Constrained Multi-Objective Post-Training for Vision-Language-Action Models

Cody Hill[1,2] Zain Ulabedeen Farhat[2] Yue Wang[2]
[1]Mississippi State University [2]University of Central Florida
{co998728, zainulabedeen.farhat, yue.wang}@ucf.edu

Abstract

Reinforcement-learning post-training of vision-language-action (VLA) models optimizes for task success alone. A policy deployed in the real world is judged by richer qualities than that. While it ensures task completion, the policy should also minimize energy usage, move smoothly, and avoid unsafe contact. This report summarizes a ten-week REU project on post-training a pre-trained VLA under a hard task-success requirement while improving these secondary objectives, rather than trading success away against them as a conventional weighted-sum reward allows. We developed a constrained multi-objective post-training method for this setting and validated it on two multi-objective control benchmarks, where it satisfies the task requirement on 77–99% of evaluated policies against 8–19% for a conventional multi-objective baseline, and recovers between $1.6\times$ *and* $3.8\times$ *as much of the achievable objective space. We further verified that the method's internal constraint-enforcement mechanism behaves as intended, applying negligible pressure while the requirement is easy to satisfy and increasing sharply as it becomes difficult to meet. We then built a complete training and evaluation pipeline for a compact VLA, MiniVLA, on the LIBERO manipulation benchmark, obtained a competent supervised baseline across its four suites, and began post-training under the new method. An analysis of the candidate secondary objectives on LIBERO found limited genuine competition among them, a finding that is now directing our choice of evaluation environment. A complete technical account, including the method's formulation, will follow as an expanded report once this work is published.*

1. Introduction

A vision-language-action model maps camera images and a natural-language instruction to robot actions. Recent work has shown that reinforcement-learning post-training can improve the task competence of these models beyond what imitation learning alone achieves. However, essentially all of that post-training optimizes to one scalar task-success reward.

Policy deployment requires more than that. A successful manipulation policy that draws large currents, jerks between waypoints, or presses hard into the table is worse than one that succeeds gently, even though both score identically on success. For this reason, we would like to post-train for secondary objectives too (energy, smoothness, and contact safety), but with a strict priority: these are qualities to improve provided that the task still gets done.

The standard approach to combining multiple objectives is a weighted sum of rewards. This scalarization fails to encode the priority structure we actually want because it treats each objective as equally interchangeable. For example, when success and energy consumption enter the same additive reward, the optimizer can always compensate for a drop in one by buying enough of the other. Nothing in the objective forbids trading away a few percentage points of task success for a larger reduction in energy. This fails because success is meant to be a hard requirement rather than an additional quantity to be traded off against the rest. Additionally, a fixed weighting produces a single, frozen policy. If an operator later wants to favor smoothness in one deployment and speed in another, there is no way to shift that balance without retraining from scratch.

The contributions of this report are summarized as follows:

- Developed a constrained multi-objective post-training method that holds task success as a hard requirement while improving secondary objectives beneath it, with a single policy spanning a family of trade-off settings.
- Validated the method on two multi-objective control benchmarks, demonstrating substantially higher constraint satisfaction and objective-space coverage relative to a conventional multi-objective baseline, and confirming that the constraint-enforcement mechanism responds correctly to both requirement strictness and preference.

- Built and validated a complete training and evaluation pipeline for a compact VLA on the LIBERO manipulation benchmark, including a competent supervised baseline across all four standard suites.
- Identified a limitation of standard LIBERO for evaluating multi-objective methods, which is now directing our choice of environment for the full VLA demonstration.

The rest of this report is structured as follows. Section 2 summarizes the reinforcement-learning and VLA background the project builds on. Section 3 describes the validation environments, computing resources, and VLA pipeline. Section 4 presents results from the control benchmarks and the VLA baseline. Section 5 concludes and describes remaining work.

