OGMP: Oracle Guided Multimodal Policies for Agile and Versatile Robot Control

Author Names Omitted for Anonymous Review. Paper-ID [102]

***Abstract*—Amidst task-specific learning-based control synthesis frameworks that achieve impressive empirical results, a unified framework that systematically constructs an optimal policy for sufficiently solving a general notion of a task is absent. Hence, we propose a theoretical framework for a task-centered control synthesis leveraging two critical ideas: 1) oracle-guided policy optimization for the non-limiting integration of sub-optimal task-based priors to guide the policy optimization and 2) task-vital multimodality to break down solving a task into executing a sequence of behavioral modes. The proposed approach results in highly agile parkour and diving on a 16-DoF dynamic bipedal robot. The obtained policy advances indefinitely on a track, performing leaps and jumps of varying lengths and heights for the parkour task. Corresponding to the dive task, the policy demonstrates front, back, and side flips from various initial heights. Finally, we introduce a novel latent mode space reachability analysis to study our policies' versatility and generalization by computing a feasible mode set function through which we certify a set of failure-free modes for our policy to perform at any given state.**

I. Introduction

Deep Reinforcement Learning (RL) as an optimizer has shown remarkable empirical success in synthesizing control policies for hybrid underactuated systems like legged robots [19]. In particular, with inherently stable quadrupedal robots, leading to extreme parkour [3, 20], agile [17] and robust [14] locomotion capabilities. Irrespective of the task at hand, the above-methods that learn from scratch follow a similar philosophy: 1) define an exhaustive observation space, 2) engineer task-specific rewards and/or curriculum, 3) perform policy distillation, and 4) extensively randomize. Despite the compelling empirical results, such an approach does not conceptually address the underlying shortcomings of applied Deep RL, like finite exploration, reward design, susceptibility to local optima, etc, while only tackling the manifested performance limitations. For instance, policy distillation deals with the finite exploration problem by first training a teacher policy [3] with privileged information or a library of policies [20] with the hope of effective exploration and then distilling them into a single policy. As a result, performing multiple subsequent trainings, only made tractable by hardware-accelerated simulators, to increase the amount of exploration as opposed to its quality and thereby the search. On the other hand, rigorous reward-shaping on a battery of heterogeneous terms is performed to meet the intended performance, which is robot and task-specific [17, 14], as even a marginally subpar combination results in the policy converging to an undesirable local optimum. Addressing this sensitivity of the optimization is unintuitive and cumbersome as the reward terms vary in scale and nature, making them seldom transferable. Furthermore, such approaches are end-to-end task-specific solutions as opposed to being a unified approach for solving a general notion of a task as in the case of Gradient and Model-based Optimal Control (GMOC) frameworks [16, 5]. Thus, developing a theoretic framework for a unified task-agnostic control synthesis by addressing finite exploration, reward design, and behavioral multimodality shall be the focus of this work.

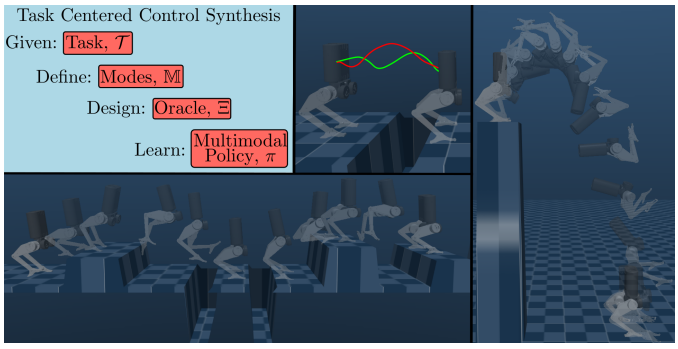

Fig. 1: Conceptual overview of the proposed synthesis approach (top left), oracle guidance in green vs. actual motion in red (top center), a frontflip dive from a 2m high platform (right), and agile parkour on a track with gaps and blocks of arbitrary dimensions (bottom left)

A popular alternative is imitation learning (IL) over demonstrations, having a unified imitation reward. With dynamically consistent demonstrations[1] that solve an intended task, IL is a reliable strategy as seen in manipulation [4]. In locomotion, we typically have dynamically inconsistent demonstrations that partially solve a task, challenging the direct application of IL. For instance, to parkour with a bipedal robot, we may have demonstrations for runs, leaps, and jumps of varying intensities from 1) motion capture on humans or 2) a GMOC framework. Nonetheless, they suffer from source-target mismatches due to morphologically dissimilar experts [2] and simplifications made to the model, cost, and constraints [6, 10, 9] respectively. Strictly imitating discrepant demonstrations can limit the policy from leveraging the full potential of the target system, inducing a sub-optimality. Moreover, naive imitation of such partial-task-solving demonstrations does not guarantee to solve the overarching task, requiring RL to train another high-level task-solving policy [2, 8]. Alongside demonstrations, task-based priors can include intuitive heuristics with design biases in providing an ansatz to the solution. Therefore, a general way to model such priors would be as oracles that

[1] through proficient human teleoperation

provide that ansatz when queried. An oracle-guided policy optimization can then be performed with the critical insight: use oracle's ansatz as a seed point for the search. Constraining the policy's search space around the local neighborhood of an oracle's ansatz results in effective exploration of a single policy optimizing a task objective of choice.

On the other hand, an essential trait to solve a high-level task is behavioral multimodality. In the context of locomotion, prior works propose learning to imitate a variety of skills [8, 11] by encoding a dataset of behaviors to a skill-embedding latent space and training a latent-conditioned policy. While they showcase a finite number of locomotion modes and transitions, generalization to fruitful variations of each mode is unexplored. For example, a locomotion task of moving from point A to point B might require us to operate in discrete locomotion modes like walk and jump, but also modulate continuous parameters like walking speed and jumping height, depending on the nature of the underlying terrain. However, mimicking a dataset with demonstrations for jumping at heights 0.1 and 0.5 m does not guarantee a policy that could jump a height of 0.2 or 0.7 m. To this end, we define a mode set and mode parameter set required for solving a task. Solving a task can then be viewed as executing a sequence of modes with a specific parameter variant chosen for each mode. We leverage the task-vital multimodality in designing an oracle, using which we train a latent-conditioned policy that generalizes to a continuous spectrum of underlying behavior parameters to solve the given task sufficiently.

Once synthesized, the reusability of such latent-conditioned policies requires understanding their limitations. The naive alternative of strapping a high-level policy outputting latent commands results in an uninterpretable pipeline [8]. Disregarding the explicit analysis, we can neither explain the approach's success nor predict an upcoming failure, which is quintessential for real robot control. Previous works validate their policy's performance through qualitative empirical results [17, 20] or skill/task-specific custom performance metrics [13, 2, 3, 18], which fail to be a standard method of analysis. In contrast, we exploit the low-dimensional latent space (or mode space) to perform a modified reachability analysis. Similar to value functions in RL that reflect a measure of optimality from a given state, we introduce a set-valued function called the Feasible Mode-set Function, which computes the set of feasible behaviors that our policy can perform from a given state. Through the proposed Latent Mode Space Reachability Analysis, we also characterize our policy's generalization in and out of the training domain.

Thus, the major contributions of our paper are threefold:

- **Oracle Guided Multimodal Policies**: A theoretical framework for task-centered control synthesis leveraging oracle-guided optimization for effective exploration and task-vital-multimodality for versatile control.
- **Experimental Validation**: Demonstration of extremely agile bipedal control tasks: parkour and dive. A single oracle-guided multimodal policy (per task) successfully performs diverse variants of distinct behaviors like pace, leap, and jump over arbitrary terrain tracks for parkour, while diving with omnidirectional flips from arbitrary initial heights.
- **Latent Mode Space Reachability Analysis**: A modified reachability analysis to quantify our policy's versatility through Feasible Mode-set Function computations

The remaining paper is structured as follows: Sec. II introduces the theoretical synthesis framework, Sec. III discusses the proposed approach, followed by Sec. IV presenting the exhaustive experimental results, analysis and ablation studies.

