中国科学院高能物理研究所
Institute of High Energy Physics, Chinese Academy of Sciences

CEPC Workshop

Track based online data preselection for CEPC ref detector

Xu Zhang

Institute of High Energy Physics (IHEP)

October 2026

Outline

01 CEPC background and research scope — 03-04
Online processing requirements and the preselection stage

02 Track-based preselection algorithms — 05-11
Hough event selection and CNN hit denoising

03 Heterogeneous online processing framework — 12-21
Data organization, scheduling, memory and validation

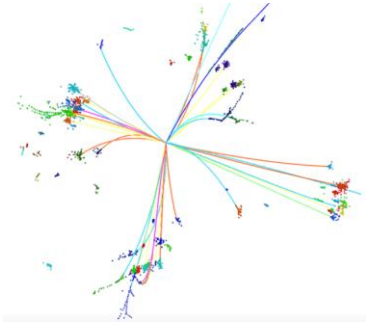
04 Summary and next steps — 22
Current results and future studies

CEPC introduction

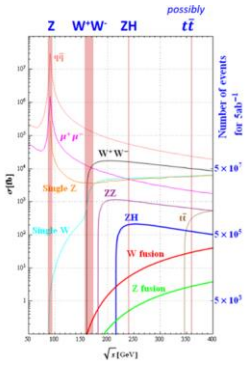
High e^+e^- collision rates call for early data reduction before detailed event reconstruction.

Operating modes considered in this study

Mode	BX interval ns	BX rate MHz
Higgs	277	~4
LowZ	69	~14
HighZ	23	~43

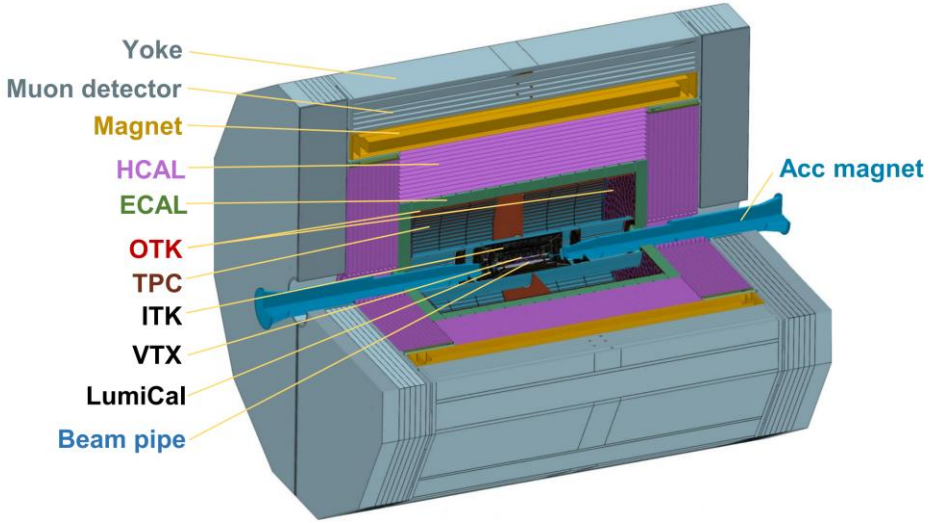


Example collision event



Precision Higgs and electroweak measurements

Z: 91.2 GeV
WW: ~160 GeV
Higgs: 240 GeV

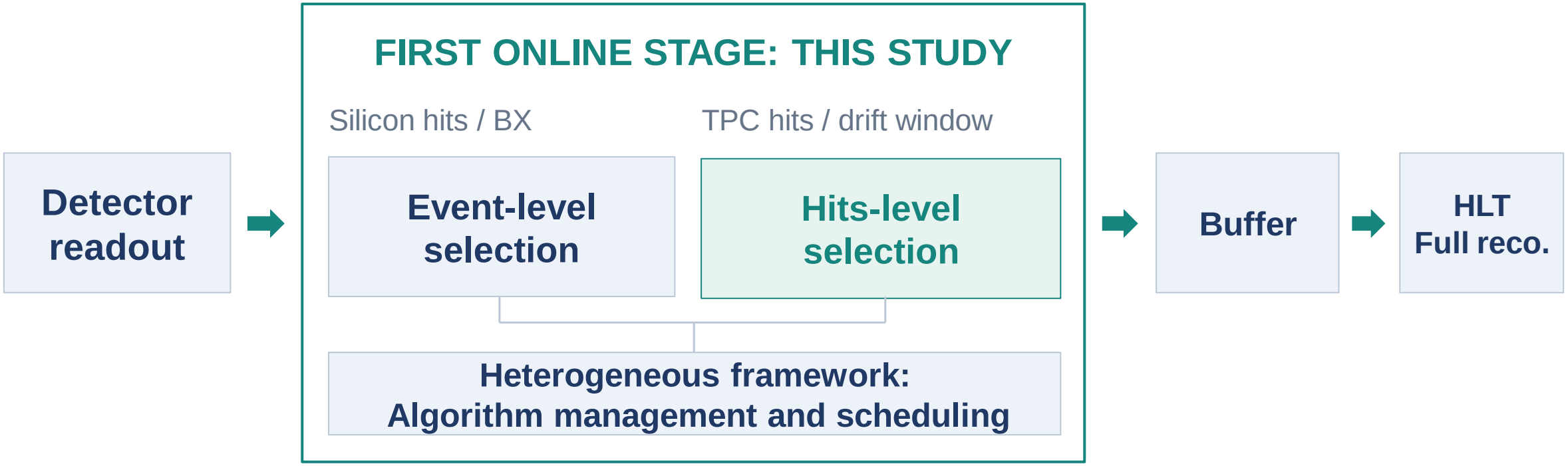


Calorimeters
Calorimeters measure the **energy** deposited by particles.

Tracking Detectors
Tracking detectors provide **precise particle position** measurements. Silicon signals are generated immediately, while TPC readout requires ionization electrons to drift for up to **34 μs**.

Research scope: Online preselection stage before High Level Trigger

Hardware triggering is the baseline, while **software triggering** provides an important complement.
This work focuses on a fully software-based trigger scheme.



Online preselection can use energy or tracking information.
Currently, for software-trigger our work focused on track-based preselection algorithms.

