ISE Methods in Building Reliable AI Systems

Mansur Maturidi Arief, Ph.D. (pronouns: Mansur, Mas, Kak, Dik, ~~Pak~~)
Postdoctoral Scholar, Stanford Intelligent Systems Lab
Email: mansur.arief@stanford.edu | Website: www.mansurarief.github.io

SISL
Stanford Intelligent Systems Laboratory

About Mansur

- Born and raised in Gowa (Sulsel), went to Pesantren IMMIM Putra Makassar
- Bachelor's degree: TI ITS (2010)
  - PBSB scholarship from Kemenag (2010-2014)
  - LSCM lab assistant (2012-2014), ITS International Office (2014), OSCM (2014)
- Master's degree: IOE, University of Michigan, Ann Arbor (2018)
  - LPDP scholarship (2016-2018)
  - ISLI Student Chapter (2018)
- PhD degree: Mechanical Engineering, Carnegie Mellon University (2023)
  - Research assistantship, funded by the CIT Dean Scholarship (2018), NSF (2019-2023)
  - Research area: decision-making under uncertainty, applied ML, intelligent transportation, AI safety
- Postdoc: Aeronautics & Astronautics, Stanford University (2025?)
  - Working with 7 PhD, 2 master's, and 1 undergrad students on various projects
  - Co-founder of Indonesian interdisciplinary scholars IndoSTEELERS and INTERSECT

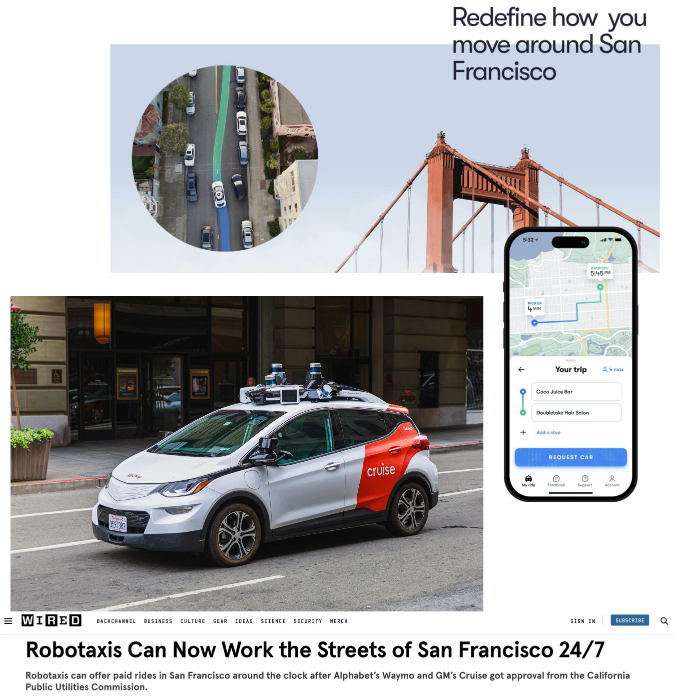
SISL
Stanford Intelligent Systems Laboratory

What we'll discuss

- One of the main methods enabling AI is **numerical optimization**!
- The algorithms use **tricks** to reach a good enough solution.
- **Formulations** include
  - model fitting (regression, classification boundary)
  - falsification and validation (FMEA, adversarial attack, importance sampling)
  - [tentative] utility maximization (MDP/POMDP — planning under uncertainty framework)
- **The PDF slide will be made available after the presentation.**

SISL
Stanford Intelligent Systems Laboratory

AI systems are here…



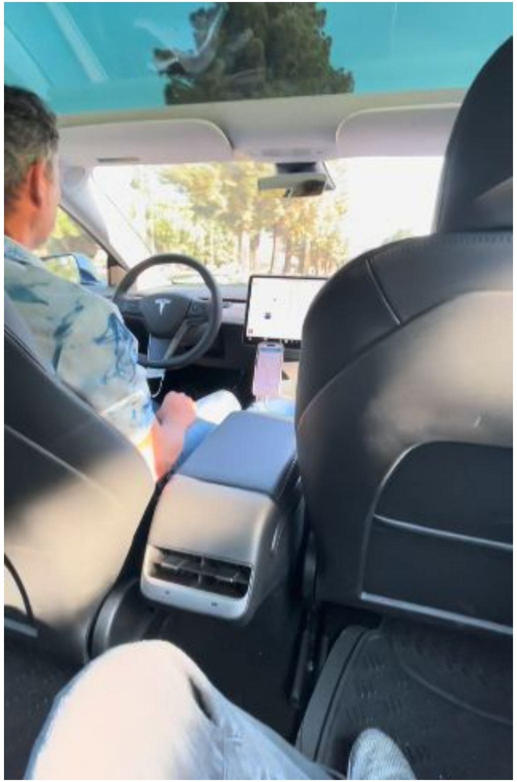
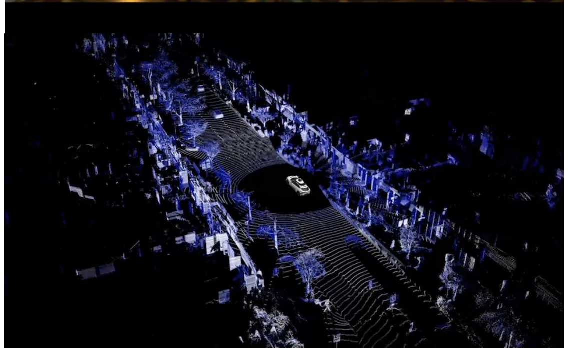


Photo I took a month ago, in an Uber ride.

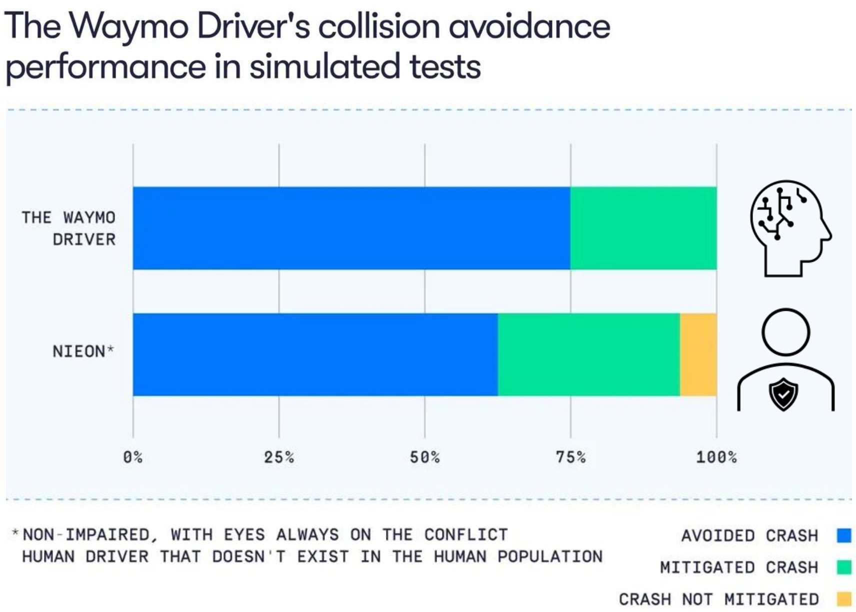
AI systems are here…

How self-driving cars "see" their environments



SISL
Stanford Intelligent
Systems Laboratory

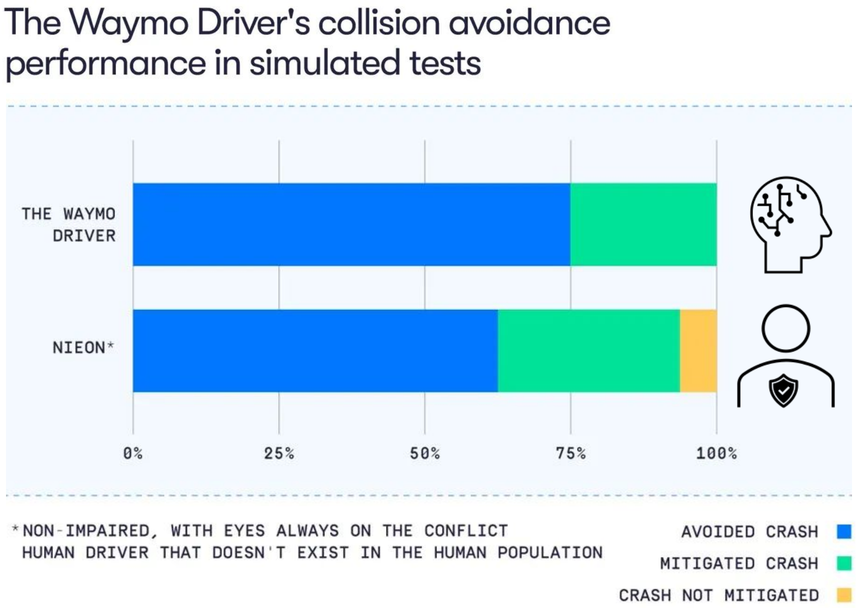
AI systems are here… and becoming more reliable everyday



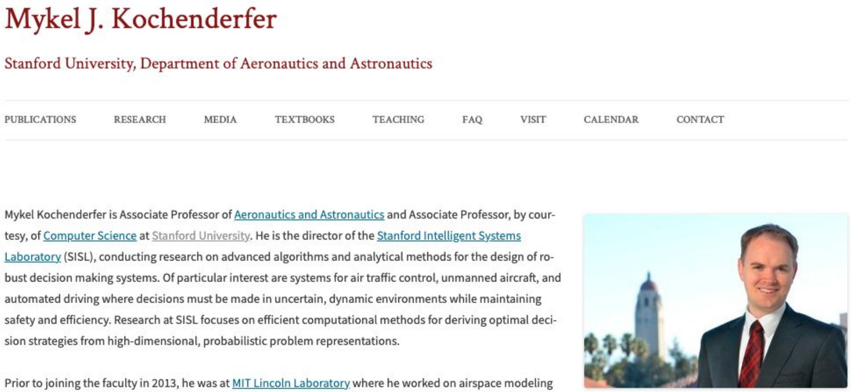
Source: https://www.theverge.com/2022/9/29/23377219/waymo-av-safety-study-response-time-crash-avoidance, https://waymo.com/waymo-one-san-francisco/.

A next-generation collision avoidance system will help pilots and unmanned aircraft safely navigate the airspace.

AI systems are here… and becoming more reliable everyday



Source: https://www.theverge.com/2022/9/29/23377219/waymo-av-safety-study-response-time-crash-avoidance, https://waymo.com/waymo-one-san-francisco/



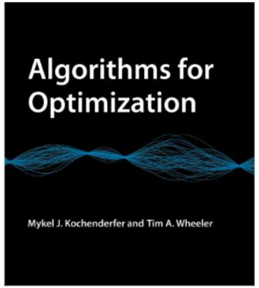
Mykel J. Kochenderfer

Stanford University, Department of Aeronautics and Astronautics

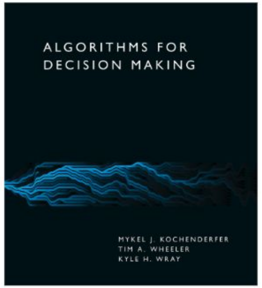
PUBLICATIONS RESEARCH MEDIA TEXTBOOKS TEACHING FAQ VISIT CALENDAR CONTACT

Mykel Kochenderfer is Associate Professor of Aeronautics and Astronautics and Associate Professor, by courtesy, of Computer Science at Stanford University. He is the director of the Stanford Intelligent Systems Laboratory (SISL), conducting research on advanced algorithms and analytical methods for the design of robust decision making systems. Of particular interest are systems for air traffic control, unmanned aircraft, and automated driving where decisions must be made in uncertain, dynamic environments while maintaining safety and efficiency. Research at SISL focuses on efficient computational methods for deriving optimal decision strategies from high-dimensional, probabilistic problem representations.

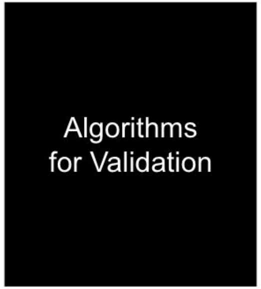
Prior to joining the faculty in 2013, he was at MIT Lincoln Laboratory where he worked on airspace modeling



2019



2022



Coming soon!

SISL
Stanford Intelligent Systems Laboratory

AI systems are here… and becoming more reliable everyday

Mykel J. Kochenderfer

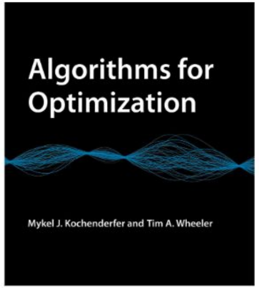
Stanford University, Department of Aeronautics and Astronautics

PUBLICATIONS RESEARCH MEDIA TEXTBOOKS TEACHING FAQ VISIT CALENDAR CONTACT

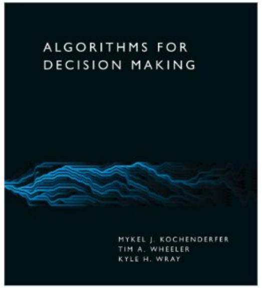
Mykel Kochenderfer is Associate Professor of Aeronautics and Astronautics and Associate Professor, by courtesy, of Computer Science at Stanford University. He is the director of the Stanford Intelligent Systems Laboratory (SISL), conducting research on advanced algorithms and analytical methods for the design of robust decision making systems. Of particular interest are systems for air traffic control, unmanned aircraft, and automated driving where decisions must be made in uncertain, dynamic environments while maintaining safety and efficiency. Research at SISL focuses on efficient computational methods for deriving optimal decision strategies from high-dimensional, probabilistic problem representations.

Prior to joining the faculty in 2013, he was at MIT Lincoln Laboratory where he worked on airspace modeling

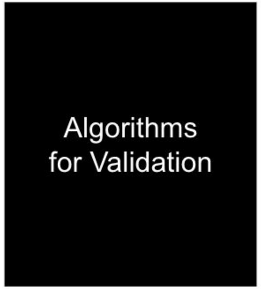
Can we push AI safety to reach aviation-level safety?



2019



2022



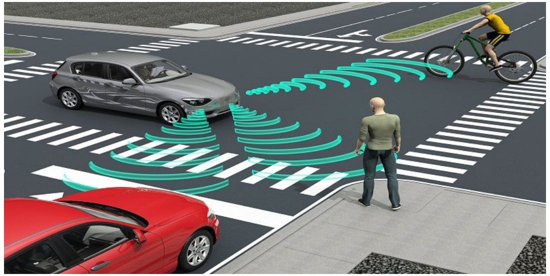
Coming soon!

SISL
Stanford Intelligent
Systems Laboratory

More high-stakes decisions are supported by AI

Autonomy stack

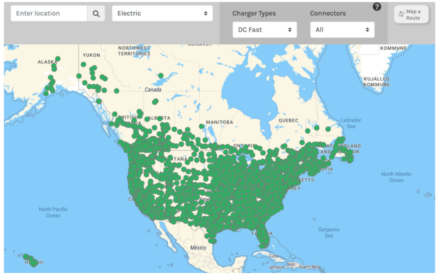
High dimensionality



Preventing accidents

EV charging stations

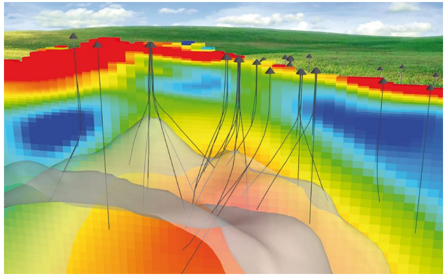
High complexity



Reliable services

Geothermal wells

High uncertainty



Sustainable life

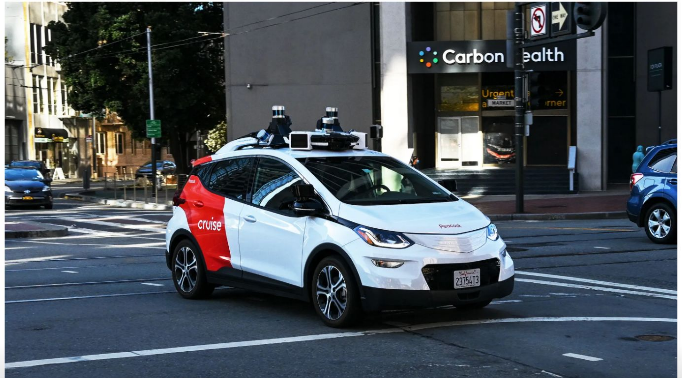
SISL
Stanford Intelligent
Systems Laboratory

Is it safe and reliable enough?

AARIAN MARSHALL BUSINESS OCT 24, 2023 4:31 PM

GM's Cruise Loses Its Self-Driving License in San Francisco After a Robotaxi Dragged a Person

The California DMV says the company's autonomous taxis are "not safe" and that Cruise "misrepresented" safety information about its self-driving vehicle technology.



WIRED

Depending on the use cases, but for critical applications, a lot more needs to be done.

SISL
Stanford Intelligent Systems Laboratory

Highlighted Topics (please ask q's and follow up for details)

1. AI definition and how it works
   a. Introduction
   b. Large-scale optimization algorithms (gradient descent and variants)
2. Risk analysis and validation
   a. Risk analysis (FTA, adversarial attacks)
   b. Probabilistic validation (importance sampling)
3. [Tentative] Planning algorithms
   a. Sequential decision-making under uncertainty (MDP/POMDP)

SISL
Stanford Intelligent
Systems Laboratory

Interactive Sessions

1. AI definition and how it works

   [DEMO 1] Object detection and captioning

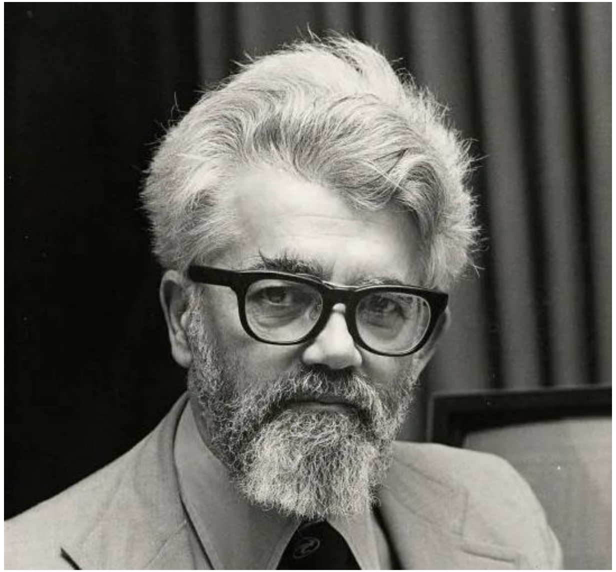
2. Risk analysis and validation

   [DEMO 2] Fool the AI!

