IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

SLoMo: A General System for Legged Robot Motion Imitation From Casual Videos

John Z. Zhang, *Graduate Student Member, IEEE*, Shuo Yang, *Member, IEEE*, Gengshan Yang, Arun L. Bishop, *Graduate Student Member, IEEE*, Swaminathan Gurumurthy, Deva Ramanan, and Zachary Manchester, *Member, IEEE*

***Abstract*—We present SLoMo: a first-of-its-kind framework for transferring skilled motions from casually captured "in-the-wild" video footage of humans and animals to legged robots. SLoMo works in three stages: 1) synthesize a physically plausible reconstructed key-point trajectory from monocular videos; 2) optimize a dynamically feasible reference trajectory for the robot offline that includes body and foot motion, as well as a contact sequence that closely tracks the key points; and 3) track the reference trajectory online using a general-purpose model-predictive controller on robot hardware. Traditional motion imitation for legged motor skills often requires expert animators, collaborative demonstrations, and/or expensive motion-capture equipment, all of which limit scalability. Instead, SLoMo only relies on easy-to-obtain videos, readily available in online repositories like YouTube. It converts videos into motion primitives that can be executed reliably by real-world robots. We demonstrate our approach by transferring the motions of cats, dogs, and humans to example robots including a quadruped (on hardware) and a humanoid (in simulation).**

***Index Terms*—Legged robots, computer vision for automation.**

I. Introduction

ONE of the grand challenges in robotics is to enable human- and animal-level agility for legged robots by directly imitating their natural counterparts. This motion-imitation procedure typically involves three steps: extracting motion primitives from videos or image sequences, processing motion primitives to ensure they are within the physical limits of the robot and, finally, executing those movements on robot hardware. Prior work on motion imitation relies on marker-based, multi-camera motion-capture (MoCap) systems, limiting the diversity of captured movements. An end-to-end motion-transfer solution that takes in raw video data and executes novel behaviors on real-world robots has, so far, remained elusive.

In this letter, we leverage recent advancements in neural rendering and reconstruction, trajectory optimization, and model-predictive control to build, to the best of the authors' knowledge, the first successful general framework for transferring human and animal motion skills captured by a single, moving camera to robot hardware. The framework, illustrated in Fig. 2, contains three modules: 1) a reconstruction pipeline that produces physically plausible 3D key-point trajectories from casual video footage; 2) a trajectory optimizer that solves for dynamically feasible robot state, control, and contact-force reference trajectories that closely mimic the key-point trajectories while respecting the physical limitations of the robot; and 3) a model-predictive controller (MPC) that runs at real-time rates on robot hardware to track reference trajectories. An important feature of our approach is explicit reasoning about contact interactions between the robot and environment at every stage of the framework, ensuring that offline reference trajectories can be safely executed on hardware and that the online MPC is robust to contact-timing and model mismatch. Additionally, model-based trajectory generation and control methods allow us to maintain explainability in each stage of the pipeline — something difficult to achieve with current black-box reinforcement learning (RL) approaches. Finally, our work also differs from prior works in its generality across robot morphology; our 3D reconstruction pipeline, trajectory optimizer, and MPC are all robot agnostic — enabling motion transfer from humans to humanoid robots and from animals to quadrupedal robots within a single, unified framework.

Our specific contributions are:

- A general-purpose motion-transfer framework, SLoMo, for enabling legged robots to mimic human and animal motions from casual videos.
- A novel offline reference-trajectory and contact-sequence generation technique that ensures physical feasibility.
- End-to-end experimental demonstrations transferring animal behaviors to a quadruped robot on hardware and human motions to a humanoid robot in simulation.

This letter is organized as follows: We review related literature in Section II. Our methodology is introduced in Section III. Results of simulation and hardware experiments are reported in Section IV. Finally, we summarize our conclusions and discuss directions for future research in Section V.

Manuscript received 3 May 2023; accepted 26 August 2023. Date of publication 11 September 2023; date of current version 25 September 2023. This letter was recommended for publication by Associate Editor H.-C. Lin and Editor A. Kheddar upon evaluation of the reviewers' comments. *(Corresponding author: John Z. Zhang.)*

John Z. Zhang, Gengshan Yang, Arun L. Bishop, Swaminathan Gurumurthy, Deva Ramanan, and Zachary Manchester are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15224 USA (e-mail: johnzhang@cmu.edu; y.gengshan@gmail.com; arunbish@andrew.cmu.edu; sgurumur@andrew.cmu.edu; deva@cs.cmu.edu; zacm@cmu.edu).

Shuo Yang is with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15224 USA (e-mail: shuo.yang.robotics@gmail.com).

Videos are available at https://slomo-www.github.io/website.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3313937, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3313937

