

# Problem Statement and Goals

## TPG

Team 3, Tangle  
Calvyn Siong  
Cyruss Allen Amante  
Edward Gao  
Richard Li  
Mark Angelo Cruz

Table 1: Revision History

Date	Developer(s)	Change
9/21/2024	Calvyn Siong	Added changes to stakeholder, environment and goals section
9/22/2024	Richard Li	Added Problem Statement section
9/23/2024	Richard Li	Made suggestions to Calvyn's sections
9/23/2024	Calvyn Siong	Implemented feedback
9/23/2024	Richard Li, Calvyn Siong	Added reflection section
11/3/2024	Calvyn Siong	Changed goals to reflect feedback appropriately (Introduced more measurable outcomes rather than desired outcomes/methodology)
3/31/2025	Richard Li	Fixed challenge level to change to Advanced level
3/31/2025	Richard Li	Constructed Problem Statement around a thesis

## 1 Problem Statement

### 1.1 Problem

Tangled Program Graphs (TPG) is a reinforcement learning framework under development at McMaster's Creative Algorithm Lab led by Dr. Stephen Kelly. The research aims to apply genetic programming principles to embedded systems and share the framework as an open-source library. Unlike conventional

deep learning approaches whose heavy training costs limit their accessibility, TPG uses evolutionary methods to reduce these costs. However, the current implementation of TPG faces two significant challenges:

1. **Software Engineering Practices:** The TPG codebase does not fully adhere to the robust software engineering standards seen in mature open-source projects. It lacks comprehensive documentation, robust testing frameworks, and automated CI/CD pipelines. This shortfall creates technical debt and hinders community contributions as researchers struggle with onboarding and maintaining the framework.
2. **Integration into Advanced Environments:** Up to this point, TPG has only been integrated and validated in fully observable, simple environments (e.g., Atari, Pac-Man, CartPole). However, real-world applications involve partially observable, dynamic scenarios. To extend the research and move towards deploying TPG in real-life embedded and robotics systems, integration with more complex simulators, like the MuJoCo physics engine, is essential.

The overarching thesis of this work is that by addressing these challenges, the TPG framework can be transformed into a scalable, efficient solution for embedded systems and real-time robotics. This transformation will not only demonstrate the practical viability of evolutionary reinforcement learning but also pave the way for widespread adoption of cost-effective intelligent systems.

## 1.2 Inputs and Outputs

The problem is subdivided into two primary areas:

### 1.2.1 Adhering to Software Engineering Practices

The central input in this area is the **TPG codebase** itself, along with a set of supplementary artifacts such as documentation, test scripts, and CI/CD configurations. The task involves:

- Integrating DevOps pipelines to run after code changes.
- Refactoring the code to adhere to best practices (e.g., SOLID principles, proper abstractions).
- Implementing automated unit testing and continuous integration.
- Drawing standards from established large-scale open-source projects such as the MuJoCo library.

The expected output is a robust and maintainable software environment where every code change is automatically validated. This change will improve the developer experience, foster community contributions, and accelerate further research developments.

### 1.2.2 MuJoCo Integration

In this area, the problem focuses on interfacing the TPG framework with the MuJoCo physics engine—a tool developed by DeepMind for real-life simulation experiments. For the integration to be successful, the interface must provide MuJoCo with detailed simulation configurations derived from TPG. This will enable:

- Accurate real-world testing of the TPG framework.
- Evaluation of TPG performance under dynamic and partially observable conditions.
- Iterative refinement of the framework based on performance data from complex simulation experiments.

The output of this effort is a fully functional simulation interface that allows TPG to be evaluated in environments that closely mirror real-world conditions.

