

# **On-lattice interatomic models**

Speakers: Tatiana Kostiuchenko  
Vadim Sotskov

Group leader: Alexander Shapeev

# Lecture content

1. The difference between **off-** and **on-lattice** models
2. Why on-lattice models are actual
3. Monte Carlo methods
4. Commonly used on-lattice models:
  - Cluster expansion method
  - Generalized perturbation method
  - Low-rank interatomic potential
5. About Cython and LRP-code applications

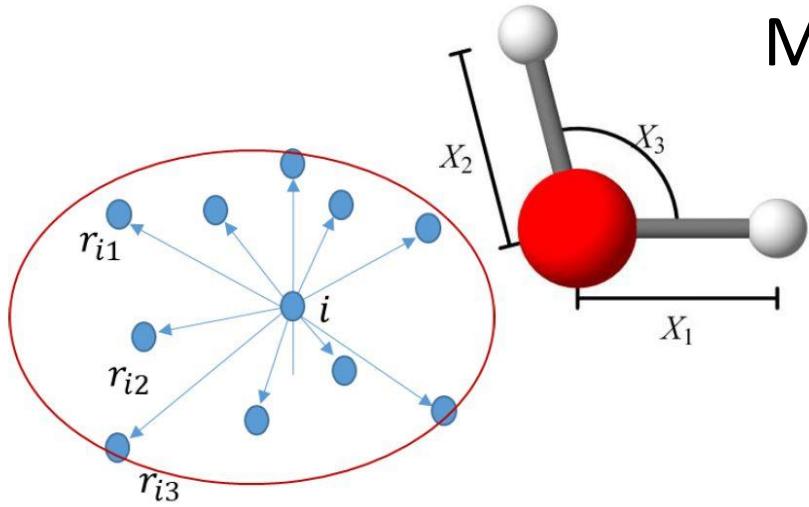
# 1. The difference between **off- and on-lattice** models

# The difference between off- and on-lattice models



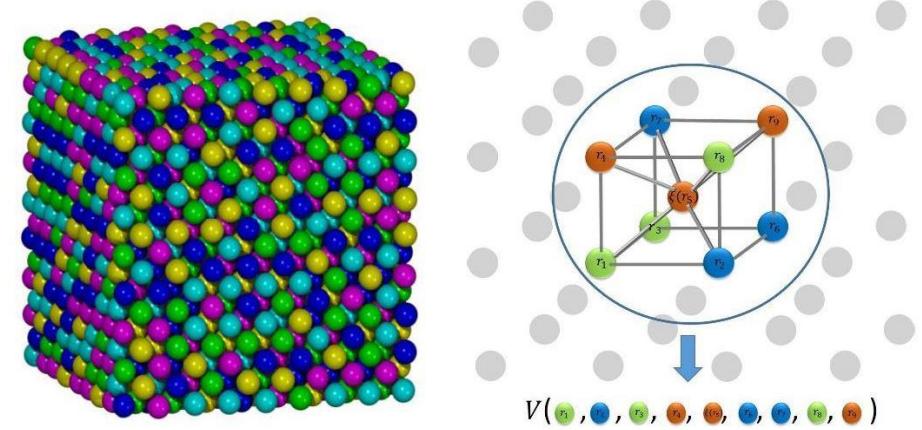
Can account for:  
Stresses and forces

Magnetic moments  
Many-body interactions



Can account for:  
Local distortions

- Fixed crystalline lattice  
Without the possibility of
- account for explicit stresses

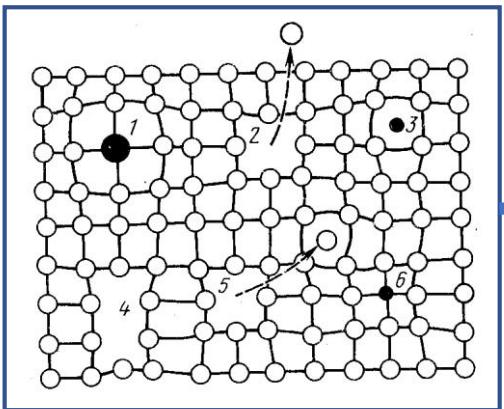


## 2. Why on-lattice models are actual

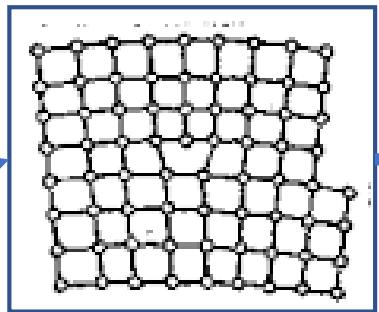
# Why on-lattice models are actual

## Multicomponent alloys

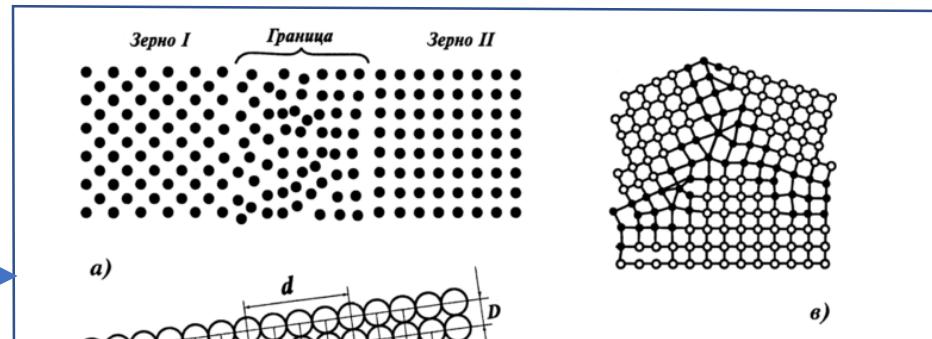
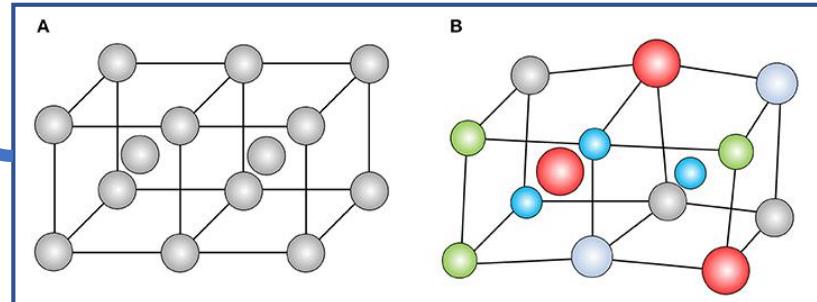
Slowing down the diffusion  
of point defects



Dislocation movement

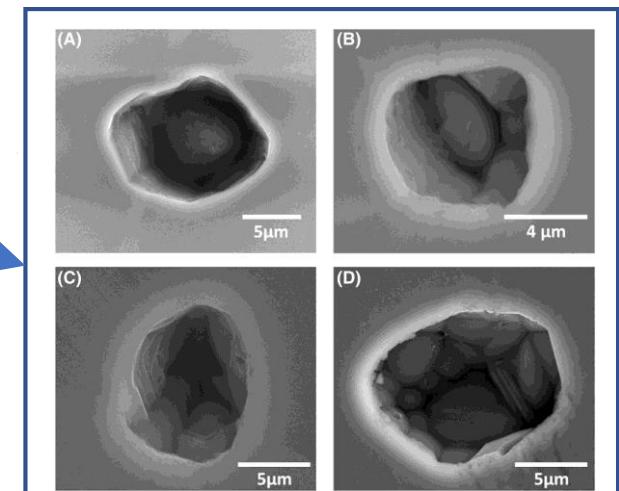


Crystal lattice distortion



Movement of grain boundaries

Formation of bulk defects

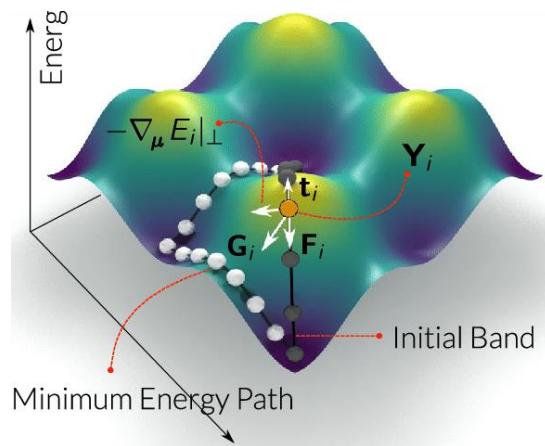
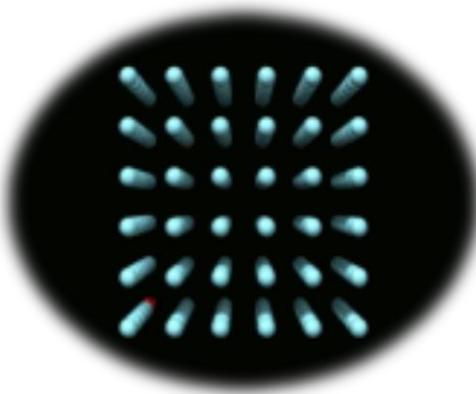


# Why on-lattice models are actual

## Multicomponent alloys

### Molecular dynamics

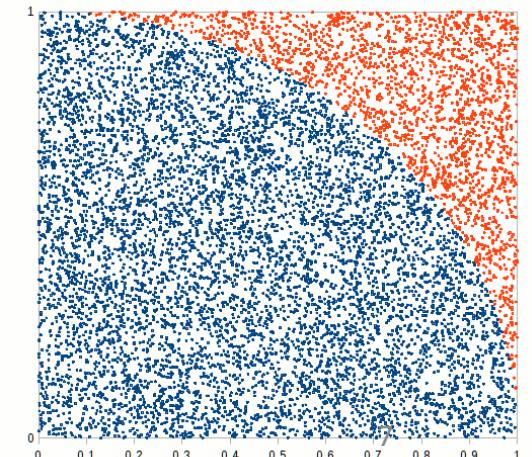
- relies on numerical solution of Newton's equations of motion
- reproduce the dynamics of a system



### Monte-Carlo method

- relies on equilibrium statistical mechanics
- generates states according to appropriate Boltzmann distribution

Ising model:  
mathematical model of  
Ferromagnetism  
in statistical mechanics

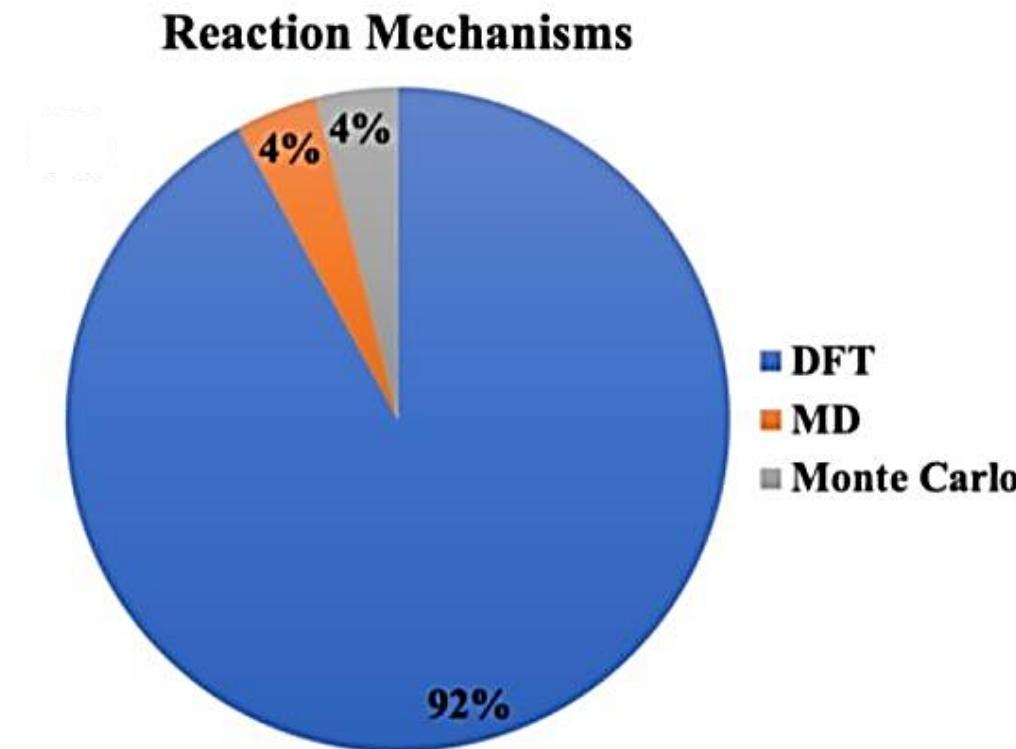


# Why on-lattice models are actual

## Multicomponent alloys

Molecular dynamics (MD)

