Synergizing Variability Modeling with Machine Learning: A Journey of Possibilities

Dr. Maxime Cordy

Interdisciplinary Center on Security and Trust (SnT), University of Luxembourg

It all started with an email…

Leopoldo Motta Teixeira 5 June 2023, 16:30

Invitation as Keynote Speaker at VAMOS 2024, Bern, Feb 7-9

To: Maxime CORDY, Cc: Marianne Huchard, Timo Kehrer Details

Dear Maxime,

On behalf of the committees of the 18th International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS 2024), we are very pleased to invite you to deliver a keynote speech at the event.

We are familiar with your previous work on variability analysis and verification, and would like to know whether you could provide a perspective on the interaction of variability, testing and machine learning, given that you have been working around these topics in the recent years, and they directly relate to the VaMoS topics.

VaMoS 2024 will be held Feb 7-9, in Bern, Switzerland (https://vamos2024.inf.unibe.ch/).

If you accept our invitation, your keynote will be scheduled on either of the conference days (exact day to be confirmed). You are, of course, also invited to attend the other sessions during the week. All your travel expenses will be covered by VaMoS and you will get a free registration to the whole event.

Over the last years, VaMoS has already counted on renowned scientists or entrepreneurs as keynote speakers, such as Yves Bossu, Marsha Chechik, Mathieu Acher, Alfonso Pierantonio, Manuel Wimmer, Herwig Schreiner, Andrzej Wąsowski, Nelly Bencomo, Norbert Siegmund and Matthias Galster.

We hope you will be interested and available for coming to Bern. It will be our great pleasure to have you as one of our keynote speakers.

Best regards,

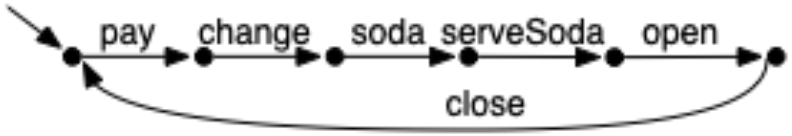
Timo Kehrer, Marianne Huchard, Leopoldo Teixeira
VaMoS 2024 General chair and PC chairs

12 years ago…

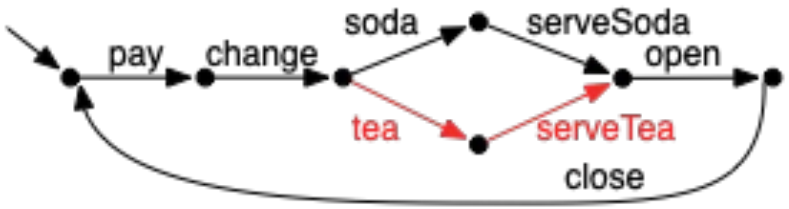
Verification of variability-intensive systems

Instead of:

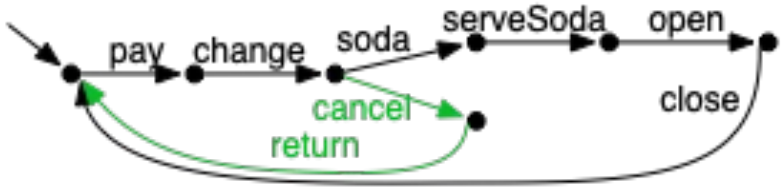
(a) Basic vending machine



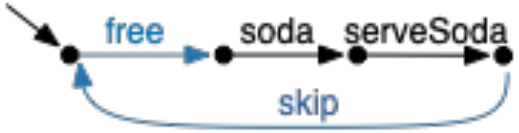
(b) Selling tea and soda



(c) With a cancel purchase function

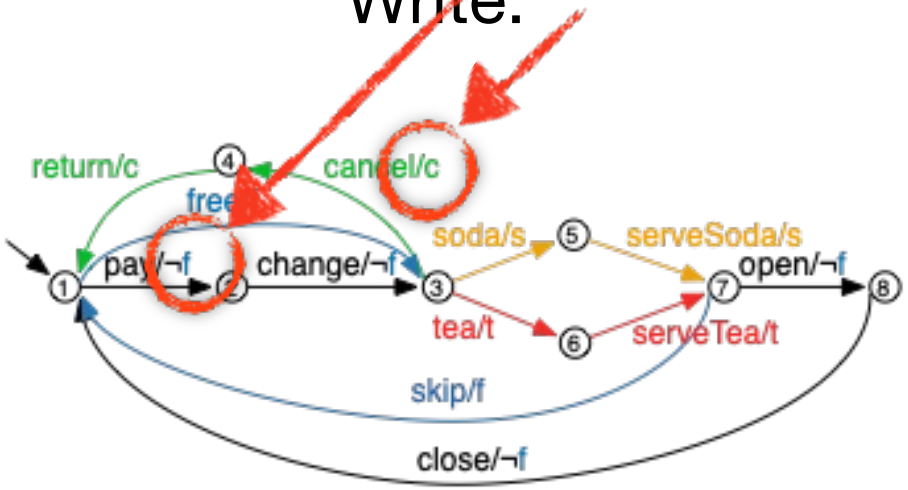


(d) Distributing soda for free

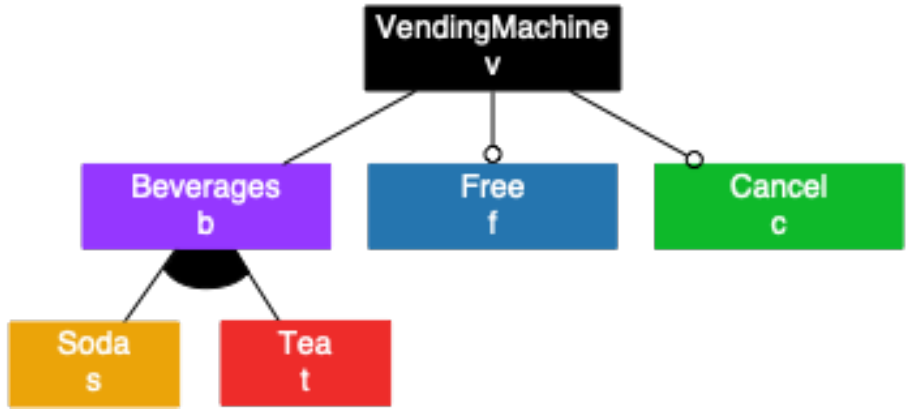


Write:

feature expressions

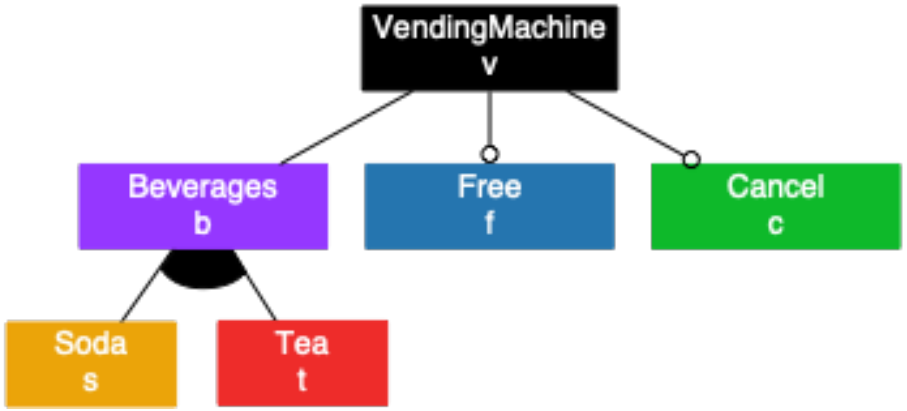


+ companion feature model:



Uniform random sampling

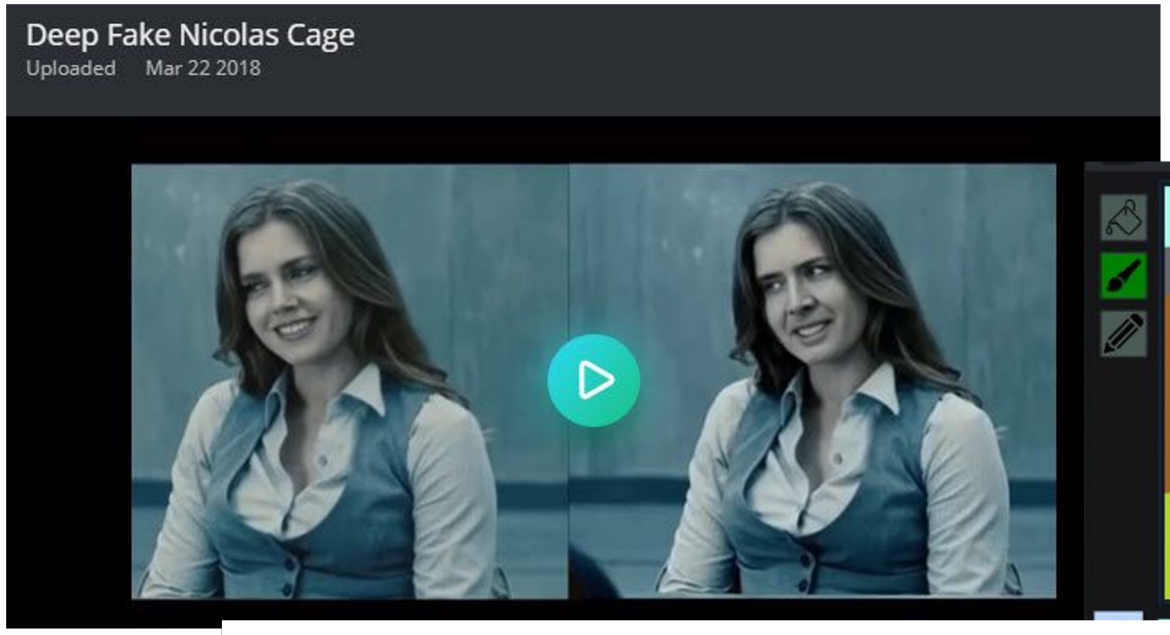
Given a feature model
(a Boolean formula)…

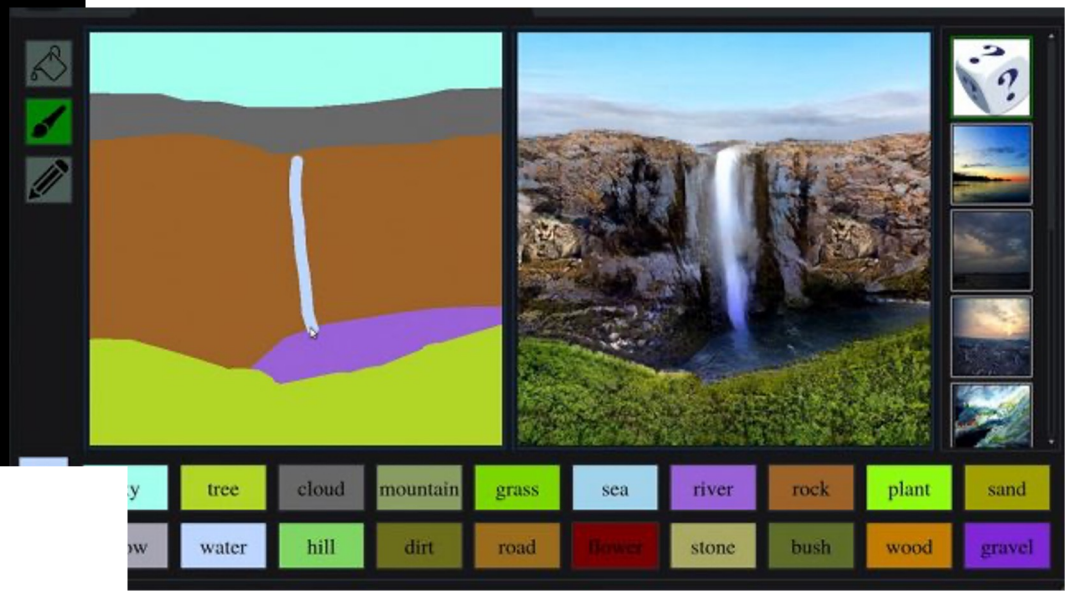


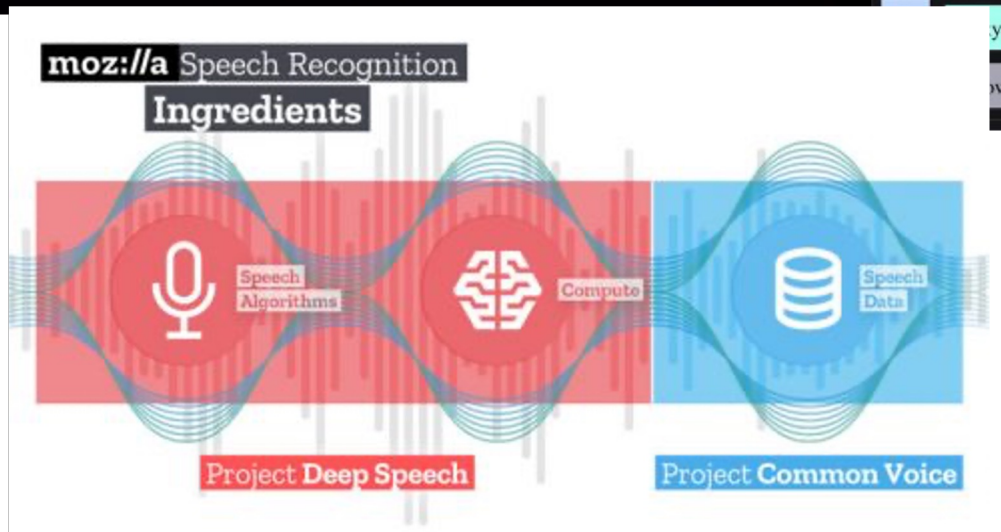
… sample a valid variant (a solution to the formula) with uniform probability:

{v, b, s}	{v, b, t}	{v, b, s, t}
{v, b, s, f}	{v, b, t, f}	{v, b, s, t, f}
{v, b, s, c}	{v, b, t, c}	{v, b, s, t, c}
{v, b, s, f, c}	{v, b, t, f, c}	{v, b, s, t, f, c}

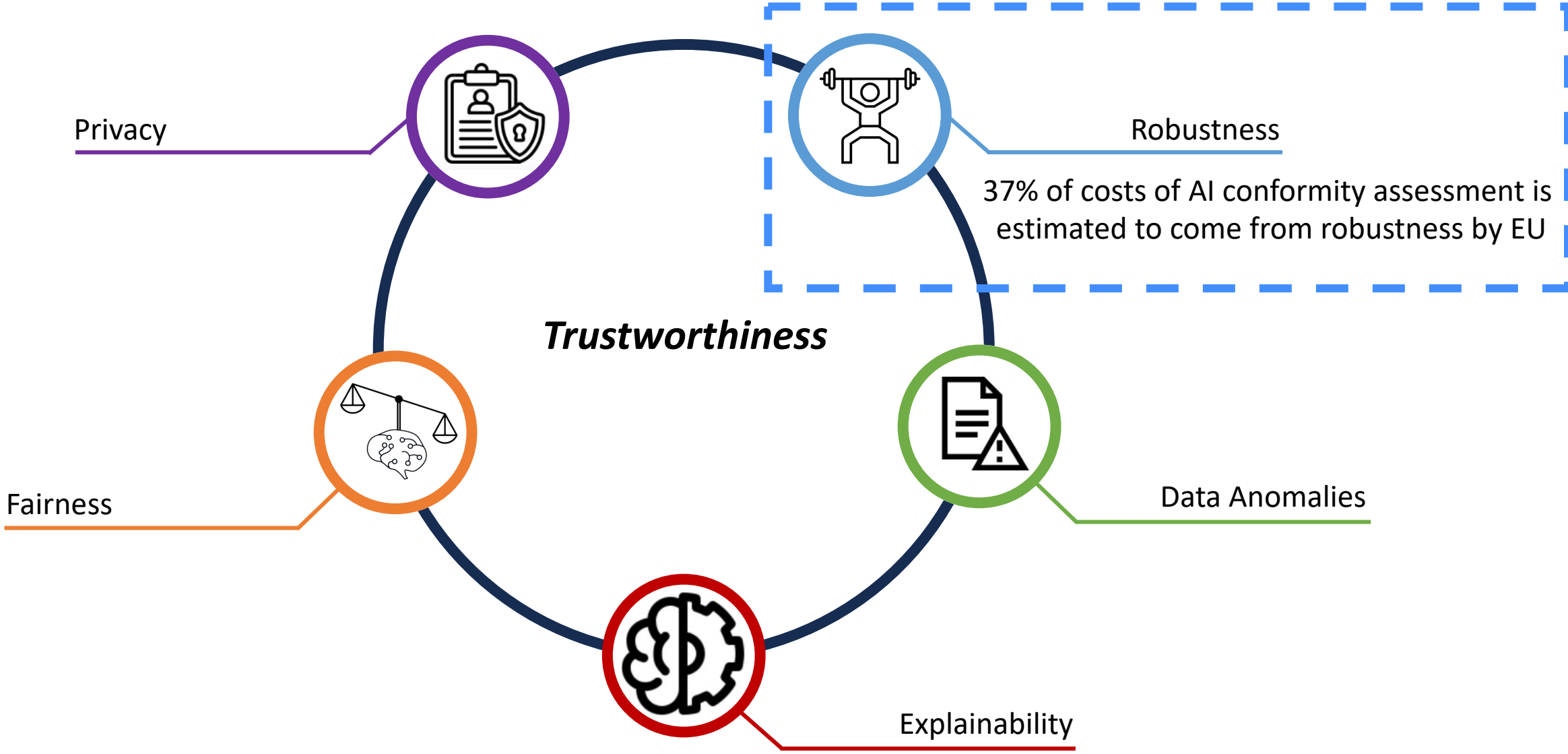
Deep learning: a new world of possibilities







Different aspects of Trustworthy AI



Robustness

"the degree to which a model's performance changes when confronted to data unseen during training"

"Natural Robustness"

Accuracy once running in production

Distribution Drift

"…an evolution of data that invalidates the data model. It happens when the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways."

Security Threats

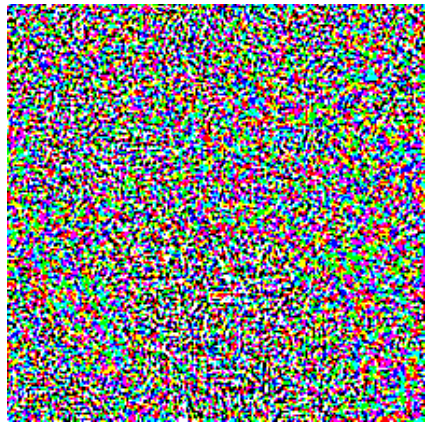
Evasion/adversarial attacks
Inference attacks
Poisoning attacks
Model theft
Etc.

Machine learning robustness (evasion attacks)

Original example

ML predicts:
“Panda”
(80% confidence)

Small adversarial noise



Adversarial example

What humans still see

Gibbon

What ML predicts: “Gibbon”
(99% confidence)

"Explaining and Harnessing Adversarial Examples", Goodfelow et al., ICLR 2015.

