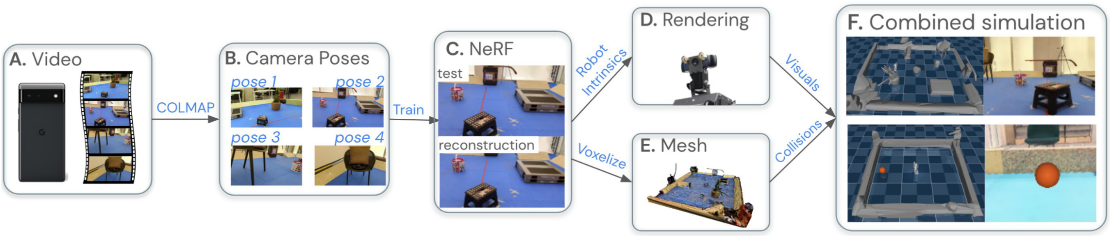
- [Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields]()

  nerf



  This method enables the creation of photorealistic simulation environments from real-world scenes, captured simply using mobile cameras. The paper demonstrates the effective training and transfer of vision-based navigation and interaction policies for humanoid robots in these environments.

- [Sim2Real2Sim: Bridging the Gap Between Simulation and Real-World in Flexible Object Manipulation]()

  visual feedback

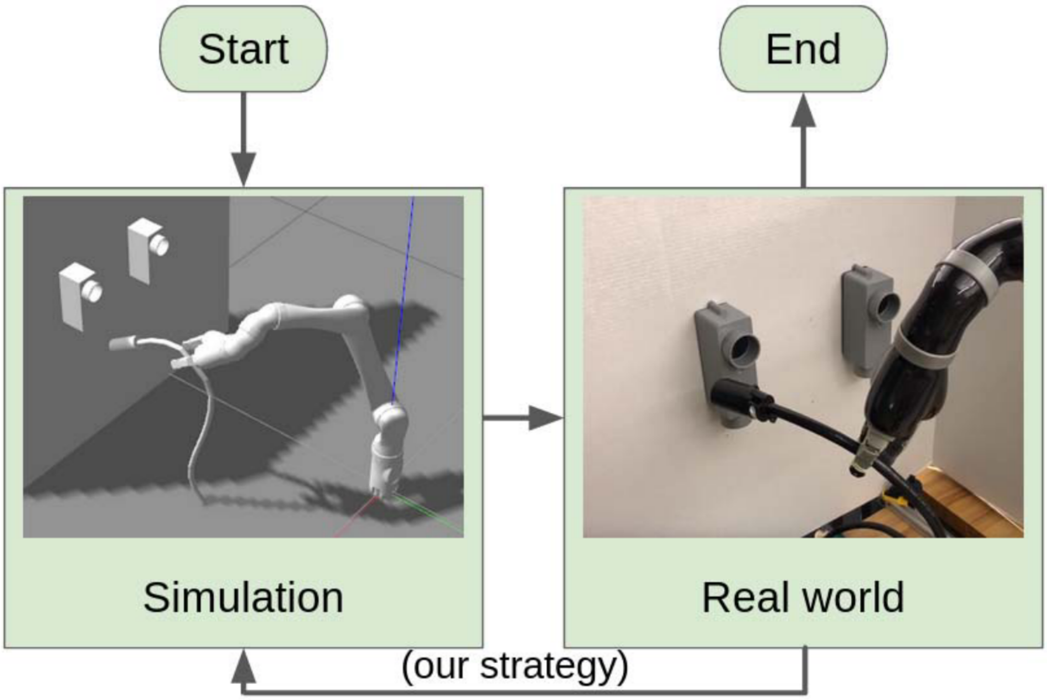


Fig. 2: Sim2Real2Sim flowchart representation.

