Evolving Reinforcement Learning Algorithms

**Motivation**
- Desire general purpose RL algorithms without manual effort.
- Problem: Meta-learn RL algorithms that generalize.

**Overview**
- **Insight:** RL algorithm as a computational graph
- **Method:** Evolve population of graphs by mutating, training, and evaluating RL agents
- **Result:** Learn new algorithms which generalize to unseen environments

**Prior Work**
- Genetic Programming
  - Mostly applied to SL
- Meta-learning in RL
  - Adaptation: Finn & Levine 2018
  - Not domain agnostic
  - Learning RL Algorithms
    - Metagradients: Krish et al. 2020, Oh et al. 2020
- Mostly applied to SL

**RL Algorithm as a Computational Graph**
- Computational graph computes loss function for agent to optimize.

**Learned Algorithms**
- **DQNClipped** as constrained optimization
  \[ L_{DQN} = 0.1 \times \mathbb{E}[Q(s, a)] + \beta \]
- **DQNReg** as entropy regularization
  \[ L_{DQN} = (Q(s, a) - (r_t + \gamma \max_{a' \neq a} Q(s', a')))^2 \]

**Results**
- Outperform baselines on train envs.
- Generalize to unseen environments.
- Benefits on Atari even though training envs. were non-image based

**Future Work**
- Extensions to actor critic, offline RL, representation learning
- Analyze and incorporate learned algorithms into existing ones
- Machine assisted algorithm development