M2 Internship Position: Traceability-based Evaluation of LLM-Generated Tests

Context

Test generation is a crucial practice in software development. Recently, large language models (LLMs) have begun to replace manually written tests by generating test cases directly from software specifications and existing code [1]. However, no established methodology exists to evaluate whether LLM-generated tests faithfully reflect the provided specifications, or whether they are primarily influenced by the model's own parametric knowledge learned during training [2]. Current evaluation methodologies focus on compilation success, test pass rates, and code coverage [3]. While useful, these metrics do not assess whether a test is *relevant to the intended specification*, which becomes critical when the expected behaviour differs from examples seen during pre-training. Such bias may cause models to reproduce prior knowledge rather than implement the actual requirements.

Objective

The goal of this internship is to develop **tracing** techniques capable of identifying which parts of generated tests are driven by the model's prior (parametric) knowledge versus the contextual information provided in the specification. We propose exploring neural attribution–based explainability methods to expose and quantify this parametric knowledge [4].

Tasks

The intern will:

- Run open-source LLMs for code and test generation.
- Generate tests from a given suite of specifications, test cases, and source code.
- Evaluate generated tests using current metrics (e.g., pass@k, coverage).
- Implement neural attribution methods to differentiate parametric and contextual knowledge in generation.
- Analyse the role of parametric knowledge in test generation with respect to standard evaluation metrics.

The expected outcome is a methodology that (i) quantifies how well LLM-generated tests align with specifications, and (ii) measures the degree of influence from the model's prior knowledge.

Supervision and Working Environment

For the timely development of the internship, supervisors will provide access to open-source LLMs for code generation, including example Jupyter notebooks running in the GRICAD environment. The internship will run alongside two M1 TER internships on related topics, with opportunities for regular discussion and collaboration.

Supervisors: Gabriela Gonzalez (gabriela-nicole.gonzalez-saez@univ-grenoble-alpes.fr), Yves Ledru (yves.ledru@univ-grenoble-alpes.fr), and Nicolas Hili (nicolas.hili@univ-grenoble-alpes.fr).

Location: VASCO team, Laboratoire d'Informatique de Grenoble (LIG), Grenoble, France.
Start date: Early 2026 **Duration:** 6 months **Funding:** MIAI — EFELIA

How to Apply: Send your application to: **gabriela-nicole.gonzalez-saez@univ-grenoble-alpes.fr** Interviews will be conducted on a rolling basis until the position is filled.

References

[1] Wang, Y., Xia, C., Zhao, W., Du, J., Miao, C., Deng, Z., ... & Xing, C. (2025). ProjectTest: A Project-level LLM Unit Test Generation Benchmark and Impact of Error Fixing Mechanisms. arXiv preprint arXiv:2502.06556.
[2] Tighidet, Z., Mei, J., Piwowarski, B., & Gallinari, P. (2024, November). Probing language models on their knowledge source. In Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP (pp. 604-614).
[3] Mündler, N., Müller, M., He, J., & Vechev, M. (2024). SWT-bench: Testing and validating real-world bugfixes with code agents. Advances in Neural Information Processing Systems, 37, 81857-81887.
[4] Yu, H., Atanasova, P., & Augenstein, I. (2024). Revealing the parametric knowledge of language models: A unified framework for attribution methods. arXiv preprint arXiv:2404.18655.