Partially observable systems and Predictive State Representation (PSR)

Nan Jiang
CS 598 Statistical RL @ UIUC

Partially observable systems

- Key assumption so far: Markov property (Markovianness)

Partially observable systems

- Key assumption so far: Markov property (Markovianness)
- Real-world is non-Markov / partially observable (PO)
    - Or you wouldn't need *memory*

Partially observable systems

- Key assumption so far: Markov property (Markovianness)
- Real-world is non-Markov / partially observable (PO)
  - Or you wouldn't need *memory*
- Examples in ML

> **Alan Mathison Turing** OBE FRS (/ˈtjʊərɪŋ/; 23 June 1912 – 7 June 1954) was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist.[2] Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the

text modeling (last word cannot predict what's next; need to capture long-term dependencies)

Partially observable systems

- Key assumption so far: Markov property (Markovianness)
- Real-world is non-Markov / partially observable (PO)
  - Or you wouldn't need *memory*
- Examples in ML

Alan Mathison Turing OBE FRS (/ˈtjʊərɪŋ/; 23 June 1912 – 7 June 1954) was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist.[2] Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the

text modeling (last word cannot predict what's next; need to capture long-term dependencies)

Prev. frame Next frame

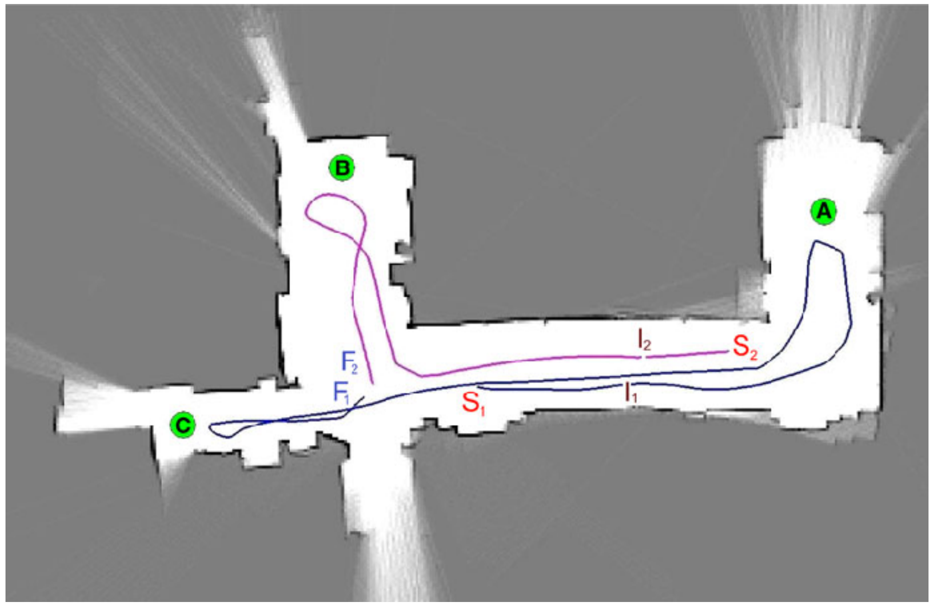
video prediction

Partially observable systems

- Key assumption so far: Markov property (Markovianness)
- Real-world is non-Markov / partially observable (PO)
  - Or you wouldn't need *memory*
- Examples in ML

Alan Mathison Turing OBE FRS (/ˈtjʊərɪŋ/; 23 June 1912 – 7 June 1954) was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist.[2] Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the

text modeling (last word cannot predict what's next; need to capture long-term dependencies)



SLAM in robotics ("this place looks familiar; *did I return to the same location*?")

"perceptual aliasing"

Prev. frame Next frame

video prediction

Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)

Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
- Actions space A (omitted for simplicity)

Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
- Actions space A (omitted for simplicity)
- System starts from initial configuration, and outputs sequences $o_1\, o_2\, o_3 \ldots$ with randomness

Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
- Actions space A (omitted for simplicity)
- System starts from initial configuration, and outputs sequences $o_1\, o_2\, o_3 \ldots$ with randomness
- Markov systems is a special case:

Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
- Actions space A (omitted for simplicity)
- System starts from initial configuration, and outputs sequences $o_1\, o_2\, o_3 \ldots$ with randomness
- Markov systems is a special case:

$$\Pr[o_{\tau+1:\tau+k} \mid o_{1:\tau}] = \Pr[o_{\tau+1:\tau+k} \mid o_\tau]$$

Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
- Actions space A (omitted for simplicity)
- System starts from initial configuration, and outputs sequences $o_1\, o_2\, o_3 \dots$ with randomness
- Markov systems is a special case:

$$\Pr[o_{\tau+1:\tau+k} \mid o_{1:\tau}] = \Pr[o_{\tau+1:\tau+k} \mid o_\tau]$$

or, $\boldsymbol{o}_{\tau+1:\tau+k} \perp \boldsymbol{o}_{1:\tau} \mid \boldsymbol{o}_\tau$ (bold r.v.; non-bold realization)

  - In words, last observation is *sufficient statistics of history* for predicting future observations

Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
- Actions space A (omitted for simplicity)
- System starts from initial configuration, and outputs sequences $o_1\, o_2\, o_3 \ldots$ with randomness
- Markov systems is a special case:

$$\Pr[o_{\tau+1:\tau+k} \mid o_{1:\tau}] = \Pr[o_{\tau+1:\tau+k} \mid o_\tau]$$

or, $\boldsymbol{o}_{\tau+1:\tau+k} \perp \boldsymbol{o}_{1:\tau} \mid \boldsymbol{o}_\tau$ (bold r.v.; non-bold realization)

  - In words, last observation is *sufficient statistics of history* for predicting future observations
- How restrictive is Markov assumption?

Complexity of Markov vs non-Markov systems

- For a Markov chain, the complexity is measured by the number of states (i.e., number of observations)

Complexity of Markov vs non-Markov systems

- For a Markov chain, the complexity is measured by the number of states (i.e., number of observations)
  - System fully specified by the transition matrix $T(o' \mid o)$

Complexity of Markov vs non-Markov systems

- For a Markov chain, the complexity is measured by the number of states (i.e., number of observations)
  - System fully specified by the transition matrix $T(o' \mid o)$
  - # model parameters = $|O|^2$

Complexity of Markov vs non-Markov systems

- For a Markov chain, the complexity is measured by the number of states (i.e., number of observations)
  - System fully specified by the transition matrix $T(o' \mid o)$
  - # model parameters = $|O|^2$
- Without Markov assumption?

Complexity of Markov vs non-Markov systems

- For a Markov chain, the complexity is measured by the number of states (i.e., number of observations)
  - System fully specified by the transition matrix $T(o' \mid o)$
  - # model parameters = $|O|^2$
- Without Markov assumption?
  - System fully specified by $\Pr[o' \mid h]$ for any history h (short for $o_{1:\tau}$)

Complexity of Markov vs non-Markov systems

- For a Markov chain, the complexity is measured by the number of states (i.e., number of observations)
  - System fully specified by the transition matrix $T(o' \mid o)$
  - # model parameters = $|O|^2$
- Without Markov assumption?
  - System fully specified by $\Pr[o' \mid h]$ for any history h (short for $o_{1:\tau}$)
  - Probabilities for different histories can be set completely independently— with horizon L, order $|O|^L$ free parameters!

Complexity of Markov vs non-Markov systems

- For a Markov chain, the complexity is measured by the number of states (i.e., number of observations)
  - System fully specified by the transition matrix $T(o' \mid o)$
  - # model parameters = $|O|^2$
- Without Markov assumption?
  - System fully specified by $\Pr[o' \mid h]$ for any history h (short for $o_{1:\tau}$)
  - Probabilities for different histories can be set completely independently— with horizon L, order $|O|^L$ free parameters!
  - Even with a finite and constant observation space, fully general dynamical systems are intractable

Complexity of Markov vs non-Markov systems

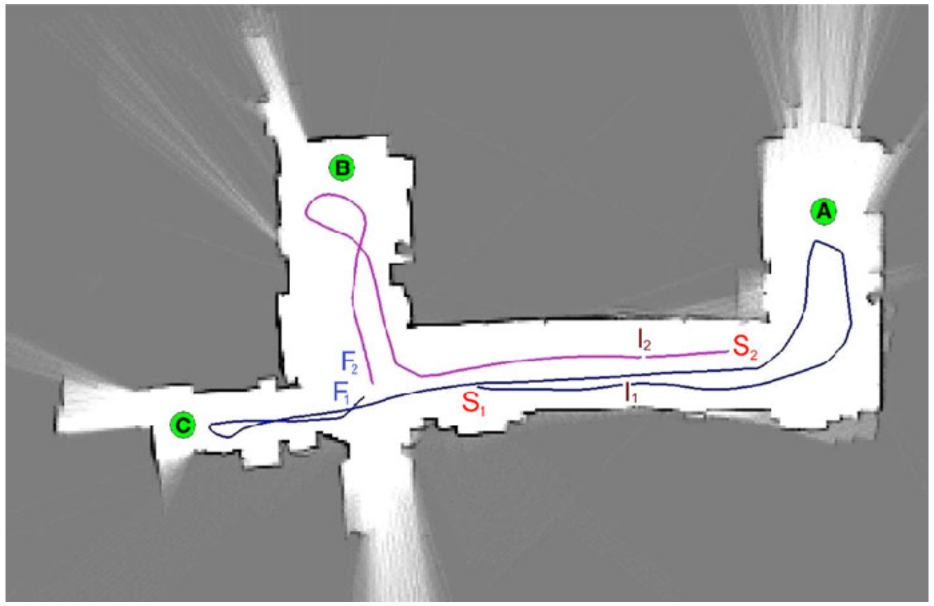
- For a Markov chain, the complexity is measured by the number of states (i.e., number of observations)
  - System fully specified by the transition matrix $T(o' \mid o)$
  - # model parameters = $|O|^2$
- Without Markov assumption?
  - System fully specified by $\Pr[o' \mid h]$ for any history h (short for $o_{1:\tau}$)
  - Probabilities for different histories can be set completely independently— with horizon L, order $|O|^L$ free parameters!
  - Even with a finite and constant observation space, fully general dynamical systems are intractable
  - Need structure…

