Application of Reinforcement Learning to the development of theoretical analysis methods

(w/ Alpha Zero For Physics)

Yoshihiro Michishita(Riken, CEMS)

arXiv: 2311.12713(2023)

（codes: https://github.com/YoshihiroMichishita/julia/AlphaZeroForPhysics ）

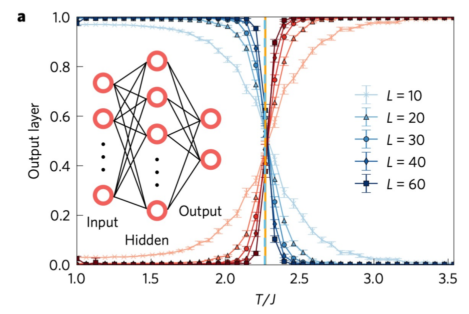
Outline

- **Introduction**
  - Machine Learning & Physics
  - What is “Theoretical Analysis method” ?
  - Purpose
- **Quick Review**
  - Tree representation of equations and its game
  - Reinforcement Learning
  - Alpha Zero
- **Results**

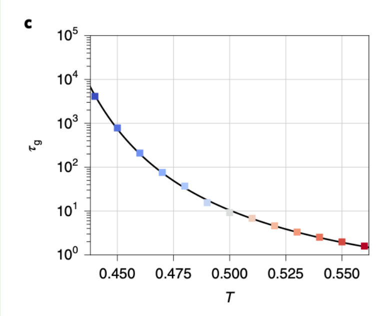
Recent application of ML to physics

(from the low-energy point of view)

Detect the phase transition



Ising magnetization
(Nat. Phys: 13.431(2017))



glass transition
(Nat. Phys: 10.1038(2020))

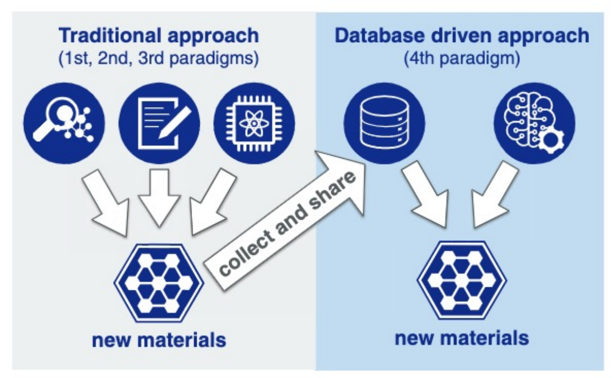
Calcu. equilibrium or steady state

Calc. spin systems by RBM(Science: 335.602-606(2017))
(PRB: 96.205152(2017))

Calc. steady state by RBM (PRB: 99.214306(2019))

Calc. GS in lattice system by RNN (PRR: 2.023358(2020))

Materials Infomatics



(Advanced Science: 6.1900808(2019))

Remove noise or enhance the accuracy

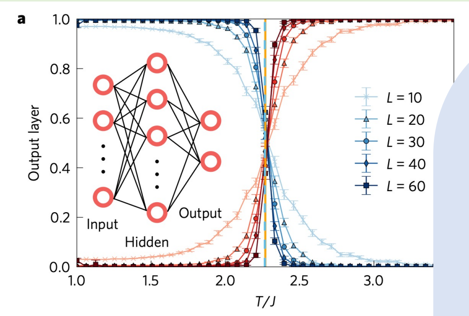
Suppose the Gaussian Process (IEICE: 10.1587(2010))

Applied to NV center (Sci. Rep.: 12.13942(2022))

Recent application of ML to physics

(from the low-energy point of view)

Detect the phase transition



Ising magnetization
(Nat. Phys: 13.431(2017

Calcu. equilibrium or steady state

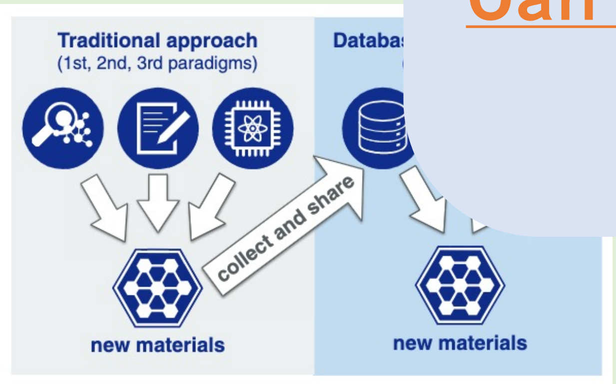
ience: 335.602-606(2017))
3: 96.205152(2017))
3: 99.214306(2019))

NN (PRR: 2.023358(2020))

Usually, machine learning is used for supporting the experiment or numerical simulation. . .

Can we use it for theoretical analysis?

Materials Infor



(Advanced Science: 6.1900808(2019))

e the accuracy

(IEICE: 10.1587(2010))

Applied to NV center (Sci. Rep.: 12.13942(2022))

What is a “Theoretical Analysis method” ?

●Scale separation & Reduction

- ➢ Nonlinear system => reduction
- ➢ The Hubbard model (Lattice model)=> Heisenberg model
- ➢ Open quantum system => Markov app., GKSL equation
- ➢ Periodic driving system => high-frequency expansion
- ➢ (Renormalization Group => cutoff scale)
- ➢ (DMRG, Tensor network => SVD & reduction)

It usually needs appropriate Unitary transformation or projection

＊The two bottom examples are classified to “scale separation & reduction” because we introduce the cutoff scale by hand and perform reduction in them.

What is a “Theoretical Analysis method” ?

●Points

- ➢ If there is a scale separation, we can perform the reduction.
- ➢ In order to perform the perturbation treatment or the reduction
  we have to find the frame where above treatment is justified

Thus, we can sum up the analytic method (by hand) into three step:

Want the reinforcement learning to find

1. **(When there is a scale separation) Find the appropriate frame**
2. Perform the reduction or perturbation to construct an effective model
3. Analyze the physical properties in the derived effective model.

