

# PQC Function Evaluation

## Carnegie Vacation Scholarship

David Amorim

Week 4  
(22/07/2024 - 26/07/2024)

# Aims for the Week

The following aims were set at the last meeting (22/07/2024):

## Change Input Layer Structure

Improve the connectivity of input layers. Each input qubit should ideally control each target qubit at some point in the network.

## Fix Parameters

Add the option to keep parameters fixed for each type of network layer.

## Improve Loss Function

Develop a distance measure taking into account digital encoding. Either incorporate this into weights for an existing loss function or define a new loss function on this basis.

# Glossary

Acronym	Meaning
CL	convolutional layer
AA-CL	all-to-all convolutional layer
NN-CL	neighbour-to-neighbour convolutional layer
IL	input layer
SAM	sign-adjusted mismatch

Table 1: Acronyms used in the following.

Variable	Meaning
$n$	input register size
$m$	target register size
$L$	number of network layers
$\Psi_{H23}$	phase function from Hayes 2023

Table 2: Variables used in the following.

# Useful Metrics

- The following metrics are useful when testing and comparing QCNN performance:

- Let  $M_j$  be the **mismatch** between  $|j\rangle |\Psi(j)\rangle$  and  $\hat{Q}_\Psi |j\rangle |0\rangle$
- Then define

$$\mu \equiv \text{Mean}(M_j) \quad (1)$$

$$\sigma \equiv \text{STDEV}(M_j) \quad (2)$$

- Consider additional figures of merit to explore the **link between mismatch and successful phase extraction**:

- Define the **error on the normalisation** post phase extraction:

$$\epsilon \equiv 1 - |\langle \psi | \psi \rangle_{\text{final}}|^2 \quad (3)$$

- Define the mean absolute **error on the extracted phase function**,  $\tilde{\Psi}$ , compared to the target function,  $\Psi$ :

$$\chi \equiv \text{Mean}(|\tilde{\Psi}(j) - \Psi(j)|) \quad (4)$$

- Finally define the **super-metric**

$$\Omega \equiv (\mu + \sigma + \epsilon + \chi)^{-1} \quad (5)$$

maybe a handy summary table... ... waffle about linking variables to evaluate performance... talk about "phase extraction problem" ....  
**FRAME THIS AS A CENTRAL THEME!!**

- Let  $M_j$  be the mismatch between  $|j\rangle |\Psi(j)\rangle$  and  $\hat{Q}_\Psi |j\rangle |0\rangle$
- Denote by  $|\psi\rangle_{\text{final}}$  the state vector post phase extraction
- Define  $\tilde{\Psi}$  as the extracted phase function, in contrast to the target function,  $\Psi$

	Definition	Description	Ideal Value
$\mu$	$\text{Mean}(M_j)$	mean mismatch	0
$\sigma$	$\text{STDEV}(M_j)$	mismatch STDEV	0
$\epsilon$	$1 -  \langle \psi   \psi \rangle_{\text{final}} ^2$	normalisation error	0
$\chi$	$\text{Mean}( \tilde{\Psi}(j) - \Psi(j) )$	phase function error	0
$\Omega$	$(\mu + \sigma + \epsilon + \chi)^{-1}$	super-metric	$\infty$

**Table 3:** Metrics introduced to quantify QCNN performance

# Table of Contents

① Changing Input Layer Structure

② Fixing Parameters

③ Improving the Loss Function

④ Next Steps

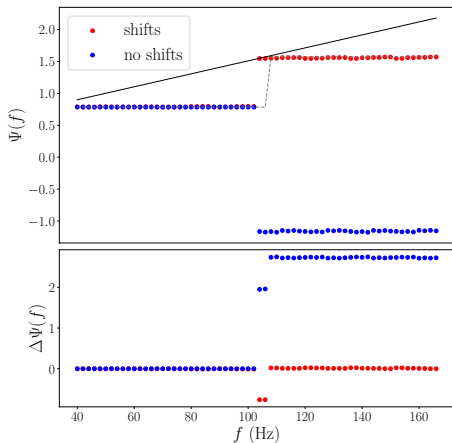
# Changing Input Layer Structure

- Previously, the  $j$ th input qubit controlled an operation on the  $j$ th target qubit (with wrap-around for  $n > m$ )
- An optional **shift parameter**,  $s$ , has now been added so that the  $j$ th input qubit controls an operation on the  $j + s$ th target qubit
- This shift parameter is incremented for each successive IL
- The QCNN is padded with additional ILs to ensure that the number of ILs is  $\geq m$
- Thus, **each input qubit now controls an operation on each target qubit** at some point in the QCNN
- Note that ILs still alternate between control states 0 and 1

