IAEA
International Atomic Energy Agency

From Grains to Gigabytes:

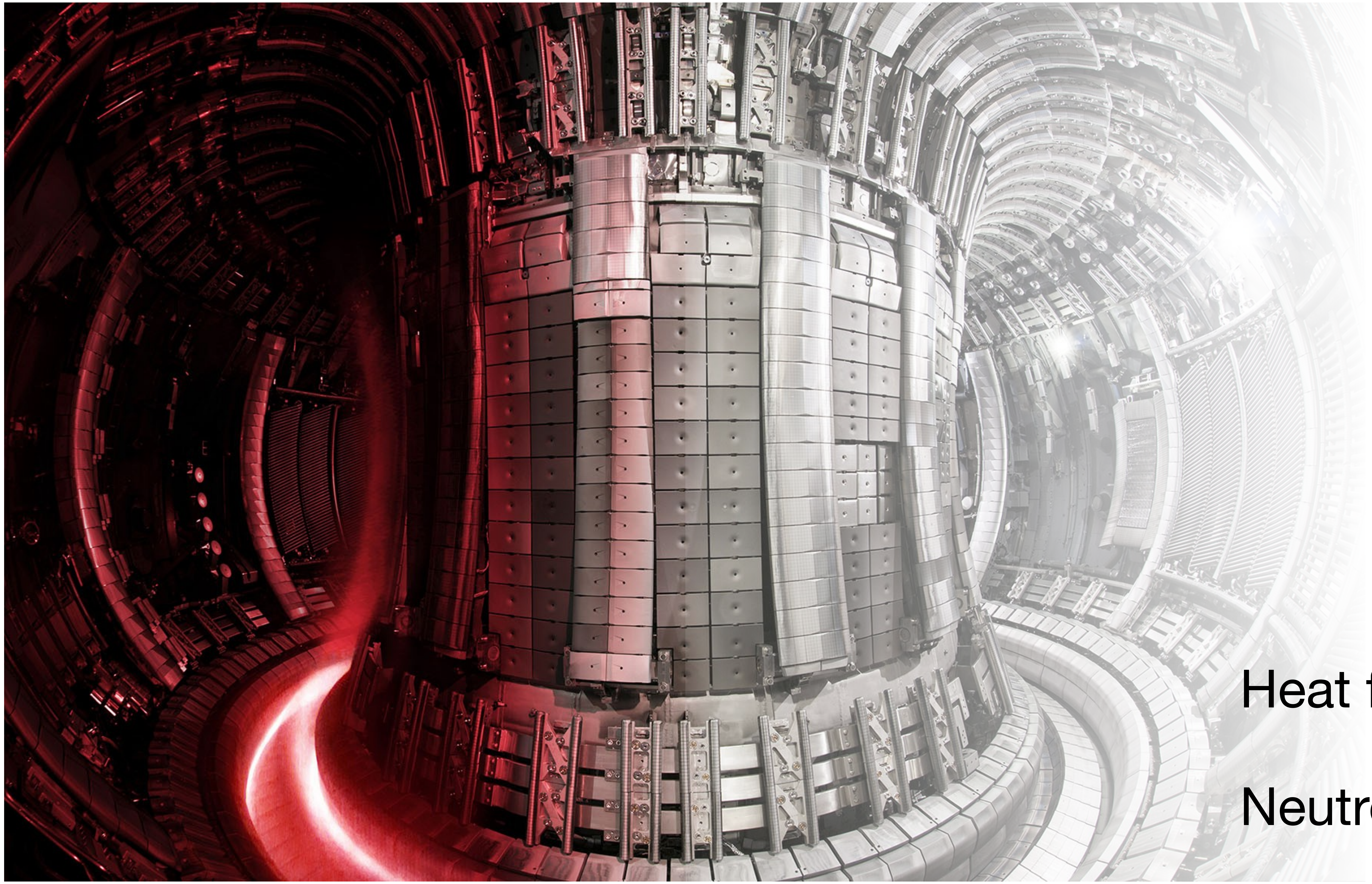
Generating Massive Virtual Specimen for Irradiation Damage Study

Younggak Shin[a,†], Keonwook Kang[a], Byeongchan Lee[b]

[a] *Yonsei University, Seoul, Republic of Korea*
[b] *Kyung Hee University, Yong-in, Republic of Korea*
[†] Presenting author: zeroangle@yonsei.ac.kr

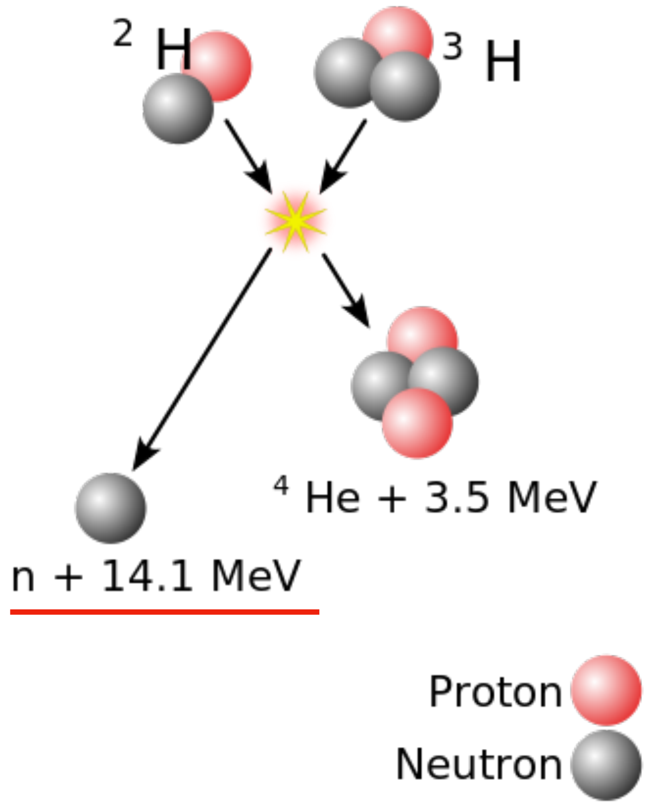
MoD-PMI, Vienna
May 26 2025

Irradiation Damage in Fusion Environment



Eurofusion - JET

D-T fusion reaction



Divertor

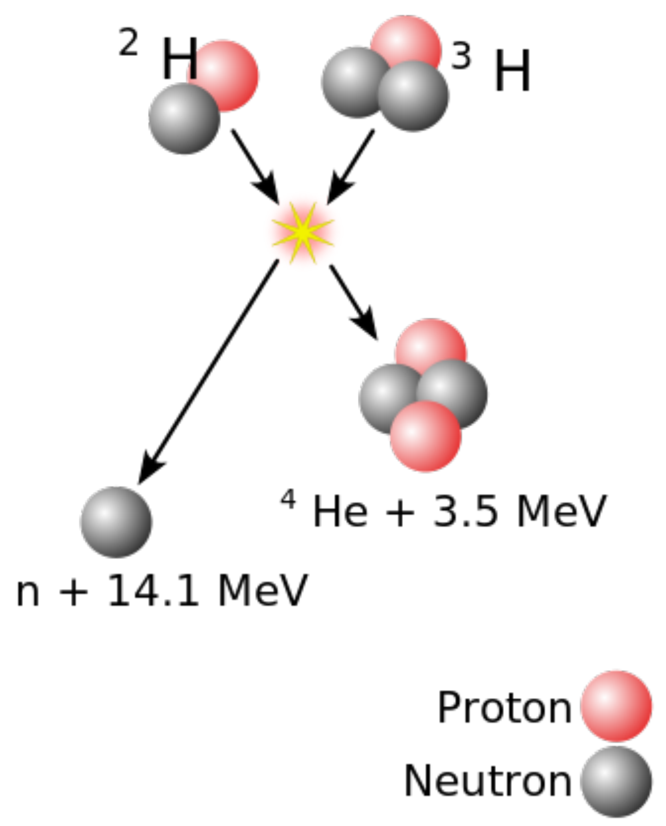
Heat flux : $10 - 20\ MW/m^2$

Neutron flux : $10^{14}\ n \cdot cm^{-2} \cdot s^{-1}$

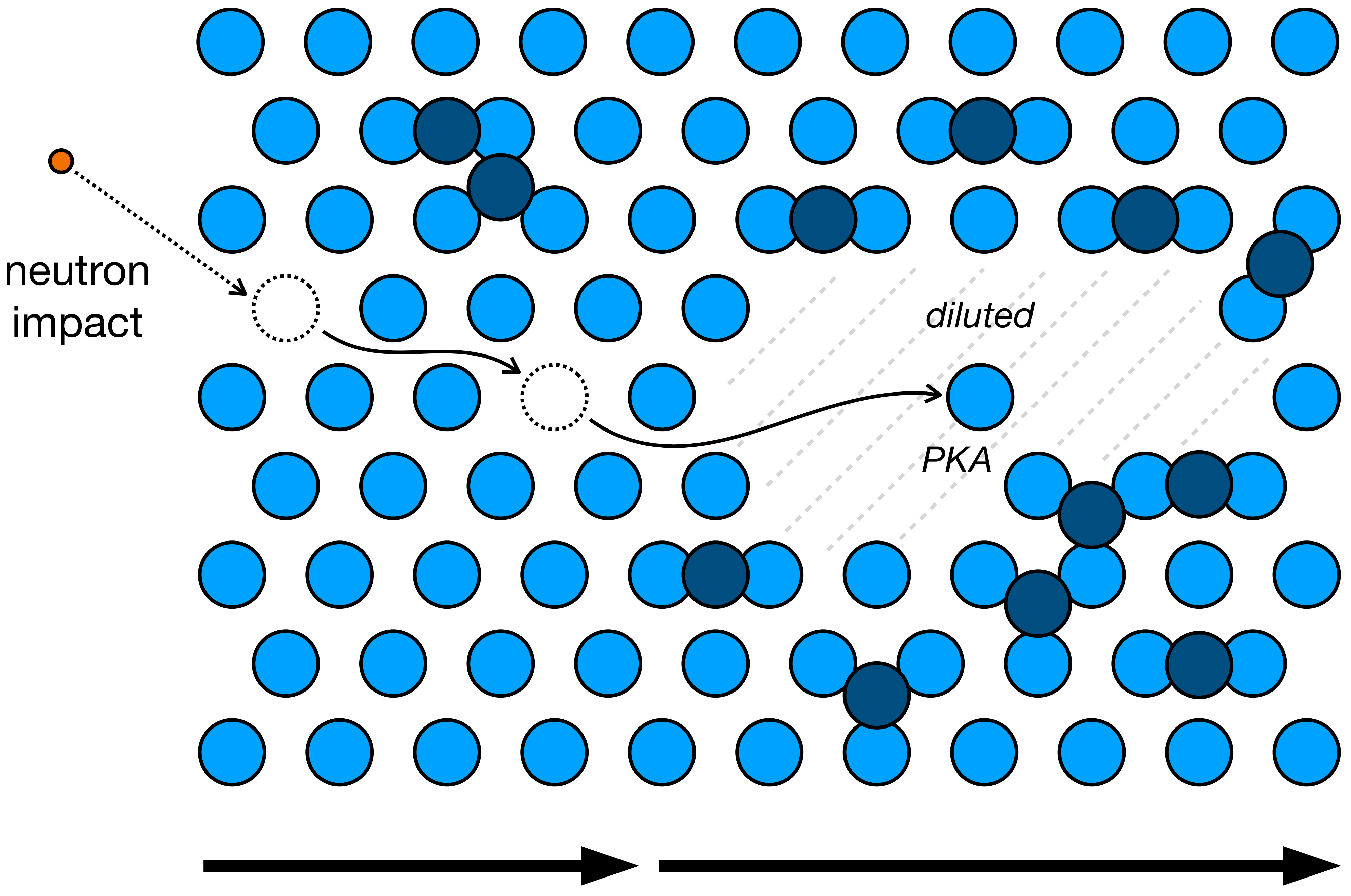
zeroangle@yonsei.ac.kr

Irradiation Damage in Fusion Environment

Collision Cascades

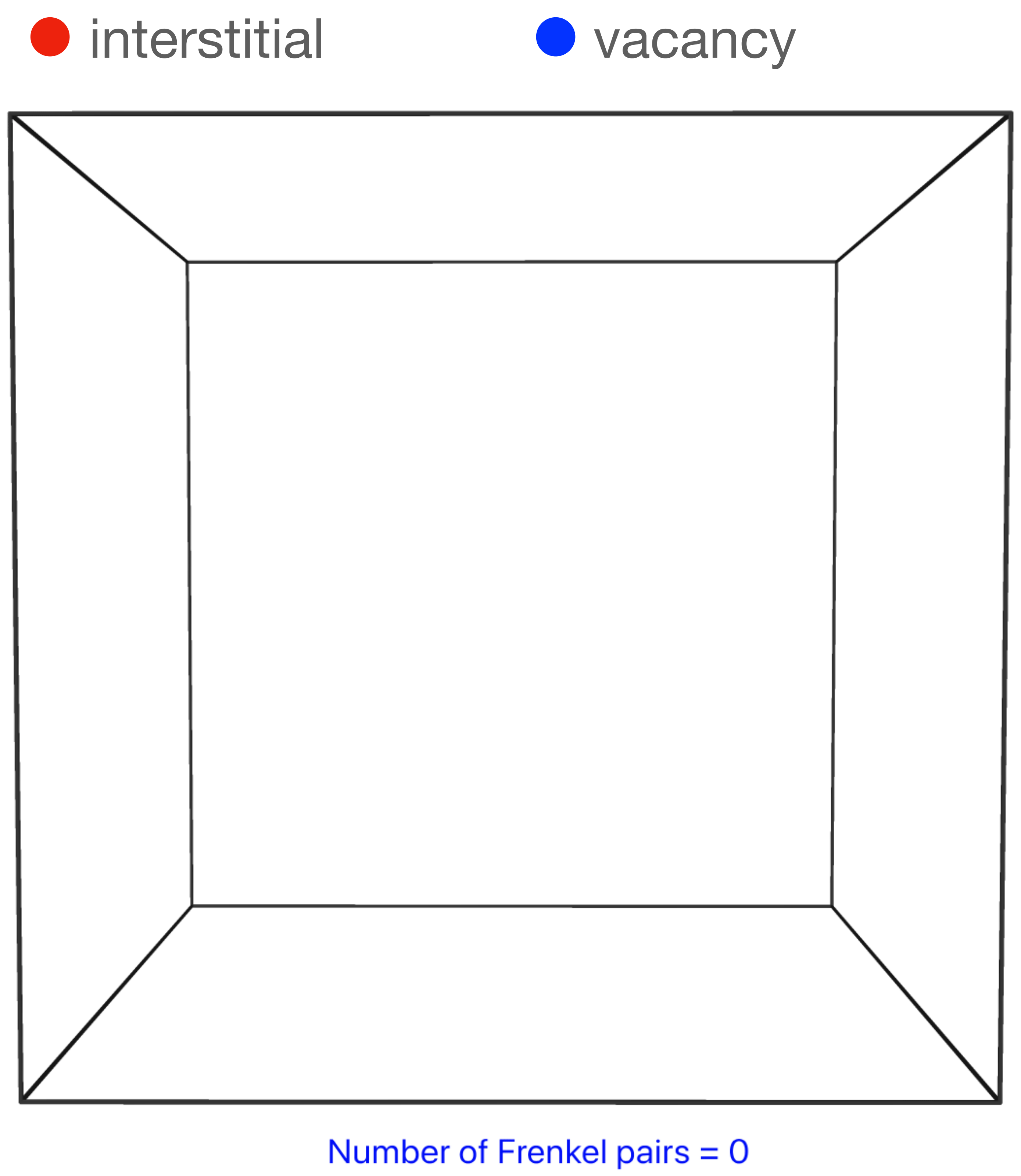


D-T
fusion reaction



ballistic collision
~ 0.1 ps

heat spike
few ps



Irradiation Damage in Fusion Environment

Collision Cascades → Volumetric Defects

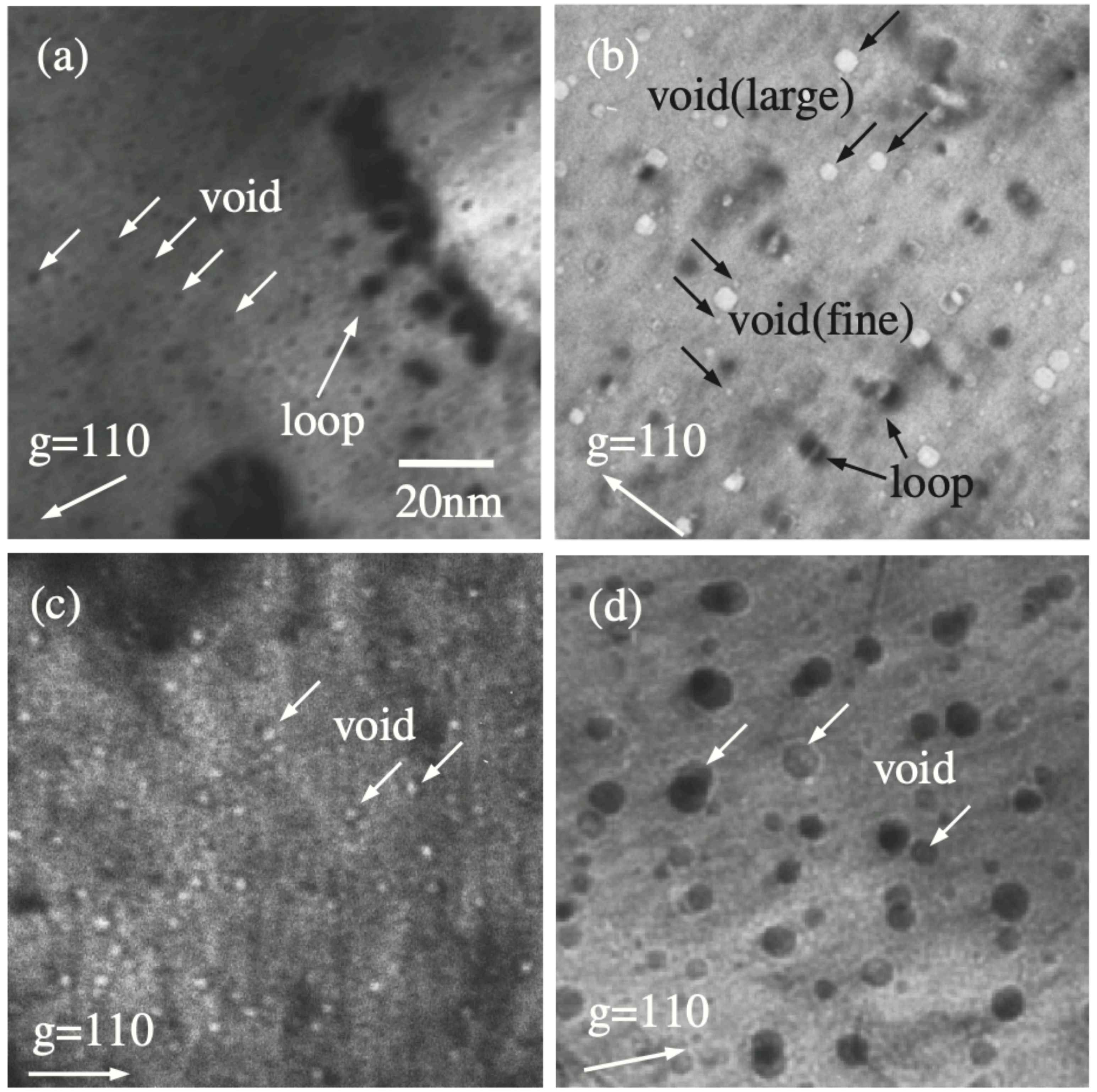


Fig. 2 Microstructural observations of pure W irradiated to (a) 0.17 dpa at 400°C, (b) 0.96 dpa at 538°C, (c) 0.40 dpa at 740°C and (d) 1.54 dpa at 750°C. The void images in (a) and (d) are black, because these were taken in an "over focused" condition.

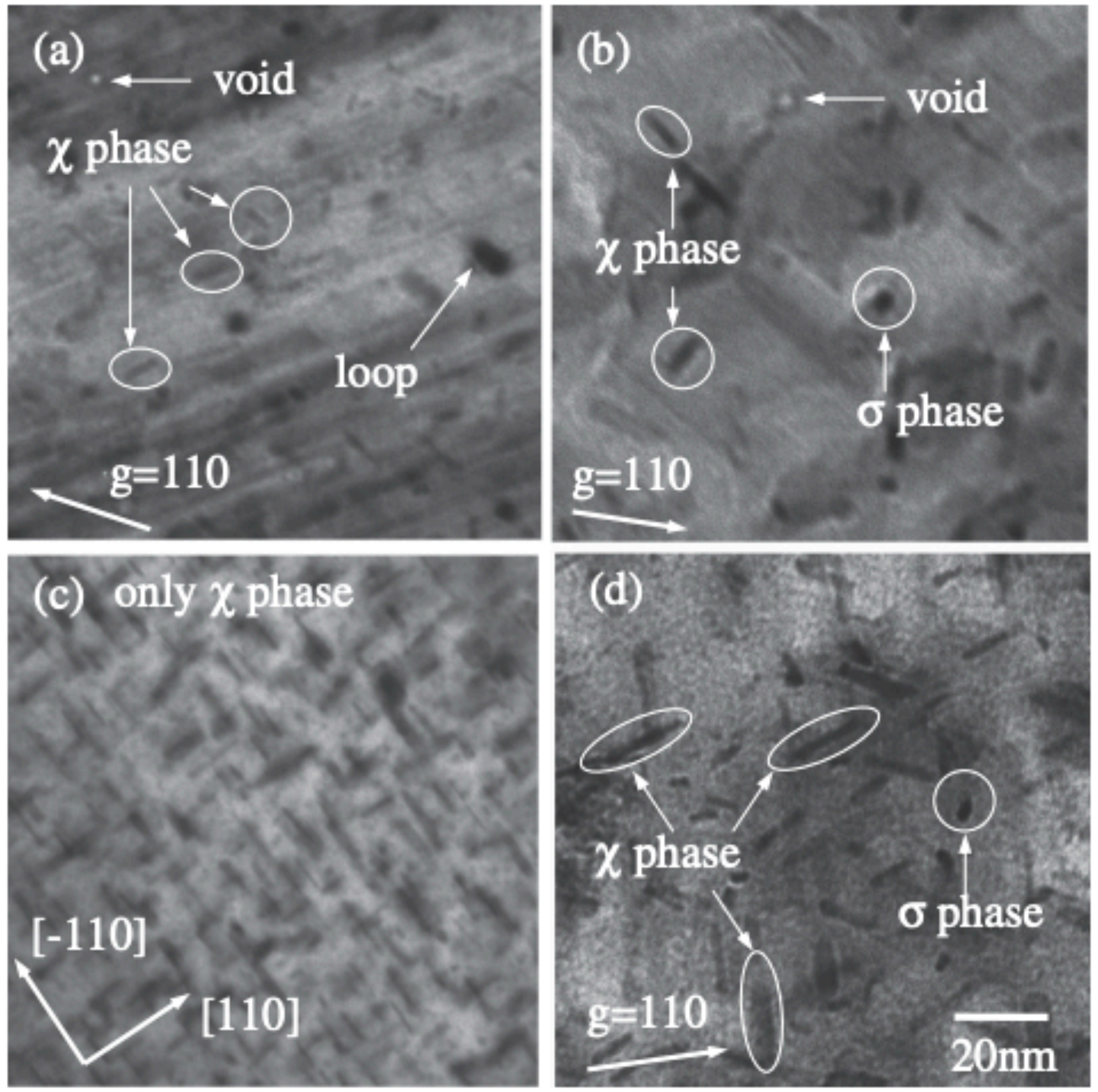


Fig. 4 Microstructural observations of (a) (b) W-5Re and (c) (d) W-10Re, irradiated to (a) (c) 0.96 dpa at 538°C and (b) (d) 1.54 dpa at 750°C. Plate-like precipitates were along {110} planes.

Tanno et al., Mater. Trans. (2011)

Irradiation Damage in Fusion Environment

Polycrystal: Grain Growth

"Recrystallisation"

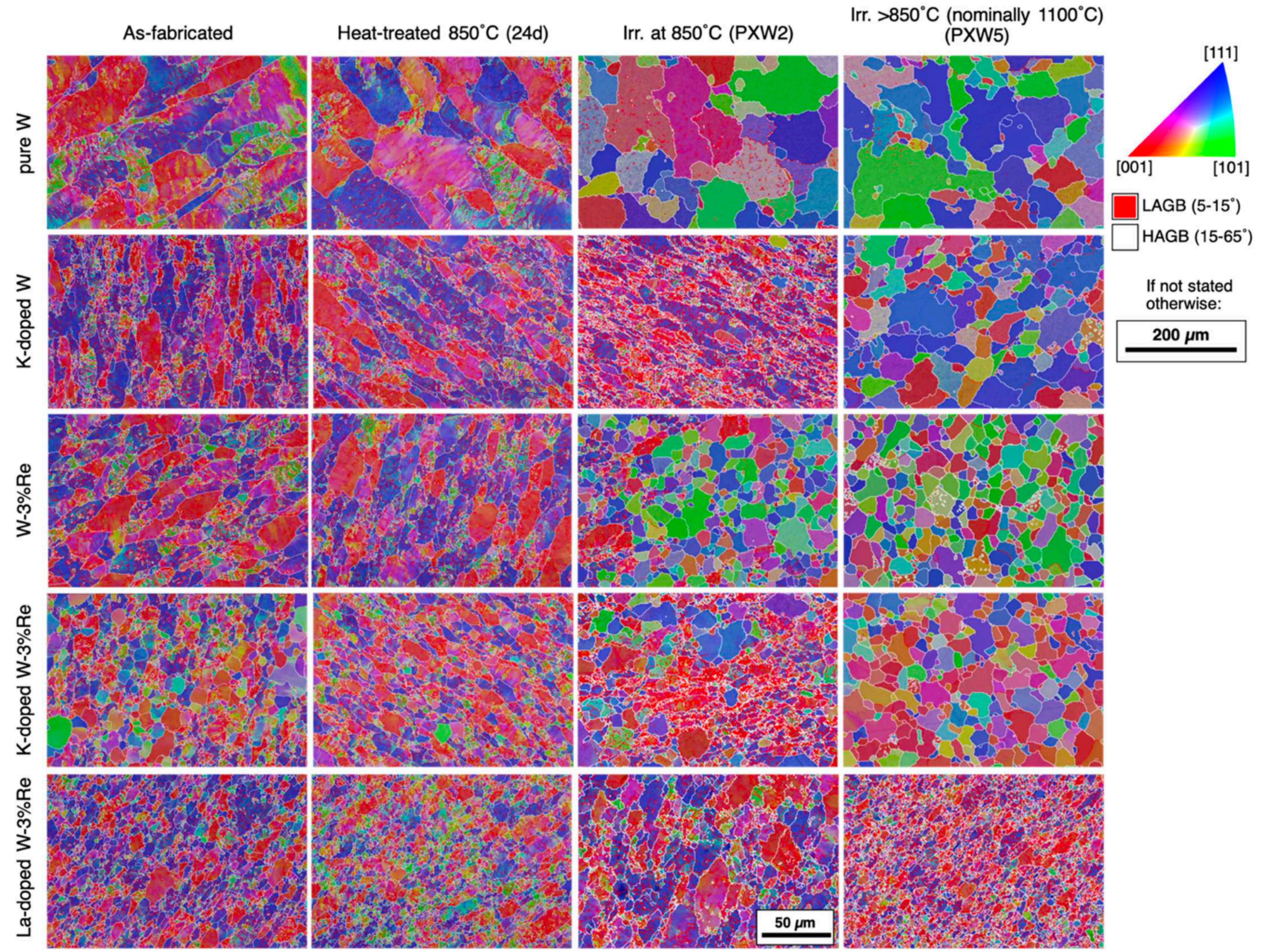
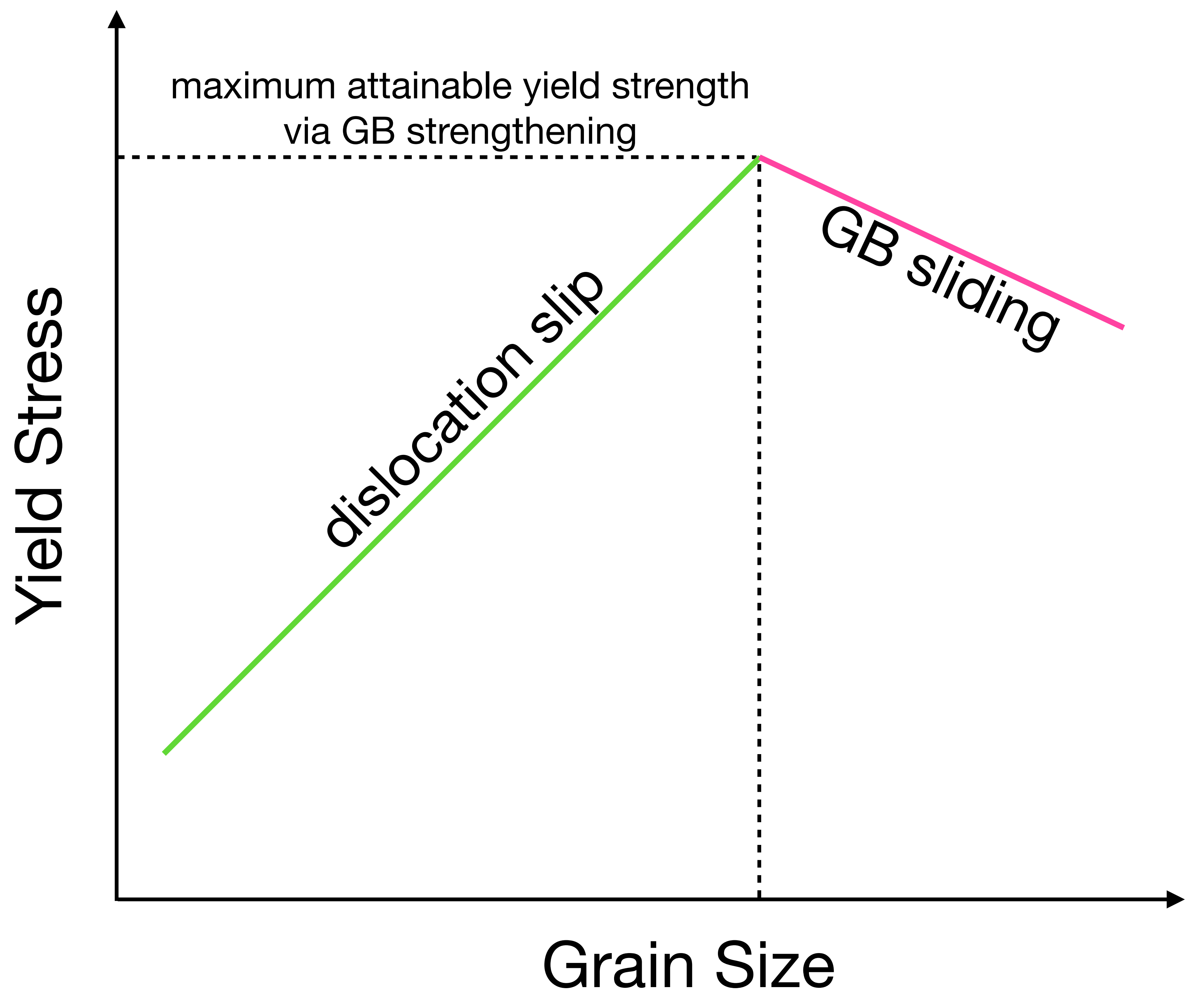


Fig. 1. EBSD orientation maps for the nonirradiated and irradiated materials after correction and spike reduction. HAGBs are marked in white, and LAGBs are marked in red.

Gietl et al., J. Alloys Compd. (2022)

Mechanical properties



Irradiation Damage in Fusion Environment

Polycrystal: Grain Growth

“Recrystallisation”

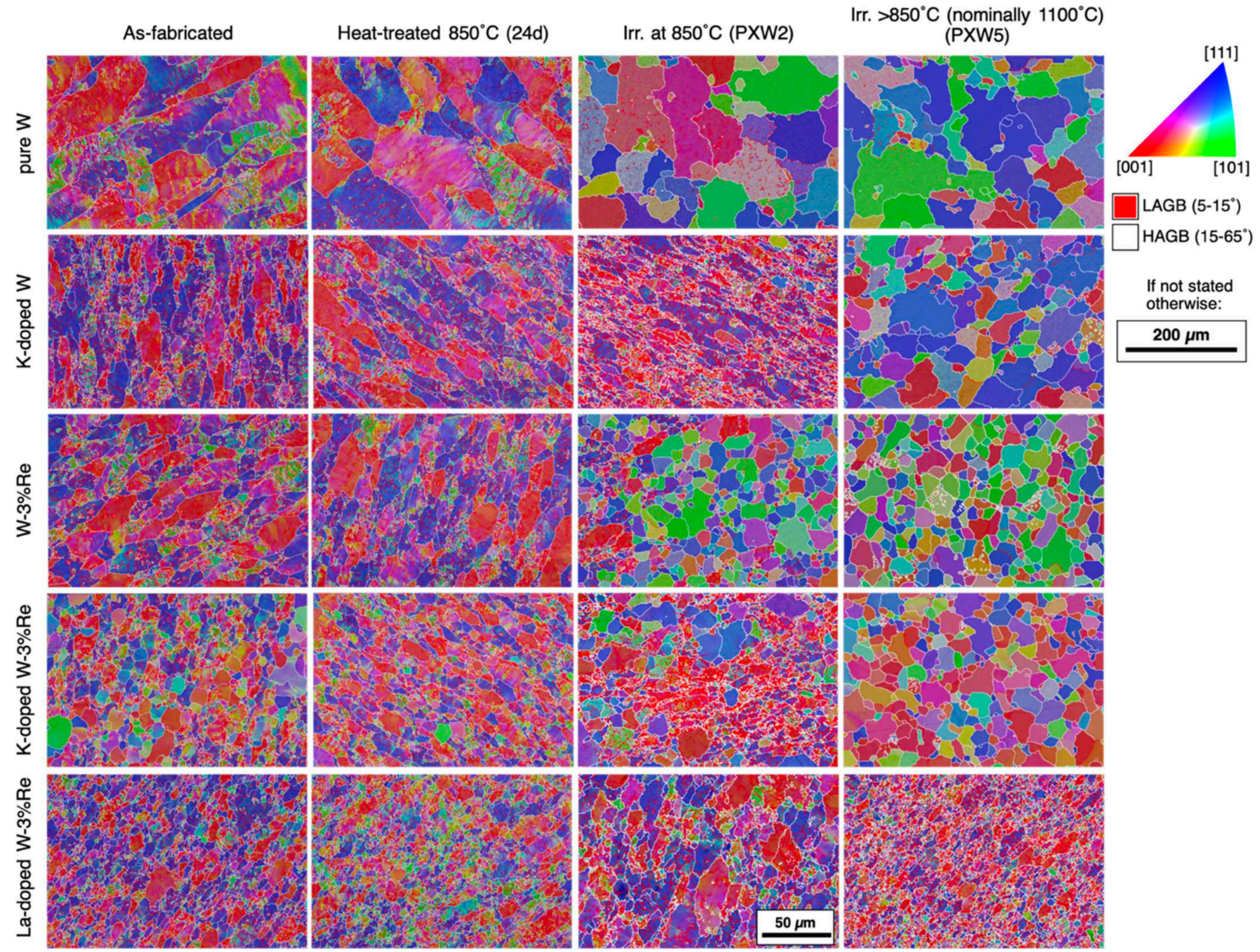
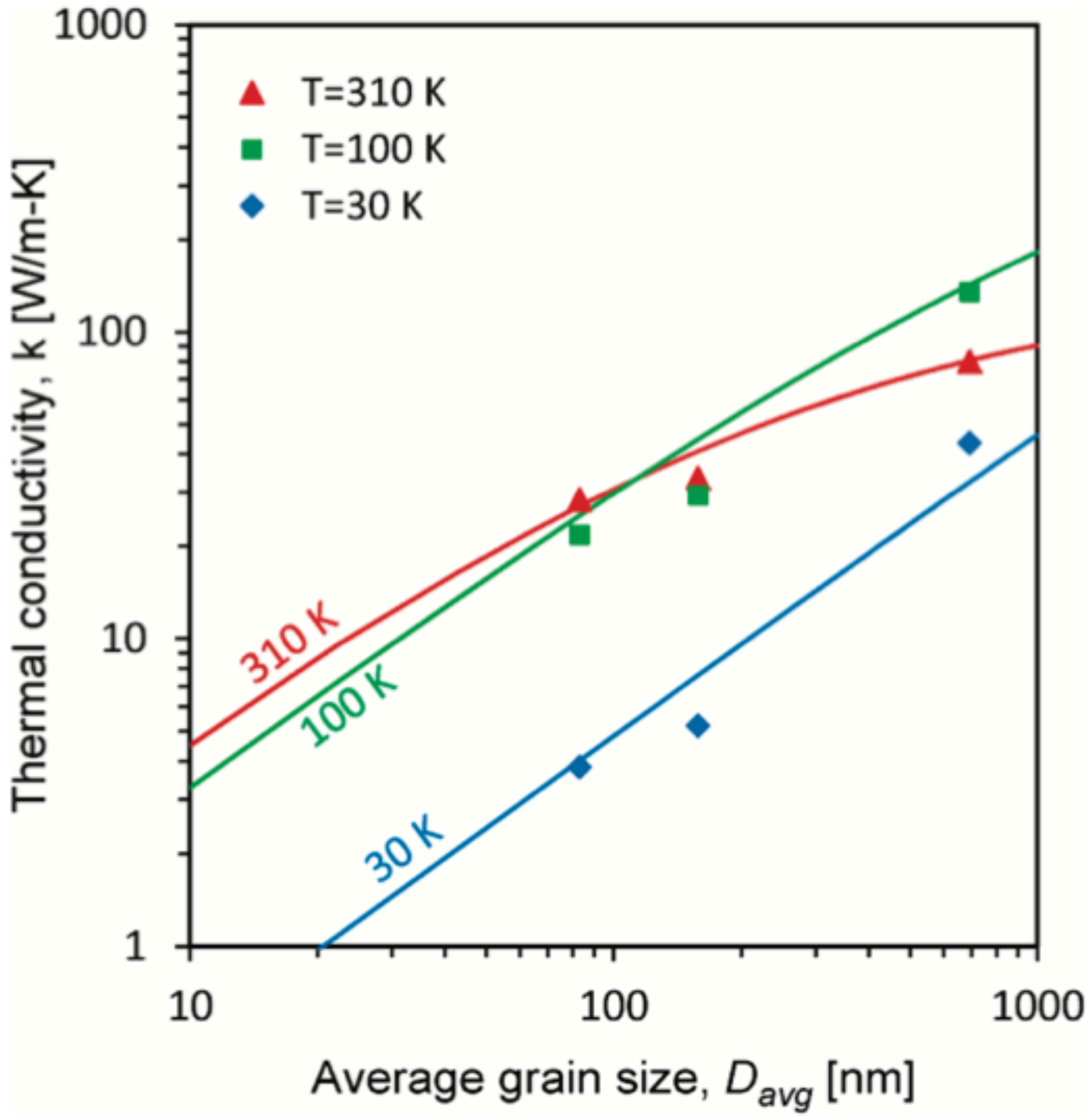


Fig. 1. EBSD orientation maps for the nonirradiated and irradiated materials after correction and spike reduction. HAGBs are marked in white, and LAGBs are marked in red.

