Planning and Reasoning

Week 4: Enabling Agents to Think Before Acting

PhD Course in Agentic Artificial Intelligence

Learning Objectives

Bloom's Taxonomy Levels

- **Remember**: Define planning, reasoning, reflection in agent context
- **Understand**: Explain hierarchical task decomposition
- **Apply**: Implement Reflexion-style self-improvement
- **Analyze**: Compare planning strategies (LATS = tree search, Plan-and-Solve)
- **Evaluate**: Assess when planning helps vs. hinders performance
- **Create**: Design planning systems for complex tasks

Plann

transforms reactive agents into deliberative problem solvers.

Why Planning Matters

Reactive Agent Limitations

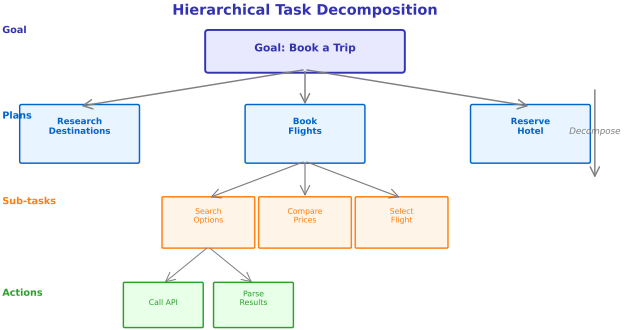
- Single-step responses miss complex dependencies
- No lookahead leads to suboptimal paths
- Hard to recover from early mistakes

Planning Enables

- Multi-step goal decomposition
- Anticipation of obstacles
- Backtracking and alternative paths
- Resource-aware execution

= thinking before acting. Essential for complex tasks.

Hierarchical Task Decomposition



Break

complex goals into manageable sub-tasks and executable actions.

Plan-and-Solve Prompting

Key Idea (Wang et al., 2023)

- Explicit planning step before execution
- "Devise a plan to solve the problem"
- Extract relevant variables first

Template Structure

1. Understand the problem
2. Extract relevant information
3. Devise step-by-step plan
4. Execute plan with verification

Plan-and-Solve improves over zero-shot CoT on math and reasoning.

LATS: Language Agent Tree Search

Key Idea (Zhou et al., 2024)

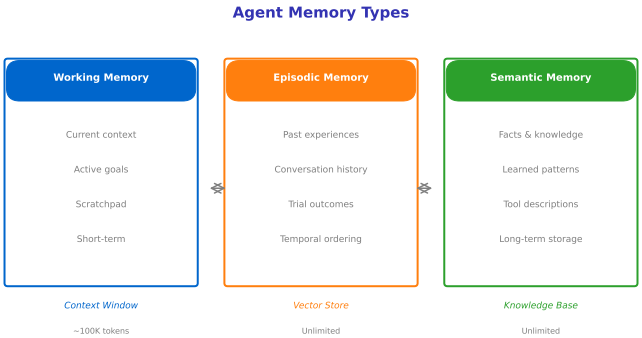
- Combines LLM reasoning with MCTS (Monte Carlo Tree Search)
- Explores multiple reasoning paths in parallel
- Uses value function to guide search

Components

- **Selection**: Choose promising nodes
- **Expansion**: Generate child actions
- **Simulation**: Evaluate outcomes
- **Backpropagation**: Update value estimates

LATS achieves state-of-the-art on HotpotQA (QA) and WebShop benchmarks.

Agent Memory Types



Mem

enables learning from past experiences and maintaining context.

Reflexion: Learning from Mistakes

Key Idea (Shinn et al., 2023)

- Verbal reinforcement learning
- Agent reflects on failures and generates insights
- Insights stored in memory for future attempts

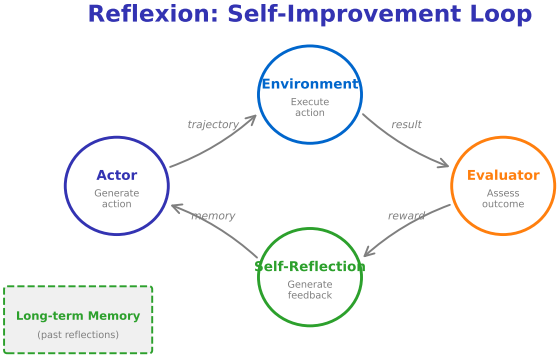
Reflexion Loop

1. Attempt task
2. Evaluate outcome
3. Generate verbal reflection
4. Store in episodic memory
5. Retry with reflection context

Refle

improves pass@1 (first-try success) on HumanEval code benchmark: 80% to 91%.

