Uncertainty-Aware Autonomous Robotic Inspection Based on Active Vision and Deep Reinforcement Learning

WEN TANG and MOHAMMAD R. JAHANSHAHI

ABSTRACT

This paper introduces an uncertainty-aware active vision framework, UADS-DRL, for autonomous robotic damage inspection. Unlike conventional methods that rely on passive raster scanning or prior learning-based agents, UADS-DRL formulates the task as a partially observable Markov decision process, enabling dynamic viewpoint selection guided by segmentation uncertainty. The agent leverages uncertainty cues to fuse predictions from multiple time steps, thereby achieving robust segmentation. Experimental results on photorealistic metallic surface inspection show that UADS-DRL achieves more than 2 times improvement in damage mIoU and over 50% reduction in inspection time compared to dense overlap raster scanning. It also surpasses the earlier ADS-DRL baseline by 20% in damage IoU with comparable inspection time. Given similar IoU performance, the proposed UADS-DRL agent can cut the inspection time by more than half compared with the previous learning baseline. These results underscore the effectiveness of integrating uncertainty estimation for accurate and time-efficient robotic inspection.

INTRODUCTION

Timely and accurate damage detection is critical for ensuring the safety and longevity of civil infrastructure, helping to prevent catastrophic failures. With advances in artificial intelligence, robotic platforms have increasingly been deployed for structural health monitoring (SHM) across diverse environments, including bridges, tunnels, buildings, and energy facilities. However, the majority of these systems have emphasized hardware design—such as locomotion and sensing—while relying on human operators or limited autonomy [1–3]. Even among autonomous systems, inspection is typically framed as a coverage or navigation problem, where the robot exhaustively scans the environment without considering the informativeness of each observation [4–6]. These approaches follow a passive perception paradigm, in which data is collected along predefined paths regardless of the presence or absence of damage. This not only leads to inefficient and redundant data collection but also hinders the system's ability to handle uncertain or ambiguous cases during analysis—especially since most data is processed offline, leaving

Wen Tang, PhD Student, Email: tang493@purdue.edu. Smart Informatix Laboratory (SIL), Lyles School of Civil and Construction Engineering, Purdue University, West Lafayette, IN, USA

no opportunity for the robot to revisit areas of interest.

In contrast, human inspectors inherently adopt an active perception strategy: they move around, adapt viewpoints, and focus on suspicious regions to improve their understanding. Emulating this behavior in robotic systems could significantly enhance inspection performance. Damage is often partially visible, occluded, or dependent on viewpoint due to lighting or structural geometry, making adaptive sensing essential. An active perception framework enables robots to make informed decisions based on uncertainty and prior observations, dynamically adjusting their trajectory to seek out the most informative views. Despite its potential, active perception remains underexplored in SHM applications. This work aims to address that gap by developing an uncertainty-aware active inspection framework that empowers robotic systems to detect damage more reliably and efficiently.

Building upon prior work on Active Damage Segmentation (ADS) [7], this study introduces several key advancements in both algorithm design and system capabilities:

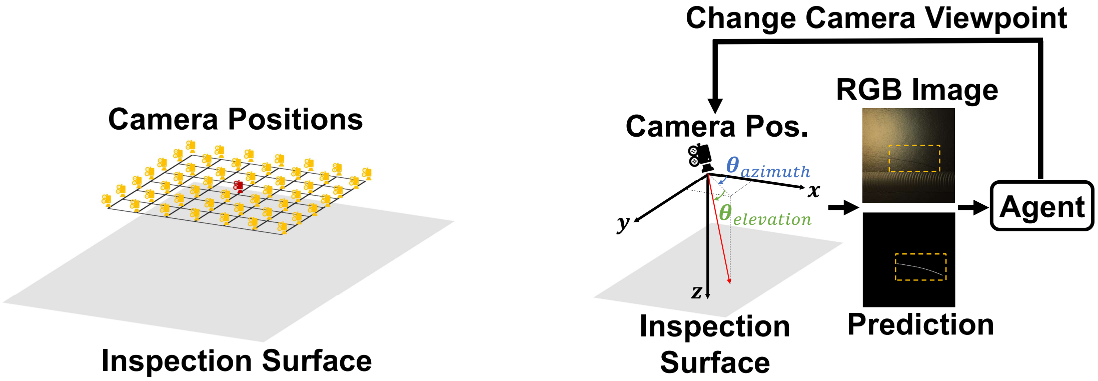
- Predictive uncertainty is estimated via entropy over softmax scores from an ensemble of segmentation models. This information is used to guide exploration and refine segmentation decisions.
- The 3D simulation environment used in prior studies is further expanded and refined to support multi-class segmentation.