Monte Carlo method



### 3. Monte Carlo methods

# Monte Carlo methods

## The basics

Mean value of thermodynamic function:

$$\bar{F} = \sum_{A_i} F(A_i) u_i, u_i \geq 0, \sum_i u_i = 1$$

Gibbs distribution:

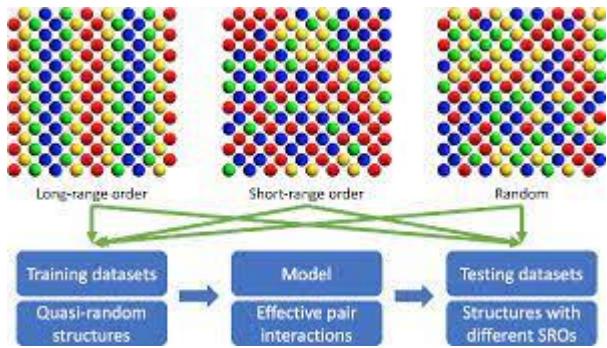
$$u_i = \frac{\exp(-\beta U_N(A_i))}{\sum_{A_j} \exp(-\beta U_N(A_j))},$$

$U_N(A_i)$  – The energy of the system in state  $A_i$

$$\beta = \frac{1}{k_B T}$$

Markov chain + Law of large numbers

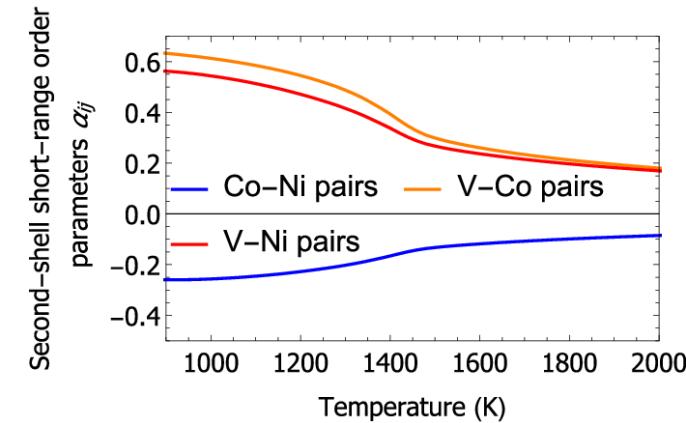
$$\bar{F} = \frac{1}{M} \sum_k^M F(A_{i_k}), A_{i_1}, \dots, A_{i_M} – \text{“events”}$$



# Monte Carlo methods

## Canonical Monte Carlo

1. We randomly **select the node** of crystalline lattice (with a probability  $\frac{1}{N}$ )
2. Randomly **select a neighbor** (neighboring node)
3. Calculate the **energy difference  $\Delta E$**  before and after event.  
The **probability** of the event is  $p = \exp(-\beta\Delta E)$
4. Generate random number  $\epsilon \in [0; 1]$  and **compare** with  $p$ .  
**If  $p > \epsilon$ , then the event is “accepted”**
5. Calculate **thermodynamic functions** for new state, **updated statistics**



# Monte Carlo methods

## Canonical Monte Carlo

1. We randomly **select the node** of crystalline lattice

(with a probability  $\frac{1}{N}$ )

2. Randomly **select a neighbor** (neighboring node)

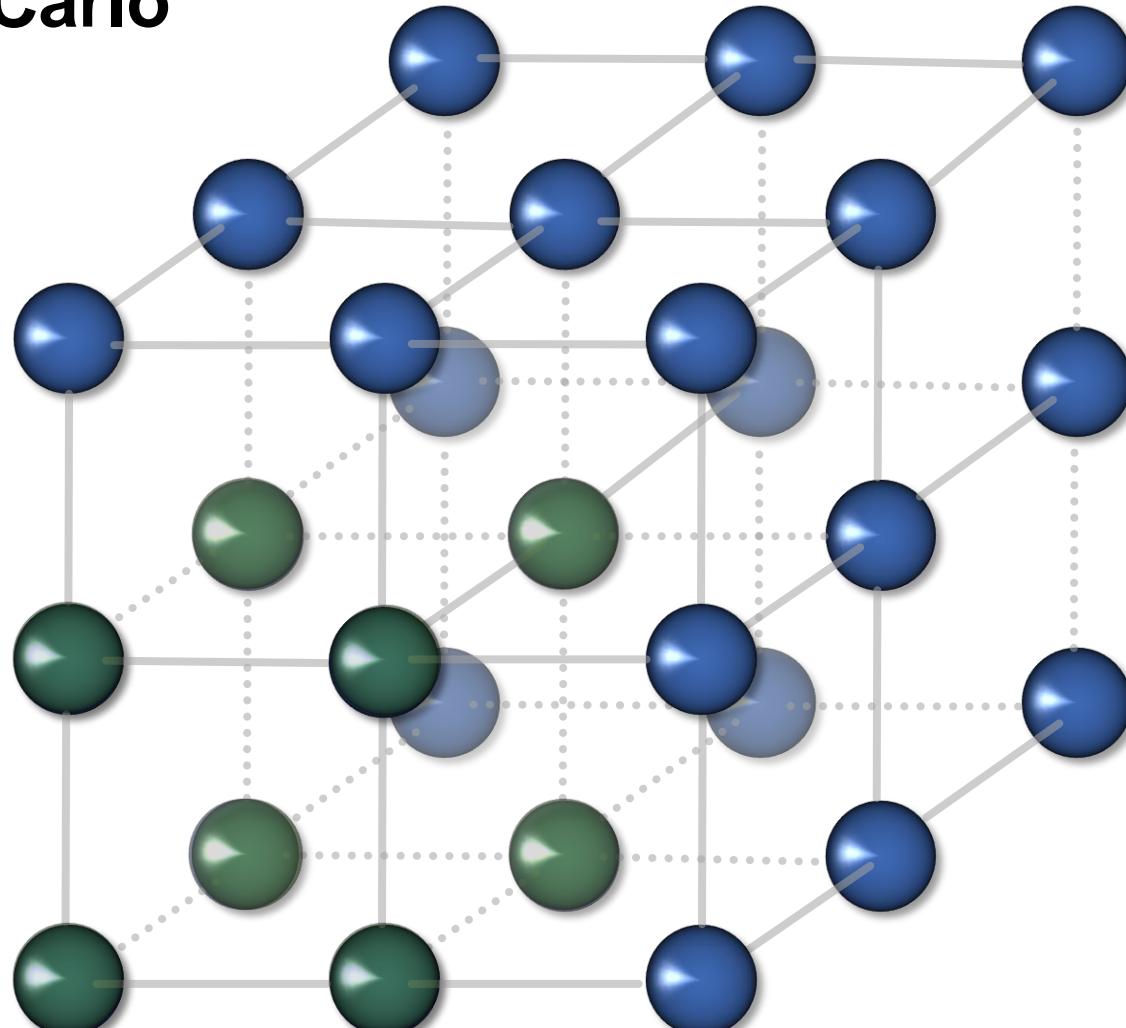
3. Calculate the **energy difference  $\Delta E$**  before and after event.

The **probability** of the event is  $p = \exp(-\beta\Delta E)$

4. Generate random **number  $\epsilon \in [0; 1]$**  and **compare** with  $p$ .

If  $p > \epsilon$ , then the event is “accepted”

5. Calculate **thermodynamic functions** for new state, **updated statistics**



# Monte Carlo methods

## Canonical Monte Carlo

1. We randomly **select the node** of crystalline lattice  
(with a probability  $\frac{1}{N}$ )

2. Randomly **select a neighbor** (neighboring node)

3. Calculate the **energy difference  $\Delta E$**  before and after event.

The **probability** of the event is  $p = \exp(-\beta\Delta E)$

4. Generate random **number  $\epsilon \in [0; 1]$**  and **compare** with  $p$ .

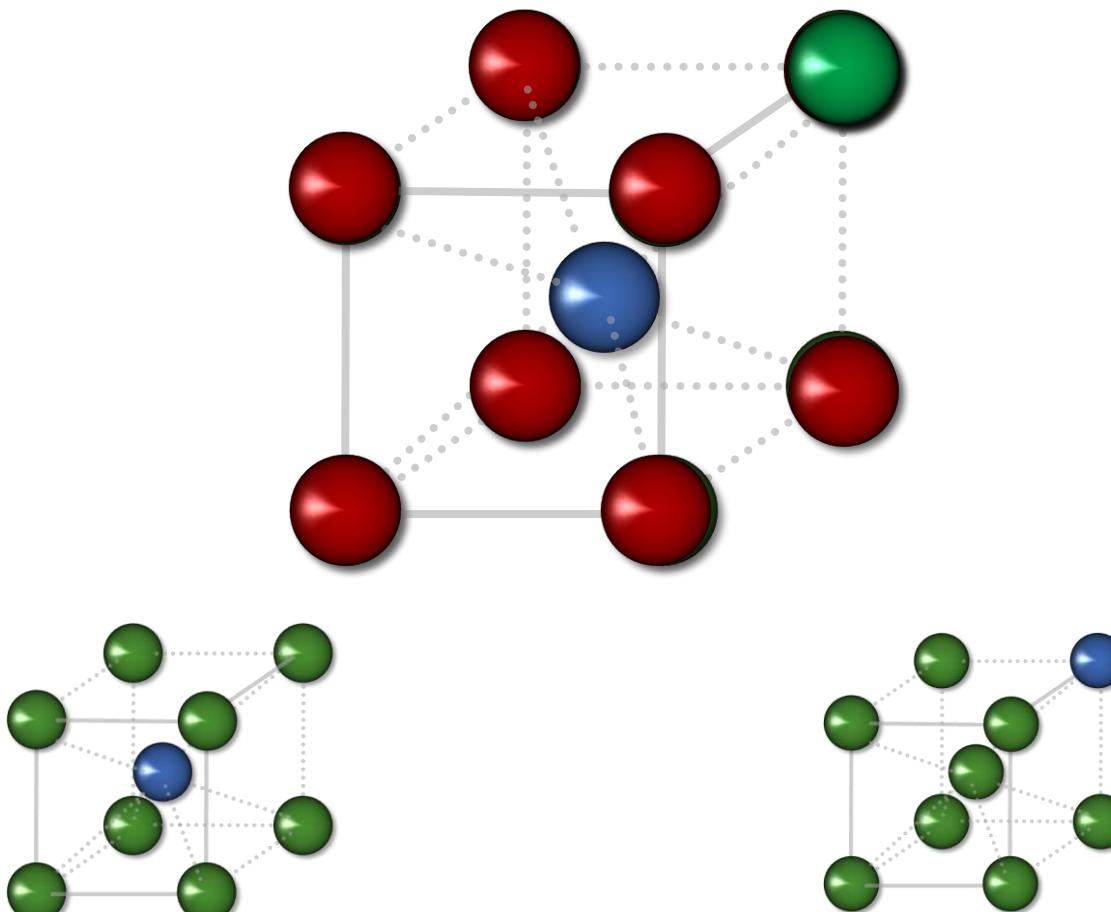
If  $p > \epsilon$ , then the event is “accepted”

