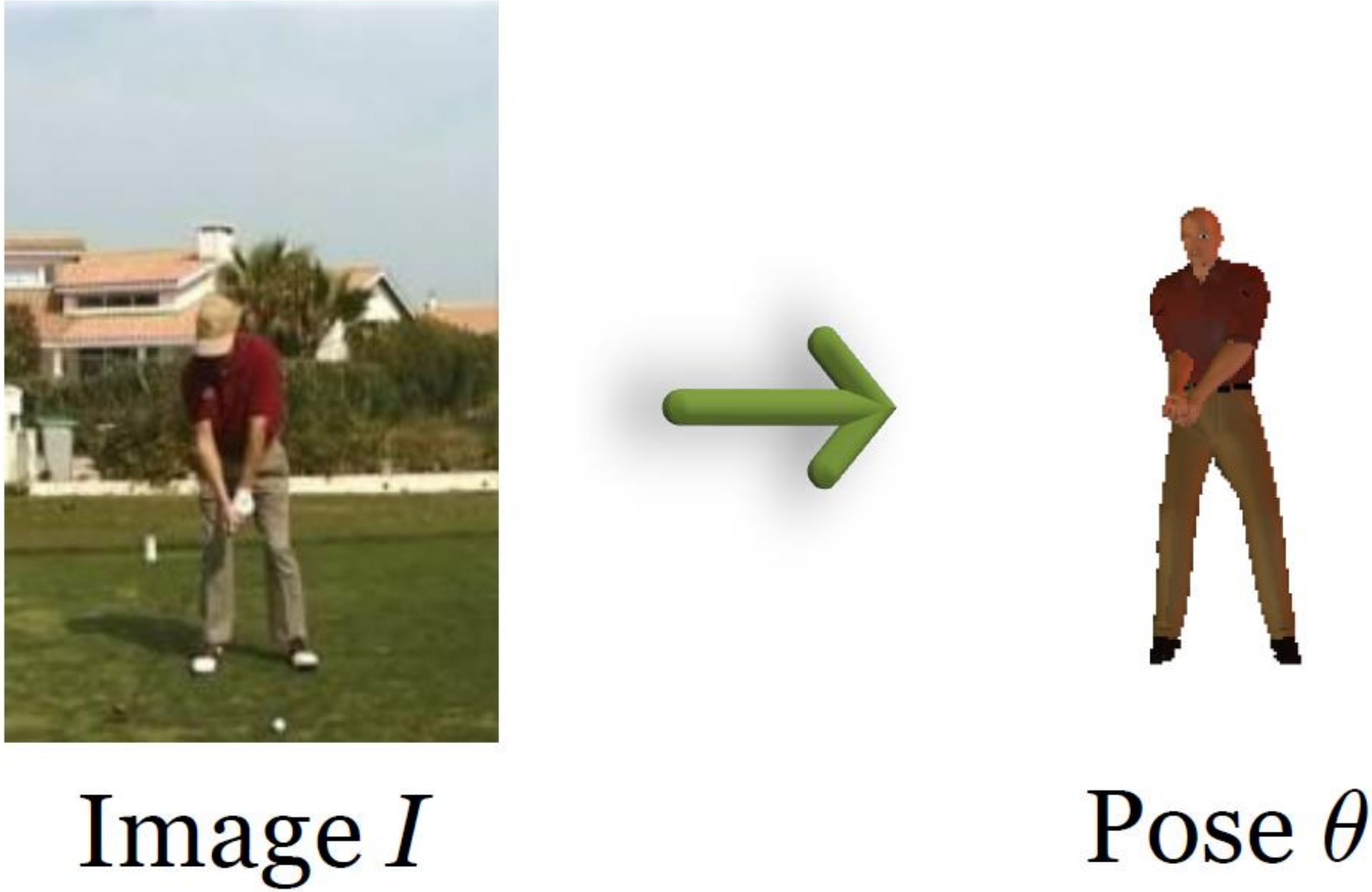
Pose Estimation

Tae-Kyun (T-K) Kim
Senior Lecturer
https://labicvl.github.io/

Further reading:
Navaratnam et al., The Joint Manifold Model for Semi-supervised Multi-valued Regression. ICCV 2007.
http://www.iis.ee.ic.ac.uk/ComputerVision/Research.html



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

A mapping function is learnt from the input image I to the pose vector θ , which is taken as a continuous variable.

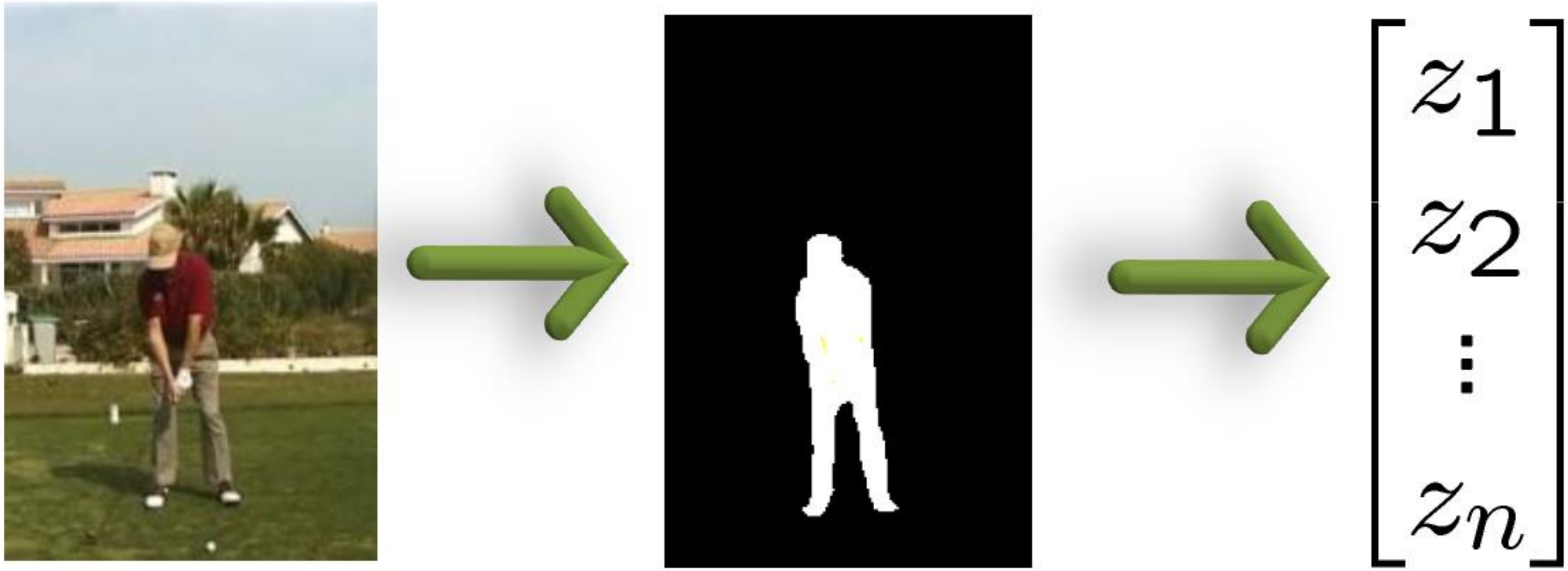
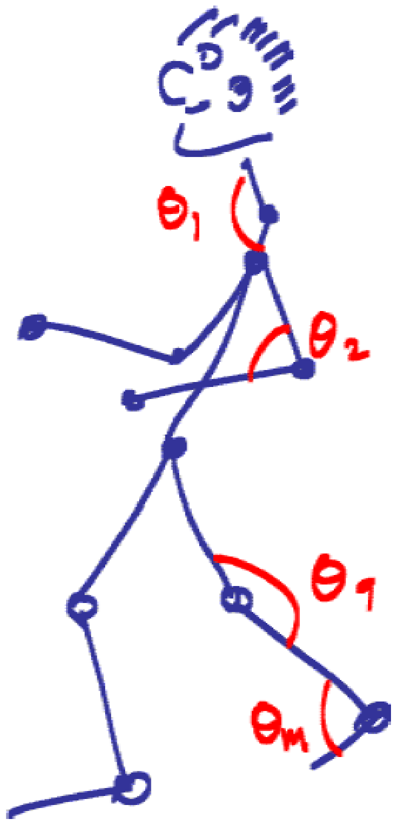


Image I

Feature vector $\mathbf{z}$

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

Typical image processing steps:
Given an image, a silhouette is segmented.
A shape descriptor is applied to the silhouette to yield a finite dimensional vector. (Belongie and Malik, Matching with Shape Contexts, 2000)



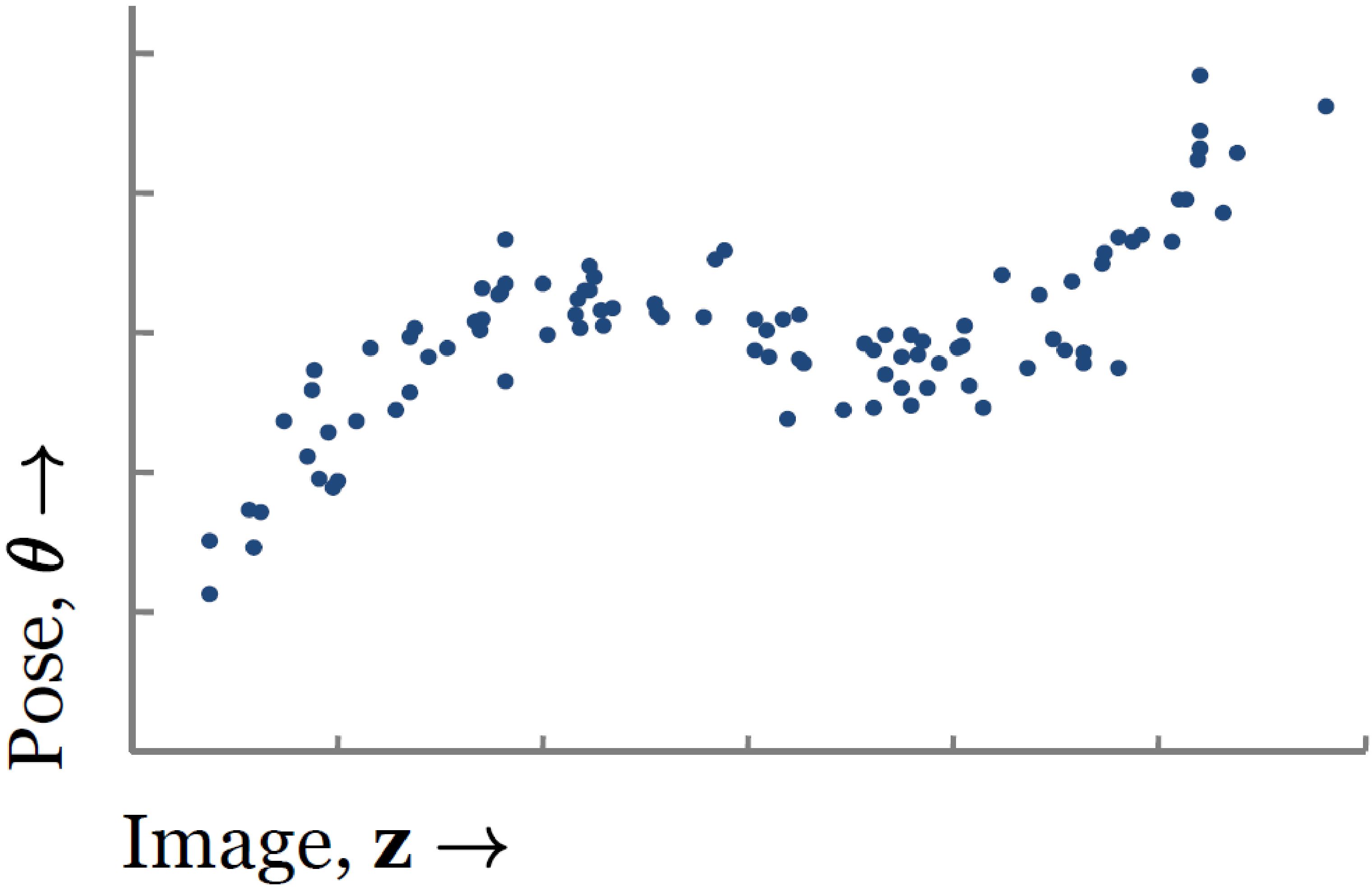
$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{bmatrix}$$

