

Embedding AI into Constrained Devices

A Multi-Faceted Approach

PhD Student: *Andrea Agiollo**

Supervisor: Andrea Omicini*

Co-Supervisor: Daniele Turrin†

*Dipartimento di Informatica – Scienza e Ingegneria (DISI)

ALMA MATER STUDIORUM – Università di Bologna

†Electrolux Professional S.P.A.

andrea.agiollo@unibo.it

PhD Dissertation,
24th June 2024, Bologna



Outline

1 Context & Research Focus

2 Contributions

- Classic Efficientisation
 - Architecturing
 - Data coresets
- Efficientisation via Knowledge Integration
 - SKI for Efficientisation

3 Conclusions



Next in Line...

1 Context & Research Focus

2 Contributions

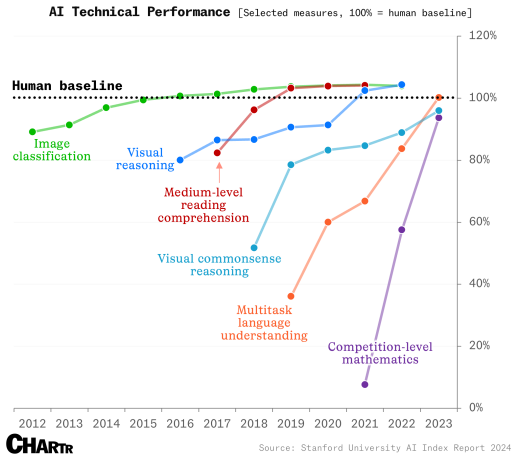
- Classic Efficientisation
 - Architecturing
 - Data coresets
- Efficientisation via Knowledge Integration
 - SKI for Efficientisation

3 Conclusions



Context

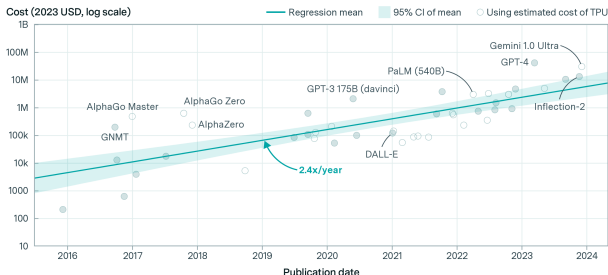
- AI proven successful for most diverse tasks and applications



Context

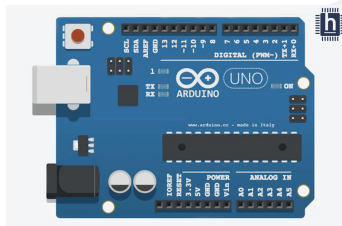
- AI proven successful for most diverse tasks and applications
- AI requires high computational capabilities

Amortized hardware and energy cost to train frontier AI models over time



Context

- AI proven successful for most diverse tasks and applications
- AI requires high computational capabilities
- Application of AI on restricted devices (e.g., IoT) open issue:
 - Minimize costs
 - Poor performances
 - Real-time results



Context

- AI proven successful for most diverse tasks and applications
- AI requires high computational capabilities
- Application of AI on restricted devices (e.g., IoT) open issue:
 - Minimize costs
 - Poor performances
 - Real-time results
- AI in IoT requested for industrial applications



Research Focus

Goal

Resource-friendly Artificial Intelligence



Research Focus

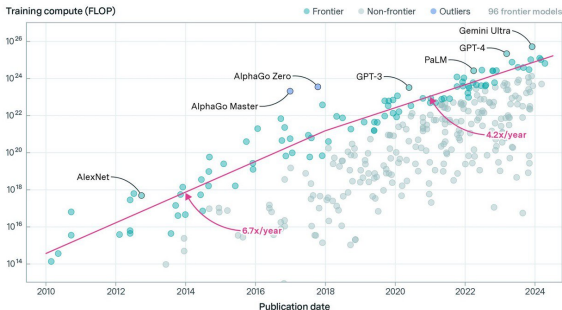
Goal

Resource-friendly Artificial Intelligence:

- *less computations*

Training compute of frontier models

EPOCH AI



Research Focus

Goal

Resource-friendly Artificial Intelligence:

- less *computations*
- less *energy*

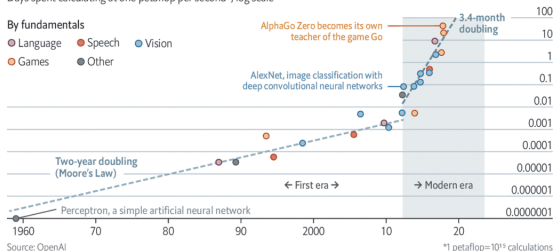
Deep and steep

Computing power used in training AI systems

Days spent calculating at one petaflop per second*, log scale

By fundamentals

- Language
- Speech
- Vision
- Games
- Other



Source: OpenAI

The Economist

Research Focus

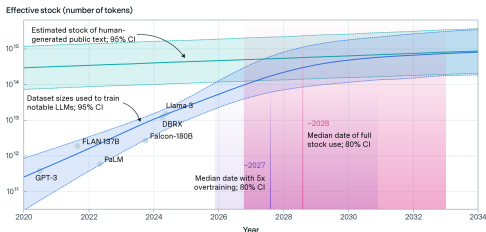
Goal

Resource-friendly Artificial Intelligence:

- less *computations*
- less *energy*
- less *data*

Projections of the stock of public text and data usage

EPOCH AI



Research Focus

Goal

Resource-friendly Artificial Intelligence:

- less *computations*
- less *energy*
- less *data*



Parallel approaches

- 1 overcome limitations of available efficientisation approaches
- 2 leverage integration of symbolic and sub-symbolic models (*NeSy*) to tackle efficientisation

Next in Line...

