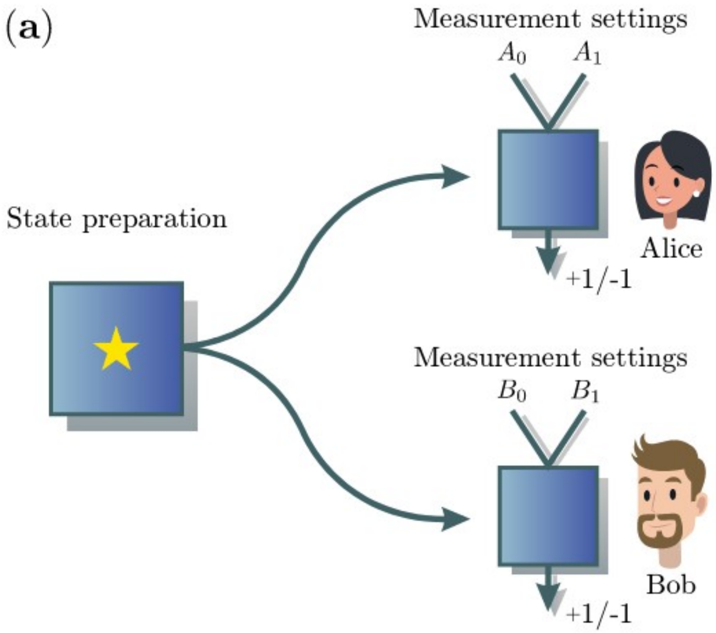
Setting up experimental Bell test with reinforcement learning

A. Melnikov, P. Sekatski, N. Sangouard

- A Bell state is a maximally entangled state
- The Bell inequality shows that no local variables exist in QM → *nonlocality*
- (which has been shown to be a valuable resource when it comes to performing some quantum information tasks in device-independent way)
- Clauser, Horne, Shimony, Holt (CHSH) test – a way to determine how much a state would violate the Bell inequality
- Goal: find setup that would lead to the highest CHSH score

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|a_1 b_1\rangle + |a_2 b_2\rangle) \neq |A\rangle \otimes |B\rangle$$

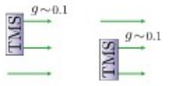
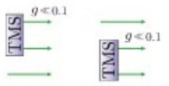
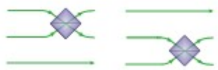
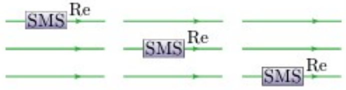
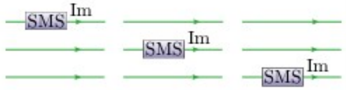
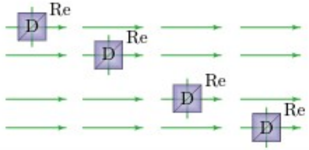
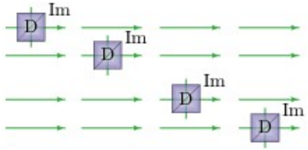


Computing the CSHS score

a, b → outcomes of the measurement for Alice and Bob respectively (possible values are +1/-1)

$$\beta = \sum_{x,y=0}^{1} (-1)^{xy} \left(p(a = b|A_x B_y) - p(a \neq b|A_x B_y) \right)$$

