XRBench: An Extended Reality (XR) Machine Learning Benchmark Suite for the Metaverse

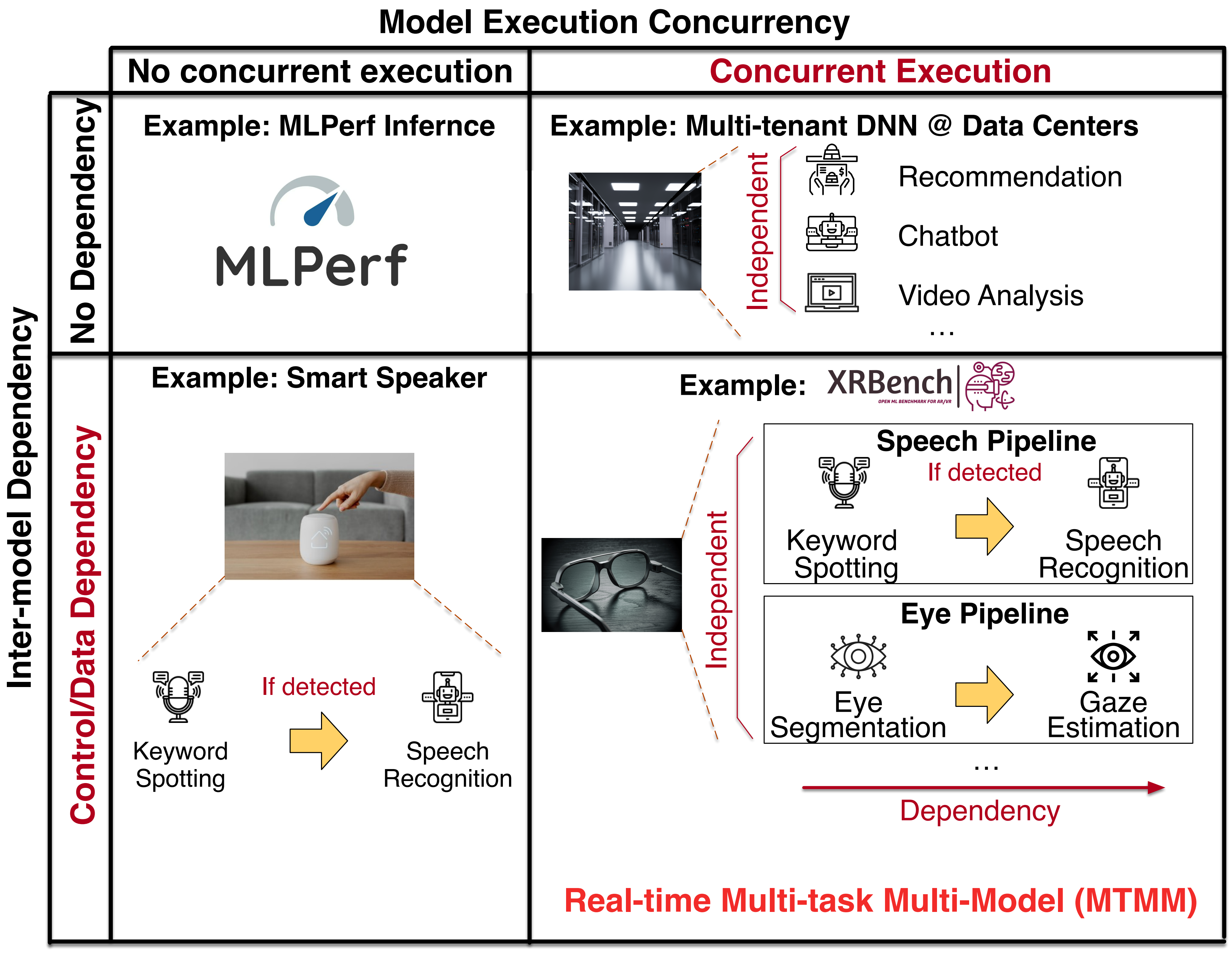
Home page: https://xrbench.ai
Github: https://github.com/XRBench

https://xrbench.ai

Hyoukjun kwon[1,2], Krishnakumar Nair[2], Jamin Seo[3], Jason Yik[4], Debabrata Mohapatra[2], Dongyuan Zhan[2], Jinook Song[2], Peter Capak[2], Peizhao Zhang[2], Peter Vajda[2], Colby Banbury[4], Mark Mazumder[4], Liangzhen Lai[2], Ashish Sirasao[2], Tushar Krishna[3], Harshit Khaitan[2], Vikas Chandra[2], Vijay Janapa Reddi[4]