1 Context & Research Focus

2 Contributions

- Classic Efficientisation
 - Architecturing
 - Data coresets
- Efficientisation via Knowledge Integration
 - SKI for Efficientisation

3 Conclusions



Focus on. . .

- 1 Context & Research Focus
- 2 Contributions
 - **Classic Efficientisation**
 - Architecturing
 - Data coresets
 - Efficientisation via Knowledge Integration
 - SKI for Efficientisation
- 3 Conclusions



Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing**: identify/search powerful efficient structures [17].

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing:** identify/search powerful efficient structures [17].
 - 1 Agiollo et al. @ EXTRAAMAS 2021 [7]
 - 2 Agiollo et al. @ UAI 2022 [11]

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing:** identify/search powerful efficient structures [17].
 - 1 Agiollo et al. @ EXTRAAMAS 2021 [7]
 - 2 Agiollo et al. @ UAI 2022 [11]
- **Data coresets:** identify few relevant samples to train on [18].

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing:** identify/search powerful efficient structures [17].
 - ① Agiollo et al. @ EXTRAAMAS 2021 [7]
 - ② Agiollo et al. @ UAI 2022 [11]
- **Data coresets:** identify few relevant samples to train on [18].
 - ① Agiollo et al. @ KDD 2024 [9]

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing:** identify/search powerful efficient structures [17].
 - ① Agiollo et al. @ EXTRAAMAS 2021 [7]
 - ② Agiollo et al. @ UAI 2022 [11]
- **Data coresets:** identify few relevant samples to train on [18].
 - ① Agiollo et al. @ KDD 2024 [9]
- **Distributed learning:** split learning among several devices [29].

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing:** identify/search powerful efficient structures [17].
 - ① Agiollo et al. @ EXTRAAMAS 2021 [7]
 - ② Agiollo et al. @ UAI 2022 [11]
- **Data coresets:** identify few relevant samples to train on [18].
 - ① Agiollo et al. @ KDD 2024 [9]
- **Distributed learning:** split learning among several devices [29].
 - ① Agiollo et al. @ WOA 2023 [24]
 - ② Agiollo et al. @ FGCS 2024 [4]

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing**: identify/search powerful efficient structures [17].
 - ① Agiollo et al. @ EXTRAAMAS 2021 [7]
 - ② Agiollo et al. @ UAI 2022 [11]
- **Data coresets**: identify few relevant samples to train on [18].
 - ① Agiollo et al. @ KDD 2024 [9]
- **Distributed learning**: split learning among several devices [29].
 - ① Agiollo et al. @ WOA 2023 [24]
 - ② Agiollo et al. @ FGCS 2024 [4]
- **Pruning**: remove non-relevant substructures of NN [21].

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing:** identify/search powerful efficient structures [17].
 - ① Agiollo et al. @ EXTRAAMAS 2021 [7]
 - ② Agiollo et al. @ UAI 2022 [11]
- **Data coresets:** identify few relevant samples to train on [18].
 - ① Agiollo et al. @ KDD 2024 [9]
- **Distributed learning:** split learning among several devices [29].
 - ① Agiollo et al. @ WOA 2023 [24]
 - ② Agiollo et al. @ FGCS 2024 [4]
- **Pruning:** remove non-relevant substructures of NN [21].
- **Quantisation:** reduce representation precision [21].

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing:** identify/search powerful efficient structures [17].
 - ① Agiollo et al. @ EXTRAAMAS 2021 [7]
 - ② Agiollo et al. @ UAI 2022 [11]
- **Data coresets:** identify few relevant samples to train on [18].
 - ① Agiollo et al. @ KDD 2024 [9]
- **Distributed learning:** split learning among several devices [29].
 - ① Agiollo et al. @ WOA 2023 [24]
 - ② Agiollo et al. @ FGCS 2024 [4]
- **Pruning:** remove non-relevant substructures of NN [21].
- **Quantisation:** reduce representation precision [21].
- **Distillation:** couple NNs to force transfer of knowledge from large model to smaller one [20].

Efficientisation of NNs

Available approaches to tackle efficientisation of NN models:

- **Architecturing:** identify/search powerful efficient structures [17].
 - ① Agiollo et al. @ EXTRAAMAS 2021 [7]
 - ② Agiollo et al. @ UAI 2022 [11]
- **Data coresets:** identify few relevant samples to train on [18].
 - ① Agiollo et al. @ KDD 2024 [9]
- **Distributed learning:** split learning among several devices [29].
 - ① Agiollo et al. @ WOA 2023 [24]
 - ② Agiollo et al. @ FGCS 2024 [4]
- **Pruning:** remove non-relevant substructures of NN [21].
- **Quantisation:** reduce representation precision [21].
- **Distillation:** couple NNs to force transfer of knowledge from large model to smaller one [20].