Online preselection methods:

Online data reduction

Event-level selection

Hough transform

VTX ITK OTK hits
One collision event

Keep / reject events

Hit-level selection

CNN

TPC hits
Complete 34 μs TPC window

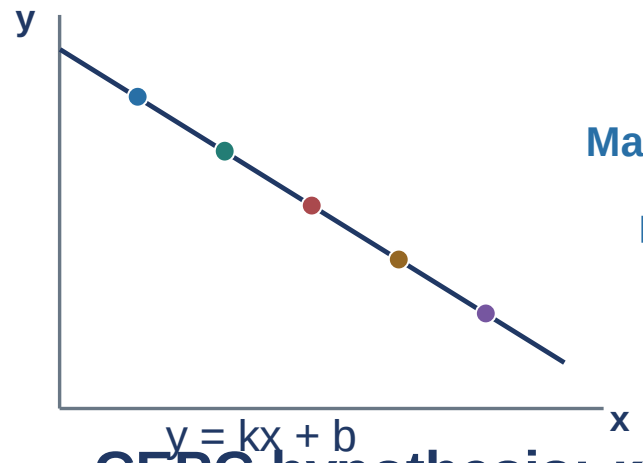
Keep / remove hits

Hough: An event is selected if a track satisfying the selection criteria can be reconstructed.

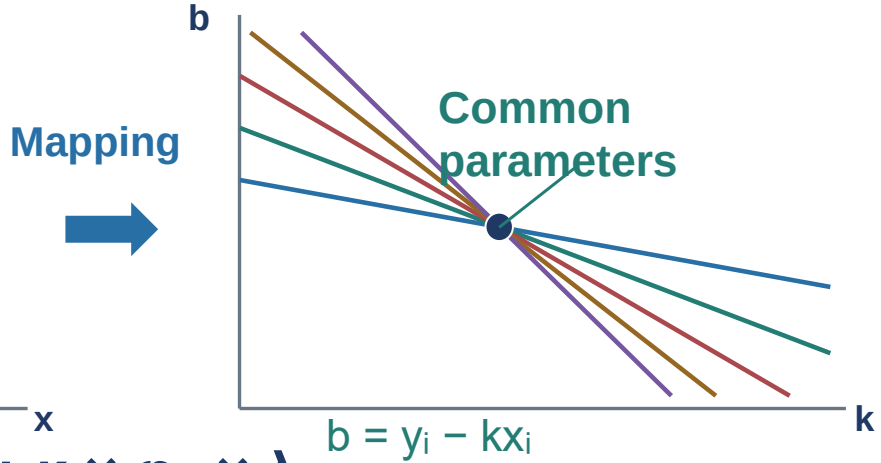
The Hough transform is a common fast track-finding method discussed in the **CEPC TDR**.

Each hit votes in parameter space to find groups of hits that satisfy the same track parameter.

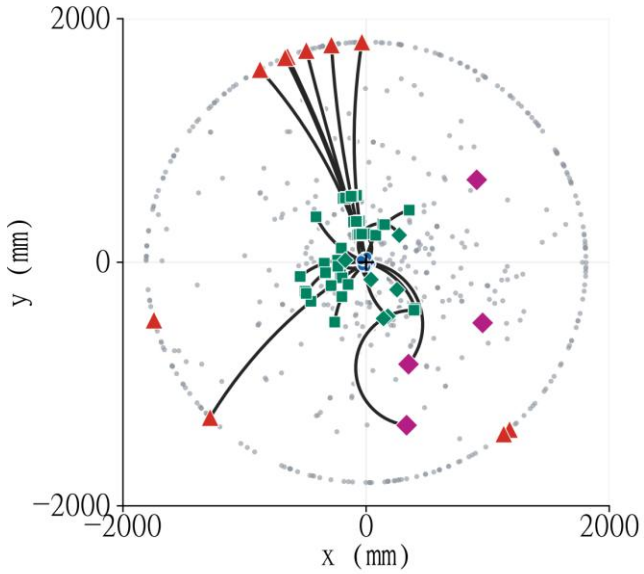
Spatial hits: straight-line example



Common parameter intersection



ROOT example: Higgs qq



CEPC hypothesis: $\kappa \times \varphi_0 \times \lambda$

3d voting table



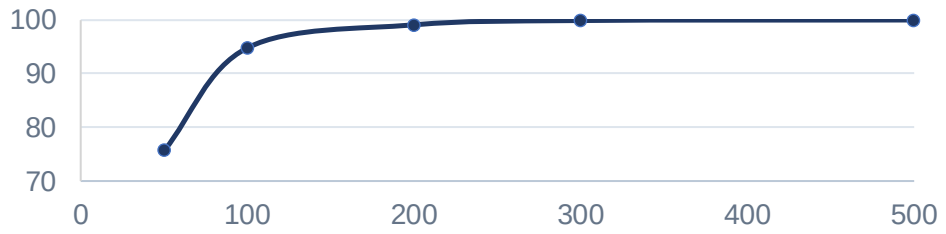
Time complexity: O(table size * number of hits)
Space complexity: O(table size ^ 3)

Hough optimization(CPU): Improve background rejection without significantly increasing computation time.

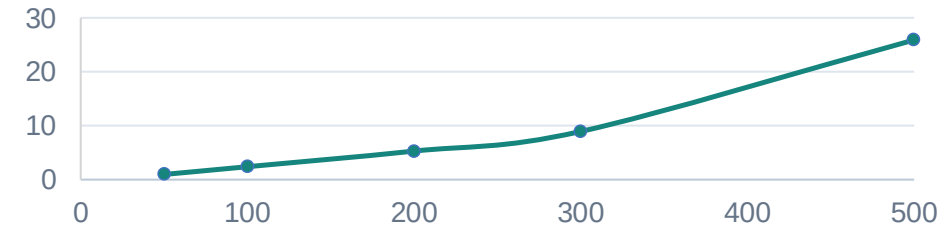
Fine grids: higher rejection, higher cost

If we want higher background rejection, we need a finer grid. But the larger voting table increases CPU time.

Background rejection(%) / Table size



CPU time(ms) / Table size



99.025% rejection, 5.179 ms/event

Our method: two level selection

Refine candidates without enlarging the voting table.

