Geometry-Aware Point Cloud Learning for Robust and Efficient 3D Vision

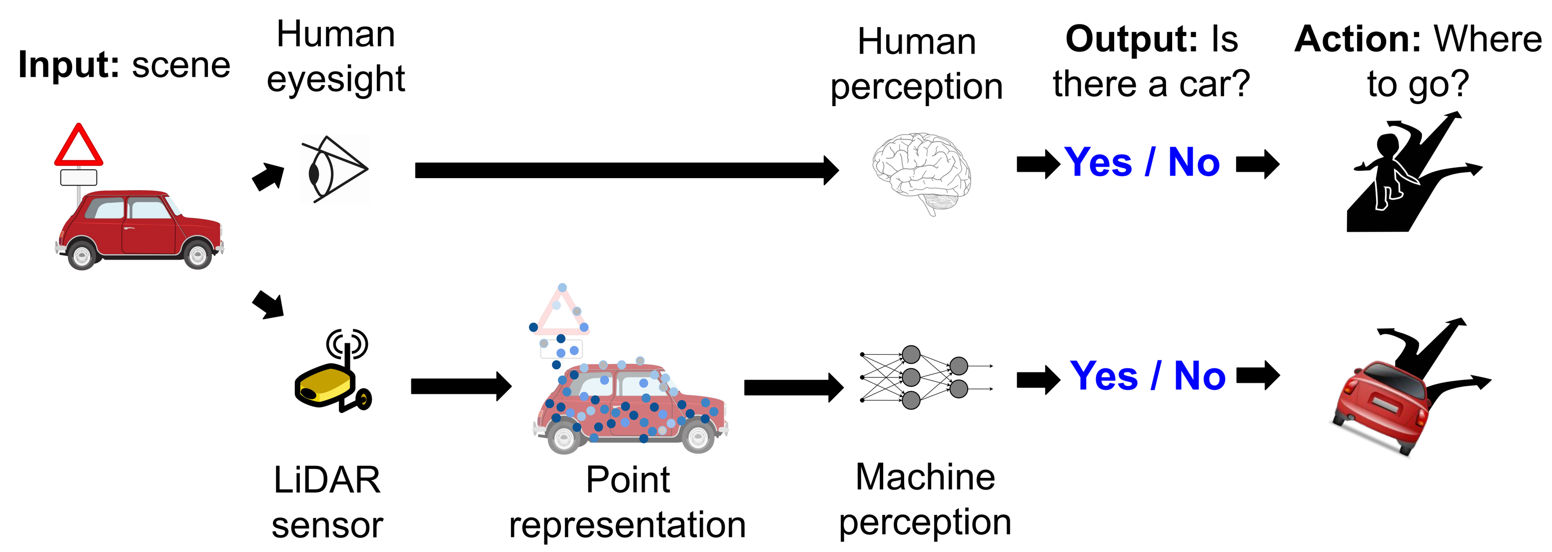
Marianne Arriola[1], Xiangru Huang[2], Justin Solomon[2]

[1]Department of Computer Science, University of California, Santa Barbara

[2]Department of Electrical Engineering & Computer Science, Massachusetts Institute of Technology

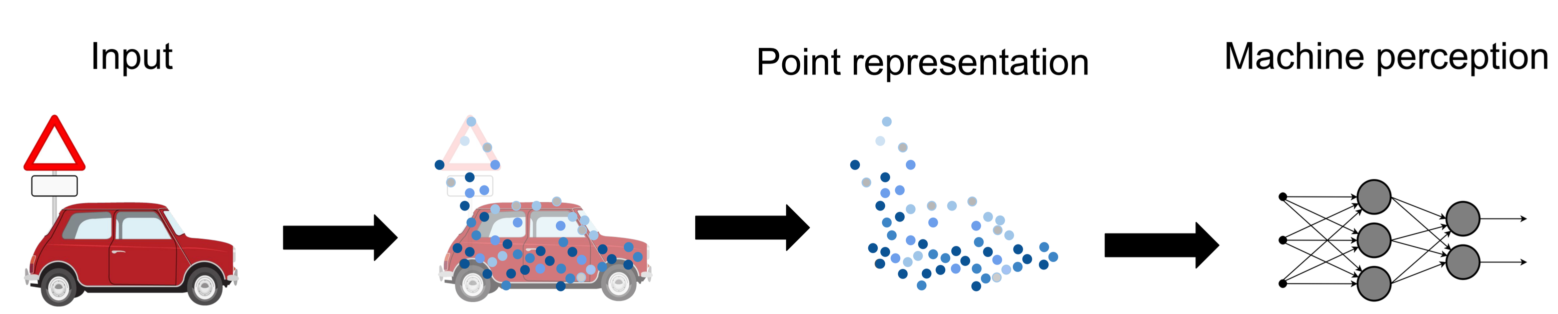
Can machines match human perception?