ACTIVE DAMAGE SEGMENTATION

Active Damage Segmentation (ADS) requires an agent to inspect regions with minimal segmentation errors, focusing on accurate crack/scratch classification on metallic surfaces. The agent begins with raster scanning for complete surface coverage, switching to active perception when ambiguity is detected—defined as over 200 pixels falling within the confidence range [0.3, 0.7] in the segmentation softmax map. Upon activation, the agent observes $o_{t=1} = (I_{t=1}, \mathcal{V}_{t=1})$, where $I_{t=1}$ is an RGB image and $\mathcal{V}_{t=1}$ encodes a 6D camera pose $[x, y, z, \theta, \phi, r]$. The agent proposes a viewpoint sequence using a learned policy π, acquiring new frames I_{t+1}, generating segmentation maps M_{seg}^{t+1}, and incrementally refining a fused map M_f^{t+1}. This process continues until either segmentation errors fall below a threshold or a maximum time horizon T_{max} is reached. The agent then resumes raster scanning for further exploration.

The problem is framed as a partially observable Markov decision process (POMDP). At each step t, the agent receives an observation o_t derived from the true state s_t, selects an action $a_t \sim \pi(a_t \mid o_{1:t})$, receives a scalar reward r_t, and transitions to s_{t+1}. The trajectory $\tau = (o_t, a_t, r_t)$ accumulates a return $R_T = \sum_{t=1}^{T} \gamma^{t-1} r_t$, and the goal is to find an optimal policy $\pi^* = \arg\max_\pi \mathbb{E}_{\tau\sim\pi}[R_T]$. Proximal Policy Optimization (PPO) [8] is used to train the policy, enabling the agent to sequentially acquire informative viewpoints that enhance segmentation accuracy.

Each observation includes an RGB image I_t and a camera pose vector $[x, y, z, \theta, \phi, r]$, which are projected into a global coordinate system using the pinhole camera model. To facilitate learning across varying viewpoints, this pose vector is embedded using NeRF-style sinusoidal encoding: $\gamma(u) = [\sin(2^0\pi u), \cos(2^0\pi u), \ldots, \sin(2^{L-1}\pi u), \cos(2^{L-1}\pi u)]$.



(a) (b)

Figure 1. (a) Visualization of yellow candidate viewpoints and red current viewpoint at each time step, and (b) description of the interactive environment and the discretization of the camera movements. The camera can move transitionally in the x-y plane. At each time step, the ADS-DRL agent selects the next viewpoint to visit based on the received RGB image and the historical information.

The full encoded pose is given by $\gamma(v) \in \mathbb{R}^{12L}$, capturing both low- and high-frequency spatial variations.

The continuous viewpoint locations are discretized into discrete locations. The agent is able to visit one of the viewpoints at each time step. If the agent concludes that it has identified a sufficient number of viewpoints to differentiate the uncertain damage shown in the initial frame, it may choose to terminate the episode early by selecting the $Terminate$ action. The entire action space $\mathcal{A}$ comprises a set of 48 discrete viewpoints around the current viewpoint and an additional $Terminate$ action as shown in Fig. 1a.

The reward function encourages the agent to choose viewpoints that improve segmentation accuracy while discouraging unnecessary movement. The reward is defined as the difference in confidence-weighted correct predictions across time steps: $r_t = \sum_{p \in \Omega} \mathbb{I}(p \in \mathcal{C})[S(p,t) - S(p,t-1)] - \text{Penalty}(a_t)$, where $\mathcal{C}$ is the set of correctly classified pixels and $S(p,t)$ is the softmax score at pixel p. A small penalty $\text{Penalty}(a_t) < 0$ discourages redundant actions and promotes efficient inspection.

UADS-DRL

The detailed architecture of the UADS-DRL agent is shown in Fig. 2, which consists of three modules: a crack segmentation network ensemble, a mask fusion module, and an action policy.

