AI Toward Autonomous Testing -

To What Extent Can Machine Replace An Analyst?

ITC Asia, Keynote 2

Li-C. Wang
University of CA, Santa Barbara

Disclaimer and Acknowledgement

- This talk is has a limited view, focusing on particular applications in particular companies. The view might or might not be applicable to other contexts

- Thanks my students
  - Jay Shan
  - Matt Nero
  - Dr. Kuo-Kai Hsieh (Pinterest)
  - Dr. Nik Sumikawa (with NXP since 2015)

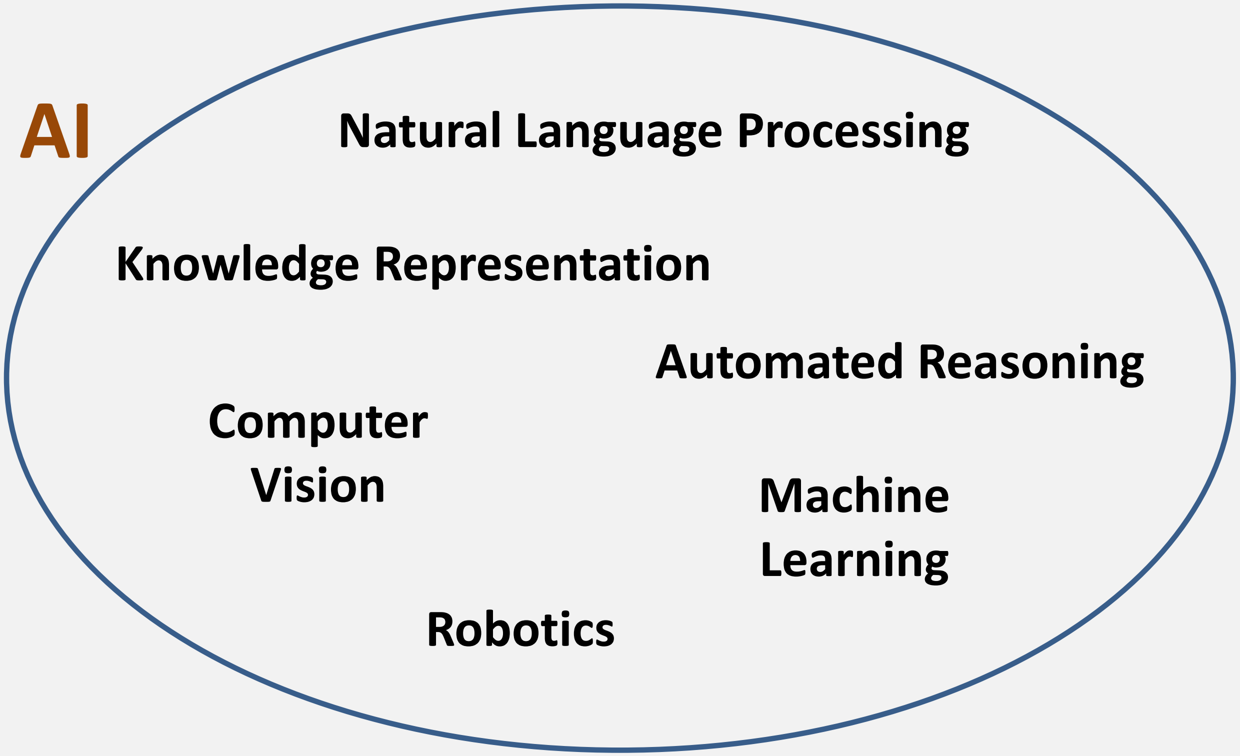
3 Questions with The Title …

- What kind of "AI" are we talking about?

- Why use the word "Autonomous"?

- To what extent machine can replace human?

What Kind of "AI"?



AI

Natural Language Processing

Knowledge Representation

Automated Reasoning

Computer Vision

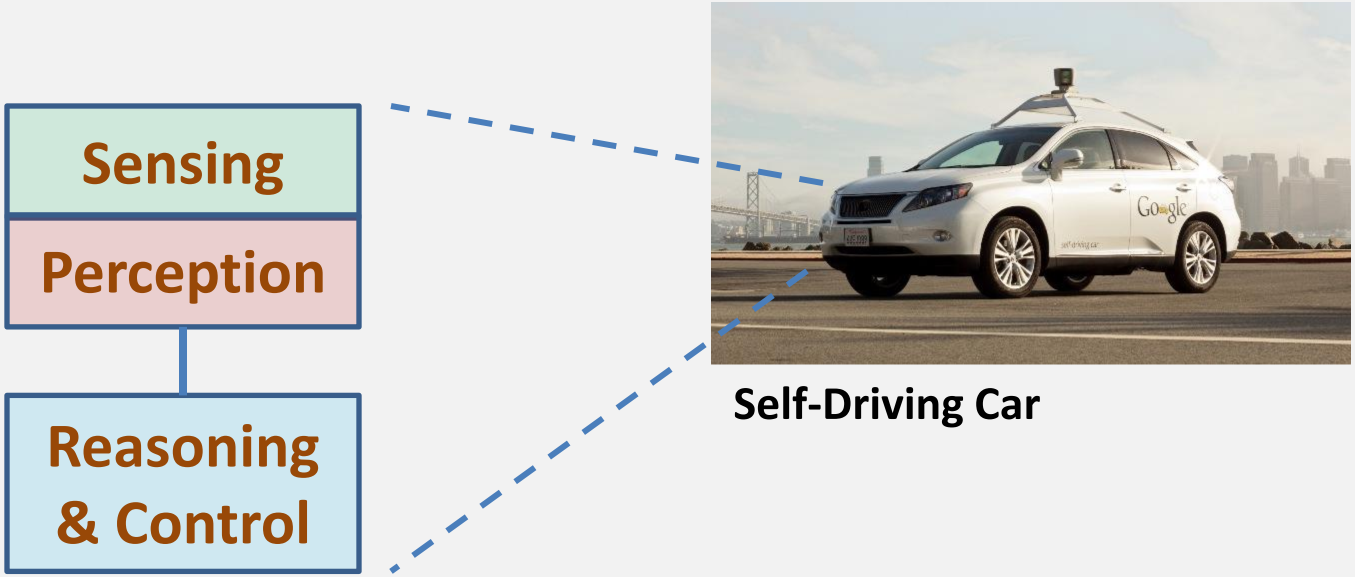
Machine Learning

Robotics

(Acting humanly – The Turing Test Approach)

- 1950 "Computing Machinery and Intelligence" – The Turing Test
- Other AI: Thinking humanly, Thinking rationally, Acting rationally

Why “Autonomous”?


Sensing

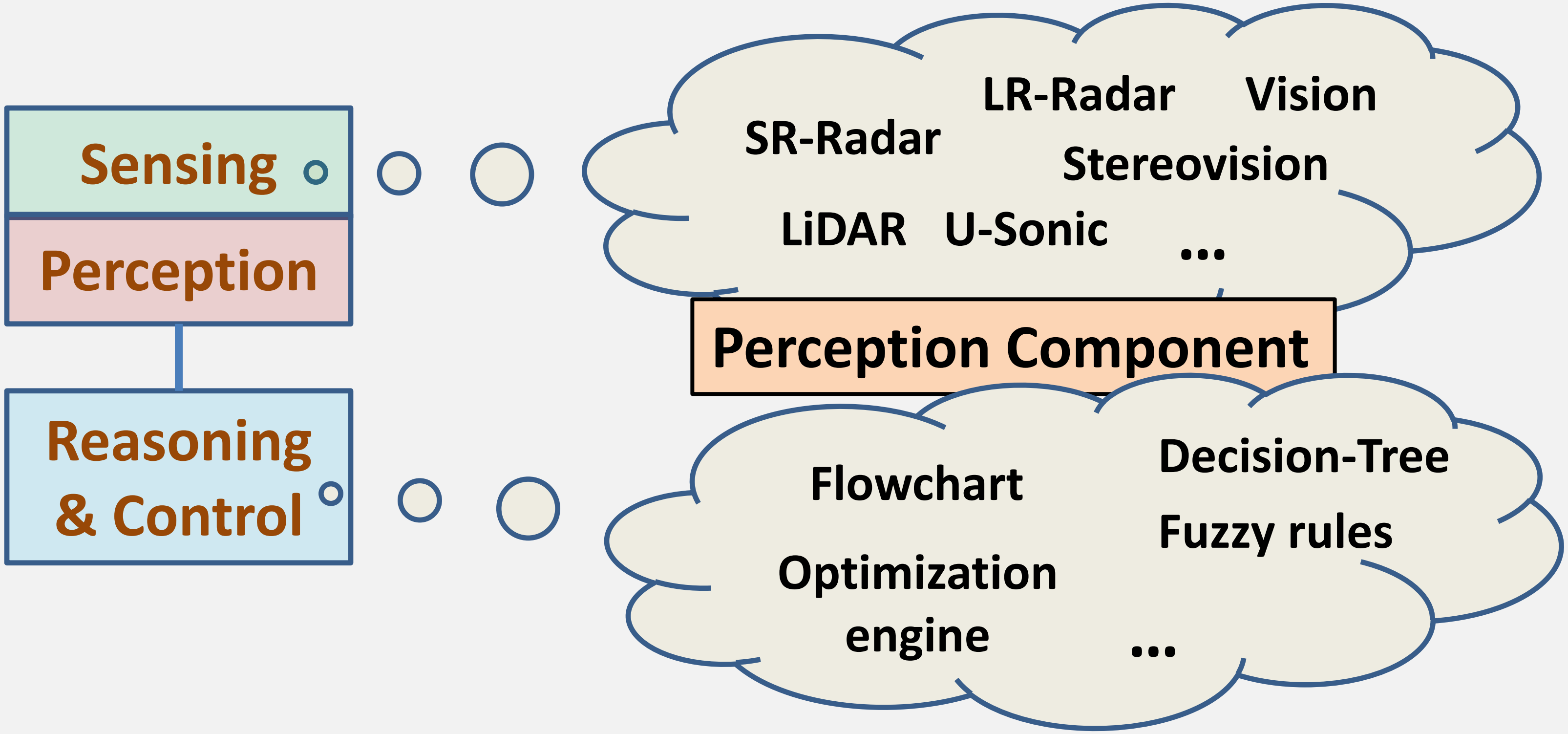
Perception

Reasoning & Control

Self-Driving Car

- **Sensing**: SR/LR Radars, LiDAR, Vision, Stereovision, U-Sonic, etc.
- **Perception**: Neuro-processors (lane detection, object recognition)
- **Reasoning & Control**: Free space calculation, path planning, speed/brake/rotation controls