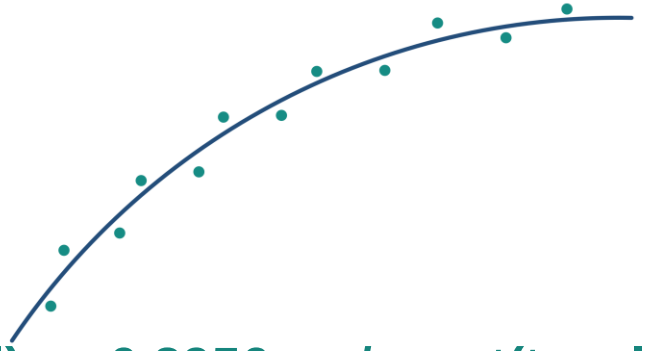
Coarse selection, fine checking.

1. Coarse Hough selection

First, perform Hough voting on a coarse grid.
12 × 24 × 96 = 27,648 voting bins.
Keep cells passing VTX/ITK/OTK cuts: (3, 2, 0).

2. Check Hits Against the Hough Track

The Hough transform identifies a candidate track and its associated hits. Check whether the **deviations** of these hits from the track are within tolerance.



deviation value: 0.05 Higgs
deviation value: 0.14 LowZ
deviation value: 0.12 HighZ

At >99% rejection in Higgs mode: 5.179(finer grid) vs 0.2956 ms/event(two-level), 17.5× faster

GPU Hough(two selection version):

Split the 3D voting table into 2D slices, each stored in a block's shared memory.

A Global table, batch128 (naïve gpu version)

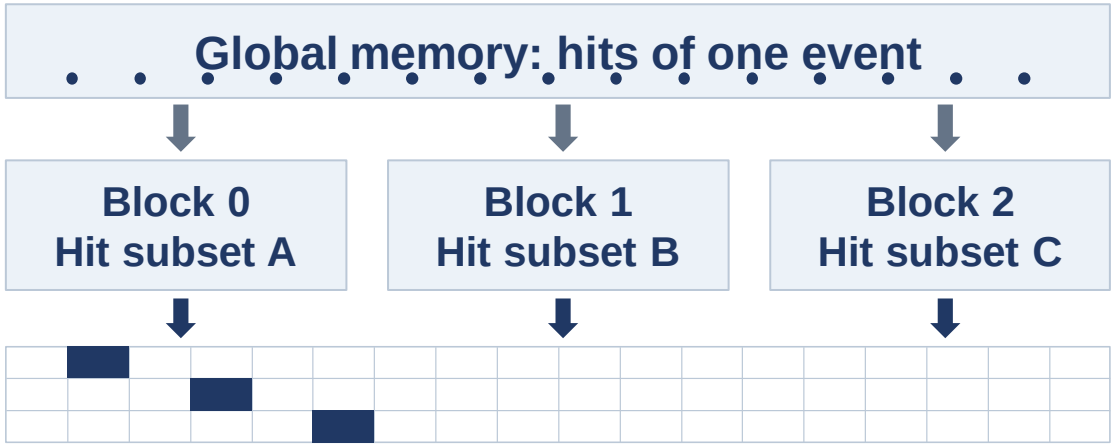
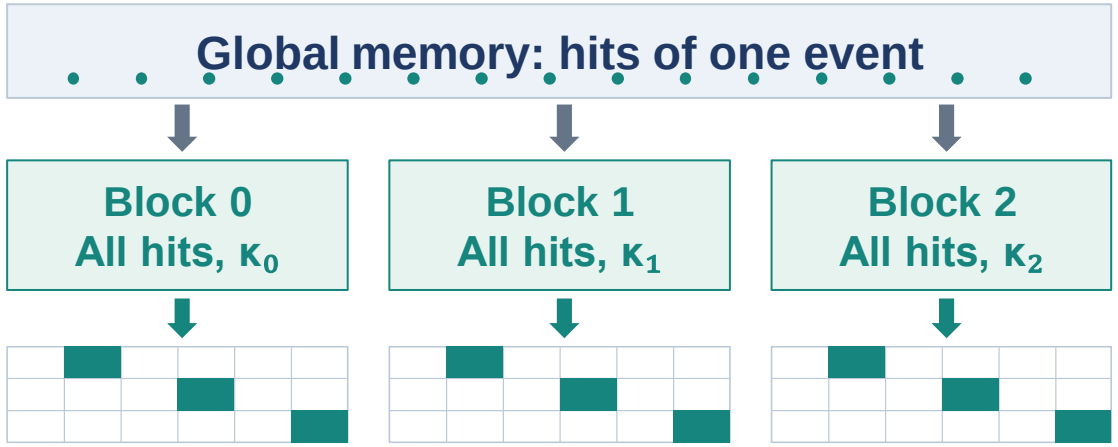


Table is stored in the global memory slices.

B Shared slices, batch2048 (optimization)



Shared κ slice 0 Shared κ slice 1 Shared κ slice 2

Each block reads the event hits and updates its own slice in shared memory

Mode	Global μs/per event	Shared μs/per event	Ratio
Higgs	2.2857	**0.8704**	**2.63×**
LowZ	1.3070	**0.3006**	**4.35×**
HighZ	1.5288	**0.3836**	**3.99×**

Hough results: physics efficiency and processing cost

We use the same data sources as the CEPC TDR, with 4,000 events of each type.

Each signal event is mixed with data from two collisions. Each background event is formed from data from two collisions.

Grid 12 × 24 × 96, hit thresholds (VTX, ITK, OTK) = (3, 2, 0). **CPU: 8558P GPU:RTX 4090**

Mode / deviation	$\mu^+\mu^-$	Z(vv)H(μμ)	Z(vv)H(bb)	qq	Background rejection
Higgs	85.075%	98.325%	98.850%	98.400%	**99.275%**

Muon-detector information can help recover $\mu^+\mu^-$ events missed by track-based selection.