To produce reliable, confidence-aware segmentation outputs, this study adopts a Deep Ensemble [9] approach consisting of multiple segmentation networks $\varepsilon = \{f_{\theta_k}\}_{k=1}^{K}$, each independently trained. Every network maps an input RGB image $x \in \mathbb{R}^{H \times W \times 3}$ to a pixel-wise class probability distribution over C classes, i.e., $f_{\theta_k} : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H \times W \times C}$. Each network follows the U-Net architecture, initialized with ImageNet-pretrained weights. Fine-tuning is performed on a set of 10,000 rendered RGB images containing structural damage. The training set is constructed from scenes disjoint from those used in

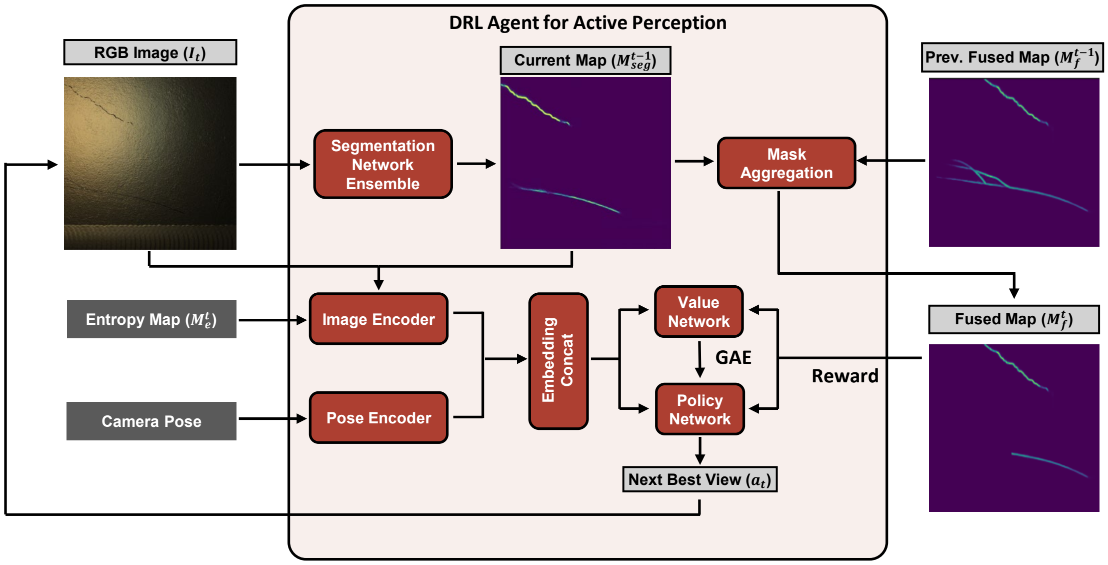


Figure 2. The network architecture of the UADS-DRL agent for active damage segmentation.

reinforcement learning to ensure generalization. Following the Deep Ensemble methodology, each model is trained independently with different random seeds, data shuffling, and bootstrapped sampling. Standard cross-entropy loss is used for supervision, and data augmentation techniques such as random flips and color jittering are applied to promote robustness.

At time step t, the agent observes an RGB image I_t, which is projected into the global frame using intrinsic and extrinsic camera parameters. The image is passed through the ensemble to obtain a set of softmax outputs:

$$\{P_k(I_t)\}_{k=1}^K, \quad \text{where } P_k(I_t) = \text{softmax}(f_{\theta_k}(I_t)) \in \mathbb{R}^{H\times W\times C}.$$

These are averaged to obtain the ensemble segmentation softmax map:

$$M_{seg}^t = \bar{P}(I_t) = \frac{1}{K}\sum_{k=1}^{K} P_k(I_t), \quad M_{seg}^t \in \mathbb{R}^{H\times W\times C}.$$

To quantify uncertainty, an entropy map $M_e^t \in \mathbb{R}^{H\times W}$ is computed from the previous fused map M_f^{t-1}, where each value corresponds to the uncertainty at a specific pixel. Let $p = (i, j) \in \Omega$ denote a pixel location in the image domain, with $\Omega = \{1, \dots, H\} \times \{1, \dots, W\}$. The entropy at pixel p is calculated as:

$$M_e^t(p) = -\sum_{c=1}^{C} M_f^{t-1}[p, c] \cdot \log M_f^{t-1}[p, c].$$

This entropy map highlights spatial regions of uncertainty and serves as a critical input for decision-making modules such as viewpoint selection and segmentation fusion.

