



SAI *Social Explainable AI*

CHIST-ERA PROJECTS SEMINAR 2022

28-30 March 2022, virtual

Andrea Passarella, *SAI PI*

Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche
Pisa, Italy



Social Explainable AI – overall vision



SAI approach to future ML:

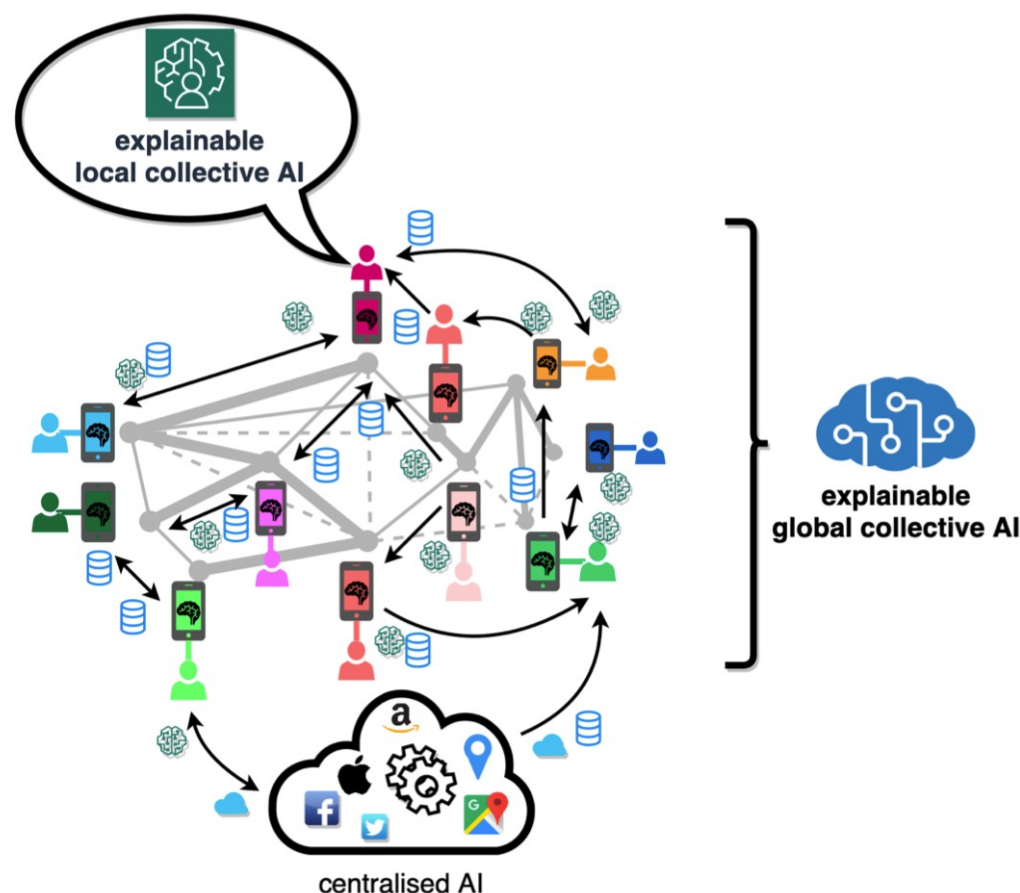
- o From *centralised, black-box* to *decentralised, transparent*

