

# Dongjie He

## 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 currently working on 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, formal semantics, and program verification.

## Education

- April 2022 **Ph.D., Computer Science and Engineering**,  
*University of New South Wales (UNSW)*, Sydney, Australia.  
Advisor: Scientia Professor Jingling Xue, IEEE Fellow
- 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: Professor Lian Li
- July 2014 **B.Sc., Astronomy**,  
*Beijing Normal University (BNU)*, Beijing, China.  
Advisor: Assoc. Prof. Heng Yu
- July 2013 **B.Sc., Educational Technology**,  
*Beijing Normal University (BNU)*, Beijing, China.  
Advisors: Lecturer Qing Chen and Professor ShengQuan Yu

## Working Experience

- March 2022 **Postdoctoral Research Fellow**,  
– Present *University of New South Wales (UNSW)*, Sydney, Australia.  
Advisor: Scientia Professor Jingling Xue, IEEE Fellow
- July 2018 – **Research Associate**,  
Feb 2019 *Institute of Computing Technology, CAS (ICT, CAS)*, Beijing, China.  
Advisor: Professor Lian Li
- July 2013 – **IT teacher**,  
Sep 2014 *Hefei No. 10 High School*, Hefei, China.
- Sep 2012 – **IT teacher (intern)**,  
Feb 2013 *The Affiliated High School of Peking University*, Beijing, China.

## Supervisory Experience

- March 2022 **Yujiang Gui**, a Ph.D. student working on Android information flow analysis,  
– present Co-supervised with Scientia Professor Jingling Xue.
- March 2023 **Hai Yan**, Master student, UNSW,  
– present Working on AI Compiler.
- Sep 2017 – **Hengjie Zheng**, an undergraduate thesis student from Wuhan University,  
May 2018 working on Android API evolution.

---

## Honors and Awards

- 2023 **ACM SIGSOFT Distinguished Artifact Award (ISSTA 2023)**
- 2019 **ACM SIGSOFT Distinguished Paper Award (ASE 2019)**
- 2018 Excellent Thesis Award from ICT, CAS  
Outstanding Student, the SKL of Computer Architecture, ICT, CAS
- 2015-2018 Professional Scholarships
- 2014 Excellent Thesis Award from the Astronomy Department of BNU
- 2009-2013 Professional Scholarships

---

## 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 state-of-the-art pointer analysis algorithms for supporting (1) Andersen's context-insensitive analysis, (2) all common flavors of method-level context-sensitivity (e.g.,  $k$ CFA and  $k$ 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 real-world applications to be context-sensitive) and then eliminates redundant contexts composed of these objects (which are usually not precision-beneficial). CONCH can 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 the currently best non-precision-preserving approach. TURNER is the first intra-procedural pre-analysis 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>

### **Fpc** A fine-grained path edge collection technique for IFDS-Based Data-Flow Analyses

FPC uses a fine-grained, data-fact-level path edge collection technique to enable IFDS-based data-flow analyses to reduce their memory footprint by removing non-live path edges while avoiding re-processing GC'ed path edges.

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

Docker image: <https://doi.org/10.5281/zenodo.7965678>

### **IctApiFinder** A systematic approach for automatically detecting incompatible API usages in Android Apps

ICTAPIFINDER is designed to reduce the incompatible API usage detection problem into an inter-procedural context-sensitive data-flow analysis problem for the first time. The tool dramatically improves the precision of Android Lint, reducing its false positive rate by 82.1%.

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

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

---

## Publications

### **OOPSLA'23** A Container-Usage-Pattern-based Context Debloating Approach for Object-Sensitive Pointer Analysis,

*Dongjie He, Yujiang Gui, Wei Li, Yonggang Tao, Changwei Zou, Yulei Sui, and Jingling Xue,*  
Conditional Accepted by **OOPSLA'23 (CORE A\*)**.

### **ASE'23** Merge-Replay: Efficient IFDS-Based Taint Analysis by Consolidating Equivalent Value Flows,

*Yujiang Gui<sup>#</sup>, Dongjie He<sup>#</sup>, and Jingling Xue,*  
Accepted by **ASE'23 (CORE A\*)**, <sup>#</sup>: equal contribution.

- ASE'23 **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*,  
 Accepted by ASE'23 (CORE A\*).
- ISSTA'23 **Reducing the Memory Footprint of IFDS-based Data-Flow Analyses Using Fine-Grained Garbage Collection**,  
*Dongjie He, Yujiang Gui, Yaoqing Gao, and Jingling Xue*,  
 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA'23 (CORE A).  
 ACM SIGSOFT Distinguished Artifact Award
- 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'23 (CORE 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'22 (CORE 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'22 (CORE A).
- 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'21 (CORE 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'21 (CORE 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'21 (CORE A).
- 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'21 (CORE 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'20 (CORE A).
- 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'20 (CORE A).
- 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'19 (CORE A\*).\n
 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'18 (CORE A\*).

## Professional Activities

### Web Chair

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

### Program Committee

ASE'23 38th IEEE/ACM International Conference on Automated Software Engineering

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

## Journal Reviewer

SPE'22 Software: Practice and Experience 2022

JCST'22 Journal of Computer Science and Technology 2022

## 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)

## SRC Reviewer

PLDI'23 44th ACM SIGPLAN Conference on Programming Language Design and Implementation

## Skills

Languages English (fluent), Mandarin (native speaker)

Others Linux, LLVM, Soot, Wala, Docker, Git, Java, C/C++, Python, Coq, OCaml

## References

**Ana Milanova**, *A. Professor*,  
Department of Computer Science,  
Rensselaer Polytechnic Institute, USA.  
✉ milanova@cs.rpi.edu

**Jingling Xue**, *Scientia Professor, IEEE Fellow*,  
School of Computer Science and Engineering,  
University of New South Wales.  
✉ jingling@cse.unsw.edu.au

**Eric Boddern**, *Professor*,  
Heinz Nixdorf Institute,  
Paderborn University, Germany.  
✉ eric.boddern@upb.de

**Ondřej Lhoták**, *Professor*,  
David R. Cheriton School of Computer Science,  
University of Waterloo.  
✉ olhotak@uwaterloo.ca