2025.1.3

LLMs for RL

1. Utilizing LLMs to decompose complex tasks and generate high-level plans
2. Using LLMs to design the reward function
3. Directly training LLMs as the behavior polices with RL algorithms

- Plan-Seq-Learn: Language Model Guided RL for Solving Long Horizon Robotics Tasks
- Learning Reward for Robot Skills Using Large Language Models via Self-Alignment
- Language-guided Skill Learning with Temporal Variational Inference

- LLM-Empowered State Representation for Reinforcement Learning
- Code as Reward: Empowering Reinforcement Learning with VLMs
- KALM: Knowledgeable Agent by Offline Reinforcement Learning from Large Language Model Rollouts

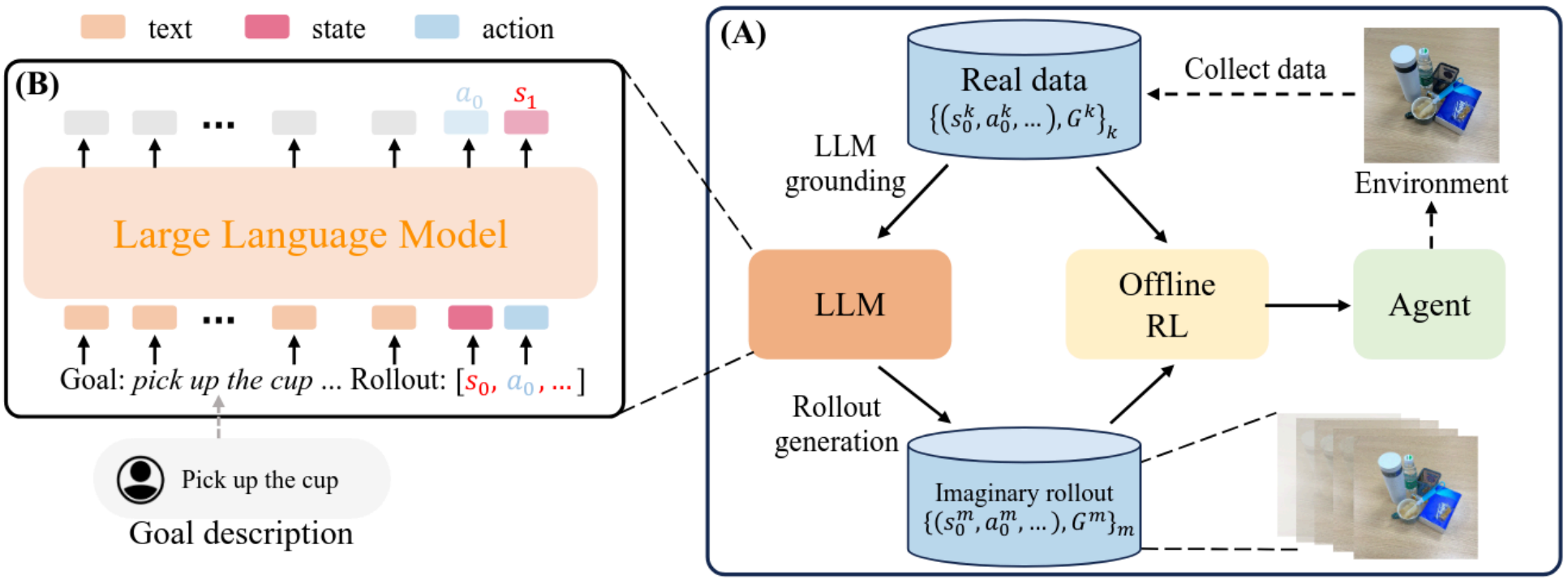

Figure 1: KALM mainly consists of three steps: (1) LLM grounding, (2) Rollout generation, and (3) offline RL.

Credit Assignment

Large Language Models (LLMs) are increasingly being applied to complex reasoning tasks, ranging from solving math problems to developing code. The most common method for training these models is Proximal Policy Optimization (PPO), which addresses the **credit assignment** problem using a value network. However, this value network can be significantly biased, which may impact performance.

- VinePPO: Unlocking RL Potential For LLM Reasoning Through Refined Credit Assignment

Misalignment issues hinder LLM-based agent's ability to complete decision-making tasks. Recent advances(GALM and TWOSOME) have showcased that misalignment can be alleviated through building the bridge between optimizing action and optimizing tokens: RL agents where sampling actions from a policy means sampling a sequence of tokens mapping to an action from a (suitably conditioned) large language model. Since actions are typically described as a sequence of more than one token, this introduces issues around **credit assignment** between tokens contributing to an action.

- Reinforcing LLM Agents via Policy Optimization with Action Decomposition

Self Correction

- Training Language Models to Self-Correct via Reinforcement Learning
- Retroformer: Retrospective Large Language Agents with Policy Gradient Optimization

RL for LLMs

- Interactive Dialogue Agents via Reinforcement Learning on Hindsight Regenerations
- AGILE: A Novel Reinforcement Learning Framework of LLM Agents
- Training Large Language Models for Reasoning through Reverse Curriculum Reinforcement Learning

Many of these problems require the agent to explicitly take the steps to gather information before making a decision. Single-turn RL for LLMs cannot learn such nuanced trategies as they attempt to solve the problem within a single step.

