Learning Legged MPC with Smooth Neural Surrogates

***Abstract*— Deep learning and model predictive control (MPC) are complementary in legged robotics, but integrating learned dynamics with online planning remains challenging. Neural network models introduce three key issues: (1) stiff transitions from contact events inherited from data; (2) additional non-physical local nonsmoothness; and (3) non-Gaussian errors induced by rapid state changes. We address (1) and (2) with the *smooth neural surrogate*, a network with tunable smoothness that yields informative predictions and derivatives for trajectory optimization through contact. To address (3), we train with a heavy-tailed likelihood that better matches empirical error distributions. Together, these choices improve the reliability, scalability, and generalizability of learned legged MPC. Across zero-shot locomotion tasks of increasing difficulty, smooth neural surrogates with robust learning reduce cumulative cost in well-conditioned settings (typically ≈10–50%) and deliver substantially larger gains where standard neural dynamics fail, enabling reliable execution (0/5 → 5/5 success) and orders-of-magnitude improvements in robustness.**

I. Introduction

Reinforcement learning (RL) and model predictive control have both seen widespread success in legged robotics. Learning provides representational flexibility and scalability, particularly through techniques such as domain randomization, enabling capabilities like blind locomotion over challenging terrain [1], [2]. Model predictive control, by contrast, enables online behavior shaping by adjusting costs and constraints, making it well suited for rapid task switching without retraining [3], [4]. Despite these complementary strengths, the effectiveness of MPC within an RL pipeline remains limited for legged robots, and existing controllers tend to favor either learned policies with limited flexibility or scenario-specific analytic models with limited generalization or scalability.

Learned MPC, which combines learned dynamics with online planning, offers a promising middle ground (Fig. 1a). However, improving the prediction error of neural dynamics does not guarantee improvements in control performance [5]. Even exact models can perform poorly when contact dynamics are not handled appropriately during optimization [3], [6]–[9].

Our starting point is a simple but important observation: *the pathologies of trajectory optimization under contact dynamics are compounded when the dynamics are modeled with standard neural networks.* We hypothesize that these failures are driven primarily by nonsmoothness which manifests in three ways:

1) Neural network dynamics often inherit the very stiff transitions present in the ground-truth contact dynamics, making these models difficult to optimize through.
2) Neural networks frequently introduce additional local nonsmoothness and, consequently, local optima that do not exist in the underlying physical dynamics.
3) Training datasets for these stiff dynamics necessarily contain rapid state changes, which can destabilize learning and violate common Gaussian noise/likelihood assumptions.

Fig. 1. **Versatile quadruped control with smooth neural dynamics. a)** We introduce two inductive biases for learning dynamics for contact-rich MPC: a smooth neural surrogate architecture and a heavy-tailed loss function. **b)** Our smooth neural surrogates learn terrain-varying dynamics and state estimation while generalizing to new tasks outside the training domain. Together, they combine the representational flexibility of deep learning with the task-level adaptability of MPC, enabling reliable whole-body control across diverse behaviors and environments.

A natural approach to (1)–(2) is to control network smoothness. Smooth or Lipschitz-constrained architectures can improve gradient quality, generalization, and robustness [10]–[14], but remain underexplored in robotics [15], [16], particularly for the stiff dynamics of legged robots. Here, smoothing is intended to provide a well-conditioned optimization problem as opposed to physical accuracy.