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

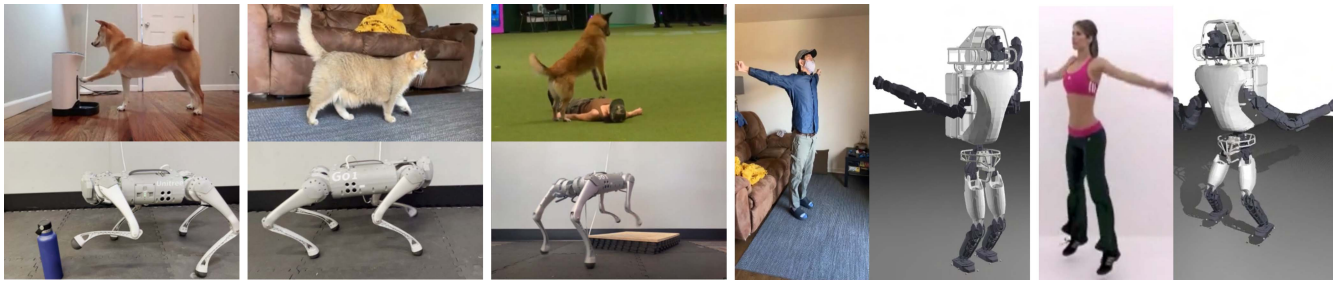

Fig. 1. Collection of video-to-robot motion-transfer demonstrations on the Unitree Go1 quadrupedal robot on hardware (left three) and the Atlas humanoid robot in simulation (right two). From left to right: A dog reaching for a water feeder with one of its front feet, a house cat pacing, a trained dog performing a Cardiopulmonary Resuscitation (CPR) exercise on its human partner, a human stretching his body and limbs, and a human demonstrating a jumping-jack exercise.

II. Background and Related Work

In this section, we review related literature on human and animal motion capture, trajectory optimization through contact, and online control for legged robots.

A. Human and Animal Motion Capture

To study human and animal body movements, MoCap systems have been widely adopted in the movie industry and research labs [1], [2], [3]. An optical MoCap system typically consists of multiple high-resolution cameras whose positions and orientations are precisely measured through a sophisticated calibration process. Multiple cameras can observe and triangulate the positions of markers, which can be mounted on human or animal subjects. Although some MoCap datasets have been made publicly available [3], obtaining novel motion sequences remains a challenge. Despite these shortcomings, most existing works in motion imitation [4], [5] rely on MoCap data. Marker-based MoCap systems have inherently limited capture volume and are very sensitive to camera configuration changes, making outdoor usage extremely difficult and unreliable. MoCap systems also typically cost tens to hundreds of thousands of dollars. All of these factors limit the diversity of environments and targets that can be studied with a MoCap system. We aim to develop a low-cost solution for imitating diverse, in-the-wild motion skills from easily-accessible videos.

Alternatively, markerless motion capture has become increasingly popular in recent years [6], [7], [8]. For example, [6] built a multi-view video-capture system to capture and reconstruct human motion. Although those systems capture whole-body movement without using markers, they still require an indoor studio with hundreds of synchronized cameras, making it challenging to generalize to in-the-wild targets and behaviors. Some recent works [9], [10] learn data-driven models to predict full body movements from a single monocular camera. However, they heavily rely on carefully-constructed template models (e.g., SMPL [11]) and do not generalize well to in-the-wild videos or non-human subjects.

Recent advances in differentiable rendering [12], [13] and robust dense point tracking (e.g. optical flow [14], [15] and DensePose [16]) have enabled test-time optimization of dense surface structure and motion given real-life videos [17], [18]. Building on these prior algorithms, our work performs key-point trajectory tracking of human and animal motions in 3D from a single, moving camera video without assuming a predefined shape template.

B. Trajectory Optimization Through Contact

Trajectory optimization is a powerful tool for designing dynamic behaviors for robotic systems. Given an initial guess, trajectory optimization formulates a nonlinear program (NLP) to solve for an optimal control sequence under robot dynamics and environmental constraints. This technique has been a major component of important breakthroughs in legged autonomy in recent years [19], [20], [21], [22].

One of the hardest problems in planning and control for legged robots is reasoning about contact forces and timing as feet make and break contact with the environment, producing discontinuous impact events. A common approach for modeling rigid-body contact interactions is to formulate the dynamics as a linear complementarity problem (LCP) [23], [24], which solves for the next system state under impact and friction constraints. These LCP dynamics can be enforced as constraints in trajectory optimization methods [25], [26].

If the contact schedule is predefined [20], the dynamics can be written as a hybrid system with known transition times. This method, commonly referred to as hybrid trajectory optimization [27], [28], can be solved quickly and is effective for periodic gaits on flat terrain. This approach has also enabled diverse locomotion behaviors on robot hardware through motion-template libraries that can then be combined and tracked online to form long-horizon locomotion behaviors over challenging terrains [22].

In contrast, if the contact schedule cannot be determined a priori, the contact interactions must be solved *implicitly*, resulting in a much larger and more challenging nonlinear optimization problem [29], [30]. This method is referred to as contact-implicit trajectory optimization [25], [26]. Reliably solving contact-implicit trajectory optimization problems to high accuracy is challenging even in offline settings, and the solution can be sensitive to model mismatch. Several previous studies have aimed to improve numerical accuracy [26], convergence properties [29], or robustness to contact model uncertainty [31], [32].

In this work, we reason about contact interactions in each stage: During 3D reconstruction, we roll out reconstructed motions in a differentiable simulator [33], where contact is