We envision a decentralized collective of personal AI

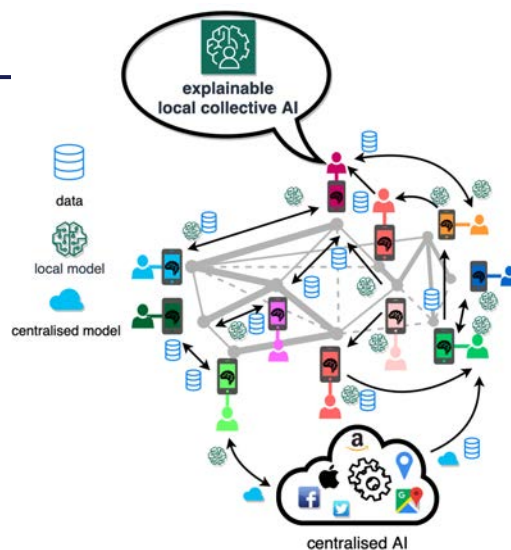
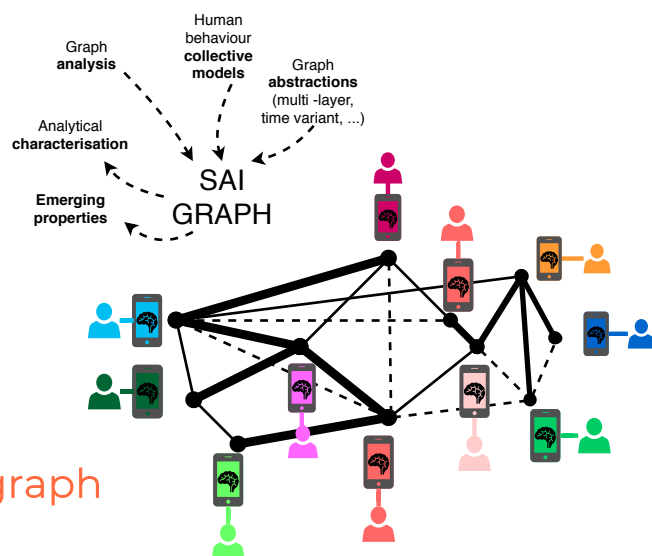
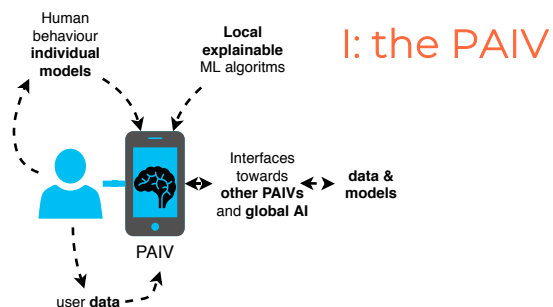
on top of a social network

interpreting data and interacting according to human-centric design principles

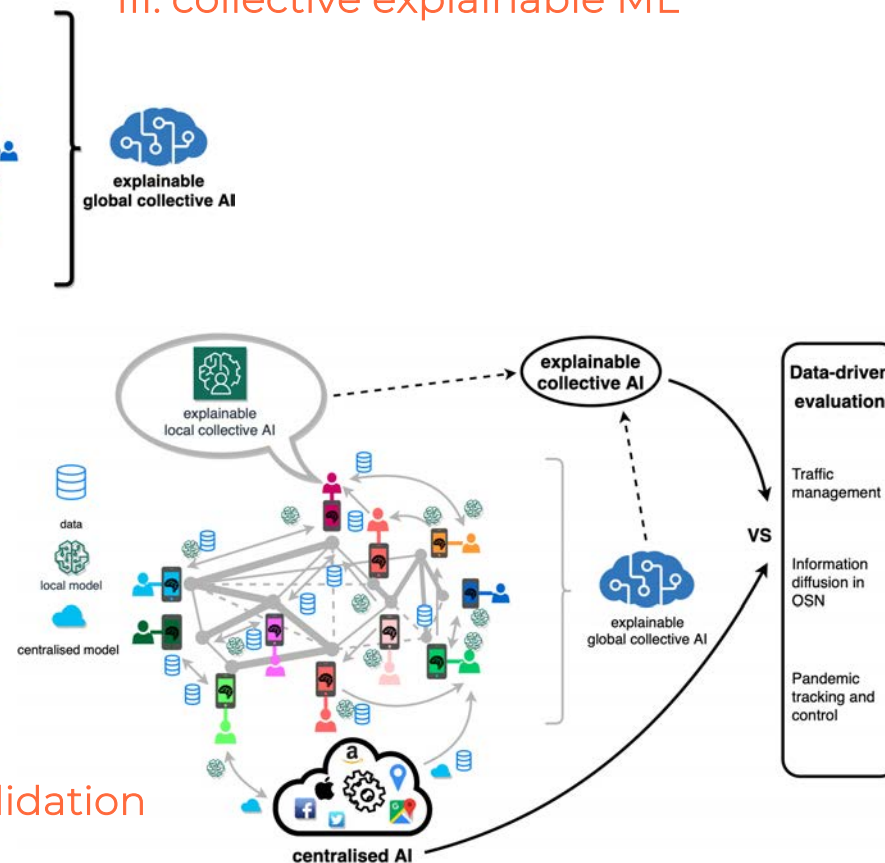
where explainability is guaranteed both at the local and collective level



