

Future-Range Conditioning for Imitation-Learned Quadruped Navigation

Rachel Kim • Ivan Sokolov • Marcus Hill

Robotics Institute, Georgia Institute of Technology, Atlanta, GA, USA

PUBLISHED

September 17, 2026

REGISTERED

September 20, 2026

RECORD TYPE

Preprint

ABSTRACT

Imitation-learned navigation policies for legged robots inherit the range representation of the data they are trained on: a policy that observes only the present range scan tends to reproduce how a particular demonstrator reacted to that scan rather than what the robot will actually encounter. This paper studies whether conditioning an imitation policy on a predicted, control-aligned future range field reduces this dependence. A transformer policy consumes the present range scan, a short range history, and the future free range predicted by a FutureRay-style module along the directions the robot is expected to travel under its current command. To keep the policy usable when the predictor is wrong, training applies range-representation dropout that masks or corrupts the future input, so that the policy learns to fall back on present-range behavior. The approach is evaluated with a simulated ANYmal quadruped in Isaac Lab, trained on 12 h of teleoperation from three operators plus 2000 simulator rollouts across 20 cluttered scenes, and tested on 10 unseen cluttered scenes. The conditioned policy reaches a success rate of 88.4 percent, against 76.1 percent for present-range-only imitation and 74.3 percent for a rule-based reactive baseline, raises collision-free distance from 3.8 to 4.2 km per collision, and attains an oracle-parity ratio of 0.81 relative to a policy trained on ground-truth future range. Future conditioning without dropout yields no gain over present-range imitation. Hardware validation remains future work.

INDEX TERMS imitation learning, quadruped navigation, future range prediction, control-aligned perception, representation dropout, ANYmal, Isaac Lab

KAGG NOTE Predicted future range helps an imitation policy only when the policy is also trained to act well without it; dropout turns a fragile extra input into a usable one.

01 Introduction

Learning-based locomotion has made quadrupeds such as ANYmal [2] robust walkers on rough and deformable ground [3, 4, 5], and the navigation layer above the locomotion controller is increasingly learned as well [11, 12, 13]. Imitation learning is an attractive way to obtain this navigation layer: human teleoperation and simulator rollouts are inexpensive to collect, behavior cloning is stable to train, and the resulting policy can be queried at the rate of the perception stack. The difficulty is that a cloned navigation policy learns a mapping from whatever range representation it observes to the demonstrator’s commands. When that representation is a present-time range scan, the policy must infer from a single snapshot how the scene will look once the robot has moved; in practice it learns the demonstrator’s habitual reactions to familiar scan patterns, and those reactions transfer poorly to unseen clutter [29, 30].

Recent perception work suggests a more useful input. FutureRay [1] predicts the free range that the robot will observe along its own predicted, control-aligned directions of travel, so that obstacle queries are expressed in the frame of motions the controller can actually execute. A future range field of this kind encodes exactly the information that a present-range imitation policy has to hallucinate: how much free space remains along the path implied by the current command. This raises a direct question for imitation learning. If a cloned policy is conditioned

on predicted future range, does it generalize better to unseen scenes, and what happens when the prediction is wrong?

This paper answers both parts in simulation. The gap addressed is not the design of a better predictor but the interface between a learned predictor and a learned policy. Simply concatenating the predicted field to the policy input does not help in our experiments: the variant without any safeguard reaches 75.1% success, slightly below the 76.1% of present-range-only imitation. The policy learns to trust an input that is unreliable precisely in the unfamiliar situations where it would be most valuable. The proposed remedy is *range-representation dropout*: during training the future input is masked, partially masked, or corrupted while the demonstrated action is kept, so the policy must remain competent on present range alone and must treat the future field as evidence rather than as ground truth.

The contributions are as follows.

1. A formulation of imitation-learned local navigation conditioned on a control-aligned future range field, with a causal, label-free choice of control proposal that keeps the training and deployment input distributions identical (Sections 3 and 4).
2. A transformer policy that fuses present range, range history, predicted future range, goal, and proprioception through typed tokens, executed at 25 Hz on top of a learned ANYmal velocity-tracking controller (Section 4).

3. Range-representation dropout, a training-time augmentation that masks or corrupts the future input and makes the policy degrade toward present-range behavior when the predictor is wrong (Section 4.4).
4. An evaluation with a simulated ANYmal in Isaac Lab on 10 unseen cluttered scenes: 88.4% success versus 76.1% for present-range imitation and 74.3% for a rule-based reactive baseline, an oracle-parity ratio of 0.81, ablations of each component, and an analysis of the recorded failure cases, including a demonstrator-bias effect (Section 5).

