Huawei Edinburgh Joint Lab Projects Summary

Peter Buneman, Jane Hillston and Jeff Pan

1st Edition
December 2023

The Huawei Edinburgh Joint Lab funds a variety of research projects in the School of Informatics at the University of Edinburgh. It started in 2016 following a very successful project with Professor Wenfei Fan at the University of Edinburgh, in which he invented a “Bounded Evaluation” technique that dramatically increased the speed of some of Huawei’s critical applications. The original title (and still the official title) of the project was “The Distributed Data Management and Processing Laboratory”. However, from the start it was clear that the School of Informatics had much more to offer than data management; notably natural language processing, AI, compilers and architecture were all of interest to Huawei. A guiding principle of the agreement, to which both Edinburgh and Huawei subscribed, is that all research generated through the Joint Lab should be in the public domain.

After the Joint Lab had been running for two years, and after a visit by their senior management, Huawei decided to create their own research centre in Edinburgh. This grew under the direction of Li Bo and greatly strengthened research interactions between Huawei and the School of Informatics in the areas of databases, knowledge graphs and architecture. The Centre has also recently added common interests in operating systems and programming languages – one of the great strengths of Informatics.

So far the Joint Lab has funded about 30 projects. This booklet is a summary of most of the current projects and some that have just started or are expected to start shortly. We have also included some summaries of previous projects that have finished. Each project has a short abstract, some information about the people associated with the project and a poster that describes one or more examples of the research undertaken in that project.

This represents just a snapshot of the vibrant research community that has grown around the Huawei Edinburgh Joint Lab and collaboration between researchers in the School of Informatics and researchers in Huawei, in Edinburgh and elsewhere. Many of the projects fund PhD students, providing excellent opportunities to incorporate industrial collaboration into research training. The topics covered reflect the breadth of interests in both the School and Huawei, and the numerous best paper awards attracted by the research outputs demonstrates the timeliness and excellence of the research undertaken.

If you wish to get in touch with the investigators associated with a project, please consult https://www.ed.ac.uk/informatics/people/academic where you will find email addresses of faculty and links to other researchers mentioned in these summaries.

Please note that this compilation was extracted from reports and presentations by the principal investigators and their colleagues. It has been edited to fit a simple format. The authors are responsible for any errors introduced. They expect to update this booklet from time to time.

Peter Buneman, University of Edinburgh
Jane Hillston, University of Edinburgh
Jeff Pan, Huawei

Semantic Parsing and Dialog for Knowledge-Graph Query

PI: Mark Steedman

CoPIs: Simon King; Mirella Lapata; Ivan Titov; Bonnie Webber

Bio: Mark Steedman is Professor of Cognitive Science in the School of Informatics, working in Computational Linguistics, AI, and Cognitive Science, on Generation of Meaningful Intonation for Speech by Artificial Agents, The Communicative Use of Gesture, Tense and Aspect, and Wide coverage parsing and robust semantics for Combinatory Categorial Grammar (CCG). He is also interested in Computational Musical Analysis and Combinatory Logic.

Summary

Information may be structured data or unstructured text. Data access for mortals has to be via natural language. Natural language access also requires dialog and spoken interaction. For both kinds of data, the central problem of natural language access is variation in language on the user side. The project investigates natural language question answering from structured data and the transformation of Semantic unstructured text into structured knowledge graphs via semantic parsing, and obtains robustness in the face of variation in language using large language models and entailment graphs.

Selected Publications

- Sherborne et al., (2023). “Optimal Transport Posterior Alignment for Cross-lingual Parsing”, *Transactions of the Association for Computational Linguistics, (to appear).*
- Jain and Lapata, (2023). “Conversational Semantic Parsing using Dynamic Context Graphs,” Proceedings of the Conference on Empirical Methods in Natural Language Processing, (to appear).
- Lindemann et al., (2023). “Compositional Generalization without Trees Using Multiset Tagging and Latent Permutations”, Proceedings of the Annual Meeting of the Association for Computational Linguistics, 14488–14506.
- Sherborne and Lapata, 2023. “Meta-Learning a Cross-lingual Manifold for Semantic Parsing”, *Transactions of the Association for Computational Linguistics, 11,* 49-76.
- Cheng et al., (2023). “Complementary Roles of Inference and Language Models in Question Answering”, Proceedings of the EMNLP Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning, (to appear).
- McKenna et al., (2023a). “Smoothing Entailment Graphs with Language Models”, Proceedings of the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (AACL/IJCNLP), 551-563. **Best Paper Award.**
- Wu et al., (2023). “Align-then-Enhance: Entailment Graph Enhancement with Soft Predicate Alignment”, Findings of the Association for Computational Linguistics: ACL, 881-894.
- Puduppully st al., (2023). “Multi-Document Summarization with Centroid-Based Pretraining”, Proceedings of the Annual Meeting of the Association for Computational Linguistics, 128-138.
- Moghe et al., (2023). “Extrinsic Evaluation of Machine Translation Metrics”, Proceedings of the Annual Meeting of the Association for Computational Linguistics, 13060-13078. ***Best Paper Award.***

Sources of Hallucination by Large Language Models on Inference Tasks

Edinburgh NLP
University of Edinburgh
Natural Language Processing

Nick Mckenna*, Tianyi Li*, Liang Cheng, Javad Hosseini, Mark Johnson, Mark Steedman | findings of EMNLP 2023

LLMs in NLI

- Recent LLMs (e.g. GPT-3/4, LLaMA, PaLM) show encouraging results in various NLP tasks including NLI. However their behavior is not well-explained.
- This paper investigates two biases as key sources of LLM hallucination.
- For test samples adversarial to these biases, LLM performance is severely degraded.

Attestation and Relative Frequency

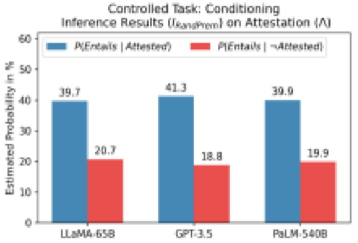
- **Attestation Bias (Λ)**: whether the hypothesis of an entailment entry is attested in LLM pre-train data.
  - Measure: LLM truthfulness predictions
- **Relative Frequency Bias (Φ)**: whether premise is less frequent than hypothesis in LLM pre-train data
  - Measure: Google N-gram frequency

Experimental Method

- Datasets: Levy/Holt ; RTE-1
- Test conditions to study :
  - **Standard Inference Task I**
  - **Random Premise Task $I_{RandPrem}$** : premises are replaced with random predicates, samples all become non-entailments;
  - **Random Argument Task $I^{RandArg}$** : arguments of both predicates are replaced with random entities; attestation is corrupted but entailment label is unchanged.
  - **Type Argument Task $I^{TypeArg}$** : arguments of both predicates are replaced with entity type labels, entailment label is unchanged.
- LLMs to Investigate:
  - LLaMA-65B
  - GPT3.5: text-davinci-003
  - PaLM-540B

Exp 1: Attestation Bias

- When a hypothesis is attested in pre-train data, LLMs are twice as likely to predict it to be ``entailed'', even by a random premise. ← *leads to hallucination!*



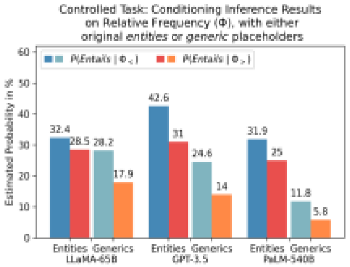
Exp 2: Entities are Indices to Memory

- When random entities are replaced in entailment entries, LLMs recognize much fewer entailments.
- However, this loss in recall is **NOT** remedied by precision gain.
- This is worse when popular entities are swapped in, meaning: the more an LLM has read about an entity in pre-training, the less capable it is of drawing novel inferences about it, even when the inferences do not require detailed knowledge of entities.

		Levy/Holt (Directional)		
Model	**Task**	Precision	Recall	*Δ-Recall*
LLaMA	I	67.0	**68.4**	*0*
	I^{GenArg}	69.0	66.9	*-1.5*
	$I^{RandArg\downarrow}$	64.0	63.8	*-4.6*
	$I^{RandArg\uparrow}$	67.2	53.7	*-14.7*
GPT-3.5	I	62.4	**92.3**	*0*
	I^{GenArg}	65.1	75.7	*-16.6*
	$I^{RandArg\downarrow}$	65.5	66.5	*-25.8*
	$I^{RandArg\uparrow}$	68.8	55.3	*-37.0*
PaLM	I	72.8	**76.2**	*0*
	I^{GenArg}	79.8	50.8	*-25.4*
	$I^{RandArg\downarrow}$	69.5	58.7	*-17.5*
	$I^{RandArg\uparrow}$	70.8	52.4	*-23.8*

Exp 3: Relative Frequency Bias

- When hypothesis is more frequent than its premise, LLMs are also more likely to predicted as ``entailed''.
- This is worse with TypeArgs, when attestation info is missing



Impact of Bias on Performance

- When entries in original NLI datasets align with these biases, LLMs perform well; when test samples are adversarial, LLMs degrade into poor, even random discriminators.

		Levy/Holt					
		Attestation (Λ)			Rel. Frequency (Φ)		
Model	**Task**	*cons.*	*adv.*	*diff.*	*cons.*	*adv.*	*diff.*
LLaMA	I	65.5	8.1	***-57.4***	42.1	32.3	***-9.8***
GPT-3.5	I	85.0	10.8	***-74.2***	53.5	43.2	***-10.3***
PaLM	I	79.1	31.5	***-47.6***	63.3	53.0	***-10.3***
LLaMA	I^{GenArg}	52.1	34.4	***-17.7***	55.3	34.9	***-20.4***
GPT-3.5	I^{GenArg}	67.1	18.8	***-48.3***	50.4	35.0	***-15.4***
PaLM	I^{GenArg}	58.1	46.6	***-11.5***	59.9	47.3	***-12.6***

Key References

- [1] Hugo Touvron, et. al., Llama: Open and Efficient Foundation Language Models. 2023.
- [2] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis Only Baselines in Natural Language Inference. SEM 2022.
- [3] Maayan Geffet and Ido Dagan. The Distributional Inclusion Hypotheses and Lexical Entailment. ACL 2005

Data Sharing: Querying and Linking Distributed and Autonomous Data

PI: Yang Cao

CoPIs: Wenfei Fan; Peter Buneman

Bio: Yang Cao is a Chancellor's Fellow at the University of Edinburgh. He has been working on database query processing, transactions, graph computations, and systems and theory of data management in general. He holds a number of awards and fellowships, including the RAEng research fellowship (2020), SIGMOD Research Highlight award (2018), and SIGMOD best paper award (2017).

Summary: In a world that consists of increasingly connected sources of data, each with its own data model, schema, and access methods, the greatest challenge is to share data securely and efficiently. This involves creating contracts that describe how data is to be shared and prevent data leaking to parties that should not see it. It requires the ability to track the movement of data: where has it come from and where is it going. In addition, it involves the long-standing problems of data linking, query optimization and data caching, but in a new environment in which we are constrained by data sharing contracts with new definitions of correctness criteria from emerging workloads. In real-life, "open" data is typically not as open as one would like, and that organisations are unlikely to share their data until they are satisfied that issues of privacy and accountability have been properly addressed.

Selected Publications:

- Shuai An and Yang Cao, "Relative Keys: Putting Feature Explanation into Context", (SIGMOD 2024)
- Shuai An and Yang Cao, "Making Cache Monotonic and Consistent", (VLDB 2023)
- Jia Li, Wenyue Zhao, Nikos Ntarmos, Yang Cao and Peter Buneman, "MITra: A Framework for Multi-Instance Graph Traversal", (VLDB 2023)
- Yang Cao, Wenfei Fan, Wenzhi Fu, Ruochun Jin, Weijie Ou and Wenliang Yi, "Extracting Graphs Properties with Semantic Joins", (ICDE 2023)
- Shuai An, Yang Cao, Wenyue Zhao, "Competitive Consistent Caching for Transactions", (ICDE 2022)
- Wenfei Fan, Ruochun Jin, Ping Lu, Resul Tugay, Wenyuan Yu, "Linking Entities across Relations and Graphs", (ICDE 2022)
- Peter Buneman, Dennis Dosso, Matteo Lissandrini, Gianmaria Silvello, "Expanding the Citation Graph for Data Citations", (SEBD 2022)

Data Sharing:
Querying and Linking Distributed and Autonomous Data

Yang Cao, Wenfei Fan, Peter Buneman

University of Edinburgh

Background

Distributed data: healthcare, business intelligence, e-government, . . .

- *Tasks*: querying, linking, sharing and learning
- *Data*: distributed, heterogeneous, large scale

Challenges:

- **Privacy & security**: data owners ("private") vs. users ("open")
- **Heterogeneity**: relations, key-value, graphs
- **Scalability**: limited resources vs. big data analytics
- **Accountability**: accountability and explainability of shared data

Querying Shared Data with Security Heterogeneity [7]

Challenges

- heterogeneous security requirements
- centralized evaluation not possible

Data sharing protocols ρ specifying:

- capsules: logic units for computations
- hosts: data owners that host capsules
- pair-wise privacy requirements:
  - permitted capsule specifications
  - secure communication overheads

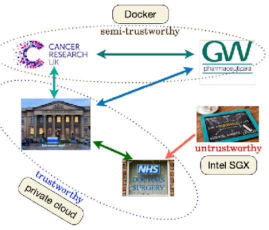


Figure: Heterogeneous data sharing

Planning under protocols: find optimal plans with bounded security *toll*:

- decidable in NEXPTIME; PSPACE-hard even for SPC with linear ρ.
- approximation with guarantees; 18X faster than SMCQL on TPCH.

Heterogeneous Entity Resolution [8]

Main task: link entities across a relational database D and a graph G

Challenge: traditional ER methods work for relations only (homogeneous).

HER: decide whether tuple $t \in D$ matches vertex v of G:

1. convert relations D into a graph G_D
2. whether t matches $v \Rightarrow$ whether v_t in G_D matches v in G
3. quadratic-time parametric simulation between G_D and G
   - graph simulation extended with label matching functions
   - learned threshold for score functions
   - parallelizable to scale over large graphs and relations

Joins Across Relations and Graphs [6]

Main task: Query relations and graphs taken together in SQL.

Challenge: semantically meaningful joins between relations $\mathcal{D}$ and graph G.

Semantic join: if tuple $t \in \mathcal{D}$ and vertex v of G refer to the same entity, $t \Join v$ complement t with properties of v extracted from G. It is based on:

1. HER that decides whether t of $\mathcal{D}$ matches v of G
2. an extraction scheme based on LSTM, path clustering and ranking, to fetch important properties of v for t from G
3. incremental maintenance of the extracted data of v for t.

Scaling via Consistent Caching [3]

Background: transactional data accesses are prevalent, *e.g.*, federated ML
Results: supporting transactional data access over caches:

- formulate consistent cache scheme; show it's NP-complete even for uni-size accesses, in contrast to trivially PTIME for conventional caching
- develop theoretically competitive and optimal consistent cache policies
- implemented and tested with Memcached@HBase: 126% improvement on average for transaction throughput, while guaranteeing consistency

Monotonic Consistent Caching [1]

Overview: further extend [3] to support

- both monotonicity and consistency for transactional data access;;
- three input models: batch, semi-online and online.

Results: a monotonic consistent cache (MCC) scheme for transactions:

- prove its NP-completeness and develop an optimal policy (batch model)
- for semi-online (reads offline writes online) and online models, develop MCC policies that take *blackbox binary* ML classifier as an oracle M:
  - [Competitiveness] optimal if M is 100% accurate, and
  - [ML-robustness] competitive even if M is adversarial (0% accuracy).
- implemented for Redis@HBase: improve transaction throughput by 77.15% while guaranteeing consistency and monotonicity

Client-centric ML Explainability over MCC caches [2]

Proposition: When serving remote/federated ML models, MCC caches enable users to explain predictions *in the context of* cached inference instances.
Relative keys: ML explanations *relative to* cached inference instances

- give users the "right-to-explain" advocated by GDPR
- work for data sharing scenarios with proprietary cloud-based ML models
- the best of both worlds: strict conformity and superior efficiency

Accountability of Shared Curated Data [5, 4]

Background: datasets are *shared/copied* $\longrightarrow$ *modified* $\longrightarrow$ *published* $\longrightarrow \cdots$

- a data collection contains contributions from multiple users
- contributions form a dependency hierarchy (copy-modify-contribute)
- however, entire dataset is often treated as one single "article"

Main task: how to account the ownership of pieces of data in a dataset

Results: a model of citation graph for databases/datasets

- method for generating data summaries of "optimal" granularity
  - stress measures of data summaries
  - compute accountability by measuring stress
- application to a collectively curated pharmacology database (GtoPdb)

From Data Sharing to Computation Sharing [9]

Multi-instance computation: Multiple algorithm instances over the same data; improve *per-thread* performance (*concurrency*) by sharing computations.