IF NOT

5. Calculate **thermodynamic functions** for new state,  
**updated statistics**

$$\Delta E = E_1 - E_2$$

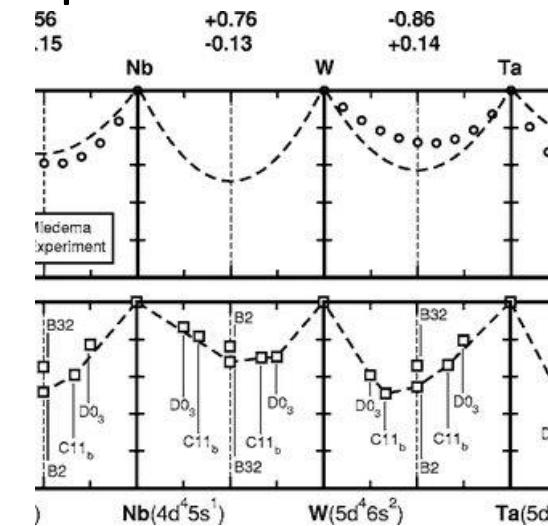
$$\rightarrow E_{current} += \Delta E \text{ or } 0$$



# Monte Carlo methods

## Grand Canonical ( $\mu VT$ ) Monte Carlo

1. We randomly **select the node** of crystalline lattice (with a probability  $\frac{1}{N}$ )
2. Randomly **select another chemical element** (from the system) for replacement
3. Calculate the **energy difference  $\Delta E$**  before and after event  
The **probability** of the event is  $p = \exp(\mu N - \beta \Delta E)$
4. Generate random **number  $\epsilon \in [0; 1]$**  and **compare** with  $p$ .  
**If  $p > \epsilon$ , then the event is “accepted”**
5. Calculate **thermodynamic functions** for new state, **updated statistics**



macrocanonical ensemble = thermodynamic equilibrium (thermal and chemical)

# Monte Carlo methods

## Grand Canonical ( $\mu VT$ ) Monte Carlo

1. We randomly **select the node** of crystalline lattice (with a probability  $\frac{1}{N}$ )

2. Randomly **select another chemical element** (from the system) for replacement

3. Calculate the **energy difference  $\Delta E$**  before and after event

The **probability** of the event is  $p = \exp(\mu N - \beta \Delta E)$

4. Generate random **number  $\epsilon \in [0; 1]$**  and **compare** with  $p$ .

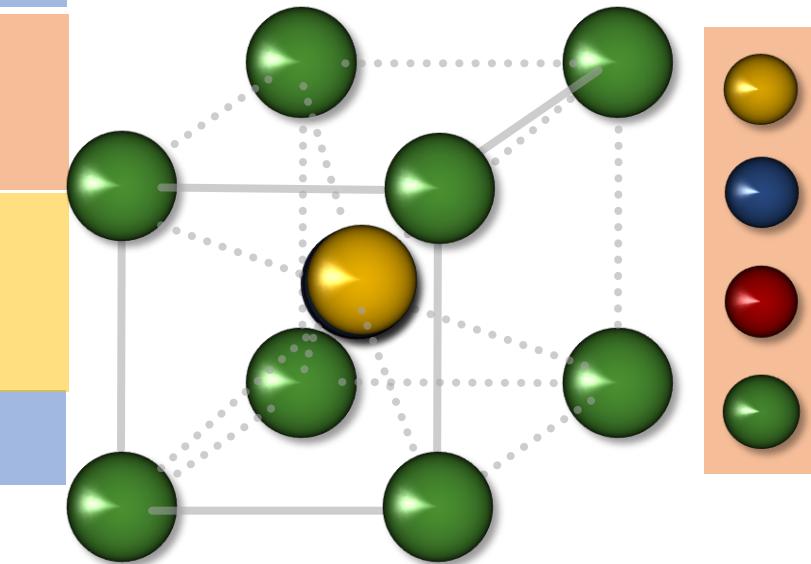
If  $p > \epsilon$ , then the event is “accepted”

IF NOT

5. Calculate **thermodynamic functions** for new state, **updated statistics**

macrocanonical ensemble = thermodynamic equilibrium (thermal and chemical)

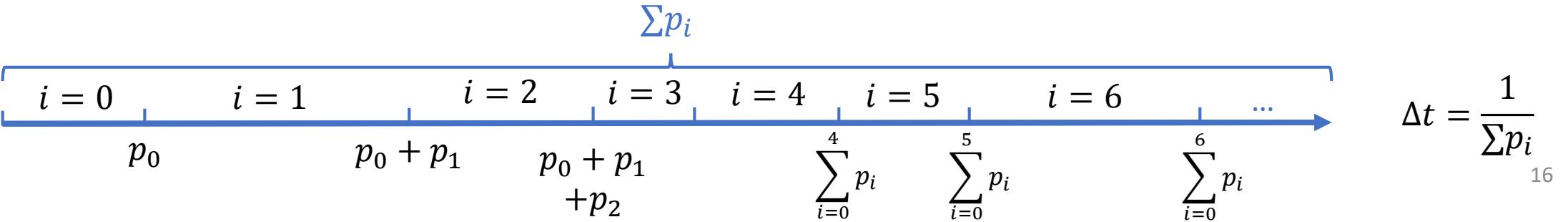
$$\Delta E = E_1 - E_2 \\ \rightarrow E_{current} += \Delta E \text{ or } 0$$



# Monte Carlo methods

## Kinetic Monte Carlo

1.  $t = 0$ , start with initial state  $k$
  2. Make a list of states  $i$  that can be obtained from  $k$  with one event step
  3. Calculate the common rate as a  $\sum p_i$ , where  $p_i$  is a probability of  $k$  transfer to  $i$
- Thus we get the sum of probabilities.
4. Generate random **number**  $\xi \in [0; 1]$ . **Compare**  $\xi \cdot \sum p_i$  and detect which state  $i$  we come to with an event
  5. Calculate **thermodynamic functions** for new state, **updated statistics**,  $t = t + \Delta t$



# Monte Carlo methods

## Kinetic Monte Carlo

1.  $t = 0$ , start with initial state  $k$

2. Make a list of states  $i$  that can be obtained from  $k$  with one event step

3. Calculate the common rate as a  $\sum p_i$ , where  $p_i$  is a probability of  $k$  transfer to  $i$

Thus we get the sum of probabilities.

4. Generate random number  $\xi \in [0; 1]$ . Compare  $\xi \cdot \sum p_i$  and detect which state  $i$  we come to with an event

5. Calculate **thermodynamic functions** for new state, **updated statistics**,  $t = t + \Delta t$

$i = 0$

$i = 1$

$i = 2$

$i = 3$

$i = 4$

$i = 5$

$i = 6$

$\dots$

$p_0$

$p_0 + p_1$

$p_0 + p_1 + p_2$

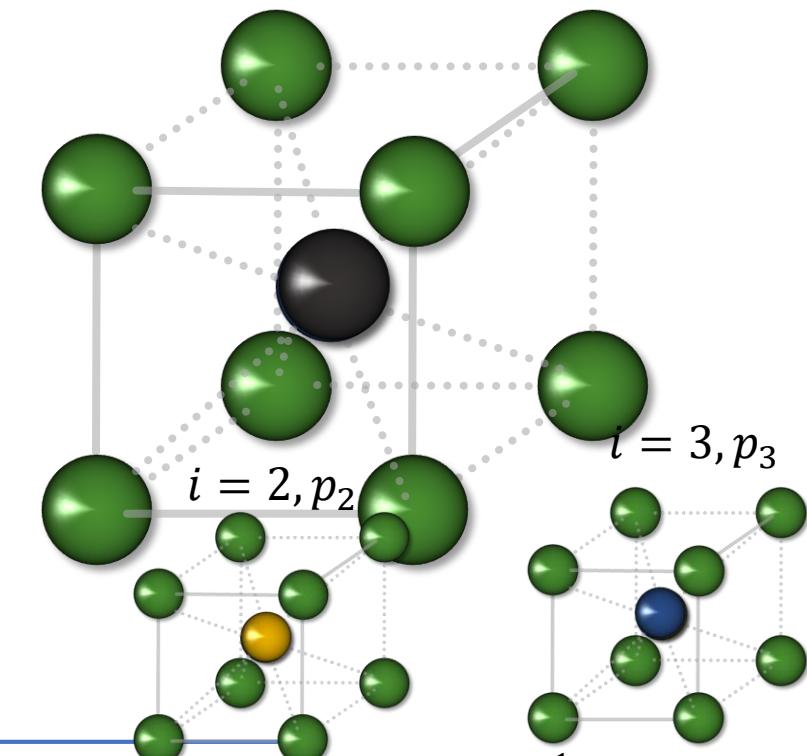
$\sum_{i=0}^4 p_i$

$\sum_{i=0}^5 p_i$

$\sum_{i=0}^6 p_i$

$\sum p_i$

$k$



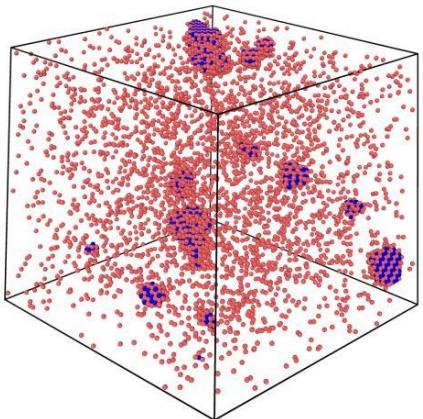
$$\Delta t = \frac{1}{\sum p_i}$$

# Monte Carlo methods

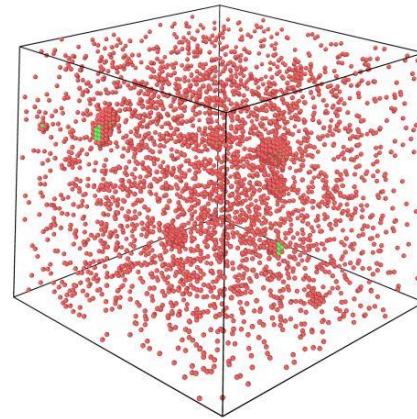
## Kinetic Monte Carlo

Mechanism of Re precipitation in irradiated W-Re alloys from kinetic Monte Carlo simulations

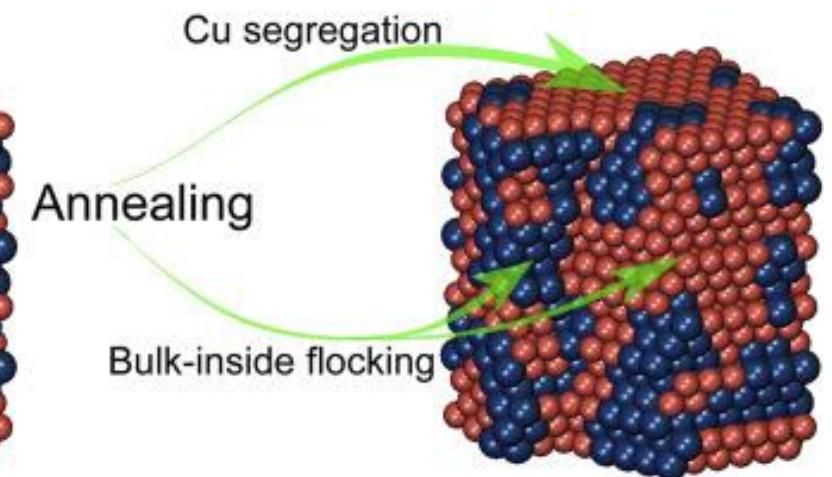
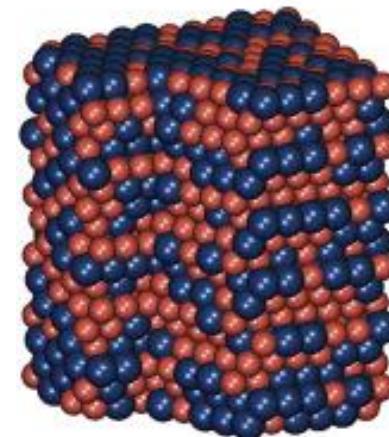
[doi: 10.1103/PhysRevB.96.094108](https://doi.org/10.1103/PhysRevB.96.094108)



(a) W-1.8at%Re alloy, 0.5 at% vacancy concentration.



