Lumen: An Open, Differentiable, GPU-Parallel Environment for Wall-Safe Endovascular AI

Colin Son, MD
Seldinger, Inc.
ORCID: 0000-0002-1782-0537

July 2026

Abstract

Endovascular agents should be judged on more than whether a device reaches a target branch. They also need wall-safety accounting, reproducible cases, image observations, labels, and episode records that can be inspected after training. Lumen is an Apache-2.0 environment for this work. It provides procedural vascular cases, tube-intrinsic contact, Newton/Warp execution paths, Gymnasium navigation tasks, synthetic fluoroscopy, luminal image generation, and replayable computer-vision datasets. This preprint describes the public release, reports the `lumen-bench/2` launch benchmark, and adds a matched branch-navigation PPO comparison against CathSim. In the 50,000-step PPO comparison, Lumen and CathSim each achieved 100% raw success on their native tasks, with 79.7 versus 12.1 evaluation steps per second. The archived safety endpoints are not commensurate: Lumen used centerline penetration in simulator units, while CathSim used native contact force, so no cross-environment safety rate is claimed. The release is intended to make native safety telemetry, simulator-derived labels, and reproducible endovascular RL results easier to audit.

1 Motivation

Autonomous catheter and guidewire navigation is now an active research direction for robotic endovascular intervention. Reinforcement learning, imitation learning, inverse reinforcement learning, and expert-trajectory learning have all been tested in simulated or physical settings [4, 5, 1, 2, 3, 6]. CathSim showed that an open simulator could support autonomous catheterization experiments with robot/device simulation and reinforcement-learning tasks [1, 2]. SOFA and BeamAdapter represent another important line of work: finite-element flexible-device modeling for interventional-radiology simulation and mechanical-thrombectomy navigation studies [7, 8, 5].

Lumen is built for the part of the problem that is easy to blur in a demo: scoring the run after it is over. It treats target reach and wall-safe target reach as different outcomes. A policy that reaches a branch after unsafe wall interaction is recorded differently from a policy that reaches the same branch cleanly. The same run can also produce image observations, masks, keypoints, and replay metadata, so the result can be checked outside the RL loop.

2 System Overview

Lumen is organized around five parts:

1. **Procedural vascular cases.** Tube, stenotic, tortuous, and branching vascular cases are generated as replayable scenes, enabling deterministic reruns and benchmark submissions.

2. **Tube-intrinsic contact and wall state.** Device progress is represented along the vessel manifold. Route progress, contact, penetration, torsion, and friction hooks are reported by the environment.

3. **GPU-parallel execution.** Newton/Warp-style kernels support high-throughput differentiable simulation paths for calibration and batched environment work [13].

4. **Observation and dataset generation.** Fluoroscopy, luminal RGB, segmentation masks, keypoints, device masks, and detector-noise variants are generated from the same scene.

5. **Gymnasium-compatible RL tasks.** Navigation tasks are registered through the Gymnasium interface, allowing standard reinforcement-learning tooling to treat Lumen as a conventional environment [14, 15].

This follows the practical pattern used by general RL benchmarks [14, 15], physics-backed control suites [22, 23], and medical-simulation frameworks [7, 9, 10], but the task definitions and scorecards are specific to endovascular navigation.

3 Observation Layer

Lumen emits both state observations and image observations from the same case. A single scene can produce biplanar digitally reconstructed radiographs, masks for vessel and device geometry, labeled keypoints, noisy fluoroscopy, and a luminal RGB camera view. These outputs are meant for reinforcement learning, computer-vision pretraining, perception debugging, and dataset construction.

