Uncertainty estimation with assisted data labelling

Agenda

SPEAKER

Jan Wasilewski

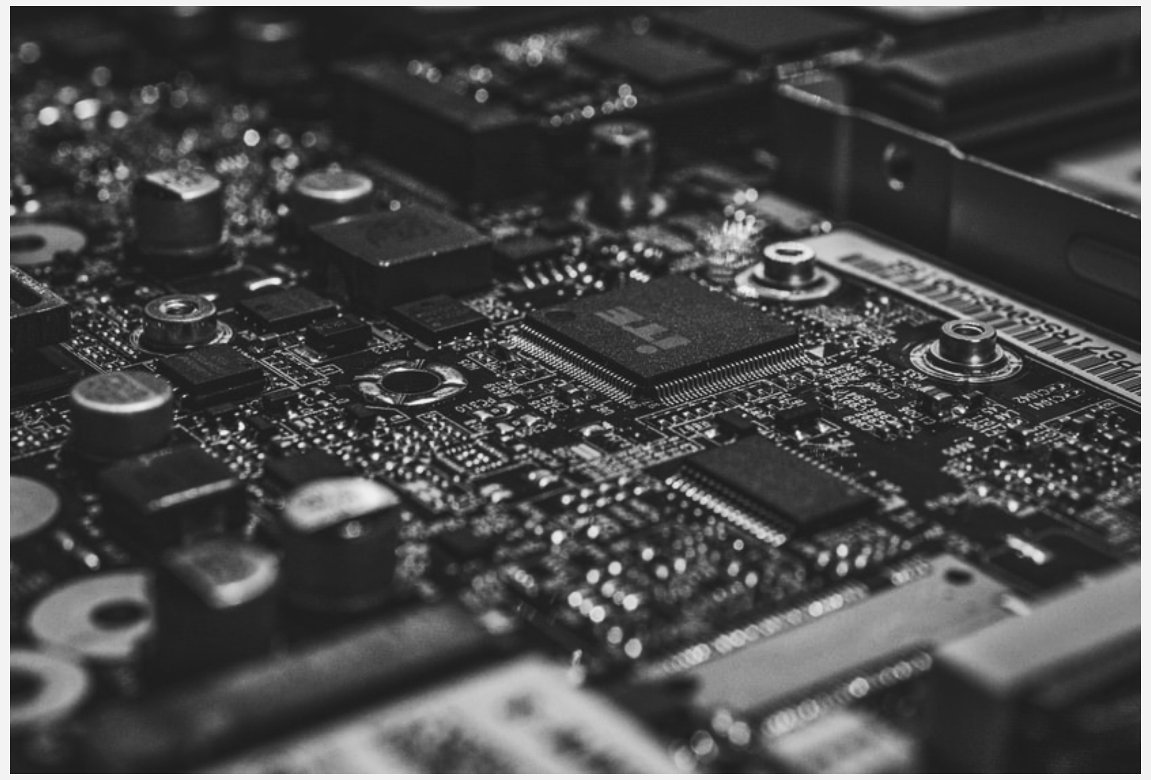
Deep learning and bayesian statistics enthusiast mainly interested in modern solutions.
Skilled in **Statistical Modeling, R, Research, Python and Deep Learning**.

Ex president of math science club
with a Master's degree focused in Mathematics
from Gdańsk University of Technology,
PhD student in Quantum Machine Learning.

