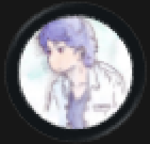
Qiskit Advocate Mentorship Program

#20
Implement new features and improve documentation in Operators



Mentor:

Ikko Hamamura

Mentees:

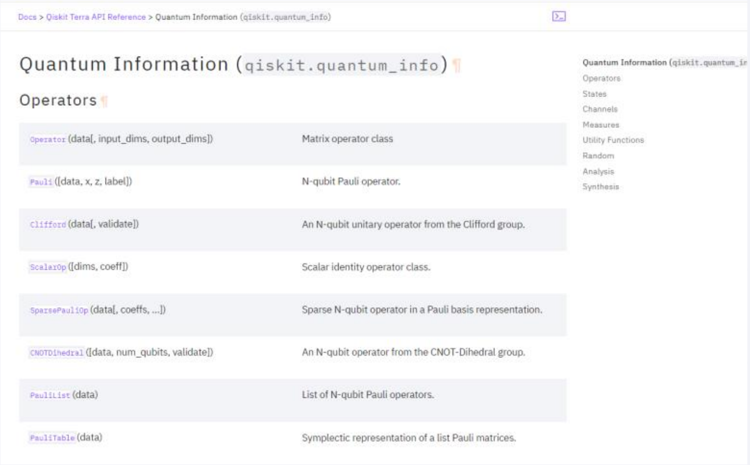
Daiki
Murata

Yuma
Nakamura

Kazumasa
Umezawa

IBM Quantum / © 2021 IBM Corporation

Make code contribution around Operators



Docs > Qiskit Terra API Reference > Quantum Information (qiskit.quantum_info)

Quantum Information (`qiskit.quantum_info`)¶

Operators¶

`Operator` (data[, input_dims, output_dims])	Matrix operator class
`Pauli` ([data, x, z, label])	N-qubit Pauli operator.
`Clifford` (data[, validate])	An N-qubit unitary operator from the Clifford group.
`ScalarOp` ([dims, coeff])	Scalar identity operator class.
`SparsePauliOp` (data[, coeffs, ...])	Sparse N-qubit operator in a Pauli basis representation.
`CNOTDihedral` ([data, num_qubits, validate])	An N-qubit operator from the CNOT-Dihedral group.
`PauliList` (data)	List of N-qubit Pauli operators.
`PauliTable` (data)	Symplectic representation of a list Pauli matrices.

Quantum Information (qiskit.quantum_in
Operators
States
Channels
Measures
Utility Functions
Random
Analysis
Synthesis

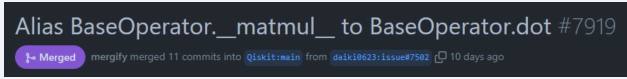
No.	Member	Summary
1		Alias BaseOperator._matmul _ to BaseOperator.dot
2		Alias BaseOperator._mul _ to BaseOperator._multiply_
3		Check input label SparsePauliOp
4		Remove deprecated classes and methods
5		Implement argsort and sort method to SparsePauliOp
6		Investigation to improve the performance of to_matrix ()
7		Implement general grouping of PauliOp/SparsePauliOp by commutation
8		Investigation to improve the performance of time evolution of Ops

Improvement of *BaseOperator*

【Motivation】

To make easier to read the code and develop, add features to *BaseOperator* and *SparsePauliOp*.

- Alias *BaseOperator* method



Alias BaseOperator.__matmul__ to BaseOperator.dot #7919

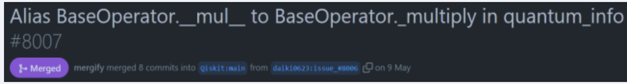
Merged mergify merged 11 commits into Qiskit:main from daiki0623:issue#7502 10 days ago

  - .__mutmul__ to .dot

equation	alias	method

AB	A @ B
	B & A
		A.dot(B)
		B.compose(A)

