KU-CSL
COMPUTER SYSTEMS LAB

MURAL: A Multi-Resolution Anytime Framework for LiDAR Object Detection Deep Neural Networks

Ahmet Soyyigit[1], Shuochao Yao[2], Heechul Yun[3]

[1,3] University of Kansas, Lawrence, KS

[2] George Mason University, Fairfax, VA

KU THE UNIVERSITY OF KANSAS

Real-Time 3D Object Detection with LiDAR

- SOTA method → Deep Neural Networks (DNN)

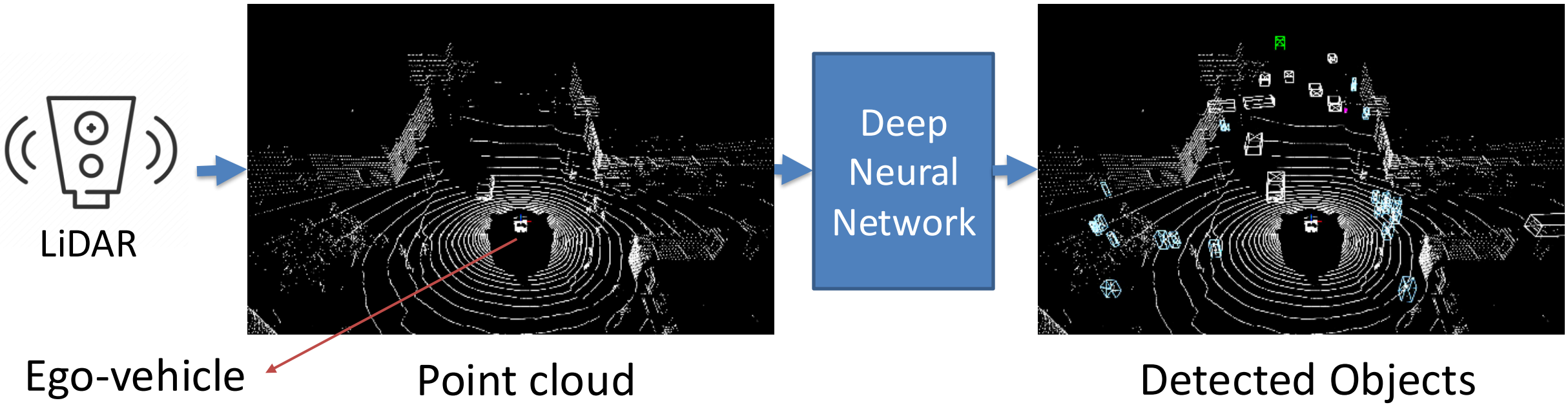
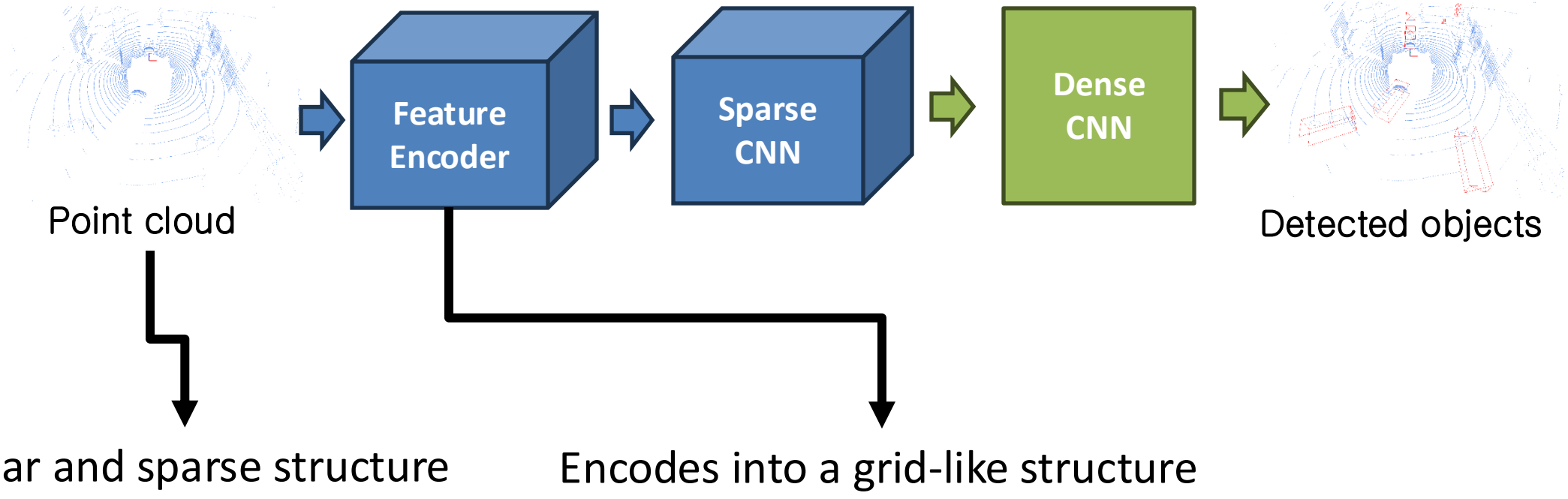


Image credits (leftmost): https://www.rdvsystems.com/lidar-icon-4/

LiDAR Object Detection DNNs



Latency and Accuracy

- High latency when executed on embedded systems, due to SWaP constraints.

- We can reduce latency with model compression.
  - Pruning, quantization, using lower input resolution, …

- HOWEVER, compression sacrifice accuracy for lower latency.
  - It makes a **trade-off between accuracy and latency.**

KU THE UNIVERSITY OF KANSAS

Latency and Accuracy

- Deployment on embedded systems requires a trade-off to be done.
- The optimal trade-off between latency and accuracy is **dynamic**.
  - Will explain why in the next slides.
- Our goal is to propose a novel dynamic latency and accuracy trade-off framework for LiDAR object detection DNNs.

Latency and Accuracy Requirements

- Simple, high-speed environment.
- Stale predictions are useless.

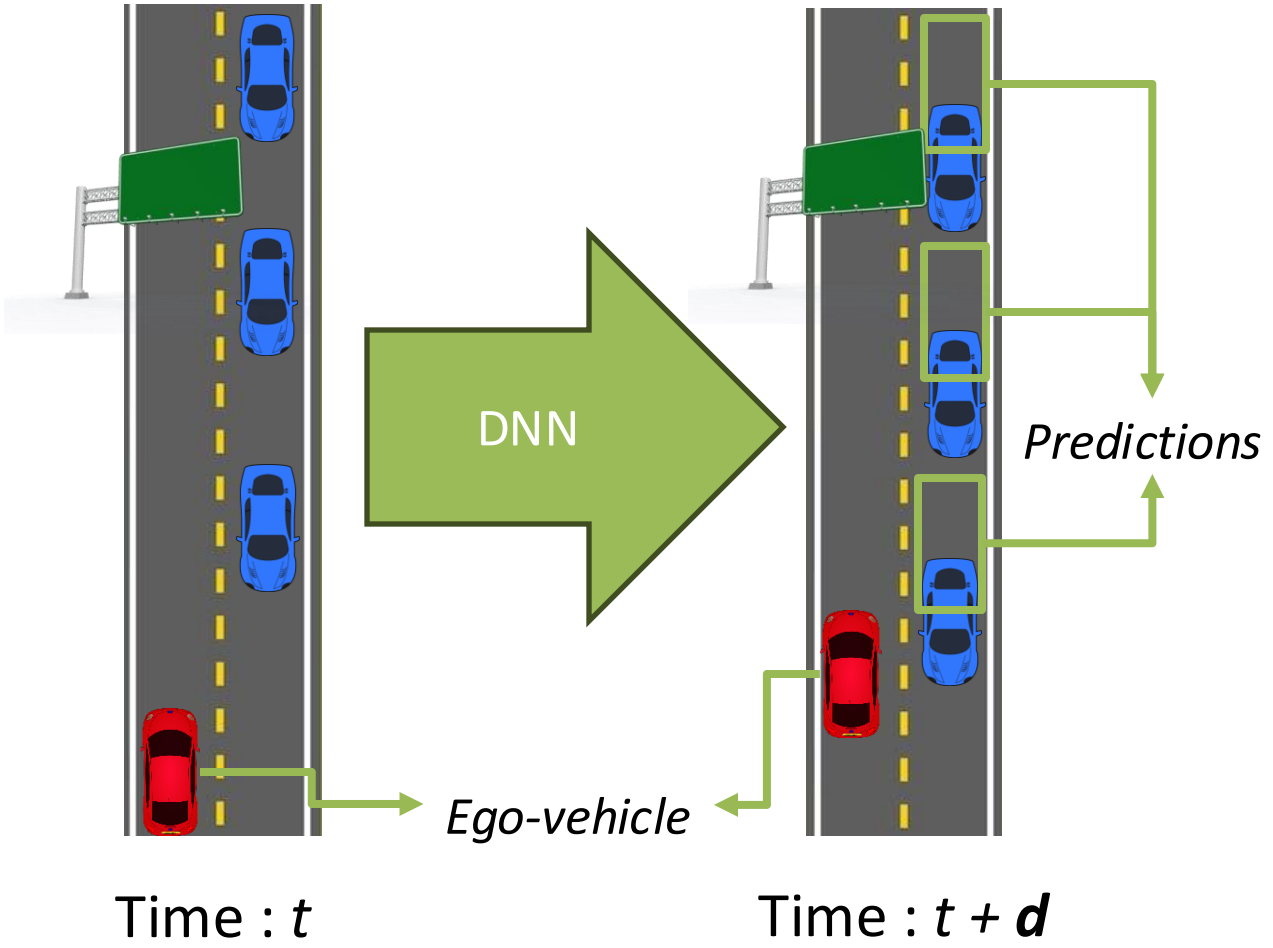


Image credits: https://tr.pinterest.com/pin/633811347586237182/

Latency and Accuracy Requirements

- Low latency prevents misalignment.
- Lesser accuracy is tolerable.