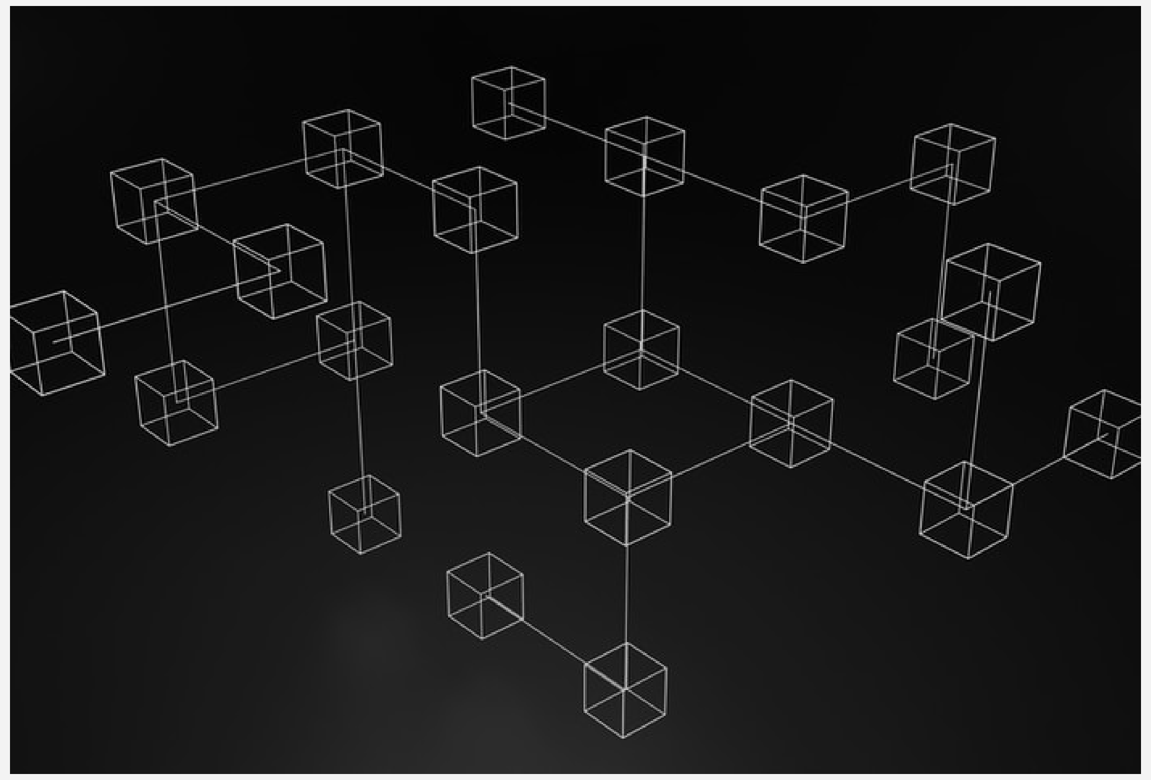
Who needs today yester-day's data?

Lots of data, many models
- big mess and inefficient
annotation process

Analytics computed directly on the edge device (Nvidia Jetson AGX Xavier)



32 GB Memory - Linux environment, Docker environment, 8 services (service for analytics = 8GB)



Model YOLO5M - speed 160fps + video encoding-decoding on GPU

Conclusion: not enough space to store data, bigger models would be too slow!

Problem:

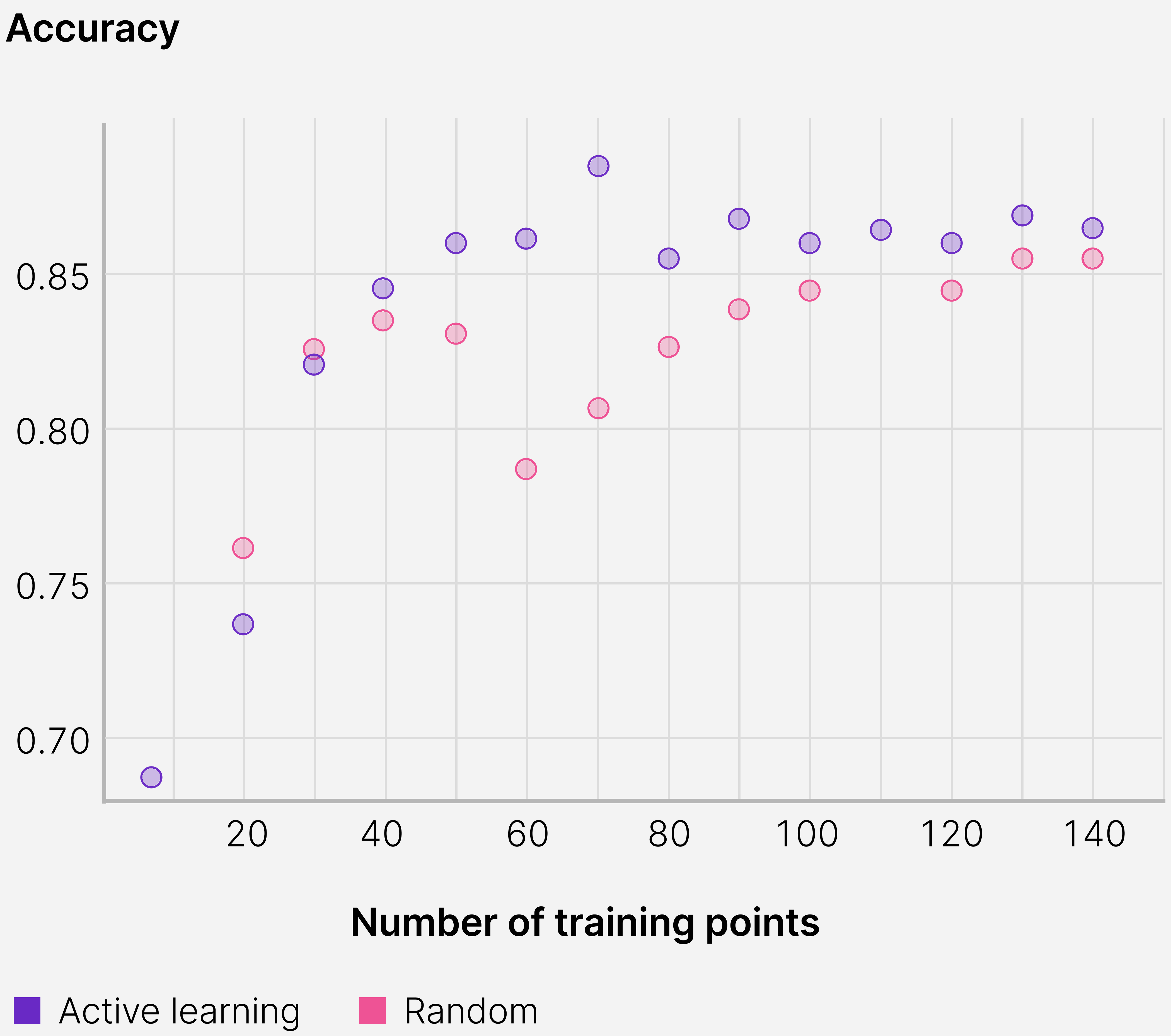
Lack of training data! – but we have access to a lot of unannotated data