Architecturing: GNN2GNN

Motivations

- NAS automatically search optimal NN structures
- NAS learn *to search* rather than *to build*
- NAS lack in flexibility and generality



Architecturing: GNN2GNN

Motivations

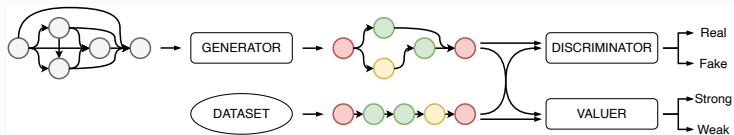
- NAS automatically search optimal NN structures
- NAS learn *to search* rather than *to build*
- NAS lack in flexibility and generality

Contributions

- GNN2GNN, the first GNN-based framework to generate powerful NN architectures, without inefficient search
- Show GNN2GNN's effectiveness over two state-of-the-art datasets, highlighting flexibility and generalization capability

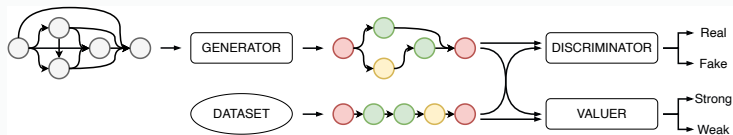
Architecting: GNN2GNN Framework

Three-way generative adversarial framework



Architecturing: GNN2GNN Framework

Three-way generative adversarial framework



$$\mathcal{L}_G = \underbrace{\lambda \cdot \log(1 - D(G(z)))}_{\text{standard GAN loss}} + (1 - \lambda) \cdot \underbrace{\mathcal{L}_R(V(G(z)))}_{\text{reward loss}}$$

Architecturing: GNN2GNN Experiments & Results

GNN2GNN vs GANs

Dataset	Model	Novelty	Top-5	Top-10	Top-50	Acc ₉₀	Acc
NAS101	GNN2GNN	82.60%	45.91%	52.21%	57.37%	81.79%	92.04%
	MOLGAN	65.41%	22.63%	27.29%	45.20%	59.71%	89.39%
	RNN	96.34%	1.69%	2.32%	4.81%	53.04%	89.32%
	Random	51.81%	11.17%	13.74%	28.43%	43.18%	88.54%



Architecturing: GNN2GNN Experiments & Results

GNN2GNN vs GANs

Dataset	Model	Novelty	Top-5	Top-10	Top-50	Acc ₉₀	Acc
NAS101	GNN2GNN	82.60%	45.91%	52.21%	57.37%	81.79%	92.04%
	MOLGAN	65.41%	22.63%	27.29%	45.20%	59.71%	89.39%
	RNN	96.34%	1.69%	2.32%	4.81%	53.04%	89.32%
	Random	51.81%	11.17%	13.74%	28.43%	43.18%	88.54%

Quality Degradation

Dataset	N	Novelty	Top-5	Top-10	Top-50	Acc ₉₀	Acc
NAS101	10%	82.60%	45.91%	52.21%	57.37%	81.79%	92.04%
	20%	83.01%	45.54%	52.32%	58.48%	82.05%	91.98%
	30%	83.67%	46.10%	51.07%	57.04%	80.14%	91.68%
	40%	84.89%	45.80%	51.03%	58.71%	81.03%	91.55%
	50%	85.00%	44.01%	48.81%	56.30%	79.24%	91.33%
	60%	84.58%	44.40%	49.12%	57.72%	80.30%	91.30%
	70%	84.20%	44.66%	49.38%	56.71%	79.52%	91.37%
	80%	84.33%	43.90%	48.27%	55.51%	78.16%	91.27%
	90%	85.71%	42.77%	46.70%	55.72%	78.24%	91.22%



Architecturing: GNN2GNN Experiments & Results

GNN2GNN vs GANs

Dataset	Model	Novelty	Top-5	Top-10	Top-50	Acc ₉₀	Acc
NAS101	GNN2GNN	82.60%	45.91%	52.21%	57.37%	81.79%	92.04%
	MOLGAN	65.41%	22.63%	27.29%	45.20%	59.71%	89.39%
	RNN	96.34%	1.69%	2.32%	4.81%	53.04%	89.32%
	Random	51.81%	11.17%	13.74%	28.43%	43.18%	88.54%

Quality Degradation

Dataset	N	Novelty	Top-5	Top-10	Top-50	Acc ₉₀	Acc
NAS101	10%	82.60%	45.91%	52.21%	57.37%	81.79%	92.04%
	20%	83.01%	45.54%	52.32%	58.48%	82.05%	91.98%
	30%	83.67%	46.10%	51.07%	57.04%	80.14%	91.68%
	40%	84.89%	45.80%	51.03%	58.71%	81.03%	91.55%
	50%	85.00%	44.01%	48.81%	56.30%	79.24%	91.33%
	60%	84.58%	44.40%	49.12%	57.72%	80.30%	91.30%
	70%	84.20%	44.66%	49.38%	56.71%	79.52%	91.37%
	80%	84.33%	43.90%	48.27%	55.51%	78.16%	91.27%
	90%	85.71%	42.77%	46.70%	55.72%	78.24%	91.22%

Generalisation

Dataset		Novelty	Top-5	Top-10	Top-50	Acc
Train	Test					
NATS _{C10}	NATS _{C10}	76.73%	1.65%	3.39%	15.61%	68.91%
NAS101	NATS _{C10}	73.67%	2.64%	5.08%	16.55%	70.25%
NATS _{C100}	NATS _{C100}	71.71%	1.30%	3.28%	13.10%	33.03%
NAS101	NATS _{C100}	72.03%	2.64%	4.82%	16.90%	35.40%
NATS _I	NATS _I	81.03%	0.91%	2.17%	8.42%	16.75%
NAS101	NATS _I	82.30%	1.93%	3.64%	11.49%	18.84%

