import numpy as npimport rosbag2\_pyfrom sensor\_msgs.msg import PointCloud2from sensor\_msgs\_py import point\_cloud2 as pc2def read\_point\_cloud\_from\_ros2\_bag(bag\_path, topic\_name): """ Read point cloud data from a ROS 2 bag file. Parameters: - bag\_path: Path to the ROS 2 bag file. - topic\_name: The topic name containing the PointCloud2 data. Returns: - A numpy array with shape (N, 4) where each row is [x, y, z, intensity]. """ storage\_options = rosbag2\_py.StorageOptions(uri=bag\_path, storage\_id="sqlite3") converter\_options = rosbag2\_py.ConverterOptions( input\_serialization\_format="cdr", output\_serialization\_format="cdr" ) reader = rosbag2\_py.SequentialReader() reader.open(storage\_options, converter\_options) # Define an empty list to collect point cloud data points\_list = [] # Iterate through messages while reader.has\_next(): (topic, data, timestamp) = reader.read\_next() if topic == topic\_name: # Deserialize the PointCloud2 message point\_cloud\_msg = PointCloud2() point\_cloud\_msg.deserialize(data) # Use pc2.read\_points to extract x, y, z, intensity fields for point in pc2.read\_points(point\_cloud\_msg, field\_names=("x", "y", "z", "intensity"), skip\_nans=True): points\_list.append(point) # Convert the list to a numpy array return np.array(points\_list)def save\_xyzi\_to\_pcd(filename, points): """ Save a point cloud with x, y, z, and intensity fields to a PCD file in ASCII format. Parameters: - filename: The name of the file to save the point cloud. - points: A numpy array of shape (N, 4), where each row is [x, y, z, intensity]. """ # Ensure points are in the correct shape assert points.shape[1] == 4, "Input array must have shape (N, 4) for x, y, z, intensity." # Define the PCD header for ASCII format header = f"""# .PCD v0.7 - Point Cloud Data file formatVERSION 0.7FIELDS x y z intensitySIZE 4 4 4 4TYPE F F F FCOUNT 1 1 1 1WIDTH {points.shape[0]}HEIGHT 1VIEWPOINT 0 0 0 1 0 0 0POINTS {points.shape[0]}DATA ascii""" # Write header and point data to the file with open(filename, 'w') as f: f.write(header) np.savetxt(f, points, fmt='%f %f %f %f') print(f"Point cloud saved as {filename}")# Example usagebag\_path = "path\_to\_your\_ros2\_bag"topic\_name = "/your\_pointcloud\_topic"points = read\_point\_cloud\_from\_ros2\_bag(bag\_path, topic\_name)save\_xyzi\_to\_pcd("output\_xyzi.pcd", points)