The four pillars



III: collective explainable ML



IV: validation

SAI – the consortium



- **Consiglio Nazionale delle Ricerche, Italy**
 - Coordinator
 - Topics:
 - human-centric models
 - Explainable AI
 - Decentralised AI
 - Social networking
- **University of Pisa, Italy**
 - Explainable AI
 - Mobility analysis
 - Complex network modelling
- **Central European University, Austria**
 - Complex network modelling
 - Emerging properties
- **University of Sheffield, UK**
 - BigData analytics and AI
 - NLP
- **Polish Academy of Sciences, Poland**
 - Explainable AI
 - Trust & AI
- **University of Tartu, Estonia**
 - Complex network modelling
 - Social networking
- **Sofia University “St. Kl. Ohridski”, Bulgaria**
 - ML models
 - Explainable AI

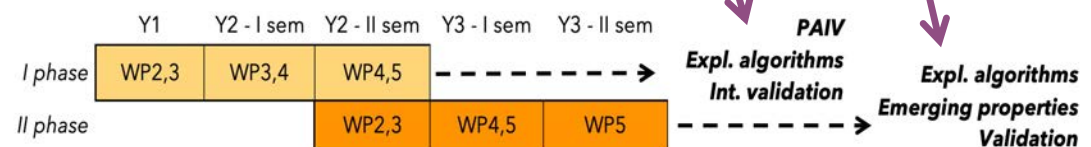


Timing



- Extended by 6 months
 - Due to hiring issues during COVID
- Phase I (~M1-M24)
 - Human behavioural models
 - PAIV & local explainable ML
 - SAI graph characterisation
 - Initial decentralised ML
 - Initial evaluation
- Phase II (~M24-M42)
 - SAI emerging properties & control
 - Decentralised ML
 - Collective explanation
 - Validation

	Y1				Y2				Y3			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
WP1: Project Management												
T1.1: Administrative and technical management												
WP2: User-centric PAIV												
T2.1: Individual and social behavioural models												
T2.2: Explainable local AI models												
T2.3: From PDS to PAIV												
WP3: SAI modelling and emerging properties												
T3.1: The SAI graph												
T3.2: SAI emerging properties												
WP4: SAI decentralised knowledge and decision making												
T4.1: Decentralised human-centric ML models and decision-making algorithms												
T4.2: Collective Explainable AI models												
T4.3: Trust and malicious behaviours												
WP5: SAI Validation												
T5.1: Tools for data collection, curation, storage and sharing												
T5.2: Validation use case #1: traffic optimisation												
T5.3: Validation use case #2: (dis-)information diffusion in OSN												
T5.4: Validation use case #3: pandemic tracking and control												
WP6: Dissemination and Communication												
T6.1: Dissemination via publications and liaising with related projects												
T6.2: Dissemination via events and communication												
T6.3: Training												
Milestones												



Selected results during Y1

- Common simulation platforms

- o Social AI Gossiping [CNR]
- o Flower platform [SU]



- “Local” XAI

- o AI on resource-constrained devices [CNR]
- o Local explainers: model and counterfactuals [UNIPi]
- o Local AI for text analysis [USFD]
- o Models of human individual behaviour [CNR]



