

Research Interests

My research interest spans programming languages, security, and systems (intersecting also with software engineering), focusing on developing novel theories, techniques, and tools for improving the performance, usability, reliability, and security of real-world computer software. In particular, I am now working on static program analysis and compiler optimization problems, including **pointer analysis** and **data-flow analysis**, for large-scale real-world software applications (with millions of lines of code). I also have broad interests in type systems and formal verification.

Education

- April 2022 **Ph.D., Computer Science and Engineering**,
University of New South Wales (UNSW), Sydney, Australia.
Advisor: Jingling Xue
- July 2018 **M.Sc.Eng., Computer Science and Technology**,
Institute of Computing Technology, CAS (ICT, CAS), and
University of Chinese Academy of Sciences (UCAS), Beijing, China.
Advisor: Lian Li
- July 2014 **B.Sc., Astronomy**,
Beijing Normal University (BNU), Beijing, China.
Advisor: Heng Yu
- July 2013 **B.Sc., Educational Technology**,
Beijing Normal University (BNU), Beijing, China.
Advisors: Qing Chen and ShengQuan Yu

Research Experience

- March 2022 **Postdoctoral Research Fellow**,
– present *University of New South Wales (UNSW)*, Sydney, Australia.
Advisor: Jingling Xue
- July 2018 – **Research Associate**,
Feb 2019 *Institute of Computing Technology, CAS (ICT, CAS)*, Beijing, China.
Advisor: Lian Li

Mentoring Experience

- March 2022 **Yujiang Gui**, a *Ph.D. student working on Android information flow analysis*,
– present Co-supervised with Prof. Jingling Xue.
- Sep 2017 – **Hengjie Zheng**, an *undergraduate thesis student from Wuhan University, working on Android API*
May 2018 *evolution*.

(Selected) Honors and Awards

- 2019 **ACM SIGSOFT Distinguished Paper Award (ASE 2019)**
- 2018 Excellent Thesis Award from ICT, CAS
Outstanding Student, the SKL of Computer Architecture, ICT, CAS
- 2014 Excellent Thesis Award from the Astronomy Department of BNU
A member of LiYun experimental class of BNU

Open-Source Projects

QILIN **A new framework for supporting fine-grained context-sensitivity in Java Pointer Analysis.**

QILIN enables different variables/objects in a method to be analysed under different context abstractions at the variable level, by supporting naturally the traditional method-level context-sensitivity as a special case. QILIN's toolbox includes already a rich set of pointer analyses for supporting (1) Andersen's context-insensitive analysis, (2) all common flavors of method-level context-sensitivity (e.g., k CFA, k OBJ), and (3) many flavors of fine-grained context-sensitivity, enabled by different pre-analyses such as EAGLE, and SELECTX. QILIN outperforms the state-of-the-art in both precision and efficiency.

Project: <https://github.com/QilinPTA/Qilin.git>

Docker image: <https://hub.docker.com/r/hdjay2013/qilin>

CONCH **A context debloating tool for accelerating all the object-sensitive pointer analysis.**

CONCH identifies context-independent objects by verifying three linearly verifiable conditions (which are almost always necessary for objects in the real-world applications to be context-sensitive) and then eliminates redundant contexts composed of these objects (which are usually not precision-beneficial). CONCH could speed up object-sensitive pointer analysis substantially and analyze more programs scalably at only a negligible loss of precision.

Project: <http://www.cse.unsw.edu.au/~corg/tools/conch/>

Docker image: <https://hub.docker.com/r/hdjay2013/conch-tosem-artifact>

TURNER **A sweet spot between accuracy and efficiency in existing pre-analyses for accelerating object-sensitive pointer analysis.**

TURNER is designed to enable object-sensitive pointer analysis to run significantly faster than the precision-preserving approach and achieve significantly better precision than existing non-precision-preserving approach. TURNER is the first intra-procedural pre-analysis technique in selecting precision-critical variables and objects.

Project: <http://www.cse.unsw.edu.au/~corg/turner/>

Docker image: <https://hub.docker.com/r/hdjay2013/turner>

ICTAPIFINDER **A systematic approach for automatically detecting incompatible API usages in Android Apps.**

We reduce the incompatible API usage detection problem to an inter-procedural context-sensitive data-flow analysis. The tool dramatically improves the precision of Android Lint, reducing its false positives by 82.1%.

Project: <https://github.com/DongjieHe/IctApiFinder.git>

Docker image: <https://hub.docker.com/r/hdjay2013/ictapifinder>

Publications

TOSEM'23 **IFDS-based Context Debloating for Object-Sensitive Pointer Analysis,**

Dongjie He, Jingbo Lu, and Jingling Xue,

ACM Transactions on Software Engineering and Methodology, TOSEM 2023 (CCF-A).