Challenge: Writing correct and efficient multi-instance algorithms is difficult.

MITra: A framework for composing multi-instance graph algorithms:

- Ease of programming: extract edge functions from textbook algorithms
- based on an expressive frontier-ranking model
- synthesize a full multi-instance algorithm from edge functions that are an order of magnitude faster than serial algorithms for common queries.

Publications

[1] S. An and Y. Cao. Making cache monotonic and consistent. In *VLDB*, 2023.

[2] S. An and Y. Cao. Relative keys: Putting feature explanation into context. In *SIGMOD (under revision)*, 2024.

[3] S. An, Y. Cao, and W. Zhao. Competitive consistent caching for transactions. In *ICDE*, 2022.

[4] P. Buneman, D. Dosso, M. Lissandrini, and G. Silvello. Data citation and the citation graph. *Quant. Sci. Stud.*, 2(4), 2021.

[5] P. Buneman, D. Dosso, M. Lissandrini, and G. Silvello. Expanding the citation graph for data citations. In *SEBD*, pages 276–283, 2022.

[6] Y. Cao, W. Fan, W. Fu, R. Jin, W. Ou, and W. Yi. Extracting graphs properties with semantic joins. In *ICDE*, 2023.

[7] Y. Cao, W. Fan, Y. Wang, and K. Yi. Querying shared data with security heterogeneity. In *SIGMOD*, 2020.

[8] W. Fan, L. Geng, R. Jin, P. Lu, R. Tugay, and W. Yu. Linking entities across relations and graphs. In *ICDE*, 2022.

[9] J. Li, W. Zhao, N. Ntarmos, Y. Cao, and P. Buneman. Mitra: A framework for multi-instance graph traversal. In *VLDB*, 2023.

In-Database Universal Analytics Using Compilation

PI: Amir Shaikhha

Bio: Amir Shaikhha is an Assistant Professor (Lecturer) in the School of Informatics at the University of Edinburgh. His research focuses on the design and implementation of data-analytics systems by using techniques from the databases, programming languages, compilers, and machine learning communities. Prior to that, he was a Departmental Lecturer at Oxford. He earned his Ph.D. from EPFL in 2018, for which he was awarded a Google Ph.D. Fellowship in structured data analysis, as well as a Ph.D. thesis distinction award. He has won the Best Paper Award at GPCE 2017 and the Most Reproducible Paper Award at SIGMOD 2017. He (co-)chaired the program committees of DBPL 2021, Scala 2022, DRAGSTERS 2023, and GPCE 2023.

Summary: The mainstream approach to machine learning over relational data consists of two steps. The training dataset is first constructed via a feature extraction query over the input database using a database system or minimalistic query engines such as Pandas. The desired model is then trained over the result of the query using a statistical software package of choice such as R, scikit-learn, or TensorFlow. The key idea of this proposal is to integrate the entire process of data analytics, including learning over relational data, into one single DSL. In this way, the DSL can take advantage of the structure of the data, such as integrity constraints and sparsity, to achieve massive performance improvements in the machine learning algorithms. Furthermore, one can benefit from the specialisation and optimization opportunities enabled by compilation. The direction of this proposal is to apply recent developments at the intersection of programming languages, data management, and machine learning to develop DSLs for data analytics.

Selected Publications

- M. Ghorbani, M. Huot, S. Hashemian, A. Shaikhha, “Compiling Structured Tensor Algebra”, OOPSLA’23.
- Jingwen Pan, Amir Shaikhha, “Compiling Discrete Probabilistic Programs for Vectorized Exact Inference”, CC’23
- Amir Shaikhha, Marios Kelepeshis, Mahdi Ghorbani, “Fine-Tuning Data Structures for Query Processing”, CGO’23
- Maximilian Schleich, Amir Shaikhha, Dan Suciu, “Optimizing Tensor Programs on Flexible Storage”, SIGMOD’23
- M. Ghorbani, A. Shaikhha, “Demonstration of OpenDBML, a Framework for Democratizing In-Database Machine Learning”, VLDB’23 (demo).

- Jingwen Pan, Amir Shaikhha, “Compiling Discrete Probabilistic Programs for Vectorized Exact Inference”, CC’23
- Amir Shaikhha, Marios Kelepeshis, Mahdi Ghorbani, “Fine-Tuning Data Structures for Query Processing”, CGO’23
- Maximilian Schleich, Amir Shaikhha, Dan Suciu, “Optimizing Tensor Programs on Flexible Storage”, SIGMOD’23

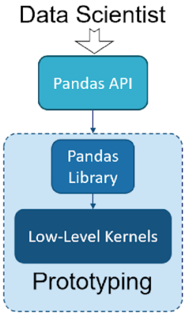
THE UNIVERSITY *of* EDINBURGH

SDQL.py: An Efficient Query Engine in Python

Hesam Shahrokhi, Amir Shaikhha

School of Informatics, University of Edinburgh

Query Processing in Python: Current State of Affairs



Pandas API

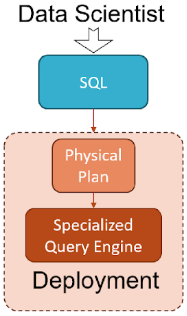
☑ Easy Coding

☑ Easy Debugging

Hand-tuned Low-level Kernels

☒ No Global Optimization

☒ Slow Execution



SQL in Python

☒ Hard to Code

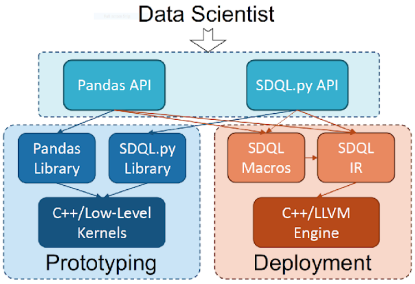
☒ Hard to Debug

Vectorized/Compiled Query Engine

☑ Global Optimization

☑ Fast Execution

Query Processing by SDQL.py



Pandas/SDQL.py API

☑ Easy Coding

☑ Easy Debugging

SDQL.py Compiler

☑ Global Optimization

☑ Fast Execution

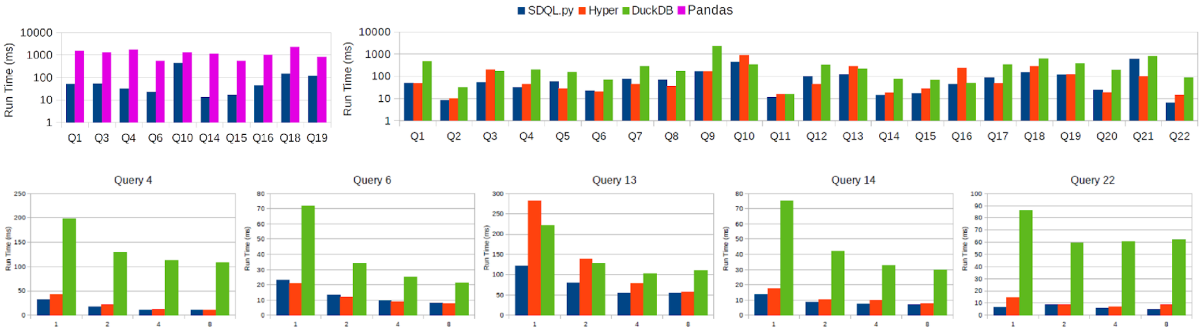
SDQL.py to SDQL IR Example

```
@sdql_compile({"R": R_type, "S": S_type}, exec_mode)
def query(R, S):
    R_indx = R.sum(lambda r: {r.k: r.c} if r.a=9 else {})
    return S.sum(lambda s:
        {record({"c": R_indx[s.k].c, "g": s.g}): True})
        if (s.e=="pass" and R_indx[s.k]) else {})
```

```
let R_indx = sum(r in R) if r.a=9 then {r.k -> r.c}
sum(s in S)
    if s.e=="pass" and R_indx[s.k] then
        {<c: R_indx[s.k].c, g: s.g> -> true}
```

Experimental Results

- Intel Core i5
- 16GB DDR4 RAM
- Ubuntu 20.04.3
- G++ 9.4.0, Python 3.8.10, and Pandas 1.4.1
- TPC-H Queries with Scaling Factor of 1



[1] Hesam Shahrokhi, Amir Shaikhha, "**Building a Compiled Query Engine in Python**", to appear in CC'23.

[2] Amir Shaikhha, Mathieu Huot, Jaclyn Smith, Dan Olteanu, "**Functional collection programming with semi-ring dictionaries**", OOPSLA'22.

Learning Structured Decompositions of Data for Interpretability

PI: N. Siddarth

Bio: Siddharth N. (Sid) is a Reader in Explainable AI in the School of Informatics at the University of Edinburgh, a part-time Senior Research Fellow at the Alan Turing Institute, a Visiting Fellow at the Department of Engineering Science at the University of Oxford, and an ELLIS Scholar. He was previously a Senior Researcher in Engineering at the University of Oxford and a Postdoctoral Scholar in Psychology at Stanford. He obtained his PhD from Purdue University in Electrical and Computer Engineering. His research interests are broadly cross-disciplinary and motivated by problems found at the intersection of machine learning, computer vision, natural-language processing, cognitive science, robotics, and neuroscience.

Summary: This project explores steps towards an approach for generalised interpretability by learning structured decompositions of data. It seeks to espouse all the relevant characteristics one would want from a human-interactable, interpretable, and explainable system in that it a) incorporates interpretability by default by capturing meaningful factors of data, b) avoids the need for supervision through annotation and labels, and c) allows for capturing dependence relations between meaningful factors. Crucially, enabling such a system requires i) learning to decompose a given observation into its constituent parts, ii) learning the representation of such constituents, and iii) ensuring that such constituents capture semantically meaningful information.

Selected Publications

- M Opper, V Prokhorov, N Siddharth. StrAE: Autoencoding for Pre-Trained Embeddings using Explicit Structure, UM-IoS . EMNLP 2022
- M Opper, V Prokhorov, N Siddharth. On the effect of curriculum learning with developmental data for grammar acquisition, BabyLM Challenge, CoNLL 2023

Does Explicit Structure help Representation Learning?

StrAE: AutoEncoding for Pretrained Embeddings using Explicit Structure

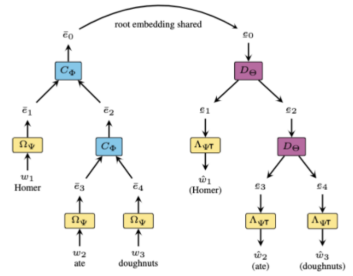
Mattia Opper, Victor Prokhorov and N. Siddharth

Overall Goals:

- Explore utility of explicit structure for unsupervised representation learning.

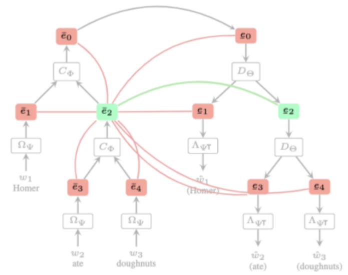
Model:

- StrAE: an autoencoder framework that operates on explicit structure
- StrAE is *faithful* to structure



Objectives:

- Cross Entropy: reconstruct the input tokens
- Contrastive: reconstruct on all levels



Tasks:

- Evaluate on the word and the sentence level
- Mostly Zeroshot

Name	Level	Task	Requires Classifier?
SimLex	Word Level	Semantic Similarity	No
WordSim S	Word Level	Semantic Similarity	No
WordSim R	Word Level	Semantic Relatedness	No
STS 12	Sentence Level	Semantic Similarity	No
STS 16	Sentence Level	Semantic Similarity	No
STS B	Sentence Level	Semantic Similarity	No
SICK R	Sentence Level	Semantic Relatedness	No
MRPC	Sentence Level	Paraphrase Detection	Yes
RTE	Sentence Level	Textual entailment	Yes

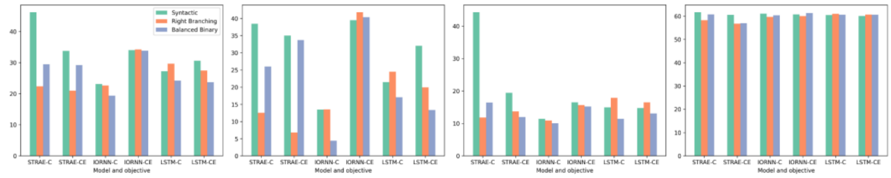
Tree Baselines:

- Less faithful than StrAE
- Tree-LSTM: cell state
- IORNN: skip connections betwenn encoder and decoder

Model	**Simlex †**	**Wordsim S †**	**Wordsim R †**	**STS 12 †**	**STS 16 †**	**STS B †**	**SICK R †**	**MRPC**	**RTE**	Score
StrAE C	15.54 ± 0.25	56.1 ± 0.47	**43.77 ± 1.01**	**34.59 ± 2.58**	**50.4 ± 0.97**	**41.83 ± 2.23**	**50.3 ± 0.88**	67.24 ± 0.39	56.16 ± 1.59	**46.21**
StrAE CE	17.57 ± 1.08	50.72 ± 2.97	36.73 ± 6.19	5.02 ± 1.1	25.86 ± 1.03	7.76 ± 0.95	39.06 ± 1.51	67.48 ± 0.3	53.6 ± 0.46	33.76
IORNN C	1.9 ± 0.25	27.29 ± 0.57	11.29 ± 0.5	-4.2 ± 0.97	17.47 ± 1.14	1.87 ±0.55	30.48 ± 0.55	67.08 ± 0.16	55 ± 1.44	23.13
IORNN CE	**24.36 ± 0.6**	57.53 ± 1.59	36.55 ± 1.78	-0.17 ± 0.92	22.68 ± 0.98	5.04 ± 0.55	38.41 ± 0.63	67.24 ± 0.63	54.28 ± 0.55	33.99
Tree-LSTM C	6.45 ± 0.4	37.39 ± 1.2	20.6 ± 1.49	-1.34 ± 1.33	23.95 ± 0.45	4.25 ± 1.11	32.9 ± 0.55	68.16 ± 0.32	52.76 ± 1.09	27.24
Tree-LSTM CE	14.23 ± 2.01	48.7 ± 3.22	33.24 ± 3.02	-1.66 ± 1.29	19.38 ± 1.21	6.7 ± 1.04	34.55 ± 1.15	68 ± 0.0	52.5 ± 1.98	30.59

Does structure matter?

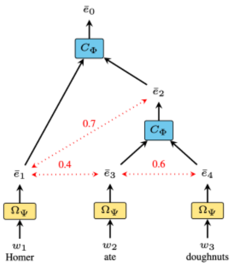
- StrAE's performance depends on informative structure
- Tree baselines performance does not
- *Faithfulness* is key



(a) Overall Score (b) Lexical Semantics (c) Sentence Level Semantics (d) Classification

Self-StrAE:

- StrAE takes structure as input
- What about if we just used an inductive bias?
- Learns its own tree with simple agglomerative clustering



Unstructured Baselines:

Model	**Simlex †**	**Wordsim S †**	**Wordsim R †**	**STS 12 †**	**STS 16 †**	**STS B †**	**SICK R †**	**MRPC**	**RTE**	**Score**
StrAE Syntactic	15.54 ± 0.25	56.1 ± 0.47	43.77 ± 1.01	34.59 ± 2.58	50.4 ± 0.97	41.83 ± 2.23	50.3 ± 0.88	67.24 ± 0.39	56.16 ± 1.59	**46.21**
Self-StrAE BPE	13.04 ± 0.07	48.19 ± 1.03	45.47 ± 0.83	34.42 ± 0.73	49.93 ± 0.38	36.68 ± 0.64	51 ± 0.3	68.91 ± 0.2	54.42 ± 0.49	44.67
Self-StrAE Word	18.21 ± 0.12	53.52 ± 0.39	**48.76 ± 0.86**	33.22 ± 0.38	49.91 ± 0.35	27.97 ± 0.2	**52 ± 0.52**	68.13 ± 0.1	54.53 ± 0.53	45.14
Fasttext	**25.8 ± 0.16**	50.8 ± 0.24	29.18 ± 0.22	4.35 ± 0.6	32.43 ± 0.1	16.93 ± 0.09	41.58 ± 0.05	69.1 ± 0.12	54.4 ± 0.67	36.06
Bi-LSTM Word	23.74 ± 1.17	**61.94 ± 2.44**	41.46 ± 1.73	11.18 ± 1.1	34.86 ± 0.86	13.46 ± 1.11	42.36 ± 0.19	68.95 ± 0.38	54.09 ± 0.95	39.12
Bi-LSTM BPE	12.88 ± 0.77	39.46 ± 2.17	32.22 ± 3.07	9.22 ± 0.87	33.78 ± 1.66	14.06 ± 2.01	40.36 ± 0.27	68.21 ± 0.41	53.9 ± 0.96	33.79
RoBERTa	9.92 ± 2.7	26.6 ± 6.71	6.2 ± 3.81	29.48 ± 3.28	50.88 ± 1.11	38.36 ± 1.9	49.58 ± 0.91	69.19 ± 0.27	54.86 ± 0.49	37.23

Scale:

Model	**Simlex †**	**Wordsim S †**	**Wordsim R †**	**STS 12 †**	**STS 16 †**	**STS B †**	**SICK R †**	**MRPC**	**RTE**	**Score**
RoBERTa	**19.28 ± 1.02**	46.38 ± 3.18	26.12 ± 3.09	35.38 ± 1.47	52.64 ± 1.11	39.74 ± 1.04	50.68 ± 0.43	**69.74 ± 0.42**	53.71 ± 0.5	43.76
Self-StrAE	13.41 ± 0.66	47.06 ± 0.61	**42.53 ± 1.71**	**46.64 ± 0.23**	52.08 ± 0.37	40.59 ± 0.83	**51.94 ± 0.4**	68.35 ± 0.46	**55.87 ± 0.3**	**46.39**

Parameters:

Model	(Self-)StrAE	IORNN	Tree-LSTM	Bi-LSTM	RoBERTa
Params	430	40,400	260,600	181,800	3,950,232

Download the Paper

THE UNIVERSITY OF EDINBURGH

JIT as a Service

PI: Björn Franke

Bio: Björn Franke holds a personal chair in Software Transformation in the School of Informatics. His research interests focus primarily on software tools for embedded systems, in particular optimising and parallelising compilers, instruction set simulators, design space exploration and performance estimation tools.

