

# Gaussian processes for choosing laser parameters for driven, dissipative Rydberg aggregates

C D B Bentley and A Eisfeld

Max Planck Institute for the Physics of Complex Systems, Nöthnitzer Strasse 38,  
01187 Dresden, Germany

E-mail: [cbentley@pks.mpg.de](mailto:cbentley@pks.mpg.de), [eisfeld@pks.mpg.de](mailto:eisfeld@pks.mpg.de)

**Abstract.** To facilitate quantum simulation of open quantum systems at finite temperatures, an important ingredient is to achieve thermalization on a given time-scale. We consider a Rydberg aggregate (an arrangement of Rydberg atoms that interact via long-range interactions) embedded in a laser-driven atomic environment. For the smallest aggregate (two atoms), suitable laser parameters can be found by brute force scanning of the four tunable laser parameters. For more atoms, however, such parameter scans are too computationally costly. Here we apply Gaussian processes to predict the thermalization performance as a function of the laser parameters for two-atom and four-atom aggregates. These predictions perform remarkably well using just 1000 simulations, demonstrating the utility of Gaussian processes in an atomic physics setting. Using this approach, we find and present effective laser parameters for generating thermalization, the robustness of these parameters to variation, as well as different thermalization dynamics.

## 1. Introduction

One often encounters a situation where the outcome of an experiment depends on many control parameters that can be varied over a large range. Usually, one is interested in achieving a particular outcome. This scenario emerges in both experimental and theoretical settings.

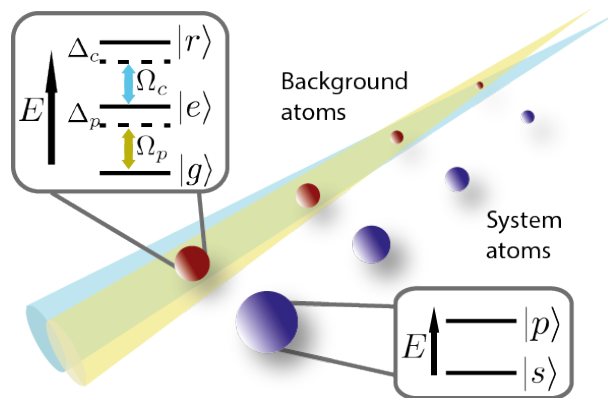
Achieving the target outcome can be a demanding task, in particular when the space of tunable parameters is high dimensional, and when there is no simple dependence of the outcome on the parameters. If one is only interested in the optimal outcome, then various approaches have been developed for this task (e.g. the methods in [1–3]). However, one is typically also interested in the ‘stability’ of the outcome, not (just) the parameter values that give the optimal outcome. Additionally, one would like to know the regions of parameter space where one is close to the desired outcome. We quantify ‘closeness’ to the target outcome by a *cost function* (or simply *cost*); when we speak about ‘close to the desired outcome’ we mean loosely that the result is below a certain value of the cost. Similarly, when we speak about ‘optimal’ we mean the set of parameters that gives a result that is closest to the desired outcome; yielding the

minimal cost. For convenience, we will denote the manifold of the cost function over parameter space as the *cost landscape*.

In the ideal case one would like to know the full cost landscape: the respective values of the cost for all choices of the tunable parameters. However, analytic descriptions of the cost landscape can rarely be found for complex systems. Also, brute force scans of the parameter space are often precluded by the time required to perform an experiment/calculation for a single set of parameters. One powerful approach is based on Gaussian processes (GPs) [4]. GPs are not only able to search for an optimum but also provide information on the full cost landscape. This is performed by regression: a Gaussian process provides a prediction of the cost landscape by fitting known data, using Bayesian updates to prior assumptions for the model. This method can provide predictions of the full cost landscape from few data points, and has been applied to predict interatomic potentials [5,6], and the related kernel ridge regression method has also been applied to predict properties of atoms in molecules [7,8].

Knowledge of the predicted cost landscape also provides valuable information for choosing subsequent parameters for the experiment/simulation 'well', to use resources efficiently. This is related to reinforcement learning, where previous 'policies of action' inform the following action policies, with some trade-off between exploring new policies and exploiting those that have worked well previously. Reinforcement learning has been applied recently for quantum control [9–12] and even to design quantum optics experiments [13]. The idea of using known points in the cost landscape to choose subsequent parameters for evaluation is known as active learning [14]. Active learning has been applied in physical settings to produce Bose-Einstein condensates [15], and for materials design [16,17]. Active learning is particularly useful when each simulation is computationally expensive.

Here we consider such a problem in the context of using interacting Rydberg molecules for simulating open quantum system dynamics. Rydberg systems have strong interactions, and can be optically addressed and positioned [18–20]. These properties can be exploited to simulate quantum systems [21–27]. In addition, Rydberg atoms exhibit state-changing interactions similar to molecular interactions [28,29]. Since the parameters of Rydberg atomic systems are considerably simpler to control than the molecular counterparts, these systems are a promising setting for simulating molecular dynamics such as excitation transport in light-harvesting complexes. Features necessary for simulating molecular systems have been demonstrated using Rydberg atoms, including tunable excitation transport [30–32], non-Markovian behaviour induced in the system [33], as well as controllable thermalization [34]. We will use the same setup as in [34] (related to the setups in [31,33]), as shown in figure 1, with a laser-driven atomic environment. Our setup has thus been used to demonstrate thermalization and flexible system dynamics, with remarkable control provided by the laser parameters. In this work, we denote the tunable laser parameters (Rabi frequencies and detunings) our *laser control parameters*, and *parameter space* is the space of possible laser control parameters.



**Figure 1.** Setup: linear arrangements of system and background atoms, with lasers addressing the background atoms. The atomic energy levels of system and background atoms are shown. The lasers drive the atomic transitions in the background atoms, and the Rabi frequencies  $\Omega_p$ ,  $\Omega_c$  and detunings  $\Delta_p$ ,  $\Delta_c$  are the control parameters for our setup.

There are key remaining questions about thermalization and thermalization dynamics to show that quantum simulation of molecular systems is possible. The first questions are about locating effective laser control parameters: which sets of parameters give rise to thermalization for a given setup? How robust are these parameters to variation? It is also important to understand the relationship between parameters and thermalization temperature: for parameters that generate thermal states, is there a smooth relationship between the parameters and thermal-state temperature? Additionally, quantum simulation requires a system with sufficient size to mimic the target system. This typically requires more than a couple of system atoms, so we also have questions about scalability: do the parameters that produce thermalization change as we scale the number of atoms in the system? If so, how? Finally, approaching a thermal state in the 'right' way - the same way as a target molecular system - is just as important as the thermalization itself. This leads to important questions about the thermalization dynamics: are there different sets of parameters that generate thermalization, with different thermalization dynamics? By varying parameters, what control do we have over the timescale of thermalization relative to interaction timescales within the system?

Given the significance of these open questions in our setup, our outcome of interest in this work is the thermalization of the Rydberg system atoms. Before we can approach the open questions, however, we require a method for exploring the cost landscape. Here the cost is a measure of the distance between the actual state and the target thermal state, and the cost landscape is over the laser control parameters.

In this paper, we will focus on the primary problem of exploring the cost landscape for a multidimensional parameter space, and costly simulations of our setup. This investigation will provide insights into some of the various questions we have posed to facilitate quantum simulation of open systems, and provide a methodology for

approaching any of these questions.

