COMBINING MACHINE LEARNING AND REASONING TASKS

Harri Valpola, CEO
Curious AI
2019-05-09

Prediction and Control

CURIOUS AI

Case 1: Oil refinery, Neste Engineering Systems

Sandvik LH410 — A loader for underground mining and tunneling


SANDVIK
CURIOUS AI
AILiveSim
15
STOP


SANDVIK
CURIOUS AI
AILiveSim
The Curious AI Autonomous Control learns to drive the machine

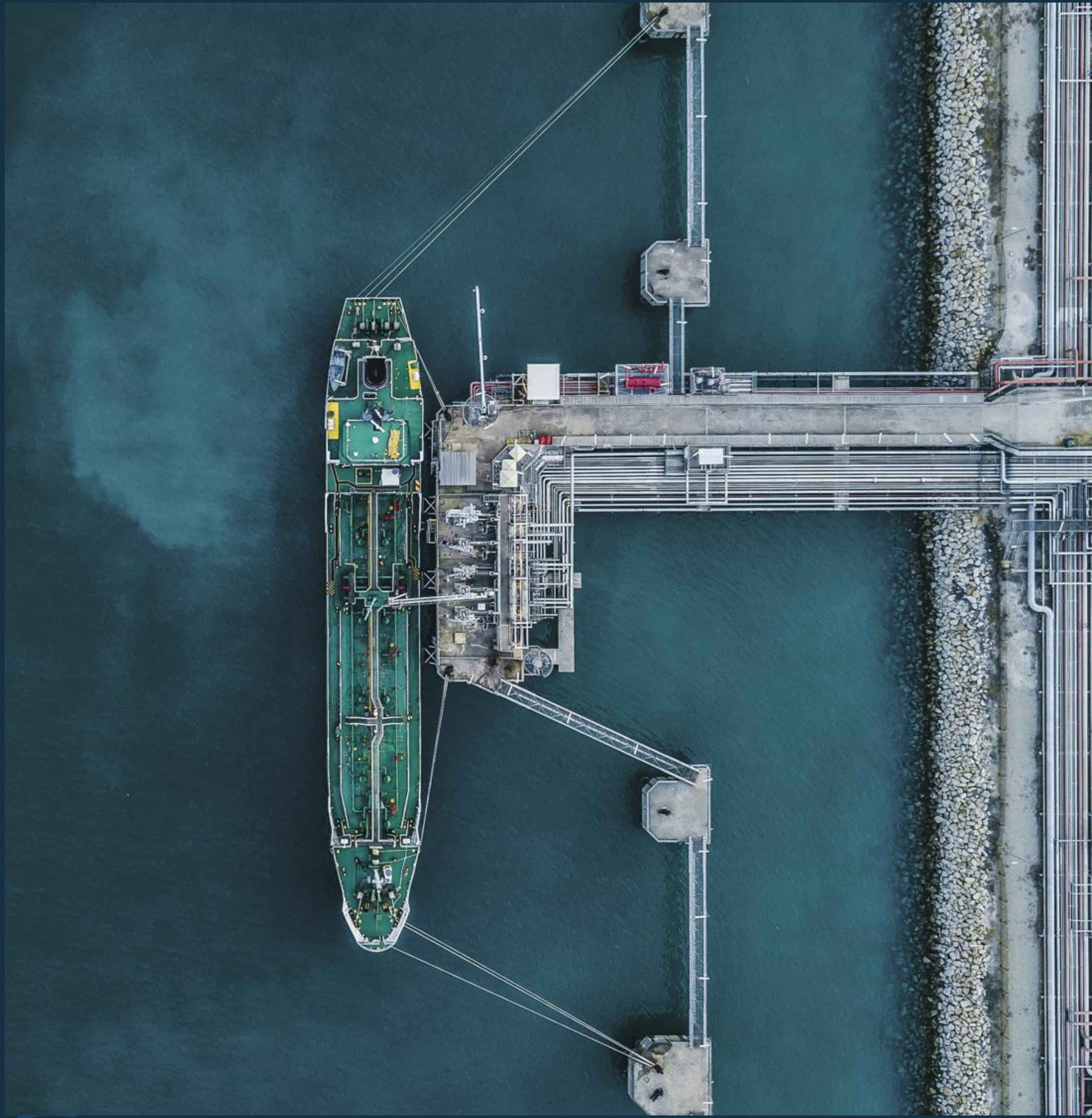
Overview

01 MODEL-BASED RL

02 WHY IS IT HARD TO USE LEARNED MODELS?

03 THE SOLUTION

04 RELEVANCE TO MATERIAL SCIENCES

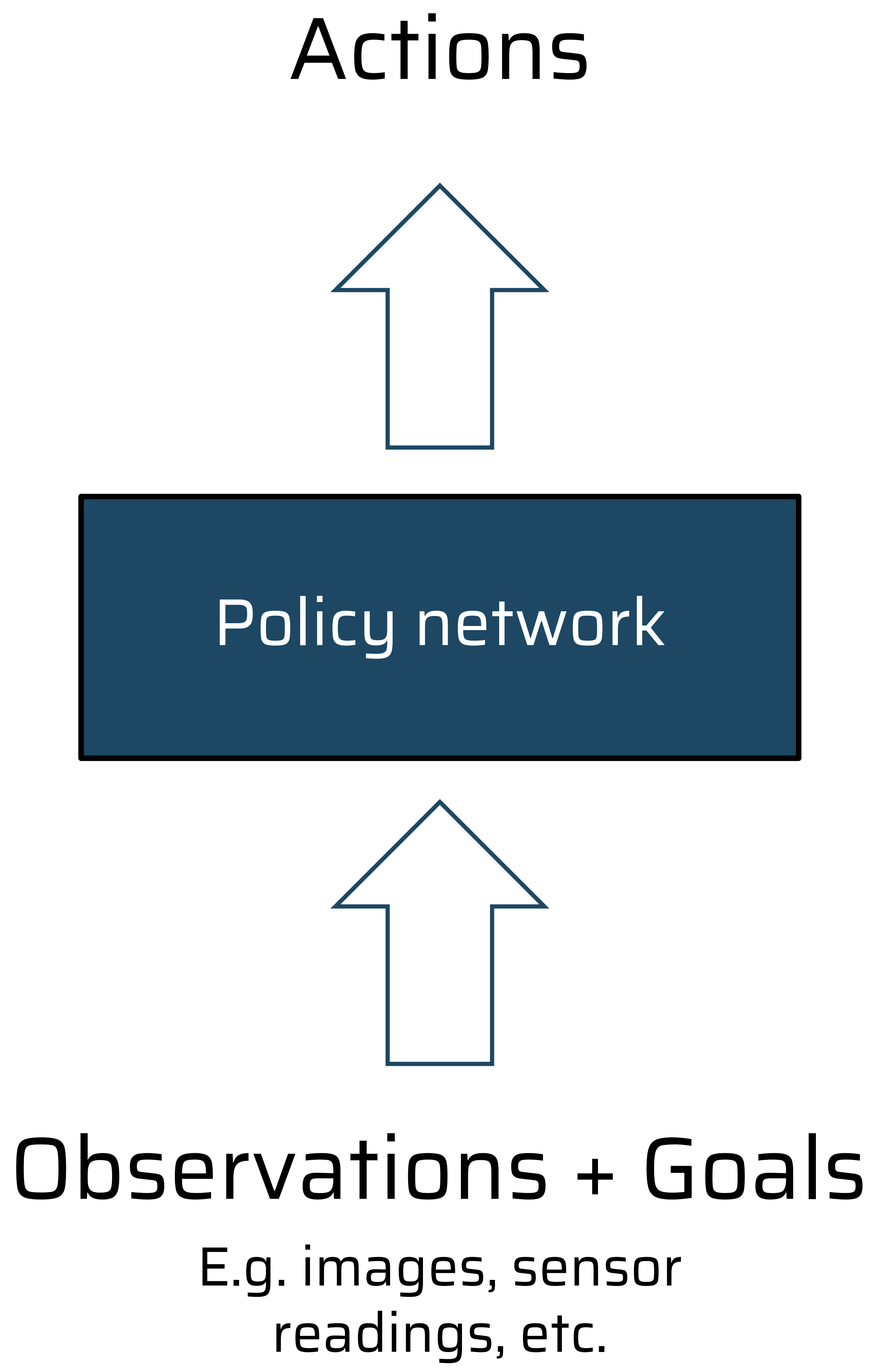


