Doctorate of Philosophy

Schrödinger's Catwalk

Brian Flynn

University of Bristol

December, 2020

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LISTINGS

ACRONYMS

AIC	Akaike information criterion. 84, 86
AICC	Akaike information criterion corrected. 84
BF	Bayes factor. 25–27, 33, 35, 48, 51, 55, 57, 61, 63, 64, 80, 81, 87, 89–91, 93–95
BFEER	Bayes factor enhanced Elo ratings. 80, 81, 93, 94, 96
BIC	Bayesian information criterion. 86
CLE	classical likelihood estimation. 13
EDH	experiment design heuristic. 18–23, 26, 35, 45, 55, 56, 83
ES	exploration strategy. 27–31, 33–35, 40, 43, 45, 49, 51, 61, 65, 67, 75, 79, 105, 108
ET	exploration tree. 28, 29, 31, 33–35, 45, 47, 75
FH	Fermi-Hubbard. 58
FP	false negatives. 78
FP	false positives. 78
GA	genetic algorithm. iii, 34, 67, 71, 74, 75, 80, 82, 83, 88, 90, 93, 96–98
GES	genetic exploration strategy. 67, 78, 93, 94, 97
HPD	high particle density. 18
IQLE	interactive quantum likelihood estimation. 13, 14, 93, 94
LTL	log total likelihood. 16
ML	machine learning. 6, 26, 27, 78
MS	model search. 27–29, 31, 35, 43, 45
MVEE	minimum volume enclosing ellipsoid. 18

NV nitrogen-vacancy. 9
NVC nitrogen-vacancy centre. 14

OF objective function. iii, 67, 68, 74, 75, 78, 80, 81, 83, 84, 88, 91–93

PGH particle guess heuristic. 19, 20, 45

QHL quantum Hamiltonian learning. 8–14, 16, 18, 20–22, 25–27, 31–35, 40, 43, 48, 55, 56, 75, 93, 94, 105
QL quadratic loss. 17
QLE quantum likelihood estimation. 13, 32
QMLA Quantum Model Learning Agent. ii, iii, vii, 8, 13, 24, 25, 27–31, 34, 35, 40, 43–49, 51, 52, 60, 61, 63–65, 67, 72, 74–77, 83, 87, 88, 90, 91, 93, 94, 96–98, 105

SMC sequential monte carlo. 11–13, 15, 18, 19, 22, 31

TLTL total log total likelihood. 16, 25–27, 35, 84, 91
TN true negatives. 78
TP true positives. 78

GLOSSARY

Jordan Wigner transformation (JWT)	Jordan Wigner transformation . 60, 61, 64
Loschmidt echo (LE)	Quantum chaotic effect described. . 14
chromosome	A single candidate in the space of valid solutions to the posed problem in a genetic algorithm. . 68
gene	Individual element within a chromosome. . 68
hyperparameter	Variable within an algorithm that determines how the algorithm itself proceeds.. 11
instance	a single implementation of the QMLA algorithm. iii, 48, 94, 97, 98, 105
likelihood	Value that represents how likely a hypothesis is.. 10, 13, 15, 18, 31, 33, 34, 36, 88
model	The mathematical description of some quantum system. 24
model space	Abstract space containing all descriptions (within defined constraints such as dimension) of the system as models. 29
probe	Input probe state, $\|\psi\rangle$, which the target system is initialised to, before unitary evolution. plural. 13, 15, 18–22, 54
results directory	Directory to which the data and analysis for a given run of QMLA are stored. . 49
run	collection of QMLA instances. iii, vii, 48, 49, 64, 65, 94, 96, 98, 105
spawn	Process by which new models are generated by combining previously considered models.. 29
success rate	. 48, 49

term	Individual constituent of a model, e.g. a single operator within a sum of operators, which in total describe a Hamiltonian. . 24
volume	Volume of a parameter distribution's credible region.. 18, 55, 56, 63
win rate	. 48, 49

1 GENETIC ALGORITHMS

The QMLA framework lends itself easily to the family of optimsation techniques called *evolutionary algorithms*, where individuals, sampled from a population of candidates, are considered, in generations, as solutions to the given problem, and iterative generations aim to efficiently search the available population, by mimicing biological evolutionary mechanisms [52]. In particular, we develop a exploration strategy (ES) which incorporates an GA in the generation of models; GAs are a subset of evolutionary algorithms where candidate solutions are expressed as strings of numbers representing some configuration of the system of interest [53]. Here we will first introduce the concept of an GA, before describing the adaptations which allow us to build a genetic exploration strategy (GES).

1.1 GENETIC ALGORITHM DEFINITION

GAs work by assuming a given problem can be optimised, if not solved, by a single candidate among a fixed, closed space of candidates, called the population, $\mathcal{P}$. A number of candidates are sampled at random from $\mathcal{P}$ into a single *generation*, and evaluated through some OF, which assesses the fitness of the candidates at solving the problem of interest. Candidates from the generation are then mixed together to produce the next generation's candidates: this *crossover* process aims to combine only relatively strong candidates, such that the average candidates' fitness improve at each successive generation, mimicing the biological mechanism whereby the genetic makeup of offspring is an even mixture of both parents. The selection of strong candidates as parents for future generations is therefore imperative; in general parents are chosen according to their fitness as determined by the OF. Buidling on this biological motivation, much of the power of GAs comes from the concept of *mutation*: while offspring retain most of the genetic expressions of their parents, some elements are mutated at random. Mutation is crucial in avoiding local optima of the OF landscape by maintaining diversity in the examined subspace of the population.

Pseudocode for a generic GA is given in Algorithm 1, but we can also informally define the procedure as follows. Given access to the population, $\mathcal{P}$,

1. Sample N_m candidates from the population at random
   (a) call this group of candidates the first generation, μ.
2. Evaluate each candidate $\gamma_j \in \mu$.
   (a) each γ_j is assigned a fitness, g_j
   (b) the fitness is computed through an objective function acting on the candidate, $g(\gamma_j)$.
3. Map the fitnesses of each candidate, $\{g_j\}$, to selection probabilities for each candidate, $\{s_j\}$

   (a) e.g. by normalising the fitnesses, or by removing some poorly-performing candidates and then normalising.

4. Generate the next generation of candidates
   (a) Reset $\mu = \{\}$
   (b) Select pairs of parents, p_1, p_2, from μ
      i. Each candidate's probability of being chosen is given by their s_j
   (c) Cross over p_1, p_2 to produce children candidates, c_1, c_2.
      i. mutate c_1, c_2 according to some random probabilistic process
      ii. keep c_i only if it is not already in μ, to ensure N_m unique candidates are tested at each generation.
   (d) until $|\mu| = N_m$, iterate to step (b).
5. Until the N_g^{th} generation is reached, iterate to step 2..
6. The strongest candidate on the final generation is deemed the solution to the posed problem.

Candidates are manifested as *chromosomes*, i.e. strings of fixed length, whose entries, called *genes*, each represent some element of the system. In general, genes can have continuous values, although usually, and for all purposes in this thesis, genes are binary, capturing simply whether or not the gene's corresponding feature is present in the chromosome.

1.1.1 *Example: knapsack problem*

One commonly referenced combinatorial optimisation problem is the *knacksack problem*: given a set of objects, where each object has a defined mass and also a defined value, determine the set of objects to pack in a knapsack which can support a limited weight, such that the value of the packed objects is maximised. Say there are n objects, we can write the vector containing the values of those objects as $\vec{v}$, and the vector of their weights as $\vec{w}$. We can then represent configurations of object sets as candidate vectors $\vec{\gamma}_j$, whose genes are binary, and simply indicate whether or not the associated object is included in the set. For example, with $n = 6$,

$$\gamma_j = 100001 \Longrightarrow \vec{\gamma}_j = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1), \tag{1.1}$$

indicates a set of objects consisting only of those indexed first and last, with none of the intermediate objects included.

The fitness of any candidate is then given by the total value of that configuration of objects, $v_j = \vec{v} \cdot \vec{\gamma}_j$, but candidates are only admitted[1] if the weight of the corresponding set of objects is less than the capacity of the knapsack, i.e. $\vec{w}_j \cdot \vec{\gamma}_j \leq w_{max}$.

1 Note there are alternative strategies to dealing with candidates who violate the weight condition, such as to impose a penalty within the OF, but for our purposes let us assume we simply disregard violators.

Algorithm 1: Genetic algorithm

Input: $\mathcal{P}$ // Population of candidate models
Input: $g()$ // objective funtion
Input: map_g_to_s() // function to map fitness to selection probability
Input: select_parents() // function to select parents among generation
Input: crossover() // function to cross over two parents to produce offspring
Input: N_g // number of generations
Input: N_m // number of candidates per generation

Output: γ' // strongest candidate

$\mu \leftarrow \text{sample}\,(\mathcal{P}, N_m)$
for $i \in 1, ..., N_g$ **do**
  for $\gamma_j \in \mu$ **do**
    $g_j \leftarrow g(\gamma_j)$ // assess fitness of candidate
  end
  $\{s_j\} \leftarrow$ map_g_to_s($\{g_j\}$) // map fitnesses to normalised selection probability
  $\mu_c = \arg\max_{s_j}\{\gamma_j\}$ // record champion of this generation

  $\mu \leftarrow \{\}$ // empty set for next generation
  while $|\mu| < N_m$ **do**
    $p_1, p_2 \leftarrow$ select_parents($\{s_j\}$) // choose parents based on candidates' s_j
    $c_1, c_2 \leftarrow$ crossover(p_1, p_2) // generate offspring candidates based on parents
    for $c \in \{c_1, c_2\}$ **do**
      if $c \notin \mu$ **then**
        $\mu \leftarrow \mu \cup \{c\}$ // keep if child is new
      end
    end
  end
end
$\gamma' \leftarrow \arg\max_{s_j}\{\gamma_j \in \mu\}$ // strongest candidate on final generation

return γ'

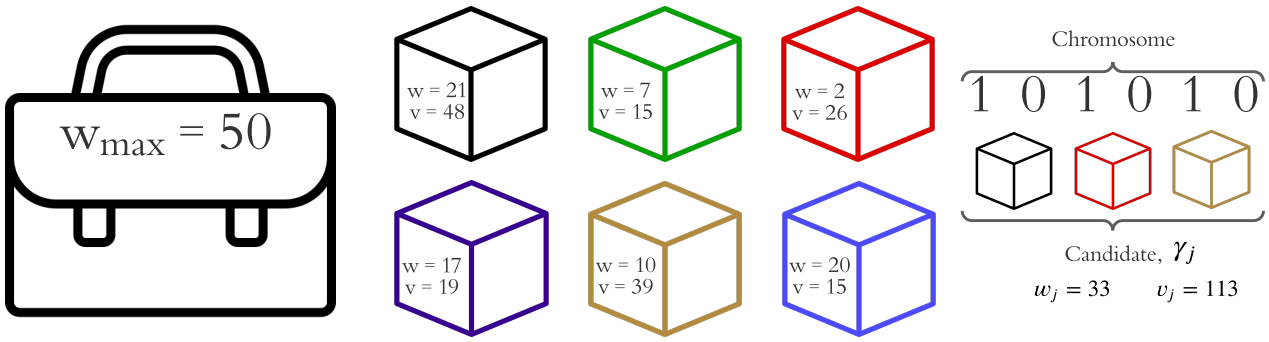


Figure 1.1: Depiction of the knapsack problem. **Left**, A knapsack which can hold any number of objects but is constrained by the total weight it can support, $w_{max} = 50$. **Centre**, A set of objects are available, each with associated weight, w, and value v. The objective is to find the subset of objects which maximise the total value, while not exceeding the capacity of the knapsack. **Right**, An example chromosome, i.e. candidate γ_j, where the bits of the chromosome indicate whether the corresponding object is included, allowing for calculation of the total weight and value of the candidate, w_j, v_j.

For example where each individual object has value < 50 and weight < 25 and $w_{max} = 50$, recalling $\gamma_j = 100001$, say,

$$\vec{v} = (48 \quad 15 \quad 26 \quad 19 \quad 39 \quad 15) \implies v_j = \vec{\gamma}_j \cdot \vec{v} = 48 + 15 = 63; \tag{1.2a}$$

$$\vec{w} = (21 \quad 7 \quad 2 \quad 17 \quad 10 \quad 20) \implies w_j = \vec{\gamma}_j \cdot \vec{w} = 21 + 20 = 41. \tag{1.2b}$$

We can hence assess the fitness of γ_j as 63 and deem it a valid candidate since it does not exceed the weight threshold. We can likewise compute the total weight and value of a series of randomly generated candidates, and deem them valid or not.

Name	Candidate	Value	Weight	Valid
γ_1	110011	117	58	No
γ_2	101010	113	33	Yes
γ_3	011110	99	36	Yes
γ_4	011011	95	39	Yes
γ_5	111000	89	30	Yes
γ_6	010111	88	54	No
γ_7	100010	87	31	Yes
γ_8	110001	78	48	Yes
γ_9	011101	75	46	Yes
γ_{10}	110000	63	28	Yes
γ_{11}	000011	54	30	Yes
γ_{12}	000101	34	37	Yes

Table 1.1: Candidate solutions to the knapsack problem.

The strongest (valid) candidates from Table 1.1 are $101010, 011110$. By spawning from these candidates through a one-point crossover at the midpoint, we get $\gamma_{c_1} = 101110, \gamma_{c_2} = 011010$, from which we can see $v_{c_1} = 132, w_{c_1} = 50$, i.e. by combining two strong candidates we produce the strongest-yet-seen valid candidate.

By repeating this procedure, it is expected to uncover candidates which optimise v_j while maintining $w_j \leq w_{max}$, or at least to produce near-optimal solutions, using far less time/resources than brute-force evaluation of all candidates, which is usually sufficient. For instance, with $n = 100$ objects to consider, there are $2^{100} \approx 10^{30}$ candidates to consider; the most powerful supercomputers in the world currently claim on the order of Exa-FLOPs, i.e. 10^{18} operations per second, of which say ~ 1000 operations are required to test each candidate, meaning 10^{15} candidates can be checked per second in a generous example. This would still require 10^{12} seconds to solve absolutely, so it is reasonable in cases like this to accept *approximately optimal* solutions[2].

1.1.2 *Selection mechanism*

A key subroutine of every GA is the mechanism through which it nominates candidates from generation μ as parents to offsping candidates in $\mu + 1$ [54]. All mechanisms have in common that they act on a set of candidates from the previous generation, where each candidate, γ_j, has

2 Simply put: in machine learning, *good enough* is good enough. We will adopt this philosophy for the remainder of this thesis and life.

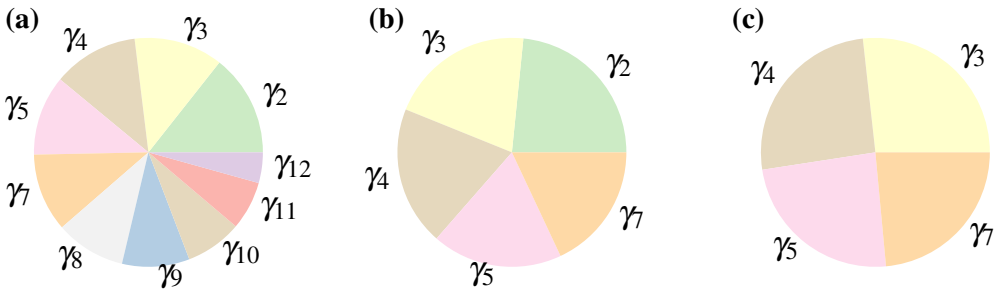


Figure 1.2: Roulette wheels showing selection probability s_i for corresponding candidates γ_i. Colours here only distinguish candidates, they do not encode any information. **b**, The set of potential parents is truncated to include only the strongest five candidates. **a**, All valid candidates are assigned selection probability based on their value in Table 1.1. **c**, After one parent (γ_2) has been chosen, it is removed from the roulette wheel and the remaining candidates' probabilities are renormalised for the selection of the second parent.

been evaluated and has fitness value, g_j. Among the viable schemes for selecting individual parents from the set of candidates, μ are

- Rank selection: candidates are selected with probabilty proportional to their ranking relative to the fitness of contemporary candidates in the same generation.
- Tournament selection: a subset of k candidates are chosen at random from μ, of which the candidate with the highest fitness is taken as the parent.
- Stochastic universal sampling: candidates are sampled proportional to their fitness, but the sampling algorithm is biased to ensure high-fitness candidates are chosen at least once within the generation.

We will only detail the mechanism used later within QMLA, the common fitness proportional selection, known as *roulette selection* [54]. This is a straightforward strategy where we directly map candidates' fitness, g_i to a selection probability, s_i, simply by normalising $\{g_i\}$, allowing us to visualise a roulette wheel of uneven wedges, eachh of which correspond to a candidate. Then we need only conceptually spin the roulette wheel to select the first parent, γ_{p_1}. We then remove γ_{p_1} from the set of potential parents, renormalise the remaining $\{s_i\}$, and spin the wheel again to choose the second parent, γ_{p_2}.

Practically, we repeat the process outlined until the next generation is filled, usually we have $|\mu| = N_m$, and desire that every generation should contain the same N_m candidates, so we repeat the roulette selection $N_m/2$ times per generation, since every pair of parents yield two offspring. It is important that meaningful differences in fitness are reflected by the selection probability, which is difficult to ensure for large N_m, e.g. with ten models, the strongest candidate is only a marginally more probable parent than the worst – this effect is amplified for larger N_m. We therefore wish to reduce the set of potential parents to ensure high quality offspring: we truncate

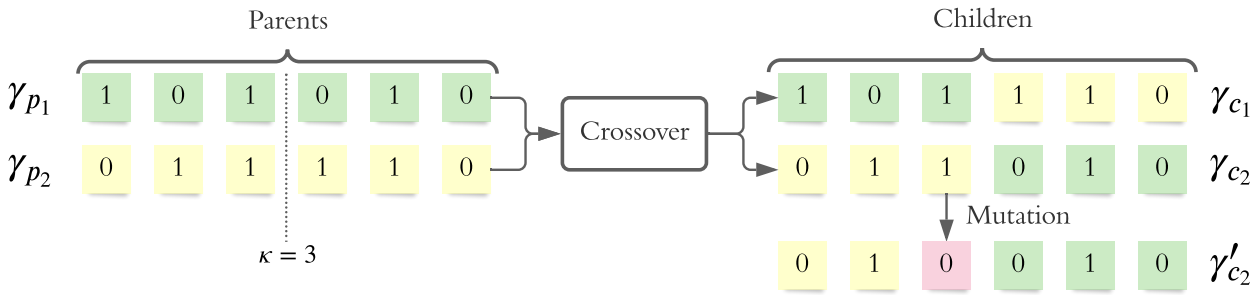


Figure 1.3: Crossover and mutation of chromosomes. Two parents, $\gamma_{p_1}, \gamma_{p_2}$, are nominated from the process in Fig. 1.2. They are then crossed-over via a one-point crossover with crossing point $\kappa = 3$, resulting in children candidates $\gamma_{c_1}, \gamma_{c_2}$. One child chromosome is mutated to yield a new candidate, γ'_{c_2}. The candidates added to the next generation are then $\{\gamma_{c_1}, \gamma'_{c_2}\}$.

μ and retain only the highest-fitness $\frac{N_m}{2}$ models as selectable parents. The roulette selection is shown in Fig. 1.2.

1.1.3 *Reproduction*

When a pair of parents have been nominated by the selection mechanism above, it remains to use those parents to *reproduce*, i.e. to produce offspring which should inherit and improve upon the properties of their parents. Here we use a *one point crossover*, whereby the two parent chromosomes are mixed together to form two offspring, about a single point, κ: for candidates of n genes, the first κ genes of γ_{p_1} are conjoined with the latter $n - \kappa$ genes of γ_{p_2}. Often κ is restricted to the midpoint of the chromosomes, although in general we need not impose this: we will instead consider $\kappa \in \left(\frac{n}{4}, \frac{3n}{4}\right)$, e.g. with $n = 12$, $\kappa \in (3, 9)$. The one-point crossover is shown for $n = 6$ with $\kappa = 3$ in Fig. 1.3, recalling the chromosome structure from Section 1.1.1.