- “Collective” XAI

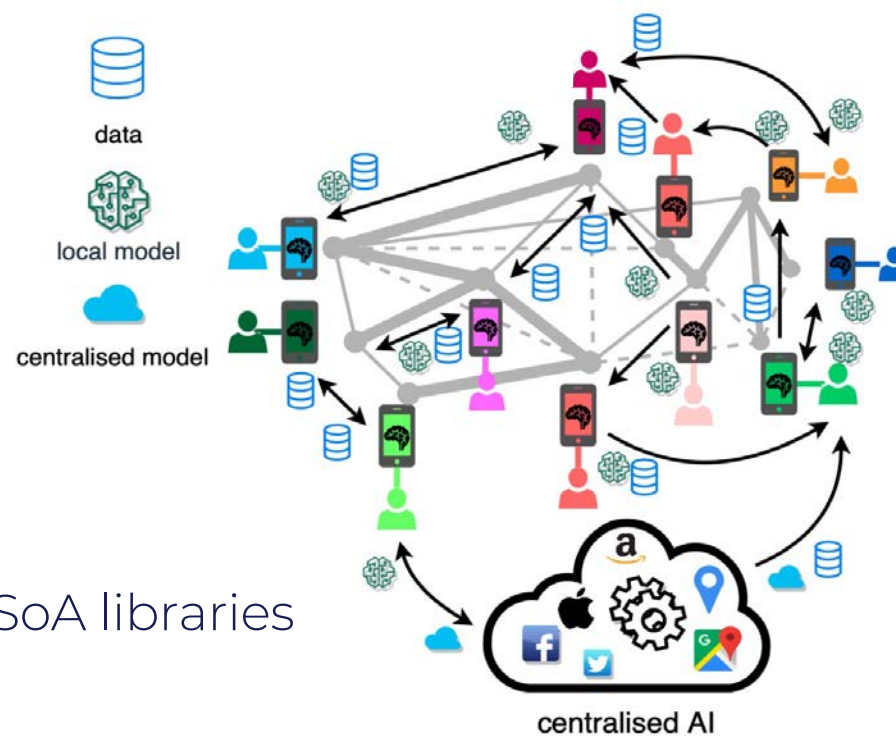
- o Social Training [CNR, CEU]
 - With model checking dynamics [IPI-PAN]
- o Opinion Dynamics [CNR, CEU]
- o Effect of selfish AI [CNR, IPI-PAN]
- o Models of human social behaviour [CEU]



Social AI gossiping: the SAI simulator (SAISim)



- Factors in
 - Social structure
 - Decentralised learning
 - Mobility (next release)
- Benchmarks included
 - Centralised learning
 - Federated learning
- Written in python, leveraging existing SoA libraries
 - Pytorch, networkx
- Modular



- Three possible inputs

- Dataset already loaded by `networkx`

- E.g. Karate graph

- Synthetic graph

- You pass the name of the generating function (e.g., `barabasi_albert_graph`) and its arguments as a string

```
ga = map(conver_to_numeric, self._args.graph_args.split(','))
self.sai_graph = nx.__getattr__(self._args.graph_type)(*ga)
```

- Edge list file

- So, basically any graph can be loaded

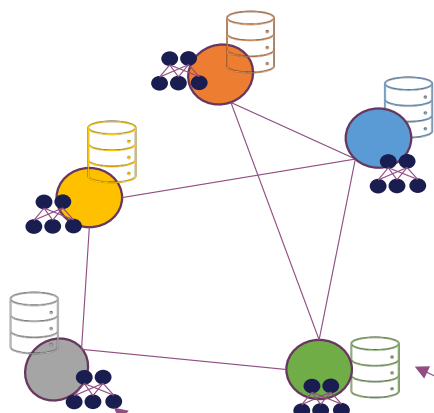
```
# load graph from edge list file
self.sai_graph = nx.read_edgelist(
    self._args.edgelist_filename, nodetype=int, data= (("weight", float),))
```