Machine learning robustness (evasion attacks)

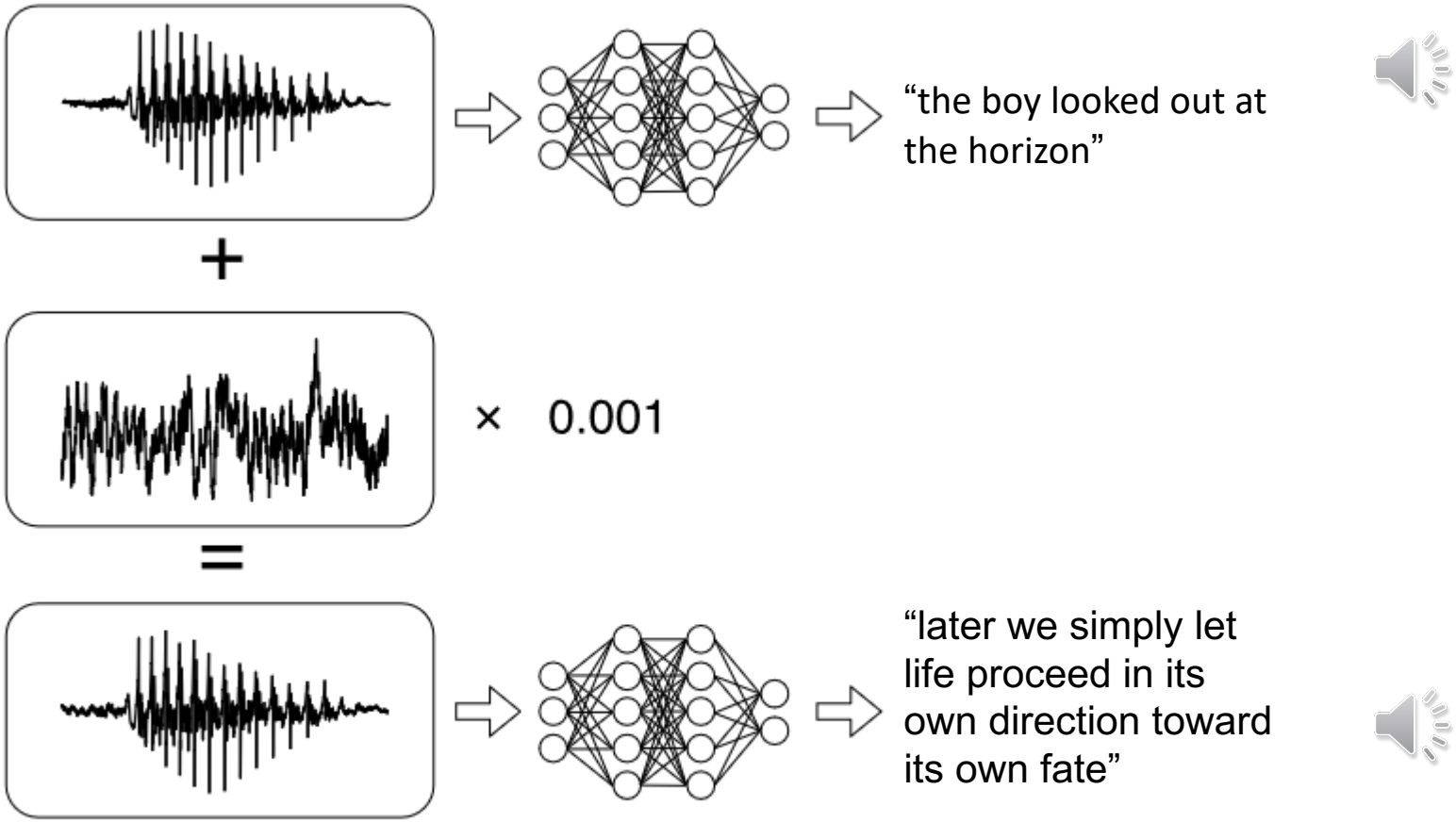
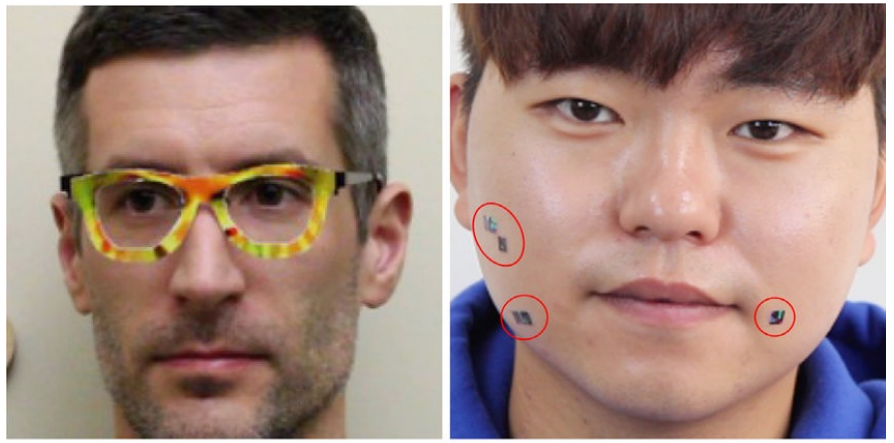


Figure 1. Illustration of our attack: given any waveform, adding a small perturbation makes the result transcribe as any desired target phrase.

Carlini, Nicholas, and David Wagner. "Audio adversarial examples: Targeted attacks on speech-to-text." *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018.

Machine learning robustness (evasion attacks)

Crowd face recognition system



(a) (b)

Journal of Information Security and Applications 60 (2021) 102874

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Adversarial attacks by attaching noise markers on the face against deep face recognition

Gwonsang Ryu [a], Hosung Park [b], Daeseon Choi [c,*]

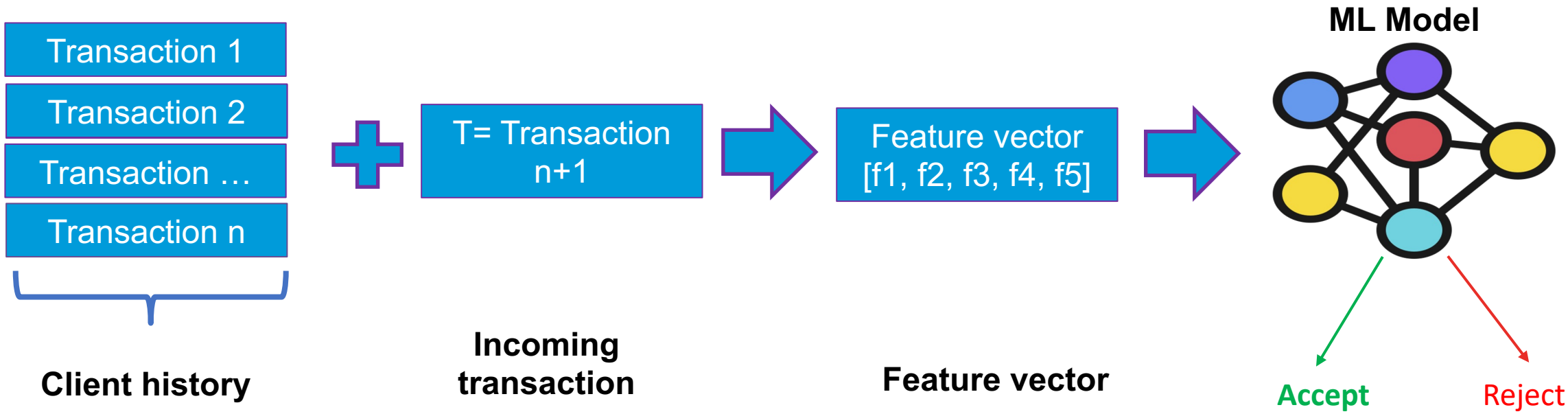
[a] Department of Software Convergence, Graduate School of Soongsil University, Seoul, 07027, South Korea
[b] Department of Cyber Security and Police, Busan University of Foreign Studies, Busan, 46234, South Korea
[c] Department of Software, Soongsil University, Seoul, 07027, South Korea

ARTICLE INFO

ABSTRACT

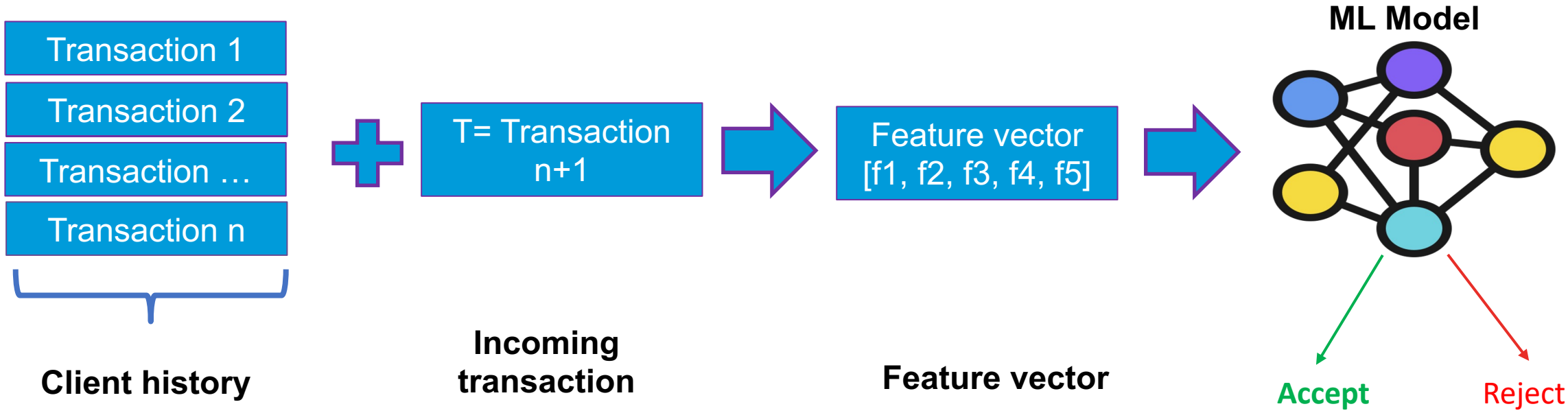
Machine learning robustness (evasion attacks)

Automated decision software in finance



Machine learning robustness (evasion attacks)

Automated decision software in finance



Evasion attack goal

Make the smallest change in transaction n+1

Such that the decision changes from reject to accept

Intersecting variability, machine learning and testing?

We are familiar with your previous work on variability analysis and verification, and would like to know whether you could provide a perspective on the interaction of variability, testing and machine learning, given that you have been working around these topics in the recent years, and they directly relate to the VaMoS topics.

Intersecting variability, machine learning and testing?

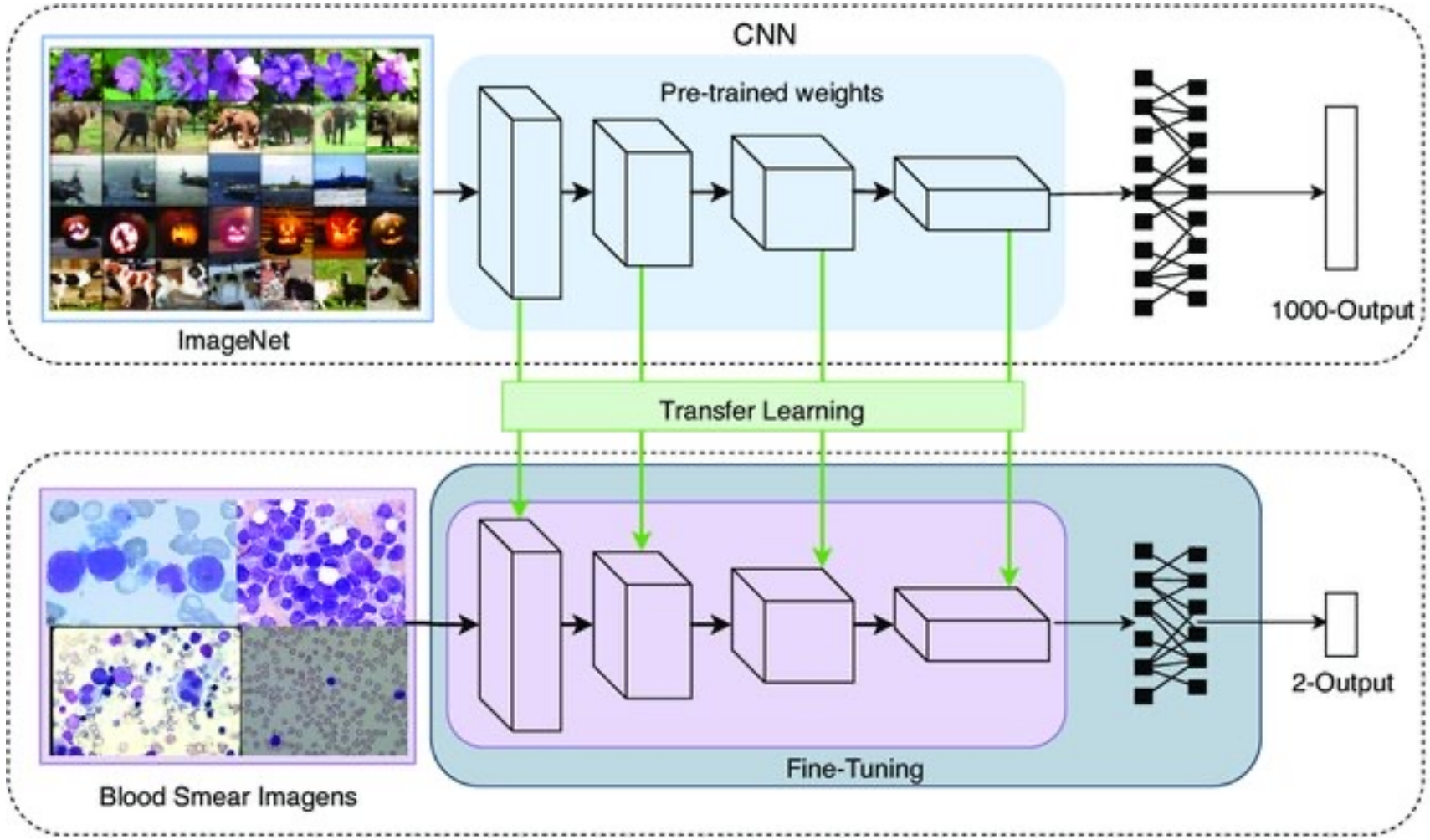
Early-stage research ahead!

Please change room or come back later for solidly established research ☺

Reusability, adaptability, and exploration (in ML)

Reusability, adaptability, and exploration (in ML)

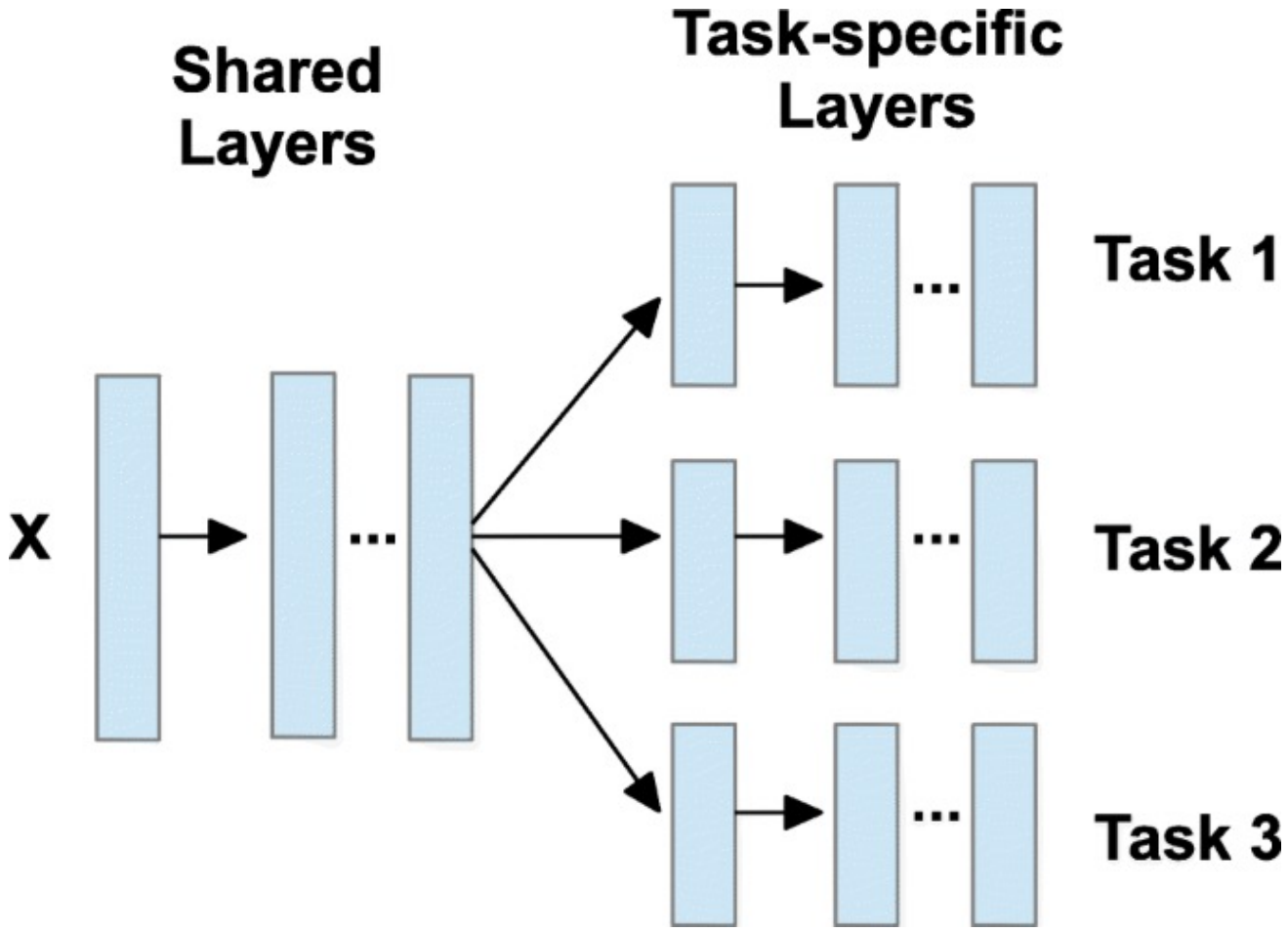
Transfer learning and fine-tuning



Luis Enrique Silva Vogado et al. "Diagnosis of Leukaemia in Blood Slides Based on a Fine-Tuned and Highly Generalisable Deep Learning Model", Sensors 21(9):2989.

