Supercharging classical simulation of quantum circuits with the ZX-calculus

Aleks Kissinger University of Oxford
Renaud Vilmart Inria Saclay
John van de Wetering Radboud University & Oxford

QPL – June 27, 2022

Quantum circuit simulation

The problem: (somewhat) **efficiently** simulating quantum circuits.

Quantum circuit simulation

The problem: (somewhat) **efficiently** simulating quantum circuits.

Broadly, two ways to do this:

- Tensor-network based methods:
  - direct state simulation
  - principle component analysis
  - contraction order finding
  - edge cutting
  - ...

Quantum circuit simulation

The problem: (somewhat) **efficiently** simulating quantum circuits.

Broadly, two ways to do this:

- Tensor-network based methods:
  - direct state simulation
  - principle component analysis
  - contraction order finding
  - edge cutting
  - ...
- Stabiliser-decomposition based methods:
  - Stabiliser extent
  - Stabiliser rank $\leftarrow$ **This talk**

Stabiliser rank

Gottesman-Knill Theorem

A quantum computation consisting of

- stabiliser states,
- Clifford unitaries,
- stabiliser measurements,

can be efficiently classically simulated.

Stabiliser rank

Gottesman-Knill Theorem

A quantum computation consisting of

- stabiliser states,
- Clifford unitaries,
- stabiliser measurements,

can be efficiently classically simulated.

Observation: Stabiliser states span entire space of states. So:

$$|\psi\rangle \;=\; \sum_{j=1}^{\chi} \lambda_j |\psi_j\rangle \text{ where } |\psi_j\rangle \text{ is stabiliser.}$$

χ is called **stabiliser rank** of $|\psi\rangle$.

Simulating using stabiliser rank

- Start with Clifford+T circuit.
- Write each T gate as magic state injection.
- Decompose T states into sum of stabilisers.
- Efficiently simulate resulting Clifford circuits.
- Add results together.
- We're done!

Simulating using stabiliser rank

- Start with Clifford+T circuit.
- Write each T gate as magic state injection.
- Decompose T states into sum of stabilisers.
- Efficiently simulate resulting Clifford circuits.
- Add results together.
- We're done!

What's the catch?
Stabiliser rank of k T states scales exponentially with k.
... But it's not just 2^k terms. We can do better!

Stabiliser ranks of T magic states

Recall $|T\rangle \ \propto \ |0\rangle + e^{i\pi/4}|1\rangle$.
So $\chi(|T\rangle) = 2$ and hence $\chi(|T\rangle^{\otimes k}) \leq 2^k$.

Stabiliser ranks of T magic states

Recall $|T\rangle \;\propto\; |0\rangle + e^{i\pi/4}|1\rangle$.
So $\chi(|T\rangle) = 2$ and hence $\chi(|T\rangle^{\otimes k}) \leq 2^k$.

But:

$$\begin{aligned} |T\rangle^{\otimes 2} \;&\propto\; |00\rangle + e^{i\pi/4}|01\rangle + e^{i\pi/4}|10\rangle + e^{i\pi/2}|11\rangle \\ &=\; (|00\rangle + i|11\rangle) + e^{i\pi/4}(|01\rangle + |10\rangle) \end{aligned}$$

Stabiliser ranks of T magic states

Recall $|T\rangle \;\propto\; |0\rangle + e^{i\pi/4}|1\rangle$.
So $\chi(|T\rangle) = 2$ and hence $\chi(|T\rangle^{\otimes k}) \leq 2^k$.

But:

$$\begin{aligned} |T\rangle^{\otimes 2} &\propto |00\rangle + e^{i\pi/4}|01\rangle + e^{i\pi/4}|10\rangle + e^{i\pi/2}|11\rangle \\ &= (|00\rangle + i|11\rangle) + e^{i\pi/4}(|01\rangle + |10\rangle) \end{aligned}$$

Hence $\chi(|T\rangle^{\otimes 2}) = 2$ so that

$$\chi(|T\rangle^{\otimes k}) = \chi((|T\rangle^{\otimes 2})^{\otimes k/2}) \leq 2^{k/2} = 2^{0.5k}.$$

Stabiliser ranks of T magic states

Recall $|T\rangle \;\propto\; |0\rangle + e^{i\pi/4}|1\rangle$.
So $\chi(|T\rangle) = 2$ and hence $\chi(|T\rangle^{\otimes k}) \leq 2^k$.

