2025 NCHC Open Hackathon

Team-04-Qa-MolGen

OpenACC
More Science, Less Programming

Team Members

Mentors

Yun-Yuan Wang
Solutions Architect
NVIDIA

Leaders

副教授 陳雨澤 (Prof.Yu-Ze, Chen)
Material Science Engineering
NCKU

教授 吳家樂 (Prof. Ka-Lok Ng)
Bioinformatics and Medical Engineering
Asia University

Members

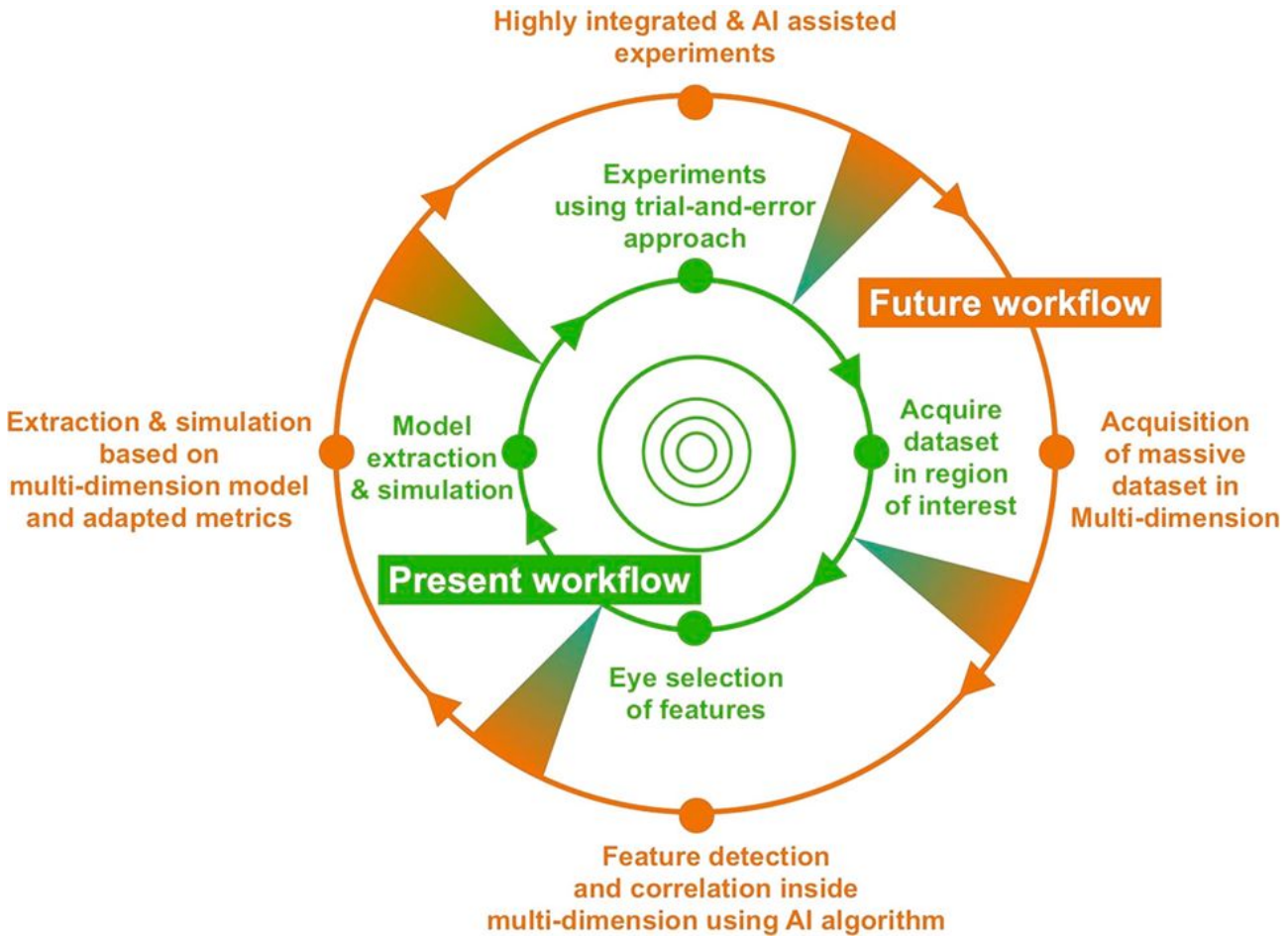
吳澍齊博士 (Dr. Shu-Chi, Wu)
Material Science Engineering
NTHU

Aninda Astuti
PhD - Bioinformatics and Medical Engineering
Asia University

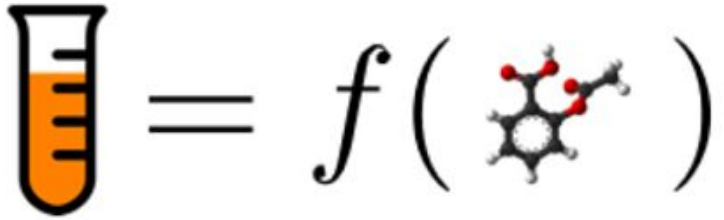
蕭宇承 (Yu-Cheng, Xiao)
Master - Material Science Engineering
NCKU