Pose vector $\boldsymbol{\theta}$

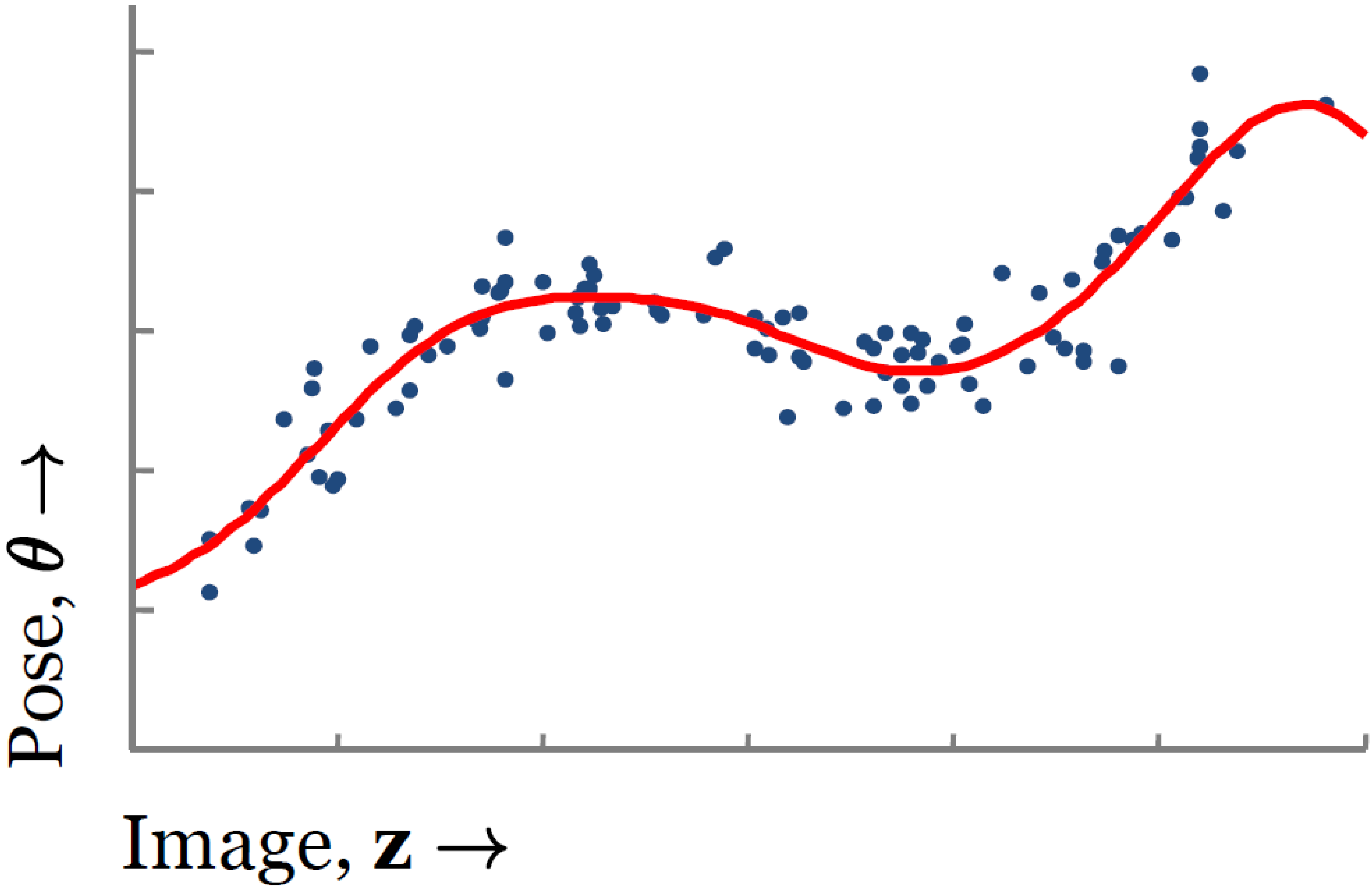
e.g. Joint angles $\boldsymbol{\theta} \in \mathbb{R}^{27}$

The output is a vector of m joint angles.

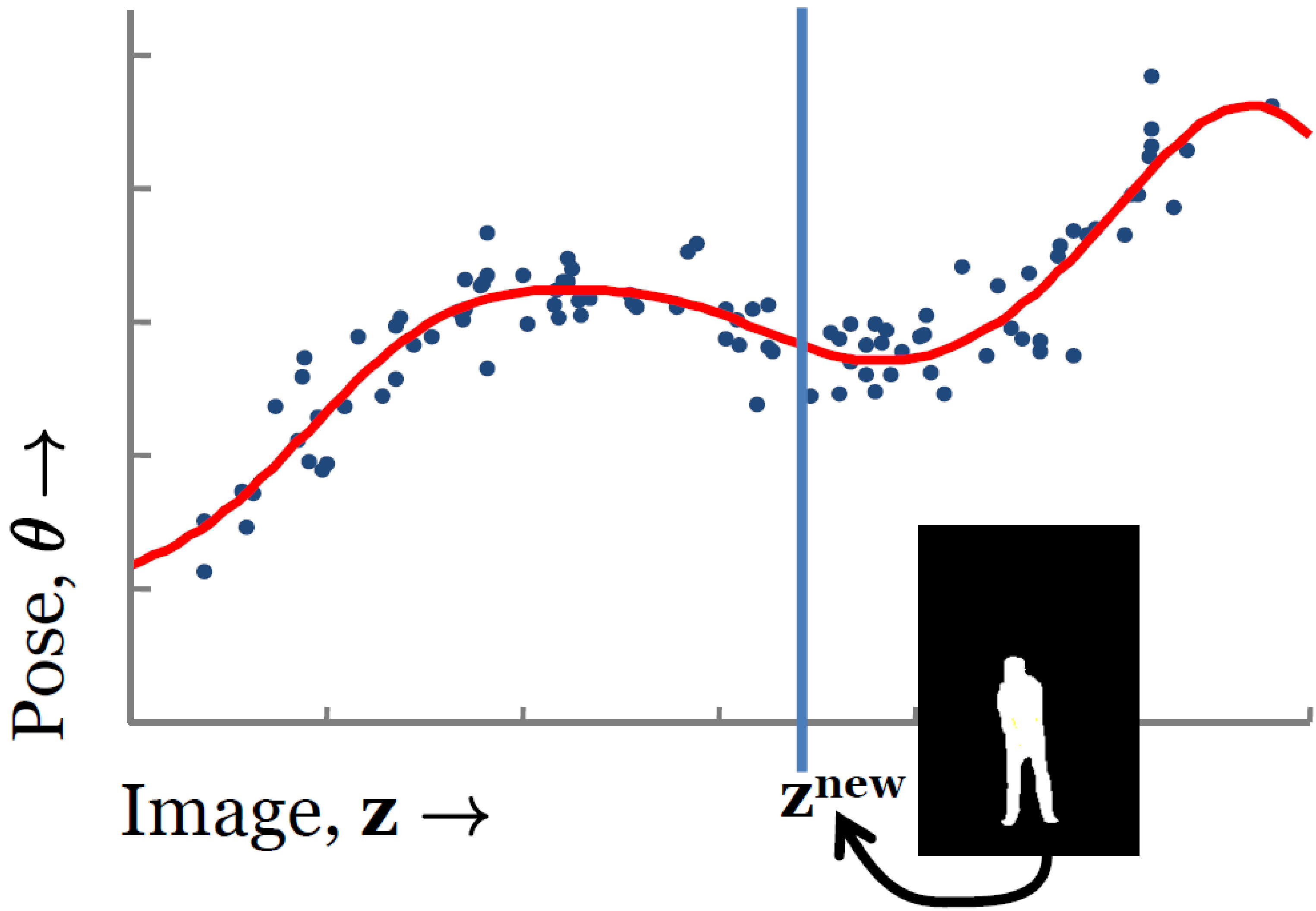
1. Obtain training samples $(\mathbf{z}_1, \boldsymbol{\theta}_1)...(\mathbf{z}_N, \boldsymbol{\theta}_N)$



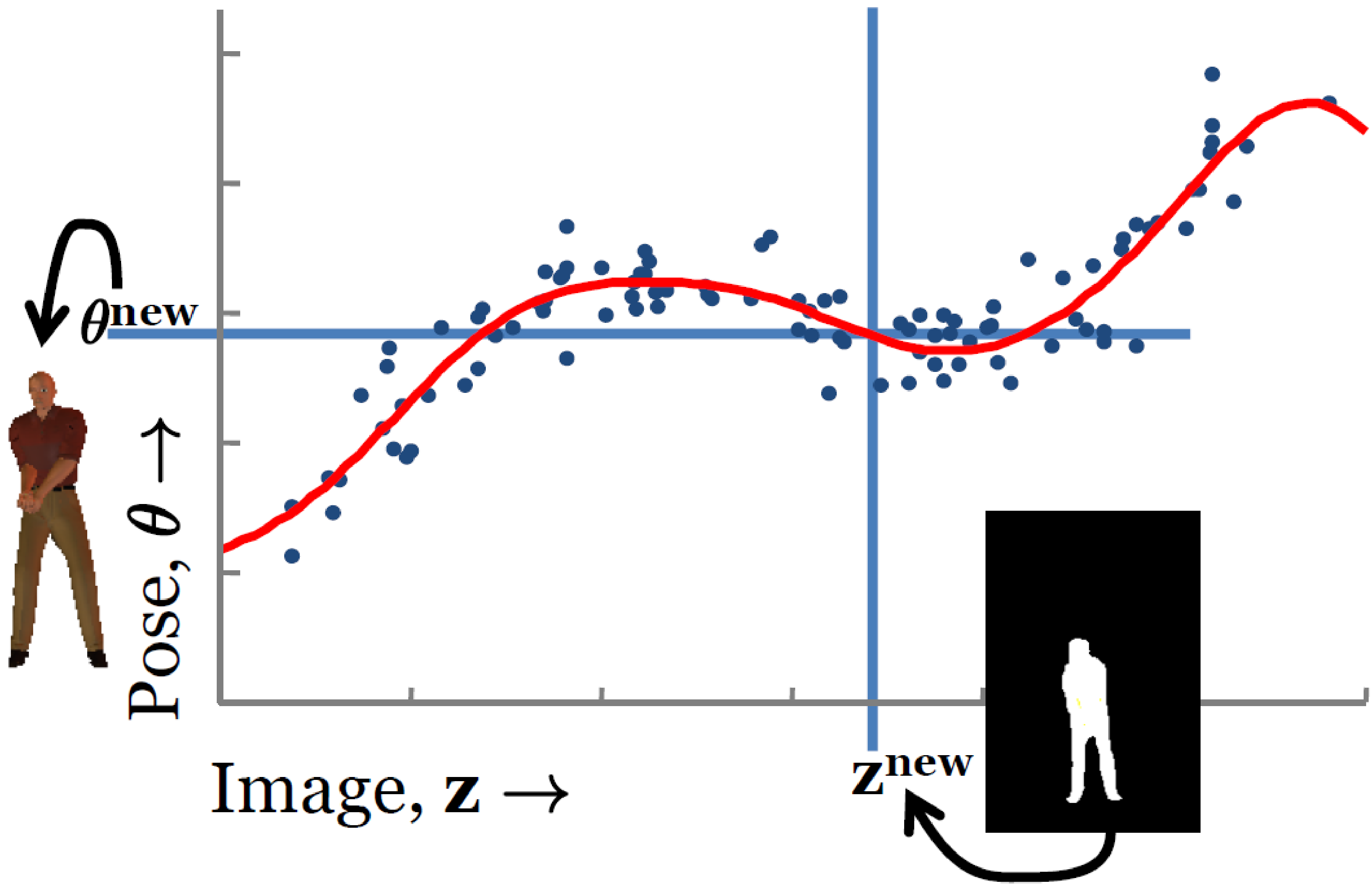
2. Training: Fit function $\boldsymbol{\theta} = f(\mathbf{z})$.