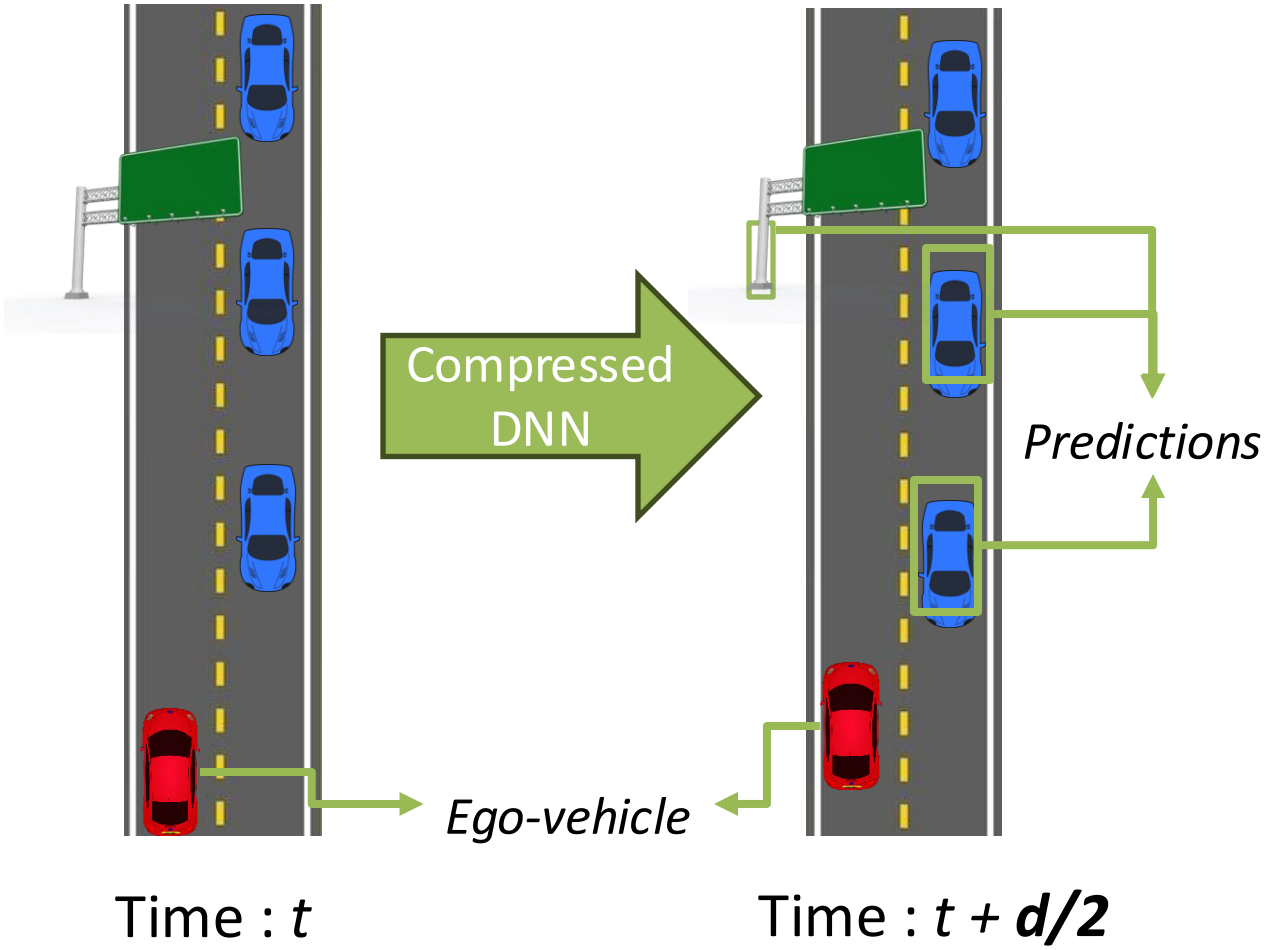
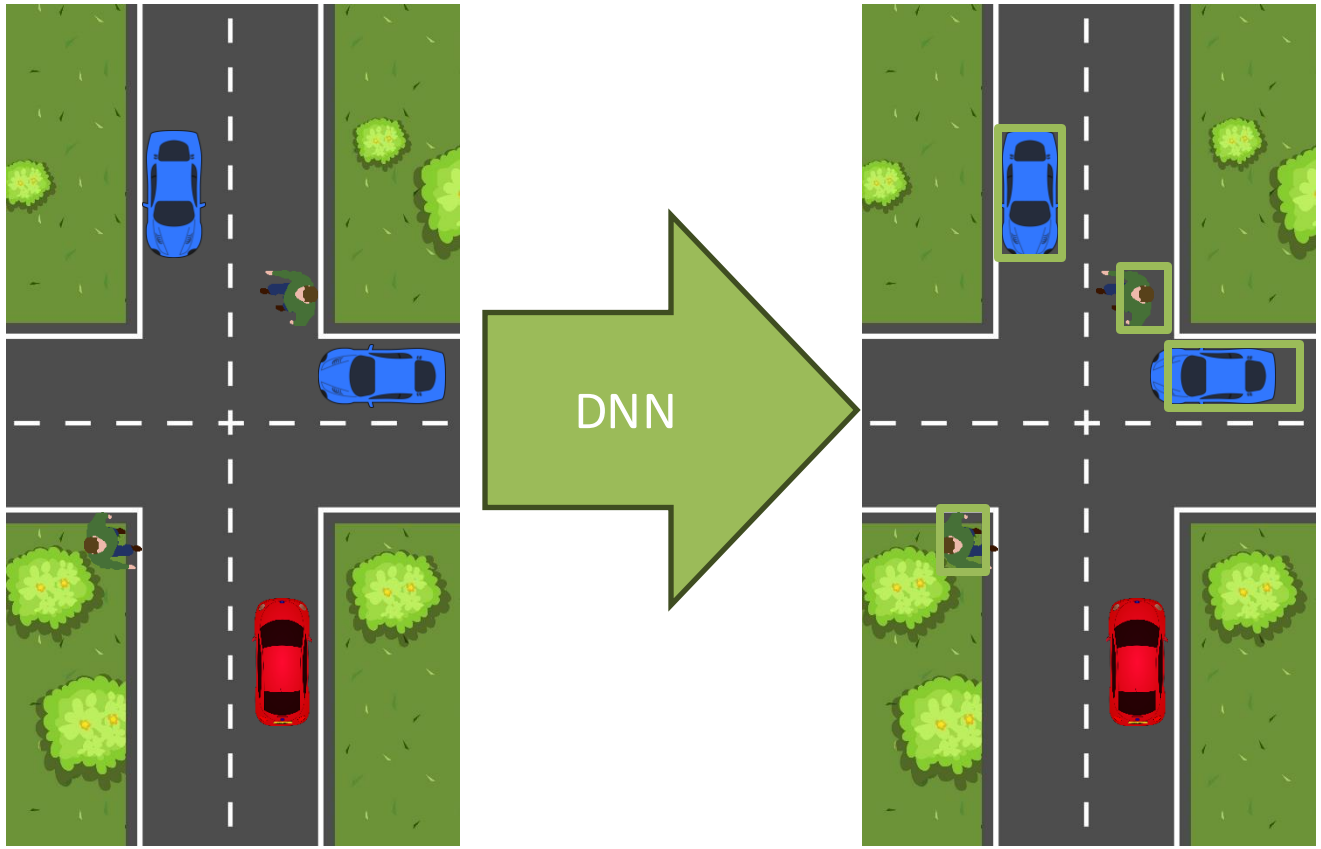


Image credits: https://tr.pinterest.com/pin/633811347586237182/

Latency and Accuracy Requirements

- Complex, low-speed environment.
  - Higher latency is tolerable.
  - Higher accuracy is favored.



Crossing image credits: brgfx on Freepik

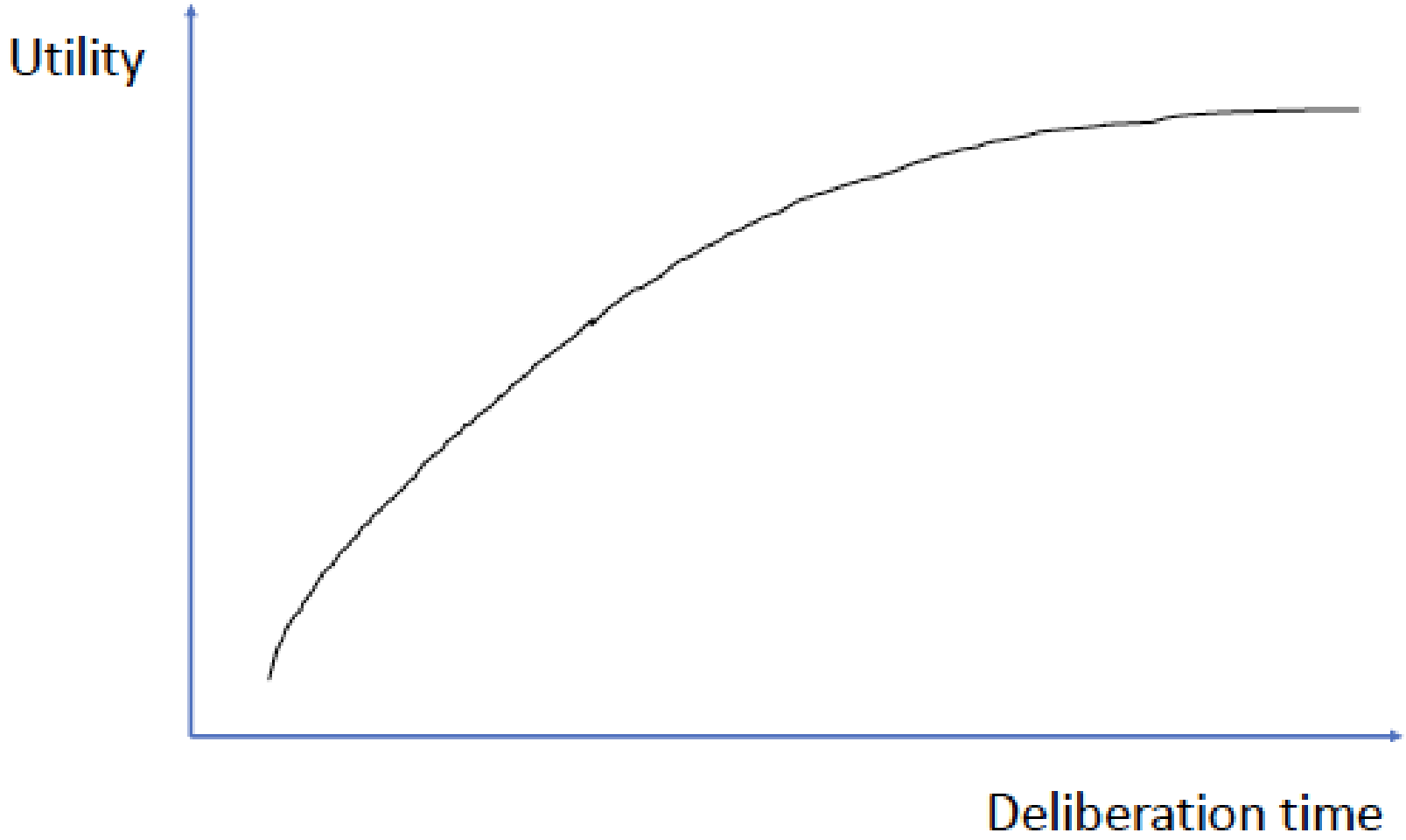
Latency and Accuracy Requirements

- Dynamically changing.

- Possible solution: Deploy alternative DNNs simultaneously.
  - High memory overhead.
  - Requires training and fine-tuning all models to be deployed.

Can we make dynamic trade-offs with a single DNN?

Anytime Algorithms



[3] Boddy, M., and Dean T. L. "Solving Time-Dependent Planning Problems." *In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989.