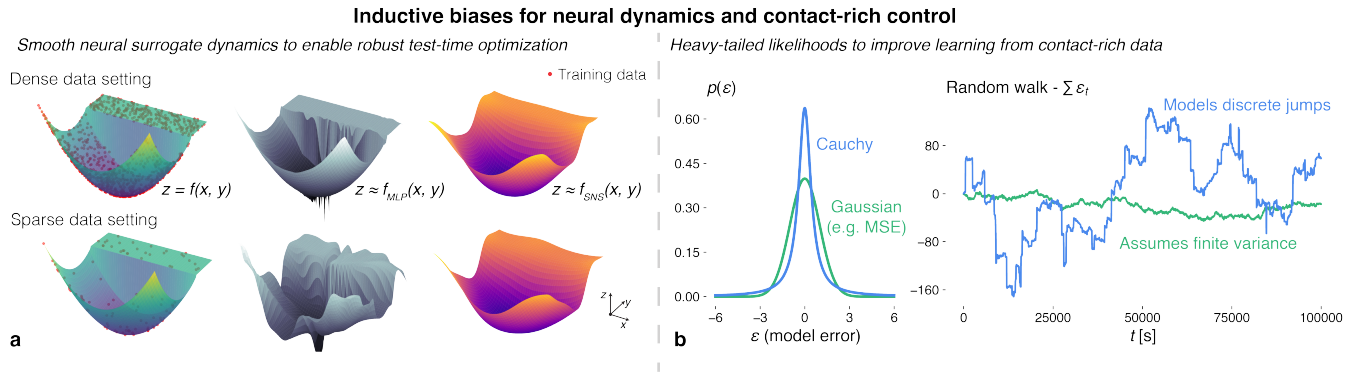

Fig. 2. **Improving model learning for contact-rich control. a)** Smooth neural surrogates, structured with Lipschitz-based weight normalization, drastically improve the generalization of learned MPC. These networks reduce both physical nonsmoothness from contact events and local nonsmoothness related to overfitting. As underlying dynamics may be nonsmooth, these smooth neural surrogates bias the learned representation to improve trajectory optimization as opposed to physical accuracy. **b)** Commonly used Gaussian likelihoods for dynamics learning, like the MSE, fail to capture discrete jumps in the model error. In contrast, Cauchy likelihoods naturally describe impulsive contacts, improving model learning from stiff, contact-rich data.

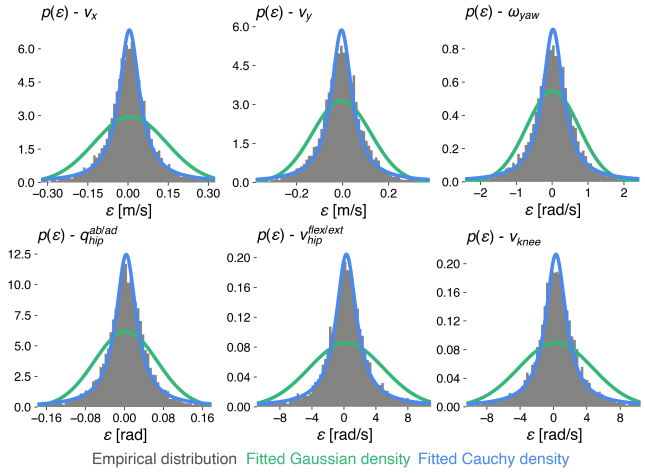

Fig. 3. **Errors for learned quadruped dynamics follow a heavy-tailed Cauchy distribution.** Empirical one-step prediction residuals ε across representative quadruped state variables. Residuals, as computed from the replay buffer, align more closely with a Cauchy density than with a Gaussian; suggesting that Gaussian losses, such as MSE, can be poorly suited for learning the stiff dynamics of legged robots.

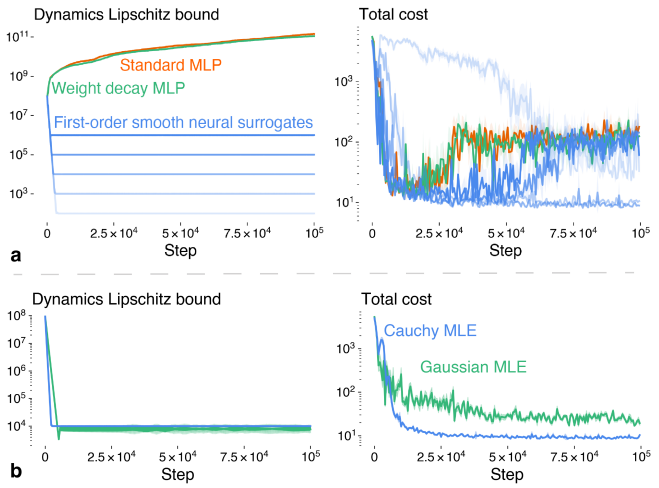

Fig. 4. **Smooth neural surrogates excel at MPC for legged robots. a)** Smooth neural surrogate dynamics with varying Lipschitz constraints are compared to standard and weight-decay MLPs. Traditional networks lose continuity during training, while smooth surrogates converge quickly to their sensitivity budgets. A trade-off emerges: moderate smoothness yields the best quadruped locomotion, while too much or too little degrades learning and planning. **b)** Heavy-tailed maximum likelihood improves training stability and MPC: Gaussian models hinder convergence, while Cauchy likelihoods improve performance.