# Changing Input Layer Structure

	shifts	no shifts
$\mu$	3.4e-2	3.2e-2
$\sigma$	1.4e-1	1.5e-1
$\epsilon$	2.0e-2	7.5e-2
$\chi$	3.2e-2	1.3e-0
$\Omega$	4.46	0.63

**Table 4:** Comparing metrics for  $\Psi(f) \sim f$  ( $L = 6$ ,  $m = 3$ , 600 epochs, SAM)



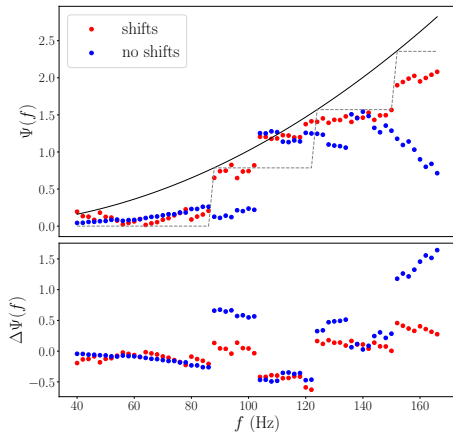
**Figure 1:** Effects of shifted ILs for  $\Psi(f) \sim f$  ( $L = 6$ ,  $m = 3$ , 600 epochs, SAM)



# Changing Input Layer Structure

	shifts	no shifts
$\mu$	1.9e-1	2.4e-1
$\sigma$	1.2e-1	1.5e-1
$\epsilon$	2.2e-1	4.7e-1
$\chi$	1.9e-1	4.3e-1
$\Omega$	1.39	0.78

**Table 5:** Comparing metrics for  $\Psi(f) \sim f^2$  ( $L = 6$ ,  $m = 3$ , 600 epochs, SAM)



**Figure 2:** Effects of shifted ILs for  $\Psi(f) \sim f^2$  ( $L = 6$ ,  $m = 3$ , 600 epochs, SAM)

# Changing Input Layer Structure

- The data for 'no shifts' was obtained by setting  $s = 0$  for all ILs instead of incrementing  $s$
- This should be equivalent to last week's circuit structure
- However, the 'no shifts' results are significantly worse than the results shown last week (???)
- Thus, the improvements due to the new IL structure are somewhat exaggerated
- Nonetheless, **increased IL connectivity clearly leads to improved performance**

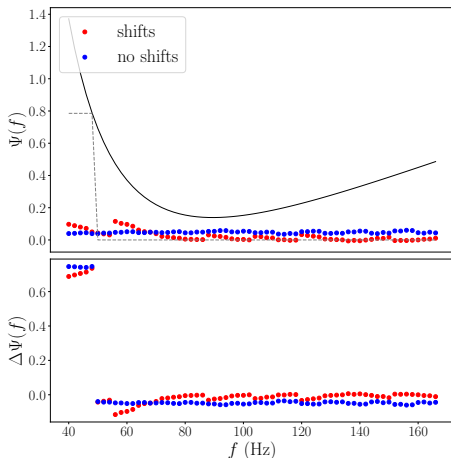


Figure 3: Effects of shifted ILs for  $\Psi_{H23}$  ( $L = 6$ ,  $m = 3$ , 600 epochs, SAM)

