Robotic Seminar

Sensor Modeling in Sim2Real

Second meeting 12-Dec

Mingyuan Chi

- RGB camera
- Depth sensor
- IMU
- Force sensor
- Encoders

RGB

Simulation Manually Adjusted

Reinforcement and Imitation Learning for Diverse Visuomotor Skills

a Kinect camera (RGBD) was visually calibrated to match the position and orientation of the simulated camera, and the simulation's dynamics parameters were manually adjusted to match the dynamics of the real arm.

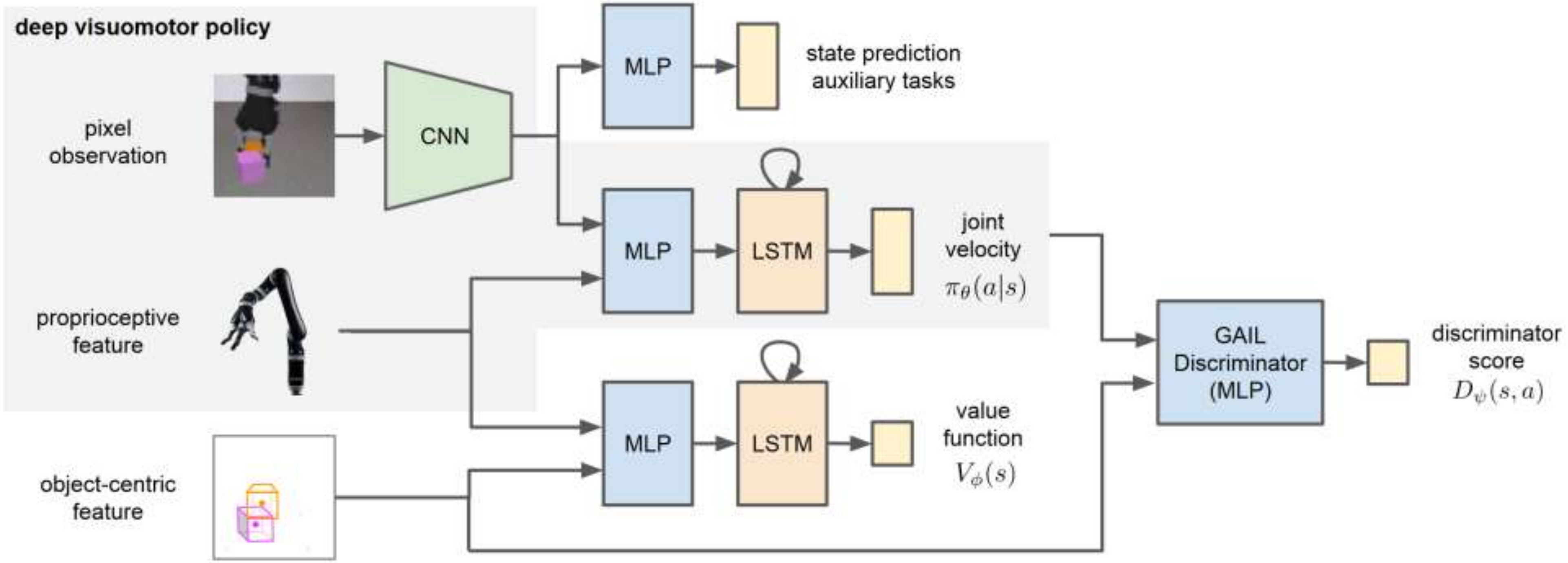


Fig. 2: Model overview. The core of our model is the deep visuomotor policy, which takes the camera observation and the proprioceptive feature as input and produces the next joint velocities.

GAIL : Generative Adversarial Imitation Learning

RGB

Neural Field

Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields

Sensor : Camera(Phone)

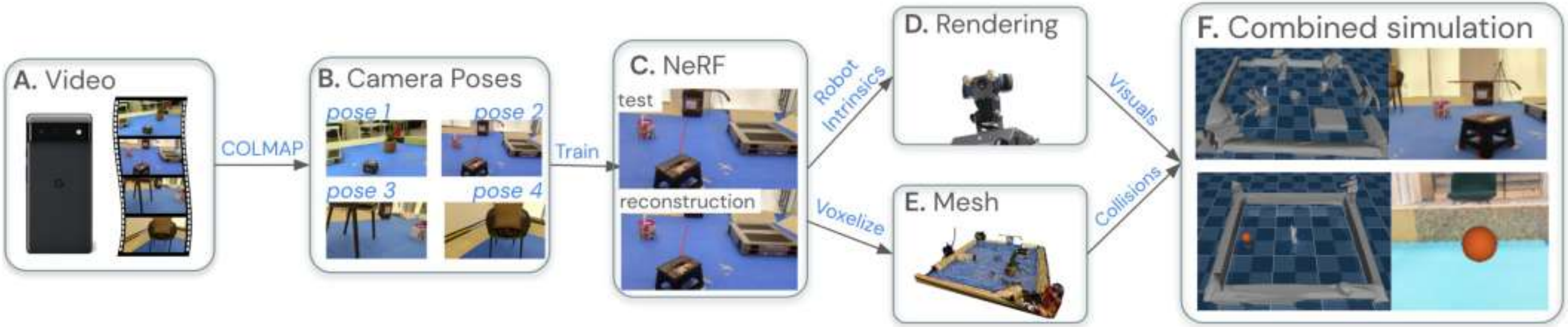


Fig. 2: Overview of our system for recreating a scene in a simulator. **A.** We collect a video of the scene using a generic phone. **B.** We use structure-from-motion software to label a subset of the video with camera poses. **C.** We train a NeRF on labeled images. **D.** We render the scene from novel views using the calibrated intrinsics of the robot's head-mounted camera. **E.** We use the same NeRF to extract the scene geometry as a mesh. We coarsen the mesh and replace the floor with a flat primitive. **F.** We combine the simplified mesh with a model of a robot, and any other dynamic objects, in a physics simulator. See Fig. 3 for further details on this step.

RGB

Neural Field

UniSim: A Neural Closed-Loop Sensor Simulator

Sensor : Camera & LiDAR

Figure 3. **Qualitative comparison.** We show simulation results in both the interpolation (rows 1, 3) and lane-shift test settings (rows 2, 4).

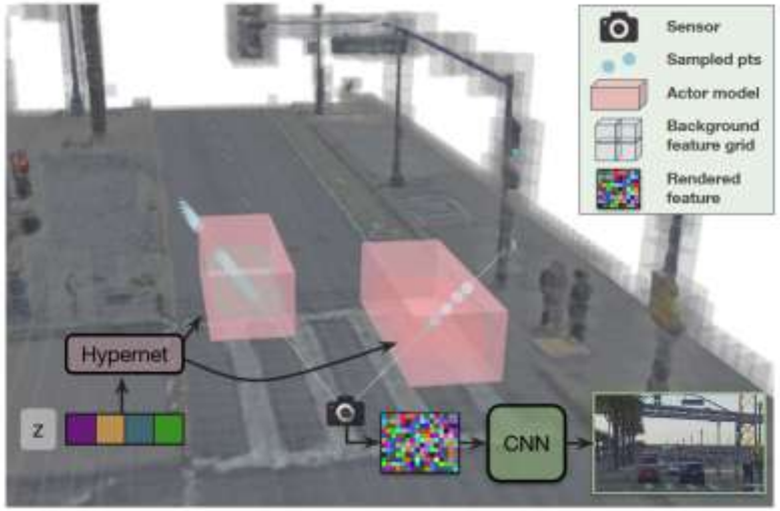


Figure 2. **Overview of our approach:** We divide the 3D scene into a static background (grey) and a set of dynamic actors (red). We query the neural feature fields separately for static background and dynamic actor models, and perform volume rendering to generate neural feature descriptors. We model the static scene with a sparse feature-grid and use a hypernetwork to generate the representation of each actor from a learnable latent. We finally use a convolutional network to decode feature patches into an image.

$$\text{Image} = \text{CNN}_{\theta_2 2}(\text{Voxel Rendering}(\text{Voxel}_{\text{static background}} + \text{Voxel}_{\text{dynamic actors},\theta_1}))$$

RGB

Synthetic Dataset

SurfelGAN: Synthesizing Realistic Sensor Data for Autonomous Driving

$$\text{Image(from Surfels)} \overset{\text{Surfel GAN}}{\leftrightarrow} \text{Images(from RGB cam)}$$

Sensor: Camera & LiDAR

Purpose : Generate realistically looking camera images (**Texture**)