### 1.3 Why is this Problem Important?

Current deep learning models for reinforcement learning are highly resource-intensive, limiting their use, especially within embedded systems and robotics. By applying evolutionary principles, TPG seeks to overcome the cost barriers associated with traditional deep learning. Achieving this requires two main advancements:

1. **Improved Software Engineering:** A robust codebase will lower the barrier for research contributions and enhance the framework’s reliability, fostering faster innovation and practical implementations.
2. **Realistic Simulation Validation:** Testing TPG in dynamic, partially observable environments is essential for transitioning from controlled mini-game simulations to real-world applications. This step is crucial to fine-tuning the framework and proving its efficacy in adaptable, cost-effective robotics.

Ultimately, addressing these challenges positions TPG as a viable alternative to traditional resource-heavy neural network approaches. This work is not only critical for advancing the framework but also has the potential to revolutionize real-world robotics by making intelligent systems more accessible and economically viable.

### 1.4 Stakeholders

#### 1. Primary Stakeholders: Dr. Stephen Kelly’s TPG Research Team

Dr. Kelly and his research team are the primary stakeholders of this project. As the original developers of the TPG framework, they are directly invested

in and benefit from improving the framework’s software engineering practices and expanding its applicability to more complex environments. The success of this project will directly contribute to their ongoing research efforts in reinforcement learning (RL) and genetic programming. The additional integrations created for TPG will allow the group to more efficiently explore new research avenues, including the application of TPG in robotics and advanced simulations.

## 2. Secondary Stakeholders: The GEGELATI Team

The GEGELATI team, a French-based TPG research organization, is a secondary stakeholder. They are adopting the TPG algorithm for research purposes that differ from Dr. Kelly’s project. Dr. Kelly has stated that this group references his TPG project for their own research purposes. While they are not directly involved in the development of this project, they stand to benefit from its outcomes. Improvements in the TPG framework, particularly in terms of software reliability and scalability, will provide valuable insights and tools that can be incorporated into their research. The establishment of an interface between TPG and modern robotics simulators may also open up new possibilities for their applications of the TPG algorithm.

## 3. Tertiary Stakeholders: The Wider Reinforcement Learning (RL) Community

The broader reinforcement learning community is a tertiary stakeholder in this project. Researchers and developers interested in the intersection of evolutionary algorithms and machine learning may benefit from the enhanced TPG framework. The improvements made through this project, including standard software engineering practices and expanded testing environments, will increase TPG’s accessibility and usability. Additionally, the publication of project results in international conferences could raise awareness of TPG’s capabilities, potentially encouraging wider adoption and collaboration within the RL community.

## 1.5 Environment

The capstone project will primarily be based on the existing TPG framework environment, which will consist of the existing C++ codebase which will be refactored and modified to comply with standard software engineering principles, such as continuous integration, automated testing, and improved documentation. In addition, a key objective of the capstone is to extend the TPG framework beyond its current application in Atari video games by integrating it into **MuJoCo (Multi-Joint dynamics with Contact)**, a widely used physics engine developed by **Google DeepMind**. MuJoCo provides highly accurate physical simulations, particularly suited for robotics. By interfacing TPG with MuJoCo, the project aims to evaluate the performance of the framework in controlling complex robotic systems, such as quadrupeds. This integration

will allow for more realistic testing environments, expanding the boundaries of TPG’s current capabilities.

## 2 Goals

1. **Improve the maintainability and scalability of the TPG framework by developing better software engineering practises.**

**Measurable Outcomes:**

- Implement a CI/CD pipeline to automate testing and validation of code changes.
- Develop unit test coverage for core components of the TPG framework.
- Provide comprehensive documentation for contributors, including setup instructions and contribution guidelines.

**Reasoning:** Ensuring the TPG framework adheres to software engineering best practices will reduce technical debt, streamline collaboration, and enable long-term scalability. This will be critical for improving productivity when working on the TPG framework.

2. **The TPG framework must have an Agent-Environment Interface between itself and MuJoCo**

**Measurable Outcomes:**

- Develop an interface between TPG and the MuJoCo physics engine to support agent-environment interactions.
- Successfully deploy TPG agents in at least three distinct MuJoCo environments.