2. Related Works

2.1. Policy Optimization for Large Models

Proximal Policy Optimization (PPO) [7] is a widely-used policy-gradient method that constrains each update to stay close to the previous policy via a clipped surrogate objective, trading off sample efficiency against training stability. Applying PPO to very large models is complicated by the need for a learned value function of comparable size, which doubles memory cost and is prone to mis-fitting in early training. Group Relative Policy Optimization (GRPO) [8] removes this requirement; rather than learning a value baseline, GRPO samples a group of trajectories under the same context and standardizes their returns within that group to form advantages directly. This lowers memory cost substantially and has made GRPO a common choice for post-training large pretrained models, which is the setting our project operates in.

2.2. Constrained and Multi-Objective Reinforcement Learning

Constrained Markov decision processes [1] formalize the problem of maximizing one objective subject to bounds on others, classically solved via Lagrangian relaxation: a multiplier is introduced for each constraint, and primal and dual variables are optimized in alternation. Paternain *et al*. [6] show that a broad class of constrained reinforcement-learning problems has zero duality gap, meaning the constrained problem can be solved exactly through its dual rather than only approximately, which is the property that licenses this approach for our floor-based requirement.

Separately, multi-objective reinforcement learning seeks policies that span a Pareto front rather than optimizing a single fixed scalarization. Preference-conditioned methods train one network across an entire family of objective weightings rather than a separate network per weighting, an idea explored in work such as the envelope Q-learning approach of Yang *et al*. [9] and extended to continuous-control settings by Lu *et al*. [5]. Our project draws on both lines: a constrained formulation for the requirement that must always hold, and preference conditioning for the objectives that should remain adjustable.

2.3. Vision-Language-Action Models

Vision-language-action models such as OpenVLA [3] fine tune a pretrained vision-language backbone to output robot actions, typically as discretized tokens compatible with the model's existing language-modeling head. MiniVLA [2] is a compact variant built for faster iteration, using a smaller language backbone alongside DINOv2 and SigLIP visual encoders. Such models are conventionally trained and evaluated for task success on manipulation benchmarks such as LIBERO [4], which provides four suites of tabletop manipulation tasks (Spatial, Object, Goal, and Long) with paired language instructions and demonstrations. We are not aware of prior work post-training a VLA under a hard success requirement together with a tunable secondary-objective trade-off, which is the gap this project addresses.

3. Materials & Methods

3.1. Validation Environments

Before applying our method to a VLA, we validated it on two multi-objective control benchmarks chosen to test different things. The first, MO-MountainCar, asks whether the method produces better outcomes than a conventional multi-objective baseline. The agent must reach the goal within a time budget while economizing on two competing categories of control effort, and we compare against the same underlying optimizer with the requirement removed, over four difficulty settings and three random seeds, training for 1,000 iterations per seed. The second, MO-Hopper, is a higher-dimensional continuous-control problem (forward progress, hop height, and control cost) used to test whether the method's internal enforcement mechanism behaves as its design intends, rather than only whether it produces good numbers; training here ran for 2,000 iterations with a single seed. In both environments, several objectives are evaluated per iteration by sampling a batch of preference and requirement settings and rolling out multiple trajectories per setting, in the style of GRPO's group sampling.

When comparing methods by how much of the achievable objective space they recover, we measure hypervolume against a single reference point shared across every method and every difficulty setting, rather than a separate reference per condition. This choice is what makes the resulting numbers comparable at all; a per-condition reference can make an easier setting look artificially better simply because its

reference point is closer to the achievable points.

3.2. Computing Resources

Control-benchmark experiments were run on CPU, since the underlying networks are small. VLA training and inference were run on the Newton GPU Cluster at the UCF Advanced Research Computing Center (ARCC), using NVIDIA H100 GPUs under SLURM for distributed post-training and supervised fine-tuning jobs.

3.3. VLA Model and Fine-Tuning

Our VLA is MiniVLA, a 0.5B-parameter model with a Qwen2.5 language backbone and DINOv2/SigLIP visual encoders, which emits discrete categorical action tokens through its language-model head. This property makes the per-token log-probabilities our post-training method needs directly available, which is why we deliberately avoided vector-quantized action chunking in our configuration.