Partially observable systems

- Example structure: small & finite *latent* state space

Partially observable systems

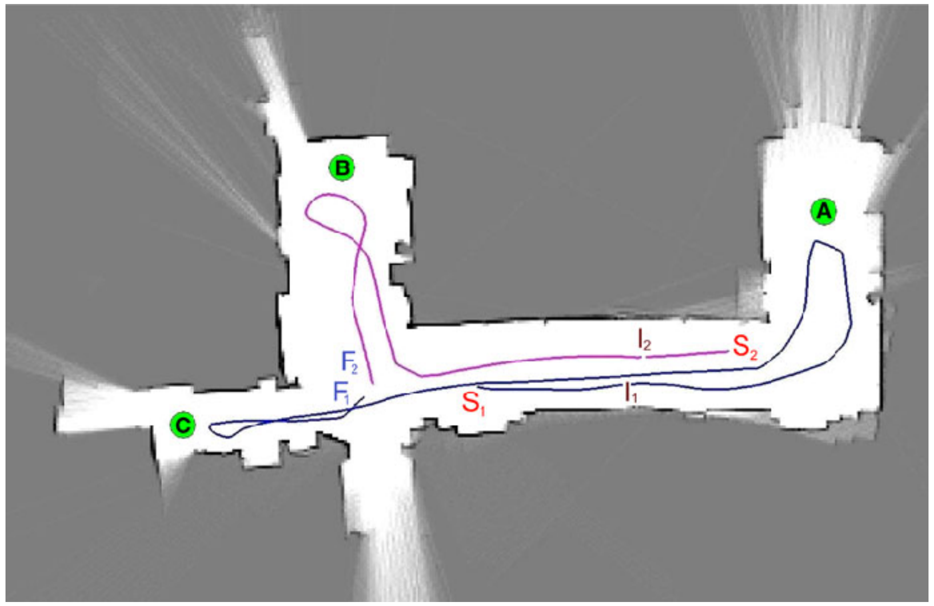
- Example structure: small & finite *latent* state space
- “this place looks familiar; did I return to the same location?”



SLAM in robotics (“this scene looks familiar; *did I return to the same location*?”)

Partially observable systems

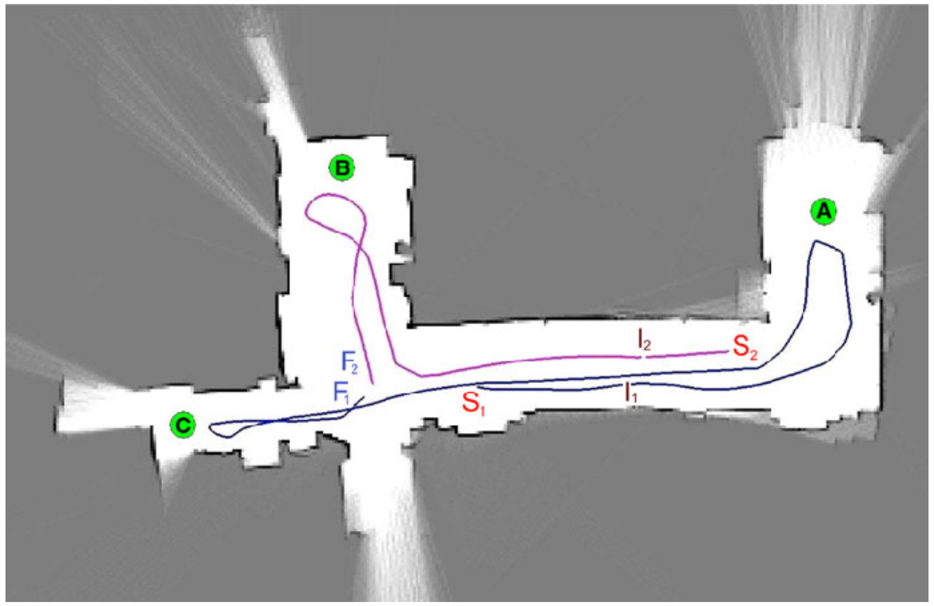
- Example structure: small & finite *latent* state space
- "this place looks familiar; did I return to the same location?"
    - General PO system: you always visit a new location



SLAM in robotics ("this scene looks familiar; *did I return to the same location*?")

Partially observable systems

- Example structure: small & finite *latent* state space
- "this place looks familiar; did I return to the same location?"
    - General PO system: you always visit a new location
    - With structural assumptions: the building only has this many different rooms. You will return to one or another.



SLAM in robotics ("this scene looks familiar; *did I return to the same location*?")

Latent Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
  - SLAM example: current sensory inputs

Latent Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
  - SLAM example: current sensory inputs
- Action space A (again will ignore for simplicity in most places)

Latent Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
  - SLAM example: current sensory inputs
- Action space A (again will ignore for simplicity in most places)
- Latent/hidden state space Z
  - SLAM example: true location

Latent Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
  - SLAM example: current sensory inputs
- Action space A (again will ignore for simplicity in most places)
- Latent/hidden state space Z
  - SLAM example: true location
- Model parameters
  - Emission probability: $E(o \mid z)$
  - Transition probability: $T(z' \mid z, a)$

Latent Models of PO systems

- Observation space O (finite & discrete w.l.o.g.)
  - SLAM example: current sensory inputs
- Action space A (again will ignore for simplicity in most places)
- Latent/hidden state space Z
  - SLAM example: true location
- Model parameters
  - Emission probability: $E(o \mid z)$
  - Transition probability: $T(z' \mid z, a)$
- Markov chain is special case: identity emission

Myth 1 about HMMs/POMDPs

- PO can stem from noisy sensors, which compresses/loses information from “world state”

Myth 1 about HMMs/POMDPs

- PO can stem from noisy sensors, which compresses/loses information from “world state”
- Noisier sensors = more PO?

Myth 1 about HMMs/POMDPs

- PO can stem from noisy sensors, which compresses/loses information from “world state”
- Noisier sensors = more PO?
- Mathematically, if we fix the underlying MDP and vary the emission function, an emission that loses more information gives a more PO process?

Myth 1 about HMMs/POMDPs

- PO can stem from noisy sensors, which compresses/loses information from "world state"
- Noisier sensors = more PO?
- Mathematically, if we fix the underlying MDP and vary the emission function, an emission that loses more information gives a more PO process?
- Wrong: If emission discards all information, the process becomes Markov!

Myth 2 about HMMs/POMDPs

- When the problem is non-Markov, people will say “oh it’s a POMDP”

Myth 2 about HMMs/POMDPs

- When the problem is non-Markov, people will say “oh it’s a POMDP”
- …which assumes POMDP is fully general?

Myth 2 about HMMs/POMDPs

- When the problem is non-Markov, people will say “oh it’s a POMDP”
- …which assumes POMDP is fully general?
- Not really: there are systems that can be succinctly represented but require infinitely many hidden states to be represented as a POMDP/HMM

Myth 2 about HMMs/POMDPs

- When the problem is non-Markov, people will say “oh it’s a POMDP”
- …which assumes POMDP is fully general?
- Not really: there are systems that can be succinctly represented but require infinitely many hidden states to be represented as a POMDP/HMM
- Again, one most generic way to specify a PO system is just $\Pr[o' \mid o_{1:\tau}]$, or $\Pr[o' \mid h]$ for short (h for history)

Major challenge in PO systems: *state* representation

- Examples

Major challenge in PO systems: *state* representation

- Examples
  - Text prediction: how to *compactly summarize* the sentence so far to predict future words? (that's what you are computing as the hidden vector in an LSTM)

Major challenge in PO systems: *state* representation

- Examples
  - Text prediction: how to *compactly summarize* the sentence so far to predict future words? (that's what you are computing as the hidden vector in an LSTM)
  - SLAM: how to map history of sensor readings to physical locations (or a belief about it)

Major challenge in PO systems: *state* representation

- Examples
  - Text prediction: how to *compactly summarize* the sentence so far to predict future words? (that's what you are computing as the hidden vector in an LSTM)
  - SLAM: how to map history of sensor readings to physical locations (or a belief about it)
- What does state mean in the PO setting?

Major challenge in PO systems: *state* representation

- Examples
  - Text prediction: how to *compactly summarize* the sentence so far to predict future words? (that's what you are computing as the hidden vector in an LSTM)
  - SLAM: how to map history of sensor readings to physical locations (or a belief about it)
- What does state mean in the PO setting?

Definition: **State** is a **function of history**, ϕ, that is a **sufficient statistics** for **predicting future**. That is, for all $t{:=}o_{\tau+1:\tau+k}$ and $h{:=}o_{1:\tau}$,

$$\Pr[t \mid h] = \Pr[t \mid \phi(h)]$$

State!

- Trivial function that is state?

