GPUMD: A highly efficient molecular dynamics code fully implemented on graphics processing units

Zheyong Fan (樊哲勇)
Aalto University

Mini CV

- 2005-2010: PhD study at Nanjing University (Zhongzhou Ren) – Nuclear Physics, etc.
- 2010-2012: Postdoc research at Xiamen University (Jin-Cheng Zheng) – Thermoelectric transport
- 2012-now: Postdoc research at Aalto University (Ari Harju) – Heat and charge transport

QMP = quantum many-body
http://physics.aalto.fi/en/groups/qmp/

What is GPUMD?

- **GPUMD** = **G**raphics **P**rocessing **U**nits **M**olecular **D**ynamics
- Implemented fully on **G**raphics **P**rocessing **U**nits (GPUs)
- Many-body potentials: Tersoff, Stilinger-Weber, EAM, …
- Highly efficient (you will see)
- Green-Kubo method for thermal conductivity calculations (LAMMPS is wrong for many-body potentials)
- NEMD method for thermal conductivity and thermal conductance calculations, with spectral decomposition
- More…

Why develop GPUMD, when LAMMPS is available?

CMOS 2017

- LAMMPS is not always correct
- LAMMPS is not always fast
- LAMMPS is not always convenient to use
- GPUs become more and more powerful
- LAMMPS cannot explore the full power of GPUs
- **Why not** create a new one that can get the most from the GPUs?

GPU Computing:
A cheap way to achieve high performance

- Price: 3000 RMB to 30000 RMB
- Speed: 3 Tflops DP (K80: 4992 cores)
- Memory: 4 GB to 24 GB
- Programming: CUDA C/C++

Why GPU is faster?

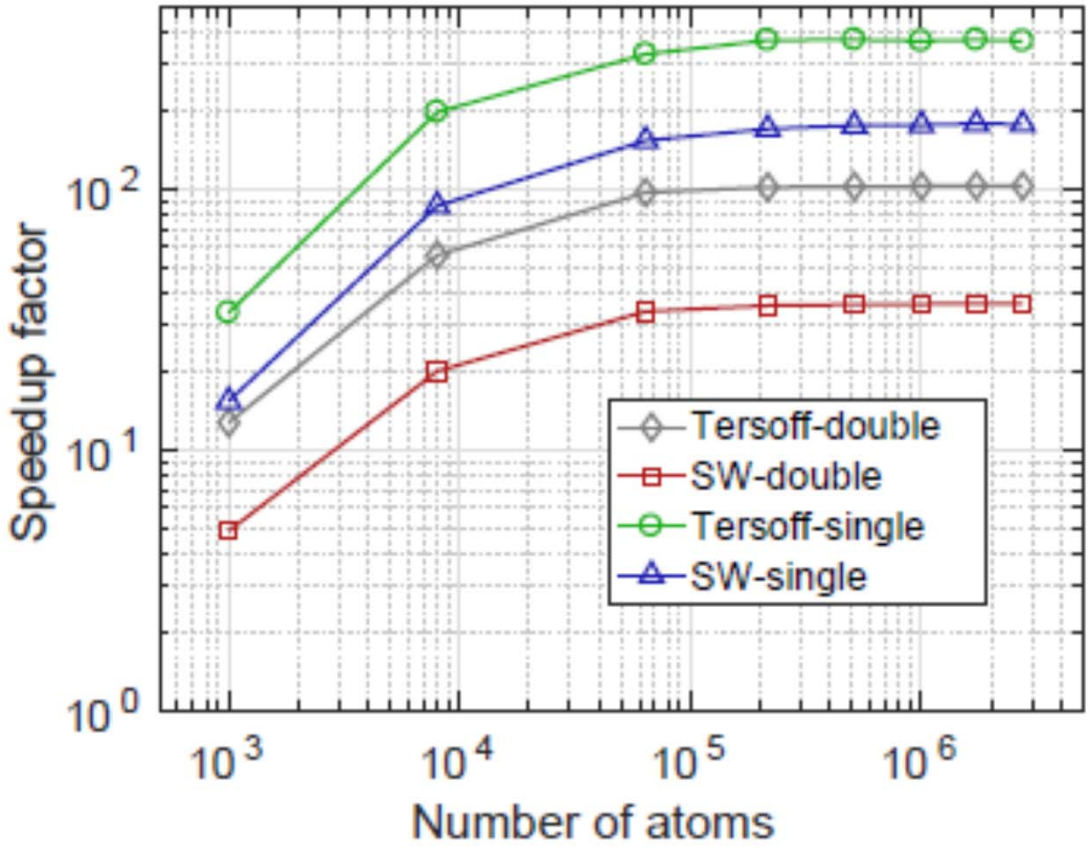
- CPU:

one or a few very fast computational threads

- GPU:

tens of thousand (not so fast) parallel threads

How fast GPUMD is compared to a single CPU core?