Why Reinforcement Learning?

- **It's difficult to gather the data in theoretical research**
  - If we gather the data by the conventional simulation methods,
    it should be just the replacement of the old methods and cannot surpass them
  - Cannot verify the validity where the conventional approach cannot access.
  - Usually the machine resource is essential (Difficult to get in academia.)
- **Not necessary to prepare the data**
  - The algorithm is essential.
  - Not so-heavy calculation. (I could run my code in my M1 MBA.)

Outline

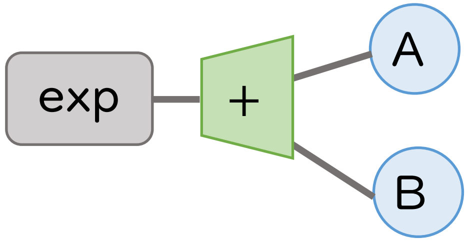
- **Introduction**
  - Machine Learning & Physics
  - What is “Theoretical Analysis method” ?
  - Purpose
- **Quick Review**
  - Tree representation of equations and its game
  - Reinforcement Learning
  - Alpha Zero
- **Results**

Tree representation of equation

- **Equation is a tree (≃ game)**

Tree can describe any equation with three types of nodes;
Function, Branch, and Variable

$\exp[A + B] =$



e.g.) When focusing on the unitary transformation and its -ilog, it should be Hermitian, and the nodes are

Function···$\sum_i$, exp, log, $\int dx$, ∂_x Branch··· +, −, −i[,], { , } Variable···$\frac{\psi_i+\psi_i^\dagger}{2}$, $-i(\psi_i - \psi_i^\dagger)$

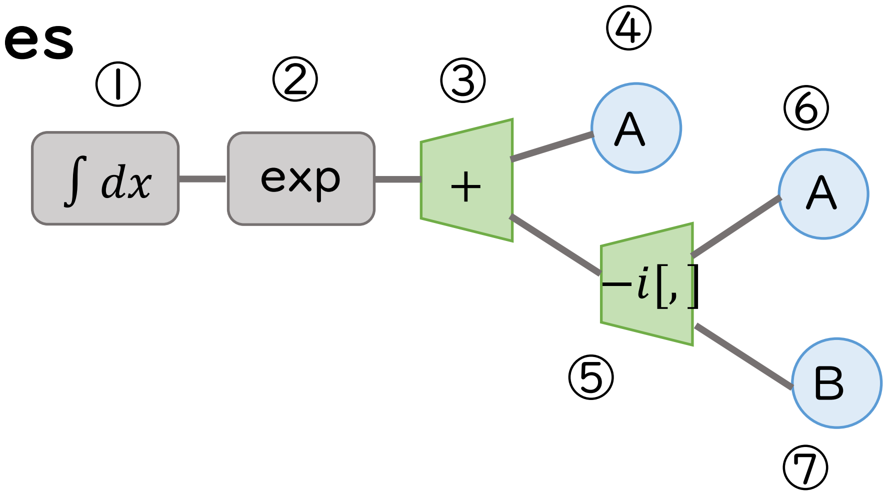
(この場合は、多体系でも手の広さは将棋の中盤と割と同じくらい？)

Rule of the game

● The rule of making an equation tree

・procedure of adding nodes

$$\int dx \exp[A - i[A, B(x)]] =$$



・Rule

1. Must satisfy (# of Variable) ≤ (# of Branch+1) (When the equality satisfied, equation is completed.)

2. Not consecutive the same node (due to the symmetry of the operation)

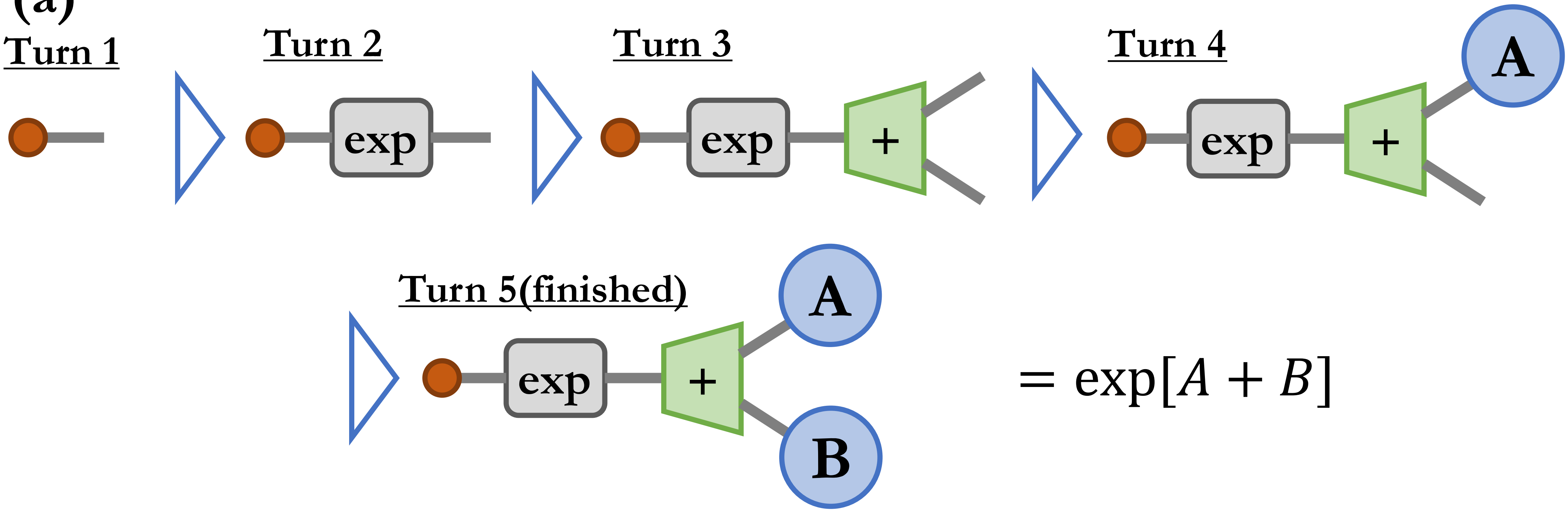
3. If function nodes include a function and its inverse, do not consective (forbid redundancy)

