AGILE3D: Adaptive Contention- and Content-Aware 3D Object Detection for Embedded GPUs

Pengcheng Wang$^{\alpha}$, Zhuoming Liu$^{\beta}$, Shayok Bagchi$^{\gamma}$, Ran Xu$^{\delta}$, Saurabh Bagchi$^{\alpha}$, Yin Li$^{\beta}$, Somali Chaterji$^{\alpha}$

α Purdue University, β University of Wisconsin-Madison, γ West Lafayette Jr./Sr. High School, δ NVIDIA

Introduction

3D object detection using LiDAR-generated point clouds is crucial for the perceptual systems in autonomous vehicles, offering detailed environmental insights. However, this process poses significant computational demands.

- Adaptability: The system must dynamically adapt to changing latency needs, influenced by diverse environmental conditions.
- Computational Challenges: Executing inference on embedded GPUs is essential to reduce end-to-end latency and maintain data privacy.
- Contention- and Content-Aware Scheduling: Choose an execution branch that is optimal based on current resource contention and input content.

Keywords: Autonomous System, Point Clouds, 3D Object Detection, Embedded GPUs.

Motivational Study

3D vs 2D Object Detection: Our study explores system design challenges in 3D object detection with point clouds, comparing 2D and 3D approaches, and highlights the complexities of tuning key parameters.

