

GPU-Accelerated Standalone Particle-to-Grid Library for Cosmological Simulations

Swiss SKA Days 2026
September 2, 2026, Neuchâtel

Osman Seckin Simsek



Platform for Advanced Scientific Computing



<https://github.com/unibas-dmi-hpc/SPH-EXA>

ETH zürich

Alexandre Refregier
Michele Bianco
Tommaso Boschi



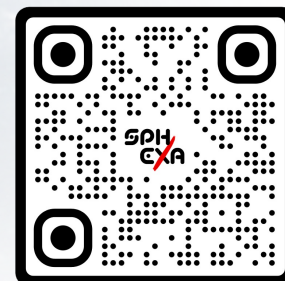
Florina Ciorba
Osman Seckin Simsek
Rubén Cabezón



Elena Gavagnin
Philipp Denzel



Christopher Bignamini
Sebastian Keller
Jean-Guillaume Piccinali



Motivation

- Cosmological simulations evolve N-body or hydrodynamical particles
- Many analyses require homogeneous Cartesian grids:
 - FFT-based power spectrum estimation (matter, 21-cm)
 - Grid-based radiative transfer (pyC²Ray, C²Ray)
- Accurate, scalable particle-to-grid (P2G) mapping is a prerequisite for these workflows
- Poor gridding propagates into biased power spectra and degraded constraints

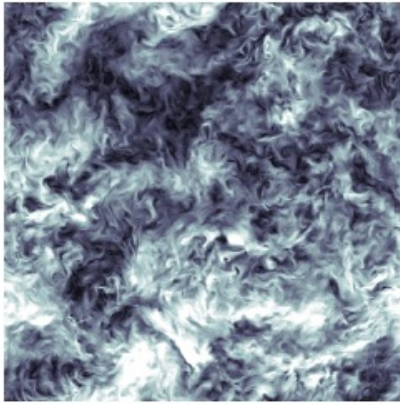
Goal: Reduce domain decomposition overhead while maintaining simulation accuracy

Motivation

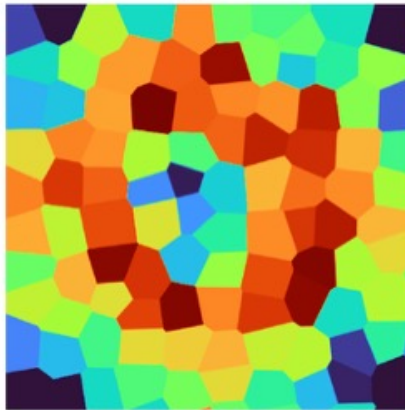
- **The Challenge:**
 - Fourier analysis needs globally consistent periodic grids
 - RT codes expect regular grid inputs from particle simulations
 - Codes such as SPH-EXA use SFC-based particle decomposition
 - FFT/RT solvers use Cartesian grid decomposition
- **The Goal:**
 - Existing approaches are often monolithic and application-specific
 - Limited GPU exploitation and poor reusability
 - Need: general GPU-resident P2G framework that:
 - produces FFT-quality homogeneous grids,
 - integrates with SFC-based MPI decomposition using Cornerstone library,
 - maintains accuracy at cosmological scales.

Target Workflow – Power spectrum of a turbulent box

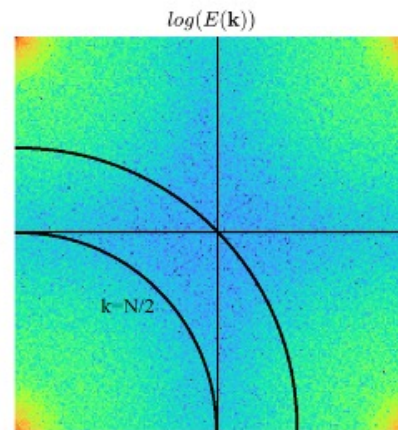
3D Hydrodynamics Simulation + Stirring



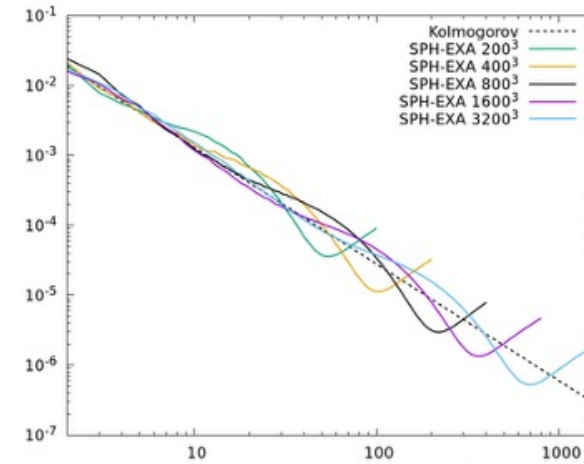
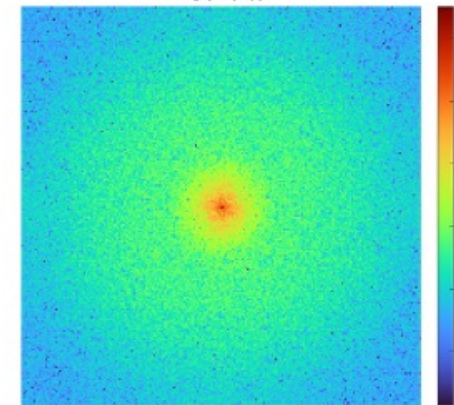
3D Particle-to-Grid
Transformation



