PQC Function Evaluation

Carnegie Vacation Scholarship

David Amorim

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

Table of Contents

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.

Additional Aim: Understanding Phase Extraction

- Phase encoding is based on two operators, $\hat{Q}_\Psi$ and $\hat{R}$, defined via

$$\hat{Q}_\Psi |j\rangle |0\rangle = |j\rangle |\Psi(j)\rangle \tag{1}$$

$$\hat{R} |l\rangle = e^{il} |l\rangle \tag{2}$$

- Encoding proceeds in three steps:

$$|j\rangle |0\rangle \xmapsto{\hat{Q}_\Psi} |j\rangle |\Psi(j)\rangle \xmapsto{\hat{I}\otimes\hat{R}} |j\rangle e^{i\Psi(j)} |\Psi(j)\rangle \xmapsto{\hat{Q}_\Psi^\dagger} e^{i\Psi(j)} |j\rangle |0\rangle , \tag{3}$$

- The phase $e^{i\Psi(j)}$ of the final state is extracted assuming the ancilla register is clear (i.e. in the state $|0\rangle$)
- However, when $\hat{Q}_\Psi$ does not produce an eigenstate of $\hat{I} \otimes \hat{R}$, applying $\hat{Q}^\dagger$ does not generally clear the ancilla
- A flawed implementation of $\hat{Q}_\Psi$ non-trivially affects the extracted phase function

Additional Aim: Understanding Phase Extraction

- To investigate this relationship between $\hat{Q}_\Psi$ and the extracted phase ('phase extraction problem'), make the following definitions:
  - M_j: mismatch between $|j\rangle\,|\Psi(j)\rangle$ and $\hat{Q}_\Psi\,|j\rangle\,|0\rangle$
  - $|\psi\rangle_{\text{final}}$: state vector post phase extraction
  - $\tilde{\Psi}$: extracted phase function (in contrast to the target function, Ψ)
- Based on the above, define five metrics that quantify the quality of $\hat{Q}_\Psi$ as well as of phase extraction:

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

Table 1: Metrics introduced to quantify QCNN performance

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 2: Acronyms used in the following.

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

Table 3: Variables used in the following.

Table of Contents

Changing Input Layer Structure

- Previously, the jth input qubit controlled an operation on the jth target qubit (with wrap-around for $n > m$)
- An optional shift parameter, s, has now been added so that the jth 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
μ	3.4e-2	**3.2e-2**
σ	**1.4e-1**	1.5e-1
ϵ	**2.0e-2**	7.5e-2
χ	**3.2e-2**	1.3e-0
Ω	**4.46**	0.63

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

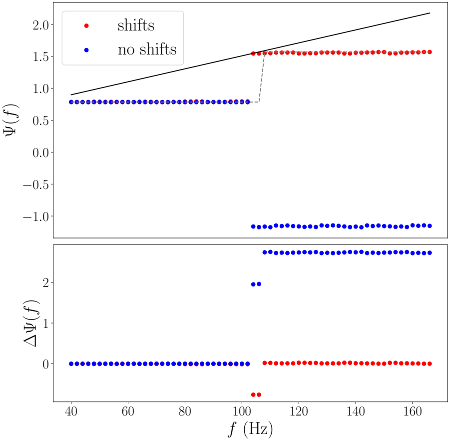

Figure 1: Effects of shifted ILs for $\Psi(f) \sim f$ ($L = 6$, $m = 3$, 600 epochs, SAM)

Changing Input Layer Structure

	shifts	no shifts
μ	**1.9e-1**	2.4e-1
σ	**1.2e-1**	1.5e-1
ϵ	**2.2e-1**	4.7e-1
χ	**1.9e-1**	4.3e-1
Ω	**1.39**	0.78

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

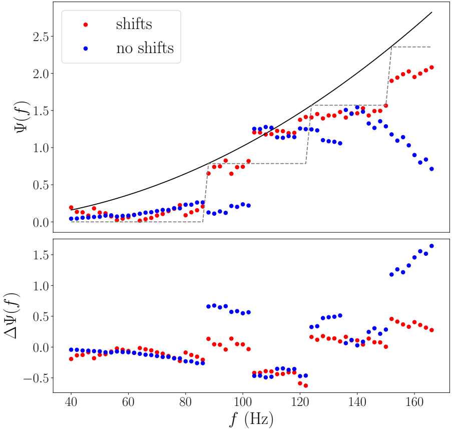

Figure 2: Effects of shifted ILs for $\Psi(f) \sim f^2$ ($L = 6$, $m = 3$,600 epochs, SAM)

