

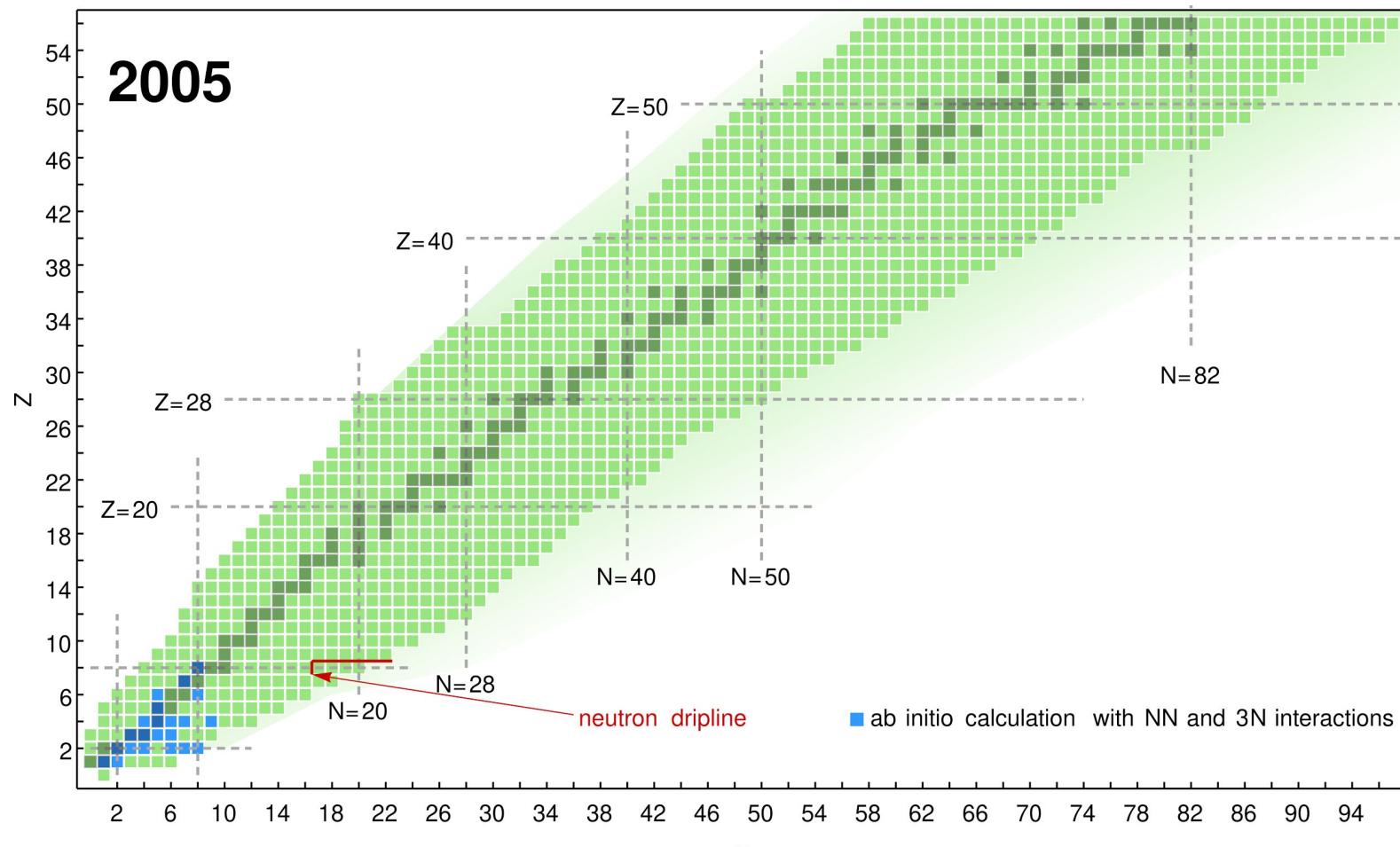
Addition and removal energies via the IM-SRG method

Fei Yuan

Committee:
Morten Hjorth-Jensen
Scott Bogner
Alexandra Gade
Carlo Piermarocchi
Scott Pratt

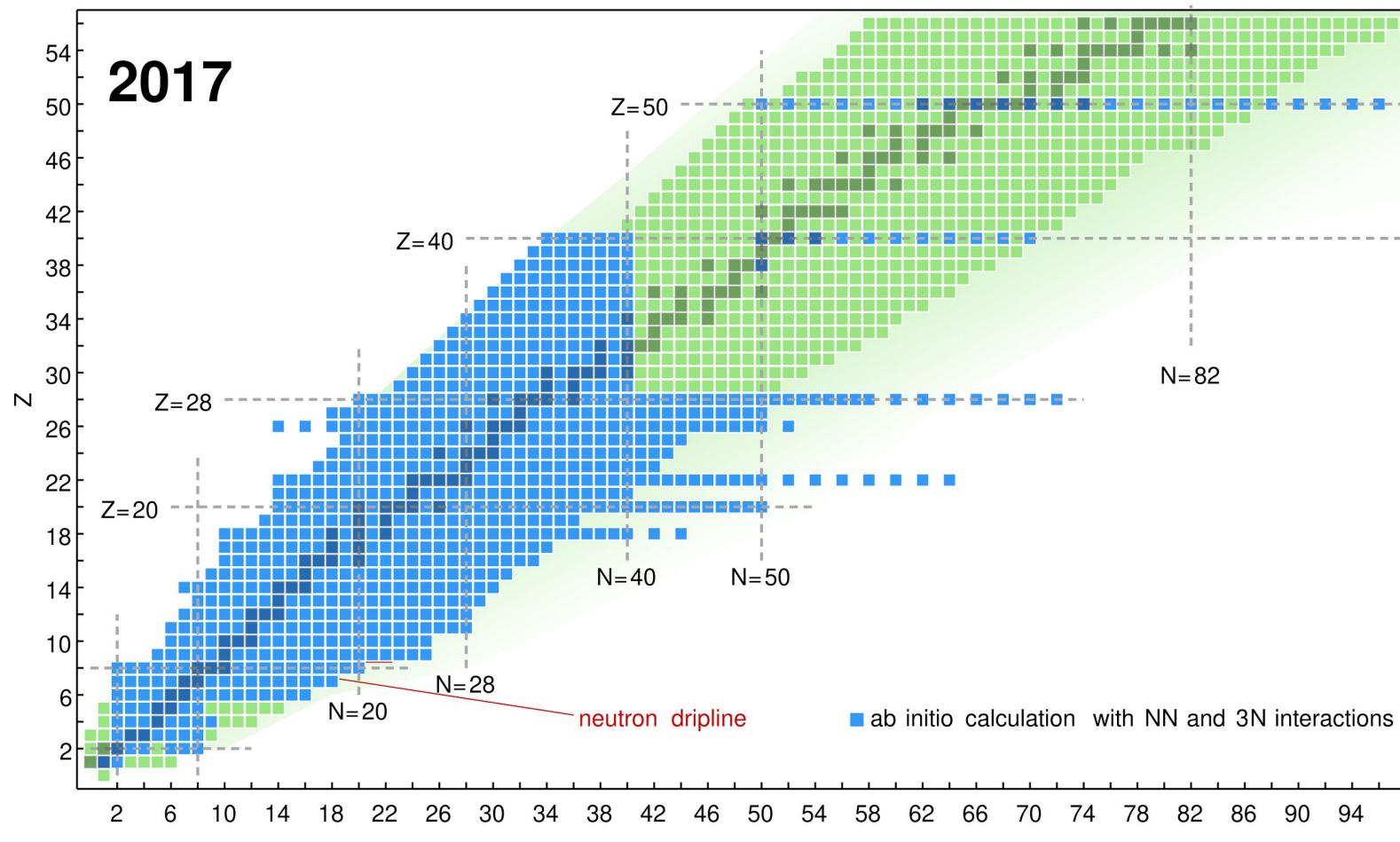
2018-01-24

Nuclear chart of *ab initio* theory (2005)



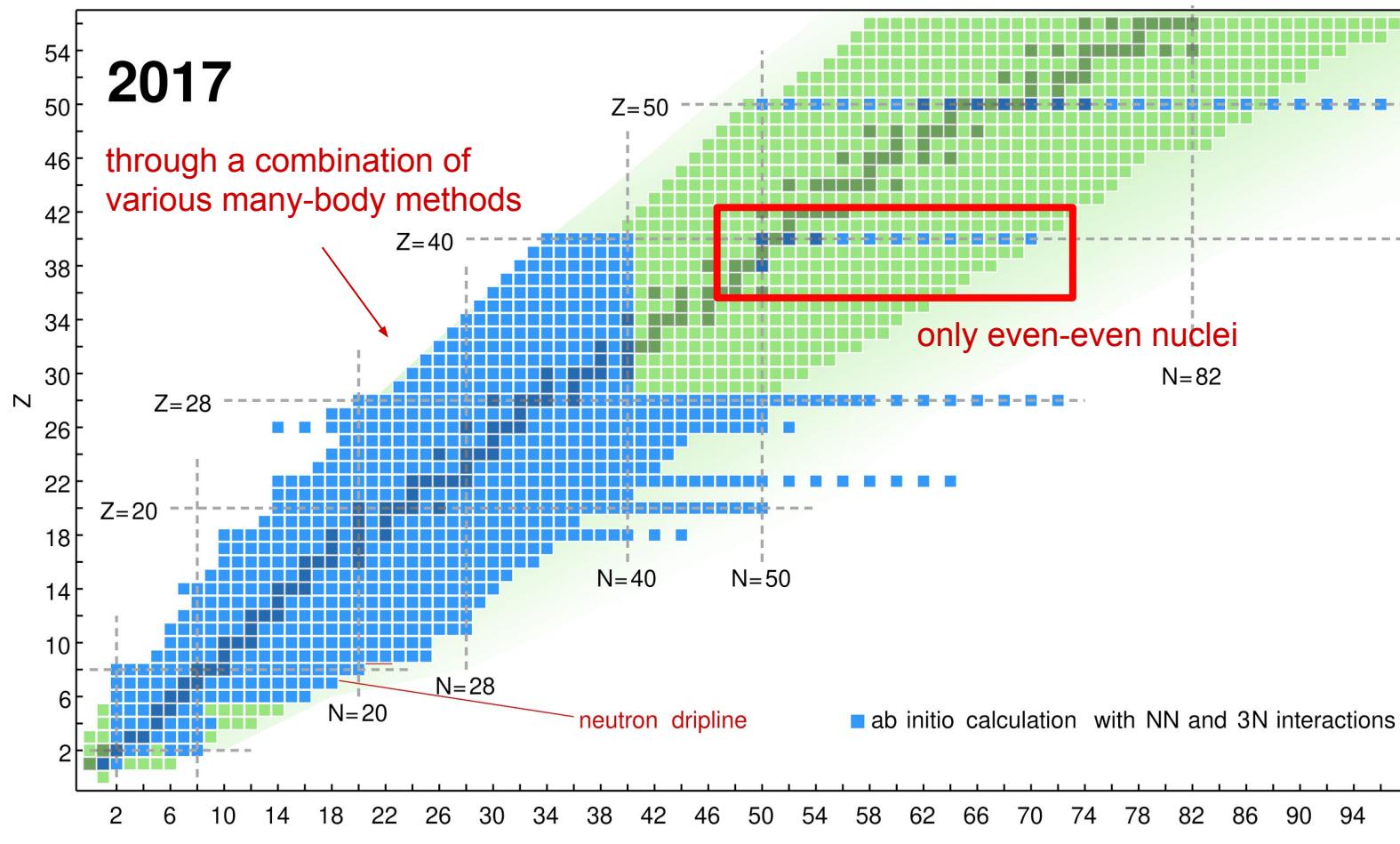
(image courtesy of Heiko Hergert)

