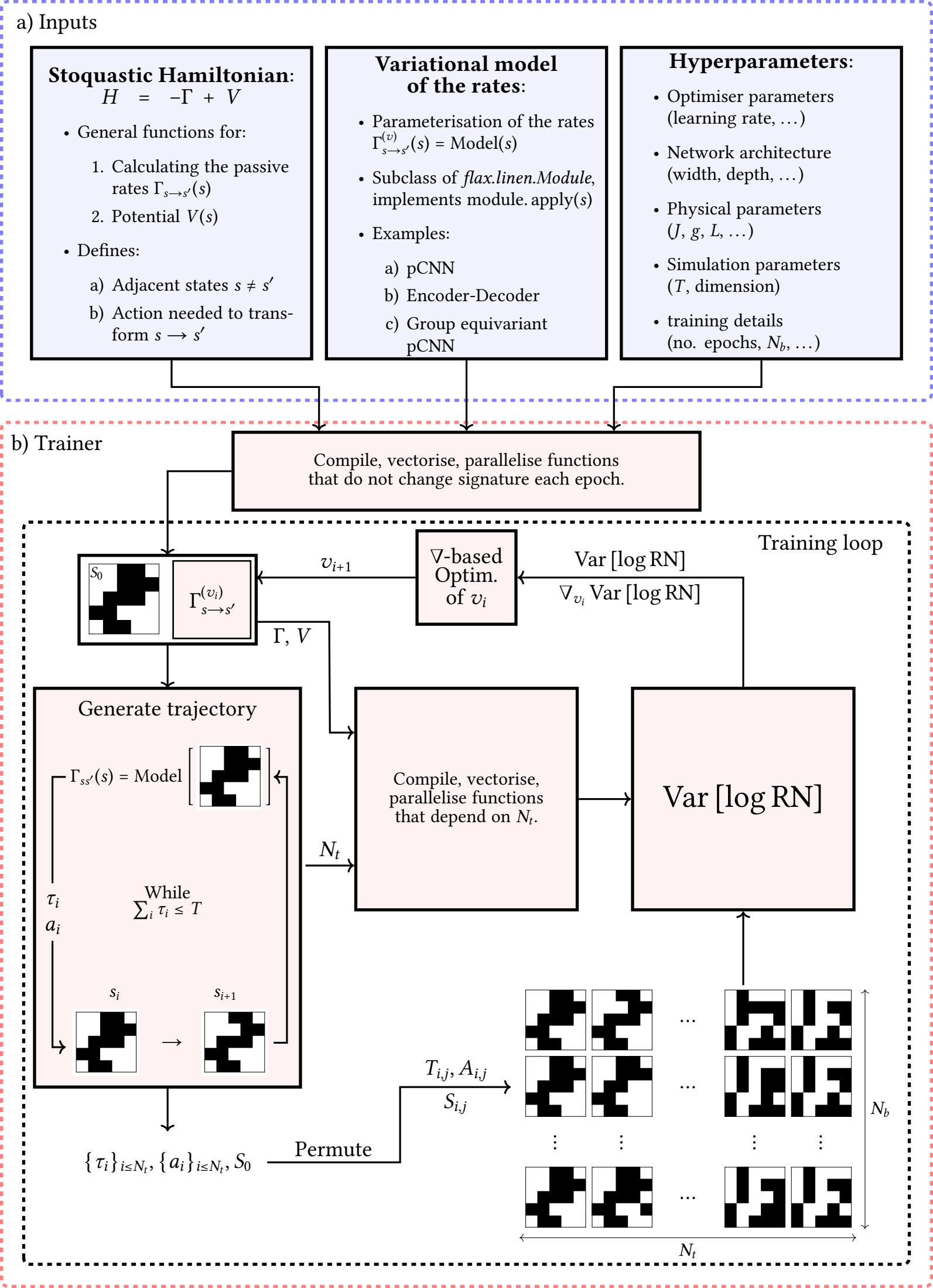
a) Inputs
Stoquastic Hamiltonian:
$H = -\Gamma + V$
• General functions for:
1. Calculating the passive rates $\Gamma_{s\to s'}(s)$
2. Potential $V(s)$
• Defines:
a) Adjacent states $s \neq s'$
b) Action needed to transform $s \to s'$
Variational model of the rates:
• Parameterisation of the rates $\Gamma^{(v)}_{s\to s'}(s)$ = Model(s)
• Subclass of *flax.linen.Module*, implements module. apply(s)
• Examples:
a) pCNN
b) Encoder-Decoder
c) Group equivariant pCNN
Hyperparameters:
• Optimiser parameters (learning rate, …)
• Network architecture (width, depth, …)
• Physical parameters (J, g, L, …)
• Simulation parameters (T, dimension)
• training details (no. epochs, N_b, …)
b) Trainer
Compile, vectorise, parallelise functions that do not change signature each epoch.
Training loop
S_0
$\Gamma^{(v_i)}_{s\to s'}$
v_{i+1}
∇-based Optim. of v_i
Var [log RN]
∇_{v_i} Var [log RN]
Γ, V
Generate trajectory
$\Gamma_{ss'}(s)$ = Model
τ_i
a_i
While $\sum_i \tau_i \leq T$
s_i
s_{i+1}
N_t
Compile, vectorise, parallelise functions that depend on N_t.
Var [log RN]
$\{\tau_i\}_{i\leq N_t}, \{a_i\}_{i\leq N_t}, S_0$
Permute
$T_{i,j}, A_{i,j}$
$S_{i,j}$
N_b
N_t