3D FFT



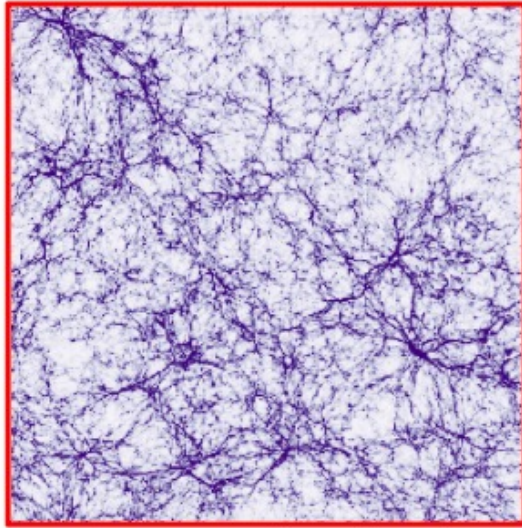
Shifting



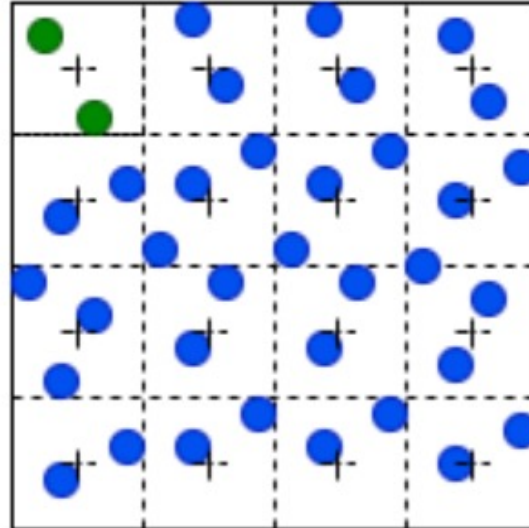
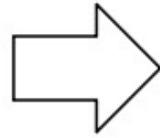
Power Spectrum



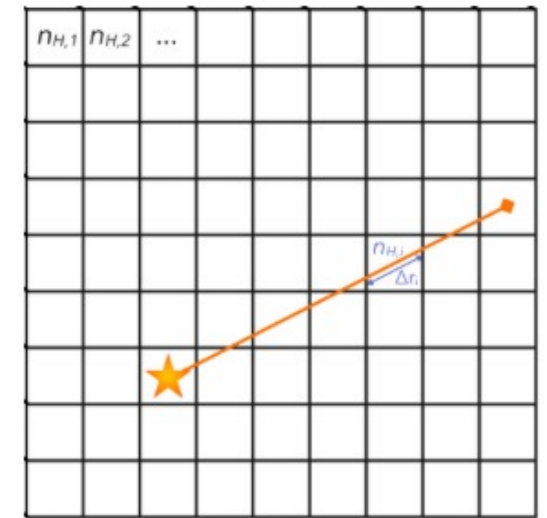
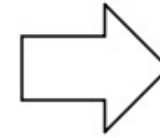
Target Workflow – Radiative Transfer (RT) calculation