Nuclear chart of *ab initio* theory (2017)



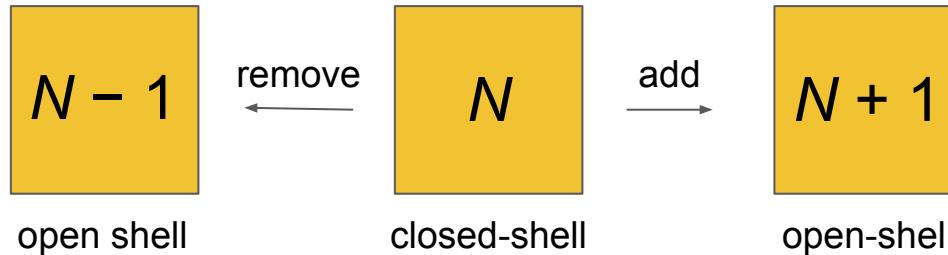
(image courtesy of Heiko Hergert)

Nuclear chart of *ab initio* theory (2017)



(image courtesy of Heiko Hergert)

Addition and removal energies



Closed-shell systems are easy to calculate

Goal: Access open-shell systems one particle away from a closed-shell system (might even be an excited state!)

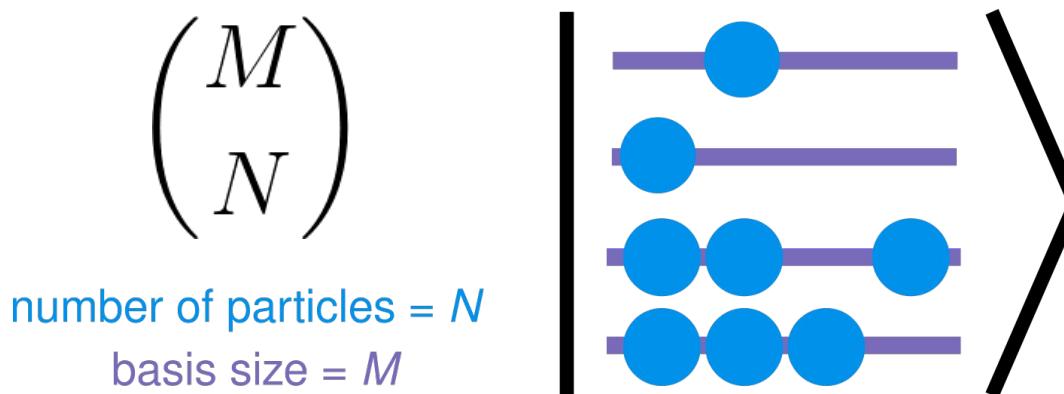
addition energy: $\varepsilon^{(+)} = E_{N+1} - E_N$

removal energy: $\varepsilon^{(-)} = E_N - E_{N-1}$

Fundamental challenge of many-body physics

There are **a lot** of states in a many-body system.

E.g. N -particle state in a single-particle basis of size M



Combinatorial growth

⇒ Rapid growth of memory and CPU time needed

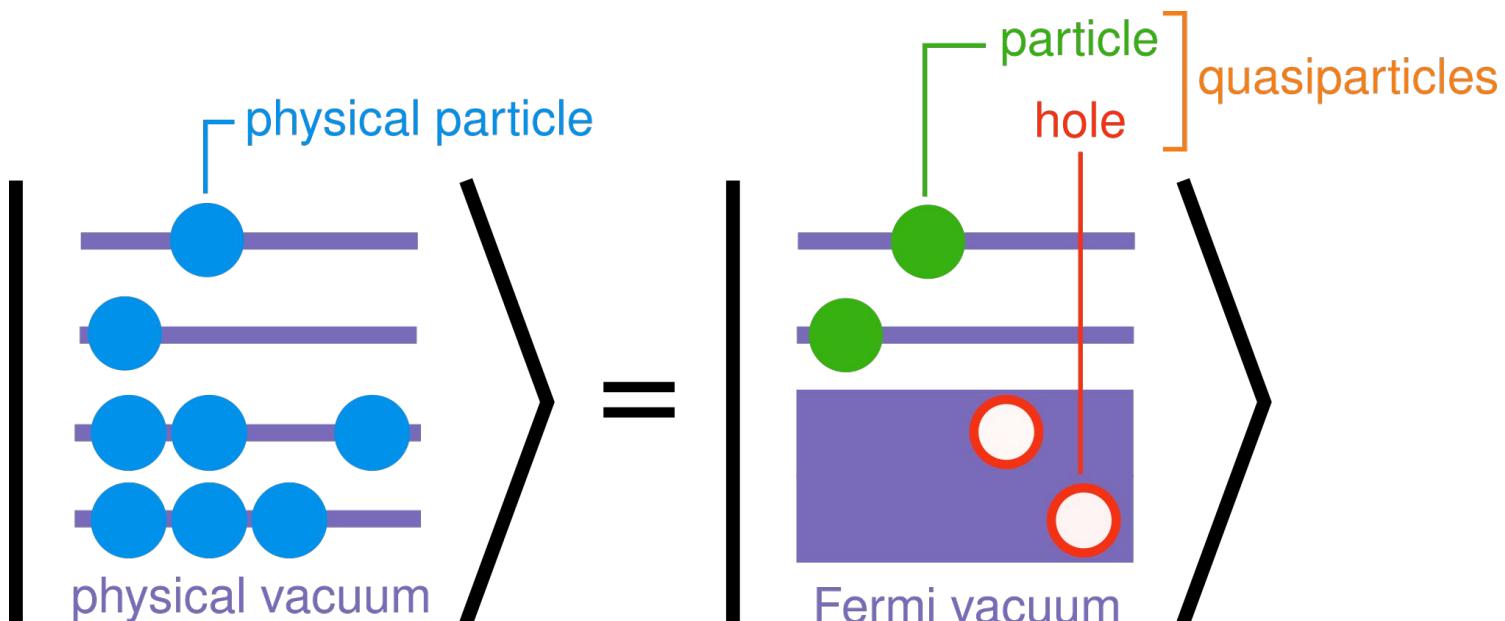
Fortunately, not all states are equally important

Encoding states using a single reference

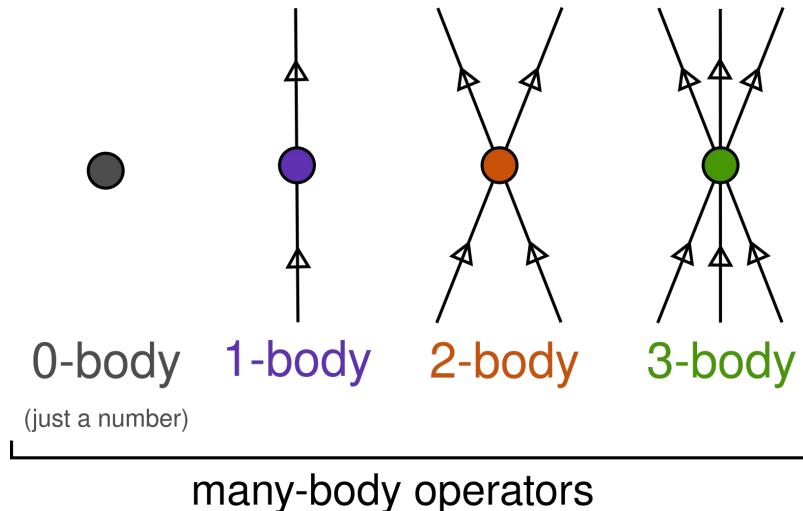
Can we encode states more efficiently in many-body theory?

Start with an approximation of target state → called a **reference state**

Use the reference state as our “background” → (**Fermi vacuum**)



Many-body operators and diagrams



Similar to Feynman diagrams:

Has both formal algebraic and intuitive physical interpretations

- Node = matrix/operator
- Line = single-particle state
- \uparrow = particle
- \downarrow = hole

Example: formal interpretation of a many-body diagram

A complex many-body diagram with nodes A (green), B (red), and C (blue). Lines connect them with various indices: p and q are incoming lines to node A; i , a , and b are outgoing lines from A to B; c is an outgoing line from A to C; r and s are incoming lines to node C. The diagram is equated to a sum of terms involving indices i, a, b, c, r, s :

$$= \frac{1}{8} \sum_{iabc} A_{ipqabc} B_{ai} C_{bcrs}$$

Similarity renormalization group (SRG)

SRG: Continuous unitary transformation (*evolution*) of Hamiltonian

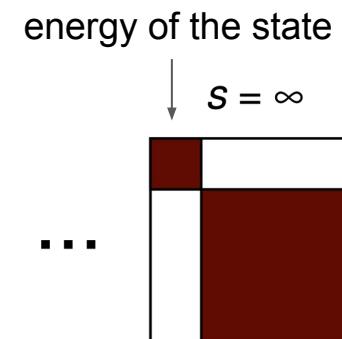
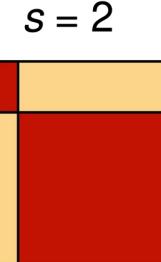
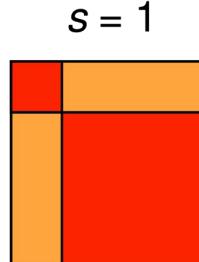
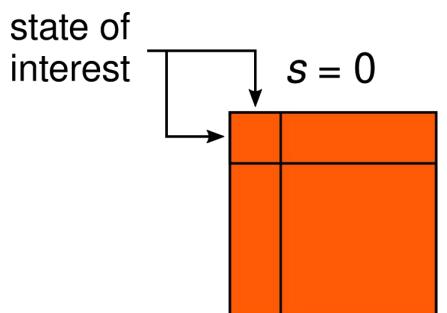
$$\hat{H}(s) = \hat{U}(s)\hat{H}(0)\hat{U}^\dagger(s)$$

unitary: $UU^\dagger = 1$

Defined by the differential equation (*flow equation*)

$$\frac{d}{ds}\hat{H}(s) = [\hat{\eta}(s), \hat{H}(s)]$$

η = generator of the transformation that we get to choose!