02 Related Work

2.1 Learned locomotion and navigation for legged robots

Reinforcement learning in simulation, followed by zero-shot transfer, is now the standard route to robust quadruped locomotion. Hwangbo et al. trained ANYmal locomotion policies with a learned actuator model [3]; Lee et al. added a teacher–student scheme in which a privileged teacher supervises a proprioceptive student [4]; and Miki et al. fused exteroception with proprioception through a belief encoder that learns when to distrust the height map [5]. Kumar et al. showed that an adaptation module estimating latent environment factors from history enables rapid adaptation without explicit system identification [7], and egocentric-vision and parkour policies extended these ideas to stairs, gaps, and highly dynamic maneuvers [8, 9, 10]. Massively parallel GPU simulation made such training fast [6, 42], and Orbit, the predecessor of Isaac Lab, provides the simulation framework used here [43]. The present work does not change the locomotion controller; it treats a learned velocity-tracking controller of this family as the actuator for a learned navigation layer. The belief encoder of Miki et al. is the closest conceptual precedent to range-representation dropout, since both teach a policy to discount an exteroceptive input when it disagrees with other evidence; the difference is that their mechanism is learned within RL with privileged noise models, whereas dropout here is applied to an imitation objective and to a predicted rather than measured representation.

Several works learn the navigation layer directly. Rudin et al. trained locomotion and local navigation end-to-end with RL [11], and Hoeller et al. learned a compact state representation for navigation in cluttered and dynamic environments before training a planner on top of it [12]; ANYmal Parkour composed perception, navigation, and skill selection into an agile system [13]. ArtPlanner instead combines learned reachability with sampling-based planning for field deployment [14], and Frey et al. learn traversability online from the robot’s own experience [15]. Truong et al. learned navigation skills for legged robots with learned embodiment models [16], ViNL coupled a visual navigation policy with an obstacle-aware locomotion policy [17], and Lee et al. demonstrated hierarchical learned navigation and locomotion for wheeled-legged robots at urban scale [18]. These systems are trained with RL or with planners that have access to explicit geometry. The present study addresses the complementary setting in which the navigation layer is obtained by imitation of human and simulated demonstrations, where the policy cannot explore its way out of a misleading representation.

2.2 Imitation learning for navigation and its failure modes

Behavior cloning for navigation dates back to end-to-end driving [27] and to learned planners for ground robots that map 2D

range scans directly to velocity commands [21]. Pfeiffer et al. are particularly relevant because they cloned a classical planner from range input and observed that the cloned policy fails in situations the demonstrator never encountered [21]. Conditional imitation learning added high-level commands to resolve ambiguity at intersections [28], and ChauffeurNet synthesized perturbations to expose the learner to recoveries [31]. Codevilla et al. documented the limits of behavior cloning, including dataset bias, high variance across training runs, and an inertia problem in which the policy correlates its action with spurious cues [29]; de Haan et al. formalized this as causal confusion, where more information in the observation can make imitation worse because the learner latches onto nuisance correlates of the expert action [30]. DAgger addresses covariate shift by querying the expert on learner-visited states [26], which is costly for human teleoperation. Our negative result for naive future conditioning (75.1% versus 76.1%) is consistent with the causal-confusion account: an extra input that is informative on the training distribution can still hurt off-distribution.

Large cross-robot navigation models such as GNM [35], ViNT [36], and NoMaD [37] show that imitation on heterogeneous data yields reusable visual navigation priors, and action-chunking and diffusion policies improve how multi-modal demonstrations are represented [33, 34]. Mandlekar et al. showed that the quality and the operator composition of human demonstrations strongly affect what is learned, with policies trained on mixed-proficiency data underperforming [32]. This observation is directly relevant to the demonstrator-bias finding in Section 5.5. Our policy is deliberately smaller than these models and uses range rather than images, because the question studied is how the choice of range representation, not model scale, shapes generalization.

2.3 Predictive and control-aligned perception

FutureRay [1] is the representation this work builds on. It predicts, for an agile quadruped, the future free range along a set of directions anchored on the robot’s predicted, control-aligned trajectory, rather than along fixed sensor-frame bearings at the current time. Obstacle queries are therefore posed in terms of what the controller can actually execute: the ray that matters at a given horizon is the one emanating from where the robot will be, pointing where it will be heading. FutureRay uses this field for agile navigation in a pipeline in which the future range is consumed by a downstream planner or controller. The present paper differs in three ways. First, the consumer of the field is an imitation-learned policy, not a hand-designed or RL-trained controller, so the field must coexist with demonstrator-specific action labels. Second, we do not train or modify the predictor; we use a FutureRay-style predictor as a fixed module and study the policy side of the interface. Third, we explicitly study the regime in which the prediction is wrong and design the policy to fall back on present range in that case. In this sense, the work can be read as an investigation of how to make a FutureRay-type representation safe to consume by a learned policy.

Predicting future sensory input for control has a longer history. Visual foresight learns an action-conditioned video prediction model and plans through it [25], and Kahn et al. learn action-conditioned predictions of future collision and progress events and select actions by optimizing over them [23, 24]. These approaches use predictions inside an explicit planner. Here the prediction is a policy input, which avoids

online optimization but makes the policy's reliance on the prediction a learned, and therefore controllable, quantity.