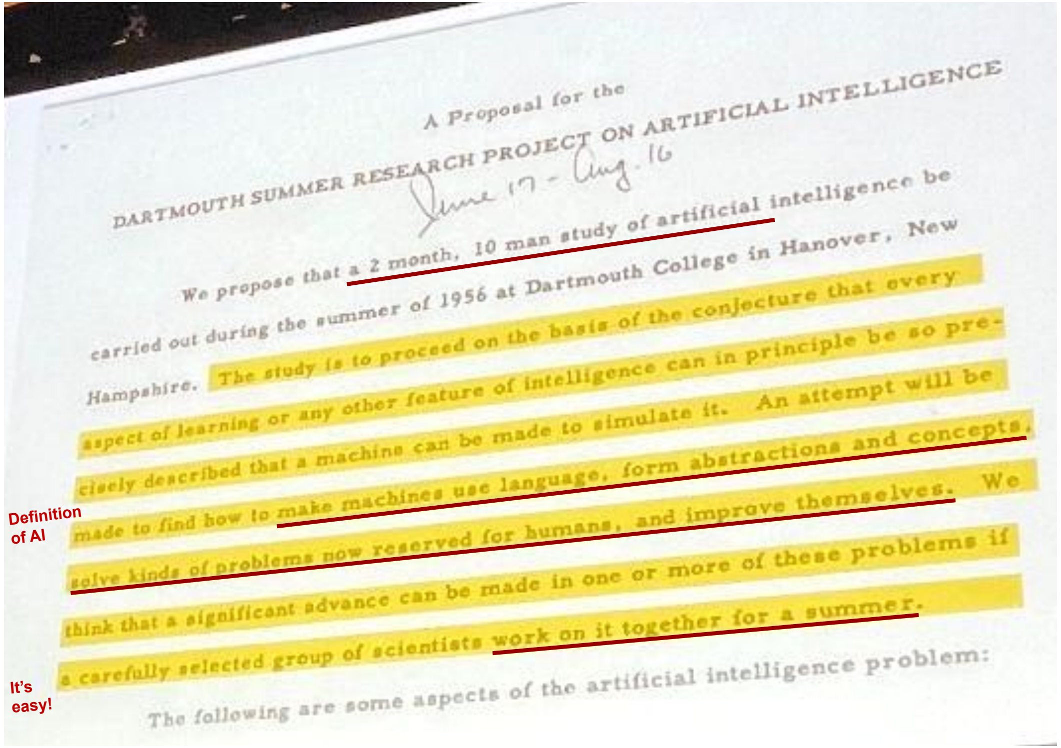
1. AI: Definition and How It Works

SISL
Stanford Intelligent Systems Laboratory

Artificial Intelligence (AI)



John McCarthy (1927-2011), Father of AI, Stanford CS Professor, http://jmc.stanford.edu/



A Proposal for the

DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

The following are some aspects of the artificial intelligence problem:

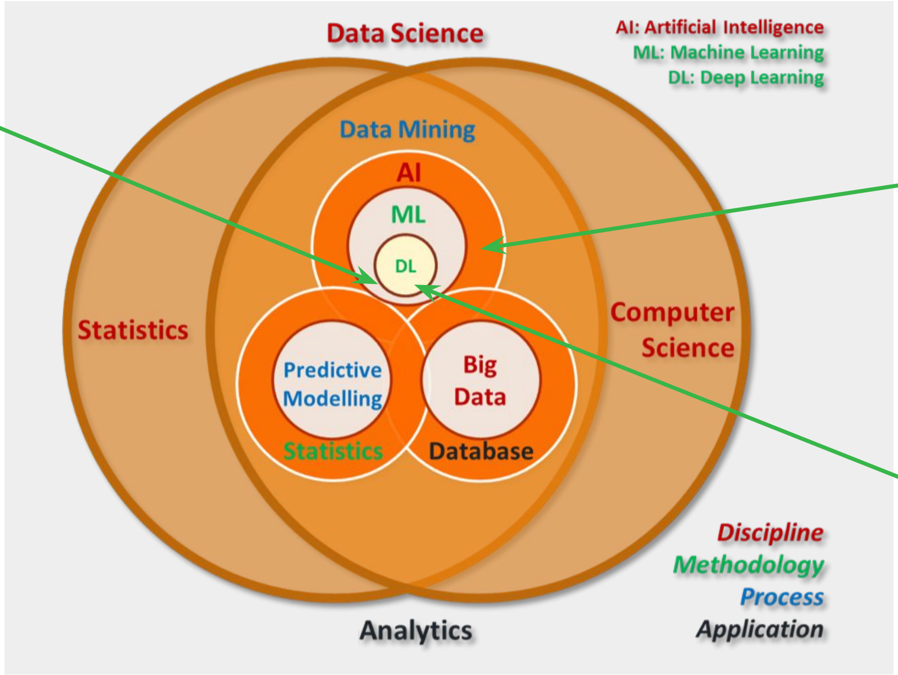
By John McCarthy (together with Marvin Minsky, Nathaniel Rochester, and Claude Shannon), Stanford Archive

SISL
Stanford Intelligent Systems Laboratory

Artificial Intelligence (AI)

- McCarthy: "the science … of making intelligent machines."

ML, but not DL
Decision Trees
Regression
…



AI, but not ML
Rule-based,
Lookup tables,
…

AI with DL
YOLO
ChatGPT
…

By Coppertreeanalytics

SISL
Stanford Intelligent
Systems Laboratory

Artificial Intelligence (AI)

```python
import streamlit as st
from transformers import YolosImageProcessor, YolosForObjectDetection
from PIL import Image
import torch
model = YolosForObjectDetection.from_pretrained('hustvl/yolos-tiny')
image_processor = YolosImageProcessor.from_pretrained("hustvl/yolos-tiny")
image = Image.open(st.file_uploader("Choose an image...", type="jpg"))
outputs = model(**image_processor(images=image, return_tensors="pt"))
results = image_processor.post_process_object_detection(outputs, 0.95, torch.tensor([image.size[::-1]]))[0]
st.pyplot(draw_bounding_boxes(image, results, model)) #st.write("Result:", results)

```

Source: https://www.its.ac.id/news/wp-content/uploads/sites/2/2018/10/WhatsApp-Image-2018-10-25-at-15.17.40.jpeg

SISL
Stanford Intelligent Systems Laboratory

Demo 1: Object detection and captioning

https://huggingface.co/spaces/mansurarief/DTSI-demo-1

SISL
Stanford Intelligent
Systems Laboratory

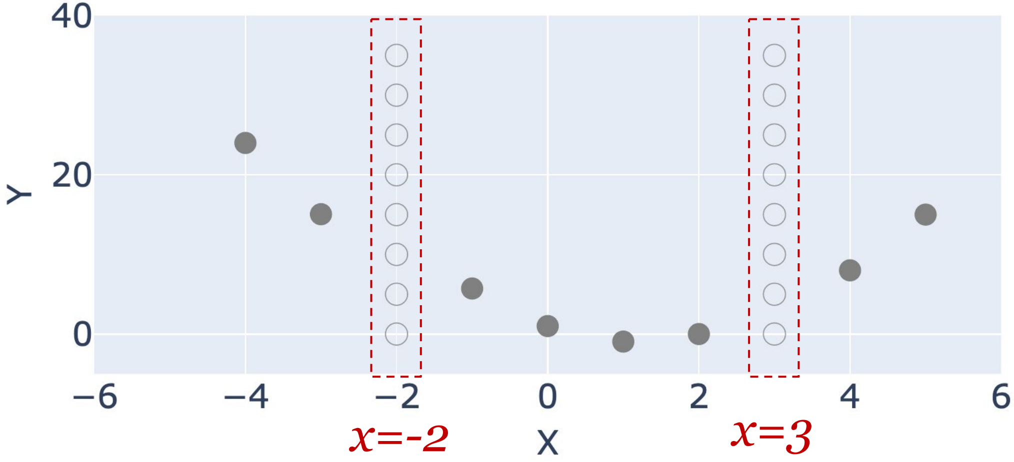
Discussion prompts:

- Any interesting/not-so-interesting findings?
- Any useful/harmful applications?
- How does it work?
  - How does the algorithm convert an image to annotated image (bounding boxes, labels, and image caption)?

SISL
Stanford Intelligent Systems Laboratory

Problem representation

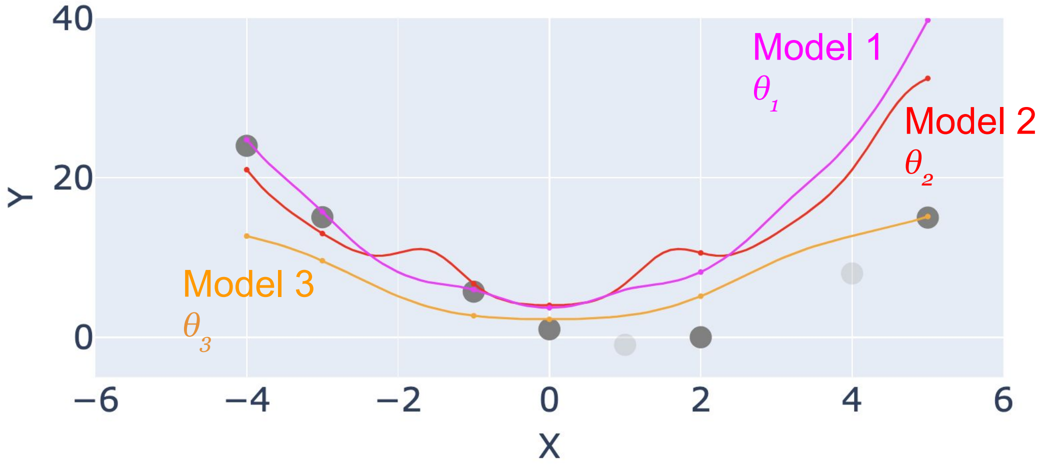
- **Regression**: given an input x, we try to predict a (continuous value) y



SISL
Stanford Intelligent Systems Laboratory

Problem representation

- **Regression**: given an input x, we try to predict a (continuous value) y



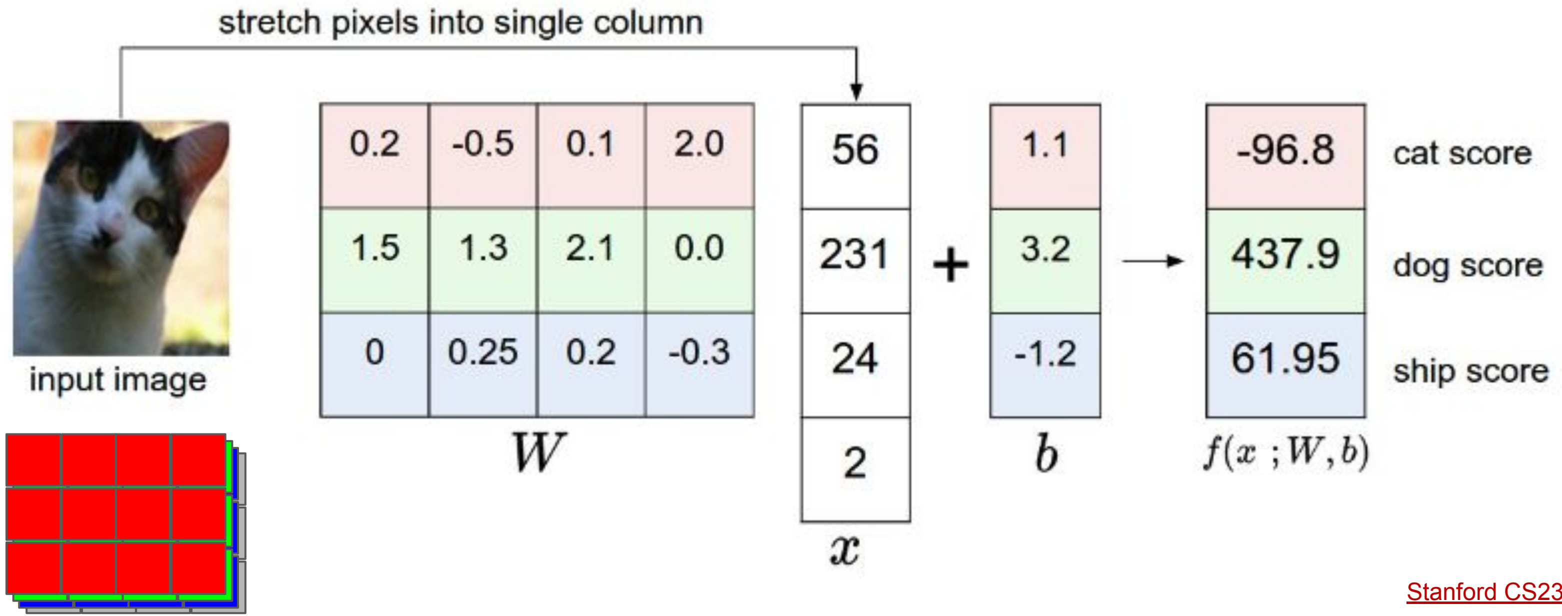
  - Pick a class of model (e.g., linear, polynomials, neural nets).
  - Compare models within the class, and select the <u>best</u> one.
  - Use the best model to predict.

How to select the best θ?

SISL
Stanford Intelligent Systems Laboratory

Problem representation

- **Regression**: given an input x, we try to predict a (continuous value) y
- **Classification**: given an input x, we try to predict a (discrete class) y



SISL
Stanford Intelligent Systems Laboratory

Problem representation

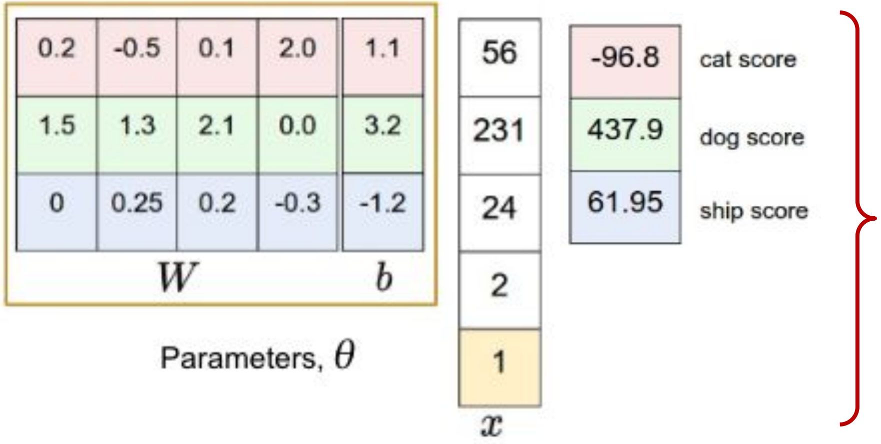
- **Regression**: given an input x, we try to predict a (continuous value) y
- **Classification**: given an input x, we try to predict a (discrete class) y

$$\underbrace{\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 \\ 1.5 & 1.3 & 2.1 & 0.0 \\ 0 & 0.25 & 0.2 & -0.3 \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}}_{b} \longleftrightarrow \boxed{\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 & 1.1 \\ 1.5 & 1.3 & 2.1 & 0.0 & 3.2 \\ 0 & 0.25 & 0.2 & -0.3 & -1.2 \end{bmatrix}} \underbrace{\begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \\ 1 \end{bmatrix}}_{x}$$

Parameters, θ

Problem representation

- **Regression**: given an input x, we try to predict a (continuous value) y
- **Classification**: given an input x, we try to predict a (discrete class) y

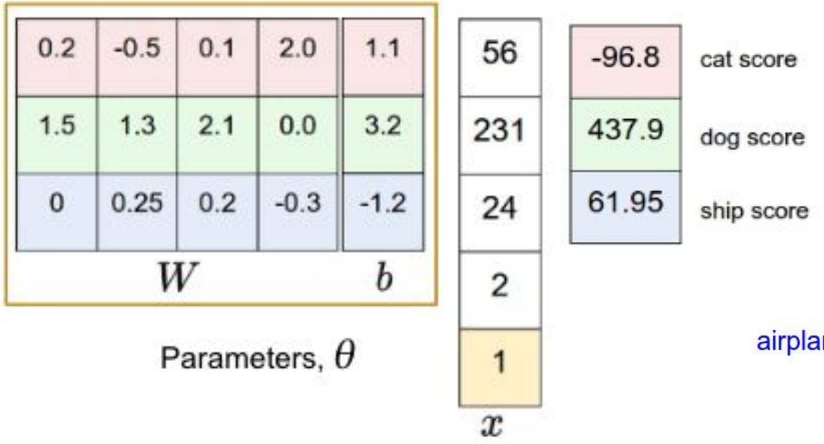


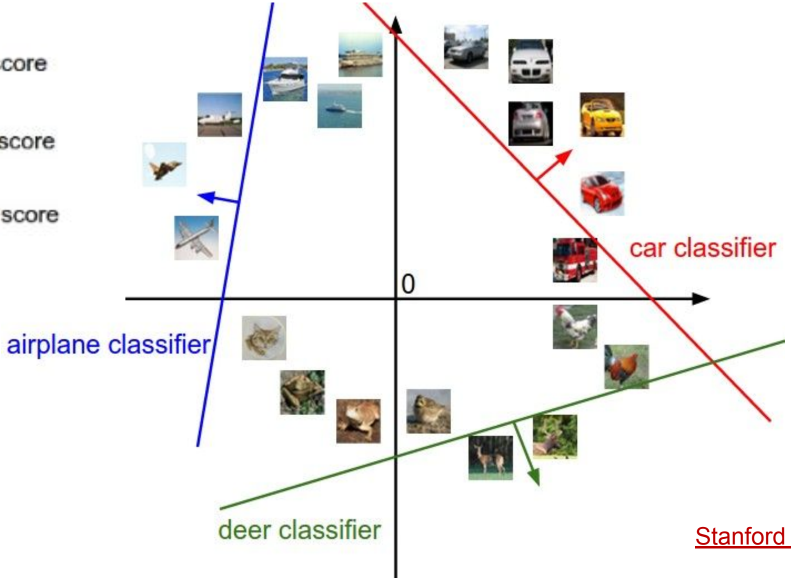
How to select the best θ?

Problem representation

- **Regression**: given an input x, we try to predict a (continuous value) y
- **Classification**: given an input x, we try to predict a (discrete class) y

(essentially, we try to find the best decision boundary)





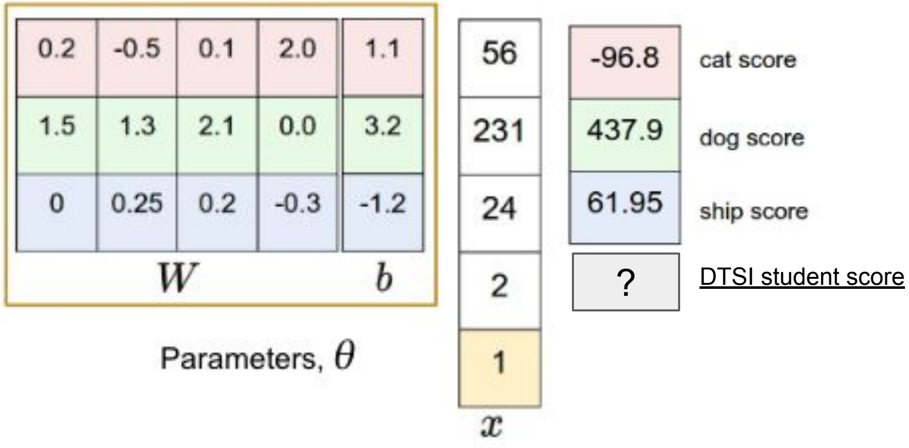
This involves a matrix multiplication

SISL
Stanford Intelligent
Systems Laboratory

Problem representation

- **Regression**: given an input x, we try to predict a (continuous value) y
- **Classification**: given an input x, we try to predict a (discrete class) y

3×5 5×1 3×1



← a new class

Hint: It's a matrix multiplication.