Function…$\sum_i$, exp, log, $\int dx$, ∂_x Branch… +, −, −i[,], { , } Variable…ψ_i, $\psi_i^\dagger$

Rule of the game

- Example

(a)



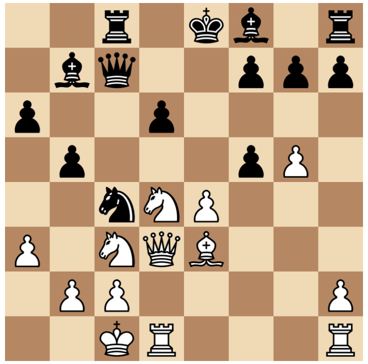
$$= \exp[A + B]$$

Quick review of RL

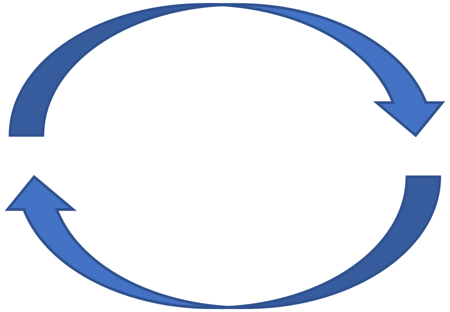
Reinforcement Learning

- consists of the environment and the agent
- Agent can know only the state of environment and the reward
- Agent can change the state of the environment by his/her action

Environment

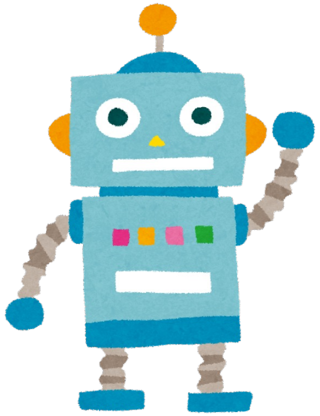


state: $s_t \in \mathcal{S}$, reward: r_t



Action: $a_t \in \mathcal{A}$

Agent



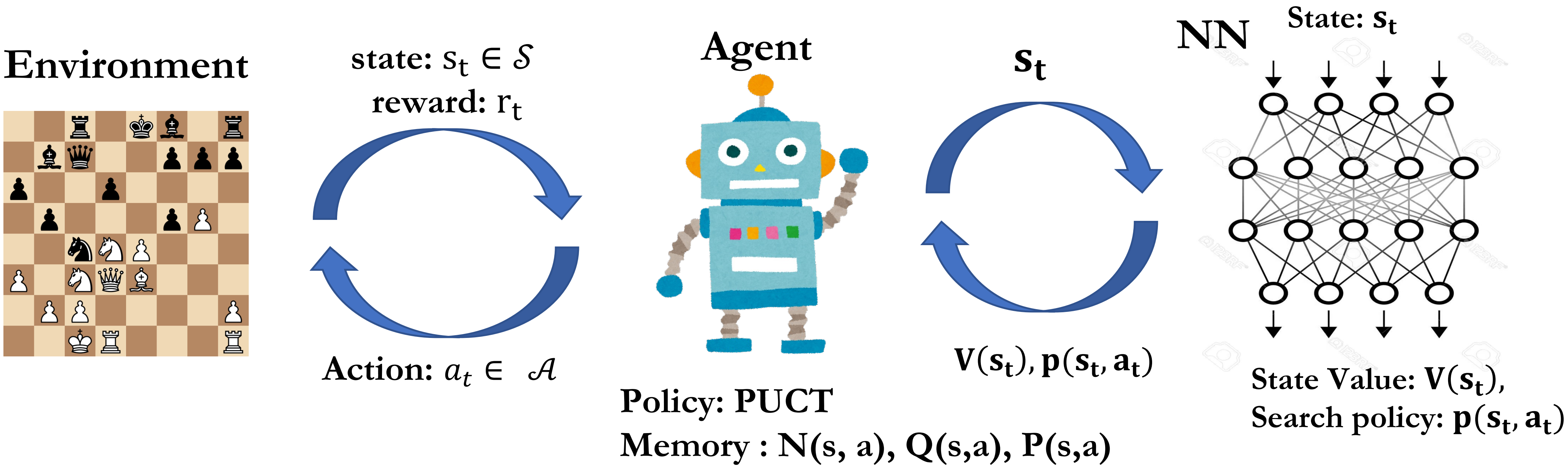
Policy: π $(s_t \rightarrow a_t)$

Reward : R (s_t, a_t)
tr. prob.: $p(s_{t+1}|s_t, a_t)$

(e.g.) update the policy to maximize $\sum_t r_t$
tradeoff of exploration and exploitation

Scratch of Alpha Zero

- Consists of an agent with neural network and environment



•Player

➢ **UCT(Upper Confidential Bound (applied to Trees))**

$$a_t = \mathrm{argmax}_{\{a \in A\}}[q_t(s_t, a) + c\sqrt{\frac{\log(\sum_a m_t(s_t, a) + 1)}{m_t(s_t, a) + 1}}]$$

(UCB… Regret(Efficiency badness of search) is upper bounded by O(log(t))

Estimate value (Exploitation)

Try small-experienced action (Exploitation)

Practice: select action until the game ends, and update N(s,a) and Q(s,a).
Repeat Practice and update the statistics
Serious game: select most practiced action.
(After some games, the experiences are used to the learning of NN.)

