

RESEARCH ARTICLE

Adsorption-Calibrated Double Factorization with λ -Metrics for Qubitization-Oriented Hamiltonians

Miguel Angel Vargas Cruz^{1,*} ¹Grupo Alianza Empresarial, Mexico City, Mexico

Abstract: Low-rank factorizations of the two-electron repulsion integral (ERI) tensor are central to reducing fault-tolerant costs of Hamiltonian simulation in quantum chemistry, particularly under qubitization/linear-combination-of-unitaries (LCU) constructions where both the block-encoding normalization constant λ and the complexity of Prepare/Select depend on the chosen decomposition. We present a CPU-only, FCIDUMP-first workflow that (i) constructs a spectral double factorization (DF) of the ERI supermatrix $G_{(pq,rs)} = (pq|rs)$, (ii) truncates the factorization at rank r and reports reconstruction residuals, (iii) evaluates DF resource proxies via λ_{LCU}^{DF} and λ_{Burg}^{DF} , and (iv) selects an operational rank by enforcing an observable-level constraint on a differential quantity: the MP2 correlation contribution to a model adsorption energy, ΔE_{ads}^{corr} . Using NH_3 as a minimal Lewis-basic site and CO_2 as adsorbate in STO-3G, we sweep intermolecular separations $d_N \cdots C \in 2.2, 2.6, 3.0 \text{ \AA}$ and determine the worst-case feasible rank r_{worst} as the smallest rank satisfying a predefined adsorption-error tolerance across the geometry set. In this toy benchmark, λ -metrics for the complex saturate at moderate ranks while the differential adsorption observable remains sensitive to intermediate-rank components, demonstrating that λ convergence alone can be insufficient for rank selection when targeting chemically relevant differences. The resulting protocol provides a reproducible bridge between DF-based resource estimation and observable-constrained truncation criteria for qubitization-oriented Hamiltonians.

Keywords: double factorization (DF), hamiltonian simulation, qubitization, quantum algorithms for chemistry, λ -metrics

1. Introduction

Fault-tolerant quantum algorithms for electronic structure combine quantum error correction with coherent Hamiltonian simulation so that logical operations can be executed at accuracies far below the physical error rates of the underlying hardware. The historical algorithmic path runs from phase estimation and early Hamiltonian-simulation formulations [1], through product-formula and truncated-Taylor approaches, to quantum signal processing and qubitization, where optimal query scaling can be achieved for block-encoded Hamiltonians [2, 3]. In electronic-structure applications, these algorithms are attractive because strongly correlated regimes can be represented without the exponential determinant-space bottlenecks of classical full configuration interaction; however, the practical fault-tolerant resource estimates are still dominated by large constants associated with Hamiltonian normalization, data loading, basis choice, and logical circuit synthesis [4, 5].

In qubitization-based simulation, the query complexity of phase-estimation-style energy readout is approximately linear in the block-encoding normalization constant λ associated with the LCU representation of $\hat{H}$, and the total logical gate count also depends on the implementation of the corresponding

Prepare/Select primitives [2–4]. Therefore, a reduction of either λ or the decomposition complexity can reduce fault-tolerant resources, provided that the chemical observable of interest is not degraded [5].

A central mechanism for reducing these costs is low-rank structure in the electron repulsion integral (ERI) tensor. In double factorization (DF), the ERI supermatrix $G_{(pq),(rs)} = (pq|rs)$ is diagonalized, and the two-body interaction is expressed as a sum of squares of one-body operators. This structure supports qubitization constructions whose resource proxies depend on the retained rank, the spectral weights, and the induced λ normalization [4, 5]. Regularized and compressed variants preserve the DF operator structure while improving numerical stability and resource trade-offs [6]; related compression and symmetry-shift strategies have also been proposed for LCU decompositions and electronic Hamiltonian norm reduction [7–11]. Conventional assessment is often based on tensor residuals, total-energy errors, or normalization proxies. In chemical applications, however, the relevant targets are frequently *differential* observables, including reaction barriers, adsorption energies, redox potentials, and binding free energies, where error cancelation between fragments and complexes can be decisive.

The objective of this study is to answer the operational question: how far can the electronic integrals of a site-adsorbate system be compressed without significantly perturbing the

*Corresponding author: Miguel Angel Vargas Cruz, Grupo Alianza Empresarial, Mexico. Email: miguelangel@grupoalianzaempresarial.com

adsorption correlation energy? A rank-selection procedure is therefore formalized in which DF truncation is constrained by the MP2 correlation contribution to an adsorption energy, $\Delta E_{\text{ads}}^{\text{corr}}$, computed for a site-adsorbate-complex triad [12, 13]. MP2 is used as a low-cost diagnostic rather than as a final accuracy model; its numerator is directly dependent on two-electron integrals, and its canonical-denominator form isolates the effect of ERI compression when orbital energies are held fixed. The methodological advantage is that the truncation rank is selected from an observable-level error criterion, while $\lambda_{\text{LCU}}^{\text{DF}}$, $\lambda_{\text{Burg}}^{\text{DF}}$, and QPE query proxies are reported simultaneously. This produces an auditable workflow linking chemical accuracy requirements to qubitization-oriented resource indicators. The pipeline is CPU-only, uses FCIDUMP as an explicit data interface, and is designed as a reproducible proof of concept rather than as a claim of general transferability to large, strongly correlated systems [14, 15].

2. Methods

2.1. Electronic hamiltonian and integral conventions

For a fixed nuclear geometry $\{\mathbf{R}_A\}$ and an orthonormal spatial-orbital basis $\{\phi_p\}_{p=1}^N$ (canonical restricted Hartree-Fock (RHF) molecular orbitals (MOs)), the second-quantized electronic Hamiltonian is taken as

$$\hat{H} = E_{\text{core}} + \sum_{pq} \sum_{\sigma \in \{\alpha, \beta\}} h_{pq} a_{p\sigma}^\dagger a_{q\sigma} + \frac{1}{2} \sum_{pqrs} \sum_{\sigma, \tau \in \{\alpha, \beta\}} (pq|rs) a_{p\sigma}^\dagger a_{q\tau}^\dagger a_{s\tau} a_{r\sigma}, \quad (1)$$

where E_{core} is the constant term (nuclear repulsion plus any frozen-core contribution), h_{pq} are one-electron MO integrals, and $(pq|rs)$ are two-electron repulsion integrals (ERIs) in chemists' notation. Standard RHF and post-Hartree-Fock (HF) definitions follow canonical references [12, 13].

2.2. FCIDUMP interchange format

All quantities $\{E_{\text{core}}, h_{pq}, (pq|rs)\}$ are imported from an FCIDUMP file, a de facto interchange format originating in determinant-based Full Configuration Interaction (FCI) implementations [14]. FCIDUMP files are generated directly from an RHF reference using PySCF tooling [15]. Psi4 can also export FCIDUMP (used optionally for cross-validation) [16].