2.4 Privileged information and robustness to missing inputs

Learning by Cheating trains a privileged agent with access to ground-truth state and distills it into a sensorimotor agent [38], and teacher-student training in legged locomotion follows the same pattern [4, 7]. Our oracle, a policy trained on ground-truth future range, plays the role of the privileged agent, but rather than distilling it we use it as a reference: the oracle-parity ratio measures how much of the benefit of perfect future information a policy recovers with predicted information. Range-representation dropout is related to unit dropout [39] and, more closely, to ModDrop [40], which drops entire input modalities during training so that a multimodal network remains functional when a modality is missing. We extend this from missing to wrong inputs by adding a corruption mode, because a predictor fails by producing a confident, plausible but incorrect field rather than by producing nothing.

2.5 Reactive navigation

Artificial potential fields [19] and the dynamic window approach [20] remain strong reactive baselines in clutter because they evaluate commands directly against present range. The dynamic window approach in particular scores constant-velocity arcs over a short horizon, which is the classical analogue of the control-aligned query used by FutureRay. Deep RL mapless navigation from sparse range [22] showed that learned reactive policies can match such controllers on small robots. The rule-based reactive baseline in our experiments belongs to this family and provides a reference for what can be achieved from present range without learning.

03 Problem Formulation

The robot operates in a planar workspace with obstacle set $\mathcal{O}$. At decision step t , with control period $\Delta t = 40$ ms, the navigation policy issues a body-frame velocity command $u_t = (v_{x,t}, v_{y,t}, \omega_t) \in \mathcal{U}$, where $\mathcal{U} = [-0.5, 1.0] \text{ m/s} \times [-0.5, 0.5] \text{ m/s} \times [-1.0, 1.0] \text{ rad/s}$. A learned locomotion controller tracks u_t ; the navigation problem therefore treats the robot as a velocity-controlled planar body with pose $x_t = (q_t, \psi_t) \in SE(2)$, position $q_t \in \mathbb{R}^2$, and heading ψ_t .

The observation at step t is

$$o_t = (r_t, R_t, F_t, g_t, \nu_t), \quad (1)$$

where $r_t \in [0, r_{\max}]^N$ is the present range scan with $N = 360$ angular bins and $r_{\max} = 10$ m; $R_t = (r_{t-k\Delta_r})_{k=1}^T$ is a range history of $T = 8$ earlier scans spaced $\Delta_r = 0.1$ s apart; $F_t \in [0, r_{\max}]^{H \times K}$ is a future range field defined below; $g_t = (d_t, \cos \gamma_t, \sin \gamma_t)$ encodes the goal distance d_t and bearing γ_t in the body frame; and $\nu_t \in \mathbb{R}^6$ contains the estimated body twist and the previously executed command.

Control-aligned future range. Let $\xi = (v_x, v_y, \omega)$ be a command held constant over a horizon of H steps of length δ . The predicted body pose after $h\delta$ seconds follows from integrating the constant twist,

$$\begin{aligned} \hat{\psi}_h &= \omega h\delta, & h &= 1, \dots, H, \\ \hat{q}_h &= \int_0^{h\delta} \text{Rot}(\omega s) \begin{bmatrix} v_x \\ v_y \end{bmatrix} ds, \end{aligned} \quad (2)$$

expressed in the body frame at time t . The control-aligned ray directions at horizon h are centered on the predicted direction of travel,

$$\begin{aligned} \phi_{h,j} &= \hat{\psi}_h + \chi + \beta_j, & j &= 1, \dots, K, \\ \beta_j &= -\bar{\beta} + \frac{2\bar{\beta}(j-1)}{K-1}, \end{aligned} \quad (3)$$

where $\chi = \text{atan2}(v_y, v_x)$ is the course angle of the command (set to 0 when $\|(v_x, v_y)\| < 0.05$ m/s), K is the number of rays, and $\bar{\beta}$ is the half-width of the ray fan. The ground-truth future range is the free distance along each ray from the predicted pose, measured against the obstacle geometry at the corresponding future time,

$$F_t^*[h, j] = \min \left\{ \rho \in [0, r_{\max}] : \hat{q}_h + \rho e(\phi_{h,j}) \in \mathcal{O}_{t+h\delta} \right\}, \quad (4)$$

with $e(\phi) = (\cos \phi, \sin \phi)$ and the convention that the minimum over an empty set equals $r_{\max}$. We use $H = 8$, $\delta = 0.2$ s (a 1.6 s look-ahead, i.e., 1.6 m at the maximum forward speed), $K = 15$, and $\bar{\beta} = 30^\circ$. In simulation F_t^* is available by ray casting; on a robot it is not, and a predictor Φ produces an estimate $\hat{F}_t$ from onboard range data and the command, as in FutureRay [1].

Imitation setting. A demonstration dataset $\mathcal{D} = \mathcal{D}_{\text{tele}} \cup \mathcal{D}_{\text{sim}}$ contains observation-command pairs (o_t, u_t^E) from human teleoperation by operators $k \in \{1, 2, 3\}$ and from simulator rollouts. A policy $\pi_\theta(u \mid o)$ is fitted to $\mathcal{D}$ and evaluated on scenes drawn from a test distribution P_{test} whose layouts do not appear in training. The objective is high success and few collisions on P_{test} , subject to the constraint that π_θ only sees predicted future range $\hat{F}_t$ at test time.

