Anytime-Lidar: Deadline Aware 3D Object Detection

Ahmet Soyyigit$^1$, Shuochao Yao$^2$, Heechul Yun$^3$

$^1,^3$ University of Kansas, Lawrence, KS

$^2$ George Mason University, Fairfax, VA
Perception in Autonomous Vehicles

• Object detection
  – Happens in 3D
    • Camera, Radar, Lidar, ...
  – Lidar-based deep neural networks
  – Timeliness
  – Time/accuracy requirements are environment dependent

Image credits (up): https://blogs.nvidia.com/blog/2017/11/23/safer-autonomous-driving/
Image credits (down): https://newsroom.intel.com/editorials/experience-counts-particularly-safety-critical-areas/#gs.8azpk6
Lidar-based Object Detection DNNs

• Point cloud to 3D bounding boxes (End-to-end)
  – Examples: Voxelnet, SECOND, PointPillars, CenterPoint

• Challenges: High computational cost, deadline-unaware
Execution Time Analysis of PointPillars

• Timing of PointPillars*:

• High computational cost (>130 ms)
• No flexibility in execution timing

(*) Executed on Jetson AGX Xavier
Architecture of PointPillars (multi-head)

- Point Cloud Transform
- Backbone (RPN)
- Detection Head(s)*

- Block 1
  - Conv
  - Deconv

- Block 2
  - Conv
  - Deconv

- Block 3
  - Conv
  - Deconv
  - Concat 1,2,3

- Detection Head (Car)
- Detection Head (Traffic cone, Pedestrian)

- NMS

(*) B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, “Class-balanced grouping and sampling for point cloud 3d object detection,” CoRR, vol. abs/1908.09492, 2019.
Anytime Perception for Lidar-based Object Detection DNNs

• Enable dynamic time and accuracy tradeoff
• Prior work on anytime perception
  – Image-based, mostly object classification [1-6]
• Our key contribution
  – First work to enable anytime perception in the lidar domain
  – Novel scheduler framework: Accuracy + Timeliness

[1] S. Heo et al., “Real-time object detection system with multi-path neural networks,” in 2020 RTAS
[2] J.-E. Kim et al., “Anytimenet: Controlling time quality tradeoffs in deep neural network architectures,” in 2020 DATE
[3] S. Bateni et al., “Apnet: Approximation-aware real-time neural network,” in 2018 RTSS
[4] S. Yao et al., “Scheduling real-time deep learning services as imprecise computations,” in 2020 RTCSA
[5] S. Lee et al., “Subflow: A dynamic induced-subgraph strategy toward real-time dnn inference and training,” in 2020 RTAS
[6] S. Liu et al., “Real-time task scheduling for machine perception in intelligent cyber-physical systems,” IEEE Transactions on Computers, pp. 1–1, 2021.
Outline

• Introduction
• Anytime-Lidar
• Evaluation
• Conclusion
Anytime-Lidar

• Enable anytime perception for lidar-based object detection DNNs
  1. Imprecise computation on the backbone
  2. Scheduling of detection heads
  3. Predicting past results of skipped heads
  4. Scheduling the above three
Imprecise Backbone

• Time and accuracy trade-off by skipping blocks
  – Added early exists to skip block 3 or blocks 2+3
  – Each block takes equal time
  – Take advantage of multi-block structure
Schedulable Detection Heads

• Allow skipping a subset of detection heads
  – Linearly save time from convolutions and NMS

• Address safety concerns
  – Proper det. head scheduling
  – Projection
Projection

- Project the past results of skipped det. heads to the current frame
- Projection/CPU - NN/GPU parallel execution
Scheduling

• Maximize detection accuracy while meeting the deadline with two-phase scheduler.

Time/accuracy statistics collected offline

Previous det. head selections and results

Backbone, det. heads and projection configuration
Scheduling

• First scheduling phase: Determine the number of backbone blocks and the number of detection heads to run

• Done using time/accuracy statistics collected offline

| RPN blocks | Detection heads |
|-----------|-----------------|
| 1         | 1 | 2 | 3 | 4 | 5 | 6 |
| 1         | 30.9 | 42.2 | 52.2 | 62.1 | 70.6 | 78.2 |
| 2         | 46.3 | 56.8 | 66.9 | 76.8 | 85.4 | 93.2 |
| 3         | 61.8 | 71.9 | 81.8 | 92.0 | 100.6 | 107.9 |

* Numbers are in milliseconds.

