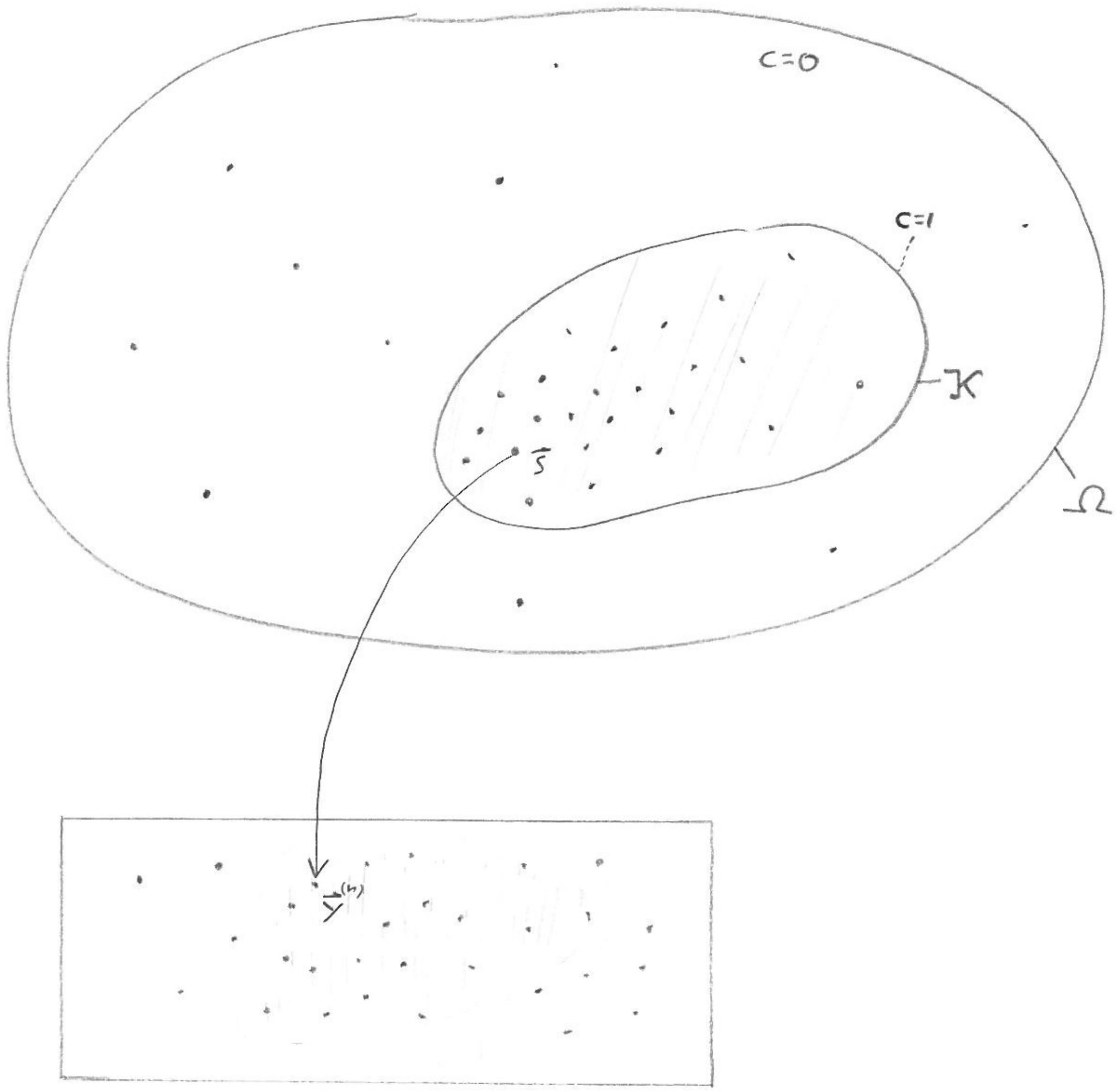
We start by considering a generative model with hidden variables **s** and observed variables **y**.
Our general strategy will be to restrict the hidden space for learning with only small losses for the accuracy. With this strategy we will come back to preselection only later.



A generative model

$$\vec{s} \sim p(\vec{s} \,|\, \Theta)$$

$$\vec{y} \sim p(\vec{y} \,|\, \vec{s}, \Theta)$$



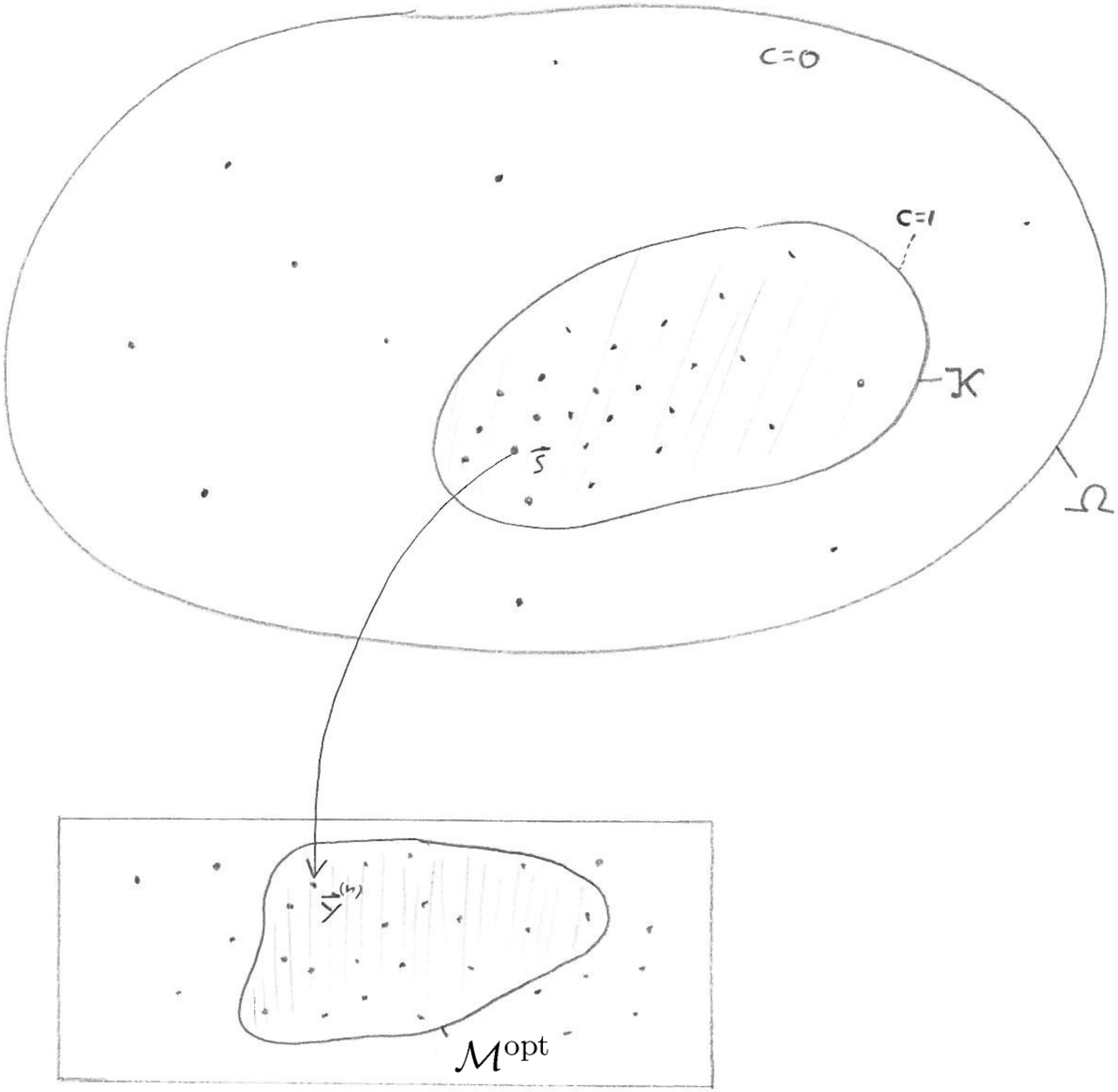
Idea 1: Define a set $\mathcal{K}$ that contains **most prior mass**.

This defines a truncated generative model:

$$\vec{s} \sim p(\vec{s} \,|\, \Theta)$$

$$\text{reject} \;\; \vec{s} \;\; \text{if} \;\; \vec{s} \notin \mathcal{K}$$

$$\vec{y} \sim p(\vec{y} \,|\, \vec{s}, \Theta)$$



Idea 1: Define a set $\mathcal{K}$ that contains **most prior mass**.

This defines a truncated generative model:

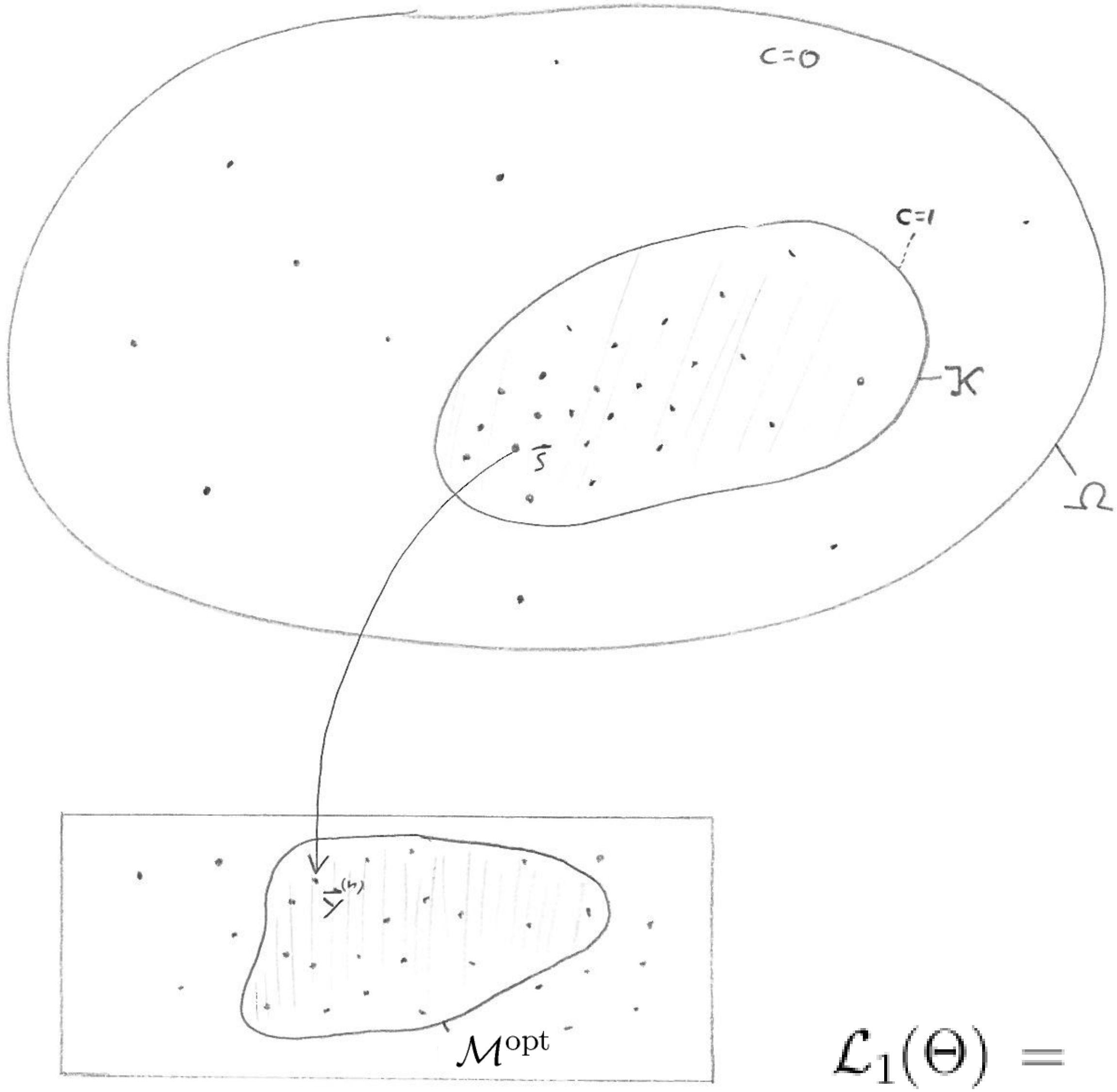
$$\vec{s} \sim p(\vec{s}\,|\,\Theta)$$

$$\text{reject } \ \vec{s} \ \text{ if } \ \vec{s} \notin \mathcal{K}$$

$$\vec{y} \sim p(\vec{y}\,|\,\vec{s}, \Theta)$$

The truncated generative model Generates data points in $\mathcal{M}^{\mathrm{opt}}$.

The set $\mathcal{K}$ should for the moment be thought of as being large enough such that it contains most prior mass throughout learning. Flexible sizes do not pose principle problems.



Idea 1: Define a set $\mathcal{K}$ that contains **most prior mass**.

This defines a truncated generative model:

$$\vec{s} \sim p(\vec{s}\,|\,\Theta)$$

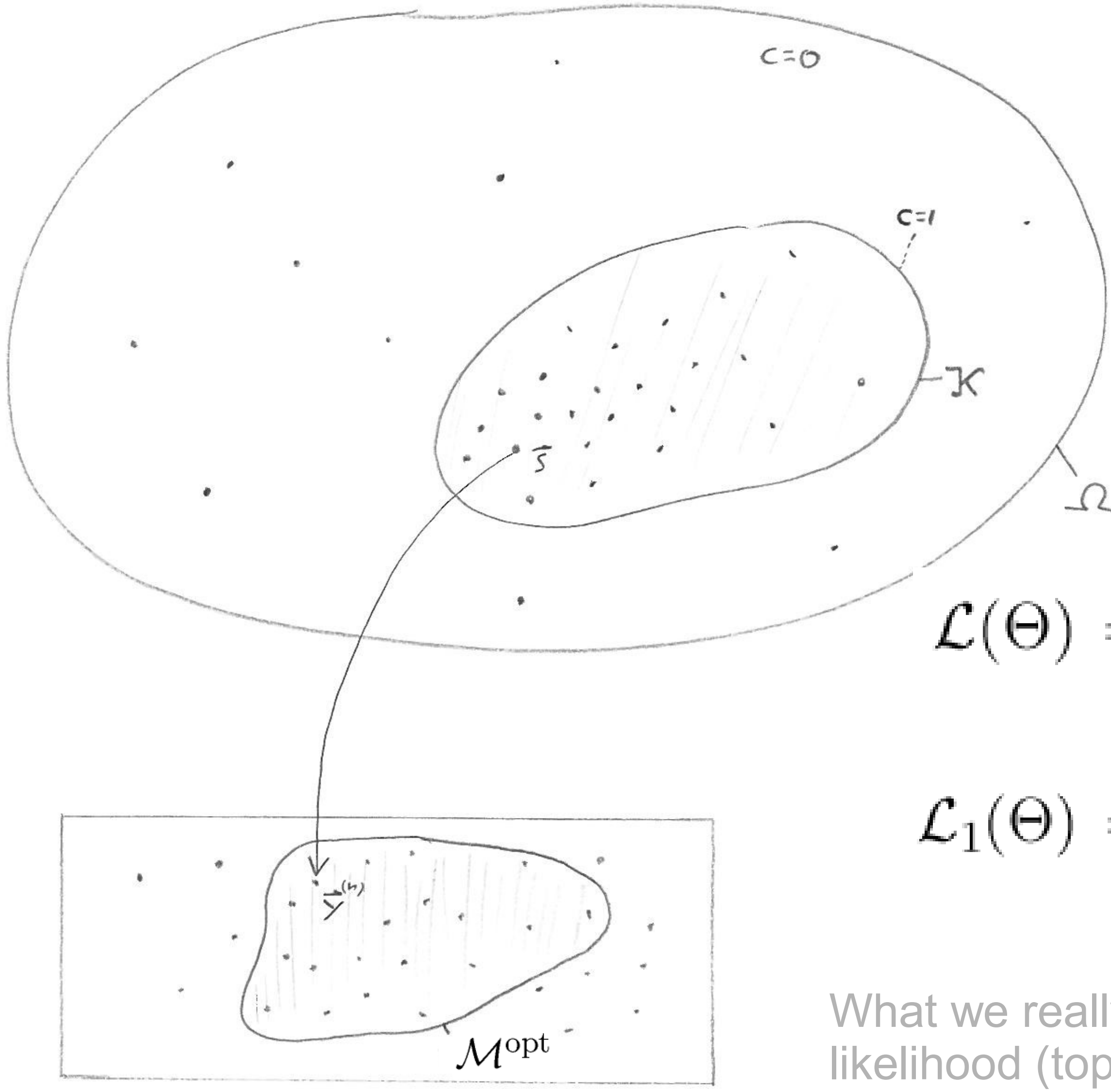
$$\text{reject} \;\; \vec{s} \;\; \text{if} \;\; \vec{s} \notin \mathcal{K}$$

$$\vec{y} \sim p(\vec{y}\,|\,\vec{s}, \Theta)$$

The truncated generative model Generates data points in $\mathcal{M}^{\mathrm{opt}}$.

$$\mathcal{L}_1(\Theta) \;=\; \sum_{n \in \mathcal{M}^{\mathrm{opt}}} \log(p(\vec{y}^{(n)}\,|\,c = 1, \Theta))$$

This defines the likelihood of the truncated model.
It is computed w.r.t. the corresponding data points.



$$\vec{s} \sim p(\vec{s}\,|\,\Theta)$$

$$\text{reject}\ \ \vec{s}\ \ \text{if}\ \ \vec{s} \notin \mathcal{K}$$

$$\vec{y} \sim p(\vec{y}\,|\,\vec{s},\Theta)$$

$$\mathcal{L}(\Theta) = \sum_{n=1,\ldots,N} \log(p(\vec{y}^{(n)}\,|\,\Theta))$$

$$\mathcal{L}_1(\Theta) = \sum_{n\in\mathcal{M}^{\mathrm{opt}}} \log(p(\vec{y}^{(n)}\,|\,c=1,\Theta))$$

The truncated generative model

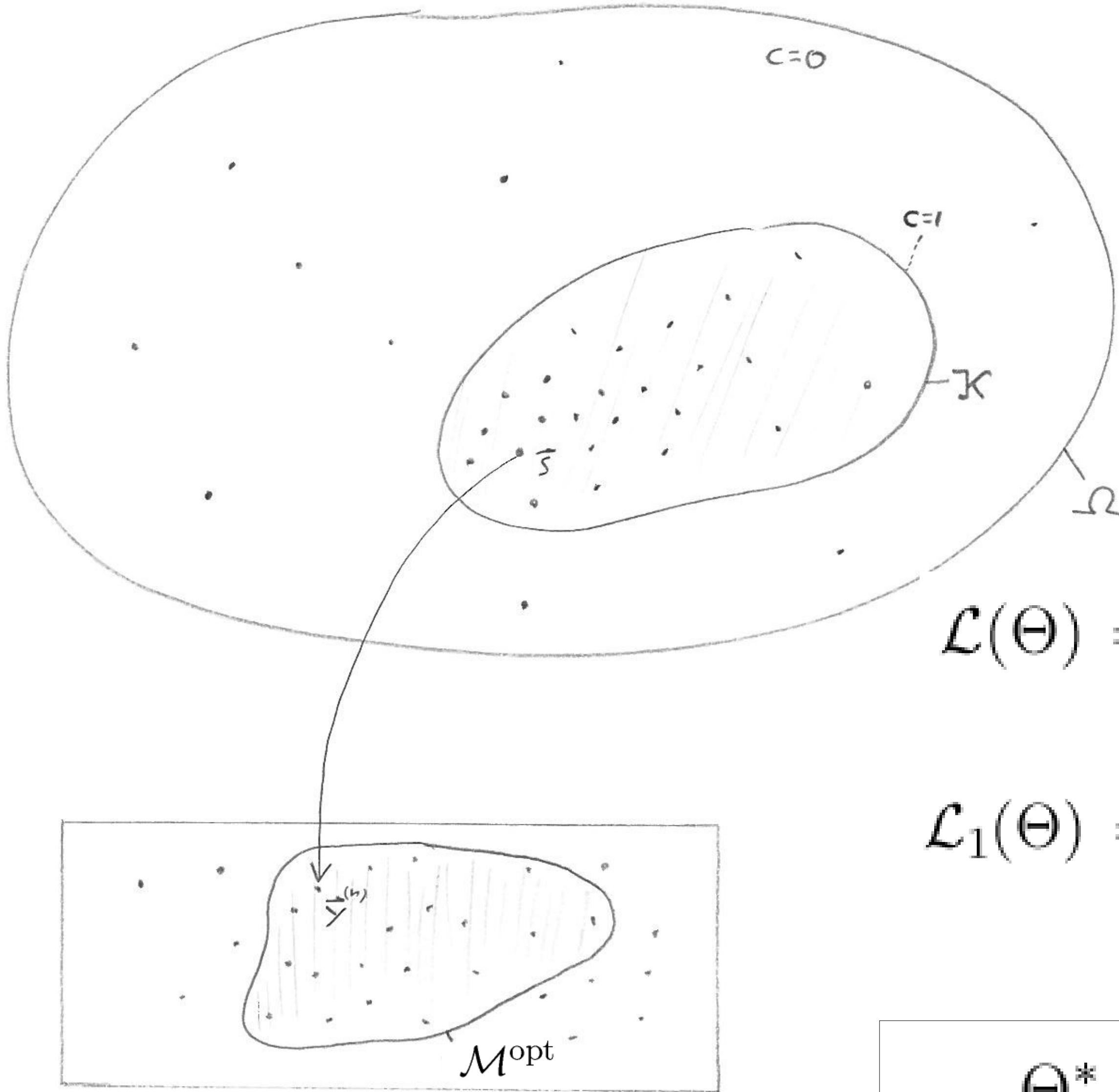
Generates data points in $\mathcal{M}^{\mathrm{opt}}$.