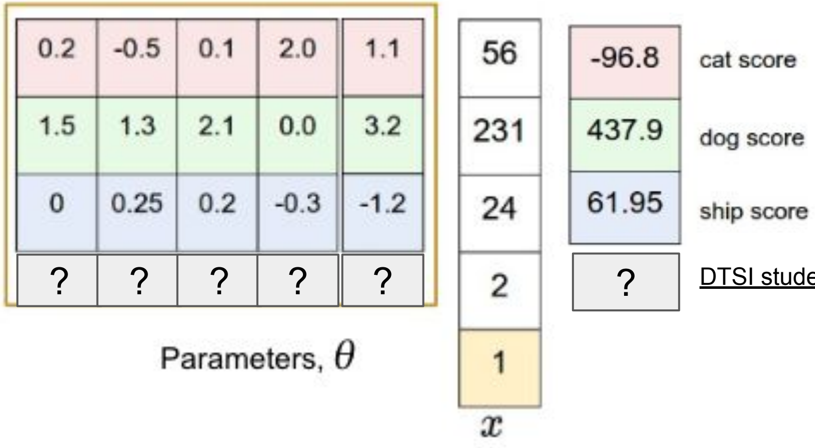
What should we change to θ to add a new class?

SISL
Stanford Intelligent Systems Laboratory

Problem representation

- **Regression**: given an input x, we try to predict a (continuous value) y
- **Classification**: given an input x, we try to predict a (discrete class) y

4×5 5×1 4×1



a new row of θ →

DTSI student score ← a new class

… changing the size of θ.

SISL
Stanford Intelligent Systems Laboratory

Problem representation

- **Regression**: given an input x, we try to predict a (continuous value) y
- **Classification**: given an input x, we try to predict a (discrete class) y
- **Image segmentation:** classification for each pixel in an image

Original Image   Semantic Segmentation

BuiltIn

SISL
Stanford Intelligent
Systems Laboratory

AI training is all about optimization

- We want to solve the following numerically

$$\underset{\theta \in \Theta}{\text{minimize}}\ J(\theta) = \sum_{i=1}^{n} (f_\theta(X_i) - Y_i)^2$$

← **Objective example**

SISL
Stanford Intelligent
Systems Laboratory

AI training is all about optimization

- We want to solve the following <u>numerically</u>

$$\underset{\theta \in \Theta}{\text{minimize}}\ J(\theta) = \sum_{i=1}^{n} (f_\theta(X_i) - Y_i)^2$$ ← **Objective example**

- A very popular algorithm is gradient descent.

- The gradient of J w.r.t. θ can be estimated using numerical automated differentiation method.

SISL
Stanford Intelligent Systems Laboratory

Gradient descent

- Recall, our problem

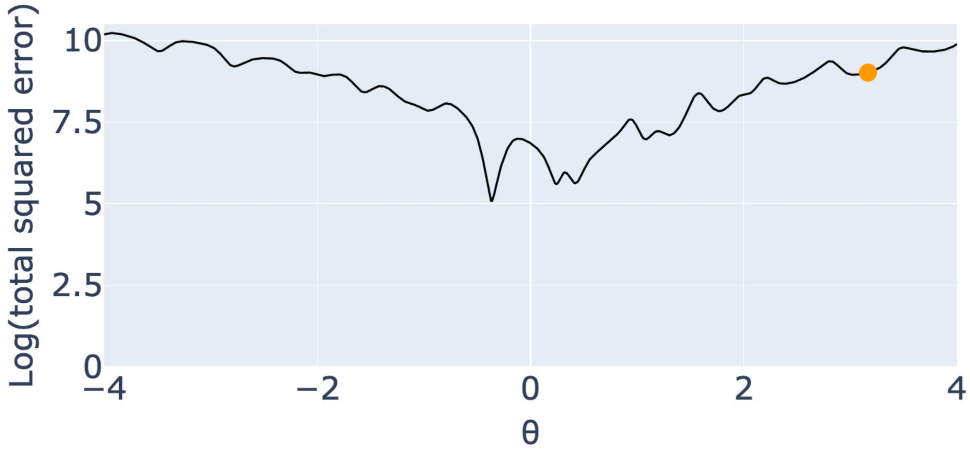
$$\underset{\theta \in \Theta}{\text{minimize}}\ J(\theta) = \sum_{i=1}^{n} \left(f_\theta(X_i) - Y_i\right)^2$$

- Suppose an oracle tells us the value of $J(\theta)$ for $\theta \in \Theta$.

SISL
Stanford Intelligent
Systems Laboratory

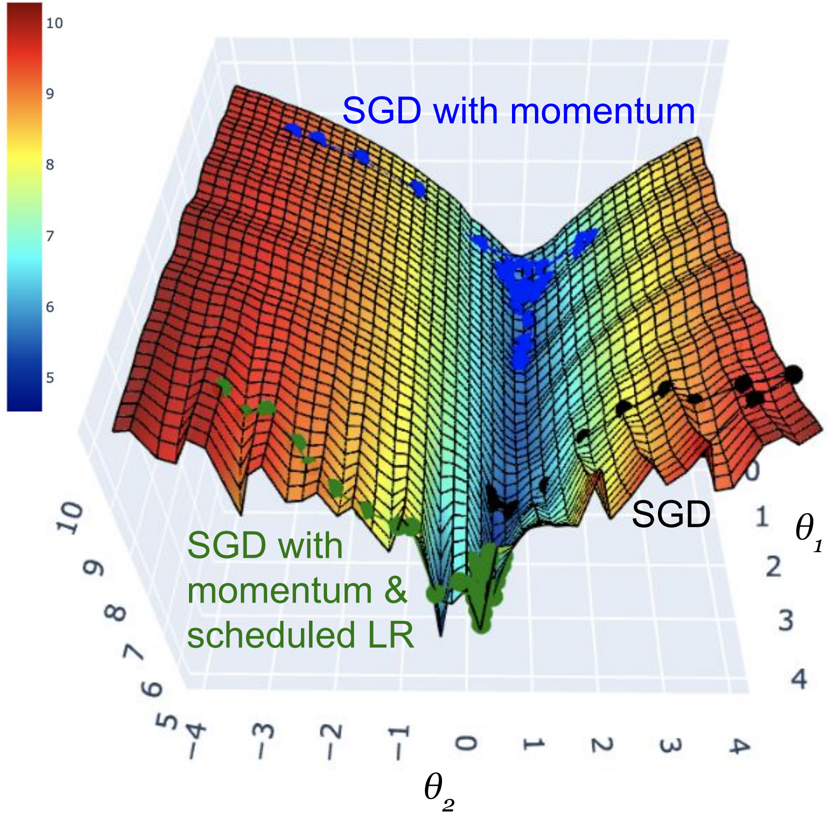
Gradient descent and variants

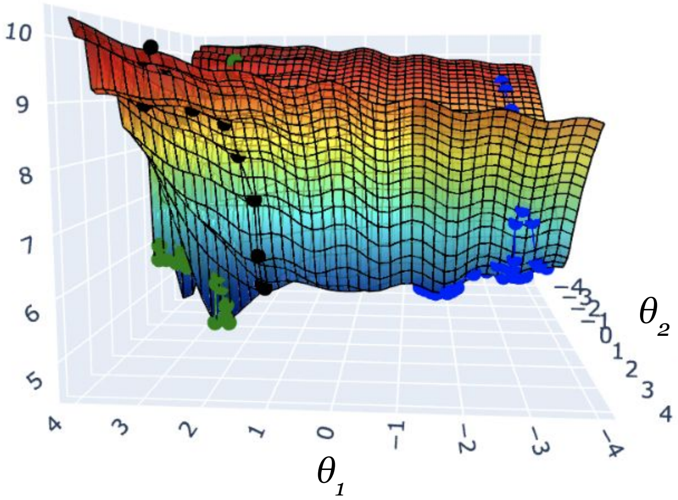
1. Random initializations
2. Batch of samples (stochastic gradient descent)
3. Momentum
4. Scheduled learning rate
5. Overparameterization



SISL
Stanford Intelligent Systems Laboratory

Gradient descent and variants

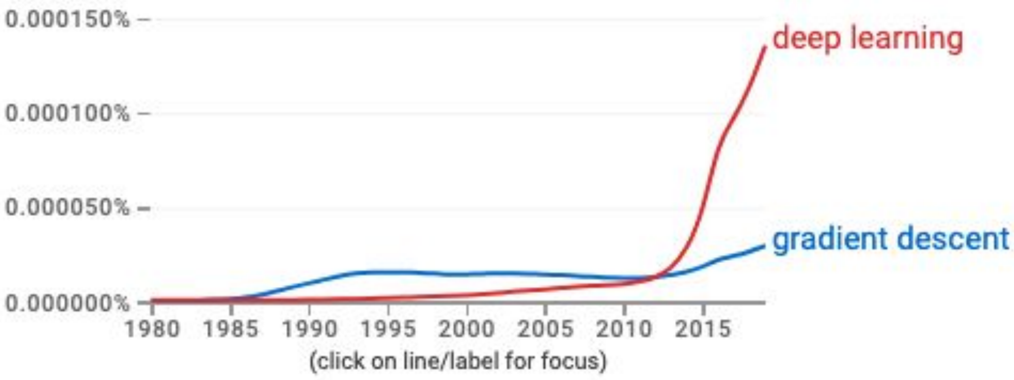




SISL
Stanford Intelligent Systems Laboratory

Gradient descent for deep learning

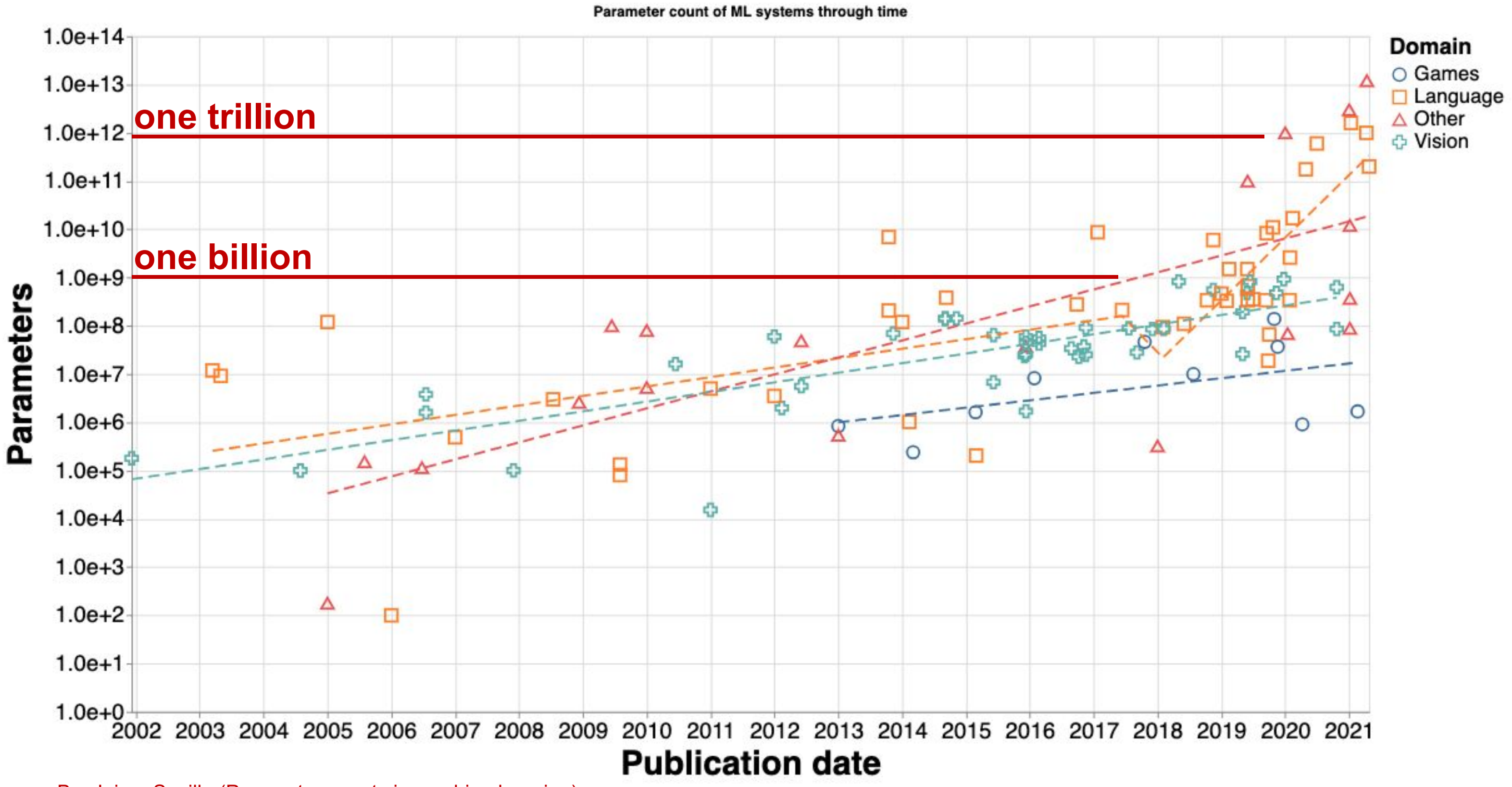
- **Why did I spend time to review gradient descent?**
  Because it's the powerhorse behind deep learning/large model training



Google NGram

SISL
Stanford Intelligent Systems Laboratory

Deep learning parameters size



By: Jaime Sevilla (Parameter counts in machine learning)

SISL
Stanford Intelligent Systems Laboratory

What we've discussed so far

- The methods enabling powerful AI are **numerical optimization**!
- The algorithms use **tricks** to reach a good enough solution
  - randomization (initialization and batching)
  - (meta) heuristics (momentum, LR scheduling)
  - large parameter space (overparameterization)
- **Formulations** include
  - model fitting (regression, classification boundary)
  - falsification and validation (FTA/FMEA, adversarial attack, importance sampling)
  - [tentative] utility maximization (MDP/POMDP)

2. Risk Analysis and Validation

SISL
Stanford Intelligent Systems Laboratory

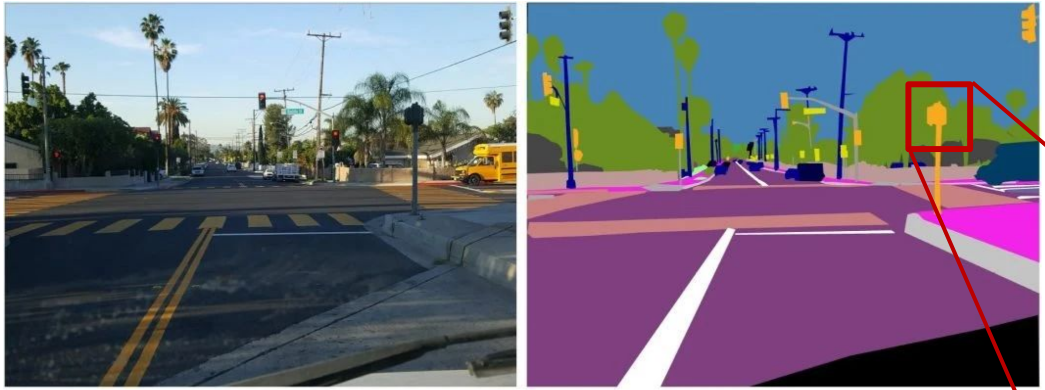
Risk analysis and validation

- If we use an AI component, how do we **manage** the risk?
- Need to know the **kinds of failure** and assess its **severity**.
- Plan for **mitigation strategy** based on the risk (probability × severity).
- Be able to detect when the failure **is about to** occur.

SISL
Stanford Intelligent Systems Laboratory

Risk analysis

- A self-driving car uses camera images to detect and classify traffic signs.
- First, it runs semantic segmentation on an input image.



VisoAI

- Then, it classifies the detected traffic sign.

What are the failure modes of this approach?

Failure Modes and Effects Analysis (FMEA)

- FMEA identifies **all possible failures** in a design, a manufacturing or assembly process, or a product or service.

Inaccurate segmentation of images

Failure mode example: False negatives in detection

Failure mode example: Misclassified traffic signs

	Failure Mode 1	**Failure Mode 2**	**Failure Mode 3**
	Inaccurate image segmentation	Traffic sign detection failure	Traffic sign misclassification
Effect	Failed traffic signs segmentation	Missed traffic sign	Incorrect traffic rule interpretation
Cause	Poor lighting, inaccurate pixel classifier	Environmental camouflage, occlusions	Inadequate training data, brittle model
Severity	High 9	High 9	High 9
Occurrence	Medium 5	Low 3	High 9
Detection	High 3	High 3	High 3
RPN	**135**	**81**	**243**

SISL
Stanford Intelligent Systems Laboratory

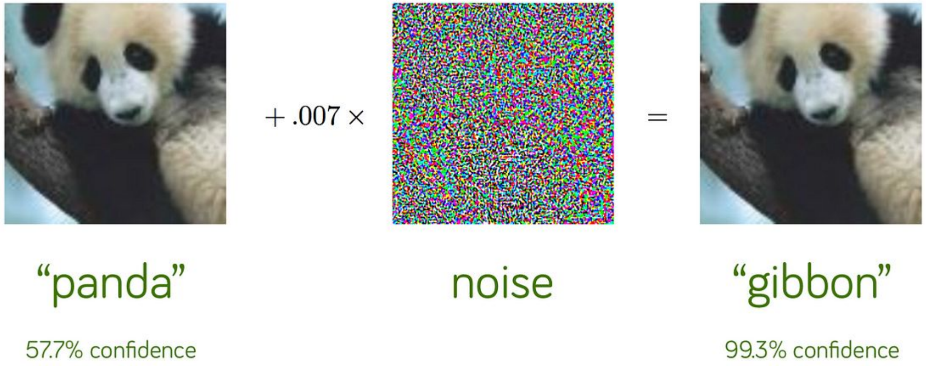
ML systems failure identification

- Domain knowledge (Out-of-Distribution cases)
- Grid search (or more systematic search)
- Adversarial attack

TITLE	CITED BY	YEAR
A survey on safety-critical driving scenario generation—A methodological perspective W Ding, C Xu, M Arief, H Lin, B Li, D Zhao IEEE Transactions on Intelligent Transportation Systems	50	2023

Adversarial attack

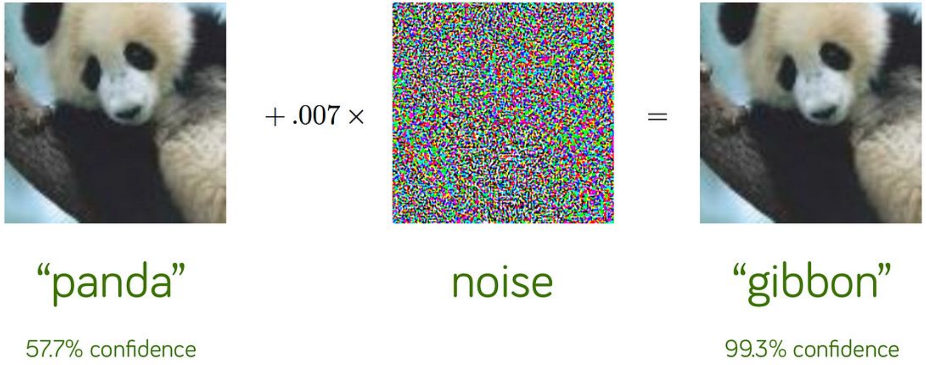
- ML classifier are susceptible to **adversarial examples**
- We can find well-crafted noise to change prediction!