2.3. Supermatrix representation of the ERI tensor

Define the composite index set $\mathcal{J} = \{(p, q): 1 \leq p, q \leq N\}$ and the reshaping map

$$G_{(p,q),(r,s)} := (pq|rs), \quad (p, q), (r, s) \in \mathcal{J}, \quad (2)$$

so that $G \in \mathbb{R}^{N^2 \times N^2}$. In an orthonormal MO basis, G is real symmetric and positive semidefinite. The eigen-decomposition is computed as

$$G = Q \text{diag}(d) Q^\top, \quad d_1 \geq d_2 \geq \dots \geq 0, \quad (3)$$

where $Q = [q_1, \dots, q_{N^2}]$ is orthogonal and $d \in \mathbb{R}_{\geq 0}^{N^2}$ are the eigenvalues. For a target rank r , define the spectral truncation

$$G^{(r)} := \sum_{t=1}^r d_t q_t q_t^\top, \quad (4)$$

and the relative Frobenius residual

$$\delta_G(r) := \frac{\|G - G^{(r)}\|_F}{\|G\|_F}, \quad \|G\|_F := \left(\sum_{pqrs} (pq|rs)^2 \right)^{1/2}. \quad (5)$$

This ‘‘double factorization’’ viewpoint aligns with low-rank simulation constructions for quantum chemistry and their regularized/compressed variants [2, 4–6]. When ranks are swept across systems with different N , r is clipped to $r_{\text{eff}} := \min(r, N^2)$, so that $G^{(r)} = G$ for $r \geq N^2$ up to numerical precision.

2.4. DF leaves as one-body operators

For each retained eigenpair (d_t, q_t) , reshape q_t into an $N \times N$ matrix

$$L_{pq}^{(t)} := (q_t)_{(p,q)}. \quad (6)$$

Then the ERI tensor admits the approximation

$$(pq|rs) \approx \sum_{t=1}^r d_t L_{pq}^{(t)} L_{rs}^{(t)}. \quad (7)$$

Diagonalize each (symmetrized) leaf matrix

$$L^{(t)} = U^{(t)} \Lambda^{(t)} (U^{(t)})^\top, \quad \Lambda^{(t)} = \text{diag}(\lambda_1^{(t)}, \dots, \lambda_N^{(t)}), \quad (8)$$

and define the rotated fermionic modes

$$b_{k\sigma}^{(t)\dagger} := \sum_{p=1}^N U_{pk}^{(t)} a_{p\sigma}^\dagger, \quad \hat{n}_{k\sigma}^{(t)} := b_{k\sigma}^{(t)\dagger} b_{k\sigma}^{(t)}. \quad (9)$$

The corresponding one-body operator (per leaf) is

$$\hat{V}_t := \sum_{k=1}^N \sum_{\sigma \in \{\alpha, \beta\}} \lambda_k^{(t)} \hat{n}_{k\sigma}^{(t)}. \quad (10)$$

Up to standard normal-ordering rearrangements (absorbed into a one-body correction), the two-body part can be represented as a weighted sum of squares,

$$\hat{H}_2 \approx \frac{1}{2} \sum_{t=1}^r d_t \hat{V}_t^2 + \hat{H}_{1,\text{corr}}, \quad (11)$$

which is precisely the structure exploited by block-encoding/LCU and qubitization-based Hamiltonian-simulation algorithms [2–6].

2.5. Modified one-body term

The modified one-electron matrix F used in DF-based λ normalizations is defined as

$$F_{pq} := h_{pq} - \frac{1}{2} \sum_{r=1}^N (pr|qr) + \sum_{r=1}^N (pq|rr), \quad (12)$$

and denote its eigenvalues by $\{\nu_p\}_{p=1}^N$, that is, $F = W \text{diag}(\nu) W^\top$. This separation follows the DF-based normalization analyses used in qubitization resource estimates [4–6].

2.6. λ -metrics and query proxy

For a given factorization, define per-leaf matrices

$$\begin{aligned} Z^{(t)} &:= g_t \lambda^{(t)} (\lambda^{(t)})^\top, \\ g_t &:= \text{sign}(d_t), \\ \lambda^{(t)} &:= (\lambda_1^{(t)}, \dots, \lambda_N^{(t)})^\top. \end{aligned} \quad (13)$$

Two DF-derived normalization proxies used in the DF/qubitization literature are reported:
Direct LCU normalization.

$$\lambda_{\text{LCU}}^{\text{DF}} := \sum_{p=1}^N |\nu_p| + \sum_{t=1}^r \left(\sum_{1 \leq k \leq \ell \leq N} |Z_{k\ell}^{(t)}| + \frac{1}{4} \sum_{k=1}^N |Z_{kk}^{(t)}| \right), \quad (14)$$

which is a coefficient-1-norm proxy aligned with LCU/block-encoding normalizations [2, 4, 6].

von Burg-style normalization.

Let $W^{(t)}$ be a matrix square root of $Z^{(t)}$ (computed via an eigen-decomposition, allowing complex values when $Z^{(t)}$ is indefinite):

$$\begin{aligned} Z^{(t)} &= U_Z^{(t)} \text{diag}(\zeta^{(t)}) (U_Z^{(t)})^\top, \\ W^{(t)} &:= U_Z^{(t)} \text{diag}(\sqrt{\zeta^{(t)}}) (U_Z^{(t)})^\top. \end{aligned} \quad (15)$$

Define

$$\lambda_{\text{Burg}}^{\text{DF}} := \sum_{p=1}^N |\nu_p| + \frac{1}{4} \sum_{t=1}^r \sum_{\ell=1}^N \left(\sum_{k=1}^N |W_{k\ell}^{(t)}| \right)^2, \quad (16)$$

corresponding to the DF grouping and normalization used in fault-tolerant catalysis resource analyses [5, 6]. For a target phase-estimation precision ε (Hartree), the QPE query proxy is evaluated as

$$N_{\text{queries}}(\lambda, \varepsilon) := \left\lceil \frac{\pi \lambda}{2 \varepsilon} \right\rceil, \quad (17)$$

where λ is substituted by either $\lambda_{\text{LCU}}^{\text{DF}}$ or $\lambda_{\text{Burg}}^{\text{DF}}$. The expression is not a full logical gate estimate; it isolates the leading qubitization/QPE dependence on the block-encoding normalization while holding precision allocation, synthesis, and error-correction overheads fixed across comparisons [1, 2, 4, 5].

2.7. MP2 diagnostic and adsorption-energy rank selection

To select a truncation rank r controlling a chemically meaningful observable, restricted MP2 correlation energies are used in canonical RHF orbitals [12, 13]. For occupied indices i, j and

virtual indices a, b , the closed-shell MP2 correlation energy is computed as

$$E_{\text{MP2}}^{\text{corr}} = \sum_{i,j \in \text{occ}} \sum_{a,b \in \text{vir}} \frac{(ia|jb) (2(ia|jb) - (ib|ja))}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}, \quad (18)$$

where $\{\varepsilon_p\}$ are RHF orbital energies. In the truncation study, the denominators $\{\varepsilon_p\}$ are held fixed at their RHF values, while the numerators use either exact ERIs or truncated ERIs derived from $G^{(r)}$ to isolate the impact of two-electron compression.

For a site (NH_3), adsorbate (CO_2), and complex ($\text{NH}_3 \dots \text{CO}_2$), define the correlation contribution to adsorption energy

$$\Delta E_{\text{ads}}^{\text{corr}} := E_{\text{MP2}}^{\text{corr}}(\text{NH}_3 \dots \text{CO}_2) - E_{\text{MP2}}^{\text{corr}}(\text{NH}_3) - E_{\text{MP2}}^{\text{corr}}(\text{CO}_2). \quad (19)$$

Let $\Delta E_{\text{ads}}^{\text{corr}}(r)$ denote the same quantity evaluated with rank- r truncated ERIs in each subsystem, and define the truncation-induced adsorption correlation error

$$\Delta_{\text{ads}}(r) := \Delta E_{\text{ads}}^{\text{corr}}(r) - \Delta E_{\text{ads}}^{\text{corr}}(r_{\text{max}}), \quad (20)$$

where r_{max} is the maximum effective rank for each subsystem (or the largest rank in the sweep that yields machine-precision reconstruction).