What we really want to optimize is the original likelihood (top). To optimize it approximately, we can make use of an interesting relation that exists between the truncated likelihood (bottom) and the original likelihood one (at least if data and model match). It is given by ...

$$\vec{s} \sim p(\vec{s}\,|\,\Theta)$$

$$\text{reject} \;\; \vec{s} \;\; \text{if} \;\; \vec{s} \notin \mathcal{K}$$

$$\vec{y} \sim p(\vec{y}\,|\,\vec{s},\Theta)$$



$$\mathcal{L}(\Theta) \;=\; \sum_{n=1,\ldots,N} \log\big(p(\vec{y}^{(n)}\,|\,\Theta)\big)$$

$$\mathcal{L}_1(\Theta) \;=\; \sum_{n\in\mathcal{M}^{\mathrm{opt}}} \log\big(p(\vec{y}^{(n)}\,|\,c=1,\Theta)\big)$$

The truncated generative model

Generates data points in $\mathcal{M}^{\mathrm{opt}}$.

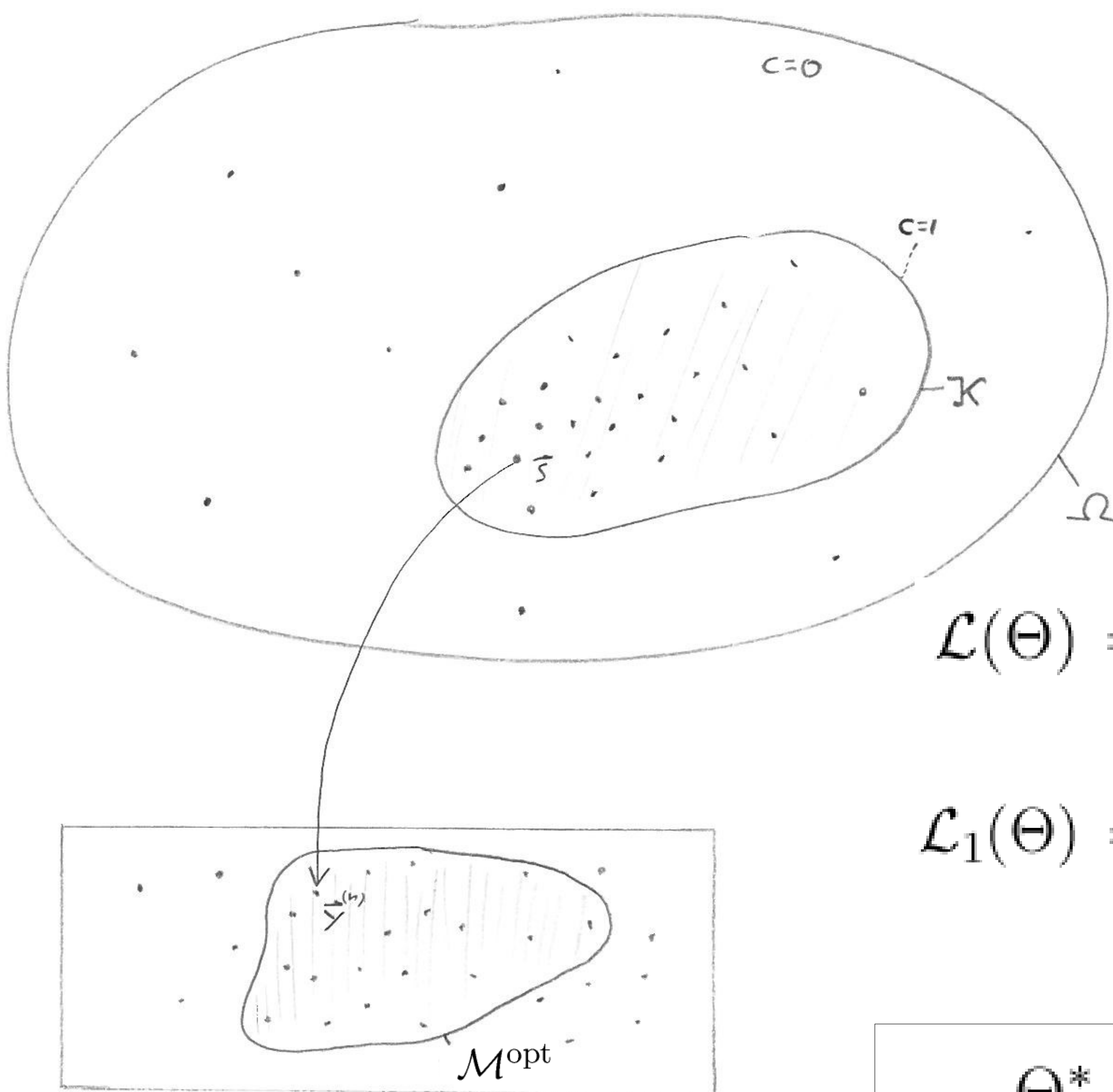
$$\Theta^* \;=\; \mathrm{argmax}_{\Theta}\{\mathcal{L}(\Theta)\}$$

$$\Rightarrow \;\; \Theta^* \;\approx\; \mathrm{argmax}_{\Theta}\{\mathcal{L}_1(\Theta)\}$$

For the usual generative models.

Problem: We do not know $\mathcal{M}^{\rm opt}$.

But: We can efficiently estimate it.



$$\mathcal{L}(\Theta) = \sum_{n=1,\ldots,N} \log(p(\vec{y}^{(n)}\,|\,\Theta)$$

$$\mathcal{L}_1(\Theta) = \sum_{n\in\mathcal{M}^{\rm opt}} \log(p(\vec{y}^{(n)}\,|\,c=1,\Theta)$$

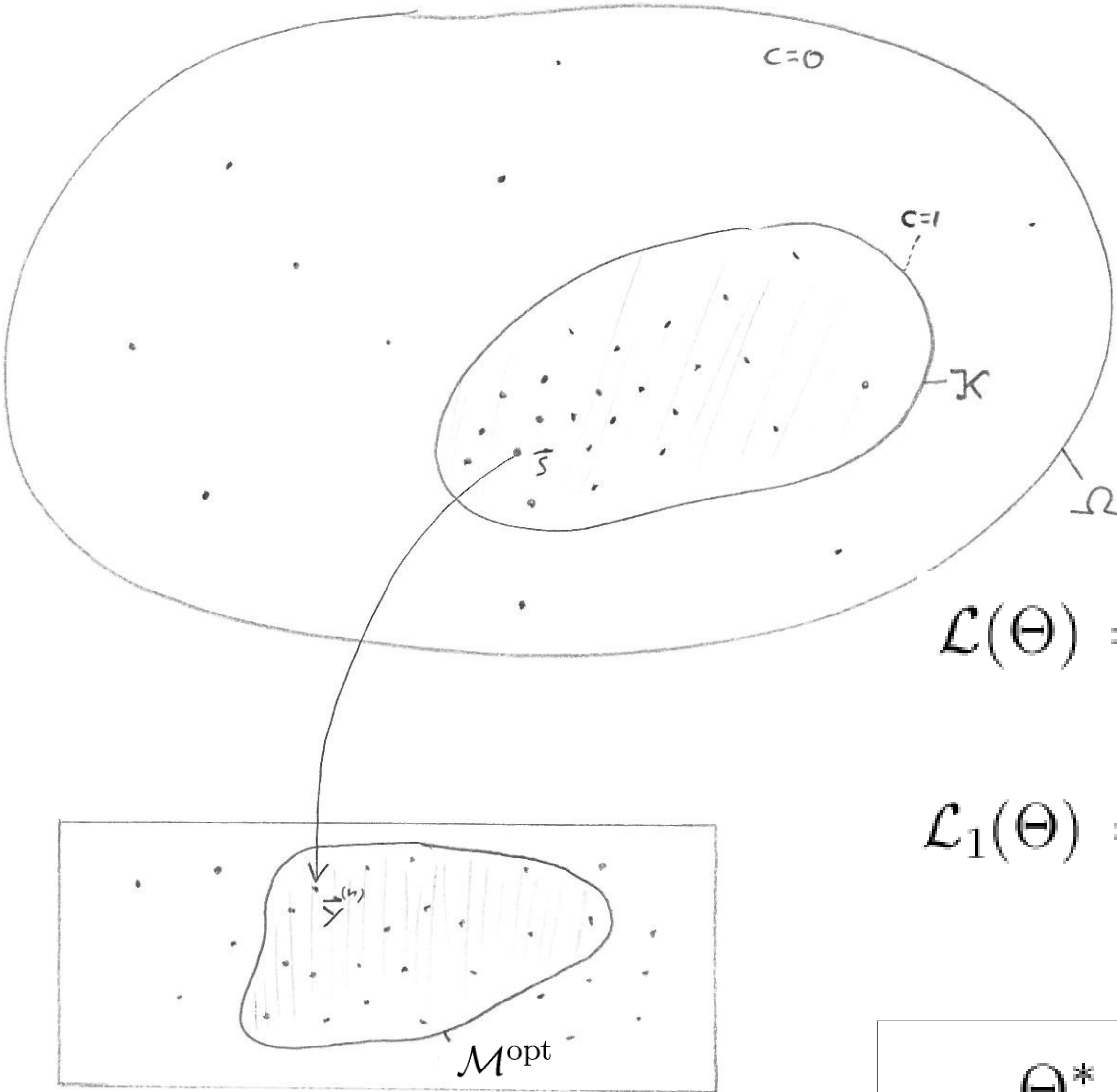
The truncated generative model

Generates data points in $\mathcal{M}^{\rm opt}$.

$$\Theta^* = \mathrm{argmax}_\Theta\{\mathcal{L}(\Theta)\}$$

$$\Rightarrow \Theta^* \approx \mathrm{argmax}_\Theta\{\mathcal{L}_1(\Theta)\}$$

For the usual generative models.



$$\vec{s} \sim p(\vec{s} \,|\, \Theta)$$

$$\text{reject} \quad \vec{s} \quad \text{if} \quad \vec{s} \notin \mathcal{K}$$

$$\vec{y} \sim p(\vec{y} \,|\, \vec{s}, \Theta)$$

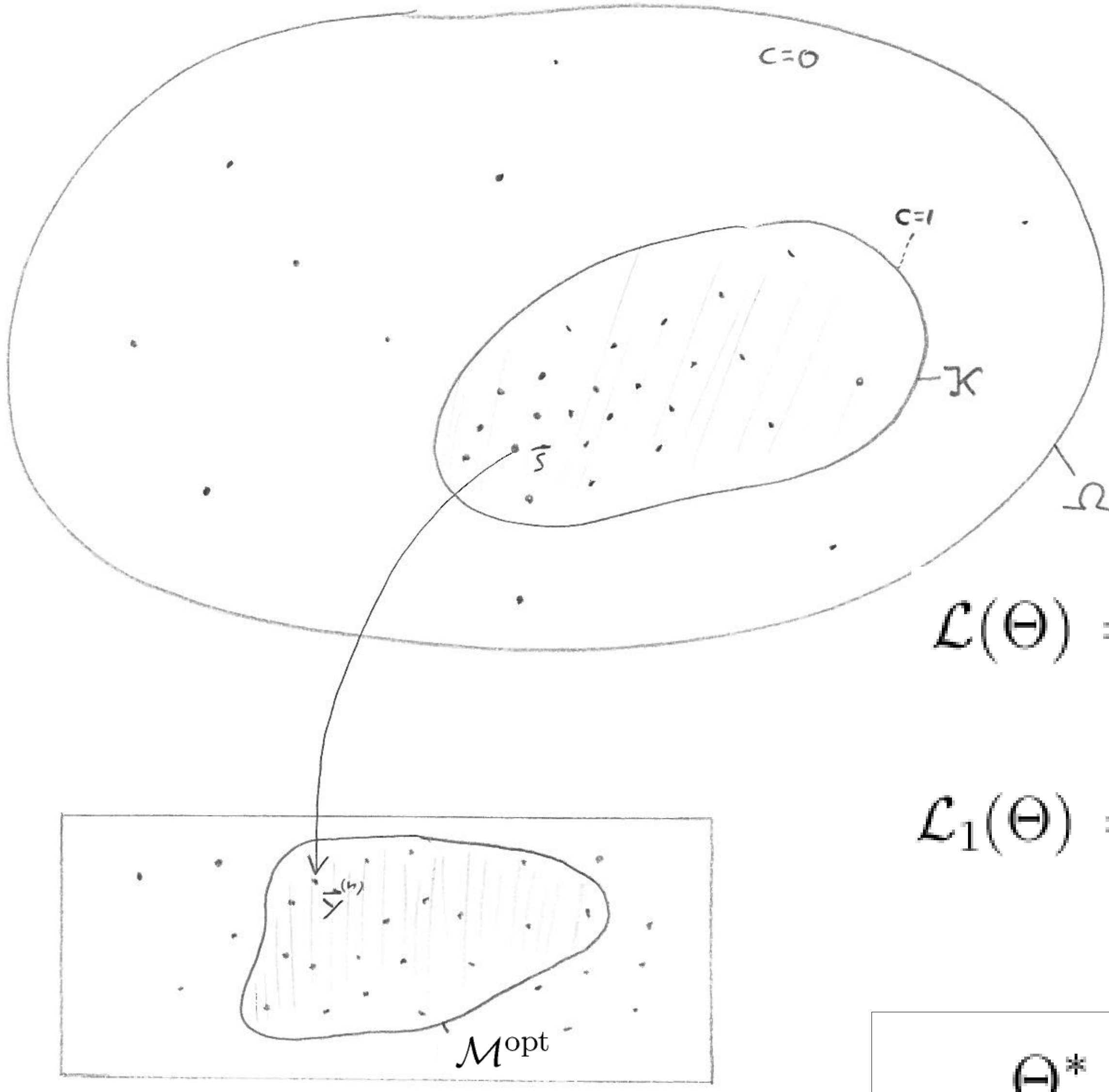
$$\mathcal{L}(\Theta) \;=\; \sum_{n=1,\ldots,N} \log(p(\vec{y}^{(n)} \,|\, \Theta)$$

$$\mathcal{L}_1(\Theta) \;=\; \sum_{n \in \mathcal{M}^{\mathrm{opt}}} \log(p(\vec{y}^{(n)} \,|\, c=1, \Theta)$$

$$\boxed{\begin{aligned} \Theta^* &= \mathrm{argmax}_\Theta\{\mathcal{L}(\Theta)\} \\ \Rightarrow \quad \Theta^* &\approx \mathrm{argmax}_\Theta\{\mathcal{L}_1(\Theta)\} \end{aligned}}$$

For the usual generative models.

On this a crucial result the following steps will be based on. It can be derived via a variational approach.
Note that it gives us a *necessary* condition for parameter optima.



$$\vec{s} \sim p(\vec{s}\,|\,\Theta)$$

$$\text{reject}\ \ \vec{s}\ \ \text{if}\ \ \vec{s} \notin \mathcal{K}$$

$$\vec{y} \sim p(\vec{y}\,|\,\vec{s},\Theta)$$

$$\mathcal{L}(\Theta) = \sum_{n=1,\ldots,N} \log(p(\vec{y}^{(n)}\,|\,\Theta))$$

$$\mathcal{L}_1(\Theta) = \sum_{n\in\mathcal{M}^{\rm opt}} \log(p(\vec{y}^{(n)}\,|\,c=1,\Theta))$$

Before we consider examples, we can use the result on the right to formulate an approximation scheme...

$$\Theta^* = \mathrm{argmax}_\Theta\{\mathcal{L}(\Theta)\}$$

$$\Rightarrow\ \Theta^* \approx \mathrm{argmax}_\Theta\{\mathcal{L}_1(\Theta)\}$$

For the usual generative models.

ET Algorithm

$$\Theta^* = \mathrm{argmax}_\Theta \{\mathcal{L}(\Theta)\}$$
$$\Rightarrow \Theta^* \approx \mathrm{argmax}_\Theta \{\mathcal{L}_1(\Theta)\}$$

$$\mathcal{L}_1(\Theta) = \sum_{n\in\mathcal{M}^{\mathrm{opt}}} \log(p(\vec{y}^{(n)}\,|\,c=1,\Theta)$$

ET Algorithm

$$\Theta^* = \mathrm{argmax}_\Theta \{\mathcal{L}(\Theta)\}$$
$$\Rightarrow \Theta^* \approx \mathrm{argmax}_\Theta \{\mathcal{L}_1(\Theta)\}$$

$$\mathcal{L}_1(\Theta) = \sum_{n \in \mathcal{M}^{\mathrm{opt}}} \log(p(\vec{y}^{(n)} \,|\, c = 1, \Theta)$$

$$\mathcal{F}_1(q, \Theta) = \sum_{n \in \mathcal{M}} \sum_{\vec{s} \in \mathcal{K}} q^{(n)}(\vec{s}; \Theta^{\mathrm{old}}) \, \log\Big(p(\vec{y}^{(n)} \,|\, \vec{s}, \Theta) \, \frac{p(\vec{s} \,|\, \Theta)}{\sum_{\vec{s}' \in \mathcal{K}} p(\vec{s}' \,|\, \Theta)} \Big) + H(q)$$

$$q^{(n)}(\vec{s}; \Theta^{\mathrm{old}}) = \frac{p(\vec{s} \,|\, \vec{y}^{(n)}, \Theta^{\mathrm{old}})}{\sum_{\vec{s}' \in \mathcal{K}} p(\vec{s}' \,|\, \vec{y}^{(n)}, \Theta^{\mathrm{old}})} \delta(\vec{s} \in \mathcal{K})$$

… this is a variational approximation.