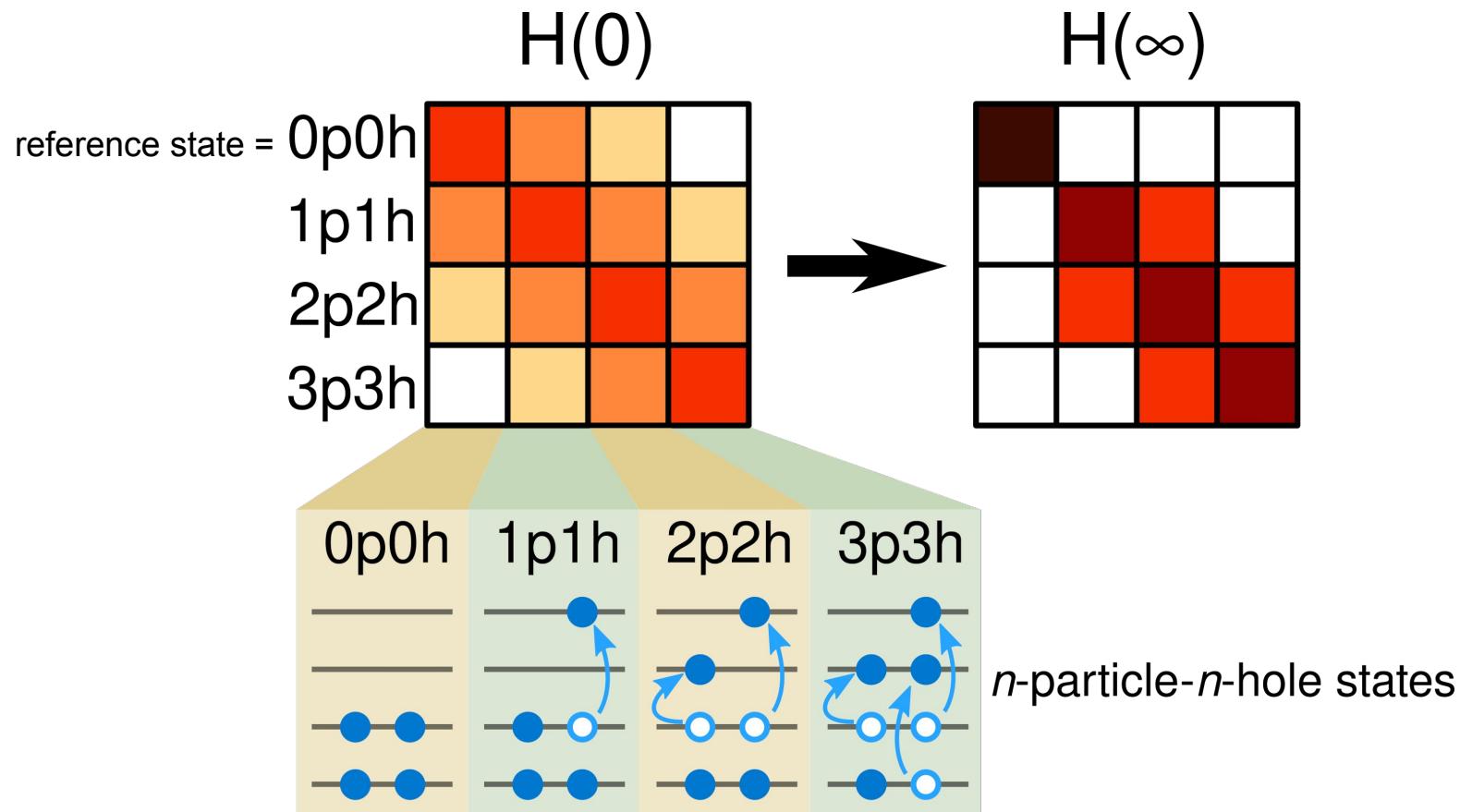


schematic view of
 H matrix

evolved Hamiltonian can be used
as input for other methods

In-medium SRG (IM-SRG)

IM-SRG: SRG using the many-body framework [Hergert et al 2016]

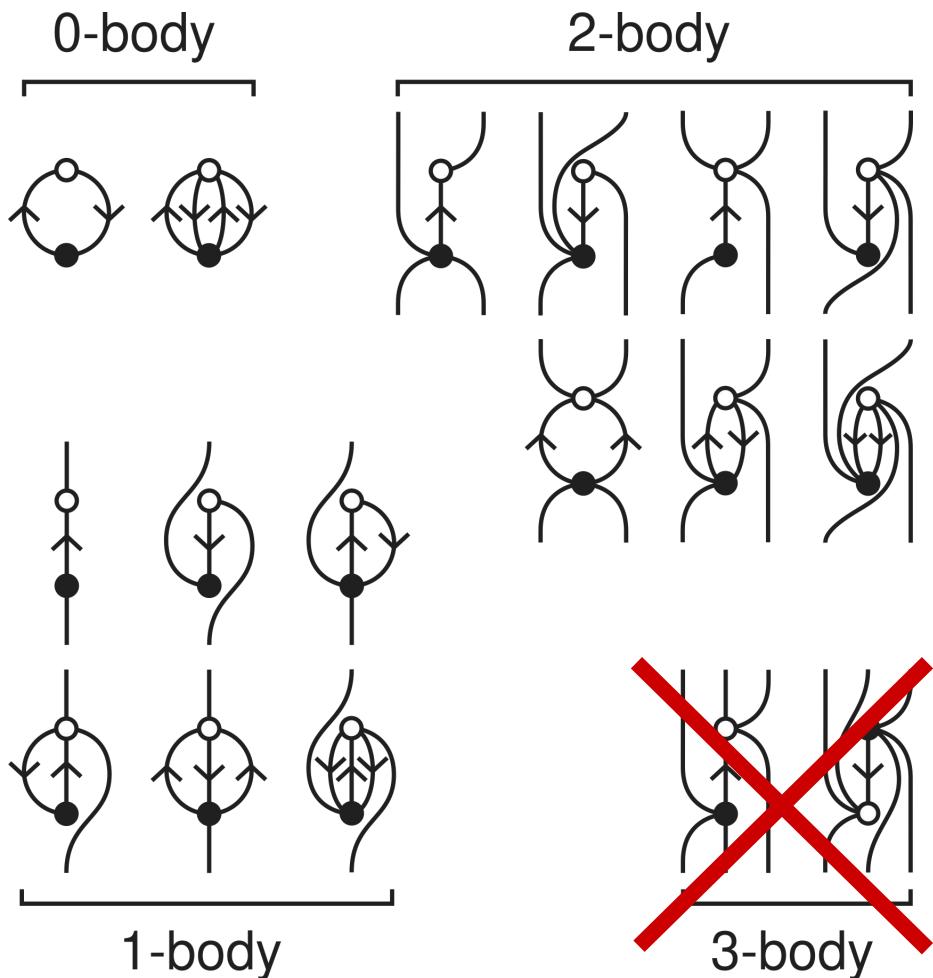


IM-SRG commutator

$$\begin{aligned}\frac{d\hat{H}}{ds} &= [\hat{\eta}, \hat{H}] \\ &= \hat{C}(\hat{\eta}, \hat{H}) - \hat{C}(\hat{H}, \hat{\eta})\end{aligned}$$

where

$$\hat{C}(\circ, \bullet) =$$



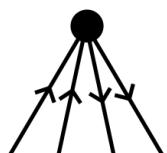
Bulk of the IM-SRG calculation
(most computationally expensive part)

truncated in
IM-SRG(2)

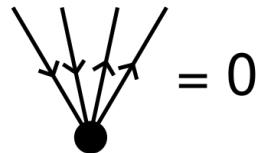
Quasidegenerate perturbation theory (QDPT)

$$\hat{H} = \hat{H}^\circ + \hat{V}$$

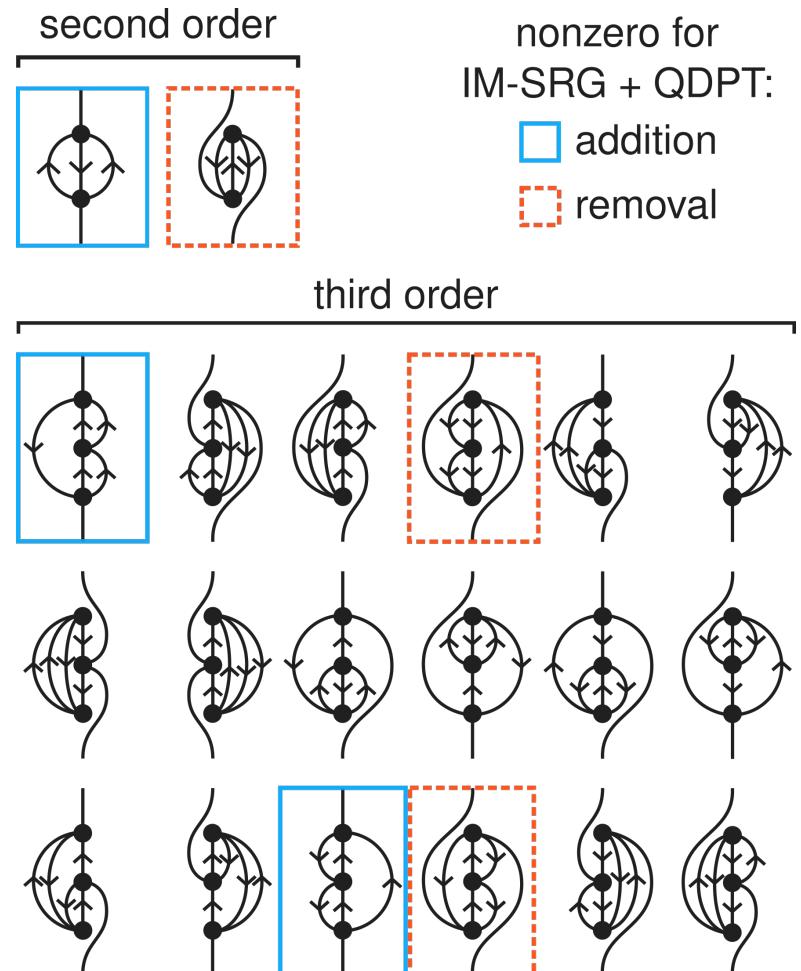
- QDPT: Perturbation theory (PT) for open-shell systems, a.k.a. open-shell PT
- Expand state in orders of V
- **Refines addition and removal energies**
- Diagrams up to 3rd order (QDPT₃) with implicit denominators
- **20** diagrams → IM-SRG eliminates all but **6** of them because



$$= 0$$

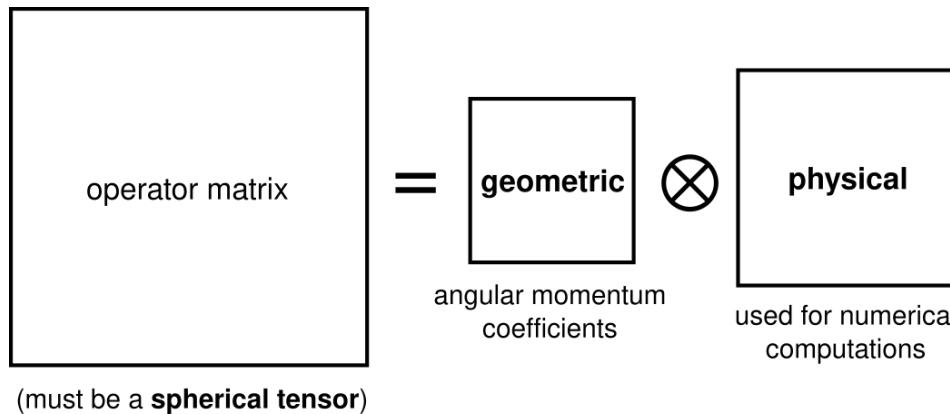


MICHIGAN STATE
UNIVERSITY



J-scheme

- Exploit spherical symmetry of nuclei for efficiency
- Achieved through **J-scheme** (antonym: **M-scheme**)
- Angular momentum coupling *increases matrix sparsity*
- Wigner–Eckart theorem:



- Products of geometric parts (*angular momentum expressions*) can be simplified!

