Offer #2024-07936

PhD Position F/M LLM-based code generation for controlling artificial agents in simulated environments

**Contract type:** Fixed-term contract  
**Level of qualifications required:** Graduate degree or equivalent  
**Function:** PhD Position

**Context**

Program synthesis (Chaudhuri et al., 2021) has been traditionally considered for different programming tasks, but very little studied for synthesizing controllers, i.e., programs controlling artificial agents in simulated environments. Such controllers are usually studied under the Reinforcement Learning (RL) paradigm in AI (Sutton & Barto, 2018), where an agent learns an action policy from experience in order to maximize cumulative reward in a simulated environment. We believe that the recent rise of Large Language Models (LLMs) opens important perspectives and novel directions for controller synthesis. The successes of Copilot and ChatGPT show that LLMs can provide help and assistance in many programming tasks, both saving time and reducing errors through for instance bug finding and code suggestions (Chen et al., 2021). Moreover, LLMs are increasingly used in RL, in particular in LLM-Augmented RL, where LLMs can interpret users' intent and translate them into concrete rewards to be integrated in a RL algorithm (Fan et al., 2022). A few recent papers have proposed to use LLMs for generating AI controllers, reward functions and tasks (Faldor et al., 2024; Lehman et al., 2022; Wang et al., 2023) opening many exciting perspectives where the code generation abilities of LLMs is leveraged to synthesize CI agents and their environments.

In this project, we propose to explore novel research directions for using LLMs to synthesize controllers in unknown and complex environments. How to generate a controller program from a natural language description of the environment dynamics and task properties? Are LLM able to generalize to the generation of controllers that combine skills from previously discovered controllers? How to generate adaptive controllers, e.g. in the form of code specifying neural architectures using standard deep learning libraries such as Pytorch? Can LLM be used to generate the morphology of embodied agents? Can LLM help to disentangle semantic vs procedural knowledge in the context of controller synthesis? In particular, we believe that the question of combining skills is the key to scale these techniques to large environments and complex tasks.

**Assignment**

Several methodologies have been identified to address the above research questions. Existing approaches for adapting/evolving LLMs such as fine-tuning using Reinforcement Learning from Human Feedback (RLHF, Ouyang et al., 2022) was made famous with ChatGPT, but alternative promising approaches such as mutation operators (Lehman et al., 2022; Wang et al., 2023)and evolution of prompts (Fernando et al., 2023; Guo et al., 2024; Wan et al., 2024) have been proposed. This will enable us to explore complementary strategies (i) fixing the prompt and adapt the LLM or (ii) evolve the prompt with a fixed LLM; or using interactive vs. autonomous adaptation. We will also consider combining such approaches with non-LLM methods such as automatic program repair. Moreover, sorting knowledge in different components, e.g., semantic vs procedural knowledge, could help evolutionary methods to be more efficient and compositional, and decouple the evolution of such representations. Indeed, our brains represent “semantics” and “procedural” knowledge in different ways: knowing that mugs, lamps or books are inanimate objects (generally described as “facts”) are different in nature from procedural knowledge (e.g., know how to bike or swim, how to sort action primitives or instructions primitives) is crucial. Enabling LLMs to store these two kinds of knowledge will help to obtain more disentangled representations and compositionality, and offer more interpretable representations.

To initiate models, we will bootstrap them with supervised learning by for instance providing a first database of problems and solutions as prompts, or fine-tuning an existing LLM with it. The evaluation of found solutions will be performed through a Python interpreter or by various feedbacks (e.g., success or failure of code execution) with a special emphasis on the ability to generalize the acquired knowledge to novel environments and tasks that are compositions of the already solved ones. Datasets and environments such as the ones proposed in stable-baselines and evogym will be relevant baselines. We will also consider data augmentation, for instance by mutating existing problems and solutions and evaluating them.

**Main activities**
The project will be structured along several milestones of increasing complexity, although this proposed plan can be adapted according to the student's interests. In each of them, we plan to first bootstrap the model with a database of problem descriptions and known solutions to solve them, using it either to prompt the LLM (if the database is sufficiently small) or to fine-tune it with either supervised learning or RLHF (e.g. if the database is too large to fit in a prompt). Then we will evaluate how the LLM can generalize to new problems, in particular environments and tasks that are compositions of the already solved ones.

In the first stage, we will study the synthesis of reactive controllers on simple navigation tasks, for example in the form of Braitenberg vehicles (Braitenberg, 1984). The database will consist of a few examples of simple problems and solutions from existing Braitenberg vehicle implementations, which will be sufficiently short to input as a prompt.

