Extended Abstract

Motivation Vision-Language-Action (VLA) models combine visual perception, natural language understanding, and action generation to enable robots to follow human instructions. While large VLA models like OpenVLA (7B parameters) demonstrate impressive generalist capabilities, their size hampers real-time deployment on small platforms. This project explores natural language based navigation for a low-cost quadruped (Spot Micro) by developing a compact, real-time VLA policy. Our motivation is to enable open-source quadruped robots to interpret commands like "Go to the banana" and autonomously navigate to the goal, addressing the lack of existing language-conditioned locomotion benchmarks in robotics. We build on recent advances such as OpenVLA, TinyVLA and SmolVLA and adapting them to quadruped control.

Method We adopt a two-pronged approach combining imitation learning and offline reinforcement learning under a unified vision-language framework. Using NVIDIA Isaac Sim with a Boston Dynamics Spot model, we replicated the Spot Micro's sensor setup (forward RGB camera) and collected a dataset of navigation trajectories. A pretrained walking policy was utilized for the spot and a custom locomotion controller was built to guide the robot to visual targets, and we logged egocentric images, language instructions, and continuous velocity commands (commanded velocity as well as observed velocity). The resulting dataset spans 63 simulated scenes (e.g., different object arrangements) with up to 15750 image-action pairs. We initially fine-tuned a TinyVLA model on this dataset. To enable closed-loop simulation, we later transitioned to the SmolVLA architecture via the LeRobot framework, leveraging its simplified and modern code. In parallel, we implemented an Implicit Q-Learning (IQL) agent, an offline RL algorithm, to compare against the supervised VLA policy. The IQL agent uses a separate pipeline (ResNet50 for image features and a Sentence-Transformer for language embedding) and learns a Q-value and policy from the same demonstration data augmented with sparse reward labels. This combination of methods allows us to benchmark supervised learning versus offline RL for language-conditioned quadruped control.

Challenges Key challenges included hardware reliability (prompting the sim pivot) and technical integration barriers. We initially attempted on-hardware data collection with a custom Spot Micro robot. After a rear-left hip servo failure hindered real-world experiments, we pivoted to simulation. Running the learned policy in Isaac Sim required resolving Docker and environment conflicts, which blocked closed-loop testing. To address this, we transitioned from TinyVLA to SmolVLA.

Results Our TinyVLA policy achieved smooth convergence – training loss dropped quickly and plateaued around 0.1, with validation loss closely tracking (no overfitting). The SmolVLA-finetuned policy successfully interprets instructions to reach target objects in simulation, demonstrating effective language grounding. In evaluation scenarios, the VLA model achieved low training loss even with minor fine-tuning, comparable to an offline RL baseline built with Implicit Q-Learning (IQL). The IQL agent, trained on the same data with reward labels, attained a 78.8% success rate and average return of 6.64. This implies that in roughly 4 out of 5 test episodes, the robot successfully executed the instruction under the IQL policy. In Isaac Sim rollouts, both policies could reach their instructed goals in some cases, yet SmolVLA's 5k step checkpoint remained decisive where IQL occasionally halted mid-path. Remarkably, SmolVLA was able to go toward an unseen object (mug) while the reward-driven IQL ignored the out-of-distribution object.

Conclusion We present the first steps toward real-time vision-language navigation on an open-source quadruped. Our main contribution is a closed-loop system that unifies high-level language understanding with low-level locomotion control. A compact VLA model successfully maps visual observations and natural-language goals to continuous actions at runtime, grounding intent without intermediate waypoints. The IQL agent shows that incorporating reward feedback can improve performance, suggesting a path for refining VLA policies through RL fine-tuning. A key outcome is the release of a language-labeled quadruped dataset, addressing the lack of public data for language-conditioned locomotion. Future work will focus on real-world deployment, dataset expansion, and hybrid learning methods that combine imitation and RL for improved safety and generality. All code, dataset and models are available in our open-source repository here.