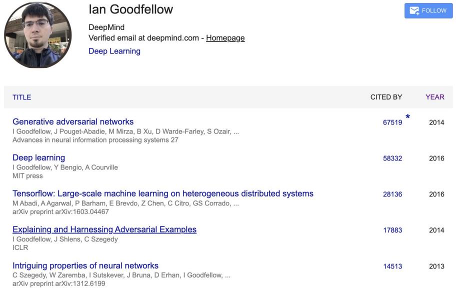


SISL
Stanford Intelligent Systems Laboratory

Adversarial attack

- ML classifier are susceptible to **adversarial examples**
- We can find well-crafted noise to change prediction!



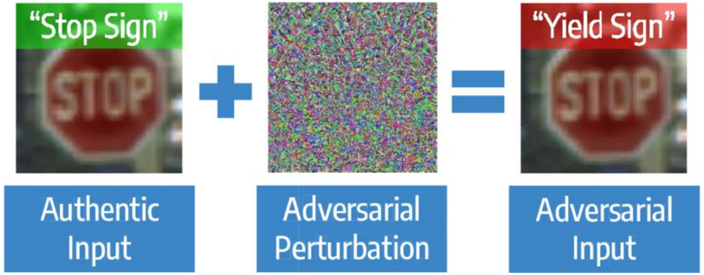


- The process is called **adversarial attack**.

SISL Stanford Intelligent Systems Laboratory

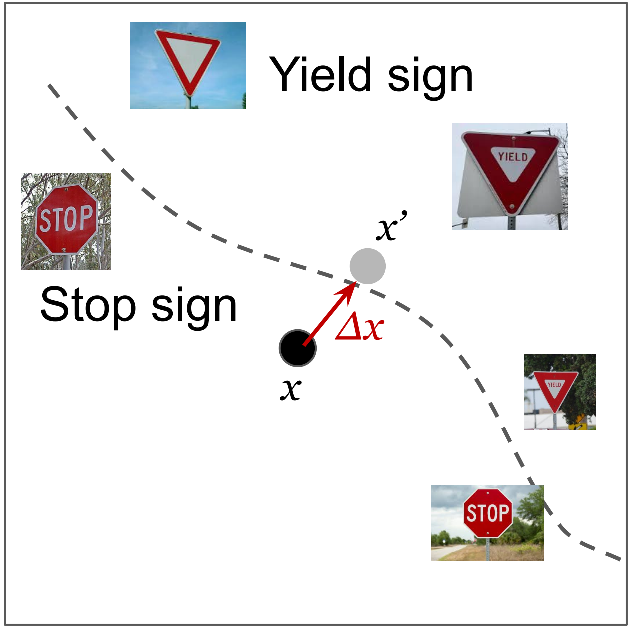
Adversarial attack (formulation)

- Idea: solve for adversarial perturbation Δx that maximize the error



$$\Delta x = \eta \cdot \nabla_x Error(f_\theta(x), y)$$

The gradient step in gradient ascent



SISL
Stanford Intelligent Systems Laboratory

Demo 2: Fool the AI!

Try to make a facial expression that the AI fails to classify

Share your successful attack with DTSI:
https://bit.ly/FooledAI

https://huggingface.co/spaces/mansurarief/DTSI-demo-2

SISL
Stanford Intelligent
Systems Laboratory

Demo 2: Fool the AI!

Try to make a facial expression that the AI fails to classify

Original

Adversarial example

https://huggingface.co/spaces/mansurarief/DTSI-demo-2

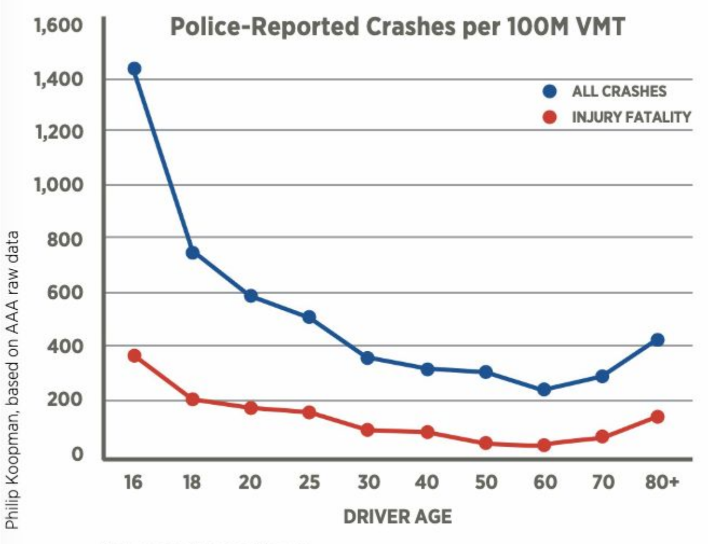
What we've discussed so far

- The methods enabling powerful AI are **numerical optimization**!
- The algorithms use **tricks** to reach a good enough solution
  - randomization (initialization and batching)
  - (meta) heuristics (momentum, LR scheduling)
  - large parameter space (overparameterization)
- **Formulations** include
  - model fitting (regression, classification boundary)
  - falsification and validation (FTA/FMEA, adversarial attack, importance sampling)
  - [tentative] utility maximization (MDP/POMDP, RL)

SISL
Stanford Intelligent
Systems Laboratory

How to compare AI vs human reliability?

- People expect autonomous vehicle safety to be higher than human.
- Standards require critical components to have extremely low failure probability.
    - ISO 26262 (functional safety)
    - ISO 21448 (SOTIF)
    - UL4600 (Safety for the Evaluation of Autonomous Products)



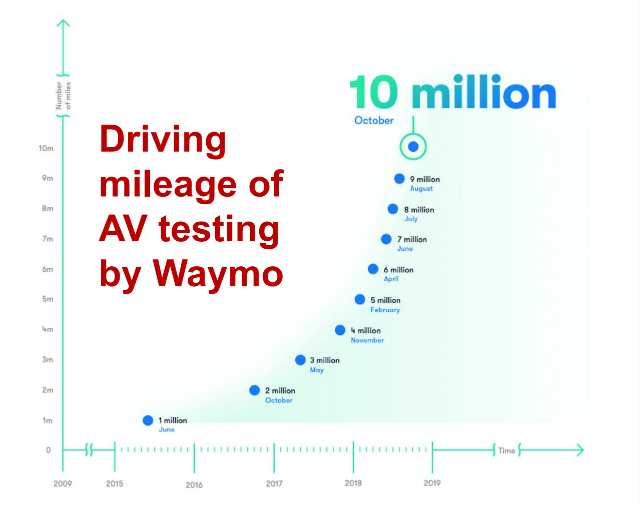
Driver crash rates by age.

Source: SAE Update p. 31

SISL
Stanford Intelligent Systems Laboratory

Autonomous vehicle (AV) testing is inefficient

- **Smaller μ** requires even **larger sample size**.



Source: Waymo

Source: https://www.nhtsa.gov/automated-vehicle-test-tracking-tool

SISL
Stanford Intelligent Systems Laboratory

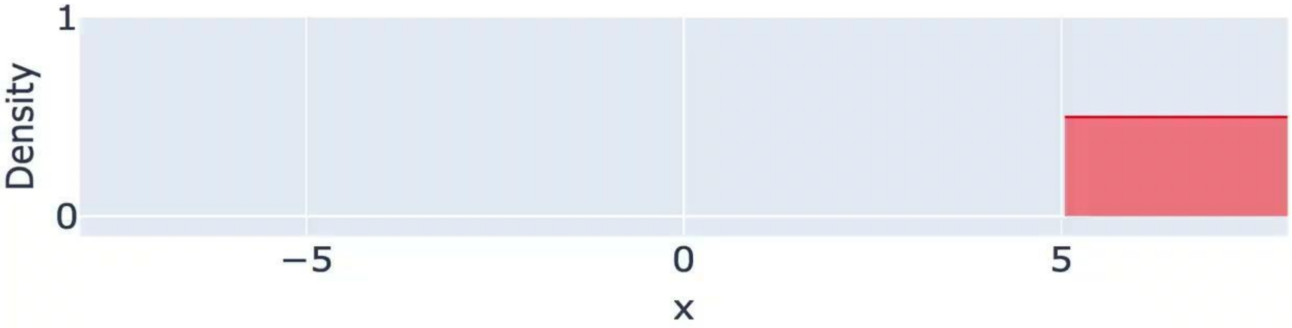
Rarity of traffic crashes (in the US)

- If the **crash rate is μ**, then on average we need **$1/\mu$ samples** to observe the first crash (geometric distribution).
  - E.g. if $\mu = 10^{-5}$, we will need millions of samples to estimate it

SISL
Stanford Intelligent Systems Laboratory

Rarity of traffic crashes (in the US)

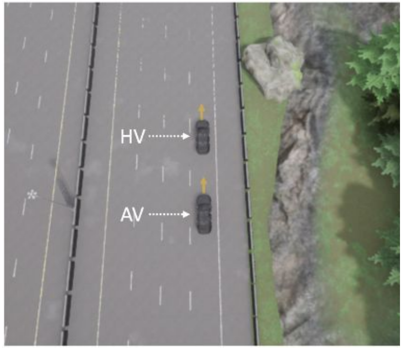
- If the **crash rate is μ**, then on average we need **$1/\mu$ samples** to observe the first crash (geometric distribution).
  - E.g. if $\mu = 10^{-5}$, we will need millions of samples to estimate it
- **Smaller μ** requires even **larger sample size**.
  - E.g. Suppose we try to estimate $\mu = P(X > 5),\ X \sim N(0,1)$

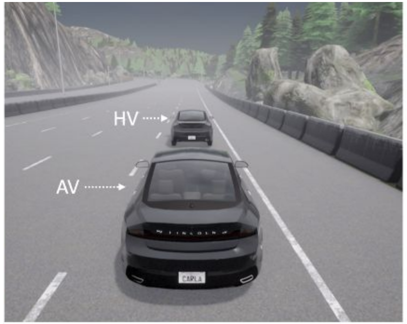


No sample ever hits this "important" region

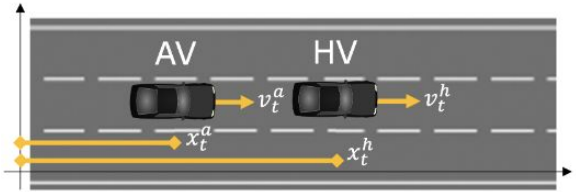
SISL
Stanford Intelligent Systems Laboratory

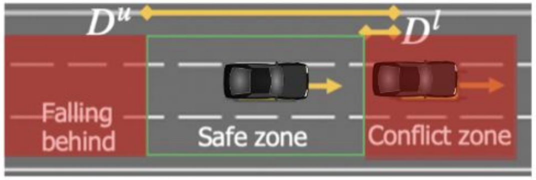
Inefficiency remains an issue in simulations





AV Simple PI Controller (SAE Level 2):





$$a_t^a = a_{t-1}^a + K_p(e_t^{thw} - e_{t-1}^{thw}) + K_i(e_t^{thw} + e_{t-1}^{thw})T_s/2$$

where a_t^a : AV acceleration at time t

e_t^{thw} : target and realization time headway error at time t

K_p, K_i : P and I gain, respectively

T_s : simulation frequency

- Naturalistic simulation takes up to **a month of runtime** to estimate $\mu = 2 \times 10^{-5}$

SISL
Stanford Intelligent
Systems Laboratory

Inefficiency remains an issue in simulations

AV Perception Algorithm (YOLOv5)

Normal cases

Extremely rare (1 in 1 million simulation)

- May take **3 months (estimated) runtime** to estimate smaller $\mu = 1 \times 10^{-6}$

SISL
Stanford Intelligent
Systems Laboratory

Perils of crude sampling technique

- Crude technique sampling is **inadequate** to evaluate rare events (such as crash events in the US)
- Consider estimating a tiny μ with an estimator $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$.
- A small ϵ & high confidence 1-δ

$$\mathbb{P}\left(|\hat{\mu}_n - \mu| > \epsilon\mu\right) \leq \delta$$

is achieved **only when**

$$n \geq \frac{\mathrm{Var}(Y_i)}{\delta\epsilon^2\mu^2}.$$

- Thus, as $\mu \to 0, n \to \infty$.

SISL
Stanford Intelligent Systems Laboratory

Perils of crude sampling technique

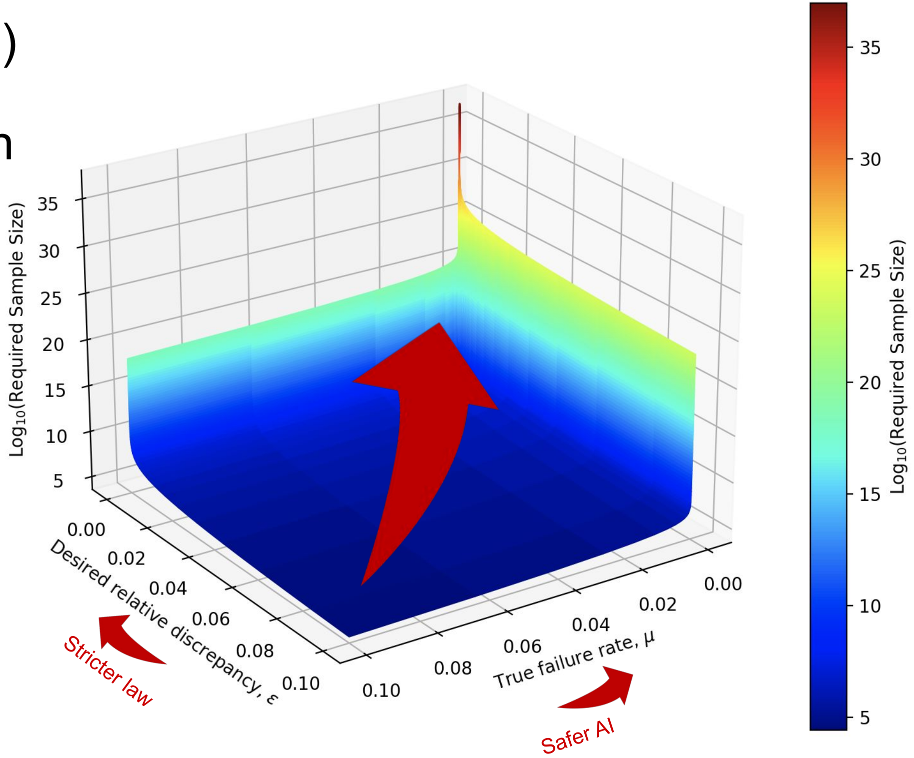
- Crude technique sampling is **inadequate** to evaluate rare events (such as crash events in the US)
- Consider estimating a tiny μ with an estimator $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$.
- A small ϵ & high confidence 1-δ

$$\mathbb{P}\left(|\hat{\mu}_n - \mu| > \epsilon\mu\right) \le \delta$$

is achieved **only when**

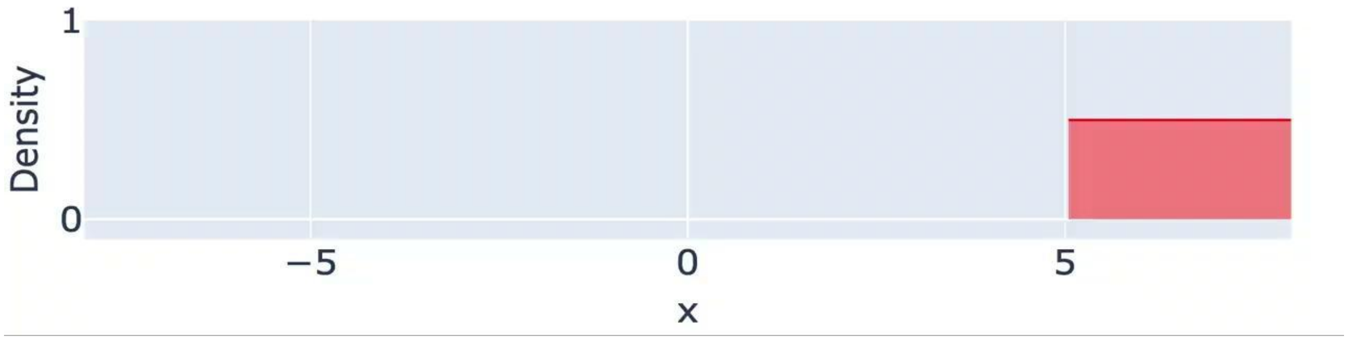
$$n \ge \frac{\mathrm{Var}(Y_i)}{\delta\epsilon^2\mu^2}.$$

- Thus, as $\mu \to 0, n \to \infty$.