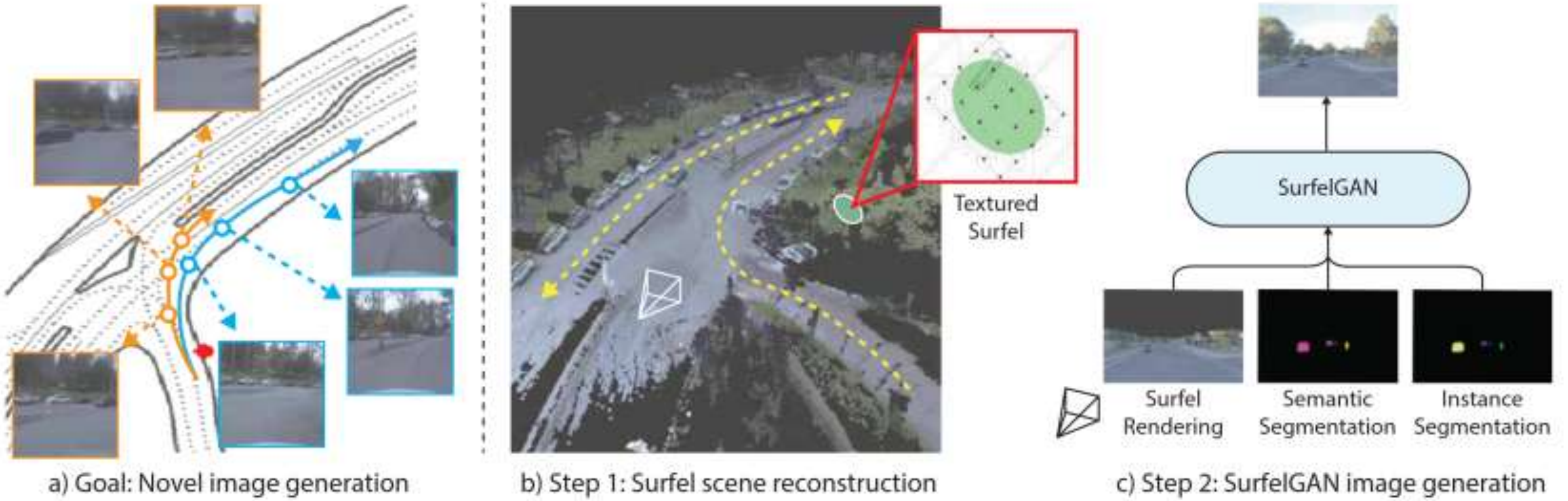
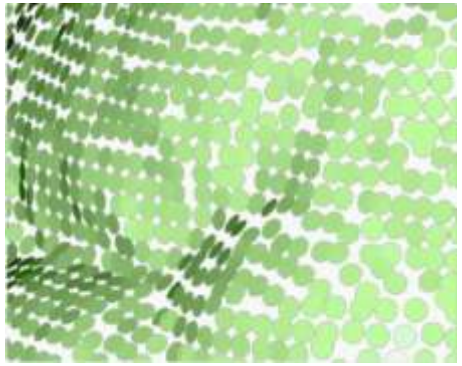


Figure 1. **Overview of our proposed system.** a) The goal of this work is the generation of camera images for autonomous driving simulation. When provided with a novel trajectory of the self-driving vehicle in simulation, the system generates realistic visual sensor data that is useful for downstream modules such as an object detector, a behavior predictor, or a motion planner. At a high level, the method consists of two steps: b) First, we scan the target environment and reconstruct a scene consisting of rich textured surfels. c) Surfels are rendered at the camera pose of the novel trajectory, alongside semantic and instance segmentation masks. Through a GAN [15], we generate realistically looking camera images.



Surfel : A surfel is a small, oriented disk used to represent a portion of a 3D surface.

RGB

Synthetic Dataset

Real2Sim2Real: Self-Supervised Learning of Physical Single-Step Dynamic Actions for Planar Robot Casting

Sensor : Logitech Brio 4K webcam

Simulator : Issac, PyBullet

Method :

1. $\mathcal{D}_{\text{phy}} = \{\text{random real interaction}\}$
2. $\theta_{\text{sim}} = \underset{\theta}{\text{argmin}}(s_{\text{real}}, s_{\text{sim},\theta})$
3. $\mathcal{D}_{sim} = \{\text{random sim interaction}\}$
4. $\pi = \text{Model}(\text{weighted combine}(\mathcal{D}_{\text{phy}}, \mathcal{D}_{\text{sim}}))$

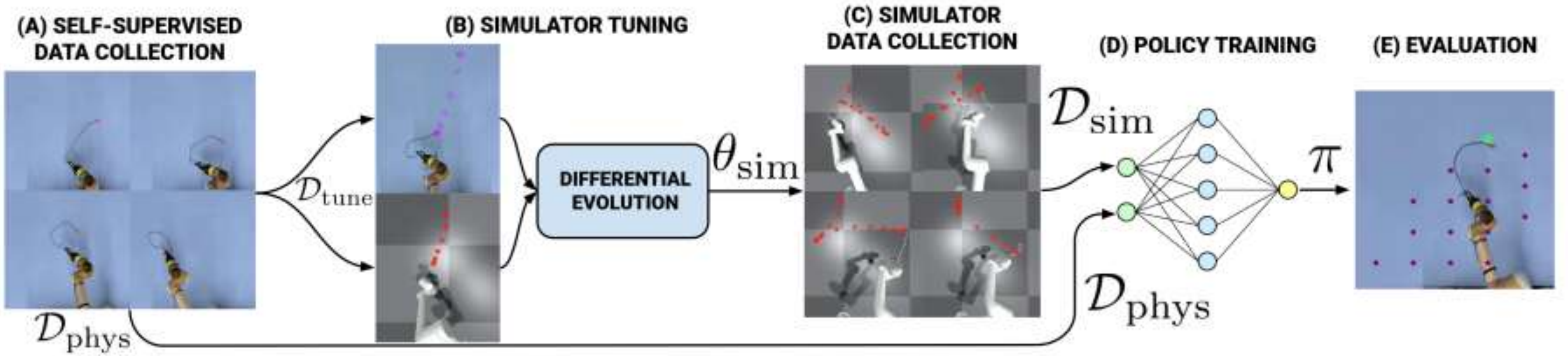


Fig. 2: The *Real2Sim2Real* pipeline for PRC. We collect a physical dataset $\mathcal{D}_{\text{phys}}$ (A) in a self-supervised manner. We subsample $\mathcal{D}_{\text{phys}}$ to generate $\mathcal{D}_{\text{tune}}$, and use it to tune simulation parameters so that its trajectories match real trajectories using Differential Evolution (B), then use the tuned simulator to generate a large dataset $\mathcal{D}_{\text{sim}}$ (C). We use a weighted combination of $\mathcal{D}_{\text{sim}}$ and $\mathcal{D}_{\text{phys}}$ to train the policy (D) and evaluate the policy in real (E).

PRC : Planar Robot Casting

Depth
Sensors

LiDAR : determining ranges by targeting an object or a surface with a laser and measuring the time for the reflected light to return to the receiver .

Kinect : the motion sensing device for the Xbox 360 gaming console. It provides RGB, Infra-Red (IR), depth, skeleton, and audio streams to an application.

Zed Camera : neural depth, built-in IMU

RealSense : long range depth camera, built-in IMU

Depth

Simulation Enhancement

Towards Zero Domain Gap: A Comprehensive Study of Realistic LiDAR Simulation for Autonomy Testing

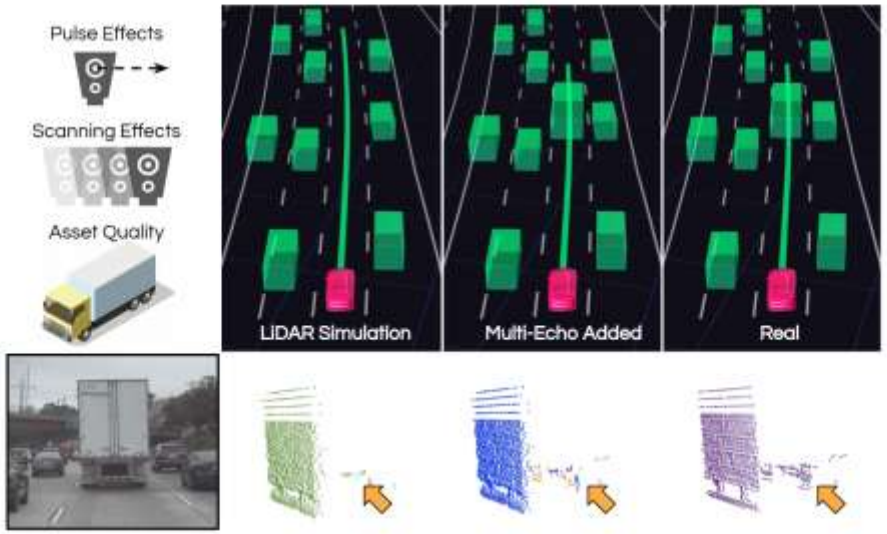


Figure 1. **Analysis Overview.** We study the impact of pulse effects, scanning effects, and asset quality on LiDAR simulation realism. We depict one example of a *pulse effect* domain gap: failure to model multiple echoes causes the object detector to fail, resulting in an unsafe autonomy plan. The bottom row depicts the front-camera view, for reference only, followed by the relevant LiDAR: original simulation, added multi-echoes, and real. We denote multi-echoes re-added by the middle method in orange. Subtle differences in the area highlighted with the arrow stem from weak returns on truck's rear wheels, impacting the domain gap.

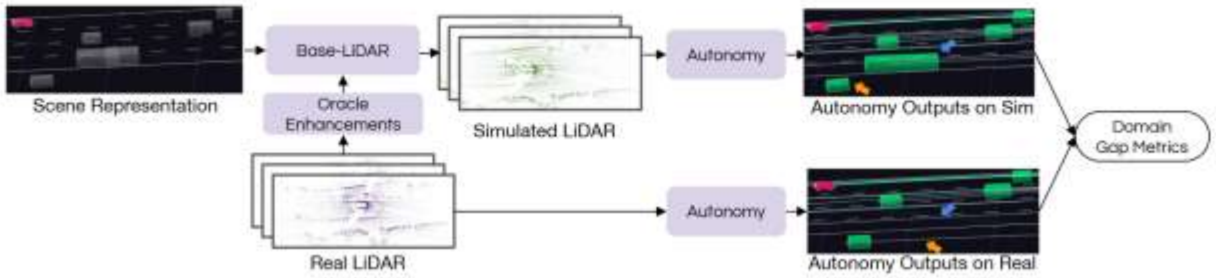


Figure 5. Given paired simulated and real LiDAR for the same scenario, we run autonomy on both in open-loop and compare the domain gap for the autonomy under test.