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

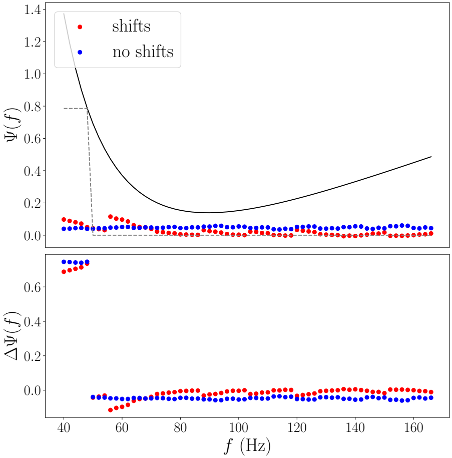

Figure 3: Effects of shifted ILs for Ψ_{H23} ($L = 6$, $m = 3$, 600 epochs, SAM)

Table of Contents

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
μ	**1.3e-1**	1.5e-1	4.5e-1	4.4e-1
σ	2.2e-1	**2.1e-1**	2.6e-1	1.6e-1
ϵ	**4.4e-2**	1.5e-1	4.9e-1	5.8e-1
χ	**6.8e-2**	8.0e-2	2.4e-1	2.3e-1
Ω	**2.14**	1.68	0.70	0.71

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

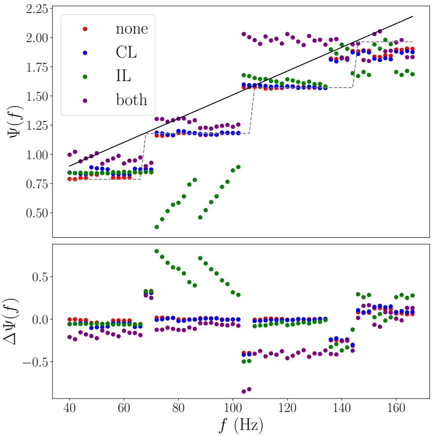

Figure 4: Effects of fixing parameters ILs for $\Psi(f) \sim f$ ($L = 6$, $m = 4$, 600 epochs, SAM)

Fixing Parameters

	none	CL	IL	both
μ	**2.3e-1**	3.5e-1	5.8e-1	5.3e-1
σ	**1.3e-1**	1.6e-1	2.5e-1	2.0e-1
ϵ	**2.7e-1**	4.4e-1	4.3e-1	3.3e-1
χ	**1.7e-1**	1.8e-1	1.9e-0	3.6e-1
Ω	**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)

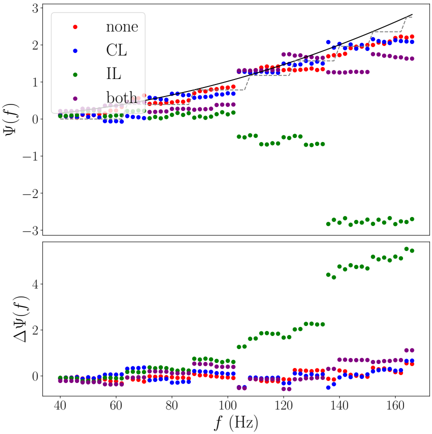

Figure 5: Effects of fixing parameters for $\Psi(f) \sim f^2$ ($L = 6$, $m = 4$, 600 epochs, SAM)

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, likely due to a reduction of the search space
- Note that '*IL*' as well as '*both*' lead to high ϵ 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

Formalisation

- 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

$$k \equiv [k]_n \diamond [k]_m \tag{4}$$

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

$$|z\rangle = \sum_{k=0}^{2^{n+m}-1} z_k |k\rangle \tag{5}$$

and referred to via its coefficients z_k

Formalisation

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

$$|y\rangle = \sum_{j=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |j\rangle_i |\Psi(j)\rangle_t \,, \tag{6}$$

  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

$$y_k = \begin{cases} \frac{1}{\sqrt{2^n}} & \text{if } k = [k]_n \diamond \Psi([k]_n) \\ 0 & \text{else} \end{cases} \tag{7}$$