By allowing κ other than the midpoint, we drastically increase the number of combinations of parents available for reproduction. Finally, then, parent selection is done by constructing a database of pairs of potentital parents with all available crossover points, with selection probability given by the product of their individual fitnesses. This is conceptually equivalent to selection via roulette wheel as above. Recalling the fitnesses (values) of Table 1.1, for example:

Parent 1	Parent 2	κ	s_{ij}
γ_2	γ_3	2	$11,187\ (= 113 \times 99)$
γ_2	γ_3	3	$11,187$
γ_2	γ_3	4	$11,187$
γ_2	γ_4	2	$10,735\ (= 113 \times 95)$
γ_2	γ_4	3	$10,735$
γ_2	γ_4	4	$10,735$
		$\vdots$	
γ_5	γ_7	2	$7,743\ (= 89 \times 87)$
γ_5	γ_7	3	$7,743$
γ_5	γ_7	4	$7,743$

Table 1.2: Example of parent selection database. Pairs of parents are selected together, with the (unnormalised) selection probability, s_{ij}, given by the product of the individual candidates' fitnesses. Pairs of parents are repeated in the database for differing κ, and all κ are equally likely.

The GA maintains diversity in the subspace of $\mathcal{P}$ it studies, by *mutating* some of the newly proposed offspring candidates. Again, there are a multitude of approaches for this step [55], but for brevity we only describe those used in this thesis. For each proposed child candidate, γ_c, we probabilistically mutate each gene with some mutation rate r_m: if a mutation occurs, the child is replaced by γ_c'. That is, γ_c' is added to the next generation, and γ_c is discarded. r_m is a *hyperparameter* of the GA: the performance of the algorithm can be optimised by finding the best r_m for a given problem.

1.1.4 *Candidate evaluation*

Within every generation of the GA, each candidate must be evaluated, so that the relative strength of candidates can be exploited in constructing candidates for the next generation. In the example of the knapsack problem used above, candidates were evaluated by the value of their contennts, but also by whether they would fit in the knapsack. Idenitfiyng the appropriate method by which to evaluate candidates is arguably the most important aspect of designing a GA: while the choice of hyperperameters (N_g, N_m, r_m) dictate the efficacy of the search, the lack of an effective metric by which to distinguish candidates would render the procedure pointless. Considerations are hence usually built into the objective function (OF).

Unlike previous aspects of generic GAs, in the context of QMLA, here we must deviate from default mechanisms. Recalling the overarching goal of QMLA, to characterise some black box quantum system, Q, we do not have access to a natural OF. We wish to optimise the modelling of $\hat{H}'$, but assume we do not know the target $\hat{H}_0$, so we can not simply invoke some loss function, for example. Instead, we must devise schemes which exploit the knowledge we *do* have about

each candidate $\hat{H}_j$, which is the primary challenge in building an ES based on a GA. We propose and discuss a number of options in Section 1.3.

Common to all proposed OFs, however, is that candidates should first be trained before evaluation, so that their assessment is based on their actual power in explaining the target system, rather than some initial paramterisation which may not capture their potential. This is a tenet of QMLA: for each candidate $\hat{H}_j(\vec{\alpha}_j)$, we use a subroutine to optimise $\vec{\alpha}_j$, again for this study we rely on quantum Hamiltonian learning (QHL).

1.2 ADAPTATION TO QMLA FRAMEWORK

Ultimately, the conceived role of a GA within QMLA is to generate the sets of models to place on successive branches of the exploration trees (ETs) in **??**. The apparatus for this is to implement an exploration strategy (ES) whose model generation subroutine calls an external GA. Recall from **??**, that we capture the space of available terms as $\mathcal{T}$, i.e. we list – in advance – the feasible terms which may be included in models[3], with $N_t = |\mathcal{T}|$ the number of terms considered. QMLA is then an optimisation algorithm, attempting to find the set $\mathcal{T}'$ which *best* represents the true terms $\mathcal{T}_0$. Note, this does not require idenitifcation of the precise true model to be successful, as insight can be gained from approximate models which capture the physics of the target system. We introduce metrics for success in Section 1.2.2. We recognise the limitations this structure imposes: we can only identify terms which were conceived in advance; this may restrict QMLA's applicability to entirely unknown systems, where such a primitive set can not even be compiled.

The structure of the overall QMLA algorithm, recall **??**, is unchanged. In a genetic exploration strategy (GES):

- models are still grouped in branches, here called generations;
- models are still trained, again through QHL;
- branches are evaluated according the the OF to be described in Section 1.3;
- new models are spawned through the genetic algorithm by selecting pairs of parents for crossover, with the resultant offspring models probabilistically mutated.

We detail the corresponding $\mathrm{generate\_models}$ subroutine in Algorithm 2. We can restate the informal description of GAs, now in the context of QMLA, as

1. Sample N_m models from $\mathcal{P}$ at random
   (a) this is the first generation, μ.
2. Evaluate each model $\hat{H}_j \in \mu$.
   (a) train $\hat{H}_j$ through QHL
   (b) apply the objective function to assign the model's fitness g_j

3 Recall that models impose structure on sets of terms: $\hat{H}_j = \vec{\alpha}_j \cdot \vec{T}_j = \sum_{k \in \mathcal{T}_j} \alpha_k \hat{t}_k$.

3. Map the fitnesses of each model, $\{g_j\}$, to selection probabilities for each model, $\{s_j\}$
   (a) e.g. by normalising the fitnesses, or by removing some poorly-performing models and then normalising.
4. Generate the next generation of models
   (a) Reset $\mu = \{\}$
   (b) Select pairs of parents, $\hat{H}_{p_1}, \hat{H}_{p_2}$, from μ
      i. Each model's probability of being chosen is given by their s_j
   (c) Cross over $\hat{H}_{p_1}, \hat{H}_{p_2}$ to produce children models, $\hat{H}_{c_1}, \hat{H}_{c_2}$.
      i. mutate $\hat{H}_{c_1}, \hat{H}_{c_2}$ according to some random probabilistic process
      ii. keep $\hat{H}_{c_i}$ only if it is not already in μ, to ensure N_m unique models are tested at each generation.
   (d) until $|\mu| = N_m$, iterate to step (b).
5. Until the N_g^{th} generation is reached, iterate to step 2..
6. The strongest model on the final generation is deemed the approximation to the system, $\hat{H}'$.

1.2.1 *Models as chromosomes*

We first need a mapping from models to chromosomes; this is straightforward given the description of chromosomes as binary strings, exemplified in Section 1.1.1. We assign a gene to every term in $\mathcal{T}$, so that candidate models are succinctly represented by bit strings of length N_t. We give an example of the mapping between models and chromosomes in Table 1.3.

1.2.2 *F_1-score*

We need a metric against which to evaluate models, and indeed the entire QMLA procedure. We can gauge the performance of QMLA's model search by the quality of candidate models produced at each generation, so we introduce a mertic to act as proxy for model quality: the F_1-score. In short, $f \in (0, 1)$ indicates the degree to which $\hat{H}_i$ captures the physics of the target system: $f = 0$ indicates that $\hat{H}_i$ shares no terms with $\hat{H}_0$, while $f = 1$ is found uniquely for $\hat{H}_i = \hat{H}_0$. We will define the concept formally next. Note that here we are able to compute f for candidate models because the target $\hat{H}_0$ is simulated, i.e. we know the true terms $\mathcal{T}_0$; this would not be available for a real system with unknown $\hat{H}_0$, but is useful for the analysis of the algorithm itself.

We emphasise that the goal of this work is to identify the *model* which best describes quantum systems, and not to improve on parameter-learning when given access to particular models, since those already exist to a high standard [3, 56]. Therefore we can consider QMLA as a

Model		Chromosome					
	$\vec{T}$	$\hat{\sigma}^x_{(1,2)}$	$\hat{\sigma}^z_{(1,2)}$	$\hat{\sigma}^y_{(2,3)}$	$\hat{\sigma}^x_{(2,3)}$	$\hat{\sigma}^y_{(2,3)}$	$\hat{\sigma}^x_{(2,3)}$
γ_{p_1}	$(\hat{\sigma}^x_{(1,2)} \quad \hat{\sigma}^z_{(1,2)} \quad \hat{\sigma}^y_{(2,3)})$	1	0	1	0	1	0
γ_{p_2}	$(\hat{\sigma}^z_{(1,2)} \quad \hat{\sigma}^y_{(2,3)} \quad \hat{\sigma}^z_{(2,3)})$	0	0	1	0	1	1
γ_{c_1}	$(\hat{\sigma}^x_{(1,2)} \quad \hat{\sigma}^z_{(1,2)} \quad \hat{\sigma}^y_{(2,3)} \quad \hat{\sigma}^z_{(2,3)})$	1	0	1	0	1	1
γ_{c_2}	$(\hat{\sigma}^z_{(1,2)} \quad \hat{\sigma}^y_{(2,3)})$	0	0	1	0	1	0
γ'_{c_2}	$(\hat{\sigma}^z_{(1,2)} \quad \hat{\sigma}^x_{(2,3)} \quad \hat{\sigma}^y_{(2,3)})$	0	0	1	1	1	0

Table 1.3: Mapping between QMLA's models and chromosomes used by a genetic algorithm. Example shown for a three-qubit system with six possible terms, $\hat{\sigma}^w_{i,j} = \hat{\sigma}^w_i \hat{\sigma}^w_j$. Model terms are mapped to binary genes: if the gene registers 0 then the corresponding term is not present in the model, and if it registers 1 the term is included. The top two chromosomes are *parents*, $\gamma_{p_1} = 101010$ (blue) and $\gamma_{p_2} = 001011$ (green): they are mixed to spawn new models. We use a one–point cross over about the midpoint: the first half of γ_{p_1} is mixed with the second half of γ_{p_2} to produce two new children chromosomes, $\gamma_{c_1}, \gamma_{c_2}$. Mutation occurs probabilistically: each gene has a 25% chance of being mutated, e.g. a single gene (red) flipping from $0 \to 1$ to mutate γ_{c_2} to γ'_{c_2}. The next generation of the genetic algorithm will then include $\gamma_{c_1}, \gamma'_{c_2}$ (assuming γ_{c_1} does not mutate). To generate N_m models for each generation, $N_m/2$ parent couples are sampled from the previous generation and crossed over.

classification algorithm, with the goal of classifying whether individual terms $\hat{t}$ from a set of available terms $\mathcal{T} = \{\hat{t}\}$ are helpful in describing data which is generated by $\hat{H}_0$, which has $\mathcal{T}_0$. Candidate models $\hat{H}_i$ then have $\mathcal{T}_i$. We can assess $\hat{H}_i$ using standard metrics used regularly in the machine learning (ML) literature, which simply count the number of terms identified correctly and incorrectly,

- true positives (TP): number of terms in $\mathcal{T}_0$ which are in $\mathcal{T}_i$;
- true negatives (TN): number of terms not in $\mathcal{T}_0$ which are also not in $\mathcal{T}_i$;
- false positives (FP): number of terms in $\mathcal{T}_i$ which are not in $\mathcal{T}_0$;
- false negatives (FP): number of terms in $\mathcal{T}_0$ which are not in $\mathcal{T}_i$.

These concepts allow us to define

- *precision*: how precisely does $\hat{H}_i$ capture $\hat{H}_0$, i.e. if a term is included in $\mathcal{T}_i$ how likely it is to actually be in $\mathcal{T}_0$, Eqn 1.3a;
- *sensitivity*: how sensitive is $\hat{H}_i$ to $\hat{H}_0$, i.e. if a term is in $\mathcal{T}_0$, how likely $\mathcal{T}_i$ is to include it, Eqn. 1.3b.

$$\text{precision} = \frac{TP}{TP + FP} \tag{1.3a}$$

$$\text{sensitivity} = \frac{TP}{TP + FN} \tag{1.3b}$$

Informally, precision prioritises that predicted terms are correct, while sensitivity prioritises that true terms are identified. In practice, it is important to balance these considerations. F_β-score is a measure which balances these, with F_1-score in particular giving them equal importance.

$$F_1 = \frac{2 \times (precision) \times (sensitivity)}{(precision + sensitivity)} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}. \tag{1.4}$$

We give an example of these quantities in Fig. 1.4, where $TP = 3, TN = 4, FP = 1, FN = 2$, giving $precision = 3/4$ and $sensitivity = 3/5$, with a final $f = 0.67$, i.e. the average of the indicators of model quality which we care about.

We adopt F_1-score as an indication of model quality because we are concerned both with precision and sensitivity of output models. We can use F_1-score to measure the success of the algorithm, by recording f for all models in all generations, allowing us to see whether or not the approximation of the system is improving on average.

Of course in realistic cases we can not assume knoweldge of $\mathcal{T}_o$ and therefore cannot compute F_1-score, but it is a useful tool in the development of the GES itself, or in cases where $\hat{H}_0$ is known, such as when the target system is simualated, e.g. in the case of device calibration. Our search for an effective OF can then be guided by seeking the method which correlates most strongly with F_1-score in test-cases.

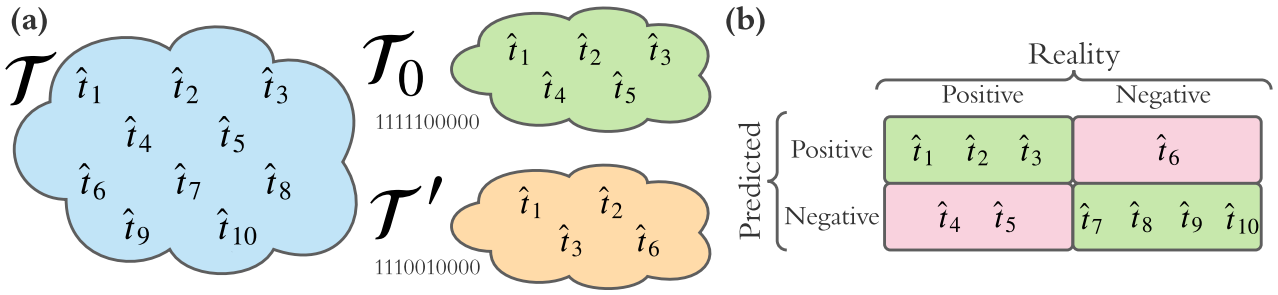


Figure 1.4: Concepts used for classification. **a**, the set of available terms $\mathcal{T}$ containing individual terms $\hat{t}_1$ to $\hat{t}_{10}$. The true model $\hat{H}_0$ is constructed from the set $\mathcal{T}_0$. Suppose a candidate $\hat{H}'$ has the set $\mathcal{T}'$. **b**, the confusion matrix for $\hat{H}'$. Correctly classified terms are true positives and true negatives (green), and incorrectly classified terms are false positives and true negatives (red).

Algorithm 2: ES subroutine: generate_models via genetic algorithm

Input: ν // information about models considered to date

Input: $g(\hat{H}_i)$ // objective function

Output: $\mathbb{H}$ // set of models

$N_m = |\nu|$ // number of models
for $\hat{H}_i \in \nu$ **do**
  $g_i \leftarrow g(\hat{H}_i)$ // model fitness via objective function
end
$r \leftarrow \text{rank}(\{g_i\})$ // rank models by their fitness
$\mathbb{H}_t \leftarrow \text{truncate}(r, \frac{N_m}{2})$ // truncate models by rank: only keep $\frac{N_m}{2}$
$s \leftarrow \text{normalise}(\{g_i\})\ \forall \hat{H}_i \in \mathbb{H}_t$ // normalise remaining models' fitness
$\mathbb{H} = \{\}$ // new batch of chromosomes/models
while $|\mathbb{H}| < N_m$ **do**
  $p_1, p_2 = \text{roulette}(s)$ // use s to select two parents via roulette selection
  $c_1, c_2 = \text{crossover}(p_1, p_2)$ // produce offspring models
  $c_1, c_2 = \text{mutate}(c_1, c_2)$ // probabilistically mutate
  $\mathbb{H} \leftarrow \mathbb{H} \cup \{c_1, c_2\}$ // add new models to batch
end
return $\mathbb{H}$

1.2.2.1 *Distinguishing F_1-score through Bayes factors*

We have so far relied on Bayes factor (BF) as the means by which to distinguish models' ability to explain data from the target system. We conjecture that models of higher F_1-score are usually stronger at this task than those of lower F_1-score, which will allow us to incorporate these statistical tools into the design of OFs. We can test this simply by training models of equally spaced F_1-score, and computing BF between all pairs.

In **??**, we show the relationships between F_1-score and or various conditions: Firstly, under a standard training regime with full BF comparisons between all pairs, we see that in most cases, the model with higher F_1-score is favoured by BF. In Fig. 1.5b, we run a complete model training subroutine, but compute the BF based on fewer experiments and particles (retaining a fraction 0.2). This verifies an earlier claim from **??**: although the strength of evidence is weaker given reduced BF resources, the direction of the evidence is usually the same, i.e. the insight is indicative of the true physics, so we can save considerable compute time by trusting these restricted BFs calculations. On the other extreme, we see in Fig. 1.5**c**, where models are trained with, and BFs based upon, even greater resources, we see a similar effect: adding resources strengthens the evidence, but does not fundamentally change the outlook. Finally, in addition to reducing the resources used per BF calculation, we reduce the number of comparisons computed, as would be required for rating models according to Bayes factor enhanced Elo ratings (BFEER), in Fig. 1.5**d**. In essence, we can see that the insight is largely the same from the most and least expensive training/comparison strategies, and by exploiting the available evidence through Elo ratings, rather than brute-force computing as much evidence as possible, we can achieve similar results. Note that the time saving reported between full and partial connectivity between models scales with N_m: here, with $N_m = 10$, the former computes 45 BFs, while the latter computes 17; for $N_m = 60$, as used in full instances/runs, these rise to 600 and 1770 respectively, so the benefit of the Elo scheme is amplified. This saving depends on the user's choice of graph connectivity, see **??**, though is typically a factor between 2-3; in general, though, assuming we can reduce the resources used within the BF, this phase is considerably less cumbersome than the model training itself, so it is feasible to compute all available BFs to inform the BFEER.

1.2.3 *Hyperparameter search*

Firstly we will validate our reasoning that F_1-score is a sensible figure of merit, by directly invoking it as the objective function. That is, we first implement a GA, using the mapping between models and chromosomes outlined above, where we fix the numbers of sites $d = 4$, and assume full connectivity between the sites, with $x-$, $y-$ and $z-$ couplings available, such that there are $N_t = 3 \times \binom{4}{2} = 18$ terms in $\mathcal{T}$, so that the total population is of size 2^{18} chromosomes. We can then sweep over the yperparameters to find a suitable configuration: in Fig. 1.6 we show how the choice of parameters affect the success rate of preciesly identifying the target

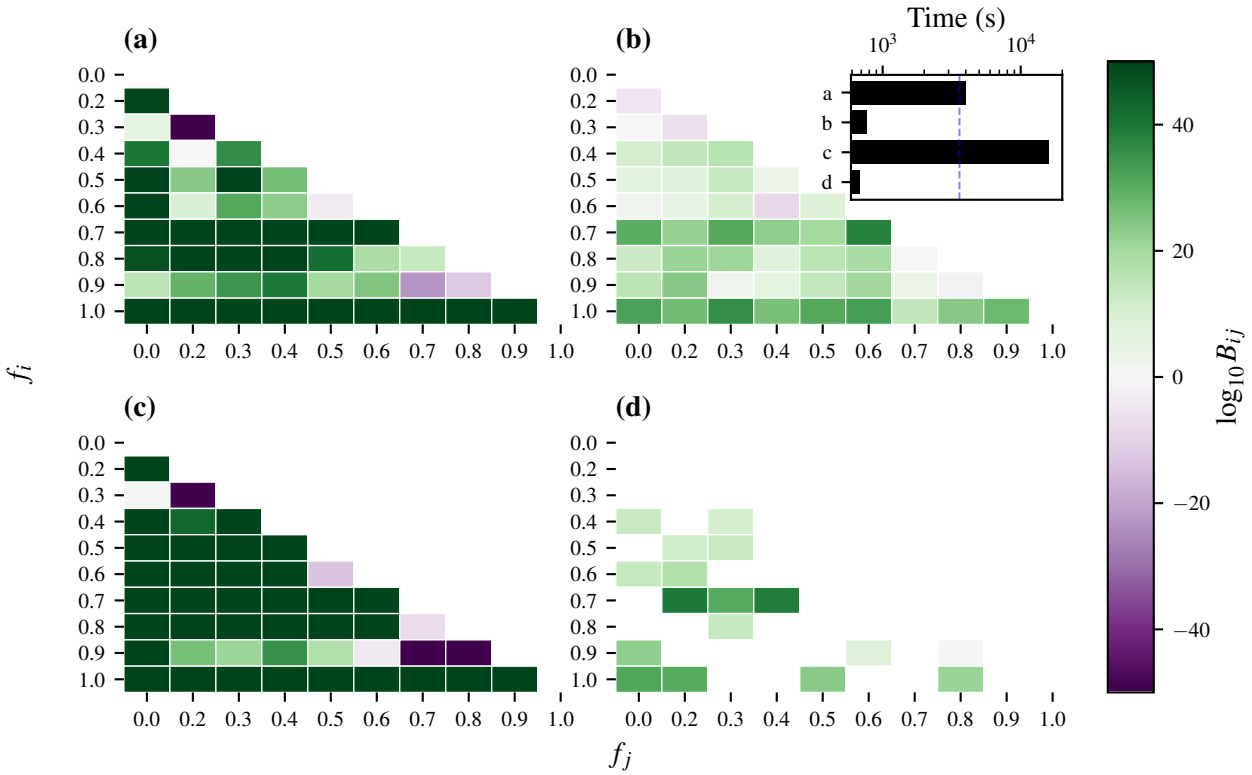


Figure 1.5: Pairwise Bayes factor, B_{ij} by F_1-score of candidates $\hat{H}_i$ (f_i on the y-axis) and $\hat{H}_j$ (f_j on the x-axis). $\log_{10} B_{ij} > 0$ (< 0), green (purple), indicates statistical evidence that $\hat{H}_i$ ($\hat{H}_j$) is the better model with respect to the observed data. Visualisation is curtailed to $\log_{10} B_{ij} = \pm 50$. **a**, Models are trained with $N_e = 500, N_p = 2500$, and all available data uis used in the calculation of BFs. **b**, $N_e = 500, N_p = 2500$ using only a fraction (0.2) of experiments/particles for BF calculation. **c**, $N_e = 1000, N_p = 5000$, using all available data in the calculation of BF. **d**, $N_e = 500, N_p = 2500$, comparing only a subset of pairs of models through BF, and using only a fraction (0.2) of experiments/particles for those calculations. This pairwise comparison strategy is used for the BFEER OF. **Inset**, timings for each approach in seconds, with $t = 1$hr marked vertically in blue.