For each intermolecular separation d in a discrete set $\mathcal{D}$ and candidate-rank set $\mathcal{R}$, the distance-specific minimal feasible rank is defined as

$$r^*(d) := \min \{r \in \mathcal{R} : |\Delta_{\text{ads}}(r; d)| \leq \tau\}, \quad (21)$$

with threshold $\tau = 1$ mHa. A stricter feasibility threshold, $\tau_{\text{safe}} = 0.1$ mHa, is also imposed to avoid ranks that merely satisfy the loose threshold by a small numerical margin. If a rank satisfies τ but not τ_{safe} , it is treated as marginally feasible: it is reported as passing the coarse adsorption criterion, but it is not selected for the operational protocol unless no rank in $\mathcal{R}$ satisfies the safe threshold. This convention prevents the selected rank from being controlled by small non-monotonic variations in $\Delta_{\text{ads}}(r; d)$ or by fortuitous cancellation between fragment and complex errors.

The operational rank is selected via a worst-case criterion over $\mathcal{D}$,

$$r_{\text{worst}} := \max_{d \in \mathcal{D}} r^*(d), \quad (22)$$

and this r_{worst} is used as the fixed truncation setting for reporting DF-based λ metrics and query-count proxies for the complex Hamiltonian [5, 6].

2.8. Software

Integral generation, RHF, and FCIDUMP export are performed using PySCF [15]. Optional cross-validation of FCIDUMP export uses Psi4 [16]. The DF truncation, λ -metrics, and rank-sweep design are aligned with low-rank/qubitization resource-estimation drivers in the quantum simulation literature [2, 4–6]. Other references [17–25].

2.8.1. Code

```
#!/usr/bin/env python3
# -*-coding:utf-8 -*-
# run: python3 co2_adsorption_end_to_end.py --activate_top_level
# End-to-end, single-file workflow (CPU-only) for a toy CO2-capture proxy:
# -Site: NH3
# -Adsorbate: CO2
# -Complex: NH3...CO2 at multiple N...C distances
# -Basis:STO-3G (chosen to keep G=(n^2 x n^2) eigen-decompositions cheap)
# What the script does from scratch:
# 1) Runs RHF (PySCF) for CO2,NH3,and NH3...CO2 at dist in {2.2,2.6,3.0} Å.
# 2) Writes FCIDUMP files for each system (FCIDUMP.CO2,FCIDUMP.NH3,FCIDUMP.NH3_CO2_dX.XX) .
# 3) Reads those FCIDUMPs back and constructs:
#     -supermatrix G[(pq),(rs)]=(pq|rs) in chemist notation
#     -truncated reconstructions G_r via top-|eig| spectral truncation
# 4) Computes MP2 correlation energies (restricted,spatial orbitals) using exact denominators
#     epsilon and compares adsorption correlation energy:
#     delta E_ads,corr=E_corr(complex)-E_corr(site)-E_corr(CO2)
#     and its truncation-induced error.
# 5) For each distance,selects best_rank(dist) by policy:
#     -"safe" rank:smallest r with |err|<=0.1 mHa
#     -else fallback:smallest r with |err|<=1.0 mHa
# 6) Chooses worst_case_best_rank=max_dist best_rank(dist)
# 7) Computes lambda_LCU^DF and lambda_Burg^DF for the COMPLEX at each distance using
#     the same spectral leaves (X-DF-like) truncated at r=worst_case_best_rank.
# 8) Writes a top-level active FCIDUMP for the worst-case geometry,keeping a backup:
#     -If ./FCIDUMP exists and ./FCIDUMP.ref does not,copy ./FCIDUMP->./FCIDUMP.ref
#     -Then copy the generated complex FCIDUMP (worst-case distance)->./FCIDUMP
# Defaults are set to the values we already validated:
# basis=sto-3g,dists=[2.2,2.6,3.0],ranks=[20,40,60,80,100,120],
# thresholds:thr_mHa=1.0,safe_mHa=0.1
# Expected outcome (based on our runs): worst_case_best_rank=100
# Requirements:
# -numpy
# -pyscf
# Notes:
# -The G eigen-decomposition scales as O(n^6) due to (n^2 x n^2) diagonalization.
# This script is strictly for small/medium active spaces/demos/evidence packages.
from __future__ import annotations
import argparse,math,re,shutil
from dataclasses import dataclass
from pathlib import Path
from typing import Dict,List,Tuple
import numpy as np
from pyscf import gto,scf
from pyscf.tools import fcidump
# -----
# FCIDUMP I/O (minimal parser)
# -----
@dataclass(frozen=True)
class FCIDumpData:
    norb:int
    nelec:int
    ms2:int
    ecore:float
    h1:np.ndarray # (n,n)
    eri:np.ndarray # (n,n,n,n) chemist (pq|rs),real,symmetrized
_FCIDUMP_HEADER_RE=re.compile(r"(\w+)\s*=\s*(\s*\w+\.|.+)"")
def parse_fcidump(path:str)->FCIDumpData:
    with open(path,"r",encoding="utf-8") as f:
        lines=f.readlines()
        header_lines=[]
```

```

body_start=0
for idx,line in enumerate(lines):
    header_lines.append(line)
    if "/" in line:
        body_start=idx+1
        break
header="".join(header_lines)
kv={m.group(1).upper():m.group(2) for m in _FCIDUMP_HEADER_RE.finditer(header)}
norb=int(kv.get("NORB","0"))
nelec=int(kv.get("NELEC","0"))
ms2=int(kv.get("MS2","0"))
if norb<=0:
    raise ValueError("FCIDUMP header missing/invalid NORB")
h1=np.zeros((norb,norb),dtype=float)
eri=np.zeros((norb,norb,norb,norb),dtype=float)
ecore=0.0
def set_eri_sym(p:int,q:int,r:int,s:int,v:float)->None:
    # enforce 8-fold+bra/ket symmetry for chemist integrals
    idxs=[
        (p,q,r,s),(q,p,r,s),(p,q,s,r),(q,p,s,r),
        (r,s,p,q),(s,r,p,q),(r,s,q,p),(s,r,q,p),
    ]
    for a,b,c,d in idxs:
        eri[a,b,c,d]=v
for line in lines[body_start:]:
    line=line.strip()
    if not line:
        continue
    parts=line.split()
    if len(parts)< 5:
        continue
    v=float(parts[0])
    i,j,k,l=(int(parts[1]),int(parts[2]),int(parts[3]),int(parts[4]))
    if i==j==k==l==0:
        ecore=v
        continue
    if k==0 and l==0:
        if i<=0 or j<=0:
            continue
        p,q=i-1,j-1
        h1[p,q]=v
        h1[q,p]=v
    else:
        if min(i,j,k,l)<=0:
            continue
        p,q,r,s=i-1,j-1,k-1,l-1
        set_eri_sym(p,q,r,s,v)
return FCIDumpData(norb=norb,nelec=nelec,ms2=ms2,ecore=ecore,h1=h1,eri=eri)
# -----
# Geometry builders (Å)
# -----
def geom_co2(co:float=1.16)->str:
    # linear CO2 along z
    return f"""
O 0.0 0.0 {-co}
C 0.0 0.0 0.0
O 0.0 0.0 {co}
""".strip()
def geom_nh3(nh:float=1.012,angle_deg:float=107.0)->str:
    # simple pyramidal NH3 (not optimized);N at origin;three H around z-axis
    ang=math.radians(angle_deg)
    r=nh*math.sin(ang)
    z=nh*math.cos(ang)

