“Joint Manifold Model”

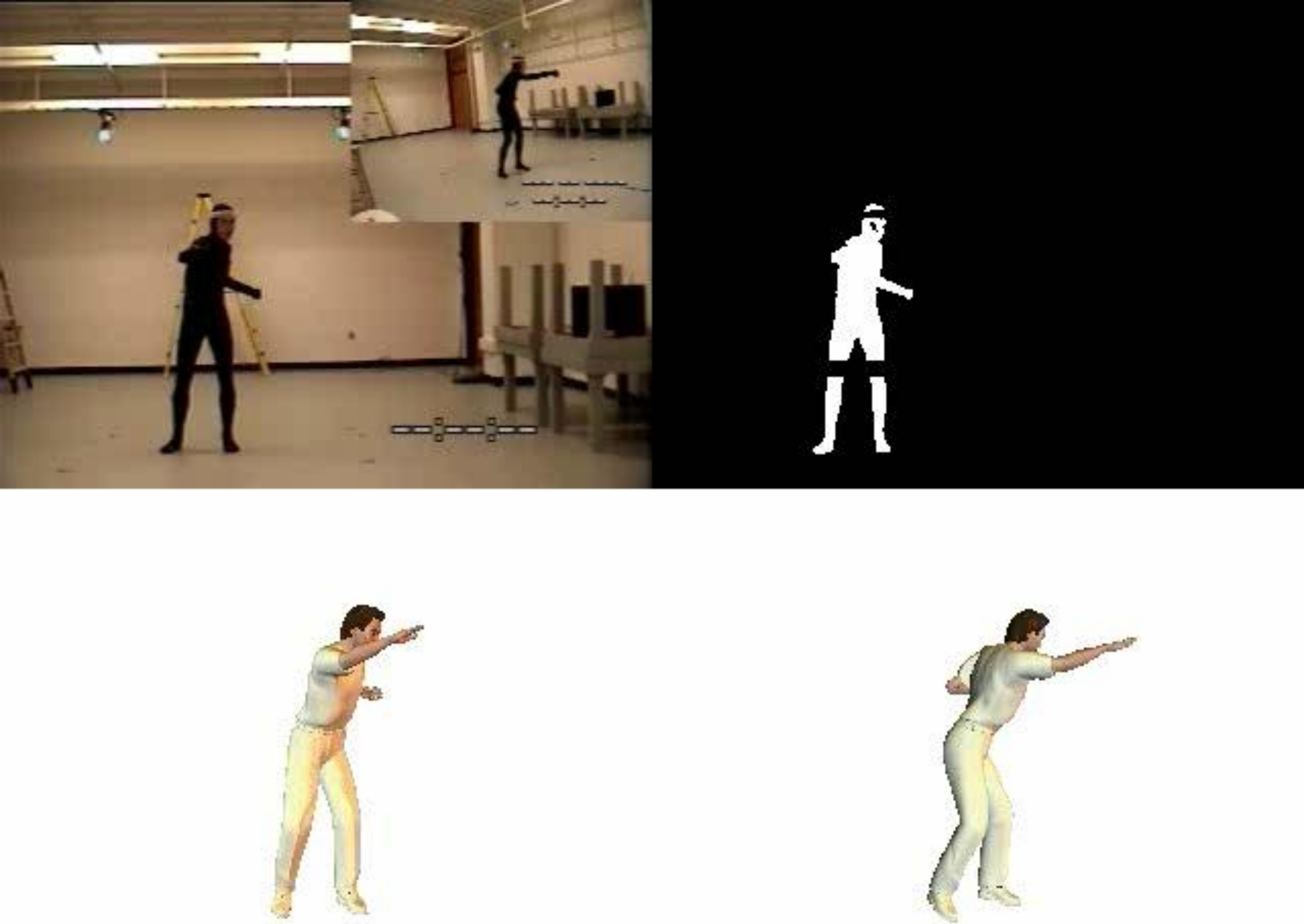
Semi-supervised Multi-valued Regression
...
Recovering real-valued parameters from images

Ramanan Navaratnam	Cambridge University
Andrew Fitzgibbon	Microsoft Research
Roberto Cipolla	Cambridge University

Image I

Pose θ

e.g. Urtasun, Fleet, Hertzmann, Fua; ICCV 2005.



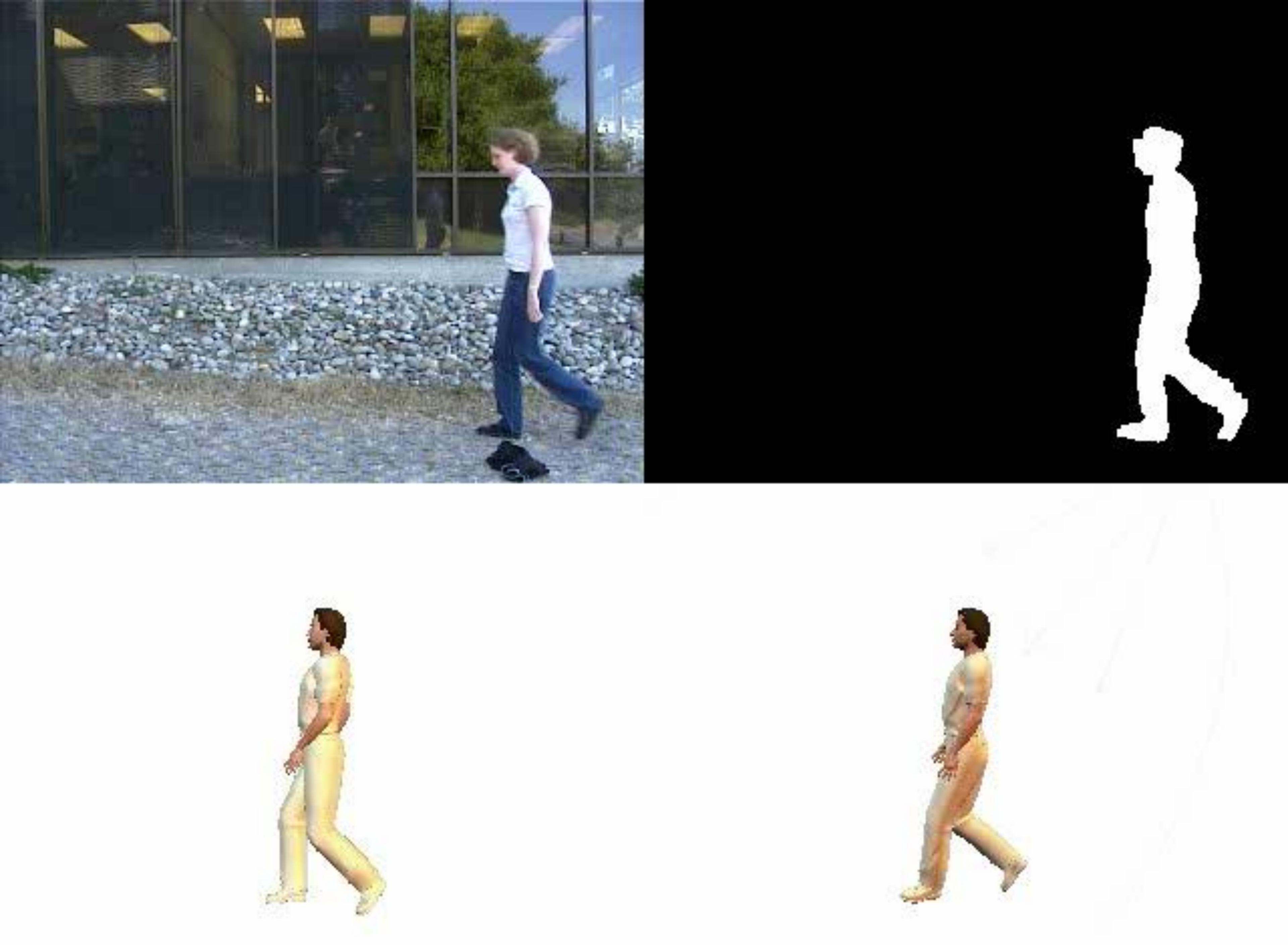
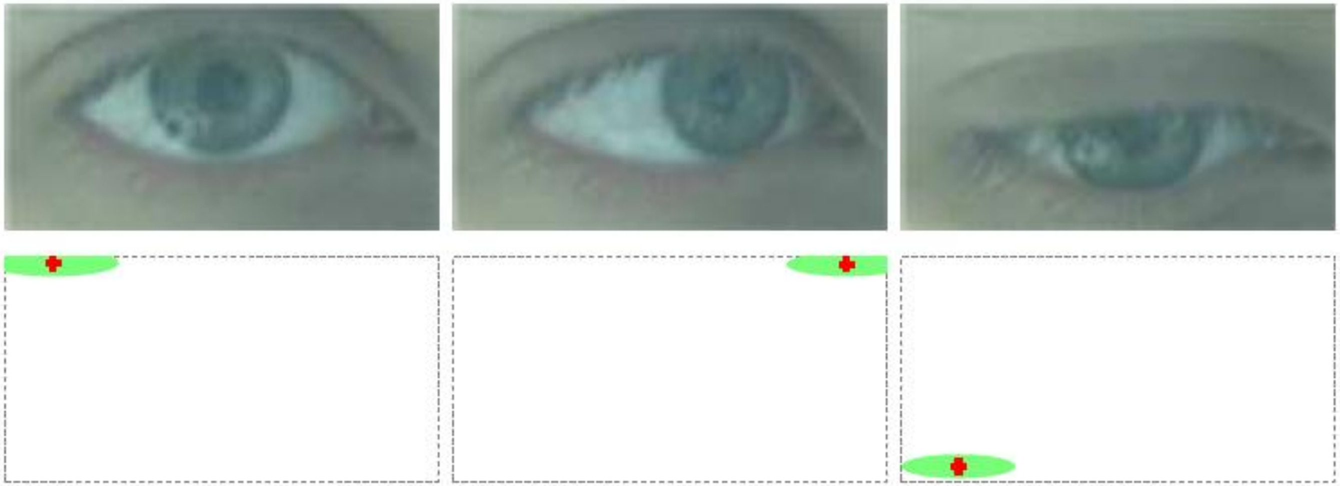


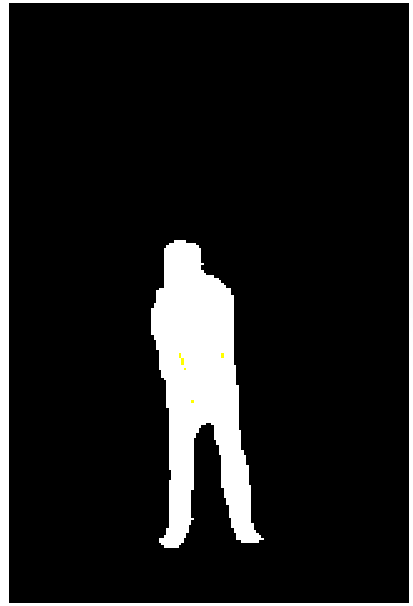
Image I

Position θ



Williams, Blake, Cipolla; CVPR 2006

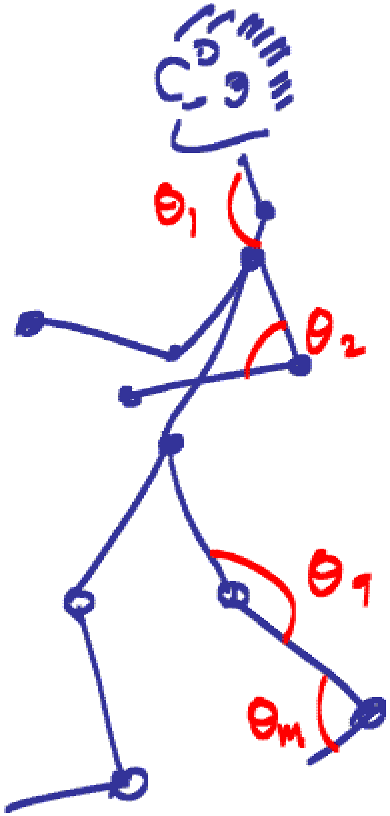
Image I



$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$$

Feature vector $\mathbf{z}$

e.g. Shape contexts on silhouette, $\mathbf{z} \in \mathbb{R}^{40}$

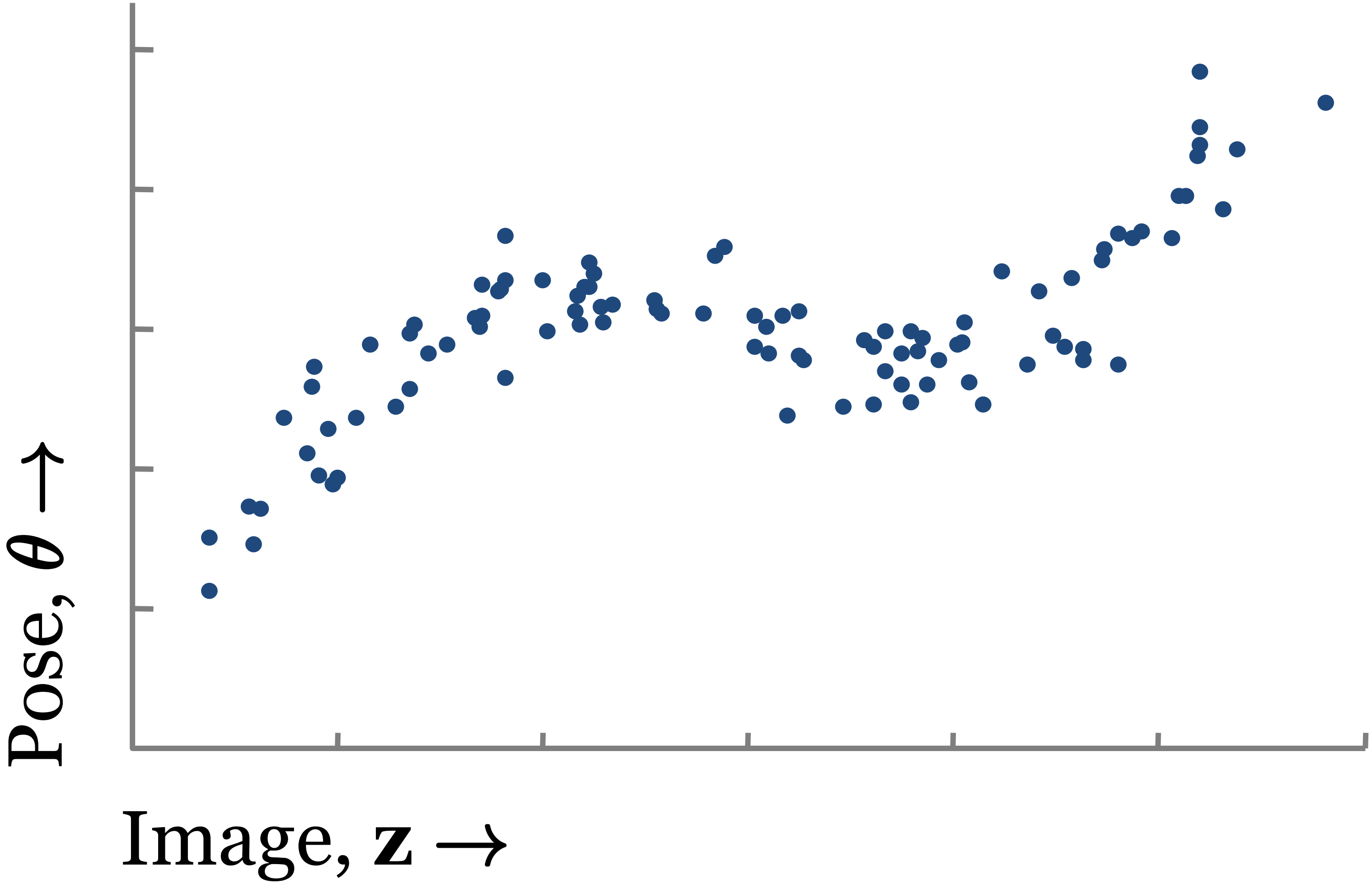


$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{bmatrix}$$

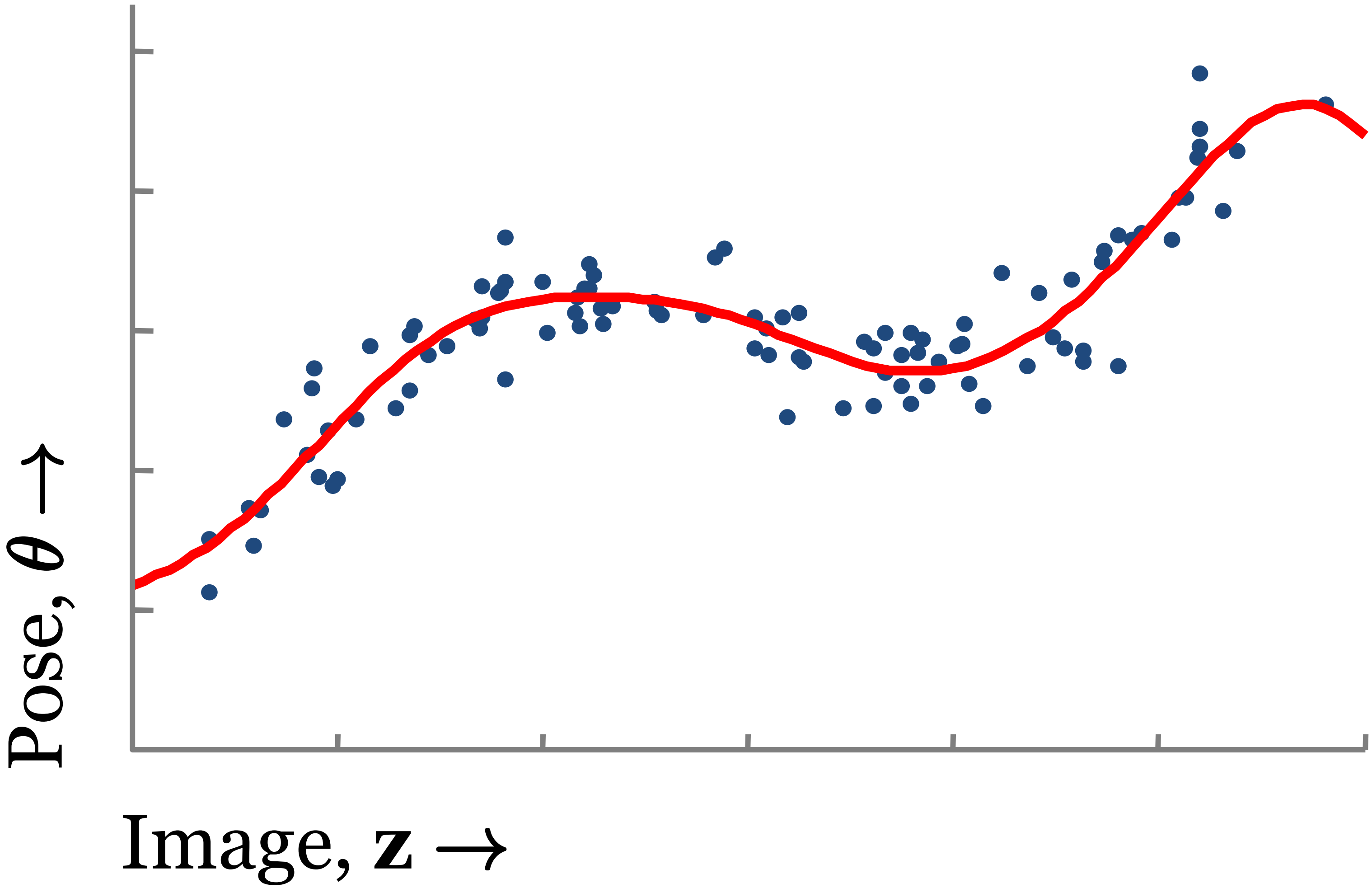
Pose vector $\boldsymbol{\theta}$

e.g. Joint angles $\boldsymbol{\theta} \in \mathbb{R}^{27}$

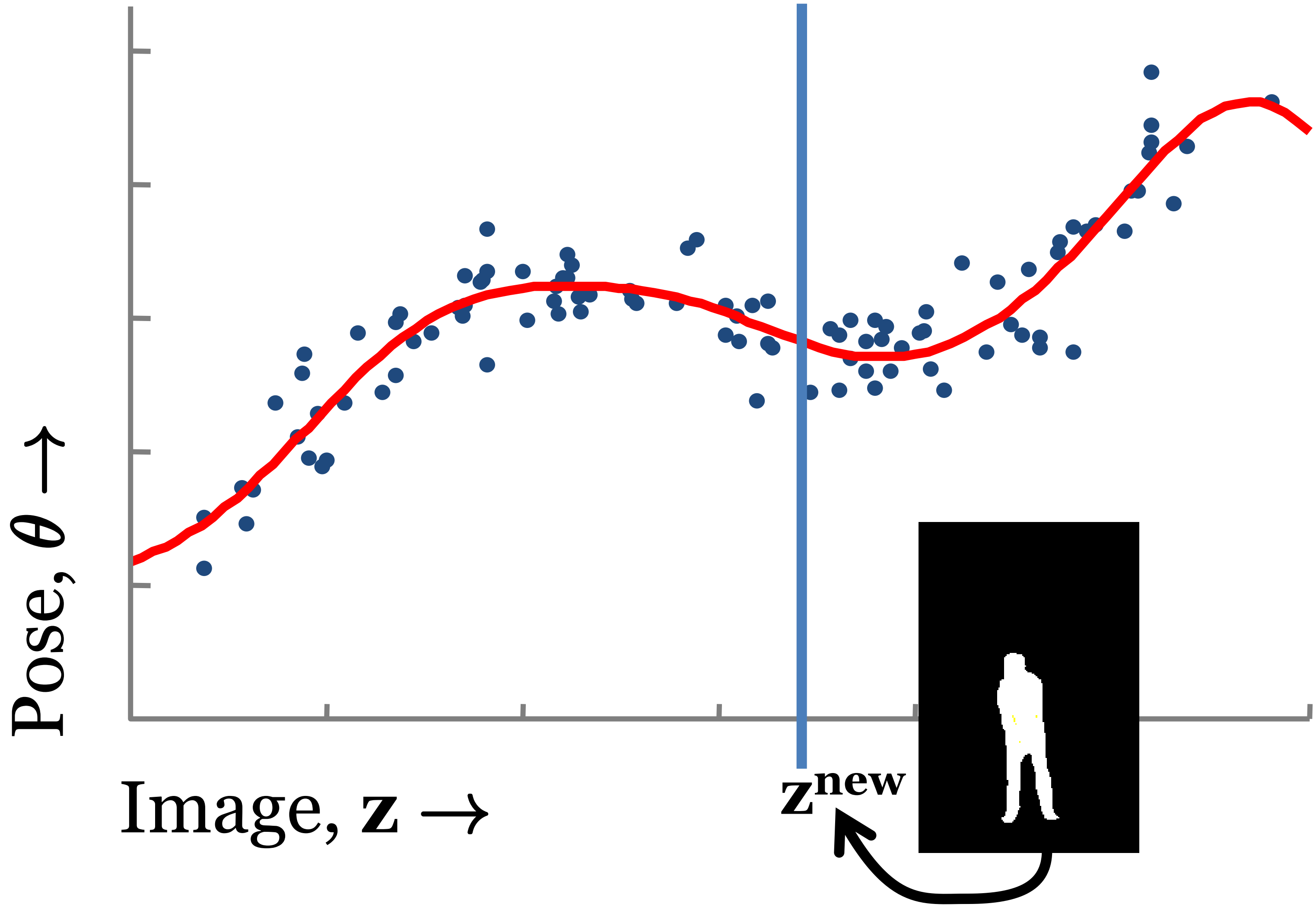
1. Obtain training samples $(\mathbf{z}_1, \boldsymbol{\theta}_1)\dots(\mathbf{z}_N, \boldsymbol{\theta}_N)$