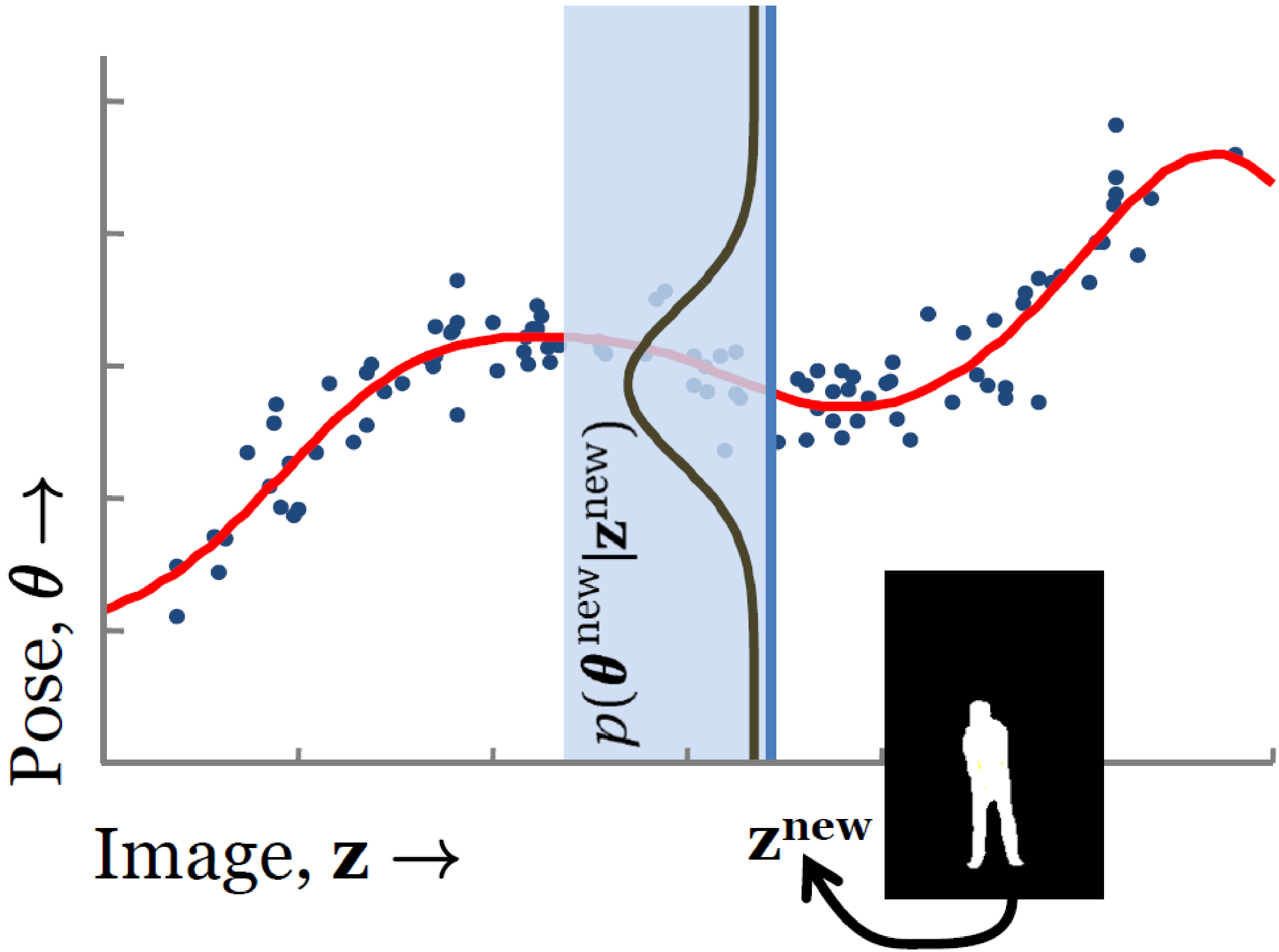
3. Given new image, $\mathbf{z}^{\text{new}}$, compute $\boldsymbol{\theta}^{\text{new}} = f(\mathbf{z}^{\text{new}})$.



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…

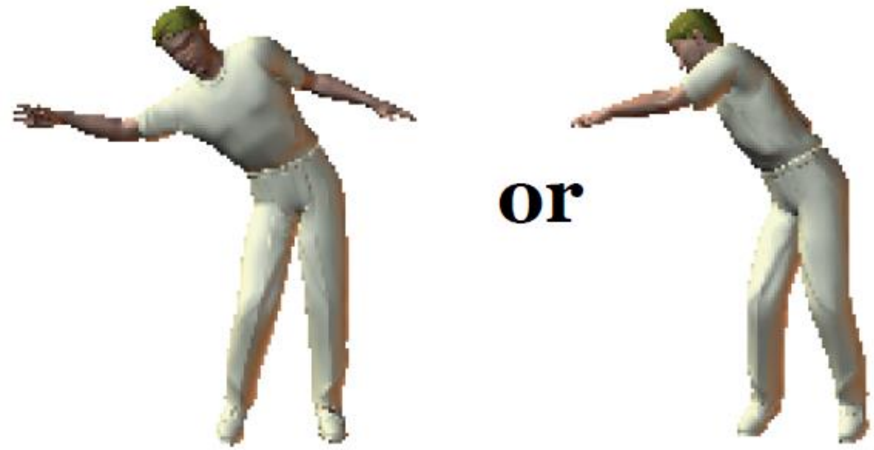
– f is multivalued

– $\mathbf{z}$ and θ live in high dimensions

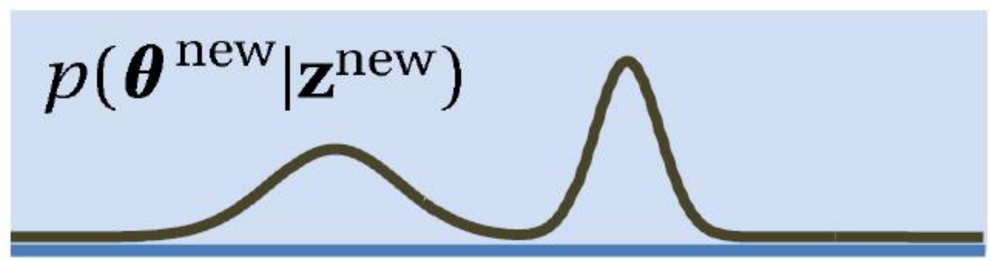
Multivalued f:

Multivalued f:

