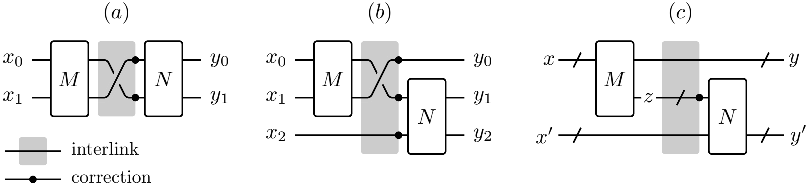
Modular quantum signal processing with gadgets

[2309.16665]

Zane Rossi

joint with Jack Ceroni and Isaac Chuang



October 3, 2023

Quantum algorithms: tough to design and interpret

Despite beautiful work in QPLs [1, 2], we still reason at gate level, most of the time

What can quantum programming *techniques* & *abstractions* learn from their classical cousins?

Quantum algorithm design

Overview

Pinpoint strengths of algorithmic class: QSP

Argue *function-first* algorithmic design with above is (1) desirable but (2) not immediately possible

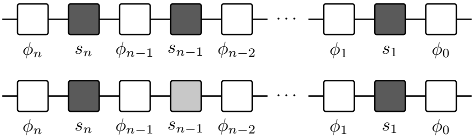
Turn a desire into a formal problem

Show solution utility, efficiency, functional programming flavor, and introduce package

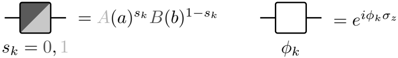
QSP: functional interpretability

QSP $\left\{ \begin{array}{l} \Phi = \{\phi_0, \phi_1, \cdots, \phi_n\} \\ s = 00\cdots0 \end{array} \right. \longmapsto$

M-QSP $\left\{ \begin{array}{l} \Phi = \{\phi_0, \phi_1, \cdots, \phi_n\} \\ s \in \{0, 1\}^n \end{array} \right. \longmapsto$



Circuit map:
$\Phi \in \mathbb{R}^{n+1}, s \in \{0, 1\}^n \mapsto$ q. gates. $\Bigg\{$



Space- and query-efficient algorithms for spectral mapping; unifies most quantum algorithms [3, 4]

The temptation

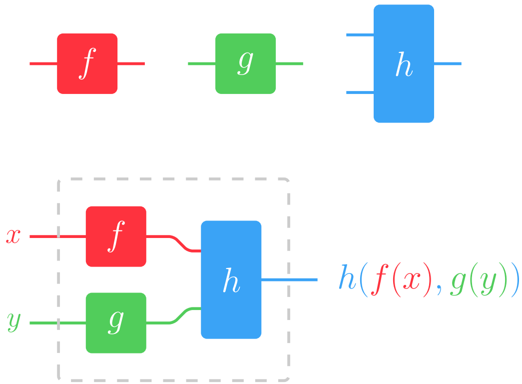
QSP and QSVT achieve* polynomial transforms

$$\langle 0|U(\Phi, x)|0\rangle = P(x)$$

Given protocols achieving $f(x)$, $g(y)$, and $h(z, w)$, it *feels* like we should be able to achieve

$$h \circ (f, g) \equiv h(f(x), g(y)).$$

Moreover, the composite protocol ought to be able to use $f(x)$ and $g(y)$ as contiguous subroutines


f
g
h
x
y
f
g
h
h(f(x), g(y))

The problem

QSP encodes privileged basis: expects X rotations as input but does not produce them as output

Even worse lifting to QSVT or with many oracles

QSP protocols aren't first class functions

Why bother

Modularity and reusability in coherent setting

QSP is great: space efficient, infinity norm dependent, strong numerical footing ...

Multivariable analogues of QSP are *hard*, while multivariable functions (comparators, control-flow, loss) are ubiquitous [5]

A note on QSVT

Commuting operators and spectral mapping

Anything we do in SU(2) is done in preserved QSVT subspaces. Let $\Pi, \tilde{\Pi}$ proj. and $A = \tilde{\Pi} U \Pi$, then

$$A = \sum_{i=1}^{d_{\min}} \xi_i |\tilde{\psi}_i\rangle\langle\psi_i| \mapsto P^{SV}(A) \equiv \sum_{i=1}^{d_{\min}} P(\xi_i)|\tilde{\psi}_i\rangle\langle\psi_i|,$$

$$U = \cdots \oplus \bigoplus_{\xi_i \neq 0,1} \begin{bmatrix} \xi_i & \sqrt{1-\xi_i^2} \\ \sqrt{1-\xi_i^2} & -\xi_i \end{bmatrix}_{\tilde{\mathcal{H}}_i}^{\mathcal{H}_i} \oplus \cdots,$$