01 MODEL-BASED REINFORCEMENT LEARNING

Model-Free RL

Policy network

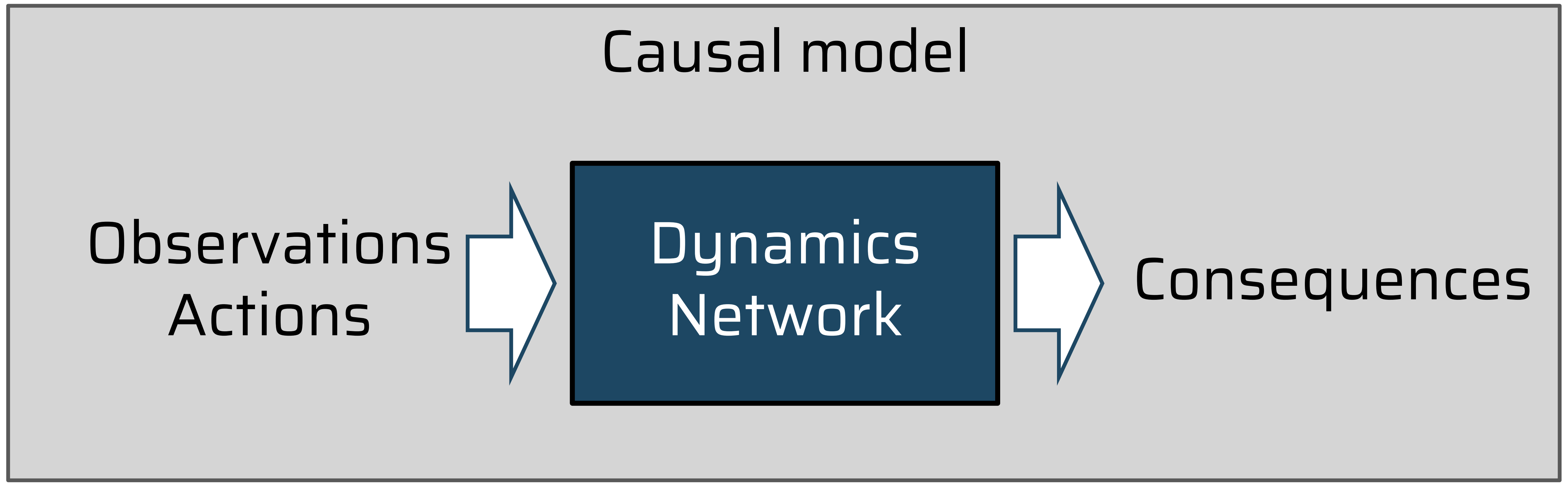
- Slow to learn (trial and error or imitation)
- Fast in operation



Model-Based RL

Model the underlying causal process

- needs much less training data
- applicable to new situations
- offers explanations, can answer “what if” questions



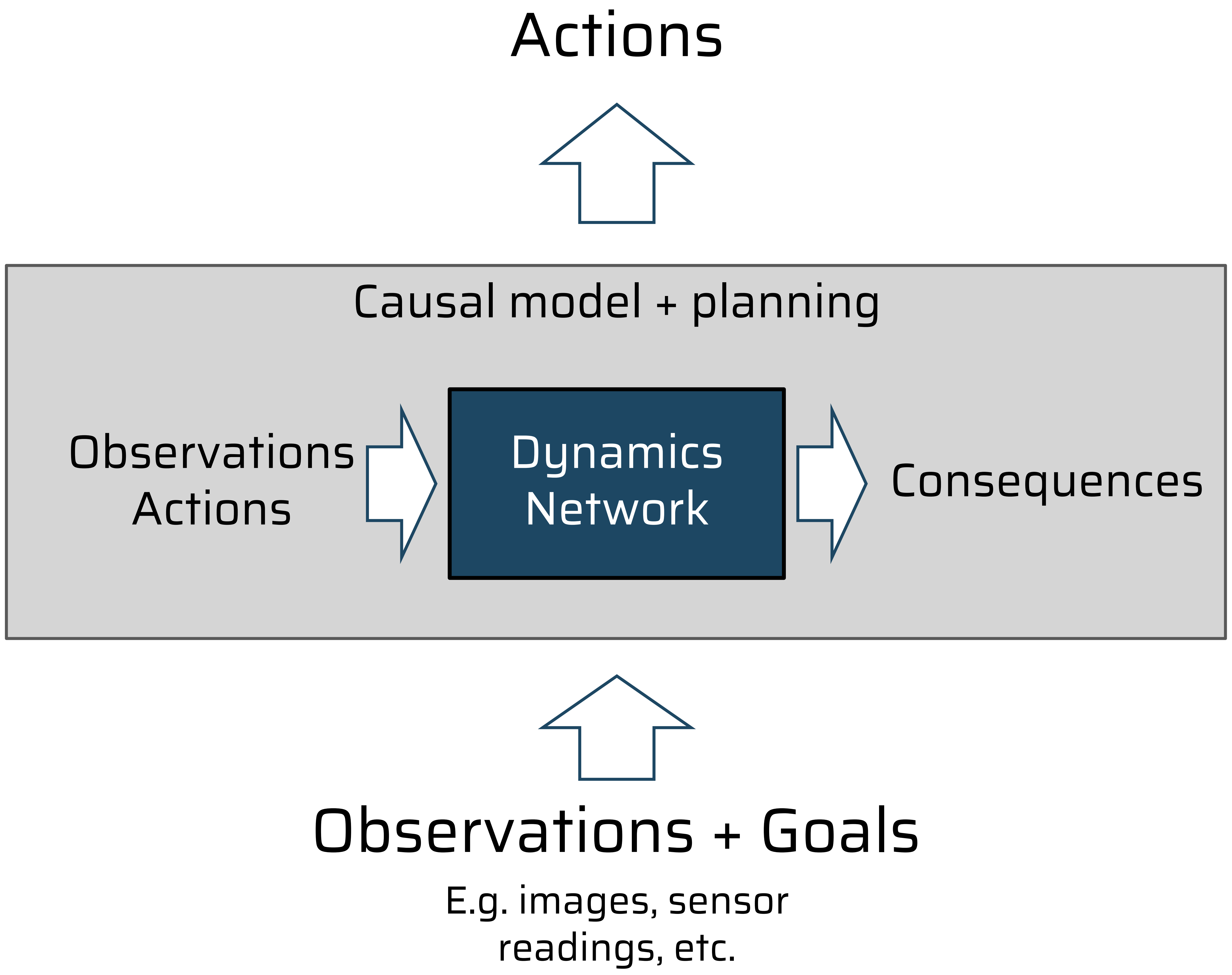
Model-Based RL

Model the underlying causal process

- needs much less training data
- applicable to new situations
- offers explanations, can answer “what if” questions