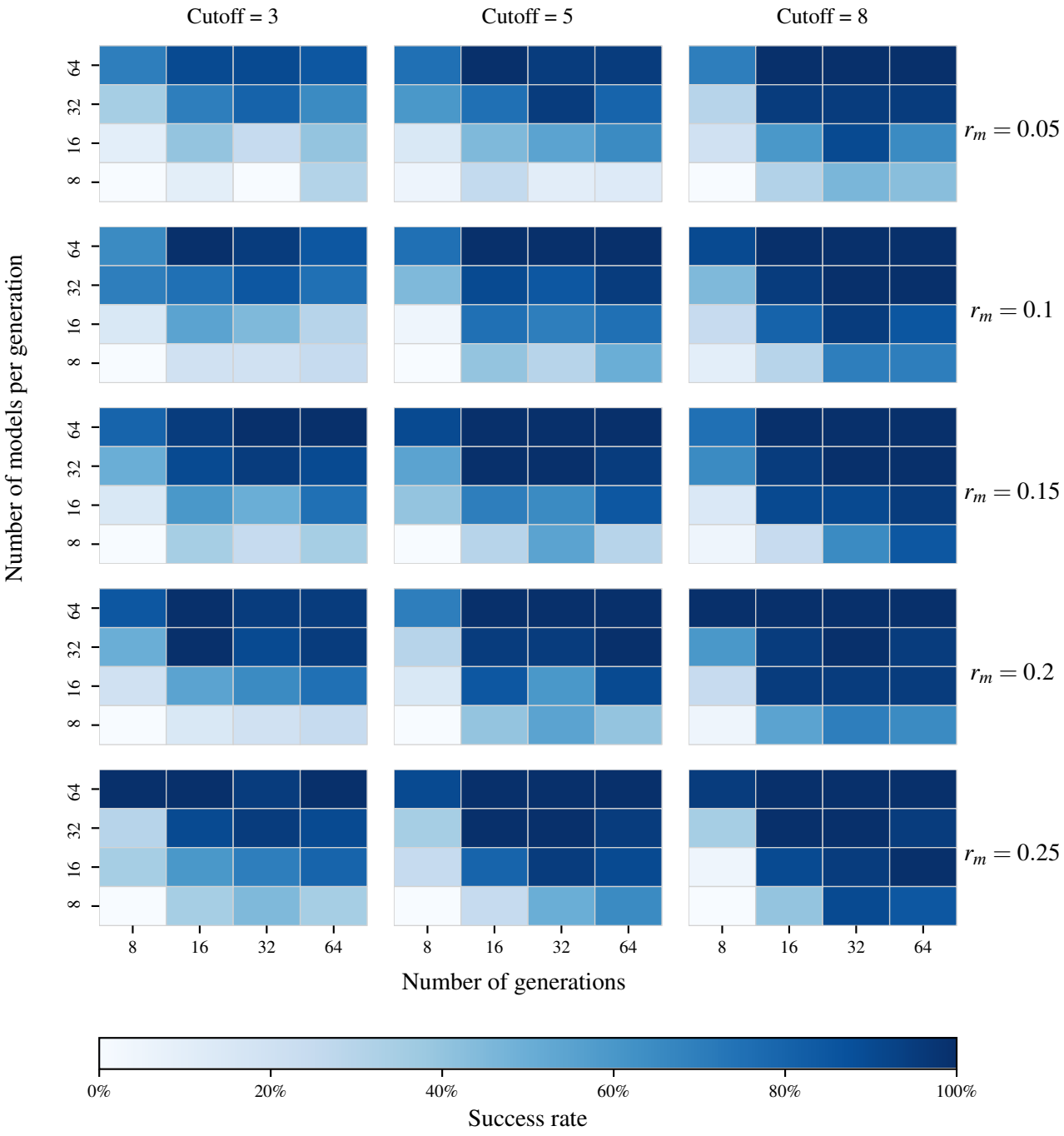


Figure 1.6: Genetic algorithm parameter sweep. Each subplot shows the success rates for varying numbers of generations, $N_G \in \{8, 16, 32, 64\}$, and numbers of models per generation, $N_m \in \{8, 16, 32, 64\}$. A subplot is generated for ranges of the mutation rate, r_m and the number of generations for which the elite model is unchanged after which the GA is cut off.

chromosome, which is chosen at random for each instance, and we run 20 instance of each configuration. The studied hyperparameters[4] are

1. number of generations;
2. number of models per generation;
3. mutation rate, r_m;
4. number of generation a candidate must reign as the strongest observed, before the search terminates, the *cutoff*.

Naturally, we expect that running for more generations with more models per generation will result in a more effective search in the model space, having examined $N_g N_m$ models. We must also consider, however, that – in realistic cases of QMLA – the total computation time scales badly with these, since training and comparing models are such expensive subroutines. Our goal is therefore to identify the set of hyperparameters which best searches the model space with minmial N_g, N_m. We see that, unsurprisingly, the GA performs poorly when run with few resources, but broadly the performances are similiar provided it is run with sufficient resources. We can bound the parameters $r_m \geq 0.1, cutoff \geq 5, N_m \geq 16, N_g \geq 16$ to ensure a reasonable search through the model space, without having to consider a prohibitive number of models. We must bear in mind, however, that this parameter sweep refers only to the trivial case where the F_1-score is used as the OF, so we do not expect such high success rates in realistic cases.

1.3 OBJECTIVE FUNCTIONS

We have alluded to the central probelm in building a GA into QMLA: how to evaluate trained candidate models in the absence of a natural OF. Here we will propose and analyse a number of potential OFs, some of which will underlie later studies in this thesis. Readers may prefer to skip to Section 1.3.8, where we conclude this study by choosing a single OF for consideration in this chapter.

We will show how each OF computes g_i for candidate models $\hat{H}_i$, and summarise the outcomes in Table 1.4. For each $\hat{H}_i$, we may refer to

- $\mathcal{L}_i$, total log total likelihood (TLTL), introduced in **??**
- k_i: the model's cardinality, i.e. number of terms in its parameterisation;
- $\mathcal{E}_i$: the bespoke set of experiments composed by the experiment design heuristic (EDH) solely for trainng $\hat{H}_i$;
- $n = |\mathcal{E}_i|$: the number of samples used in trainng $\hat{H}_i$.

4 These and further hyperparameters can be swept using code given in the QMLA codebase, in the directory scripts/genetic_alg_param_sweep.

In Table 1.4, we consider six models as examples of the outcome using each OF; the models, randomly generated of varying quality with respect to the target $\hat{H}_0$, are

$$\begin{aligned}
\hat{H}_0 &= \sigma^z_{(1,2)}\sigma^z_{(1,3)}\sigma^z_{(2,3)}\sigma^z_{(2,5)}\sigma^z_{(3,5)};\\
\hat{H}_a &= \sigma^z_{(1,5)}\sigma^z_{(3,4)}\sigma^z_{(4,5)};\\
\hat{H}_b &= \sigma^z_{(1,4)}\sigma^z_{(1,5)}\sigma^z_{(2,5)}\sigma^z_{(3,4)};\\
\hat{H}_c &= \sigma^z_{(1,2)}\sigma^z_{(1,5)}\sigma^z_{(2,4)}\sigma^z_{(2,5)}\sigma^z_{(4,5)};\\
\hat{H}_d &= \sigma^z_{(1,3)}\sigma^z_{(1,4)}\sigma^z_{(1,5)}\sigma^z_{(2,4)}\sigma^z_{(2,5)}\sigma^z_{(3,4)}\sigma^z_{(3,5)};\\
\hat{H}_e &= \sigma^z_{(1,2)}\sigma^z_{(1,3)}\sigma^z_{(1,5)}\sigma^z_{(2,3)}\sigma^z_{(2,5)}\sigma^z_{(4,5)};\\
\hat{H}_f &= \sigma^z_{(1,2)}\sigma^z_{(1,3)}\sigma^z_{(2,3)}\sigma^z_{(2,4)}\sigma^z_{(2,5)}\sigma^z_{(3,4)}\sigma^z_{(3,5)}.
\end{aligned} \tag{1.5}$$

1.3.1 *Inverse Log-likelihood*

$\mathcal{L}_i$ can be thought of as a measure of the success of a given model at explaining data from any set of experiments, $\mathcal{E}$. This can be immediately interpreted as an OF, provided each candidate model computes a meaningful total log total likelihood (TLTL), requiring that they are all based on the same set of experiments, $\mathcal{E}_v$, which are designed explicitly for the purpose of model evaluation.

TLTL are negative and the strongest model has lowest $|\mathcal{L}_i|$ (or highest $\mathcal{L}_i$ overall), so the corresponding OF for candidate $\hat{H}_i$ is

$$g^L_i = \frac{-1}{\mathcal{L}_i}. \tag{1.6}$$

In our tests, Eqn. 1.6 is found to be too generous to poor models, assigning them non-negligible probability. Its primary flaw, however, is its reliance on $\mathcal{E}_v$: in order that the TLTL is significant, it must be based on meaningful experiments, the design of which can not be gauranteed in advance, or at least risks introducing strong bias.

1.3.2 *Akaike Information Criterion*

A common metric in model selection is Akaike information criterion (AIC) [57]. Incorporating TLTL, AIC objectively quantifies how well a given model accounts for data from the target system, and punishes models which use extraneous parameters, by incurring a penalty on k_i. AIC is given by

$$AIC_i = 2k_i - 2\mathcal{L}_i. \tag{1.7}$$

Method		$\hat{H}_a$	$\hat{H}_b$	$\hat{H}_c$	$\hat{H}_d$	$\hat{H}_e$	$\hat{H}_f$
	F_1	0.0	0.2	0.4	0.5	0.7	0.8
	k	3	4	5	7	6	7
	$\overline{l_e}$	0.86 ± 0.29	0.84 ± 0.29	0.77 ± 0.27	0.78 ± 0.29	0.79 ± 0.26	0.79 ± 0.26
	$\mathcal{L}_i$	-143	-152	-131	-150	-125	-124
Inverse log-likelihood	g_i^L	0.00698	0.00659	0.00766	0.00669	0.00803	0.00804
	%	23	0	25	0	26	26
Akaike Info Criterion	AIC	293	311	271	313	261	263
	AICc	293	312	272	314	262	264
	w_i^A	1.81e-07	1.4e-11	0.00724	4.15e-12	1	0.334
	g_i^A	1.17e-05	1.03e-05	1.35e-05	1.01e-05	1.46e-05	1.43e-05
	%	22	0	25	0	27	26
Bayesian Info Criterion	BIC	301	322	284	331	277	281
	w_i^B	5.49e-66	1.26e-70	1.97e-62	1.11e-72	8.43e-61	8.95e-62
	g_i^B	1.11e-05	9.65e-06	1.24e-05	9.11e-06	1.31e-05	1.27e-05
	%	23	0	25	0	27	26
Bayes factor points	g_i^p	0	2	3	2	3	5
	%	0	13	20	13	20	33
Ranking points	Ranking	6	5	2	4	3	1
	g_i^R	0	0	0.3	0.1	0.2	0.4
	%	0	0	30	10	20	40
Elo rating	Rating	909	944	1042	1007	1011	1084
	g_i^E	0	35	133	98	102	175
	%	0	0	26	19	20	34
Residuals	mean$\{\tilde{r_p^e}\}$	0.132	0.146	0.114	0.138	0.0858	0.0715
	g_i^r	0.753	0.729	0.785	0.743	0.836	0.862
	%	23	0	24	0	26	27

Table 1.4: Examples of how each objective function, g as described in Section 1.3.1 to Section 1.3.7, assign selection probability (denoted %) to the same set of candidate models, $\{\hat{H}_i\}$, when attempting to learn data from $\hat{H}_0$, listed in Eq. (1.5). For each model we first summarise its average likelihood l_e (Eqn. **??**), total log-likelihood $\mathcal{L}_i$ (Eqn. **??**), as well as F_1-score and number of terms k. We use $n = 250$ samples, i.e. $\mathcal{L}_i$ is a sum of n likelihoods. The set of models is truncated so that only the strongest four are assigned selection probability.

In practice we use a slightly modified form of Eqn. 1.7 which corrects for the number of samples $n = |\mathcal{E}_i|$, called the Akaike information criterion corrected (AICC),

$$AICC_i = AIC_i + 2k_i \frac{k_i + 1}{n - k_i - 1}. \tag{1.8}$$

Model selection from a set of candidates occurs simply by selecting the model with $AICC_{\min}$. A suggestion to retrieve selection probability, by using Eqn. 1.8 as a measure of *relative likelihood*, is to compute *Akaike weights* (as defined in in Chapter 2 of [57]),

$$w_i^A = exp\left(\frac{AICC_{\min} - AICC_i}{2}\right), \tag{1.9}$$

where $AICC_{\min}$ is the lowest $AICC$ observed among the models under consideration e.g. all models in a given generation.

Clearly, Akaike weights impose quite strong penalties on models which do not explain the data well, but also punish models with extra parameters, i.e. overfitting models, effectively searching for the strongest and simplest model simultaneously. The level of punishment for poorly performing models is likely too drastic: very few models will be in a range sufficiently close to $AICC_{\min}$ to receive a meaningful Akaike weight, suppressing diversity in the model population. Indeed, we can see from Table 1.4 that this results in most models being assigned negligible weight, which is not useful for parent selection. Instead we compute a straightforward quantity as the AIC–inspired fitness, Eqn. 1.10,

$$g_i^A = \left(\frac{1}{AICc_i}\right)^2, \tag{1.10}$$

where we square the inverse AIC to amplify the difference in quality between models, such that stronger models are generously rewarded.

1.3.3 *Bayesian Information Criterion*

Related to the idea of AIC, Eqn. 1.7, is that of Bayesian information criterion (BIC),

$$BIC_i = k_i \, ln(n_i) - 2\mathcal{L}_i, \tag{1.11}$$

where k_i, n_i and $\mathcal{L}_i$ are as defined on Page 17. Analagously to Akaike weights, *Bayes weights* as proposed in §7.7 of [58], are given by

$$w_i^B = exp\left(-\frac{BIC_i}{2}\right). \tag{1.12}$$

BIC is harsher than AIC in its punishment of the number of parameters in each model, therefore requiring strong statistical justification for the addition of any parameters. Again,

this may be overly cumbersome for our use case: with such a relatively small number of parameters, the punishment is disproportionate; moreover since we are trying to uncover physical interactions, we do not necessarily want to suppress models merely for their cardinality, since this might result in favouring simple models which do not capture the physics. As with Akaike weights, then, we opt instead for a simpler objective function,

$$g_i^B = \left(\frac{1}{BIC_i}\right)^2. \tag{1.13}$$

1.3.4 *Bayes factor points*

A cornerstone of model selection within QMLA is the calculation of Bayes factor (BF) (see **??**). We can compute the pairwise BF between two candidate models, B_{ij}, according to Eqn. **??**. B_{ij} can be based on some evaluation dataset, $\mathcal{E}_v$, but can also be calculated from $\mathcal{E}_i \cup \mathcal{E}_j$: this is a strong advantage since the resulting insight (Eqn. **??**) is based on experiments which were bespoke to both $\hat{H}_i, \hat{H}_j$. As such we can be confident that this insight accurately points us to the stronger of two candidate models.

We can utilise this facility by simply computing the BF between all pairs of models in a set of N_m candidates $\{\hat{H}_i\}$, i.e. compute $\binom{N_m}{2}$ BFs. Note that this is computationally expensive: in order to train $\hat{H}_i$ on $\mathcal{E}_j$ requires a further $\left|\mathcal{E}_j\right|$ experiments, each requiring N_P particles[5], where each particle corresponds to a unitary evolution and therefore the caluclation of a matrix exponential. The combinatorial scaling of the model space is then quite a heavy disadvantage. However, in the case where all pairwise BF are performed, we can assign a point to $\hat{H}_i$ for every comparison which favours it.

$$g_i^p = \sum_{j \in \mu} b_{ij}, \quad b_{ij} = \begin{cases} 1, & B_{ij} > 1 \\ 0, & \text{otherwise} \end{cases} \tag{1.14}$$

This is a straightforward mechanism, but is overly blunt because it does not account for the strength of the evidence in favour of each model. For example, a dominant model will receive only a slightly higher selection probability than the second strongest, even if the difference between them was $B_{ij} = 10^{100}$. Further, the unfavourable scaling make this an expensive method.

1.3.5 *Ranking*

Related to Section 1.3.4, we can rank models in a generation based on their number of BF points. BF points are assigned as in Eqn. 1.14, but instead of corresponding directly to fitness, we assign models a rank R, i.e. the model with highest g_i^p gets $R = 1$, and the model with n^{th} highest g_i^p gets $R = n$. Note here we truncate μ, meaning we remove the worse-performing models and

5 Caveat the reduction in overhead outlined in **??**.

retain only N'_m models, before calculating R. This is because computing R using all N_m models results in less distinct selection probabilities.

$$g_i^R = \frac{N'_m - R_i + 1}{\sum_{n=1}^{N'_m} n}, \tag{1.15}$$

where R_i is the rank of $\hat{H}_i$ and N'_m is the number of models retained after truncation.