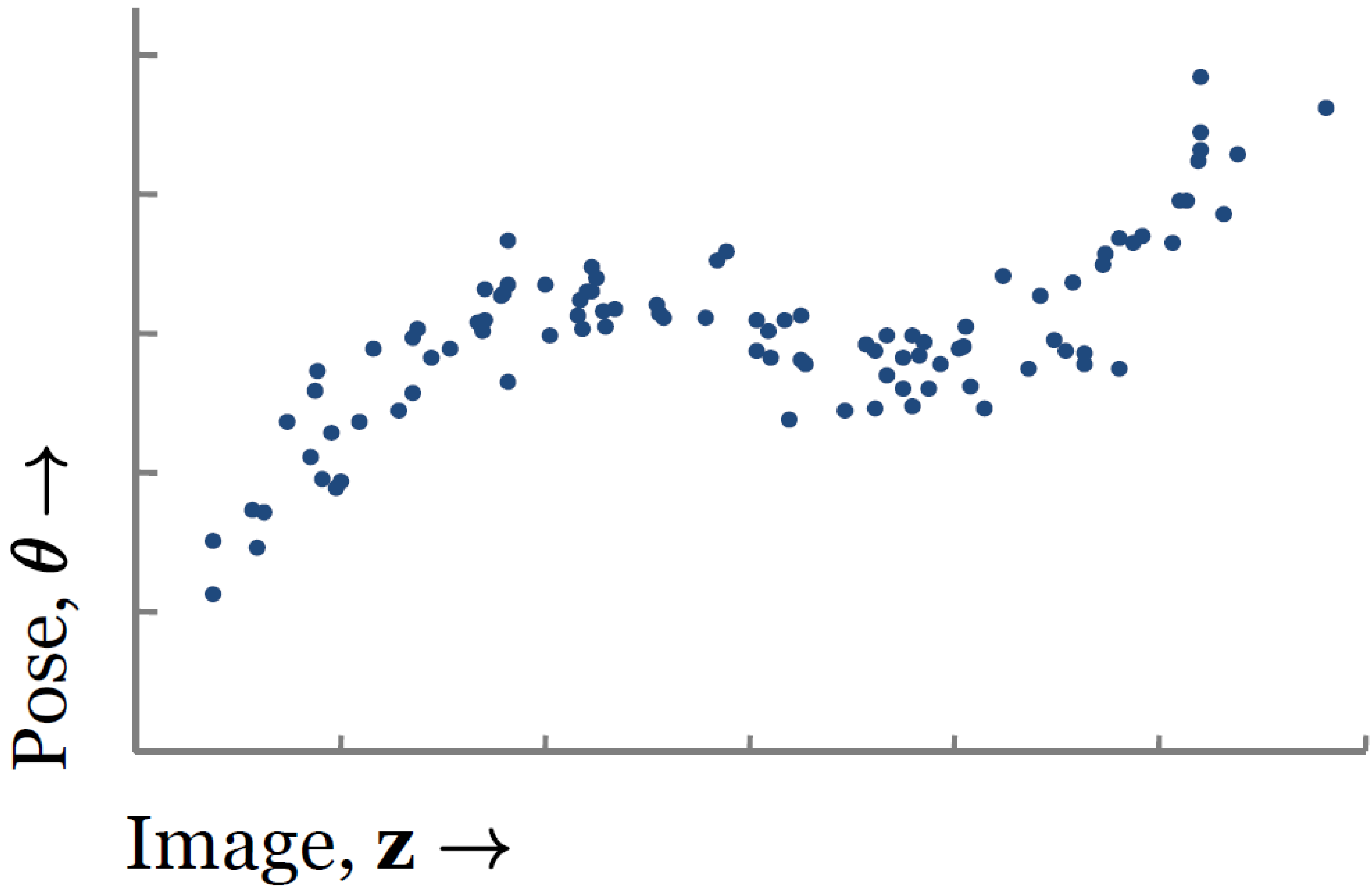
$\mathbf{z}^{\text{new}}$



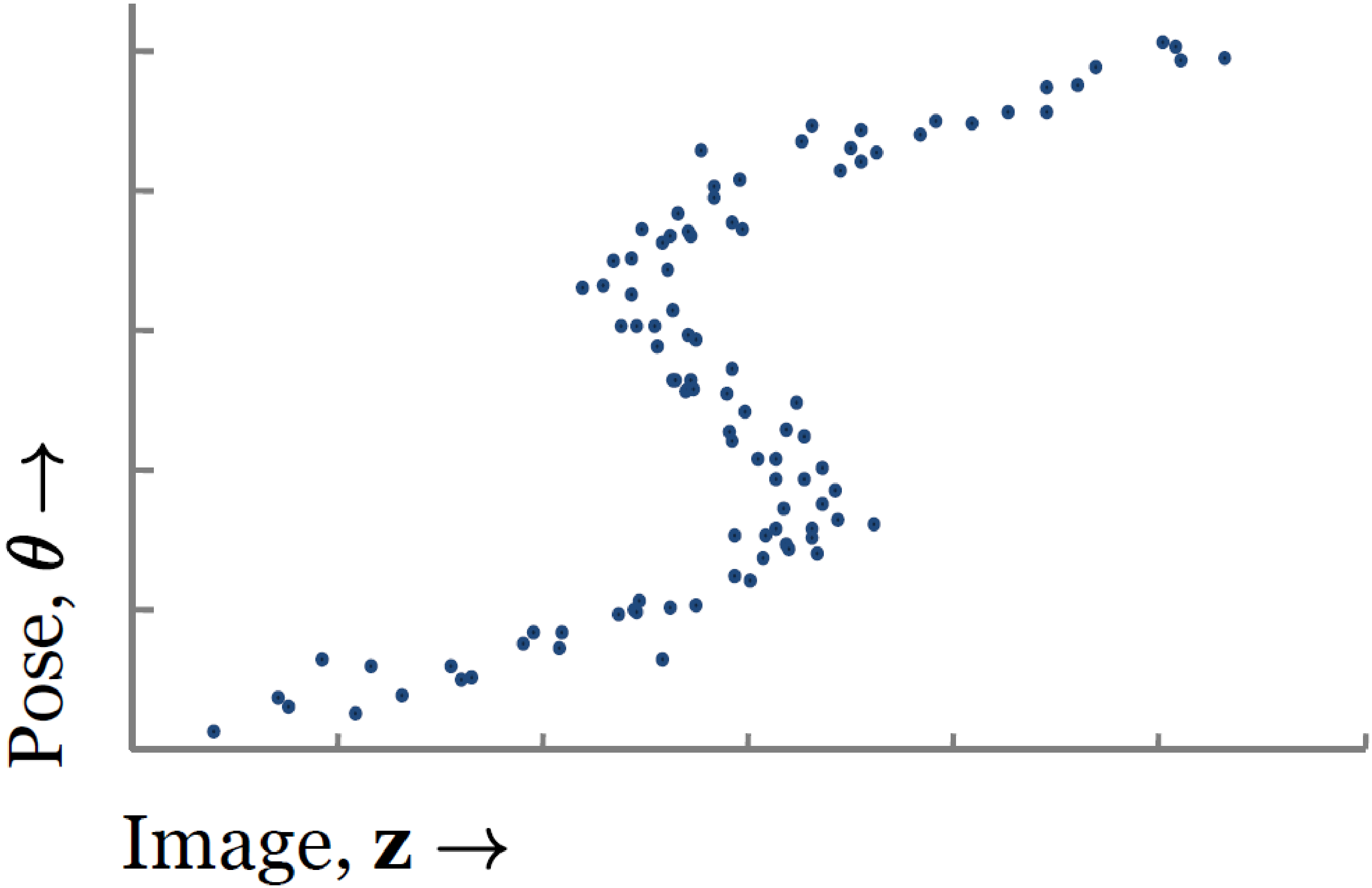
?



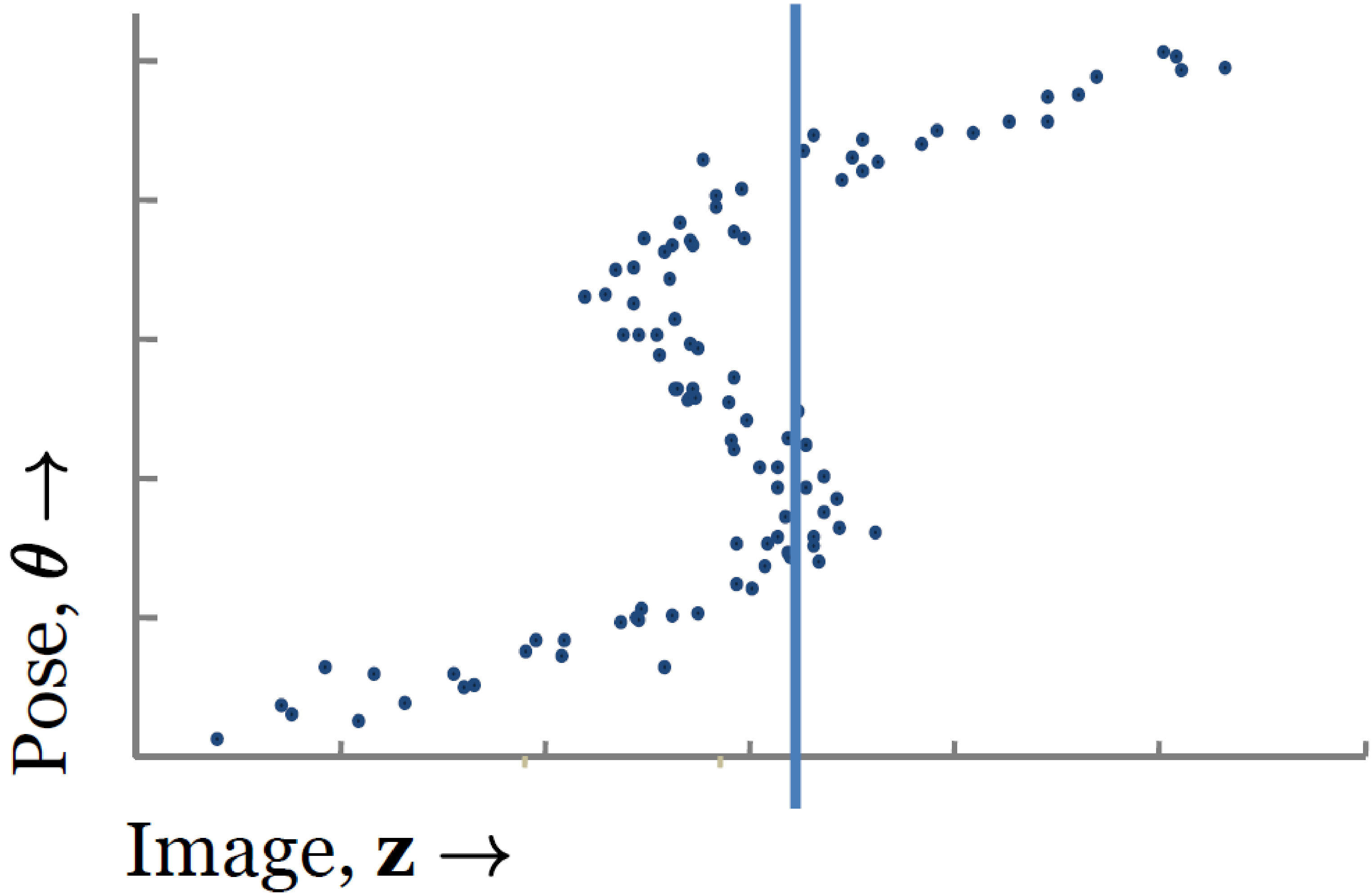
Instead of this:



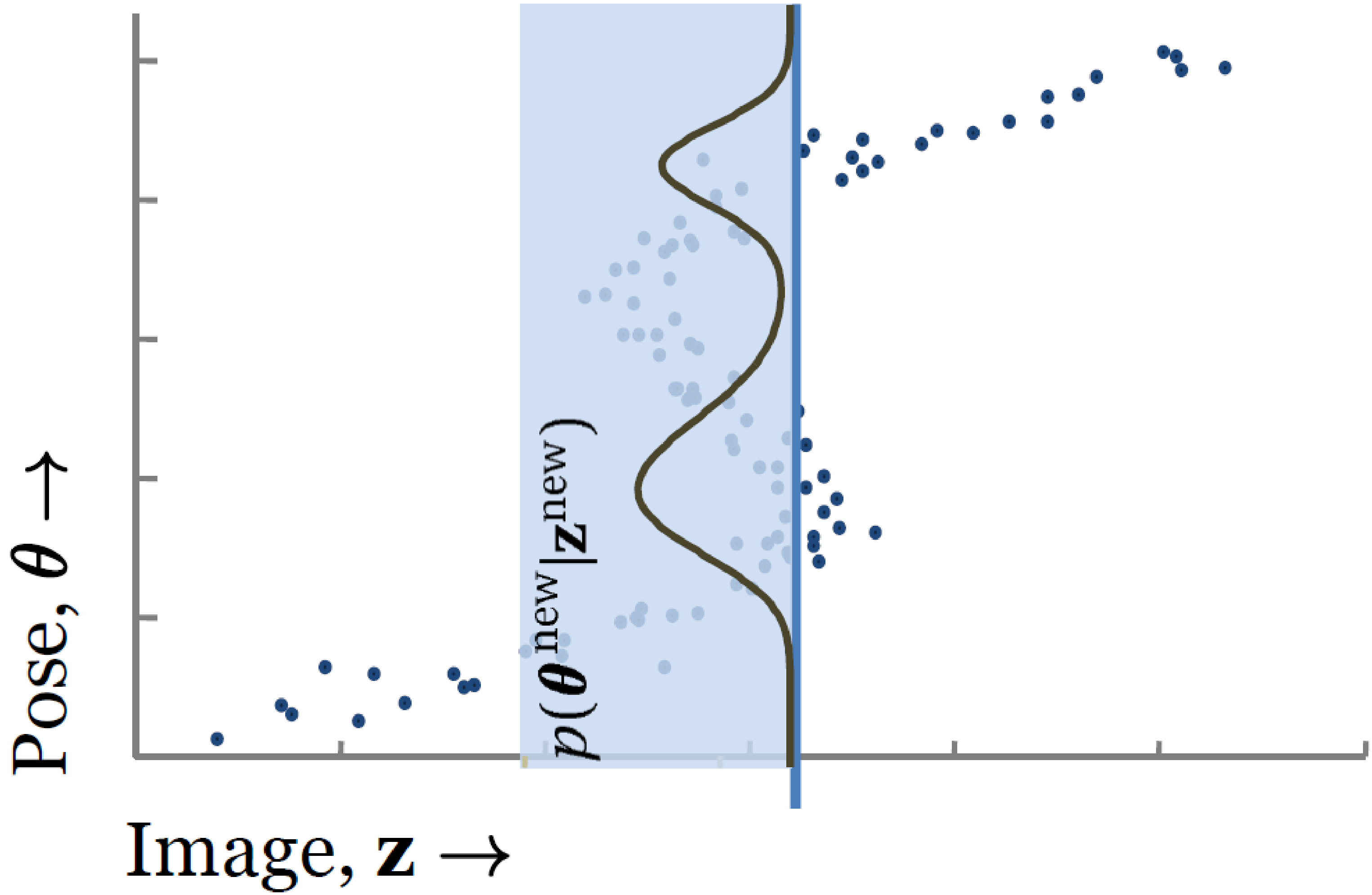
We have this:



We have this:

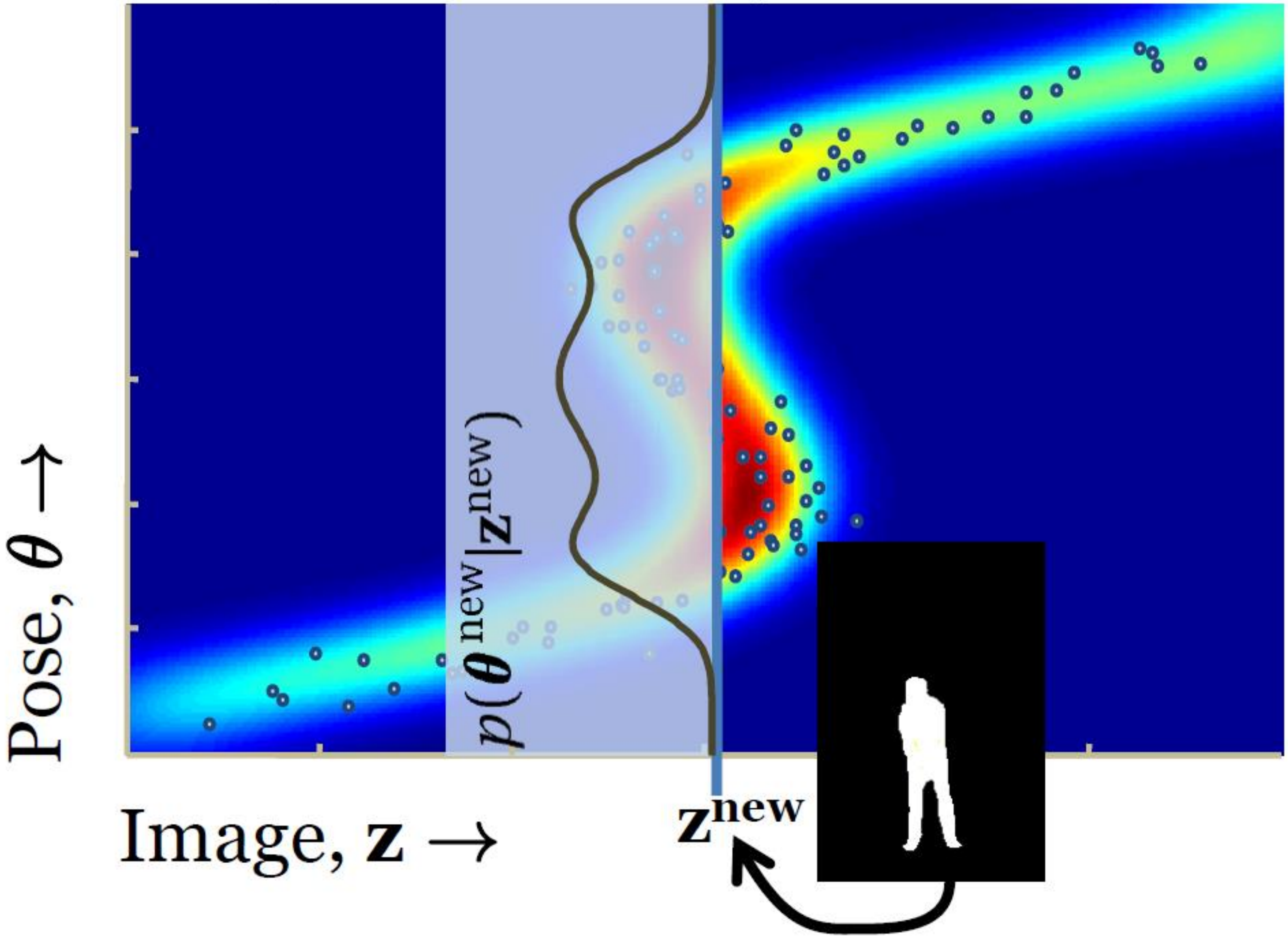


We have this:

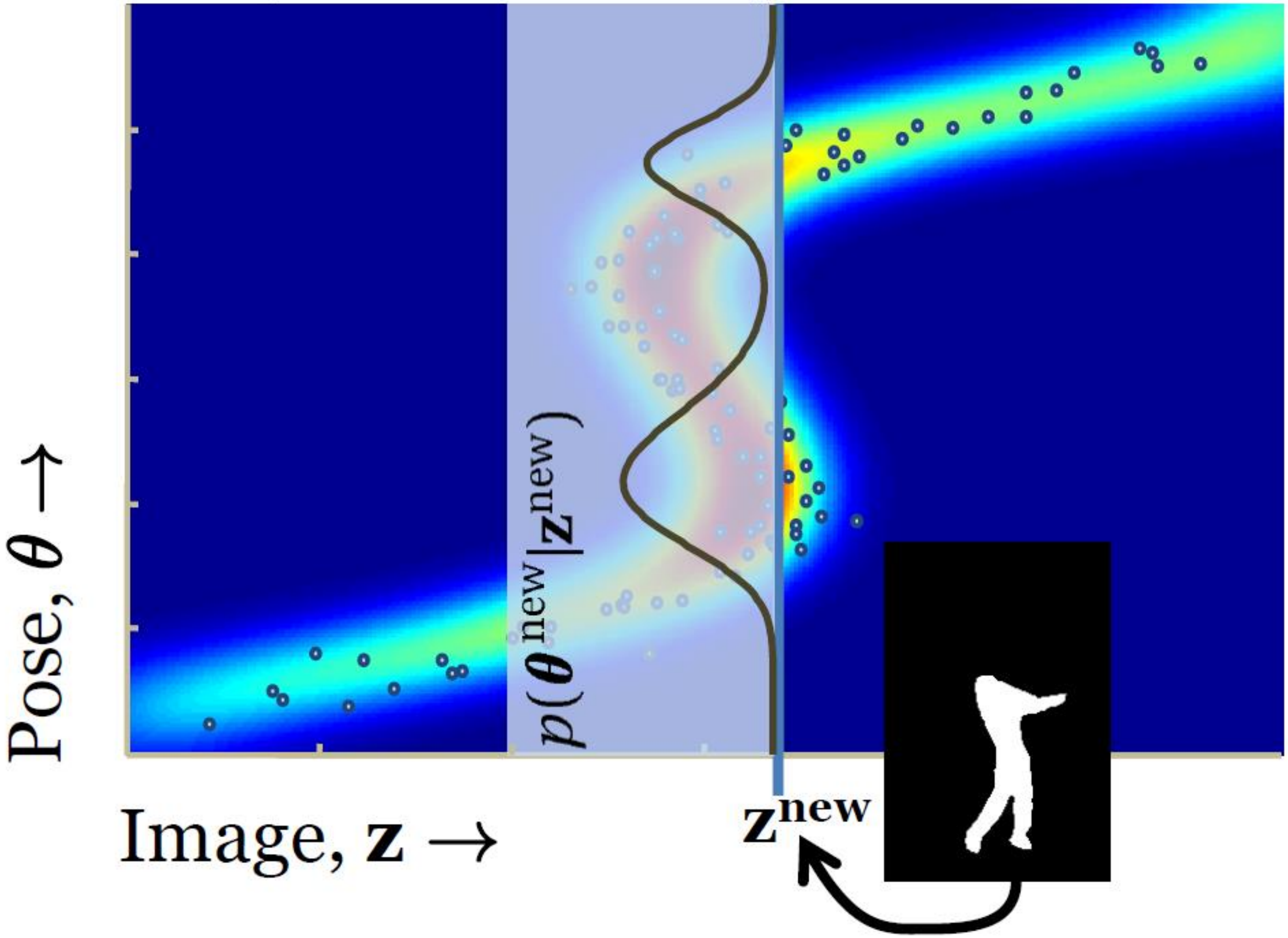


$p(\boldsymbol{\theta}^{new}|\mathbf{z}^{new})$
Pose, $\boldsymbol{\theta} \rightarrow$
Image, $\mathbf{z} \rightarrow$

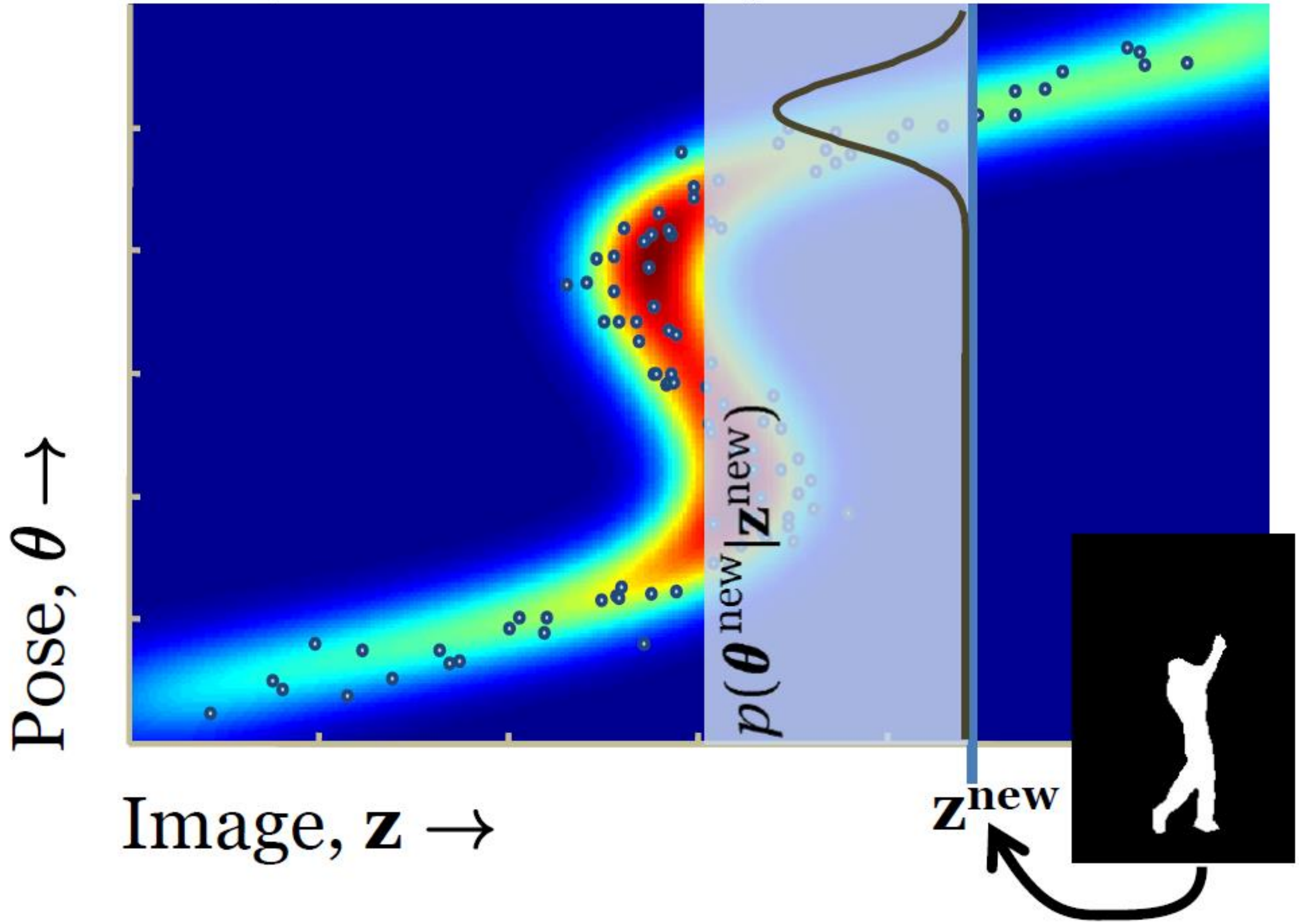
Given new image $\mathbf{z}^{\text{new}}$, conditional $p(\boldsymbol{\theta}|\mathbf{z}^{\text{new}})$ is computed from the joint.