The structure of the manuscript is as follows: first we introduce our setup, methodology and performance measures in section 2. We then present results from scanning the parameter space for a dimer (two atom) aggregate system in section 3. GPs are applied to predict the cost landscape for a dimer system in section 4. Here the predicted landscape is presented and analysed using our performance measures. Then, in section 5, the cost landscape is predicted using GPs for a quadromer system, and we discuss the physical utility and the performance of the prediction. Finally, we present our conclusions in section 6.

## 2. Setup, basic definitions and methods

### 2.1. The system and its tunable parameters

We consider the setup shown in figure 1. This model has been discussed in detail in Refs [31, 34].

The system is a Rydberg  $k$ -mer of  $k$  atoms interacting via resonant-dipole-dipole interactions. These atoms remain in one of two Rydberg states: the lower-energy  $|s\rangle$  state and the excited  $|p\rangle$  state. The Hamiltonian for the system is given by:

$$\mathcal{H}_{\text{sys}} = \sum_{n \neq m} W_{nm} |\pi_n\rangle \langle \pi_m|, \quad (1)$$

where  $|\pi_n\rangle = |s \cdots s p s \cdots s\rangle$  represents the system when a single  $|p\rangle$  excitation is localized at atom  $n$ . A single excitation is shared between the system atoms in our setup. The resonant dipole-dipole interaction  $W_{nm} = C_3/(\mathbf{R}_n - \mathbf{R}_m)^3$ , where  $\mathbf{R}_n$  is the position of atom  $n$  and  $C_3$  is a state-dependent coefficient. The system atoms are arranged linearly, with separation distance  $d$  between atoms.

The environment is composed of laser-driven atoms. These atoms are a distance  $\delta$  from the system atoms, such that the vectors along  $d$  (between system atoms) and  $\delta$  (between a system atom and environment atom) are perpendicular. The environment atoms are treated as 'three-level' atoms: they have a ground state  $|g\rangle$ , a short-lived excited state  $|e\rangle$  and a Rydberg state  $|r\rangle \neq |p\rangle, |s\rangle$ . The excited state is coupled to the zero-temperature photonic continuum, inducing spontaneous emission with decay rate  $\Gamma_p$  to the ground state  $|g\rangle$ . Both the  $|g\rangle \leftrightarrow |e\rangle$  and  $|e\rangle \leftrightarrow |r\rangle$  transition are optically driven, with Rabi frequencies  $\Omega_p$  and  $\Omega_c$  and detunings  $\Delta_p$  and  $\Delta_c$  respectively. The Hamiltonian for the environment atoms, in the rotating wave approximation, is given by:

$$\begin{aligned} \mathcal{H}_{\text{env}} = \sum_{\alpha} \left[ \frac{\Omega_p}{2} [|e\rangle \langle g|]_{\alpha} + \frac{\Omega_c}{2} [|r\rangle \langle e|]_{\alpha} + \text{H.c.} \right] - \Delta_p [|e\rangle \langle e|]_{\alpha} \\ - (\Delta_p + \Delta_c) [|r\rangle \langle r|]_{\alpha} + \sum_{\alpha < \beta} V_{\alpha\beta}^{(rr)} [|r\rangle \langle r|]_{\alpha} [|r\rangle \langle r|]_{\beta}. \end{aligned} \quad (2)$$

The final term in the environment Hamiltonian corresponds to the inter-atomic van der Waals interaction  $V_{\alpha\beta}^{(rr)}$  between atoms in Rydberg states  $|r\rangle$ , where  $\alpha$  and  $\beta$  label the

environment atoms.

The system and environment atoms interact via the Rydberg states of the environment atoms,  $|r\rangle$ . These interactions depend on the state of the system atoms:

$$\mathcal{H}_{\text{int}} = \sum_{n,\alpha} \bar{V}_{n\alpha} |\pi_n\rangle \langle \pi_n| [|r\rangle \langle r|]_{\alpha}, \quad (3)$$

where  $\bar{V}_{n\alpha} = V_{n\alpha}^{(pr)} + \sum_{m \neq n} V_{m\alpha}^{(sr)}$  is the strength of the interaction between the system in the state  $|\pi_n\rangle$  and a specific environment atom  $\alpha$  in the Rydberg state  $|r\rangle$ . The interaction between system atom  $n$  in state  $|p\rangle$  ( $|s\rangle$ ) and environment atom  $\alpha$  in state  $|r\rangle$  is given by  $V_{n\alpha}^{(pr)}$  ( $V_{n\alpha}^{(sr)}$ ). These pairwise interactions depend strongly on distance, such that the strongest interactions are between adjacent environment and system atoms,  $\alpha = n$ .

For the given setup, we can obtain the system dynamics in the following manner. We first define an initial state for our setup that is experimentally accessible and corresponds to a localized excitation:  $\rho(0) = (|\pi_1\rangle \otimes |g\dots g\rangle) (\langle \pi_1| \otimes \langle g\dots g|)$ . The state  $\rho$  is then propagated in time according to the master equation:

$$\frac{d}{dt}\rho(t) = -i[\mathcal{H}, \rho(t)] + \sum_{\alpha} \mathcal{D}[L_{\alpha}]\rho(t), \quad (4)$$

where  $\mathcal{H} = \mathcal{H}_{\text{sys}} + \mathcal{H}_{\text{env}} + \mathcal{H}_{\text{int}}$  and  $\mathcal{D}[L_{\alpha}]\rho = L_{\alpha}\rho L_{\alpha}^{\dagger} - \frac{1}{2}(L_{\alpha}^{\dagger}L_{\alpha}\rho + \rho L_{\alpha}^{\dagger}L_{\alpha})$  terms describe the effect of spontaneous emission on the setup. Here  $L_{\alpha} = \sqrt{\Gamma_p} [|g\rangle \langle e|]_{\alpha}$ .

We specify the inter-atomic distances, atoms and states in the setup as in [34]. The aggregate spacing is  $d = 5 \mu\text{m}$  and the aggregate-environment atom spacing is  $\delta = 2 \mu\text{m}$ . We choose the states of the aggregate atoms to be  $|p\rangle = |43p\rangle$  and  $|s\rangle = |43s\rangle$  of  $^{87}\text{Rb}$ . A dimer (two system atoms) then has a lifetime of approximately  $56 \mu\text{s}$  [35], which is much longer than the timescale of dynamics that we will consider. For the environment atoms we choose the states  $|g\rangle = |5s\rangle$ ,  $|e\rangle = |5p\rangle$  and  $|r\rangle = |38s\rangle$  of  $^{87}\text{Rb}$ . The decay rate from  $|e\rangle$  is  $\Gamma_p = 6.1 \text{ MHz}$ .

In this setup, we want to obtain thermalization of the system. This provides a resource for quantum simulation of systems in real (thermal) environments. We are thus interested in preparing a thermal state:

$$\rho_{T_{\text{eff}}}^{\text{th}} = \frac{1}{Z} \sum_n e^{-E_n/(kT_{\text{eff}})} |\varphi_n\rangle \langle \varphi_n|, \quad (5)$$

where  $T_{\text{eff}}$  is the effective temperature,  $k$  is the Boltzmann constant, and  $Z = \text{Tr}\{e^{-\mathcal{H}_{\text{sys}}/(kT_{\text{eff}})}\}$ . The eigenstates and eigenenergies of  $\mathcal{H}_{\text{sys}}$  (equation 1) are denoted by  $|\varphi_n\rangle$  and  $E_n$ , respectively.

Note that the temperature scale  $kT_{\text{eff}}$  is not given by an ambient temperature of a thermal bath. The ambient temperature is typically on the order of  $\mu\text{K}$ , and we also have an additional atomic component of our environment. Here the temperature scale is given in terms of the interaction strength  $W$ , which determines the eigenenergies  $E_n$ .