1.3.6 *Residuals*

Recall at each experiment, N_P particles are compared against a single experimental datum, d. For consistency with QInfer [7] – on which QMLA's code base extends – we call the expectation value for the system $\Pr_Q(0)$, and that of each particle $\Pr_p(0)$, recall **??**. Typically, $\Pr_Q(0) = \left|\langle\psi| e^{-i\hat{H}_0 t} |\psi\rangle\right|^2$, but this can be changed to match given experimental schemes, e.g. the Hahn-echo sequence applied in [1]. By definition, the datum d is the binary outcome of the measurement on the system under experimental conditions e. That is, d encodes the answer to the question: after time t under Hamiltonian evolution, did Q project onto the basis we have labelled 0 (usually the same as the input probe state $|\psi\rangle$)? However, in practice we often have access also to the likelihood , i.e. rather than a binary value, a number representing the probability that Q will project on to 0 for a given experiment e, $\Pr_Q(0|e)$. Likewise, we can simulate this quantity for each particle, $\Pr_p(0|e)$. This allows us to calculate the *residual* between the system and individual particles' likelihoods, r_p^e, as well as the mean residual across all particles in a single experiment r^e:

$$\begin{aligned} r_p^e &= \left|Pr_Q(0|e) - Pr_p(0|e)\right| \\ r^e &= \underset{p}{\text{mean}}\{r_p^e\} \end{aligned} \tag{1.16}$$

Residuals capture how closely the particle distribution reproduced the dynamics from Q: $r_p^e = 0$ indicates perfect prediction, while $r_p^e = 1$ is completely incorrect. We can therefore maximise the quantity $1 - r$ to find the best model, using the OF

$$g_i^r = \left|1 - \underset{e\in\mathcal{E}}{\text{mean}}\{r^e\}\right|^2. \tag{1.17}$$

This OF can be thought of in frequentist terms as similar to the residual sum of squares, although instead of summing the residual squares, we average to ensure $0 \leq r \leq 1$. g_i^r encapsulates how well a candidate model can reproduce dynamics from the target system, as a proxy for whether that candidate describes the system. This is not always a safe figure of merit: in most cases, we do not expect parameter learning to perfectly optimise $\vec{\alpha}_i$. Reproduced dynamics alone can not capture the likelihood that $\hat{H}_i = \hat{H}_0$. However, this OF provides a useful

test for QMLA's GA: by simulating the case where parameters *are* learned perfectly, such that we know that g_i^r truly represents the ability of $\hat{H}_i$ to simulate $\hat{H}_0$, then this OF gaurantees to promote the strongest models, especially given that $\hat{H}_i = \hat{H}_0 \implies r_p^e = 0 \; \forall \; \{e, p\}$. In realistic cases, however, the non-zero residuals – even for strong $\hat{H}_i$ – may arise from imperfectly learned parameters, rendering the usefulness of this OF uncertain. Finally, it does not account for the cardinality, k_i, of the candidate models, which could result in favouring severely overfitting models in order to gain marginal improvement in residuals, which all machine learning protocols aim to avoid in general.

1.3.7 *Bayes factor enhanced Elo-ratings*

A popular tool for rating individual competitors in sports and games is the *Elo rating* scheme, e.g. used to rate chess players and soccer teams [59, 60], also finding application in the study of animal hierarchies [61]. Elo ratings allow for evaluating relative quality of individuals based on incomplete pairwise competitions, e.g. despite two football teams having never played against each other before, it is possible to quantify the difference in quality between those teams, and therefore to predict a result in advance [62]. We recognise a parallel with these types of competitions by noting that in our case, we similarly have a pool of individuals (models), which we can place in direct competition, and quantify the comparitive outcome through BF.

Elo ratings are transitive: given some interconnectivity in a generation, we need not compare *every* pair of models in order to make meaningful claims about which are strongest; it is sufficient to perform a subset of comparisons, ensuring each individual is tested robustly. We can take advantage of this transitivity to reduce the combinatorial overhead usually associated with computing bespoke BF between all models (i.e. using their own training data $\mathcal{E}_i$ instead of a generic $\mathcal{E}_v$). In practice, we map models within a generation to nodes on a graph, which is then sparsely connected. In composing the list of edges for this graph, we primarily prioritise each node having a similar number of edges, and secondarily the distance between any two nodes. For example, with 14 nodes we overlay edges such that each node is connected with 5,6 or 7 other nodes, and all nodes at least share a competitor in common.

The Elo rating scheme is as follows: upon creation, $\hat{H}_i$ is assigned a rating R_i; every comparison with a competitor $\hat{H}_j$ results in B_{ij}; R_i is updated according to the known strength of its competitor, R_j, as well as the result B_{ij}. The Elo update ensures that winning models are rewarded for defeating another model, but that the extent of that reward reflects the quality of its opponent. As such, this is a fairer mechanism than BF points, which award a point for every victory irrespective of the opposition: if $\hat{H}_j$ is already known to be a strong or poor model, then ΔR_i proportionally changes the credence we assign to $\hat{H}_i$. It achieves this by first computing the *expected* result of a given comparison with respect to each model, based on the current ratings,

$$E_i = \frac{1}{1 + 10^{\frac{R_j - R_i}{400}}}; \tag{1.18a}$$

Model		R_i	E_i	S_i	B_{ij}	$\log_{10}(B_{ij})$	ΔR_i	R'_i
$\hat{H}_a > \hat{H}_b$	$\hat{H}_a$	1000	0.76	1	1e+100	100	0.24	1024.0
	$\hat{H}_b$	800	0.24	0	1e-100	100	-0.24	776.0
$\hat{H}_b > \hat{H}_a$	$\hat{H}_a$	1000	0.76	0	1e-100	100	-0.76	924.0
	$\hat{H}_b$	800	0.24	1	1e+100	100	0.76	876.0

Table 1.5: Example of Elo rating updates. We have two models, where $\hat{H}_a$ is initially believed to be a stronger candidate than $\hat{H}_b$, i.e. has a higher starting Elo rating. We show the effect of strong evidence[6] in favour of each model following BF comparison, $B_{ij} \sim 10^{100}$. In the case where $\hat{H}_a$ defeats $\hat{H}_b$, because this was so strongly expected given their initial ratings, the reward for $\hat{H}_a$ (and cost to $\hat{H}_b$) is relatively small, compared with the case where – contrary to prediction – $\hat{H}_b$ defeats $\hat{H}_a$.

$$E_i + E_j = 1, \tag{1.18b}$$

Then, we find the binary *score* from the perspective of each model:

$$\begin{cases} B_{ij} > 1 & \Rightarrow \; S_i = 1; \; S_j = 0 \\ B_{ij} < 1 & \Rightarrow \; S_i = 0; \; S_j = 1 \end{cases} \tag{1.19}$$

which is used to determine the change to each model's rating:

$$\Delta R_i = \eta \times (S_i - E_i)\,. \tag{1.20}$$

An important detail is the choice of η, i.e. the *weight* of the change to the models' ratings. In standard Elo schemes this is a fixed constant, but here – taking inspiration from football ratings where η is the number of goals by which one team won – we weight the change by the strength of our belief in the outcome: $\eta \propto \left|B_{ij}\right|$. That is, similarly to the interpretation of Eqn. **??**, we use the evidence in favour of the winning model to transfer points from the loser to the winner, albeit we temper this by instead using $\eta = log_{10}(B_{ij})$, since BF can give very large numbers. In total, then, following the comparison between models $\hat{H}_i, \hat{H}_j$, we can perform the Elo rating update

$$R'_i = R_i + \log_{10}(B_{ij}) \left(S_i - \frac{1}{1 + 10^{\frac{R_j - R_i}{400}}} \right). \tag{1.21}$$

This procedure is easiest to understand by following the example in Table 1.5.

Finally, it remains to select the starting rating R_i^0 to assign models upon creation. Although this choice is arbitrary, it can have a strong effect on the progression of the algorithm. Here

6 Note to achieve $B_{ij} = 10^{100} = e^{\mathcal{L}_i - \mathcal{L}_j} \implies \mathcal{L}_i - \mathcal{L}_j = ln(10^{100}) \approx 7$.

we impose details specific to the QMLA GA: at each generation we admit the top two models automatically for consideration in the next generation, such that strongest models can stay alive in the population and ultimately win. These are called *elite* models, $\hat{H}_e^1, \hat{H}_e^2$. This poses the strong possibility for a form of generational wealth: if elite models have already existed for several generations, their Elo ratings will be higher than all alternatives by defintion. Therefore by maintaining a constant R_i^0, i.e. a model born at generation 12 gets the same R_i^0 as $\hat{H}_e^1$ – which was born several generations prior and has been winning BF comparisons ever since – we bias the GA to continue to favour the elite models. Instead, we would prefer that newly born models can overtake the Elo rating of elite models. We achieve this through an imprecise mechanism: newly born models are given the Elo rating of the second-most-elite model, $\hat{H}_e^2$. This performs three key functions:

i. new models are immediately *within range* of the elite models; if they perform well enough, they have a realistic and fair chance of overtaking them;

ii. the strongest model retains some of its advantage gained over previous generations – in order that the ratings are meaningful, there must be some advantage accrued over series of victories;

iii. $\hat{H}_e^2$ is allowed continue to compete, but has no advantage: in order to retain its status as elite, it must perform well *again* in this generation, so it can not simply rely on results from previous generations – against inferior opposition – to remain active in the gene pool.

Given the arbitrary scaling of the Elo rating scheme, and in order to derive a meaningful selection probability, we ought to ground the raw Elo rating somehow at each generation μ. We do this by subtracting the lowest rating among the entertained models, R^μ_{min}. This serves to ensure the range of remaining R_i is defined only by the difference between models as assessed within μ: a very strong model might have much higher R_i than its contemporaries, but that difference was earned exclusively by comparison within μ, so it deserves higher fitness. We perform this step before truncation, so that the reamining models post-truncation all have non-zero fitness. Finally, then, we name this OF the *Bayes-factor enhanced Elo rating*, whereby the fitness of each model is attained directly from its rating after undergoing Elo updates in the current generation minus the minimum rating of any model in the same generation μ,

$$g_i^E = R_i^\mu - R^\mu_{\text{min}}. \tag{1.22}$$

The advantage of this OF is that it gives a meaningful value on the absolute quality of every model, allowing us to determine the strongest, and importantly to find the relative strength between models. Further, it exploits bespoke BFs, i.e. based on the considered models' individually designed $\mathcal{E}_i$, removing the impetus to design $\mathcal{E}_v$ which can evaluate models definitively. One disadvantage is that it does not explicitly punish models based on their cardinality, however this feature is partially embedded by adopting BF for the comparisons, which are known to protect against overfitting.

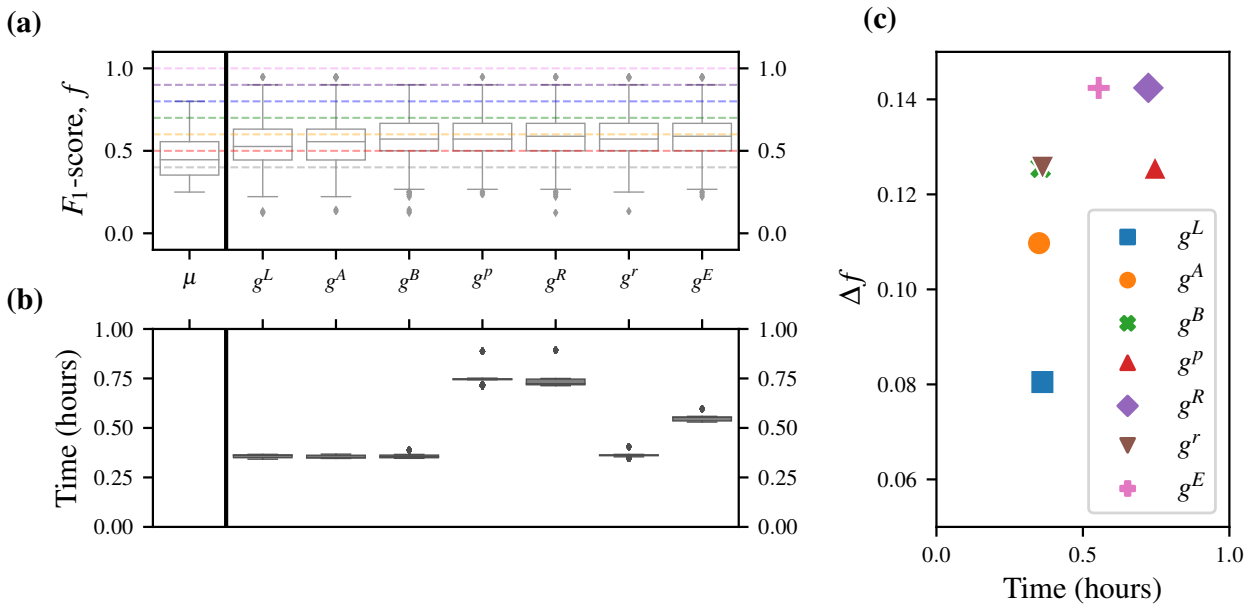


Figure 1.7: Comparison between proposed OFs. Each OF trains the same initial generation of $N_m = 28$ models with resources $N_E = 500, N_P = 2000$, and then design a new set of N_m models through the same roulette strategy, such that the only difference between OF's output is how they assign selection probability. We run each OF 25 times for the same target system, a 4–qubit Heisenberg–XYZ model. **(a)** shows the box–plot of new models' F_1-score, f, where the median and inter–quartile ranges are indicated by the boxes, as well as those of the initial generation μ centered on $f_\mu = 0.45$. We mark $f = \{0.4, 0.5, ..., 1.0\}$ for ease of interpretation. **(b)** shows box–plots of the time taken to compute the single generation in each case. In **(c)** we report the difference between the median f among the newly proposed models from f_μ, Δf, plotted against the time to achieve the result.

1.3.8 *Choice of objective function*

Having proposed a series of possible objective functions, we are now in a position to analyse which of those are most appropriate for QMLA. Recall from Section 1.2.2 the figure of merit we use for models, F_1-score, which we will use to distinguish between the outputs of each OF.

First we can remark on the examples listed in Table 1.4. The OFs which rely on the TLTL, i.e. g^L, g^A, g^B, g^r, are effectively tricked by the log likelihood, which appears reasonably convincing for poor models, e.g. $\hat{H}_a, \hat{H}_c$. This underlines the risk in buidling $\mathcal{E}_v$, which can be biased towards weak models, for example resulting in high selection probability for $\hat{H}_a$ which has $f = 0$, while $\hat{H}_d$, with $f = 0.4$ is discarded. On the other hand, OFs grounded by the BF (g^p, g^R, g^E) invariably promote models of higher F_1-score, justifying the role of statistical evidence used for those calculations. Overall, however, the insights from this complete example are insufficient to

make general claims about the performance of each OF, so here we examine their outputs systematically.

Returning to the task of determining our favoured OF, we choose some random target $\hat{H}_0$, and run a single generation using each OF, judging them by the quality of models they produce. We train the same batch of $N_m = 28$ random models in each case, and allow each OF to compute the selection probabilities for those models, and therefore direct the design of the hypothetical next generation of models. We plot the distribution of F_1-score that each OF produces in Fig. 1.7, also accounting for the time taken in each case, i.e. we report the time to train and evaluate the single generation on a 16–core node.

Overall, then, we can see that a strong balance of outcome with resource considerations are achieved by the Bayes factor enhanced Elo rating strategy, Section 1.3.7 so we use that for the case study presented in this chapter. We strongly emphasise, however, that the performance of each objective function can vary under alternative conditions, and therefore similar analysis may be warranted for future applications. For instance, if t_{max} is known to be small, in smaller model spaces, using g^r results in higher success rates. We retain BFEER, however, for generality and novelty, but it is important to recognise that the results listed do not reflect an upper limit of QMLA's performance, but rather reflect the constraints of the system under study; each Q will bring its own unique considerations which can result in significantly stronger or weaker performance. In particular, we will later use the OF based on residuals, Section 1.3.6, to study a much larger model space under assumptions of perfect parameter learning, Chapter 3.

1.4 APPLICATION

Having introduced all the necessary concepts of GAs, mapped them to the QMLA framework and chosen a suitable OF, we can finally use the GES for model search. In summary of this chapter so far, we use the following settings.

- Models are mapped to a bit string (chromosome), where each bit represents whether a given model term (gene) is present; chromosomes are of length N_t genes.
- A maximum of N_g generations are run, each with N_m unique models.
- Candidate models are trained using QHL, specifically by using interactive quantum likelihood estimation (IQLE)[7]for parameter estimation.
- Models' fitness are determined by their BFEER, after having been trained by QHL and compared against some set of competing candidate models.
- For generating models on $\mu + 1$, the models on μ are first truncated (the worst-performing $N_m/2$ are discarded), and then assigned selection probability based on their fitness.
- Models are selected to become parents sequentially using roulette selection. Highly favoured models can parent many children models.

- Selected parent models are crossed over via a one point cross-over, at crossover location $\kappa \in \left(\frac{N_t}{4}, \frac{3N_t}{4}\right)$, and probabilistically mutated with rate $r_m = 0.25$.
- The top two elite models from μ are included on the $\mu + 1$.
- If, after 5 generations, the highest-fitness (elite) model is unchanged, we terminate the search and declare that model as the champion, $\hat{H}'$.
- Otherwise, after N_g generations, the highest-fitness model on the final generation is declared $\hat{H}'$.

We will use a four-qubit model space under the Heisenberg formalism, **??**, such that any pair of sites $\langle k, l \rangle$ can be coupled by any of $\hat{\sigma}^x_{\langle k,l \rangle}, \hat{\sigma}^y_{\langle k,l \rangle} m \hat{\sigma}^z_{\langle k,l \rangle}$, so in total there are $N_t = |\mathcal{T}| = 3 \times \binom{4}{2} = 18$ terms, giving a model space of $2^{18} \approx 250,000$ viable models/chromosomes. For practical reasons[8], we set $N_m = 60$ and $N_g = 16$, although in most cases the elitism clause is triggered so the search terminates long before N_g is reached. The true parameters $\vec{\alpha}_0$ are assigned randomly in the range $(0.25, 0.75)$; within QHL the prior is set as a multivariate normal distribution 0.5 ± 0.125. We choose $\hat{H}_0$ at random to contain half the available terms[9],

$$\hat{H}_0 = \sigma^{yz}_{(1,2)} \sigma^z_{(1,3)} \sigma^y_{(1,4)} \sigma^{xy}_{(2,3)} \sigma^x_{(2,4)} \sigma^{xz}_{(3,4)}. \tag{1.23}$$

1.4.1 *Analysis*

We will analyse the GES from four perspectives: a single model, a single generation, a single QMLA instance, and the overall performance across many instances, i.e. a run.

Recall that BFEER are mediated through random graphs: given N_m models on μ, a given model $\hat{H}_i$ undergoes some $N^i_{BF} < N_m$ BF comparisons. In Fig. 1.8 we show the BF results and effects on the rating of a random model, $\hat{H}_i$, where $N_m = 60$ and $N^i_{BF} = 12$, i.e. $\hat{H}_i$ is directly compared against $\sim 20\%$ of contemporary models on μ. We see that $\hat{H}_i$'s rating is effected by whether it wins a given comparison, but also by the strength of evidence provided by the comparison (the BF), and the quality of its opposition (its rating).

We extend the single model analysis of Fig. 1.8 to all N_m models in the first generation in Fig. 1.9. The general trend is that models of higher F_1-score have their ratings increased, at the expense of models of lower F_1-score. After assessing models thus, the set of models is truncated to retain only the strongest candidates, which are assigned probability of being chosen to become a parent during roulette selection, as in Section 1.1.3. The very strongest two models are granted a position in the subsequent generation: these are the *elite* models.

7 IQLE assumes complete access to the target system see **??**. This restricts the present analysis to simulateable, rather than physical, use cases, e.g. device calibration.

8 This is to ensure, with 15 available worker nodes, and accounting for some slowly-learning models, that all N_m models in a generation are trained within $2t_{qhl}$, where t_{qhl} is the time to train a single model.

9 Note we use a compact model representation, e.g. $\hat{H}_i = \sigma^{yz}_{(1,2)} \sigma^z_{(1,3)} = \sigma^y_{(1,2)} + \sigma^z_{(1,2)} + \sigma^z_{(1,3)}$.