SISL
Stanford Intelligent Systems Laboratory

Perils of crude sampling technique

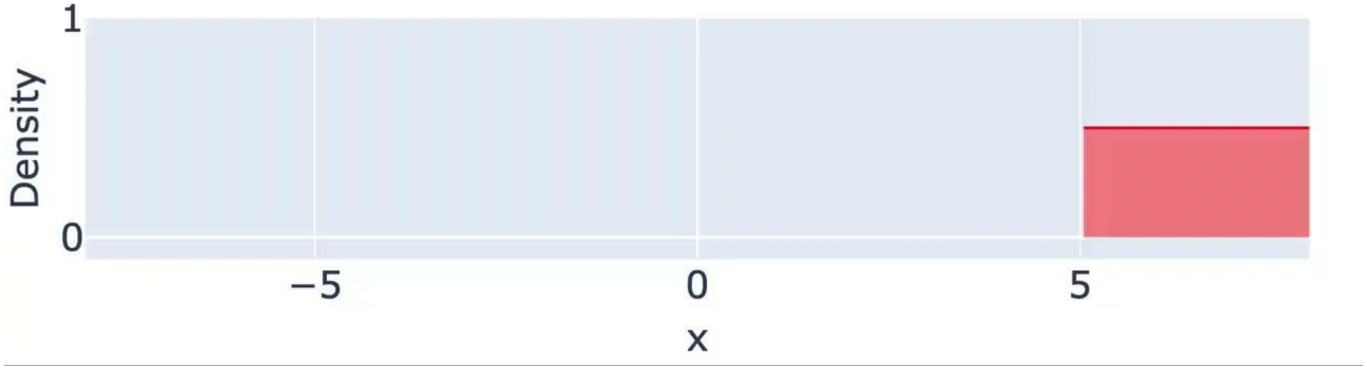


No sample ever hits this "important" region

SISL
Stanford Intelligent Systems Laboratory

Importance Sampling (IS) Idea

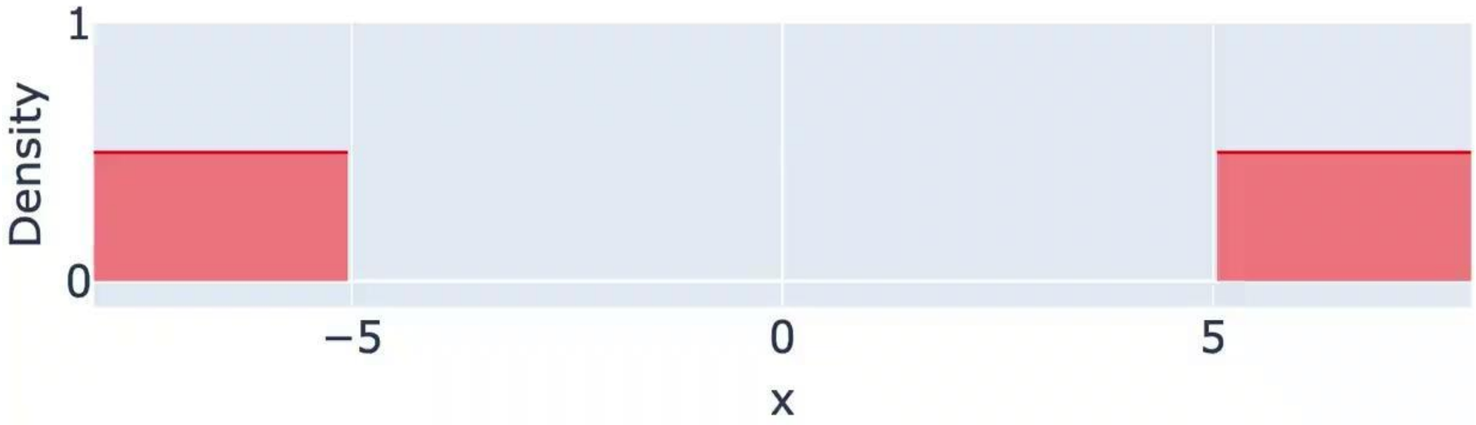
- Use a skewed distribution to sample more "aggressive scenarios"
- The skewing is performed by mean-shifting toward the importance region
- Then debias the result using importance ratio as the weights



SISL
Stanford Intelligent Systems Laboratory

Importance Sampling (IS) Idea

- What if we have multiple important regions? **Use a mixture model!**



SISL
Stanford Intelligent
Systems Laboratory

Importance Sampling (IS) Idea

- **Importance Sampling (IS)** uses proposal distribution $\tilde{p}$ and computes

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left(X_i \in \mathcal{S}_\gamma\right) L(X_i) = \frac{1}{n}\sum_{i=1}^{n} Y_i L(X_i),$$

$$L(X_i) = \frac{p(X_i)}{\tilde{p}(X_i)}.$$ ⇒ called **the likelihood ratio**

SISL
Stanford Intelligent
Systems Laboratory

Importance Sampling (IS) Idea

- IS is provably unbiased

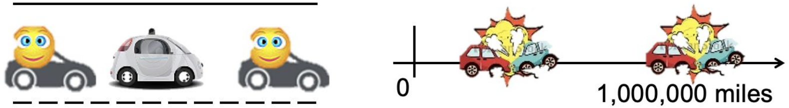
$$
\begin{aligned}
\boxed{\mathbb{E}_{X\sim\tilde{p}}[\hat{\mu}_n]} &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left(X_i\in\mathcal{S}_\gamma\right)L(X_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\boxed{\mathbb{E}}\left[\mathbb{1}\left(X_i\in\mathcal{S}_\gamma\right)\frac{p(X_i)}{\tilde{p}(X_i)}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\boxed{\int_{\mathbb{R}^d}}\mathbb{1}\left(X_i\in\mathcal{S}_\gamma\right)\frac{p(X_i)}{\boxed{\cancel{\tilde{p}(X_i)}}}\boxed{\cancel{\tilde{p}(X_i)}}dX_i \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_{\mathbb{R}^d}\mathbb{1}\left(X_i\in\mathcal{S}_\gamma\right)p(X_i)dX_i \\
&\boxed{= \mu.}
\end{aligned}
$$

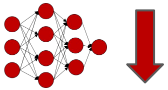
SISL
Stanford Intelligent Systems Laboratory

Importance Sampling (IS) Idea

- **High-level idea**

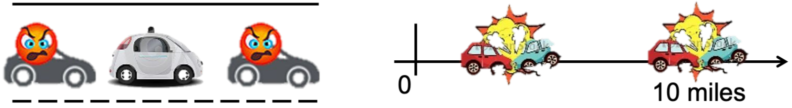
Naturalistic driving conditions:

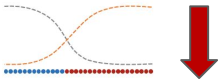




$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} 1(X_i \in \hat{\mathcal{S}}_\gamma)$$

Aggressive driving conditions:





$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} 1(X_i \in \hat{\mathcal{S}}_\gamma)\frac{p(X_i)}{\tilde{p}(X_i)}$$

Unbiased result

Key steps:

1. Start with normal driving
2. Learn the statistical model
3. Bias the statistics toward more aggressive driving
4. Use importance weights to obtain unbiased result
5. Return unbiased statistics

[1] Arief, Mansur, Zhepeng Cen, Zhenyuan Liu, Zhiyuan Huang, Bo Li, Henry Lam, and Ding Zhao. "Certifiable Evaluation for Autonomous Vehicle Perception Systems Using Deep Importance Sampling (Deep IS)." In *Proceedings of* the *2022 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022. [Link]

SISL
Stanford Intelligent Systems Laboratory

Scalable Importance Sampling Algorithms

- **Objective**: deal with extreme rarity and high-dimensional inputs
- **Key ingredients**:
  - Machine learning classifier to approximate the failure set from data
  - Adversarial attack or optimization to find failure case
  - Importance sampling for unbiased and efficient rare failure rate estimation
- **Proposed algorithms**:
  - **Deep IS**: Deep Importance Sampling[1]
  - **Deep-PrAE**: Deep Probabilistic Accelerated Evaluation[2]
  - **CERTIFY**: Computationally Efficient and Robust Evaluation of Safety[3]

[1] Arief, Mansur, Zhepeng Cen, Zhenyuan Liu, Zhiyuan Huang, Bo Li, Henry Lam, and Ding Zhao. "Certifiable Evaluation for Autonomous Vehicle Perception Systems Using Deep Importance Sampling (Deep IS)." In *Proceedings of* the *2022 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022. [Link]

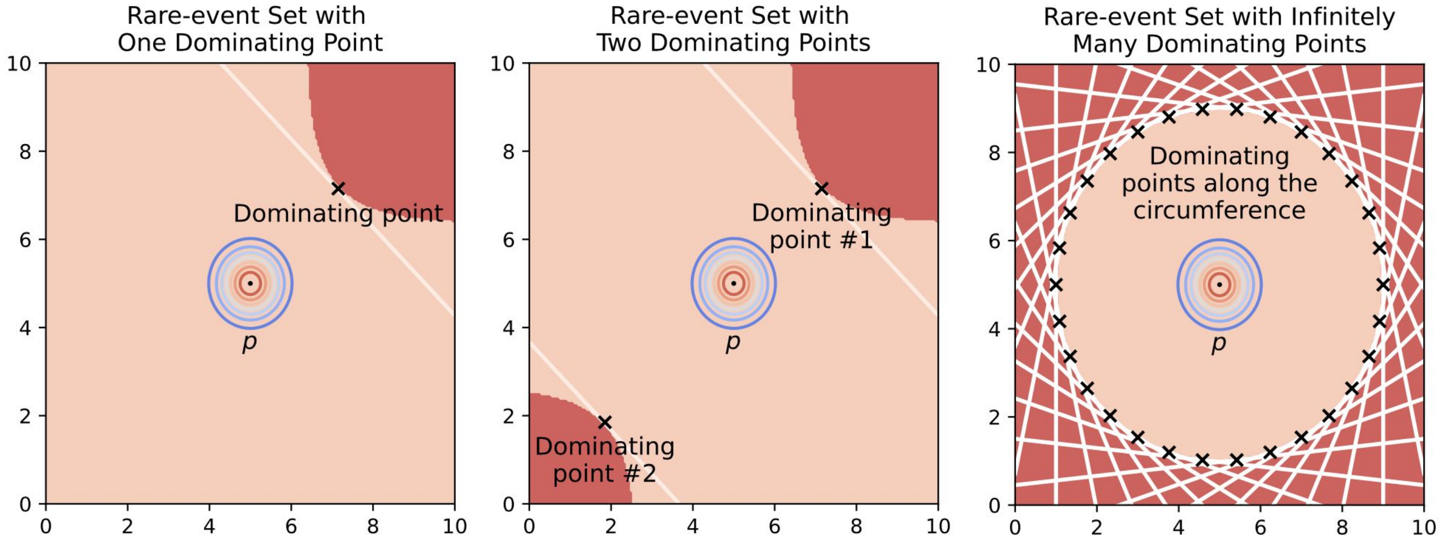
[2] Arief, Mansur, Zhiyuan Huang, Guru Koushik Senthil Kumar, Yuanlu Bai, Shengyi He, Wenhao Ding, Henry Lam, and Ding Zhao. "Deep Probabilistic Accelerated Evaluation: A Certifiable Rare-Event Simulation Methodology for Black-Box Autonomy." In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2021. [Link]

[3] Arief, Mansur, Zhepeng Cen, Huan Zhang, Henry Lam, and Ding Zhao. "CERTIFY: Computationally Efficient Rare-failure Certification of Autonomous Vehicles." *Under review for IEEE T-IV.* [Link]

SISL
Stanford Intelligent Systems Laboratory

Adversarial examples for IS

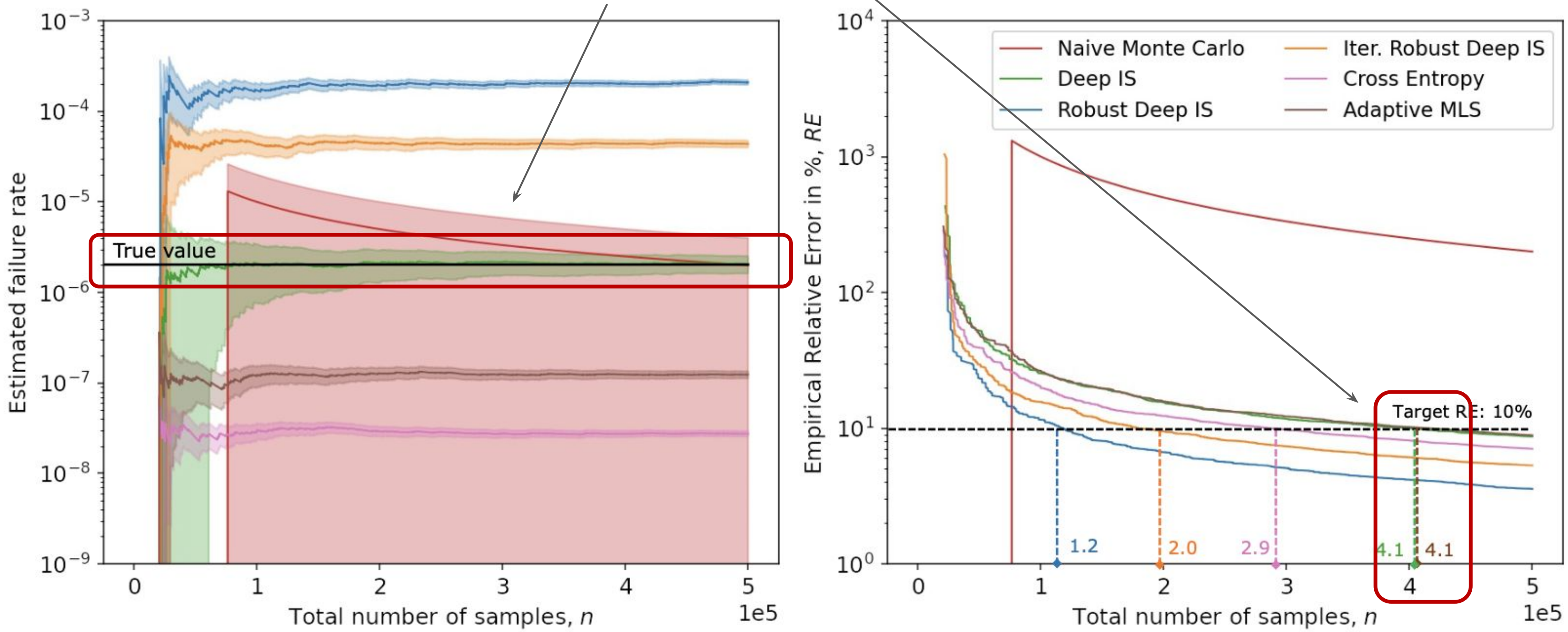
- Adversarial examples can be used as IS mean shift targets (dominating points), i.e. the most likely failure modes naturalistically.



SISL
Stanford Intelligent Systems Laboratory

Numerical experiments

- **Main result: Deep IS is unbiased and sample-efficient**

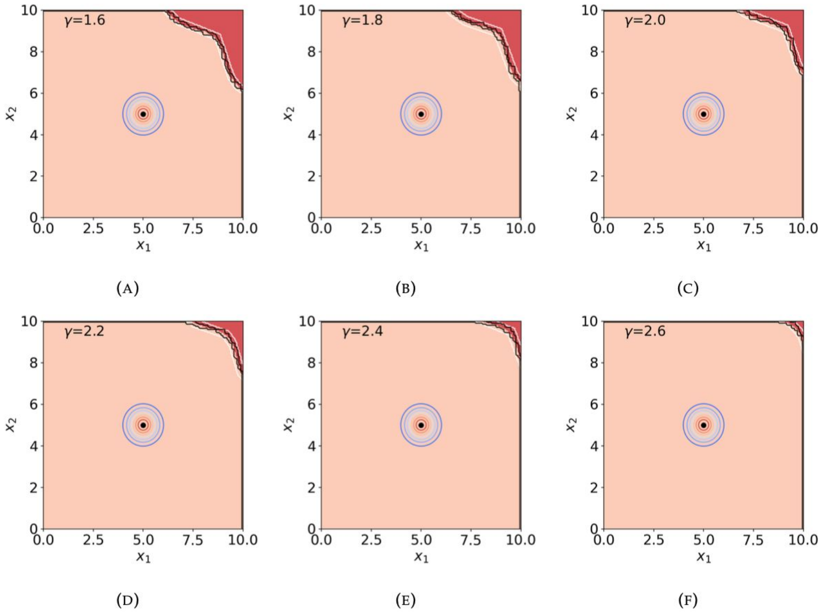


[1]Arief, Mansur, Zhepeng Cen, Zhenyuan Liu, Zhiyuan Huang, Bo Li, Henry Lam, and Ding Zhao. "Certifiable Evaluation for Autonomous Vehicle Perception Systems Using Deep Importance Sampling (Deep IS)." In *Proceedings of* the *2022 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022. [Link]

SISL
Stanford Intelligent Systems Laboratory

Numerical experiments

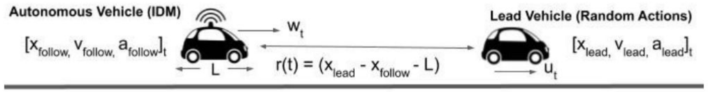
- **Our approach can learn the rough structure of rare failure set**

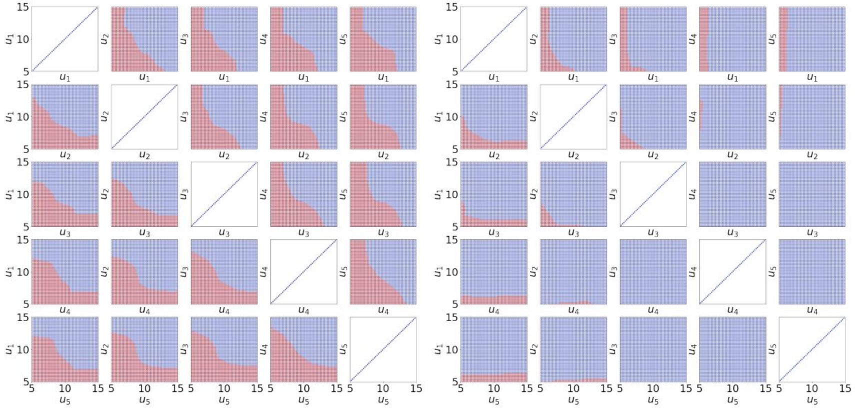


[2]Arief, Mansur, Zhiyuan Huang, Guru Koushik Senthil Kumar, Yuanlu Bai, Shengyi He, Wenhao Ding, Henry Lam, and Ding Zhao. "Deep Probabilistic Accelerated Evaluation: A Certifiable Rare-Event Simulation Methodology for Black-Box Autonomy." In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2021. [Link]

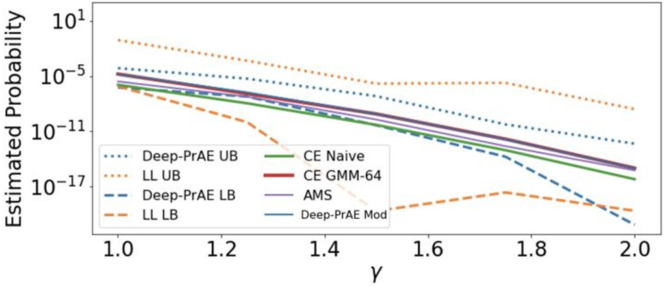
Numerical experiments

- **Autonomy evaluation example: We dominate the efficiency**

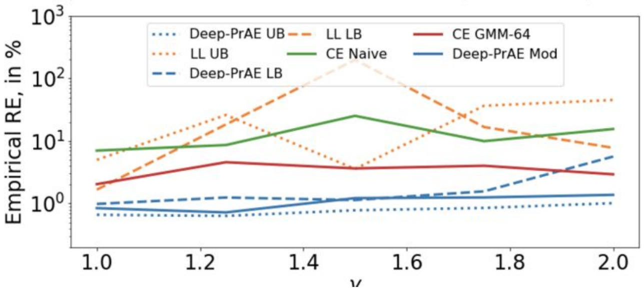




(a) Case $\gamma = 1.0$ (b) Case $\gamma = 2.0$



extreme rarity



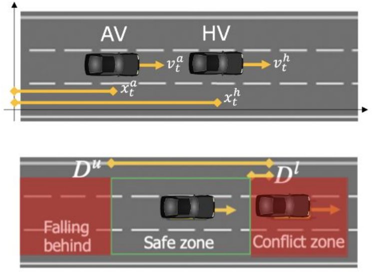
lowest error

[2]Arief, Mansur, Zhiyuan Huang, Guru Koushik Senthil Kumar, Yuanlu Bai, Shengyi He, Wenhao Ding, Henry Lam, and Ding Zhao. "Deep Probabilistic Accelerated Evaluation: A Certifiable Rare-Event Simulation Methodology for Black-Box Autonomy." In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2021. [Link]