Just add planning

- the main drawback is that simulations can be costly ⇒ not a replacement of normal stimulus-response but a perfect complement



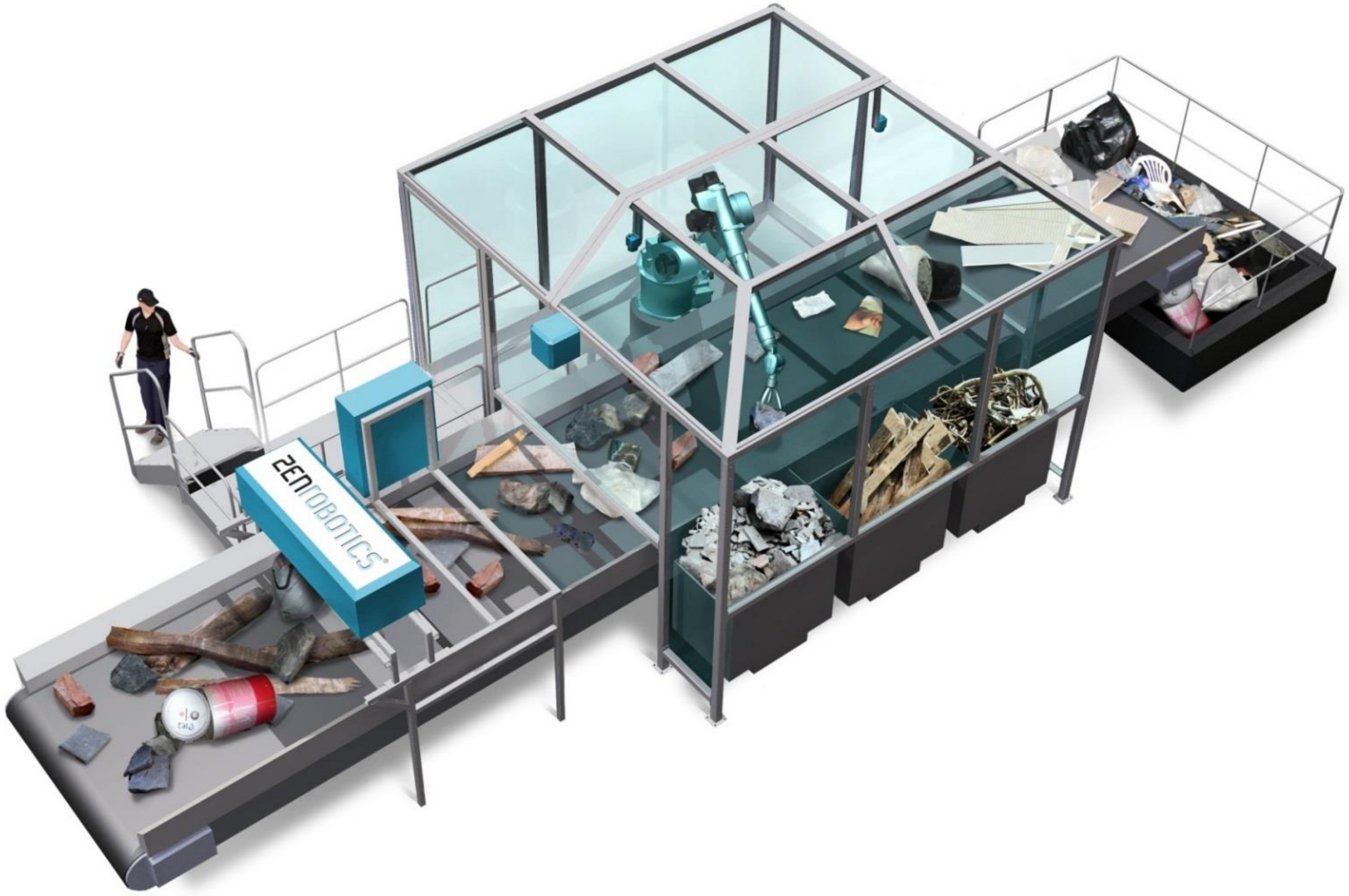
Model-Based Control with Simulator as the Model

Synthesis of Complex Behaviors
with
Online Trajectory Optimization

(preliminary results)

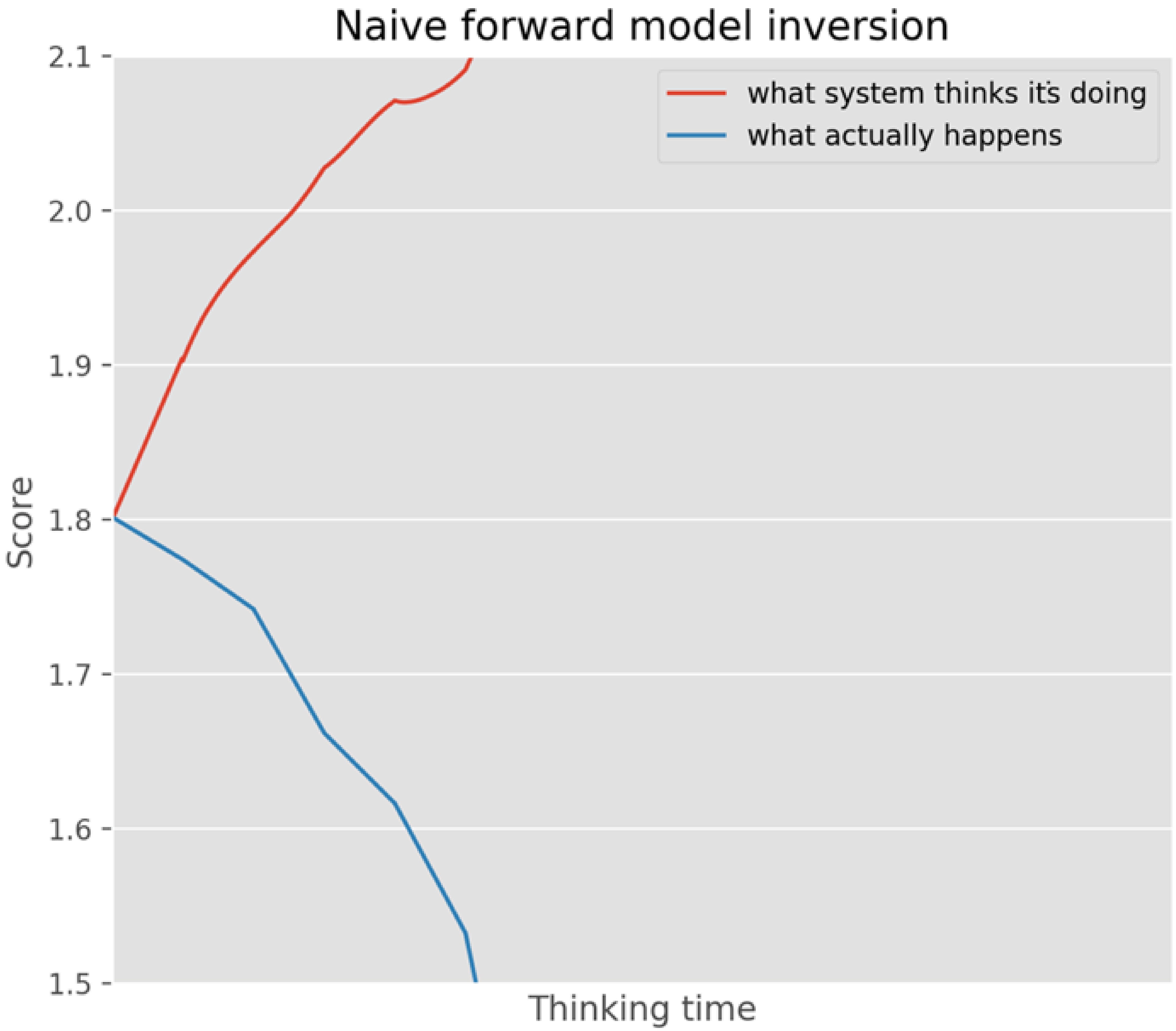
Emanuel Todorov, Tom Erez and Yuval Tassa (2012)