(b) W-1.4at%Re alloy, 0.1 at% mixed-interstitials.



## 4. Commonly used on-lattice models

## Commonly used on-lattice models

- Cluster expansion method
- Generalized perturbation method
- Low-rank interatomic potential

# Commonly used on-lattice models

## Cluster expansion method

Energy of the system with determined cfg

$J_\alpha$  – fit to first principal calculations

Empty cluster energy  
cluster

$$E(\sigma) = J_{\text{null}} + \sum_i J_i \sigma_i + \sum_{ij} J_{ij} \sigma_i \sigma_j + \sum_{ijk} J_{ijk} \sigma_i \sigma_j \sigma_k + \dots$$

**The further the accurate...**

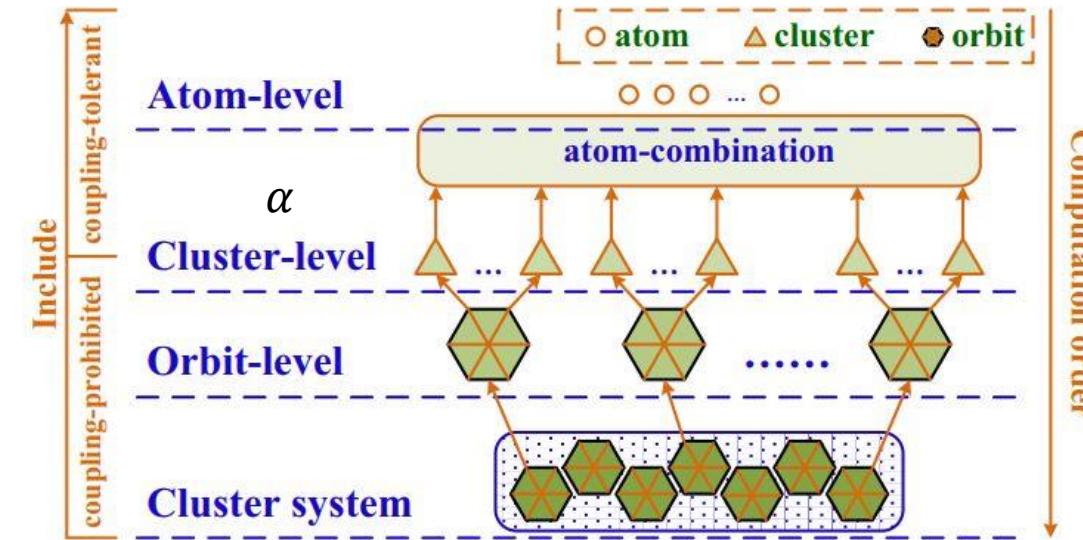
Point cluster	pair cluster	triplets cluster
------------------	-----------------	---------------------

Effective cluster interaction

$$m_\alpha J_\alpha \langle \prod_{p \in \alpha'} \sigma_p \rangle$$

site

Atoms positioning



Example for binary alloy

# Commonly used on-lattice models

## Effective interaction model

$J_{ij}$  exchange-coupling parameter for the magnetic moments

$$H = \frac{1}{2} \sum_i \sum_j V_{ij} + \frac{1}{2} \sum_i \sum_j J_{ij} \overrightarrow{M}_i \overrightarrow{M}_j , \quad \overrightarrow{M}_i = (M_i, \phi, \theta)$$

chemical interactions between atoms

$\Delta E = \frac{1}{2} \sum_i \sum_j J_{ij} \cdot \Delta(\overrightarrow{M}_i \overrightarrow{M}_j)$  – fit for same chem cfg, but different magnetic states

**Table 1**

Fitted parameters (in meV) for the magnetic ( $J_{ij}$ ) and chemical ( $V_{ij}$ ) pair interactions in the EIM (Eq. (1)).

	1nn	2nn	3nn	4nn	5nn
$J_{Fe-Fe}$	-11.65800	-1.67630	0.51512	0.49840	0.22120
$J_{Co-Co}$	-7.60200	-6.58240	-5.16990	1.95370	0.13135
$J_{Fe-Co}$	-5.98790	-2.69460	-2.00000	-0.04000	0.01000
$V_{Fe-Fe}$	10.62000	-4.94300	-	-	-
$V_{Co-Co}$	10.62000	-4.94300	-	-	-
$V_{Fe-Co}$	-11.92500	4.10800	-	-	-

# Commonly used on-lattice models

# Generalized perturbation method

+Effective interaction model

- 1) Effective medium (homogeneous random alloy without any short-range order) which represents a random alloy configuration on average.

$$\widetilde{V_S^{(n)}} = \frac{1}{2n} V_S^{(n)}$$

## Concentration-variable effective interaction from GPM

- 2) Consider the energetics of the corresponding clusters embedded in this effective medium.

Configuration determination	$\xi_S^{(n)} = \frac{1}{n} \sum_{p \in S} \prod_{i=1,n} \sigma_{i_p}$	Geometry class	Sites in cluster
--------------------------------	---	-------------------	---------------------

$$E_{conf} = \sum_{n,S} V_S^{(n)} \xi_S^{(n)}$$

Effective  
interaction

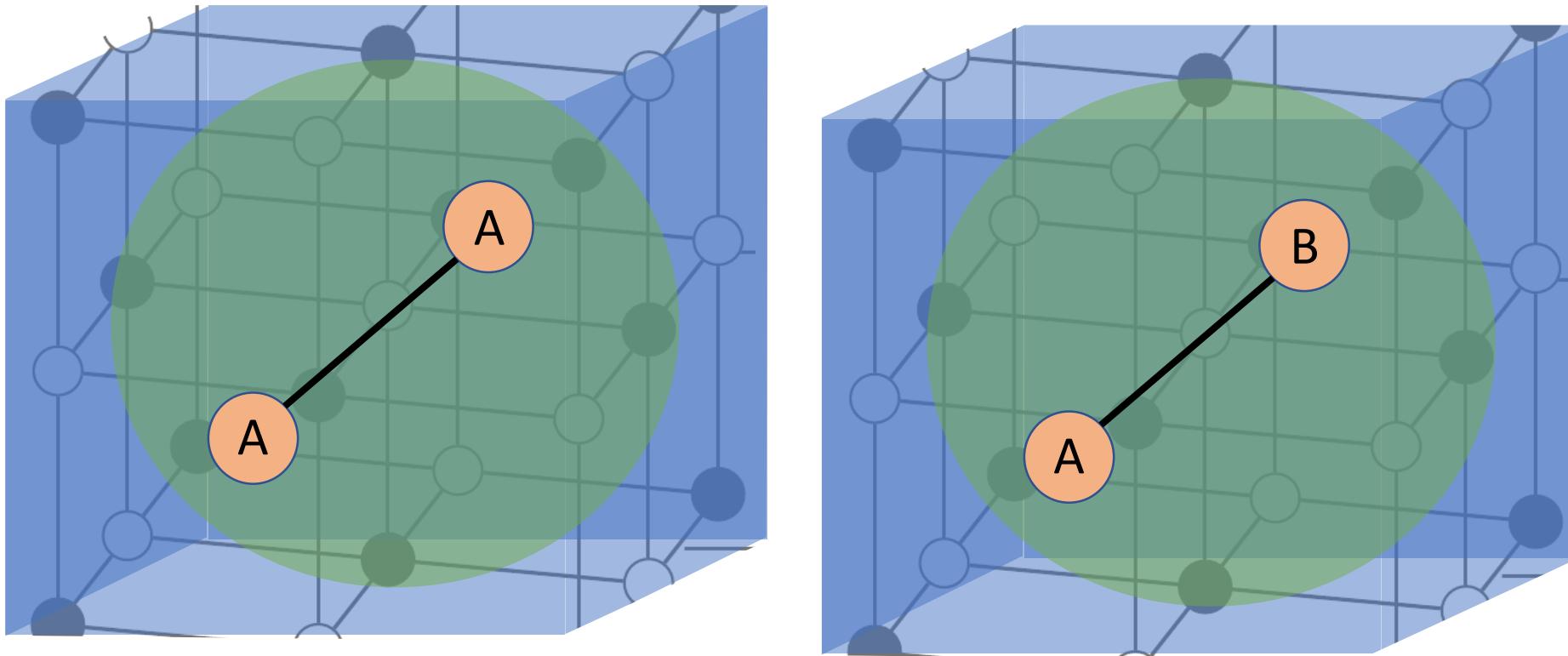
$$\xi_S^{(n)} = \langle \sigma \rangle^n$$

all the clusters with  
an even (odd) number  
of A atoms embedded

# Commonly used on-lattice models

## Generalized perturbation method

### +Effective interaction model



Effective medium = homogeneous random alloy

# Commonly used on-lattice models

## Low-rank interatomic potential

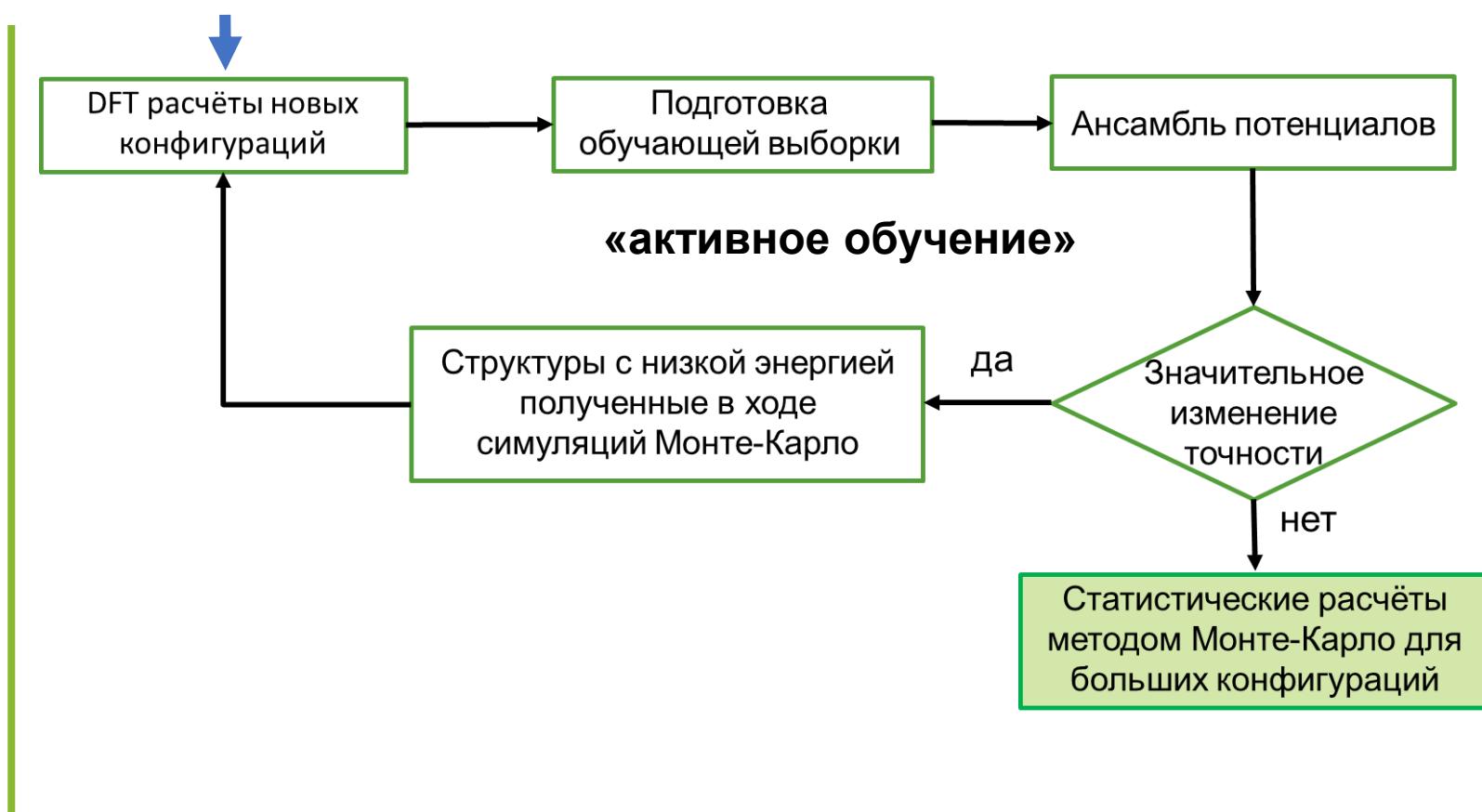
### Methodology

#### Общая схема решения

Теория функционала плотности  
(DFT)  
расчёты для небольших конфигураций

Машинно-обучаемый потенциал  
«на решётке» (LRP) [1]