Gietl et al., J. Alloys Compd. (2022)

Thermal properties



Wang et al., Nano Lett. (2011)

Irradiation Damage in Fusion Environment

Polycrystal: Grain Boundary as Defect Sink

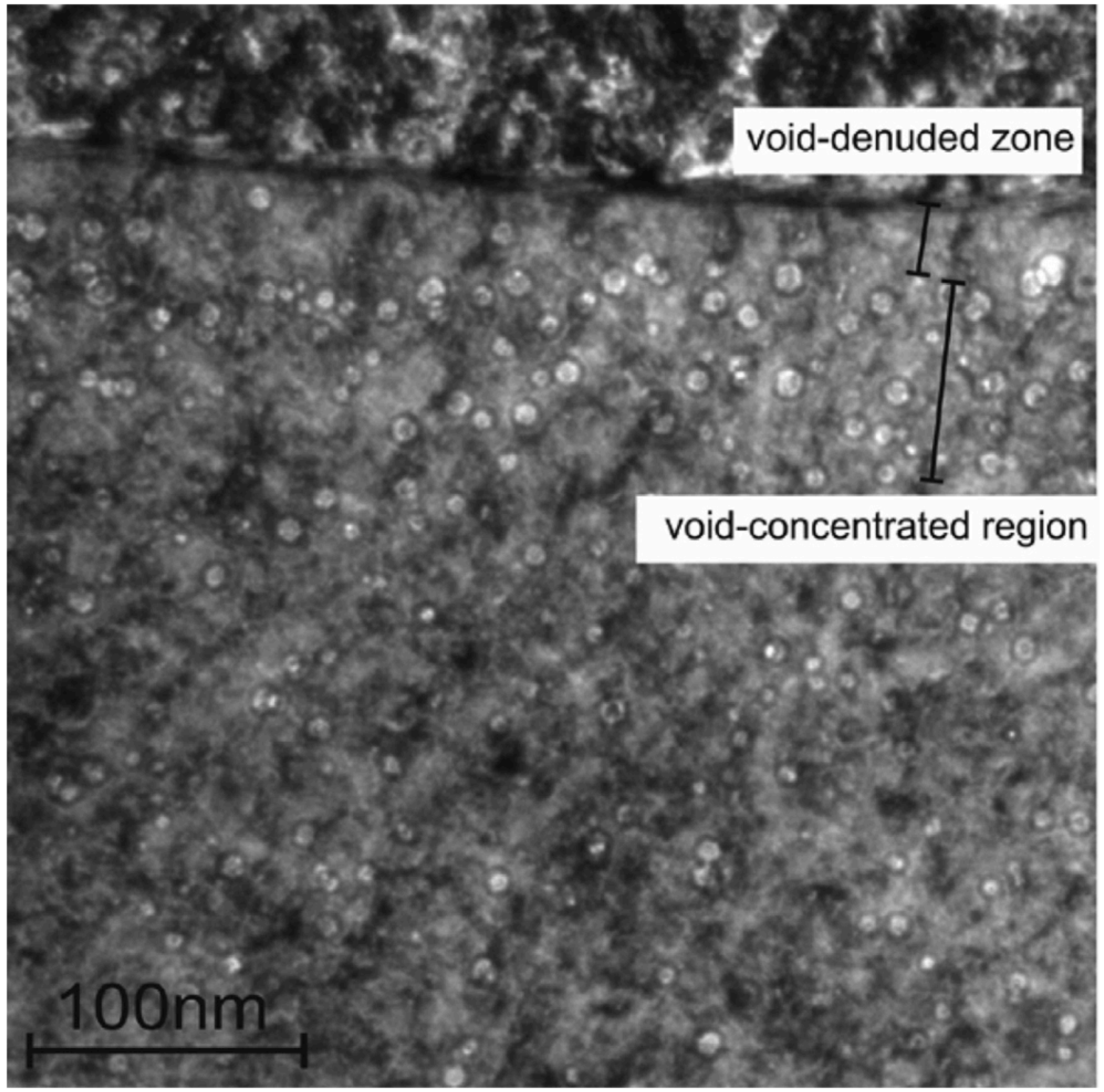


Fig. 7. TEM image of cavities in the region near a grain boundary.

Klimenkov, Nucl. Mater. Energy (2016)

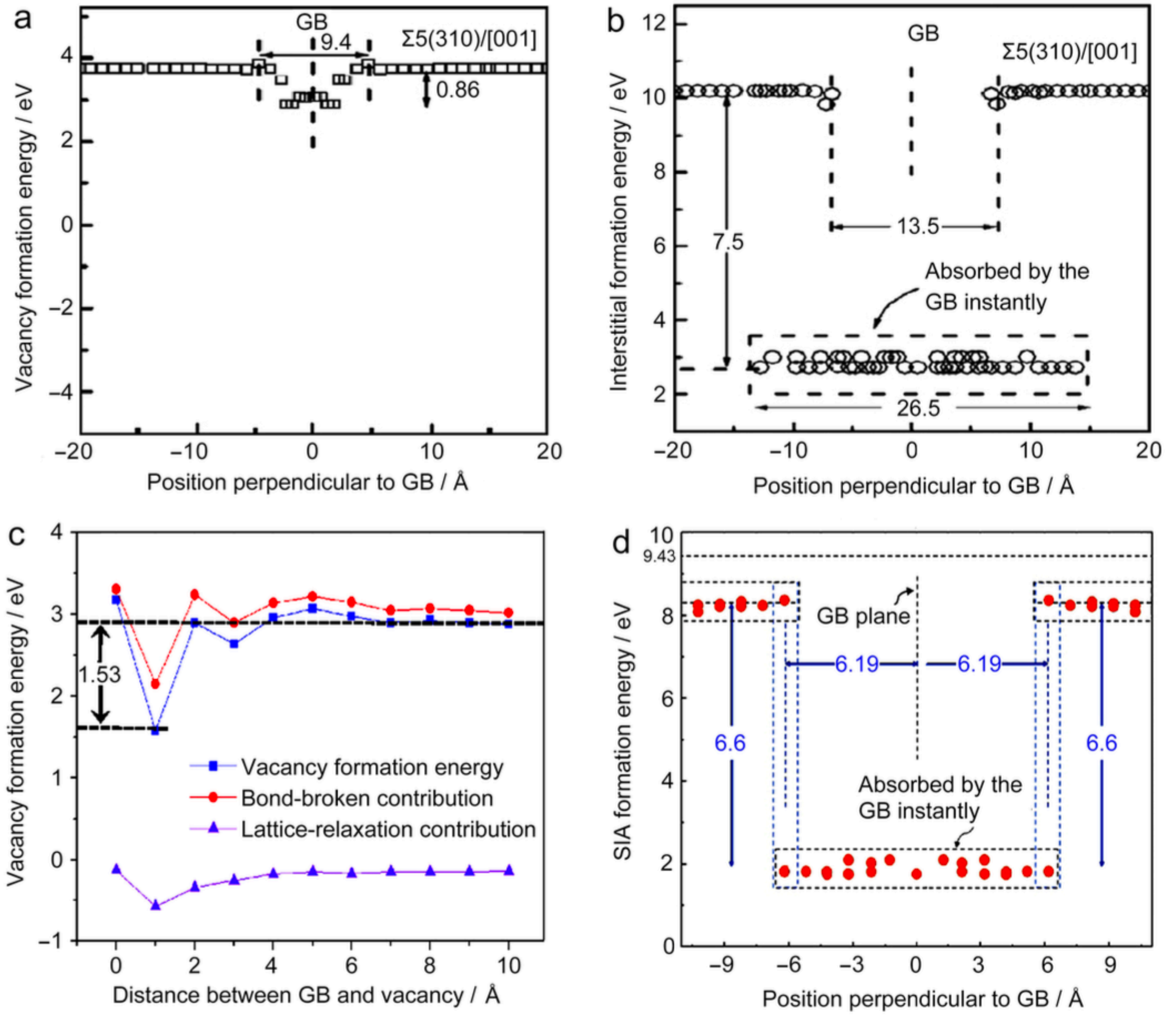
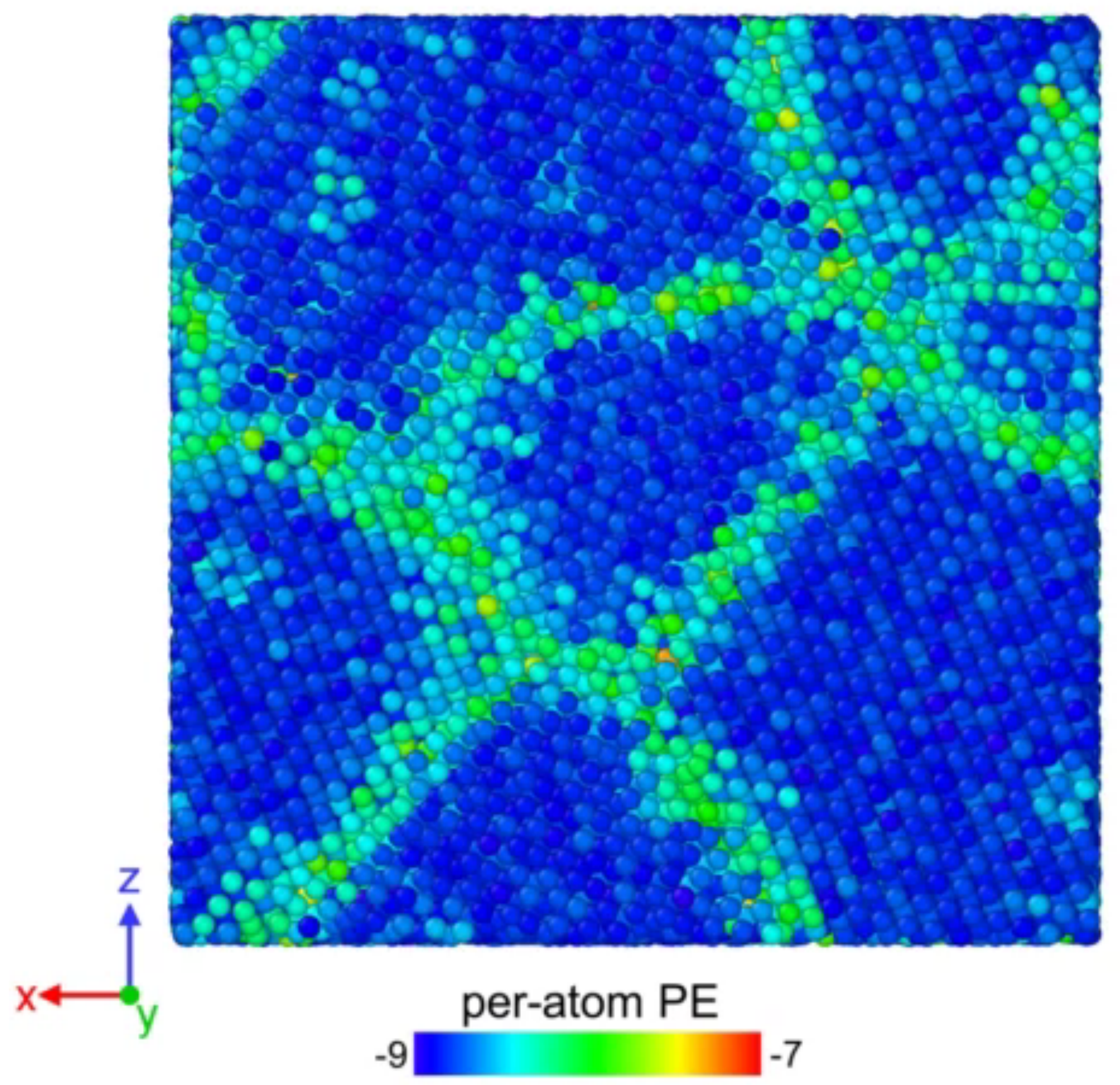


Fig. 4 Variation of the V/SIA formation energy with the initial V/SIA distance from the GB of Σ5(3 1 0)/[0 0 1] in W for **a** Vs and **b** SIAs, respectively. Reproduced with permission from Ref. [20] . Copyright 2013 IAEA. **c** and **d** Corresponding results calculated using a first-principles method. Reproduced with permission from Ref. [23]. Copyright 2017 Elsevier

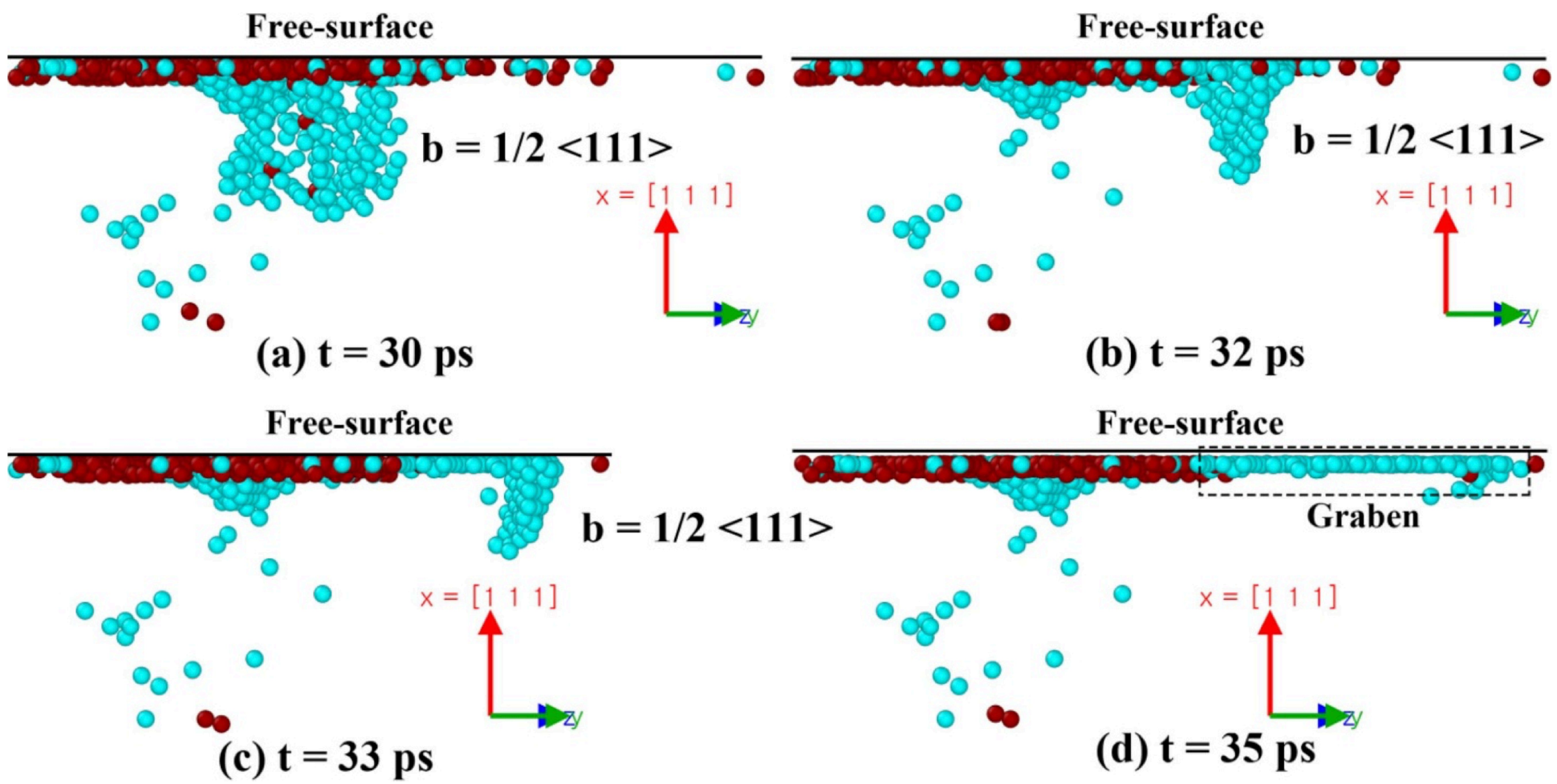
Li et al, Tungsten 2 (2020)

Why We Need Massive-Scale MD Simulation

Tiny Model



Unpublished, courtesy of Yoo



Lee et al., Nucl. Fusion (2020)

Unrealistically small grains

Severe recrystallisation

Large interfacial area

Intense absorption of self-intersititial atoms

Why We Need Massive-Scale MD Simulation

Tiny Model versus Massive Model

	Tiny (< tens of millions of atoms)	Massive (~ 10 billion atoms)
Microstructure fidelity	singlecrystal bicrystal ultrafine-grained*	Full grain-size spectrum
Phenomena captured	aggressive recrystallisation no grain boundary pinning	GB-defect interaction Hall-Petch strengthening grain boundary sliding
Parameter transferability	impractical for multiscale modelling	direct hand-off comparable with experimental data

Massive MD: Present

OAK RIDGE National Laboratory website

Frontier HPC achieved ExaFLOPS in 2023

Are we ready for this?

Massive MD: Present

"No parallel atomic structure generation code exists."

"No detail control of polycrystalline structure."

"LAMMPS community discarded parallel I/O system."

Massive MD: Present

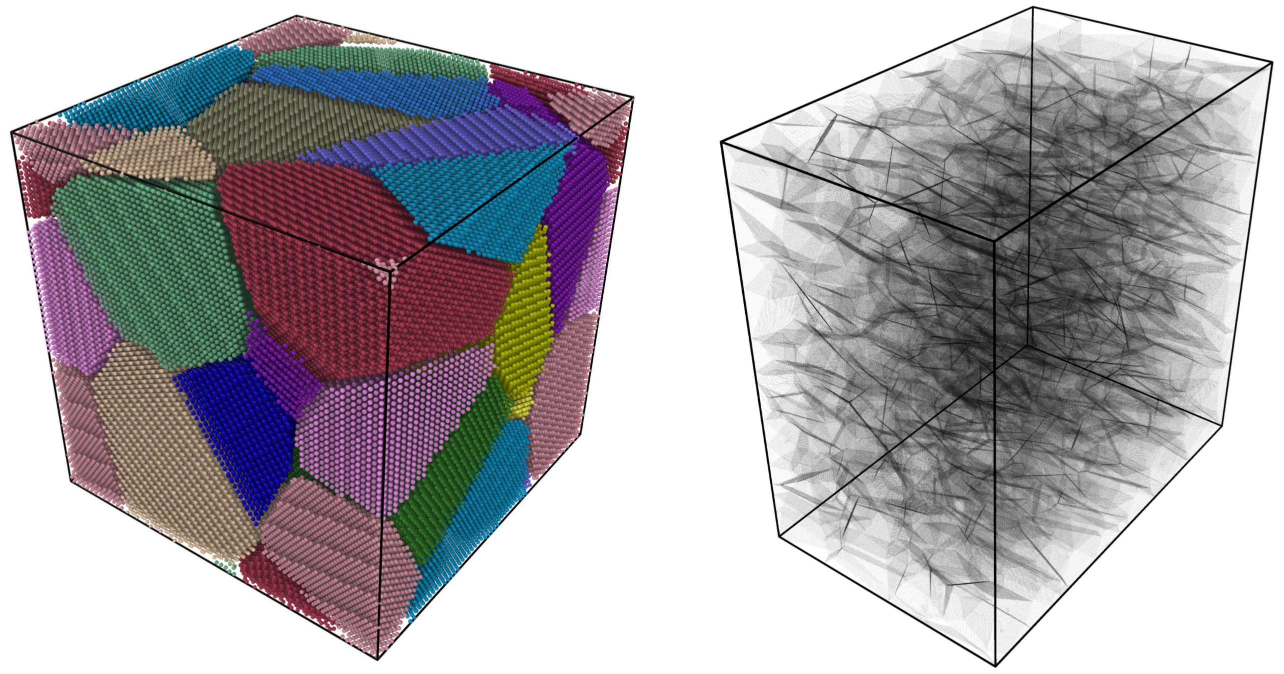
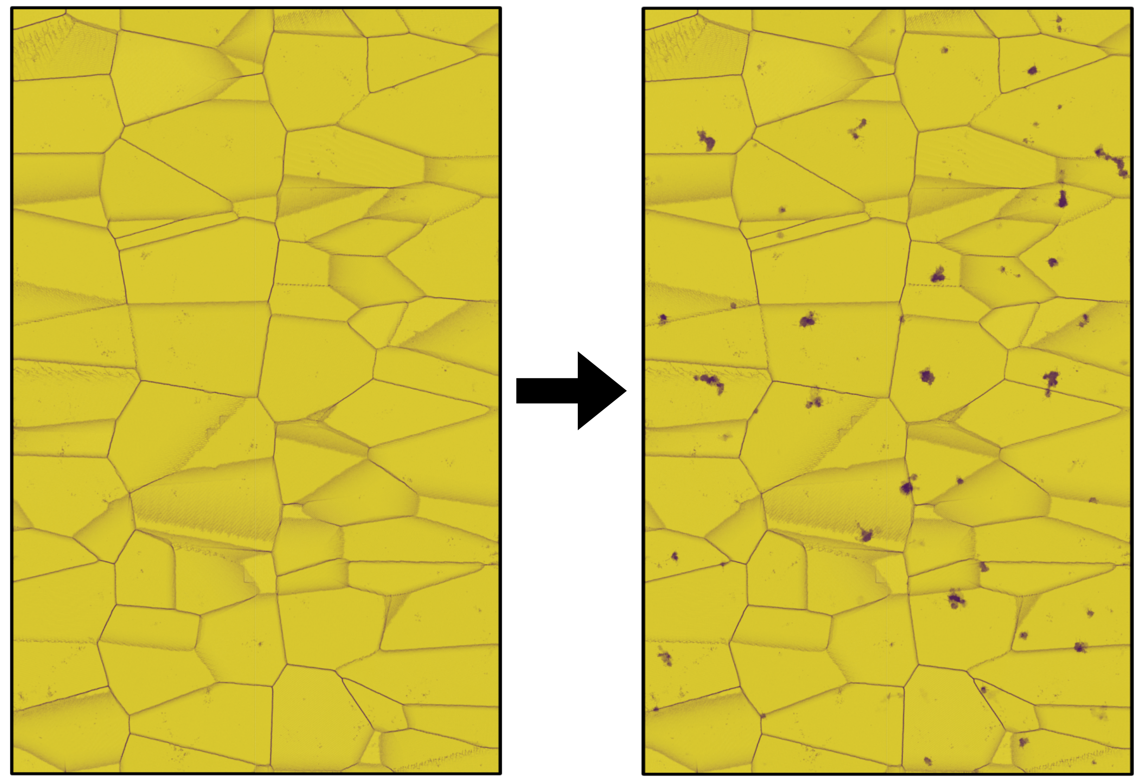
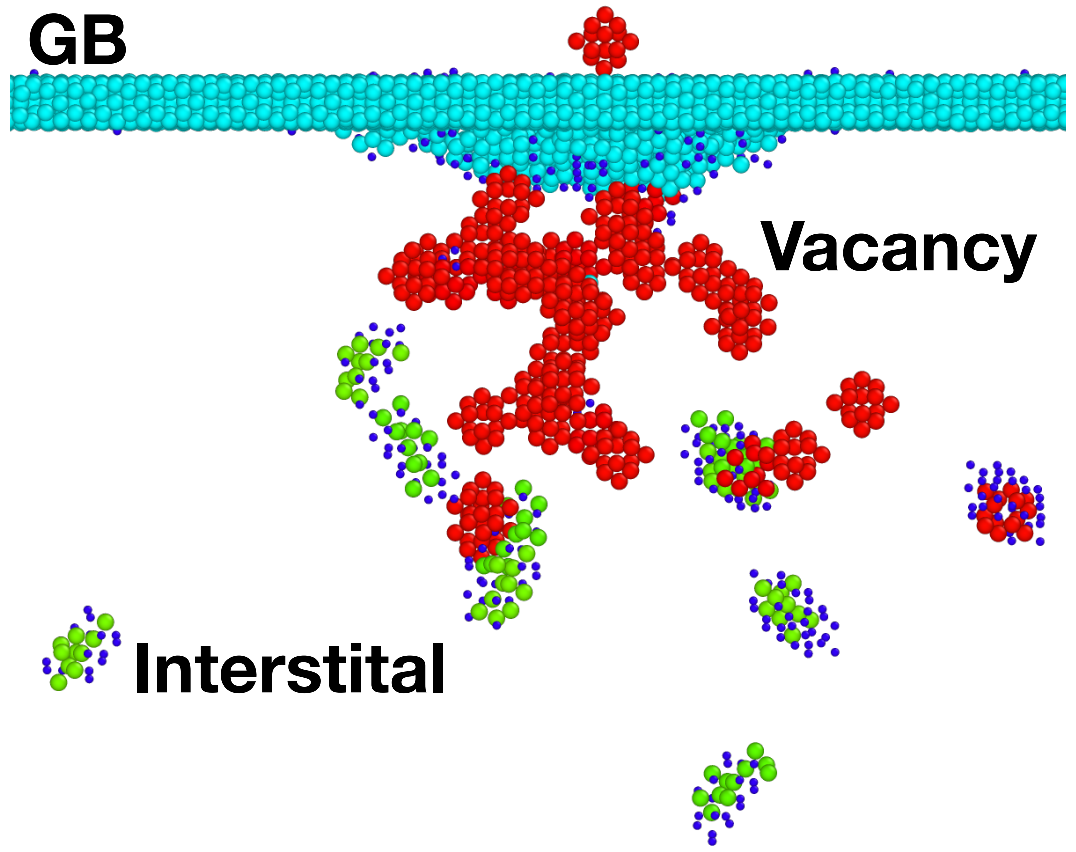
"No parallel atomic structure generation code exists."

"No detail control of polycrystalline structure."

"LAMMPS community discarded parallel I/O system."

"No parallel post-processing code exists."

Massive MD: Future

	Atomic structure generation	MD simulation	Post-processing
	PolyPal (published)	*VITMAS* (developed)	*Untitled* (in progress)
Parallel codes			

- Integrated parallel I/O framework
- Efficient memory management
- GB generation and appropriate defect analysis

\+ Multi-PKA (Primary Knock-on Atom) simulation feature

PolyPal

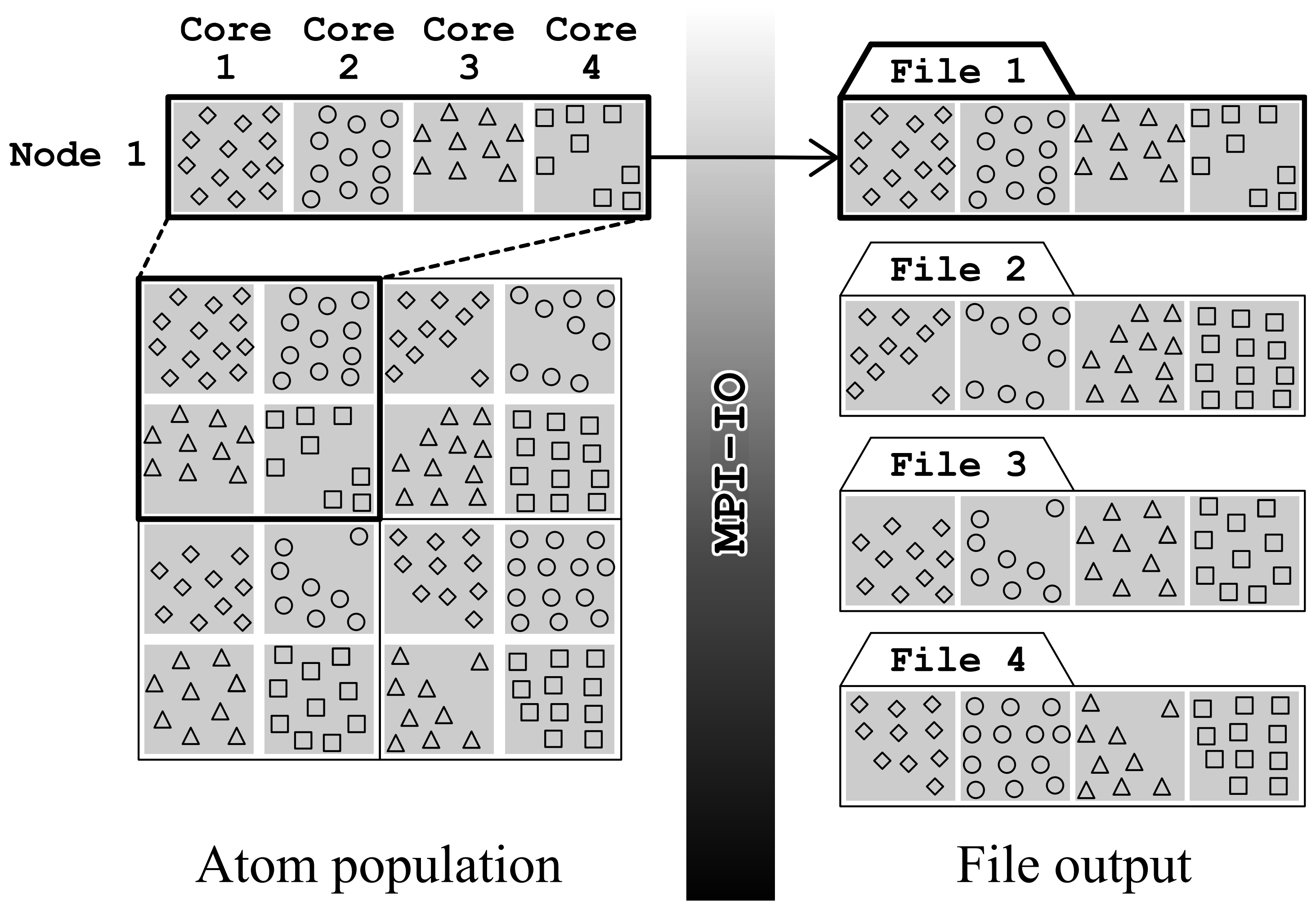
Parallel I/O

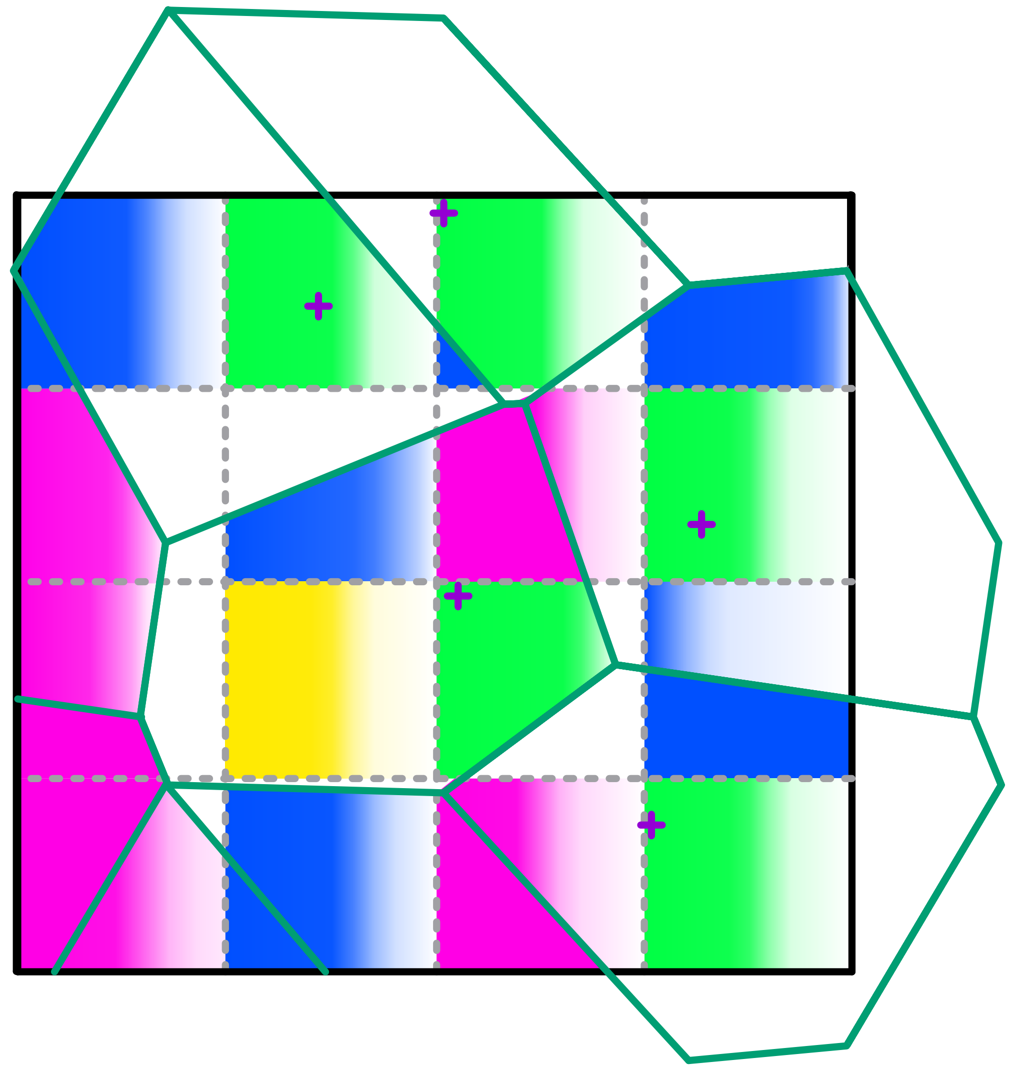
Parallelised task without
inter-core communication