Modeling Phenomenons:

- drop points (pulse amplitude too low $\frac{1}{R^4}$)
- add points (different surface,...)
- spurious points (beam divergance, ...)
- noisy points (peak in waveform is ambiguous)

Depth

Domain Randomization

LiDAR Data Noise Models and Methodology for Sim-to-Real Domain Generalization and Adaptation in Autonomous Driving Perception

Sensor : LiDAR

Task : Semantic Segmentation and Object Detection

Method :

$$\text{Training Set} = \text{Error Model}(\text{Simulated Data})$$

$$\text{Error Model} = \text{Noise Model} + \text{Point Dropout Model}$$

Noise Model : Gaussian additive model

$\hat{d} \sim \mathcal{N}(d, \sigma)$

$\sigma = k_1\alpha^3 + k_2 d^2 + k_3 d\alpha + k_4\alpha + k_5 d + k_6$

Point Dropout Model : Bernoulli distribution model

$p_r = \frac{np_t - np_a}{np_t}$

$p_r = p_1\alpha^2 + p_2 d^2 + p_3 d\alpha + p_4\alpha + p_5 d + p_6$

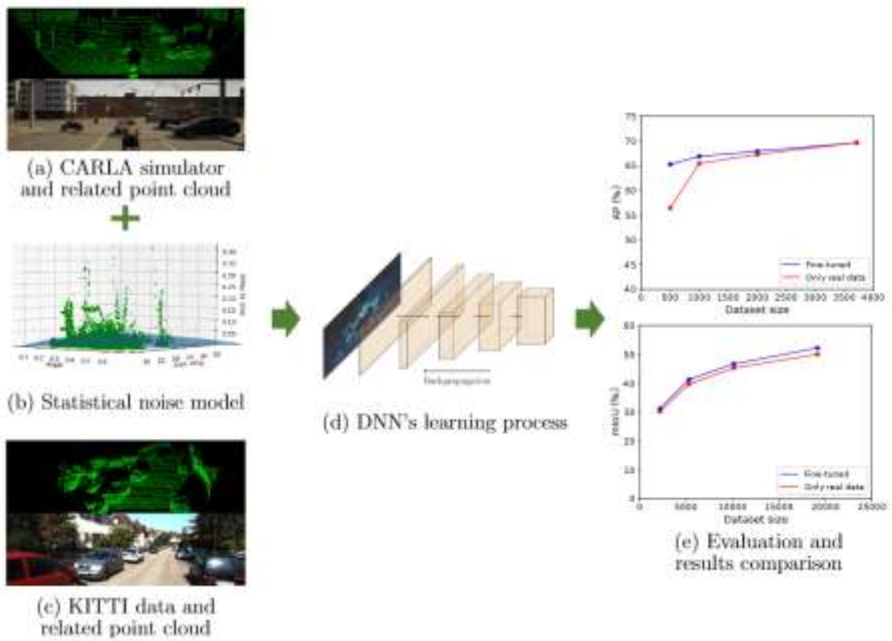


Fig. 1. Pipeline of the proposed approach for LiDAR sim-to-real domain generalization and adaptation. (a) and (b) represent, respectively, the artificially generated data and the sensor noise modeling, whereas (c) depicts the KITTI data. After performing the training phase (d), the results from the best models are obtained and compared. (e) illustrates the results from Section V-C: the top chart for object detection, and the bottom chart for semantic segmentation. The values in red are obtained after training the DNNs from scratch with real-world data, whereas the values in blue are achieved when initializing the DNNs with pre-trained weights on artificial data.

Depth

Realistic Simulation

LiDAR Sensor modeling and Data augmentation with GANs for Autonomous driving

Sensor: LiDAR

$$\text{Sensor Modeling} \leftrightarrow \text{Image 2 Image Translation(Real LiDAR data} \leftrightarrow \text{Simulation LiDAR data)}$$

$$\text{Simulated LiDAR Feature} \overset{\text{CycleGAN}}{\leftrightarrow} \text{Real World LiDAR Feature}$$

Depth

Data Augmentation

Unsupervised Neural Sensor Models for Synthetic LiDAR Data Augmentation

- Cycle GAN :
  $$\underset{G,F}{\text{argmin}} \max_{D_X, D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, X, Y) + \lambda[\mathcal{L}_{R_Y}(G, F, Y) + \mathcal{L}_{R_X}(F, G, X)]$$
  - X, Y : original / real data
  - G, F : forward/ backward network, $G : X \to Y, F : Y \to X$
  - D_X, D_Y : discriminators
- NST : $\underset{G}{\text{argmin}} \underbrace{\lambda_s \mathcal{L}_s(p)}_{\text{style loss}} + \underbrace{\lambda_c \mathcal{L}_c(p)}_{\text{content loss}}$
  - p : generated image

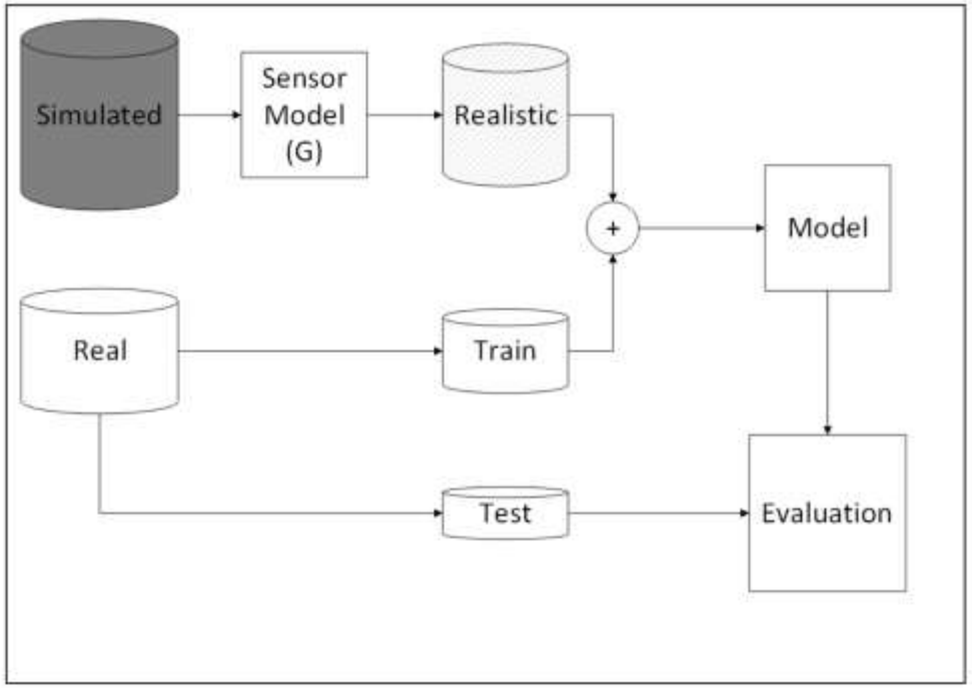


Figure 2: Data augmentation framework

- $G = G_{cyc}$: the CycleGAN sensor model.
- $G = G_{NST}$: the NST sensor model.

Depth

Kinect Survey

Characterizations of Noise in Kinect Depth Images: A Review

- noise models of Kinect
  - geometric of Pin-Hole Cameral Models
  - Empirical Models
  - Statistical Noise Models
- characterization of Kinect noise
  - Spatial Noise
  - Temporal Noise
  - Inference Noise

Depth

Realistic Simulation

Sim2Real2Sim: Bridging the Gap Between Simulation and Real-World in Flexible Object Manipulation

Simulator : Gazebo

Task : DRC Plug Task

Method :

- real world : visual servoing approach to align the cable-tip pose with the socket pose.
- simulation : Recursive Newton Euler

$$\underset{K,D}{\operatorname{argmin}} \| M\ddot{\mathbf{q}} + C\dot{\mathbf{q}} + G + J^\top \mathbf{f}_{\text{ext}} + K\mathbf{q} + D\dot{\mathbf{q}} - \tau \|$$

  - K : stiffness
  - D : Damping
  - M : inertia matrix
  - C : centrifugal and Coriolis forces
  - G : gravitational forces or torque
  - τ : joint torque

Sensor : Kinect(RGBD)