郭子翎 (Tsu-Ling, Kuo)
Undergraduate - Intelligence Computing and Big Data
CYCU

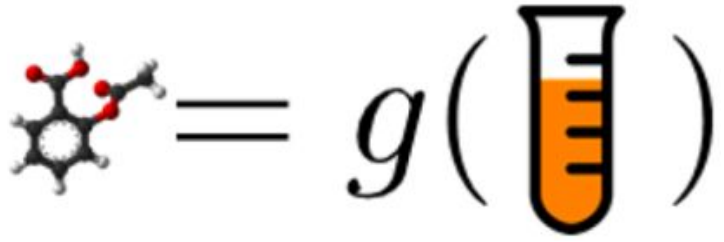
AI & Quantum meet Materials Science?



Property prediction



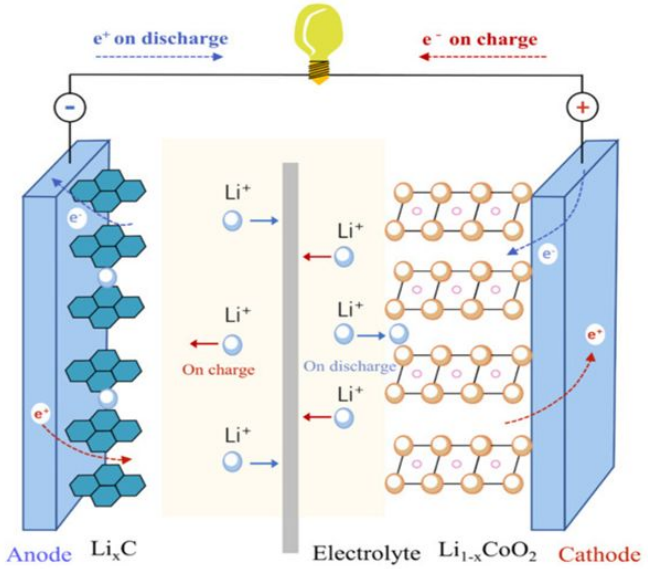
Molecule generation



Lombardo, T.; Duquesnoy, M.; El-Bouysidy, H.; Årén, F.; Gallo-Bueno, A.; Jørgensen, P. B.; Franco, A. Chem. Rev. 2021, 122 (12), 10899–10969.

Case study - Aqueous zinc ion battery

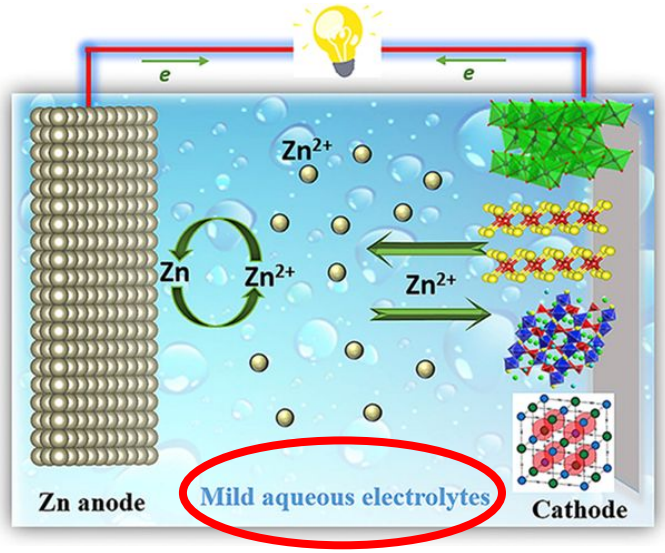
Lithium-ion batteries



Aqueous electrolyte

Non-flammable
Non-toxic

Aqueous zinc ion battery



Method: Electrolyte Optimization
Organic electrolyte additives provide versatile functionalities that effectively inhibit undesirable side reactions.

New appropriate molecules ???

Liu, Y.; Wang, H.; Li, Q.; Zhou, L.; Zhao, P.; Holze, R.*Energies* **2023**, *16* , 7443.