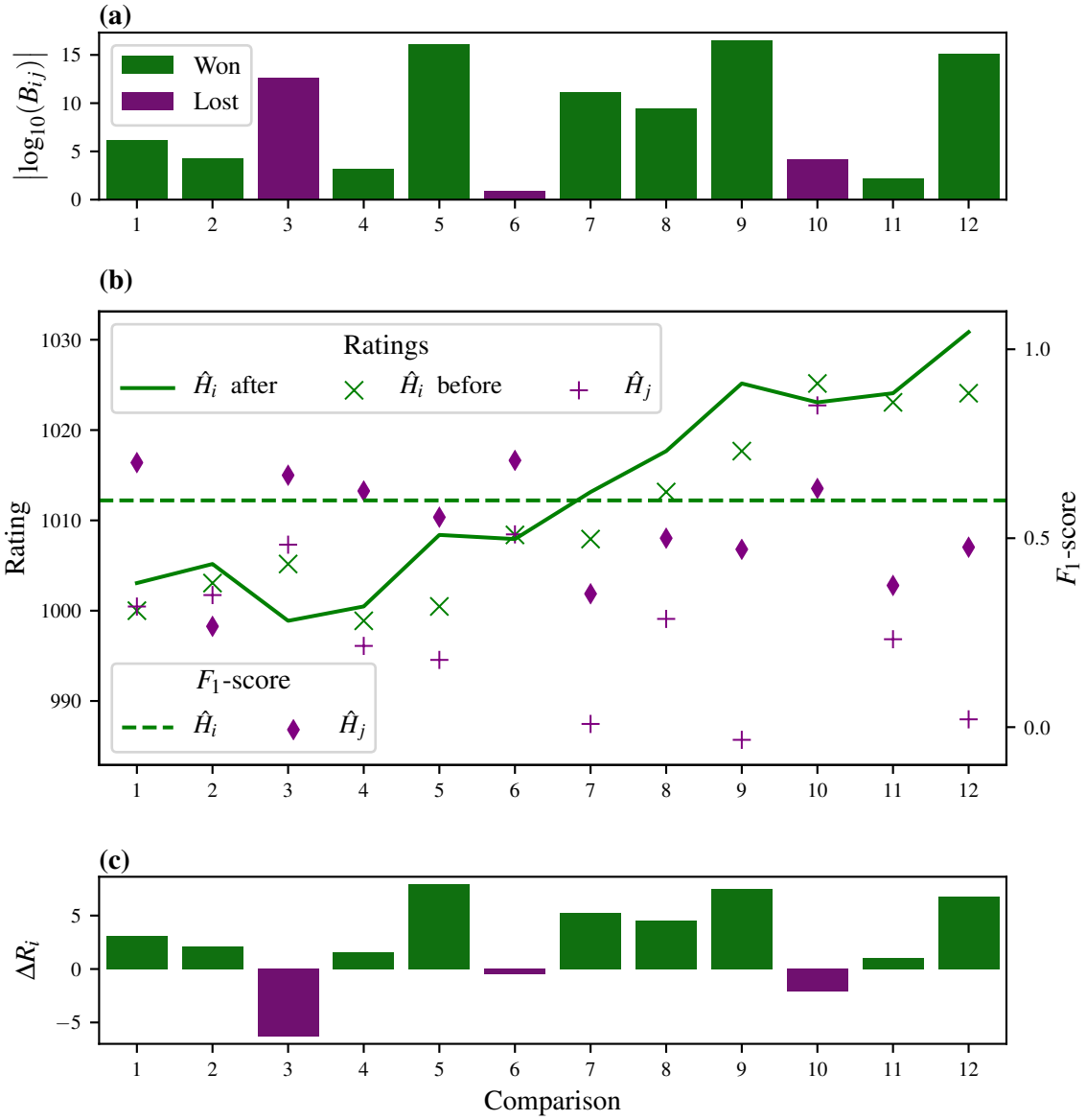


Figure 1.8: Bayes factor enhanced Elo ratings (BFEER) progression for a single candidate, $\hat{H}_i$, within a single generation. **a**, The BFs between $\hat{H}_i$ and some opponents, $\{\hat{H}_j\}$, from the perspective where $\hat{H}_i$ *wins* given $B_{ij} > 1 \Rightarrow \log_{10} B_{ij} > 0$, and loses otherwise. **b**, $\hat{H}_i$'s rating is shown (solid green line) changing according to the BFs comparisons with 12 other models from the same generation. Before each comparison, $\hat{H}_i$'s rating is shown (green cross) as well as the rating of its opponent, $\hat{H}_j$ (purple plus). The F_1-scores are also shown for $\hat{H}_i$ (dashed green line) and $\hat{H}_j$ (purple diamond). **c**, The corresponding change in $\hat{H}_i$'s rating, ΔR_i.

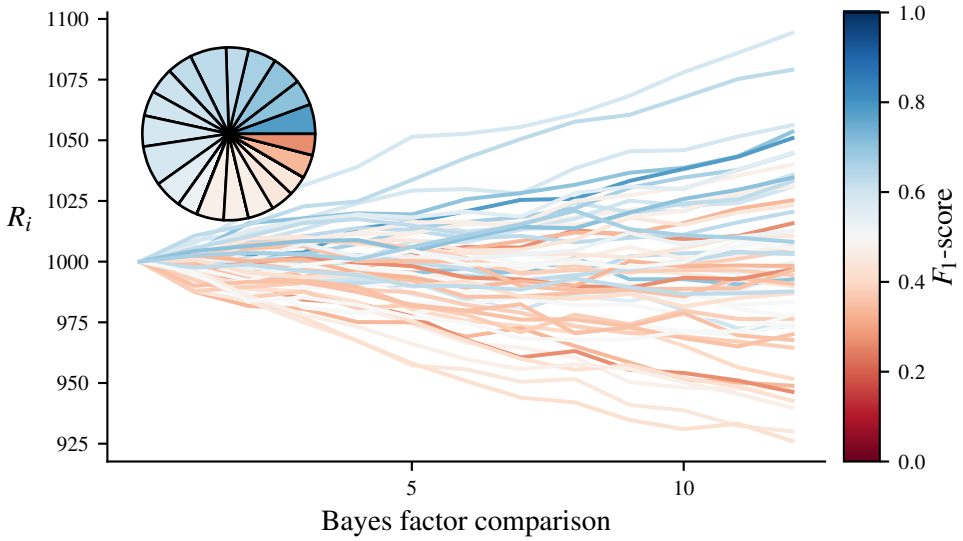


Figure 1.9: Ratings of all models in a single GA generation. Each line represents a unique model and is coloured by the F_1-score of that model. **Inset**, the selection probabilities resulting from the final ratings of this generation. Only a fraction of models are assigned selection probability, while the remaining poorer-performing models are truncated.

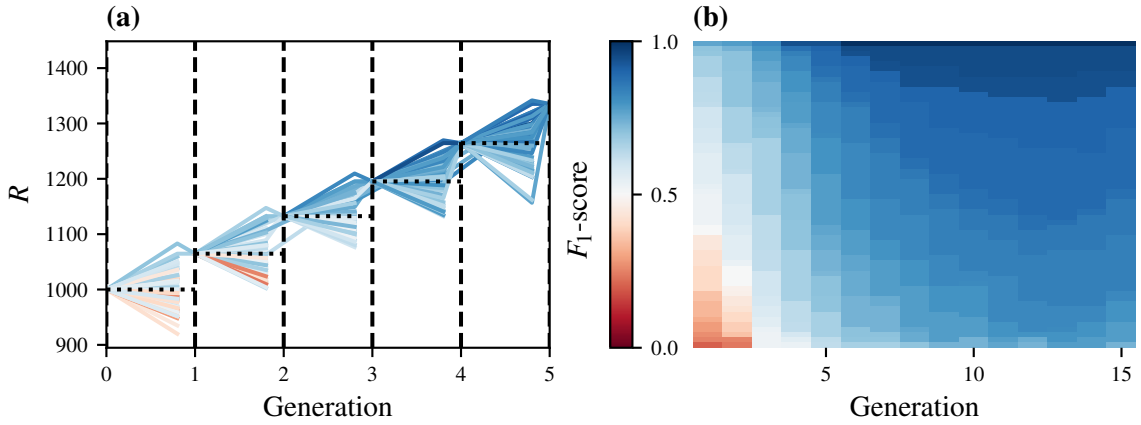


Figure 1.10: A single instance of the QMLA GA. **a**, Ratings of all models for the first five generations. Each line in each generation represents a model by its F_1-score. Horizontal dotted lines show the starting rating at that generation. **b**, Gene pool progression for $N_m = 60, N_g = 15$. Each time at each generation represents a model by its F_1-score.

Similarly we can consider the quality of models across at each generation, and their ratings. In Fig. 1.10 we see the trend suggested by Fig. 1.9 continue, i.e. that models of higher quality overall tend to achieve higher BFEER. The gene pool as a whole tends towards a homogeneous set of models, all with $f \geq 0.85$. Consequently, even in cases where the precise model, $\hat{H}_0$, is not identified, the champion model is highly informative, in that it captures many of the same interactions, therefore most-likely providing meaningul insight on the system's physics.

Finally, to understand the performance of the QMLA algorithm overall, we run 100 independent instances in a run, Fig. 1.11. We see that, while the overall model space can be characterised by the distribution of models' F_1-score, where we sample where $f = 0.5 \pm 0.14$, that QMLA quickly moves to the subspace of high-quality models, i.e. the models explored have median $f = 0.76 \pm 0.15$. This exploration is based on 430 ± 45 chromosomes per instance, i.e. QMLA trains only 0.16% of the 2^{18} potential models. Ultimately QMLA nominates champion models, $\{\hat{H}'\}$ with $f \geq 0.88$ in all instances, and precisely identifies $\hat{H}' = \hat{H}_0$ in 72% of instances. Considering the big picture, where the remit of QMLA is to identify the interactions the target system is subject to, we show how often each term/gene is included in $\hat{H}'$ in Fig. 1.11**d**. Crucially, we see that terms which really are within the true Hamiltonian, $\hat{t} \in \mathcal{T}_0$, are found significantly more frequently than those without, $\hat{t} \notin \mathcal{T}_0$. This level of analysis can be used to post-validate the outcome of QMLA, i.e. rather than relying on $\hat{H}'$ from a single instance, trusting the terms' individual frequencies as evidence that they are of importance when describing the system of interest. Of particular

1.4.2 *Device characterisation*

This adaptive GES may prove a useful application of QMLA in the domain of device calibration, in particular to characterise some untrusted quantum simulator. That is, by using the simulator to implement some target $\hat{H}_0$, QMLA can identify which operator is *actually* implemented. For instance, implementation of a four–qubit model relies on high-fidelity two-qubit gates between arbitrary qubit pairs, and QMLA can effectively reconstruct which operations were and were not faithfully computed.

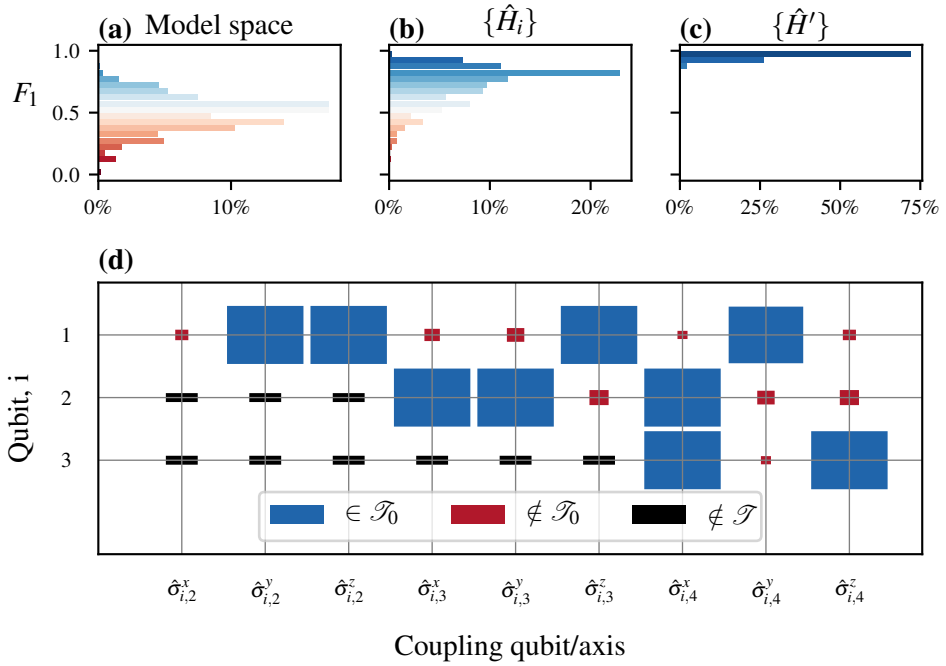


Figure 1.11: A run of the QMLA GA, consisting of 100 independent instances. **(a)**, The model space contains $2^{18} \approx 250,000$ candidate models; normally distributed around $f = 0.5 \pm 0.14$. **(b)**, The models explored during the model search of all instances combined, $\{\hat{H}_i\}$, show that QMLA tends towards stronger models overall, with $f = 0.76 \pm 0.15$ from $\sim$ 43,000 chromosomes across the instances, i.e. each instance trains $\sim$ 430 distinct models. **(c)**, Champion models from each instance, showing QMLA finds strong models in general, and in particular finds the true model $\hat{H}_0$ (with $f = 1$) in 72% of cases, and $f \geq 0.88$ in all instances. **(d)**, Hinton diagram showing the rate at which each term is found in the winning model. The size of the blocks show the frequency with which they are found, while the colour indicates whether that term was in the true model (blue) or not (red). Terms are couplings between two qubits, e.g $\hat{\sigma}^x_{(1,3)}$ couples the first and third qubits along the x-axis. We test four qubits with full connectivity, resulting in 18 unique terms (terms with black rectangles are not considered by the GA).

Part I

EXPERIMENTAL STUDIES

2

NITROGEN VACANCY CENTRE

It is of primary interest to apply the QMLA algorithm to real-life, experimental systems. In this chapter we devise an ES to operate in conjunction with experimental data in order to characterise an electron spin in an NV centre in diamond. In particular, we model, through Hamiltonian terms, interactions between the spin and the spin bath in which it resides, so that QMLA is finding an effective model for the open system dynamics.

Here we will first introduce a basic picture of nitrogen-vacancy centres (NVCs), using basic but nonstandard nomenclature for simplicity; for thorough descriptions of the underlying physics, readers are referred to [63]. We next discuss the target system with respect to its modelling, determining the suitable terms which *might* represent the NVC's interactions, to inform the starting point for the QMLA. Finally we describe the implementation of an ES for the examination of the NVC, and the results of the QMLA procedure.

2.1 NV-centres

NV centers are point defects in diamond, occuring naturally [64] or synthetically [65, 66]. A substitutional nitrogen-14 (^{14}N) isotope is embedded in a lattice of carbon atoms in diamond, adjacent to a lattice vacancy, such that it is surrounded by three carbon-13s (^{13}Cs) [67]. Of the ^{14}N atom's five valence electrons, three bond with nearby ^{13}Cs; the remaining two unbonded electrons for a lone pair and can be thought of as a single spin-$\frac{1}{2}$ particle. The adjacent lattice vacancy has three unbonded electrons, two of which bond together leaving a single unpaired electron. The single electron in the lattice vacancy, together with the effective lone pair of the ^{14}N, form a a system of two spin-$\frac{1}{2}$ particles. Such systems have been thoroughly studied; of particular interest are the resultant *triplet* states, i.e. the allowed permutations of the two particles with total quantum spin $S = 1$, with magnetic spin multiplicity allowing $m_s = -1, 0, 1$, giving rise to three distinct energy levels for the system.

A *manifold* is a set of states differing only slightly, for example states near the absolute ground state manifold might differ only in magnetic spin quantum number, and can be characterised as the ground state manifold. We consider two principle manifolds of the system: the ground state and excited manifolds, each consisting of three states, corresponding to the allowed values for magnetic spin m_s, see Fig. 2.1a. For brevity, we denote states with reference to their magnetic spin and manifold, e.g. the state in the ground state manifold with $m_s = 0$ is denoted $|m_s = 0\rangle_g$. In the absence of a magnetic field, the states corresponding to $|m_s = \pm 1\rangle$ are degenerate, but in the presence of a magnetic field, B, they have distinct energy levels, referred to as the Zeeman effect, Fig. 2.1b.

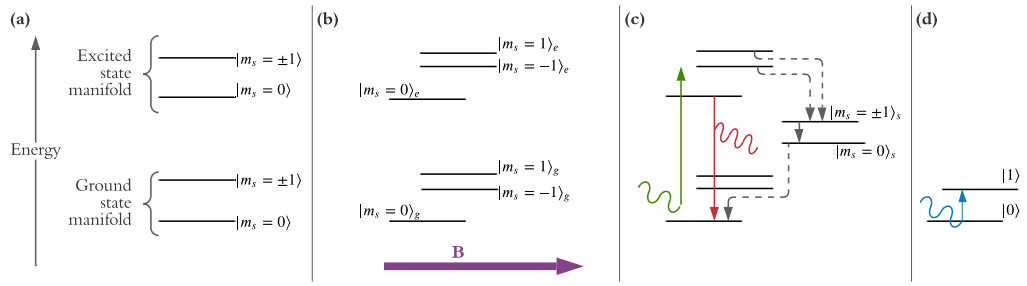


Figure 2.1: Simplified depiction of energy levels of the nitrogen-vacancy centre, corresponding to its triplet state. **a**, With no external field, the system simply has excited and ground-state manifolds, each of which consist of two energy levels depending on the magnetic spin, m_s. **b**, In the presence of a magnetic field (purple, B), the magnetic spins have distinct energy levels, i.e. Zeeman splitting. States are denoted by their magnetic spin and subscripted by their manifold. **c**, Application of a green ($532nm$) laser excites the NVC from any of the states in the ground state manifold to the excited manifold. The dominant decay mechanism for the excited states are shown: (i) $|m_s = 0\rangle_e \to |m_s = 0\rangle_g$ (red line) through the emission of a red ($637nm$) photon; (ii) $|m_s = \pm 1\rangle_e \to |m_s = 0\rangle_g$ (dotted grey lines) via the shelving manifold which allows for non-spin-preserving transition, and does not emit a photon. **d**, Computational basis states $|0\rangle$ and $|1\rangle$ are assigned to the two lowest energy states. The difference in energy between these states is such that a microwave (MW, blue) photon can trigger transition from $|0\rangle$ to $|1\rangle$, as well as states in between such as $|+\rangle$, allowing for the implementation of quantum logic gates.

We designate the state $|m_s = 0\rangle_g$ as the basis state $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and $|m_s = -1\rangle_g$ as $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, such that we have defined a qubit and computational basis, Fig. 2.1d. By shining a laser of 532nm (green) on the NVC, it is excited to the excited manifold, from which it decays back to the ground state manifold. Importantly, the process of this decay can be exploited for the preparation of the NVC in the computational basis state $|0\rangle$. That is, the dominant decay process from $|m_s = 0\rangle_e$ is spin-preserving, so it ends in $|m_s = 0\rangle_g$. On the other hand, had the NVC been in the $|m_s = \pm 1\rangle_e$, the dominant decay process is through a shelving (singlet) state, and does not preserve spin, such that it also decays to the $|m_s = 0\rangle_g$. Therefore, irrespective of the initial state, by shining the green laser on the NVC, it is most likely that it has been prepared in $|m_s = 0\rangle_g = |0\rangle$, providing us a starting point from which to perform computation.

The difference in energy between $|0\rangle$ and $|1\rangle$ is addressible optically: by shining a pulse at the NVC, we can oscillate the qubit between the two levels. Likewise, having initialised the state to $|0\rangle$, we can perform a $\pi/2$ rotation about the logical z-axis, by running the laser for half the time, resulting in the state $|+\rangle$. We can similarly devise radiation to achieve quantum gates and operations on our NVC qubit. We depict these cycles in Fig. 2.1c.

We can further exploit the decay mechanism to compose a readout procedure, to infer the population of $\{|0\rangle, |1\rangle\}$ at a given instant, for example following the application of gates to the system. We know that the excitation due to the green laser is spin-preserving, i.e. when the NVC has been excited to $|m_s = 0\rangle_e$, it had originated in $|m_s = 0\rangle_g$. We also know that the decay $|m_s = 0\rangle_e \to |m_s = 0\rangle_g$ is spin preserving, with the emission of a red photon: by simply counting the number of photons emitted, we quantify the population of $|0\rangle$ at the time of query. On the contrary, when the $|m_s = -1\rangle_g$ is excited, spin is also preserved, so it goes to $|m_s = -1\rangle_e$; but $|m_s = -1\rangle_e$ decays through the shelving state as outlined earlier, *without* the emission of a photon. We can hence infer the population of $|m_s = -1\rangle_g$ at the time of query by the fraction of incidents which don't emit a photon. That is, say we first calibrate the system by retaining the green laser for some time: after a few μs, a steady state is achieved where the majority of the time, the triplet is in the state $|0\rangle = |m_s = 0\rangle_g$. Then, excitation from the same laser results in the excitation to $|m_s = 0\rangle_e$, which decays back to $|m_s = 0\rangle_g$ and emits a photon in the process; by counting the red photons emitted in a certain time window – equivalently, measuring the photoluminescence (PL) signal – we benchmark the population of $|0\rangle$ when nothing else has happened as p_0. Now, when we apply gates (i.e. pulses) to the NVC, we can similarly read out the population of $|0\rangle$ as p_0', and infer that the likelihood that the is found in the initial state $|0\rangle$ is p_0'/p_0. We can use this quantity as the likelihood within quantum likelihood estimation (QLE), allowing us to learn from the NVC, as we will discuss in the next sections.

