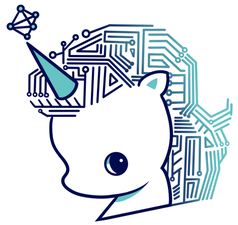
Advanced Topics I

Principles of Deep Learning (AI502/IE408/IE511)

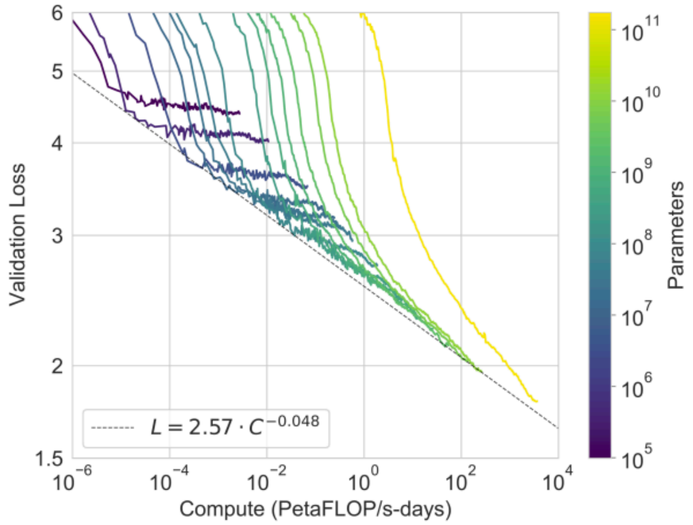
Sungbin Lim (UNIST AIGS & IE)

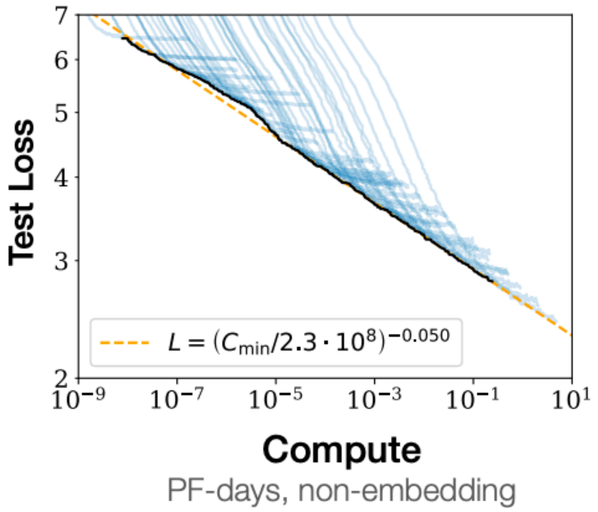


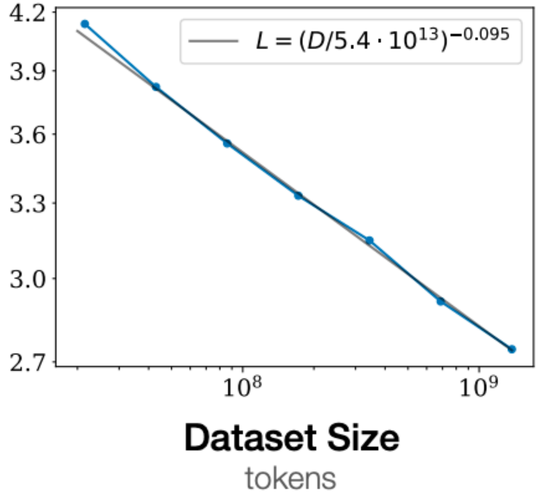
Contact: ai502deeplearning@gmail.com

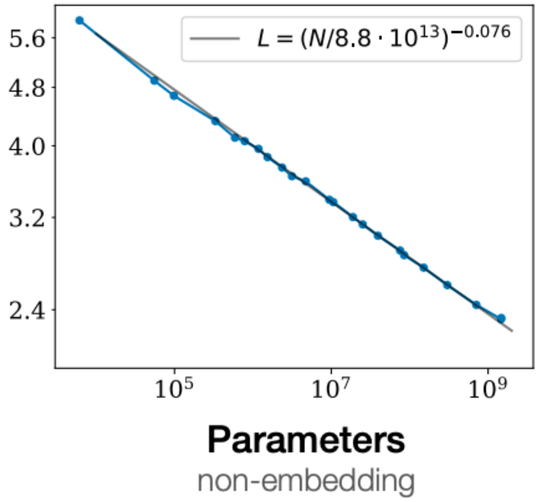
Scaling Law in Transformer

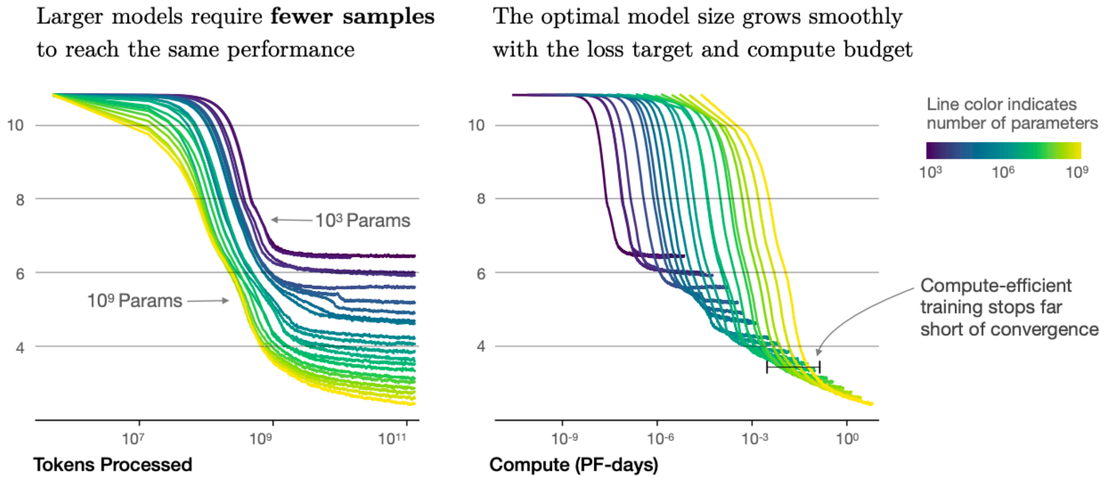
- The empirical scaling behaviors in large transformer models show ***model size***, ***dataset size***, and ***compute budget*** are factors for better performance



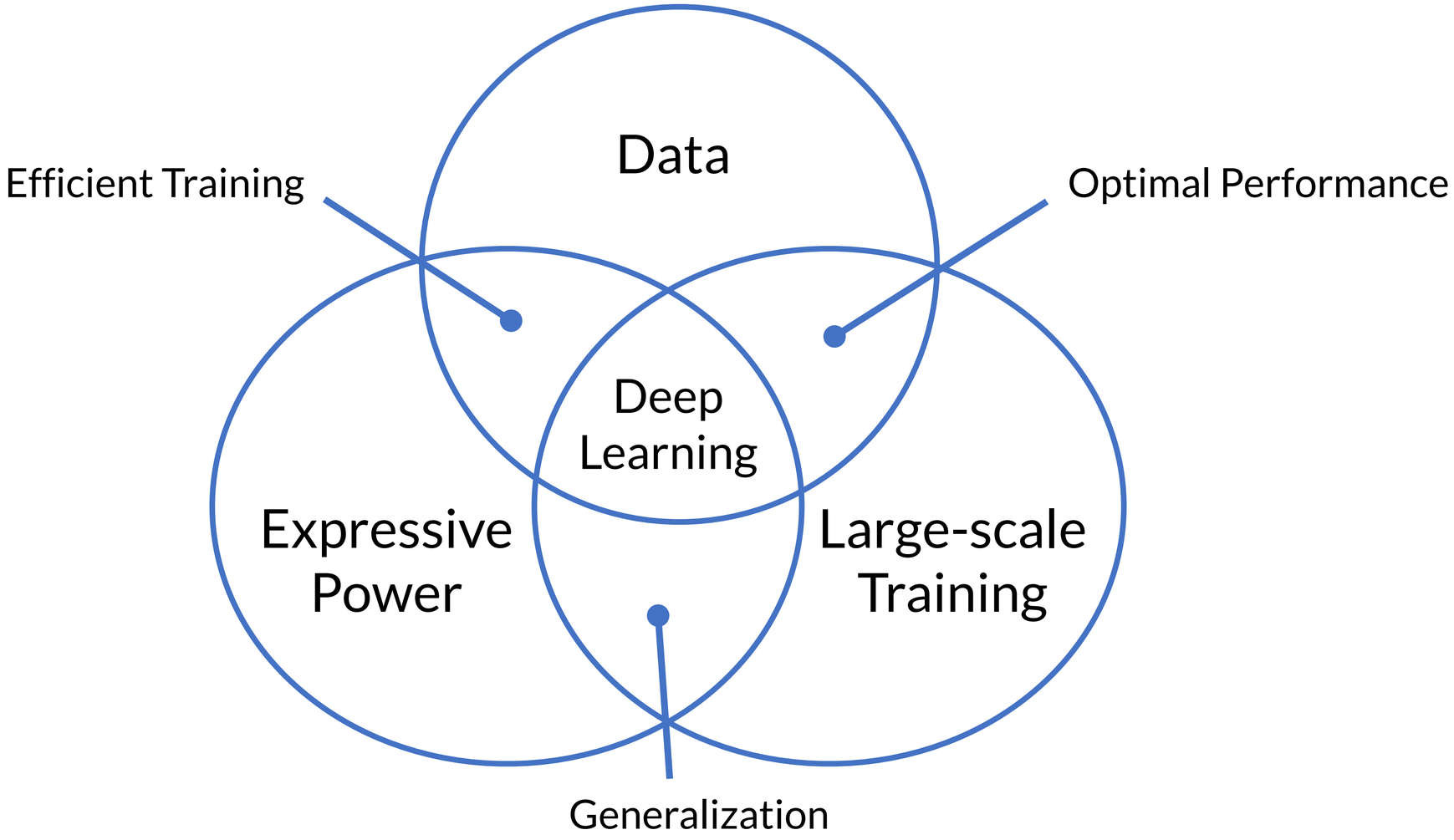








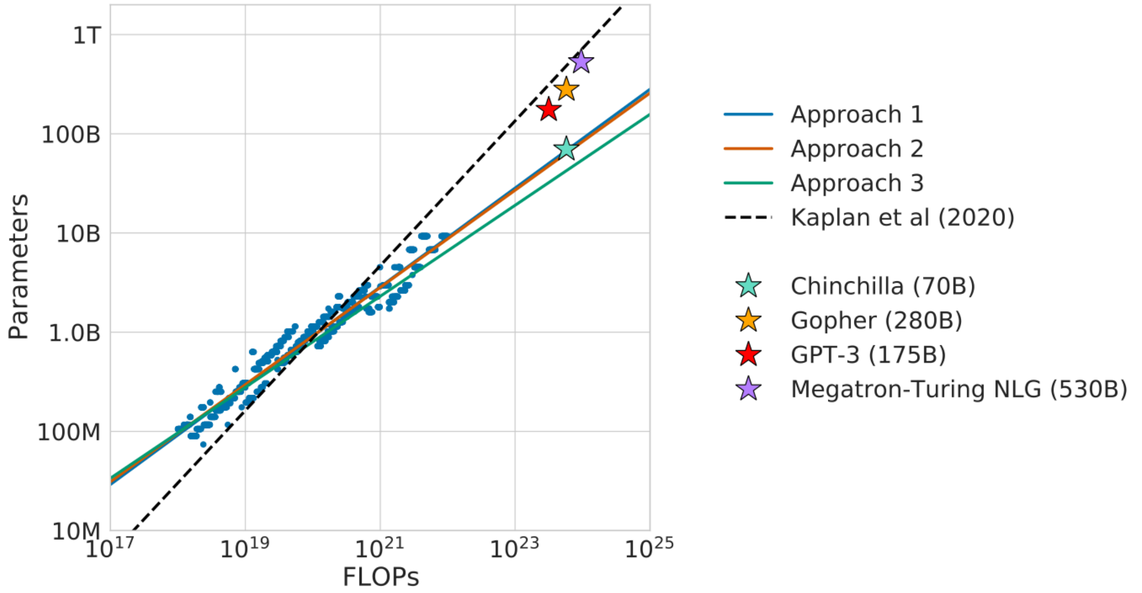
Essence of Deep Learning



Copyright © 2022 by UNIST & LIM Lab

The Era of Large-Scale Language Models

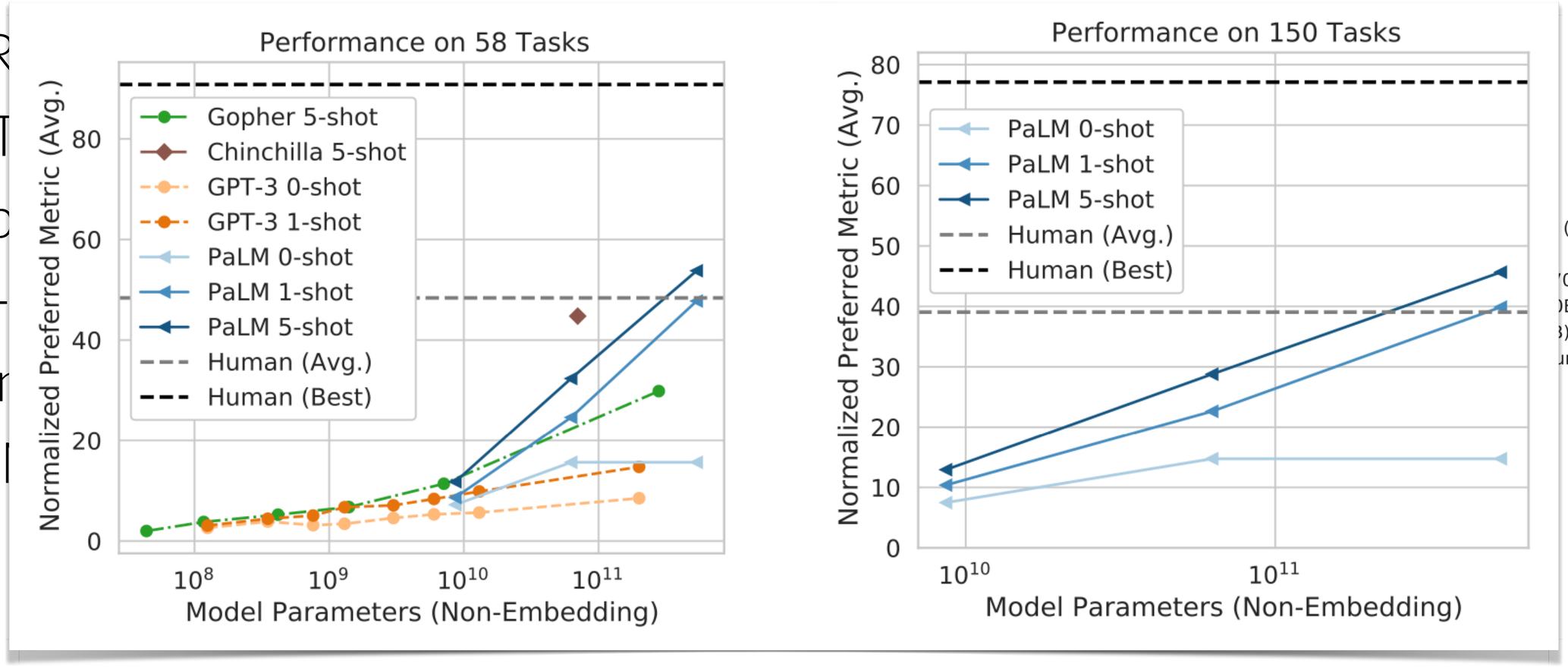
- BERT (Devlin et al., ***NAACL*** 2019)
- GPT-3 (Brown et al., ***NeurIPS*** 2020)
- Gopher (Rae et al., 2021)
- MT-NLG (Smith et al., 2022)
- Chinchilla (Hoffmann et al., 2022)
- PaLM (Chowdhery et al., 2022)



Copyright © 2022 by UNIST & LIM Lab

The Era of Large-Scale Language Models

- BER
- GPT
- Gop
- MT-
- Chir
- PaLI



(2020)

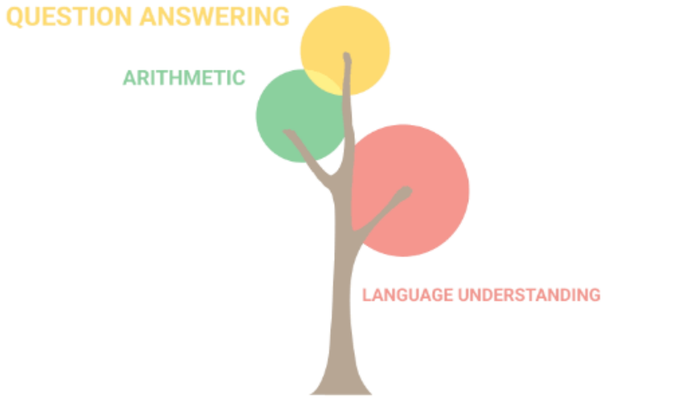
0B)

)B)

)

uring NLG (530B)

LLM Can Learn Multi-Tasks



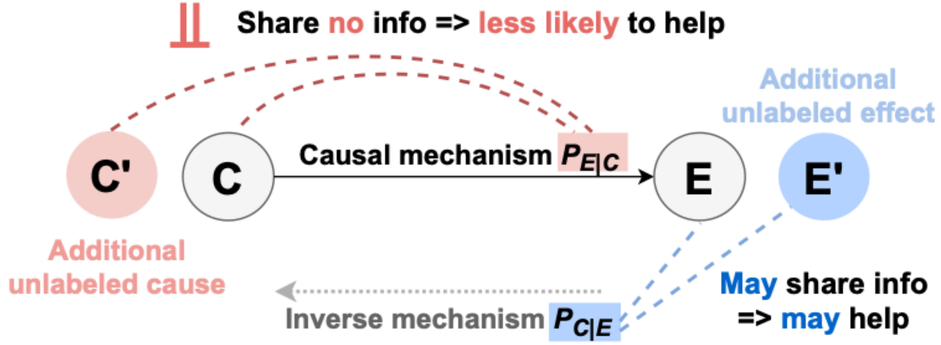
Google AI Blog (2022)

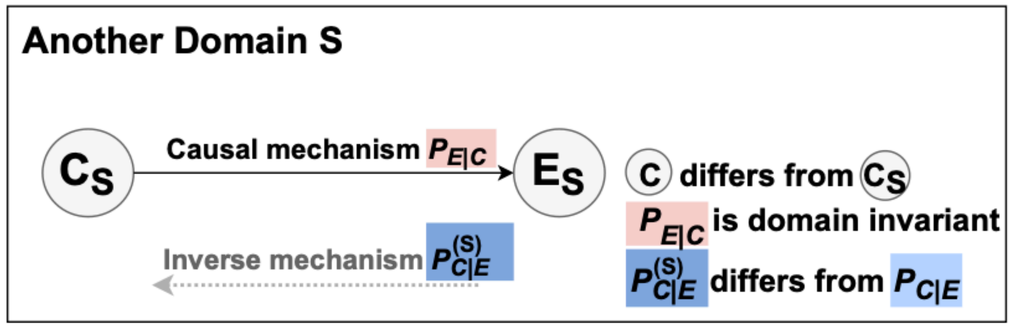
PaLM: Scaling Language Modeling with Pathways, Chowdhery et al., 2022

Data Collection Matters!