Real-Time Vision-Language-Action Control for Open-Source Quadruped Robots

Sagar Manglani
Department of Computer Science
Stanford University
sagarm@stanford.edu

Abstract

Real-time language-conditioned control of robots promises intuitive and flexible operation of complex systems. In this work, we investigate Vision-Language-Action (VLA) models for commanding a quadrupedal robot via natural language instructions. We target an open-source Spot Micro quadruped and develop a system that perceives its environment through onboard vision and responds to high-level text commands with low-level locomotion actions. Our approach encompasses two learning paradigms: (1) SmolVLA, a compact diffusion-based policy fine-tuned on robot demonstration data, and (2) Implicit Q-Learning (IQL), an offline reinforcement learning agent optimized on the same data augmented with reward signals. The SmolVLA policy learns to map visual observations and language to continuous velocity commands, achieving low prediction error and logical behavior mirroring the demonstrations. The IQL agent, leveraging reward feedback, attains an average success rate of about 79% on held-out test scenarios, slightly outperforming the imitation policy in goal completion. We also present what we believe is the first comparison of VLA and offline RL approaches on quadruped locomotion. Key contributions include the new simulated instruction-following dataset and insights into the trade-offs between supervised and RL-based language grounding. All code, dataset and models are available in our open-source repository here.

1 Introduction

Endowing robots with the ability to follow natural language commands is a long-standing goal in robotics and AI, offering an intuitive interface for human-robot interaction. The Vision-Language-Action (VLA) paradigm provides a compelling framework for this capability: a model perceives the environment through sensors, interprets user intent via language, and produces low-level motor actions, thereby closing the loop between vision, language, and control. Recent advances in large-scale multimodal models, such as OpenVLA, have demonstrated that massive policies trained on hundreds of thousands of demonstrations can generalize across a wide range of robotic tasks. However, these systems demand significant compute and are largely limited to domains like manipulation and indoor navigation.

In contrast, quadruped robots present a unique and underexplored challenge. Unlike arms or wheeled bases, quadrupeds operate in dynamic 3D terrains and require rapid, high-frequency control to maintain stability. This makes real-time language grounding especially difficult, as perception, comprehension, and action must be tightly integrated under latency constraints. Moreover, there is a lack of standardized datasets or benchmarks for language-conditioned quadruped locomotion.

In this work, we address this gap by developing a compact, real-time VLA policy for quadruped navigation using the open-source Spot Micro platform. We build a modified Spot Micro platform

((a)) Spot Micro Original ((b)) Customized Robot ((c)) Custom sensing

Figure 1: The Spot Micro Platform

(as shown in Figure 1) to accommodate an Intel Realsense D455 for sensing and a Nvidia Jetson Xavier NX for inference on the edge. Initially planned with real-world data collection, the project pivoted to simulation due to hardware failure, using NVIDIA Isaac Sim to generate training data across diverse environments. We pursue two parallel strategies: (1) an imitation learning approach using a lightweight VLA model called SmolVLA, and (2) an offline reinforcement learning agent trained with Implicit Q-Learning (IQL).

Our contributions are threefold:

1. a novel dataset of vision-language-locomotion trajectories on a quadruped robot,
2. a trained basic VLA and IQL policies for quadrupeds for control with language
3. a comparative analysis of imitation and offline RL approaches for language-guided locomotion.

Ultimately, we aim to push the frontier toward low-cost, instruction-following legged robots—and to make progress on the open challenge of real-time language grounding in complex motor domains.

2 Related Work

2.1 Vision-Language-Action Models

Vision-Language-Action (VLA) models aim to map visual observations and natural language commands directly to motor actions. OpenVLA Kim et al. (2024) is a prominent large-scale VLA model, featuring a 7B-parameter transformer trained on the Open X-Embodiment dataset, comprising nearly one million robot demonstrations across multiple platforms. It integrates dual vision encoders (DINOv2 and SigLIP) with a LLaMA2-based language model, achieving state-of-the-art performance across manipulation tasks. However, the model's size and latency make it impractical for real-time control on low-cost hardware.

Several efforts aim to improve VLA efficiency. OpenVLA-OFT Kim et al. (2025) enables optimized fine-tuning with continuous actions, while TinyVLA Wen et al. (2024) introduces a diffusion-based policy head and lightweight vision-language backbones to generate continuous actions orders-of-magnitude faster inference than OpenVLA. FAST (Frequency-space Action Sequence Tokenization) Pertsch et al. (2025) compresses continuous action sequences into a small set of frequency-domain tokens via discrete cosine transforms. This reduces token sequence length, though its autoregressive decoding can increase per-step latency. SmolVLA Shukor et al. (2025), based on SmolVLM Marafioti et al. (2025), released more recently, builds on these ideas with an asynchronous perception-control architecture. A pretrained 500M vision-language backbone is paired with a lightweight action prediction module. The model outputs low-frequency action chunks asynchronously, allowing real-time deployment on modest hardware. SmolVLA is specifically designed for affordable platforms and open-source training pipelines, aligning closely with our project goals.