Post-training under our method requires a base policy that already succeeds some of the time; if success is near zero for every sampled trajectory, the group-relative comparison the optimizer relies on carries no signal. We therefore first supervised fine-tuned MiniVLA separately on each of LIBERO's four standard suites, training with per-device batch size 16 and image augmentation enabled, for step budgets of 80,000 (Spatial, Object, Goal) and 180,000 (Long) chosen to match samples seen across suites of different demonstration length. Results are reported in Section 4.

3.4. Approach

The method we developed treats task success, or a task-progress proxy, as a requirement the policy must satisfy, and improves the remaining objectives beneath that requirement rather than alongside it in a weighted sum. The requirement is enforced through primal-dual optimization, so that the pressure applied to maintain it adjusts automatically as the requirement becomes easier or harder to satisfy. The policy is conditioned on the trade-off setting among secondary objectives, so a single trained model spans the whole family of trade-offs and an operating point is selected at evaluation time rather than fixed during training. Because this method is novel and is being prepared for publication, we do not give its formulation here; the remainder of this report concerns how it was validated, what was built, and what was measured.

4. Results

MO-MountainCar. Table 1 reports the comparison against the conventional unconstrained multi-objective baseline across four difficulty settings, ordered from strictest to loosest. Our method satisfies the task requirement on 77–99% of evaluated policies against 8–19% for the baseline, and recovers between 1.6× and 3.8× as much of the achievable secondary-objective space, with the largest margins at the strictest settings.

Setting	Method	Requirement met (%)	Coverage
strictest	baseline	8.1	325
	ours	**98.7**	**1236**
	baseline	8.9	502
	ours	**99.3**	**1642**
	baseline	11.9	1477
	ours	**91.5**	**2467**
loosest	baseline	19.3	2206
	ours	**77.1**	**3511**

Table 1. MO-MountainCar, mean over three seeds at four difficulty settings. "Requirement met" is the fraction of evaluated policies satisfying the task requirement; "coverage" is the recovered hypervolume of the achievable secondary objective space, computed against a single shared reference. Our method wins on both metrics at every setting.

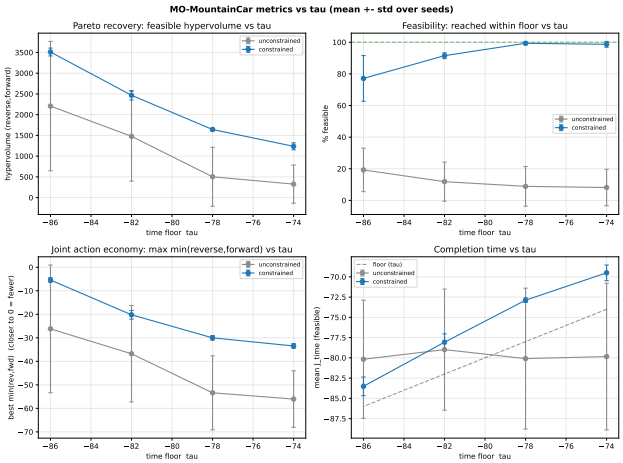

Figure 1. MO-MountainCar performance across difficulty settings, mean ± std over three seeds. Our method satisfies the task requirement at nearly every setting while the conventional baseline does not, and recovers more of the achievable objective space throughout.

Two results in this table are worth noting individually. The baseline's satisfaction rate is nearly insensitive to the strictness of the requirement, remaining between 8% and 19% regardless of what is asked of it, which is the specific failure mode our method is designed to correct. Our method is also roughly an order of magnitude more consistent across seeds than the baseline, whose run-to-run variance is large enough that whether it produces a usable result at all is close to a coin flip.

LIBERO suite	Success rate
Spatial	72%
Object	70%
Goal	75%
Long	40%

Table 2. MiniVLA success after supervised fine-tuning on each standard LIBERO suite, establishing the base policy for post-training.