What The 3 Components Do



- **Sensing**: Collect all relevant data
- **Perception**: Recognize what data mean
- **Reason & Control**: What to do next

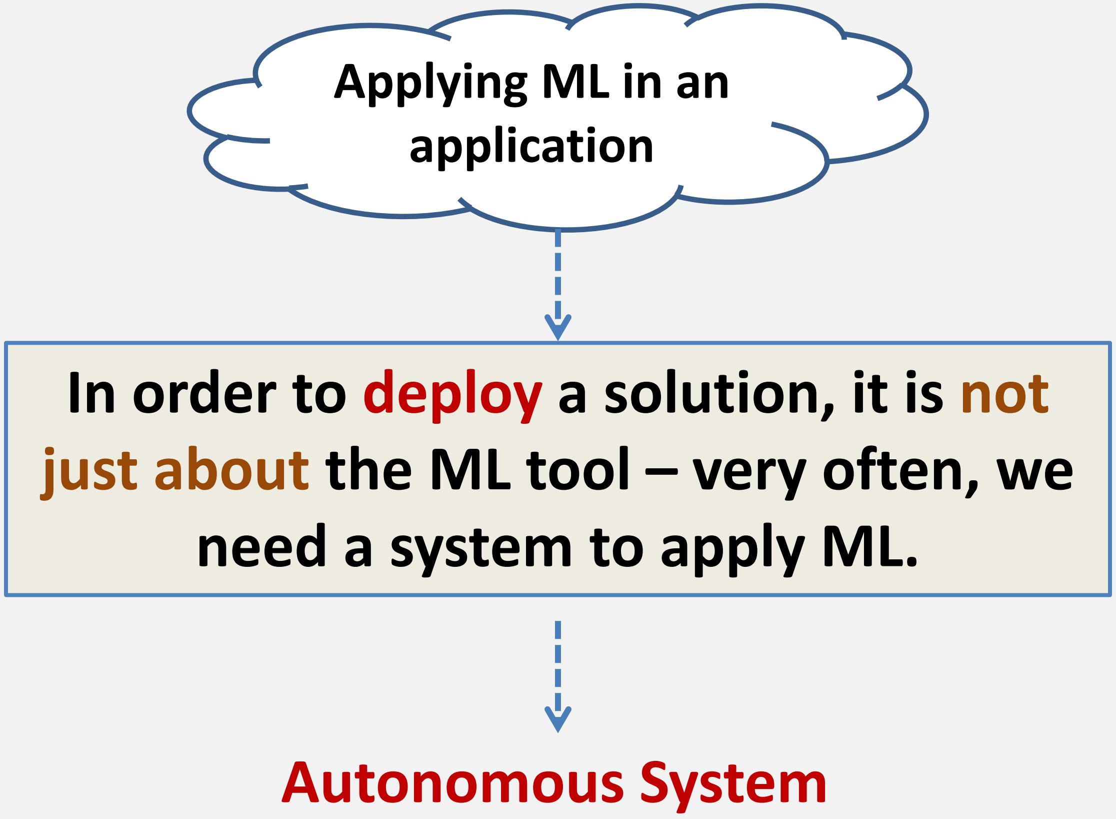
Ok, that’s interest …

But why we talk about it?

Most of us don’t build the autonomous system in a self-driving car …

That's true, but most of us are interested in applying ML in our respective applications …

A Key Message: System View for Applying ML



Applying ML in an application

In order to **deploy** a solution, it is **not just about** the ML tool – very often, we need a system to apply ML.

Autonomous System

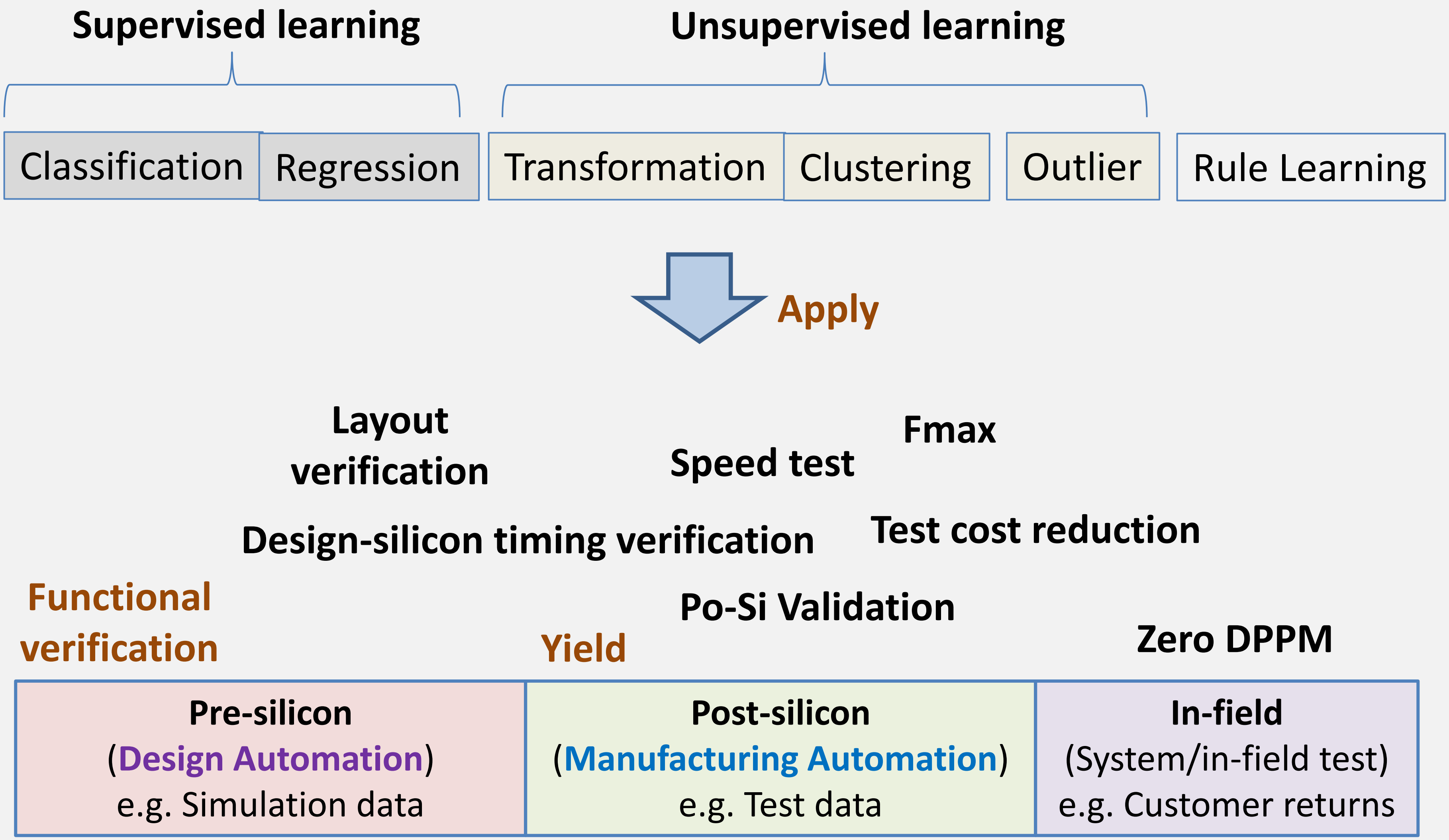
- ➢ **This is especially the case when the ML solution is deployed in design/test processes**

Applying ML → Autonomous System

Next …

Let’s talk about this journey …

Applying ML in Design/Test (2003-2013)



Supervised learning

Unsupervised learning

Classification | Regression | Transformation | Clustering | Outlier | Rule Learning

Apply

Layout verification

Speed test

Fmax

Design-silicon timing verification

Test cost reduction

Functional verification

Po-Si Validation

Yield

Zero DPPM

Pre-silicon
(**Design Automation**)
e.g. Simulation data

Post-silicon
(**Manufacturing Automation**)
e.g. Test data

In-field
(System/in-field test)
e.g. Customer returns

Applying ML in Design/Test (2003-2013)

Supervised learning

Unsupervised learning

Classificatio

Rule Learning

IEEE Trans. On CAD Paper (Oct 2016):

"Experience of Data Analytics in EDA and Test – Principles, Promises, and Challenges"

Functiona verificatio

DPPM

Pre-silicon	**Post-silicon**	**In-field**
(Design Automation)	(Manufacturing Automation)	(System/in-field test)
e.g. Simulation data	e.g. Test data	e.g. Customer returns

Applying ML in Design/Test (2003-2013)

Supervised learning

Unsupervised learning

Classificatio

ule Learning

This journey went through multiple stages …

Functiona verification

DPPM

Pre-silicon	**Post-silicon**	**In-field**
(Design Automation) e.g. Simulation data	(Manufacturing Automation) e.g. Test data	(System/in-field test) e.g. Customer returns

Applying ML in Design/Test (2003-2013)