II. Oracle Guided Multimodal Policies

This section presents the two synergetic ideas behind our theoretical task-centered control synthesis framework: oracle-guided policy optimization and task-vital multimodality.

A. Oracle Guided Policy Optimization

Let $\mathcal{T}$ be an infinite horizon task with a task parameter set, $\psi_{\mathcal{T}} \in \Psi_{\mathcal{T}}$. Sufficiently solving $\mathcal{T}$ requires maximizing a task objective, $J_{\mathcal{T}}$ over the task parameter distribution, $p(\psi_{\mathcal{T}})$. Given the corresponding state space of interest, $x \in \mathcal{X}$ let x_t, $x[a, b]$ denote a state at time t and a state trajectory from time $t \in [a, b]$ respectively. We define Ξ to be a receding horizon oracle that provides a finite horizon state trajectory ($x^{\Xi}[t,\, t + \Delta t]$) from any given state ($x_t$) until Δt into the future for any task-variant ($\psi_{\mathcal{T}}$), such that $x^{\Xi}[t,\, t + \Delta t]$ is always within an ϵ-neighborhood of an optimal state trajectory. Formally,

$$x^{\Xi}[t,\, t + \Delta t] = \Xi(x_t, \psi_{\mathcal{T}}) \tag{1a}$$

$$\text{s.t.} \quad \exists\ x^{\Xi}[t,\, t + \Delta t] \quad \forall\, (x_t, \psi_{\mathcal{T}}) \tag{1b}$$

$$\|x_t^{\Xi} - x_t^*\|_W < \epsilon \quad \forall\, t \in [0,\, \infty) \tag{1c}$$

where ϵ is the maximum deviation bound, a constant for a given pair $(\mathcal{T}, \Xi)$, and W is a diagonal weight matrix. Note that $x^*[0, \infty]$, ϵ are unknown and are only provided for constructing the conceptual argument. Designing oracles that satisfy the above conditions is straightforward as demonstrated later in Sec. III-B.

Given the admissible control and observation spaces, $u \in \mathcal{U}$ and $o \in \mathcal{O}$, we would like to obtain the optimal policy $\pi^*(u|o)$ from a policy class Π guided by Ξ, that sufficiently solves $\mathcal{T}$. Specifically, we intend to integrate the oracle in a non-restrictive manner such that the sub-optimality of Ξ does not limit π. Since Ξ provides a reference in the state space, we propose constraining the permissible states for π to be within a ρ-neighbourhood of the oracle's guidance. Formally,

$$\pi^* := \arg\max_{\pi \in \Pi} J_{\mathcal{T}} \tag{2a}$$

$$\text{s.t.} \quad \|x_t^{\pi} - x_t^{\Xi}\|_W < \rho \quad \forall t \in [0, \infty) \tag{2b}$$

Where x^{π} represents states generated while rolling out policy π, ρ is the permissible state bound. In the context of learning, ρ directly controls the quality of oracle-guided exploration. Hence, one can now argue that

$$\begin{aligned} \|x_t^{\pi} - x_t^*\|_W &\leq \|x_t^{\pi} - x_t^{\Xi}\|_W + \|x_t^{\Xi} - x_t^*\|_W \\ &= \epsilon + \rho \qquad \forall t \in [0, \infty) \end{aligned} \tag{3}$$

Thus, by bounding the exploration of π within a ρ - neighborhood of x^{Ξ}, we have ensured that the optimal policy, π^* can still be found within the constrained search space[2] i.e. (ϵ+ρ)-neighborhood of x^* as visualized in Fig. 2.b. The choice of ρ for a given pair ($\mathcal{T}$, Ξ) is dependent on the corresponding ϵ. ϵ being maximum deviation of x^{Ξ}, an oracle with low ϵ will generate references closer to x^*. In contrast, for an oracle with high ϵ, there should be sufficient search space for π to explore around x^{Ξ} and converge to x^*. Intuitively, as $\epsilon \to \infty$, it is as if the optimization is unguided, thus needing $\rho \to \infty$, and conversely, as $\epsilon \to 0$, the search can be localized to a tight neighborhood of x^{Ξ}, implying $\rho \to 0$. Hence, we propose that optimal ρ for a given ϵ can be chosen as $\rho^* = \alpha(\epsilon)$, where α is a non-decreasing function. However, as ϵ is unknown, in practice, we perform a grid search over ρ for any given ($\mathcal{T}$, Ξ) and empirically verify the $\alpha(\epsilon)$ trend in Sec. IV.

B. Task Vital Multimodality

Any high-level task $\mathcal{T}$ can naturally be broken down into sub-tasks, which may require different modes of operation in the policy. Similar to [11], we think of these modes as time-invariant finite-horizon state trajectories representing different behaviors. Thus, W.L.O.G, we define a finite set of behavioral modes, $\mathbb{M}$, required to solve $\mathcal{T}$ sufficiently. Each mode $m \in \mathbb{M}$ in turn has a set of characteristic parameters, Ψ_m. Hence, the set of mode parameters, $\Psi_{\mathbb{M}}$, and the corresponding task parameter set, $\Psi_{\mathcal{T}}$, can now be defined as,

$$\Psi_{\mathbb{M}} := \bigcup_{i=1}^{|\mathbb{M}|} \Psi_{m_i}, \Psi_{\mathcal{T}} := \Psi_{\mathbb{M}}^N \tag{4}$$

where N is the number of horizons encountered in the task. It is worth noting that despite $\mathbb{M}$ being discrete, the corresponding parameter sets, $\Psi_m, \Psi_{\mathbb{M}}, \Psi_{\mathcal{T}}$ can all be continuous. For example, say to sufficiently solve a task like parkour ($\mathcal{T}$); we need three locomotion modes: jump, leap, and run ($\mathbb{M}$); then we should be able to run at different speeds, leap different distances and jump different heights ($\Psi_{\mathbb{M}}$) as visualized in Fig 2.a. The ideas of breaking down a task into a sequence of modes and the oracle providing guidance for a horizon Δt are complementary as modes are temporal abstractions of states. Thus, Ξ can be queried for a mode switch once every Δt ($\geq dt$, π's control period) with only the information of the mode parameter set as opposed to knowing the complete task-variant or task parameter set, i.e., $\Xi(x_t, \psi_{\mathcal{T}}) \equiv \Xi(x_t, \Psi_{\mathbb{M}})$

III. Proposed Approach

In this section, we first define two agile control tasks, followed by our specific design methodology in applying OGMP to solve them effectively, as visualized in Fig 2. Given a task, we first define the fundamental modes of operation and their corresponding parameter sets required to solve a task sufficiently (Fig 2.a). Using these defined sets, we design an oracle that can generate state trajectories as references from any given state. Spanning the mode parameter set, we use the oracle to generate a custom dataset of diverse behaviors and then train a mode encoder to construct the induced latent mode space (Fig 2.c). Finally, we train a multimodal policy conditioned on this latent mode space guided by the oracle (Fig 2.d). During policy optimization, the oracle is periodically queried to generate references online and is used to constrain the policy's search space to effectively explore (Fig 2.b). Once trained, a novel reachability analysis is performed to quantify the policy's versatility and generalization, as shown in Fig 4. The above approach is explained in detail as follows.

$\mathcal{T}: J_{\mathcal{T}}$	$m \in \mathbb{M}$	$\Psi_m \in \Psi_{\mathbb{M}}$	$p(\psi_{\mathcal{T}})$
parkour: advance forward indefinitely along the track	pace	$\{v \mid v$ is the heading speed$\}$	U($\psi_{\mathcal{T}}$)
	jump	$\{(w,h) \mid w, h$ are the jump length and height$\}$	
	leap	$\{(w,d) \mid w, d$ are the approach and leap distance$\}$	
dive: do a 360° flip and land	settle	$\{v \mid v$ is the touchdown velocity to be driven to zero$\}$	U($\psi_{\mathcal{T}}$)
	flip	$\{(r,h) \mid r, d$ are the flip direction and fall height$\}$	

TABLE I: Task Vital Multimodality for parkour and diving

A. Task Definitions

We demonstrate the above conceptual framework applied in two bipedal control tasks demanding extreme agility and versatility: parkour and diving, as explained below.