2.2 Language-Guided Quadruped Locomotion

Applying VLA models to quadrupeds introduces unique challenges. The QUAR-VLA paradigm Ding et al. (2023) represents one of the first efforts in this space. QUART, a transformer-based

model trained on the QUARD dataset, supports tasks such as navigation and manipulation on legged robots. An extension, QUART-Online Tong et al. (2024), tackles real-time control using Action Chunk Discretization (ACD), which maps continuous actions to prototype "chunks" for low-latency execution. While QUART-Online reports an approximate 65% relative increase in task-success over the original QUART baseline on the QUARD benchmark, its trained weights, source code, and a downloadable version of the QUARD dataset have not yet been released, and its Action-Chunk Discretisation (ACD) represents each continuous trajectory with a small set of prototype chunks, limiting fine-grained locomotion control.

In contrast, our policy preserves the full continuous action space: a flow-matching action expert predicts multi-step action chunks that are executed asynchronously, enabling real-time control without any discretisation. To the best of our knowledge, this is the first open-source VLA controller that operates a quadruped at control-loop frequency while maintaining continuous, high-bandwidth joint commands.

2.3 Offline Reinforcement Learning

Offline reinforcement learning (RL) enables training policies from fixed datasets without additional environment interaction. Among these, Implicit Q-Learning (IQL) Kostrikov et al. (2021) is a state-of-the-art method that learns an optimal policy via advantage-weighted regression. It avoids common pitfalls of older offline RL algorithms, such as instability from out-of-distribution queries. Prior works have explored combining offline RL with language-conditioned tasks, though mainly in manipulation or simulation domains. In our work, we extend this idea to quadruped locomotion: using the same dataset of vision-language-annotated episodes, we train an IQL agent and present it as an alternative to our SmolVLA imitation policy.

3 Method

Our system consists of two learned components: (1) a Vision-Language-Action (VLA) policy network, SmolVLA, trained by imitation learning, and (2) an Implicit Q-Learning (IQL) agent trained on the same dataset with reward annotations. Both take as input the robot's camera image and a text instruction but differ in architecture and training paradigm.

3.1 Data Collection

We first attempted real-world data collection using a Spot Micro quadruped, where teleoperated trajectories were recorded as $(\text{image}, \text{instruction}, \text{action})$ sequences. However, a mechanical failure halted this effort early. We then transitioned to simulation using Isaac Sim by NVIDIA, deploying a Spot robot in diverse indoor scenarios (rooms, corridors, landmarks) and defining natural language goals (e.g., "Go to the banana"). A pre-trained flat-terrain RL policy was deployed on this robot to enable control through commanded velocity.

Expert trajectories were generated via manual control to certain locations. Data was recorded at 50 Hz, with episodes capped at 5 seconds (250 timesteps), though many terminated earlier upon task completion. The robot's RGB camera and odometry (linear/angular velocity) formed the state input, while expert velocity commands were used as actions.

Each episode is a sequence paired with a single natural-language instruction L:

$$\{(s_t, a_t)\}_{t=0}^{T}, \quad T \leq 250 \tag{1}$$

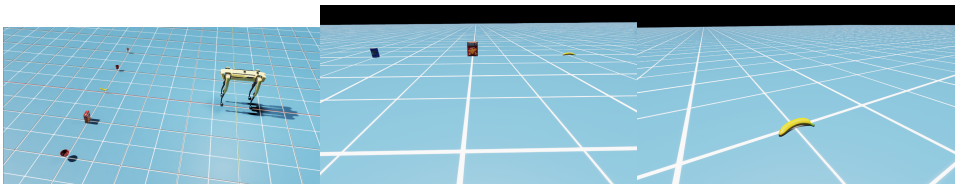
The state s_t includes the image and base velocity components:

$$s_t = \left(I_t,\, v_t^x, v_t^y, \omega_t\right) \tag{2}$$

The action a_t represents the commanded velocity:

$$a_t = \left(v_{\text{cmd},t}^x, v_{\text{cmd},t}^y, \omega_{\text{cmd},t}\right) \tag{3}$$

The resulting dataset includes 63 episode. Idle prefixes (where robot initialized) and suffixes (where robot was stationary) were trimmed to reduce redundancy. Images were collected at 1280×720 resolution and downsampled to 640×360, while textual instructions were concise, limited to 3–5 words. Figure 2 shows example images from the dataset.