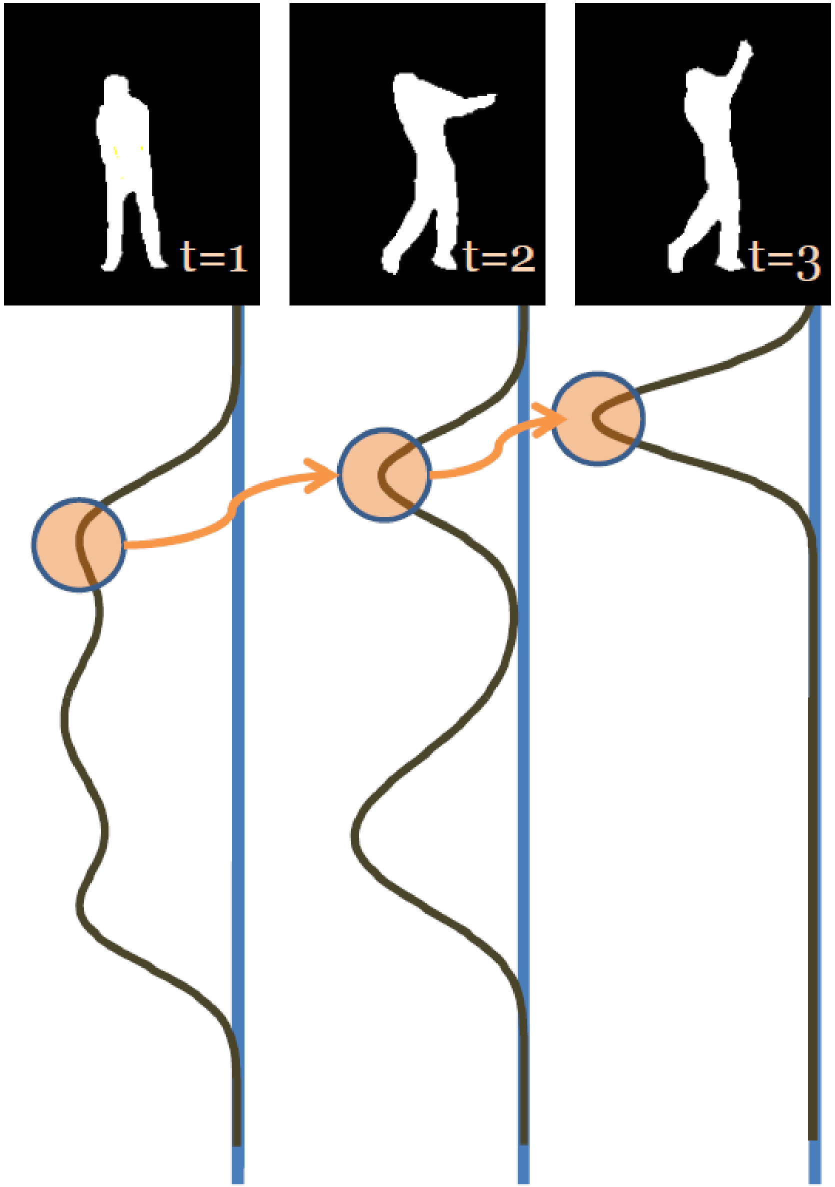


Given new image $\mathbf{z}^{\text{new}}$, conditional $p(\boldsymbol{\theta}|\mathbf{z}^{\text{new}})$ is computed from the joint.



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



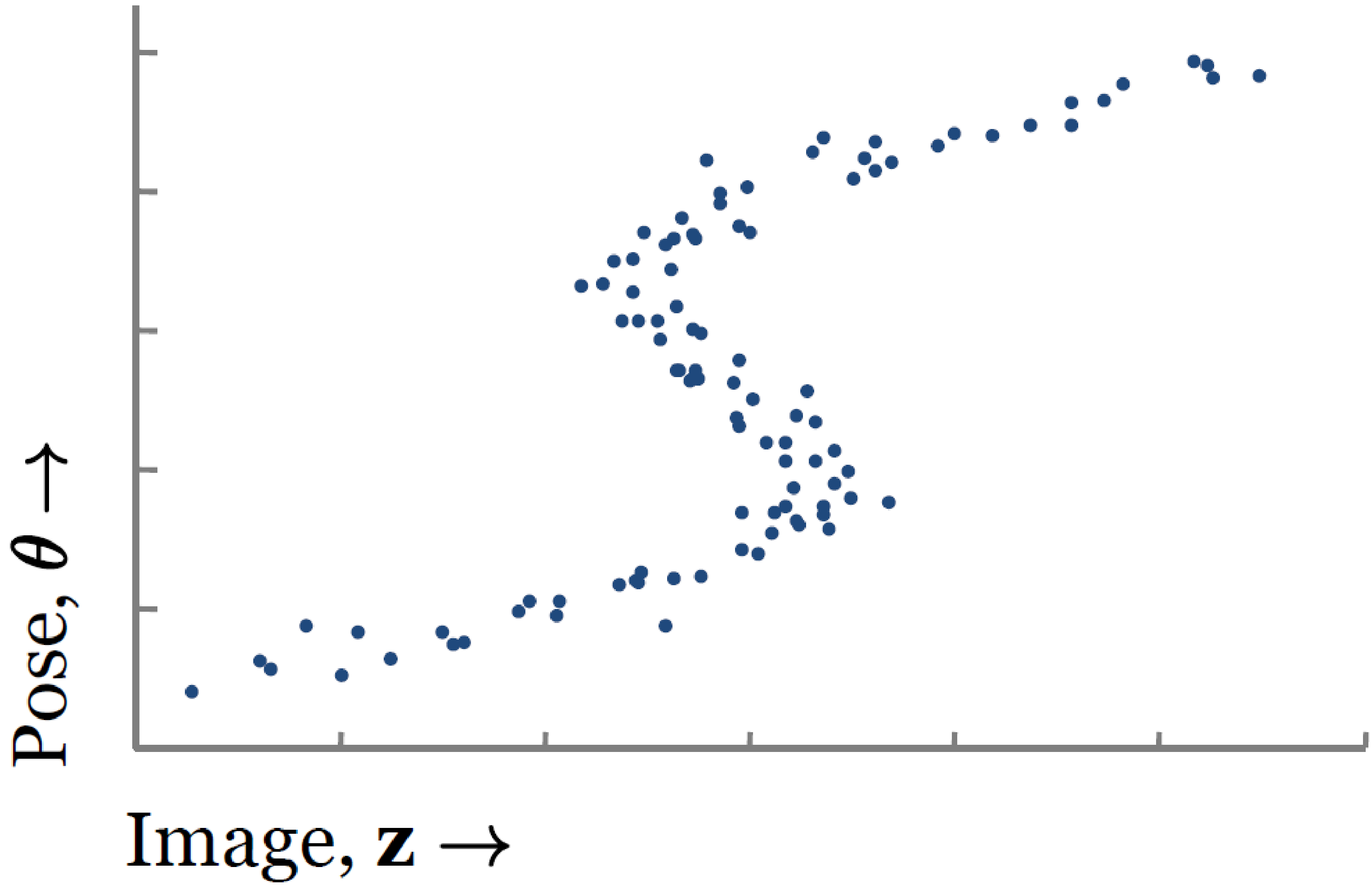


For a video sequence:

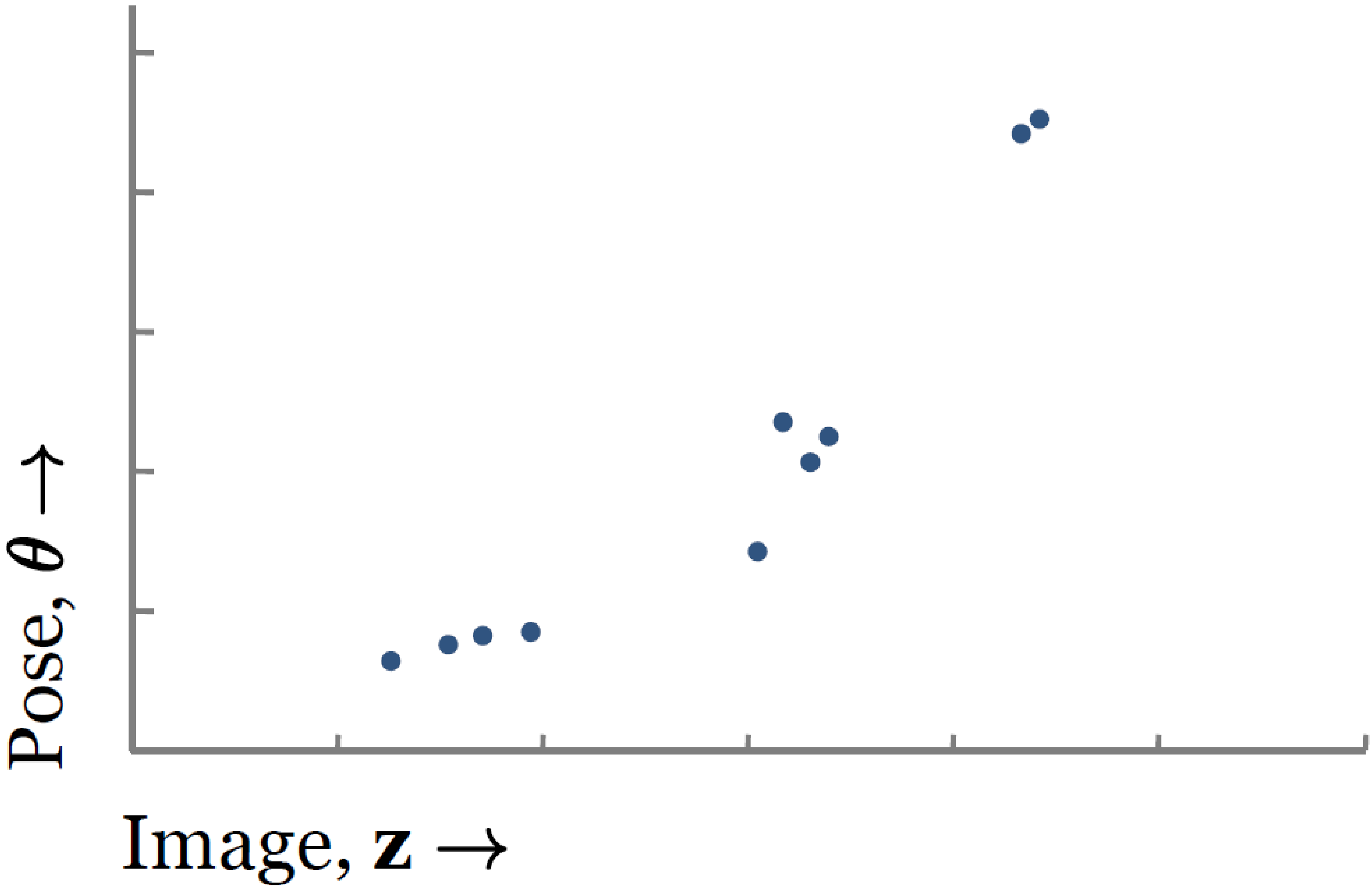
- Compute modes of conditional at every frame
- Choose sequence of modes to maximize product of likelihood and temporal smoothness using Viterbi

But...