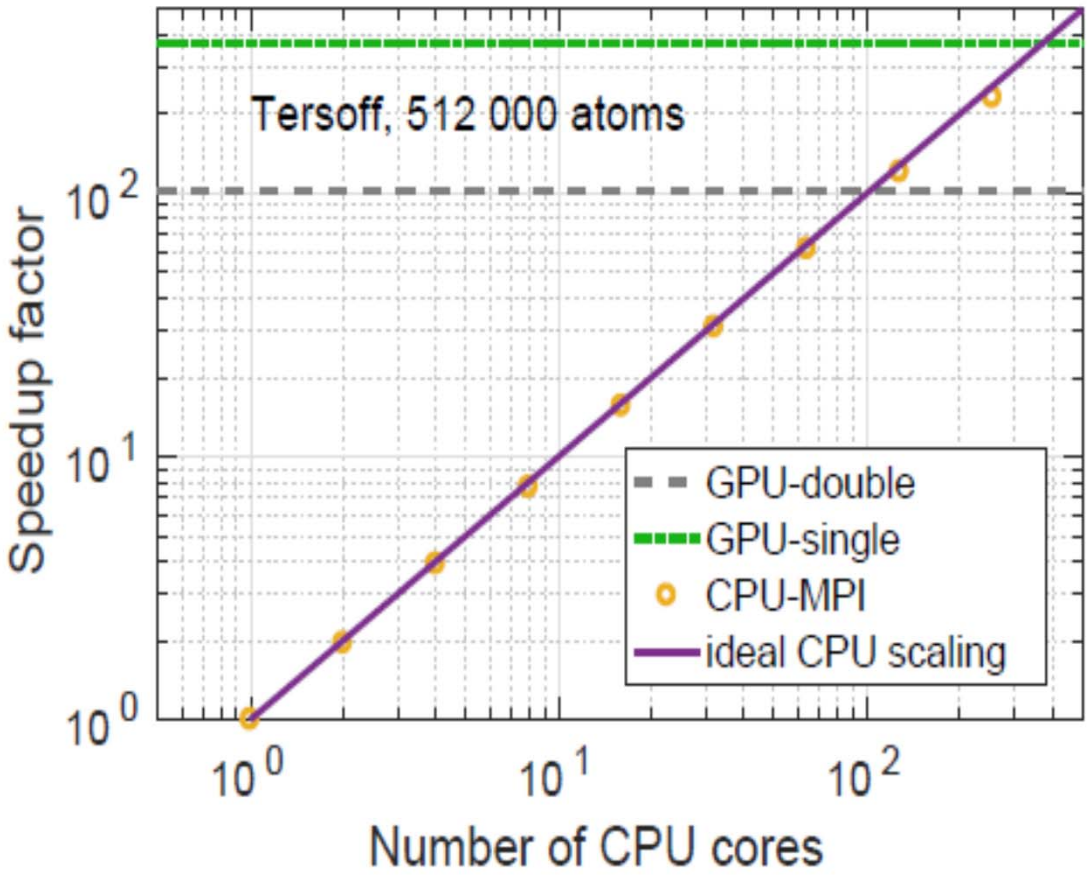


- CPU: Intel Xeon X5670@2.93GHz
- GPU: K40 (2880 CUDA cores)
- Model: Si@300K

Zheyong Fan–Aalto University: zheyongfan@163.com

How fast GPUMD is compared to many CPU cores?