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$
  - There is another one. will reveal later…

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$
  - There is another one. will reveal later…
- For HMMs/POMDPs, belief state, $(\Pr[\boldsymbol{z}_\tau{=}z \mid h])_{z \in Z}$, *is state*

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$
  - There is another one. will reveal later…
- For HMMs/POMDPs, belief state, $(\Pr[\boldsymbol{z}_\tau = z \mid h])_{z \in Z}$, *is state*
- To an old-school RL person, be careful when you say “state” without a modifier…

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$
  - There is another one. will reveal later…
- For HMMs/POMDPs, belief state, $(\Pr[\boldsymbol{z}_\tau = z \mid h])_{z \in Z}$, *is state*
- To an old-school RL person, be careful when you say “state” without a modifier…
- Things that are not states and people call “state”

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$
  - There is another one. will reveal later…
- For HMMs/POMDPs, belief state, $(\Pr[\boldsymbol{z}_\tau = z \mid h])_{z \in Z}$, *is state*
- To an old-school RL person, be careful when you say “state” without a modifier…
- Things that are not states and people call “state”
  - Observation: e.g., Atari game frame

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$
  - There is another one. will reveal later…
- For HMMs/POMDPs, belief state, $(\Pr[z_\tau = z \mid h])_{z \in Z}$, *is state*
- To an old-school RL person, be careful when you say “state” without a modifier…
- Things that are not states and people call “state”
  - Observation: e.g., Atari game frame
  - Hidden state (“World state”) : Why?

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$
  - There is another one. will reveal later…
- For HMMs/POMDPs, belief state, $(\Pr[z_\tau = z \mid h])_{z \in Z}$, *is state*
- To an old-school RL person, be careful when you say “state” without a modifier…
- Things that are not states and people call “state”
  - Observation: e.g., Atari game frame
  - Hidden state (“World state”) : not a function of history

State!

- Trivial function that is state?
  - History itself (identity map): $\phi(h) = h$
  - There is another one. will reveal later…
- For HMMs/POMDPs, belief state, $(\Pr[z_\tau = z \mid h])_{z \in Z}$, *is state*
- To an old-school RL person, be careful when you say "state" without a modifier…
- Things that are not states and people call "state"
  - Observation: e.g., Atari game frame
  - Hidden state ("World state") : not a function of history
  - Agent state: can be approximately a state

Issues with Latent Variable Models

- Typical learning algorithm for HMMs: EM

Issues with Latent Variable Models

- Typical learning algorithm for HMMs: EM
- Subject to local optimum

Issues with Latent Variable Models

- Typical learning algorithm for HMMs: EM
- Subject to local optimum
- More deeply: hidden state is an *ungrounded* object. If we re-order the hidden state, that gives exactly the same process (over observables)!

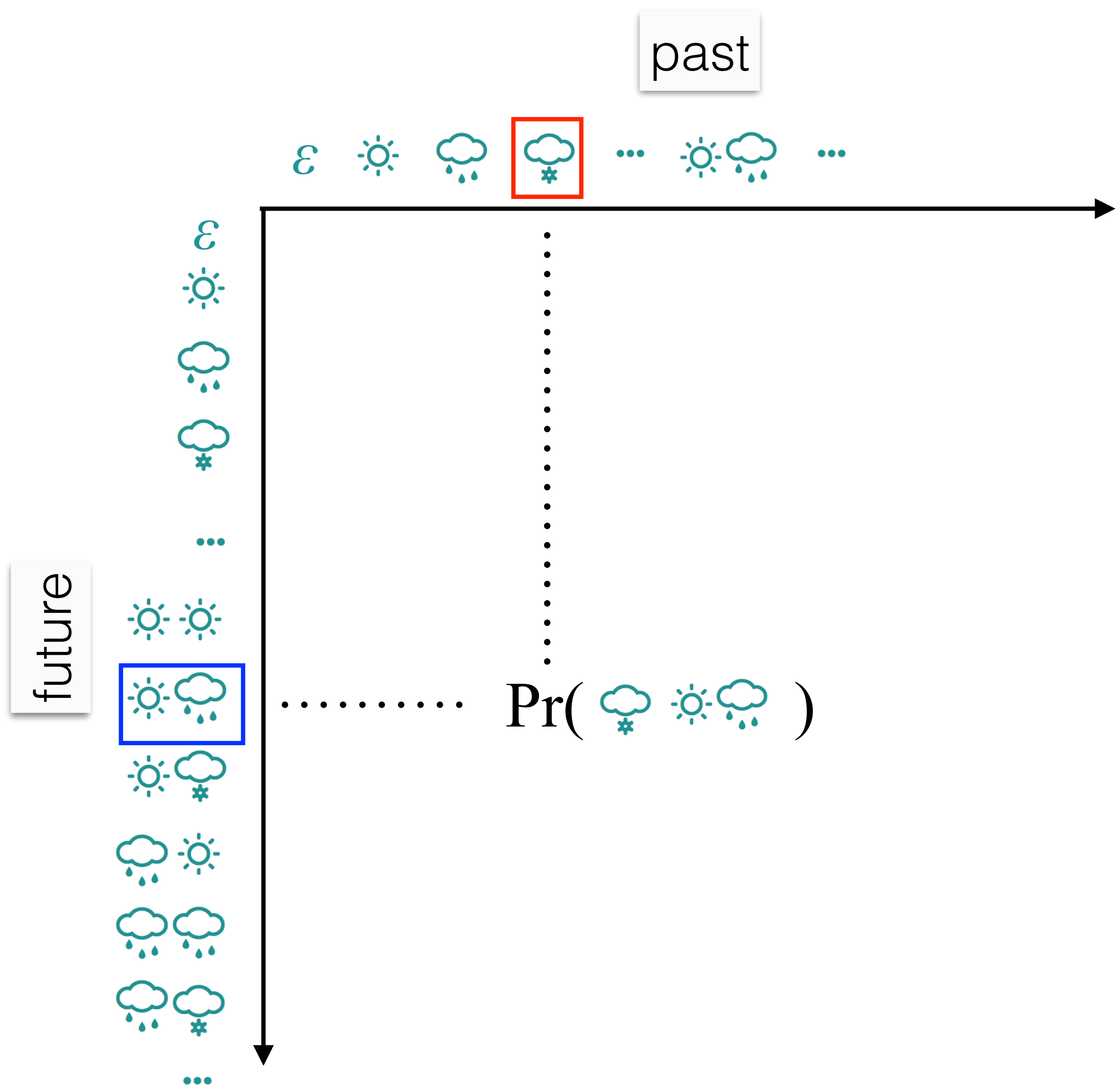
Issues with Latent Variable Models

- Typical learning algorithm for HMMs: EM
- Subject to local optimum
- More deeply: hidden state is an *ungrounded* object. If we re-order the hidden state, that gives exactly the same process (over observables)!
- World state is illusion; all matters is our sensory-motor experience. "*to be is to be perceived*" (George Berkeley)

Issues with Latent Variable Models

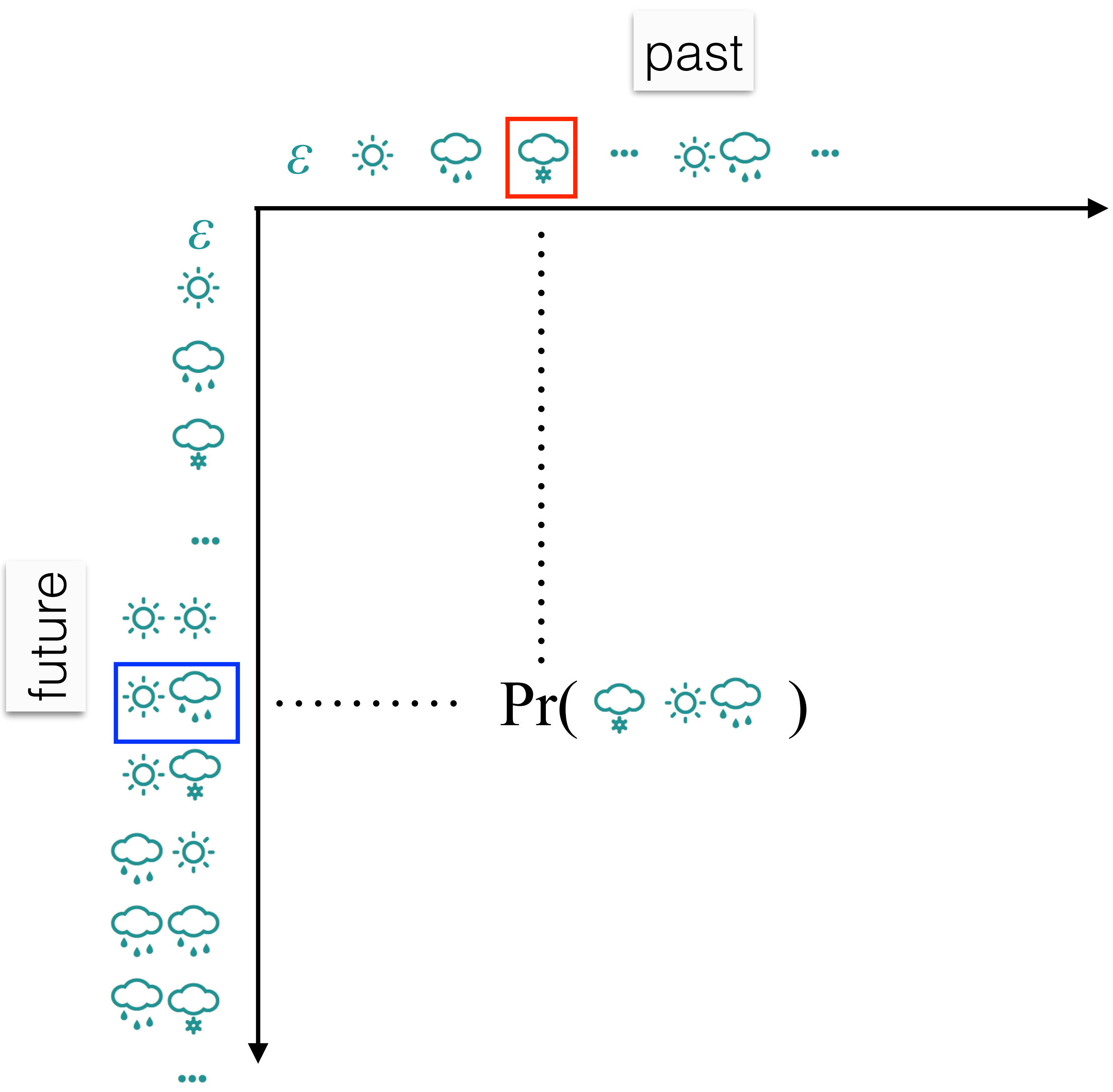
- Typical learning algorithm for HMMs: EM
- Subject to local optimum
- More deeply: hidden state is an *ungrounded* object. If we re-order the hidden state, that gives exactly the same process (over observables)!
- World state is illusion; all matters is our sensory-motor experience. "*to be is to be perceived*" (George Berkeley)
- But how to inject structure???