Desired degradation property. Let $\epsilon_t = \max_{h,j} |\hat{F}_t[h, j] - F_t^*[h, j]|$ be the predictor error and let $\pi_\theta(\cdot \mid o_t^\emptyset)$ denote the policy with the future input masked. We seek a policy that (i) approaches the oracle π^* , trained and evaluated with F^* , when ϵ_t is small, and (ii) approaches $\pi_\theta(\cdot \mid o_t^\emptyset)$, a present-range policy, when ϵ_t is large, rather than extrapolating from an incorrect field.

04 Method

4.1 Control structure

The navigation stack has three layers running at different rates. (i) The navigation policy runs at 25 Hz ($\Delta t = 40$ ms) and outputs u_t . (ii) A rate limiter enforces acceleration bounds on the executed command,

$$u_t^{\text{exec}} = u_{t-1}^{\text{exec}} + \text{clip}(u_t - u_{t-1}^{\text{exec}}, -a_{\max}\Delta t, a_{\max}\Delta t), \quad (5)$$

with $a_{\max} = (1.5 \text{ m/s}^2, 1.5 \text{ m/s}^2, 2.0 \text{ rad/s}^2)$ applied elementwise, which keeps command changes within what the locomotion controller tracks reliably. (iii) A learned velocity-tracking locomotion policy of the type introduced in [6], trained in Isaac Lab, maps u_t^{exec} and proprioception to joint position targets at 50 Hz, which are tracked by joint-level PD control. The navigation policy never commands joints directly; this separation lets the same locomotion controller serve all compared navigation methods, so that differences in the results reflect the navigation layer.

4.2 Future range input and control proposal

The future field depends on the command used to define the control-aligned rays in Eqs. (2)–(3). A natural choice would be the command the policy is about to issue, but that command is the label being predicted: using the demonstrator’s own action to compute $\hat{F}_t$ during training would leak the label into the input and invite exactly the causal confusion described in [30]. We therefore define the control proposal as the previously executed command,

$$\xi_t := u_{t-1}^{\text{exec}}, \quad \hat{F}_t = \Phi(r_t, R_t, \xi_t), \quad (6)$$

which is causal, available identically in the demonstration logs and at deployment, and close to the command about to be issued because Δt is short and Eq. (5) bounds the change. The field answers the question “if the robot keeps doing what it is doing, how much room will it have along its path over the next 1.6 s?”, which is the information an imitation policy needs to decide whether to continue, slow down, or turn. Φ is a FutureRay-style predictor [1]; it is trained separately on simulator data with F^* targets and is frozen during policy training and evaluation. The oracle policy π^* is identical to the proposed policy except that it receives F_t^* computed with the same proposal ξ_t .

4.3 Policy architecture

The policy is a transformer encoder [41] over typed tokens. The present scan is divided into $S = 30$ angular sectors of 12° ; a shared 1D convolutional encoder maps each sector’s 12 ranges, in inverse-range form $1/\max(r, 0.1 \text{ m})$ to emphasize nearby obstacles, to a token $z_s^r \in \mathbb{R}^d$ with $d = 128$. The range history is encoded per sector by a temporal convolution over the $T = 8$ past scans and the present one, producing history tokens z_s^R that carry apparent obstacle motion relative to the robot. Each row $\hat{F}_t[h, :]$ of the future field is embedded by a linear map to a future token z_h^F . Goal and proprioception are embedded into one token z^g , and a learned action token z^a collects the output. The input sequence is

$$Z_t = [z^a; z^g; z_{1:S}^r + \eta_{1:S}^r; z_{1:S}^R + \eta_{1:S}^R; z_{1:H}^F + \eta_{1:H}^F] \in \mathbb{R}^{(2+2S+H) \times d}, \quad (7)$$

where η are learned type-and-position embeddings (sector angle for range and history tokens, horizon index for future tokens). Four encoder layers with four heads each process Z_t , and the output at the action token parameterizes a squashed Gaussian over commands,

$$u_t = u_{\max} \odot \tanh(\mu_\theta(Z_t) + \sigma_\theta(Z_t) \odot \varepsilon), \quad \varepsilon \sim \mathcal{N}(0, I), \quad (8)$$

with an asymmetric affine rescaling for the forward-velocity range. At deployment the mean command $u_{\max} \odot \tanh(\mu_\theta)$ is executed. Attention over typed tokens was chosen over concatenation into a single vector because it allows the action token to attend to a subset of future rows, for example only the near horizon, and to ignore masked rows without changing the input dimension, which the dropout scheme below relies on.