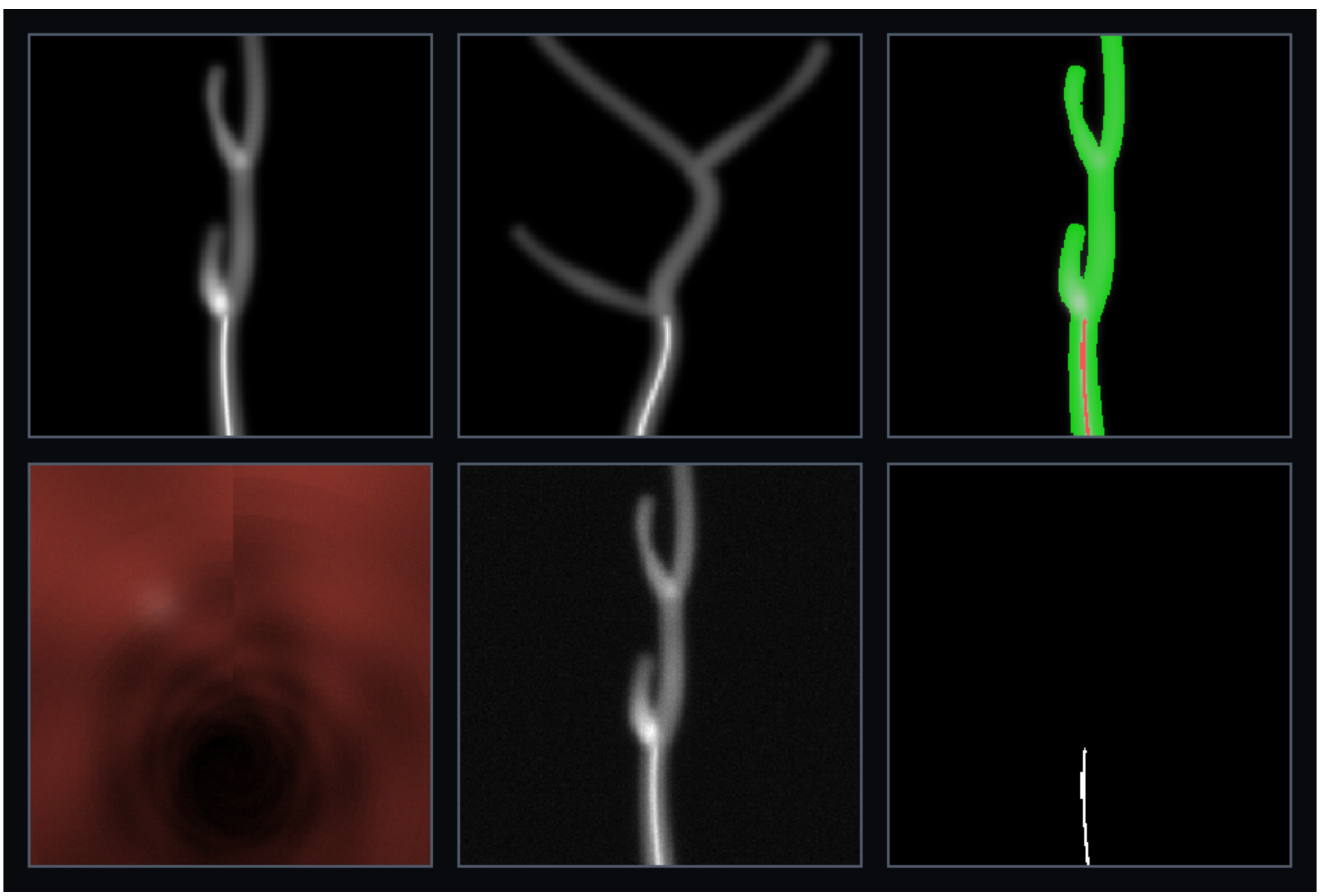


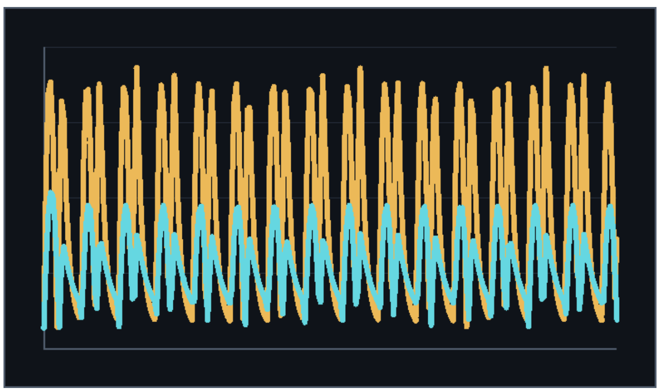
Figure 1: Multi-modal scene outputs from one Lumen case: biplanar fluoroscopy, vessel/device labels, luminal RGB, detector noise, and masks. The same geometry drives control state and image data.