We set a time  $t_f$  for which thermalization should have happened to a given thermal state. As addressed in the introduction, control over the thermalization timescale and

the temperature of the target thermal state are important aspects for simulating general open systems. In this work, we focus on cost landscape prediction and analysing the results. We thus fix the thermalization timescale to be faster than a given  $t_f$ , where the target thermal state has a given temperature  $kT_{\text{eff}}$ . However, our approach can be extended by varying the values  $t_f$  and  $kT_{\text{eff}}$ . We will comment on the choices of  $t_f$  and  $kT_{\text{eff}}$  in section 3.

In Ref. [34], it was shown that for the given setup, thermalization to a tunable temperature can be achieved for particular choices of the laser driving parameters ( $\Omega_p$ ,  $\Omega_c$ ,  $\Delta_p$  and  $\Delta_c$ ). Here, we are similarly interested in investigating thermalization of the system by controlling the laser driving parameters for the background atoms. However, unlike in Ref. [34], in this paper we are interested in knowing the cost landscape generally (how well thermalization can be performed over the full parameter space). This in turn equips us to answer other physical questions about the setup (e.g. robustness of thermalization to parameter variations or scaling the system size), as described in the introduction.

## 2.2. Quantifying the outcome

We want to quantify how well we prepare the target thermal state. To do this, we apply the trace distance  $D(\rho_S(t_f), \rho_{T_{\text{eff}}}^{\text{th}})$  to measure the distinguishability of a given state of the system  $\rho_S(t_f)$  from the target thermal state  $\rho_{T_{\text{eff}}}^{\text{th}}$ :

$$D(\rho_S(t_f), \rho_{T_{\text{eff}}}^{\text{th}}) = \frac{1}{2} \text{Tr} \{ |\rho_S(t_f) - \rho_{T_{\text{eff}}}^{\text{th}}| \}, \quad (6)$$

with  $|\rho| = \sqrt{\rho^\dagger \rho}$ . The state  $\rho_S(t_f)$  is obtained by time-propagation of our setup  $\rho(t)$  (equation (4)), followed by tracing out the environment atoms. We will use  $D(\rho_S(t_f), \rho_{T_{\text{eff}}}^{\text{th}})$  (denoted by  $D$  for convenience) as our cost function.

Note that the cost function is based on a state comparison at a single point in time,  $t_f$ . We assume that the cost quantifies how well states have thermalized to the target state. However, the cost does not distinguish whether the propagated state is still changing in time; such a state could dynamically pass 'close' to the target state without being effectively thermalized. In section 3 of the SI, we show that states that are not yet steady have a minimal effect in our case, validating our association of the cost with effective thermalization. We also provide an alternate cost function in the SI that could be used in cases where this transience issue arises.

## 2.3. Gaussian processes

In this manuscript, we numerically investigate the cost landscape. The challenge is that simulations of our setup are computationally costly. Also, an analytic model for the cost landscape has not been found for our setup. The ideal case - knowing the full cost landscape - is thus impractical even for a Rydberg dimer system (just two system atoms).

Using machine learning, we can gain insight into the full cost landscape through Gaussian process (GP) regression and prediction. GP regression fits known points in the landscape (parameters with known associated costs) and can then predict the full cost landscape. This prediction of the cost landscape attributes a Gaussian predictive distribution for the cost at any given 'test' parameter set. From this distribution, the predicted mean cost and standard deviation can be extracted. A functional form is assumed for the covariance of predicted costs over parameter space. The covariance can depend on length-scales for characteristic variation in the cost as a function of each parameter, and these length-scales can also be estimated to provide the best fit of the cost landscape by the GP regression.

The predicted cost landscape can be used within a numerical routine to balance optimization (landscape exploitation) with landscape exploration. We are most interested in sets of parameters providing a low cost (close to 0). However, we are not just interested in the minimal cost and its associated parameters. We want to know every low-cost parameter region, along with its extent in parameter space. This information requires an exploration of the cost landscape, with higher 'priority' of exploration given to regions that may have low cost values. As explained in the SI (section 4), the predicted cost landscape can be used to identify likely low-cost regions of the cost landscape. Similarly, the standard deviation for the predicted cost landscape can identify regions of parameter space that should be explored due to a lack of knowledge (uncertainty in the predicted costs, such that the costs could be low). In this way, we obtain a numerical routine that uses previous *runs* (simulations and GP regression) to guide the parameters for subsequent simulations. We use the optimization package MLOOP [15,36], based on the Gaussian process regression algorithm (2.1) from [4] and implemented in scikitlearn [37].

*2.3.1. Performance measures* We would like quantitative measures of how well the GP-based numerical routine performs. We will consider measures that compare GP-predicted regions of the cost landscape with exact calculations of these same cost-landscape regions by scanning parameter space (extrinsic measures). These scans are only possible for limited regions of the cost landscape, and for small system sizes. Thus, we will also consider measures that only depend on the (region of the) cost landscape predicted by the numerical routine (intrinsic measures). In both cases, we evaluate 2D cross-sections of the cost landscape.

*Extrinsic measures* The extrinsic measures allow us to analyse the accuracy of the predicted landscape: how well it matches the exact landscape. We introduce three measures to quantify this: (a) *accuracy*, (b) *accuracy for  $D < C_i$* , and (c) *penalty*.

- (a) The *accuracy* measure is the absolute difference between the scanned cross-sections and the predicted cross-sections. The absolute difference is taken point-wise, then averaged over every point in the cross-sections. Note that the costs obtained by both parameter scans and predictions are discrete samples from the cost landscape.

- (b) The *accuracy for  $D < C_l$*  follows the same procedure as (a), but only points where the scanned cross-sections have trace distance  $D$  below some low-cost threshold  $C_l$  are included in the average. This measure thus quantifies accuracy for the low-cost regions of the cost landscape, with the choice of  $C_l$  discussed in section 4.1.
- (c) The *penalty* is a measure for how closely the real cost landscape fits within the standard deviation for the predicted cost landscape. The contribution to the *penalty* for a given cost-landscape point is the absolute distance between the real cost and the standard-deviation interval about the predicted cost (this distance is 0 when the real point is within the interval). To calculate the *penalty*, we take the sum of these point-wise contributions over every point in each cross-section, then average over the cross-sections.

It is important to note that we would like to apply the routine in a regime where such extrinsic measures are not possible. The main benefit of using Gaussian processes to predict the landscape is that this prediction can occur where scans are not feasible: in such cases, we will need to rely on intrinsic measures.

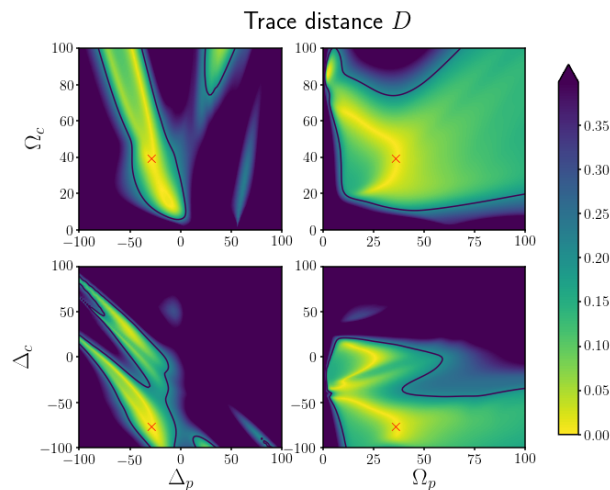
*Intrinsic measures* We introduce four intrinsic measures: (d) *precision*, (e) *precision for  $D < C_l$* , (f) *absolute convergence* and (g) *best cost*.