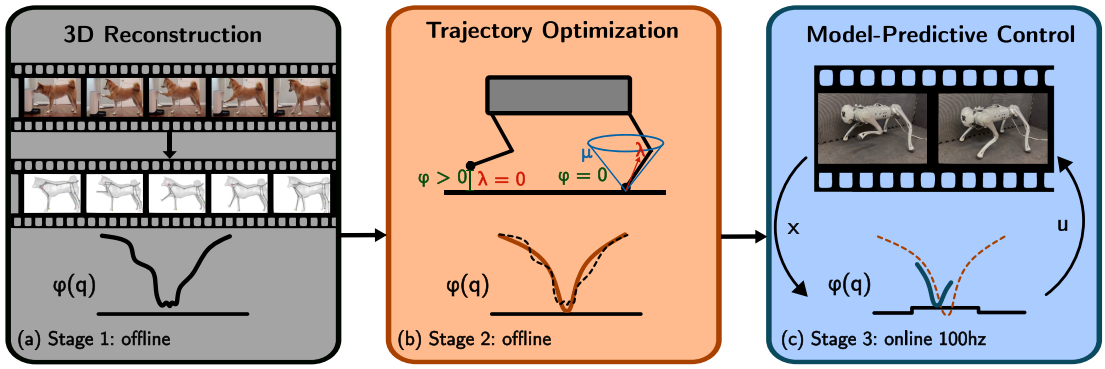

Fig. 2. SLoMo algorithm: (a) Stage 1: Process visual RGB inputs and generate body and foot trajectories in $\mathbb{R}^3$. This reconstruction considers physics but does not produce a trajectory that is dynamically feasible on the target robot. (b) Stage 2: Solve for open-loop robot state, control, and contact-force reference trajectories that imitate the 3D reconstruction while obeying robot dynamics and contact constraints. (c) Stage 3: Track the reference trajectory on robot hardware while managing model and contact-timing mismatch and disturbances.

TABLE I
COMPARISON BETWEEN RECENT MOTION IMITATION METHODS FOR LEGGED SYSTEMS

	[42]	[4]	[43]	[44]	[5]	SLoMo
Single RGB Camera	✗	✗	✗	✓	✗	✓
No Manual Label	✓	✓	✗	✓	✓	✓
Robot Hardware	✗	✓	✓	✓	✓	✓
System Agnostic	✓	✗	✗	✗	✗	✓

approximated with a spring-damper model. In trajectory optimization, we enforce rigid-body contact dynamics using LCP constraints [26]. Finally, during online tracking, we solve a simplified contact-implicit problem that can fully reason about contact mode timing.

C. Online Control for Legged Robots

Online feedback control has been widely studied for both quadrupedal [20] and bipedal [34] robots in recent years. MPC [20], [35] and RL [36], [37] have emerged as popular approaches for executing dynamic locomotion behaviors on legged systems.

Different from offline trajectory optimization, MPC typically uses a simplified model and only solves a finite-horizon variant of the optimal control problem to achieve real-time performance. Notably, [20] uses a heuristic foothold location scheduler and solves a convex quadratic program (QP) for desired stance-foot forces. This method achieves robust locomotion without any offline computation. A lower-level whole-body controller (WBC) [38] or inverse kinematics (IK) is then used to map the MPC solution into the robot joint space. At the motor level, a hand-tuned proportional-derivative (PD) controller is used to track this joint-level trajectory on the robot. The offline trajectory optimization and online MPC pipeline have enabled several impressive real-world robot behaviors, such as quadruped jumping [39], [40], humanoid parkour [22], and even coordinated heterogeneous muti-robot dance routines [41].

Model-free RL aims to learn a feedback control policy, typically represented by a deep neural network, by collecting experience in a simulated environment or the real world. The learned policy is optimized by maximizing a reward function using gradient descent algorithms. Similar to model-based pipelines, recent RL methods also rely on hand-tuned low-level PD controllers to execute behaviors in the real world [36], [37]. RL has also been successfully applied to animal imitation from motion capture reference data on a quadrupedal robot [4].

In this work, we adopt a general-purpose contact-implicit MPC (CI-MPC) algorithm [35], capable of controlling both quadruped and humanoid robots, and pair it with a low-level joint-space controller based on IK and PD feedback for hardware execution. Our contact-implicit MPC formulation can reason about contact timing and forces in real time, enabling robust tracking of complex behaviors.

D. Motion Imitation

Imitating motion primitives from nature can be an effective strategy for producing natural-looking robot behaviors while avoiding tedious, manual trajectory design. Researchers have studied various approaches for mimicking human motions [45], [46]. For example, learning-from-observation (LFO) converts MoCap motion sequences into reference motion primitives that can be tracked by a zero-moment point (ZMP) controller, which enables a humanoid robot to graciously perform traditional Japanese dance routines [47].

Recently, imitation learning has been used to produce animal-like movements for animation [42]. Ref. [4], [44], [48] demonstrated imitated motions where the sim-to-real gap was bridged through online adaptation on a quadrupedal robot. Similarly, model-based imitation [5], [43] has also produced animal-like locomotion on a real-world quadrupedal robot using hybrid trajectory optimization, where foothold locations and timing were predetermined by thresholding MoCap data. These prior works rely on marker-based motion capture to acquire motion priors. [44] uses RGB videos but still relies on manual labeling to acquire trajectories. As discussed in Section II-A, these methods have some major limitations. Recent videos from Boston Dynamics [41] demonstrate impressive robot dancing through complicated choreography design, character animation,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

and robot trajectory optimization procedures; an expensive, time-consuming process.

In robot manipulation, recent work [49], [50] takes essential steps towards acquiring in-the-wild robotic skills from real-world RGB videos through reinforcement learning. The key insight from these two studies is that computer-vision techniques can now process widely-available human and animal footage into representations that can be very effective priors for acquiring robotic skills. In this work, we tackle the problem of imitating locomotion skills by extracting target motions from videos using 3D reconstruction. Different from previous RL-based imitation methods that only consider robot kinematics, we use optimal control methods that allow us to explicitly reason about dynamics.

We propose a simple and cost-effective method: synthesize legged robot motion primitives directly from casual RGB videos without manual labels and leverage a model-based controller for robust execution on robot hardware. Additionally, we demonstrate that our control strategy generalizes across quadruped and humanoid robots. Moreover, the model-based approach allows us to interpret the output trajectories and explicitly reason about hardware limitations like torque limits.