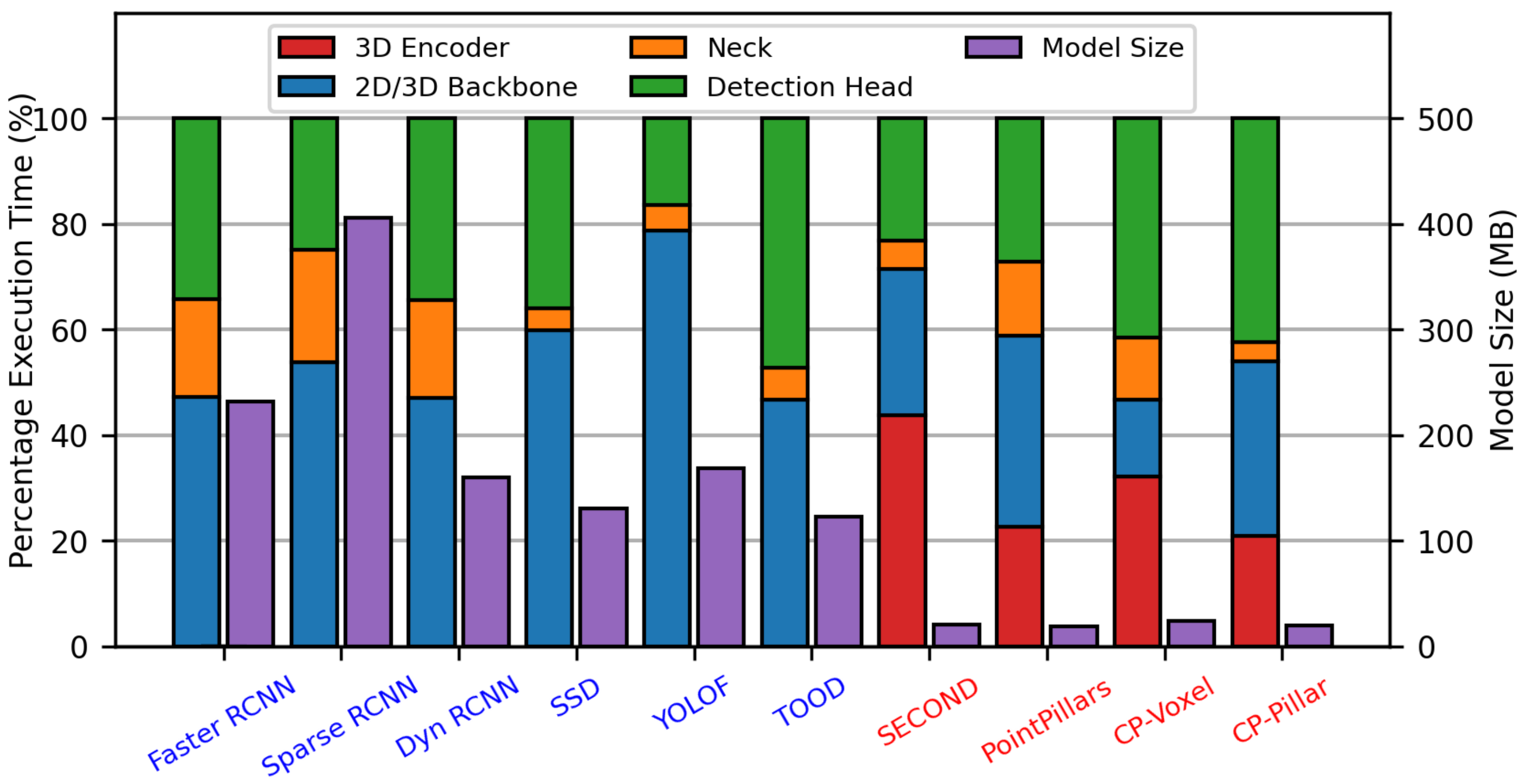

Figure 1. Comparison of execution time and model size for 2D and 3D models. 3D models require higher computation for point clouds but offer better memory efficiency, averaging 20.53 MB versus 203.32MB for 2D models.

In 3D models, the 3D Encoder, which includes the voxel layer, voxel encoder, and 3D spatial encoder, dominates the computational latency of the 2D/3D CNN Backbone. This highlights the key differences between 2D and 3D models, including the interdependencies among modules and variations in memory consumption, emphasizing the need for novel approaches in 3D detection to address these challenges.

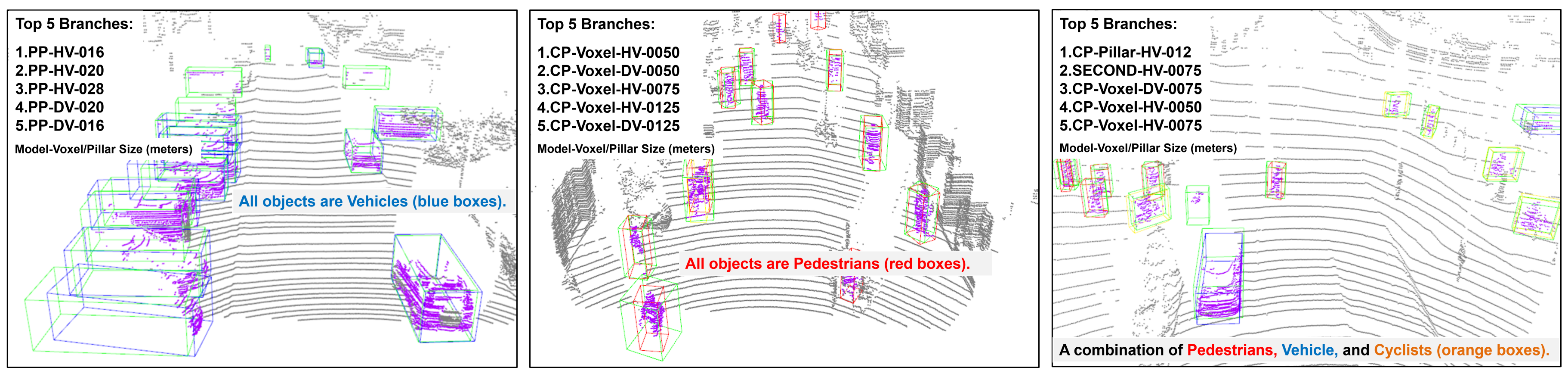

Figure 2. Visualization of diverse point clouds: Vehicles [L], Pedestrians [M], and a mix of Pedestrians, Cyclists, and Vehicles [R]. Ground-truth boxes are green, with top branch predictions for Pedestrians red, Cyclists orange, and Vehicles blue. The top-5 model ranking varies by context: for [L], pillar-based models prevail due to the straightforward geometry; for [M], voxel and center-based detection excel due to their robustness to smaller, varied orientations; and for mixed-object scenes [R], the complexity defies simple explanations, motivating Agile3D's multi-branch and content-aware controller design.