Mode / deviation	Bhabha	$\mu^+\mu^-$	$\tau^+\tau^-$	qq	Background rejection
LowZ	99.425%	97.350%	97.375%	99.950%	**99.425%**
HighZ	99.300%	97.225%	96.850%	99.950%	**99.425%**

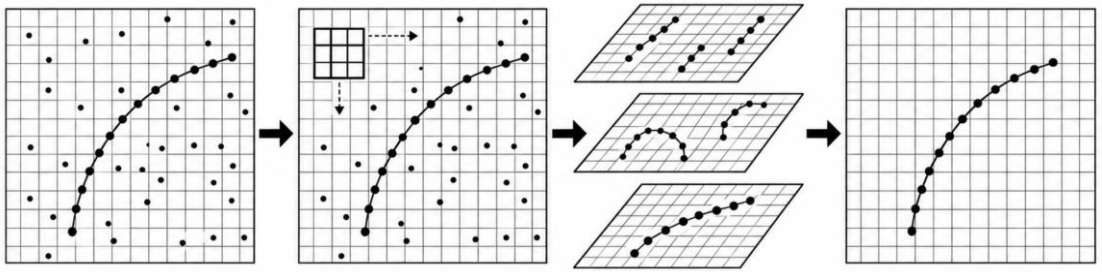
Processing cost

Mode	CPU ms/event	GPU μs/event, resident input	GPU SM Active (%)
Higgs	0.2956	**0.8704**	**97.60%**
LowZ	0.0653	**0.3006**	**95.88%**
HighZ	0.0985	**0.3836**	**97.24%**

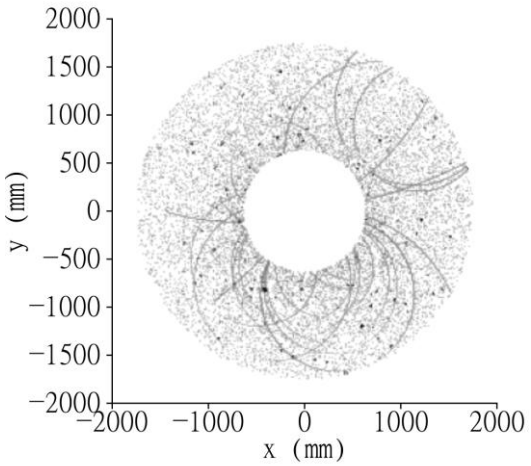
TPC CNN: hit selection through image denoising

The CNN input is an image formed from all TPC hits within a 34 μs window.

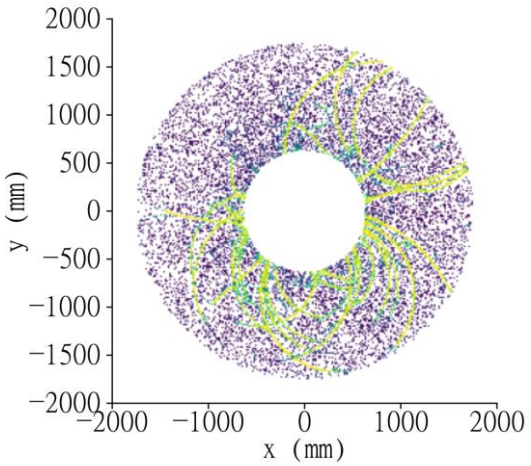
Local and multiscale CNN features



Mixed input



CNN score



Mode	collisions / 34 μs
Higgs	245
LowZ	986
HighZ	2957

CNN results: all physics processes in three modes

S: retained signal hits. R: removed background hits.

Higgs, cut 0.15

Process	S %	R %
$\mu^+\mu^-$	99.65	95.88
Z(vv)H(μμ)	99.68	95.44
Z(vv)H(bb)	98.46	79.97
qq	98.12	83.63
BG only	n/a	97.50

LowZ, cut 0.20

Process	S %	R %
Bhabha	99.45	97.66
$\mu^+\mu^-$	99.89	97.74
$\tau^+\tau^-$	98.79	97.21
qq	98.17	86.59
BG only	n/a	98.74

HighZ, cut 0.31

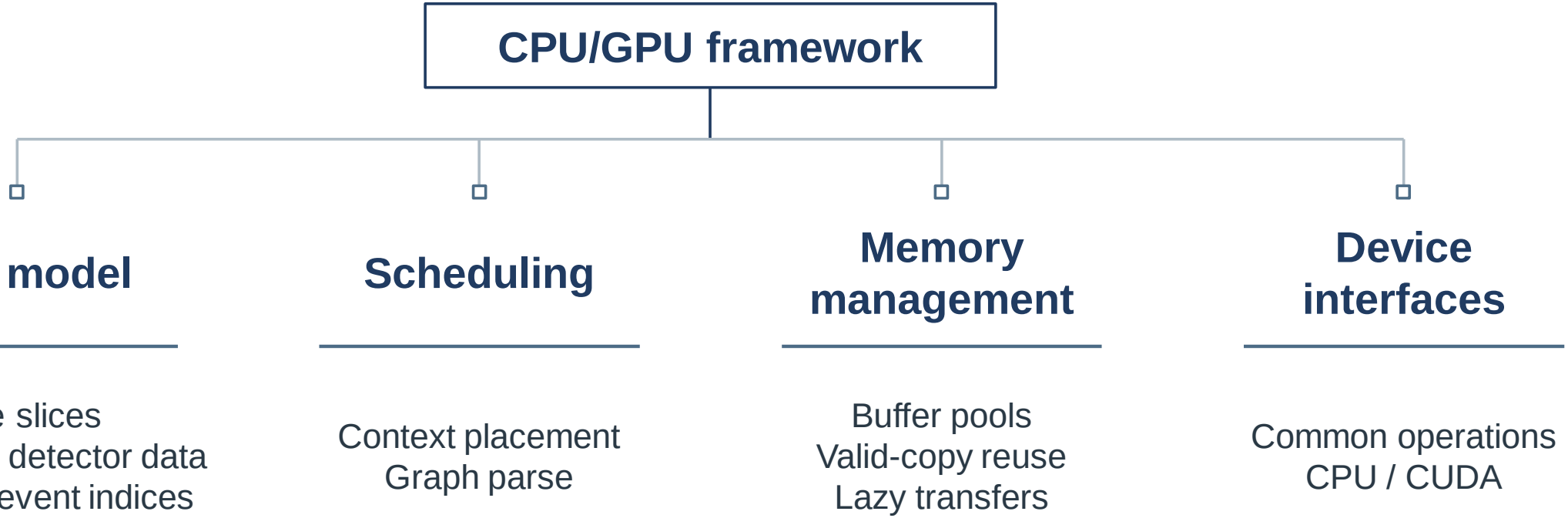
Process	S %	R %
Bhabha	98.26	89.85
$\mu^+\mu^-$	98.88	89.94
$\tau^+\tau^-$	96.77	88.54
qq	**97.55**	**53.73**
BG only	n/a	94.13