**Reasoning:** This interface will enable the TPG framework to be tested in more realistic and complex environments. Successful integration with MuJoCo will validate TPG’s performance in a cutting-edge robotics simulator, essential for showcasing its real-world capabilities. Furthermore, the current testing environment for TPG is stationary (all states are known/given), and integrating with MuJoCo enables for testing with dynamic environments, where the agents will have to change over time.

3. **The TPG Integration in MuJoCo should have determinable behavior when ran**

**Measurable Outcomes:**

- Provide the capacity to generate a series of experiments to evaluate the behavior and performance of agents controlled by the TPG algorithm within any specified MuJoCo environment.
- Quantify task success rates and fitness levels for TPG agents in MuJoCo environments.

- Compare efficiency of TPG experiments in dynamic environments to static environments.

**Reasoning:** Extending upon the previous goal, these experiments will provide insights into how well TPG can control complex agents in a physics-based simulator like MuJoCo. Understanding the agent’s behavior in a dynamic environment will be essential for evaluating the practical applicability of TPG, similar to the previous goal. Demonstrating TPG’s effectiveness in dynamic environments will also validate its potential for real-world applications and future research/development decisions.

### 3 Stretch Goals

1. **The TPG framework is integrated with additional Robotics Simulators**

Extend the agent-environment interface to support additional simulators

**Reasoning:** Supporting multiple simulators will increase the flexibility of TPG, allowing it to be tested in diverse environments. This would expand the framework’s use cases, enhancing its appeal to the wider RL and robotics communities.

2. **The TPG framework is linked to a User-Friendly GUI for Experiment Management**

Create a graphical user interface (GUI) for researchers to manage, run, and visualize experiments within the TPG framework.

**Reasoning:** A user-friendly GUI would make the TPG framework more accessible and easier to use for the stakeholders involved. This could result in more efficient usage of the framework.

3. **Enable parallel execution of TPG agents across multiple MuJoCo environments.**

Successfully run TPG agents with  $\geq 2$  distinct MuJoCo environments simultaneously.

**Reasoning:** Supporting concurrent experimentation in multiple environments would test TPG’s scalability and resource efficiency. The ability to multi task reinforcement learning would validate TPG’s readiness for more complex systems.

### 4 Challenge Level and Extras

This project is classified as **advanced** due to the inclusion of a research experiment that investigates the effects of Dynamic Memory on the performance of the Tangled Program Graphs (TPG) framework within MuJoCo environments.

While the TPG framework has a proven foundation in simpler contexts, extending it to handle dynamic memory management in sophisticated robotics simulations adds both novel research dimensions and technical challenges.

The primary research component focuses on studying how Dynamic Memory influences the performance and stability of TPG in the context of MuJoCo. This involves designing experiments, collecting and analyzing performance data, and iterating on the integration approach based on empirical results. The research findings will be compiled into a detailed report, offering insights that bridge theoretical analysis and practical application.

In addition to the research experiment, the project continues to integrate essential software engineering practices. These include implementing continuous integration, developing a comprehensive testing suite, and maintaining detailed user documentation. Together, these components ensure that both the research and the engineering aspects meet high standards of reliability and reproducibility.

## Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

### 1. What went well while writing this deliverable?

When writing this deliverable, there were several things that went well.

One of the most notable aspects was the effective communication and teamwork within our group. We made an emphasis to split up tasks in a balanced way, which helped keep the workload manageable and clear. This clear division of labor allowed us to set realistic expectations, including communicating our individual estimated timelines (ETAs) for task completion in our group chats, and ensuring that everyone stayed on track.

Another key element was our commitment to reviewing each other's work. By frequently reviewing pull requests and providing constructive feedback, we were able to ensure the quality of our submission and catch any potential issues early. This also fostered a culture of openness to feedback, where we valued and incorporated each other's suggestions without hesitation, improving the consistency between our differing writing styles.