Reusability, adaptability, and exploration (in ML)

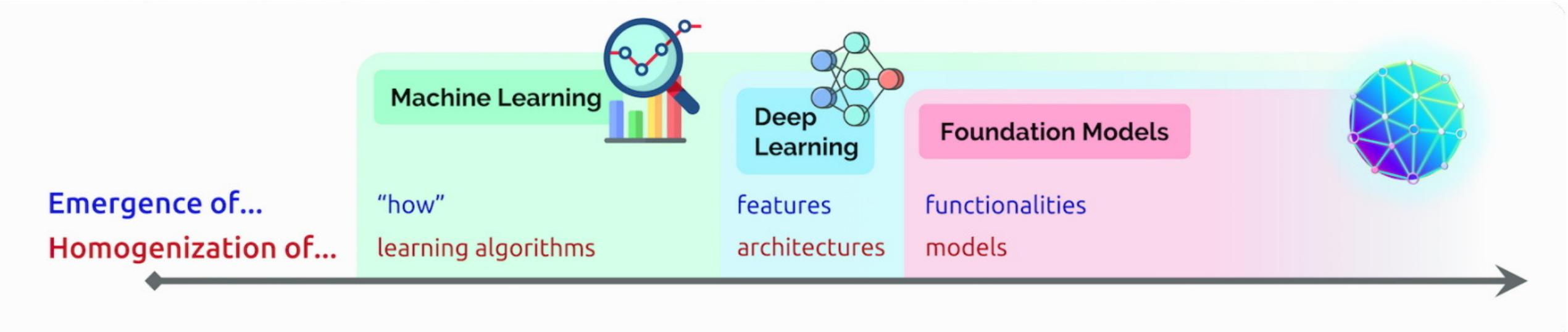
Multi-task learning



Kim-Han Thung and Chong-Yaw Wee. “A brief review on multi-task learning”, Multimedia Tools and Applications. Volume 77, pages 29705–29725, (2018).

Reusability, adaptability, and exploration (in ML)

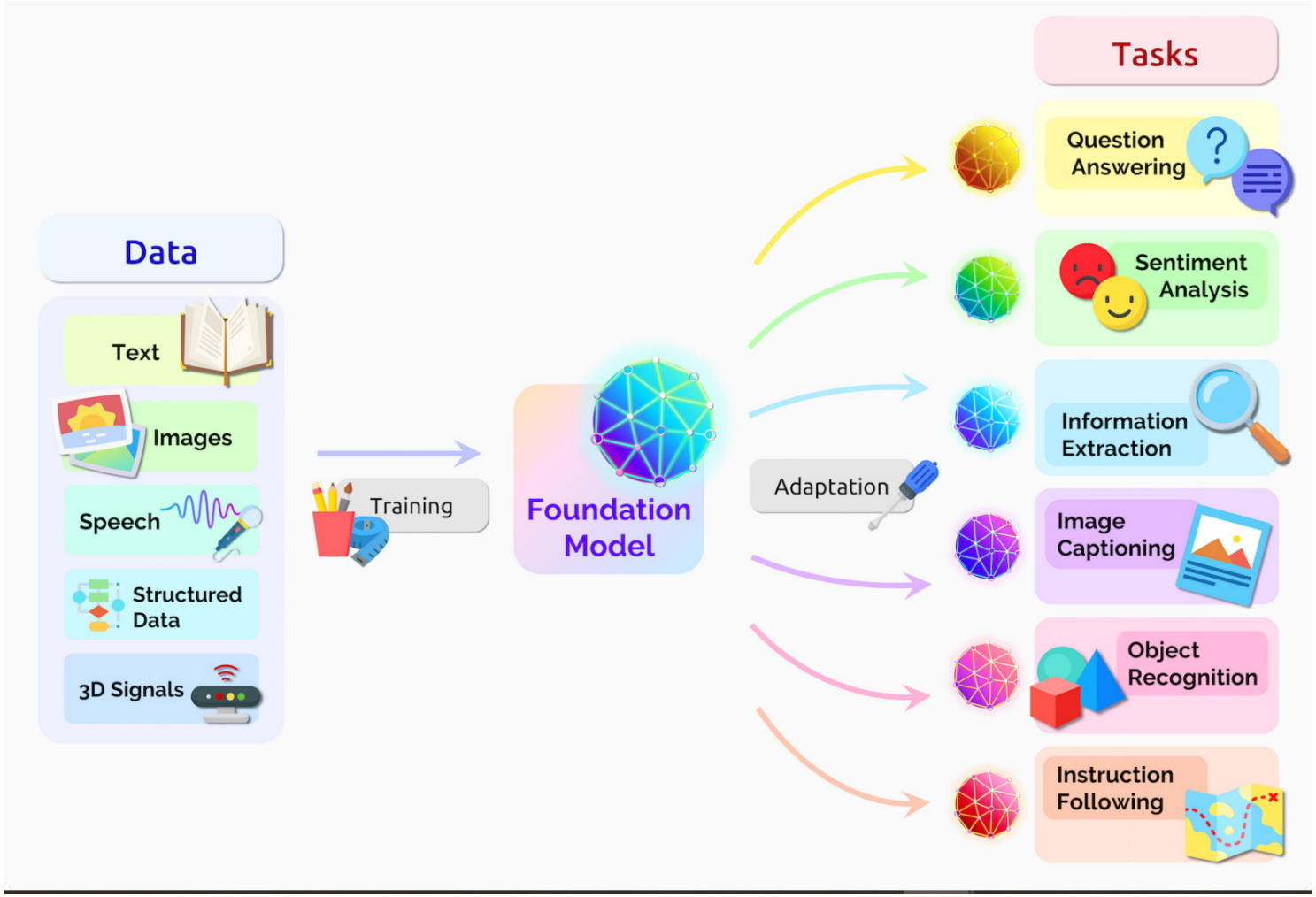
Foundation Models



https://blogs.nvidia.com/blog/what-are-foundation-models/

Reusability, adaptability, and exploration (in ML)

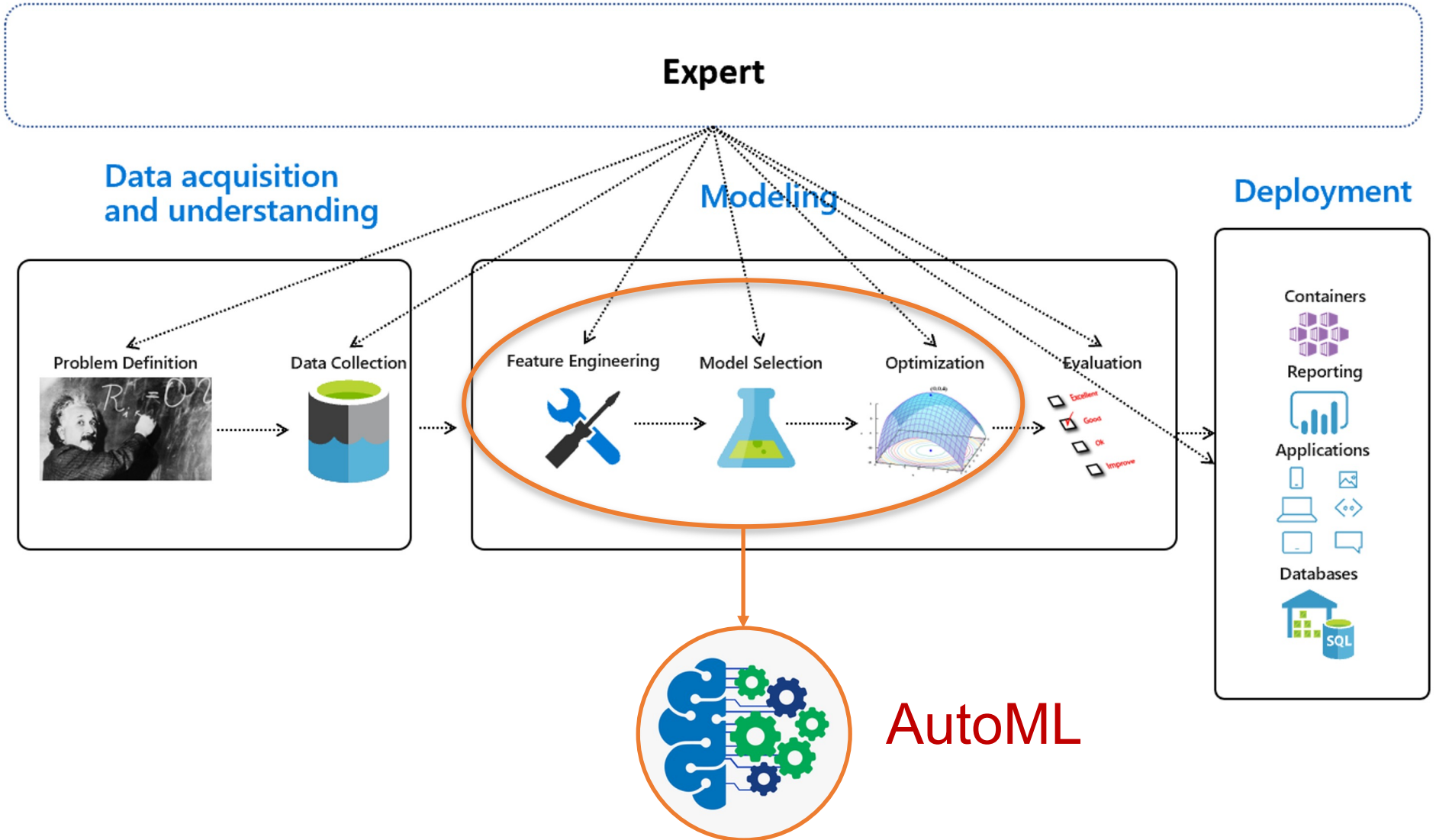
Foundation Models



https://blogs.nvidia.com/blog/what-are-foundation-models/

Reusability, adaptability, and exploration (in ML)

AutoML



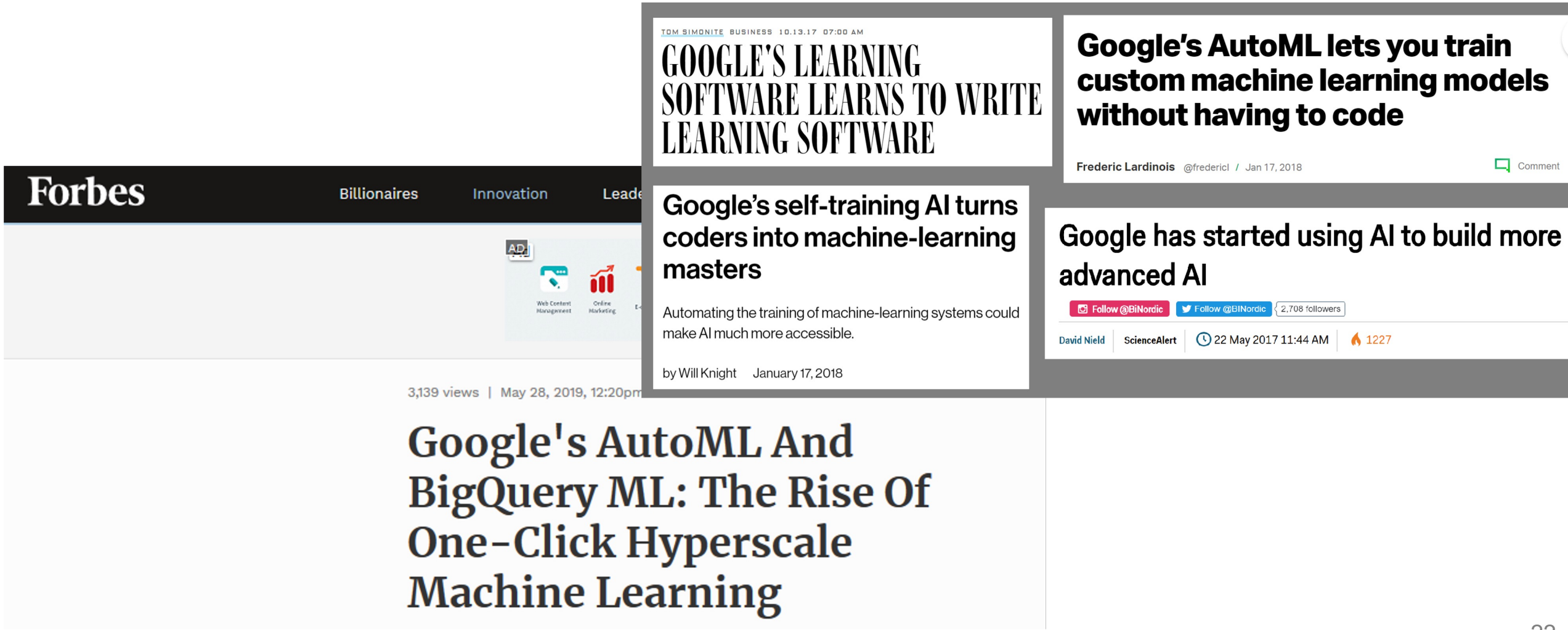
Automated Search for Configurations of Convolutional Neural Network Architectures

Salah Ghamizi, Maxime Cordy,
Mike Papadakis, Yves Le Traon

https://github.com/yamizi/FeatureNet

SPLC Paris 2019
23rd Systems and Software Product Line Conference

AutoML: exponential growth since 2017!



Our research questions

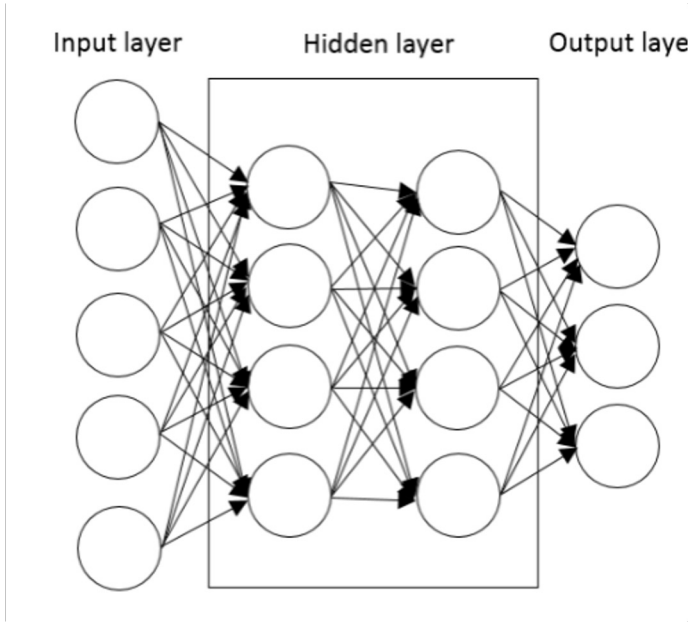
RQ1: Can we develop a variability model that represents all possible DNN architectures?

RQ2: Can we effectively search the configuration space and identify well-performing DNN architectures?

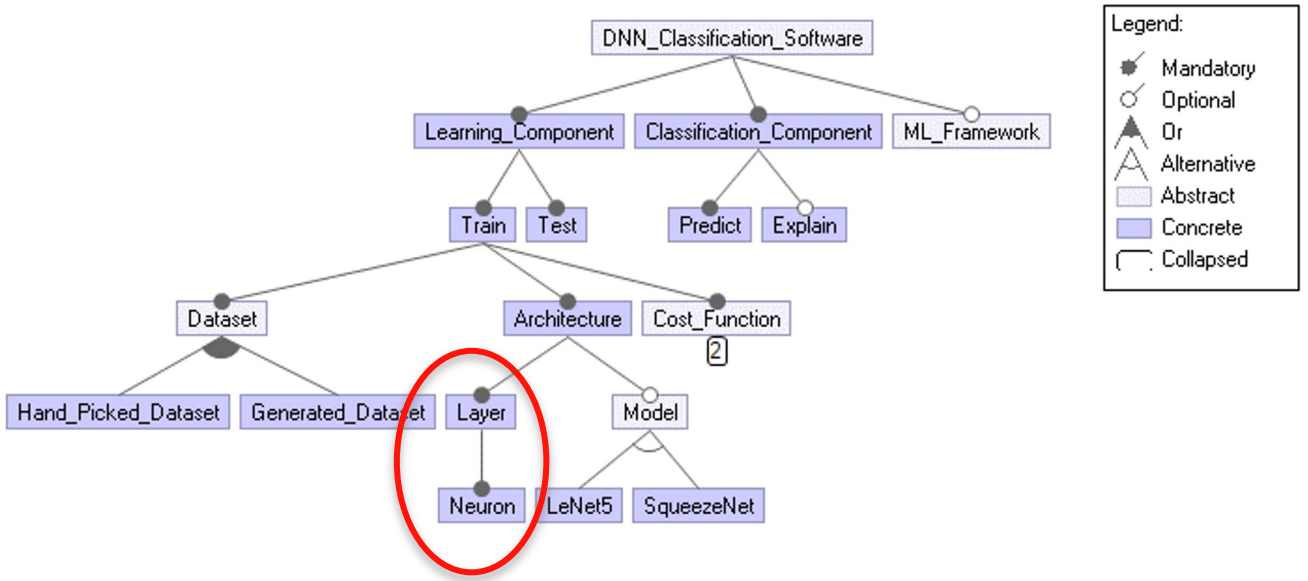
RQ3: Does our technique finds DNN architectures that outperform the state of the art?