Novelty : optimize simulation from real

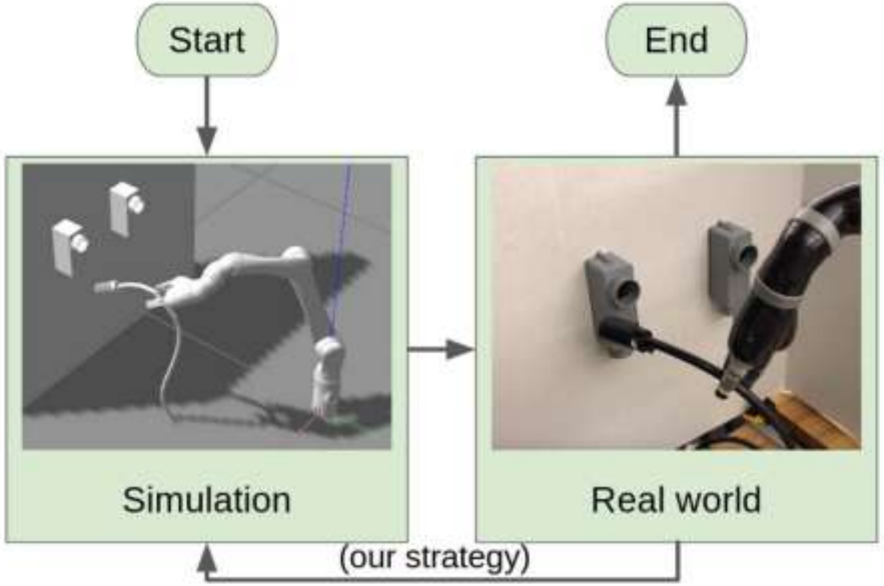
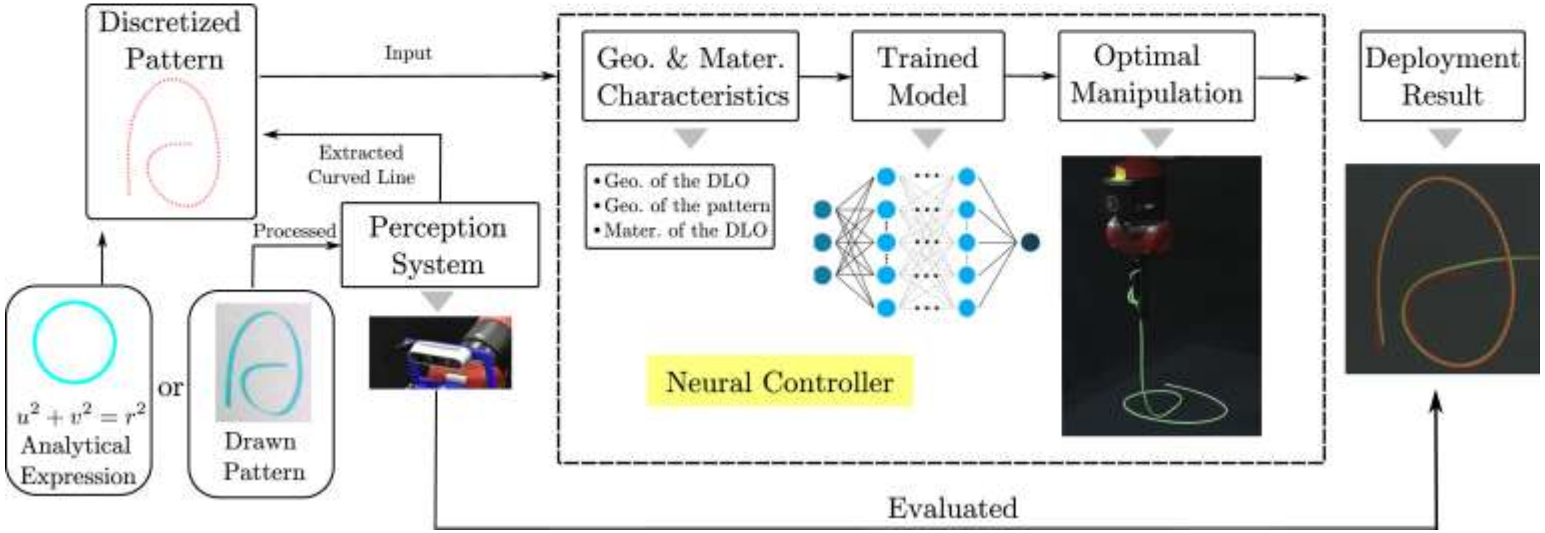


Fig. 2: Sim2Real2Sim flowchart representation.

Depth

Naive

Sim2Real Neural Controllers for Physics-Based Robotic Deployment of Deformable Linear Objects



DLO : Deformable linear objects

IMU

Policy Modal

Policies Modulating Trajectory Generators

Sample at a certain freqeuncy

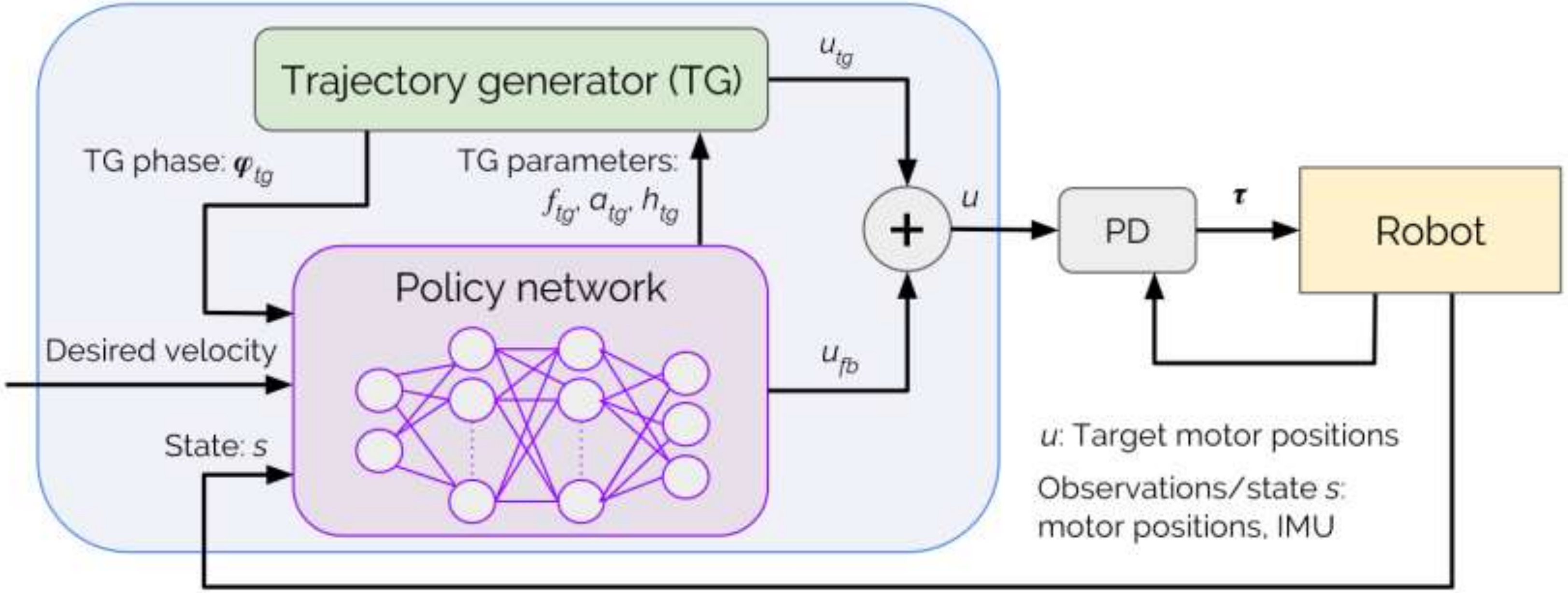


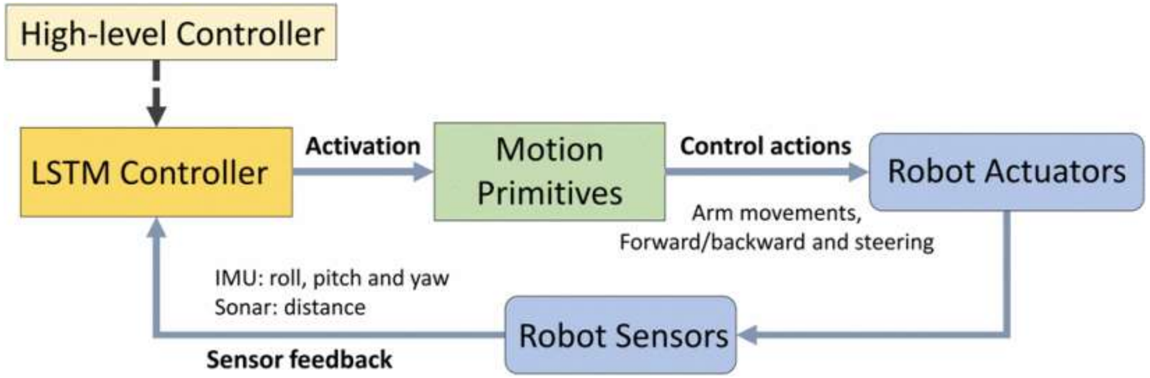
Figure 5: Adaptation of PMTG to the quadruped locomotion problem.