Multi-turn RL for LLMs can become sample inefficient in multi-step settings that require interaction with an external environment. Existing methods either consider a single token as an action or consider an utterance as a single action.

- ArCHer: Training Language Model Agents via Hierarchical Multi-Turn RL

LLM Reasoning via Planning

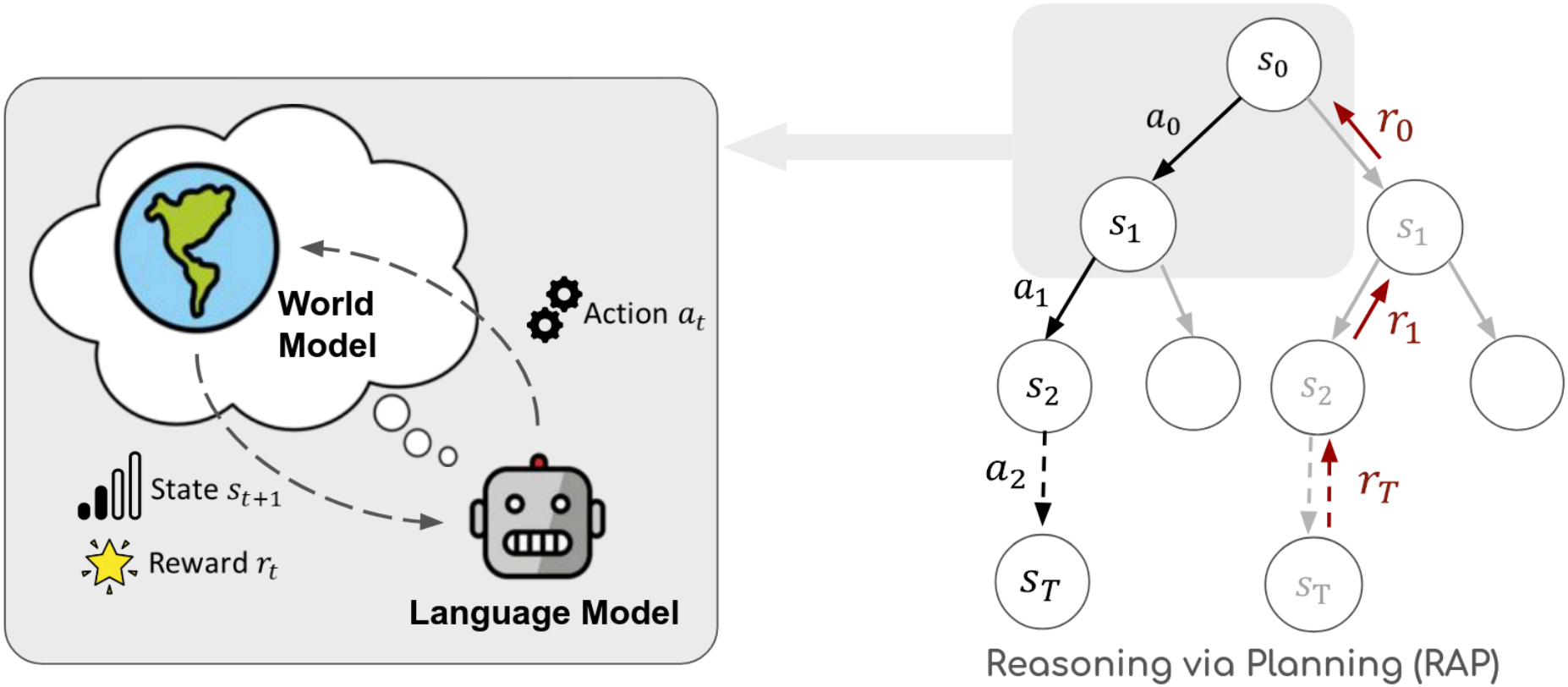

Figure 2: An LLM-based policy generates actions and an LLM-based World Model predicts next state under state-action pairs.

- Reasoning with Language Model is Planning with World Model
- Empowering LLM Agents with Zero-Shot Optimal Decision-Making through Q-learning
- Language Agent Tree Search Unifies Reasoning, Acting, and Planning in Language Models

Speculative Decoding

- Block Verification Accelerates Speculative Decoding
- SWIFT: On-the-Fly Self-Speculative Decoding for LLM Inference Acceleration

Caching

- Accelerating Diffusion Transformers with Token-wise Feature Caching
- FasterCache: Training-Free Video Diffusion Model Acceleration with High Quality

KV Cache

- VL-Cache: Sparsity and Modality-Aware KV Cache Compression for Vision-Language Model Inference Acceleration
- Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs

Sparse Attention

- CoreInfer: Accelerating Large Language Model Inference with Semantics-Inspired Adaptive Sparse Activation
- MInference 1.0: Accelerating Pre-filling for Long-Context LLMs via Dynamic Sparse Attention