# 18-742:
# Computer Architecture & Systems

# Tartan: Microarchitecting a Robotic Processor

Prof. Phillip Gibbons

Spring 2026, Lecture 20

# "Agents of Autonomy: A Systematic Study of Robotics on Modern Hardware"

**Mohammad Bakhshalipour, Phillip Gibbons 2023**

- **Architecture/Robotics gap: Only 3/150 papers in ISCA'22/Micro'22**
  - Quality benchmark suite needed to jump-start efforts

**Table 1.** Feature comparison of related work and RoWild. More ✔ is better.

| Paper/Repository | Scope | End-to-End | High-Perf. | Simulator-Friendly | Multi-Platform | Versatile & Modular | Open-Source | System. Study |
|---|---|---|---|---|---|---|---|---|
| Lin et al. [143] Yu et al. [208] | Self-Driving Cars | ✔ | ✔✔ | Unknown | ✔✔ | Unknown | ✘ | ✔ |
| MAVBench [74] | Drones | ✔ | ✔ | ✣ | ✔ | ✘ | ✔ | ✔ |
| One-off [55, 80, 118] | Single Task | ✘ | ✣ | ✣ | ✘ | ✔ | ✔ | ✘ |
| ROS [51] | Broad | ✘ | ✣ | ✘ | ✘ | ✔ | ✔ | ✘ |
| Educational [14, 47] | Broad | ✘ | ✘ | ✣ | ✘ | ✘ | ✔ | ✘ |
| RTRBench [68] | Broad | ✘ | ✔ | ✔ | ✘ | ✔ | ✔ | ✘ |
| *RoWild* | Broad | ✔ | ✔✔ | ✔ | ✔✔ | ✔ | ✔ | ✔✔ |

✣ Depending on the case (e.g., kernel, simulator), it can be ✘ or ✔. See Appendix §E.

# RoWild Workloads

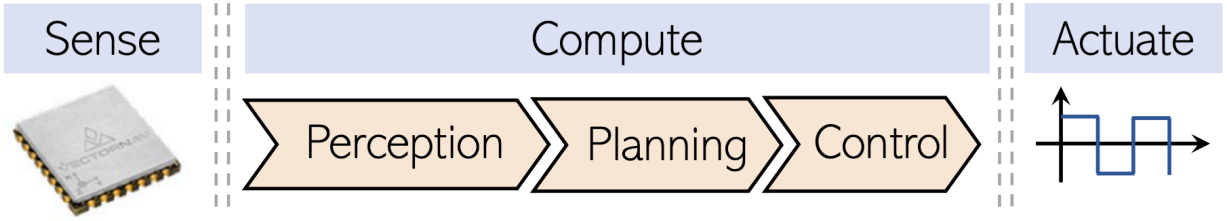
| Stage | Task | Algorithm(s) in RoWild |
|---|---|---|
| Perception | State Estimation | MCL[†‡ϒ¶ησ] [210], AMCL[ζ] [195] |
| | Localization and Mapping | EKF[ϒ§τφΛ] [197], Fast[‡¶ζ] [158], Graph[ϑφ] [187], ORB[ςϱ] [82] |
| | Scene Reconstruction | Point-Based Fusion[†‡ϒκυ] [205] |
| | Grid Generation | Probabilistic Occupancy Map[†κϑ] [104] |
| | Object Detection | MobileNet-SSD[‡ϒϱχΛ] [207] |
| Planning | $(x, y)$ Search | WA⋆[†¶ζηαΠΛ] [176], GA⋆ [215], RA⋆ [69], IDA⋆[ς] [129], DQN [157] |
| | $(x, y, z)$ Search | WA⋆[†¶ζηαΠΛ] [176], GA⋆ [215], RA⋆ [69], IDA⋆[ς] [129], DQN [157] |
| | $(x, y, \theta)$ Search | WA⋆[†¶ζηαΠΛ] [176], GA⋆ [215], RA⋆ [69], IDA⋆[ς] [129] |
| | Timed Search | WA⋆[†¶ζηαΠΛ] [176] + Backward Dijkstra [71] |
| | 2D $n$-DoF Search | RRT[†‡ϒκ] [103], RRT⋆[ϒωγδ] [105], PRM[ϒϑψ] [191], Shortcut [107] |
| | 3D $n$-DoF Search | RRT[†‡ϒκ] [103], RRT⋆[ϒωγδ] [105], PRM[ϒϑψ] [191], Shortcut [107] |
| | Reactive Planning | Behavior Tree[†λϑφ] [108] |
| | Task Scheduling | Symbolic Planning[§ϑδ] [72] |
| Ctrl. | Motion Control | MPC[ϒΠϑ†καψε] [100] PID[ϒϑζθφ] [203], PP[‡ϒκφ] [170] |
| | Parameter Learning | DMP[ϒϑξαθι] [131], CEM[§λ] [173], BO[ϱ] [119] |

**Real-World Robots:**

[†]Spot [10]
[‡]DJI Phantom [41]
[ϒ]LoCoBot [2]
[§]Atlas [9]
[¶]AMR [48]
[Π]AscTec Pelican [7]
[ζ]Roomba 980 [49]
[υ]Roomba i7+ [50]
[η]Husky [21]
[ϑ]YuMi [65]
[κ]Jackal [28]
[ξ]LBR [33]
[ϱ]Pepper [53]
[σ]TurtleBot [60]
[ς]TurtleBot3 [61]
[τ]MiR [34]
[φ]AG [31]
[Λ]Pioneer 3-DX [43]
[φ]UR10e [57]
[χ]TALOS [54]
[ψ]KR60-3 [32]
[ω]Grizzly [13]
[γ]Skydio [52]
[δ]M-200iA/2300 [18]
[ε]PerceptIn [207]
[α]Boxbot [11]
[θ]UR5e [58]
[ι]Franka Panda [19]
[λ]PAL TIAGo [59]

**Table 2.** RoWild's workloads. Algorithms in **color** are characterized in this paper in the context of end-to-end robotic applications.

# Robotic Software Pipeline




# End-to-End Applications

| Name | Mission | Resembling | Environment |
|---|---|---|---|
| DeliBot | Delivery | Spot | Our Campus |
| PatrolBot | Patrolling | Pioneer 3-DX | Our Campus |
| MoveBot | Manipulation | LoCoBot | Synthetic |
| HomeBot | Cleaning | Roomba i7+ | Hypersim |
| FlyBot | Photography | AscTec Pelican | FR Campus |
| CarriBot | Transportation | Boxbot | Intel Lab |