- (d) The *precision* is the standard deviation averaged over every point in the cross-sections.
- (e) The *precision for  $D < C_l$*  follows the same procedure as (d), but only points where the predicted cost cross-sections have trace distance  $D < C_l$  are included in the average.
- (f) The *absolute convergence* is the absolute difference between the current and previous predicted cost landscapes (averaged over points). This measure is a function of the number of *runs* performed by the numerical routine. A single *run* of the numerical routine involves selection of parameters to test, and one simulation (see section 4 of the SI for details). We calculate the predicted cost landscape (cross-sections) to evaluate the performance measures every 20 runs, for numerical convenience. The 'previous' predicted cost landscape is thus calculated using 20 fewer runs than the 'current' predicted cost landscape.
- (g) The *best cost* is the lowest cost that has been obtained: it is the optimal cost from the set of known (evaluated) points of the cost landscape.

### 3. Scanning parameter space for a dimer

The naive approach to obtain the cost landscape is to simply scan the variable parameters. Recall that we have four such parameters:  $(\Omega_p, \Omega_c, \Delta_p, \Delta_c)$ , which means an enormous number of calculations.





**Figure 2.** Parameter scan for a Rydberg dimer. The cost  $D$  is evaluated after  $2 \mu\text{s}$ . The cross marks the center of the 2D cross-sections:  $(\Omega_p, \Omega_c, \Delta_p, \Delta_c) = (37.1, 40.1, -26.6, -74.7)$ , which has  $D = 6 \times 10^{-5}$ . The solid black lines mark the contour defined by  $D = 0.2$ , for comparison with the cost landscapes to follow.

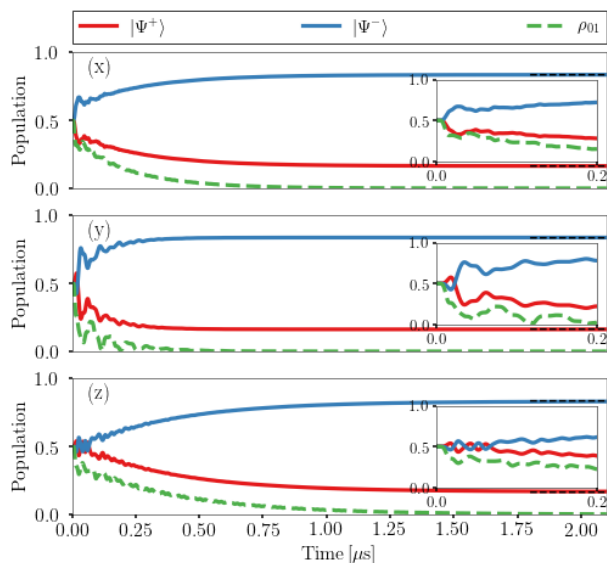
As a reference for the Gaussian process approach of the following sections we perform such a scan for the simplest nontrivial setup where a scan is tractable: a Rydberg dimer.

We choose the target thermal state temperature to be given by  $kT_{\text{eff}} = 1.2 W$ . This temperature has no special significance for the thermal state, except that it is not a limiting case of nearly zero or infinite temperature (in these cases all population is in the lowest eigenstate, and the eigenstates are equally populated, respectively). The methods we present can be applied for any temperature.

In addition, we require that this thermal state is reached before a certain time, which we choose here to be  $t_f = 2 \mu\text{s}$ . Again, as for the choice of the temperature this time is chosen arbitrarily (but is much smaller than the lifetime of the Rydberg states of the system ( $\sim 56 \mu\text{s}$  [35])). As discussed in section 2.1, by choosing a time  $t_f$  we set the maximal timescale of thermalization. We propagate the state for  $2 \mu\text{s}$  and compare the resulting state with the reference thermal state as described in section 2.2.

Even though we look at just two system atoms, a full scan of the cost landscape is not feasible. Therefore in figure 2 we present exemplary 2D cross-sections of the 4D parameter space. We choose a low-cost central point for these cross-sections:  $(\Omega_p, \Omega_c, \Delta_p, \Delta_c) = (37.1, 40.1, -26.6, -74.7) \equiv x$ , which has cost  $D = 6 \times 10^{-5}$ . The ranges of the laser parameters are chosen to be experimental achievable [38, 39], with detunings small enough to avoid unwanted resonances.

Each of the 2D cross-sections in figure 2 are composed of  $100 \times 100$  points (i.e. we evaluated 10,000 sets of parameters by time-propagating the state). This number of points allows us to resolve features in parameter space up to one or two MHz. The computational time for each point in a given 2D cross-section was  $\sim 1$  s (on a single core). Many such 2D cross-sections are required to obtain information about the full



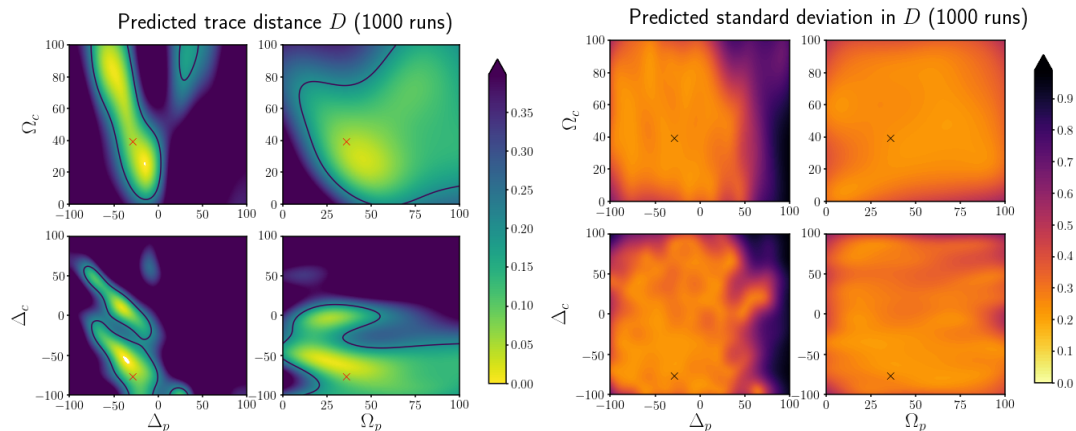
**Figure 3.** Population dynamics for different parameters from the 2D cost landscape cross-sections centered at  $x$  (as in figure 2).  $(x)$  corresponds to the center of the cross-sections, and  $(x)$ ,  $(y)$  and  $(z)$  are marked in the six cross-sections from varying each pair of parameters shown in figure 1 in the SI. Each point satisfies  $D < 0.01$ . The eigenstates and the off-diagonal element  $\rho_{01}$  are plotted over time, and black dashed lines show the target eigenstate populations. The target value of  $\rho_{01}$  is zero. The inset is zoomed in on the initial dynamics of the corresponding subplot, for clarity.

4D parameter space.

In the introduction, we raised various questions about simulation of open quantum system dynamics with Rydberg atoms, and about thermalization in particular. For the dimer system, the cross-sections in figure 2 begin to answer some of these questions. We can observe a range of parameters that give rise to low-cost thermalization (though still within a small subspace of the full parameter space). We can observe the robustness of these parameters in the planes of the 2D cross-sections. From the low-cost parameter regions, we can also sample particular sets of parameters to observe the thermalization dynamics. As shown in figure 3, we found that different thermalization dynamics can be obtained from different sets of parameters. However, note that due to the computational cost, this (scanning) approach is not scalable to larger systems, and nor can it be extended generally to explore the larger 4D parameter space even for the Rydberg dimer.