The system dynamics matrix M



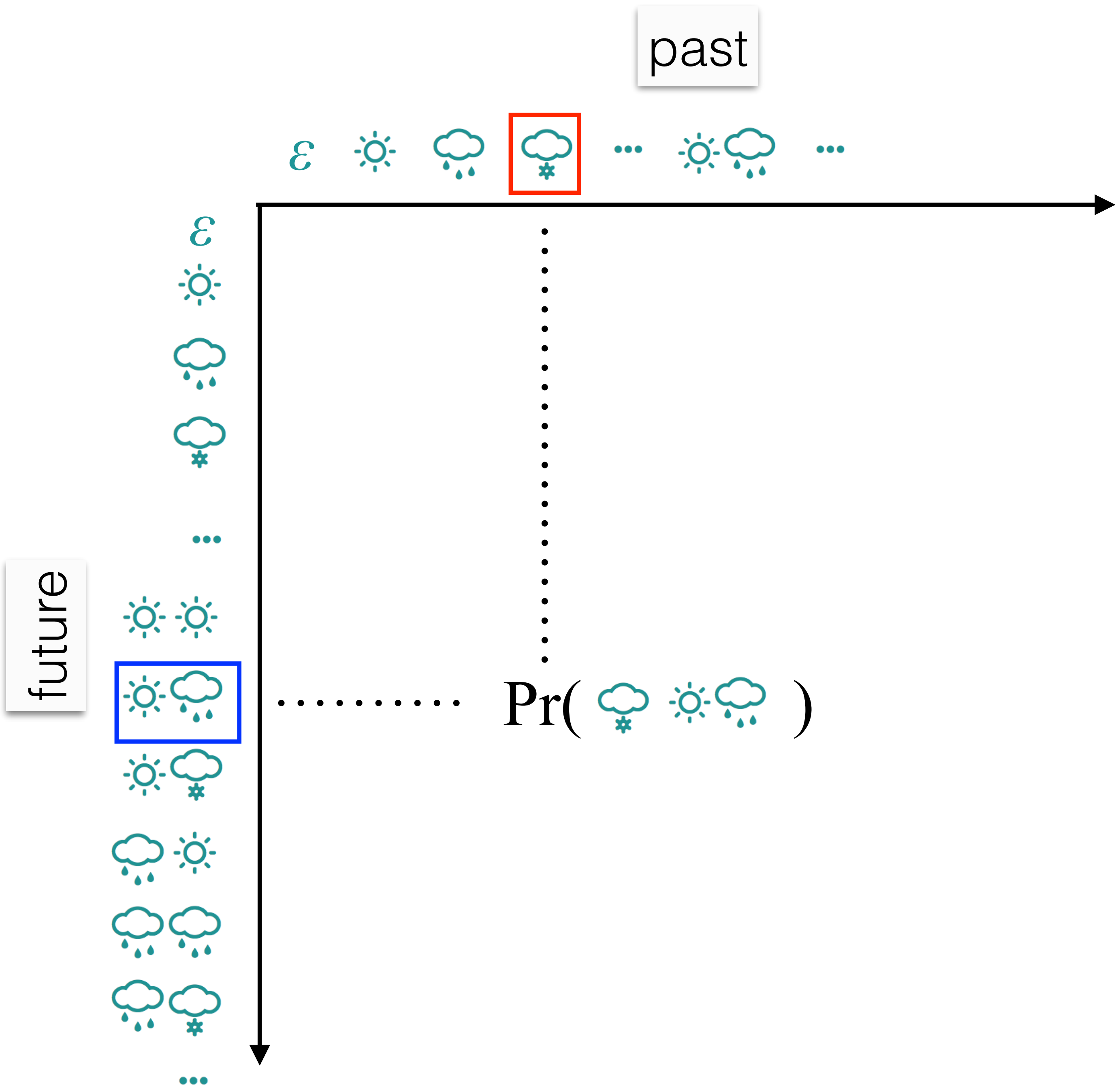
The system dynamics matrix M

- Recall that $\Pr[o' \mid h]$ fully specifies a PO system.



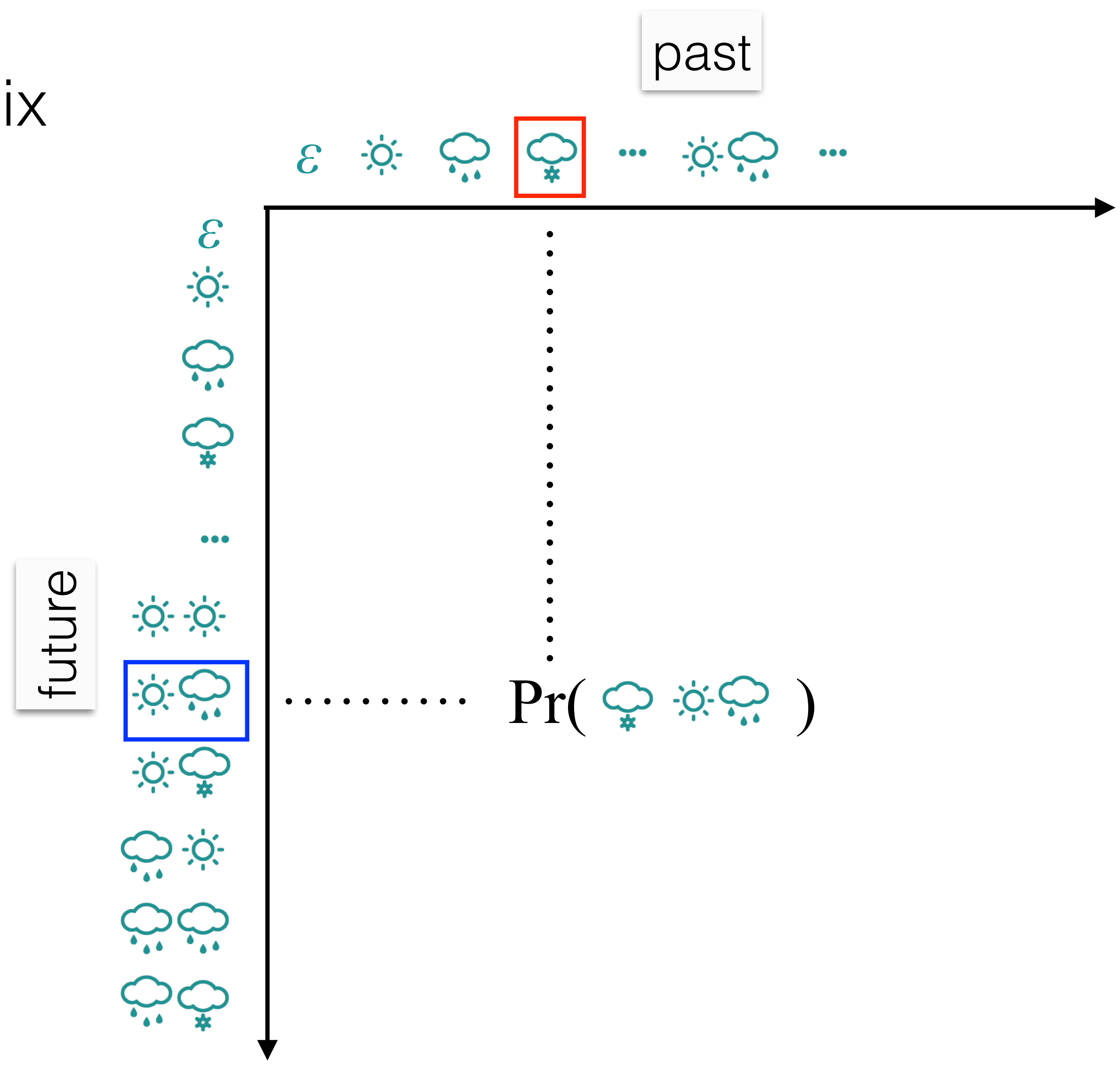
The system dynamics matrix M

- Recall that $\Pr[o' \mid h]$ fully specifies a PO system.
- Alternatively, $\Pr[h]$ also does the job (w/ some redundancy; can you tell?)