Available Training paradigms in SAI-Sim

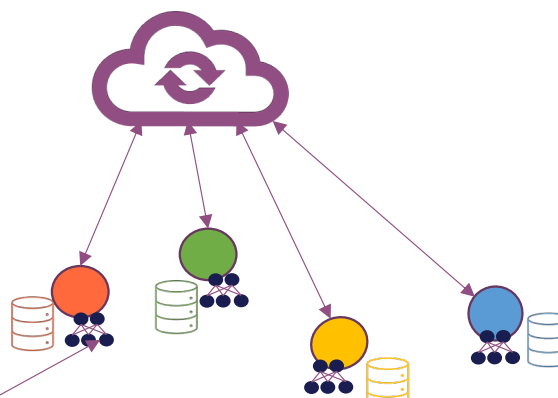


Decentralised Learning



Local ML model

Traditional Federated Learning [benchmark]



Local dataset

Centralised Learning (e.g. cloud) [benchmark]



Global model

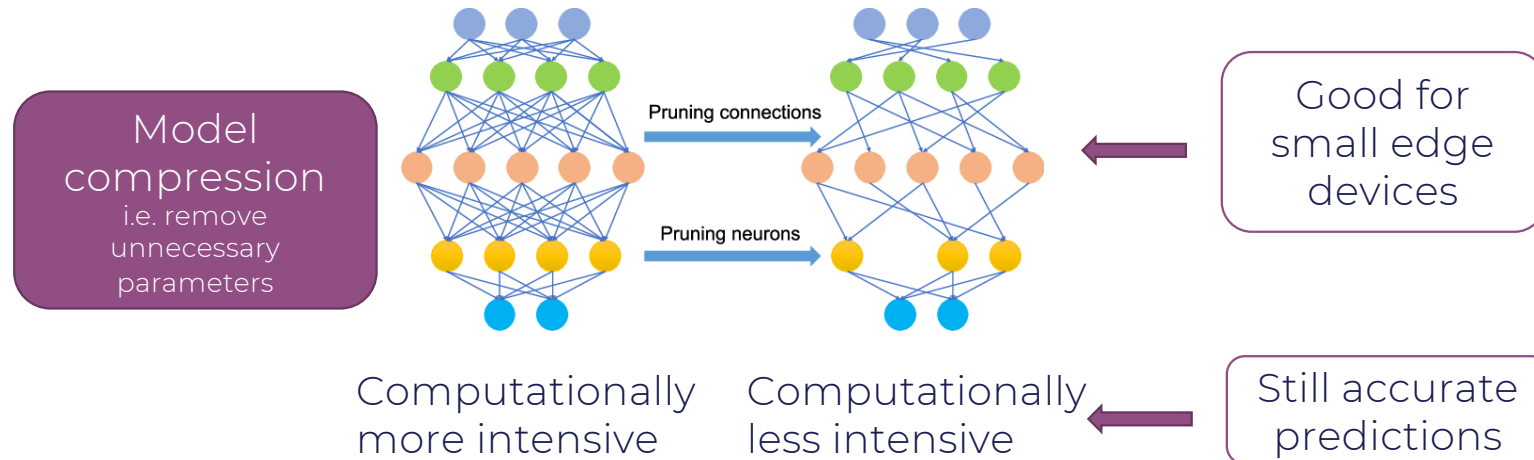
Centralised dataset



AI on resource-constrained devices



- Scenario
 - Train complex AI model on resource-constrained devices
 - E.g., users' personal smartphones, wearable devices
- Novel issues:
 - Not only Communication related: churning, limited bandwidth,
 - but also Computation related: limited computational capacity



