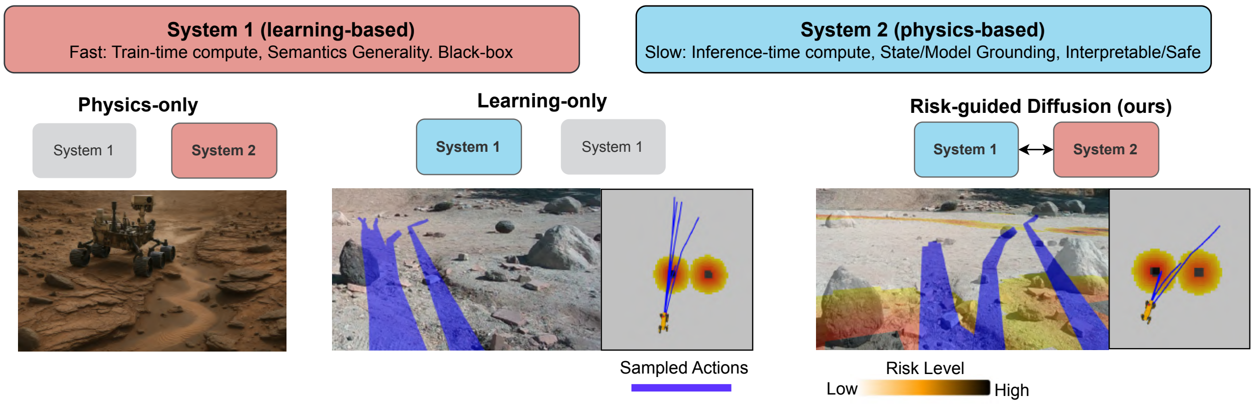
Risk-Guided Diffusion: Toward Deploying Robot Foundation Models In Space, Where Failure Is Not An Option

*Rohan Thakker[1], *Adarsh Patnaik[1,3], Vince Kurtz[2], Jonas Frey[3], Jonathan Becktor[1], Sangwoo Moon[1], Rob Royce[1], Marcel Kaufmann[1], Georgios Georgakis[1], Pascal Roth[3], Joel Burdick[2], Marco Hutter[3], Shehryar Khattak[1]



(a) Rover drives away from potential evidence of water due to lack of semantic awareness.

(b) Navigation foundation model fails to identify a hazard not seen in training data.

(c) Robot successfully avoids hazard while following semantic image-based goals.

Fig. 1: An example robotic planetary surface exploration mission scenario to provide motivation for the proposed hybrid learning and physics-based approaches. (a) shows ignoring semantic information completely may be detrimental towards overall mission success, (b) shows failure of learning-based approaches to adapt to out-of-distribution hazards during runtime, (c) shows viability of the proposed hybrid approaches.

Abstract— Safe, reliable navigation in extreme, unfamiliar terrain is required for future robotic space exploration missions. Recent generative-AI methods learn semantically aware navigation policies from large, cross-embodiment datasets, but offer limited safety guarantees. Inspired by human cognitive science [1], we propose a risk-guided diffusion framework that fuses a fast, learned "System-1" with a slow, physics-based "System-2," sharing computation at both training and inference to couple adaptability with formal safety. Hardware experiments conducted at the NASA JPL's Mars-analog facility, Mars Yard, show that our approach reduces failure rates by up to 4× while matching the goal-reaching performance of learning-based robotic models by leveraging inference-time compute without any additional training.

I. Introduction

NASA's surface exploration missions currently rely on pre-mission maps and ground supervision. This limits robotic exploration of distant worlds like Enceladus or Europa, where one-way communication latency nears 50 minutes and outages can persist for weeks [2, 3]. Future robots need to autonomously traverse GNSS-denied, visually degraded terrain with no prior maps to successfully conduct their missions. Robotic foundation models trained on cross-embodiment datasets have demonstrated strong vision-language reasoning [4, 5, 6], but their reliability drops for out-of-distribution scenarios. This work asks: Can foundation models deliver adaptive autonomous mobility in space environments without compromising safety?