But:

$$\begin{aligned} |T\rangle^{\otimes 2} \;&\propto\; |00\rangle + e^{i\pi/4}|01\rangle + e^{i\pi/4}|10\rangle + e^{i\pi/2}|11\rangle \\ &=\; (|00\rangle + i|11\rangle) + e^{i\pi/4}(|01\rangle + |10\rangle) \end{aligned}$$

Hence $\chi(|T\rangle^{\otimes 2}) = 2$ so that

$$\chi(|T\rangle^{\otimes k}) = \chi((|T\rangle^{\otimes 2})^{\otimes k/2}) \leq 2^{k/2} = 2^{0.5k}.$$

Turns out $\chi(|T\rangle^{\otimes 6}) \leq 7$ so
$\chi(|T\rangle^{\otimes k}) \leq 2^{\alpha k}$ where $\alpha = \log_2(7)/6 \approx 0.467$.
Found by Bravyi, Smith, Smolin (BSS) in 2016.

Our idea

Do this simulation, but with *ZX-diagrams* instead of circuits!

Benefit: optimise intermediate ZX-diagrams to reduce T-count.

Our idea

Do this simulation, but with *ZX-diagrams* instead of circuits!

Benefit: optimise intermediate ZX-diagrams to reduce T-count.

Described in two papers:

- Kissinger & vdW. *Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions*. arXiv:2109.01076
- Kissinger, Vilmart & vdW. *Classical simulation of quantum circuits with partial and graphical stabiliser decompositions*. arXiv:2202.09202

New algorithm

- Write Clifford+T circuit as ZX-diagram.

New algorithm

- Write Clifford+T circuit as ZX-diagram.
- Simplify diagram with your favourite algorithm.

New algorithm

- Write Clifford+T circuit as ZX-diagram.
- Simplify diagram with your favourite algorithm.
- Pick some spiders with T-like phase and decompose them.

New algorithm

- Write Clifford+T circuit as ZX-diagram.
- Simplify diagram with your favourite algorithm.
- Pick some spiders with T-like phase and decompose them.
- Simplify resulting sum of diagrams.

New algorithm

- Write Clifford+T circuit as ZX-diagram.
- Simplify diagram with your favourite algorithm.
- Pick some spiders with T-like phase and decompose them.
- Simplify resulting sum of diagrams.
- Repeat decomposition+optimisation
  until diagrams have reached desired simplicity.

New algorithm

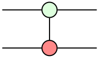
- Write Clifford+T circuit as ZX-diagram.
- Simplify diagram with your favourite algorithm.
- Pick some spiders with T-like phase and decompose them.
- Simplify resulting sum of diagrams.
- Repeat decomposition+optimisation
  until diagrams have reached desired simplicity.
- Simply evaluate all the diagrams to get your outcome!

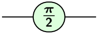
New algorithm

- Write Clifford+T circuit as ZX-diagram.
- Simplify diagram with your favourite algorithm.
- Pick some spiders with T-like phase and decompose them.
- Simplify resulting sum of diagrams.
- Repeat decomposition+optimisation
  until diagrams have reached desired simplicity.
- Simply evaluate all the diagrams to get your outcome!

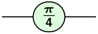
Let's go through these steps in more detail.

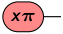
Writing Clifford+T circuit as ZX-diagram

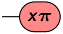
CNOT $=\sqrt{2}$ 

S $=$ 

Had $=$ 

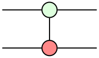
T $=$ 

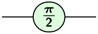
$|x\rangle = \frac{1}{\sqrt{2}}$ 

$\langle x| = \frac{1}{\sqrt{2}}$ 

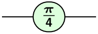
$(x \in \{0,1\})$

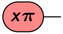
Writing Clifford+T circuit as ZX-diagram