Gadgets 📟

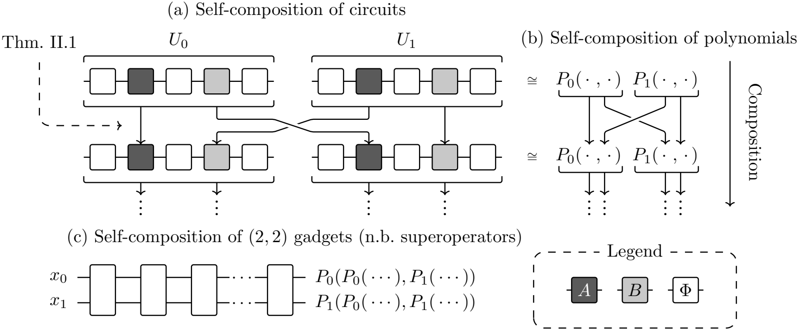
An (a, b) gadget is a unitary* superoperator:

$$\bigotimes_{k\in[a]} e^{i\theta_k\sigma_x} \mapsto \bigotimes_{j\in[b]} e^{i\phi_j\sigma_z} e^{i\psi_j\sigma_x} e^{-i\phi_j\sigma_z},$$

over θ in known range. Each of b output legs achieves $\cos\psi_j$, often polynomial in $\cos\theta_k, k \in [a]$.

Functionally interpretable boxes: a inputs, b outputs

Gadget Rosetta Stone



Thm 1: linking gadgets 🔗

Let $\varepsilon, \delta > 0$, $\mathfrak{G}$ an (a, b) gadget and (Ξ, S) an atomic (c, d) gadget, where $\Xi \equiv \{\Phi_0, \dots, \Phi_{d-1}\}$ and $S \equiv \{s_0, \dots, s_{d-1})$. Suppose $\mathfrak{G}$ achieves

$$F(x) \equiv \{f_0(x_0, \cdots, x_{a-1}), f_1(x_0, \cdots, x_{a-1}), \dots, f_{b-1}(x_0, \cdots, x_{a-1})\}$$

over $x \in [-1, 1]^{\times a}$, and (Ξ, S) achieves

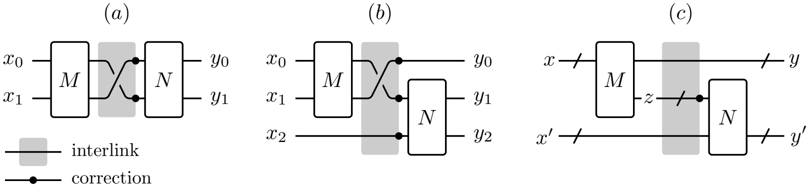
$$G(y) \equiv \{g_0(y_0, \cdots, y_{c-1}), g_1(y_0, \cdots, y_{c-1}), \dots, g_{d-1}(y_0, \cdots, y_{c-1})\}$$

over $y \in [-1, 1]^{\times c}$. Let $\mathfrak{I} = (B, C, W)$ an interlink. Then, there exists a gadget $\mathfrak{G}'$ which ε-approximately achieves

$$H(x, y') \equiv \bigcup_{k \in [d]} g_k \left(\bigcup_{j \in B} f_{W(j)}(x_0, \dots, x_{a-1}) \cup \bigcup_{k \notin C} y_k \right) \cup \bigcup_{k \notin B} f_k(x_0, \dots, x_{a-1})$$

over $(F(x), y') \in \mathcal{D}$, where y' is the subset of y_k such that $k \notin C$ and $\mathcal{D}$ is a domain determined by the correction protocol used.

$\mathfrak{G}'$ uses description of (Ξ, S) and $\widetilde{\mathcal{O}}(d|\Xi|_\infty \zeta)$ black-box calls to result of running $\mathfrak{G}$.



Generally, $\zeta = \widetilde{O}(\text{polylog}(\varepsilon^{-1})\text{poly}(\delta^{-1}))$ for $\delta = \mathcal{O}(1)$. Note $|\Xi|_\infty$ is max length of elem in Ξ.