1 UCI 2 Meta 3 Georgia Tech 4

ML Workload Taxonomy



Characteristics of Real-time MTMM ML Workloads

Characteristic	Example Implications to ML System Design
Concurrent and Cascaded Models	Concurrency and dependency-aware scheduling
Realtime Processing	Deadline-oriented optimizations (Just minimizing latency beyond the deadline does not contribute to better user experiences)
SoC Level Pipeline	Input sensor- and output device-aware scheduling
Multi-Modal Inputs and Models	Model heterogeneity-aware hardware design
User Input-driven Dynamism	Runtime software to support for user input-driven dynamic workloads
Context-driven Workloads	Highly diverse workload depending on

To understand this new class of ML workloads, real-time MTMM, we need a well-defined workload suites based on realistic use cases!

XRBench v0.1

Unit Models

Category	Task	Model	Dataset	Model Perf. Requirement
Interaction	Hand Tracking (HT)	Hand Shape/Pose (Ge et al., 2019)	Stereo Hand Pose (Zhang et al., 2017)	AUC PCK, GT 0.948
	Eye Segmentation (ES)	RITNet (Chaudhary et al., 2019)	OpenEDS 2019 (Garbin et al., 2019)	mIoU, GT 90.54
	Gaze Estimation (GE)	Eyecod (You et al., 2022)	OpenEDS 2020 (Palmero et al., 2021)	Angular Error, LT 3.39
	Keyword Detection (KD)	Key-Res-15 (Tang & Lin, 2018)	Google Speech Cmd (Google, 2017)	Accuracy, GT 85.60
	Speech Recognition (SR)	Emformer (Shi et al., 2021)	LibriSpeech (Panayotov et al., 2015)	WER (others), LT 8.79
Context Understanding	Semantic Segmentation (SS)	HRViT (Gu et al., 2022)	Cityscape (Cordts et al., 2016)	mIoU, GT 77.54
	Object Detection (OD)	D2Go (Meta, 2022b)	COCO (Lin et al., 2014)	boxAP, GT 21.84
	Action Segmentation (AS)	TCN (Lea et al., 2017)	GTEA (Fathi et al., 2011)	Accuracy, GT 60.8
	Keyword Detection (KD)	Key-Res-15 (Tang & Lin, 2018)	Google Speech Cmd (Google, 2017)	Accuracy, GT 85.60
	Speech Recognition (SR)	Emformer (Shi et al., 2021)	LibriSpeech (Panayotov et al., 2015)	WER (others), LT 8.79
World Locking	Depth Estimation (DE)	MiDaS (Ranftl et al., 2020)	KITTI (Geiger et al., 2012)	$\delta > 1.25$,LT 22.9
	Depth Refinement (DR)	Sparse-to-Dense (Ma & Karaman, 2018)	KITTI (Geiger et al., 2012)	δ_1, GT 85.5(100 samples)
	Plane Detection (PD)	PlaneRCNN (Liu et al., 2019)	KITTI (Geiger et al., 2012)	$AP^{0.6m}$, GT 0.37