The dataset workflow includes command-line capture, validation, indexing, and train/validation/test splitting. Episode sidecars record label, mask, and replay metadata before the dataset is consumed by a model.

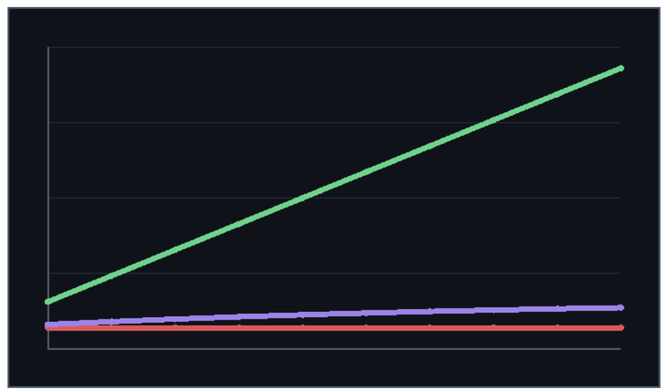
4 Physics and Device Modules

Lumen includes reduced-order modules for events that should affect policy behavior:

- deformable-wall semantics inspired by arterial-wall mechanics and anisotropic constitutive modeling [11, 12];
- contact and penetration metrics reported in environment state;
- tortuosity, route progress, and branch-target scoring;
- flow-diverter effects and aneurysm inflow traces;
- clot-field interaction, stentriever retrieval, fragmentation, and damage bookkeeping;
- synthetic fluoroscopy and luminal rendering for image-observation agents.



(a) Aneurysm inflow before and after a flow-diverter state change.



(b) Stentriever retrieval, captured clot, and damage traces.

Figure 2: Simulator-native advanced module outputs from the current implementation. These are raw Lumen captures, not website screenshots or generated promotional imagery.

These modules keep more of the procedure in the recorded state. Contact, flow, clot/device state, image formation, and replay metadata can be captured together and reproduced from case metadata.

5 Benchmark Protocol and Results

The following subsection is a **archived v2 result**, retained only for provenance. The launch benchmark `lumen-bench/2` evaluated three fixed procedural tasks: `nav_tube`, `nav_stenotic`, and `nav_tree_branch`, with five seeded episodes per task. Its 0.3 simulator-unit threshold and reported 0.178 value used centerline-based penetration semantics. They are not current v3 scores and must not be reinterpreted after the finite-device-radius change. The v3 contract reports device-surface overlap and native wall-load curves in simulator units; a physical calibration remains unresolved.

Figure 3: Navigation rollouts in branching and tortuous vessels. Lumen reports route progress, target reach, wall penetration, and safety status so benchmark results can distinguish safe target reach from unsafe reach.

Table 1: Archived `lumen-bench/2` result for the deterministic `forward-baseline` policy. The scorecard is included with the release artifacts for provenance only; its centerline-based penetration semantics are superseded by v3.

Task	Tier	Episodes	Safe success	Unsafe success	Mean steps	Max pen.
`nav_tube`	Easy	5	1.00	0.00	18.6	0.000
`nav_stenotic`	Medium	5	1.00	0.00	19.8	0.000
`nav_tree_branch`	Hard	5	1.00	0.00	54.0	0.178
Overall	–	15	1.00	0.00	–	0.178

The forward baseline is intentionally simple: it advances the proximal end at full rate. It is useful as a reproducibility check and a floor for trained policies, not as a ceiling. Standard reinforcement-learning algorithms such as PPO and SAC can be applied through the Gymnasium interface [16, 17, 18], while leaderboard submissions preserve per-task scorecards for comparison.