➢ **P+UCT**

$$a_t = \mathrm{argmax}_{\{a \in A\}}[q_t(s_t, a) + c\, p(s_t, a)\sqrt{\frac{\log(\sum_a m_t(s_t, a))}{m_t(s_t, a)}}]$$

Policy for searching (important for deep search)

Alpha Zero

• Neural networks and its learning

➢ Architecuture

(CNN+ Dual(Dueling) network)

(3*3 Convolution+BatchNorm+ReLu+ResNet)*19 (judge the situation)

1*1 Conv. +BatchNorm+ReLu
(predictive)(advice)

BatchNorm+ReLu + tanh
(Win rate)(estimated value of state)

➢ Learning

$$L(m) = \frac{1}{N}\sum_{n}\left(r_n - v(m(s_n))\right)^2 - \pi_n \log p(m(s_n)) + \eta \sum m.w^2$$

MSE of Value estimation — effectiveness of advices — weight decay (Ban hard coaching)

Outline

- **Introduction**
  - Machine Learning & Physics
  - What is "Theoretical Analysis method" ?
  - Purpose
- **Quick Review**
  - Tree representation of equations and its game
  - Reinforcement Learning
  - Alpha Zero
- **Results**

Quick review of Floquet theory & RF

➢ Formalism

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t) \qquad i\frac{d}{dt}|\psi(t)> = \hat{H}(t)|\psi(t)> \qquad \hat{U}(t) = \exp[i\hat{K}(t)]$$

$$i\frac{d}{dt}|\tilde{\psi}(t)> = i\frac{d}{dt}\hat{U}(t)|\psi(t)> = \hat{H}_r(t)|\tilde{\psi}(t)> \qquad \hat{H}_r(t) = \hat{U}(t)(\hat{H}(t) - i\partial_t)\hat{U}^\dagger(t)$$

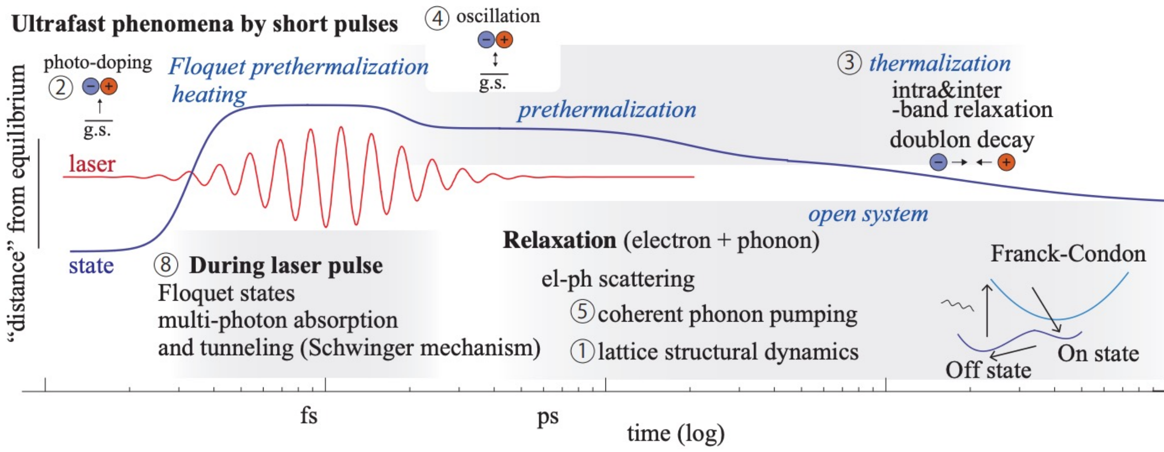
When the deriving is periodic $\hat{H}(t) = \hat{H}(t+T)$, There is $U_F(t)$ satisfying $\hat{U}_F(t) = \hat{U}_F(t+T)$, $\hat{H}_r(t) = \hat{H}_F$
In the high-frequency regime $||H_0|| \ll \Omega$, There are methods which perturbatively construct $U_F(t), H_F$

$$H_r(t) = H_F^{(n)} + O(\frac{1}{\Omega^{n+1}}, t)$$

➢ Floquet pre-thermalization

When Hamiltonian is local and the time $t = mT$,
(arXiv:1509.03968(2015))

$$||\mathcal{T}\exp[-i\int ds\, H(s)] - \exp\left[-iH_F^{(n)}t\right]|| \leqq \exp[-O(\Omega)]$$



Quick review of Floquet theory & RF

➢ Formalism

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t) \qquad i\frac{d}{dt}|\psi(t)> = \hat{H}(t)|\psi(t)> \qquad \hat{U}(t) = \exp[i\hat{K}(t)]$$

$$i\frac{d}{dt}|\tilde{\psi}(t)> = i\frac{d}{dt}\hat{U}(t)|\psi(t)> = \hat{H}_r(t)|\tilde{\psi}(t)> \qquad \hat{H}_r(t) = \hat{U}(t)(\hat{H}(t) - i\partial_t)\hat{U}^\dagger(t)$$

When the deriving is periodic $\hat{H}(t) = \hat{H}(t+T)$, There is $U_F(t)$ satisfying $\hat{U}_F(t) = \hat{U}_F(t+T)$, $\hat{H}_r(t) = \hat{H}_F$

In the high-frequency regime $||H_0|| \ll \Omega$, There are methods which perturbatively construct $U_F(t), H_F$

$$H_r(t) = H_F^{(n)} + O(\frac{1}{\Omega^{n+1}}, t)$$

➢ Floque

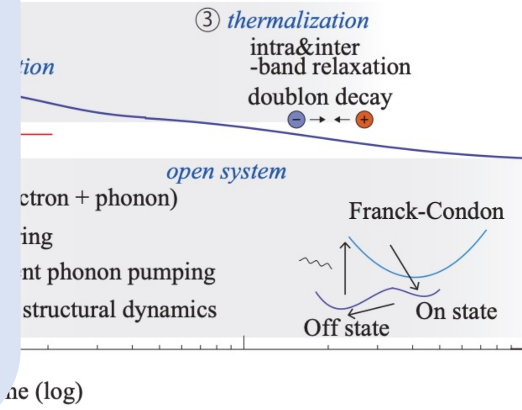
When Har

(arXiv:150

$||\mathcal{T} \exp[-i\int d$

• Demonstration

Can Alpha Zero for Physics derive “High-frequency expantion” ?



