Learning to Walk over Structured Terrains by Imitating MPC

Chao Ni
Robotic Systems Lab
ETH Zürich
chaoni@ethz.ch

Alexander Reske
Robotic Systems Lab
ETH Zürich
areske@ethz.ch

Takahiro Miki
Robotic Systems Lab
ETH Zürich
tamiki@ethz.ch

Marco Hutter
Robotic Systems Lab
ETH Zürich
mahutter@ethz.ch

Abstract: Walking over structured terrains requires perception awareness of the environment. The robot needs to extract information from exteroceptive sensors and generate control signals by either model-based approaches, such as Model Predictive Control (MPC), or inferencing a neural network, trained with Reinforcement Learning (RL) or Imitation Learning (IL). MPC-Net is an IL approach that learns from an MPC expert. It minimizes the control Hamiltonian of the Optimal Control (OC) problem instead of imitating the observation-action mapping directly as Behavioral Cloning (BC). In this work, we add perception to MPC-Net, using demonstrations from Perceptive MPC, which can walk over structured obstacles. We benchmark our MPC-Net approach, validate the policy on the hardware, and show that our policy outperforms the MPC expert under noisy perception inputs.

Keywords: Imitation Learning, Legged Robots

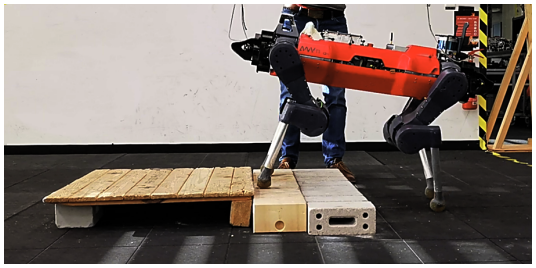

Figure 1: ANYmal [1] walking over an obstacle.

1 Introduction

Legged robots, such as ANYmal [1], Laikago, and Cassie, have recently shown great ability to cope with challenging terrains, including slopes [2, 3], stairs [4, 5], and stepping stones [6, 7]. The agility of the leg makes it suitable for conducting tasks in human-orientated environments, such as industrial inspection, environment exploration, and autonomous navigation. One common challenge of these tasks is to traverse structured obstacles. This requires the robot to be aware of the obstacle by onboard sensors, namely cameras or LiDAR. With visual information, the robot can traverse without compromising its speed, avoid collisions with obstacles, and achieve smooth motions. However, incorporating visual information into the robot locomotion control remains an active area of research.

Model-based methods have been used to cope with such scenarios. The task can be formulated as an Optimal Control (OC) problem and walking over obstacles can be abstracted as selecting feasible foothold locations directly [8, 9] or encoded as constraints that the end-effectors have to satisfy [6]. While Model Predictive Control (MPC) can produce accurate solutions if the obstacle

can be perfectly detected and depicted, it is computationally intense, which adds extra computational constraints to the perception module. Furthermore, the exteroceptive sensors are noisy and not perfect, which may be crucial to the modeling of the problem.

On the other hand, learning-based approaches have seen rapid development in the past years [10, 11, 12, 13, 14, 15]. Among them Reinforcement Learning (RL) [16] and Imitation Learning (IL) [17] are the two main paradigms. RL encodes the problem as a Markov Decision Process (MDP) and guides the agent with task-specific rewards. While RL is capable of overcoming various structured terrains [14, 15], it requires engineering efforts in reward design and until recently took days to train the policy [18]. In contrast, IL learns the policy from expert demonstrations and can train the policy in a short time. It either seeks to replicate the behavior of the expert through learning the observation-action mapping directly, which is called Behavioral Cloning (BC) or learn the demonstrator's reward function for RL purpose, as known as Inverse Reinforcement Learning (IRL). When good demonstrations are available, sample efficiency can be significantly improved compared to RL [19]. However, expert demonstrations for quadrupedal locomotion are often difficult to obtain. Carius et al. [12] proposed to learn locomotion from an MPC expert by MPC-Net. Instead of imitating the action directly, it minimizes the control Hamiltonian, which also encodes the underlying constraints of the OC problem, and therefore improves constraint satisfaction practically. The effectiveness of MPC-Net has also been shown in [13], where a robust multi-gait policy is learned. Nevertheless, most imitation approaches for locomotion are limited on flat terrain [11, 12, 13], where the terrain perception is not available and walking over structured obstacles often leads to failure.