Category	Example NLP Tasks
Causal learning	Summarization, parsing, tagging, data-to-text generation, information extraction
Anticausal learning	Author attribute classification, review sentiment classification
Other/mixed (depending on data collection)	Machine translation, question answering, question generation, text style transfer, intent classification

- Independent Causal Mechanisms (ICM) principle implies:
  - the causal NLP tasks are less likely to show improvements by SSL
  - DA should be easier for causal NLP tasks than anti-causal tasks





- Annotating the causal direction is necessary when collecting NLP data

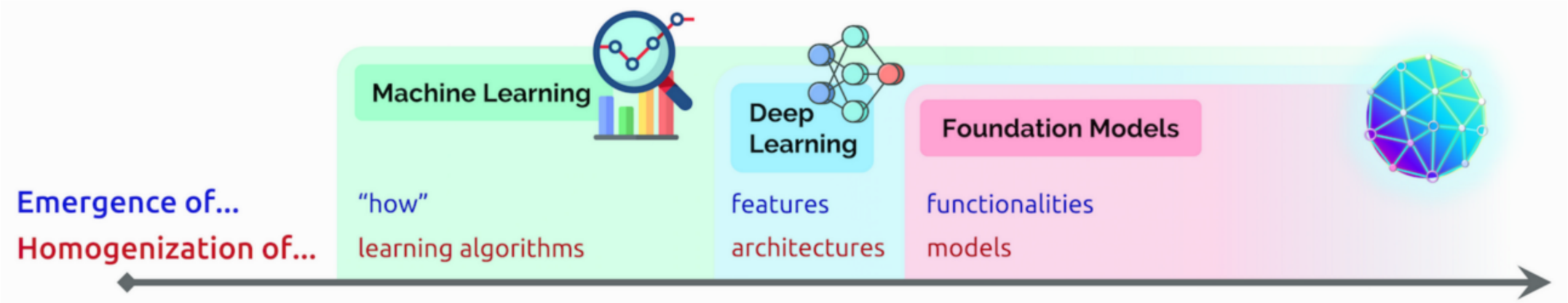
Causal Direction of Data Collection Matters: Implications of Causal and Anticausal Learning for NLP, Jin et al., ***EMNLP*** 2021

Foundation Models

Copyright © 2022 by UNIST & LIM Lab

What is a Foundation Model?

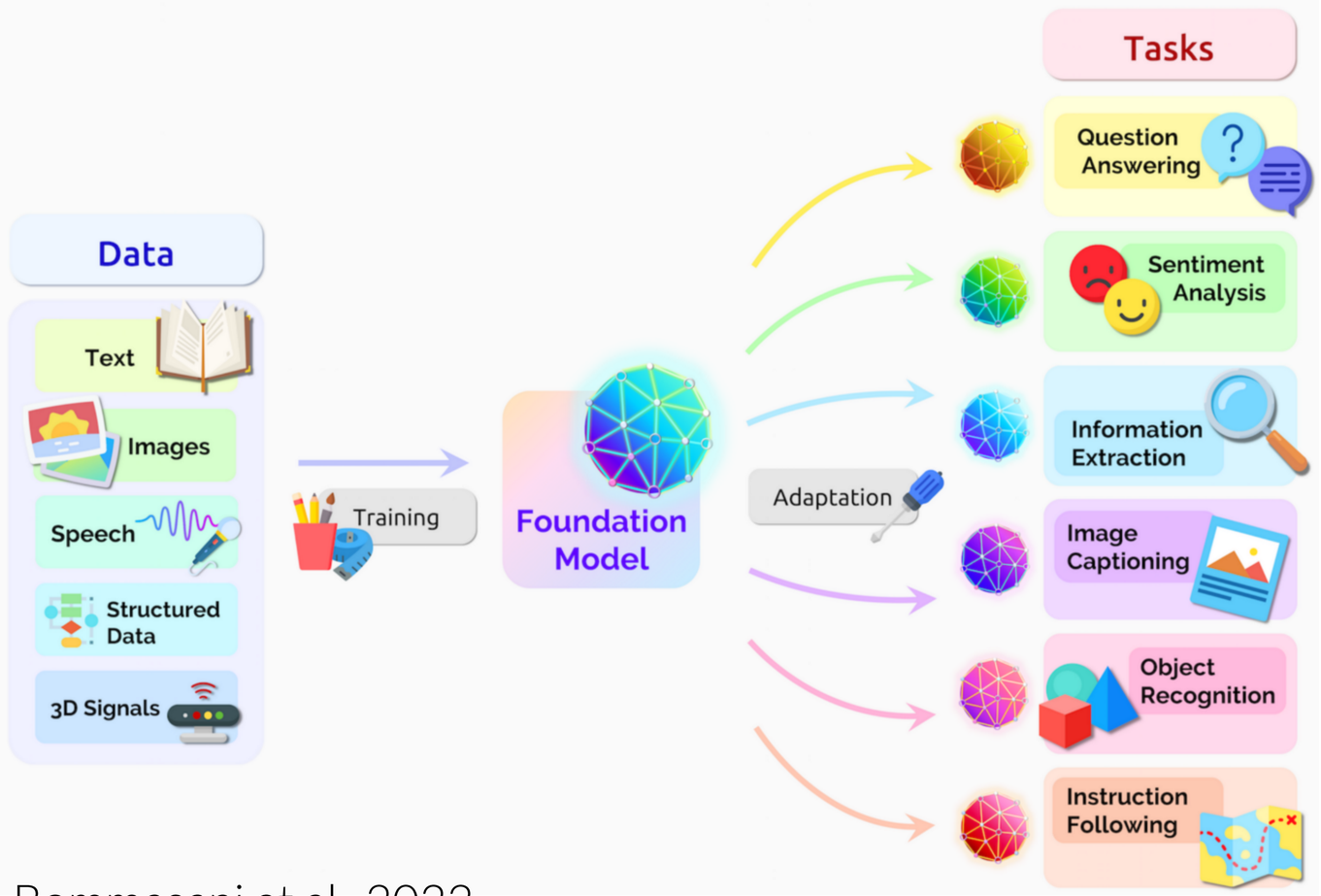
- Stanford announced establishing the Center for Research on Foundation Models (CRFM) as part of the Stanford Institute for Human Centered AI
- Emergence & Homogenization
  -



On the Opportunities and Risks of Foundation Models, Bommasani et al., 2021

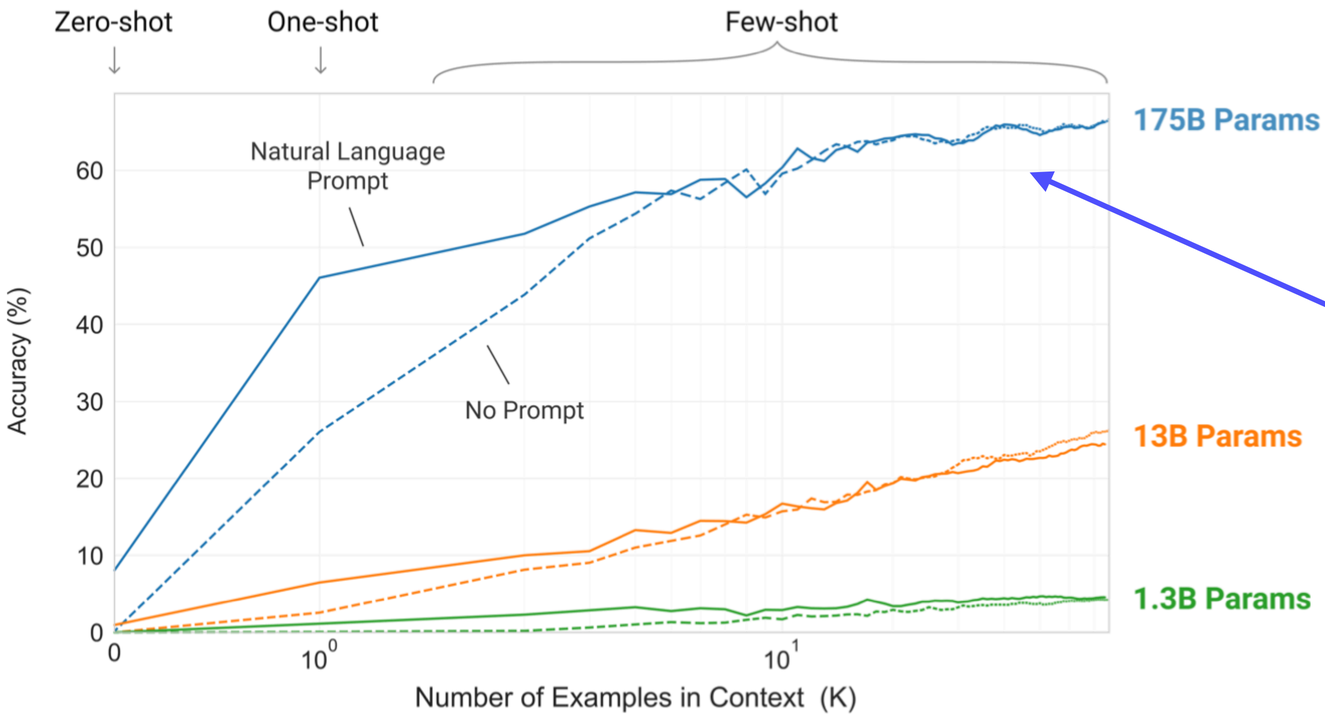
What is a Foundation Model?

- Foundation Model is defined as any model that is trained on broad data at scale
- Can be adapted to a wide range of downstream tasks
  - fine-tuning
  - context learning

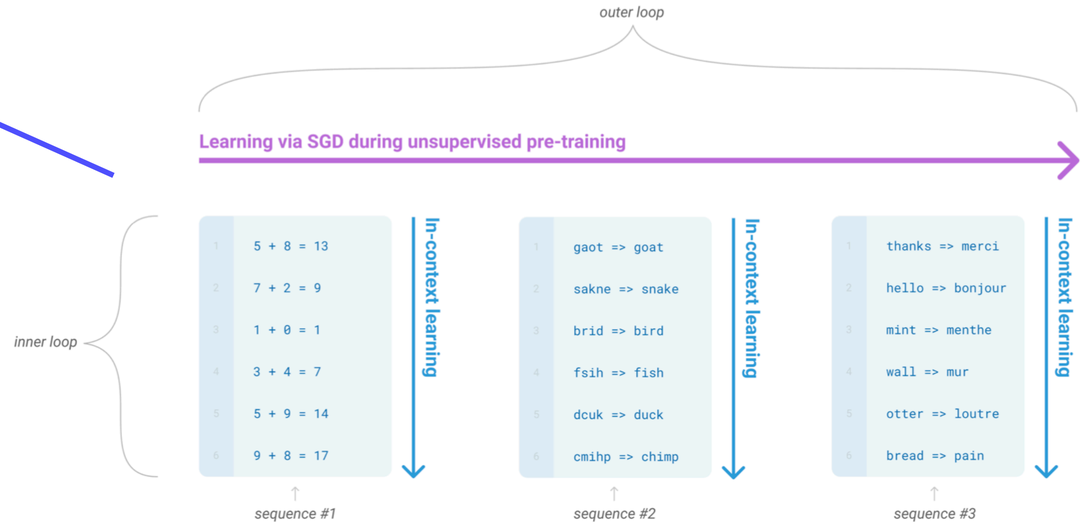


On the Opportunities and Risks of Foundation Models, Bommasani et al., 2022

GPT-3: LMs are *in-context* Learners



require 355 years and
$4.5M to train (Tesla V100)
+ trained with 499 billion tokens



without fine-tuning, GPT-3 achieves improved performance with in-context demonstration

Language Models are Few-Shot Learners, Brown et al., ***NeurIPS*** 2020

Chain-of-Thought: Let's think step by step

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

this is all you need!

Large Language Models are Zero-Shot Reasoners, Kojima et al., 2022

Chain-of-Thought: Let's think step by step

	MultiArith	GSM8K
Zero-Shot	**17.7**	**10.4**
Few-Shot (2 samples)	33.7	15.6
Few-Shot (8 samples)	33.8	15.6
Zero-Shot-CoT	**78.7**	**40.7**
Few-Shot-CoT (2 samples)	84.8	41.3
Few-Shot-CoT (4 samples : First) (*1)	89.2	-
Few-Shot-CoT (4 samples : Second) (*1)	90.5	-
Few-Shot-CoT (8 samples)	93.0	48.7
Zero-Plus-Few-Shot-CoT (8 samples) (*3)	**92.8**	**51.5**
Finetuned GPT-3 175B (*2)	-	33
Finetuned GPT-3 175B + verifier (*2)	-	55
PaLM 540B: Zero-Shot	**25.5**	**12.5**
PaLM 540B: Zero-Shot-CoT	**66.1**	**43.0**
PaLM 540B: Few-Shot (*2)	-	17.9
PaLM 540B: Few-Shot-CoT (*2)	-	58.1

MultiArith

Q: Adam bought 2 boxes of chocolate candy and 5 boxes of caramel candy. If each box has 4 pieces inside it, how much candy did he have total?

A: Let's think step by step. Adam bought 2 boxes of chocolate candy and 5 boxes of caramel candy. We know that each box has 4 pieces inside it. So, we can multiply 2 by 4 to find out how many pieces of chocolate candy Adam bought. This will give us 8. We can also multiply 5 by 4 to find out how many pieces of caramel candy Adam bought. This will give us 20. Now, we can add 8 and 20 together to find out how much candy Adam bought in total. 8 + 20 = 28 Adam bought 28 pieces of candy in Therefore, the answer (arabic numerals) is 28. ✓

GSM8K

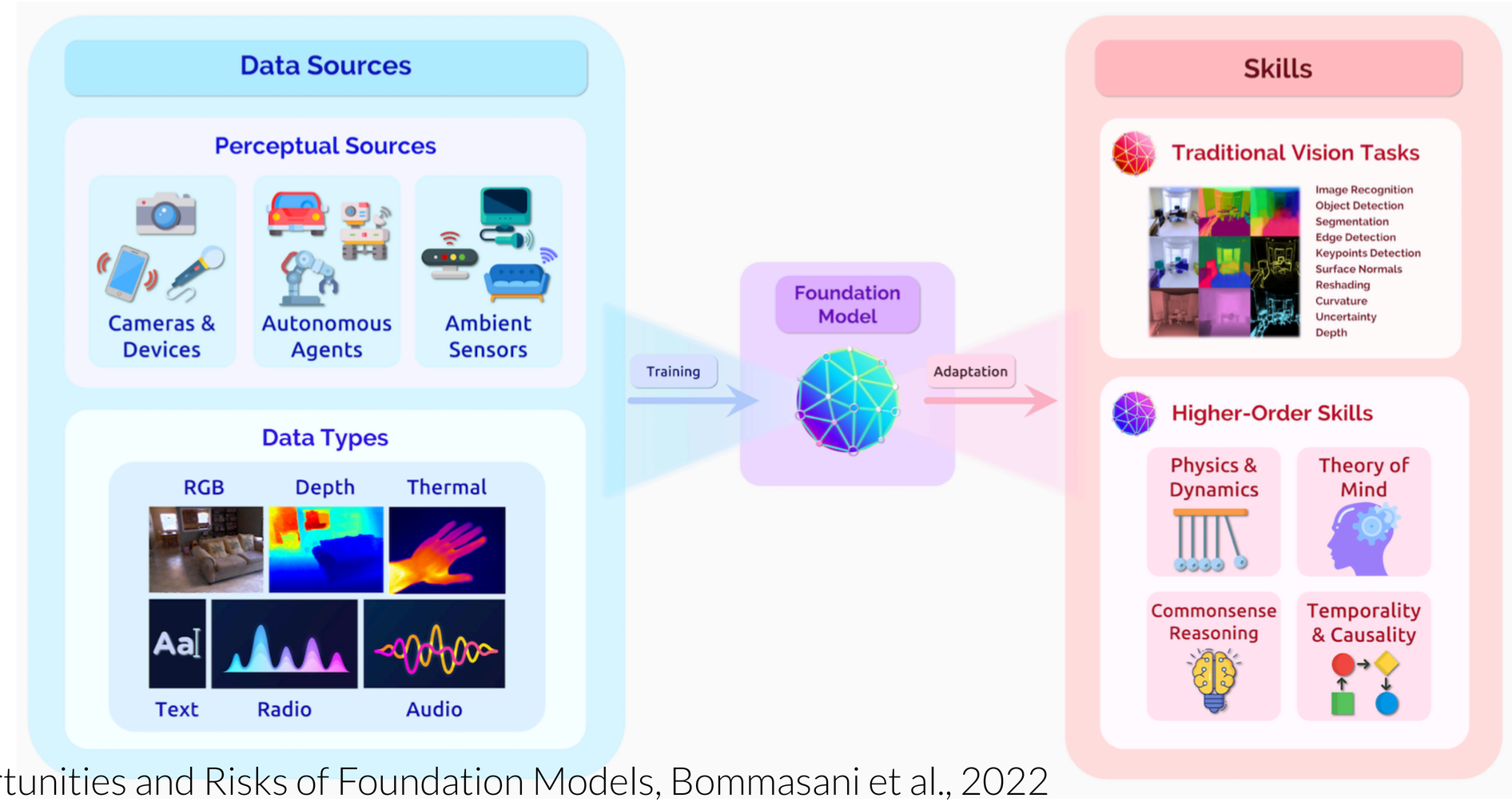
Q: James buys a plane. The plane cost $150,000. He pays $5000 a month to rent a hanger to keep it in. He also spends twice as much as that on fuel per month. How much did it cost him to get and maintain the plane for the first year?

