

I designed and implemented the core ScalingEnv environment in `gym_scaling/envs/scaling_env.py`, creating a realistic cloud autoscaling simulation with multiple load patterns (SINE_CURVE, RANDOM, STEADY, SPIKE, POISSON, SINUSOIDAL). This included defining the state space (load, queue size, instances, influx rate), action space (scale up/down/maintain), and reward function balancing cost optimization with performance maintenance.

Algorithm Implementation: I implemented three distinct reinforcement learning approaches from scratch: Q-Learning: Developed `train_qlearning_extended.py` with tabular RL, epsilon-greedy exploration, and state discretization for baseline comparison

Deep Q-Network (DQN): Created both `train_dqn_simple.py` and `train_dqn_m4_gpu.py` versions, implementing experience replay, target networks, and GPU acceleration using Metal

Performance Shaders (MPS) for M4 MacBook optimization

Proximal Policy Optimization (PPO): Built `train_ppo_m4_gpu.py` with policy gradient methods, clipped surrogate objectives, and advanced evaluation callbacks

Technical Optimization: I optimized the entire system for M4 MacBook Air hardware, implementing automatic GPU detection and MPS acceleration that achieved 3-4x training speedup. I also created comprehensive checkpointing systems, progress tracking with `tqdm`, and robust error handling throughout all training scripts.

Evaluation and Validation: I developed extensive evaluation frameworks including `verify_all_models.py` for model integrity testing, `demo_all_models.py` for visual policy comparison, and comprehensive Jupyter notebook analysis comparing all algorithms against threshold baselines. The evaluation includes performance metrics, cost analysis, and behavioral visualization.

Challenges Overcome

The most significant challenge was optimizing deep RL training for Apple Silicon hardware, requiring custom MPS integration and memory management. I also solved complex environment wrapper compatibility issues between legacy Gym and modern Gymnasium APIs, and implemented robust model serialization handling multiple RL frameworks.