```

```

        return f"""
N 0.0 0.0 0.0
H {r:.8f} 0.0 {z:.8f}
H {-0.5*r:.8f} {math.sqrt(3)/2*r:.8f} {z:.8f}
H {-0.5*r:.8f} {-math.sqrt(3)/2*r:.8f} {z:.8f}
""".strip()
def geom_nh3_co2(dist_NC:float,co:float=1.16)->str:
    # place CO2 along x;carbon at (dist_NC,0,0),oxygen at+/-co along x
    Cx=dist_NC
    Ox1=Cx-co
    Ox2=Cx+co
    return "\n".join([
        geom_nh3(),
        f"C {Cx:.8f} 0.0 0.0",
        f"O {Ox1:.8f} 0.0 0.0",
        &f"O {Ox2:.8f} 0.0 0.0",
    ])
# -----
# Supermatrix, rank truncation (safe clipping),MP2 correlation energy
# -----
def supermatrix_from_eri(eri:np.ndarray)->np.ndarray:
    n=eri.shape[0]
    G=eri.reshape((n*n,n*n))
    return 0.5*(G+G.T)
def build_rank_truncated_G_series(G:np.ndarray,ranks:List[int]) ->Tuple[np.ndarray,
np.ndarray,List[np.ndarray],List[float],List[int]]:
    #Returns spectral truncations at requested ranks,with per-system clipping:
    #   r_eff=min(r,n2) where n2=dim(G).
    #Output:
    #   d_sorted,Q_sorted,
    #   Ghats aligned with ranks_sorted,
    #   deltaG aligned with ranks_sorted,
    #   ranks_eff aligned with ranks_sorted.
    n2=G.shape[0]
    ranks_sorted=sorted(set(int(r) for r in ranks))
    ranks_eff=[min(r,n2) for r in ranks_sorted]
    d,Q=np.linalg.eigh(G) # ascending
    idx=np.argsort(np.abs(d))[::-1]
    d=d[idx]
    Q=Q[:,idx]
    max_eff=max(ranks_eff)
    Ghat=np.zeros((n2,n2),dtype=float)
    Ghats_eff:Dict[int,np.ndarray]={0:Ghat.copy()}
    cur_r=0
    k=0
    need=set(ranks_eff)
    while cur_r< max_eff:
        qt=Q[:,k]
        Ghat+=float(d[k])*np.outer(qt,qt)
        cur_r+=1
        k+=1
        if cur_r in need:
            Ghats_eff[cur_r]=Ghat.copy()
    Gnorm=np.linalg.norm(G,ord="fro")
    Ghats_out:List[np.ndarray]=[]
    delta_out:List[float]=[]
    for r_eff in ranks_eff:
        Gr=Ghats_eff[r_eff]
        Ghats_out.append(Gr)
        delta_out.append(float(np.linalg.norm(G-Gr,ord="fro")/Gnorm))
    return d,Q,Ghats_out,delta_out,ranks_eff
def mp2_corr_energy(eri:np.ndarray,eps:np.ndarray,nelec:int)->float:
    # Restricted MP2 correlation energy in spatial orbitals (chemist notation).

```

```

n=eri.shape[0]
nocc=nelec // 2
occ=range(nocc)
vir=range(nocc,n)
emp2=0.0
for i in occ:
    ei=eps[i]
    for j in occ:
        ej=eps[j]
        for a in vir:
            ea=eps[a]
            for b in vir:
                eb=eps[b]
                denom=ei+ej-ea-eb
                iajb=eri[i,a,j,b]      # (ia|jb)
                ibja=eri[i,b,j,a]      # (ib|ja)
                emp2+=iajb*(2.0*iajb-ibja)/denom
    return float(emp2)
# -----
# lambda metrics (X-DF-like spectral leaves)
# -----
def modified_one_body_F(h1:np.ndarray,eri:np.ndarray)->np.ndarray:
    # F_pq:=h_pq-1/2 sum_r (p r | q r)+sum_r (p q | r r)
    n=h1.shape[0]
    J=np.zeros((n,n),dtype=float)
    K=np.zeros((n,n),dtype=float)
    for r in range(n):
        J+=eri[:, :, r, r]
        K+=eri[:, r, :, r]
    F=h1-0.5*K+J
    return 0.5*(F+F.T)
def lambda_LCU_DF(F_eigs:np.ndarray,Z:np.ndarray)->float:
    lam=float(np.sum(np.abs(F_eigs)))
    nt,n,_=Z.shape
    for t in range(nt):
        Zt=Z[t]
        lam+=float(np.triu(np.abs(Zt),k=1).sum()+0.25*np.abs(np.diag(Zt)).sum())
    return lam
def lambda_Burg_DF(F_eigs:np.ndarray,Z:np.ndarray)->float:
    lam=float(np.sum(np.abs(F_eigs)))
    acc=0.0
    for t in range(Z.shape[0]):
        w,U=np.linalg.eigh(Z[t])
        sqrtw=np.sqrt(w.astype(complex))
        W=(U.astype(complex)*sqrtw[np.newaxis,:]) @ U.T.astype(complex)
        col_sums=np.sum(np.abs(W),axis=0)
        ;acc+=float(np.sum(col_sums**2))
    return lam+0.25*acc
def lambdas_for_complex_at_rank(data:FCIDumpData,d_sorted:np.ndarray,Q_sorted:np.
ndarray,r_eff:int)
->Tuple[float,float]:
    # Build Z^t from the first r_eff spectral leaves of G and compute lambda_LCU,
lambda_Burg.
    n=data.norb
    F=modified_one_body_F(data.h1,data.eri)
    F_eigs=np.linalg.eigvalsh(F)
    nt=r_eff
    Z=np.zeros((nt,n,n),dtype=float)
    for t in range(nt):
        dt=float(d_sorted[t])
        gt=1.0 if dt>=0.0 else -1.0
        Vt=math.sqrt(abs(dt))*Q_sorted[:,t].reshape((n,n))
        Vt=0.5*(Vt+Vt.T)

```

```

        lam,_=np.linalg.eigh(Vt)
        Z[t]=gt*np.outer(lam,lam)
        return lambda_LCU_DF(F_eigs,Z),lambda_Burg_DF(F_eigs,Z)
def qpe_queries(lambda_val:float,eps_mHa:float)->int:
    # Simple scaling proxy:queries ~ ceil((pi/2)*lambda/eps),with eps in Eh.
    eps_Eh=eps_mHa*1e-3
    return int(math.ceil((math.pi/2.0)*lambda_val/eps_Eh))
# -----
# PySCF driver:RHF+FCIDUMP
# -----
@dataclass
class Sys:
    name:str
    fcidump_path:Path
    hf_etot:float
    eps:np.ndarray
    data:FCIDumpData
def run_rhf_write_fcidump(name:str,atom_xyz:str,basis:str,outdir:Path)->Sys:
    mol=gto.M(atom=atom_xyz,basis=basis,unit="Angstrom",charge=0,spin=0,verbose=0)
    mf=scf.RHF(mol)
    mf.kernel()
    if not mf.converged:
        raise RuntimeError(f"SCF did not converge for {name}")
    outdir.mkdir(parents=True,exist_ok=True)
    fpath=outdir/f"FCIDUMP.{name}"
    fcidump.from_scf(mf,str(fpath),tol=1e-12)
    data=parse_fcidump(str(fpath))
    eps=np.array(mf.mo_energy,dtype=float)
    return Sys(name=name,fcidump_path=fpath,hf_etot=float(mf.e_tot),eps=eps,data=data)
# -----
# Rank selection policy for adsorption detla E (corr)
# -----
def choose_best_rank_from_errors(ranks_sorted:List[int],err_mHa: List[float],thr_mHa:float,
safe_mHa:float)->int:
    # "safe": smallest r with |err|<=safe_mHa; else fallback to threshold thr_mHa
    safe_candidates=[r for r,e in zip(ranks_sorted,err_mHa) if abs(e)<=safe_mHa]
    if safe_candidates:
        return min(safe_candidates)
    thr_candidates=[r for r,e in zip(ranks_sorted,err_mHa) if abs(e)<=thr_mHa]
    if not thr_candidates:
        raise RuntimeError("No rank satisfies the threshold.")
    return min(thr_candidates)
# -----
# FCIDUMP backup/activate
# -----
def preserve_and_activate_top_level_fcidump(source:Path,active_name: str="FCIDUMP",
ref_name:str="FCIDUMP.ref")->None:
    active=Path(active_name).resolve()
    ref=Path(ref_name).resolve()
    if active.exists() and not ref.exists():
        shutil.copy2(active,ref)
        print(f"[ok] Preserved reference:{ref}")
    shutil.copy2(source,active)
    print(f"[ok] Activated:{active}<-{source.resolve()}")
# -----
# Main workflow
# -----
def main()->None:
    ap=argparse.ArgumentParser()
    ap.add_argument("--basis",default="sto-3g")
    ap.add_argument("--dists",default="2.2,2.6,3.0")
    ap.add_argument("--ranks",default="20,40,60,80,100,120")
    ap.add_argument("--thr_mHa",type=float,default=1.0)