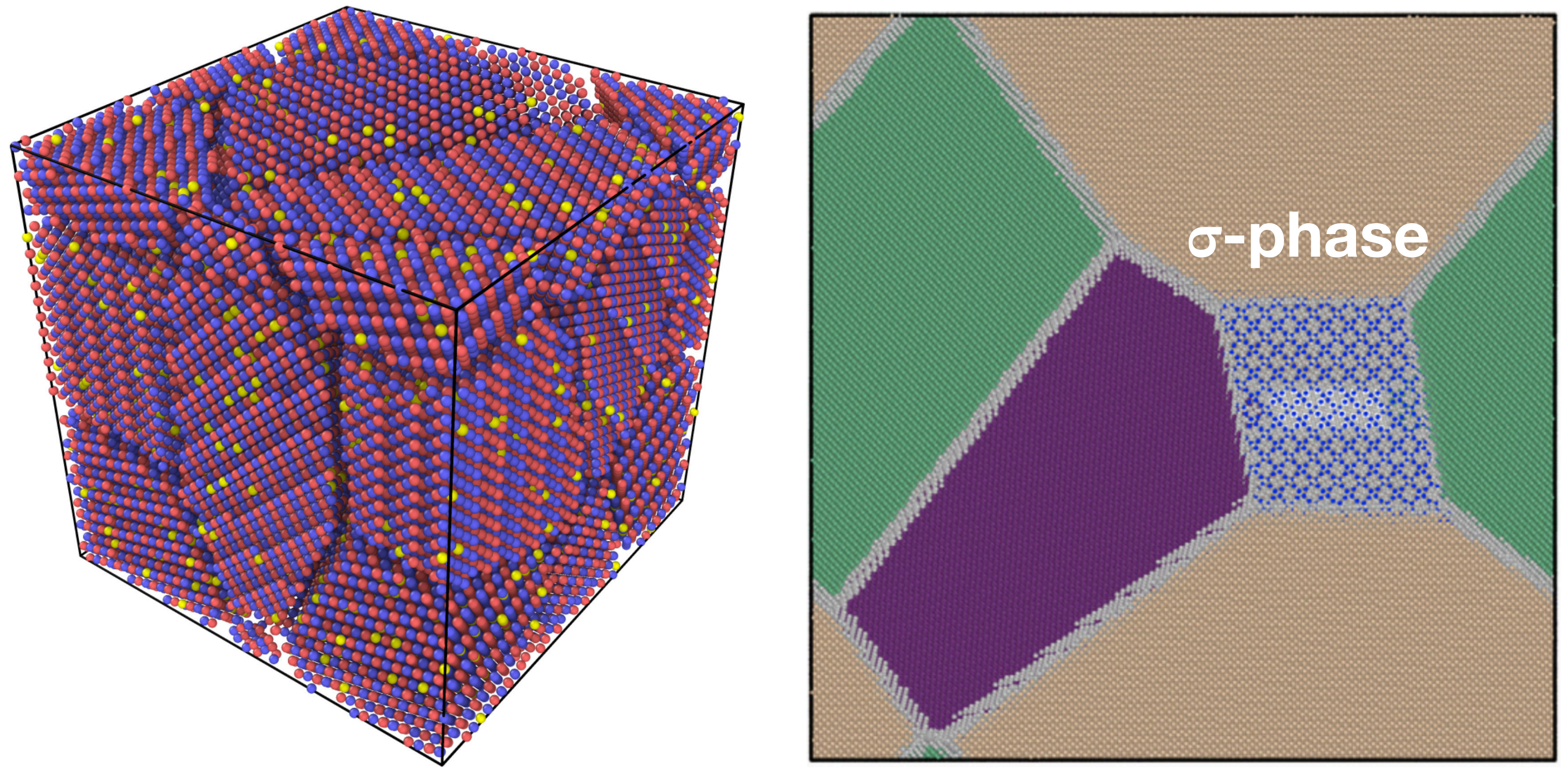
Versatile options

PolyPal: Features

Parallel I/O	Parallelised task without inter-core communication	Versatile options
- Fast - Memory efficient - Domain-preserving →Elimination of atom sorting	- Fast - Memory efficient	- Grain orientation control - Grain morphology control - Multi-phase - Species substitution







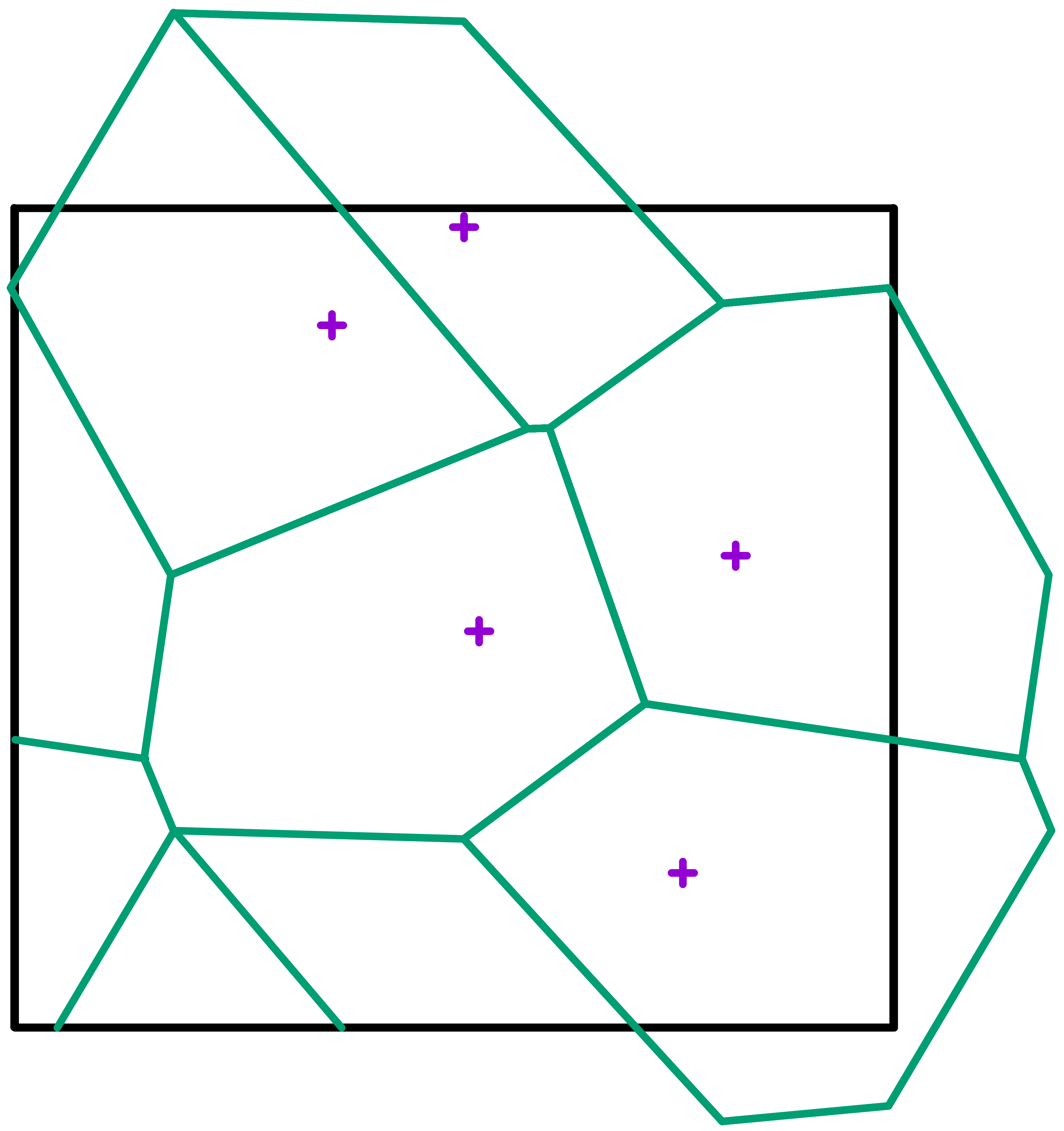
PolyPal: Features

File Format

	LAMMPS	DL_POLY	*VITMAS*
ASCII text	✓	✓	✓
Binary	✗	✗	✓
Mutil-flle	✓	✗	✓

PolyPal: Features

Structure Design: Grain Morphology



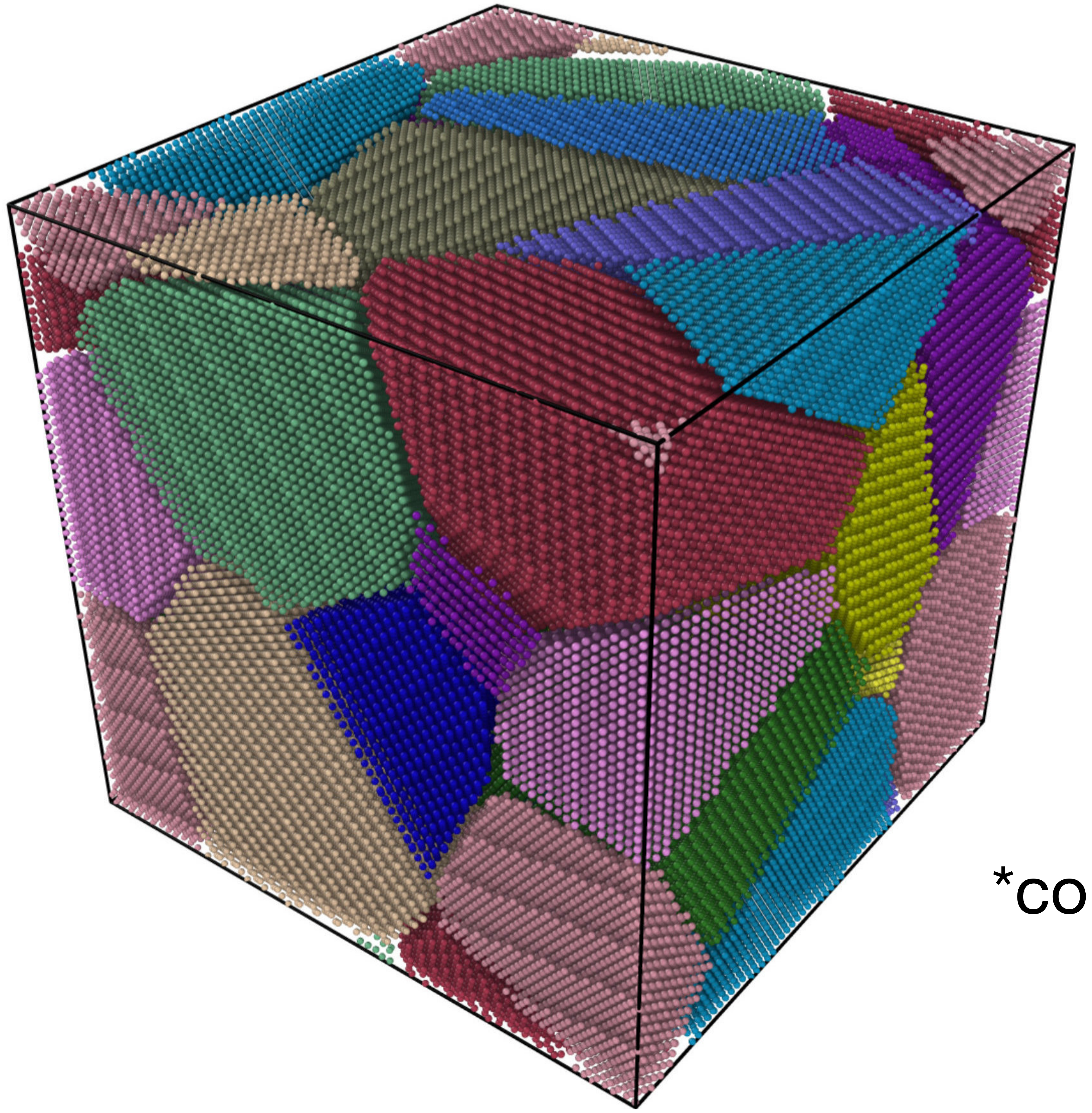
Seed distribution

=

Grain morphology

PolyPal: Features

Structure Design: Grain Morphology

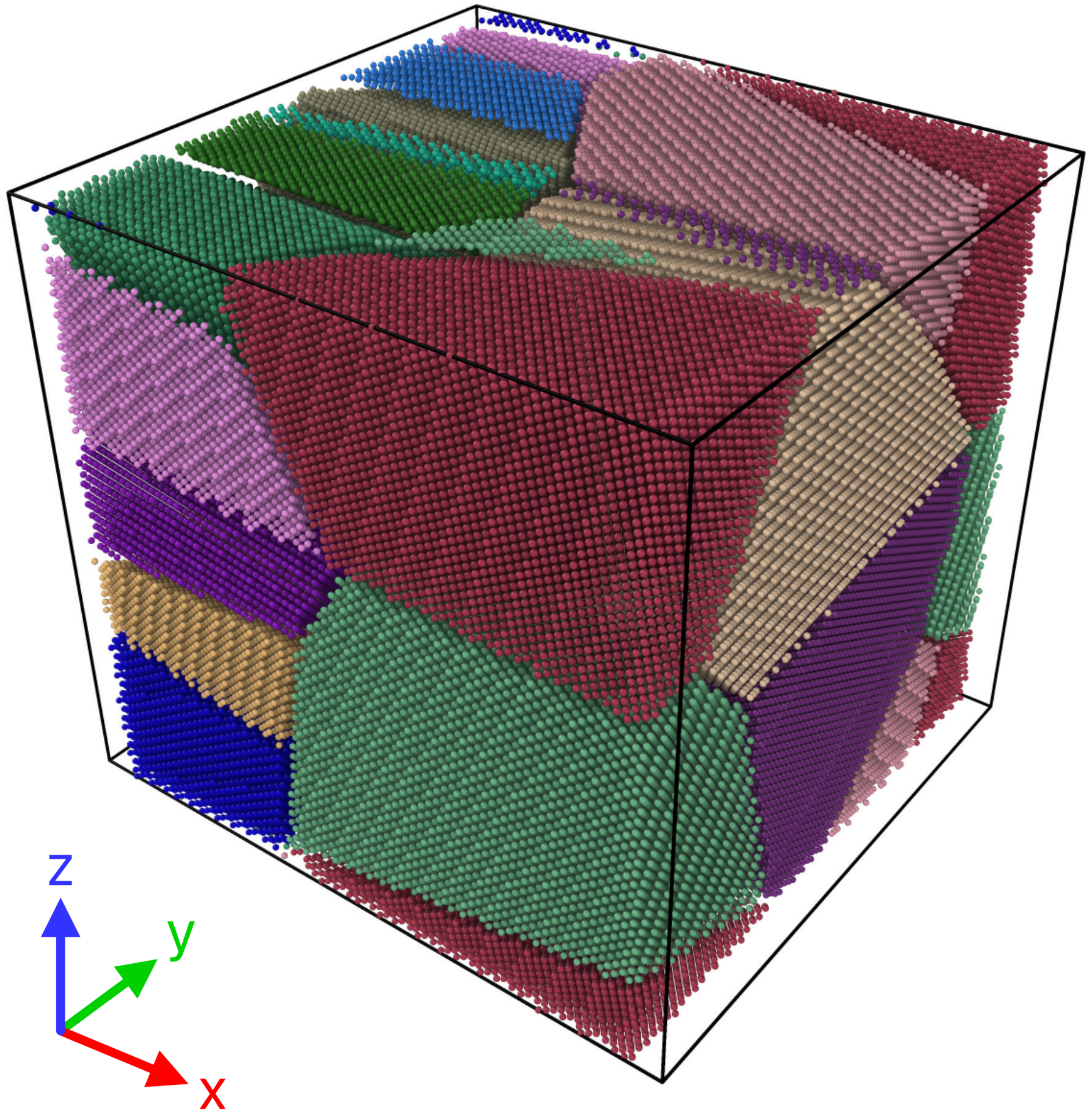


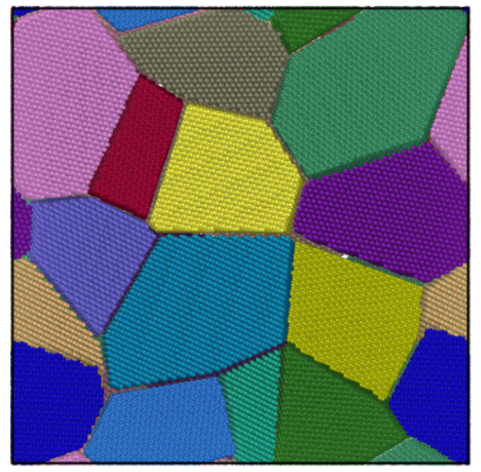
*coloured by lattice orientation

Uniformly and randomly distributed

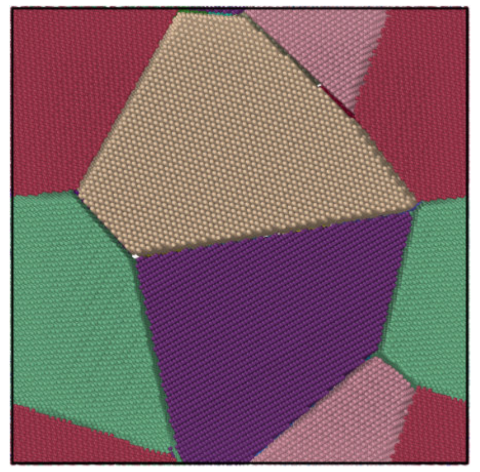
PolyPal: Features

Structure Design: Grain Morphology





Bottom



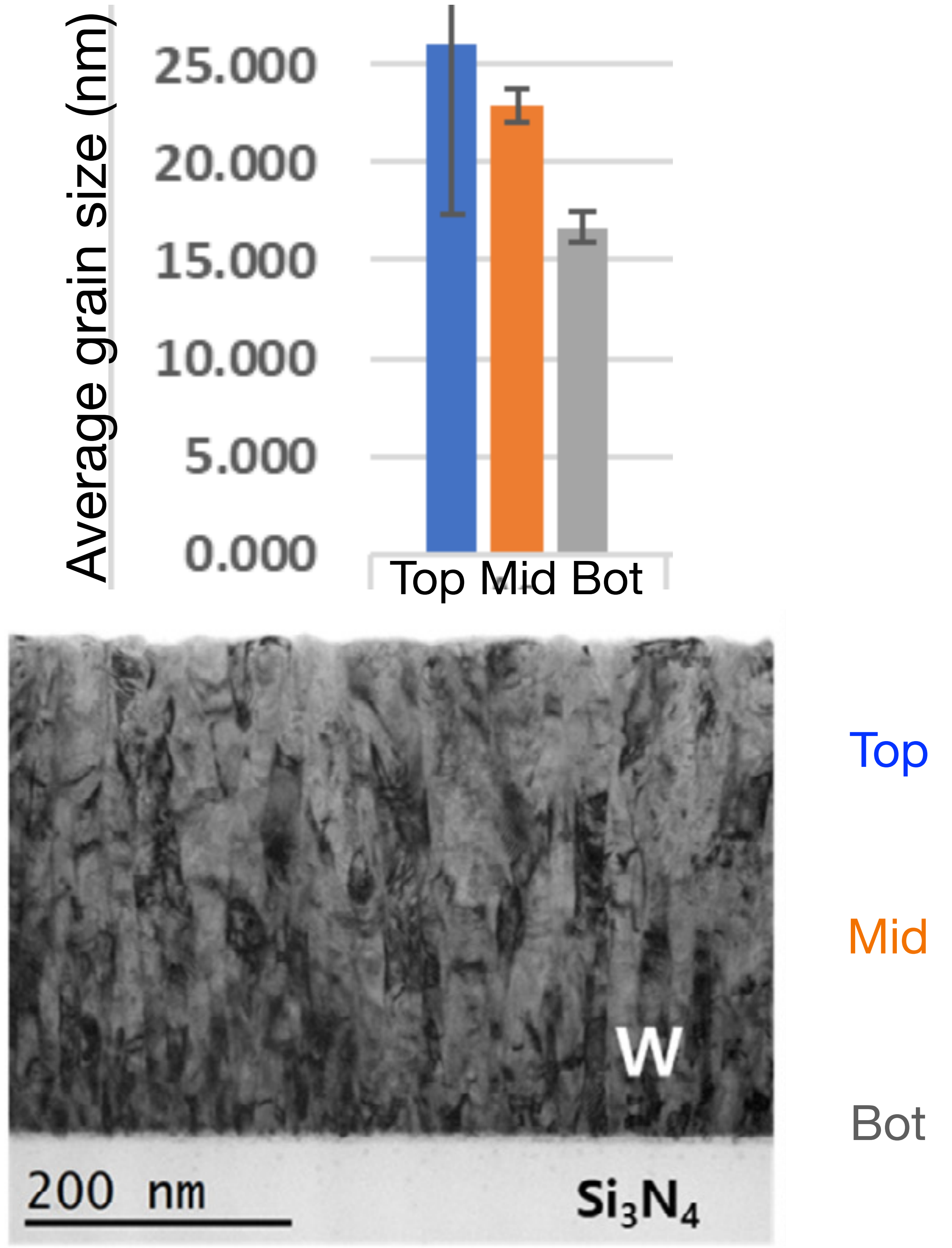
Top

Layered structure with free surfaces

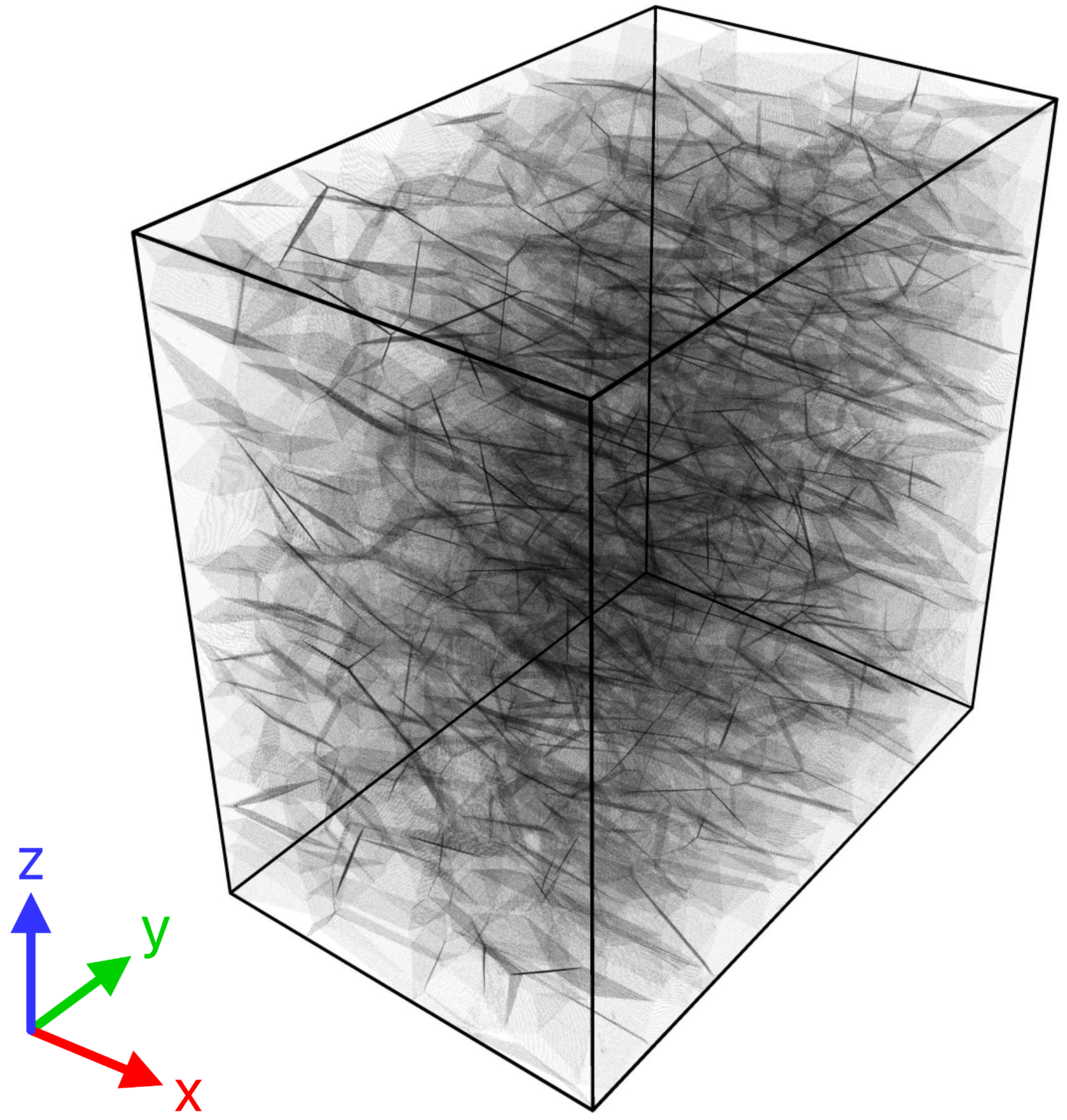
PolyPal: Features

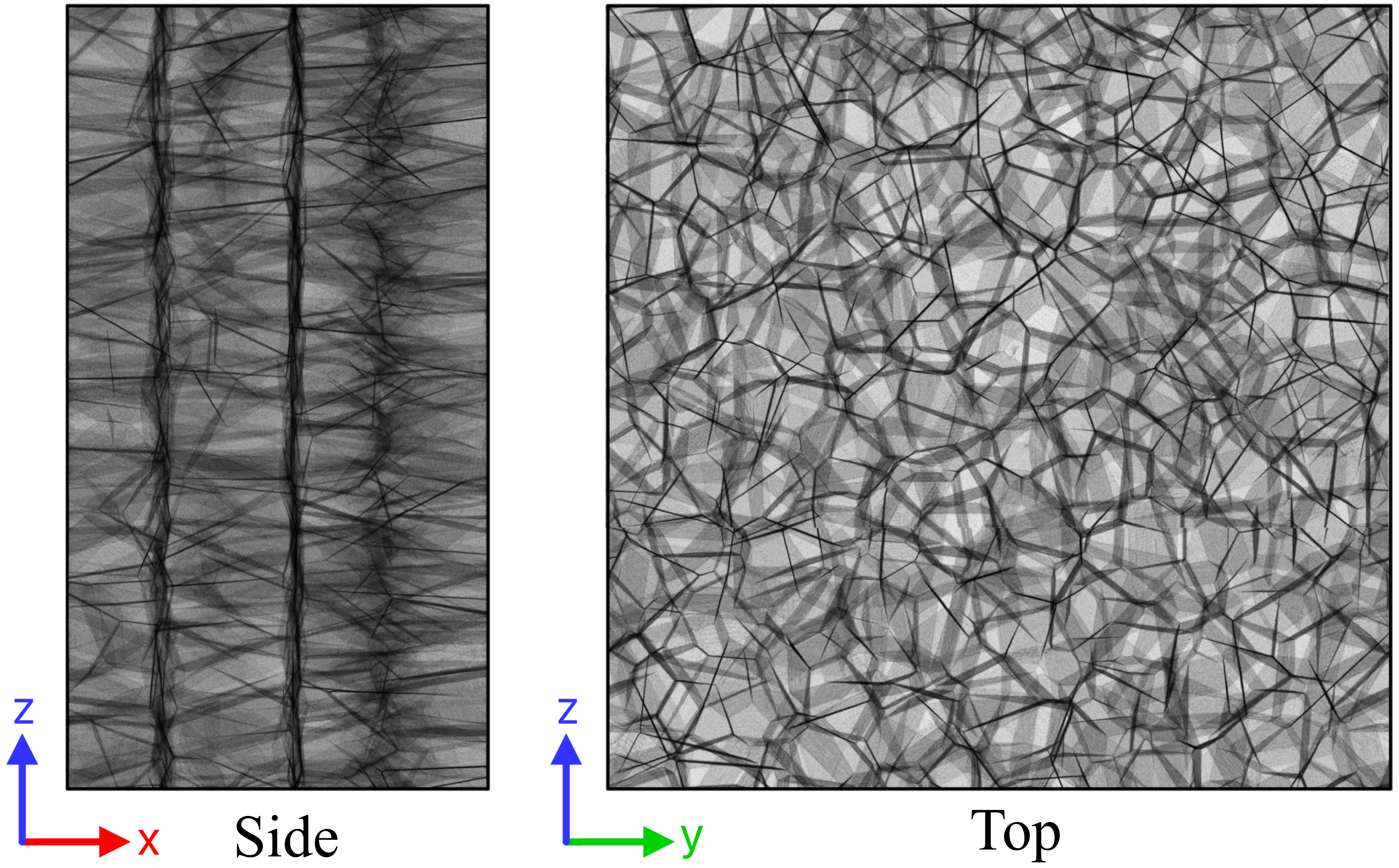
Structure Design: Grain Morphology

*Only non-crystalline atoms are visualised.



Oh et al., Extreme Mech. Lett. (2020)

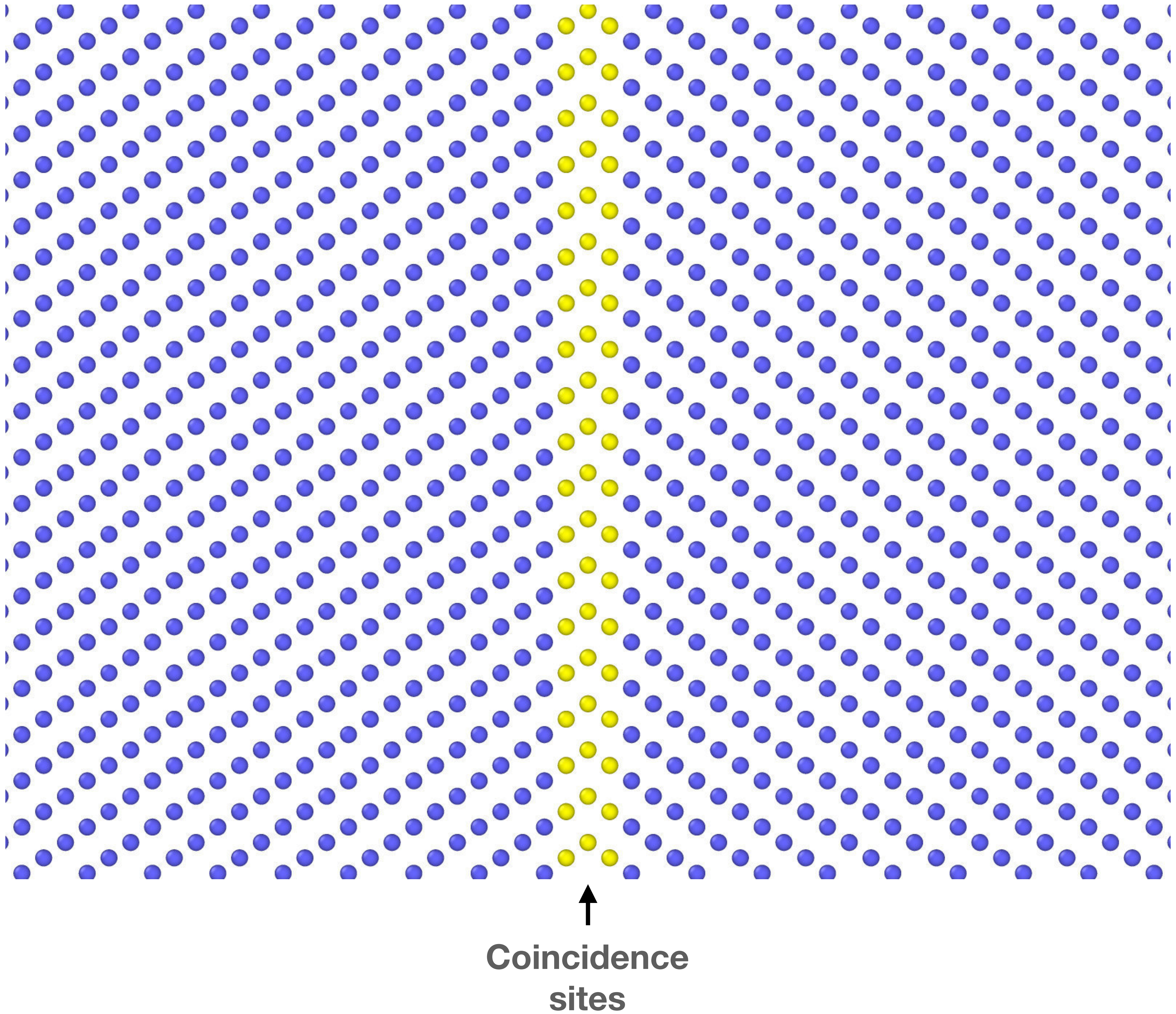




Massive virtual test specimen (~8 billion atoms)