Supervised learning

Unsupervised learning

Classificatio

ule Learning

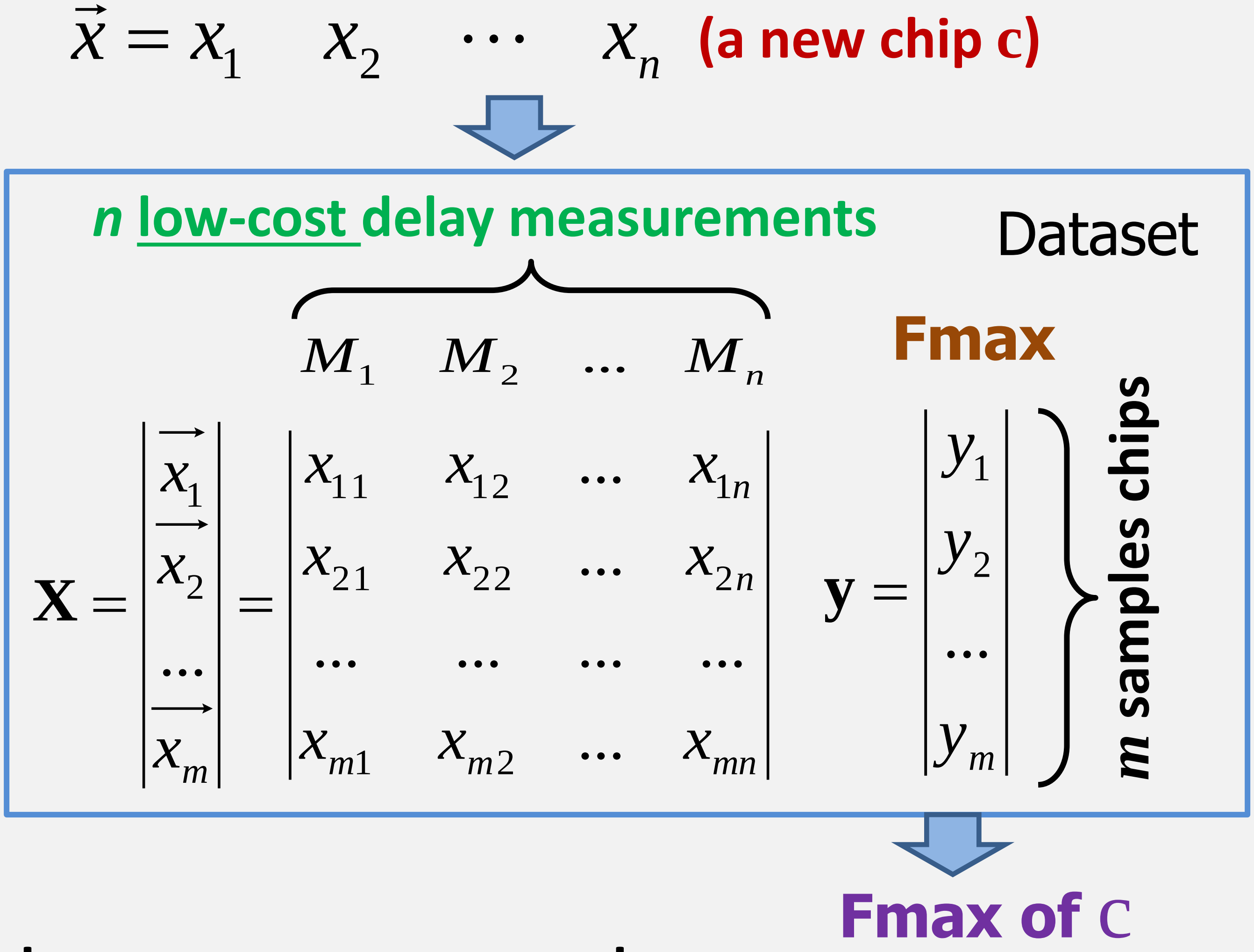
At first, it was "Algorithmic Focus" – What is the best ML algorithm to use?

Functiona verificatio

DPPM

Pre-silicon	**Post-silicon**	**In-field**
(Design Automation)	(Manufacturing Automation)	(System/in-field test)
e.g. Simulation data	e.g. Test data	e.g. Customer returns

An Application Example – Fmax Prediction



$\vec{x} = x_1 \quad x_2 \quad \cdots \quad x_n$ **(a new chip c)**

n **low-cost delay measurements**

Dataset

Fmax

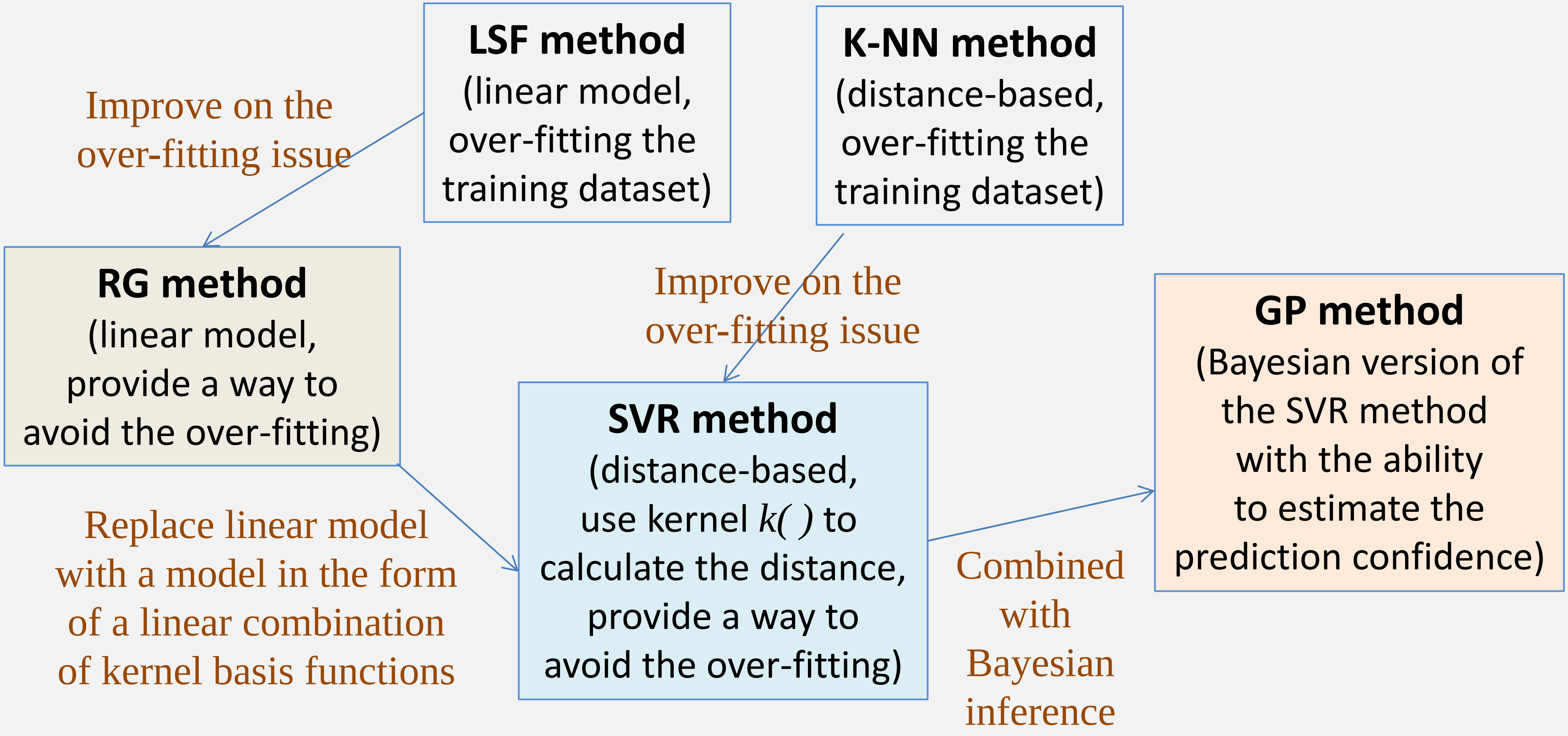
$$\mathbf{X} = \left|\begin{array}{c} \vec{x_1} \\ \vec{x_2} \\ \cdots \\ \vec{x_m} \end{array}\right| = \begin{array}{cccc} M_1 & M_2 & \cdots & M_n \\ x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{array} \qquad \mathbf{y} = \left|\begin{array}{c} y_1 \\ y_2 \\ \cdots \\ y_m \end{array}\right|$$

m **samples chips**

Fmax of c

- **Delay measurements can be**
  - **FF based, pattern based, path based, or RO based**

Example Algorithms For Regression



LSF method
(linear model, over-fitting the training dataset)

K-NN method
(distance-based, over-fitting the training dataset)

Improve on the over-fitting issue

RG method
(linear model, provide a way to avoid the over-fitting)

Improve on the over-fitting issue

GP method
(Bayesian version of the SVR method with the ability to estimate the prediction confidence)