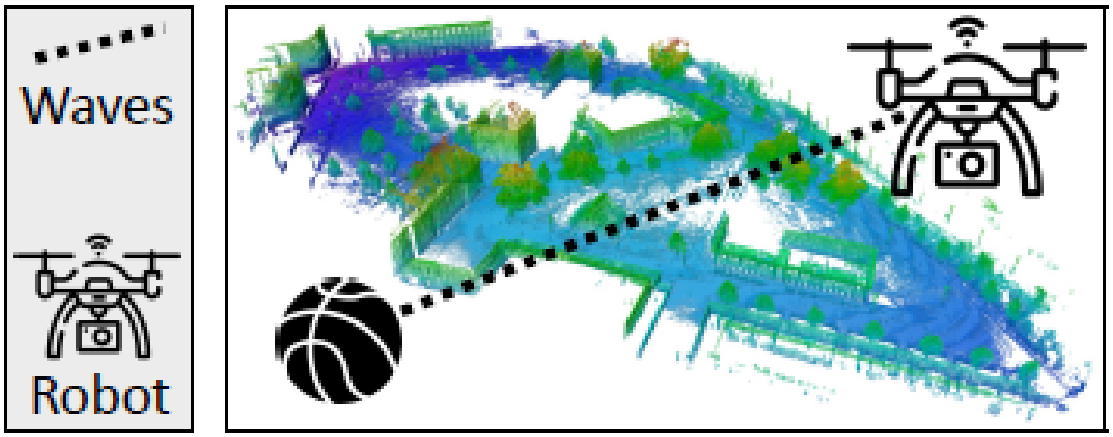
# Evaluated Compute Platforms

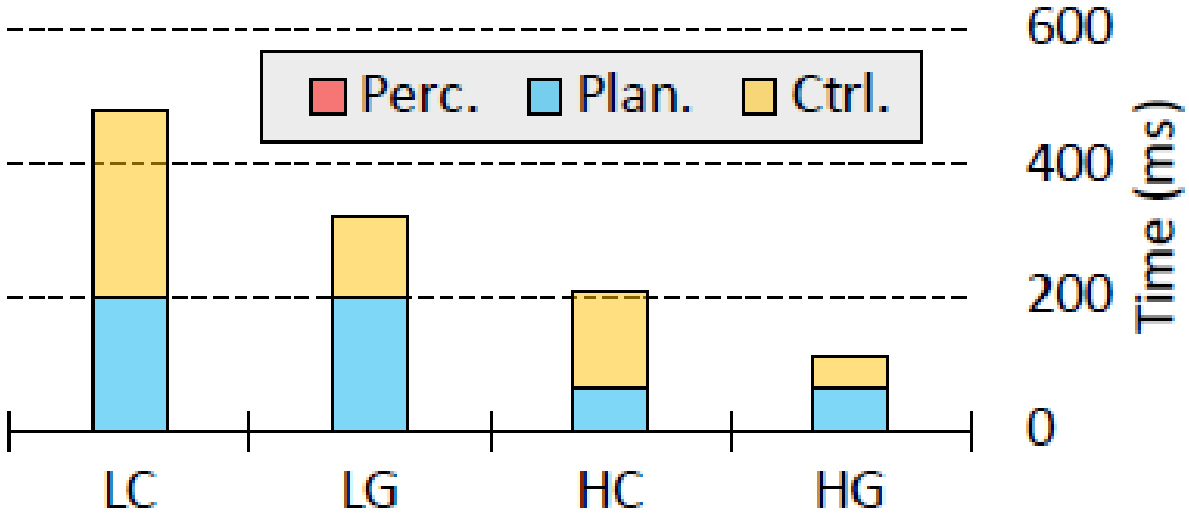
**Table 3.** The evaluated compute platforms. The prices are as of August 9, 2023.

| Acronym | Platform | Cores | Freq. (GHz) | TDP (W) | Memory (GB) | Price (US $) |
|---|---|---|---|---|---|---|
| ***LC*** | ARM Cortex A57 CPU | 4 | 1.43 | 10 | 4 | 149 |
| ***LG*** | Nvidia Maxwell GPU | 128 | 0.92 | 10 | 4 | 149 |
| ***HC*** | Intel Xeon Gold CPU | 20 (×2) | 2.10 | 125 | 384 | 1493 |
| ***HG*** | Nvidia Titan X GPU | 3584 | 1.41 | 250 | 12 | 999 |

Also: Avnet Ultra96-V2 FPGA (Verilog implementations)

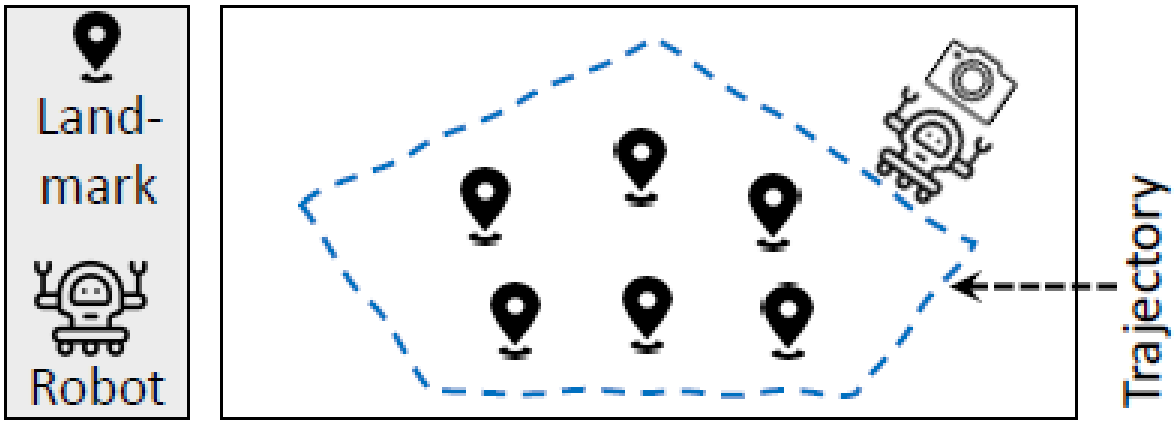
# FlyBot: Drone Performing Aerial Photography