Cosmological
simulation

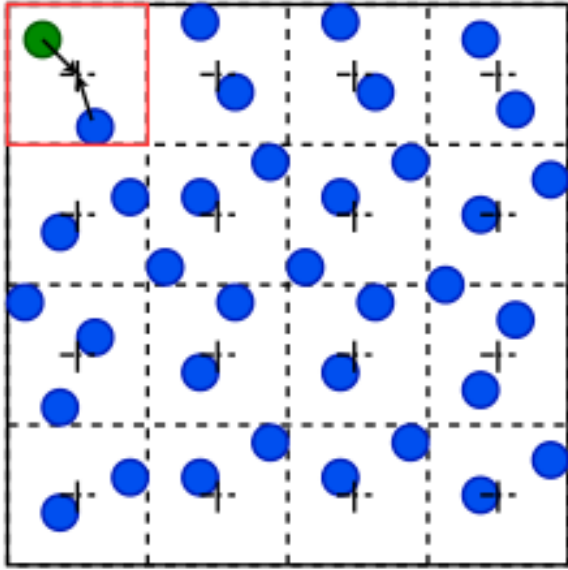


3D Particle-to-Grid
transformation

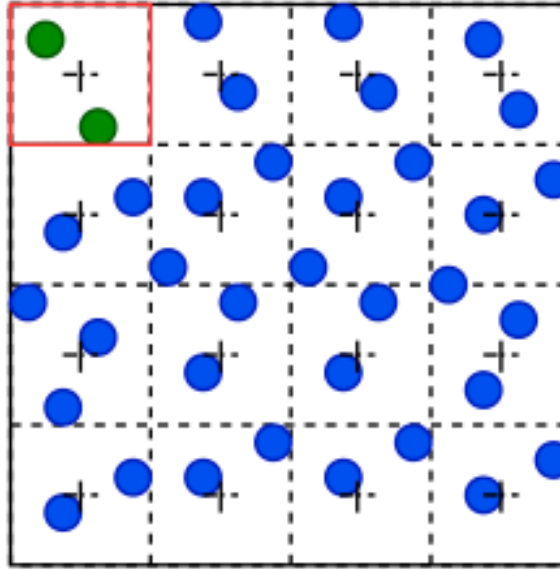


Radiative Transfer
calculation

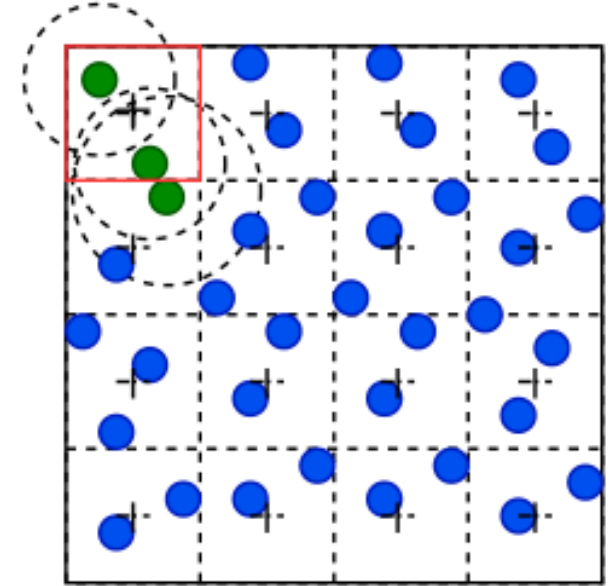
Interpolation Methods



Nearest-neighbor



Cell-average



SPH

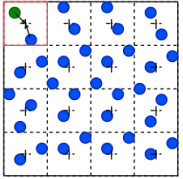
Output grid: 1D z-slab decomposition (each rank owns multiple z-layers)

All three methods share common remote-contribution pattern:

- Local GPU accumulation via atomics
- MPI all-to-all variable-length exchange (GPU-Direct)
- Pack remote contributions per destination rank
- Second kernel accumulates received contributions

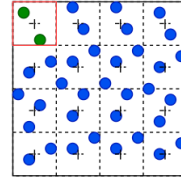
Multi-field optimization: single classification pass, multiple scalar fields (mass, velocities, etc.)

Interpolation Methods



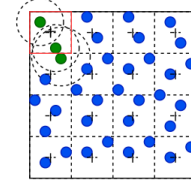
Nearest-neighbor

- Each particle's mass assigned entirely to the closest grid cell
- Uses Hilbert key decoding for fast integer-based cell lookup