4.4 Range-representation dropout

During training each sample receives one of four dropout modes m , namely none, full, partial, or corrupt, with probabilities 0.50, 0.25, 0.15, and 0.10, respectively. Masking acts at the token level:

$$\begin{aligned} z_h^F &= (1 - b_h) z_h^F + b_h z^{\text{mask}}, \\ b_h &= \begin{cases} 0 & m = \text{none}, \\ 1 & m = \text{full}, \\ \text{Bernoulli}(0.5) & m = \text{partial}, \end{cases} \end{aligned} \quad (9)$$

where z^{mask} is a learned embedding. In the corrupt mode the field itself is perturbed before embedding,

$$\begin{aligned} \tilde{F}_t[h, j] &= \text{clip}(s \hat{F}_t[h, j] + n_{h,j}, 0, r_{\max}), \\ s &\sim \mathcal{U}[0.5, 1.5], \quad n_{h,j} \sim \mathcal{N}(0, \sigma_c^2), \end{aligned} \quad (10)$$

with $\sigma_c = 1.0 \text{ m}$, while the demonstrated command is left unchanged. The three modes have distinct purposes. Full masking trains the network to be a competent present-range policy, so that $\pi_\theta(\cdot | o_t^{\mathcal{O}})$ in the degradation property of Section 3 is a meaningful fallback rather than an untrained input region. Partial masking prevents the policy from relying on any single horizon. Corruption, the mode that distinguishes this scheme from modality dropout [40], presents a confident but wrong field together with an action that was chosen from the true scene; the only way to fit such samples is to cross-check $\hat{F}_t$ against the present scan and history, and to discount the future tokens when they disagree. A scale perturbation s was included in addition to additive noise because predictor errors in clutter are typically systematic over- or under-estimates of free space along an entire fan rather than independent per-ray noise.

4.5 Training objective and data

The policy is trained by maximum likelihood on the demonstrations under the dropout distribution,

$$\begin{aligned} \mathcal{L}(\theta) &= \sum_c w_c \mathbb{E}_{(o_t, u_t^E) \sim \mathcal{D}_c} \mathbb{E}_m \left[-\log \pi_\theta(u_t^E | \tilde{o}_t^{(m)}) \right] \\ &\quad + \lambda \mathbb{E}[\|\mu_\theta(Z_t) - \mu_\theta(Z_{t-1})\|_2^2], \end{aligned} \quad (11)$$

where $\tilde{o}_t^{(m)}$ is the observation after applying mode m , c ranges over the teleoperation and simulation sources, the weights w_c balance the two sources so that each contributes half of every mini-batch, and the second term with $\lambda = 0.01$ discourages command chatter between consecutive steps of the same trajectory. The likelihood is evaluated on the pre-squash Gaussian with the change-of-variables correction for tanh. Teleoperation data comprise 12 h from three operators; simulator data comprise 2000 rollouts in the training scenes. Because the field $\hat{F}_t$ is a deterministic function of logged quantities and the frozen predictor (Eq. (6)), it is recomputed offline for every logged step.

4.6 Behavior when the predictor is wrong

At deployment nothing is masked explicitly; the policy always receives $\hat{F}_t$. The fallback in the degradation property arises from training: since full-mask samples teach a present-range solution and corrupt samples teach the policy to discount fields inconsistent with r_t and R_t , an inconsistent field at test time is treated similarly to a masked one. The policy thus does not attempt to extrapolate from a field it cannot reconcile with its present observations. This is a deliberate trade-off: the policy cannot benefit from a correct prediction that happens to contradict the present scan, for example a prediction of free space

Table 1: Design parameters of the navigation layer.

Parameter	Value
Policy period Δt	40 ms (25 Hz)
Locomotion controller rate	50 Hz
Command bounds $\mathcal{U}$	$v_x \in [-0.5, 1.0]$ m/s, $v_y \in [-0.5, 0.5]$ m/s, $\omega \in [-1, 1]$ rad/s
Present scan	$N = 360$ bins, $r_{\max} = 10$ m, $S = 30$ sectors
Range history	$T = 8$ scans, $\Delta_r = 0.1$ s
Future field	$H = 8$, $\delta = 0.2$ s, $K = 15$, $\bar{\beta} = 30^\circ$
Transformer	4 layers, 4 heads, $d = 128$
Dropout modes	none / full / partial / corrupt: 0.50 / 0.25 / 0.15 / 0.10
Corruption	$s \sim \mathcal{U}[0.5, 1.5]$, $\sigma_c = 1.0$ m

behind a currently visible but moving obstacle. Section 5.5 examines how this behavior appears in the recorded trials.

05 Experiments

5.1 Platform and setup

All experiments use a simulated ANYmal quadruped in Isaac Lab [43]. Training uses 20 cluttered scenes with 5 seeds each. The demonstrations are 12 h of teleoperation by 3 operators and 2000 simulator rollouts. The policy is the transformer imitation model of Section 4. Evaluation is performed on 10 unseen cluttered scenes whose layouts do not occur in training. No hardware trials are reported in this version of the manuscript; the platform name refers to the simulated ANYmal model throughout.

REPORTING NOTE

All values in Tables 2–4 are the final recorded aggregates for each method and are reproduced verbatim. Per-seed dispersion was not recorded with these aggregates and is therefore not reported; differences of about one percentage point between methods should be read as ties.

5.2 Compared methods and metrics

Four navigation layers are compared on top of the same locomotion controller. *Imitation*, *present range only* uses the architecture of Section 4 without future tokens. *Imitation*, *present + future (no dropout)* receives $\hat{F}_t$ but is trained without range-representation dropout. *Rule-based reactive baseline* is a hand-designed controller that maps the present scan and goal to velocity commands in the tradition of [19, 20]. *Ours* is the full method: future-conditioned, with range-history encoding and range-representation dropout. The oracle π^* is used only as a reference for the parity ratio.