CMOS 2017

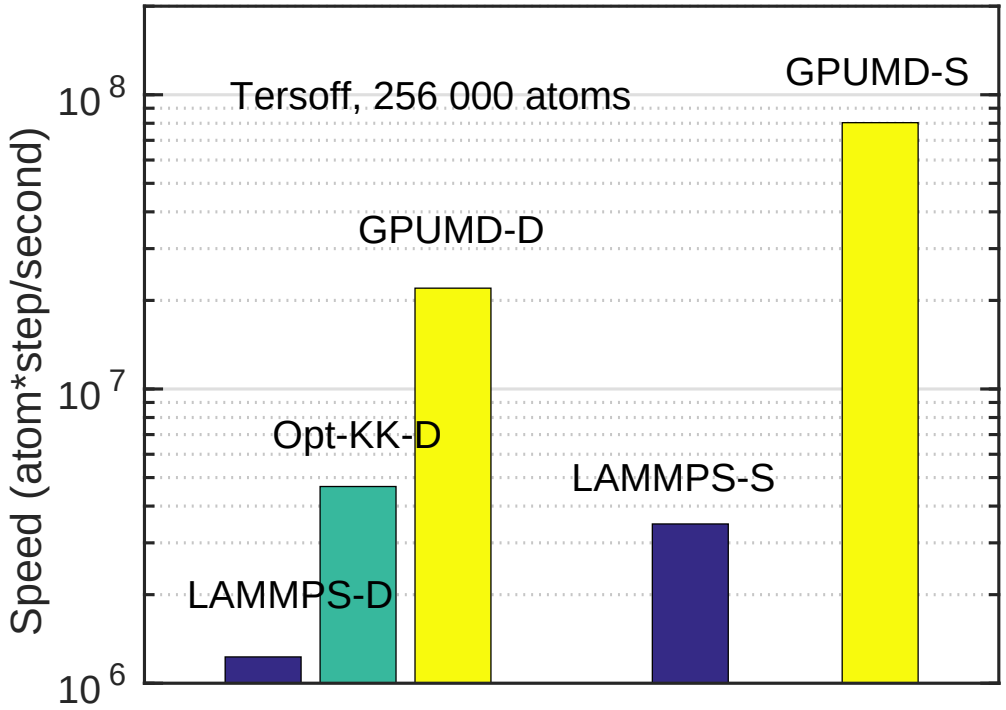


- CPU: Intel Xeon X5670@2.93GHz
- GPU: K40 (2880 CUDA cores)
- Model: Si@300K

How fast GPUMD is compared to the GPU versions of LAMMPS?

CMOS 2017

GPU = Tesla K40 Model = Si@300K



For details, see

Computer Physics Communications

Volume 218, September 2017, Pages 10-16

Why GPUMD is faster?

- It is fully implemented on GPUs
- It uses a set of elegant formulas for any many-body potential:
- Pair (!) force

$$\boldsymbol{F}_{ij} = -\boldsymbol{F}_{ji} = \frac{\partial U_i}{\partial \boldsymbol{r}_{ij}} - \frac{\partial U_j}{\partial \boldsymbol{r}_{ji}} = \frac{\partial \left(U_i + U_j\right)}{\partial \boldsymbol{r}_{ij}}$$

- Per-atom virial

$$\mathbf{W}_i = -\frac{1}{2}\sum_{j \neq i} \boldsymbol{r}_{ij} \otimes \boldsymbol{F}_{ij}$$

- Per-atom heat current

$$\boldsymbol{J}_i^{\mathrm{pot}} = \sum_{j \neq i} \boldsymbol{r}_{ij} \left(\frac{\partial U_j}{\partial \boldsymbol{r}_{ji}} \cdot \boldsymbol{v}_i \right)$$

- For details, see Phys. Rev. B **92**, 094301 (2015).

Doubt about the pairwise force expression?