#### 4. Predicting the cost landscape using Gaussian processes for a dimer

We have seen that the computational cost is a fundamental challenge for numerical investigation of thermalization in our setup (and also more generally for simulation of scalable quantum systems). We thus want to obtain as much information as possible from as few simulations as possible. The information that we desire is a balance of optimization and landscape exploration: we are interested in the (multiple) regions of



**Figure 4.** Predicted  $D$  landscape (left) and its standard deviation (right) for thermal state preparation of a Rydberg dimer. The cross marks the center of the cross-sections, which is the same as in Figure 2. The solid black lines mark the contour defined by  $D = 0.2$ . The landscape is predicted for a  $2 \mu\text{s}$  evolution time, using 1000 runs of the Gaussian process routine. All four parameters were allowed to vary: for  $(\Omega_p, \Omega_c, \Delta_p, \Delta_c)/(2\pi)$  (MHz): the minimum boundary, maximum boundary and optimal parameters were  $(0.1, 0.1, -100., -100.)$ ,  $(100., 100., 100., 100.)$  and  $(18.9, 92.4, 9.3, 42.5)$  respectively.

parameter space that give rise to low-cost thermal states, including the breadth of these regions.

#### 4.1. Gaussian process prediction

We use GPs, outlined in section 2.3, to predict the cost landscape for the Rydberg dimer setup. We apply a numerical routine that uses a GP to both explore the cost landscape, and to exploit the landscape by focusing on low-cost regions as explained in section 4 in the SI.

In figure 4 we show the predicted landscape and its associated standard deviation after 1000 runs of our numerical routine (i.e. 1000 sets of parameters were simulated).

To compare directly with the numerical scans of the parameter space, we display the 2D cross-sections centered at the same point  $x$  as the scans in figure 2. This comparison shows that qualitatively, the landscape prediction is very good, i.e. it identifies almost all low-cost regions. This is remarkable since just 1000 simulations have been performed. The four laser control parameters are allowed to vary freely within the ranges shown in the cross-sections; the parameters can be sampled from the full 4D parameter space and are not restricted to lie within the displayed cross-sections. This is in contrast to the  $100 \times 100$  points for *each* cross-section displayed for the parameter scans, where each of the 10,000 points lies within the displayed cross-sections.

We have noted that the parameters can vary within the full 4D space; the 2D cross-sections displayed in figure 4 are not 'preferred' in any sense by the numerical routine. For instance, the optimal set of parameters  $(\Omega_p, \Omega_c, \Delta_p, \Delta_c)/(2\pi)$  (MHz)

$= (18.9, 92.4, 9.3, 42.5)$ , with  $D < 0.001$ , does not lie within these cross-sections. Since these arbitrary 2D cross-sections of the predicted 4D landscape are very good, we expect that the full 4D space is predicted similarly well.

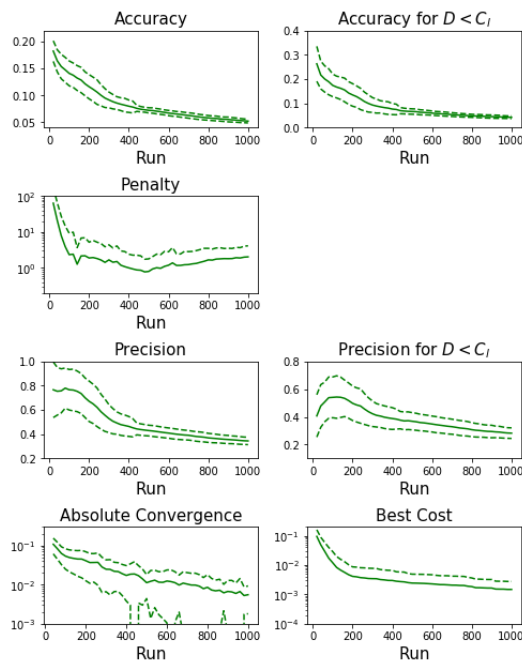
In figure 4, we show the predicted landscape after 1000 runs. This number of runs is chosen based on the following considerations. Firstly, in our implementation, prediction is more expensive for a GP with more completed runs. A possible way to circumvent this issue is described in [4]. Secondly, the standard deviation of the predicted cost landscape typically decreases with the number of runs, see e.g. figure 5. There is thus a trade-off between the computational resources required for a GP-based numerical routine, and the precision of the predicted cost landscape. For the dimer setup, 1000 runs is well within the 'sweet spot': firstly, it is computationally much cheaper to use Gaussian processes than to scan parameters to investigate the cost landscape. Secondly, the predicted landscape is very close to the actual landscape after 1000 runs: the typical *accuracy*  $< 0.1$ , as shown in figure 5. Similarly, considering an intrinsic measure, the standard deviation in the predicted cost is sufficiently small to identify likely low-cost regions of the cost landscape. As seen in figure 4, after 1000 runs, the standard deviation for the predicted cross-sections typically takes a value between 0.2 and 0.3 for the low-cost regions. Note that the 'sweet spot', where there is both accurate landscape prediction and a computational resource reduction from scans, grows as each simulation becomes computationally more expensive (for more details, see section 4 in the SI).

We now specify our choice of threshold  $C_l$  for low-cost regions of the landscape. We have noted that the standard deviation for the predicted cross-section is  $\sim 0.2 - 0.3$  for regions with cost  $D$  below 0.4, as seen in figure 4. It is thus prudent to choose a value for  $C_l$  close to this standard deviation value: then we include regions with predicted low cost, where the standard deviation for the prediction is around the size of the distance from zero cost (perfect thermalization). This way, the intrinsic likelihood (from the predicted cost) of a low cost guides our low-cost threshold. By focusing on regions of the cost landscape with a predicted cost below 0.2, we can rule out vast regions of the cost landscape after 1000 runs. This is demonstrated by the contours in the predicted-cost cross-sections of figure 4 (left), and we expect arbitrary cross-sections to have smaller low-cost regions (the cross-sections in figure 4 are centered on a particularly low-cost point). We thus set the 'low-cost' threshold  $C_l = 0.2$ .

#### 4.2. Quantifying prediction performance

To provide a quantitative analysis of the GP performance, we apply the extrinsic and intrinsic performance measures defined in section 2.3.1. Each of these measures (aside from the *best cost*) is averaged over the six 2D cross-sections produced by varying the four laser parameters (pairwise) about the central point.

Note that the numerical routine is stochastic, so the resulting landscape prediction, as in figure 4, varies for each instance of the numerical routine. To quantify how well the GP-based numerical routine predicts the cost landscape, we have performed 100



**Figure 5.** Performance measures for the six 2D cross-sections centered as in figures 2 and 4. These measures are defined in section 2.3.1. The upper three measures displayed are extrinsic, comparing the predicted cost landscape cross-sections with the scans. The lower four measures are intrinsic, using just the predicted cost landscape cross-sections and their standard deviations. The measures are calculated using 100 instances of the (stochastic) numerical routine, each of which were used to generate six 2D cross-sections of the predicted cost landscape and the associated uncertainty. We calculated the performance measures for each instance, and display the mean and the standard deviation of the performance measures from the resulting measure distributions. Note that we do not show a standard deviation below the mean in plots where this is often less than zero, for clarity.

instances of the numerical routine. In figure 5, the mean over the instances is plotted for each performance measure, along with the standard deviation.