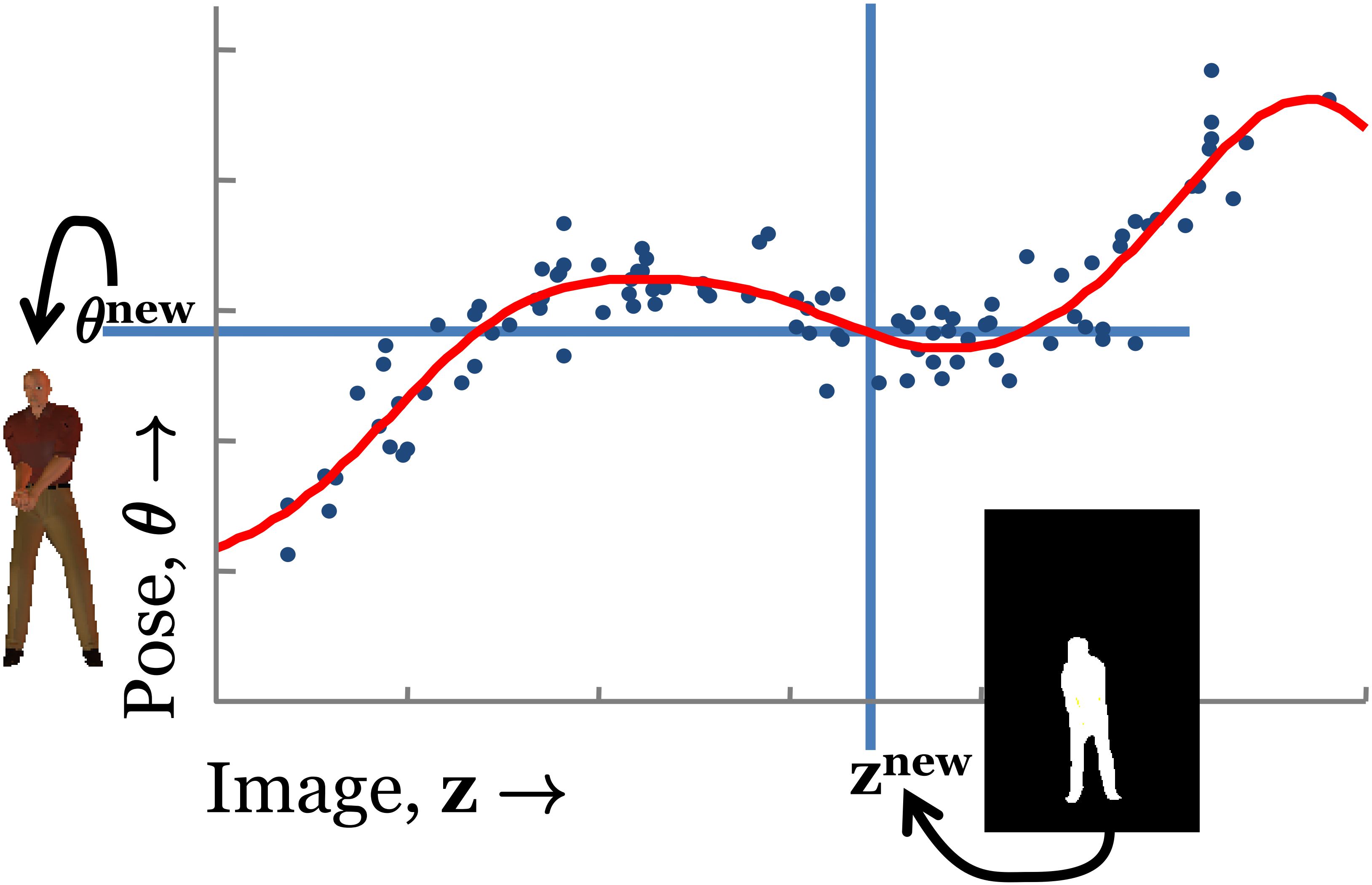
2. Training: Fit function $\boldsymbol{\theta} = f(\mathbf{z})$.



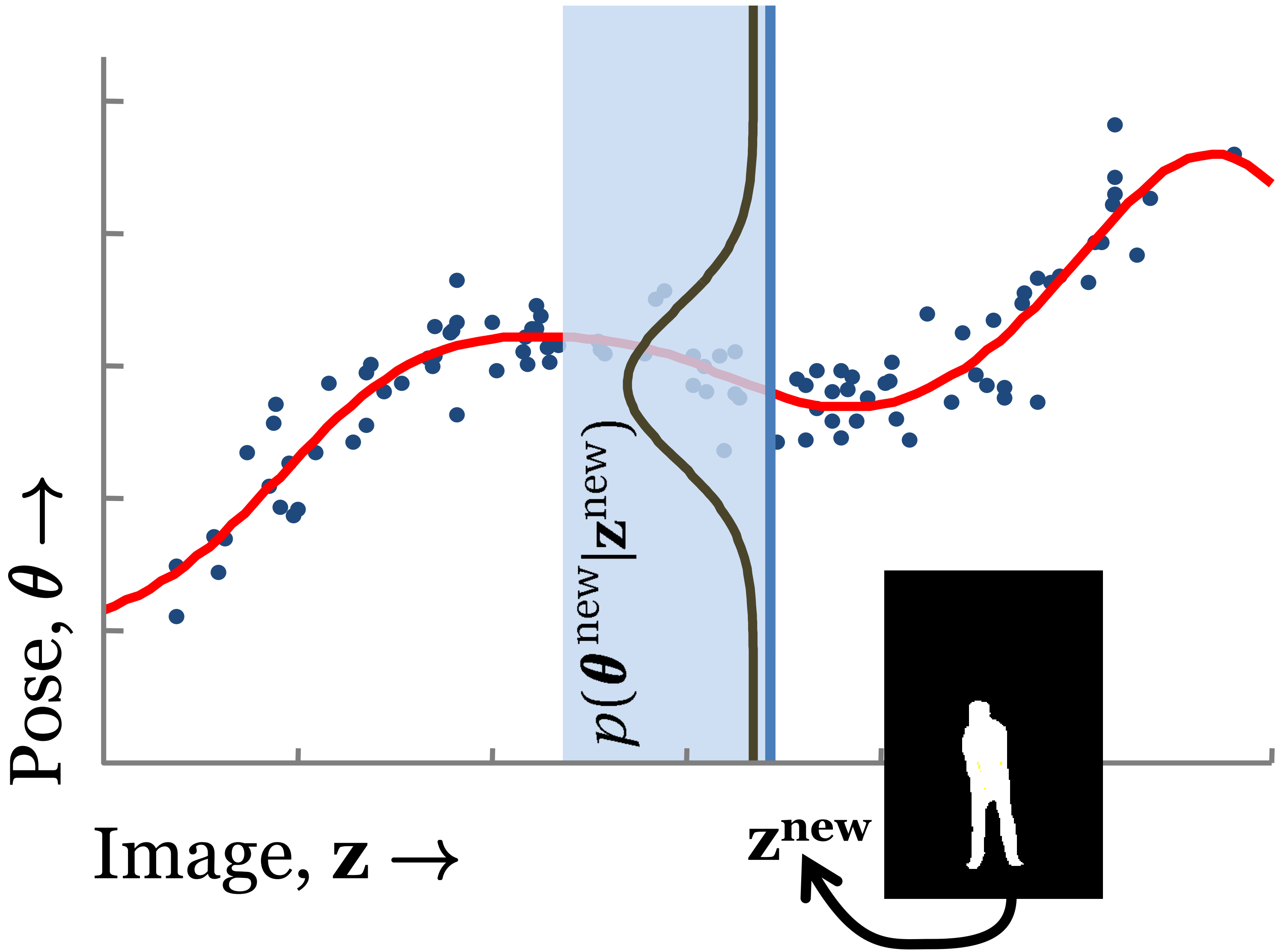
3. Given new image, $\mathbf{z}^{\text{new}}$, compute $\boldsymbol{\theta}^{\text{new}} = f(\mathbf{z}^{\text{new}})$.



3. Given new image, $\mathbf{z}^{\text{new}}$, compute $\boldsymbol{\theta}^{\text{new}} = f(\mathbf{z}^{\text{new}})$.



3. Or, more usefully, compute $p(\boldsymbol{\theta}^{\text{new}}|\mathbf{z}^{\text{new}})$.

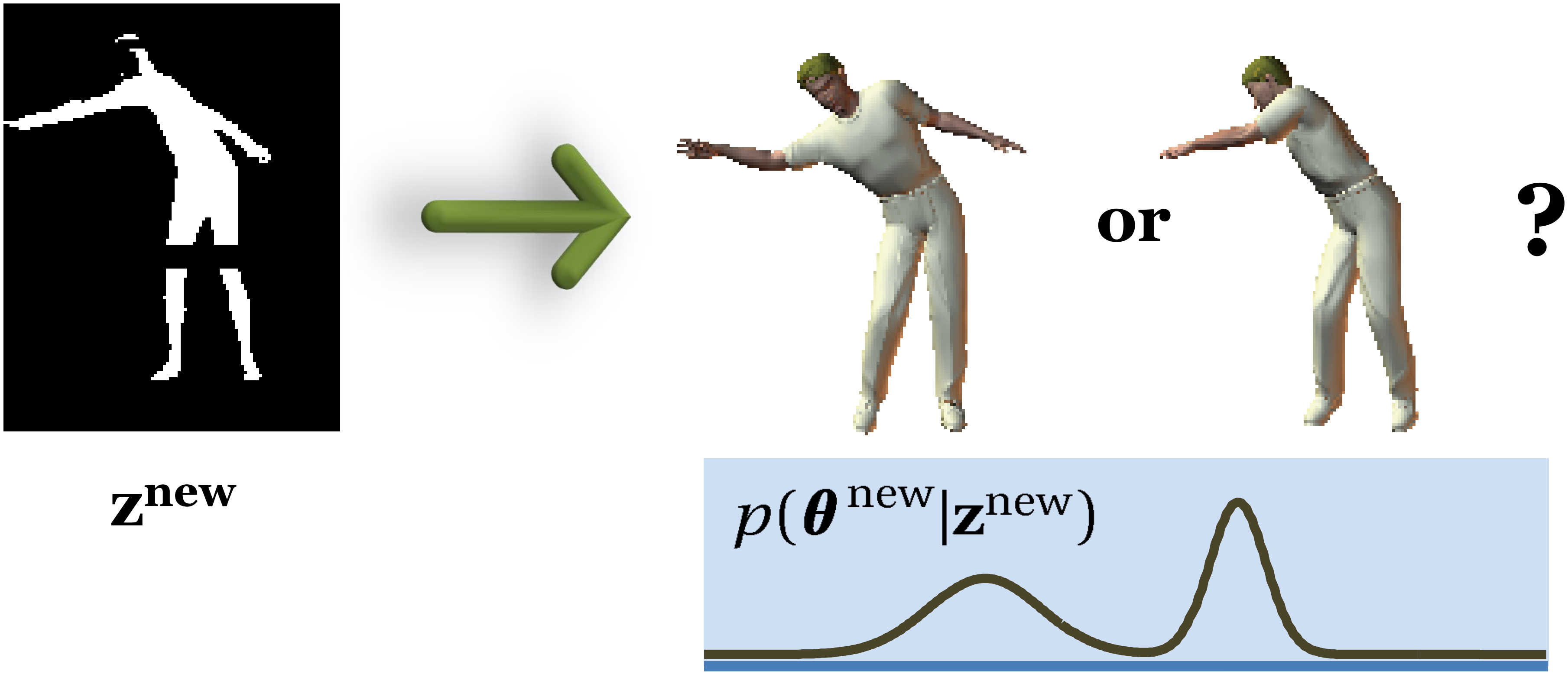


It’ll never work...

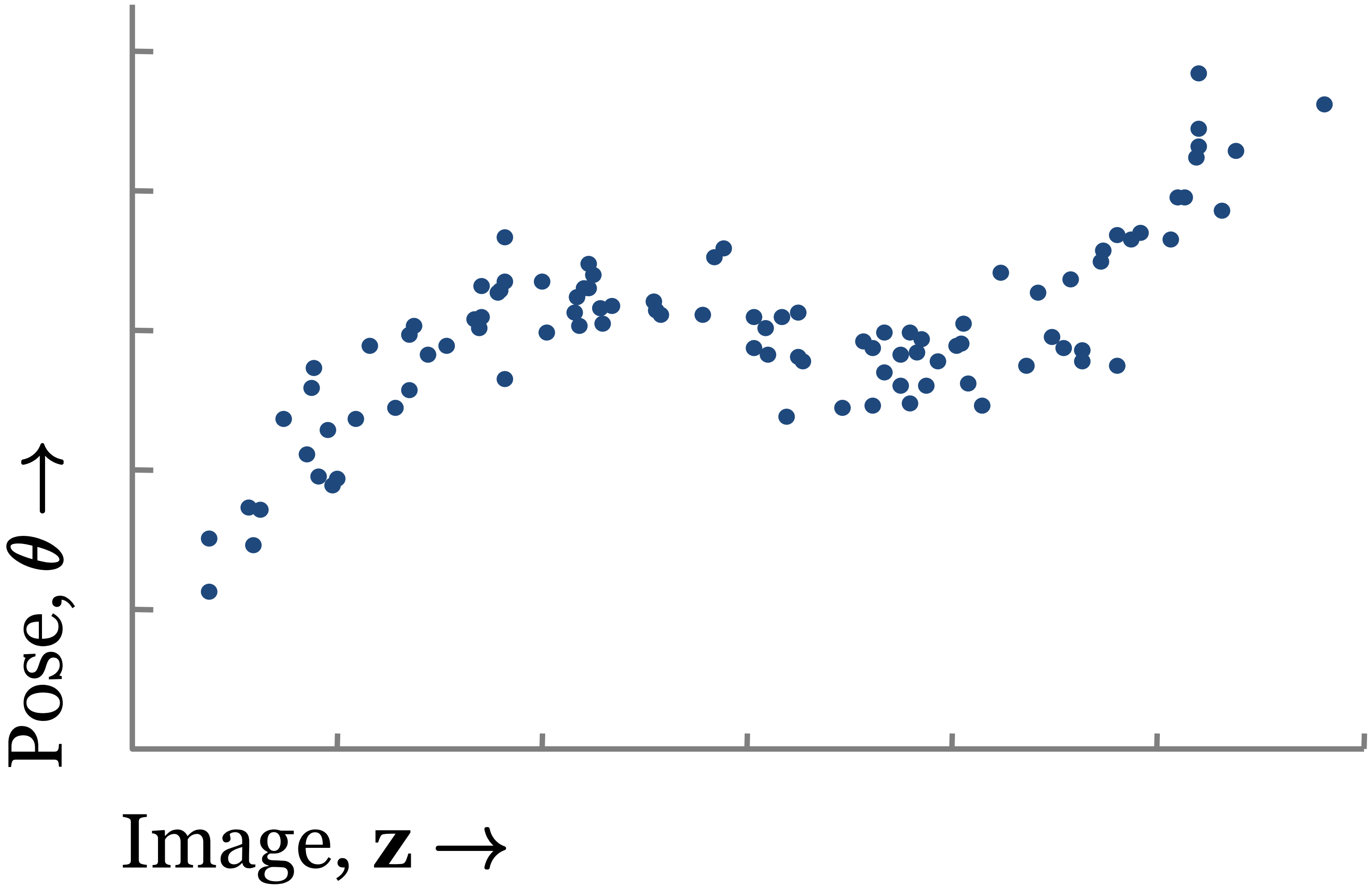
- f is multivalued
- $\mathbf{z}$ and $\boldsymbol{\theta}$ live in high dimensions

Multivalued f:

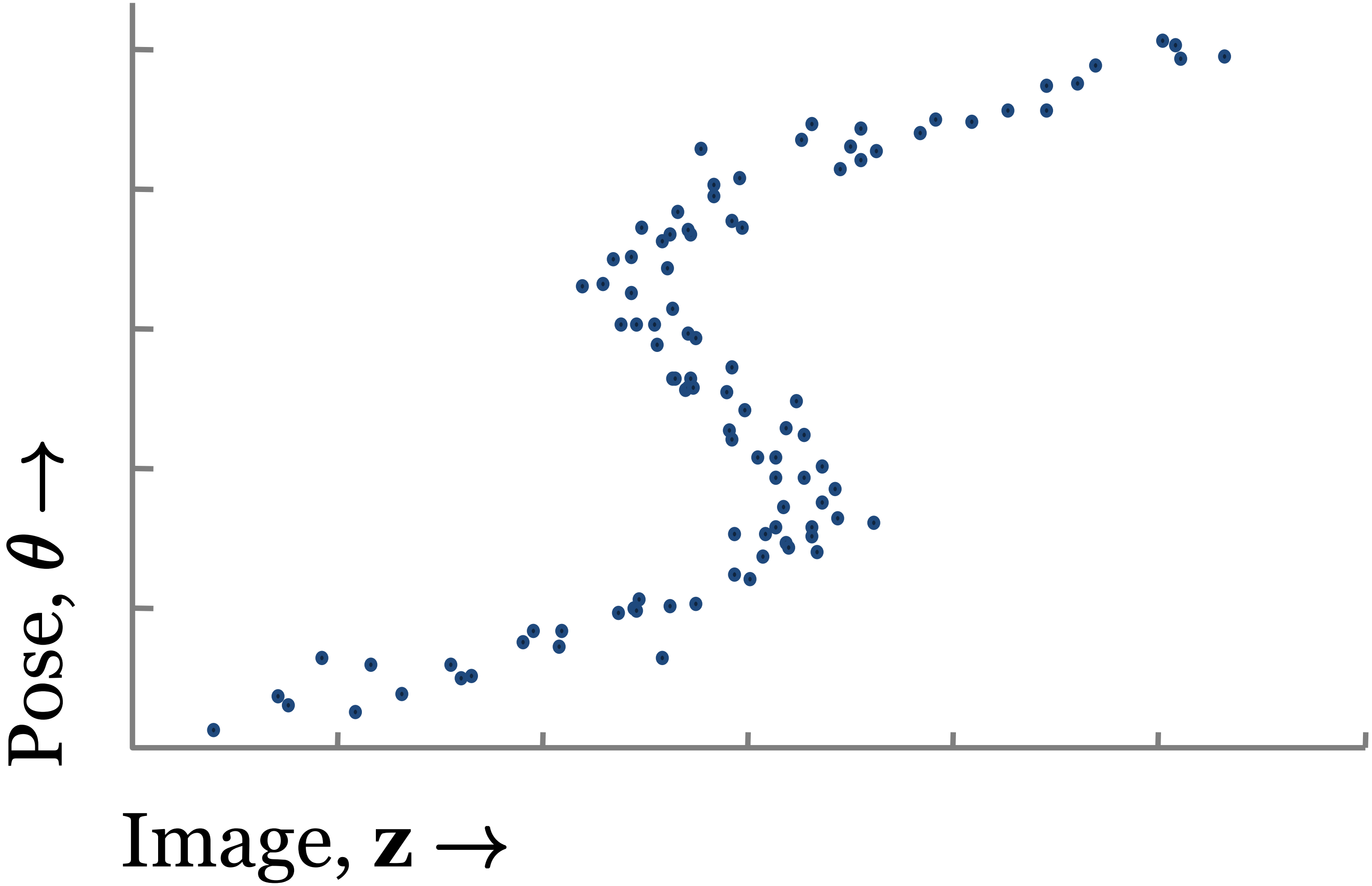
Multivalued f:



Instead of this:

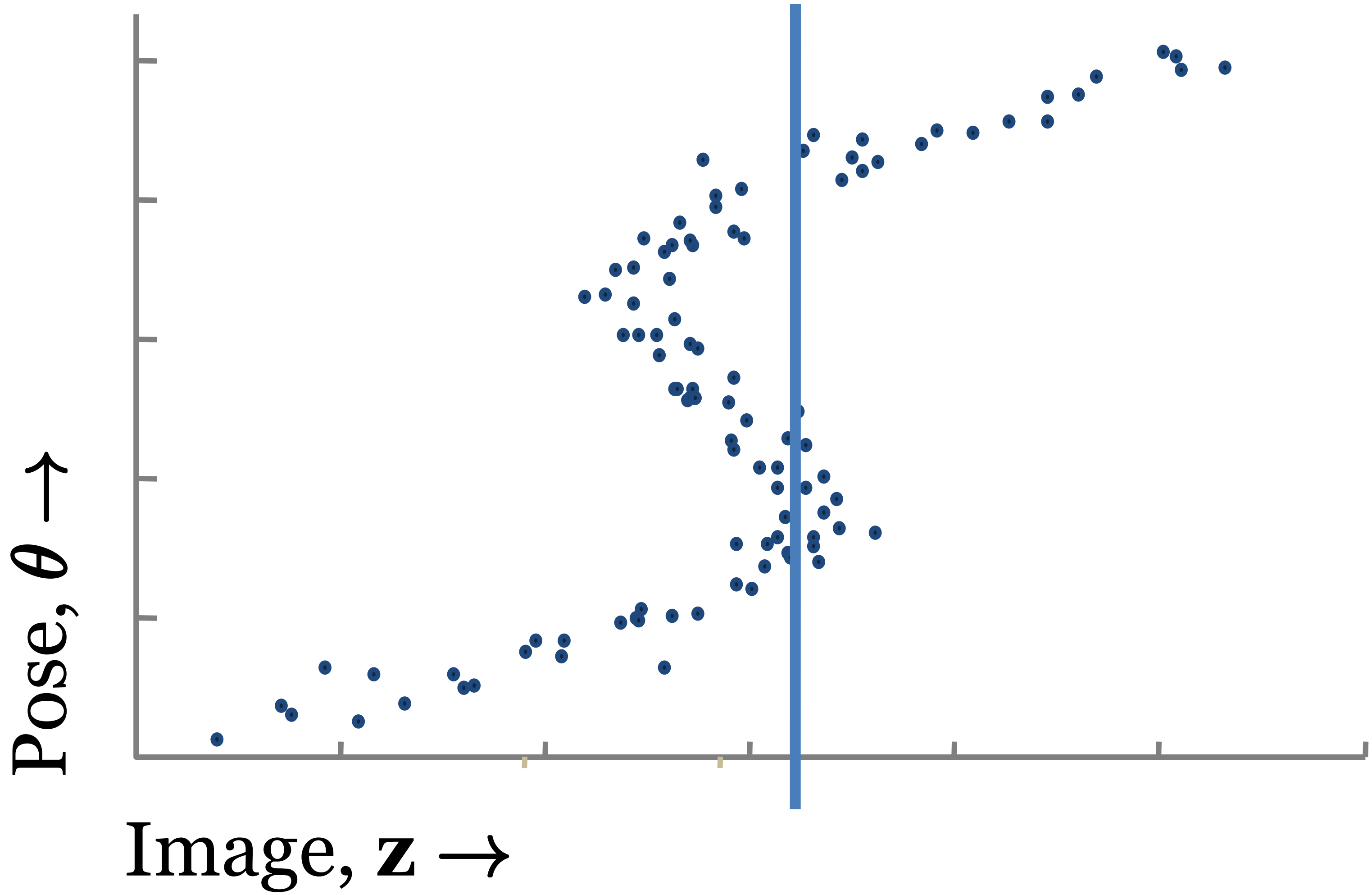


We have this:

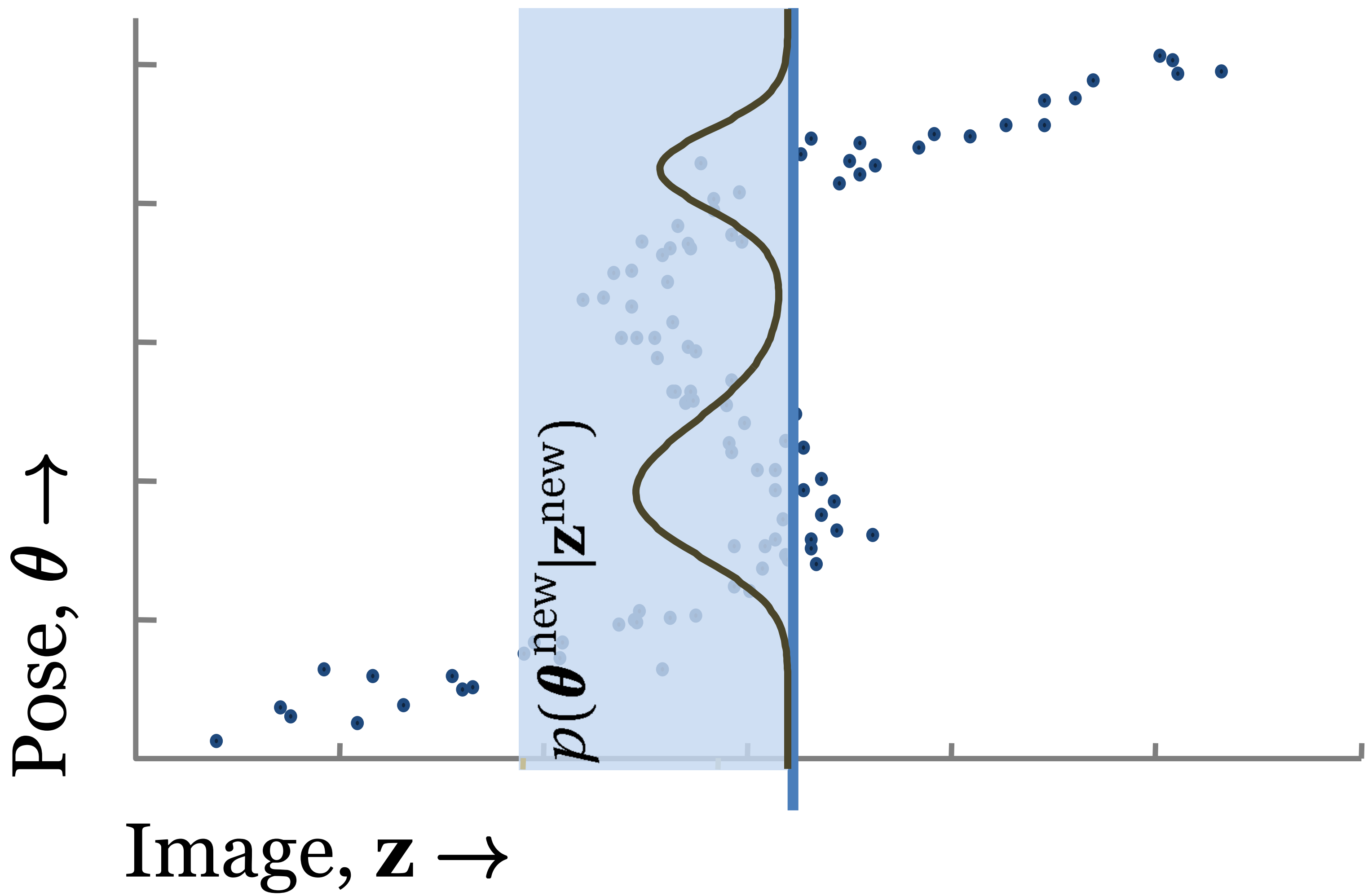


Pose, $\boldsymbol{\theta} \rightarrow$
Image, $\mathbf{z} \rightarrow$

We have this:

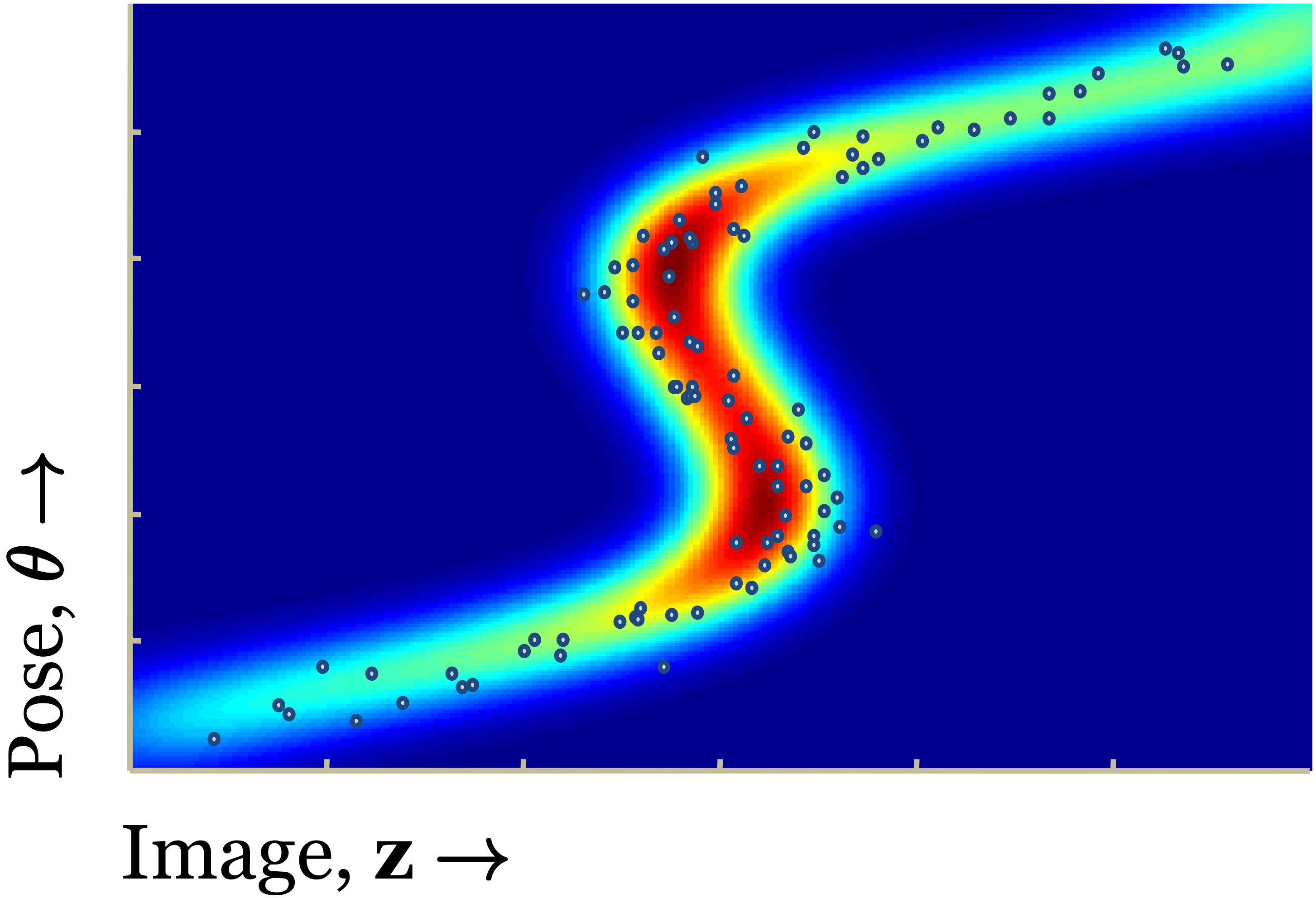

Pose, $\boldsymbol{\theta} \rightarrow$
Image, $\mathbf{z} \rightarrow$

We have this:

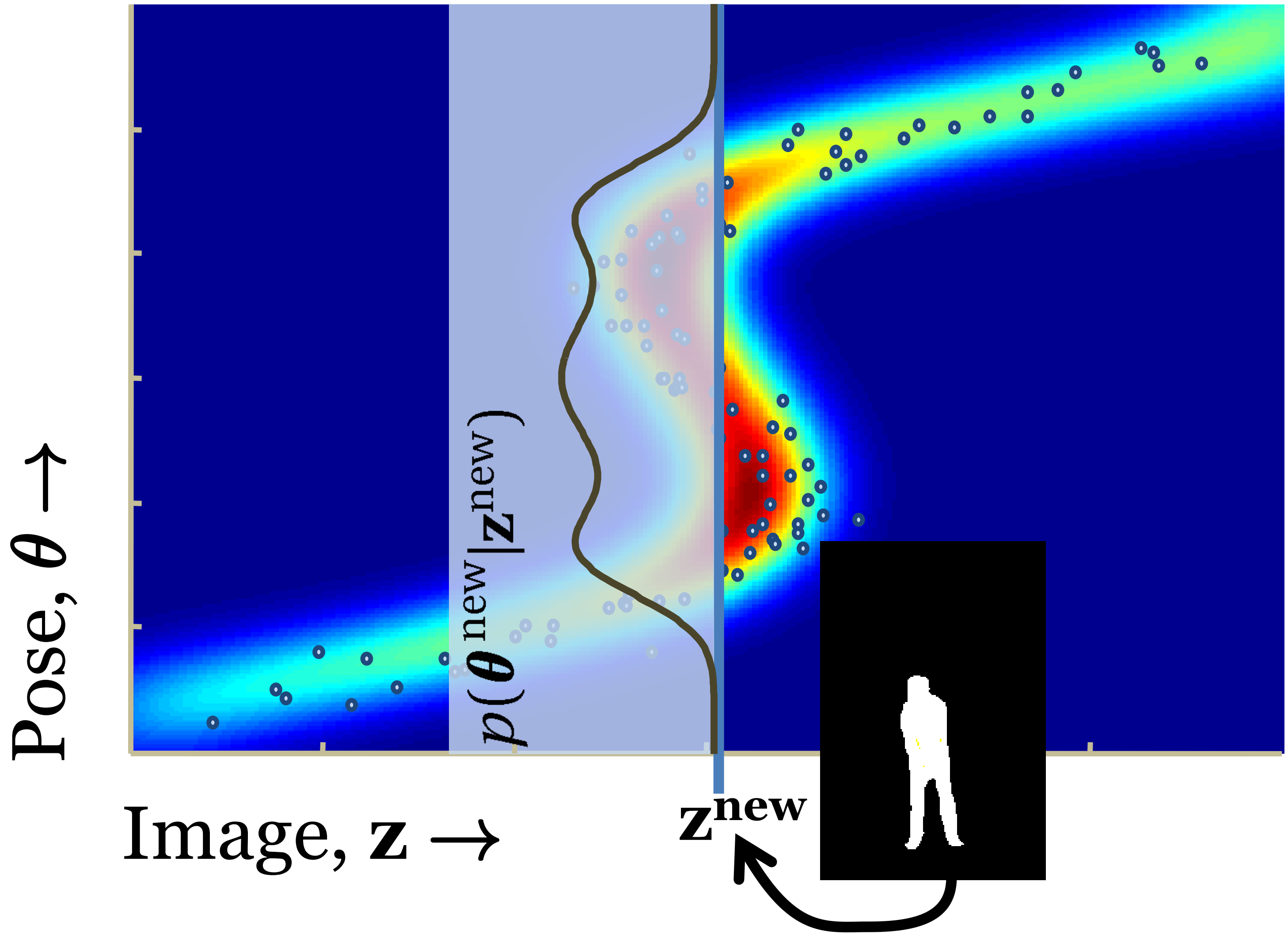


$p(\boldsymbol{\theta}^{\text{new}}|\mathbf{z}^{\text{new}})$
Pose, $\boldsymbol{\theta} \rightarrow$
Image, $\mathbf{z} \rightarrow$

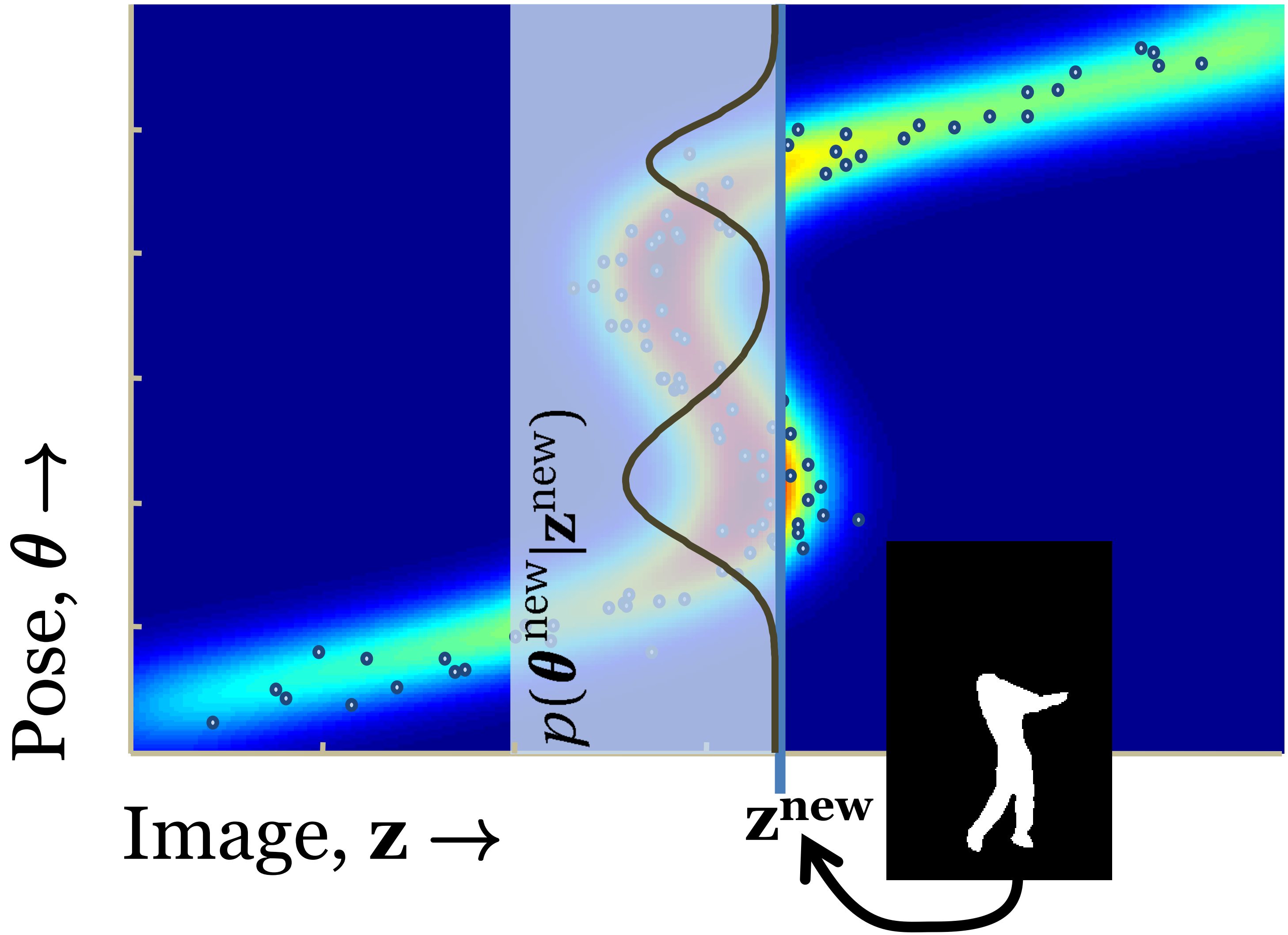
Instead of fitting $p(\boldsymbol{\theta}|\mathbf{z})$, fit $p(\boldsymbol{\theta},\mathbf{z})$, e.g. with GMM, Parzen, or GPLVM.