ET Algorithm

$$\mathcal{L}_1(\Theta) = \sum_{n \in \mathcal{M}^{\mathrm{opt}}} \log(p(\vec{y}^{(n)} \,|\, c = 1, \Theta)$$

$$\Theta^* = \mathrm{argmax}_\Theta \{\mathcal{L}(\Theta)\}$$
$$\Rightarrow \Theta^* \approx \mathrm{argmax}_\Theta \{\mathcal{L}_1(\Theta)\}$$

$$\mathcal{F}_1(q, \Theta) = \sum_{n \in \mathcal{M}} \sum_{\vec{s} \in \mathcal{K}} q^{(n)}(\vec{s}\,; \Theta^{\mathrm{old}}) \, \log\left(p(\vec{y}^{(n)} \,|\, \vec{s}, \Theta) \frac{p(\vec{s} \,|\, \Theta)}{\sum_{\vec{s}\,' \in \mathcal{K}} p(\vec{s}\,' \,|\, \Theta)} \right) + H(q)$$

$$q^{(n)}(\vec{s}\,; \Theta^{\mathrm{old}}) = \frac{p(\vec{s} \,|\, \vec{y}^{(n)}, \Theta^{\mathrm{old}})}{\sum_{\vec{s}\,' \in \mathcal{K}} p(\vec{s}\,' \,|\, \vec{y}^{(n)}, \Theta^{\mathrm{old}})} \delta(\vec{s} \in \mathcal{K})$$

Algorithm 1: Expectation Truncation (step 1)

Initial: select a state space volume $\mathcal{K}$

Data classification: find a data set $\mathcal{M}$ that approximates $\mathcal{M}^{\mathrm{opt}}$

E-step: compute $q^{(n)}(\vec{s}\,; \Theta^{\mathrm{old}}) = \dfrac{p(\vec{s}, \vec{y}^{(n)} \,|\, \Theta^{\mathrm{old}})}{\sum_{\vec{s} \in \mathcal{K}} p(\vec{s}, \vec{y}^{(n)} \,|\, \Theta^{\mathrm{old}})}$ for all $\vec{y}^{(n)}$ and $\vec{s} \in \mathcal{K}$

M-step: find parameters Θ such that

$$\frac{d}{d\Theta} \sum_{n \in \mathcal{M}} \sum_{\vec{s} \in \mathcal{K}} q^{(n)}(\vec{s}\,; \Theta^{\mathrm{old}}) \, \log\left(p(\vec{y}^{(n)} \,|\, \vec{s}, \Theta) \frac{p(\vec{s} \,|\, \Theta)}{\sum_{\vec{s}\,' \in \mathcal{K}} p(\vec{s}\,' \,|\, \Theta)} \right)$$

Example

$$\mathcal{K} = \{\vec{s} \mid \sum_j s_j \leq \gamma\}$$

The choice assumes that on average only few components generate a data point.

Binary Sparse Coding (BSC):

$$p(\vec{s} \,|\, \Theta) = \prod_h \pi^{s_h} \,(1 - \pi \;)^{1 - s_h}$$

$$p(\vec{y} \,|\, \vec{s}, \Theta) = \mathcal{N}(\vec{y}; \sum_h s_h \vec{W}_h, \sigma^2 \, \mathbb{1})$$

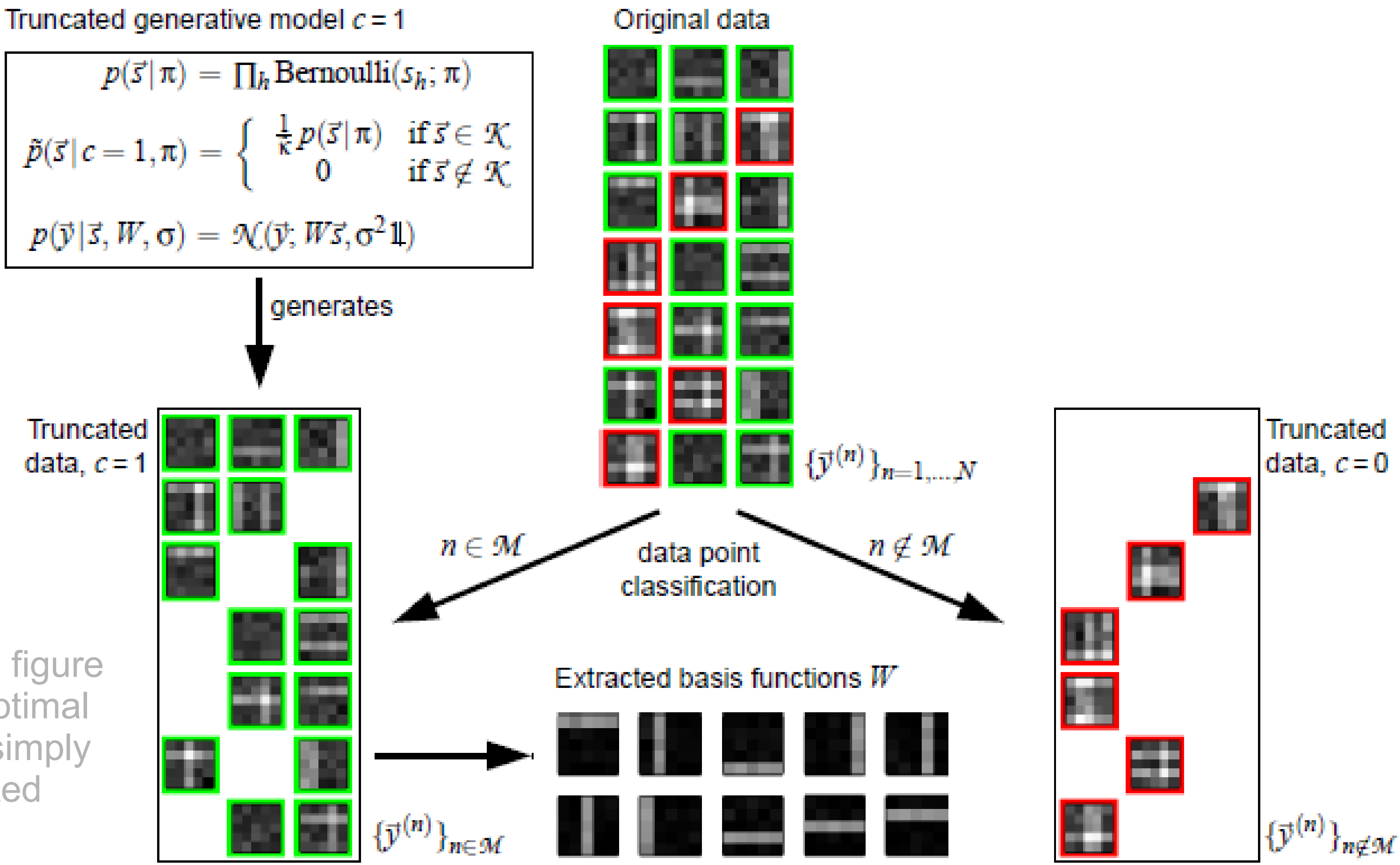
Henniges et al., 2010

This is a sparse coding generative model with binary hidden units.

Example

$$\mathcal{K} = \{\vec{s} \mid \textstyle\sum_j s_j \leq \gamma\}$$

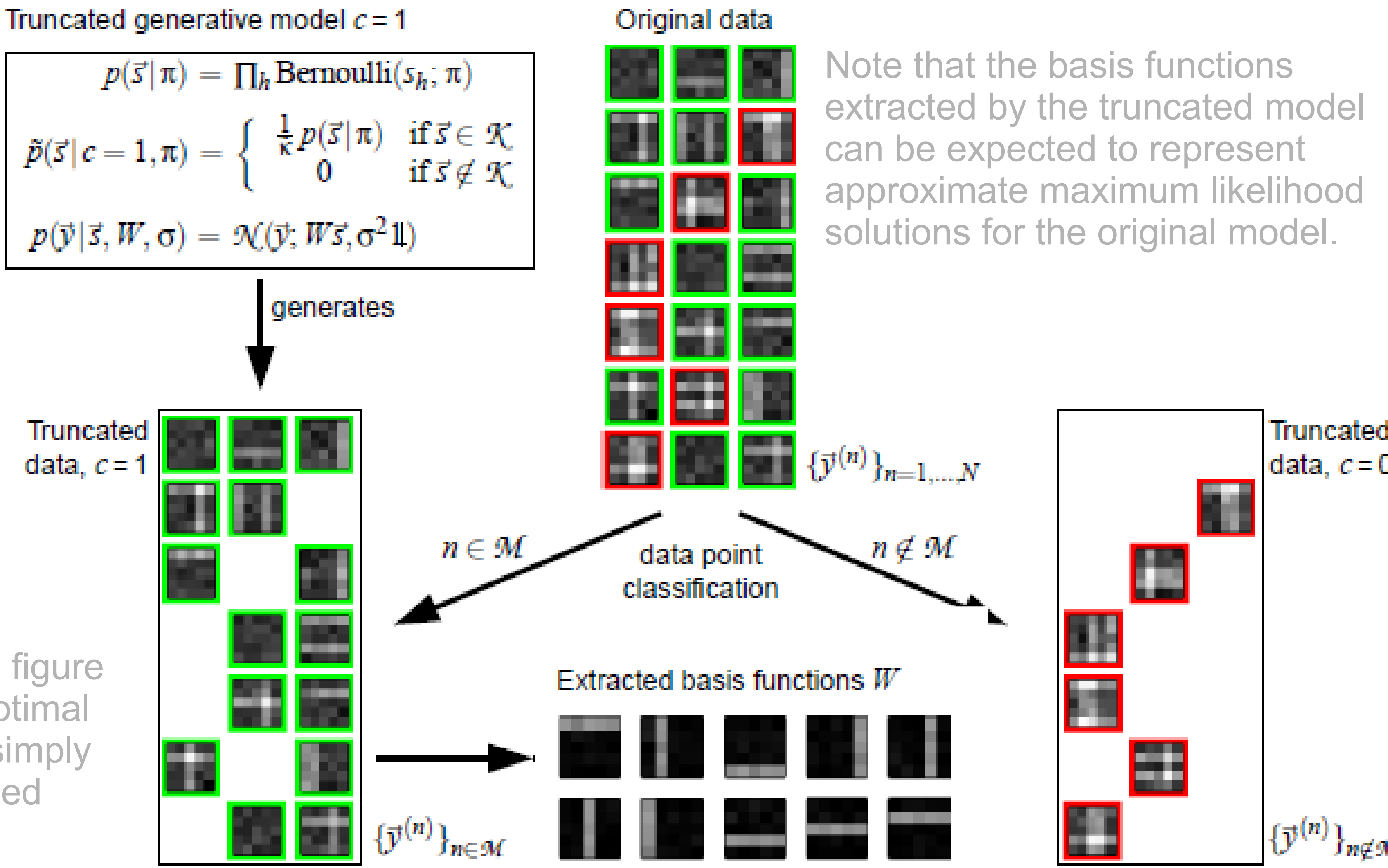
The choice assumes that on average only few components generate a data point. In the figure gamma is equal to two.



In the figure the optimal M is simply denoted by M.

Example

$$\mathcal{K} = \{\vec{s} \mid \textstyle\sum_j s_j \leq \gamma\}$$

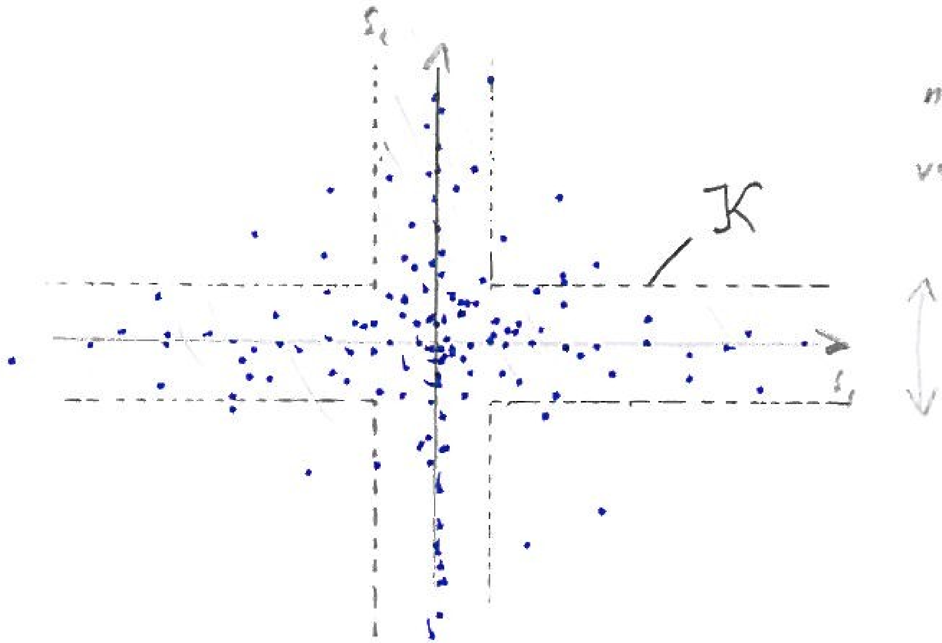


Note that the basis functions extracted by the truncated model can be expected to represent approximate maximum likelihood solutions for the original model.

In the figure the optimal M is simply denoted by M.

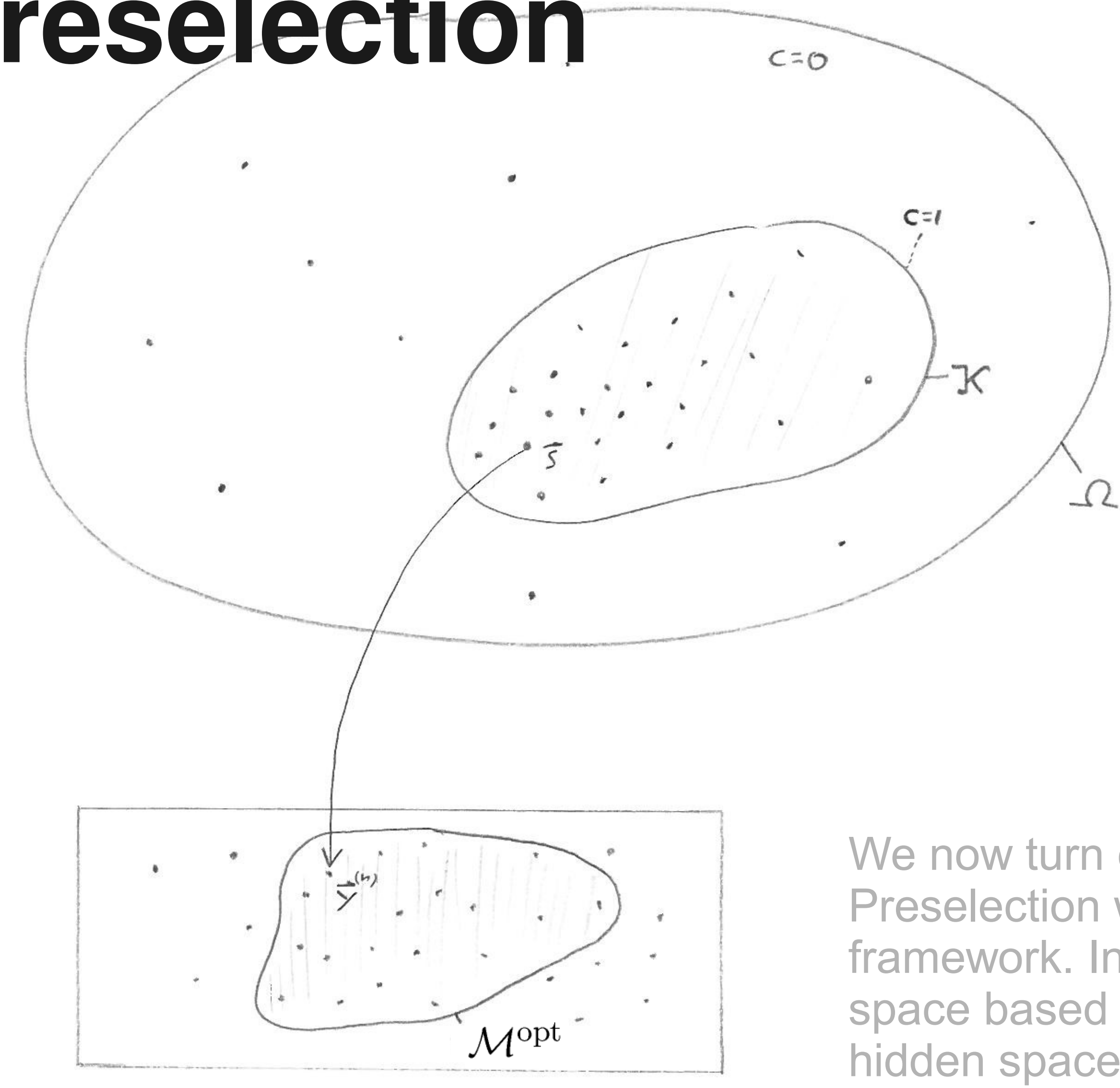
Example Sparse Coding

$$p(\vec{s}\,|\,\Theta) = \prod_{h=1}^{H} \mathrm{Cauchy}(s_h; \pi)$$



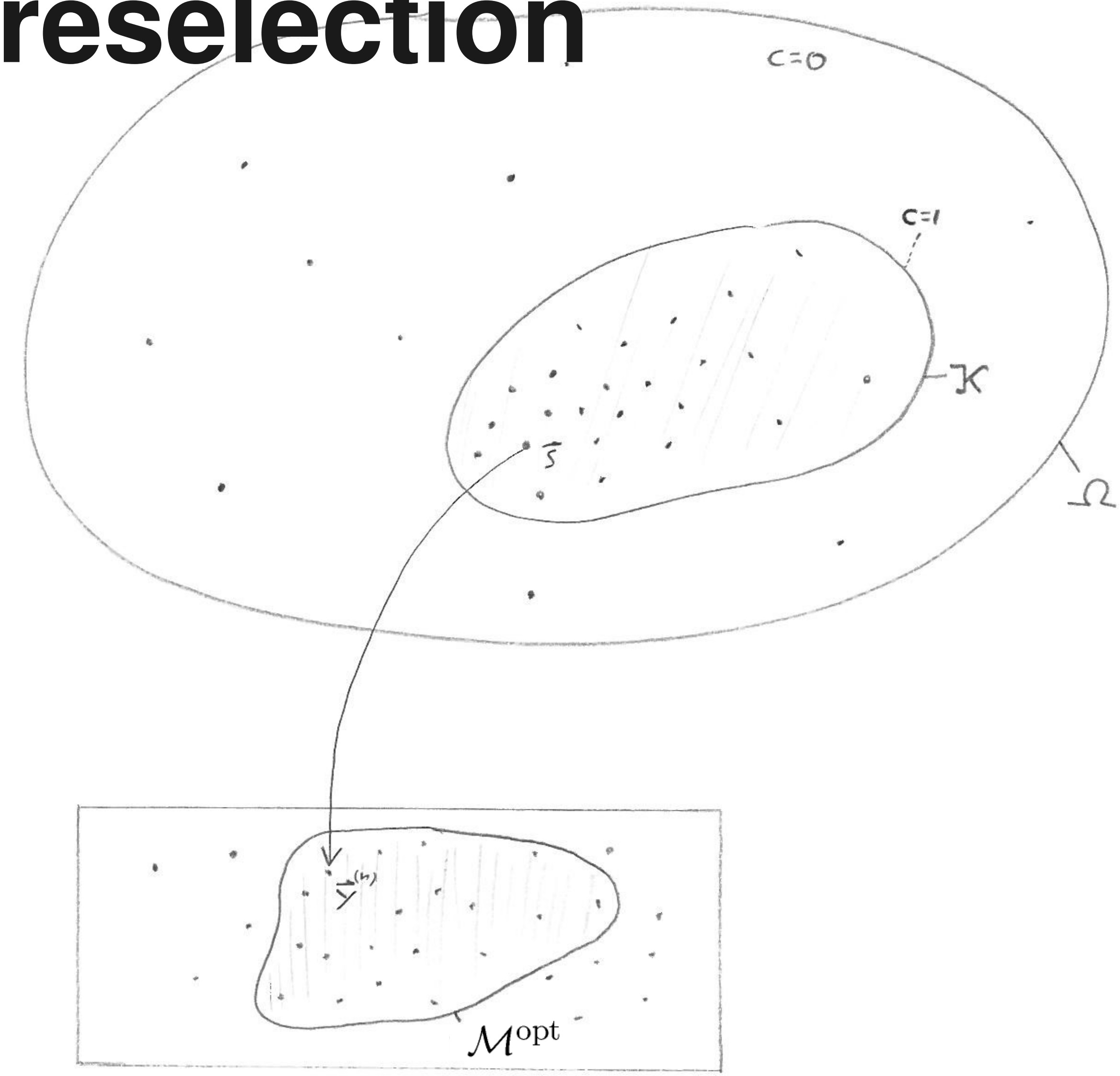
note that the volume of $\mathcal{K}$ is 'much smaller' than the volume of Ω