- We process the 3D physical world daily using human perception
- 3D vision models attempt to **match** human perception by making predictions using **points** picked up from the environment [Qi17]
- 3D vision models are used for autonomous driving, where **accurately** and **efficiently** detecting objects is crucial for safe and fast response



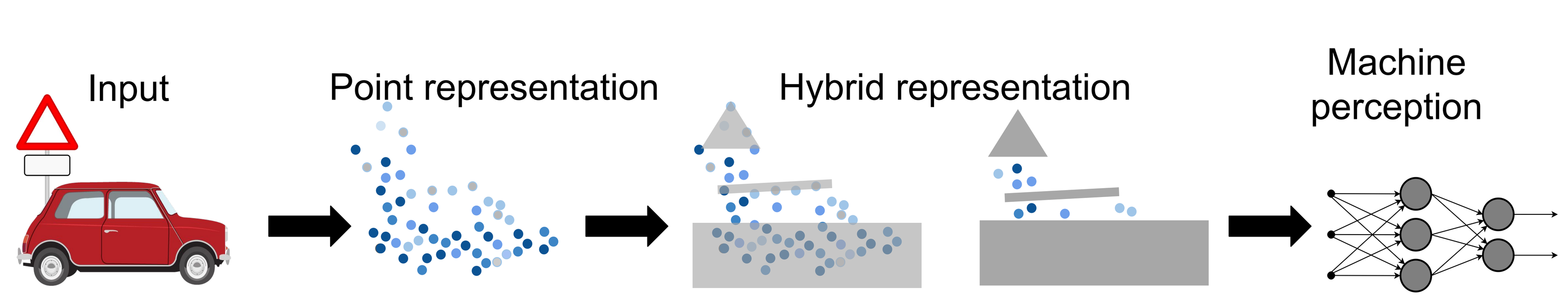
Point cloud learning is inefficient and not robust

- Point cloud learning is **inefficient** because it processes scenes that contain thousands/millions of points
- Point cloud learning is **not robust** because it must infer how **noisy**, **unevenly sampled** points compose coherent shapes



Can geometry information improve model efficiency and robustness?

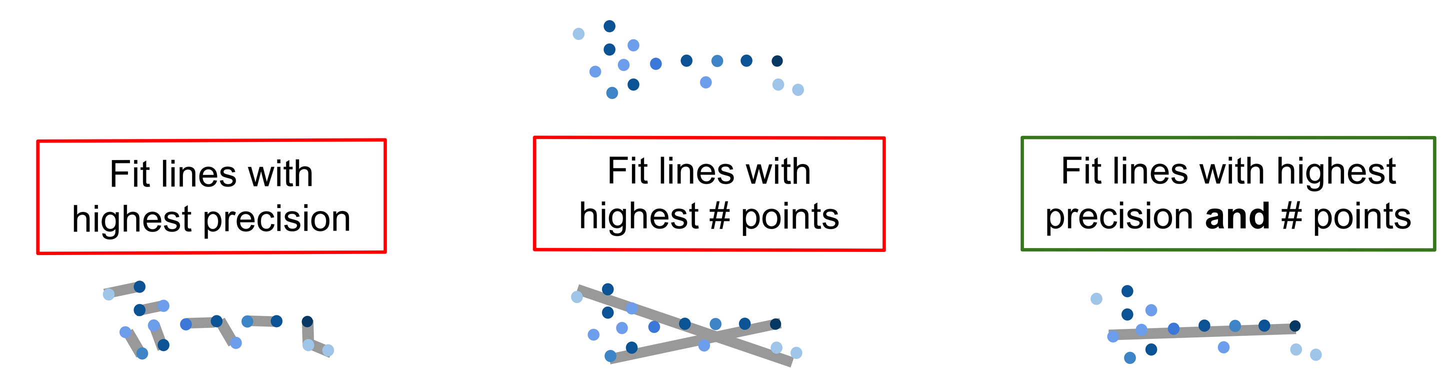
- We hypothesize that **summarizing** points into the shapes that they constitute results in more **efficient, robust** predictive models
- Point summarization **shrinks** the size of the model input and provides added **shape and object information** to the model



Summarizing points into lines

- We test our approach by fitting **lines** to points [Fis81]
- We find the best line with the highest confidence by **balancing** two factors tuned by σ and T:
  - Distance of points from the line $dist$
  - Density of points along the line $dens$

$$c(l) = e^{-\left[\left(\frac{max(dist_l)}{\sigma}\right)^2 + \frac{T}{dens_l}\right]}$$