A straightforward variability model for DNNs

a simple (dense) DL model



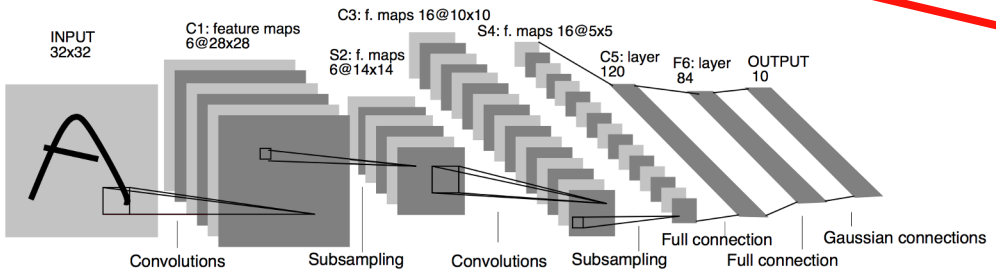
a naive feature model



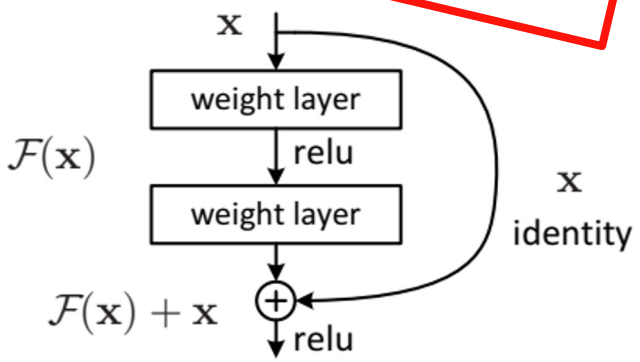
insufficient to specify newer architectures

Various architectures and hyperparameters

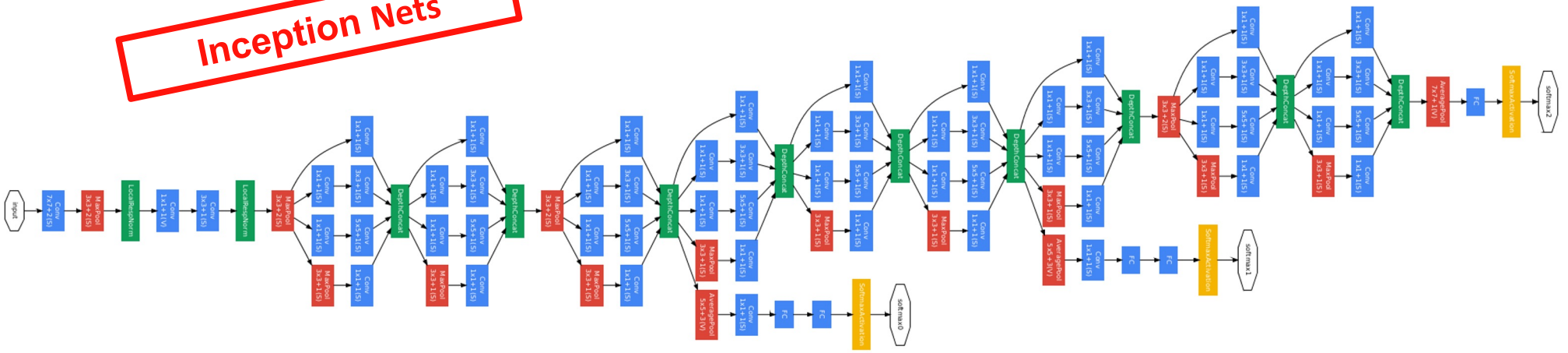
Convolutional Nets



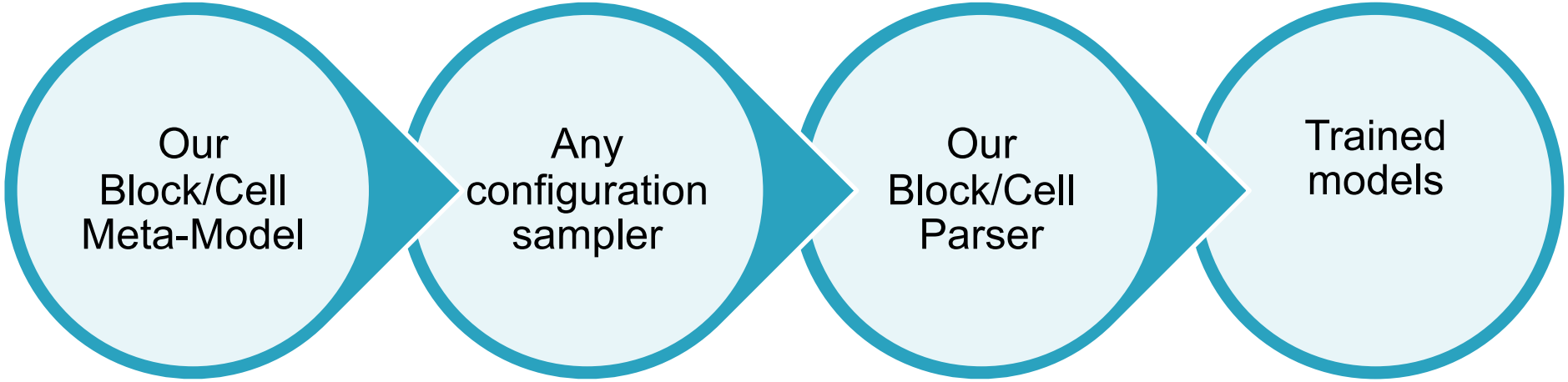
Residual Nets

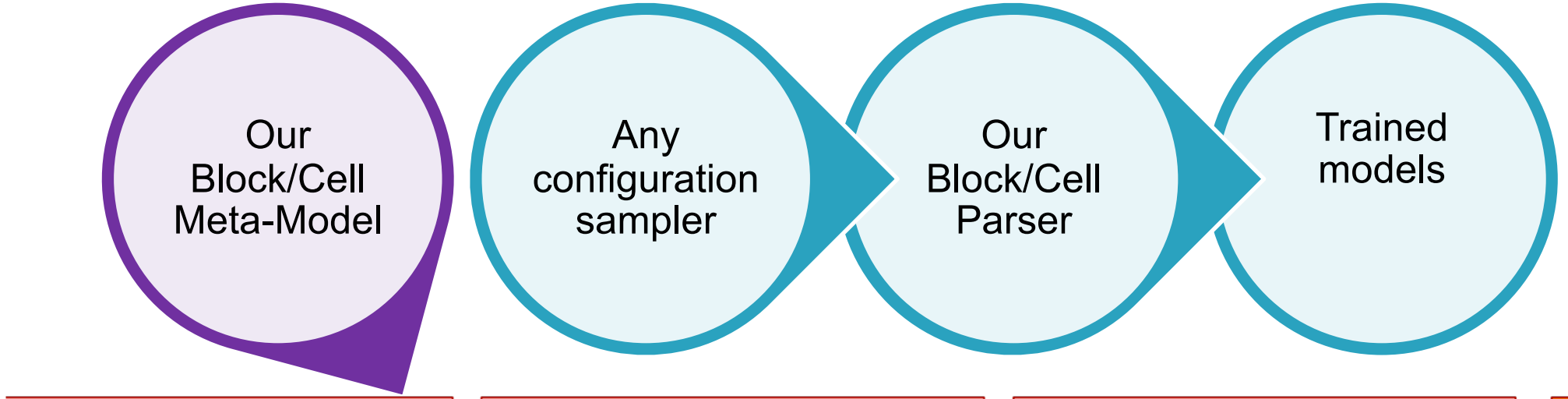


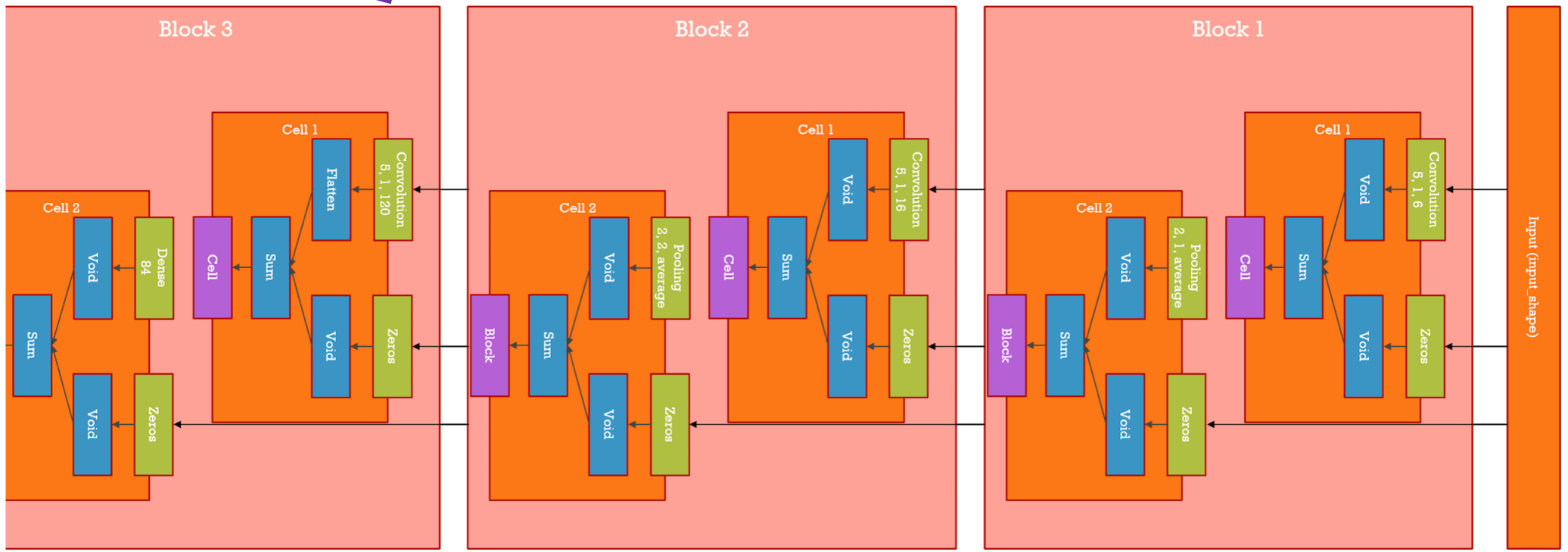
Inception Nets



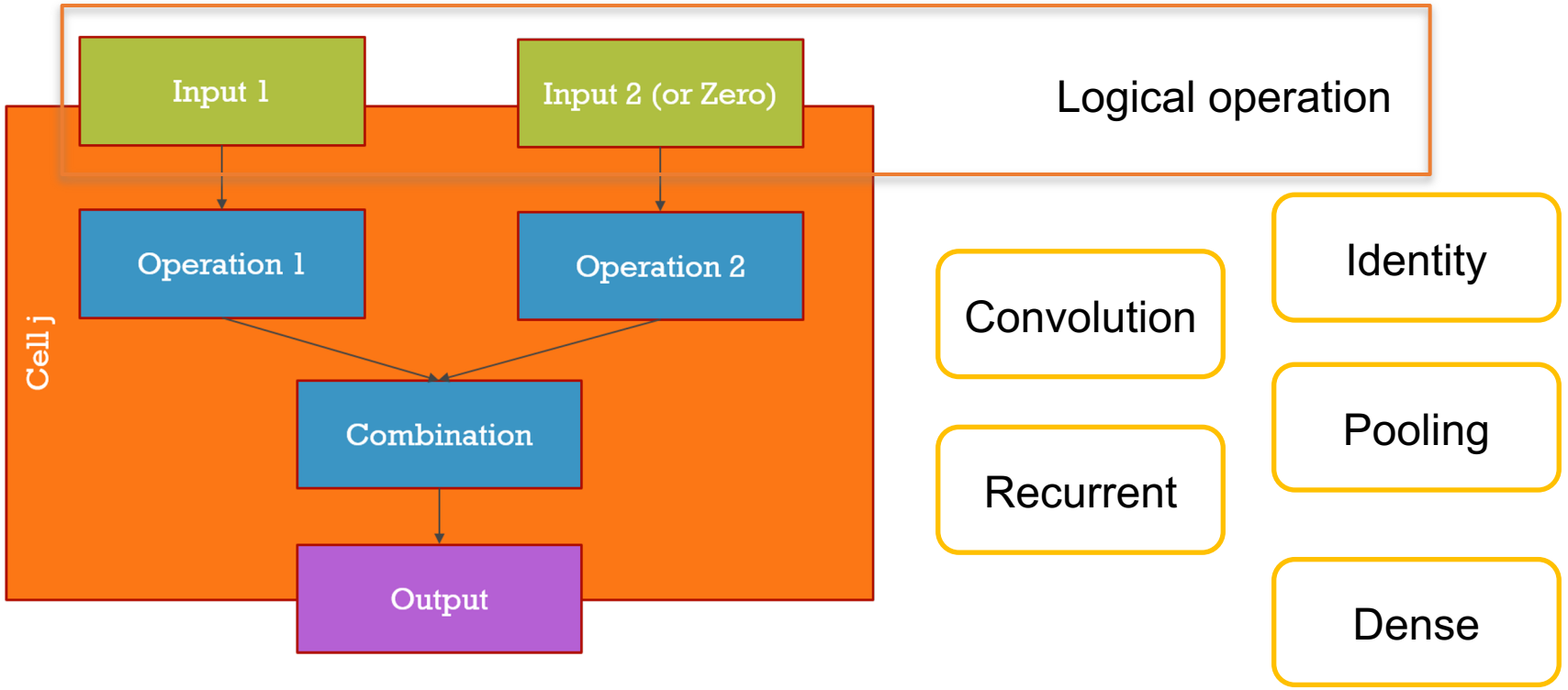
Our DNN configuration pipeline



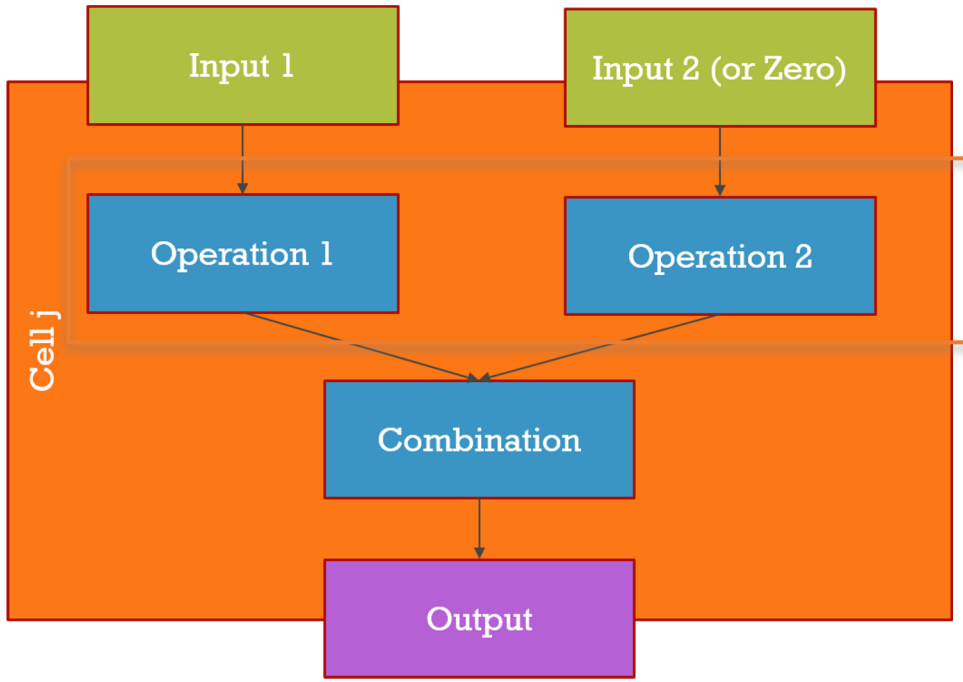

Our
Block/Cell
Meta-Model
Any
configuration
sampler
Our
Block/Cell
Parser
Trained
models


Block 3
Cell 1
Cell 2
Convolution 5, 1, 120
Flatten
Dense 84
Void
Zeros
Cell
Sum
Block 2
Convolution 5, 1, 16
Pooling 2, 2, average
Block 1
Convolution 5, 1, 6
Pooling 2, 1, average
Input (input_shape)

Meta-modelling DNNs with “Cells”



Meta-modelling DNNs with “Cells”