Addressing (3) requires revisiting loss assumptions. Gaussian error models are standard [5], [17]–[23], yet often unverified and ill-suited for datasets with heavy-tailed errors, which we observe in legged-robot dynamics. Heavy-tailed likelihoods such as the Cauchy are better matched to impulsive noise [24].

We study these issues in full-order MPC for legged robots, learning both dynamics and state estimation from scratch. On quadrupeds, smooth neural dynamics combined with heavy-tailed training substantially improve control performance. Under domain randomization, this yields versatile whole-body behaviors (Fig. 1b), enabling tasks that are challenging for perception-free model-free RL or classical MPC.

II. Learning Smooth Neural Surrogates

We describe two complementary inductive biases (Fig. 2) for learned MPC in contact-rich tasks: enforcing smooth learned dynamics and using heavy-tailed likelihoods.

Learned MPC. Given states x and actions u, learned dynamics are commonly modeled as [5], [18]

$$x_{t+1} = x_t + d\hat{\mu}_\theta(x_t, u_t)dt + \varepsilon, \qquad \varepsilon \sim P, \tag{1}$$

where P is an unspecified residual distribution. Training typically uses one- or multi-step Gaussian losses (e.g., MSE). In general, dynamics may depend on histories $(x_{t-H:t}, u_{t-H:t})$ to improve accuracy or handle partial observability [17], [23], [26]. In our setup, terrain and friction are randomized during data collection in MuJoCo [27]; history-dependent models and a learned state estimator compensate for this variability.

As the dynamics are learned, we solve the following

TABLE I

DYNAMICS TEST ERROR FOR DIFFERENT MODELS AND LIKELIHOODS

Method	One-step MAE
MLP–Gaussian	0.540 ± 0.001
MLP–Cauchy (ES)	0.545 ± 0.001
MLP–Cauchy	$\mathbf{0.476 \pm 0.001}$
SNS–Gaussian	0.604 ± 0.001
SNS–Cauchy	0.571 ± 0.001

Notes. Cauchy likelihoods reduce error (**mean** $\pm$ **95% CI**) across architectures. Smooth neural surrogates introduce bias under nonsmooth dynamics, but yield better performance in both sampling- and gradient-based MPC. ES denotes early stopping at 12,500 steps.

TABLE II

MODEL, LIKELIHOOD, AND OPTIMIZER PERFORMANCE ACROSS SAMPLE BEHAVIORS AT TEST-TIME

Method	**Fwd. Trot.** (1 [m/s])	**Fwd. Gallop.** (1.5 [m/s])
MLP–Gaussian–DIAL	10.44±0.61 (5/5)	24.54±0.95 (5/5)
MLP–Gaussian–GGN	5.15±3.19 (5/5)	33.35±3.62 (0/5)
SNS–Gaussian–DIAL	8.50±0.54 (5/5)	30.53±4.27 (3/5)
SNS–Gaussian–GGN	3.87±0.03 (5/5)	17.13±2.10 (5/5)
MLP–Cauchy–DIAL (early stopped)	11.97±0.86 (5/5)	27.39±1.01 (4/5)
MLP–Cauchy–GGN (early stopped)	4.40±0.06 (5/5)	23.02±2.81 (0/5)
MLP–Cauchy–DIAL	20.10±12.01 (4/5)	27.34±8.39 (4/5)
MLP–Cauchy–GGN	3.86±0.02 (5/5)	53.27±14.94 (0/5)
SNS–Cauchy–DIAL	7.62±0.29 (5/5)	24.94±1.07 (5/5)
SNS–Cauchy–GGN	**3.46±0.03** (5/5)	**15.35±0.11** (5/5)