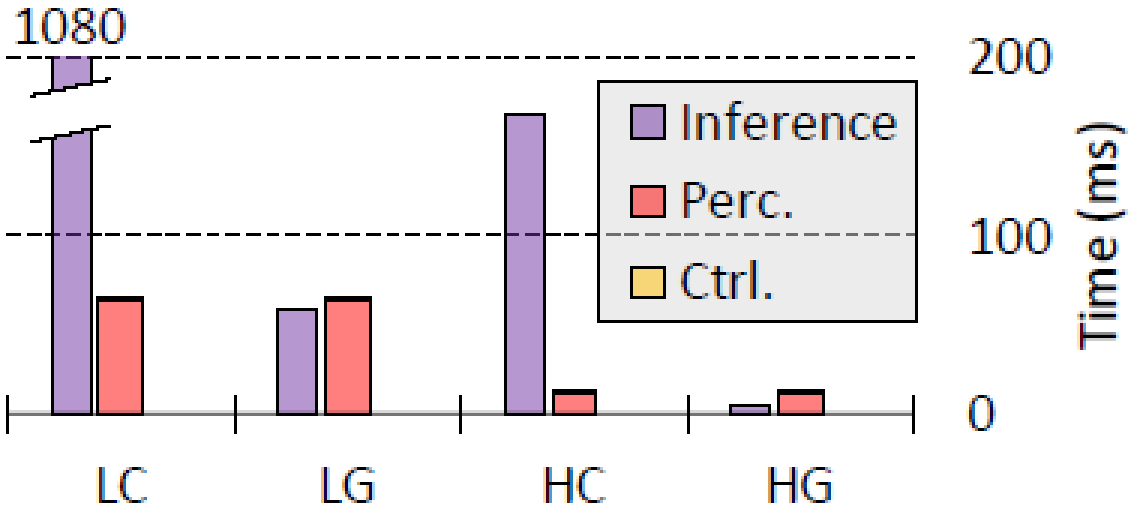



**Planning & Control dominate. Control benefits from GPU.**
**Planning requires high single-threaded performance**

# PatrolBot: Wheeled Robot Patrolling

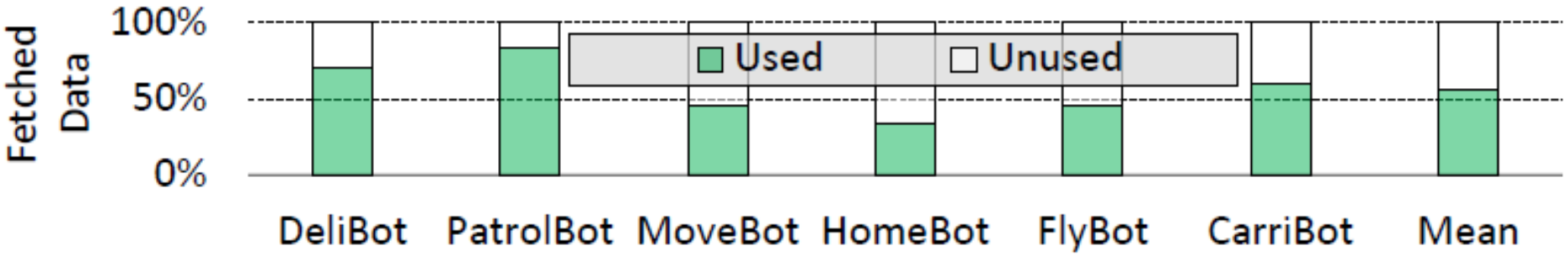



**Inference (object classification) is bottleneck for CPU-only platforms. LG ($149) outperforms HC ($1493).**

# Challenge: Poor Spatial Locality

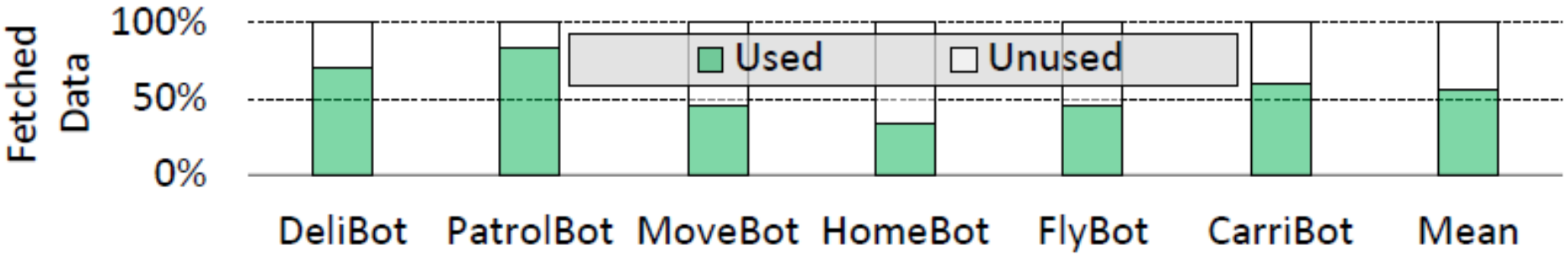
Infinite Sized Cache, No HW prefetching, 64B cache lines



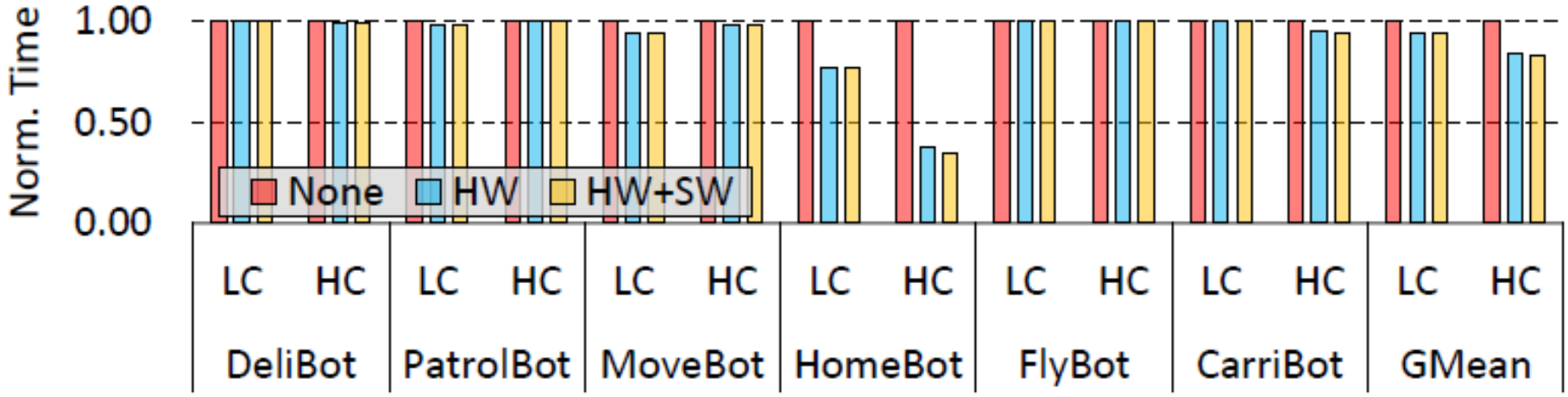

**44% of fetched data never used.** Existing prefetchers don't help much.

# Challenge: Poor Spatial Locality

Infinite Sized Cache, No HW prefetching, 64B cache lines




# Challenge: Prefetchers Inadequate




**44% of fetched data never used. Existing prefetchers don't help much.**

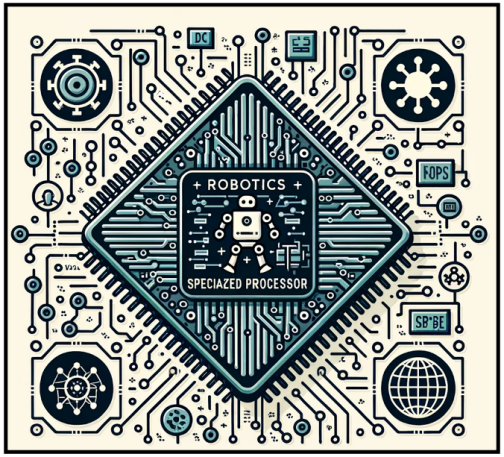
# “Tartan: Microarchitecting a Robotic Processor”

## Mohammad Bakhshalipour, Phillip Gibbons 2024

- **Mohammad: Sharif U of Tech MS, now Nvidia**
  - CMU PhD. 2025 SIGMICRO best dissertation

# Executive Summary

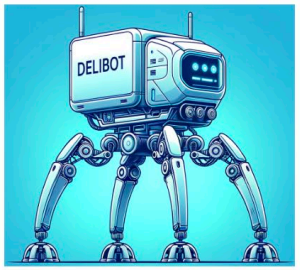
- **Architecting a domain-specific processor for robotics**
- **Extensive profiling of robotics workloads, finding main execution bottlenecks**
- **Architectural enhancements to address the bottlenecks in robotics workloads**
  - Oriented vectorization
  - Approximate acceleration
  - Robot-semantic prefetching
  - Intra-application cache partitioning



# Methodology

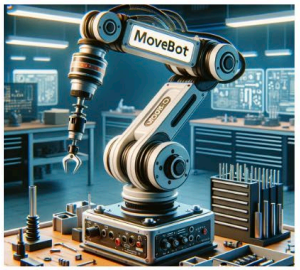
- **Simulation framework**
  - ZSim [Sanchez+, ISCA'13]
- **Baseline processor: Intel Core i7-10610U**
  - 4 cores, 8MB L3 cache
  - Integrated in NASA's Valkyrie
- **Workloads**