IMU

Controller Modal

Learning Autonomous Mobility Using Real Demonstration Data)

- Butterworth loss-pass filter with 15Hz to denoise and outlier removal
- LSTM interpret time-seq signals



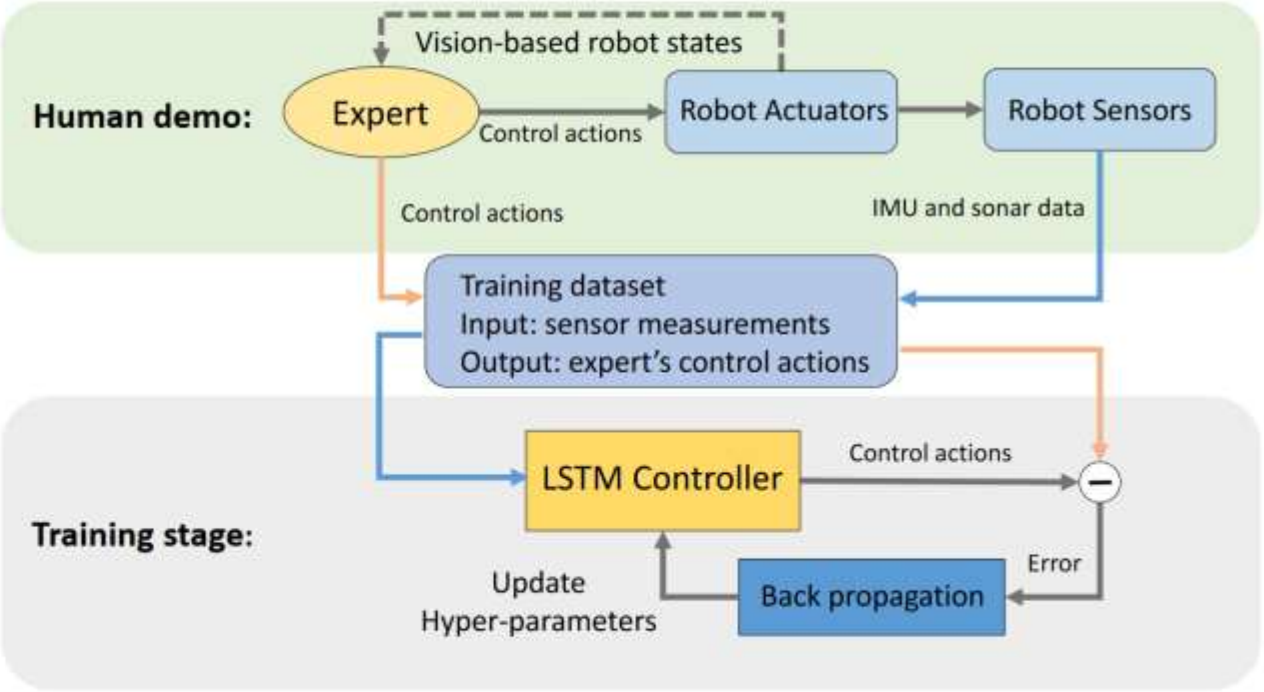


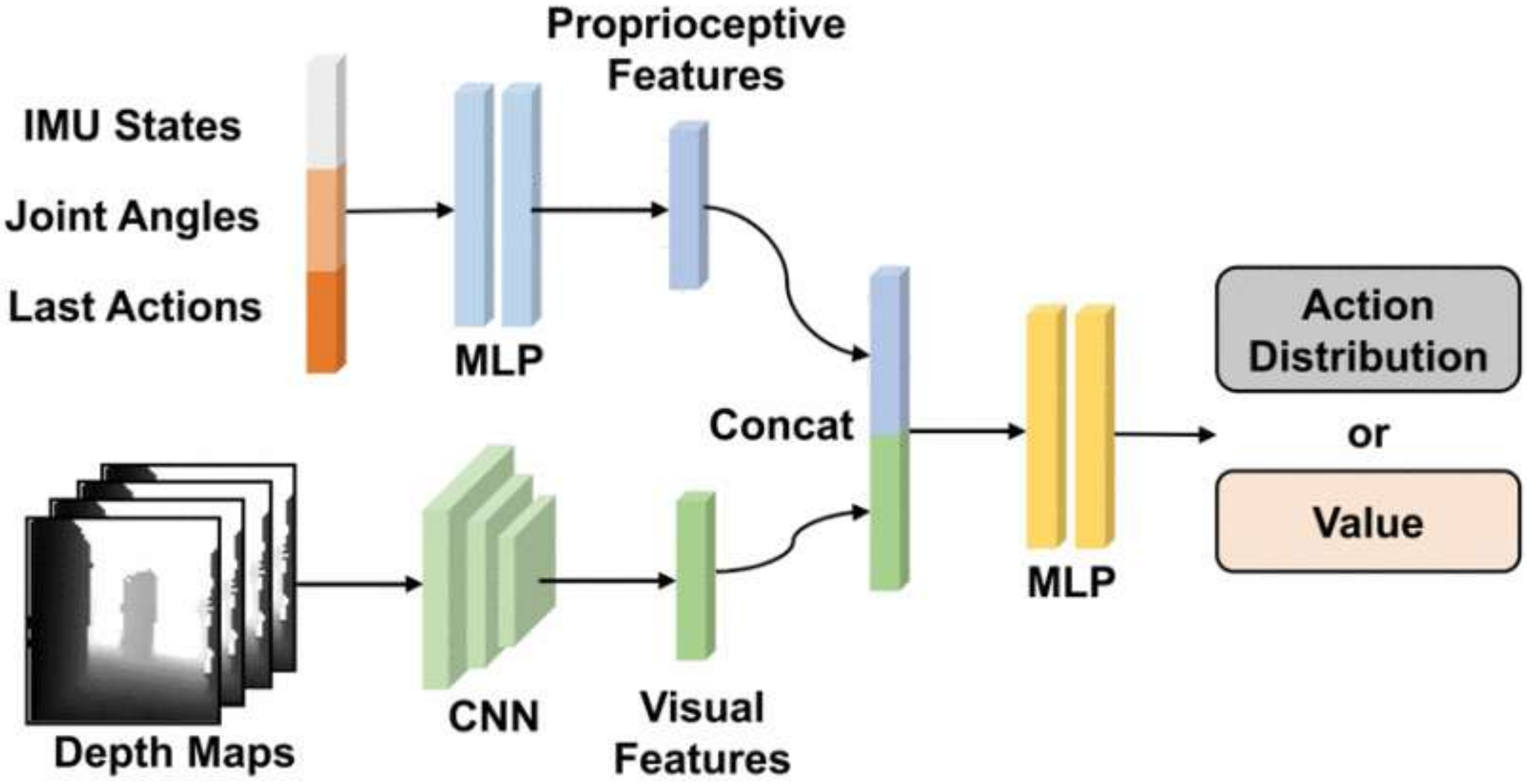
Fig. 5: Training procedure.

IMU

Policy Modal

Sim-to-Real Strategy for Spatially Aware Robot Navigation in Uneven Outdoor Environment

Observation buffer



IMU

Policy Modal

Sim-to-Real Strategy for Spatially Aware Robot Navigation in Uneven Outdoor Environment

- Extract surface vibration using PCA (first two principle components)
- DWA: Dynamic Window Approach to penalize velocities that could cause robot flip-overs

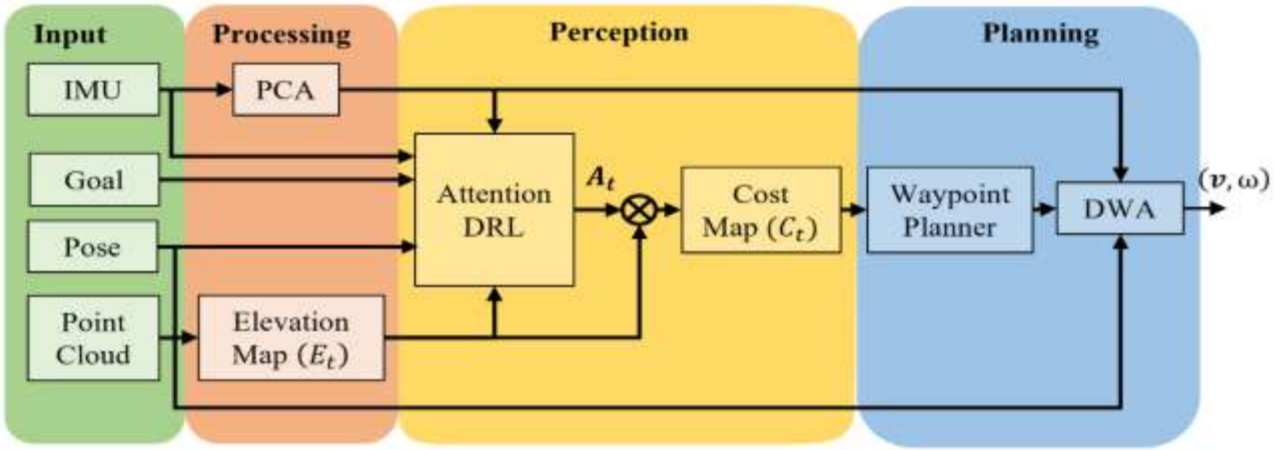


Fig. 2: **Our Overall System Architecture:** We propose a hybrid architecture to combine perception from the DRL module with our planning module. Instead of using actions from the end-to-end DRL network, we extract an intermediate output (A_t) from it to compute a navigation cost-map (C_t) to couple with our planner. This formulation displays comparable or better navigation performance in both simulated and real-world environments. Detailed analysis about the benefits of our method is presented in Section III-D.

IMU

Error Estimation

Zero-Shot Policy Transferability for the Control of a Scale Autonomous Vehicle

Advantage:
- explainable coupled with IMU signal

Limitation :
- naive control
- IMU signal is assumed to be exact

$$\text{IMU signal} \rightarrow \text{heading} \rightarrow \text{error state} \rightarrow NN \rightarrow \text{left/right control}$$

IMU for error estimation

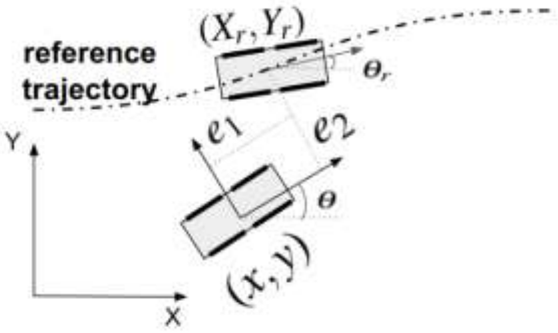


Fig. 1: Error state relative to target reference trajectory.