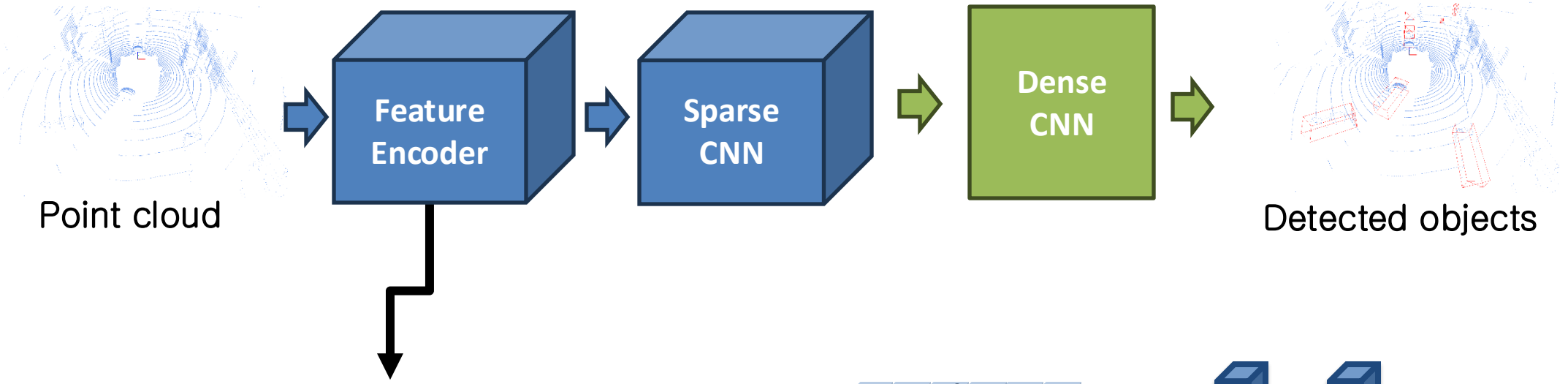
Anytime Computing with **Dynamic Input Resolution**

- Decide input resolution at run-time to make trade-offs between latency and accuracy for a single DNN.

- Prior work enabled this on DNNs that process camera images.
  - Did not explore it for the DNNs that process point clouds, which are architecturally different.

- Our work is the first to explore for LiDAR.

How Is Input Resolution Determined?

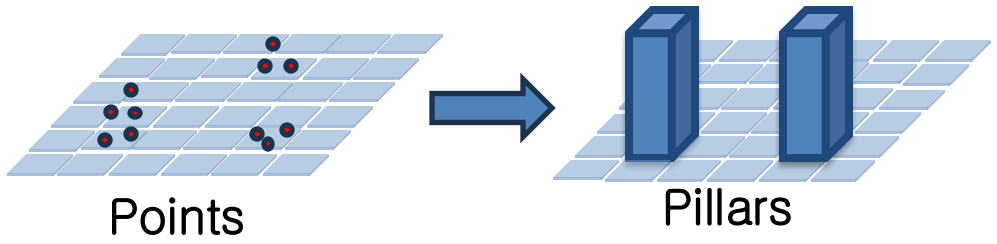


Encodes into a grid of pillars

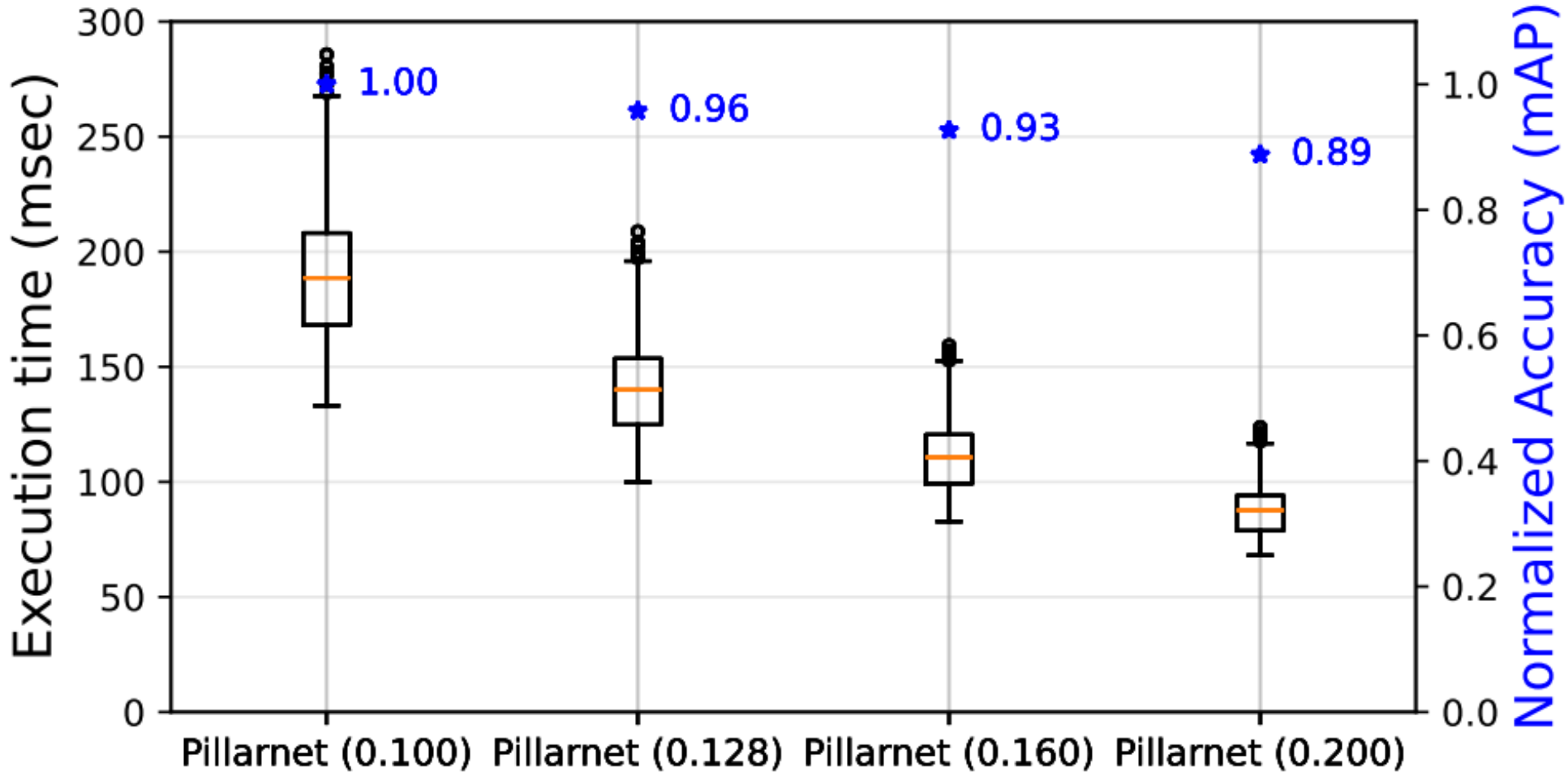
Pillar size determines

grid size

(i.e. input resolution)



Trade-offs with Dynamic Input Resolution



KU THE UNIVERSITY OF KANSAS

One Naive Approach

- Dynamically changing the pillar size for a high-accuracy model, Pillarnet (0.100).

- Results:

Inference pillar size (m^2)	Actual Normalized Accuracy (%) of Pillarnet (0.100)	Expected Normalized Accuracy (%)
0.100^2	100	**100**
0.128^2	79	**96**
0.160^2	41	**93**
0.200^2	18	**89**

KU THE UNIVERSITY OF KANSAS

How can we do it?

- In the baseline, DNN layers are trained while considering a single input resolution.
- Instead, they should be exposed to multiple resolutions.
- **Good news:** Convolutional layers can be trained to adapt multiple resolutions.*

KU THE UNIVERSITY OF KANSAS

*Y. Wang et al. “Resolution switchable networks for runtime efficient image recognition,” ECCV, 2020

How can we do it?

- Another issue is batch normalization (BN) layers, which normalize the data w.r.t. collected input statistics.

- These statistics differ concerning input resolution.
  - A single BN layer cannot adapt to multiple resolutions.

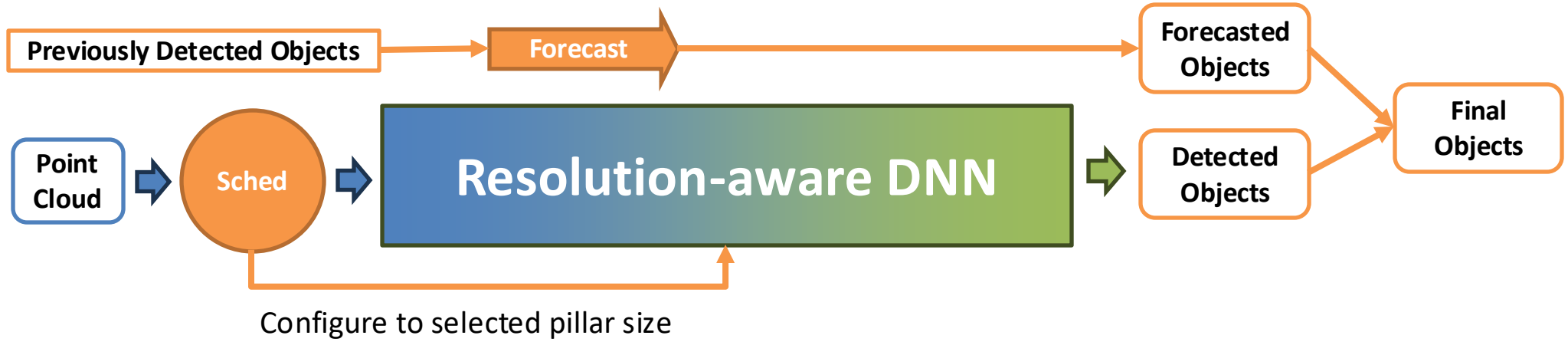
- Separate batch normalization layers are needed.*

KU THE UNIVERSITY OF KANSAS

*M Zhu et al. "Dynamic Resolution Network," NIPS, 2021

MURAL: MUlti-Resolution Anytime LiDAR

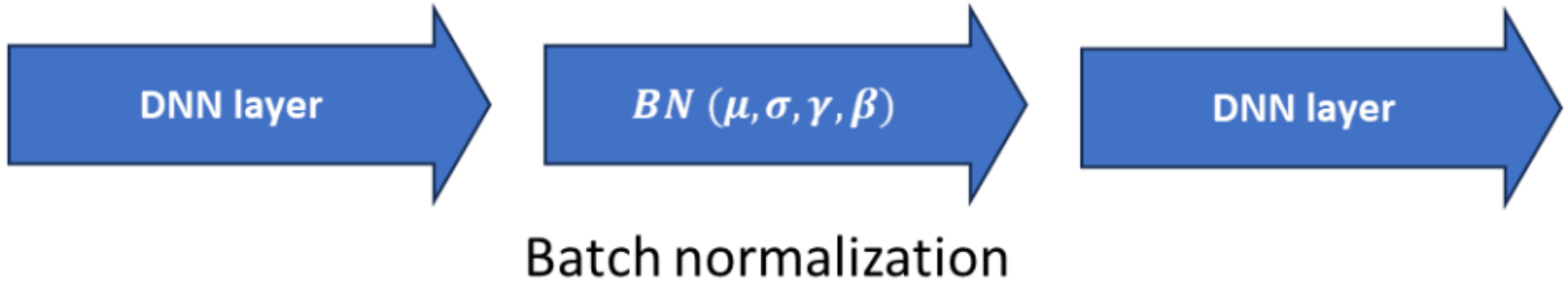
- MURAL enables anytime computing with ***dynamic pillar size scaling***
  - Makes DNN Resolution-aware (RA):
    - Modify batch normalization layers
    - Smartly train DNN to adapt multiple pillar sizes
  - Deadline-aware scheduler
  - Forecasting and dense CNN optimizations from our prior work, VALO*