TSE'22 **Selecting Context-Sensitivity Modularly for Accelerating Object-Sensitive Pointer Analysis,**

Dongjie He, Jingbo Lu, Yaoqing Gao, and Jingling Xue,

IEEE Transactions on Software Engineering, TSE 2022 (CCF-A).

ECOOP'22 **Qilin: A New Framework for Supporting Fine-Grained Context-Sensitivity in Java Pointer Analysis,**

Dongjie He, Jingbo Lu, and Jingling Xue, The 36th European Conference on Object-Oriented Programming, ECOOP 2022 (CCF-B).

ASE'21 **Context Debloating for Object-Sensitive Pointer Analysis,**

Dongjie He, Jingbo Lu, and Jingling Xue, The 36th IEEE/ACM International Conference on Automated Software Engineering, ASE 2021 (CCF-A).

SAS'21 **Selective Context-Sensitivity for k-CFA with CFL-Reachability,**

Jingbo Lu, Dongjie He, and Jingling Xue,

The 28th Static Analysis Symposium, SAS 2021 (CCF-B).

ECOOP'21 **Accelerating Object-Sensitive Pointer Analysis by Exploiting Object Containment and Reachability,**

Dongjie He, Jingbo Lu, Yaoqing Gao, and Jingling Xue,

The 35th European Conference on Object-Oriented Programming, ECOOP 2021 (CCF-B).

TOSEM'21 **Eagle: CFL-Reachability-based Precision-Preserving Acceleration of Object-Sensitive Pointer Analysis,**

Jingbo Lu, Dongjie He, and Jingling Xue,

ACM Transactions on Software Engineering and Methodology, TOSEM 2021 (CCF-A).

ISSRE'20 **Exposing Android Event-Based Races by Selective Branch Instrumentation,**

Diyu Wu, Dongjie He, Shiping Chen, and Jingling Xue,

The 31st International Symposium on Software Reliability Engineering, ISSRE 2020 (CCF-B).

- ISSRE'20 **Correlating UI Contexts with Sensitive API Calls: Dynamic Semantic Extraction and Analysis**,
Jie Liu, **Dongjie He**, Diyu Wu, and Jingling Xue,
The 31st International Symposium on Software Reliability Engineering, ISSRE 2020 (CCF-B).
- ASE'19 **Performance-Boosting Sparsification of the IFDS Algorithm with Applications to Taint Analysis**,
Dongjie He, Haofeng Li, Lei Wang, Haining Meng, Hengjie Zheng, Jie Liu, Shuangwei Hu, Lian Li, and Jingling Xue,
34th IEEE/ACM International Conference on Automated Software Engineering, ASE 2019 (CCF-A).
ACM SIGSOFT Distinguished Paper Award
- ASE'18 **Understanding and Detecting Evolution-Induced Compatibility Issues in Android Apps**,
Dongjie He, Lian Li, Lei Wang, Hengjie Zheng, Guangwei Li, and Jingling Xue,
33rd IEEE/ACM International Conference on Automated Software Engineering, ASE 2018 (CCF-A).
- Draft **A CFL-Reachability Formulation of Callsite-Sensitive Pointer Analysis with Built-in On-the-Fly Call Graph Construction**,
Dongjie He[#], Jingbo Lu[#], and Jingling Xue,
submitted to TOPLAS 2023 (CCF-A), [#]: [equal contribution](#).
- Draft **Automatic Generation of Precision-Aware Stand-Alone Library Summaries for Whole-Program Pointer Analysis**,
Jingbo Lu, **Dongjie He**, Wei Li, Yaoqing Gao, and Jingling Xue,
submitted to ECOOP 2023 (CCF-B).
- Draft **Reducing the Memory Footprint of IFDS-based Data-Flow Analyses Using Fine-Grained Garbage Collection**,
Dongjie He, Yujiang Gui, Yaoqing Gao, and Jingling Xue,
submitted to ISSTA 2023 (CCF-A).

Professional Activities

Web Chair

- CGO'20 18th ACM/IEEE International Symposium on Code Generation and Optimization

Journal Reviewer

- SPE'22 Software: Practice and Experience 2022
JCST'22 Journal of Computer Science and Technology 2022

Program Committee

- LCTES'23 24th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems

Extended Review Committee

- ECOOP'22 36th European Conference on Object-Oriented Programming
OOPSLA'22 2022 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications

Sub-Reviewer

- OOPSLA'20, ICSE'21

Artifact Evaluation Committee

- OOPSLA'21 2021 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
CGO'22 20th ACM/IEEE International Symposium on Code Generation and Optimization
ECOOP'22 36th European Conference on Object-Oriented Programming
CC'22 31st ACM SIGPLAN International Conference on Compiler Construction
OSDI'22 16th USENIX Symposium on Operating Systems Design and Implementation
ATC'22 2022 USENIX Annual Technical Conference
OOPSLA'22 2022 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
ASE'22 37th IEEE/ACM International Conference on Automated Software Engineering
Programming The Art, Science, and Engineering of Programming (2023)