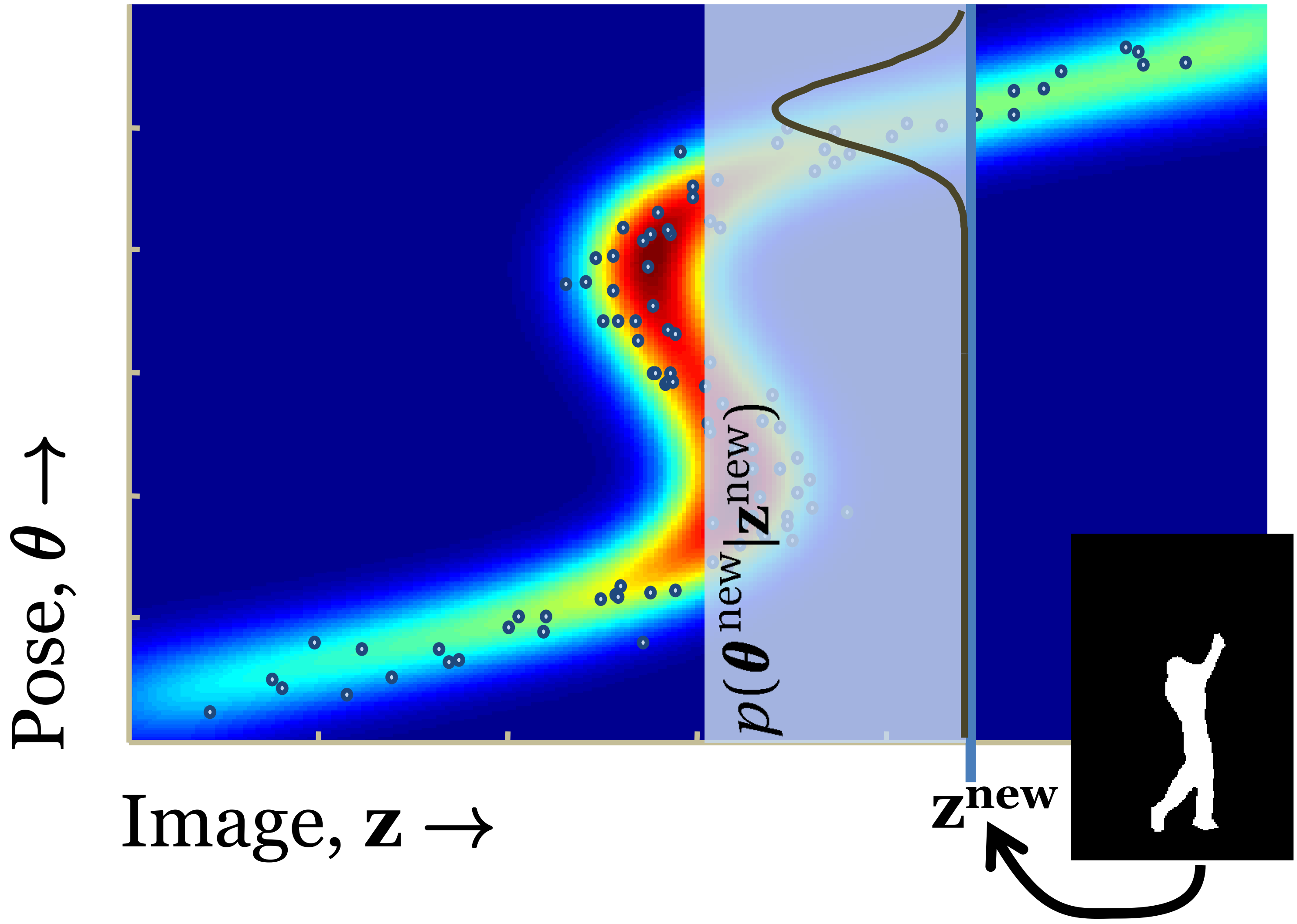
Given new image $\mathbf{z}^{\text{new}}$, conditional $p(\boldsymbol{\theta}|\mathbf{z}^{\text{new}})$ is computed from the joint.

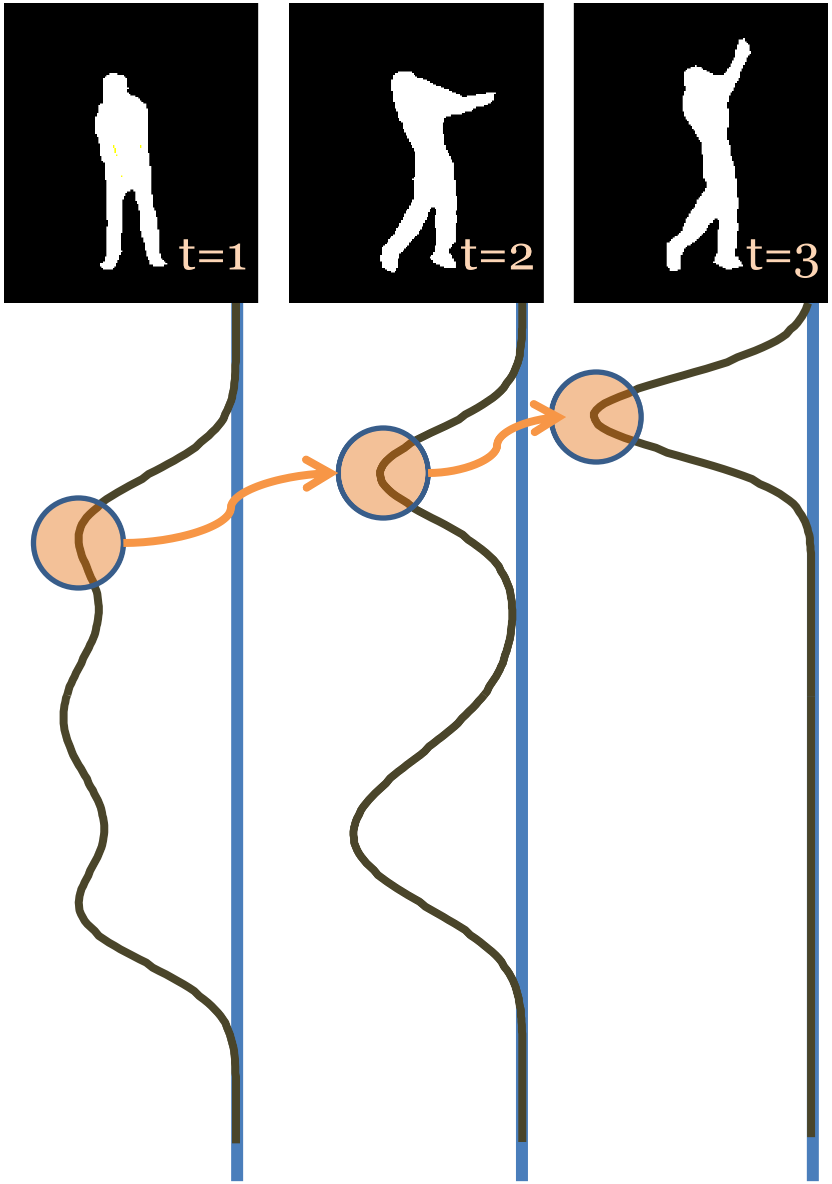


Given new image $\mathbf{z}^{new}$, conditional $p(\boldsymbol{\theta}|\mathbf{z}^{new})$ is computed from the joint.



Given new image $\mathbf{z}^{\text{new}}$, conditional $p(\boldsymbol{\theta}|\mathbf{z}^{\text{new}})$ is computed from the joint.



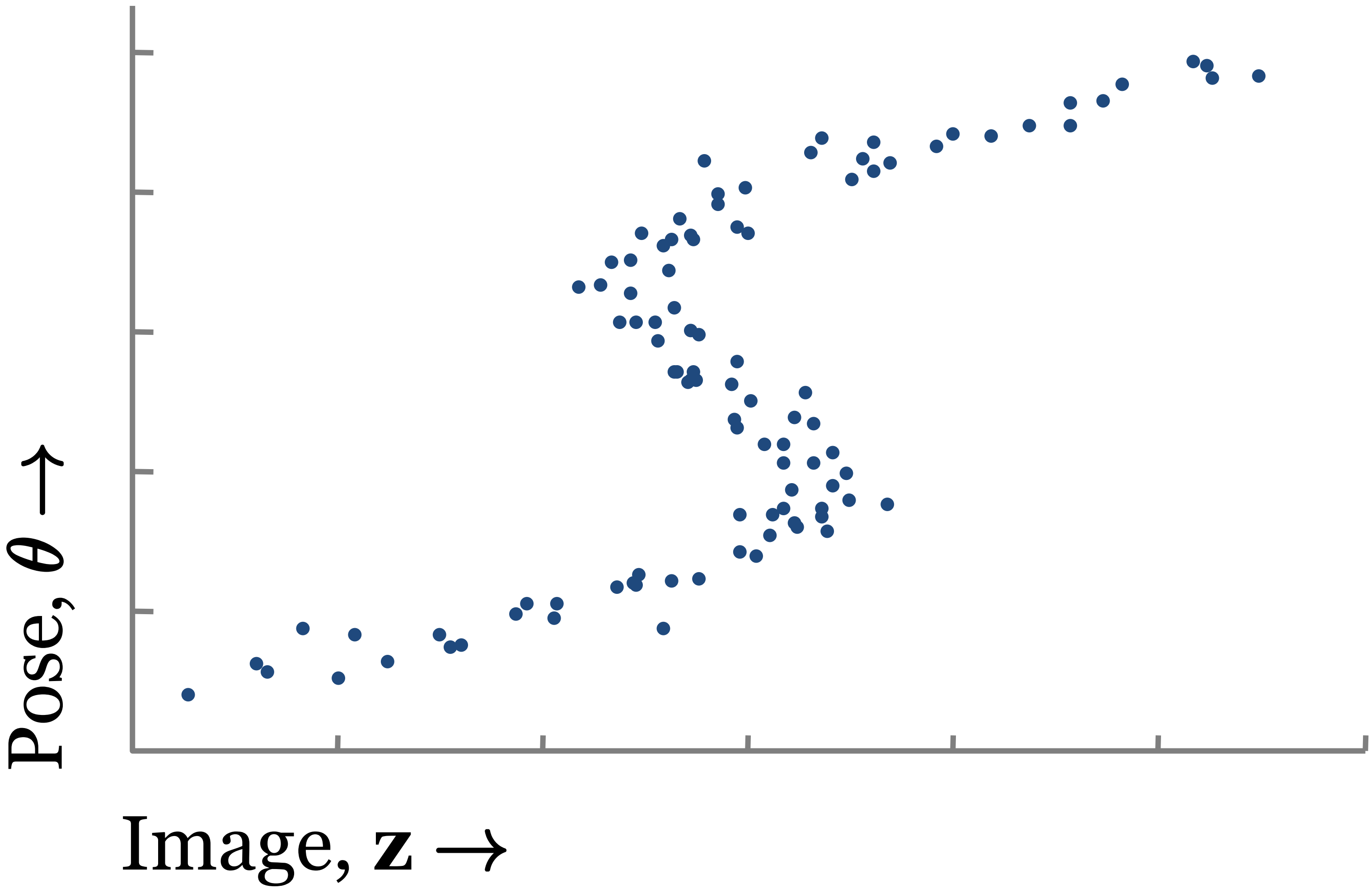


For a video sequence:

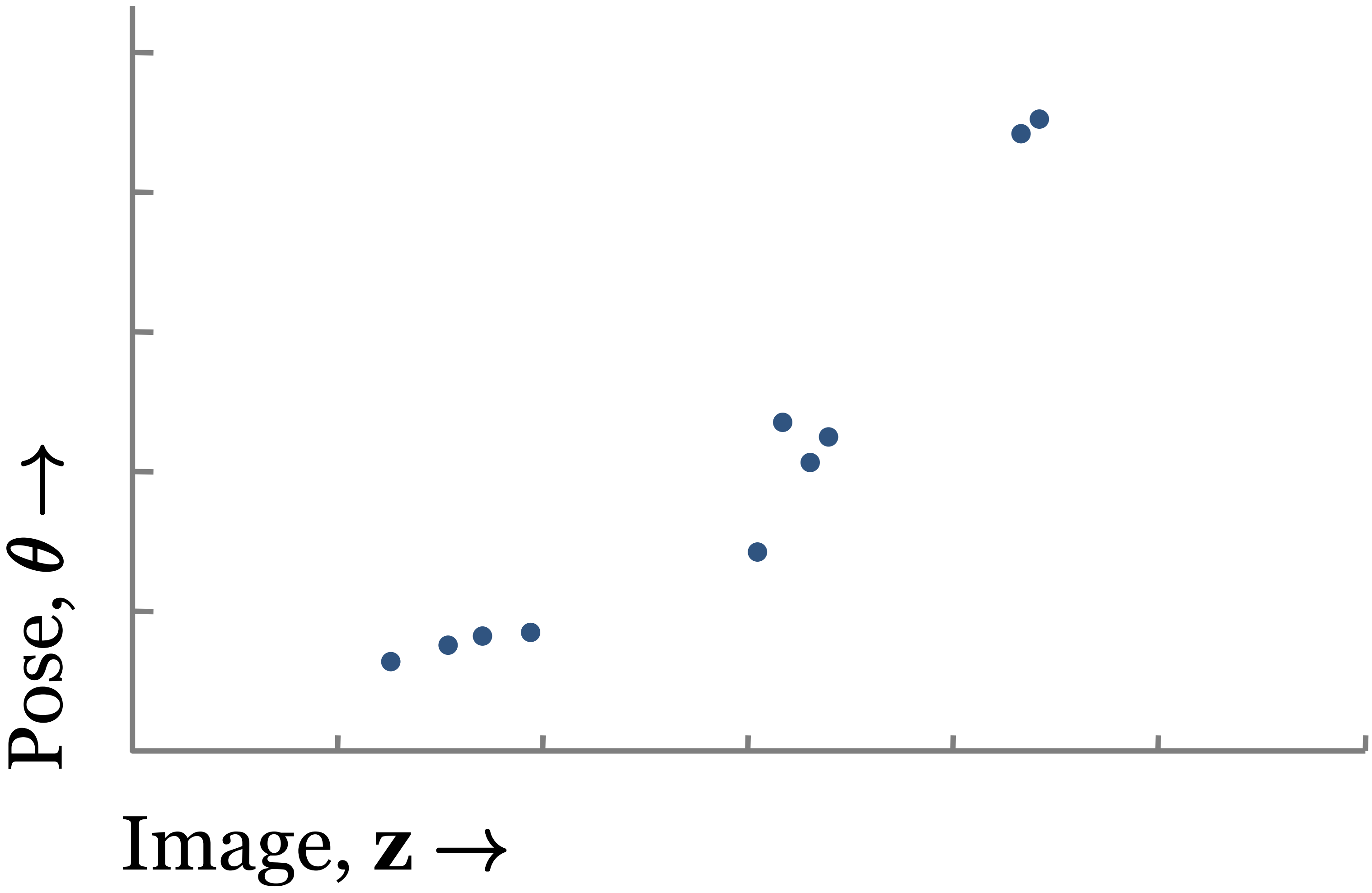
- Compute modes of conditional at every frame
- Choose sequence of modes to maximize product of likelihood and temporal smoothness using Viterbi

But…

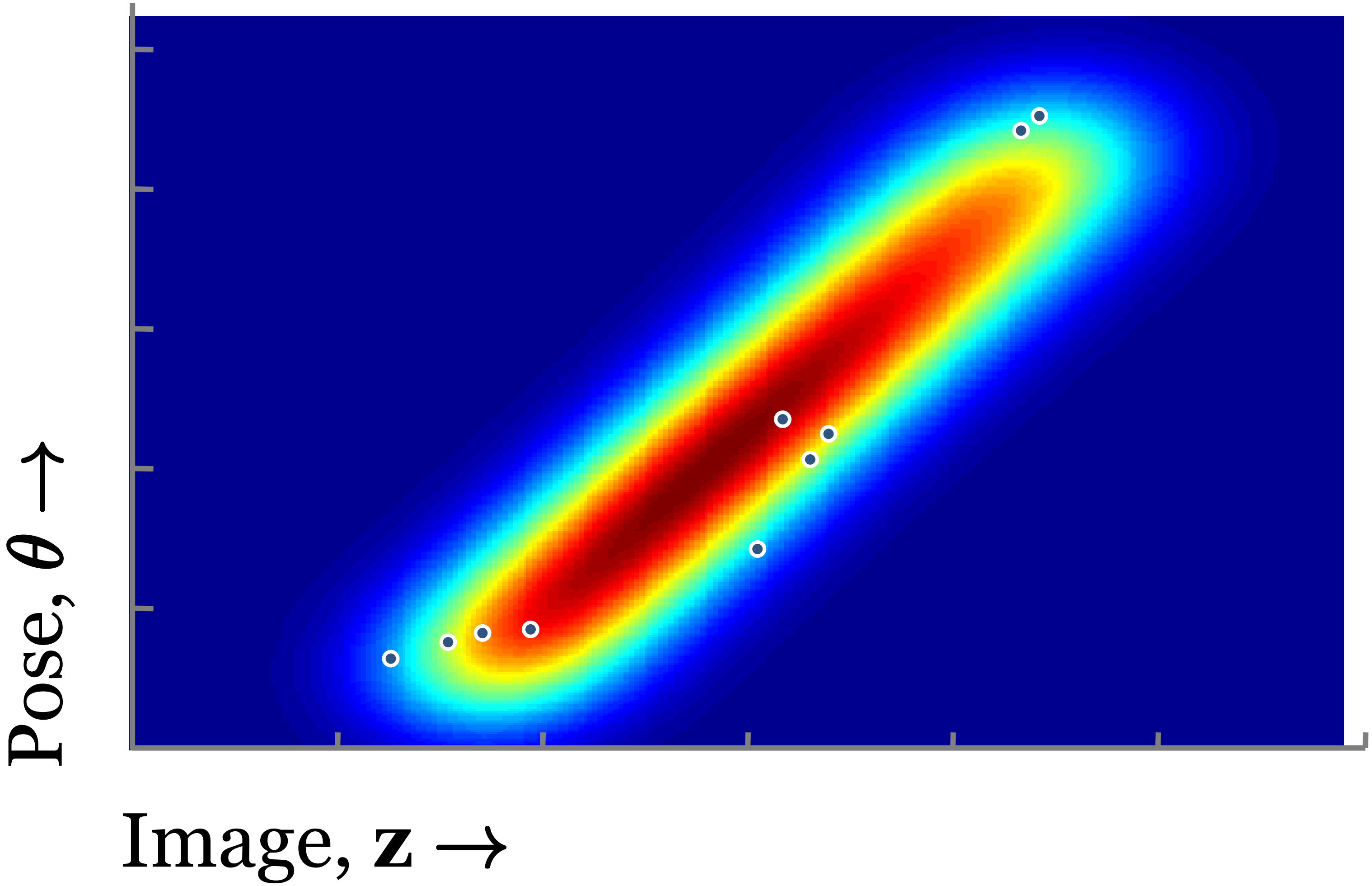
Instead of this:



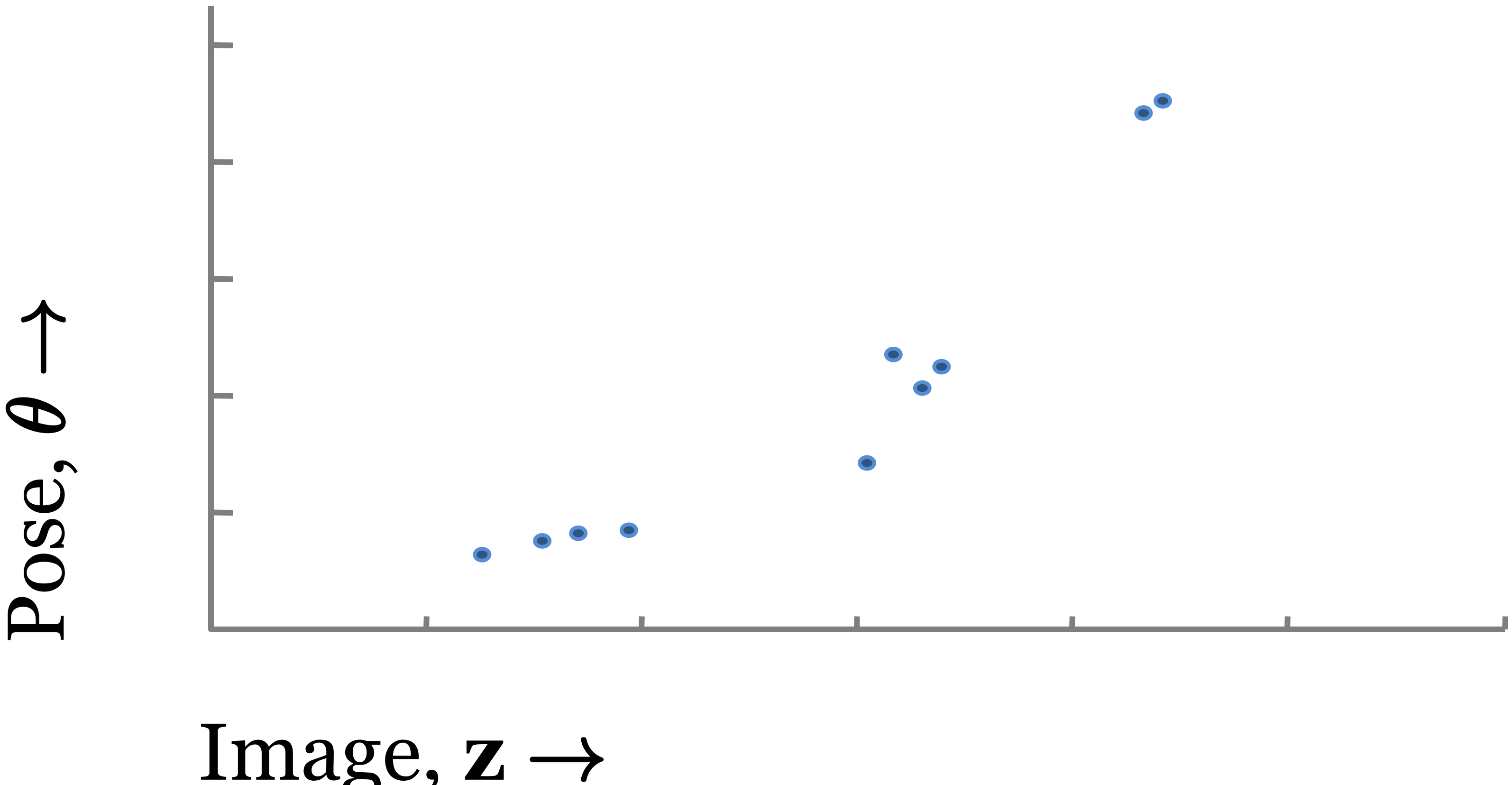
We have this:



Of which a not unreasonable density estimate is:

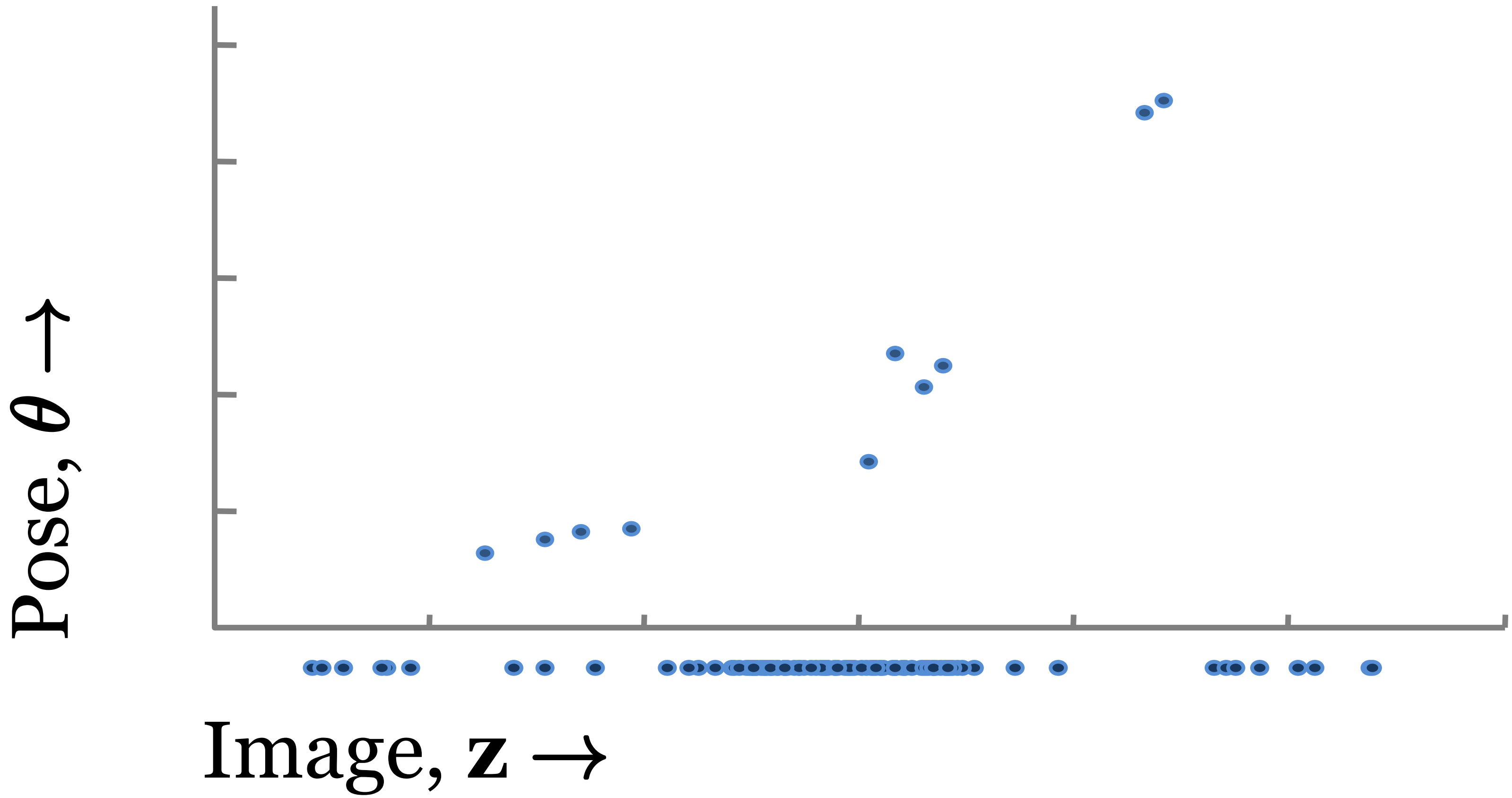


We have too little training data, i.e. too few labelled ($\mathbf{z}$,θ) pairs

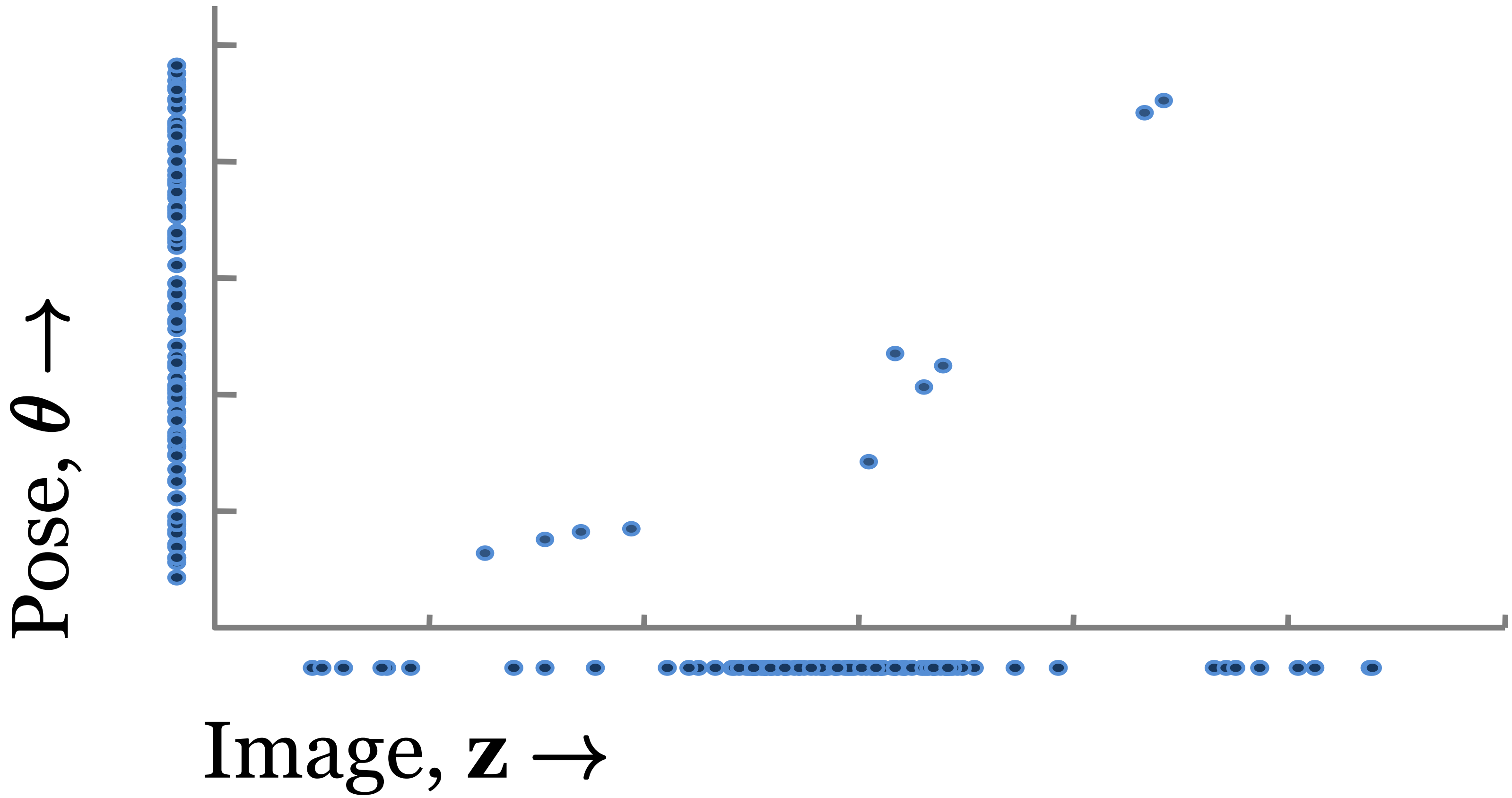


We can't get more because labelling images is expensive...

But we can easily capture more ***unlabelled*** images, i.e. (z,*) pairs

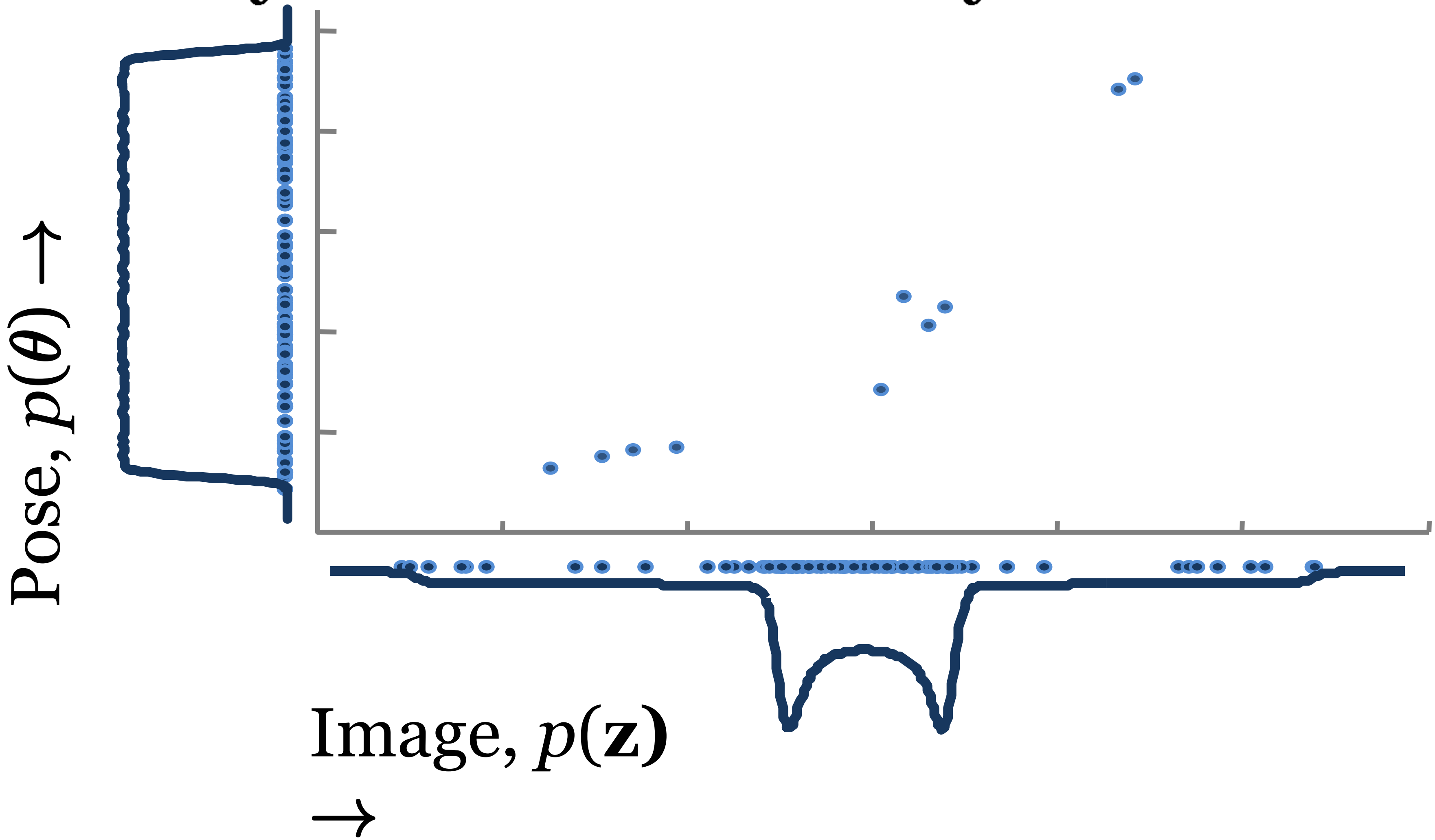


And we can easily download more mocap data without images, i.e. more $(*,\theta)$ pairs

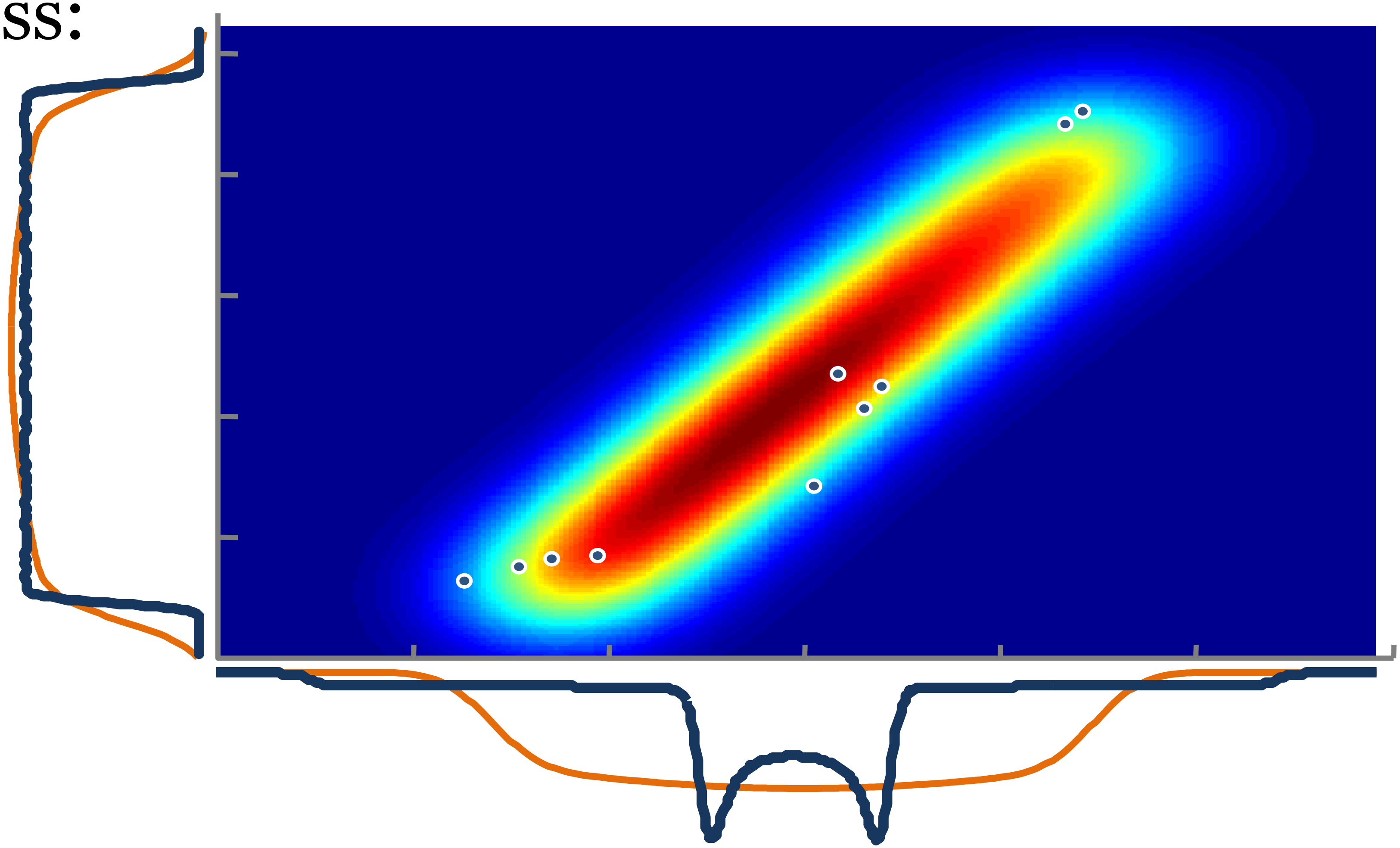


In fact, it's as if we know the **marginals**

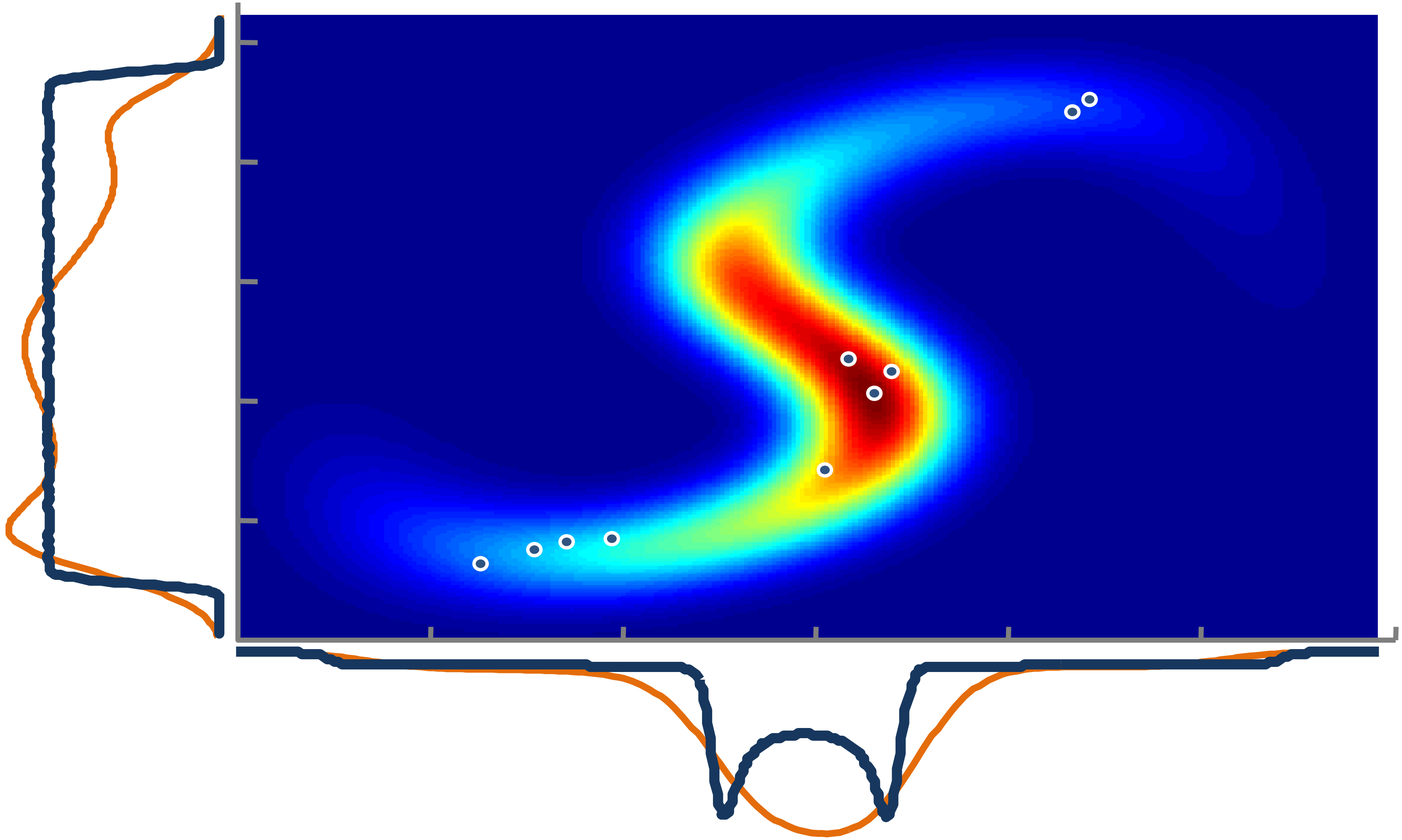
$p(\boldsymbol{\theta}) = \int p(\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z}$ and $p(\mathbf{z}) = \int p(\mathbf{z}, \boldsymbol{\theta}) d\boldsymbol{\theta}$