Notes. Each entry reports cumulative cost and success rate over 11 s episodes (top: **mean** $\pm$ **95% CI**, bottom: **success rate**). A trial is **successful** if no undesired ground contact occurs with the **base or hips**. Method names follow **NN–Likelihood–Optimizer**: MLP (standard) or SNS (ours); Gaussian or Cauchy likelihood; and DIAL or GGN for DIAL-MPC [25] and our generalized Gauss–Newton solver. All trials use the ground truth states $x_0^{\text{est.}} = x_0^{\text{g.t.}}$.

optimal control problem (OCP):

$$\begin{aligned} \min_{x_{0:T},\, u_{0:T-1}} \quad & \sum_{t=0}^{T} \ell\big(x_t, u_t\big) \\ \text{s.t.} \quad & x_{t+1} - x_t - d\hat{\mu}_\theta(x_t, u_t)dt = 0 \\ & x_0 - x_0^{\text{est.}} = 0 \\ & g_i\big(x_t, u_t\big) \le 0. \end{aligned} \tag{2}$$

The OCP is recast in single-shooting form using differentiable rollouts. We compare a generalized Gauss–Newton (GGN) method [28]–[30] with the sampling-based DIAL-MPC [25], deploying GGN on hardware; both use spline-parameterized actions and solve the OCP at 50 Hz. During training, reference trajectories defining ℓ are randomized, while at test time new costs induce new behaviors and assess generalization.

Quantifying continuity and smoothness. Convergence of Eq. (2), for gradient-based methods, often depends on continuity and smoothness [31]. A function f is c-Lipschitz if $|f(a_0) - f(a_1)|_p \le c|a_0 - a_1|_p$. Smoothness requires Lipschitz-continuous derivatives. While direct control of Lipschitz constants is difficult, randomized smoothing can enforce differentiability at additional compute [32], [33].

Neural networks admit tractable smoothness bounds. For an L-layer multi-layered perceptron (MLP), $c_{\text{MLP}} \lesssim \prod_\ell |W^{(\ell)}|_p$, with equality under 1-Lipschitz activations. Though conservative, this bound enables efficient global smoothness control without post hoc smoothing.

Lipschitz-based weight normalization. Following [10], we normalize weights using learned scalars c_ℓ that parameterize layerwise ∞-norms:

$$\begin{aligned} \hat{W}_{ij}^{(\ell)} &= \text{normalization}\Big(W_{ij}^{(\ell)},\, c_\ell\Big), \qquad c_\ell > 0, \\ &= \min\left(1,\, \frac{c_\ell}{\sum_k |W_{ik}^{(\ell)}|}\right) W_{ij}^{(\ell)}. \end{aligned} \tag{3}$$

We parameterize $c_\ell = \exp(\theta_c^{(\ell)})$ for improved stability over softplus in [10]. Defining $C := \prod_{\ell=1}^{L} c_\ell$ and $S := \sum_{\ell=1}^{L} c_\ell \prod_{j<\ell} c_j$, the Jacobian Lipschitz constant satisfies $d_{\text{MLP}} \lesssim CS$.

Smooth neural surrogate objectives. We train the smooth neural surrogate MLP (SNS-MLP) using the k-th order objective ($k \in 1, 2$):

$$\mathcal{L}_k = \mathcal{L}_{\text{main}} + \lambda \max\left(1,\, \frac{C\, S^{k-1}}{B_k}\right), \tag{4}$$

where B_k is a smoothness budget. This scaling keeps λ independent of smoothness magnitude. First-order ($k = 1$, $B_1 = c_{\text{ub}}$) bounds sensitivity, while second-order ($k = 2$, $B_2 = d_{\text{ub}}$) bounds curvature. The second-order objective also constrains first-order behavior, though both constraints can be enforced jointly via a basic modification of Eq. (4).

