

Distributed Large-Scale Traffic Simulation for Urban and Regional Mobility Analysis

Paulo Silva, Pavlína Smolková, Radek Fumánek,
João Barbosa, Matej Špet'ko, Emanuele Vitali,
Kateřina Slaninová, Jan Martinovič



Why simulate traffic?

- What if demand changes?
- What if a lane closes?
- What if drivers reroute?



Example visualization output from Ruth Traffic Simulator

Why we need HPC

Within a simulation

Routing and rerouting on a regional network

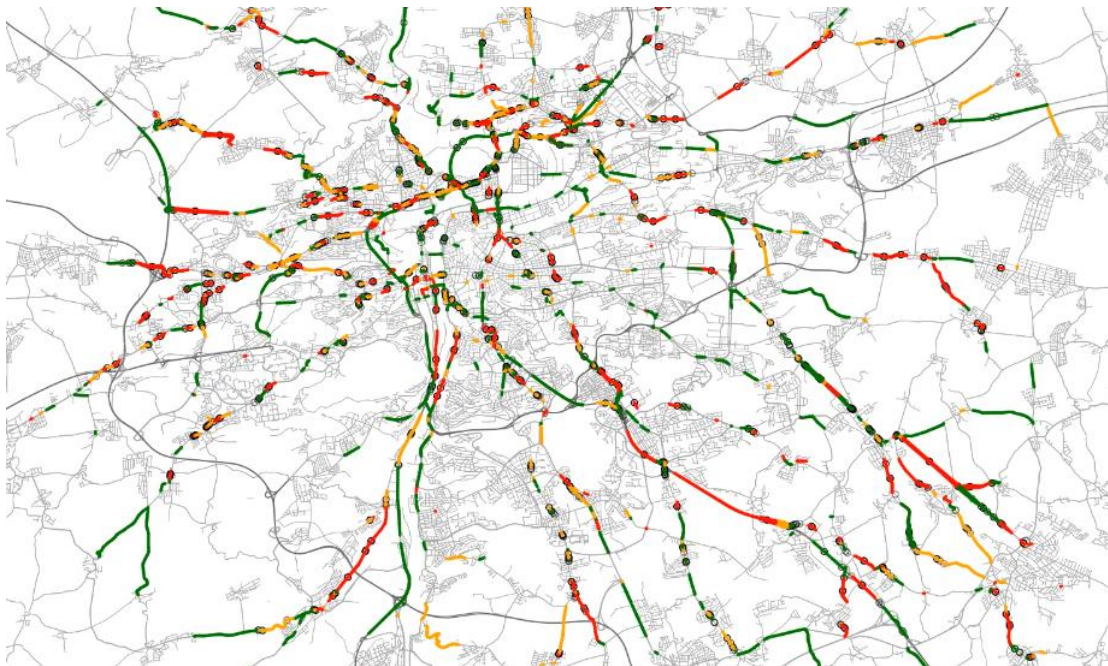
Distribute route requests across workers.

Across a study

Run different configurations. Process and compare the outputs.

Computing time and storage for the whole study.

What Ruth actually simulates



Mesoscopic simulation · Ruth's built-in 2D viewer

- **Individual vehicles**

Inputs: road network and travel demand.

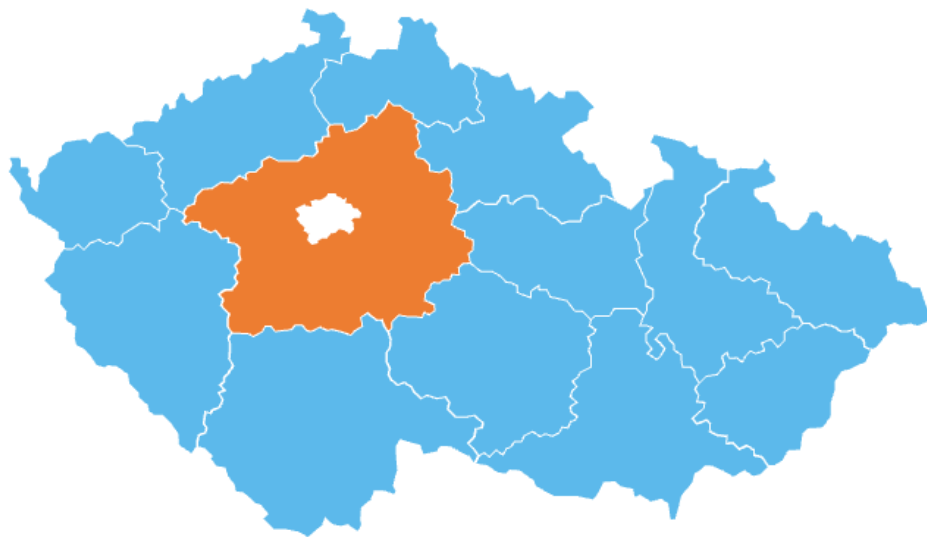
- **Congestion-dependent speeds**

Density changes the speed on each road segment.

- **Rerouting during a run**

Routes use updated network conditions.

The Central Bohemia Region use case



Central Bohemian Region (CBR), Czechia

Simulation network: 194,585 nodes · 478,517 segments

936,000

Vehicles in the largest
configuration

~50 GB

Floating Car Data (FCD):
Vehicle movements and positions
Interactions
Routing Decisions

How we use EuroHPC

Local development and runs

Prepare scenarios and run small cases.

Direct batch jobs

Run configurations and campaigns on the HPC systems.

LEXIS front end

Configure and manage simulations.

Available at: ruth.lexis.tech



RUTH

Main

- Dashboard
- Simulate
- Simulations List**

Home > Simulations List

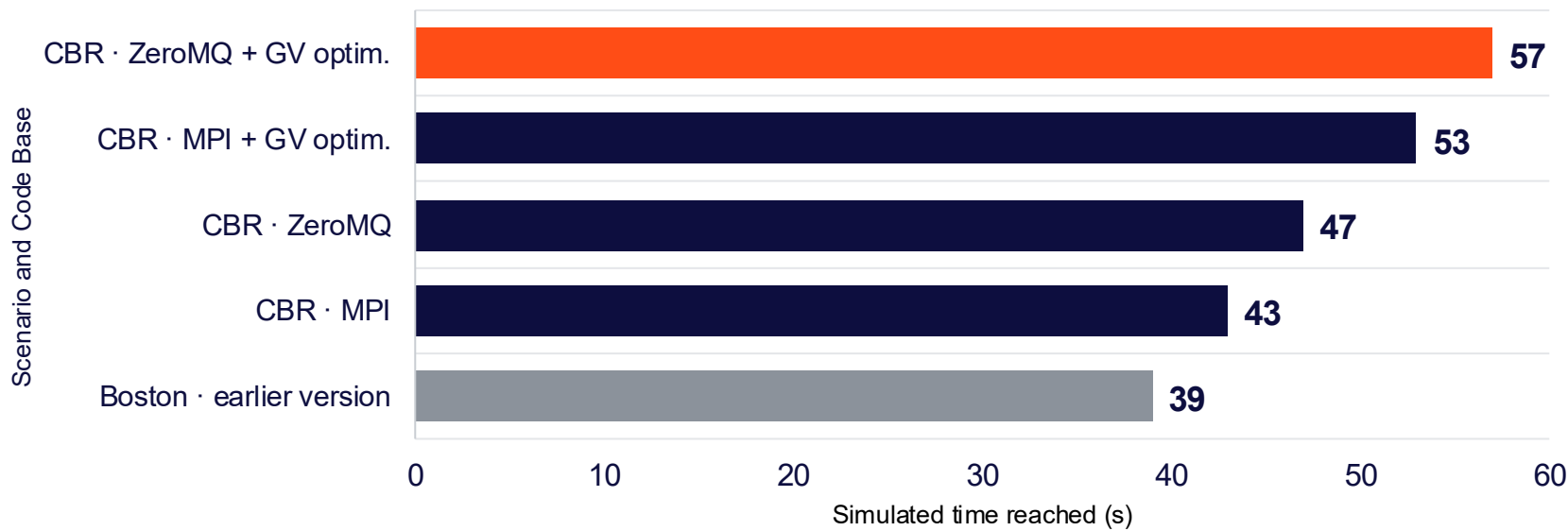
Refresh

	SIMULATION NAME	STATE	START DATE
⌵	Traffic-64ab821d-c814-4e8f-84c4-18da34f97f4a	success	16th of D
⌵	Traffic-simulation-906ffa83-283e-4fc9-b208-b24bb1781849	success	31st of O

Karolina CPU · reported results
LUMI-C · campaigns and workflow checks

Simulation progress in the same time budget

Simulated time reached (s) · ~22-minute budget · 2 Karolina CPU nodes

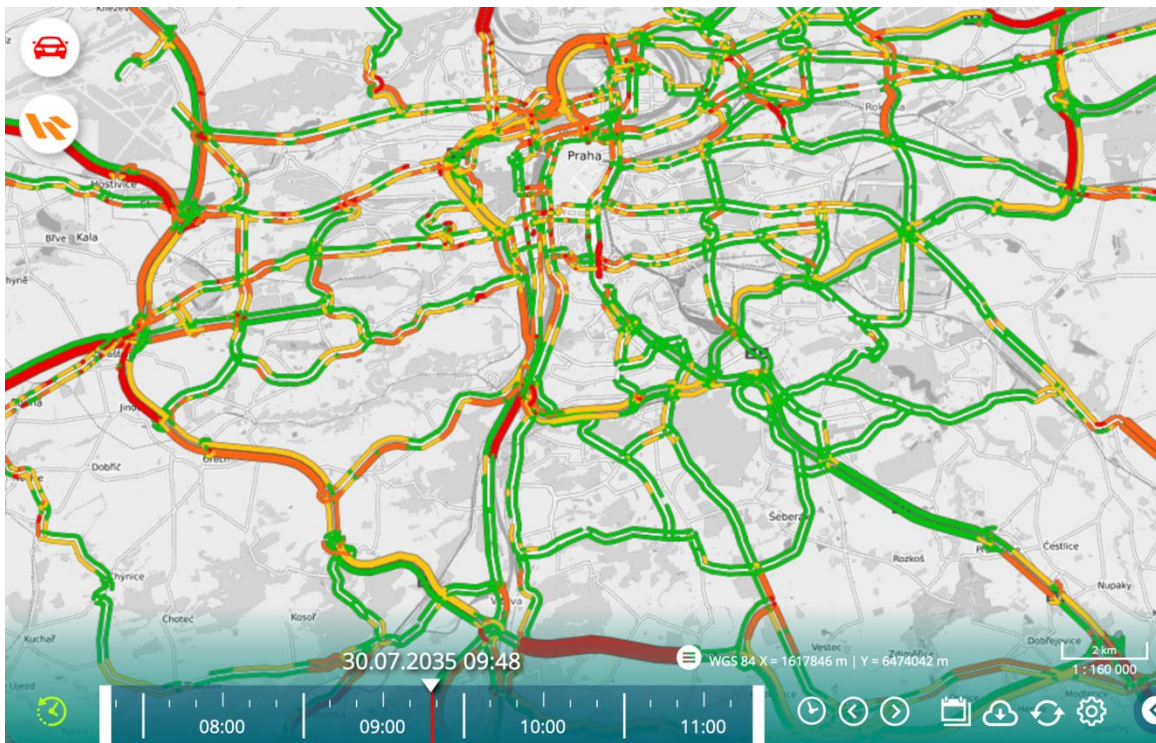


GV optim. = GlobalView optimisation · End-to-end comparison of different workloads

Inspecting simulations in FLOREON+



- Where does congestion form?
- When does it appear?
- What changed between runs?



Next steps for Ruth

- **Validate against observations**

Counts, travel times and congestion patterns.

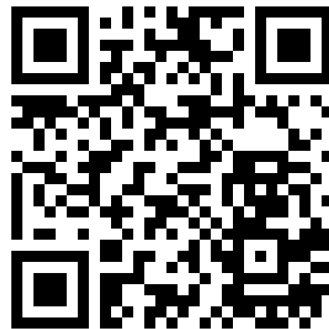
- **Make models easier to exchange**

Clear interfaces for demand, traffic flow, routing and output.

- **Extend to nationwide studies**

Carry the scenario and analysis workflow to larger networks.

RUTH



Thank you!

Questions?