Demonstration

●Model

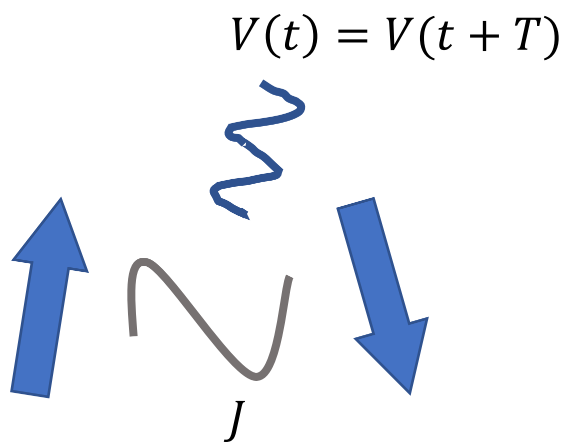
➢ Interacting quantum Two-Spin model under driving

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t)$$

$$\widehat{H_0} = -\sum_{\alpha} (J_\alpha \widehat{S_1^\alpha} \otimes \widehat{S_2^\alpha} + h_\alpha \sum_i \widehat{S_1^\alpha})$$

$$\hat{V}(t) = -\sum_{\alpha} \xi_\alpha sin(\Omega t) \sum_i \widehat{S_i^\alpha}$$

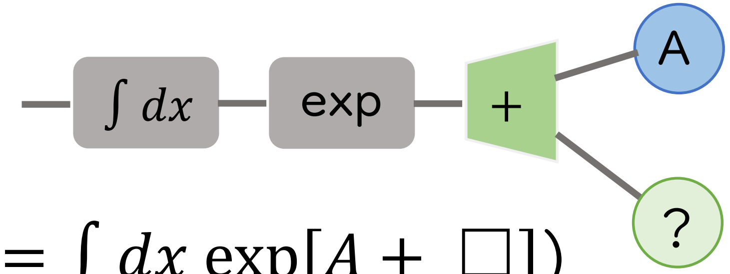
$$\vec{J} = (J_x, J_y = 0, J_z), \qquad \vec{h} = (h_x = 0, h_y = 0, h_z), \vec{\xi} = (\xi, 0, 0)$$



Setup

●Algorithm

Input: state of environment
= equation in the making

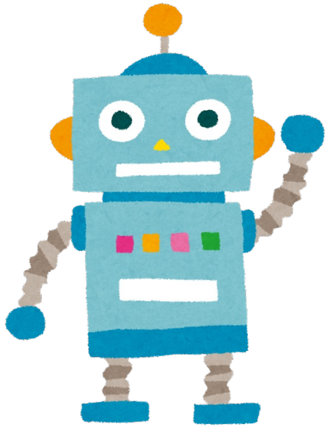


$(= \int dx \exp[A + \square])$

①Tell the state

②select the next symbol

Agent(Alpha Zero)

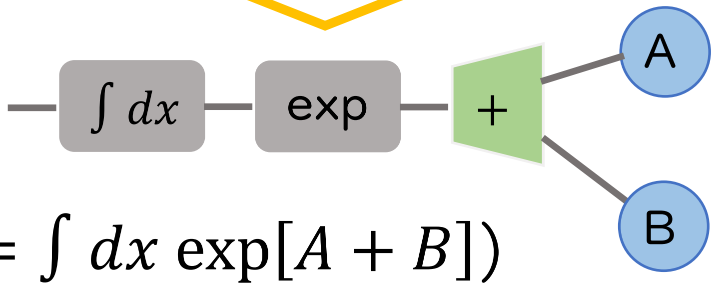


NN calculate(speculate) the evaluation values (output)

⑤Learning from the actual score

候補手	A	B	$-i[,]$	$-$	$\int dt$
評価値	−1	**4**	1	0	2

③Update the equation



$(= \int dx \exp[A + B])$

④ If the equation is completed, calculate the score of the game

Score(reward)

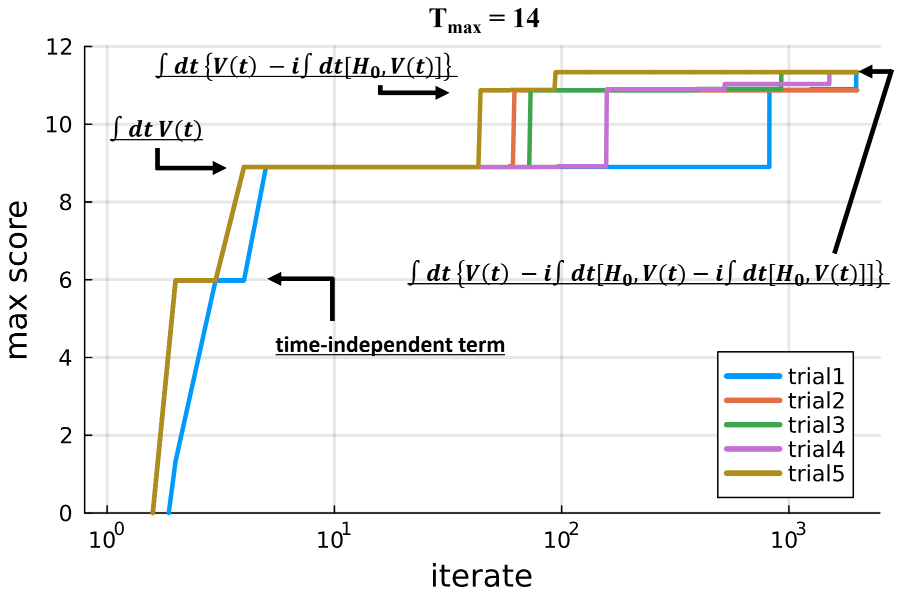
$= -log\int dt\ tr(\widehat{H_r}(t) - \hat{H}_r(t - \delta t))^2$