DeliBot




PatrolBot


MoveBot

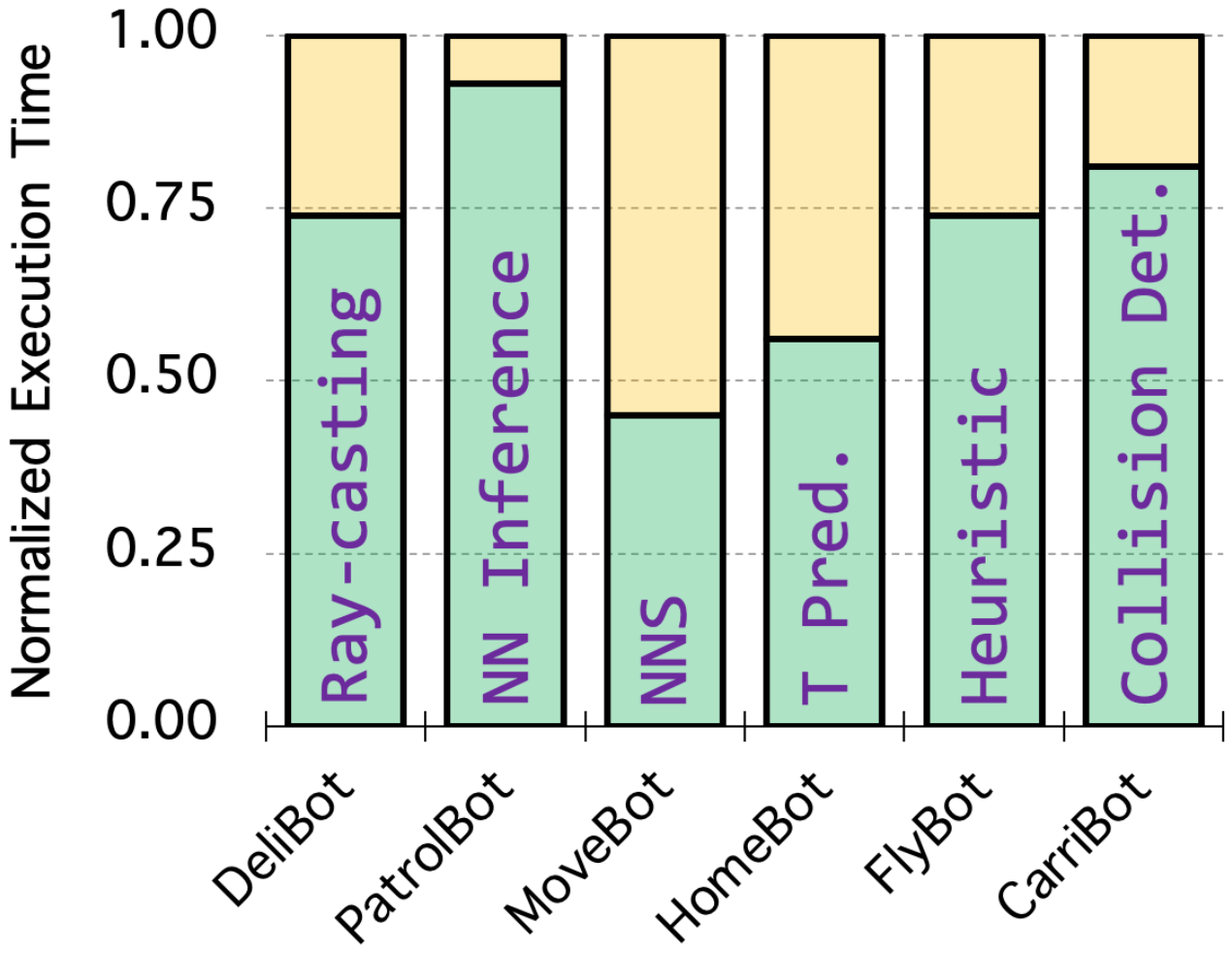



HomeBot


FlyBot

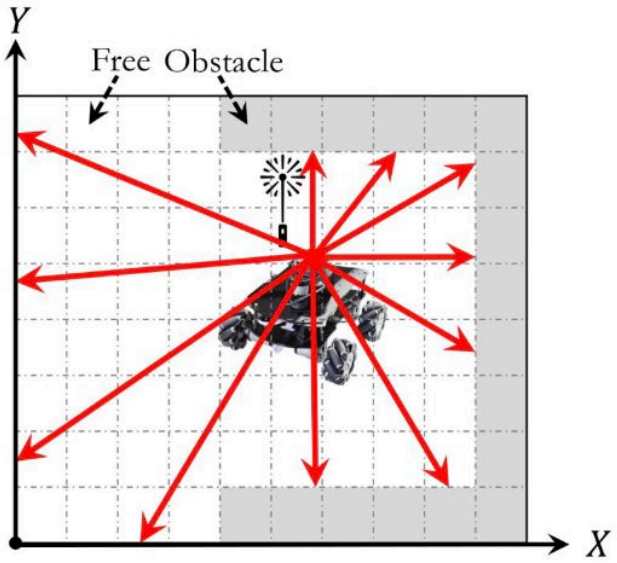

CarriBot

# Identifying Performance Bottlenecks

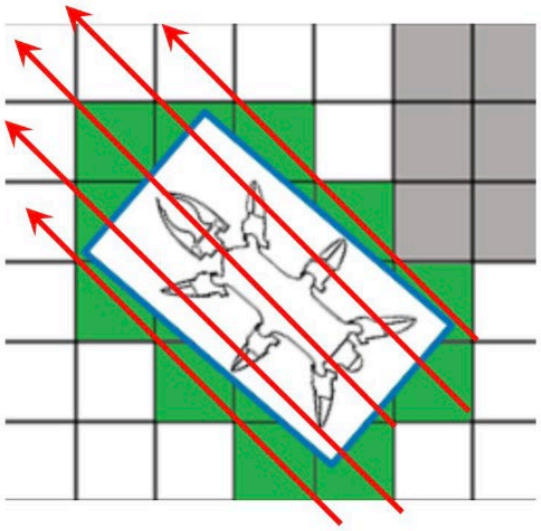


# Oriented Vectorization

- Tartan extends CPUs' vectorization unit to handle slanted-line address patterns




Ray Casting



Collision Detection

# Oriented Vectorization

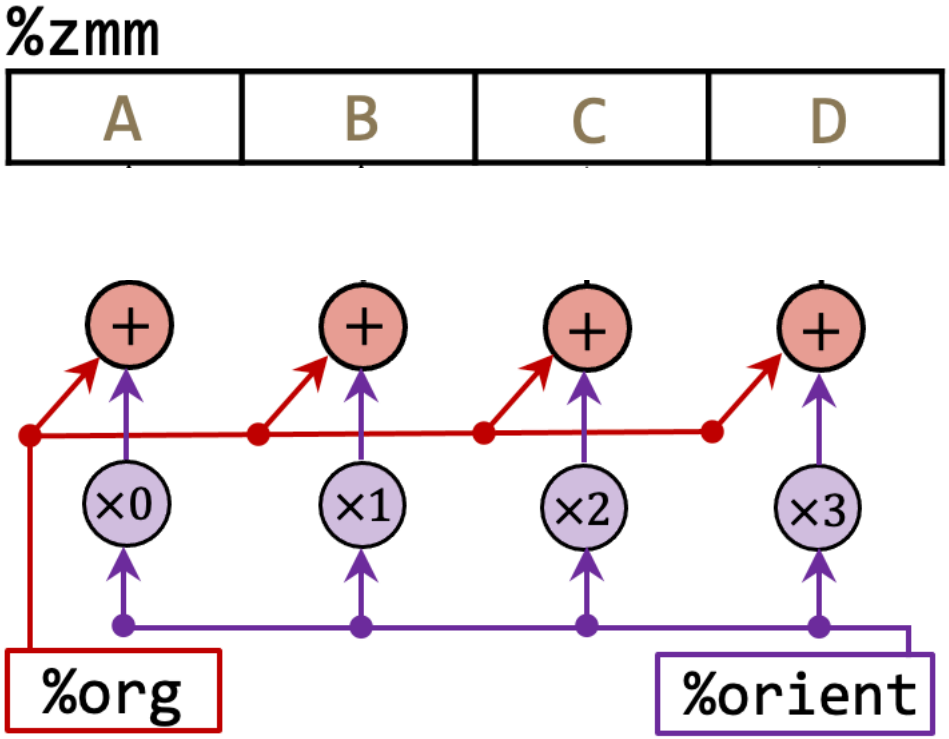
- **Traditional vector instruction**

```
MOVE %zmm, (%org)
```

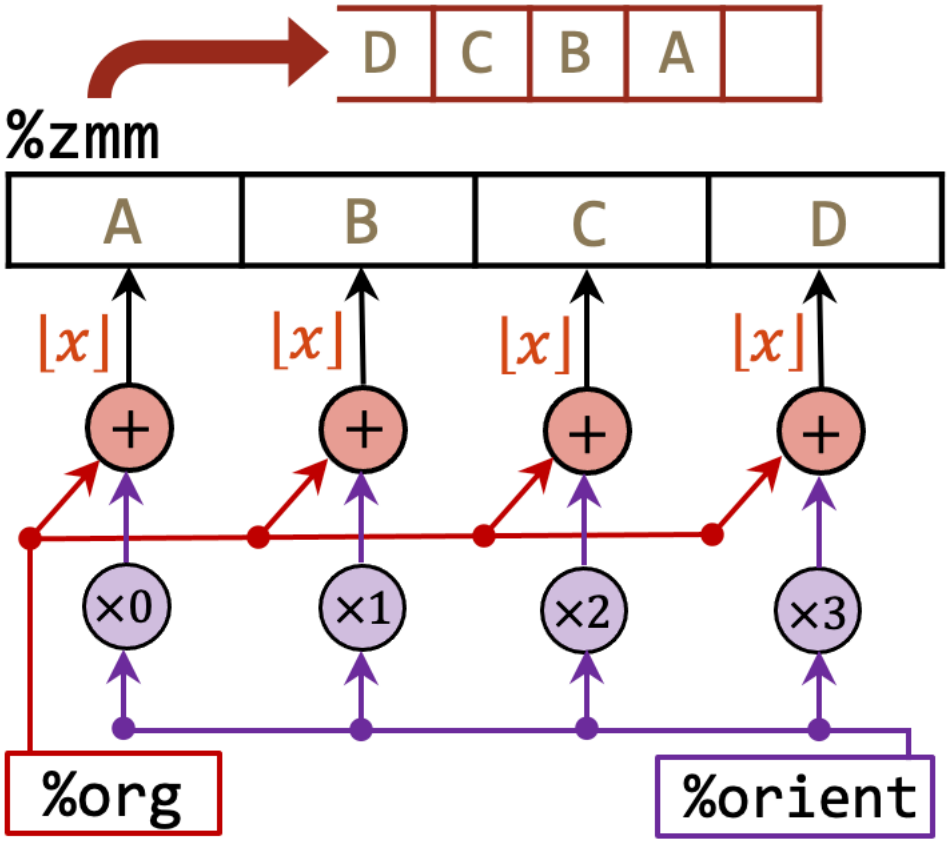
- **New instruction**

```
O_MOVE %zmm, (%org), %orient
```