Preselection



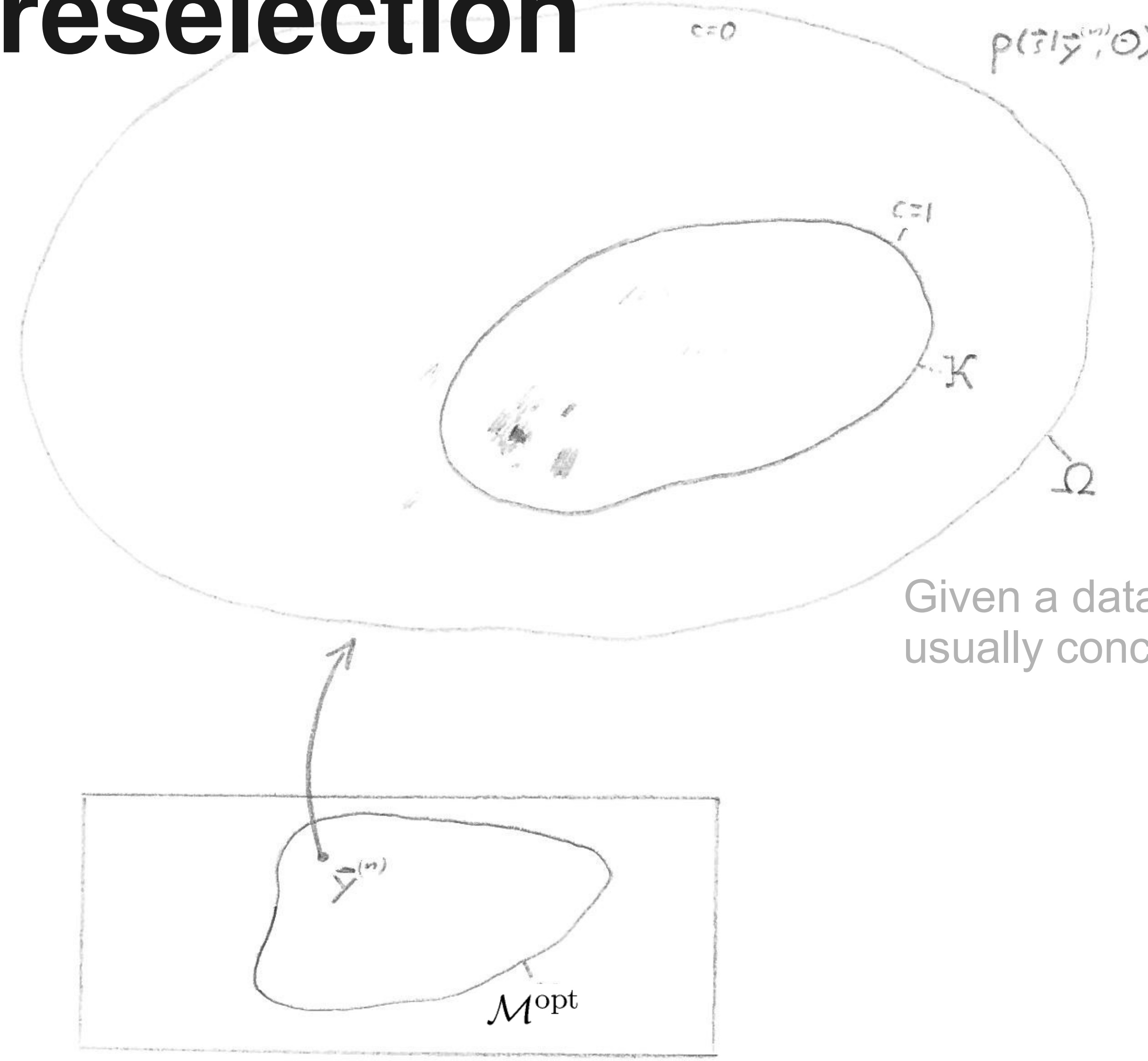
We now turn our attention back to preselection. Preselection will be formulated within the same framework. Instead of constraining the hidden space based on the prior, it will constrain the hidden space based on the posterior.

Preselection



Idea 1: Define a set $\mathcal{K}$ that contains **most prior mass**.

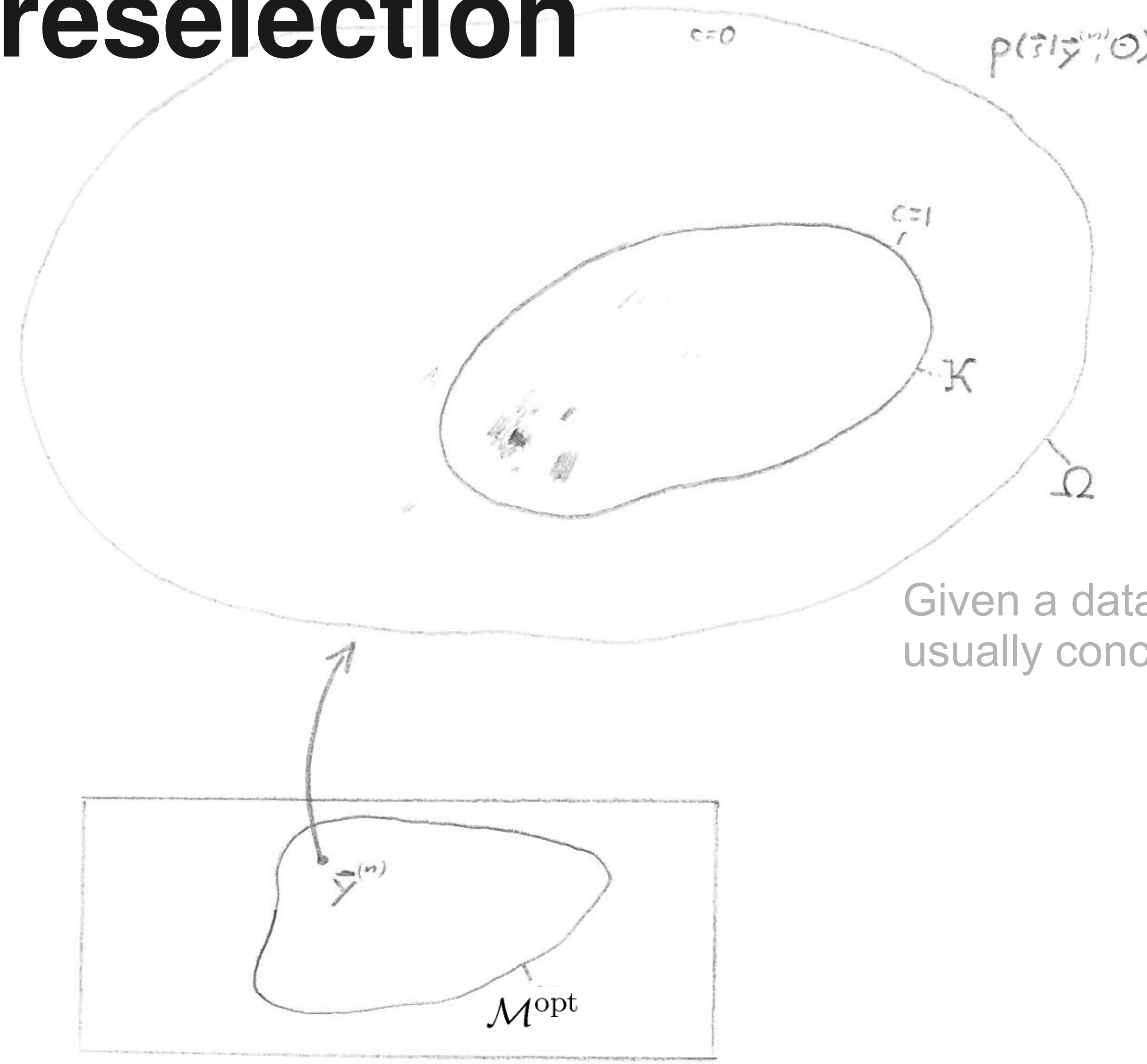
Preselection



Idea 1: Define a set $\mathcal{K}$ that contians **most prior mass**.

Given a data point, the posterior mass is usually concentrated in small volumes (grey).

Preselection

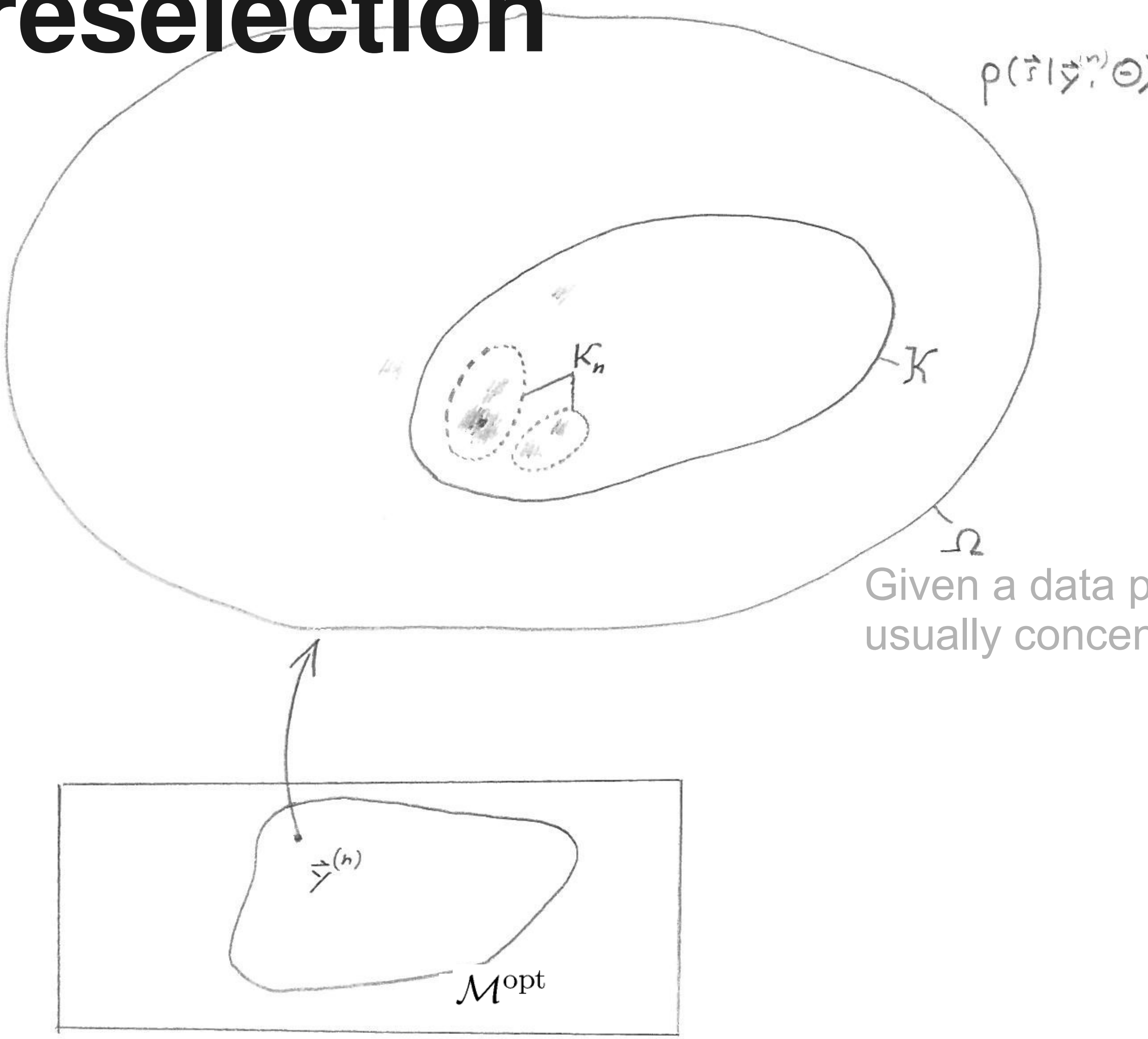


Idea 1: Define a set $\mathcal{K}$ that contians **most prior mass**.

Idea 2: Define a set $\mathcal{K}_n$ that contains **most posterior mass**.

Given a data point, the posterior mass is usually concentrated in small volumes (grey).

Preselection

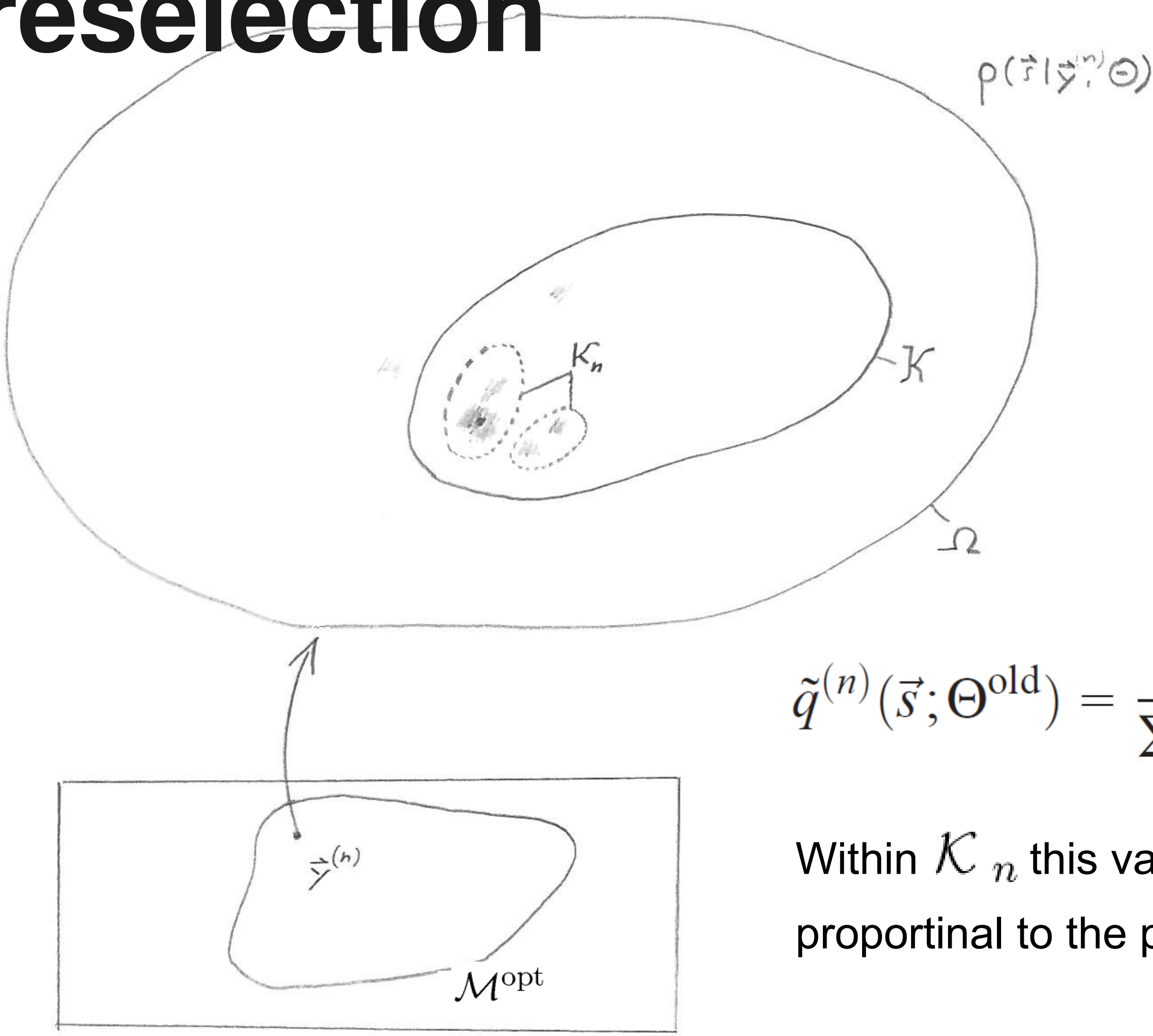


Idea 1: Define a set $\mathcal{K}$ that contians **most prior mass**.

Idea 2: Define a set $\mathcal{K}_n$ that contains **most posterior mass**.

Given a data point, the posterior mass is usually concentrated in small volumes (grey).