One would expect that with more runs, i.e. more samples of the cost landscape, the landscape prediction becomes more accurate, more precisely known and approaches convergence (the landscape predictions become more similar as the number of samples increases). In figure 5, this expectation is validated: the measures are almost always improving with the number of runs. The *accuracy* (for  $D < C_l$ ) decreases with the number of runs to a final value  $\sim 0.05$  ( $\sim 0.04$ ), which means that this is the average distance between the predicted and exact landscapes for all points (points with  $D < 0.2$ ) after 1000 runs. The *precision* and *precision for  $D < C_l$*  measures in figure 5 demonstrate poor initial predictions of the landscape, where the standard deviation is under-estimated prior to around 100 runs, most noticeably for the  $D < C_l$  regions. Then, after around 100 runs, the *precision* measures decrease with the number of runs (the precision improves for the predicted landscape). As expected, the *absolute convergence* also demonstrates increasing consistency in the predicted landscape as the number of

runs increases.

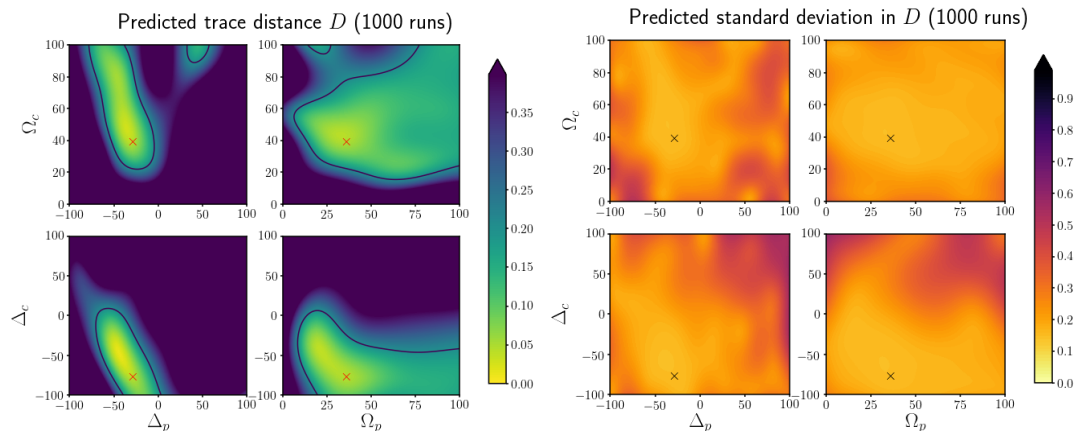
The improvement in the *precision* measures, along with the improvement in *accuracy*, reflects successful landscape exploration by our numerical routine. The *best cost* also improves with the number of runs: this demonstrates the optimization aspect of our numerical routine.

Recall that the *penalty* grows when actual cost values lie outside the predicted standard deviation in the predicted cost. In figure 5, the *penalty* decreases as the prediction improves, then increases fractionally with the number of runs. The main change in this measure is the decrease in the *penalty* due to the improving landscape prediction (and more accurate standard deviation). Initially the prediction and its standard deviation are poor due to very little information from few previous samples of the cost landscape. The *penalty* measure improves significantly within 100 runs. The subsequent slight increase in the *penalty* could be due to narrow features in the landscape that have not been sampled. The *penalty* for these features would become worse as the standard deviation in these incorrect regions of the predicted landscape is reduced with the number of runs.

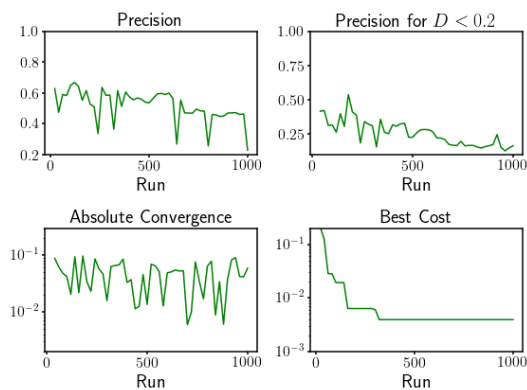
If we did not have access to the scanned landscape, we would only have the intrinsic measures. It is thus encouraging to note that the *precision* measure (which averages the standard deviation in the cost landscape) is an upper bound for the *accuracy* of the landscape. This means that knowing the *precision* allows us to conservatively estimate the landscape *accuracy*, at least for the Rydberg dimer setup. The standard deviation predictions are dependent on the choice of the covariance function for the Gaussian process (as well as the optimized fitting parameters within the covariance function). The covariance function describes how the predicted landscape can change away from the known points in the landscape. We have seen that the squared-exponential covariance function [40] employed by our routine (see section 4 of the SI) provides conservative standard deviation predictions. We thus expect the same behaviour when we consider larger systems using the same covariance function. Then the *precision* could be used as a conservative estimate of the *accuracy* for general system sizes.

From another intrinsic measure, the *absolute convergence* measure, it appears that we could set a certain convergence threshold as a criteria to stop the numerical routine. That is, when the predicted landscapes change less than a given amount between subsequent predictions, we might expect that the routine has converged 'close' (as determined by the threshold) to the correct cost landscape. This approach is limited, however, when we consider individual instances of the numerical routine. Although in figure 5 the *absolute convergence* measure (almost) monotonically decreases with the number of runs for 100 routine instances, figure 7 for the quadromer demonstrates that a single instance involves much more fluctuation in this measure. This is due to the stochastic nature of both the sampling and the GP regression to fit the known points in the cost landscape.

The results that we have presented are for a particular choice of numerical routine, which we have found to work well for our setup. This choice is explained in detail in



**Figure 6.** 2D slices of the quadromer (left) predicted cost landscape and (right) its standard deviation, after 1000 runs of the numerical routine. The crosses mark the center of each cross-section. Contours are at  $D = 0.2$  in the cost landscape cross-sections.



**Figure 7.** Intrinsic performance measures calculated using six 2D cross-sections of the cost landscape, centered at  $x$ . The six cross-sections are shown in figure 3 of the SI.

section 5 of the SI, along with alternative routines.

## 5. Predicting the cost landscape using Gaussian processes for a quadromer

We have seen that scanning parameter space does not provide an approach that can be scaled to larger system sizes within a reasonable computational time. However, the prediction procedure using Gaussian processes performed remarkably well within a much shorter time. We now apply this approach to gain insights from the cost landscape of a Rydberg quadromer system.

We use our numerical routine to predict 2D cross-sections of the cost landscape centered at the same point  $x$  as for figures 2 and 4, for comparison. These cross-sections are presented in figure 6. Here, as for the predicted cost landscape for the dimer in figure 4, 1000 points are sampled from the full 4D space to predict the landscape.

In figure 6, low-cost regions are predicted within the given cross-sections. The predicted landscape gives us a sense of the robustness of these regions. The low-cost regions are similar to those in figure 4 for the dimer. This similarity suggests that the dimer and quadromer setups possess related parameter dependence with respect to thermal state preparation. This parameter-dependence could be extrapolated (with more data) to larger systems. It was found in [34] that particular laser control parameters were associated with low-cost thermal state preparation in different-sized systems. This finding was for particular points in the cost landscape of different-sized systems. In this paper, access to the (predicted) cost landscape over large regions of parameter space allows us to explore the relationship between parameters and system size in much more detail. For example, for the cross-sections displayed in figure 6, the minimal cost regions have higher cost than the dimer predictions. We expect this increase to continue with system size, and one could investigate whether the lower cost 'peaks' for each system size can be smoothly followed through the landscape as a function of the laser parameters.