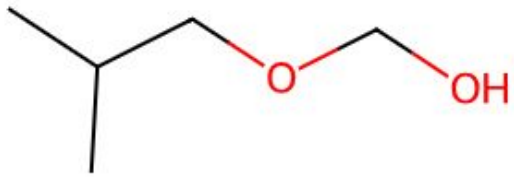
Quantum-based Molecule generation (QMG)

Advantages:

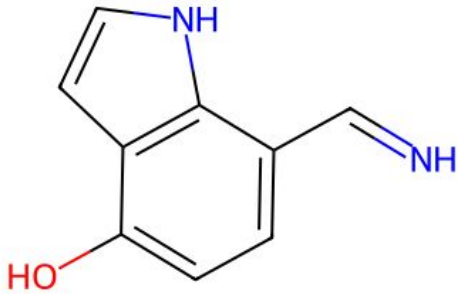
- Support **flexible choice of atom types** via quantum superposition.
- Fix **specific groups** (e.g., –OH, –CN, –SO_3) while generating valid molecule.
- Use **fewer parameters** (10^2) than typical generators (10^4~10^6).

3 main Functionalities

De novo generation	Scaffold decoration	Linker generation



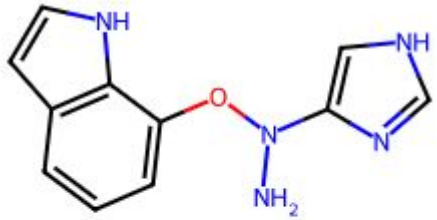
C[C@@H](C)COCO
Count: 2



N=Cc1ccc(O)c2cc[nH]c12

n=1 | SA=3.06

MW=160.2 | LogP=1.9



NN(Oc1cccc2cc[nH]c12)c1c[nH]cn1
Count: 1

OpenACC More Science, Less Programming

OPEN HACKATHONS

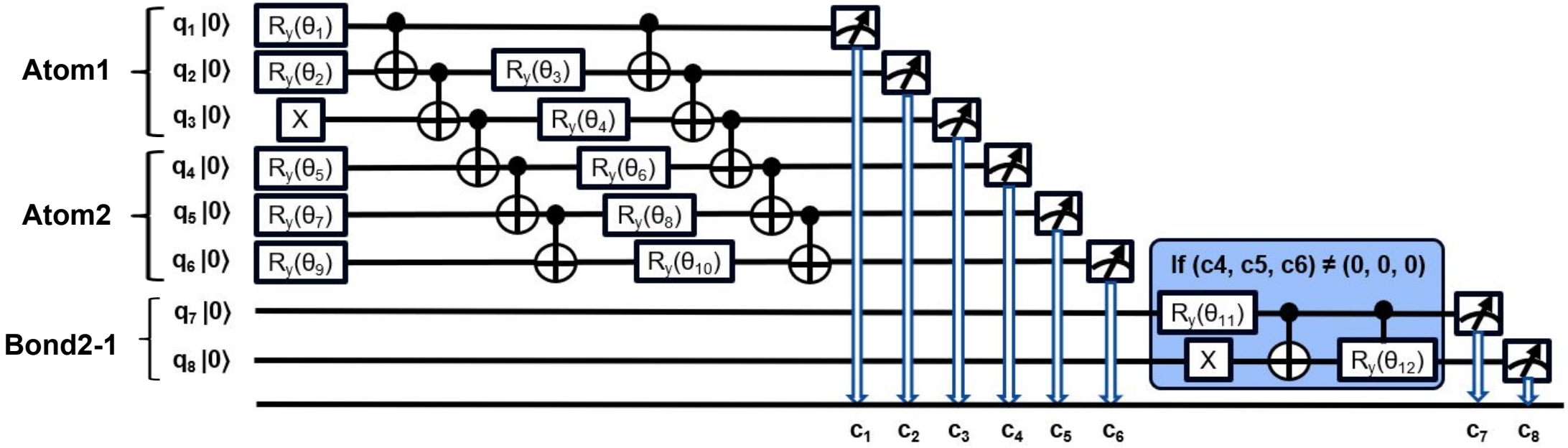
NCHC 國家實驗研究院 國家高速網路與計算中心 National Center for High-performance Computing

NVIDIA

Quantum-based Molecule generation (QMG)

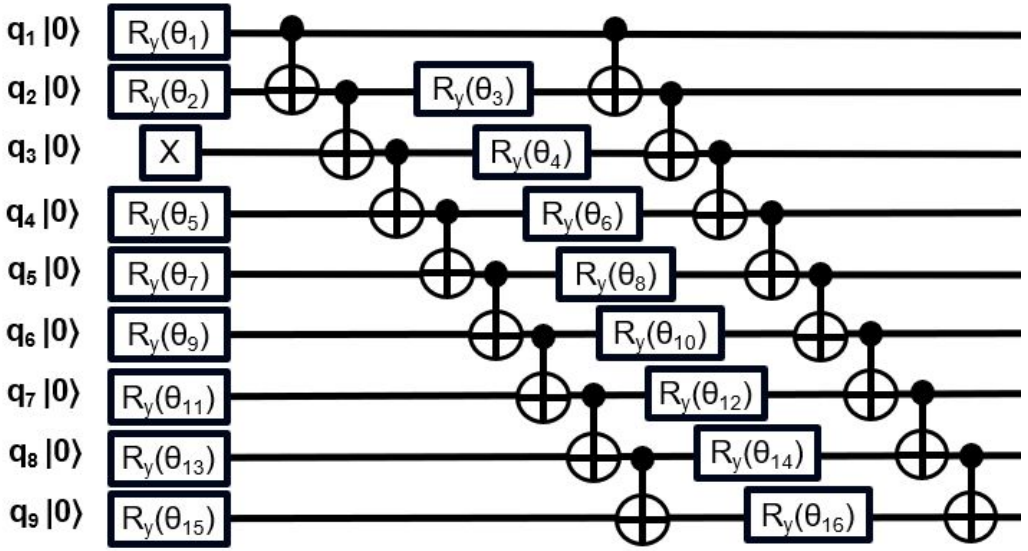
Subcircuits for generating first two atoms & bond 2-1 type