The historical launch comparison was a matched branch-navigation PPO run with 50,000 training steps, seed 0, and 30 deterministic episodes. Lumen used `nav_tree_branch`; CathSim used `phantom3_bca`. Both recorded 100% raw target reach on their native tasks. The safety observations are not commensurate: Lumen's archived result used centerline penetration in simulator units, whereas CathSim recorded contact force in its native units. The current comparison contract therefore reports both curves and endpoint labels but does not publish a cross-environment safety rate. The historical rows below are retained only for provenance and must not be read as a calibrated safety comparison.

Table 2: Historical 50,000-step PPO run. Raw reach and throughput are shown; native safety endpoints are labelled but intentionally not collapsed into a cross-environment rate.

Environment	Task	Raw success	Native safety endpoint	Eval steps/s
Lumen	`nav_tree_branch`	1.00	centerline penetration (sim units)	79.7
CathSim	`phantom3_bca`	1.00	contact force (native units)	12.1

The same external-comparison harness also ran 30-episode pilot policies across Lumen, CathSim, and the SOFA/stEVE Docker stack. These historical records retain each environment's native endpoint curves and labels; they do not provide a shared safety rate. Lumen forward and sweep policies reached 100% raw success on all five native Lumen navigation tasks. CathSim forward and sweep pilots reached 0% success on the two `phantom3` branch targets. SOFA/stEVE executed BasicWireNav and ArchVariety without crashes; its best pilot rates were 20% and 13.3%, respectively. The raw per-episode records are included with the release artifacts.

6 Comparison to CathSim and SOFA/BeamAdapter

CathSim, SOFA/BeamAdapter, and Lumen occupy related but distinct positions in the open endovascular-simulation landscape. CathSim is the closest open autonomous-catheterization benchmark reference, with MuJoCo-based robot/device simulation, force sensing, and reinforcement-learning tasks [1, 2]. SOFA/BeamAdapter is a mature flexible-instrument simulation framework for catheters, guidewires, and coils, and has been used in SOFA mechanical-thrombectomy navigation studies [7, 8, 5]. Lumen focuses on a public benchmark path where wall safety, image labels, replayable datasets, and trained-agent scorecards are emitted as ordinary artifacts of a run.

Table 3: Capability positioning of Lumen against commonly cited open endovascular-simulation references.

Surface	Lumen	CathSim	SOFA/BeamAdapter
Primary target	Wall-safe AI benchmark and data generation	Autonomous catheterization simulator and RL tasks	Flexible catheter/guidewire mechanics
RL interface	Gymnasium tasks plus portable scorecards	PPO/SAC-style RL evaluation reported	Used by navigation studies; no common leaderboard
Safety endpoint	Native surface-overlap and wall-load curves in simulator units; no cross-environment aggregate	Native contact-force curve; no calibrated shared threshold	SOFA-scene contact mechanics
Imaging and labels	Fluoro, luminal RGB, masks, keypoints, noisy detector output	Visual simulation and segmentation surfaces	Scene-dependent rendering and mechanics
Geometry	Replayable tube, stenotic, tortuous, and branch cases	Aorta/robot simulation assets	User-authored SOFA scenes and vascular models
Release emphasis	Reproducible benchmark artifacts, CV-ready captures, Apache-2.0	Open autonomous catheterization simulator	Open mechanics framework and plugin ecosystem

The practical contribution is the packaging: native endpoint reporting, procedural task reproducibility, multi-modal labels, trained-agent evaluation, and replay artifacts are available in one public release. A physical force–injury calibration and a shared cross-environment safety endpoint remain open validation work.

7 Availability

The public repository is github.com/SeldingerMed/seldinger-lumen. The release includes the benchmark scorecard, simulator-native figures, launch video, source LaTeX, generated PDF, and raw comparison outputs.