Challenges

Embedded 3D detection faces three key challenges:

1. Computational resources are limited on embedded devices, which often run multiple applications in parallel, leading to resource contention.
2. The transition from 2D to 3D detection introduces new complexities, including the need for a specialized 3D Encoder for feature extraction, which involves operations like voxelization and sparse convolutions that increase latency and resource usage.
3. Real-time systems require adaptability to dynamic environments. Modern autonomous systems, often equipped with multiple sensors like cameras, LiDARs, and radars, may have multiple applications sharing the same hardware resources. Additionally, input data from each sensor varies from scene to scene, creating both internal (resource contention) and external (content variability) dynamics. These dynamics require an adaptive solution.

Adaptive 3D Object Detection System

Agile3D incorporates a Multi-branch Execution Framework (**MEF**) and a Contention- and Content-Aware RL-based controller (**CARL**). It tunes key 3D components (Fig. 3) to balance latency and accuracy by selecting the optimal branch at runtime. Our scheduler aims to select the optimal execution branch that meets the latency SLO and maximizes accuracy, achieved through Supervised initial training and Direct Preference Optimization (DPO) fine-tuning .

Model Zoo: The MEF architecture begins with 80+ detector models. We prune suboptimal models (slow inference or low accuracy) during the initial training stage, leaving 50+ models that balance speed and precision. The 50+ models collectively require less than 8GB of memory, well within the 32GB capacity of the embedded GPU. All models are preloaded during system initialization, minimizing runtime latency, with branch switching overhead averaging only 1 ms.

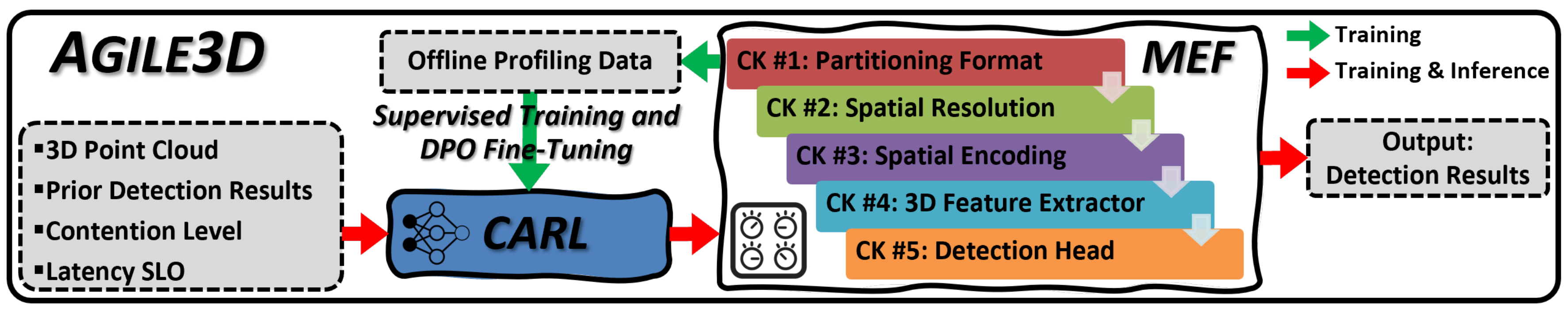

Figure 3. Agile3D integrates MEF and CARL for dynamic branch selection based on inputs, contention levels, and latency SLOs. RL-based training along with five novel control knobs ensure adaptability across diverse scenarios.