- Atom types (3 qubits) → “None”: 0, "C": 1, "O":2, "N": 3, "S": 4, "P": 5, "F": 6, "Cl": 7
- Bond types (2 qubits) → None: 0, Single: 1, Double: 2, Triple: 3



Dynamic circuit - 3 heavy atoms

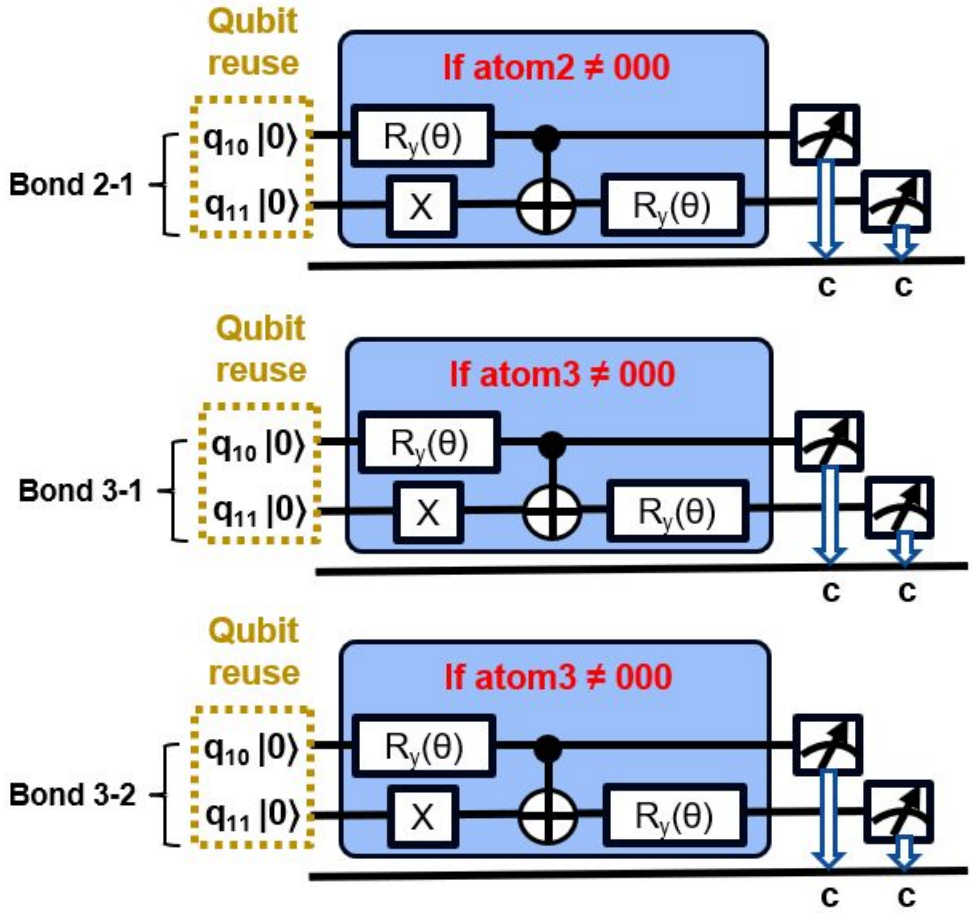
Atom generation



- Total qubits = 3*N + 2
- why qubit reuse?

  Bond number = C(N, 2)

Bond generation



Acceleration Method

Motivation:

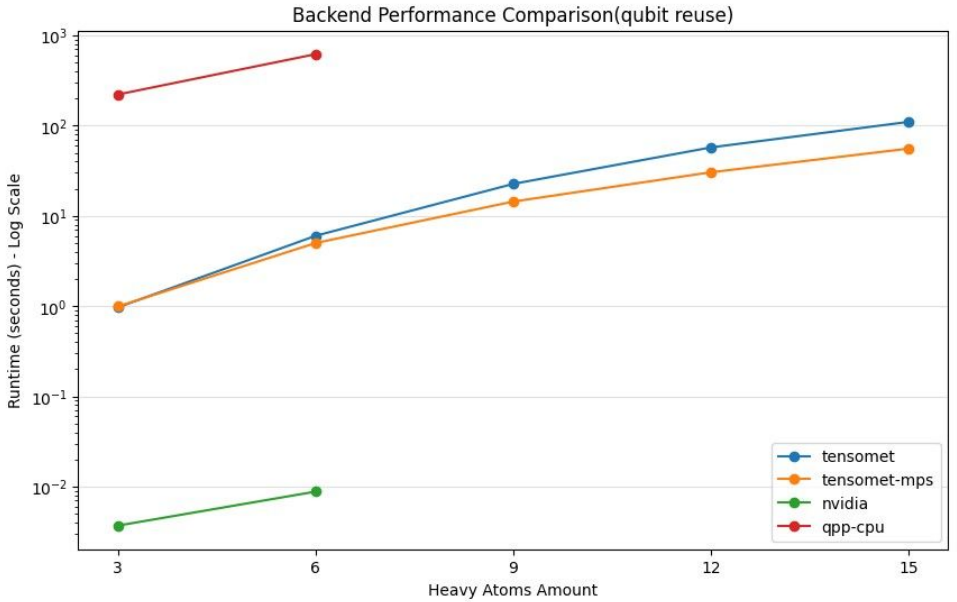
- **Heavy-atom count ↑ ⇒ required qubits ↑ ⇒ circuit width / depth ↑ ⇒ exponential slowdowns on simulators/backend**

Strategy:

- **Benchmark multiple acceleration paths and pick the most effective under fixed accuracy constraints**

'qpp-cpu'	Runs small CPU simulations with a handful of qubits
'nvidia'	Accelerates simulation with single GPU using 'fp64' or 'fp32' (default)
'tensornet'	Runs multi-GPU multi-Node simulations of circuits using tensor networks
'tensornet-mps'	Runs GPU accelerated matrix product state simulations

Backend Comparison - qubit reuse



Qiskit		
# heavy atoms	3 (11 qubits)	6 (14 qubits)
runtime (s)	0.881	823

tensornet_mps			
# heavy atoms	3 (11 qubits)	6 (14 qubits)	9 (20 qubits)
runtime (s)	0.991	5	14.4

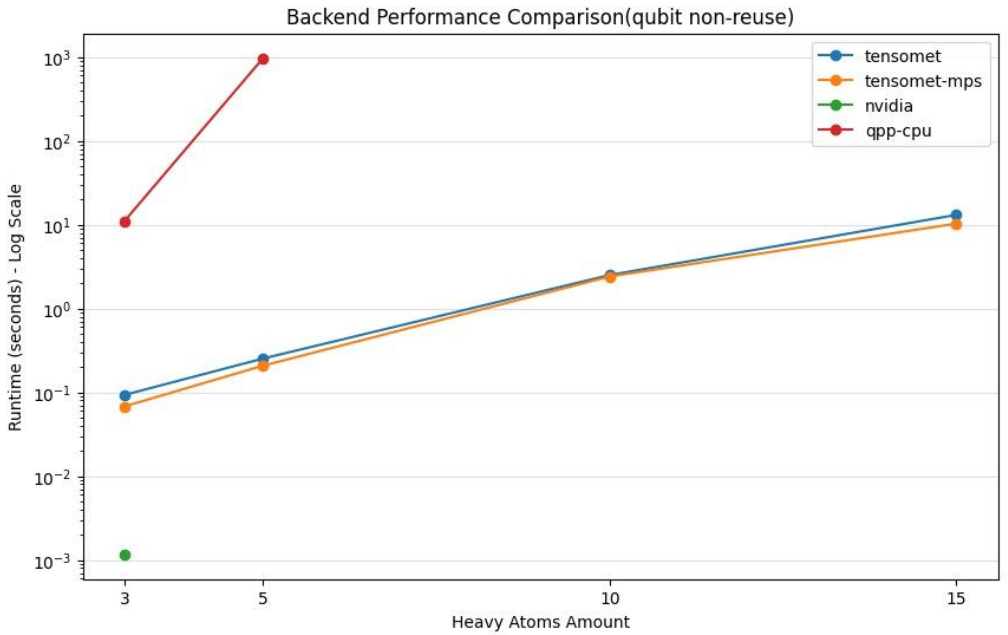
```
# for atom generation
param_idx = 4
for atom_idx in range(1, num_atoms+1):
    ry(np.pi * weight_vector[0], q[0])
    ry(np.pi * weight_vector[1], q[1])
    x(q[2])
    if atom_idx >= 2:
        for i in range(3, 3*atom_idx):
            idx_even = 2 * i - 2
                ry(np.pi * weight_vector[idx_even], q[i])
```

6 heavy atoms

Using the tensornet_mps backend

achieved a 164.6× speedup!!!