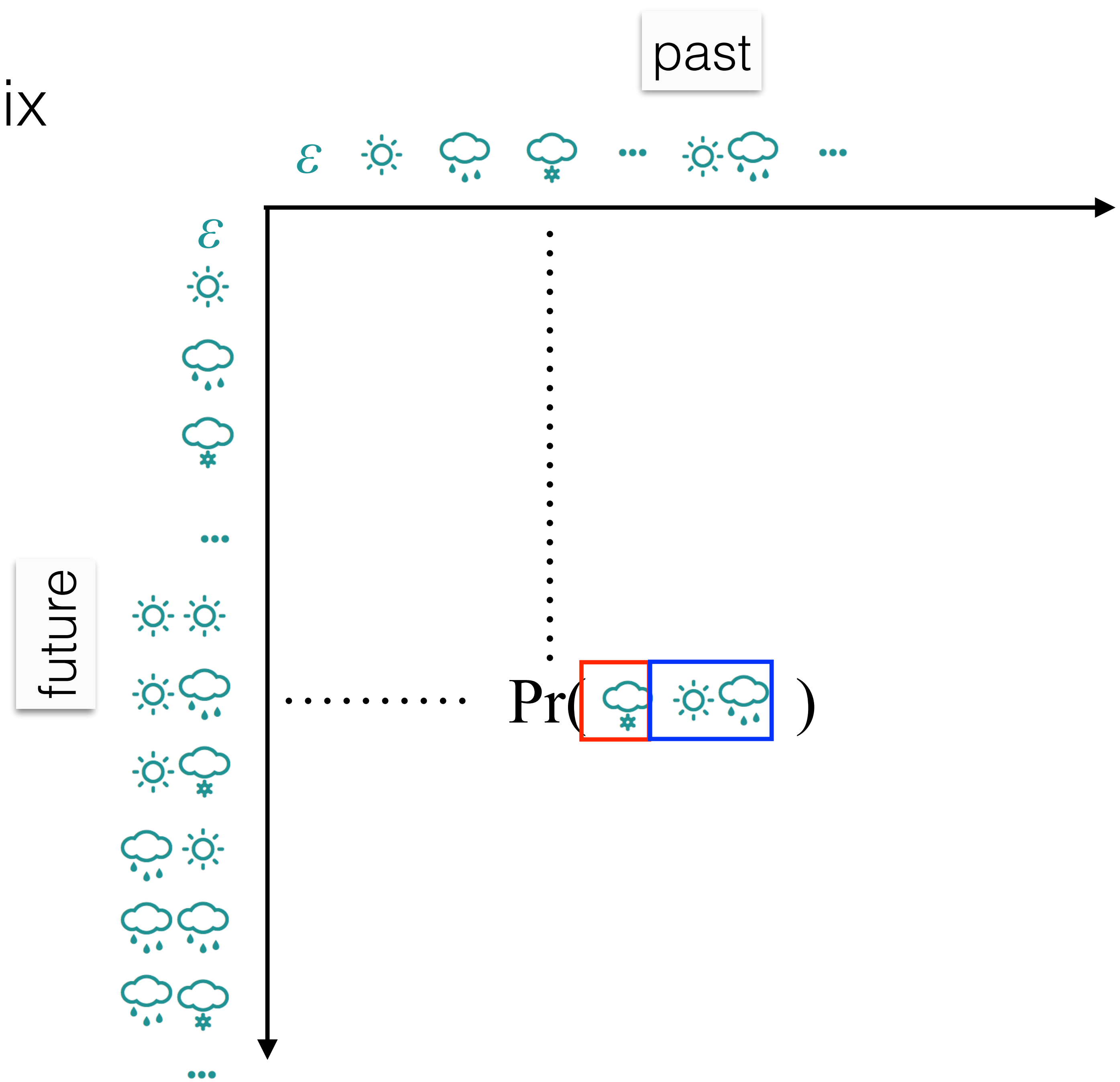
The system dynamics matrix M

- Recall that $\Pr[o' \mid h]$ fully specifies a PO system.
- Alternatively, $\Pr[h]$ also does the job (w/ some redundancy; can you tell?)
- Let's stack them in a matrix



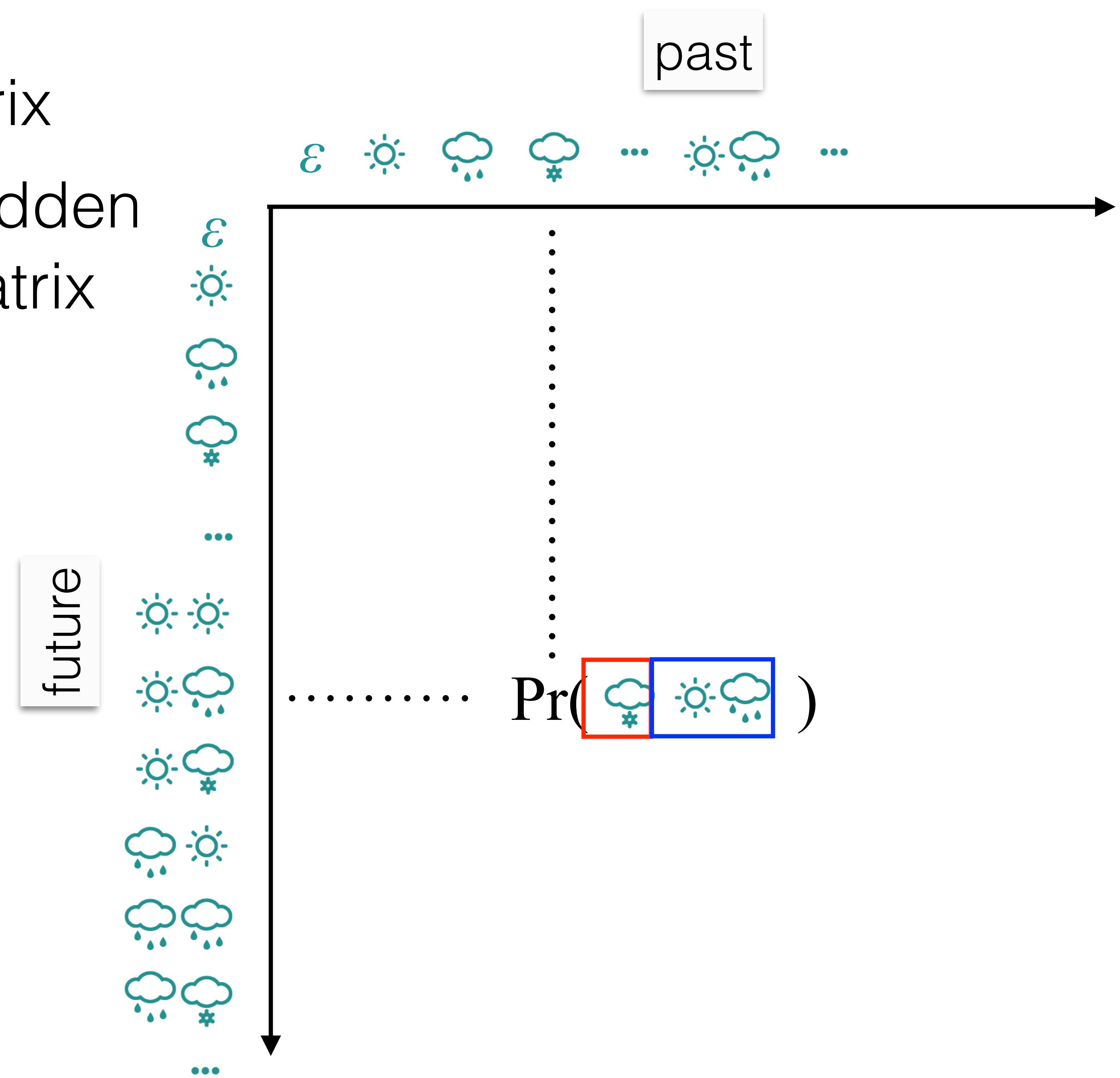
The system dynamics matrix M

- Recall that $\Pr[o' \mid h]$ fully specifies a PO system.
- Alternatively, $\Pr[h]$ also does the job (w/ some redundancy; can you tell?)
- Let's stack them in a matrix