Four metrics are reported. *Success rate* is the percentage of evaluation episodes in which the robot reaches the goal region before the time limit. *Collision-free distance* is the total distance traveled divided by the number of collision events, in km per collision; larger is better. *Oracle-parity ratio* compares each method with the oracle trained on ground-truth future range, using success weighted by path length (SPL) [44]:

$$\text{parity}(\pi) = \frac{\text{SPL}(\pi)}{\text{SPL}(\pi^*)}, \quad \text{SPL} = \frac{1}{M} \sum_{i=1}^M S_i \frac{\ell_i}{\max(p_i, \ell_i)}, \quad (12)$$

where S_i is the success indicator, ℓ_i the shortest-path length, and p_i the executed path length of episode i . *Decision latency* is the 95th-percentile time from range-scan arrival to command output, including predictor inference for the future-conditioned variants; the budget is the 40 ms control period.

5.3 Main results

Table 2 summarizes the comparison. The full method reaches 88.4% success, 12.3 percentage points above present-range-only imitation (76.1%), 13.3 points above future conditioning without dropout (75.1%), and 14.1 points above the rule-based reactive baseline (74.3%). In terms of failures, the failure rate falls from 23.9% for present-range imitation to 11.6%, a reduction of slightly more than half. Collision-free distance improves from 3.8 to 4.2 km per collision, an increase of about 10.5%, or from roughly 0.26 to 0.24 collisions per km. The gain in collision-free distance is thus considerably smaller than the gain in success; most of the improvement comes from episodes that previously ended without reaching the goal rather than from fewer contacts per kilometer.

The oracle-parity ratio of 0.81 indicates that the conditioned policy recovers 81% of the SPL of a policy given perfect future range; it does not match the oracle, and the remaining gap of 0.19 is attributable to predictor error and to the conservative fallback described in Section 4.6. Present-range imitation attains 0.72, and naive future conditioning attains 0.77. The latter is notable: adding the predicted field without dropout raises parity by 0.05 yet lowers success by 1.0 point and leaves collision-free distance essentially unchanged (3.9 versus 3.8). Because parity is SPL-based, this pattern suggests that the undefended policy uses the field to take more efficient paths in episodes where the prediction is accurate, while losing other episodes in which it trusts a wrong prediction; the recorded aggregates are consistent with this interpretation but do not isolate it.

All methods meet the 40 ms budget at the 95th percentile. The full method has the lowest recorded p95 latency, 18.7 ms, 2.2 ms below present-range imitation, despite running the predictor. The recorded data do not include a timing breakdown, so the source of this difference is not established here; it should not be read as a computational advantage of future conditioning.

5.4 Ablation

Table 3 removes one component at a time. Removing future-range conditioning causes the largest drop, 12.5 points to 75.9%, which is within 0.2 points of the present-range-only baseline in Table 2 and confirms that the future field is the main source of the improvement. Removing representation dropout costs 7.7 points (80.7%), and removing range-history encoding costs 4.5 points (83.9%). The history result is consistent with the role assigned to history in Section 4.4: the corrupt-mode samples can only be fit by cross-checking the future field against present and recent observations, so history supports the dropout mechanism as well as providing motion cues.

The ablation without dropout (80.7%) is markedly better than the main-table variant “present + future (no dropout)” (75.1%). The two are not the same configuration: the ablation removes dropout from the full model and retains every other component, whereas the main-table variant represents the plain future-conditioned imitation baseline. The recorded results do not by themselves identify which of the remaining differences accounts for the 5.6-point gap; the interaction between history

Table 2: Main comparison on 10 unseen cluttered scenes (simulated ANYmal, Isaac Lab).

Method	Success rate (%)	Collision-free distance (km per collision)	Oracle-parity ratio (vs ground-truth future range)	Decision latency p95 (ms)
Imitation, present range only	76.1	3.8	0.72	20.9
Imitation, present + future (no dropout)	75.1	3.9	0.77	21.3
Rule-based reactive baseline	74.3	3.6	0.74	20.7
Ours	88.4	4.2	0.81	18.7

Table 3: Ablation of the full method on the unseen scenes; each row removes one component.

Variant	Success rate (%)
Ours (future-conditioned + dropout)	88.4
w/o future-range conditioning	75.9
w/o representation dropout	80.7
w/o range-history encoding	83.9

encoding and dropout is the most plausible candidate, and a factorial ablation would be required to confirm it.

5.5 Failure cases

Dense low clutter. Unseen scene 7 contains dense, low obstacles. Here present-range-only imitation collides 2.1 times per km, compared with 0.5 times per km for the full method, a factor of 4.2. In collision-free-distance terms this is roughly 0.48 versus 2.0 km per collision, both well below the aggregate figures in Table 2, so scene 7 is difficult for every method. The gap is nevertheless the largest recorded, and it is where future conditioning matters most: in dense clutter the present scan shows many nearby returns but does not reveal which of them lie on the robot’s path over the next second, which is exactly what the control-aligned field encodes. Even with conditioning the policy still collides about once every 2 km in this scene, so dense low clutter remains an open case.