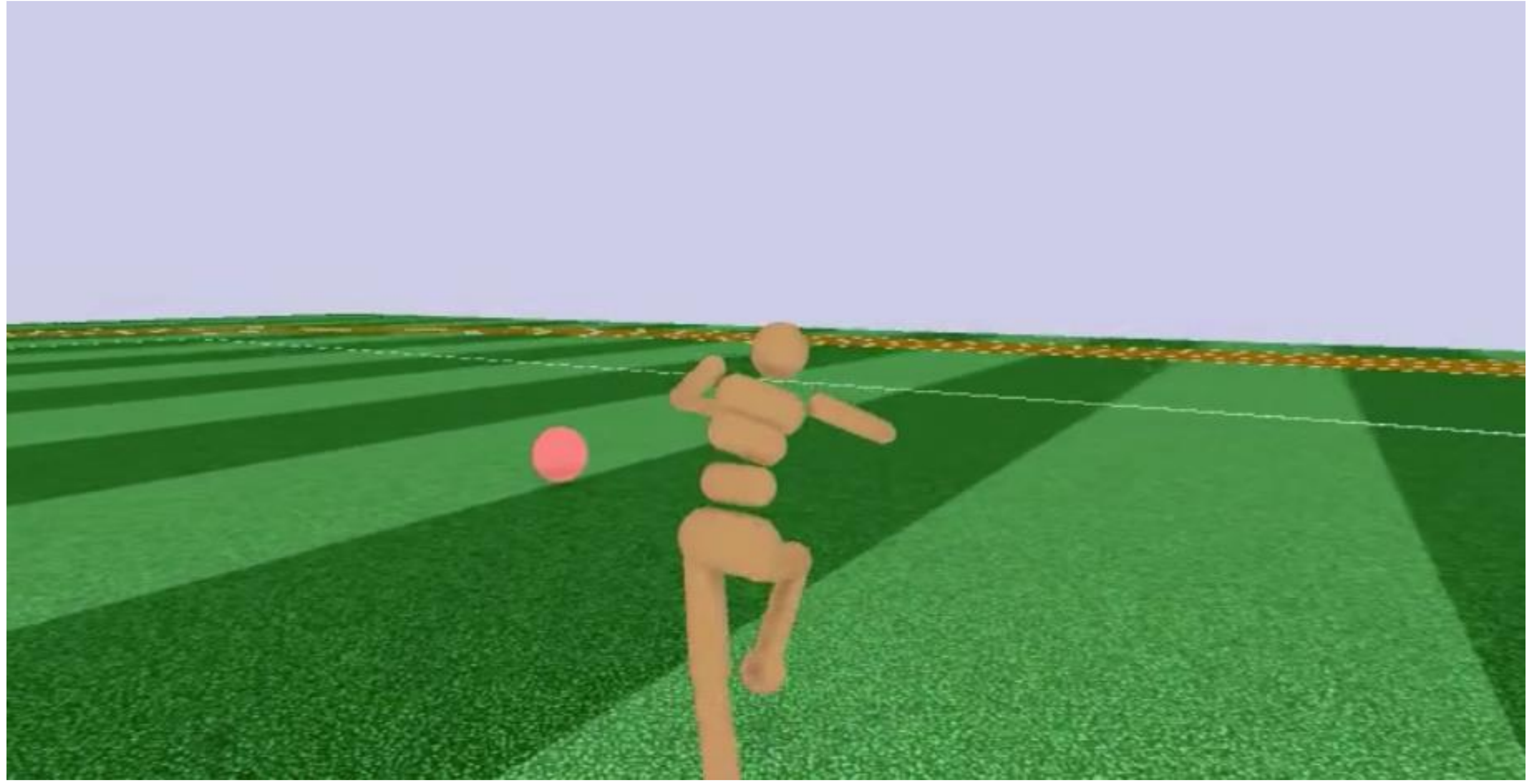
2007 –
ZenRobotics Ltd.



02 WHAT’S THE PROBLEM?

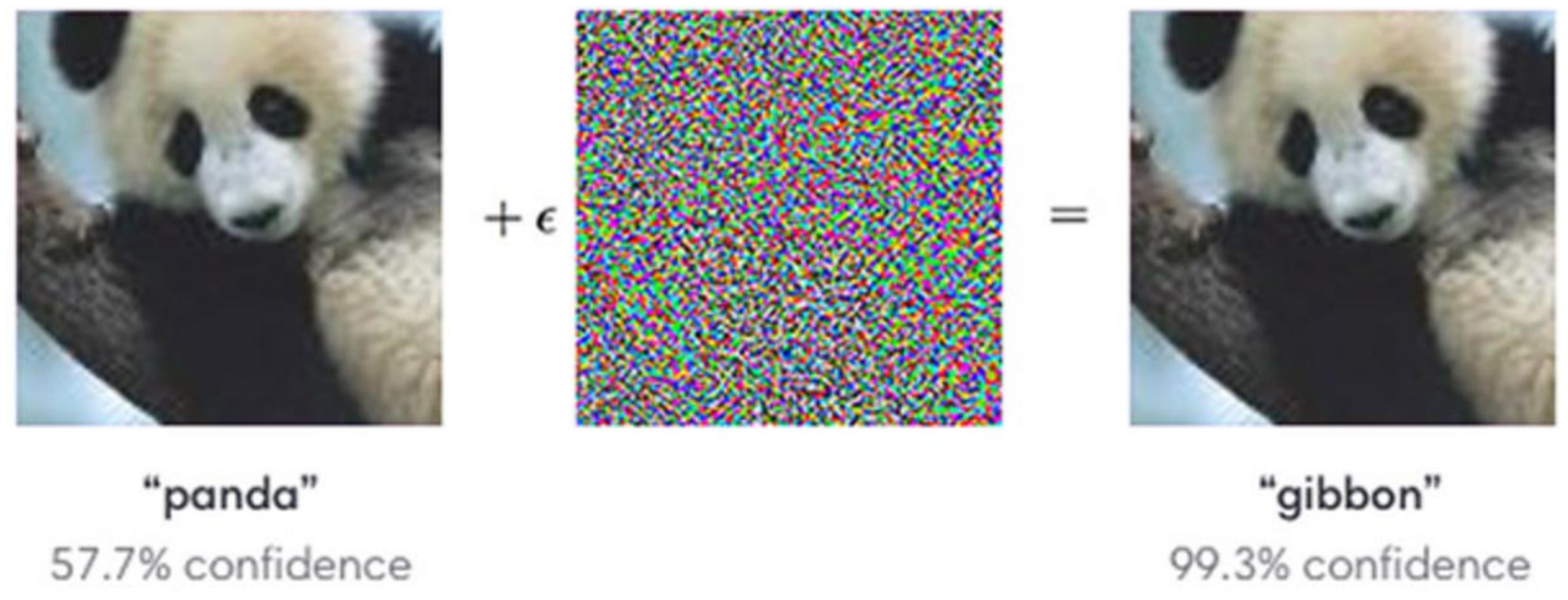
Learned Dynamics Model → Delusional Planning