# Table of Contents

① Changing Input Layer Structure

② Fixing Parameters

③ Improving the Loss Function

④ Next Steps

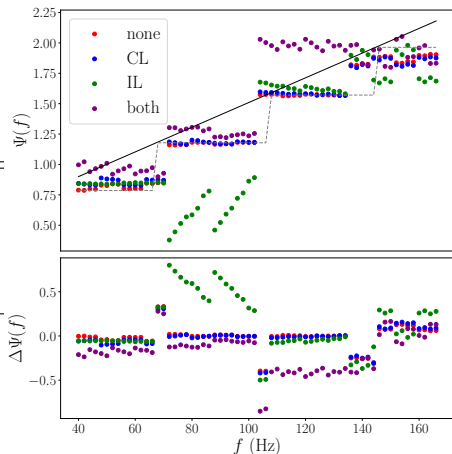
# Fixing Parameters

- Implemented the option to **fix parameters for each layer type**
- This means that each instance of a layer type (IL, AA-CL, NN-CL) uses the same set of parameters
- This significantly **reduces the number of trainable parameters** at large  $L$
- Surprisingly, reducing the parameter space **produces no noticeable speed-up** (so-called qiskit primitives, i.e. the sampler, take up roughly 95% of the computational time)

# Fixing Parameters

	none	CL	IL	both
$\mu$	1.3e-1	1.5e-1	4.5e-1	4.4e-1
$\sigma$	2.2e-1	2.1e-1	2.6e-1	1.6e-1
$\epsilon$	4.4e-2	1.5e-1	4.9e-1	5.8e-1
$\chi$	6.8e-2	8.0e-2	2.4e-1	2.3e-1
$\Omega$	2.14	1.68	0.70	0.71

**Table 6:** Comparing metrics for  $\Psi(f) \sim f$  ( $L = 6$ ,  $m = 4$ , 600 epochs, SAM)

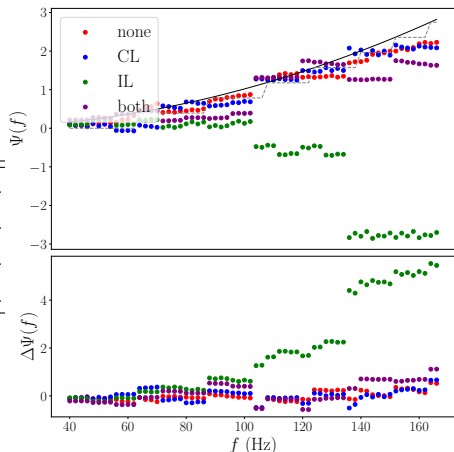


**Figure 4:** Effects of fixing parameters ILs for  $\Psi(f) \sim f$  ( $L = 6$ ,  $m = 4$ , 600 epochs, SAM)

# Fixing Parameters

	none	CL	IL	both
$\mu$	2.3e-1	3.5e-1	5.8e-1	5.3e-1
$\sigma$	1.3e-1	1.6e-1	2.5e-1	2.0e-1
$\epsilon$	2.7e-1	4.4e-1	4.3e-1	3.3e-1
$\chi$	1.7e-1	1.8e-1	1.9e-0	3.6e-1
$\Omega$	1.26	0.88	0.32	0.71

**Table 7:** Comparing metrics for  $\Psi(f) \sim f^2$  ( $L = 6$ ,  $m = 4$ , 600 epochs, SAM)



**Figure 5:** Effects of fixing parameters for  $\Psi(f) \sim f^2$  ( $L = 6$ ,  $m = 4$ , 600 epochs, SAM)

# Fixing Parameters

- Legend for the plots on the previous slide:
  - 'none' : no parameters fixed
  - 'CL': only CL parameters fixed
  - 'IL': only IL parameters fixed
  - 'both': all parameters fixed
- Evidently, keeping parameters fixed leads to (slightly, for 'CL', or drastically, for 'IL' and 'both') **worse performance**
- This is likely due to a reduction of the search space
- Note that 'IL' as well as 'both' lead to **incomplete phase extraction** and hence somewhat meaningless results
- Thus, especially taking into account the equivalent computational times, **not keeping parameters fixed yields better results**
- These results suggest a particular importance of ILs compared to CLs