In this work, we propose to traverse structured obstacles with IL via leveraging the knowledge from MPC. MPC can provide an accurate solution given the perfect exteroceptive information in simulation, and we benefit from sample efficiency and improved constraint satisfaction by extending MPC-Net. We use a teacher-student learning framework [20] to incorporate the noisy exteroceptive information. To the best of our knowledge, this is the first approach that achieves perceptive IL for overcoming structured obstacles and transfers to the real robot. Our work is built upon the theoretical principle of a Hamiltonian loss for policy search [12] and the application in robust multi-gait locomotion [13], it contributes the following advances:

- We add exteroception to MPC-Net to traverse structured obstacles.
- Benchmarking experiments confirm that the teacher policy trained with MPC-Net leads to better performance compared to BC.
- Simulation results show that the student policy is more robust to noisy environments than the MPC expert.
- Sim-to-real transfer shows that Perceptive MPC-Net outperforms Blind MPC-Net over a 10 cm step setup.

2 Related Work

This section covers a subset of related work on robot locomotion over uneven obstacles and IL in robotics tasks.

2.1 Locomotion over Uneven Terrain

A large portion of model-based approaches focuses on the selection of foothold location. Jenelten et al. [21] defined a grid map centered at the nominal foothold, where each cell is scored based on manually selected features. Winkler et al. [22] presented a framework for dynamic quadrupedal locomotion over challenging terrains and used a terrain cost map to select foothold locations. Kalakrishnan et al. [23] proposed to decompose the control into subsystems and used expert demonstrations to learn the footholds. Instead of selecting footholds separately, Grandia et al. [6] added the end-effector constraints into the optimization objective. Prediction of the foothold can also be done by a pretrained convolutional neural network (CNN) [8, 9]. Meduri et al. [24] proposed to encode the

foothold selection on uneven terrains as a MDP and trained a stepper via deep RL. Besides predicting foothold locations, there are also works on learning to walk over uneven terrains directly through RL. An end-to-end planner and controller are proposed in [25] where the perception-aware robot learns to walk on difficult terrains through deep RL. Lee et al. [14] trained a policy that can walk on steps by gradually increasing the terrain difficulty during training. Miki et al. [15] incorporated the exteroceptive input explicitly and used a Recurrent Neural Network (RNN) encoder to combine multi-modal information to handle the noisy exteroceptive information.

2.2 Imitation Learning

Learning from expert demonstration has also been widely used in multiple autonomous tasks. Peng et al. [11] proposed to learn quadrupedal locomotion by imitating natural animals based on motion capture data. Johns [26] proposed a coarse-to-fine process to imitate manipulation from a single demonstration. Cao and Sadigh [27] investigated the imperfect demonstrations and score the demonstrations by their feasibility and optimality. An adaptive locomotion skill is learned from multiple motion clips via Generative Adversarial Imitation Learning (GAIL) in [28]. Despite the agility it achieves, it is still limited to the area of animation. Chen et al. [20] proposed a "learning by cheating" schedule for IL, where a privileged agent is trained with ground truth observation in the first stage, and in the second stage, the sensorimotor agent, which has no access to ground truth, learns to imitate the privileged agent.

3 Preliminary

In this section, we give a brief background on MPC and the minimum-principle guided policy search approach: MPC-Net [12].

3.1 Model Predictive Control

We consider the following OC problem

$$\begin{aligned}
\underset{\boldsymbol{u}(\cdot)}{\text{minimize}} \quad & \Phi(\boldsymbol{x}(t_f)) + \int_0^{t_f} l(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\, \mathrm{d}t, && (1)\\
\text{subject to} \quad & \boldsymbol{x}(0) = \boldsymbol{x}_0, && (2a)\\
& \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t), && (2b)\\
& \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t) = \boldsymbol{0}, && (2c)\\
& \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, t) \geq \boldsymbol{0}, && (2d)
\end{aligned}$$

where $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ are the state and input at time t, t_f the time horizon, $\boldsymbol{x}_0$ the initial state, $\Phi(\cdot)$ the final cost and $l(\cdot)$ the intermediate cost. The system dynamics is defined by $\boldsymbol{f}(\cdot)$ and has vectorized equality constraints $\boldsymbol{g}(\cdot)$ and inequality constraints $\boldsymbol{h}(\cdot)$.