- Finite-difference

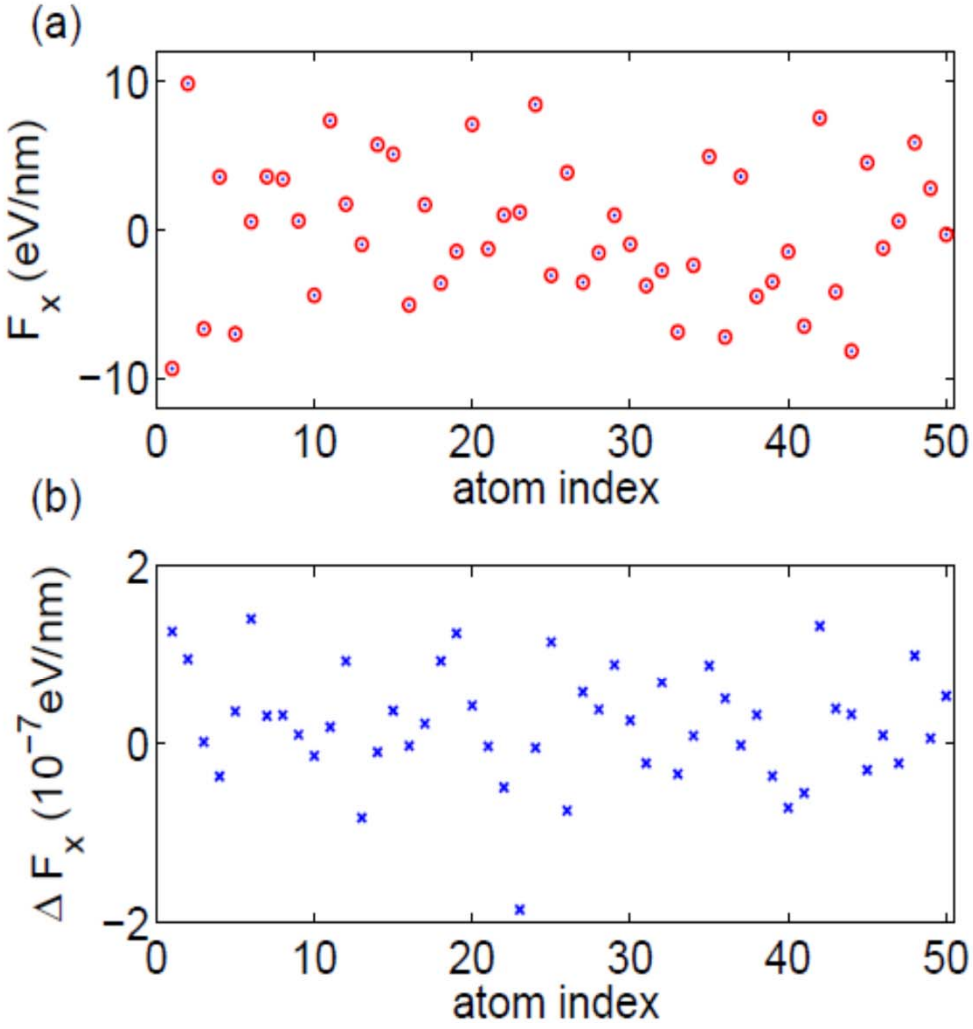
$$F_{ix} = \frac{U(\cdots, r_i - \Delta x e_{\boldsymbol{x}}, \cdots) - U(\cdots, r_i + \Delta x e_{\boldsymbol{x}}, \cdots)}{2\Delta x}$$

- Pairwise force

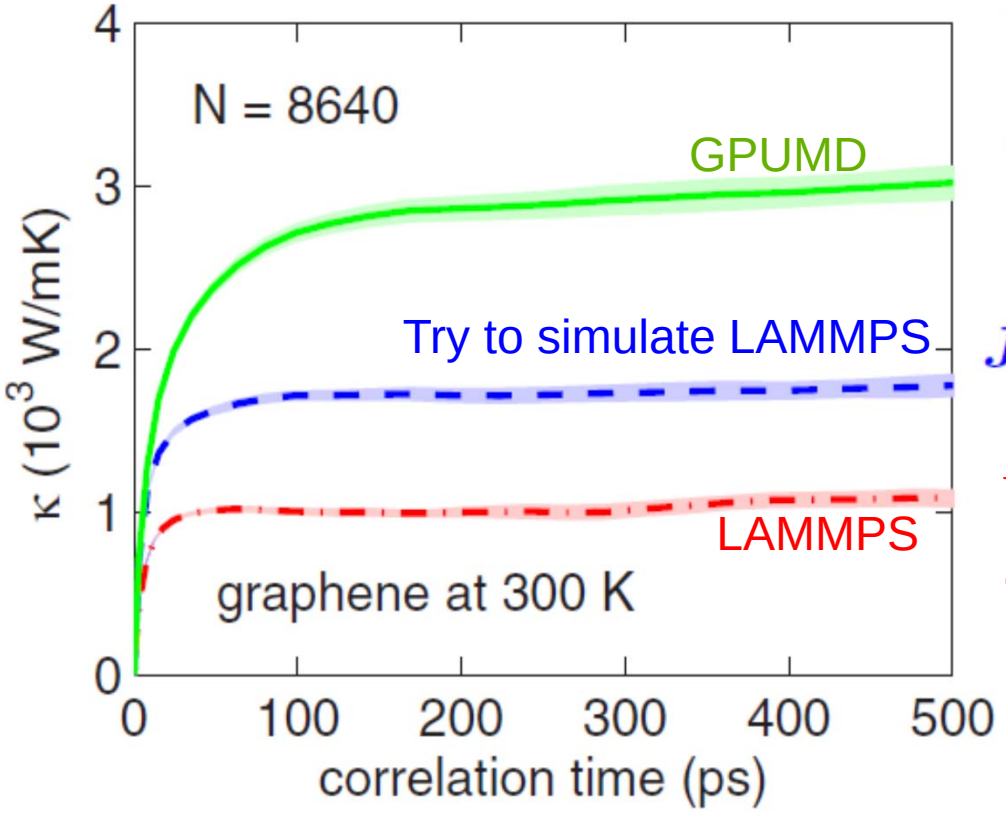
$$\boldsymbol{F}_{ij} = -\boldsymbol{F}_{ji} = \frac{\partial U_i}{\partial \boldsymbol{r}_{ij}} - \frac{\partial U_j}{\partial \boldsymbol{r}_{ji}} = \frac{\partial \left(U_i + U_j\right)}{\partial \boldsymbol{r}_{ij}}$$