8 Conclusion

Lumen provides an open endovascular AI environment where vessel navigation, wall safety, synthetic imaging, labels, and replayable benchmark artifacts are produced from the same run. The launch benchmark establishes a reproducible baseline for safe target reach, and the external comparison records how Lumen, CathSim, and SOFA/stEVE behave under the same pilot and PPO evaluation harnesses.

References

[1] T. Jianu, B. Huang, M. E. M. K. Abdelaziz, M. N. Vu, S. Fichera, C.-Y. Lee, P. Berthet-Rayne, F. Rodriguez y Baena, and A. Nguyen. CathSim: An open-source simulator for endovascular intervention. *arXiv:2208.01455*, 2022.

[2] T. Jianu, B. Huang, M. E. M. K. Abdelaziz, M. N. Vu, S. Fichera, C.-Y. Lee, P. Berthet-Rayne, F. Rodriguez y Baena, and A. Nguyen. CathSim: An open-source simulator for endovascular intervention. *IEEE Transactions on Medical Robotics and Bionics*, 2024.

[3] T. Jianu, B. Huang, T. Vo, M. N. Vu, J. Kang, H. Nguyen, O. Omisore, P. Berthet-Rayne, S. Fichera, and A. Nguyen. Autonomous catheterization with open-source simulator and expert trajectory. *arXiv:2401.09059*, 2024.

[4] H. Robertshaw, L. Karstensen, B. Jackson, H. Sadati, K. Rhode, S. Ourselin, A. Granados, and T. C. Booth. Artificial intelligence in the autonomous navigation of endovascular interventions: A systematic review. *Frontiers in Human Neuroscience*, 17, 2023.

[5] H. Robertshaw, L. Karstensen, B. Jackson, A. Granados, and T. C. Booth. Autonomous navigation of catheters and guidewires in mechanical thrombectomy using inverse reinforcement learning. *International Journal of Computer Assisted Radiology and Surgery*, 2024.

[6] Y. Liu et al. Image-guided autonomous guidewire navigation in robot-assisted endovascular intervention. *arXiv:2403.05748*, 2024.

[7] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni. SOFA: An open source framework for medical simulation. *Studies in Health Technology and Informatics*, 125:13–18, 2007.

[8] SOFA Consortium. BeamAdapter: Kirchhoff-rod finite-element simulation for catheters, guidewires, coils, and interventional-radiology deployment. Software documentation and source package, 2026.

[9] A. Updegrove, N. M. Wilson, J. Merkow, H. Lan, A. L. Marsden, and S. C. Shadden. SimVascular: An open source pipeline for cardiovascular simulation. *Annals of Biomedical Engineering*, 45:525–541, 2017.

[10] R. Izzo, D. Steinman, S. Manini, and L. Antiga. The Vascular Modeling Toolkit: A Python library for the analysis of tubular structures in medical images. *Journal of Open Source Software*, 3(25):745, 2018.

[11] G. A. Holzapfel, T. C. Gasser, and R. W. Ogden. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *Journal of Elasticity*, 61:1–48, 2000.

[12] T. C. Gasser, R. W. Ogden, and G. A. Holzapfel. Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. *Journal of the Royal Society Interface*, 3(6):15–35, 2006.

[13] NVIDIA. Warp: A Python framework for high-performance simulation and spatial computing. Software documentation, 2026.

[14] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *arXiv:1606.01540*, 2016.

[15] M. Towers et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv:2407.17032*, 2024.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

[17] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[18] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor-critic algorithms and applications. *arXiv:1812.05905*, 2018.

[19] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 2nd edition, 2018.

[20] V. Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[21] T. P. Lillicrap et al. Continuous control with deep reinforcement learning. *International Conference on Learning Representations*, 2016.

[22] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[23] E. Coumans and Y. Bai. PyBullet, a Python module for physics simulation for games, robotics and machine learning. Software reference, 2021.