In summary then, by assigning basis states $|0\rangle, |1\rangle$ to energy levels of the ground state manifold, we are able to ensure the preparation of the NVC in $|0\rangle$ by first shining a green laser on the NVC. We can then apply radiation to achieve quantum logical gates on the system, and read out the final state of the system, again by shining a green laser and observing the emitted photons (PL) and inferring the population level of each basis state. We represent these concepts in a simplified format in Fig. 2.1.

2.2 TARGET SYSTEM

We take the axis of the NVC, i.e. the axis connecting the ^{14}N with the lattice vacancy, as the z-axis. There are clearly a huge number of interactions to which the NVC is subject: we choose to focus on its interactions with the environment, i.e. with nuceli in the same diamond, which dictate its decoherence. These interactions are characterised by hyperfine terms [68]. The overall Hamiltonian for such systems, where the set of nuclear sites is $\{\chi\}$, is given by

$$\hat{H}_{\text{full}} = \Delta_{\text{gs}}\hat{S}_z^2 + \mu_B g\mathbf{B}\cdot\mathbf{S} + \mathbf{S}\cdot\sum_\chi (\mathbf{A}_\chi\cdot\hat{I}_\chi) + P\hat{I}_z^2 + \mu_n g\mathbf{B}\cdot\sum_\chi \hat{I}_\chi. \tag{2.1}$$

We will describe each term, as well as approximations which enable us to simplify the space considerably.

- Isolated-spin terms, i.e. describing the spin independent of the environmental nuclei

  - $\Delta_{gs}\hat{S}_z^2$: the *zero-field* splitting, i.e. without any external (magnetic) field, the spin oscillates rapidly, with $\Delta_{gs} \sim$ GHz.
  - $\mu_B g \mathbf{B} \cdot \mathbf{S}$: the spin's precession about the magnetic field, $\mathbf{B} = \begin{pmatrix} B_x & B_y & B_z \end{pmatrix}$, via the total spin operator[1] $\mathbf{S} = \begin{pmatrix} \hat{S}_x & \hat{S}_y & \hat{S}_z \end{pmatrix}$, where μ_B is the Bohr magneton and g is the g-factor (≈ 2, simplified from the g-factor tensor).
- Hyperfine terms
  - $\hat{S} \cdot \sum_\chi (\mathbf{A}_\chi \cdot \hat{I}_\chi)$: The NVC total spin operator $\mathbb{S}$ couples with each site, χ. At each site there is a nucleus which has total spin operator $\mathbf{I}_\chi = \begin{pmatrix} \hat{I}_x & \hat{I}_y & \hat{I}_z \end{pmatrix}_\chi$. $\mathbf{A}$ is the hyperfine tensor, containing the hyperfine parameters of interest.
- Bath-only terms, i.e. describing the other nuceli independent of the spin
  - $P\hat{I}_z^2$: the quadrupole splitting, i.e. this term gives the splitting of the ^{14}N's hyperfine splitting, which does not meaningfully contribute to the short-decay processes in the experiments described in the next section.
  - $\mu_n g \mathbf{B} \cdot \sum_\chi \hat{I}_\chi$: μ_n is the nuclear magneton and g is again the g-factor.

Given that we are interested in the spin and its interactions with the environment only, we can immediately drop the bath-only terms, by assuming the bath is static apart from its interactions with the NVC. This is a usual assumption in the treatment of open system dynamics, to allow for focus on the dominant interactions for the system of interest [69]. Additionally, since the zero field splitting contributes a constant shift in energy, we can safely omit it by moving to the rotating frame. We are then left only with the second and third terms of Eq. (2.1).

2.2.1 *Mapping to model terms*

Next we will focus on mapping the remaining terms to operators to compose the set of terms $\mathcal{T}$ to use in our ES. In our modelling, the NVC spin is represented by the first logical qubit, with a further $|\{\chi\}|$ qubits, each representing a unique nuclear site, as discussed later in this section. As standard, we take the axis[2]of the NVC as parallel to the qubit's z-axis.

The first terms included come from the spin's precession about the magnetic field. It is usually assumed that the external, applied magnetic field is well-aligned with the spin qubit's z-axis; here we will treat this determination as the role of QMLA, i.e. we will endeavour to establish whether the x-, y-axis components of the magnetic field are important for describing the spin's dynamics. Then, we have

$$\mu_B g \mathbf{B} \cdot \mathbf{S} = \mu_B g (B_x \quad B_y \quad B_z) \cdot (\hat{S}_x \quad \hat{S}_y \quad \hat{S}_z) \rightarrow \alpha_x \hat{S}_x + \alpha_y \hat{S}_y + \alpha_z \hat{S}_z, \tag{2.2}$$

1 We invoke an inexact representation of high dimensional tensors here for ease of interpretation. For instance, the total nuclear spin operator exists in arbitrary dimension (depending on the number of sites modelled), but we present it simply as $\mathbf{I} = (\hat{I}_x \ \hat{I}_y \ \hat{I}_z)$ at each site to convey that we can separate the terms in the construction of models.

2 The axis connecting the ^{14}N with the lattice vacancy.

with $\alpha_i = \mu_B g B_i$. The spin's rotation terms to be included in QMLA's deliberations are therefore

$$\mathcal{T}_s = \{\hat{S}_x, \hat{S}_y, \hat{S}_z\} \tag{2.3}$$

Next, let's focus on the hyperfine coupling term. In general we sum over the nuclear sites $\{\hat{H}_i\}$, since the spin will interact with every nucleus to some extent; We show in [1] that a realistic system requires modelling a finite-size bath of $|\{\chi\}| \sim 15$ nuclei to capture the dynamics of interest, which is infeasible for complete characterisation via classical simulation, where we are limited to ~ 11 qubit calculations[3]. Instead, by focusing only on the *short-time* dynamics of the NVC, we can isolate the effects of dominant interactions, most notably with a single nearby ^{13}C. Indeed, by assigning a first qubit as representing the NVC spin, we can map the entire environment onto a generic second *environmental qubit*, representing the amalgamation of said interactions, though we can think of the two-qubit system as the NVC coupled with a single ^{13}C [68].

$$\mathbf{S} \cdot \sum_{\chi} (\mathbf{A}_\chi \cdot \mathbf{I}_\chi) \rightarrow \mathbf{S} \cdot \mathbf{A} \cdot \mathbf{I} \tag{2.4}$$

This clearly reduces the dimension of our approximation, the number of qubits, n_q from $n_q = 1 + |\{\chi\}|$ to $n_q = 2$, since now we only retain qubits for the NVC and the ^{14}N (which also represents the entire bath). The hyperfine tensor $\mathbf{A}$ consists of the hyperfine parameters, i.e. the strength of correspdoning interactions.

$$\mathbf{A} = \begin{pmatrix} A_\perp & 0 & 0 \\ 0 & A_\perp & 0 \\ 0 & 0 & A_\parallel \end{pmatrix}, \tag{2.5}$$

where $A_\perp$ is the non-axial hyperfine coupling term and $A_\parallel$ is the axial coupling term, since the axis of the NVC is used to define the z-axis for our qubits.

The total spin operators – of the NVC operating on the first logical qubit and the environmental qubit on the second – can be thought of as

$$\begin{aligned} \mathbf{S} &= (\hat{S}_x^{(1)} \quad \hat{S}_y^{(1)} \quad \hat{S}_z^{(1)}) \\ \mathbf{I} &= (\hat{I}_x^{(2)} \quad \hat{I}_y^{(2)} \quad \hat{I}_z^{(2)}) \end{aligned} \tag{2.6}$$

So we can write,

$$\begin{aligned} \mathbf{S} \cdot \mathbf{A} \cdot \mathbf{I} =& A_\perp \hat{S}_x \hat{I}_x + A_\perp \hat{S}_y \hat{I}_y + A_\parallel \hat{S}_y \hat{I}_y \\ &+ A_{xy} \left(\hat{S}_x \hat{I}_y + \hat{S}_y \hat{I}_x \right) \\ &+ A_{xz} \left(\hat{S}_x \hat{I}_z + \hat{S}_z \hat{I}_x \right) \\ &+ A_{yz} \left(\hat{S}_y \hat{I}_z + \hat{S}_z \hat{I}_y \right) \end{aligned} \tag{2.7}$$

Off-diagonal terms, referred to hereafter as *transverse* terms ($\hat{S}_i\hat{I}_j$ where $i \neq j$), are usually neglected [70]; here we will emply QMLA to determine whether their contributions are worthy of inclusion, although we consider only $\{\hat{S}_x\hat{I}_y, \hat{S}_x\hat{I}_z, \hat{S}_y\hat{I}_z\}$ for brevity. In total then, the hyperfine terms to be entertained by QMLA are

$$\mathcal{T}_{HF} = \begin{Bmatrix} \hat{S}_x\hat{I}_x, & \hat{S}_y\hat{I}_y, \hat{S}_z\hat{I}_z, \\ \hat{S}_x\hat{I}_y, & \hat{S}_x\hat{I}_z, \hat{S}_y\hat{I}_z \end{Bmatrix}. \tag{2.8}$$

Finally, combining Eq. (2.3) and Eq. (2.8), we have the full set of terms to incorporate into QMLA ES:

$$\mathcal{T}_{NV} = \begin{Bmatrix} \hat{S}_x, & \hat{S}_y, \hat{S}_z, \\ \hat{S}_x\hat{I}_x, & \hat{S}_y\hat{I}_y, \hat{S}_z\hat{I}_z, \\ \hat{S}_x\hat{I}_y, & \hat{S}_x\hat{I}_z, \hat{S}_y\hat{I}_z \end{Bmatrix}. \tag{2.9}$$

We introduce a shorthand notation to ease model representation for the remainder of this chapter. Recall that we have defined a two-qubit Hilbert space for model construction. Terms which affect only the spin act only on the first qubit, $\hat{S}_i = \hat{\sigma}_i \otimes \hat{\mathbb{1}}$, where $\hat{\sigma}_i$ is the Pauli operator giving rotation about the i-axis, and $\hat{\mathbb{1}}$ is the one-qubit identity matrix. Retaining the hyperfine notation, for the expectedly-dominant diagonal terms, we denote $\hat{A}_i = \hat{S}_i\hat{I}_i = \hat{\sigma}_i \otimes \hat{\sigma}_i$. We refer to the less-dominant off-diagonal terms as *transverse* terms, $\hat{T}_{kl} = \hat{S}_k\hat{I}_l = \hat{\sigma}_k \otimes \hat{\sigma}_l$. We can hence rewrite Eq. (2.9) as

$$\mathcal{T}_{NV} = \begin{Bmatrix} \hat{S}_x, & \hat{S}_y, \hat{S}_z, \\ \hat{A}_x, & \hat{A}_y, \hat{A}_z, \\ \hat{T}_{xy}, & \hat{T}_{xz}, \hat{T}_{yz} \end{Bmatrix}. \tag{2.10}$$

We also use a succinct representation for brevity, e.g. $\hat{S}_{xy}\hat{A}_z = \hat{S}_x + \hat{S}_y + \hat{A}_z$, where parameters $\alpha_x, \alpha_y, \alpha_z$ are implicitly assumed.

2.2.2 *Prior knowledge*

The spin-only terms, $\hat{S}_i$, are consequences of the magnetic field, which is usually assumed as aligned with z-axis, such the x-, y-axis compoenents are negligible; $\hat{S}_z$ is expected in the range $\sim 2 - 3$MHz. Likewise, the hyperfine terms, $\hat{A}_i$ are expected in the range of $\sim$ MHz [71], while in the *secular approximation* only the z-component is expected to contribute substantially [72]. The non axial hyperfine terms, i.e. the transverse terms $\hat{T}_{kl}$ are not usually included in effective models, but can be found of order $\sim \mathcal{O}(10\text{kHz})$ [73]. We utilise this prior understanding of the system to inform the parameter range used for training candidate models: for each of the terms

3 This limitation arises from the requirement to compute the total evolution of the global state, involving calculation of $e^{-i\hat{H}t}$, i.e. the characterisation of an n_q-qubit model depends on classical exponentiation of the $2n_q \times 2n_q$ Hamiltonian for each particle and experiment in classical likelihood estimation (CLE), which is a prohibitive expense.

in Eq. (2.10), we will adopt a normal prior distribution of 4 ± 1.5MHz. This is sufficiently specific to ensure the training subroutine operates in a physically meaningful – and likely appropriate – space, whlie also broad enough to allow for significant differences between expectation and reality, as well as supporting hypotheses where each parameter is zero, such that such negligible contributions can be identified.

2.2.3 *Experimental procedure*

We have decided to characterise the hyperfine interactions of the NVC; these are evident most strongly at very short timescales [74]. To isolate the effects of the hyperfine interactions, we run Hahn echo experiments, which are known to emphasise weak interactions. Hahn echo sequences attempt to decouple the spin's dynamics from the nuclear bath [75, 70, 74, 76, 77] providing a helpful platform for studying residual contributions of terms in Eq. (2.10). The NVC qubit undergoes a series of evolutions – either according to application of quantum logic gates or the natural evolution of the system interacting with its environmen We depict the stages of the experiment in Fig. 2.2, starting from the initialised $|0\rangle$ through to its final state which is read out through PL, both of which as described in Section 2.1.

In particular, the final state, $|\psi\rangle_5$, is read out, effectively by projection onto $|0\rangle$; we can interpret the PL after evolution time t as the likelihood that the NVC is found in $|0\rangle$ after evolution of its *true* Hamiltonian, $\hat{H}_0$. That is, we assign this projection as the quantity $\Pr(0|\hat{H}_0, t)$ (the likelihood), and it can be used within likelihood estimation in order to refine a candidate model $\hat{H}_j$, effectively[4] by changing the structure of $\hat{H}_j$ until $\Pr(0|\hat{H}_0, t) \approx \Pr(0|\hat{H}_j, t)\ \forall t$.

By varying the evolution time of the Hahn-echo sequence, we can map the likelihood against time, which we can view as capturing the dynamics of the NVC spin **??**. We vary time up to $t \sim 4\mu s$ in the short-time range in intervals of $\Delta t = 50ns$, so we have 425 data points. Importantly, the data for the studied NVC is taken once and analysed offline, i.e. QMLA does not have complete authority to design experiments to run on the NVC, although it can aim to choose the most informative t available in the predefined set; we will discuss the consequences of this restriction later in this chapter.

2.3 EXPLORATION STRATEGY

Finally, then, we are in a position to describe the specific implementation details which allow QMLA to study this NVC system. Recalling the terminology of QMLA from **??**, we design an exploration strategy (ES) specifically for the system under study. The ES will account for the details listed in this chapter so far, in summary:

- we attempt to assign a model, $\hat{H}'$, to the NVC;
  - we especially focus on its hyperfine interactions;

4 Of course this is a gross simplification of QHL which is described fully in **??**

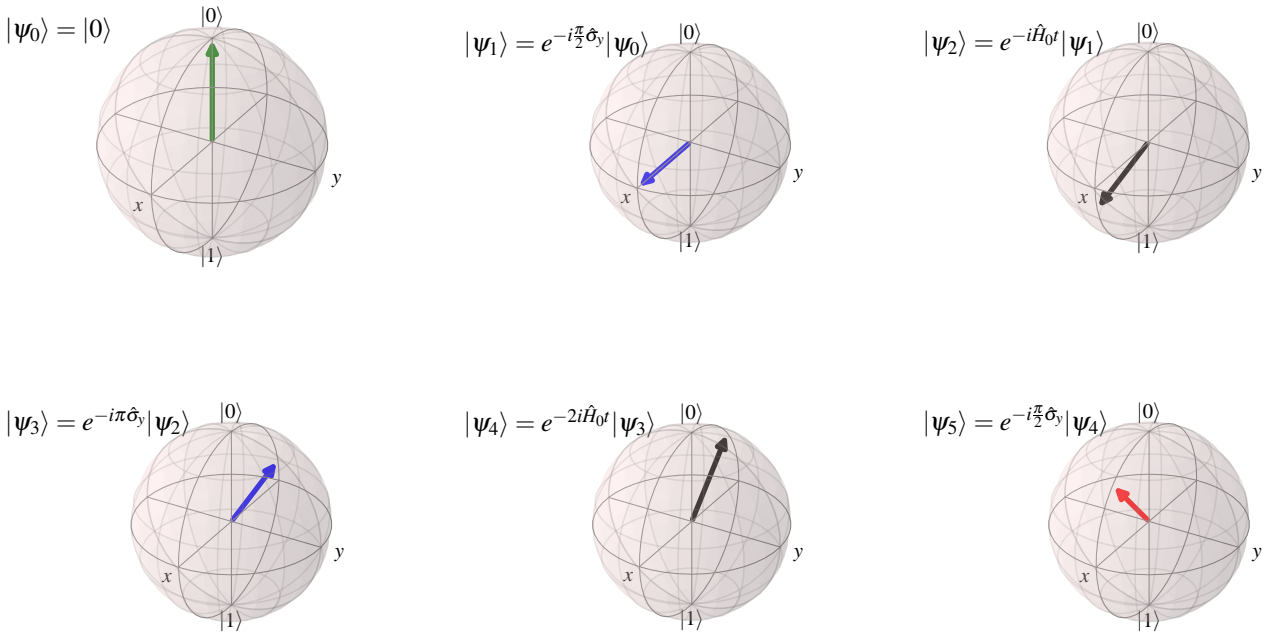


Figure 2.2: States of spin qubit at each stage of Hahn echo sequence.
The state of the NVC spin is initialised by a green laser into state $|\psi_0\rangle = |0\rangle$. We then apply a rotation about the y-axis (i.e. a $\frac{\pi}{2}$ pulse), yielding the state $|\psi_1\rangle = |+\rangle$. The system is then allowed to evolve according to its own $\hat{H}_0$ for t, $|\psi_2\rangle = e^{-i\hat{H}_0 t}|+\rangle$. We apply a second π pulse, this time for a π-rotation about the y-axis, $|\psi_3\rangle = e^{-i\pi\hat{\sigma}_y}e^{-i\hat{H}_0 t}|+\rangle$. Again the system evolves according to interactions with the environment, this time for $2t$. We apply a final $\frac{\pi}{2}$ pulse to rotate about the y-axis again, effectively either returning the spin to near the $|0\rangle$ axis, or near the $|1\rangle$ axis. Here $|\psi\rangle_5$ is roughly half way between $|0\rangle$ and $|+\rangle$, i.e. along the z-axis. The spin is read out from $|\psi\rangle_5$ via the NVC's photoluminescence. Here $\hat{H}_0 = 0.25\,\hat{\sigma}_y$ was evolved for $t = 0.5$ (arbitrary units), and the final state overlap with the initial state, i.e. the likelihood of measuring the spin in $|0\rangle$ is $\Pr(0|\hat{H}_0, t) = 0.865$.

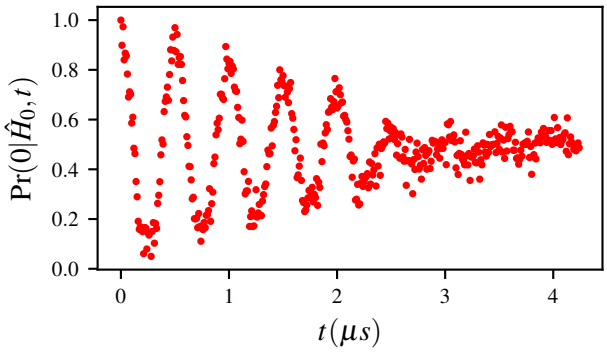


Figure 2.3: Raw data for the nitrogen-vacancy centre's dynamics. The y-axis shows the photoluminescence of the NVC, equivalently the likelihood $\Pr(0|\hat{H}_0, t)$.