Preselection



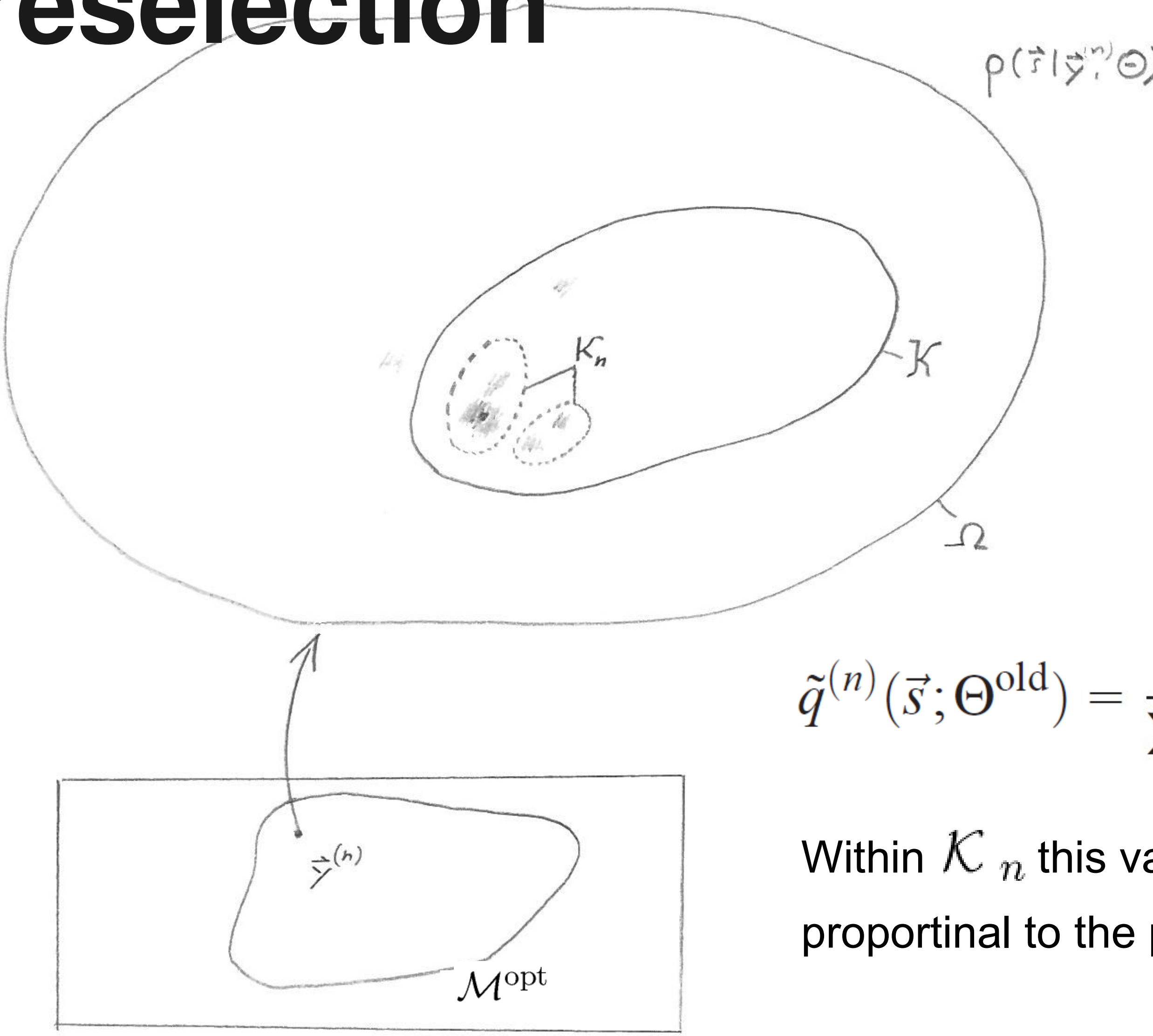
Idea 1: Define a set $\mathcal{K}$ that contians **most prior mass**.

Idea 2: Define a set $\mathcal{K}_n$ that contains **most posterior mass**.

$$\tilde{q}^{(n)}(\vec{s};\Theta^{\text{old}}) = \frac{p(\vec{s}, \vec{y}^{(n)} \,|\, \Theta^{\text{old}})}{\sum_{\vec{s}' \in \mathcal{K}_n} p(\vec{s}', \vec{y}^{(n)} \,|\, \Theta^{\text{old}})} \, \delta(\vec{s} \in \mathcal{K}_n)$$

Within $\mathcal{K}_n$ this variational distribution is proportinal to the posterior in $\mathcal{K}$.

Preselection



Idea 1: Define a set $\mathcal{K}$ that contians **most prior mass**.

Idea 2: Define a set $\mathcal{K}_n$ that contains **most posterior mass**.

$$\tilde{q}^{(n)}(\vec{s};\Theta^{\mathrm{old}}) = \frac{p(\vec{s},\vec{y}^{(n)}\,|\,\Theta^{\mathrm{old}})}{\sum_{\vec{s}\,'\in\mathcal{K}_n} p(\vec{s}\,',\vec{y}^{(n)}\,|\,\Theta^{\mathrm{old}})}\,\delta(\vec{s}\in\mathcal{K}_n)$$

Within $\mathcal{K}_n$ this variational distribution is proportinal to the posterior in $\mathcal{K}$.

Idea:

Find $\mathcal{K}_n$ by fast preselection!

ET Algorithm

$$\Theta^* = \mathrm{argmax}_\Theta \{\mathcal{L}(\Theta)\}$$
$$\Rightarrow \Theta^* \approx \mathrm{argmax}_\Theta \{\mathcal{L}_1(\Theta)\}$$

$$\mathcal{L}_1(\Theta) = \sum_{n \in \mathcal{M}^{\mathrm{opt}}} \log(p(\vec{y}^{(n)} \,|\, c = 1, \Theta))$$

$$\tilde{q}^{(n)}(\vec{s};\Theta^{\mathrm{old}}) = \frac{p(\vec{s}, \vec{y}^{(n)} \,|\, \Theta^{\mathrm{old}})}{\sum_{\vec{s}' \in \mathcal{K}_n} p(\vec{s}', \vec{y}^{(n)} \,|\, \Theta^{\mathrm{old}})} \delta(\vec{s} \in \mathcal{K}_n)$$

Algorithm 2: Expectation Truncation (step 1 + 2)

Preselection:	select a state space volume $\mathcal{K}_n$ for each data point $\vec{y}^{(n)}$
Data classification:	find a data set $\mathcal{M}$ that approximates $\mathcal{M}^{\mathrm{opt}}$
E-step:	compute $\tilde{q}^{(n)}(\vec{s};\Theta^{\mathrm{old}}) = \frac{p(\vec{s},\, \vec{y}^{(n)} \,|\, \Theta^{\mathrm{old}})}{\sum_{\vec{s} \in \mathcal{K}_n} p(\vec{s},\, \vec{y}^{(n)} \,|\, \Theta^{\mathrm{old}})}$ for all $\vec{y}^{(n)}$ and $\vec{s} \in \mathcal{K}_n$
M-step:	find parameters Θ such that $\frac{d}{d\Theta} \sum_{n \in \mathcal{M}} \sum_{\vec{s} \in \mathcal{K}_n} \tilde{q}^{(n)}(\vec{s};\Theta^{\mathrm{old}}) \log\left(p(\vec{y}^{(n)} \,|\, \vec{s}, \Theta) \frac{p(\vec{s} \,|\, \Theta)}{\sum_{\vec{s}' \in \mathcal{K}} p(\vec{s}' \,|\, \Theta)}\right) = 0$

Example BSC

$$\mathcal{K} = \{\vec{s} \mid \sum_j s_j \leq \gamma\}$$

Binary Sparse Coding (BSC):

$$p(\vec{s}\,|\,\Theta) = \prod_h \pi^{s_h}\,(1-\pi\ \ )^{1-s_h}$$

$$p(\vec{y}\,|\,\vec{s},\Theta) = \mathcal{N}(\vec{y}; \sum_h s_h \vec{W}_h, \sigma^2\,\mathbb{1})$$

Henniges et al., 2010

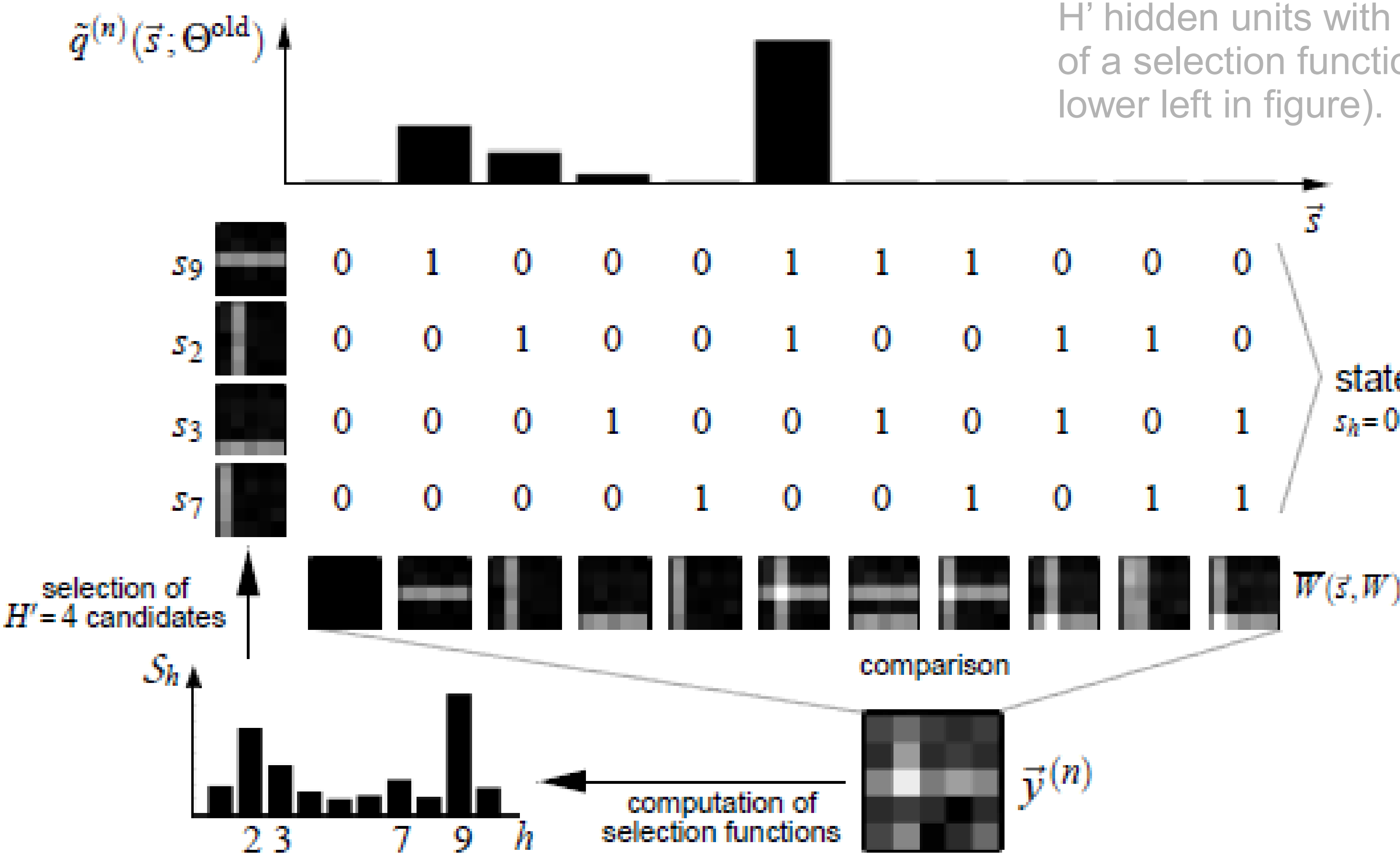
This is a sparse coding generative model with binary hidden units.

Example BSC

$$\mathcal{K} = \{\vec{s} \mid \sum_j s_j \leq \gamma\}$$

$$\mathcal{K}_n = \{\vec{s} \mid \sum_j s_j \leq \gamma \text{ and } (\forall i \notin I : s_i = 0)\}$$

The set ***I*** is the index set of the H' hidden units with largest values of a selection function S_h (see lower left in figure).



Example BSC

$$\mathcal{K} = \{\vec{s} \mid \sum_j s_j \le \gamma\}$$

$$\mathcal{K}_n = \{\vec{s} \mid \sum_j s_j \le \gamma \text{ and } (\forall i \notin I : s_i = 0)\}$$

Binary Sparse Coding (BSC):

$$p(\vec{s} \mid \Theta) = \prod_h \pi^{s_h} (1 - \pi)^{1 - s_h}$$

$$p(\vec{y} \mid \vec{s}, \Theta) = \mathcal{N}(\vec{y}; \sum_h s_h \vec{W}_h, \sigma^2 \mathbb{1})$$

Henniges et al., 2010

This is a sparse coding generative model with binary hidden units.

Exact EM updates

$$W^{\text{new}} = \left(\sum_n \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_p^T \right) \left(\sum_n \langle \boldsymbol{s}\,\boldsymbol{s}^T \rangle_p \right)^{-1}$$

$$\sigma^{\text{new}} = \sqrt{\frac{1}{N D} \sum_n \left\langle \|\boldsymbol{y}^{(n)} - W \boldsymbol{s}\|^2 \right\rangle_p}$$

$$\pi^{\text{new}} = \frac{1}{ND} \sum_n \langle |\boldsymbol{s}| \rangle_{p^n}$$

$$\text{with } |\boldsymbol{s}| = \sum_{h=1}^{H} s_h$$

Example BSC

$$\mathcal{K} = \{\vec{s} \mid \sum_j s_j \leq \gamma\}$$

$$\mathcal{K}_n = \{\vec{s} \mid \sum_j s_j \leq \gamma \text{ and } (\forall i \notin I : s_i = 0)\}$$

Binary Sparse Coding (BSC):

$$p(\vec{s} \mid \Theta) = \prod_h \pi^{s_h} (1 - \pi)^{1 - s_h}$$

$$p(\vec{y} \mid \vec{s}, \Theta) = \mathcal{N}(\vec{y}; \sum_h s_h \vec{W}_h, \sigma^2 \mathbb{1})$$

Henniges et al., 2010

This is a sparse coding generative model with binary hidden units.

Exact EM updates

$$W^{\text{new}} = \left(\sum_n \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_p^T \right) \left(\sum_n \langle \boldsymbol{s}\, \boldsymbol{s}^T \rangle_p \right)^{-1}$$

$$\sigma^{\text{new}} = \sqrt{\frac{1}{N D} \sum_n \left\langle \|\boldsymbol{y}^{(n)} - W \boldsymbol{s}\|^2 \right\rangle_p}$$

$$\pi^{\text{new}} = \frac{1}{ND} \sum_n \langle |\boldsymbol{s}| \rangle_{p^n}$$

$$\text{with } |\boldsymbol{s}| = \sum_{h=1}^{H} s_h$$

ET-EM updates

$$W^{\text{new}} = \left(\sum_{n \in \mathcal{M}} \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_{q_n}^T \right) \left(\sum_{n' \in \mathcal{M}} \langle \boldsymbol{s}\, \boldsymbol{s}^T \rangle_{q_{n'}} \right)^{-1}$$

$$\sigma^{\text{new}} = \sqrt{\frac{1}{|\mathcal{M}|\, D} \sum_{n \in \mathcal{M}} \left\langle \|\boldsymbol{y}^{(n)} - W \boldsymbol{s}\|^2 \right\rangle_{q_n}}$$

These update rules are essentially the same. Only the summation and expectations change.

Example BSC

$$\mathcal{K} = \{\vec{s} \mid \sum_j s_j \leq \gamma\}$$

$$\mathcal{K}_n = \{\vec{s} \mid \sum_j s_j \leq \gamma \text{ and } (\forall i \notin I : s_i = 0)\}$$

Binary Sparse Coding (BSC):

$$p(\vec{s} \mid \Theta) = \prod_h \pi^{s_h} (1 - \pi)^{1 - s_h}$$

$$p(\vec{y} \mid \vec{s}, \Theta) = \mathcal{N}(\vec{y}; \sum_h s_h \vec{W}_h, \sigma^2 \mathbb{1})$$

Henniges et al., 2010

This is a sparse coding generative model with binary hidden units.

Exact EM updates

$$W^{\text{new}} = \left(\sum_n \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_p^T \right) \left(\sum_n \langle \boldsymbol{s}\,\boldsymbol{s}^T \rangle_p \right)^{-1}$$

$$\sigma^{\text{new}} = \sqrt{\frac{1}{N D} \sum_n \left\langle ||\boldsymbol{y}^{(n)} - W\,\boldsymbol{s}||^2 \right\rangle_p}$$

$$\pi^{\text{new}} = \frac{1}{ND} \sum_n \langle |\boldsymbol{s}| \rangle_{p^n}$$

$$\text{with } |\boldsymbol{s}| = \sum_{h=1}^{H} s_h$$

ET-EM updates

$$W^{\text{new}} = \left(\sum_{n \in \mathcal{M}} \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_{q_n}^T \right) \left(\sum_{n' \in \mathcal{M}} \langle \boldsymbol{s}\,\boldsymbol{s}^T \rangle_{q_{n'}} \right)^{-1}$$

$$\sigma^{\text{new}} = \sqrt{\frac{1}{|\mathcal{M}|\, D} \sum_{n \in \mathcal{M}} \left\langle ||\boldsymbol{y}^{(n)} - W\,\boldsymbol{s}||^2 \right\rangle_{q_n}}$$

$$\pi^{\text{new}} = \frac{A(\pi)\,\pi}{B(\pi)} \frac{1}{|\mathcal{M}|} \sum_{n \in \mathcal{M}} \langle |\boldsymbol{s}| \rangle_{q_n}$$

Because of the modified free-energy the update of prior parameters changes more significantly.

Example BSC

$$\mathcal{K} = \{\vec{s} \mid \sum_j s_j \leq \gamma\}$$

$$\mathcal{K}_n = \{\vec{s} \mid \sum_j s_j \leq \gamma \text{ and } (\forall i \notin I : s_i = 0)\}$$

Binary Sparse Coding (BSC):

$$p(\vec{s} \mid \Theta) = \prod_h \pi^{s_h} (1 - \pi)^{1 - s_h}$$

$$p(\vec{y} \mid \vec{s}, \Theta) = \mathcal{N}(\vec{y}; \sum_h s_h \vec{W}_h, \sigma^2 \mathbb{1})$$

Henniges et al., 2010

This is a sparse coding generative model with binary hidden units.