Regularization operation
Pre-merge operation

Dropout

Batch Normalisation

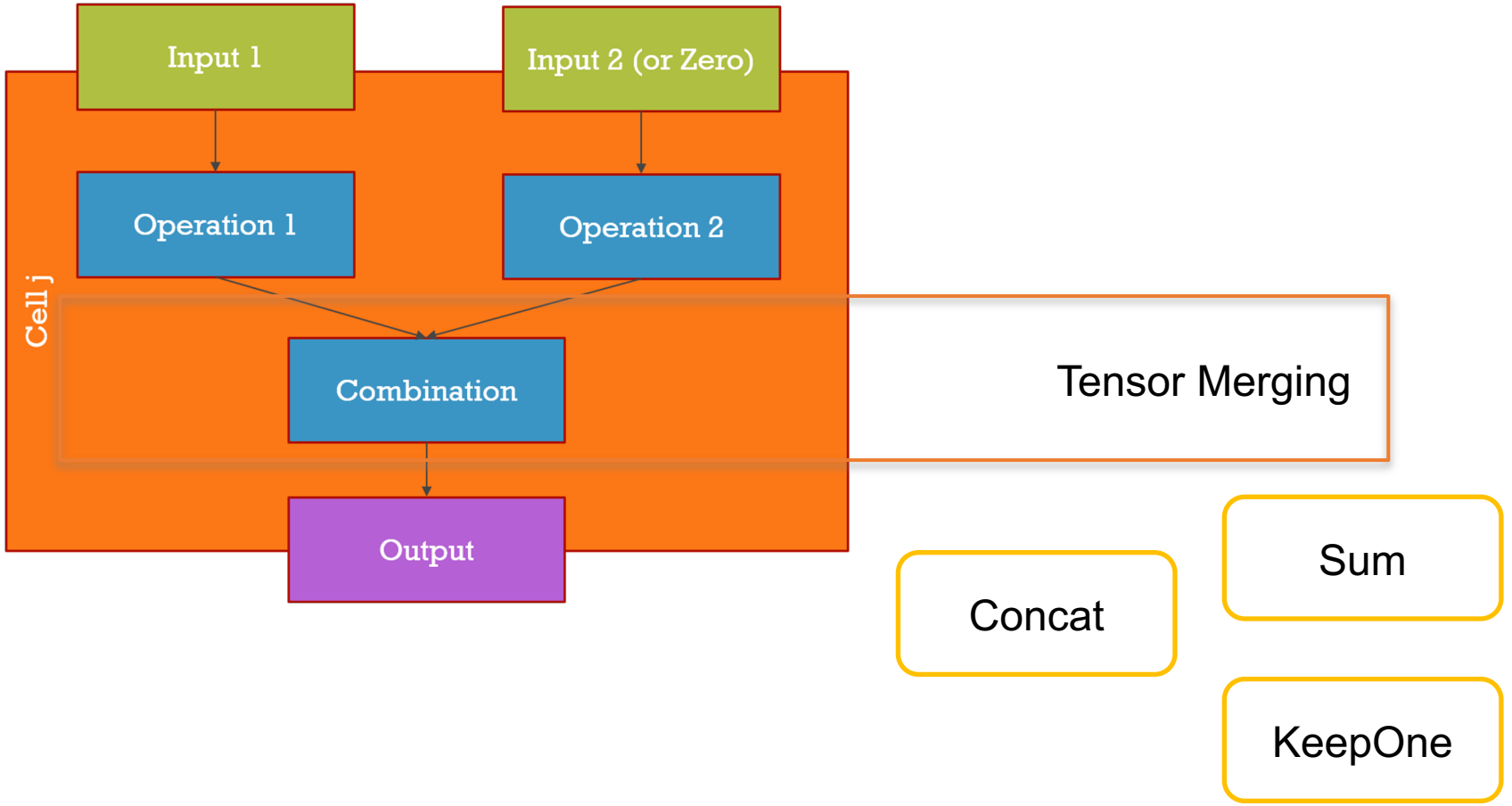
Void

Activation

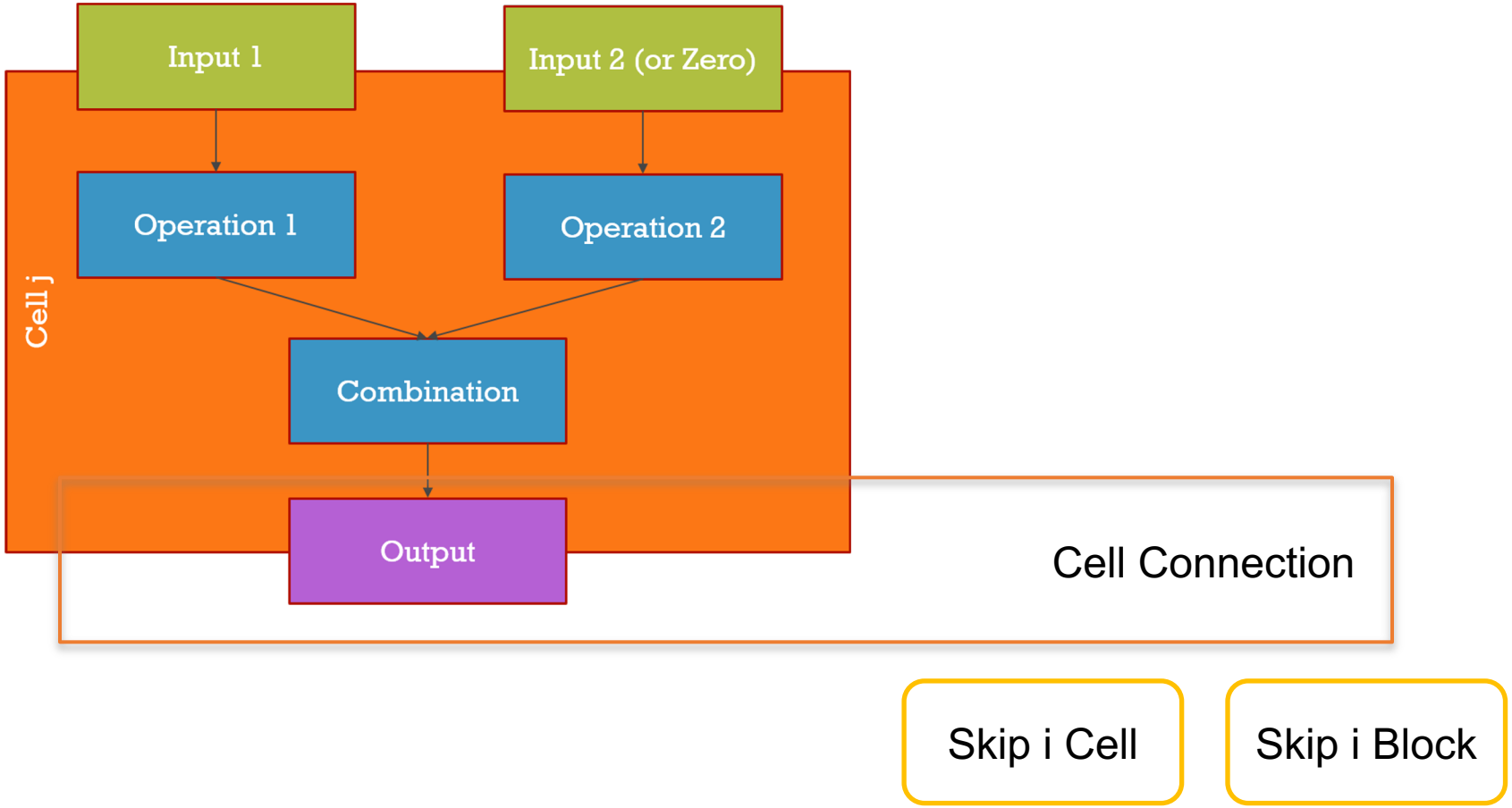
Sparsing

Resizing

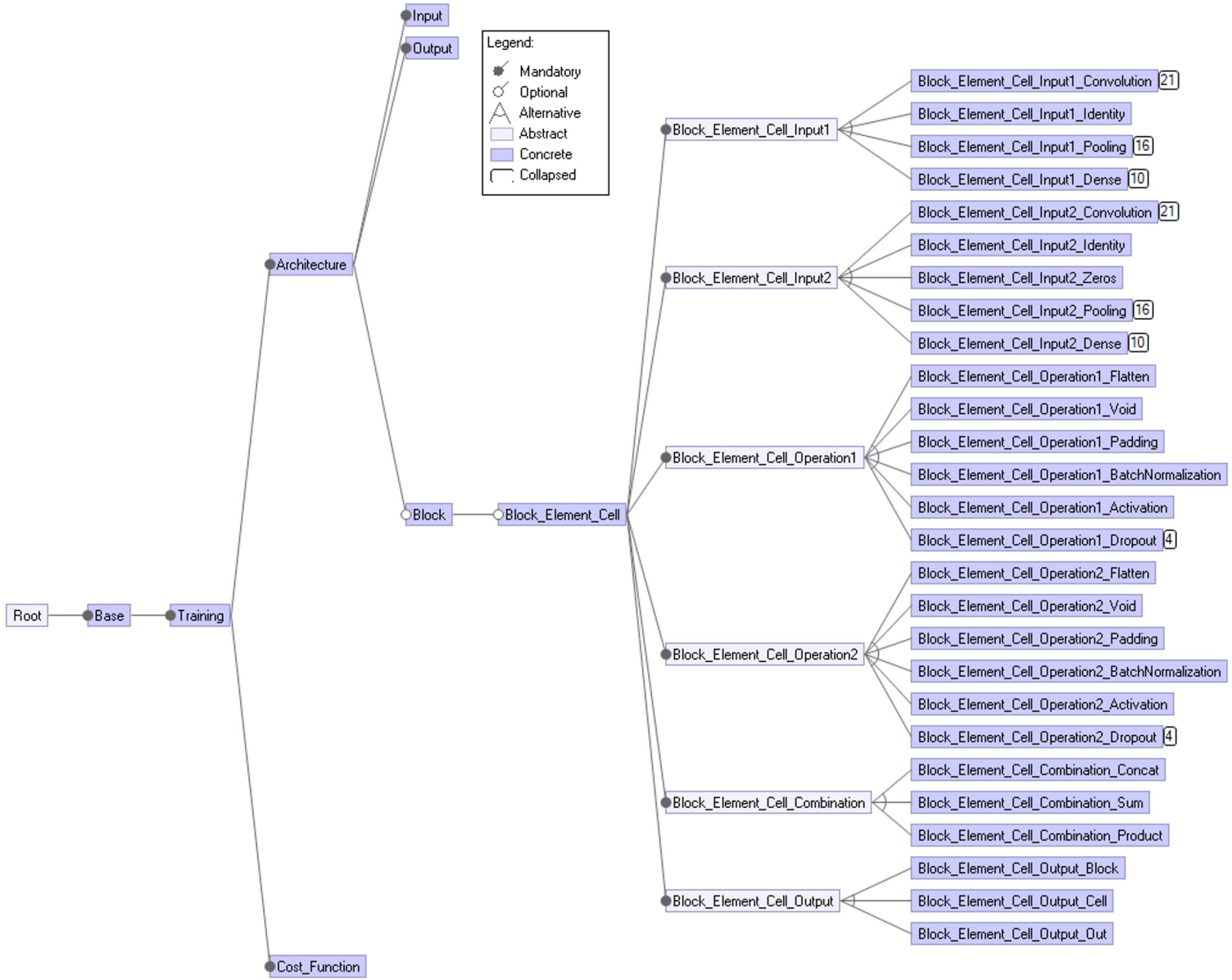
Meta-modelling DNNs with “Cells”



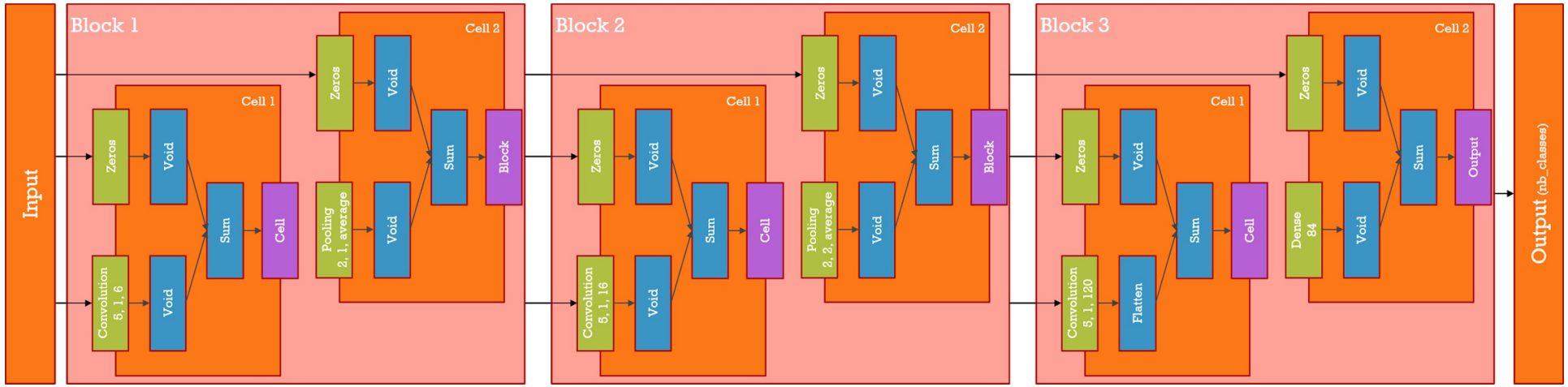
Meta-modelling DNNs with “Cells”



The corresponding variability model

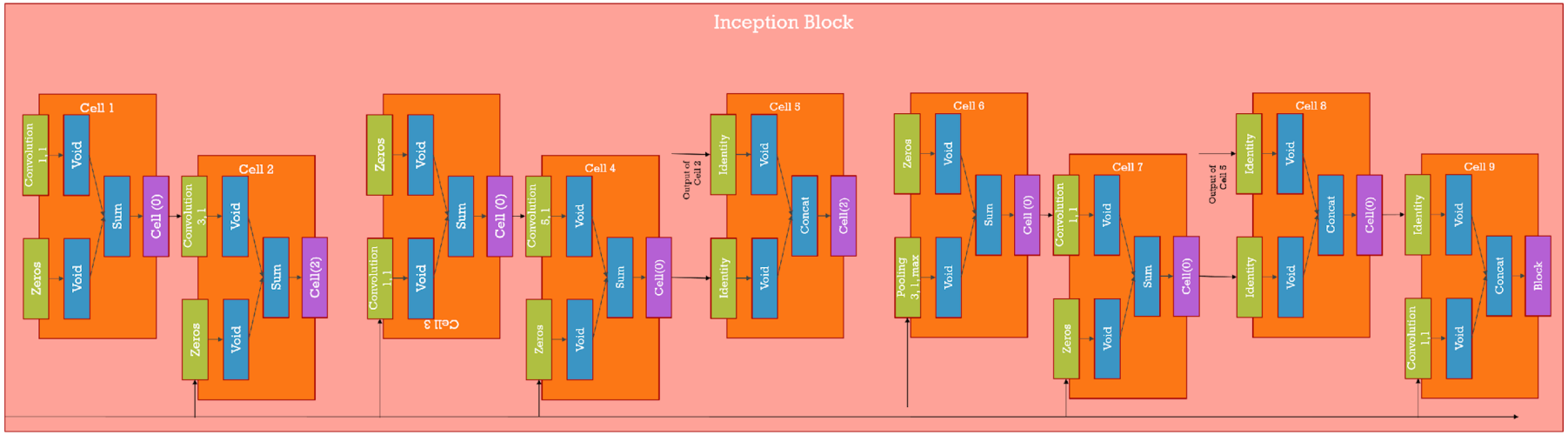


Recreating state of the art architecture



LeNet5 architecture

Inception Block



Experimental settings

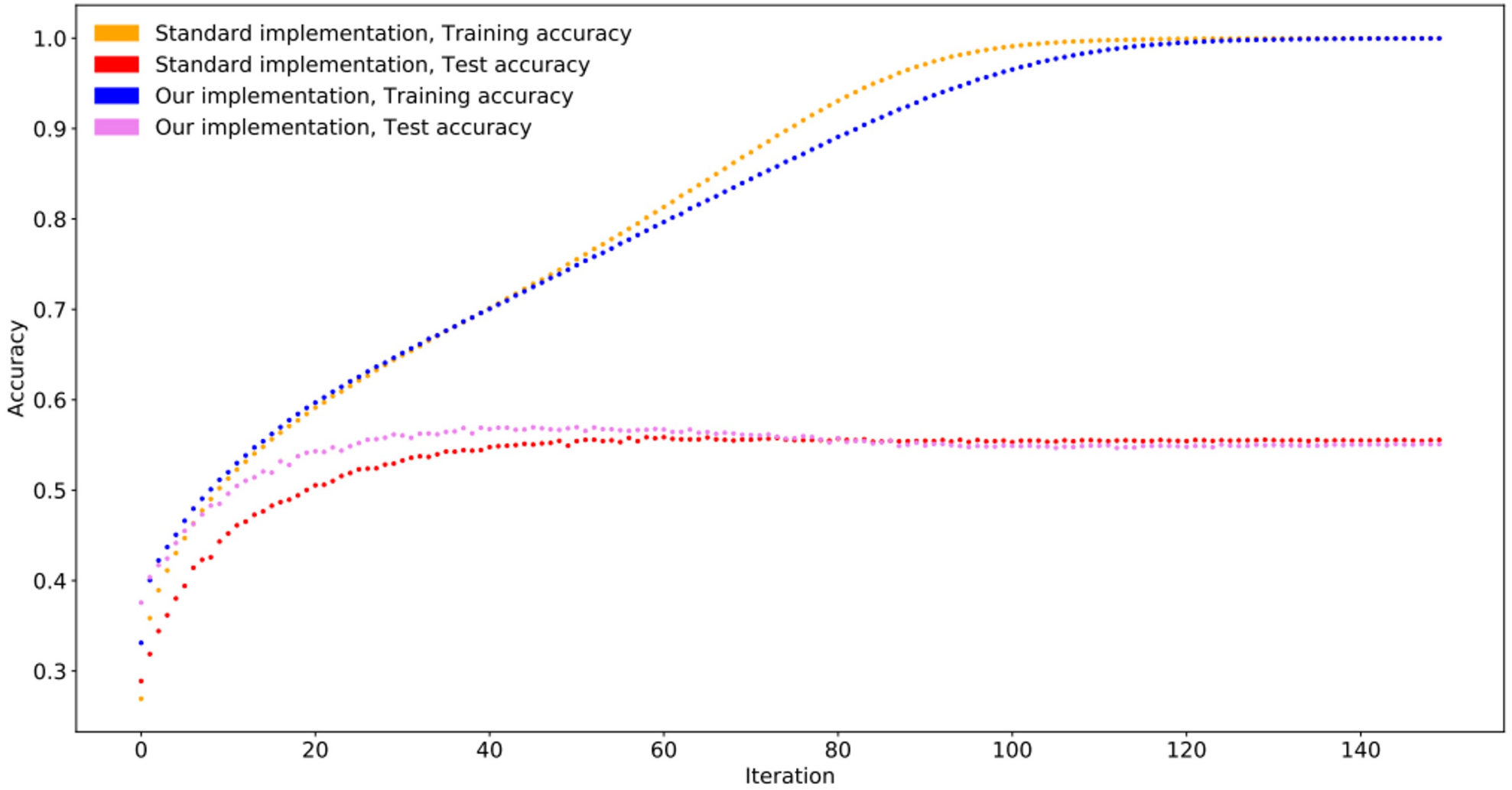
- Computational limitations
  - Maximum blocks and cells per block set to 5
  - Limitation on convolution layers' filters (128) and dense layers' neurons (512)
  - No batch normalization, padding, concatenation and product operations
  - Only one output

- SGD with 0.01 learning rate, no decay, no momentum, batch size of 128 (no optimization)

- Datasets: MNIST (60k+10k images)and CIFAR-10 (50k + 10k images)

- Quality metrics: classification accuracy and *efficiency* (accuracy divided by number of weights)

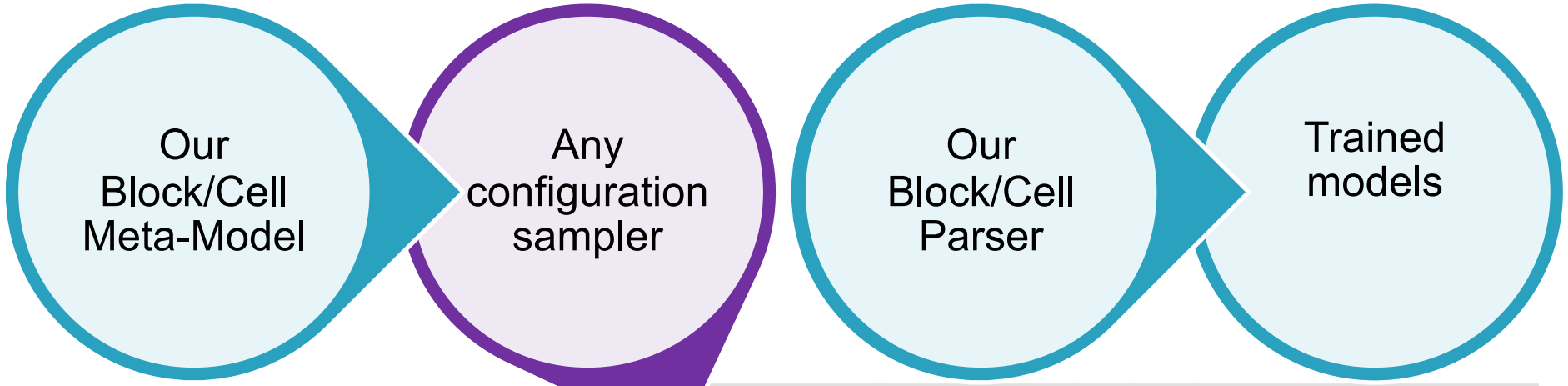
Original implementation vs ours



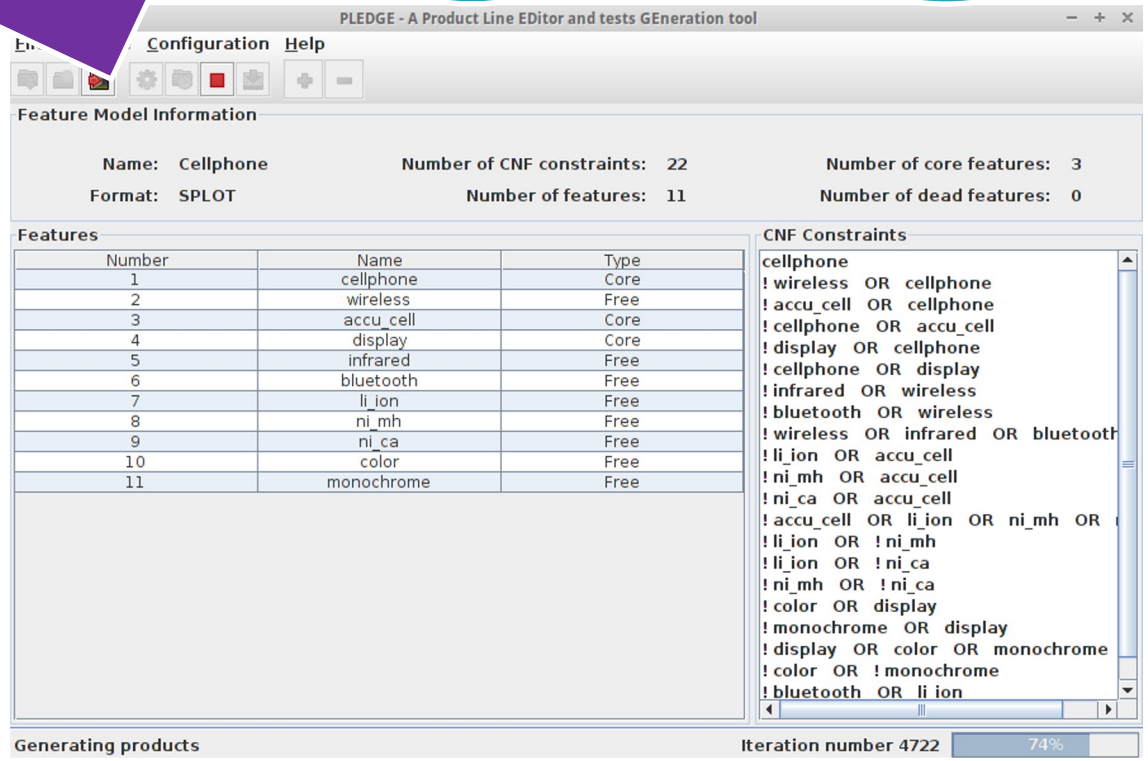
Our research questions

RQ1: Can we develop a variability model that represents all possible DNN architectures?