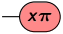
CNOT $= \sqrt{2}$ 

S $=$ 

Had $=$ 

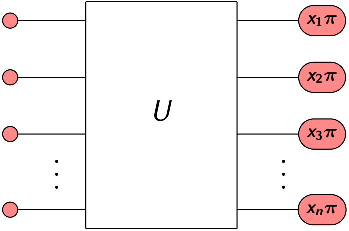
T $=$ 

$|x\rangle = \frac{1}{\sqrt{2}}$ 

$\langle x| = \frac{1}{\sqrt{2}}$ 

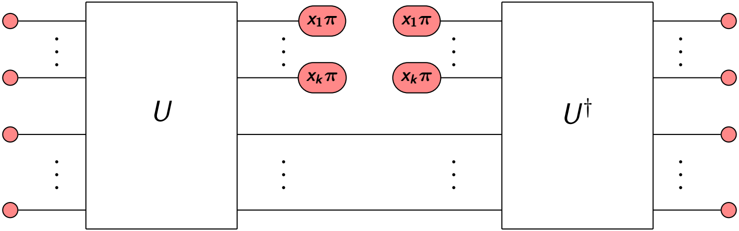
$(x \in \{0, 1\})$

Calculating single amplitude:

$\langle \vec{x} | U | 0 \cdots 0 \rangle = \left(\frac{1}{\sqrt{2}}\right)^{2n}$ 

Marginal probabilities

To calculate marginal probability, use *doubling* technique:



Strong simulation vs Weak simulation

Weak sim: approx. sample from the same output distribution.
Strong sim: approx. calculate any marginal probability.

Strong simulation vs Weak simulation

Weak sim: approx. sample from the same output distribution.
Strong sim: approx. calculate any marginal probability.

We are doing *exact* strong simulation here.

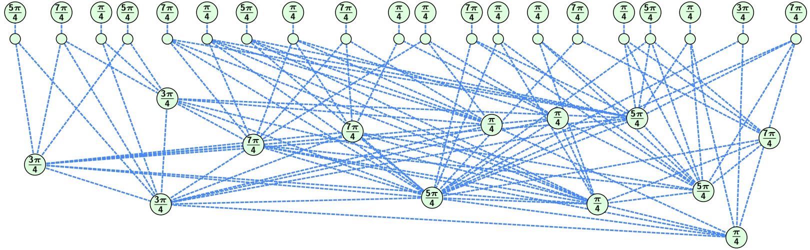
- Writing circuit as ZX-diagram ✓
- Optimising ZX-diagram
- Decomposing magic states

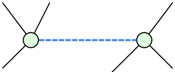
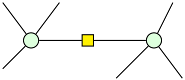
Simplifying ZX-diagrams

We use strategy from our previous paper
Reducing T-count with the ZX-calculus.

Simplifying ZX-diagrams

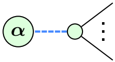
We use strategy from our previous paper
Reducing T-count with the ZX-calculus.
This transforms every ZX-diagram into something like this:



Here  := 

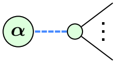
Properties of reduced diagram

The important part:

- Every spider carries a non-Clifford phase,
- or is part of a *phase gadget*: 

Properties of reduced diagram

The important part:

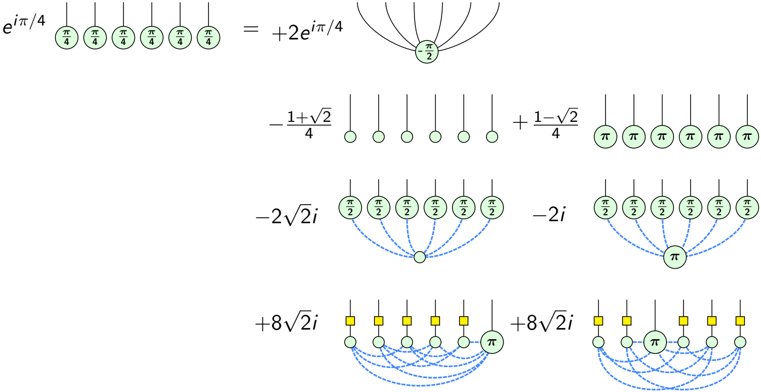
- Every spider carries a non-Clifford phase,
- or is part of a *phase gadget*: 

Particularly: if original circuit had k T gates, resulting diagram has $\leq 2k$ spiders (regardless of #qubits or #gates).