To solve the optimization problem, we use a Sequential Linear-Quadratic (SLQ) algorithm [29], which is a variant of the Differential Dynamic Programming (DDP) algorithm. The equality constraints are incorporated via Lagrange multipliers $\boldsymbol{\nu}$ [29] and inequality constraints are handled by a barrier function $b(\cdot)$ [30]. The full Lagrangian of the OC problem (1, 2) is given by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}, t) = l(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\nu}(\boldsymbol{x}, t)^\top \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t) + \sum_i b(h_i(\boldsymbol{x}, \boldsymbol{u}, t)). \tag{3}$$

The solution to the OC problem (1,2) consists of a nominal state-input trajectory $\{\boldsymbol{x}_{\text{nom}}(\cdot), \boldsymbol{u}_{\text{nom}}(\cdot)\}$ and a time-dependent feedback gain $\boldsymbol{K}(t)$, leading to the control policy

$$\boldsymbol{\pi}_{\text{mpc}}(\boldsymbol{x}, t) = \boldsymbol{u}_{\text{nom}}(t) + \boldsymbol{K}(t)(\boldsymbol{x} - \boldsymbol{x}_{\text{nom}}(t)). \tag{4}$$

The OC problem (1, 2) has an associated optimal cost-to-go function

$$V(\boldsymbol{x}, t) = \min_{\substack{\boldsymbol{u}(\cdot) \\ \text{s.t. (2)}}} \left\{ \Phi(\boldsymbol{x}(t_f)) + \int_t^{t_f} l(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau)\, \mathrm{d}\tau \right\}, \tag{5}$$

and the control Hamiltonian is given by

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{u}, t) = \mathcal{L}(\boldsymbol{x}, \boldsymbol{u}, t) + \partial_{\boldsymbol{x}} V(\boldsymbol{x}, t)^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t), \tag{6}$$

which for all t and $\boldsymbol{x}$ satisfies the Hamilton-Jacobi-Bellman (HJB) equation

$$0 = \min_{\boldsymbol{u}(\cdot)} \left\{ \mathcal{H}(\boldsymbol{x}, \boldsymbol{u}, t) + \partial_t V(\boldsymbol{x}, t) \right\}. \tag{7}$$

3.2 Perceptive MPC

Perceptive MPC [6] handles obstacles by first extracting a convex segmentation from the elevation map, coming from measured point clouds or perfect terrain in simulation, and forms state-only constraints

$$h_i^{\text{st}}(\boldsymbol{x}) = \boldsymbol{A}_i \cdot \boldsymbol{p}_{E_i}(\boldsymbol{x}) + \boldsymbol{c}_i \geq \boldsymbol{0}, \tag{8}$$

where $\boldsymbol{p}_{E_i} : \mathbb{R}^{24} \to \mathbb{R}^3$ maps the robot state to the foot position and the matrix $\boldsymbol{A}_i$ and $\boldsymbol{c}_i$ project the position of the foot i onto the target segmentation and form a set of half-space constraints. These state-only constraints are then incorporated into (2) and integrated into the objective by barrier functions [30].

3.3 MPC-Net

MPC-Net [12] imitates MPC and learns a policy by minimizing the Hamiltonian, which encodes the constraints of the OC problem including obstacle-related constraints (8). The policy $\boldsymbol{\pi}(\boldsymbol{x}, t; \boldsymbol{\theta})$ is parametrized by a mixture-of-experts network (MEN), each expert specializing for different modes, where each mode represents a contact configuration. Accordingly, the policy can be written as follows

$$\boldsymbol{\pi}(\boldsymbol{x}, t; \boldsymbol{\theta}) = \sum_{i=1}^{E} p_i(\boldsymbol{x}, t; \boldsymbol{\theta}) \boldsymbol{\pi}_i(\boldsymbol{x}, t; \boldsymbol{\theta}), \tag{9}$$