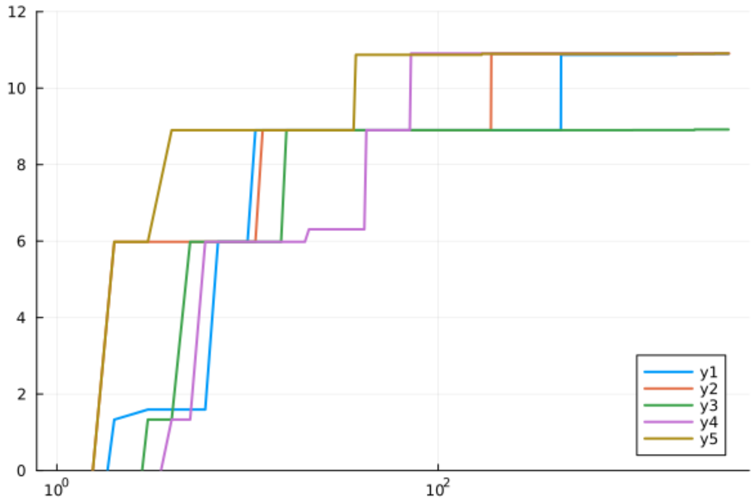
Results

- **Alpha Zero VS e-greedy ($T_{max} = 14$)**

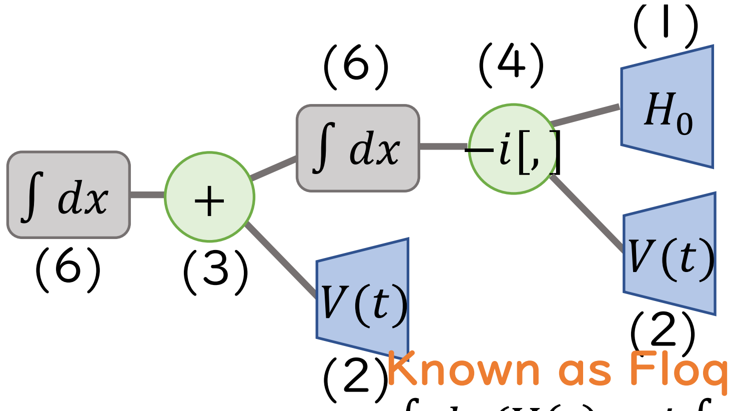
AlphaZero

ε -greedy methods(old approach)






```
val=1.5820317, pol=0.61543405
val=1.1977102, pol=0.60332686
val=1.0977467, pol=0.695877
val=1.2222252, pol=0.6558915
val=1.6906229, pol=0.6552366
val=2.8830054, pol=0.69791865
 68.607700 seconds (107.28 M allocations: 28.289 GiB, 4.51% gc time)
store data
435
------------
head = 1;
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
```



Known as Floquet-Magnus!

$$= \int dt\, (V(t) - i\int dt\, [H_0, V(t)])$$

Remarks:

Model:

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t)$$

$$\widehat{H_0} = \sum_{\alpha} (J_\alpha \widehat{S_1^\alpha} \otimes \widehat{S_2^\alpha} + h_\alpha \sum_i \widehat{S_1^\alpha})$$

$$\hat{V}(t) = -\sum_\alpha \xi_\alpha sin(\Omega t) \sum_i \widehat{S_i^\alpha}$$

Formalism:

$$\hat{H}_r(t) = \hat{U}(t)(\hat{H}(t) - i\partial_t)\hat{U}^+(t)$$

$$\hat{U}(t) = \exp[i\hat{K}(t)]$$

$$\hat{K}'(t) = \frac{d}{dt}\hat{K}(t)$$

Reward:

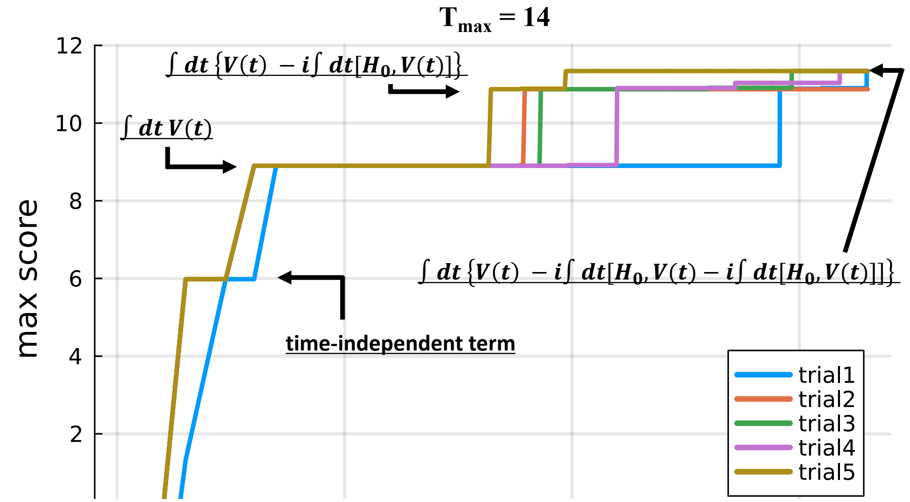
$$-log\int dt\, tr(\widehat{H_r}(t) - \widehat{H}_r(t-\delta t))\text{^}2$$