A lot of time necessary to annotate!

Our response:

SPIN 💪

Active learning



Acquisition functions need information about uncertainty of predictions.

How to calculate them?

Other cases where uncertainty is important:

- Safety-critical systems (autonomous driving/medical applications)
- Bayesian Optimization

Other benefits from active learning:

- Reduced costs and time of expensive data annotation process
- Reduced time of model training

Statistical view on learning theory:

$$y_i = NN_W(x_i) + \epsilon_i$$

$$p(y_i | x_i, W) = N(NN_W(x_i), \sigma)$$

$$p(y | x, W) = \prod_{i=1}^{N} p(y_i | x_i, W)$$

Learning classical models

– minimization of the loss function using some stochastic descent algorithm

$$\arg\max_W p(y|x, W) = \arg\max_W \prod_{i=1}^{N} p(y_i|x_i, W) =$$

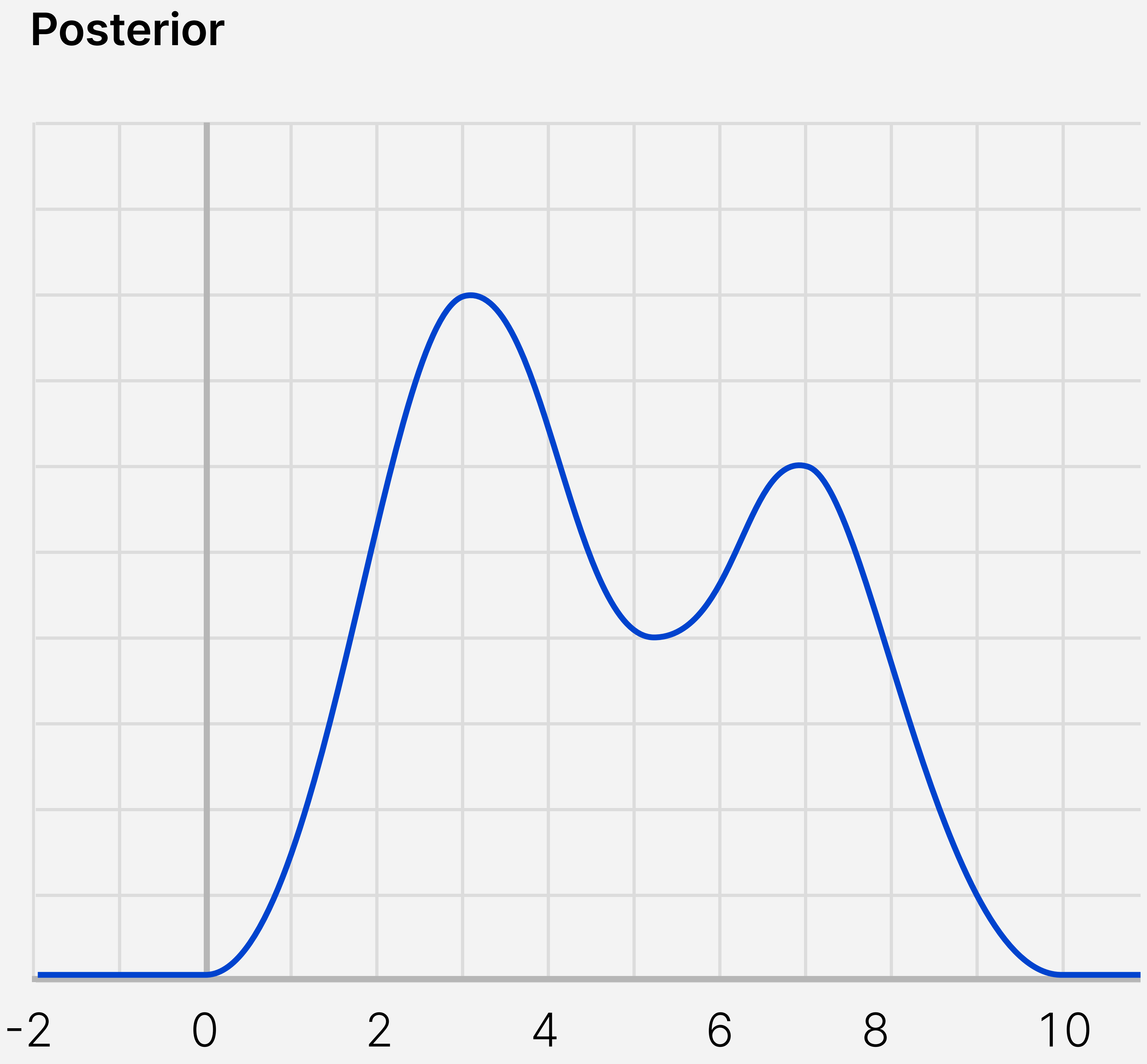
$$\arg\max_W \prod_{i=1}^{N} \exp\left\{-\frac{(NN_W(x_i) - y_i)^2}{2\sigma^2}\right\} = \arg\max_W \exp\left\{-\sum_{i=1}^{N} \frac{(NN_W(x_i) - y_i)^2}{2\sigma^2}\right\} =$$

$$\arg\max_W -\sum_{i=1}^{N}(NN_W(x_i) - y_i)^2 = \arg\min_W \sum_{i=1}^{N}(NN_W(x_i) - y_i)^2$$

Bayesian point of view:

$$p(W|D) = \frac{p(y|x, W)p(W)}{p(D)}$$

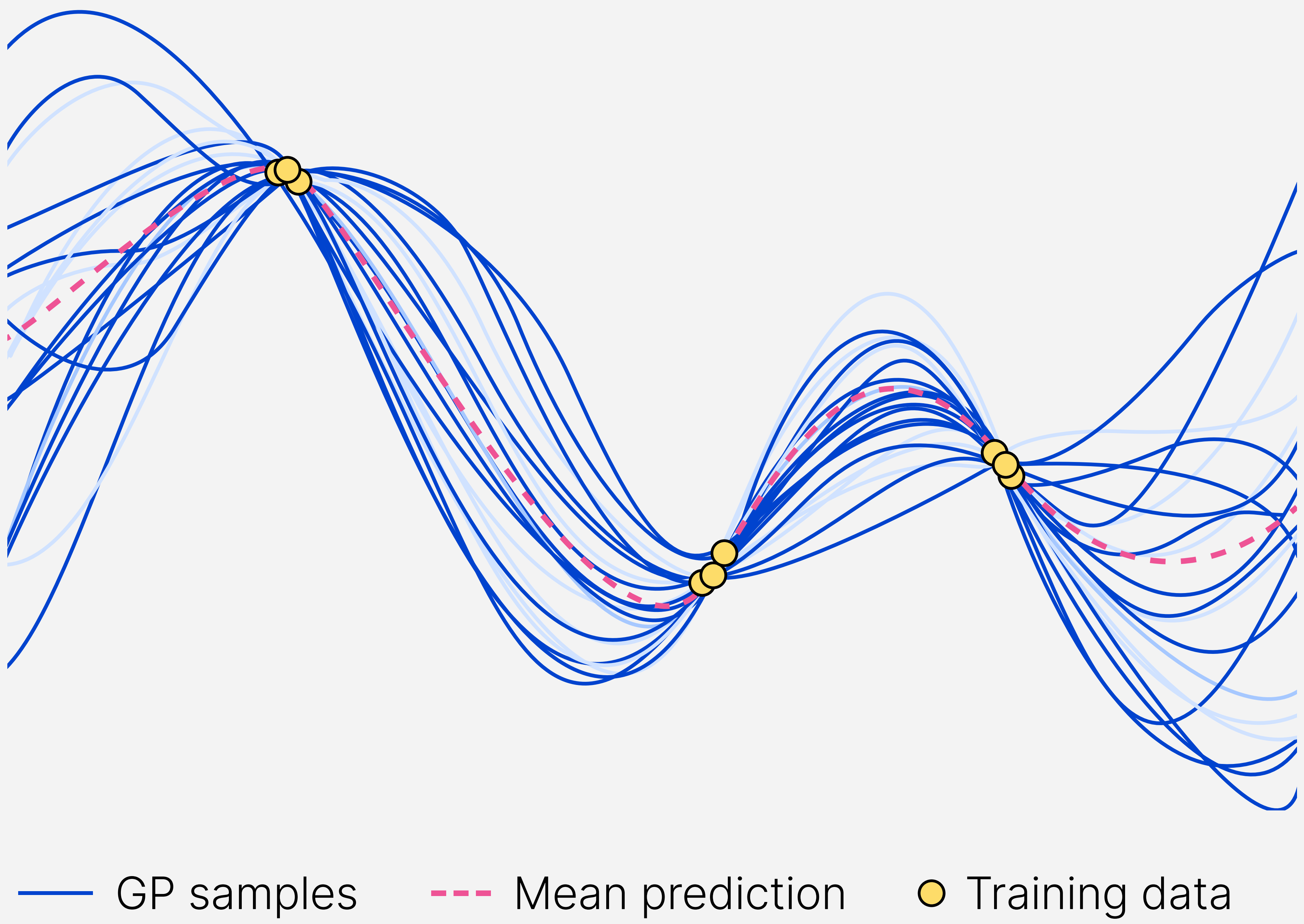
Evolution of weights distribution