Summary:

In this project we rethink the foundations of how we provide JIT compilation to clients. We firmly believe that high-bandwidth, low-latency 5G communications is a game changer not only for the mobile phone market, but its impact will directly affect most areas of computing. By 2020, 50 billion smart devices are expected to be in use. 5G will help support the massive growth of the Internet-of-Things and enable devices to communicate with each other seamlessly through the convergence of mobile communications and computing. 5G is much more than a new communications standard. 5G's combination of high-speed wireless communications and efficient cloud computing means that even the tiniest devices can access virtually unlimited computing power. 5G communication liberates JIT compilation from the shackles of device constraints, e.g. limited processing speed, memory or energy in mobile devices. Instead, JIT compilation can be done anywhere in the cloud – from edge to data centre. This enables entirely new opportunities for JIT compilers, for example, by offloading JIT compilation, caching and sharing already compiled code between users and devices, or through the use of hardware accelerators deployed in edge or data centre servers and specifically designed to deliver highest code quality at low latency/power consumption.

Selected publications

- José Wesley De Souza Magalhães, Jackson Woodruff, Elizabeth Polgreen, Michael FP O'Boyle. "C2TACO: Lifting Tensor Code to TACO". The 22nd International Conference on Generative Programming: Concepts & Experiences, 2023.
- Jordi Armengol-Estapé, Jackson Woodruff, Chris Cummins, Michael F.P. O'Boyle. "SLaDe: A Portable Small Language Model Decompiler for Optimized Assembler." Accepted for publication in CGO'24.
- Alexander Brauckmann, Elizabeth Polgreen, Tobias Grosser, Michael F. P. O'Boyle. "mlirSynth: Automatic, Retargetable Program Raising in Multi-Level IR using Program Synthesis." The 32nd International Conference on Parallel Architectures and Compilation Techniques (PACT), October 2023.

SplitJIT:
Just-In-Time Compilation as a Service

Amir Khordadi[1], Kim Stonehouse[1], Björn Franke[1], Tom Spink[2]

[1]University of Edinburgh
[2]University of St Andrews

1. Introduction

Traditional approaches to translating source code into machine code involve ahead-of-time (AOT) compilation and interpretation. AOT compilation treats the entire program as a single unit, allowing for powerful optimizations and generating highly efficient code. However, this code is specific to a particular architecture, limiting its portability. On the other hand, interpretation translates and executes instructions at runtime, enabling the execution of source code without prior compilation. However, the process of considering instructions one at a time hinders advanced optimizations, resulting in slower code execution.

Just-In-Time (JIT) compilation offers a solution that combines the flexibility of interpretation with the speed of compilation. JIT compilers operate by considering groups of instructions, enabling more extensive optimizations compared to interpreters. Additionally, they have access to runtime information that is not available during AOT compilation, providing further optimization opportunities. Another advantage of JIT compilation is the ability to perform lazy compilation, which means that functions are compiled only when they are called, skipping those that are not invoked.

However, despite its advantages, JIT compilation also presents challenges. The process of code optimization itself takes time, and performing it at runtime may potentially offset the benefits of fast code execution.

In the following sections, we will introduce SplitJIT, a novel approach that tackles some of the limitations and trade-offs associated with JIT compilation, offering a promising solution for efficient code execution.

2. Motivation

In traditional compilation, there is a trade-off between compilation time and execution speed. However, Just-In-Time (JIT) compilation introduces a cyclic problem where reducing execution time increases compilation time, impacting overall performance.

To fully leverage the optimization benefits of JIT, minimizing compilation time is essential. Tiered compilation partially addresses this challenge by using multiple compiler tiers to deliver optimized code. However, managing multiple compilers within a single runtime is complex for developers.

JIT compilation also poses issues such as resource consumption and limited feasibility in resource-constrained environments like embedded systems. The cost of adding extra hardware for JIT acceleration in every embedded device quickly becomes prohibitive.

Moreover, when multiple devices compile the same applications in the same way, a centralized management or communication framework is necessary to address the problem effectively.

To overcome these challenges, we present SplitJIT, a novel solution that mitigates JIT limitations, reduces compilation latency, optimizes resource usage, and enables efficient compilation management across devices.

4. Architecture

The SplitJIT architecture consists of three main components: a Compiler for source code translation, a Runtime for execution management, and a Code Repository for storing application code.

This architecture reduces code redundancy and centralizes code parsing, benefiting both latency and bandwidth usage. Multiple instances of each component are possible, allowing flexibility in choosing the most optimal Compiler and Runtime combination. An orchestrator component can further optimize runtime selection and compilation strategies based on specific needs, such as minimizing execution time and reducing costs.

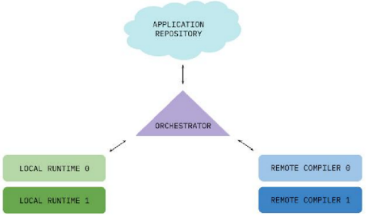
By adopting this architecture, SplitJIT improves code management, enables efficient runtime selection, and optimizes compilation strategies, resulting in enhanced performance and resource utilization.

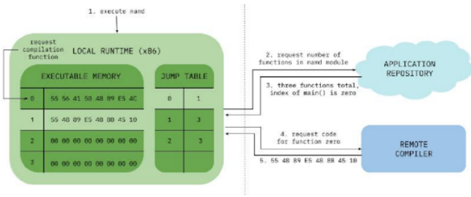
5. Implementation

In our implementation of SplitJIT, we utilize WebAssembly as the input format for its widespread adoption, language support, compactness, speed, universal compatibility, and hardware independence. To convert WebAssembly code to native code, we developed an LLVM frontend that translates it into LLVM Intermediate Representation (IR). Leveraging LLVM optimizations, we generate efficient native code for improved performance.

For efficient communication between components, we employ gRPC. gRPC enables seamless interaction and scalable communication among the Compiler, Runtime, and Code Repository components of SplitJIT.

By leveraging WebAssembly, employing LLVM, and utilizing gRPC for inter-component communication, our implementation ensures efficient code translation, optimized performance, and effective coordination within SplitJIT.





3. Proposal

Our proposal, SplitJIT, addresses the challenges of JIT compilation by decoupling it from program execution. This architecture offloads compilation to remote compilers, allowing local devices to focus solely on application execution.

To maintain performance, the remote compilers need to be fast enough to compensate for the added communication overhead. We propose three key approaches:

Caching Code: By caching compiled code between clients, compilation costs are shared, avoiding redundant compilations and improving efficiency.

Tiered Compilation: SplitJIT enables the use of a dedicated compiler, making tiered compilation more feasible. This simplifies runtime development, provides resources back to the application, and enhances overall performance.

Hardware Acceleration: SplitJIT makes hardware acceleration economical by sharing resources among clients, reducing costs while benefiting from the accelerated compilation.

These strategies optimize compilation time, resource utilization, and performance in JIT compilation. In the subsequent sections, we delve into the architecture and mechanisms of SplitJIT, highlighting its efficiency and potential for just-in-time compilation improvement.

6. Future Work

In future work, we plan to evaluate SplitJIT by quantifying the overhead introduced by network communication during compilation. This evaluation will guide our efforts to improve the compiler and offset network latency. We aim to introduce tiers with increased optimizations and explore the possibility of implementing a hardware-based initial tier. Additionally, we will leverage hardware acceleration for JIT studies to further enhance the system's performance. These future endeavors will drive the continuous advancement of SplitJIT in terms of efficiency and capabilities.

Data-Centric Parallelisation

PI: Björn Franke

Bio: **Björn Franke** holds a personal chair in Software Transformation in the School of Informatics. His research interests focus primarily on software tools for embedded systems, in particular optimising and parallelising compilers, instruction set simulators, design space exploration and performance estimation tools.

Summary: Automatic parallelisation is loop parallelisation. Traditional parallelising compilers attempt, for each loop individually, to identify a suitable parallel execution schedule, which honours dependencies dictated by the underlying sequential program semantics. In contrast, manual parallelisation is a holistic process. Human programmers consider the entire program and relationships between its individual components. It is standard practice for a human expert to initially rewrite and restructure a program before attempting parallelisation. We propose a fundamental paradigm shift by attempting to mimic what human experts would do: We aim to enable automatic parallelisation to incorporate whole program context and knowledge of the most widely used abstract data types in order to overcome the limitations of today's compilers. Our main aim is to develop a novel "data first" paradigm for automatic parallelisation of sequential legacy code, outperforming every existing parallelising compiler on irregular, pointer-based or control flow dominated applications. We aim to make automatic parallelisation a viable alternative to manual parallelisation, i.e. resulting in competitive parallel performance levels whilst reducing manual human intervention to a minimum.

Selected Publication

Björn Franke, Zhibo Li, Magnus Morton, Michel Steuwer. "Collection Skeletons: Declarative Abstractions for Data Collections". Proc. 15th ACM SIGPLAN International Conference on Software Language Engineering (SLE 2022), November 2022. **Best Paper Award.**

Collection Skeletons: Declarative Abstractions for Data Collections

Björn Franke[1], Zhibo Li[1], Magnus Morton[2], Michel Steuwer[1]

[1]University of Edinburgh
[2]Huawei Central Software Institute

1. Introduction

Modern programming languages provide programmers with rich abstractions for data collections as part of their standard libraries. Typically, these collections frameworks are organised as hierarchies that provide programmers with common abstract data types (ADTs) like lists, queues, and stacks. While convenient, this approach introduces problems which ultimately affect application performance due to users *over-specifying collection data types* limiting implementation flexibility. We develop **Collection Skeletons** which provide a **novel**, **declarative** approach to data collections. Using our framework, programmers *explicitly select properties* for their collections, thereby truly *decoupling specification from implementation*. By making collection properties explicit immediate benefits materialise in form of reduced risk of over-specification and increased implementation flexibility. We have prototyped our declarative abstractions for collections as a C++ library, and demonstrate that benchmark applications rewritten to use Collection Skeletons incur little or no overhead. We also show how Collection Skeletons help **shielding** the application developer from parallel implementation details, either by encapsulating *implicit parallelism* or through *explicit properties* that capture the requirements of parallel algorithmic skeletons. Across several benchmarks we observe performance speedups (on average between 2.57 to 2.93, and up to 16.37) resulting from the use of Collection Skeletons, while also enhancing performance portability across three different hardware platforms.

2. Properties

We identify eight groups of properties to help model the data collections. Properties belonging to the same group exhibit similar features or a twofold symmetrical dichotomy. We classify the properties into two categories - **semantics properties** and **interface properties**. Semantic properties specify the behaviour of the collections and methods with which collections are accessed or modified; Interface properties specify certain functionality, usually in form of access methods to be provided by the collection.

- Semantics Properties - **Uniqueness**, **Circularity**
- Interface Properties - **Iterable**, **Accessibility**, **Variability**, **Splitability**, **UnionFind**
- Hybrid Properties - **Order**

4. Implicit & Explicit Parallelism

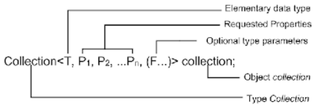
We also demonstrate that the Collection Skeletons also shield the parallelism from the users through,

- Implicit Parallelism
  - Transparent to the applications programmer
  - Hiding parallel implementations *behind the scenes*, e.g. parallel STL
  - Encapsulated parallelism, e.g. a parallel `find`
  - Just another collection implementation in our backend; no difference in application code
- Explicit Parallelism
  - Parallel algorithmic skeletons integration
  - Data-Parallel skeletons, e.g. iterable, random access
  - Divide & Conquer, e.g. splitable, by value or position; mergeable
  - Stencil, wavefront, and more

3. Prototyped Library and API

The Collection Skeletons library mainly consists of two components: (i) The property-based API, where the user declares properties as template arguments, acting as the **frontend** of the programming model; (ii) The template deduction mapping mechanism with the integration of various data structures as the **backend**.

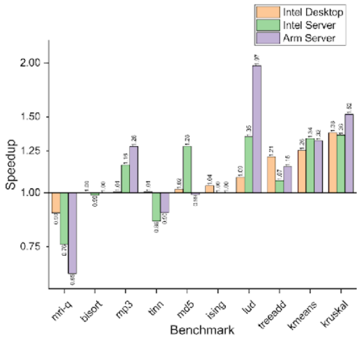
The first type parameter `T` is the type of the *elementary data* to be stored in the data collection. Following `T`, `P1`, `P2` to `Pn` are *properties* of the desired data collection. Variadic type parameters `(F...)` are *optional type parameters* that may need to be applied when declaring a data collection, e.g., a sorted data collection where the user request a self-defined comparator.

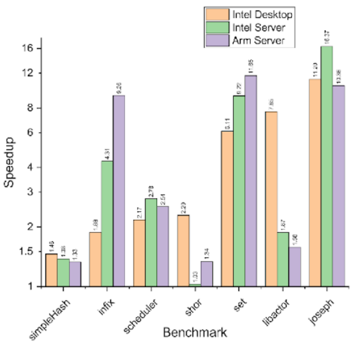


5. Evaluation

We evaluated our prototype library against 17 benchmark suites such as Olden benchmark. For the purpose of this evaluation we have manually rewritten the legacy benchmarks written in C, but have replaced the low-level user-defined data structures and their access functions with their equivalent collections from our framework. No other code rewriting has been performed and the same input data have been used to facilitate a like-for-like comparison. We ran the benchmark suites on three different hardware platforms (6-core Intel NUC 10, 72-core Intel Gold 6154 and 4-core Oracle Cloud Arm server) to compare the performance of the orginal programs and the rewritten programs.

The experimental results show that little-to-no overhead has been introduced by the prototyped library of Collection Skeletons. Some of the replaced benchmarks have much better performance than the baseline ones, suggesting that for some benchmarks, by replacing the data structures of the original benchmarks, the performance can be greatly improved. The average speedup of the 17 benchmarks across the three architectures is between 2.57-2.93.





6. Future Work

- Wider range of supported platforms, e.g. GPUs and other accelerators
- More on integration of Collection Skeletons and Parallel Algorithmic Skeletons
- Dynamic implementation adaptation at runtime
- Other problem domains: graphs, matrices,... with dynamic/data-dependent properties

For More Info

EdgeBoost: Data-Driven Traffic Engineering & Configuration Verification for Cloud Service Assurance

PI: Mahesh Marina

Bio: Mahesh Marina is a Professor in the School of Informatics at the University of Edinburgh and a Turing Fellow at the Alan Turing Institute in London. Before joining Edinburgh, he had a two-year postdoctoral stint at the UCLA Computer Science Department. He earned his PhD in Computer Science in 2004 from the State University of New York at Stony Brook. He has previously held visiting researcher positions at ETH Zurich and Ofcom London. He is a Distinguished Member of the ACM and a Senior Member of the IEEE.