Канонический Монте-Карло  
для больших конфигураций



[1] Alexander Shapeev. Accurate representation of formation energies of crystalline alloys with many components. Computational Materials Science, 139:26-30, 2017 (The Low-Rank Interatomic Potential)

# Commonly used on-lattice models

## Low-rank interatomic potential

Представление энергии атомной системы:

$$E(\sigma) = \sum_{\xi \in \Omega} V(\sigma(\xi + r_1), \dots, \sigma(\xi + r_9)) \quad (1)$$

химические типы ближайших  
атомов-соседей

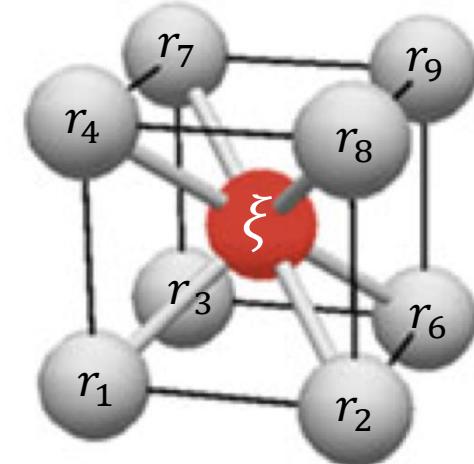
$V$  — потенциал, тензор с  $4^9$  параметрами.

**Разложение «тензорного поезда» [3]** позволяет уменьшить число параметров до  $4 \cdot 9 \cdot 5^2$  ( $\text{rank} = 5$ ).  
Задача регрессии:

$$\frac{1}{K} \sum_{k=1}^K |E(\sigma^{(k)}) - E^{\text{qm}}(\sigma^{(k)})|^2 \quad (2)$$

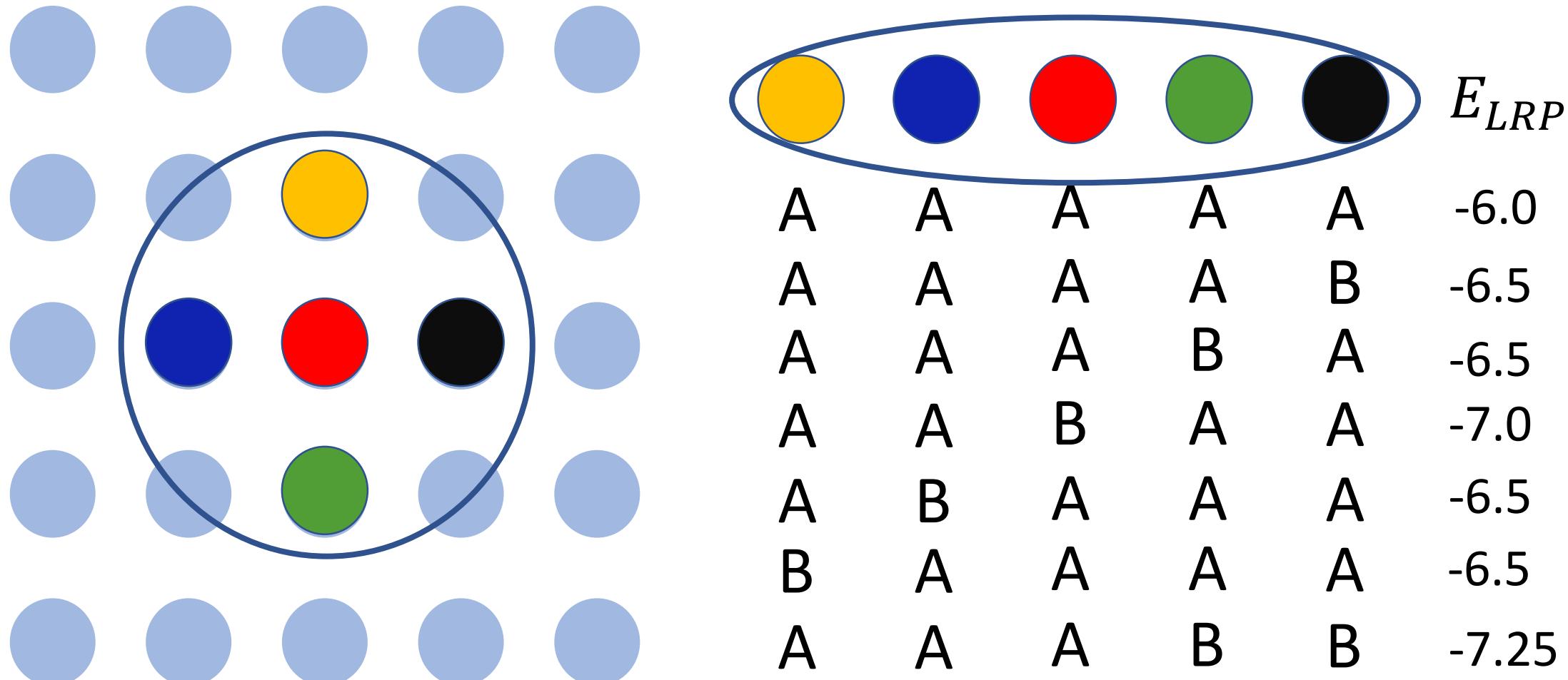
DFT

$$\sigma(x) = \left\{ \begin{array}{l} \text{Nb} \\ \text{Mo} \\ \text{Ta} \\ \text{W} \end{array} \right\}$$



# Commonly used on-lattice models

## Low-rank interatomic potential

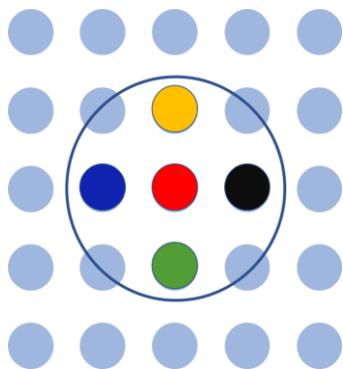


[1] Alexander Shapeev. Accurate representation of formation energies of crystalline alloys with many components. Computational Materials Science, 139:26-30, 2017 (The Low-Rank Interatomic Potential),  
[3] Ivan V Oseledets. Tensor-train decomposition. SIAM Journal on Scientific Computing, 33(5):2295-2317, 2011.

# Commonly used on-lattice models

## Low-rank interatomic potential

Разложение «тензорного поезда» [3] позволяет уменьшить число параметров до  $4 \cdot 9 \cdot 5^2$  ( $rank = 5$ ).



E <sub>LRP</sub>					
A	A	A	A	A	-6.0
A	A	A	A	B	-6.5
A	A	A	B	A	-6.5
A	A	B	A	A	-7.0
A	B	A	A	A	-6.5
B	A	A	A	A	-6.5
A	A	A	B	B	-7.25

$$A_{ij} = \begin{array}{c} \text{[blue bar]} \\ \times \\ \text{[blue bar]} \\ \times \\ \text{[blue bar]} \end{array}$$

In BCC 9-dimensional tensor

$$\begin{array}{c} \text{[blue 3x3 grid]} \\ = \\ \text{[blue vertical bar]} \times \text{[blue horizontal bar]} \end{array}$$

$$A_{23} = \begin{array}{c} i=2 \\ \parallel \\ j=3 \end{array} \times$$

[1] Alexander Shapeev. Accurate representation of formation energies of crystalline alloys with many components. Computational Materials Science, 139:26-30, 2017 (The Low-Rank Interatomic Potential),

[3] Ivan V Oseledets. Tensor-train decomposition. SIAM Journal on Scientific Computing, 33(5):2295-2317, 2011.

# Commonly used on-lattice models

## Low-rank interatomic potential

A diagram illustrating the low-rank interatomic potential. On the left, a blue 3D cube represents a 3D tensor. A green arrow points from this cube down to a 1D vector representation below. The vector consists of three blue rectangular blocks labeled  $i$ ,  $j$ , and  $k$  from left to right, separated by multiplication signs ( $\times$ ).

$$A_{ijk} = i \times j \times k$$

Tensor train (TT)  $nmr^2$

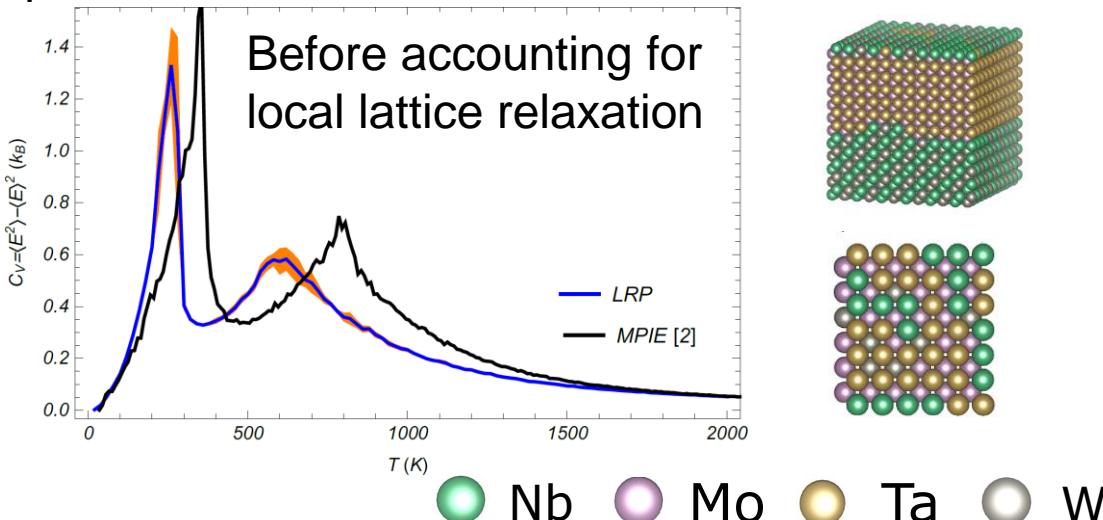
A diagram illustrating the Tensor Train (TT) decomposition of a tensor  $A_{2324}$ . The tensor is shown as a sequence of four tensors (matrices) connected by multiplication signs ( $\times$ ). The ranks of these matrices are indicated by red arrows: 2, 3, 2, and 4. Above the sequence, the total rank  $n = 4$  is specified. Blue arrows above the matrices indicate their dimensions:  $r \times 1$ ,  $r \times r$ ,  $r \times r$ , and  $1 \times r$ .