Architecturing: GNN2GNN Experiments & Results

GNN2GNN vs GANs

Dataset	Model	Novelty	Top-5	Top-10	Top-50	Acc ₉₀	Acc
NAS101	GNN2GNN	82.60%	45.91%	52.21%	57.37%	81.79%	92.04%
	MOLGAN	65.41%	22.63%	27.29%	45.20%	59.71%	89.39%
	RNN	96.34%	1.69%	2.32%	4.81%	53.04%	89.32%
	Random	51.81%	11.17%	13.74%	28.43%	43.18%	88.54%

Quality Degradation

Dataset	N	Novelty	Top-5	Top-10	Top-50	Acc ₉₀	Acc
NAS101	10%	82.60%	45.91%	52.21%	57.37%	81.79%	92.04%
	20%	83.01%	45.54%	52.32%	58.48%	82.05%	91.98%
	30%	83.67%	46.10%	51.07%	57.04%	80.14%	91.68%
	40%	84.89%	45.80%	51.03%	58.71%	81.03%	91.55%
	50%	85.00%	44.01%	48.81%	56.30%	79.24%	91.33%
	60%	84.58%	44.40%	49.12%	57.72%	80.30%	91.30%
	70%	84.20%	44.66%	49.38%	56.71%	79.52%	91.37%
	80%	84.33%	43.90%	48.27%	55.51%	78.16%	91.27%
	90%	85.71%	42.77%	46.70%	55.72%	78.24%	91.22%

Generalisation

Dataset		Novelty	Top-5	Top-10	Top-50	Acc
Train	Test					
NATS _{C10}	NATS _{C10}	76.73%	1.65%	3.39%	15.61%	68.91%
NAS101	NATS _{C10}	73.67%	2.64%	5.08%	16.55%	70.25%
NATS _{C100}	NATS _{C100}	71.71%	1.30%	3.28%	13.10%	33.03%
NAS101	NATS _{C100}	72.03%	2.64%	4.82%	16.90%	35.40%
NATS _I	NATS _I	81.03%	0.91%	2.17%	8.42%	16.75%
NAS101	NATS _I	82.30%	1.93%	3.64%	11.49%	18.84%

GNN2GNN vs NAS

Model	Acc	Best Acc	Best Rank
DARTS [22]	92.21%	93.02%	57079
NAO [23]	92.59%	93.33%	19552
ENAS [26]	91.83%	92.54%	96939
GNN2GNN ₉₀	92.04%	94.32%	5372
GNN2GNN ₆₀	91.68%	94.18%	7843
GNN2GNN ₃₀	91.30%	94.01%	9371
GNN2GNN ₁₀	91.22%	93.69%	12570

Data coresets: SAMIS

Motivations

- dataset pruning and summarization to identify small subset of highly informative data to train on [25, 28]
- empirical findings suggest relationship between sample complexity and relevance for data efficiency [27, 19]



Data coresets: SAMIS

Motivations

- dataset pruning and summarization to identify small subset of highly informative data to train on [25, 28]
- empirical findings suggest relationship between sample complexity and relevance for data efficiency [27, 19]

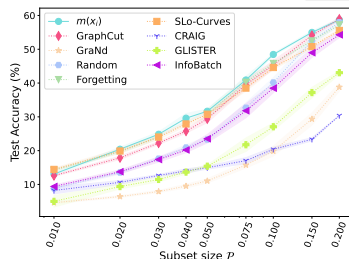
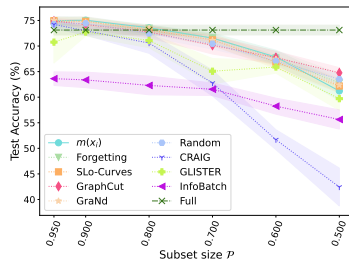
Contributions

- memorization scores used for data summarization
- leverage discrepancy between flat-seeking minimizer and SGD to approximate memorization
- memorization proxies used for data summarization, overcoming state-of-the-art

Data coresets: SAMIS Findings

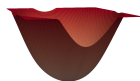
- memorization scores for data coresets outperform state-of-the-art

$$m(\mathbf{x}_i) = P \left[h_{\mathcal{T}}^A(\mathbf{x}_i) = y_i \right] - P \left[h_{\mathcal{T} \setminus i}^A(\mathbf{x}_i) = y_i \right]$$

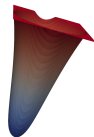


Data coresets: SAMIS Findings

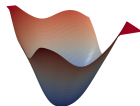
- memorization scores for data coresets outperform state-of-the-art
- minima flatness depends on data cleanliness



CIFAR100 **clean** samples



CIFAR100 **cumbersome** samples



CIFAR10 **clean** samples

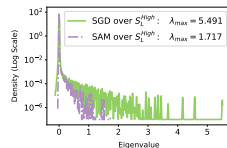
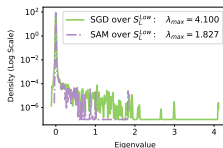


CIFAR10 **cumbersome** samples

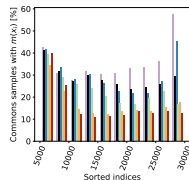
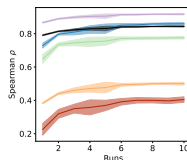
Data coresets: SAMIS Findings

- memorization scores for data coresets outperform state-of-the-art
- minima flatness depends on data cleanliness
- using minima flatness we can approximate memorization: **SAMIS** (SAM - SGD)

$$S_L(\mathbf{x}_i) = |\mathcal{L}(h_{S \setminus i, \dots, z}^{SAM}(\mathbf{x}_i)) - \mathcal{L}(h_{S \setminus i, \dots, z}^{SGD}(\mathbf{x}_i))|$$



