

# Bimal Gaudel

Blacksburg, VA | [bimalgaudel@gmail.com](mailto:bimalgaudel@gmail.com) | [bgaudel.com](http://bgaudel.com) | [github.com/bimalgaudel](https://github.com/bimalgaudel) | [linkedin.com/in/bimalgaudel](https://linkedin.com/in/bimalgaudel) |

Google Scholar

## PROFESSIONAL SUMMARY

Research engineer with 7+ years building high-performance scientific software at the intersection of compiler design, distributed computing, and quantum chemistry. PhD from Virginia Tech (2025). Lead designer and implementer of SeQuant's complete compiler pipeline -- embedded DSL, graph-canonicalized IR, DP optimizer, tree-walk interpreter, and MPI runtime -- over seven years. Deep expertise in modern C++ (17/20/23), HPC runtime design, tensor algebra, and symbolic computation. Track record of compressing months of researcher development work into days through principled system design.

## CORE COMPETENCIES

Modern C++ (17/20/23)

HPC Runtime Design

Compiler & DSL Engineering

Distributed Systems (MPI)

Numerical & Tensor Libraries

Graph-Theoretic Optimization

Scientific Software Architecture

Performance Engineering

## WORK EXPERIENCE

### Virginia Tech, Valeev Group

2018 -- 2025

#### Research Assistant — Blacksburg, VA

- Led the design and implementation of SeQuant's complete compiler pipeline from scratch: an embedded DSL that lets researchers express quantum chemistry equations as native C++ (no parser, no separate syntax, full type safety), plus an optimizing runtime that executes them at supercomputer scale.
- Designed the IR as a full binary tree with extensible per-backend metadata and canonical node identity derived from graph (tensor network) canonicalization, enabling robust common-subexpression elimination across separate equations.
- Implemented bottom-up dynamic-programming tensor network contraction ordering (TNCO) that runs at execution time using actual index-space extents, producing tighter cost estimates than offline compilers that work symbolically.
- Built the tree-walk interpreter (post-order traversal): leaf evaluator fetches raw tensor data and permutes to canonical layout; each internal node queries a canonical-identity-keyed cache before computing; cache evicts dead intermediates via pending-use counters.
- Designed a backend interface so clean that an external contributor integrated a new numerical framework (TAPP) in under 30 minutes; BTAS and TiledArray backends ship with SeQuant.
- Built dual execution modes: runtime interpretation via TiledArray, validated in MPQC production on NERSC Perlmutter; and transpilation to C++/Python targeting downstream tensor compilers (TACO), enabling SeQuant to act as a domain-aware front-end for hardware-specific kernel generation.
- Automated derivation and implementation of explicitly correlated (F12) and coupled-cluster methods, reducing researcher development time from months to days.
- Extended TiledArray's operation basis to support tensor-of-tensor data structures (outer-only, inner-only, and mixed contractions; TT×T and TT×TT products; full TT dot products), enabling block-wise compression and PNO-like local correlation methods.
- Impact: virtually every member of the Valeev Group now depends on SeQuant to advance their research.

## PROJECTS

---

### SeQuant

Lead designer and implementer | 2018 -- 2025 | C++20, CMake, MPI, TiledArray | [github.com/ValeevGroup/SeQuant](https://github.com/ValeevGroup/SeQuant)

A system for automating the derivation and numerical execution of advanced electronic structure methods.

- **Front-end (embedded DSL):** SeQuant expressions serve directly as the AST -- researchers write equations in idiomatic C++; no parser, no separate grammar, full type safety.
- **Middle-end:** IR as a full binary tree; canonical node identity via tensor network graph canonicalization; optimization passes include bottom-up DP TNCO at runtime, symbolic CSE via TN canonicalizer, and tensor network fusion.
- **Back-end (interpreter):** Post-order tree-walk interpreter with canonical-identity-keyed memoization cache; leaf evaluator and canonical-layout permutation; cache eviction via pending-use counters; validated in MPQC production with TiledArray backend.
- **Back-end (transpiler):** IR lowering to C++/Python targeting TACO and other downstream tensor compilers; SeQuant as a domain-aware front-end for hardware-specific kernel generation.

### TiledArray

Contributor | 2018 -- 2025 | C++, MPI | [github.com/ValeevGroup/TiledArray](https://github.com/ValeevGroup/TiledArray)

Massively-parallel, block-sparse distributed tensor framework; the primary numerical execution backend for SeQuant.

- Extended the operation basis to support tensor-of-tensor data structures, enabling a new class of PNO-like local correlation algorithms that depend on nested/dependent index spaces.

## SELECTED PUBLICATIONS

---

1. **Gaudel, B.**, Adam, R. G., Melekamburath, A., Masteran, C., Teke, N., Besharatnik, A., Köhn, A., & Valeev, E. F. (2025). *SeQuant Framework for Symbolic and Numerical Tensor Algebra. I. Core Capabilities*. The Journal of Chemical Physics. [doi:10.1063/5.0311913](https://doi.org/10.1063/5.0311913)
2. Masteran, C., **Gaudel, B.**, & Valeev, E. F. (2025). *Toward a Balanced Description of Ground and Excited States with Transcorrelated F12 Methods*. Journal of Chemical Theory and Computation, 21(20), 10329--10339.
3. Powell, S. R., Surjuse, K. A., **Gaudel, B.**, & Valeev, E. F. (2025). *Slimmer Geminals For Accurate F12 Electronic Structure Models*. Journal of Chemical Theory and Computation, 21(18), 8833--8842.
4. Teke, N. K., Melekamburath, A., **Gaudel, B.**, & Valeev, E. F. (2024). *"Best" Iterative Coupled-Cluster Triples Model? More Evidence for 3CC*. The Journal of Physical Chemistry A, 128(45), 9819--9828.
5. Masteran, C., Kumar, A., Teke, N., **Gaudel, B.**, Yanai, T., & Valeev, E. F. (2023). *Comment on "Canonical transcorrelated theory with projected Slater-type geminals" [J. Chem. Phys. 136, 084107 (2012)]*. The Journal of Chemical Physics, 158(5), 57101.

## EDUCATION

---

### PhD, Theoretical & Computational Chemistry — Virginia Tech

2018 -- 2025

Dissertation: *Automated Implementation of Advanced Electronic Structure Methods*. Advisor: Edward Valeev.

### MS, Physical Chemistry — Tribhuvan University

2014 -- 2016

Kathmandu, Nepal.

## SKILLS

---

**Languages:** C++ (17/20/23) -- expert; CMake, Python, Shell -- proficient; Haskell -- familiar

**HPC & Distributed:** MPI, TiledArray, NERSC Perlmutter (Cray EX), Slurm-based clusters

**Numerical Libraries:** BLAS, LAPACK, Eigen, BTAS, HDF5; familiar with ScaLAPACK

**Compiler & Language Design:** Embedded DSL design (C++), AST/IR design, tree-walk interpretation, code generation / transpilation (C++/Python), graph-theoretic optimization (TNCO, CSE)

**Scientific Computing:** Coupled-cluster theory, reduced-scaling coupled-cluster, many-body perturbation theory, F12/R12 explicitly correlated methods, symbolic algebra, tensor network optimization

**Performance Engineering:** perf, Valgrind, Tracy; distributed-memory profiling and bottleneck diagnosis

**Tools:** Git, GitHub Actions (CI/CD across Linux/macOS), Docker