Exact EM updates

$$W^{\text{new}} = \left(\sum_n \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_p^T\right) \left(\sum_n \langle \boldsymbol{s}\,\boldsymbol{s}^T \rangle_p\right)^{-1}$$

$$\sigma^{\text{new}} = \sqrt{\frac{1}{N\,D} \sum_n \left\langle ||\boldsymbol{y}^{(n)} - W\,\boldsymbol{s}||^2 \right\rangle_p}$$

$$\pi^{\text{new}} = \frac{1}{ND} \sum_n \langle |\boldsymbol{s}| \rangle_{p^n}$$

$$\text{with } |\boldsymbol{s}| = \sum_{h=1}^{H} s_h$$

ET-EM updates

$$W^{\text{new}} = \left(\sum_{n \in \mathcal{M}} \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_{q_n}^T\right) \left(\sum_{n' \in \mathcal{M}} \langle \boldsymbol{s}\,\boldsymbol{s}^T \rangle_{q_{n'}}\right)^{-1}$$

$$\sigma^{\text{new}} = \sqrt{\frac{1}{|\mathcal{M}|\,D} \sum_{n \in \mathcal{M}} \left\langle ||\boldsymbol{y}^{(n)} - W\,\boldsymbol{s}||^2 \right\rangle_{q_n}}$$

$$\pi^{\text{new}} = \frac{A(\pi)\,\pi}{B(\pi)} \frac{1}{|\mathcal{M}|} \sum_{n \in \mathcal{M}} \langle |\boldsymbol{s}| \rangle_{q_n}$$

$$A(\pi) = \sum_{\gamma'=0}^{\gamma} \binom{H}{\gamma'} \pi^{\gamma'} (1 - \pi)^{H - \gamma'} \qquad B(\pi) = \sum_{\gamma'=0}^{\gamma} \gamma' \binom{H}{\gamma'} \pi^{\gamma'} (1 - \pi)^{H - \gamma'}$$

Example BSC

$$\mathcal{K} = \{\vec{s} \mid \sum_j s_j \leq \gamma\}$$

$$\mathcal{K}_n = \{\vec{s} \mid \sum_j s_j \leq \gamma \text{ and } (\forall i \notin I : s_i = 0)\}$$

Binary Sparse Coding (BSC):

$$p(\vec{s} \mid \Theta) = \prod_h \pi^{s_h} (1 - \pi)^{1-s_h}$$

$$p(\vec{y} \mid \vec{s}, \Theta) = \mathcal{N}(\vec{y}; \sum_h s_h \vec{W}_h, \sigma^2 \mathbb{1})$$

Henniges et al., 2010

This is a sparse coding generative model with binary hidden units.

$$\langle g(\boldsymbol{s}) \rangle_{q^{(n)}} = \frac{\sum_{\boldsymbol{s} \in \mathcal{K}_n} p(\boldsymbol{s}, \boldsymbol{y}^{(n)} \mid \Theta^{\mathrm{old}})\, g(\boldsymbol{s})}{\sum_{\tilde{\boldsymbol{s}} \in \mathcal{K}_n} p(\tilde{\boldsymbol{s}}, \boldsymbol{y}^{(n)} \mid \Theta^{\mathrm{old}})}$$

Efficiently computable ET expectation value.

ET-EM updates

$$W^{\mathrm{new}} = \left(\sum_{n \in \mathcal{M}} \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_{q_n}^T \right) \left(\sum_{n' \in \mathcal{M}} \langle \boldsymbol{s}\, \boldsymbol{s}^T \rangle_{q_{n'}} \right)^{-1}$$

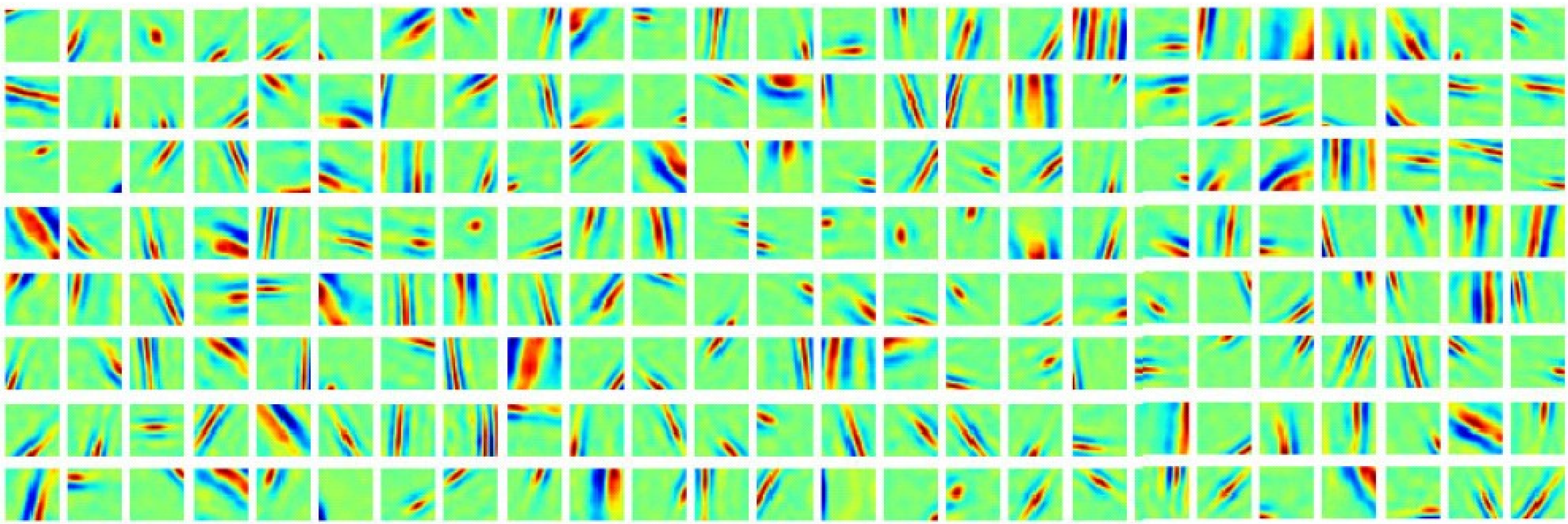
$$\sigma^{\mathrm{new}} = \sqrt{\frac{1}{|\mathcal{M}|\, D} \sum_{n \in \mathcal{M}} \left\langle \left|\left| \boldsymbol{y}^{(n)} - W \boldsymbol{s} \right|\right|^2 \right\rangle_{q_n}}$$

$$\pi^{\mathrm{new}} = \frac{A(\pi)\,\pi}{B(\pi)} \frac{1}{|\mathcal{M}|} \sum_{n \in \mathcal{M}} \langle |\boldsymbol{s}| \rangle_{q_n}$$

$$A(\pi) = \sum_{\gamma'=0}^{\gamma} \binom{H}{\gamma'} \pi^{\gamma'} (1 - \pi)^{H-\gamma'} \qquad B(\pi) = \sum_{\gamma'=0}^{\gamma} \gamma' \binom{H}{\gamma'} \pi^{\gamma'} (1 - \pi)^{H-\gamma'}$$

Example BSC

Binary Sparse Coding can now be scaled up
and can, e.g., be applied to image patches:

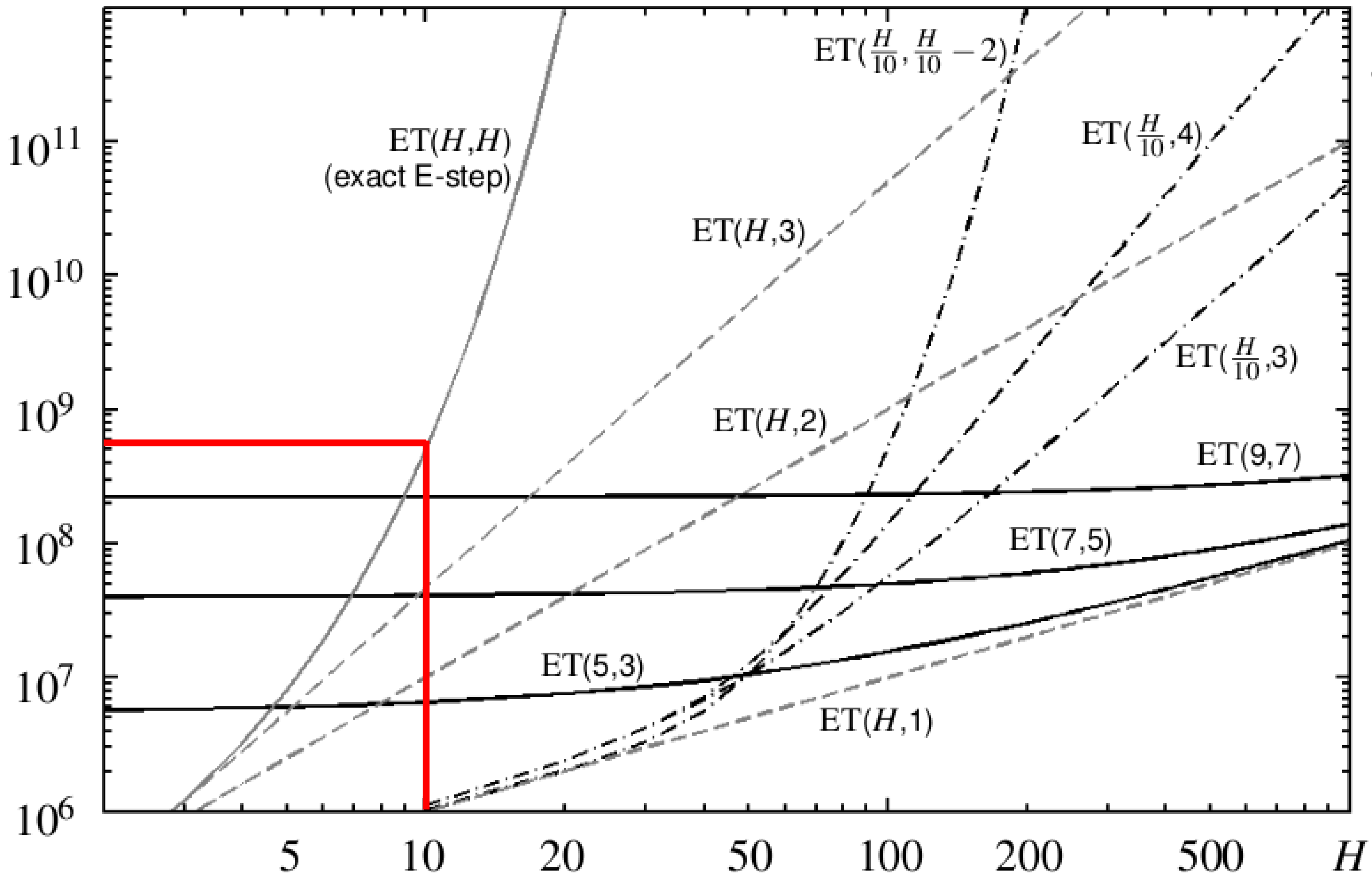


Random selection of 200 of 700 basis functions if Binary Sparse Coding is applied to natural image patches (Henniges et al., Proc. LVA/ICA 2010).

Animations showing basis function modifications and the selection of data set M are provided on:
fias.uni-frankfurt.de/cnml → Selected Publications

Complexity of BSC

E-step complexity



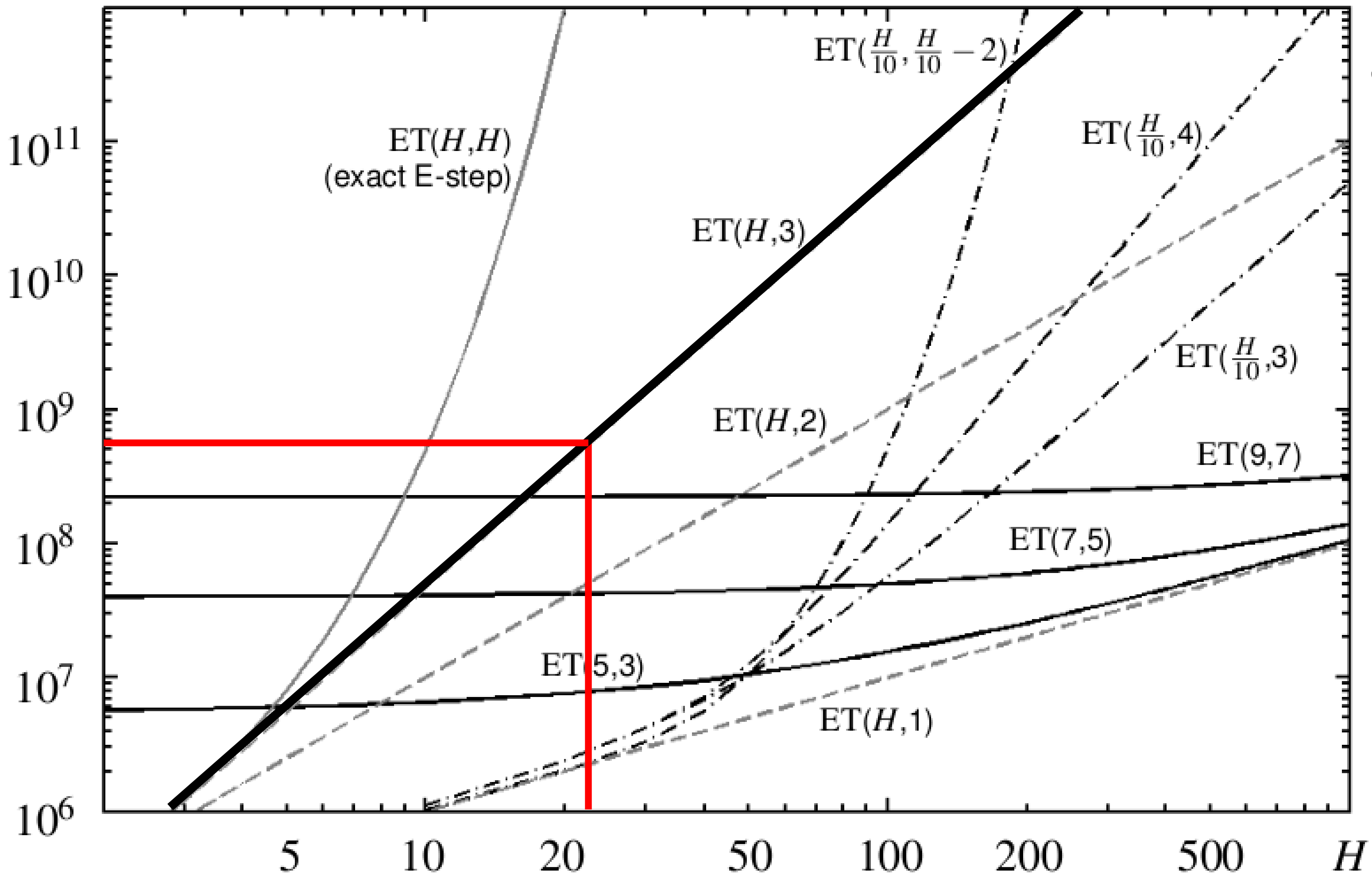
Lücke, Eggert, *JMLR* 2010;

exact EM

$\mathcal{O}(e^H)$ Complexity of an exact E-step. No approximation.

Complexity of BSC

E-step complexity



Lücke, Eggert, *JMLR* 2010;

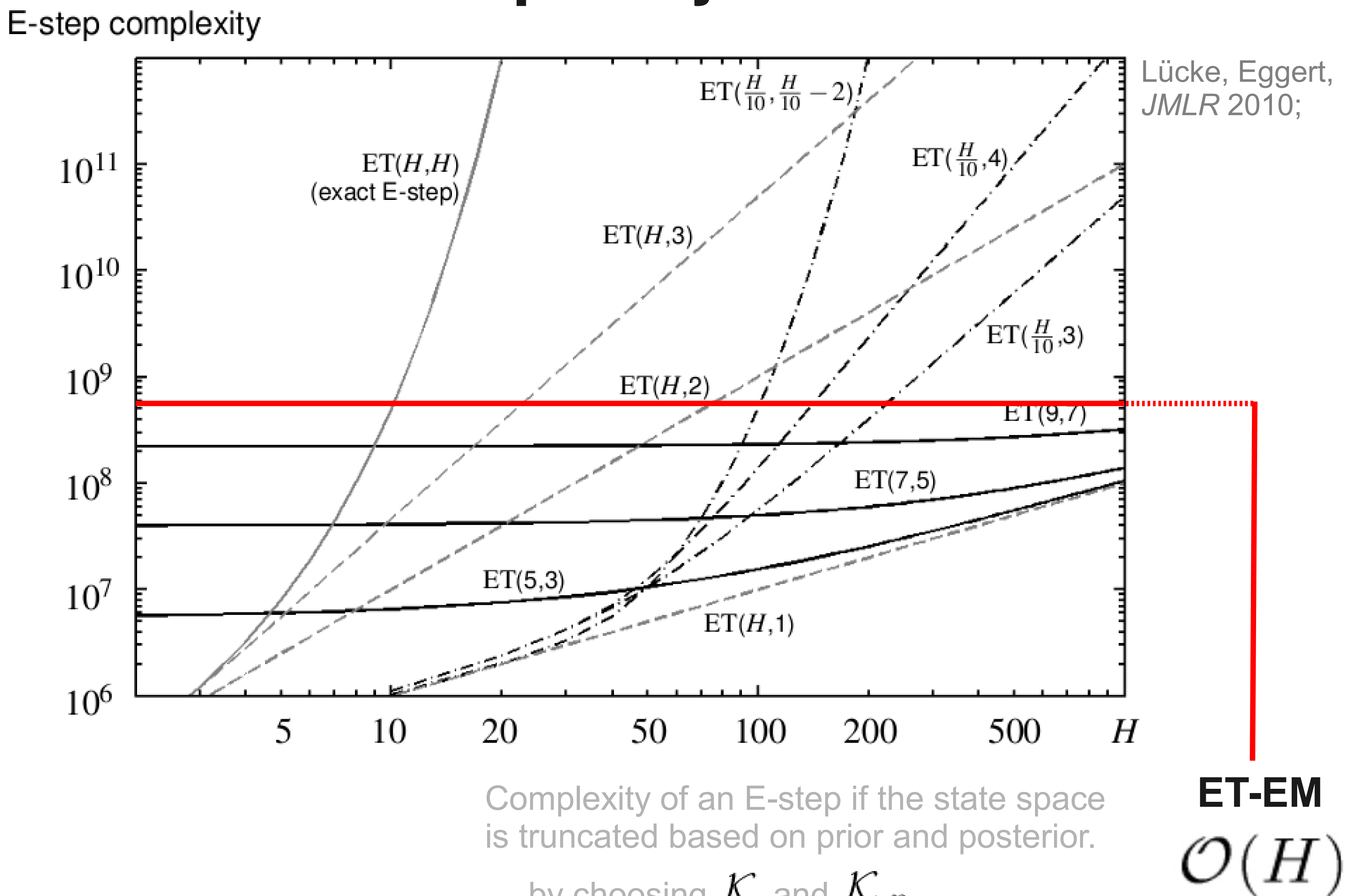
approx. EM

$$\mathcal{O}(H^k)$$

Complexity of an E-step if the state space is truncated based on the prior.

… by choosing $\mathcal{K}$.