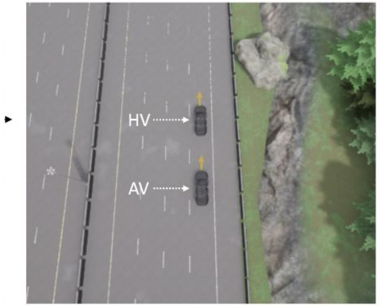
SISL
Stanford Intelligent Systems Laboratory

Numerical experiments

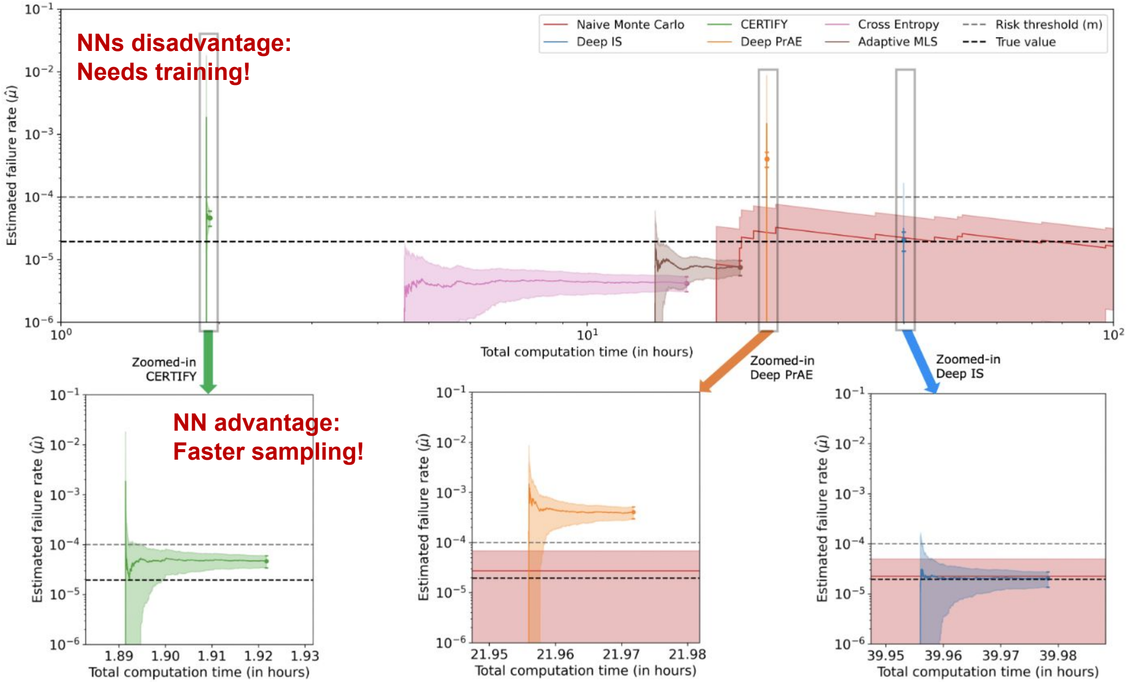
- **Evaluation of a simple ACC under car-following scenario**



(a) Schematic diagram



(b) CARLA topview camera

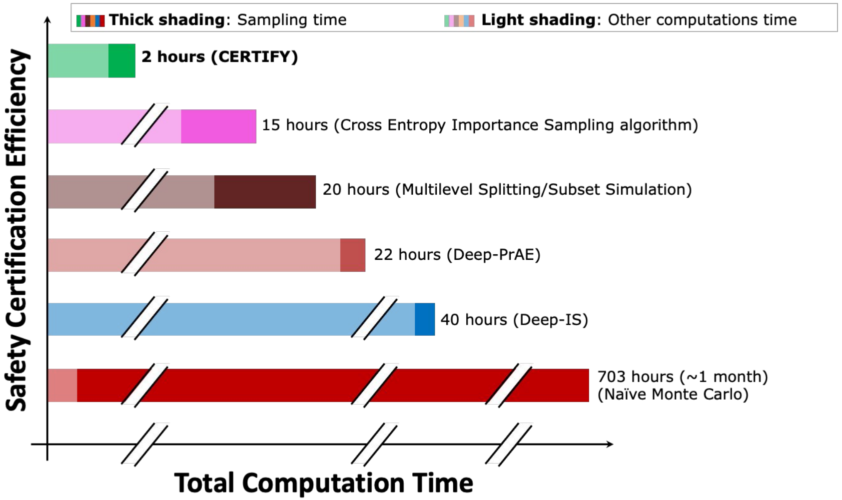


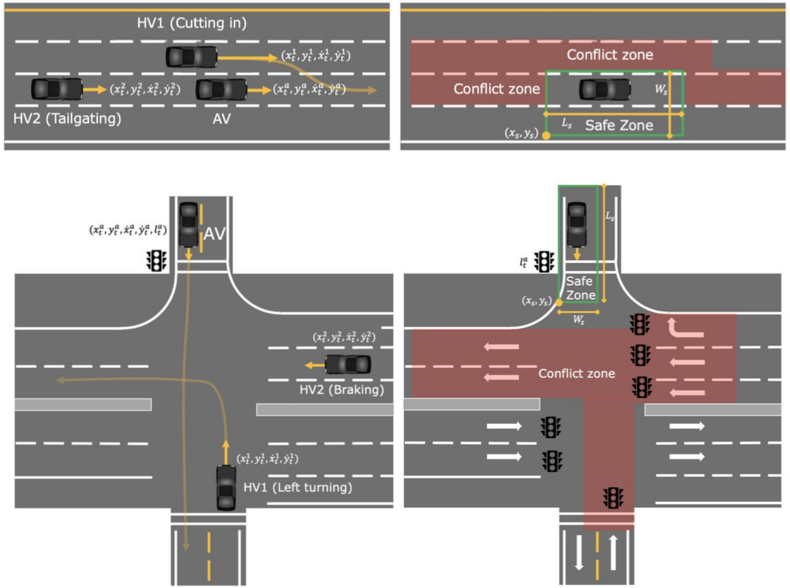
[3]Arief, Mansur, Zhepeng Cen, Huan Zhang, Henry Lam, and Ding Zhao. "CERTIFY: Computationally Efficient Rare-failure Certification of Autonomous Vehicles." *Under review.*

SISL
Stanford Intelligent
Systems Laboratory

Numerical experiments

- **Main result: high compute-efficiency on various scenarios!**





[3]Arief, Mansur, Zhepeng Cen, Huan Zhang, Henry Lam, and Ding Zhao. "CERTIFY: Computationally Efficient Rare-failure Certification of Autonomous Vehicles." *Under review.*

SISL
Stanford Intelligent
Systems Laboratory

What we've discussed so far

- The methods enabling powerful AI are **numerical optimization**!
- The algorithms use **tricks** to reach a good enough solution
  - randomization (initialization and batching)
  - (meta) heuristics (momentum, LR scheduling)
  - large parameter space (overparameterization)
- **Formulations** include
  - model fitting (regression, classification boundary)
  - falsification and validation (FMEA, adversarial attack, importance sampling)
  - [tentative] utility maximization (MDP/POMDP)

3. Planning under Uncertainty (skipped, but can discuss offline)

SISL
Stanford Intelligent Systems Laboratory

Partially Observable Markov Decision Process (POMDP)

- Markov Decision Process (MDP) is a stochastic dynamic programming
- POMDP is MDP with state uncertainty
- Defined by these components

Variable	Description
$\mathcal{S}$	State space
$\mathcal{A}$	Action space
$\mathcal{O}$	Observation space
$T(s' \mid s, a)$	Transition function
$R(s, a)$	Reward function
$O(o \mid s')$	Observation function
$\gamma \in [0, 1]$	Discount factor

SISL
Stanford Intelligent Systems Laboratory

Partially Observable Markov Decision Process (POMDP)

- Objective function:

$$\text{maximize}_\pi\, U(\pi) = \mathbb{E}_{s_0 \sim p}\left[\sum_{t=0}^{T} \gamma^t \underbrace{\sum_{s_{t+1} \in \mathcal{S}} R(s_{t+1}, \pi(b_t)) P(s_{t+1} | s_t, \pi(b_t))}_{\text{Expected reward over possible state transitions}} \Big| s_0\right]$$

- The key difference is policy π in POMDP uses belief b_t as input, not state s_t

SISL
Stanford Intelligent Systems Laboratory

POMDP Example: Crying baby problem

- A simple POMDP with 2 states, 2 actions, and 2 observations

$$\mathcal{S} = \{\texttt{hungry}, \texttt{full}\}$$
$$\mathcal{A} = \{\texttt{feed}, \texttt{ignore}\}$$
$$\mathcal{O} = \{\texttt{crying}, \texttt{quiet}\}$$

- We cannot directly tell if the baby is truly hungry
- We can only observe the crying and update our belief about the true state using this information.

SISL
Stanford Intelligent Systems Laboratory

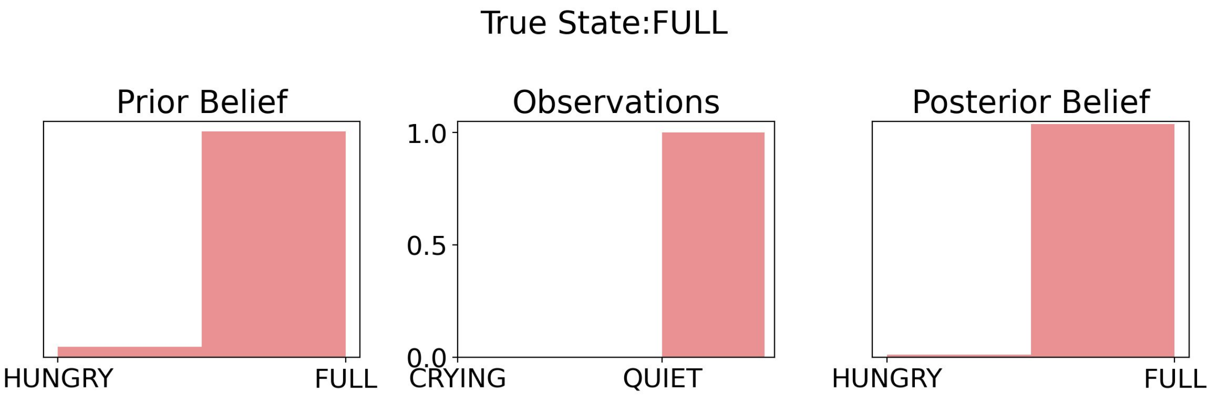
POMDP Example: Crying baby problem

- Suppose we have the following observation model

$$O(\text{crying} \mid \text{hungry}) = 80\%$$
$$O(\text{quiet} \mid \text{hungry}) = 20\%$$

$$O(\text{crying} \mid \text{full}) = 10\%$$
$$O(\text{quiet} \mid \text{full}) = 90\%$$

- We can start with some prior belief and update it as we observe data



SISL
Stanford Intelligent Systems Laboratory

Solving a POMDP

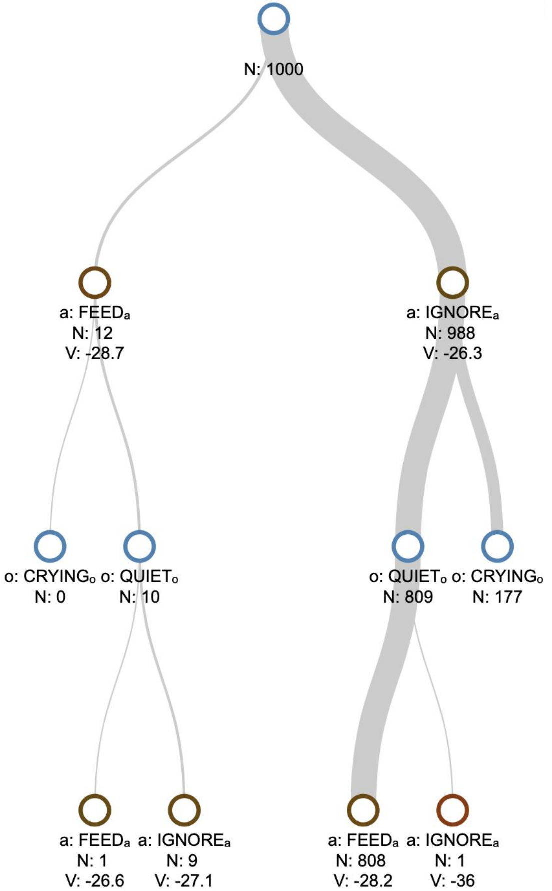
```
using POMDPs, POMDPModelTools, QuickPOMDPs

@enum State hungry full
@enum Action feed ignore
@enum Observation crying quiet

pomdp = QuickPOMDP(
    states       = [hungry, full],  # 𝒮
    actions      = [feed, ignore],  # 𝒜
    observations = [crying, quiet], # 𝒪
    initialstate = [full], # Deterministic
    discount     = 0.9, # γ

    transition = function T(s, a)
        if a == feed
            return SparseCat([hungry, full], [0, 1])
        elseif s == hungry && a == ignore
            return SparseCat([hungry, full], [1, 0])
        elseif s == full && a == ignore
            return SparseCat([hungry, full], [0.1, 0.9])
        end
    end,

    observation = function O(s, a, s′)
        if s′ == hungry
            return SparseCat([crying, quiet], [0.8, 0.2])
        elseif s′ == full
            return SparseCat([crying, quiet], [0.1, 0.9])
        end
    end,

    reward = (s,a)->(s == hungry ? -10 : 0) + (a == feed ? -5 : 0)
)
```

Package	State Spaces	Actions Spaces	Observation Spaces
QMDP.jl	Discrete	Discrete	Discrete
FIB.jl	Discrete	Discrete	Discrete
BeliefGridValueIteration.jl	Discrete	Discrete	Discrete
SARSOP.jl	Discrete	Discrete	Discrete
BasicPOMCP.jl	Continuous	Discrete	Discrete
ARDESPOT.jl	Continuous	Discrete	Discrete
MCVI.jl	Continuous	Discrete	Continuous
POMDPSolve.jl	Discrete	Discrete	Discrete
IncrementalPruning.jl	Discrete	Discrete	Discrete
POMCPOW.jl	Continuous	Continuous	Continuous
AEMS.jl	Discrete	Discrete	Discrete
PointBasedValueIteration.jl	Discrete	Discrete	Discrete

SISL
Stanford Intelligent Systems Laboratory

Solving a POMDP

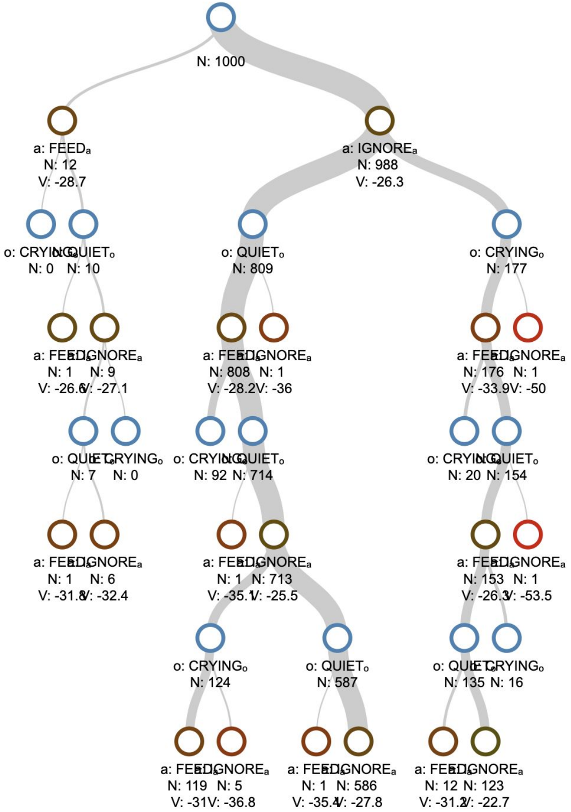
```
using POMDPs, POMDPModelTools, QuickPOMDPs

@enum State hungry full
@enum Action feed ignore
@enum Observation crying quiet

pomdp = QuickPOMDP(
    states       = [hungry, full],  # 𝒮
    actions      = [feed, ignore],  # 𝒜
    observations = [crying, quiet], # 𝒪
    initialstate = [full], # Deterministic
    discount     = 0.9, # γ

    transition = function T(s, a)
        if a == feed
            return SparseCat([hungry, full], [0, 1])
        elseif s == hungry && a == ignore
            return SparseCat([hungry, full], [1, 0])
        elseif s == full && a == ignore
            return SparseCat([hungry, full], [0.1, 0.9])
        end
    end,

    observation = function O(s, a, s′)
        if s′ == hungry
            return SparseCat([crying, quiet], [0.8, 0.2])
        elseif s′ == full
            return SparseCat([crying, quiet], [0.1, 0.9])
        end
    end,

    reward = (s,a)->(s == hungry ? -10 : 0) + (a == feed ? -5 : 0)
)
```



SISL Stanford Intelligent Systems Laboratory

Solving a POMDP

```
using POMDPs, POMDPModelTools, QuickPOMDPs

@enum State hungry full
@enum Action feed ignore
@enum Observation crying quiet

pomdp = QuickPOMDP(
    states       = [hungry, full],  # 𝒮
    actions      = [feed, ignore],  # 𝒜
    observations = [crying, quiet], # 𝒪
    initialstate = [full], # Deterministic
    discount     = 0.9, # γ

    transition = function T(s, a)
        if a == feed
            return SparseCat([hungry, full], [0, 1])
        elseif s == hungry && a == ignore
            return SparseCat([hungry, full], [1, 0])
        elseif s == full && a == ignore
            return SparseCat([hungry, full], [0.1, 0.9])
        end
    end,

    observation = function O(s, a, s′)
        if s′ == hungry
            return SparseCat([crying, quiet], [0.8, 0.2])
        elseif s′ == full
            return SparseCat([crying, quiet], [0.1, 0.9])
        end
    end,

    reward = (s,a)->(s == hungry ? -10 : 0) + (a == feed ? -5 : 0)
)
```