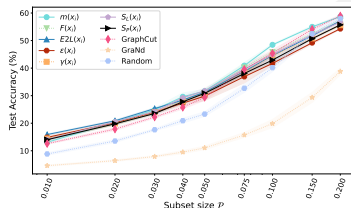
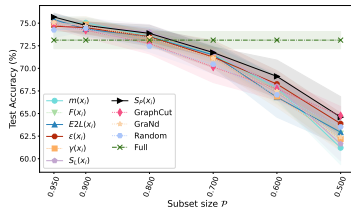
SAM vs. SGD flatness



SAMIS correlates well with memorization

Data coresets: SAMIS Findings

- memorization scores for data coresets outperform state-of-the-art
- minima flatness depends on data cleanliness
- using minima flatness we can approximate memorization: **SAMIS** (SAM - SGD)
- SAMIS for data coresets mimicks memorization and outperform state-of-the-art



SAMIS outperforms sota

Focus on. . .

- 1 Context & Research Focus
- 2 Contributions
 - Classic Efficientisation
 - Architecturing
 - Data coresets
 - **Efficientisation via Knowledge Integration**
 - SKI for Efficientisation
- 3 Conclusions



Combining Symbolic and Sub-Symbolic in AI

Idea

Hybridisation of sub-symbolic – e.g., NNs – AI mechanisms with symbolic a-priori or expert knowledge allows to produce more efficient NNs.

Motivations

- logical information complicated to extract, elaborate, treat with NNs
- symbolic knowledge is more reliable than data-driven approaches
- injecting logical knowledge into NNs removes part of the learning burden from NNs

Combining Symbolic and Sub-Symbolic in AI

There exist multiple approaches for combining symbolic and sub-symbolic in AI, with different aims:

- Symbolic Knowledge Injection (**SKI**): integrating symbolic rules into a NN



Combining Symbolic and Sub-Symbolic in AI

There exist multiple approaches for combining symbolic and sub-symbolic in AI, with different aims:

- Symbolic Knowledge Injection (**SKI**): integrating symbolic rules into a NN
 - 1 Agiollo et al. @ WOA 2022 [1]
 - 2 Agiollo et al. @ JAAMAS 2023 [13]



Combining Symbolic and Sub-Symbolic in AI

There exist multiple approaches for combining symbolic and sub-symbolic in AI, with different aims:

- Symbolic Knowledge Injection (**SKI**): integrating symbolic rules into a NN
 - 1 Agiollo et al. @ WOA 2022 [1]
 - 2 Agiollo et al. @ JAAMAS 2023 [13]
- Symbolic Knowledge Extraction (**SKE**): gathering symbolic rules from a NN



Combining Symbolic and Sub-Symbolic in AI

There exist multiple approaches for combining symbolic and sub-symbolic in AI, with different aims:

- Symbolic Knowledge Injection (**SKI**): integrating symbolic rules into a NN
 - 1 Agiollo et al. @ WOA 2022 [1]
 - 2 Agiollo et al. @ JAAMAS 2023 [13]
- Symbolic Knowledge Extraction (**SKE**): gathering symbolic rules from a NN
 - 1 Agiollo et al. @ EXTRAAMAS 2023 [5]
 - 2 Agiollo et al. @ JAAMAS 2024 [14]



Combining Symbolic and Sub-Symbolic in AI

There exist multiple approaches for combining symbolic and sub-symbolic in AI, with different aims:

- **Symbolic Knowledge Injection (SKI):** integrating symbolic rules into a NN
 - 1 Agiollo et al. @ WOA 2022 [1]
 - 2 Agiollo et al. @ JAAMAS 2023 [13]
- **Symbolic Knowledge Extraction (SKE):** gathering symbolic rules from a NN
 - 1 Agiollo et al. @ EXTRAAMAS 2023 [5]
 - 2 Agiollo et al. @ JAAMAS 2024 [14]



SKI for Efficientisation

Motivations

- SKI is most beneficial integration approach for NNs efficiency
- no SKI approach study its efficiency benefits



SKI for Efficientisation

Motivations

- SKI is most beneficial integration approach for NNs efficiency
- no SKI approach study its efficiency benefits

Contributions

- definition of *Quality-of-Service* metrics for SKI
- analysis of SKI efficientisation effects using QoS metrics
- analysis of SKI robustness and resiliency

SKI for Efficientisation: Findings

- *Quality-of-Service* metrics for SKI

- memory footprint:

$$\mu(\mathcal{I}) = \Psi(N) - \Psi(\mathcal{I}(K, N))$$

- energy consumption:

$$\varepsilon^i(\mathcal{I}) = \Upsilon_v^i(N, T) - \Upsilon_v^i(\mathcal{I}(K, N), T)$$

$$\varepsilon^t(\mathcal{I}) = \Upsilon_{v,\gamma}^t(e, N, T) - \Upsilon_{v,\gamma}^t(e, \mathcal{I}(K, N), T)$$

- latency:

$$\lambda(\mathcal{I}) = \frac{1}{|T|} \sum_{t \in T} (\Theta(N, t) - \Theta(\hat{N}, t))$$

- data efficiency:

$$\delta(\mathcal{I}) = \Delta_{\pi}(e, N, D, T) - \Delta_{\pi}(e, \mathcal{I}(K, N), D', T)$$



SKI for Efficientisation: Findings

- *Quality-of-Service* metrics for SKI
- our QoS metrics show how **SKI brings data efficiency benefits**