Paulo Silva, Pavlína Smolková, Radek Fumánek,
João Barbosa, Matej Špet'ko, Emanuele Vitali,
Kateřina Slaninová, Jan Martinovič

paulo.silva@vsb.cz





EuroHPC
User Days

Dublin Ireland
22→24
September 2026



LBFAST

A Lightweight Moment-Represented Lattice Boltzmann Solver for Multi-GPU Architectures

M. Lauricella, A. Montessori, G. Amati, F. Spiga, A. Tiribocchi,
M. Bernaschi, S. Succi

Acknowledgments



Funded by
the European Union



European Research Council
Established by the European Commission



ERC PoC LBFAST: Grant No. 101187935 — LBFAST



Ministero
dell'Università
e della Ricerca



PRIN: Grant No. 2022N4ZNH3 — MOBIO5

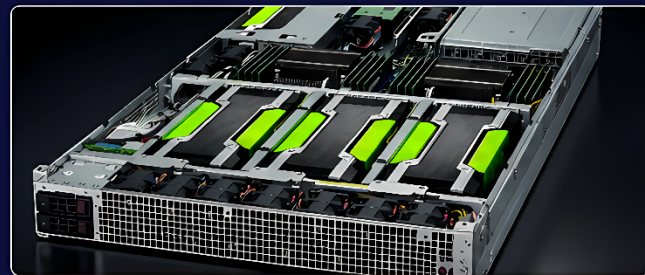
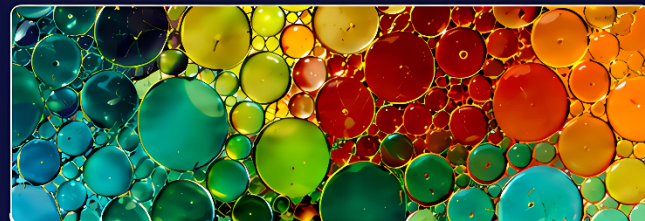
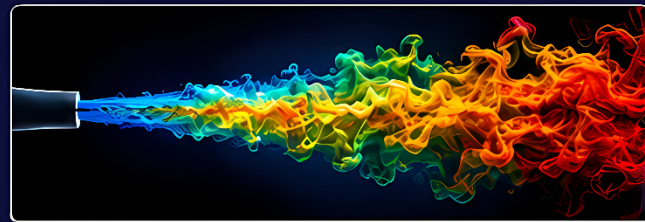


Motivation and Context

- **Single-component CFD:** turbulence, complex flows, and large 3D simulations still require high spatial resolution and long time integration.
- **Multicomponent / multiphase flows:** droplets, bubbles, emulsions, interface dynamics, breakup and coalescence, porous and process flows.
- **HPC motivation:** modern GPU CFD is often limited by memory capacity and bandwidth; VRAM is a critical and expensive resource.
- **Need for lightweight methods:** reduce stored variables and global-memory traffic to enable larger domains, faster runs, and more affordable simulations.

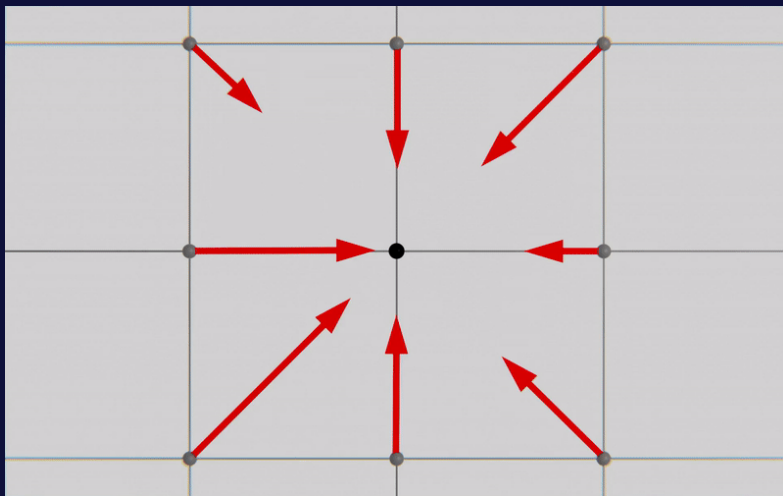


Goal: accurate CFD for mono- and multi-component flows with lower VRAM footprint.



Lattice Boltzmann

$$f_i(x_\alpha + c_{i\alpha}\Delta t, t + \Delta t) = f_i(x_\alpha, t) - \omega(f_i - f_i^{eq}) + \left(1 - \frac{\omega}{2}\right) S_i$$



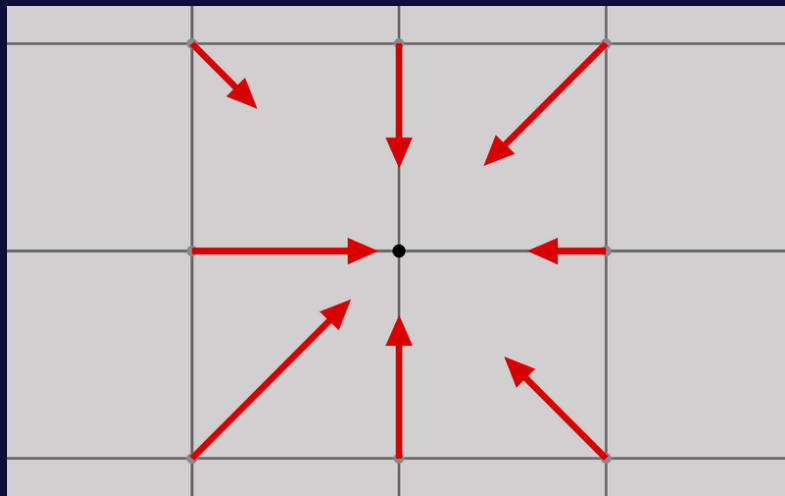
$$a_{eq}^0 = p^*(x_\alpha, t) = \sum_i f_i(x_\alpha, t)$$

$$a_{eq,\alpha}^1 = u_\alpha(x_\alpha, t) = \sum_i f_i(x_\alpha, t) c_{i\alpha} + \frac{1}{2\rho} F_\alpha$$

$$\begin{aligned} a_{neq,\alpha\beta}^2 &= \sum_i (c_{i\alpha} c_{i\beta} - c_s^2 \delta_{\alpha\beta}) (f_i - \overline{f_i^{eq}}) = \\ &= -\frac{c_s^2}{\omega} (\partial_\beta u_\alpha + \partial_\alpha u_\beta) + \frac{1}{2\rho} (F_\alpha u_\beta + F_\beta u_\alpha) \end{aligned}$$

Lightweight LB

$$f_i(x_\alpha, t + \Delta t) = f_i^{eq}(x_\alpha - c_{i\alpha}\Delta t, t) + (1 - \omega)f_i^{neq}(x_\alpha - c_{i\alpha}\Delta t, t) + \frac{1}{2}S_i(x_\alpha - c_{i\alpha}\Delta t, t)$$



$$H_i^{(0)} = 1, \quad H_{i\alpha}^{(1)} = c_{i\alpha}, \quad H_{i\alpha\beta}^{(2)} = (c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta})$$

$$f_i^{eq} = w_i \sum_n \frac{1}{c_s^{2n} n!} \mathcal{H}_{i\alpha_1 \dots \alpha_n}^{(n)} a_{eq, \alpha_1 \dots \alpha_n}^{(n)}$$

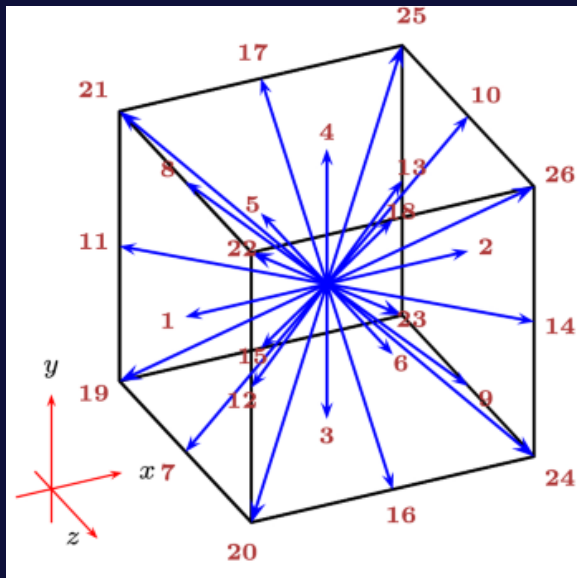
$$f_i^{neq} = w_i \sum_n \frac{1}{c_s^{2n} n!} \mathcal{H}_{i\alpha_1 \dots \alpha_n}^{(n)} a_{neq, \alpha_1 \dots \alpha_n}^{(n)}$$

$$a_{eq, \alpha_1, \dots, \alpha_n}^{(n)} = \sum_i \mathcal{H}_{i\alpha_1, \dots, \alpha_n}^{(n)} f_i^{eq}$$

$$a_{neq, \alpha_1, \dots, \alpha_n}^{(n)} = \sum_i \mathcal{H}_{i\alpha_1, \dots, \alpha_n}^{(n)} f_i^{neq}$$

Lightweight LB

Standard LB stores populations and derives moments; Lightweight LB stores moments and reconstructs populations only when needed.



VRAM footprint for 512^3 in FP64 with D3Q27

Standard LB - A/B populations + 4 moments

$2 \times 27 + 4 = 58$ arrays

58 GiB

Lightweight LB - A/B 10 moments ($p^*, u_\alpha, a_{neq, \alpha\beta}^2$)

$2 \times 10 = 20$ arrays

20 GiB

65.5% lower VRAM footprint

Minimum Hermite representation for LB:

2nd order Hermite expansion

$$H_i^{(0)} = 1, \quad H_{i\alpha}^{(1)} = c_{i\alpha}, \quad H_{i\alpha\beta}^{(2)} = (c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta})$$

is sufficient to recover the weakly compressible Navier–Stokes equations

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) &= -\nabla p + \nabla \cdot \left(\rho \nu \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right] \right) + \mathbf{F} \end{aligned}$$

$$\mathbf{F} = \mathbf{F}^g + \mathbf{F}^s + \dots$$

$$\mathbf{F}^s = \nabla \cdot [\sigma(\mathbf{I} - \mathbf{n} \otimes \mathbf{n})]$$

Proceedings of the fourth EuroHPC User Days 2026

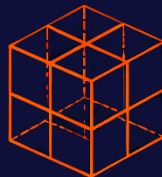
LBFAST: A Lightweight Moment-Represented Lattice Boltzmann
Solver for Multi-GPU Architectures

LBFAST at glance



GPU IMPLEMENTATION

- Fortran + CUDA
- Single- and multi-GPU



MPI PARALLELIZATION

- 1D / 2D / 3D domain decomposition
- GPU-aware communication
- Collective MPI-IO



FP64 / FP32

PRECISION MODES

- FP64 (double precision)
- FP32 (single precision)
- Mixed (FP32 storage / FP64 computation)

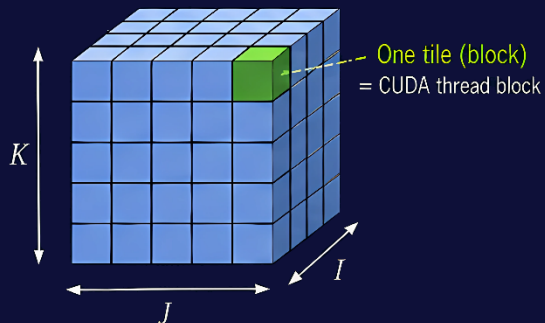


VELOCITY SETS & RECONSTRUCTION

- D3Q19
- D3Q27
- 2nd-order (minimal)
- High-order reconstruction (up to 27 moments with D3Q27)

Data Layout

1) Block-structured in 3D thread-block tiles



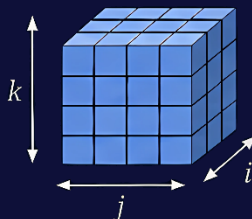
- 3D tiles match CUDA thread blocks
- Block-structured storage for coalesced access
- Each tile contains $\text{TILE_DIM}_x \times \text{TILE_DIM}_y \times \text{TILE_DIM}_z$ lattice nodes

2) Each field array is a 5D array

```
hfields(i, j, k, field, block)
```

or

```
(TILE_DIMx, TILE_DIMy, TILE_DIMz, nhfields, nblocks)
```



Local coordinates
inside a tile

field = 0 ... *nhfields* - 1
(physical fields)

block = 0 ... *nblocks* - 1
(tiles covering the domain)

Outer dimensions (*field*, *block*) select which field
and which tile; *i*, *j*, *k* index within the tile.

3) Double buffering with flip / flop (separate arrays, same layout)

Input / read (flip)

```
hfields_flip(i, j, k, field, block)
```



Swap roles
each step

Output / write (flop)

```
hfields_flop(i, j, k, field, block)
```

Flip and flop have identical 5D layout.
They are **TWO SEPARATE** arrays —
not a dimension of the array.