Summary: Most networked end-user applications today are reliant on the cloud. To better serve their users, cloud service providers have their own private WAN infrastructure with footprint extending to the network edge. For cost and coverage reasons, such private WANs alone are insufficient to serve cloud based applications. So in practice private WANs and public Internet are used together (i.e., hybrid WAN) but this introduces non-determinism in application performance due to the use of public Internet. It is in this context that traffic engineering has a key role to play in cloud service assurance. In view of the above, the broad aim of this project is on end-to-end cloud service assurance in hybrid WAN via traffic engineering and network diagnosis.

Selected Publications:

L. Xue, M. K. Marina, G. Li and K. Zheng, “PAINT: Path Aware Iterative Network Tomography for Link Metric Inference,” in IEEE ICNP 2022.

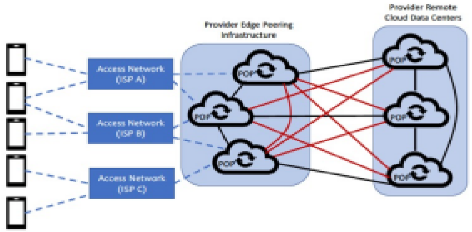
L. Xue, Z. Yuan, W. Ahsan and M. K. Marina, “TUBO: Robust Demand Matrix Forecasting for Proactive Network Traffic Engineering,” Submitted to IEEE INFOCOM 2024.

EdgeBoost: Data-Driven Traffic Engineering and Configuration Verification for Cloud Service Assurance

PI: Prof. Mahesh K. Marina
Project team members: Leyang Xue, Zhihang Yuan, Waleed Ahsan, Tanya Shreedhar

PROJECT CONTEXT AND MOTIVATION

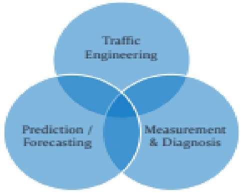
- Most networked end-user applications today are reliant on cloud
- Cloud service providers have private WAN footprint extending to the edge to better serve their users
- But in fact, private WAN *and* public Internet used together for cloud-based services for cost and coverage reasons
- Use of public Internet introduces non-determinism
- *Traffic engineering* has a key role to play in cloud service assurance


Provider Edge Peering Infrastructure
Provider Remote Cloud Data Centers
Access Network (ISP A)
Access Network (ISP B)
Access Network (ISP C)
POP

OBJECTIVE AND CHALLENGES

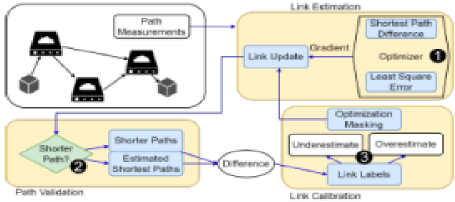
- Broad aim of the project is on end-to-end cloud service assurance in hybrid WAN via data-driven approaches for traffic engineering and network diagnosis

- Several challenges need to be addressed to meet this aim:
  - Accurate prediction of service demands and link qualities over time
  - Adaptive traffic engineering at service level
  - Identifying path segments that are faulty or performance bottlenecks
  - Enforcing and verifying control plane routing decisions over the data plane



PAINT: PATH AWARE ITERATIVE NETWORK TOMOGRAPHY [ICNP'22]

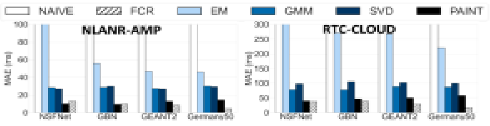
- Knowledge of link level performance key to assuring quality of cloud based and other services via TE, robust network operations and beyond
- Direct measurement of each link incurs too much overhead or is infeasible due to various reasons
- Existing approaches to network tomography make unrealistic assumptions on link stability, path controllability and visibility
- Motivated by this, we propose a new online iterative network tomography algorithm called PAINT



- With PAINT, the link metrics are iteratively estimated by minimizing their least square error (LSE) and calibrated based on the comparison of weight between the estimated shortest paths (SPs) and best-known paths from end-to-end path measurements
- ***Key insight:*** *when there is inconsistency between the estimated and measured paths, then weights of links on the estimated SP are likely mis-estimated, triggering a further round of estimation to refine the estimated link metrics*

PAINT EVALUATION

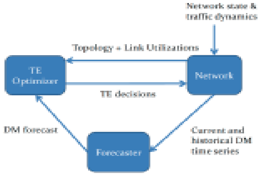
- We evaluated PAINT for link delay estimation using four different real network topologies and two real-world delay measurement datasets



- ***Takeaway:*** *PAINT yields up to 3x reduction in absolute link delay estimation error compared to existing approaches*

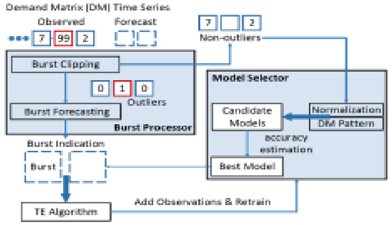
PROACTIVE NETWORK TRAFFIC ENGINEERING (TE)

- *Traditional and current approach to TE is reactive*
  - i.e., makes TE decisions based on current network state (observed traffic demands, ...)
  - Limited by the inability to anticipate and proactively respond to network state evolution
- ***Proactive TE paradigm*** can overcome this limitation
- ***Model Predictive Control (MPC) [1] view of TE***
- ***Robust demand matrix (DM) forecasting is a key challenge***


Network state & traffic dynamics
Topology + Link Utilizations
TE Optimizer
Network
TE decisions
DM forecast
Forecaster
Current and historical DM time series

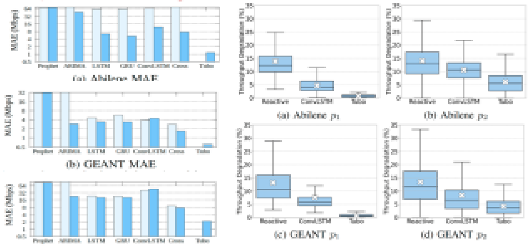
[1] S. Keshav, A Tutorial on Model Predictive Control, Dec 2020.

TUBO ML FRAMEWORK FOR ROBUST DM FORECASTING



TUBO EVALUATION

> 4x reduction in DM forecasting error



6-16x gain in TE efficiency

Performance Optimisation of Distributed Trusted Middleware System -- Enhancing the Speed of Byzantine Fault Tolerance using Stochastic Optimisation

PI: Tiejun Ma

Bio: Tiejun Ma holds a personal Chair in Financial Computing (Risk Modelling) in the School of Informatics at the University of Edinburgh. Prior to this he was Associate Professor at Southampton Business School, based in the Department of Decision Analytics and Risk. His research focuses on risk analysis and decision-making using quantitative modelling and data analysis techniques applied to FinTech, Cyber-Risk and resilience of distributed systems.

Summary: The Byzantine fault-tolerance (BFT) algorithm is a fundamental building block for distributed systems. Huawei's cloud and distributed system would need such a component to enhance Huawei online services dependability. However, BFT algorithms suffer from exponentially increasing message communication complexity, which restricts BFT algorithms' scalability and performance. We propose a parallel BFT (ParBFT) algorithm to tackle the scalability and performance challenges through a redesign of BFT to enable parallelisation as well as a mixed-integer programming (MIP)-based optimisation model. ParBFT partitions the entire consensus network into several subnets (i.e., consensus committees) which allows Byzantine consensus to run in parallel in each subnet. Together with the use of multi-signature techniques, ParBFT's message complexity is minimised to a constant level (i.e., O(c)) in each committee.

Selected Publications

- Btissam Er-Rahmadi and Tiejun Ma, "Data-driven mixed-Integer linear programming-based optimisation for efficient failure detection in large-scale distributed systems", European Journal of Operational Research 2021.

- Yifei Xie, Btissam Er-Rahmadi, Xiao Chen, Tiejun Ma and Jane Hillston, "A Stochastic Programming Approach for an Enhanced Performance of a Multi-committees Byzantine Fault Tolerant Algorithm", Euro-Par 2022 PhD Symposium

A Robust Optimization Model for Parallel BFT Consensus Algorithm

Yifei Xie, Tiejun Ma

School of Informatics, University of Edinburgh

Introduction

Byzantine fault tolerance (BFT) algorithms are a fundamental building block for distributed trusted systems (e.g. Distributed data management and blockchain), which requires high throughput of transactions and fault tolerance. To address scalability issues, the peer set is partitioned into multiple committees and execute consensus in parallel. This project aims to propose mathematical models to optimally parallelize BFT network. The robust optimization and stochastic programming model are designed to guarantee high performance and scalability in random delays and failures conditions.

The project adopts mathematical optimization techniques to model and explore the optimal trade-off of our BFT algorithm parallelization toward achieving 1040K tps transactions per second (TPS) without compromising trust requirements. The main contributions are highlighted:

- ***Parallel Algorithm***: ParBFT – a consensus algorithm with parallel architecture.
- ***Robust Optimization***: A robust optimization based clustering scheme to support BFT network.
- ***Failure Detection***: A function to optimize consensus and committee reconfiguration operations.

ParBFT Algorithm

Traditional BFT algorithms multicast messages with complexity. To improve the consensus efficiency, ParBFT is designed with the following advantages:

- ✓ ***Parallelism***: A parallel architecture based on stochastic optimisation seeing *Consensus Committee*, which can achieve *Prepare* operation in parallel via a multi-signature mechanism.
- ✓ ***Complexity***: A peer-to-peer communication designed for *Prepare* and *Commit* stages is able to reduce the message complexity from to .
- ✓ ***Scalability***: Each committee (*Consensus Committee* or *Verification Committee*) supports scalable increase of peers on the basis of a *Scalability Model* as seen in the following section.

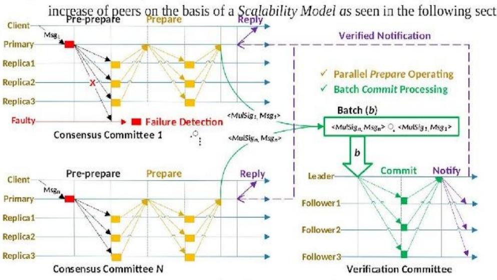


Fig. 1 ParBFT Algorithm Framework

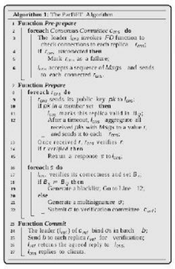
Correctness:

Theorem 1 (Safety): All non-faulty peers can commit processed Msgs/Txs in the same order.

Proof: The first round consensus in *Prepare* stage ensures each non-faulty peer agreeing on the order/correctness of Msgs/Txs; and then *Commit* stage achieves *consistency* by ensuring each non-faulty peer holding a backup of agreed Msg/Tx sequence.

Theorem 2 (Liveness): All requested Msgs/Txs are eventually replied to by the consensus peers.

Proof: 1). The *Prepare* stage ensures enough consensus responses collected from peers within a given timeout. 2). The *Commit* stage ensures that the verified Msgs are held by all non-faulty peers and a reply to the client within the timeout. 3). The scalability model guarantees the Byzantine/faulty peers can be replaced within a timeout. All above three techniques guarantees each Msg will be eventually processed and replied to.



```
Algorithm 1: The ParBFT Algorithm
1  Function Pre-prepare
2    foreach Consensus Committee $c_{cns}$ do
3      The leader $l_{cns}$ invokes FD function to check connections to each replica $r_{cns}$;
4      if $r_{cns}$ unconnected then
5        Mark $r_{cns}$ as a failure;
6      $l_{cns}$ accepts a sequence of Msgs and sends to each connected $r_{cns}$;
7  Function Prepare
8    foreach $r_{cns}$ do
9      $r_{cns}$ sends its public key pk to $l_{cns}$;
10     if pk in a member set then
11       $l_{cns}$ marks this replica valid in $B_0$;
12       After a timeout, $l_{cns}$ aggregate all received pks with Msgs to a value r, and sends it to each $r_{cns}$;
13     Once received r, $r_{cns}$ verifies r;
14     if r verified then
15       Return a response s to $l_{cns}$;
16   foreach s do
17     $l_{cns}$ verifies its correctness and set $B_s$;
18     if $B_s \neq B_0$ then
19       Generate a blacklist; Go to Line 12;
20     else
21       Generate a multisignature σ;
22       Submit σ to verification committee $c_{ver}$;
23 Function Commit
24   The leader ($l_{ver}$) of $c_{ver}$ bind σs in batch b;
25   Send b to each replica $r_{ver}$ for verification;
26   $r_{ver}$ returns the agreed reply to $l_{cns}$;
27   $l_{cns}$ replies to clients.
```

Scalability and Optimization Framework

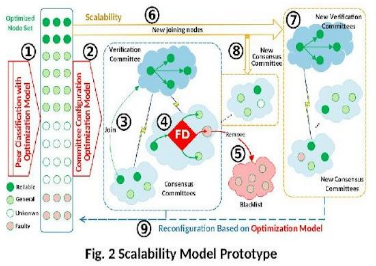


Fig. 2 Scalability Model Prototype

Sorting Mechanism (Steps ① and ② in Fig. 2) :

a) Generate various rating levels of peers based on optimization model;

b) Filter Byzantine peers and improve fault tolerance from to .

Failure-handling Mechanism (Steps ③ , ④ and ⑤):

c) Trace operations of each peer and blacklist failed and malicious nodes with FD ;

d) Upgrade/degrade peers for reconfigurations.

Scalable System Architecture (Steps ⑥, ⑦ and ⑧):

a) Accept joining peers and form a new consensus committee of an existing system;

b) Accept massive joining peers and form a new branch of verification and consensus committees.

Performance Evaluation of ParBFT with Stochastic Model

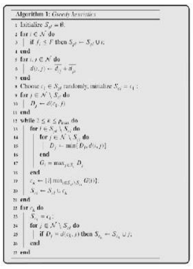
- ✓ **Test Environment:** Five Microsoft Azure E8s v3 8vCPUs Virtual Machines (South UK Located).
- ✓ **Throughput: 1040k tps** (Block size: 1 MB; Tx size: 100 Bytes; Node amount: 20 ~ 200).

Number of peers	Heuristics	Cplex Optimizer
50	0.041s	0.368s
100	0.069s	3.757s
150	0.096s	4.458s
200	0.127s	184.3s
500	0.489s	>2h
1k	0.885s	>2h
10k	12.389s	--
100k	140.37s	--

Tab. 1 Computation time of Heuristics

Key Benefits of ParBFT with Stochastic model:

- ✓ **>100%** increased throughput at 100 nodes; **>300%** increased throughput at 200 nodes, compared with currently fastest FastBFT.
- ✓ **Stable throughput** due to approximately constant-level message complexity with growing node count.
- ✓ **High scalability** based on the parallel architecture of ParBFT, which means that the ParBFT support new participants without serious performance loss.
- ✓ **Stochastic model** considers the randomness of network to optimally partition the peer set.
- ✓ **Heuristics** designed for optimization model to support large-scale distributed network, which can solve the configuration of 100k peer set.



```
Algorithm 1: Greedy heuristics
Initialize $S_{pl} = \emptyset$;
for $i \in \mathcal{N}$ do
  if $f_i \le F$ then $S_{pl} \leftarrow S_{pl} \cup i$;
end
for $i, j \in \mathcal{N}$ do
  $d(i,j) \leftarrow \overline{d_{ij}} + \overline{d_{ji}}$
end
Choose $c_1 \in S_{pl}$ randomly, initialize $S_{c_1} = c_1$;
for $j \in \mathcal{N} \setminus S_{pl}$ do
  $D_j \leftarrow d(c_1, j)$
end
while $2 \le k \le p_{max}$ do
  for $i \in S_{pl} \setminus S_{c_i}$ do
    for $j \in \mathcal{N} \setminus S_{c_i}$ do
      $D_j \leftarrow \min\{D_j, d(i,j)\}$
    end
    $G_i = \max_{j \in S_i} D_j$
  end
  $c_k \leftarrow \{i \mid \min_{i \in S_{pl} \setminus S_{c_i}} G(i)\}$;
  $S_{c_l} \leftarrow S_{cl} \cup c_k$
end
for $c_k$ do
  $S_{c_k} = c_k$;
  for $j \in \mathcal{N} \setminus S_{cl}$ do
    if $D_j = d(c_k, j)$ then $S_{c_k} \leftarrow S_{c_k} \cup j$;
  end
end
```

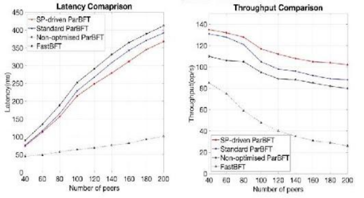


Fig. 3 Performance of current fastest BFT-like algorithms

The ParBFT with SP model increased 25% throughput compared to standard ParBFT. The heuristics designed support ParBFT network with 100k per set .