Dataset	Model	Set	Data efficiency (KB)	Energy (mWh)	Memory (FLOPs)	Latency (ms)	Accuracy (%)
BCW	uneducated		–	–	–	–	94.53
	KBANN	train test	35.89	-1.47 -0.10	3933	-1.70	95.45
	KILL	train test	4.09	-0.99 0	0	0.35	94.63
	KINS	train test	-9.97	-1.22 -0.09	-559	-1.41	94.29
PSJGS	uneducated		–	–	–	–	93.91
	KBANN	train test	-4946.81	-4.67 -0.22	-66944	-2.56	92.84
	KILL	train test	553.89	-3 0	0	0.04	94.02
	KINS	train test	-954.80	-6.53 -0.51	-161779	-4.75	93.70
CI	uneducated		–	–	–	–	84.63
	KBANN	train test	1653.79	-1.41 -0.02	-2468	-0.43	84.78
	KILL	train test	4016.90	-0.70 0	4200	0	84.81
	KINS	train test	4263.50	-1.41 -0.02	-6220	-0.44	84.77

SKI for Efficientisation: Findings

- *Quality-of-Service* metrics for SKI
- our QoS metrics show how **SKI brings data efficiency benefits**
- define SKI robustness metric

$$\rho_{N,D}(\mathbf{D}) = \frac{1}{n} \sum_{\Delta D \in \mathbf{D}} \|\Delta D\| \cdot \frac{\pi(N_{\Delta D}, D \circ \Delta D)}{\pi(N, D)}$$

$$R_{N,D}(\mathcal{I}) = \frac{\rho_{\hat{N},D}(\mathbf{D})}{\rho_{N,D}(\mathbf{D})}$$



SKI for Efficientisation: Findings

- *Quality-of-Service* metrics for SKI
- our QoS metrics show how **SKI brings data efficiency benefits**
- define SKI robustness metric
- robustness metric confirms how **SKI brings data efficiency benefits**

Dataset	$R_{N,D}(\mathcal{I})$ drop			$R_{N,D}(\mathcal{I})$ noise			$R_{N,D}(\mathcal{I})$ flip		
	KINS	KILL	KBANN	KINS	KILL	KBANN	KINS	KILL	KBANN
BCW	1.0493	1.0318	1.0382	0.9960	0.9985	1.0109	0.9994	1.0184	0.9520
PSJGS	1.0045	0.9968	0.8425	0.9950	0.9984	1.0145	0.9962	1.0026	1.6749
CI	0.9998	1.0039	1.0043	0.9992	1.0012	0.9965	0.9897	1.1703	0.9815

Next in Line...

1 Context & Research Focus

2 Contributions

- Classic Efficientisation
 - Architecturing
 - Data coresets
- Efficientisation via Knowledge Integration
 - SKI for Efficientisation

3 Conclusions



Conclusions

Research findings summary

- traditional *efficientisation* approaches have some issues and limitations

Conclusions

Research findings summary

- traditional *efficientisation* approaches have some issues and limitations
- overcome NAS focus on inefficient *search* via learning to *build*

Conclusions

Research findings summary

- traditional *efficientisation* approaches have some issues and limitations
- overcome NAS focus on inefficient *search* via learning to *build*
- leverage NN *memorization* property help in data-efficient settings

Conclusions

Research findings summary

- traditional *efficientisation* approaches have some issues and limitations
- overcome NAS focus on inefficient *search* via learning to *build*
- leverage NN *memorization* property help in data-efficient settings
- find *efficient memorization proxy* measuring discrepancy between flat-seeking and standard optimizers

Conclusions

Research findings summary

- traditional *efficientisation* approaches have some issues and limitations
- overcome NAS focus on inefficient *search* via learning to *build*
- leverage NN *memorization* property help in data-efficient settings
- find *efficient memorization proxy* measuring discrepancy between flat-seeking and standard optimizers
- NeSy techniques may help towards NN efficientisation

Conclusions

Research findings summary

- traditional *efficientisation* approaches have some issues and limitations
- overcome NAS focus on inefficient *search* via learning to *build*
- leverage NN *memorization* property help in data-efficient settings
- find *efficient memorization proxy* measuring discrepancy between flat-seeking and standard optimizers
- *NeSy* techniques may help towards NN efficientisation
- *SKI* brings data efficiency benefits reducing data for train

List of Published Papers I

Journals:

- ❶ *A. Agiollo, L. Siebert, P. Murukannaiah, A. Omicini: **From Large Language Models to Small Logic Programs: Building Global Explanations from Disagreeing Local Post-hoc Explainers.** In *Autonomous Agents and Multi-Agent Systems* (2024) [14].*
- ❷ *G. Ciatto, F. Sabbatini, A. Agiollo, M. Magnini, A. Omicini: **Symbolic Knowledge Extraction and Injection with Sub-symbolic Predictors: A Systematic Literature Review.** In *ACM Computing Surveys* (2024) [16].*
- ❸ *A. Agiollo, P. Bellavista, M. Mendula, A. Omicini: **EneA-FL: Energy-aware Orchestration for Serverless Federated Learning.** In *Future Generation Computer Systems* (2024) [4].*

List of Published Papers II

- 4 A. Agiollo, E. Bardhi, M. Conti, N. Dal Fabbro, R. Lazzeretti: **Anonymous Federated Learning via Named-Data Networking.** In *Future Generation Computer Systems* (2024) [2].
- 5 A. Agiollo, A. Rafanelli, M. Magnini, G. Ciatto, A. Omicini: **Symbolic knowledge injection meets intelligent agents: QoS metrics and experiments.** In *Autonomous Agents and Multi-Agent Systems* (2023) [13].
- 6 A. Agiollo, M. Conti, P. Kaliyar, T.N. Lin, L. Pajola: **DETONAR: Detection of routing attacks in RPL-based IoT.** In *IEEE Transaction on Network Service Management* (2021) [8].
- 7 A. Agiollo, A. Omicini: **Load Classification: A Case Study for Applying Neural Networks in Hyper-Constrained Embedded Devices.** In *Special Issue "Artificial Intelligence and Data Engineering in Engineering Applications"* (2021) [10].