Adding safety to learning-based navigation methods has seen considerable work over the years. Prior works like [7, 8] have shown the benefits of using a traversability estimation module to plan safe paths. Several foundation models have benefited from test time compute scaling [9]. Works such as [10] use a binary traversability mask at training time and modify the loss function of a diffusion model to output trajectories in traversable regions. SafeDiffuser [11] provides safety guarantees using a Control Barrier Function (CBF) [12] at inference time, but does not account for stochasticity in traversability estimation. Furthermore, CBF construction can be challenging for stochastic systems.

Inspired by Kahneman [1], several recent works have been proposed to combine slow and fast thinking [13, 14, 15]. In this work, we explore the possibility of using a learning based navigation foundation model as fast System 1 and physics-based risk-assessment as slow System 2. Contributions: (1) We enable safe adaptation to out-of-training-distribution terrains by leveraging stochastic physics-based traversability estimation [7]. (2) [16, 17] achieve cross-embodied transfer by abstracting the action space to a sequence of waypoints with an underlying low-level tracking controller. However, every robot has

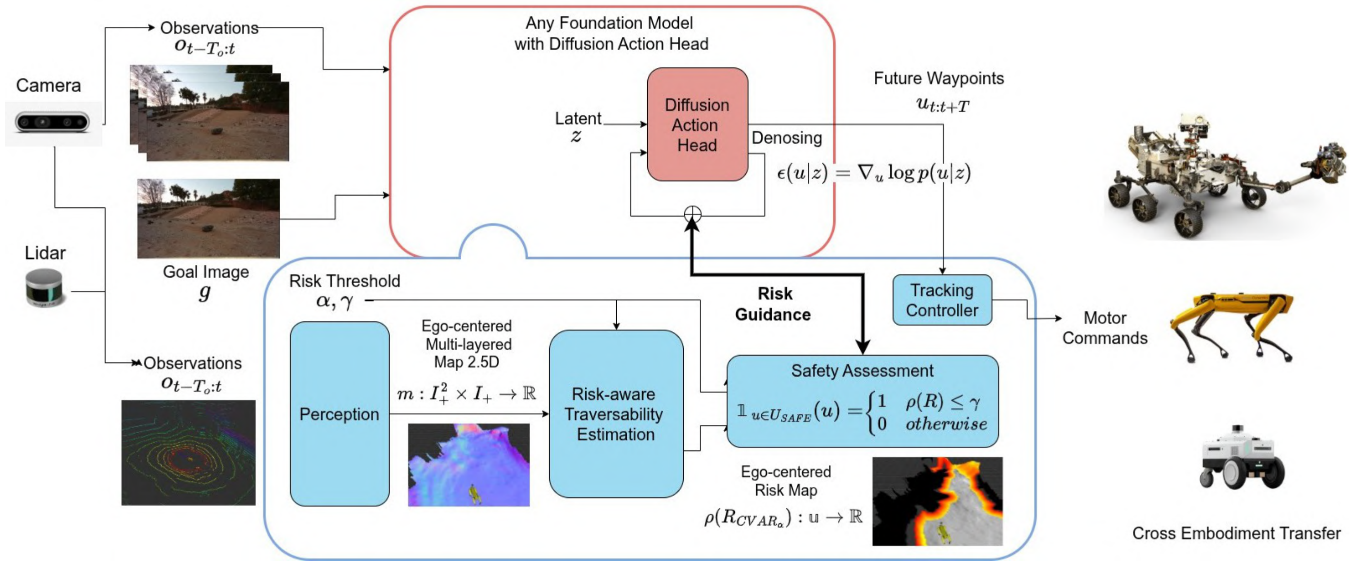


Fig. 2: Risk-Guided Diffusion Architecture Overview.

a different size and traversability capability. To address this, we leverage a diffusion-based action head to learn a multi-modal action distribution that represent different homotopy classes of candidate trajectories. (3) We leverage inference-time compute with physics-based models to improve performance without additional training data. Using an intuitive 1-D example, we show that standard classifier guidance [18] is insufficient for constraint satisfaction and propose an extension of a projection-based guidance strategy [19]. (4) Experimental validation in NASA JPL's Mars-analog facility shows our approach reduces failure rates up to 4× without reducing the goal-reaching performance of learning-based models.