Which contradict the marginals of our earlier guess:



[ffwd] Using the marginal samples gives this:



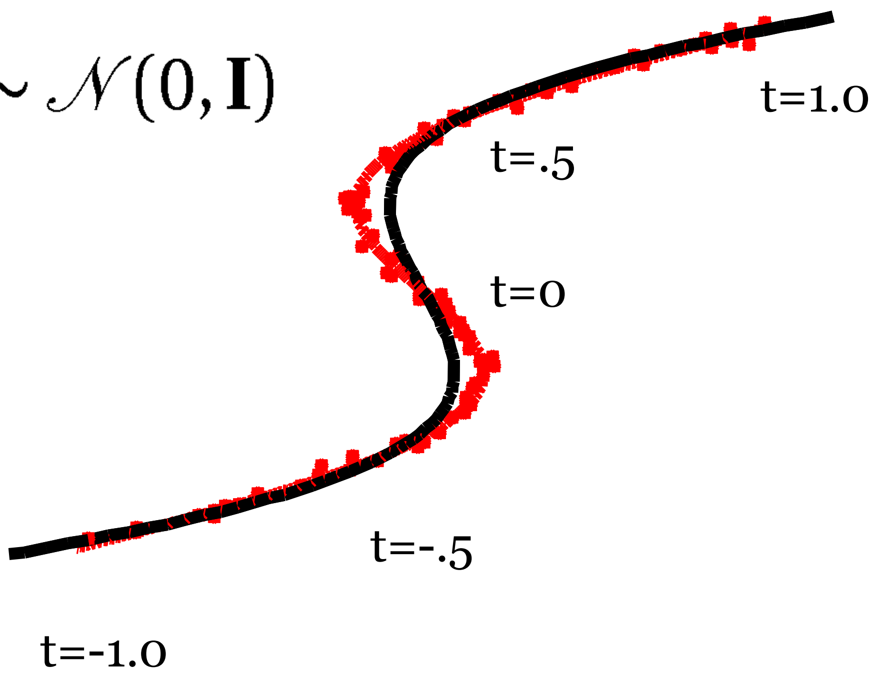
Joint manifold model

Joint density $p(\mathbf{z}, \boldsymbol{\theta}) = \int p(\boldsymbol{\theta}|\mathbf{t})p(\mathbf{z}|\mathbf{t})p(\mathbf{t})d\,\mathbf{t}$

Or, loosely, the "spine" of the joint density is a manifold

$$\begin{pmatrix} \boldsymbol{\theta}(\mathbf{t}) \\ \mathbf{z}(\mathbf{t}) \end{pmatrix} \qquad \mathbf{t} \sim \mathcal{N}(0, \mathbf{I})$$

plus noise.



Joint manifold model

Find latent variables $\mathbf{t}$ to maximize the posterior of training data $\mathbf{D} = \{(\boldsymbol{\theta}_l, \mathbf{z}_l)\}_{l=1}^{L}$

$$
\begin{aligned}
p(\mathbf{t}_{1..L}|\mathbf{D}) &\propto p(\mathbf{D}|\mathbf{t}_{1..L})p(\mathbf{t}_{1..L}) \\
&= p(\boldsymbol{\theta}_{1..L}, \mathbf{z}_{1..L}|\mathbf{t}_{1..L})p(\mathbf{t}_{1..L}) \\
&= p(\boldsymbol{\theta}_{1..L}|\mathbf{t}_{1..L})p(\mathbf{z}_{1..L}|\mathbf{t}_{1..L})p(\mathbf{t}_{1..L}).
\end{aligned}
$$

Adding the marginal samples…

Maximize the posterior of

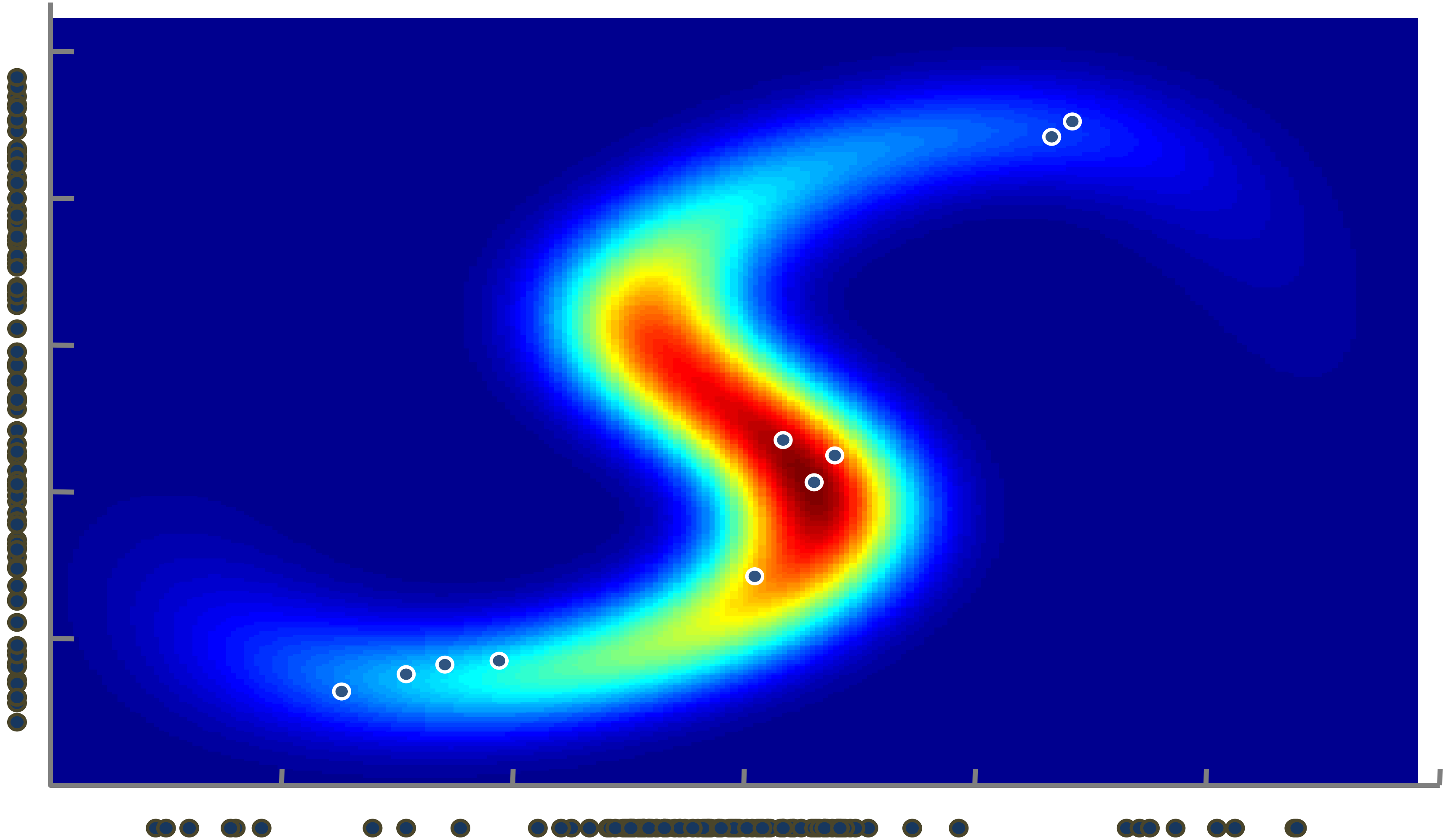
- joint training data $\mathbf{D} = \{(\boldsymbol{\theta}_l, \mathbf{z}_l)\}_{l=1}^{L}$
- marginal $\boldsymbol{\theta}$ data $\mathbf{M} = \{(\boldsymbol{\theta}_l^*, *)\}_{l=1}^{L}$
- marginal $\mathbf{z}$ data $\mathbf{N} = \{(*, \mathbf{z}_l^*)\}_{l=1}^{L}$

$$p(\boldsymbol{\theta}_{1..L}|\mathbf{t}_{1..L})p(\mathbf{z}_{1..L}|\mathbf{t}_{1..L})\times$$

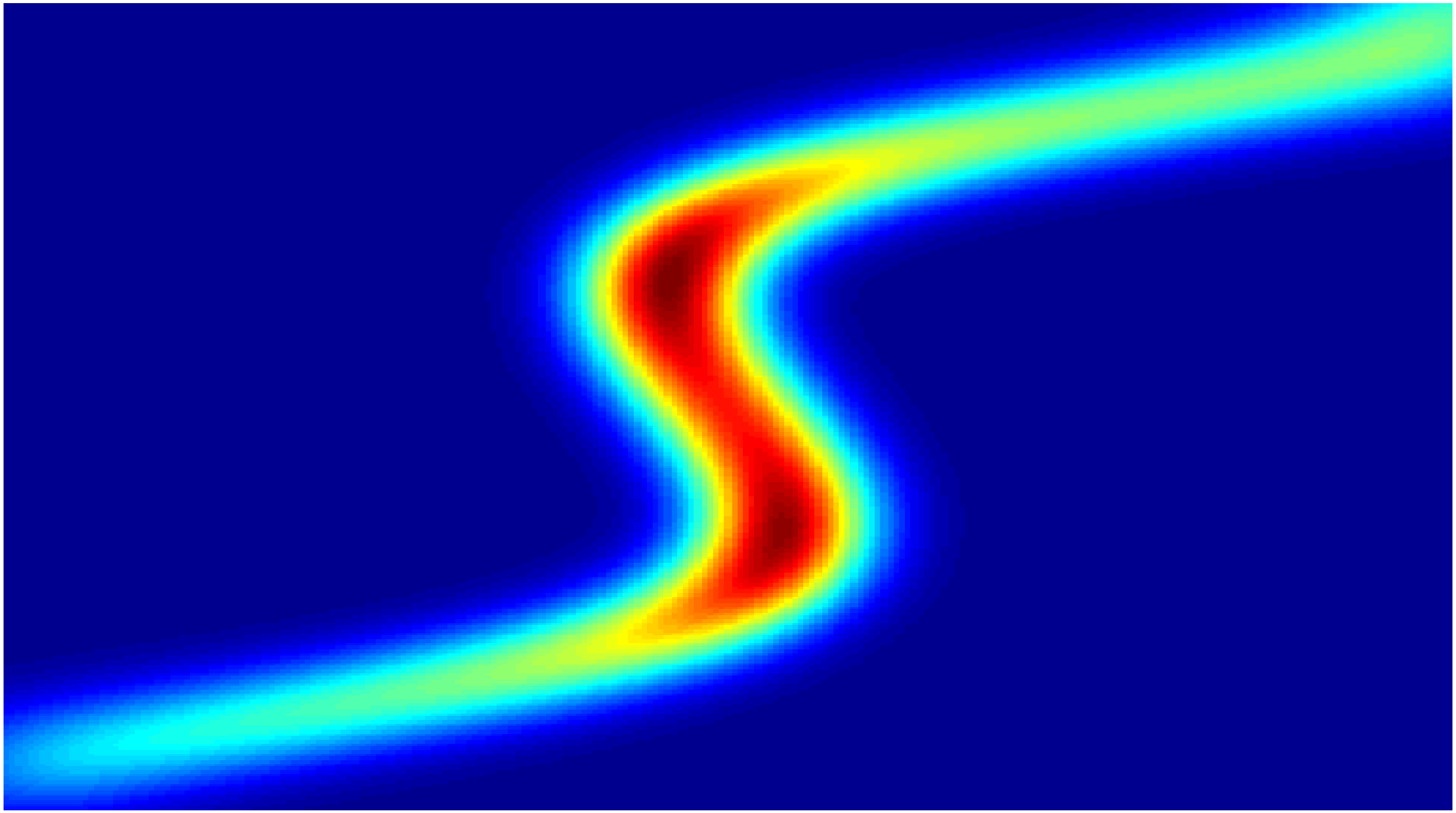
$$p(\boldsymbol{\theta}^*_{1..L}|\mathbf{t}^{\boldsymbol{\theta}}_{1..L}) \times p(\mathbf{z}^*_{1..L}|\mathbf{t}^{\mathbf{z}}_{1..L})\times$$

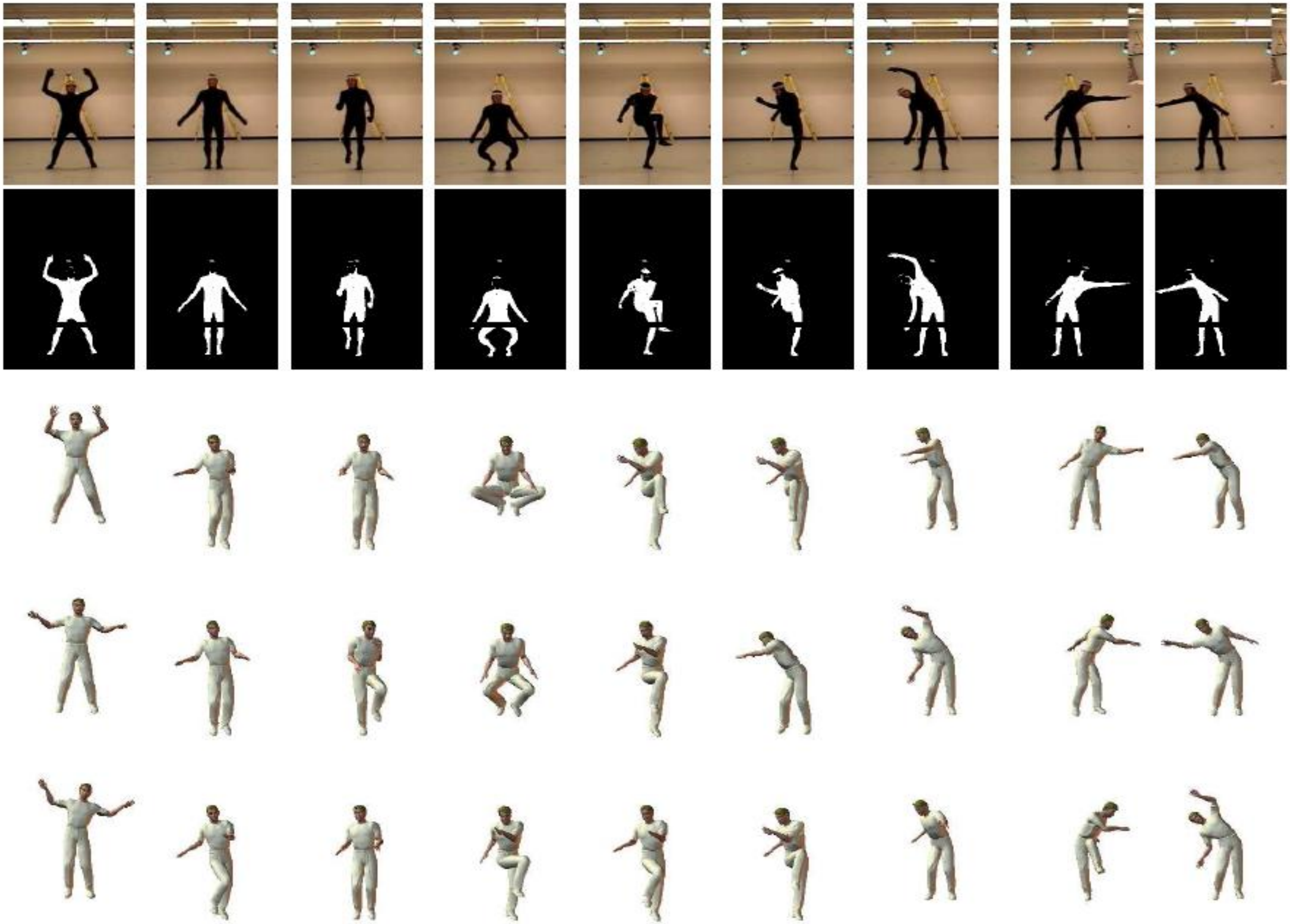
$$p(\mathbf{t}_{1..L}, \mathbf{t}^{\boldsymbol{\theta}}_{1..L}, \mathbf{t}^{\mathbf{z}}_{1..L})$$

Optimizing using scaled CG gives:

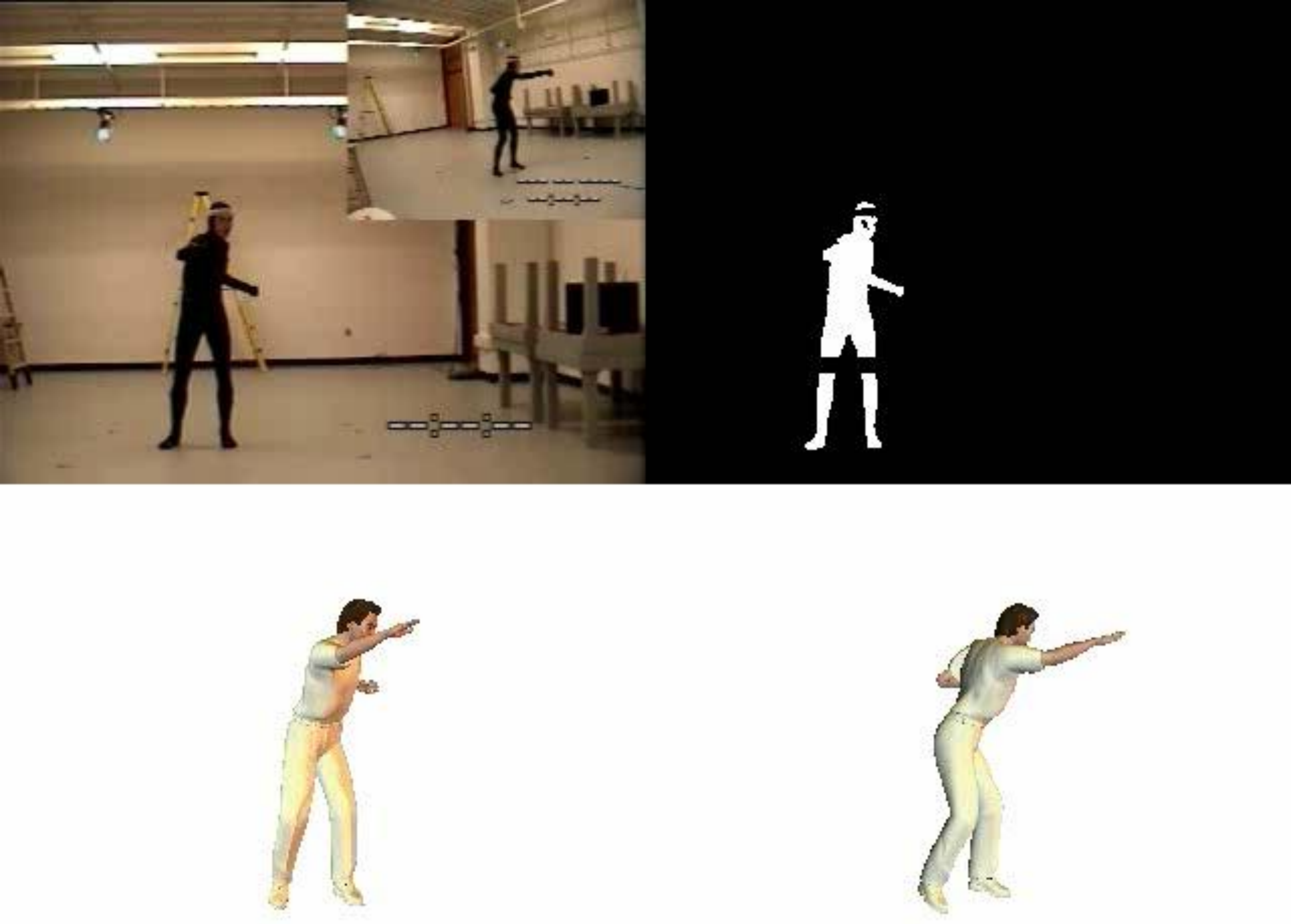


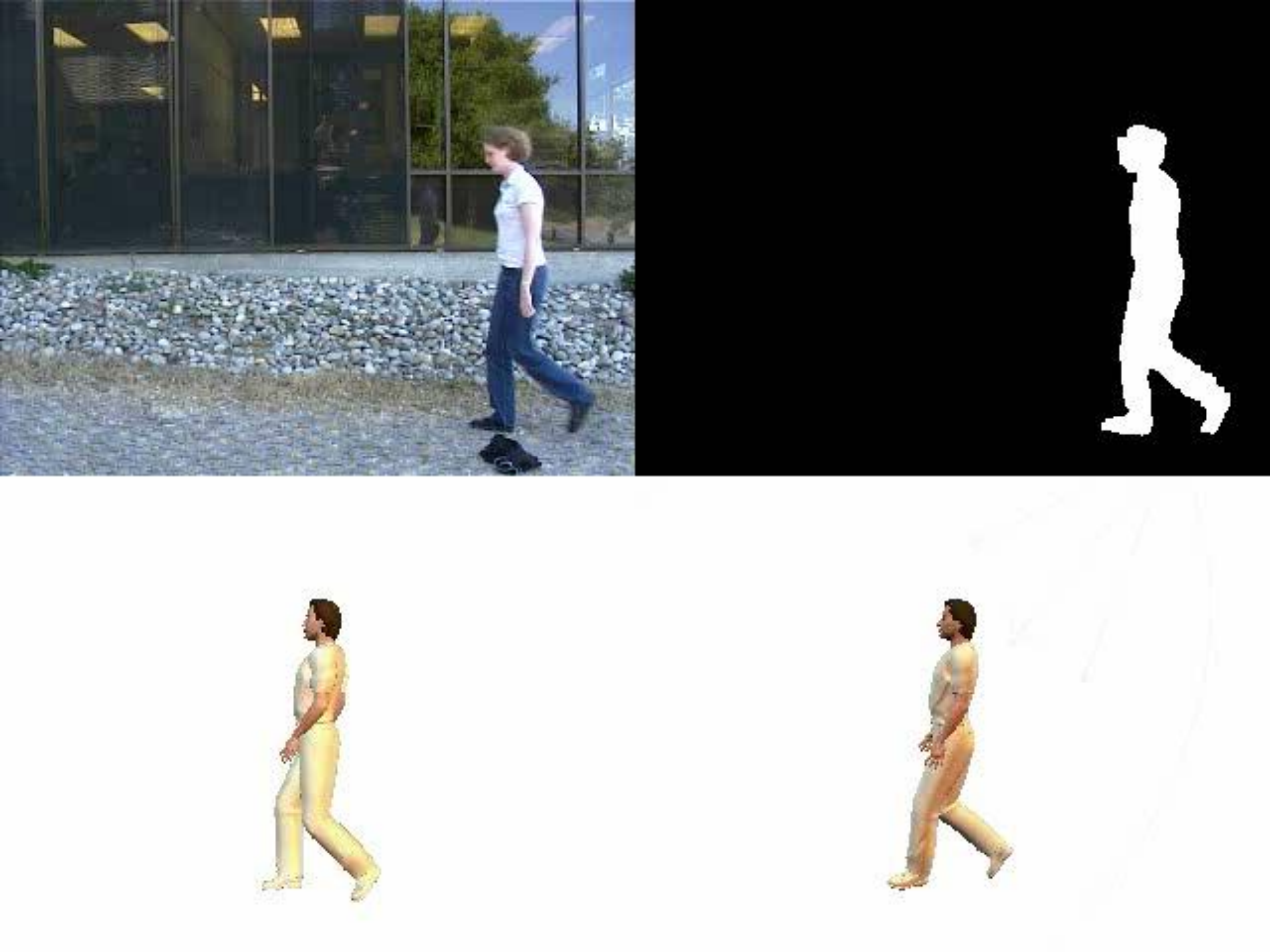
The estimate from 100 training pairs:

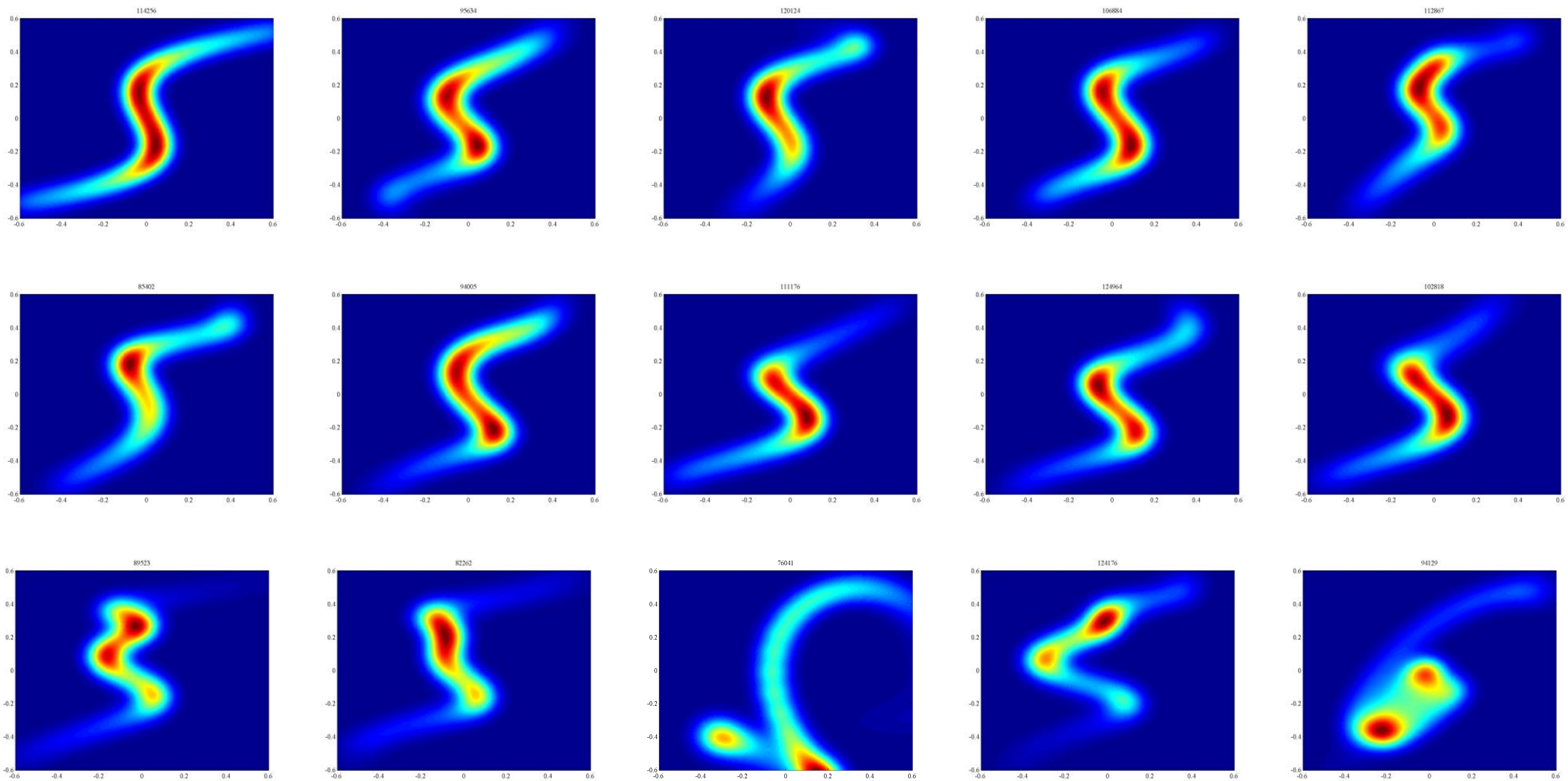




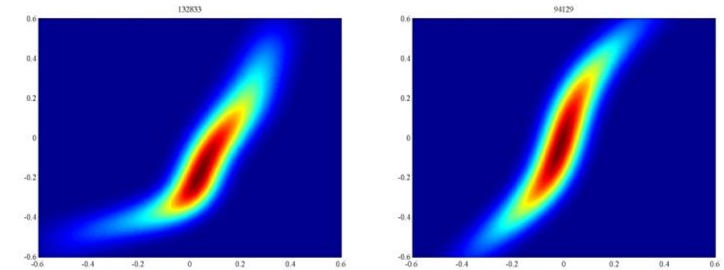
Applied to real-world example



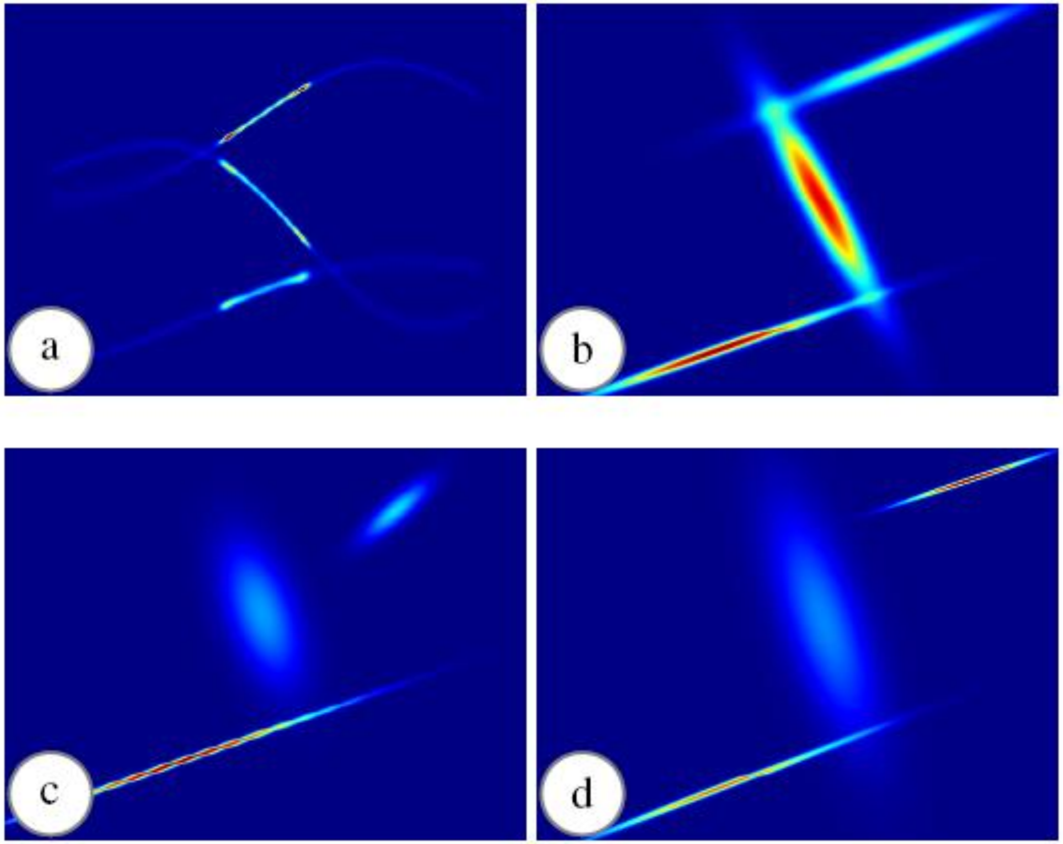




Multiple runs...

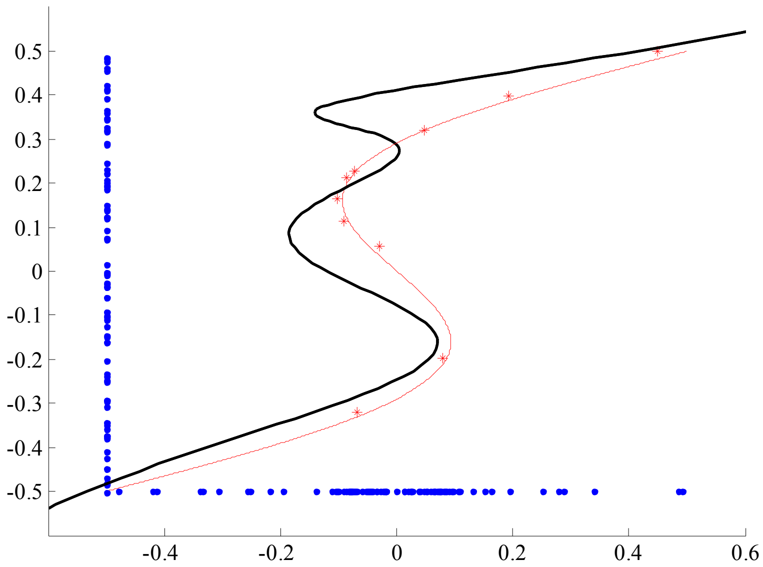


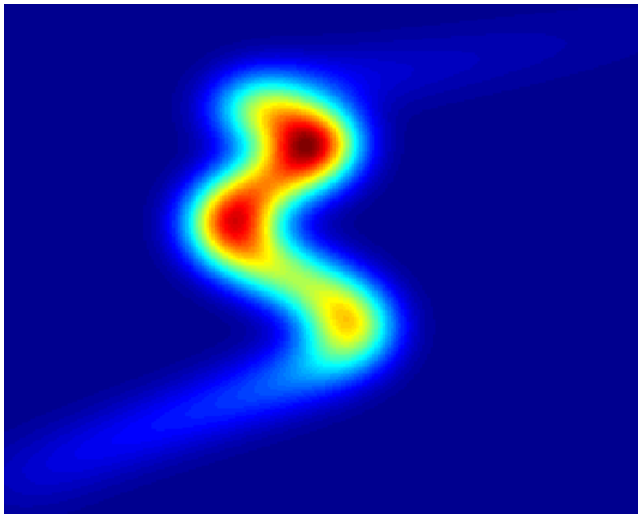
Other methods



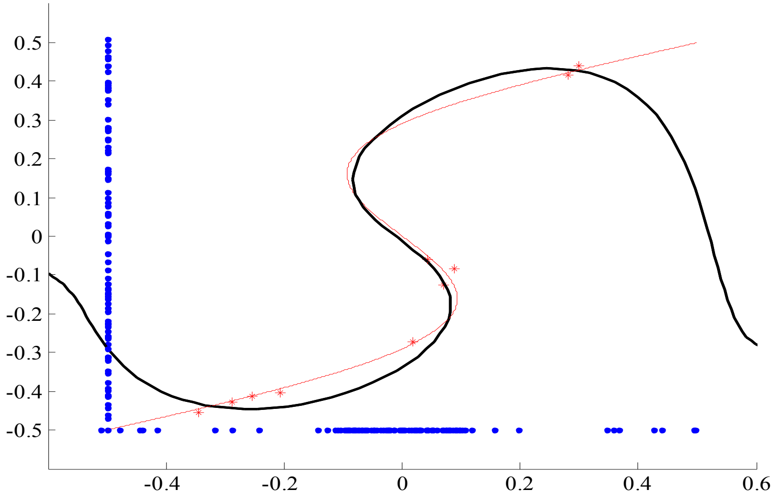
a. Mixture of experts, 15J
b. GMM, 15J
c. GMM, 8J
d. GMM, 8J, 104M