A: Let's think step by step. James buys a plane. This costs him $150,000. He pays $5000 a month to rent a hanger. This costs him $60,000 for the year. He spends twice as much as that on fuel per month. This costs him $120,000 for the year. So the total cost for the first year is $150,000 + $60,000 + $120,000 = $330,000. Therefore, the answer (arabic numerals) is 330,000. ✓

Large Language Models are Zero-Shot Reasoners, Kojima et al., 2022

Copyright © 2022 by UNIST & LIM Lab

Computer Vision

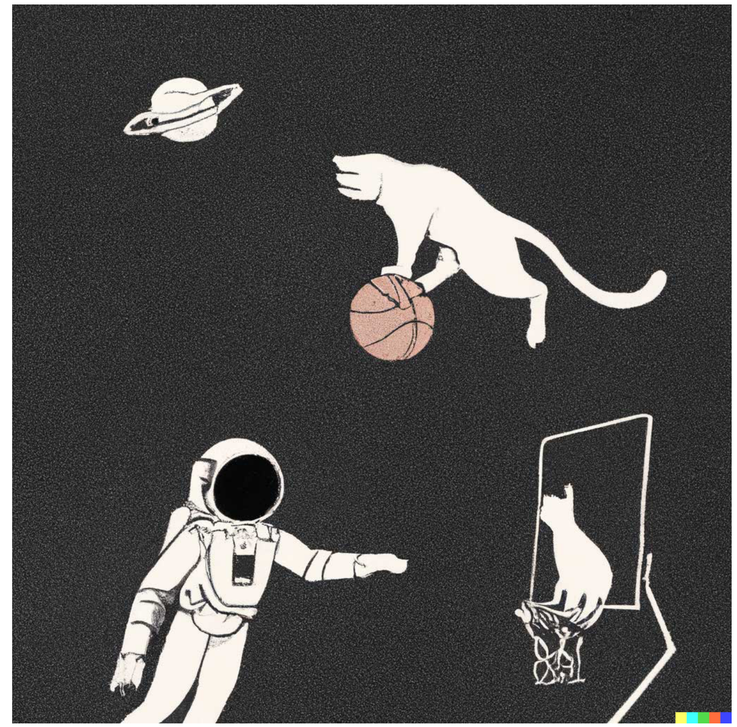


On the Opportunities and Risks of Foundation Models, Bommasani et al., 2022

DALL-2: Text2Image

An astronaut riding a horse
in a photorealistic style

An astronaut lounging in a
tropical resort in space as pixel art

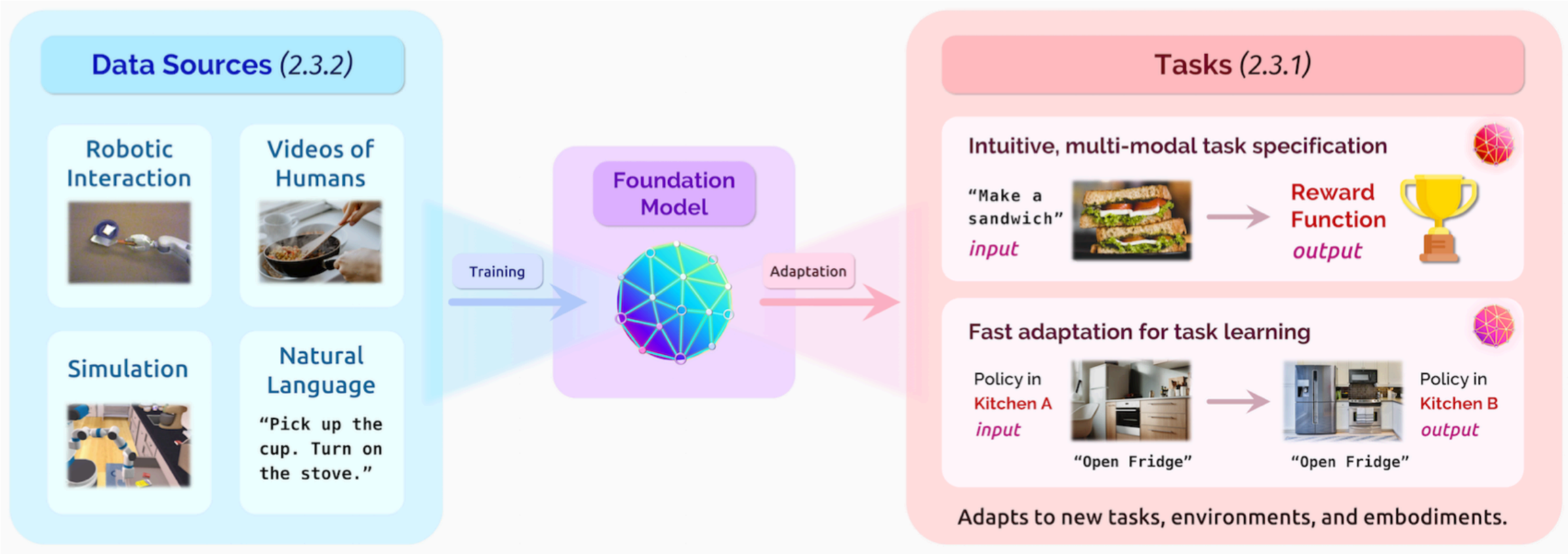


An astronaut playing basketball
with cats in space in a minimalist style

Hierarchical Text-Conditional Image Generation with CLIP Latents, Ramesh et al., 2022

Copyright © 2022 by UNIST & LIM Lab

Robotics



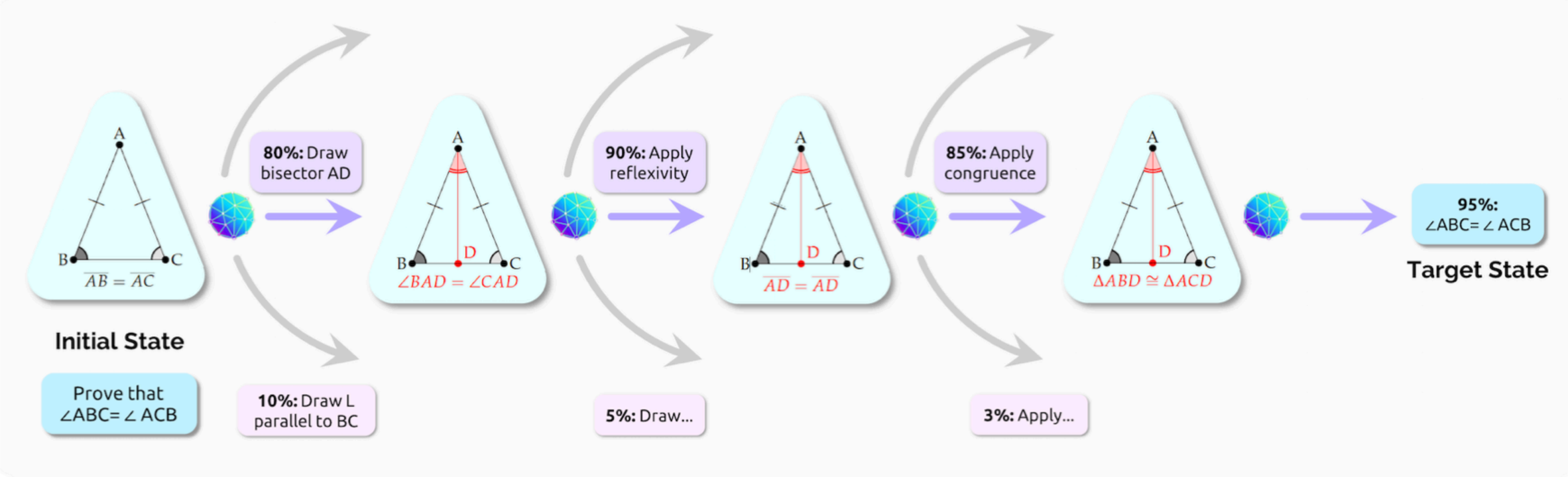
On the Opportunities and Risks of Foundation Models, Bommasani et al., 2022

Inner Monologue: Text2Planning

Inner Monologue: Embodied Reasoning through Planning with Language Models, Huang et al., 2022

Copyright © 2022 by UNIST & LIM Lab

Reasoning and Search



On the Opportunities and Risks of Foundation Models, Bommasani et al., 2022

Input: When I found out my grandma was in the hospital I felt a particular color. When someone cut me off in traffic I felt a different color. What is the most likely color I would see if I combined these two colors?

Model Output: The color that I felt when I found out my grandma was in the hospital was likely a sad color, like blue. The color that I felt when someone cut me off in traffic was likely an angry color, like red. If I combined these two colors, I would likely see purple. The answer is "purple".

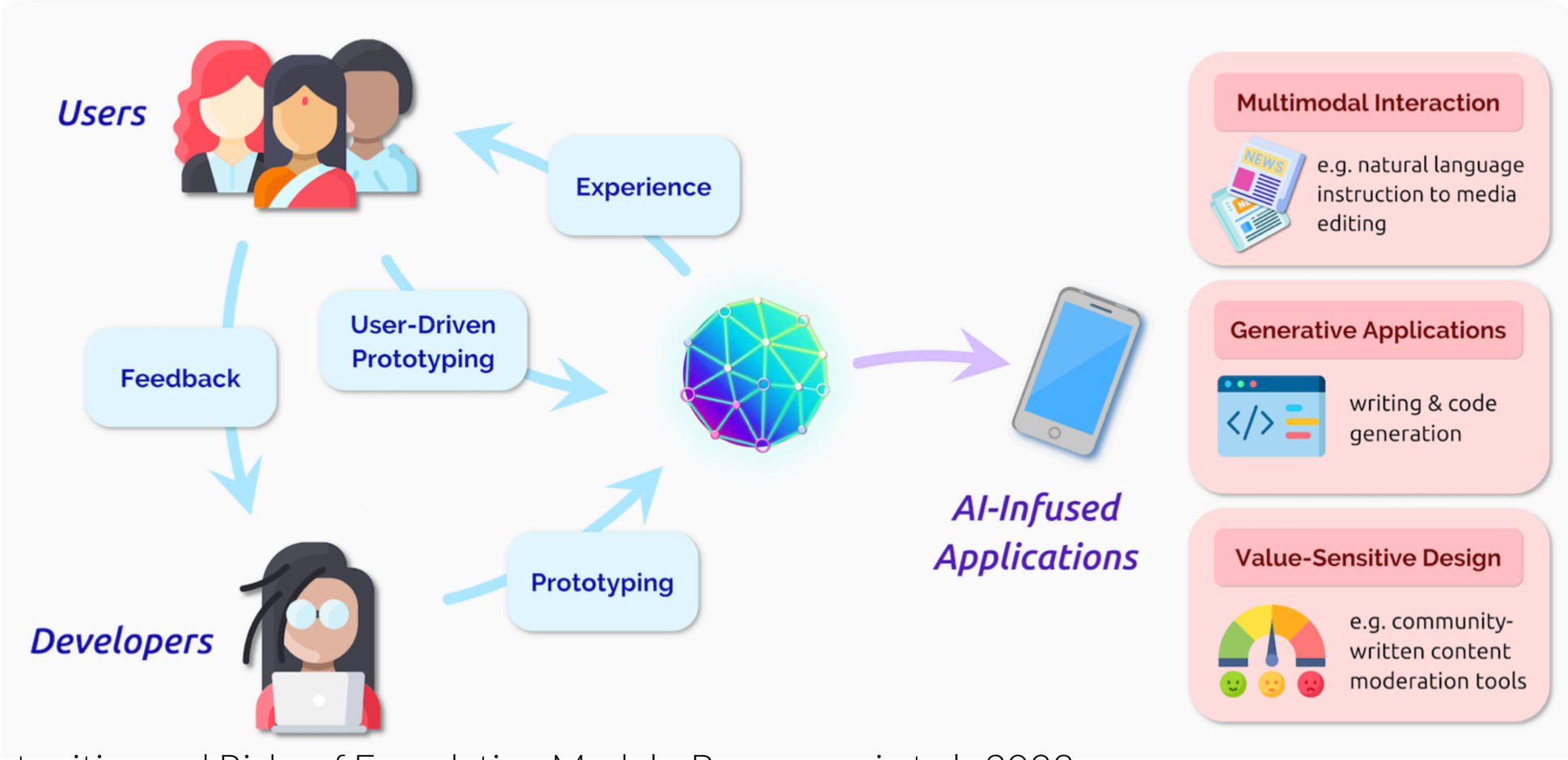
Input: Michael is at that really famous museum in France looking at its most famous painting. However, the artist who made this painting just makes Michael think of his favorite cartoon character from his childhood. What was the country of origin of the thing that the cartoon character usually holds in his hand?

Model Output: The most famous painting in the Louvre is the Mona Lisa. The artist who made the Mona Lisa is Leonardo da Vinci. Leonardo da Vinci is also the name of the main character in the cartoon Teenage Mutant Ninja Turtles. Leonardo da Vinci is from Italy. The thing that Leonardo da Vinci usually holds in his hand is a katana. The country of origin of the katana is Japan. The answer is "Japan".

PaLM: Scaling Language Modeling with Pathways, Chowdhery et al., 2022

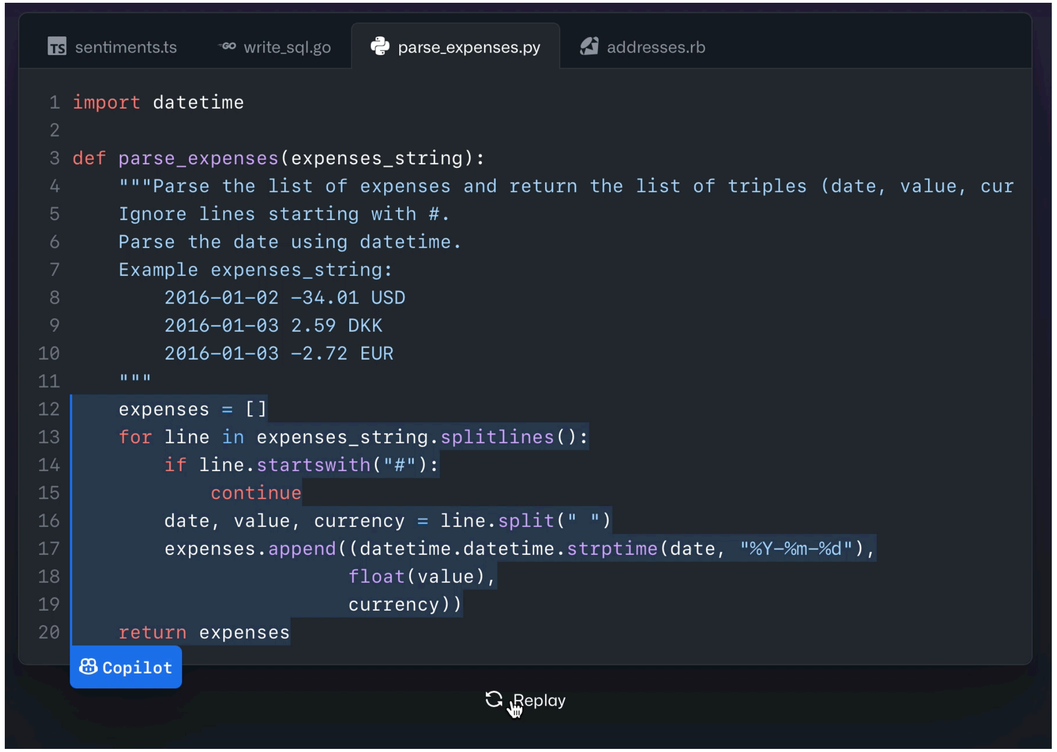
Copyright © 2022 by UNIST & LIM Lab

Interaction

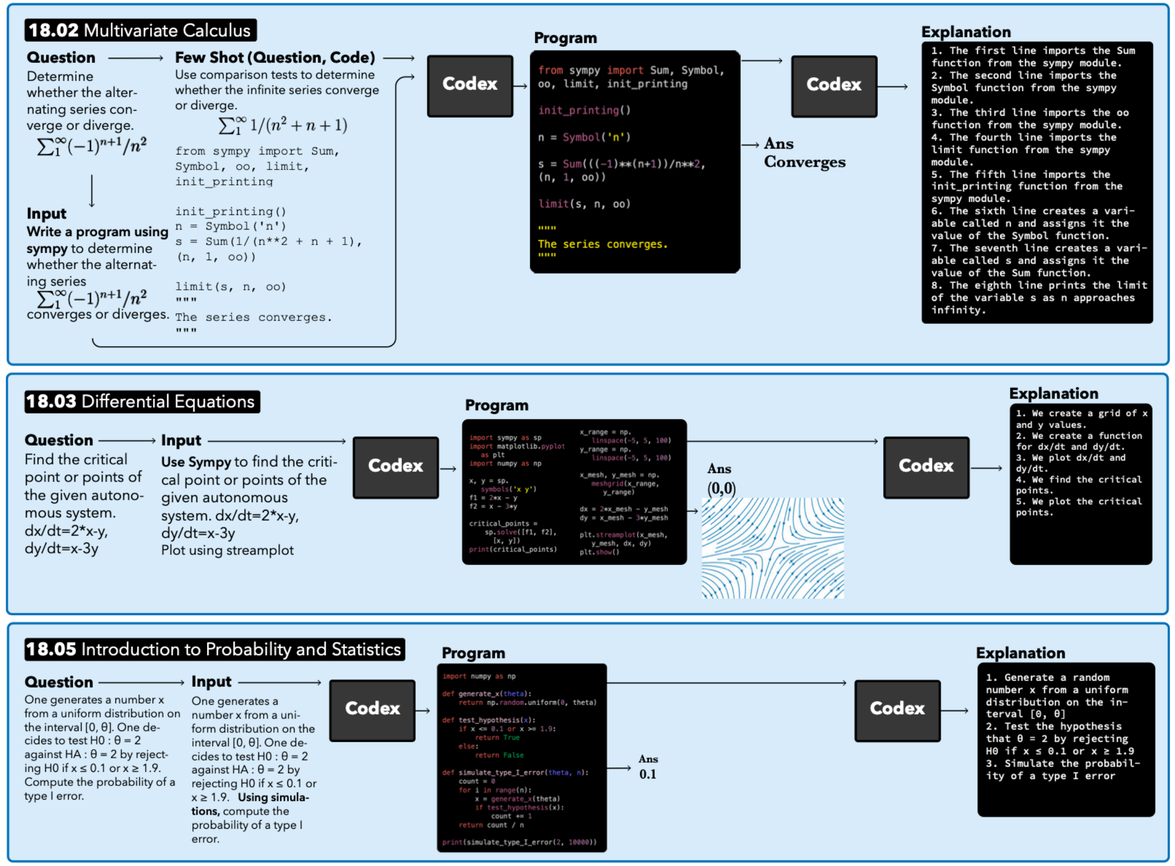


On the Opportunities and Risks of Foundation Models, Bommasani et al., 2022

Codex: Text2Code



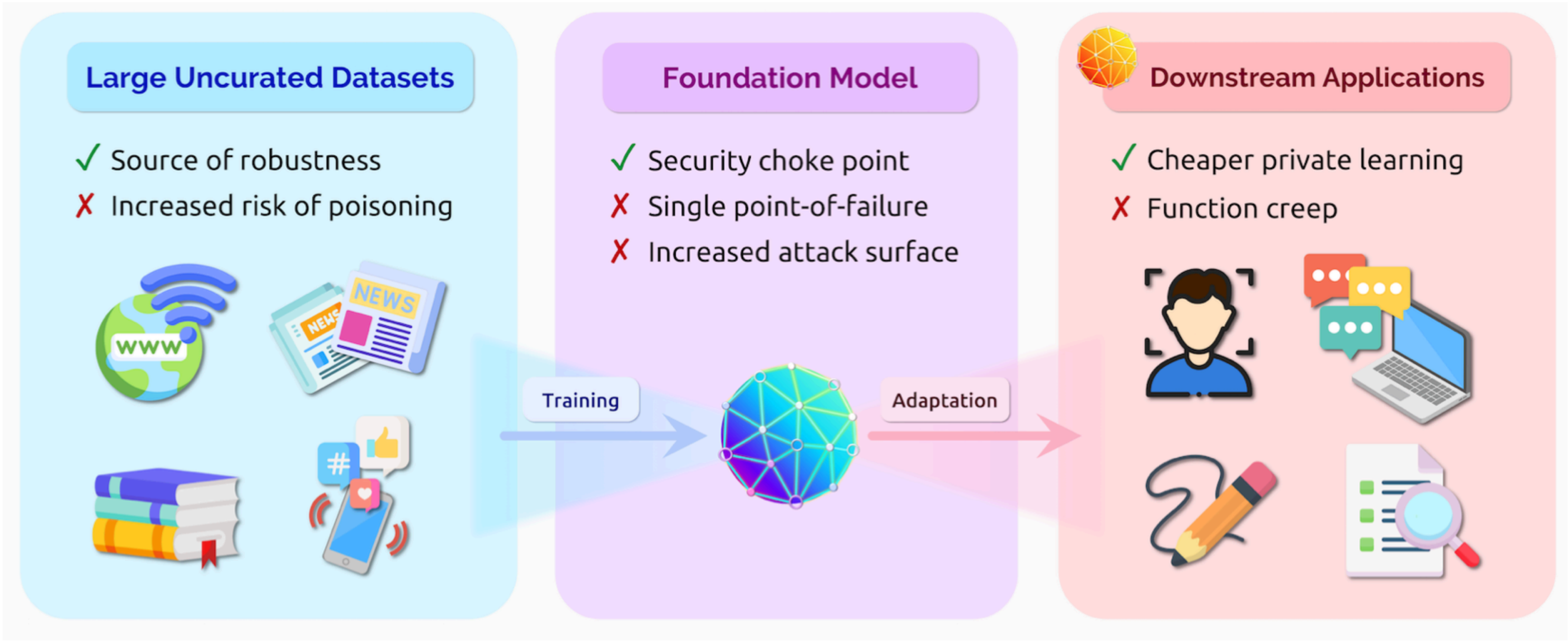
GitHub Copilot



Drori et al., (2022)