II. Approach

Inspired by Kahneman [1], this section presents a framework that combines a fast, intuitive "System 1" (pre-trained foundation model) with a slow, deliberate "System 2" (physics-based risk estimator).

A. System 1: Learning-based Foundation Model

Mathematically, System 1 is a pre-trained control policy $\pi(u|o, g)$ that outputs N_u waypoints $u_{N_u} \in \mathbb{R}^{2\times N_u}$ given N_o image observations o_{N_o} and goal image o_g. Following [16], we encode high-dimensional observations in a latent representation z. The policy uses a latent diffusion model $\pi(u|z)$ defined by the score function $\epsilon_\theta(u^t, t|z)$, which is trained with standard techniques [20],

$$\min_\theta \mathbb{E}_{u,z\sim\mathcal{D},t\sim Uniform(0,T)}[||\epsilon - \epsilon_\theta(u^t, t|z)||^2], \quad (1)$$

$$u^t = \sigma^t u^0 + \alpha^t \epsilon, \quad \epsilon \sim N(0, I), \quad (2)$$

where $u^T \sim \mathcal{N}(0, I)$, $u^0 \sim \pi(u|z)$, and α^t, σ^t are noise schedule parameters. We use $(.)_{t_{trajectory}}^{t_{diffusion}}$ notation for diffusion vs. trajectory time.

This system has three key features: (1) it uses train-time compute to learn a foundation model that excels at intuitive reasoning in high-dimensional observation spaces like images; (2) diffusion action-heads provide flexibility in learning multi-modal action distributions, capturing multiple homotopy classes; (3) the foundation model can leverage data from multiple robot embodiments of different sizes and traversability capabilities. However, System 1 is a black-box model, which cannot provide any safety guarantees.

B. System 2: Physics-based Risk Estimation

System 2 complements System 1 by providing interpretable safety guarantees for our specific robot and task. Like the thoughtful, deliberate System 2 described in [1], our System 2 leverages inference-time compute to estimate the risk associated with given actions.

Specifically, we use a physics-based stochastic traversability estimate [7] to create risk maps from ego-centric $2.5D$ maps m. This estimate is based on a traversability cost $r = \mathcal{R}(m, x, u)$ representing the degree to which the robot can traverse a given state. This becomes a random variable $R : (\mathcal{M} \times \mathcal{U}) \to \mathbb{R}$ via the belief $p(m|u, o)$. We employ CVaR as a risk metric:

$$\rho(R) = \mathrm{CVaR}_\alpha(R) = \inf_{z\in\mathbb{R}} \mathbb{E}\left[z + \frac{(R - z)_+}{1 - \alpha}\right] \quad (3)$$

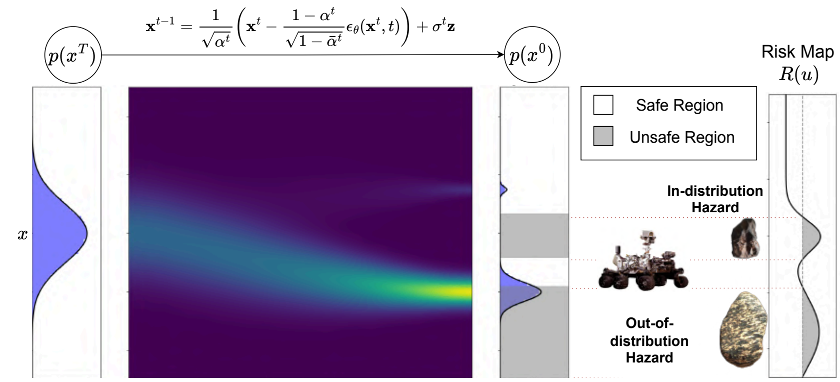
where $(.)_+ = max(., 0)$ and $\alpha \in (0, 1]$ denotes the risk probability level. Finally, we obtain a safe set of control inputs by obtaining a user-defined risk-tolerance threshold γ as follows:

$$\mathbb{1}_{u\in U_{SAFE}}(u) = \begin{cases} 1 & \rho(R(u)) \leq \gamma \\ 0 & otherwise \end{cases} \quad (4)$$