- 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:

$$\text{SAM}(|x\rangle, |y\rangle) = \left|1 - \sum_k x_k y_k\right| \tag{8}$$

- By construction, this is closely related to the mismatch

$$\text{M}(|x\rangle, |y\rangle) = 1 - \left|\sum_k x_k y_k\right| \tag{9}$$

- 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

$$\Big|[a]_m - \Psi([a]_n)\Big| > \Big|[b]_m - \Psi([b]_n)\Big| \tag{10}$$

- 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 (WIM),

$$\text{WIM}(\left|x\right\rangle, \left|y\right\rangle) = \left|1 - \sum_k \tilde{w}_k x_k y_k\right|, \tag{11}$$

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

SAM and Beyond

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

$$w_k = \sum_{\substack{l=0, \\ l \neq [k]_m}}^{2^m-1} \left| x_{[k]_n \diamond l} \right| \left| l - \Psi([k]_n) \right|, \tag{12}$$

  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: is adding weights sufficient to alter SAM's fundamental dynamic of neglecting x_k fpr k with $y_k = 0$?

Introducing WILL

- To improve on SAM, it could instead be beneficial to return to a loss function which more directly takes into account all x_k
- Define $\mathsf{L_p}$ loss as

$$\mathsf{LL_p}(|x\rangle, |y\rangle) = \left(\sum_k |x_k - y_k|^p \right)^{1/p} \tag{13}$$

- 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 $\mathsf{L_p}$ loss (WILL) can factor in an appropriate distance measure for the state space:

$$\mathsf{WILL_{p,q}} = \left(\sum_k \left| x_k - y_k \right|^p \left| [k]_m - \Psi([k]_n) \right|^q \right)^{1/p} \tag{14}$$

Testing WILL

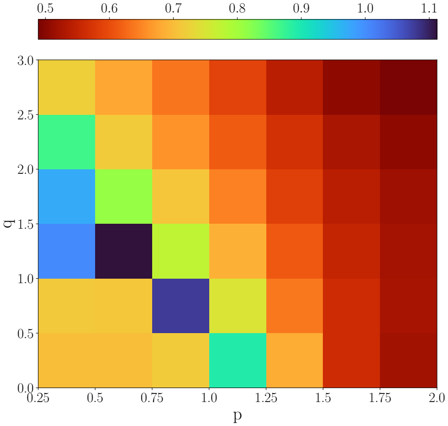

Figure 6: Comparing Ω for various p, q ($L = 6$, $m = 3$, 600 epochs, $\Psi(f) \sim f$)

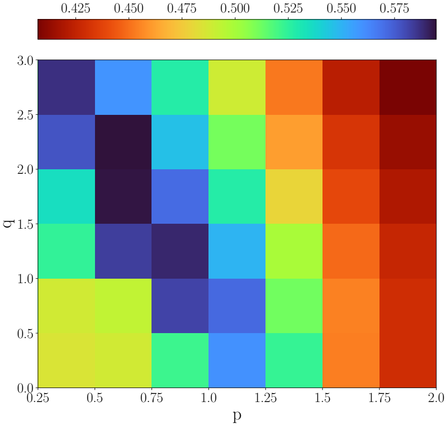

Figure 7: Comparing Ω for various p, q ($L = 6$, $m = 3$, 600 epochs, $\Psi(f) \sim f^2$)

Testing WILL

- These tests already reveal WILL's subpar performance

	$\Psi(f) \sim f$	$\Psi(f) \sim f^2$	Ψ_{H23}
(p, q)	(0.75, 1.50)	(0.75, 2.50)	(0.5, 1.50)
μ	3.0e-1	4.7e-1	7.1e-2
σ	1.8e-2	1.5e-1	2.1e-1
ϵ	1.9e-1	3.8e-1	2.7e-2
χ	3.9e-1	6.8e-1	7.0e-2
Ω	1.11	0.60	2.63

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

Comparing SAM, WIM, and WILL

	SAM	WIM	WILL
μ	**3.4e-2**	6.0e-2	3.0e-1
σ	1.4e-1	**1.1e-1**	1.7e-2
ϵ	**1.9e-2**	9.2e-2	1.9e-1
χ	**3.2e-2**	5.1e-2	3.9e-1
Ω	**4.46**	3.19	1.11