*Soyyigit et. al., VALO: A Versatile Anytime Framework for LiDAR based Object Detection Deep Neural Network, EMSOFT'24

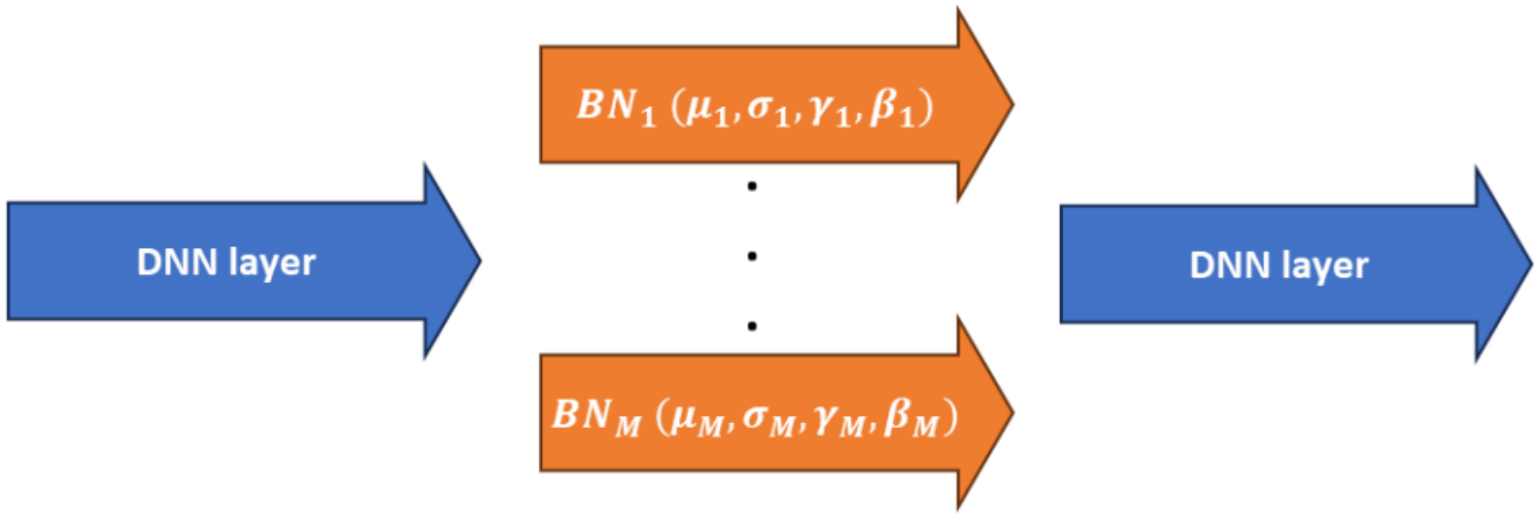
Batch Normalization (BN)

- A single BN cannot adapt to multiple input resolutions.



Resolution-aware Batch Normalization (BN)

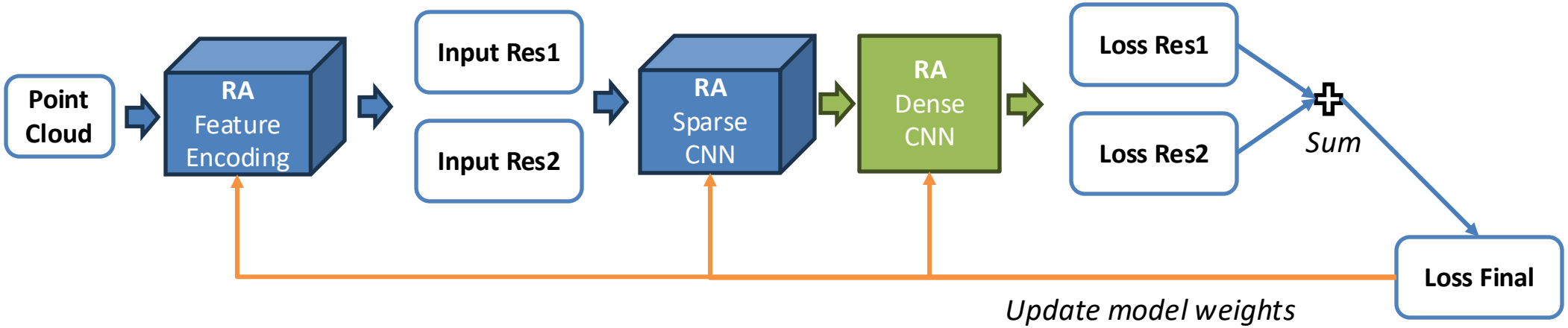
- Use separate BN for each input resolution.
- Incurs negligible memory overhead.



Resolution-aware batch normalization

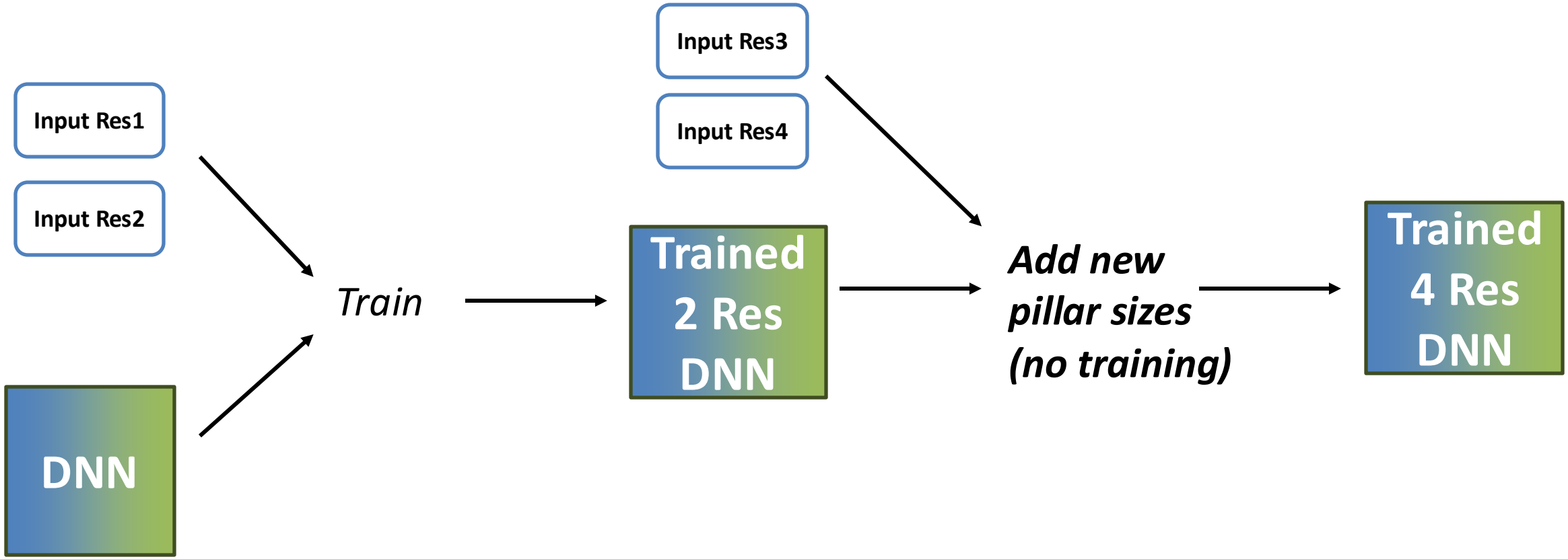
Training Procedure

- We utilize all targeted input resolutions to adapt the DNN to multiple resolutions.



- Importantly, we achieve same or comparable accuracy for all resolutions w.r.t. baselines targeting a single resolution.

Allow Introducing Input Resolutions After Training



*Y. Wang et al. "Resolution switchable networks for runtime efficient image recognition," ECCV, 2020

Introducing New Input Resolutions

- Add BN layers for each additional input resolution after the training.
- To predict their parameters, we model the relationship between existing BN parameters and input resolutions.
- We then do interpolation*/extrapolation on these models.

KU THE UNIVERSITY OF KANSAS

*Y. Wang et al. “Resolution switchable networks for runtime efficient image recognition,” ECCV, 2020

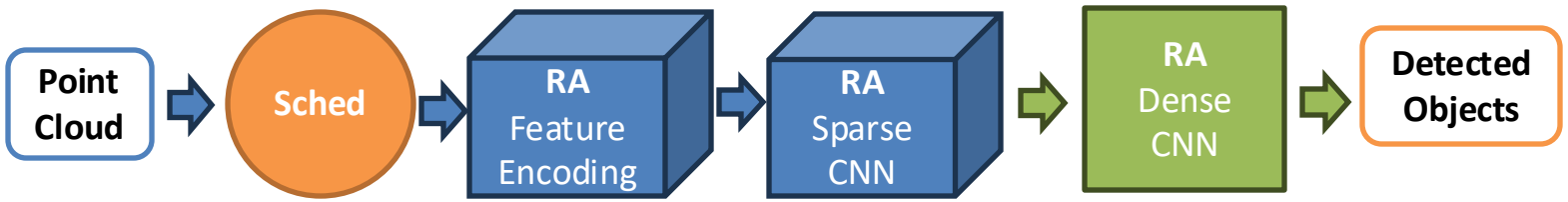
Deadline-Aware Scheduling

- Select smallest pillar size that can meet a given deadline
  - Assume higher resolution yields better accuracy

- Requires accurately predicting the latency for each pillar size at runtime
  - Challenging because latency of sparse CNN is dependent on the spatial alignment of pillars
  - We enhanced our prior work's* time prediction

*Soyyigit et. al., VALO: A Versatile Anytime Framework for LiDAR based Object Detection Deep Neural Network, EMSOFT'24

Latency Prediction



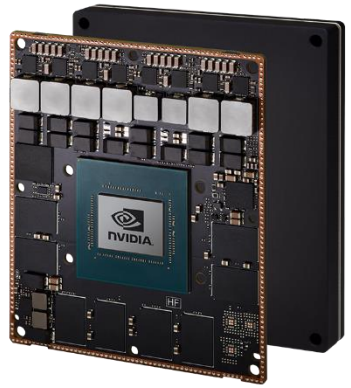
$$L_{PFE} + L_{SC} + L_{DC} + L_{PP}$$

Added

Before: *History-based*
Now: *Pinpoint prediction with rapid emulation*

Evaluation

- Applied MURAL on:
  - Pillarnet
    - Feature Encoder → Sparse CNN→ Dense CNN
  - PointPillars (CenterHead version)
    - Feature Encoder → Dense CNN
- Evaluated on:
  - NVIDIA Jetson AGX Xavier (30 W)
  - NVIDIA Jetson AGX Orin (30 W)
- Utilized nuScenes dataset



NUSCENES by Motional

KU THE UNIVERSITY OF KANSAS

Training Results for Pillarnet

- Better or comparable accuracy than separately trained baselines

Pillar size (m^2)	Pillarnet	MURAL
0.100^2	0.564	0.564 (+0.000)
0.128^2	0.537	0.560 (+0.023)
0.200^2	0.506	0.499 (-0.007)

