LidarNAS: Unifying and Searching Neural Architectures for 3D Point Clouds

Chenxi Liu Zhaoqi Leng Pei Sun Shuyang Cheng Charles R. Qi Yin Zhou Mingxing Tan Dragomir Anguelov
Waymo LLC

ECCV TEL AVIV 2022

Motivation

Observation:

- Neural architectures for 3D point clouds exhibit a large variety
- Diverse set of concepts in architecture names: PointNet, VoxelNet, PointPillars, Range Sparse Net, ...
- This level of variety is not observed in say 2D images

Sources of Variety:

	2D images	3D point clouds
Views	perspective	perspective, unordered set, top-down, ...
Sparsity	dense	dense, sparse
Layers	conv2d	mlp, conv2d, sparse conv2d, sparse conv3d, ...

Our Goal:

- A **unified framework** that can interpret and organize the variety of neural architecture designs
- Materialize this framework into an architecture search space, which unlocks and enables a principled **Neural Architecture Search** for 3D
- Demonstrate **improved performance** as well as **interesting lessons** about neural architectures for 3D

Unify Neural Architectures for 3D

Philosophy

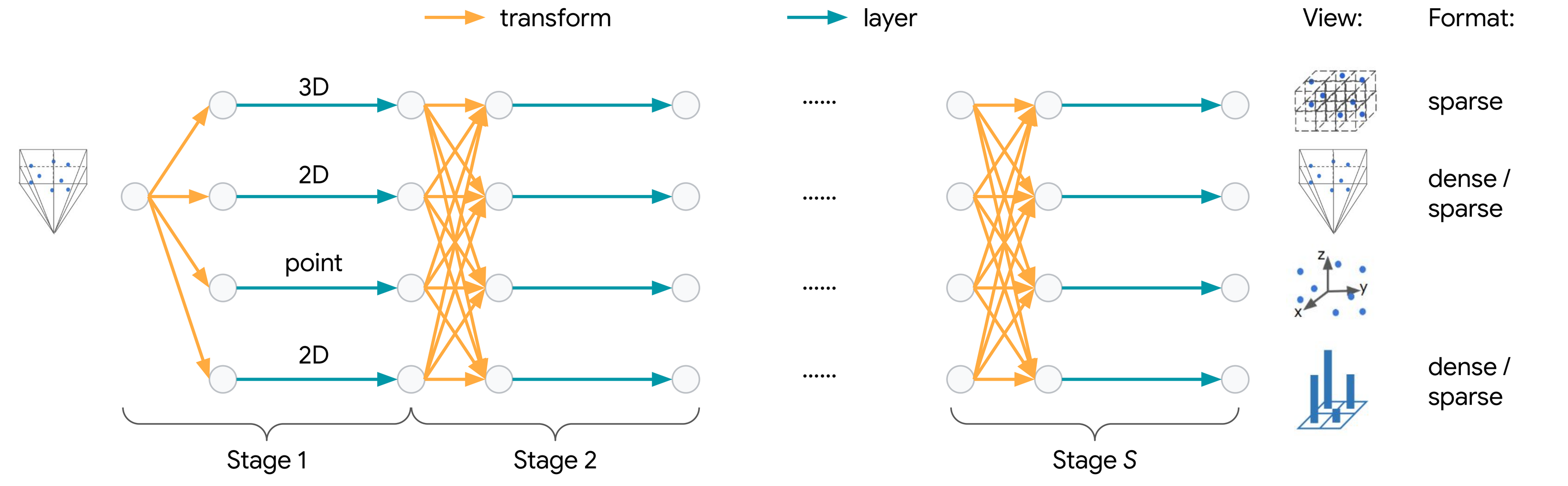
- Despite the variety on the surface, the underlying principle is surprisingly congruent: finding *some neighborhood* of the 3D points and then *aggregating information* within.
  - "neighborhood" =
    - Euclidean ball (PointNet++)
    - 3D neighborhood from Cartesian (x, y, z) (VoxelNet)
    - 2D neighborhood from Cartesian (x, y) (PointPillars)
    - 2D neighborhood from pixel index (i, j) (LaserNet)
  - "aggregation" = some form of convolution / pooling
- Different data views can transform between each other back and forth. However, once the data view is determined, it *restricts* the type of layers that can be applied.