Neural Network Inversion is not Stable

BY SIMPLY OPTIMIZING INPUTS BY GRADIENT DESCENT, NEURAL NETWORKS CAN BE FOOLED

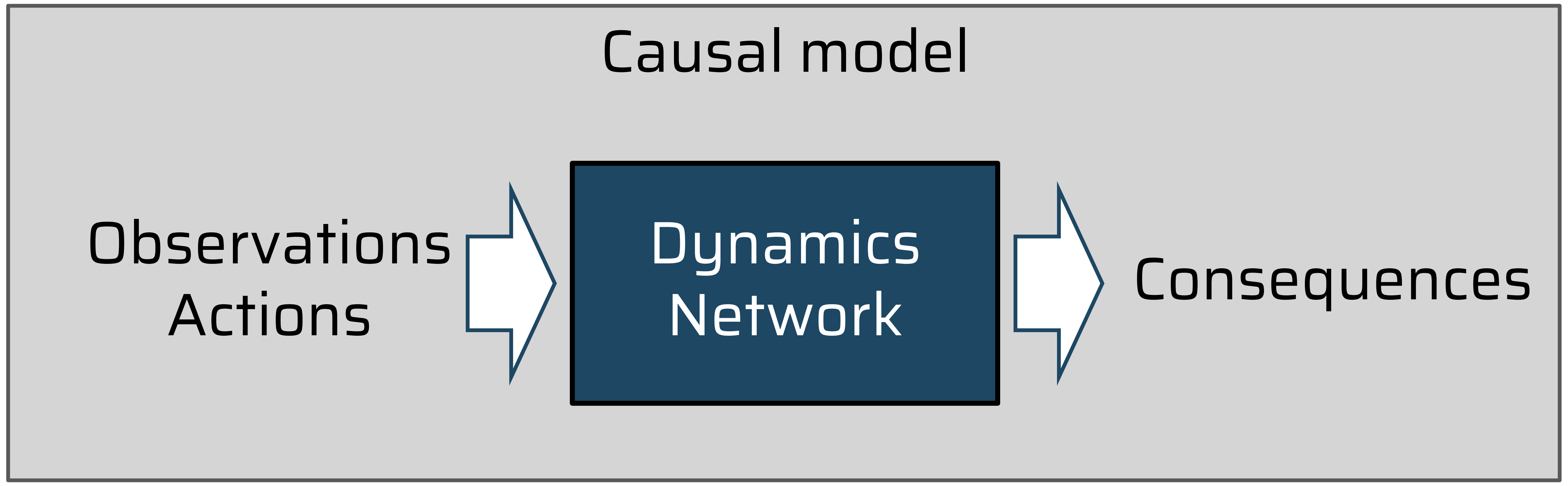


Model-Based RL

Backpropagate from desired consequences to actions

→

adversarial examples of actions



Neural Network Inversion is not Stable

Inverting a neural network gives nonsensical results (so called adversarial examples)

Control does not work if dynamics models are learned by neural networks

Root Cause:

Neural networks fail without warning outside their training manifold

They don't understand their own uncertainty

03 THE SOLUTION

First: What Doesn’t Work

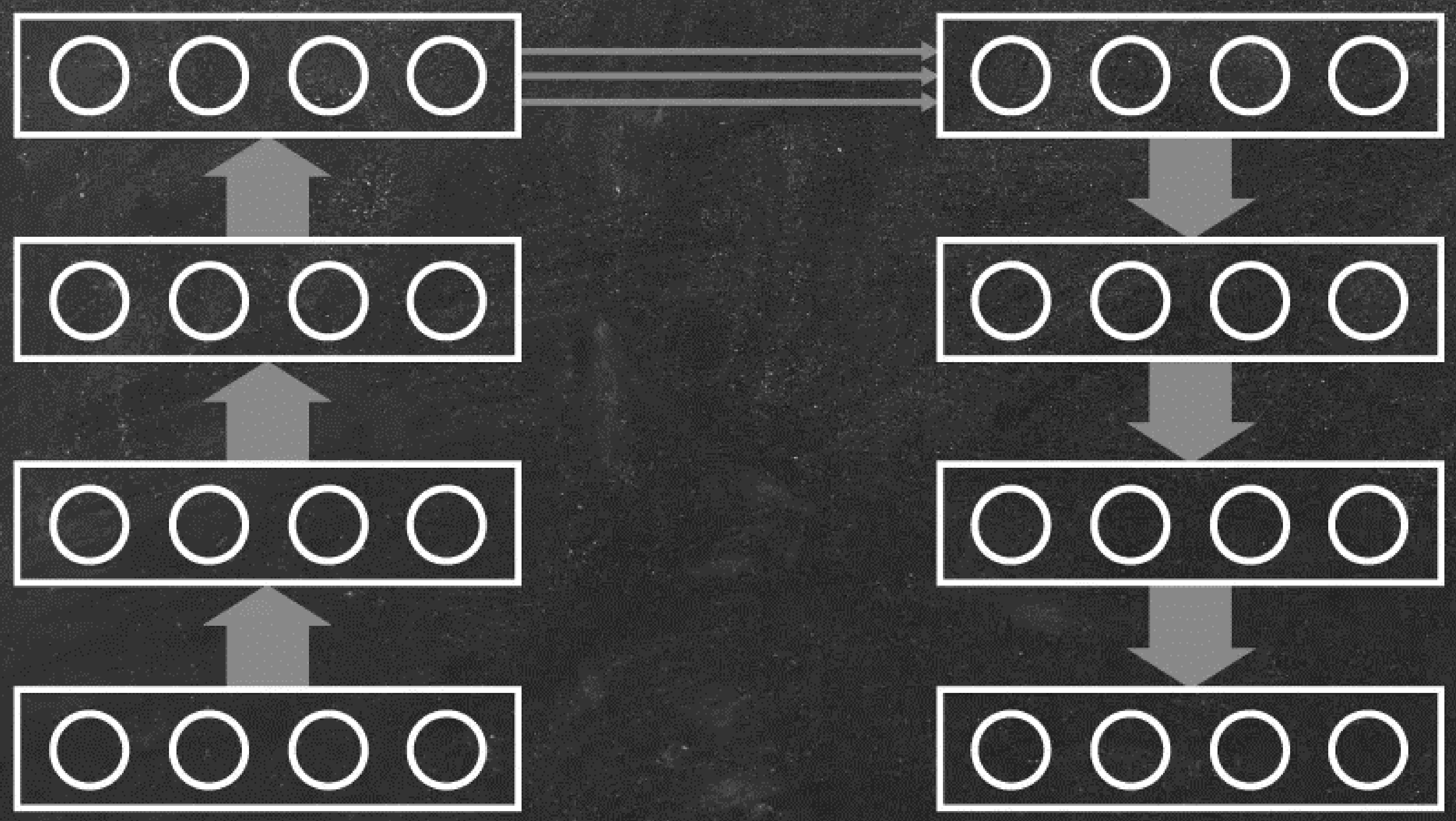
We tried several ways to estimate the uncertainty