256^2 image size, FP16 **RTX4090, FP16, batch8**

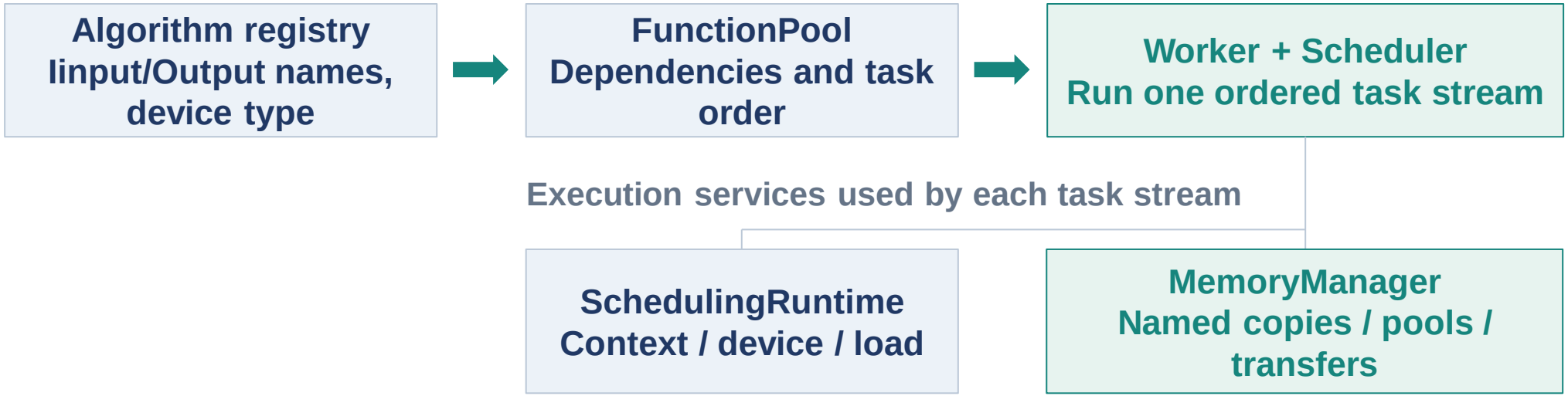
Mode	ms / 34 μs	GPU SM Active
Higgs	**0.6517**	**87.32%**
LowZ	**0.6458**	**88.41%**
HighZ	**0.7386**	**79.93%**

Framework overview: four parts of one execution system

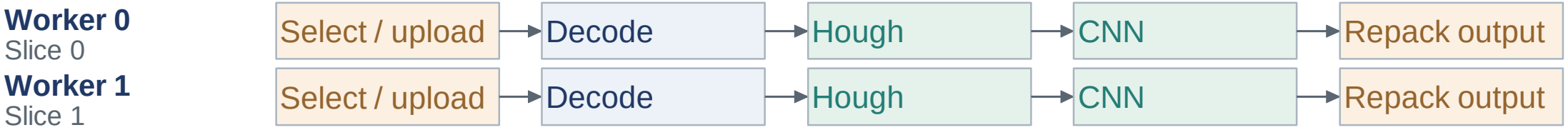
The framework separates algorithms from infrastructure and unifies execution and resource management.



Framework structure:



One stream per Worker, data parallelism across slices





Input data: time-slice data

Individual events have overlapping TPC readout ranges. A time slice batches several BX IDs.

Per-BX packet: append a drift range for each event

One BX event

VTX	ITK	OTK	ECAL	HCAL	Muon	TPC
						34 μs

Time-slice packet: many BX IDs, one final drift tail

BX 0	VTX	ITK	OTK	ECAL	HCAL	Muon	TPC
BX 1	VTX	ITK	OTK	ECAL	HCAL	Muon	TPC
… BX n	VTX	ITK	OTK	ECAL	HCAL	Muon	TPC
							34 μs

Grouping data into time slices enables TPC data reuse and helps exploit GPU parallelism.

Data payload: offsets array enable parallel data access

CPU and GPU addresses differ, but **offsets** remain the same.

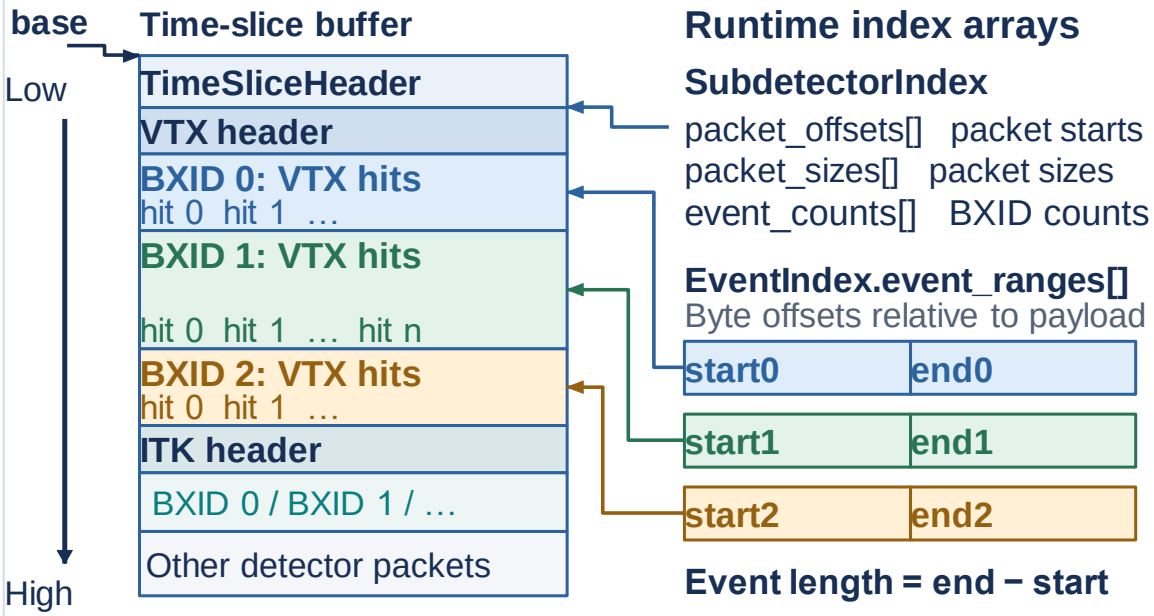
Offset arrays record data bias, enabling uniform access on both CPUs and GPUs.