Predictor errors. In 5% of trials the predictor is wrong by more than 1.5 m. In these trials the dropout-trained policy reverts to present-range behavior instead of extrapolating from the incorrect field. This is the behavior targeted by the degradation property in Section 3 and is a direct consequence of the full-mask and corrupt modes. It also bounds what the method can deliver: in these trials the policy performs at best like a present-range policy, so improving the predictor in the regimes where it fails is a direct route to further gains.

Demonstrator bias. Operator 2 teleoperated with aggressive turns. This style transfers worst: on that operator’s distribution the oracle-parity ratio is 0.74, against 0.83 overall in the same per-operator breakdown. The per-operator aggregate of 0.83 was computed separately from the unseen-scene figure of 0.81 in Table 2, and the recorded data do not reconcile the two aggregations; both are reported as recorded. The qualitative conclusion does not depend on this: imitation reproduces the demonstrator’s control style. A plausible mechanism, not isolated by the recorded data, is that aggressive turning reduces the value of a field computed under the constant-command proposal of Eq. (6), since that proposal is least accurate when the command changes quickly. This mirrors the sensitivity to demonstrator composition reported for manipulation [32].

06 Limitations

Simulation only. All results come from a simulated ANYmal in Isaac Lab. No physical ANYmal trials are reported here, and the sim-to-real behavior of both the predictor and the dropout fallback is untested. Simulated range data lack the dropouts, multipath returns, and vegetation of real LiDAR, which are likely to increase the 5% rate of large predictor errors.

Statistical reporting. Results are aggregates over 5 seeds and 10 unseen scenes without recorded dispersion, so small differences, such as 76.1% versus 75.9% or 3.8 versus 3.9 km per collision, cannot be distinguished from noise. The large effects (a 12.3-point success gain, a 12.5-point ablation drop) are well beyond such differences, but confidence intervals should accompany any future version.

Demonstrator bias. The policy inherits demonstrator style, as the Operator 2 result shows. With only three operators, the study cannot separate style effects from operator skill, and weighting or filtering demonstrations by style was not explored.

Proposal and fallback design. The constant-command proposal is causal and leak-free, but it is a poor description of the robot’s future motion during sharp maneuvers, which is plausibly part of the Operator 2 effect. The fallback is conservative by construction and forgoes correct predictions that contradict the present scan. Finally, the predictor is frozen; joint training of predictor and policy could reduce the 0.19 parity gap but would reintroduce the risk of the policy co-adapting to predictor idiosyncrasies.

Scope of metrics. The evaluation covers success, collision-free distance, SPL-based parity, and latency. It does not include completion-time measures, dynamic obstacles, or rough terrain, all of which are relevant for a legged platform.

07 Conclusion

This paper studied how to condition an imitation-learned quadruped navigation policy on a predicted, control-aligned future range field of the FutureRay type. Conditioning alone does not help: without safeguards the policy reaches 75.1% success, no better than present-range imitation at 76.1%. With range-representation dropout, which masks or corrupts the future input during training, the policy reaches 88.4% success on unseen cluttered scenes, raises collision-free distance from 3.8 to 4.2 km per collision, attains an oracle-parity ratio of 0.81 relative to a policy trained on ground-truth future range, and reverts to present-range behavior in the 5% of trials in which the predictor is wrong by more than 1.5 m. The ablations attribute the largest share of the improvement to future conditioning and a substantial share to dropout. The principal open steps are validation on physical ANYmal hardware, reporting of dispersion across seeds, and mitigation of demonstrator bias in the teleoperation data.

References

Table 4: Recorded failure cases and their interpretation.

Case	Recorded observation	Interpretation
Unseen scene 7 (dense low clutter)	Present-range-only imitation: 2.1 collisions per km; ours: 0.5 collisions per km	The largest per-scene gap; consistent with future conditioning being the source of the improvement
Predictor error above 1.5 m (5% of trials)	Dropout-trained policy degrades to present-range behavior instead of extrapolating	Intended fallback of Section 4.6
Operator 2 (aggressive turns)	Oracle-parity ratio 0.74 on this operator's distribution vs 0.83 overall	Demonstrator bias inherited through imitation