Selection Criteria

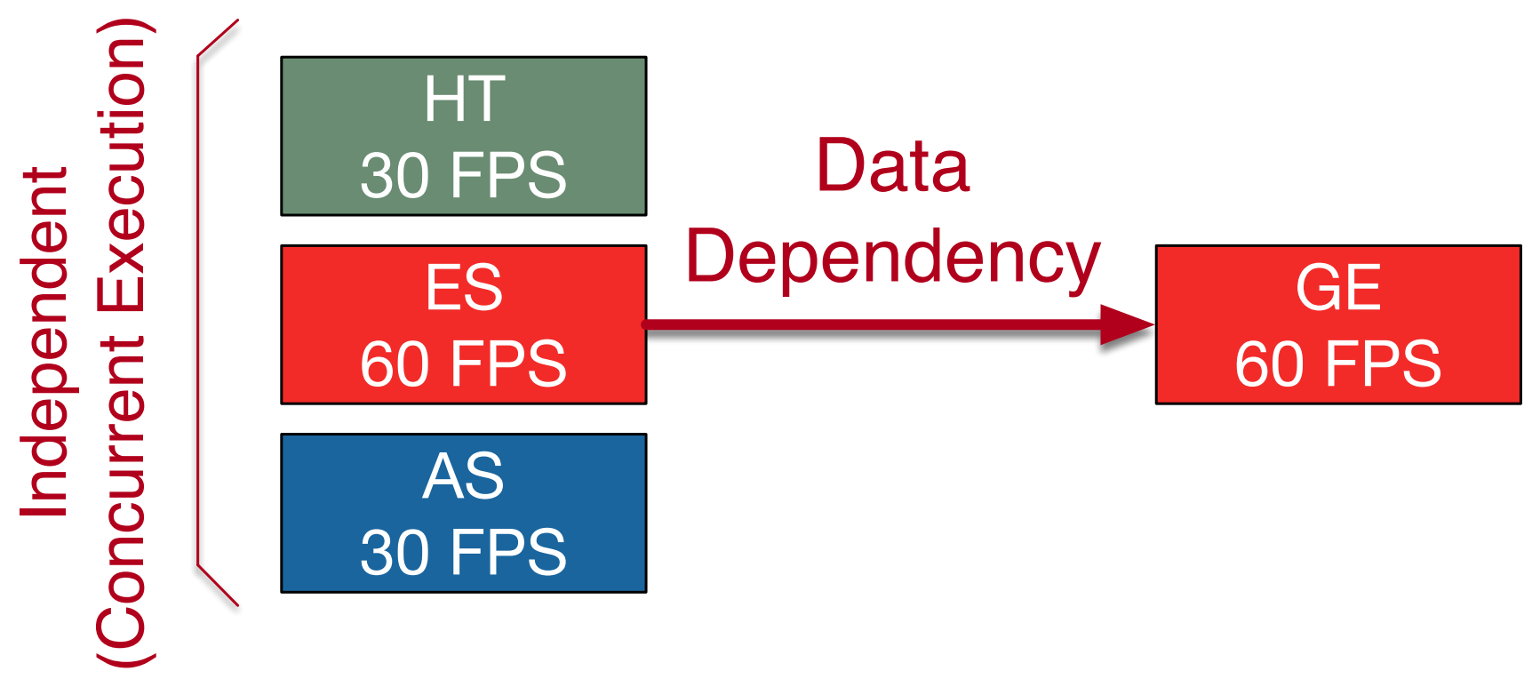
- Recommendation from industry ML researchers and engineers
- Model performance (e.g., accuracy) and efficiency

Envisioning On-device AI

Usage Scenarios

Usage Scenario	HT	Eye Pipeline		Speech Pipeline		SS	OS	AS	DE	DR	PD	Example Usage Scenario Description
		ES → GE (dep: D)		KD → SR (dep: C)								
Social Interaction A	30	60	60							30		AR messaging with AR object rendering
Social Interaction B		60	60					30				In-person interaction with AR glasses
Outdoor Activity A				3	3	10	30					Hiking with smart photo capture
Outdoor Activity B				3	3		30					Rest during hike
AR Assistant				3	3	10	10		30		30	Urban walk with informative AR objects
AR Gaming	45								30		30	Gaming with AR object
VR Gaming	45	60	60									Highly-interactive Immersive VR gaming

Example Usage Scenario Diagram (Social Interaction B)