PolyPal: Features

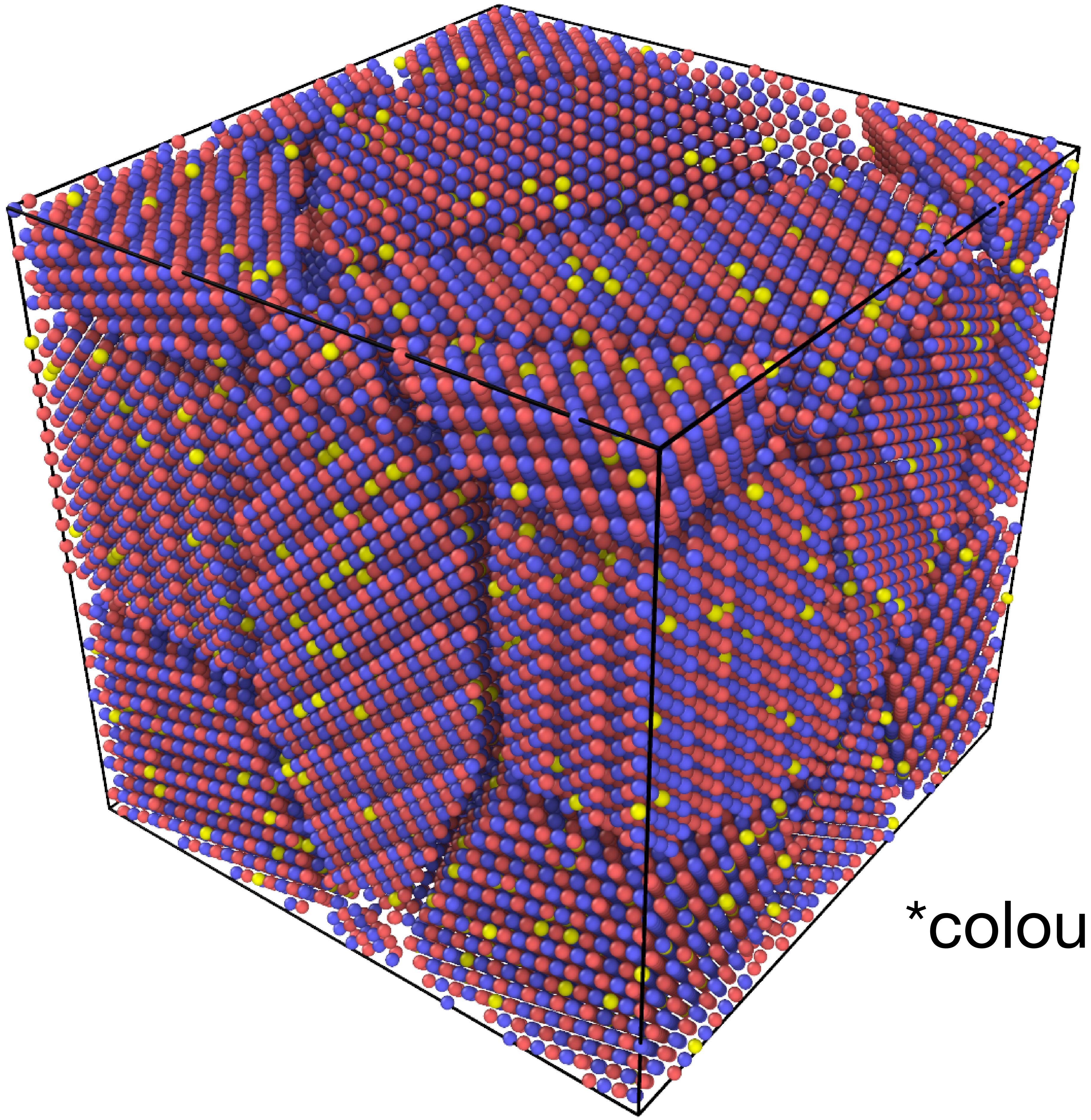
Structure Design: Grain Morphology



Bicrystal: bcc Σ3(112) symmetric tilt

PolyPal: Features

Structure Design: Texture

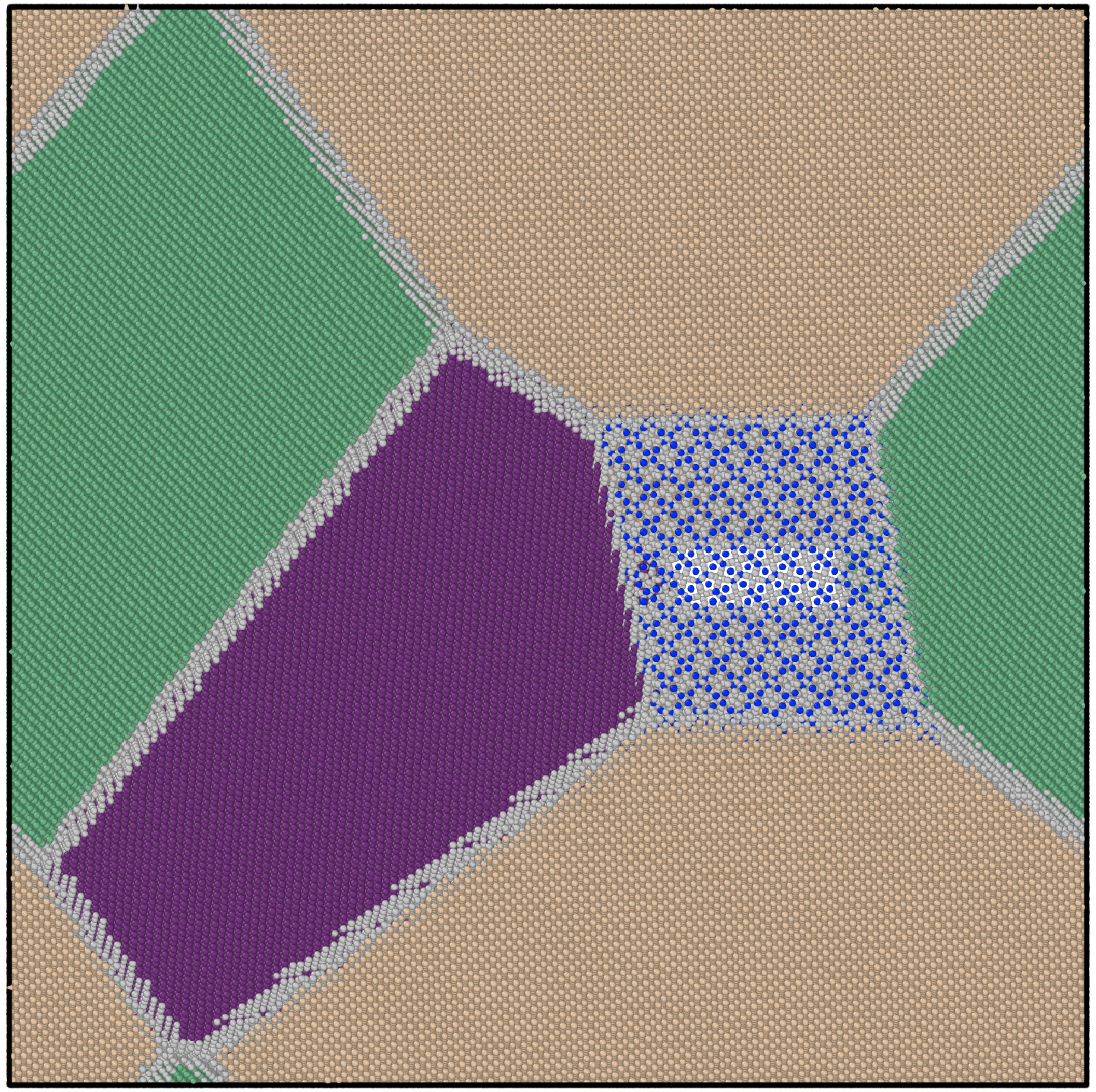


*coloured by atom species

Equiatomic fcc with solute atoms

PolyPal: Features

Structure Design: Texture



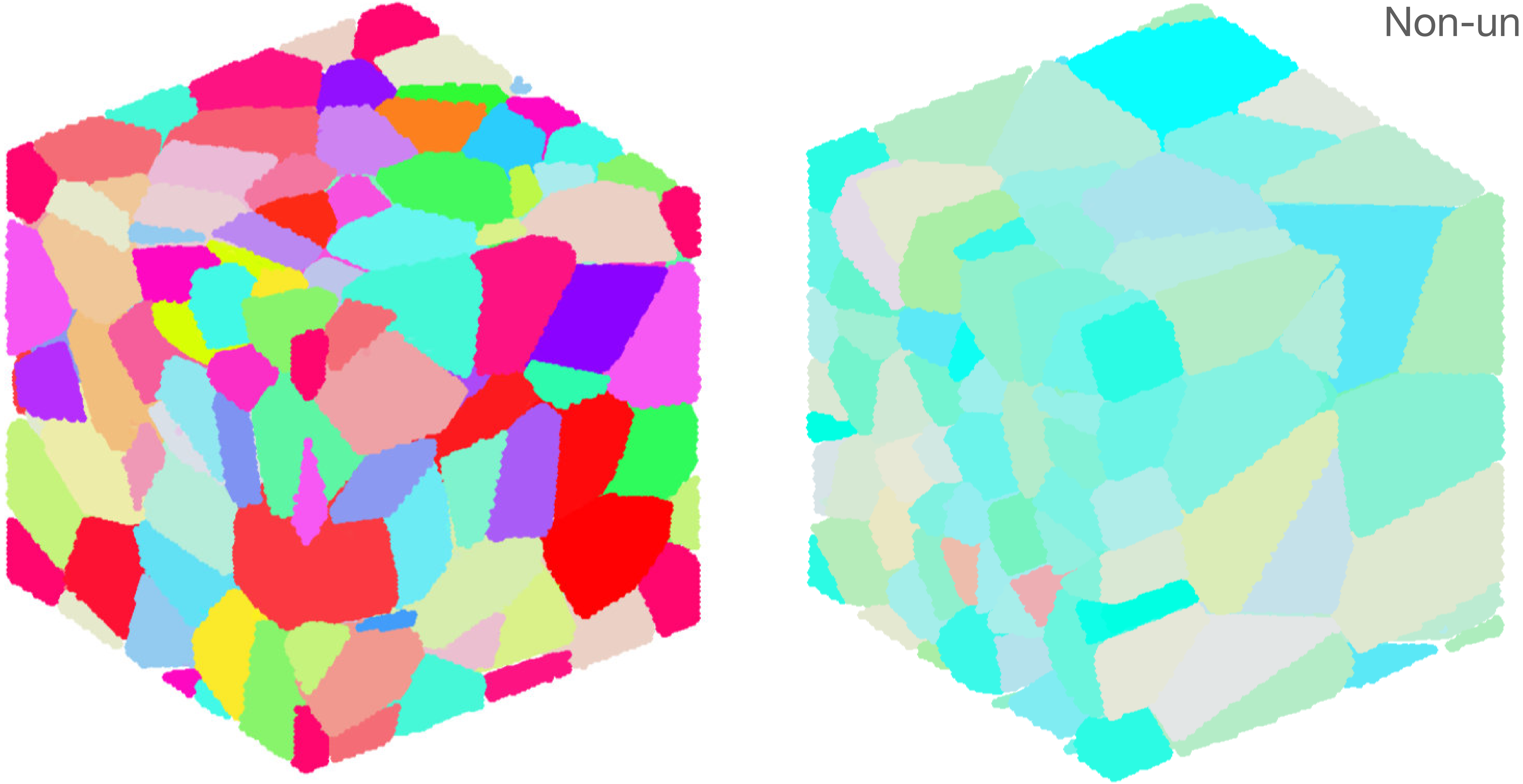
Multiple unit cells

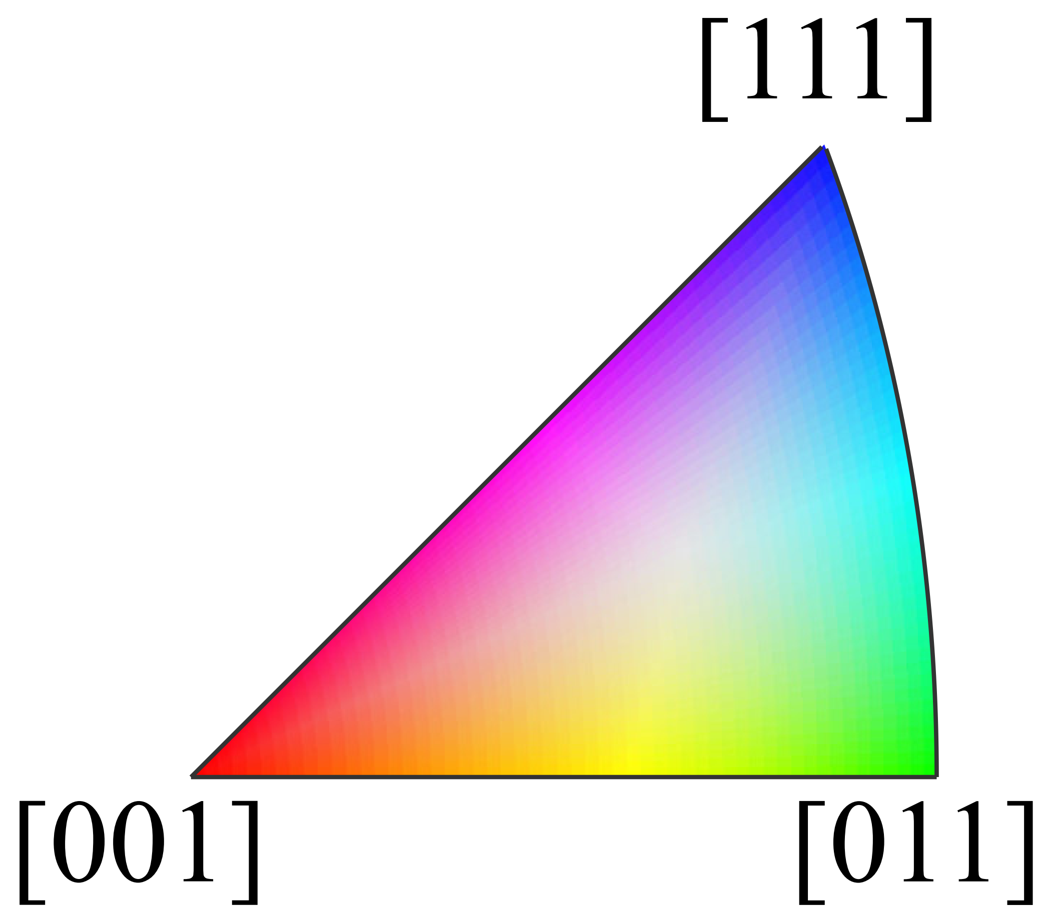
σ-phase precipitates

PolyPal: Features

Structure Design: Texture

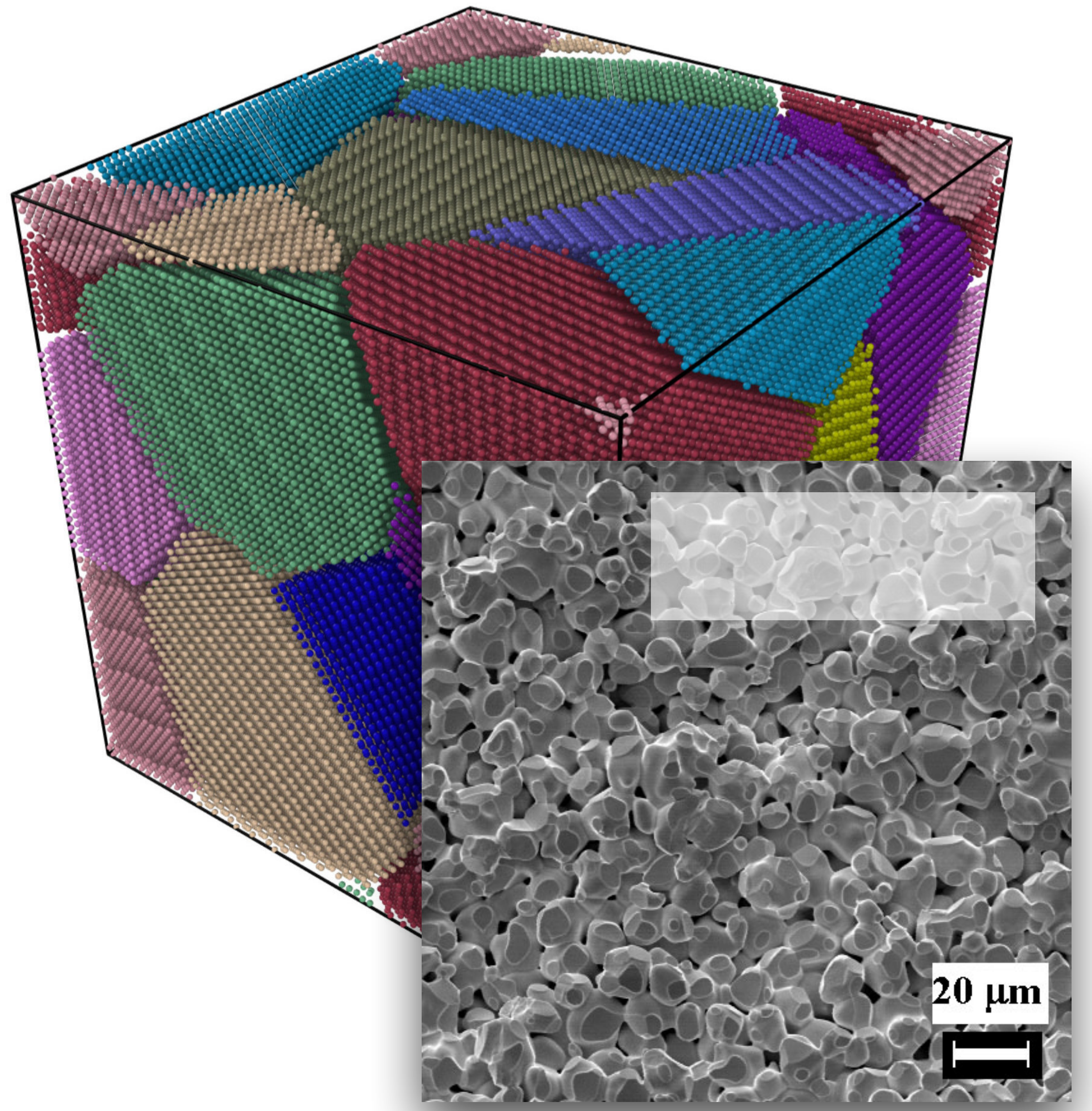
Biased crystallographic direction
Non-uniform grain size distribution



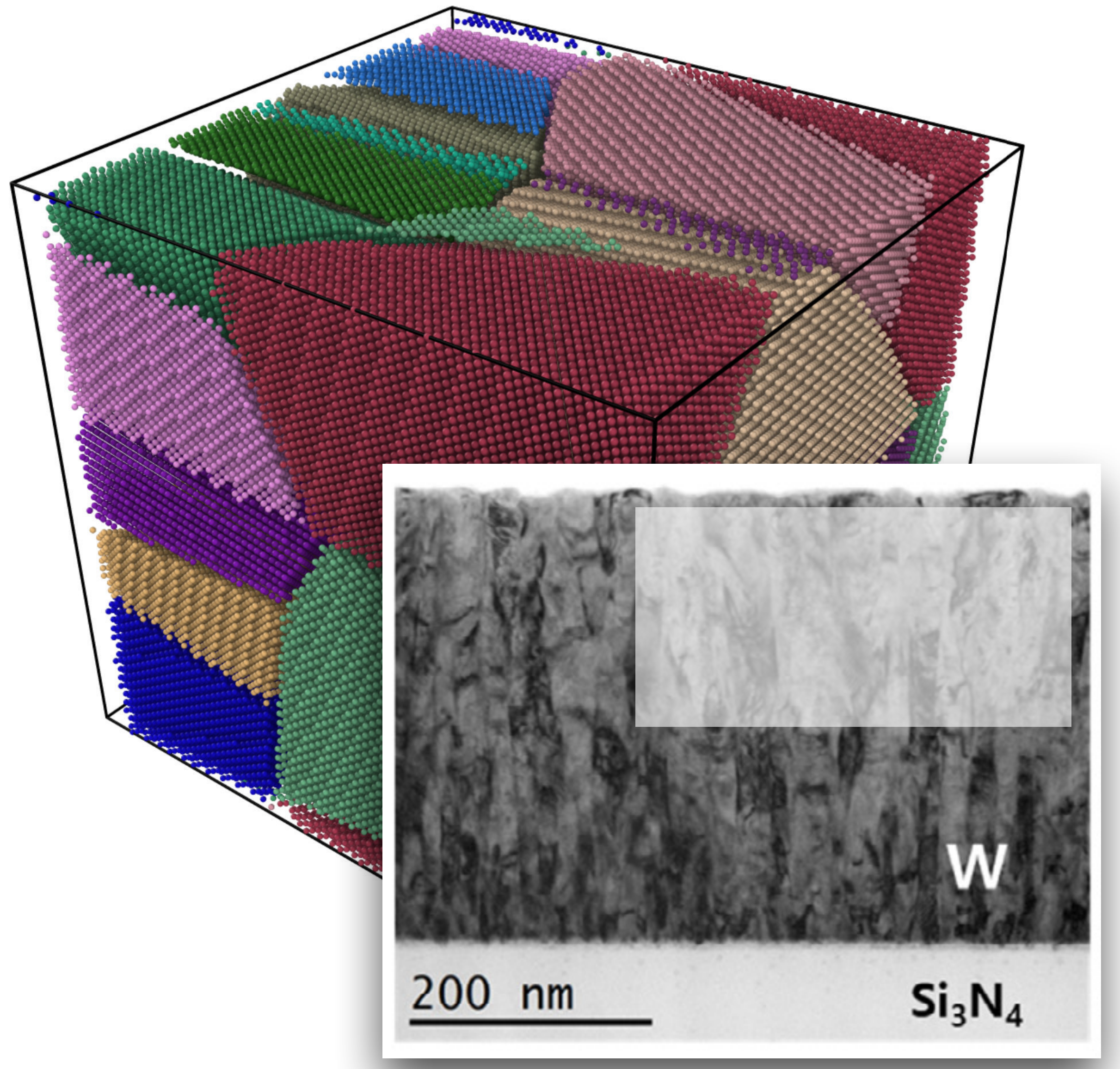


Isotropic versus anisotropic

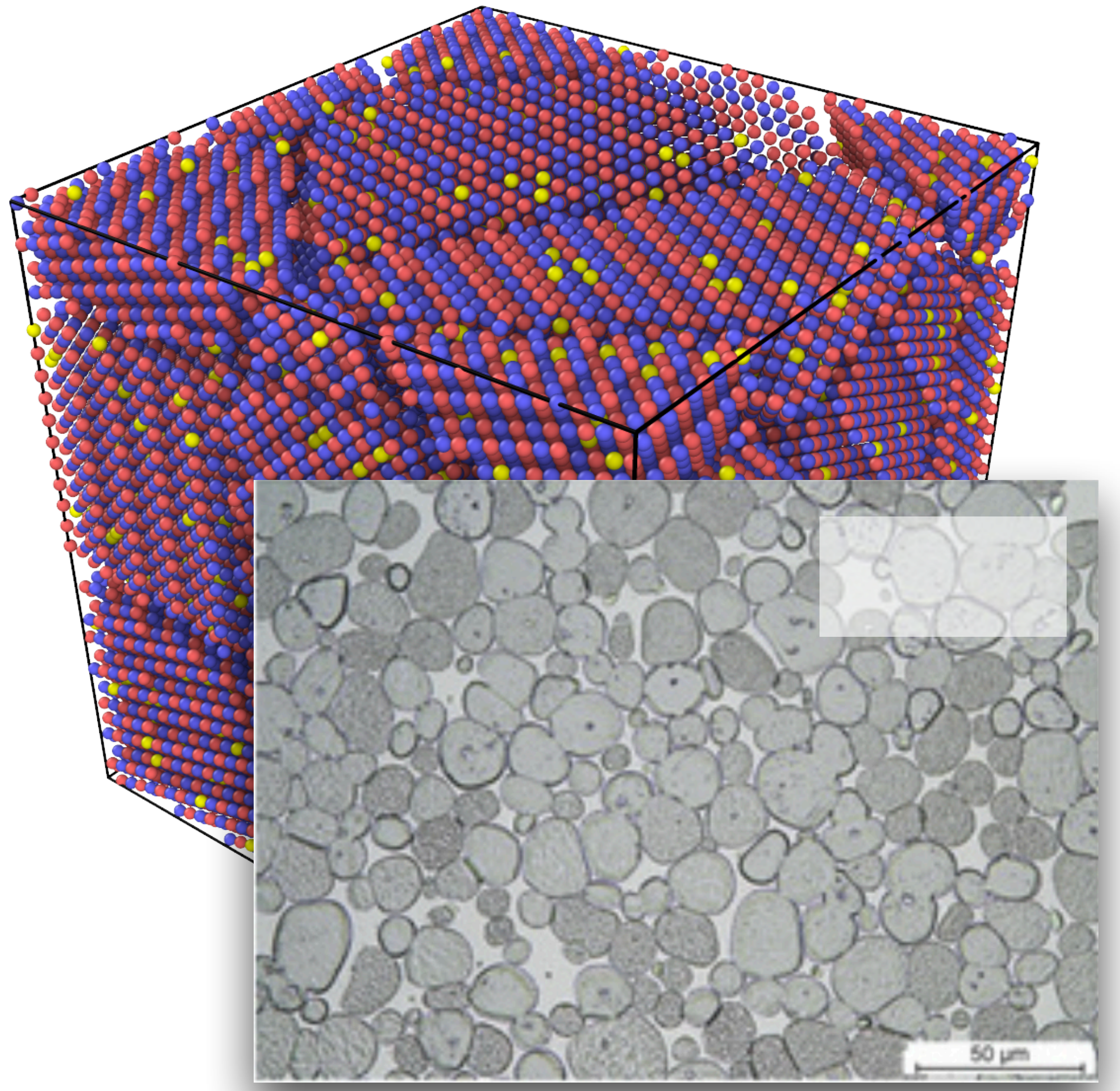
PolyPal Group Photo: Applications



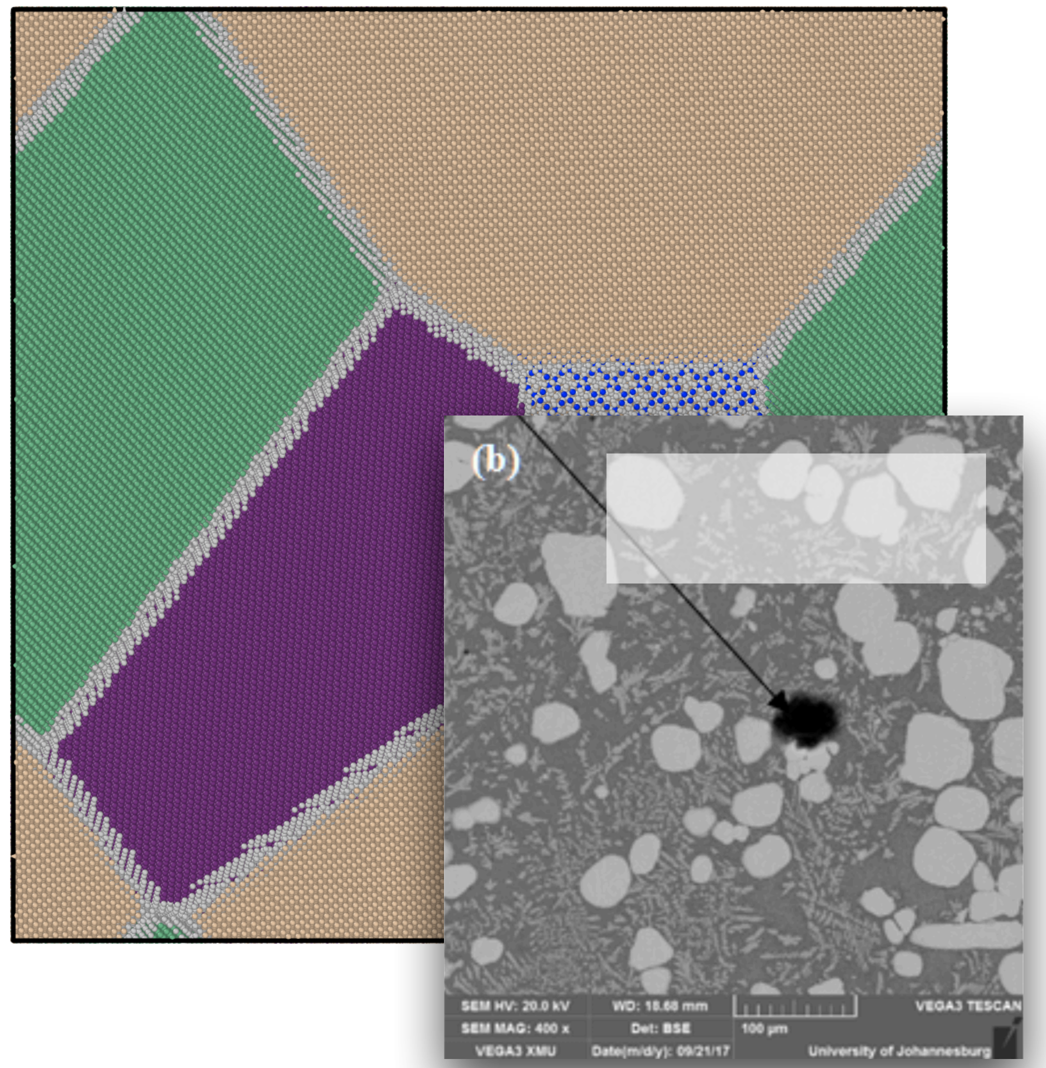
Grabis et al., Proc. Est. Acad. Sc Eng. (2004)



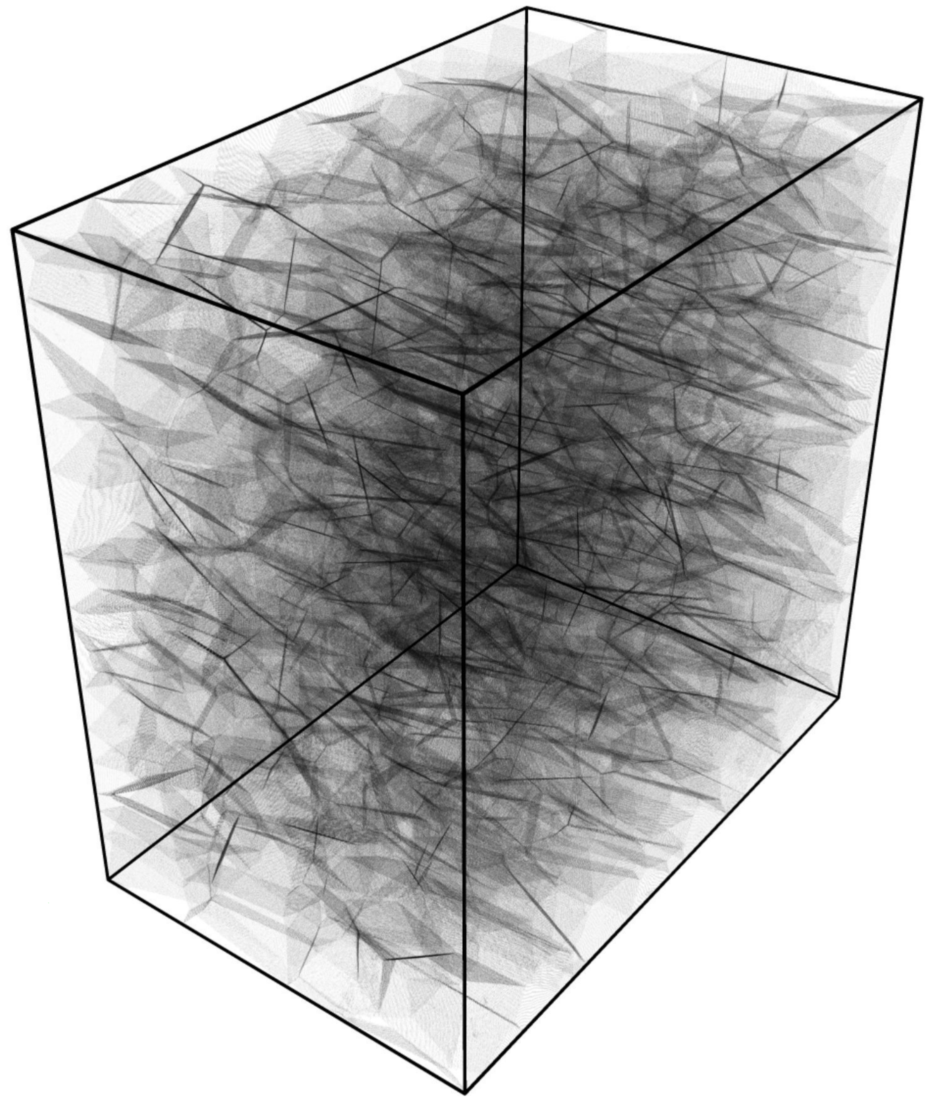
Oh et al., Extreme Mech. Lett. (2020)

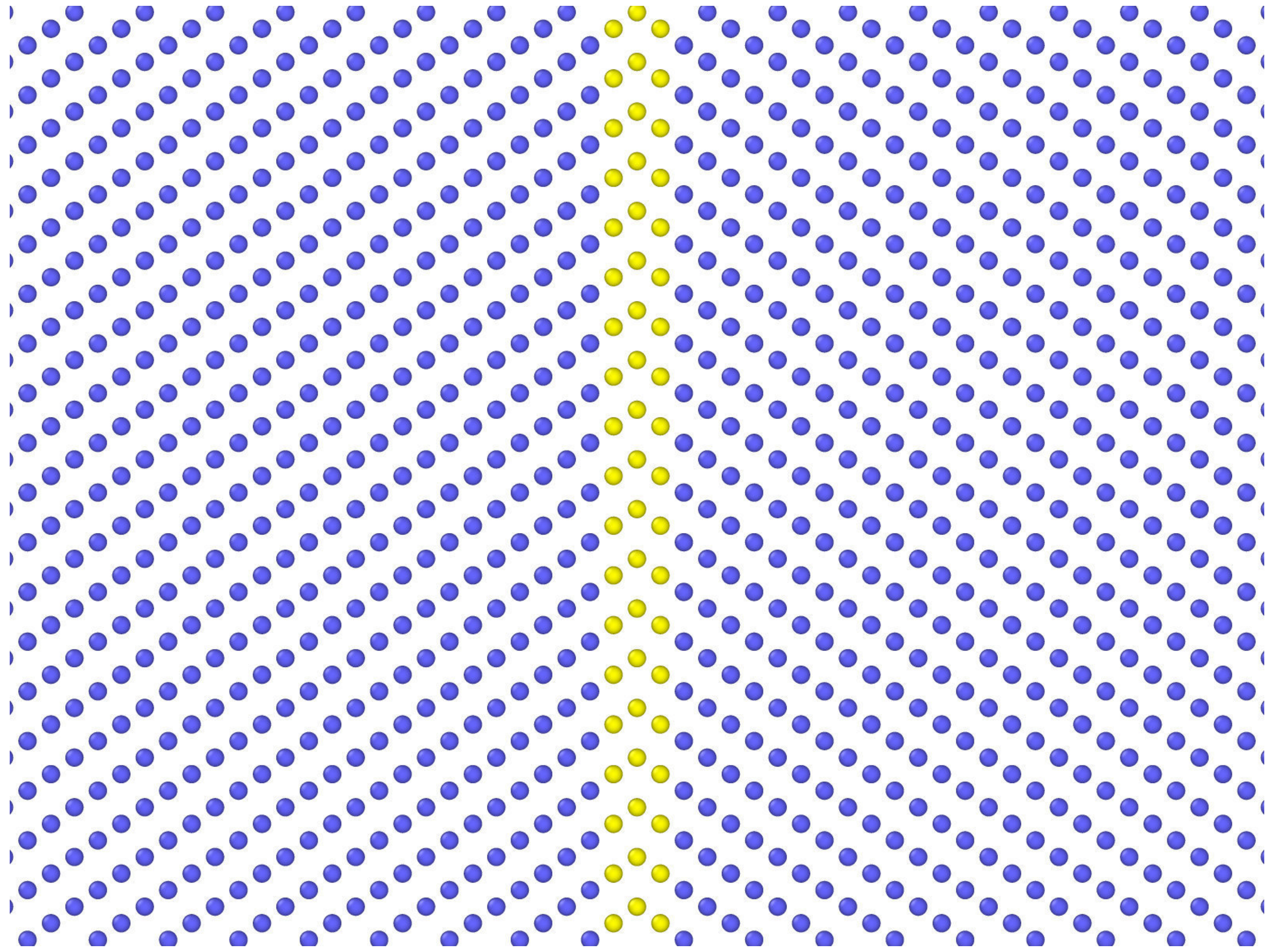


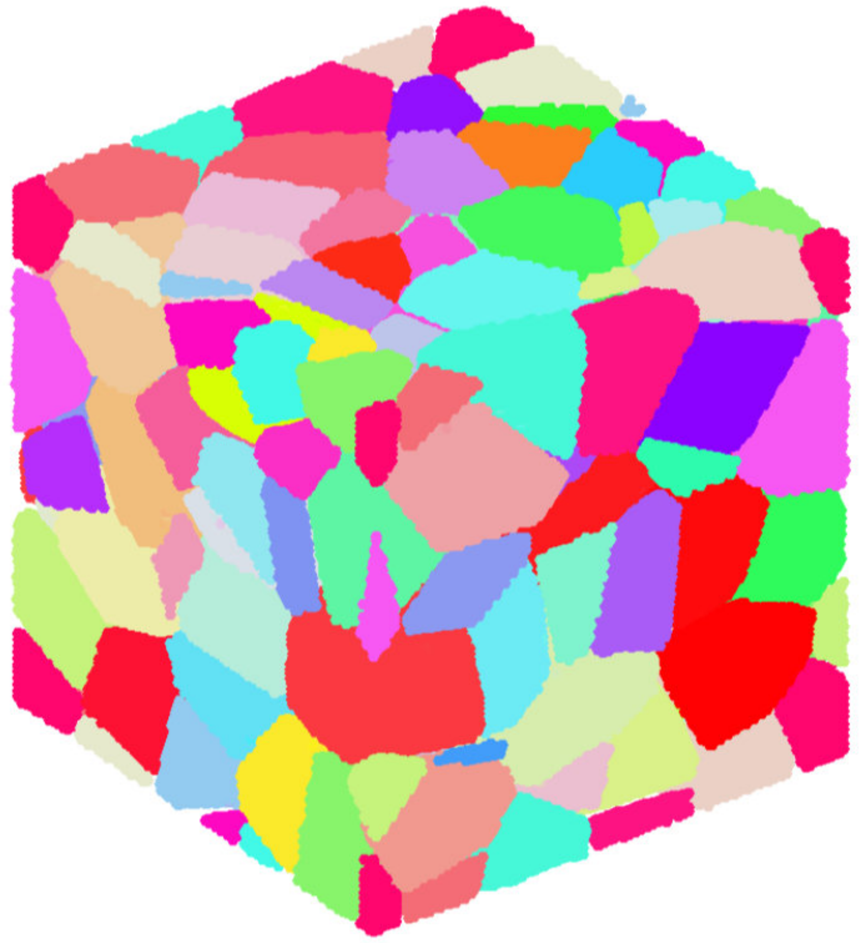
Skoczylas et al., Materials (2021)