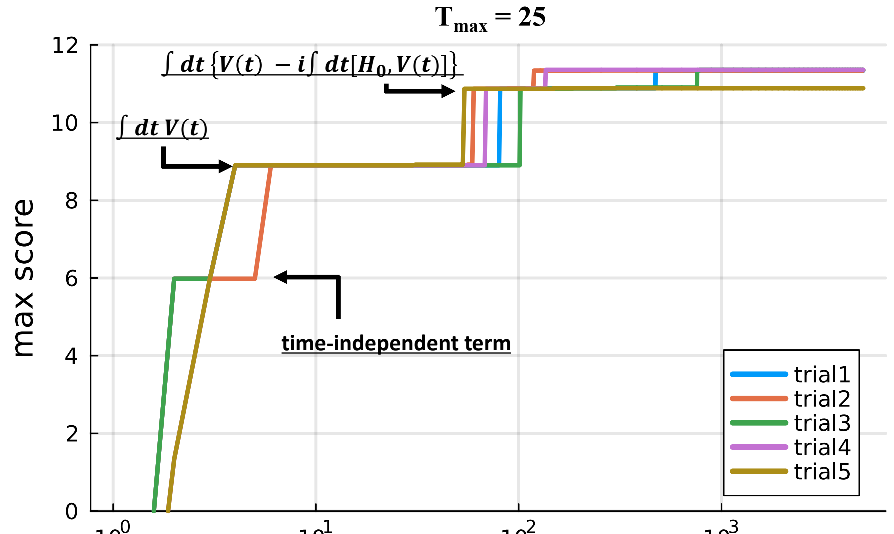
Find a tree maximizing reward!

Results

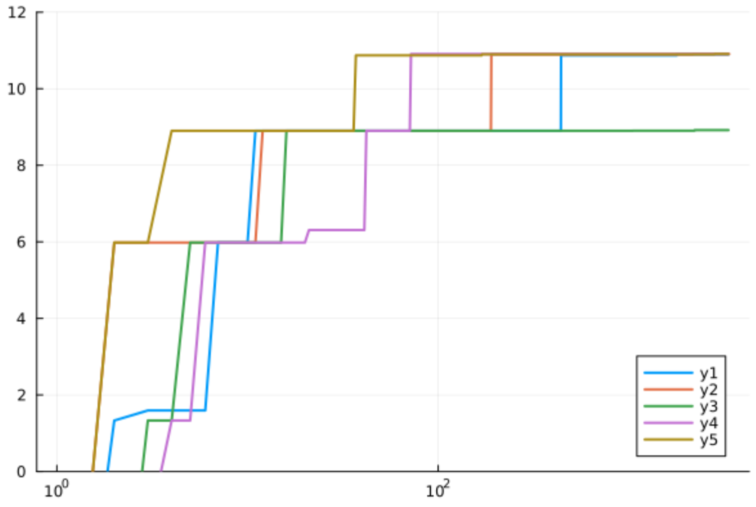
- **Alpha Zero VS e-greedy ($T_{max} = 14$)**

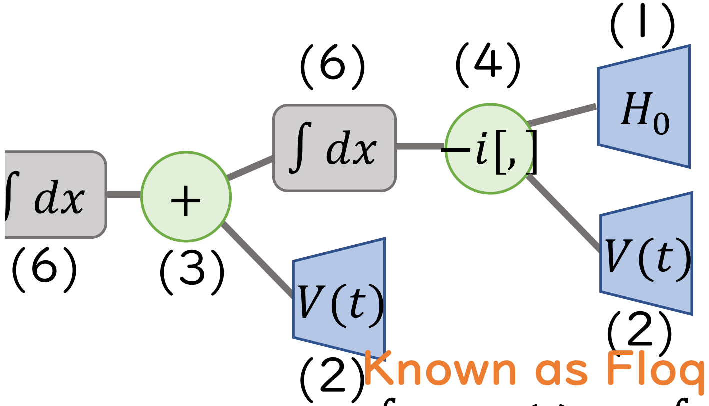
AlphaZero





ε -greedy methods(old approach)





Known as Floquet-Magnus!

$$= \int dt\, (V(t) - i \int dt\, [H_0, V(t)])$$

Remarks:

Model:

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t)$$

$$\widehat{H_0} = \sum_{\alpha} (J_\alpha \widehat{S_1^\alpha} \otimes \widehat{S_2^\alpha} + h_\alpha \sum_i \widehat{S_1^\alpha})$$

$$\hat{V}(t) = -\sum_{\alpha} \xi_\alpha \sin(\Omega t) \sum_i \widehat{S_i^\alpha}$$

Formalism:

$$\hat{H}_r(t) = \hat{U}(t)(\hat{H}(t) - i\partial_t)\hat{U}^+(t)$$

$$\hat{U}(t) = \exp[i\hat{K}(t)]$$

$$\hat{K}'(t) = \frac{d}{dt}\hat{K}(t)$$

Reward:

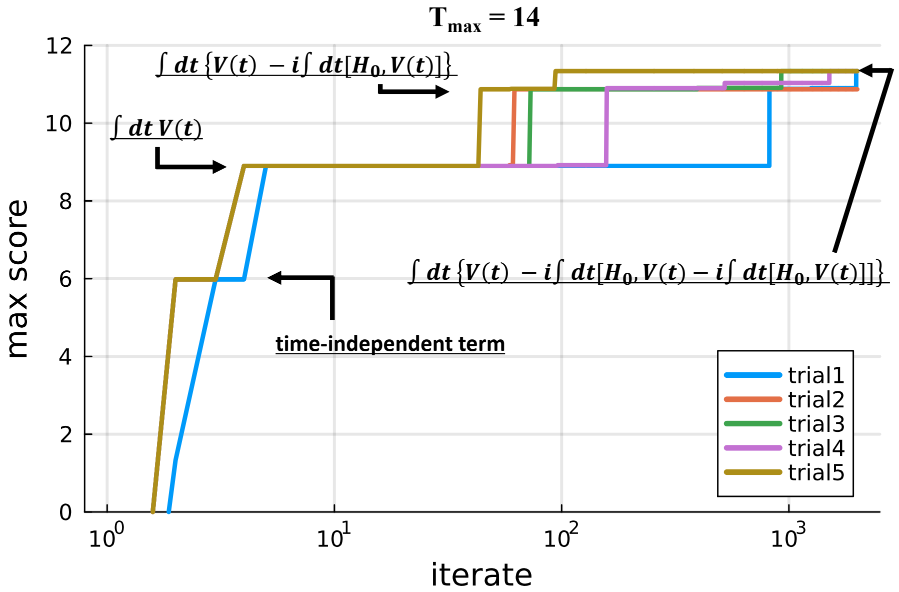
$$-\log \int dt\, tr(\widehat{H}_r(t) - \widehat{H}_r(t - \delta t))\text{^}2$$

Find a tree maximizing reward!