Evaluating Large Language Models Trained on Code, Chen et al., 2021

Societal Impact



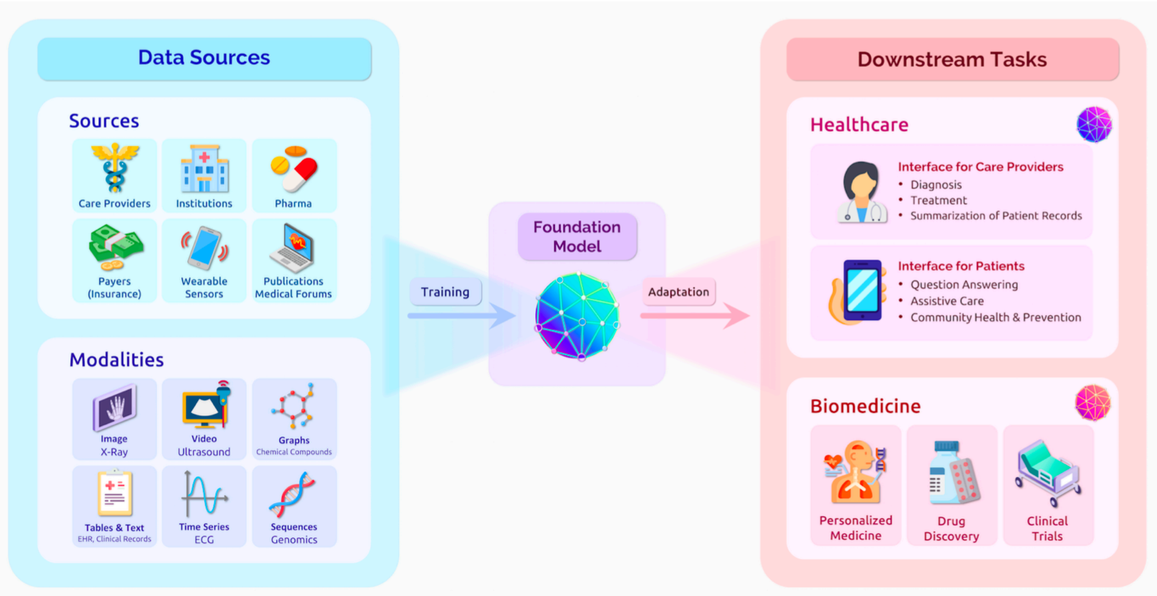
On the Opportunities and Risks of Foundation Models, Bommasani et al., 2022

The Paradigm Shift

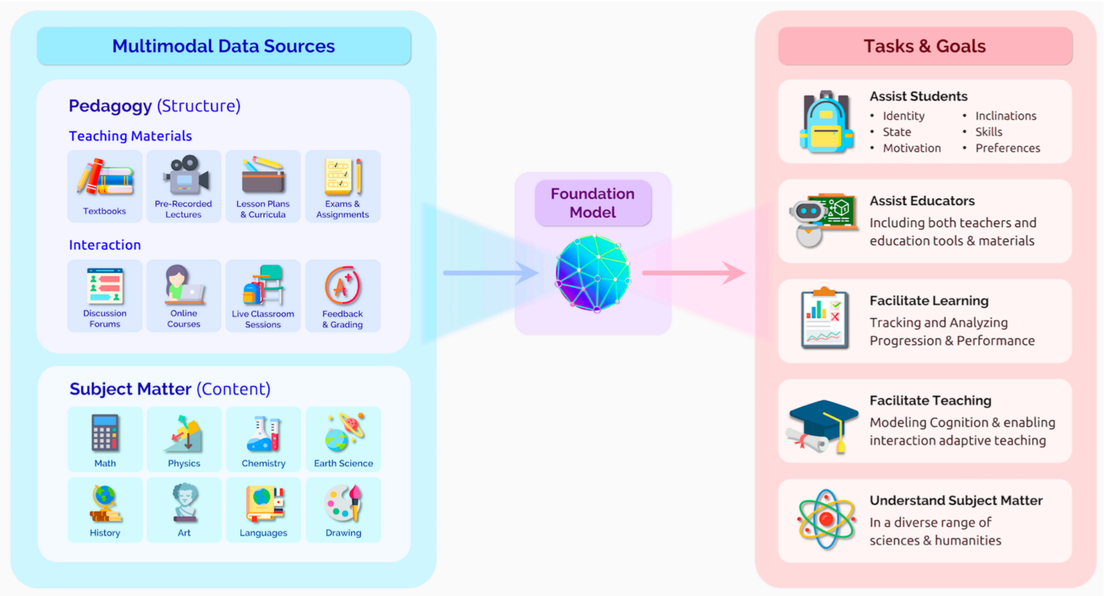
- A few years ago, most NLP tasks you had to gather a decent training dataset then train a model for solving the task from scratch
  - you cannot train a large network as it is a data-hungry process
- The situation has changed completely!
  - models like BERT are easily available and modern libraries like **`Huggingface`** Transformers lowered the barriers even further
  - you can have a really good model having spent only tens to hundreds of dollars training it
- AI democratization is happening

Copyright © 2022 by UNIST & LIM Lab

Applications of Foundation Models



Healthcare

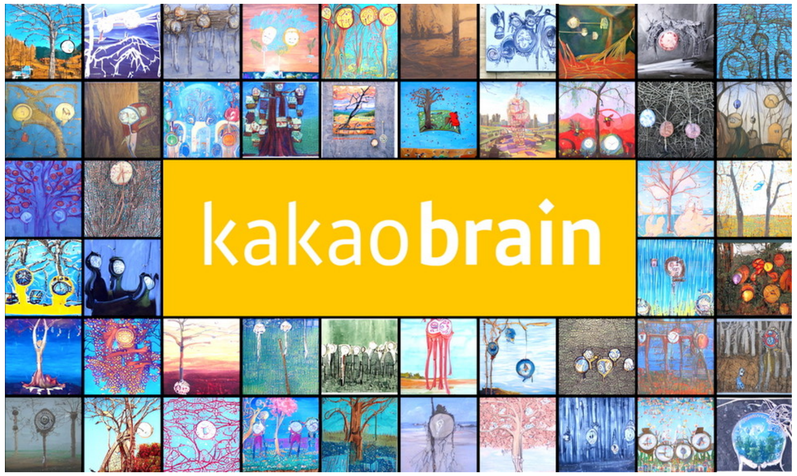


Education

On the Opportunities and Risks of Foundation Models, Bommasani et al., 2022

Large-Scale Foundation Models in Korea

HyperCLOVA (NAVER)



KoGPT, minDALL-E (kakaobrain)

EXAONE (LG AI Research)

Holistic Evaluation of Language Models

Percy Liang† Rishi Bommasani† Tony Lee†[1]
Dimitris Tsipras* Dilara Soylu* Michihiro Yasunaga* Yian Zhang* Deepak Narayanan* Yuhuai Wu*[2]

Ananya Kumar Benjamin Newman Binhang Yuan Bobby Yan Ce Zhang
Christian Cosgrove Christopher D. Manning Christopher Ré Diana Acosta-Navas
Drew A. Hudson Eric Zelikman Esin Durmus Faisal Ladhak Frieda Rong Hongyu Ren
Huaxiu Yao Jue Wang Keshav Santhanam Laurel Orr Lucia Zheng Mert Yuksekgonul
Mirac Suzgun Nathan Kim Neel Guha Niladri Chatterji Omar Khattab Peter Henderson
Qian Huang Ryan Chi Sang Michael Xie Shibani Santurkar Surya Ganguli
Tatsunori Hashimoto Thomas Icard Tianyi Zhang Vishrav Chaudhary William Wang
Xuechen Li Yifan Mai Yuhui Zhang Yuta Koreeda

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

Holistic Evaluation of Language Models, Liang et al., 2022

Copyright © 2022 by UNIST & LIM Lab

Pearl Causal Hierarchical

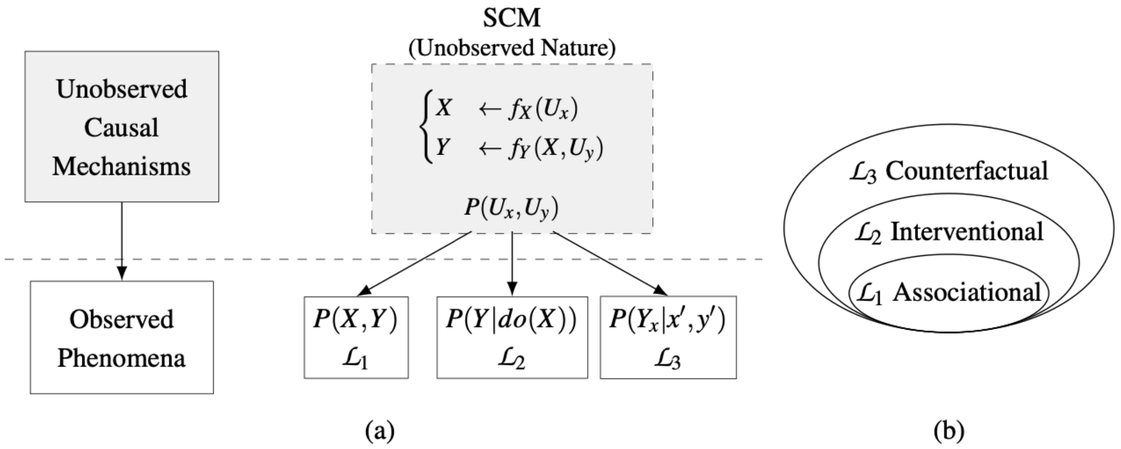


Figure 1.1: (a) Collection of causal mechanisms (or SCM) generating certain observed phenomena (qualitatively different probability distributions). (b) PCH's containment structure.

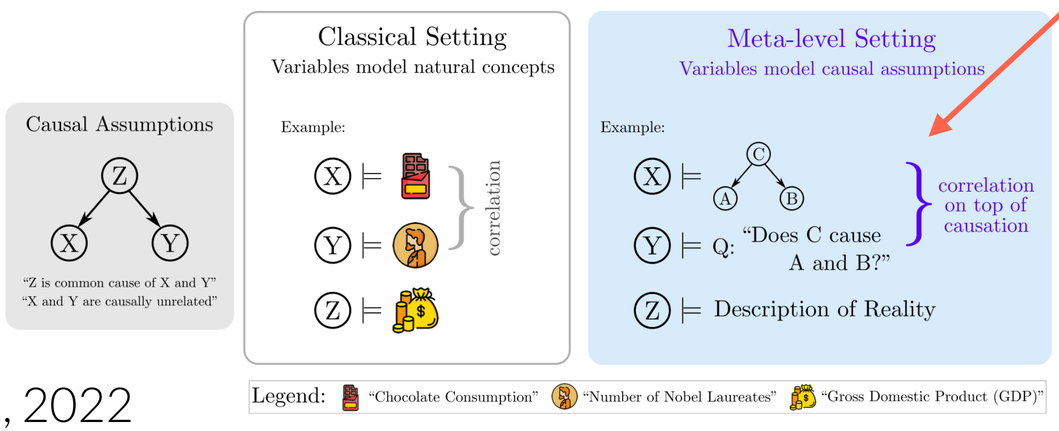
	Layer (Symbolic)	Typical Activity	Typical Question	Example	Machine Learning
$\mathcal{L}_1$	Associational $P(y\|x)$	Seeing	What is? How would seeing X change my belief in Y?	What does a symptom tell us about the disease?	Supervised / Unsupervised Learning
$\mathcal{L}_2$	Interventional $P(y\|do(x), c)$	Doing	What if? What if I do X?	What if I take aspirin, will my headache be cured?	Reinforcement Learning
$\mathcal{L}_3$	Counterfactual $P(y_x\|x', y')$	Imagining	Why? What if I had acted differently?	Was it the aspirin that stopped my headache?	

Theorem 1. *[Causal Hierarchy Theorem (CHT), informal version] The* PCH *almost never collapses. That is, for almost any SCM, the layers of the hierarchy remain distinct.* ■

On Pearl's Hierarchy and the Foundations of Causal Inference, Bareinboim et al., ***Probabilistic and Causal Inference***, 2022

CHT and Foundation Models

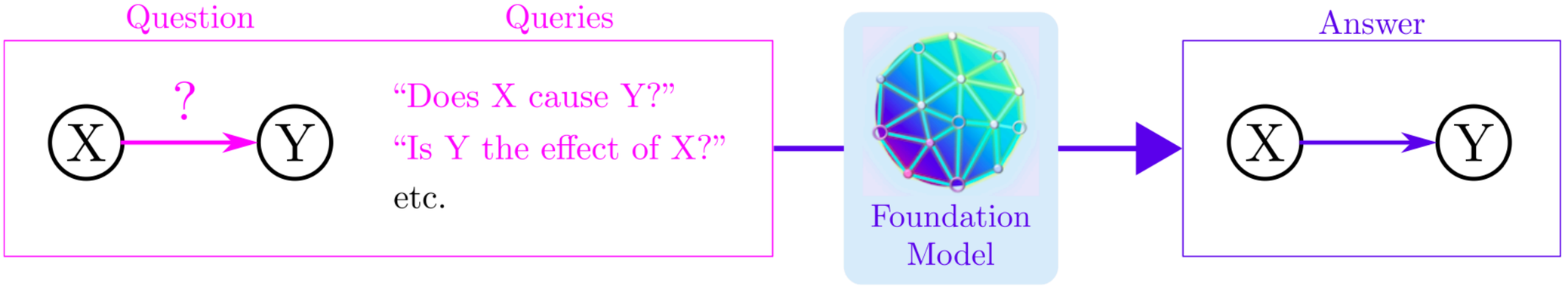
- According to CHT, no matter how much we scale our models, we will never be able to perform causal inference.
  - what about foundation models?
- What happens if the causal assumptions are represented in data?
  - structural causal model (SCM) can model data generating processes!
  - understanding ≠ knowing
  - FMs can be used ***head start*** learning



read books!

Can Foundation Models Talk Causality?, Willig et al., 2022

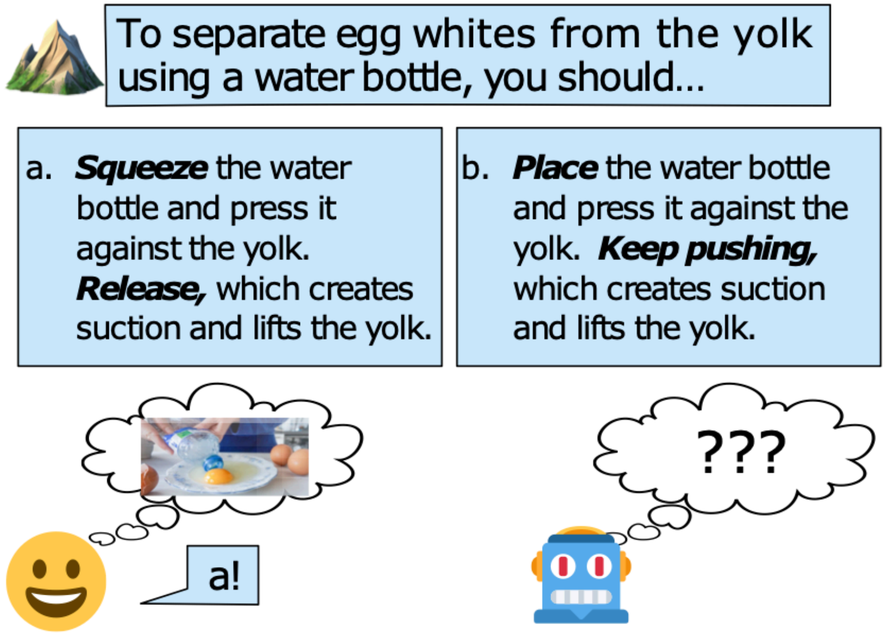
Structure Discovery via Foundation Models



- (Q1) How do the FM graph predictions compare to settings where the causal graph is known?
- (Q2) How do the FM graph predictions perform in commonsense settings that involve abstract reasoning and intuitive physics?
- (Q3) How do synonyms or more general variable name altercations affect the FM graph prediction?

Can Foundation Models Talk Causality?, Willig et al., 2022

Can LLMs infer physical commonsense?



a. Shape, Material, and Purpose

[Goal] Make an outdoor pillow
[Sol1] Blow into a tin can and tie with rubber band ✘
[Sol2] Blow into a trash bag and tie with rubber band ✔

[Goal] To make a hard shelled taco,
[Sol1] put seasoned beef, cheese, and lettuce onto the hard shell. ✘
[Sol2] put seasoned beef, cheese, and lettuce into the hard shell. ✔

[Goal] How do I find something I lost on the carpet?
[Sol1] Put a solid seal on the end of your vacuum and turn it on. ✘
[Sol2] Put a hair net on the end of your vacuum and turn it on. ✔

b. Commonsense Convenience

[Goal] How to make sure all the clocks in the house are set accurately?

[Sol1] Get a solar clock for a reference and place it just outside a window that gets lots of sun. Use a system of call and response once a month, having one person stationed at the solar clock who yells out the correct time and have another person move to each of the indoor clocks to check if they are showing the right time. Adjust as necessary. ✘

[Sol2] Replace all wind-ups with digital clocks. That way, you set them once, and that's it. Check the batteries once a year or if you notice anything looks a little off. ✔

Model	Size	Accuracy (%) Validation	Test
Random Chance		50.0	50.0
Majority Class		50.5	50.4
OpenAI GPT	124M	70.9	69.2
Google BERT	340M	67.1	66.8
FAIR RoBERTa	355M	79.2	77.1
Human		94.9	

GPT-3 - 82.8%, PaLM - 85.2%
they cannot beat human (94.4%) yet

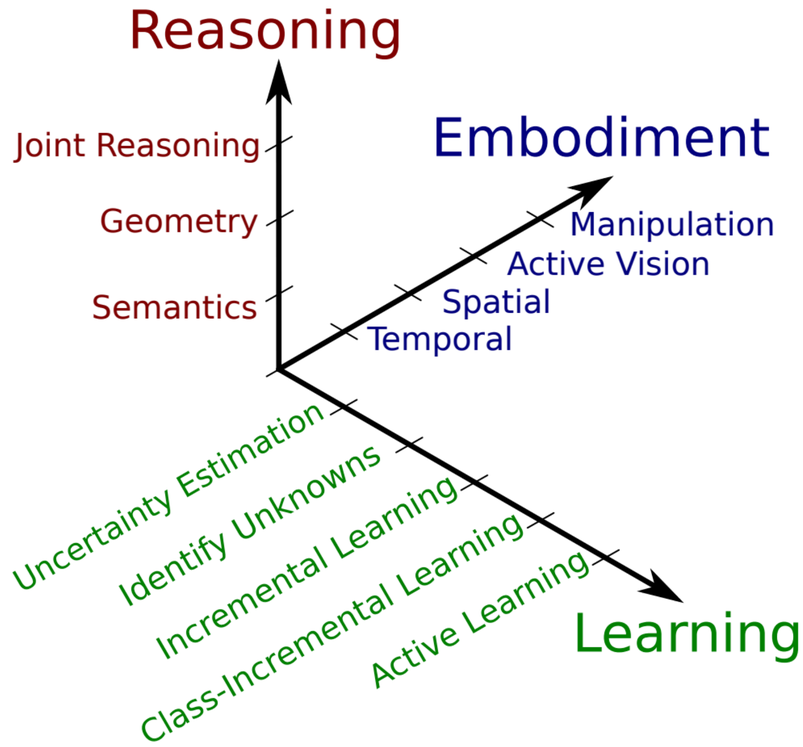
PIQA: Reasoning about Physical Commonsense in Natural Language, Bisk et al., ***AAAI*** 2020

Automated Machine Learning

Copyright © 2022 by UNIST & LIM Lab

The Direction of AI Research

- Challenges of AI Research
  - seeing like human (Detectron)
  - listening like human (LibriSpeech)
  - speaking like human (Tacotron)
  - drawing like human (GAN, Diffusion)
  - understanding like human (BERT)
  - **learning** like human (AutoML?)
  - **reasoning** like human (FM?)