((a)) Scene setup ((b)) Start frame ((c)) End (goal) frame

Figure 2: Images from the dataset. Each frame is paired with natural language instruction, visual observations, and action/state velocities.

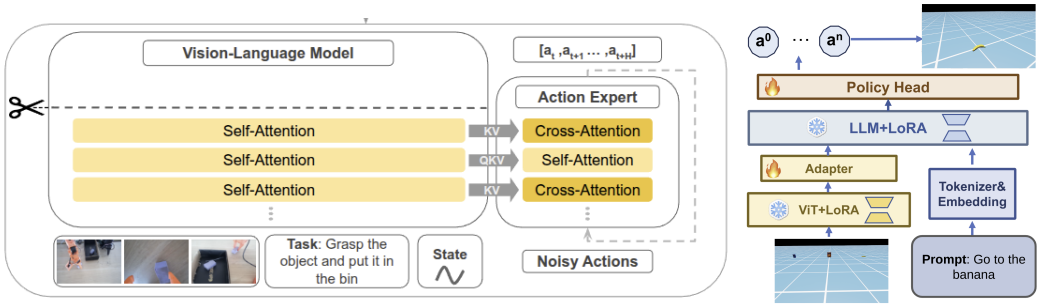
Another variation of this dataset was created for training the IQL agent, where each episode is modeled as an offline MDP with sparse reward annotations. Rewards were assigned as follows:

- $+10$ at success (final step),
- -0.01 per step to encourage shorter paths,
- -0.01 per action to encourage smoother, more efficient behaviors.

This formulation enabled the IQL agent to optimize for both goal completion and control efficiency using the same visual, language, and state inputs.

3.2 VLA Policy Architecture

We fine-tune the SmolVLA architecture, illustrated in Figure 3, which builds on SmolVLM by repurposing its later layers into attention-based modules specialized for action prediction. The model takes in three input modalities: vision, language, and robot state. For visual input, a frozen SigLIP encoder processes the RGB camera images into latent embeddings that capture spatial and semantic cues from the scene. The instruction is handled by a frozen language encoder, which converts the natural-language prompt into a dense representation. Meanwhile, the robot's proprioceptive state—its linear and angular velocities v_x, v_y, w_z is first normalized and then passed through a small multilayer perceptron (MLP) to produce a state embedding. These three streams are fused and passed through the model's transformer-based policy head during fine-tuning.



((a)) SmolVLA Architecture ((b)) TinyVLA Architecture

Figure 3: SmolVLA and TinyVLA architectures: a lightweight multimodal transformer combining visual, language, and state inputs to predict future velocity commands.

These three modalities are tokenized and fed into a multimodal transformer (16 layers, ∼90M parameters). The latter layers, responsible for action prediction (the "Action Expert"), are initialized from scratch. The output is an action chunk — a short sequence of future velocity commands (here, $H = 10$ steps = 0.2s horizon). Note that we use $H = 50$ steps during training, giving it a horizon of 1s. We apply only the first action from the predicted chunk at each step (MPC-style). Training

performs flow matching between predicted and expert sequences, using the AdamW optimizer ($1e{-}4$, cosine decay with 100-step warmup).

Another architecture we worked with during this project is TinyVLA, as shown in Figure 3. For this, we encode the image using a ViT encoder and the prompt using a tokenizer. A diffusion-based policy head is trained (with a small MLP projector in front) while the VLM is fine-tuned via LoRA.

3.3 Implicit Q-Learning Agent

Our IQL implementation follows the practical structure proposed in the IQL paper, consisting of three losses. The Q-functions are trained with a temporal difference loss:

$$L_Q = \mathbb{E}(s, a, r, s') \left[\left(Q(s, a) - (r + \gamma V(s')) \right)^2 \right]. \tag{4}$$

The value function $V(s)$ is trained using expectile regression, assigning different weights to over- and under-estimates:

$$L_V = \mathbb{E}(s, a) \left[(\mathbb{1}Q(s, a) > V(s) \cdot \tau + \mathbb{1}Q(s, a) \leq V(s) \cdot (1 - \tau)) \cdot \left(Q(s, a) - V(s) \right)^2 \right], \tag{5}$$

where $\tau \in (0, 1)$ controls the asymmetry. Finally, the policy is trained via advantage-weighted behavior cloning:

$$L_\pi = -\mathbb{E}_{(s,a)} \left[\log \pi(a \mid s) \cdot \exp \left(\frac{Q(s, a) - V(s)}{\lambda} \right) \right], \tag{6}$$

which encourages high-advantage actions while avoiding out-of-distribution exploration.