They all work at least somehow when used in ordinary prediction…

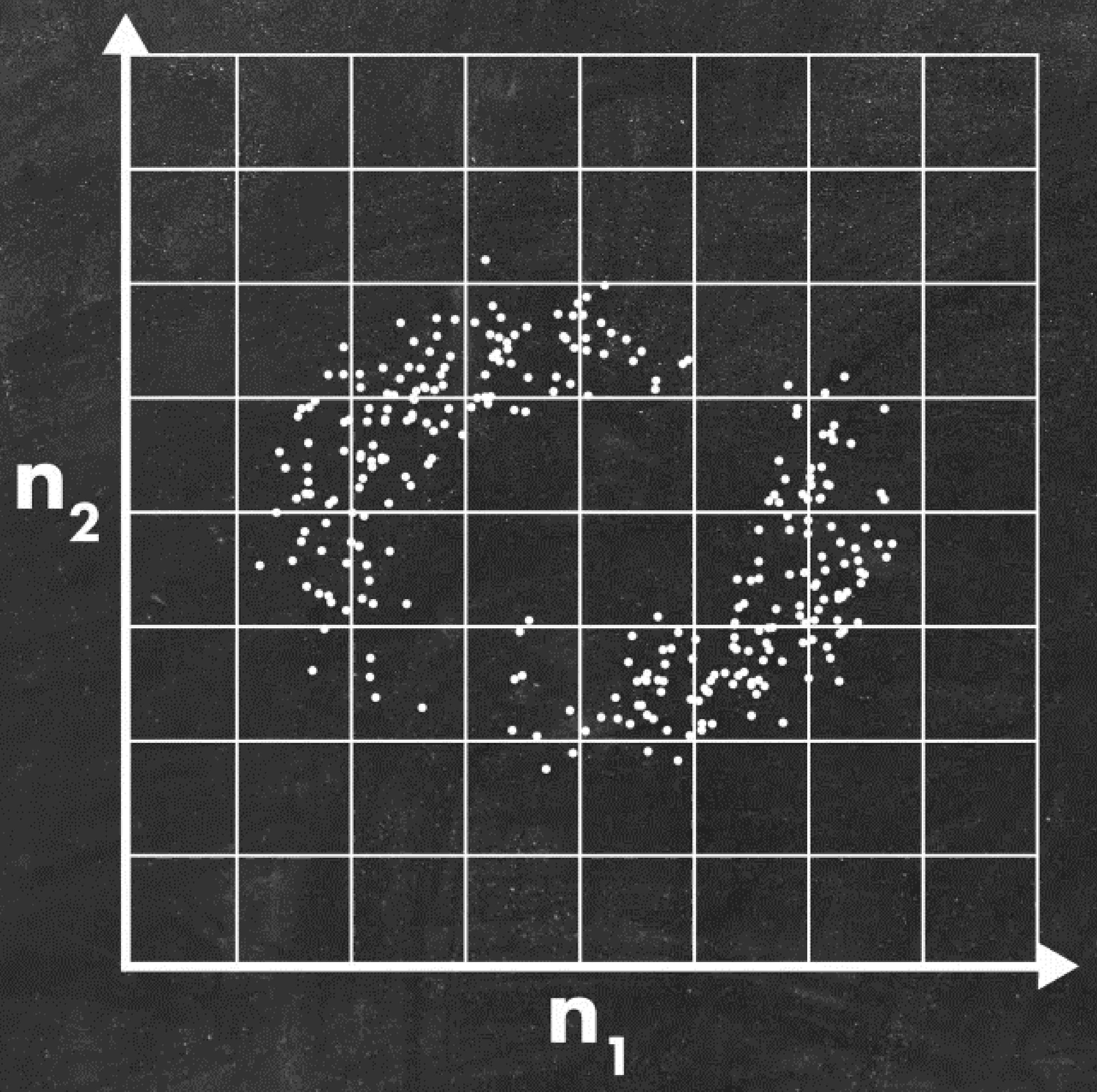
… but almost all fail under “adversarial planning attack”

When you model the uncertainty or familiarity directly, optimization finds the weak spots