Thm 2: gadget-poly equivalence ⚖️

Let $\mathcal{L}(x_1, \ldots, x_n)$ polynomials achievable by atomic gadgets over $x_1, \ldots, x_n$. Suppose $P(x_1, \ldots, x_n)$ has degree D and can be split into a tower of $m = \mathcal{O}(\log(D))$ interlinked polynomials,

$$P = P^{(m-1)} \circ_{\mathfrak{I}_{m-2}} \left(P^{(m-2)} \circ_{\mathfrak{I}_{m-3}} \circ \left(\cdots \left(P^{(1)} \circ_{\mathfrak{I}_0} P^{(0)}\right) \cdots\right)\right)$$

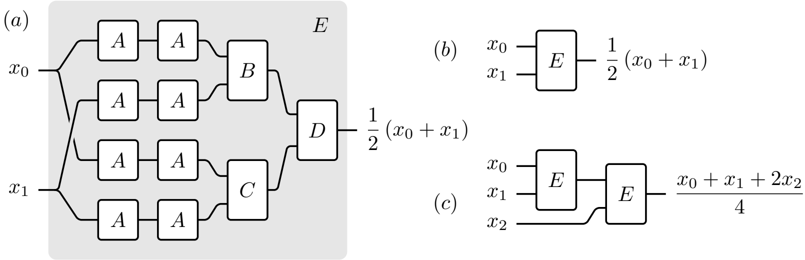
such that $P_i^{(j)} \in \mathcal{L}(x_1, \ldots, x_n)$ for all i, j. If the P_k are separated from $\{0, \pm 1\}$ by $\delta \in \mathcal{O}(1)$ for all k over $\mathcal{D}_k$. Then, there exists an assemblage of m atomic, snappable gadgets which ε-approximately achieves P on $\mathcal{D}_k$ with aggregate query cost $\widetilde{O}(\text{poly}(D)\,\text{polylog}(\varepsilon))$.

Algebraic structure	Manipulation	Algebraic form	Reference
QSP/M-QSP	Polynomial application	$x_0 \mapsto P(x_0)$,	Thm. D.1
	Multivariable variant†	$x_0, \ldots, x_n \mapsto P(x_0, \ldots, x_n)$,	Thm. D.2
Polynomial ring‡	Addition	$x_0, x_1 \mapsto (x_0 + x_1)/2$,	Thm. IV.3
	Multiplication	$x_0, x_1 \mapsto x_0 x_1$,	Thm. IV.2
	Scalar multiplication	$x_0 \mapsto \min(\alpha x_0, 1), \alpha \in \mathbb{R}$,	Ex. IV.4/5
Composition monoid	Polynomial composition	$f(x_0), g(x_0) \mapsto (f \circ g)(x_0)$,	Thm. III.1
General gadgets	Gadget linking	$\mathfrak{G}, \mathfrak{G}' \mapsto \mathfrak{G}\mathfrak{I}\mathfrak{G}'$,	Thm. III.1

Provided examples

- Negation/inversion (Ex. IV.1)
- Angle sum/difference (Ex. IV.2)
- Affine shift (Ex. IV.7)
- Step function (Ex. IV.8)
- General mean (Ex. IV.10)
- Bandpass function (Ex. IV.11)
- Majority vote (Ex. IV.12)
- Functional interpolation (Ex. IV.13)
- Chebyshev inverse (Thm. IV.1)

Sum gadget



A: square root gadget, B, C: angle sum/difference gadgets, D: product gadgets.

Prior work

Cascaded classical filters for sharpening. [6, 7]

Recursively defined quantum (computational) subroutines. [8, 9, 10, 11, 12, 13, 14]

Single-variable self-embedded QSP/QSVT. [15, 16]

Multivariable QSP/QET and LCU-based methods.

Functional programming techniques: quantum and classical. [1, 2, 17]

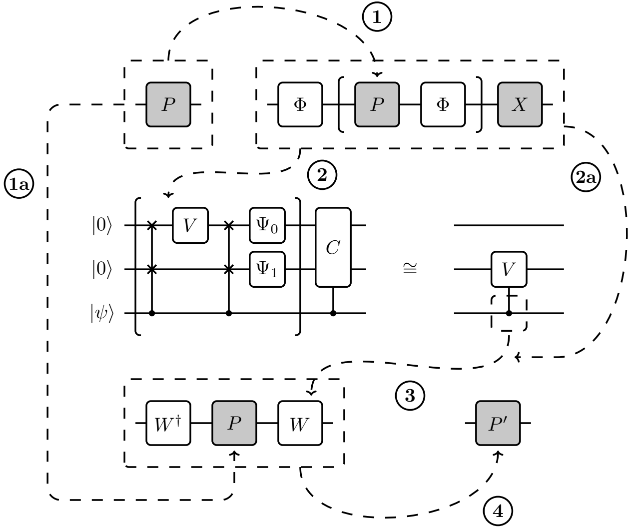
Correction protocol 🩹

Main idea: repeated use of twisted signal to obliviously, approximately cancel its own twist

$$e^{i\phi\sigma_z}e^{i\psi\sigma_x}e^{-i\phi\sigma_z} \mapsto \left\{e^{i\phi\sigma_z}e^{i\psi\sigma_x}e^{-i\phi\sigma_z}, e^{2i\phi\sigma_z}\right\} \mapsto e^{i\psi\sigma_x}.$$