```

```

ap.add_argument("--safe_mHa",type=float,default=0.1)
ap.add_argument("--eps_mHa",type=float,default=1.0)
ap.add_argument("--outdir",default="fcidumps_end_to_end")
ap.add_argument("--activate_top_level",action="store_true",help="Copy worst-case
complex FCIDUMP into ./FCIDUMP (backup ./FCIDUMP.ref).")
args=ap.parse_args()
basis=args.basis
dists=[float(x) for x in args.dists.split(",") if x.strip()]
ranks=[int(x) for x in args.ranks.split(",") if x.strip()]
ranks_sorted=sorted(set(ranks))
outdir=Path(args.outdir)
# Fixed fragments (computed once)
sys_co2=run_rhf_write_fcidump("CO2",geom_co2(),basis,outdir)
sys_site=run_rhf_write_fcidump("NH3",geom_nh3(),basis,outdir)
# Precompute exact MP2 correlation for fragments (used in adsorption)
emp2_co2_ref=mp2_corr_energy(sys_co2.data.eri,sys_co2.eps,sys_co2.data.nelec)
emp2_site_ref=mp2_corr_energy(sys_site.data.eri,sys_site.eps,sys_site.data.nelec)
# For fragments,build rank series once (reuse across distances)
G_co2=supermatrix_from_eri(sys_co2.data.eri)
dC,QC,GhatC_list,_,ranks_eff_C=build_rank_truncated_G_series(G_co2,ranks_sorted)
eriC_list=[Ghat.reshape((sys_co2.data.norb,)*4) for Ghat in GhatC_list]
G_site=supermatrix_from_eri(sys_site.data.eri)
dS,QS,GhatS_list,_,ranks_eff_S=build_rank_truncated_G_series(G_site,ranks_sorted)
eriS_list=[Ghat.reshape((sys_site.data.norb,)*4) for Ghat in GhatS_list]
# For each distance:compute complex,adsorption error vs rank,best_rank,and lambda
at final chosen rank
best_rank_per_dist:Dict[float,int]={}
adsorption_tables:Dict[float,List[Tuple[int,float,float]]]={} # dist->[(r,deltaG_complex,
err_mHa)]
complex_systems:Dict[float,Sys]={}
print(f"# basis={basis}")
print(f"# ranks={ranks_sorted} (thr_mHa={args.thr_mHa},safe_mHa={args.safe_mHa})")
print(f"# fragments:CO2 (NORB={sys_co2.data.norb},NELEC={sys_co2.data.nelec})
NH3 (NORB={sys_site.data.norb},NELEC={sys_site.data.nelec})")
for dist in dists:
    name=f"NH3_CO2_d{dist:.1f}"
    sys_cplx=run_rhf_write_fcidump(name,geom_nh3_co2(dist_NC=dist),basis,outdir)
    complex_systems[dist]=sys_cplx
    emp2_cplx_ref=mp2_corr_energy(sys_cplx.data.eri,sys_cplx.eps,sys_cplx.data.nelec)
    dEads_corr_ref=emp2_cplx_ref-emp2_site_ref-emp2_co2_ref
    G_cplx=supermatrix_from_eri(sys_cplx.data.eri)
    dX,QX,GhatX_list,deltaX_list,ranks_eff_X=build_rank_truncated_G_series(G_cplx,
ranks_sorted)
    eriX_list=[Ghat.reshape((sys_cplx.data.norb,)*4) for Ghat in GhatX_list]
    # Evaluate adsorption correlation error vs rank
    err_mHa_list:List[float]=[]
    rows:List[Tuple[int,float,float]]=[]
    for r_req,dGx,eriX,eriS,eriC in zip(ranks_sorted,deltaX_list,
eriX_list,eriS_list,eriC_list):
        emp2_X=mp2_corr_energy(eriX,sys_cplx.eps,sys_cplx.data.nelec)
        emp2_S=mp2_corr_energy(eriS,sys_site.eps,sys_site.data.nelec)
        emp2_C=mp2_corr_energy(eriC,sys_co2.eps,sys_co2.data.nelec)
        dEads_corr_hat=emp2_X-emp2_S-emp2_C
        err_Eh=dEads_corr_hat-dEads_corr_ref
        err_mHa=err_Eh*1e3
        err_mHa_list.append(err_mHa)
        rows.append((r_req,dGx,err_mHa))
    adsorption_tables[dist]=rows
    best_rank=choose_best_rank_from_errors(ranks_sorted,err_mHa_list,args.thr_mHa,
args.safe_mHa)
    best_rank_per_dist[dist]=best_rank
    print(f"\n# dist_NC={dist:.3f} Å    complex NORB={sys_cplx.data.norb}
NELEC={sys_cplx.data.nelec}    best_rank={best_rank}")

```

```

    print("r,deltaG_complex,dEads_corr_err_mHa")
    for r_req,dGx,err_mHa in rows:
        print(f"{r_req},{dGx:.3e},{err_mHa:.6f}")
    worst_case_best_rank=max(best_rank_per_dist.values())
    # Pick a representative worst-case distance (smallest distance among those reaching
worst_case_best_rank)
    worst_dists=[d for d,r in best_rank_per_dist.items() if r==worst_case_best_rank]
    worst_dist=min(worst_dists)
    print("\n# best_rank_per_dist:")
    for dist in sorted(best_rank_per_dist.keys()):
        print(f"# {dist:.3f} Å->{best_rank_per_dist[dist]}")
    print(f"# worst_case_best_rank={worst_case_best_rank} (selected worst_dist={worst_
dist:.3f} Å)")
    # Compute lambda and query proxies for complex at each distance using worst_
case_best_rank
    print("\n# Resource proxies for the COMPLEX at each distance using worst_
case_best_rank")
    print("dist_NC,NORB,NELEC,rank_used,lambda_LCU,lambda_Burg,queries_LCU(eps=1mHa) ,
queries_Burg(eps=1mHa)")
    for dist in sorted(dists):
        sys_cplx=complex_systems[dist]
        G_cplx=supermatrix_from_eri(sys_cplx.data.eri)
        dX,QX,_,_,_=build_rank_truncated_G_series(G_cplx,[worst_case_best_rank])
        # rank_used is clipped to n^2~inside build_rank_truncated_G_series;reconstruct
it here:
        rank_used=min(worst_case_best_rank,sys_cplx.data.norb**2)
        lam_lcu,lam_burg=lambdas_for_complex_at_rank(sys_cplx.data,dX,QX,rank_used)
        q_lcu=qpe_queries(lam_lcu,args.eps_mHa)
        q_burg=qpe_queries(lam_burg,args.eps_mHa)
        print(f"{dist:.3f},{sys_cplx.data.norb},{sys_cplx.data.nelec},{rank_used},"
            f"{lam_lcu:.15f},{lam_burg:.15f},{q_lcu},{q_burg}")
    # Optionally activate top-level ./FCIDUMP from the worst-case complex geometry
    if args.activate_top_level:
        src=complex_systems[worst_dist].fcidump_path
        preserve_and_activate_top_level_fcidump(src)
    # Write a compact summary file
    summary=outdir/"summary.txt"
    with open(summary,"w",encoding="utf-8") as f:
        f.write(f"basis={basis}\n")
        f.write(f"dists={dists}\n")
        f.write(f"ranks={ranks_sorted}\n")
        f.write(f"thr_mHa={args.thr_mHa}\n")
        f.write(f"safe_mHa={args.safe_mHa}\n")
        f.write(f"worst_case_best_rank={worst_case_best_rank}\n")
        f.write(f"worst_dist={worst_dist}\n")
        for dist in sorted(best_rank_per_dist.keys()):
            f.write(f"best_rank[{dist:.3f}]= {best_rank_per_dist[dist]}\n")
    print(f"\n[ok] Wrote:{summary.resolve()}")
if __name__=="__main__":
    main()