The policy network is an actor-critic style network that takes in the current RGB frame I_t, current entorpy mask M_e^t, current segmentation masks M_{seg}^{t-1}, current camera

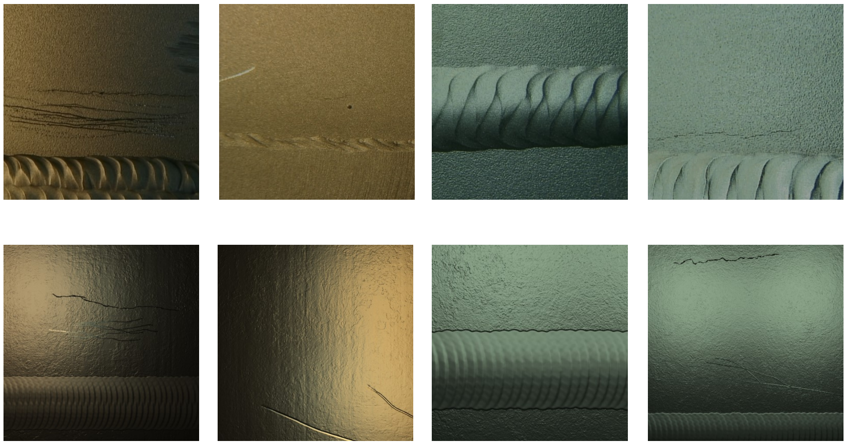


Figure 3. Comparison between images captured from field inspections and images rendered from the simulation environment. The images on the top row are from field inspection, and those on the bottom row are images rendered in the simulation environment.

pose p_{cam}^t, and previous fused masks M_f^{t-1}, and outputs a probability distribution over the action space A. During test time, the action with the largest probability is taken by the agent. The CNN feature extractor in Figure 2 uses a simple 5-layer CNN to extract the features and form the state embeddings of the DRL agent. The embeddings are then concatenated and fed into two separate branches called actor-network and action-network, which follow the convention of the actor-critic network. The actor-network and the critic network employ the same architecture, which contains a Gated Recurrent Unit (GRU) layer followed by two fully connected layers. The Generalized Advantage Estimation (GAE) is used to stabilize the variance of the expected rewards. The entire policy network is trained using PPO algorithm.

EXPERIMENTAL SETUP

The proposed UADS-DRL agent is evaluated in a photorealistic 3D simulation environment designed to replicate robotic inspection of metallic surfaces, particularly under conditions similar to underwater nuclear reactor maintenance. The environment abstracts away low-level robotic control, focusing instead on high-level viewpoint selection. At each selected viewpoint, the simulator returns an RGB image and camera pose, supporting the active perception loop central to ADS.

A total of 60 synthetic inspection scenes are generated using Houdini [10], guided by real inspection footage. These scenes include welds, scratches, grinding marks, and cracks of varying width (0.1–0.5 mm) on metallic surfaces. Each scene is illuminated in a viewpoint-dependent manner to emulate real-world lighting effects that make subtle damages, such as hairline cracks, visible only under specific conditions. The camera pose space is discretized with centers spaced 8 mm apart in the x-y plane. The camera can move up, down, left, and right in the x-y plane as shown in Fig. 1b. For each viewpoint, the azimuth angle of the camera orientation is between [0,360°] and the elevation angle is between [30°,90°], where the angles are further discretized into 30° increments. To simplify the problem, only cases in which cameras are pointed perpendicular to the

TABLE I. Quantitative performance of entire surface inspection for pure raster scanning, random policies, and various configurations of reinforcement learning (RL) agents on Active Damage Segmentation (ADS) tasks. Metrics include crack IoU (Crk. IoU), scratch IoU (Scr. IoU), entropy Reduction (Ent. Red.), and inspection time in seconds. Entropy reduction refers to the percentage reduction in softmax entropy sum, representing improved model confidence after multi-view fusion.

