

# DiffSim2Real: Deploying Quadrupedal Locomotion Policies Purely Trained in Differentiable Simulation

Joshua Bagajo<sup>\*,1</sup>, Clemens Schwarke<sup>\*,1</sup>, Victor Klemm<sup>1</sup>, Ignat Georgiev<sup>2,3</sup>, Jean-Pierre Sleiman<sup>1,3</sup>, Jesus Tordesillas<sup>1,4</sup>, Animesh Garg<sup>2</sup>, and Marco Hutter<sup>1</sup>

**Abstract**—Differentiable simulators provide analytic gradients, enabling more sample-efficient learning algorithms and paving the way for data intensive learning tasks such as learning from images. In this work, we demonstrate that locomotion policies trained with analytic gradients from a differentiable simulator can be successfully transferred to the real world. Typically, simulators that offer informative gradients lack the physical accuracy needed for sim-to-real transfer, and vice-versa. A key factor in our success is a smooth contact model that combines informative gradients with physical accuracy, ensuring effective transfer of learned behaviors. To the best of our knowledge, this is the first time a real quadrupedal robot is able to locomote after training exclusively in a differentiable simulation.

## I. INTRODUCTION AND APPROACH

The majority of Reinforcement Learning (RL) algorithms rely on Zeroth-order Gradient (ZoG) estimates during optimization, allowing the use of conventional physics simulators that are typically non-differentiable. However, differentiable simulators offer analytically computed First-order Gradients (FoGs), with lower variance [1], [2], [3] and therefore improved sample efficiency and asymptotic policy performance [4], [5]. Thus, leveraging FoGs offers the potential to learn from pixels [6] or to learn policies for systems with many degrees of freedom [7]. Unfortunately, contact interactions are often simulated in a discontinuous manner, making FoG-based optimization challenging. Some simulators address this by using soft contact models, which are continuous and smooth but less physically accurate for typical locomotion problems compared to discontinuous hard contact models [8]. Additionally, penalty-based soft contact models often require smaller time steps and thus increase computational demand and lengthen gradient chains. Consequently, learning the contact-rich task of quadrupedal locomotion and transferring the learned behavior to the real world with either hard or penalty-based contact has not yet succeeded [9], [10], [11]. Instead, we adopt an analytically smooth contact model, introduced in our previous work [9], that provides a smoothed optimization surface while maintaining physical accuracy, combining the advantages of hard and soft contact. The contact model draws inspiration from the role of stochasticity in current learning frameworks, a key factor in the success of RL [12], [13]. We then employ