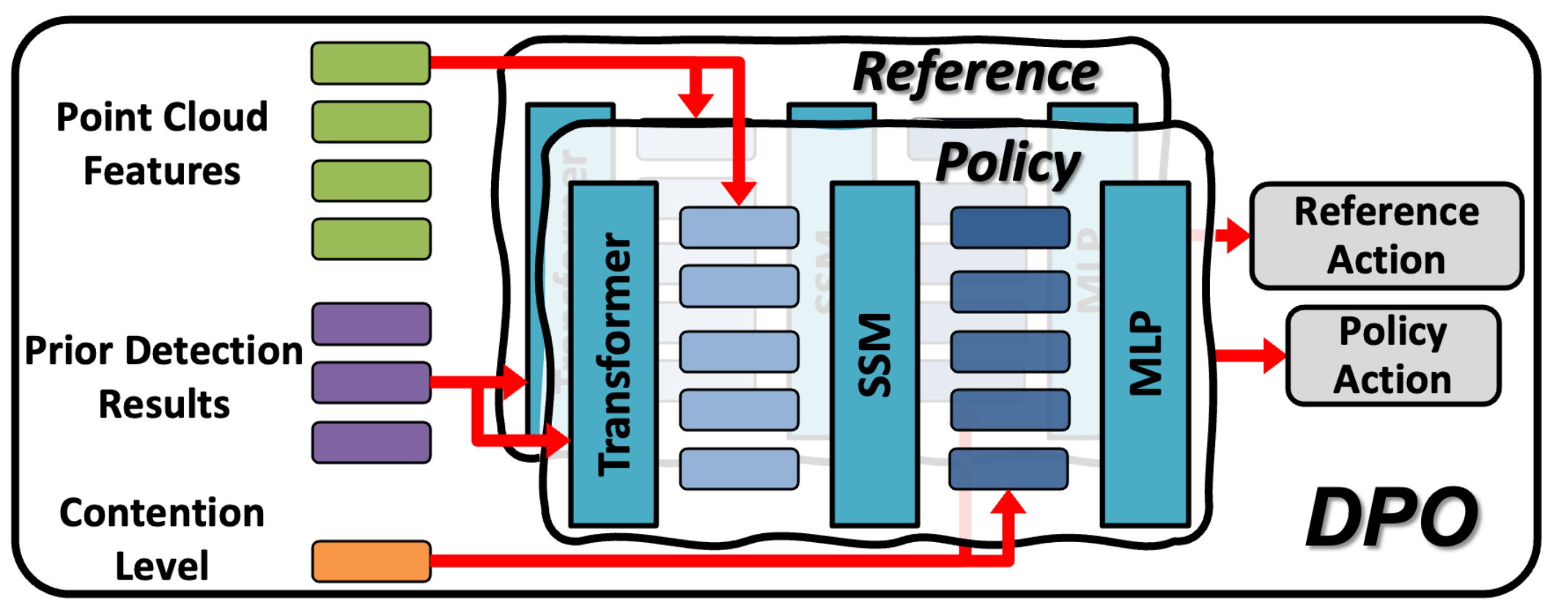

Figure 4. The CARL controller integrates GD-MAE for 3D features, transformers for prior detection embedding, SSM for temporal dependencies, and positional embeddings for latency objectives, enabling adaptive branch selection.

Our CARL controller dynamically schedules tasks by considering contention levels and frame-specific input content. It employs supervised training for initial learning, followed by DPO fine-tuning with preference labels provided by the Approximate Oracle controller using Beam Search. DPO refines branch selection through preference comparisons instead of absolute scores, ensuring efficient optimization.

Evaluation

Dataset	KITTI	nuScenes	Waymo
Task	Detection, Tracking	Detection, Tracking	Detection, Tracking
Classes	3 classes: Vehicle, Pedestrian, Cyclist	10 classes: car, truck, bus, bicycle, pedestrian, etc.	4 classes: Vehicle, Pedestrian, Cyclist, Sign
Frequency	10 Hz	20 Hz	10 Hz
# of Scenes	149	850	1000
Data Split	3,712 training and 3,769 validation point clouds (PCs)	28k and 6k annotated PCs for training and validation	150k and 40k annotated PCs for training and validation

Table 1. KITTI, nuScenes, and Waymo datasets' characteristics.

We evaluate Agile3D on two embedded GPUs, NVIDIA AGX Xavier and Orin, using three datasets, comparing it to state-of-the art baselines, including two system controllers: Chanakya [NeurIPS '24], LiteReconfig [EuroSys '22], and six 3D models: CenterPoint [CVPR'21], Part-A2 [TPAMI'20], SSN [ECCV'20], PointRCNN [CVPR'19], PointPillars [CVPR'19], SECOND [Sensors'18].