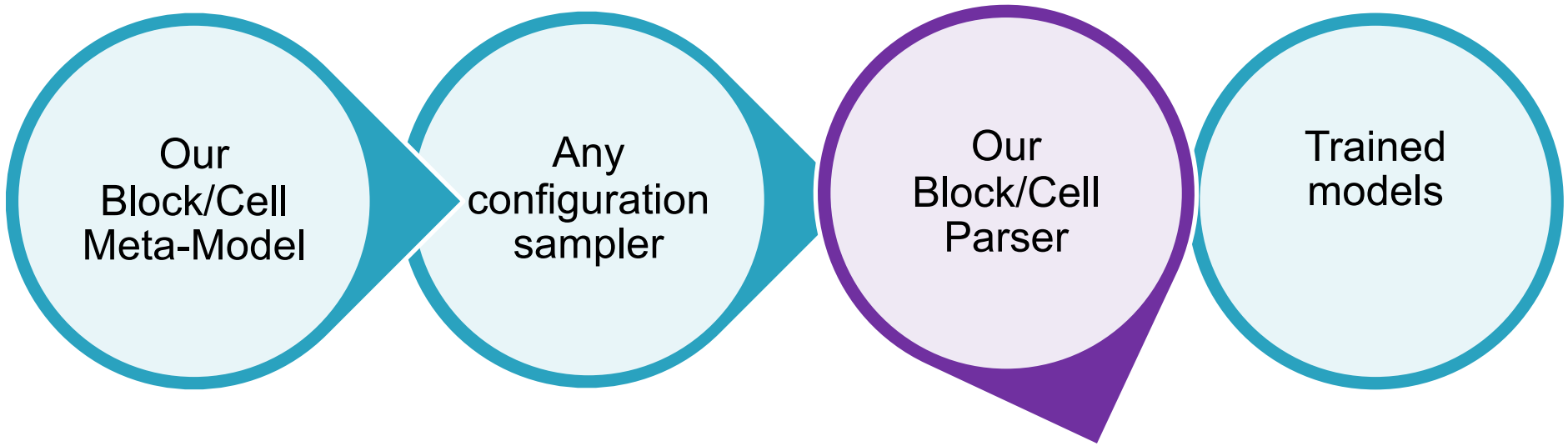
RQ2: Can we effectively search the configuration space and identify well-performing DNN architectures?



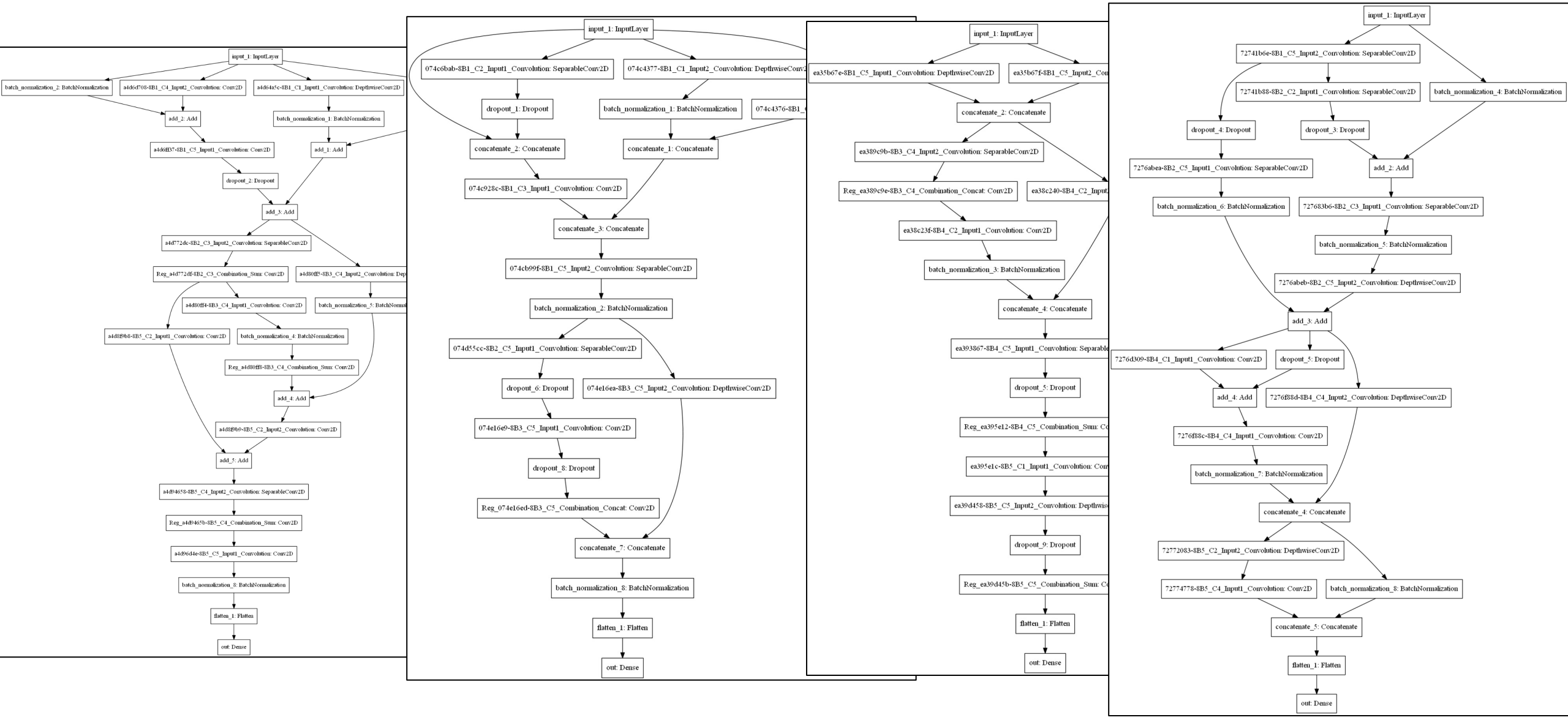
PLEDGE:
Tool for Diversity based sampling



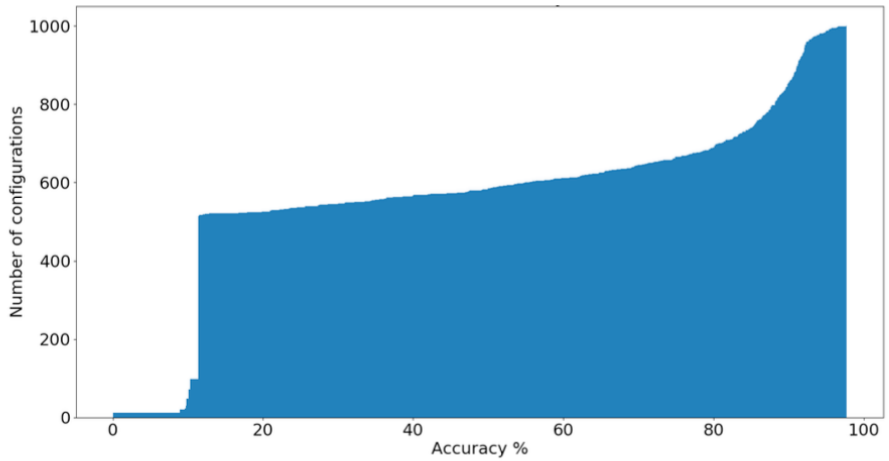
Christopher Henard, Mike Papadakis, Gilles Perrouin, Jacques Klein, and Yves Le Traon. 2013. PLEDGE: a product line editor and test generation tool (SPLC '13 Workshops).


Our
Block/Cell
Meta-Model
Any
configuration
sampler
Our
Block/Cell
Parser
Trained
models
TensorFlow
Keras

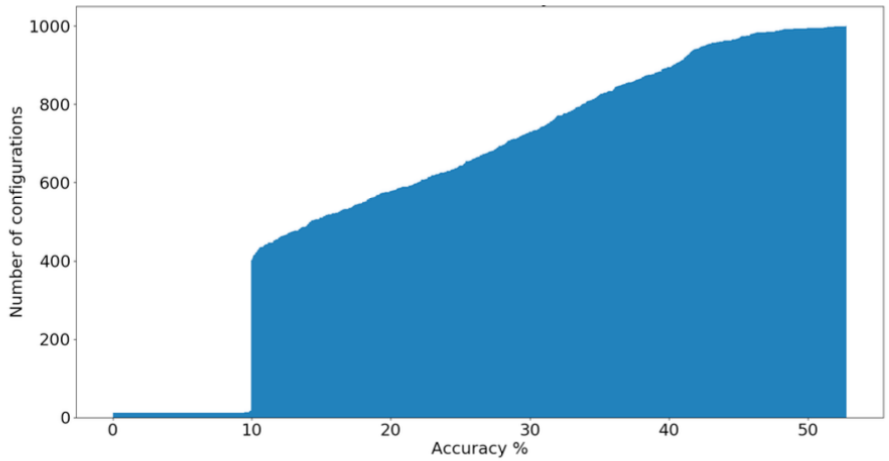
Our technique generates a wide range of architectures



… with diverse accuracy



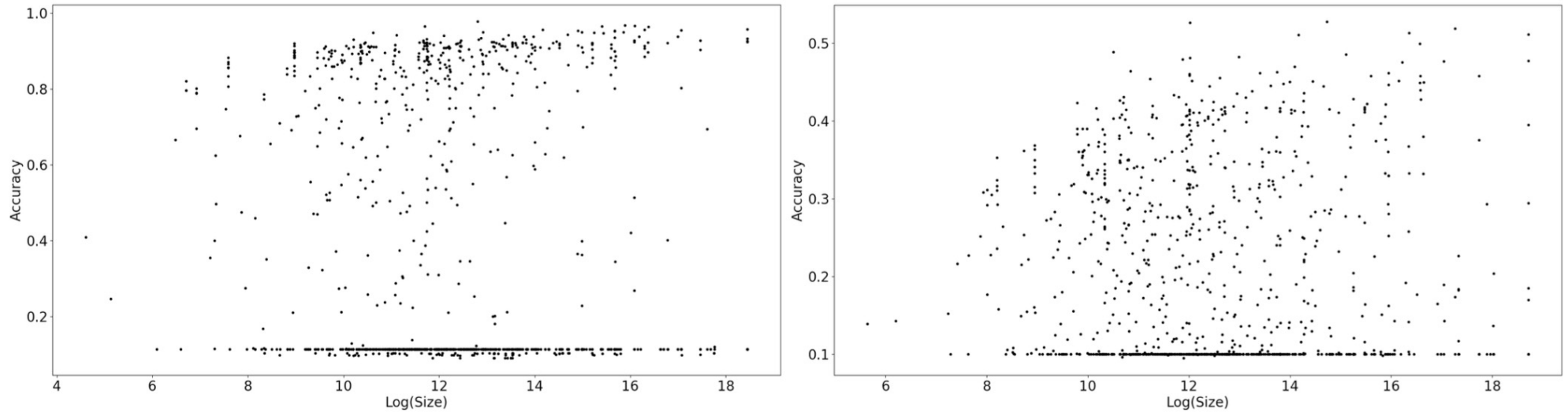
(a) On MNIST



(b) On Cifar-10

Fig. 7. Distribution of the 1,000 generated architectures over all percentages of accuracy on two datasets. Any point (x, y) of the graph denotes that y architectures achieve an accuracy lower than $x\%$.

… and with diverse size



(a) On MNIST

(b) On Cifar-10

Fig. 8. Distribution of the accuracy and size of 1,000 generated architectures. The size is given in terms of number of trainable weights of the architectures. By sampling a diverse set of configurations (in terms of feature differences), our technique is able to generate architectures with a wide range of sizes an few thousands to millions of weights. Moreover, we observe that architectures with high accuracy are not necessarily the ones with the largest size, while architectures performing poorly are found in all size ranges.

Our research questions

RQ1: Can we develop a variability model that represents all possible DNN architectures?

RQ2: Can we effectively search the configuration space and identify well-performing DNN architectures?

RQ3: Does our technique finds DNN architectures that outperform the state of the art?

FeatureNet vs. SotA architectures * (MNIST)

Table 1: Accuracy, size, and efficiency of LeNet5, SqueezeNet and our best sampled architectures, with a 12-epoch training, on MNIST (top part) and CIFAR-10 (bottom part).

Dataset	Architecture	Accuracy	Size	Efficiency
MNIST	LeNet5	97.14%	545546	1.78
	Top S1	97.74%	365194	2.68
	Top S2	97.65%	570218	1.71
	Top S3	92.31%	43578	21.18
	SqueezeNet	43.67%	858154	0.51
CIFAR-10	LeNet5	49.13%	868406	0.57
	Top S1	52.77%	2494858	0.21
	Top S2	57.79%	862646	0.67
	Top S3	37.44%	38842	9.64
	SqueezeNet	17.96%	876970	0.51

1,000 architectures with previous restrictions

100 LeNet5 architectures

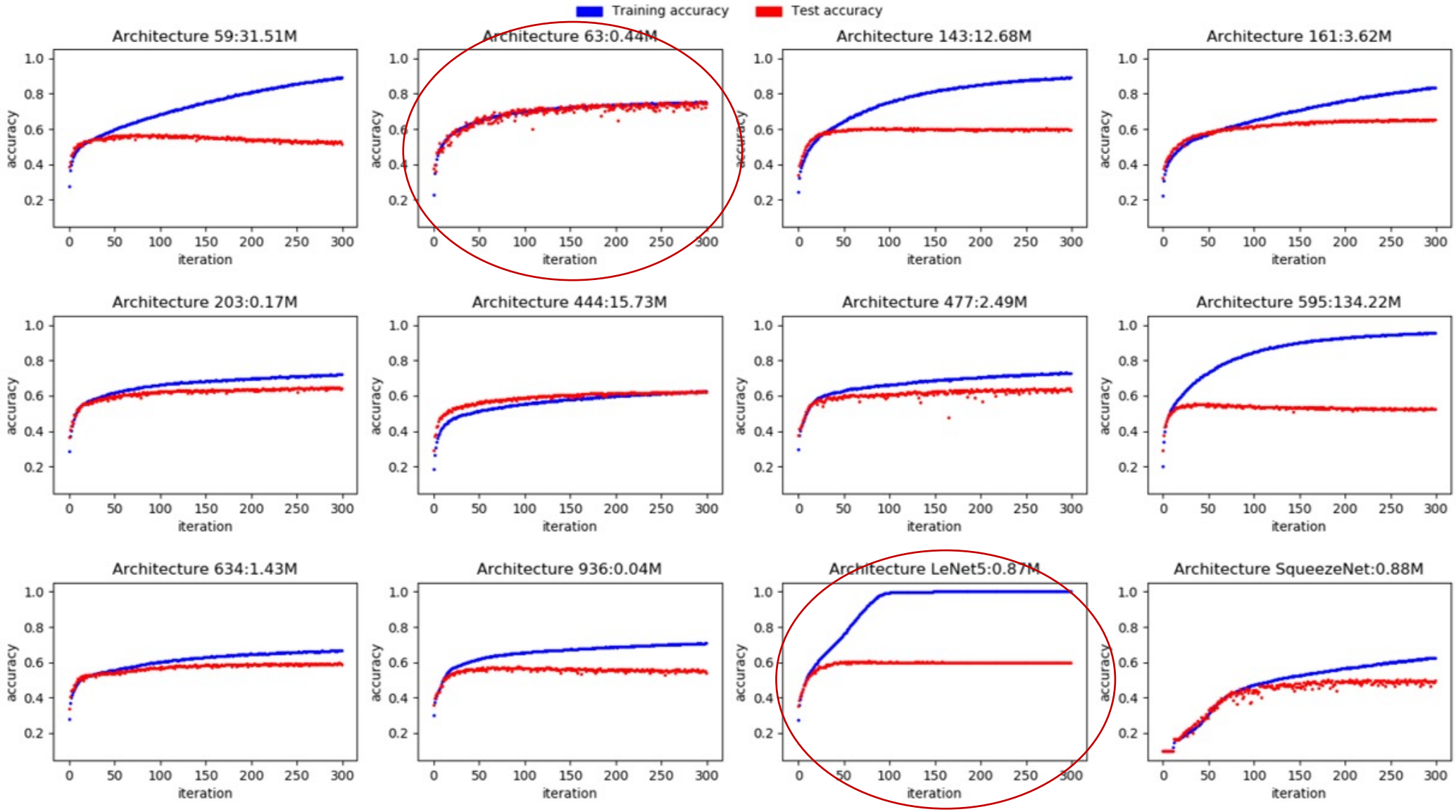
100 smaller architectures

FeatureNet vs. SotA architectures * (CIFAR-10)

Table 2: Accuracy of the 10 best architectures from S1, LeNet5 and SqueezeNet on CIFAR-10 and at 12, 300 and 600 training epochs. Architectures are ordered by descending order of accuracy at 600 epochs. * indicates shortened training.