# Oriented Vector Address Generation

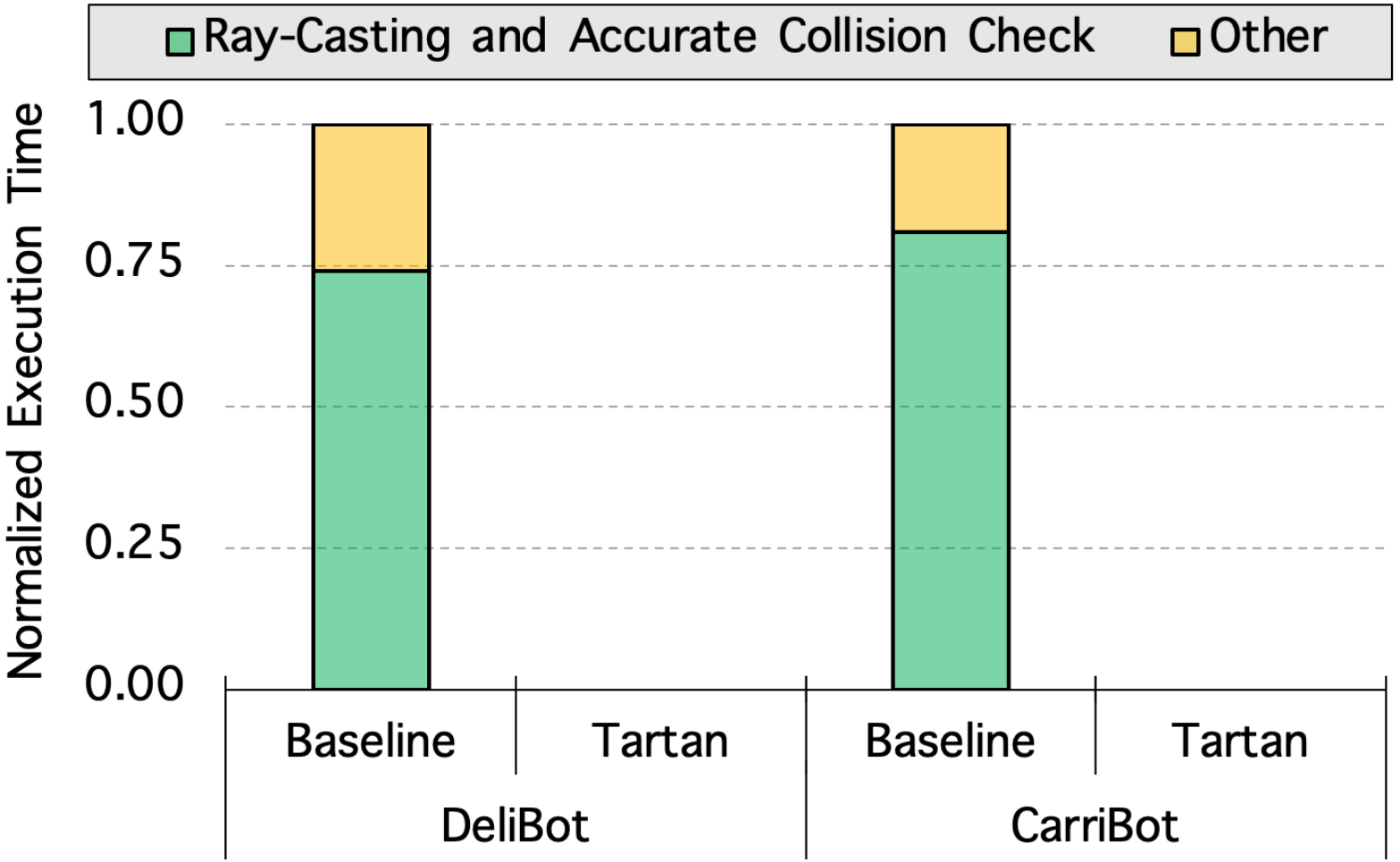


# Oriented Vector Address Generation



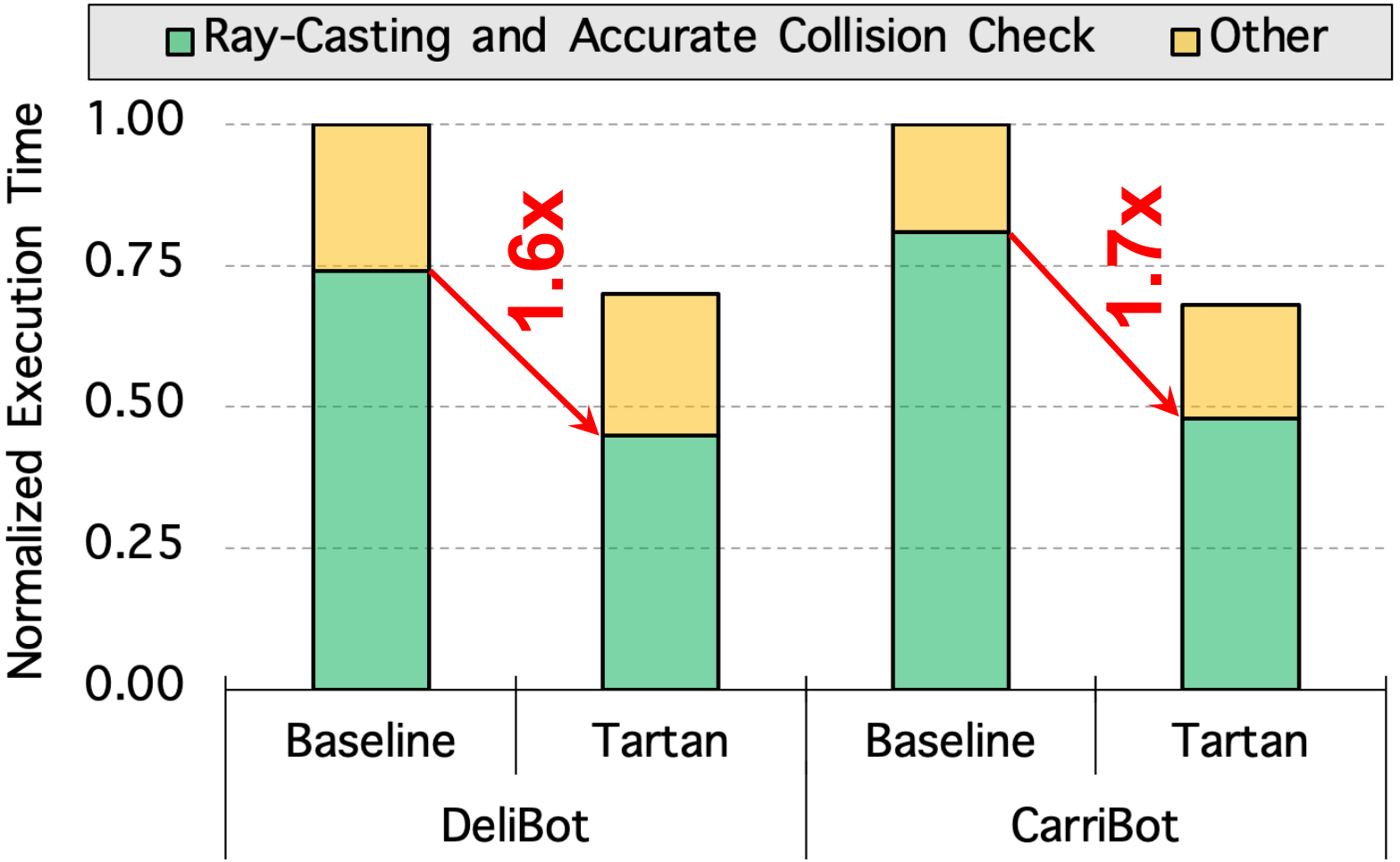
# Results: Oriented Vectorization

- DeliBot (Ray Casting)
- CarriBot (Collision Detection)