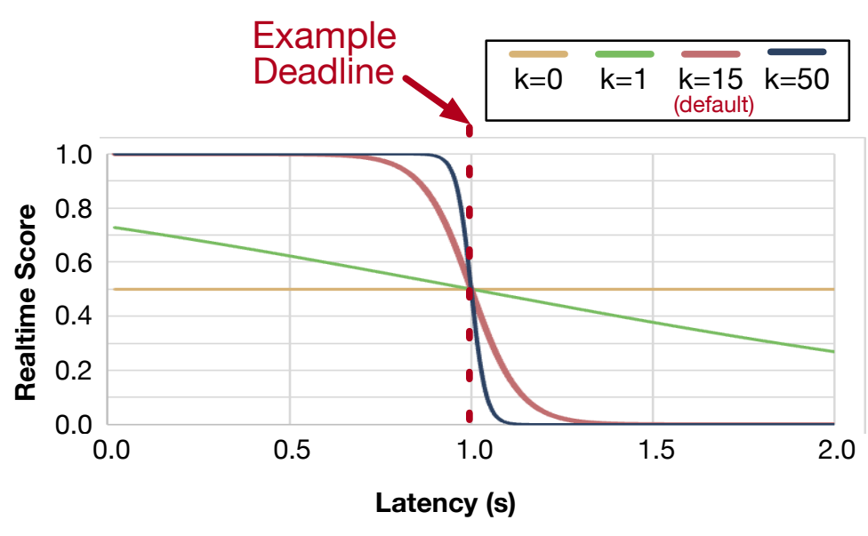
* HT: hand tracking, ES: eye segmentation
AS: action segmentation

- **ML pipeline:** Cascade multiple models to implement complicated functionalities (e.g., ES -> GE in this example; eye pipeline)
- **High FPS eye pipeline:** Enables low latency human-device interaction
- **Dependency in eye pipeline:** ES results are used as inputs of GE

Score Metric

Four Unit Scores

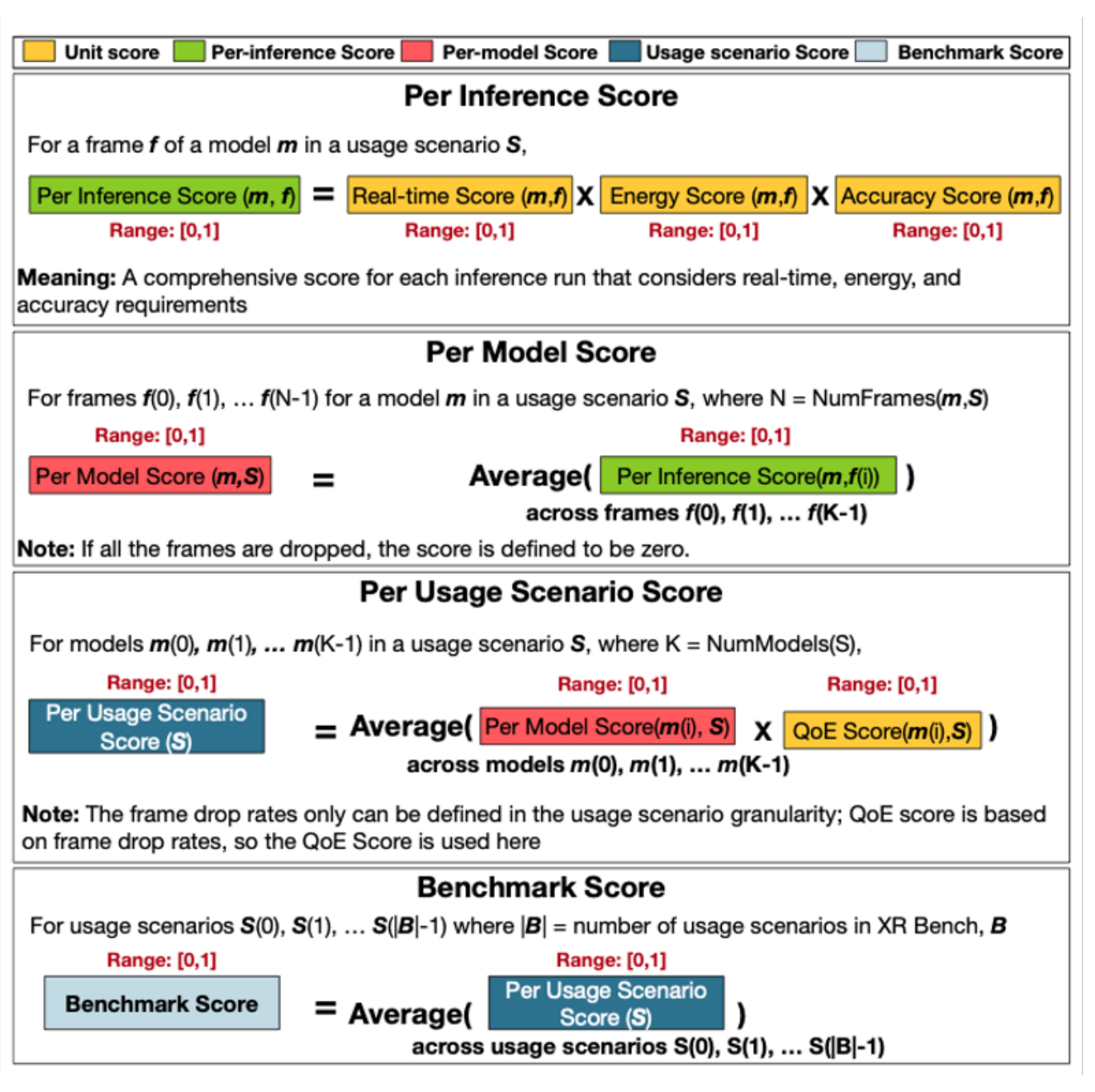
Unit Score	What does it measure?
Real-time	Degree of deadline violations (Not absolute latency!)
Energy	Energy consumption
Accuracy	Relative model performance compared to reported numbers in original papers
Quality of Experience (QoE)	Frame drop rate