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

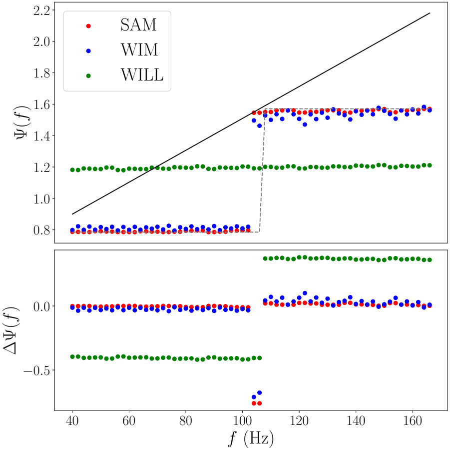

Figure 8: Comparing extracted phase functions for $\Psi(f) \sim f$ ($L = 6$, $m = 4$, 600 epochs)

Comparing SAM, WIM, and WILL

	SAM	WIM	WILL
μ	**1.9e-1**	2.3e-1	4.7e-1
σ	1.2e-1	**1.0e-1**	1.5e-1
ϵ	**2.2e-1**	4.2e-1	3.8e-1
χ	**1.9e-1**	2.0e-1	6.8e-1
Ω	**1.39**	1.05	0.60

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

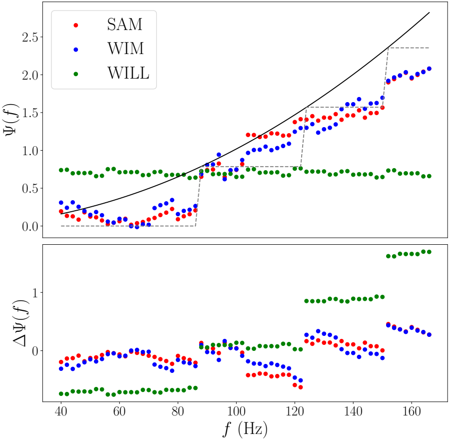

Figure 9: Comparing extracted phase functions for $\Psi(f) \sim f^2$ ($L = 6$, $m = 4$, 600 epochs)

Comparing SAM, WIM, and WILL

	SAM	WIM	WILL
μ	**6.8e-2**	8.4e-2	7.1e-2
σ	1.8e-1	**1.2e-1**	2.1e-1
ϵ	4.5e-2	1.8e-1	**2.7e-2**
χ	7.4e-2	1.0e-1	**7.0e-2**
Ω	**2.75**	2.07	2.63

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

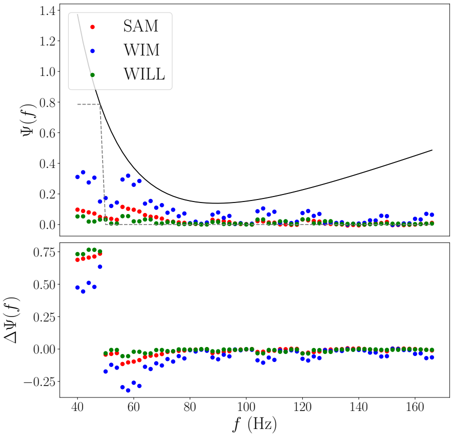

Figure 10: Comparing extracted phase functions for Ψ_{H23} ($L = 6$, $m = 4$, 600 epochs)

Loss Function Conclusion

- The attempts to improve upon SAM via WIM and WILL have failed
- No further *ansatz* to design a new loss function could be identified
- Further QCNN improvements might have to come from something other than the loss function

Table of Contents

Exploring Correlations

- The following investigates correlations between the metrics μ, σ, χ, and ϵ in an attempt to better understand the relationship between phase extraction and $\hat{Q}_\Psi$
- This is done by analysing the outputs of $N = 267$ different QCNNs with various different configurations (i.e. L, m, loss functions, Ψ,...)
- Key questions to answer are:
  - Is a low normalisation error ϵ an indicator of a good fit, i.e. low χ?
  - Is a low mean mismatch μ a guarantee of low ϵ, χ?
  - What is the role of σ?

Exploring Correlations

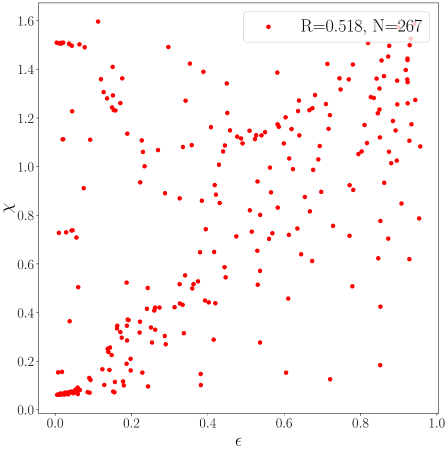

Figure 11: Correlation between ϵ and χ.