# Table of Contents

① Changing Input Layer Structure

② Fixing Parameters

③ Improving the Loss Function

④ Next Steps



# Preliminary Definitions

- Consider a computational basis state,  $|k\rangle$ , of the **combined** input-register-target-register **system**
- The state  $|k\rangle$  is associated with a **bit string**  $k = \{0, 1\}^{n+m}$
- Denote by  $[k]_n$  and  $[k]_m$  the bit strings of length  $n$  and  $m$ , respectively, associated with each of the registers and write their **concatenation** as

$$k \equiv [k]_n \diamond [k]_m \quad (6)$$

- A general state of the two-register system is then written as

$$|z\rangle = \sum_{k=0}^{2^{n+m}-1} z_k |k\rangle \quad (7)$$

and referred to via its **coefficients**  $z_k$

# Preliminary Definitions

- When training in superposition, the **desired state** of the system is

$$|y\rangle = \sum_{j=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |j\rangle_i |\Psi(j)\rangle_t, \quad (8)$$

where the subscripts  $i$  and  $t$  indicate basis states of the input and target registers, respectively

- This state  $|y\rangle$  can be written in terms of the **combined basis**  $\{|k\rangle\}$  via

$$y_k = \begin{cases} \frac{1}{\sqrt{2^n}} & \text{if } k = [k]_n \diamond \Psi([k]_n) \\ 0 & \text{else} \end{cases} \quad (9)$$

- Further denote the **output state** produced by the QCNN by  $|x\rangle$ , with associated coefficients  $x_k$

# SAM and Beyond

- Recall the definition of **SAM**:

$$\text{SAM}(|x\rangle, |y\rangle) = \left| 1 - \sum_k x_k y_k \right| \quad (10)$$

- By construction, this is closely related to the **mismatch**

$$\text{M}(|x\rangle, |y\rangle) = 1 - \left| \sum_k x_k y_k \right| \quad (11)$$

- While effective, SAM's **fundamental flaw** is that it does not directly take into account the amplitudes  $x_k$  for  $k$  where  $y_k = 0$  (i.e. where  $[k]_m \neq \Psi([k]_n)$ )

# SAM and Beyond

- Consider  $k = a$  and  $k = b$  with  $[k]_m \neq \Psi([k]_n)$  for both and

$$\left| [a]_m - \Psi([a]_n) \right| \geq \left| [b]_m - \Psi([b]_n) \right| \quad (12)$$

- To improve performance, the loss function should punish a non-zero  $x_a$  more than a non-zero  $x_b$  which is not the case for SAM
- This could be achieved via a **weighted mismatch**,

$$\text{WIM}(|x\rangle, |y\rangle) = \left| 1 - \sum_k \tilde{w}_k x_k y_k \right|, \quad (13)$$

where  $\tilde{w}_k \in \mathbb{R}_+$  are appropriate weights

- It was discussed at the last meeting to base the weights on

$$w_k = \sum_{\substack{l=0, \\ l \neq [k]_m}}^{2^m-1} \left| x_{[k]_n \diamond l} \right| \left| l - \Psi([k]_n) \right|, \quad (14)$$

with the  $\tilde{w}_k$  obtained from the  $w_k$  via normalisation and smoothing

- Implementing this **proved to be ineffective**, with no improvement on SAM (see later)
- This raises broader questions about the **feasibility of WIM**

# The Limits of WIM

- SAM is effective as it learns to maximise the  $x_k$  for  $k$  with  $y_k \neq 0$  without regard for the remaining  $x_k$
- It effectively optimises over a **reduced set of states**
- This is, presumably, why it outperforms more global loss functions, which directly take into account all coefficients, like  $L_1$  loss
- Thus, it seems that SAM's '**fundamental flaw**', of disregarding most  $x_k$ , is really the **basis of its success**
- Adding weights to SAM (WIM) likely is insufficient to alter its fundamental dynamic (or will alter it detrimentally)
- Therefore, **WIM will be abandoned for the foreseeable future**

# Introducing WILL

- To improve on SAM, it could be beneficial to return to a loss function which more directly takes into account all  $x_k$  (e.g.  $L_1$  loss)
- More generally, define  $L_p$  loss as

$$LL_p(|x\rangle, |y\rangle) = \left( \sum_k |x_k - y_k|^p \right)^{1/p} \quad (15)$$

