Libra *Summer School and Workshop 2024*

Alexey Akimov

University at Buffalo, SUNY

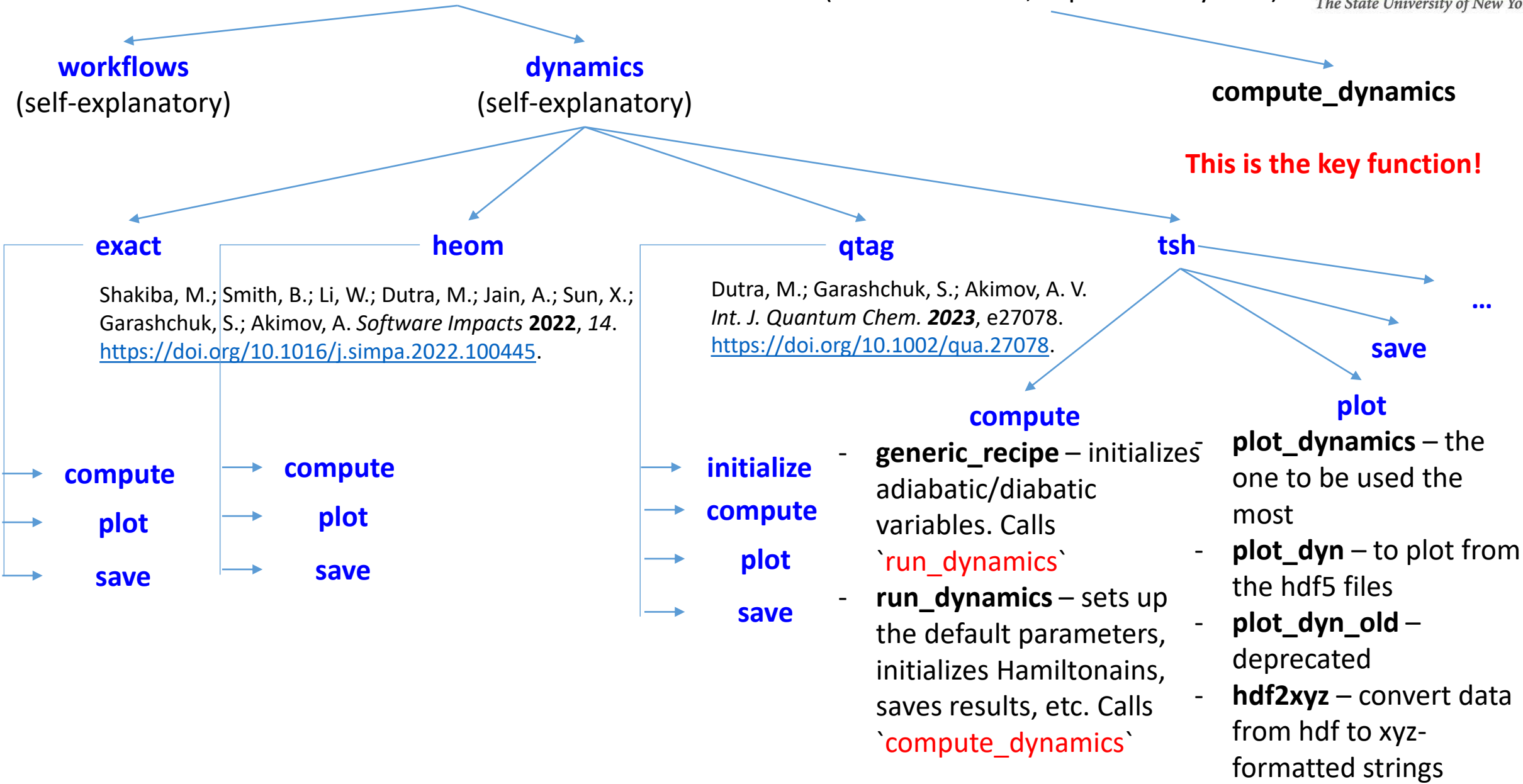July 11, 2024

Structure of Libra package

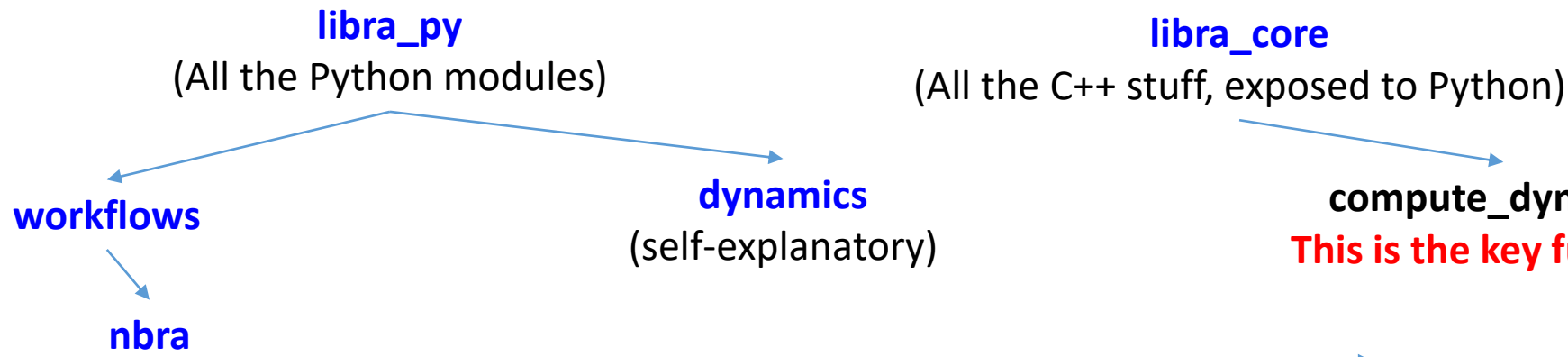*Atomistic Workflows*

libra_py
(All the Python modules)

libra_core
(All the C++ stuff, exposed to Python)

workflows

dynamics
(self-explanatory)