How to implement BNN's in practice?

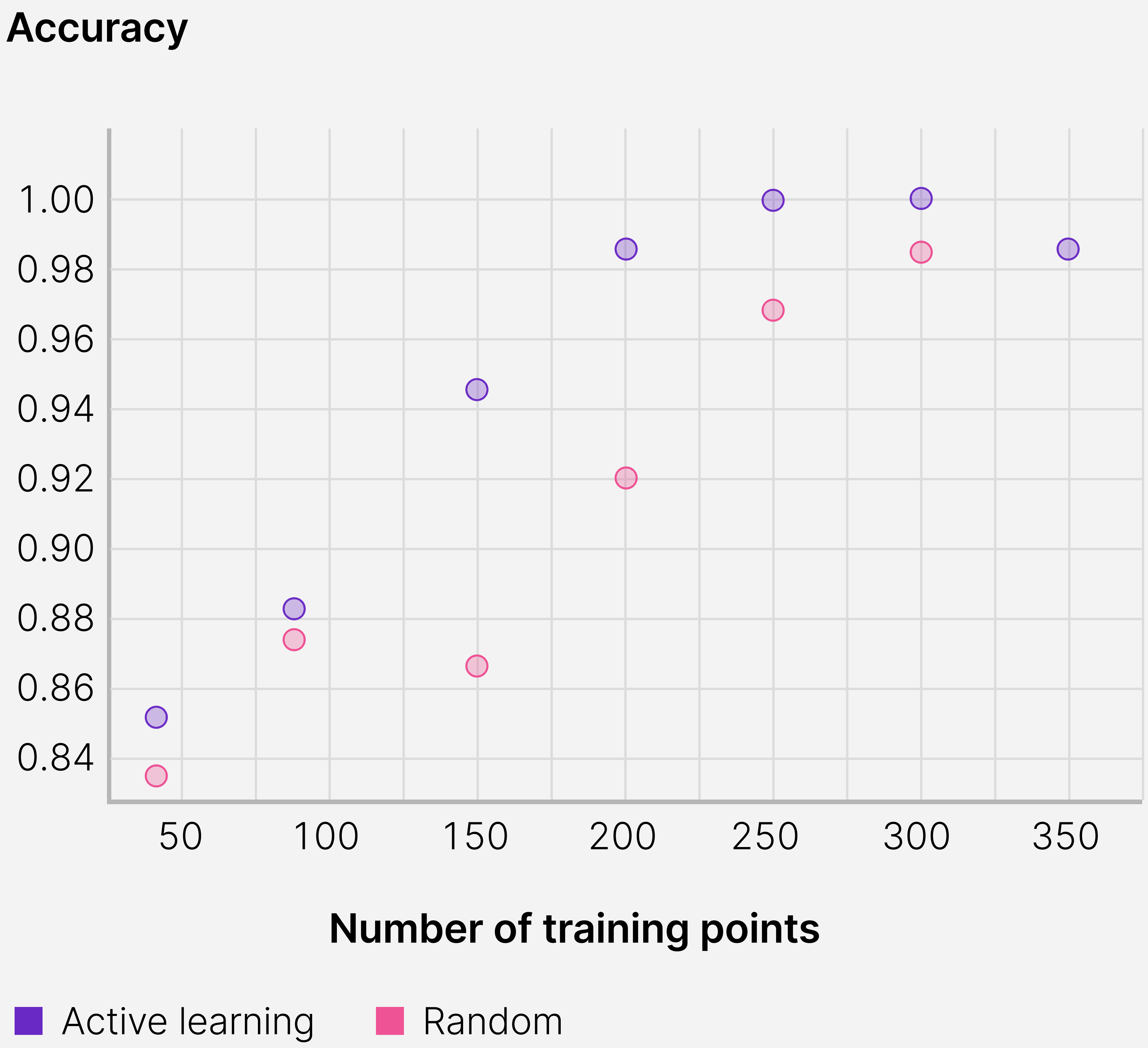
- Laplace
- MCMC
- Variational Inference