Example of a derivation in J-scheme

- Derivation of equations begin in M-scheme, which is much easier to work with
- *Example:* Convert this M-scheme equation into J-scheme:

$$C_{pq} = \sum_{iab} A_{ipab} B_{abiq}$$

- Technical detail: For extra fun, we will assume A, B, and C to be *spherical tensor operators* instead of scalars. This is not used in our code, but appears in [Parzuchowski (2017) *PhD thesis*]

Example: equation before simplification

$$\begin{aligned}
 & (-1)^{j_p - m_p} \begin{pmatrix} j_p & j_C & j_q \\ m_p & m_C & m_q \end{pmatrix} \bar{C}_{pq} \\
 = & \sum_{j_{ip} m_{ip} j_{ab} m_{ab} j'_{ab} m'_{ab} j_{iq} m_{iq} iab} \\
 & (-1)^{2j_B + j_C - m_C} \sqrt{2j_C + 1} \begin{pmatrix} j_A & j_C & j_B \\ m_A & -m_C & m_B \end{pmatrix} \\
 & (-1)^{2j_p + j_{ip} - m_{ip}} \sqrt{2j_{ip} + 1} \begin{pmatrix} j_i & j_{ip} & j_p \\ m_i & -m_{ip} & m_p \end{pmatrix} \\
 & (-1)^{2j_b + j_{ab} - m_{ab}} \sqrt{2j_{ab} + 1} \begin{pmatrix} j_a & j_{ab} & j_b \\ m_a & -m_{ab} & m_b \end{pmatrix} \\
 & (-1)^{j_{ip} - m_{ip}} \begin{pmatrix} j_{ip} & j_A & j_{ab} \\ m_{ip} & m_A & m_{ab} \end{pmatrix} \bar{A}_{ipab} \\
 & (-1)^{2j_b + j'_{ab} - m'_{ab}} \sqrt{2j'_{ab} + 1} \begin{pmatrix} j_a & j'_{ab} & j_b \\ m_a & -m'_{ab} & m_b \end{pmatrix} \\
 & (-1)^{2j_q + j_{iq} - m_{iq}} \sqrt{2j_{iq} + 1} \begin{pmatrix} j_i & j_{iq} & j_q \\ m_i & -m_{iq} & m_q \end{pmatrix} \\
 & (-1)^{j'_{ab} - m'_{ab}} \begin{pmatrix} j'_{ab} & j_B & j_{iq} \\ m'_{ab} & m_B & m_{iq} \end{pmatrix} \bar{B}_{abiq}
 \end{aligned}$$

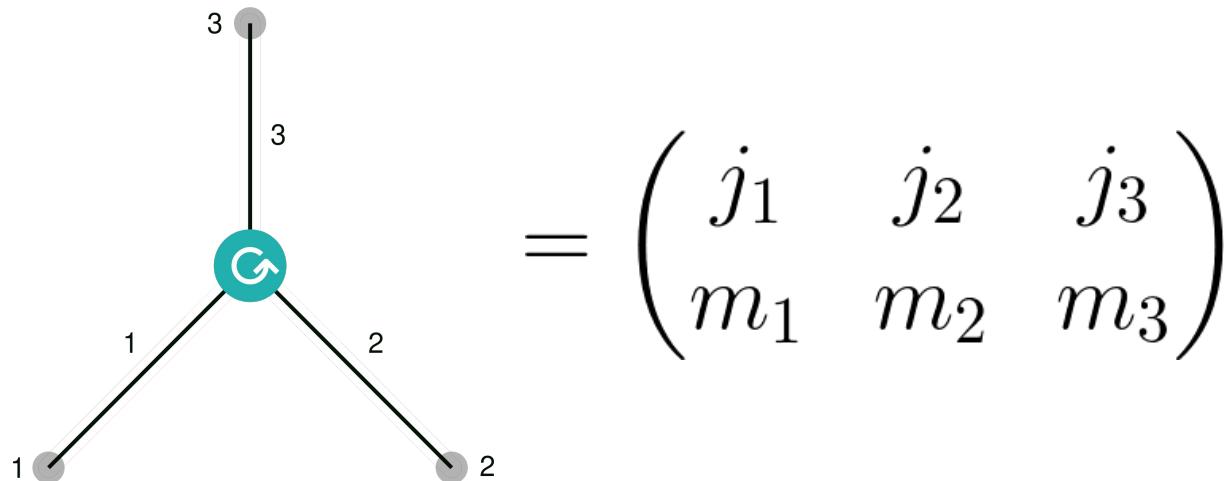
- ~ 32 variables
- Algebraic approach:
error-prone + non-intuitive

3-jm symbols: building blocks of angular momentum expressions

physical (reduced) matrix element

Intermission: Angular momentum diagrams

- Inventor: [Jucys/Yutsis 1962]
- Useful for manipulating angular momentum expressions
- Each node is a 3-jm symbol
- Each line is a (j, m) pair



Example: Diagrammatic derivation with **jucys**

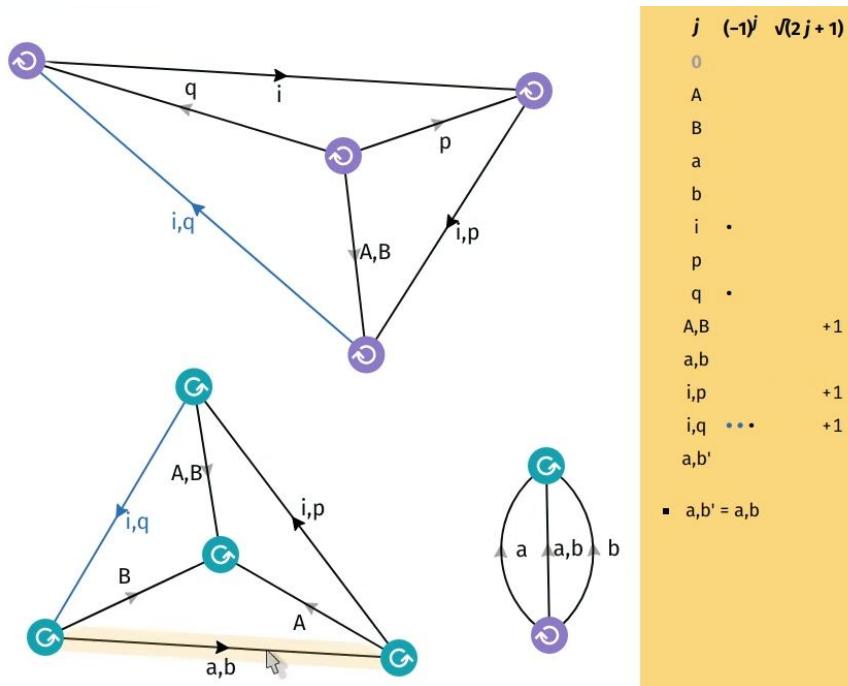
$$C_{pq} = \sum_{iab} A_{ipab} B_{abiq}$$

Input:

```
wet (i + p) A (a + b)
wet (a + b) B (i + q)
wet p (A + B) q
```

<https://rufflewind.com/img/jucys-diagram.mp4>

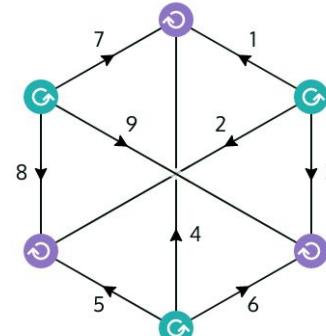
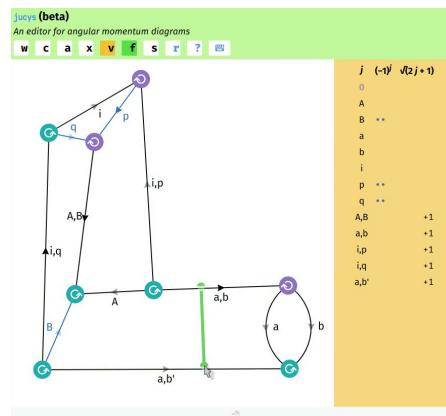
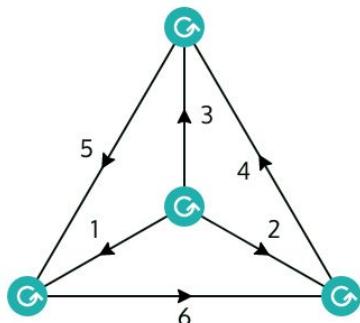
Example: Interpreting the diagrammatic result