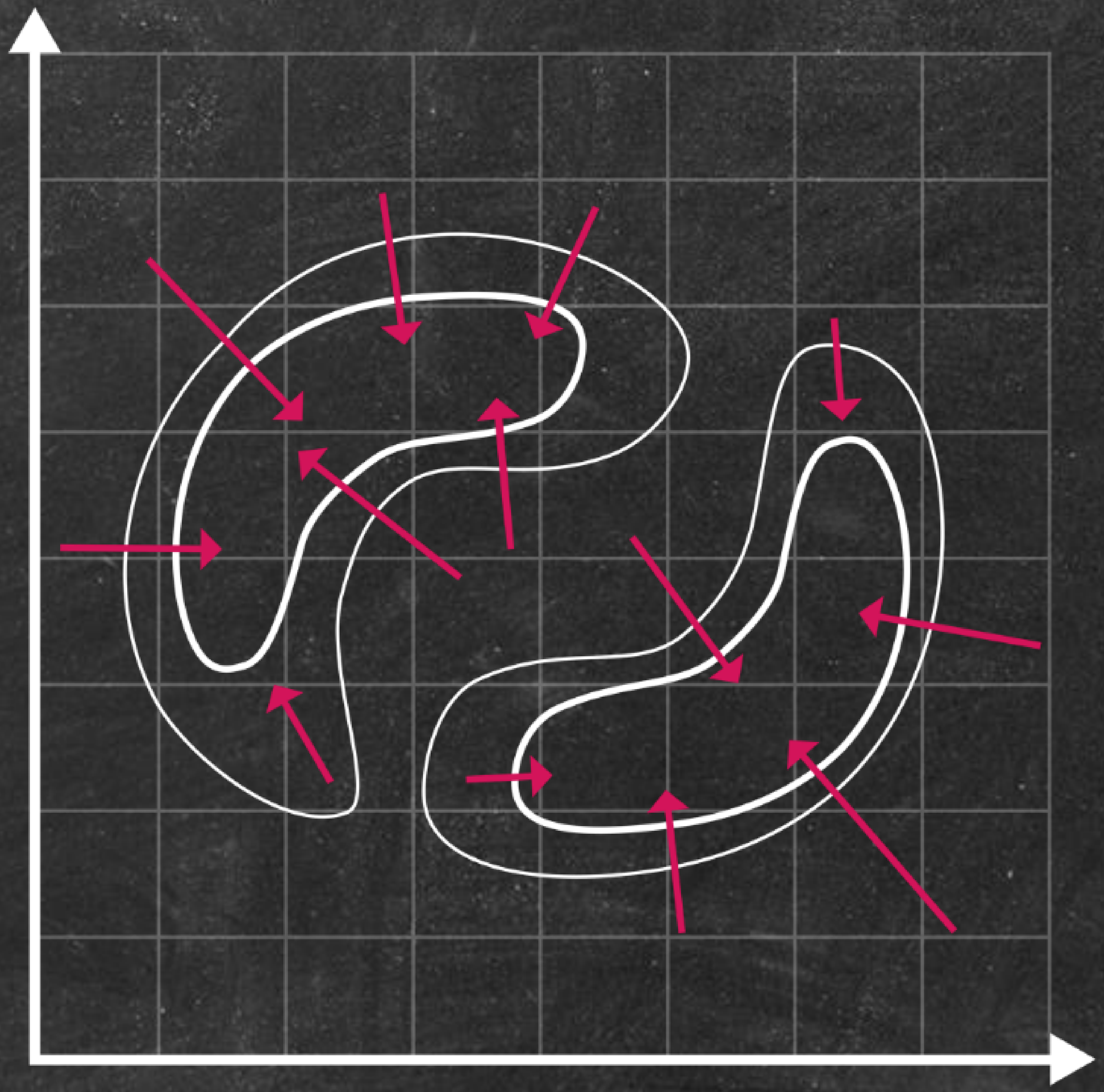
Denoising Auto-Encoder



Original Sample Distribution



Denoising Auto-Encoders Learn $\nabla \log p(x)$



The Winner

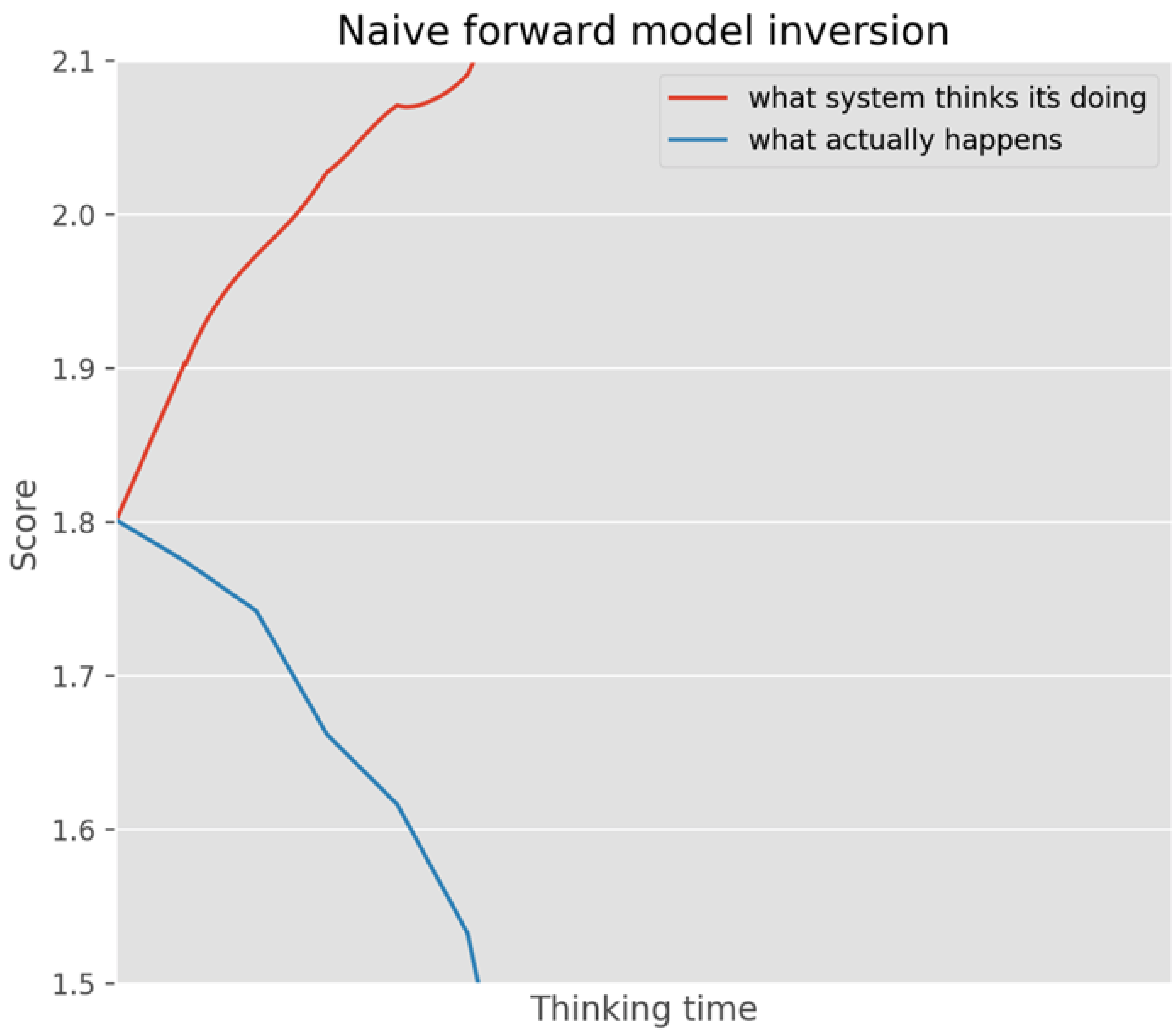
Use denoising autoencoders to explicitly model $\nabla \log p(x)$

→

The gradient is sometimes wrong but optimization is not pulled towards these weak spots