Results are in mAP

Adding Pillar Sizes After Training

Pillar size (m^2)	Grid area	mAP
0.100^2	1024^2	0.564
0.109^2	928^2	0.568
0.128^2	800^2	0.560
0.151^2	672^2	0.540
0.200^2	512^2	0.499
0.263^2	384^2	0.390

Interpolated

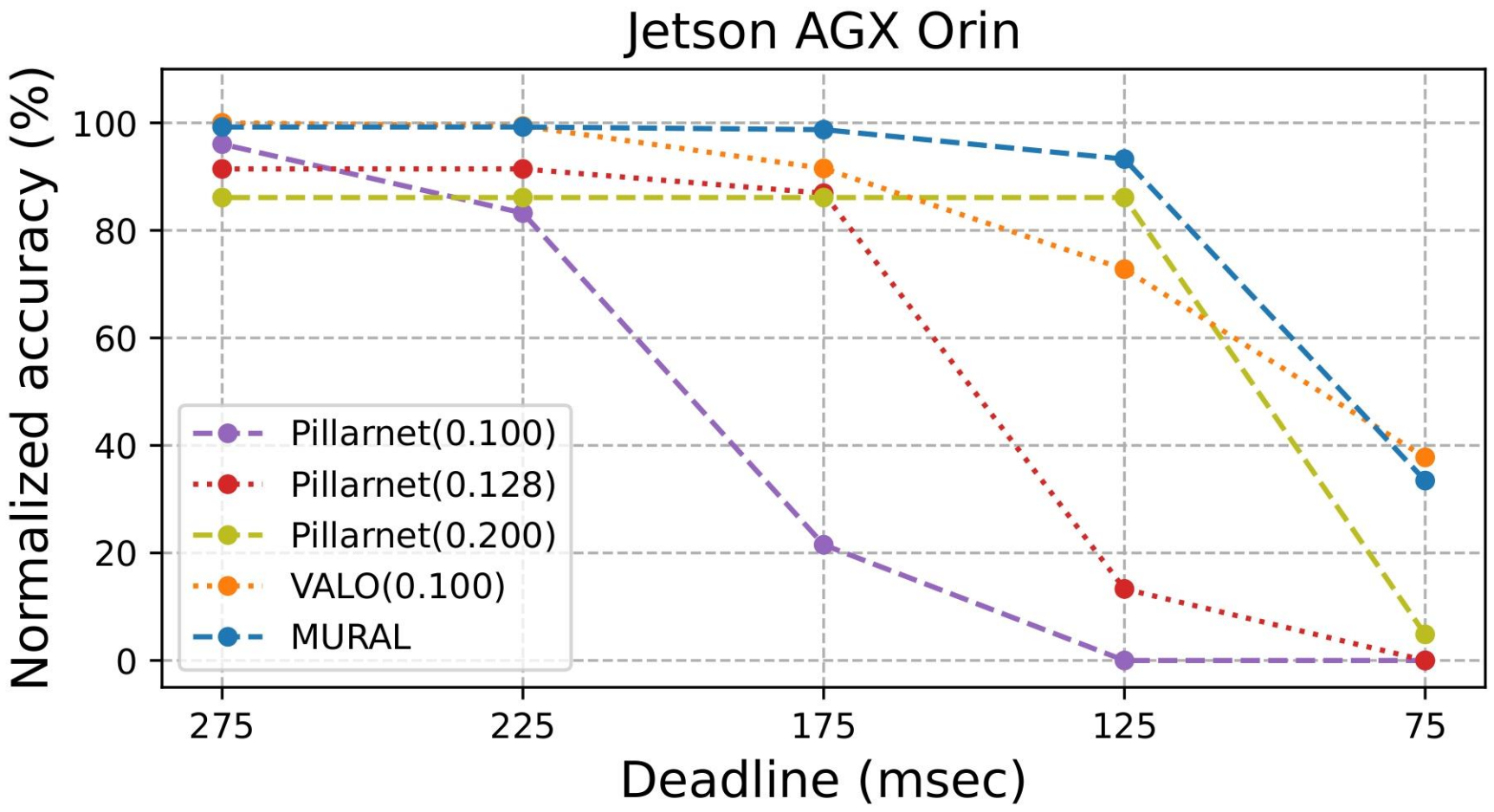
Extrapolated

All with a single model

Hard-deadline Evaluation

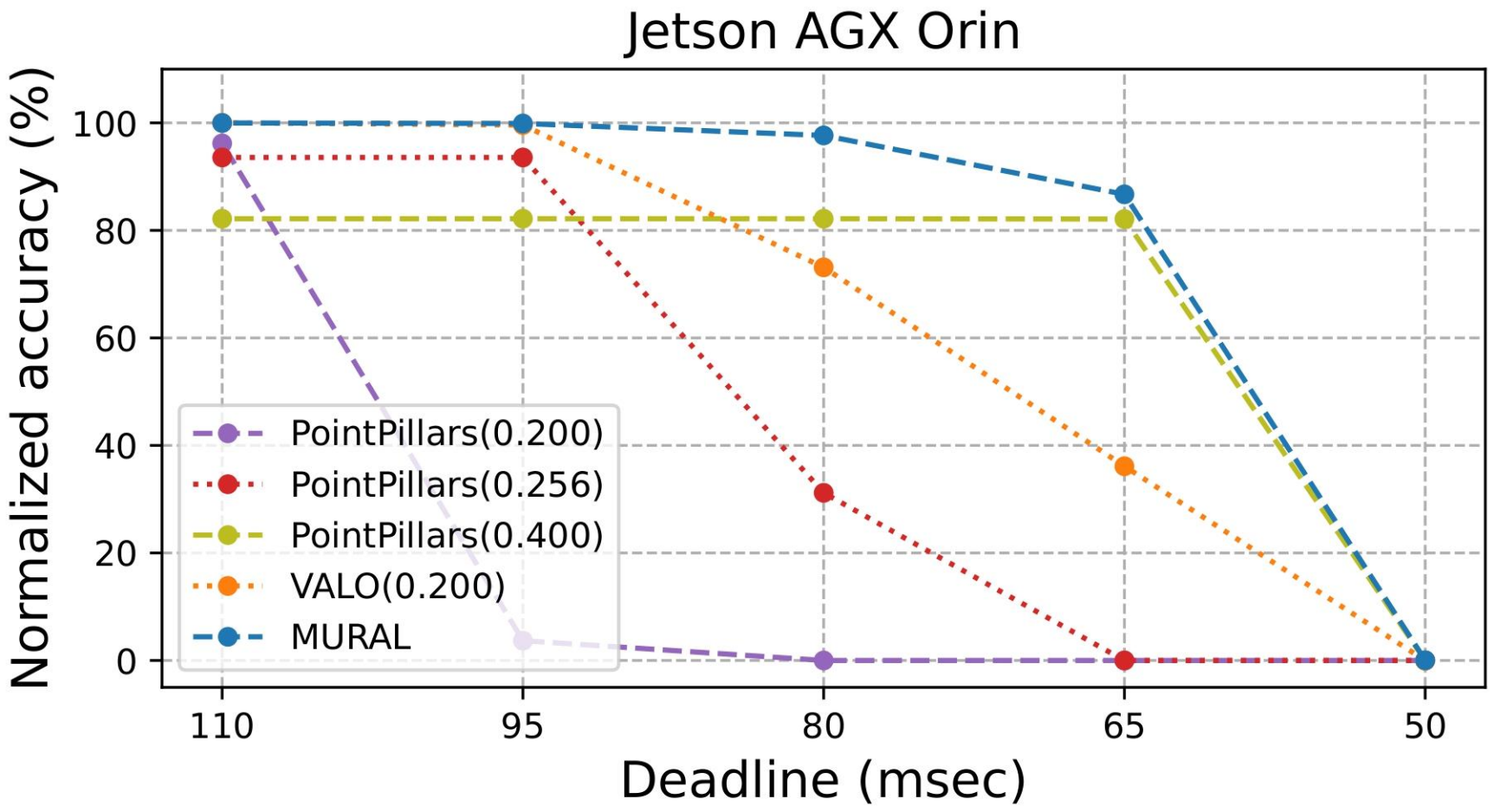
- Consider a range of hard deadlines.
- Nullify detection result on deadline misses.
- Compared MURAL against:
  - Baseline Pillarnet/PointPillars models.
  - Our prior SOTA data scheduling approach, VALO.

Pillarnet Results



KU THE UNIVERSITY OF KANSAS

PointPillars Results



Memory Requirement

	Pillarnet	PointPillars
Baseline	61.0 x 6 MB	24.0 x 6 MB
MURAL	61.4 MB	24.3 MB

Conclusion

- MURAL: First deadline-aware runtime resolution scaling framework for LiDAR detection DNNs
  - Resolution-aware batch normalization
  - Support arbitrary resolutions via BN inter/extrapolation
  - Deadline-aware resolution scheduling
- Balances accuracy and latency dynamically
- Memory-efficient: single model supports multiple resolutions
- Achieves state-of-the-art anytime performance
- Enables practical deployment on embedded platforms
- Code is available at: https://github.com/CSL-KU/MURAL

KU THE UNIVERSITY OF KANSAS

Thank You

Disclaimer:

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

Rest is appendix

LiDAR Object Detection DNNs

- High complexity incurs high latency when executed on embedded systems, due to SWaP constraints.

- Latency of a deployment-friendly* DNN on **Jetson AGX Orin (30 W)**:

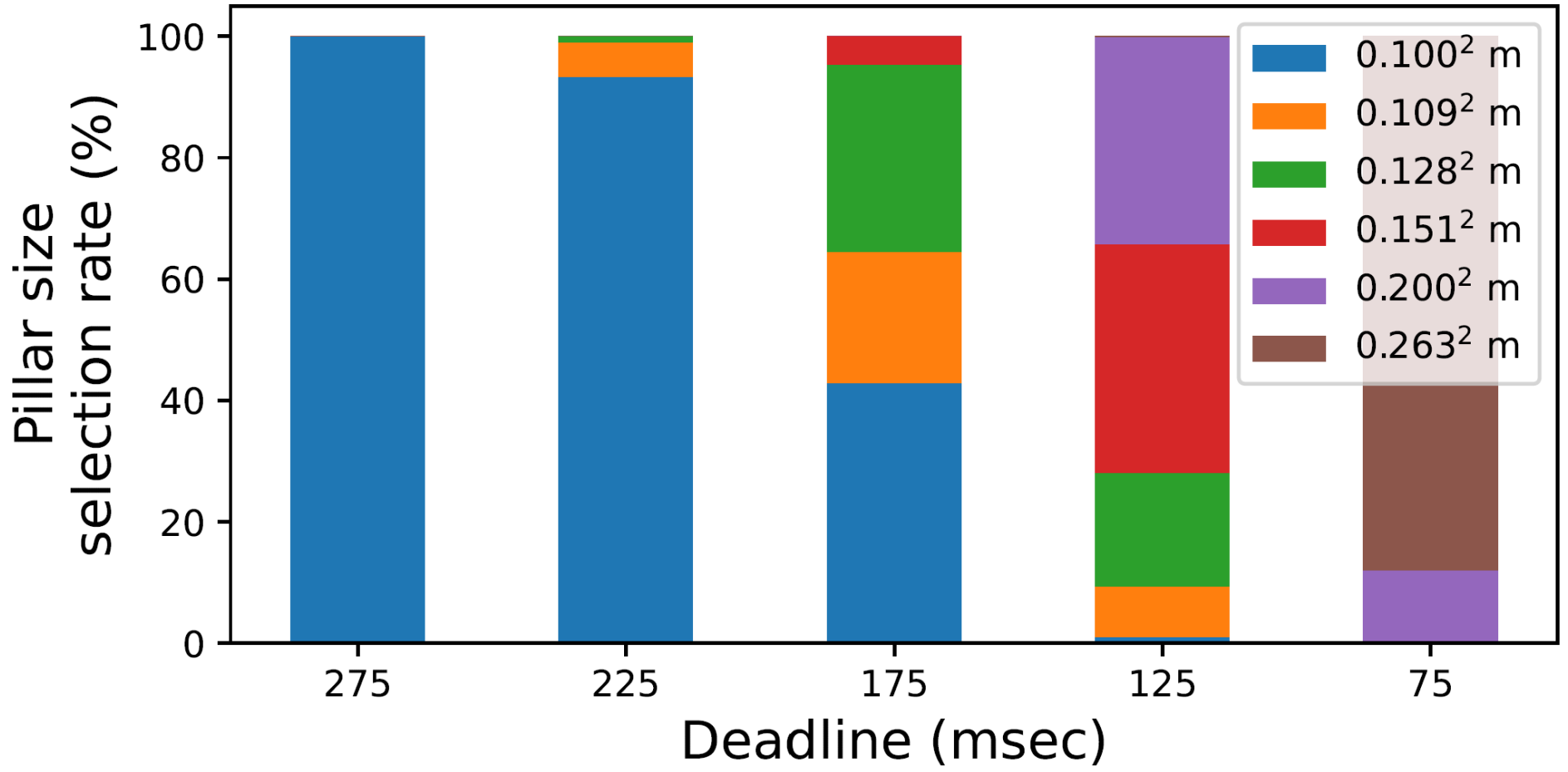
Min	Mean	Max
131 ms	186 ms	282 ms

In some driving scenarios, most tolerable is 100 ms.

KU THE UNIVERSITY OF KANSAS

Pillarnet, pillars size = 0.100*0.100

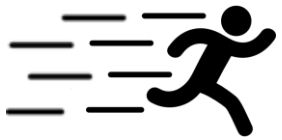
Pillarnet Results



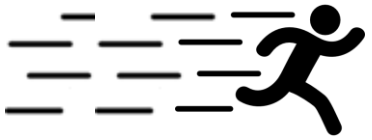
Trade-offs with Dynamic Input Resolution

Pillar size: 0.1m x 0.1m
Grid size: 600 x 600

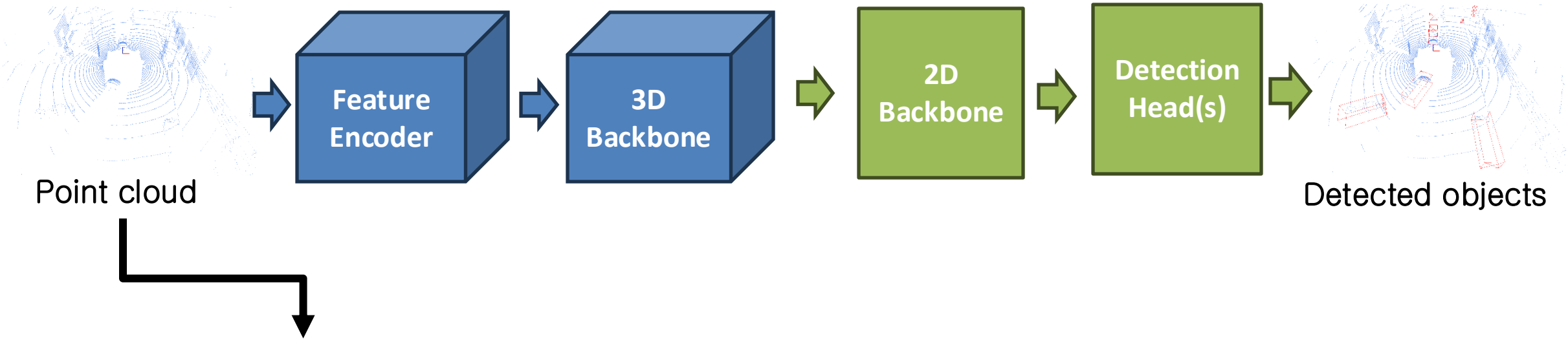
Pillar size: 0.3m x 0.3m
Grid size: 200 x 200



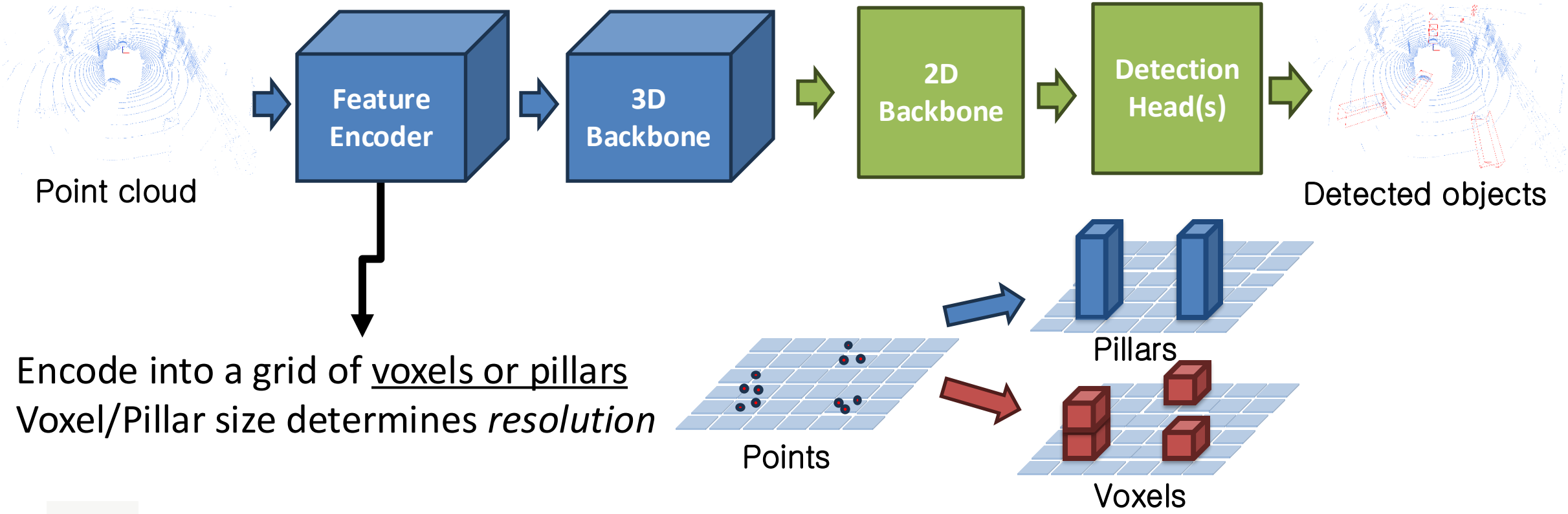
Pillar size: 0.5m x 0.5m
Grid size: 120 x 120