mIoU, accuracy, mAP, etc.

All formulated to be higher-is-better metric within [0,1] range

Benchmark Score



Per Inference Score

For a frame ***f*** of a model ***m*** in a usage scenario ***S***,

Per Inference Score (***m***, ***f***) = Real-time Score (***m***,***f***) X Energy Score (***m***,***f***) X Accuracy Score (***m***,***f***)

Meaning: A comprehensive score for each inference run that considers real-time, energy, and accuracy requirements

Per Model Score

For frames ***f***(0), ***f***(1), … ***f***(N-1) for a model ***m*** in a usage scenario ***S***, where N = NumFrames(***m***,***S***)

Per Model Score (***m***,***S***) = **Average(** Per Inference Score(***m***,***f***(i)) **)** **across frames *f*(0), *f*(1), … *f*(K-1)**

Note: If all the frames are dropped, the score is defined to be zero.

Per Usage Scenario Score

For models ***m***(0)**,** ***m***(1)**, …** ***m***(K-1) in a usage scenario ***S***, where K = NumModels(S),

Per Usage Scenario Score (***S***) = **Average(** Per Model Score(***m***(i), ***S***) X QoE Score(***m***(i),***S***) **)** **across models *m*(0), *m*(1), … *m*(K-1)**

Note: The frame drop rates only can be defined in the usage scenario granularity; QoE score is based on frame drop rates, so the QoE Score is used here

Benchmark Score

For usage scenarios ***S***(0), ***S***(1), … ***S***(|***B***|-1) where |***B***| = number of usage scenarios in XR Bench, ***B***

Benchmark Score = **Average(** Per Usage Scenario Score (***S***) **)** **across usage scenarios S(0), S(1), … S(|B|-1)**

Hierarchical Formulation

- Define scores from fine-grained to coarse-grained execution units
  (each inference -> all the inference runs for a model -> usage scenario -> benchmark)

Composable Formulation

- Range of unit scores: [0,1]
- Product of unit scores => benchmark score

Single-metric: Provides comprehensive insights and facilitates industry score submissions

(Break-down scores are still available from the benchmark)

Full detailed formulation is available in the paper!