- GPU: one thread per particle, atomic accumulation to local grid
- Remote contributions recorded and exchanged via MPI all-to-all
- $O(1)$ per-particle complexity — fastest method
- Piecewise-constant field; minimal memory traffic



Cell-average

- Particle masses that belong within a cell are averaged
- Uses Hilbert key decoding for fast integer-based cell lookup
- Same local/remote classification and atomic accumulation as nearest-neighbor
- ~23% overhead vs. nearest-neighbor at 8 nodes
- Reduces grid-scale noise relative to nearest-neighbor



SPH

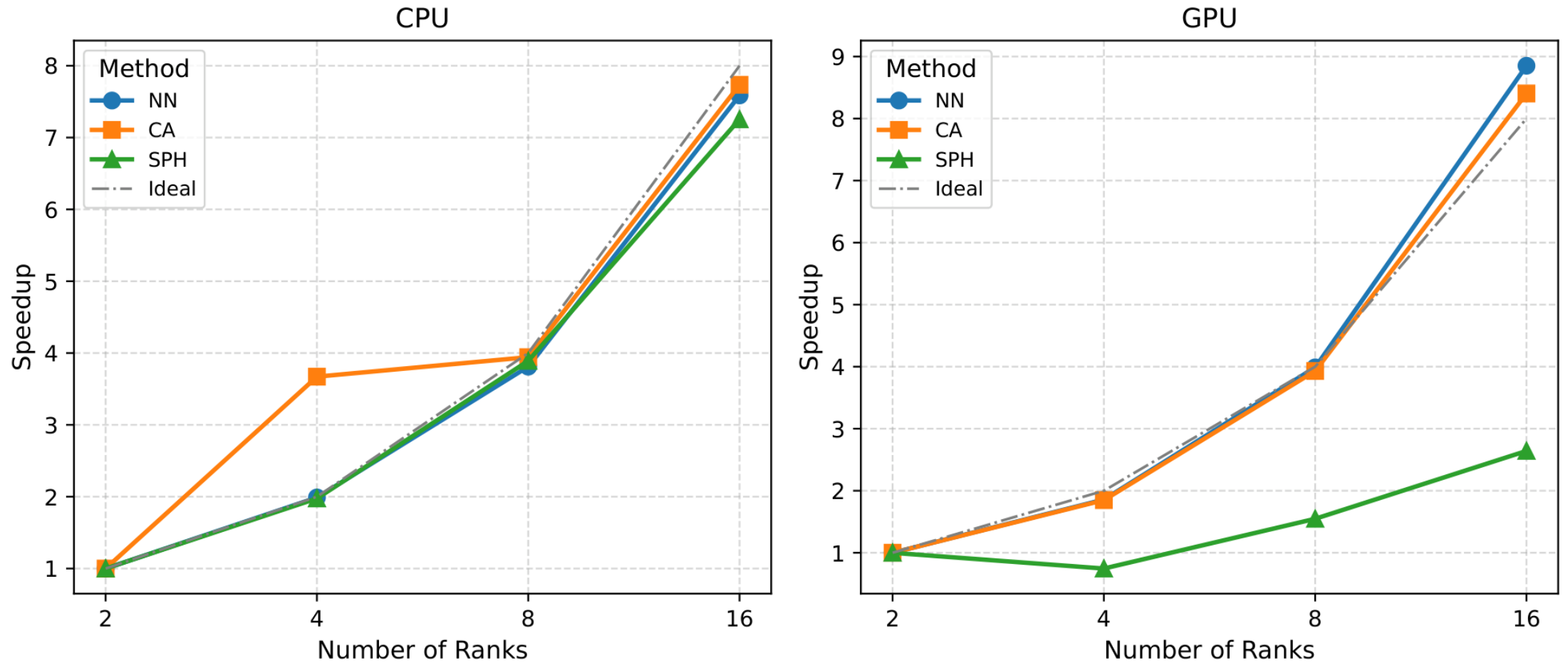
- Spreads each particle across cells within compact support radius $2h$
- Standard 3D cubic spline kernel $W(r, h)$
- Variable stencil size: $(2s+1)^3$ cells per particle, $s = \lceil 2h/\Delta x \rceil + 1$
- Bounded remote buffer (16M entries) prevents GPU memory exhaustion
- Early termination for cells outside spherical support
- 2.8× slower than NN — higher fidelity, irregular access patterns