- Writing circuit as ZX-diagram ✓
- Optimising ZX-diagram ✓
- Decomposing magic states

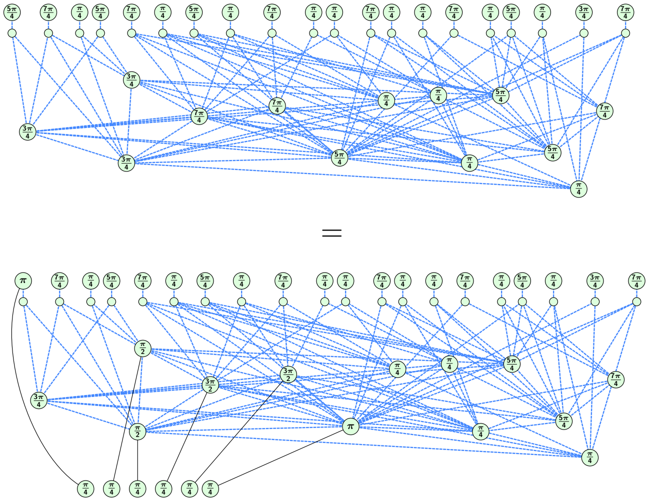
Decomposing T-like spiders

The 6-to-7 magic state decomposition in ZX is:



Applying the decomposition

We pick some spiders to decompose and unfuse the phases:



And now we can apply the magic state decomposition.

Full algorithm

- After first decomposition we have 7 diagrams.
- Repeat whole procedure for each of these diagrams, end up with $7 * 7$ diagrams.
- Repeat until there aren't 6 magic states to decompose.
- Diagrams are then small enough to calculate value of directly.
- Sum values to get the amplitude/prob we were calculating.

Full algorithm

- After first decomposition we have 7 diagrams.
- Repeat whole procedure for each of these diagrams, end up with $7 * 7$ diagrams.
- Repeat until there aren't 6 magic states to decompose.
- Diagrams are then small enough to calculate value of directly.
- Sum values to get the amplitude/prob we were calculating.

But we can do better...

Better decompositions

Improved upper bounds on the stabilizer rank of magic states

Hammam Qassim [*†‡ §]   Hakop Pashayan [*¶ ‖]   David Gosset [*¶]

June 16, 2021

$\Rightarrow$ improved stabiliser decompositions, including 6-to-6 decomp
(giving $\alpha \approx 0.431$ instead of $\alpha \approx 0.467$),
and other decomps giving $\alpha < 0.40$.

Cat states

Qassim et al. uses *cat states*:

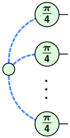
$$|\,\mathrm{cat}_n\rangle := \frac{1}{\sqrt{2}}(\mathbb{I}^{\otimes n} + Z^{\otimes n})|\,T\rangle^{\otimes n} = \frac{1}{\sqrt{2}^{n+1}}\left(\textcircled{\scriptsize $\frac{\pi}{4}$}\!-\!\!-^{\otimes n} + \textcircled{\scriptsize $\frac{5\pi}{4}$}\!-^{\otimes n} \right)$$

Cat states

Qassim et al. uses *cat states*:

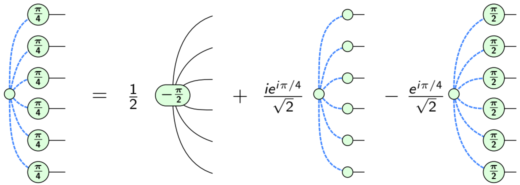
$$|\,\mathrm{cat}_n\rangle := \frac{1}{\sqrt{2}}(\mathbb{I}^{\otimes n} + Z^{\otimes n})|\,T\rangle^{\otimes n} = \frac{1}{\sqrt{2}^{n+1}}\left(\textcircled{\tfrac{\pi}{4}}\!-^{\otimes n} + \textcircled{\tfrac{5\pi}{4}}\!-^{\otimes n} \right)$$