Heavy-tailed maximum likelihood estimation. In our experiments on legged robots, we observe that one-step prediction residuals exhibit heavy-tailed behavior (Fig. 3). So, if we assume $\varepsilon \sim \mathcal{C}(x; \mu, \Sigma)$, we want to minimize any loss proportional to the *Cauchy negative log-likelihood* during training. In the n-dimensional case:

$$\begin{aligned} -\log \mathcal{C}(x; \mu, \Sigma) = \tfrac{n+1}{2} \log(1 + (x-\mu)^\top \Sigma^{-1}(x-\mu)) \\ + \tfrac{1}{2} \log|\Sigma| + \text{const.}, \end{aligned} \tag{5}$$

Unlike Gaussian losses, the Cauchy objective saturates for large residuals, reducing sensitivity to impulsive noise. In our experiments Σ is considered fixed (homoscedastic) and isotropic $\Sigma = \sigma^2 I$. In this case, the location coincides with the median, and the dispersion parameter σ is proportional to the median absolute deviation (MAD) from the median.

III. RESULTS

All experiments are conducted in simulation with key results validated on hardware. We first compare prediction error across learned models on a held-out test set (Table I). Experiments throughout this section are generated in simulation, and the results are verified on hardware. As hypothesized, Cauchy likelihoods stabilize learning and reduce error, whereas smoothing introduces slight bias and increases MAE.