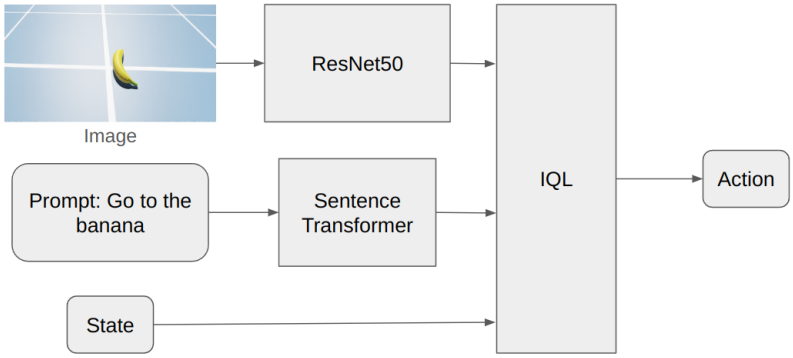


Figure 4: Custom IQL architecture: separate actor and critic networks process visual, language, and state input to compute Q-values and actions.

Our IQL agent uses a custom architecture that processes visual, language, and proprioceptive input through separate encoders before feeding them into the policy and critic networks, as shown in Figure 4. Images are passed through a frozen ResNet-50 pretrained on ImageNet, while instructions are encoded using a SentenceTransformer model. The robot's state is represented by a normalized 3D velocity vector. These three modalities are concatenated into a single observation vector, which is then used by both the actor and critic to predict actions and Q-values, respectively.

These are concatenated and fed into MLPs for the critic and actor networks. The critic predicts both $Q(s, a)$ and $V(s)$, with actions concatenated to features. The actor predicts $\mu(s)$ (mean action), trained via advantage-weighted regression.

The IQL agent uses the same inputs as VLA models, but is trained with sparse reward annotations as described above in the dataset part.

4 Experimental Setup

4.1 Data and Task Description

All experiments were conducted in Isaac Sim 4.5. A 12-DoF Spot was controlled via target base linear (v_x, v_y) and angular velocities (w_z), while a pretrained RL policy generated joint torques. Scenes were flat indoor environments populated with static obstacles and fixed lighting. The front RGB camera is pitched 24 degrees towards the ground and provided 1280×720 images at 50Hz.

All three methods, SmolVLA, TinyVLA, and IQL, were trained on the same set of 63 instruction-following trajectories, comprising approximately 150–200 timesteps each. For SmolVLA, we followed the original guidelines and froze the pre-trained encoders while optimizing the 16-layer "Action Expert" using AdamW with a learning rate of 1e-4 and cosine decay.

TinyVLA was similarly trained on the same GPU but with an emphasis on speed: only the LoRA adapters and lightweight diffusion head were updated.

For IQL, we implemented 5-fold cross-validation on the training set. A ResNet-50 visual encoder (pre-trained and frozen) and SentenceTransformer language encoder were used to compute features. Actor–critic networks with a hidden size of 1024 were optimized using the expectile variant of IQL, producing value estimates and advantage-weighted actions. Fold-specific splits ensured independent validation on disjoint scenes.

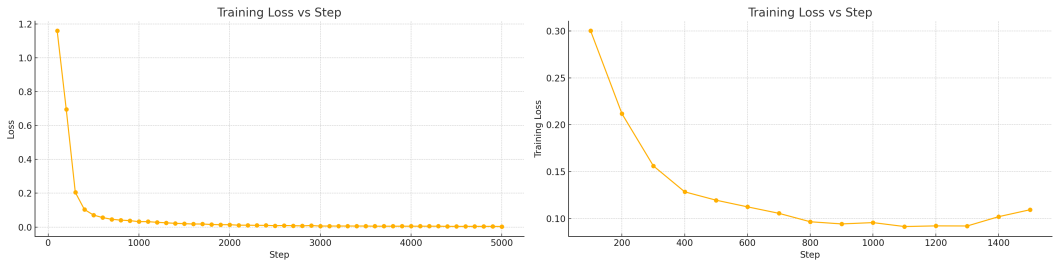
4.2 Evaluation Metrics

1. Imitation loss (MSE): for SmolVLA and TinyVLA, computed on the 4 held-out episodes.
2. Task success rate (IQL only): fraction of trials that reached the goal with >5 reward, measured via reward flag.
3. Return $R = \sum_t r_t$ (IQL only): total reward per episode, where $r_t = +10$ at goal, -0.01 per step, -0.01 per action.
4. Qualitative behavior: visual inspection of trajectory smoothness, responsiveness, and semantic alignment with the instruction.