Algorithm 1: Recursive Newton Euler algorithm

Forward recursion:{

Initial: velocity and acceleration of the base link both equal to 0

Compute link angular velocity:

$w_i = (R_i^{i-1})^T(w_{i-1} + \dot{q}_i z_{i-1})$

Compute link angular acceleration:

$\dot{w}_i = (R_i^{i-1})^T(\dot{w}_{i-1} + \ddot{q}_i z_{i-1} + \dot{q}_i w_{i-1} \times z_{i-1})$

Compute linear acceleration of origin of frame i:

$a_i = (R_i^{i-1})^T a_{i-1} + \dot{w}_i \times L_{i-1,i} + w_i \times (w_i \times L_{i-1,i})$

Compute linear acceleration of centre of C_i:

$a_{ci} = a_i + \dot{w}_i \times L_{i,c_i} + w_i \times (w_i \times L_{i,c_i})$ }

Backward recursion:{

Compute force exerted by link $i-1$ on link i:

$f_i = R_{i+1}^i f_{i+1} + m_i a_{ci}$

Compute moment exerted by link $i-1$ on link i:

$\mu_i = -f_i \times (L_{i-1,i} + L_{i,c_i}) + \mu_{i+1} + f_{i+1} \times L_{i,c_i} + I_i \dot{w}_i + w_i \times (I_i w_i)$

Compute torque exerted by link $i-1$ on link i:

$\tau_i = \mu_i^T z_{i-1} + \tau_{inertia}$ }

Algorithm 2: Visual servoing approach

Result: "cable_tip" frame = PRE-INSERT frame

while *"cable_tip" frame != PRE-INSERT frame* **do**

- Calculate the difference between the PRE-INSERT frame and the "cable_tip" frame.
- Transfer the position difference to be Cartesian velocity ($\mathbf{v}$) by dividing by a constant time.
- Obtain the joint velocity ($\dot{\mathbf{q}}$) through Cartesian velocity ($\mathbf{v}$) and robot Jacobian (J): $\dot{\mathbf{q}} = J^{-1}\mathbf{v}$.
- Use PID controller to control the joint velocity.
- Send joint velocity to the robot.

end

Sim2Real2Sim adds an essential step of feedback and refinement.

What makes Sim2Real2Sim innovative is its additional phase of refining the simulation models based on real-world data and experiences. After the initial transfer from simulation to the real world, the observed differences and inaccuracies in the real-world application are used to update and improve the simulation models. This process creates a feedback loop where the simulation continuously evolves and becomes more accurate and representative of the real world.

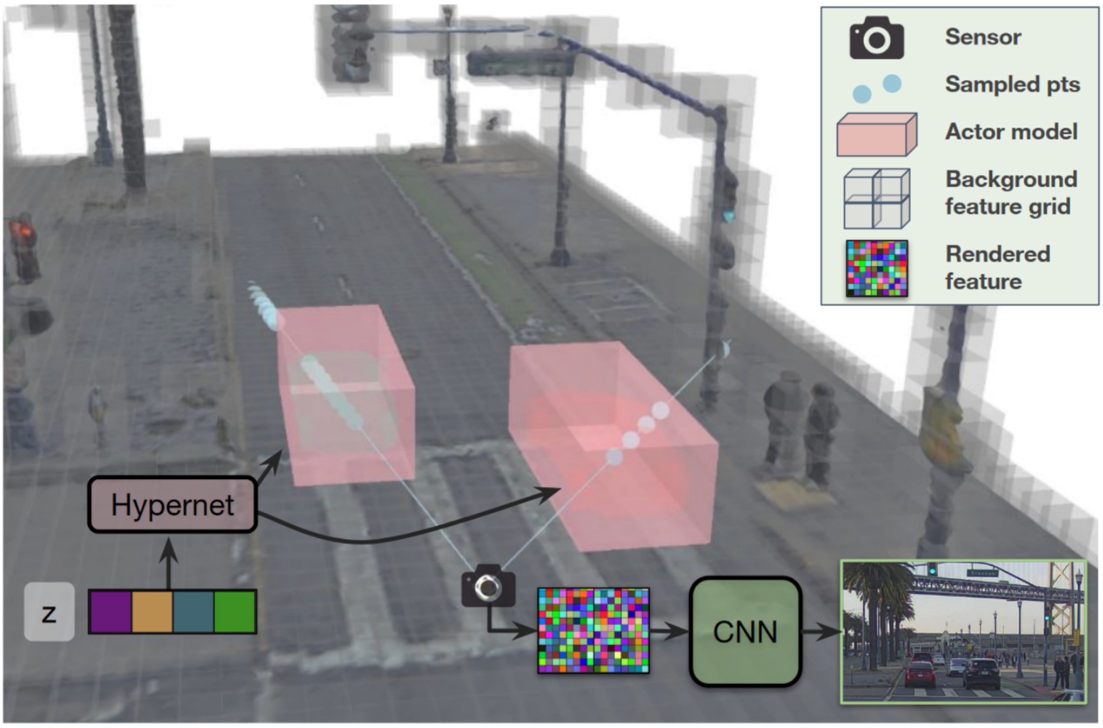
- [LiDAR Sensor modeling and Data augmentation with GANs for Autonomous driving]()

  cycle-GAN

  $$\text{Realistic LiDAR Data} = \text{CycleGAN}(\text{Simulated LiDAR Data}, \text{Real-world LiDAR Features})$$