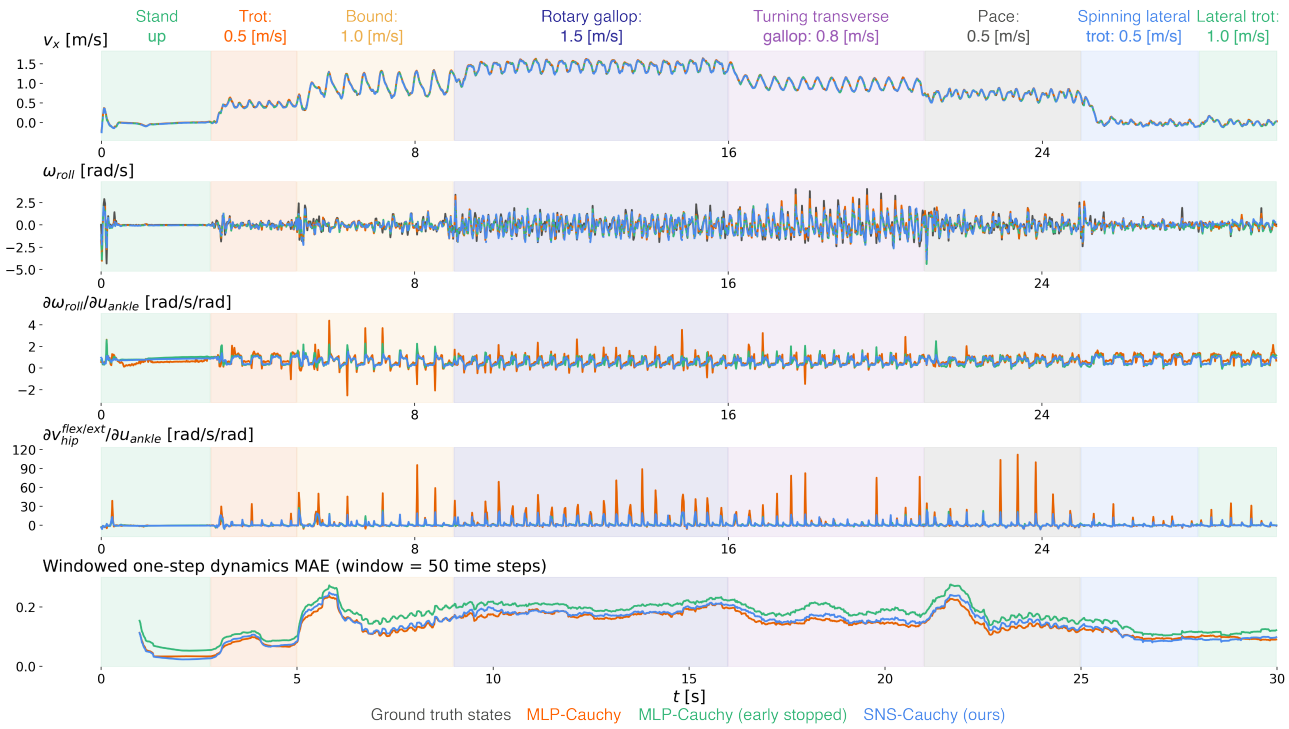

Fig. 5. **First-order model properties correlate with planning performance, while zeroth-order error can be misleading.** Single-step predictions of representative states and derivatives along a multi-behavior trajectory. Although this behavior sequence was generated with an SNS-based controller (all others fail), we compare with standard and smooth MLPs to understand failure modes. The fully trained MLP learns stiffer, sharper dynamics (up to $\approx 3\times$ larger derivative range than SNS) and achieves slightly lower MAE (SNS: 0.141 ± 0.004, MLP: 0.136 ± 0.004, early-stopped: 0.166 ± 0.004), reflecting nonsmooth ground-truth dynamics. Early stopping yields smoother derivatives but still introduces stiff/noisy regions and degrades zeroth-order accuracy. In contrast, SNS enforces globally smooth derivatives and delivers substantially better control performance (Table II), despite minor bias. In some regions, SNS also matches or improves MAE, indicating smoothing can enhance both first- and zeroth-order behavior.

Fig. 4 instead reports cumulative control cost for multi-directional trotting over a 5 s episode using GGN–MPC. Only SNS-based dynamics with moderate Lipschitz bounds continue to improve control throughout training (Fig. 4a). Standard and weight-decay-regularized MLPs become progressively less smooth (Fig. 4a), improving prediction error but degrading trajectory optimization through stiff contacts. Even with SNS dynamics, Cauchy MLE improves convergence and prevents large contact-induced errors from dominating the learning signal (Fig. 4b).

Table II summarizes forward trotting and galloping across model classes, optimizers, and likelihoods. Forward trotting is the most favorable setting for standard MLPs, yet smoothing still improves control under both GGN- and DIAL-MPC. The multi-directional case (Fig. 4a), which includes lateral velocity commands—a common failure mode—further highlights the advantages of smooth neural surrogates as task complexity increases. Galloping remains challenging for standard MLPs and Gaussian likelihoods (Table II), with observed failures (0/5 success) contrasted by consistent success for smooth surrogates (5/5).

Finally, we evaluate a complex multi-gait sequence (Fig. 5). Only MPC with smooth neural surrogates successfully executes the sequence. Notably, along a successful rollout, the standard MLP (without early stopping) still achieves the lowest MAE; however, zeroth- and first-order predictions are substantially smoother with the SNS-MLP. Early stopping mitigates extreme gradients but degrades prediction quality and remains insufficient for control. On hardware (Fig. 1b)), we observe the ability to complete tasks that are challenging for classical MPC and model-free RL.

IV. Conclusion

We present a framework for learning neural dynamics that are explicitly structured for optimization in stiff contact-rich systems. By combining smooth neural surrogates with heavy-tailed maximum likelihood estimation, we obtain models that provide informative gradients and robust predictions under impulsive dynamics. Our results show that enforcing smoothness and revisiting statistical assumptions are both critical for reliable learned MPC. More broadly, this work highlights that model design for control should prioritize optimization compatibility, not just predictive accuracy.

Future work. While effective for quadrupedal locomotion, several extensions remain open. Studies in contact-rich manipulation, humanoid robotics would help further understand the benefits and limitations of smooth neural surrogates. Finally, although Lipschitz constraints have been proposed for more expressive architectures such as CNNs and transformers, the practical benefits of smoothing in high-capacity networks for robotics and control remain largely unexplored.

REFERENCES

[1] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[2] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[3] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4906–4913.

[4] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," *arXiv preprint arXiv:2212.00541*, 2022.

[5] M. Lutter, L. Hasenclever, A. Byravan, G. Dulac-Arnold, P. Trochim, N. Heess, J. Merel, and Y. Tassa, "Learning dynamics models for model predictive agents," *arXiv preprint arXiv:2109.14311*, 2021.

[6] M. Posa and R. Tedrake, "Direct trajectory optimization of rigid body dynamical systems through contact," in *Algorithmic Foundations of Robotics X: Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics*. Springer, 2013, pp. 527–542.

[7] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM review*, vol. 59, no. 4, pp. 849–904, 2017.

[8] G. Kim, D. Kang, J.-H. Kim, S. Hong, and H.-W. Park, "Contact-implicit model predictive control: Controlling diverse quadruped motions without pre-planned contact modes or trajectories," *The International Journal of Robotics Research*, vol. 44, no. 3, pp. 486–510, 2025.