where E is the number of experts and $\boldsymbol{p} = (p_1, ..., p_E)$ is the weight for each expert computed by a gating network. To improve expert specialization, the cross-entropy is used that allows the incorporation of domain knowledge [13]:

$$CE(\tilde{p}, p) = -\sum_{i=1}^{E} \tilde{p}_i(t) \log \left(p_i(\boldsymbol{x}, t; \boldsymbol{\theta}) \right), \tag{10}$$

where $\tilde{p}_i(t)$ is the probability to observe mode i at time t and is defined as $\tilde{p}_i(t) = \mathbb{1}_{\{m(t)=i\}}$, where $m(t)$ is the commanded mode schedule. This leads to the following loss function for MPC-Net

$$\mathscr{L}_{MPCNet} = \mathcal{H}\left(\boldsymbol{x}, \boldsymbol{\pi}(\boldsymbol{x}, t; \boldsymbol{\theta}), t\right) + CE(\tilde{p}, p), \tag{11}$$

One common problem in IL is the state distribution mismatch [31]. MPC-Net uses a behavioral policy to address the mismatch:

$$\boldsymbol{\pi}_b(\boldsymbol{x}, t; \boldsymbol{\theta}) = \alpha \boldsymbol{\pi}_{\text{mpc}}(\boldsymbol{x}, t) + (1 - \alpha) \boldsymbol{\pi}(\boldsymbol{x}, t; \boldsymbol{\theta}), \tag{12}$$

where α linearly decreasing from one to zero in the training. The behavioral policy executes the rollout in the data generation and the collected data is later used for training.

As proposed in [13], we replace the absolute state with the so-called relative state to achieve better tracking and use a generalized time to guide the gait based on the commanded mode schedule. Please refer to the supplementary material for details. As our method combines Perceptive MPC with MPC-Net, we refer to our approach in later sections as Perceptive MPC-Net.

4 Approach

Perceptive MPC-Net empowers the autonomous agent with the visual ability to traverse structured terrain. Apart from the proprioceptive state estimation, the agent receives exteroceptive information

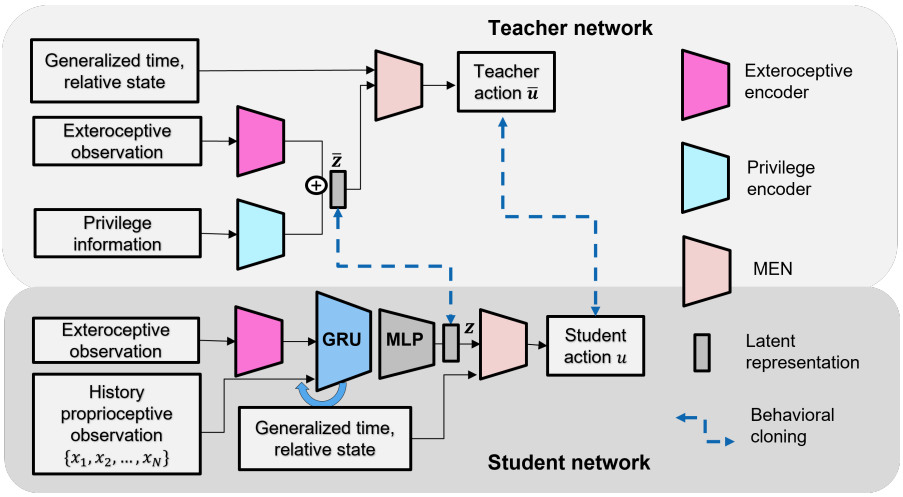

Figure 2: *Top*: Teacher network. Inputs include 1) generalized time and relative state; 2) exteroceptive observation; 3) privilege information (measured contact states, measured contact normals, measured contact forces, as well as measured swing and stance time). Privilege information is obtained from the RaiSim [32] simulator. The encoded exteroceptive observation and privilege information form the latent $\bar{z}$, which together with generalized time and relative state [13], is the input to the mixture-of-experts network (MEN) policy. *Bottom*: Student network. The multi-expert policy is copied from the teacher network and is frozen in the student training. The exteroceptive observation and a sequence of proprioceptive observation are used to reconstruct the robot's consensus about the environment with a Gated Recurrent Network (GRU) [33].

and makes decisions based on the observation. We utilize a two-stage "learning by cheating" framework [20]. First, a privileged teacher is trained by imitating the MPC expert given by Perceptive MPC [6]. Then, we train a student to imitate the behavior of the teacher. The teacher is confined to simulation, while the student can be deployed on the hardware. To avoid confusion about the usage of teacher in two stages in the rest of the paper, we refer to MPC as **expert**, privileged agent as **teacher** and the deployable agent as **student**.