Relative offsets for addressing

CPU and GPU use their local base address plus the same offset.

CPU local address
CPU_base + bias
→ **Bulk copy** →
GPU local address
GPU_base + bias

Same layout: the base changes, offsets do not.



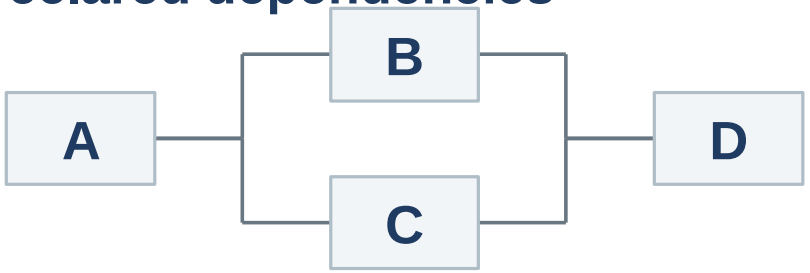
Offsets locate each data range
Parallel across detectors, events and hits

These arrays enable parallel CPU/GPU access across detectors and events data.

Graph generate: task order and time-slice parallelism

Algorithms declare their dependencies through configuration at registration.

Declared dependencies



Topological order

One stream

A → B → C → D

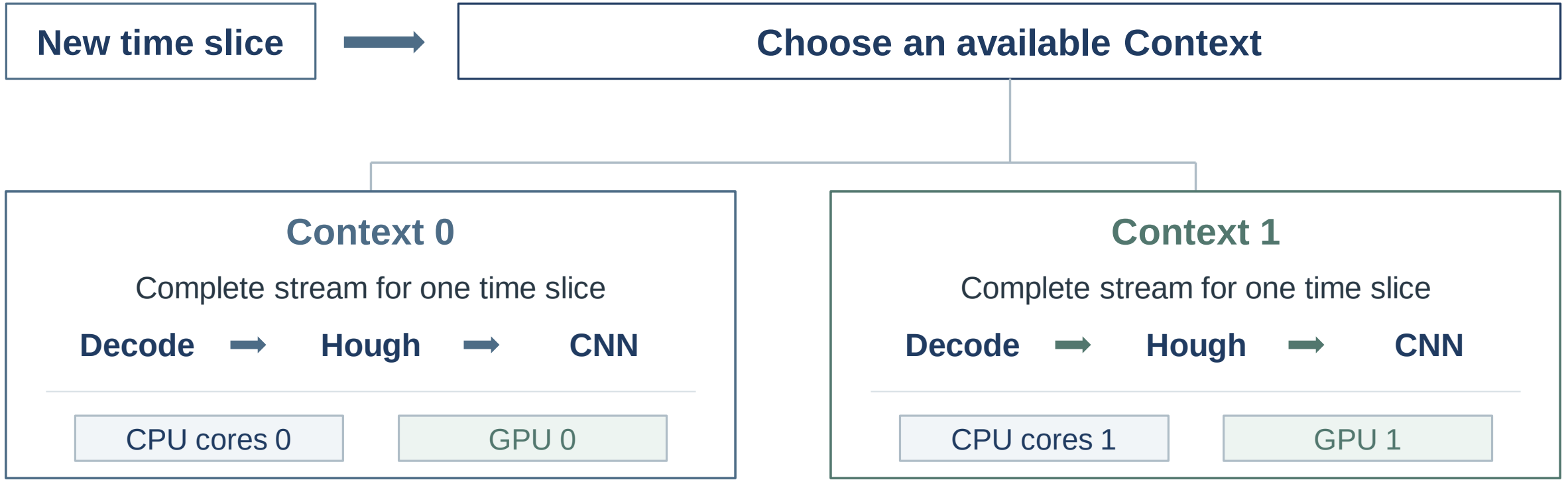
Each time slice data and its associated tasks are assigned to a single worker for execution.

Worker 0 / Slice 0	Select / upload	Decode	Hough	CNN	Repack
Worker 1 / Slice 1	Select / upload	Decode	Hough	CNN	Repack
Worker 2 / Slice 2	Select / upload	Decode	Hough	CNN	Repack

Sequential within a Worker. Data parallelism across time slices.

Schedule strategy: Each time slice is processed within a single device group.

A Context groups CPU cores and a GPU.
Each worker runs within a single context, allowing tasks in the same time slice to reuse the results.



The worker will be scheduled to the context with the lightest load.

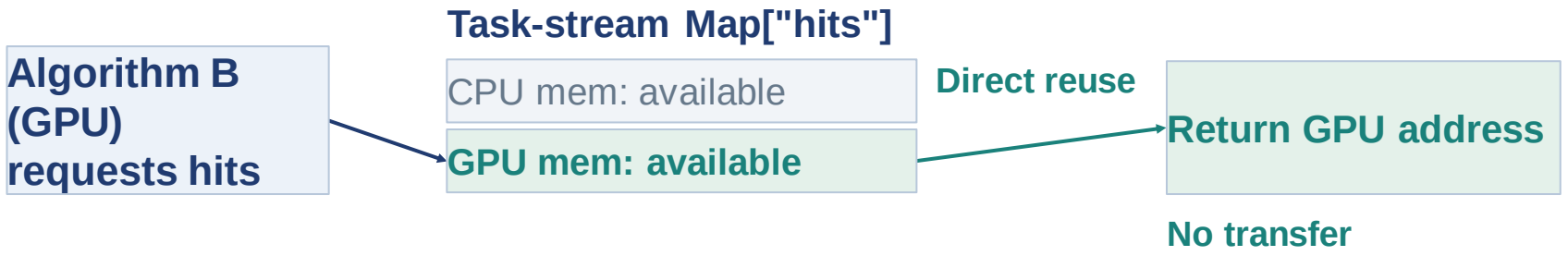
Memory manager:

Our memory management is based on a per-time-slice map and memory pools.