Factorial Design of Experiments For Performance Analysis

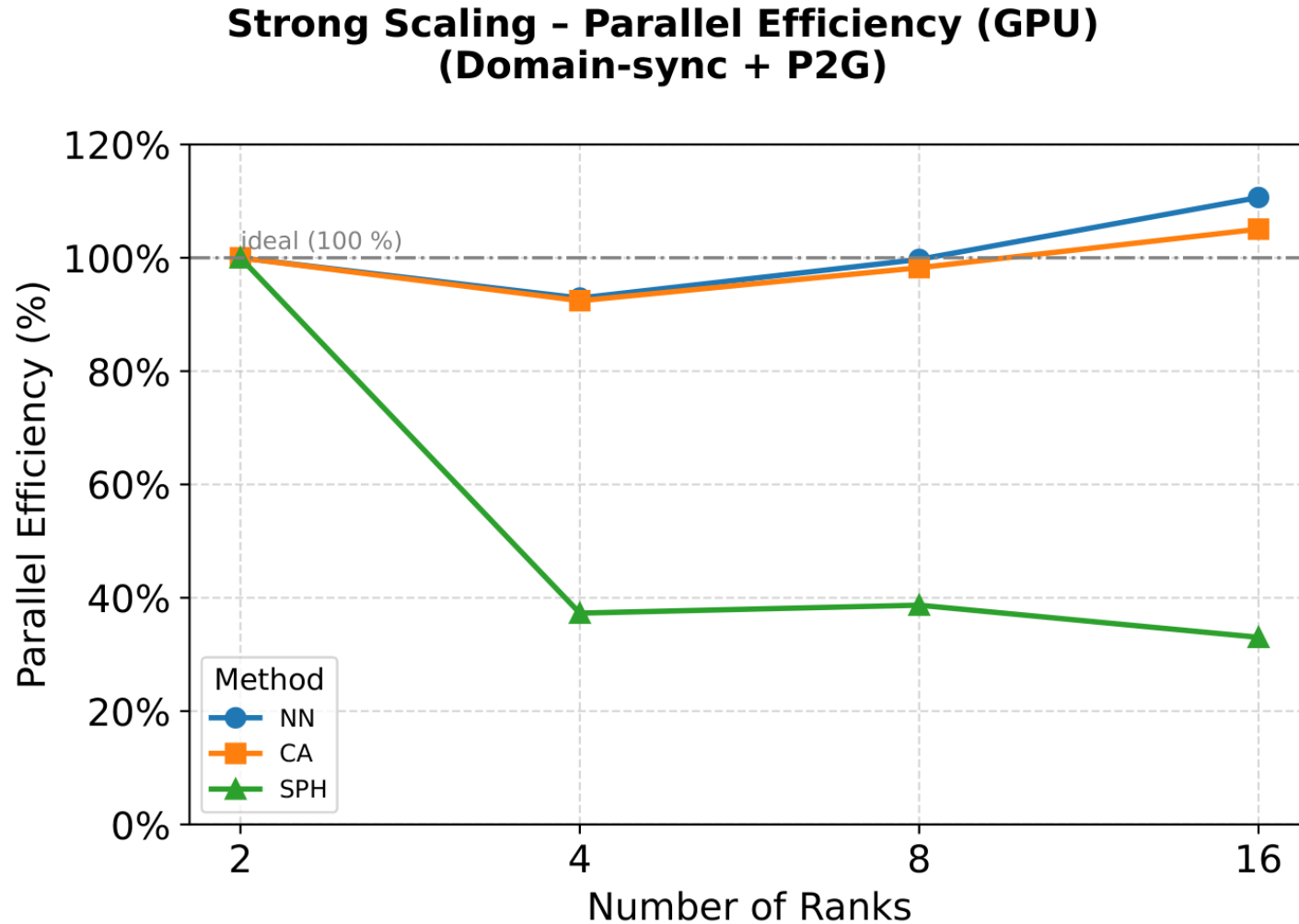
Factors	Values	Properties
Datasets	400 ³ 800 ³ 1600 ³ 3200 ³ number of SPH particles	SPH-EXA turbulence subsonic simulation outputs at t=10.0s physical time.
Interpolation methods	Nearest-neighbor Cell-average SPH	Three P2G schemes with increasing smoothness and computational cost.
Computing Nodes	Daint,Alps partition @ CSCS	NVIDIA Grace-Hopper Superchip 4 sockets per node 1xCPU with 72 ARM cores with 128 GB LPDDR 5X RAM + 1x H100 GPU with 96 GB HBM3 memory.
Strong Scaling	1 B particles on 2 4 8 16 nodes	1 MPI rank associated with 1 GPU device
Weak Scaling	125 M particles per MPI rank on 1 2 4 8 nodes	1 MPI rank associated with 1 GPU device
Metrics	$T_{PAR} = \max (T_i), i = 0, MPI_ranks-1$	Parallel performance
	Max P2G time	Particle-to-grid kernel time on the GPU
	Communication overhead	MPI communication time