In the following subsections, we explain each training component in detail.

4.1 Teacher

The privileged teacher has access to the ground truth elevation map and privilege information regarding its contact status. The teacher observes a list of scan points around the foothold as its exteroceptive observation. The exteroceptive observation and privilege information are processed with Multilayer Perceptron (MLP) and form the latent representation $\bar{z}$. The downstream MEN is similar to [13] and we extend the input such that it also receives the latent representation of the environment. The complete architecture is shown at the top of Fig. (2). The teacher training is supervised in the MPC-Net fashion and the training objective is defined by (11), where the components are provided by the MPC solver.

4.2 Student

After the teacher policy is trained, we copy the MEN and the exteroceptive encoder from the teacher's network. We freeze the MEN in the course of student training. Instead of the privileged information, a sequence of proprioceptive observations is received. The key assumption is that a sequence of proprioceptive observations helps to reconstruct the latent representation of the environment and the contact status of the agent. The architecture of the student network is shown at the bottom of Fig. (2). We use a Gated Recurrent Network (GRU) [33] to encode the exteroceptive observation as well as proprioceptive information. The noise of the exteroceptive observation can be eased through the averaging effect of the history proprioceptive observations as well as the GRU, and the contact status is related to the phase of locomotion. The GRU outputs a latent representation

$\boldsymbol{z}$ supervised by the teacher's latent feature $\bar{\boldsymbol{z}}$. This representation $\boldsymbol{z}$ is further fed into the MEN with generalized time and relative state, and final action is generated. The student is trained with BC and the objective is defined as

$$\mathscr{L}_{BC} = ||\boldsymbol{u} - \bar{\boldsymbol{u}}||_{\boldsymbol{R}} + ||\boldsymbol{z} - \bar{\boldsymbol{z}}||_2 + \lambda\,||\boldsymbol{z} - \bar{\boldsymbol{z}}||_1, \tag{13}$$

where $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{z}}$ are obtained by evaluating the teacher policy with privileged information given by the RaiSim [32] simulator and $\boldsymbol{R}$ is the cost matrix for balancing different input dimensions. We use a combination of 2-norm and 1-norm loss with a regularizer coefficient λ for the reconstruction of the latent representation.

5 Implementation

This section introduces the setup of our quadrupedal robot ANYmal [1] and the structured terrain it traverses. For training and deployment details, please see the supplementary material.

5.1 Control

The kinodynamic model used by the MPC expert has a 24-dimensional state (base pose, base twist, joint angles) and 24-dimensional inputs (contact forces, joint velocities). The intermediate cost and final cost for the OC problem (1, 2) are formed as follows

$$\Phi(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_d(t_f))^\top \boldsymbol{Q}_f\,(\boldsymbol{x} - \boldsymbol{x}_d(t_f)), \tag{14}$$

$$l(\boldsymbol{x}, \boldsymbol{u}, t) = (\boldsymbol{x} - \boldsymbol{x}_d(t))^\top \boldsymbol{Q}\,(\boldsymbol{x} - \boldsymbol{x}_d(t)) + \boldsymbol{u}^\top \boldsymbol{R}\,\boldsymbol{u}, \tag{15}$$

where $\boldsymbol{x}_d(\cdot)$ is the desired state given by a user-defined reference trajectory. $\boldsymbol{Q}_F$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are cost matrices. The robot is controlled by torques, which are generated by inverse dynamics and PD control:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{id} + \boldsymbol{K}_p \cdot (\boldsymbol{q}_{j,d} - \boldsymbol{q}_{j,m}) + \boldsymbol{K}_d \cdot (\dot{\boldsymbol{q}}_{j,d} - \dot{\boldsymbol{q}}_{j,m}), \tag{16}$$

where $\boldsymbol{\tau}_{id}$ is the inverse dynamics torque, $\boldsymbol{q}_{j,d}$, $\boldsymbol{q}_{j,m}$ are desired and measured joint position, and $\dot{\boldsymbol{q}}_{j,d}$, $\dot{\boldsymbol{q}}_{j,m}$ are desired and measured joint velocities. Moreover, we use a trotting gait in this work.