  - .__mul__ to .multiply

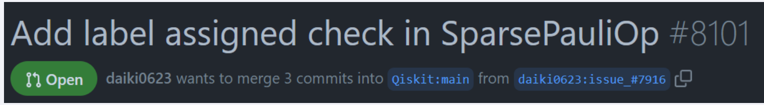


Alias BaseOperator.__mul__ to BaseOperator._multiply in quantum_info #8007

Merged mergify merged 8 commits into Qiskit:main from daiki0623:issue_#8006 on 9 May

Being able to execute right multiplication such as *Pauli(“X”) * -1* (raise QiskitError before)

Improvement of *SparsePauliOp*

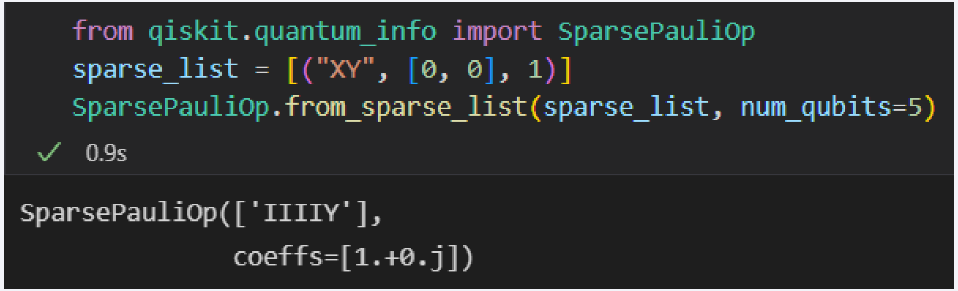


Add label assigned check in SparsePauliOp #8101

daiki0623 wants to merge 3 commits into Qiskit:main from daiki0623:issue_#7916

There is a limitation of initializing *SparsePauliOp* from list.

Initialization success even if multiple gates are on the same qubit.



```
from qiskit.quantum_info import SparsePauliOp
sparse_list = [("XY", [0, 0], 1)]
SparsePauliOp.from_sparse_list(sparse_list, num_qubits=5)
```

✓ 0.9s

```
SparsePauliOp(['IIIIY'],
              coeffs=[1.+0.j])
```

Added list check process, and raise Error when above case.

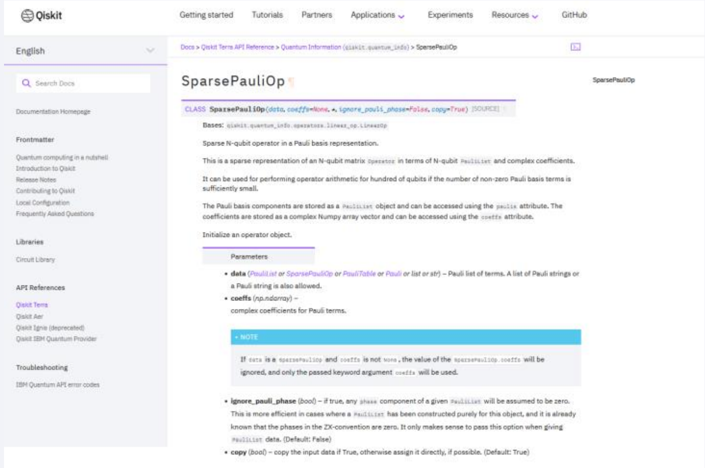
OK: [(“XY”, [0, 1], 1)]

NG: [(“XY”, [0, 0], 1)], [(“XI”, [0, 0], 1)], [(“IX”, [0, 0], 1)]

Implement argsort and sort method to SparsePauliOp

【SparsePauliOp】

This is a sparse representation of an N-qubit matrix Operator in terms of N-qubit PauliList and complex coefficients.



SparsePauliOp

`CLASS SparsePauliOp(data, coeffs=None, *, ignore_pauli_phase=False, copy=True)` [SOURCE]

Bases: `qiskit.quantum_info.operators.linear_op.LinearOp`

Sparse N-qubit operator in a Pauli basis representation.

This is a sparse representation of an N-qubit matrix `Operator` in terms of N-qubit `PauliList` and complex coefficients.

It can be used for performing operator arithmetic for hundred of qubits if the number of non-zero Pauli basis terms is sufficiently small.

The Pauli basis components are stored as a `PauliList` object and can be accessed using the `paulis` attribute. The coefficients are stored as a complex Numpy array vector and can be accessed using the `coeffs` attribute.