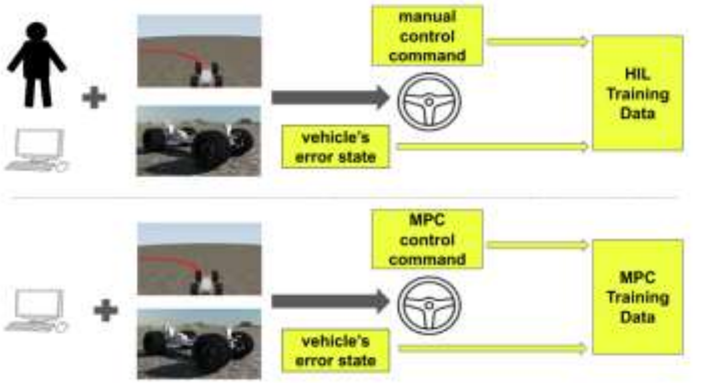


Fig. 2: Training Process Demonstration: upper half is the pipeline for collecting HIL (manual control) training data; lower half shows data collection using MPC.

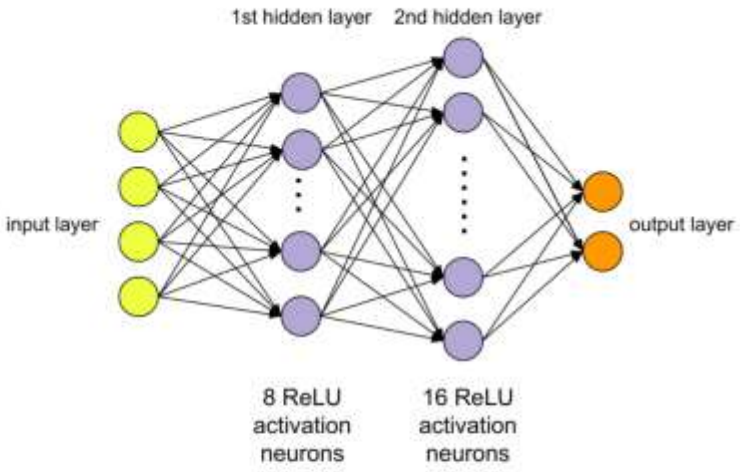


Fig. 3: Feed Forward Layers Setup.

Force Sensor

Review

Robotic tactile perception of object properties: A review

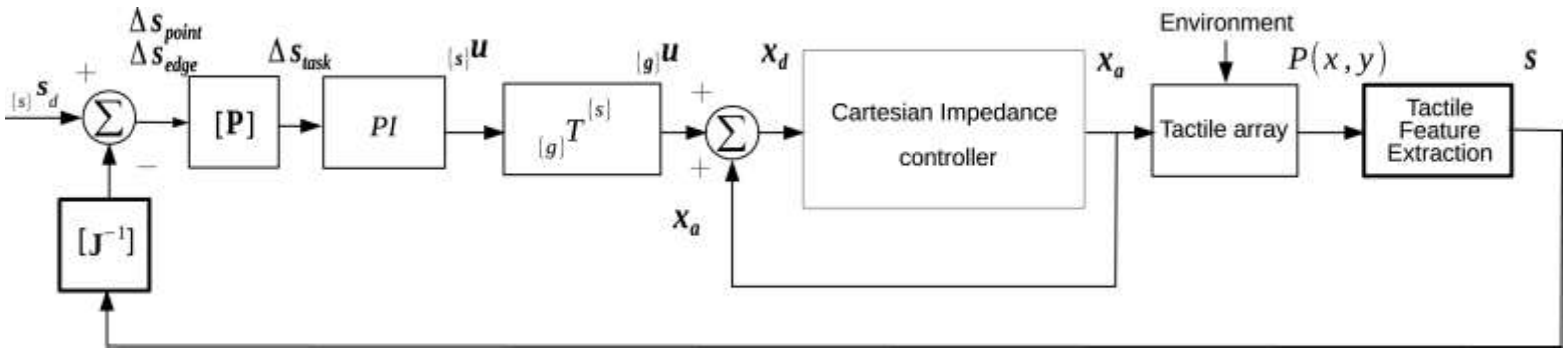
- Single-point contact sensor :
    - measure contact force : ATI Nano 17 force-torque sensor
    - measure vibration : biomimeticwhiskers
- Tactile Array: fiber optics, MEMS barometers, RoboTouch, DigiTacts
- optical tactile sensor, high resolution : GelSight, GelTip, TacTip, DIGIT

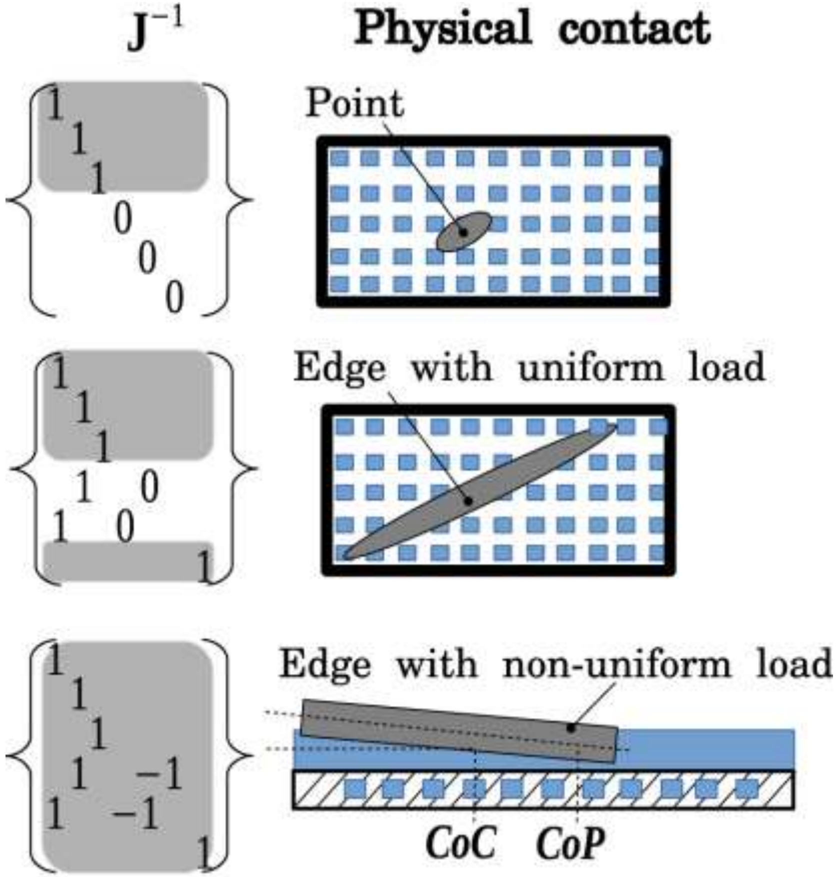
Force Sensor

Jacobian feed forward

Touch driven controller and tactile features for physical interactions

$$\text{Physical Contact} \rightarrow J^{-1} \rightarrow \text{Controller}$$





Force Sensor

FEM Simulation

Sim-to-Real for Robotic Tactile Sensing via Physics-Based Simulation and Learned Latent Projections

sensor: BioTac, single point pressure sensor

$E(\text{Young's modulus}), \mu(\text{friction}), \nu(\text{poisson ratio})$ are free parameters

same indentation are applied in simulation

simulation: linear elasticity FEM

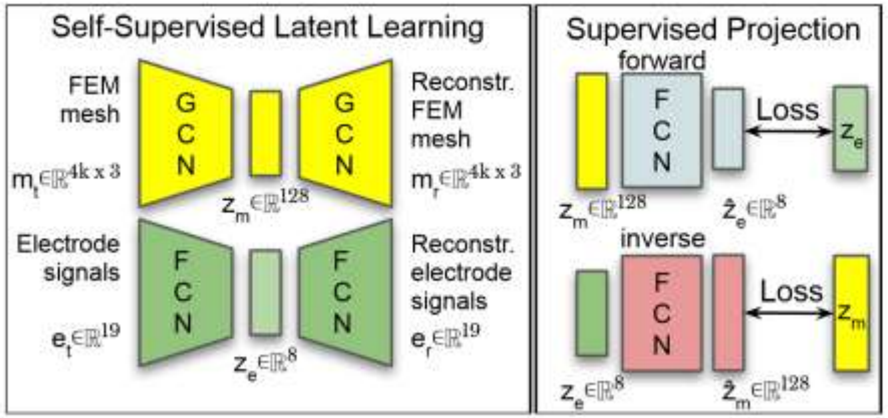


Fig. 3. Learning structure. To map between FEM deformations and BioTac electrode signals, modality-specific latent representations were learned via self-supervision. Specifically, graph convolutional networks (GCN) compressed deformed meshes with 4000 nodes to a 128-dim. latent space, and fully-connected networks (FCN) compressed the 19 electrode signals to an 8-dim. latent space. Next, FCNs were used on a small supervised dataset to learn forward and inverse projections between the latent spaces.