DynHP – Train & Compress DNN under a fixed memory budget

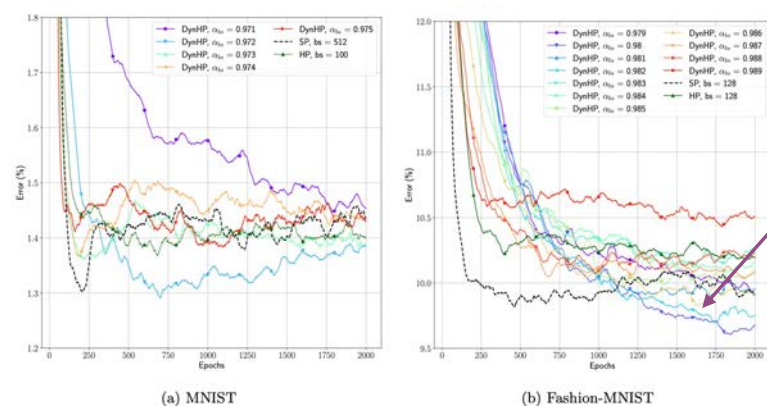


Figure 5: Misclassification error (%) of DynHP, SP, HP as a function of the number of training epochs by varying α_{bs} .

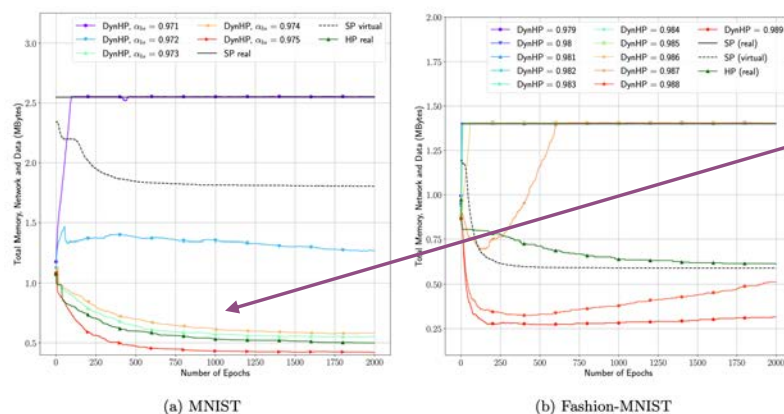


Figure 7: Total memory occupation for DynHP with different configurations, HP (best) and SP (best).