$$A_{2324} =$$

# Low-rank interatomic potential

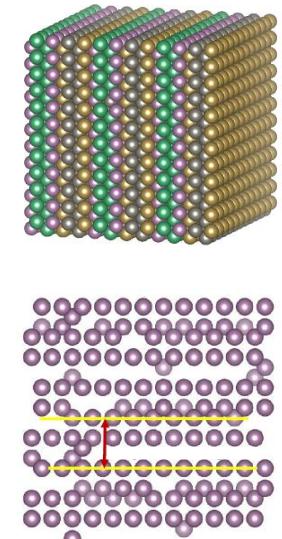
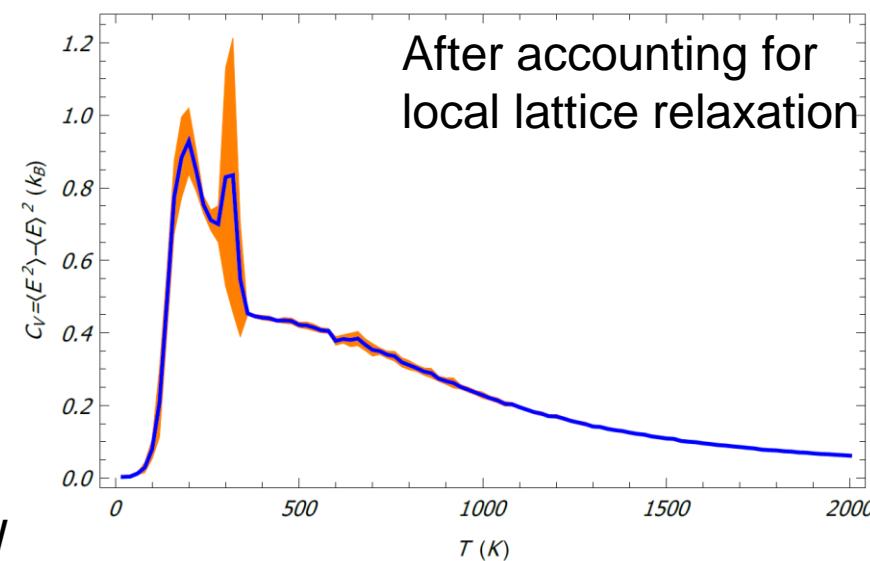
- A qualitatively new methodology has been developed and tested for the study of complex multicomponent alloys with low-rank machine learning potentials
- Investigation of the influence of local lattice distortions in the equiatomic system Nb-Mo-Ta-W

Impact of lattice relaxations on phase transitions in a high-entropy alloy studied by machine-learning potentials



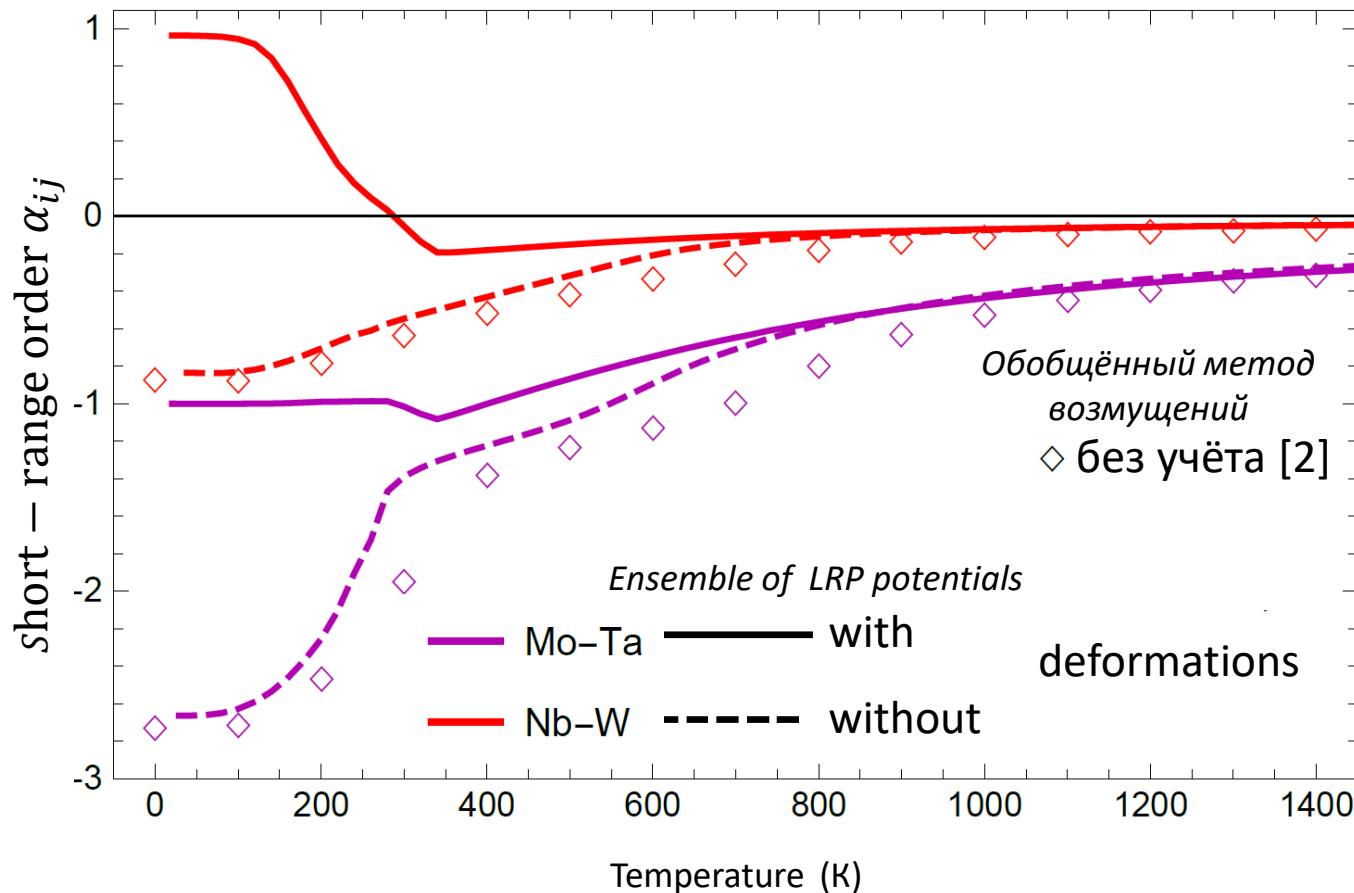
npj nature partner journals

Collaborators: Fritz Körmann  
Jörg Neugebauer



# Low-rank interatomic potential

## With and without taking into account local deformations



Warren–Cowley short-range  
order parameters:

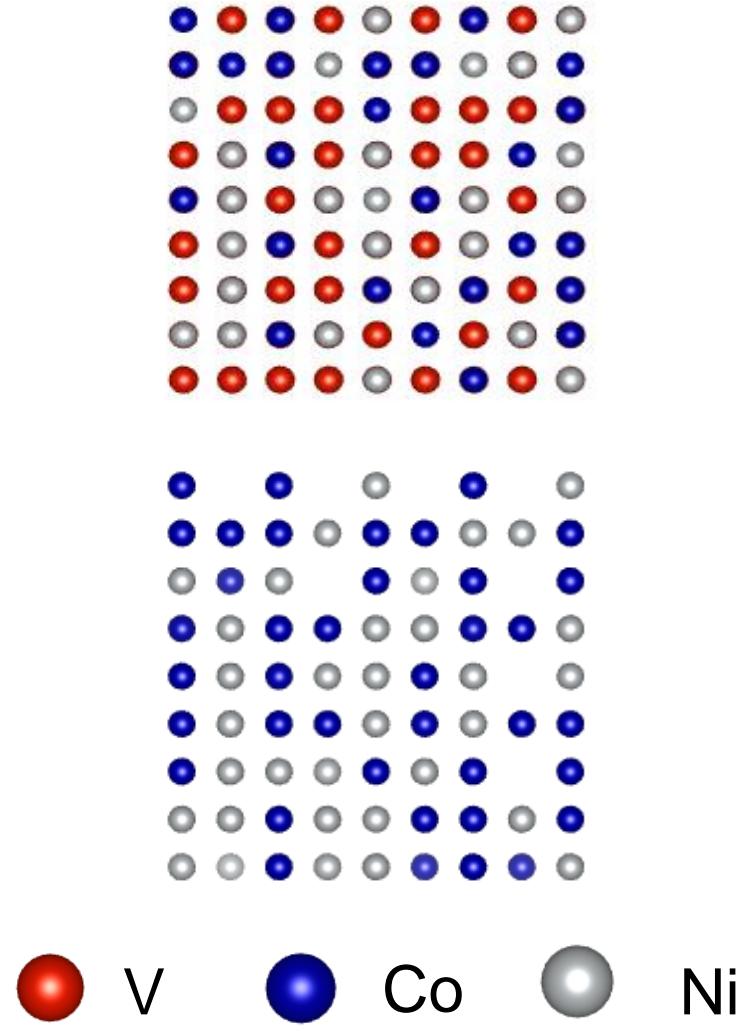
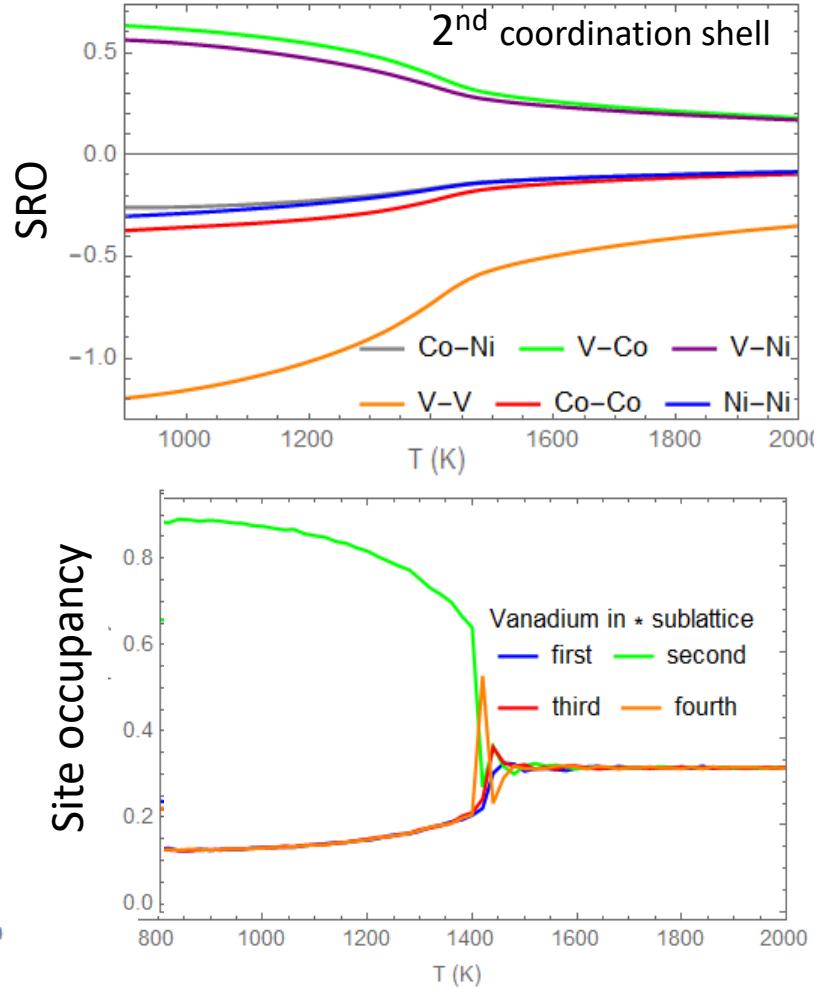
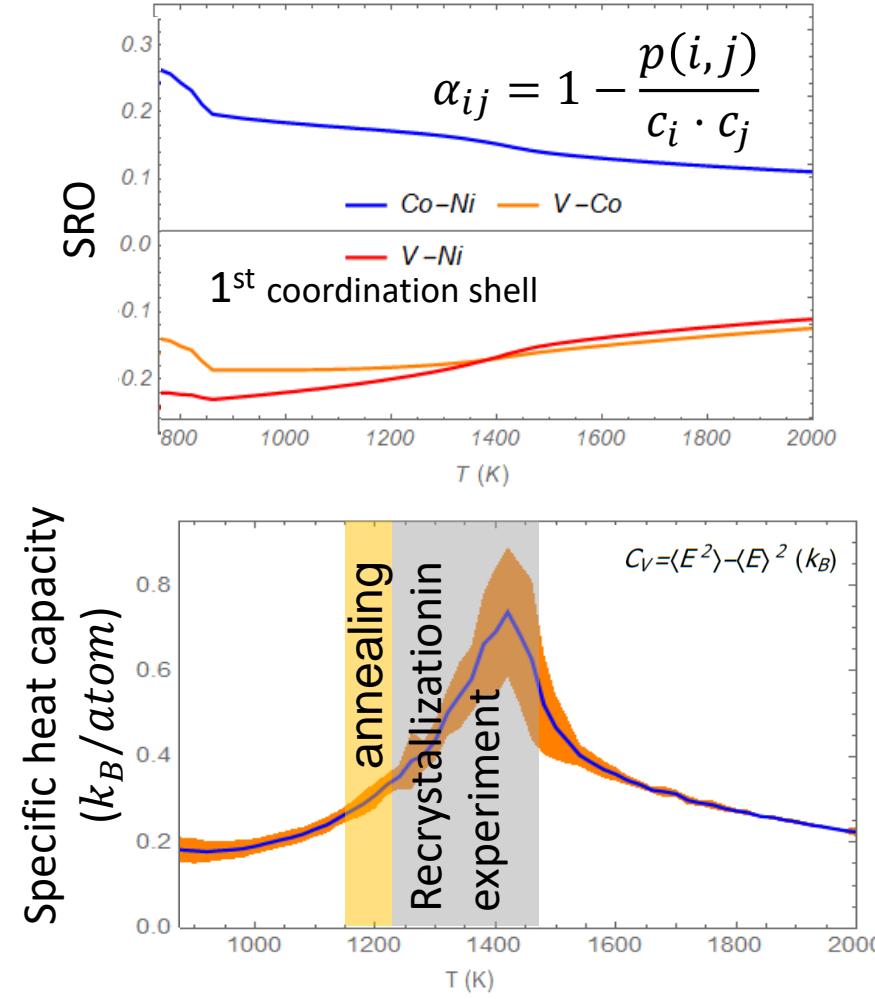
$$\alpha_{ij} = 1 - \frac{p(i,j)}{c_i \cdot c_j}$$