Robust Optimization (RO) Model for Trusted ParBFT

Decision variables:

: Integer variable represent the number of committees formed.

: Binary variables represent the type of the committee, equal to 0 if it only has TEE peers, equal to 1 if it has non-TEE peers.

: Binary variables equal to 1 if peer is the leader of peer ; equal to 0 if peer is not the leader of peer .

Objective function:

}

Peer set partition:

Bandwidth constraints:

Security and stability:

Switch between committee types:

- ✓ **Limitations of Stochastic Model:** SP model requires us to know the distribution of the random parameters, which is hard to estimate in the network environment. RO model only needs to know the value range.
- ✓ **Robustness of the model:** RO model provides the service manager the adjustable conserveness of the decision that is immune to uncertainties.
- ✓ **Reduced Complexity:** Compared with SP model, RO model does not require the scenarios of distributions.
- ✓ **Allocation of TEE peers:** The model proposed can also address how to allocate the TEE peers and utilize them to maximize the reliability and improve the performance.

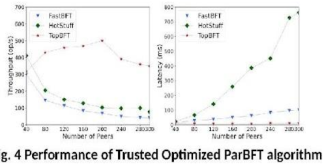
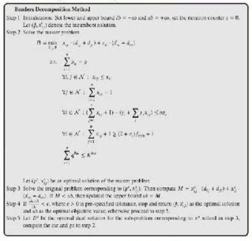


Fig. 4 Performance of Trusted Optimized ParBFT algorithm



Benders Decomposition Method

Step 1 Initialization: Set lower and upper bound $lb = -\infty$ and $ub = +\infty$, set the iteration counter $s = 0$. Let $(\hat{p}, x^{\hat{}}_{ij})$ denote the incumbent solution.

Step 2 Solve the master problem

$$lb = \min_{x_{ij}, p} \; x_{ij} \cdot (d_{ij} + d_{ji}) + x_{ij} \cdot (d_{ij} - d_{ji})$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ii} = p$$

$$\forall i, j \in \mathcal{N}: \; x_{ij} \le x_{ii}$$

$$\forall j \in \mathcal{N}: \; \sum_{i=1}^{n} x_{ij} = 1$$

$$\forall i \in \mathcal{N}: \; (\sum_{j=1}^{n} x_{ij} + 1) - (y_i + \sum_j t_j x_{ij}) \le n\sigma_i$$

$$\forall i \in \mathcal{N}: \; \sum_{j=1}^{n} x_{ij} + 1 \ge (2 + \sigma_i) f_{min} + 1$$

$$\sum_{i=1}^{n} q^{bw} \le K^{bw}$$

Let (p^s, x^s_{ij}) be an optimal solution of the master problem.

Step 3 Solve the original problem corresponding to (p^s, x^s_{ij}). Then compute $M = x^s_{ij} \cdot (d_{ij} + d_{ji}) + x^s_{ij} \cdot (d_{ij} - d_{ji})$. If $M < ub$, then updated the upper bound $ub = M$.

Step 4 If $\frac{ub - lb}{lb} < \epsilon$, where $\epsilon > 0$ is pre-specified tolerance, stop and return $(\hat{p}, x^{\hat{}}_{ij})$ as the optimal solution and ub as the optimal objective value; otherwise proceed to step 5.

Step 5 Let D^s be the optimal dual solution for the sub-problem corresponding to x^s solved in step 3, compute the cut and go to step 2.

	Set-up	Number of peers	Percentage of TEE peers	Uncertainty set	Main Purpose
A. Deterministic Optimization	1MB blockchain size 250 payloads	[20, 200]	50%	Original uncertainty set	Show the performance improvement of our proposed model.
B. Percentage of TEE peers	1MB blockchain size 250 payloads	200	[20%, 80%]	Original uncertainty set	Show the performance improvement with various percentages of TEE peers
C. Robust Optimization	1MB blockchain size 250 payloads	[20, 200]	50%	Random scheme v.s. Deterministic model v.s. Robust model	Present the performance under different settings robust uncertainty sets of random parameters.
D. Comparison between RO model and henders decomposition	1MB blockchain size 250 payloads	[20,200]	50%	Original uncertainty set	Compare the performance between the RO model and decomposition

THE UNIVERSITY *of* EDINBURGH

Argument Mining for Argumentation-based Decision Making

PI: Nadin Kokciyan

Bio: Nadin Kökciyan is currently a Lecturer in Artificial Intelligence at the School of Informatics and is a member of the Artificial Intelligence and its Applications Institute (AIAI). Her research interests include multiagent systems, agreement technologies (argumentation and negotiation), privacy in social software,AI Ethics, Explainable AI and Responsible AI.

Summary: Argument mining (AM) is a growing area that combines machine learning and natural language processing. AM is promising since it focuses on the automatic extraction of structured arguments and the relations between them. Combining AM together with computational argumentation techniques is an important step towards building explainable AI models. Combining logic-based approaches together with machine-learning approaches is a growing research area, such hybrid techniques can offer new ways to improve research in the field of explainable AI. The developed framework will be a general framework that can be adapted to various domains. Such a framework can be used as a decision-support system to help humans to make better decisions while providing explanations. In recent work, we show that conversational agents such as chatbots are useful in facilitating communication with humans to assist them in their decision-making. Or such a framework could be used to provide a summary to the user. For example, on Twitter, a user may need a summary about a specific topic, and this framework could be used to display a set of arguments and counter-arguments while providing further explanations.

Selected Publications:

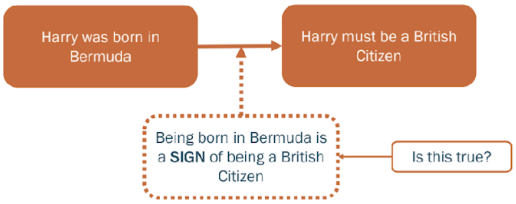
- Sandrine Chausson, Ameer Saadat-Yazdi, Xue Li, Jeff Pan, Vaishak Belle and Nadin Kökciyan, N.; and Ross, B. A Web-based Tool for Detecting Argument Validity and Novelty. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pages 3053–3055, 2023.
- Ameer Saadat-Yazdi, Jeff Z. Pan, and Nadin Kokciyan. “Uncovering Implicit Inferences for Improved Relational Argument Mining”, in the Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL-23), pages 2476–2487, 2023.
- Ameer Saadat Yazdi, Xue Li, Sandrine Chausson, Vaishak Belle, Bjorn Ross, Jeff Z. Pan and Nadin Kokciyan, "KEViN: A Knowledge Enhanced Validity and Novelty Classifier for Arguments", in the Proceedings of the 9th Workshop on Argument Mining, pp. 104-110

Constructing A Dataset of Argument Types

Ameer Saadat-Yazdi, Jeff Z. Pan, Nadin Kökciyan

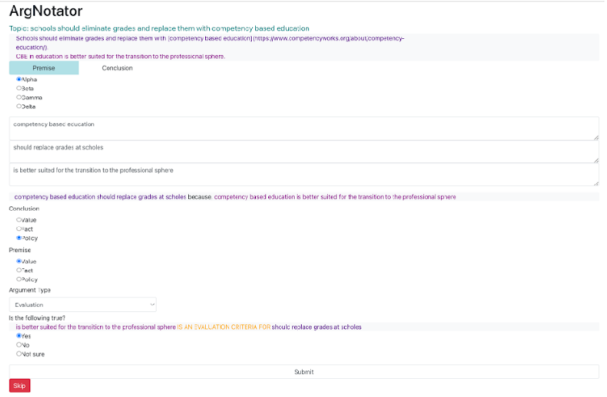
Background

- An argument is a form of verbal reasoning in which a claim is backed by a premise
- Automatic identification of argument structure allows us to reason about the validity of argument and resolve conflicting information
- The challenge is that people often leave out or assume information when arguing
- By identifying the type of argument being used we can identify the assumption that the arguer is making
- We can use this information to determine whether the argument is valid


Harry was born in Bermuda
Harry must be a British Citizen
Being born in Bermuda is a **SIGN** of being a British Citizen
Is this true?

ArgNotator

- There is currently no high quality dataset of argument types available to analyse and train models on
- Building such a dataset is difficult because of inter-annotator disagreement and the need for expert knowledge
- So, we designed an annotation tool specifically for this task to simplify the process for non-expert annotators



ArgNotator

Topic: schools should eliminate grades and replace them with competency based education

Schools should elminate grades and replace them with [competency based education](https://www.competencyworks.org/about/competency-education/).
CBE in education is better suited for the transition to the professional sphere.

Premise | Conclusion

● Alpha
○ Beta
○ Gamma
○ Delta

competency based education

should replace grades at scholes

is better suited for the transition to the professional sphere

competency based education should replace grades at scholes because competency based education is better suited for the transition to the professional sphere

Conclusion
○ Value
○ Fact
● Policy

Premise
● Value
○ Fact
○ Policy

Argument Type
Evaluation

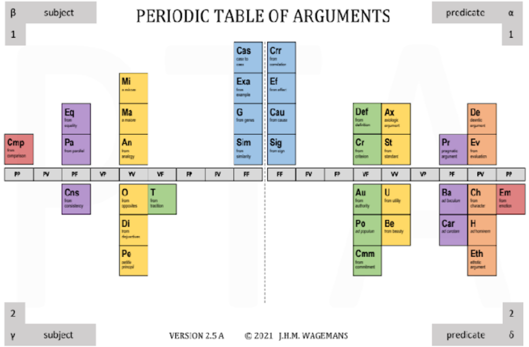
Is the following true?
is better suited for the transition to the professional sphere IS AN EVALUATION CRITERIA FOR should replace grades at scholes
● Yes
○ No
○ Not sure

Submit

Skip

The Periodic Table of Arguments

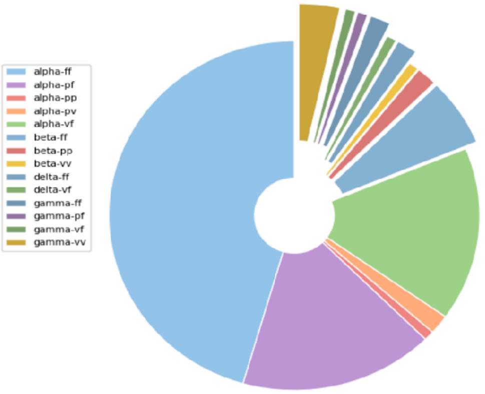
- The Periodic Table of Arguments [1] is a theoretical framework for the identification of argument types
- It breaks down the identification of arguments into three steps:
  1. Identification of the argument form $(\alpha, \beta, \gamma, \delta)$
  2. Identification of the argument substance (Policy (P), Fact(F), Value(V))
  3. Identification of the argument type



For more info please visit www.periodic-table-of-arguments.org

Future Work

- We plan on annotating 700 arguments by the end of the year
- This annotations will allow us to gain more insight into the syntactic structure of argument types
- We plan on using this insight to build models to classify and evaluate arguments based on their type



Distribution of argument types in the current annotated dataset of 200 arguments

[1] Wagemans, J. (2016, May). Constructing a periodic table of arguments. In *Argumentation, objectivity, and bias: Proceedings of the 11th international conference of the Ontario Society for the Study of Argumentation (OSSA), Windsor, ON: OSSA* (pp. 1-12)

THE UNIVERSITY of EDINBURGH informatics
aiai Artificial Intelligence and its Applications Institute
HUAWEI

Efficient Training of Giant Neural Networks

PI: Luo Mai

Bio: Luo Mai is an Assistant Professor (Lecturer) in the School of Informatics at theUniversity of Edinburgh. He is a member of the Institute of Computing Systems Architecture where he leads the Large-Scale System Software Group. Mai is the founder of the Open Machine Learning System Community, serving as an educational platform that fosters the development of AI system software. He has contributed to the field of AI systems through his textbook Machine Learning Systems: Design and Implementation, published by the Springer Nature in English and Tsinghua University Press in Chinese

Summary: Large deep learning models have achieved unprecedented performance in many key data and AI applications recently. The training of these models, however, often requires tremendous computational resources, which prevent them from being widely adopted in practice. In this project, we will explore how to design a system that can efficiently train giant neural networks. Such a system will have novel algorithms that can effectively partition the networks and coordinate the training on distributed nodes. It will also have a high-performance runtime that can efficiently exploit distributed heterogeneous processors and memory in training the network.

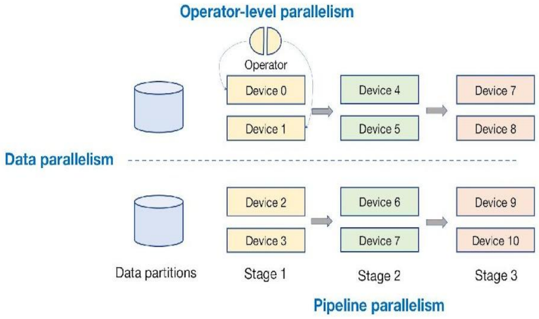
Selected Publications

- HSE-Parallel: A Fast Parallelism Planner for Giant Neural Networks, 2022
- TorchOpt: An Efficient Library for Differentiable Optimization, JMLR 2023

Efficient Distributed Training of Large AI Models

Luo Mai, Man-Kit Sit, Congjie He
University of Edinburgh

Many strategies are available to parallelise the training of large AI models

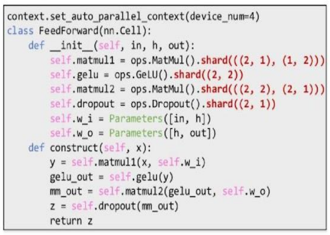


Large AI models: Trillions parameters (e.g., GPT-4, Switch transformers)

Parallel training: 100s parallel CPU/GPU devices must be coordinated to overcome memory limits

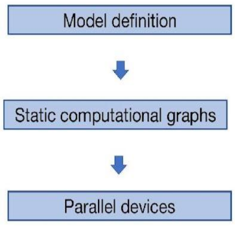
Research question: How to find best parallelism strategies that can minimise model training time?

State-of-the-art depend on either **manual configuration** or **computational graphs**



```
context.set_auto_parallel_context(device_num=4)
class FeedForward(nn.Cell):
    def __init__(self, in, h, out):
        self.matmul1 = ops.MatMul().shard(((2, 1), (1, 2)))
        self.gelu = ops.GeLU().shard((2, 2))
        self.matmul2 = ops.MatMul().shard(((2, 2), (2, 1)))
        self.dropout = ops.Dropout().shard((2, 1))
        self.w_i = Parameters([in, h])
        self.w_o = Parameters([h, out])
    def construct(self, x):
        y = self.matmul1(x, self.w_i)
        gelu_out = self.gelu(y)
        mm_out = self.matmul2(gelu_out, self.w_o)
        z = self.dropout(mm_out)
        return z
```

Manual configuration
(DeepSpeed, Megatron)



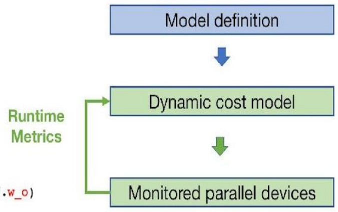
Computational graph dependency
(Alpa, MindSpore Auto-Parallel)

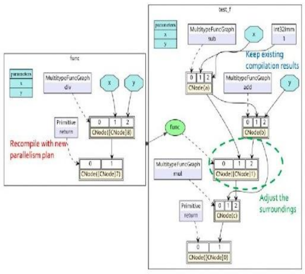
Issues:

- Manual configuration requires static parallelism rules, which can not be generalized
- Dependency on computational graphs makes popular dynamic execution lack parallelism optimization

Designing an **adaptive dynamic parallelism system** for large AI models

```
@auto_parallel
class MyModel(nn.Cell):
    def __init__(self, in, h, out):
      self.matmul1 = ops.MatMul()
      self.gelu = ops.GeLU()
      self.matmul2 = ops.MatMul()
      self.dropout = ops.Dropout()
      self.w_i = Parameters([in, h])
      self.w_o = Parameters([h, out])
    def construct(self, x):
      y = self.matmul1(x, self.w_i)
      gelu_out = self.gelu(y)
      mm_out = self.matmul2(gelu_out, self.w_o)
      z = self.dropout(mm_out)
    return z
```





#1: **Transparent parallelism** for dynamic execution programs

#2: Augmenting dynamic execution with **runtime parallelism metrics**

#3: **Dynamic scheduling** for parallelism tasks

Learning to infer missing graphical knowledge and produce better predictions from multimodal data and incomplete graphs

PI: Amos Storkey

Bio: Amos Storkey is Professor of Machine Learning and AI in the School of Informatics, University of Edinburgh and an ELLIS Fellow. He now leads a research team focused on deep neural networks, transfer learning, efficient learning and inference. He balances work on fundamental methods in machine learning and applications in medical imaging and geosciences. He is currently director of the EPSRC Centre for Doctoral Training in Data Science, was AI Lead for the Boneseyes Horizon Project. He was Programme Chair for the AI and Statistics conference.