MO-Hopper. The control-benchmark results above show that the method produces better outcomes; they do not by themselves show that it does so for the reason we claim. On MO-Hopper we examined the internal constraint-enforcement mechanism directly, which makes a specific, falsifiable prediction: while the requirement is easy to satisfy the pressure to maintain it should stay near zero, and as the requirement approaches the limit of what the policy can achieve that pressure should rise and begin to dominate the objective. This is exactly what we observed. Over the easier portion of the requirement range the enforcement pressure remained flat and small; past a threshold it turned upward steeply, increasing by more than an order of magnitude, and the fraction of sampled batches in which the requirement was actively binding rose over the same interval. The mechanism also responded coherently to the preference setting: weighting a competing objective more heavily raised the enforcement pressure monotonically. We take these results as evidence that the requirement is being actively enforced by a working mechanism rather than satisfied incidentally. This configuration was run with a single seed, so we read it as evidence about mechanism rather than as a precise quantitative measurement.

VLA baseline. Table 2 reports MiniVLA success after supervised fine-tuning on each standard LIBERO suite. The three short-horizon suites reach the low seventies, which is sufficient for our post-training method to have a signal to work with. LIBERO-Long reaches 40%; diagnosing this gap exposed an episode-length cap in our evaluation harness that was truncating long-horizon tasks before they could complete, and the reported figure is after that fix.

Benchmark analysis. While beginning post-training on the VLA, we analyzed how the candidate secondary objectives (control energy, motion smoothness, and contact safety) actually behave across the standard LIBERO suites. The analysis found that these objectives do not vary independently of one another or of task success on standard LIBERO, which limits how much any multi-objective method can be exercised there regardless of the method used. This result has redirected our choice of evaluation environment: we are now evaluating richer manipulation benchmarks, including a safety-oriented LIBERO variant with graded collision objectives, as the setting for our full VLA demonstration. We defer the analysis itself to the forthcoming publication.

5. Conclusion and Future Work

5.1. Conclusion

In this project, we developed a constrained multi-objective post-training method that holds task success as a hard requirement while improving secondary objectives beneath it, with a single policy conditioned on the full family of trade-off settings. We validated the method on two multi-objective control benchmarks: on MO-MountainCar it substantially outperforms a conventional multi-objective baseline in both constraint satisfaction and objective-space coverage, and on MO-Hopper its internal enforcement mechanism reproduces the behavior the underlying theory predicts. We built and validated a complete training and evaluation pipeline for MiniVLA on LIBERO, obtained a competent supervised baseline across all four suites, and began post-training under the new method. An analysis of the candidate secondary objectives on standard LIBERO found limited genuine competition among them, which is directing our choice of evaluation environment going forward.

5.2. Future Work

Three items remain. Post-training on the VLA needs to be carried through to a full demonstration in an environment that affords genuine multi-objective competition, most likely a safety-oriented LIBERO variant. Because demonstrations in such datasets are typically collision-free, we plan to fine-tune only to the point of task competence rather than to convergence, so that the safety objective remains something the method has to actively steer rather than something already solved by imitation. Finally, the combined results need to be prepared for submission; a complete technical account, including the method's formulation, will follow as an expanded report once that publication is available.

Acknowledgments

We thank Dr. Mubarak Shah, Dr. Niels da Vitoria Lobo, and Melody Halbert at the UCF Center for Research in Computer Vision, and Dr. Yue Wang for his advising throughout this project, for their input on this report. We thank Zain Ulabedeen Farhat, who carried out the MO-MountainCar experiments reported in Section 4. We also thank NSF REU for this opportunity.

References

[1] Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999. 2

[2] Suneel Belkhale and Dorsa Sadigh. Minivla: A better vla with a smaller footprint, 2024. 2

[3] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 2

[4] Bo Liu, Yifeng Zhu, Chongkai Gao, et al. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023. 2

[5] Haoye Lu, Daniel Herman, and Yaoliang Yu. Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[6] Santiago Paternain, Luiz F. O. Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. 2017. 2

[8] Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. Introduces Group Relative Policy Optimization (GRPO). 2

[9] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2