$$|\beta| \leq 2\sqrt{2}$$

Action	Elements name	Elements description	Elements notation
1 – 2	Two-mode squeezing: TMS_{12}, TMS_{23}	$U_{\mathrm{TMS}} = \exp\left(g(a_1^\dagger a_2^\dagger - a_1 a_2)\right)$	
3 – 4	Two-mode squeezing, small squeezing: TMS'_{12}, TMS'_{23}	$U_{\mathrm{TMS}'} = \exp\left(10^{-4} g(a_1^\dagger a_2^\dagger - a_1 a_2)\right)$	
5 – 6	Beam splitters: BS_{12}, BS_{23}	$U_{\mathrm{BS}} = \exp\left(i\theta(a_1^\dagger a_2 + a_2^\dagger a_1)\right)$	
7 – 9	Single-mode squeezing, real value part: $\mathrm{SMS}_1^{\mathrm{Re}}$, $\mathrm{SMS}_2^{\mathrm{Re}}$, $\mathrm{SMS}_3^{\mathrm{Re}}$	$U_{\mathrm{SMS^{Re}}} = \exp\left(\frac{g}{2}\left(a^2 - (a^\dagger)^2\right)\right)$	
10–12	Single-mode squeezing, imaginary value part: $\mathrm{SMS}_1^{\mathrm{Im}}$, $\mathrm{SMS}_2^{\mathrm{Im}}$, $\mathrm{SMS}_3^{\mathrm{Im}}$	$U_{\mathrm{SMS^{Im}}} = \exp\left(-i\frac{g}{2}\left(a^2 + (a^\dagger)^2\right)\right)$	