Key takeaways

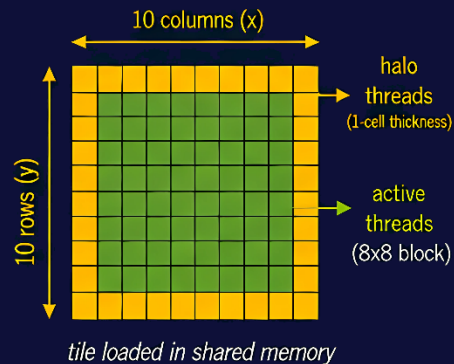
- ✓ 3D thread-block tiles on the GPU
- ✓ 5D arrays: (*i*, *j*, *k*, *field*, *block*)
- ✓ flip / flop = separate input/output arrays
- ✓ Layout maximizes locality and bandwidth

The layout is designed so that memory organization matches the thread-block geometry used by *fused_lb_kernel*.

fused_LB_kernel

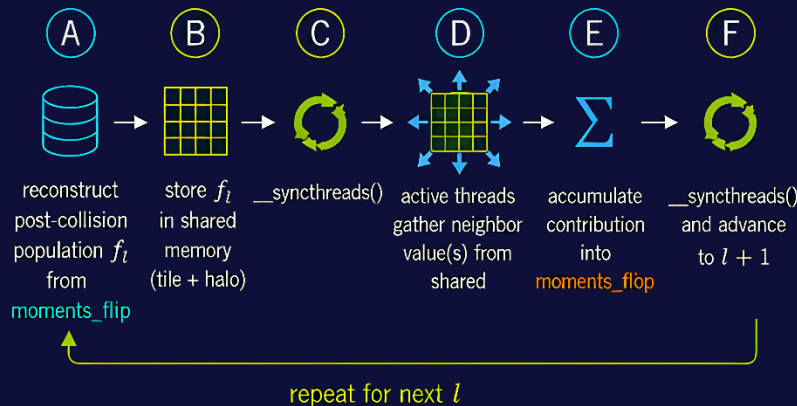
on-the-fly population reconstruction, shared-memory gather, and moment accumulation

1) Thread organization



- active block (8x8 threads)
- 1-cell halo
- ↔ gather directions (from neighbors)

2) Per-direction update (loop over $l = 1 \dots Q$)



one population
at a time

`moments_flip` += contribution from f_l

3) Memory strategy



Global memory (HBM)

- read `moments_flip`
- write `moments_flip` only at the end
- no full population arrays stored



Registers

- local accumulators:
 $p^*, u_\alpha, \sigma_{\alpha\alpha\beta, \alpha\beta}^{(eq)}$
- current reconstructed f_l

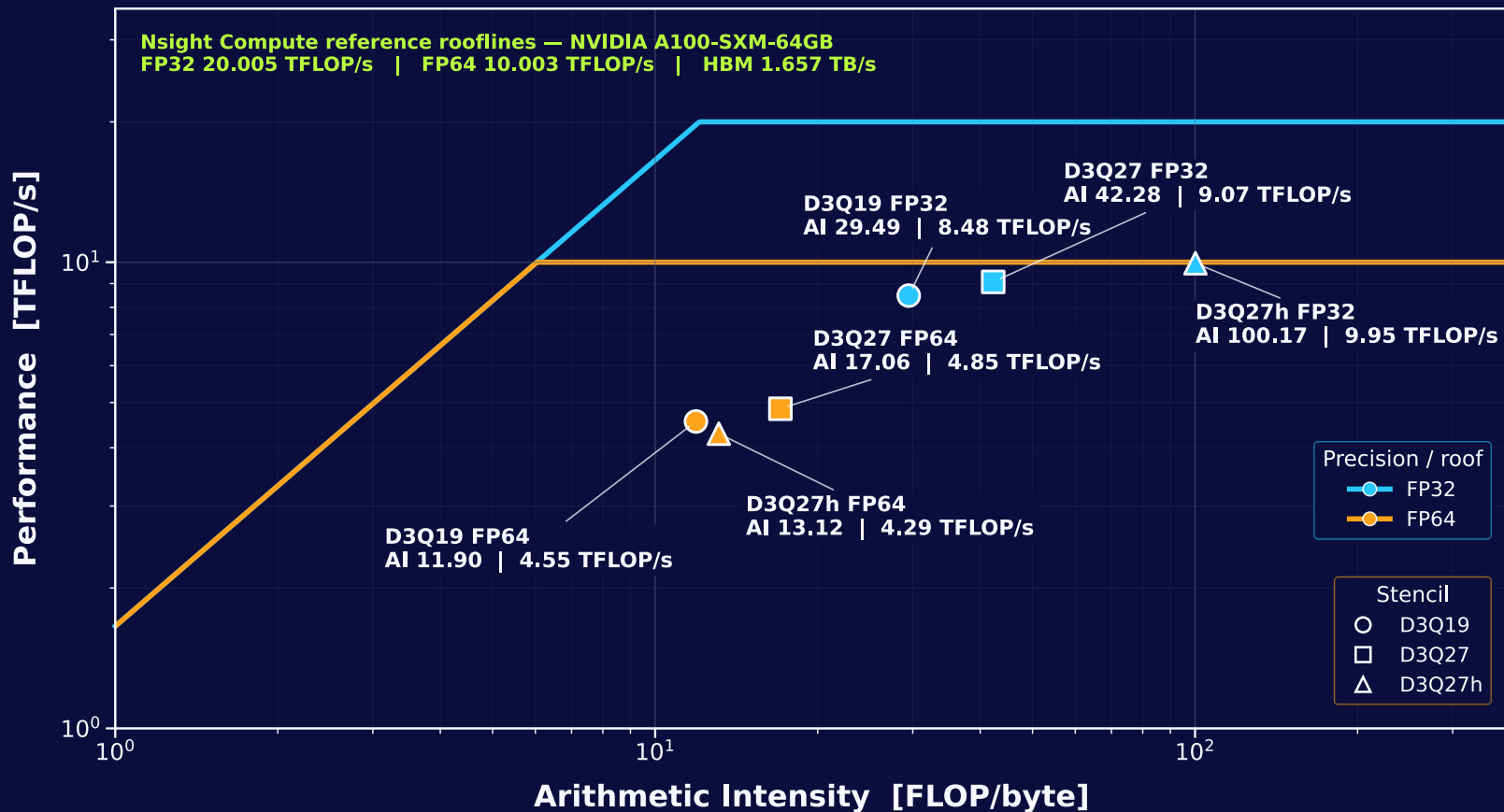


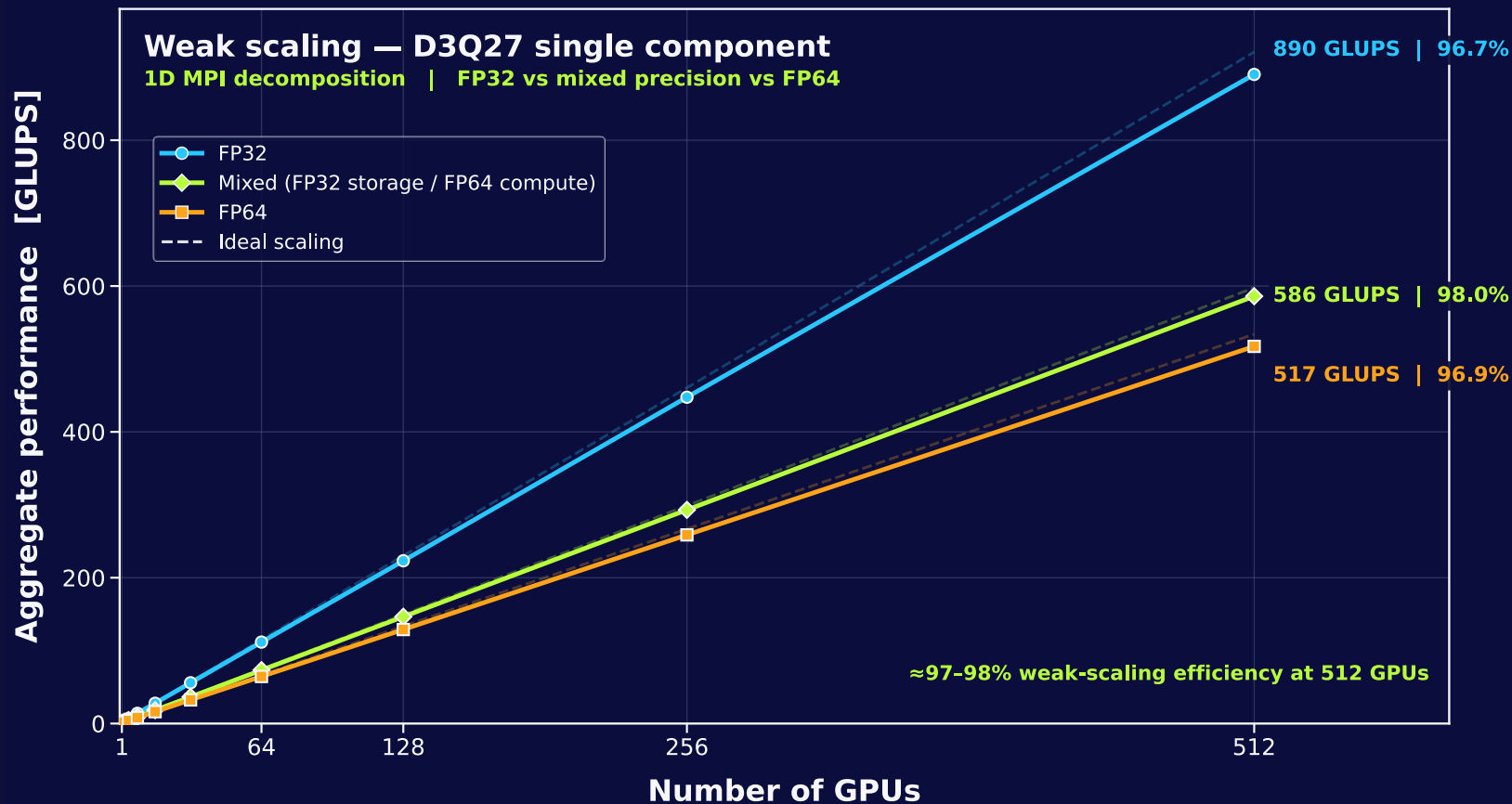
Shared memory

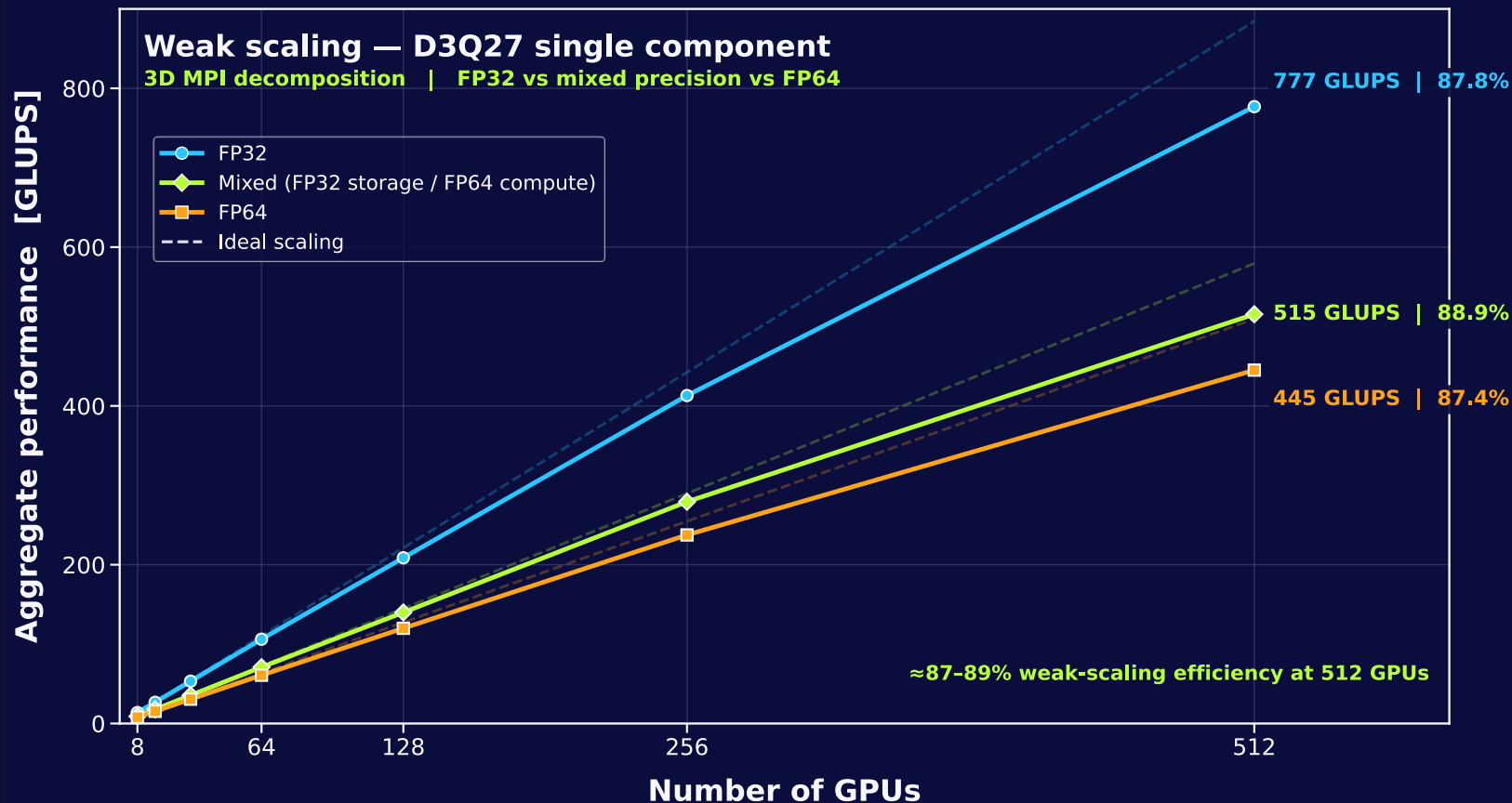
- temporary storage of one propagated population f_l on tile + halo
- reused at each direction



HBM stores `moments` only;
shared memory holds one population at a time.