Each worker has its own map for the current time slice. Different time slices do not share this map.

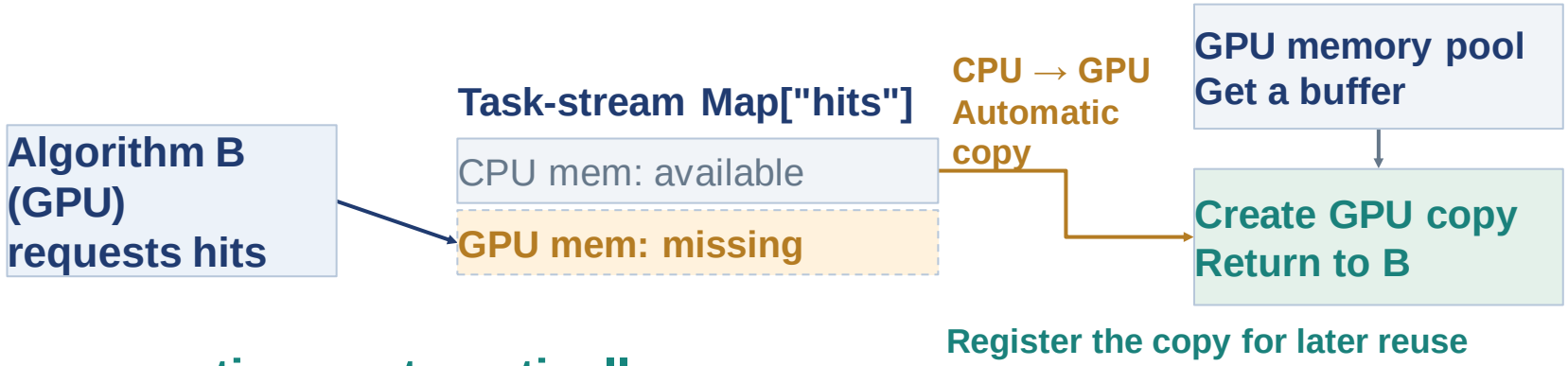
1 GPU memory exists

Reuse the resident data address.
No new buffer or transfer.



2 GPU memory missing

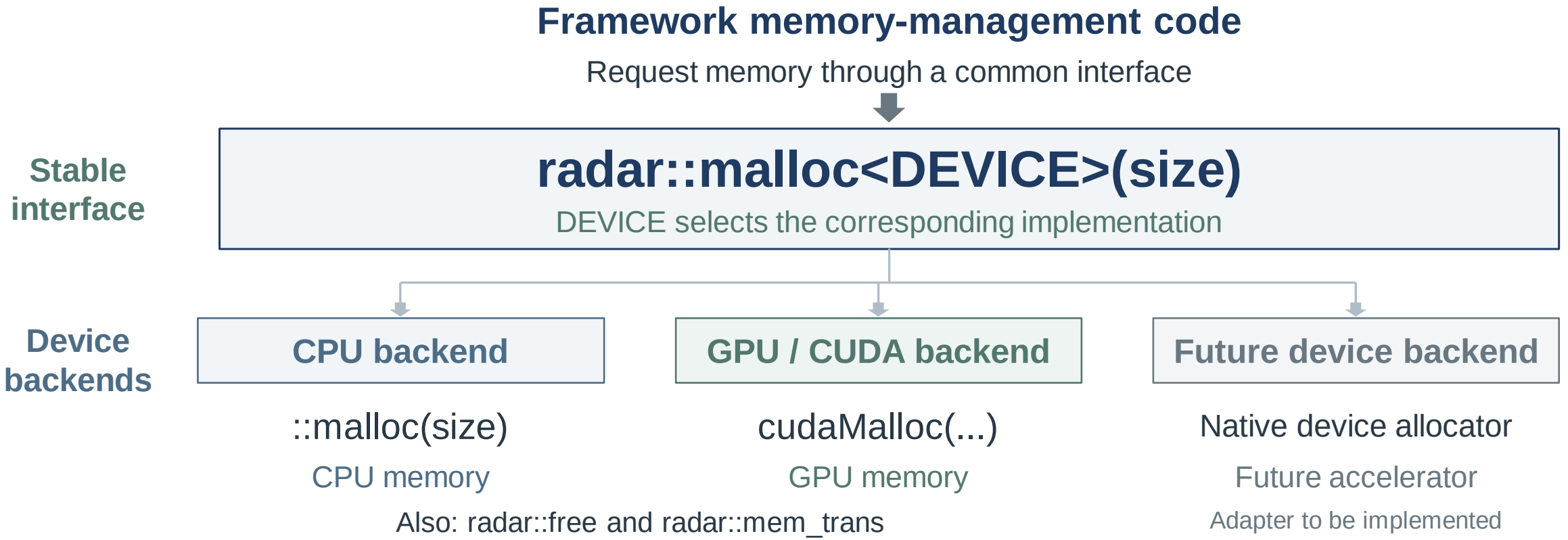
If no local memory exists, find one on another device.
Transfer it on demand.

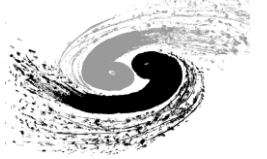


The framework handles these operations automatically.
Users only specify the data they need, without managing transfers.

Device interface:

We abstract all device operations to make the system easier to extend.
Upper-level code calls radar::malloc. The device backend handles the allocation.





Validation: full chain(framework + algorithm)

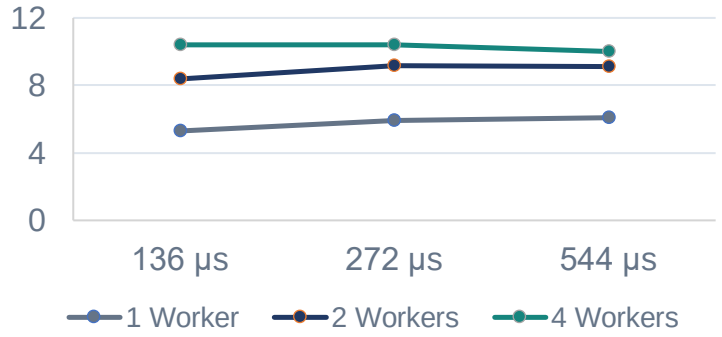
1 Worker, 136 μs timeslice, one RTX4090.