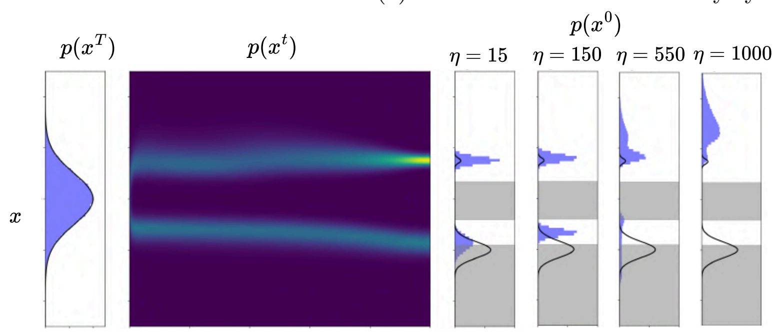
C. Thinking fast and slow: combining System 1 and 2

We develop a unified mathematical approach that integrates both systems for safe and adaptive navigation. Mathematically, this can be formulated as a constrained sampling problem:

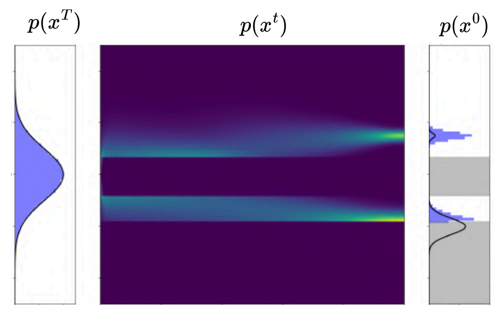
$$u \sim p(u|z) = \pi(u|z)\mathbb{1}_{u\in U_{SAFE}}(u), \quad (5)$$



(a) Diffusion model without any system 2 guidance.



(b) Classifier-based guidance [18] with varying penalty weights η



(c) Projection-based guidance [19]

Fig. 3: Langevin sampling dynamics for different guidance strategies in 1D rover navigation.

where π is the unconstrained policy of System 1's latent diffusion model.

A straight-forward approach is to use Classifier-like Risk Guidance [18] by converting $\mathbb{1}_{u \in U_{SAFE}}(u)$ to differentiable $c_{risk}(u)$ where $c_{risk}(u) \leq 0 \iff u \in U_{SAFE}$. This enables sampling from $u \sim p_{t,\theta}(u|z)e^{-\eta c_{risk}(u)}$ by modifying the diffusion score:

$$\epsilon = \underbrace{\epsilon_\theta(u, z, t)}_{\text{foundation model}} + \underbrace{\eta \nabla_u c_{risk}(u)}_{\text{risk guidance}}. \tag{6}$$

However, this penalty approach cannot guarantee constraint satisfaction since the implicit task objective $\min J(u, z)$ in $\pi(u|z) \propto \exp(-J(u, z))$ trades off with risk penalty. Parameter sensitivity is severe: low η violates constraints, while high η eliminates valid modes (Fig. 3).

These limitations can be addressed with a projection-based guidance strategy [19]. The procedure: before each diffusion step, check if the next action violates the risk constraint. If so, project to a nearby safe action; otherwise continue diffusion. This guarantees constraint satisfaction by keeping actions in the safe set. This strategy is effective (Fig. 3, last row). Unlike naive safety filters, enforcing constraints during diffusion spreads probability from constraint boundaries, encouraging exploration in informative areas. A key challenge in applying this strategy is that projecting actions to the safe set is non-unique and in general the robot motion planning problem is PSPACE-hard [21, 22].