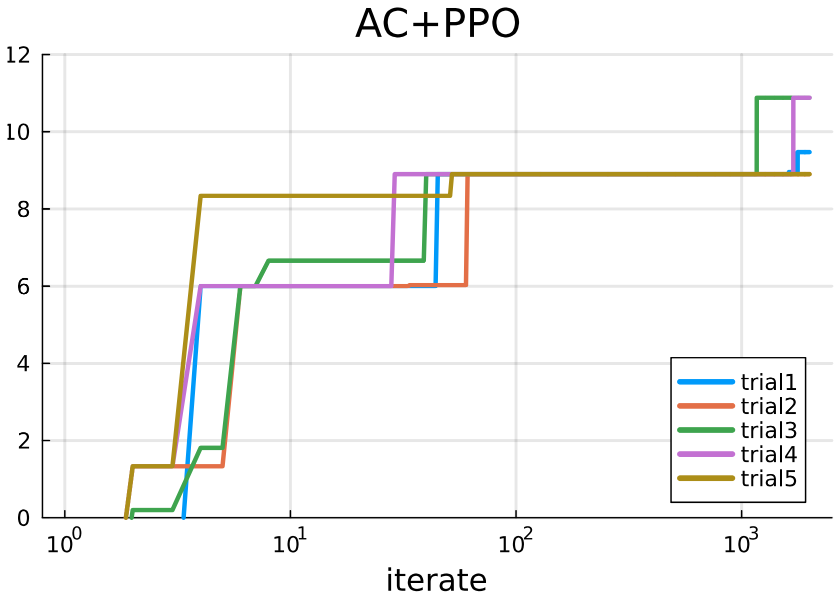
Results

- **Alpha Zero VS AC+PPO ($T_{max} = 14$)**

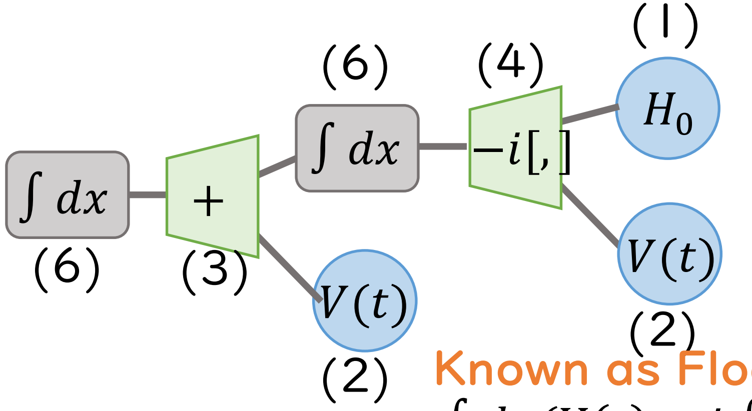
AlphaZero



Actor-Critic+PPO




```
it=2;
496.071510 seconds (401.56 k allocations: 134.160 MiB, 0.01% gc time, 0.02%
val=1.5820317, pol=0.61543405
val=1.1977102, pol=0.60332686
val=1.0977467, pol=0.695877
val=1.2222252, pol=0.6558915
val=1.6906229, pol=0.6552366
val=2.8830054, pol=0.69791865
 68.607700 seconds (107.28 M allocations: 28.289 GiB, 4.51% gc time)
store data
435
------------
head = 1;
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
```



Known as Floquet-Magnus!

$$= \int dt\ (V(t) - i \int dt\ [H_0, V(t)])$$

Remarks:

Model:

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t)$$

$$\widehat{H_0} = \sum_{\alpha} (J_\alpha \widehat{S_1^\alpha} \otimes \widehat{S_2^\alpha} + h_\alpha \sum_i \widehat{S_1^\alpha})$$

$$\hat{V}(t) = -\sum_\alpha \xi_\alpha sin(\Omega t) \sum_i \widehat{S_i^\alpha}$$

Formalism:

$$\hat{H}_r(t) = \hat{U}(t)(\hat{H}(t) - i\partial_t)\hat{U}^+(t)$$

$$\hat{U}(t) = \exp[i\hat{K}(t)]$$

$$\hat{K}'(t) = \frac{d}{dt}\hat{K}(t)$$

Reward:

$$-log \int dt\ tr(\widehat{H}_r(t) - \widehat{H}_r(t - \delta t))\text{^}2$$

Find a tree maximizing reward!

Remarks & Outlook

Remarks

- **Maybe UCT is enough for the Floquet problem**
  - PUCT show its true value when there are many kinds of nodes or deep tree.
  - Just by replacing the part of the environment in code, we can apply to other problems
  - Because not doing self-play, we do not make use of the strong points.
  - In this calculation, we set the maximum length of the equation trees 14, 25
- **This framework is not for the problem where the score calculation needs much time**
  - the system should be simple because we focus on the symbolic representation of equation

Remarks & Outlook

●Outlook

- **Application to the other problems**
  - derivation of the effective model for unknown nonlinear dynamics
  - effective model renormalized from the original model
  - Construction of an efficient quantum circuit.

- **Better algorithm or fine tuning ?**

Summary

Development of theoretical analysis methods
=Finding an equation with good properties

Finding the construction of a tree with good properties

Finding the strategy of the game to high-score

Alpha Zero for Physics can solve (find the strategy)