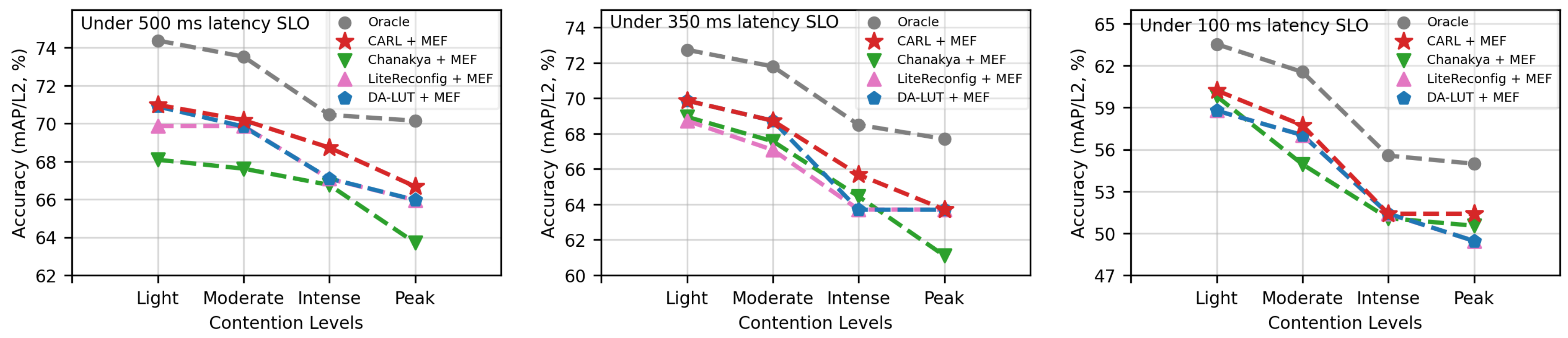

Figure 5. End-to-end evaluation of Agile3D across varying contention levels (*Light / Moderate / Intense / Peak*) and latency SLOs (500 ms [L], 350 ms [M], and 100 ms [R]) using the Waymo dataset and on Orin GPU. Agile3D consistently achieves superior accuracy, shining on the Pareto frontier across all contention levels and latency SLOs.

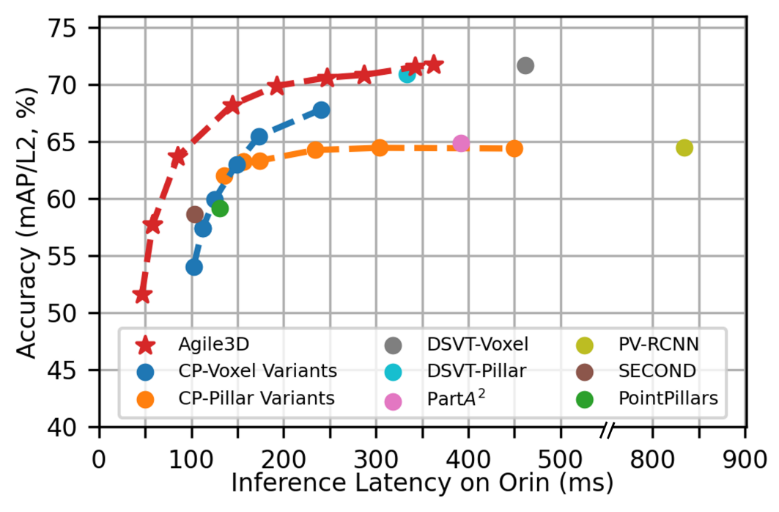

Figure 6. Agile3D vs. baselines on Waymo (Orin). Ours achieves 1-2.5% higher accuracy while adapting to latency SLOs of 50-350 ms, outperforming CP, PartA2, and PV-RCNN.

