

Supplemental Information for The Role of High-Order
Electron Correlation Effects in a Model System for
Non-valence Correlation-bound Anions

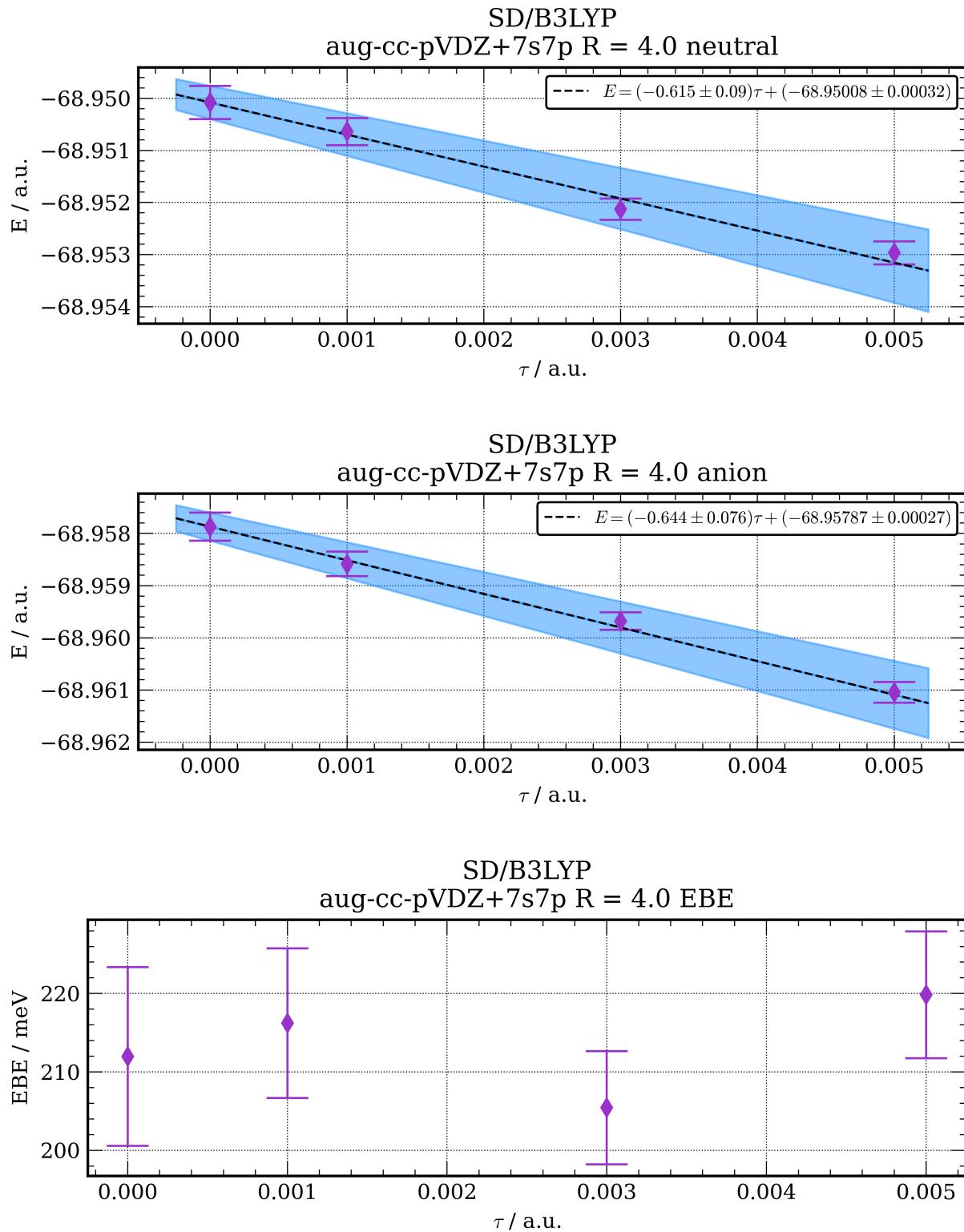
Shiv Upadhyay, Amanda Dumi, James Shee, Kenneth D. Jordan