- For details, see

arXiv:1503.06565



What's wrong with LAMMPS?



$$\boldsymbol{J}_i^{\mathrm{pot}} = \sum_{j \neq i} \boldsymbol{r}_{ij} \left(\frac{\partial U_j}{\partial \boldsymbol{r}_{ji}} \cdot \boldsymbol{v}_i \right)$$

$$\boldsymbol{J}^{\mathrm{pot}} = -\frac{1}{2} \sum_i \sum_{j \neq i} (\boldsymbol{r}_{ij} \otimes \boldsymbol{F}_{ij}) \cdot \boldsymbol{v}_i$$

Who knows the exact expression?

For details, see
Phys. Rev. B **92**, 094301 (2015).

Zheyong Fan–Aalto University: *zheyongfan@163.com*

The Green-Kubo method in GPUMD

Running thermal conductivity with the in-out decomposition

$$\kappa_{xx}^{\rm in}(t) = \frac{1}{k_B T^2 V}\int_0^t dt' C_{xx}^{\rm in}(t');$$

$$\kappa_{xx}^{\rm out}(t) = \frac{1}{k_B T^2 V}\int_0^t dt' C_{xx}^{\rm out}(t');$$

$$\kappa_{xx}^{\rm cross}(t) = \frac{1}{k_B T^2 V}\int_0^t dt' C_{xx}^{\rm cross}(t').$$

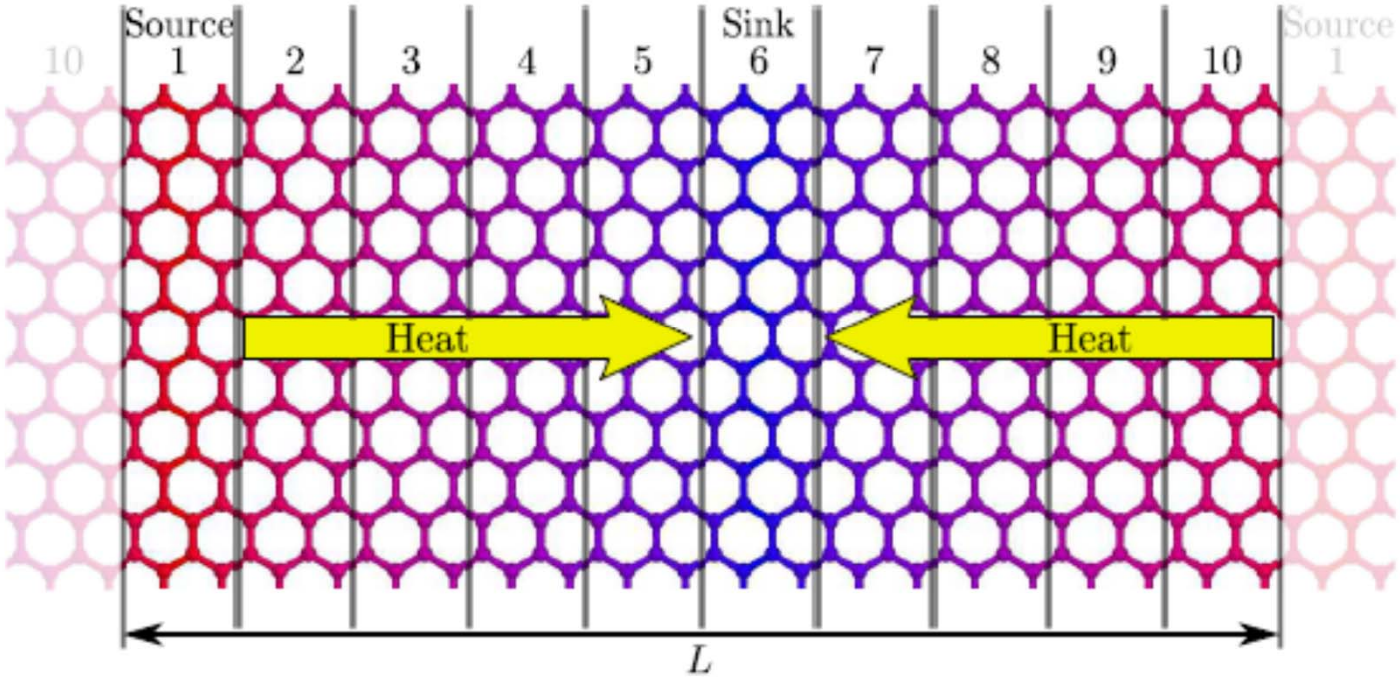
GPUMD input script:

```
potential   potentials/c_tersoff_fan_2017.txt
ensemble    npt_ber 300 300 0.01 0 0 0 0.0005
time_step   1
dump_thermo 1000
run         1000000

ensemble    nve
compute_hac 20 50000 10
run         10000000
```