Case Studies and Conclusion

Evaluation: Various ML Accelerator Systems

Acc. ID	Acc. Style	Dataflow
A	FDA	WS
B		OS
C		RS
D	SFDA[1]	WS + WS (1:1 partitioning)
E		OS + OS (1:1 partitioning)
F		RS + RS (1:1 partitioning)
G		WS + WS + WS + WS (1:1:1:1 partitioning)
H		OS + OS + OS + OS (1:1:1:1 partitioning)
I		RS + RS + RS + RS (1:1:1:1 partitioning)
J	HDA	WS + OS (1:1 partitioning)
K		WS + OS (3:1 partitioning)
L		WS + OS (1:3 partitioning)
M		WS + OS + WS + OS (1:1:1:1 partitioning)

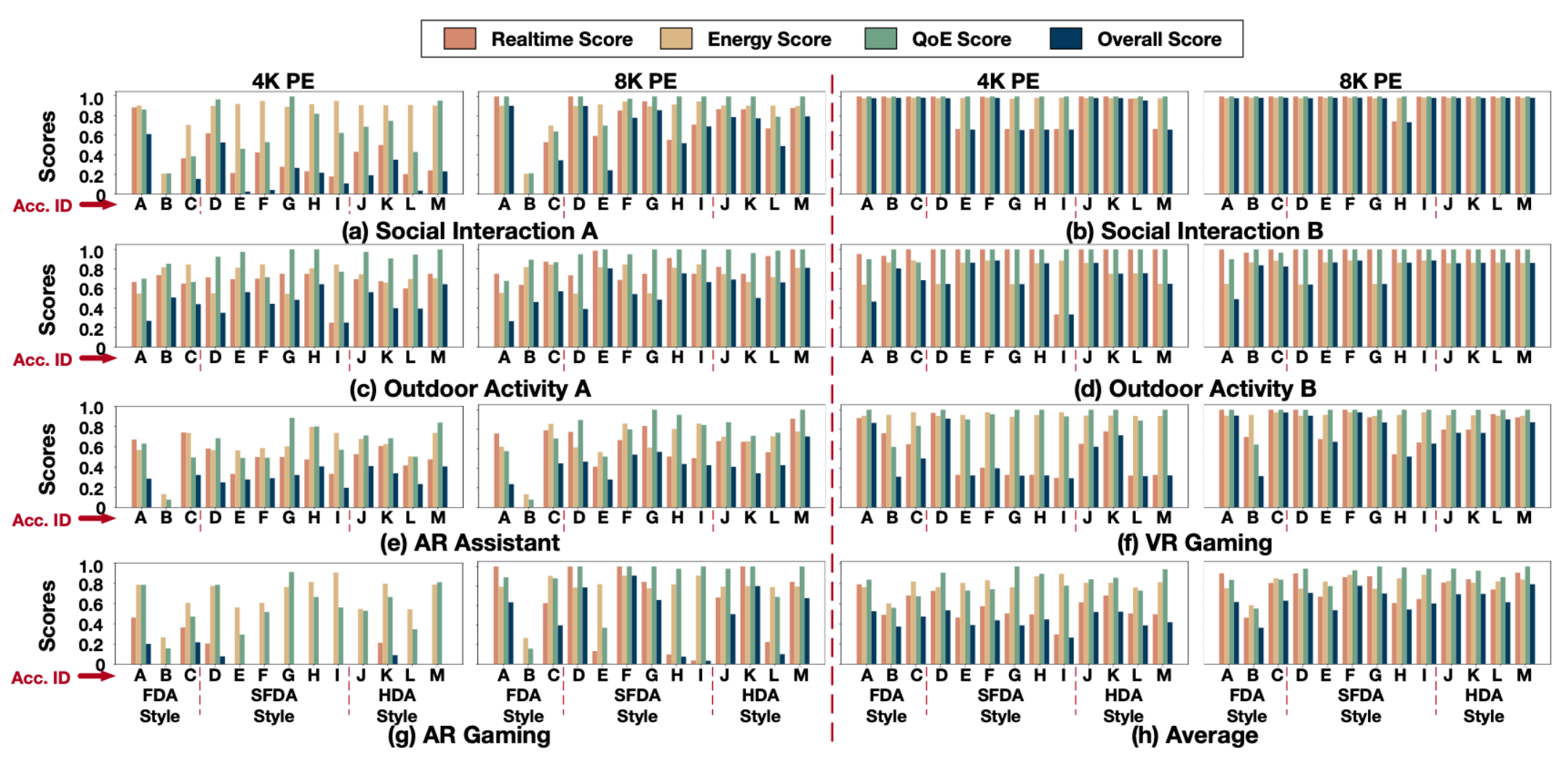
Accelerator Styles

- FDA: Fixed dataflow accelerator
- SFDA: Scaled-out FDA
- HDA: Heterogeneous dataflow accelerator