Initialize an operator object.

Parameters

- **data** (*PauliList or SparsePauliOp or PauliTable or Pauli or list or str*) – Pauli list of terms. A list of Pauli strings or a Pauli string is also allowed.
- **coeffs** (*np.ndarray*) – complex coefficients for Pauli terms.

NOTE

If `data` is a `SparsePauliOp` and `coeffs` is not `None`, the value of the `SparsePauliOp.coeffs` will be ignored, and only the passed keyword argument `coeffs` will be used.

- **ignore_pauli_phase** (*bool*) – if true, any `phase` component of a given `PauliList` will be assumed to be zero. This is more efficient in cases where a `PauliList` has been constructed purely for this object, and it is already known that the phases in the ZX-convention are zero. It only makes sense to pass this option when giving `PauliList` data. (Default: False)
- **copy** (*bool*) – copy the input data if True, otherwise assign it directly, if possible. (Default: True)

【Motivation】

PauliList has argosort() method and sort() method .

We need these features in SparcePauliOp .

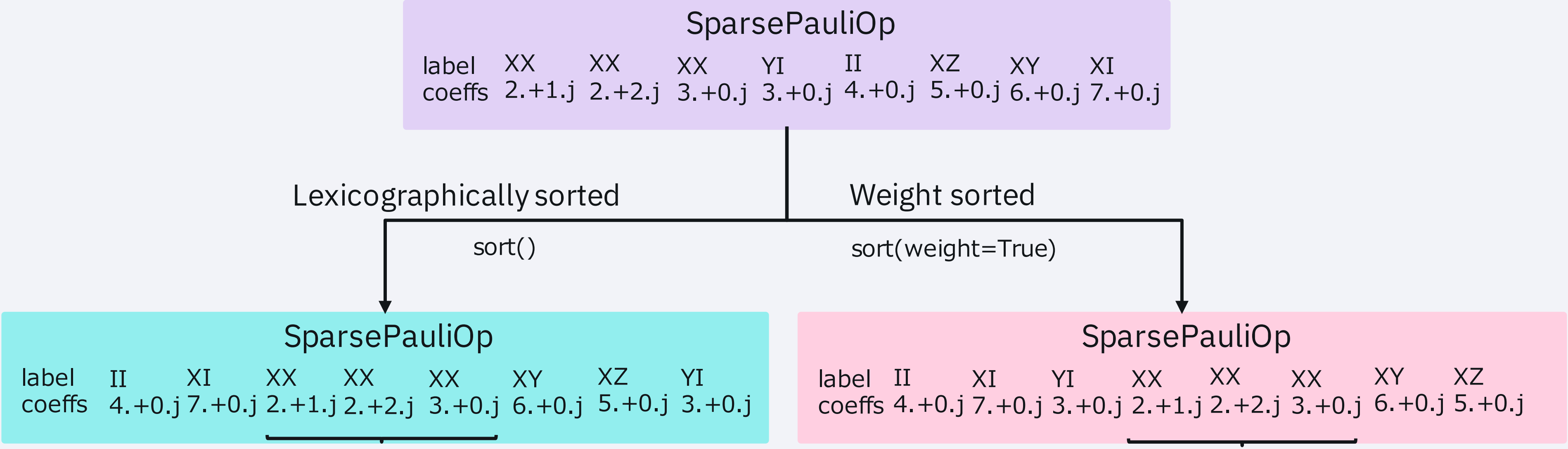
【Done】

1. Added two methods for the following functions to SparsePauliOp.
2. Added 7 cases of tests related to them.
3. sent a pull request.

https://github.com/Qiskit/qiskit-terra/pull/8016

Features of SparcePauliOp sort

【 Features of SparcePauliOp sort 】
After sorting the coefficients using numpy's argsort, sort by Pauli.