13–16	Displacements, real value part: $\mathrm{D}_{A_0}^{\mathrm{Re}}, \mathrm{D}_{A_1}^{\mathrm{Re}}, \mathrm{D}_{B_0}^{\mathrm{Re}}, \mathrm{D}_{B_1}^{\mathrm{Re}}$	$U_{\mathrm{D^{Re}}} = \exp\left(\alpha(a^\dagger - a)\right)$	
17–20	Displacements, imaginary value part: $\mathrm{D}_{A_0}^{\mathrm{Im}}, \mathrm{D}_{A_1}^{\mathrm{Im}}, \mathrm{D}_{B_0}^{\mathrm{Im}}, \mathrm{D}_{B_1}^{\mathrm{Im}}$	$U_{\mathrm{D^{Im}}} = \exp\left(i\alpha(a^\dagger + a)\right)$	

n bosonic modes initialised in the ground state, that are then manipulated by single-mode operations

State preparation is complete if the desired outcome of *click* or *no click* (which is realised with probabilities p_{click} and $p_{\text{no click}}$ respectivelly) are observed on *n-m* detectors

The remaining m modes are shared between Alice and Bob who locally apply a combination of operations depending on the measurement settings

All modes are measured

The examples included in the paper would consider the cases {n,m} = {3,2} and {2,2}

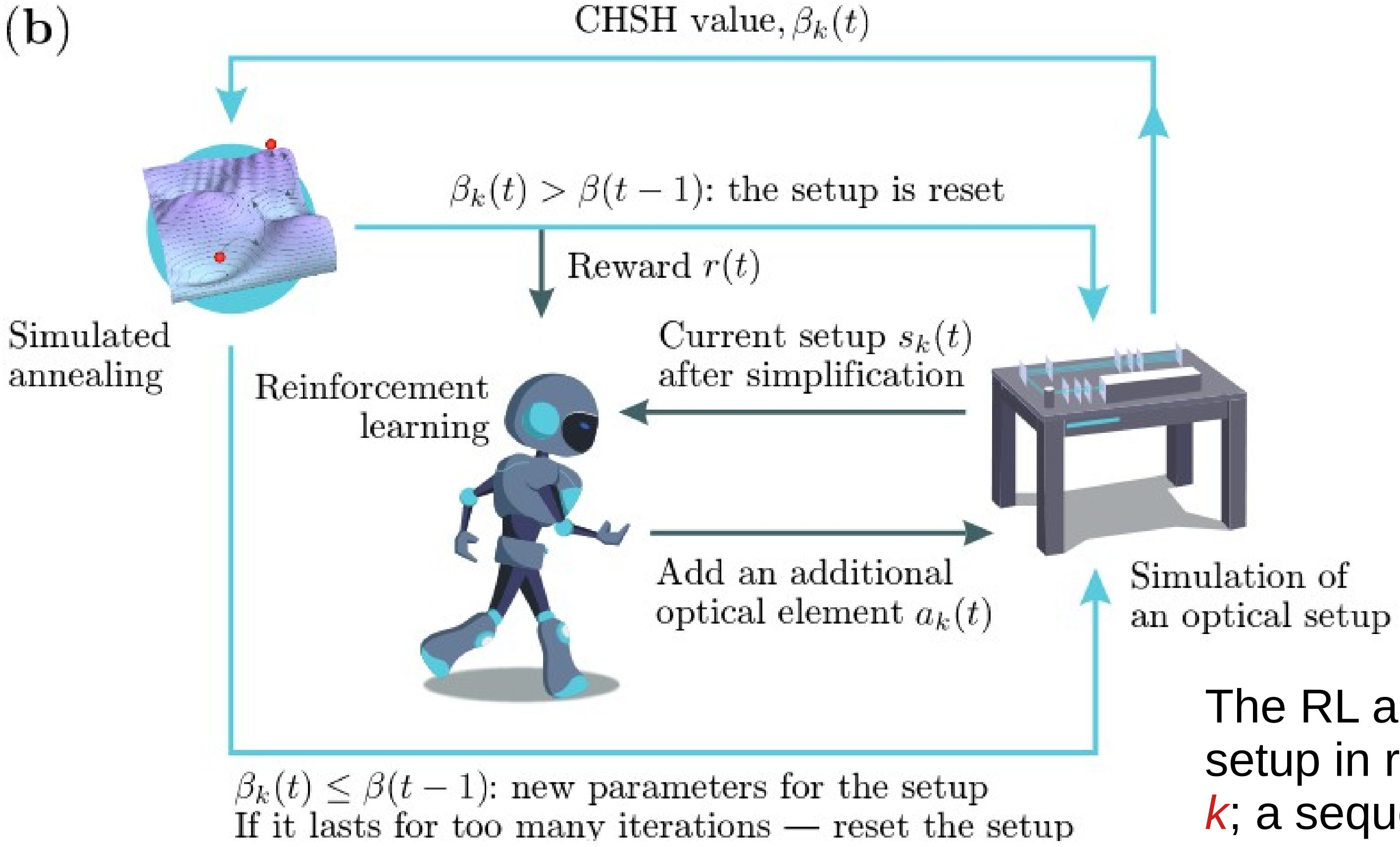
Why machine learning?