The Limits and Potentials of Deep Learning for Robotics, Niko et al., (2018)

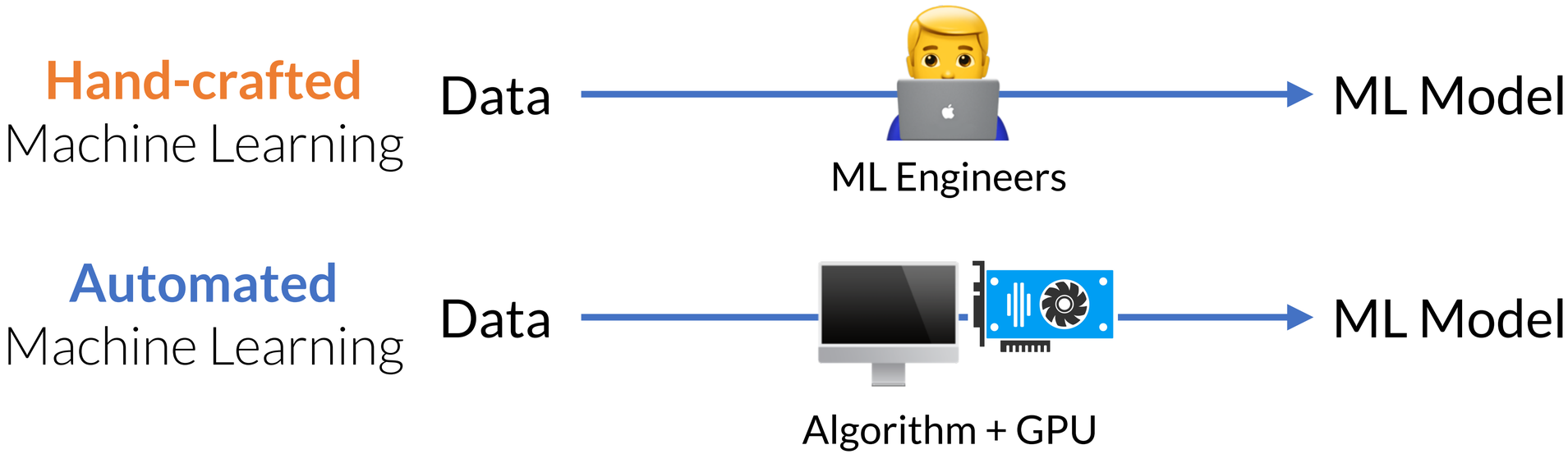
Everyone knows ML/DL is good

- But currently employed architectures have mostly been developed manually by **human experts**
  - time consuming
  - error prone process
- Growing interest in automating machine learning

Copyright © 2022 by UNIST & LIM Lab

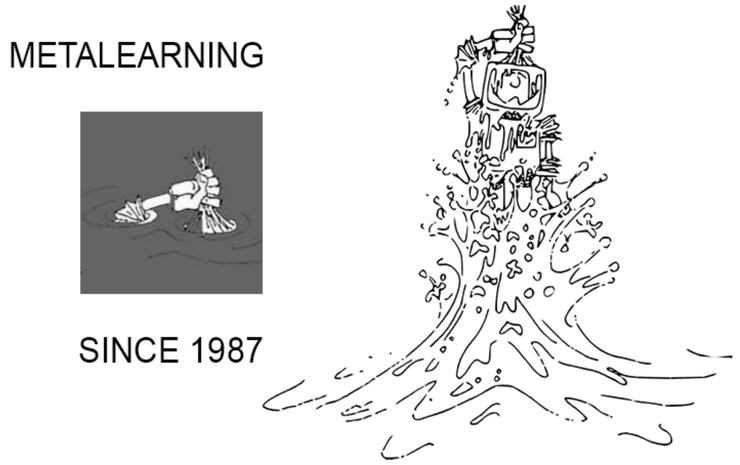
What is AutoML?

- Millions of organizations worldwide has ML problems
- AutoML allows **non-experts** to solve ML problems



Learning vs Meta Learning

- Learning Algorithm A
  - input: $\mathcal{D}_{\texttt{train}} = \{(x_i, y_i)\}$
  - output: M
  - objective: $\mathcal{D}_{\texttt{test}} = \{(\tilde{x}_i, \tilde{y}_i)\}$
- Meta Learning Algorithm $\mathfrak{A}$
  - input: $\mathfrak{D}_{\texttt{meta-train}} = \{(\mathcal{D}^{(t)}_{\texttt{train}}, \mathcal{D}^{(t)}_{\texttt{test}})\}_{t=1}^{T}$
  - output: A
  - objective: $\mathfrak{D}_{\texttt{meta-test}} = \{(\tilde{\mathcal{D}}^{(t)}_{\texttt{train}}, \tilde{\mathcal{D}}^{(t)}_{\texttt{test}})\}_{t=1}^{\tilde{T}}$



Jürgen Schmidhuber (1987)

Bilevel optimization framework

- Bilevel formulation for HPO and Meta Learning

$$\min\{f(\lambda) : \lambda \in \Lambda\}, \tag{1}$$

where function $f : \Lambda \to \mathbb{R}$ is defined at $\lambda \in \Lambda$ as

$$f(\lambda) = \inf\{E(w_\lambda, \lambda) : w_\lambda \in \arg\min_{u \in \mathbb{R}^d} L_\lambda(u)\}. \tag{2}$$

model $g_w : \mathcal{X} \to \mathcal{Y}$

inner objective $L_\lambda(w) = \sum_{(x,y) \in D_{\text{tr}}} \ell(g_w(x), y) + \Omega_\lambda(w)$

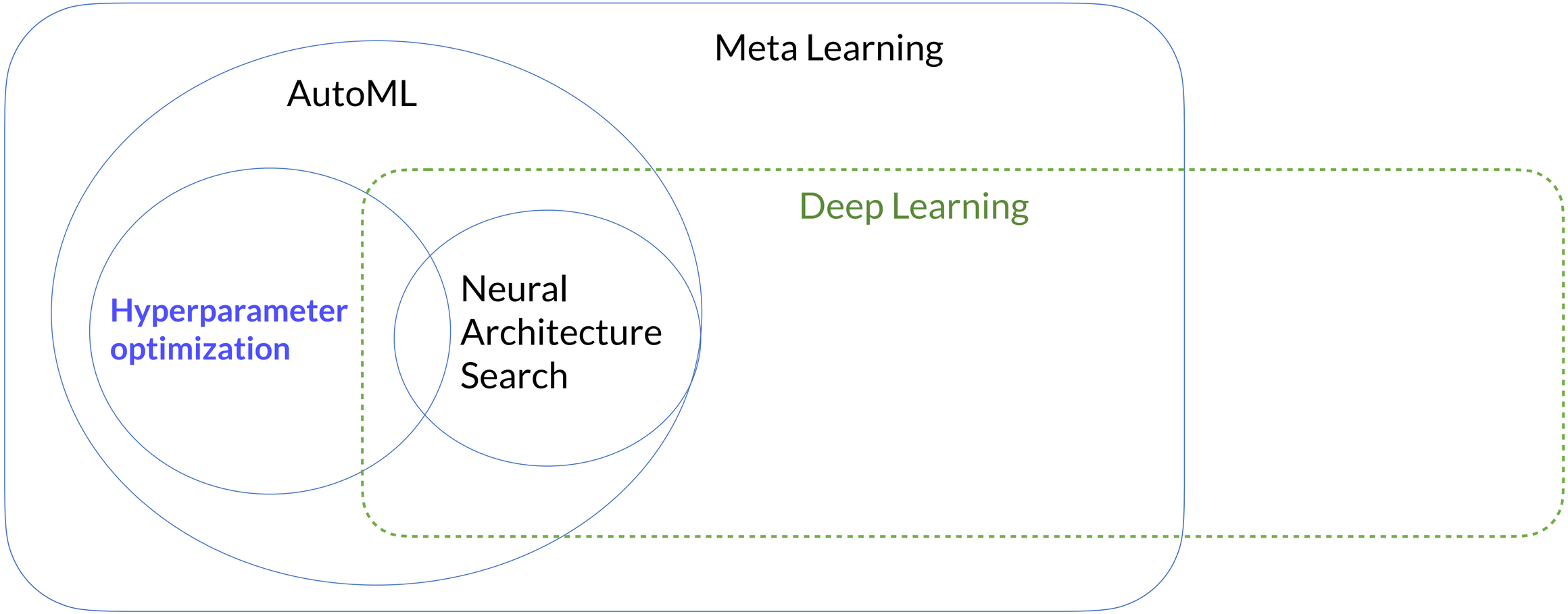
outer objective $E(w, \lambda) = \sum_{(x,y) \in D_{\text{val}}} \ell(g_w(x), y).$

Bilevel programming	Hyperparameter optimization	Meta-learning
Inner variables	Parameters	Parameters of Ground models
Outer variables	Hyperparameters	Parameters of Meta-learner
Inner objective	Training error	Training errors on tasks (Eq. 3)
Outer objective	Validation error	Meta-training error (Eq. 4)

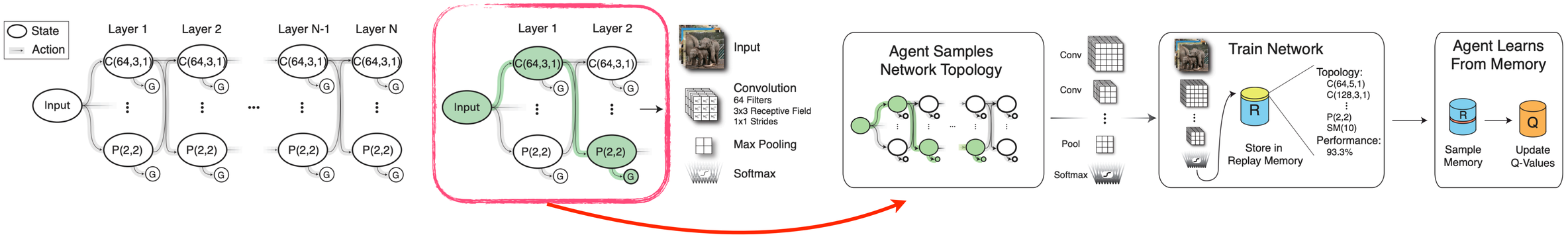
Bilevel Programming for Hyperparameter Optimization and Meta-Learning, Franceschi et al., ***ICML*** 2018

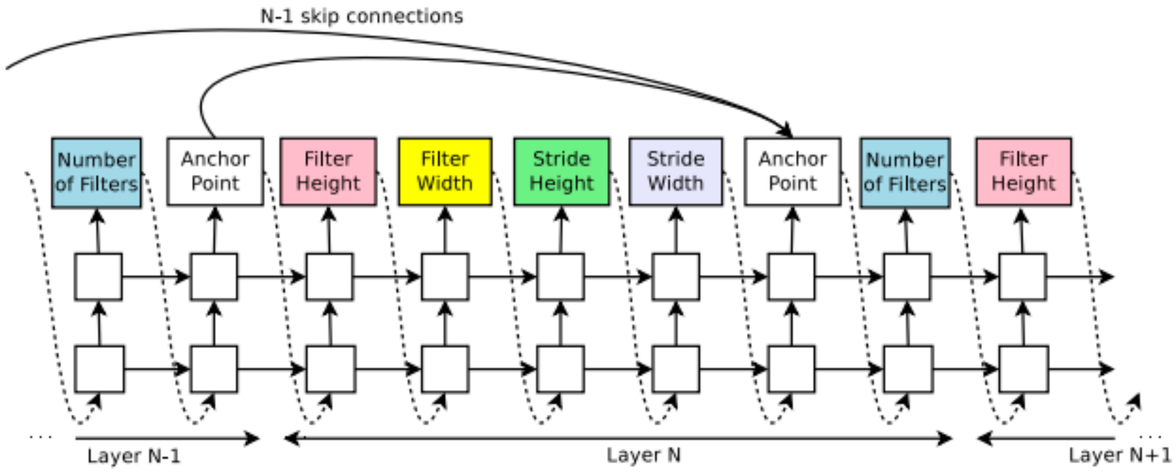
Copyright © 2022 by UNIST & LIM Lab

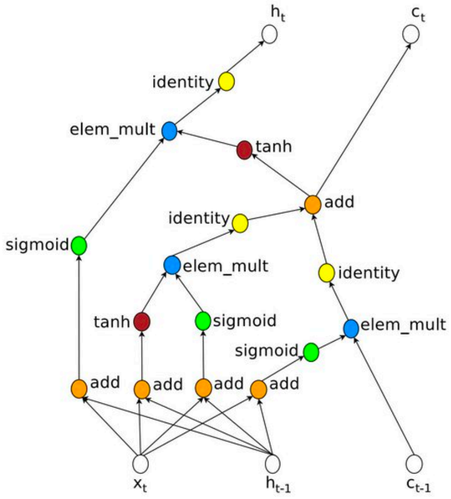
Vénn Diagram for AutoML

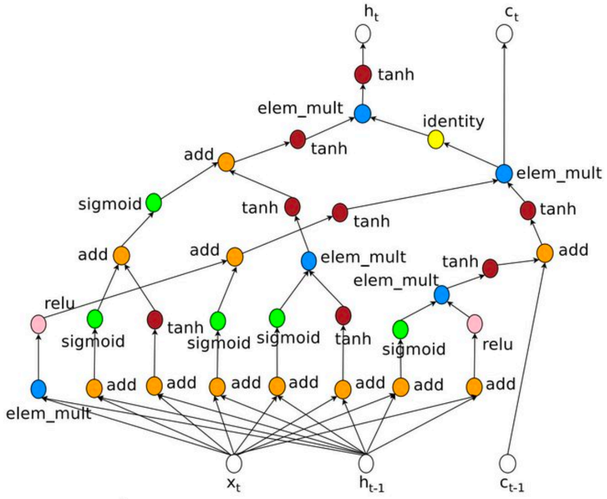


Neural Architecture Search









Search Space

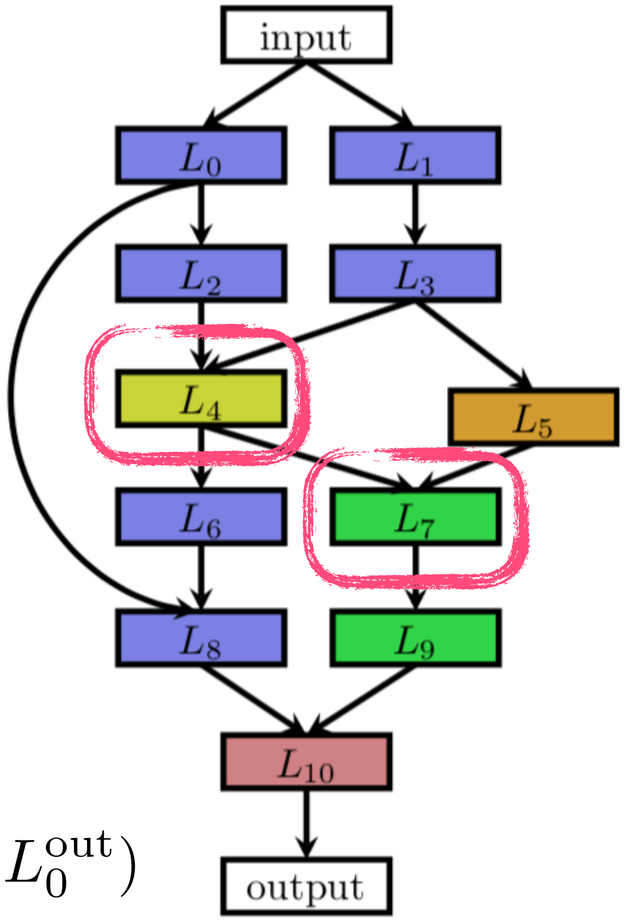
- Chain-structured Neural Nets:
  - number of layers
  - type of operation
    - ☑ pooling
    - ☑ convolution
  - hyperparameters of the operation
    - ☑ number of filters
    - ☑ kernel size
    - ☑ stride

$$\mathrm{A} = L_n \circ \cdots L_1 \circ L_0$$

Parametrization of search space is **not** fixed-length! → **RNN**

Search Space

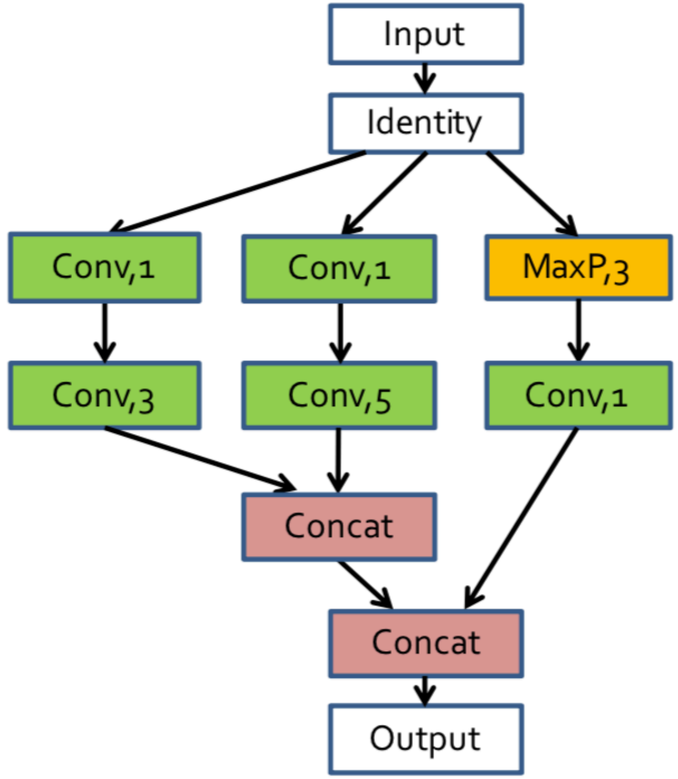
- Multi-branch Neural Nets
  - input of layer $: i \quad g_i(L_{i-1}^{\text{out}}, \ldots, L_0^{\text{out}})$
  - more degrees of freedom
  - special cases:
    - ☑ Chain-structured: $g_i(L_{i-1}^{\text{out}}, \ldots, L_0^{\text{out}}) = L_{i-1}^{\text{out}}$
    - ☑ ResNets: $g_i(L_{i-1}^{\text{out}}, \ldots, L_0^{\text{out}}) = L_{i-1}^{\text{out}} + L_j^{\text{out}}$
    - ☑ DenseNets: $g_i(L_{i-1}^{\text{out}}, \ldots, L_0^{\text{out}}) = \text{concat}(L_{i-1}^{\text{out}}, \ldots, L_0^{\text{out}})$



Network Structure Code

- Layer Representation of DAG
  - better initialization for search
  - generate optimal block structure with a quick training process

Name	Index	Type	Kernel Size	Pred1	Pred2
Convolution	T	1	1, 3, 5	K	0
Max Pooling	T	2	1, 3	K	0
Average Pooling	T	3	1, 3	K	0
Identity	T	4	0	K	0
Elemental Add	T	5	0	K	K
Concat	T	6	0	K	K
Terminal	T	7	0	0	0