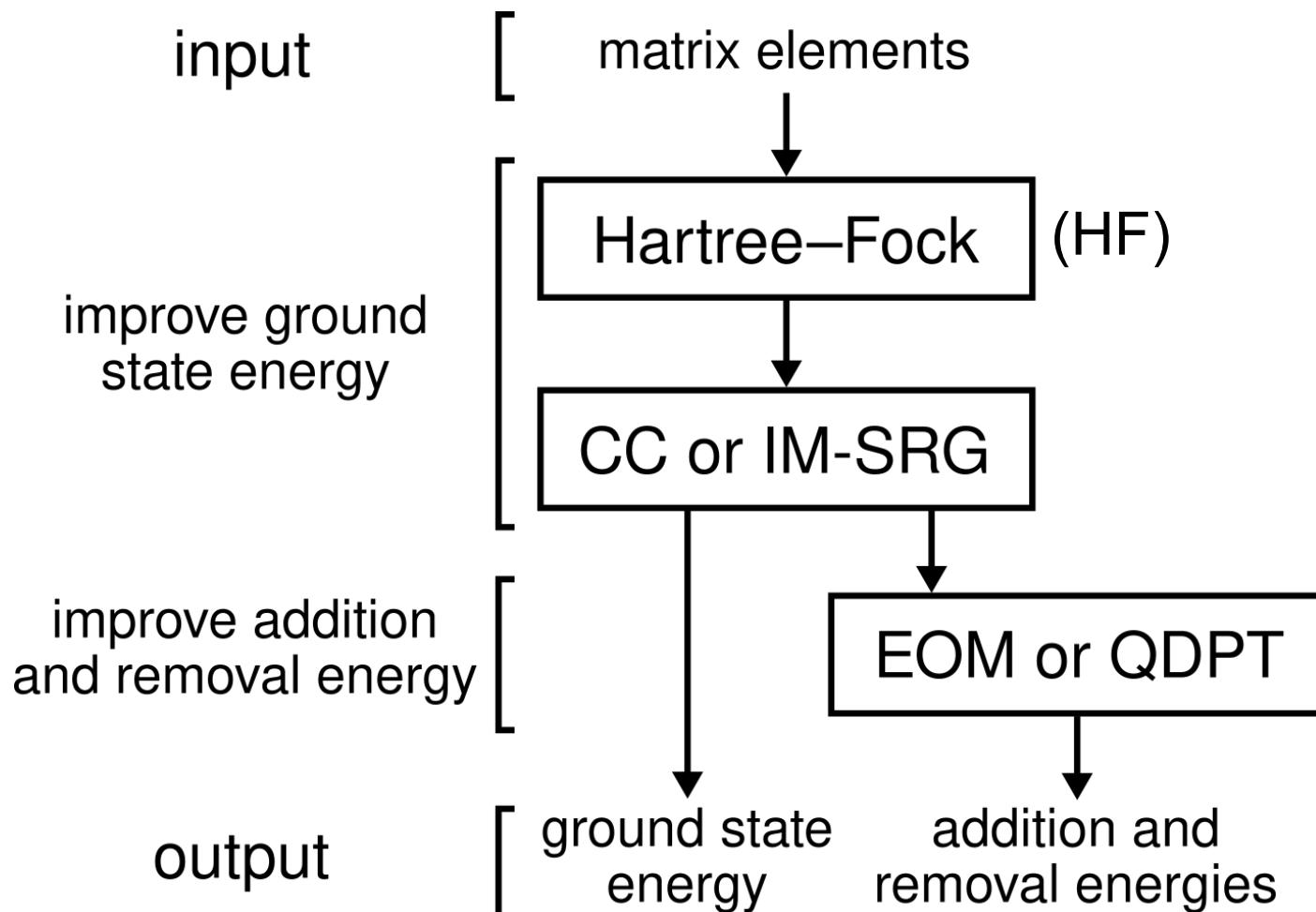
$$\bar{C}_{pq} = \sum_{j_{ip} j_{ab} j_{iq} i^{ab}} (-1)^{j_i + j_q - j_{iq}} \sqrt{2j_C + 1} \sqrt{2j_{ip} + 1} \sqrt{2j_{iq} + 1} \\ \left\{ \begin{matrix} j_p & j_C & j_q \\ j_{iq} & j_i & j_{ip} \end{matrix} \right\} \left\{ \begin{matrix} j_C & j_B & j_A \\ j_{ab} & j_{ip} & j_{iq} \end{matrix} \right\} \bar{A}_{ipab} \bar{B}_{abiq}$$

Graphical tool for angular momentum diagrams

- My open-source graphical JavaScript app:
 - <https://rufflewind.com/jucys>
 - Try it online! (requires keyboard + mouse)
- Display/draw/edit any angular momentum diagram
- **Input tool:** create diagrams from a textual description
- **Reduction mode:** ensure all change preserve equality; enforce rules of the “game” to prevent mistakes



Theory pipeline



Implementation of many-body methods

- My open-source Rust library for many-body theory:
 - <https://github.com/xrf/lutario>
 - Implements: HF, MP2, **IM-SRG(2)**, QDPT3
 - Systems: **quantum dots**, homogeneous electron gas, infinite nuclear matter, **nuclei** (soon: neutron drops)
- Flexible: general many-body machinery that works for any quantum system, either J- or M-scheme
- Extending code to support new systems involves just two aspects: input matrix elements + single-particle basis

MP2: 2nd order Møller–Plesset PT

Rust: a modern systems programming language focused on performance, safety, and concurrency

Code quality

- Extensive testing and cross-checks for HF, IM-SRG(2) commutator, and QDPT3 in multiple quantum systems
- Automated testing and failure notification (continuous integration)
- API documentation available through Rustdoc
- Adheres to official style convention of language

Struct lutario::half::Half

```
pub struct Half<T>(pub T);
```

[–] Type that logically behaves like half-integers, but what is actually stored is twice its logical value.

For example, `Half(3)` represents the fraction $3/2$.

Methods

```
impl<T> Half<T>
[–] pub fn twice(self) -> T
    Unwrap to twice its logical value.
```

```
impl<T: Add<Output = T> + Clone> Half<T>
pub fn double(self) -> Self
```

```
impl<T: Signed> Half<T>
pub fn abs(self) -> Half<T>
```

```
impl<T: Signed + Ord> Half<T>
pub fn in_multiplet_of(self, j: Self) -> bool
```

```
impl<T: Clone + Div<Output = T> + Rem<Output = T> + [src]
Zero + One> Half<T>
```

```
[–] pub fn try_get(self) -> Result<T, Self>
    Get the value if it's half-even. Otherwise, returns Err(self).
```

[–] [src]

Crate lutario

[–] Lutario is an implementation of IM-SRG(2) and QDPT3.

The best way to get started is to look at the tests and examples. The codebase are guaranteed to stay up to date thanks to automated testing.

The code is currently organized in a very monolithic manner to enable submodules stabilize and become sufficiently generic, they may even help reduce bloat as well as compilation times.

Modules

ang_mom	Angular momentum coupling.
basis	Basis manipulation.
block	Block-diagonal matrices and similar things.
block_mat	Packed block-diagonal matrices.
blkrc_trn_mat	Doubled block-diagonal lower-triangular matrices.

```
/// Term 011
///
/// ````text
/// C[] -> α Σ[i a] Ji^2 A[i a] B[a i]
/// ``
pub fn c011(
    alpha: f64,
    a1: &OpJ100<f64>,
    b1: &OpJ100<f64>,
    c0: &mut f64,
)
{
    let scheme = a1.scheme();
    for i in scheme.states_10(&[occ:I]) {
        for a in i.costates_10(&[occ:A]) {
            *c0 += alpha * i.jweight(2) * a1.at(i, a) * b1.at(a, i);
        }
    }
}
```

Systems: quantum dots and nuclei

Quantum dots

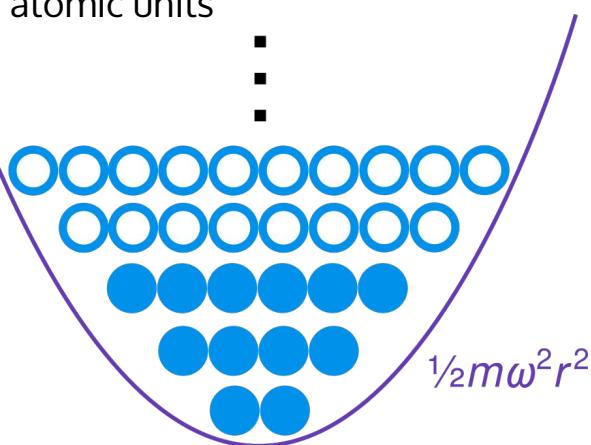
2D system of N electrons in a quadratic well

Potential: harmonic oscillator of frequency ω

Interaction: standard Coulomb repulsion

Basis: harmonic oscillator of same frequency ω
(K = number of shells)

Units: atomic units



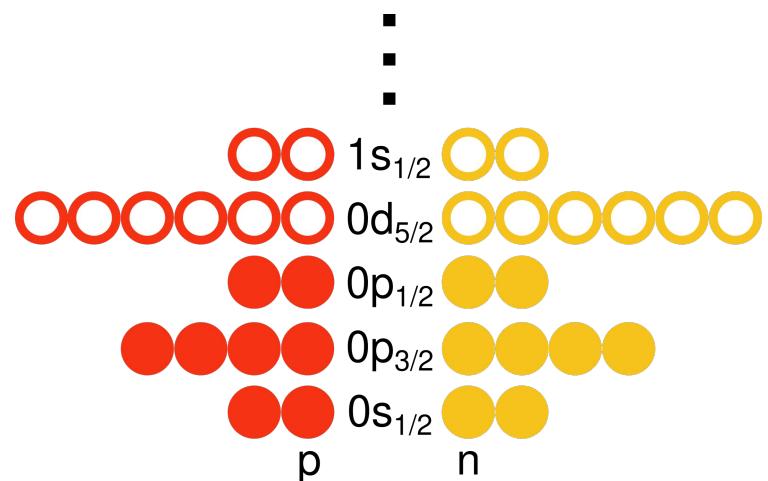
Nuclei

3D self-bound system of Z protons and N neutrons

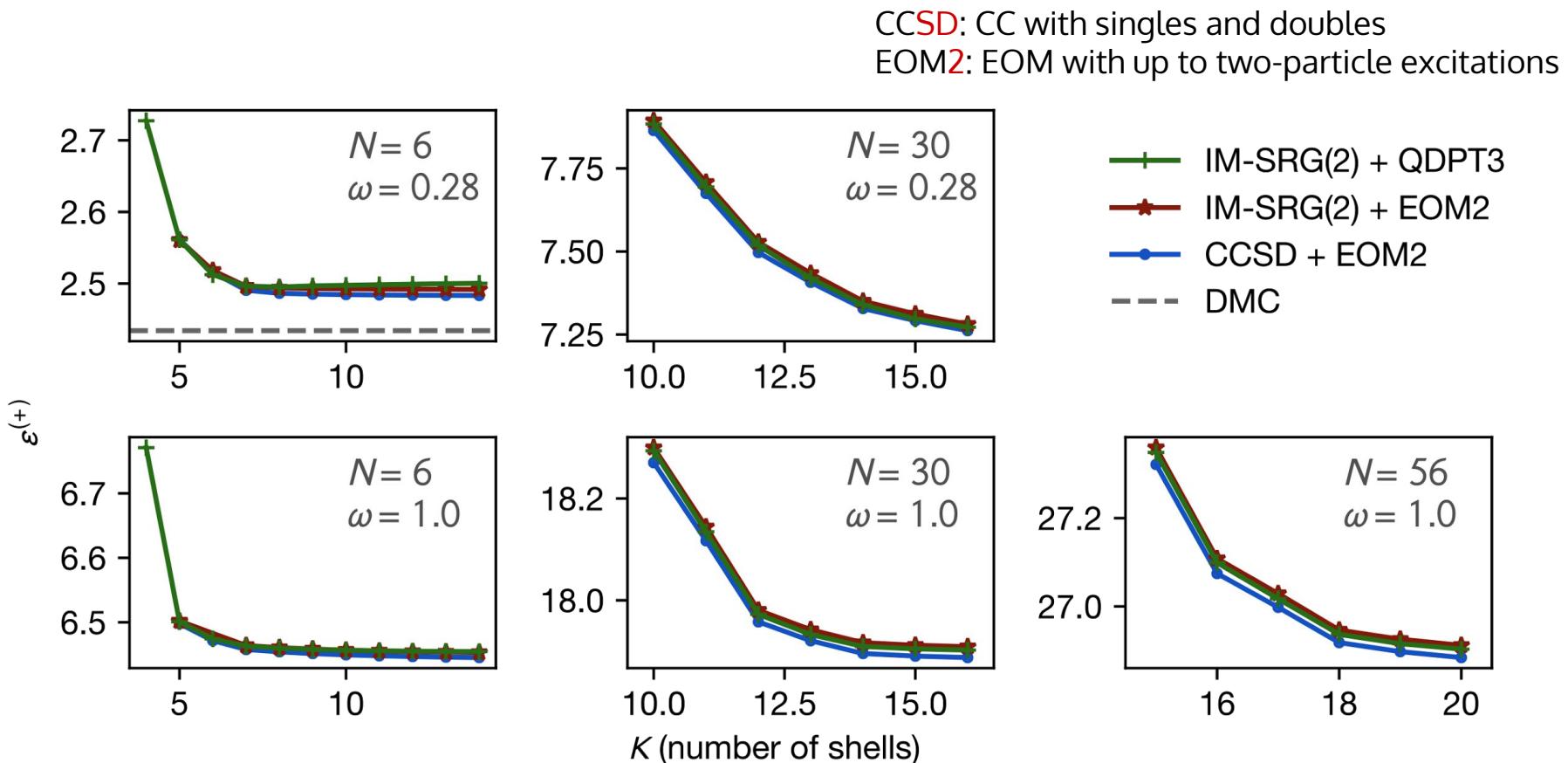
Interaction: Entem–Machleidt (2003) chiral EFT
 N^3LO ($\Lambda = 500$ MeV) SRG-softened to $\lambda = 2 \text{ fm}^{-1}$