1) Parkour Task: Unlike unstructured open-world parkour, we work with an abstracted version where the underlying terrain's height (h) is varied only along the heading direction, and the robot is expected to advance along this track[3]. The parkour track is composed of an arbitrary permutation of three terrain elements: block ($h > 0$), gap ($h < 0$), and flat ($h = 0$) with random heights. While a gap is non-traversible, requiring a single leap to cross, the robot can step on to traverse the block and flat elements. Thus, to sufficiently solve this task, a policy must leap, jump, and pace over various terrain elements while seamlessly switching between these modes. The above problem can now be encoded in our conceptual framework as shown in table I

2) Dive Task: In this task, the policy is expected to dive down a block of arbitrary height (h) and land on the ground, doing a 360° flip along the roll or pitch DoF. The motion can be split into three phases: take-off, flight, and landing. In the flight phase, the angular momentum is conserved; only allowing minimal changes to the angular velocity by morphing the robot's inertia. Therefore, it is crucial for the policy to reach the required initial angular velocity to complete a rotation by the end of the take-off phase. Given this coupling effect, we combine these two phases into a single mode: flip, which, depending on the direction, can have 4 variants: front, back, left, and right. Upon completing a successful flip, the policy has to land and transition to settle mode, which is to remain in the nominal pose perpetually as defined in table I.

Evident from the above two tasks, $J_{\mathcal{T}}$ is task-dependent and could be hard to design. In the absence of a reasonable $J_{\mathcal{T}}$, a

[2] $\rho \geq \epsilon$ is assumed to be trivially satisfied

[3] This simplification is only to conduct a systematic behavior analysis and is not limited by the proposed approach

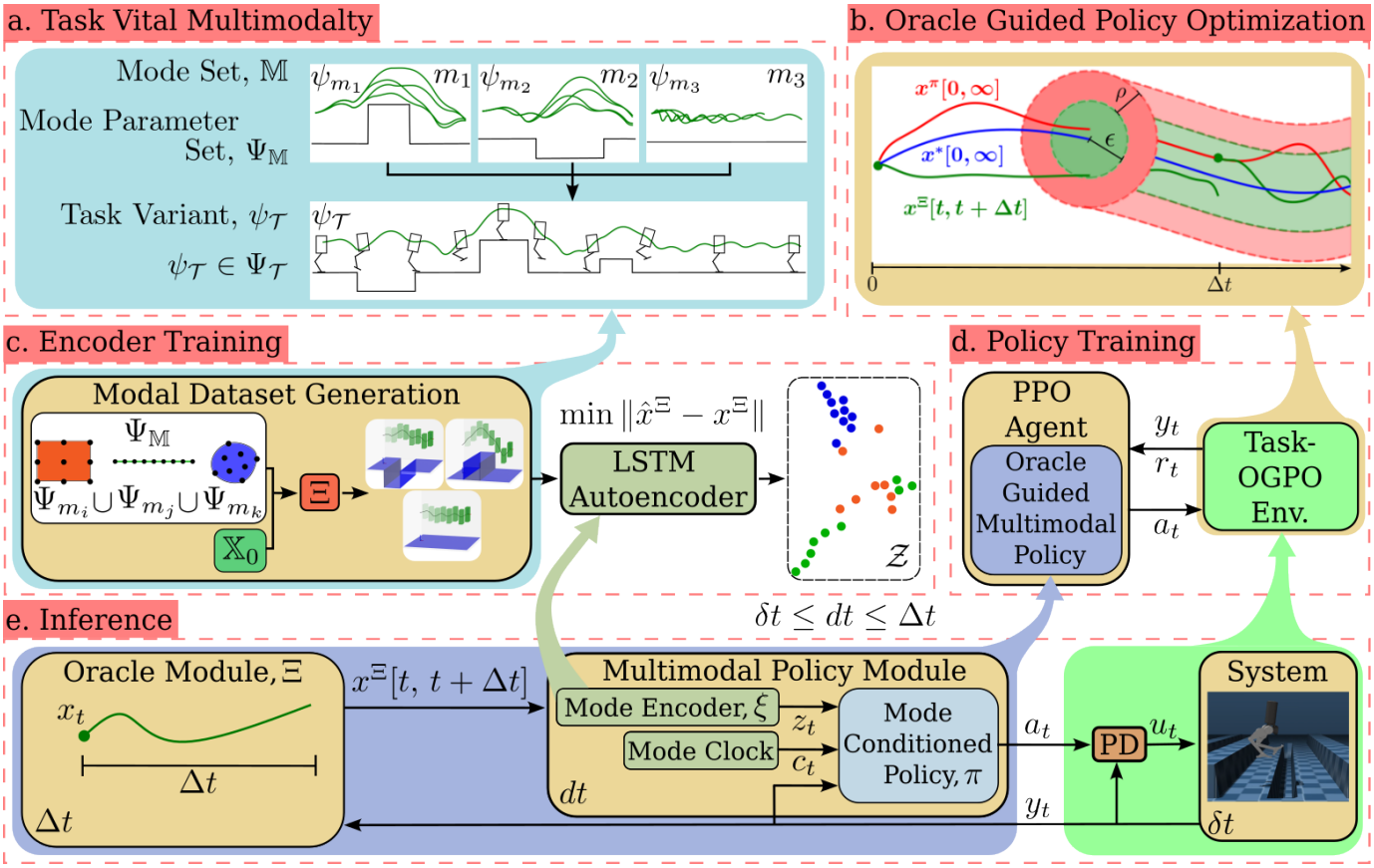

Fig. 2: Overview of OGMP: a) The breakdown of a task into its principal modes with a mode and mode parameter set b) Guided exploration by constraining the search space around the local neighborhood of the oracle's reference c) Mode encoder: an LSTM autoencoder trained on a custom modal dataset by minimizing reconstruction loss d) Multimodal policy trained with oracle guided policy optimization on a task environment e) The closed-loop inference pipeline with the high-level oracle and the low-level multimodal policy

compelling alternative would be to "track" the oracle as they provide ϵ-neighbourhood reference to a task-solving optimal solution. Hence, we propose the surrogate objective

$$\tilde{J}_{\mathcal{T}} := \sum_{t=0}^{\infty} \|x_t^{\pi} - x_t^{\Xi}\| \tag{5}$$

Thus, minimizing $\tilde{J}_{\mathcal{T}}$ subject to the dynamics and control constraints leads to a feasible motion that is as close to the reference as possible. Strictly minimizing $\tilde{J}_{\mathcal{T}}$ within the oracle-constrained search space can cause Ξ to limit π; however, in practice, choosing an optimal ρ and long enough Δt causes $\pi^*_{\tilde{J}_{\mathcal{T}}} \rightarrow \pi^*_{J_{\mathcal{T}}}$[4] as verified in Sec IV-C. For tasks like dive, where finding a reasonable $J_{\mathcal{T}}$ is nontrivial, we show that the task-independent $\tilde{J}_{\mathcal{T}}$ is a good bet in practice. We study the empirical validity of using $\tilde{J}_{\mathcal{T}}$ in Sec. IV and a per-step reward for an equivalent maximization objective in Sec. III-B.

B. Design Methodology

For the above-defined tasks, we discuss the specific design choices made to OGMP. Fig 2 provides a visual overview of the training (Fig 2.c and d) and inference (Fig 2.e) pipelines explained below.