SparsePauliOp

label	XX	XX	XX	YI	II	XZ	XY	XI
coeffs	2.+1.j	2.+2.j	3.+0.j	3.+0.j	4.+0.j	5.+0.j	6.+0.j	7.+0.j

Lexicographically sorted

sort()

Weight sorted

sort(weight=True)

SparsePauliOp

label	II	XI	XX	XX	XX	XY	XZ	YI
coeffs	4.+0.j	7.+0.j	2.+1.j	2.+2.j	3.+0.j	6.+0.j	5.+0.j	3.+0.j

If Pauli is the same, it will be sorted by coefficient.

SparsePauliOp

label	II	XI	YI	XX	XX	XX	XY	XZ
coeffs	4.+0.j	7.+0.j	3.+0.j	2.+1.j	2.+2.j	3.+0.j	6.+0.j	5.+0.j

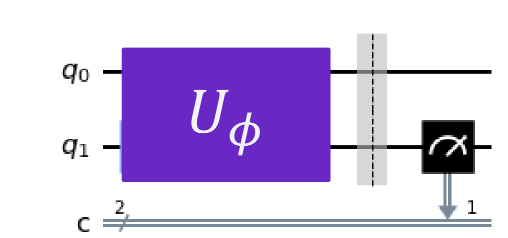
If Pauli is the same, it will be sorted by coefficient.

Efficient Evaluation of Observable

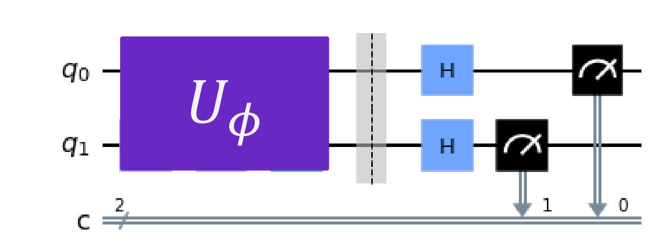
Background

Evaluation of observable requires multiple measurement,
where commutating observable can be evaluated simultaneously (grouping).

e.g. H = <I@Z> + <X@X> + <I@X> can be evaluated only by calculating <I@Z> and <X@X>



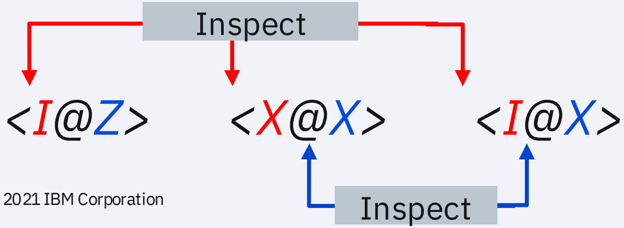
Measurement for <I@Z>



Measurement for <X@X>

Existing Grouping Algorithm

qubit-wide commutation is inspected



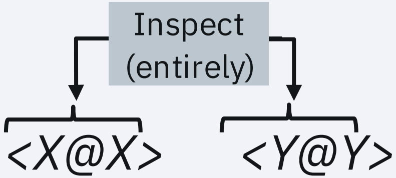
Inspect

<I@Z> <X@X> <I@X>

Inspect

New Algorithm

General commutating rule is employed



Inspect (entirely)

<X@X> <Y@Y>

<X@X> and <Y@Y> are commutating
but qubit-wise commutating rule cannot detect!!

Efficient Evaluation of Observable

Impact

Grouping observable reduce
the number of required measurement.

Its reduction contributes to **faster** VQE.

Pull Request

→ https://github.com/Qiskit/qiskit-terra/pull/7874

Molecule	Transformation	Number of Groups		
		No-grouping	Qubit-wide Commutation	General Commutation
LiH	JW	631	136	35
	Parity		165	35
	BK		211	35
BeH_2	JW	1150	215	58
	Parity		323	58
	BK		341	58
H_2O	JW	1858	380	84
	Parity		495	82
	BK		515	82
NH_3	JW	4973	1052	117
	Parity		1091	115
	BK		1086	115
HCl	JW	4427	906	110
	Parity		1098	112
	BK		1434	112

*value referenced from Ikko Hamamura and Takashi Imamichi, npj Quant. Info. 6, 56 (2020)

Summary

- We are aiming to contribute Qiskit.quantum_info
  - Alias BaseOperator in quantum_info
  - Sort Pauli operators
  - Grouping Pauli operator for simultaneous measurement

Qiskit

Thank you