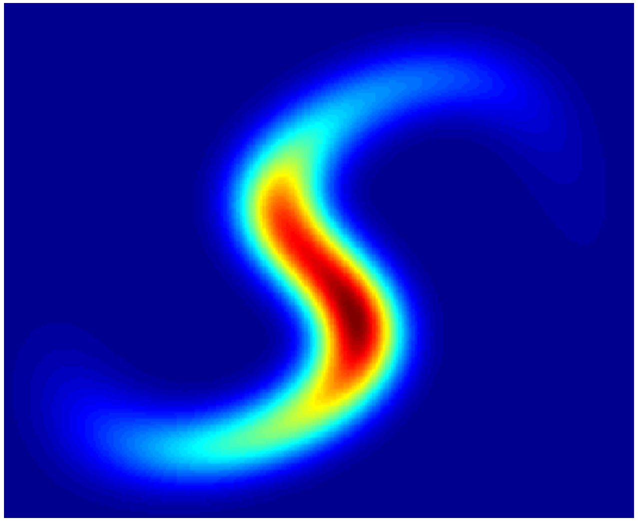
Failure modes



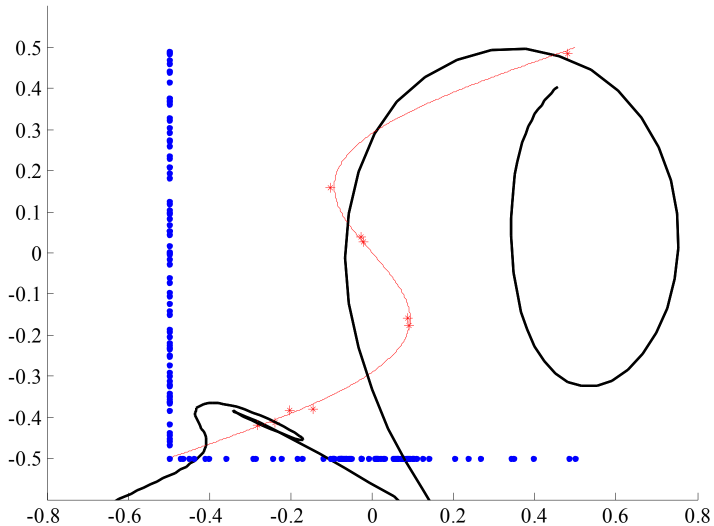


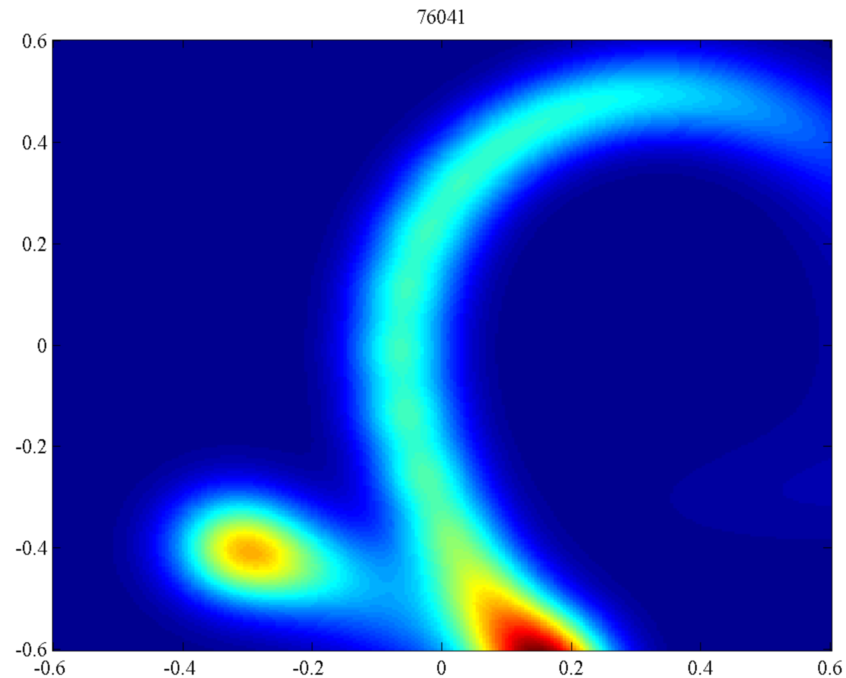
Bad convergence



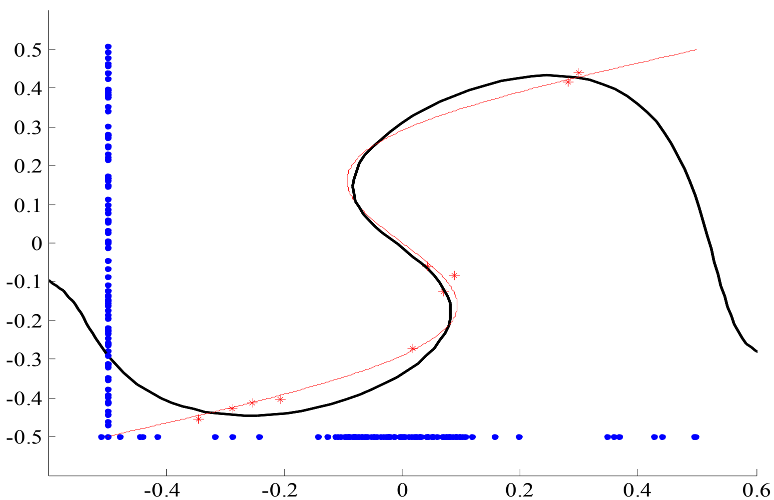


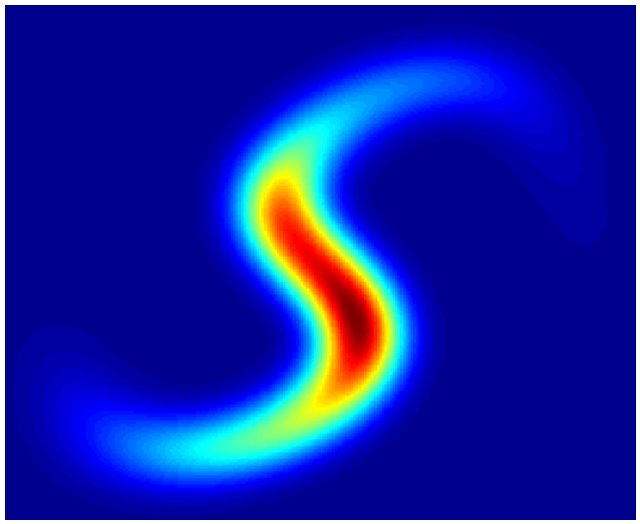
Failure modes



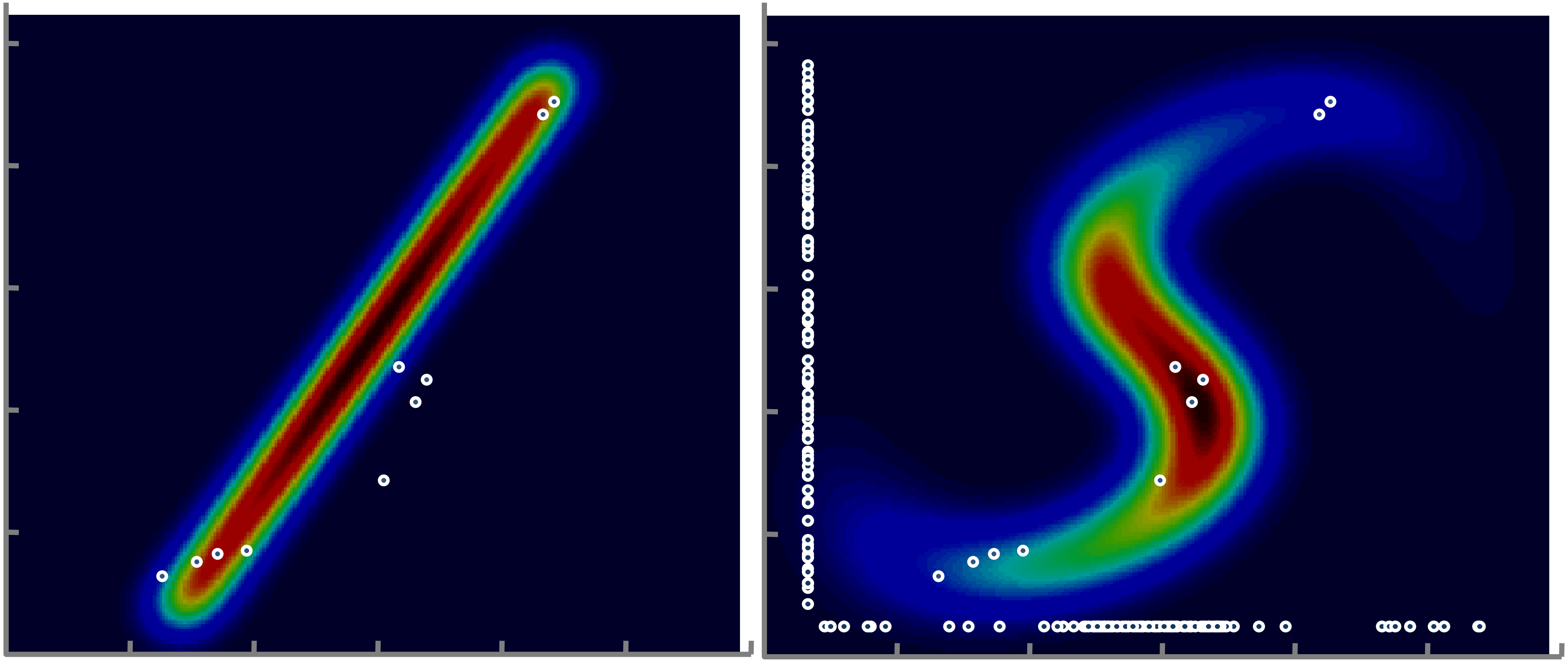


Bad initialization





Summary: Use all your data



http://www.research.microsoft.com/~awf/jmm

Discussion points

1. "Semi-supervised" is oft-heard
   - Pre-fitting manifold with PCA is "semi-supervised"
   - Simultaneous learning of manifolds (Shon et al) could easily be made SS (indeed, see our experiments in the paper)
   - But the S curve example cannot benefit from manifold learning in each space
2. Not a regressor, nor a joint density model
3. Sensitivity to initialization
4. Can more marginal samples make it worse?

Related work

Surely this has been done a million times?

It depends how you look at it

- GMM with missing data: Ghahramani & Jordan '94
- Joint from marginals: Maxent, Roth, Sigal & Black '04

The whole "semi-supervised" story....

- SS Classification: Discrete, rarely uses p(\theta)
- SS Regression: Survey Zhu '05, never uses p(\theta)
- S^3GP [Williams et al, '06]: univalued, p(z) only.

- Shared GPLVM: univalued, no missing data

Extra slides: Radial basis functions

Radial basis functions/kernel functions centred on kernel centres $\mathbf{z}_k$.

$f(\mathbf{z}) = \sum_k w_k \kappa(\mathbf{z}, \mathbf{z}_k)$, e.g. $\kappa(\mathbf{z}, \mathbf{z}) = e^{-\lambda_k \|\mathbf{z} - \mathbf{z}\|}$

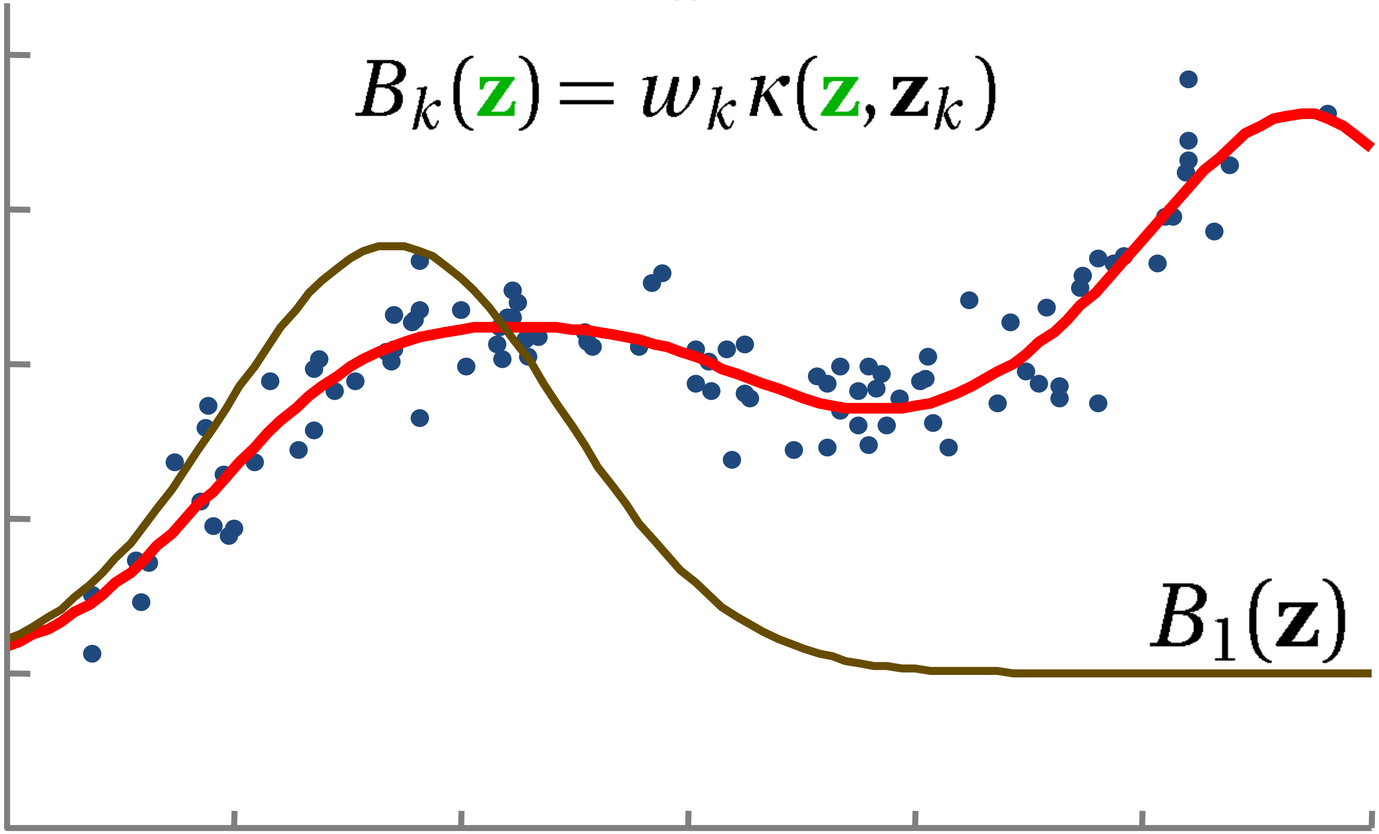
With the parameters $w_k, \lambda_k, \mathbf{z}_k$ found by minimizing

$$\sum_k \|\boldsymbol{\theta}_k - f(\boldsymbol{\theta}_k)\|^2$$

Change this to a Gaussian process or Relevance Vector Machine to get covariances.

$$f(\mathbf{z}) = \sum_k B_k(\mathbf{z})$$

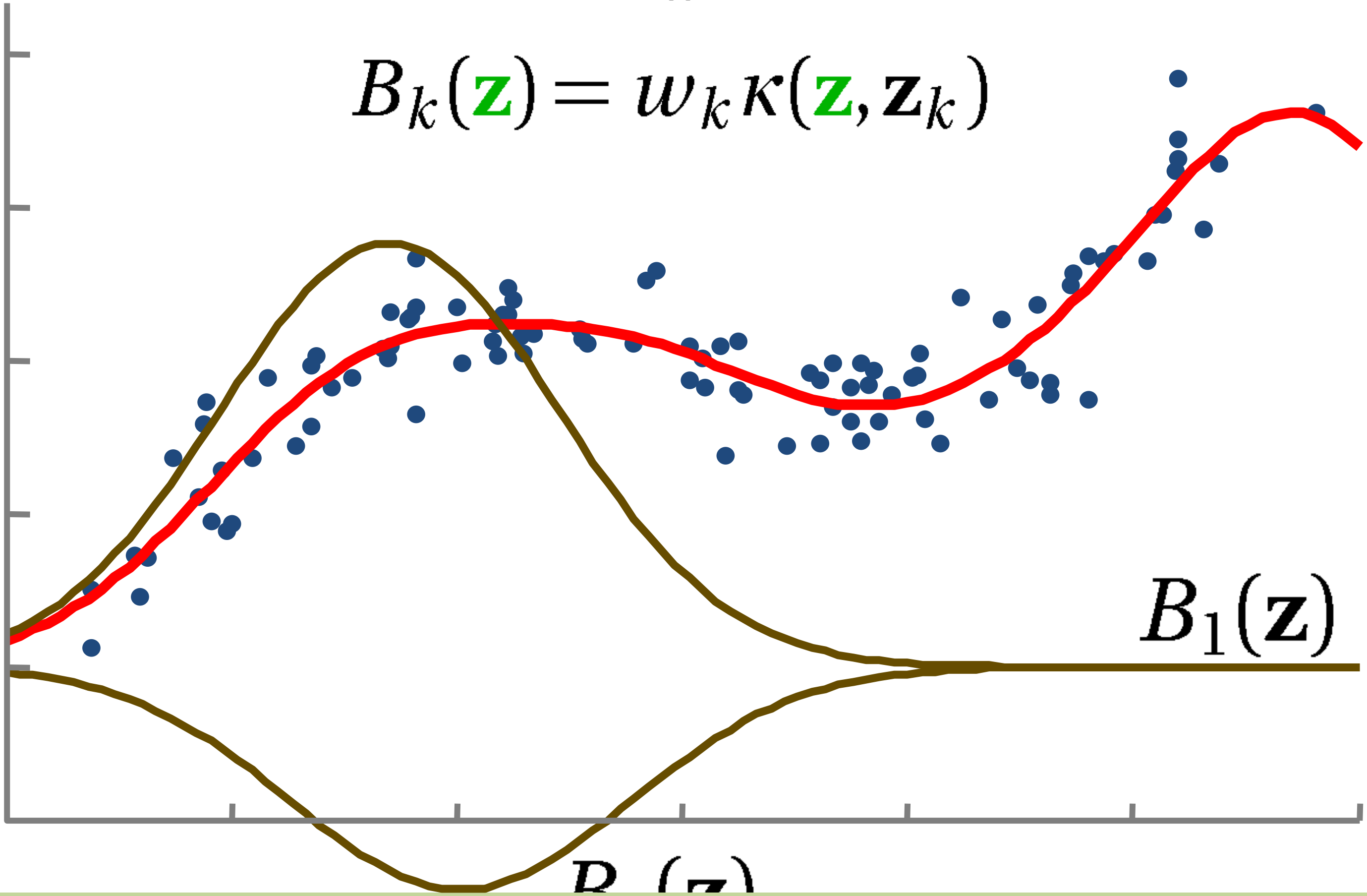
$$B_k(\mathbf{z}) = w_k \kappa(\mathbf{z}, \mathbf{z}_k)$$



A sum of [radial] basis functions

$$f(\mathbf{z}) = \sum_k B_k(\mathbf{z})$$

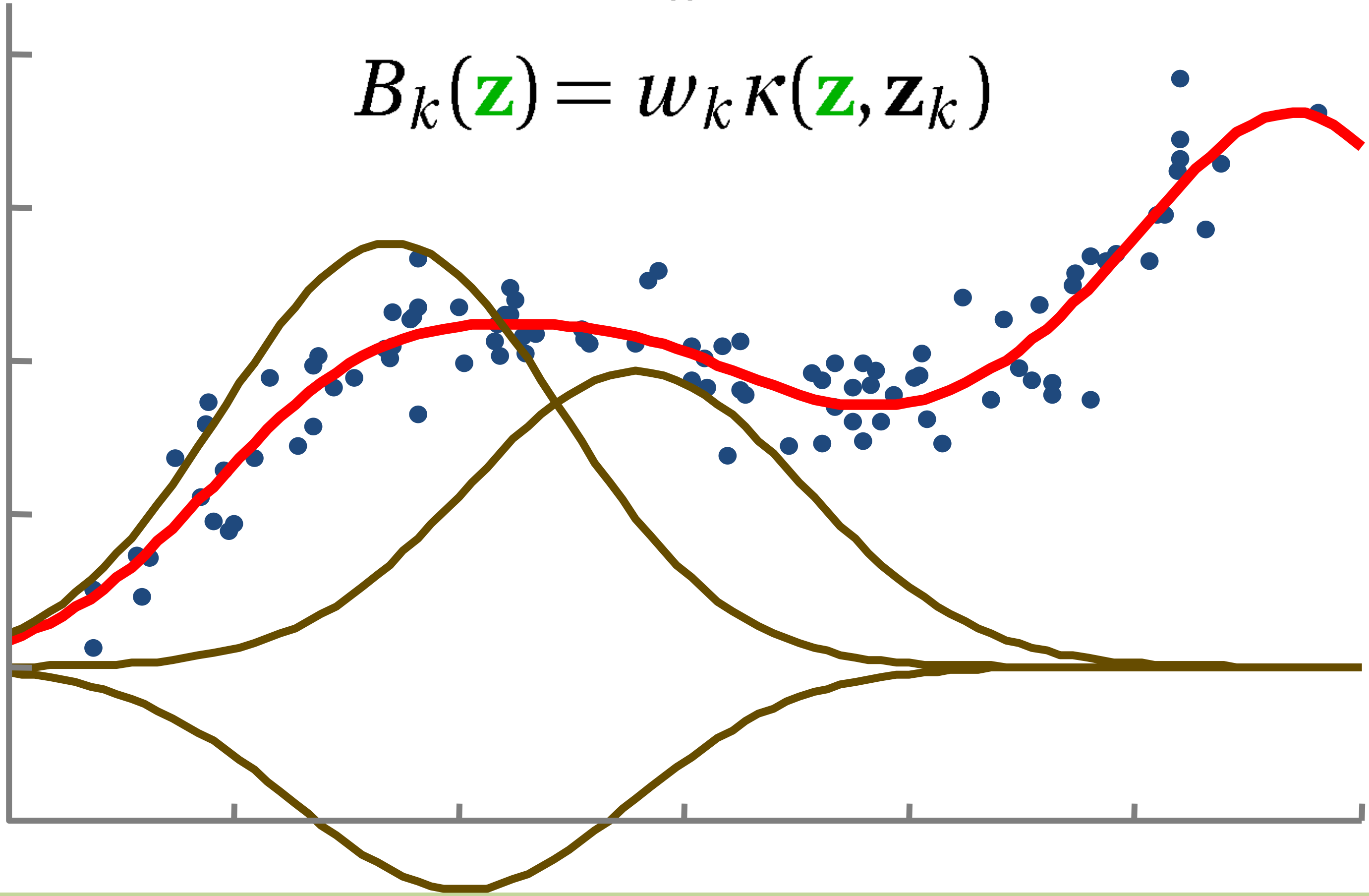
$$B_k(\mathbf{z}) = w_k \kappa(\mathbf{z}, \mathbf{z}_k)$$



Radial basis functions

$$f(\mathbf{z}) = \sum_k B_k(\mathbf{z})$$

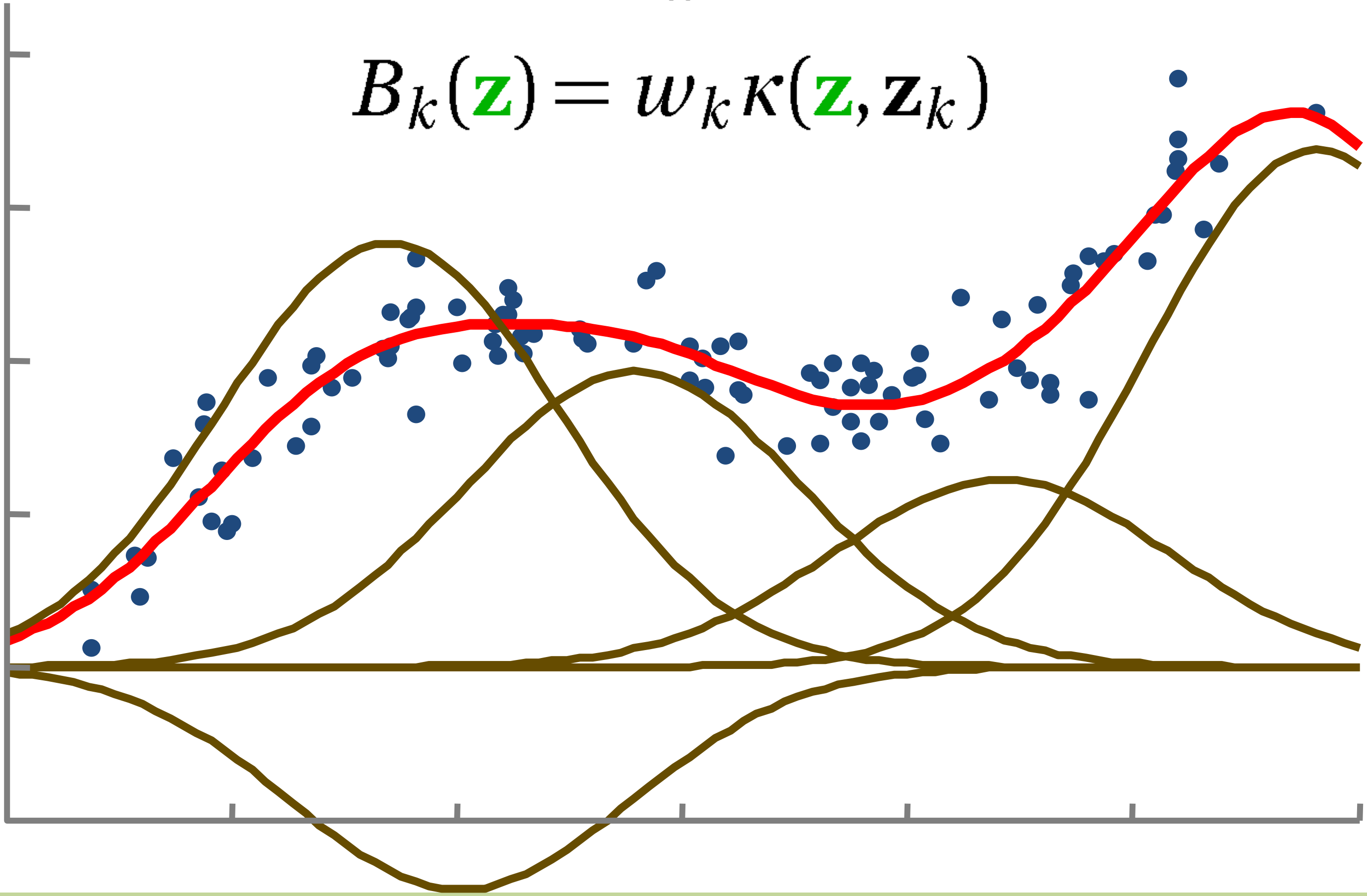
$$B_k(\mathbf{z}) = w_k \kappa(\mathbf{z}, \mathbf{z}_k)$$



Radial basis functions

$$f(\mathbf{z}) = \sum_k B_k(\mathbf{z})$$

$$B_k(\mathbf{z}) = w_k \kappa(\mathbf{z}, \mathbf{z}_k)$$



Radial basis functions