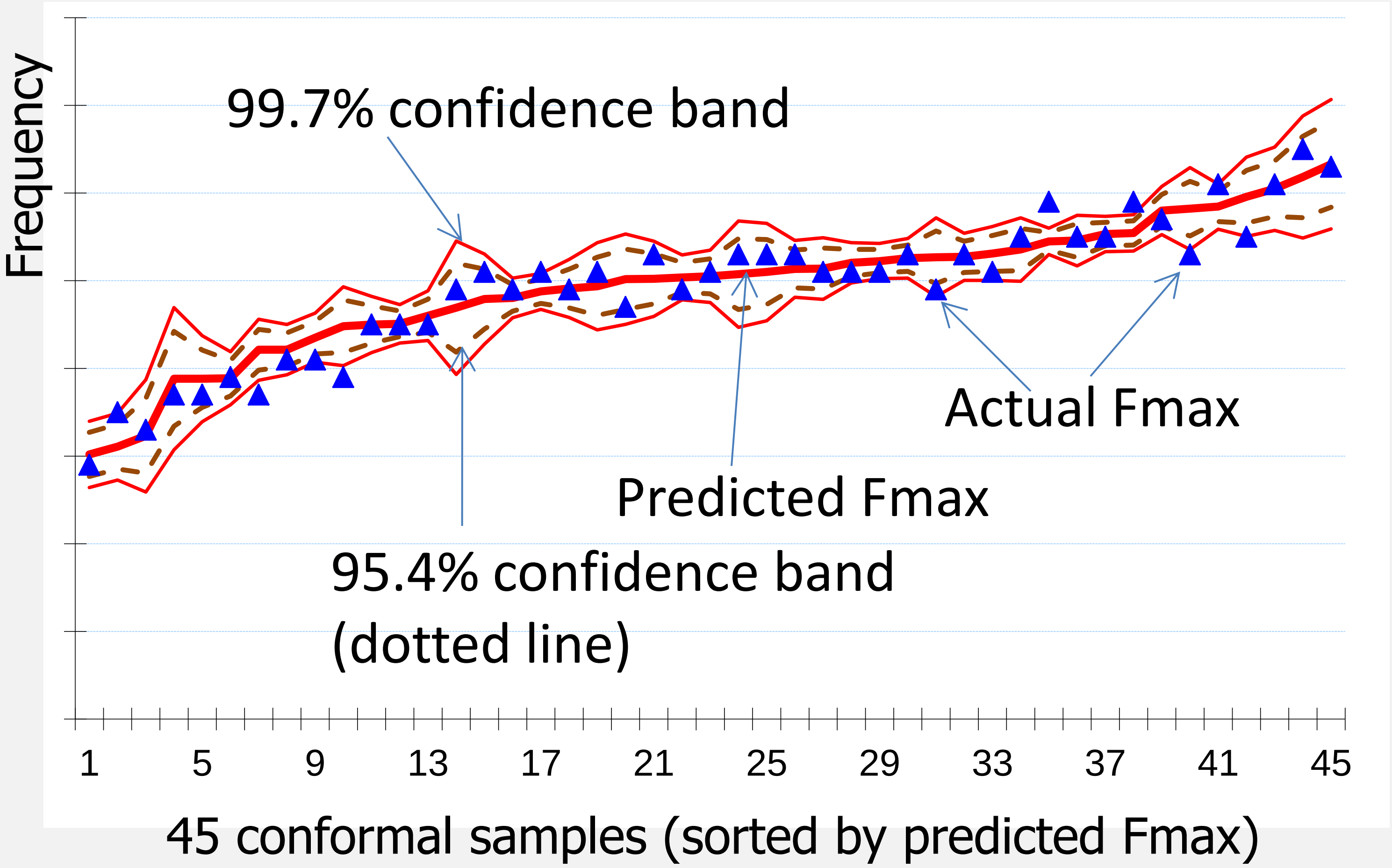
SVR method
(distance-based, use kernel $k(\ )$ to calculate the distance, provide a way to avoid the over-fitting)

Replace linear model with a model in the form of a linear combination of kernel basis functions

Combined with Bayesian inference

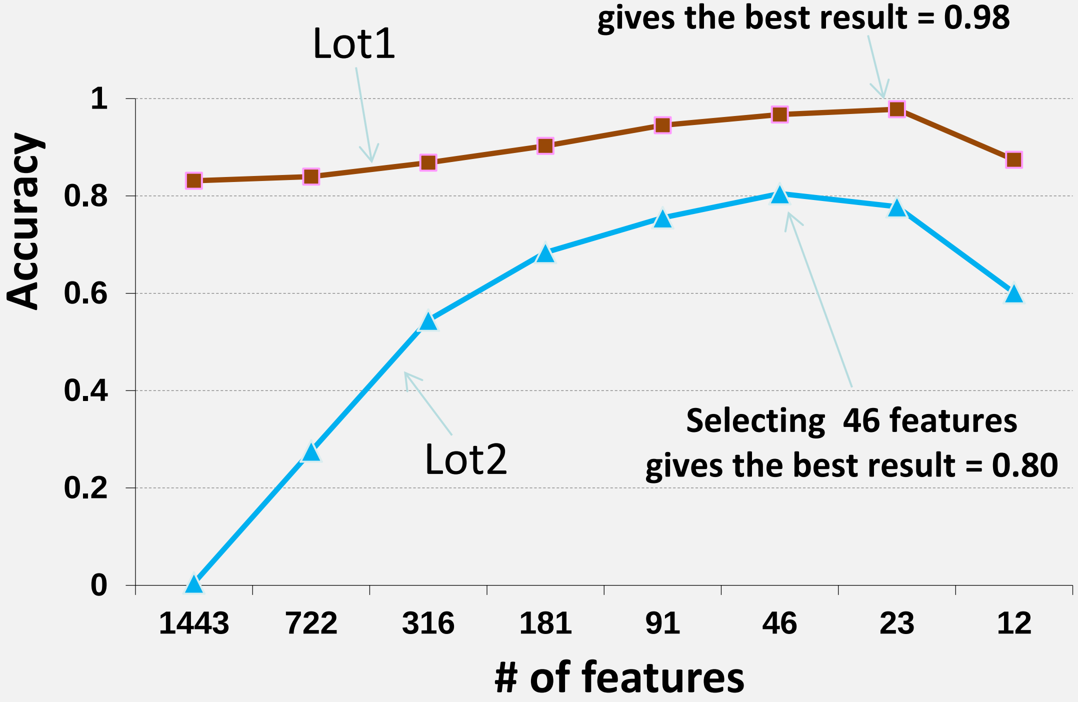
- **See Janine Chen et al. (ITC 2009)**
  - **"Data Learning Techniques and Methodology for Fmax Prediction"**

GB Was The Best! (Conformal Check)



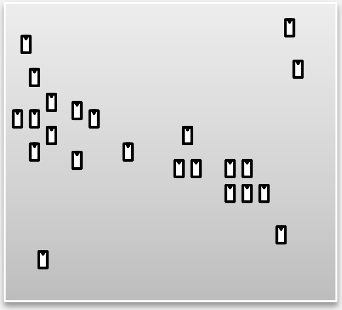
- **See Janine Chen et al. (ITC 2009)**
  - **"Data Learning Techniques and Methodology for Fmax Prediction"**

A Barrier for Deployment

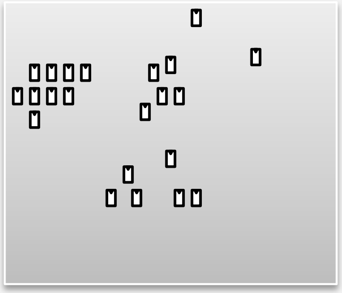


Selecting 23 features gives the best result = 0.98

Selecting 46 features gives the best result = 0.80



Lot1



Lot2

- Can’t deploy a model without having a consistent set of features across all lots

Applying ML in Design/Test (2003-2013)

Supervised learning

Unsupervised learning

Classificatio

Rule Learning

In the 2nd stage, it was "Methodology Centric" – What is the most effective methodology to enable deployment of a ML-based solution/model?

Functiona
verificatio

DPPM

Pre-silicon (Design Automation) e.g. Simulation data	**Post-silicon** (Manufacturing Automation) e.g. Test data	**In-field** (System/in-field test) e.g. Customer returns

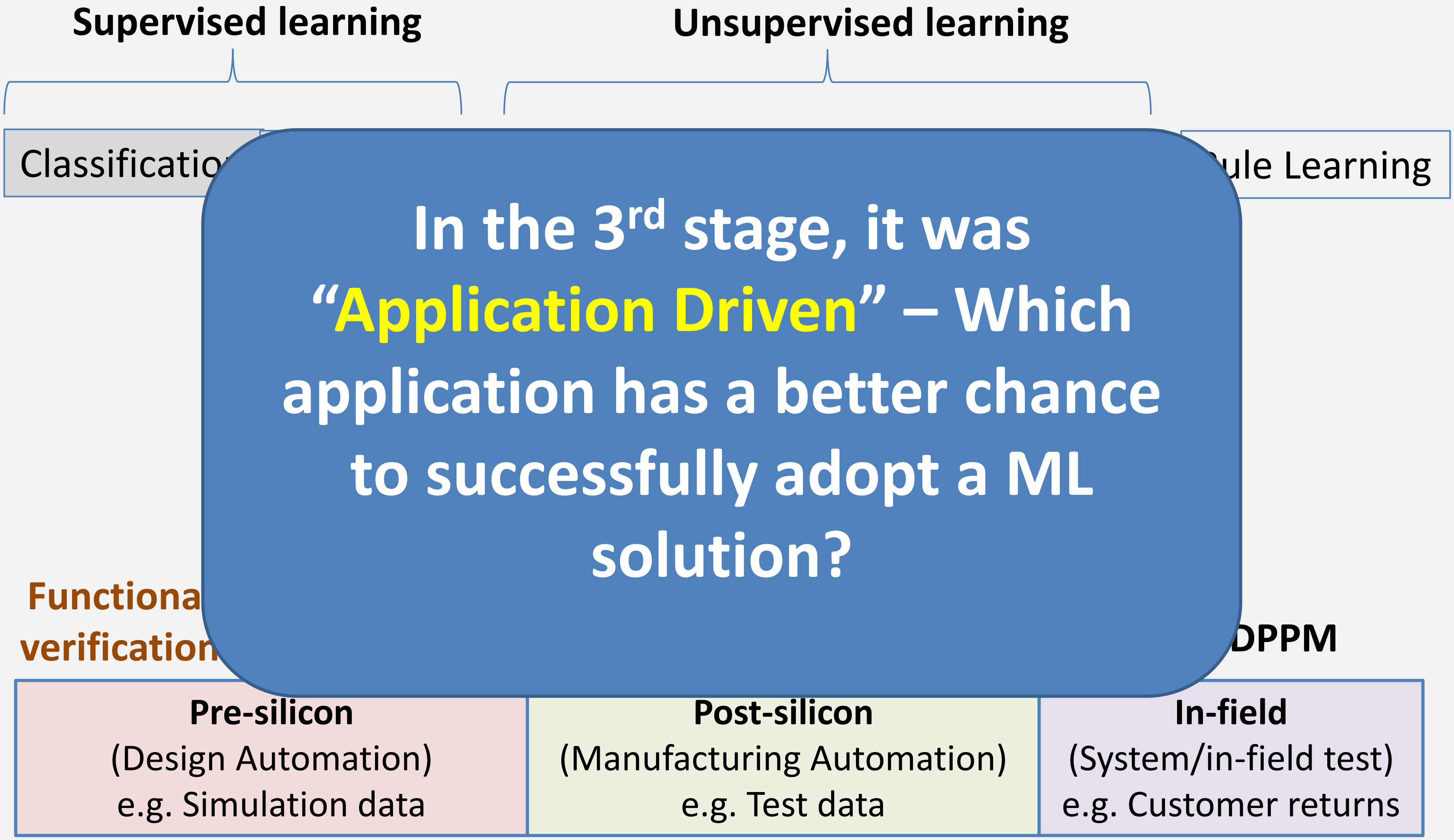
The Need for Domain Expert

- A domain expert won’t accept a solution if he/she can’t see the value, or don’t understand it
  - Interpretable and actionable model
  - Added value to their existing solutions already in place

- Let the methodology start with an expert, by
  - Asking for a set of “reasonable” features
  - Collecting sufficient data for learning feature importance

- But …
  - If the engineer knows what features are relevant, why even apply so-called “Machine Learning”?
  - If they don’t know, how much data is needed?
  - If collecting the data is hard, will it ever get done?
  - If it is too costly, what’s the added value?

