

a) Inputs

Stoquastic Hamiltonian:

$$H = -\Gamma + V$$

- General functions for:

- Calculating the passive rates $\Gamma_{s \rightarrow s'}(s)$
- Potential $V(s)$

- Defines:

- Adjacent states $s \neq s'$
- Action needed to transform $s \rightarrow s'$

Variational model of the rates:

- Parameterisation of the rates $\Gamma_{s \rightarrow s'}^{(v)}(s) = \text{Model}(s)$
- Subclass of *flax.linen.Module*, implements *module.apply(s)*
- Examples:
 - pCNN
 - Encoder-Decoder
 - Group equivariant pCNN

Hyperparameters:

- Optimiser parameters (learning rate, ...)
- Network architecture (width, depth, ...)
- Physical parameters (J, g, L, \dots)
- Simulation parameters ($T, \text{dimension}$)
- training details (no. epochs, N_b, \dots)

b) Trainer

Compile, vectorise, parallelise functions that do not change signature each epoch.

Training loop