Basis: harmonic oscillator of frequency ω
($e_{\max} + 1$ = number of shells)

Units: nuclear units (MeV and fm)



Addition energies of quantum dots



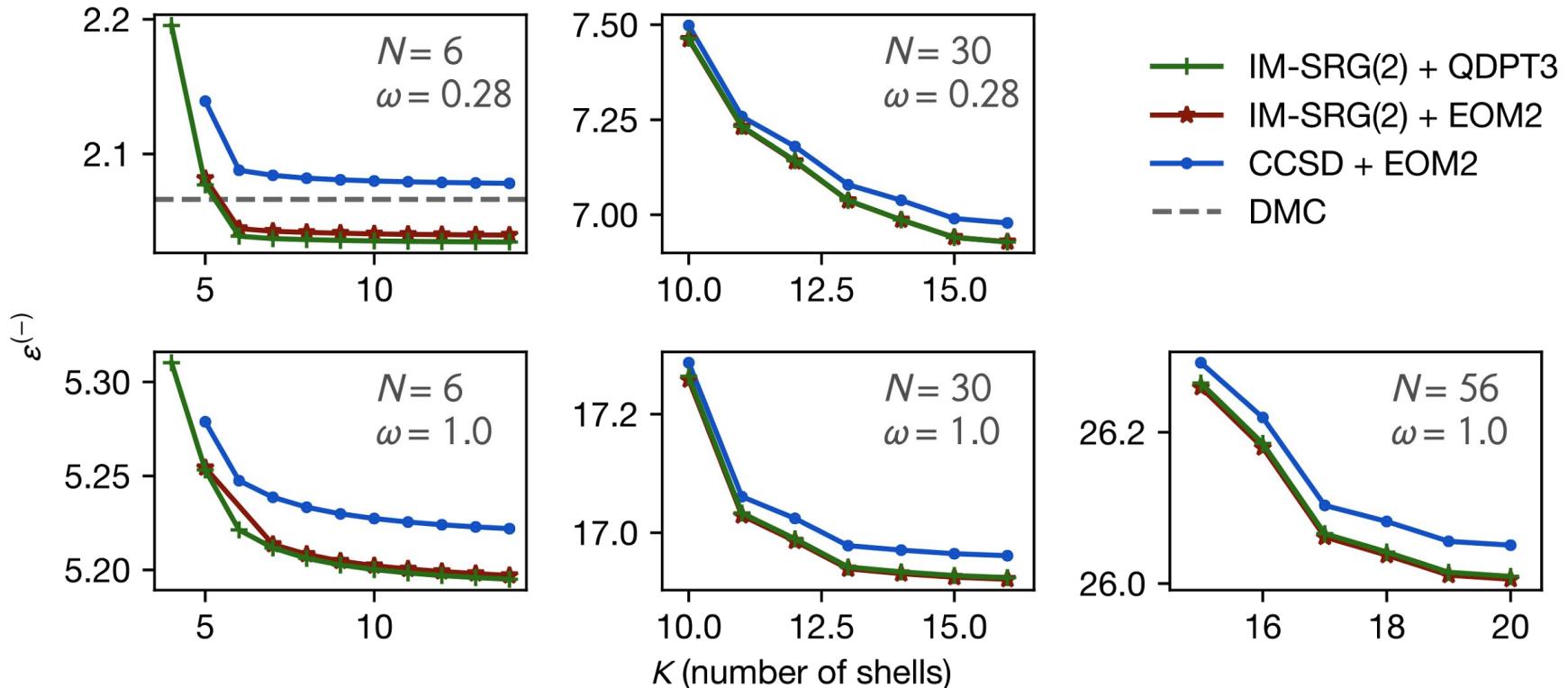
Collaboration:

IM-SRG(2) + EOM2 results by Nathan M. Parzuchowski

CCSD + EOM2 results by Samuel J. Novario

Yuan et al. *JCP* (2017) [doi:10.1063/1.4995615](https://doi.org/10.1063/1.4995615)

Removal energies of quantum dots



Collaboration:

IM-SRG(2) + EOM2 results by Nathan M. Parzuchowski

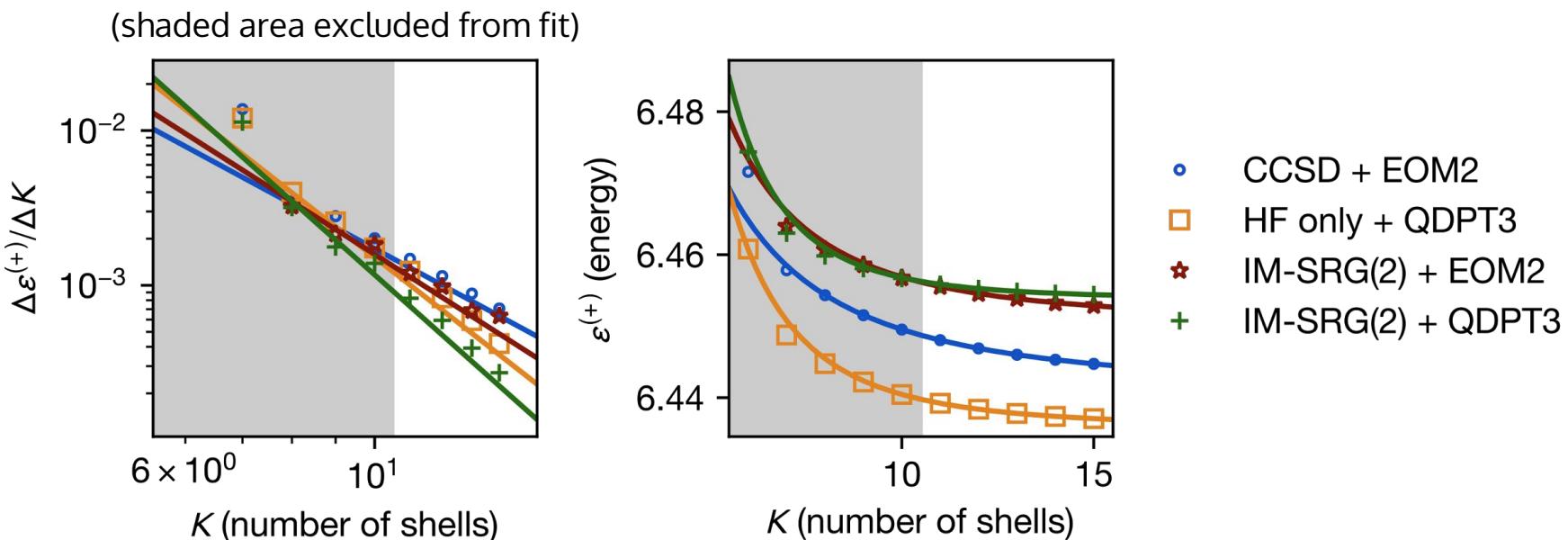
CCSD + EOM2 results by Samuel J. Novario

Yuan et al. *JCP* (2017) [doi:10.1063/1.4995615](https://doi.org/10.1063/1.4995615)

Extrapolation of quantum dot energies

$$\varepsilon = \alpha K^{-\beta} + \gamma$$

Kvaal (2009) *PRB*



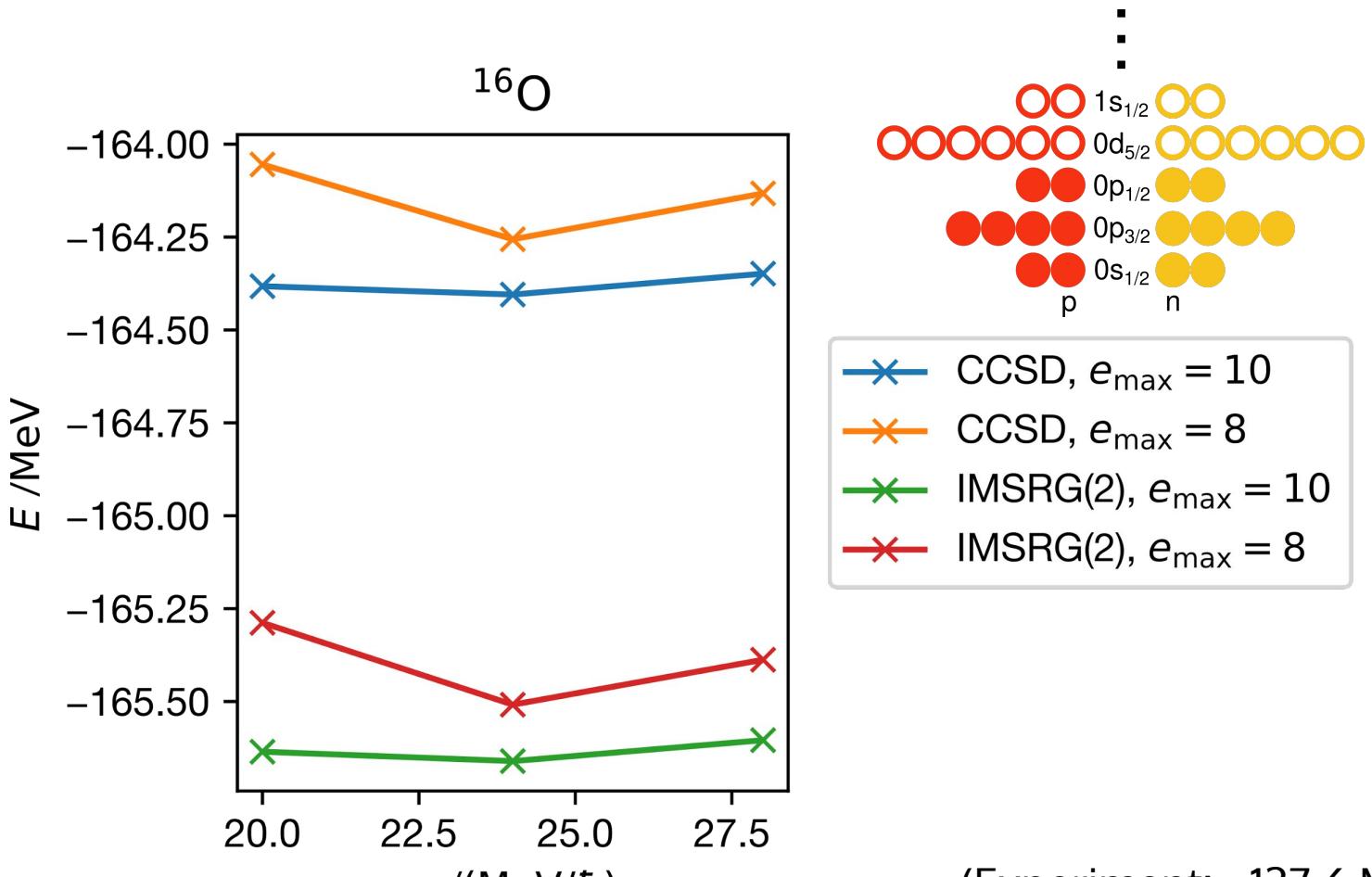
Collaboration:

IM-SRG(2) + EOM2 results by Nathan M. Parzuchowski

CCSD + EOM2 results by Samuel J. Novario

Yuan et al. (2017) *JCP* [doi:10.1063/1.4995615](https://doi.org/10.1063/1.4995615)

Ground state energy of ^{16}O

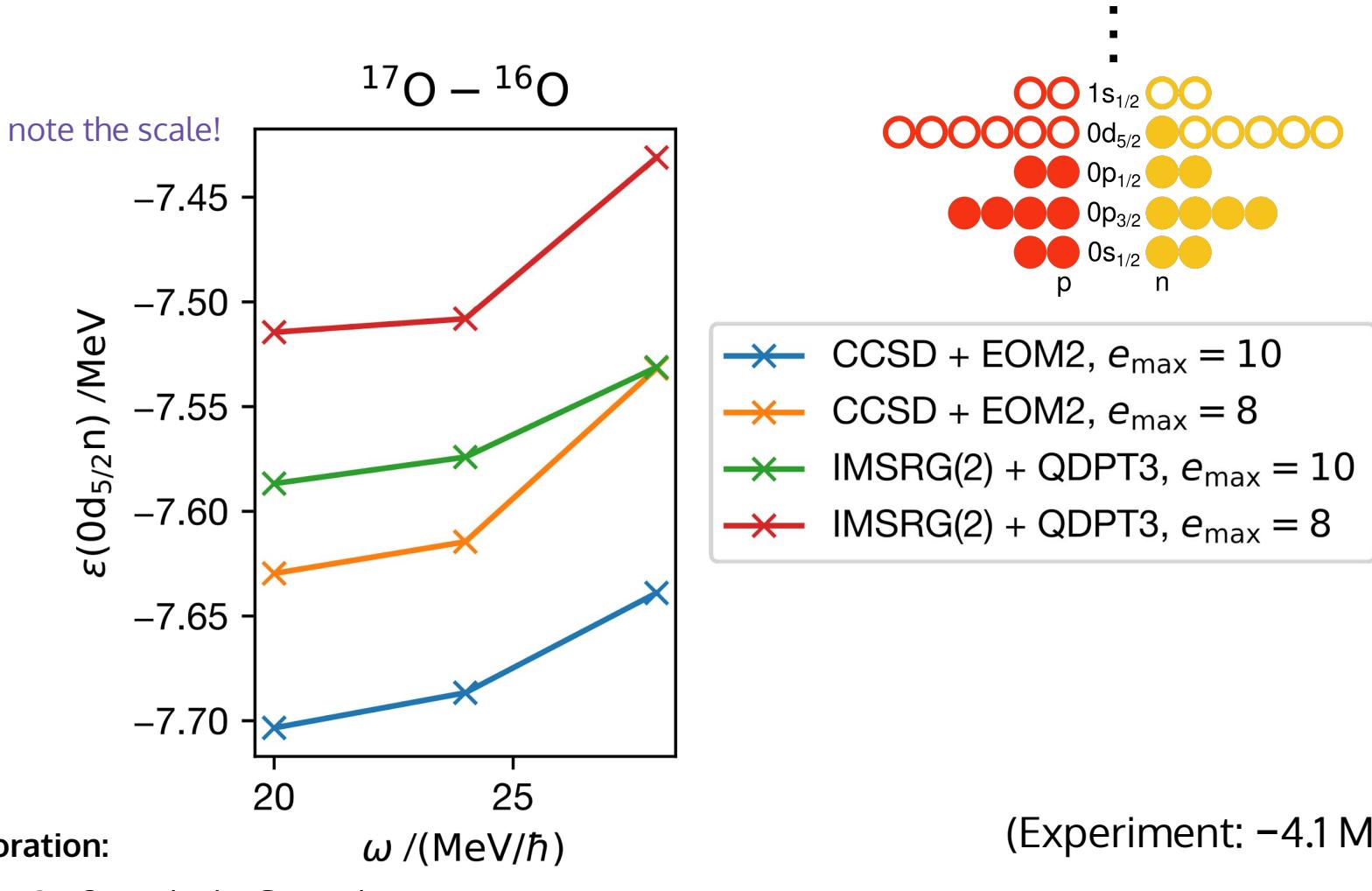


Collaboration:

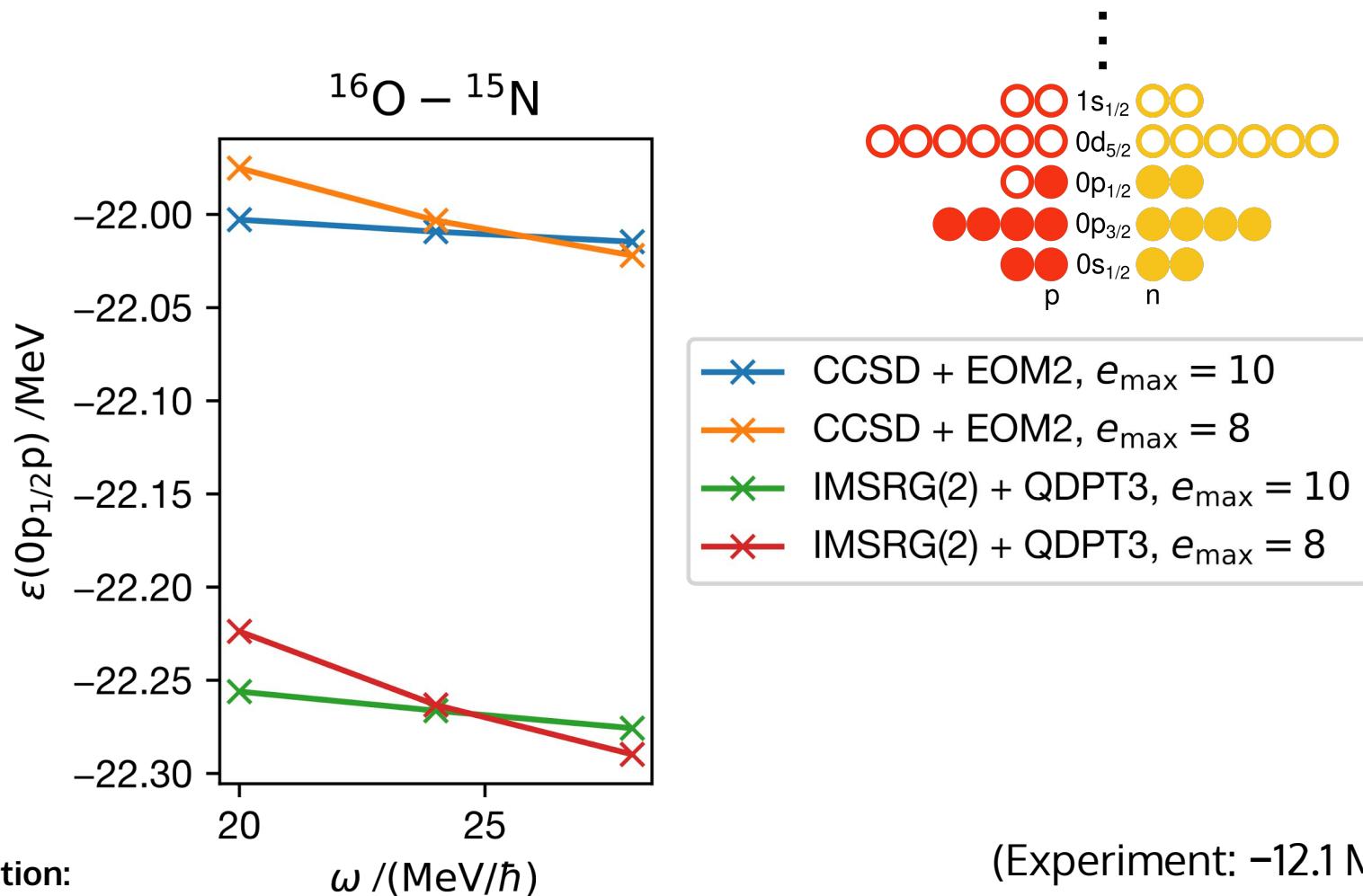
CCSD + EOM2 results by Samuel J. Novario

(Experiment: -127.6 MeV)