Architecture	Size	(12)	(300)	(600)
#063	0.45M	48.25%	**74.28%**	**74.74%**
#203	0.17M	52.64%	64.55%	65.25%
#161	3.62M	48.57%	64.46%	65.24%
#477	2.49M	**52.77%**	63.43%	64.25%
#444	15.73M	49.97%	62.40%	62.80%
#143	12.68M	51.37%	60.46%	60.17%
LeNet5	0.87M	49.13%	59.42%	59.26%
#634	1.43M	51.11%	59.76%	* 59.06%
#936	0.04M	48.92%	54.26%	56.33%
#595	134.22M	52.16%	52.64%	* 53.64%
SqueezeNet	0.88M	17.96%	46.17%	49.69%
#059	31.51M	51.9%	52.10%	49.47%

Variability in performance



vs. other NAS methods

On Cifar-10:

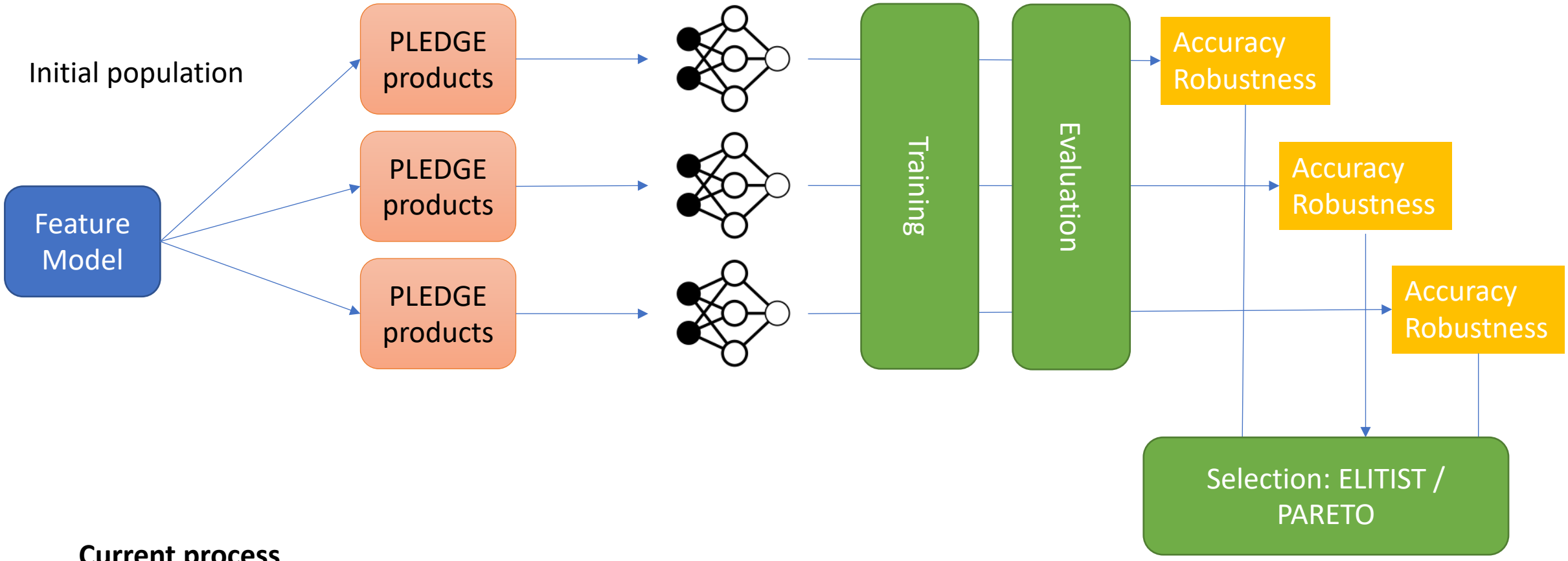
NAS technique	Error
FeatureNet	25%
AutoKeras (TexasU,18')	15%
NasNet (Google, 18')	2.4%

But FeatureNet:

- Deals only with the architectures, others also consider:
  - Training optimization
  - Data Augmentation
- Is more efficient:
  - our full search took less than 1 GPU day
  - NasNet requires hundreds to thousands of GPU days

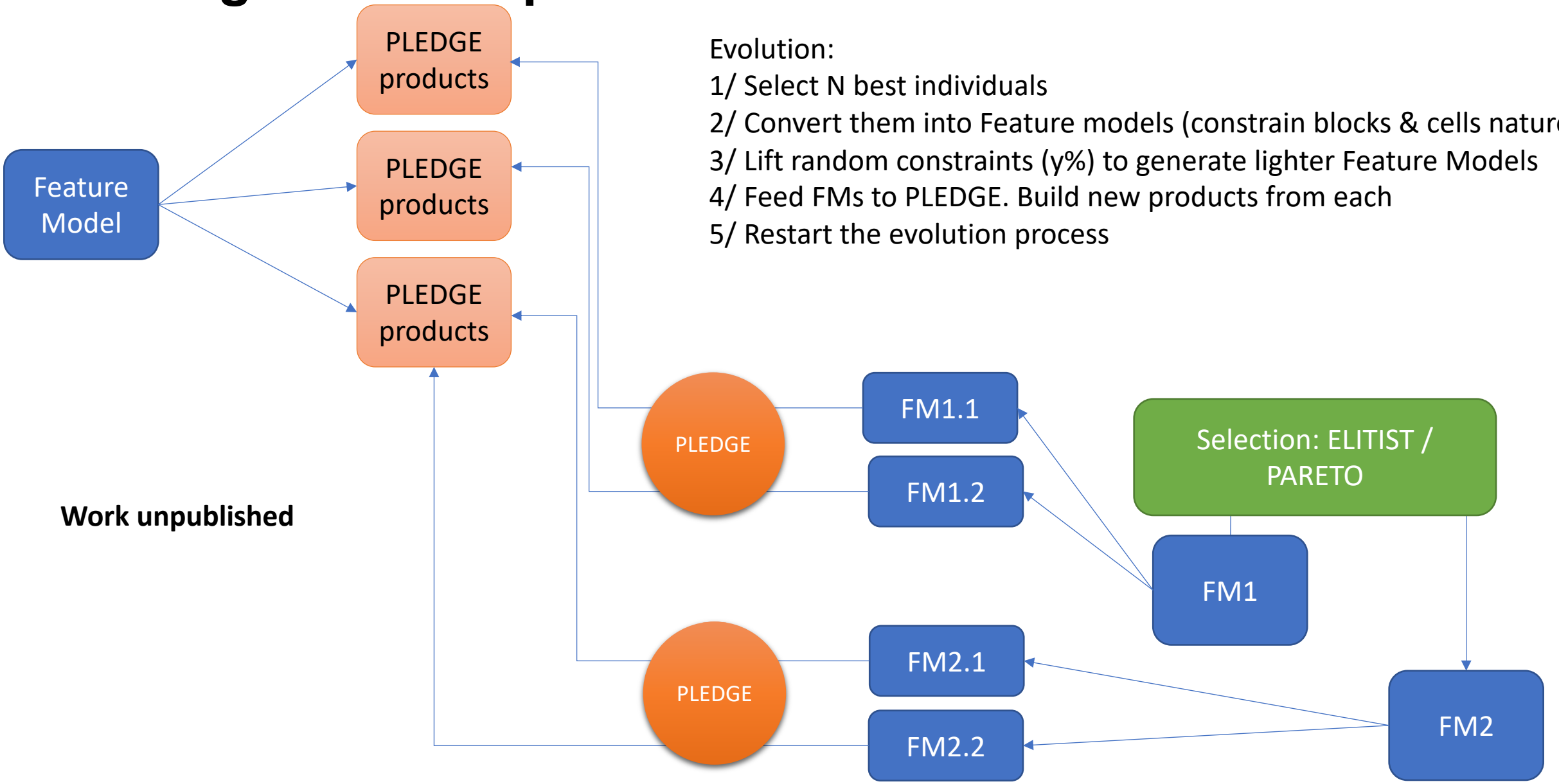
Moving ahead

Moving ahead: improved search



Current process

Moving ahead: improved search



Evolution:
1/ Select N best individuals
2/ Convert them into Feature models (constrain blocks & cells nature)
3/ Lift random constraints (y%) to generate lighter Feature Models
4/ Feed FMs to PLEDGE. Build new products from each
5/ Restart the evolution process

Work unpublished

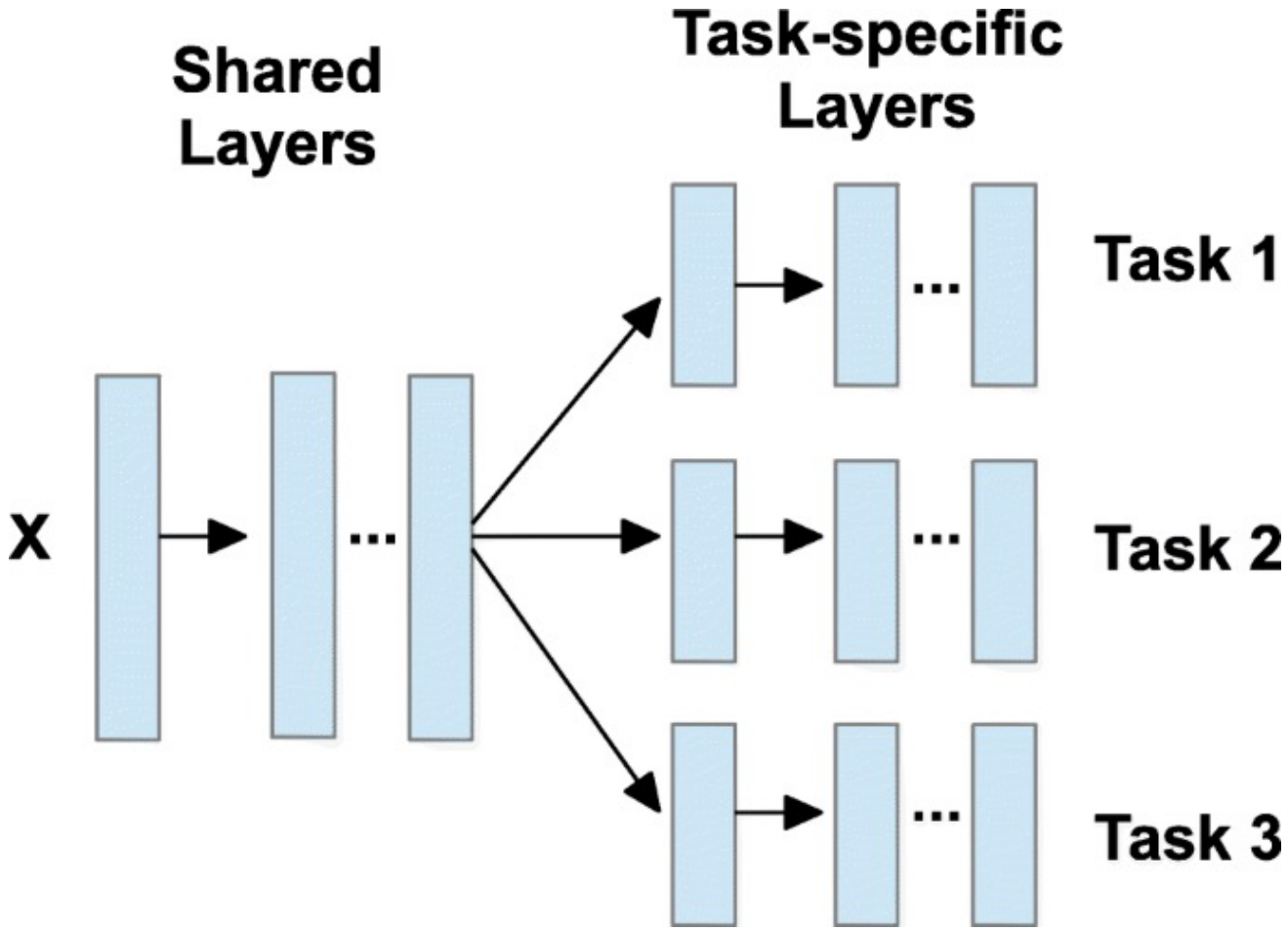
Adversarial Robustness in Multi-Task Learning: Promises and Illusions

Salah Ghamizi, Maxime Cordy,

Mike Papadakis, Yves Le Traon

Reusability, adaptability, and exploration (in ML)

Multi-task learning



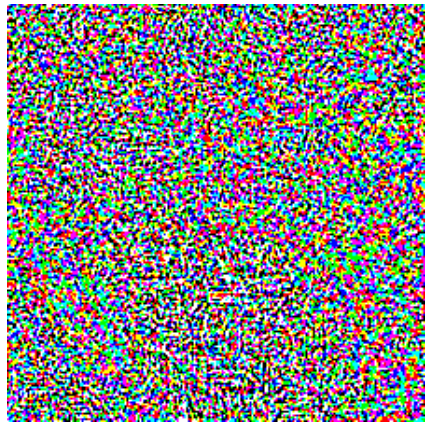
Kim-Han Thung and Chong-Yaw Wee. “A brief review on multi-task learning”, Multimedia Tools and Applications. Volume 77, pages 29705–29725, (2018).

Machine learning robustness (evasion attacks)

Original example

ML predicts:
“Panda”
(80% confidence)

Small adversarial noise



Adversarial example

What humans still see

Gibbon

What ML predicts: “Gibbon”
(99% confidence)

"Explaining and Harnessing Adversarial Examples", Goodfelow et al., ICLR 2015.

Principles of evasion attacks

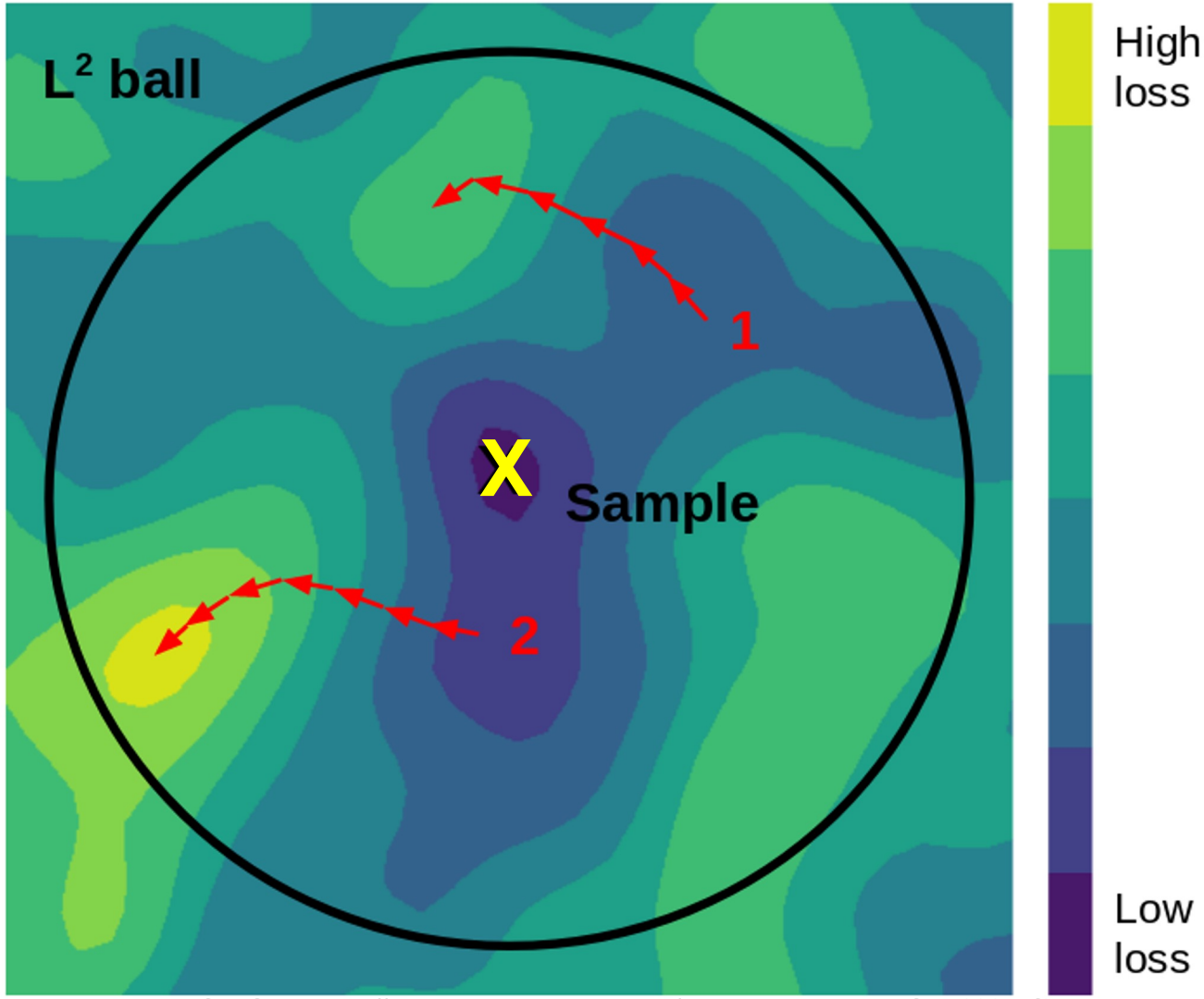


Image modified from https://towardsdatascience.com/know-your-enemy-7f7c5038bdf3

Objective: for x find δ

- ✓ With $f(x) \neq f(x+\delta)$
- ✓ With $L_p(x, x+\delta) < \epsilon$

By solving the optimization problem:

$$\underset{\delta}{\operatorname{argmax}}\ \mathcal{L}(x+\delta, y)$$

$$\text{s.t. } ||\delta||_p \leq \epsilon$$

Evasion attacks in multi-task learning

An attacker seeks the perturbation δ that will maximize the joint loss function of the attacked tasks – i.e. the summed loss, within a p-norm bounded distance ϵ.

$$\underset{\delta}{\operatorname{argmax}} \mathscr{L}_i(x+\delta, y_i) \text{ s.t. } \|\delta\|_p \leq \epsilon$$

Single-task adversarial attacks

$$\underset{\delta}{\operatorname{argmax}} \mathscr{L}(x+\delta, \bar{y}) \text{ s.t. } \|\delta\|_p \leq \epsilon$$

Multi-task adversarial attacks

Does learning multiple tasks increase robustness?

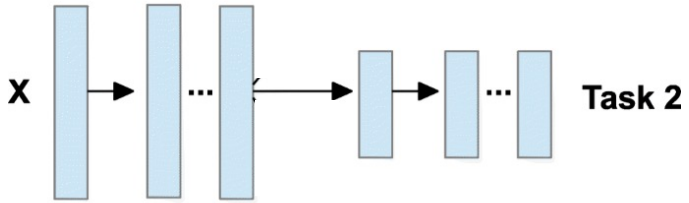
Multitask Learning Strengthens Adversarial Robustness

Chengzhi Mao, Amogh Gupta*, Vikram Nitin*,
Baishakhi Ray, Shuran Song, Junfeng Yang, and Carl Vondrick

Columbia University, New York, NY, USA
mcz,rayb,shurans,junfeng,vondrick@cs.columbia.edu,
ag4202,vikram.nitin@columbia.edu

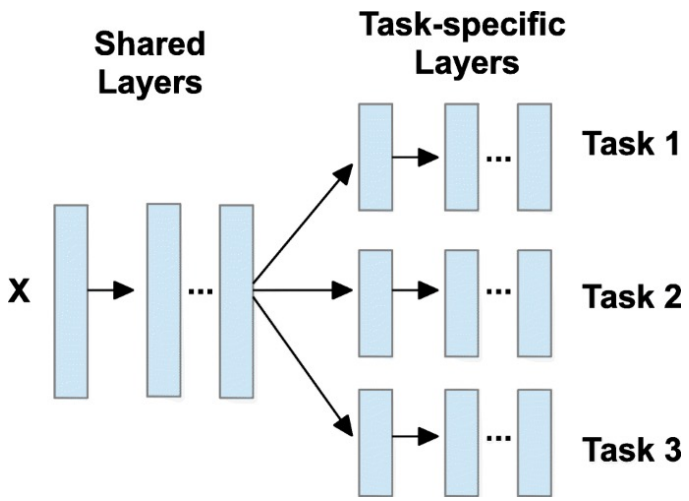
Abstract. Although deep networks achieve strong accuracy on a range of computer vision benchmarks, they remain vulnerable to adversarial attacks, where imperceptible input perturbations fool the network. We present both theoretical and empirical analyses that connect the adversarial robustness of a model to the number of tasks that it is trained on. Experiments on two datasets show that attack difficulty increases as the number of target tasks increase. Moreover, our results suggest that when models are trained on multiple tasks at once, they become more robust to adversarial attacks on individual tasks. While adversarial defense remains an open challenge, our results suggest that deep networks are vulnerable partly because they are trained on too few tasks.