# Results: Oriented Vectorization

- DeliBot (Ray Casting)
- CarriBot (Collision Detection)



# Discussion: Summary Question #1

## What Did the Paper Get Right?

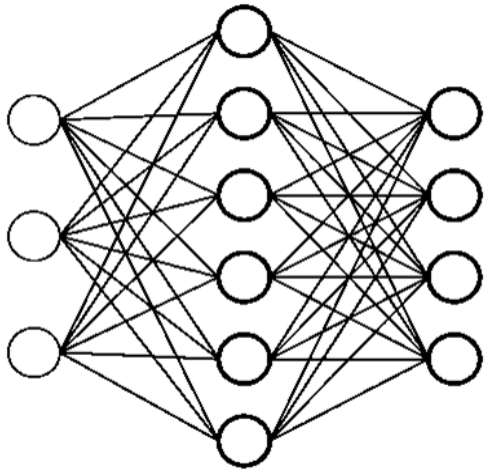
**State the 3 most important things the paper says.**

These could be some combination of the motivations, observations, interesting parts of the design, or clever parts of the implementation.

# Approximate Acceleration

- **Many robotic applications allow for approximation**
  - Application-level: Cleaning
  - Algorithm-level: Pathfinding heuristic cost calculation*

- **Tartan replaces approximable, costly functions by a neural network**
  - Trained offline
  - Executed by an NPU

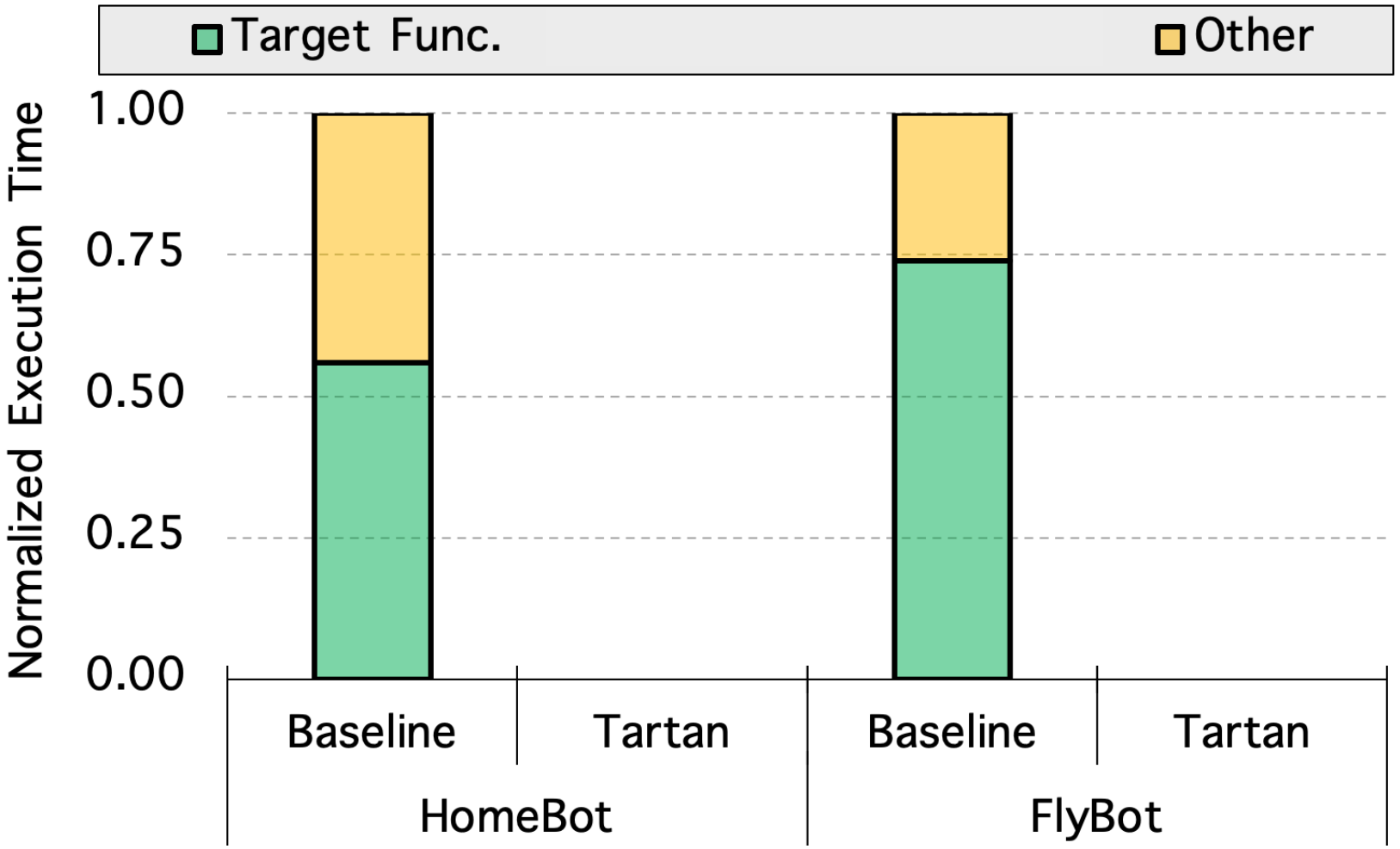


***Approximate Execution Accurate Results (AXAR)**

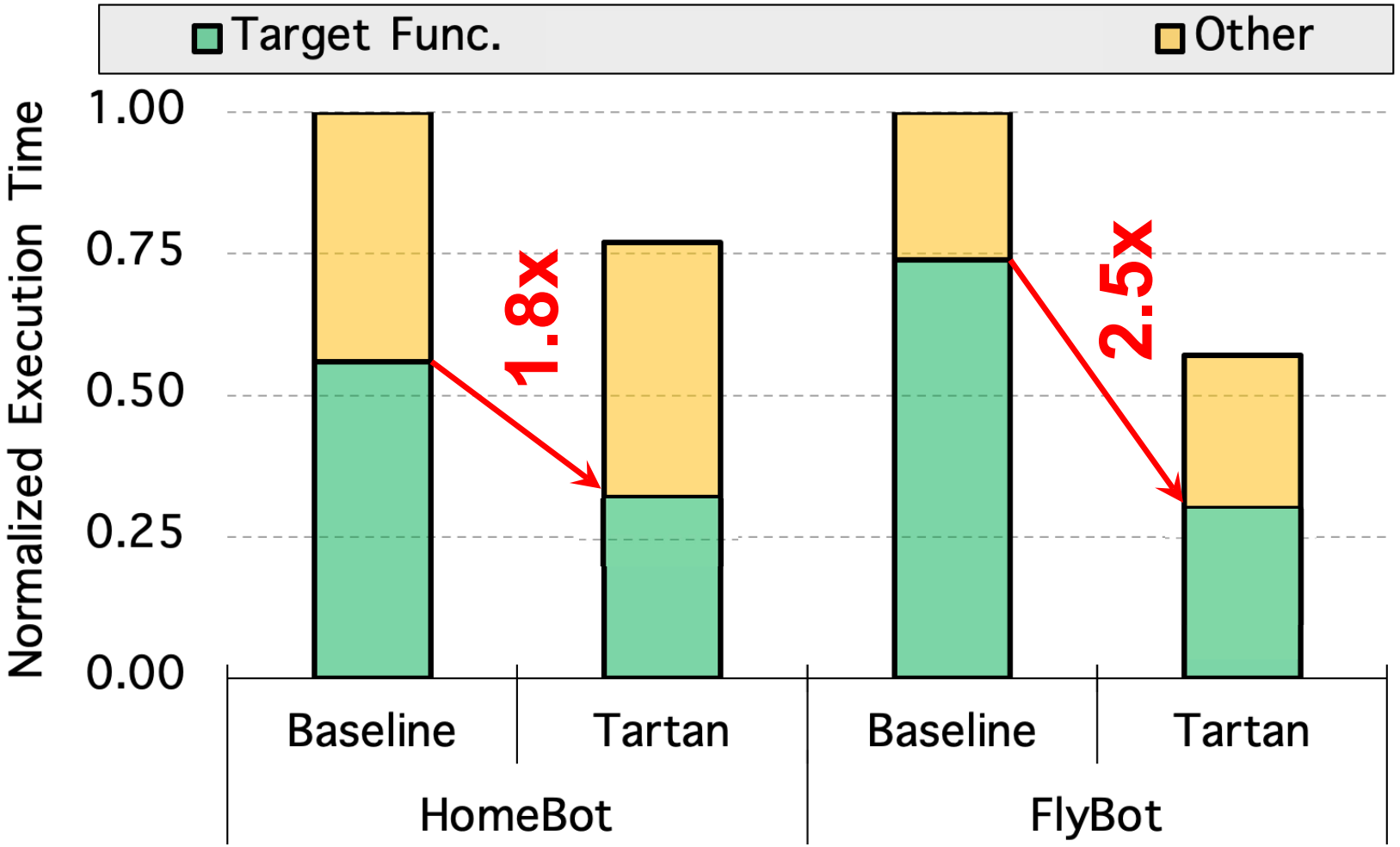
Heuristic must never overestimate the true cost