[4] $\pi^*_J :=$ optimal policy w.r.t objective, J

1) Oracle Module: For any given task, a simple strategy for an oracle would be to linearly interpolate the relevant state variables from the initial to the desired goal states. In parkour, since the overarching agenda is to advance along the track, the linear interpolation would be from the current to a target position along the track while adapting the robot's height for blocks (height increase for a jump) and gaps (ramp-up and down for a leap) as shown in Fig. 3 (left). For the dive, the oracle can linearly interpolate the base height and corresponding rotational DoF from 0° to 360°. Once we obtain a position trajectory, the velocity can be the finite difference. Naming this oracle as Ξ_{li} (Fig. 3), one can notice its high ϵ as the generated state-trajectories do not consider the system's inertia and gravity. To this end, considering a simplified single rigid body model as in [12], the dynamics in the world coordinates are given by:

$$m(\ddot{p} + g) = \sum_{i=1}^{2} f_i \tag{6}$$

$$\frac{d}{dt}(I\omega) \approx I\dot{\omega} = \sum_{i=1}^{2}(r_i \times f_i + \tau_i) = \sum_{1}^{2} \bar{\tau}_i \tag{7}$$

where $\ddot{p}$, ω is the robot COM acceleration and the angular velocity, r_i, f_i, τ_i are the position, force and moment vectors

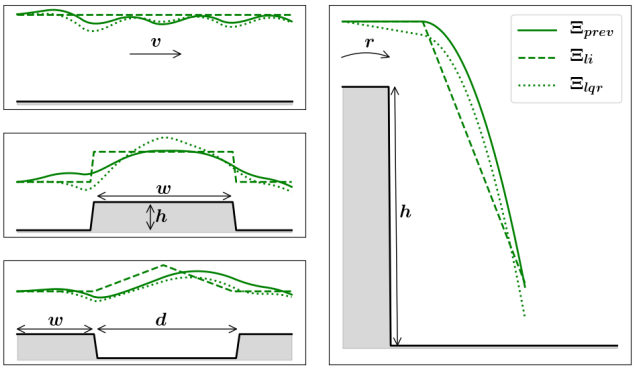

Fig. 3: Reference trajectories from different oracles for the modes of parkour (left) and diving (right) with the parameters in table I

from the i^{th} contact point and m, I, g are the mass, moment of inertia and gravity. Typically, r_i is from a predefined contact schedule, leading to time-varying dynamics. Since, by definition, an oracle need not provide realistic control, we define an auxiliary control $\bar{\tau}_i$, encompassing the overall moment, which in turn makes the dynamics time-invariant. Additionally, the rotation and rotation rate matrices are made constant by considering the average reference orientation over a horizon. Upon these approximations to Eq. 7 and discretization leads to a linear time-invariant (LTI) system over the current horizon:

$$x_{t+1} = Ax_t + Bu_t,\ y_{t+1} = Cx_t,\ u_t = [f_1, f_2, \bar{\tau}_1, \bar{\tau}_2]^T \quad (8)$$

where $x_t \in \mathbb{R}^{13}$, y_t, $u_t \in \mathbb{R}^{12}$ are the gravity-augmented state, relevant output, and control vectors. Thus, a new oracle can be constructed considering two distinct phases: flight and contact. In flight, $u_t = 0$ as there are no contacts, and during contact, an optimizer of choice can be used to compute the optimal control for a given objective, $u_t = u_t^*$. Applying the corresponding control and forward simulating Eq. 8, the state trajectory is obtained. Given an LTI model and the absence of constraints, we use the analytic solution of preview control[15] as opposed to sophisticated numerical optimizers, thereby allowing long-horizon computations. As a desirable attribute for an oracle, preview control incorporates look-ahead behavior with a previewable demand term in its control law as opposed to LQR. Thus, using Ξ_{li} as the reference for a tracking objective, optimizing with preview control and LQR results in oracles, Ξ_{prev} and Ξ_{lqr} respectively, having a ϵ smaller than Ξ_{li}, as they capture the dominant dynamics of the hybrid system as shown in Fig 3.

2) Multimodal Policy Module: We propose a multimodal policy class (Fig 2.e) comprising a mode encoder and a mode-conditioned policy trained subsequently, as explained below and shown in Fig 2.c and d.

Mode Encoder: The motivation for encoding modes is to obtain a compact representation space for conditioning our policy, for which we first train the mode encoder, ξ, on a dataset of diverse locomotion modes. Since modes are relative-state time-invariant state trajectories, we require the ξ to be a map from a trajectory space into a latent space, i.e

$$z = \xi(x[t,\, t+\Delta t]), \quad \dim(z) \ll \frac{\Delta t}{dt}\dim(\mathcal{X}) \quad (9)$$

As opposed to using a fixed and possibly unbalanced dataset of motion capture from a dynamically different source, given the oracle (Ξ), a mode parameter set ($\Psi_{\mathbb{M}}$), and a set of initial states ($\mathbb{X}_0$), we can generate a well-balanced dataset of any desired size containing sufficient samples of each mode, prior to the policy training. Since $\Psi_{\mathbb{M}}$ is a closed set, we discretize it, uniformly spanning all possible parameter variations for each mode, generating a rich and balanced dataset. 9 variants per mode, $\dim(z)$ = 2, and a 32 neurons single hidden layer LSTM auto-encoder where chosen for training ξ. Minimizing a reconstruction loss on the custom dataset generates a set of latent mode points as visualized in Fig. 2.c. We observe a structured clustering, which hints at the existence of an underlying regularized latent mode space $\mathcal{Z}$, which is explored later in Sec. III-C and IV-B.

Mode Conditioned Policy: A stationary policy mapping from the observation space, $o_t \in \mathcal{O}$ to the admissible controls space, $u_t \in \mathcal{U}$ designed as follows,

$$\begin{aligned} u_t &:= K_p \cdot (a_t - q_t) - K_d \cdot \dot{q}_t \\ a_t &\sim \pi(a_t|o_t) \end{aligned} \quad (10)$$

where q_t, $\dot{q}_t$, K_p, K_d are the motor positions, velocities, proportional and derivative gain matrices. The defined action space, $a_t \in \mathcal{A}$, can be interpreted as motors' PD setpoints. With the inherent partial observability of the system, the choice of observation space becomes crucial. To this end, the observation space should provide feedback from the system and the desired mode command. Hence, we choose $o_t = [y_t,\, z_t,\, c_t]$, where y_t is the measurable output of the system consisting of base height, orientation, velocity, joint states, and optionally a local terrain map in front of the robot, z_t is the latent mode command and the clock signal $c_t = [\sin(\phi), \cos(\phi)]$ is defined on a phase variable, ϕ, which runs from 0 to 1, from the start to end of a given mode. For the reward function, we define

$$\begin{aligned} r_t &:= r_{\text{track}} + r_{\text{regulation}} \\ r_{\text{regulation}} &:= 0.3 \cdot \mathbb{1}(\text{if non-toe contact}) - 0.05e^{-0.01\|u_t\|} \\ r_{\text{track}} &:= 0.475e^{-5\|er_p\|} + 0.475e^{-5\|er_o\|} \end{aligned} \quad (11)$$

where er_p, er_o are the errors in base position and orientations and r_t is the per-step reward term of an equivalent maximization objective as $\tilde{J}_{\mathcal{T}}$. The proposed oracle-guide constraint (Sec. II) is programmed as a termination condition. To further improve performance, the episode is also terminated when r_t is below a threshold. The full termination condition is,

$$\text{terminate episode} := \mathbb{1}(\|x_t^{\pi} - x_t^{\Xi}\|_W > \rho) \vee \mathbb{1}(r_t < 0.5) \quad (12)$$

We analyze the choice of the ρ and its effect on performance later in Sec. IV-C. W_{11} and W_{33} are set to 1 for the parkour and dive tasks, respectively, with the remaining entries as zero. Thus, the above modules result in a random horizon POMDP for which we use off-the-shelf PPO to train a policy: 128 nodes per layer, 2 layer LSTM network, where each episode is an arbitrary variant of the task, $\psi_{\mathcal{T}}$ uniformly sampled from $\Psi_{\mathcal{T}}$.