```

3. Results

3.1. Rank-dependent adsorption correlation error and feasibility

CO₂, NH₃, and the noncovalent complex NH₃ ...CO₂ were evaluated at fixed separations $d_{N...C} \in \{2.2, 2.6, 3.0\}$ Å in the STO-3G basis using an RHF reference and FCIDUMP export through PySCF [14, 15]. For each subsystem, the ERI supermatrix G was diagonalized and truncated at ranks $r \in \mathcal{R} = \{20, 40, 60, 80, 100, 120\}$ to form $G^{(r)}$, with per-system clipping $r_{\text{eff}} = \min(r, N^2)$ when necessary. The truncation impact was assessed on the adsorption correlation energy (Eq. 19) and on the rank-induced error $\Delta_{\text{ads}}(r; d)$ defined relative to the maximal effective-rank reference for each subsystem.

The physical interpretation of this diagnostic is direct: the compressed ERI tensor modifies the pair-correlation amplitudes entering the MP2 numerator, whereas the RHF orbital-energy denominators are held fixed. Therefore, changes in $\Delta E_{\text{ads}}^{\text{corr}}$ isolate how DF truncation perturbs dispersion-like and induction-assisted correlation contributions to the site-adsorbate interaction. The shortest separation, $d_{N...C} = 2.2$ Å, is expected to be most sensitive because greater orbital overlap and stronger electrostatic polarization increase the importance of intermediate-rank ERI components. At larger separations, correlation contributions are weaker, and the adsorption observable is less demanding under the same absolute mHa criterion.

To connect rank selection to both (i) an observable-level constraint and (ii) simulation-oriented normalization behavior,

Figure 1 reports (left) the absolute adsorption correlation-energy truncation error $|\Delta_{\text{ads}}(r; d)|$ and (right) the DF normalization proxy $\lambda_{\text{Burg}}^{\text{DF}}(r)$ for the complex Hamiltonian as functions of the spectral DF rank r for a representative separation ($d_{N...C} = 2.6$ Å). At this separation, the adsorption correlation error decreases rapidly with rank: the error is $\mathcal{O}(10)$ mHa at $r = 20$, remains several mHa at $r = 40$, crosses below 1 mHa at $r = 60$, and drops below 0.1 mHa at $r = 80$; for example, $|\Delta_{\text{ads}}(80; 2.6 \text{ Å})| = 1.4144 \times 10^{-2}$ mHa.

Application of the safe feasibility threshold $\tau_{\text{safe}} = 0.1$ mHa gives

$$r^*(2.2 \text{ Å}) = 100, \quad r^*(2.6 \text{ Å}) = 80, \quad r^*(3.0 \text{ Å}) = 80, \quad (23)$$

and therefore

$$r_{\text{worst}} = \max_{d \in \{2.2, 2.6, 3.0\} \text{ Å}} r^*(d) = 100. \quad (24)$$

The feasibility decision is summarized in Table 1. Ranks that satisfy 1mHa but not 0.1mHa are classified as marginal and are not used for the operational setting unless no safe rank exists in the sweep. This distinction is important because the absolute adsorption error is a difference of three separately truncated correlation energies; accidental cancellation can cause a rank to appear acceptable under a loose threshold while remaining unstable under a stricter one.

Figure 1

Left: Absolute adsorption correlation-energy truncation error, $|\Delta_{\text{ads}}(r; d)|$, versus spectral DF rank r for NH₃ ...CO₂ at fixed separation d (example shown: $d_{N...C} = 2.6$ Å). Errors are defined relative to the maximal effective-rank reference and plotted on a logarithmic scale. **Right:** DF normalization proxy $\lambda_{\text{Burg}}^{\text{DF}}$ versus r for the complex Hamiltonian at the same geometry. The juxtaposition shows that normalization saturation does not guarantee convergence of the differential chemical observable

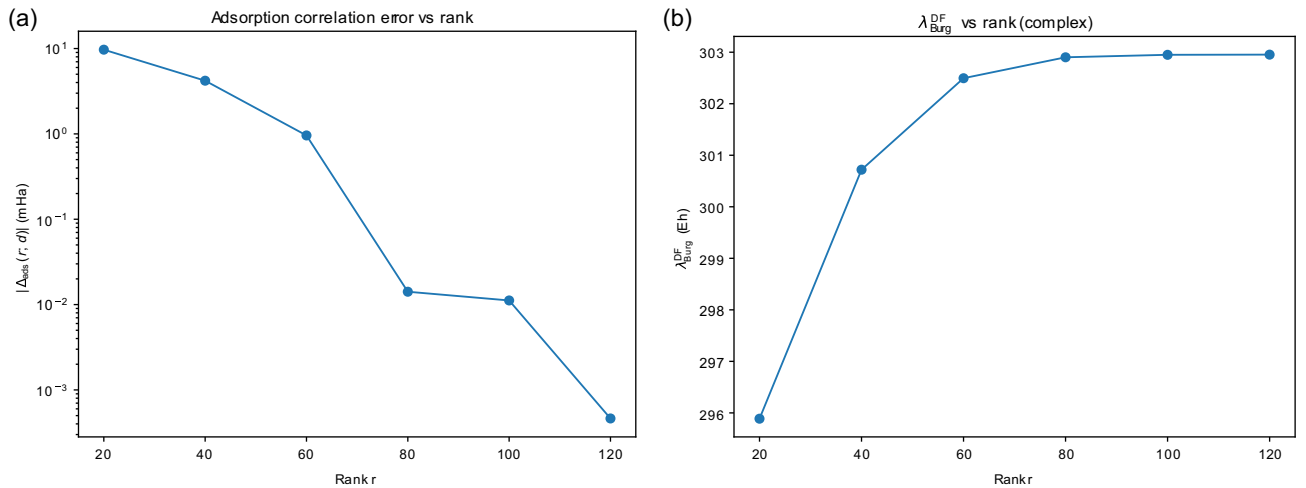


Table 1
Observable-constrained rank selection for the NH₃ ...CO₂ adsorption correlation diagnostic. The operational rank is the worst-case safe rank across separations

$d_{N...C} = 2.6$ Å	Safe threshold (mHa)	Selected $r^*(d)$	Interpretation
2.2	0.1	100	Strongest compression sensitivity
2.6	0.1	80	Intermediate separation
3.0	0.1	80	Weaker coupling regime
$r_{\text{worst}} = 100$			

3.2. Convergence of DF normalization metrics with rank

For each complex geometry, DF-based normalization proxies $\lambda_{\text{LCU}}^{\text{DF}}$ and $\lambda_{\text{Burg}}^{\text{DF}}$ were computed for the truncated factorization following the DF/qubitization resource-estimation framework [2, 4–6]. As shown in Figure 1 (right), $\lambda_{\text{Burg}}^{\text{DF}}$ saturates quickly with rank: increasing rank from $r = 60$ to $r = 120$ changes $\lambda_{\text{Burg}}^{\text{DF}}$ by only approximately 0.15% (302.49 to 302.95 in Hartree units at $d_{\text{N} \dots \text{C}} = 2.6 \text{ \AA}$), whereas Figure 1 (left) shows that the adsorption correlation error decreases from approximately 0.96 mHa to approximately 4.6×10^{-4} mHa over the same interval.