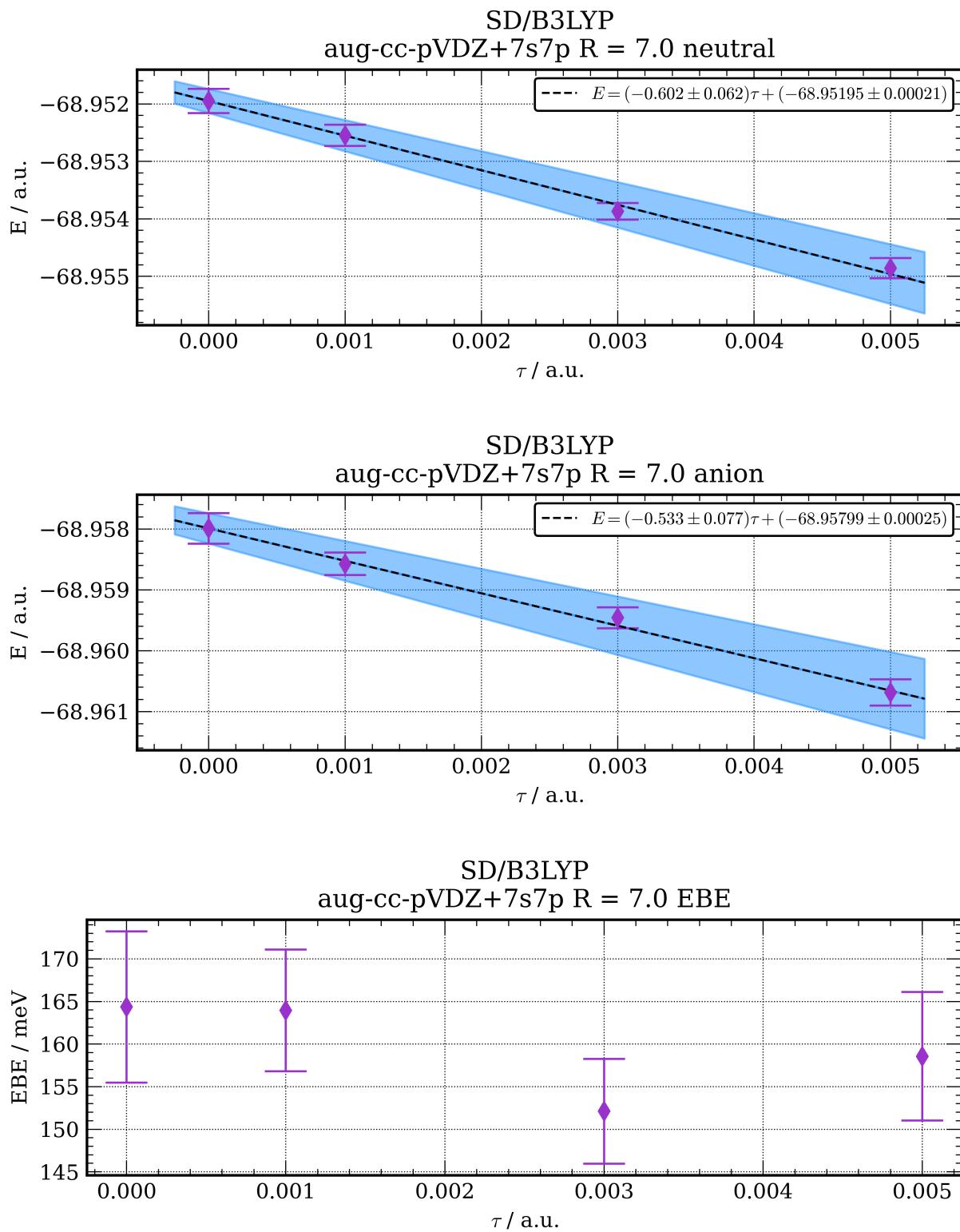
September 23, 2020

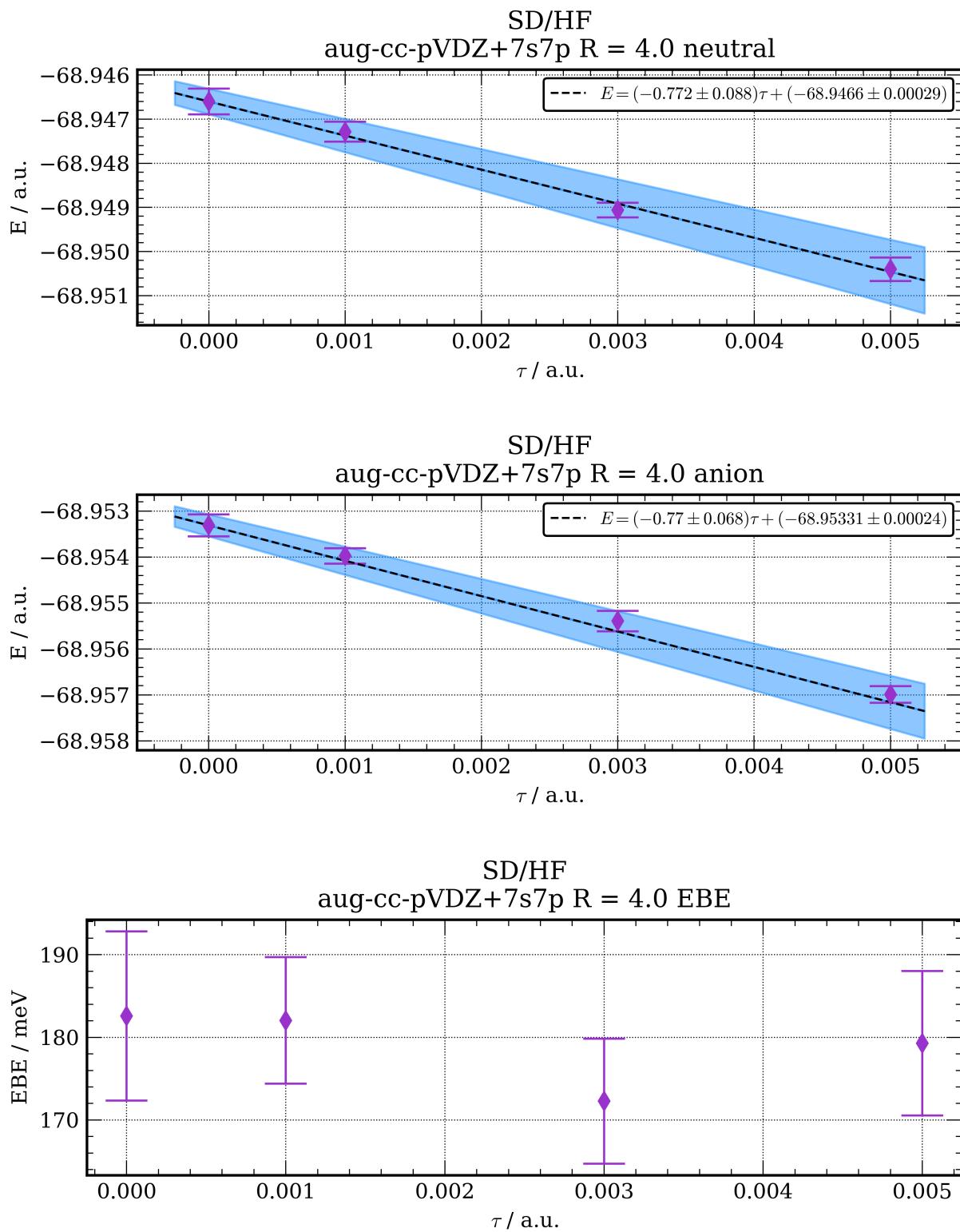
CONTENTS

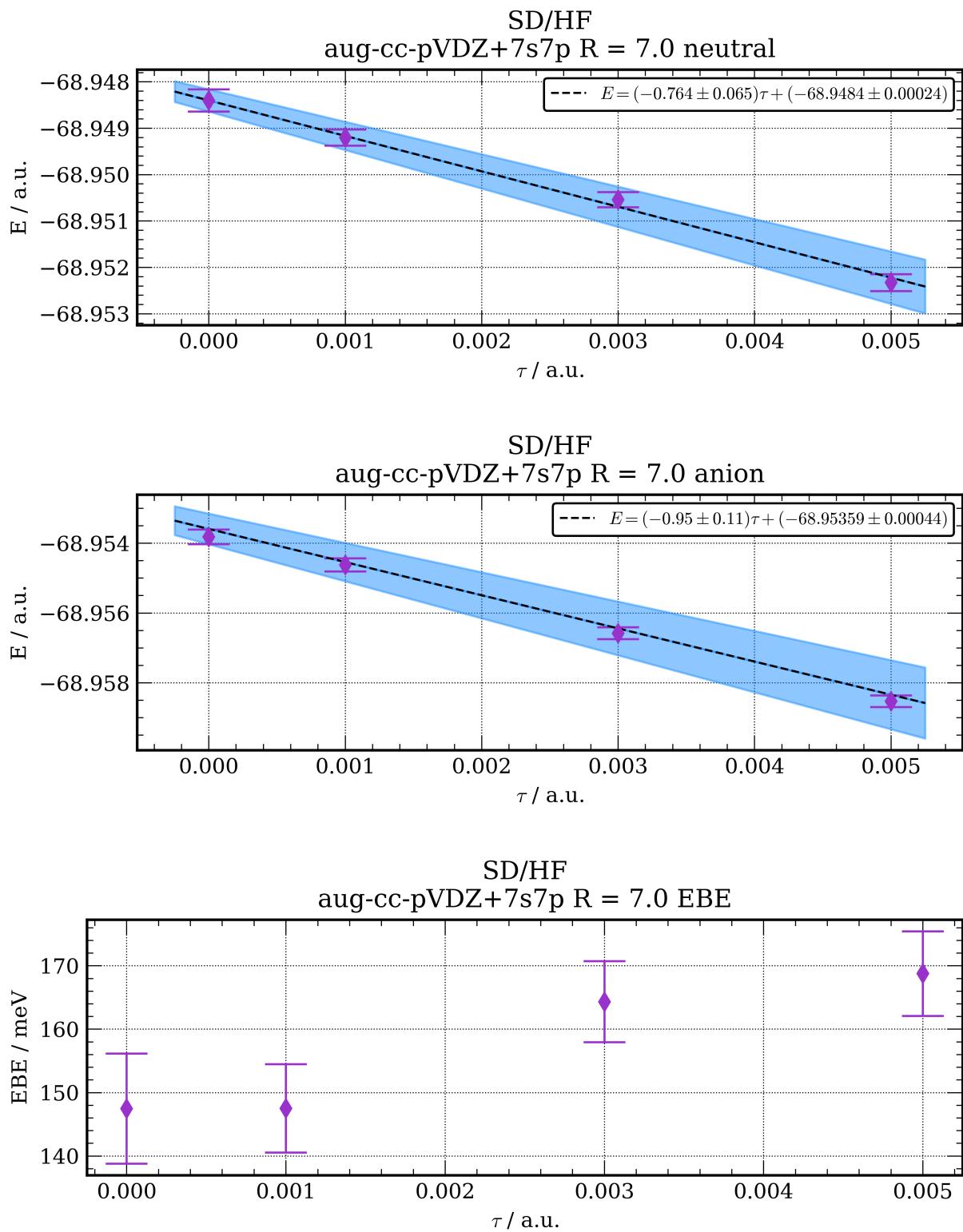
1 SUPPLEMENTAL BASIS FUNCTIONS

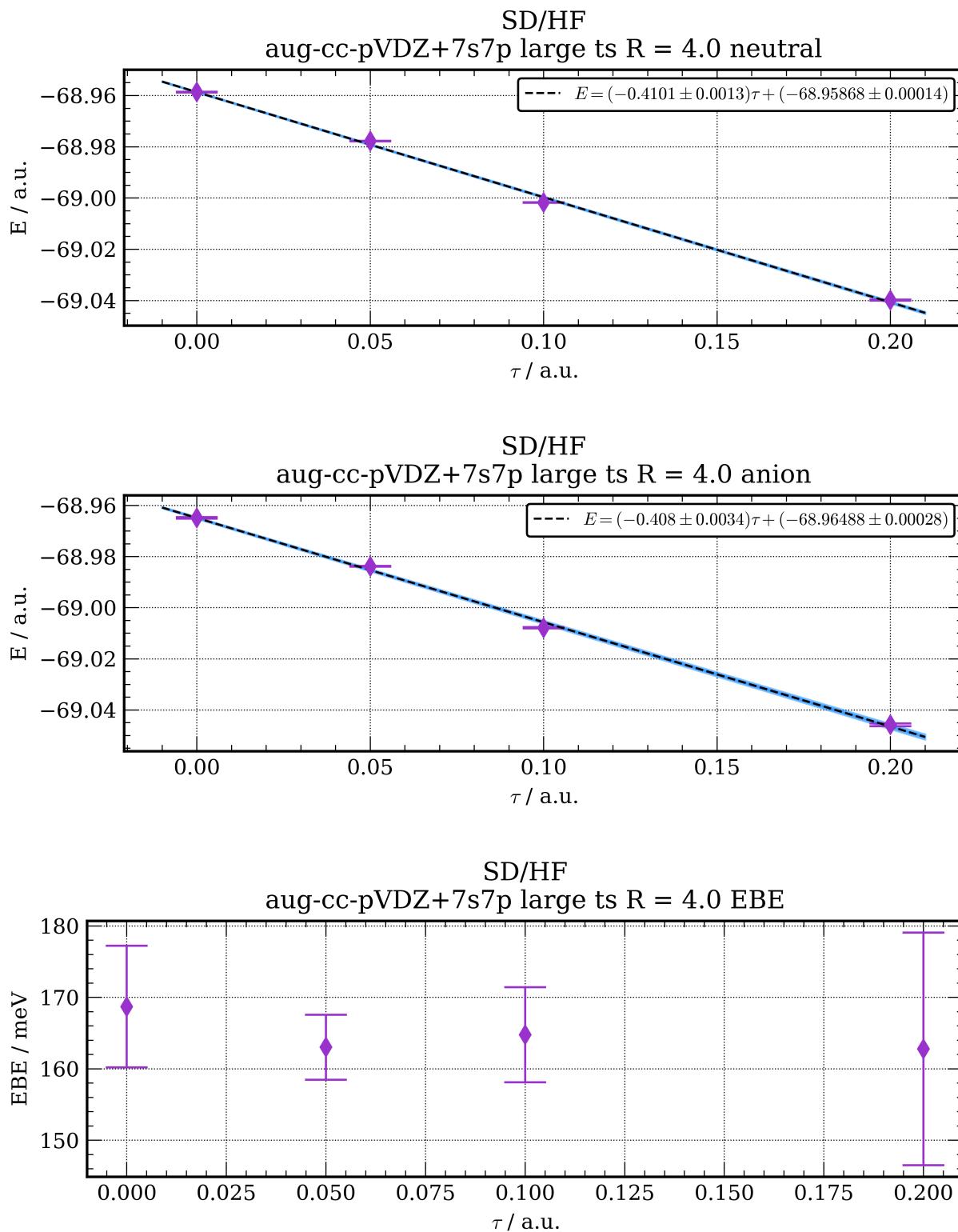
2 DMC EXTRAPOLATION

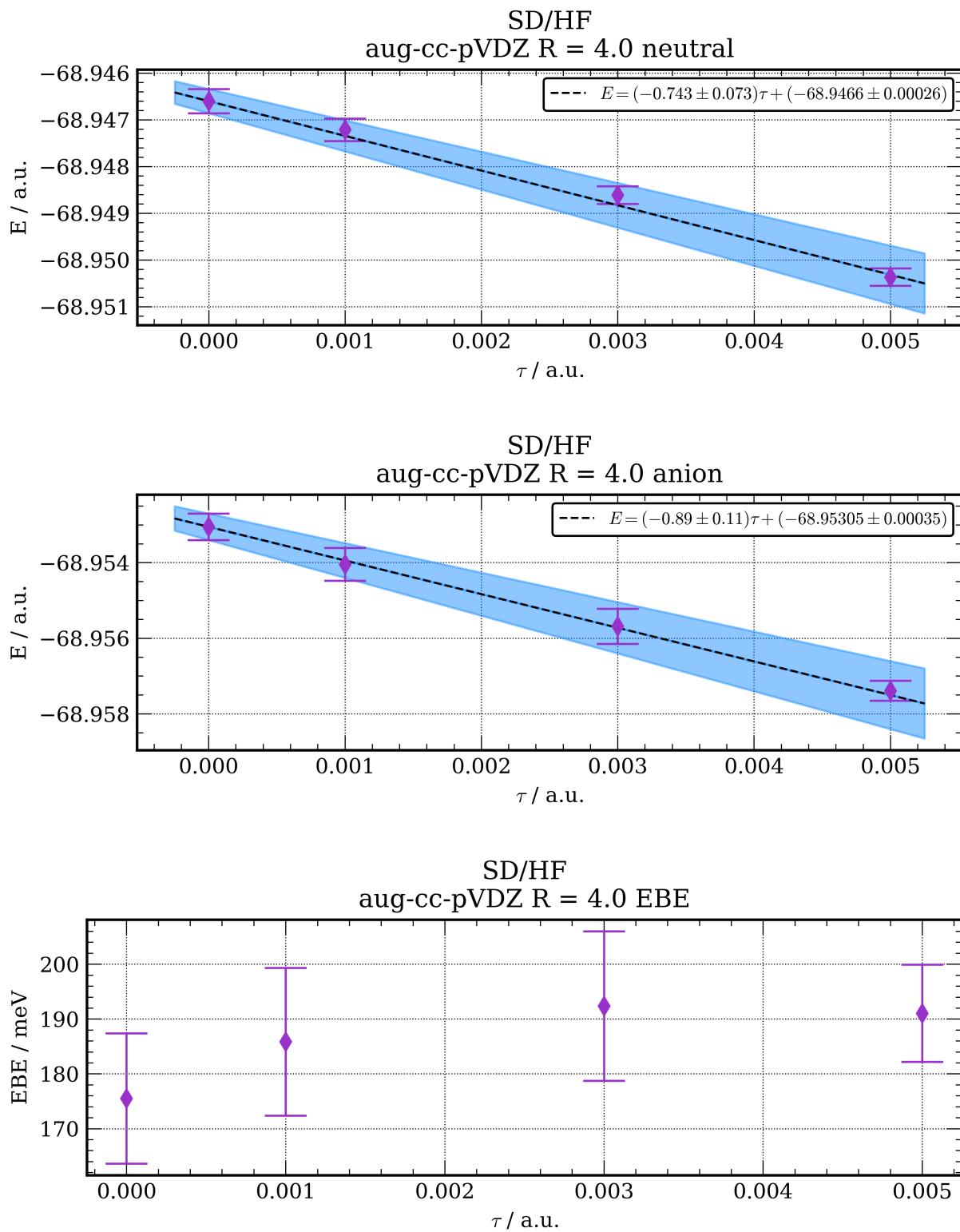


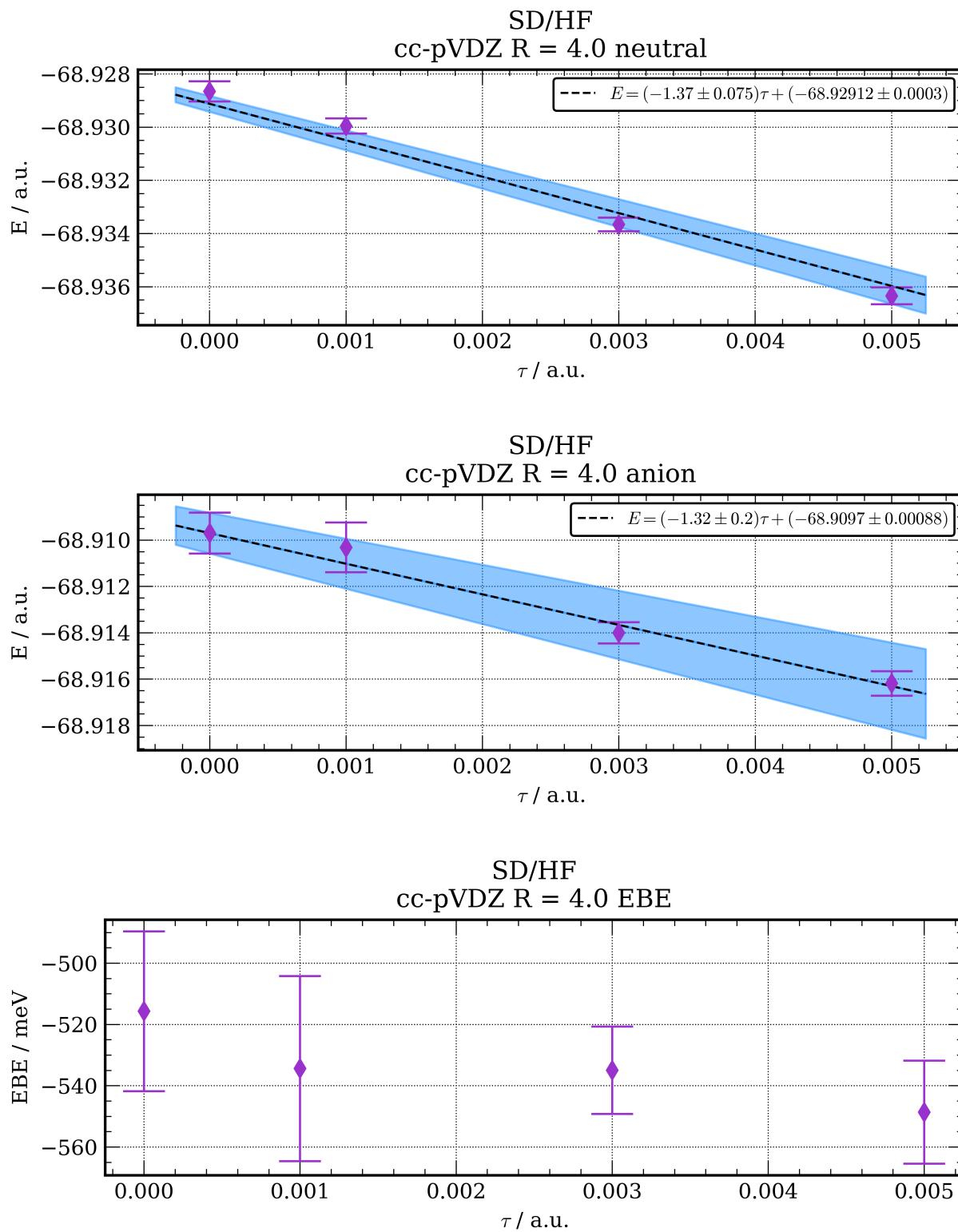


