

Preface

Eleonora Iotti¹, Greta Dolcetti², Vincenzo Arceri¹ and Sergio Maffeis³

¹University of Parma, Italy

²Ca' Foscari University of Venice, Italy

³Imperial College London, United Kingdom

The second edition of the Generative Code Intelligence Workshop (GeCoIn 2026) was held in conjunction with the 35th International Joint Conference on Artificial Intelligence (IJCAI-ECAI 2026), in Bremen, Germany, August 16, 2026. The workshop arose from the desire to bring together a research community that has, in recent years, witnessed rapid progress in the application of large language models to code generation. While these models are often capable of producing syntactically correct and, in many cases, executable code, important questions remain regarding the trustworthiness of generated artifacts, the verification of their semantic correctness, and their ability to support increasingly long, complex, and domain-specific tasks. It is within this shift in perspective, from basic generative capability toward reliability, verifiability, and control over the generative process, that the scientific contributions collected in this volume are situated.

The call for papers invited contributions on all aspects related to generative code intelligence, including but not limited to: static and dynamic analysis of AI-generated code; safety and security of AI-generated code; training, fine-tuning and RAG techniques for code generation and analysis; explainable methods for AI code generation; LLM applications in software engineering; and benchmarking of LLMs for code tasks. The paper submission deadline was set for May 21, 2026, with notification of acceptance on June 12, 2026, and camera-ready papers due on July 15, 2026.

The workshop received 19 submissions, which underwent a double-blind review process, with each paper receiving 2 reviews. Of these, 8 were accepted as full papers and are published in this volume, while 2 additional contributions were accepted as extended abstracts for oral presentation only.

The program opened with an invited talk by Giulio Zizzo (IBM Research), “From Python to Qiskit: Red Teaming AI at Scale,” devoted to the challenges of red teaming large language models at scale and to their extension to emerging quantum programming languages such as Qiskit and Qrisp: a theme that effectively opens the discussion on the security of AI-generated code.

The contributions accepted into the volume, together with the two extended abstracts presented orally, are organized around several recurring thematic threads. A first group of works addresses the security and reliability of automatically analyzed or generated code, proposing architectures that move beyond superficial lexical pattern recognition to capture the actual semantics of vulnerabilities. A second thread focuses on the rigorous evaluation of code generation, with benchmarks and frameworks that question the equivalence between executable code and correct code, examining aspects such as cross-lingual robustness, adherence to explicit specifications and contracts, and the ability to isolate individual stages of an automated software repair process. A third group of contributions turns instead to the agentic dimension of code generation, studying how harness design, rather than raw model capability, determines the success of agents engaged in long-horizon optimization and search tasks, extending even to scientific writing itself as a form of structured code generation. A fourth thread, finally, brings these themes into the specific context of high-performance computing and GPU acceleration, where hardware and infrastructural constraints demand tailored solutions for the translation, portability, and automated execution of scientific code: an area further explored by one of the two extended abstracts, devoted to code generation for scientific analysis in climate science environments; the other extended

GeCoIn 2026: Generative Code Intelligence Workshop, co-located with the 35th International Joint Conference on Artificial Intelligence (IJCAI-ECAI 2026), August 15–17, 2026 — Bremen, Germany

✉ eleonora.iotti@unipr.it (E. Iotti); greta.dolcetti@unive.it (G. Dolcetti); vincenzo.arceri@unipr.it (V. Arceri); sergio.maffeis@imperial.ac.uk (S. Maffeis)

🆔 0000-0001-7670-2226 (E. Iotti); 0000-0002-2983-9251 (G. Dolcetti); 0000-0002-5150-0393 (V. Arceri); 0000-0003-1514-6857 (S. Maffeis)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

abstract, in turn, offers a more theoretical reflection on extending code understanding beyond text, toward the visual and spatio-temporal dimension.

Taken together, the contributions of this edition outline three complementary axes of interpretation: the pursuit of reliable and verifiable code generation, the progressive shift of the research problem from the model itself to the agentic infrastructure surrounding it, and the need for domain-sensitive solutions where scientific or hardware constraints cannot be ignored.

We wish to thank the members of the program committee for their rigor and dedication throughout the review process, the authors for their contributions, and the organizers of IJCAI-ECAI 2026 for hosting the workshop.

Program Committee

- Lorenzo Cazzaro, University of Luxembourg, Luxembourg
- Lucas C. Cordeiro, University of Manchester, United Kingdom
- Abhishek Dharmaratnakar, Google, United States
- Adam Jones, Imperial College London, United Kingdom
- Penghui Li, Columbia University, United States
- Dalila Ressi, Ca' Foscari University of Venice, Italy
- Davide Taibi, University of Oulu, Finland
- Aditya V Thakur, University of California Davis, United States
- Giulio Zizzo, IBM Research Europe, Ireland

Organizing Committee

- Eleonora Iotti, University of Parma, Italy
- Greta Dolcetti, Ca' Foscari University of Venice, Italy
- Vincenzo Arceri, University of Parma, Italy
- Sergio Maffei, Imperial College London, United Kingdom

Workshop website: <https://gecoin-workshop.github.io/>