Summary: The research challenge centres on the efficient management and interpretation of increasing volumes of multimodal data. The complexity and variety of today's data, including video, online meetings, and internet traffic, make it uniquely challenging to process and extract valuable insights. Crucially, the challenge also involves inferring relationships within and between datasets, such as inferring a personal knowledge graph in private image/video collections to aid personalized search.

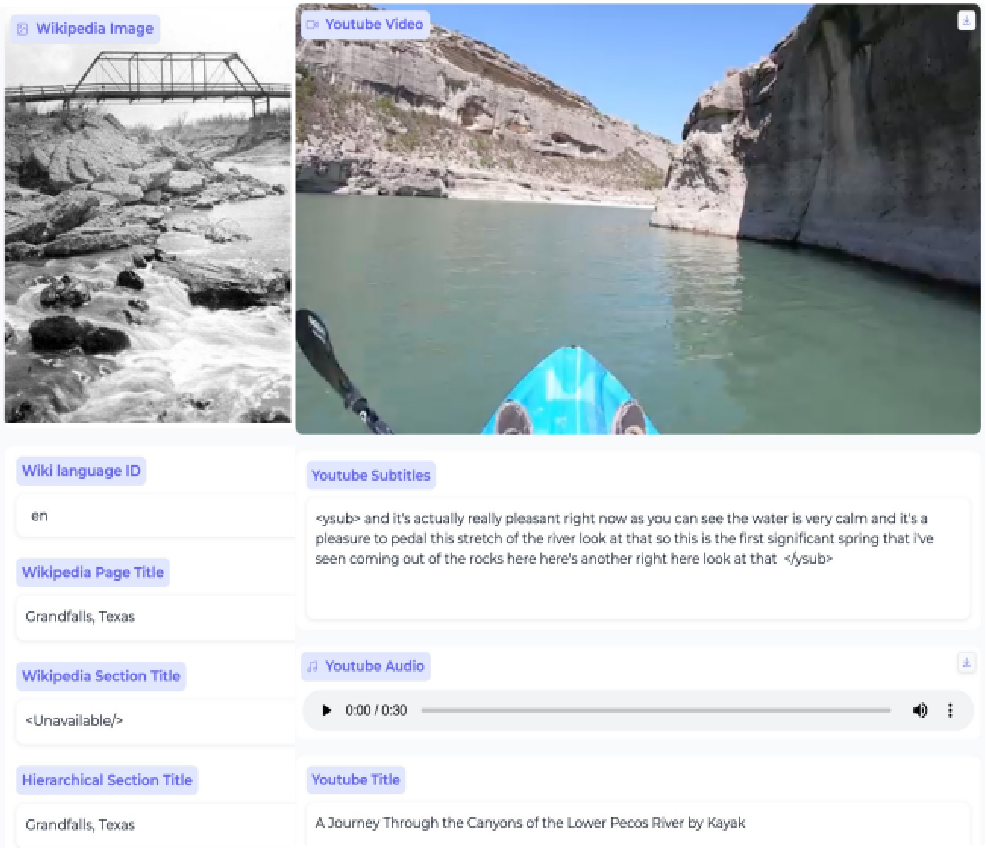
Selected publications:

- Fontanella, A., Antoniou, A., Li, W., Wardlaw, J., Mair, G., Trucco, E. and Storkey, A., 2023. ACAT: Adversarial Counterfactual Attention for Classification and Detection in Medical Imaging. *ICML 2023*
- Jelley, A., Storkey, A., Antoniou, A. and Devlin, S., 2023. Contrastive Meta-Learning for Partially Observable Few-Shot Learning. ICLR 2023
- Rezk, F., Antoniou, A., Gouk, H. and Hospedales, T., 2023. Is Scaling Learned Optimizers Worth It? Evaluating The Value of VeLO's 4000 TPU Months. Neurips 2023, ICBINB Workshop.

TALI: A Large Scale Quadra-Modal Dataset consisting of Temporally and Semantically Aligned Audio, Language and Images

Antreas Antoniou, Eleni Triantafyllou, Hugo Larochelle, Jeff Pan, Yi Liu, and Amos Storkey

- Problem: Current self-supervised learning models primarily use uni-modal datasets or multi-modal datasets that lack temporal alignment across modalities. This gap restricts models from fully exploiting multi-modal learning and understanding real-world temporal dynamics.
- Proposal: We're launching TALI, a tetramodal dataset. It bridges this gap by providing semantically and temporally aligned video, images, audio, and text from YouTube videos and Wikipedia articles. This alliance facilitates a wide array of self-supervised tasks and bolsters multi-modal research.
- Benefits: TALI propels advancements in multi-modal learning and self-supervised task creation. Its temporally aligned modalities enable models to better interpret real-world dynamics and temporal sequencing, vital for many applications. It also offers semantically aligned Wikipedia text and images, offering richer context, contributing to a holistic learning environment.
- Collection Methodology: To curate TALI, we initiated with the Wikipedia Image to Text (WiT) dataset. With the help of CLIP models, we utilized semantic alignment techniques to extract multilingual content from YouTube under Creative Commons licensing. We segmented videos into 30-second clips, ranked them based on relevance, and conserved useful metadata - resulting in a wealth of fine-grained multimodal data points.



Wikipedia Image

Youtube Video

Wiki language ID

en

Wikipedia Page Title

Grandfalls, Texas

Wikipedia Section Title

<Unavailable/>

Hierarchical Section Title

Grandfalls, Texas

Youtube Subtitles

<ysub> and it's actually really pleasant right now as you can see the water is very calm and it's a pleasure to pedal this stretch of the river look at that so this is the first significant spring that i've seen coming out of the rocks here here's another right here look at that </ysub>

Youtube Audio

Youtube Title

A Journey Through the Canyons of the Lower Pecos River by Kayak

EPOCH: Effectful programming on capability hardware

Note: This is a new project and is either about to start or has only just started

PI: Sam Lindley and Ian Stark

Bios:

Sam Lindley is a Reader in Programming Language Design and Implementation and holds a UKRI Future Leaders Fellowship in Effect Handler Oriented Programming. He also provides senior technical consultancy to Huawei through his company Effect Handlers Limited.

Ian Stark is a Senior Lecturer in Computer Science at the University of Edinburgh. His research is on mathematical models for programming languages and concurrent systems, ranging from formal specification of instruction set architectures to biochemical modelling with the continuous pi-calculus.

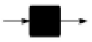
Summary: Effect handlers are a powerful programming-language abstraction for working with computational effects like exceptions, concurrency, and probabilistic programming. Current implementations of effect handlers use a range of methods to map them to existing hardware features, either through libraries or compiler transformations. We propose to replace these with a direct translation using the hardware-checked capabilities provided by CHERI: a novel machine architecture that efficiently enforces resource guarantees (such as memory protection and software compartmentalization) in hardware.

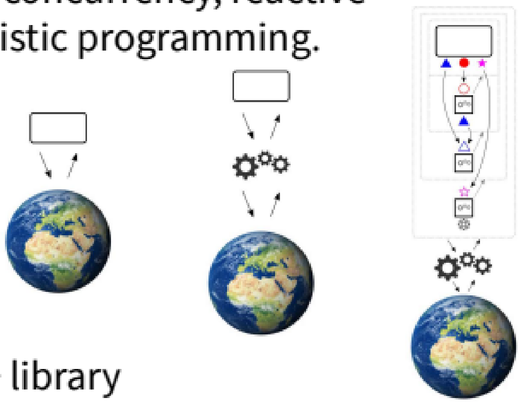
EPOCH: Effectful Programming on Capability Hardware

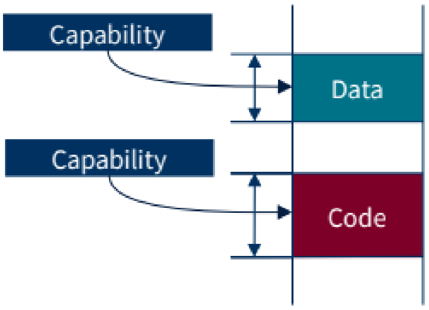
THE UNIVERSITY *of* EDINBURGH
informatics

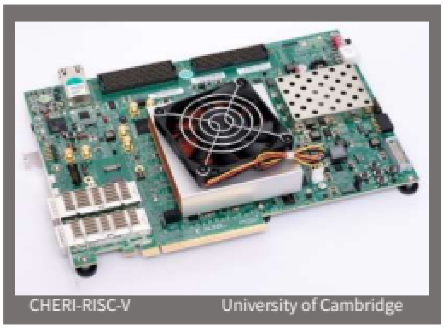
Effect Handlers let programmers define, customize, and compose a wide range of *computational effects* as first-class language abstractions: giving flexible access to features such as lightweight concurrency, reactive programming, dynamic binding, and probabilistic programming.

- **GitHub:** `semantic` code analysis library
- **Facebook:** *React UI* library
- **Uber:** *Pyro* statistical inference
- **OCaml 5:** Concurrency
- **WasmFX:** Effects for web assembly
- **Huawei Edinburgh:** `cpp-effects` C/C++ library








CHERI-RISC-V University of Cambridge

CHERI is an instruction set architecture that replaces integer addresses with richly structured *capabilities*. These enable principles of *least privilege* and *intentionality*: code executes with only the capabilities it needs and must be explicit in using them.

- **Memory safety:** no overflows, out-of-bounds
- **Control-flow integrity:** sealed capabilities
- **Compartmentalization:** containers & objects

CHERI capabilities are low-overhead; hardware-enforced; fine-grained; and with a rich permission structure.

EPOCH will use hardware capabilities to directly implement effect handlers, mapping their sophisticated control of interacting computational effects down to low-level guarantees on machine execution.

- **C/C++ libraries:** `cpp-effects`, `libhandler`, `libmprompt`
- **Research languages:** Koka, OCaml
- **Enhanced capabilities:** linear, uninitialized, directed
- **Reliability:** isolation and recovery of application/handler code
- **Rich handlers:** dynamic, multishot, customizable

Sam Lindley

Ian Stark

Hardware-accelerated reliable replication protocols

Note: This is a new project and is either about to start or has only just started

PI: Boris Grot

Bio: Boris Grot is a Professor in the School of Informatics at the University of Edinburgh, where he leads the EASE Lab (Edinburgh Architecture and Systems Lab). His research focuses on understanding and alleviating efficiency bottlenecks and capability shortcomings of processing platforms for data-intensive applications. Boris is a member of the MICRO Hall of Fame and a recipient of multiple awards for his research. He was the Program Co-Chair for MICRO 2022 and is the General Chair for HPCA 2024.

Summary: This project addresses the need for a high-performance and fault-tolerant replication protocol at datacenter scale. Our observation is that while data volumes continue to increase, improvements in CPU performance are stagnating. This means that existing replication protocols, which run in software on CPUs, may become major performance bottlenecks going forward.

The key hypothesis explored in the project is that – in the absence of failures – strongly-consistent replication protocols resemble cache coherence protocols. The latter have been shown to be efficiently handled through simple state machines fully implemented in hardware. Our goal is to apply the rich knowledge from decades of hardware-offloaded cache coherence protocols to distributed systems at datacenter scale. To that end, we will develop and prototype a consistency controller, a hardware-accelerated protocol engine on the NIC that will handle consistency actions, thus freeing the CPU to work exclusively on the application logic. We will further study the space of consistency protocols to understand which protocols are fundamentally better suited toward hardware offload.

THE UNIVERSITY *of* EDINBURGH
informatics

HUAWEI

Hardware-accelerated reliable replication protocols

M.R. Siavash Katebzadeh, Vijay Nagarajan, Boris Grot
University of Edinburgh

Antonis Katsarakis, Vasilis Gavrielatos, Nikos Ntarmos
Huawei Research

Background

Distributed Datastores

riak redis mongoDB M

- Backbone of modern online services
- Read/write API

Available

Data replication for fault tolerance

Consistent

Programmability → Consistent replicas

Performant

Exploit replicas for low latency & high throughput

Reliable Replication Protocols

- Defines actions upon reads and writes
  - Purpose: maintain consistency and fault-tolerance
  - → Determines datastore's performance
- Ensuring fault tolerance:
  - Keep multiple replicas within a datacenter
  - Faults are infrequent → few replicas suffice [1]
- Keeping replicas consistent:
  - Trade-off: consistency guarantees vs performance and programmability

[1] Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network. SIGCOMM'15

Consistency Trade-offs

	Programmability	Performance
Weak consistency	✗	✓
Strong consistency	✓	✗

How to provide both strong consistency and high performance?

Insight: Strongly-consistent replication protocols resemble cache coherence protocols

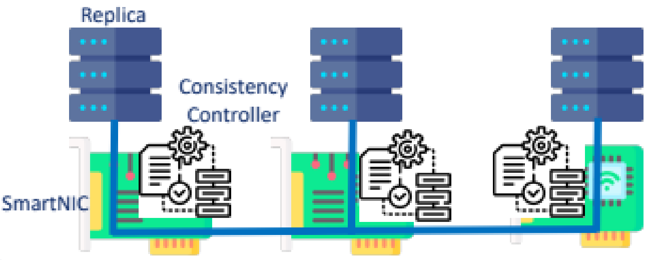
Idea: Offloading consistency actions into SmartNICs frees CPU to work on application logic
→ Improves performance

Dual-mode consistency protocol

- **Fast path:** hardware-offloaded consistency enforcement in the absence of failures
- **Slow path:** software-based consistency enforcement in the presence of infrequent cases of failure and network fault

Consistency controller

Hardware-accelerated protocol engine integrated into NICs



Incremental Evaluation of Property Graph Queries

Note: This is a new project and is either about to start or has only just started

PI: Milos Nikolic

Bio: Milos Nikolic is a lecturer in database systems in the School of Informatics, University of Edinburgh. His research interests are in databases and large-scale data management systems. He has recently made contributions to in-database learning, stream processing, incremental computation, and query compilation.

Summary: Graph database systems such as Neo4J, TigerGraph, and JanusGraph serve as the backbone of many applications that need to store and analyse graph-structured data. In a growing number of domains – e.g., financial fraud detection, recommendation systems, and social platforms – graph applications rely on complex queries and require real-time responses when the underlying data changes. Existing graph database systems, unfortunately, provide limited support for the incremental evaluation of graph queries. For example, Neo4J supports graph updates but lacks the support for automatic incrementalisation of graph queries, meaning that the user needs to explicitly specify how changes are propagated for a given query; this approach is tedious and error-prone. Recent work by Szárnyas et al. proposes transforming property graph queries to relational algebra queries and using existing database techniques for their incremental evaluation. This work, however, exposes limitations that may restrict its adoption in practice: for example, it considers only bounded graph reachability queries with a naïve translation to relational algebra, disregards optimisation opportunities arising from the structure of the query, and misses out on recent developments in the areas of join processing and incremental query evaluation (e.g., worst-case optimal join algorithms). Thus, automatic and practical incremental evaluation of graph queries remains an open problem.

Incremental Evaluation of Property Graph Queries

STARTS IN
JANUARY 2024

PI: Milos Nikolic
milos.nikolic@ed.ac.uk

THE UNIVERSITY *of* EDINBURGH
informatics

HUAWEI

Property Graphs

Powerful data model for highly-connected data

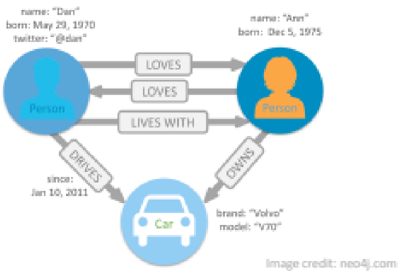


Image credit: neo4j.com

Property graph queries extract information

```
MATCH (:Person {name: "Dan"}) - [:DRIVES] -> (car)
RETURN car;
```

Challenges

Scale

Large graphs make query execution slow and resource-intensive

Complexity

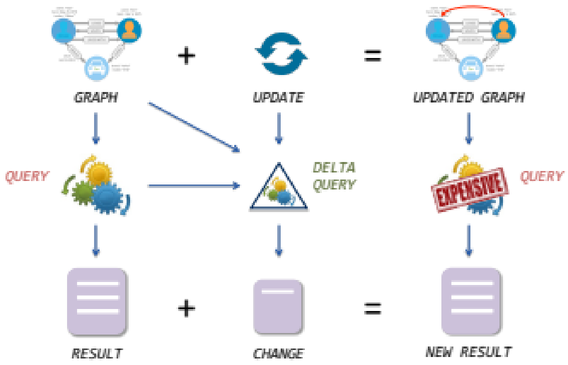
Querying complex patterns can be time-consuming

Updates

Graphs often change, requiring re-evaluation of queries

Reevaluating complex queries is inefficient!