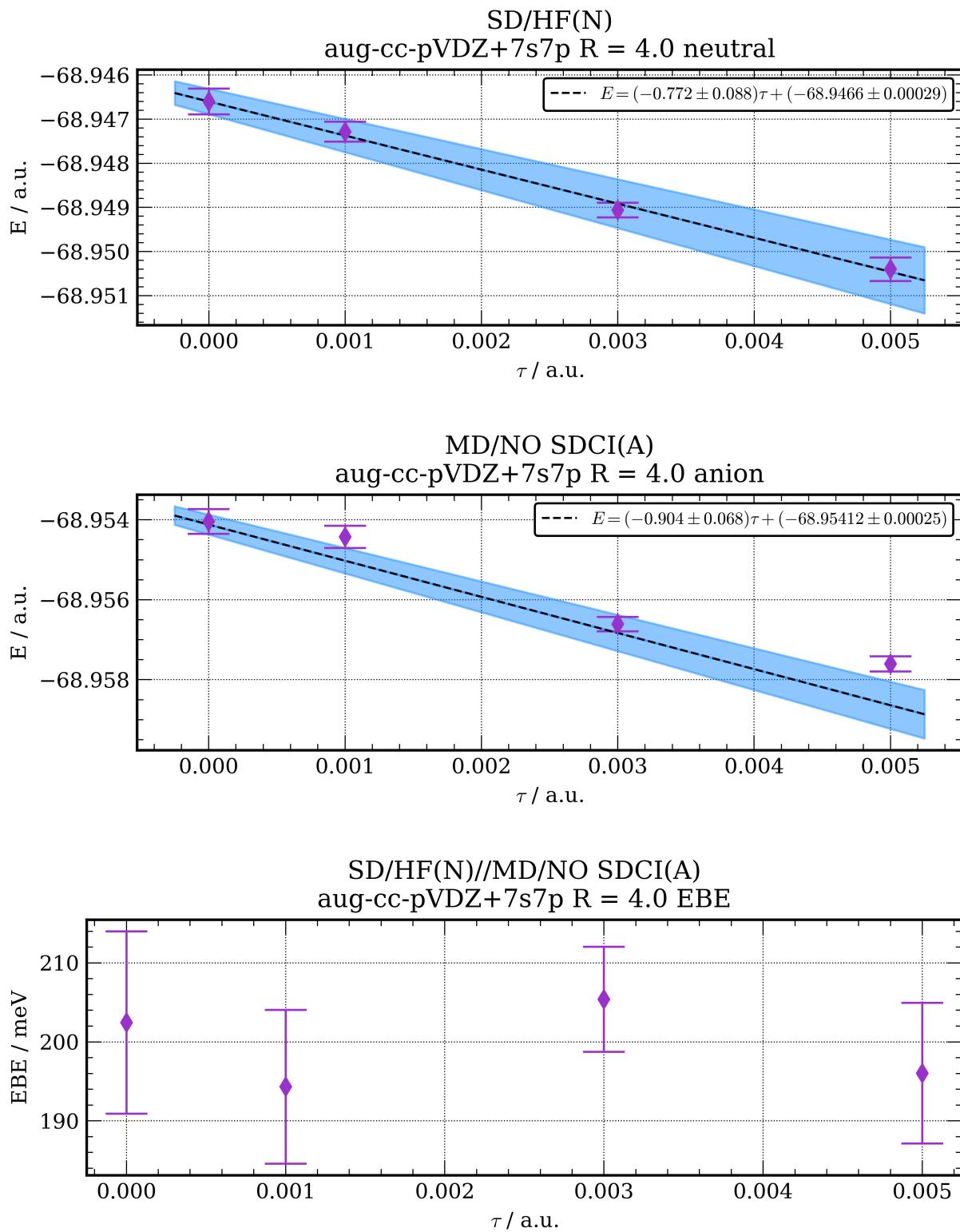


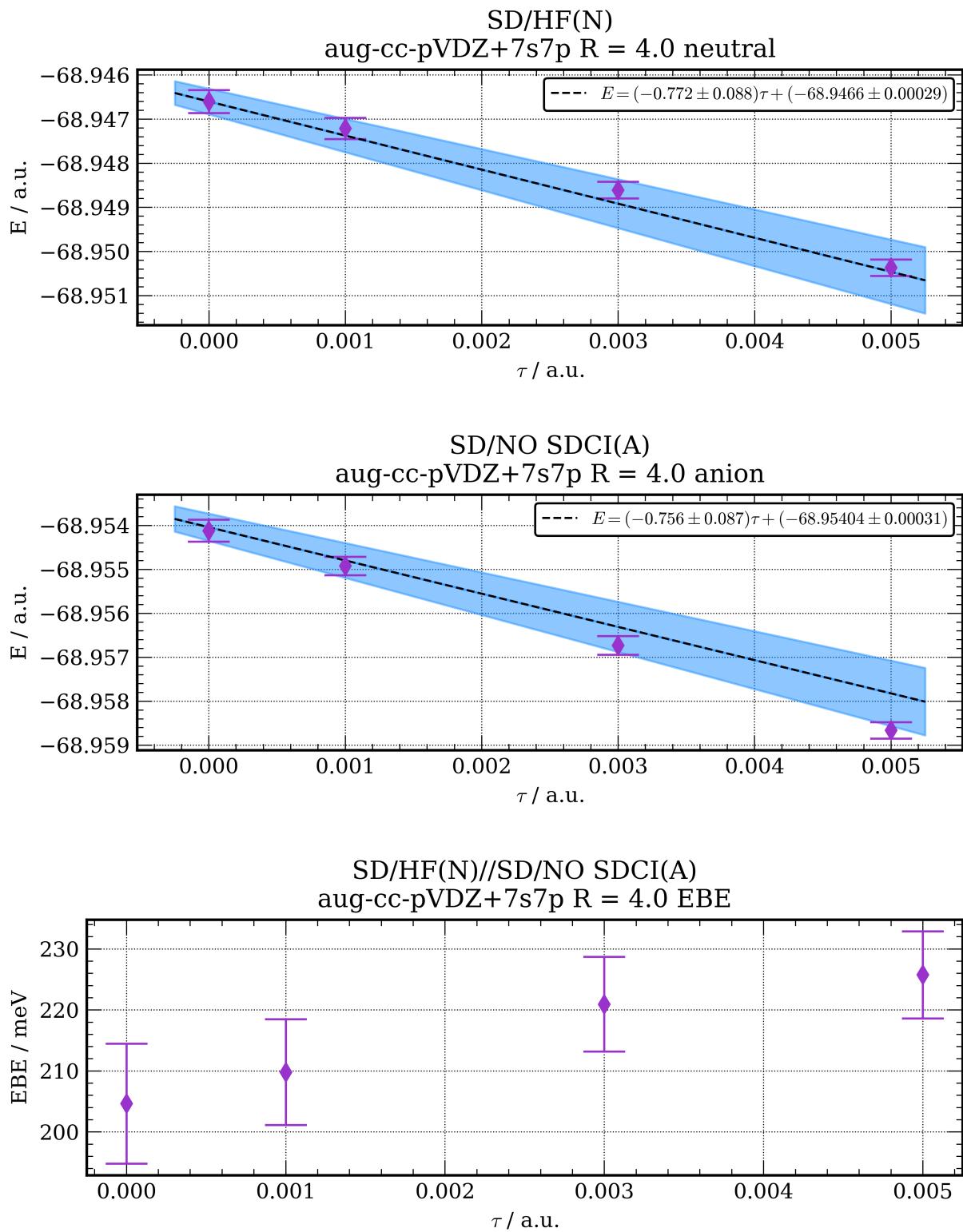


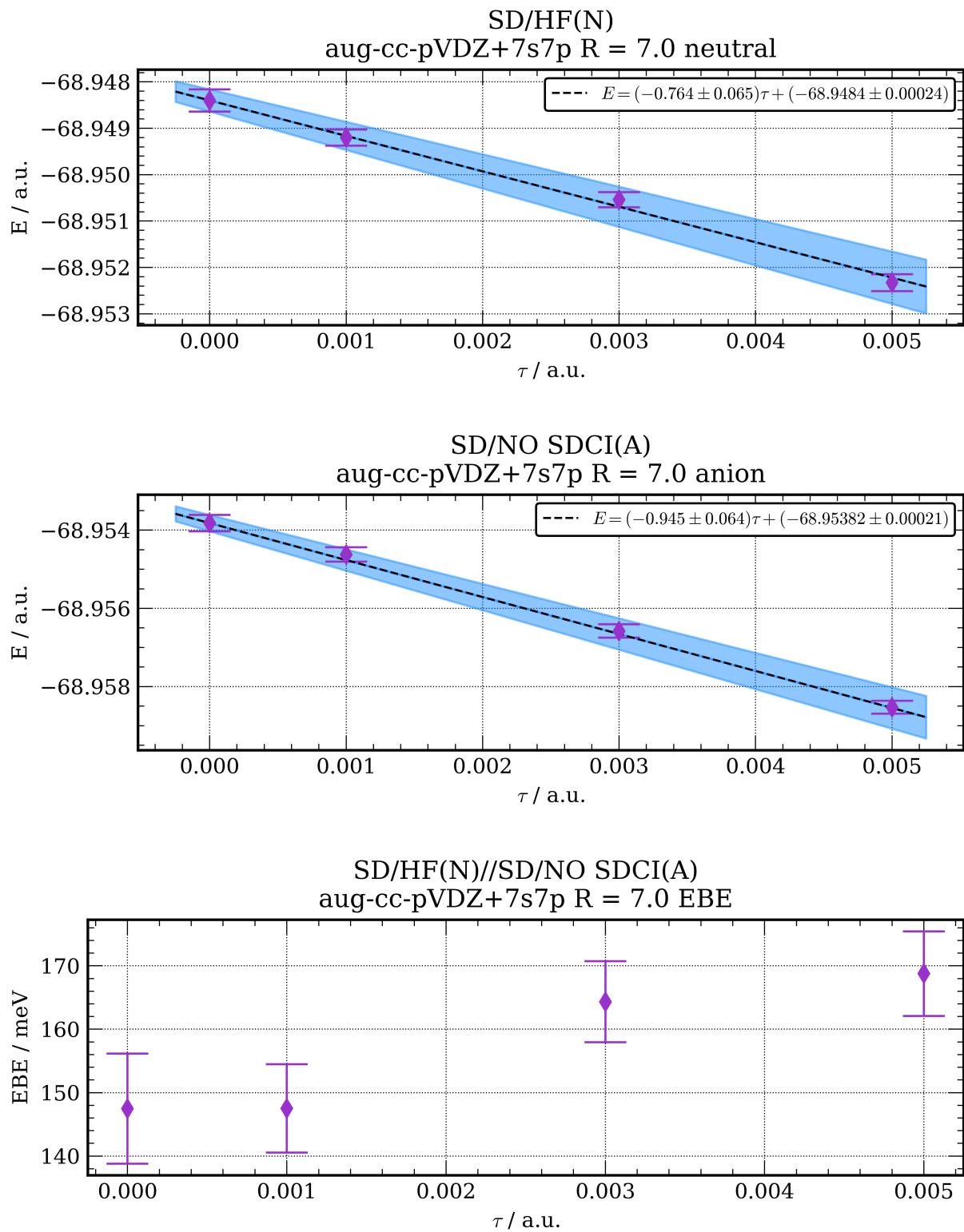












3 RADIAL ORBITAL DENSITY PLOTS

The radial orbital density, $\bar{\psi}^2(r)$, plots are created by integrating over angular portion of the norm of the single particle wavefunction.