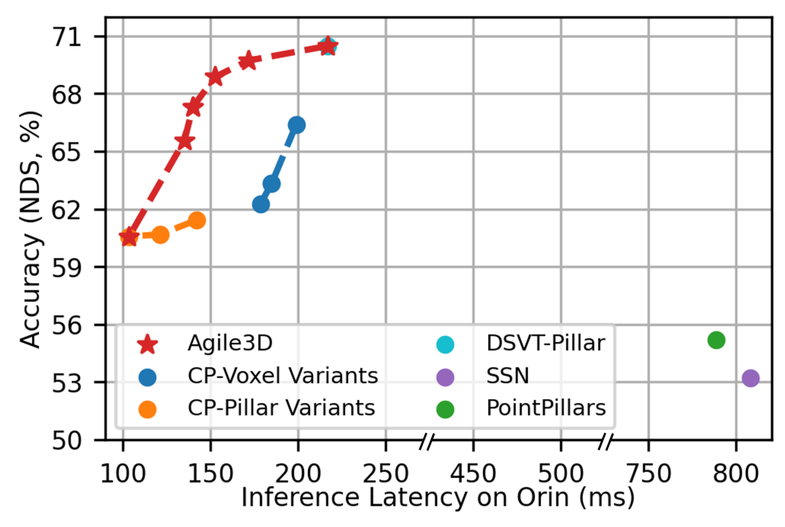

Figure 7. Agile3D vs. baselines on nuScenes (Orin). Ours has 7-16% higher accuracy than CP-Pillar, PP, and SSN, and meets SLOs of 100-250 ms, while baselines require over 400 ms.

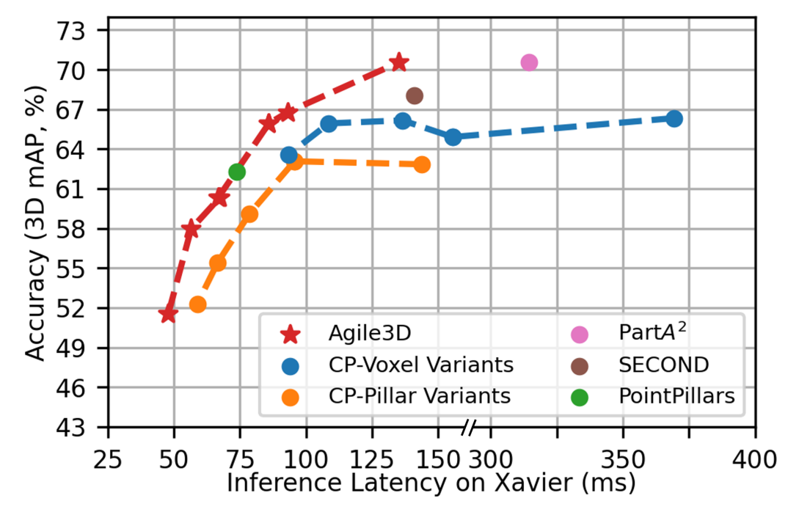

Figure 8. Agile3D vs. baselines on KITTI (Xavier). Agile3D adapts to latency SLOs of 50-150 ms, achieving 5-7% higher accuracy than PP and CP under the 100 ms SLO.

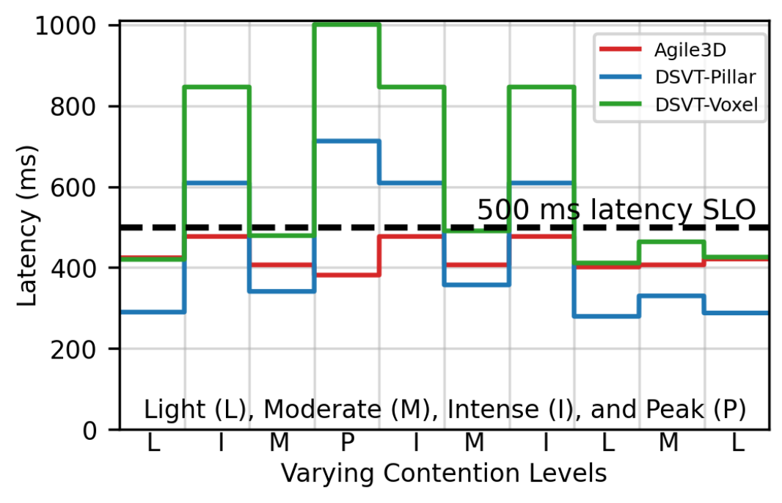

Figure 9. Agile3D adapts to changing contention levels on the Waymo test set on Orin under 500 ms latency