5.2 Perception

We evaluate our approach mainly on two different terrains: gaps and steps, as is shown in Supplementary Fig. (1). A curriculum factor is set for each terrain and represents the level of traversability. During training, we randomize the parameter of the terrains and the initial position of the robot to increase the variability of exteroceptive observations.

The robot is equipped with two RS-Bpearl LiDARs and an elevation map is constructed from the point clouds. A list of scan points around the foothold is observed and the scan points are circled uniformly around the foothold projection onto the terrain in different radii. For each point, we take its vertical distance to the end-effector as scan input. The scan configuration is shown in Supplementary Table. 2.

6 Results

We evaluate our approach with thorough comparison results in simulation and hardware tests.

6.1 Teacher Benchmarking

We benchmark our teacher policy with BC and MPC expert in the RaiSim simulator used for training. In the case of BC, we replace the objective in (11) with

$$\mathscr{L} = ||\boldsymbol{\pi}_{\text{mpc}} - \boldsymbol{\pi}(t, \boldsymbol{x}; \boldsymbol{\theta})||_{\boldsymbol{R}} + CE(\tilde{p}, p), \tag{17}$$

Perceptive MPC-Net achieves a better performance than the BC policy for both obstacles. The detailed results are shown in Table 1.

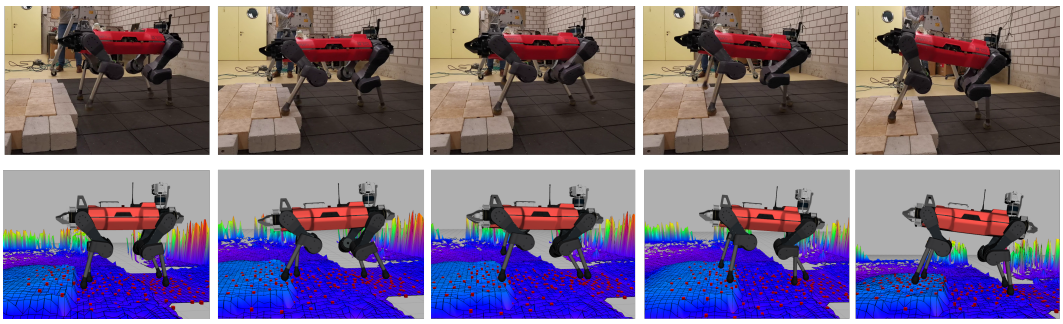

Figure 3: Stepping on a 10 cm step. We create the obstacle with bricks and wooden boxes. The red points are selected scan points. Top: Hardware experiments, Bottom: A replay visualization. From left to right: 1). The robot observes the step and prepares for the lift of its left front (LF) leg; 2). Finishes LF step; 3). Prepares right front (RF) leg; 4) Finishes RF leg; 5). Stabilizes and prepares to walk forward. The full traversal of the obstacles is in the supplementary video.

Table 1: Survival time comparison against MPC expert and teacher policy trained with BC. The evaluation is on a 8 cm wide gap and $\{10, 12, 14\}$ cm high step terrain respectively. We collected 50 episodes, each with maximum 30 seconds.

		MPC	Perceptive MPC-Net	BC
Step	10 cm	$\mathbf{30.00 \pm 0.00}$	$\mathbf{30.00 \pm 0.00}$	29.06 ± 4.61
	12 cm	$\mathbf{30.00 \pm 0.00}$	29.10 ± 4.39	25.05 ± 9.61
	14 cm	$\mathbf{30.00 \pm 0.00}$	26.65 ± 6.99	23.25 ± 9.83
Gap	8 cm	$\mathbf{27.41 \pm 4.56}$	26.21 ± 5.98	24.09 ± 6.67