Accelerator Dataflow

- WS: Weight-stationary
- OS: Output-stationary

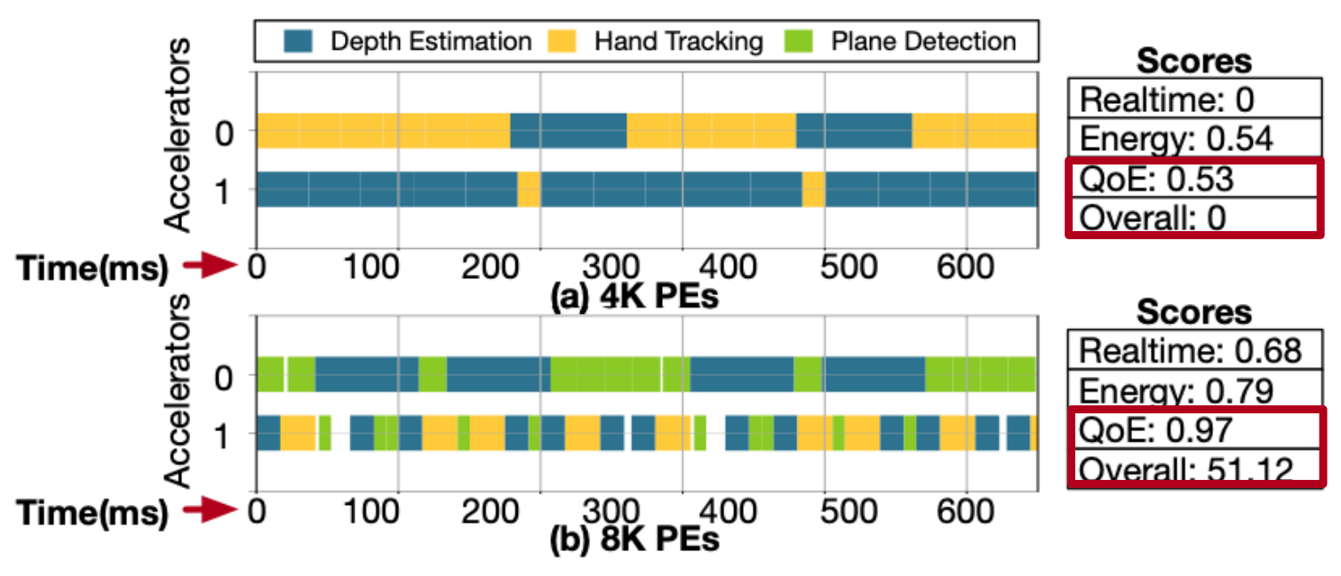
Main Evaluation Results and Insights



(a) Social Interaction A
(b) Social Interaction B
(c) Outdoor Activity A
(d) Outdoor Activity B
(e) AR Assistant
(f) VR Gaming
(g) AR Gaming
(h) Average

- ML Systems for XR needs to be co-designed with usage scenarios
- Optimal accelerator style depends on the chip scale
- Multi-accelerator systems are friendly to XR systems

More insights: Example implication to ML system design for XR



AR gaming execution timeline using accelerator J (WS + OS HDA)

100 % Hardware Utilization 47% frame drop!

~90% Hardware Utilization ~3% frame drop

XR Bench score successfully captures this

Hardware utilization is an incorrect metric for XR ML system design!

Conclusion

- **Real-time MTMM workloads have unique characteristics and new implications to ML system design**
- **We developed XRBench to invite everyone for this new problem domain: ML system for real-time MTMM workloads**
- **XRBench is an open project**

We look forward to working with the community!

https://xrbench.ai