For details, see Phys. Rev. B **95**, 144309 (2017).

The NEMD method in GPUMD



Fourier's law

$$\kappa = \frac{Q}{|\nabla T|}.$$

GPUMD input script:

```
# Initialization and equilibration here...

ensemble     heat_nhc 300 100 10 1 6
compute_temp 1000
run          2000000
```


Zheyong Fan–Aalto University: zheyongfan@163.com

The spectral-decomposition method in GPUMD

$$K^{\rm out}_{A\to B}(t)=\sum_{i\in A}\sum_{j\in B}\left\langle \frac{\partial U_i}{\partial z_{ij}}(0)v_{zj}(t)-\frac{\partial U_j}{\partial z_{ji}}(0)v_{zi}(t)\right\rangle$$

$$\tilde{K}^{\rm in/out}_{A\to B}(\omega)=\int_{-\infty}^{\infty}dt e^{i\omega t}K^{\rm in/out}_{A\to B}(t).$$

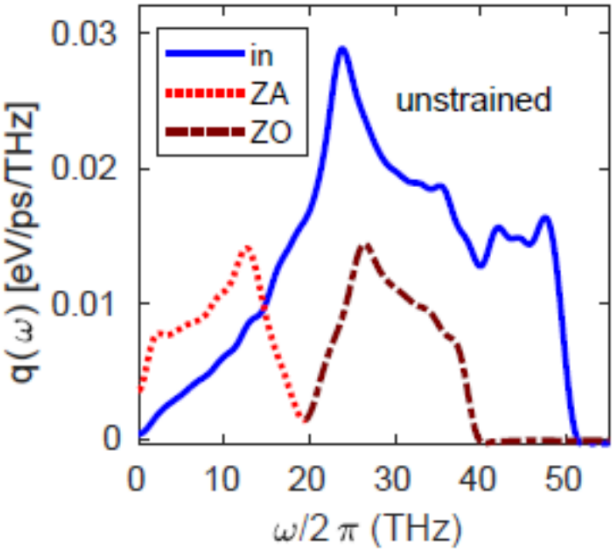
$$Q^{\rm in/out}_{A\to B}=\int_0^{\infty}\frac{d\omega}{2\pi}\left[2\tilde{K}^{\rm in/out}_{A\to B}(\omega)\right]\equiv\int_0^{\infty}\frac{d\omega}{2\pi}q^{\rm in/out}_{A\to B}(\omega)$$

$$\kappa^{\rm in/out}_{A\to B}(\omega)=\frac{q^{\rm in/out}_{A\to B}(\omega)}{S|\nabla T|}$$

$$g^{\rm in/out}_{A\to B}(\omega)=\frac{q^{\rm in/out}_{A\to B}(\omega)}{S|\Delta T|}$$