Results – Strong Scaling: Speedup

**Strong Scaling - Speedup (relative to 2 ranks)
(Domain-sync + P2G)**

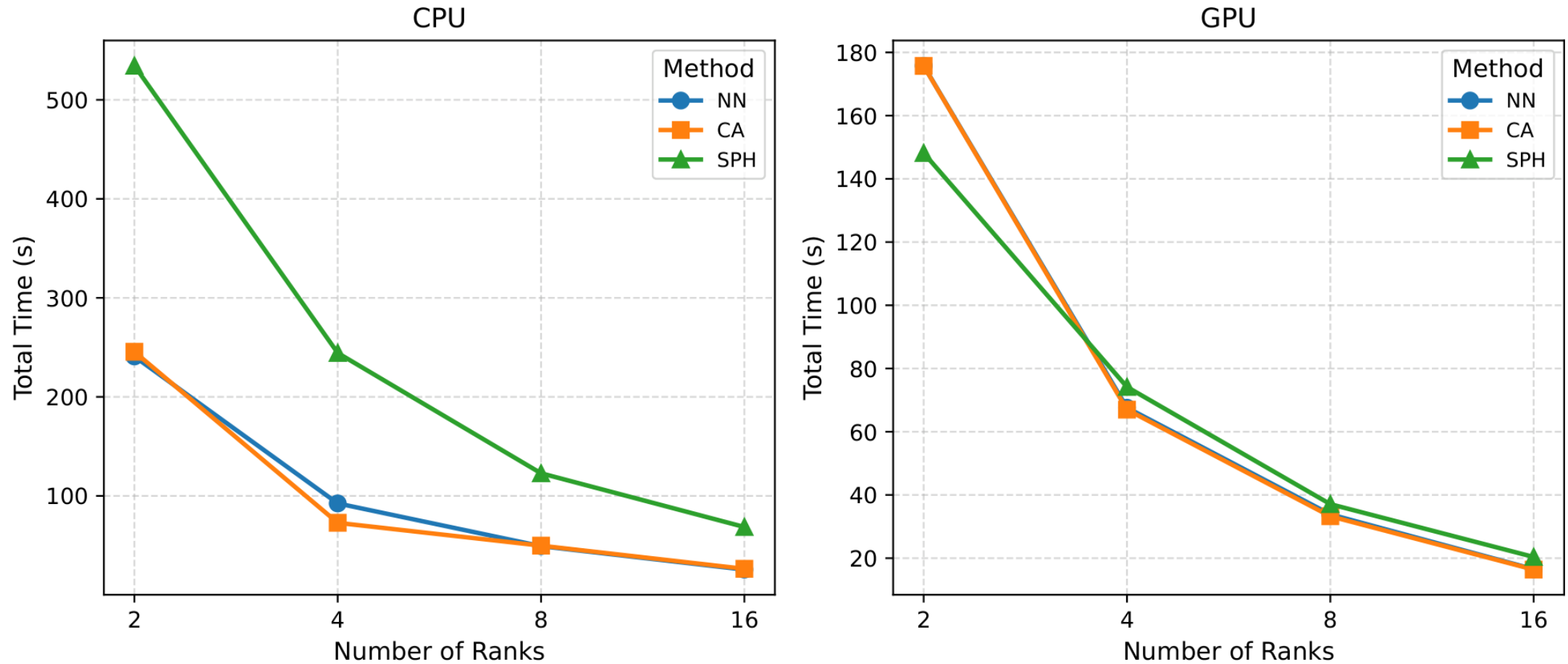


Results – Strong Scaling: Parallel Efficiency



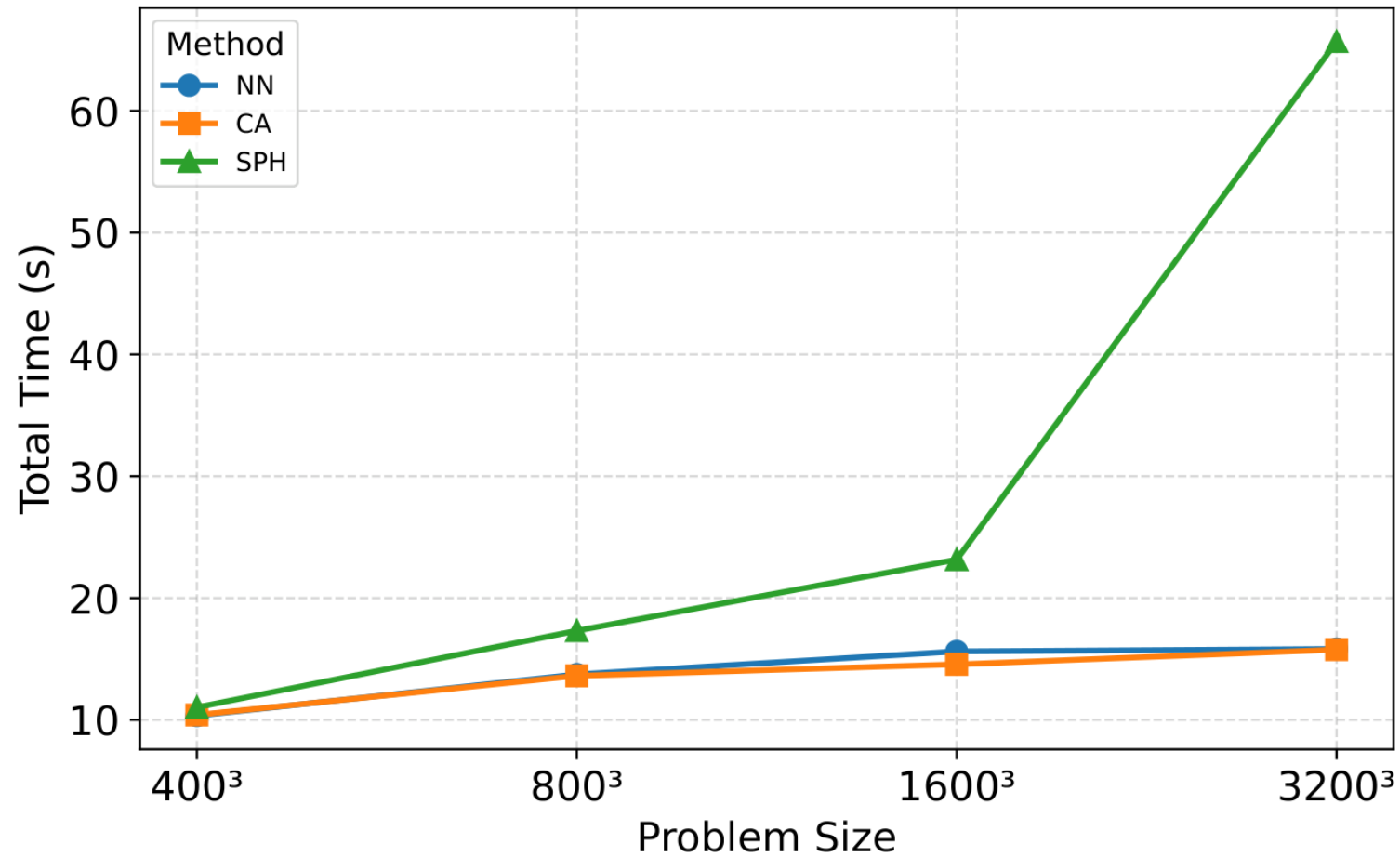
Results – Interpolation Methods Comparison

Strong Scaling - Total Time per Method (Domain-sync + P2G + Power Spectrum)