Complexity of BSC

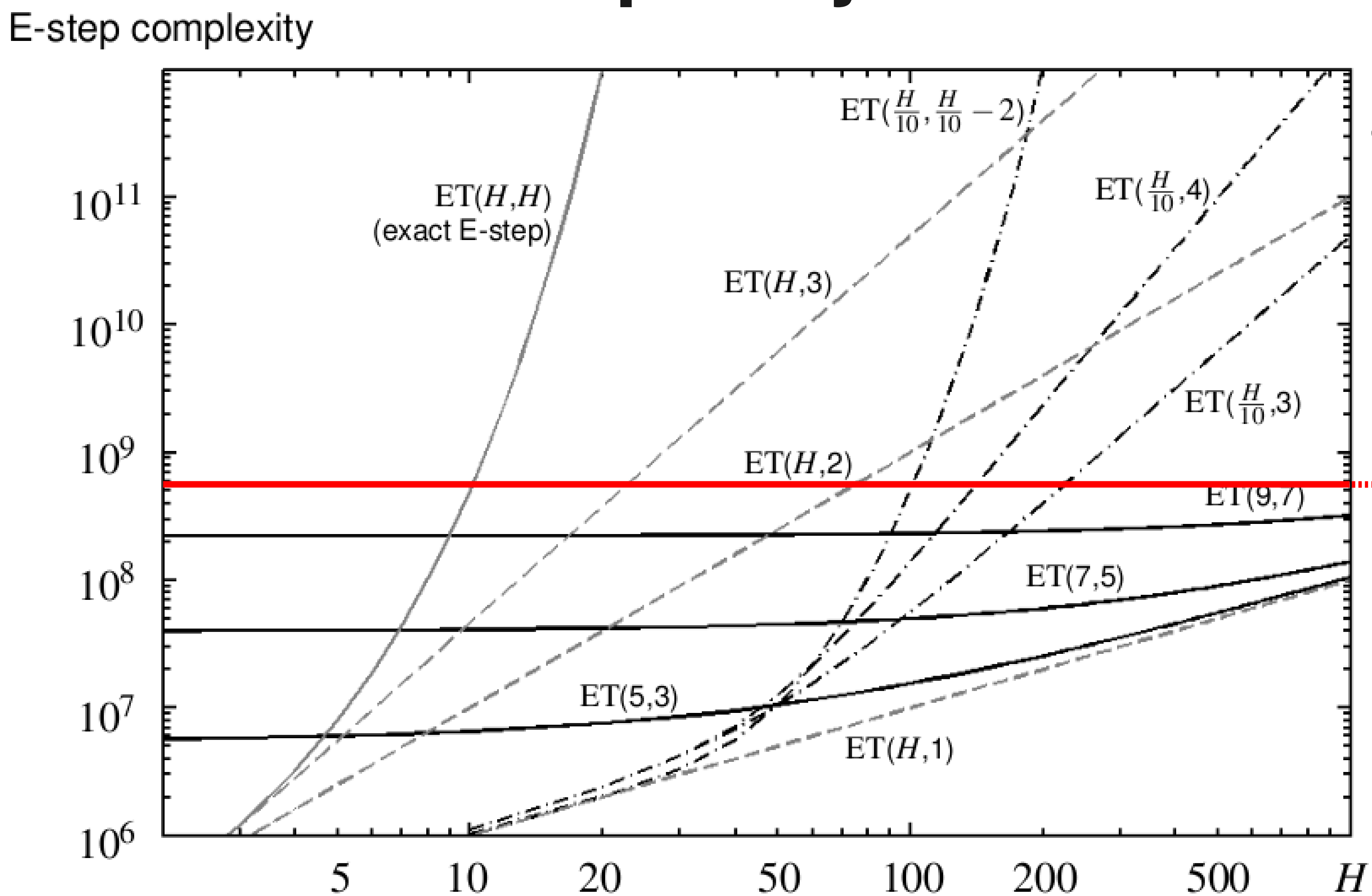


Lücke, Eggert, *JMLR* 2010;

Complexity of an E-step if the state space is truncated based on prior and posterior.

… by choosing $\mathcal{K}$ and $\mathcal{K}_n$.

Complexity of BSC



Lücke, Eggert, *JMLR* 2010;

ET-EM

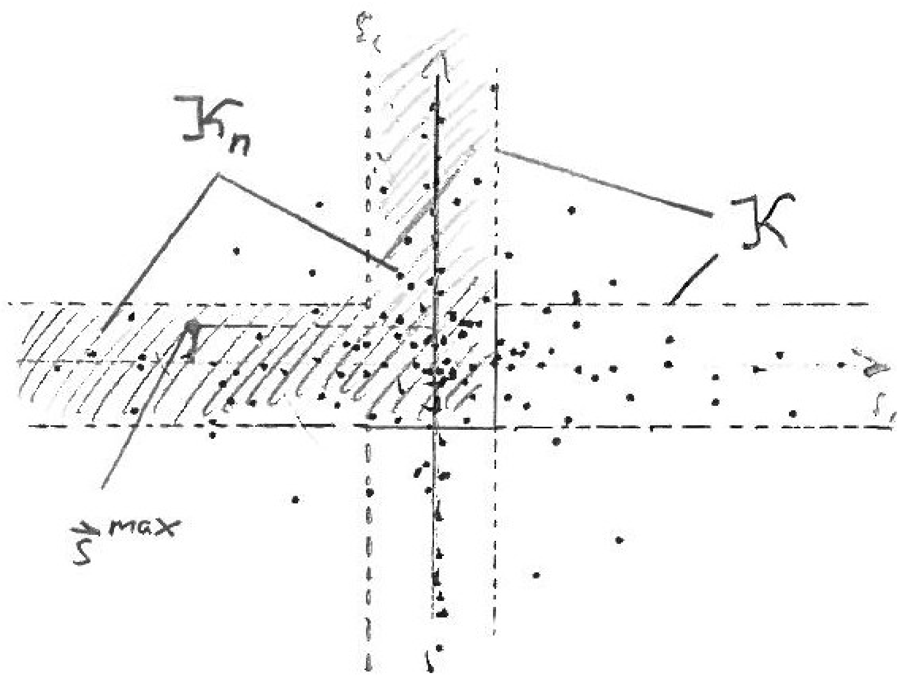
$\mathcal{O}(H)$

Expectation Truncation

ET allows to optimize prior parameters

Puertas et al., *NIPS* 2010;
Henniges et al., *LVA/ICA* 2010;
Lücke, Eggert, JMLR 2010;

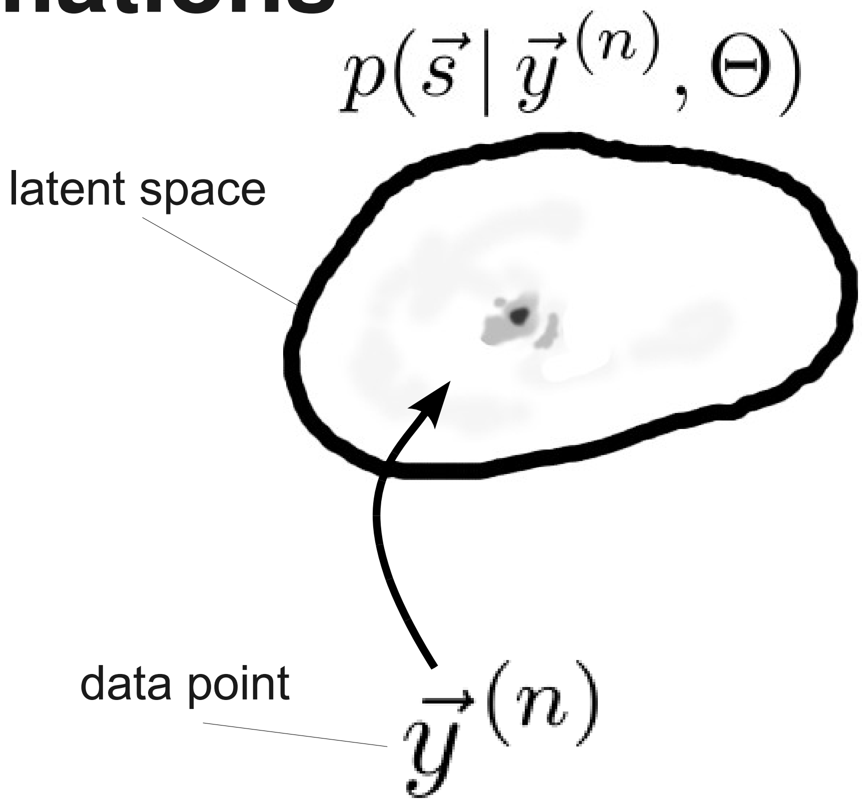
Example Sparse Coding



$$\vec{s}^{\,max} = \mathrm{argmax}_{\vec{s}} \{ p(\vec{s} \,|\, \vec{y}^{(n)}, \Theta) \}$$

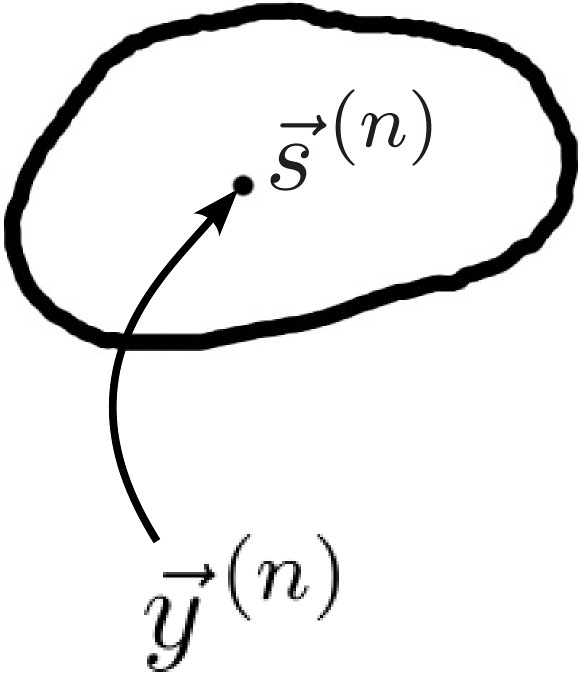
… not further elaborated.

Relation to Other Approximations

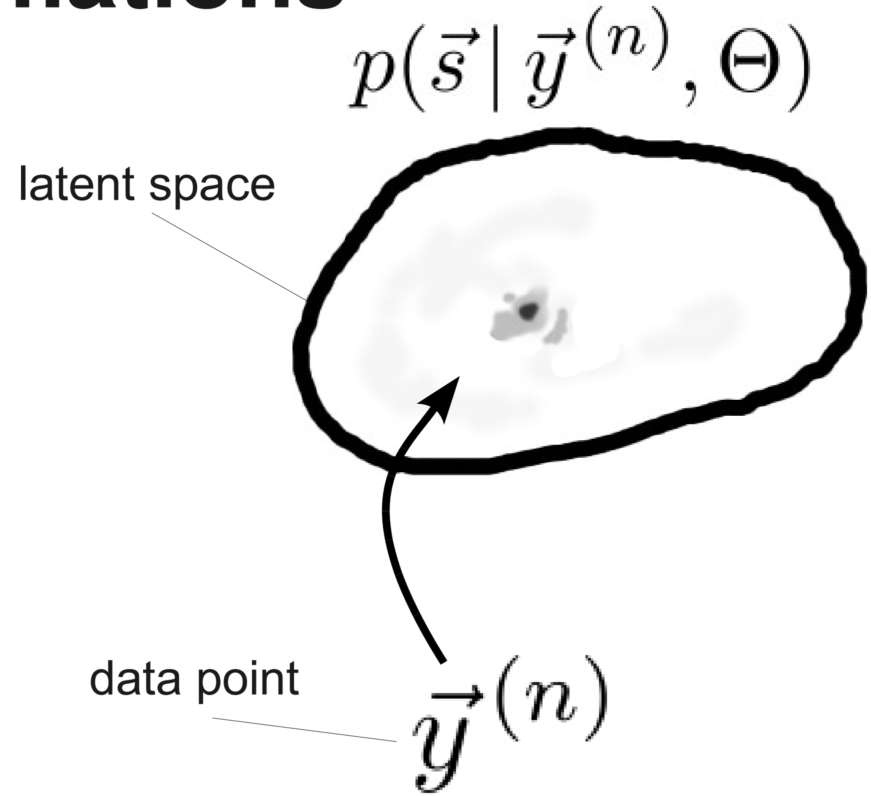


optimal case

Relation to Other Approximations

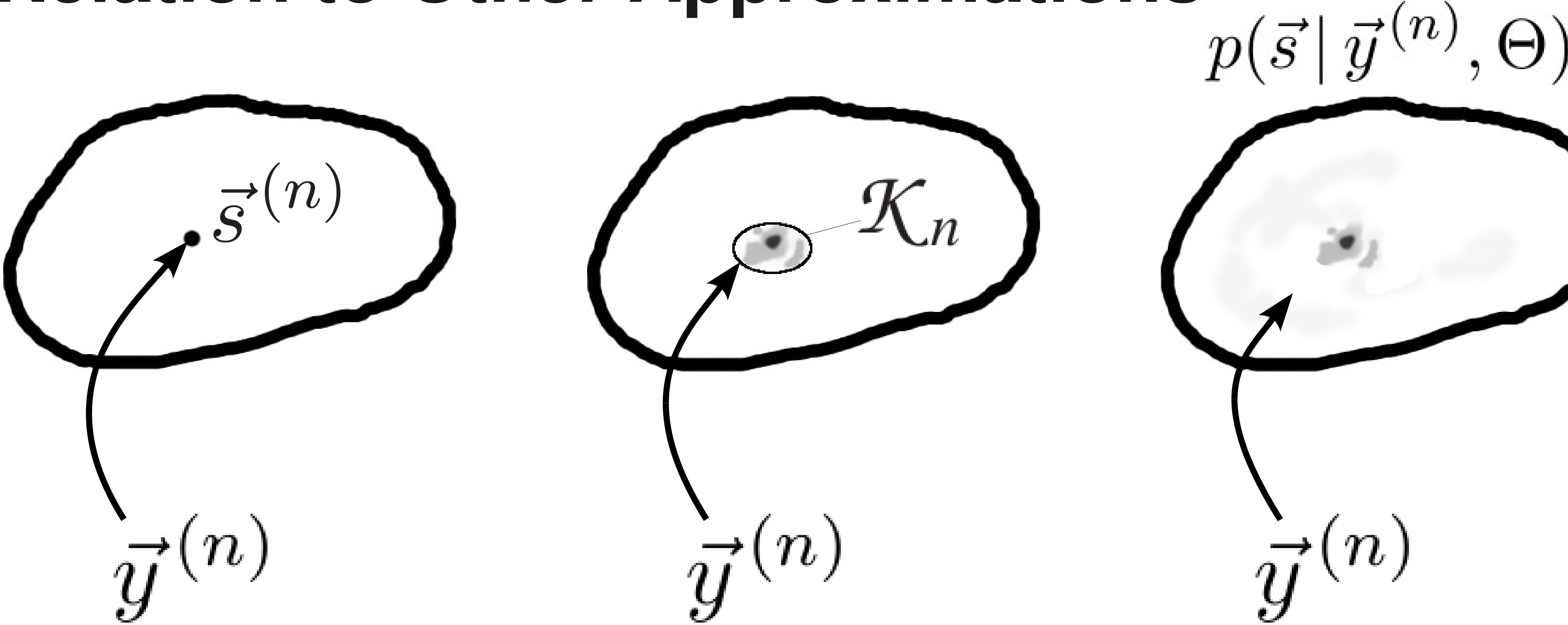


deterministic



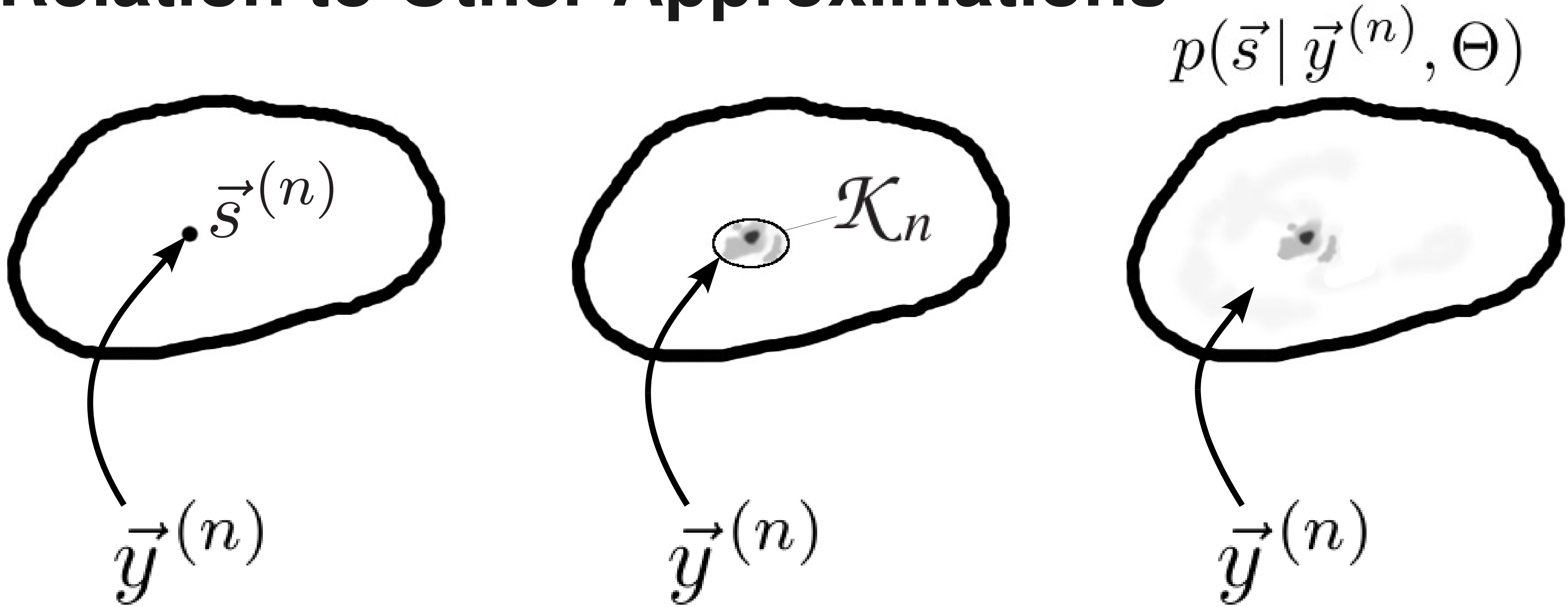
optimal case

Relation to Other Approximations



deterministic **ET** **optimal case**

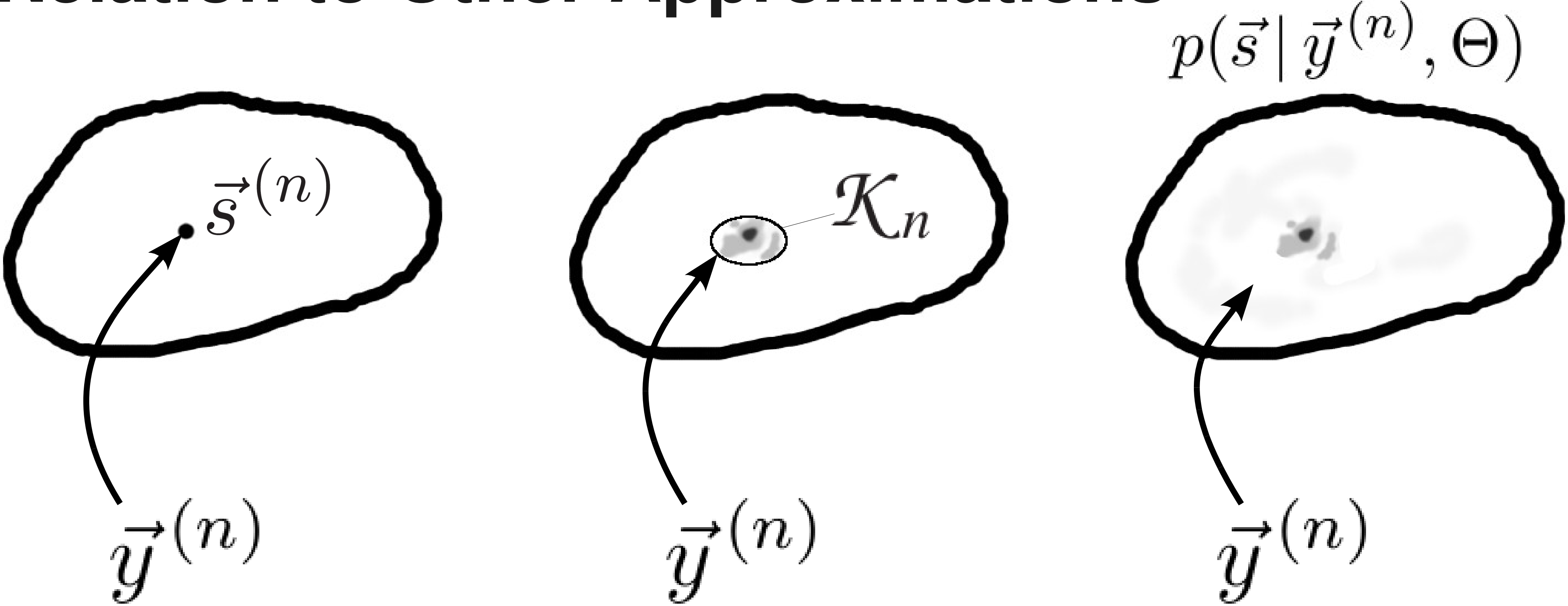
Relation to Other Approximations



Exact: $$q_n(\vec{s};\Theta) \;=\; p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)$$