$$\bar{\psi}^2(r) = \int_0^{2\pi} \int_0^\pi \psi^2(r, \theta, \phi) r^2 \sin(\theta) dr d\theta d\phi \quad (1)$$

Discretizing this expression using a uniform radial grid and a Lebedev-Laikov quadrature for the angular components, yields a form that can be readily evaluated.

$$\bar{\psi}^2(r_i) = 4\pi r_i^2 \sum_j^{N^{ang}} w_j^{ang} \psi^2(r_i, \theta_j, \phi_j) \quad (2)$$

The function $\bar{\psi}^2(r_i)$ from Equation ?? can be plotted with the points r_i serving as the abscissa. Since the singly occupied orbitals are normalized, the proximity of the sum of the radial quadrature to unity is used as a check.

$$\sum_i^{N^{rad}} \bar{\psi}^2(r_i) w_i^{rad} = \sum_i^{N^{rad}} \bar{\psi}^2(r_i) \Delta r \approx 1 \quad (3)$$

3.a Dependency versions

dependency	version
numpy	1.18.4
quadpy	0.16.2
pyscf	1.7.0
cclib	1.6.3

3.b Step 1: Generating a Molden file

Molden files were generated using `cclib`, with the exception of the natural orbital from the CIPSI calculations. Since QuantumPackage is not supported by `cclib`, Molden files were created using the native utility in QuantumPackage 2.0. For the Molden files generated with

cclib, the `-g/--ghost` flag indicates the presence of a ghost atom. The `-n/--naturalorbitals` was created to allow natural orbitals to be written instead of molecular orbitals. This flag is not yet available in the official distribution, but a request to incorporate it in the official distribution has been opened (<https://github.com/cclib/cclib/pull/948>).

3.c Step 2: Integrating over the angular components of the singly occupied orbital

`quadpy` was used to generate the Lebedev-Laikov integration weights and points. The singly occupied molecular/natural orbital was evaluated at these points using PySCF.