Skoczylas et al., Materials (2021)



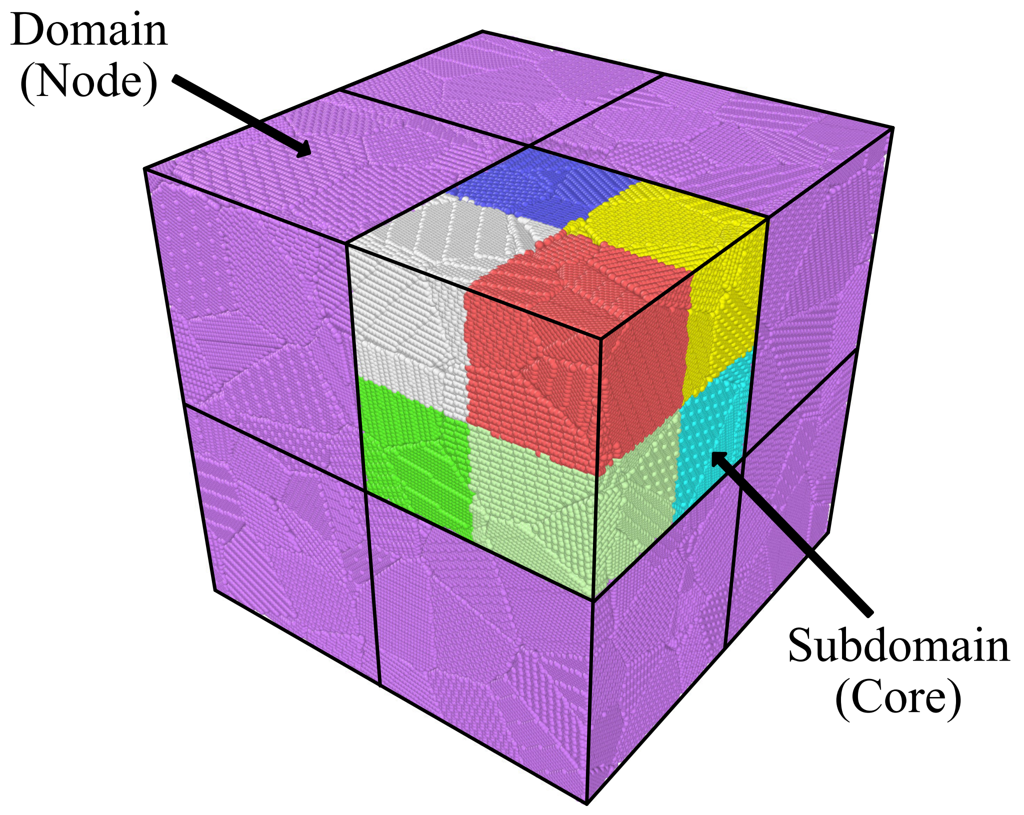




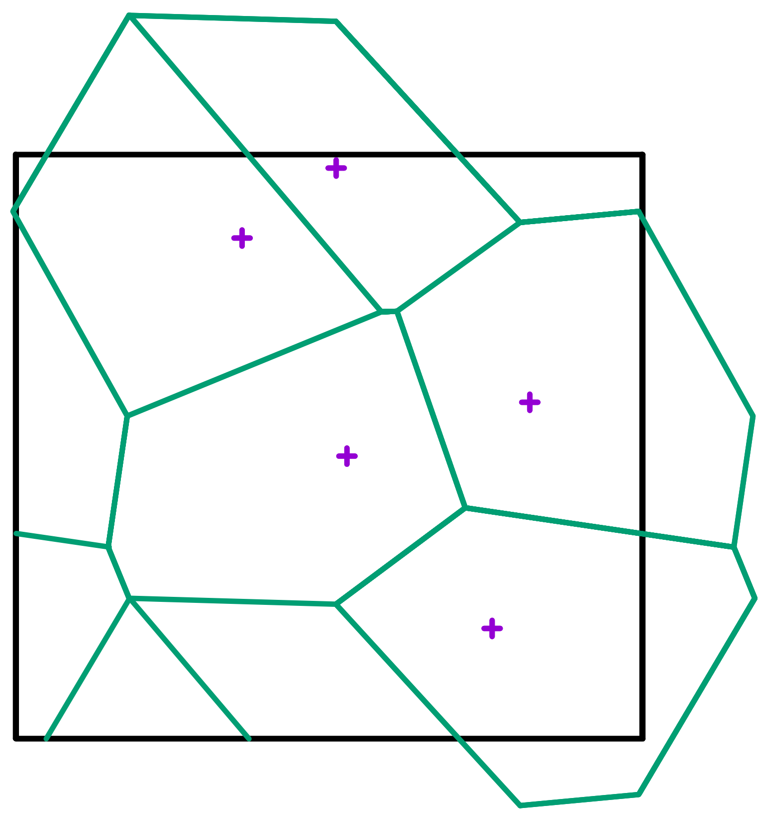
PolyPal

Workflow Overview

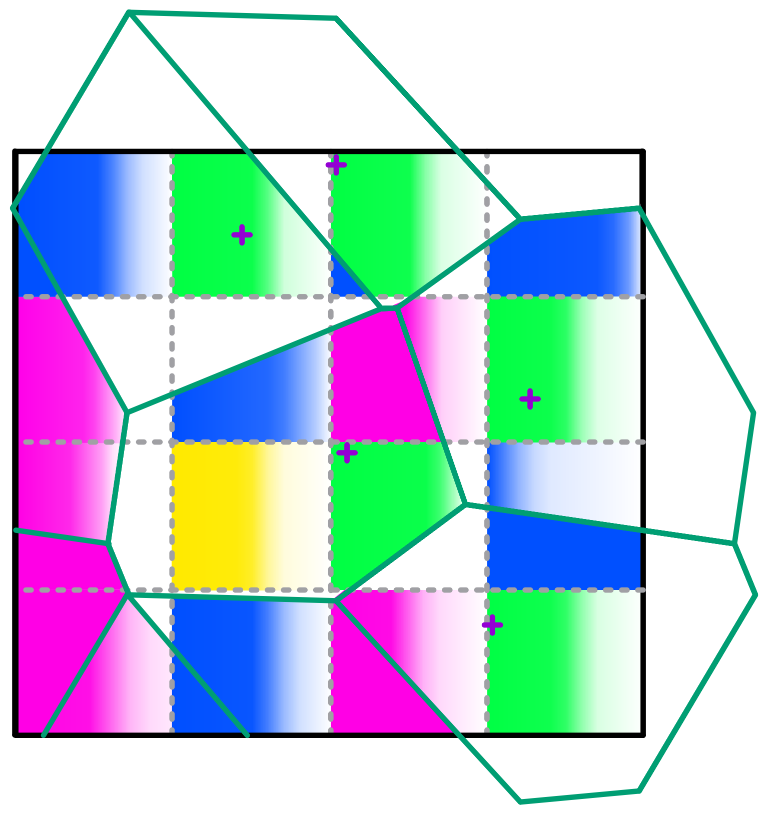
(1) Domain decomposition



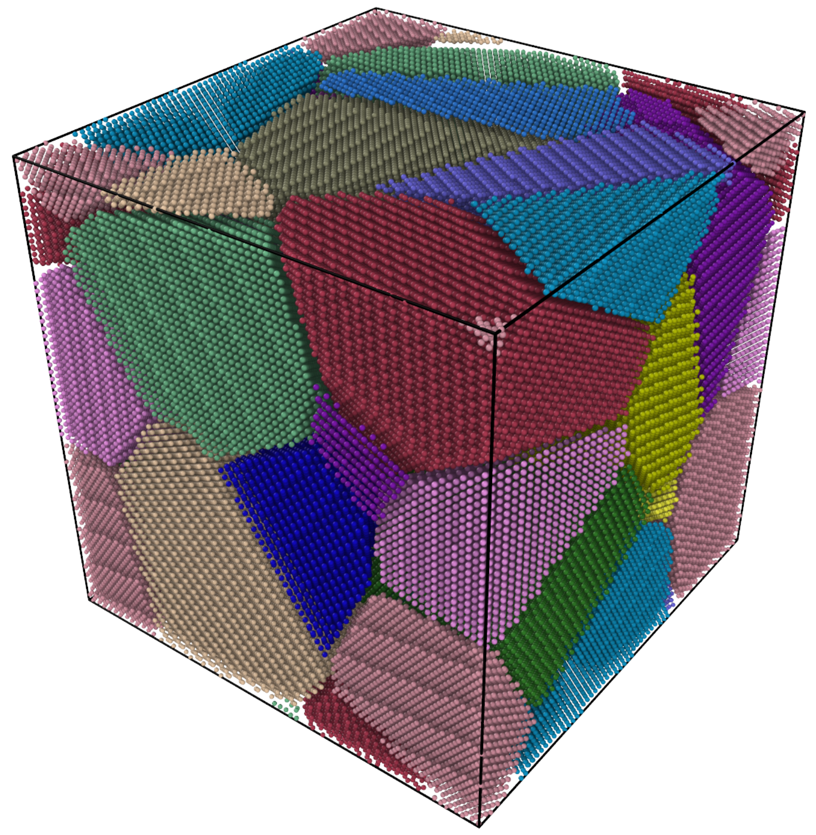
(2) Grain population



(3) Grain distribution & atom filling

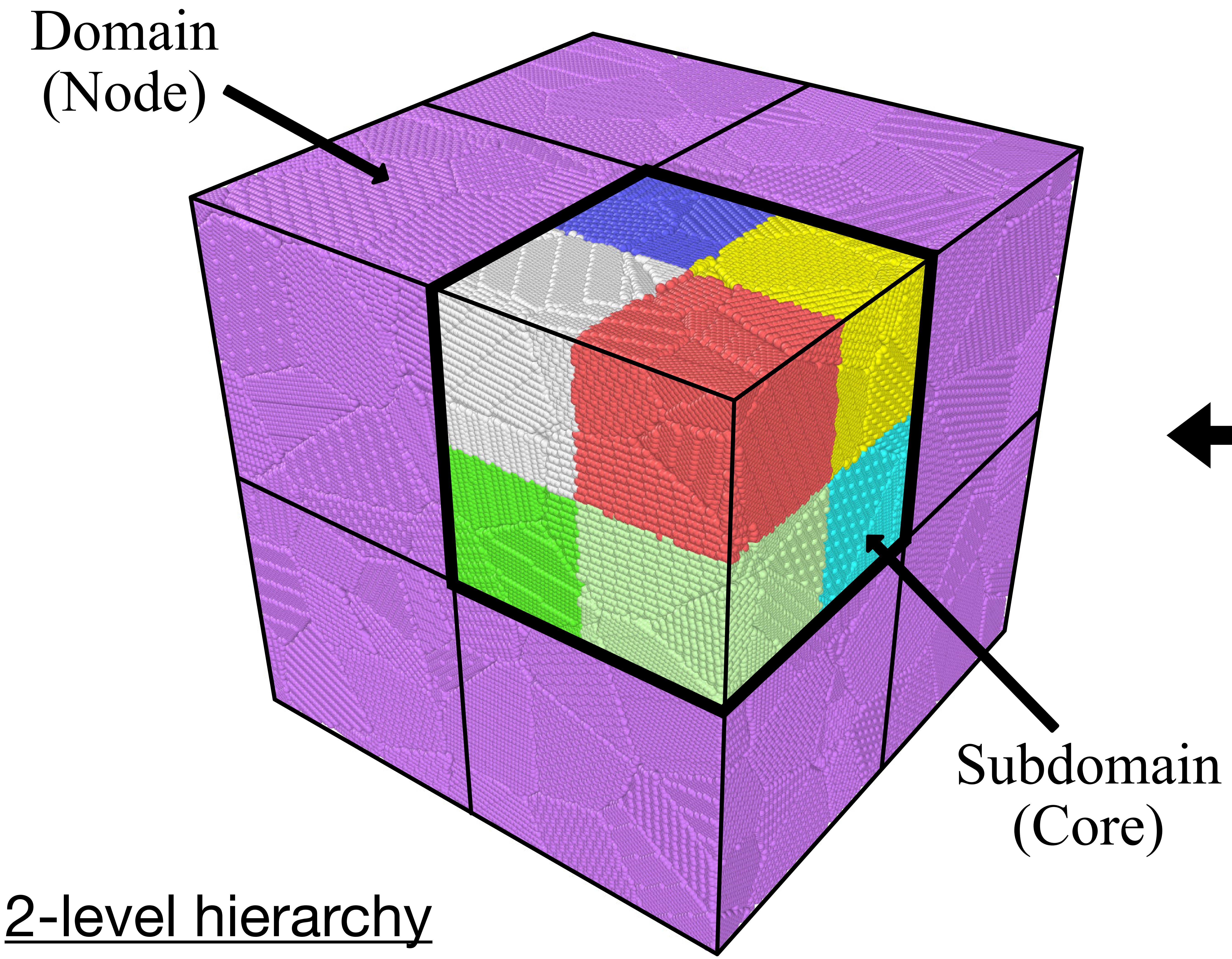


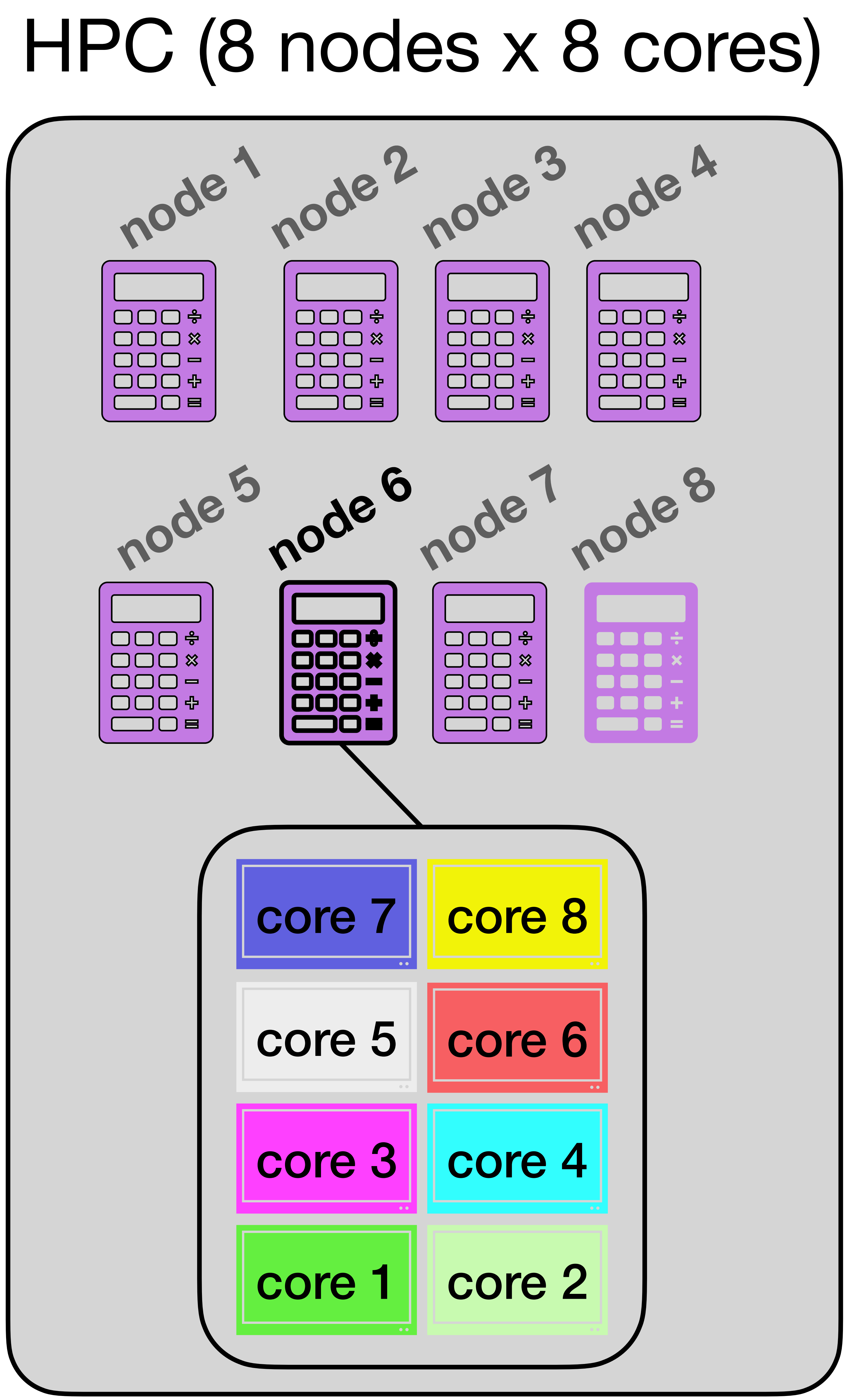
(4) File output



PolyPal

Domain Decomposition

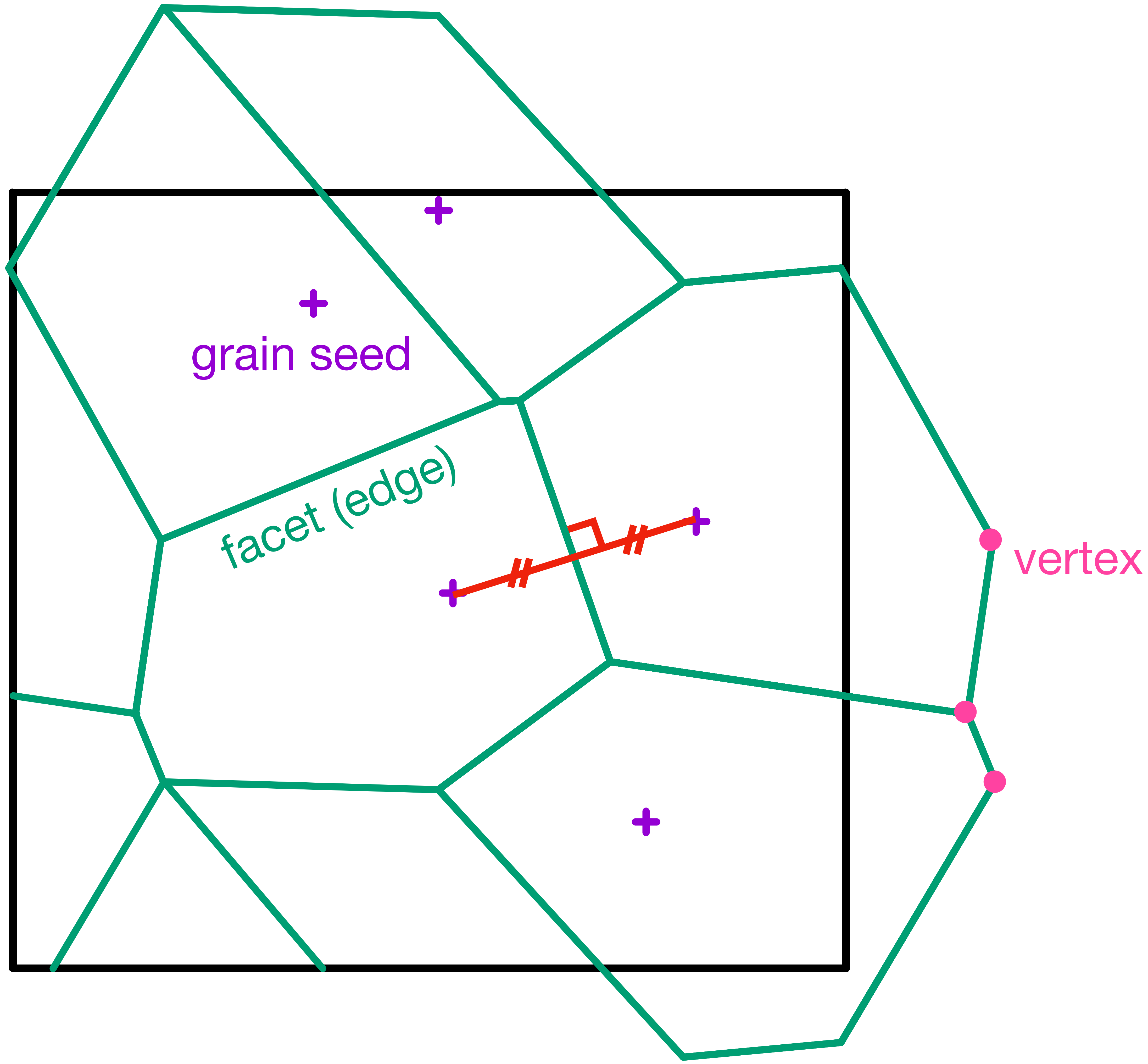




2-level hierarchy

PolyPal

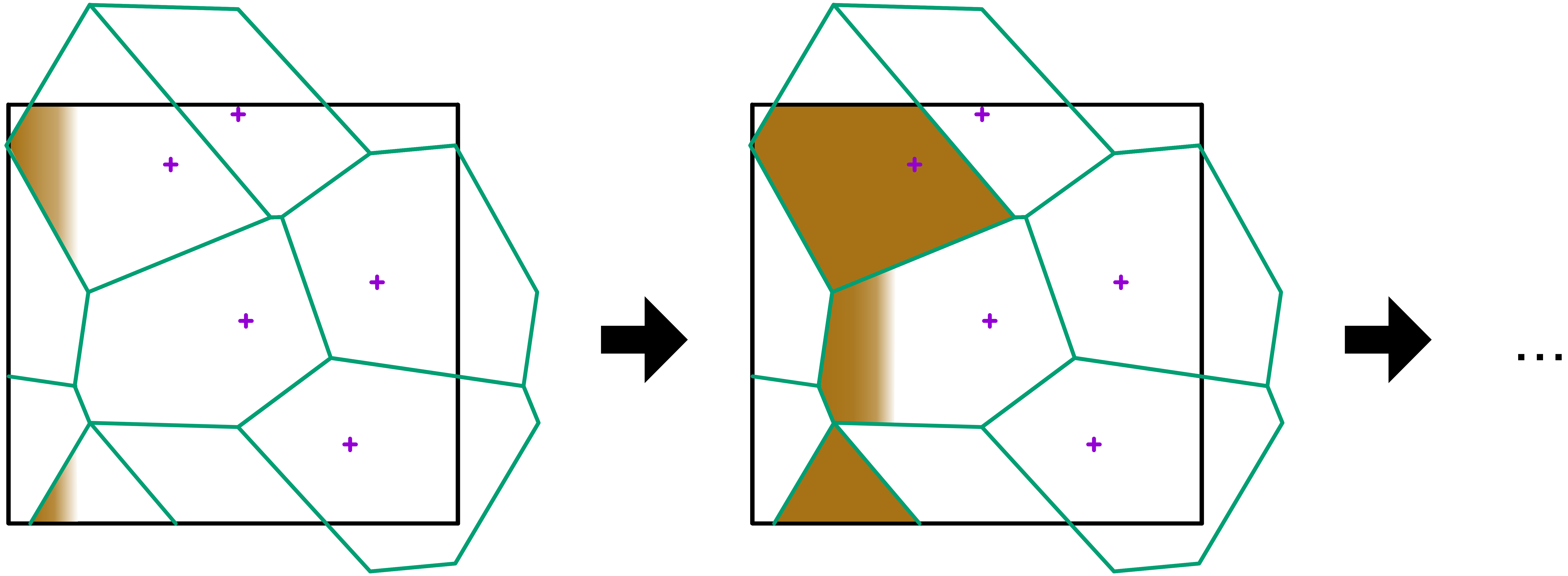
Grain Population



- **Centroidal Voronoi tessellation**
- **Grains constructed from**
    - seeds with crystal orientation
- **Grain information**
    - crystallographic orientation
    - seed points
    - vertices
    - edges
    - list of neighbour grains

PolyPal

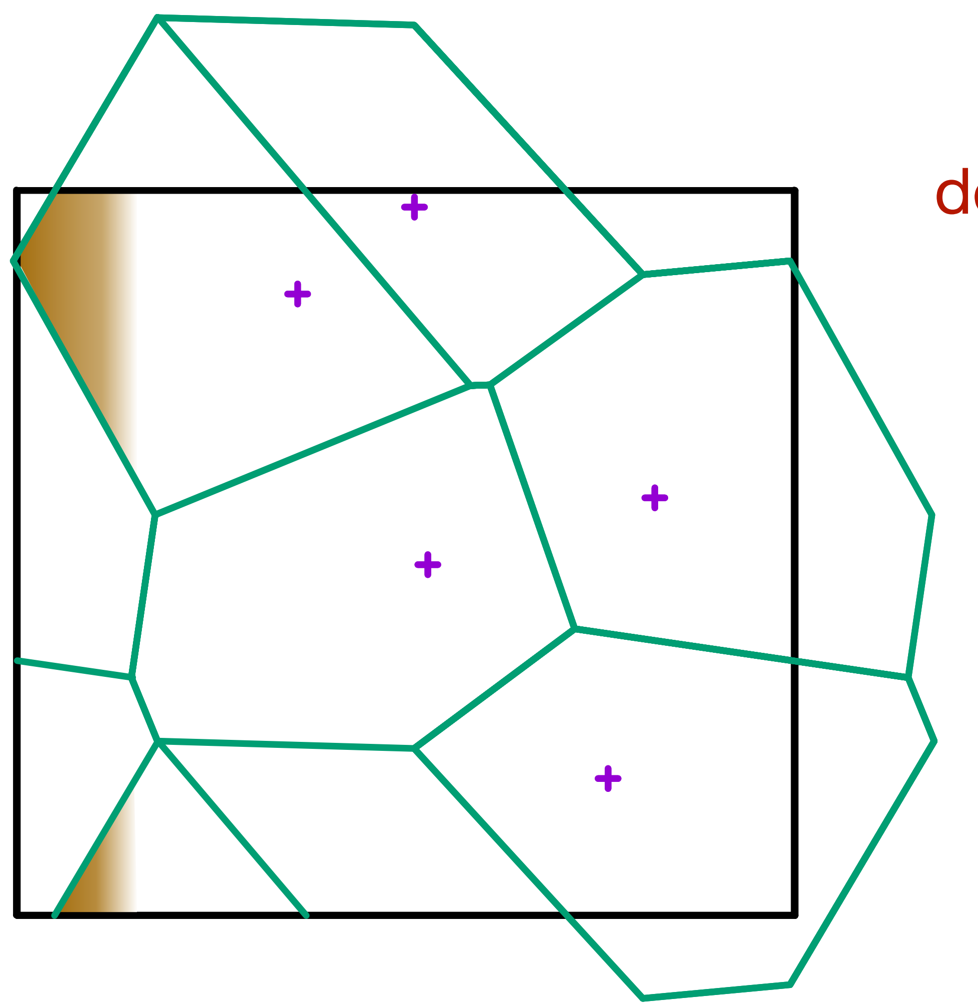
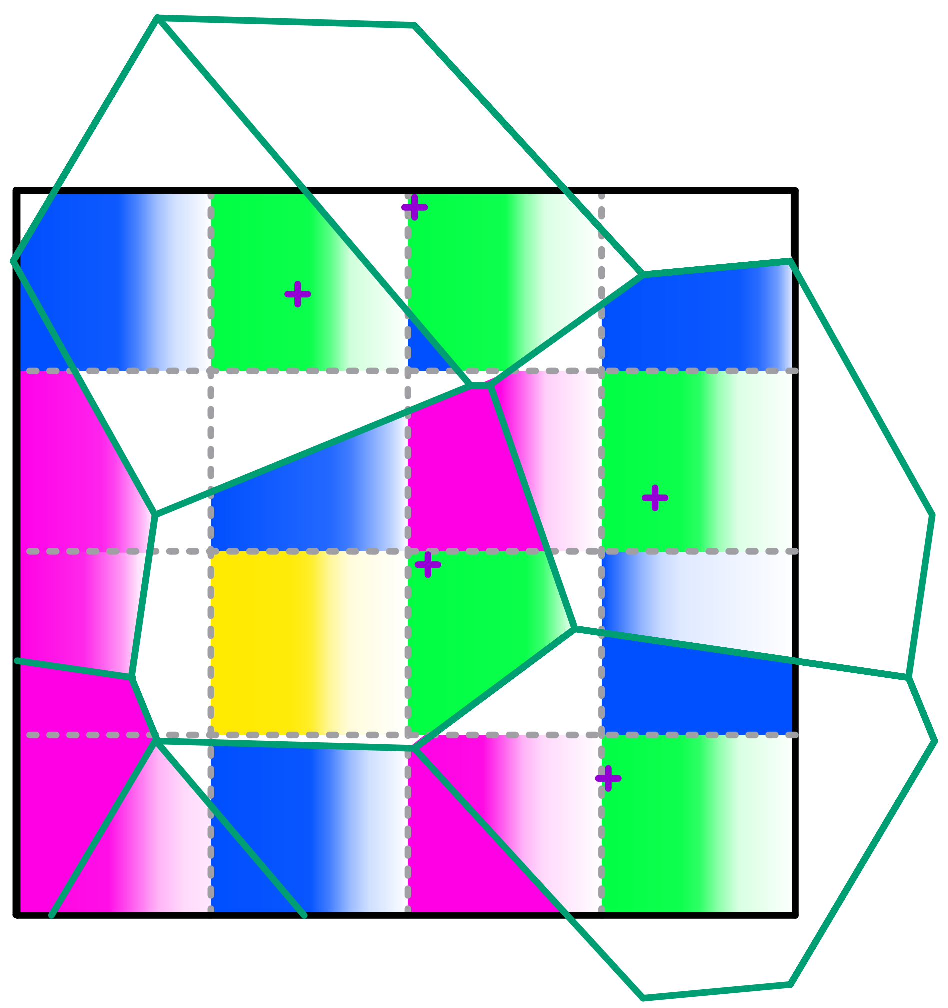
Grain Distribution: Serial



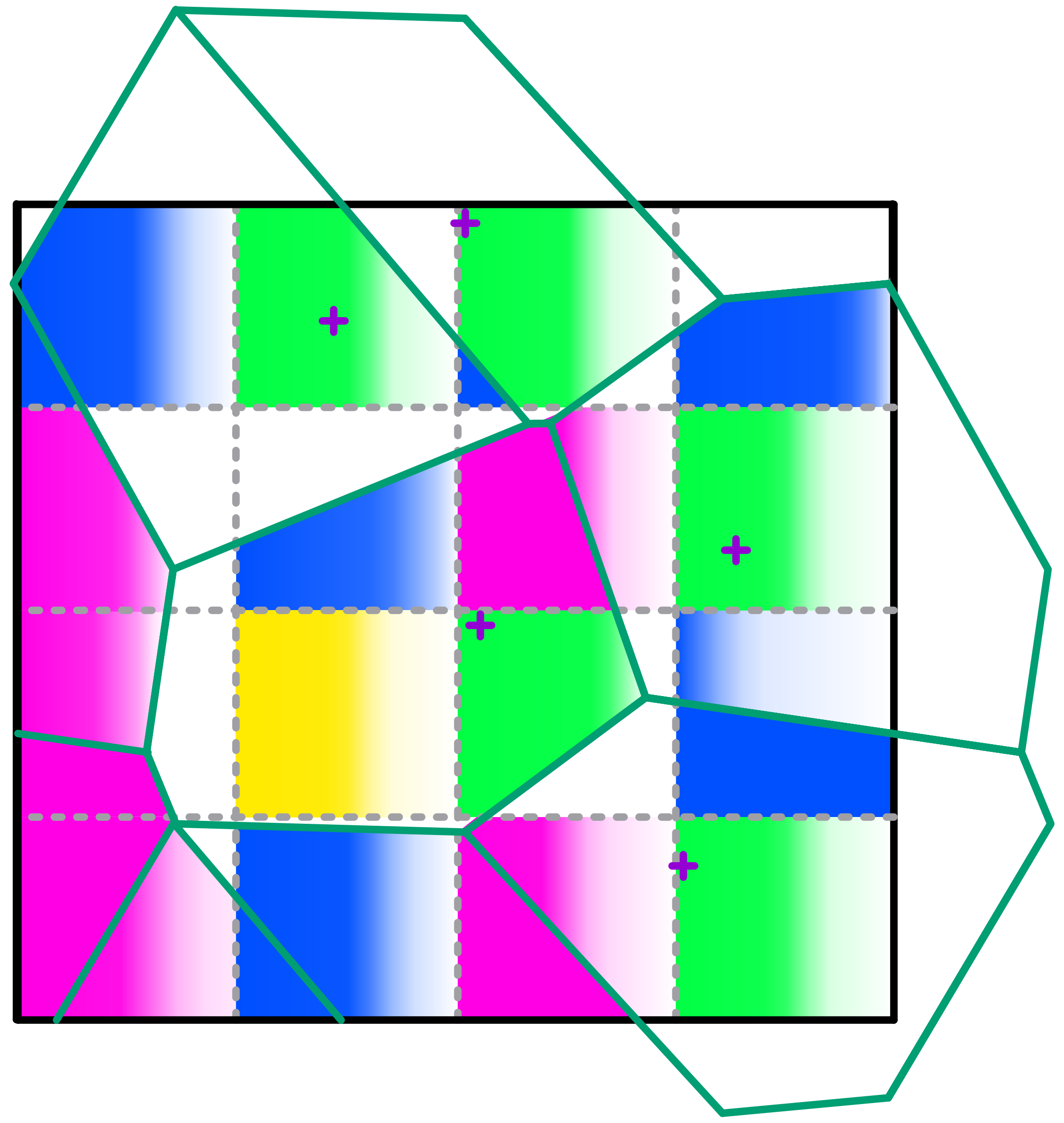
1st grain 2nd grain

One grain at a time: no grain distribution required in serial approach

Grain Distribution: Parallel

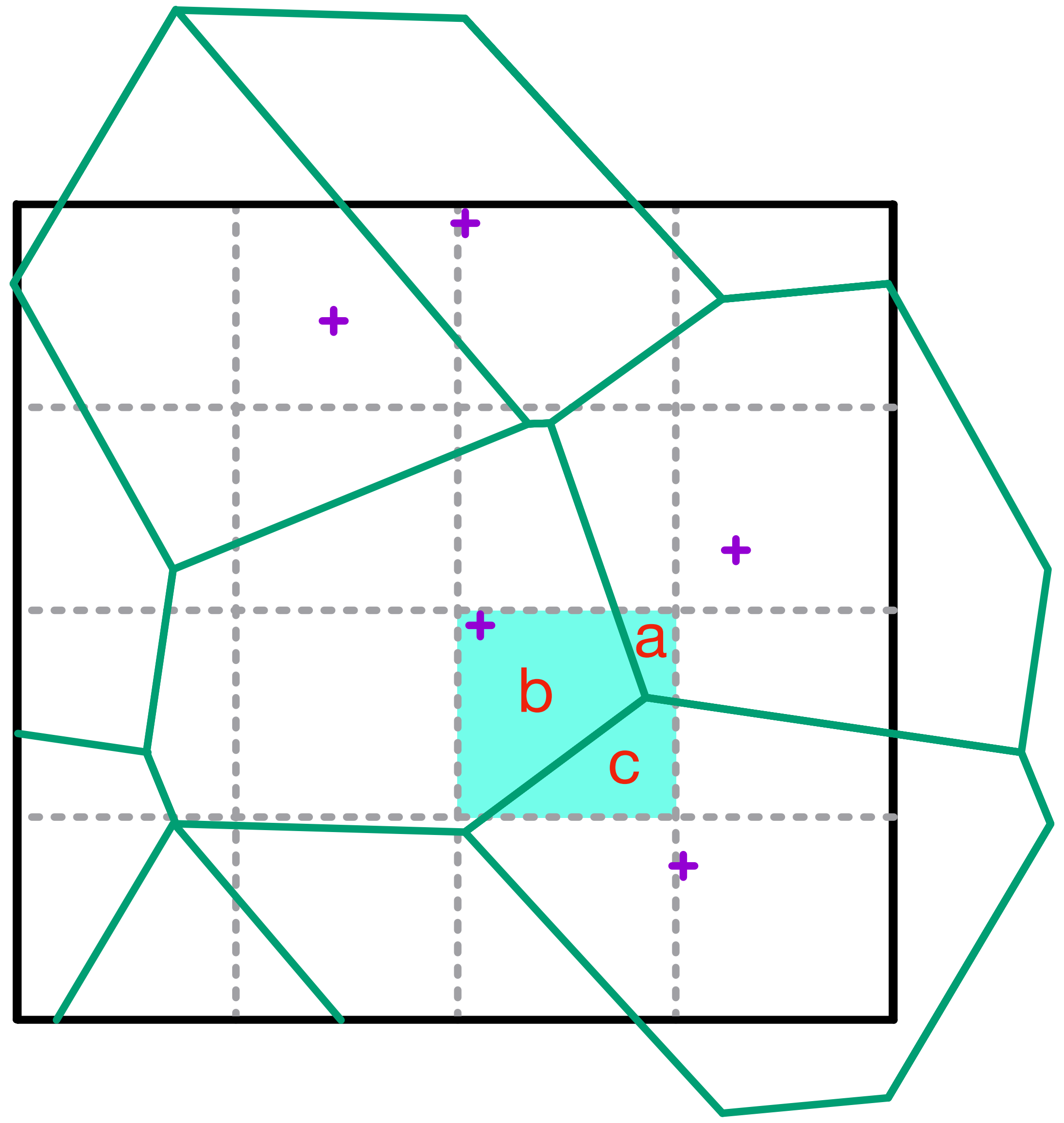
Parallelization strategy	 Each core assigned to a grain	 All detected grains in a subdomain partially filled by corresponding core
# working cores	restricted to number of grains	all cores
Inter-core comm.	required if parallel I/O applied	no
Memory	potential memory lackage issue	all cores require similar amount of memory

Grain Distribution and Atom Generation



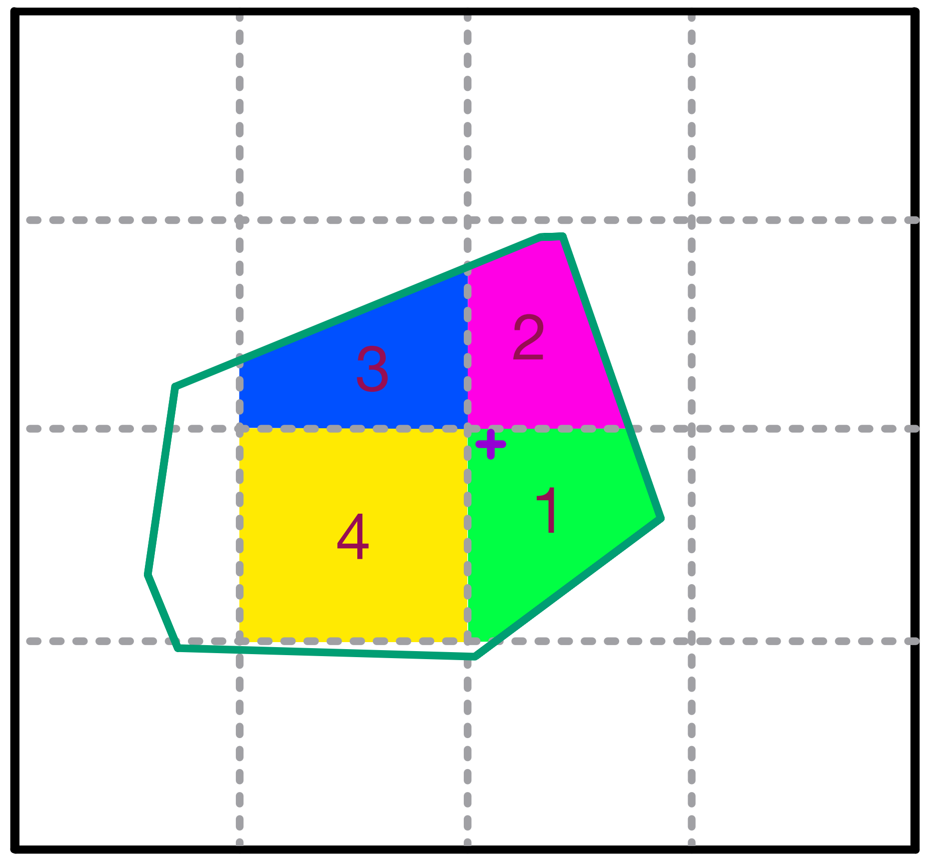
1) Each core finds grains in its subdomain.