List of Published Papers III

Conferences:

- ❶ A. Agiollo, Y. I. Kim, R. Khanna: **Approximating Memorization Using Loss Surface Geometry for Dataset Pruning and Summarization** In *KDD 2024 - SIGKDD Conference on Knowledge Discovery and Data Mining* [9].
- ❷ A. Agiollo, A. Omicini: **Measuring Trustworthiness in Neuro-Symbolic Integration**. In *FedCSIS 2023 - 18th Conference on Computer Science and Intelligence Systems* [12].
- ❸ A. Agiollo, E. Bardhi, M. Conti, R. Lazzeretti, E. Losiouk, A. Omicini: **GNN4IFA: Interest Flooding Attack Detection With Graph Neural Networks**. In *EuroS&P 2023 - 8th European Symposium on Security and Privacy* [3].

List of Published Papers IV

- ④ A. Agiollo, A. Omicini: **GNN2GNN: Graph Neural Networks to Generate Neural Networks**. In *UAI 2022 - 38th Conference on Uncertainty in Artificial Intelligence* [11].

Workshops:

- ① A. Agiollo, L. C. Siebert, P. K. Murukannaiah, A. Omicini: **The Quarrel of Local Post-hoc Explainers for Moral Values Classification in Natural Language Processing**. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems (2023)* [5].
- ② V. Bull, N. Finne, A. Agiollo, P. Kaliyar, L. Pajola, T. Voigt, M. Conti: **DETONAR-Light: An IoT Network Intrusion Detection Using DETONAR without a Sniffer Network**. In *ADIoT 2023 - 6th International Workshop on Attacks and Defenses for Internet-of-Things* [15].

List of Published Papers V

- ③ *M. Passeri, A. Agiollo, and A. Omicini: **Peer-Reviewed Federated Learning**. In WOA 2023 – 24th Workshop “From Objects to Agents” [24].*
- ④ *A. Agiollo, A. Rafanelli, A. Omicini: **Towards quality-of-service metrics for symbolic knowledge injection**. In WOA 2022 – 23rd Workshop “From Objects to Agents” [1].*
- ⑤ *A. Agiollo, G. Ciatto, A. Omicini: **Shallow2Deep: Restraining Neural Networks Opacity Through Neural Architecture Search**. In International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems (2021) [7].*
- ⑥ *A. Agiollo, G. Ciatto, A. Omicini: **Graph Neural Networks as the Copula Mundi between Logic and Machine Learning: A Roadmap**. In WOA 2021 – 22nd Workshop “From Objects to Agents” [6].*

Conclusions

Special Thanks

- *Electrolux Professional S.P.A.*
- my supervisor Prof. Andrea Omicini
- Prof. Rajiv Khanna for hosting me at *Purdue University*
- Prof. Catholijn Jonker for hosting me at *TU Delft*
- my girlfriend, family and friends for the constant support



Embedding AI into Constrained Devices

A Multi-Faceted Approach

PhD Student: *Andrea Agiollo**

Supervisor: Andrea Omicini*

Co-Supervisor: Daniele Turrin[†]

*Dipartimento di Informatica – Scienza e Ingegneria (DISI)

ALMA MATER STUDIORUM – Università di Bologna

[†]Electrolux Professional S.P.A.

andrea.agiollo@unibo.it

PhD Dissertation,
24th June 2024, Bologna



References I

- [1] Andrea Agiollo, Rafanelli Andrea, Andrea Omicini, et al.
Towards quality-of-service metrics for symbolic knowledge injection.
In *CEUR WORKSHOP PROCEEDINGS*, volume 3261, pages 30–47. Sun SITE Central Europe, RWTH Aachen University, 2022.
- [2] Andrea Agiollo, Enkeleda Bardhi, Mauro Conti, Nicolò Dal Fabbro, and Riccardo Lazzeretti.
Anonymous federated learning via named-data networking.
Future Generation Computer Systems, 2023.
- [3] Andrea Agiollo, Enkeleda Bardhi, Mauro Conti, Riccardo Lazzeretti, Eleonora Losiouk, and Andrea Omicini.
Gnn4ifa: Interest flooding attack detection with graph neural networks.
In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 615–630. IEEE, 2023.
- [4] Andrea Agiollo, Paolo Bellavista, Matteo Mendula, and Andrea Omicini.
EneA-FL: Energy-aware orchestration for serverless federated learning.
Future Generation Computer Systems, 154:219–234, May 2024.
Special Issue “Serverless Computing in the Cloud-to-Edge Continuum”.
- [5] Andrea Agiollo, Luciano Cavalcante Siebert, Pradeep Kumar Murukannaiah, and Andrea Omicini.
The quarrel of local post-hoc explainers for moral values classification in natural language processing.
In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 97–115. Springer, 2023.
- [6] Andrea Agiollo, Giovanni Ciatto, and Andrea Omicini.
Graph neural networks as the copula mundi between logic and machine learning: A roadmap.
In Roberta Calegari, Giovanni Ciatto, Enrico Denti, Andrea Omicini, and Giovanni Sartor, editors, *WOA 2021 – 22nd Workshop “From Objects to Agents”*, volume 2963 of *CEUR Workshop Proceedings*, pages 98–115. Sun SITE Central Europe, RWTH Aachen University, October 2021.
22nd Workshop “From Objects to Agents” (WOA 2021), Bologna, Italy, 1–3 September 2021. Proceedings.