III. Video-to-Robot Motion Transfer

In this section, we present the SLoMo algorithm for robot motion imitation from in-the-wild videos: Section III-A (Fig. 2(a)) describes the 3D reconstruction pipeline for generating physically-plausible key-point trajectories from casual footage. Section III-B (Fig. 2(b)) explains the offline trajectory-optimization problem used for constructing dynamically feasible robot reference trajectories and foot-contact sequences that mimic key-point movements. Section III-C (Fig. 2(c)) details the contact-implicit MPC algorithm [35] for tracking the optimal reference online.

A. Physics-Informed Reconstruction From Casual Videos

Given videos of a target animal or human, our goal is to estimate its kinematic key-point trajectory in the world coordinates. Similar to prior work [17], [18], [51], we simultaneously reconstruct articulated shapes and kinematic skeleton trajectories, connected by a blend skinning model. To reconstruct physically plausible trajectories, we set up a physics-informed optimization by coupling differentiable rendering costs with an additional physics-roll-out cost.

We define the $\mathbf{x}(t) \in \mathbb{R}^{3\,N}$ to be the estimated key-point trajectory, where N is the total number of key-points, $t \in \{1 \dots T\}$ to be discrete time steps, and T to be the number of frames in a given video.

Shape Model: We use a Multi-Layer Perceptron (MLP) parameterized by σ to represent the visual properties of the default state of the object:

$$(d, \mathbf{c}) = \mathbf{MLP}_{\boldsymbol{\sigma}}(\mathbf{X}), \tag{1}$$

where $d \in \mathbb{R}$ is the assigned signed distance and $\mathbf{c} \in \mathbb{R}^3$ is the color at each point $X \in \mathbb{R}^3$. The points that have distance $d = 0$ are on the surface of the object. This representation is similar to a neural radiance field (NeRF) [13] except we remove the view dependence of color.

Fig. 3. Physics-informed 3D reconstruction pipeline: Differentiable rendering from a monocular video (left) and differentiable physics simulation (right) update key-point motion estimates (middle). Solid red and green arrows represent rendering and physics-roll-out costs; dashed arrows are corresponding gradients.

Kinematic Skeleton Model: To model the motion of the subject animal or human, we use a predefined target kinematic skeleton model consisting of B rigid links and N spherical joints, among which a root link is selected to define the model's location and orientation in space. For simplicity, we utilize joint locations as key points (Fig. 3). The key-point trajectory $\mathbf{x(t)}$ can be calculated using forward kinematics. When the skeleton changes configuration, the object's shape should deform and move along with the skeleton. This motion is described by a neural blend-skinning model $\mathcal{W}_t(\mathbf{X})$ [18] as an MLP parameterized by λ:

$$\mathcal{W}_t(\mathbf{X}) = \mathbf{MLP}_{\boldsymbol{\lambda}}(\mathbf{X}; Q, G, t). \tag{2}$$

The neural blend-skinning model warps 3D points of the new configuration to the default configuration, which can then be used to query the visual properties of the object. We further parameterize the joint angles $Q = \mathbf{MLP}_{\kappa}(t)$ and SE(3) root-link transformation $G = \mathbf{MLP}_{\eta}(t)$. Then, after training, the zero-level set of d given by

$$(d, \mathbf{c}) = \mathbf{MLP}_{\boldsymbol{\sigma}}(\mathcal{W}_t(\mathbf{X})) \tag{3}$$

represents the surface of the object at time t.

Differentiable Volume Rendering: In addition to the object shape model, we train another parameterized background scene model similar to (1). The shape and scene models can then be used together to differentiably render images given a camera's view transformation and intrinsic parameters [52]. We achieve this by performing ray casting for each image pixel p. For all 3D points along the ray within a given distance range, we use (3) to query their 3D color and density and perform a weighted average to compute $\hat{\mathbf{c}}_t(p)$, the pixel values on the rendered image, including 2D color and object silhouette. We further render optical flow from t to $t+1$ by warping and projecting the queried points from the current configuration to the next frame configuration.

Rendering Cost: We compare the expected color, object silhouette, and optical flow of pixels $\hat{\mathbf{c}}_t(p)$ on the rendered image with the input video observations at time t. The observation $\bar{\mathbf{c}}_t(p)$ comes from basic image-processing and segmentation methods [14], [53]. A rendering cost function is defined as:

$$\mathcal{L}_{render} = \sum_t \sum_p \|\hat{\mathbf{c}}_t(p) - \bar{\mathbf{c}}_t(p)\|^2 \tag{4}$$

We use its gradients to update the model parameters σ, λ, κ, and η with the Adam optimizer [54].

Physics Roll-Out Cost: Using only the rendering cost, the trained shape model can generate motions from the same view point [18]. However, these motions are often physically unrealistic due to piece-wise scale ambiguity at each individual patch [55] — a common issue for reconstructing monocular videos. For example, a dog can be up close and floating in the air or far away and on the ground. Both are equally valid from the same monocular visual evidence, but only the latter obeys physics. To resolve this ambiguity, we introduce a physics-roll-out cost in the optimization problem to encourage a physically plausible rendering solution.

Treating the key-point skeleton as a floating-base multi-rigid-body system, we denote its generalized coordinates as $q(t)$. During differentiable rendering, using the latest parameters κ and η, we can generate a reference trajectory $q_d(t)$ with $t \in \{1 \dots T\}$. Then, in a differentiable simulator [33], we simulate the multi-body key-point skeleton model with a simple PD controller attempting to track $q_d(t)$ to compute the following cost,

$$\mathcal{L}_{physics} = \sum_t \|q(t) - q_d(t)\|^2. \tag{5}$$

This cost function is differentiable with respect to κ, η, and ξ. Note that, in this stage, the physics cost only encourages physical realism in a "soft" way and does not guarantee dynamic feasibility like later stages in SLoMo. Thus, we deem the rendered key-point trajectory physically *plausible*.

Coordinate-Descent Optimization: In theory, the cost $\mathcal{L}_{render} + \mathcal{L}_{physics}$ can be optimized together. However, in practice, the volume rendering and the physics simulator run at different frequencies, making joint optimization inefficient. Instead, we use coordinate descent to alternately minimize each cost function. In each iteration, we first minimize the rendering cost, during which we regularize Q and G's output toward the previous simulator-generated trajectory. We then perform a physics roll-out optimization step, where only physics parameters ξ are updated. We also use other strategies to improve convergence such as over-parameterization. At convergence, we take the rendered key-point trajectories $x^*(t)$ as output to the next stage of the motion-imitation pipeline. More details of the reconstruction optimization can be found in [18], [51].

B. Contact-Implicit Trajectory Optimization

After generating key-point trajectories from monocular videos, we solve a trajectory-optimization problem subject to robot dynamics and contact constraints. In particular, we use contact-implicit trajectory optimization [25], [26], which jointly solves for robot states, controls, and contact forces with a direct collocation formulation. Compared to hybrid trajectory optimization [27], [28], the contact-implicit method does not require a predefined contact sequence, which is an important advantage in our motion-imitation workflow since the reconstructed key-point trajectories can suffer from dynamically infeasible artifacts (e.g. foot sliding), even with the physics roll-out cost, which makes applying a heuristic contact schedule impractical. By using the contact-implicit formulation, we allow the optimizer to automatically generate a feasible robot gait sequence and corresponding reference trajectory that is similar to the key-point trajectory without separately predefining a contact schedule.

The offline contact-implicit trajectory-optimization problem has the following form,

$$\begin{aligned}
\underset{\mathcal{H},\mathcal{X},\mathcal{U},\lambda}{\text{minimum}} \quad & \sum_{t=1}^{T-1} \frac{h_t}{2} \left[(x_t - x_t^*)^\top Q (x_t - x_t^*) + u_t^\top R u_t \right] \\
& + (x_N - x_N^*)^\top Q_N (x_N - x_N^*) \\
\text{subject to} \quad & x_{t+1} = \mathbf{NCP}_t(h_t, x_t, u_t, \lambda_t), \\
& u_t \le u_{\max}, \\
& u_t \ge u_{\min},
\end{aligned} \tag{6}$$

where h_t is the time step, $x_t = (q_t, v_t)$ is the state, and u_t are the controls, and λ_t are contact forces at time t, respectively. $\mathbf{NCP}(x, u)$ represents the robot's nonlinear dynamics, including contact constraints, as a nonlinear complementarity problem [24], [26]. We optimize a quadratic tracking cost where x_t^* is the reconstructed key-point state at time t and Q and R are diagonal weighting matrices. This formulation allows the solver to infer a contact sequence and corresponding robot trajectories by imitating noisy and dynamically infeasible key-point data. We refer to a solution of (6) as a *reference trajectory*.

C. Contact-Implicit Model-Predictive Control

To stabilize and track reference trajectories in real-time, we use contact-implicit model-predictive control [35]. CI-MPC is a general method for controlling robots that make and break contact with their environment. Different from standard convex MPC approaches, CI-MPC models the robot's dynamics with a time-varying linear complementarity problem. This LCP can be thought of as a local approximation of the nonlinear complementarity problem in (6) about the reference trajectory, which makes it computationally easier to solve. Importantly, however, it maintains the ability to reason about discontinuous contact-switching events.

The CI-MPC tracking problem is:

$$\begin{aligned}
\underset{\mathcal{X},\mathcal{U}}{\text{minimize}} \quad & \sum_{t=1}^{H} \frac{1}{2} \left[(x_t - \bar{x}_t)^\top Q (x_t - \bar{x}_t) \right. \\
& \left. + (u_t - \bar{u}_t)^\top R (u_t - \bar{u}_t) \right] \\
\text{subject to} \quad & x_{t+1} = \mathbf{LCP}_t(x_t, u_t), \\
& u_t \le u_{\max}, \\
& u_t \ge u_{\min},
\end{aligned} \tag{7}$$

where x_t, u_t are state and control decision variables and $\bar{x}_t$, $\bar{u}_t$ are the reference state and control trajectories from (6) and H is the MPC horizon, which is generally shorter than the full length T of the reference trajectory to enable faster online solution times. The interested reader is referred to [35] for more details on CI-MPC.

IV. Experiments and Results

In this section, we present the results of experiments demonstrating the capabilities of SLoMo on a variety of robotic systems in simulation and on hardware. In particular, we demonstrate two important features of our method: 1) our entire framework can successfully transfer motions from casual videos to robot hardware and 2) our pipeline is model agnostic and can support both quadruped and humanoid robots. We carefully identify canonical legged robot movements that require no contact switching, periodic contact switching, and dynamic non-periodic contact switching. Experiment videos are available on our website. An implementation of the framework can be found on GitHub.