Reflexion Loop Architecture



Refle

enables learning through verbal self-reflection and memory.

Self-Refine Framework

Key Idea (Madaan et al., 2023)

- Iterative self-improvement without external feedback
- Generate, critique, refine cycle
- Single LLM plays multiple roles

Process

1. **Generate**: Initial output
2. **Feedback**: Self-critique the output
3. **Refine**: Improve based on critique
4. Repeat until satisfactory

Self-Refine works for code, math, summarization, and more.

Reasoning with World Models

RAP: Reasoning via Planning (Hao et al., 2023)

- LLM as both world model and reasoning agent
- Simulates future states before acting
- Uses MCTS for strategic exploration

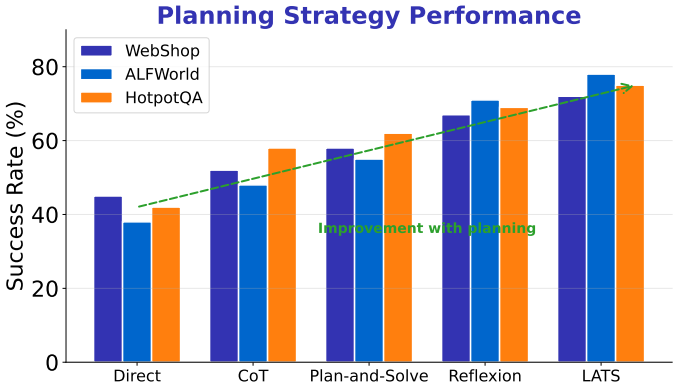
Key Components

- **World Model**: Predicts next state given action
- **Reward Model**: Evaluates state quality
- **Search Algorithm**: Explores action space

World

models enable lookahead reasoning and counterfactual analysis.

Planning Strategy Performance



More sophisticated planning consistently improves performance across benchmarks.

When Planning Helps (and Hurts)

Planning Beneficial

- Multi-step reasoning tasks
- Tasks with irreversible actions
- Resource-constrained execution
- Novel problem domains

Planning Overhead

- Simple, well-defined tasks
- Time-critical responses
- Highly dynamic environments
- When exploration cost exceeds benefit

Matc

planning depth to task complexity and time constraints.

Implementation Patterns

Plan-First Architecture

- Generate complete plan before execution
- Good for stable, predictable tasks
- Risk: Plan may be invalid by execution time

Interleaved Planning

- Plan a few steps, execute, replan
- Adapts to changing conditions
- Higher overhead but more robust

Hierarchical Execution

- High-level planner + low-level executor
- Separates strategic and tactical decisions

Choo

architecture based on task dynamics and failure costs.

Required Readings

This Week

- Shinn et al. (2023). "Reflexion." arXiv:2303.11366

Supplementary

- Wang et al. (2023). "Plan-and-Solve." arXiv:2305.04091
- Zhou et al. (2024). "LATS." arXiv:2310.04406
- Hao et al. (2023). "RAP." arXiv:2305.14992
- Madaan et al. (2023). "Self-Refine." arXiv:2303.17651

Refle

is required; others provide alternative planning approaches.

Summary and Key Takeaways

Key Concepts

- **Hierarchical Planning**: Decompose goals into sub-tasks
- **Reflexion**: Learn from verbal self-reflection
- **Tree Search**: Explore multiple reasoning paths (LATS)
- **Memory**: Working, episodic, semantic for context

Design Decisions

- Plan-first vs. interleaved planning
- Depth of planning vs. execution speed
- When to reflect and when to retry

Next Week: Multi-Agent Architectures

Plann

transforms reactive LLMs into deliberative agents.