$i, j$  – chemical elements

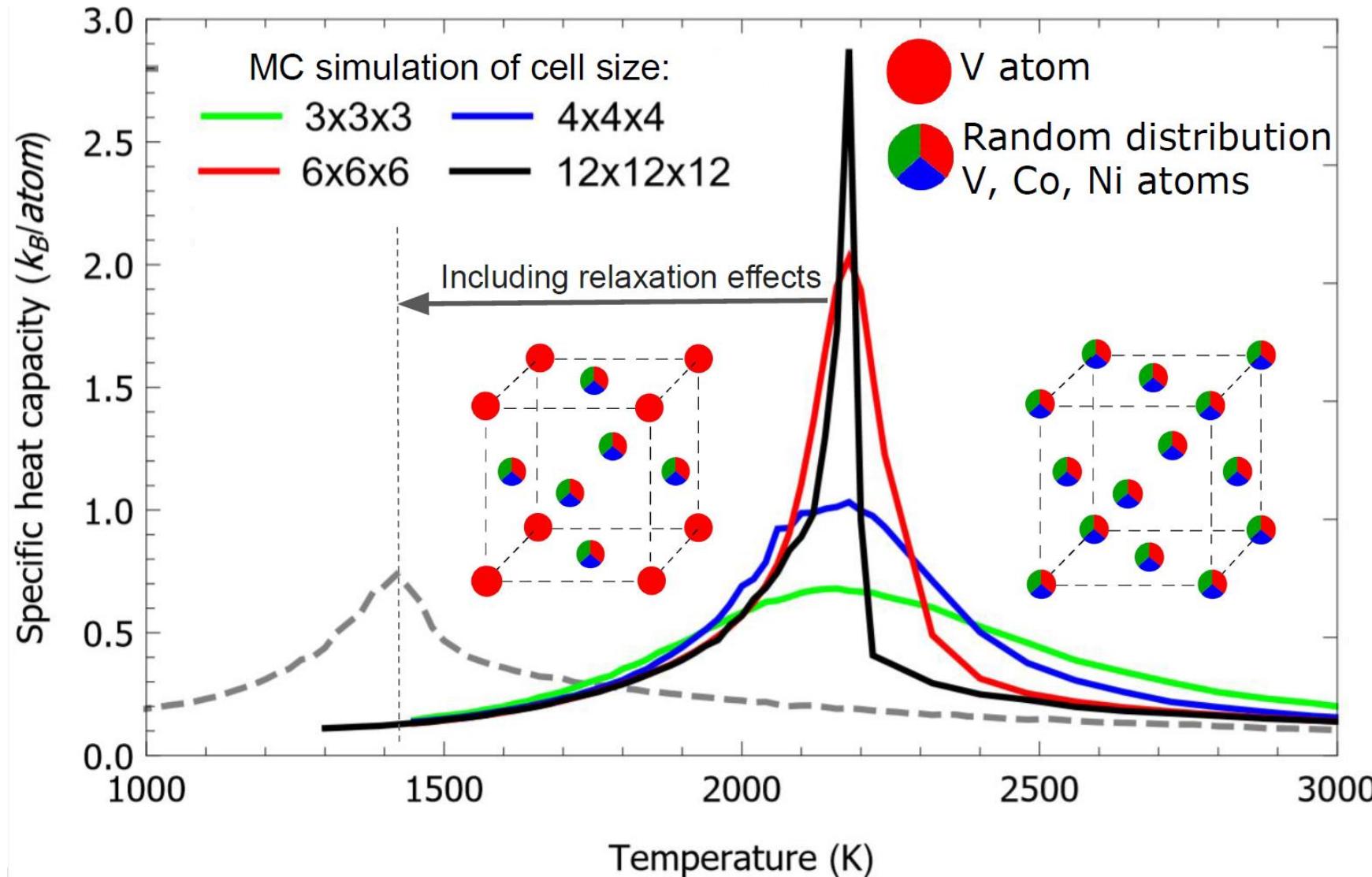
$c_j, c_i$  – concentrations

$p(i,j)$  – probability to find as  
neighbors

# Low-rank interatomic potential



# Low-rank interatomic potential

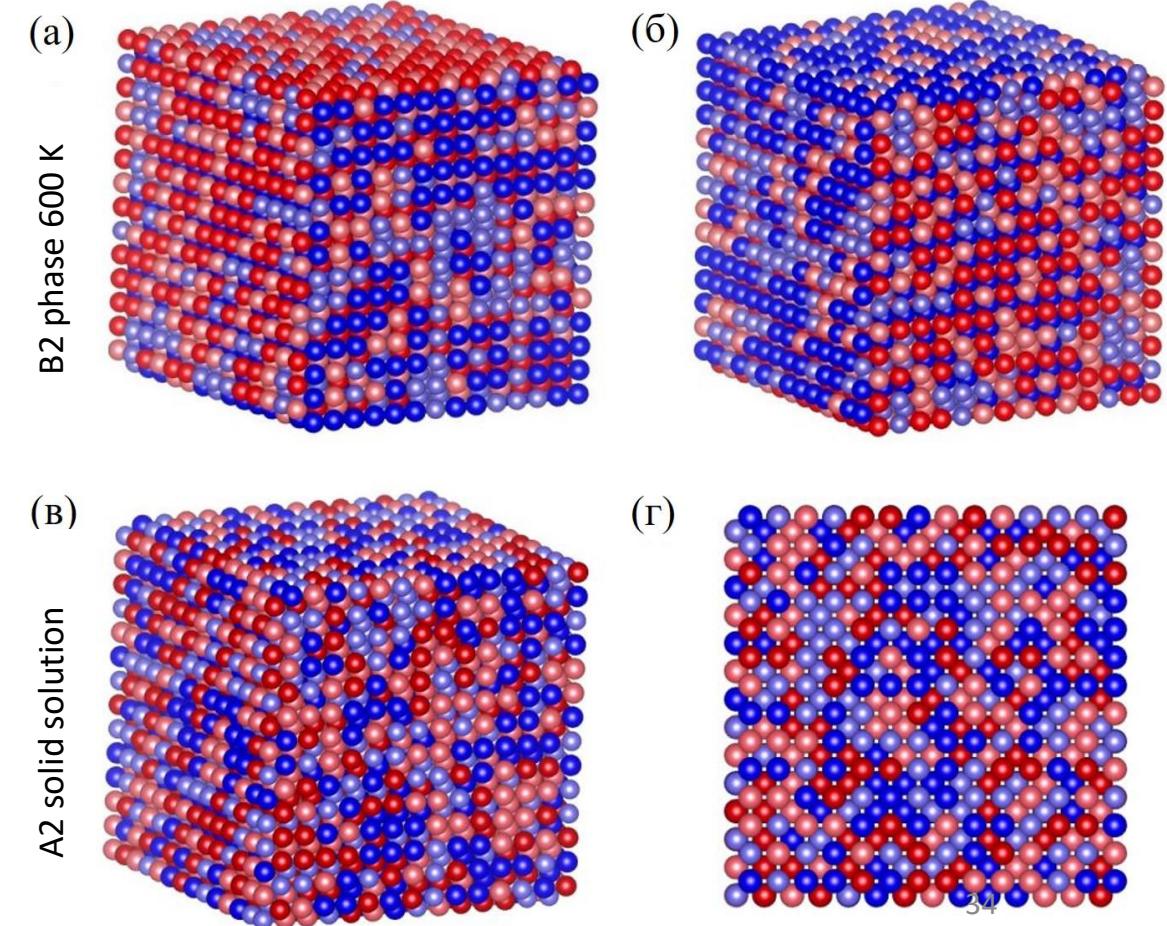
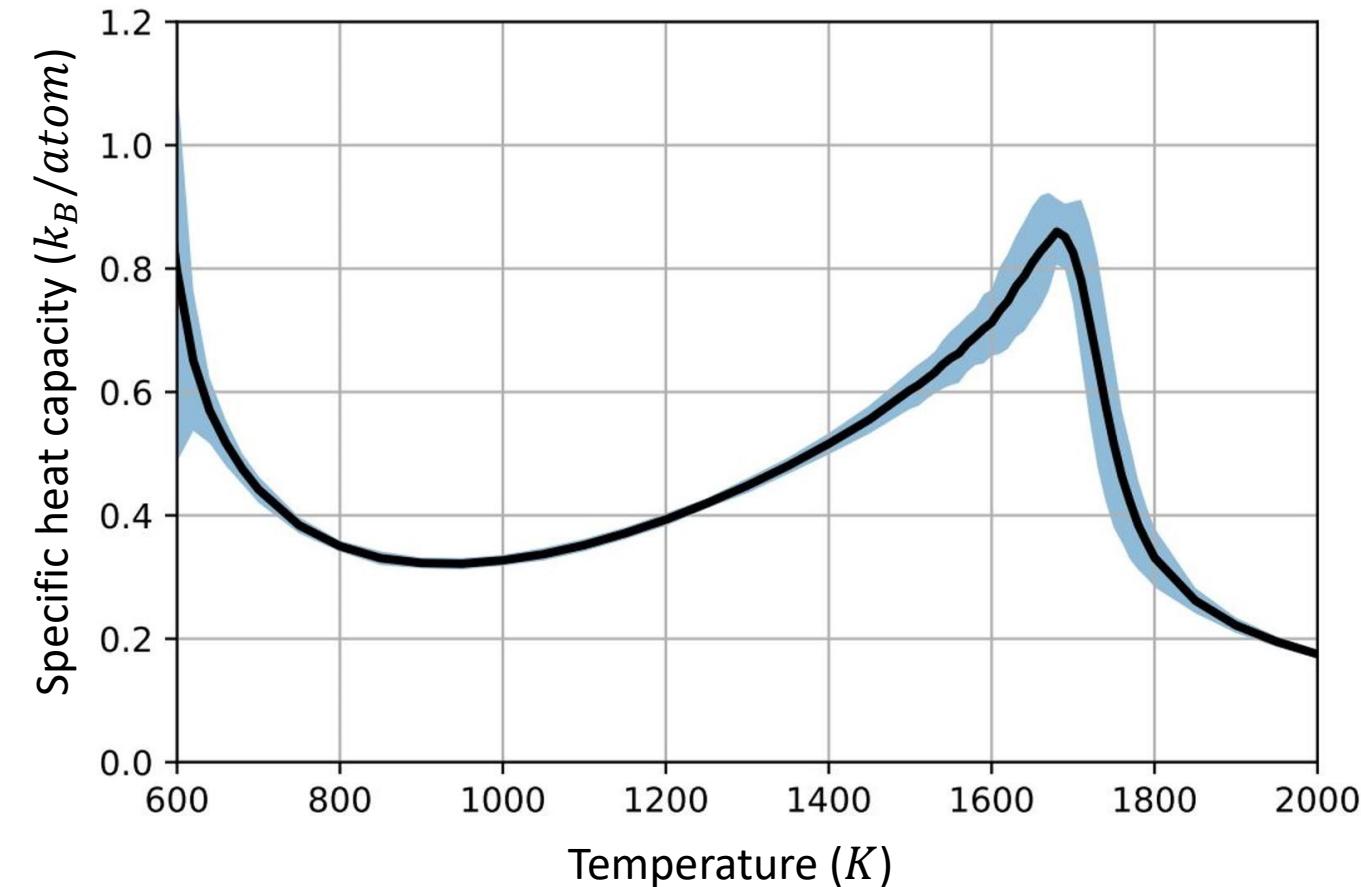