References II

- [7] Andrea Agiollo, Giovanni Ciatto, and Andrea Omicini.
Shallow2Deep: Restraining neural networks opacity through neural architecture search.
In Davide Calvaresi, Amro Najjar, Michael Winikoff, and Kary Främling, editors, *Explainable and Transparent AI and Multi-Agent Systems. Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3–7, 2021, Revised Selected Papers*, volume 12688 of *Lecture Notes in Computer Science*, pages 63–82. Springer Nature, Basel, Switzerland, 2021.
- [8] Andrea Agiollo, Mauro Conti, Pallavi Kaliyar, Tsung-Nan Lin, and Luca Pajola.
DETONAR: detection of routing attacks in rpl-based iot.
IEEE Trans. Netw. Serv. Manag., 18(2):1178–1190, 2021.
- [9] Andrea Agiollo, Young In Kim, and Rajiv Khanna.
Approximating memorization using loss surface geometry for dataset pruning and summarization.
In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2024.
- [10] Andrea Agiollo and Andrea Omicini.
Load classification: A case study for applying neural networks in hyper-constrained embedded devices.
Applied Sciences, 11(24), December 2021.
Special Issue “Artificial Intelligence and Data Engineering in Engineering Applications”.
- [11] Andrea Agiollo and Andrea Omicini.
Gnn2gnn: Graph neural networks to generate neural networks.
In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.
- [12] Andrea Agiollo and Andrea Omicini.
Measuring trustworthiness in neuro-symbolic integration.
In *Proceedings of the 18th Conference on Computer Science and Intelligence Systems, FedCSIS 2023, Warsaw, Poland, September 17-20, 2023*, volume 35 of *Annals of Computer Science and Information Systems*, pages 1–10, 2023.

References III

- [13] Andrea Agiollo, Andrea Rafanelli, Matteo Magnini, Giovanni Ciatto, and Andrea Omicini. Symbolic knowledge injection meets intelligent agents: Qos metrics and experiments. *Autonomous Agents and Multi-Agent Systems*, 37(2):27, 2023.
- [14] Andrea Agiollo, Luciano Cavalcante Siebert, Pradeep Murukannaiah, and Andrea Omicini. From large language models to small logic programs: Building global explanations from disagreeing local post-hoc explainers. *Autonomous Agents and Multi-Agent Systems*, 2024. Under submission.
- [15] Victoria Bull, Niclas Finne, Andrea Agiollo, Pallavi Kaliyar, Luca Pajola, Thiemo Voigt, and Mauro Conti. Detonar-light: An iot network intrusion detection using detonar without a sniffer network. In *European Symposium on Research in Computer Security*, pages 198–213. Springer, 2023.
- [16] Giovanni Ciatto, Federico Sabbatini, Andrea Agiollo, Matteo Magnini, and Andrea Omicini. Symbolic knowledge extraction and injection with sub-symbolic predictors: A systematic literature review. *ACM Computing Surveys*, 2024.
- [17] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20:55:1–55:21, 2019.
- [18] Dan Feldman. Introduction to core-sets: an updated survey. *CoRR*, abs/2011.09384, 2020.



References IV

- [19] Isha Garg and Kaushik Roy.
Samples with low loss curvature improve data efficiency.
 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023.
- [20] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao.
Knowledge distillation: A survey.
International Journal of Computer Vision, 129(6):1789–1819, 2021.
- [21] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang.
Pruning and quantization for deep neural network acceleration: A survey.
Neurocomputing, 461:370–403, 2021.
- [22] Hanxiao Liu, Karen Simonyan, and Yiming Yang.
DARTS: differentiable architecture search.
 In *ICLR 2019, New Orleans, LA, USA*.
- [23] Renqian Luo, Fei Tian, Tao Qin, Enhong Chen, and Tie-Yan Liu.
Neural architecture optimization.
 In *NeurIPS 2018, Montréal, Canada*.
- [24] Mattia Passeri, Andrea Agiollo, and Andrea Omicini.
Peer-reviewed federated learning.
 In Rino Falcone, Cristiano Castelfranchi, Alessandro Sapienza, and Filippo Cantucci, editors, *WOA 2023 – 24th Workshop “From Objects to Agents”*, volume 3579 of *CEUR Workshop Proceedings*, pages 49–65. Sun SITE Central Europe, RWTH Aachen University, November 2023.

References V

- [25] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite.
Deep learning on a data diet: Finding important examples early in training.
In *35th Annual Conference on Neural Information Processing Systems, NeurIPS, December 6-14, 2021, virtual*, pages 20596–20607, 2021.
- [26] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean.
Efficient neural architecture search via parameter sharing.
In *ICML 2018, Stockholm, Sweden*.
- [27] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon.
An empirical study of example forgetting during deep neural network learning.
In *7th International Conference on Learning Representations, ICLR, New Orleans, LA, USA, May 6-9, 2019*.
OpenReview.net, 2019.
- [28] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li.
Dataset pruning: Reducing training data by examining generalization influence, 2023.
- [29] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao.
A survey on federated learning.
Knowledge Based Systems, 216:106775, 2021.