Incremental Evaluation of Property Graph Queries



Improved efficiency

Faster execution and better resource utilization

Real-time responsiveness

Enables real-time updates to query results

Scalability

Could scale well with the size of property graphs

No support for automatic incremental evaluation in existing graph database systems

Project Goals

Identify classes of graph queries that admit efficient incremental evaluation

Akin to *q*-hierarchical queries in relational DBs
Graph databases exhibit different types of updates

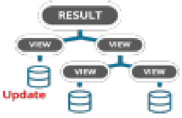
Derive incremental queries via algebraic rules

Extension of relational algebra with a fixpoint operator
Sufficient feature coverage of a practical graph query language (e.g., openCypher)

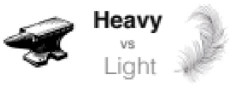
Devise an incremental-friendly data representation

Existing representations (e.g., CSR) are optimized for efficient storage and traversal but not for updates

Propose maintenance strategies for graph queries



Higher-order computation
[VLDB 2014, SIGMOD 2016, 2018, 2020]



Adaptive computation
[ICDT 2019, PODS 2020]

RobustCheck: Testing Robustness of Compiler Optimisations and Deep Learning Framework

Note: This is a new project and is either about to start or has only just started

PI: Ajitha Rajan

Bio: Ajitha Rajan is a Reader in the School of Informatics, University of Edinburgh. Prior to arriving in Edinburgh, she was a post-doc at Oxford University and at Laboratoire d'Informatique de Grenoble in France. She graduated with a PhD in Computer Science from the University of Minnesota. Her interests include: automated software testing techniques considering test generation, test oracles, test coverage metrics and biomedical artificial intelligence focusing on cancer survival models, interpretability for biological sequences and medical images.

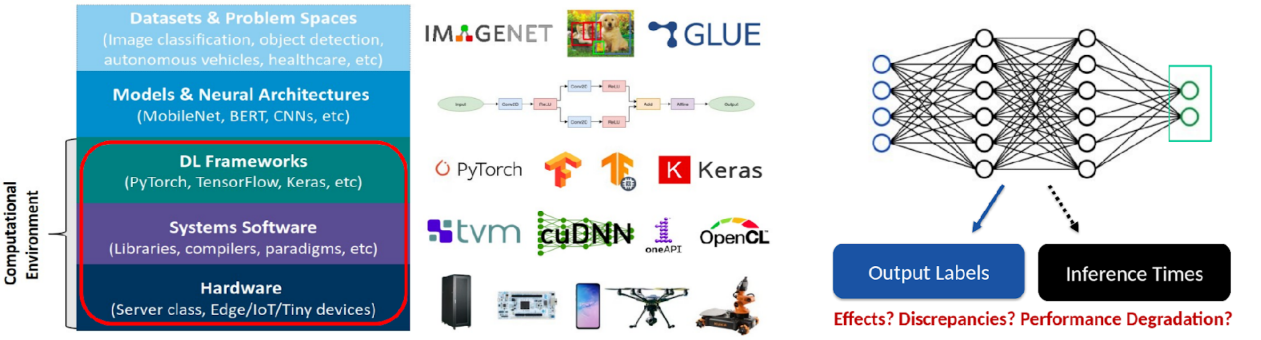
Summary: Compilers and deep learning (DL) frameworks play a fundamental role in aiding deployment of a deep learning model on a device. The process of model deployment involving the frameworks and compiler is not without errors. For instance, a study involving 3023 Stack Overflow posts built a taxonomy of faults and highlighted the difficulty of DNN deployment [4]. Another study explores the effect of DNN faults on mobile devices by identifying 304 faults from GitHub and Stack Overflow[5]. Existing techniques for testing models fail to consider interactions with the underlying computational environment -- conversions between Deep Learning (DL) frameworks (e.g., TensorFlow, PyTorch,TensorFlow Lite) and compiler optimizations (e.g., operator fusion, loop unrolling, etc.). The application domain for this project is a prediction model for cancer vaccine design. The ability to predict whether a peptide will get presented on MHC Class I molecules is a critical step in designing vaccines so they can activate the immune system to destroy the invading disease protein. Cancer vaccines targeting tumour proteins have become the main direction of tumour vaccine in recent years. We aim to provide testing capabilities and quality measurement for prediction models in vaccine design that will assess robustness with respect to changes in DL frameworks and compiler optimisations in the computational environment and use it to provide an operational boundary.

Differential Testing, Fault Localization, and Repair of Image Recognition Models

Nikolaos Louloudakis[1], Perry Gibson[2], José Cano[2], and Ajitha Rajan[1]

[1]University of Edinburgh, [2]University of Glasgow, Scotland, UK

Problem



Datasets & Problem Spaces
(Image classification, object detection, autonomous vehicles, healthcare, etc)

Models & Neural Architectures
(MobileNet, BERT, CNNs, etc)

DL Frameworks
(PyTorch, TensorFlow, Keras, etc)

Systems Software
(Libraries, compilers, paradigms, etc)

Hardware
(Server class, Edge/IoT/Tiny devices)

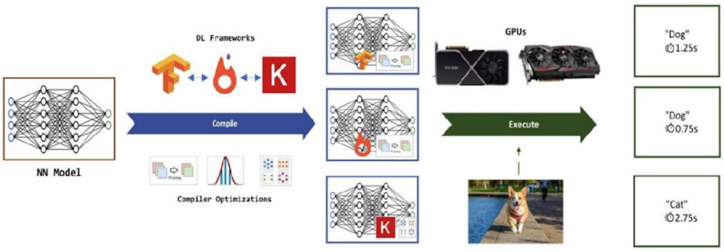
Computational Environment

Output Labels

Inference Times

Effects? Discrepancies? Performance Degradation?

Differential Testing



Experiments

3 Neural Network Models
InceptionV3, ResNet101, MobileNetV2

4 GPU Devices
1 NVidia Tesla, 1 NVidia Volta, 1 GEN9 HD Graphics, 1 Arm Mali-G72

Conversions Across 4 Deep Learning Frameworks
Keras, PyTorch, TensorFlow, TFLite

3 Compiler Optimization Settings
Basic (Simplify Expressions)
Default (Operator Fusion, etc)
Extended (Fastmath, etc)

ILSVRC2017 Object Detection Test Dataset
5.5k images, 224x224

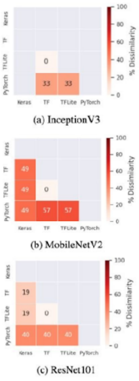
Results

Robustness of Output Label Prediction

- **DL Framework (Conversion) Effect:**
  - MobileNetV2: up to 68% (PyTorch to TF) 🟦
  - ResNet101: up to 72% (PyTorch to TF) 🟩
  - InceptionV3: up to 32% (PyTorch to Keras) 🟨
- **Compiler Optimization Effect:**
  - No output label discrepancies

Robustness of Model Inference Time

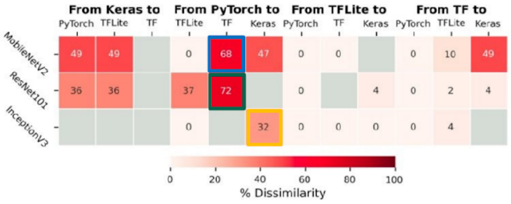
- **DL Framework Effect:**
  - 4-16% Performance Discrepancies
- **Compiler Optimization Effect:**
  - Up to 54% improvement (expected)
  - Up to 81% unexpected performance degradation
  - Varying behavior across models and devices



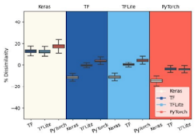
(a) InceptionV3

(b) MobileNetV2

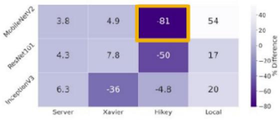
(c) ResNet101



DL Framework Conversion Effect (Output Label Prediction)

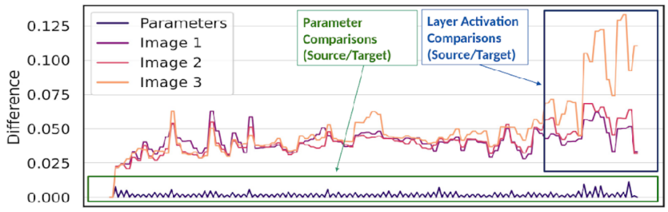


DL Framework Backend Effect (Output Label Prediction (Left) & Inference Time (Top))



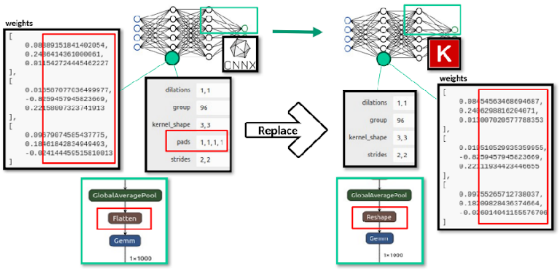
Compiler Optimization Effect (Inference Time)

Fault Localization



Conversion of TF to TFLite for InceptionV3 (Comparison of Conv2D & Bias Add Layers)
Effective Repair Strategy: Replace Source model parameters to Target

Fault Repair



THE UNIVERSITY of EDINBURGH

Nikolaos Louloudakis
n.louloudakis@ed.ac.uk

University of Glasgow

DBShaker: Smart Recompilation for Adaptive Analytical Query Processing

Note: This is a new project and is either about to start or has only just started

PIs: Antonio Barbalace, Amir Shaikhha

Bios:

Antonio Barbalace is a Senior Lecturer in Operating Systems in the School of Informatics. His research interests include operating systems, virtualization environment, compiler, linker, runtime systems for parallel, distributed, and heterogeneous computer architectures (including near data processing platforms), real-time and general-purpose scheduling, targeting large deployments (data-center) as well as small-devices (embedded/IoT).

Amir Shaikhha is an Assistant Professor (Lecturer) in the School of Informatics at the University of Edinburgh. His research focuses on the design and implementation of data-analytics systems by using techniques from the databases, programming languages, compilers, and machine learning communities. Prior to that, he was a Departmental Lecturer at Oxford. He earned his Ph.D. from EPFL in 2018, for which he was awarded a Google Ph.D. Fellowship in structured data analysis, as well as a Ph.D. thesis distinction award. He has won the Best Paper Award at GPCE 2017 and the Most Reproducible Paper Award at SIGMOD 2017. He (co-)chaired the program committees of DBPL 2021, Scala 2022, DRAGSTERS 2023, and GPCE 2023.

Summary: This project looks at the intersection between databases and compilers, and it investigates techniques to improve the performance of statically compiled database queries at runtime – based on the observed data. The target use case is the analytical queries in data warehouses and data lakes. While statically compiled database queries are optimised for performance, they do not benefit from runtime optimizations based on the observed data – and state, which are instead (somewhat) implemented in mainstream databases. Add to that, optimizations based on the ISA/microarchitecture where they execute are missing. Hence, statically compiled database queries can be further performance optimised by using runtime observed data and platform information.

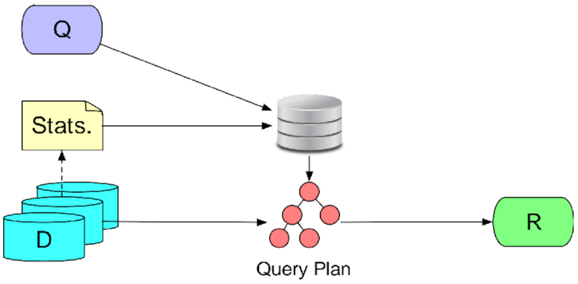
THE UNIVERSITY *of* EDINBURGH

DBShaker: Smart Recompilation for Adaptive Analytical Query Processing

Antonio Barbalace, Amir Shaikhha

School of Informatics, University of Edinburgh

Traditional Databases



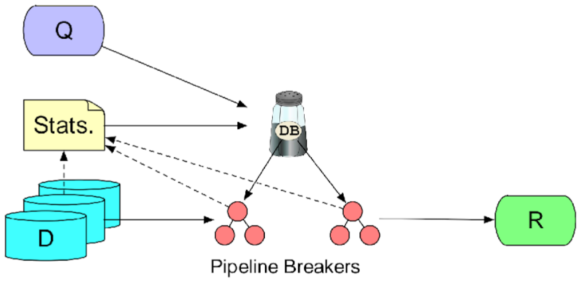
Query Planning

Query Compilation

Sensitive to Imprecise Statistics

Costly Recompilation

Proposed Engine: DBShaker



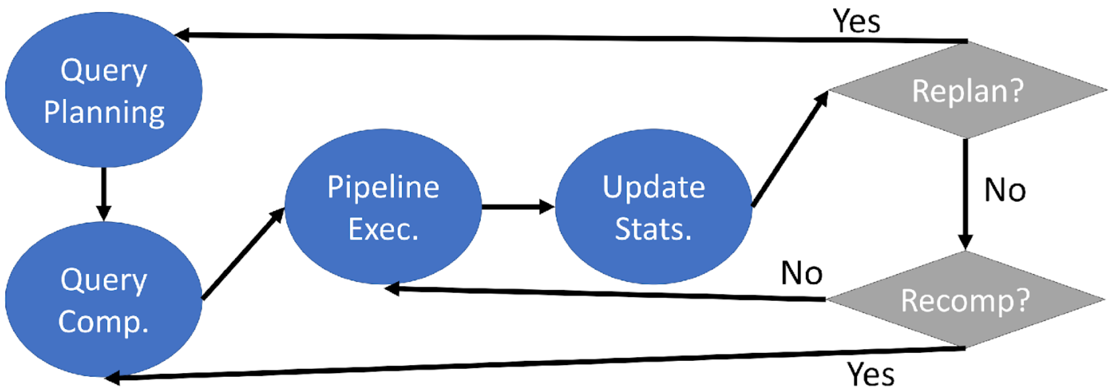
Adaptive Query Planning

Smart Query Compilation

Statistics Refreshed During Query Execcution

Fast Recompilation of Query Fragments

Workflow



Query Planning

Query Comp.

Pipeline Exec.

Update Stats.

Replan?

Yes

No

Recomp?

No

Yes

Joint Inference Multi-Task Networks

Note: This project finished in 2022. It was one of the first awards the PI received and provided a significant boost to his research

PI: Hakan Bilen

Bio: **Hakan Bilen** is a Reader (associate professor) in the School of Informatics in the University of Edinburgh. Hakan joined the Institute of Perception, Action and Behaviour (IPAB) at Edinburgh in 2017 and is leading several computer vision and machine learning projects in the institute. His core expertise is in computer vision and machine learning with a focus on weakly supervised, multi-task, data-efficient learning. He is co-author of more than 30 publications in the major international conferences and journals. He has been an area chair for ECCV 2018, BMVC 2018–19, CVPR 2021, ICCV 2021 and a regular reviewer for all the major computer vision and machine learning conferences and journals. He is an active member of the international computer vision community and organised workshops and tutorials at CVPR 2017–18, ECCV 2020. He received the best paper award in BMVC 2011 and outstanding reviewer award in ICCV 2017.

Summary: The project aims to learn machine learning models that can perform multiple tasks jointly. This results in computation and memory efficient models, which is especially important for platforms with limited resources, e.g. mobile devices. The main research challenge is to learn a single set of parameters that can do well in all of the tasks. These models are not only computation efficient but also data efficient, they can generalise to previously unseen tasks from few training samples well by transfer learning.

Selected publications:

- W.-H. Li, Xialei Liu, H. Bilen, (2022). “Cross-domain Few-shot Learning with Task-specific Adapters Proceedings of Conference on Computer Vision and Pattern Recognition” (CVPR). ◦
- W.-H. Li, Xialei Liu, H. Bilen, (2021). “Knowledge Distillation for Multi-task Learning. Proceedings of the International Conference on Computer Vision” (ICCV). ◦
- W.-H. Li, H. Bilen, (2020). “Knowledge Distillation for Multi-task Learning. Proceedings of the European Conference on Computer Vision” (ECCV) Workshop. ◦
- L. Deecke, I. Murray, & H. Bilen, (2019). “Mode Normalization. International Conference on Learning Representations” (ICLR).