2) All cores fill individual subdomains simultaneously. (solving plane equations)



A core can have multiple grains.

A grain can span over multiple subdomains.



(1-green) A seed located inside subdomain

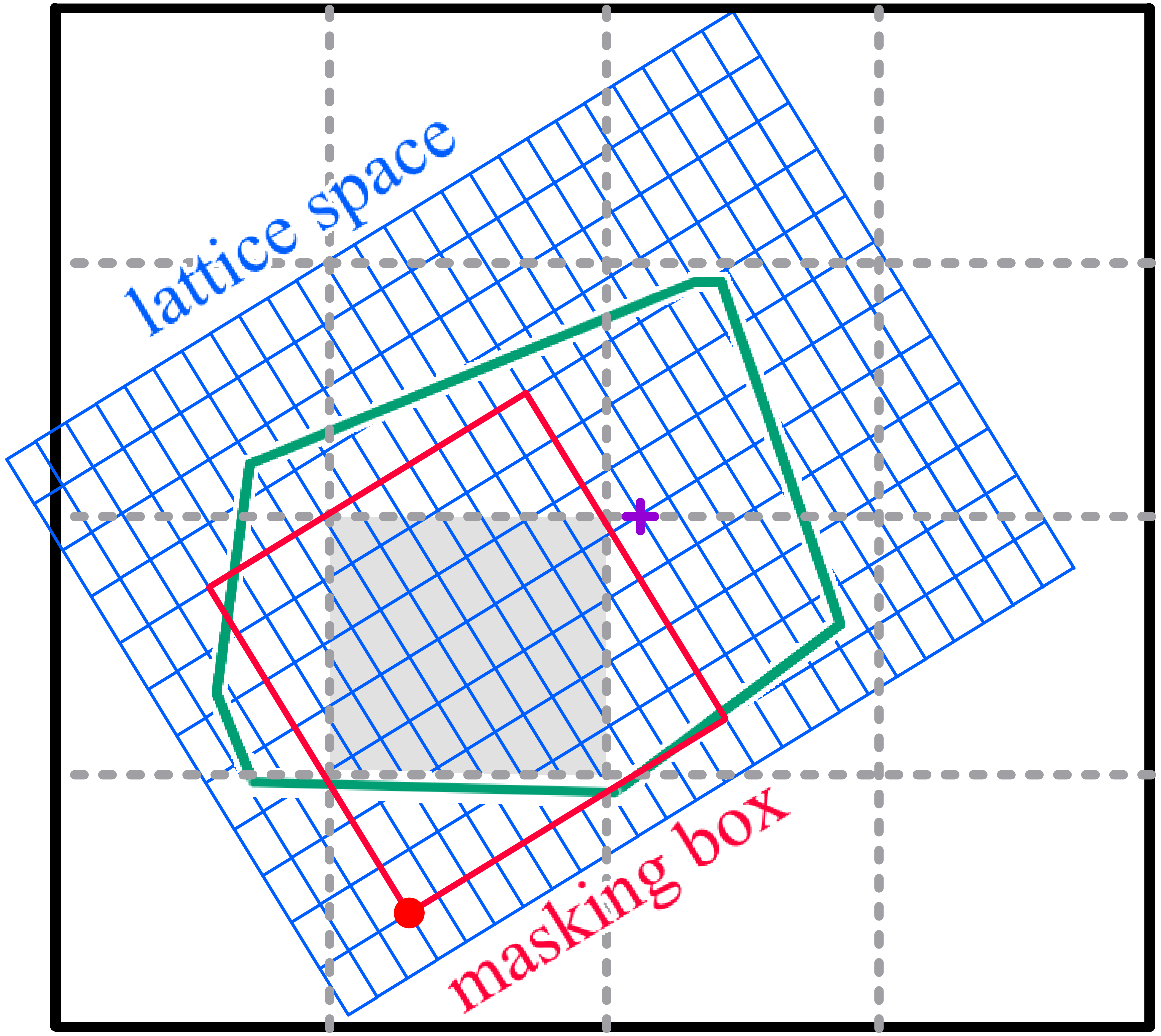
(2-magenta) At least one vertex inside subdomain

(3-blue) At least one edge penetrating across the subdomain

(4-yellow) A grain wrapping the subdomain.

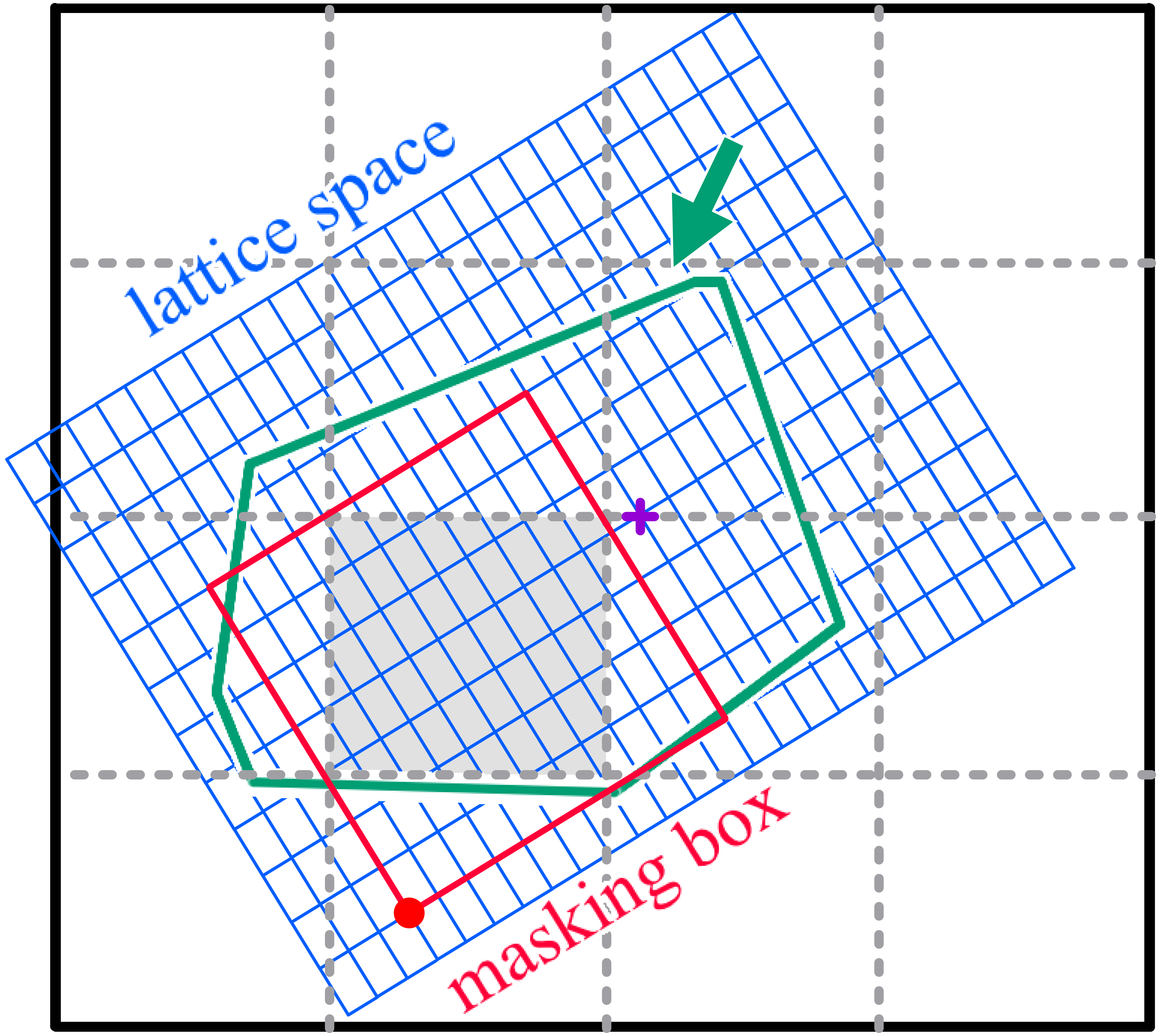
PolyPal

Grain Distribution and Atom Generation



PolyPal

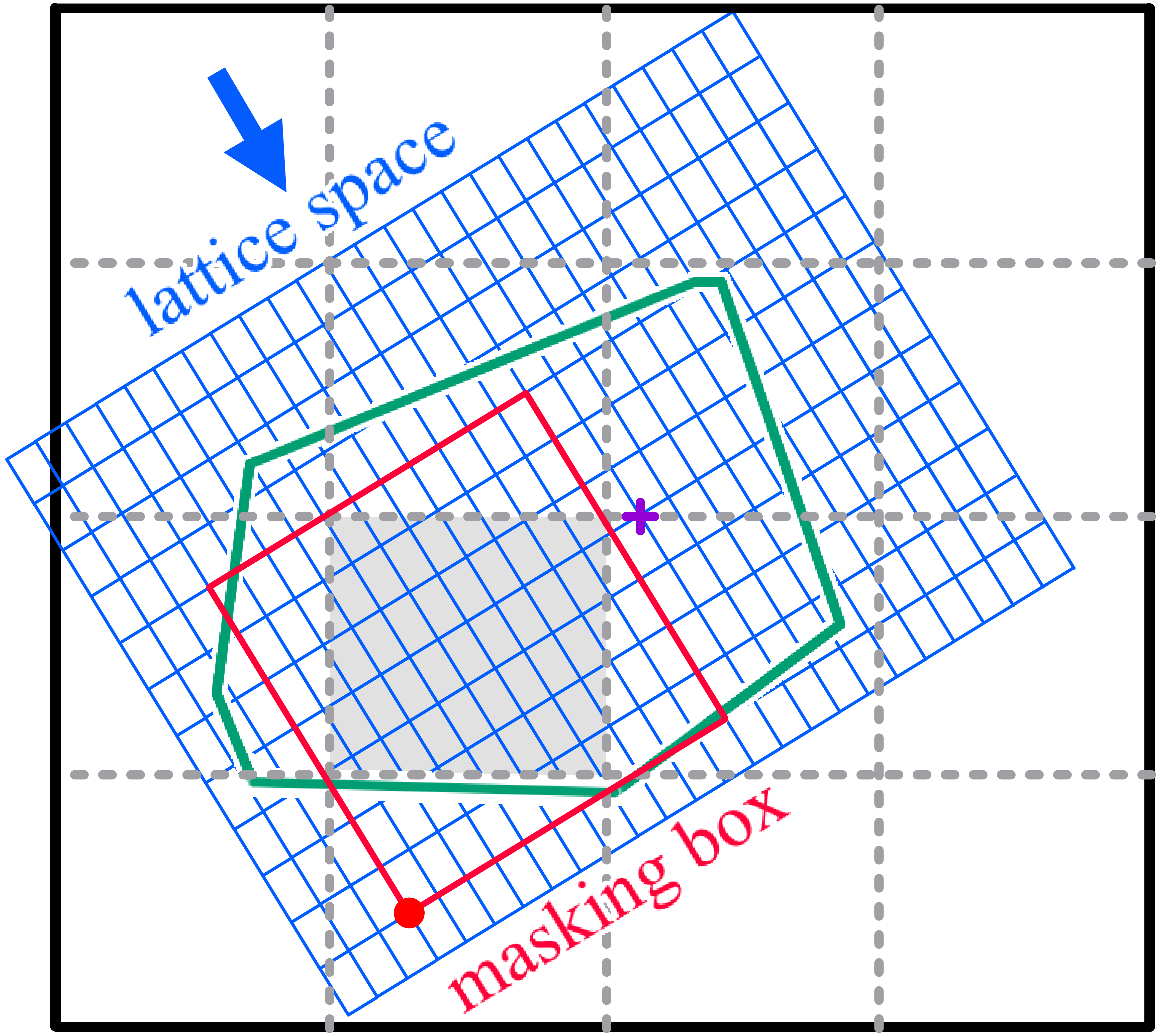
Grain Distribution and Atom Generation



- A grain can span over multiple subdomains.

PolyPal

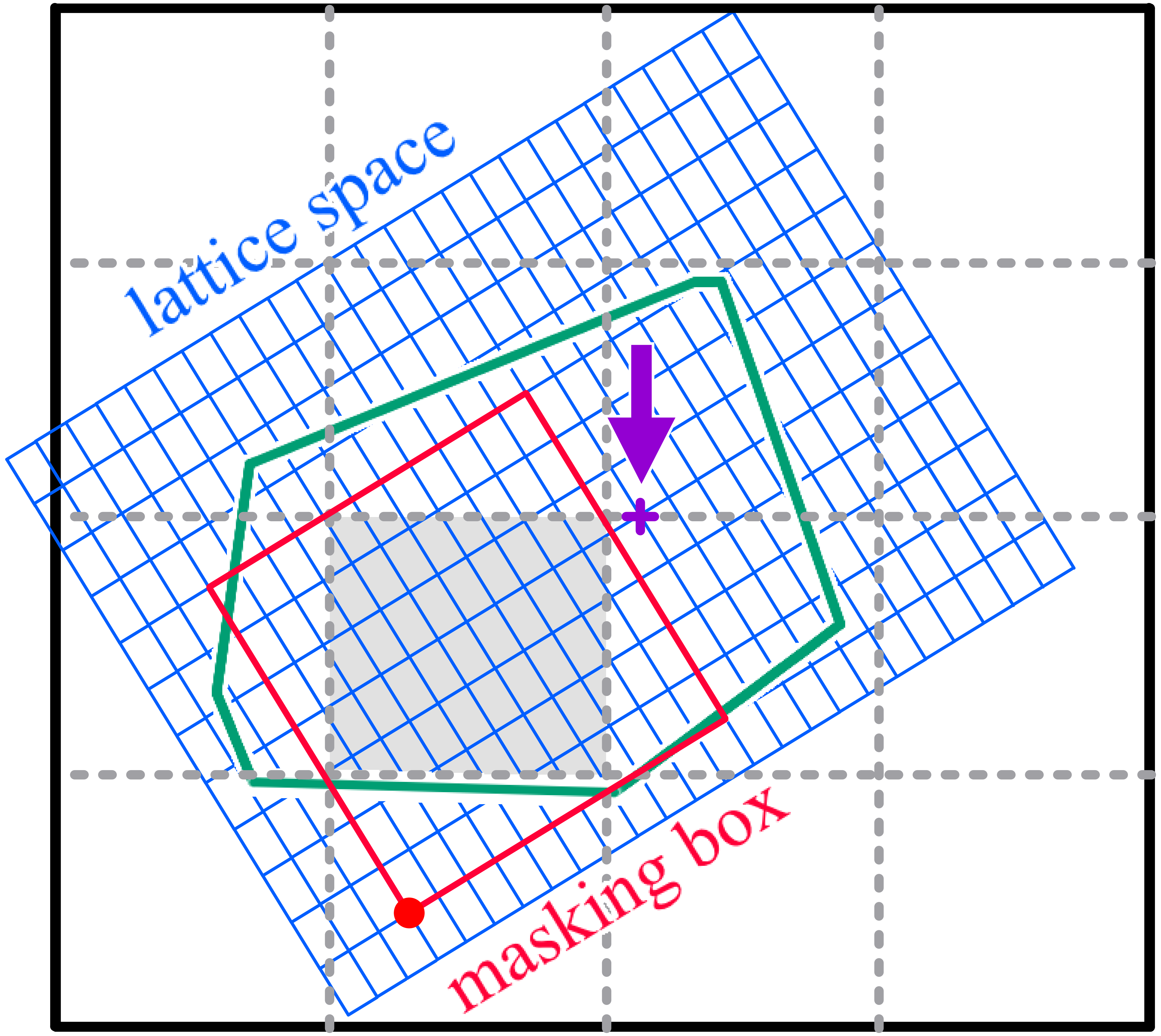
Grain Distribution and Atom Generation



- A grain can span over multiple subdomains.
- To ensure the continuity, cores share the same lattice space.

PolyPal

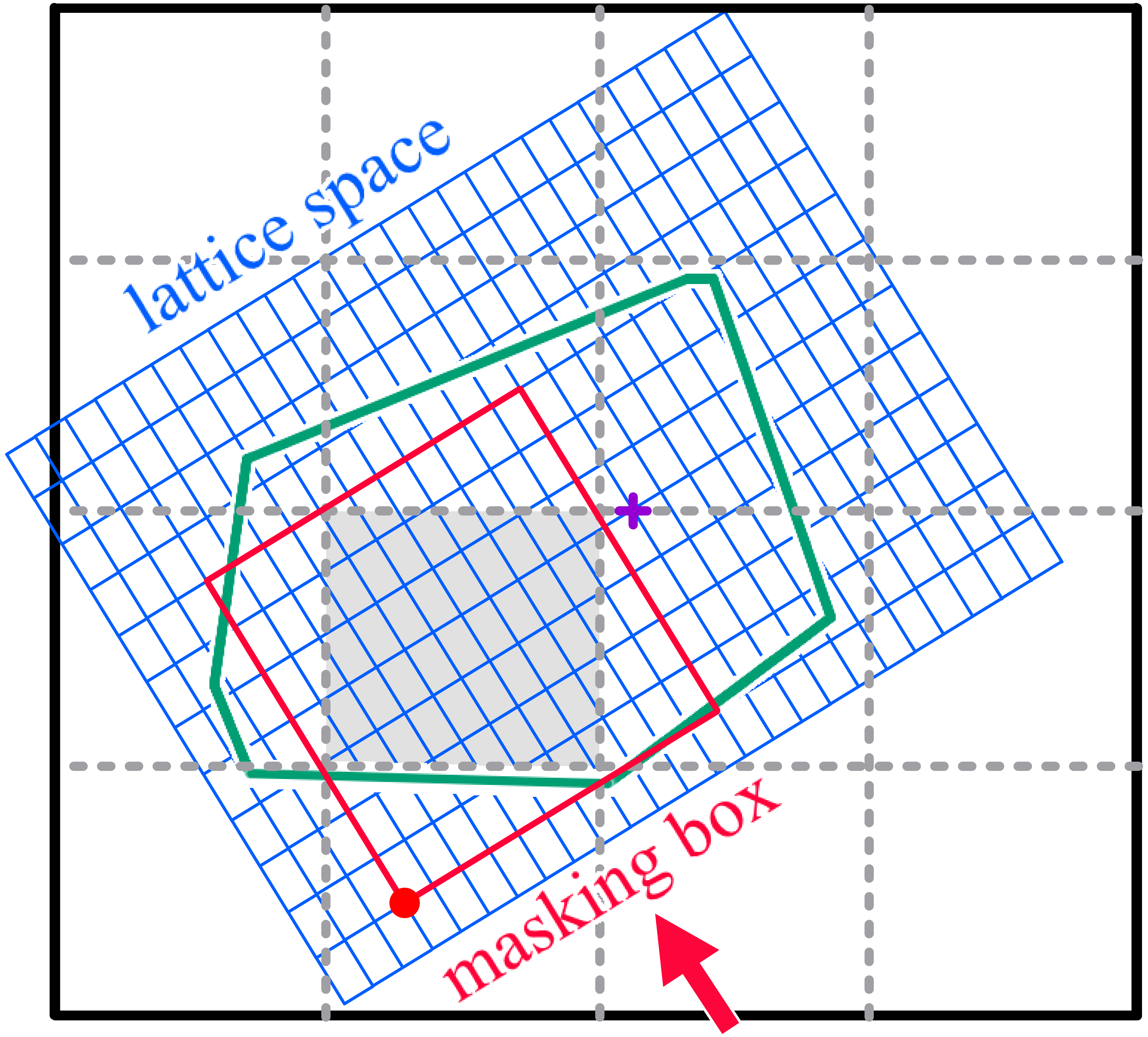
Grain Distribution and Atom Generation



- A grain can span over multiple subdomains.
- To ensure the continuity, cores share the same lattice space.
- The grain seed point (+) is set as the zero point of the lattice space.

PolyPal

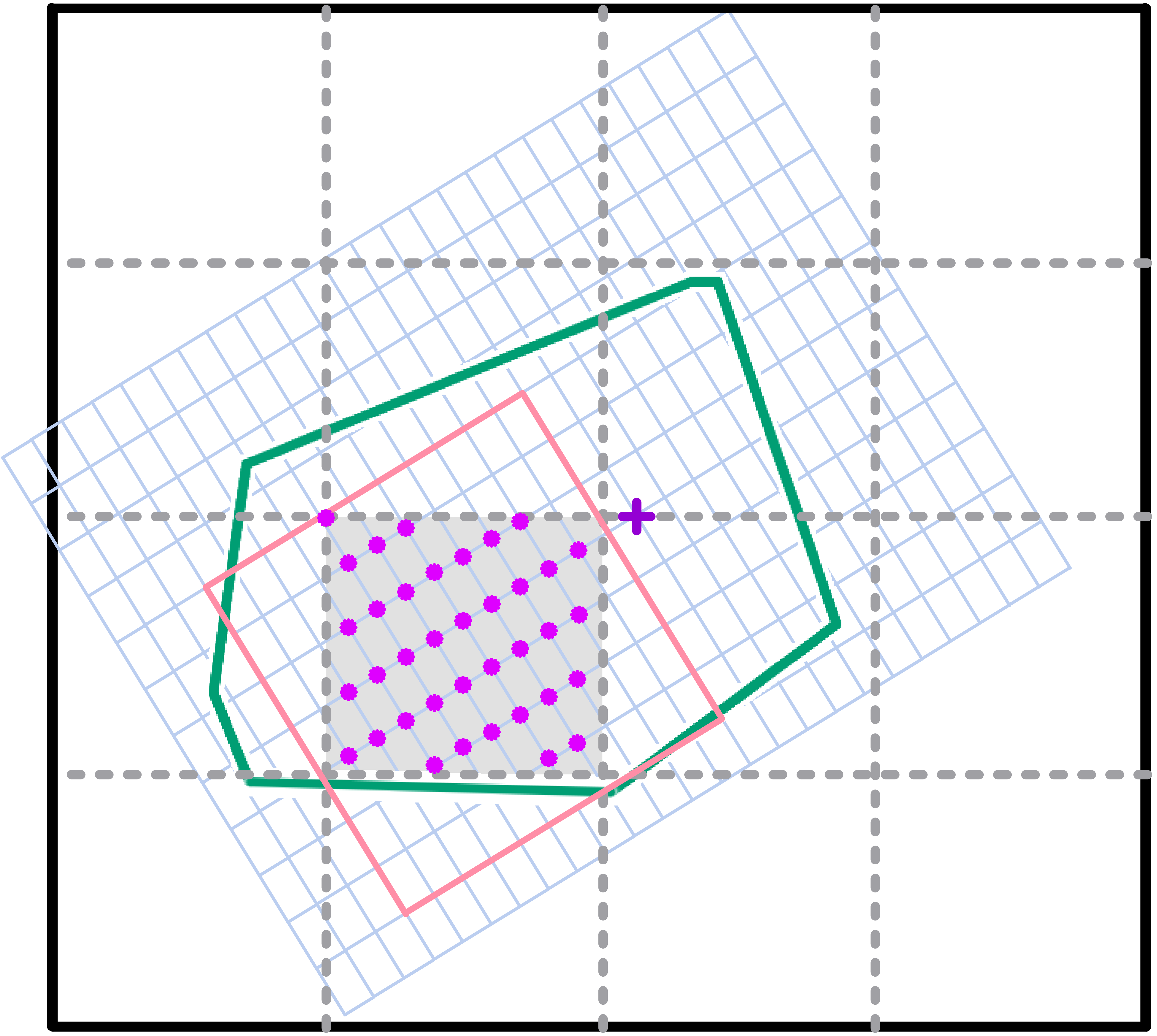
Grain Distribution and Atom Generation



- A grain can span over multiple subdomains.
- To ensure the continuity, cores share the same lattice space.
- The grain seed point (+) is set as the zero point of the lattice space.
- A fictitious masking box, bigger enough to cover the subdomain, is created.

PolyPal

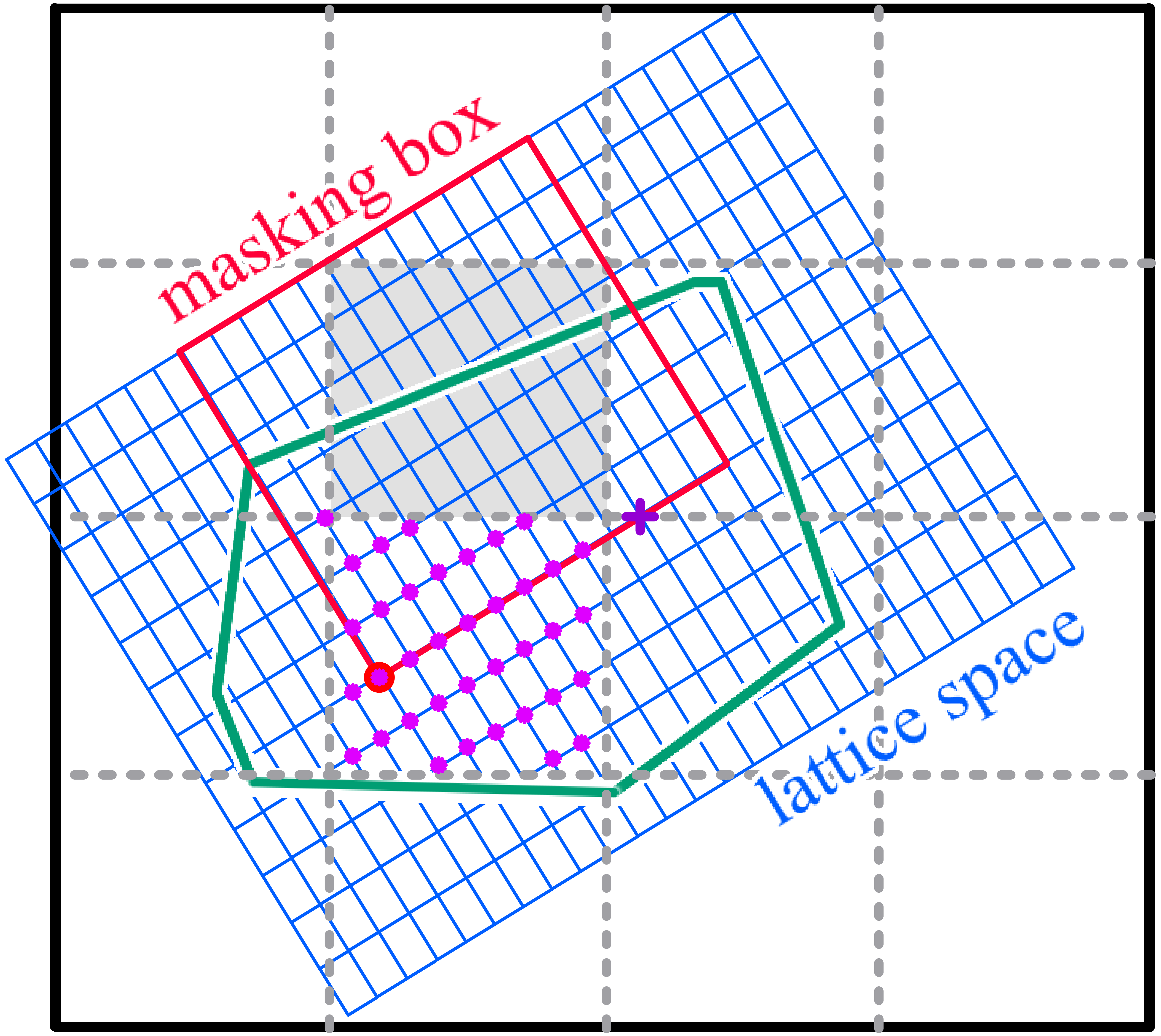
Grain Distribution and Atom Generation



- A grain can span over multiple subdomains.
- To ensure the continuity, cores share the same lattice space.
- The grain seed point (+) is set as the zero point of the lattice space.
- A fictitious masking box, bigger enough to cover the subdomain, is created.
- Sweep lattice points inside the masking box and fill the space with atoms.

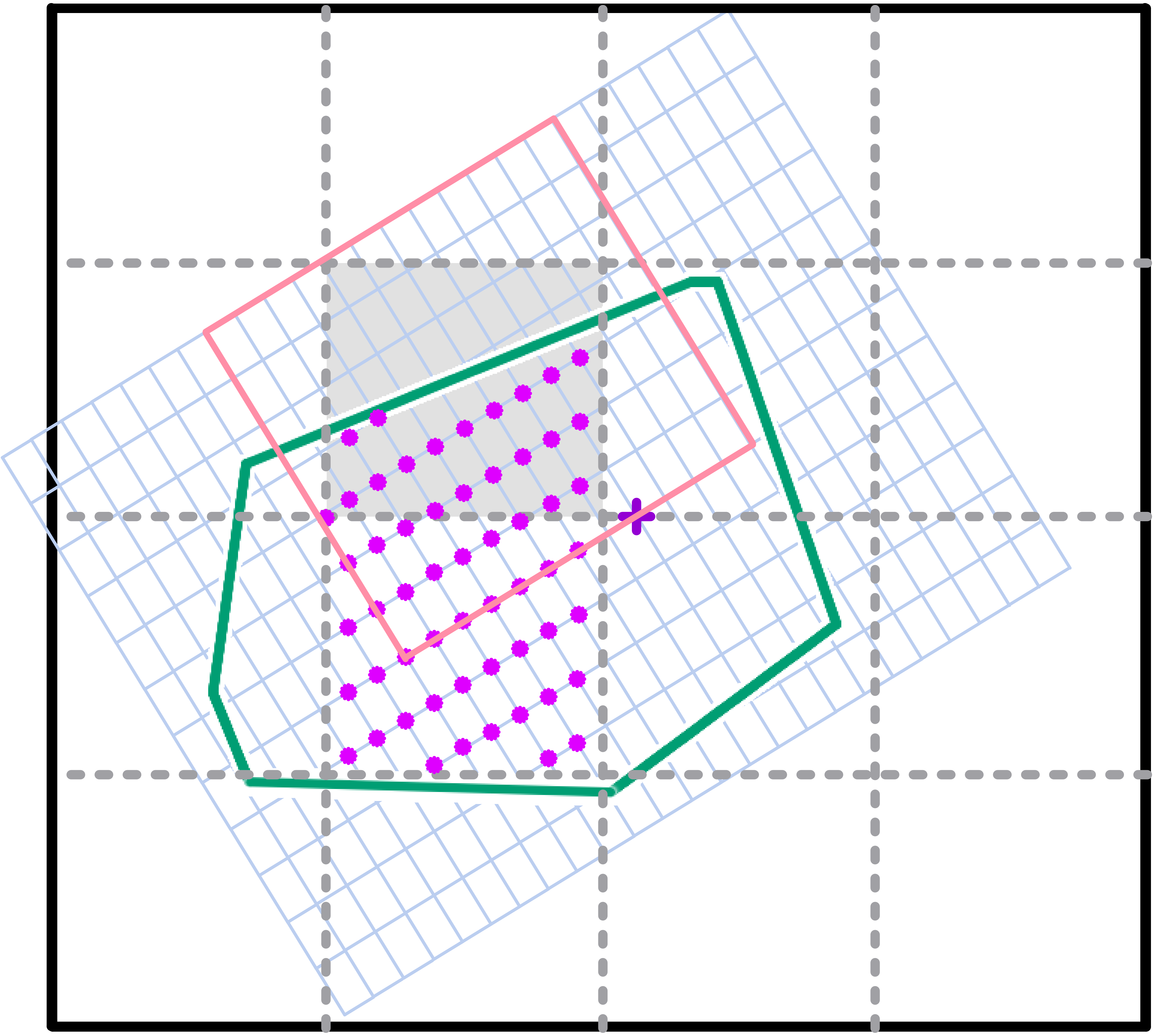
PolyPal

Grain Distribution and Atom Generation



- A grain can span over multiple subdomains.
- To ensure the continuity, cores share the same lattice space.
- The grain seed point (+) is set as the zero point of the lattice space.
- A fictitious masking box, bigger enough to cover the subdomain, is created.
- Sweep lattice points inside the masking box and fill the space with atoms.