Algorithm 1 Risk-Guidance Algorithm

Require: Current sample u^t, risk map $\mathbb{1}_{u \in U_{SAFE}}(u)$

if $t > t_2$ **then**
    while $c_{risk}(\hat{u}, z, t) > 0$ **do** ▷ Rejection sample
        $w \sim \mathcal{N}(0, I)$
        $\hat{u} \leftarrow \frac{1}{\sqrt{\alpha^t}}\left(u^t - \frac{1-\alpha^t}{\sqrt{1-\bar{\alpha}^t}}\epsilon_\theta(u^t, z, t)\right) + \sigma^t w$
    end while
else if $t > t_1$ **then**
    while $c_{risk}(\hat{u}, z, t) > 0$ **do**
        $\hat{u} \leftarrow (1-\alpha)\hat{u} + \alpha u^t$ ▷ Previous projection
    end while
else
    while $c_{risk}(\hat{u}, z, t) > 0$ **do**
        $\hat{u} \leftarrow (1-\alpha)\hat{u}$ ▷ Small-action projection
    end while
end if
return u^{t-1}

We address these challenges by employing three domain-specific projection strategies, as outlined in Algorithm 1. Rejection sampling: draws new noise values w until constraints are satisfied—this is inefficient for narrow passages but effective in early iterations. Previous projection: assuming the robot started in the safe set and no actions enter unsafe set during the diffusion process, the previous action from the diffusion process will also be safe. Small action projection: exploits the fact that

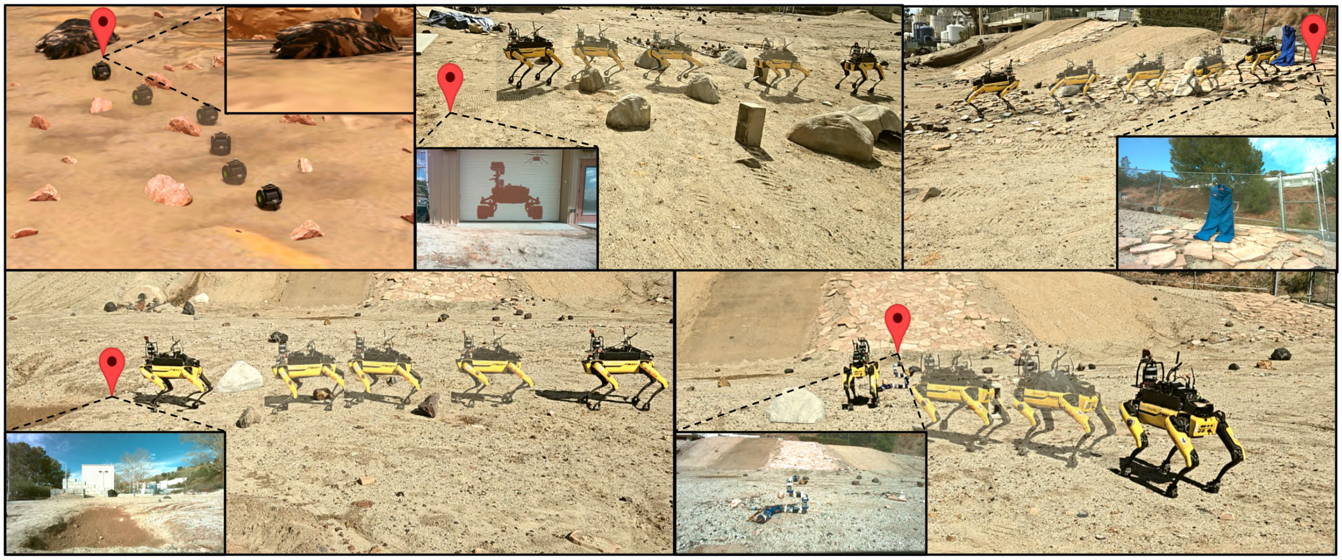


Fig. 4: Test setups: simulation (top-left) and Mars Yars—Hard (top-right) Easy (bottom) with goal images.