- in particular, we use a 2-qubit approximation;
  - the first qubit represent the spin itself;
  - the second qubit represents the environment in which the NVC resides;
- we query the NVC by performing Hahn echo experiments (Fig. 2.2);
- the outcome of those experiments are thought of as the system's likelihoods (Fig. 2.3);
- likelihoods are used for the training of individual candidate models;
- candidate models are composed of the terms defined in Eq. (2.10).

As outlined in **??**, the central role of any ES is to specify the model generation procedure, which QMLA relies upon for deciding the next set of candidate models to test. In this case, we exploit some intuition and prior knowledge of how such systems work, to design a bespoke model generation subroutine: we can think of this as a midway point between the completely specified ESs used for identifying the underlying lattices from a prescribed set in **??**, and the entirely general genetic algorithm which does not restrict model generation, of Chapter 1. We use the standart structure of ETs introduced in **??**, where models are placed on consecutive branches, μ, with branches consolidated by complete pairwise comparisons, mediated through BFs, resulting in the selection of a single branch champion, $\hat{H}_{C(\mu)}$.

We use a *greedy search rule*: terms are added one-by-one to gradually increase the complexity of candidate models until terms are exhausted [31]. We break the ET into three distinct tiers, each corresponding to an intuitive degree of complexity: the first tier involves the spin-only terms, $\mathcal{T}_1 = \{\hat{S}_i\}$; the second considers the hyperfine terms, $\mathcal{T}_2 = \{\hat{A}_i\}$; the final tier the transverse terms, $\mathcal{T}_3 = \{\hat{T}_i\}$. Within each tier, terms are added greedily to the previous branch's champion, $\hat{H}_{C(\mu)}$. So, the first branch is given by $\mu = \{\hat{S}_x, \hat{S}_y, \hat{S}_z\}$; say $\hat{H}_{C(\mu)} = \hat{S}_y$, then $\mu + 1$ determines that $\hat{S}_x, \hat{S}_z$ are not yet considerd, so it constructs the models $\hat{S}_{x,y}, \hat{S}_{y,z}$, e.g. Fig. 2.4. After

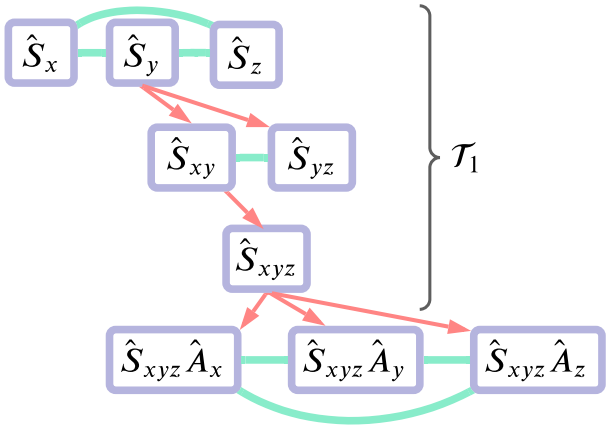


Figure 2.4: Greedy model search. Models (purple) are placed on branches, trained and consolidated (green) as in **??**, with the branch champion spawning (red) candidates to place on the subsequent branch. Branches are grouped in tiers, correspdoning to levels of approximation: the first tier of the model generation strategy is shown, where $\mathcal{T}_1 = \{\hat{S}_x, \hat{S}_y, \hat{S}_z\}$ is explored. The final champion from the first tier seeds the second tier.

exhausting all tiers, we consolidate the set of branch champions, $\mathbb{H}_C = \{\hat{H}_{C(\mu)}\}$, to determine the best model considered globally, $\hat{H}'$.

Clearly, this growth rule is partially deterministic, insofar as some models are gauranteed to be considered, while others are not reachable; indeed, the space of available models are heavily constrained, in particular models in later tiers will always involve all of the tier 1 models, e.g. $\hat{S}_x\hat{A}_y$ does not occur naturally. We account for this shortcoming with a final redcibility test, triggered if any of the parameters of $\hat{H}'$ are potentially negligible, i.e. are within one standard deviation of 0. This test simply removes such low-parmeter terms from $\hat{H}'$, giving a reduced global champion, $\hat{H}'_r$, and computes the BF between $\hat{H}', \hat{H}'_r$: if the BF indicates strong evidence in favour of the reduced version, we replace the champion model, $\hat{H}' \leftarrow \hat{H}'_r$. In effect, we thus verify the statistical significance of each term included in $\hat{H}'$.

The total model in Eq. (2.1) supports $N_t = 1 + 3 + 6|\chi| + 3 + 3|\chi| = 7 + 9|\chi|$ terms, which we reduced to a space of $N_t = 3 + 6|\chi| = 9$ through several approximations; even so, the remainng 2^9 permitted models were reduced further by building the logic of this ES from our intuition aroud existing knowledge of typical NVC systems, such the ES will only ever consider < 20 models per instance. The described ES seems overly perscriptive, but should be viewed as a first attempt at a generalisable approach: essentially we can view the tiers of the greedy search as characterising the system at various approximations, e.g. the first tier examines one-qubit terms, while subsequent tiers inspect 2-qubit terms. We can envision future work where the greedy

search is gradually extended to less rigid approximations, enabling study of more complex quantum systems. This leads to some important remarks:

1. Realistic, near-term applications of QMLA can not be thought of as a solution to black-box characterisation problems: it must be used in conjunction with domain expertise for the system under study.
2. While this test-case yields promising results, the outcome of QMLA here may not be especially insightful, since the available model terms were so deliberately constrained – we demonstrate a use-case in Chapter 3 where a broader scope is enabled in simulation.

A common charge against QMLA supposes to first write down the most complex model, train it fully, and then infer which terms are negligible, in a similar process to the champion reduction test outlined here. While this may be feasible in the case described here, with $N_t = 9$ and a closed term set, this would scale poorly: adding just a second nuclear site increases the model search to a space of $N_t = 15$. Models of higher cardinality ($|\vec{\alpha}|$) demand higher N_e, N_p to train well, so immediately trainng the most involved model would require infeasible resources[5], and risks significantly overfitting the data. It seems more appropriate to work "from the ground up", testing terms and keeping them only when justified, instead of training all terms and attempting to decouple their effects post-hoc.

2.3.1 *Test in simulation*

Before considering the real experimental data (Fig. 2.3), we first test the ES in simulation under ideal conditions. That is, we assume the ability to prepare arbitrary probe states, and use a random probe set (see **??**), and use the full expectation value as the likelihood, $|\langle\psi|e^{-i\hat{H}_j t}|\psi\rangle|^2$. Of course, this is infeasible since we presume access to the full state at the time of measurement, but this can be seen as a best-case scenario for this application, because the realistic case loses information by tracing out the environmental qubit at measurement. We vary the target $\hat{H}_0$, among a series of ten models, which are all valid models achievable by the ES.

Varying $\hat{H}_0$, allowing arbitrary probe preparation and particle guess heuristic (PGH) EDH, i.e. case (i) from [1].

2.4 EXPERIMENT DESIGN CONSTRAINTS

Moving to analyse the experimental setup, there are a number of constraints which we must account for in training models. Firstly, the $\pi/2$-pulse applied to the prepared qubit ($|\psi_0\rangle \rightarrow |\psi_1\rangle$ in Fig. 2.2) means that the state before evolution is always $|+\rangle$ in the computational basis; this

5 Note: in the case studies presented in this thesis, it was found that the same resources were sufficient for the simplest and most complex models, due to the relatively small number of terms therein. We expect for larger models, e.g. $|\vec{\alpha}| > 10$, that the resources allocated ought to be proportional to the cardinality, which is an in-built option in the QMLA software.

is a severe limitation on model training, as we saw in **??**. Moreover, this puts a bias on the interactions QMLA is likely to identify: we show in **??** how QHL performs in training the same model using (i) the probe set available experimentally; (ii) a more general (random) probe set. This has a noteworthy impact on the outcome of this study, since this suppression of terms means we are more likely to find some interactions than others, so we must add this caveat to the champion model.

The experiment was run until the NVC was deemed to have decohered, so Fig. 2.3 ceases at $t_{\text{max}} \sim 4\mu s$. As discussed in **??**, usually it is helpful to allow a experiment design heuristic (EDH) to choose the experimental controls, including the evoltuion time, t, against which the model is trained at each experiment; the default particle guess heuristic (PGH) attempts to select t at the upper boundary of times where the model is expected to be predictive, such as to maximise the information gained by the experiment (see **??**). Here, however, we can not allow the EDH to select arbitrary t, since we do not have data beyond t_{max}.

Clearly, we require a custom EDH to account for the constraints outlined, with the following considerations:

1. We may only assume access to the probe $|+\rangle$ on the spin qubit
   (a) we further assume the environmental spin is polarised by the same microwave pulse, such that the global probe available is $|\psi\rangle = |+\rangle\,|+'\rangle$, with $|+'\rangle = \frac{|0\rangle + e^{i\phi}|1\rangle}{\sqrt{2}}$ and ϕ is random.
2. We can not allow the choice of any t
   (a) Any $t > t_{\text{max}}$, arising from a thin parameter distribution, must be mapped to some $0 < t \leq t_{\text{max}}$.
   (b) All nominated t must be mapped to the nearest available t in the dataset so that the likelihoods are as close as possible to simulating the true system.
3. Much of the physics of interest occurs at relatively high times, i.e. because the rotation (MHz) terms dominate, the decay of the peaks can be seen as evidence of the bath, notably through hyperfine terms in the model.
   (a) We therefore wish to enforce that all models are trained on those data ($t \geq 2\mu s$), even if their parameter distribution is insufficiently narrow to yield those times naturally.

Accounting for these, we construct an EDH which mixes the robust, adaptive nature of PGH, useful for refining an initially broad $\Pr(\vec{\alpha})$, with a primitive, linear time-selection, useful to ensure the trained parameters at least attempt to account for the physics we are actually interested in. That is, with each model trained for N_e experiments, we train according to the standard PGH for the first $N_e/2$, but force the training to mediate over the available data for the latter $N_e/2$.

Model	Experiment		Simulation	
	Wins	R^2	Wins	R^2
$\hat{S}_{y,z}\hat{A}_z$	9	0.8	1	0.26
$\hat{S}_y\hat{A}_{x,z}$	2	0.63		
$\hat{S}_{x,y,z}\hat{A}_z$	45	0.86	61	0.97
$\hat{S}_{x,y,z}\hat{A}_y$			1	-0.54
$\hat{S}_{x,y,z}\hat{A}_{x,y}$	3	0.81		
$\hat{S}_{x,y,z}\hat{A}_{y,z}$	14	0.83	10	0.96
$\hat{S}_{x,y,z}\hat{A}_{x,z}$	6	0.64	15	0.99
$\hat{S}_{x,y,z}\hat{A}_{x,y,z}$	2	0.72	5	0.97
$\hat{S}_{x,y,z}\hat{A}_{x,z}\hat{T}_{xz}$			1	0.68
$\hat{S}_{x,y,z}\hat{A}_{x,y,z}\hat{T}_{xz}$			5	0.77
$\hat{S}_y\hat{A}_{x,y,z}\hat{T}_{xy,xz,yz}$	2	0.31		
$\hat{S}_{x,y,z}\hat{A}_{x,y,z}\hat{T}_{xy,xz}$	4	0.67	1	0.32

Table 2.1: QMLA win rates and R^2 for models based on experimental data and simulations. We state the number of QMLA instances won by each model and the average R^2 for those instances as an indication of the predictive power of winning models.

2.5 RESULTS

We apply the ES described in Section 2.3 to the raw data of Fig. 2.3; the results are summarised in Fig. 2.5.

We first focus on the overall outcomes: the most blunt figure of merit of interest is simply whether QMLA overfits or underits the true parameterisation. In preliminary analysis we run 500 instances with varying $\hat{H}_0$, and where the cardinality of $\hat{H}_0$ ranges, so we can broadly gauge the tendency towards over- and under-fitting: we see that in $\sim$ 50% of instances the correct cardinality is found, rising to $\sim$ 86% by allowing one fewer (additional) terms, Fig. 2.5**(b)**. Moreover, the champion models' are highly predictive: the median coefficient of determination between the systems' and corresponding champion models' data is $R^2 = 0.84$.

Then, considering the performance of the algorithm on whole, we perform runs of 100 instances on the exerpimental data as well as simulated data, where the simulation assumes[6] $\hat{H}_0 = \hat{S}_{xyz}\hat{A}_z$, The set of models selected most frequently are shown in Fig. 2.5**(c)**, and each model is trained with $N_e = 1000, N_p = 3000$, with the volumes of those models (in

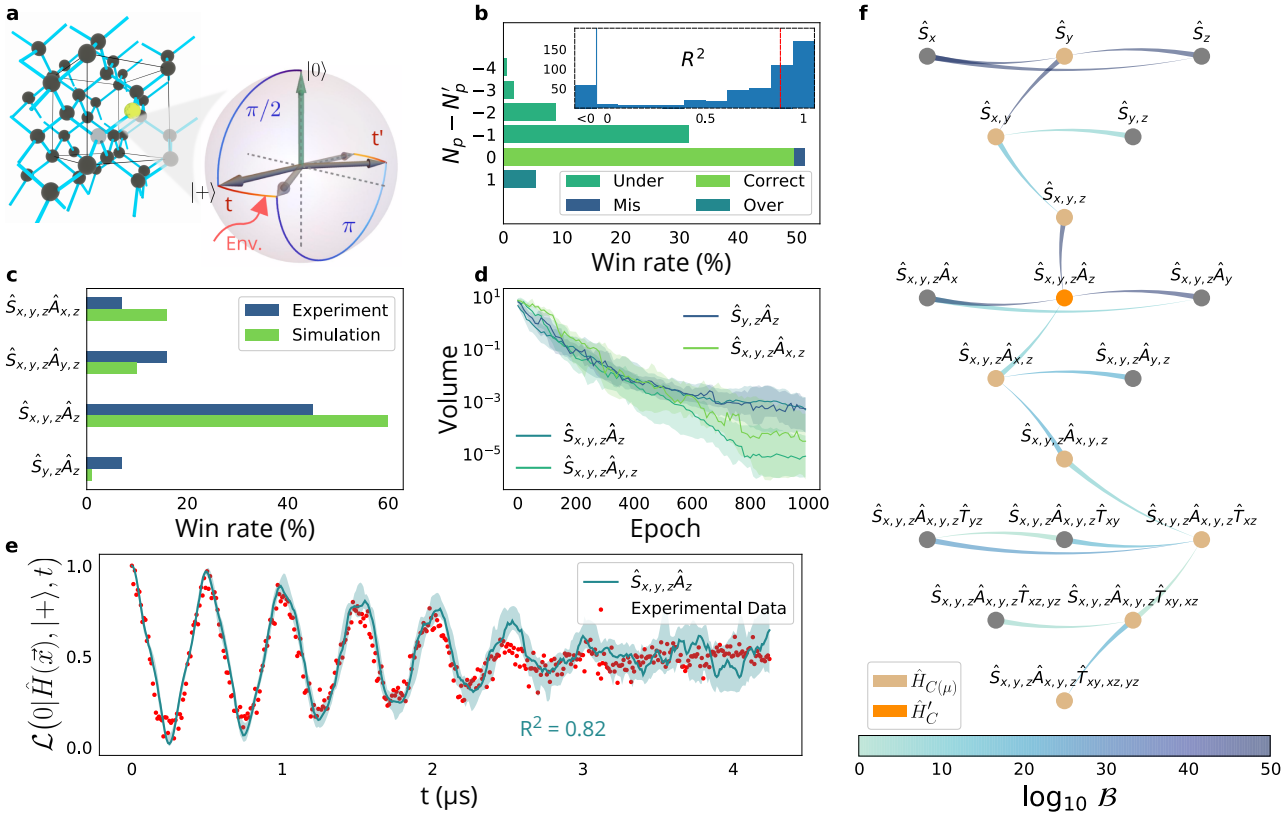


Figure 2.5: **a**, The carbon lattice providing the outer environment for the NV centre, along with the time evolution of the electron spin state (represented on a Bloch sphere) during the pulses for the Hahn-echo sequences. The final $\pi/2$ pulse is omitted. **b**, Simulation of 500 independent QMLA instances, where $\hat{H}_0$ is chosen randomly. The win rate is reported against the difference $(N_p - N'_p)$ between the number of parameters in $\hat{H}'$ and $\hat{H}_0$, respectively. The *under-parameterised (over-parameterised)* class refers to models with less (more) parameters than $\hat{H}_0$. *Correct* indicates that exactly $\hat{H}_0$ was found. The *mis-parameterised* class groups models with the same parametrisation cardinality as $\hat{H}_0$, but different Hamiltonian terms. **Inset**, Histogram of occurrences of R^2 values for each retrieved $\hat{H}'$ against a sampling of datapoints from $\hat{H}_0$, with median $R^2 = 0.84$ (red dotted line). **c**, Win rates of top four models for 100 QMLA instances, against both simulated and experimental data. Simulations use $\hat{H}_0 = \hat{S}_{x,y,z}\hat{A}_z$. **d**, Total volume spanned by the parameters' prior across epochs, for the models in **c**. Shaded areas indicate 66% credible regions. **e**, Simulated likelihoods reproduced by the model with the highest win rate ($\hat{S}_{x,y,z}\hat{A}_z$, turquoise), compared with corresponding NV-centre system experimental data (red-dots, extracted from the observed PL of the first Hahn-echo decay). Error bars smaller than the dots (see Methods). **f**, A single QMLA instance against experimental data in **e**, depicted as a . The thin end of each edge points to the favoured model; the colour of the edges maps $\log_{10} \mathcal{B}$ as in the bar legend at the bottom. Layer champions are in light brown, whereas the global champion $\hat{H}'$ is in orange.

the experimental case) shown in Fig. 2.5**(d)**. In particular, the most prominent models, $\{\hat{S}_{x,y,z}\hat{A}_z, \hat{S}_{x,y,z}\hat{A}_{y,z}, \hat{S}_{x,y,z}\hat{A}_{x,z}, \hat{S}_{y,z}\hat{A}_z\}$ are found collectively in 74% (87%) of instances on the experimental (simulated) data; the win rates and R^2 of all models (which won at least one instance) are reported in Table 2.1. It is noteworthy that even in the simulated case, the same models mislead QMLA: this suggests that the resultant physics from these models is substantially similar to that of the true model[7]. These models are defensible with respect to the descriptions of Section 2.2, since in each case they detect the interaction between the spin qubit and the environmental qubit, i.e. the hyperfine terms $\hat{A}_i$, especially $\hat{A}_z$ which occurs in 97% (99%) of champions. We discuss some physical insights from these results in Section 2.5.1.

The most frequently found model, $\hat{H}' = \hat{S}_{x,y,z}\hat{A}_z$, is found in 45% (61%) of instances: we show its attempt to reproduce the dynamics of Fig. 2.3 in Fig. 2.5**(e)**, showing excellent agreement with the raw data, with $R^2 = 0.82$. This serves as an essential sanity check: we can intuitively see that QMLA has distilled a model which captures at least *some* of the most important physical interactions the target NVC system is subject to; otherwise we would not see such clear overlap between the prediced and true dynamics.

Finally we display the model search as a in Fig. 2.5**(f)**, where models are represented on nodes on the graphs layers (equivalent to ET branches), and their parents are resident on the branch immediately above their own. Comparisons between models, $\hat{H}_i, \hat{H}_j$, are shown as edges between nodes on the graph, coloured by the strength of evidence of the outcome, the BF, B_{ij}. Each layer, μ, nominates their branch champion, $\hat{H}_{C(\mu)}$; the set of branch champions are consolidated to determine the global champion, $\hat{H}'_C$.

2.5.1 *Analysis*

Considering the runs summarised in Fig. 2.5, here we will present some furhter perspecitives. Fig. 2.6 first details all models considered in the 200 instances comprising the experimental and simulated QMLA runs, as well as the win rate of each model. This ES is designed to study a small subspace of the overall available space: only 40 unique models are constructed. We highlight a number of *credible* models which we deem especially valid approximations of the target system, i.e. which contain the most viable approximations.

Fig. 2.7 shows the reproduction of dynamics of the top four models from both simulated and experimental runs. We see that each model faithfully captures the essential dynamics arising from the respective target systems; this alone is insufficient to conclude that the true model has been identified, but serves as a valuable *sanity-check*, convincing us that the output of QMLA is at least a sensible approximation of $\hat{H}_0$, if not the absolutely true model.