LiDAR Object Detection DNNs

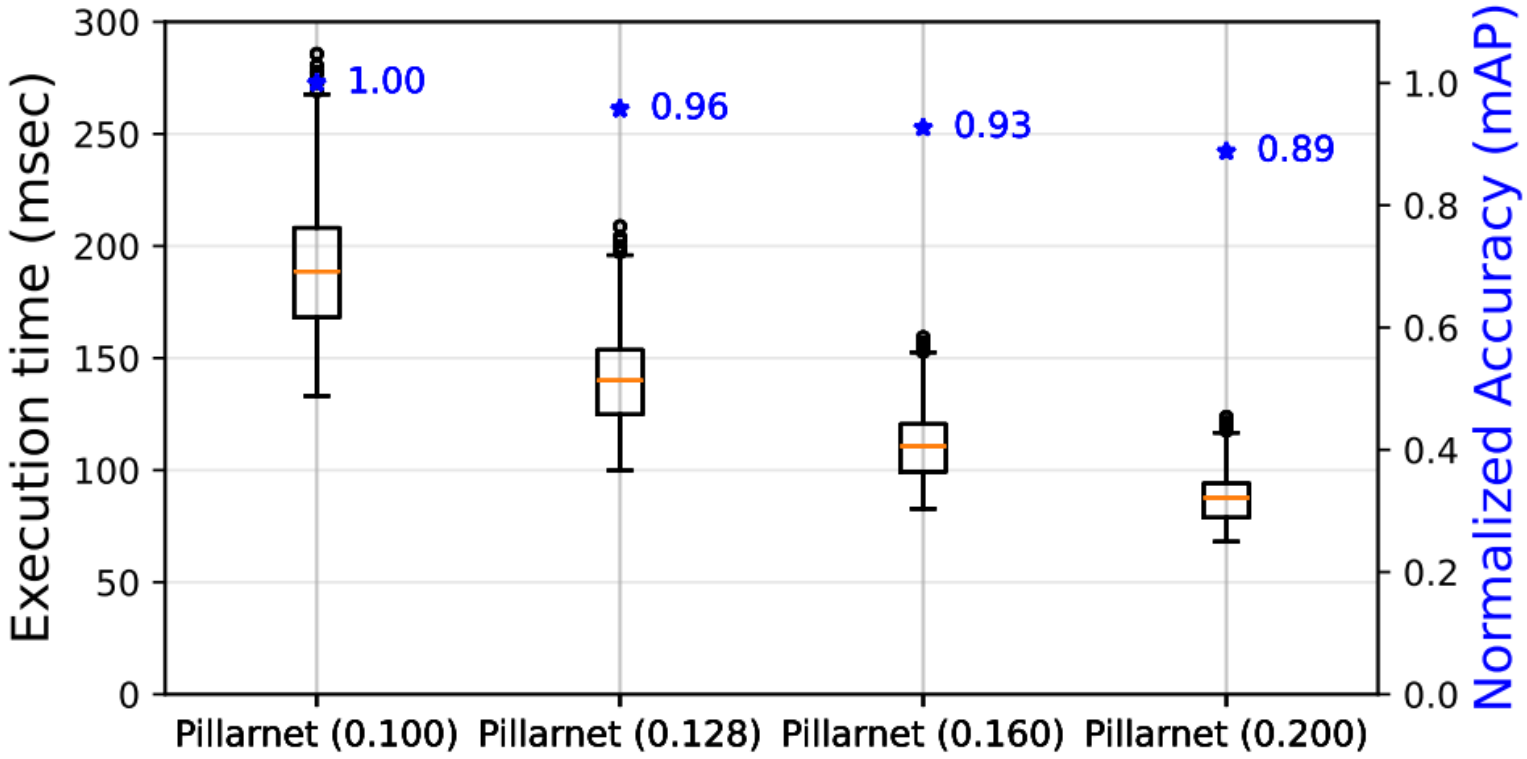


LiDAR Object Detection DNNs



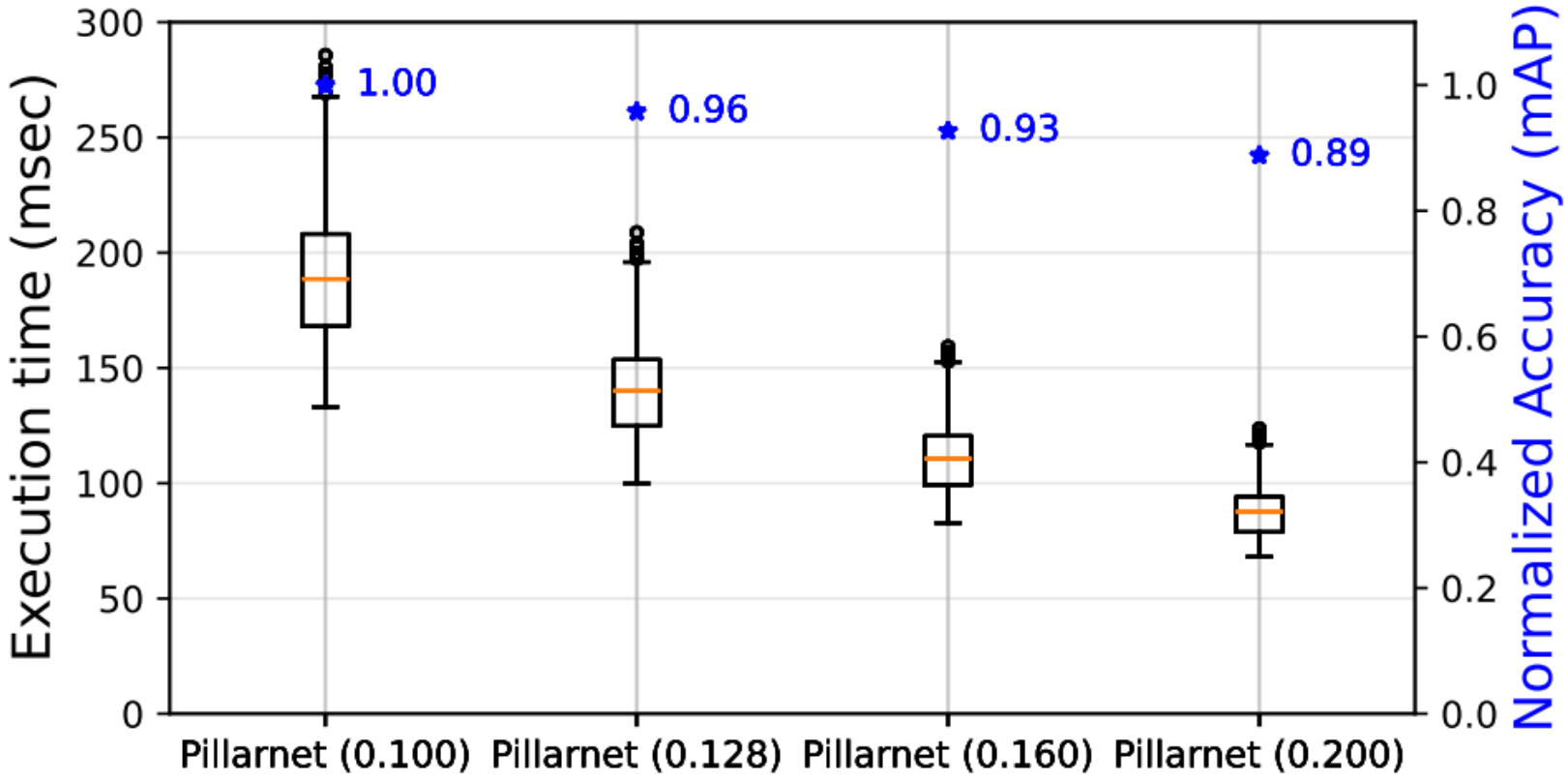
Pillar Size Scaling

- Excellent tradeoffs with multiple models.
- However, memory requirement grows linearly.

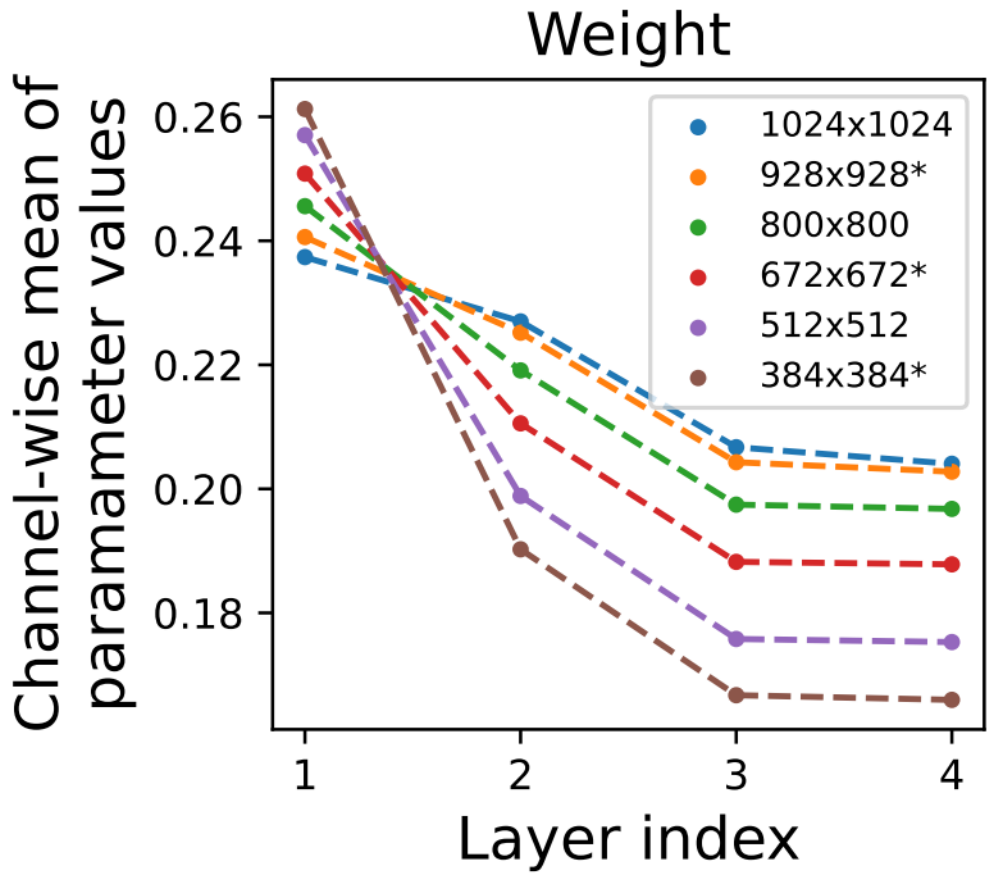


Trade-offs with Dynamic Input Resolution

- Excellent tradeoffs with multiple models
- However, memory requirement grows linearly.

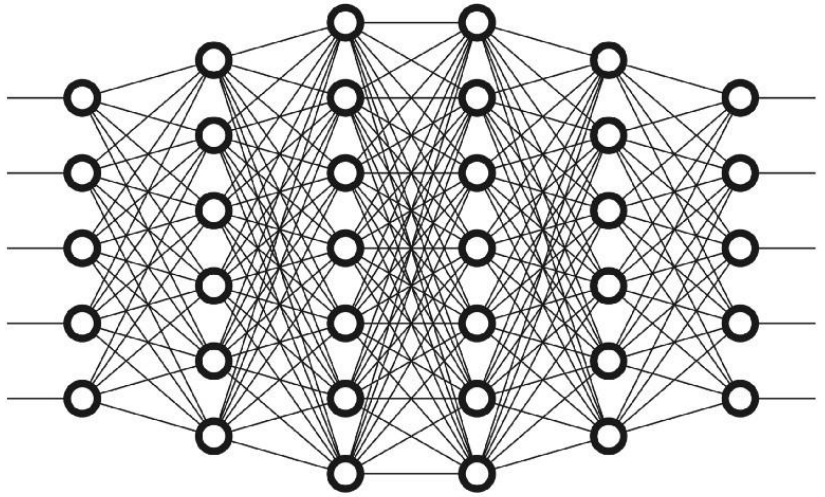


$$y = \gamma \cdot \frac{x - \mu}{\sigma} + \beta$$



Latency and Accuracy Trade-offs

- Anytime perception for LiDAR object detection DNNs is needed.
- But the execution of DNNs is rigid.

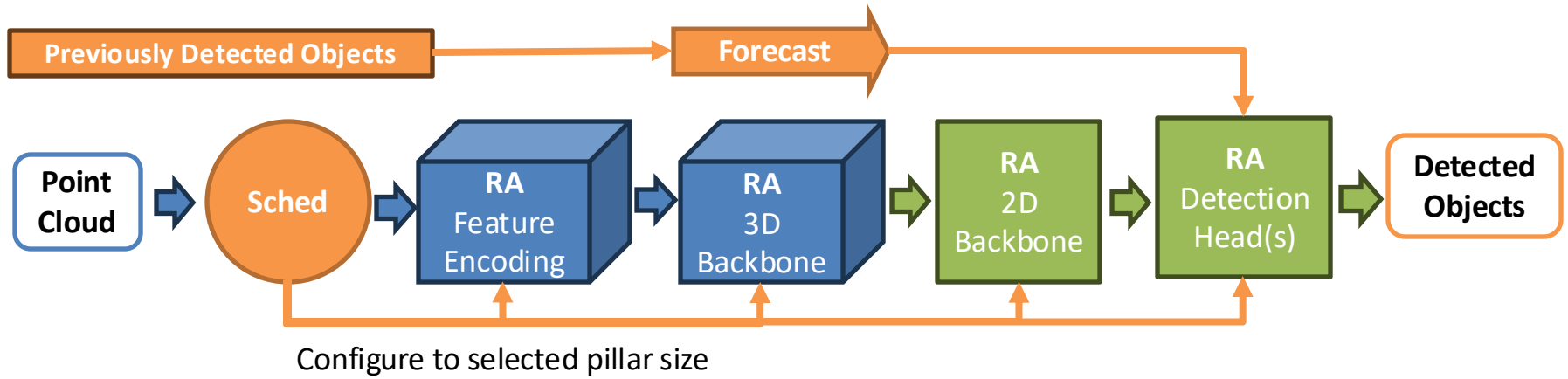


KU THE UNIVERSITY OF KANSAS

Image credits: https://www.freecodecamp.org/news/want-to-know-how-deep-learning-works-heres-a-quick-guide-for-everyone-1aedeca88076/