Instead of this:

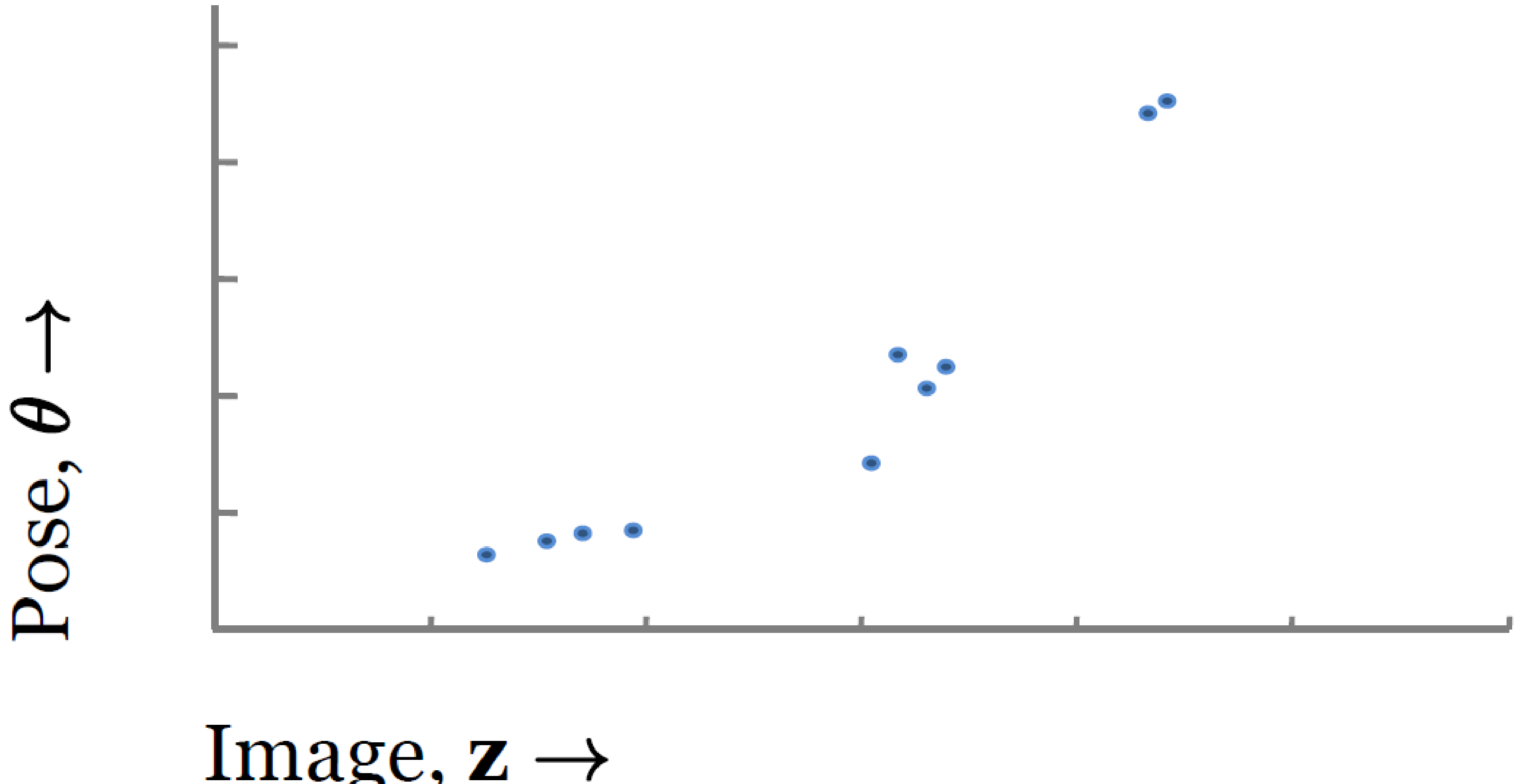


We have this:



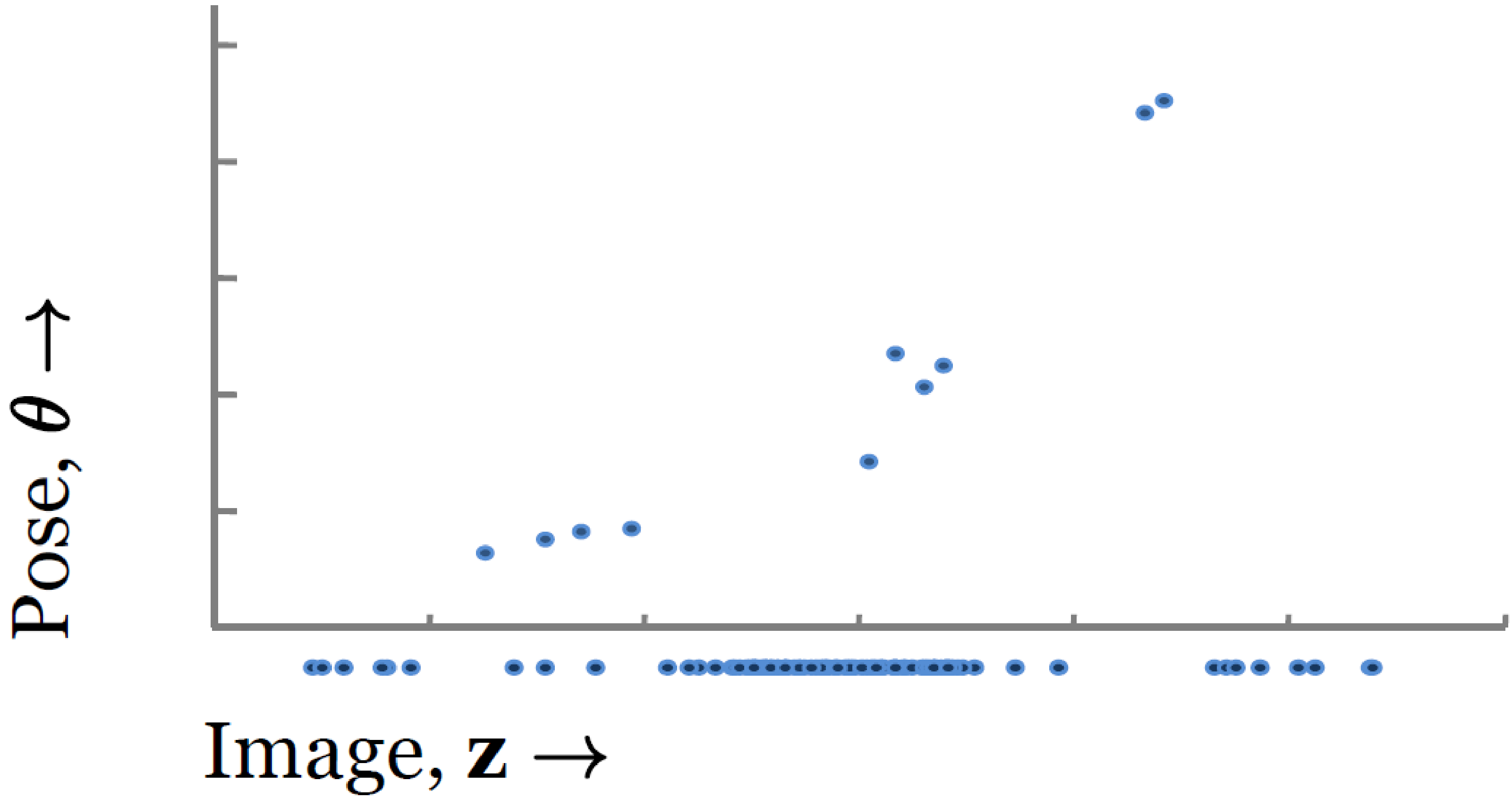
Pose, $\boldsymbol{\theta} \rightarrow$
Image, $\mathbf{z} \rightarrow$

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

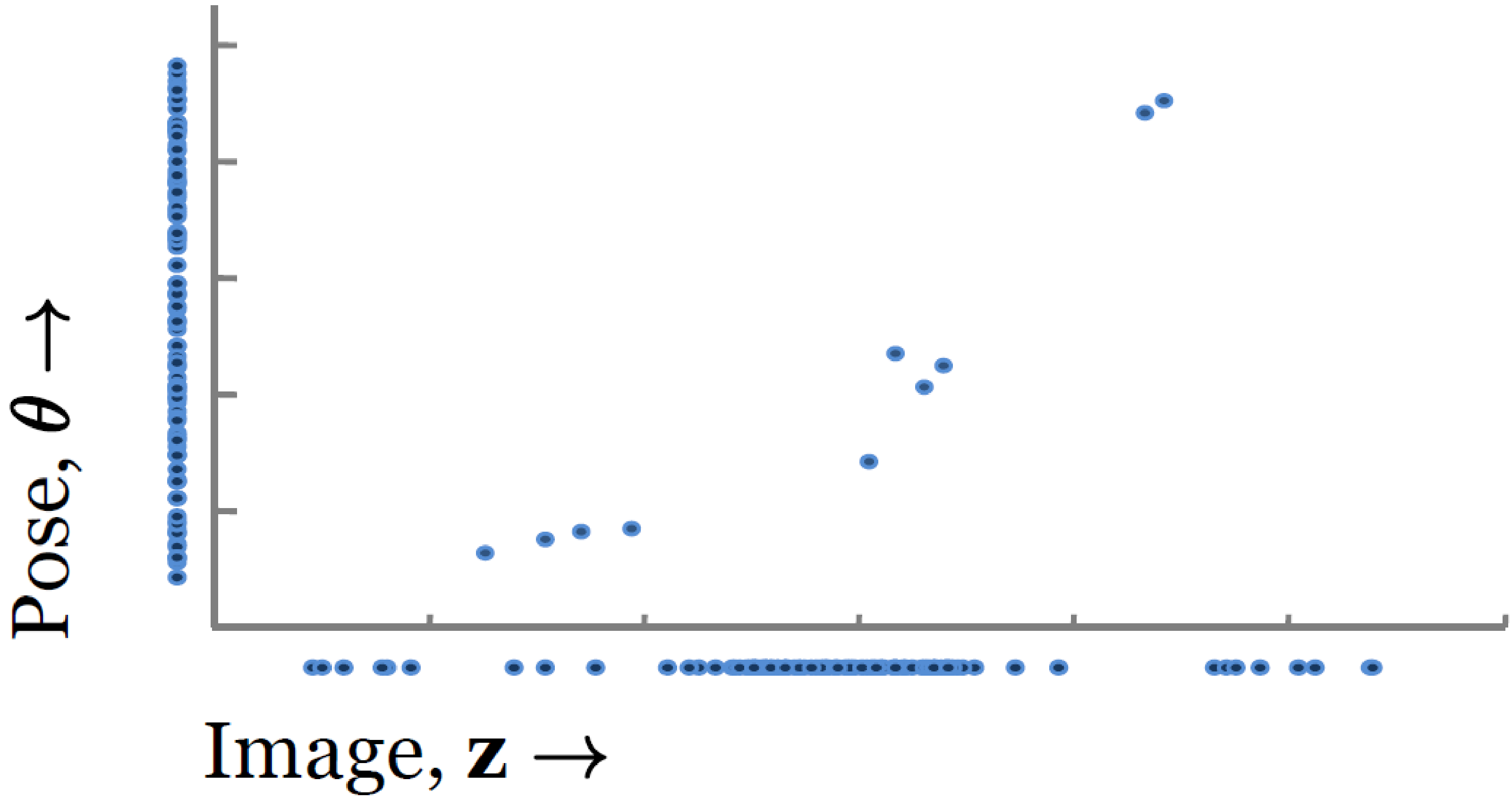


We can't get more because labelling images is expensive .