Off diagonal element's phase sent to $\approx \pi/2$

$$\begin{bmatrix} P & iQ\sqrt{1-x^2} \\ * & * \end{bmatrix} \mapsto \begin{bmatrix} P & i\sqrt{1-P^2} \\ * & * \end{bmatrix}$$


1
P
Φ
P
Φ
X
1a
2
2a
|0⟩
|0⟩
|ψ⟩
V
Ψ0
Ψ1
C
≅
V
3
W†
P
W
P′
4

Computing costs 💴

For $P(x)$ close to ± 1, QSP suffers 'gimbal lock'

Take $\delta = \mathcal{O}(1)$, and $P(x) \in [-1+\delta, 1-\delta]^b$ over $x \in [-1+\delta, 1-\delta]^a$: simple bounding box

Cost per correction is $\mathcal{O}(\text{poly}(\delta^{-1})\,\text{polylog}(\varepsilon^{-1}))$

On query/space complexity 💾

An (a, b) gadget has a cost for each output leg, up to desired precision, in queries to each input leg: $a \times b$ cost matrix C

Up to padding rows & cols, C of gadget computed by multiplying C_k of sub-gadgets.

Space/query complexity breaks into cases: single-variable, psd Q, controlled access.

Is this efficient? Or, why not LCU?

Poly-many, poly-size gadgets, log depth.
Maintains QSP's infinity norm scaling.
Poly-logarithmic in inverse precision.
No post-selection required.
Hierarchical, distributed, modular.
Linking high-degree gadgets is where method shines.
Potentially huge space saving (psd Q); coupled to questions in poly decomposition theorems.
Not truly in competition with LCU.

Method	Queries	Space	Norm
[18]	$\text{poly}(r) d^{\text{poly}(r)}$	$\text{poly}(r)[\text{poly}(d) + \text{poly}(s)]$	$\\|\cdot\\|_1$
Ours	d	$s + c$	$\\|\cdot\\|_\infty$

Table: Asymptotic query and space complexity, as well as relevant norm. Here d is generalized degree, r is number of variables, and s is qubits needed for block encodings. Note the case that $c = 0$ is well-understood.

Functional programming 🔧

Semantics and syntax

Monad: monoid in the category of endofunctors

$$\texttt{unit} : T \to M\ T,$$
$$\texttt{bind} : (M\ T, T \to M\ U) \to M\ U,$$

Unwrapped type is scalar value (or X rotation), wrapped is QSP output; this work constructs `bind`

Bonus: attribute grammar & language of gadgets

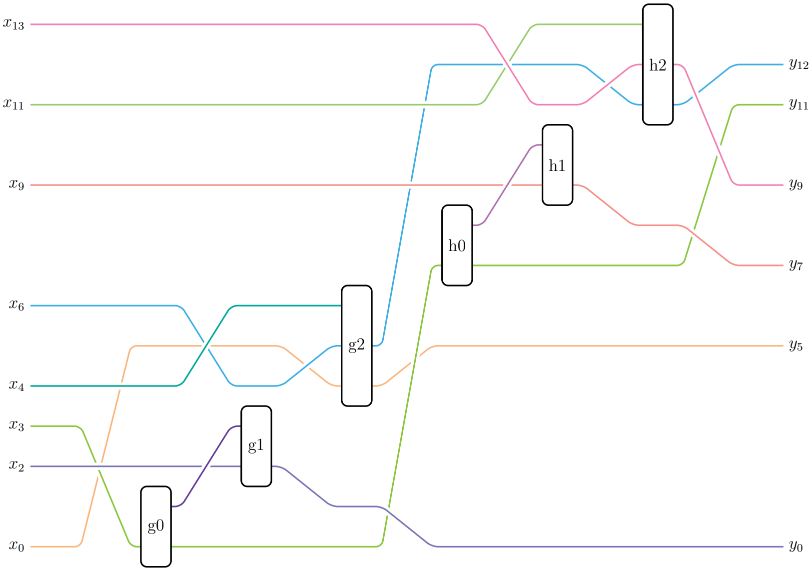
Python package 📦

gadget_init.py UNREGISTERED

```python
xi_0 = [[0, 0, 0, 0], [0, 0, 0, 0]]
xi_1 = [[0, 0, 0, 0], [0, 0, 0, 0]]

s_0 = [[0, 1, 0], [1, 0, 1]]
s_1 = [[0, 1, 0], [1, 0, 1]]

# Create two (2, 2) gadgets named "g0" and "g1".
g0 = AtomicGadget(2, 2, "g0", xi_0, s_0)
g1 = AtomicGadget(2, 2, "g1", xi_0, s_1)

# Link first output of g0 to first input of g1.
linking_guide = [(("g0", 0), ("g1", 0))]
a0 = g0.wrap_gadget()
a1 = g1.wrap_gadget()
a2 = a0.link_assemblage(a1, linking_guide)
```