This separation of behavior has a clear numerical origin. The λ proxies are weighted coefficient norms dominated by the leading spectral components of G and by the modified one-body contribution. Once these dominant components have been retained, additional intermediate-rank factors can have only a small effect on λ while still altering specific ERI contractions entering the MP2 adsorption numerator. The observable is a difference of three correlated quantities, so residual errors in the complex and fragments can either cancel or reinforce. Rank selection based only on λ would therefore be blind to part of the chemically relevant truncation error.

3.3. Implication for QPE query-count scaling proxies

Using the explicit qubitization/QPE proxy (Eq. 17) with fixed target precision ε , the near saturation of λ implies that increasing the rank beyond the observable-selected regime changes the leading query count only weakly [1, 2, 4]. This quantity should not be interpreted as a hardware-level resource estimate because it excludes state-preparation synthesis, arithmetic precision, error-correction overhead, layout constraints, and the cost of implementing the DF rotations. The dominant practical advantage of reducing rank is therefore expected to arise from the reduced complexity of Prepare/Select and associated orbital-rotation data, rather than from a large decrease in the leading λ/ε query factor alone.

3.4. Relation to alternative compression strategies

The present spectral DF protocol is not the only possible compression route for electronic Hamiltonians. Cholesky decomposition provides a deterministic integral-factorization framework with a threshold directly tied to ERI residuals; tensor hypercontraction introduces lower-dimensional auxiliary structures that can further reduce storage and circuit-loading costs; regularized compressed DF modifies the DF factors to improve λ and numerical conditioning; and symmetry-shift methods reduce LCU norms by exploiting block-invariant transformations [4, 6–11]. The contribution of the present workflow is orthogonal to these alternatives: the compression method can be replaced, but the rank or threshold should still be tested against the target differential observable. A systematic numerical comparison against Cholesky, tensor hypercontraction, and regularized DF is left as an extension because the present manuscript is restricted to a controlled proof-of-concept benchmark.

4. Discussion

The results show that an observable-level constraint based on a differential energy can impose a stricter and more chemically meaningful rank requirement than integral-norm residuals

or λ convergence alone. In the $\text{NH}_3 \dots \text{CO}_2$ sweep, $\lambda_{\text{Burg}}^{\text{DF}}$ is effectively converged by $r \approx 80$ –100, yet the adsorption correlation error at $d = 2.2 \text{ \AA}$ requires $r = 100$ to satisfy the 0.1 mHa safe threshold. This behavior is consistent with the distinction between a decomposition normalization, which drives simulation query complexity, and an adsorption-energy error, which depends on cancellations between subsystem correlation energies [12, 13].

The pipeline separates three notions of accuracy: (i) tensor reconstruction accuracy in $\delta_G(r)$, (ii) simulation normalization through λ [2, 4, 5], and (iii) accuracy of the target chemical observable. The rank-selection rule enforces (iii) directly, while (ii) is monitored to estimate simulation-level scaling. The methodological advantage is that a decomposition that appears efficient under a Hamiltonian norm criterion is not accepted unless it also preserves the adsorption diagnostic. This is aligned with fault-tolerant resource-estimation studies in which algorithmic decompositions are assessed by induced normalization and implementation overheads, but it adds an explicit chemical-observable filter [5, 6].

The demonstration has deliberate limitations. STO-3G and fixed, non-optimized geometries were selected to keep CPU-only diagonalizations of $G \in \mathbb{R}^{N^2 \times N^2}$ tractable. The $\text{NH}_3 \dots \text{CO}_2$ complex is small and weakly interacting, and it does not test strong static correlation, large active spaces, basis-set convergence, geometry relaxation, or finite-temperature adsorption thermodynamics. MP2 is used strictly as a diagnostic proxy; it can be sensitive to basis-set quality, orbital gaps, and weak-interaction balance, and it should not be interpreted as a universally reliable adsorption model. Moreover, the reported QPE query proxy is not a full fault-tolerant gate-level estimate because Prepare/Select synthesis, rotation precision, logical compilation, and error-correction overheads are not included.

Several extensions follow directly. The MP2 diagnostic can be replaced by Coupled Cluster with Singles, Doubles, and perturbative Triples (CCSD(T)), Selected Configuration Interaction (CI), Density Matrix Renormalization Group (DMRG), Auxiliary-Field Quantum Monte Carlo (AFQMC), or experimentally calibrated active-space observables in systems where those references are available. The same selection logic can be applied to Cholesky, tensor hypercontraction, regularized DF, symmetry-compressed DF, or Block-Invariant Symmetry-Shift (BLISS)-shifted Hamiltonians. Larger basis sets and chemically optimized adsorption geometries should be used to test whether the separation between λ convergence and observable convergence persists under realistic chemical conditions. Finally, a complete fault-tolerant assessment should include explicit Prepare/Select circuits, precision allocation, data-loading costs, surface-code overheads, and hardware-connectivity assumptions, following catalysis and materials resource-estimation analyses [5, 19, 20, 22].

5. Conclusion

A CPU-only, FCIDUMP-driven workflow was introduced for selecting the rank of a spectral DF truncation by enforcing an observable-level accuracy constraint on a differential quantity: the MP2 correlation contribution to an adsorption energy [12–15]. The principal findings are as follows. First, in the $\text{NH}_3\text{--CO}_2$ model sweep at $d_{\text{N} \dots \text{C}} \in \{2.2, 2.6, 3.0\} \text{ \AA}$ in STO-3G, the worst-case safe rank under a 0.1 mHa threshold is $r_{\text{worst}} = 100$. Second, $\lambda_{\text{LCU}}^{\text{DF}}$ and $\lambda_{\text{Burg}}^{\text{DF}}$ are already near saturation at ranks where the adsorption correlation error can still

change appreciably. Third, the explicit QPE proxy $[(\pi/2)\lambda/\varepsilon]$ indicates that further rank increases mainly affect implementation complexity rather than the leading query factor once λ is saturated.

The present result should be interpreted as a proof of concept. It is limited by the minimal STO-3G basis, non-optimized geometries, a small weakly interacting complex, full spectral diagonalization, and the use of MP2 as a diagnostic rather than as a definitive adsorption-energy model. The approach nevertheless advances the resource-estimation workflow by coupling low-rank Hamiltonian compression to a target chemical observable instead of relying only on tensor residuals or Hamiltonian normalization. Future work should apply the same observable-constrained rank-selection principle to larger active spaces, optimized adsorption structures, higher-level correlated benchmarks, alternative factorizations, and complete logical-resource models for qubitization-based phase estimation.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by the author.

Conflicts of Interest

The author declares that he has no conflicts of interest to this work.