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

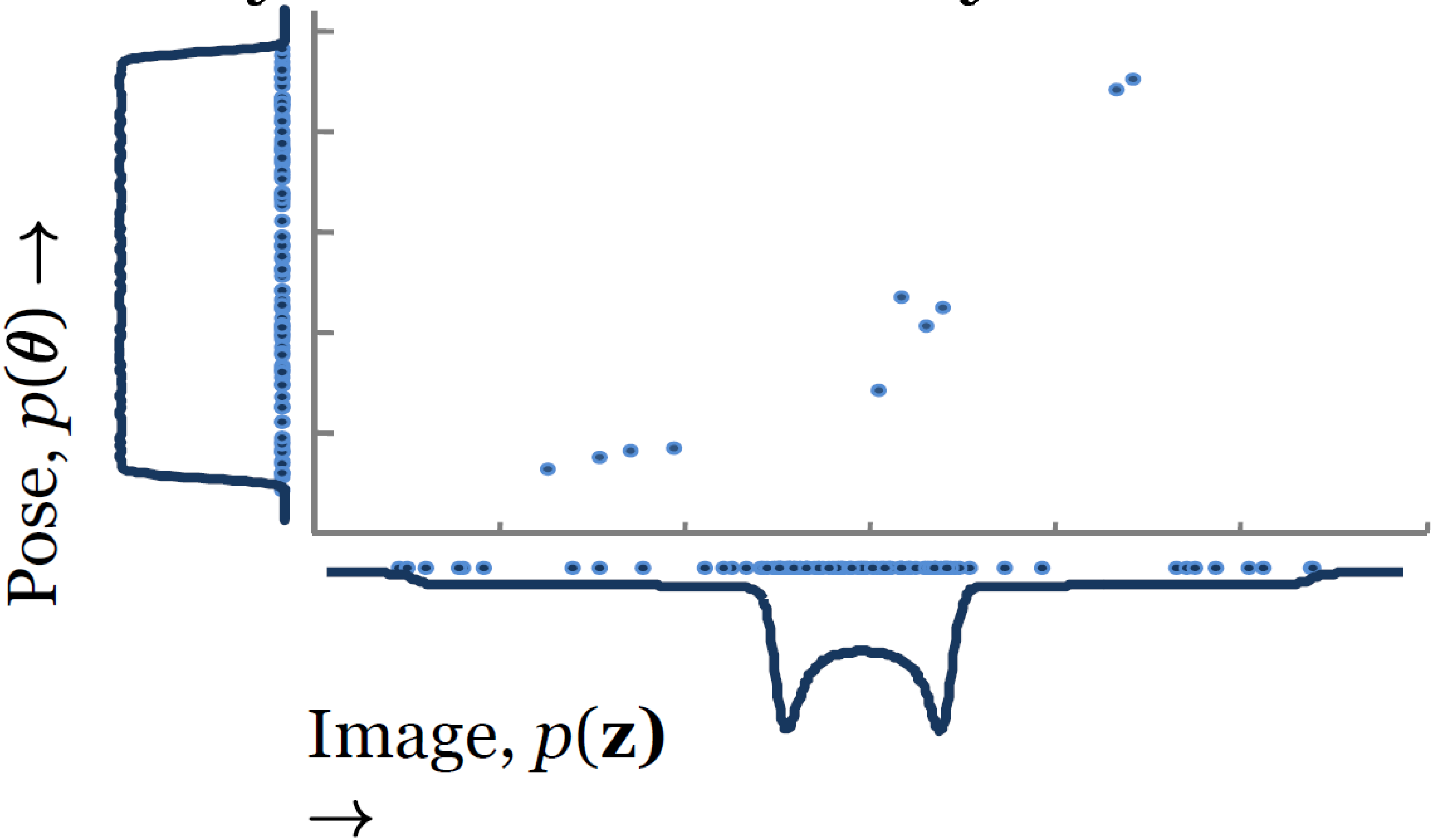


And we can easily download more mocap data without images, i.e. more $(*,\theta)$ pairs

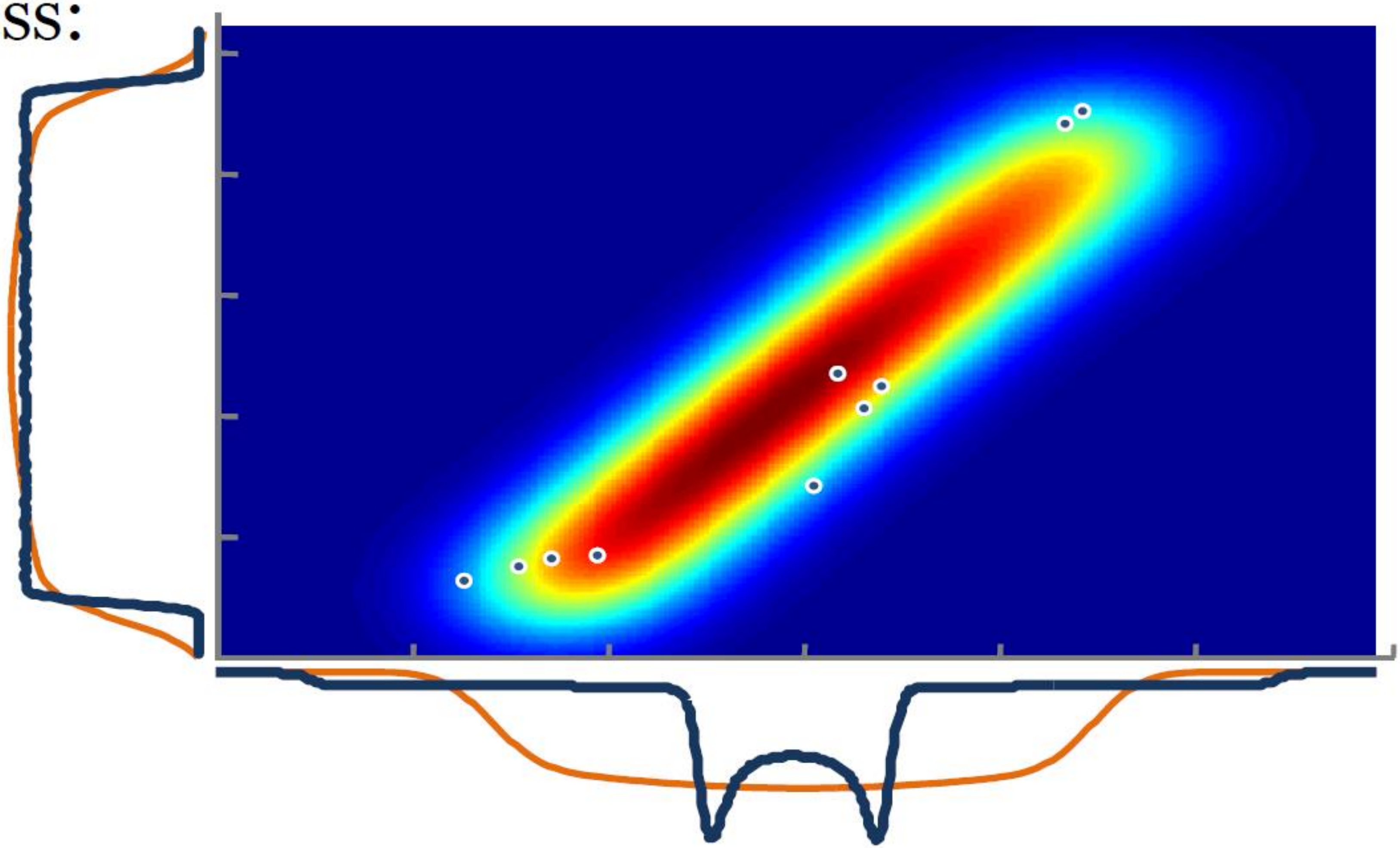


In fact, it's as if we know the **marginals**

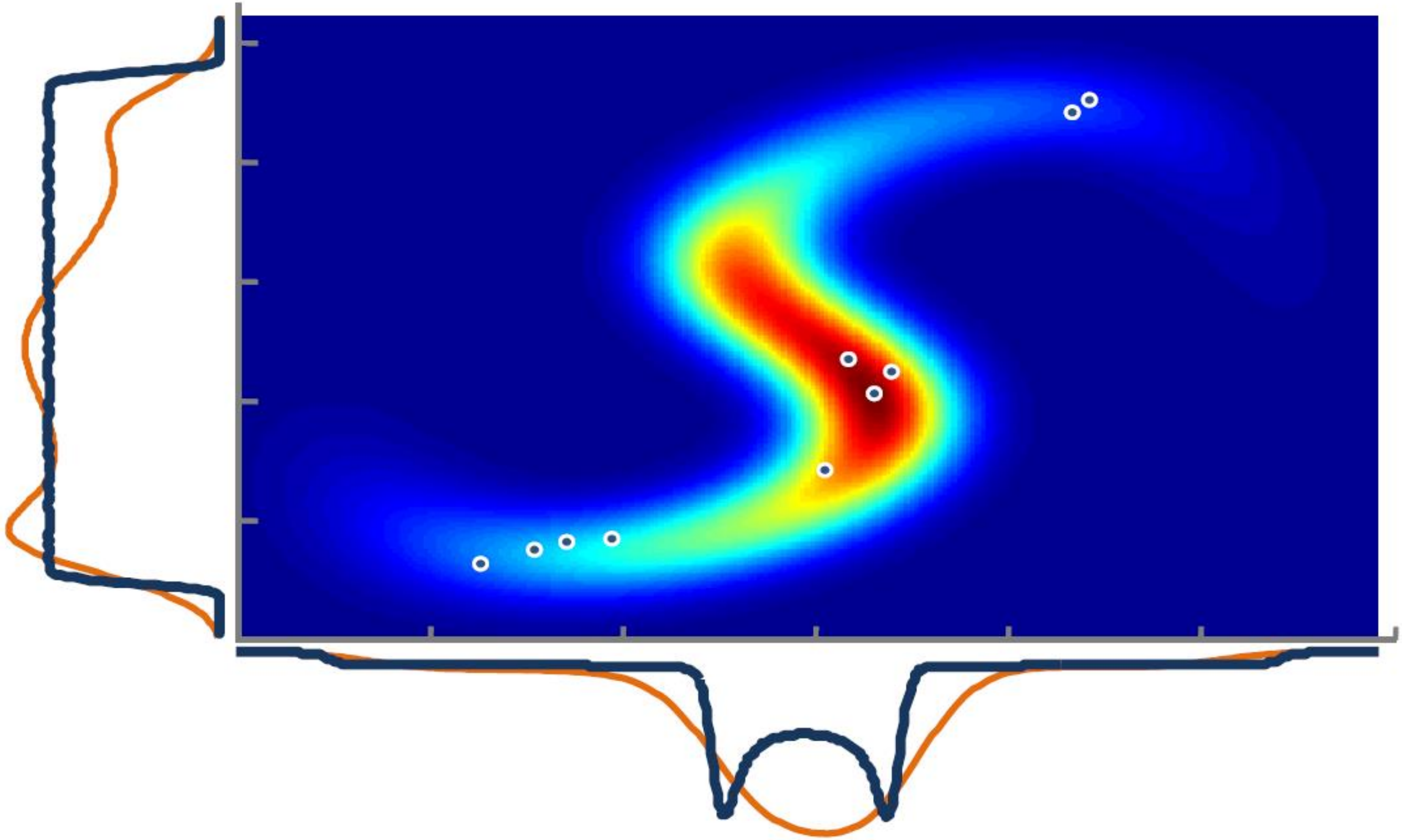
$p(\boldsymbol{\theta}) = \int p(\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z}$ and $p(\mathbf{z}) = \int p(\mathbf{z}, \boldsymbol{\theta}) d\boldsymbol{\theta}$



Which contradict the marginals of our earlier guess:



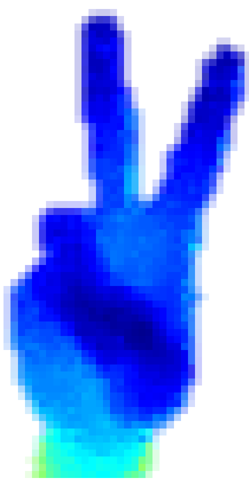
[ffwd] Using the marginal samples gives this:



Hand Pose Estimation

- Given an input depth image, the system yields an output vector of joint angles/locations.
- The joint angles/locations take continuous values, this is formulated as a regression problem.

Input Depth Image



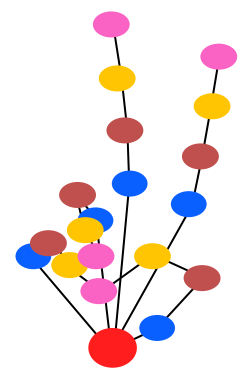
Z

Extract joint angles

$\theta \in \mathbb{R}^d$

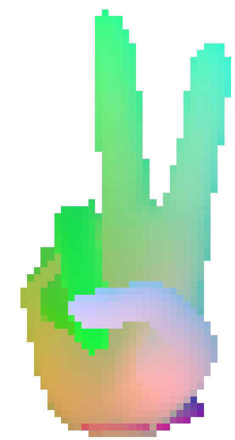
for current frame

Skeleton



θ

Rendered Depth Image



R_θ