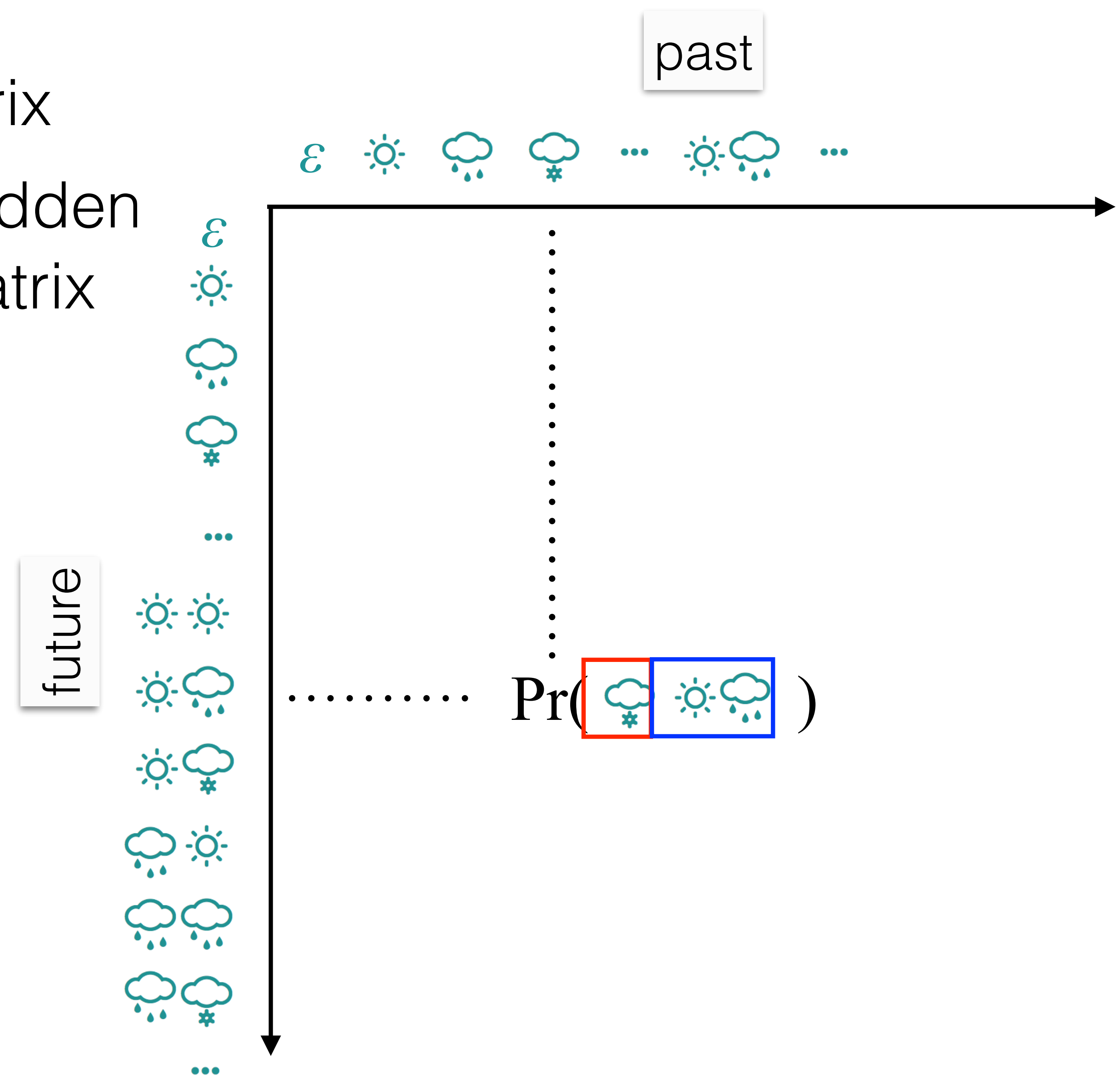
The system dynamics matrix M

- Recall that $\Pr[o' \mid h]$ fully specifies a PO system.
- Alternatively, $\Pr[h]$ also does the job (w/ some redundancy; can you tell?)
- Let's stack them in a matrix
- Claim: For HMM with n hidden states, the rank of this matrix is at most n



The system dynamics matrix M

- Recall that $\Pr[o' \mid h]$ fully specifies a PO system.
- Alternatively, $\Pr[h]$ also does the job (w/ some redundancy; can you tell?)
- Let's stack them in a matrix
- Claim: For HMM with n hidden states, the rank of this matrix is at most n



See project ref page for classical refs for PSRs
http://nanjiang.cs.illinois.edu/cs598project/

Low-rankness of SDM

- Proof: for any past h and future t, let the current timestep be τ

$$
\begin{aligned}
\Pr[ht] &= \sum_{z \in \mathcal{Z}} \Pr[ht, \mathbf{z}_\tau = z] \\
&= \sum_{z \in \mathcal{Z}} \Pr[h, \mathbf{z}_\tau = z] \Pr[t \mid \mathbf{z}_\tau = z, h] \\
&= \sum_{z \in \mathcal{Z}} \Pr[h, \mathbf{z}_\tau = z] \Pr[t \mid \mathbf{z}_\tau = z].
\end{aligned}
$$

Low-rankness of SDM

- Proof: for any past h and future t, let the current timestep be τ

$$\begin{aligned}\Pr[ht] &= \sum_{z\in\mathcal{Z}} \Pr[ht, \mathbf{z}_\tau = z] \\ &= \sum_{z\in\mathcal{Z}} \Pr[h, \mathbf{z}_\tau = z]\Pr[t \mid \mathbf{z}_\tau = z, h] \\ &= \sum_{z\in\mathcal{Z}} \Pr[h, \mathbf{z}_\tau = z]\Pr[t \mid \mathbf{z}_\tau = z].\end{aligned}$$

- Dot-product between two vectors of dimension |Z|: one only depends on history and the other only depends on future—implies low-rankness

Low-rankness of SDM

- Proof: for any past h and future t, let the current timestep be τ

$$\begin{aligned}
\Pr[ht] &= \sum_{z \in \mathcal{Z}} \Pr[ht, \mathbf{z}_\tau = z] \\
&= \sum_{z \in \mathcal{Z}} \Pr[h, \mathbf{z}_\tau = z] \Pr[t \mid \mathbf{z}_\tau = z, h] \\
&= \sum_{z \in \mathcal{Z}} \Pr[h, \mathbf{z}_\tau = z] \Pr[t \mid \mathbf{z}_\tau = z].
\end{aligned}$$

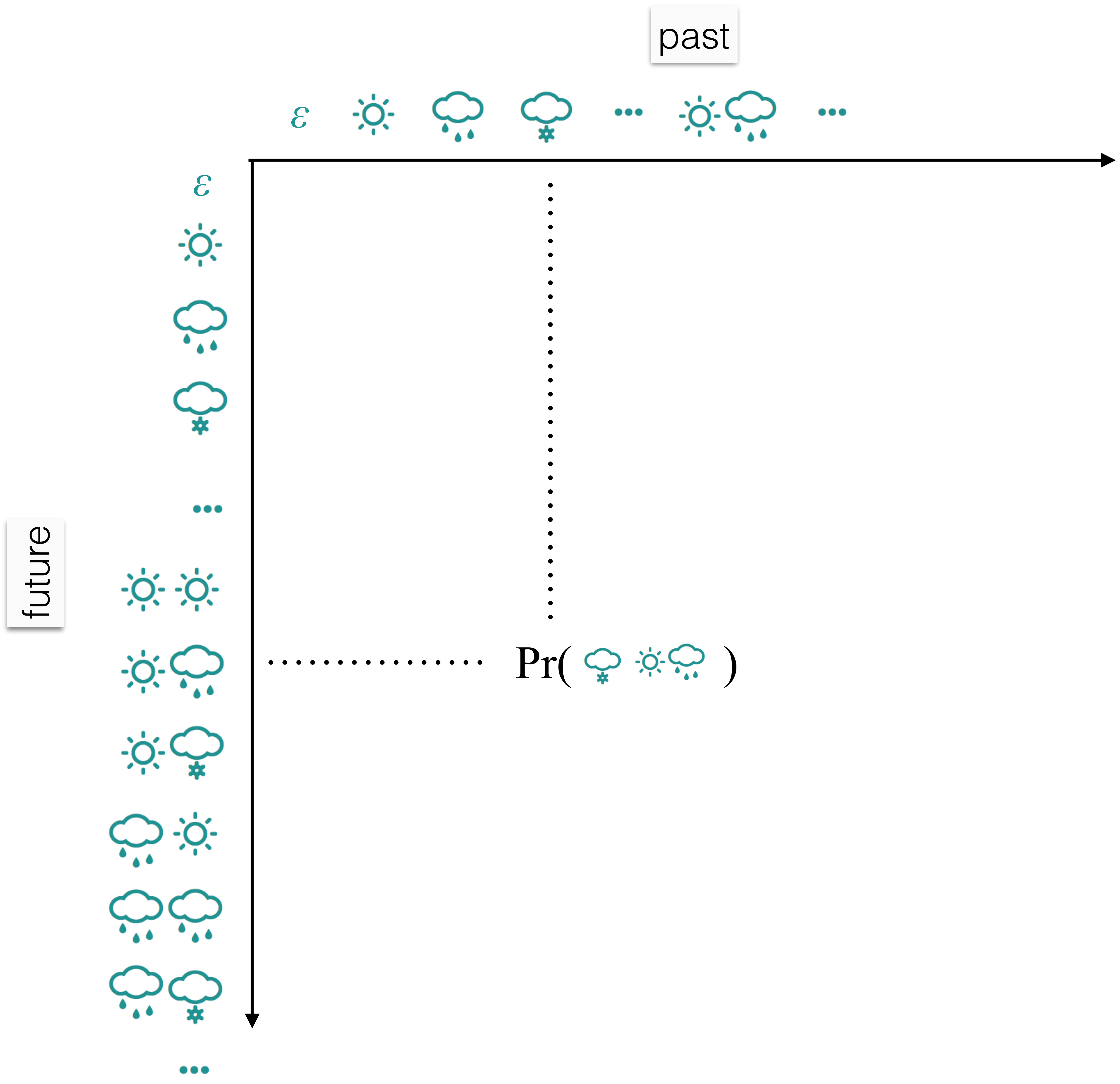
- Dot-product between two vectors of dimension |Z|: one only depends on history and the other only depends on future—implies low-rankness
- rank of SDM is known as the *linear dimension* of the system