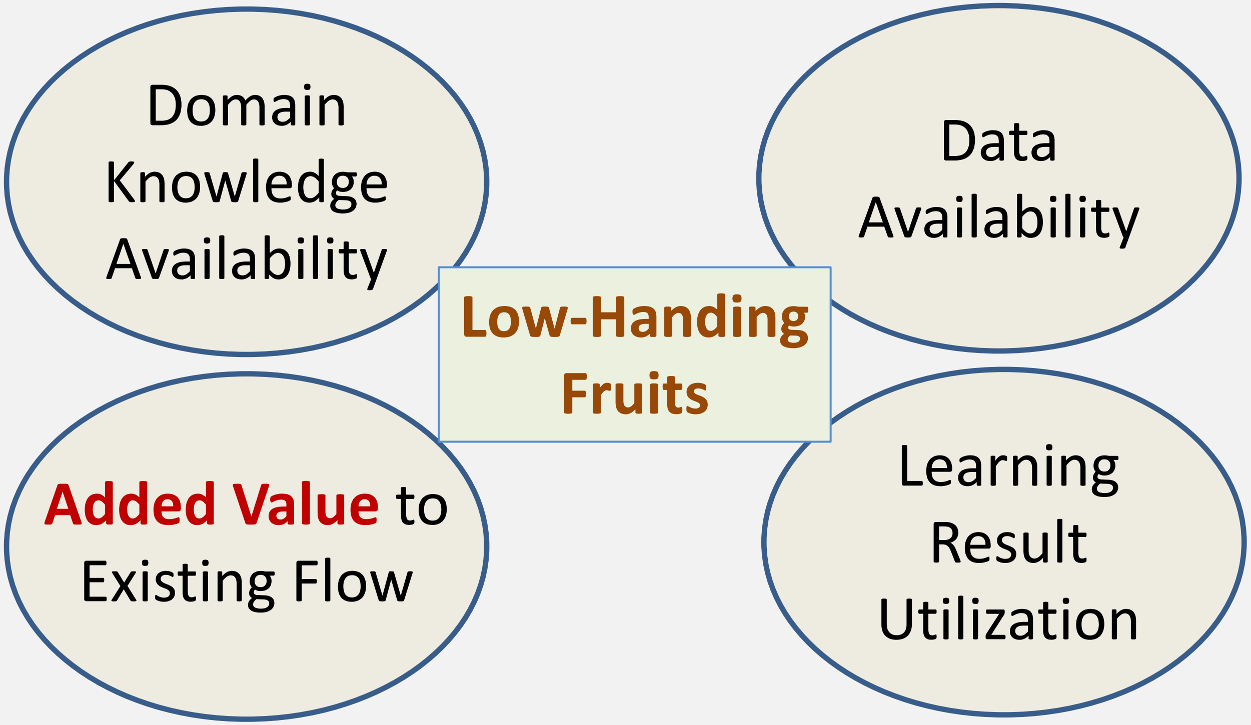
Applying ML in Design/Test (2003-2013)



In the 3rd stage, it was "**Application Driven**" – Which application has a better chance to successfully adopt a ML solution?

Pre-silicon	Post-silicon	In-field
(Design Automation)	(Manufacturing Automation)	(System/in-field test)
e.g. Simulation data	e.g. Test data	e.g. Customer returns

My Old Slide: Four Key Considerations


Domain Knowledge Availability
Data Availability
Low-Handing Fruits
Added Value to Existing Flow
Learning Result Utilization

- In this picture, I did not mention a “learning algorithm” because it was not as deciding a factor than these four for realizing a practical methodology for an application

Applying ML in Design/Test (Since 2013)

Supervised learning

Unsupervised learning

Classification

le Learning

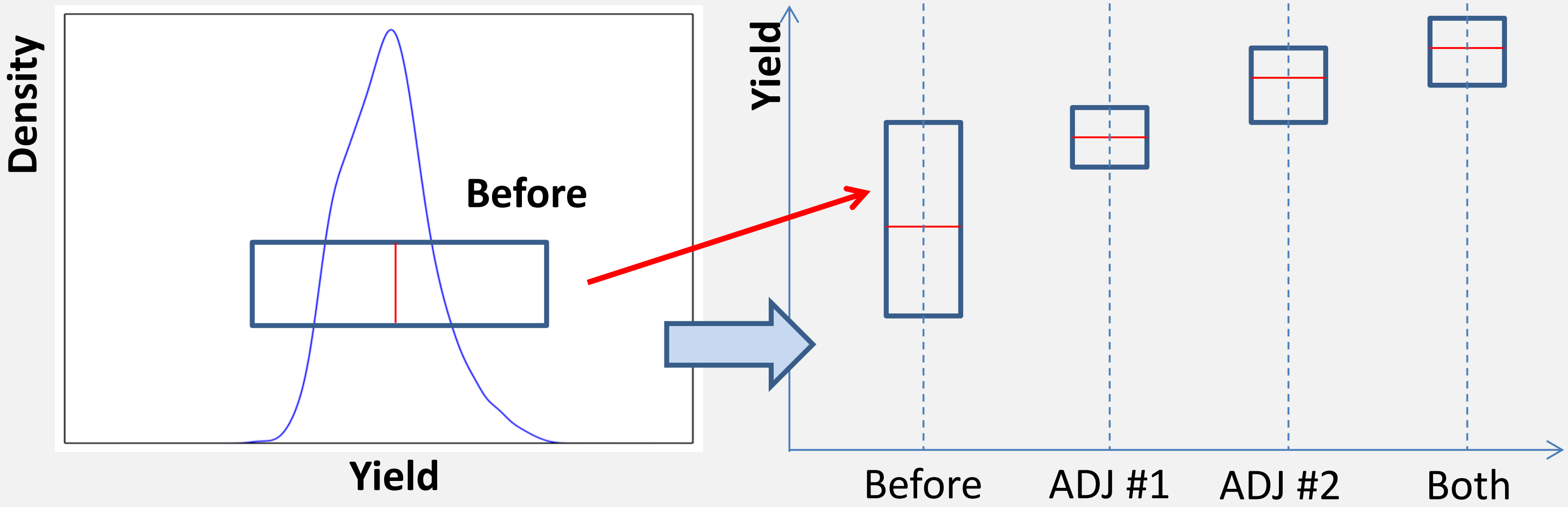
In the 4th stage, it was all about the "Deployment" – What does it take to deploy a ML solution and actually see an "Added Value?"

Functional verification

PPM

Pre-silicon	Post-silicon	In-field
(Design Automation) e.g. Simulation data	(Manufacturing Automation) e.g. Test data	(System/in-field test) e.g. Customer returns

Added Value – Yield Improvement (2013)



- Yield fluctuated for the SoC product line, and the product/design teams could not solve the problem for months after several design and test revisions, and several process tuning recipes
- By apply ML tools on silicon data, we found 5 process parameters to be tuned
- Foundry accepted them and implemented as two adjustments
- Significant yield improvement observed in production
- ITC 2014 paper documents the story

An Important Question Next …

- In the yield example, we and the product team had access to the same set of ML tools

- So, why we succeeded and they did not?

- Because we had the knowledge enabling us to conduct a more effective analytic process to apply the ML tools

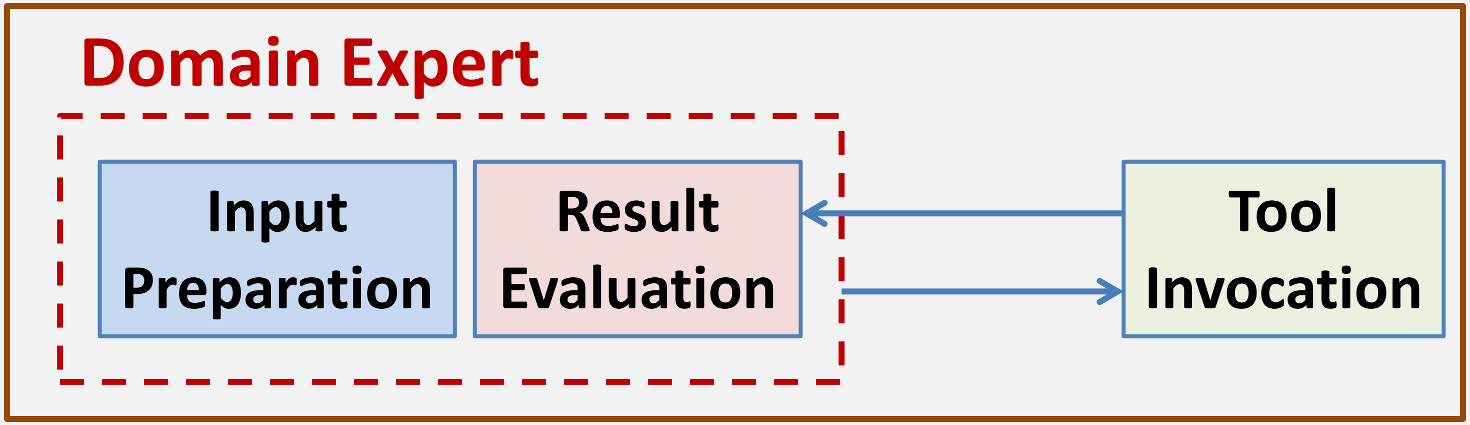
- It was that piece of knowledge made the difference, not the tools in use

An Important Deployment Question

- For deploy a solution, I can't just package the ML tools and give it to the product team

- I needed to package my "knowledge" – How am I going to do that?