These actually have nice representation in ZX:

$$|\,\mathrm{cat}_n\rangle = \frac{1}{\sqrt{2}}$$ 

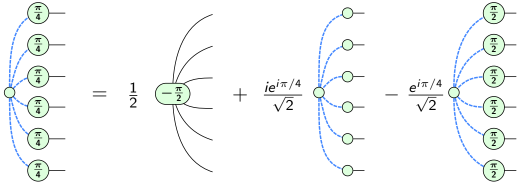
Cat decompositions

They find good decomp of $|\mathrm{cat}_6\rangle$:



Cat decompositions

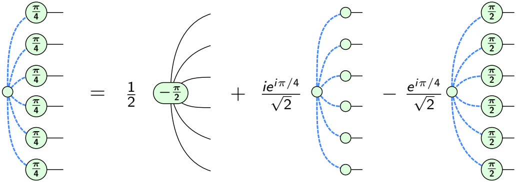
They find good decomp of $|\mathrm{cat}_6\rangle$:



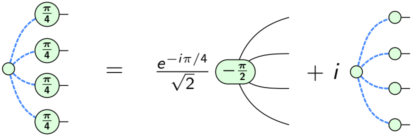
So, if we only have $|\mathrm{cat}_6\rangle$ diagrams, we would have $\alpha \approx 0.264$.

Cat decompositions

They find good decomp of $|\mathrm{cat}_6\rangle$:



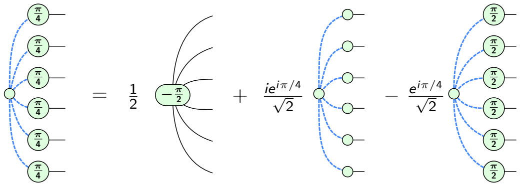
So, if we only have $|\mathrm{cat}_6\rangle$ diagrams, we would have $\alpha \approx 0.264$.
But we can do even better!



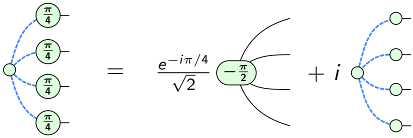
This would give $\alpha = 0.25$.

Cat decompositions

They find good decomp of $|\mathrm{cat}_6\rangle$:



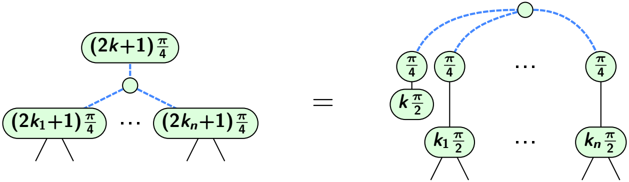
So, if we only have $|\mathrm{cat}_6\rangle$ diagrams, we would have $\alpha \approx 0.264$.
But we can do even better!



This would give $\alpha = 0.25$.
Using these we get good decompositions for $|\mathrm{cat}_k\rangle$ with $k \leq 6$.

Phase gadgets are cat states

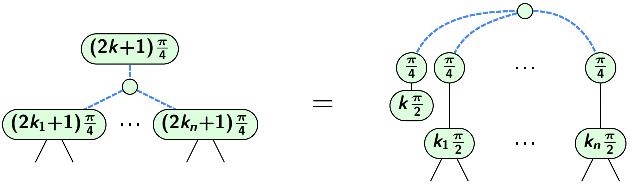
We find cat states as phase gadgets:



So n-legged phase gadget is $|\mathrm{cat}_{n+1}\rangle$ state.

Phase gadgets are cat states

We find cat states as phase gadgets:



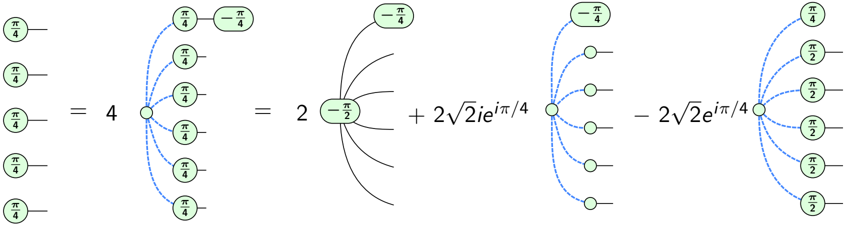
So n-legged phase gadget is $|\mathrm{cat}_{n+1}\rangle$ state.

So as long as there are phase gadgets with ≤ 5 legs, we can use these decompositions.

Partial stabiliser decomp

But what if there are no phase gadgets?