Methods	**Simple Scenes**					**Difficult Scenes**				
	Crk. IoU	Scr. IoU	Num. VPs	Ent. Red.	Time	Crk. IoU	Scr. IoU	Num. VPs	Ent. Red.	Time
Raster (25% overlap)	0.121	0.183	-	-	31.9	0.058	0.101	-	-	32.3
Raster (81% overlap)	0.202	0.254	-	-	101.7	0.135	0.187	-	-	102.1
Random	0.106	0.169	12.0	-2.1	49.9	0.041	0.104	12.0	-1.1	49.7
ADS-DRL	0.465	0.537	6.4	54.7	39.7	0.271	0.341	6.3	35.2	39.9
UADS-DRL	0.542	0.606	6.5	61.4	40.2	0.331	0.396	6.3	40.8	39.8

inspection surface are considered in this study. Therefore, as shown in Fig. 1a, the action space comprises a set of 48 viewpoints (denoted by the yellow cameras) surrounding the current viewpoint(denoted by the red camera), where the ADS-DRL agent can select one of the viewpoints at each time step. Upon taking an action, the corresponding image from the dataset is queried by the agent. More complex actions, such as non-perpendicular angles along with geometric transformation of the images, will be taken into account in future studies.

Each scene covers an area of roughly 219 mm × 153 mm. Of the 60 scenes, 40 are used for training and 20 for evaluation. RGB images and pixel-wise annotations are pre-rendered and indexed to support efficient RL training without real-time rendering costs.

Two levels of inspection difficulty are defined. In simpler scenes, broad illumination allows full fusion of the segmentation map. In harder scenes, limited lighting restricts visibility, requiring the agent to selectively fuse only informative sub-regions.

DISCUSSION AND COMPARISON

As for baseline comparison, different baselines are used, namely pure raster, random, and ADS-DRL [7]. For pure raster, the camera raster scans the entire surface with 80% overlap between frames. For other methods, the agent would first raster scan at 25% overlap and switch to active perception. ADS-DRL is the previous work on active damage segmentation with CNN and LSTM architecture.

Table I presents a comparison of segmentation accuracy, entropy reduction, and inspection efficiency across multiple baselines and the proposed UADS-DRL framework. Evaluation is conducted over the full inspection surface, where agents alternate between 25% overlap raster scanning and active perception upon detecting segmentation uncertainty.

Test scenes are divided into simple and difficult categories based on lighting and sur-

TABLE II. Performance of Raster, ADS-DRL, and UADS-DRL Methods Under Varying Frame Overlap Percentages on the Entire Inspection Surface for Active Damage Segmentation. Metrics include Crack IoU (Crk. IoU), Scratch IoU (Scr. IoU), and Total Inspection Time (s).

Frame Overlap (%)	Raster			ADS-DRL			UADS-DRL		
	Crk. IoU	Scr. IoU	Time	Crk. IoU	Scr. IoU	Time	Crk. IoU	Scr. IoU	Time
25	0.092	0.144	31.9	0.368	0.439	40.1	0.436	0.638	40.5
45	0.128	0.179	44.7	0.389	0.456	58.3	0.467	0.547	58.8
63	0.155	0.207	61.2	0.405	0.476	81.2	0.486	0.571	81.4
81	0.167	0.219	101.9	0.425	0.493	138.2	0.510	0.592	138.7

face visibility. In both simple scenes and difficult scenes, all RL-based agents outperform non-learning baselines, but UADS-DRL consistently achieves the highest segmentation accuracy, with crack IoU of 0.437 and scratch IoU of 0.501. It also reduces prediction entropy by 51.1%, indicating higher confidence with fewer observations.

Inspection efficiency is summarized in Table II. UADS-DRL achieves more than 2 times higher damage IoU than the 83% overlap raster baseline while cutting inspection time by more than half. Compared to ADS-DRL with 45% overlap, UADS-DRL improves IoU by 20% with similar inspection time. When comparing UADS-DRL at 25% overlap against ADS-DRL at 81% overlap, UADS-DRL delivers similar IoU performance in under half the time.