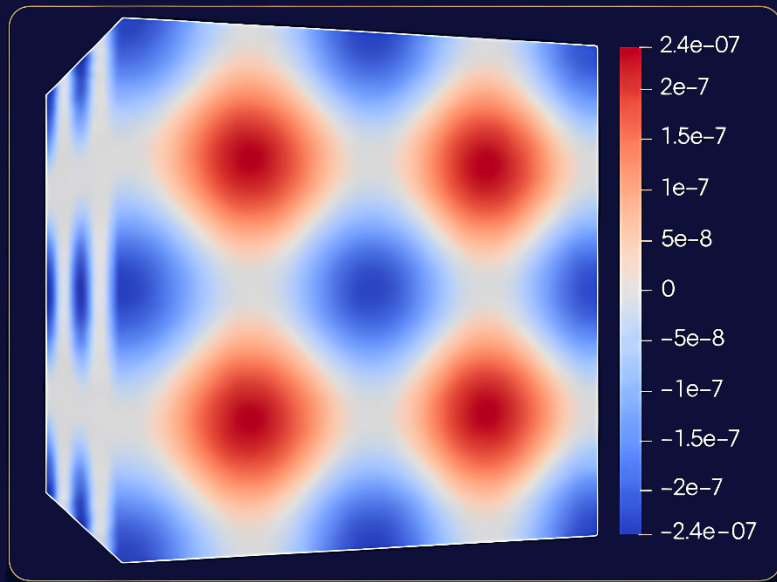






Validation cases

- One-component Taylor–Green vortex in a **512³** cubic box



GOVERNING SETUP

$$x = [2\pi(i-1)]/N, \quad y = [2\pi(j-1)]/N, \quad z = [2\pi(k-1)]/N.$$

$$u_x(x, y, z) = U_0 \sin(x) \cos(y) \cos(z),$$

$$u_y(x, y, z) = -U_0 \cos(x) \sin(y) \cos(z),$$

$$u_z(x, y, z) = 0,$$

$$p(x, y, z) = \frac{U_0^2}{16} [\cos(2x) + \cos(2y)][2 + \cos(2z)].$$



VALIDATION QUANTITIES

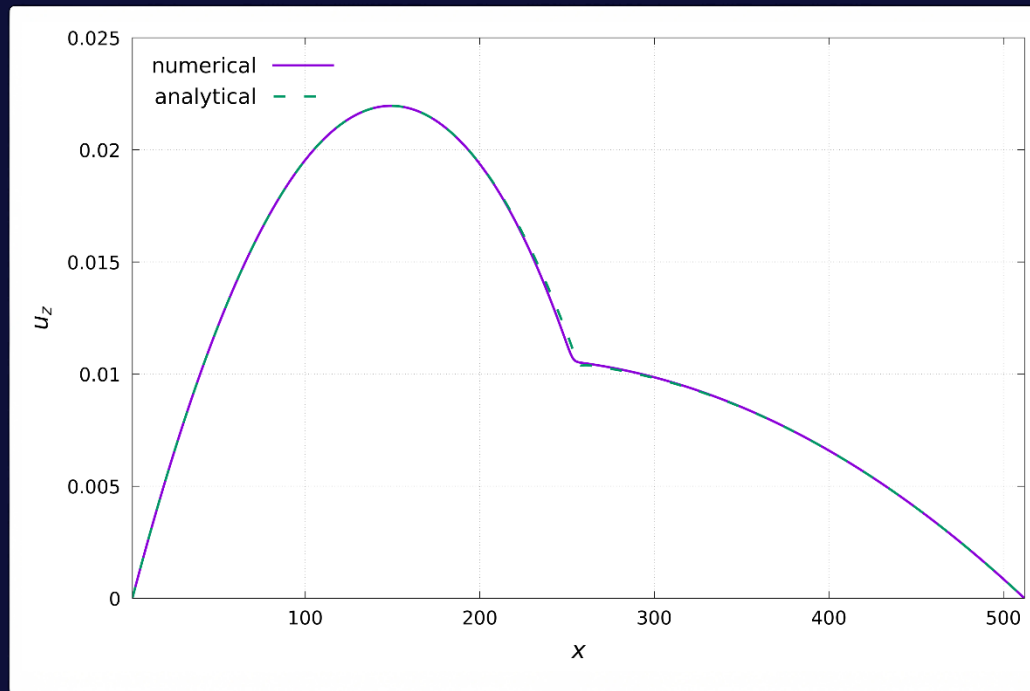
$$\frac{E(t)}{E(0)} = \exp(-6\nu k^2 t), \quad k = \frac{2\pi}{N}, \quad E(t) = \frac{\langle u_\alpha u_\alpha \rangle}{2}.$$

$$Q = 0.5 (\|\Omega\|^2 - \|S\|^2).$$



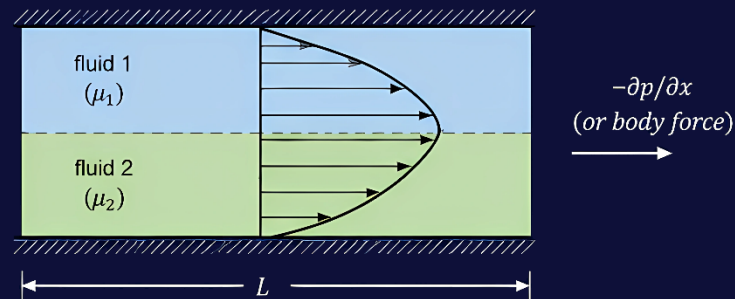
$$\text{Relative error} \sim 1.3 \times 10^{-3}$$

Validation cases



Two-layered Poiseuille flow

- Analytical benchmark for stratified laminar channel flow.
- Two superposed fluid layers with different viscosities.
- Steady velocity profile compared against the exact solution.
- Excellent agreement in the bulk; discrepancy localized at the diffuse interface.
- Largest relative error at the interface $\sim 4.18 \times 10^{-2}$.



**Validates viscosity coupling and
interface-consistent momentum transport.**

Two-component model

Conservative Allen–Cahn + force coupling

Phase-field transport

$$\partial_t \phi + u_\alpha \partial_\alpha \phi = D \partial_\alpha \partial_\alpha \phi + S_{AC}$$

$$S_{AC} = -\lambda \partial_\alpha [\phi(1-\phi)n_\alpha]$$

$$n_\alpha = \frac{\partial_\alpha \phi}{|\nabla \phi|}, \quad \lambda = \frac{4D}{W}$$

interface-compression source term counteracts numerical diffusion and preserves a sharp interface

Momentum equation and extra forcing

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = -\partial_\alpha p + \partial_\beta [\rho \nu (\partial_\beta u_\alpha + \partial_\alpha u_\beta)] + F_\alpha$$

$$F_\alpha = F_\alpha^s + F_\alpha^b + F_\alpha^p + F_\alpha^v$$

$$F_\alpha^s = \mu_\phi \partial_\alpha \phi$$

$$\mu_\phi = 4\beta \phi(\phi-1) \left(\phi - \frac{1}{2}\right) - \kappa \nabla^2 \phi$$

$$F_\alpha^p = -p^* c_s^2 \partial_\alpha \rho$$

$$F_\alpha^v = \nu \omega c_s^2 a_{neq,\alpha\beta}^{*(2)} \partial_\beta \rho$$

Property reconstruction

$$\rho(\phi) = \rho_L + \phi(\rho_H - \rho_L)$$

$$\nu(\phi) = \nu_L + \phi(\nu_H - \nu_L)$$

$$\beta = \frac{12\sigma}{W}, \quad \kappa = \frac{3\sigma W}{2}$$

Used in the LBFAST multiphase extension.

Summary and perspectives

Summary

- ✓ Moment-represented LBFAST enables a compact GPU-friendly formulation.
- ✓ Block-structured data layout and fused kernels reduce memory traffic.
- ✓ Shared-memory tiling handles non-local reconstruction efficiently.
- ✓ Roofline analysis shows compute-oriented behavior, especially for D3Q27 / D3Q27h.
- ✓ Strong weak-scaling behavior was demonstrated up to large GPU counts.

Perspectives

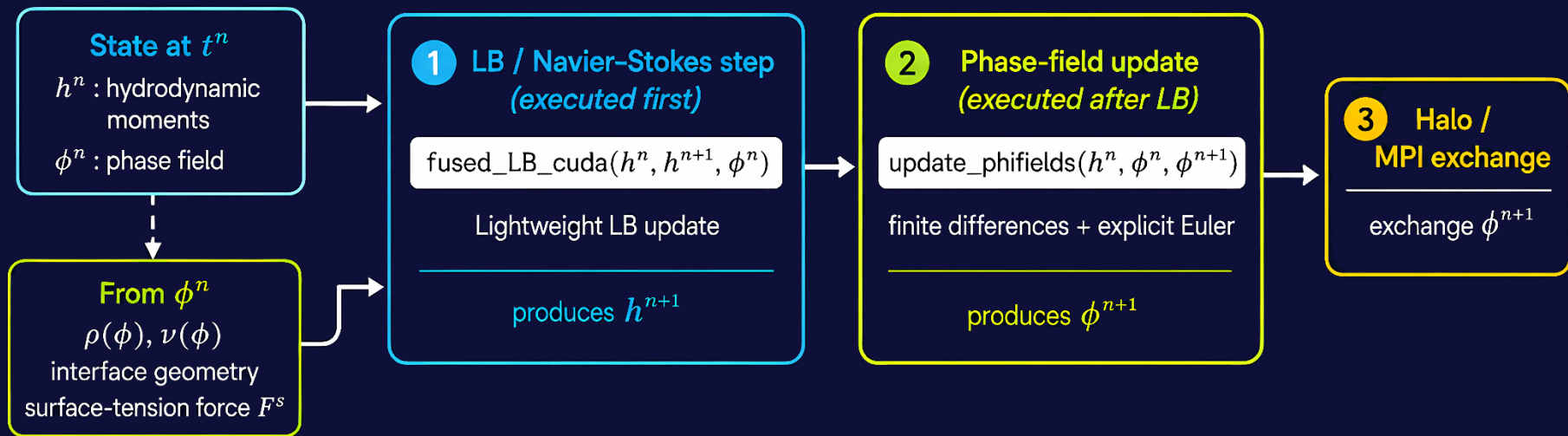
- ➔ Further reduce register pressure and improve occupancy.
- ➔ Integrate the phase-field evolution via moment-represented LB.
- ➔ Enlarge validation and production-scale benchmark campaigns.

Thank you for your attention

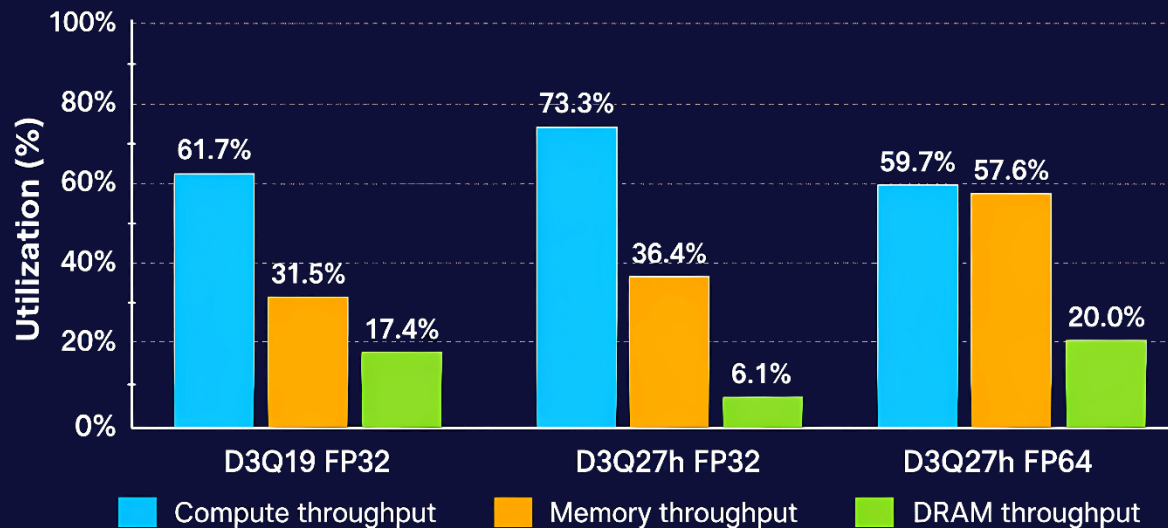
Questions?

Backup slides

Two-component implementation



Nsight compute – Resource utilization



- D3Q19 FP32 — compute already dominates over memory
- D3Q27h FP32 — 73% compute utilization, only 6% DRAM → strongly compute-oriented
- D3Q27h FP64 — compute ≈ memory, but DRAM only 20% → latency/cache-local traffic rather than HBM saturation

 64-register cap used to preserve occupancy; register spilling is being investigated for the high-order FP64 kernel.

Baseline discussion – which comparison is fairer?