Yashraj Narang*[1], Balakumar Sundaralingam*[1], Miles Macklin[1], Arsalan Mousavian[1], Dieter Fox[1,2]

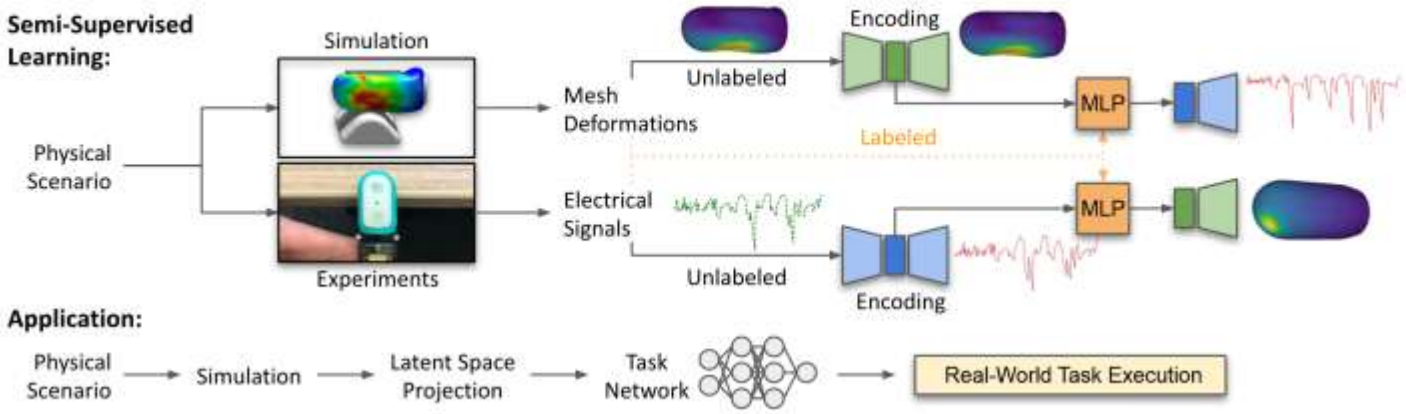


Fig. 1: Overview. We develop an efficient 3D FEM model of a SynTouch BioTac sensor to simulate contact interactions, and we conduct similar real-world experiments. In a learning phase, we train autoencoders to reconstruct unlabeled FEM deformations and real-world electrical signals. With a small amount of labeled data, we subsequently train MLPs to project between the FEM and electrical latent spaces. At test time, we use these learned latent projections to perform cross-modal transfer between FEM and electrical data for unseen contact interactions. During downstream application, we 1) accurately synthesize BioTac electrical signals, and 2) estimate the shape and location of contact patches, facilitating real-world task execution.

Force Sensor

Simulation Enhancement

Generation of GelSight Tactile Images for Sim2Real Learning

- Sensor : GelSight
- Simulator : Gazebo

$$H_{\text{GelSight}} = \text{GF}(H_{\text{truth}})$$
$$\text{RGB} = \text{Phong}(H_{\text{GelSight}})$$

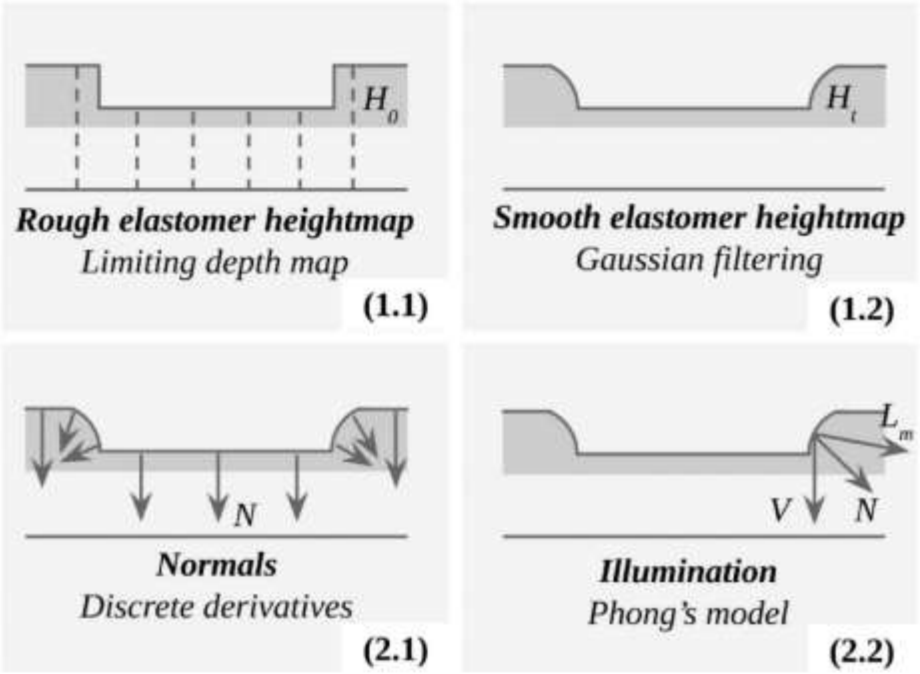


Fig. 3. The two steps in our proposed approach: 1) the elastomer heightmap is first approximated from a depth map captured by a depth camera, by **(1.1)** limiting the depth map and **(1.2)** smoothing it using Gaussian filtering; 2) then the elastomer internal illumination is rendered by **(2.1)** computing its surface normals as discrete derivatives and **(2.2)** applying Phong's illumination model.

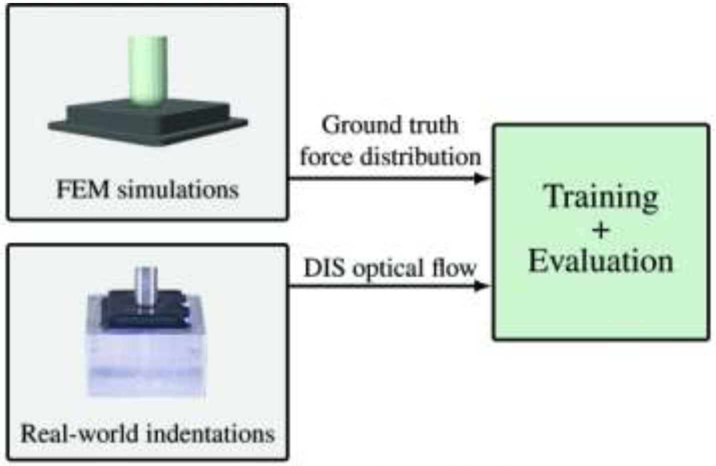
Force Sensor

FEM Simulation

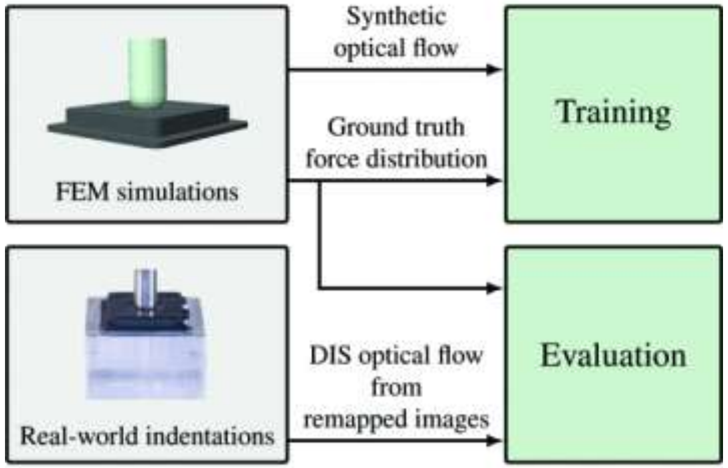
Learning the sense of touch in simulation: a sim-to-real strategy for visionbased tactile sensing

figure a is corresponding to paper "Ground Truth Force Distribution for Learning-Based Tactile Sensing: A Finite Element Approach”

DIS: Dense Inverse Search (Fast Optical Flow using Dense Inverse Search)



(a) Dataset generation as in [11]