[9] S. Le Cleac'h, T. A. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester, "Fast contact-implicit model predictive control," *IEEE Transactions on Robotics*, vol. 40, pp. 1617–1629, 2024.

[10] H.-T. D. Liu, F. Williams, A. Jacobson, S. Fidler, and O. Litany, "Learning smooth neural functions via lipschitz regularization," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–13.

[11] M. Rosca, T. Weber, A. Gretton, and S. Mohamed, "A case for new neural network smoothness constraints," in *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, ser. Proceedings of Machine Learning Research, J. Zosa Forde, F. Ruiz, M. F. Pradier, and A. Schein, Eds., vol. 137. PMLR, 12 Dec 2020, pp. 21–32. [Online]. Available: https://proceedings.mlr.press/v137/rosca20a.html

[12] C. Anil, J. Lucas, and R. Grosse, "Sorting out lipschitz function approximation," in *International conference on machine learning*. PMLR, 2019, pp. 291–301.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[14] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," *arXiv preprint arXiv:1705.10941*, 2017.

[15] X. Song, J. Duan, W. Wang, S. E. Li, C. Chen, B. Cheng, B. Zhang, J. Wei, and X. S. Wang, "Lipsnet: a smooth and robust neural network with adaptive lipschitz constant for high accuracy optimal control," in *International Conference on Machine Learning*. PMLR, 2023, pp. 32 253–32 272.

[16] W. G. Y. Tan and Z. Wu, "Robust machine learning modeling for predictive control using lipschitz-constrained neural networks," *Computers & Chemical Engineering*, vol. 180, p. 108466, 2024.

[17] J. Xu, E. Heiden, I. Akinola, D. Fox, M. Macklin, and Y. Narang, "Neural robot dynamics," *arXiv preprint arXiv:2508.15755*, 2025.

[18] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *Advances in neural information processing systems*, vol. 31, 2018.

[19] J. Amigo, R. Khorrambakht, E. Chane-Sane, N. Mansard, and L. Righetti, "First order model-based rl through decoupled backpropagation," *arXiv preprint arXiv:2509.00215*, 2025.

[20] A. Byravan, L. Hasenclever, P. Trochim, M. Mirza, A. D. Ialongo, Y. Tassa, J. T. Springenberg, A. Abdolmaleki, N. Heess, J. Merel *et al.*, "Evaluating model-based planning and planner amortization for continuous control," *arXiv preprint arXiv:2110.03363*, 2021.

[21] M. Parmar, M. Halm, and M. Posa, "Fundamental challenges in deep learning for stiff contact dynamics," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5181–5188.

[22] P. Roth, J. Frey, C. Cadena, and M. Hutter, "Learned perceptive forward dynamics model for safe and platform-aware robotic navigation," *arXiv preprint arXiv:2504.19322*, 2025.

[23] W. Xiao, H. Xue, T. Tao, D. Kalaria, J. M. Dolan, and G. Shi, "Anycar to anywhere: Learning universal dynamics model for agile and adaptive mobility," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8819–8825.

[24] P. Tsakalides and C. Nikias, "Maximum likelihood localization of sources in noise modeled as a cauchy process," in *Proceedings of MILCOM'94*. IEEE, 1994, pp. 613–617.

[25] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 4974–4981.

[26] S. A. Moore, B. P. Mann, and B. Chen, "Automated global analysis of experimental dynamics through low-dimensional linear embeddings," *arXiv preprint arXiv:2411.00989*, 2024.

[27] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[28] N. N. Schraudolph, "Fast curvature matrix-vector products for second-order gradient descent," *Neural computation*, vol. 14, no. 7, pp. 1723–1738, 2002.

[29] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3402–3421, 2023.

[30] F. Messerer, K. Baumgärtner, and M. Diehl, "Survey of sequential convex programming and generalized gauss-newton methods," *ESAIM: Proceedings and Surveys*, vol. 71, pp. 64–88, 2021.

[31] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

[32] 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.

[33] T. Pang, H. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *IEEE Transactions on robotics*, vol. 39, no. 6, pp. 4691–4711, 2023.