Self-Evolving Workflows

Nathaniel J. Houk
Independent Researcher
njhouk@gmail.com

May 2025

Contents

Abstract

This paper presents a formal computational theory for self-evolving AI workflows. We explore recursive AI task delegation, self-modifying AI execution, and the emergence of autonomous optimization. Our approach models AI workflows as dynamically evolving dependency graphs, allowing systems to reconfigure and optimize execution pathways in response to changes in objectives and available resources. This work has implications for AI-driven automation, meta-learning, and continuous self-improvement in AI systems.

1 Introduction

Modern AI systems rely on predefined workflows to execute tasks, from data preprocessing to model inference. However, these workflows are typically static, requiring human intervention for modifications and optimizations. This limitation prevents AI from autonomously improving its processes over time.

In this paper, we introduce the concept of *self-evolving AI workflow systems*, wherein AI agents dynamically reconfigure their execution sequences based on real-time constraints, performance feedback, and evolving objectives. Our framework treats workflows as graph-based structures that can be modified by AI meta-learning strategies, enabling self-improving computation.

1.1 Motivation

Self-evolving AI workflows are crucial for:

- **Autonomous Optimization**: AI systems should be capable of refining their own execution without external intervention.

- **Meta-Learning**: The system should analyze past performance and iteratively improve future workflow decisions.
- **Scalability**: As AI tasks become more complex, workflow reconfiguration should enable efficient resource utilization.

2 Related Work

Automated Machine Learning (AutoML) aims to optimize machine learning pipelines automatically [1]. However, AutoML focuses primarily on hyperparameter tuning and model selection rather than dynamic workflow evolution.

Reinforcement Learning (RL) for Workflow Optimization has been explored in environments where tasks can be sequenced for improved outcomes [2]. Our work differs by treating workflows as evolving structures that modify their own architecture rather than selecting from predefined options.

AI Planning and Task Delegation [3] provides methods for hierarchical task execution but lacks mechanisms for self-modification. Our model extends these concepts by enabling AI systems to recursively refine their own plans.

3 Theoretical Model

3.1 Workflow Representation

We represent an AI workflow as a directed acyclic graph (DAG), where:

- Nodes represent atomic computational tasks.
- Edges denote dependencies between tasks.
- The system maintains a *state vector* S_t capturing execution history and performance metrics at time t.

Given a workflow $W = (N, E)$, where N is a set of nodes (tasks) and E is a set of directed edges (dependencies), we define a workflow modification function:

$$W_{t+1} = f(W_t, S_t) \tag{1}$$

where f dynamically adjusts the structure of W_t based on historical performance stored in S_t.

3.2 Recursive Task Delegation

A key mechanism in self-evolving workflows is recursive task delegation, where an AI agent responsible for executing a task may:

1. Identify a more optimal sequence of subtasks.
2. Modify the workflow graph to reflect this new decomposition.
3. Store performance metrics for future adaptations.

Definition (Recursive Workflow Evolution): A workflow W is self-evolving if there exists an optimization function g such that:

$$W_{t+1} = g(W_t, S_t) \quad \text{with} \quad g : (W, S) \to W' \tag{2}$$

3.3 Optimization via Meta-Learning

Meta-learning enables the system to refine its workflow structure over multiple iterations. We employ a reinforcement learning approach where the workflow optimizer selects modifications based on a reward function:

$$R_t = \sum_i \alpha_i M_i(W_t) \tag{3}$$

where M_i are performance metrics (e.g., execution time, accuracy, computational cost) and α_i are weights reflecting their importance.

4 Implementation Considerations

4.1 Algorithm for Self-Evolving Workflows

Algorithm 1 Self-Evolving Workflow Algorithm

```
Initialize workflow W_0 with predefined tasks
for each execution cycle t do
    Compute state S_t based on past executions
    Identify possible workflow modifications
    Apply W_{t+1} = g(W_t, S_t) if improvement is detected
    Store S_t for future learning
end for
```

4.2 Practical Applications

Self-evolving AI workflows can be applied in:

- **AI-assisted research**: Automating hypothesis testing and experiment refinement.
- **Software development**: Continuous optimization of CI/CD pipelines.
- **Autonomous systems**: Adaptive robotic task planning.

5 Conclusion

This paper introduced the concept of self-evolving AI workflow systems, providing a formal model for recursive AI task delegation and workflow optimization. Future work will explore:

- Theoretical limits of self-modifying workflows.
- Integrating explainability into the workflow evolution process.
- Real-world case studies validating our approach.

Acknowledgments

The author gratefully acknowledges the contributions of collaborators, funding sources, and peer reviewers whose insights have improved this work.

References

[1] F. Hutter et al., *Automated Machine Learning*, 2019.

[2] Y. Li, "Reinforcement Learning for Workflow Optimization," in *Proceedings of NeurIPS*, 2020.

[3] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*, Elsevier, 2004.