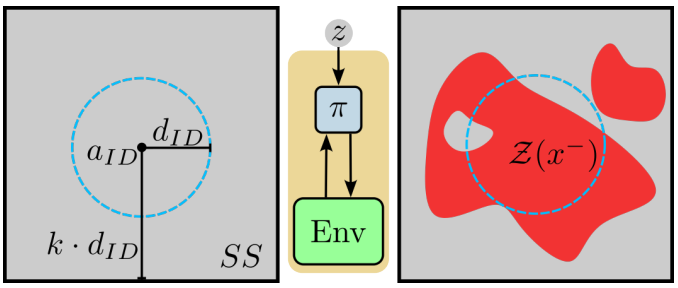

Fig. 4: Overview of LMSR Test: left) search space, SS defined based on π's estimated a_{ID} and d_{ID}. center) closed-loop system under analysis. right) the computed finite mode-set function, $\mathcal{Z}(x^-)$ in red

C. Latent Mode Space Reachability Analysis

Given the low-dimensional latent mode space ($\dim(\mathcal{Z}) = 2$), we leverage it to quantify the policy's mode diversity. Equivalently, this implies that the policy should have generalized to a significant number of points in the latent space. To this end, we perform a modified reachable set computation as discussed below. We first define $\mathcal{X}_{\text{fail}}$, a set of failure states we want the system to avoid. In the context of the legged robots, a generic notion of failure is the robot falling down.

$$\mathcal{X}_{\text{fail}} := \{x \mid \text{base height} < \text{ nominal min. base height}\} \quad (13)$$

Without the oracle and mode encoder in the pipeline, considering a closed-loop system with the multimodal policy and the environment, the latent mode command (z) is the only input as in Fig 4 (center). Given a trained π, since mode switches occur once every Δt, the discrete-time closed-loop mode-controlled dynamics for a task variant $\psi_{\mathcal{T}}$ can be written out as

$$x^+ = F_\pi(x^-, z, \psi_{\mathcal{T}}) \quad (14)$$

where x^+ is the state of the system after Δt seconds having transitioned to the mode z from the state x^-. For a given x^-, there should exist a set of modes executing which the system won't reach $\mathcal{X}_{\text{fail}}$ within Δt. Formally, we define the feasible mode-set function, $\mathcal{Z}(x)$, which maps a state to the set of all feasible modes in the latent mode space as follows,

$$\mathcal{Z}(x) := \{z \,|x^+ \notin \mathcal{X}_{\text{fail}}, x = x^-\} \quad (15)$$

Since Eq. 14 is not assumed to be known analytically, we compute $\mathcal{Z}(x)$ by explicitly rolling out the policy, which we call the Latent Mode Space Reachability (LMSR) test.

To compute $\mathcal{Z}(x)$, we first define a search space, $SS := \{z \,|\, \|z - a_{ID}\|_\infty < k \cdot d_{ID}\}$, where a_{ID} and d_{ID} are the center and radius of the circumcircle of the latent points in-domain (ID) of the training as shown in Fig 4 (left). Intuitively, one can expect a policy to be successful when commanded from $ID := \{z \,|\, \|z - a_{ID}\| < d_{ID}\}$ as it is within the training domain. ID, however, is a conservative estimate of the modes that the policy might have learned; hence, to account the out-of-domain(OD) mode generalization of π, we include points $OD := \{z \,|\, \|z - a_{ID}\| \geq d_{ID}, z \in SS\}$ by choosing $k > 1$. Thus, an exhaustive $\mathcal{Z}(x)$ is computed from a given x and $\psi_{\mathcal{T}}$ within SS by rolling out F_π in simulation as visualized in Fig 4 (right). Since the defined SS is continuous, we discretize it as a grid of 30 points per axis, leading to 900 points in SS. Once $\mathcal{Z}(x)$ is computed, a direct measure of the diversity would be $|\mathcal{Z}(x)|$, as it is the number of feasible modes that π could execute from x. To quantify the generalization of π in a domain $\mathcal{D}$ we define a probability measure $\nu_{\mathcal{D}}$

$$\nu_{\mathcal{D}}(\mathcal{Z}(x)) := \frac{\text{no. of points in } (\mathcal{Z}(x) \cap \mathcal{D})}{\text{no. of points in } (\mathcal{D})} \quad (16)$$

where $\nu_{\mathcal{D}} \in [0, 1]$ measures the probability of success when a point z is chosen from a domain $\mathcal{D}$ at a given $(x, \psi_{\mathcal{T}})$. Taking $\mathcal{D} = ID, OD$ we measure the in-domain and out-of-domain generalization of our policies in Sec. III-C.

IV. Results

This section contains the experimental results of OGMPs trained and validated on the MuJoCo simulator. The overall performance metrics quantifying our approach's agility and versatility are presented first, followed by a hyperparameter analysis of the design choices.

The twofold motivation of our approach was extreme agility and mode versatility to synthesize policies that solve fruitful overarching tasks. To this end, a single multimodal policy is trained per task, as π_{parkour} and π_{dive} for parkour and dive, respectively. The supplementary video and Fig 1 showcase π_{parkour} successfully traversing challenging parkour tracks with blocks and gaps of arbitrary dimensions placed at random intervals indefinitely. The seamless transitions between different leap lengths and jump heights further highlight the policy's versatile agility. On the other hand, π_{dive} performs successful front, back, left, and right flips from various initial heights and smoothly transitions to the nominal pose, landing the robot. Despite the absence of oracle reference for the actuated DoFs, π_{dive} learns to curl and extend its legs for the flip and landing, respectively. The emergent behaviors are for increasing the angle velocity in flight while landing with a minimum-impact touchdown. For quantitative benchmarking, we report the expectation of different performance metrics over the task parameter distribution $p(\Psi_{\mathcal{T}})$, approximated by the sample mean measured over n episode rollouts. If l is a defined metric, the values in table II and Fig. 8 are computed as: $\mathbb{E}_{p(\psi_{\mathcal{T}})}[l] \approx \frac{1}{n}\sum_i^n l_i$, where l_i is the i^{th} sample. Empirically, we found $n = 100$ for every metric's mean to settle as a reliable estimate. We define a test environment with a track length of 10m with an episode length of 400 steps for the parkour and an episode length of 150 for the dive. In each case, the episode is terminated only if the episode length is reached or the robot falls down (if the terrain-relative base height < 0.3 m). The analysis on the most relevant task (i.e., either π_{parkour} or π_{dive}.) is reported for each result below.

A. Agility

While there is no one metric to quantify agility, the maximum of Heading Acceleration (M.H.A) [7], Froude number (M.F), and Heading Speed (M.H.S) [13] have been used to measure agility of legged robots. We include the average

metric, units	z_t	z_t, c_t	h_t	h_t, c_t	h_t, z_t	h_t, z_t, c_t
M.H.A, m/s^2	**4.7g**	3.5g	3.2g	3.6g	3.5g	3.1g
M.H.S (v), m/s	1.4	1.57	1.66	1.74	1.74	**1.77**
M.F. ($\frac{v^2}{g \cdot ll}$) , $-$	0.48	0.56	0.64	0.69	0.70	**0.72**
E.L., $-$	72	172	252	320	264	**336**

TABLE II: Metrics of agility for variants of π_{parkour} with different observation spaces

episode length (E.L) of the test environment, as ideally π_{parkour} is expected to advance indefinitely despite the challenging track. The above metrics are estimated for π_{parkour} with different choices of observation spaces as in table II. As shown, the synthesized policies reach accelerations of up to $4.7g$ and heading speeds of up to $1.77\ m/s$ with Froude numbers ranging from $0.48 - 0.72$. The best variant has an average episode length of 84% the maximum possible episode length, showing its efficacy. Despite π_{parkour} being unaware of the complete track, it dynamically advances along the track instead of converging to conservative quasi-static motions by trading speed for stability. Given a stringent termination condition, we notice π_{parkour} is forced to learn precise foot placements for landing and take-off, thereby not needing multiple attempts to climb a block or cross the gap, leading to highly agile motions. Due to the radically changing terrain height, π_{parkour} rapidly modulates its momentum as reflected in the high accelerations in table II, harnessing the full potential robot's actuator. On the other hand, the Fraude number aims to compare the dynamic similarity between quadrupeds and bipeds. [1] observed a switch from energy-efficient walking to agile jumping gaits in bipedalism when they have a Fraude number around 0.5, which is interestingly consistent with our results.