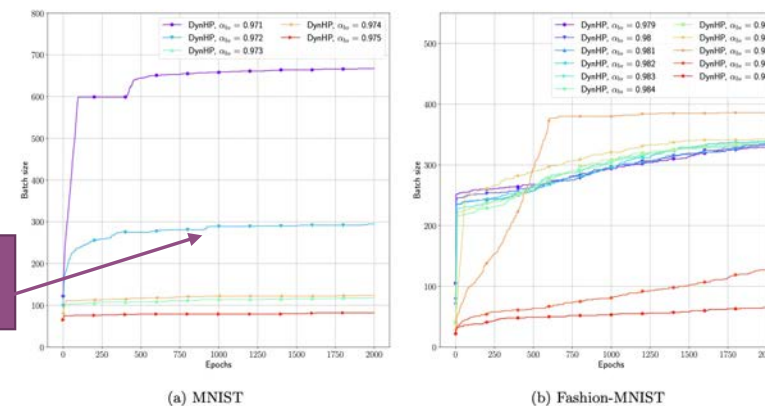
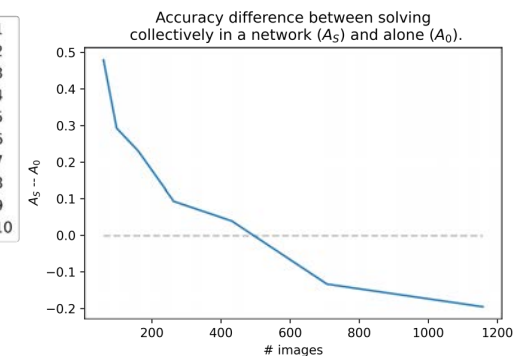
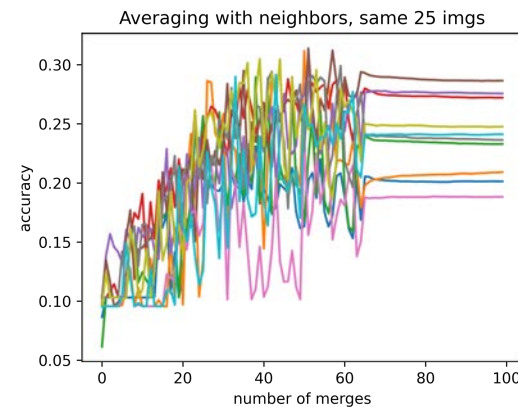
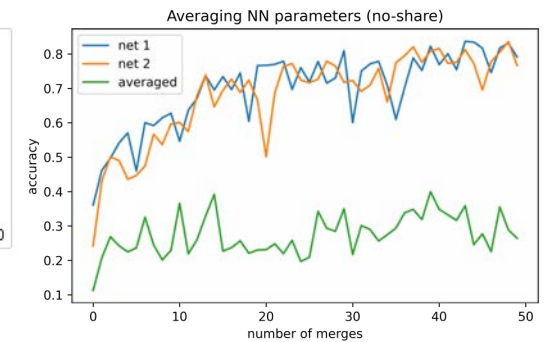
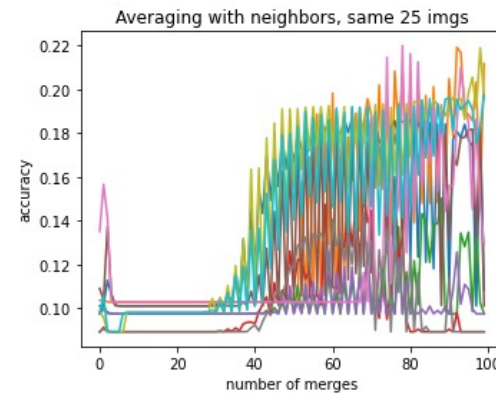


Figure 6: Size of mini-batches as a function of training epochs for different values of the smoothing parameter α_{bs} .

Method	α_{bs}	Misclassification Error (%)	Model Size (%)	Training Memory (MBytes)
SP ($bs = 128$)	—	9.96 (—)	1.041 (—)	2.866 (—)
HP ($bs = 128$)	—	10.20 (+2%)	0.236 (−77%)	1.377 (−52%)
DynHP	0.979	9.97 (0%)	0.430 (−59%)	2.874 (0%)
DynHP	0.980	9.64 (−3%)	0.415 (−60%)	2.874 (0%)
DynHP	0.981	9.96 (0%)	0.418 (−60%)	2.874 (0%)
DynHP	0.982	9.77 (−2%)	0.399 (−62%)	2.874 (0%)
DynHP	0.983	10.21 (+2%)	0.407 (−61%)	2.872 (0%)
DynHP	0.984	10.12 (+2%)	0.405 (−61%)	2.872 (0%)
DynHP	0.985	10.06 (+1%)	0.410 (−61%)	2.872 (0%)
DynHP	0.986	9.93 (0%)	0.386 (−63%)	2.858 (0%)
DynHP	0.987	10.08 (+1%)	0.252 (−76%)	2.573 (−10%)
DynHP	0.988	10.23 (+3%)	0.135 (−87%)	0.829 (−71%)
DynHP	0.989	10.50 (+5%)	0.124 (−88%)	0.611 (−79%)

Compression (gain)

- Training NNs in (social) networks
 - Structure: Granovetterian network (2 groups of 5 vertices each) with a NN in each vertex.
 - Train with smaller data (25 images), IID-data for each vertex. Keep same data all the time.
 - Each vertex trains and then...
 - Each node averages its parameters with their neighbors, or
 - Each node copies the best strategy in the network, or
 - Each node exchanges with its neighbors if the accuracy is better.
 - (Repeat 100 times)





FWF

EPSRC



chist-era



Thank you for your attention!