Approach	Mars Sim - Carter	
	Safety Failure (%) ↓	Goal Success (%) ↑
Vanilla NoMaD	100	0
NoMaD Finetuned	50	46.67
NoMaD with Safety Filter	0	46.67
Our Method	0	50

TABLE I: Simulation environment

Approach	Mars Yard - Spot (Easy)		Mars Yard - Spot (Hard)	
	Safety Failure (%) ↓	Goal Success (%) ↑	Safety Failure (%) ↓	Goal Success (%) ↑
Vanilla NoMaD	100	0	100	0
NoMaD Finetuned	30	70	48.57	25.92
Our Method	6.67	66.67	14.28	23.07

TABLE II: Real world environment

the robot remains safe if stationary.

III. Experiments & Results

We conduct quantitative analysis in simulation using Isaac Sim with Carter robot and at the JPL Mars Yard using the NeBula Spot robot [23] with randomized start/goal configurations (Fig. 4). We compare against: (1) Vanilla NoMaD [16], (2) Finetuned NoMaD, (3) NoMaD with safety filter, and (4) Risk-guided diffusion (Ours). Implementation details are provided in the appendix. Our primary metrics are Goal Success rate (GS) and Safety Failure rate (SF). GS quantifies the percentage of trials where the robot successfully avoids obstacles and reaches the goal. SF quantifies the percentage of trials where the robot falls or steps on a high risk hazard.

Table I and II show vanilla NoMaD leads to collisions in all of our simulation and real-world test cases. This shows that such models are very brittle when it comes to out-of-distribution scenarios. Even after fine-tuning, the model is not capable of avoiding collisions in a majority of the test cases. In contrast, the safety filter and our method reduced the safety violations to 0 in simulation and by almost 4× for real world environments.

Contrary to expectations, our method only slightly outperforms the safety filter. The model's bias toward out-of-distribution obstacles—and the training data's failure to learn multimodal actions across homotopies—renders guidance no better than filtering. In cluttered scenes the robot dodges obstacles but then loses sight of the goal; with limited memory it fails to reacquire it, lowering success. Enhancing policy memory is a key next step. We expect these limitations to reduce as foundation models grow in scale and improve architecturally, our modular risk-guided diffusion framework should inherit these gains with minimal changes.

IV. Conclusion and Future Work

Risk-Guided Diffusion shows that robot navigation foundation models can be made mission-safe while retaining their semantic navigation capabilities. A simple projection step—essentially a fast collision check rather than the quadratic programs typical of CBF pipelines—lets a learned diffusion policy collaborate with a physics-based risk map, cutting safety violations up to 4× while maintaining goal success at the JPL Mars Yard.

The current gains over a basic safety filter are modest, limited by trajectory diversity and short-term memory in today's foundation models. We therefore invite the community to push these fronts—richer multimodal training, longer-horizon memory, and tighter guarantees—so that the method can mature into a dependable navigator for Mars lava tubes, the icy terrains of Europa and Enceladus, and other uncharted worlds.

References

[1] K. Daniel, Thinking, fast and slow, 2017.

[2] M. Ono et al., “To boldly go where no robots have gone before–part 1: Eels robot to spearhead a new one-shot exploration paradigm with in-situ adaptation,” in AIAA Scitech Forum, 2024.

[3] R. Thakker et al., “To boldly go where no robots have gone before–part 4: Neo autonomy for robustly exploring unknown, extreme environments with versatile robots,” in AIAA SCITECH Forum, 2024.

[4] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi et al., “Openvla: An open-source vision-language-action model,” arXiv preprint arXiv:2406.09246, 2024.

[5] O.-X. E. Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh et al., “Open x-embodiment: Robotic learning datasets and rt-x models,” arXiv preprint arXiv:2310.08864, 2023.

[6] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, “Gnm: A general navigation model to drive any robot,” in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 7226–7233.

[7] A. Dixit, D. D. Fan, K. Otsu, S. Dey, A.-A. Agha-Mohammadi, and J. W. Burdick, “Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation; results from the darpa subterranean challenge,” arXiv preprint arXiv:2303.01614, 2023.

[8] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, “Fast Traversability Estimation for Wild Visual Navigation,” in Proceedings of Robotics: Science and Systems, Daegu, Republic of Korea, July 2023.