Backend Comparison - No qubit reuse



Qiskit	
# heavy atoms	3 (15 qubits)
runtime (s)	363

tensornet_mps			
# heavy atoms	3 (15 qubits)	5 (35 qubits)	10 (120 qubits)
runtime (s)	0.067	0.208	2.41

```
# All atom generation
param_idx = 4+6*(num_atoms-1)
ry(np.pi * weight_vector[0], q[0])
ry(np.pi * weight_vector[1], q[1])
x(q[2])
for i in range(3, 3*num_atoms):
    idx_even = 2 * i - 2
    if idx_even < num_params:
        ry(np.pi * weight_vector[idx_even], q[i])
```

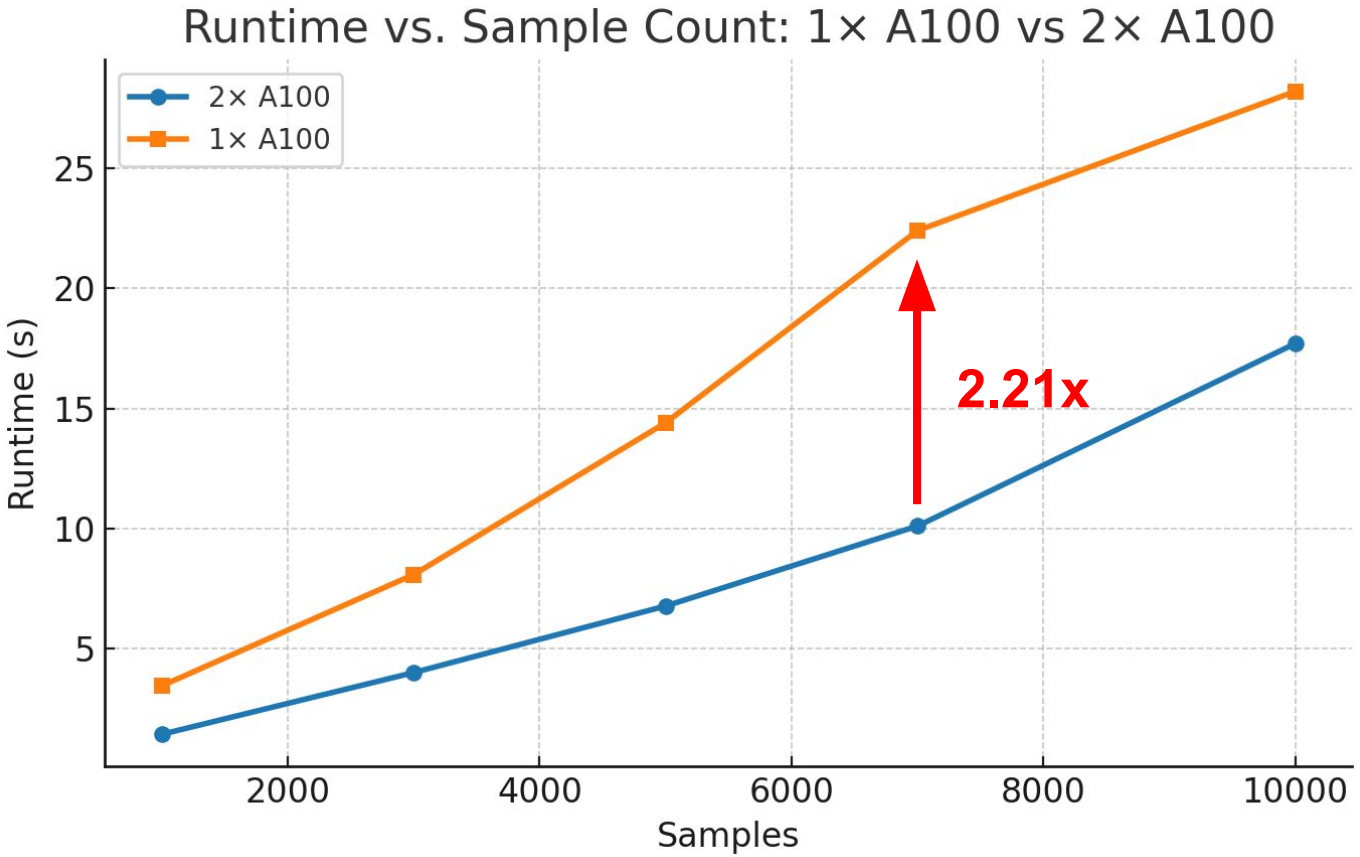
3 heavy atoms

Using the tensornet_mps backend

achieved a 5417.9× speedup!!!