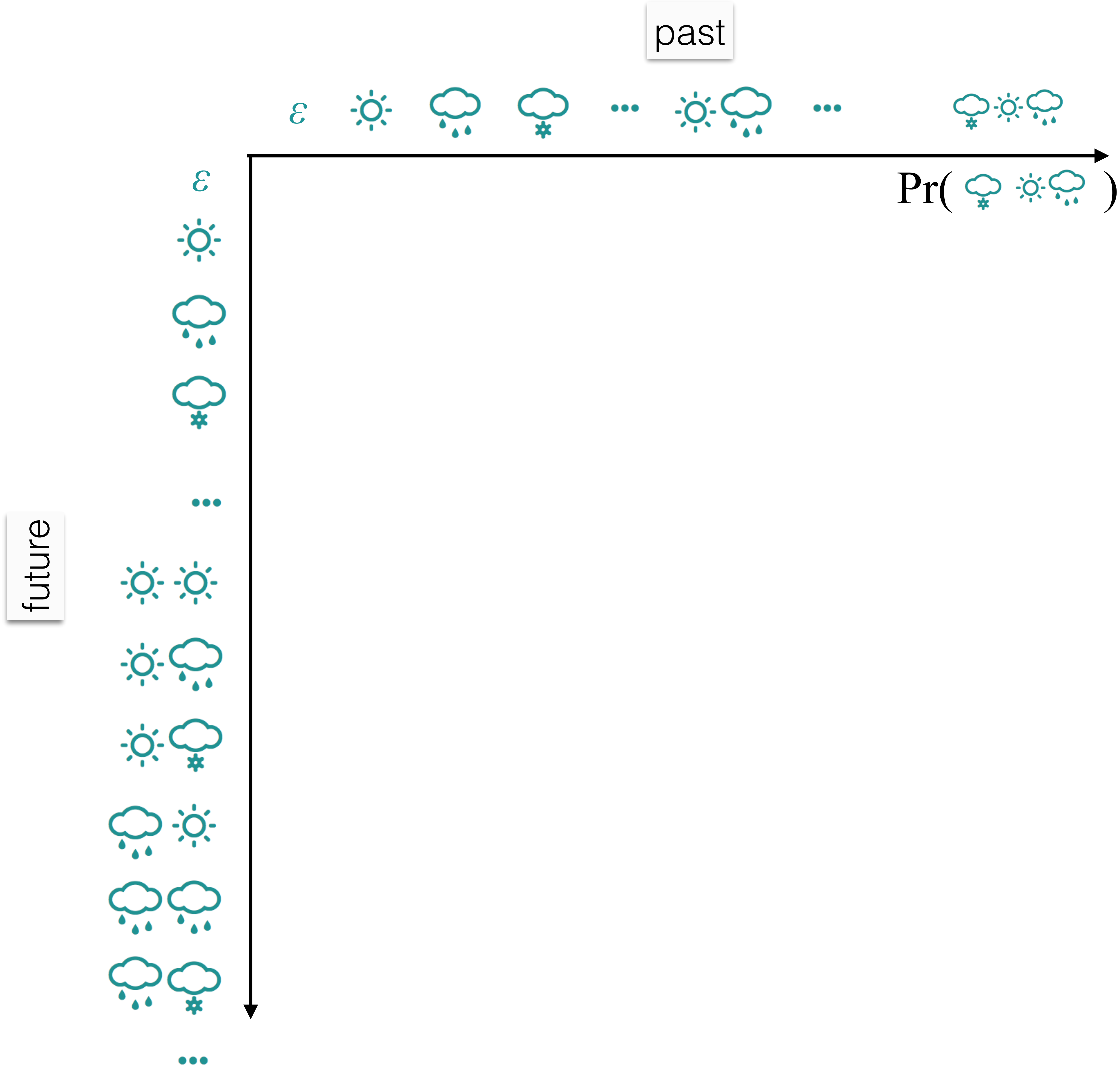
Low-rankness of SDM

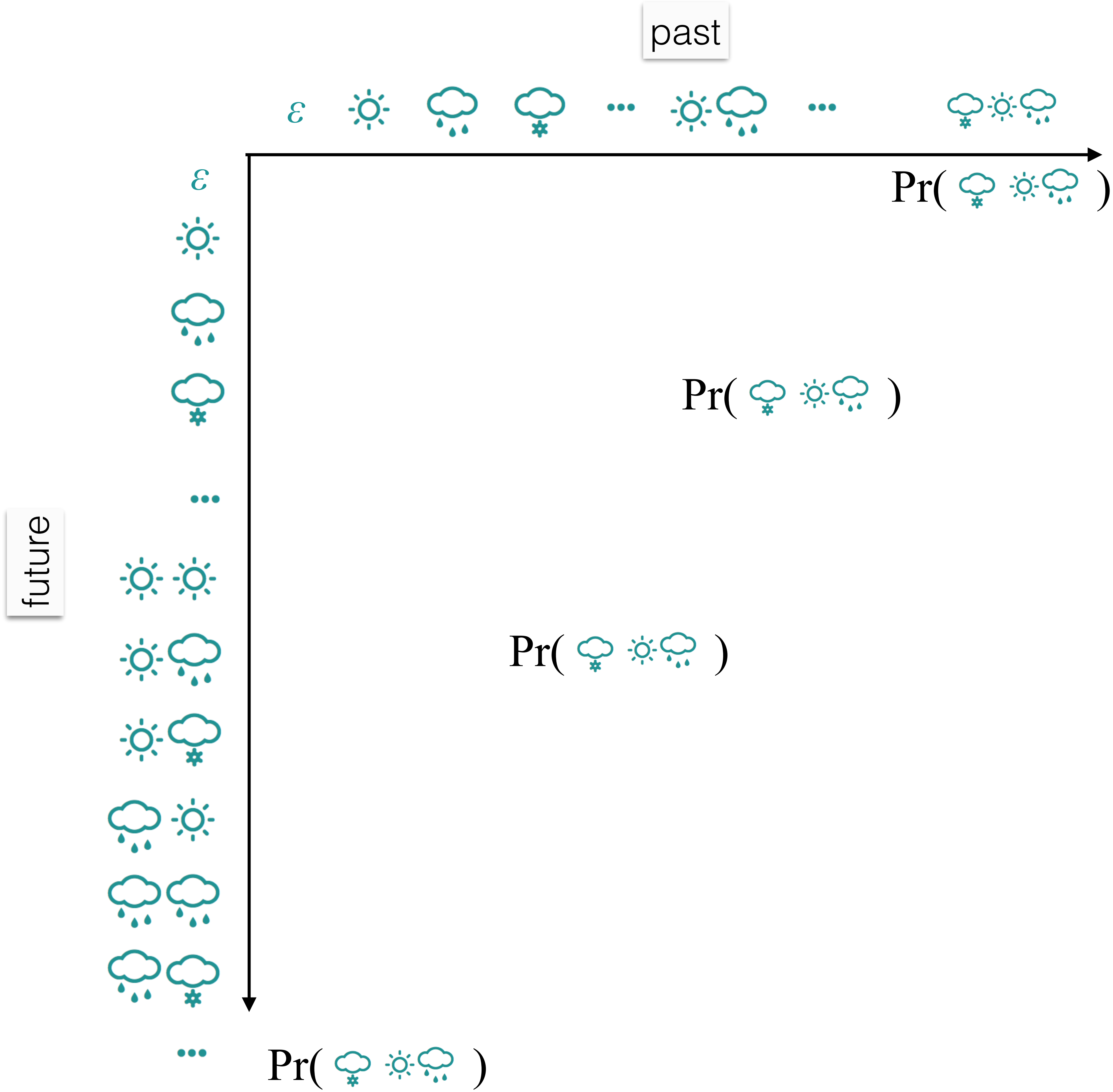
- Proof: for any past h and future t, let the current timestep be τ

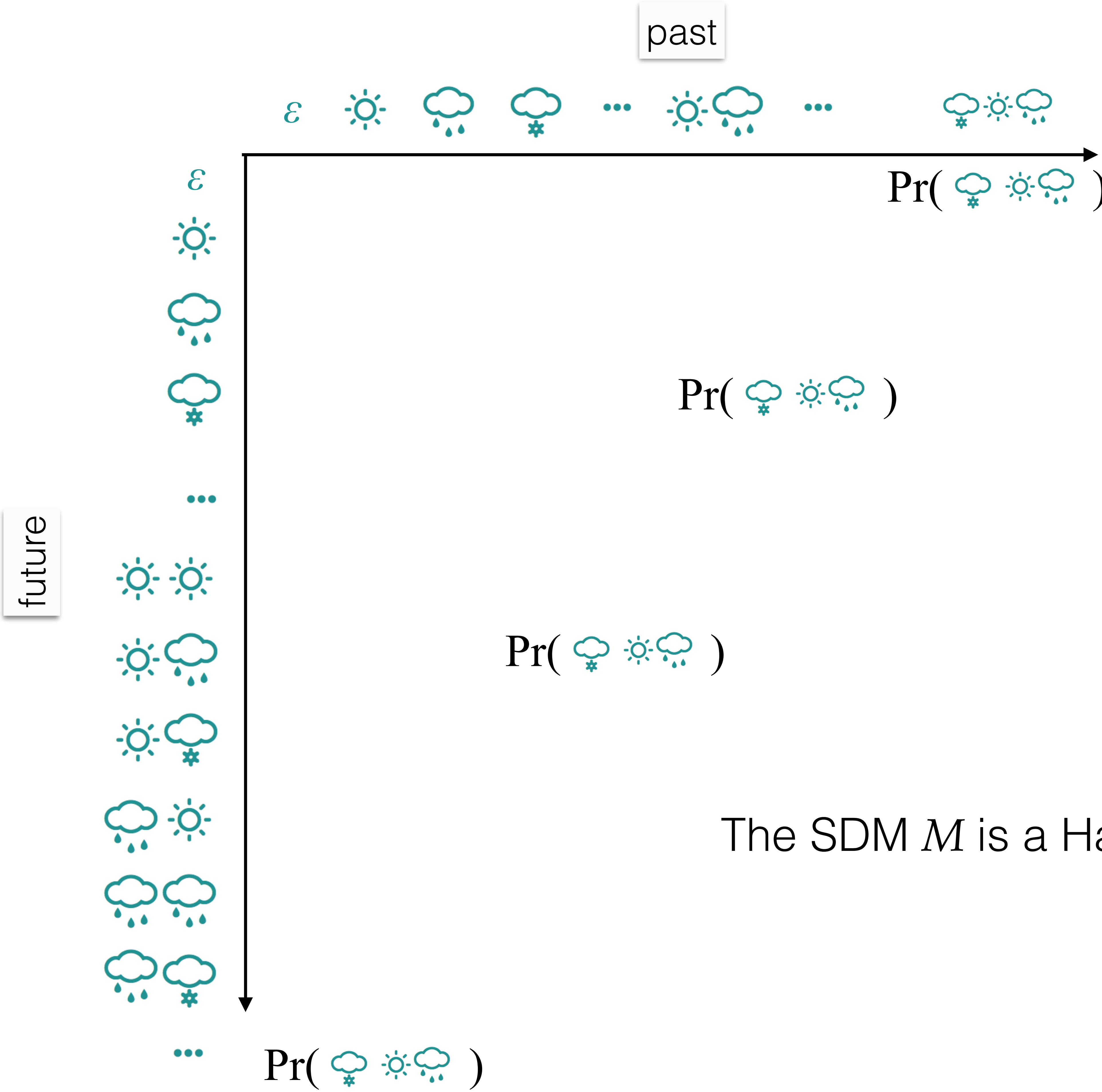
$$\begin{aligned}
\Pr[ht] &= \sum_{z \in \mathcal{Z}} \Pr[ht, \mathbf{z}_\tau = z] \\
&= \sum_{z \in \mathcal{Z}} \Pr[h, \mathbf{z}_\tau = z] \Pr[t \mid \mathbf{z}_\tau = z, h] \\
&= \sum_{z \in \mathcal{Z}} \Pr[h, \mathbf{z}_\tau = z] \Pr[t \mid \mathbf{z}_\tau = z].
\end{aligned}$$