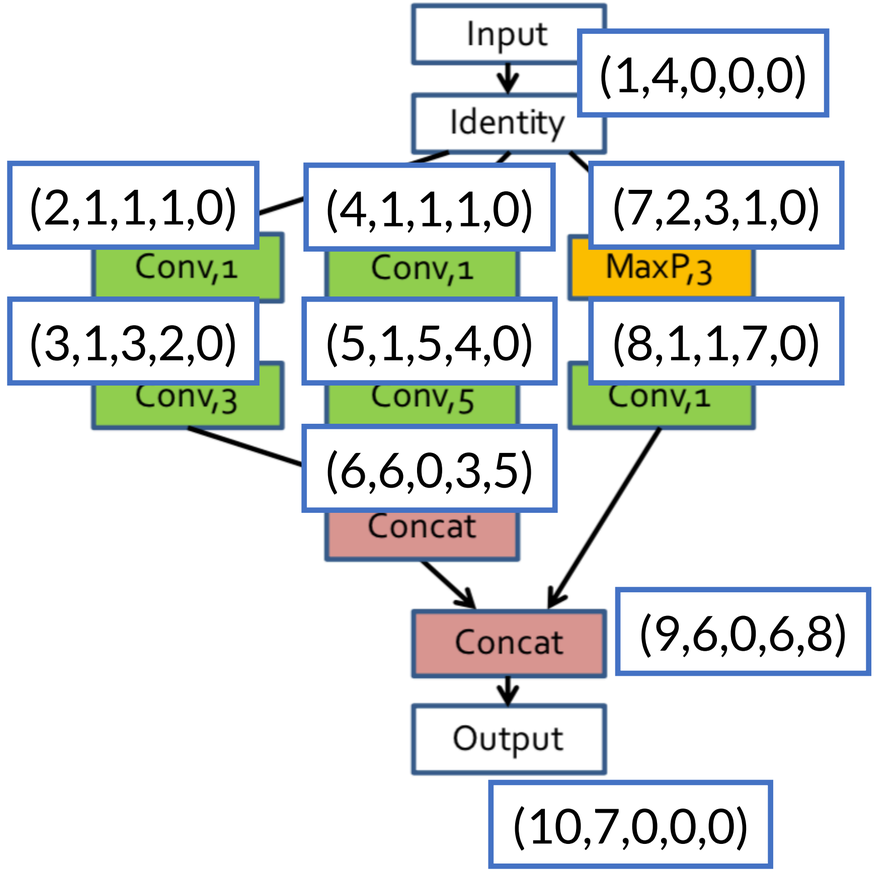


BlockQNN: Efficient Block-wise Neural Network Architecture Generation, Zhong et al., ***CVPR*** 2018

Network Structure Code

- Layer Representation of DAG
  - better initialization for search
  - generate optimal block structure with a quick training process

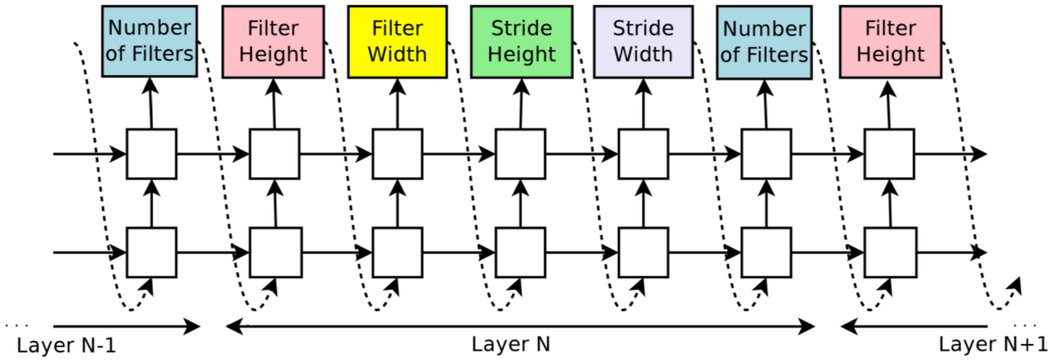
Name	Index	Type	Kernel Size	Pred1	Pred2
Convolution	**T**	1	1, 3, 5	**K**	0
Max Pooling	**T**	2	1, 3	**K**	0
Average Pooling	**T**	3	1, 3	**K**	0
Identity	**T**	4	0	**K**	0
Elemental Add	**T**	5	0	**K**	**K**
Concat	**T**	6	0	**K**	**K**
Terminal	**T**	7	0	0	0



BlockQNN: Efficient Block-wise Neural Network Architecture Generation, Zhong et al., ***CVPR*** 2018

RNN Controller for Search

- Controller generates architecture
- **Build** architecture → **train** network
- Maximizes validation accuracy



$$J(\theta_c) = \mathbb{E}_{P_{\theta_c}(a_{1:T})}[R]$$

parameter of the controller

generated architectures

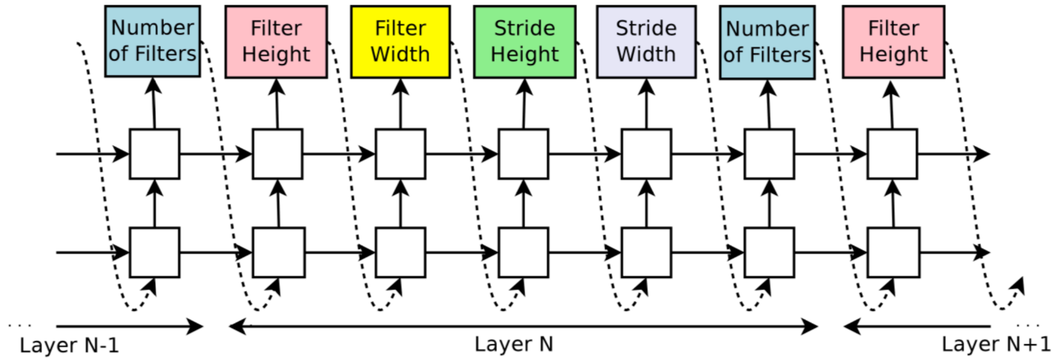
validation accuracy

NAS: Neural Architecture Search with Reinforcement Learning, Zoph et al., ***ICLR*** 2018

Copyright © 2022 by UNIST & LIM Lab

RNN Controller for Search

- Controller generates architecture
- **Build** architecture → **train** network
- Maximizes validation accuracy



update

$$J(\theta_c) = \mathbb{E}_{P_{\theta_c}(a_{1:T})}[R]$$

$$\nabla_{\theta_c} J(\theta_c) = \sum_{t=1}^{T} \mathbb{E}_{P_{\theta_c}(a_{1:T})}[\nabla_{\theta_c} \log P_{\theta_c}(a_t|a_{t-1})R]$$

parameter of the controller

generated architectures

validation accuracy

non-differentiable

REINFORCE

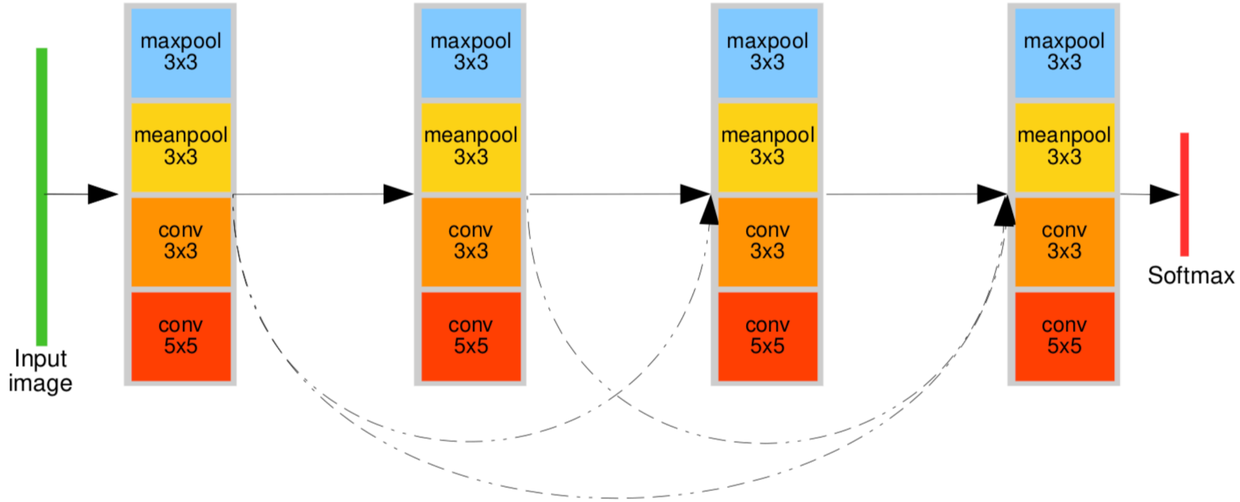
you will learn how to do this in RL coursework!

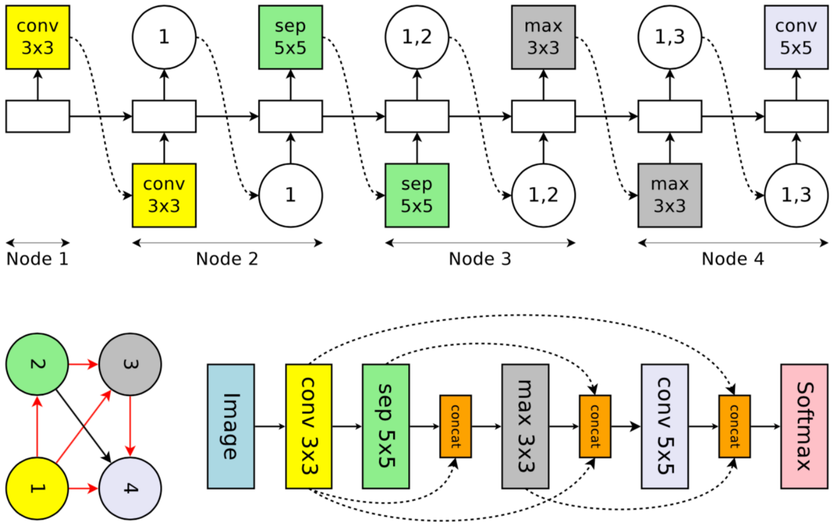
NAS: Neural Architecture Search with Reinforcement Learning, Zoph et al., ***ICLR*** 2018

Copyright © 2022 by UNIST & LIM Lab

Marco Search

- RNN Controller designs the entire convolutional network

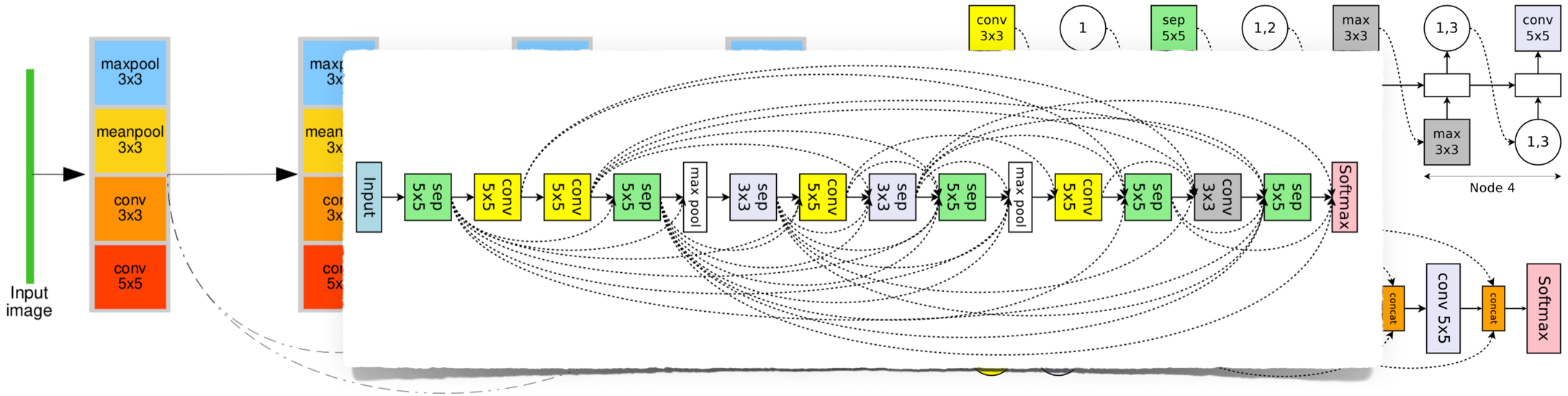




ENAS: Efficient Neural Architecture Search via Parameter Sharing, Pham et al., ***ICML*** 2018

Marco Search

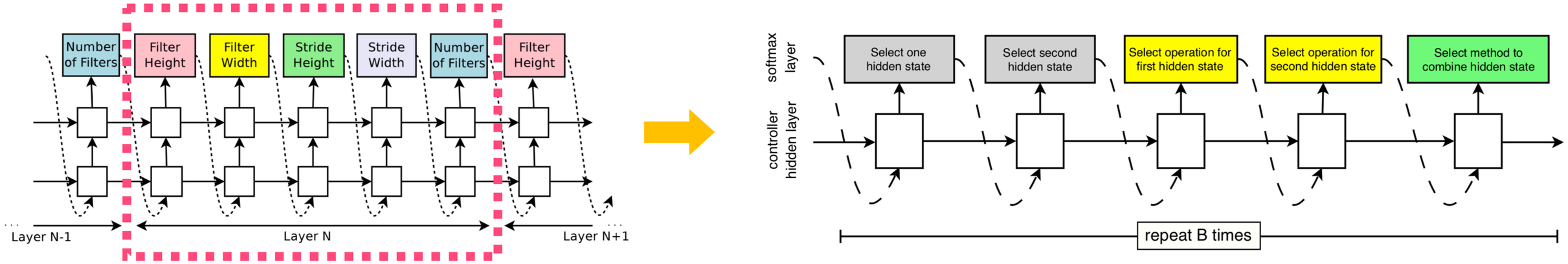
- RNN Controller designs the entire convolutional network



ENAS: Efficient Neural Architecture Search via Parameter Sharing, Pham et al., ***ICML*** 2018

Micro Search

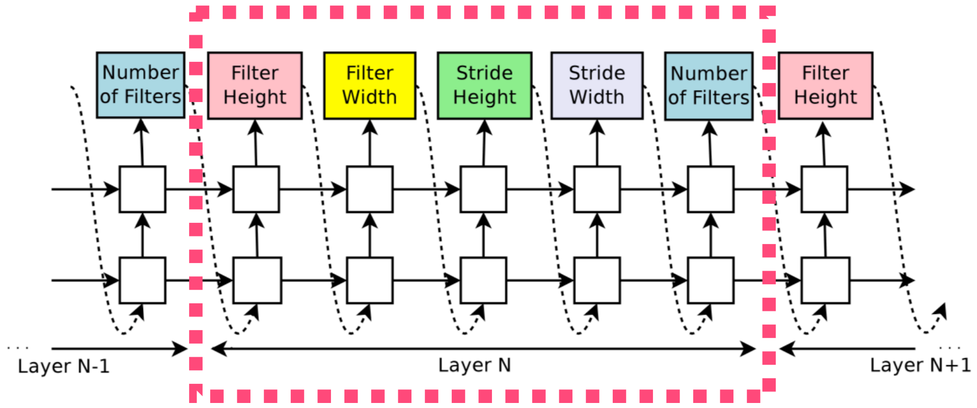
- Repeated **motifs**: cell is stacked in series
- Each cell has the same architecture but different weights

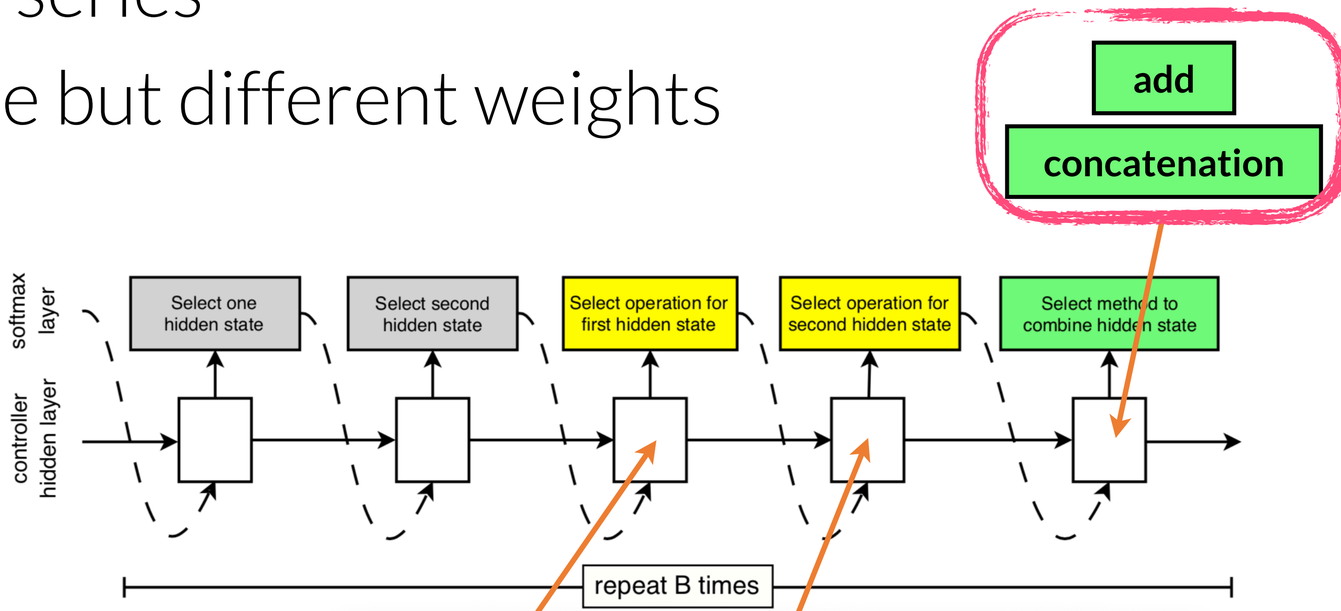


Learning Transferable Architectures for Scalable Image Recognition, Zoph et al., ***CVPR*** 2018

Micro Search

- Repeated **motifs**: cell is stacked in series
- Each cell has the same architecture but different weights





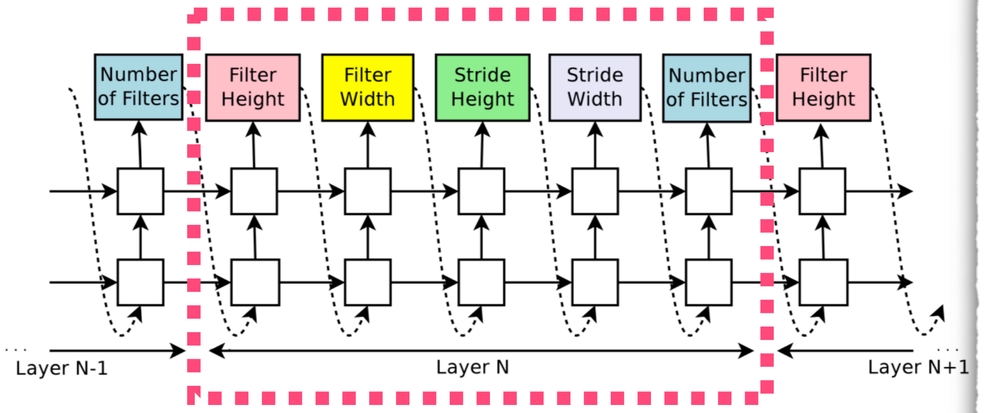
fixed set of operations

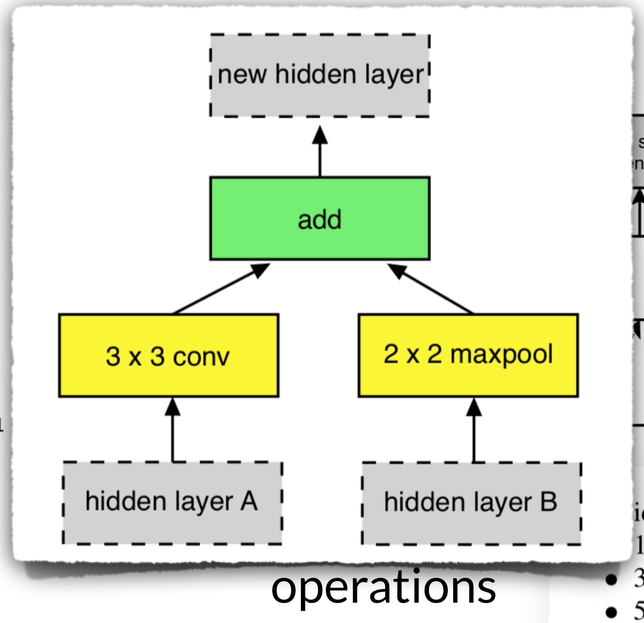
- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-seperable conv