Cross-domain Few-shot Learning with Task-specific Adapters

Wei-Hong Li, Xialei Liu, Hakan Bilen
VICO group, School of Informatics, University of Edinburgh, United Kingdom

Cross-domain Few-shot Learning

Cross-domain Few-shot learning (CDFSL):

- ⋆ Learning a classifier to for tasks from unseen domains with few labeled samples.

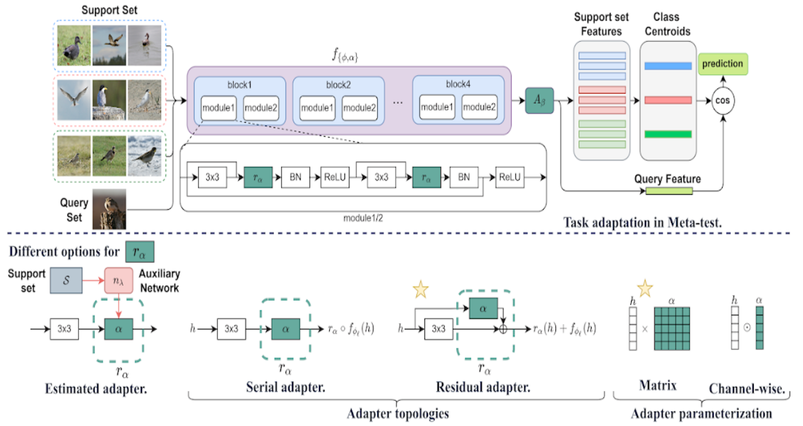
Challenges:

- ⋆ Large generalization gap between seen and unseen domains
- ⋆ Efficiently learn task-specific parameters from very few labeled samples

Contributions:

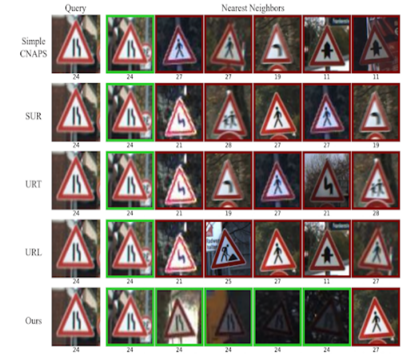
- ⋆ Systematic study of task-specific adapters in terms of their:
  - estimation,
  - connection topology,
  - parameterization.
- ⋆ We show that the optimal combination significantly boosts the performance (particularly in unseen domains) with negligible increase in computations.

Task adaptation with different adapter estimation, topology, and parameterization



Qualitative results on Meta-Dataset

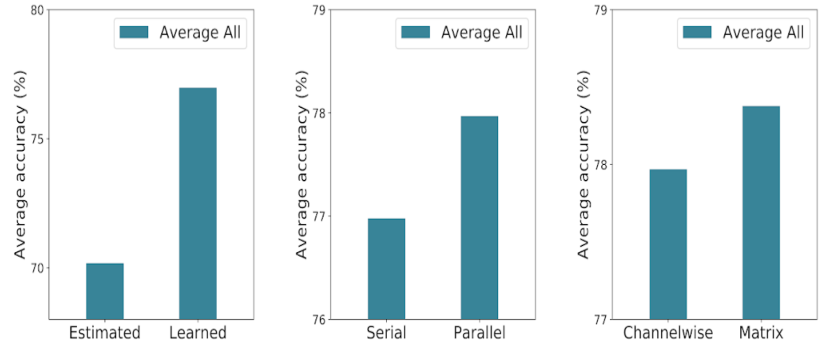
Qualitative analysis in testing only (unseen) dataset. Green and red colors indicate correct and false predictions respectively.



Traffic Sign (Unseen dataset)

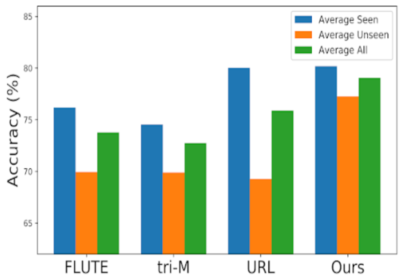
Ablation Study

- ⋆ Average classiciation accuracy on all datasets for different adapter designs.

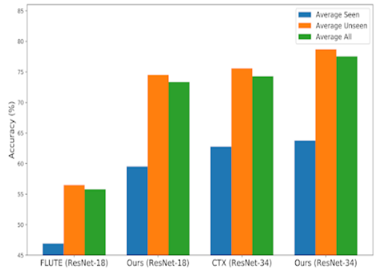


State-of-the-art comparison on augmented Meta-Dataset

- ⋆ Augmented Meta-Dataset (Triantafillou et al., 2020): 8 datasets for train/val/test and 5 datasets for test only.
- ⋆ Average classification accurayc of FLUTE (Triantafillou et al. 2021), tri-M (Liu et al., 2021), URL (Li et al., 2021), CTX (Doersch et al., 2020) and the best configuration found in this work (Ours).



Multi-domain Learning.



Single-domain Learning.

Probabilistic property-based testing

Note: This project finished in 2023

PI: **Vaishak Belle**

Co-PI: James Cheney

Bio: Vaishak Belle is Reader at the University of Edinburgh, an Alan Turing Fellow, and a Royal Society University Research Fellow. He has made a career out of doing research on the science and technology of AI. He has published close to 100 peer-reviewed articles, won best paper awards, and consulted with banks on explainability. As PI and CoI, he has secured a grant income of close to 8 million pounds. Dr Vaishak Belle is a Chancellor's Fellow and Reader at the School of Informatics, University of Edinburgh. He is an Alan Turing Institute Faculty Fellow, a Royal Society University Research Fellow, and a member of the RSE (Royal Society of Edinburgh) Young Academy of Scotland.

James Cheney is Professor of Programming Languages and Systems in the Laboratory for Foundations of Computer Science at the University of Edinburgh. He held a Royal Society University Research Fellowship 2008—2016, and an ERC Consolidator Grant 2016—2021. His research interests include databases and provenance, programming languages, logic and automated theorem proving.

Summary: The original aim of the project was to explore using insights from probabilistic programming as a basis for improving property-based testing, a popular technique for automated testing of computer systems. In general, property-based testing tests a declarative specification of intended program behaviour by randomly generating inputs and testing whether the expected property holds for each one. This can be viewed as a simple form of probabilistic programming in which the naïve method of rejection sampling is used. The original aim was to explore the question: what if we use more sophisticated sampling or inference methods from the probabilistic programming community for property-based testing?

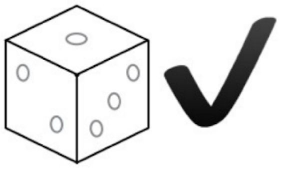
Selected publications

- Jesse Sigal, "Automatic Differentiation via Effects and Handlers: An Implementation in Frank", PEPM 2020 (short paper, non-proceedings)
- Jonathan Feldstein and Vaishak Belle, "Lifted Reasoning meets Weighted Model Integration", UAI 2021
- Jesse Sigal and Chris Heunen. "Duoidally enriched Freyd categories". RAMICS 2023
- Jonathan Feldstein, Dominic Philipps, and Efthymia Tsamoura. "Principled and Efficient Motif Finding for Structure Learning in Lifted Graphical Models". AAAI23.

Probabilistic property-based testing

Vaishak Belle, James Cheney, Jonathan Feldstein, Chris Heunen, and Jesse

THE UNIVERSITY of EDINBURGH

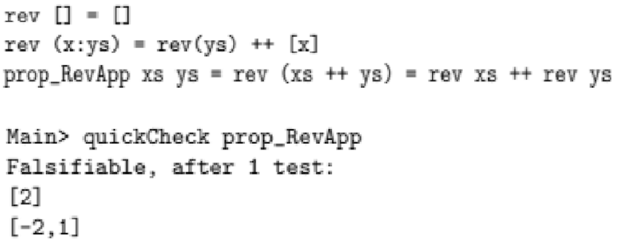


Ariane 5 first test flight on June 4, 1996

Failed due to incorrect data conversion error

The problem:
- Ensuring software quality is hard
- *Verification* takes massive human effort and requires a clear specification
- *Unit testing* expensive, cannot find all bugs
- *Model checking* is automatic, but limited
- *Property-based testing* (PBT) is a popular alternative: generate many (random) tests and check specification properties.

Property-based testing
- Specify desired behavior using logical *properties*
- Randomly generate test data to check properties
- Report counterexamples, sometimes using *shrinking* to make the counterexamples smaller / easier to understand
- First developed in Haskell and Erlang (QuickCheck), now in most major PLs



```
rev [] = []
rev (x:ys) = rev(ys) ++ [x]
prop_RevApp xs ys = rev (xs ++ ys) = rev xs ++ rev ys

Main> quickCheck prop_RevApp
Falsifiable, after 1 test:
[2]
[-2,1]
```

This shows the program and specification are inconsistent; actually in this case, the *specification* is wrong!

Challenges for PBT:
- Lack of programmer expertise: hard to program properties
- Random data generation struggles when bug behavior is rare
- Programmer effort needed to write generators and shrinking functions
- Errors can be hard to understand

- **Our hypothesis:** *property-based testing can be viewed as a form of general probabilistic programming*
- **Research question #1:** *Can we exploit relational and algebraic structure to speed up inference in probabilistic programs?*
- **Research question #2:** *Can we develop formal semantics and programming techniques for differentiable programming?*

Lifted Reasoning Meets Weighted Model Integration (Feldstein, Belle [2])
- Weighted model integration (WMI) supports efficient inference over *mixed discrete* and*continuous domains*
- Can we reason with *relational models* by avoiding grounding ?
- Extend *knowledge compilation* techniques to support efficient inference
- Scalability: inference on up to 100k data instances in <1s

Combining differentiation and probabilities (Heunen & Sigal [3])
- An algebraic framework has been developed to handle computational effects indexed by other computational effects, in the form of duoidally enriched Freyd categories

Differentiable programming using algebraic effects/handlers (Sigal & Heunen)
- A number of implementations of differentiable programming techniques have been implemented
- Asymptotically efficient and hopefully easier to prove correct (work in progress)

References

[1] K. Claessen and J. Hughes. QuickCheck: a lightweight tool for random testing of Haskell programs. In ICFP, 2000.
[2] J. Feldstein and V. Belle, Lifted Reasoning Meets Weighted Model Integration, UAI 2021.
[3] C. Heunen, J. Sigal, "Duoidally enriched Freyd categories", to appear, RAMICS 2023
[4] S. Speichert and V. Belle. Learning probabilistic logic programs in continuous domains. CoRR, abs/1807.05527, 2018.

The Tegola Rural Broadband Project

Note: Although both the PIs receive funding from the Joint Lab, this project has not been funded through the Joint Lab. However, it has been visited a few times by groups from Huawei. It is included here not only for the technology but also for the scenery, which may be one of the attractions of the project.

PIs: Peter Buneman and Mahesh Marina

Bios:

Peter Buneman is Professor of Database Systems at the University of Edinburgh. He has spent much of his time trying to improve the interface between programming languages and databases and to reconciling the principles of the two subjects. He has also made contributions to the principles of phylogeny, to data integration, to semi-structured and graph databases and to data provenance.

Mahesh Marina is a Professor in the School of Informatics at the University of Edinburgh and a Turing Fellow at the Alan Turing Institute in London. Before joining Edinburgh, he had a two-year postdoctoral stint at the UCLA Computer Science Department. He earned his PhD in Computer Science in 2004 from the State University of New York at Stony Brook. He has previously held visiting researcher positions at ETH Zurich and Ofcom London. He is a Distinguished Member of the ACM and a Senior Member of the IEEE.

Tegola: *Fast Broadband for Remote Rural Communities*

Mino Bernardi
Amazon Web Services

Peter Buneman
University of Edinburgh

Finlay MacKenzie
Administrate Ltd

Mahesh Marina
University of Edinburgh

William Waites
University of Strathclyde

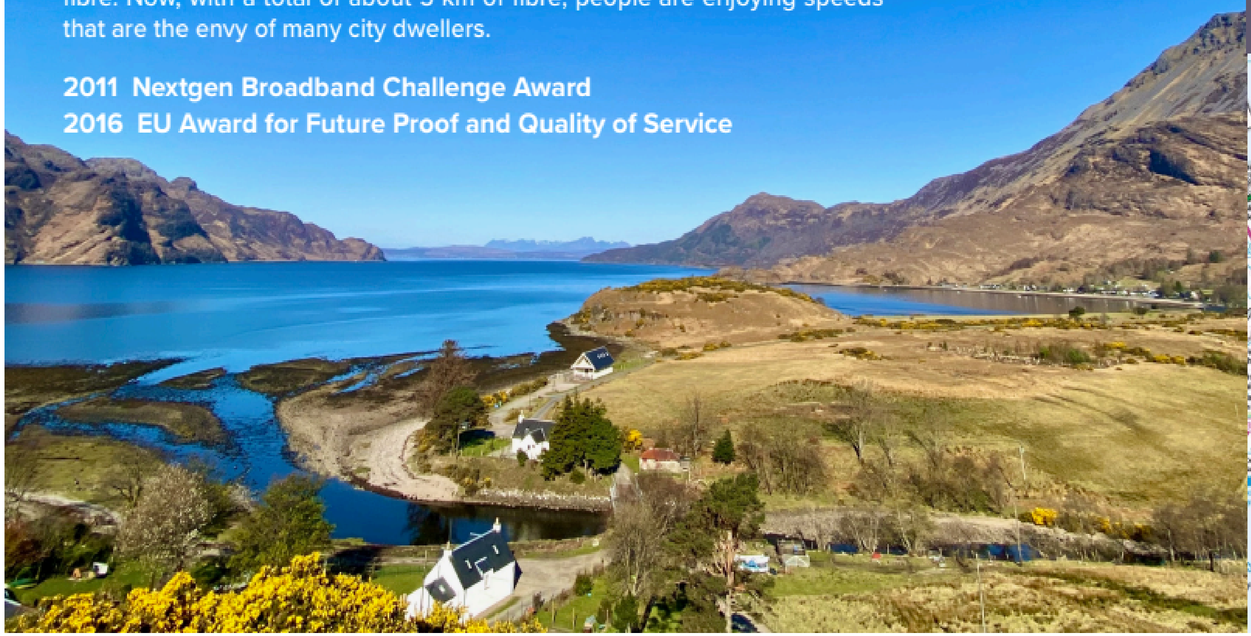
HISTORY: In 2008, stimulated by research at the University of Edinburgh's School of Informatics and with support from Sabhal Mòr Ostaig the community started to build its own network. It was baseon the same wifi technology that we use in our homes and had to be built on the cheap. By 2010 the network was up and running and was being copied by other communities across Scotland. It also led to the formation of HUBS, an organisation dedicated to providing internet services to communities, which now reaches from Applecross to the Scottish Borders.

Over the years the network continued to improve, but wireless transmission to houses has its problems. It has limited capacity; trees get in the way; and the equipment does not do well in the harsh weather of the West Coast of Scotland. Slowly, over the past four years the community has dug in local fibre. Now, with a total of about 5 km of fibre, people are enjoying speeds that are the envy of many city dwellers.

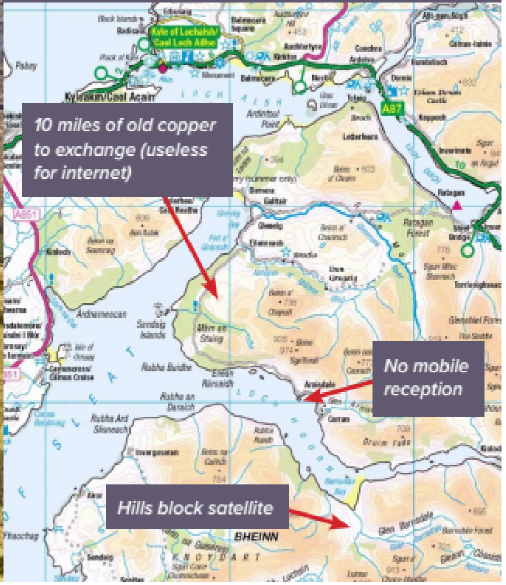
2011 Nextgen Broadband Challenge Award
2016 EU Award for Future Proof and Quality of Service

Tegola has served as a testbed for:

- Network organization
  (robust, dependable and simple)
- Use of wireless spectrum
  (clever use of limited frequencies, AI to predict tidal fading)
- West Highland engineering



Corran's new comms cabinet – keeps us dry while splicing fibre!


10 miles of old copper to exchange (useless for internet)
No mobile reception
Hills block satellite
Kyleakin/Caol Acain
A87
A851
Arnisdale
Corran
BHEINN