6.2 Student Benchmarking and Sim-to-Sim Transfer

We compared the performance of our student policy with MPC expert under noisy environments. We set up the step experiments in Gazebo and created a noisy elevation map for both MPC experts and our student policy. The performance is evaluated by the success rate, which is defined as the number of successful traversals over the step divided by total attempts. The noise level of the elevation map is controlled by the standard deviation σ, and the noisy elevation map is defined as:

$$\tilde{p}_{x,y} = p_{x,y} + \mathcal{N}(0, \sigma), \tag{18}$$

where $p_{x,y}$ is the true elevation value and $\tilde{p}_{x,y}$ is the noisy value at the coordinate (x, y) and $\mathcal{N}(0, \sigma)$ is a zero-mean normal distribution. When the noise level or step height increases, MPC fails to find the solution because of incorrect segmentation. On the other hand, our student policy outperforms under the noisy elevation map. The detailed results are in Table. 2.

6.3 Sim-to-Real Transfer

In this subsection, we validate the student policy on the quadruped ANYmal and show the benefit of using two-stage learning with practical evidence. We limit the hardware experiment to the step obstacles. As shown in Fig. (3), the robot detects the obstacle through scan points and lifts the legs high enough to clear the obstacle. It is able to stabilize after stepping on the obstacle and sometimes even compensate for the missed step.

6.3.1 Comparison with Blind MPC-Net

To show the visual ability of our policy, we compared it to the previous work [13], where no perception input is given. We refer to it as Blind MPC-Net. As shown in Fig. (4), since the robot has no perception inputs, it was unable to walk over high steps and failed the task.

Table 2: Success rate comparison against MPC expert under noisy environment. For each case, we forward the robot with same twist command and tried 50 attempts. Note that our policy is only trained with a maximum height 14 cm.

	10 cm		12 cm		14 cm		16 cm	
σ	MPC	Ours	MPC	Ours	MPC	Ours	MPC	Ours
0.000	**1.00**	**1.00**	**1.00**	0.98	**0.98**	0.90	**1.00**	0.08
0.030	**1.00**	**1.00**	**1.00**	0.96	0.30	**0.86**	**0.32**	0.06
0.035	0.48	**1.00**	0.32	**0.96**	0.16	**0.82**	**0.04**	**0.04**
0.040	0.14	**1.00**	0.18	**0.90**	0.08	**0.80**	0.00	**0.04**
0.100	0.00	**1.00**	0.00	**0.80**	0.00	**0.76**	0.00	0.00

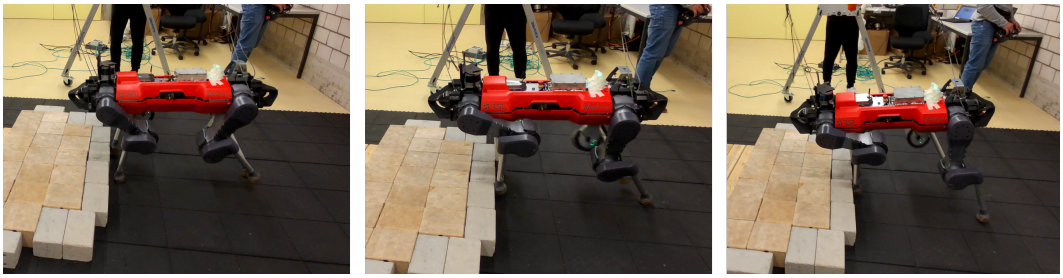

Figure 4: Blind MPC-Net fails to step on high obstacles. From left to right, the robot is not aware of the existance of the obstacle, and therefore keeps the same trotting height. It hit the steps and fails to move forward.

6.3.2 Necessity of the Privileged Teacher

In the end, we investigated the necessity of using a two-stage learning approach. We trained a teacher without using privilege information and only used a MEN to generate the control input. However, it was unable to step over obstacles and was much more sensitive to noises.

7 Limitations

As behavioral cloning, MPC-Net probes the state-dependent distribution of the optimal action from the MPC expert, this can lead to a distribution mismatch, where the student's trajectory diverges from the expert demonstration. One could try to learn to jointly match the state-action distribution to solve this problem. In addition, while MPC-Net significantly improves sample efficiency, it is also confronted with challenges when optimization problem is not feasible. This happens often when the obstacle is too difficult to traverse or the optimization does not generate a physically feasible solution to be executed in the simulator.