Then we can do the following ‘partial’ decomp:



This trades 5 magic states for 3 terms with 1 magic per term.
So effectively removes 4 magic states.
This is then a 4-to-3 decomp: $\alpha \approx 0.396$.

Full strategy

We are hence looking for the following things to decompose:

1. a phase gadget with 3 legs ($\alpha = 0.25$),
2. a phase gadget with 5 legs ($\alpha \approx 0.264$),
3. a phase gadget with 4 legs ($\alpha \approx 0.317$),
4. a phase gadget with 2 legs ($\alpha = 1/3 \approx 0.333$),
5. any 5 T-spiders ($\alpha \approx 0.396$).

So how well does all this work?

Asymptotic benefit

- Worst case: no T-like phases killed during simplification.

Asymptotic benefit

- Worst case: no T-like phases killed during simplification.
- Then every diagram only needs constant number of rewrites.

Asymptotic benefit

- Worst case: no T-like phases killed during simplification.
- Then every diagram only needs constant number of rewrites.
- Size of diagrams are $O(k)$, so a rewrite costs $O(k^2)$.

Asymptotic benefit

- Worst case: no T-like phases killed during simplification.
- Then every diagram only needs constant number of rewrites.
- Size of diagrams are $O(k)$, so a rewrite costs $O(k^2)$.
- There are $O(2^{\alpha k})$ diagrams, so total cost is $O(2^{\alpha k} k^2)$.

Asymptotic benefit

- Worst case: no T-like phases killed during simplification.
- Then every diagram only needs constant number of rewrites.
- Size of diagrams are $O(k)$, so a rewrite costs $O(k^2)$.
- There are $O(2^{\alpha k})$ diagrams, so total cost is $O(2^{\alpha k} k^2)$.
- (Bravyi et al., 2016) gave $O(2^{\alpha k} k^3)$.
- Benefit comes from preventing 'double work': we 'partially evaluate' the stabilisers by simplifying the diagrams.

Actual benefit

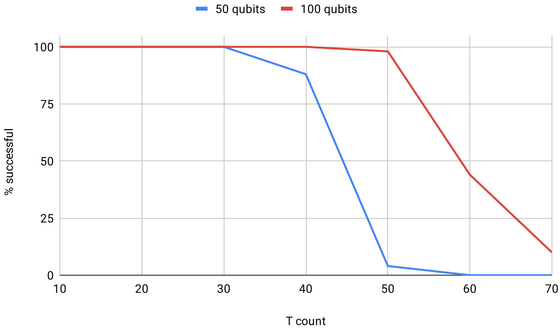
We benchmarked our method on two families of circuits:

- 50- and 100-qubit random Clifford+T circuits built out of Pauli exponentials.
- 50-qubit hidden-shift circuits (type of CCZ circuit).

We are sampling from the output distribution
(using strong simulation).

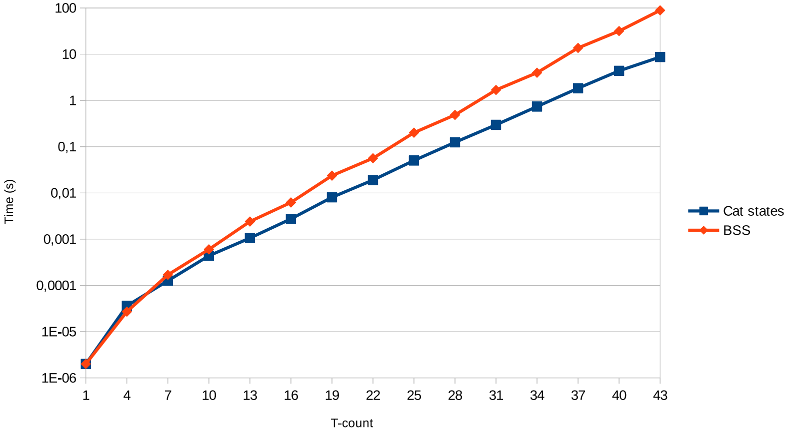
Code is implemented in `quizx`, a Rust port of `PyZX`.