Multi GPU - Simulation

- 3 heavy atoms -> 17 qubits
- Backend: tensornet_mps



Future work

- **Make the code more efficient**: Qubit reuse made explicit. Eliminate hidden overwrites.
- **Circuit design**. Parameter sharing across atoms, shallow entanglers, layout that matches connectivity.
- **Chemistry constraints**: Probabilistic for atom types and single > double > triple bond
- **Model training**: Using RDKit to do property-guided training.
- Model functionalities

Energy Efficiency

INPUTS	
# CPU Cores	64
# GPUs (A100)	2
Application Speedup	5417.9x
Node Replacement	10835.8x

GPU NODE POWER SAVINGS			
	AMD Dual Rome 7742	8x A100 80GB SXM4	Power Savings
Compute Power (W)	11,919,380	6,500	11,912,880
Networking Power (W)	503,179	93	503,086
Total Power (W)	**12,422,559**	**6,593**	12,415,966

Node Power efficiency	**1884.2x**

ANNUAL ENERGY SAVINGS PER GPU NODE			
	AMD Dual Rome 7742	8x A100 80GB SXM4	Power Savings
Compute Power (kWh/year)	104,413,769	56,940	104,356,829
Networking Power (kWh/year)	4,407,849	814	4,407,035
Total Power (kWh/year)	**108,821,618**	**57,754**	**108,763,864**

$/kWh	$	0.18
Annual Cost Savings	$	19,577,495.55
3-year Cost Savings	$	58,732,486.65

Metric Tons of CO_2	77,114
Gasoline Cars Driven for 1 year	16,641
Seedlings Trees grown for 10 years	1,274,712

(source: Link)

Quantum-based molecule generation

- 目標：以 Quantum-based Molecule Generation（QMG）產生具化學限制的候選分子，用於電解液添加劑與功能基導向設計。
- 設計要點：每個原子以 3 qubits 表示，鍵結以 2 qubits 表示；支援 動態電路、量測後重複使用 qubits，以降低參數與記憶體占用。
- 規模：在單卡 A100 上，已能 穩定運行到 ≥15 個 heavy atoms，並保留中途量測資訊以避免鍵結與「空原子」相連。

Caption describing figure in simple terms

Hackathon Objectives and Approach

目標

- 擴大可處理原子數並減少參數量。
- 端到端效能優化：電路建構 / transpile / 執行 / 後處理。

作法

- 後端策略： qppcpu 驗證正確性與單元測試。
- nvidia / tensornet 加速抽樣；大型電路用 tensornet_mps 省記憶體。

Technical Accomplishments and Impact

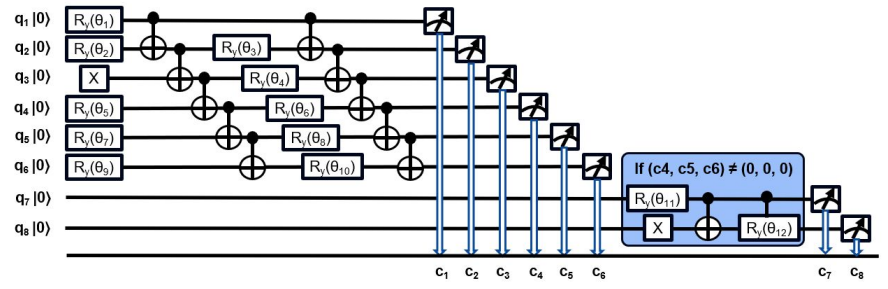
成果

- 可擴充度：由舊版 2 qubits per atom，提升到 3 qubits per atom 穩定抽樣與後處理。
- 效率：以 tensornet/nvidia 後端與早停規則，明顯降低抽樣時間與記憶體占用；後處理向量化縮短 I/O 與 SMILES 轉換時間。

影響

- 較小的參數量與 qubit 重用降低硬體需求，使 單卡 A100 即可探索更大分子圖。
- 穩定與可重現的工作流，有利於後續 效能基準、對照試驗 與與傳統生成模型的公平比較。

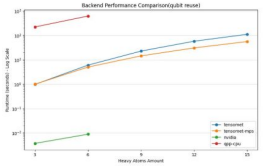
量子分子生成模型 (Quantum-based molecule generation)



Qa-MolGen 團隊來自成功大學材料系「陳雨澤教授」以及亞洲大學資工系「吳家樂教授」帶領的實驗室成員，將量子分子生成模型加速了5417.9倍！！

在材料科學領域，由於材料種類以及數量繁多，傳統做法靠人寫規則或大量實驗，速度慢且成本高。當前 QMG 把「原子三位元、鍵兩位元」的生成邏輯放進量子電路，用機率分佈一次抽樣很多候選分子，同時可依照個人需求進行原子的更換以及擴增。
模型進入大量抽樣時，位元數量會線性增長、計算量與記憶體會指數上升，因此我們採用 NVIDIA cudaq 提升推論的速度，甚至能增加可使用的位元數量，方便跑大規模的 seeds、上千次抽樣，快速統計分子結構的 validity 與 uniqueness，對於探索複雜的化學空間有嶄新的突破。