8 Conclusion

In this work, we added perception to MPC-Net to walk over structured obstacles by learning from a perceptive MPC expert. A teacher-student framework was used to handle the noisy exteroceptive information and performed better than a single-stage method. We compared the performance under noisy environments against MPC expert and showed our policy is more robust to the noisy elevation map than the MPC expert. The simulation results showed the benefit of using MPC-Net compared against BC. Finally, we validated the viability of the approach on hardware by demonstrating a successful traverse over the structured obstacle.

This work proposed to learn to walk over obstacles by leveraging knowledge from MPC and opens the door to a wider range of traversable terrains. Future work includes training a single policy for different types of obstacles in an uncertain dynamical environment.

References

[1] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, et al. Anymal-a highly mobile and dynamic quadrupedal robot. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016.

[2] J. Ding, Y. Wang, M. Yang, and X. Xiao. Walking stabilization control for humanoid robots on unknown slope based on walking sequences adjustment. *Journal of Intelligent & Robotic Systems*, 90(3):323–338, 2018.

[3] M. Bando, M. Murooka, S. Nozawa, K. Okada, and M. Inaba. Walking on a steep slope using a rope by a life-size humanoid robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 705–712. IEEE, 2018.

[4] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[5] E. Moore and M. Buehler. Stable stair climbing in a simple hexapod robot. Technical report, MCGILL RESEARCH CENTRE FOR INTELLIGENT MACHINES MONTREAL (QUEBEC), 2001.

[6] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *International Conference on Robotics and Automation (ICRA 2021)*, page 3969, 2021.

[7] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath. 3d dynamic walking on stepping stones with control barrier functions. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 827–834. IEEE, 2016.

[8] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini. Mpc-based controller with terrain insight for dynamic legged locomotion. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2436–2442. IEEE, 2020.

[9] O. A. V. Magana, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini. Fast and continuous foothold adaptation for dynamic locomotion through cnns. *IEEE Robotics and Automation Letters*, 4(2):2140–2147, 2019.

[10] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine. Learning to walk via deep reinforcement learning. In *Robotics: Science and Systems*, 2019.

[11] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 07 2020.

[12] J. Carius, F. Farshidian, and M. Hutter. Mpc-net: A first principles guided policy search. *IEEE Robotics and Automation Letters*, 5(2):2897–2904, 2020.

[13] A. Reske, J. Carius, Y. Ma, F. Farshidian, and M. Hutter. Imitation learning from mpc for quadrupedal multi-gait control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[14] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47), 2020.

[15] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62), 2022.

[16] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[17] T. Osa, J. Pajarinen, G. Neumann, J. Bagnell, P. Abbeel, and J. Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018.

[18] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[19] W. Sun, A. Venkatraman, G. J. Gordon, B. Boots, and J. A. Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*, pages 3309–3318. PMLR, 2017.

[20] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020.

[21] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter. Perceptive locomotion in rough terrain–online foothold optimization. *IEEE Robotics and Automation Letters*, 5(4):5370–5376, 2020.

[22] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini. Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*.

[23] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal. Learning, planning, and control for quadruped locomotion over challenging terrain. *The International Journal of Robotics Research*, 30(2):236–258, 2011.

[24] A. Meduri, M. Khadiv, and L. Righetti. Deepq stepper: A framework for reactive dynamic walking on uneven terrain. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2099–2105, 2021.

[25] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2):3699–3706, 2020.

[26] E. Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4613–4619. IEEE, 2021.

[27] Z. Cao and D. Sadigh. Learning from imperfect demonstrations from agents with varying dynamics. *IEEE Robotics and Automation Letters*, 6(3):5231–5238, 2021.

[28] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4): 1–20, 2021.

[29] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli. An efficient optimal planning and control framework for quadrupedal locomotion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 93–100. IEEE, 2017.

[30] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter. Feedback mpc for torque-controlled legged robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4730–4737. IEEE, 2019.

[31] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pages 627–635. PMLR, 11–13 Apr 2011.

[32] J. Hwangbo, J. Lee, and M. Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018. URL `www.raisim.com`.

[33] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.