Key Concepts

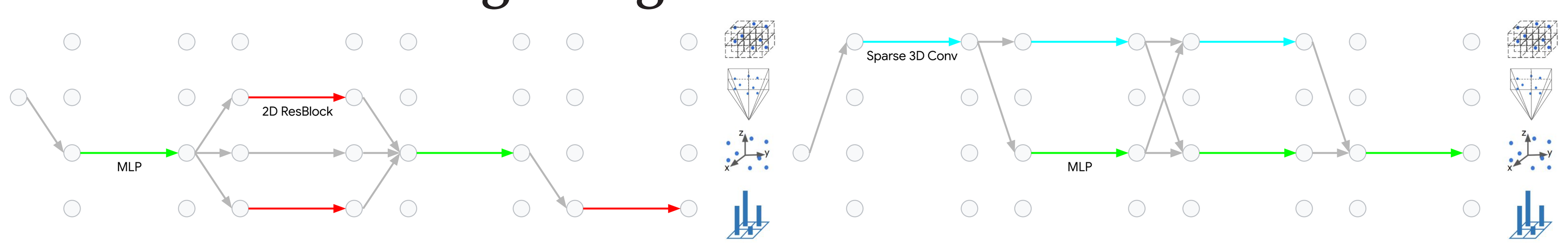
- *Views* and *formats* (6): Point, Pillar, Pillar (sparse), Voxel (sparse), Perspective, Perspective (sparse)
- *Transforms* ($6^2 = 36$): From one view-format combination to another
- *Layers*: Depending on the view-format combination
- *Stages*: Each one = sequential pair of possible transforms and their associated layers. Entire backbone = S stages.

LidarNAS Framework / Search Space

The LidarNAS Framework



Inclusion of Existing Designs



(a) Multi-View Fusion (b) Sparse Point-Voxel

Search Neural Architectures for 3D

From Framework to Search Space

- Transforms
  - No pillar to voxel. From voxel, only to pillar.
  - 31 / 36: still high coverage
- Layers
  - Point: multiple layers of dense-normalization-ReLU
  - 2D dense: U-Net with residual blocks (conv2d)
  - 2D sparse: U-Net with residual blocks (sparse conv2d)
  - 3D sparse: U-Net with residual blocks (sparse conv3d)
- Stages: $S = 3$

Search Algorithm: Regularized Evolution

Why evolutionary NAS, not weight-sharing NAS?

- Evolutionary NAS arguably makes the least approximations
- Weight-sharing NAS is too GPU memory intensive for 3D tasks, which already had a small batch size (< 10) per GPU

First randomly select a stage, then randomly apply one of the following six mutation choices to this stage:

- Add a view: if the stage does not have all four views, then randomly add a view not yet present
- Remove a view: if the stage has more than one view, then randomly remove an existing view
- Switch the view: if the stage has exactly one view, then switch the view to another
- Adjust the pillar / voxel size: multiply by either 0.8 or 1.2
- Adjust the number of channels: multiply by either 0.8 or 1.2
- Adjust the layer progression: increase or decrease the number of dense-normalization-ReLU repeats / U-Net scales

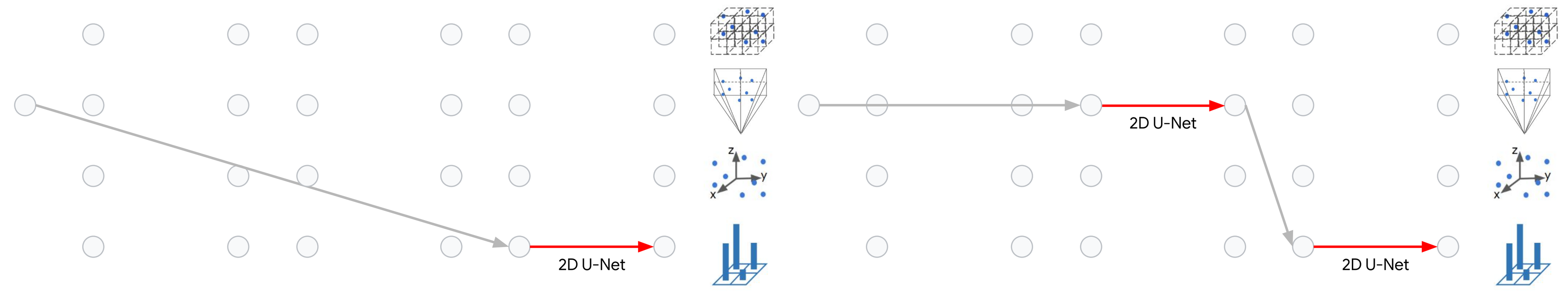
The first four focus on "transform"; the last two focus on "layer".