B. Versatility

The idea of task vital multimodality offers two interpretable sets: the mode parameter set $\Psi_{\mathbb{M}}$, and the latent mode space, $\mathcal{Z}$ to study the generalization of the policies over $p(\psi_{\mathcal{T}})$. While $\Psi_{\mathbb{M}}$ can be used to test individual locomotion modes, it does not scale for validating transitions for which we utilize $\mathcal{Z}$.

1) Mode Parameter Set Test: Since the defined $\Psi_{\mathbb{M}}$ of our training is a convex set, we dilate it and define higher test ranges for each parameter, thereby testing for both in-domain (ID) and out-of-domain (OD) generalization. Discretizing this test set, we evaluate π_{parkour} and plot the undiscounted returns obtained for blocks and gaps with varying dimensions in Fig 5. Different blocks and gaps require jumps and leaps of varying magnitudes, showcasing our policies' versatility. The training sets Ψ_{jump} and Ψ_{leap} are the regions within the boundary marked in black in each plot. Thus, π_{parkour} shows consistent performance for variants within the black boundary (ID) while also extrapolating its skills by jumping and leaping over unseen terrain variants outside the black boundary (OD) as seen in the supplementary video and Fig 5.

2) Latent Mode Space Reachability (LMSR) Tests: Two variants of the LMSR test in Sec III-C is performed on π_{parkour}: 1) Flat Ground Test: to quantify the mode-diversity from a nominal state and terrain and 2) Transition Test: to quantify the bandwidth of modes available to transition from a given

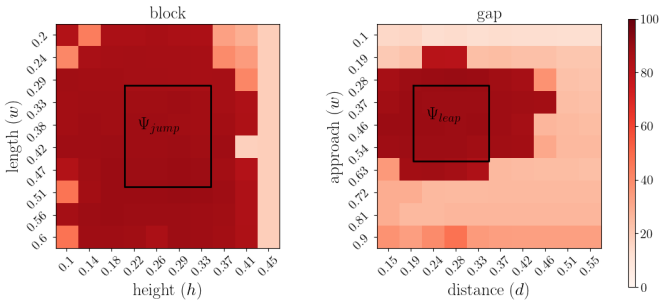

Fig. 5: Heatmap of π_{parkour}'s undiscounted returns performing jumps and leaps of varying magnitudes both in and out of $\Psi_{\mathbb{M}}$ while traversing blocks and gaps of different dimensions

state. $a_{ID} = (-0.07, 0.58)$, and $d_{ID} = 0.82$ were estimated from latent modes encountered (blue dots in Fig. 6.a) when testing π_{parkour} in-domain of $p(\psi_{\mathcal{T}})$ similar to the metrics in table II, to compute the corresponding search space.

Flat Ground LMSR Test: Taking $\psi_{\mathcal{T}}$ = flat ground and x = nominal initial state of the robot, we roll out $F_{\pi_{\text{parkour}}}$ as in Eq. 14 for one horizon, (Δt) for each z in the above-defined search space taking $k = 2$, as seen in ref Fig. 6.a. From the computed $\mathcal{Z}(x)$ in Fig 6.b, we observe that the π_{parkour} shows significant in and out-of-domain generalization. Precisely, the probability of success in ID and OD of π_{parkour} are $\nu_{ID} = 0.99$ and $\nu_{OD} = 0.49$. Furthermore, we performed K-means clustering over a dataset of state trajectories corresponding to each point in $\mathcal{Z}(x)$ (Fig. 6.e). Varying the number of clusters from $n = 2$ to 15, we note that the optimal number of clusters with the highest silhouette score corresponds to $n = 4$ consistent with the elbow plot, as seen in Fig. 6.c. This observation validates our initial design hypothesis of requiring a finite number of modes and fruitful variations of each mode to solve a task. The distinct modes and mode clusters are visualized in Fig 6 (bottom) and d. From left to right in Fig 6 (bottom), we can loosely label these modes as jump, leap, hop, and step, respectively. By varying the parameters of each of these modes, π_{parkour} gracefully performs a continuous spectrum of behaviors. Interestingly, though we defined a single mode to move forward in the parkour task, π_{parkour} performs two distinct variations of pace with different contact patterns: hopping and stepping. In the supplementary video, we see that π_{parkour} hops to traverse a strip of flat terrain in the track while steps to make minor adjustments before taking off or after landing in the leap and jump modes.

Transition LMSR Test: In an arbitrary track, $\psi_{\mathcal{T}}$ and an episode of length $2\Delta t$, we perform the second variant of the LMSR test. Starting from the same nominal state at $t = 0$, π_{parkour} follows the optimal latent mode command, z_0^* (given by the oracle), until $t = \Delta t$. At the state of the transition point, $x_{\Delta t}$, we compute $\mathcal{Z}(x = x_{\Delta t})$ to obtain the feasible modes that π_{parkour} could successfully transition to. From the computed $\mathcal{Z}(x)$ and the corresponding state trajectories in Fig. 7(center), we observe a significant number of modes-to-transition both in and out of the training domain with success probabilities, $\nu_{ID} = 1.0$ and $\nu_{OD} = 0.77$

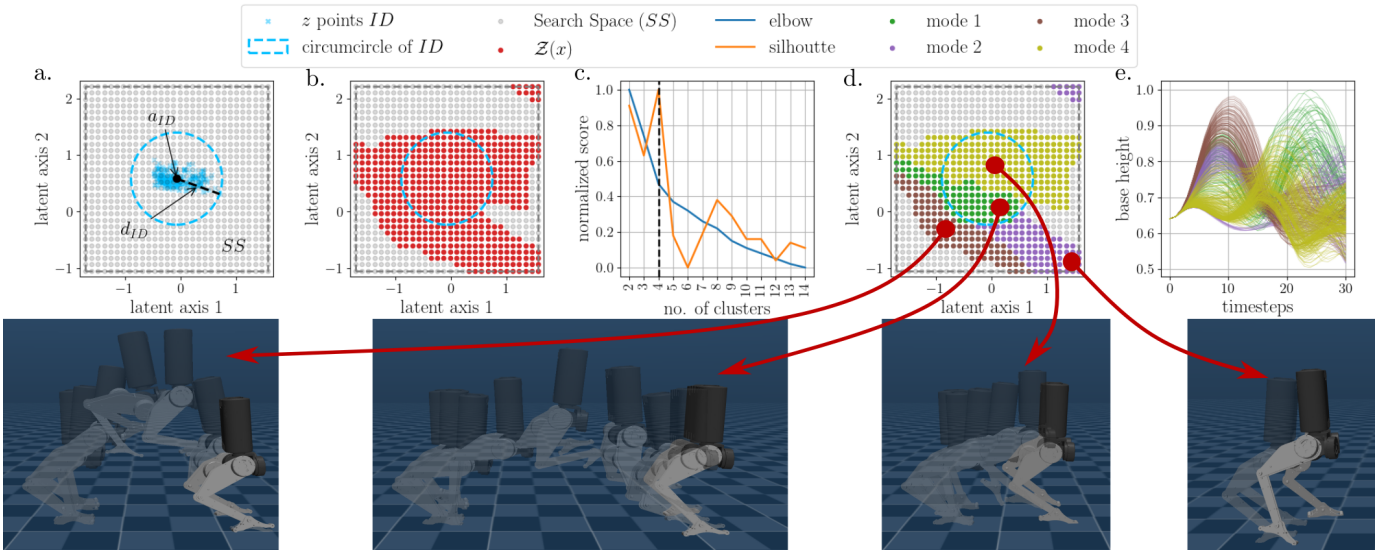

Fig. 6: Overview of the Flat ground LMSR Test: a) The defined search space SS based on a_{ID}, and d_{ID} of π_{parkour}. b) The Finite Mode Set Function ($\mathcal{Z}(x)$) computed from the nominal initial state and terrain. c) The elbow and silhouette scores for varied 'K' of K-Means clustering over the state trajectories corresponding to points in $\mathcal{Z}(x)$. d) Mode clusters corresponding to the optimal number of clusters, K=4 c) the base height trajectory corresponding to the different points in $\mathcal{Z}(x)$, colored consistent with their mode cluster.

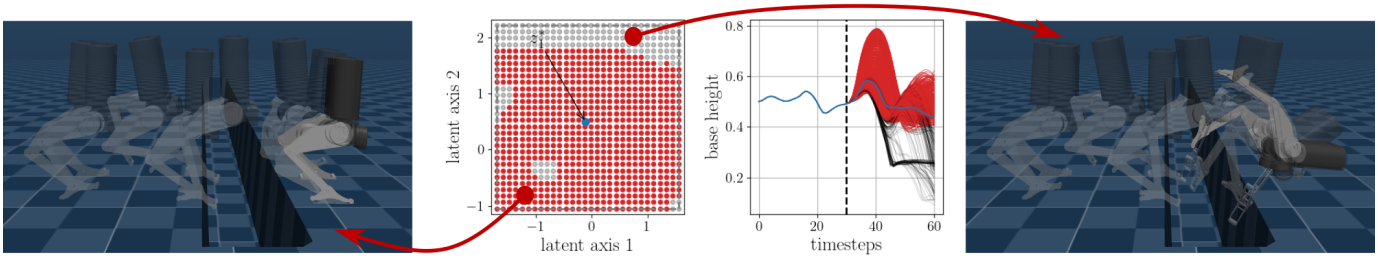

Fig. 7: Overview of the Transition LMSR Test: left) a feasible transition corresponding to a point in $\mathcal{Z}(x)$. center) the feasible mode set function at the transition state and the corresponding base height trajectories shown in red. right) an infeasible transition from a point not in $\mathcal{Z}(x)$. The optimal mode-to-transition picked by the corresponding Ξ and its trajectory is shown in blue.

respectively. Naturally, the most optimal transition as per the oracle is to mode z_1^* (shown in blue in Fig. 7) is ID. However, z_1^* is not the only feasible transition π_{parkour} is capable of performing, which proves that π_{parkour} was not limited by the corresponding Ξ and learns novel transition maneuvers through effective exploration within the $\epsilon + \rho$ neighborhood around Ξ's reference. Given the reasonable "bandwidth" of mode transitions of π_{parkour}, one can reuse these OGMPs to solve unseen tasks by designing a corresponding new oracle, Ξ_{new}. We conjecture that if Ξ_{new} selects mode commands $z \in \mathcal{Z}(x)\, \forall x$ at mode-switches, the same low-level policy can solve novel tasks (with common modes) without having to be retrained.

C. Hyperparameter Analysis

Finally, we analyze the design choices that significantly impact the performance in practice. For each parameter under scrutiny below, we compare the undiscounted returns estimated while testing. Since we use a surrogate objective $\tilde{J}_{\mathcal{T}}$, this may not be the best measure of performance. For parkour, a candidate true objective, $J_{\mathcal{T}}$, could be the displacement along the track. Thus, we also study the validity of using $\tilde{J}_{\mathcal{T}}$ and its exploitation by π_{parkour} for specific design choices.

1) Choice of observation space (o_t): Our observation space has 4 major components: robot's partial proprioceptive state ($\tilde{x}_t$), clock (c_t), latent mode command(z_t), and a local height map(h_t). While $\tilde{x}_t$ is indispensable, we compare other combinations that are blind: [z_t], [z_t, c_t] and terrain-aware: [h_t], [h_t, c_t], [z_t, h_t] and [z_t, h_t, c_t] as in table II. Variants with c_t consistently outperform their counterparts without the clock signal, as in Fig. 8.a. The blind variants tend to be myopic and aggressive with higher accelerations, as in table II, which is not ideal. Since the only terrain information held by blind π_{parkour} is through a compressed encoding (z_t), a significant loss of information makes them sub-optimal, unlike the terrain-aware variants. Once trained, a π_{parkour} without z_t can operate independently of the oracle if it has an exhaustive observation space as these policies learn to replicate Ξ during training. Therefore, an observation space with h_t is task-complete for parkour, as seen by their superior performance. In contrast, a π_{parkour} with z_t treats the oracle as a reactive planner and drives the system to the commanded mode. Thus, z_t is a

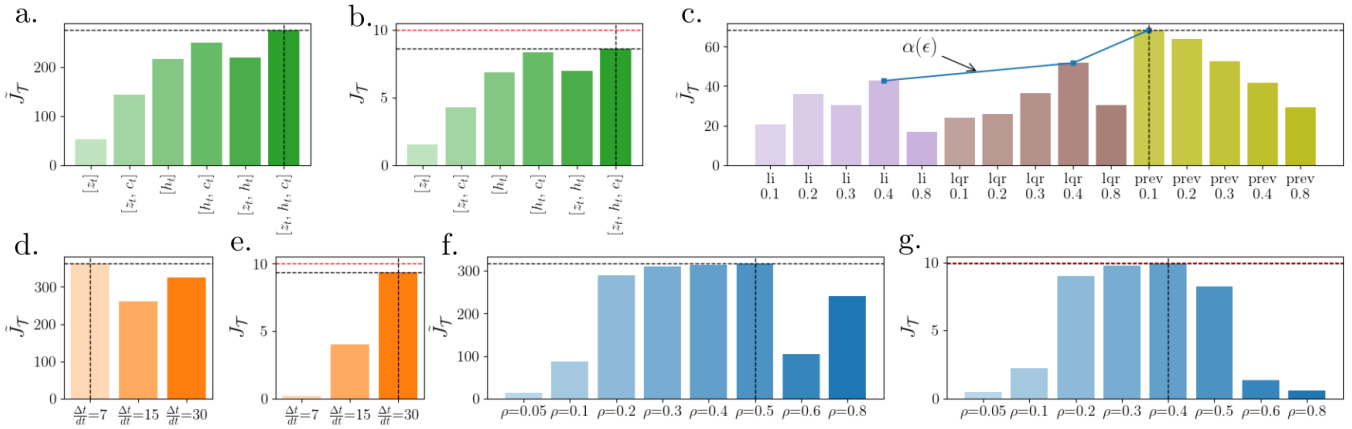

Fig. 8: The estimated surrogate ($\tilde{J}_\mathcal{T}$) and true ($J_\mathcal{T}$) objectives vs. choice of 1) observation space, o_t (a and b), 2) oracles, Ξ (c), 3) horizon length, Δt (d and e) and 4) permissible state bound, ρ (f and g). In b, e, and g the upper bound of $J_\mathcal{T}$ is marked in red.

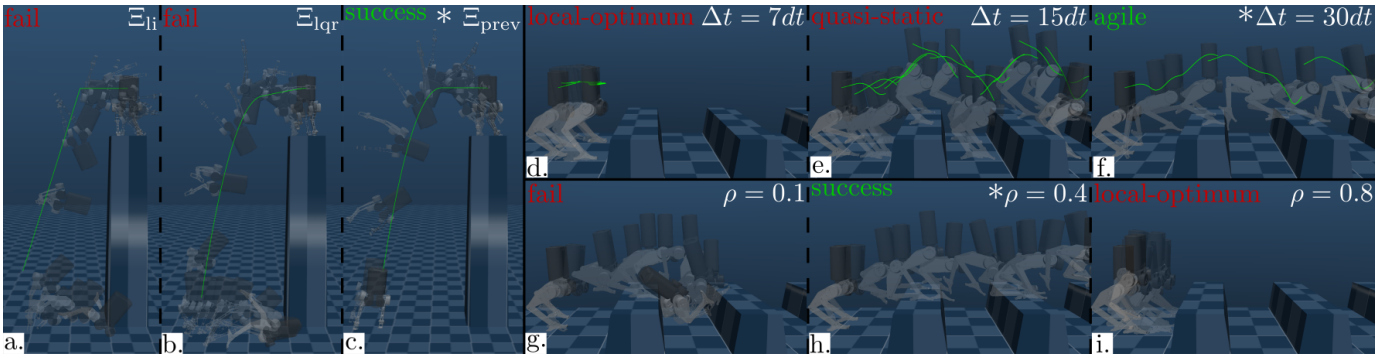

Fig. 9: Motion traces comparing the choice of 1) oracles (a, b, c) where Ξ_{li}, Ξ_{lqr} fail while only Ξ_{prev} successful in a side flip dive from a block of 2m 2) horizon length (d, e, f) where the longest horizon ($\Delta t = 30dt$) results in the desired agile performance 3) permissible state bound (g, h, i) where the tighter ($\rho = 0.1$) and lenient ($\rho = 0.8$) bounds fail

generalization of commands[5] given by a high-level planner and tracked by low-level controllers. From Fig. 8.a and b, a similar trend of $J_\mathcal{T}$ and $\tilde{J}_\mathcal{T}$, we conclude a lack of disparity caused by observation space variations.