A. Experimental Setup

Our 3D reconstruction stage is implemented with PyTorch. Processing a one-minute video takes eight hours on a computer with 8 NVIDIA GeForce RTX 3080 GPUs. We run offline trajectory optimization and online MPC on a workstation computer equipped with an Intel i9-12900KS CPU and 64 GB of memory. This workstation computer is connected to the robot via Ethernet. All hardware experiments are run on a Unitree Go1 quadruped robot.

B. Quadruped

We verify our video-to-robot motion transfer approach on three video input examples: a dog reaching for water (*dog-reach*) — shown in Fig. 1 (left); a cat pacing across a living room (*cat pace*) — shown in Fig. 1 (second to left); and a dog performing CPR on a human (*dog CPR*) — shown in Fig. 1 (second to right).

Point-foot quadruped model: We use a simplified robot dynamics model for offline trajectory optimization and online control. This point-foot representation of the robot dynamics neglects the leg dynamics and instead models the feet as point masses. The model has 18 degrees of freedom and 12 control inputs. The controls are modeled as three-dimensional internal forces between each foot and the body. Note that this model is different from the skeleton model used during reconstruction, which contains additional links and joints (e.g. legs, tails, torso, etc.).

Hardware setup: For the hardware experiments, a hand-tuned PD controller running at 1000 Hz is used to track the forces and foot positions computed by the MPC policy on the Unitree Go1 robot. State estimation is also provided at 1000 Hz with a Kalman Filter that utilizes robot joint encoders and an onboard IMU. We take time discretization of 0.05 s offline and track the reference with online MPC at 100 Hz with a prediction horizon of 0.15 s on the Unitree Go1 robot.

Dog reach: We take a video clip of a dog reaching for an automatic water feeder and compute an offline reference trajectory that is 5.0 s long. (Fig. 1 left).

Cat pace: We take a video clip of a cat pacing across a living room (Fig. 1 second to left) and compute an offline reference trajectory that is 3.0 s long, then repeated to form a continuous forward walking gait.

Dog CPR: We take a video clip of a dog performing CPR on a human (Fig. 4 left) and compute an offline reference trajectory that is 0.65 s long We model the terrain as a step where the front feet plan for a contact distance 0.1 m higher than the back feet. This motion primitive is fairly dynamic and would be difficult to design manually. Fig. 5 compares the output trajectories of each stage of SLoMo for each foot, where the key-point trajectory (blue) contains infeasible ground penetration, while the optimized reference (red) eliminates this unphysical artifact.

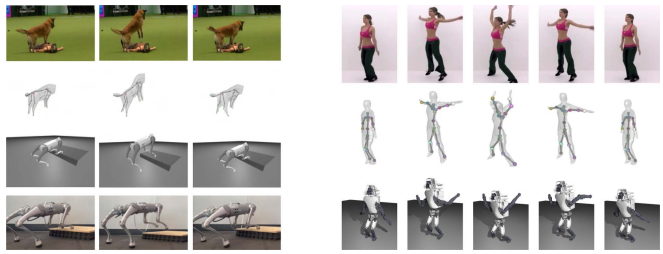

Fig. 4. *Dog CPR* (left), *Human jumping jack* (right) motion imitation demonstrations. The top row shows frames from the original video, the second row shows the key-point trajectory, and the third row shows the dynamically feasible trajectory being executed by the Unitree Go1 and Atlas robots. The bottom row on the left shows *dog CPR* on hardware.

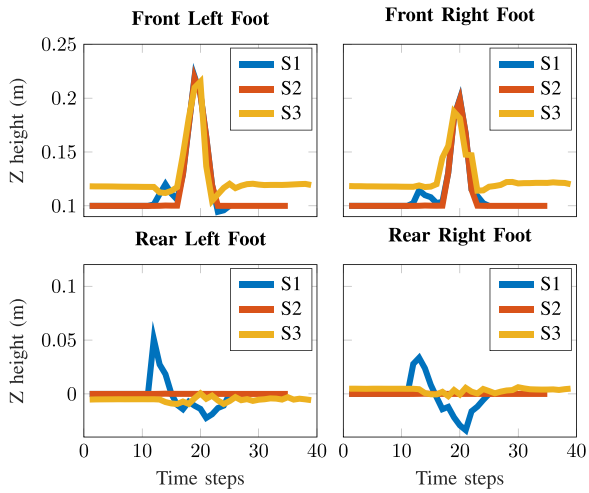

Fig. 5. Foot height trajectories comparison for the dog-CPR experiment. Each plot corresponds to one foot of the robot. The blue lines are trajectories generated by the 3D reconstruction stage (S1). The red lines are the outputs of the trajectory optimization stage (S2). The yellow lines are the state feedback on robot hardware showing the result foot trajectories of the MPC policy (S3).

C. Humanoid

We show that our framework can also be applied to imitating human movements on humanoid robots in simulation on the Atlas robot.

Point-foot humanoid model: We use a simplified model to represent the dynamics of the humanoid robot. We model each foot as a rectangular prism with two contact points: one at the toe and the other at the heel. We similarly model each hand as a point mass. The model has 24 degrees of freedom and 18 control inputs. We use time discretization of 0.1 s for the experiments below.

Human Stretching: We take a video of a human raising both arms in a stretching motion (Fig. 1 second to left) and compute an offline reference trajectory that is 3.8 s long.

Fig. 6. Comparison between the reconstructed key-point reference (yellow), optimized reference (teal, ours), and learned imitation policy [4] (purple).

Human Jumping Jacks: We take a video of a human performing a jumping jack exercise (Fig. 4 right) and compute an offline reference trajectory that is 1.5 s long.