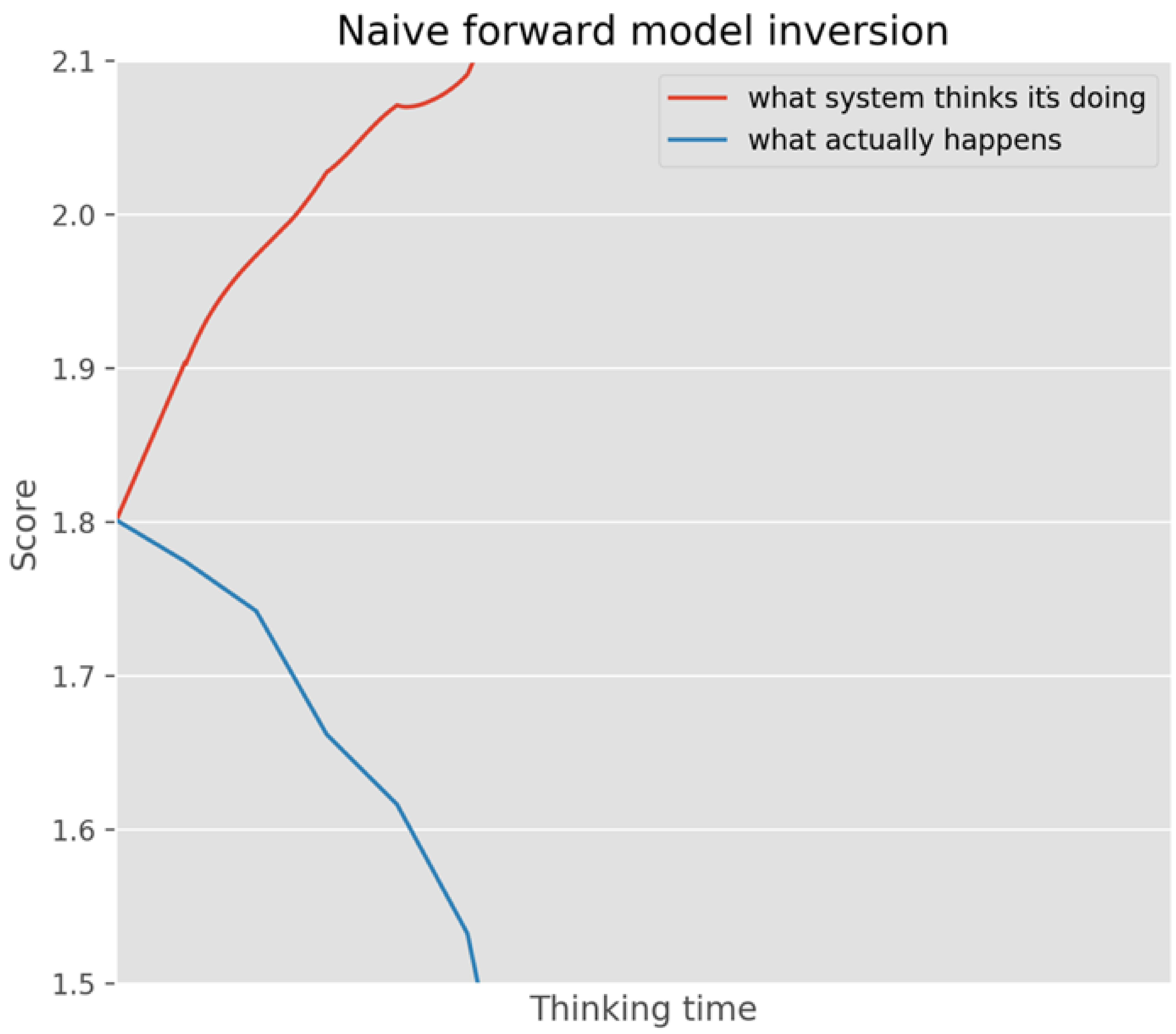
Denoising Autoencoder as a Safety Net → Stable Planning

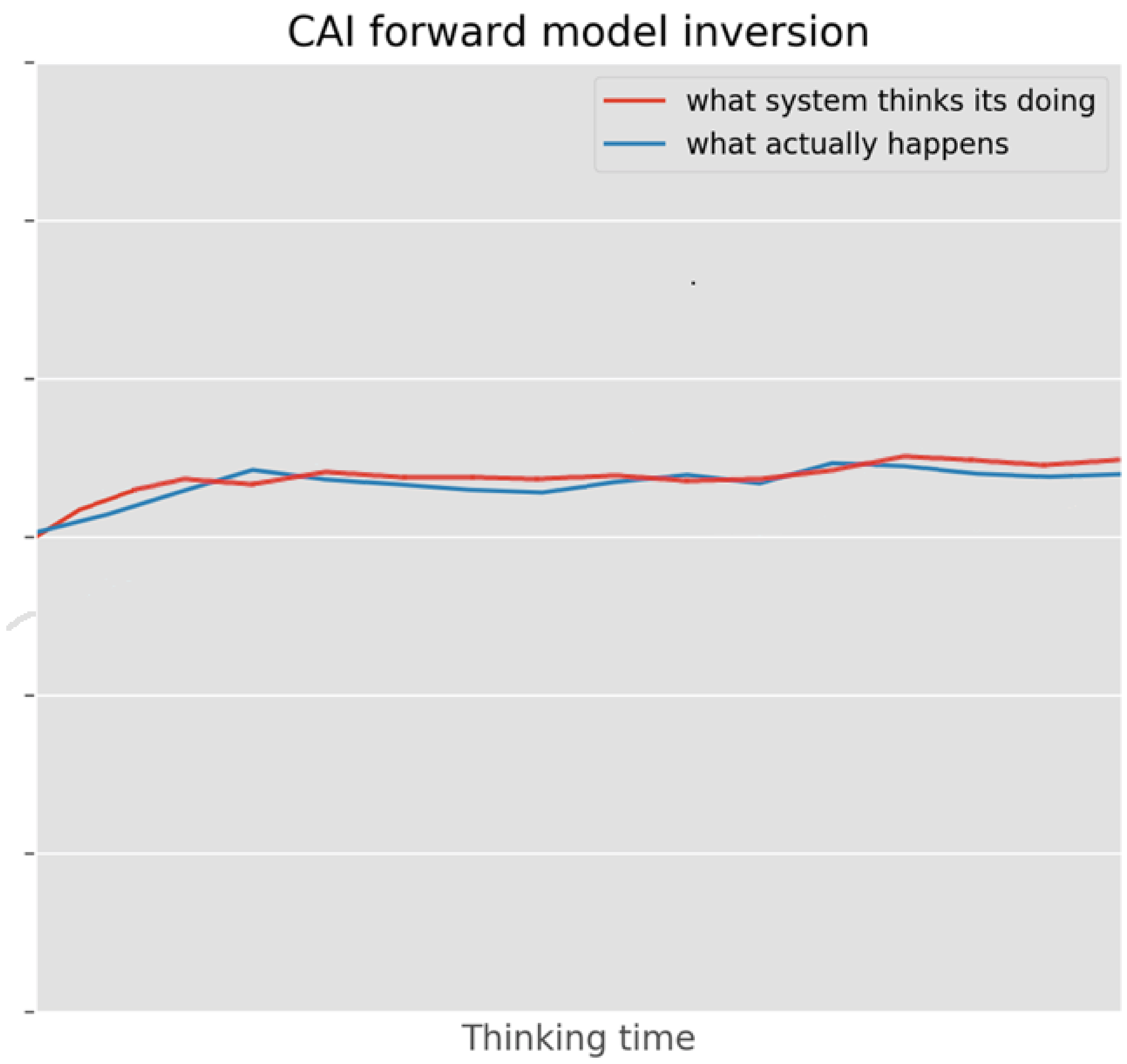
CURIOUS AI



Denoising Autoencoder as a Safety Net → Stable Planning

CURIOUS AI





04 RELEVANCE TO MATERIAL SCIENCES

Starting point: blind tinkering

1. Define the desired properties of a new material
2. Try a recipe for cooking up something
3. Measure the properties of the resulting material
4. If not happy, change the recipe and go back to step 2

Speeding things up

Try a recipe for cooking up something

Simulate the properties of the resulting material

Learn to predict the outcome of the simulation and real experiment

ML can help:

- Learn the predictions
- Learn the uncertainty
- Learn efficient search

Thanks!

harri@cai.fi