- As discussed, **computational basis states are not equidistant** for phase encoding: their distance depends on the value they encode on the input and target registers
- A **weighted  $L_p$  loss (WILL)** can factor in an appropriate distance measure for the state space:

$$WILL_{p,q} = \left( \sum_k |x_k - y_k|^p \left| [k]_m - \Psi([k]_n) \right|^q \right)^{1/p} \quad (16)$$

# Testing WILL

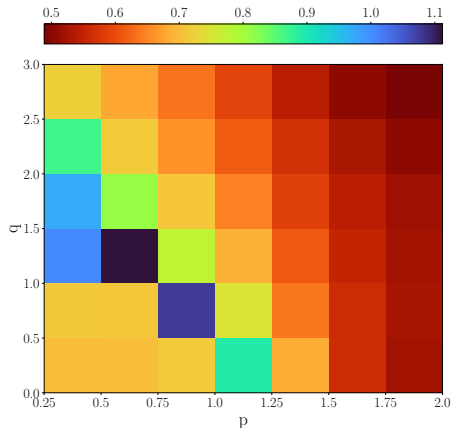


Figure 6: Comparing  $\Omega$  for various  $p, q$  ( $L = 6, m = 3, 600$  epochs,  $\Psi(f) \sim f$ )

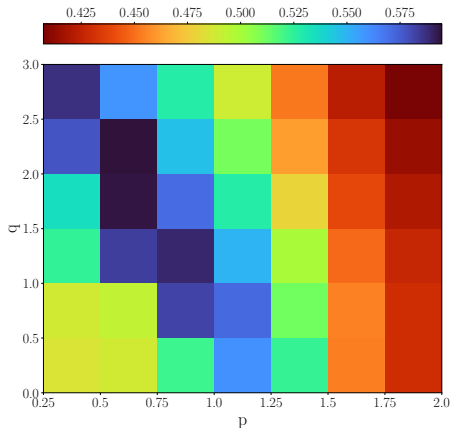


Figure 7: Comparing  $\Omega$  for various  $p, q$  ( $L = 6, m = 3, 600$  epochs,  $\Psi(f) \sim f^2$ )



# Testing WILL

- These tests already reveal WILL's **subpar performance**

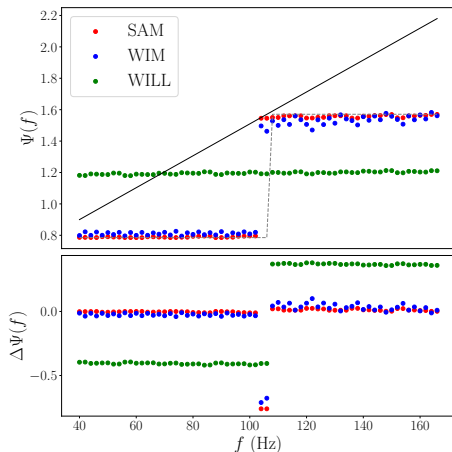
	$\Psi(f) \sim f$	$\Psi(f) \sim f^2$	$\Psi_{H23}$
$(p, q)$	$(0.75, 1.50)$	$(0.75, 2.50)$	$(0.5, 1.50)$
$\mu$	3.0e-1	4.7e-1	7.1e-2
$\sigma$	1.8e-2	1.5e-1	2.1e-1
$\epsilon$	1.9e-1	3.8e-1	2.7e-2
$\chi$	3.9e-1	6.8e-1	7.0e-2
$\Omega$	1.11	0.60	2.63

**Table 8:** WILL metrics for different  $\Psi$  using the best identified  $p, q$  combination ( $L = 6$ ,  $m = 4$ , 600 epochs)

# Comparing SAM, WIM, and WILL

	SAM	WIM	WILL
$\mu$	<b>3.4e-2</b>	6.0e-2	3.0e-1
$\sigma$	1.4e-1	<b>1.1e-1</b>	1.7e-2
$\epsilon$	<b>1.9e-2</b>	9.2e-2	1.9e-1
$\chi$	<b>3.2e-2</b>	5.1e-2	3.9e-1
$\Omega$	<b>4.46</b>	3.19	1.11