- [1] T. Zang, S. Wang, H. Huang, and S. Cheng. FutureRay: Control-Aligned Future Range for Agile Quadruped Navigation. arXiv:2608.01197, 2026.
- [2] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger. ANYmal - A Highly Mobile and Dynamic Quadrupedal Robot. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.
- [3] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning Agile and Dynamic Motor Skills for Legged Robots. Science Robotics, 4(26), 2019. arXiv:1901.08652.
- [4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. Science Robotics, 5(47), 2020. arXiv:2010.11251.
- [5] 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. arXiv:2201.08117.
- [6] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. Conference on Robot Learning (CoRL), 2021. arXiv:2109.11978.
- [7] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. Robotics: Science and Systems (RSS), 2021. arXiv:2107.04034.
- [8] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. Conference on Robot Learning (CoRL), 2022. arXiv:2211.07638.
- [9] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot Parkour Learning. Conference on Robot Learning (CoRL), 2023. arXiv:2309.05665.
- [10] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. IEEE International Conference on Robotics and Automation (ICRA), 2024. arXiv:2309.14341.
- [11] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022. arXiv:2209.12827.
- [12] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. IEEE Robotics and Automation Letters, 6(3), 2021. arXiv:2103.04351.
- [13] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. Science Robotics, 9(88), 2024. arXiv:2306.14874.
- [14] L. Wellhausen and M. Hutter. ArtPlanner: Robust Legged Robot Navigation in the Field. Field Robotics, 3, 2023. arXiv:2303.01420.
- [15] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast Traversability Estimation for Wild Visual Navigation. Robotics: Science and Systems (RSS), 2023. arXiv:2305.08510.
- [16] J. Truong, D. Yarats, T. Li, F. Meier, S. Chernova, D. Batra, and A. Rai. Learning Navigation Skills for Legged Robots with Learned Robot Embodiments. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021. arXiv:2011.06188.
- [17] S. Kareer, N. Yokoyama, D. Batra, S. Ha, and J. Truong. ViNL: Visual Navigation and Locomotion Over Obstacles. IEEE International Conference on Robotics and Automation (ICRA), 2023. arXiv:2210.14791.
- [18] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter. Learning Robust Autonomous Navigation and Locomotion for Wheeled-Legged Robots. Science Robotics, 9(89), 2024. arXiv:2405.01792.
- [19] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. The International Journal of Robotics Research, 5(1):90-98, 1986.
- [20] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. IEEE Robotics and Automation Magazine, 4(1):23-33, 1997.
- [21] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena. From Perception to Decision: A Data-Driven Approach to End-to-End Motion Planning for Autonomous Ground Robots. IEEE International Conference on Robotics and Automation (ICRA), 2017. arXiv:1609.07910.
- [22] L. Tai, G. Paolo, and M. Liu. Virtual-to-Real Deep Reinforcement Learning: Continuous Control of Mobile Robots for Mapless Navigation. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017. arXiv:1703.00420.
- [23] G. Kahn, A. Villafior, B. Ding, P. Abbeel, and S. Levine. Self-Supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation. IEEE International Conference on Robotics and Automation (ICRA), 2018. arXiv:1709.10489.
- [24] G. Kahn, P. Abbeel, and S. Levine. BADGR: An Autonomous Self-Supervised Learning-Based Navigation System. IEEE Robotics and Automation Letters, 6(2), 2021. arXiv:2002.05700.
- [25] C. Finn and S. Levine. Deep Visual Foresight for Planning Robot Motion. IEEE International Conference on Robotics and Automation (ICRA), 2017. arXiv:1610.00696.
- [26] S. Ross, G. J. Gordon, and J. A. Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. International Conference on Artificial Intelligence and Statistics (AISTATS), 2011. arXiv:1011.0686.
- [27] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to End Learning for Self-Driving Cars. arXiv:1604.07316, 2016.
- [28] F. Codevilla, M. Mueller, A. Lopez, V. Koltun, and A. Dosovitskiy. End-to-End Driving via Conditional Imitation Learning. IEEE International Conference on Robotics and Automation (ICRA), 2018. arXiv:1710.02410.

- [29] F. Codevilla, E. Santana, A. M. Lopez, and A. Gaidon. Exploring the Limitations of Behavior Cloning for Autonomous Driving. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. arXiv:1904.08980.
- [30] P. de Haan, D. Jayaraman, and S. Levine. Causal Confusion in Imitation Learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. arXiv:1905.11979.
- [31] M. Bansal, A. Krizhevsky, and A. Ogale. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. *Robotics: Science and Systems (RSS)*, 2019. arXiv:1812.03079.
- [32] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarini, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martin-Martin. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. *Conference on Robot Learning (CoRL)*, 2021. arXiv:2108.03298.
- [33] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. *Robotics: Science and Systems (RSS)*, 2023. arXiv:2304.13705.
- [34] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *Robotics: Science and Systems (RSS)*, 2023. arXiv:2303.04137.
- [35] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. arXiv:2210.03370, 2022.
- [36] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A Foundation Model for Visual Navigation. arXiv:2306.14846, 2023.
- [37] A. Sridhar, D. Shah, C. Glossop, and S. Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. arXiv:2310.07896, 2023.
- [38] D. Chen, B. Zhou, V. Koltun, and P. Kraehenbuehl. Learning by Cheating. *Conference on Robot Learning (CoRL)*, 2019. arXiv:1912.12294.
- [39] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. arXiv:1207.0580, 2012.
- [40] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. ModDrop: Adaptive Multi-Modal Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8), 2016. arXiv:1501.00102.
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1706.03762.
- [42] V. Makovychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac Gym: High Performance GPU-Based Physics Simulation for Robot Learning. arXiv:2108.10470, 2021.
- [43] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. *IEEE Robotics and Automation Letters*, 8(6), 2023. arXiv:2301.04195.
- [44] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir. On Evaluation of Embodied Navigation Agents. arXiv:1807.06757, 2018.