Checkpoints were saved every 300 steps for TinyVLA and 1 000 steps for SmolVLA. All results reported in results section use the best validation checkpoint for each method.

5 Results

5.1 Quantitative Evaluation



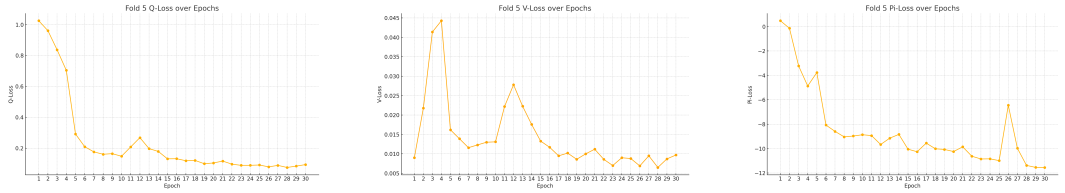
((a)) SmolVLA Train Loss (5k steps)

((b)) TinyVLA Train Loss

Figure 5: Training loss curves for SmolVLA and TinyVLA methods. Both converge rapidly within 1k steps.

Figure 5 illustrates how SmolVLA and TinyVLA both exhibit extremely steep loss decay in the first few hundred optimisation steps: SmolVLA's mean–squared error plunges from ≈ 1.2 to under 0.1 in under 400 updates and falls below 0.01 by step 2000. TinyVLA, with its lighter backbone and diffusion head, reaches a comparable minimum slightly sooner, around step 1000, and shows virtually

flat training/validation traces thereafter, implying it has exhausted the information content of the 63 demonstrations well before the nominal 1.5 k-step budget. Beyond step ≈5k the SmolVLA validation curve begins to oscillate, signalling the onset of over-fitting even though the run was scheduled for 40k total steps; subsequent checkpoints delivered no measurable improvement.



((a)) Q-Loss of Fold 5 ((b)) V-Loss of Fold 5 ((c)) pi-Loss of Fold 5

Figure 6: Fold 5: Implicit Q-Learning loss curves across Q, V, and policy networks.

Figure 6 tracks the Implicit Q-Learning training on Fold 5. All three critic/actor objectives ($\mathcal{L}_Q$, $\mathcal{L}_V$, $\mathcal{L}_\pi$) descend smoothly for roughly 30 epochs and then stabilise: the Q-loss settles near 0.1–0.2, the V-loss mirrors this scale, and the (negative) policy loss plateaus around -10, indicating the actor is reliably selecting high-advantage actions. Additional epochs produced only marginal changes and occasionally increased variance, so training was halted at epoch 30 for every fold. Taken together, the curves confirm that the vision-language policies learn almost instantaneously on this small dataset, whereas the offline RL pipeline converges more gradually but still within a modest computational budget.

Fold	Mean Return	Std Dev	Max Return	Min Return	Success (%)	Episode Length
Fold 1	6.10	4.08	9.54	-2.99	66.7	184.83
Fold 2	6.67	3.96	9.38	-5.53	85.7	158.43
Fold 3	6.93	2.99	9.47	0.39	83.3	184.83
Fold 4	6.40	3.42	9.37	-1.73	76.9	170.54
Fold 5	7.08	3.43	9.51	-4.38	81.3	138.56
Mean	6.64	3.58	9.45	-2.85	78.8	167.44

Table 1: Offline RL performance across 5 folds. Evaluation was done on 4 held-out instruction-following missions.

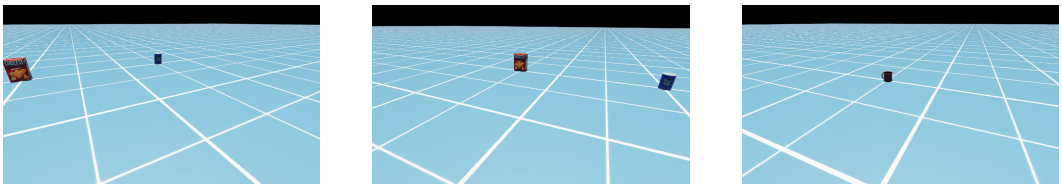
The IQL agent consistently achieved strong task performance as shown in Table 1, with mean returns in the 6.4–7.1 range and a peak success rate of 85.7% in Fold 2. The average return of 6.64 suggests completion of tasks with minimal penalties from time or action costs. Notably, negative returns occurred only in a small fraction of episodes, implying stable policy learning with rare critical failures.