- Brute-force approach wouldn’t be as efficient in optimising all the parameters (the total number of parameters must also account for the measurement settings)
- If the elements include imperfections, the set of transformations which is accessible by combining individual elements is in general unknown
- Brute-force search is unsuitable when one of the goals is to keep the number of elements low

2 levels of the RL task:

1) Top level → specifies the order in which the elements are applied to the modes → *RL agent*

2) Second level → specifies the value of the parameters for each element → *Simulated annealing optimization algorithm*



The RL agent interacts with the setup in rounds/interaction steps *k*; a sequence of interaction steps that leads to a feedback signal/reward is called a trial *t*.

k=1 → empty optical setup

Simulated annealing algorithm

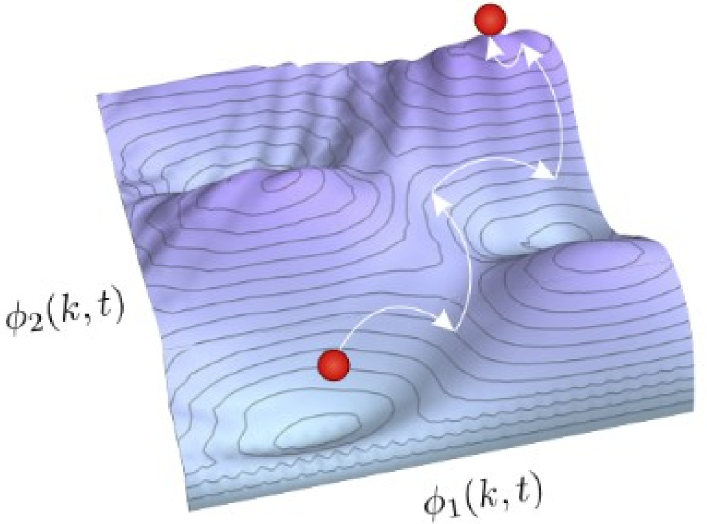
SA emulates the physical process where a solid system is slowly cooled so that eventually its structure is "frozen" at a minimum energy configuration

An optimization tool for the RL agent to design better experimental setups

Set the parameters g, θ, α for each setup $s_k(t)$ so that β is maximised;

Trials marked with t with $l_k(t) \leq k \leq k_{max}$ interaction steps $\rightarrow$

SA is optimising parameters $\Phi(k,t) = (\phi_1(k,t), \ldots, \phi_l(k,t))$ in l-dim space



Visualization for two-parameter optimization; height corresponds to CHSH value

Simulated annealing algorithm

$$\Phi(k,t) = (\phi_1(k,t),\ldots,\phi_l(k,t))$$

$$\phi_{m(i)}(k,t) \to \phi_{m(i)}(k,t) + \xi_i$$

Calculate $\beta_k^{(i)}$:

If $\beta_k^{(i)} > \max_k \beta_k(t) : \beta_k(t) \leftarrow \beta_k^{(i)}$

+ reward, trial t ends

else if $\beta_k^{(i)}(t) < \beta_k^{(i)}(t)$:

revert the change in $\phi_{m(i)}(k,t)$ with probability

$$p_k^{(i)}(t) = 1 - \exp\left(-\frac{\beta_k(t) - \beta_k^{(i)}(t)}{T_i(t)}\right)$$

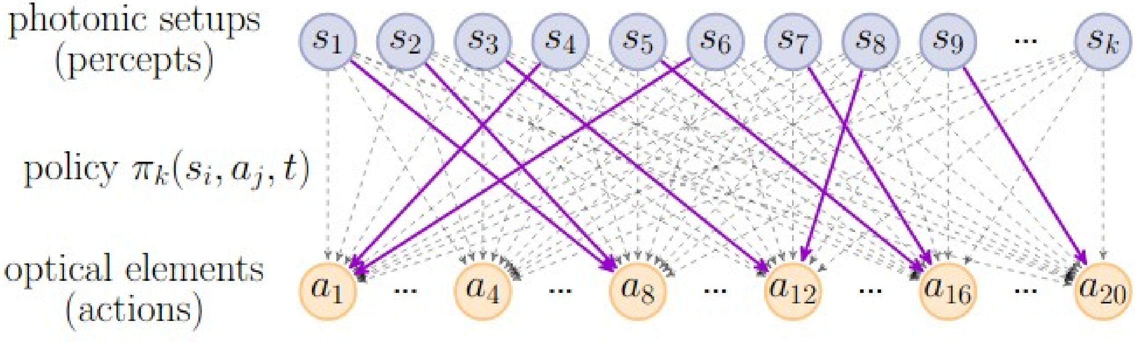
if $i < N(t)$: iteration $i+1$ begins

else if $i = N(t)$ interaction step k ends and setup is reset to $s_0(t+1)$

$$T_i(t) = \frac{N(t)}{i} T_{min}, T_{min} = 0.001$$

Effective temperature of SA,
N(t) = 95+5t,

Projective simulation model



The RL agent is a projective simulation agent which is represented by a two-layer PS network:

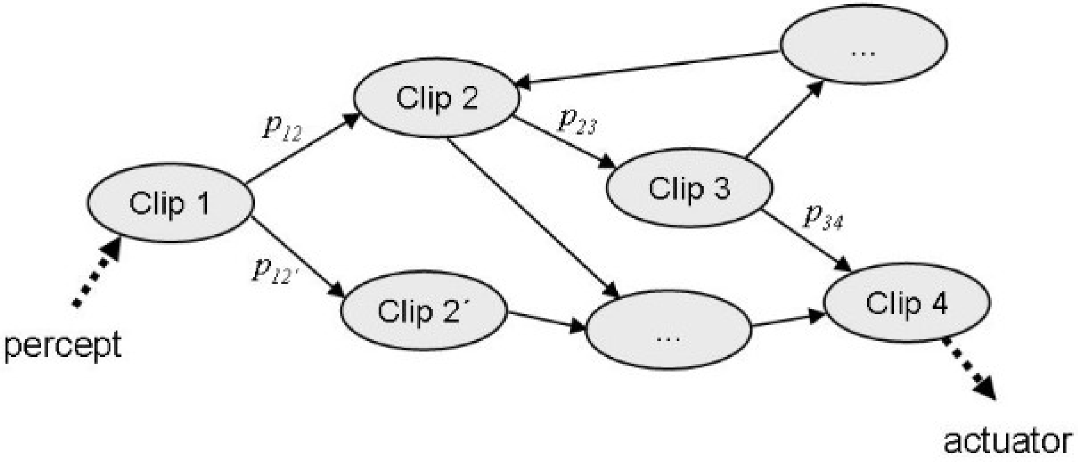
a layer of clips representing the states $s_k(t)$

a layer of clips representing the actions $a_k(t)$

weights $h_k(s, a, t)$ connecting the clips which fully define the policy:

$$\pi_k(s, a, t) = \frac{\exp h_k(s, a, t)}{\sum_{a'} \exp h_k(s, a', t)}$$

- Projective simulation allows the agent to project itself into future situations based on previous experience
- So it can learn not only from previous experience but also from *fictitious experience*
- Fragments of previous experience – *clips* – can be stored in a neural-network-type structure where a perceptual input trigger causes a probabilistic random walk through *episodic memory space*



i) Encounter percept $s \in S$ with probability $P^{(t)}$ which triggers excitation of clip $c \in C$ w/ probablity $I(c|s)$

ii) Random walk through clip space w/ probabilities $p^{(t)}(c'|c)$

iii) Exit of memory through activation of action a described by a function $O(a|c)$