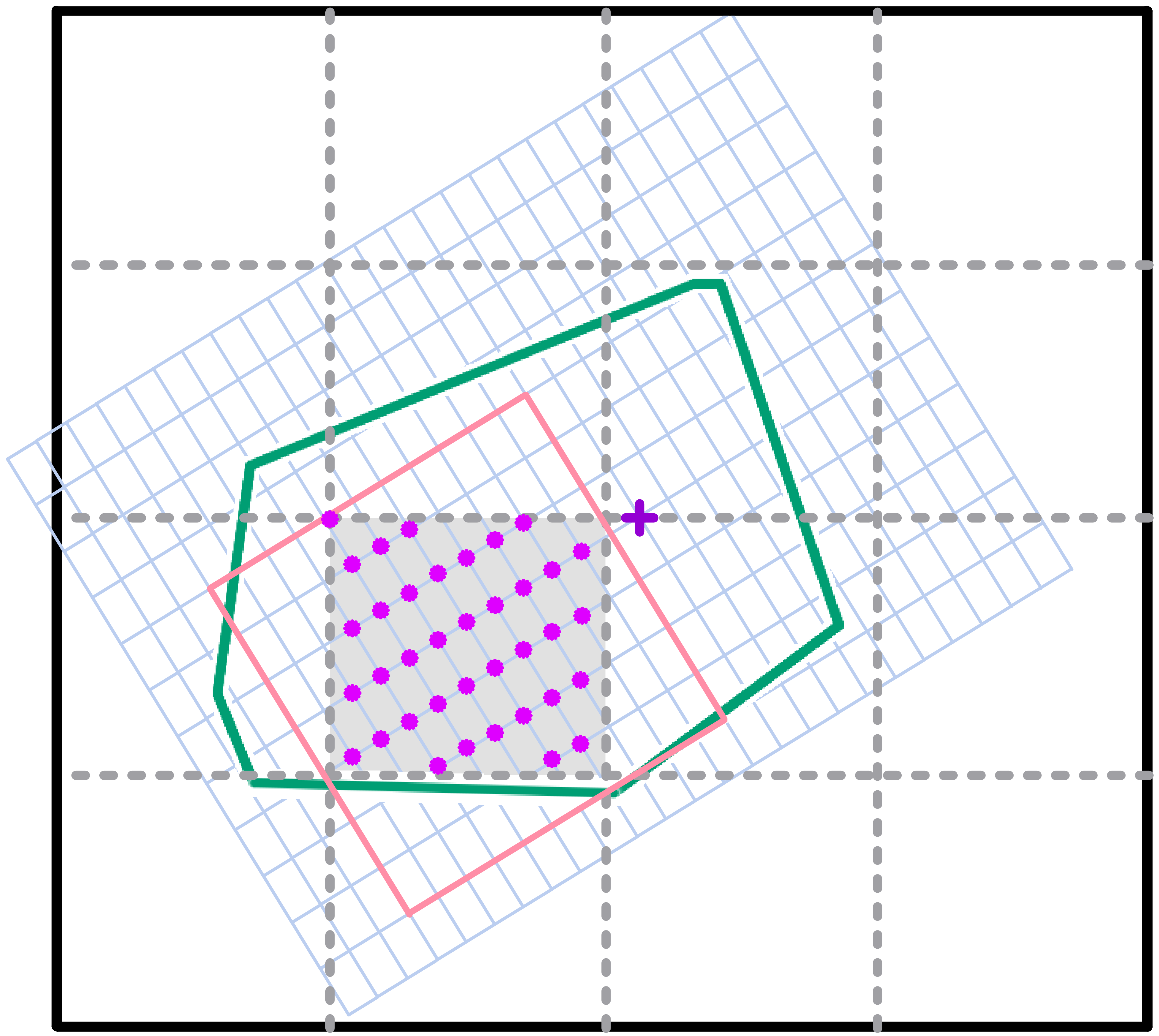
Grain Distribution and Atom Generation



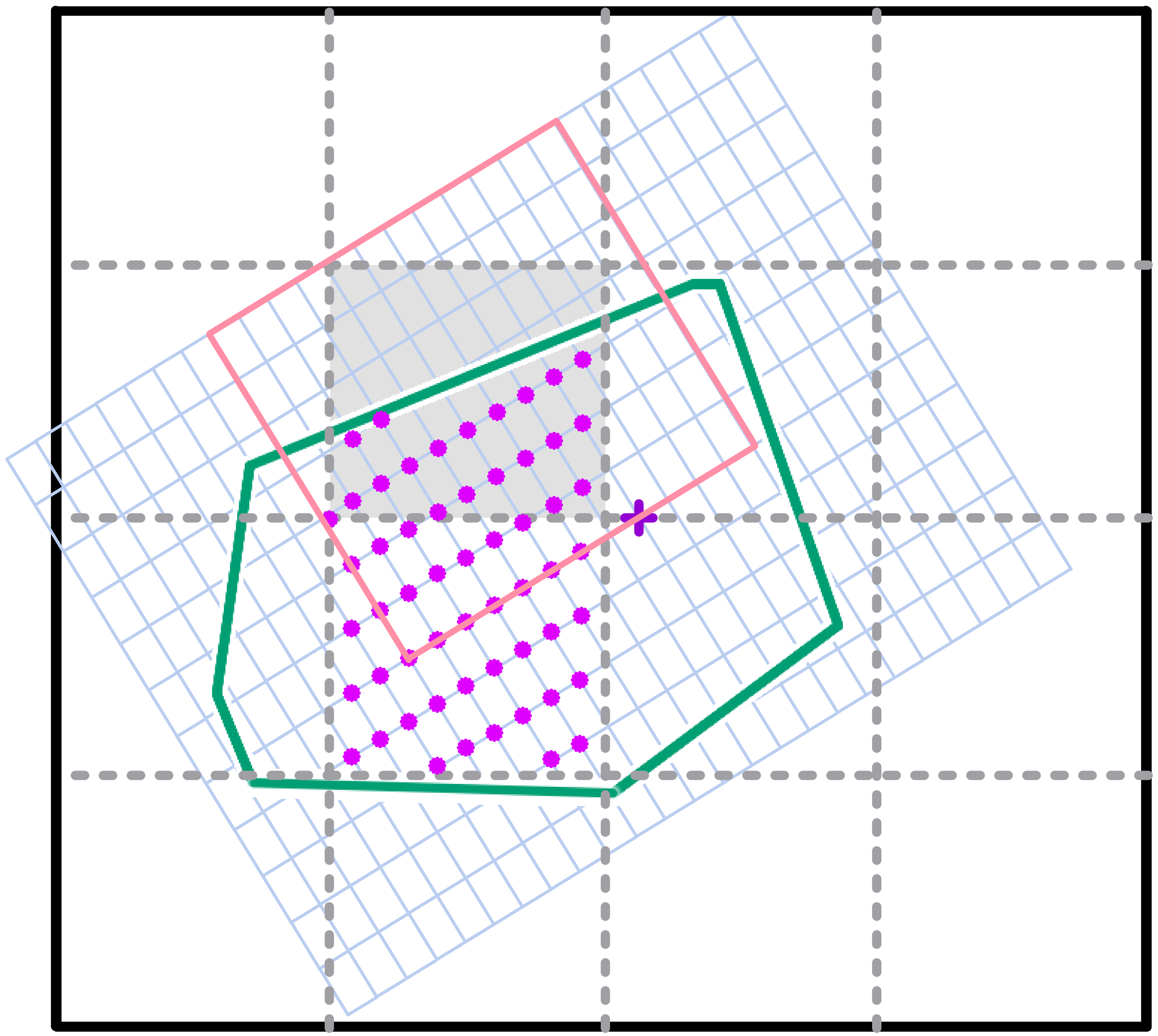
- A grain can span over multiple subdomains.
- To ensure the continuity, cores share the same lattice space.
- The grain seed point (+) is set as the zero point of the lattice space.
- A fictitious masking box, bigger enough to cover the subdomain, is created.
- Sweep lattice points inside the masking box and fill the space with atoms.

PolyPal

Grain Distribution and Atom Generation



1st subdomain filled



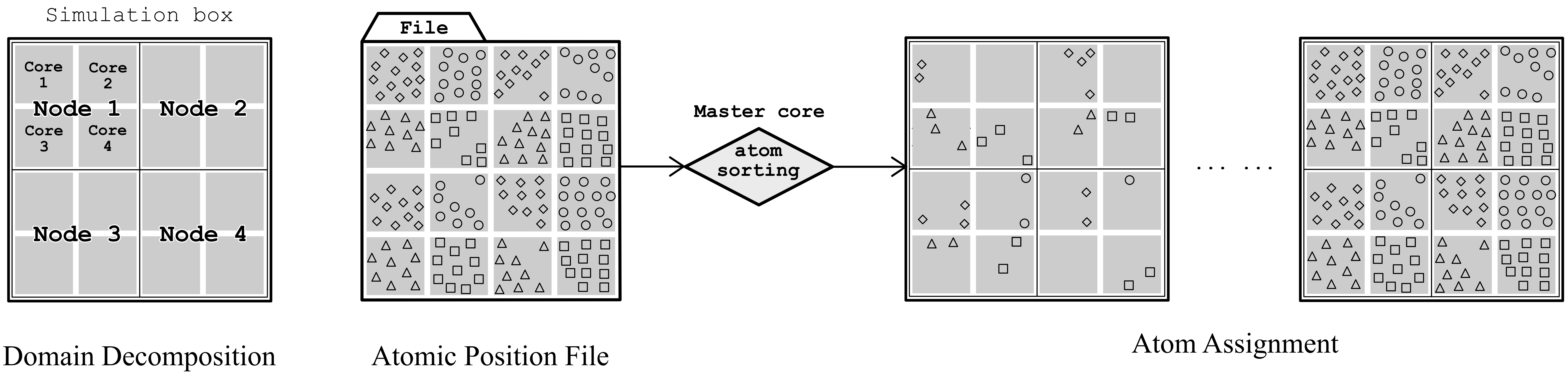
1st & 2nd subdomains filled

Parallel I/O

Parallel I/O

PolyPal

Position File: Conventional MD Code



Node: 2x2 = 4
Core per node: 2x2 = 4

File reading and atom distribution by a master core

Single-core processing & extensive core-to-core communications

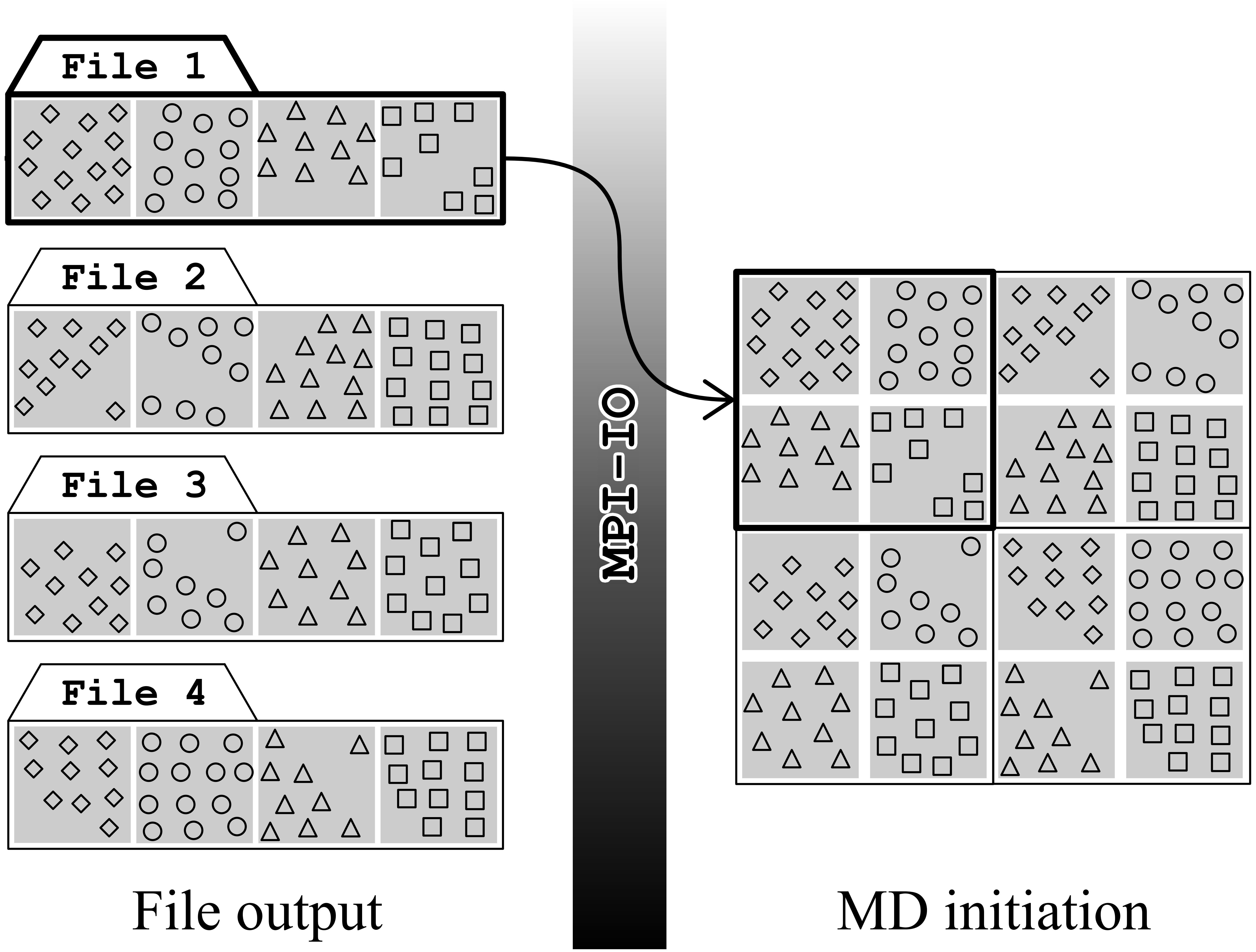
Position File: Domain-Preserving Parallel I/O

Shin et al., Comput. Phys. Commun. (2025)

* File-per-node
* Data chunk organised by cores

PolyPal

Position File: Domain-Preserving Parallel I/O



MD init. test

1 billion atoms

100 x 64 cores

PolyPal + VITMAS

- File per node
- MPI-IO

1m 30s

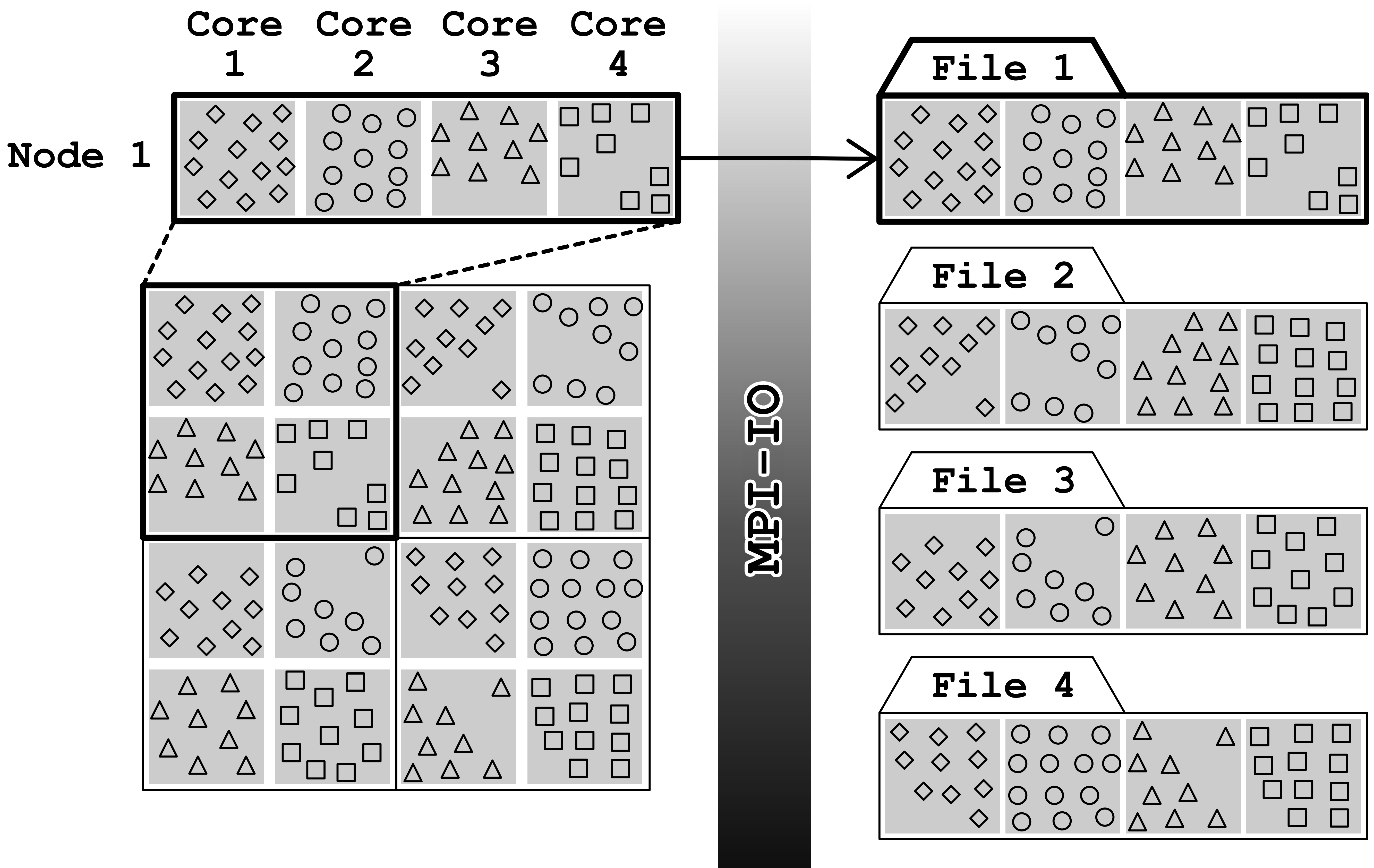
LAMMPS

- Single file
- no MPI-IO

7 hrs (x264)

PolyPal

Position File: Domain-Preserving Parallel I/O



Atom population

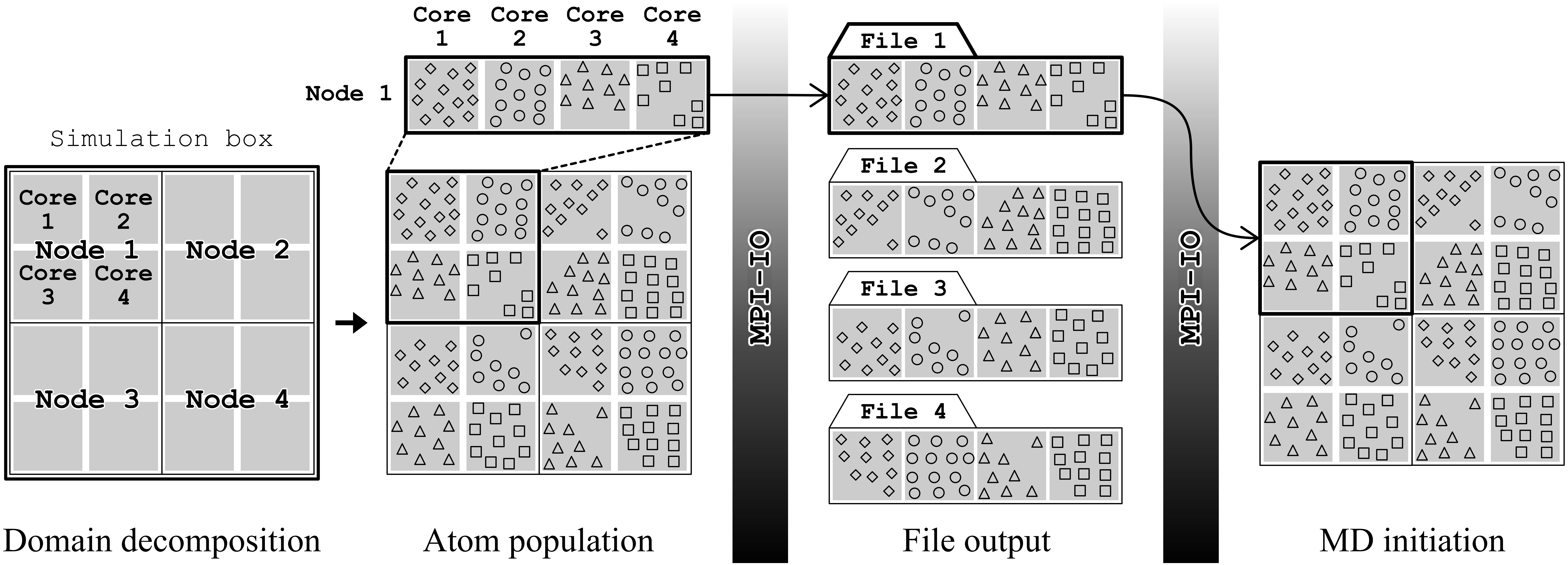
File output

header = map

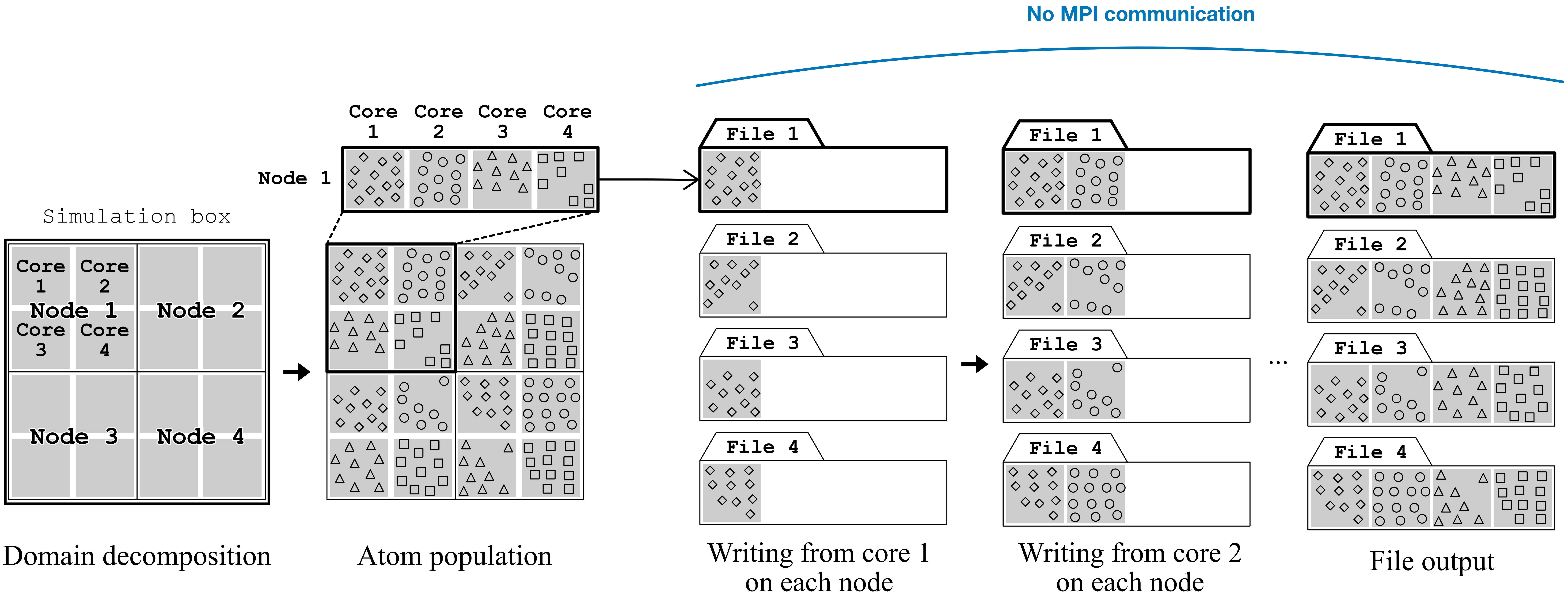
```
2                  Number of species
74  75             Atomic numbers (W, Re)
15558  304         Numbers of  W/Re atoms  in core 1
15479  254         Numbers of  W/Re atoms  in core 2
15367  270         Numbers of  W/Re atoms  in core 3
15601  311         Numbers of  W/Re atoms  in core 4
316.51900  0.00028  0.11128
316.51900  0.00028  3.27648
…  …
```

PolyPal

Position File: File-per-Node MPI-IO



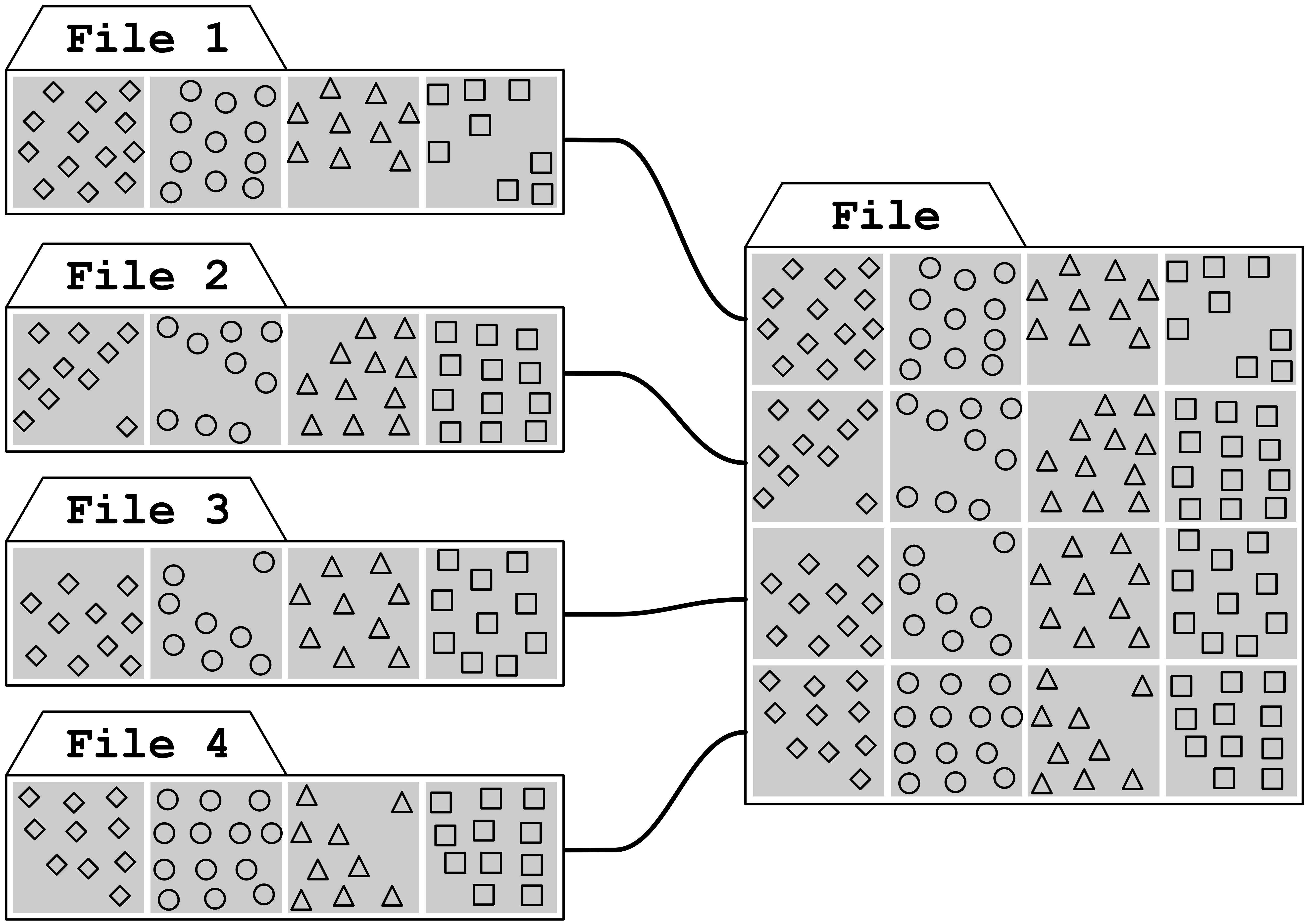
Position File: File-per-Node Non-MPI-IO



Asynchronous data dump from all cores.

PolyPal

Position File: Single-File Non-MPI-IO



Concatenation

No additional memory required

No additional processing required

PolyPal

Performance Metrics

Output				Wall time (s)	
Format	File policy	MPI-IO	Writer	File output	Others
Binary	Single	×	Global master	1099	151
Binary	Single	×	All cores*	10	151
Binary	File/node	×	All cores*	8	151
Binary	File/node	∘	All cores	3	151
ASCII text	Single	×	Global master	4423	151
ASCII text	Single	×	All cores*	100	150
ASCII text	File/node	×	All cores*	54	151
ASCII text	File/node	∘	All cores	45	151

HPC system: 6400 cores (64 cores/node) from Nurion-5 at KISTI
- Intel Xeon Phi 7250 1.4 GHz CPU
- 96 GB/node memory with Lustre file system

Atomic structure: total 10,030,152,440 atoms in 300 grains

Shin et al., Comput. Phys. Commun. (2025)

PolyPal

Performance Metrics

Output				Wall time (s)	
Format	File policy	MPI-IO	Writer	File output	Others
Binary	Single	×	Global master	1099	151
Binary	Single	×	All cores*	10	151
Binary	File/node	×	All cores*	8	151
Binary	File/node	∘	All cores	3	151
ASCII text	Single	×	Global master	4423	151
ASCII text	Single	×	All cores*	100	150
ASCII text	File/node	×	All cores*	54	151
ASCII text	File/node	∘	All cores	45	151

HPC system: 6400 cores (64 cores/node) from Nurion-5 at KISTI
- Intel Xeon Phi 7250 1.4 GHz CPU
- 96 GB/node memory with Lustre file system

Atomic structure: total 10,030,152,440 atoms in 300 grains

Shin et al., Comput. Phys. Commun. (2025)

file output: 2%

others: 98%

Only takes **2m 30s** to generate a **10-billion**-atom system.

PolyPal

Performance Metrics

Output				Wall time (s)	
Format	File policy	MPI-IO	Writer	File output	Others
Binary	Single	×	Global master	1099	151
Binary	Single	×	All cores*	10	151
Binary	File/node	×	All cores*	8	151
Binary	File/node	∘	All cores	3	151
ASCII text	Single	×	Global master	4423	151
ASCII text	Single	×	All cores*	100	150
ASCII text	File/node	×	All cores*	54	151
ASCII text	File/node	∘	All cores	45	151

HPC system: 6400 cores (64 cores/node) from Nurion-5 at KISTI
- Intel Xeon Phi 7250 1.4 GHz CPU
- 96 GB/node memory with Lustre file system

Atomic structure: total 10,030,152,440 atoms in 300 grains

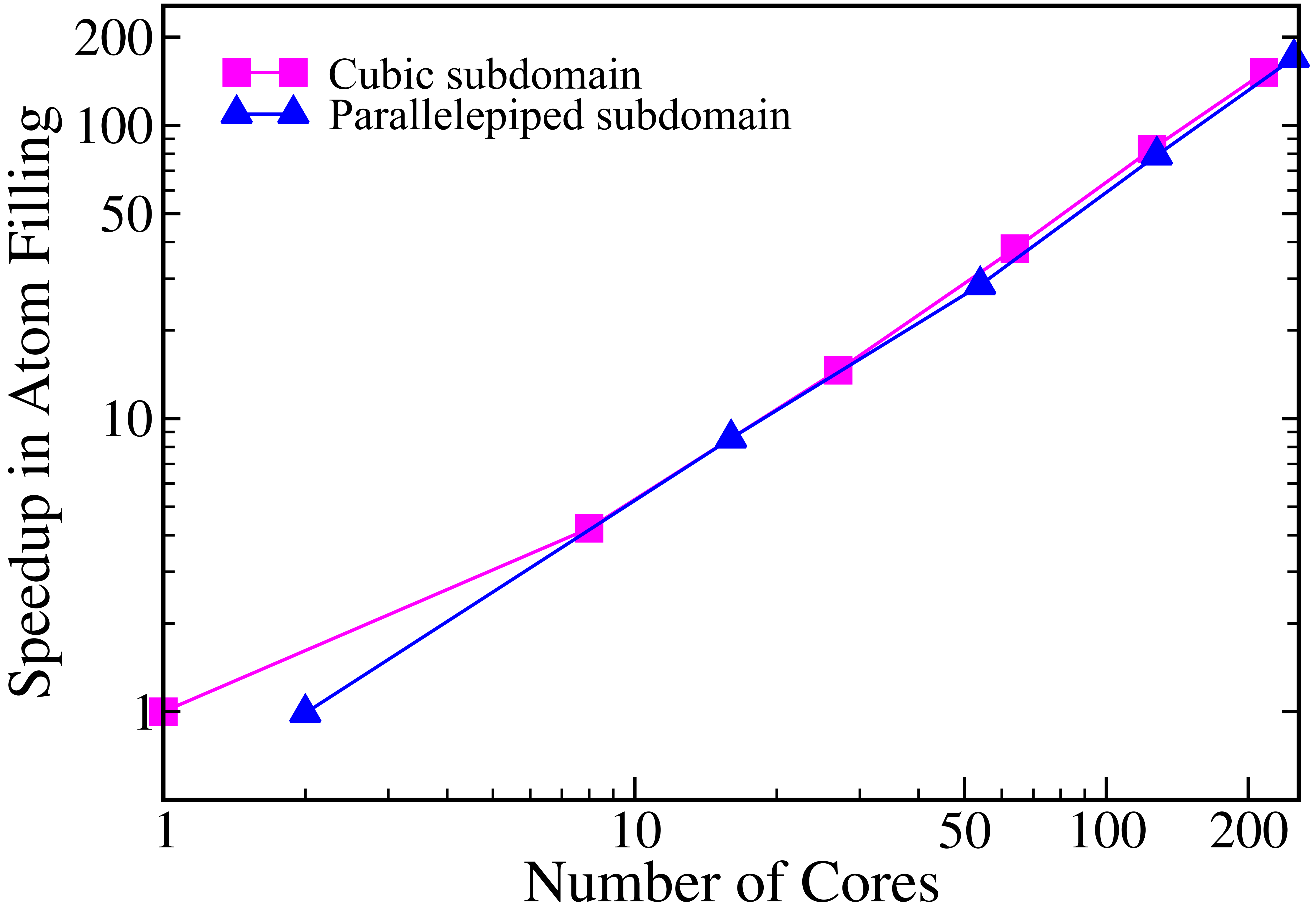
Shin et al., Comput. Phys. Commun. (2025)

*binary ~225gb / ASCII ~534gb

Binary format output: faster and light-weighted*

PolyPal

Performance Metrics: Strong Scaling Test



Shin et al., Comput. Phys. Commun. (2025)

Near-perfect load balancing + no inter-core comm. = **excellent scalability**

PolyPal

Takeaway

- Massive-scale polycrystalline structures → realistic simulations
- File I/O: a huge bottleneck for massive-scale simulation
- PolyPal & VITMAS domain-preserving file I/O strategy
- Excellent scalability due to near-perfect load balancing and no inter-core communication
- Versatile options for tailoring desired polycrystalline structure

“We need appropriate tools for massive-scale
MD simulations.”