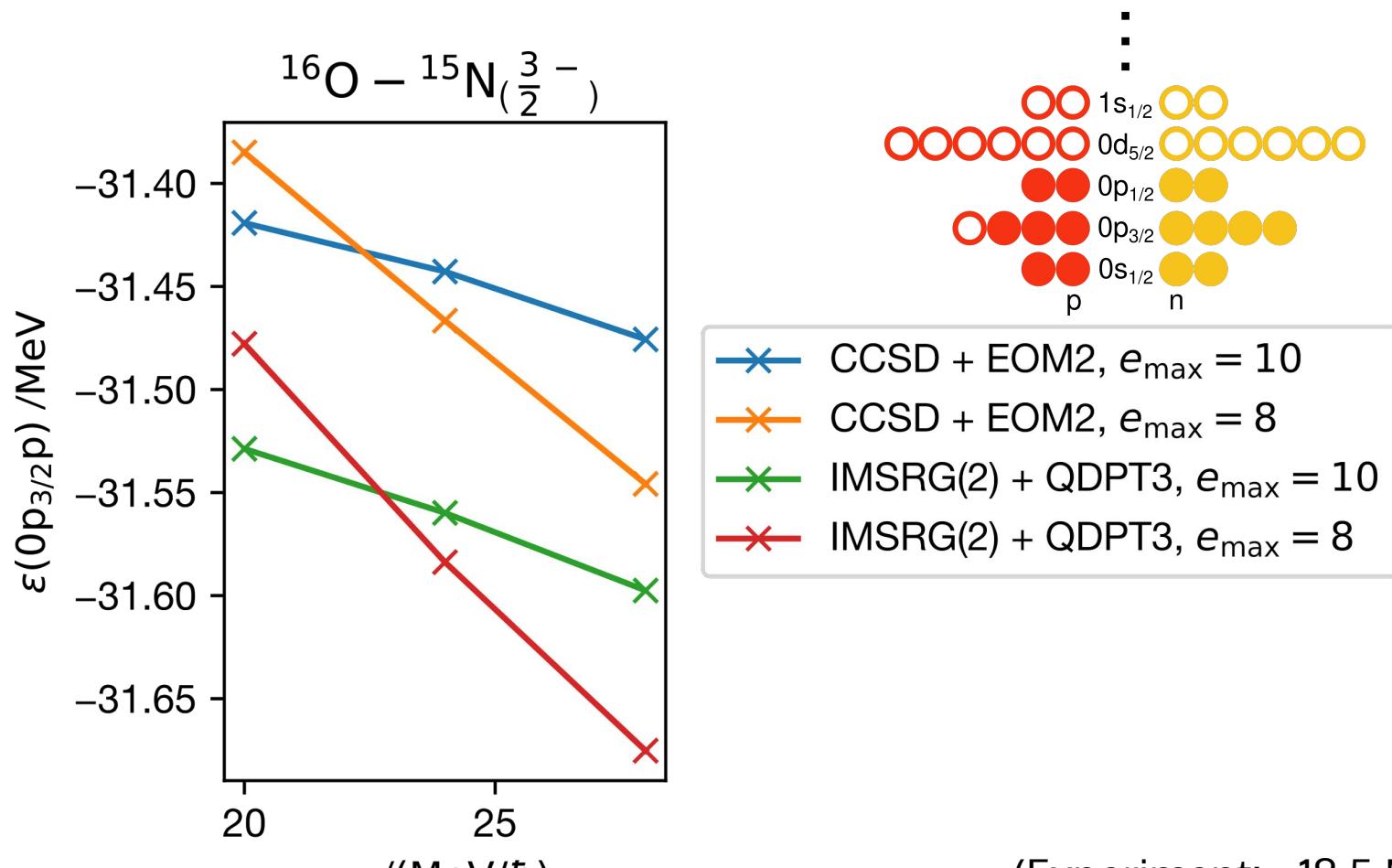
Addition energy of ^{17}O from ^{16}O



Removal energy of $^{15}\text{N}(1/2^-)$ from ^{16}O



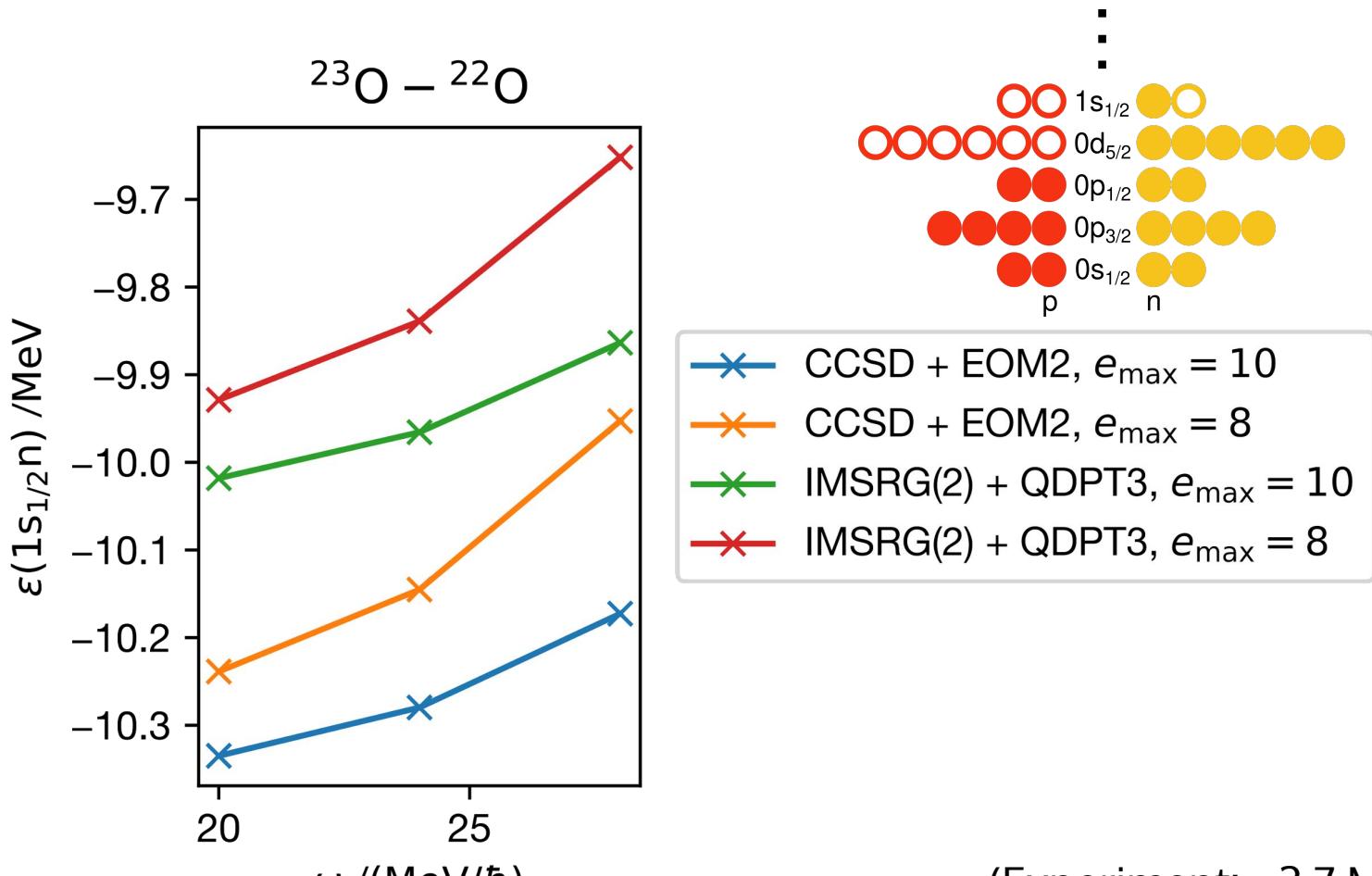
Removal energy of excited $^{15}\text{N}(3/2^-)$ from ^{16}O



Collaboration:

CCSD + EOM2 results by Samuel J. Novario

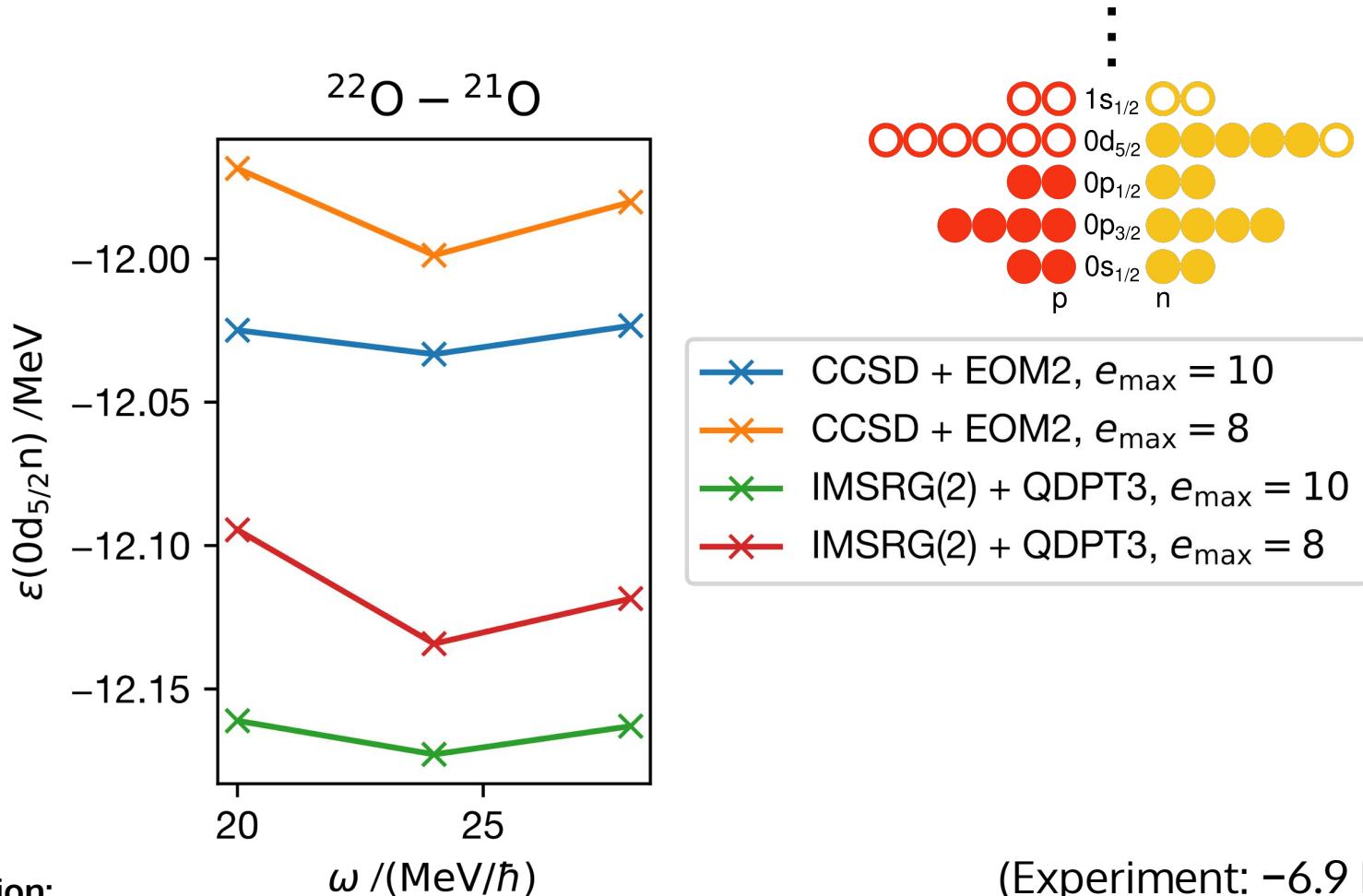
Addition energy of ^{23}O from ^{22}O



Collaboration:

CCSD + EOM2 results by Samuel J. Novario

Removal energy of ^{21}O from ^{22}O



Collaboration:

CCSD + EOM2 results by Samuel J. Novario

MICHIGAN STATE
UNIVERSITY

Recap

- **jucys**: Graphical tool for angular momentum diagrams
 - **lutario**: Flexible, open-source code for J- and M-scheme many-body theories
-
- Calculated addition and removal energies of quantum dots and nuclei using IM-SRG + QDPT
 - Results in good agreement with collaborators'
 - Nuclei: qualitative agreement with experimental values

Future directions

- Explore more parameters: basis size, isotope, interaction
- Analyze convergence with respect to
 - size of single-particle basis
 - frequency of the harmonic oscillator basis
- Extrapolate to infinite basis limit
- Approximate inclusion of 3-body forces
- Higher order IM-SRG and QDPT (e.g. IM-SRG(2*), QDPT4)
- Code optimizations → larger system, higher precision
- Study other systems: infinite matter, neutron drops
- Explore other observables

Thank you!

Many-body theory group

- Morten Hjorth-Jensen (*advisor*)
- Scott Bogner (*co-advisor*)
- Heiko Hergert
- Nathan M. Parzuchowski
(EOM-IMSRG results)
- Samuel J. Novario
(EOM-CC results)
- Titus D. Morris
- Justin Lietz

Committee

- Morten Hjorth-Jensen (*chair*)
- Scott Bogner
- Alexandra Gade
- Carlo Piermarocchi
- Scott Pratt