D. Comparisons to an RL Policy

We show that it is also possible to replace the trajectory-optimization and MPC steps of our method with an RL method [4] in simulation. We train RL policies for the Dog Reach and Cat Pace examples with 4 random seeds each. In Fig. 6, we showcase the highest-performing policy on the robot. Notably, while the best policy shows reasonable performance, we find that the RL policies exhibit a high degree of variance across different seeds in the Dog Reach example. This high variance makes fair, rigorous comparisons between model-based optimization and model-free RL difficult. However, we still believe imitating animal movements using RL is a promising approach and deserves further investigation.

V. Discussion and Conclusions

We present SLoMo, a first-of-its-kind framework for enabling legged robots to imitate animal and human motions captured from real-world casual monocular videos. Our research highlights that recent advancements in 3D reconstruction [18], [51] are effective at extracting physically plausible motion trajectories solely from monocular RGB videos. These trajectories are quite noisy and dynamically infeasible, even with the physics-based roll-out cost, making them unsafe to execute directly on real robots. To overcome this, an off-the-shelf trajectory optimizer [26] or RL method [4] is necessary to plan for dynamically feasible trajectories that track the retrieved motions from videos.

In this letter, we use a contact-implicit trajectory optimization method to reason about contact events and timing, and an MPC that is robot-agnostic and handles non-periodic contacts. This specific offline trajectory optimization and MPC combination facilitates motion transfer regardless of behavior periodicity, allowing us to generalize to arbitrary human and animal behaviors.

A. Limitations

SLoMo is a promising first step towards imitating human and animal behaviors on real-world robots from in-the-wild video footage. However, several limitations remain that should be addressed in future research: First, we make key model simplifications and assumptions in Sections III-B and III-C by using a point-foot model to represent the quadruped and humanoid dynamics. It should be possible to extend this work to use full-body dynamics in both offline and online optimization steps to fully leverage a robot's capabilities. For example, humans and animals are capable of making contact with the world in very rich ways (e.g. a dog can use its head to move an object), but executing such behaviors for legged robots remains an open research problem. Second, the reconstruction step is computationally expensive, and we manually scale the reconstructed character to better match the kinematics of the target robot in Section III-A. It should be possible to automate this scaling in Problem (6). Addressing morphological differences between video characters and corresponding robots in a principled, automated manner and accelerating reconstruction will be crucial for scaling our framework to large video datasets.

B. Future Work

Many exciting directions for future research remain: First, several trade-offs should be investigated in which components of our framework are swapped out. For example, leveraging RGB-D video data can likely improve reconstruction quality at the expense of data availability. Secondly, it should be possible to deploy the SLoMo pipeline on humanoid hardware, imitate more challenging humanoid behaviors, and execute behaviors on more challenging terrains where humans and animals have demonstrated highly athletic and robust behaviors but manually designing trajectories for robots can be challenging. Finally, this work can benefit from a proven, high-performance online control stack (e.g. whole-body control [38]). We believe that real-world visual data can be a rich source of robot behaviors, and taking advantage of recent advancements in reasoning about such visual data can be extremely powerful for robotics.

Acknowledgment

The authors would like to thank Taylor A. Howell, Simon Le Cleac'h, and members of the Robotic Exploration Lab at CMU for their insightful discussions and feedback.

References

[1] D. Kang, F. De Vincenti, N. C. Adami, and S. Coros, "Animal motions on legged robots using nonlinear model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 11955–11962.
[2] H. Luo et al., "Artemis: Articulated neural pets with appearance and motion synthesis," *ACM SIGGRAPH*. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3528223.3530086#sec-cit
[3] "Carnegie-Mellon mocap database," [Online]. Available: http://mocap.cs.cmu.edu
[4] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Proc. Robot.: Sci. Syst. XVI. Robot.: Sci. Syst. Found.*, 2020, p. 64.
[5] D. Kang, S. Zimmermann, and S. Coros, "Animal gaits on quadrupedal robots using motion matching and model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8500–8507.
[6] H. Joo et al., "Panoptic studio: A massively multiview system for social interaction capture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 190–204, Jan. 2019.
[7] J. S. Yoon, Z. Yu, J. Park, and H. S. Park, "HUMBI: A large multiview dataset of human body expressions and benchmark challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 623–640, Jan. 2023.
[8] P. C. Bala, B. R. Eisenreich, S. B. M. Yoo, B. Y. Hayden, H. S. Park, and J. Zimmermann, "Openmonkeystudio: Automated markerless pose estimation in freely moving macaques," *Nat. Library Med.*. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32917899/
[9] M. Kocabas, N. Athanasiou, and M. J. Black, "VIBE: Video inference for human body pose and shape estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5253–5263.

[10] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3D human dynamics from video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5614–5623.

[11] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 851–866, 2015.

[12] A. Tewari et al., "Advances in Neural Rendering," *Comput. Graph. Forum*, vol. 41, no. 2, pp. 703–735, 2022.

[13] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[14] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019.

[15] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 402–419.

[16] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense human pose estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7297–7306.

[17] G. Yang et al., "LASR: Learning articulated shape reconstruction from a monocular video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15975–15984.

[18] G. Yang, M. Vo, N. Neverova, D. Ramanan, A. Vedaldi, and H. Joo, "BANMo: Building animatable 3D neural models from many casual videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2853–2863.

[19] S. Kuindersma et al., "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Auton. Robots*, vol. 40, pp. 429–455, 2016.