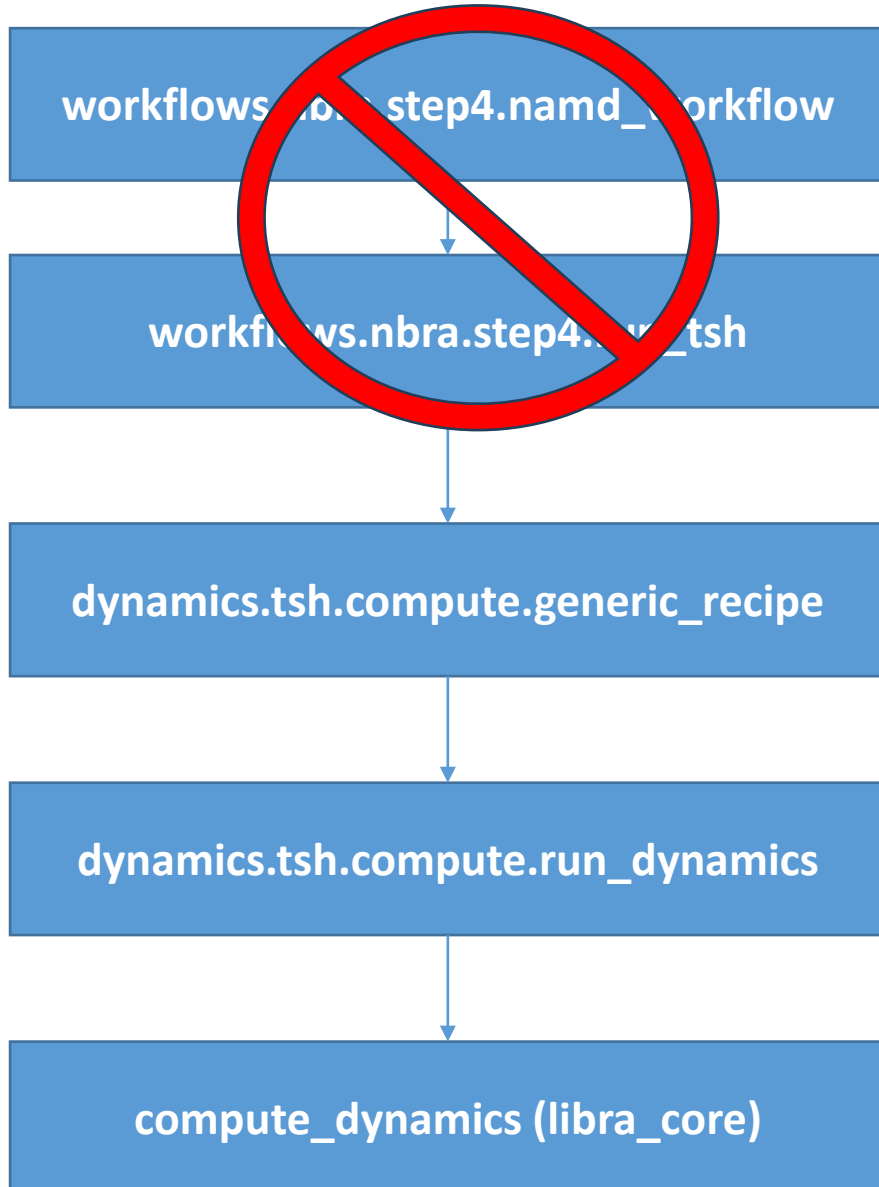**compute_dynamics**
This is the key function!

nbra
(neglect of back-reaction approximation)

step2: time-overlaps & NAC calculations

step2_many_body: SP time-overlaps & NAC calculations, and CI info

step2_dftb: time-overlaps & NAC calculations for DFTB+ specifically

step2_ergoscf: time-overlaps & NAC calculations for ErgoSCF specifically

step2_analysis: compute compositions of the MB states, spectra for QE, etc.

mapping: functions for mapping single-particle properties (e.g. couplings) to the many-body ones. Most of it is to be replaced by `**mapping2**`

mapping2: new, simplified and cleaned up version of the mapping module.

step3: phase corrections, re-indexing, SD-basis properties, etc.

step3_many_body: functions for computing NACs in the MB basis.

step4: to be deprecated. use `**compute_dynamics**` instead

ann: complete module for training ANN on the NBRA data and for forecasting it for longer times

decoherence_times: to compute dephasing times and influence spectra from the NBRA data

qsh: module for quasi-stochastic Hamiltonian construction

lz: module for Belyaev-Lebedev-Landau-Zener calculations

etc

University at Buffalo
The State University of New York

Brief Overview of the Deprecated step4

workflows.nbra.step4.namd_workflow — Parallelization, multiple methods/initial conditions

workflows.nbra.step4.run_tsh — Initialization of nuclear variables

dynamics.tsh.compute.generic_recipe — Initialization of electronic variables, transformation to the desired representation, nHamiltonian object construction and initialization

dynamics.tsh.compute.run_dynamics — Initialization of default parameters, writing files, computing some observables, thermostat variables, iteration over nuclear timesteps

compute_dynamics (libra_core) — TSH/Ehrenfest and decoherence algorithms, trajectory coupling, Hamiltonian properties updates (calling external Python functions)