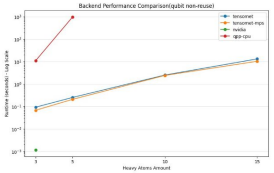
qubit reuse



Qiskit		
# heavy atoms	3 (11 qubits)	6 (14 qubits)
runtime (s)	0.881	823

tensornet_mps			
# heavy atoms	3 (11 qubits)	6 (14 qubits)	9 (20 qubits)
runtime (s)	0.991	5	14.4

qubit noreuse

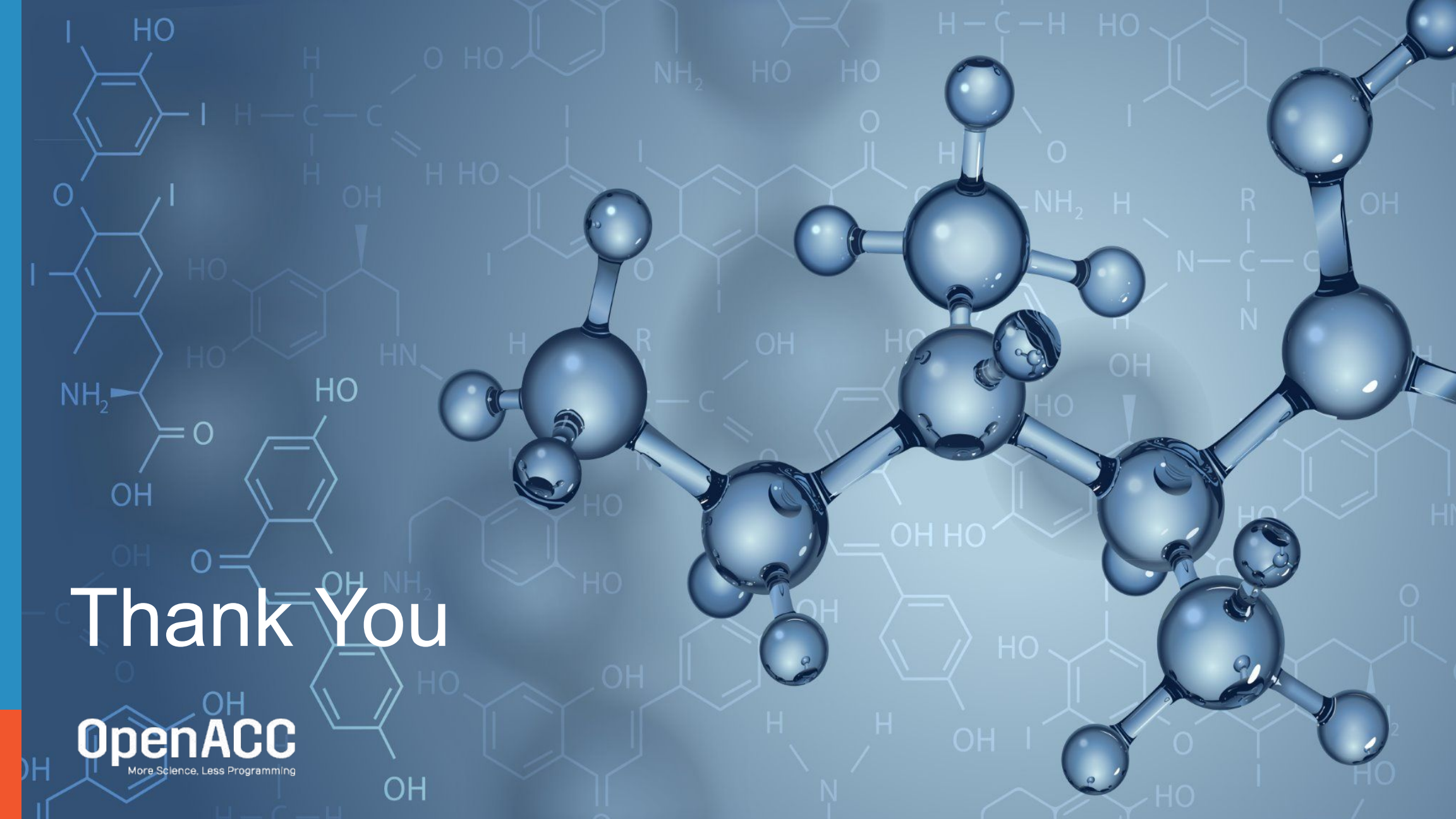


Qiskit		
# heavy atoms	3 (15 qubits)	
runtime (s)	363	

tensornet_mps			
# heavy atoms	3 (15 qubits)	5 (35 qubits)	10 (120 qubits)
runtime (s)	0.067	0.208	2.41

報告投影片連結 (由國網上傳到 github)

OpenACC More Science, Less Programming
OPEN HACKATHONS
NCHC 國家實驗研究院 國家高速網路與計算中心 National Center for High-performance Computing
NVIDIA


Thank You
OpenACC
More Science, Less Programming