[20] G. Bledt, M. J. Powell, B. Katz, J. D. Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2245–2252.

[21] B. Katz, J. Di Carlo, and S. Kim, "Mini Cheetah: A platform for pushing the limits of dynamic quadruped control," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 6295–6301.

[22] Boston Dynamics, "More parkour atlas,"

[23] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *Int. J. Numer. Methods Eng.*, vol. 39, no. 15, pp. 2673–2691, 1996.

[24] T. A. Howell, S. L. Cleac'h, J. Z. Kolter, M. Schwager, and Z. Manchester, "Dojo: A differentiable physics engine for robotics," 2022, *arXiv:2203.00806*.

[25] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 69–81, 2014.

[26] Z. Manchester, N. Doshi, R. J. Wood, and S. Kuindersma, "Contact-implicit trajectory optimization using variational integrators," *Int. J. Robot. Res.*, vol. 38, no. 12/13, pp. 1463–1476, 2019.

[27] D. Pardo, M. Neunert, A. Winkler, R. Grandia, and J. Buchli, "Hybrid direct collocation and control in the constraint-consistent subspace for dynamic legged robot locomotion," in *Proc. Robot.: Sci. Syst. XIII. Robot.: Sci. Syst. Found.*, 2017, p. 42.

[28] C. Mastalli et al., "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2536–2542.

[29] T. A. Howell, S. L. Cleac'h, K. Tracy, and Z. Manchester, "CALIPSO: A differentiable solver for trajectory optimization with conic and complementarity constraints," in *Proc. Int. Symp. Robot. Res.*, 2022, pp. 504–521.

[30] "Ipopt: Documentation," 2005. [Online]. Available: https://coin-or.github.io/Ipopt/

[31] L. Drnach and Y. Zhao, "Robust trajectory optimization over uncertain terrain with stochastic complementarity," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1168–1175, Apr. 2021.

[32] L. Drnach, J. Z. Zhang, and Y. Zhao, "Mediating between contact feasibility and robustness of trajectory optimization through chance complementarity constraints," *Front. Robot. AI*, vol. 8, 2022, Art. no. 785925.

[33] M. Macklin, "Warp: A high-performance python framework for GPU simulation and graphics," in *Proc. NVIDIA GPU Technol. Conf.*, Mar. 2022. [Online]. Available: https://github.com/nvidia/warp

[34] M. J. Powell, E. A. Cousineau, and A. D. Ames, "Model predictive control of underactuated bipedal robotic walking," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 5121–5126.

[35] S. L. Cleac'h et al., "Fast contact-implicit model-predictive control," 2021, *arXiv:2107.05616*.

[36] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," in *Proc. Conf. Robot Learn.*, 2022, pp. 138–149.

[37] L. Smith, I. Kostrikov, and S. Levine, "A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning," *RSS*, 2023. [Online]. Available: https://www.roboticsproceedings.org/rss19/p056.pdf

[38] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," Sep. 2019, *arXiv:1909.06586*.

[39] C. Nguyen, L. Bao, and Q. Nguyen, "Continuous jumping for legged robots on stepping stones via trajectory optimization and model predictive control," in *Proc. IEEE 61st Conf. Decis. Control*, 2022, pp. 93–99.

[40] Z. Zhou, B. Wingo, N. Boyd, S. Hutchinson, and Y. Zhao, "Momentum-aware trajectory optimization and control for agile quadrupedal locomotion," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7755–7762, Jul. 2022.

[41] Boston Dynamics, "No time to dance boston dynamics,"

[42] X. B. Peng, P. Abbeel, S. Levine, and M. V. D. Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, 2018.

[43] T. Li, J. Won, S. Ha, and A. Rai, "FastMimic: Model-based motion imitation for agile, diverse and generalizable quadrupedal locomotion," *Robotics*, 2023. [Online]. Available: https://www.mdpi.com/2218-6581/12/3/90

[44] Q. Yao et al., "Imitation and adaptation based on consistency: A quadruped robot imitates animals from videos using deep reinforcement learning," *Robotics*, vol. 12, 2022, Art. no. 90.

[45] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2002, pp. 1390–1397.

[46] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 2806–2812.

[47] S. Nakaoka et al., "Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances," *Int. J. Robot. Res.*, vol. 26, no. 8, pp. 829–844, 2007.

[48] S. Kim, M. Sorokin, J. Lee, and S. Ha, "Human motion control of quadrupedal robots using deep reinforcement learning," in *Proc. Robot.: Sci. Syst. XVIII. Robot.: Sci. Syst. Found.*. 2022, p. 21.

[49] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," in *Proc. Robot.: Sci. Syst. XVIII. Robot.: Sci. Syst. Found.*, 2022, p. 26.

[50] A. Sivakumar, K. Shaw, and D. Pathak, "Robotic Telekinesis: Learning a robotic hand imitator by watching humans on YouTube," *RSS*, 2022. [Online]. Available: https://www.roboticsproceedings.org/rss18/p023.html

[51] G. Yang, S. Yang, J. Z. Zhang, Z. Manchester, and D. Ramanan, "Physically plausible reconstruction from monocular videos," in *Proc. Int. Conf. Comput. Vis.*, 2023.

[52] M. Niemeyer and A. Geiger, "GIRAFFE: Representing scenes as compositional generative neural feature fields," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11453–11464.

[53] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9799–9808.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[55] S. Kumar, Y. Dai, and H. Li, "Superpixel soup: Monocular dense 3D reconstruction of a complex dynamic scene," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1705–1717, May 2021.