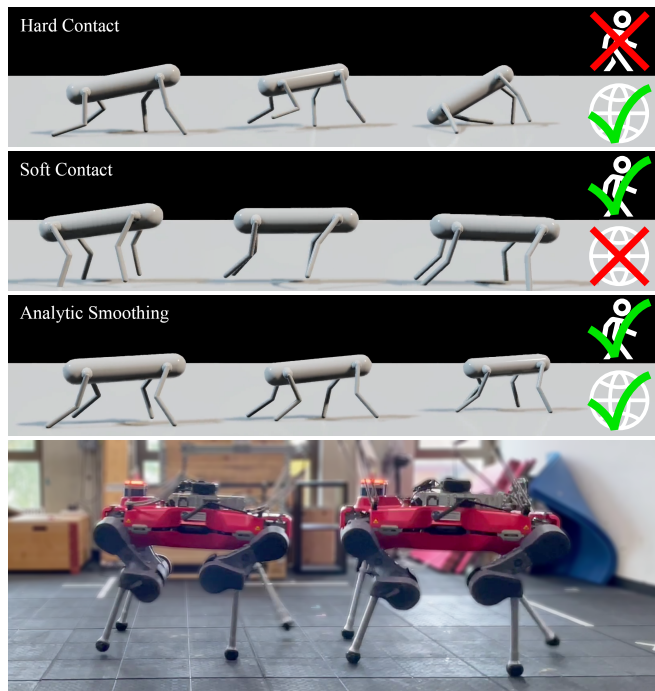


Fig. 1: A quadrupedal robot learning to walk on flat terrain in a differentiable simulation. Policies trained with a *hard contact model* follow unreasonable foothold patterns and do not learn to locomote robustly. Training with a *soft contact model* results in stable locomotive gaits but the learned behaviors do not transfer to real hardware. Policies trained with an *analytically smooth contact model* exhibit effective and stable locomotive gaits and transfer to the real world. Video: <https://youtu.be/2wZmmUyqUQM>.

the Short-Horizon Actor-Critic (SHAC) algorithm [7] that leverages FoGs to enhance learning efficiency over purely ZoG-based algorithms such as Proximal Policy Optimization (PPO) [14]. Finally, we demonstrate that locomotion policies learned with this approach successfully transfer to the real world.

Previous attempts were confined to simulation. The first locomotion policy purely trained in a differentiable simulator was presented in [11], but exhibited undesirable behaviors like front flips. Table I summarizes relevant simulators and their approaches to differentiation and contact modeling. Nimble [15] implements symbolic differentiation and solves a sparse Linear Complementarity Problem (LCP) to resolve contact, while DiffTaichi [16] uses impulse-based methods to avoid differentiating the LCP of contact. Warp [17] and

<sup>\*</sup>Shared 1st authorship. <sup>1</sup>Robotic Systems Lab, ETH Zürich, Switzerland. <sup>2</sup>Georgia Institute of Technology, United States. <sup>3</sup>The AI Institute, United States. <sup>4</sup>Institute for Research in Technology, ICAI School of Engineering, Comillas Pontifical University, Spain.  
CoRL 2024 Workshop 'Differentiable Optimization Everywhere'

Brax [10] leverage GPU acceleration for fast rigid-body simulations and support multiple contact models. Dojo [18] emphasizes physical accuracy but is limited by slower execution and lacks parallelism. At the time of writing, none of the current simulators offer parallelization combined with accurate dynamics and informative gradients to learn transferable locomotion behaviors. A sim-to-real transfer for quadrupedal locomotion policies learned using FoGs was only achieved with a second non-differentiable simulator to ensure accurate physics [19]. In this work, we extended Warp with custom physics to benefit from GPU parallelization.

TABLE I: Differentiable Rigid-Body Simulators

Name	Differentiation	Contact Modeling	Device
Nimble [15]	Symbolic	LCP	CPU
DiffTaichi [16]	Automatic	Impulse-based	GPU
Warp [17]	Automatic	XPBD [20], Soft	GPU
Brax [10]	Automatic	MuJoCo [21], PBD [22]	GPU
Dojo [18]	Symbolic	NCP	CPU

## II. CONTACT SIMULATION

Our simulation is based on Moreau’s time stepping scheme [23]. However, the Gauss-Seidel algorithm used to compute contact forces is slightly modified from implementations such as [24] to smooth the originally hard contact model. Contact forces are scaled by a sigmoid function that depends on the distance between potentially contacting bodies. For more details on the simulation, we refer to [9]. The analytically smooth contact model has several advantages over hard and soft contact models. First, it smooths the discontinuities of hard contact. While stochastic smoothing would have similar effects on the dynamics, gradients would still remain uninformative FoGs as explained in Fig. 2. Second, the contact model remains stable for larger simulation time steps compared to traditional soft contact models. Lastly, the similarity to stochastically smoothed dynamics suggests that the hard contact case is implicitly within the domain of the analytically smooth contact model, promising successful transfer to hard contact or the real world.