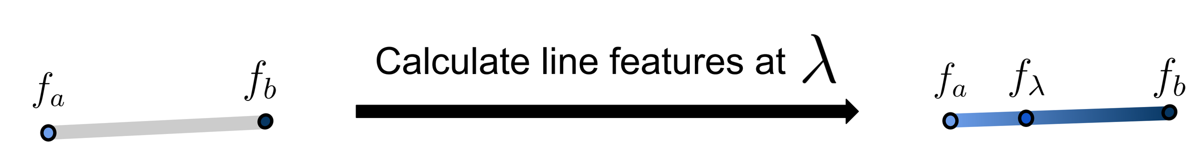


Line representation definition

- Features at any point λ along the line with endpoint features f_a, f_b is:

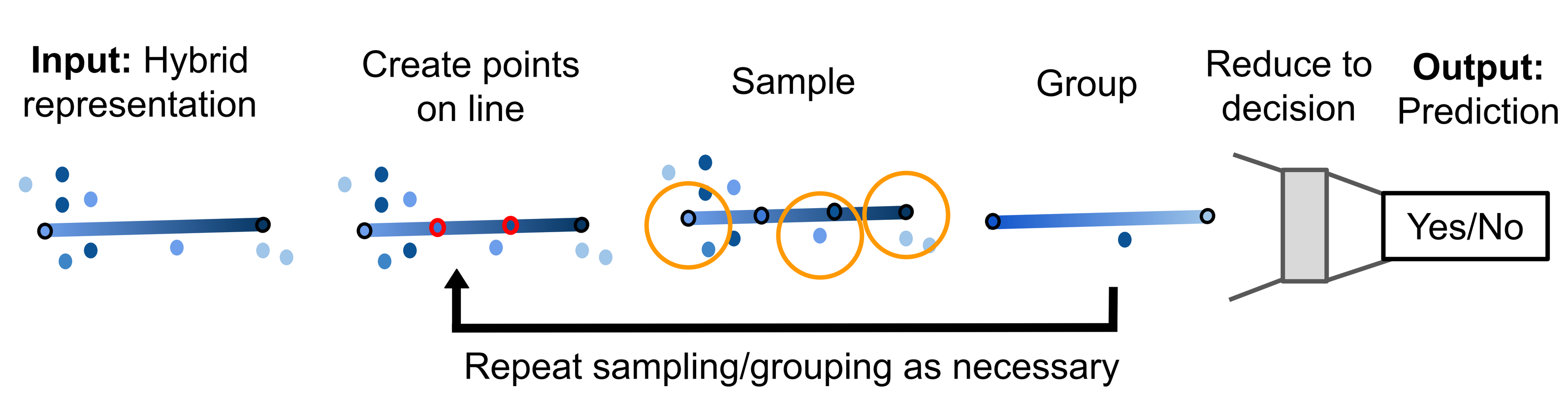
$$f(\lambda) = \lambda(f_a) + (1-\lambda)(f_b), \lambda \in [0...1]$$

- Calculating features along the line can be visualized as:



How do we learn from points *and* lines?

- We learn an **informative hybrid representation** for prediction by **sampling** the input and **grouping** information from nearby points
- Our model **learns from lines** by creating points on the line for sampling and preserving the line for the final predictions



- Features of a sample combine nearby point and line features, f_p, f_l:

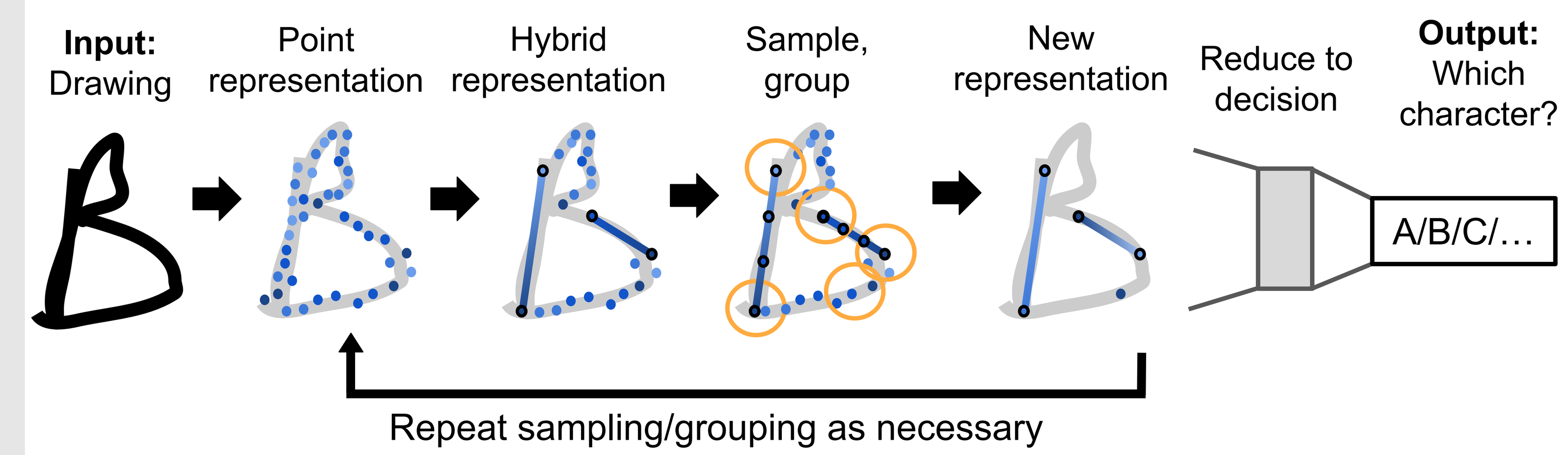
$$f(x) = f_p(x) + f_l(x)$$

- Feature **importance** is learned by a neural network g
- Points/lines are weighted by w using their **distance** to the sample

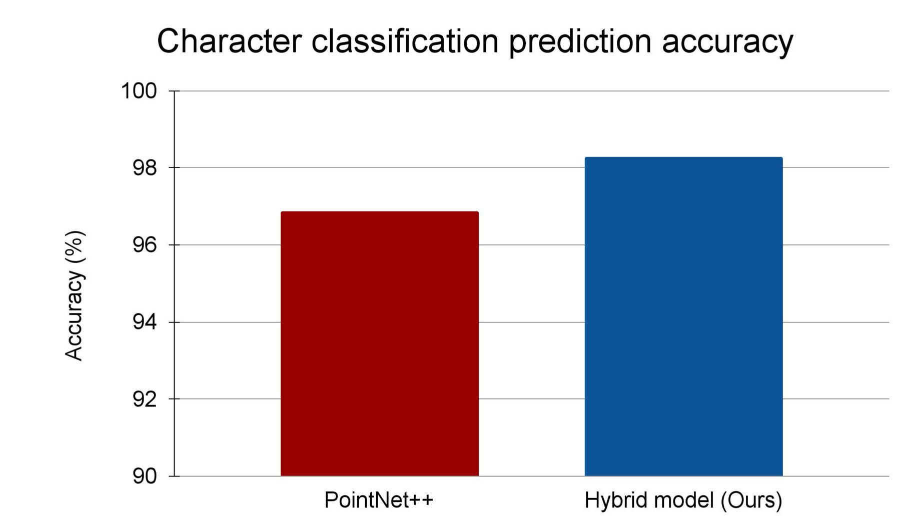
$$f_p(x) = \frac{\Sigma_{i=1}^{k} w_i(x) g(f_i)}{\Sigma_{i=1}^{k} w_i(x)} \qquad f_l(x) = \int_{m=x_a}^{x_b} w_m(x) g(f_m) \Rightarrow \frac{\Sigma_{m=1}^{q} w_m(x) g(f_m)}{\Sigma_{m=1}^{q} w_m(x)}$$

Hybrid representations improve learning

- We test our model on a small-scale 2D example:
  - Given a drawing of a character, predict which character it is



- We **compare** the predictive accuracy of:
  - Using only the **point** representation (PointNet++) [Qi17]
  - Using the **hybrid** representation

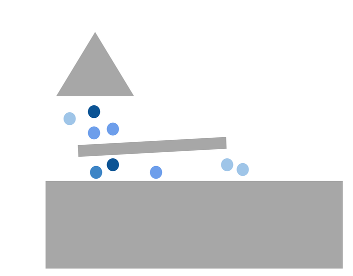


- Our architecture is **more accurate** at predicting ~1,500 character examples by **2%** than the state-of-the-art point-based architecture

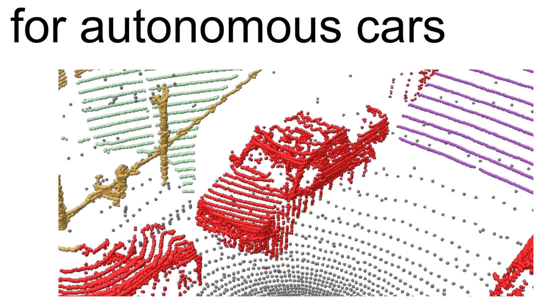
Geometry-aware learning shows promise

- Our positive results show that hybrid point cloud learning is feasible for **larger-scale** experiments
- This work can be extended to learn from hybrid representations with **diverse** shapes
  - Triangles, planes, etc.

Example:



- Future experiments needed for **real-world 3D data**
  - LiDAR sensor data for autonomous cars



1. [Qi17] C Qi, et al. *NeurIPS* **30** (2017)
2. [Fis81] M Fischler, et al. *ACM* **24**, 381-395 (1981)