(b) Dataset generation proposed here

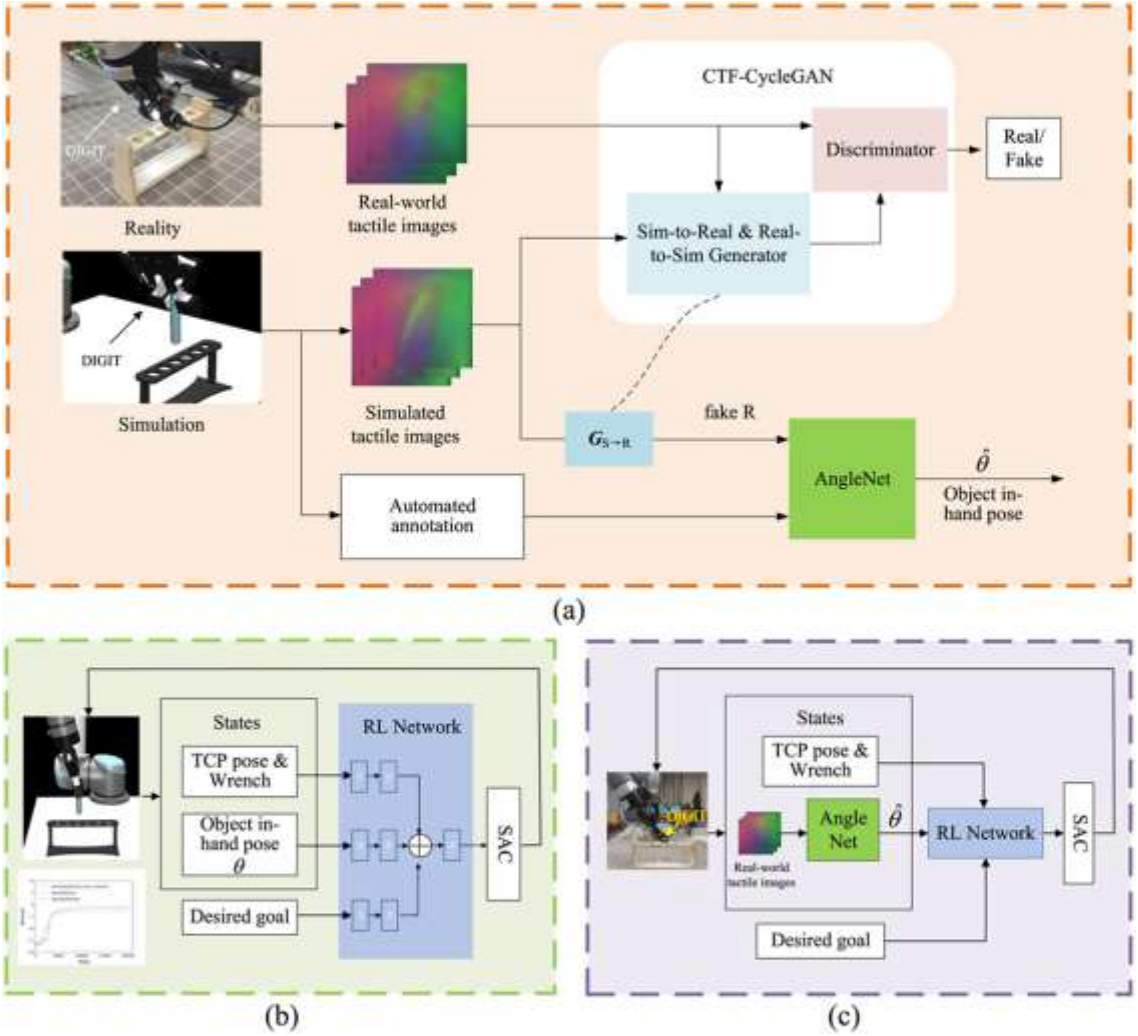
Force Sensor

Simulation Enhancement

Skill generalization of tubular object manipulation with tactile sensing and Sim2Real learning

Sensor : DIGIT

$$\text{Image(from simulation)} \overset{\text{CycleGAN}}{\leftrightarrow} \text{Image(from real world)}$$



- SAC : Soft Actor-Critic
- TCP : Tool Center Point

Encoders

Uncertainty Estimation

NeuronsGym: A Hybrid Framework and Benchmark for Robot Tasks with Sim2Real Policy Learning

$$\tilde{\omega}_i(t) = \omega_i(t) + n^e, n^e \sim \mathcal{N}(\mu_e, \sigma_e)$$

A simulator framework

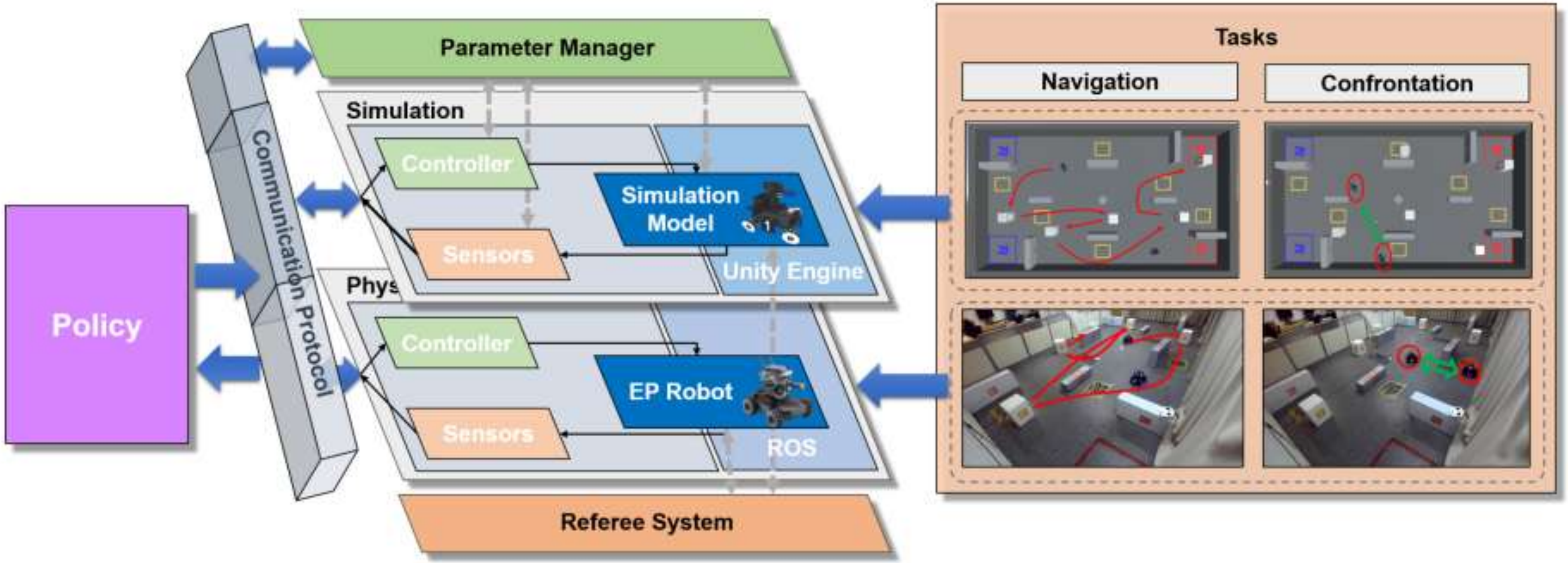


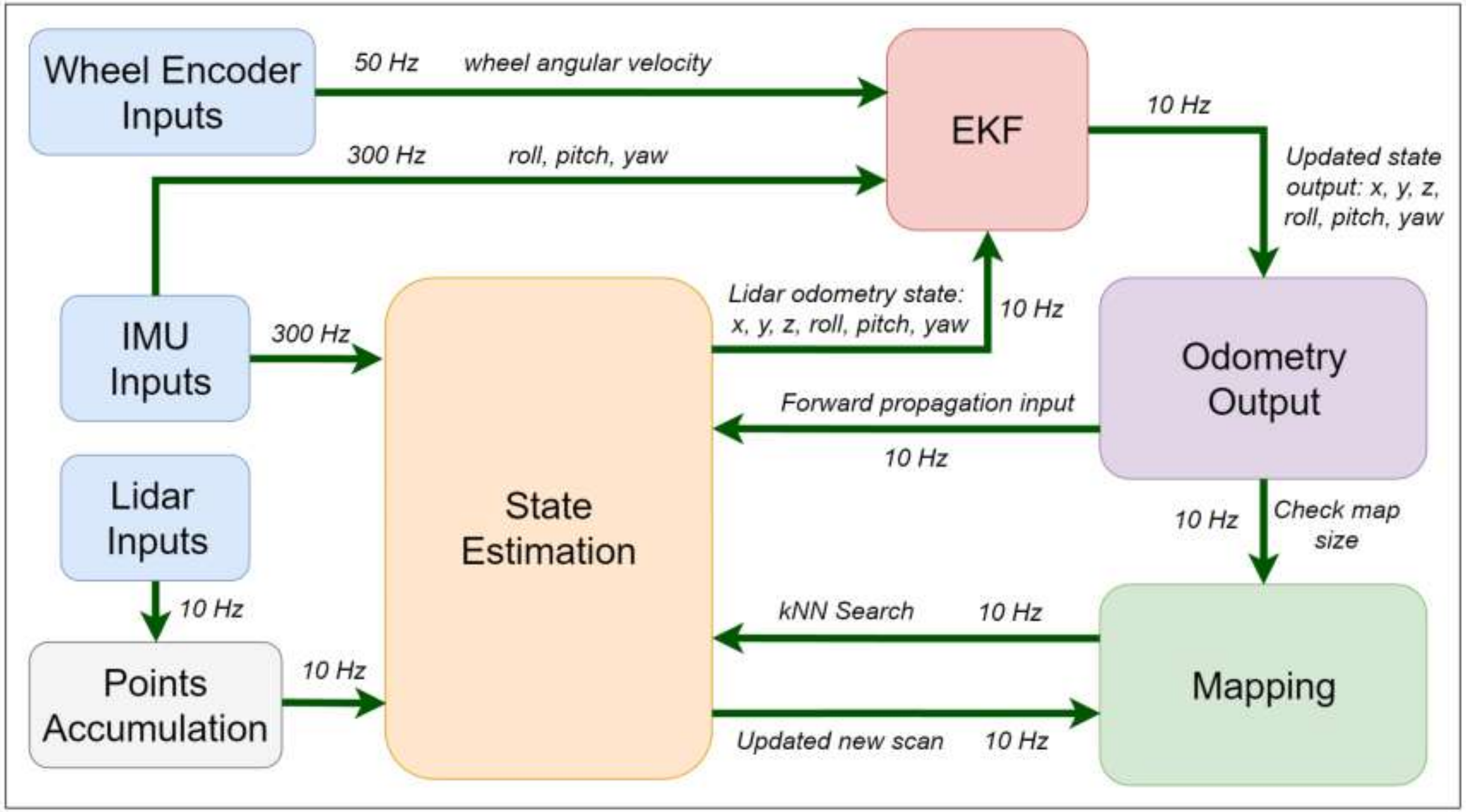
Fig. 1: Overview of the hybrid framework - NeuronsGym. The framework is composed of simulation and physical system. Agents can interact with simulation systems or physical systems through communication protocols to achieve agent training or evaluation. The agent policy can access the parameter manager to adjust parameters of the robot model or environment in the simulation system. In addition, the same scenario and task are set in each system to study sim2real of the robot policy.

Encoders

Sensor Fusion

LiDAR SLAM with a Wheel Encoder in a Featureless Tunnel Environment

EKF synchronize frequency



Advantage :
- combine IMU, Encoders, LiDAR using extended Kalman Filter

Limitation :
- the algorithm is only validated in flat and inclined terrian

Thank you