- *Percept space*

$$s = (s_1, \ldots, s_N) \in S_1 \times \cdots \times S_N = S, s_i = 1, \ldots, |S_i|$$

a cartesian product of sets, reflecting the compositional/categorical structure of percepts/objects (ex. colour, shape, size etc)

- *Actuator space*

$$a = (a_1, \ldots, a_M) \in A_1 \times \cdots \times A_M = A, a_i = 1, \ldots, |A_i|$$

reflects the degrees of freedom of the agent's actions

clip ~ sequence of remembered real or fictitious percepts and actions

- *Clip space*

$$c = (c_1, \ldots, c_L) \in C; c_i \in \mu(S) \bigcup \mu(A); L \text{ is the lenght of the clip}$$

- *Emotion space*

consist of tags attached to transitions between different clips in episodic memory/remembered rewards/internally defined while rewards are external parameters

Projective simulation model

Policy is updated after each iteration step:

$$h_{k+1}(s,a,t) = h_k(s,a,t) - \gamma_{PS}(h_k(s,a,t) - 1) + g_{k+1}(s,a,t)r(t)$$

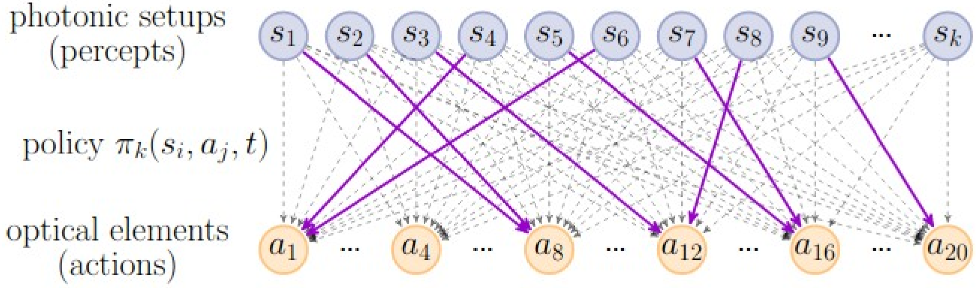
initial weights: $h_1(s,a,1) = 1$

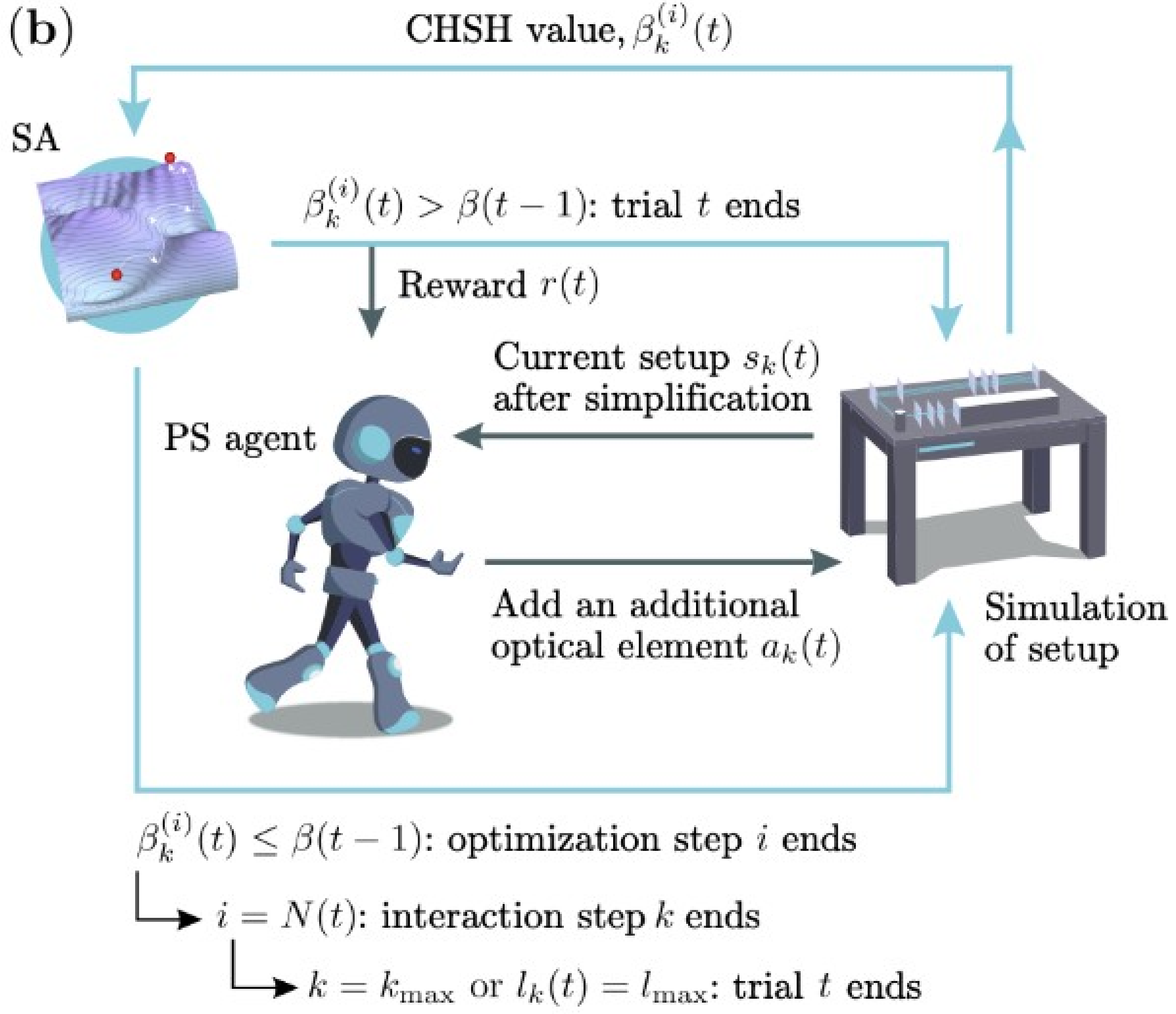
$g_{k+1}(s,a,t) = 1$ if the set(s,a)appeared in step k