$$C_v = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}$$

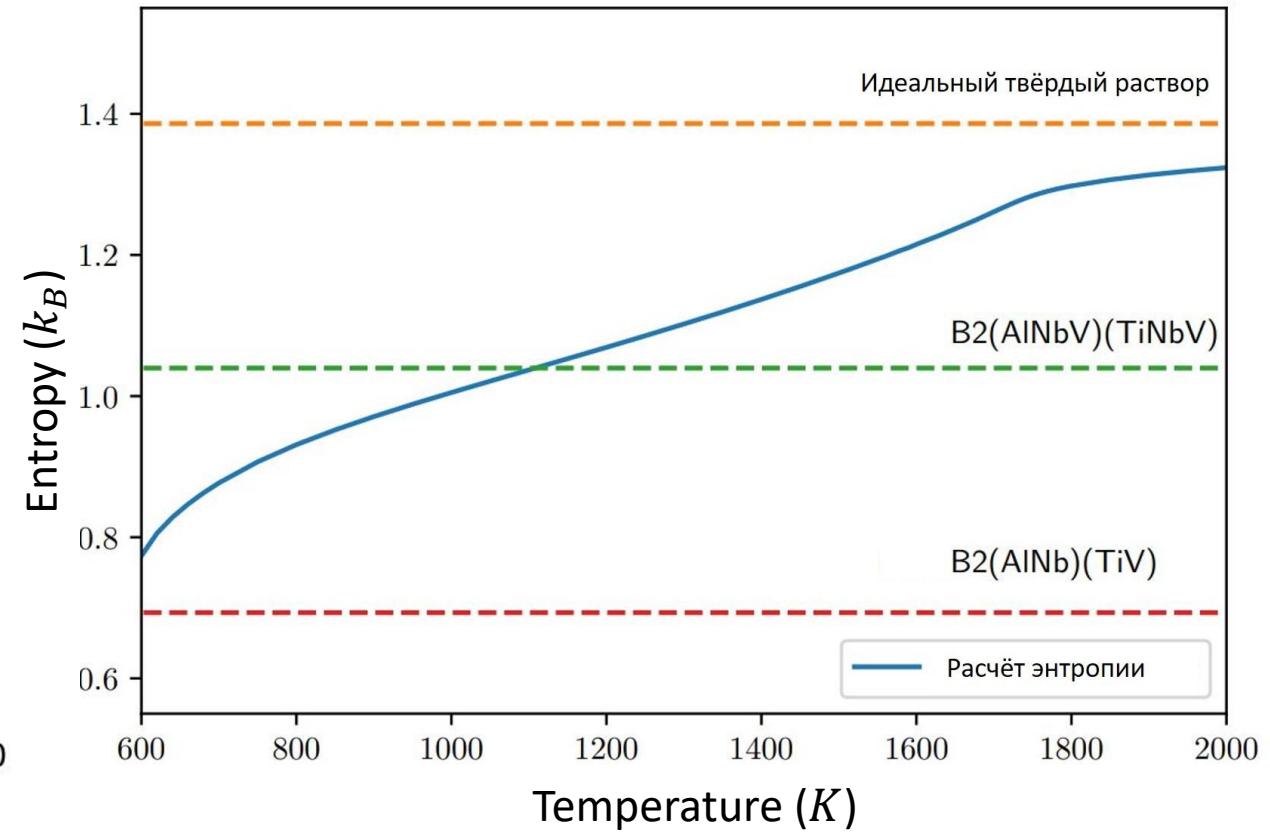
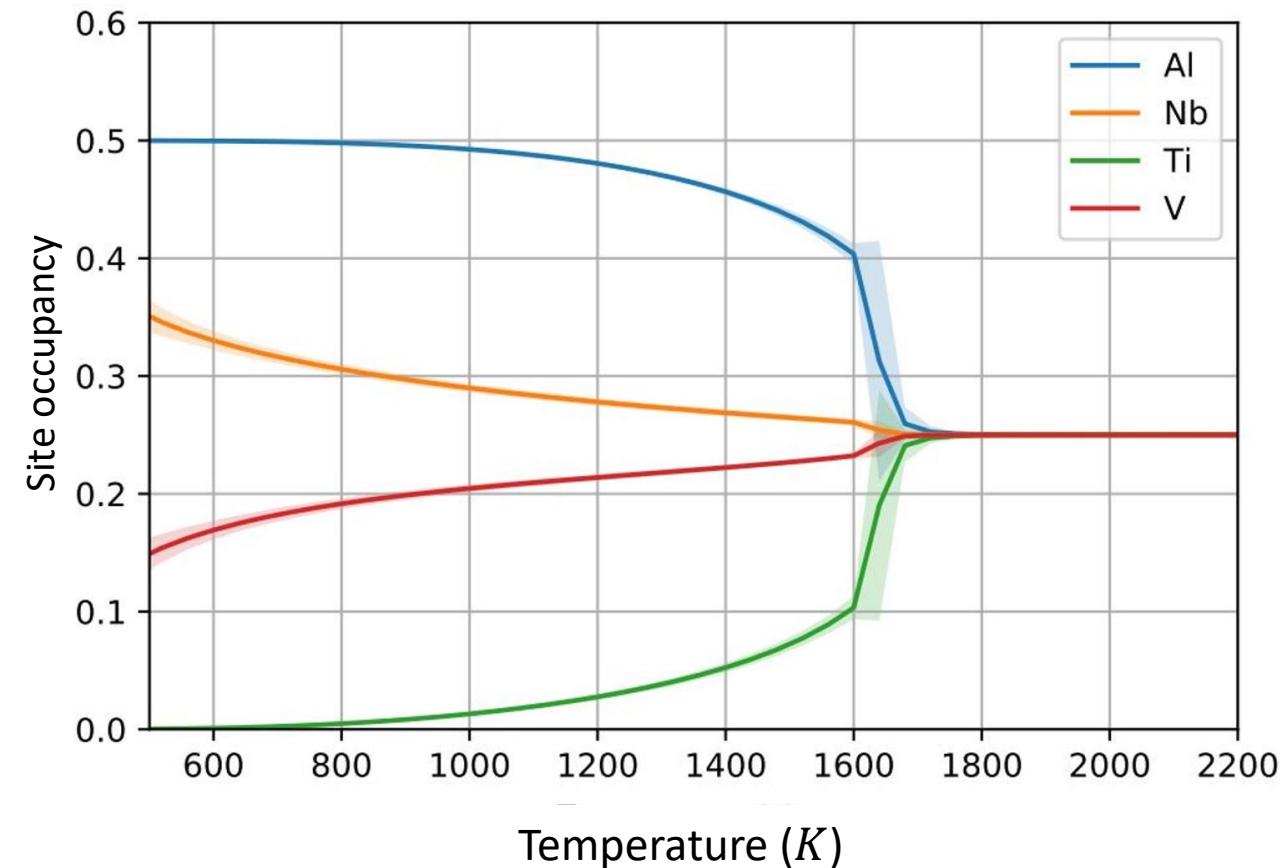
# Low-rank interatomic potential

## AlNbTiV and B2 phase formation

B2(AlNb)(TiV)? ← B2(AlNbV)(TiNbV) ← A2



# Low-rank interatomic potential



## 5. About Cython and LRP-code applicat

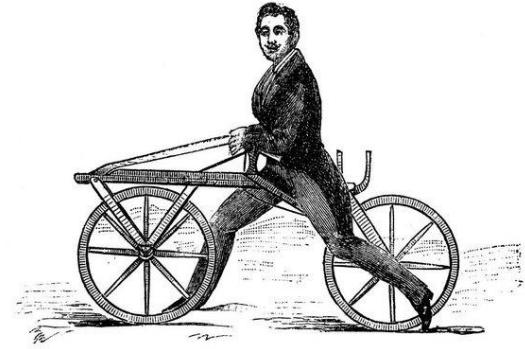


# About Cython and LRP-code application

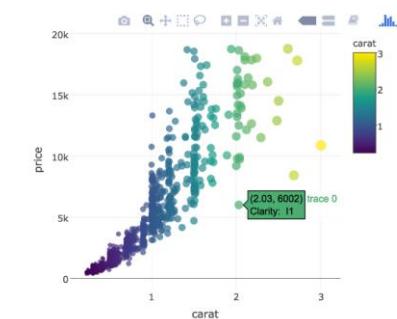
**Cython** is an **optimising static compiler** for both the **Python** programming language and the extended Cython programming language (based on **Pyrex**). It makes writing C extensions for Python as easy as Python itself.

**Cython gives you the combined power of Python and C to let you**

- write Python code that calls back and forth from and to C or C++ code natively at any point.
- easily tune readable Python code into plain C performance by adding static type declarations, also in Python syntax.
- use combined source code level debugging to find bugs in your Python, Cython and C code.
- interact efficiently with large data sets, e.g. using multi-dimensional NumPy arrays.
- quickly build your applications within the large, mature and widely used CPython ecosystem.
- integrate natively with existing code and data from legacy, low-level or high-performance libraries and applications.



```
In [5]: library(plotly)
set.seed(100)
d <- diamonds[sample(nrow(diamonds), 1000), ]
plot_ly(d, type = "scatter", mode = "markers",
        x = carat, y = price,
        color = -carat, size = -carat,
        text = -paste("Clarity:", clarity))
```



```
19. Sep 09:31 boot
21. Sep 15:50 dev
19. Sep 09:32 etc
21. Sep 15:52 home
7 30. Sep 2015 lib -> usr/lib
7 30. Sep 2015 lib64 -> usr/lib
34 23. Jul 10:01 lost+found
96 1. Aug 22:45 mnt
996 30. Sep 2015 opt
16 21. Sep 15:52 private -> /home/encrypted
4096 12. Aug 08:15 proc
560 21. Sep 15:37 root
7 30. Sep 15:50 run
4096 30. Sep 2015 sbin -> usr/bin
6 21. Sep 2015 srv
300 21. Sep 15:51 sys
4096 12. Aug 15:45 var
4096 23. Jul 10:25 var
```