SISL
Stanford Intelligent Systems Laboratory

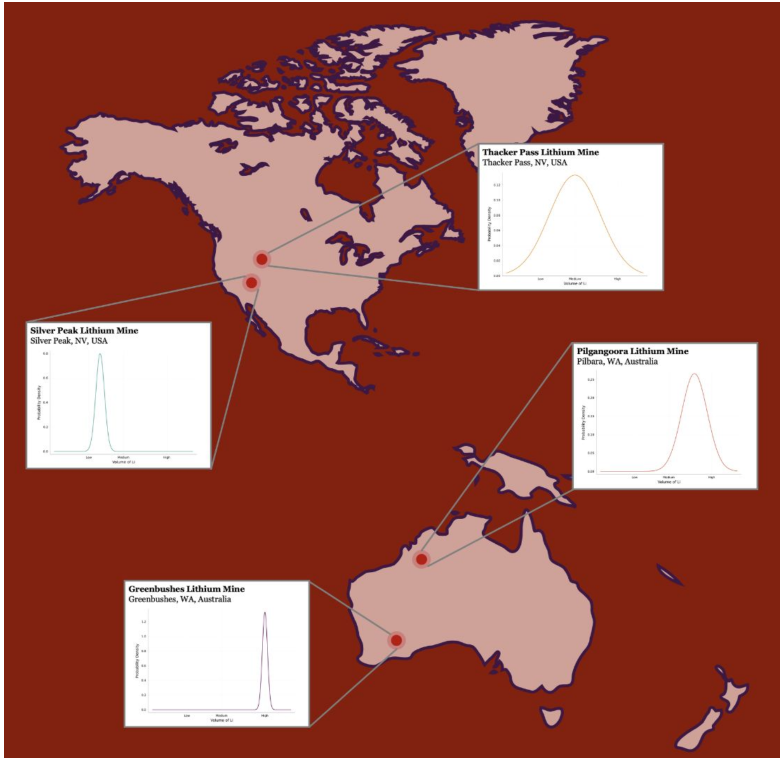
On-going project: StrokePOMDP

- State: aneurysm{T, F}, AVM{T, F}, occ{T, F}, time{0:24}
- Action: Send_home, Observe, MRA, DSA, Surgery
- Observation: Siriraj_score, CT_score
- Reward:
  - penalty for unnecessary observe, MRA, DSA, surgery
  - penalty for lengthy treatment (time > 12)
  - penalty for sending home sick patient
  - reward for effective MRA, DSA, surgery

SISL
Stanford Intelligent
Systems Laboratory

On-going project: LiSC_POMDP

- State: mineral deposits volume{R^+}
- Action: Explore1, …, ExploreN,
  Mine1, …, MineN
- Observation: mineral deposit estimate where exploration occurs
- Reward:
  - Reward for delayed mining for as long as possible, WHILE meeting target demand
  - Penalty for emission when mining locally



SISL
Stanford Intelligent Systems Laboratory

What we've discussed so far

- The methods enabling powerful AI are **numerical optimization**!
- The algorithms use **tricks** to reach a good enough solution
  - randomization (initialization and batching)
  - (meta) heuristics (momentum, LR scheduling)
  - large parameter space (overparameterization)
- **Formulations** include
  - model fitting (regression, classification boundary)
  - falsification and validation (FTA/FMEA, adversarial attack, deep importance sampling)
  - [tentative] utility maximization (MDP/POMDP)

SISL
Stanford Intelligent
Systems Laboratory

How to improve AI?

We can use the found failure modes to retrain our AI agent.

Enhancing Visual Perception in Novel Environments via Incremental Data Augmentation Based on Style Transfer

Abhibha Gupta[1], Rully Agus Hendrawan[2], Mansur Arief[3]

Abstract—**The deployment of autonomous agents in real-world scenarios is challenged by "unknown unknowns", i.e. novel unexpected environments not encountered during training, such as degraded signs. While existing research focuses on anomaly detection and class imbalance, it often fails to address truly novel scenarios. Our approach enhances visual perception by leveraging the Variational Prototyping Encoder (VPE) to adeptly identify and handle novel inputs, then incrementally augmenting data using neural style transfer to enrich underrepresented data. By comparing models trained solely on original datasets with those trained on a combination of original and augmented datasets, we observed a notable improvement in the performance of the latter. This underscores the critical role of data augmentation in enhancing model robustness. Our findings suggest the potential benefits of incorporating generative models for domain-specific augmentation strategies.**

Fig. 1: Examples of degraded traffic signs in the real-world

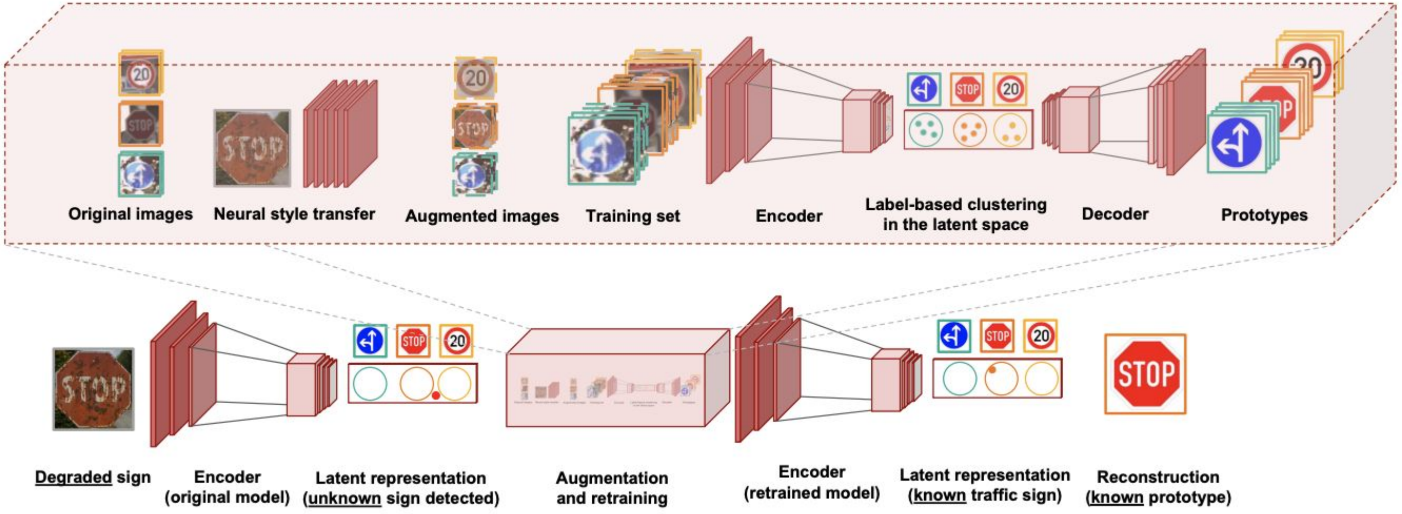
examples of the underrepresented class are available in the training set. In contrast, unknowns emerge when training data are not available for certain cases in the real world [13]. For instance, a traffic sign that has been heavily invaded by rust may not be present in the training set, and such cases will eventually occur during deployment. Arguably, the

Gupta, Abhibha, Rully Agus Hendrawan, and Mansur Arief. "Enhancing Visual Perception in Novel Environments via Incremental Data Augmentation Based on Style Transfer." *arXiv preprint arXiv:2309.08851* (2023).

SISL
Stanford Intelligent
Systems Laboratory

How to improve AI?

We can use the found failure modes to retrain our AI agent.

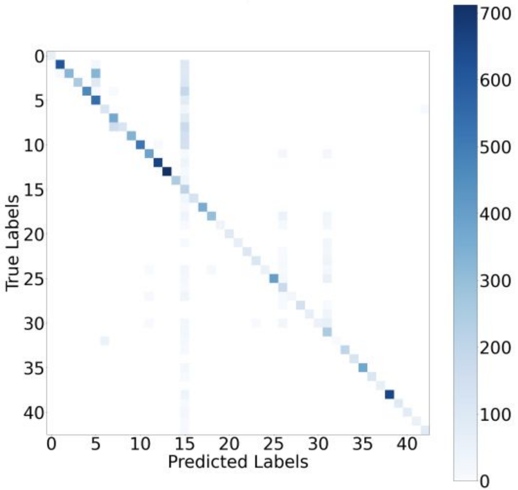


Gupta, Abhibha, Rully Agus Hendrawan, and Mansur Arief. "Enhancing Visual Perception in Novel Environments via Incremental Data Augmentation Based on Style Transfer." *arXiv preprint arXiv:2309.08851* (2023).

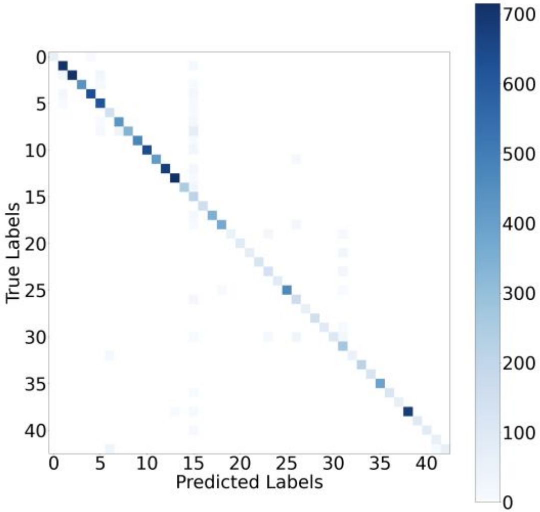
SISL
Stanford Intelligent
Systems Laboratory

How to improve AI?

We can use the found failure modes to retrain our AI agent.



Confusion matrix when AV sees degraded traffic signs



Confusion matrix post retraining

Gupta, Abhibha, Rully Agus Hendrawan, and Mansur Arief. "Enhancing Visual Perception in Novel Environments via Incremental Data Augmentation Based on Style Transfer." *arXiv preprint arXiv:2309.08851* (2023).

SISL
Stanford Intelligent
Systems Laboratory

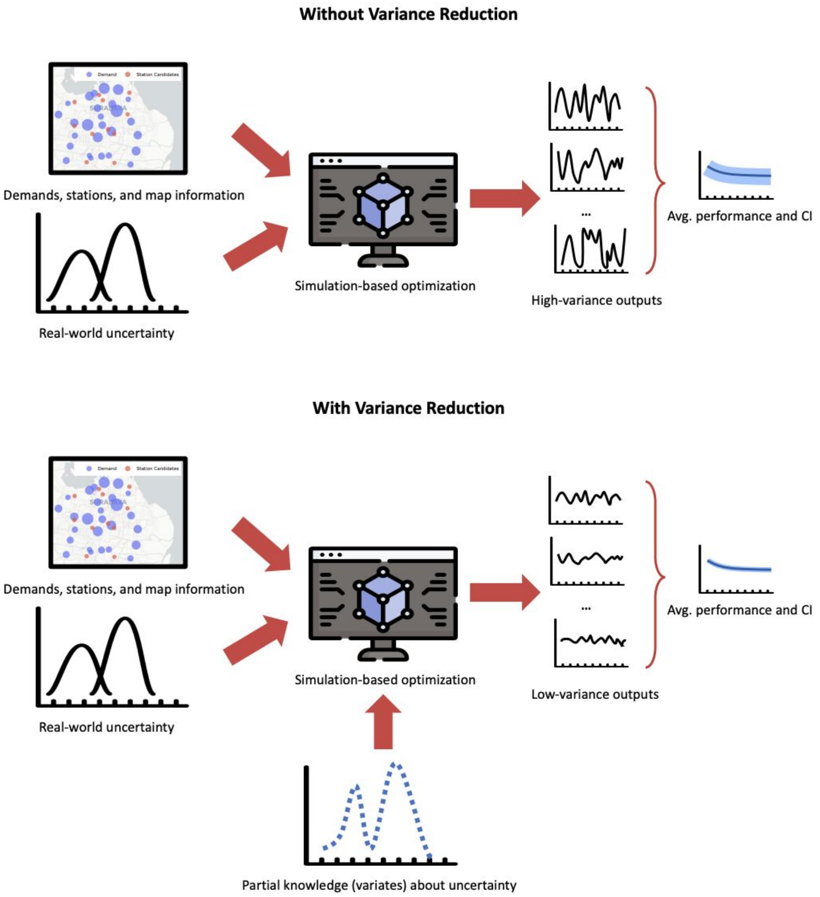
Another application

How to place AV charging stations if we have (rare) electricity outage?

A Robust and Efficient Optimization Model for Electric Vehicle Charging Stations in Developing Countries under Electricity Uncertainty

Mansur M. Arief[a,*], Yan Akhra[b], Iwan Vanany[b]

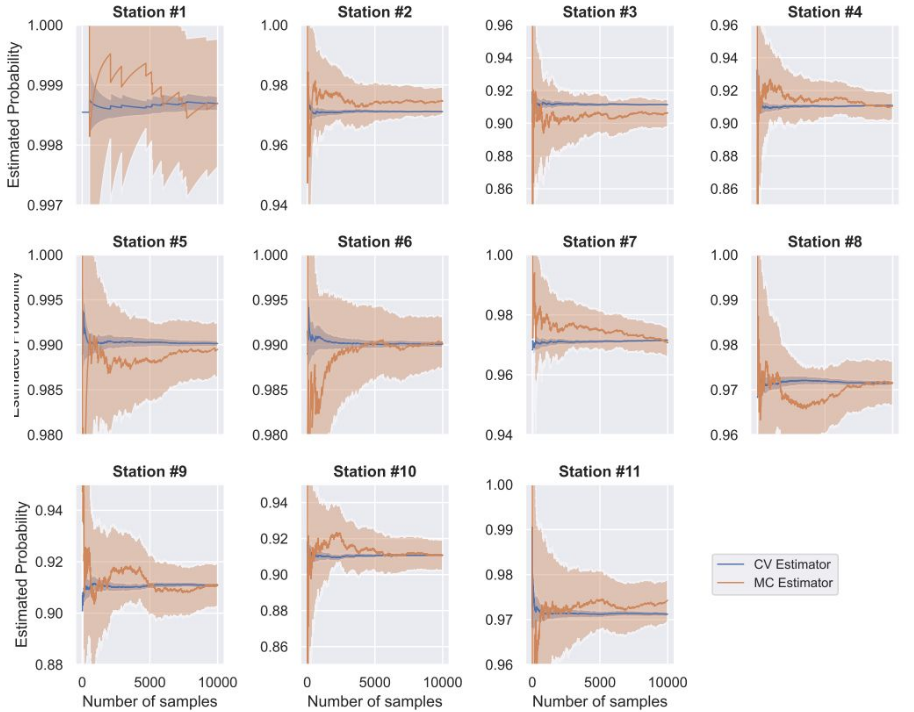
[a] *Department of Aeronautics and Astronautics Engineering, Stanford University, 450 Serra Mall, Stanford, 94305, CA, USA*
[b] *Department of Industrial and Systems Engineering, Institut Teknologi Sepuluh Nopember, Sukolilo, Surabaya, 60111, East Java, Indonesia*



SISL
Stanford Intelligent
Systems Laboratory

Another application

How to place AV charging stations if we have (rare) electricity outage?



A Robust and Efficient Optimization Model for Electric Vehicle Charging Stations in Developing Countries under Electricity Uncertainty

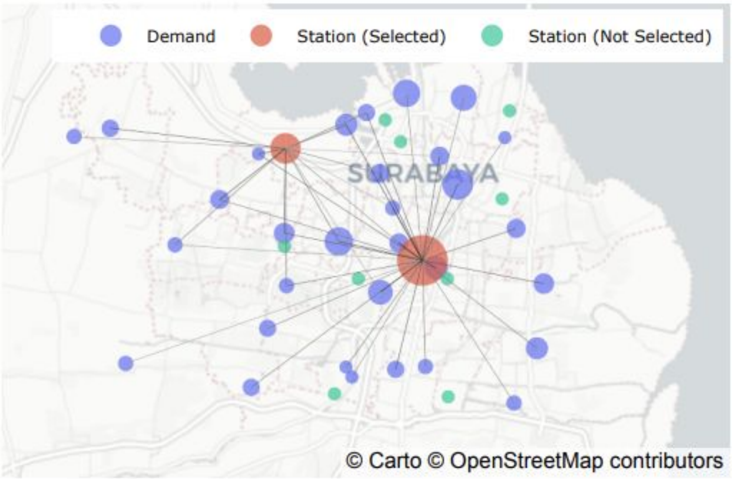
Mansur M. Arief[a,*], Yan Akhra[b], Iwan Vanany[b]

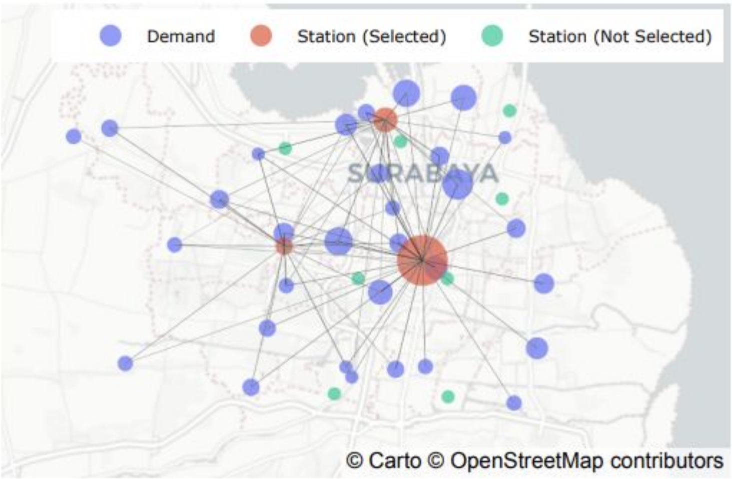
[a] *Department of Aeronautics and Astronautics Engineering, Stanford University, 450 Serra Mall, Stanford, 94305, CA, USA*
[b] *Department of Industrial and Systems Engineering, Institut Teknologi Sepuluh Nopember, Sukolilo, Surabaya, 60111, East Java, Indonesia*

SISL
Stanford Intelligent Systems Laboratory

Another application

How to place AV charging stations if we have (rare) electricity outage?

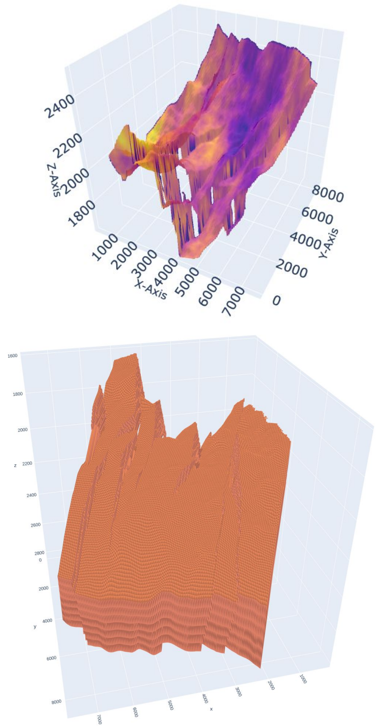


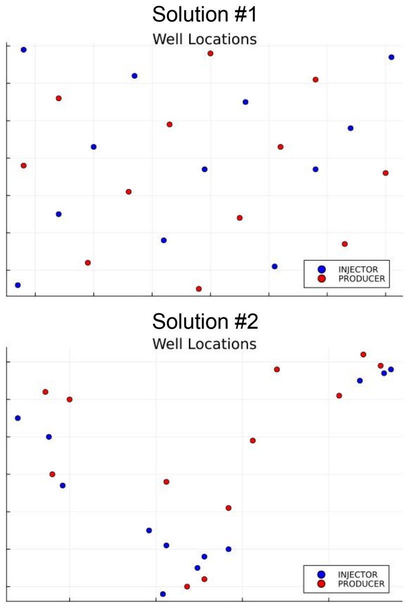


SISL
Stanford Intelligent Systems Laboratory

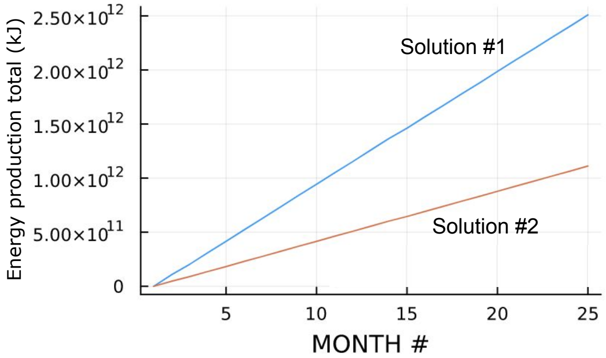
Other "new" application

Where to build geothermal wells in a reservoir?