  The core of the paper is the formulation of the problem as an image-to-image translation from unpaired data using CycleGANs. This approach is used to solve the sensor modeling problem for LiDAR, enabling the production of realistic LiDAR data from simulated LiDAR (sim2real) and generating high-resolution realistic LiDAR from lower resolution data (real2real).

- [UniSim: A Neural Closed-Loop Sensor Simulator]()

  They divide the 3D scene into a static background (grey) and a set of dynamic actors (red). Then 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

- [Unsupervised Neural Sensor Models for Synthetic LiDAR Data Augmentation]()

  Data Augment

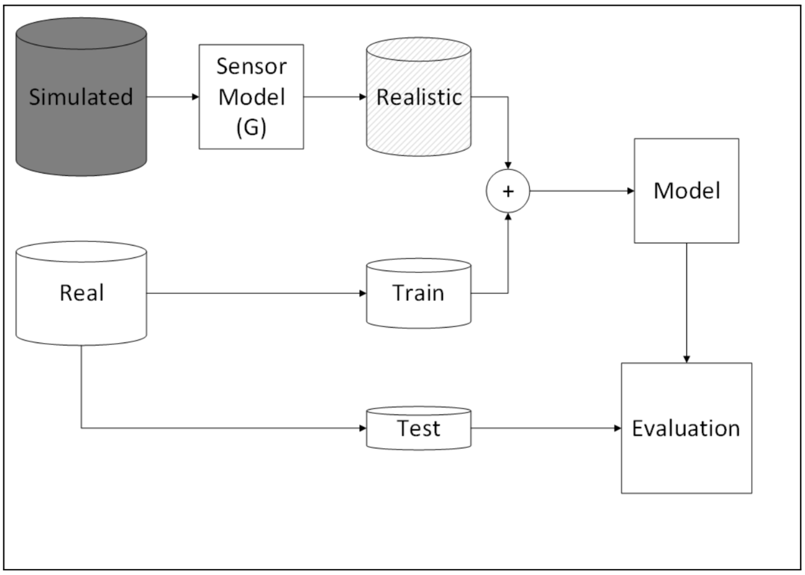


Figure 2: Data augmentation framework

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

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

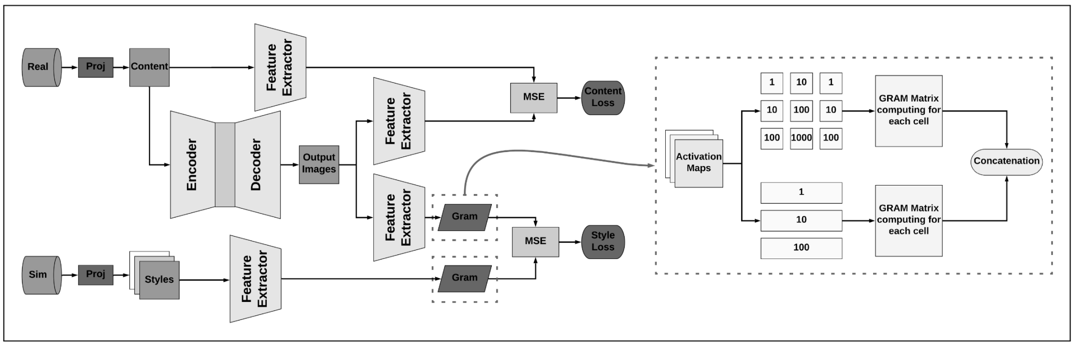


Figure 4: Neural style transfer model

two main neural sensor models (NSMs) for LiDAR data augmentation using synthetic data: CycleGAN and Neural Style Transfer (NST).

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

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

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

Domain random

In the learning process of the study, the sensor error modeling is used to make the simulated LiDAR data more realistic by introducing noise and point dropout, as described by the noise and dropout models. This process helps in training the neural networks to be more robust to the imperfections commonly found in real-world LiDAR data.