In the second stage, we will study the synthesis of adaptive controllers, i.e. controllers that can learn for experience (e.g. in the form of learnable decision trees or neural networks). For this aim, we will bootstrap the model with a database of descriptions of RL environments and known RL algorithms to solve them, e.g. from the library and documentation of stable-baselines. Since this will be too large to input as a prompt, we will instead use supervised learning or RLHF to fine-tune the LLM on these examples.

Finally, in the third stage, we will study the synthesis of meta-learners, i.e. an adaptive controller that can perform well on a wide distribution of environments, as usually studied in the field of meta reinforcement learning (Duan et al., 2016). A possible direction here will be to explore how to train a Transformer model taking as input the sequence of agent’s observations, actions and rewards in a given environment in order to infer properties of the task at hand and outputting it in a natural language form. This natural language description will then be used as input to the LLM in order to generate the controller or to adapt an existing one.

In addition, in each of the three proposed stages, we will in parallel study the ability to synthesize agent's morphologies as JSON text files that can be interpreted in a 3D simulator such as Mujoco. This will allow the study of body-controller co-evolution, which is an important problem in both AI and Artificial Life (Bhatia et al., 2021). We believe that LLM-assisted program synthesis could lead to important breakthroughs in this domain.

Take this work plan mostly as a suggestion at this point. If you have your own ideas on alternative methodology or objectives we will be glad to discuss them.

References (most relevant are in bold)


**Skills**

- Excellent programming skills, preferably in Python. Experience with Pytorch or JAX.
- Prior experience with foundational models, deep reinforcement learning and data analysis.
- Strong interest in implementing artificial agents able to acquire an open-ended repertoire of skills.
- Prior experience with running large-scale experiments on CPU/GPU clusters is a plus.
- Fluent English

**Benefits package**

- Subsidized meals
- Partial reimbursement of public transport costs
- Possibility of teleworking and flexible organization of working hours
- Professional equipment available (videoconferencing, loan of computer equipment, etc.)
- Social, cultural and sports events and activities
- Access to vocational training
- Social security coverage

**Remuneration**

2100€ / month (before taxes) during the first 2 years,
2190€ / month (before taxes) during the third year.

**General Information**

- **Theme/Domain**: Robotics and Smart environments
- **Town/city**: Talence
- **Inria Center**: Centre Inria de l'université de Bordeaux
- **Starting date**: 2024-10-01
- **Duration of contract**: 3 years
- **Deadline to apply**: 2024-08-04

**Contacts**

- **Inria Team**: FLOWERS
- **PhD Supervisor**: Moulin-frier Clément / clement.moulin-frier@inria.fr

**About Inria**

Inria is the French national research institute dedicated to digital science and technology. It employs 2,600 people. Its 200 agile project teams, generally run jointly with academic partners, include more than 3,500 scientists and engineers working to meet the challenges of digital technology, often at the interface with other disciplines. The Institute also employs numerous talents in over forty different professions. 900 research support staff contribute to the preparation and development of scientific and entrepreneurial projects that have a worldwide impact.

**The keys to success**

Please don’t wait the application deadline to contact us. At first you can just send us your CV by email, explaining why you are particularly interested in this position.

Later on you’ll have to send a more formal cover letter, as well reference letters. Please also send documents or reports describing previous projects you have been working on (even if they are not directly related to the topic), as well as your Master grades and links to some code repositories.

For information, we will proceed in two phases:

- A first phase where we will make a selection of candidates based on the CV, the cover letter and the provided documents;
- A second phase of interviews of the selected candidates.
If you are selected for an interview, it will be the occasion to have a scientific discussion on the topics of the PhD. We highly recommend that you take the time to have a look at some of the mentioned references. We don't expect you to have read all of them, neither to have fully understood all the related concepts: during the interview you will be able to explain what you have understood and to ask questions about what was less clear.

**Warning:** you must enter your e-mail address in order to save your application to Inria. Applications must be submitted online on the Inria website. Processing of applications sent from other channels is not guaranteed.

### Instruction to apply

Thank you to send:
- CV
- Cover letter
- Master marks and ranking
- Support letter(s)

**Defence Security:**
This position is likely to be situated in a restricted area (ZRR), as defined in Decree No. 2011-1425 relating to the protection of national scientific and technical potential (PPST). Authorisation to enter an area is granted by the director of the unit, following a favourable Ministerial decision, as defined in the decree of 3 July 2012 relating to the PPST. An unfavourable Ministerial decision in respect of a position situated in a ZRR would result in the cancellation of the appointment.

**Recruitment Policy:**
As part of its diversity policy, all Inria positions are accessible to people with disabilities.