(We use a software technique based on ATA*)

# Results: Approximate Acceleration



# Results: Approximate Acceleration



# Discussion: Summary Question #2

## What Did the Paper Get Wrong?

**Describe the paper's single most glaring deficiency.**

Every paper has some fault. Perhaps an experiment was poorly designed or the main idea had a narrow scope or applicability.

# End-to-End Improvements

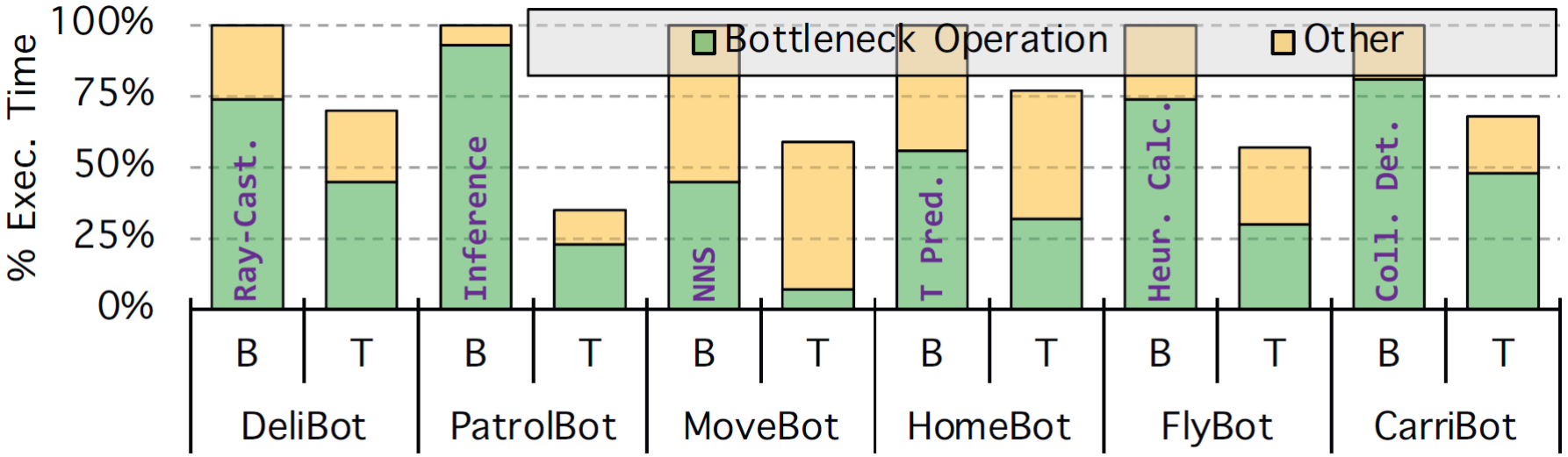
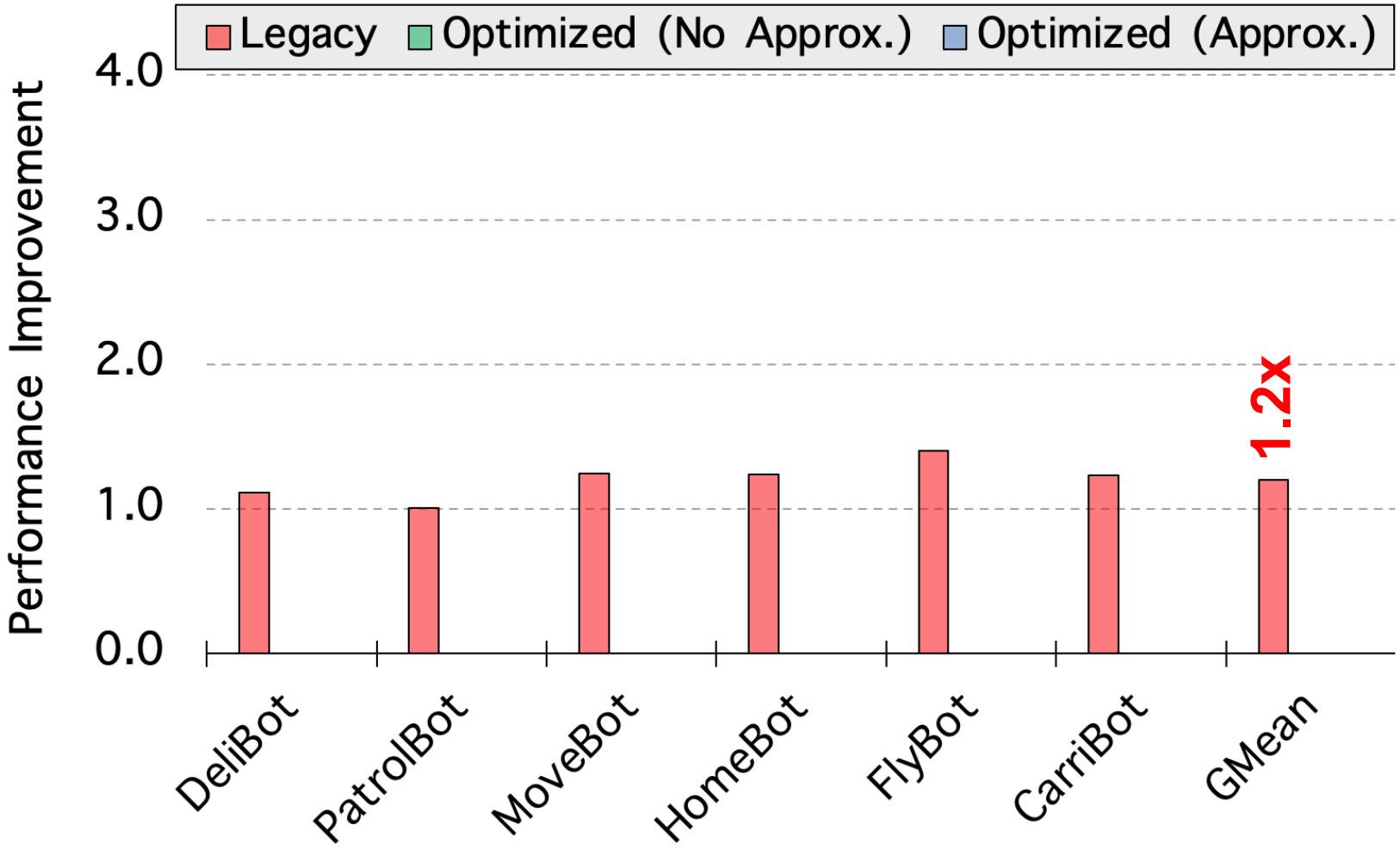
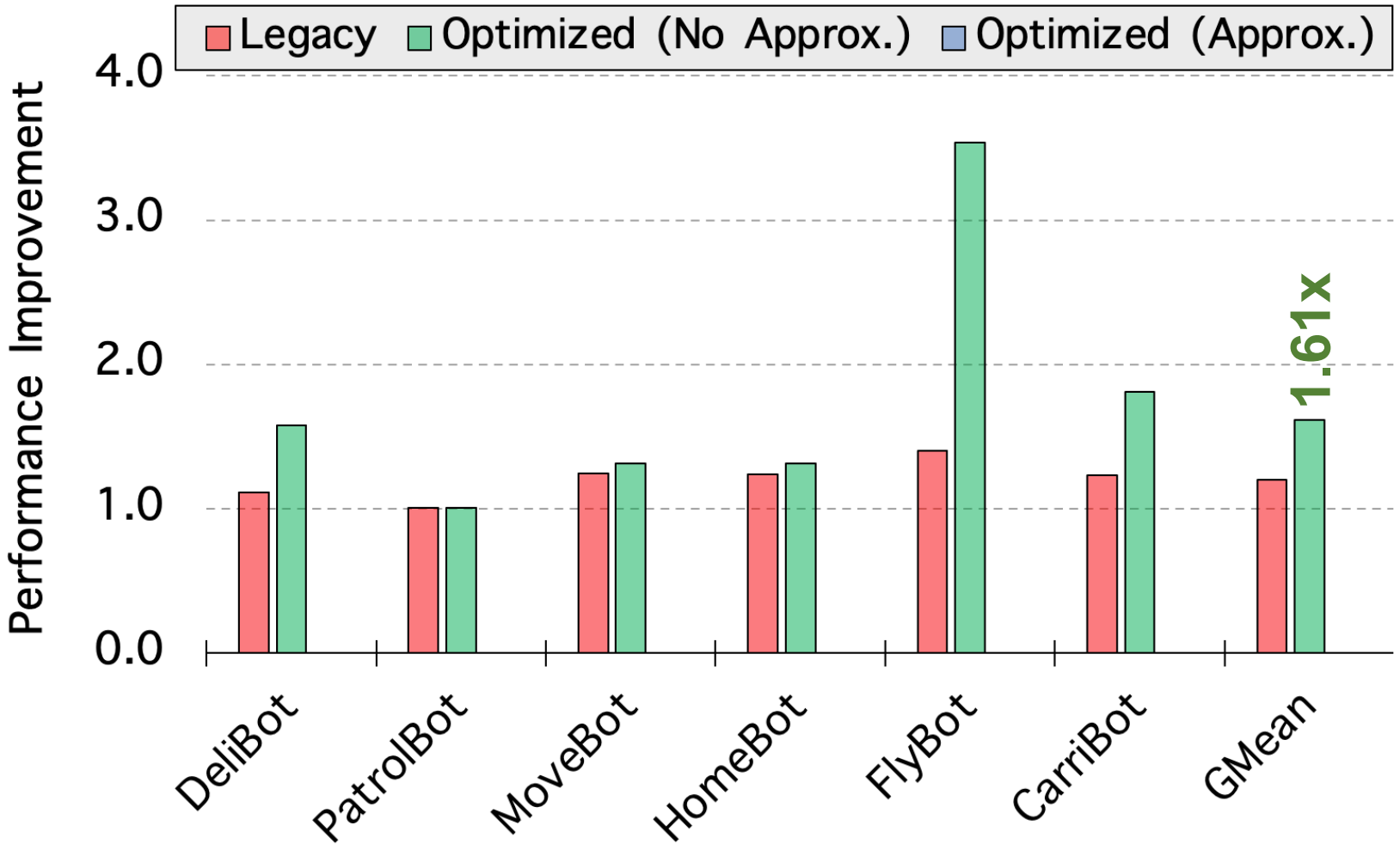