- SurfelGAN: Synthesizing Realistic Sensor Data for Autonomous Driving

Synthetic Dataset

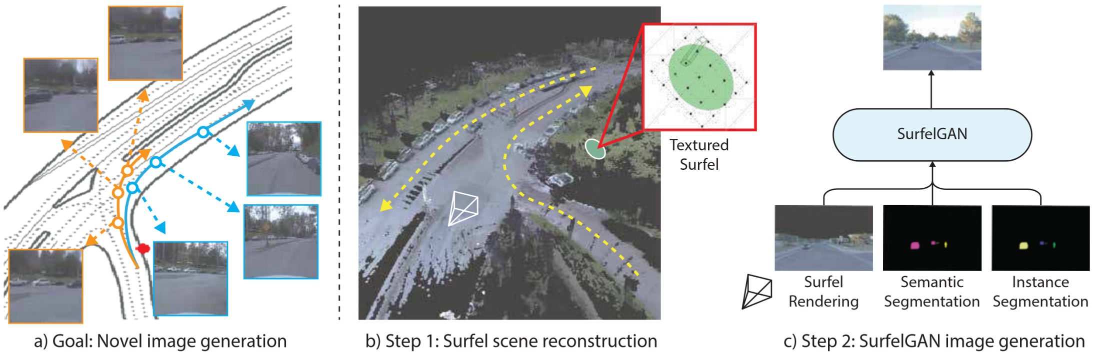


Figure 1. **Overview of our proposed system.** a) The goal of this work is the generation of camera images for autonomous driving simulation. When provided with a novel trajectory of the self-driving vehicle in simulation, the system generates realistic visual sensor data that is useful for downstream 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.

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

gap evaluation

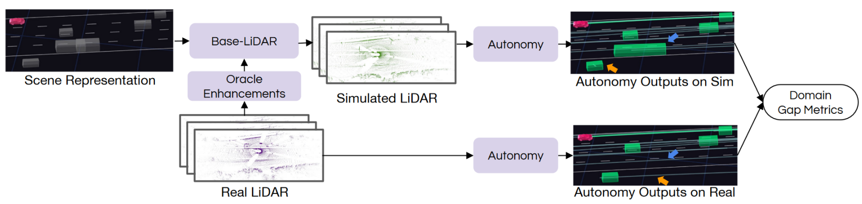
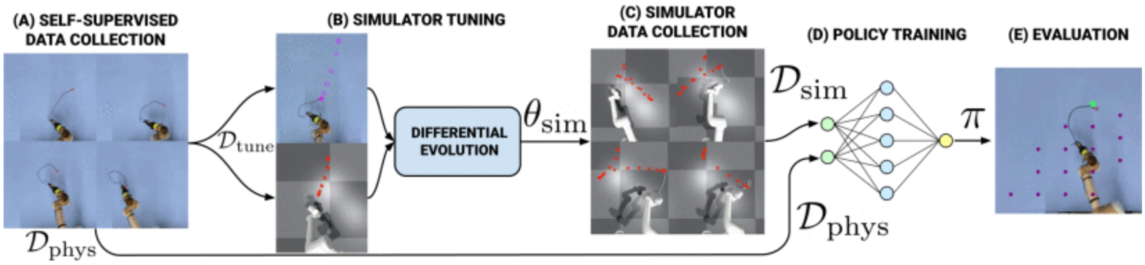


Figure 5. Given paired simulated and real LiDAR for the same scenario, we run autonomy on both in open-loop and compare the domain gap for the autonomy under test.

proposes a novel "paired-scenario" approach for evaluating the domain gap in LiDAR simulators. This involves reconstructing digital twins of real-world scenarios, then simulating LiDAR data in these scenarios and comparing it with actual LiDAR data. The study focuses on how different aspects of LiDAR simulation, like pulse phenomena, scanning effects, and asset quality, affect the domain gap in relation to an autonomy system's perception, prediction, and motion planning capabilities.

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



The Real2Sim2Real framework proposed in this paper is self-supervised and focuses on efficiently learning a PRC(Planar Robot Casting) policy for different types of cables. It does so by collecting physical trajectory data, using these to tune the parameters of a dynamics simulator through Differential Evolution, and then generating simulated examples. The learning policy is derived from a weighted combination of both simulated and physical data. The methodology was tested using three different simulators and two function approximators (Gaussian Processes and Neural Networks) on cables with varying physical properties.

-