In contrast, SmolVLA's imitation loss (MSE) plateaued around 0.03 on held-out samples, indicating high-fidelity reproduction of expert motion. However, it lacked mechanisms to optimize for reward or efficiency beyond mimicking the data.

5.2 Qualitative Observations

While most of the development was focused on training pipelines, limited model-in-loop evaluations were performed on the final day of project. These real-time rollouts offer initial insights into the behavioral tendencies of the two models.

In early tests, the 40k step checkpoint for SmolVLA was found to be overfit, causing the policy to rigidly reproduce the trajectories in training data—including hesitation or unnecessary turns. As a corrective, we reverted to the 5k step checkpoint, which retained some generality. In this setting, SmolVLA demonstrated more successful behavior: it consistently reached destination goals, often replicating a path similar to but not identical to the training data. Figure 7 shows some of the instances where the model is guiding the spot robot towards different objects. In failure cases, the model still kept moving in a plausible direction, suggesting strong intent grounding from the vision-language fusion, even in uncertain contexts.



((a)) Prompt: Go to the can ((b)) Prompt: Go to cracker box ((c)) Prompt: Go to the mug

Figure 7: Evals: heading towards goal objects

IQL, on the other hand, exhibited mixed outcomes. In some scenarios, the policy was able to complete the instructed trajectory efficiently. However, in other scenes, it stalled mid-path despite clear targets in view. This behavior suggests that the IQL agent had learned penalties for certain state–action pairs (e.g., due to high action cost or ambiguous reward), causing it to default to inaction when uncertain. Unlike SmolVLA's persistent movement even in failure, IQL appeared risk-averse due to an absence of positive examples for out-of-distribution (OOD) scenarios.

One particularly informative test involved placing an OOD object, a mug, in the robot's view. With IQL, the agent consistently ignored the object, failing to incorporate it into its motion plan. This is expected, as the dataset lacked such objects or corresponding rewards. Surprisingly, SmolVLA exhibited emergent attention behavior: the robot moved toward the mug, even without explicit supervision for it. This indicates that SmolVLA's frozen VLM backbone likely encoded semantic priors about common objects like "mug," enabling rudimentary generalization. This highlights the benefit of leveraging pretrained language–vision encoders even in small-data settings.

These findings point toward a complementary relationship: SmolVLA excels in instruction comprehension and perceptual grounding, often producing semantically appropriate behaviors even for unseen objects. However, it inherits the flaws of the demonstration data. IQL, guided by sparse rewards, learns task success more explicitly, but its lack of semantic priors leads to degraded behavior in OOD contexts.

In combination, these observations suggest a promising avenue in future work: leveraging SmolVLA for intent understanding and using offline RL methods like IQL to refine the resulting behaviors toward reward-driven success.

6 Discussion

This work bears several limitations, as described below:

1. Simulation-only data: All training and evaluation occurred in Isaac Sim after a hardware failure halted real-robot trials. This raises familiar sim-to-real concerns: perception domain-shift, unmodeled joint backlash, and contact-dynamics mismatch.
2. Dataset scale and diversity: The 63 recorded episodes cover a narrow slice of indoor navigation and use short, single-goal commands. Policies therefore lack exposure to multi-step tasks, moving obstacles, or rough terrain.
3. Action interface assumptions: Both policies operate on high-level base-velocity targets and assume a perfect low-level controller; uneven ground or foot-slip could violate that assumption.
4. Sparse reward shaping: For IQL we hand-tuned a simple $+10/-0.01$ reward. Different reward shapes (e.g. dense rewards) could change convergence behavior, but we did not explore that space.
5. Language scope: For IQL, pre-trained encoders grant some robustness, yet compositional or abstract phrases ("patrol the room") remain out-of-distribution.