Backup slide: representative A100 throughput from reported sources

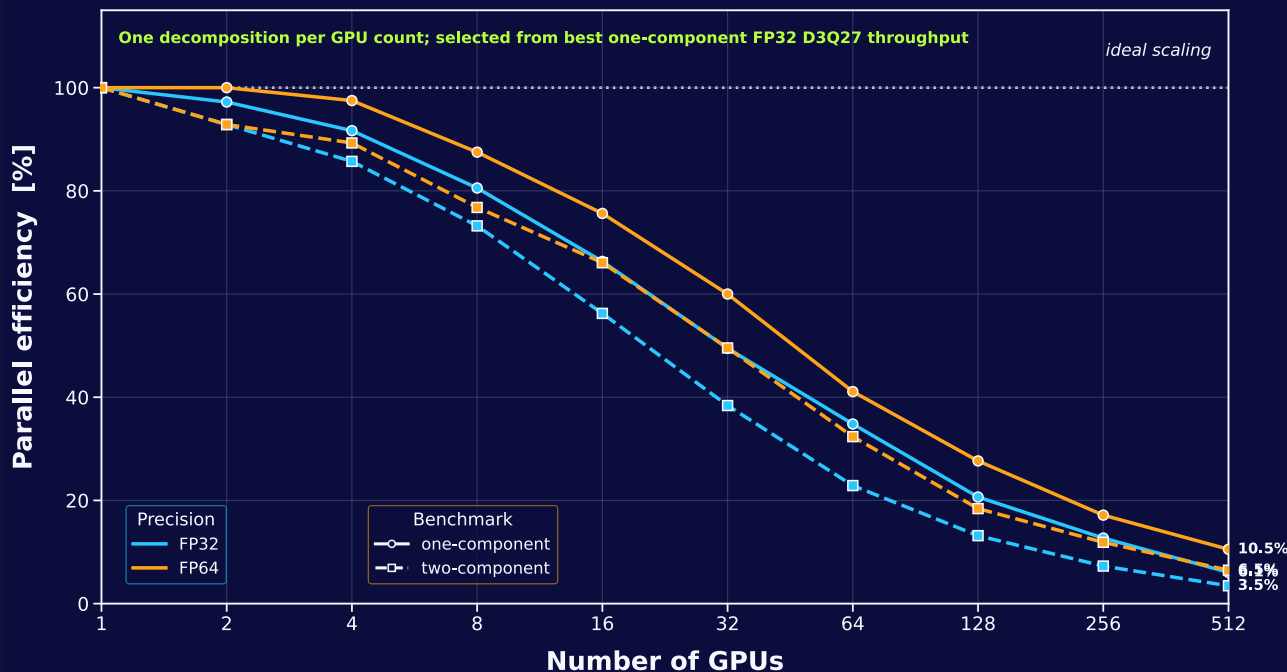
Code	Numerical model / storage	Representative A100 throughput (qualitative comparison)
FluidX3D (M. Lehmann et al.)	Population-based, highly optimized mixed-precision GPU LBM	Different model Not a fully fair one-to-one baseline (single-phase, different physics)
accLB (A. Montessori et al.)	Regularized LB + phase-field, population-based, OpenACC	Fairer baseline Same physical model and single-phase LB approach
LBFAST (this work)	Moment-represented LB, reconstructed populations, CUDA Fortran	Qualitative comparison target Same physical model and single-phase LB approach

Take-home message

- ◆ FluidX3D uses a different physical/modeling approach and is not a fully fair one-to-one baseline.
- ◆ accLB is the closer algorithmic baseline for LBFAST.
- ◆ Direct throughput rankings alone are not enough: stencil, collision model, memory footprint, and scalability also matter.

Strong scaling

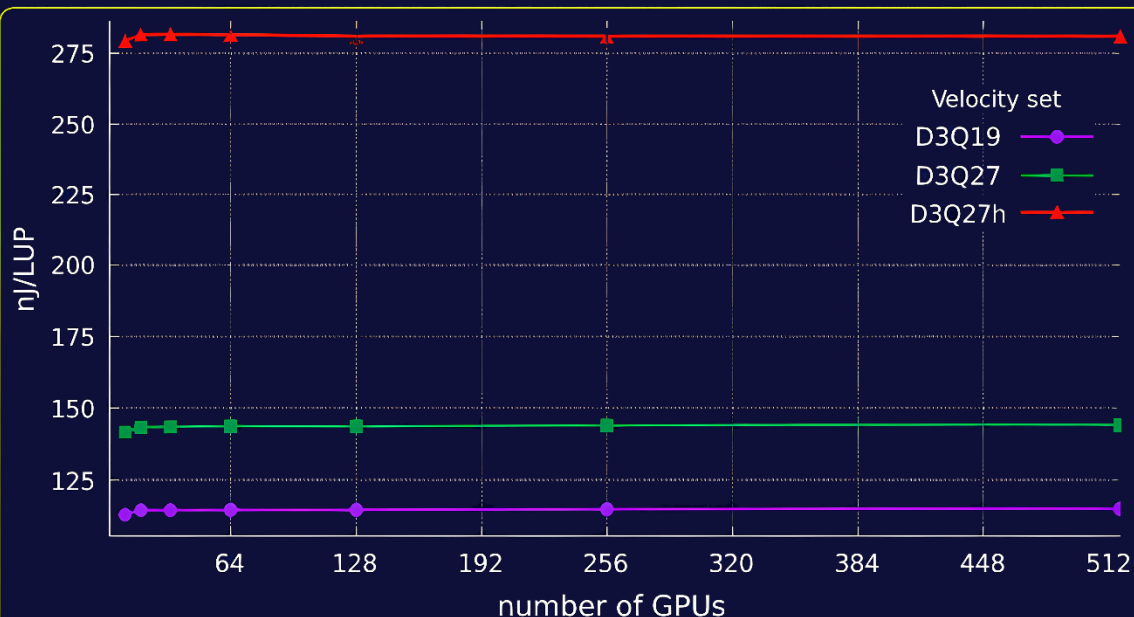
Strong scaling — D3Q27 (fixed global domain: 512^3)



- Efficiency decreases monotonically as communication becomes dominant at larger GPU counts.
- One-component runs retain the best efficiency; two-component cases scale slightly worse due to the heavier workload and communication overhead.
- FP64 remains competitive and, for this D3Q27 dataset, shows slightly higher efficiency than FP32 at large GPU counts.

Energy efficiency in weak scaling

Stable energy per lattice update across GPU count



- Measurement method: NVIDIA Management Library (NVML)
- Total energy was obtained from the difference of the cumulative counter returned by "nvmlDeviceGetTotalEnergyConsumption" between the beginning and the end of each run.
- Definition: " $\eta/LUP = E_{tot} / N_{LUP}$ "
- With " $N_{LUP} = L_x \cdot L_y \cdot L_z \cdot N_t$ "
- The energy per lattice update remains approximately constant up to 512 GPUs.
- D3Q19 is the most energy-efficient velocity set.
- D3Q27h has the highest energy cost per update because of the larger reconstruction workload.



Good weak scaling is accompanied by a stable energy footprint.



UNIVERSITAT DE
BARCELONA



Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

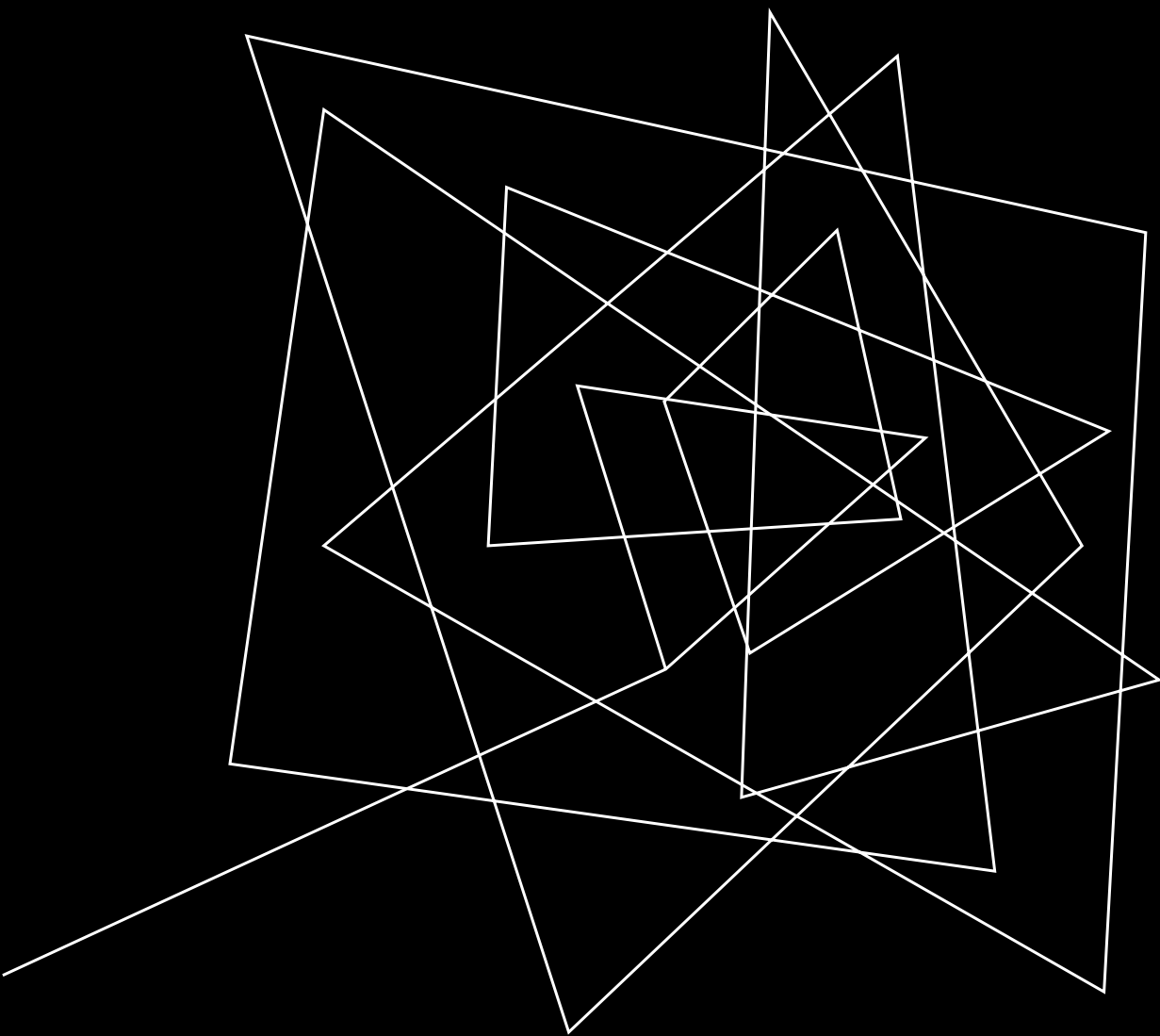
TOWARDS
SCALABLE VISUAL FOOD
UNDERSTANDING
THROUGH
DATA-CENTRIC DEEP
LEARNING



Petia **RADEVA**
University of Barcelona



EuroHPC User Days
Dublin Ireland



ABOUT US!!!

UNIVERSITY OF BARCELONA

A research-intensive university at the heart of
Barcelona

Founded **1450**

Over 575 years of academic history

60,000+ Students

€196M

Research funding secured

6,479 Teaching &
research staff



7,768 Scientific
publications

#1 in Spain #68 Europe
#165 worldwide



ARTIFICIAL INTELLIGENCE AND BIOMEDICAL APPLICATIONS (AIBA)

“Consolidated” Research Group

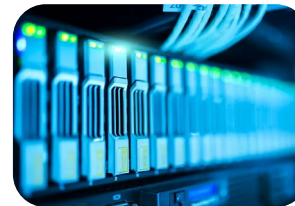
Formed by 8 Doctors (Full professors, Associate professors and Assistant professors) in Computer Science with more than 30 PhD students

Advanced ML/DL & CV algorithms focusing on Data-Centric Deep Learning, Probabilistic modeling, Generative AI, Explainable AI

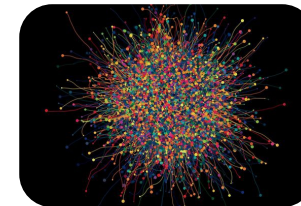
Our portfolio contains >60 research projects, >50 technology transfers, 25 international patents and 2 spin-offs

OUR RESEARCH

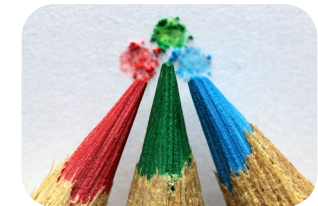
From fundamental AI methods
to scalable, reliable solutions for real-world domains