MAP: $$q_n(\vec{s};\Theta) \;=\; \delta(\vec{s}-\vec{s}^{\,\max})$$

Relation to Other Approximations



Exact: $$q_n(\vec{s};\Theta) = p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta)$$

ET: $$q_n(\vec{s};\Theta) = \frac{1}{A}\,p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta)\,\delta(\vec{s}\in\mathcal{K}_n)$$

MAP: $$q_n(\vec{s};\Theta) = \delta(\vec{s}-\vec{s}^{\,\mathrm{max}})$$

Relation to Other Approximations

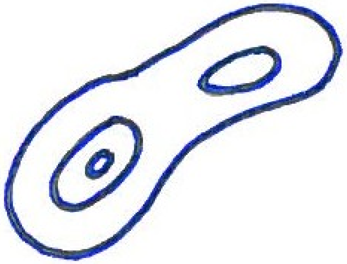
exact: $$q_n(\vec{s};\Theta) \;=\; p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta)$$

MAP: $$q_n(\vec{s};\Theta) \;=\; \delta(\vec{s}-\vec{s}^{\,\mathrm{max}})$$

Laplace: $$q_n(\vec{s};\,\Theta) = \mathcal{N}(\vec{s};\,\vec{s}^{\,\mathrm{max}},\Sigma)$$

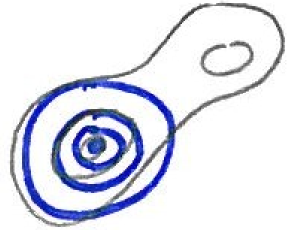
factored: $$q_n(\vec{s};\,\Theta) = \prod_h q^{(n)}_{h,\vec{\lambda}_n}(s_h;\Theta)$$

truncated: $$q_n(\vec{s};\Theta) \;=\; \frac{1}{A}\,p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta)\,\delta(\vec{s}\in\mathcal{K}_n)$$



exact

$$q_n(\vec{s};\Theta) \;=\; p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)$$

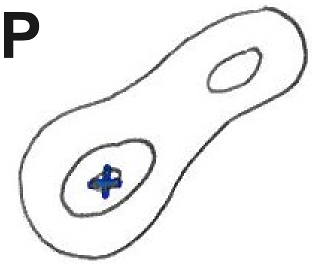


factored (mean-field)

$$q_n(\vec{s};\,\Theta) \;=\; \prod_h q^{(n)}_{h,\vec{\lambda}_n}(s_h;\Theta)$$

minimize $\mathrm{KL}(q,p)$

MAP

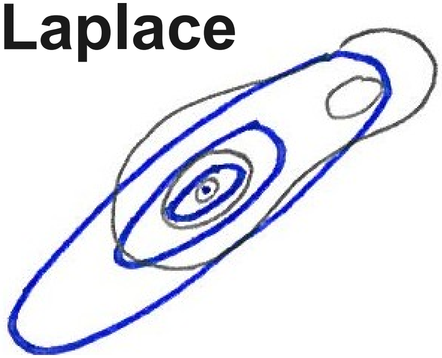


$$q_n(\vec{s};\Theta) \;=\; \delta(\vec{s}-\vec{s}^{\max})$$

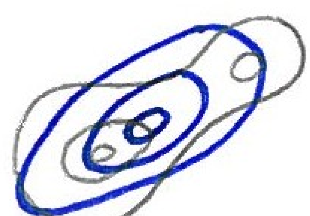
truncated (ET)

$$q_n(\vec{s};\Theta) \;=\; \tfrac{1}{A}\,p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)\,\delta(\vec{s}\in\mathcal{K}_n)$$

Laplace



$$q_n(\vec{s};\,\Theta) = \mathcal{N}(\vec{s};\,\vec{s}^{\max},\Sigma)$$



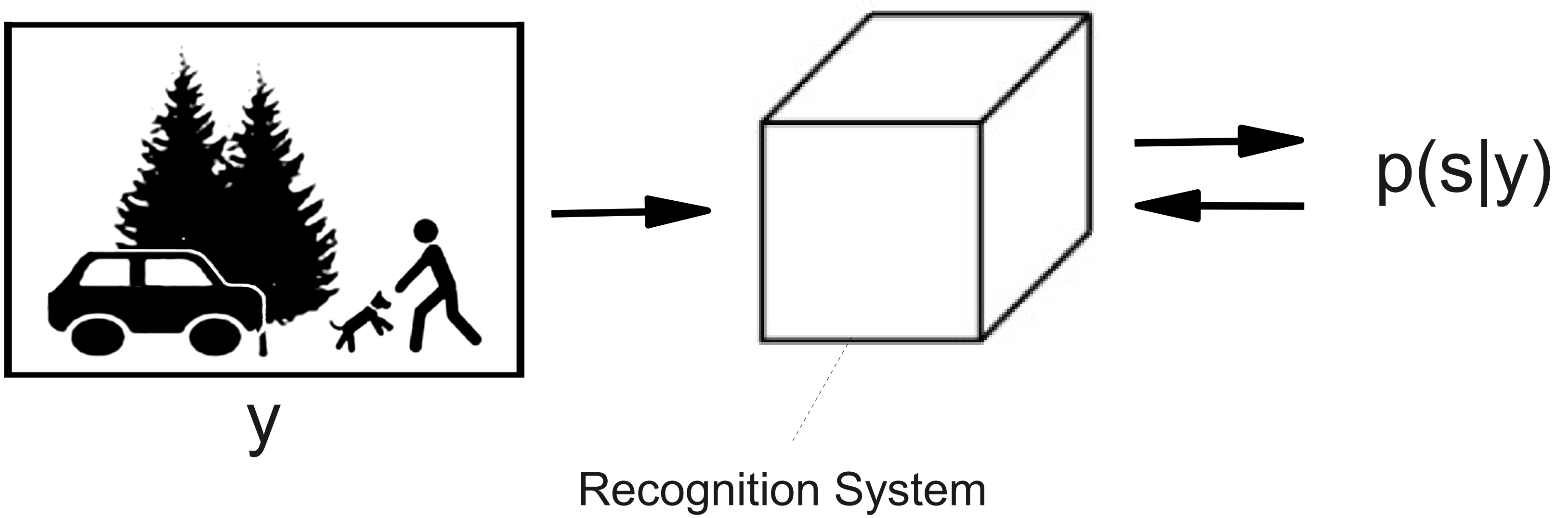
expectation propagation

minimize $\mathrm{KL}(p,q)$

Visualization of variational approaches can differ based on different functional forms of the factor distributions or the selected set K .

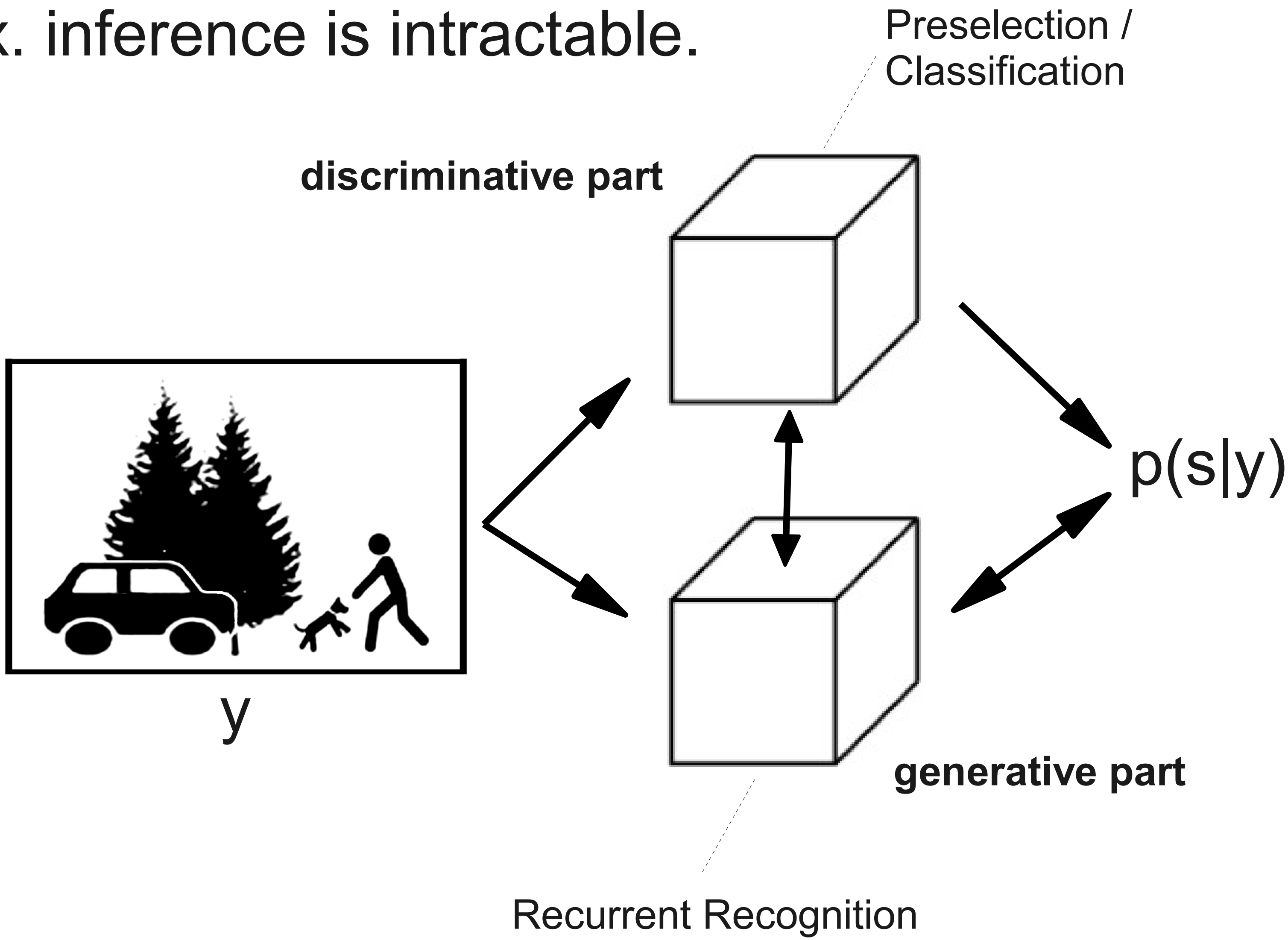
Conclusion

Problem:
Exact inference is intractable.

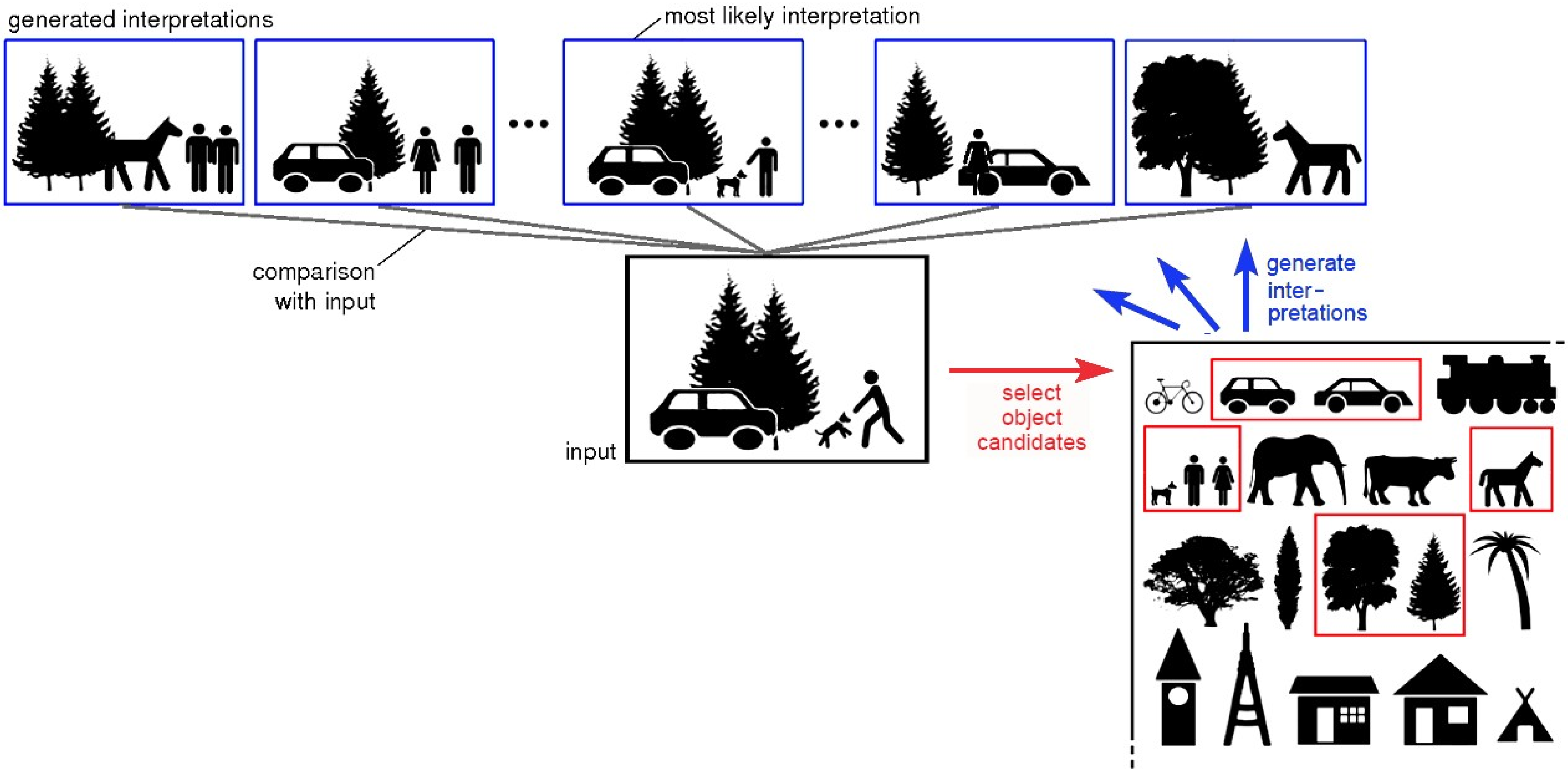


Conclusion

Problem:
Approx. inference is intractable.

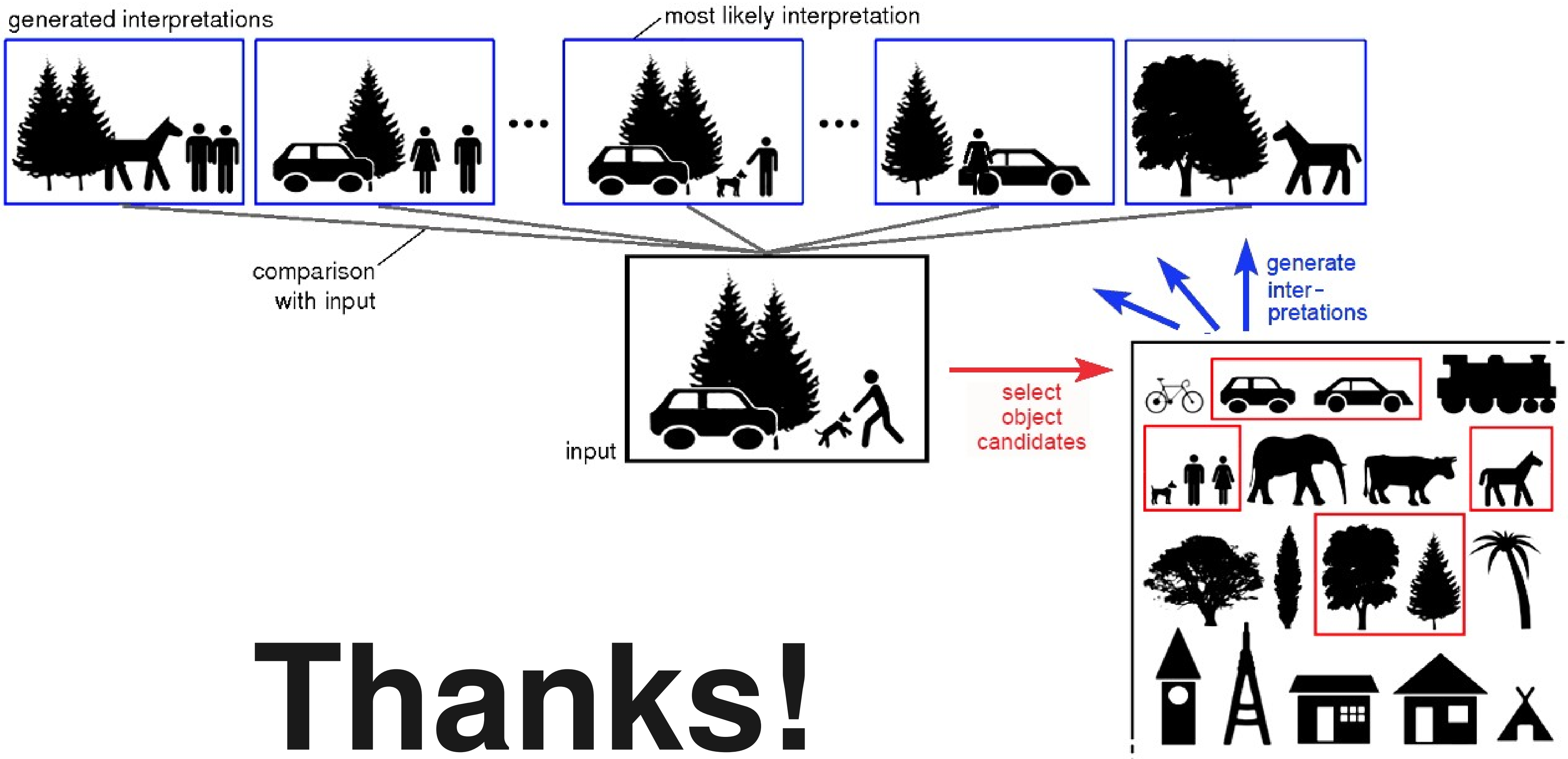


Conclusion



generated interpretations
most likely interpretation
comparison with input
input
select object candidates
generate inter-pretations

Conclusion



Thanks!