GPUMD input script:

```
#some commands
compute_shc 2 250 100000 4 5
# some commands
```



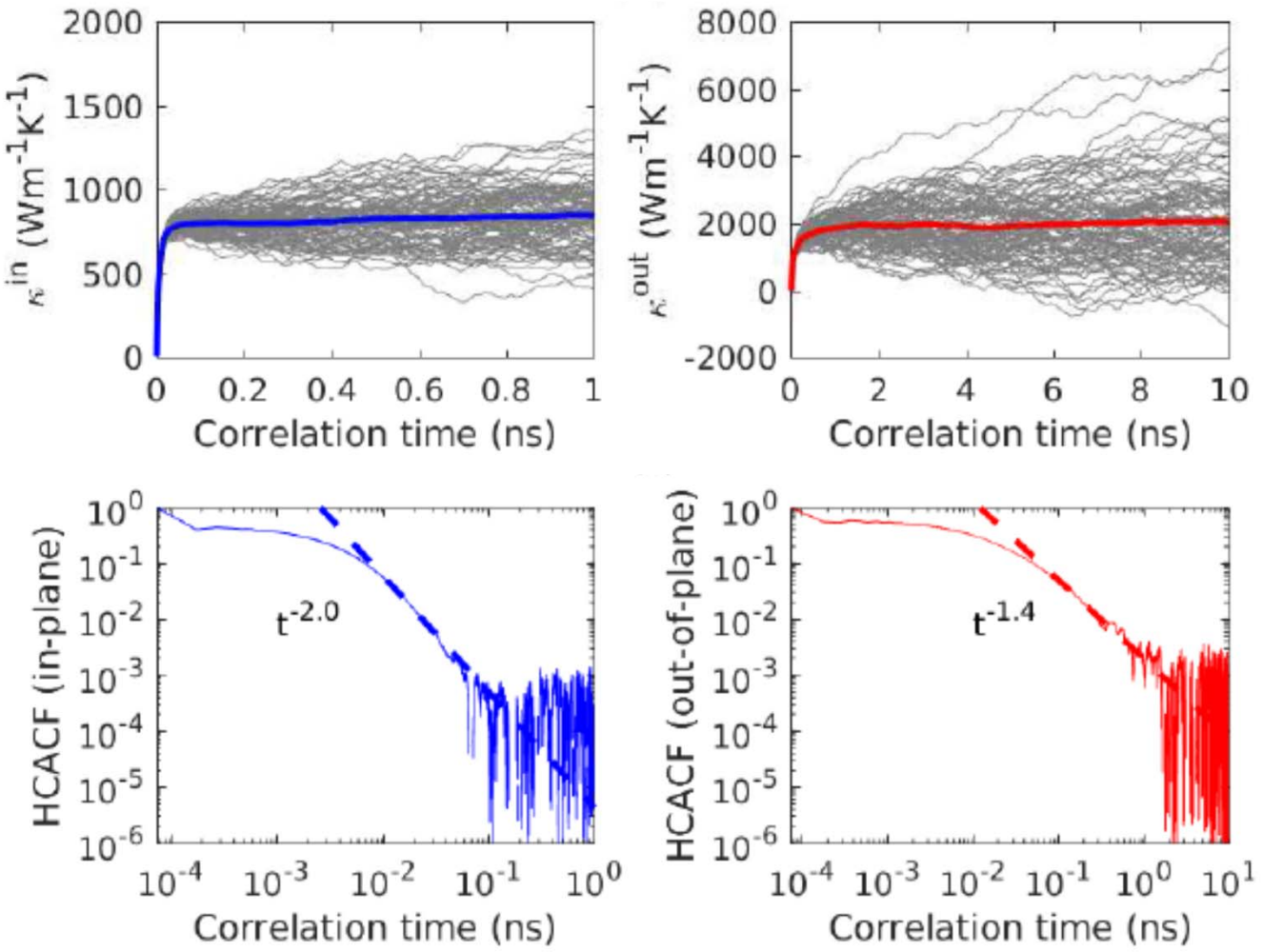
For details, see Phys. Rev. B **95**, 144309 (2017).