We Need A System View To Apply ML

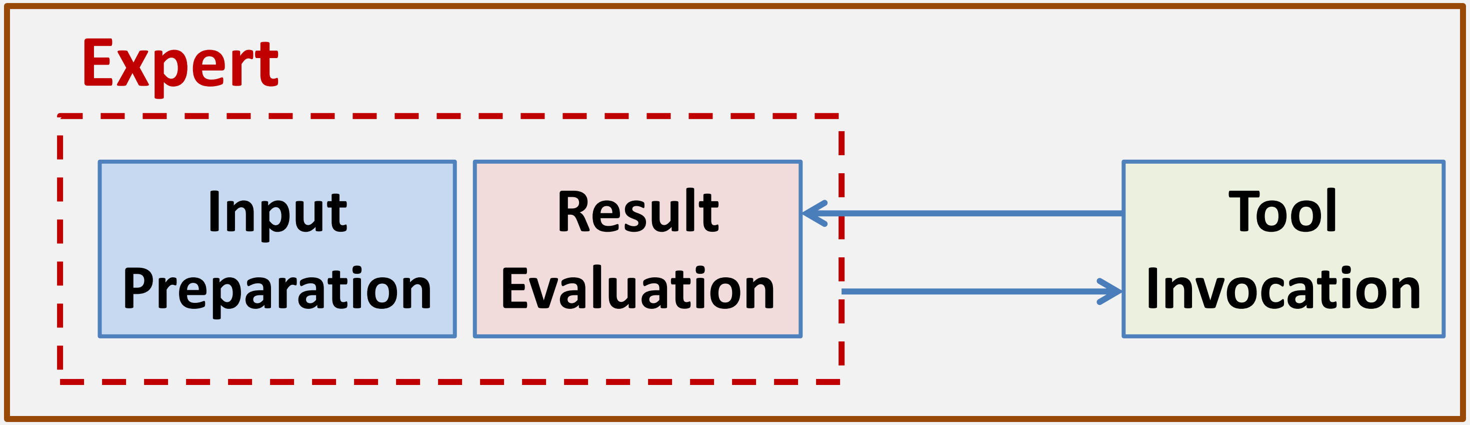


Need Automation of all three components

- **So in short, why the system view?**

- **Because we need domain knowledge**

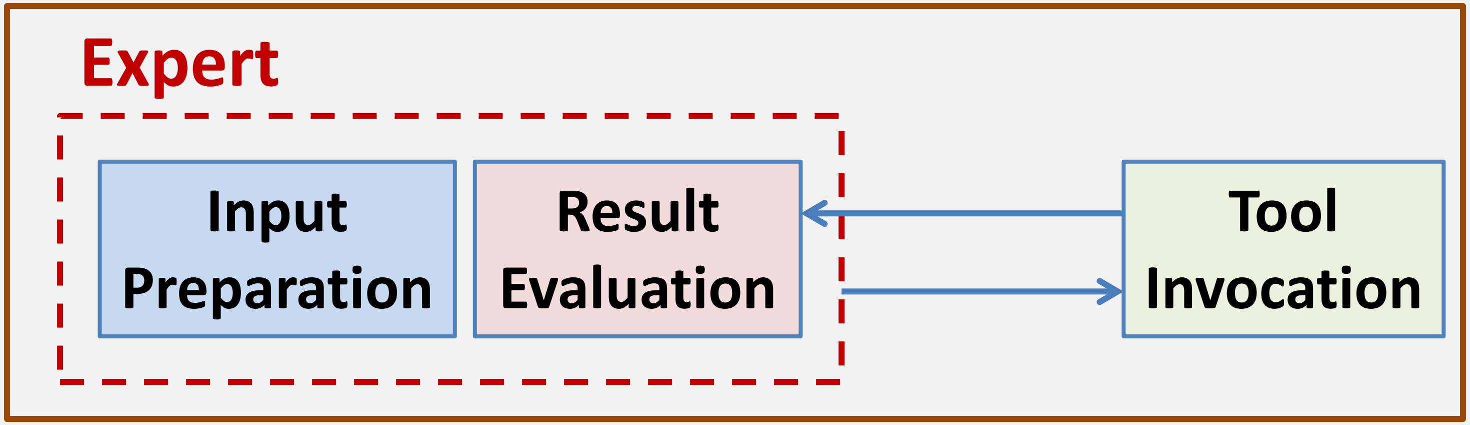
We Need A System View To Apply ML



- Why do we need domain knowledge?

- Mainly, because we have limited data

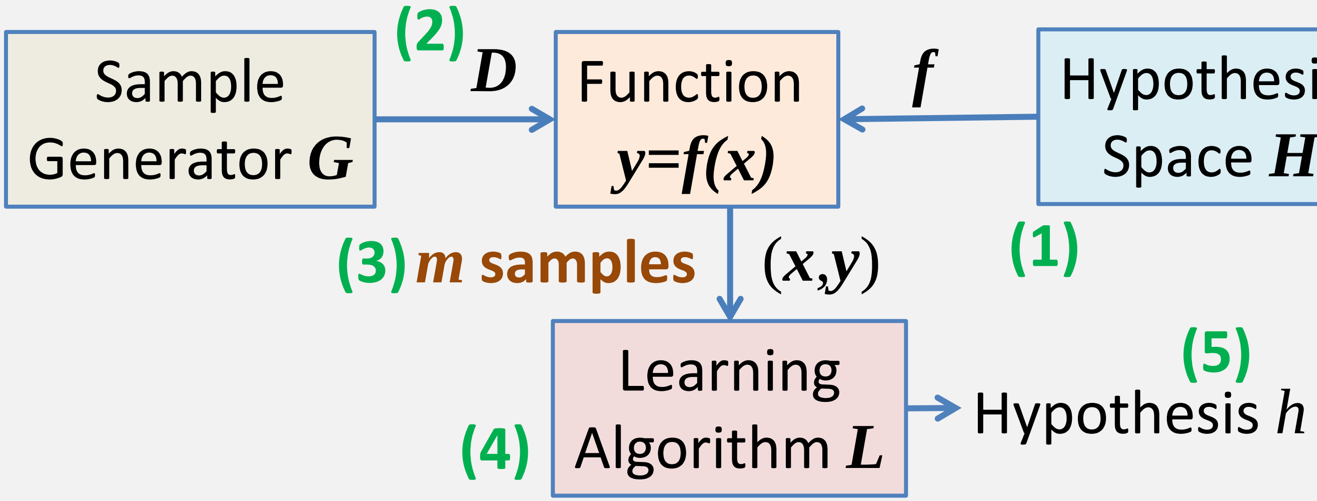
We Need A System View To Apply ML



- What is so special about applying ML in view of “limited data?”
- *“Learning from Limited Data in VLSI CAD”* – an upcoming book chapter - preview at our lab web site: https://iea.ece.ucsb.edu/
- Because there are theoretical assumptions made to achieve ML, and with limited data those assumptions would be hard to meet in practice → we need domain knowledge to compensate ML

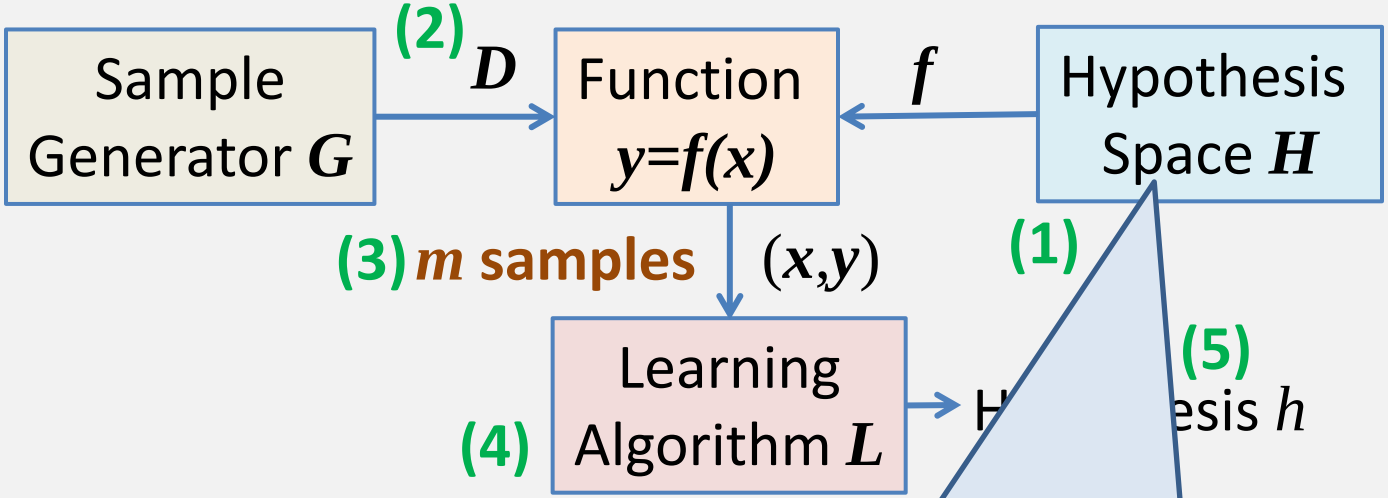
What Theoretical Assumptions for Machine Learning?