Bayesian neutral networks



Laplace's approximation - code example

```
def train_bnn(train_dataloader, my_nn):
    la = Laplace(my_nn, 'classification',
                 subset_of_weights='all',
                 hessian_structure='full')
    la.fit(train_dataloader)
    return la
```

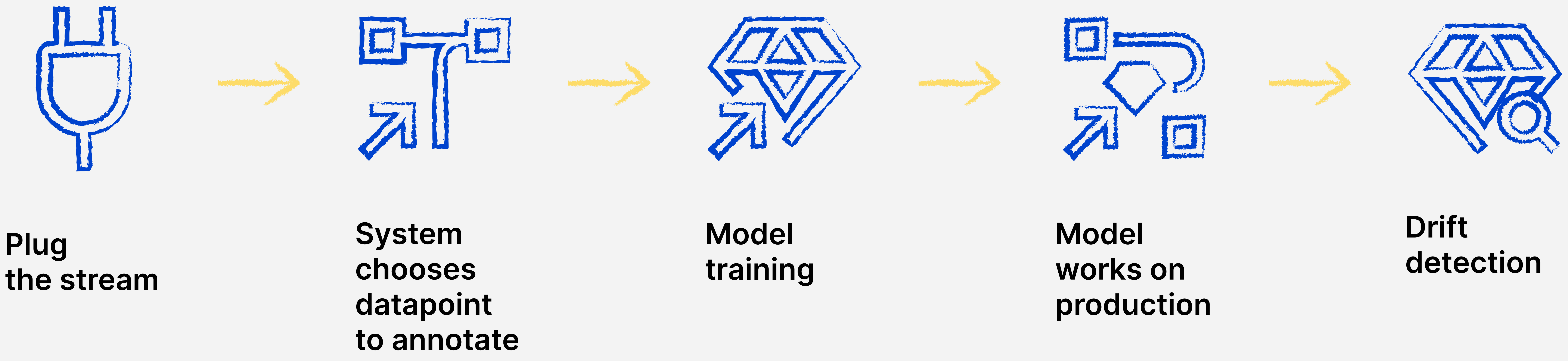


Back to our idea – another features:

- Multimodal annotation
- Repository of models
- Stream data
- Generative methods
- Concept drift detection
- Live quality control

Use case

Automatization of quality control



Project is in its infancy
— your feedback is crucial!

→ What use-cases are in your companies?

→ What constraints in usage of SPIN system in your company do you see?

We are open for cooperation

Let’s talk!

Jan
Wasilewski

Tomasz
Ludwisiak

Diana
Bogusz

Get in touch!