MAV-EI – Budgeted Multi-Verifier Selection for Embodied Control

Abstract

Inspired by the success of multi-agent verification in improving LLM outputs, we propose a similar "society of verifiers" at test-time that evaluates and steers (optionally) action proposals from fast policy/plan generators for manipulation and navigation tasks. Unlike approaches that only scale the number of sampled candidates with a single VLM verifier, MAV-EI would scale with both the number of candidates and diversity/number of verifiers (language, geometry/physics, and affordances). Since low-latency is critical in embodied intelligence tasks, we would also implement a scheduler that chooses how many candidates to sample and which verifiers to consult to maximize success-per-millisecond.

Research Questions

1. At a fixed latency, do diverse verifiers beat repeated inquiries to a single VLM?
2. Does a budgeted cascade (of verifiers cheap → expensive) yield higher success-per-ms than fixed pipelines?
3. Does MAV-EI generalize across generators (SDP vs. Video-to-Action)?

Methodology

1. Bring up one fast proposal generator (week 1)
    a. Start with Streaming Diffusion Policy (SDP) for low-latency proposals and reproduce 1-2 tasks to sanity-check our tech stack
2. Implement a minimal Best-of-N baseline (week 1)
    a. Sample K candidates/actions per step from the generator and pick one at random. No verifiers
3. Build the verifier zoo (weeks 2-4)
    a. V1: Instruction-adherence (kind of like the use of VLMs here)
        i. Use a small open-source VLM like Qwen2.5-VL-7B and create 3-5 yes/no "aspects" to verify (e.g., goal completion, object identity, constraint obedience)
    b. V2: Geometry/physics feasibility
        i. If in simulation, check for minimum link-to-scene clearance, penetration margin, and table support test (for placements)
    c. V3: Reachability / Inverse Kinematics (IK) Feasibility
        i. Check IK solvability with joint-limit and manipulability penalties for each candidate waypoint
    d. V4: Affordance/value critic
        i. Train a small MLP or CNN on synthetic data to predict success or grasp quality from local crops and poses

e. Selection rule (BoN-MAV aggregator):
    i. For each candidate trajectory, each verifier outputs either a calibrated score in [0, 1] and/or binary approval
    ii. We select the candidate with the highest score or "most approvals"
4. Budgeted, anytime scheduler (week 5)
    a. Implement the scheduler given a per-step budget B to choose K candidates and a subset of verifiers using a cheap-to-expensive cascade (V3 + V2 → V4 → V1)
5. Evaluation (weeks 3-8)
    a. Suites: LIBERO, Meta-World, CALVIN, iTHOR, or Habitat. Start with 2-3 tasks each
    b. Baselines: Best-of-N (no verifiers), Best-of-N and single VLM verifier (RoboMonkey), and MAV-EI
    c. Reports: success, wall-clock per step, success-per-ms, etc.
6. Ablations (weeks 7-10)
    a. Remove each verifier, vary K and B, check diversity vs. count of verifiers, and check cross-generator generality (SDP vs. Video to Action)
7. Packaging (weeks 10-12)
    a. Release repo and write paper

Equipment and Resources

In a compute cluster or shared lab server:

- GPU: 1 RTX 4090 for running SDP and a local VLM for verification
- CPU/RAM: 16-24 cores and 64 GB of ram for our simulations, VLM, and logging
- Storage: 1-2 TB for our datasets, checkpoints, and videos

Software stack (open-source):

- Generators:
    - Streaming Diffusion Policy
    - Video to Action
- Simulation and physics: MuJoCo or PyBullet
- VLM verifiers: Qwen2.5-VL-7B or LLaVA-1.5/NeXT
- Benchmarks: LIBERO, Meta-World, CALVIN, iTHOR, or Habitat

Deliverables

1. Open-source verifier library with clean interfaces
2. Evaluation harness with budgeted scheduling and logging
3. Benchmark report with compute-performance curves and ablations
4. Minimal configs for quick reproduction of results