Five Assumptions for Supervised Learning



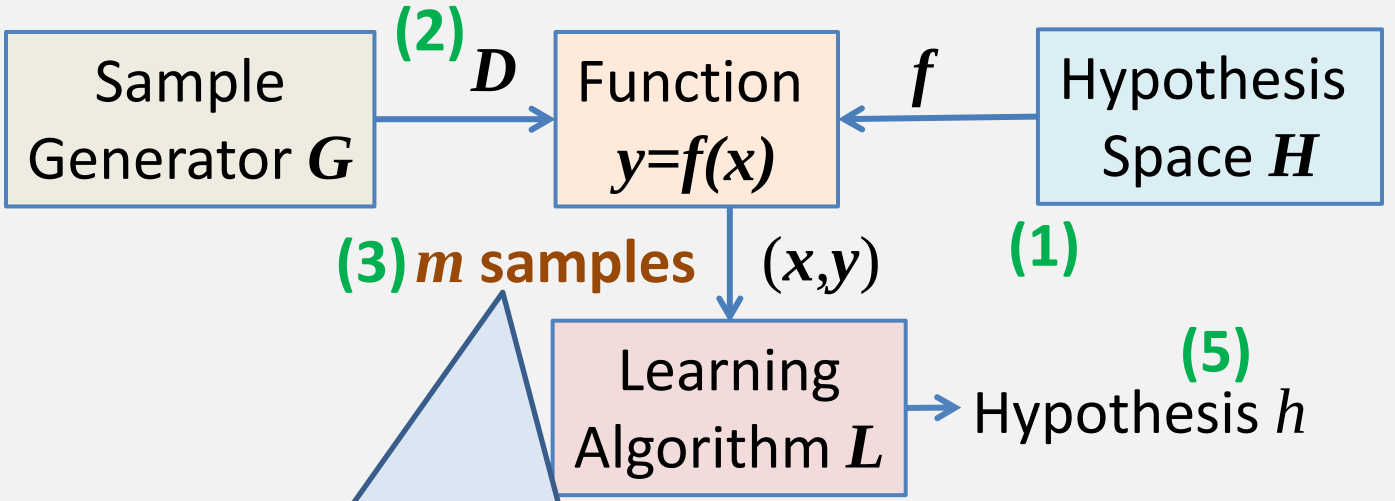
- **A restriction on *H* (otherwise, NFL)**
- **An assumption on *D* (i.e. not time-varied, e.g silicon data)**
- **Assuming size *m* is in order O(poly(*n*)), *n*: # of features**
- **Making sure a practical algorithm *L* exists**
- **Assuming a way to measure error, e.g. *Err(f(x), h(x))***

In Practice, Issue #1



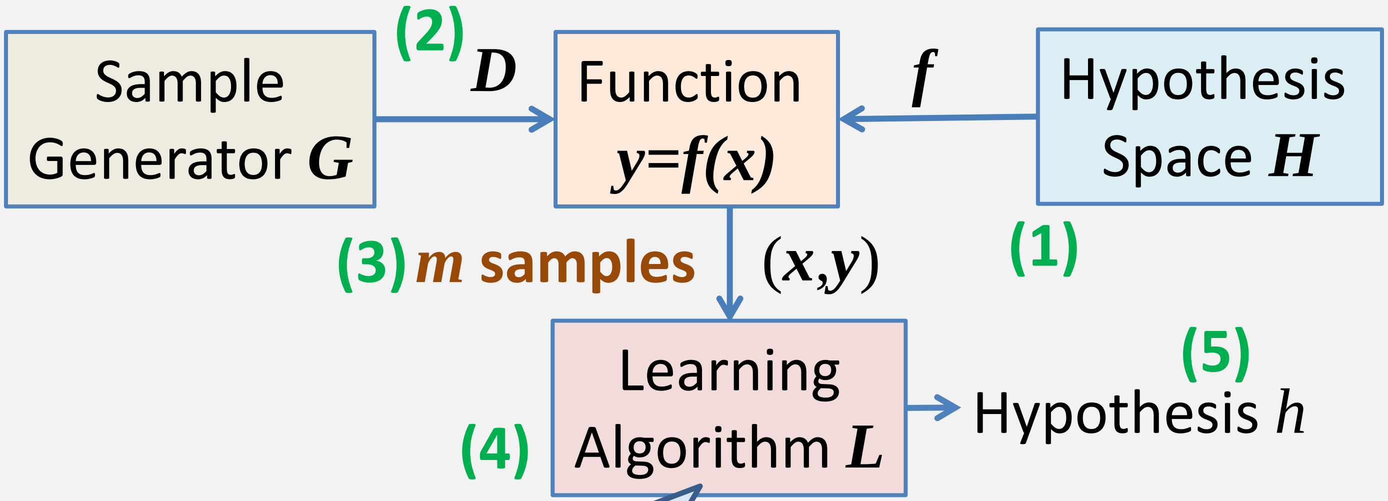
Because we don't know how complex *H* should be, we assume the most complex *H* we can afford in training

In Practice, Issue #2



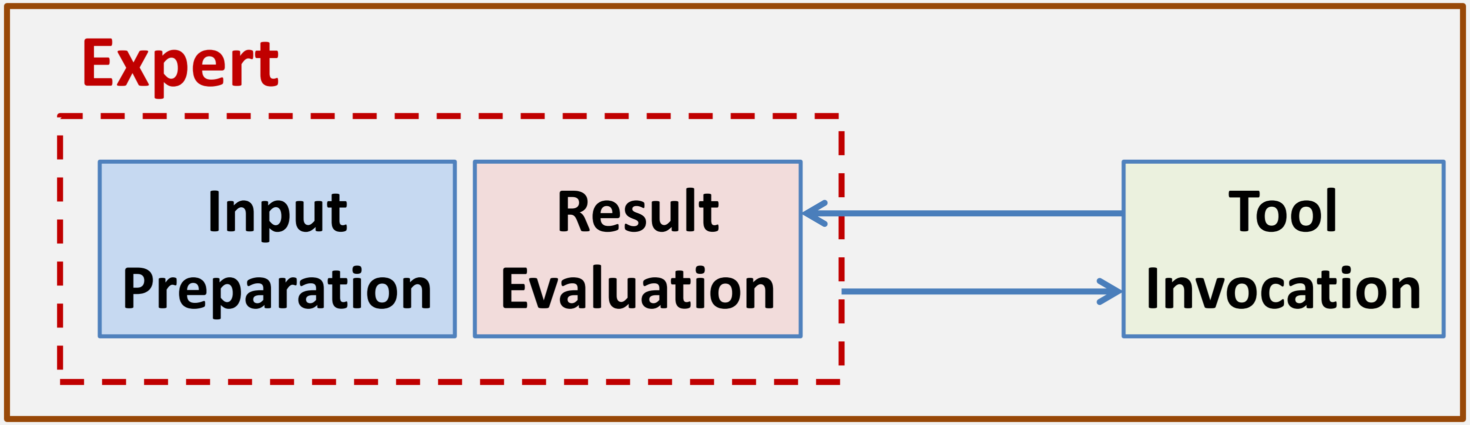
For a complex *H* we need a large amount of data, but we usually don’t know if we have enough in advance

In Practice, Issue #3



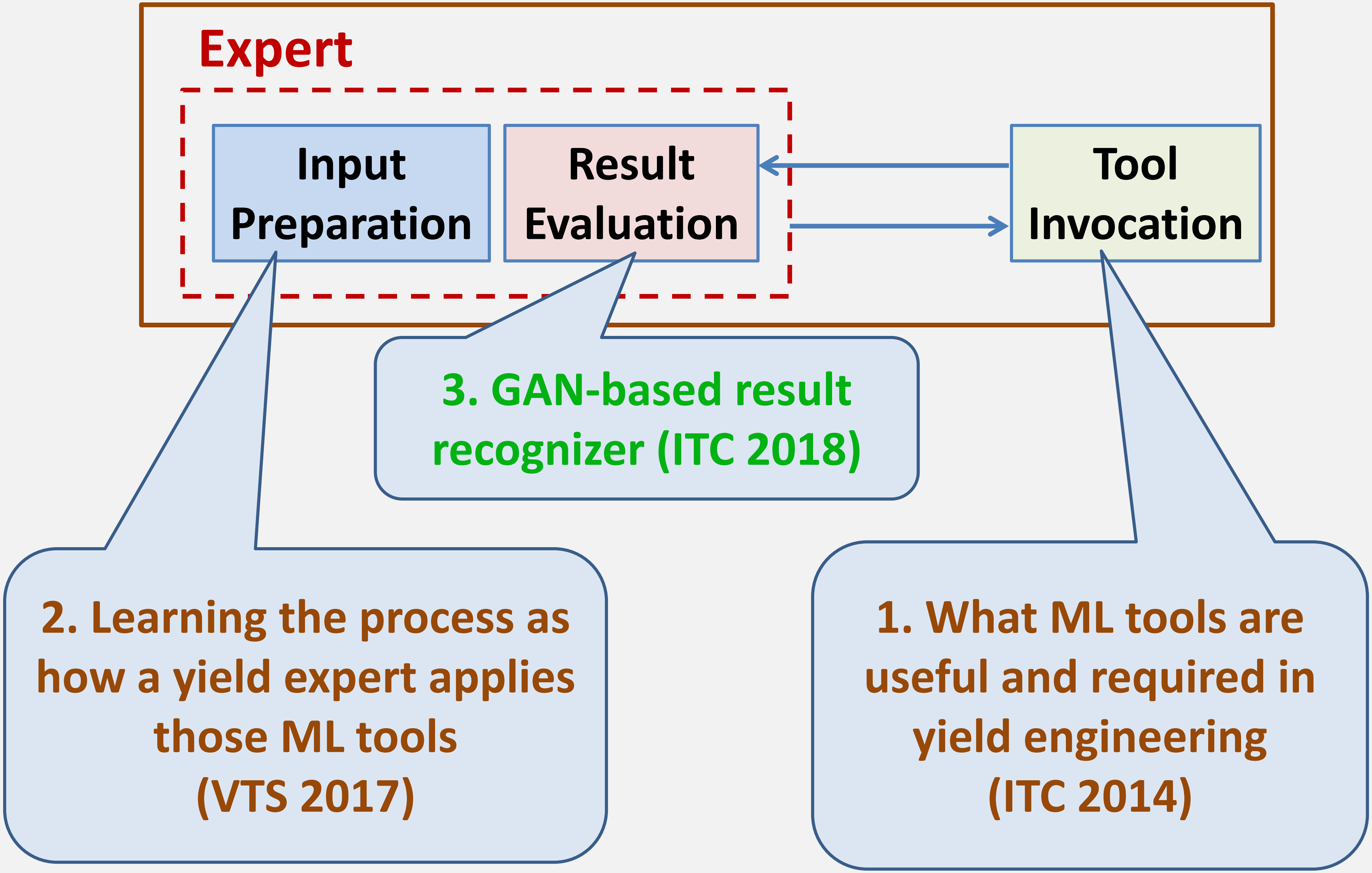
Because non-convex optimization is hard, some **heuristic is used**, and the solution is often a local minimum