Line 15, Column 43 Tab Size: 4

github.com/ichuang/pyqsp/tree/beta


x_{13}
x_{11}
x_9
x_6
x_4
x_3
x_2
x_0
g0
g1
g2
h0
h1
h2
y_{12}
y_{11}
y_9
y_7
y_5
y_0

To take home 🏡

Efficient, function-first quantum algorithm design

Coherent use of QSP protocols as QSP oracles with functional interpretability

Readymade repo at `ichuang/pyqsp/tree/beta`

Formal syntax and semantics, coupled to deep lit in (Q)PLs: compilation, verification [1, 19, 20]

Ancilla-free gadgets: minimal functional description?

[1] Peter Selinger. Towards a quantum programming language. *Math. Struct.*, 14(4):527–586, 2004.

[2] Peter Selinger. Towards a semantics for higher-order quantum computation. In *Proceedings of the 2nd International Workshop on Quantum Programming Languages, TUCS General Publication*, volume 33, pages 127–143. Citeseer, 2004.

[3] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019.

[4] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4), dec 2021.

[5] Zane M. Rossi and Isaac L. Chuang. Multivariable quantum signal processing (M-QSP): prophecies of the two-headed oracle. *Quantum*, 6:811, Sep 2022.

[6] J Kaiser and R Hamming. Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(5):415–422, 1977.

[7] Tapio Saramaki. Design of FIR filters as a tapped cascaded interconnection of identical subfilters. *IEEE Transactions on Circuits and Systems*, 34(9):1011–1029, 1987.

[8] Lov K Grover. Fixed-point quantum search. *Phys. Rev. Lett.*, 95(15):150501, 2005.

[9] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. Fixed-point quantum search with an optimal number of queries. *Phys. Rev. Lett.*, 113(21):210501, 2014.

[10] Guang Hao Low, Theodore J Yoder, and Isaac L Chuang. Optimal arbitrarily accurate composite pulse sequences. *Phys. Rev. A*, 89(2):022341, 2014.

[11] Jonathan A Jones. Nested composite NOT gates for quantum computation. *Phys. Lett. A*, 377(40):2860–2862, 2013.

[12] Sami Husain, Minaru Kawamura, and Jonathan A Jones. Further analysis of some symmetric and antisymmetric composite pulses for tackling pulse strength errors. *J. Magn. Reson.*, 230:145–154, 2013.

[13] F. M. Toyama, S. Kasai, W. van Dijk, and Y. Nogami. Matched-multiphase Grover algorithm for a small number of marked states. *Phys. Rev. A*, 79:014301, Jan 2009.

[14] Elica Kyoseva and Nikolay V Vitanov. Arbitrarily accurate passband composite pulses for dynamical suppression of amplitude noise. *Phys. Rev. A*, 88(6):063410, 2013.

[15] Zane M. Rossi and Isaac L. Chuang. Semantic embedding for quantum algorithms. *arXiv preprint, arXiv:2304.14392*, 2023.

[16] Kaoru Mizuta and Keisuke Fujii. Recursive quantum eigenvalue/singular-value transformation: Analytic construction of matrix sign function by Newton iteration. *arXiv preprint, arXiv:2304.13330*, 2023.

[17] Simon J. Gay. Quantum programming languages: Survey and bibliography. *Math. Struct.*, 16(4):581–600, 2006.

[18] Yonah Borns-Weil, Tahsin Saffat, and Zachary Stier. A quantum algorithm for functions of multiple commuting Hermitian matrices. *arXiv preprint, arXiv:2302.11139*, 2023.

[19] Donald E Knuth. Semantics of context-free languages. *Mathematical systems theory*, 2(2):127–145, 1968.

[20] Philip Wadler. Monads for functional programming. In *Advanced Functional Programming: First International Spring School on Advanced Functional Programming Techniques Båstad, Sweden, May 24–30, 1995 Tutorial Text 1*, pages 24–52. Springer, 1995.