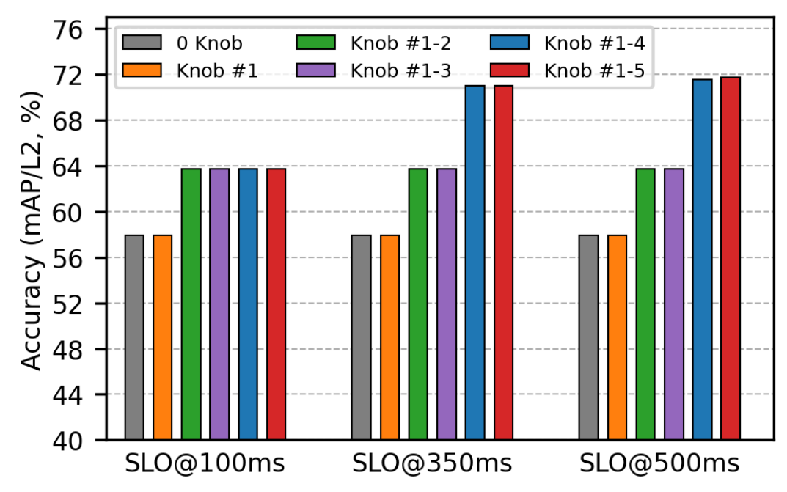

Figure 10. Agile3D on Waymo (Orin) under three latency SLOs: Activating more control knobs improves accu-

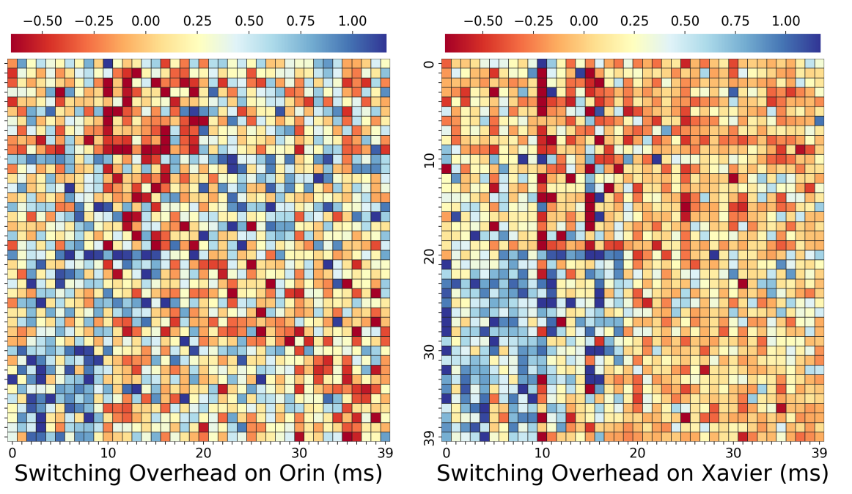

Figure 11. Switching overhead between branches. Mean overhead <1 ms with pre-buffered models.

Conclusion

- Agile3D, our adaptive 3D object detection system for embedded GPUs, excels in achieving SOTA accuracy while consistently meeting stringent runtime latency SLOs across diverse resource contention levels.
- By leveraging the MEF and CARL controller, Agile3D efficiently buffers all 3D models in GPU memory, enabling rapid model switching within 1 ms. The system features two complementary and innovative controllers: CARL controller for high contention scenarios and DA-LUT controller for contention-free scenarios.
- Across multiple datasets and hardware platforms, Agile3D demonstrates superior adaptability and better accuracy. It consistently meets latency SLOs—100-500 ms on Waymo (Orin), 100-250 ms on nuScenes (Orin), 33-75 ms on KITTI (Orin), and 50-100 ms on KITTI (Xavier).

NSF-CAREER (Award ID# 2146449) and NSF-FRONTIER (Award ID# 2333487)