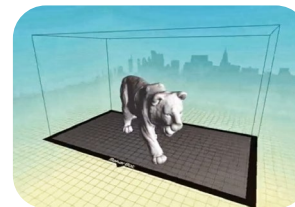
Data-Centric AI



Representation
Learning



Multimodal &
Generative AI



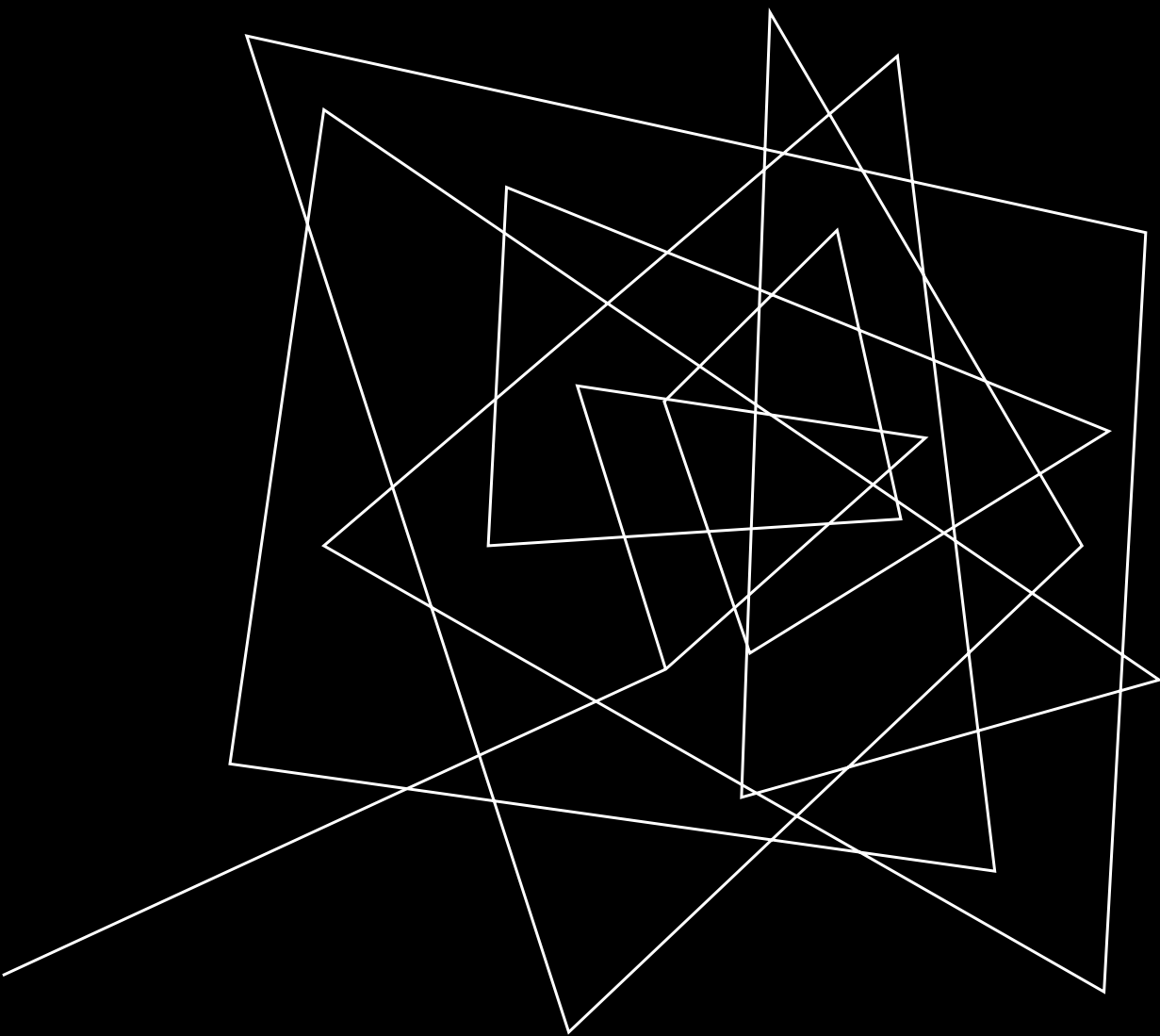
3D Vision &
Understanding



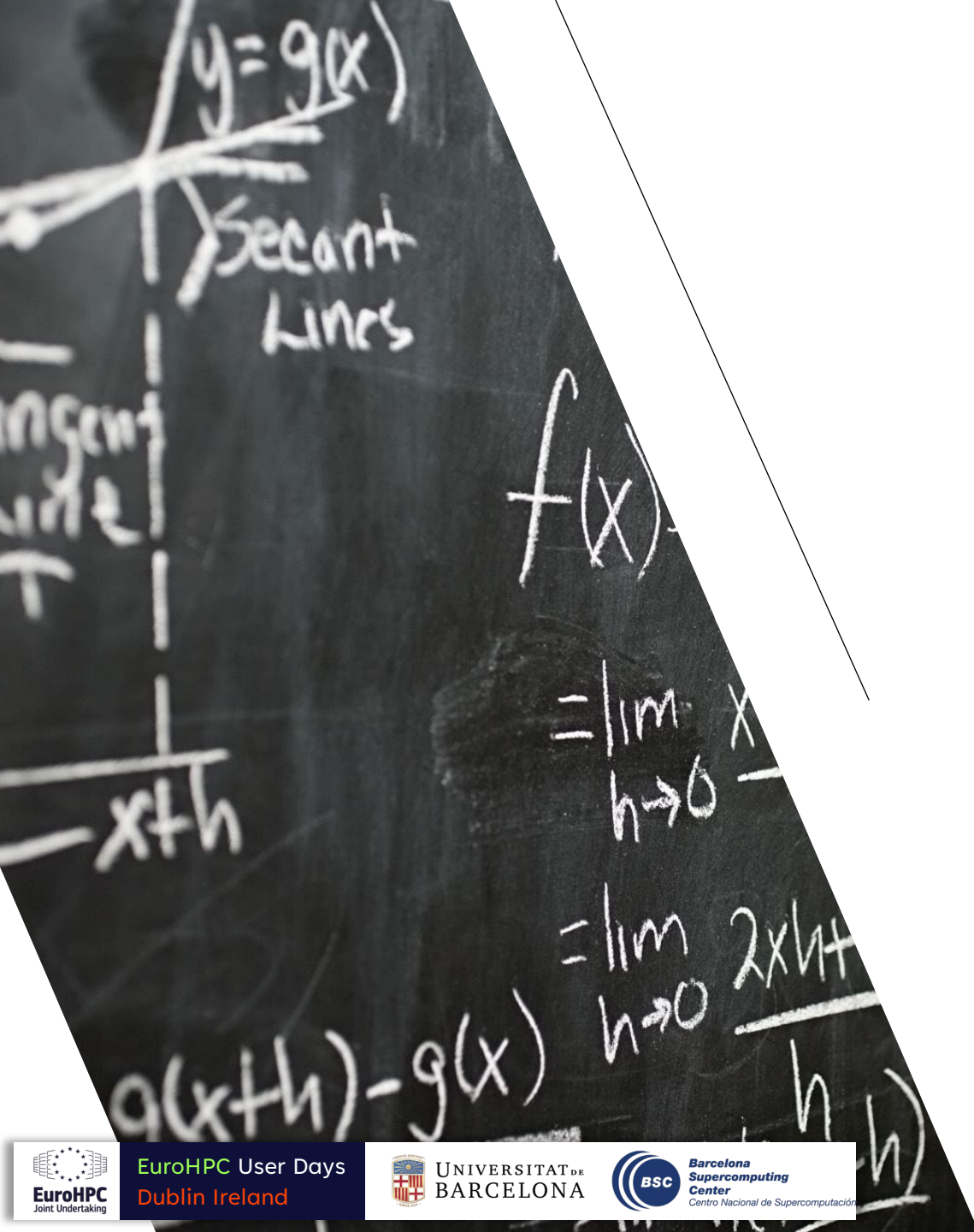
Medical
Imaging



Food AI



THE PROBLEM



HYPOTHESIS

Better data
+
Better learning **>** **Simply bigger models**

Better Data

Scale and improve the data, not only the model

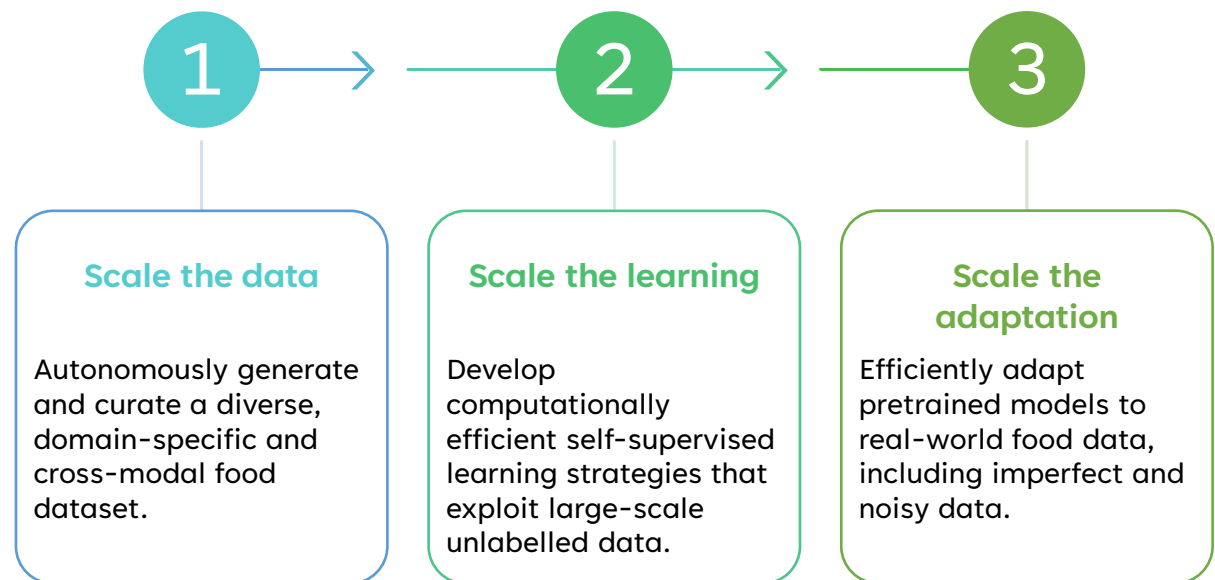
Better Representations

Learning transferrable representations

Better Adaptation

Remaining robust to imperfect data

OBJECTIVES



OUR ALLOCATION

32,568

Node Hours

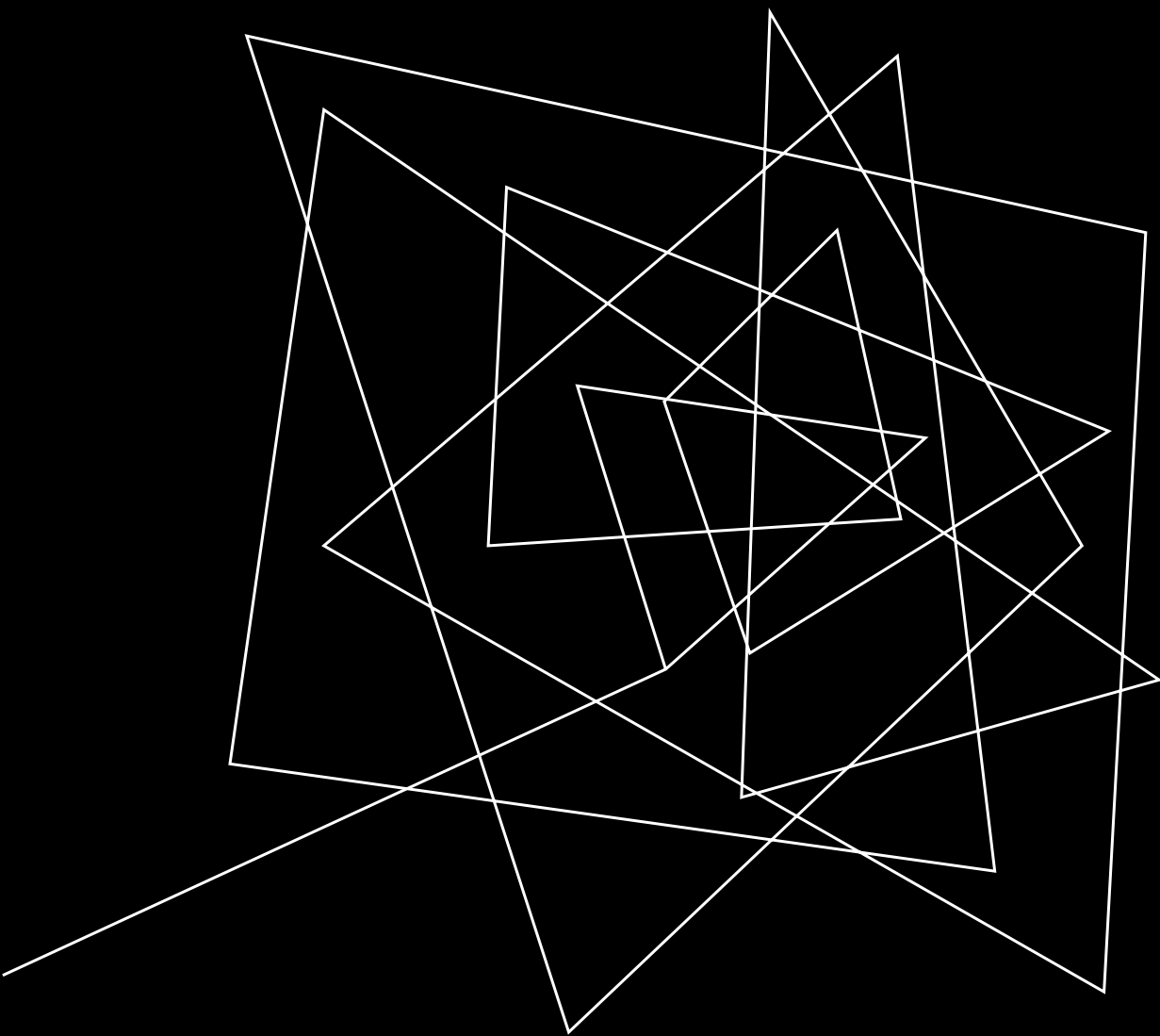
6 Months

Execution Period

MareNostrum 5
ACC

4x Nvidia Hopper GPU
with 64 HBM2 memory

EuroHPC AI Factory for Science & Collaborative EU Projects
Proposal EHPC-AIF-2025SC01-047



SCALABLE AUTONOMOUS DATASET GENERATION AND CURATION



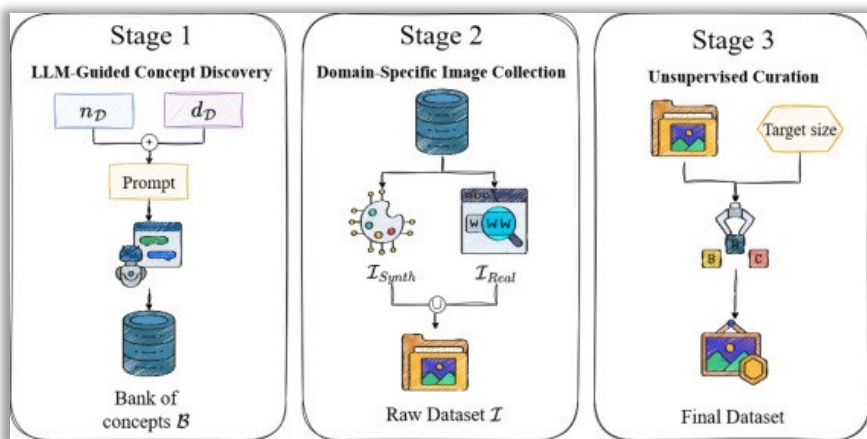
“We scale models.
But what about
scaling the data?”