**Table 9:** Comparing loss function metrics for  $\Psi(f) \sim f$  ( $L = 6$ ,  $m = 4$ , 600 epochs)

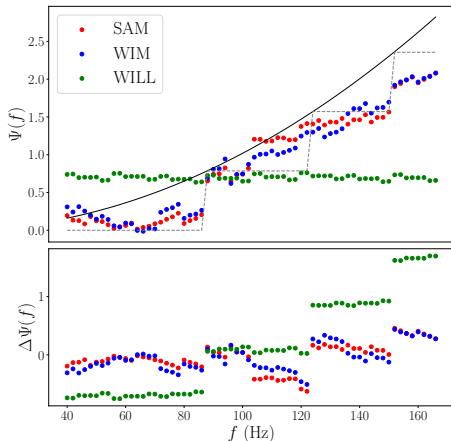


**Figure 8:** Comparing extracted phase functions for  $\Psi(f) \sim f$  ( $L = 6$ ,  $m = 4$ , 600 epochs)

# Comparing SAM, WIM, and WILL

	SAM	WIM	WILL
$\mu$	<b>1.9e-1</b>	2.3e-1	4.7e-1
$\sigma$	1.2e-1	<b>1.0e-1</b>	1.5e-1
$\epsilon$	<b>2.2e-1</b>	4.2e-1	3.8e-1
$\chi$	<b>1.9e-1</b>	2.0e-1	6.8e-1
$\Omega$	<b>1.39</b>	1.05	0.60

**Table 10:** Comparing loss function metrics for  $\Psi(f) \sim f^2$  ( $L = 6$ ,  $m = 4$ , 600 epochs)

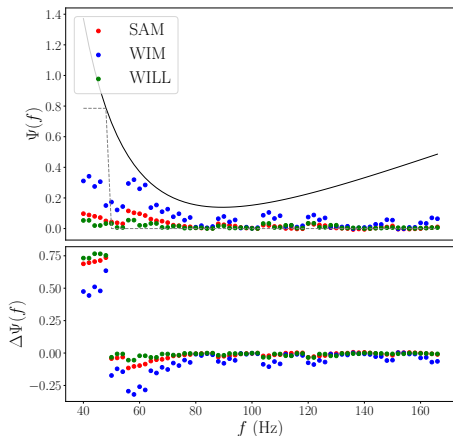


**Figure 9:** Comparing extracted phase functions for  $\Psi(f) \sim f^2$  ( $L = 6$ ,  $m = 4$ , 600 epochs)

# Comparing SAM, WIM, and WILL

	SAM	WIM	WILL
$\mu$	<b>6.8e-2</b>	8.4e-2	7.1e-2
$\sigma$	1.8e-1	<b>1.2e-1</b>	2.1e-1
$\epsilon$	4.5e-2	1.8e-1	<b>2.7e-2</b>
$\chi$	7.4e-2	1.0e-1	<b>7.0e-2</b>
$\Omega$	<b>2.75</b>	2.07	2.63

**Table 11:** Comparing loss function metrics for  $\Psi_{H23}$  ( $L = 6$ ,  $m = 4$ , 600 epochs)



**Figure 10:** Comparing extracted phase functions for  $\Psi_{H23}$  ( $L = 6$ ,  $m = 4$ , 600 epochs)

# Loss Function Conclusion

...attempt to construct a wavefunction that improves upon SAM failed...  
in the process developed formalisation which makes discussion more rigorous and could be helpful for future loss function design...for now: no more ideas on how to improve .... Also identified 5 key metrics for QCNN performance in general ( $\mu, \sigma, \chi, \epsilon, \Omega$ )  
AT THIS POINT MOSTLY WINDOW DRESSING...  
ALSO CALCULATE METRICS FOR LAYER STRUCTURE AND FIXED PARAMETERS??!!

# Table of Contents

① Changing Input Layer Structure

② Fixing Parameters

③ Improving the Loss Function

④ Next Steps

# Next Steps

- formalise and investigate ‘completeness of phase extraction’
- look into adding more ILs compared to CLs ?