Figure 4 illustrates a set of samples comparing the prediction masks obtained through

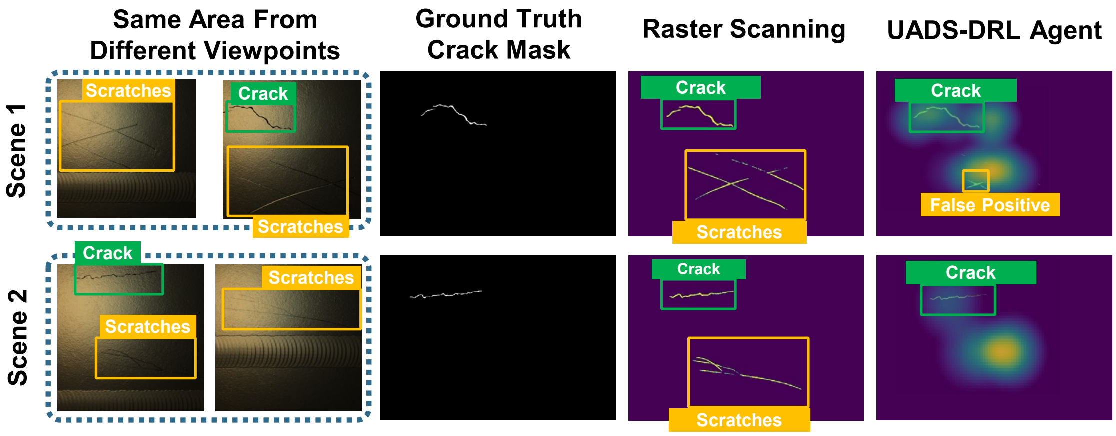


Figure 4. Sample predictions obtained from raster scanning and active damage segmentation. The highlighted area in the heatmap is where the agent looks more frequently.

CONCLUSIONS

This study introduces UADS-DRL, an uncertainty-aware active damage segmentation. Trained in a photorealistic simulation environment, the agent dynamically selects informative viewpoints and refines segmentation through entropy-guided exploration.

UADS-DRL outperforms both non-learning baselines and prior RL approaches. It improves damage mIoU by over 2× compared to dense raster scanning, while cutting inspection time by more than half. Against the previous ADS-DRL, it achieves 20% higher damage IoU with similar efficiency. Given similar IoU performance, the proposed UADS-DRL agent can cut the inspection time by more than half compared with the previous learning baseline. This work provides a robust framework for real-world robotic inspection and offers future extensions toward sim-to-real deployment in complex environments.

REFERENCES

1. Liu, Y., J. Yao, X. Lu, R. Xie, and L. Li. 2019. “DeepCrack: A deep hierarchical feature learning architecture for crack segmentation,” *Neurocomputing*, 338:139–153.
2. Mondal, T. G., M. R. Jahanshahi, R.-T. Wu, and Z. Y. Wu. 2020. “Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance,” *Struct. Control Health Monit.*, 27.
3. Chen, F.-C. and M. R. Jahanshahi. 2020. “NB-FCN: Real-Time Accurate Crack Detection in Inspection Videos Using Deep Fully Convolutional Network and Parametric Data Fusion,” *IEEE Transactions on Instrumentation and Measurement*, 69:5325–5334.
4. Narazaki, Y., V. Hoskere, G. Chowdhary, and B. F. Spencer Jr. 2022. “Vision-based navigation planning for autonomous post-earthquake inspection of reinforced concrete railway viaducts using unmanned aerial vehicles,” *Automation in Construction*, 137:104214.
5. Yoon, S., B. F. Spencer, S. Lee, H. Jung, and I. Kim. 2022. “A novel approach to assess the seismic performance of deteriorated bridge structures by employing UAV based damage detection,” *Struct. Control Health Monit.*, 29.
6. Potenza, F., C. Rinaldi, E. Ottaviano, and V. Gattulli. 2020. “A robotics and computer-aided procedure for defect evaluation in bridge inspection,” *J. Civ. Struct. Health Monit.*, 10:471–484.
7. Tang, W. and M. R. Jahanshahi. 2024. “Active perception based on deep reinforcement learning for autonomous robotic damage inspection,” *Machine Vision and Applications*, 35(5), doi:10.1007/s00138-024-01591-7.
8. Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. “Proximal Policy Optimization Algorithms,” *ArXiv*, abs/1707.06347.
9. Lakshminarayanan, B., A. Pritzel, and C. Blundell. 2017. “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Advances in neural information processing systems*, pp. 6402–6413.
10. SideFX. 2025, “Houdini 3D Animation Tools,” `https://www.sidefx.com`, version 20.5.