[9] A. Setlur, N. Rajaraman, S. Levine, and A. Kumar, “Scaling test-time compute without verification or rl is suboptimal,” arXiv preprint arXiv:2502.12118, 2025.

[10] J. Liang, A. Payandeh, D. Song, X. Xiao, and D. Manocha, “Dtg : Diffusion-based trajectory generation for mapless global navigation,” IEEE International Conference on Intelligent Robots and Systems (IROS), 2024.

[11] W. Xiao, T.-H. Wang, C. Gan, and D. Rus, “Safediffuser: Safe planning with diffusion probabilistic models,” arXiv preprint arXiv:2306.00148, 2023.

[12] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” IEEE Transactions on Automatic Control, vol. 62, no. 8, pp. 3861–3876, 2016.

[13] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang et al., “Gr00t n1: An open foundation model for generalist humanoid robots,” arXiv preprint arXiv:2503.14734, 2025.

[14] A. Figure, “Helix: A vision-language-action model for generalist humanoid control,” Figure AI News, 2024.

[15] J. Yoon, H. Cho, D. Baek, Y. Bengio, and S. Ahn, “Monte carlo tree diffusion for system 2 planning,” arXiv preprint arXiv:2502.07202, 2025.

[16] A. Sridhar, D. Shah, C. Glossop, and S. Levine, “Nomad: Goal masked diffusion policies for navigation and exploration,” in IEEE International Conference on Robotics and Automation (ICRA), 2024.

[17] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” The International Journal of Robotics Research, 2024.

[18] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” Advances in neural information processing systems, vol. 34, pp. 8780–8794, 2021.

[19] J. K. Christopher, S. Baek, and N. Fioretto, “Constrained synthesis with projected diffusion models,” Advances in Neural Information Processing Systems, 2024.

[20] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in International Conference on Learning Representations (ICLR), 2021.

[21] J. Canny, “The complexity of robot motion planning,” MIT press, 1988.

[22] S. M. LaValle, Planning algorithms. Cambridge university press, 2006.

[23] A. Bouman et al., “Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion,” in IEEE International Conference on Intelligent Robots and Systems (IROS), 2020.

Appendix I
Baseline implementation details

1) Vanilla NoMaD : For the Vanilla NoMaD baseline, we utilize the publicly available pre-trained weights released by the original authors and evaluate the model in a zero-shot setting across both simulated and real-world environments. We adhere to the normalization parameters provided in the NoMaD framework and apply velocity-based unnormalization to obtain the final action outputs from the diffusion model.
2) Finetuned NoMaD : We collect approximately one hour of driving data in both simulated and real-world environments via teleoperation of the robot, following the methodology employed in the open-source datasets used by the NoMaD framework. For data preprocessing, we adopt a distance-based waypoint sampling strategy applied across both our collected datasets and the original NoMaD datasets. Waypoints are sampled at uniform intervals of 0.2 meters, or at shorter distances when the change in heading exceeds a predefined angular threshold. Using this dataset, we first train the original NoMaD model using the same hyperparameters as the baseline, and subsequently finetune the model on our custom dataset.
3) Safety Filter: To ensure safety, we implement a simple truncation-based safety filter that truncates the output trajectory at the waypoint immediately preceding the first predicted collision. This approach guarantees that the resulting trajectory remains entirely within safe bounds.
4) Risk Guidance Diffusion : We implement the projected risk guidance mechanism as described in this work, utilizing the risk map generated using [7]. In simulation, we sample a total of 50 trajectories, while in the real-world setting, we sample 8 trajectories. Risk-guidance is performed in parallel across all trajectories to ensure fast inference.

Appendix II
Affliation and Copyright

[1]Jet Propulsion Laboratory (JPL), California Institute of Technology(Caltech), Pasadena, CA, USA

[2]Department of Civil and Mechanical Engineering, California Institute of Technology, Pasadena, CA, USA

[3]Swiss Federal Institute of Technology (ETH Zurich), Robotic Systems Lab, Switzerland

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).