PolyPal

Publication

Computer Physics Communications 308 (2025) 109458

Computer Physics Communications 308 (2025) 109458

ELSEVIER

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computer Programs in Physics

Check for updates

PolyPal: A parallel microscale virtual specimen generator ☆

Younggak Shin [a,b], Vichhika Moul [a], Keonwook Kang [b], Byeongchan Lee [a,*]

[a] *Kyung Hee University,1732 Deogyeong-daero, Yongin, Gyeonggi 17104, Republic of Korea*
[b] *Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 03722, Republic of Korea*

A R T I C L E   I N F O

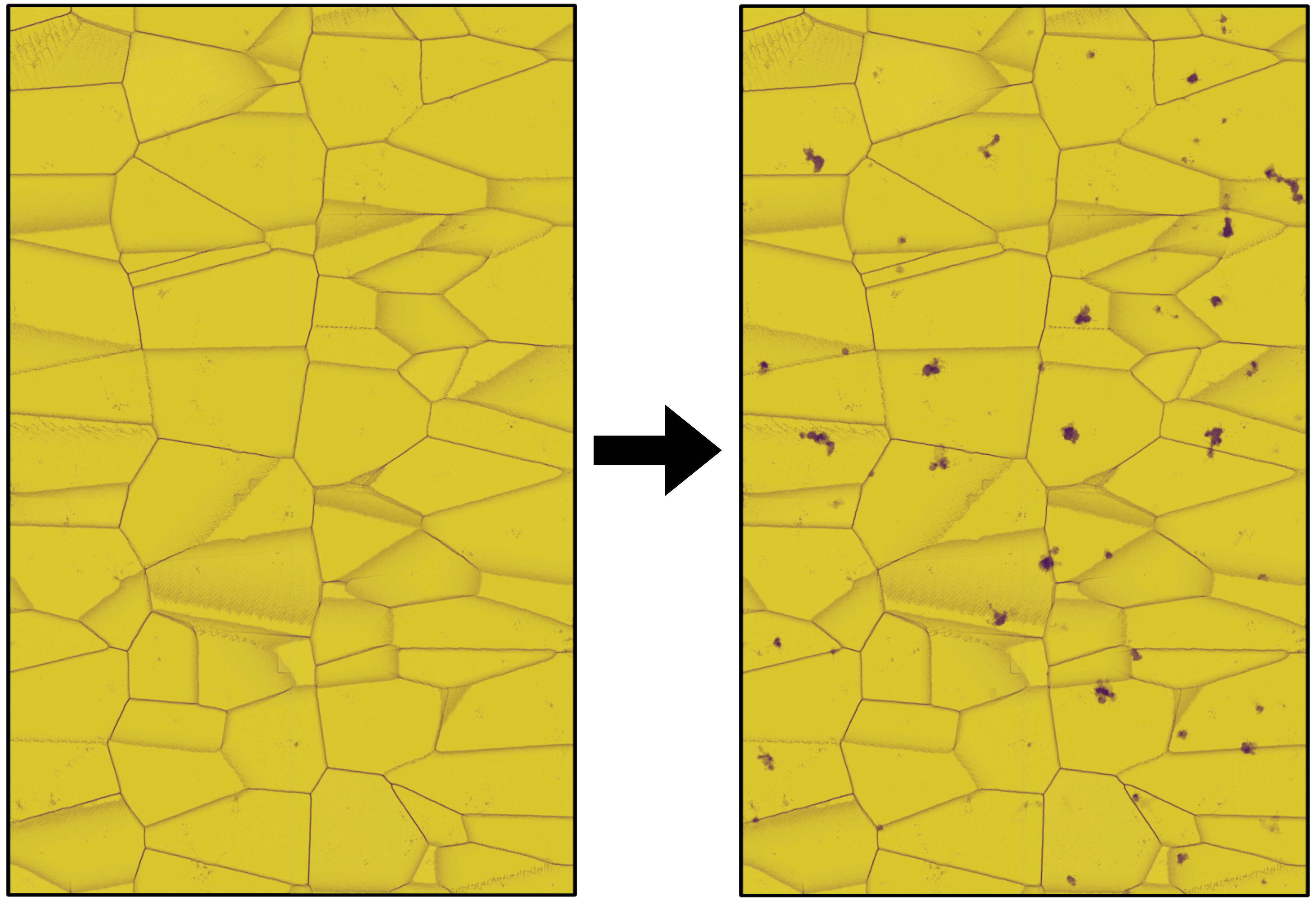
Keywords:
Atomistic simulations
Polycrystal
Microscale
Parallel computing

A B S T R A C T

We present an open source program, PolyPal, that can generate a polycrystalline virtual specimen in the micrometer scale for atomistic calculations and visualization. Unlike regular meshes or perfect lattices, atomic positions in polycrystalline materials need to be defined before calculations, and the capability of an atom-generation code is evaluated by the maximum size of the virtual specimen it can generate as well as by the efficiency of the necessary input-output process. Present atom-generation codes are implemented in a serial fashion, and the maximum size of the virtual specimen is limited by the on-board memory. Furthermore, it is difficult to handle a single position file with billions of atoms not only because it takes a long time to read in a row but also full domain decomposition takes hours. PolyPal addresses these challenges with a fully parallelized MPI input-output scheme that supports multiple export options on a Linux cluster. It has no limit in the system size with virtually perfect scalability. Additionally by controlling the size distribution and homogeneity of grains, the program can simulate different microstructures, as typically found in the bulk system or in thin-film samples, prepared with different fabrication processes. PolyPal will harness molecular dynamics codes in the coming age of the exascale computing.

Future Plans

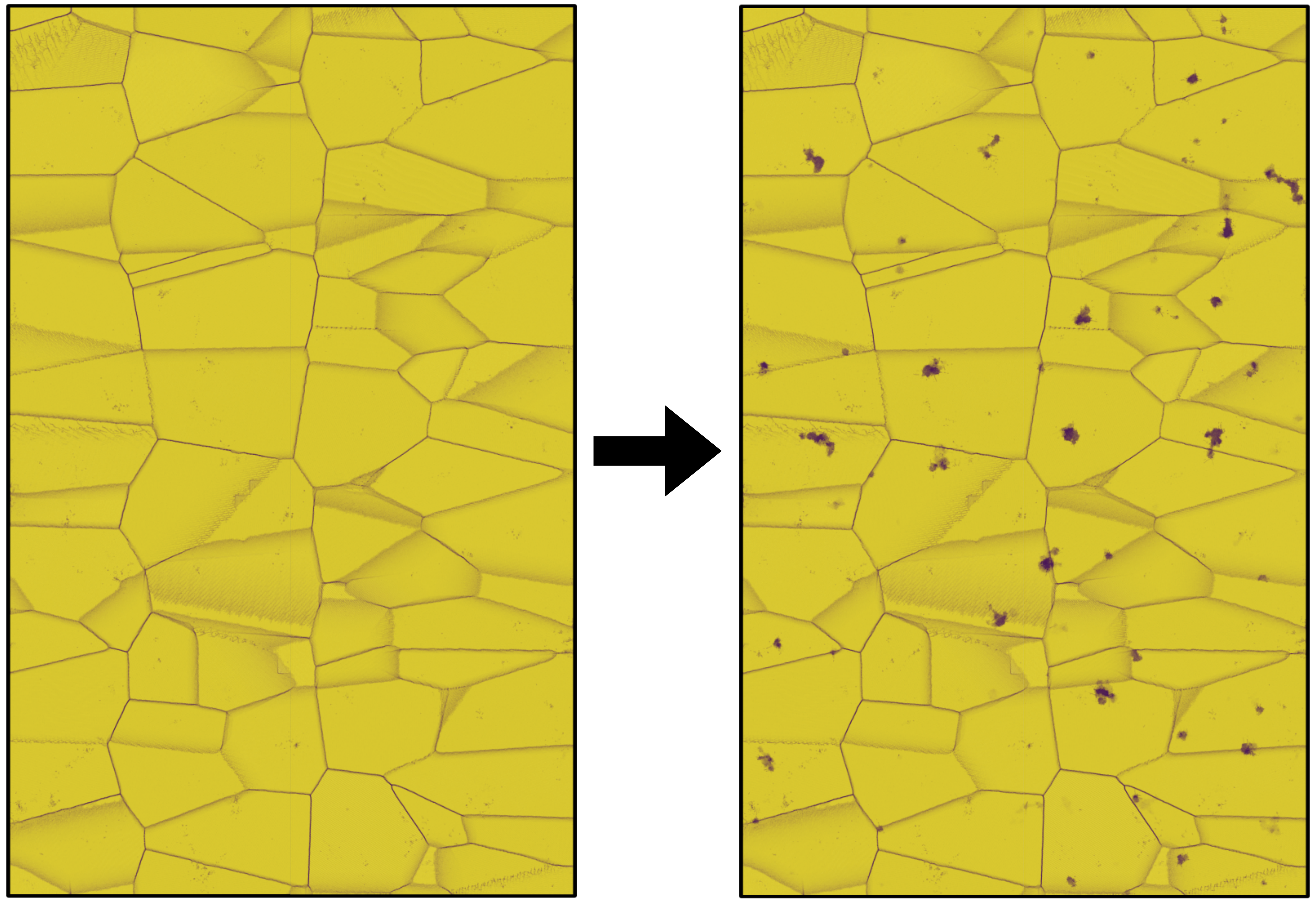
Analysis on Massive-Scale MD Simulation



- Virtual test sample containing 8-billion atoms that mimics W thin film specimen
- Multiple-PKA simulations using *VITMAS*

Future Plans

Analysis on Massive-Scale MD Simulation



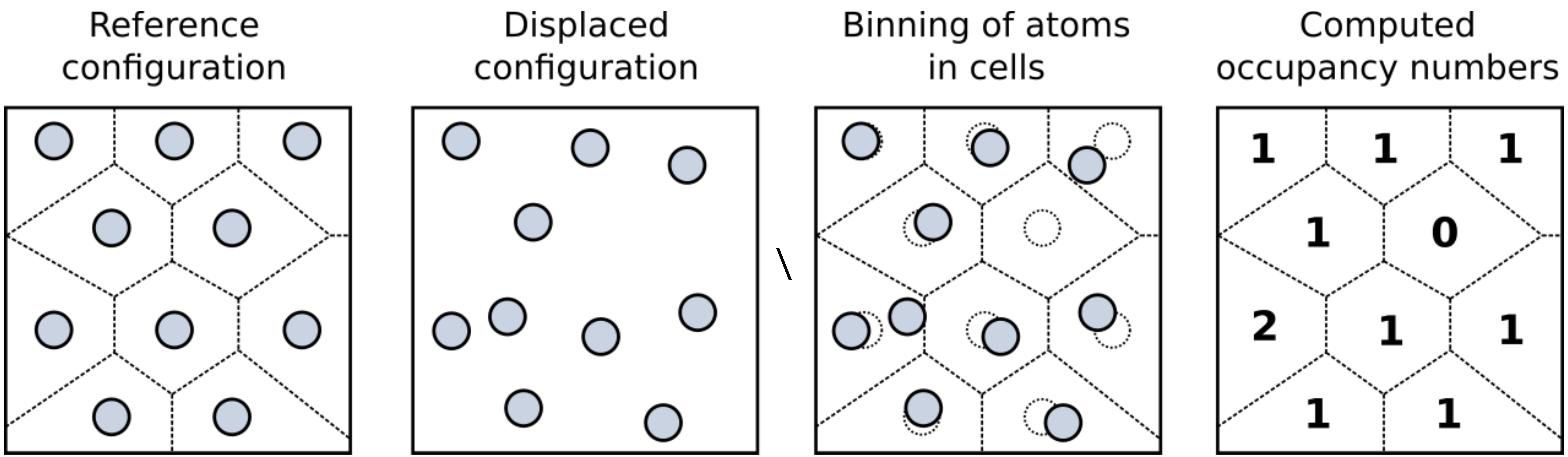
“We’ve created and simulated…!”

“Now… analysis…?”

- Virtual test sample containing 8-billion atoms that mimics W thin film specimen
- Multiple-PKA simulations using *VITMAS*

Future Plans

Wigner-Seitz Method Fails on Polycrystal

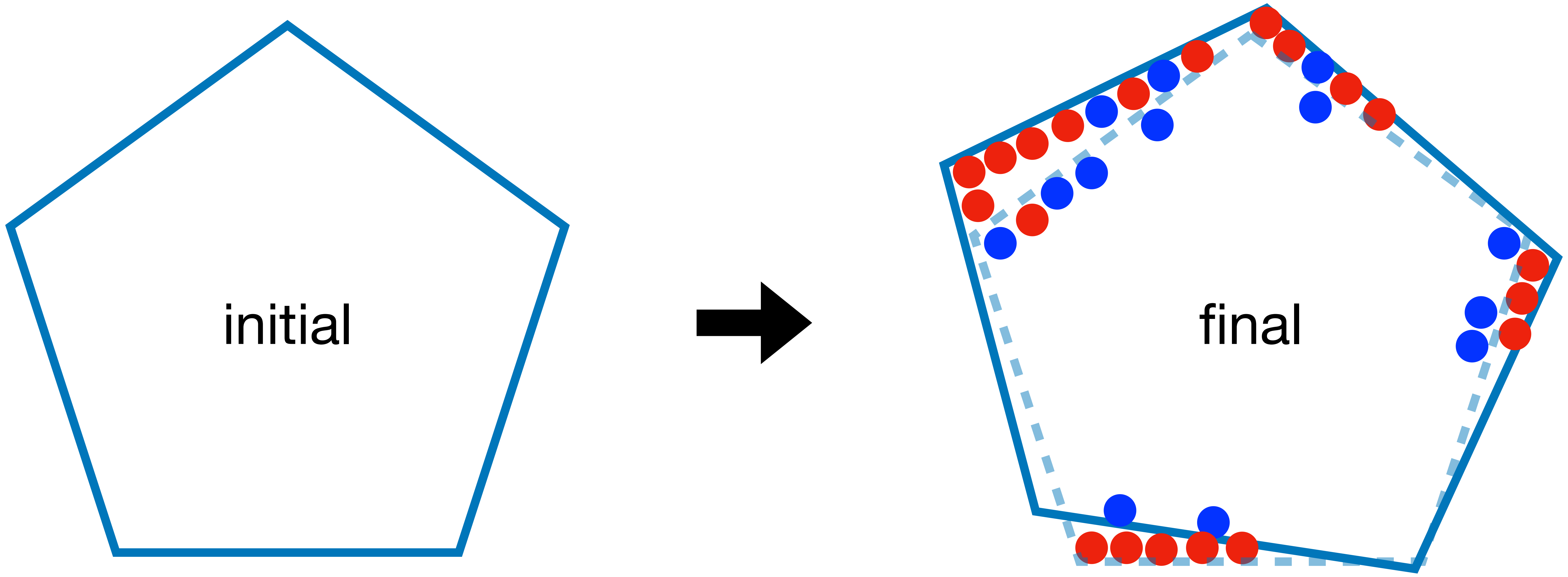


Stukowski, Modelling Simul. Mater. Sci. Eng (2010º

- WS is frequently used to locate and quantify defects in PKA simulations.
- It creates WS cells at the initial and check occupancy; 0: vacancy, >1: interstitial.

Future Plans

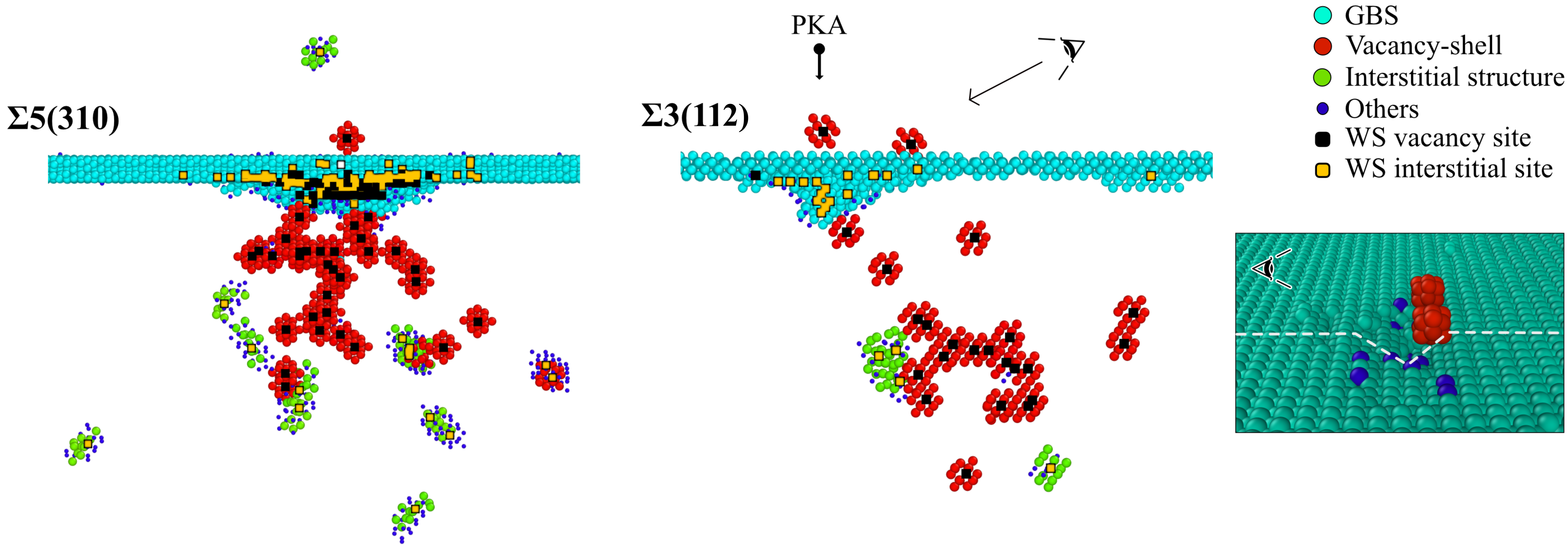
Wigner-Seitz Method Fails on Polycrystal



- Grains rotate, translate and grow.
- W-S cells at the interfaces are larger than those in bulk.
- Miscount of point defects occurs.

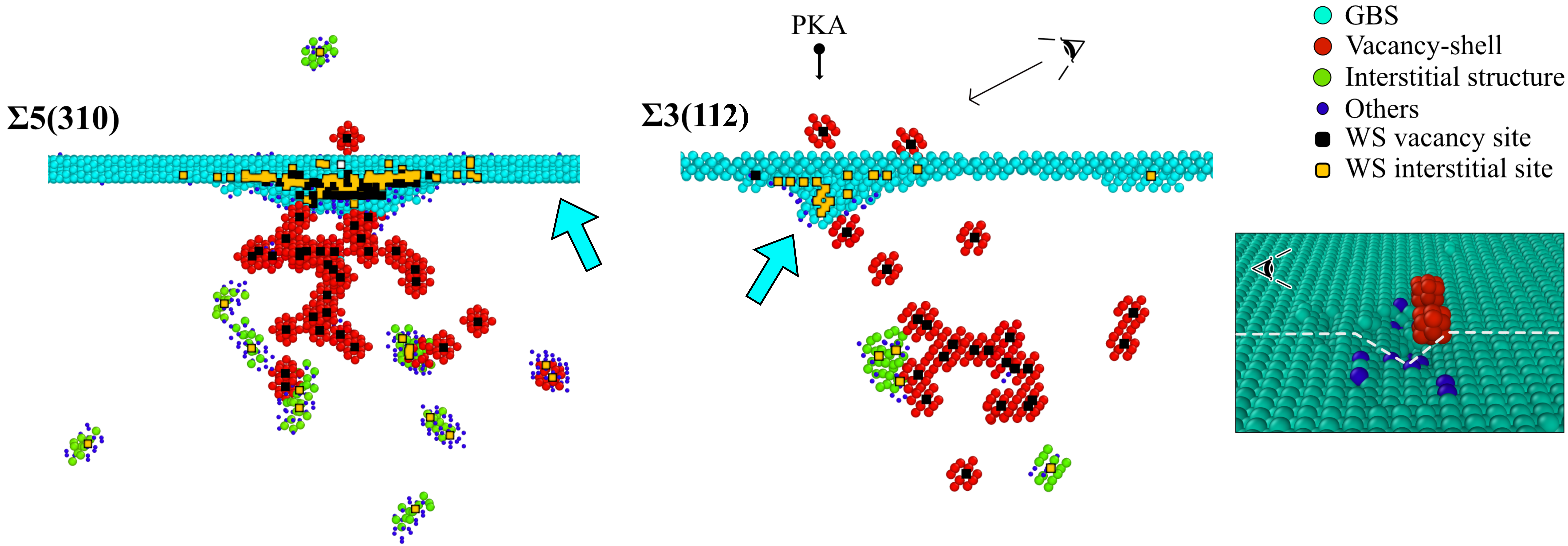
Future Plans

Alternative: CNA + Graph-Theoretical Pattern Recognition



Future Plans

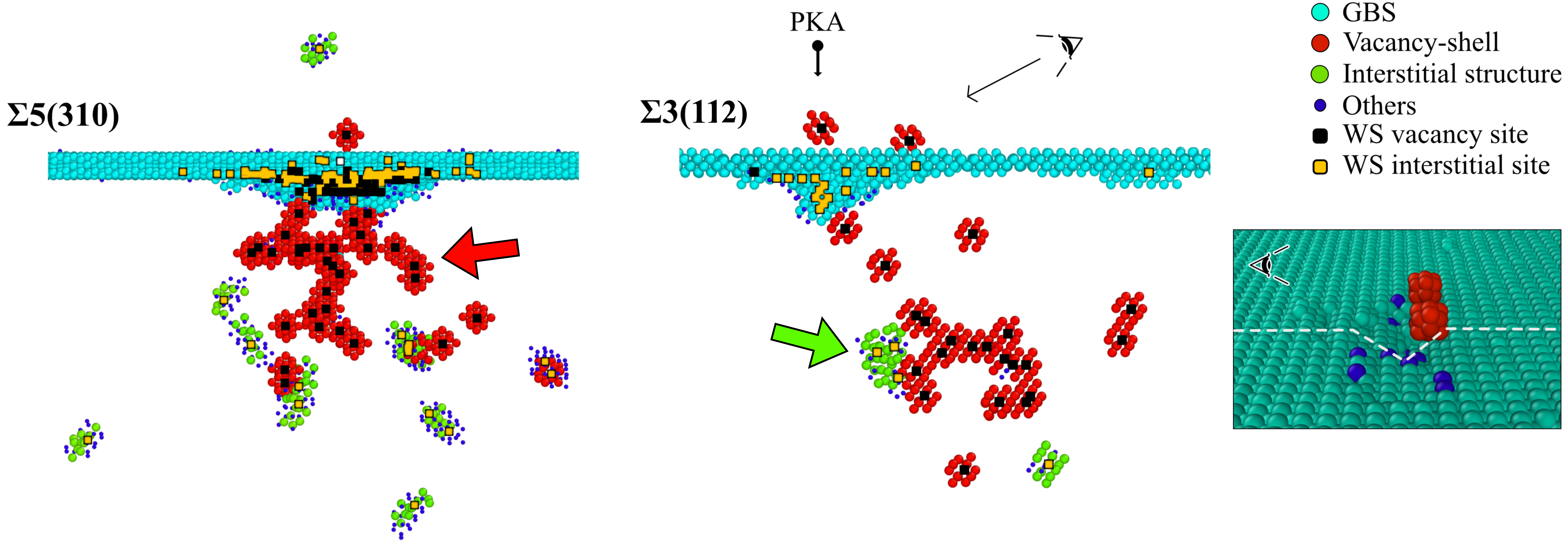
Alternative: CNA + Graph-Theoretical Pattern Recognition



Separation of grain boundary structure (GBS)

Future Plans

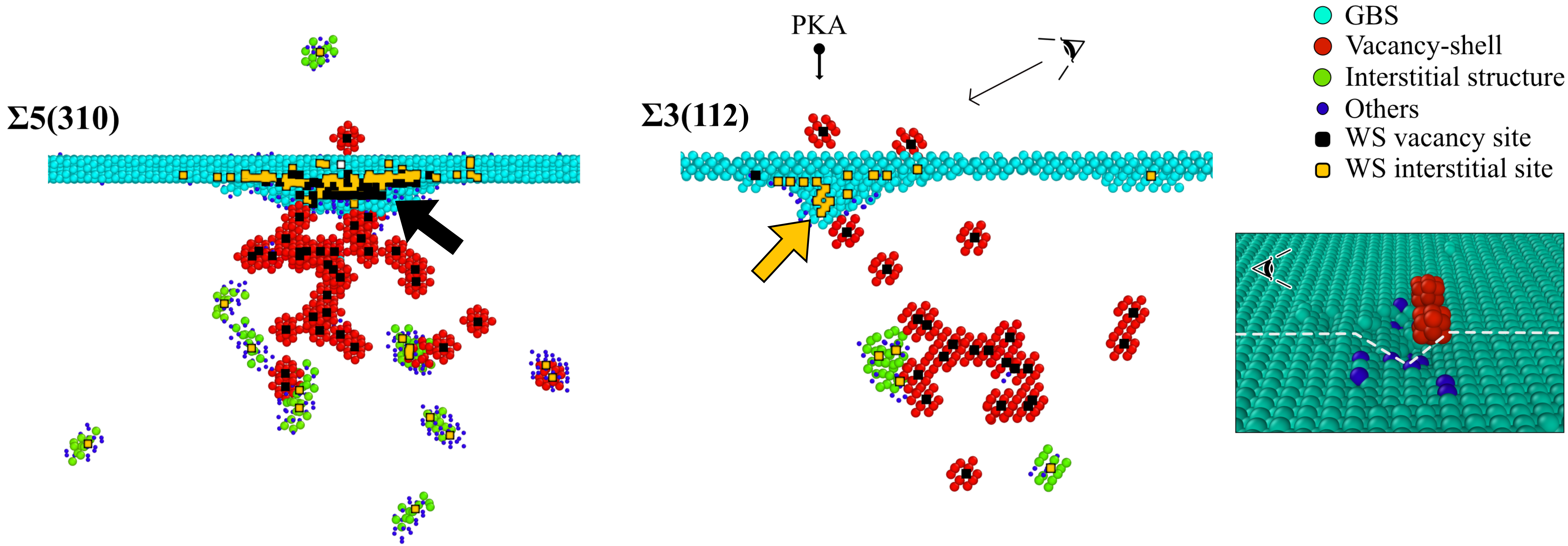
Alternative: CNA + Graph-Theoretical Pattern Recognition



Type-resolved defect identification: 〈110〉 / 〈111〉 interstitial or vacancy

Future Plans

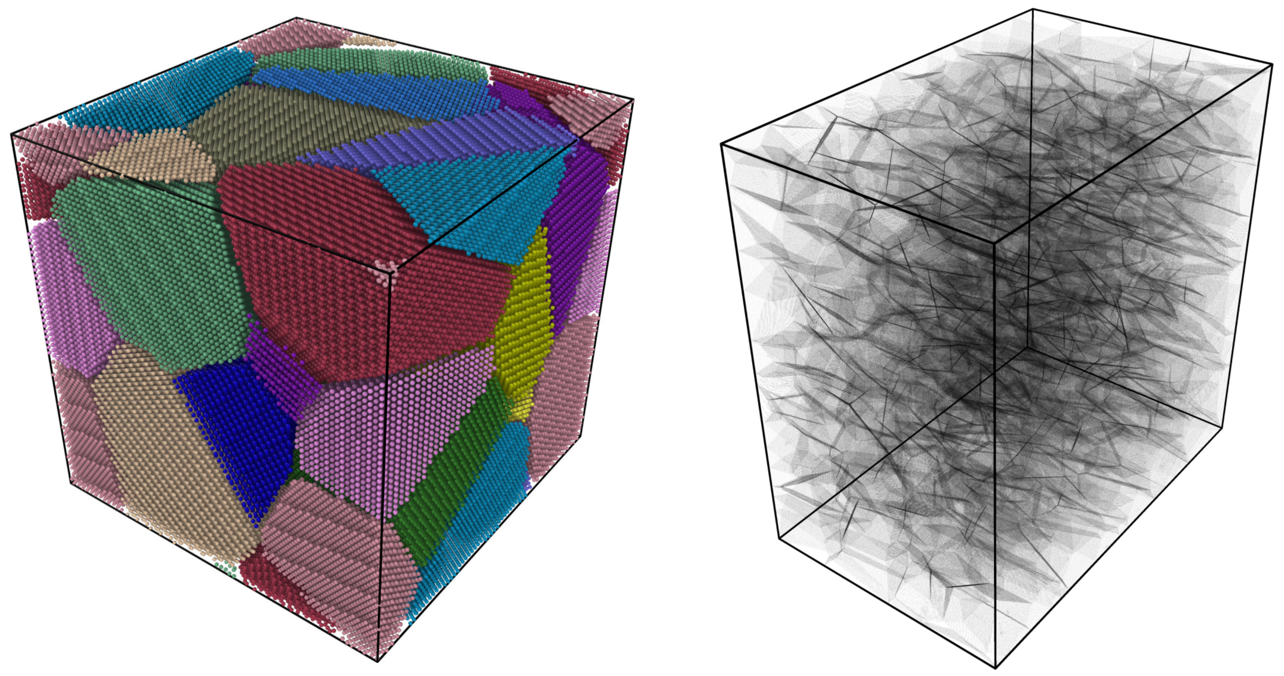
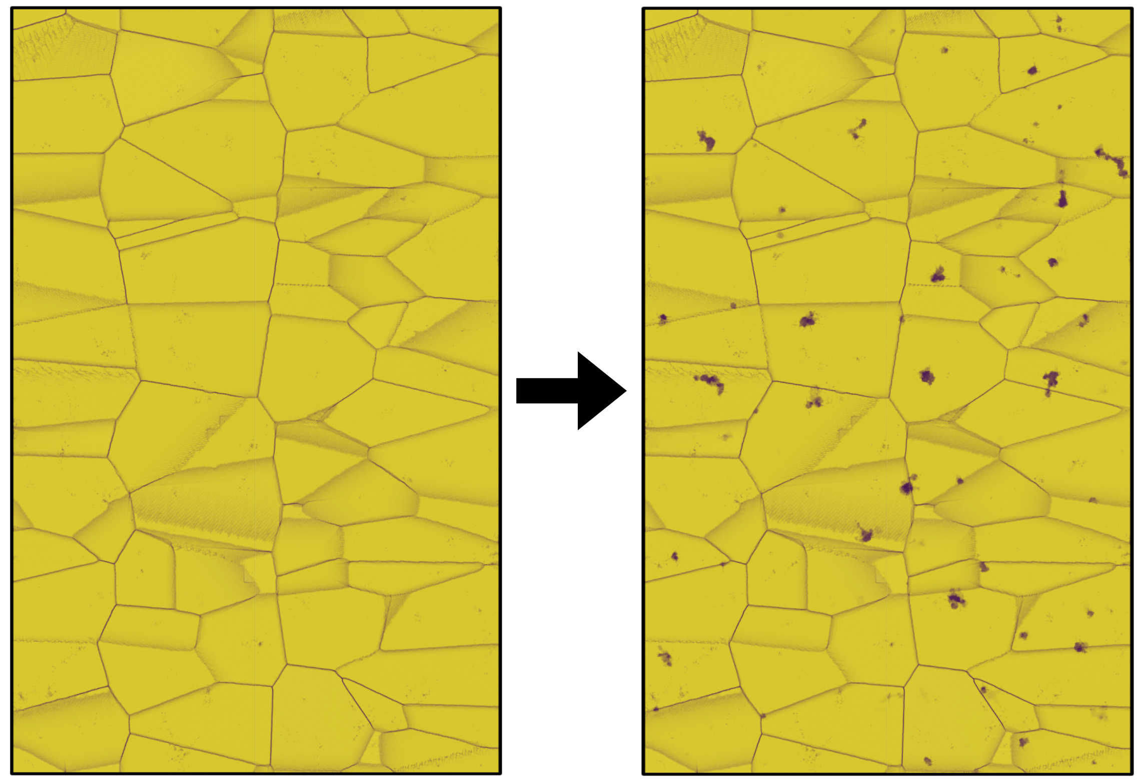
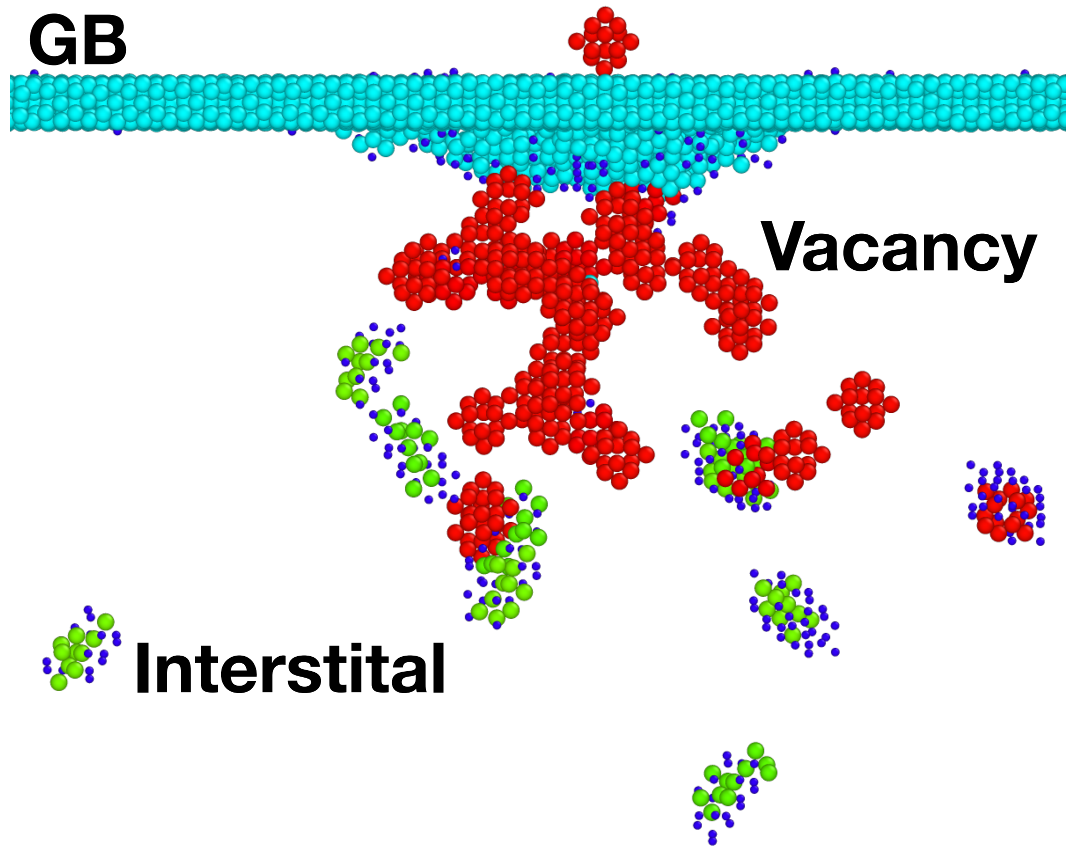
Alternative: CNA + Graph-Theoretical Pattern Recognition



No false detection of point defect at grain boundary

Future Plans

Look into Our Masterplan Again

	Atomic structure generation	MD simulation	Post-processing
Parallel codes	*PolyPal* (published) 	*VITMAS* (developed) 	*Untitled* (in progress) 
Status	Publicly open	Final feature update and testing	Verification in serial

“What I cannot create, I do not understand.”

Richard Feynman

Thank You

This research was supported by the Nano & Material Technology Development Program through the National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT(RS-2024-00445448)