Learning Transferable Architectures for Scalable Image Recognition, Zoph e

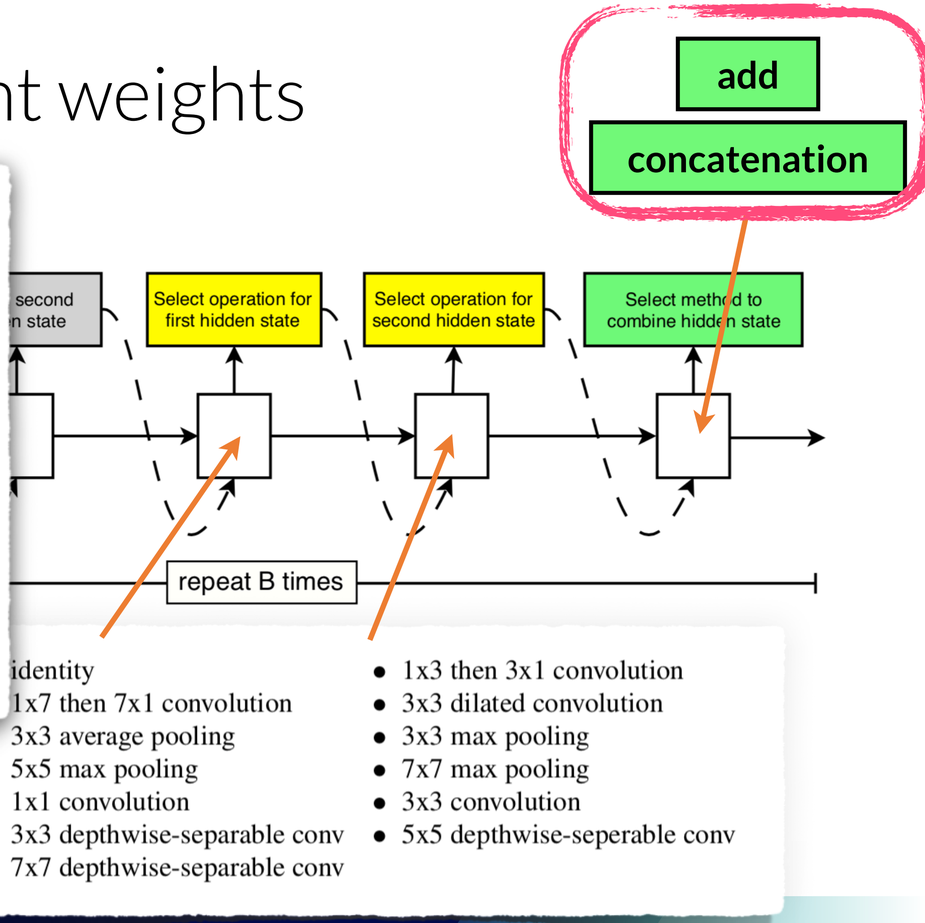
Micro Search

- Repeated **motifs**: cell is stacked in series
- Each cell has the same architecture but different weights





operations

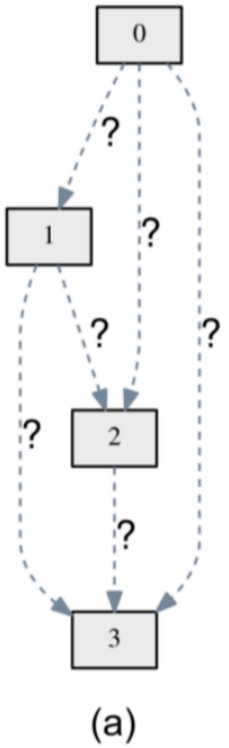


Learning Transferable Architectures for Scalable Image Recognition, Zoph e

Copyright © 2022 by UNIST & LIM Lab

Continuous Relaxation of Search Space

- Cell-based search space: $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$



(a)

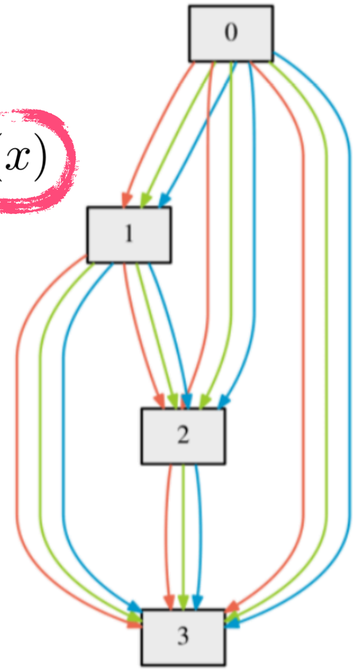
DARTS: Differentiable Architecture Search, Liu et al., ***ICLR*** 2019

Continuous Relaxation of Search Space

- Cell-based search space:
  - convex combination

$$\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$$

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$



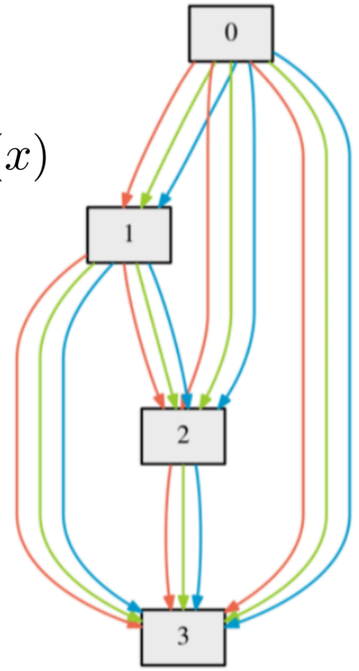
(b)

DARTS: Differentiable Architecture Search, Liu et al., ***ICLR*** 2019

Continuous Relaxation of Search Space

- Cell-based search space: $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$
  - convex combination
  - differentiable parameter

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$



(b)

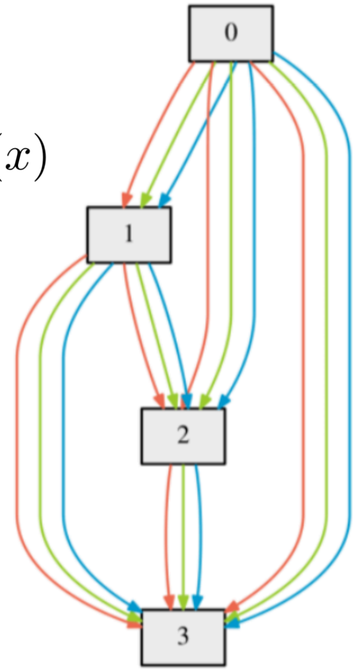
DARTS: Differentiable Architecture Search, Liu et al., ***ICLR*** 2019

Continuous Relaxation of Search Space

- Cell-based search space: $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$
  - convex combination
  - differentiable parameter
- Bilevel optimization

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ \text{s.t.} \quad & w^*(\alpha) = \text{argmin}_w \ \mathcal{L}_{train}(w, \alpha) \end{aligned}$$



(b)

DARTS: Differentiable Architecture Search, Liu et al., ***ICLR*** 2019

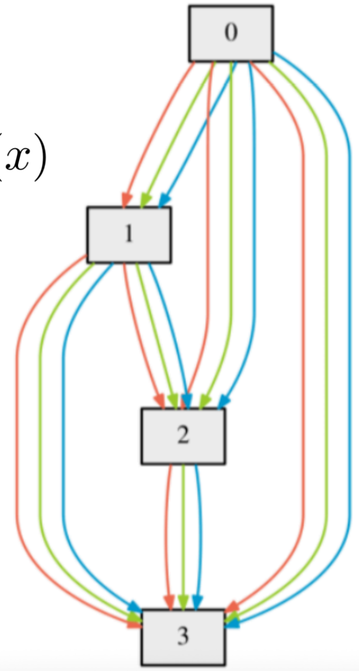
Continuous Relaxation of Search Space

- Cell-based search space: $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$
  - convex combination
  - differentiable parameter
- Bilevel optimization

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

$$\min_{\alpha} \quad \mathcal{L}_{val}(w^*(\alpha), \alpha)$$

$$\text{s.t.} \quad w^*(\alpha) = \operatorname{argmin}_w \ \mathcal{L}_{train}(w, \alpha)$$



Algorithm 1: DARTS – Differentiable Architecture Search

Create a mixed operation $\bar{o}^{(i,j)}$ parametrized by $\alpha^{(i,j)}$ for each edge (i, j)

while *not converged* **do**

1. Update weights w by descending $\nabla_w \mathcal{L}_{train}(w, \alpha)$
2. Update architecture α by descending $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$

Replace $\bar{o}^{(i,j)}$ with $o^{(i,j)} = \operatorname{argmax}_{o \in \mathcal{O}} \alpha_o^{(i,j)}$ for each edge (i, j)

DARTS: Differentiable Architecture Search, Liu et al.,

Copyright © 2022 by UNIST & LIM Lab

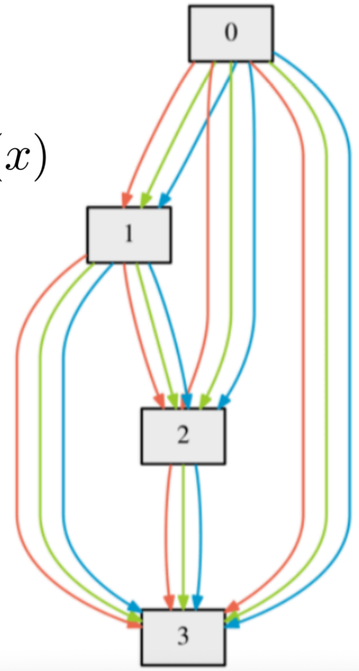
Continuous Relaxation of Search Space

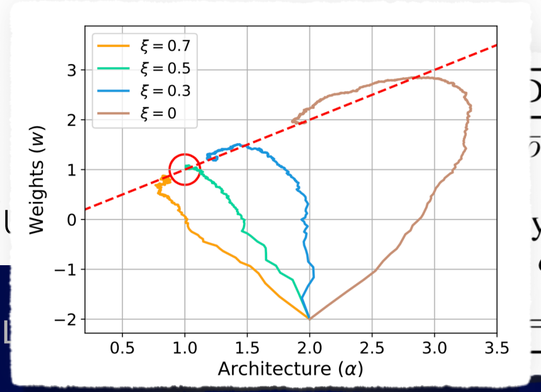
- Cell-based search space: $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$
  - convex combination
  - differentiable parameter
- Bilevel optimization

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

$$\min_{\alpha} \quad \mathcal{L}_{val}(w^*(\alpha), \alpha)$$

$$\text{s.t.} \quad w^*(\alpha) = \text{argmin}_w \; \mathcal{L}_{train}(w, \alpha)$$





Differentiable Architecture Search

$\bar{o}^{(i,j)}$ parametrized by $\alpha^{(i,j)}$ for each edge (i, j)

y descending $\nabla_w \mathcal{L}_{train}(w, \alpha)$

α by descending $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$

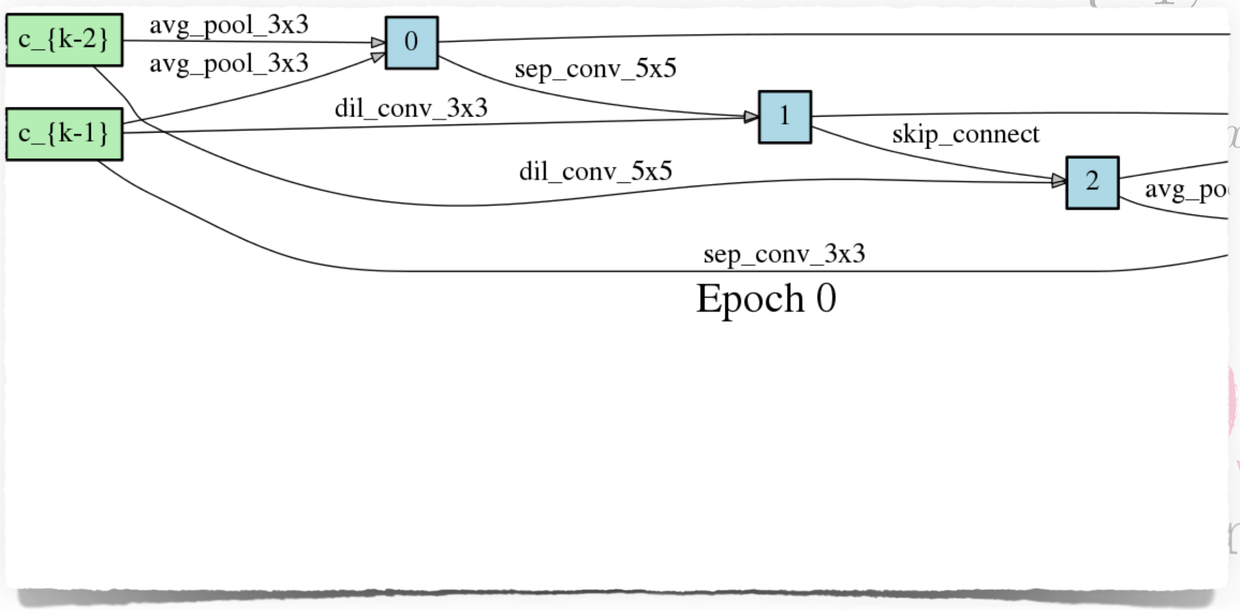
$= \text{argmax}_{o \in \mathcal{O}} \alpha_o^{(i,j)}$ for each edge (i, j)

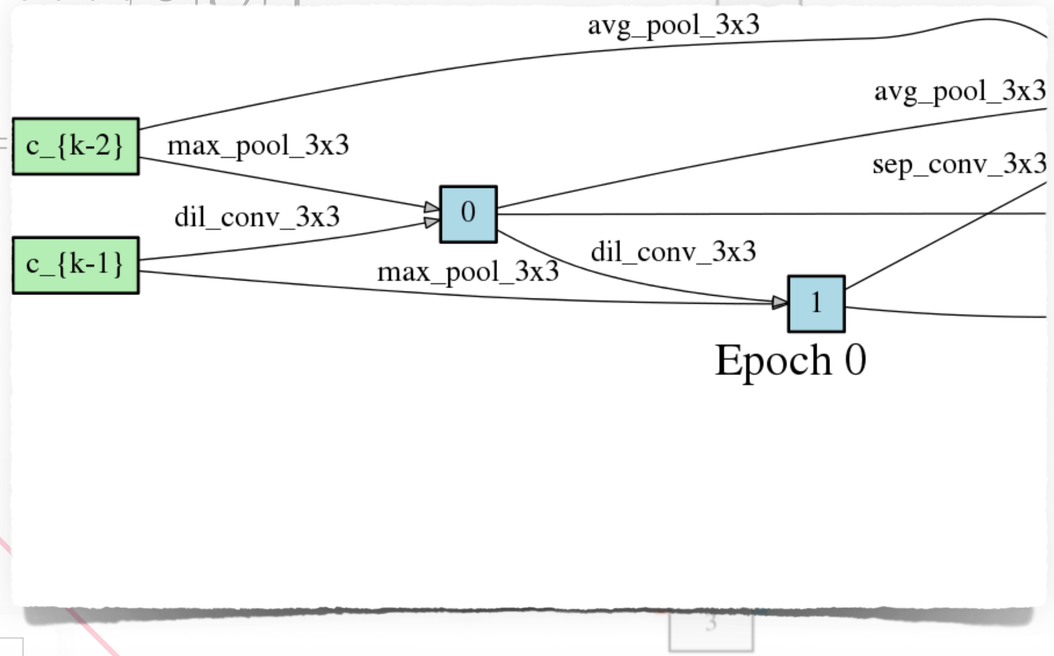
DARTS: Differentiable Architecture Search, Liu

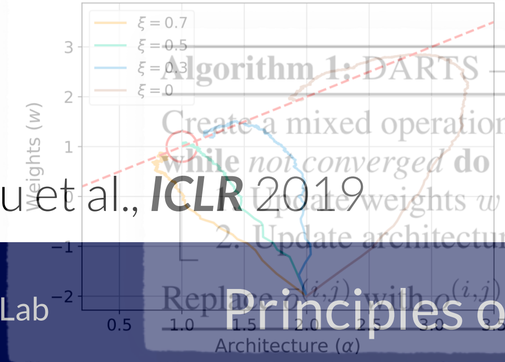
Copyright © 2022 by UNIST & LIM L

Continuous Relaxation of Search Space

- Cell-based search space: $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$
-







Algorithm 1: DARTS – Differentiable Architecture Search

Create a mixed operation $\bar{o}^{(i,j)}$ parametrized by $\alpha^{(i,j)}$ for each edge (i, j)

while *not converged* **do**

1. Update weights w by descending $\nabla_w \mathcal{L}_{train}(w, \alpha)$
2. Update architecture α by descending $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$

Replace each $o^{(i,j)}$ with $o^{(i,j)} = \text{argmax}_{o \in \mathcal{O}} \alpha_o^{(i,j)}$ for each edge (i, j)

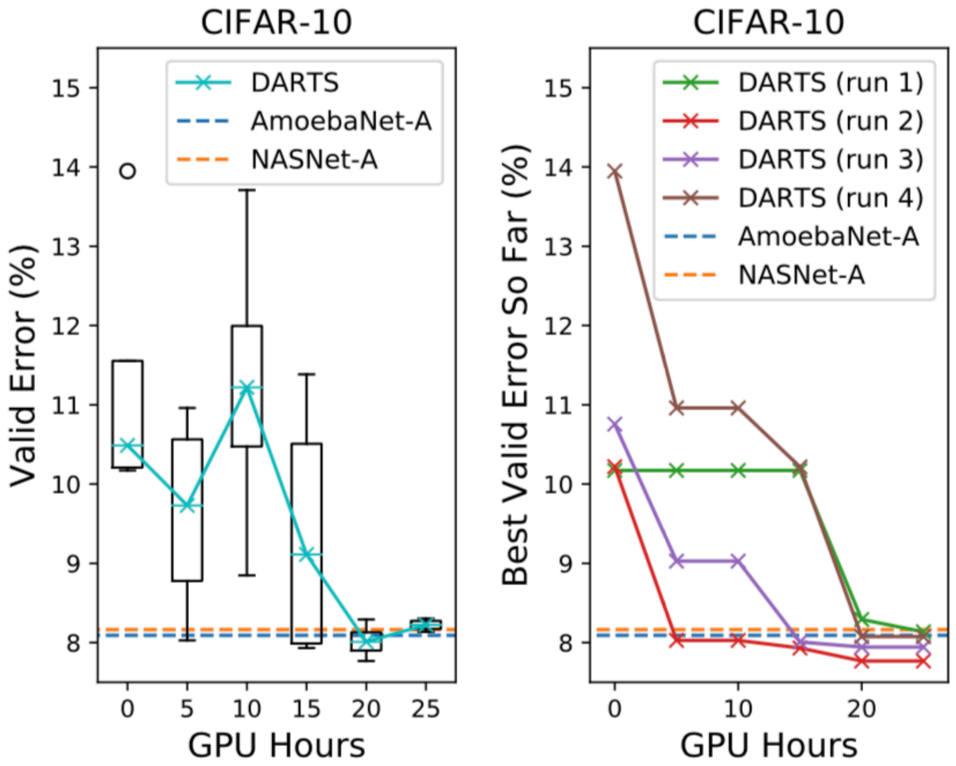
DARTS: Differentiable Architecture Search, Liu et al., ***ICLR*** 2019

DARTS: Experiment

Table 1: Comparison with state-of-the-art image classifiers on CIFAR-10 (lower error rate is better). Note the search cost for DARTS does not include the selection cost (1 GPU day) or the final evaluation cost by training the selected architecture from scratch (1.5 GPU days).