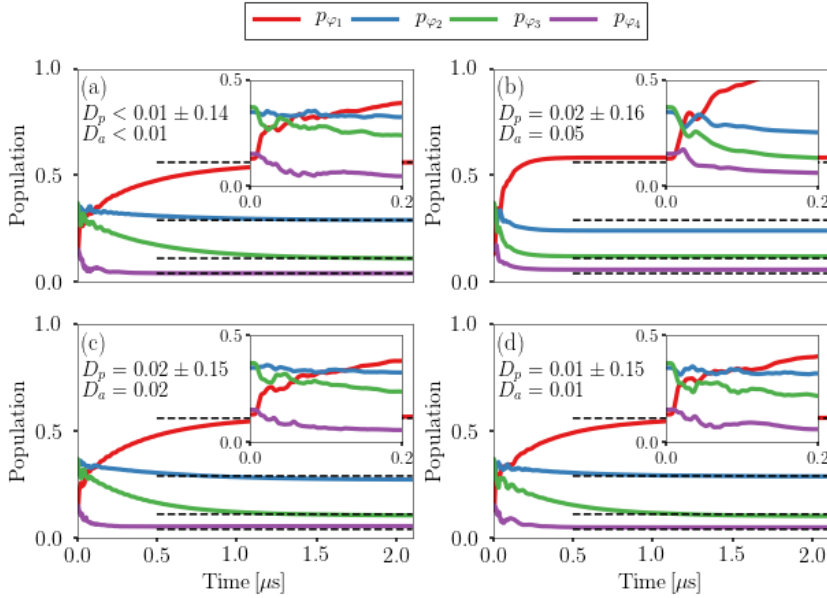
Since scanning the cost landscape is too computationally expensive, the extrinsic performance measures were not calculated for the quadromer. This is an example of a setup where simulations are costly (a single run takes  $\sim 200$  s on two cores), however a single instance of our numerical routine can be used to predict the cost landscape (as in figure 6).

The intrinsic performance measures were calculated using the six 2D cross-sections of the cost landscape produced by fixing four laser parameters and varying two at a time about the central point (four of the cross-sections are shown in figure 6, all six are shown in section 1 of the SI). The results are shown (as a function of the number of runs) in figure 7. Since the performance measures here are calculated using a single instance of the numerical routine, rather than 100 instances as in figure 5, we observe much more fluctuation. Nonetheless, it can be observed from figure 7 that there is an improving trend for each measure, which is similar to the trend in figure 5 for the dimer setup.

In figure 7, the *precision* drops to  $\sim 0.23$  after 1000 runs. Thus, we again set the 'low-cost' threshold  $C_l = 0.2$ . The final value of the *precision for  $D < C_l$* , which is typically lower than the *precision*, is  $\sim 0.16$ . As for the dimer, we expect that the *precision* measures provide a conservative upper bound on the accuracy of the predicted landscape (see the discussion in section 4.2). Thus, we expect the predicted cost landscape for the quadromer to have *accuracy*  $< 0.23$  overall, and we expect the *accuracy for  $D < C_l$*  measure to be less than 0.16.

The *best cost* in figure 7 reaches a value of  $\sim 4 \times 10^{-3}$  after 300 runs and remains at this value. This is a little higher than the mean *best cost* over 100 instances for the dimer case ( $\sim 1.5 \times 10^{-3}$ ). Nonetheless, this *best cost* value demonstrates that very-low-cost ( $D \sim 0.01$ ) regions exist for the quadromer setup. These can be seen in the predicted cost landscape and standard deviation centered on the *best cost*, which are shown in the SI (figure 4).





**Figure 8.** Quadromer dynamics plots. The parameters are chosen from the predicted cost landscape cross-sections, as marked in the 2D cross-sections (figure 3 and figure 4) in the SI. Each set of parameters was chosen from predicted low-cost regions. Here the population in the  $j$ th lowest energy eigenstate  $|\varphi_j\rangle$  is denoted by  $p_{\varphi_j}$ . We only show the eigenstate population dynamics for clarity, and the target thermal state populations are also shown (dashed lines). The cost compares the full state with the target thermal state; the predicted cost  $D_p$  (with standard deviation) and the actual cost  $D_a$  are shown for each subplot. The inset is a zoomed-in display of the first  $0.2 \mu\text{s}$  from the respective subplot.

Achieving a target *precision for*  $D < C_l$  could be used as a stopping criterion for the numerical routine (such that no more runs are performed), as it is a useful low-cost identification measure and has a relatively smooth dependence on the number of runs. In contrast, the *absolute convergence* fluctuates dramatically, which would make this measure difficult to apply as a stopping criterion.

As we did for the dimer, we can now use the predicted cost landscape to provide a preliminary investigation of how or whether different parameters give rise to different thermalization dynamics. In figure 6, as well as in figure 4 in the SI (which is centered at the *best cost*), different low-cost regions are identified in the predicted cost landscape. We sampled points from these regions and we show their dynamics in figure 8. The four eigenstate populations undergo different dynamics, which is shown with an inset zoomed in on the initial dynamics for clarity. The predicted costs for each point are provided with the actual costs in the figure, and the differences in each case are much lower than the standard deviation in the predicted cost (as is also the case in almost every point that we validated from the quadromer landscape). While the populations in the subplots (a), (c) and (d) become steady very close to the desired eigenstate populations, the dynamics in (b) are not as close (with higher cost). Nonetheless, this point in the cost landscape (which also has a faster timescale to reach a steady-state) could be used

to perform local optimization and locate a nearby lower cost with similar associated dynamics.

## 6. Conclusions

We have tackled the general problem of extracting much information from few costly simulations. We demonstrated that Gaussian processes perform admirably at this task in the setting of quantum simulation in atomic physics.

The successful application of GPs also provided physical insight into various questions required for simulating molecular systems. In particular, we identified sets of parameters that give rise to thermalization, as well as the robustness of these parameter regions. Using the parameter space information provided by GPs, we demonstrated that different thermalization dynamics can be observed in our setup. We also obtained preliminary information about how parameters that result in thermalization vary as the system size changes. Our method provides a useful approach for further study of this relationship. Similarly, one could vary the temperature of the target thermal state, and observe the resulting changes in the low-cost regions of parameter space. Our approach thus supports the development of physical insight into the controllability of our setup for molecular simulation.

It is important to note that the cost landscape could be scanned experimentally, since the experiment duration is just  $2 \mu\text{s}$  (independent of the system size), and we are interested in a parameter space described by experimentally tunable laser parameters. Our investigations in this paper confirmed that our setup does provide key elements required for molecular simulation, and added to the quantum simulation toolbox by identifying low-cost parameter regions, robustness, and different thermalization dynamics. This provides a foundation and motivation for experimental exploration of the setup.

## Acknowledgements

The authors thank Jan-Michael Rost for fruitful discussions. We acknowledge funding from the DFG: grant EI 872/4-1 through the Priority Program SPP 1929 (GiRyd).

## References

- [1] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.
- [2] R. Storn and K. Price, “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *J. Global Optim.*, vol. 11, pp. 341–359, 1997.
- [3] G. Dueck and T. Scheuer, “Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing,” *J. Comput. Phys.*, vol. 90, pp. 161–175, 1990.
- [4] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge: the MIT Press, 2006.

