

INTERACTING WITH HADWARE

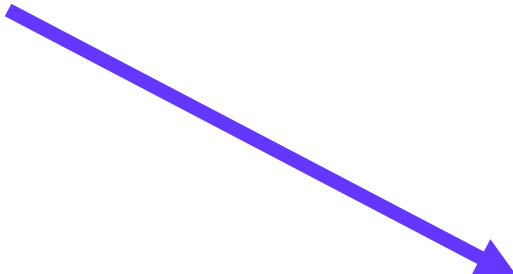
Bridge the Gap

- Between Bloqade and hardware!

$$\frac{H}{\hbar} = \sum_i \frac{\Omega(t)}{2} (e^{i\phi(t)} |g_i\rangle\langle r_i| + e^{-i\phi(t)} |r_i\rangle\langle g_i|) - \sum_i \Delta_i(t) n_i + \sum_{i < j} V_{ij} n_i n_j$$



```
BloqadeSchema.submit_to_braket
```



```
emulate!(prob)
```

Learning Objectives

By the end of this class, you will be able to:

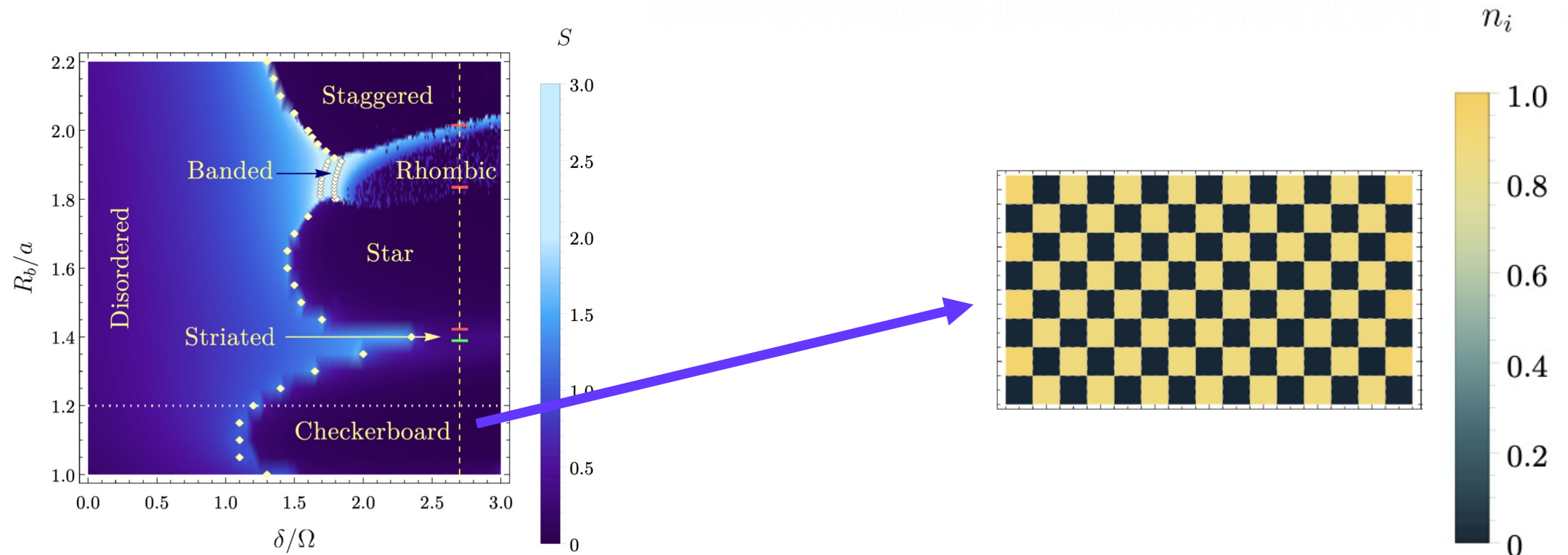
- **Describe** the Bloqade to Hardware pipeline
- **Understand** how to work within Hardware Constraints
- **Design, Submit, and Retrieve** Hamiltonians for Hardware

Big Picture

- 1. **Define** your problem
 - If you can, emulate to verify dynamics!
- 2. **Transform** to Hardware
- 3. **Validate**
- 4. **Tweak** problem (if necessary)
- 5. **Submit!**
- 6. **Retrieve** data

Start With A Problem

- Attempt to recreate **2D Checkerboard Phase**



R. Samajdar et al., Phys. Rev. Lett. 124, 103601 (2020)

Why Transform?

- Hardware has some important limitations to consider
 - Atoms have a minimum distance they can be placed next to each other
 - Have finite-valued Rabi, Detunings, and Phase
 - Final waveforms on hardware must be piecewise Linear/Constant (discretization necessary)
 - Your slope or “slew rate” cannot exceed certain maximums
 - Minimum time resolution to consider (smallest increment of time you can define)

Find Hardware Constraints documented Here:
<https://queracomputing.github.io/Bloqade.jl/dev/capabilities/>

Global Rydberg Values

Capability	Field	Value
Rydberg Interaction Constant	<code>c6_coefficient</code>	$5.42 \times 10^6 \text{ rad}/\mu\text{s} \times \mu\text{m}^6$
Minimum Rabi Frequency	<code>rabi_frequency_min</code>	$0.00 \text{ rad}/\mu\text{s}$
Maximum Rabi Frequency	<code>rabi_frequency_max</code>	$15.8 \text{ rad}/\mu\text{s}$
Rabi Frequency Resolution	<code>rabi_frequency_resolution</code>	$0.0004 \text{ rad}/\mu\text{s}$
Maximum Rabi Frequency Slew Rate	<code>rabi_frequency_slew_rate_max</code>	$250.0 \text{ rad}/\mu\text{s}^2$
Minimum Detuning	<code>detuning_min</code>	$-125.0 \text{ rad}/\mu\text{s}$
Maximum Detuning	<code>detuning_max</code>	$125.0 \text{ rad}/\mu\text{s}$
Detuning Resolution	<code>detuning_resolution</code>	$2.0 \times 10^{-7} \text{ rad}/\mu\text{s}$
Maximum Detuning Slew Rate	<code>detuning_slew_rate_max</code>	$2500.0 \text{ rad}/\mu\text{s}^2$
Minimum Phase	<code>phase_min</code>	-99.0 rad

Objective of Transform

- **MAXIMIZE** your ability to design arbitrary Hamiltonians
- **MINIMIZE** the need to keep all considerations in mind

Validation

- Occasionally, there are certain things that can't be automatically transformed
 - Most commonly with atom position constraints
- Require some form of user intervention, this is where validation is necessary!
- Treatable as a *catch-all* for any incompatible Hamiltonians

Submitting

- If the Hamiltonian passes Validation, you'll need your AWS Credentials
- Upon submission, may need to wait a bit as tasks go on a queue that the machine will consume from when it's open

Tuesdays 16:00:00 to 20:00:00 UTC

Wednesdays 16:00:00 to 20:00:00 UTC

Thursdays 16:00:00 to 18:00:00 UTC

*Outside of hours, tasks can be submitted to the Amazon Braket queue

Retrieval

- Heavy lifting done by Braket.jl
- Results can be saved in HDF5-Compatible format or JSON for usage inside Python
 - JSON is the friendlier format!

Summary

- **Describe** the Bloqade to Hardware pipeline
- **Understand** how to work within Hardware Constraints
- **Design, Submit, and Retrieve** Hamiltonians for Hardware