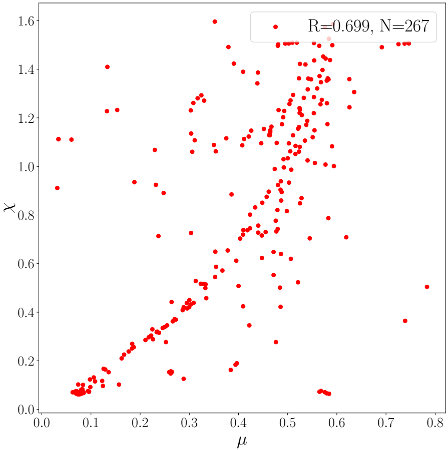

Figure 12: Correlation between μ and χ.

Exploring Correlations

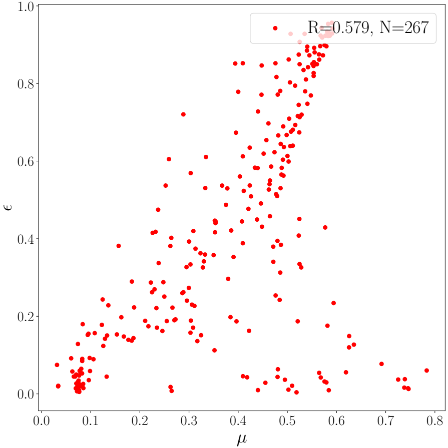

Figure 13: Correlation between μ and ϵ.

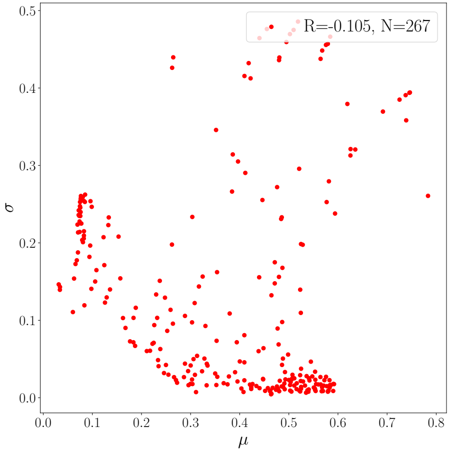

Figure 14: Correlation between μ and σ.

Exploring Correlations

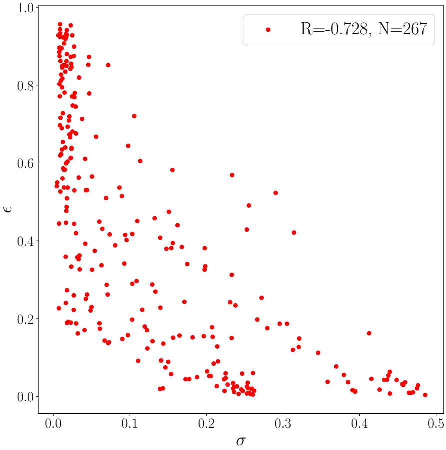

Figure 15: Correlation between σ and ϵ.

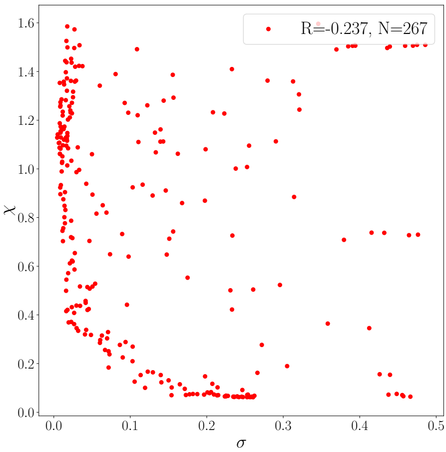

Figure 16: Correlation between σ and χ.

Table of Contents

Next Steps

- Try to formalise and further investigate the relationship between $\hat{Q}_\Psi$ and the extracted phase (phase extraction problem)
- Look into adding more ILs compared to CLs
- Look into barren plateau mitigation techniques
- ...?