The project encountered three principal difficulties. The first was hardware fragility: while testing a walking gait RL policy on spot, the loss of a hip servo on the physical robot forced a pivot to simulation and cost valuable data-collection time. The second challenge was building custom URDF model for Spot Micro to make it work with Isaac Sim - this model lead to several crashes, prompting

to revert to Boston Dynamics' original Spot model. The third challenge was software integration: Isaac Sim's stack had to be reconciled with TinyVLA's dependencies, which relied on outdated package versions. To address this, we transitioned to SmolVLA, a more modern and recently released alternative. The fourth was maintaining stability during offline reinforcement learning; early IQL runs diverged because of value overestimation and were rescued only after introducing advantage clipping, reducing actor update magnitude, and normalising the replay buffer. Despite these obstacles, the final system demonstrates that compact multimodal models can deliver real-time, language-conditioned locomotion, highlighting both the promise of the approach and the outstanding challenges on the path to robust deployment.

7 Conclusion

This project set out to explore real-time natural language control for quadruped robots using vision-language models and offline reinforcement learning. Despite an early hardware failure of the Spot Micro robot, we successfully transitioned to simulation using NVIDIA Isaac Sim, ultimately training two distinct policies: a SmolVLA-based imitation learning model and an Implicit Q-Learning (IQL) agent. These models offer complementary strengths: SmolVLA leverages pre-trained visual and language encoders to produce smooth, instruction-conditioned locomotion, while IQL optimizes for task completion by learning from reward-annotated trajectories.

Our experiments showed that both models could operate in real time at 50 Hz. Meanwhile, SmolVLA benefited from pretraining to generalize better across varied visual and linguistic inputs, offering robust behavior when the language phrasing were perturbed slightly. Together, these results highlight the trade-offs between behavior cloning and offline RL in the context of language-conditioned locomotion.

A key contribution of this work is the release of a small but carefully curated dataset for vision-language quadruped control. Each episode includes synchronized image, state, action, and natural language instruction sequences, offering a reproducible benchmark for future work. This dataset, combined with open-source training pipelines and pre-trained models, enables other researchers to build on our results or extend them to real-world hardware.

Nonetheless, several limitations remain. The results are confined to simulation, and transferring these models to physical robots will likely require further adaptation, particularly to address visual domain shift and dynamics mismatch. Additionally, our instruction set is also relatively simple; expanding to multi-step commands or abstract goals would push the limits of current architectures and likely require larger models or additional data.

Looking ahead, future directions include deploying the trained models on the real Spot Micro platform, scaling the dataset with more scripted demonstrations, and combining imitation with offline or online fine-tuning to balance generalization with reward optimization. Additionally, expanding the action space to include manipulation, or integrating memory and planning for long-horizon tasks, could enable richer, more interactive behavior.

In summary, this work demonstrates that with compact models, modest data, and careful design, it is possible to achieve responsive, language-driven control of legged robots. Our findings suggest a path forward for accessible, open-source research in language-conditioned locomotion - an area with growing relevance as robots move into everyday human environments.

All code, dataset and models are available in our open-source repository here.

8 Team Contributions

This project was a solo effort by Sagar Manglani. All aspects of the work, from conceptualizing the idea, building and fixing the Spot Micro hardware, setting up the simulation environment, coding the data collection, training and evaluation pipelines, to analyzing results and writing this report, were solely conducted by Sagar Manglani.

References

Pengxiang Ding, Han Zhao, Wenjie Zhang, Wenxuan Song, Min Zhang, Siteng Huang, Ningxi Yang, and Donglin Wang. 2023. QUAR-VLA: Vision-Language-Action Model for Quadruped Robots. *arXiv preprint arXiv:2312.14457* (2023).

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2024. OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv preprint arXiv:2406.09246* (2024).

Moo Jin Kim, Chelsea Finn, and Percy Liang. 2025. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. *arXiv preprint arXiv:2502.19645* (2025).

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169* (2021).

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. 2025. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299* (2025).

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. 2025. FAST: Efficient Action Tokenization for Vision-Language-Action Models. *arXiv preprint arXiv:2501.09747* (2025).

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. 2025. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844* (2025).

Xinyang Tong, Pengxiang Ding, Yiguo Fan, Donglin Wang, Wenjie Zhang, Can Cui, Mingyang Sun, Han Zhao, Hongyin Zhang, Yonghao Dang, et al. 2024. Quart-online: Latency-free large multimodal language model for quadruped robot learning. *arXiv preprint arXiv:2412.15576* (2024).

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, et al. 2024. TinyVLA: Towards Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation. *arXiv preprint arXiv:2409.12514* (2024).