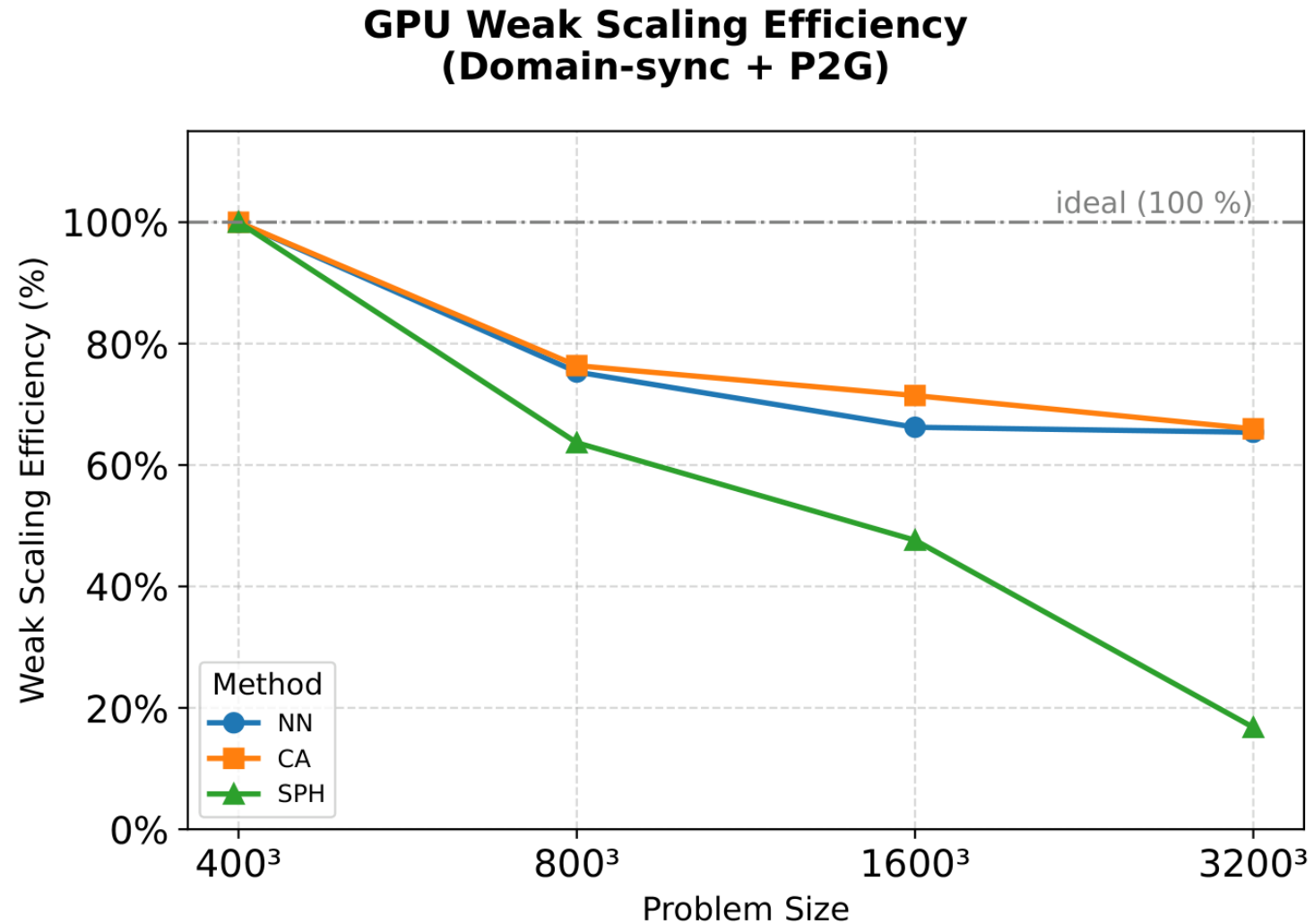


Results – Weak Scaling: Execution Time

**Weak Scaling - Total Time per Method (GPU)
(Domain-sync + P2G)**

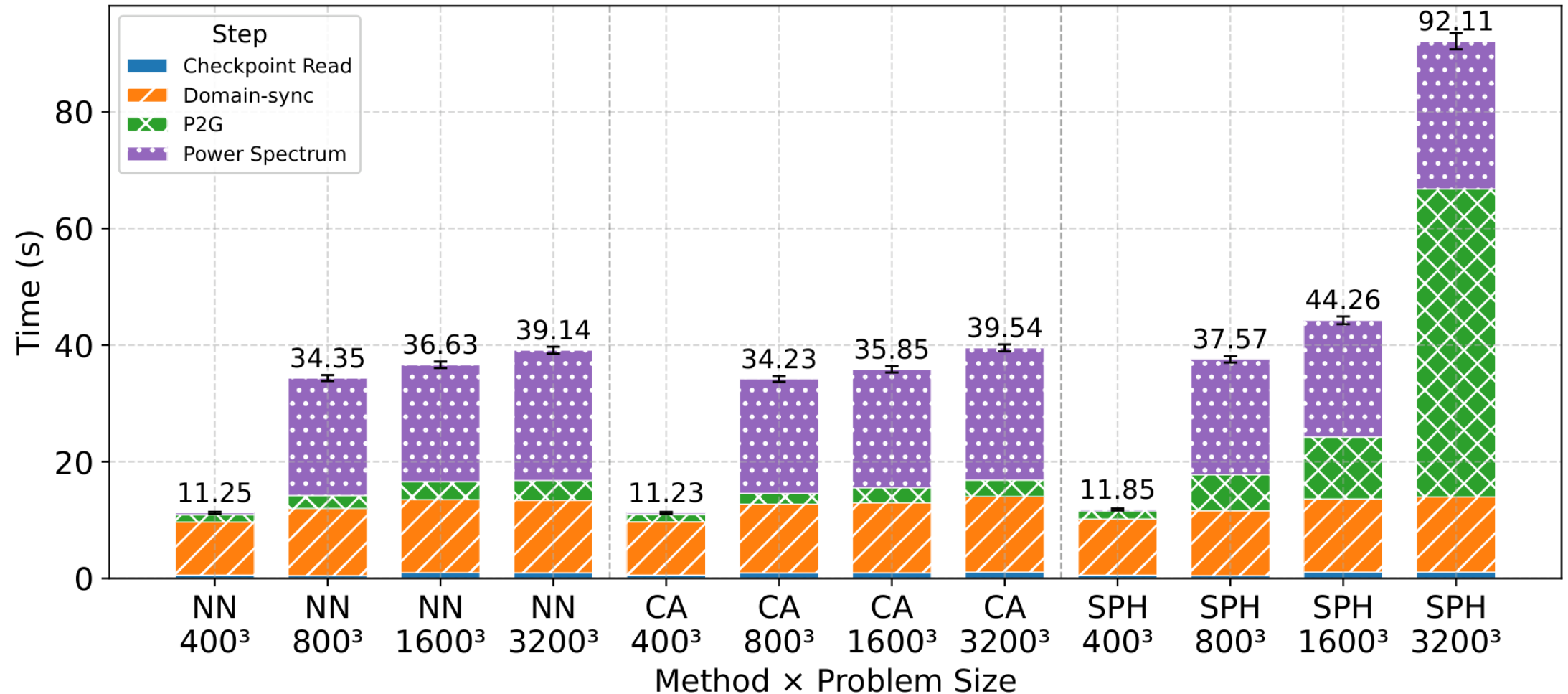


Results – Weak Scaling: Efficiency



Results – End-to-end Timing Breakdown

Total Time Breakdown per Method & Problem Size (GPU)