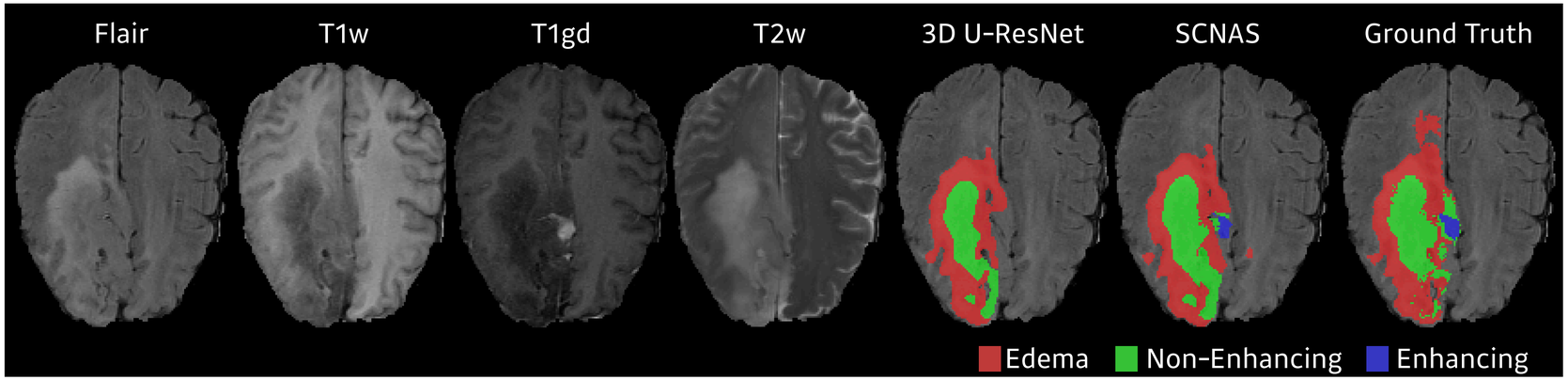
Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	#ops	Search Method
DenseNet-BC (Huang et al., 2017)	3.46	25.6	–	–	manual
NASNet-A + cutout (Zoph et al., 2018)	2.65	3.3	2000	13	RL
NASNet-A + cutout (Zoph et al., 2018)†	2.83	3.1	2000	13	RL
BlockQNN (Zhong et al., 2018)	3.54	39.8	96	8	RL
AmoebaNet-A (Real et al., 2018)	3.34 ± 0.06	3.2	3150	19	evolution
AmoebaNet-A + cutout (Real et al., 2018)†	3.12	3.1	3150	19	evolution
AmoebaNet-B + cutout (Real et al., 2018)	2.55 ± 0.05	2.8	3150	19	evolution
Hierarchical evolution (Liu et al., 2018b)	3.75 ± 0.12	15.7	300	6	evolution
PNAS (Liu et al., 2018a)	3.41 ± 0.09	3.2	225	8	SMBO
ENAS + cutout (Pham et al., 2018b)	2.89	4.6	0.5	6	RL
ENAS + cutout (Pham et al., 2018b)*	2.91	4.2	4	6	RL
Random search baseline‡ + cutout	3.29 ± 0.15	3.2	4	7	random
DARTS (first order) + cutout	3.00 ± 0.14	3.3	1.5	7	gradient-based
DARTS (second order) + cutout	2.76 ± 0.09	3.3	4	7	gradient-based

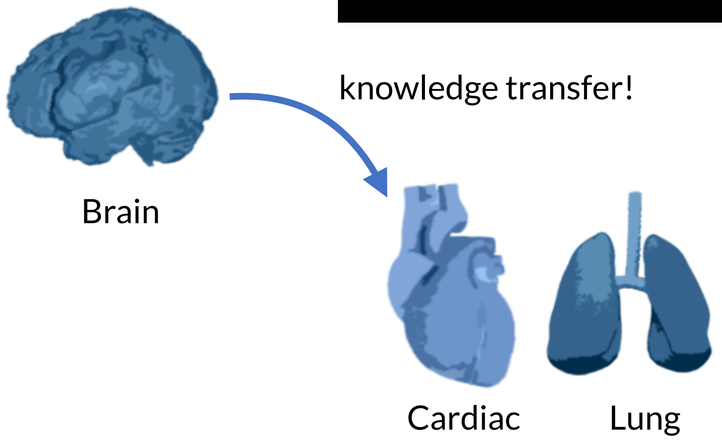
* Obtained by repeating ENAS for 8 times using the code publicly released by the authors. The cell for final evaluation is chosen according to the same selection protocol as for DARTS.
† Obtained by training the corresponding architectures using our setup.
‡ Best architecture among 24 samples according to the validation error after 100 training epochs.



DARTS: Differentiable Architecture Search, Liu et al., ICLR 2019

NAS for 3D Medical Images





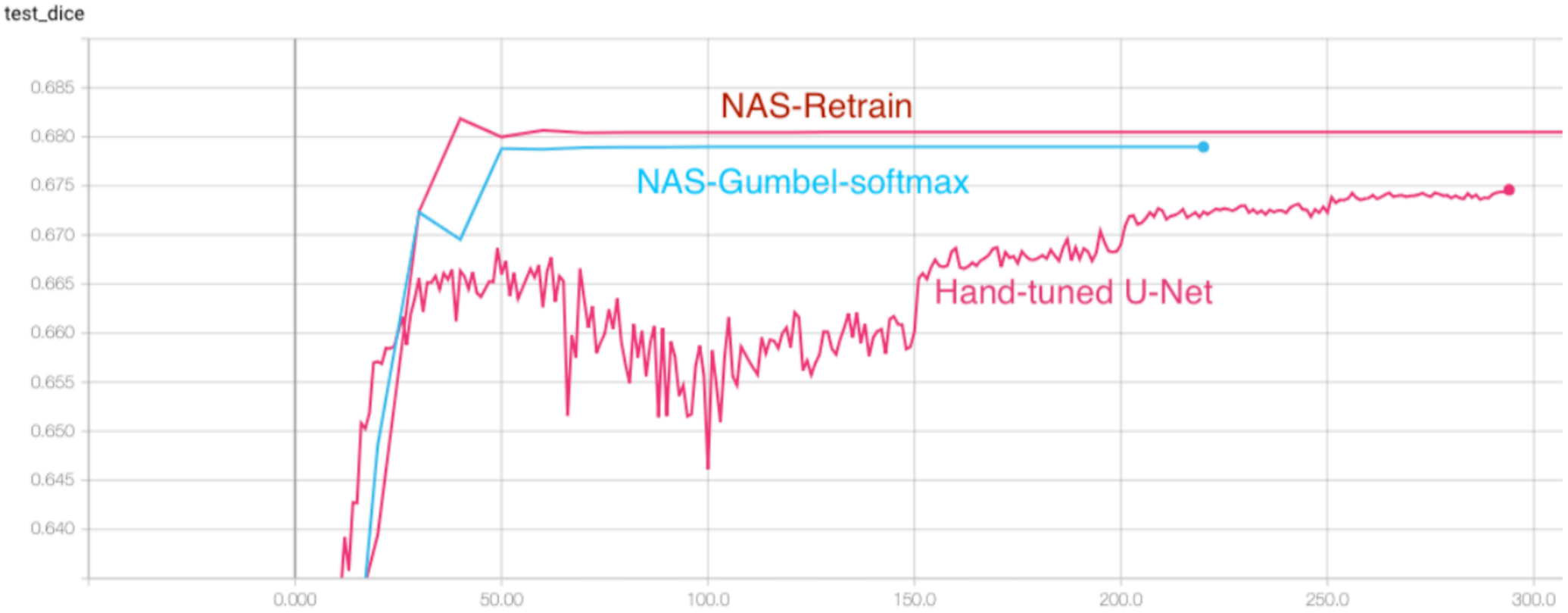
	Brain Tumor (MRI)				Heart (MRI)	Lung (CT)
Label	Edema	Non-Enhancing	Enhancing	Average	Left Atrium	Tumor
3D nnU-Net [2]	80.71	62.22	79.07	74.00	92.45	55.87
3D U-ResNet	70.74	56.69	73.23	66.89	91.48	63.28
SCNAS	80.41	59.85	78.50	72.92	91.29	64.82
SCNAS(transfer)	-	-	-	-	91.91	68.62

Scalable Neural Architecture Search for 3D Medical Image Segmentation, ***MICCAI*** (2019)

Joint work with S. Kim (Kakao Brain), I. Kim (Kakao Brain), W. Baek (Kakao Brain), C. Kim (Kakao Brain), H. Cho (SNU), B. Yoon (Kakao Brain), T. Kim (MILA)

Copyright © 2022 by UNIST & LIM Lab

NAS for 3D Medical Images



Brain

(CT)
ıor
37
28
32
62

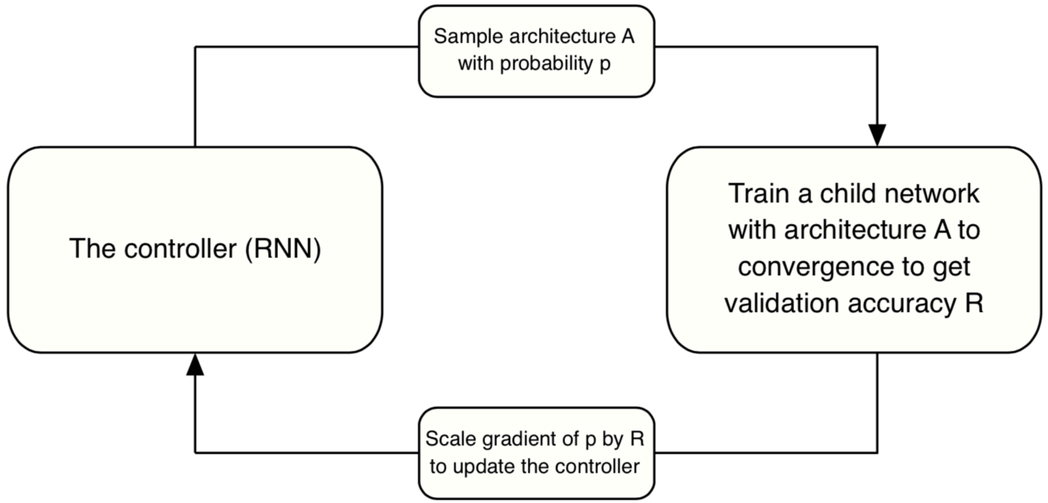
Scalable Neural Architecture Search for 3D Medical Image Segmentation, ***MICCAI*** (2019)

Joint work with S. Kim (Kakao Brain), I. Kim (Kakao Brain), W. Baek (Kakao Brain), C. Kim (Kakao Brain), H. Cho (SNU), B. Yoon (Kakao Brain), T. Kim (MILA)

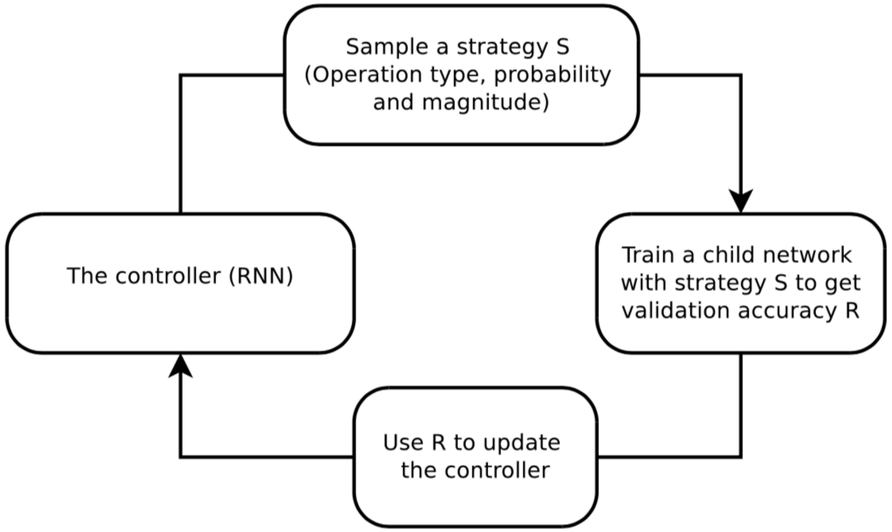
Copyright © 2022 by UNIST & LIM Lab

AutoAugment for Insufficient Data

- NAS = automated **architecture** search
- AutoAugment = automated **augmentation** search



NAS



AutoAugment

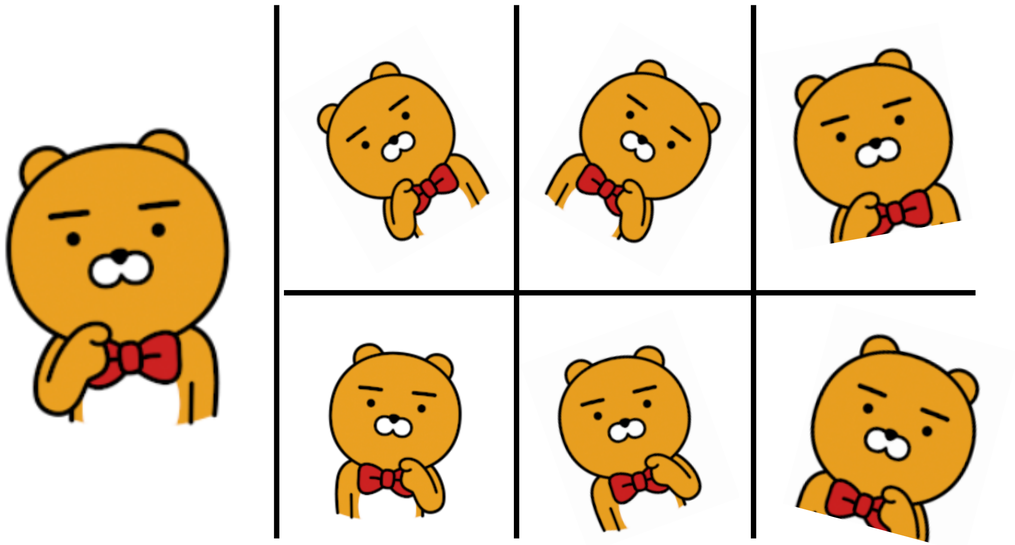
Which augmentation is better?

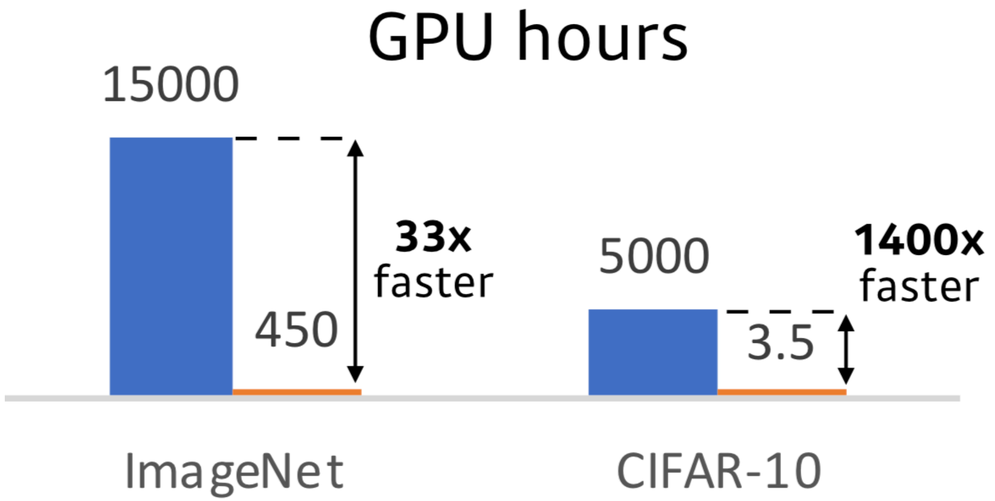
	Original	Sub-policy 1	Sub-policy 2	Sub-policy 3	Sub-policy 4	Sub-policy 5
Batch 1						
Batch 2						
Batch 3						
		Equalize, 0.4, 4 Rotate, 0.8, 8	Solarize, 0.6, 3 Equalize, 0.6, 7	Posterize, 0.8, 5 Equalize, 1.0, 2	Rotate, 0.2, 3 Solarize, 0.6, 8	Equalize, 0.6, 8 Posterize, 0.4, 6

Copyright © 2022 by UNIST & LIM Lab

Fast AutoAugment

- Pre-training is important
- Random augmentation shows poor performance



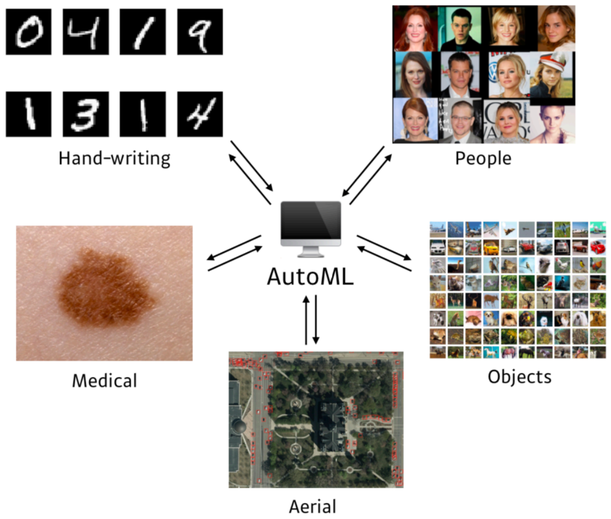


Fast AutoAugment, ***NeurIPS*** (2019)

Joint work with I. Kim (Kakao Brain), C. Kim (Kakao Brain), T. Kim (MILA), S. Kim (Kakao Brain)

NeurIPS AutoDL Challenge

- 20min with 1 GPU
  - **Fast AutoAugment**



Username	Rank	Major DL Framework	Strategy	Prize
kakaobrain	1st	PyTorch	Fast AutoAugment	$2000
tanglang	2nd	PyTorch	Inspired by Fast AutoAugment	$1500
kvr	3rd	PyTorch	Inspired by Fast AutoAugment	$500

NIPS 2019 Auto Computer Vision Challenge

Top **1st** Place on AutoCV & AutoCV2

NeurIPS AutoDL challenges

Enter AutoSpeech (deadline Oct 15)

Enter AutoWeakly (deadline October 29)

Congratulations to the **ECML PKDD conf**. AutoCV2 winners:

- **First place:** kakaobrain [**GitHub repo**]
- **Second place:** tanglang [**GitHub repo**]
- **Third place:** kvr [**GitHub repo**]

They will be invited to present at the **discovery challenge workshops of ECML PKDD**.

The **IJCNN conf.** AutoCV winners [**slides**] were:

- **First place:** KakaoBrain [**GitHub repo**]
- **Second place:** DKKimHCLee [**GitHub repo**][**slides**]
- **Third place:** base_1 [**GitHub repo**][**slides**]

Copyright © 2022 by UNIST & LIM Lab

Q&A