otherwise $g_{k+1}(s,a,t) = (1-\eta_{PS})g_k(s,a,t), \eta_{PS} = 0.3$

$\gamma_{PS} = 10^{-3}$;dampting rate, responsible for forgetting

$$\pi_k(s,a,t) = \frac{\exp h_k(s,a,t)}{\sum_{a'} \exp h_k(s,a',t)}$$





(b)
CHSH value, $\beta_k^{(i)}(t)$
SA
$\beta_k^{(i)}(t) > \beta(t-1)$: trial t ends
Reward $r(t)$
Current setup $s_k(t)$ after simplification
PS agent
Add an additional optical element $a_k(t)$
Simulation of setup
$\beta_k^{(i)}(t) \leq \beta(t-1)$: optimization step i ends
$i = N(t)$: interaction step k ends
$k = k_{\max}$ or $l_k(t) = l_{\max}$: trial t ends

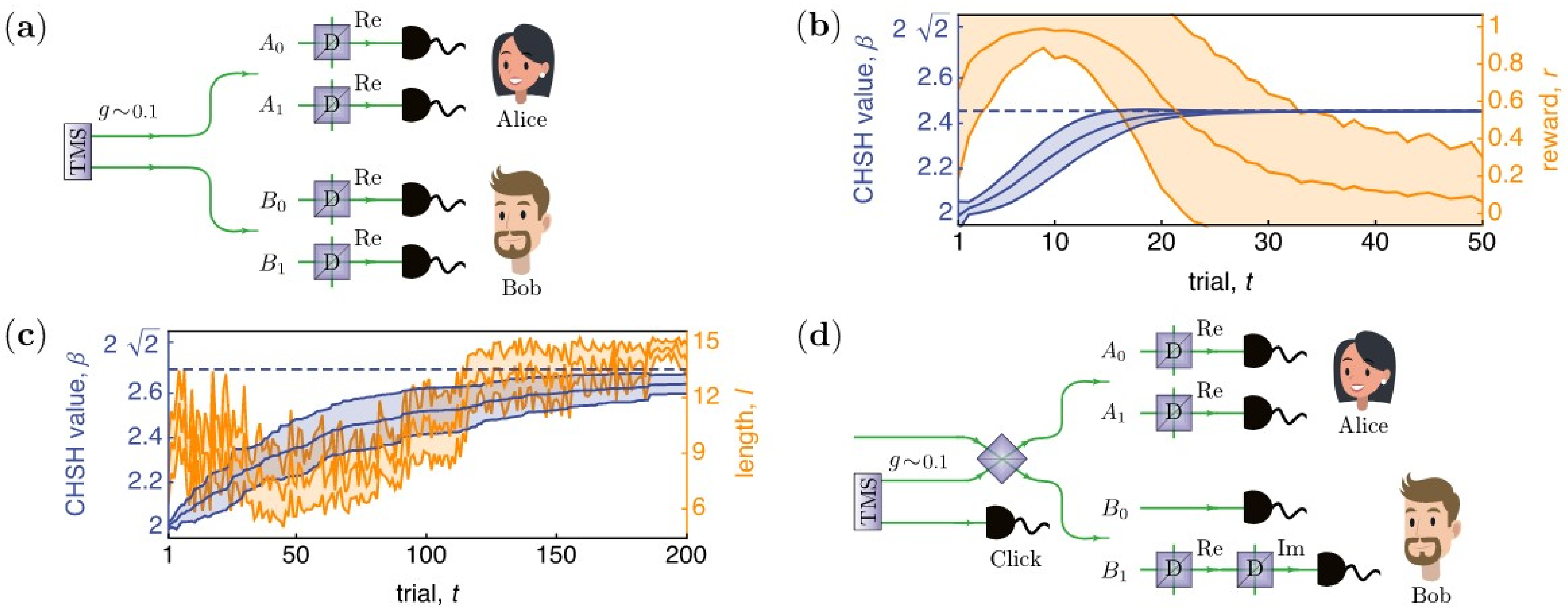
Results

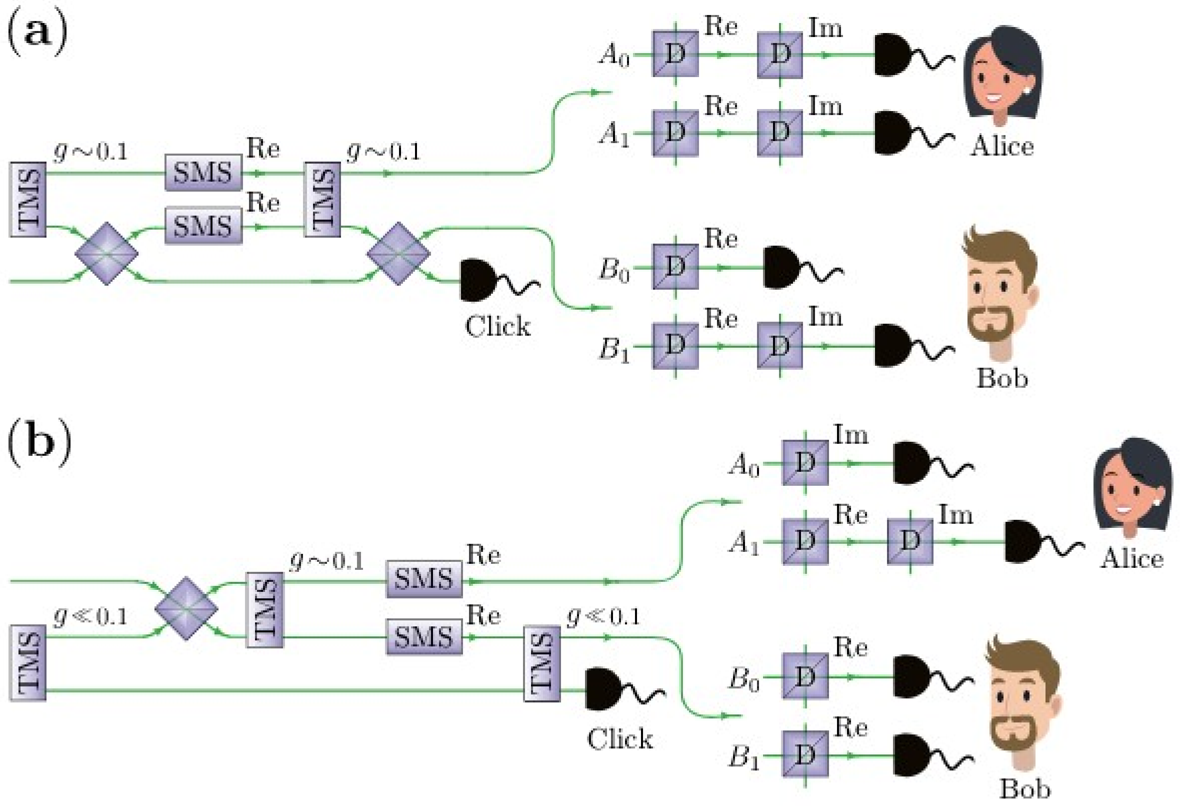
Setup	Setup description	β	p_{click}	Setup parameters
Fig. 2(a)	$\{\text{TMS}_{12}, \text{D}^{\text{Re}}_{A_0}, \text{D}^{\text{Re}}_{A_1}, \text{D}^{\text{Re}}_{B_0}, \text{D}^{\text{Re}}_{B_1}\}$	2.4546	deterministic	{0.7350, -0.1636, 0.5240, 0.1562, -0.5276}
Fig. 2(d)	$\{\text{TMS}_{23}, \text{BS}_{12}, \text{D}^{\text{Re}}_{A_0}, \text{D}^{\text{Re}}_{A_1}, \text{D}^{\text{Re}}_{B_1}, \text{D}^{\text{Im}}_{B_1}\}$	2.6401	2.2×10^{-3}	{0.0472, -0.7609, 0.2855, -0.4733, -0.0087, -0.6572}
Fig. 3(a) ★	$\{\text{TMS}_{12}, \text{BS}_{23}, \text{SMS}^{\text{Re}}_{1}, \text{SMS}^{\text{Re}}_{2}, \text{TMS}_{12}, \text{BS}_{23}, \text{D}^{\text{Re}}_{A_0}, \text{D}^{\text{Im}}_{A_0}, \text{D}^{\text{Re}}_{A_1}, \text{D}^{\text{Im}}_{A_1}, \text{D}^{\text{Re}}_{B_0}, \text{D}^{\text{Re}}_{B_1}, \text{D}^{\text{Im}}_{B_1}\}$	2.7242	2.9×10^{-4}	{-0.0855, -0.1279, -0.1247, -0.1572, 0.1047, 0.0746, -0.1896, -0.0437, 0.5477, 0.0153, -0.1704, 0.6167, 0.0157}