In Summary, Four Barriers To Consider …



- A result after considering those 4 barriers
  - Data barrier
  - Theoretical barrier
  - Computational barrier
  - Deployment barrier (over an existing solution)
- The system is largely domain-knowledge-driven

The Yield Context



Expert

Input Preparation

Result Evaluation

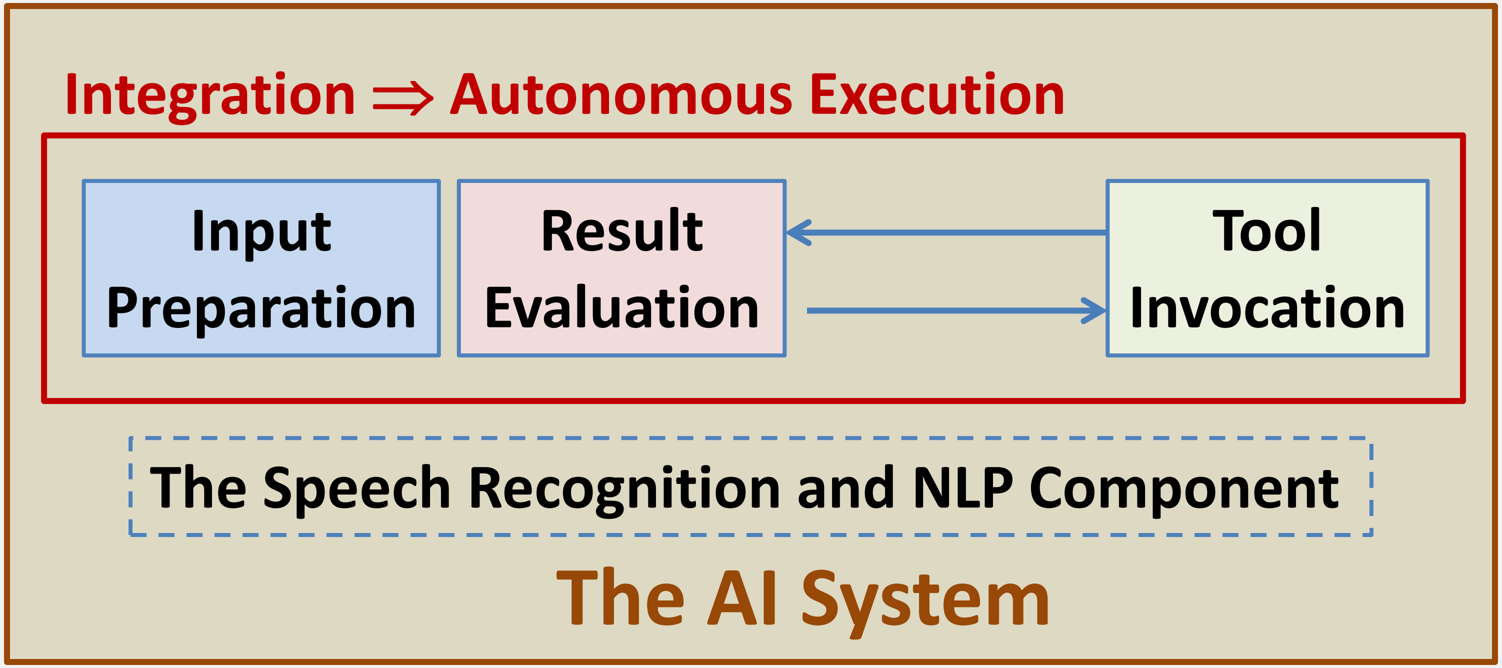
Tool Invocation

3. GAN-based result recognizer (ITC 2018)

2. Learning the process as how a yield expert applies those ML tools (VTS 2017)

1. What ML tools are useful and required in yield engineering (ITC 2014)

The AI System


Integration ⇒ Autonomous Execution
Input Preparation
Result Evaluation
Tool Invocation
The Speech Recognition and NLP Component
The AI System

- The core of this AI system view is the autonomous execution of the workflow

So we launched the IEA project (Intelligence Engineering Assistant)

The IEA research lab: https://iea.ece.ucsb.edu/

Now, let’s have a glance of what IEA looks like …

<IEA Demo>

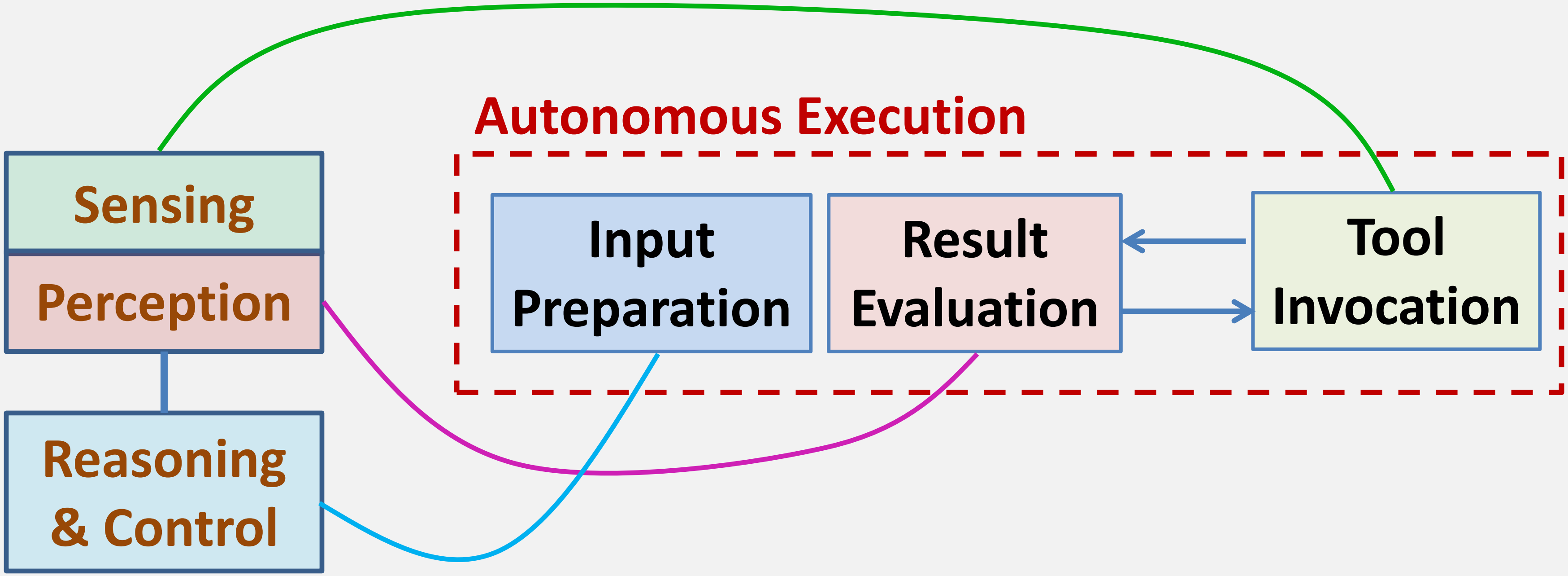
Future Plan of IEA

- Tutorial and courses
  - IEA tutorial (ITC 2018 on Oct 30)
  - IEA courses (Fall and Winter quarters)

- Two IEA systems in progress:
  - IEA for production yield engineer
  - IEA for verification engineer

In summary …

Autonomous System => IEA

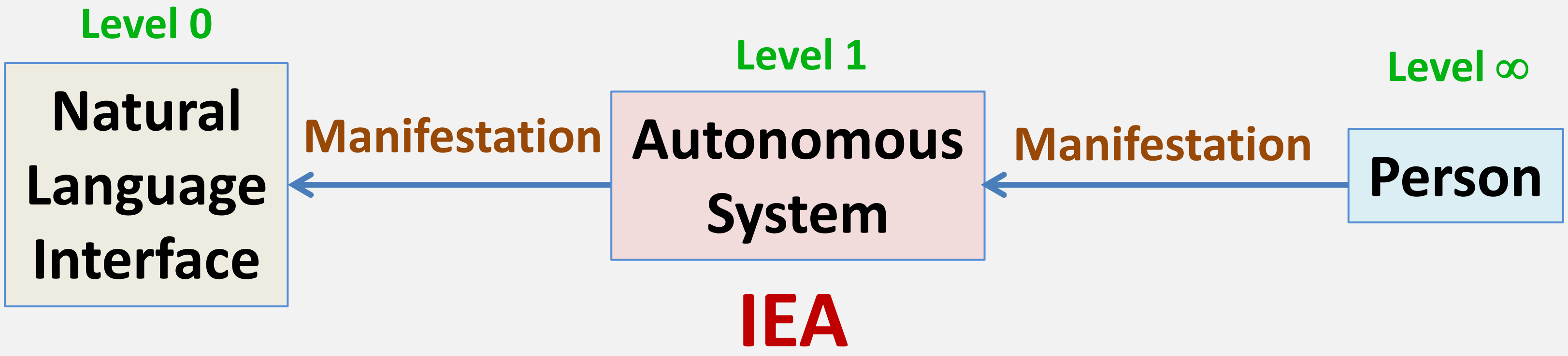


- **Tool Invocation**: Collect all relevant “data”
- **Result Evaluation**: Recognize what “data” mean
- **Reason & Control**: What to do next

What "Engineering Intelligence" Means in IEA

- One my student asked: "Can I use an API for IEA"?
  - Image what would be like by taking away the wheel and asking people to drive car using their voice …
  - The intelligence is not about using natural language to invoke commands to instruct IEA how to do the task

- The language interface is mostly used for queries of results after the autonomous execution is completed



What’s Would Be A Good Way To Conclude?

Matrix – The Moive

- The Movie Matrix: "There is No Spoon"

Matrix – The Moive

IEA Quote

- “Do not try achieving AI … That is impossible.”
- “Instead … only try to realize the truth.”
- “What truth?” (you may say)
- “There is no AI!”
- “Then, you will see. It is not the AI that’s achieved …”
- “It is only yourself.”

- So, I haven’t answered the question “To what extent can machine replace a person?”, or have I?
- Well … I still don’t know exactly … but I suspect it will be close to 100% in the foreseeable future

THANK YOU