Data Availability Statement

Data are available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Miguel Angel Vargas Cruz: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

- [1] Kitaev, A. Y. (1995). *Quantum measurements and the Abelian Stabilizer Problem*. arXiv. <https://doi.org/10.48550/arXiv.quant-ph/9511026>
- [2] Low, G. H., & Chuang, I. L. (2019). Hamiltonian simulation by qubitization. *Quantum*, 3, 163. <https://doi.org/10.22331/q-2019-07-12-163>
- [3] Low, G. H., & Chuang, I. L. (2017). Optimal Hamiltonian simulation by quantum signal processing. *Physical review letters*, 118(1), 010501. <https://doi.org/10.1103/PhysRevLett.118.010501>
- [4] Berry, D. W., Gidney, C., Motta, M., McClean, J. R., & Babbush, R. (2019). Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization. *Quantum*, 3, 208. <https://doi.org/10.22331/q-2019-12-02-208>
- [5] von Burg, V., Low, G. H., Häner, T., Steiger, D. S., Reiher, M., Roetteler, M., & Troyer, M. (2021). Quantum computing enhanced computational catalysis. *Physical Review Research*, 3(3), 033055. <https://doi.org/10.1103/PhysRevResearch.3.033055>
- [6] Oumarou, O., Scheurer, M., Parrish, R. M., Hohenstein, E. G., & Gogolin, C. (2024). Accelerating quantum computations of chemistry through regularized compressed double factorization. *Quantum*, 8, 1371. <https://doi.org/10.22331/q-2024-06-13-1371>
- [7] Loaiza, I., Khah, A. M., Wiebe, N., & Izmaylov, A. F. (2023). Reducing molecular electronic Hamiltonian simulation cost for linear combination of unitaries approaches. *Quantum Science and Technology*, 8(3), 035019. <https://doi.org/10.1088/2058-9565/acd577>
- [8] Loaiza, I., & Izmaylov, A. F. (2023). Block-invariant symmetry shift: Preprocessing technique for second-quantized Hamiltonians to improve their decompositions to linear combination of unitaries. *Journal of Chemical Theory and Computation*, 19(22), 8201–8209. <https://doi.org/10.1021/acs.jctc.3c00912>
- [9] Patel, S., Brahmachari, A. S., Cantin, J. T., Wang, L., & Izmaylov, A. F. (2025). Global minimization of electronic Hamiltonian 1-norm via linear programming in the block invariant symmetry Shift (BLISS) method. *Journal of Chemical Theory and Computation*, 21(2), 703–713. <https://doi.org/10.1021/acs.jctc.4c01390>
- [10] Deka, K., & Zak, E. (2025). Simultaneously optimizing symmetry shifts and tensor factorizations for cost-efficient fault-tolerant quantum simulations of electronic Hamiltonians. *Journal of Chemical Theory and Computation*, 21(9), 4458–4465. <https://doi.org/10.1021/acs.jctc.4c01722>
- [11] Rocca, D., Cortes, C. L., Gonthier, J. F., Ollitrault, P. J., Parrish, R. M., Anselmetti, G. L., ..., & Streif, M. (2024). Reducing the runtime of fault-tolerant quantum simulations in chemistry through symmetry-compressed double factorization. *Journal of Chemical Theory and Computation*, 20(11), 4639–4653. <https://doi.org/10.1021/acs.jctc.4c00352>
- [12] Møller, C., & Plesset, M. S. (1934). Note on an approximation treatment for many-electron systems. *Physical review*, 46(7), 618. <https://doi.org/10.1103/PhysRev.46.618>
- [13] Helgaker, T., Jorgensen, P., & Olsen, J. (2013). *Molecular electronic-structure theory*. USA: John Wiley & Sons.
- [14] Knowles, P. J., & Handy, N. C. (1989). A determinant based full configuration interaction program. *Computer physics communications*, 54(1), 75–83. [https://doi.org/10.1016/0010-4655\(89\)90033-7](https://doi.org/10.1016/0010-4655(89)90033-7)
- [15] Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., ..., & Chan, G. K. L. (2018). PySCF: The Python-based simulations of chemistry framework. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 8(1), e1340. <https://doi.org/10.1002/wcms.1340>
- [16] Smith, D. G., Burns, L. A., Simmonett, A. C., Parrish, R. M., Schieber, M. C., Galvelis, R., ..., & Sherrill, C. D. (2020). PSI4 1.4: Open-source software for high-throughput quantum chemistry. *The Journal of Chemical Physics*, 152(18), 184108. <https://doi.org/10.1063/5.0006002>
- [17] Ollitrault, P. J., Gonthier, J. F., Rocca, D., Anselmetti, G. L., Degroote, M., Moll, N., ..., & Streif, M. (2025). Improving the runtime of quantum phase estimation for chemistry through basis set optimization. *Journal of Chemical Theory and Computation*, 21(23), 12034–12047. <https://doi.org/10.1021/acs.jctc.5c01512>
- [18] Zini, M. S., Delgado, A., dos Reis, R., Casares, P. A. M., Mueller, J. E., Voigt, A. C., & Arrazola, J. M. (2023). Quantum simulation of battery materials using ionic pseudopotentials. *Quantum*, 7, 1049. <https://quantum-journal.org/papers/q-2023-07-10-1049/>

- [19] Kim, I. H., Liu, Y. H., Pallister, S., Pol, W., Roberts, S., & Lee, E. (2022). Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules. *Physical Review Research*, 4(2), 023019. <https://doi.org/10.1103/PhysRevResearch.4.023019>
- [20] Rubin, N. C., Berry, D. W., Malone, F. D., White, A. F., Khattar, T., DePrince, III, E. A., . . . , & Babbush, R. (2023). Fault-tolerant quantum simulation of materials using Bloch orbitals. *PRX Quantum*, 4(4), 040303. <https://doi.org/10.1103/PRXQuantum.4.040303>
- [21] O'Brien, T. E., Streif, M., Rubin, N. C., Santagati, R., Su, Y., Huggins, W. J., . . . , & Babbush, R. (2022). Efficient quantum computation of molecular forces and other energy gradients. *Physical Review Research*, 4(4), 043210. <https://doi.org/10.1103/PhysRevResearch.4.043210>
- [22] Steudtner, M., Morley-Short, S., Pol, W., Sim, S., Cortes, C. L., Loipersberger, M., . . . , & Streif, M. (2023). Fault-tolerant quantum computation of molecular observables. *Quantum*, 7, 1164. <https://doi.org/10.22331/q-2023-11-06-1164>
- [23] Bellonzi, N., Kunitsa, A., Cantin, J. T., Campos-Gonzalez-Angulo, J. A., Radin, M. D., Zhou, Y., . . . , & Cao, Y. (2024). Feasibility of accelerating homogeneous catalyst discovery with fault-tolerant quantum computers. *arXiv Preprint: 2406.06335*
- [24] Ollitrault, P. J., Loipersberger, M., Parrish, R. M., Erhard, A., Maier, C., Sommer, C., . . . , & Streif, M. (2024). Estimation of electrostatic interaction energies on a trapped-ion quantum computer. *ACS Central Science*, 10(4), 882–889. <https://doi.org/10.1021/acscentsci.4c00058>
- [25] Pedersen, T. B., Lehtola, S., Galván, Fdez., I., & Lindh, R. (2024). The versatility of the Cholesky decomposition in electronic structure theory. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 14(1), e1692. <https://doi.org/10.1002/wcms.1692>

How to Cite: Vargas Cruz, M. A. (2026). Adsorption-Calibrated Double Factorization with λ -Metrics for Qubitization-Oriented Hamiltonians. *Archives of Advanced Engineering Science*. <https://doi.org/10.47852/bonviewAAES62029081>