Application (1) – Thermal conductivity components of pristine graphene

CMOS 2017

in-plain phonons

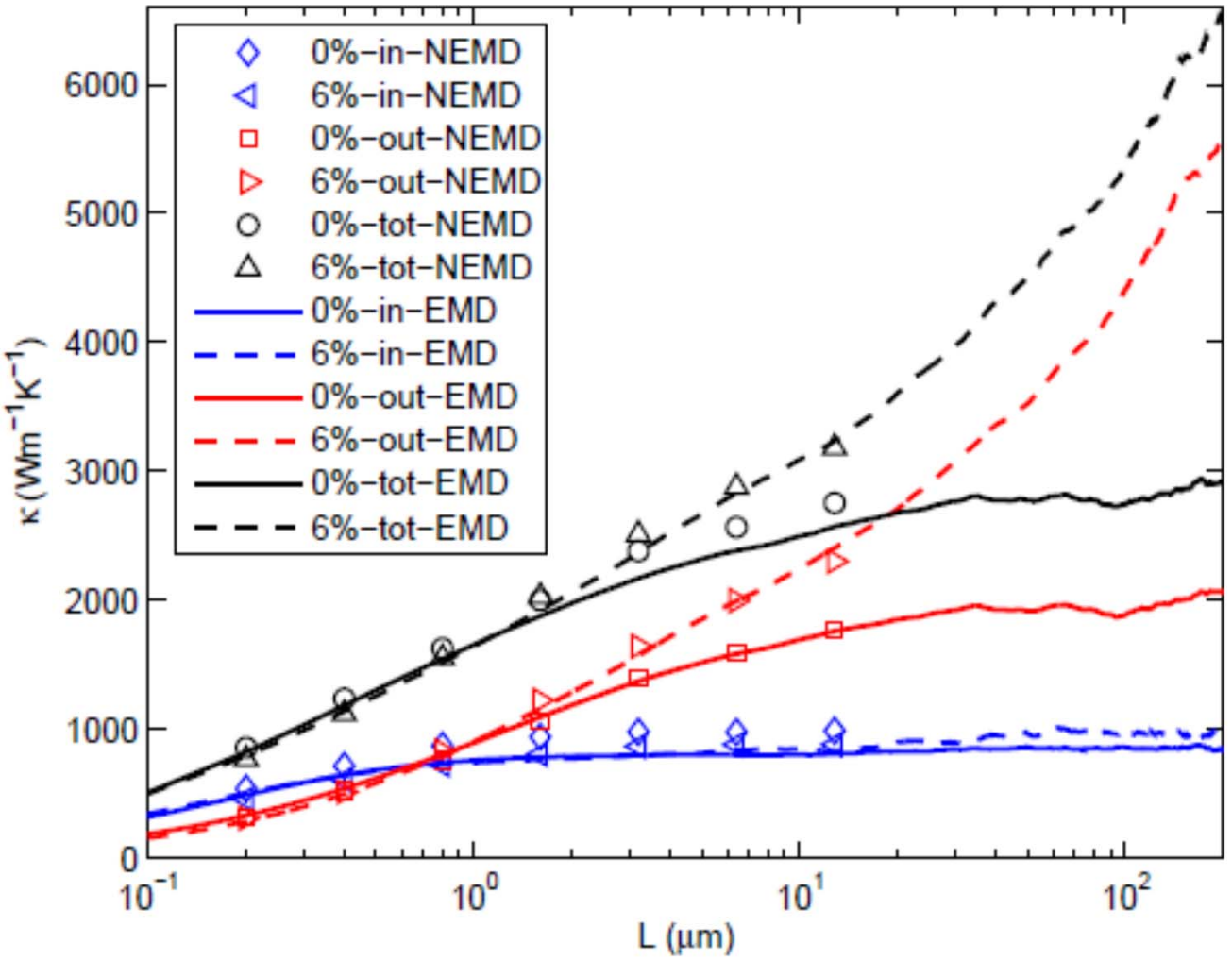
out-of-plain phonons



Zheyong Fan–Aalto University: *zheyongfan@163.com*

Application (2) – Thermal conductivity
"divergence" of graphene under tensile strain

CMOS 2017



"divergent"

convergent

Zheyong Fan–Aalto University: zheyongfan@163.com

In summary,

- There is a fast MD code called GPUMD
- And I want to share it with you all
- https://github.com/brucefan1983/GPUMD

- I want to make it more powerful
- And your comments will be helpful to me

- The current version of GPUMD can already do many things
- And collaborations are welcome

Last words: