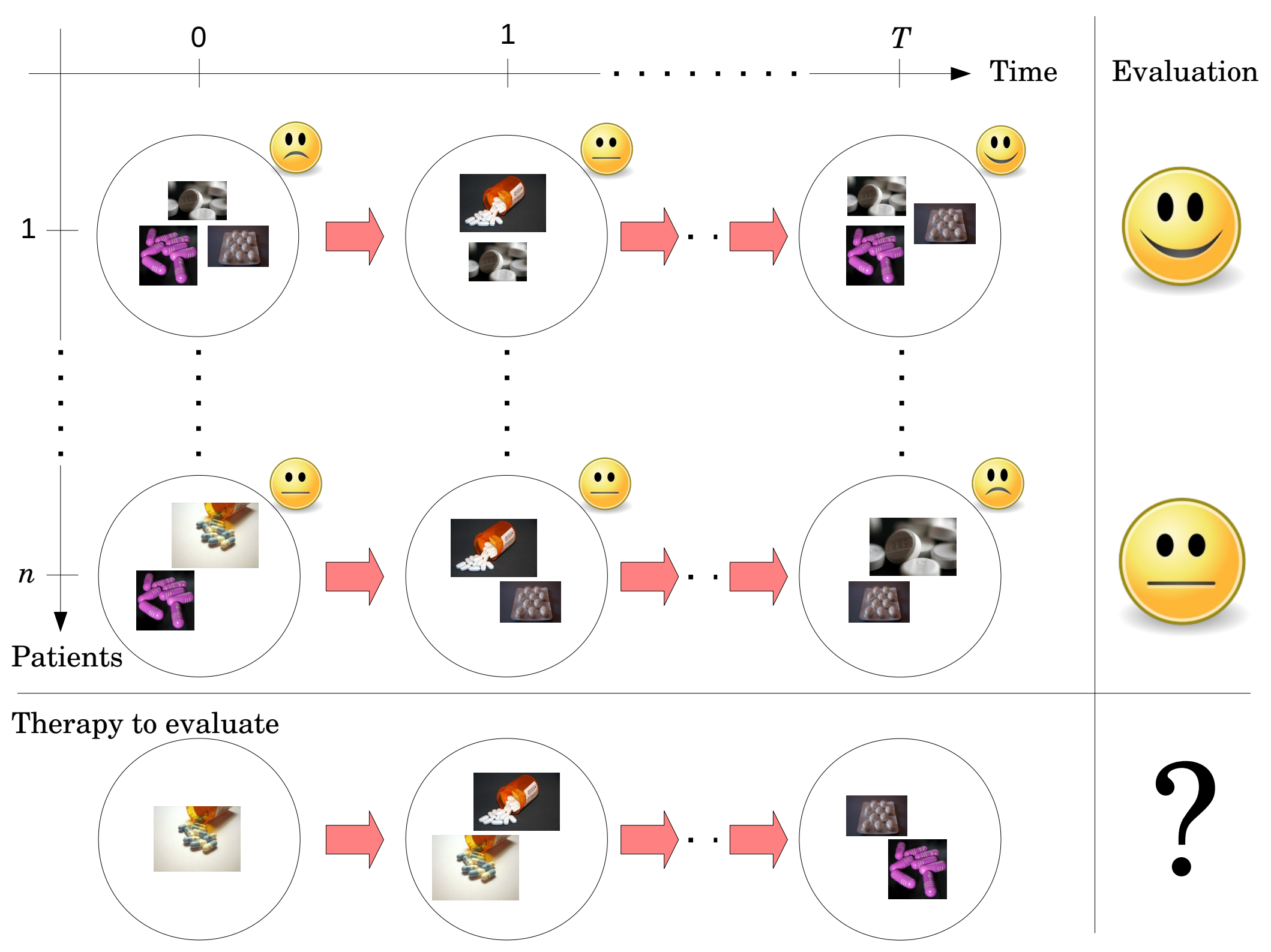
University of Liège – University of Michigan

Model-free Monte Carlo-like Policy Evaluation

Raphaël Fonteneau, Susan Murphy, Louis Wehenkel, Damien Ernst

May, 19th 2010

CAp 2010, Clermont-Ferrand, France


0
1
T
Time
Evaluation
1
n
Patients
Therapy to evaluate
?

Outline

Introduction

Problem statement

The Monte Carlo estimator

The Model-Free Monte Carlo estimator

MFMC estimator: analysis

Illustration

Conclusions and future work

Introduction

- Discrete-time stochastic optimal control problems arise in many fields (finance, medecine, engineering,...)
- Many techniques for solving such problems use an oracle that **evaluates the performance of any given policy** in order to determine a (near-)optimal control policy
- When the system is accessible to experimentation, such an oracle can be based on a **Monte Carlo** (MC) approach
- In this paper, the only information is contained in a sample of one-step transitions of the system
- In this context, we propose a **Model-Free Monte Carlo** (MFMC) estimator of the performance of a given policy that mimics in some way the Monte Carlo estimator.

Problem statement

- We consider a discrete-time system whose dynamics over T stages is given by

$$x_{t+1} = f(x_t, u_t, w_t)$$

- All x_t lie in a normed state space X, all u_t lie in a normed action space U, w_t are i.i.d. according to a probability distribution $p_W(.)$
- An instantaneous reward $r_t = \rho(x_t, u_t, w_t)$ is associated with the action u_t while being in state x_t
- A policy h: $\{0,...,T\text{-}1\} \times X \to U$ is given, and we want to **evaluate its performance**.

Problem statement

- The **expected return** of the policy h when starting from an initial state x_0 is given by

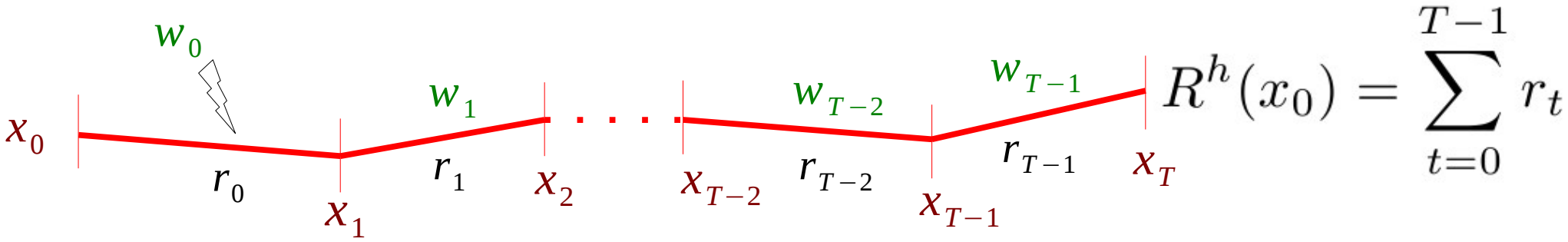
$$J^h(x_0) = \mathop{\mathbb{E}}_{w_0,\ldots,w_{T-1}\sim p_{\mathcal{W}}(.)}\left[R^h(x_0)\right]$$

where

$$R^h(x_0) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t), w_t)$$

with

$$x_{t+1} = f(x_t, h(t, x_t), w_t)$$



Problem statement

- **Problem**: the functions f, ρ and $p_W(.)$ are **unknown**
- They are replaced by a sample of system transitions

$$\mathcal{F}_n = [(x^l, u^l, r^l, y^l)]_{l=1}^n$$

where the pairs (x^l, u^l) are arbitrary chosen and the pairs (r^l, y^l) are determined by $(f(x^l, u^l, w^l), \rho(x^l, u^l, w^l))$, where w^l is drawn according to $p_W(.)$

How to evaluate $J^h(x_0)$ in this context?

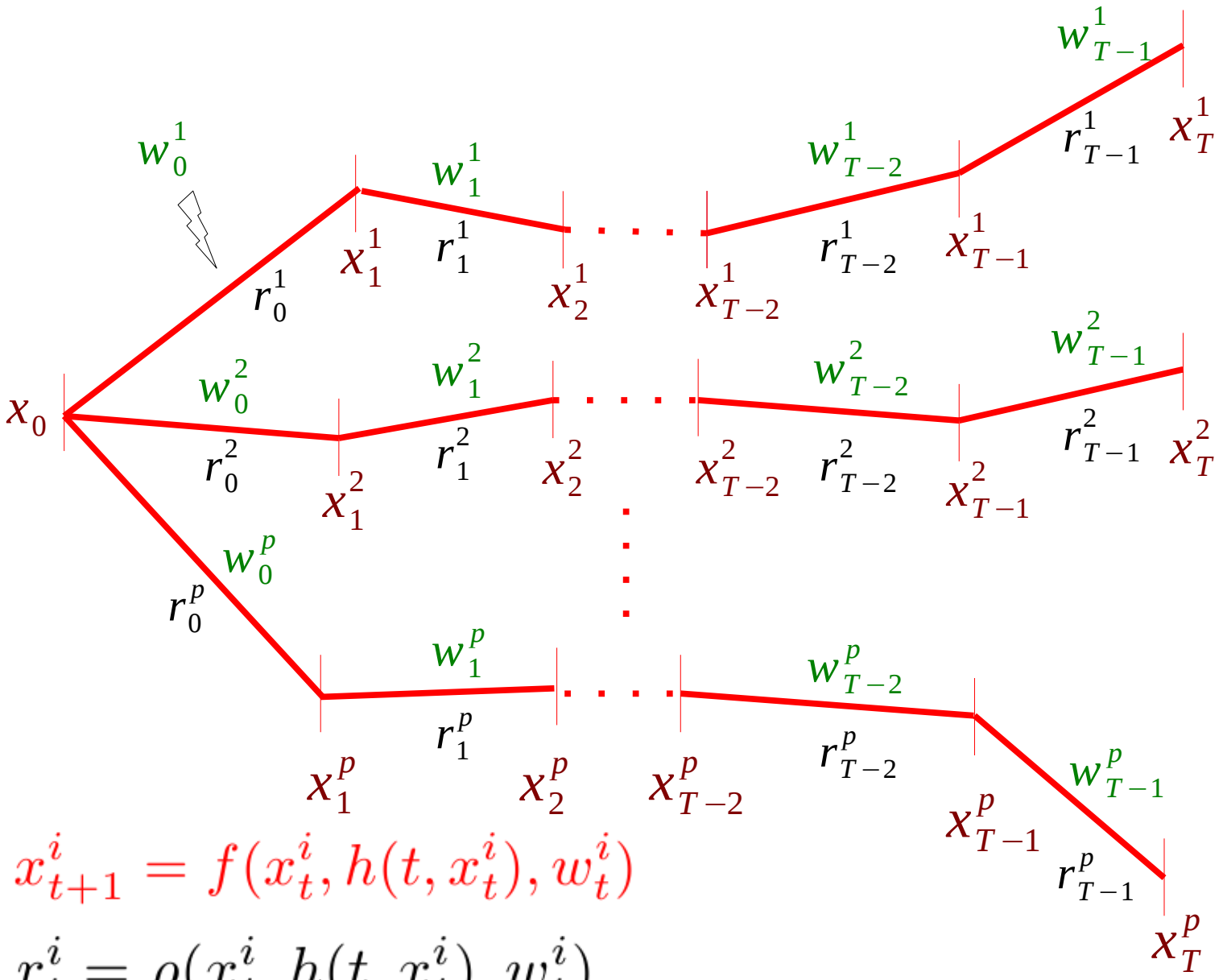
The Monte Carlo estimator

- We define the **Monte Carlo estimator** of the expected return of h when starting from the initial state x_0:

$$\mathbb{M}_p^h(x_0) = \frac{1}{p} \sum_{i=1}^{p} \sum_{t=0}^{T-1} \rho(x_t^i, h(t, x_t^i), w_t^i)$$

with $$\forall t \in [\![0, T-1]\!], \forall i \in [\![1, p]\!] :$$
$$w_t^i \sim p_{\mathcal{W}}(.), x_0^i = x_0 \ , x_{t+1}^i = f(x_t^i, h(t, x_t^i), w_t^i)$$

The Monte Carlo estimator



$$x_{t+1}^i = f(x_t^i, h(t, x_t^i), w_t^i)$$

$$r_t^i = \rho(x_t^i, h(t, x_t^i), w_t^i)$$

$$w_t^i \sim p_W(.)$$

$$\sum_{t=0}^{T-1} r_t^1 \qquad \sum_{t=0}^{T-1} r_t^2 \qquad \sum_{t=0}^{T-1} r_t^p$$

MC Estimator

$$\frac{1}{p} \sum_{i=1}^{p} \sum_{t=0}^{T-1} r_t^i$$

The Monte Carlo estimator

- We assume that the random variable $R^h(x_0)$ admits a finite variance

$$\sigma^2_{R^h}(x_0) = \underset{w_0,\ldots,w_{T-1}\sim p_{\mathcal{W}}(.)}{Var}\left[R^h(x_0)\right]$$

- The **bias** and **variance** of the Monte Carlo estimator are

$$\underset{w_t^i\sim p_{\mathcal{W}}(.),i=1\ldots p,t=0\ldots T-1}{\mathbb{E}}\left[\mathbb{M}_p^h(x_0) - J^h(x_0)\right] = 0$$

$$\underset{w_t^i\sim p_{\mathcal{W}}(.),i=1\ldots p,t=0\ldots T-1}{Var}\left[\mathbb{M}_p^h(x_0)\right] = \frac{\sigma^2_{R^h}(x_0)}{p}$$

The Model-free Monte Carlo estimator

- Here, the MC approach is not feasible, since the system is unknown
- We introduce the **Model-Free Monte Carlo estimator**
- From the sample of transitions, we build p sequences of **different** transitions of length T called ***``broken trajectories''***
- These broken trajectories are built so as to minimize the discrepancy (using a distance metric Δ) with a classical MC sample that could be obtained by simulating the system with the policy h
- We average the cumulated returns over the p broken trajectories to compute an estimate of the expected return of h
- The algorithm has complexity $O(npT)$.

The Model-free Monte Carlo estimator

MFMC sampling *(arguments :* $\mathcal{F}_n, h(.,.), x_0, \Delta(.,.), T, p$*)*

Let $\mathcal{G}$ denote the current set of not yet used one-step transitions in $\mathcal{F}_n$;

Initially, set $\mathcal{G} = \mathcal{F}_n$;

For $i = 1$ to p, extract a broken trajectory by doing :

Set $t = 0$ and $x_t^i = x_0$;

While $t < T$ do

Set $u_t^i = h(t, x_t^i)$;

Compute the set $\mathcal{H} = \underset{(x,u,r,y)\in\mathcal{G}}{\arg\min} \left(\Delta((x,u),(x_t^i,u_t^i))\right)$;

Let l_t^i be the lowest index in $\mathcal{F}_n$ of the transitions that belong to $\mathcal{H}$;

Set $t = t + 1$, $x_t^i = y^{l_t^i}$;

Set $\mathcal{G} = \mathcal{G} \setminus \{(x^{l_t^i}, u^{l_t^i}, r^{l_t^i}, y^{l_t^i})\}$;

end **While**

end **For**

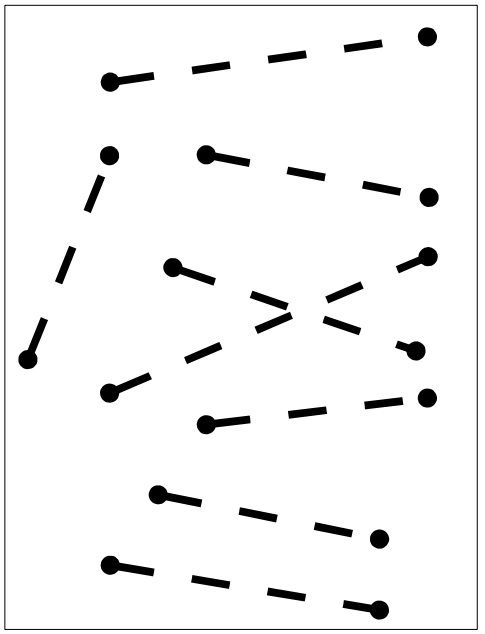
Return the set of indices $\{l_t^i\}_{i=1,t=0}^{i=p,t=T-1}$.

The Model-free Monte Carlo estimator

How does it work ?

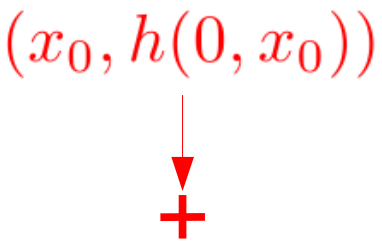
Example with T=3, p=2, n=8

$$\mathcal{F}_n = [(x^l, u^l, r^l, y^l)]_{l=1}^{n}$$



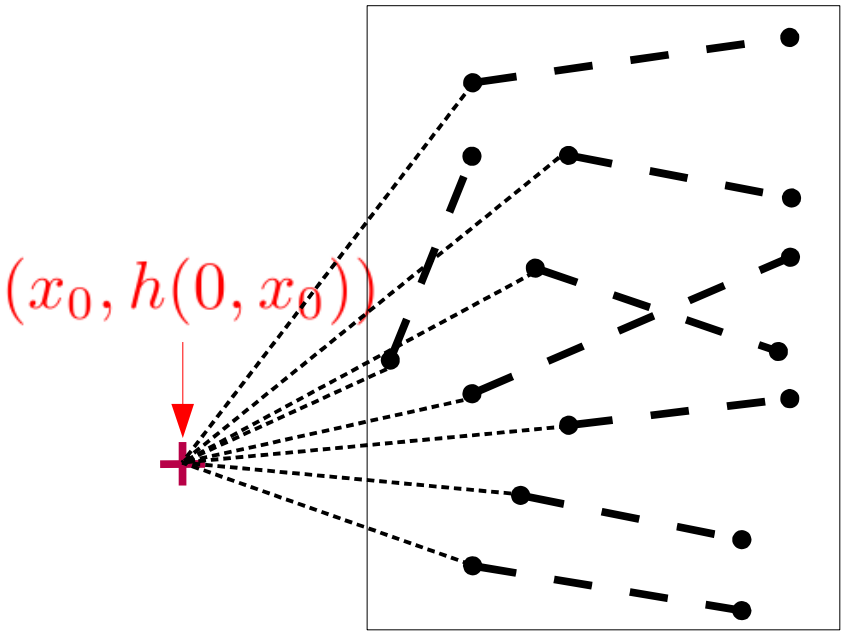
The Model-free Monte Carlo estimator

How does it work ?



The Model-free Monte Carlo estimator

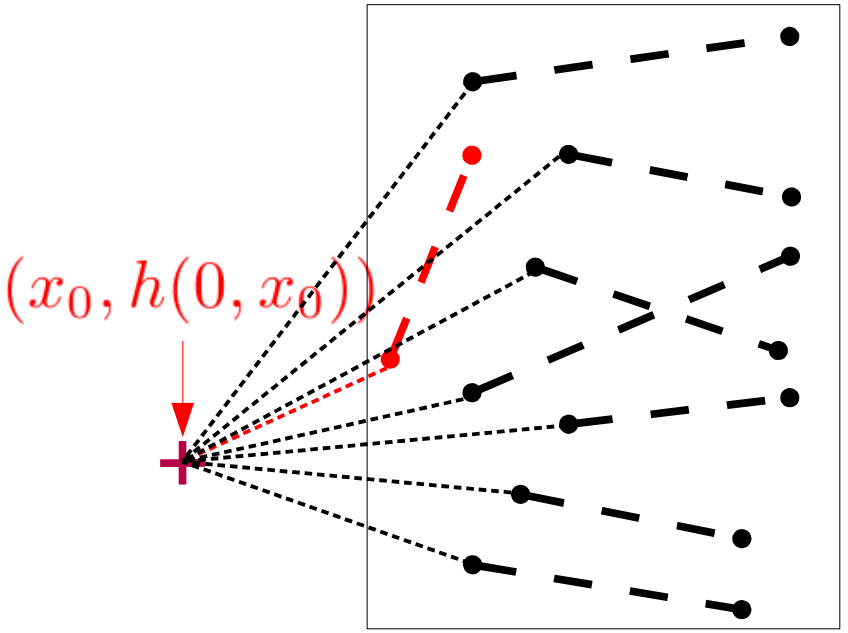
How does it work ?



$$\mathcal{G} = \mathcal{F}_n$$

The Model-free Monte Carlo estimator

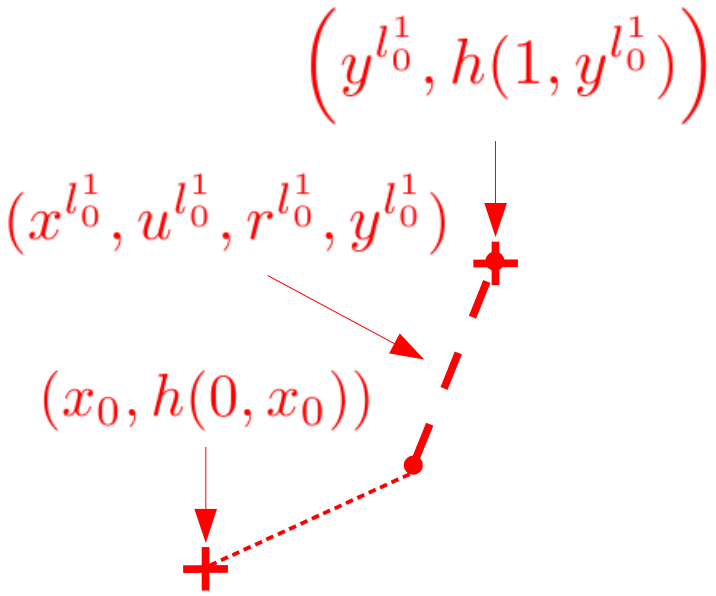
How does it work ?



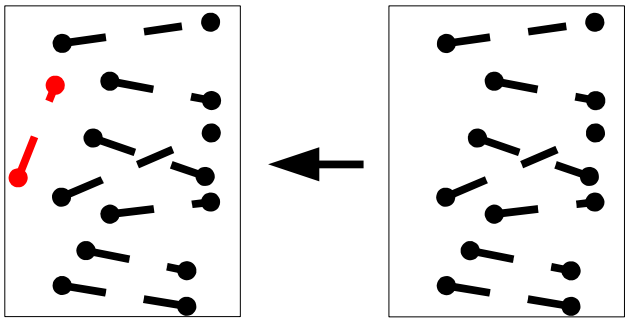
$$\mathcal{G} = \mathcal{F}_n$$

The Model-free Monte Carlo estimator

How does it work ?

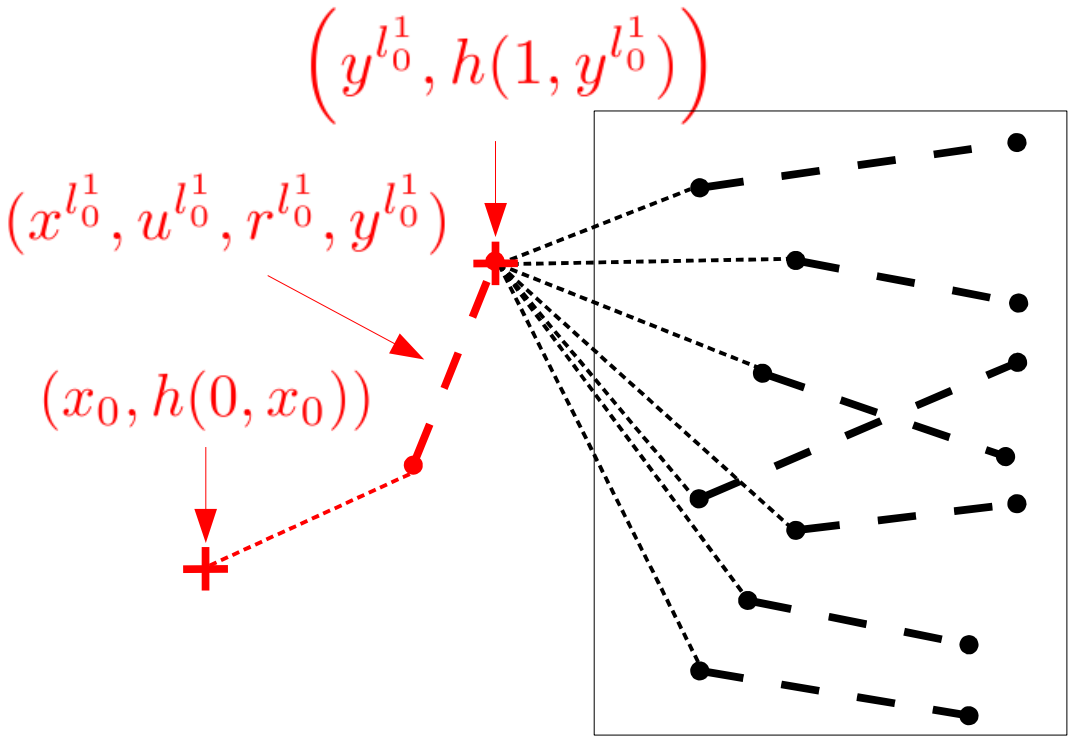


$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_0^1}, u^{l_0^1}, r^{l_0^1}, y^{l_0^1})\}$$



The Model-free Monte Carlo estimator

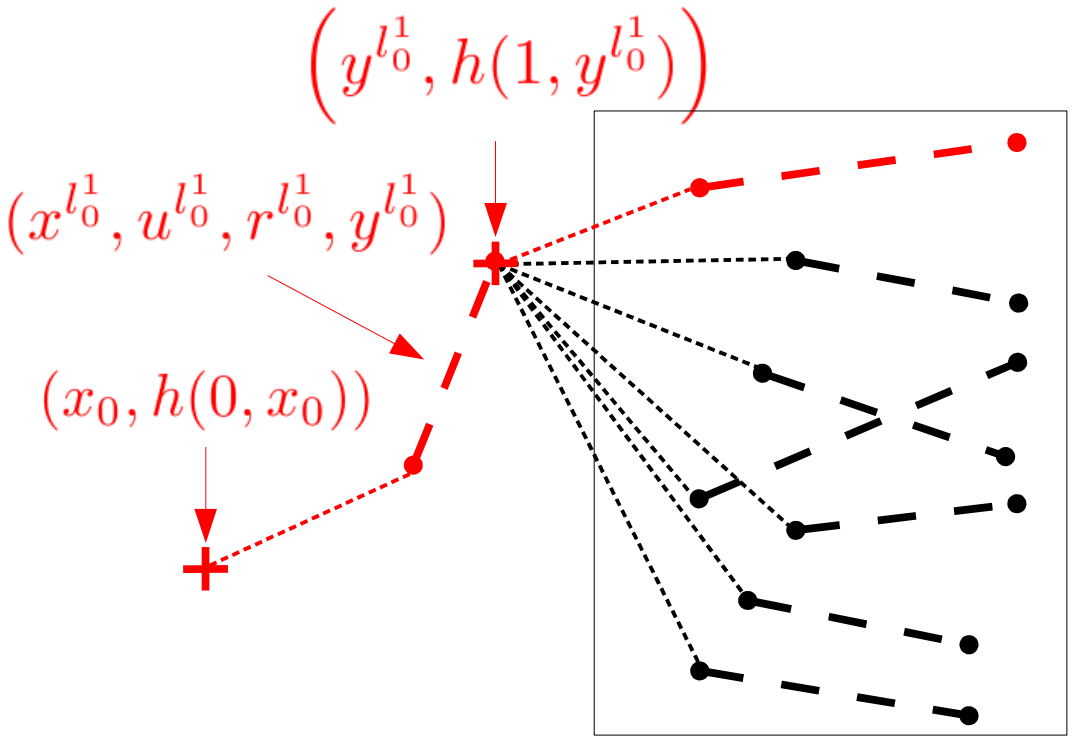
How does it work ?



$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_0^1}, u^{l_0^1}, r^{l_0^1}, y^{l_0^1})\}$$

The Model-free Monte Carlo estimator

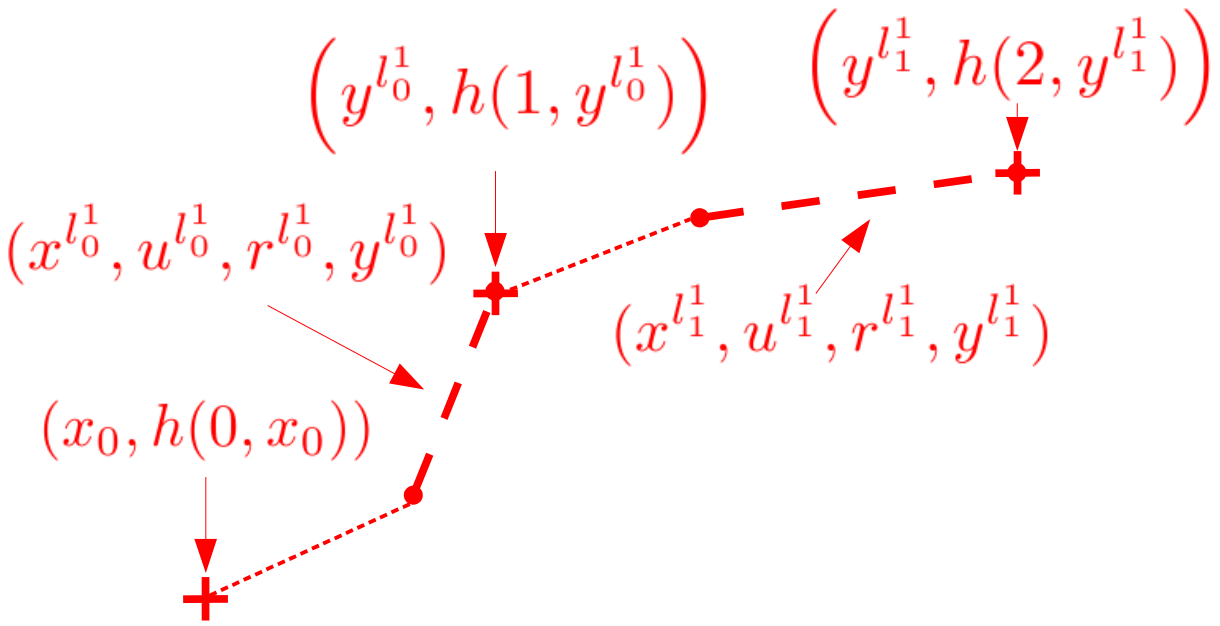
How does it work ?



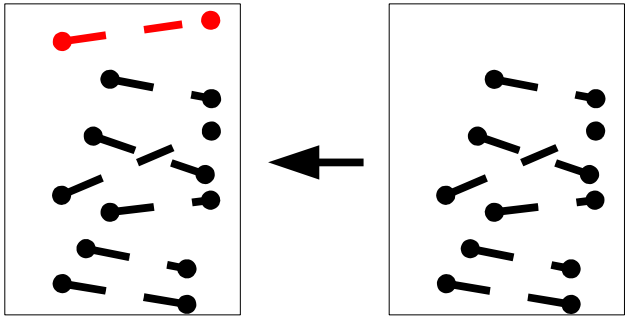
$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_0^1}, u^{l_0^1}, r^{l_0^1}, y^{l_0^1})\}$$

The Model-free Monte Carlo estimator

How does it work ?

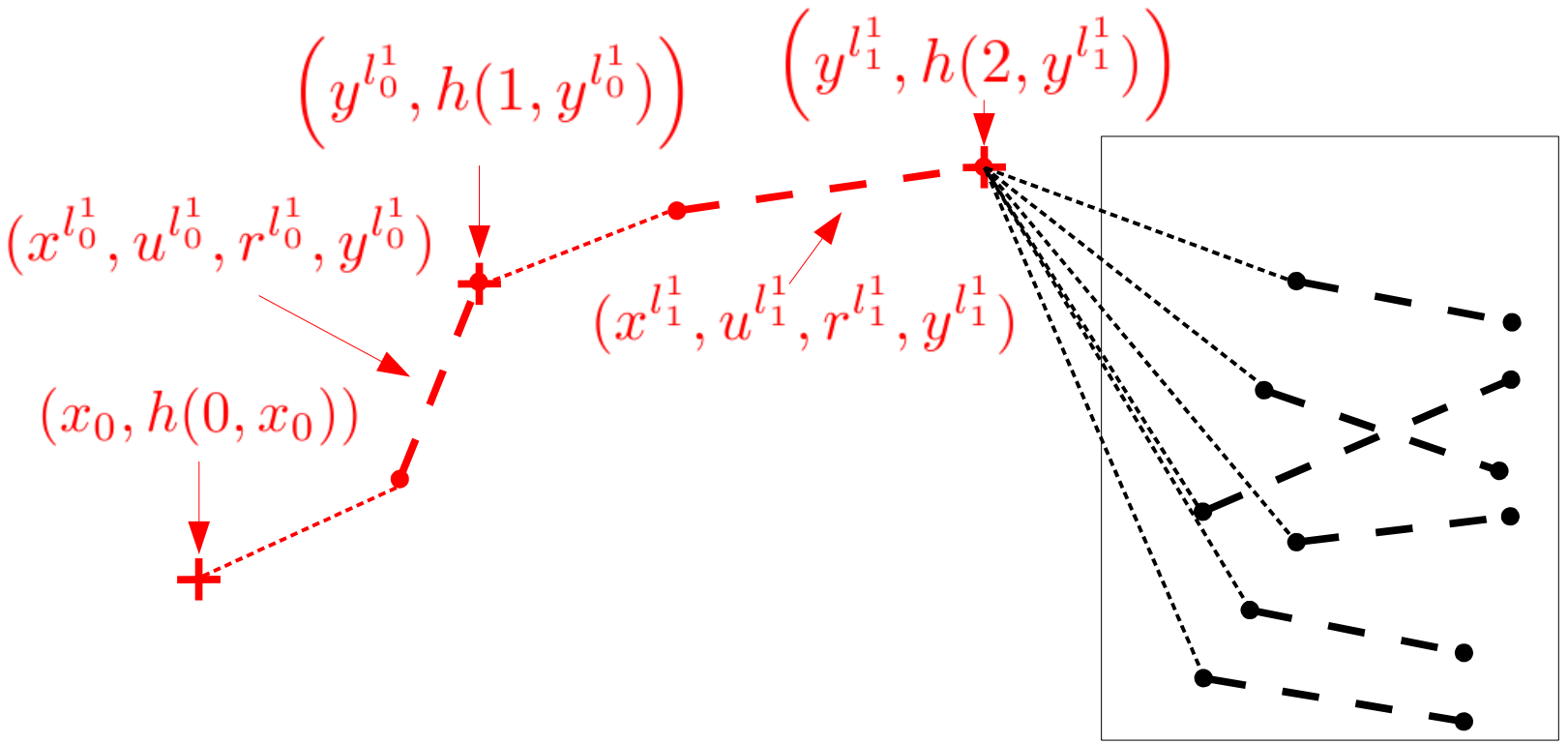


$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_1^1}, u^{l_1^1}, r^{l_1^1}, y^{l_1^1})\}$$



The Model-free Monte Carlo estimator

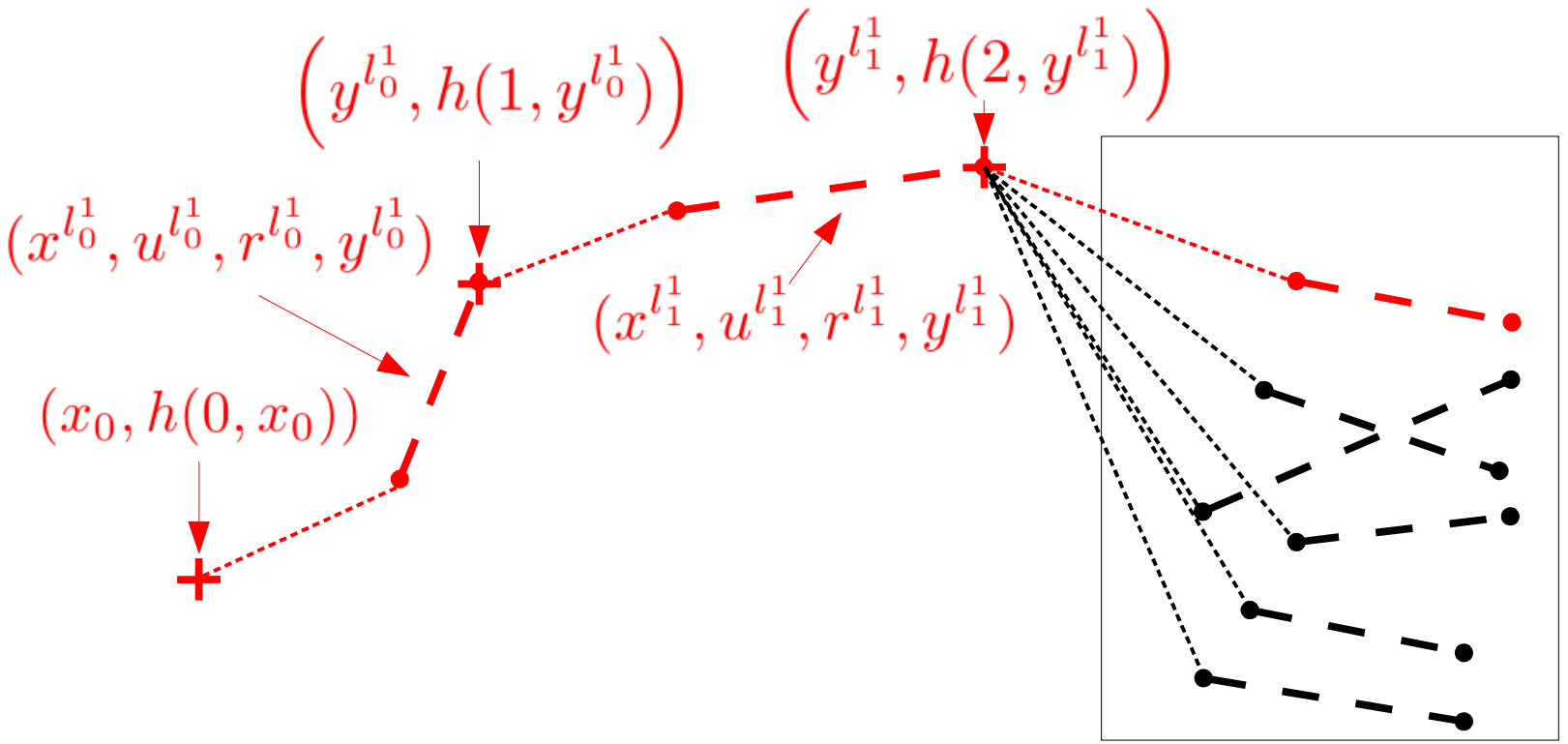
How does it work ?



$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_1^1}, u^{l_1^1}, r^{l_1^1}, y^{l_1^1})\}$$

The Model-free Monte Carlo estimator

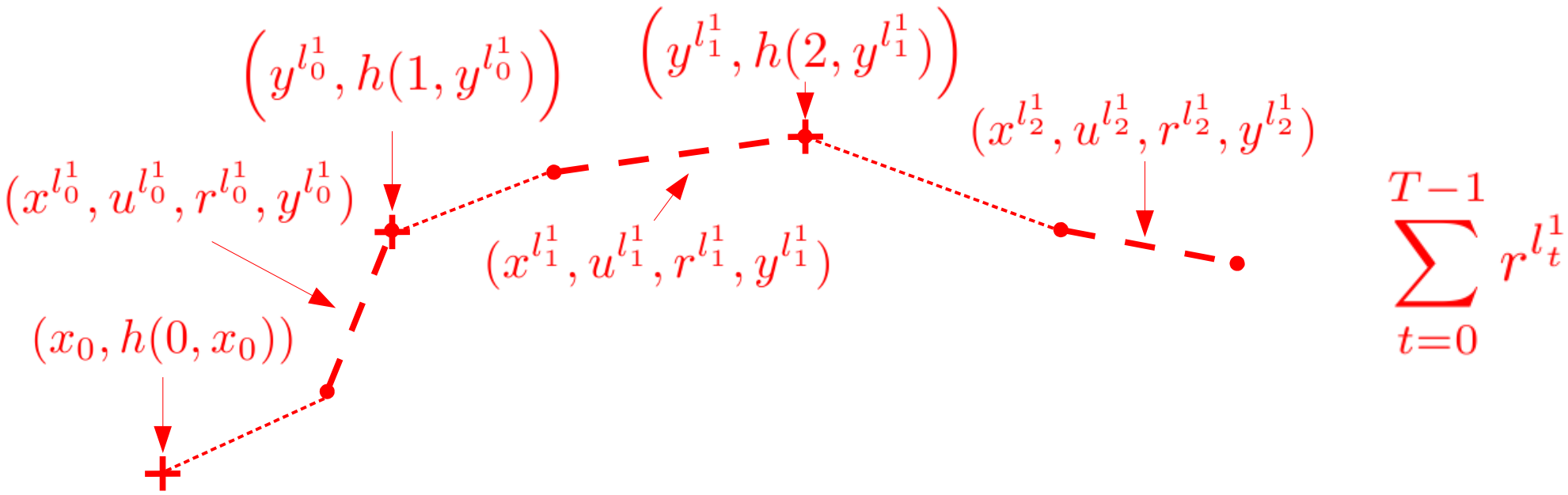
How does it work ?



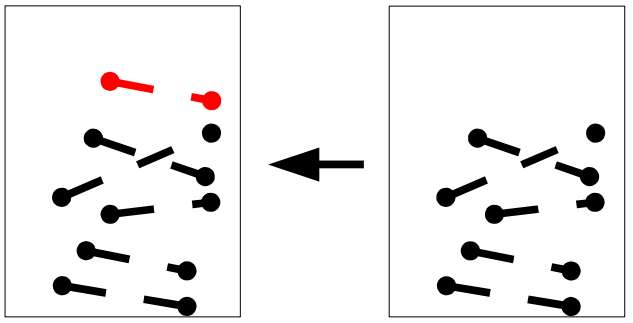
$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_1^1}, u^{l_1^1}, r^{l_1^1}, y^{l_1^1})\}$$

The Model-free Monte Carlo estimator

How does it work ?

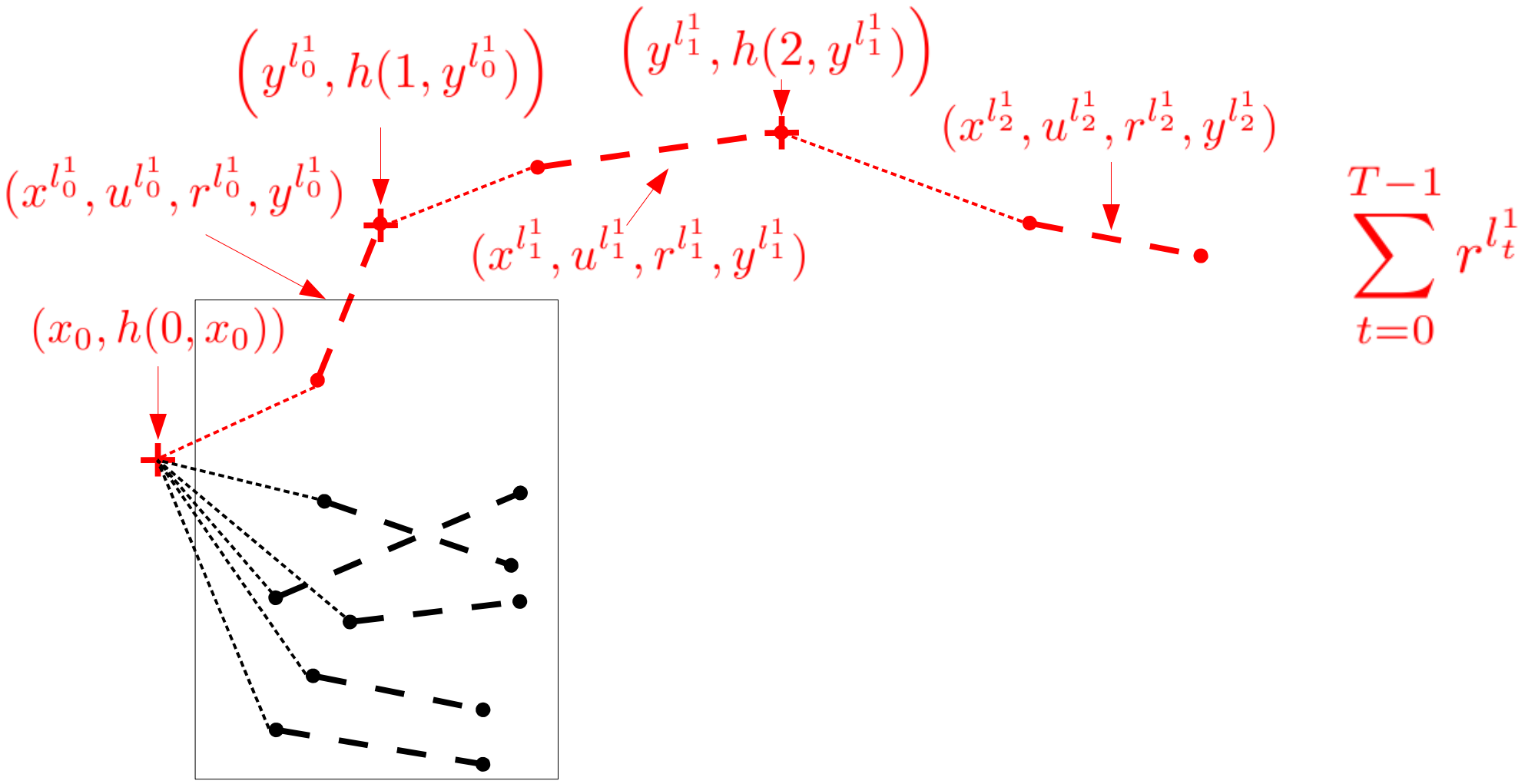


$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_2^1}, u^{l_2^1}, r^{l_2^1}, y^{l_2^1})\}$$



The Model-free Monte Carlo estimator

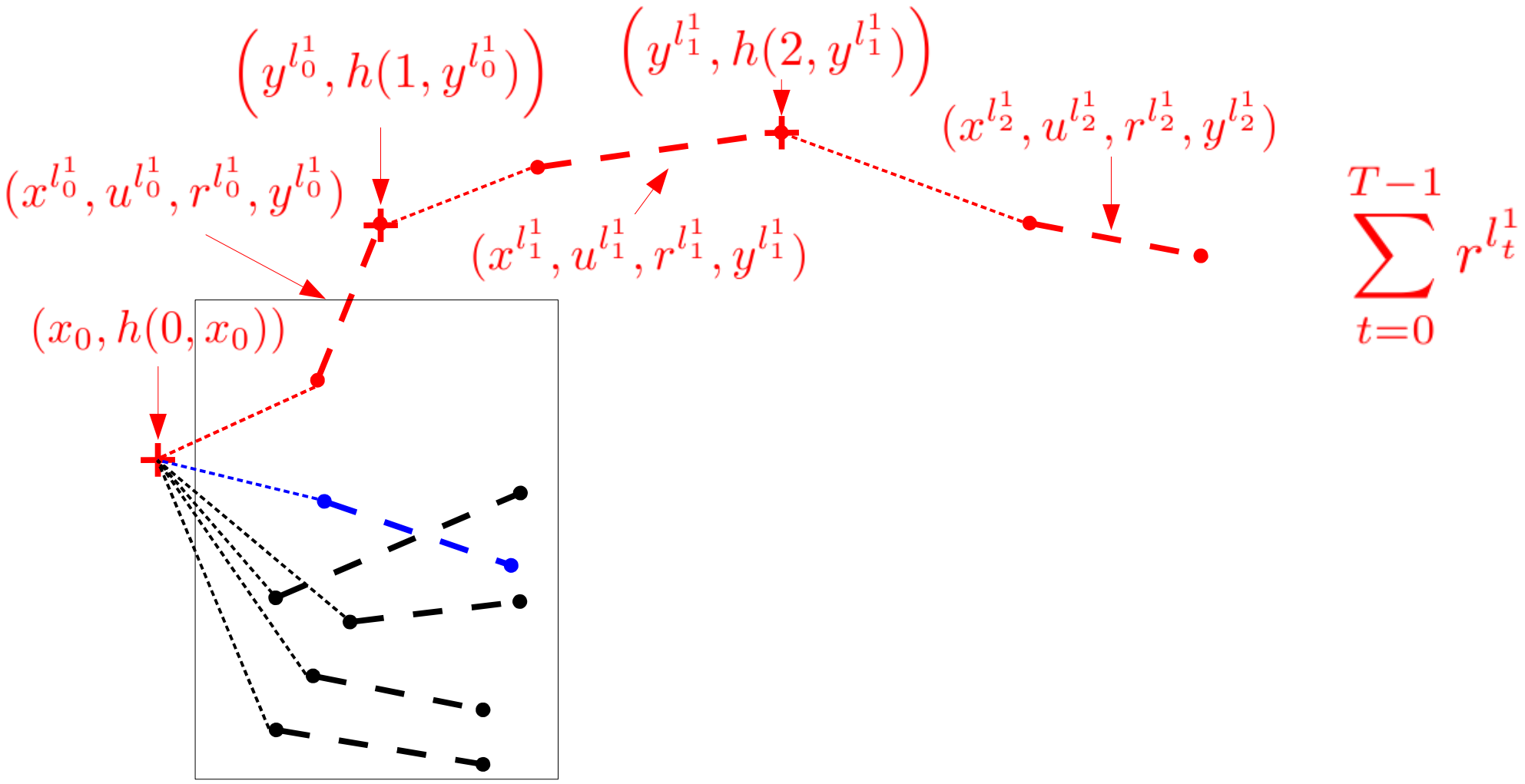
How does it work ?



$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_2^1}, u^{l_2^1}, r^{l_2^1}, y^{l_2^1})\}$$

The Model-free Monte Carlo estimator

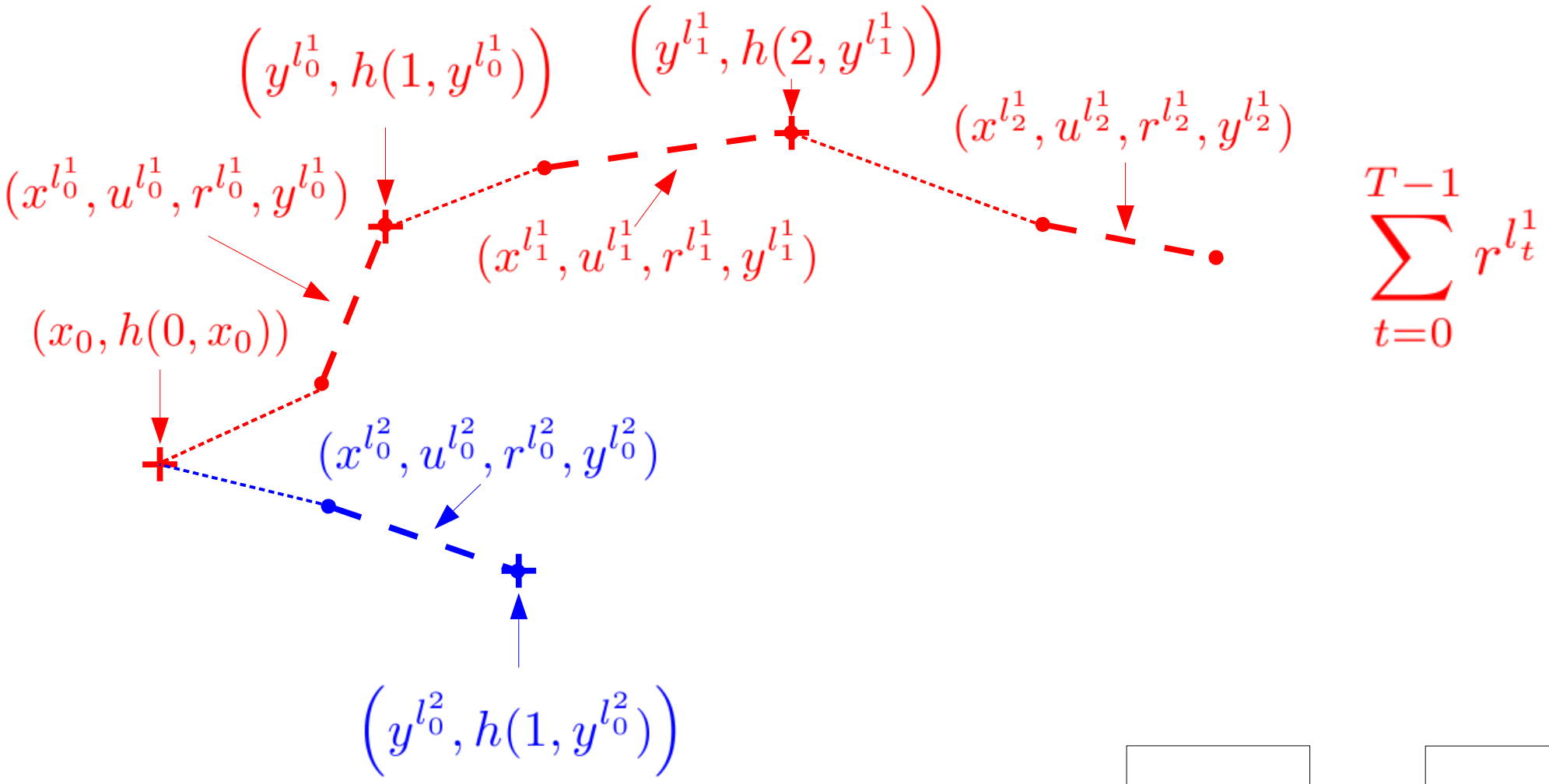
How does it work ?



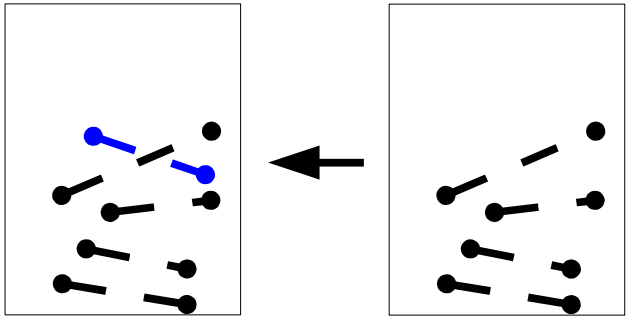
$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_2^1}, u^{l_2^1}, r^{l_2^1}, y^{l_2^1})\}$$

The Model-free Monte Carlo estimator

How does it work ?

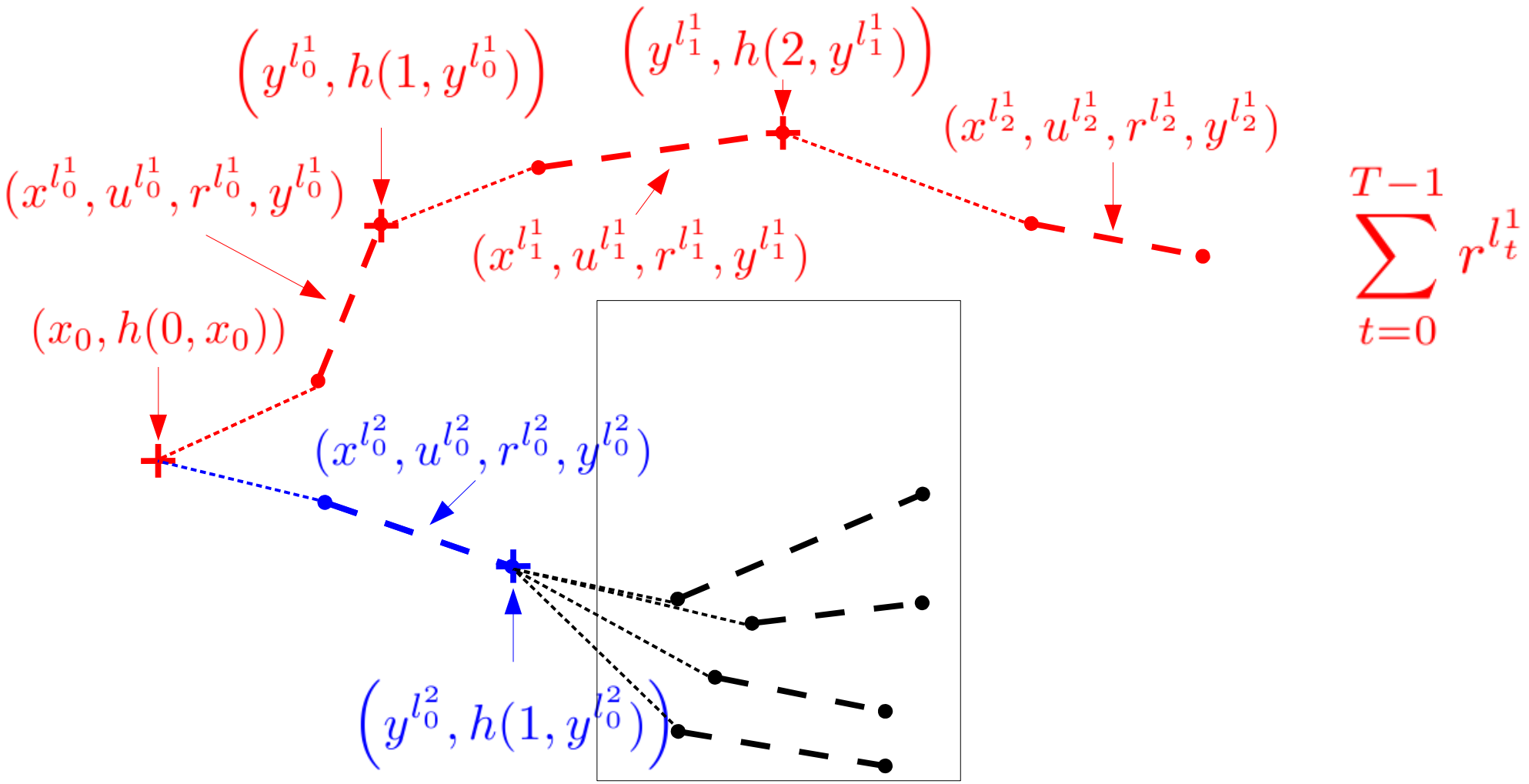


$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_0^2}, u^{l_0^2}, r^{l_0^2}, y^{l_0^2})\}$$



The Model-free Monte Carlo estimator

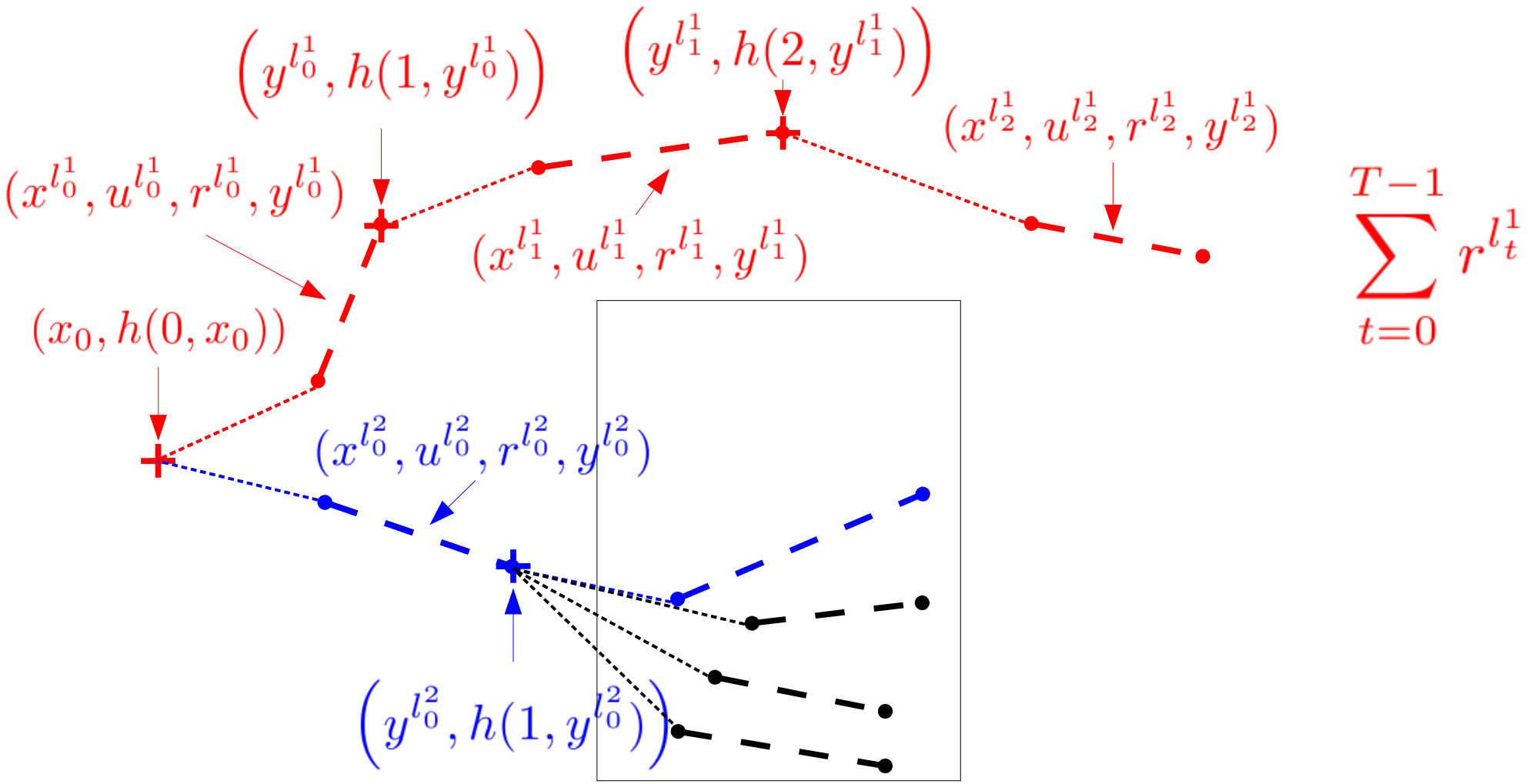
How does it work ?



$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_0^2}, u^{l_0^2}, r^{l_0^2}, y^{l_0^2})\}$$

The Model-free Monte Carlo estimator

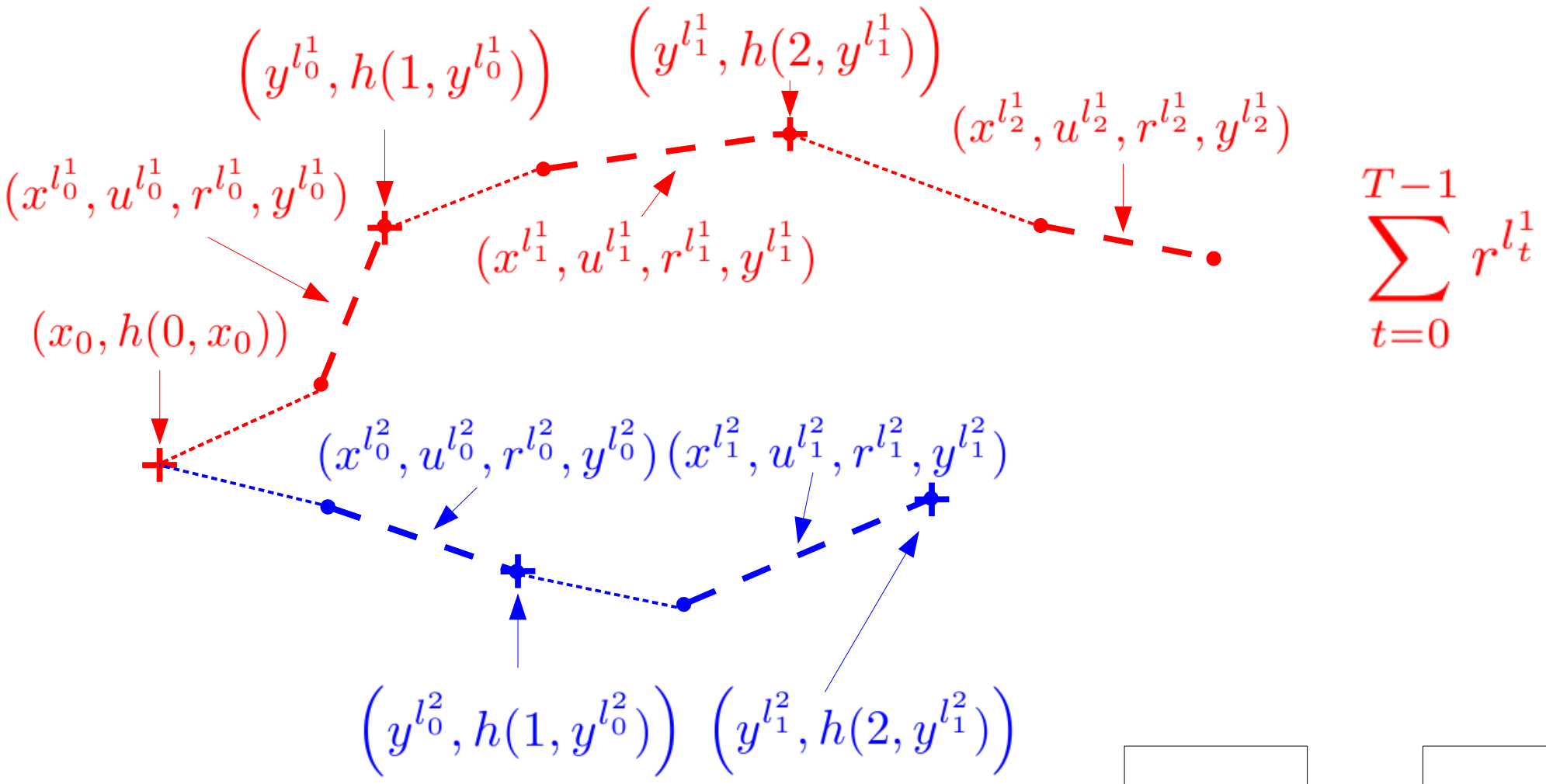
How does it work ?



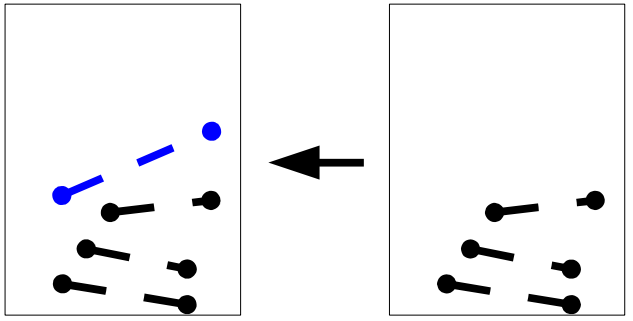
$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_0^2}, u^{l_0^2}, r^{l_0^2}, y^{l_0^2})\}$$

The Model-free Monte Carlo estimator

How does it work ?

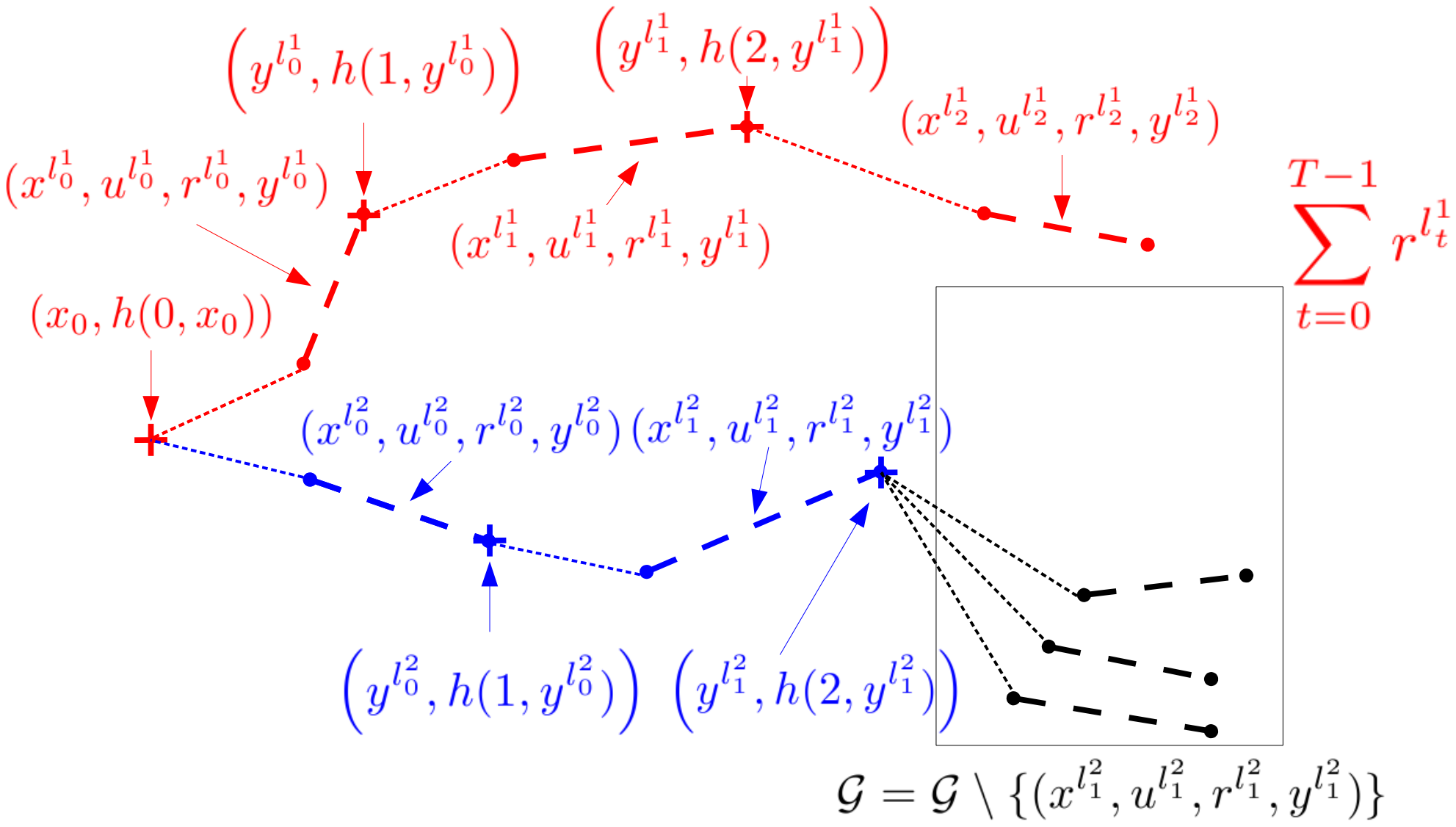


$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_1^2}, u^{l_1^2}, r^{l_1^2}, y^{l_1^2})\}$$



The Model-free Monte Carlo estimator

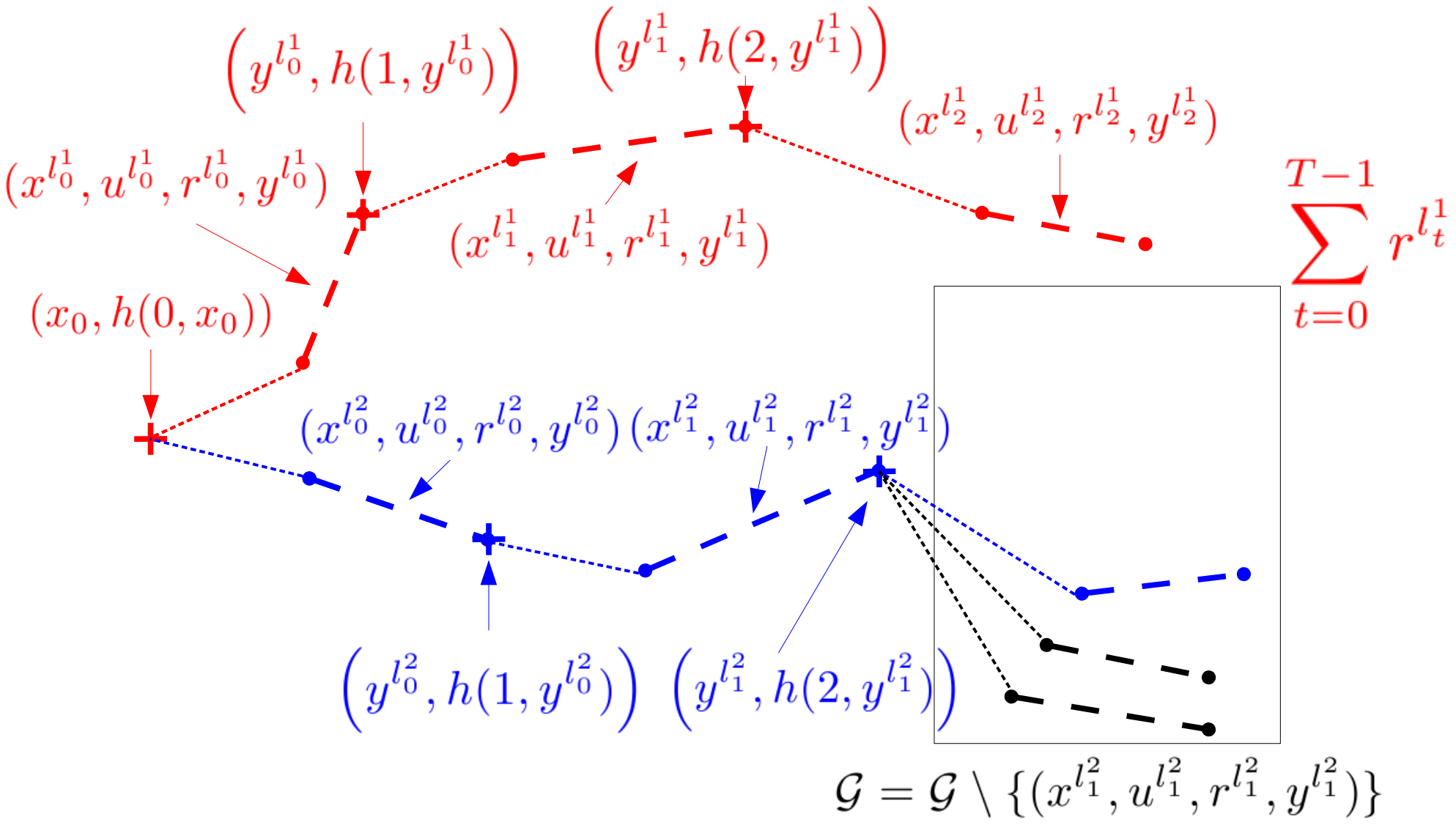
How does it work ?



$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_1^2}, u^{l_1^2}, r^{l_1^2}, y^{l_1^2})\}$$

The Model-free Monte Carlo estimator

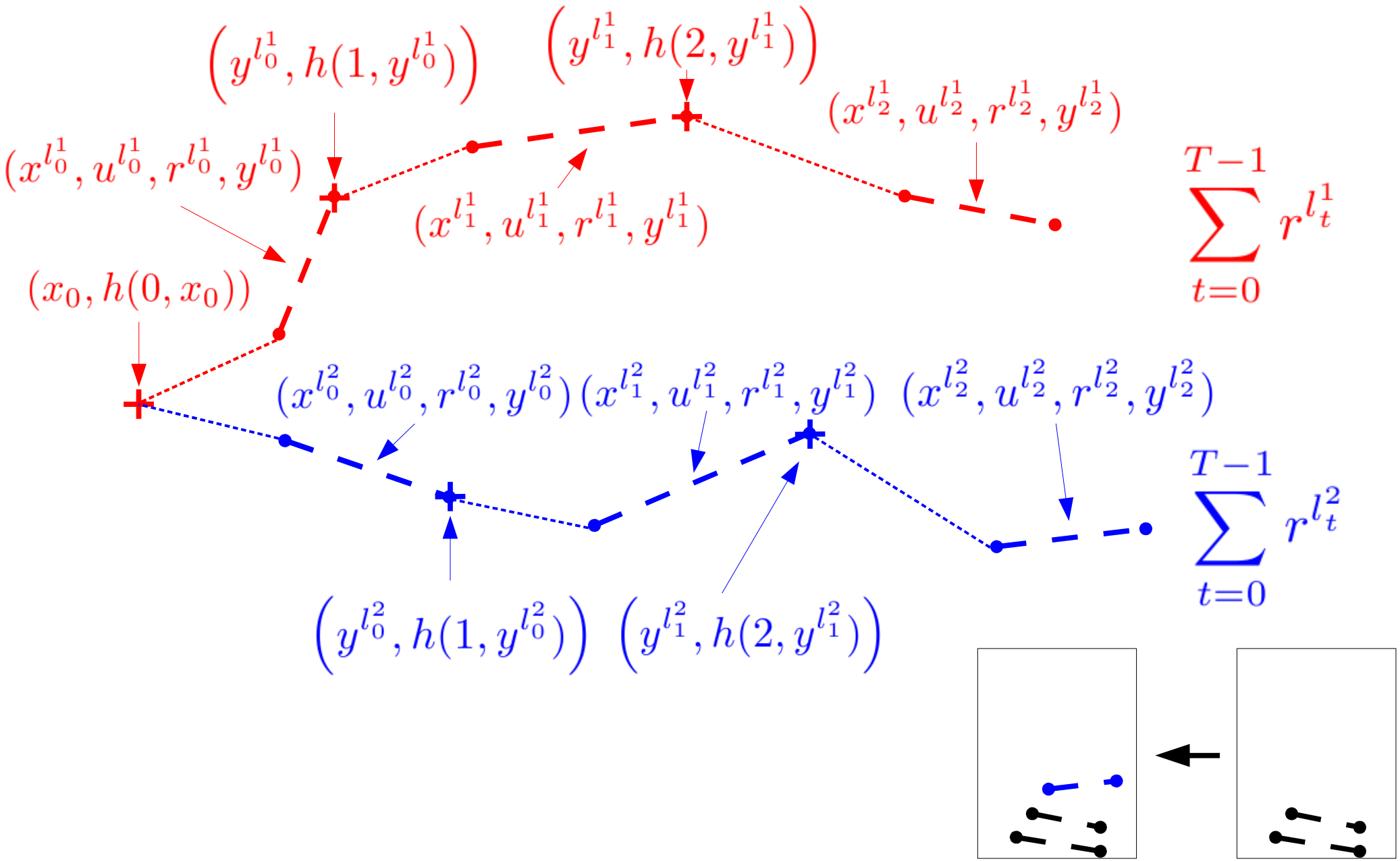
How does it work ?



$$\mathcal{G} = \mathcal{G} \setminus \{(x^{l_1^2}, u^{l_1^2}, r^{l_1^2}, y^{l_1^2})\}$$

The Model-free Monte Carlo estimator

How does it work ?



MFMC estimator: analysis

- **Assumption**: the functions f, ρ and h are **Lipschitz continuous**

$$\exists L_f, L_\rho, L_h \in \mathbb{R}^+ : \forall\, (x, x', u, u', w) \in \mathcal{X}^2 \times \mathcal{U}^2 \times \mathcal{W},$$

$$\|f(x, u, w) - f(x', u', w)\|_{\mathcal{X}} \leq L_f(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}),$$

$$|\rho(x, u, w) - \rho(x', u', w)| \leq L_\rho(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}),$$

$$\forall t \in [\![0, T-1]\!], \|h(t, x) - h(t, x')\|_{\mathcal{U}} \leq L_h \|x - x'\|_{\mathcal{X}}$$

MFMC estimator: analysis

- The only information available on the system is gathered in a sample of n one-step transitions

$$\mathcal{F}_n = [(x^l, u^l, r^l, y^l)]_{l=1}^n$$

- We define the random variable $\tilde{\mathcal{F}}_n$ as follows:

  The set of pairs $\mathcal{P}_n = [(x^l, u^l)]_{l=1}^n$ is arbitrary chosen,

  whereas the pairs (r^l, y^l) are determined by $\left(f(x^l, u^l, w^l), \rho(x^l, u^l, w^l)\right)$ where w^l is drawn according to $p_W(.)$

- $\mathcal{F}_n$ is a **realization** of the random set $\tilde{\mathcal{F}}_n$.

MFMC estimator: analysis

- **Distance metric Δ**

$$\forall (x, x', u, u') \in \mathcal{X}^2 \times \mathcal{U}^2,$$
$$\Delta((x,u),(x',u')) = (\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}})$$

- ***k*-sparsity**

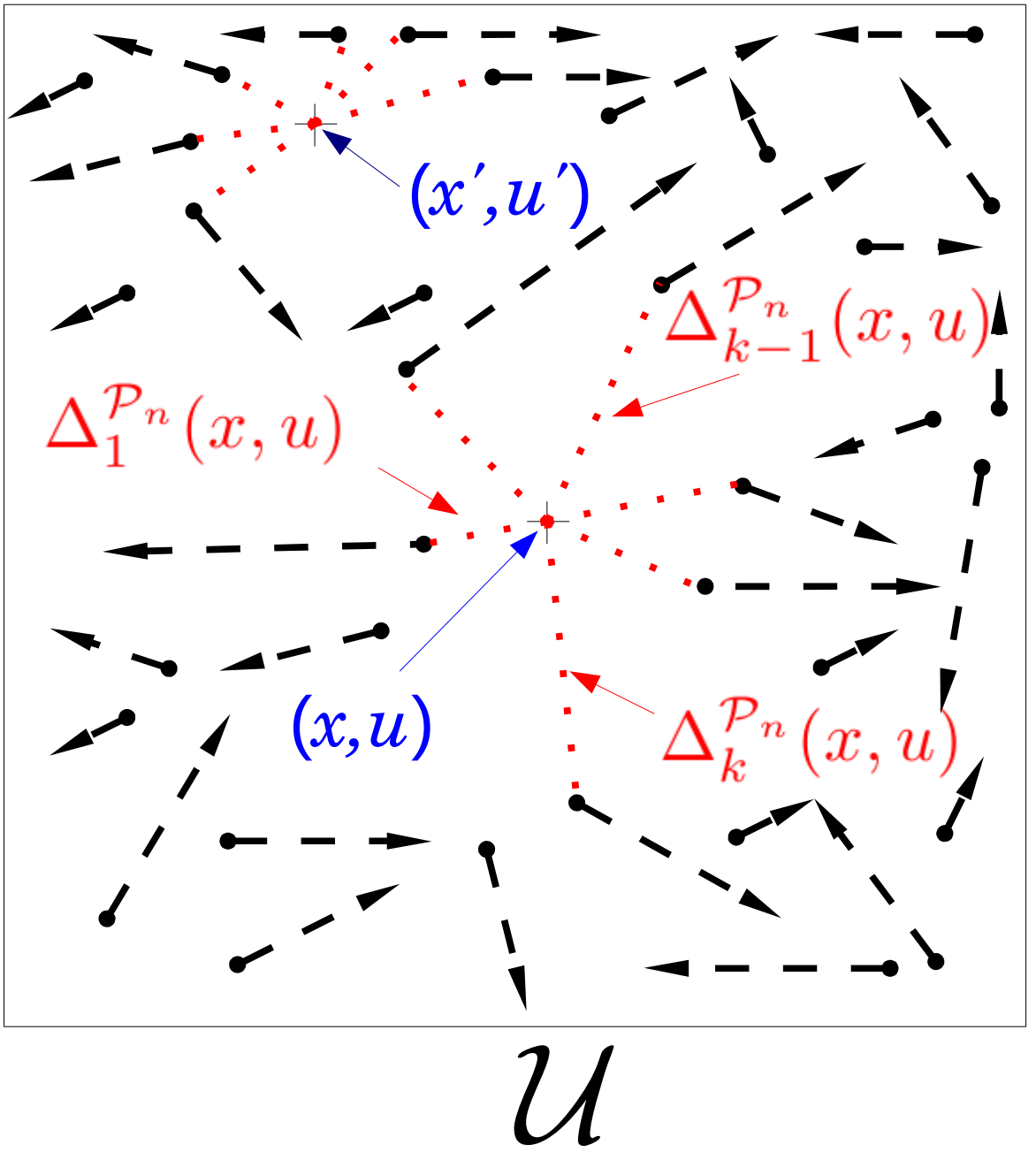
$$\alpha_k(\mathcal{P}_n) = \sup_{(x,u)\in\mathcal{X}\times\mathcal{U}} \left\{\Delta_k^{\mathcal{P}_n}(x,u)\right\}$$

- $\Delta_k^{\mathcal{P}_n}(x,u)$ denotes the distance of (x,u) to its k-th nearest neighbor (using the distance Δ) in the sample $\mathcal{P}_n = [(x^l, u^l)]_{l=1}^n$

MFMC estimator: analysis

The k-sparsity can be seen as the smallest radius such that all Δ-balls in $X \times U$ contain at least k elements from

$$\mathcal{P}_n = [(x^l, u^l)]_{l=1}^n$$



MFMC estimator: analysis

- **Expected value** of the MFMC estimator

$$E^h_{p,\mathcal{P}_n}(x_0) = \mathop{\mathbb{E}}_{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)} \left[\mathfrak{M}^h_p(\tilde{\mathcal{F}}_n, x_0) \right]$$

- **Theorem**

$$\left| J^h(x_0) - E^h_{p,\mathcal{P}_n}(x_0) \right| \leq C \alpha_{pT}(\mathcal{P}_n)$$

$$with\ C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left[L_f (1 + L_h) \right]^i$$

MFMC estimator: analysis

- **Variance** of the MFMC estimator

$$V_{p,\mathcal{P}_n}^h(x_0) = \underset{w^1,\dots,w^n \sim p_{\mathcal{W}}(.)}{Var}\left[\mathfrak{M}_p^h(\tilde{\mathcal{F}}_n, x_0)\right]$$

$$= \underset{w^1,\dots,w^n \sim p_{\mathcal{W}}(.)}{\mathbb{E}}\left[\left(\mathfrak{M}_p^h(\tilde{\mathcal{F}}_n, x_0) - E_{p,\mathcal{P}_n}^h(x_0)\right)^2\right]$$

- **Theorem**

$$V_{p,\mathcal{P}_n}^h(x_0) \leq \left(\frac{\sigma_{R^h}(x_0)}{\sqrt{p}} + 2C\alpha_{pT}(\mathcal{P}_n)\right)^2$$

$$with\ C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left[L_f(1+L_h)\right]^i$$

Illustration

- **System** $$x_{t+1} = \sin\left(\frac{\pi}{2}(x_t + u_t + w_t)\right)$$

$$\rho(x_t, u_t, w_t) = \frac{1}{2\pi} e^{-\frac{1}{2}(x_t^2 + u_t^2)} + w_t$$

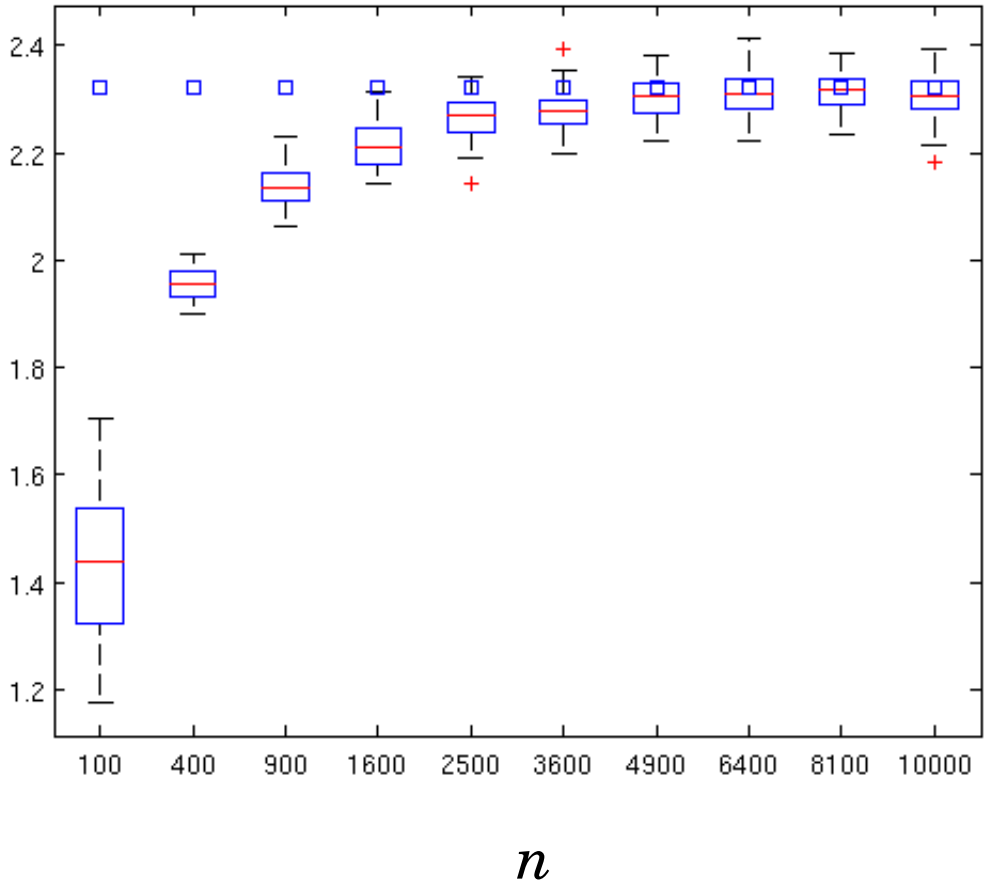
$$h(t, x) = -\frac{x}{2}, \forall x \in \mathcal{X}, \forall t \in [\![0, T-1]\!]$$

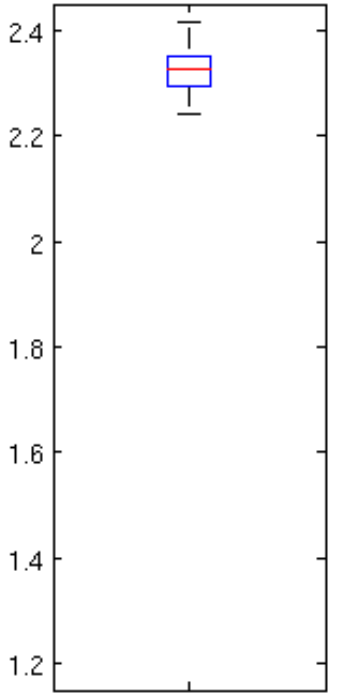
$$\mathcal{X} = [-1, 1], \mathcal{U} = [-\tfrac{1}{2}, \tfrac{1}{2}], \mathcal{W} = [-\tfrac{\epsilon}{2}, \tfrac{\epsilon}{2}] \text{ with } \epsilon = 0.1$$

- $p_W(.)$ is uniform over W, $T = 15$, $x_0 = -0.5$.

Illustration

- Simulations for p = 10, ***n* = 100 … 10 000**, uniform grid



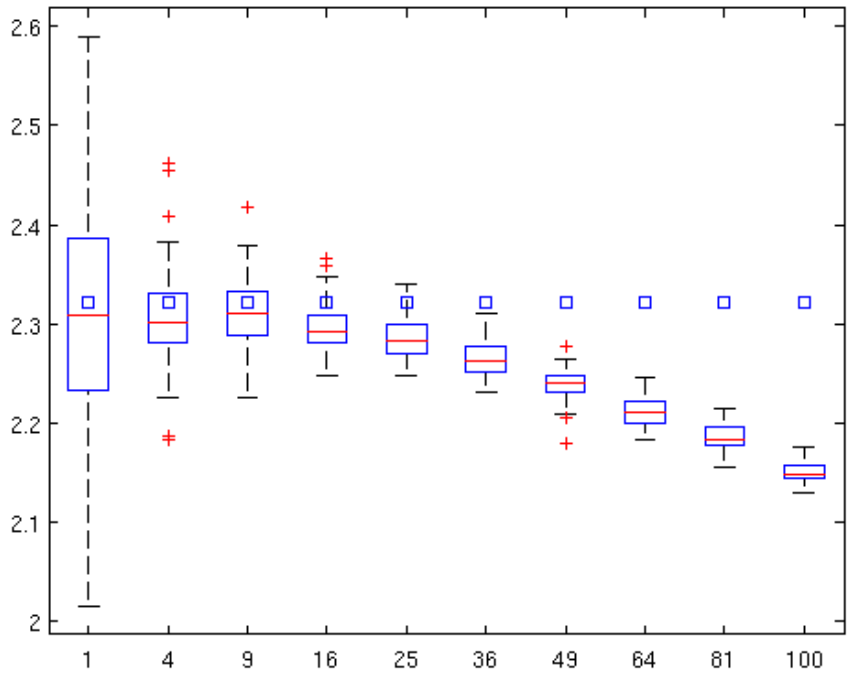


Model-free Monte Carlo estimator

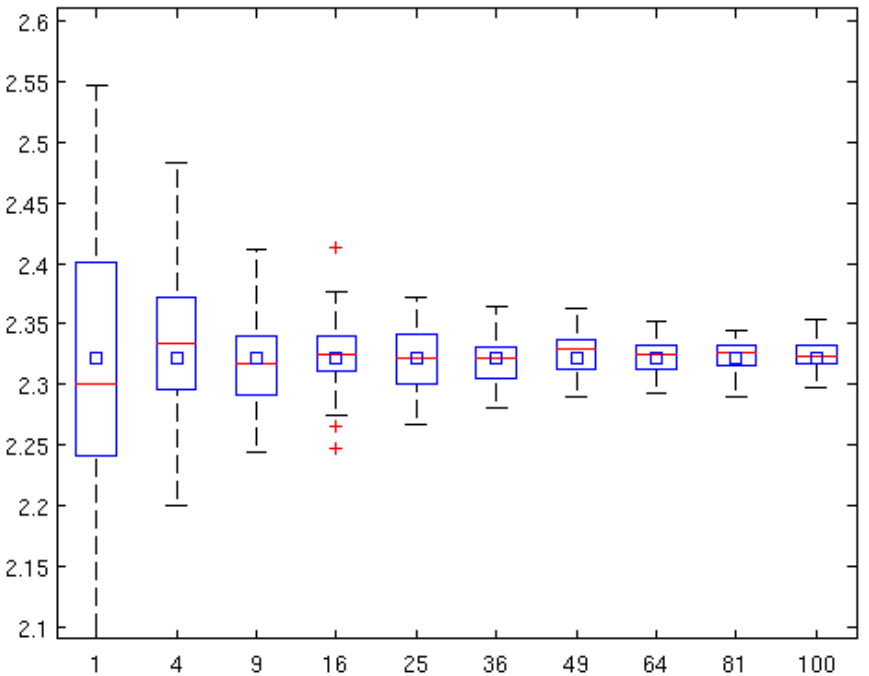
Monte Carlo estimator

Illustration

- Simulations for $p = 1 \ldots 100$, n = 10 000, uniform grid



Model-free Monte Carlo estimator



Monte Carlo estimator

Conclusions and Future work

Conclusions

- We have proposed in this paper an estimator of the expected return of a policy in a model-free setting, the MFMC estimator
- We have provided bounds on the bias and variance of the MFMC estimator
- The bias and variance of the MFMC estimator converge to the bias and variance of the MC estimator

Future work

- MFMC estimator in a direct policy search framework
- One could extend this approach to evaluate return distributions (and not only expected values). This could allow to develop "safe" policy search techniques based on Value at Risk (VaR) criteria.


Thank you