Dr. Kelly's involvement also played a pivotal role in our progress. From the beginning, he took time to explain reinforcement learning concepts to us in a way that deepened our understanding and directly contributed to our ability to contextualize Tangled Program Graphs (TPG) within his larger research goals. His willingness to meet regularly and support us synchronously and asynchronously—gave us the guidance we needed to overcome challenges and stay aligned with the direction of his research.

### 2. What pain points did you experience during this deliverable, and how did you resolve them?

One of the primary challenges we faced while working on this deliverable was defining clear and precise goals without having sufficient context. Initially, we found ourselves navigating in ambiguity, which made it difficult to establish a solid foundation for our project. This lack of context created uncertainty and hindered our ability to align our objectives effectively with the overarching research aims.



To tackle this issue, Calvyn took the initiative to draft a preliminary set of goals prior to our scheduled meeting with Dr. Kelly. This rough draft served as a starting point but also highlighted the gaps in our understanding, since we were only basing our work off the original project description in PotentialProjects.pdf. After our meeting, where Dr. Kelly provided valuable insights into reinforcement learning and his research on Tangled Program Graphs (TPG), it became evident that our initial goals did not fully capture the project's scope and intentions.

Recognizing this, Richard reviewed Calvyn's draft, and together they realized the difficulty of defining meaningful goals without adequate context. This realization prompted a collaborative effort to refine and redefine our objectives. Over the course of the week, Calvyn and Richard synchronized their efforts, incorporating Dr. Kelly's feedback and the newfound understanding of the scope of Dr. Kelly's goals for our capstone project.

Through open communication and iterative refinement, we were able to resolve the initial ambiguity. This process not only clarified our goals but also strengthened our teamwork and alignment. The experience underscored the importance of seeking clarity from supervisors and maintaining continuous dialogue within the team to ensure everyone is on the same page.

By proactively addressing the lack of context and leveraging our collective efforts, we turned a significant pain point into an opportunity for growth and improved collaboration.

A second painpoint for this capstone project has been the lack of time to fully engage with the foundational aspects of the project itself. While crafting the proposal and development plan, the team largely feels like there has been little opportunity to thoroughly familiarize ourselves with the C++ library, experiment with new techniques, or test out different approaches that may help us define more conclusive statements in the problem statement and goals document. This time constraint has been particularly frustrating because building a strong technical foundation is critical for the success of the project, yet the tight deadline for the proposal has left limited space to explore and solidify these elements.

Due to this, the team may have to rely on estimation for certain details in the proposal, especially for the goals of the project. While these estimates are based on existing knowledge and given context, it certainly introduces uncertainty into the project planning process. The team believes the immediate need to meet proposal deadlines balanced with the longer-term need to fully grasp and develop the technical aspects of the project has been a major pain point throughout this phase.

3. **How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?**

Adjusting the scope of our goals was a crucial step to ensure our Capstone project was both achievable and appropriately challenging. Initially, Dr. Kelly provided us with two primary objectives: code refactoring and interface development with MuJoCo. While these goals were clear, we recognized that their open-ended nature could lead to ambiguity if not scoped properly.

To address this, we decided to break down these overarching goals into smaller, more manageable subgoals. This decomposition allowed us to outline specific tasks, set realistic milestones, and better understand the steps required to achieve each objective. For example, under code refactoring, we identified the key components in a CI/CD pipeline, and for interface development, we listed the features that were essential for the project's success.

A significant factor in adjusting our scope was the need to fully comprehend the context of Dr. Kelly's research. Given that the scope of his work in machine learning and Tangled Program Graphs was outside our collective comfort zones, we initially found it challenging to gauge the complexity and time requirements of the tasks. None of us had prior experience in machine learning research, which made it difficult to estimate how hard each goal would be to accomplish.

To mitigate this uncertainty, we engaged in open discussions with Dr. Kelly to gain deeper insights into the project's demands. He reassured us that, with the right guidance, the challenges were very doable within our eight-month timeline. His confidence and willingness to provide support helped us feel more comfortable with the adjusted scope.