## III. SIM-TO-REAL TRANSFER

To transfer locomotion policies learned in our differentiable simulation to ANYbotics’ ANYmal D robot, we first align our learning setup with [25], [26], which have demonstrated successful sim-to-real transfer. To test the validity of the dynamics of our simulation, we train policies with PPO in our simulator and transfer them to IsaacSim, a hard contact simulation used in [25], [26]. After ensuring that the learned behaviors successfully transfer, we progress to training policies with SHAC, making use of the FoGs computed by the simulator. However, the learning setup designed for PPO does not immediately lead to the desired behaviors with our method. Instead, our method requires a simplified inertia model (only diagonal inertia components with lower magnitudes) to find a reasonable locomotion policy. The reward formulation needs adaptation as well. We

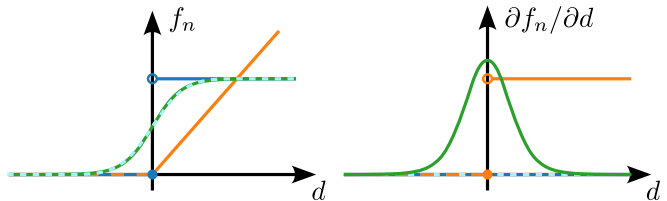


Fig. 2: The normal contact force (left) and its gradient (right) with respect to the penetration depth between two contacting bodies. *Hard contact* (blue) exhibits a discontinuity at  $d = 0$ . Its analytical gradient is zero almost everywhere. *Soft contact* (orange) is continuous but does not accurately model stiff contact without becoming unstable because the normal force is unbounded. *Stochastically smoothing hard contact* (cyan) removes the discontinuity, but the FoG gradient remains zero and thus uninformative. *Analytically smoothing hard contact* (green) induces similar effects on the dynamics as stochastic smoothing, with the advantage of an informative FoG.

find that combining rewards from [9] with rewards from [26] that allow for differentiation results in successful learning.

Key elements for sim-to-real transfer, according to [26], are domain randomization and the integration of a learned actuator model. Initially, we adopt the domain randomization method from [26], though the required extent of randomization for our approach remains to be determined. While domain randomization helps to close the sim-to-real gap and smooths out local minima in the learning objective, we observe that higher levels of randomization lead to slower convergence during training. Incorporating an actuator model, which typically contains a history or memory architecture, would significantly increase the complexity of the computational graph. Instead, we implement a PD-controller with velocity-based torque saturation, using system identification-derived parameters [27]. This approach provides efficient gradient propagation and achieves performance comparable to the learned actuator model in the relevant actuation domain, without adding unnecessary complexity to the computational graph.

## IV. RESULTS AND LIMITATIONS

In previous work [9], we found that common soft and hard contact models do not lead to transferable locomotion policies, as shown in Fig. 1. Our introduction of analytic smoothing enabled smooth gaits that successfully transferred to hard contact simulation. In this work, we further show that policies learned in a differentiable simulator also transfer effectively to real-world environments. However, learning locomotion with FoGs is sensitive to physical parameters and the reward function, and our approach has not yet surpassed state-of-the-art RL policies in terms of locomotion behavior. Nevertheless, learning with SHAC requires significantly fewer samples—over an order of magnitude less—compared to PPO. Although we presented a proof of concept in this preliminary work, an in-depth analysis will be necessary in future research. Furthermore, we plan to introduce roughness to the terrain to enhance the locomotion behavior, as the current flat terrain limits stepping height and robustness.