Keywords: Multitask Learning, Adversarial Robustness

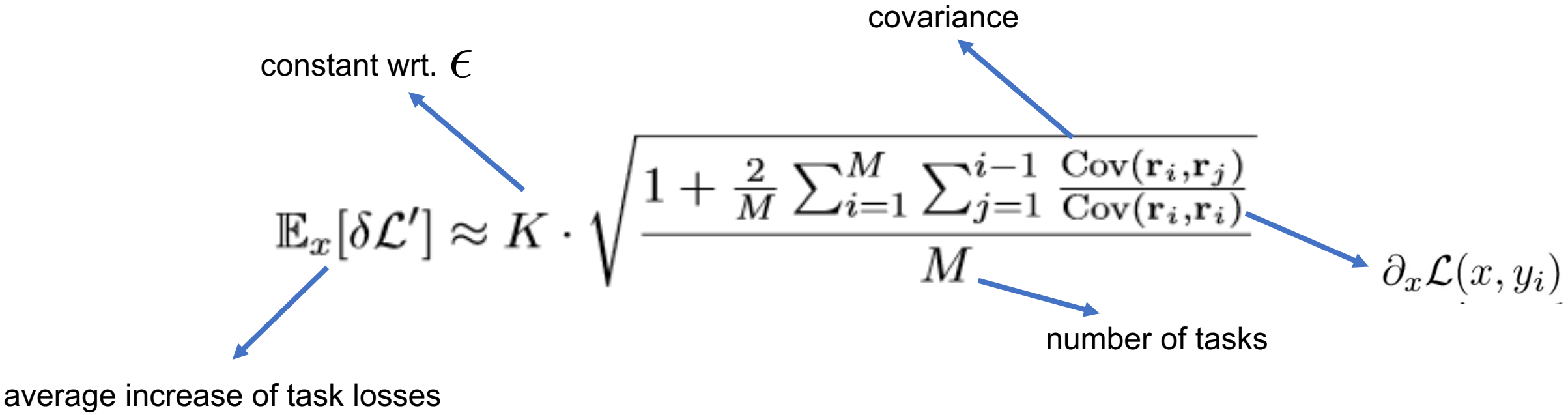


accuracy ++

robustness ++



Does learning multiple tasks increase robustness?



Conclusion: Robustness increases more as

- more tasks are learnt
- tasks are less correlated.

Experimental settings

- Dataset: **Taskonomy**
- Tasks: Semantic Segmentation (**s**), Depth z-buffer Estimation (**d**), Depth euclidian Estimation (**D**), Surface Normal Prediction (**n**), SURF Keypoint Detection in 2D (**k**) and 3D **(K),** Canny Edge Detection (**e**), Edge Occlusion (**E**), Principal Curvature (**p**), Reshading (**r**) and Auto-Encoders (**A**)
- Models: ResNet18 + custom task decoder (same as Taskonomy paper)
- Metrics: cross-entropy (segmentation)
- Evasion attack: **PGD** (25 steps, step size = 2/255, e = 8/255)

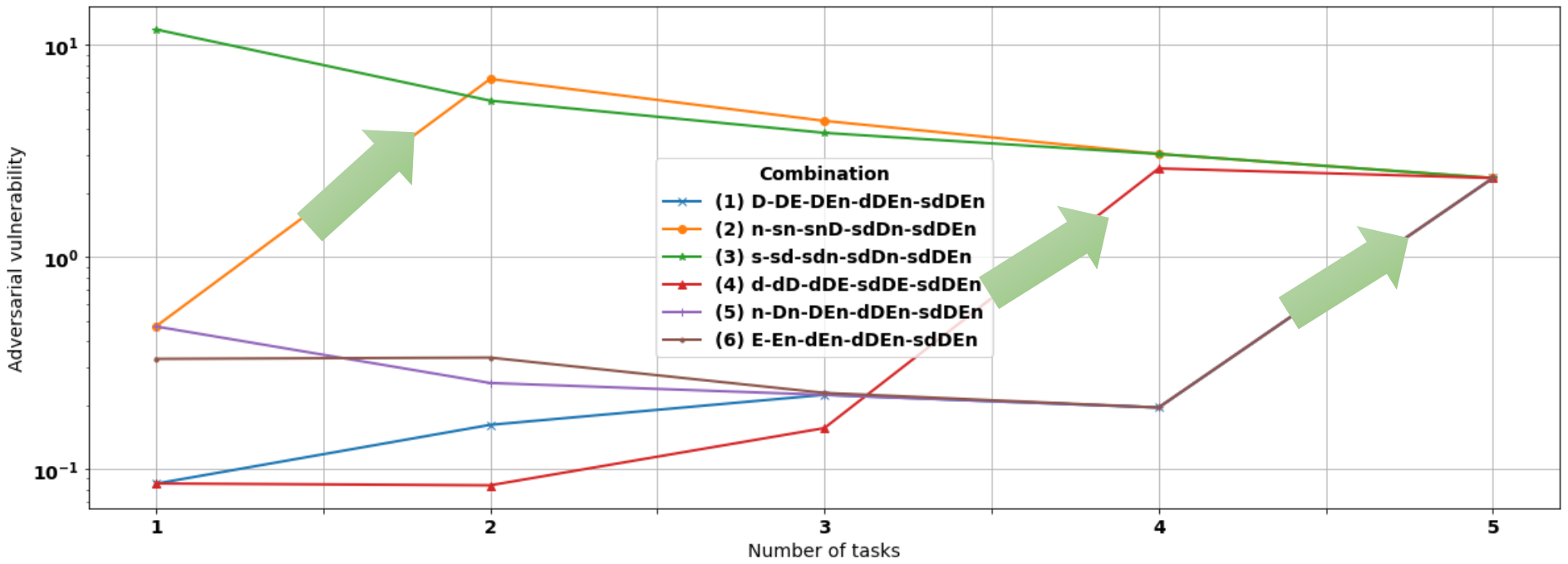
Is the robustness increase generalized?

Auxiliary →		s	d	D	n	E
Single	s	*0.82*	0.86	0.97	0.96	0.93
	d	5.74	*5.61*	5.28	6.88	6.41
	D	5.93	6.14	*6.4*	7.12	8.31
	n	7.43	9.48	8.93	*10.82*	9.08
	E	12.93	19.29	18.44	15.16	*22.57*

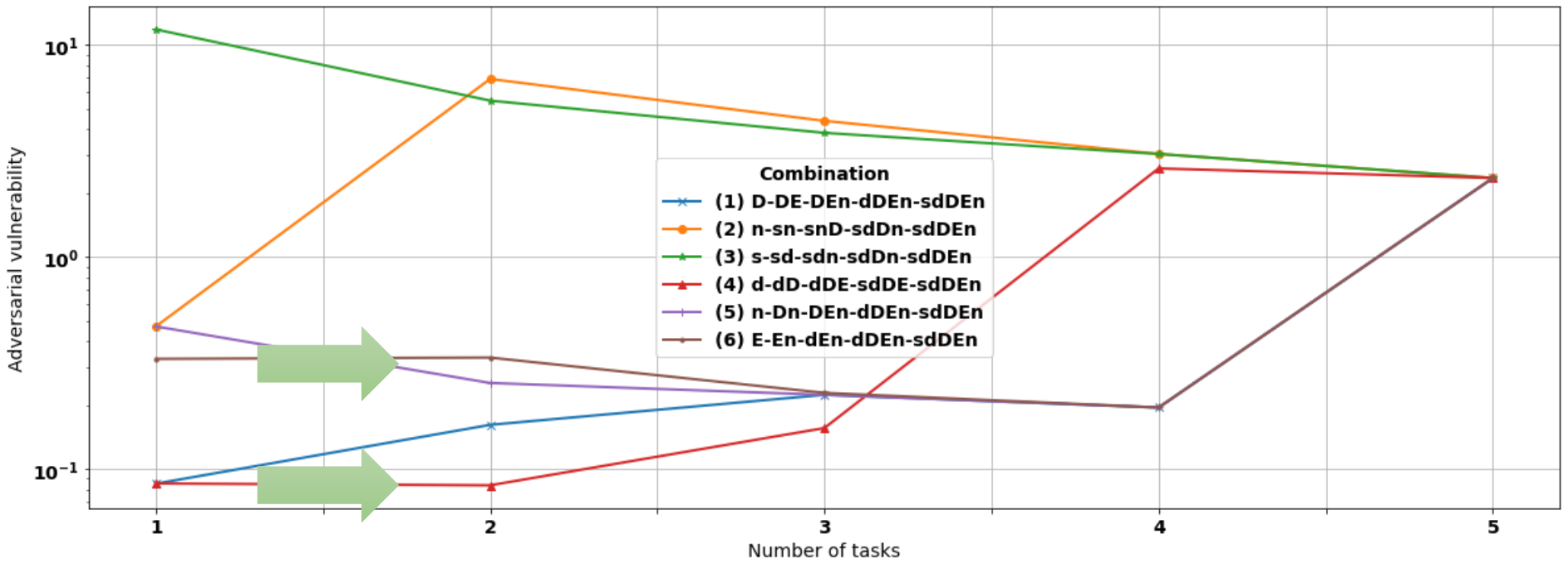
Adversarial vulnerability of a single main task after adding an auxiliary task (lower is better).

Diagonal elements represent models with a single task.

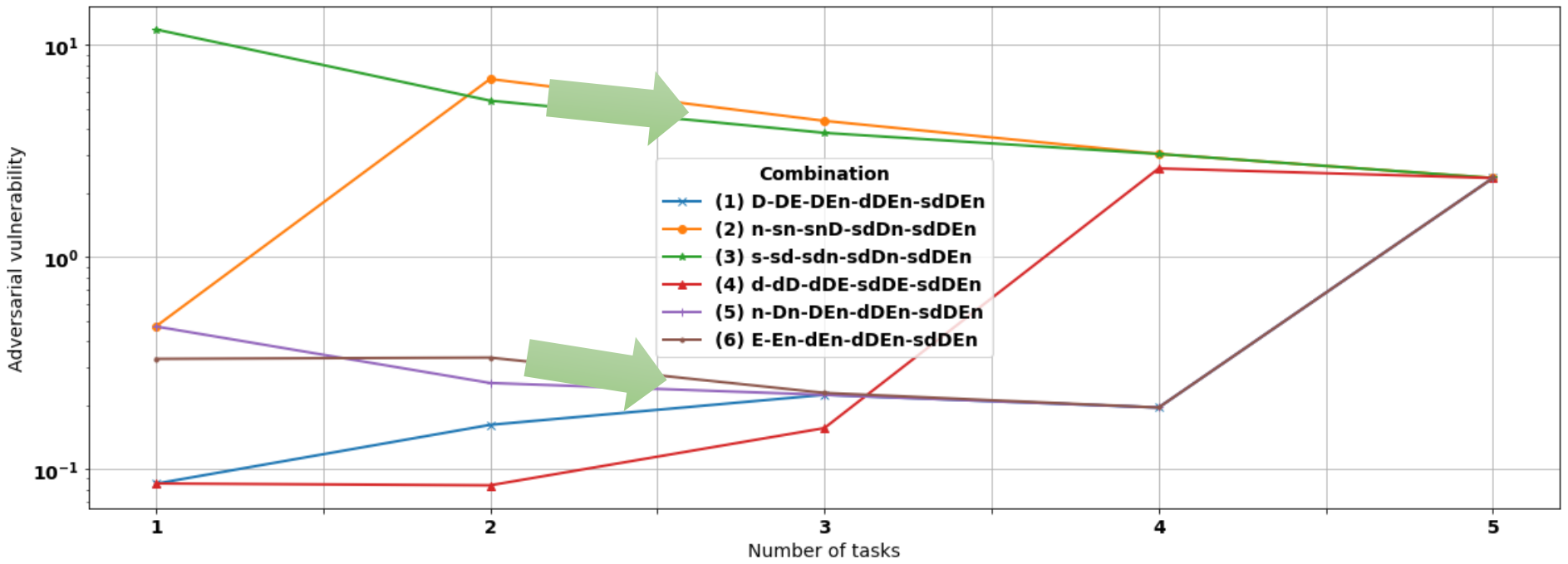
Marginal vulnerability of multi-task models



Marginal vulnerability of multi-task models



Marginal vulnerability of multi-task models



New theorem

vulnerability of the new task

$$\widetilde{\Delta_N \mathbb{E}_x[\delta \mathcal{L}']} \leq \epsilon \cdot (\mathbb{E}_x[|| \mathbf{r}_{N+1} ||] + \max_{i<N+1} \mathbb{E}_x[|| \mathbf{r}_i ||])$$

average loss increase
after adding task N+1

vulnerability of the
most vulnerable previous task

Conclusion: Robustness after addning a new task depends on

- the intrinsic vulnerability of the new task
- the most vulnerable previous task

Generalization to weighted tasks

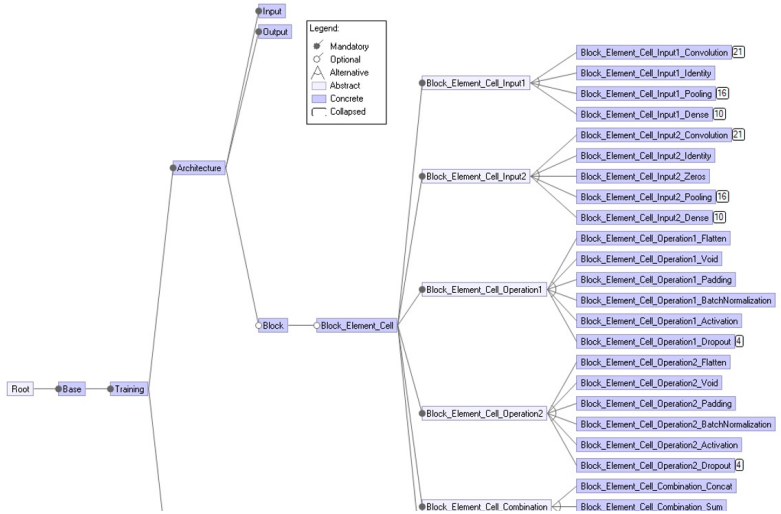
$$\widetilde{\Delta_N \mathbb{E}_x[\delta \mathcal{L}']} \leq \epsilon \cdot ((N+1) \cdot w_{N+1} \mathbb{E}_x[|| \mathbf{r}_{N+1} ||] + N \cdot \max_{i<N+1} w_i \mathbb{E}_x[|| \mathbf{r}_i ||])$$

learning weight of task i

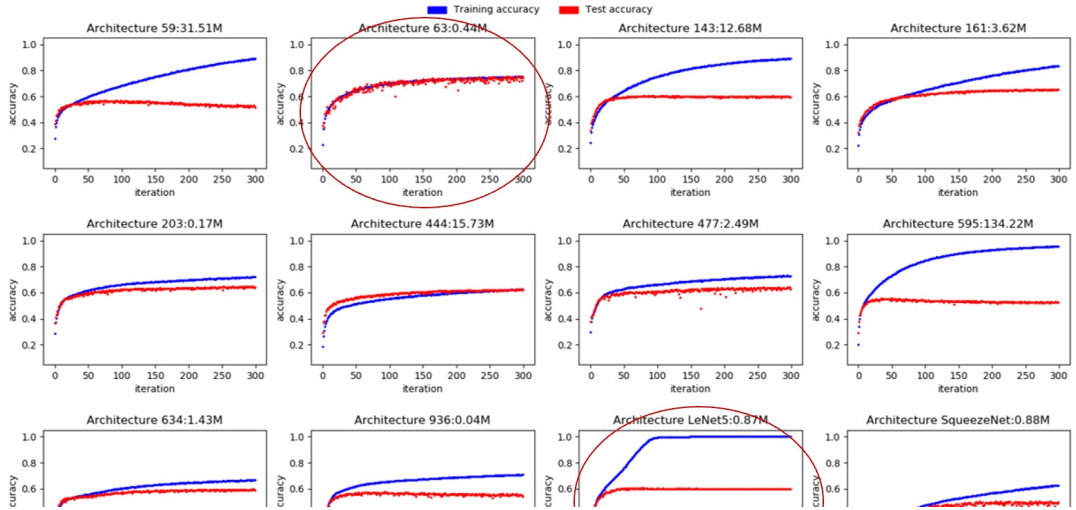
Conclusion: appropriate task weighting can increase robustness!

Conclusion

The corresponding variability model

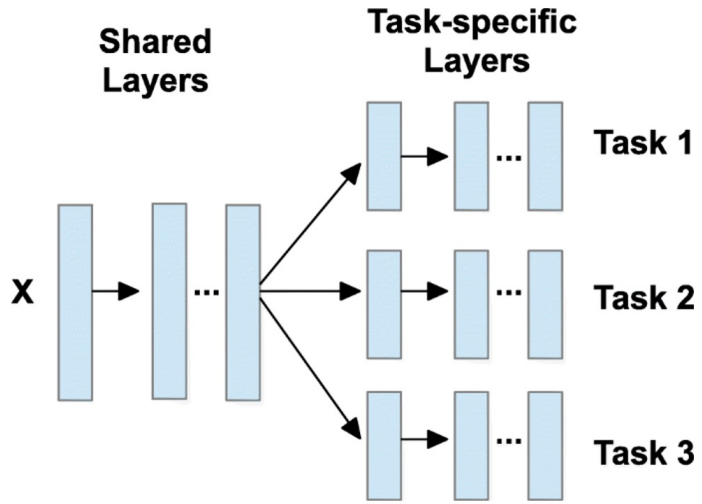


Variability in performance



Reusability, adaptability, and exploration (in ML)

Multi-task learning



Is the robustness increase generalized?

Auxiliary →		s	d	D	n	E
Single	s	*0.82*	0.86	0.97	0.96	0.93
	d	5.74	*5.61*	5.28	6.88	6.41
	D	5.93	6.14	*6.4*	7.12	8.31
	n	7.43	9.48	8.93	*10.82*	9.08
	E	12.93	19.29	18.44	15.16	*22.57*

Adversarial vulnerability of a single main task after adding an auxiliary task (lower is better).

Diagonal elements represent models with a single task.

THANK YOU

maxime.cordy@uni.lu