**Fig. 1:** Execution time breakdown and bottleneck analysis.

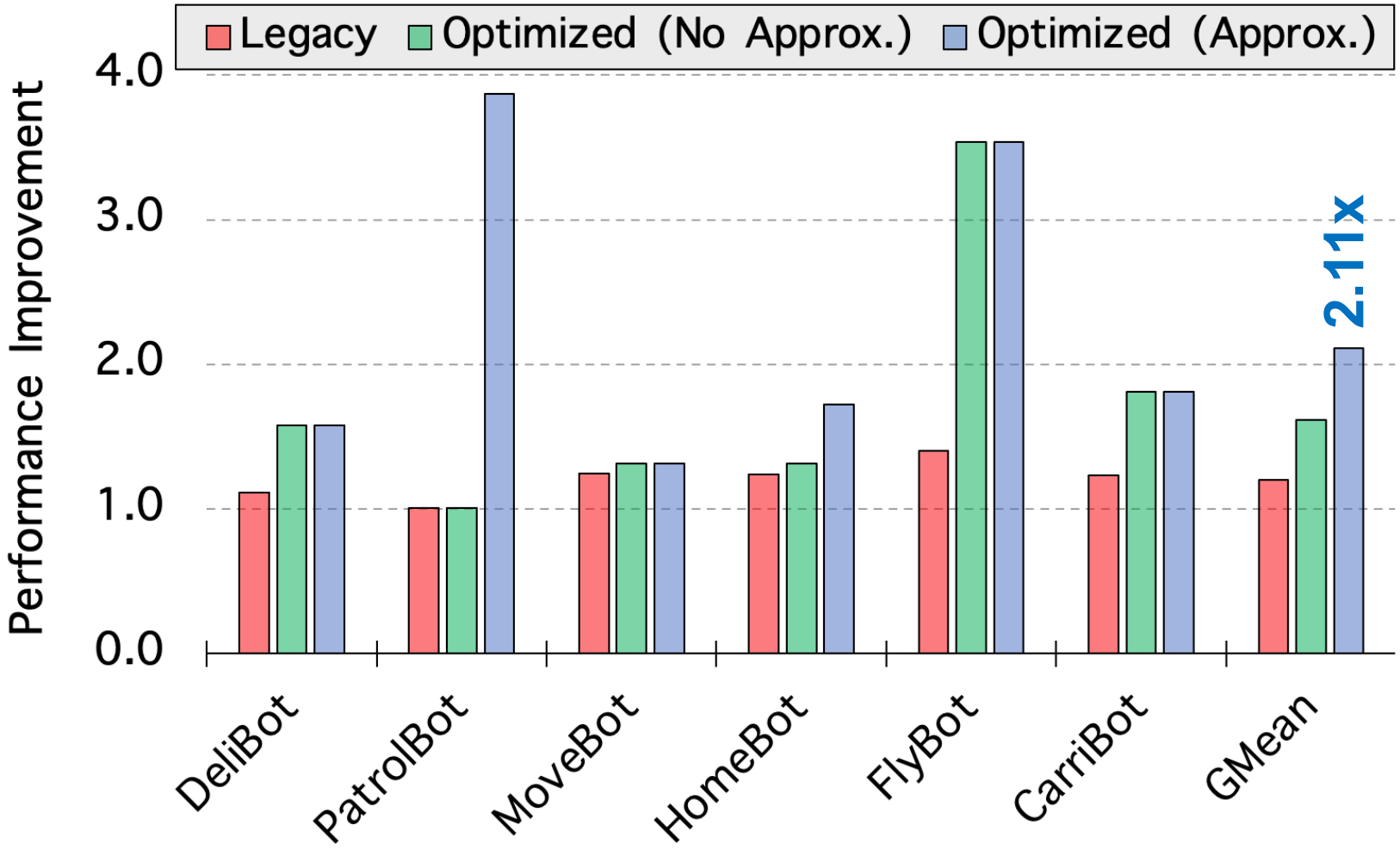
# Improvements vs. Software Changes



# Improvements vs. Software Changes



# Improvements vs. Software Changes



# To Read for Wednesday

**"In-Datacenter Performance Analysis of a Tensor Processing Unit"**

Norman Jouppi, Cliff Young, Nishant Patil, David Patterson, et al. 2017

**Optional Further Reading:**

**"Ten Lessons From Three Generations Shaped Google's TPUv4i"**

Norman Jouppi, Doe Hyun Yoon, Matthew Ashcraft, et al. 2021