Evaluate solutions



Summary

- Numerous ISE methods are actually used in AI development
  - mathematical modeling
  - training and validating DL models (numerical optimization)
  - uncertainty quantification (numerical simulation)
  - risk analysis (FMEA, FTA, HAZOP)
  - planning under uncertainty (dynamic programming, time value of money)
- ISE researchers need to engage in interdisciplinary studies
  - intelligent systems, robotics, manufacturing and supply chains
  - sustainability and energy
- AI research area widely open for ISE graduates:
  - AI design, monitoring, deployment, and post-operative
  - AI-human teaming, mixed-autonomy systems
  - AI safety and sustainability

SISL
Stanford Intelligent Systems Laboratory

Research Areas

VERIFICATION & VALIDATION

Development of efficient verification and validation algorithms for autonomous systems.

1. Ding, Wenhao, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, Ding Zhao. "A Survey on Safety-Critical Driving Scenario Generation—A Methodological Perspective." *T-ITS, 2023.* https://ieeexplore.ieee.org/abstract/document/10089194
2. Arief, Mansur. "Certifiable Evaluation for Safe Intelligent Autonomy." *Carnegie Mellon University, 2023.* https://www.proquest.com/openview/45f55565d4810a203cc28fc50dd878a6
3. Arief, Mansur, Zhepeng Cen, Zhenyuan Liu, Zhiyuan Huang, Bo Li, Henry Lam, and Ding Zhao. "Certifiable Evaluation for Autonomous Vehicle Perception Systems Using Deep Importance Sampling (Deep IS)." *ITSC, 2022.* https://ieeexplore.ieee.org/abstract/document/9922202
4. Arief, Mansur, Yuanlu Bai, Wenhao Ding, Shengyi He, Zhiyuan Huang, Henry Lam, and Ding Zhao. "Certifiable Deep Importance Sampling for Rare-Event Simulation of Black-Box Systems." *Under Review.* https://arxiv.org/abs/2111.02204
5. Arief, Mansur, Zhiyuan Huang, Guru Koushik Senthil Kumar, Yuanlu Bai, Shengyi He, Wenhao Ding, Henry Lam, and Ding Zhao. "Deep Probabilistic Accelerated Evaluation: A Certifiable Rare-Event Simulation Methodology for Black-Box Autonomy. "*AISTATS, 2021.* https://proceedings.mlr.press/v130/arief21a/arief21a.pdf
6. Chen, Rui, Mansur Arief, Weiyang Zhang, and Ding Zhao. "How to Evaluate Proving Grounds for Self-Driving? A Quantitative Approach." *T-ITS, 2020.* https://ieeexplore.ieee.org/document/9094370
7. Huang, Zhiyuan, Mansur Arief, Henry Lam, and Ding Zhao. "Evaluation Uncertainty in Data-Driven Self-Driving Testing." *ITSC, 2019.* https://ieeexplore.ieee.org/abstract/document/8917406
8. Arief, Mansur, Peter Glynn, and Ding Zhao. "An Accelerated Approach to Safely and Efficiently Test Pre-production Autonomous Vehicles on Public Streets." *ITSC, 2018.* https://ieeexplore.ieee.org/document/9094370

SISL
Stanford Intelligent
Systems Laboratory

Research Areas

AUTONOMOUS DRIVING PERCEPTION

Works that design robust perception systems for autonomous driving applications.

1. Abdussyukur, Hafizh, Mahmud Dwi Sulistiyo, Ema Rachmawati, Mansur Arief, Gamma Kosala. "Semantic Segmentation for Identifying Road Surface Damages Using Lightweight Encoder-Decoder Network." *ICACNIS, 2022*. https://ieeexplore.ieee.org/abstract/document/10056030
2. Arief, Hasan Asy'ari, Mansur Arief, Guilin Zhang, Zuxin Liu, Manoj Bhat, Ulf Geir Indahl, Håvard Tveite, and Ding Zhao. "SAnE: Smart Annotation and Evaluation Tools for Point Cloud Data." *IEEE Access, 2020*. https://ieeexplore.ieee.org/iel7/6287639/8948470/09143095.pdf
3. Liu, Zuxin, Mansur Arief, and Ding Zhao. "Where Should We Place LiDARs on the Autonomous Vehicle? An Optimal Design Approach." *ICRA, 2019*. https://ieeexplore.ieee.org/document/8793619
4. Arief, Hasan Asy`ari, Mansur Arief, Manoj Bhat, Ulf Geir Indahl, Håvard Tveite, and Ding Zhao. "Density-Adaptive Sampling for Heterogeneous Point Cloud Object Segmentation in Autonomous Vehicle Applications." *CVPR Workshops, 2019*. https://openaccess.thecvf.com/content_CVPRW_2019/papers/UG2+%20Prize%20Challenge/Arief_Density-Adaptive_Sampling_for_Heterogeneous_Point_Cloud_Object_Segmentation_in_Autonomous_CVPRW_2019_paper.pdf

SISL
Stanford Intelligent
Systems Laboratory

Research Areas

EV & INFRASTRUCTURE

Studies focused on vehicle electrification and infrastructure designs.

1. Arief, Mansur, Yan Akhra, Iwan Vanany. "A Robust and Efficient Optimization Model for Electric Vehicle Charging Stations in Developing Countries under Electricity Uncertainty." *Under Review.* https://arxiv.org/abs/2307.05470
2. Amilia, Nissa, Zulkifli Palinrungi, Iwan Vanany, Mansur Arief. "Designing an Optimized Electric Vehicle Charging Station Infrastructure for Urban Area: A Case Study from Indonesia." *ITSC, 2022.* https://ieeexplore.ieee.org/abstract/document/9922278

OPTIMIZATION UNDER UNCERTAINTY

Exploration of optimization and simulation techniques for in the context of decision-making under uncertainty.

1. Ziyad, Muhammad, Kenrick Tjandra, Mushonnifun Faiz Sugihartanto, Mansur Arief. "An Optimized and Safety-aware Maintenance Framework: A Case Study on Aircraft Engine." *ITSC, 2022.* https://ieeexplore.ieee.org/abstract/document/9922187
2. Oktavian, Muhammad Rizki, Diana Febrita, Mansur Arief. "Cogeneration Power-Desalination in Small Modular Reactors (SMRs) for Load Following in Indonesia." *ICST, 2018.* https://ieeexplore.ieee.org/abstract/document/8528706
3. Pujawan, Nyoman, Mansur Arief, Benny Tjahjono, and Duangpun Kritchanchai. "An Integrated Shipment Planning and Storage Capacity Decision under Uncertainty." *International Journal of Physical Distribution & Logistics Management (IJPDLM), 2015.* https://www.emerald.com/insight/content/doi/10.1108/IJPDLM-08-2014-0198/full/html

SISL
Stanford Intelligent Systems Laboratory

Thanks to collaborators!

- Mykel Kochenderfer, AeroAstro Stanford
- Ding Zhao, MechE CMU
- Henry Lam, IEOR Columbia
- Bo Li, CS UIUC
- Zhiyuan Huang, SoM Tongji
- Huan Zhang, EE UIUC
- Iwan Vanany, ISE ITS
- Jef Caers, MineralX Stanford
- Nur Ahmad Khatim, IF ITS
- Yan Akhra, ISE ITS
- Azmul Asmar, FK UIN SH
- Amaliya Mata'ul, FK UIN SH
- Rully Hendrawan, SCS Pitt (IS ITS)
- Abhibha Gupta, SCS Pitt
- Yasmine Alonso, CS Stanford
- Anthony Corso, AeroAstro Stanford
- IndoSTEELERS members
- SISL members
- CMU Safe AI members

SISL
Stanford Intelligent
Systems Laboratory

Let's stay in touch

Mansur Maturidi Arief

Email: mansur.arief@stanford.edu

Web: https://mansurarief.github.io/

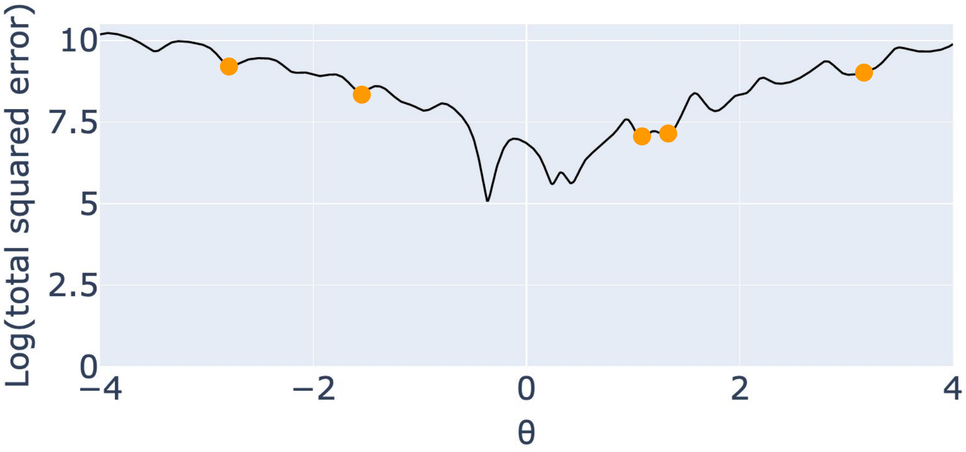
SISL
Stanford Intelligent
Systems Laboratory

Appendix

SISL
Stanford Intelligent Systems Laboratory

Gradient descent and variants

1. Use a handful of **random initializations**:
   - Sample n_o number of θ_0's.
   - For each, perform gradient descent algorithm.
   - Compare the results and pick the best one!

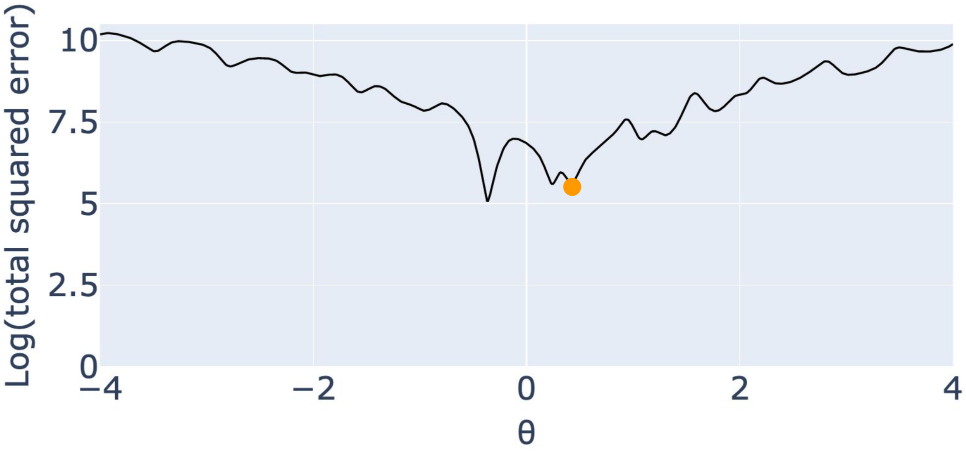


SISL
Stanford Intelligent
Systems Laboratory

Gradient descent and variants

2. Use **batch of samples** in each iteration (stochastic gradient descent)

- In iteration k, sample $n_k \leq n$ data points without replacement
- Re-compute the objective function $J_k(\theta) = \sum_{i=1}^{n_k} (f_\theta(X_i) - Y_i)^2$
- Use $\nabla J_k(\theta_k)$ to update the iterate θ_k



Insights:

a) $\mathbb{E}[\nabla J_k(\theta_k)] \to \nabla J(\theta_k)$

b) The noisy gradient estimate allows us to jump out of local optima at times.

c) For HUGE data points, using batch of samples is also more practical.

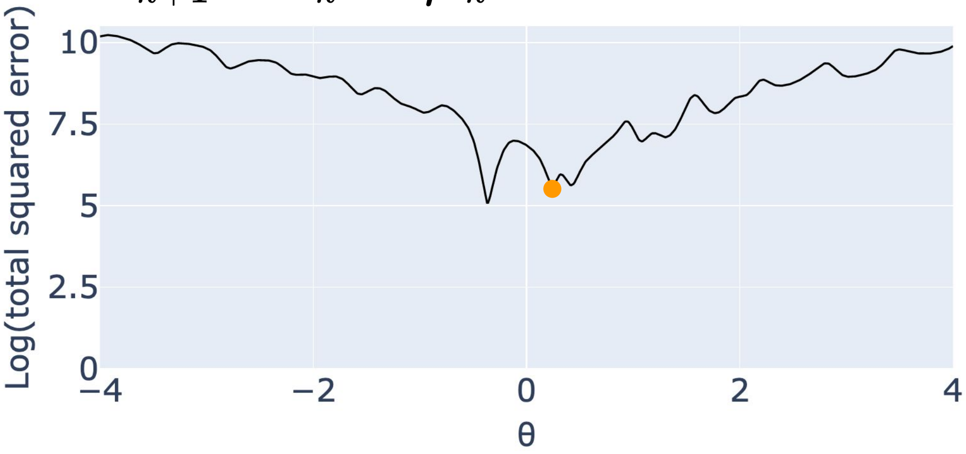
SISL
Stanford Intelligent Systems Laboratory

Gradient descent and variants

3. Use **momentum** as energy signature in each gradient step

- In iteration k, compute momentum v_k with momentum weight β
- Perform gradient step using v_k

$$v_k = \beta v_{k-1} + (1-\beta)\nabla J(\theta_k)$$
$$\theta_{k+1} = \theta_k - \eta v_k$$

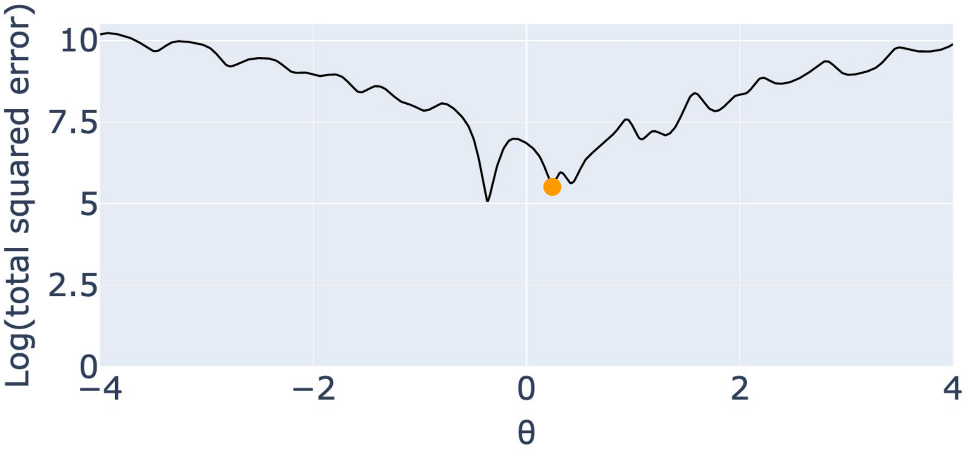


Insights: Momentum adds inertia to gradient descent:
- If we have been making long steps, we tend to make another long step.
- Conversely, if we have been making shorter steps, we more likely take another short step.

SISL
Stanford Intelligent Systems Laboratory

Gradient descent and variants

4. Use **scheduled learning rate**
   - Start with large learning rate, then gradually reduce it
     - Constant discounting $\eta_k = \eta_{k-1}\gamma$
     - Scheduled learning rate $\eta_k = \eta(k)$

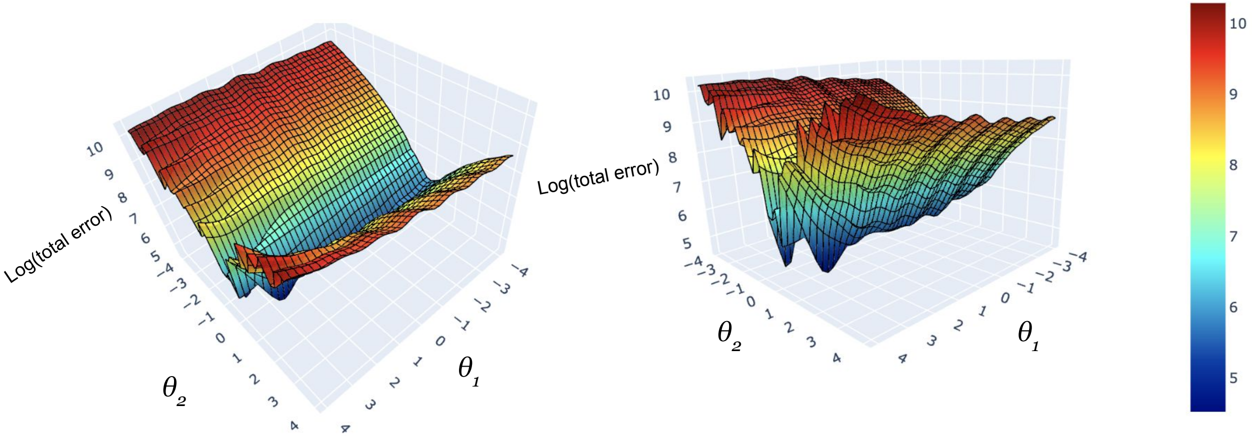


Insights: Scheduling learning rate is an adaptation of simulated annealing, where larger learning rate is equivalent to hotter temperature at earlier steps.

SISL
Stanford Intelligent Systems Laboratory

Gradient descent and variants

5. **Increase the dimensionality** of the space (overparameterization)
   - Use larger model if possible (might be counterintuitive at first)
   - Consider $f_\theta(x) = \theta_1 x^2 + \theta_1 cos(\theta_2 x) + \theta_1\theta_2 + \exp(\theta_2 \sin^2(\theta_1 x)),$
     $\Theta = \{(\theta_1, \theta_2) : \theta_1 \in [-4, 4], \theta_2 \in [-4, 4]\}$



Insights: For larger dimensional problems, gradient descent more likely
a) to find a descent direction
b) to find a better-valued local optima