Experimental Results

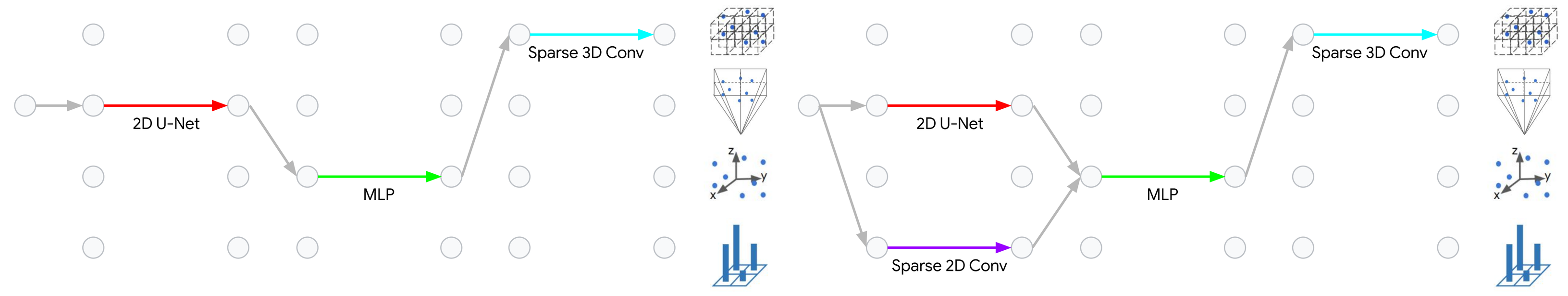
Improved Detection on Waymo Open Dataset

model	frame	Vehicle L1 AP 3D	Vehicle L1 AP BEV	Vehicle L1 AP latency	Pedestrian L1 AP 3D	Pedestrian L1 AP BEV	Pedestrian L1 AP latency
LaserNet		52.1	71.2	64.3	63.4	70.0	64.3
PointPillars		63.3	82.5	49.0	68.9	76.0	49.0
PV-RCNN		70.3	83.0	-	-	-	-
Pillar-based	1	69.8	87.1	66.7	72.5	78.5	66.7
PV-RCNN	2	77.5	-	300	78.9	-	300
RCD	1	69.0	82.1	-	-	-	-
MVF++	1	74.6	87.6	-	78.0	83.3	-
CenterPoint	2	76.7	-	-	79.0	-	-
PPC		65.2	80.8	-	75.5	82.2	-
RangeDet	1	72.9	-	-	75.9	-	-
PointPillars-like	1	67.6	85.3	-	-	-	-
LidarNASNet-P	1	**73.2**	**88.2**	-	-	-	-
RSN	1	75.2	87.7	46.5	77.1	81.7	21.0
LidarNASNet-R	1	**75.6**	**88.6**	49.3	**77.4**	**82.0**	22.6

Comparison of Warm Start and Evolved Architectures



(a) PointPillars-like (b) LidarNASNet-P



(c) Range Sparse Net (d) LidarNASNet-R

Lessons from Sampled Architectures

- Search space is non-trivial and challenging (bottom left figure)
- Mutating "transforms" results in larger performance changes than mutating "layers" only
- Later stages matter more; top-down views (voxel and pillar) influence detection AP positively, while perspective view negatively (bottom right figure)
- Sparse $\neq$ fast: More sparse branches result in smaller latency if pillar view, but larger latency if perspective view