## REFERENCES

- [1] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning," *Journal of Machine Learning Research*, vol. 21, no. 132, pp. 1–62, 2020.
- [2] N. Wiedemann, V. Wüest, A. Loquercio, M. Müller, D. Floreano, and D. Scaramuzza, "Training efficient controllers via analytic policy gradient," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1349–1356.
- [3] Y. Zhang, Y. Hu, Y. Song, D. Zou, and W. Lin, "Back to newton's laws: Learning vision-based agile flight via differentiable physics," *arXiv preprint arXiv:2407.10648*, 2024.
- [4] H. J. Suh, M. Simchowitz, K. Zhang, and R. Tedrake, "Do differentiable simulators give better policy gradients?" in *International Conference on Machine Learning*. PMLR, 2022, pp. 20 668–20 696.
- [5] I. Georgiev, K. Srinivasan, J. Xu, E. Heiden, and A. Garg, "Adaptive horizon actor-critic for policy learning in contact-rich differentiable simulation," *arXiv preprint arXiv:2405.17784*, 2024.
- [6] J. Y. Luo, Y. Song, V. Klemm, F. Shi, D. Scaramuzza, and M. Hutter, "Residual policy learning for perceptive quadruped control using differentiable simulation," *arXiv preprint arXiv:2410.03076*, 2024.
- [7] J. Xu, V. Makoviyuchuk, Y. Narang, F. Ramos, W. Matusik, A. Garg, and M. Macklin, "Accelerated policy learning with parallel differentiable simulation," in *International Conference on Learning Representations*, 2021.
- [8] C. Gehring, R. Diethelm, R. Siegwart, G. Nützi, and R. Leine, "An evaluation of moreau's time-stepping scheme for the simulation of a legged robot," in *IDETC/CIE 2014*, no. DETC2014-34374, 2014.
- [9] C. Schwarke, V. Klemm, J. Tordesillas, J.-P. Sleiman, and M. Hutter, "Learning quadrupedal locomotion via differentiable simulation," *arXiv preprint arXiv:2404.02887*, 2024.
- [10] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, "Brax-a differentiable physics engine for large scale rigid body simulation," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.
- [11] J. Degraeve, M. Hermans, J. Dambre, and F. Wyffels, "A differentiable physics engine for deep learning in robotics," *Frontiers in Neuro-robotics*, vol. 13, no. March, pp. 1–9, 2019.
- [12] H. J. T. Suh, T. Pang, and R. Tedrake, "Bundled Gradients Through Contact Via Randomized Smoothing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4000–4007, 2022.
- [13] T. Pang, H. J. Suh, L. Yang, and R. Tedrake, "Global Planning for Contact-Rich Manipulation via Local Smoothing of Quasi-Dynamic Contact Models," *IEEE Transactions on Robotics*, pp. 1–20, 2023.
- [14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv e-prints*, 2017.
- [15] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu, "Fast and Feature-Complete Differentiable Physics for Articulated Rigid Bodies with Contact," *Robotics: Science and Systems*, 2021.
- [16] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, "DiffTaichi: Differentiable programming for physical simulation," in *ICLR*, 2019.
- [17] M. Macklin, "Warp: A high-performance python framework for gpu simulation and graphics," <https://github.com/nvidia/warp>, March 2022, NVIDIA GPU Technology Conference (GTC).
- [18] T. A. Howell, S. Le Cleac'h, J. Z. Kolter, M. Schwager, and Z. Manchester, "Dojo: A differentiable simulator for robotics," *arXiv preprint arXiv:2203.00806*, vol. 9, 2022.
- [19] Y. Song, S. Kim, and D. Scaramuzza, "Learning quadruped locomotion using differentiable simulation," *arXiv preprint arXiv:2403.14864*, 2024.
- [20] M. Macklin, M. Müller, and N. Chentanez, "Xpbd: position-based simulation of compliant constrained dynamics," in *Proceedings of the 9th International Conference on Motion in Games*, 2016, pp. 49–54.
- [21] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.
- [22] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position based dynamics," *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 109–118, 2007.
- [23] J. J. Moreau, "Unilateral contact and dry friction in finite freedom dynamics," in *Nonsmooth mechanics and Applications*. Springer, 1988, pp. 1–82.
- [24] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization with implicit hard contacts," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3316–3323, 2018.
- [25] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, p. 3740–3747, Jun. 2023. [Online]. Available: <http://dx.doi.org/10.1109/LRA.2023.3270034>
- [26] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.
- [27] F. Bjelonic, F. Tischhauser, and M. Hutter, "Towards bridging the gap: Scalable sim-to-real transfer for legged locomotion," 2024, unpublished.