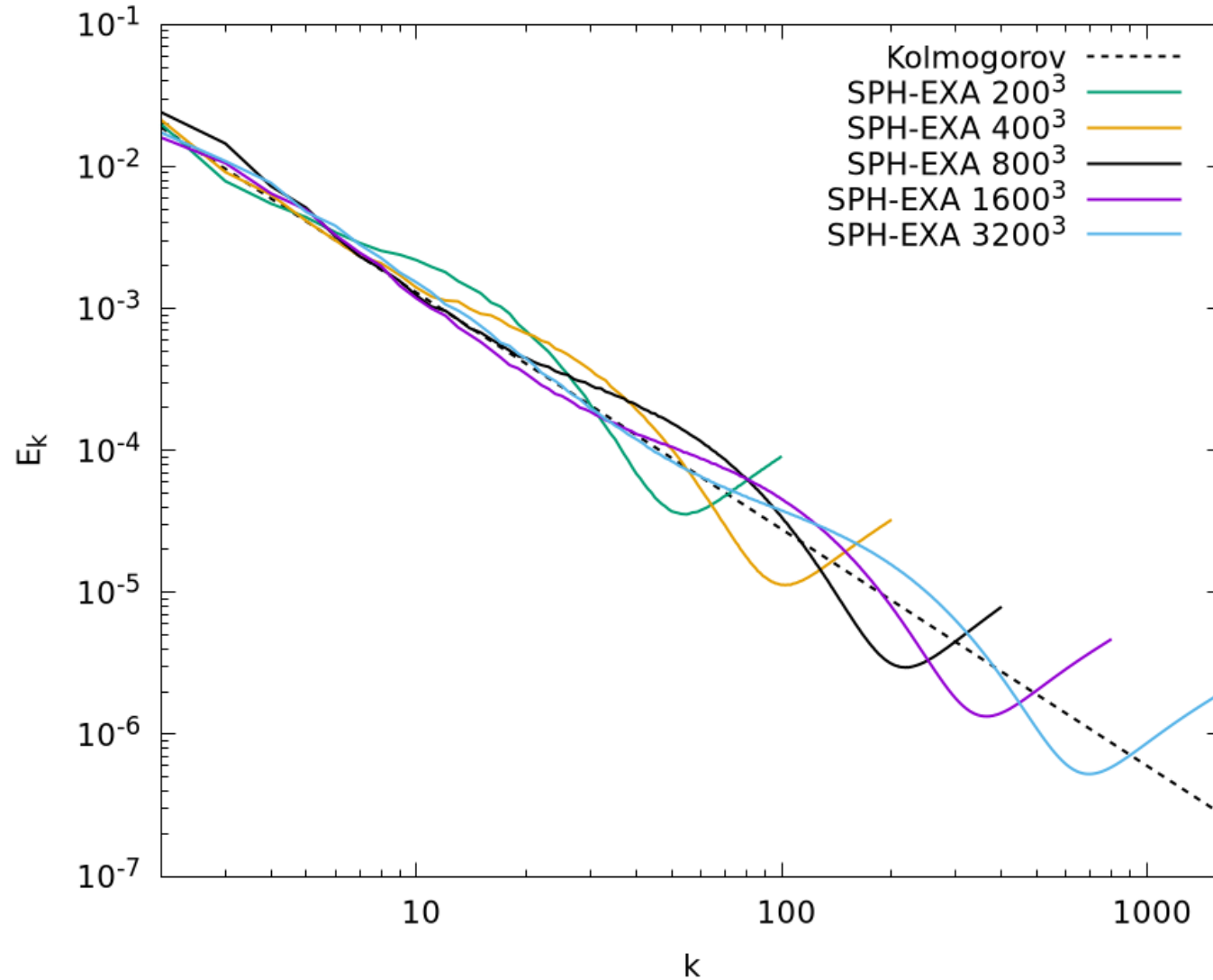


Conclusion

- GPU-resident, MPI-parallel P2G framework for extreme-scale cosmological simulations
- Transforms SFC particle decomposition into z-slab Cartesian grids for FFT & RT workflows
- Three interpolation schemes: nearest-neighbor, cell-average, SPH trade-offs between cost and smoothness
- NN and CA: excellent scaling up to 128 GPUs; SPH practical but more expensive
- In-situ deployment for SPH-EXA reduces the overhead of re-decomposition into SFC to few-second transformation
- Standalone library reusable for power spectrum estimation and radiative transfer pipelines

Backup

Power Spectrum of Subsonic Turbulence Using Cell_average Interpolation



Target Workflow – Radiative Transfer (RT) calculation (future work)