TOWARDS AUTONOMOUS DATA-CENTRIC AI

Building domain-specific datasets at scale still requires substantial manual effort in concept definition, collection, cleaning and annotation.

Precision-at-Scale (PaS) shifts data scaling from a human annotation problem to a computational scaling problem.

PRECISION-AT-SCALE



Domain	Concepts	Raw Images	Curated Dataset	Syn / Real
Food	3.3k	1.6M	1.2k	60/40
Birds	5.0k	1.5M	1.2k	45/54
Insects	16.6k	2.5M	1.9k	90/10



LLM-guided
Domain concept
discovery

Starting from “just” domain name and a brief description, we discover a concept bank



Domain-specific
Image Collection

Large-scale image retrieval from web-scale databases and synthetic data generation



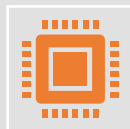
Autonomous
dataset curation

Eliminating redundant and filtering out-of-distribution images through system-curation techniques

Rodríguez-de-Vera, Jesús M., Imanol G. Estepa, Ignacio Sarasúa, Bhalaji Nagarajan, and Petia Radeva. "Precision at scale: domain-specific datasets On-Demand." Pattern Recognition (2025): 112236.

HPC IMPACT

At scale,
data curation itself becomes
an HPC problem.



HPC-scale workload

Scaling PaS to millions of images transforms dataset curation into a compute-intensive workload.



High-throughput representation learning

Generating CLIP ViT-H/14 embeddings at scale requires HPC-class GPU acceleration.



Memory-intensive search & curation

Large-scale FAISS similarity search, deduplication and OOD filtering demand high GPU memory capacity and bandwidth.



Distributed scalability

Multi-node parallel processing enables PaS to scale efficiently as dataset size grows.

RESULTS AND IMPACT

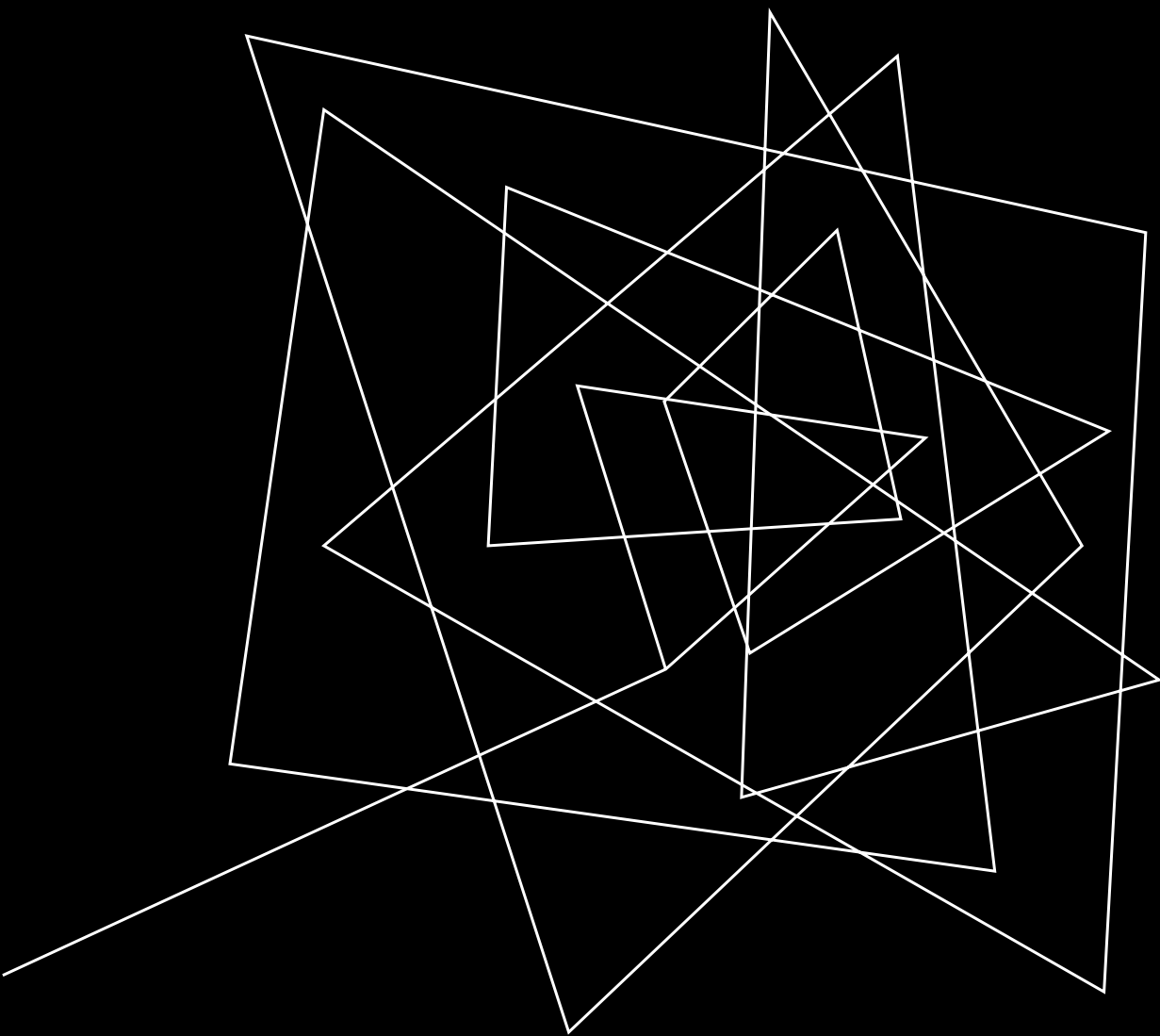
- ❑ Demonstrated a scalable **data-centric** alternative to manually constructed datasets.
- ❑ PaS-Food moved beyond an internal dataset and supported the **Dishcovery Challenges** at **CVPR-MetaFood Workshop 2025 and 2026**.
- ❑ Received **“RES OUTSTANDING PAPER AWARD”** at the 19th RES (Spanish Supercomputing Network (RES)) Users Conference 2025.



Hugging Face

All Models and Datasets are
openly released





ADVANCING
EFFICIENT
REPRESENTATION
LEARNING

“Can we learn
better from all that
unlabelled data”

FROM SCALABLE DATA TO SCALABLE LEARNING

More data is only useful if we can learn from it efficiently.

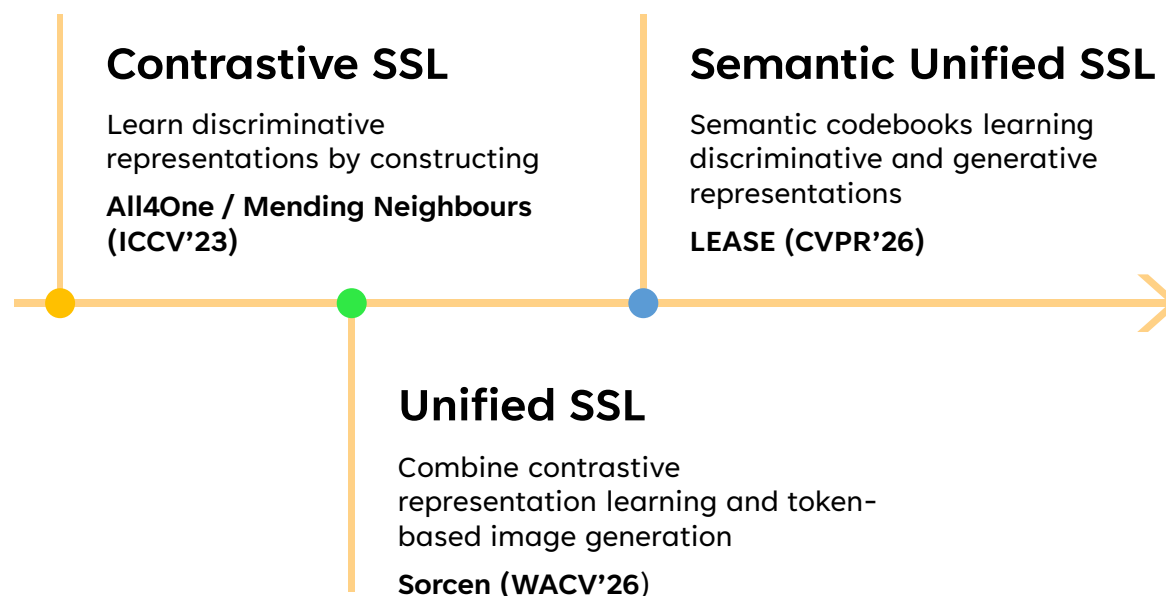
PaS showed that better, domain-specific data gives us better representations.

Here we ask how we can make learning those representations substantially more efficient.

The challenge is not only scaling SSL
but making SSL efficient enough to scale.

REPRESENTATION LEARNING

Learning meaningful and transferable visual representations from large-scale unlabelled data.



Pretraining	Food-101	FoodX-251
ImageNet-1K	56.4	43.2
Food-2K	65.4	50.6
PaS-F	77.3	61.9

PaS datasets are effective SSL pretrainers!!!!
[More results reported in the article](#)

Learn **MORE**
abel **LESS**

LEARNING FROM SEMANTIC DICTIONARIES (LEASE)



Semantic Dictionaries

Paired semantic codebooks organize visual information into discriminative and generative tokens.

Unified Learning

A single framework jointly learns visual representations + image generation.

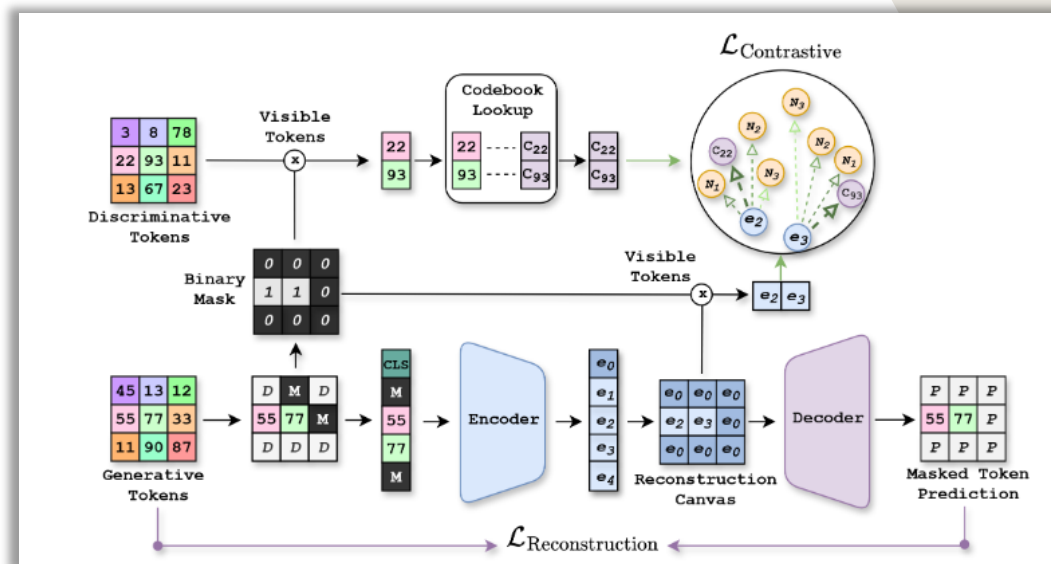
Semantically Aligned Representations

Allow semantics learned for understanding to guide generation.

Efficient Design

Operates in a discrete token space without online tokenizers, teacher networks, or extensive augmentations.

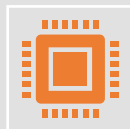
Can a single representation learn to both understand and generate images?