WCET table

| RPN blocks | Detection heads |
|-----------|-----------------|
| 1         | 1 | 2 | 3 | 4 | 5 | 6 |
| 1         | 67.0 | 67.5 | 70.7 | 74.4 | 79.2 | 80.6 |
| 2         | 75.4 | 77.5 | 82.1 | 88.2 | 91.9 | 93.3 |
| 3         | 79.8 | 84.9 | 90.7 | 95.6 | 98.9 | 100.0 |

Normalized accuracy table
Scheduling

- **Second scheduling phase**: Decide **which** detection heads to execute
  - Provides safety while optimizing accuracy
  - Priority = Age x Confidence

| Object                  | Age | Confidence | Priority |
|-------------------------|-----|------------|----------|
| Car                     | 1   | 3.5        | 3.5      |
| Truck, Constr. Vehicle  | 2   | 0.7        | 1.4      |
| Bus, Trailer            | 3   | 0.6        | 1.8      |
| Barrier                 | 3   | 2.0        | 6.0      |
| Motorcycle, Bicycle     | 4   | 1.2        | 4.8      |
| Traffic cone, Pedestrian| 1   | 4.5        | 4.5      |

\[ A \times C = P \]

\[
\begin{align*}
1 \times 3.5 &= 3.5 \\
2 \times 0.7 &= 1.4 \\
3 \times 0.6 &= 1.8 \\
3 \times 2.0 &= 6.0 \\
4 \times 1.2 &= 4.8 \\
1 \times 4.5 &= 4.5
\end{align*}
\]
Outline

• Introduction
• Anytime-Lidar
• Evaluation
• Conclusion
Evaluation

• Implemented by modifying Multi-head PointPillars (OpenPCDet*, PyTorch)
• Evaluated on NVIDIA Jetson AGX Xavier
  – 512-core Volta iGPU
  – 8 core ARM v8.2 64-bit CPU
  – 16 GBs of RAM
• Evaluated using nuScenes dataset
  – Used ten scenes each being 20 seconds

(*) OpenPCDet Development Team, “OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds,” https://github.com/open-mmlab/OpenPCDet, 2020.
Evaluation

• Divide the dataset of ten scenes into two equal sets
  – Calibration set
  – Testing set

• Collect time/accuracy statistics for all requiring methods (calibration)

• For each method being evaluated:
  – For each deadline in a list of deadlines from 140ms to 60ms:
    • Process all samples in the testing scenes one by one
    • Nullify detection results for samples where deadline is missed
    • Calculate NDS* (nuScenes Detection Score)

(*) H. Caesar, et Al., “nuScenes: A multimodal dataset for autonomous driving,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11618–11628, 2020.
### Evaluation

- Methods used for comparison:

| Method          | Number of model parameters | Number of RPN blocks | RPN stage selection | Detection head scheduling            |
|-----------------|---------------------------|----------------------|---------------------|--------------------------------------|
| PointPillars-3  | 6078K                     | 3                    |                     | Circulating                          |
| PointPillars-2  | 2626K                     | 2                    |                     | Class scores sum                     |
| PointPillars-1  | 1723K                     | 1                    |                     | Aging + Ground Truth                 |
| MultiStage      |                           |                      |                     | Aging + Aged confidences             |
| RoundRobin      |                           |                      |                     |                                      |
| ClsScrSum       | 9235K                     | 3                    | ✓                   |                                      |
| NearOptimal     |                           |                      |                     |                                      |
| Ours            |                           |                      |                     |                                      |
Effect of Enabling Fine-grained Anytime Perception

- Meet tighter deadlines (60ms vs 100ms)
- Maintain superior accuracy all the time
Effect of Head Scheduling Method

- Disabled projection when testing
- Our method schedules the detection heads close to optimal

![Graph showing normalized accuracy vs deadline with overhead comparison]

| Method                      | Overhead (ms) |
|-----------------------------|---------------|
| ClsScrSum-NoPrj            | 4.75          |
| RoundRobin-NoPrj           | 0.50          |
| Ours-NoPrj                 | 1.50          |
| NearOptimal-NoPrj          | 70.3          |
| Current-NoPrj              | 61.1          |
| Current-Prj                | 62.8          |
| NearOptimal-Prj            | 66.9          |
Effect of Projection

- Projection can work with any head selection scheme and increases accuracy by 10% on average
Conclusion

• In this work, we presented:
  – A novel scheduling framework for lidar-based AI pipelines
    • Enables anytime perception through a combination of methods
      – Imprecise backbone, detection head scheduling, projection
  – We implemented our method on Multi-head PointPillars and evaluated its performance on Jetson AGX Xavier
  – Results show that our method significantly surpass baseline methods and enables anytime perception for lidar-based AI pipelines

• GitHub Link: https://github.com/CSL-KU/Anytime-Lidar
Thank You

Disclaimer:
This research is supported in part by NSF grants CNS1815959, CPS-2038923, and CPS-2038658

More details can be found in the following publication.
Ahmet Soyyigit, Shuochao Yao, Heechul Yun. “Anytime-Lidar: Deadline Aware 3D Object Detection.” IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), IEEE, 2022