- [5] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, “Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons,” *Phys. Rev. Lett.*, vol. 104, no. 136403, 2010.
- [6] A. P. Bartók and G. Csányi, “Gaussian Approximation Potentials: A Brief Tutorial Introduction,” *Int. J. Quantum Chem.*, vol. 115, pp. 1051–1057, 2015.
- [7] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, “Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning,” *Phys. Rev. Lett.*, vol. 108, no. 058301, 2012.
- [8] M. Rupp, R. Ramakrishnan, and O. A. von Lilienfeld, “Machine Learning for Quantum Mechanical Properties of Atoms in Molecules,” *J. Phys. Chem. Lett.*, vol. 6, pp. 3309–3313, 2015.
- [9] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, “Reinforcement Learning in Different Phases of Quantum Control,” *arxiv preprint: 1705.00565v2*, 2017.
- [10] M. August and J. M. Hernández-Lobato, “Taking gradients through experiments: LSTMs and memory proximal policy optimization for black-box quantum control,” *arxiv preprint: 1802.04063v2*, 2018.
- [11] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, “Reinforcement Learning with Neural Networks for Quantum Feedback,” *arxiv preprint: 1802.05267v2*, 2018.
- [12] F. Albarrán-Arriagada, J. C. Retamal, E. Solano, and L. Lamata, “Measurement-based adaptation protocol with quantum reinforcement learning,” *arxiv preprint: 1803.05340*, 2018.
- [13] A. A. Melnikov, H. P. Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, “Active learning machine learns to create new quantum experiments,” *PNAS*, 2018.
- [14] B. Settles, “Active Learning Literature Survey,” Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
- [15] P. B. Wigley, P. J. Everitt, A. van den Hengel, J. W. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. D. Quinlivan, P. Manju, C. C. N. Kuhn, I. R. Petersen, A. N. Luiten, J. J. Hope, N. P. Robins, and M. R. Hush, “Fast machine-learning online optimization of ultra-cold-atom experiments,” *Scientific Reports*, vol. 6, no. 25890, 2016.
- [16] S. Ju, T. Shiga, L. Feng, Z. Hou, K. Tsuda, and J. Shiomi, “Designing nanostructures for phonon transport via Bayesian optimization,” *Phys. Rev. X*, vol. 7, no. 021024, 2017.
- [17] C. Li, D. R. d. C. Leal, S. Rana, S. Gupta, A. Sutti, S. Greenhill, T. Slezak, M. Height, and S. Venkatesh, “Rapid Bayesian optimisation for synthesis of short polymer fiber materials,” *Scientific Reports*, vol. 7, no. 5683, 2017.
- [18] A. Browaeys, D. Barredo, and T. Lahaye, “Experimental investigations of dipole-dipole interactions between a few Rydberg atoms,” *J. Phys. B: At. Mol. Opt. Phys.*, vol. 49, no. 152001, 2016.
- [19] F. Nogrette, H. Labuhn, S. Ravets, D. Barredo, L. Béguin, A. Vernier, T. Lahaye, and A. Browaeys, “Single-atom trapping in holographic 2D arrays of microtraps with arbitrary geometries,” *Phys. Rev. X*, vol. 4, no. 021034, 2014.
- [20] M. Schlosser, S. Tichelmann, J. Kruse, and G. Birkl, “Scalable architecture for quantum information processing with atoms in optical micro-structures,” *Quantum Inf. Process.*, vol. 10, pp. 907–924, 2011.
- [21] S. Wüster and J.-M. Rost, “Rydberg aggregates,” *J. Phys. B: At. Mol. Opt. Phys.*, vol. 51, no. 032001, 2018.
- [22] H. Weimer, M. Müller, I. Lesanovsky, P. Zoller, and H. P. Büchler, “A Rydberg quantum simulator,” *Nat. Phys.*, vol. 6, pp. 382–388, 2010.
- [23] H. Labuhn, D. Barredo, S. Ravets, S. de Léséleuc, T. Macrì, T. Lahaye, and A. Browaeys, “Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models,” *Nature*, vol. 534, p. 667, 2016.
- [24] D. Barredo, H. Labuhn, S. Ravets, T. Lahaye, A. Browaeys, and C. S. Adams, “Coherent excitation transfer in a spin chain of three Rydberg atoms,” *Phys. Rev. Lett.*, vol. 114, no. 113002, 2015.
- [25] J. P. Hague and C. MacCormick, “Quantum simulation of electron-phonon interactions in strongly deformable materials,” *New Journal of Physics*, vol. 14, no. 033019, 2012.

- [26] I. Lesanovsky, “Liquid ground state, gap, and excited states of a strongly correlated spin chain,” *Phys. Rev. Lett.*, vol. 108, no. 105301, 2012.
- [27] O. Mülken, A. Blumen, T. Amthor, C. Giese, M. Reetz-Lamour, and M. Weidemüller, “Survival probabilities in coherent exciton transfer with trapping,” *Phys. Rev. Lett.*, vol. 99, no. 090601, 2007.
- [28] F. Robicheaux, J. V. Hernández, T. Topçu, and L. D. Noordam, “Simulation of coherent interactions between Rydberg atoms,” *Phys. Rev. A*, vol. 70, no. 042703, 2004.
- [29] C. S. E. van Ditzhuijzen, A. F. Koenderink, J. V. Hernández, F. Robicheaux, L. D. Noordam, and H. B. v. L. van den Heuvel, “Spatially resolved observation of dipole-dipole interaction between Rydberg atoms,” *Phys. Rev. Lett.*, vol. 100, no. 243201, 2008.
- [30] G. Günter, H. Schempp, M. Robert-de Saint-Vincent, V. Gavryusev, S. Helmrich, C. S. Hofmann, S. Whitlock, and M. Weidemüller, “Observing the Dynamics of Dipole-Mediated Energy Transport by Interaction-Enhanced Imaging,” *Science*, vol. 342, no. 6161, p. 954, 2013.
- [31] D. W. Schönleber, A. Eisfeld, M. Genkin, S. Whitlock, and S. Wüster, “Quantum Simulation of Energy Transport with Embedded Rydberg Aggregates,” *Phys. Rev. Lett.*, vol. 114, no. 123005, 2015.
- [32] H. Schempp, G. Günter, S. Wüster, M. Weidemüller, and S. Whitlock, “Correlated Exciton Transport in Rydberg-Dressed-Atom Spin Chains,” *Phys. Rev. Lett.*, vol. 115, no. 093002, 2015.
- [33] M. Genkin, D. W. Schönleber, S. Wüster, and A. Eisfeld, “Non-Markovian dynamics in ultracold Rydberg aggregates,” *J. Phys. B: At. Mol. Opt. Phys.*, vol. 49, no. 134001, 2016.
- [34] D. W. Schönleber, C. D. B. Bentley, and A. Eisfeld, “Engineering thermal reservoirs for ultracold dipole-dipole-interacting Rydberg atoms,” *New J. Phys.*, vol. 20, no. 013011, 2018.
- [35] I. I. Beterov, I. I. Ryabtsev, D. B. Tretyakov, and V. M. Entin, “Quasiclassical calculations of blackbody-radiation-induced depopulation rates and effective lifetimes of Rydberg nS, nP, and nD alkali-metal atoms with  $n \leq 80$ ,” *Phys. Rev. A*, vol. 79, no. 052504, 2009.
- [36] MLOOP source code and documentation found at URL: <http://m-loop.readthedocs.io>.
- [37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [38] S. Helmrich, A. Arias, N. Pehoviak, and S. Whitlock, “Two-body interactions and decay of three-level Rydberg-dressed atoms,” *J. Phys. B: At. Mol. Opt. Phys.*, vol. 49, no. 03LT02, 2016.
- [39] P. Schauss, *High-resolution imaging of ordering in Rydberg many-body systems*. PhD thesis, LMU München: Fakultät für Physik, 2014.
- [40] Scikit-learn documentation found at URL: [http://scikit-learn.org/stable/modules/gaussian\\_process.html#gaussian-process](http://scikit-learn.org/stable/modules/gaussian_process.html#gaussian-process).