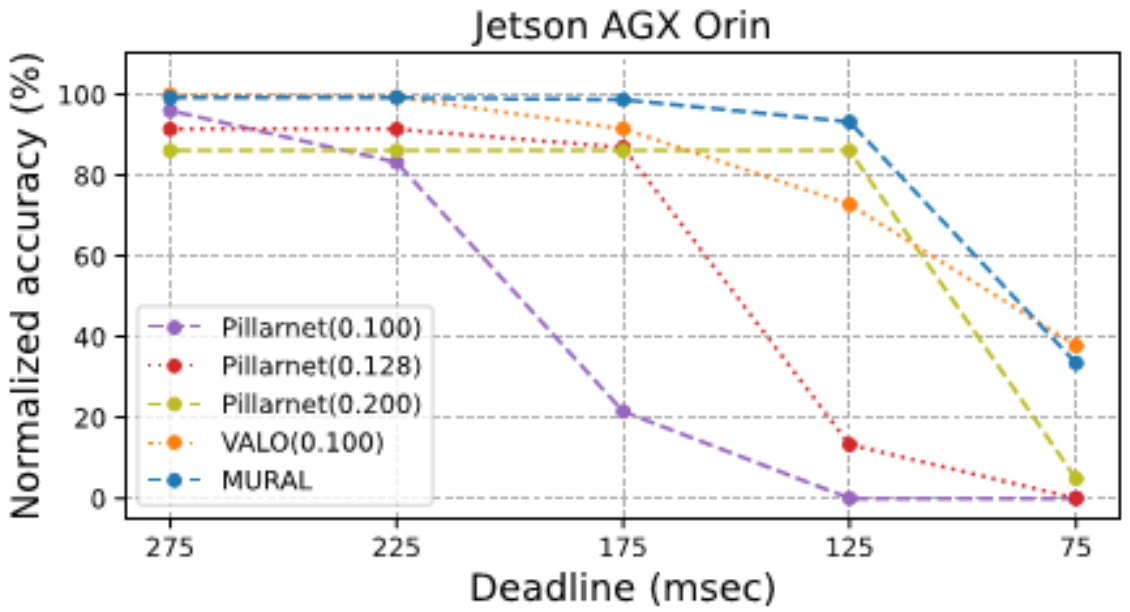
MURAL: Multi-resolution Anytime LiDAR

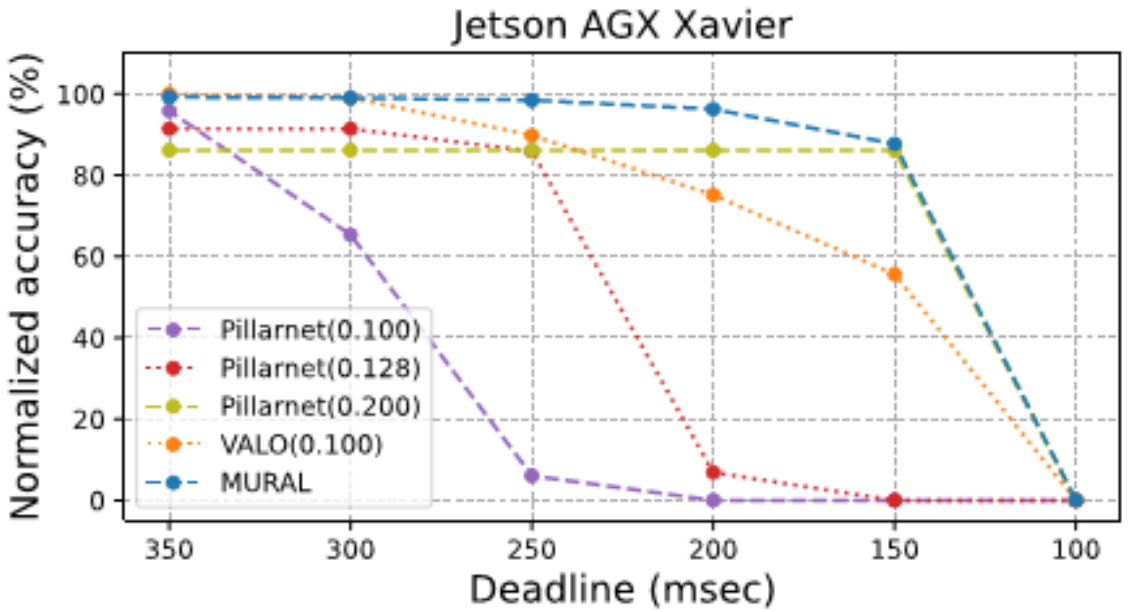
- MURAL enables anytime computing with ***dynamic pillar size scaling***
  - Resolution-aware (RA) batch normalization and training
  - Introduce additional pillar sizes after training
  - Deadline-aware scheduler
  - Forecasting and detection head optimization from our prior work*



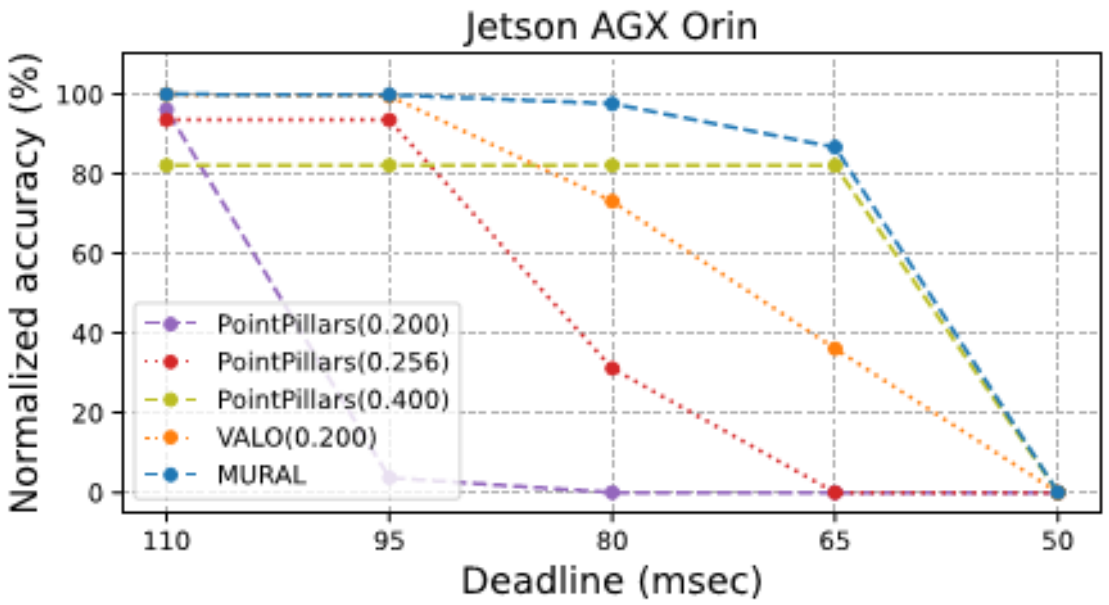
*Soyyigit et. al., VALO: A Versatile Anytime Framework for LiDAR based Object Detection Deep Neural Network, EMSOFT'24

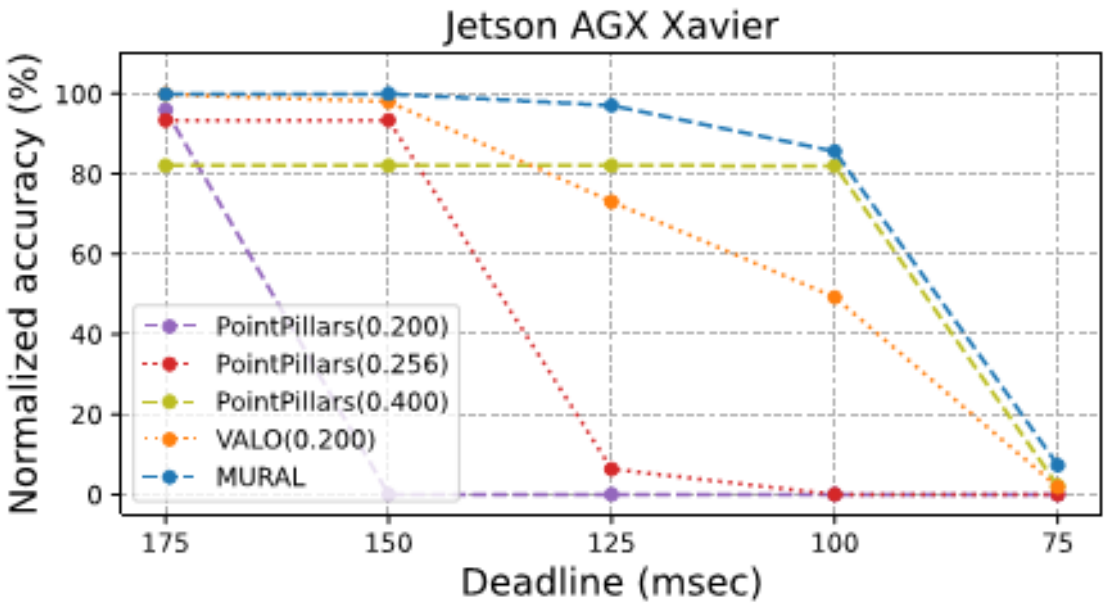
Pillarnet Results





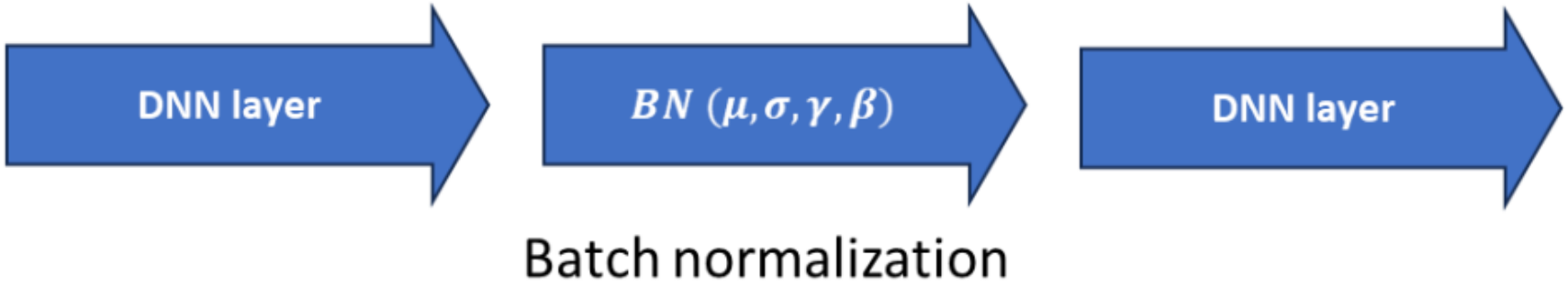
PointPillars Results





Batch Normalization (BN)

- Baseline BN learns statistics of its input.
- Statistics are dependent on input resolution.
- BN can only adapt to single input resolution.



One Naive Approach

- Dynamically changing the pillar size for a single high accuracy model.

- Results for Pillarnet:

Used for training

Pillar size (m^2)	Normalized mAP (%)
0.100^2	100.0
0.128^2	78.8
0.160^2	41.0
0.200^2	18.0

Accuracy plummets…

Conclusion

- We explored **dynamic input resolution** for LiDAR object detection DNNs.
- Evaluated on Pillarnet and PointPillars using Jetson AGX Xavier and Jetson AGX Orin.
- Results established MURAL as the *state-of-the-art on deadline-aware anytime LiDAR object detection*.
- Code is available at: https://github.com/CSL-KU/MURAL

KU THE UNIVERSITY OF KANSAS