the same ★	the same	2.7424	2.6×10^{-5}	{-0.2465, -0.0131, -0.1212, -0.1341, 0.2585, 0.0265, -0.1608, 0.0396, 0.6276, -0.0137, -0.2018, 0.5886, -0.0315}
Fig. 3(b) ★	$\{\text{TMS}'_{23}, \text{BS}_{12}, \text{TMS}_{12}, \text{SMS}^{\text{Re}}_{1}, \text{SMS}^{\text{Re}}_{2}, \text{TMS}'_{23}, \text{D}^{\text{Im}}_{A_0}, \text{D}^{\text{Re}}_{A_1}, \text{D}^{\text{Im}}_{A_1}, \text{D}^{\text{Re}}_{B_0}, \text{D}^{\text{Re}}_{B_1}\}$	2.7454	1.1×10^{-9}	{0.3483, 0.7059, 0.0025, 0.1221, -0.1717, -0.0306, -0.1822, -0.0192, 0.6047, -0.1967, 0.6131}

★ → *designed by the agent*

Used to test the reliability of SA

Reached the maximum theoretical value w/ a difference of 10^{-4}




(a)
$g \sim 0.1$
SMS
Re
$g \sim 0.1$
TMS
SMS
Re
TMS
Click
A_0
D
Re
D
Im
A_1
D
Re
D
Im
Alice
B_0
D
Re
B_1
D
Re
D
Im
Bob
(b)
A_0
D
Im
A_1
D
Re
D
Im
Alice
$g \sim 0.1$
SMS
Re
$g \ll 0.1$
TMS
SMS
Re
$g \ll 0.1$
TMS
TMS
Click
B_0
D
Re
B_1
D
Re
Bob

<u>This presentation was based on the following article:</u>

A. Melnikov, P. Sekatski, N. Sangouard, *"Setting up experimental Bell test with reinforcement learning"*, Phys. Rev. Lett. **125**, 160401 (2020)

<u>with additional information from:</u>

H. Briegel, G. Cuevas, *Projective simulation for artificial intelligence*, Sci. Rep. 2, **400** (2012)

D. Bertsimas, J. Tsitsiklis, *Simulated annealing*, Statistical Science **8** 10-15 (1993)