Estepa, Imanol G., Jesús M. Rodríguez-de-Vera, Bhalaji Nagarajan, and Petia Radeva.
 "Learning from semantic dictionaries: Discriminative codebook contrastive learning for unified visual representation and generation."
 In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22456-22466. 2026.

HPC IMPACT

At volumes,
learning better representations
efficiently becomes an HPC
challenge.



Large-scale SSL

Millions of unlabelled images ×
repeated pretraining experiments



Large Batch Sizes

ViT-L training requires large
aggregate GPU memory and up to
64 GPUs / 16 nodes in the
targeted setup



Efficient GPU Training

PyTorch 2.0 + FlashAttention v2
maximize GPU utilization and
reduce memory overhead



Distributed Training

High-bandwidth multi-node
communication enables
approximately 32.5 h / training
run in the targeted configuration

RESULTS AND IMPACT



❑ LEASE

- ❑ Paired semantic codebooks enable simultaneous discriminative and generative representation learning
- ❑ No online tokenizer - No teacher network - Reduced augmentation dependence



❑ Frequency-Aware Representation Learning

- ❑ Better fine-grained representations under domain shift

❑ Redundancy-Reduced MAE

- ❑ Better representations with less computation

48.7%

LEASE is faster than MAGE

~7h

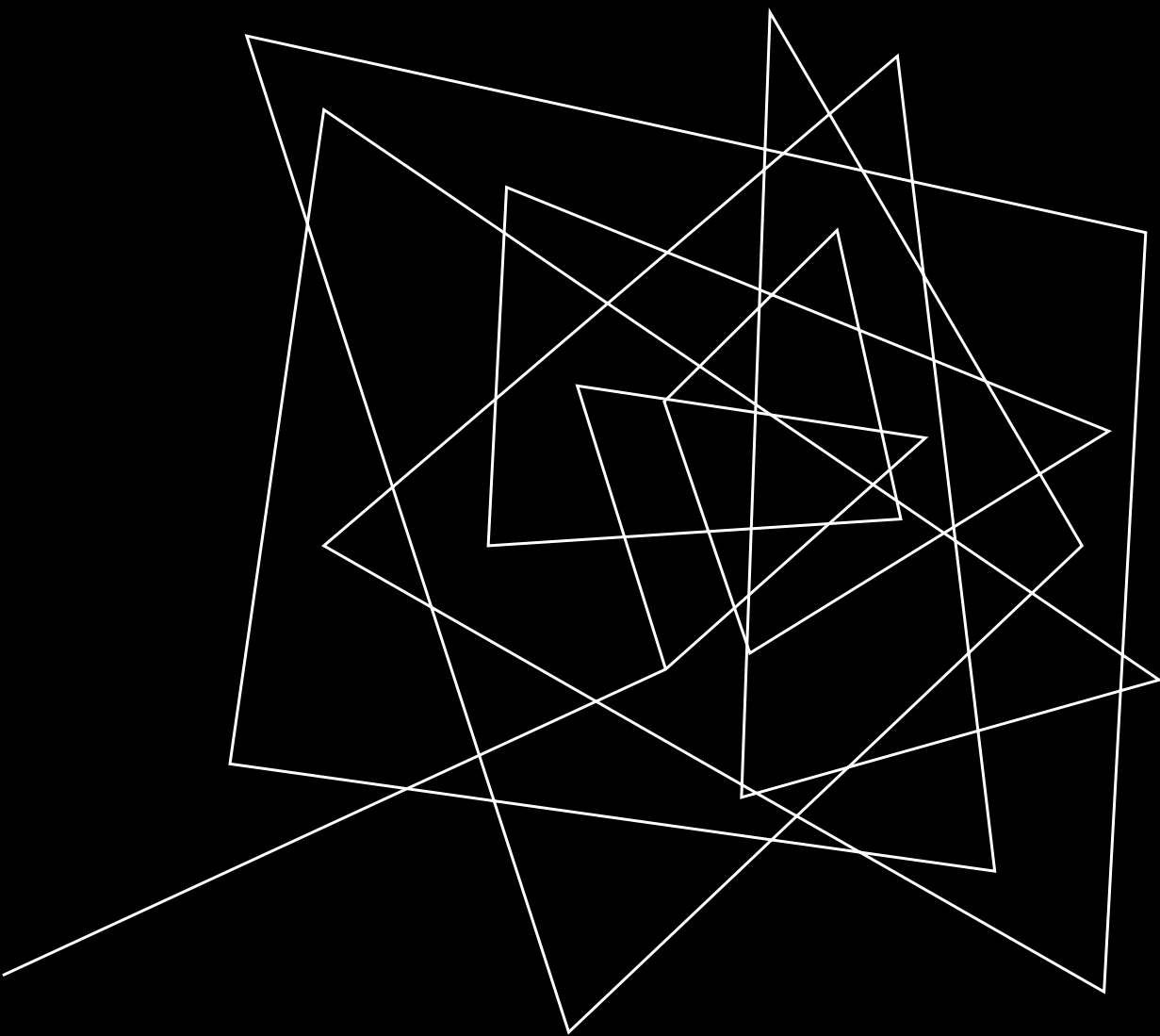
~7h GPU hours
precomputation

$\frac{1}{5}$

~951 GPU hours
overhead for MAGE

>82%↓

RRMAE vs Hydrid SSL SoTA



SCALABLE DOWNSTREAM MODELS AND APPLICATIONS



“Can we **adapt**
reliably when
downstream data is
imperfect?”

EFFICIENT & RELIABLE ADAPTATION

Powerful foundation models are only useful if we can adapt them efficiently and reliably.

Here we ask how we can adapt these models efficiently, while remaining robust to imperfect real-world supervision.

The challenge is not only adapting large models, but making adaptation efficient and reliable enough to scale.

SCALABLE DOWNSTREAM ADAPTATION



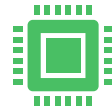
Parameter-Efficient Adaptation

Adapt large pretrained models by optimizing only a small fraction of their parameters.



Learning with Imperfect Data

Develop adaptation strategies robust to noisy and unreliable supervision.



Vision-Language Models

Exploit pretrained visual-semantic knowledge rather than training task-specific models from scratch.



Scalable Experimentation

Evaluate adaptation across models, datasets and configurations.

PEFT

**reduces training cost
but not the size of the model**

Large VLMs such as **EVA-CLIP (~5B)** impose substantial VRAM requirements.

**PEFT reduces what we train
HPC determines what we can explore.**

RESULTS AND IMPACT



❑ MART

- ❑ Real-world noise is not homogeneous: datasets contain both mislabeled in-distribution samples and open-world outliers
- ❑ Disentangle noise mechanisms + anchor-routed training rather than treating every unreliable sample identically

❑ DP-SEP

- ❑ PEFT can still overfit incorrect labels when downstream supervision is noisy
- ❑ Use dual-polarity prompts + stochastic entropy perturbation to improve robustness during VLM adaptation

Efficient Domain Adaptation of PaS

Dataset	SigLIP	SigLIP + PaS	Δ
CUB	70.8	71.3	+0.5
iNat Birds	82.6	82.9	+0.3
Birds	70.0	70.7	+0.7
Food-101	89.7	90.1	+0.4
Food-2K	93.9	94.2	+0.3

A small amount of domain-specific data further improves a large pretrained VLM.

SUCCESS CASES: EUROPEAN PROJECTS

- Validithi: Based on LogMeal - a HealthApp and API in the cloud that is able to automatically recognize and analyze food from images.



- Musae: Multisensory User-centred Solutions for Accessible Environments based on AI, Robotics and Wearables



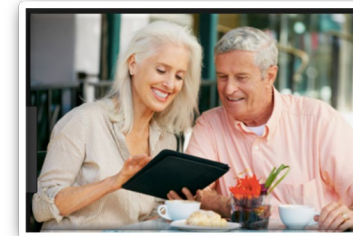
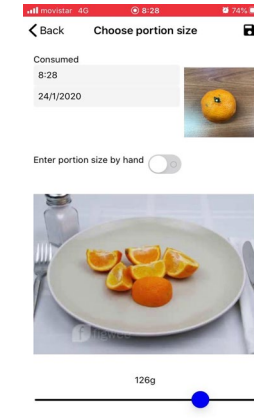
- Nestore: FOOD intake monitoring for malnutrition prevention in elderly

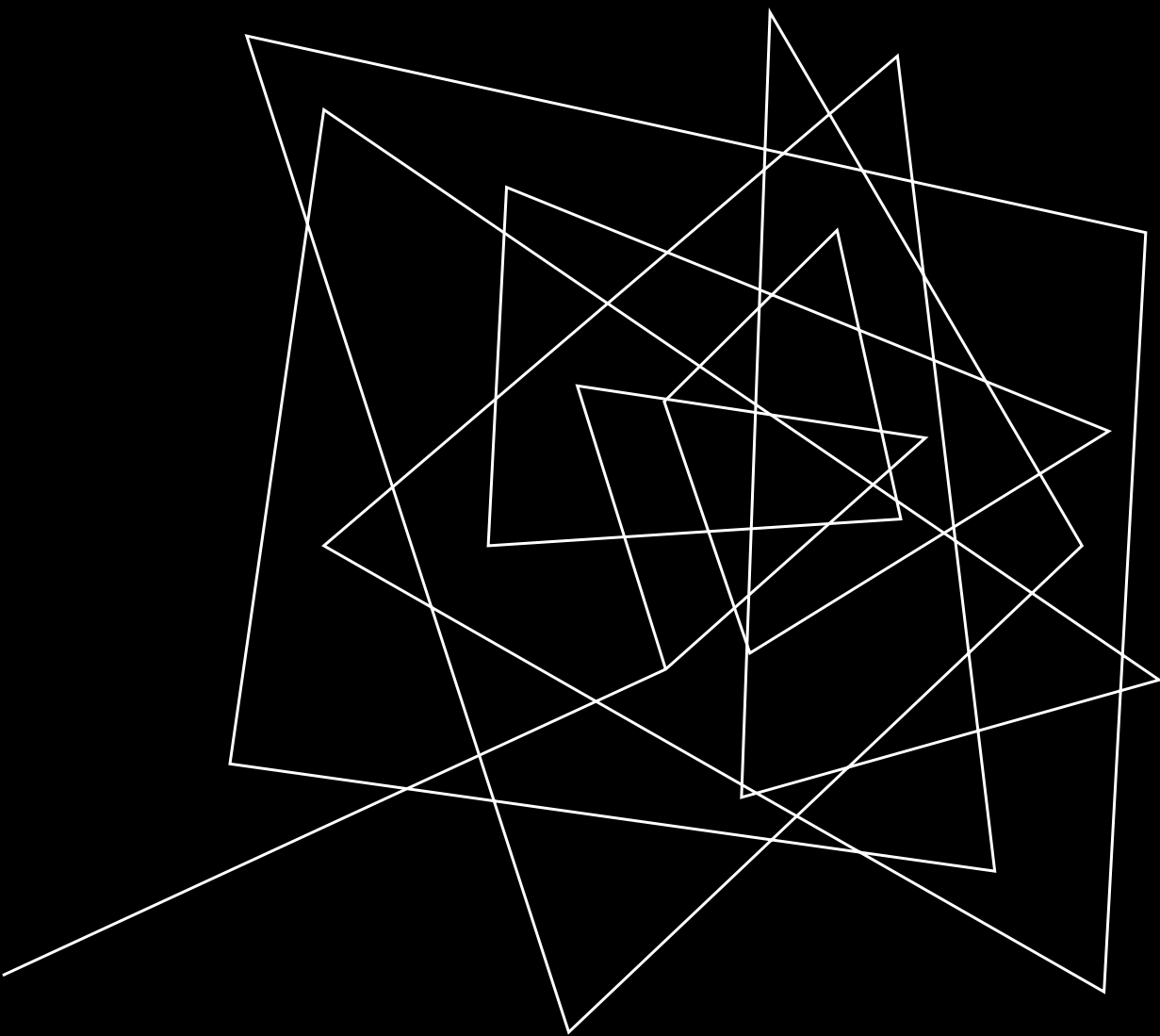


- Greenhabit: Greenhabit: a serious game to promote change behaviour



- Diacare: mHealth app to assist diabetic patients





SUMMARY

FROM HPC SCALE TO EFFICIENT AI

- ❖ Better Data

Use HPC to create and curate better data

- ❖ Better Representations

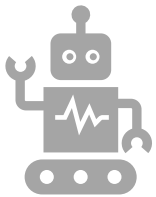
Use HPC to learn better representations while designing algorithms that require less compute

- ❖ Better Adaptation

Efficiently specialize foundation models without expensive full-model retraining

**Towards
Scalable
Efficient
Reliable
Visual
Understanding**

WHAT COMES NEXT?



From domain-specific AI →
general-purpose, scalable
& reliable AI

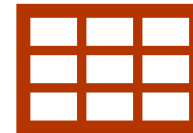


Transfer data-centric
learning to challenging
scientific domains.

Scale representation
learning to settings with
limited supervision, noise
and distribution shifts.



Continue building datasets,
benchmarks and
community challenges.



**The next frontier
requires scaling not
only models
but data, modalities,
domains and
experimentation.**

A series of white, overlapping geometric lines and polygons on a black background, located on the left side of the slide.

THANK YOU

Prof. Petia Radeva

Dept. of Mathematics and Computer Science

University of Barcelona

petia.Ivanova@ub.edu