Benchmark: Clifford+T



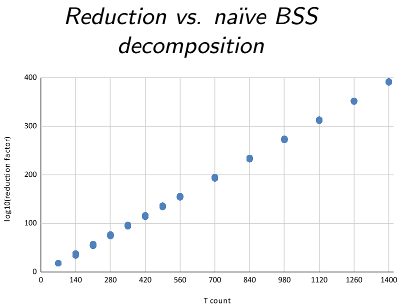
Percentage of random 50- and 100-qubit circuits of a given T-count that were successfully sampled in under 5 minutes. For each T-count 50 random circuits were generated.

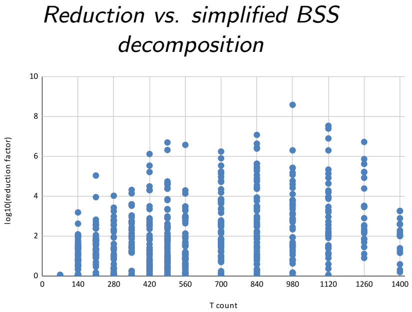
Benchmark: Clifford+T cat-decomp comparison



Runtime of random 20-qubit Clifford+T circuit simulations (avg of 10 runs per T-count).

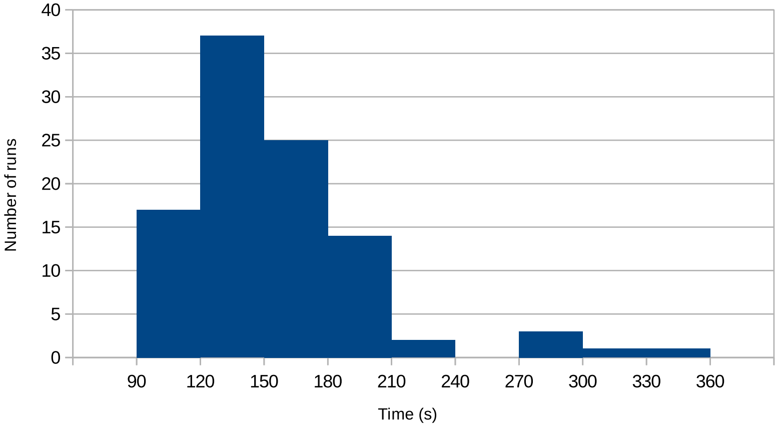
Benchmark: hidden-shift circuit term reduction





Reduction in term count on 50-qubit hidden shift circuits vs. naïve BSS decomposition (left) and BSS decomposition after single ZX-simplification (right).

Benchmark: hidden-shift 50-qubit simulation time



The time distribution of simulating 100 random 50-qubit hidden-shift circuits with T-count 1400 using our new decompositions.

Conclusions

- Using ZX we can greatly speed-up stabiliser rank simulations.
- Especially for structured circuits.
- It allows us to use better decompositions for substructures of diagrams, and to introduce partial stabiliser decompositions.

Conclusions

- Using ZX we can greatly speed-up stabiliser rank simulations.
- Especially for structured circuits.
- It allows us to use better decompositions for substructures of diagrams, and to introduce partial stabiliser decompositions.

Moral of the story: optimisation and simulation are not separate. They are two sides of the same coin.

Conclusions

- Using ZX we can greatly speed-up stabiliser rank simulations.
- Especially for structured circuits.
- It allows us to use better decompositions for substructures of diagrams, and to introduce partial stabiliser decompositions.

Moral of the story: optimisation and simulation are not separate. They are two sides of the same coin.

Future work:

- Use more diagram optimisations.
- Find heuristics for picking good spiders to decompose.
- Approximate simulation and better weak simulation.
- stabiliser *extent* methods.
- Use *quantum measurement w/o computing marginals* technique.

Thank you for your attention!

Further reading:

- Kissinger & vdW. *Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions*. arXiv: 2109.01076
- Kissinger, Vilmart & vdW. *Classical simulation of quantum circuits with partial and graphical stabiliser decompositions*. arXiv: 2202.09202
- Qassim, Pashayan, Gosset. *Improved upper bounds on the stabilizer rank of magic states*. arXiv: 2106.07740
- Bravyi, Gosset. *Improved classical simulation of quantum circuits dominated by Clifford gates*. arXiv: 1601.07601