Stage	Higgs ms/slice	Higgs GB/s	LowZ ms/slice	LowZ GB/s	HighZ ms/slice	HighZ GB/s
01 Metadata / indices	0.0138	2995.70	0.0420	787.22	0.1235	1459.72
02 Silicon upload	0.9054	45.82	0.8073	40.93	3.4953	51.56
03 Silicon decode	0.0580	715.02	0.0941	351.16	0.2949	611.16
04 Full Hough selection	0.4660	89.02	0.6056	54.56	2.2753	79.21
05 Data selection	0.0149	2789.82	0.0453	729.78	0.1947	925.82
06 TPC upload	0.5690	72.91	0.4605	71.75	2.2352	80.64
07 TPC decode	0.0360	1151.60	0.0326	1012.25	0.1278	1410.09
08 CNN imaging / selection	**3.8101**	**10.89**	**3.7497**	**8.81**	**4.6587**	**38.69**
09 TPC hit compaction	0.1144	362.55	0.1075	307.31	0.2769	650.80
10 TPC download	0.1177	352.58	0.0936	352.85	0.9166	196.63
11 Other-detector compaction	0.6864	60.44	0.4879	67.73	2.6628	67.69
12 Output repacking	0.8971	46.25	0.6783	48.72	4.0240	44.79
Checks / overhead	0.1270	326.58	0.1145	288.48	0.3639	495.31
Full chain	**7.8158**	**5.31**	**7.3190**	**4.51**	**21.6497**	**8.33**

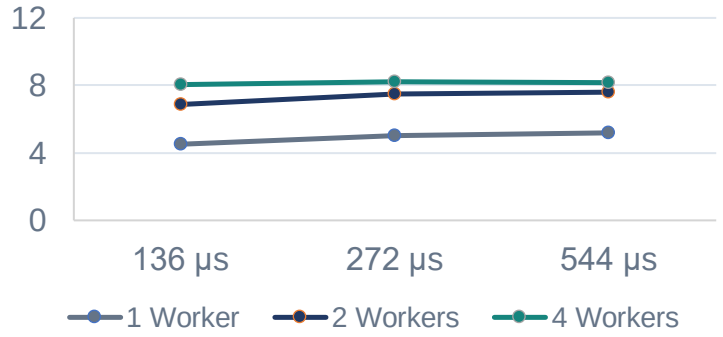
CNN-based hit-level selection is the main computational bottleneck.

Single-process throughput with different time-slice sizes and worker counts (single RTX 4090).

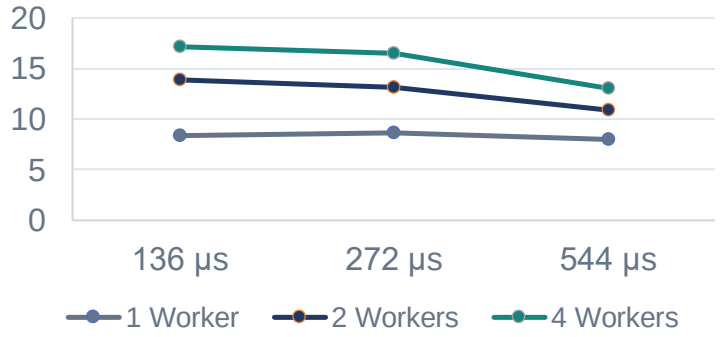
Higgs(GB/s / us)



LowZ(GB/s / us)



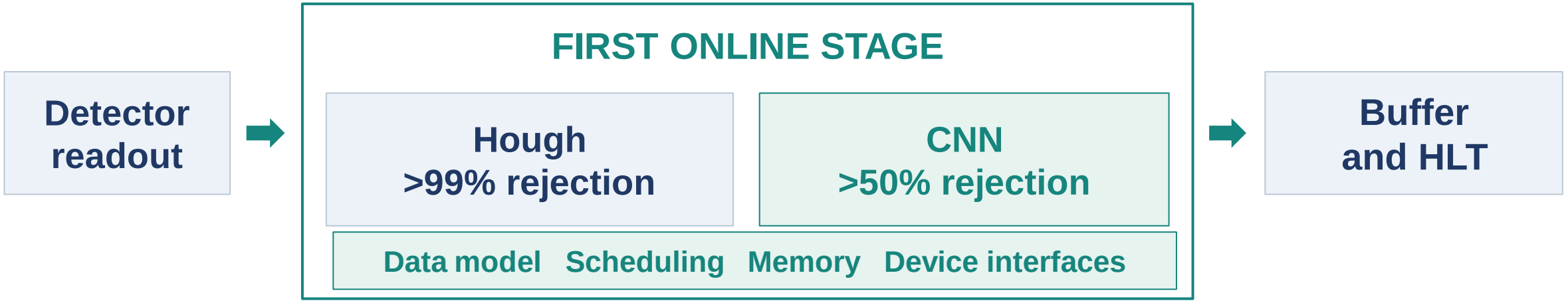
HighZ(GB/s / us)



Resource usage(preselection stage):

Mode	Input MHz (total)	Time slice / Workers	Chain GB/s per RTX 4090	Equivalent input MHz per RTX 4090	RTX 4090 count
Higgs	~4	136 μs / 4	**10.42**	**~0.123**	**~30**
LowZ	~14	272 μs / 4	**8.22**	**~0.490**	**~30**
HighZ	~43	136 μs / 4	**17.19**	**~0.564**	**~80**

Summary: A track-based, software-only trigger scheme.



The end-to-end track-based preselection chain achieves over **99%** background event rejection, with over **97%** signal event retention for most processes.
Within TPC windows, it removes over **50%** of background hits, with over **98%** signal hit retention for most processes.

For preselection stage, **30** RTX 4090 GPUs each for Higgs and LowZ, and **80** for HighZ.

Next Plan:

Integrate algorithms based on **muon** detector and **calorimeter**.

Consider **streaming data processing** and more complex event-mixing scenarios.

Conduct further framework tests, including **stability** and **multi-GPU** testing.