2) Choice of oracle (Ξ): By design, the three proposed oracles: Ξ_{li}, Ξ_{lqr} and Ξ_{prev}, taken in order have a non-increasing trend of the maximum deviation bound, ϵ. From the proposed theory, since the optimal choice of the exploration bound (ρ^*) is dependent on the value of ϵ, we simultaneously vary the value of ρ from 0.1 to 0.8 for each oracle and compare the performances. For the dive task, the variants of Ξ_{prev} perform significantly better as in Fig 8.c and 9.c. Interestingly, a tighter bound ($\rho = 0.1$) performs the best for Ξ_{prev}, as Ξ_{prev} has the lowest ϵ being the closest reference to the optimal trajectory. In contrast, for Ξ_{lqr} and Ξ_{li}, higher exploratory deviation ($\rho = 0.4$) results in the best performance. Note that $\epsilon_{\text{li}} \geq \epsilon_{\text{lqr}} \geq \epsilon_{\text{prev}} \implies \rho^*_{\text{li}} = \rho^*_{\text{lqr}} > \rho^*_{\text{prev}}$ validating $\rho^* = \alpha(\epsilon)$, where $\alpha()$ is observed to be a non-decreasing function as in Fig. 8.c.

3) Choice of oracle's horizon (Δt) : Based on the oracle's horizon length (Δt), a policy could be either myopic or long-horizon optimal due to which we compare $\Delta t = 7dt$, $15dt$, and $30dt$. For parkour, $\Delta t = 7dt$ variant converges to a local optimum (Fig. 9.d), exploiting the high replanning frequency by effectively tracking the short-horizon reference standing in place, having the highest $\tilde{J}_\mathcal{T}$ and yet the lowest $J_\mathcal{T}$ as in Fig. 8.d and e. While the variant $\Delta t = 15dt$ advances along the track, it tends to be quasi-static. As shown in Fig. 9.e, on a block-flat-block terrain, the policy, unaware of the farther second block, descends the first block and jumps back up, which is a sub-optimal behavior. In contrast, with a longer horizon ($\Delta t = 30dt$), π_{parkour} leaps directly from block to block (Fig. 9.f), which is the optimal solution as it leverages multimodality, leading to agile motion as intended. From Fig 8.d and e, we observe that the choice of Δt can significantly deviate the surrogate objective from the true objective as being myopic improves the scope of imitation, conversely increasing Δt takes $\tilde{J}_\mathcal{T}$ closer to the underlying task-objective $J_\mathcal{T}$.

4) Choice of permissible state bound (ρ) : To study the effect of varying ρ for a given ϵ taking Ξ_{prev} for parkour we vary ρ from 0.05 to 0.8 as seen in Fig. 8.f and g. With $\tilde{J}_\mathcal{T}$ note that the $\rho^* = 0.5$ and moving away in either direction decreases the performance. For $\rho < \rho^*$ (Fig 9.g) the optimal solution might be outside the ρ-neighouthood of x^Ξ. In contrast $\rho > \rho^*$ (Fig. 9.i) increases the number of sub optima within the ρ-vicinity of x^Ξ, resulting in π_{parkour} converging to a local optimum. Empirically, this can be seen

[5]typical choices being velocity commands or state references trajectories

in 9.i for $\rho = 0.8$, where the policy exploits $\tilde{J}_{\mathcal{T}}$ by imitating the reference without moving forward (low $J_{\mathcal{T}}$). With respect to $J_{\mathcal{T}}$, $\rho^* = 0.4$ (Fig. 9.h) is an isolated optimum in the given range of ρ. Thus, a value of ρ around ρ^* reduces the disparity between $\tilde{J}_{\mathcal{T}}$ and $J_{\mathcal{T}}$ observed from Fig. 8.f and g.

V. Conclusion

This paper presents oracle-guided multimodal policies, a theoretical framework for task-centered control synthesis with the following philosophy: break down a task into its principal modes of operation, design an oracle with task-based priors, and learn a multimodal policy guided by the oracle. A single multimodal policy solves a task sufficiently by generalizing to the defined mode and mode parameter sets without being limited by the oracle's sub-optimality. The proposed framework is validated over two bipedal control tasks: parkour and diving, leading to extremely dynamic motions over a wide range of task variants, showing the policy's agility and versatility. Secondly, a novel latent mode space reachability analysis is proposed to characterize the policy's versatility and generalization by computing a feasible mode set function of the policy from a given state. Ongoing and future works include the sim-to-real transfer of these policies and extending the frameworks to more challenging contact-rich loco-manipulation tasks.

References

[1] R. McN. Alexander. The gaits of bipedal and quadrupedal animals. *IJRR*, 1984.

[2] Steven Bohez, Saran Tunyasuvunakool, Philemon Brakel, Fereshteh Sadeghi, Leonard Hasenclever, Yuval Tassa, Emilio Parisotto, Jan Humplik, Tuomas Haarnoja, Roland Hafner, et al. Imitate and repurpose: Learning reusable robot movement skills from human and animal behaviors. *arXiv preprint arXiv:2203.1713*, 2022.

[3] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *Robo-Letics: Workshop @CoRL 2023*, 2023.

[4] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2023.

[5] Matthew Chignoli, Donghyun Kim, Elijah Stanger-Jones, and Sangbae Kim. The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, 2021.

[6] Yuni Fuchioka, Zhaoming Xie, and Michiel Van de Panne. Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors. In *ICRA*, 2023.

[7] Duncan W. Haldane, Justin K. Yim, and Ronald S. Fearing. Repetitive extreme-acceleration (14-g) spatial jumping with salto-1p. In *IROS*, 2017.

[8] Leonard Hasenclever, Fabio Pardo, Raia Hadsell, Nicolas Heess, and Josh Merel. CoMic: Complementary task learning amp; mimicry for reusable skills. In *ICML*, Proceedings of Machine Learning Research, 2020.

[9] Fabian Jenelten, Junzhe He, Farbod Farshidian, and Marco Hutter. Dtc: Deep tracking control. *Science Robotics*, 2024.

[10] Dongho Kang, Jin Cheng, Miguel Zamora, Fatemeh Zargarbashi, and Stelian Coros. Rl + model-based control: Using on-demand optimal control to learn versatile legged locomotion. *IEEE RAL*, 2023.

[11] Lokesh Krishna and Quan Nguyen. Learning multimodal bipedal locomotion and implicit transitions: A versatile policy approach. In *IEEE IROS*, 2023.

[12] Junheng Li and Quan Nguyen. Force-and-moment-based model predictive control for achieving highly dynamic locomotion on bipedal robots. In *IEEE CDC*, 2021.

[13] Gabriel B. Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid Locomotion via Reinforcement Learning. In *Proceedings of RSS*, 2022.

[14] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 2022.

[15] Masaki Murooka, Mitsuharu Morisawa, and Fumio Kanehiro. Centroidal trajectory generation and stabilization based on preview control for humanoid multi-contact motion. *IEEE RAL*, 2022.

[16] Brahayam Ponton, Majid Khadiv, Avadesh Meduri, and Ludovic Righetti. Efficient multicontact pattern generation with sequential convex approximations of the centroidal dynamics. *IEEE T-RO*, 2021.

[17] Nikita Rudin, David Hoeller, Marko Bjelonic, and Marco Hutter. Advanced skills by learning locomotion and local navigation end-to-end. In *IEEE IROS*, 2022.

[18] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *ICLR*, 2020.

[19] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind Bipedal Stair Traversal via Sim-to-Real Reinforcement Learning. In *RSS*, 2021.

[20] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher G Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *CoRL*, 2023.