6 Here we work backwards by setting the target model as that which QMLA deemed most appropriate for the available data. We posit that this choice is arbitrary and doesn't fundamentally change the discussion of this chapter, merely aiding in analysing the performance of the algorithm with respect to a concrete example.

7 Alternatively, that the same systematic error misdirects the search in both cases.

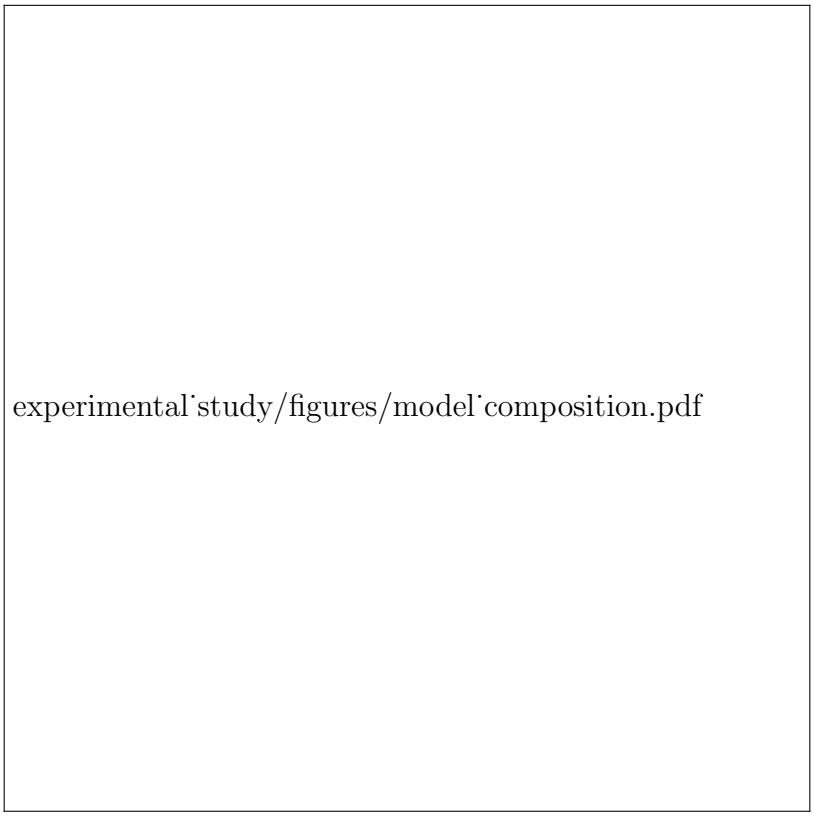
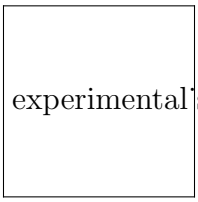
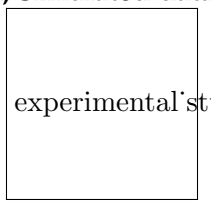


Figure 2.6: **Left**: map of the various Hamiltonian terms included in each of the possible 40 candidate models explored by QMLA during any of the 100 instances on either simulated or experimental data. The explicit form of each operator can be inferred from **??**. IDs of candidate models are on the vertical axis, and labels for the terms on the horizontal axis. The true model $\hat{H}_0$ for the simulated case is highlighted in green and a subset of credible models in blue, i.e. models which may reasonbaly be expected to describe the NVC from theoretical arguments. **Right**: wins for each of the candidate models as above out of 100 independent QMLA runs, reported as a histogram for cases adopting simulated data (empty bars) or the experimental dataset.



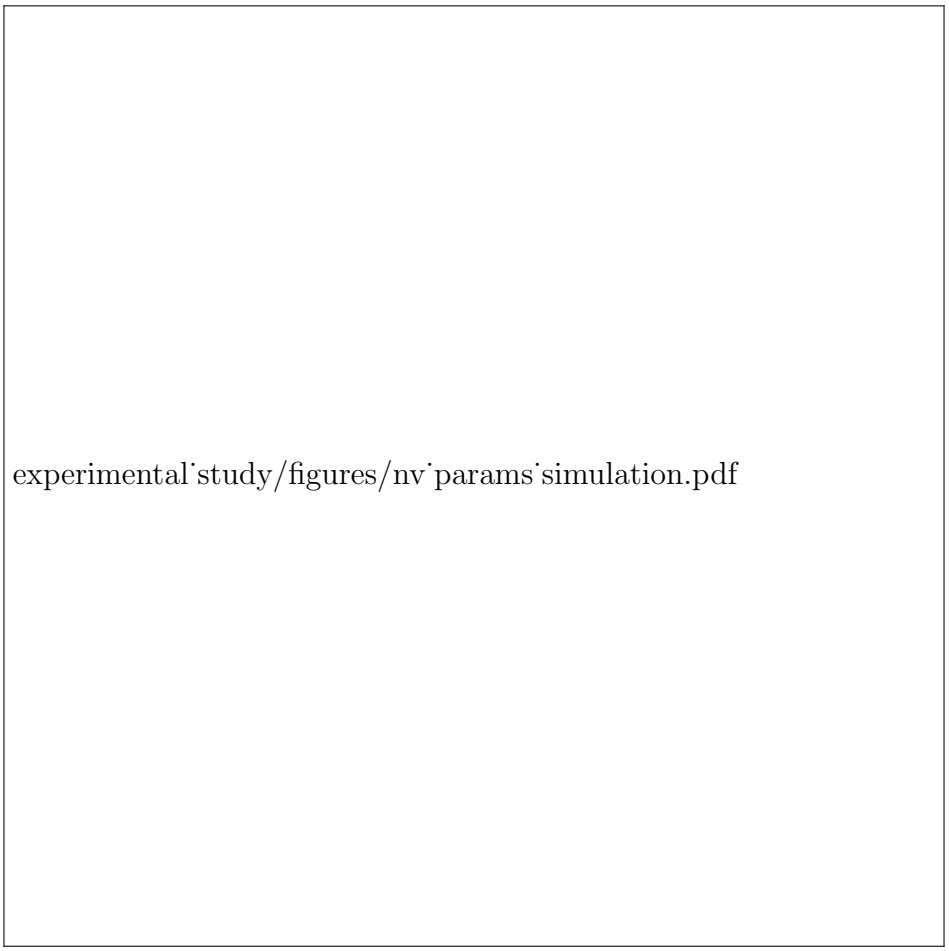
a, Simulated data



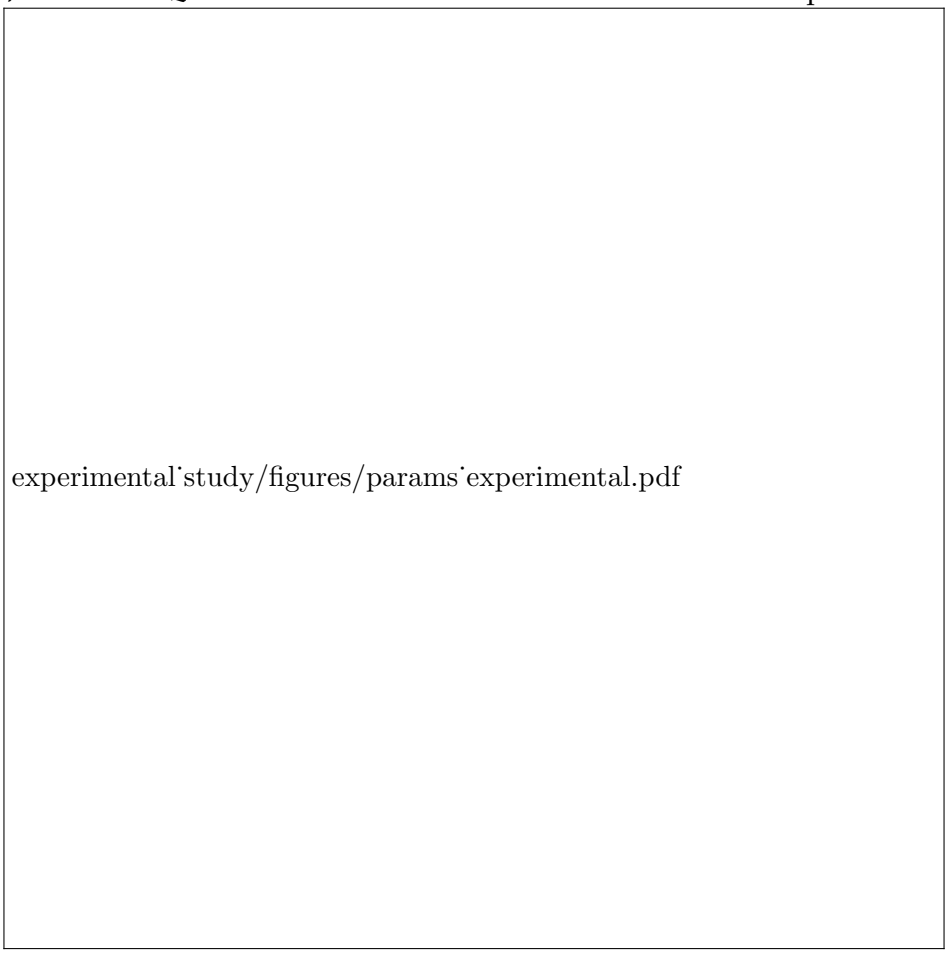
b, Experimental data

Figure 2.7: Dynamics reproduced by various QMLA champion models for simulated/experimental data. Expectation values are shown on y–axis with time on x–axis. Red dots give the true dynamics of $\hat{H}_0$, while the blue lines show the median reconstruction by $\hat{H}'$ from all the instance where that model was deemed $\hat{H}'$, with light blue showing the standard deviation of those likelihoods. $\hat{H}'$ is listed on top of each plot; the number of instances won by each model can be read from Table 2.1.

The key insight promised by QMLA is to identfiy the interactions present in the studied system. In Fig. 2.8 we show the number of times each of the terms permitted (Eq. (2.10)) is included in the champion model; we also show the distribution of parameter estimates for those terms. From the simulated case, we see that those terms which are in $\hat{H}_0$, i.e. $\mathcal{T}_0$, are found in almost all instances; furthermore, while some terms are erroneoulsy found in $> 40\%$ of instances, they are found with less than a quarter of the frequency of the true terms, so these may be reasonably ruled out in post-processing the QMLA results. The inaccurate terms found most often are seen to have (almost) negligible parameters: in conjunction with domain expertise, users can determine whether the inclusion of these terms are meaningful or simply artefacts of slight overfitting. In the experimental run, on the other hand, we see a similar gulf in frequency between some terms. Namely, $\{\hat{S}_x, \hat{S}_y, \hat{S}_z, \hat{A}_z\}$ are found in 50+ more instances than all others: we therefore conclude that those terms contribute most strongly to the NVC. The parameters found for these terms are of the same order of magnitude as we would have predicted originally, but disagreee in the precise value. For example, the rotation about the y-axis, $\hat{S}_y$, has frequency 5.7MHz, roughly twice the predicted value of 2.7MHz.



a, Simluated QMLA instances. Red dotted lines show the true parameters.



b, Experimental QMLA instances.

3 EXTENSION TO MANY QUBITS

3.1 GENETIC ALGORITHM

APPENDIX

A

FIGURE REPRODUCTION

Most of the figures presented in the main text are generated directly by the QMLA framework. Here we list the implementation details of each figure so they may be reproduced by ensuring the configuration in Table A.1 are set in the launch script. The default behaviour of QMLA is to generate a results folder uniquely identified by the date and time the run was launched, e.g. results can be found at the *results directory* qmla/Launch/Jan_01/12_34. Given the large number of plots available, ranging from high-level run perspective down to the training of individual models, we introduce a plot_level $\in \{1, ..., 6\}$ for each run of QMLA: higher plot_level informs QMLA to generate more plots.

Within the results directory, the outcome of the run's instances are stored, with analysis plots broadly grouped as

4. evaluation: plots of probes and times used as the evaluation dataset.
5. single_instance_plots: outcomes of an individual QMLA instance, grouped by the instance ID. Includes results of training of individual models (in model_training), as well as sub-directories for anlaysis at the branch level (in branches) and comparisons.
6. combined_datasets: pandas dataframes containing most of the data used during analysis of the run. Note that data on the individual model/instance level may be discarded so some minor analyses can not be performed offline.
7. exploration_strategy_plots plots specifically required by the ES at the run level.
8. champion_models: analysis of the models deemed champions by at least one instance in the run, e.g. average parameter estimation for a model which wins multiple instances.
9. performance: evaluation of the QMLA run, e.g. the win rate of each model and the number of times each term is found in champion models.
10. meta analysis of the algorithm' implementation, e.g. timing of jobs on each process in a cluster; generally users need not be concerned with these.

In order to produce the results presented in this thesis, the configurations listed in Table A.1 were input to the launch script. The launch scripts in the QMLA codebase consist of many configuration settings for running QMLA; only the lines in snippet in Listing A.1 need to be set according to altered to retrieve the corresponding figures. Note that the runtime of QMLA grows quite quickly with N_E, N_P (except for the AnalyticalLikelihood ES), especially for the entire QMLA algorithm; running QHL is feasible on a personal computer in < 30 minutes for $N_e = 1000; N_p = 3000$.

```
#!/bin/bash
```

```
###############
# QMLA run configuration
###############
num_instances=1
run_qhl=1 # perform QHL on known (true) model
run_qhl_mulit_model=0 # perform QHL for defined list of models.
exp=200 # number of experiments
prt=1000 # number of particles

###############
# QMLA settings
###############
plot_level=6
debug_mode=0

###############
# Choose an exploration strategy
###############

exploration_strategy='AnalyticalLikelihood'
```

Listing A.1: "QMLA Launch scipt"

Figure	Exploration Strategy	Algorithm	N_E	N_P	Data
??	AnalyticalLikelihood	QHL	500	2000	Nov_16/14_28
??	DemoIsing	QHL	500	5000	Nov_18/13_56
??	DemoIsing	QHL	1000	5000	Nov_18/13_56
??	DemoIsing	QHL	1000	5000	Nov_18/13_56
??	IsingLatticeSet	QMLA	1000	4000	Nov_19/12_04
	IsingLatticeSet	QMLA	1000	4000	Sep_30/22_40
??	HeisenbergLatticeSet	QMLA	1000	4000	Oct_22/20_45
	FermiHubbardLatticeSet	QMLA	1000	4000	Oct_02/00_09
Fig. 1.5	DemoBayesFactorsByFscore	QMLA	500	2500	Dec_09/12_29
	DemoFractionalResourcesBayesFactorsByFscore	QMLA	500	2500	Dec_09/12_29
	DemoBayesFactorsByFscore	QMLA	1000	5000	Dec_09/12_29
	DemoBayesFactorsByFscoreEloGraphs	QMLA	500	2500	Dec_09/12_29

Table A.1: Implementation details for figures used in the main text.

B

EXAMPLE EXPLORATION STRATEGY RUN

A complete example of how to run the ;sqmla framework, including how to implement a custom ES, and generate/interpret analysis, is given.

BIBLIOGRAPHY

[1] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[2] John Henry Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[3] Sean Luke. 1 essentials of metaheuristicsl. 1.

[4] Lothar M Schmitt. Theory of genetic algorithms. *Theoretical Computer Science*, 259(1-2):1–61, 2001.

[5] Nathan Wiebe, Christopher Granade, Christopher Ferrie, and David Cory. Quantum hamiltonian learning using imperfect quantum resources. *Physical Review A*, 89(4):042314, 2014.

[6] Eyal Bairey, Itai Arad, and Netanel H Lindner. Learning a local hamiltonian from local measurements. *Physical review letters*, 122(2):020504, 2019.

[7] Burnham KP Anderson DR. Model selection and multimodel inference: a practical information-theoretic approach, 2002.

[8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[9] Christopher Granade, Christopher Ferrie, Steven Casagrande, Ian Hincks, Michal Kononenko, Thomas Alexander, and Yuval Sanders. QInfer: Library for statistical inference in quantum information, 2016.

[10] Antonio A. Gentile, Brian Flynn, Sebastian Knauer, Nathan Wiebe, Stefano Paesani, Christopher E. Granade, John G. Rarity, Raffaele Santagati, and Anthony Laing. Learning models of quantum systems from experiments, 2020.

[11] Arpad E Elo. *The rating of chess players, past and present*. Arco Pub., 1978.

[12] FIFA.com. Who we are - news - 2026 fifa world cup™: Fifa council designates bids for final voting by the fifa congress, Jun 2018.

[13] Christof Neumann, Julie Duboscq, Constance Dubuc, Andri Ginting, Ade Maulana Irwan, Muhammad Agil, Anja Widdig, and Antje Engelhardt. Assessing dominance hierarchies: validation and advantages of progressive evaluation with elo-rating. *Animal Behaviour*, 82(4):911–921, 2011.

[14] Lars Magnus Hvattum and Halvard Arntzen. Using elo ratings for match result prediction in association football. *International Journal of forecasting*, 26(3):460–470, 2010.

[15] Marcus W Doherty, Neil B Manson, Paul Delaney, Fedor Jelezko, Jörg Wrachtrup, and Lloyd CL Hollenberg. The nitrogen-vacancy colour centre in diamond. *Physics Reports*, 528(1):1–45, 2013.

[16] Gordon Davies and MF Hamer. Optical studies of the 1.945 ev vibronic band in diamond. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 348(1653):285–298, 1976.

[17] J Meijer, B Burchard, M Domhan, C Wittmann, Torsten Gaebel, I Popa, F Jelezko, and J Wrachtrup. Generation of single color centers by focused nitrogen implantation. *Applied Physics Letters*, 87(26):261909, 2005.

[18] AM Edmonds, UFS D'Haenens-Johansson, RJ Cruddace, ME Newton, K-MC Fu, C Santori, RG Beausoleil, DJ Twitchen, and ML Markham. Production of oriented nitrogen-vacancy color centers in synthetic diamond. *Physical Review B*, 86(3):035201, 2012.

[19] A Lenef and SC Rand. Electronic structure of the n-v center in diamond: Theory. *Physical Review B*, 53(20):13441, 1996.

[20] Benjamin Smeltzer, Lilian Childress, and Adam Gali. 13c hyperfine interactions in the nitrogen-vacancy centre in diamond. *New Journal of Physics*, 13(2):025021, 2011.

[21] Heinz-Peter Breuer, Francesco Petruccione, et al. *The theory of open quantum systems*. Oxford University Press on Demand, 2002.

[22] MS Blok, Cristian Bonato, ML Markham, DJ Twitchen, VV Dobrovitski, and R Hanson. Manipulating a qubit through the backaction of sequential partial measurements and real-time feedback. *Nature Physics*, 10(3):189–193, 2014.

[23] Adam Gali, Maria Fyta, and Efthimios Kaxiras. Ab initio supercell calculations on nitrogen-vacancy center in diamond: Electronic structure and hyperfine tensors. *Physical Review B*, 77(15):155206, 2008.

[24] MV Gurudev Dutt, L Childress, L Jiang, E Togan, J Maze, F Jelezko, AS Zibrov, PR Hemmer, and MD Lukin. Quantum register based on individual electronic and nuclear spin qubits in diamond. *Science*, 316(5829):1312–1316, 2007.

[25] P-Y Hou, L He, F Wang, X-Z Huang, W-G Zhang, X-L Ouyang, X Wang, W-Q Lian, X-Y Chang, and L-M Duan. Experimental hamiltonian learning of an 11-qubit solid-state quantum spin register. *Chinese Physics Letters*, 36(10):100303, 2019.

[26] L Childress, MV Gurudev Dutt, JM Taylor, AS Zibrov, F Jelezko, J Wrachtrup, PR Hemmer, and MD Lukin. Coherent dynamics of coupled electron and nuclear spin qubits in diamond. *Science*, 314(5797):281–285, 2006.

[27] L_G Rowan, EL Hahn, and WB Mims. Electron-spin-echo envelope modulation. *Physical Review*, 137(1A):A61, 1965.

[28] Forrest T Charnock and TA Kennedy. Combined optical and microwave approach for performing quantum spin operations on the nitrogen-vacancy center in diamond. *Physical Review B*, 64(4):041201, 2001.

[29] Antonio Andreas Gentile. *Operating practical quantum devices in the pre-threshold regime*. PhD thesis, University of Bristol, 2020.

[30] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.