- Dot-product between two vectors of dimension |Z|: one only depends on history and the other only depends on future—implies low-rankness
- rank of SDM is known as the *linear dimension* of the system
- Can we directly work with systems whose SDM has low-rank, instead of going through the latent variable route???

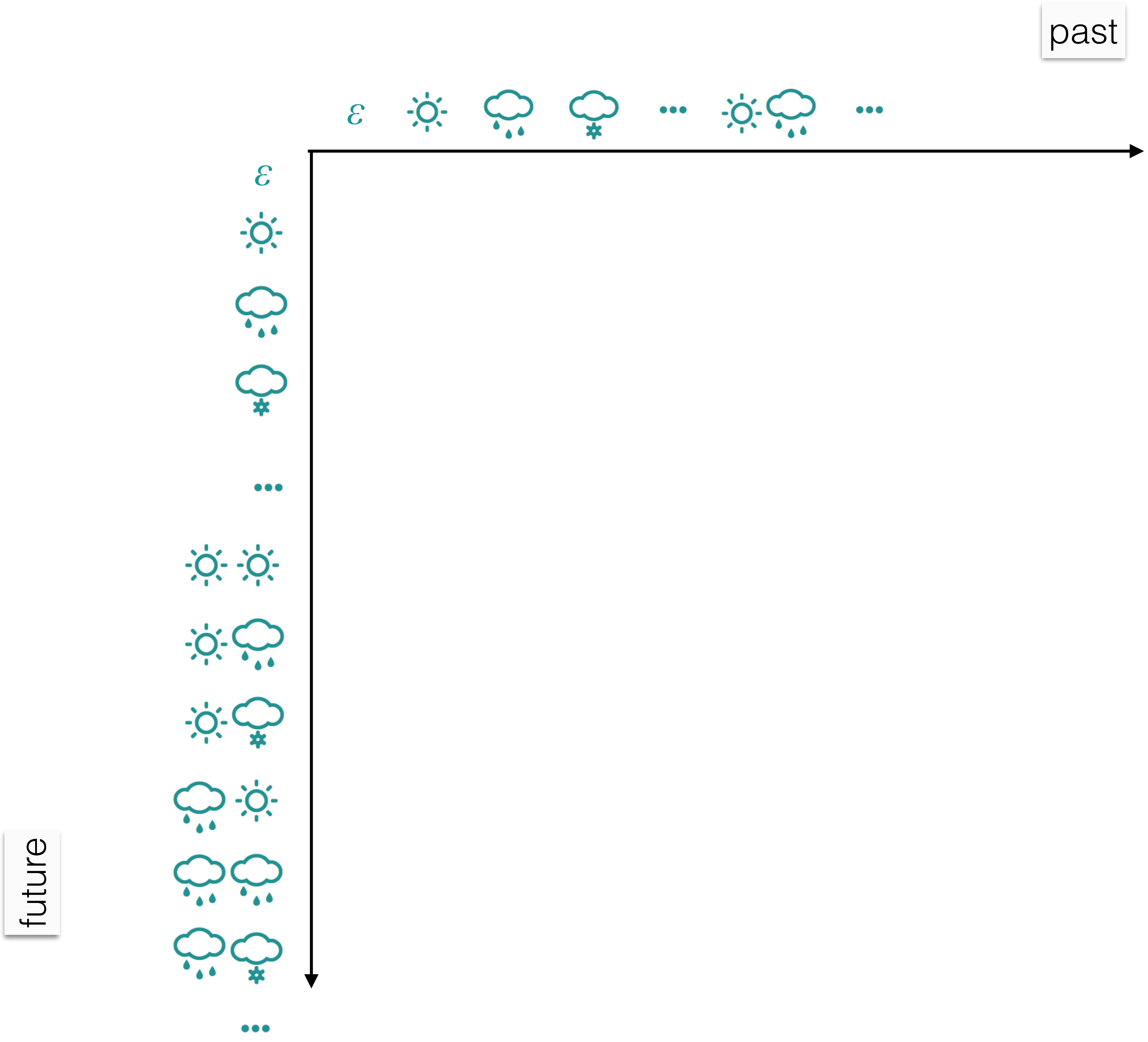

past
ε
ε
future
Pr( )

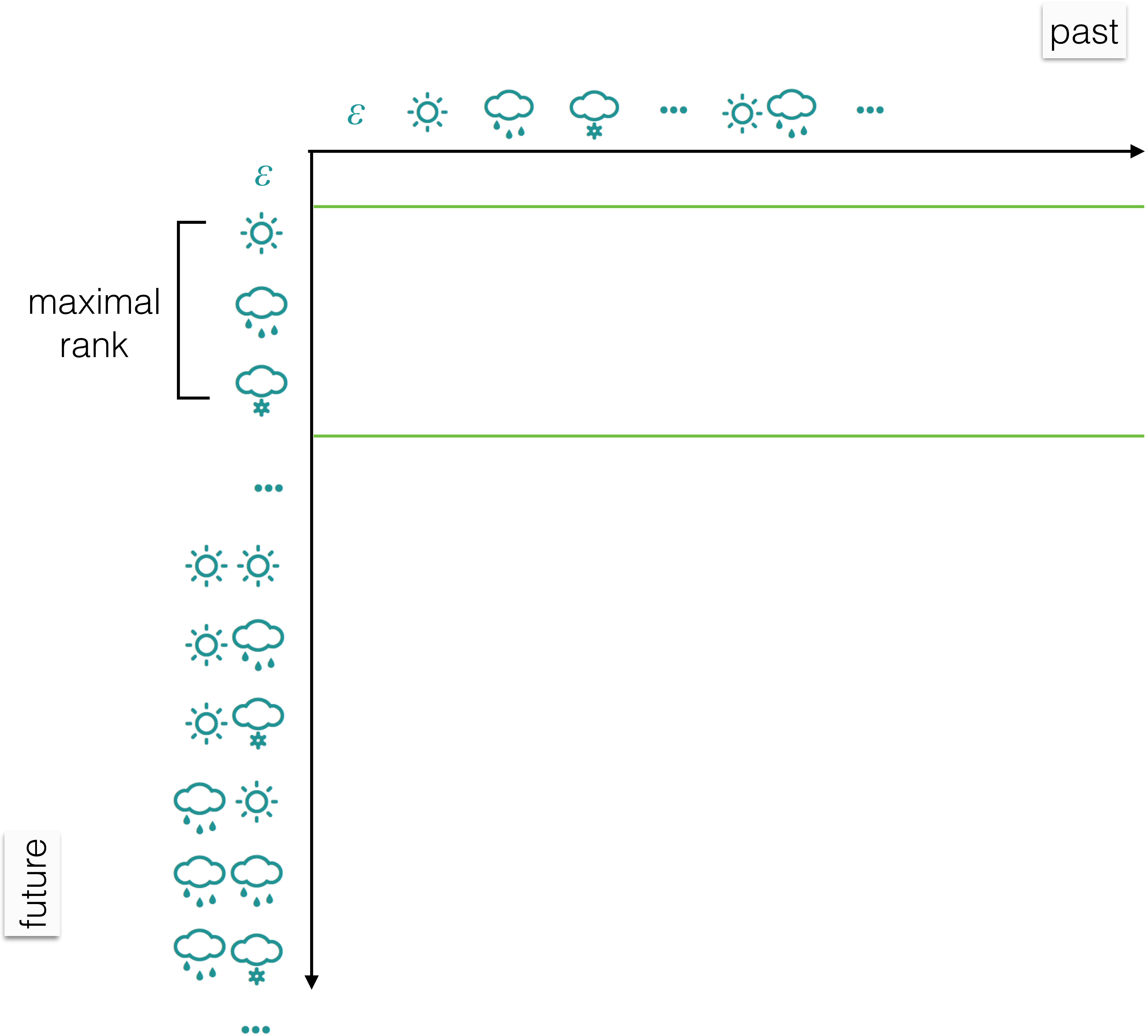

past
ε
Pr( )
ε
future

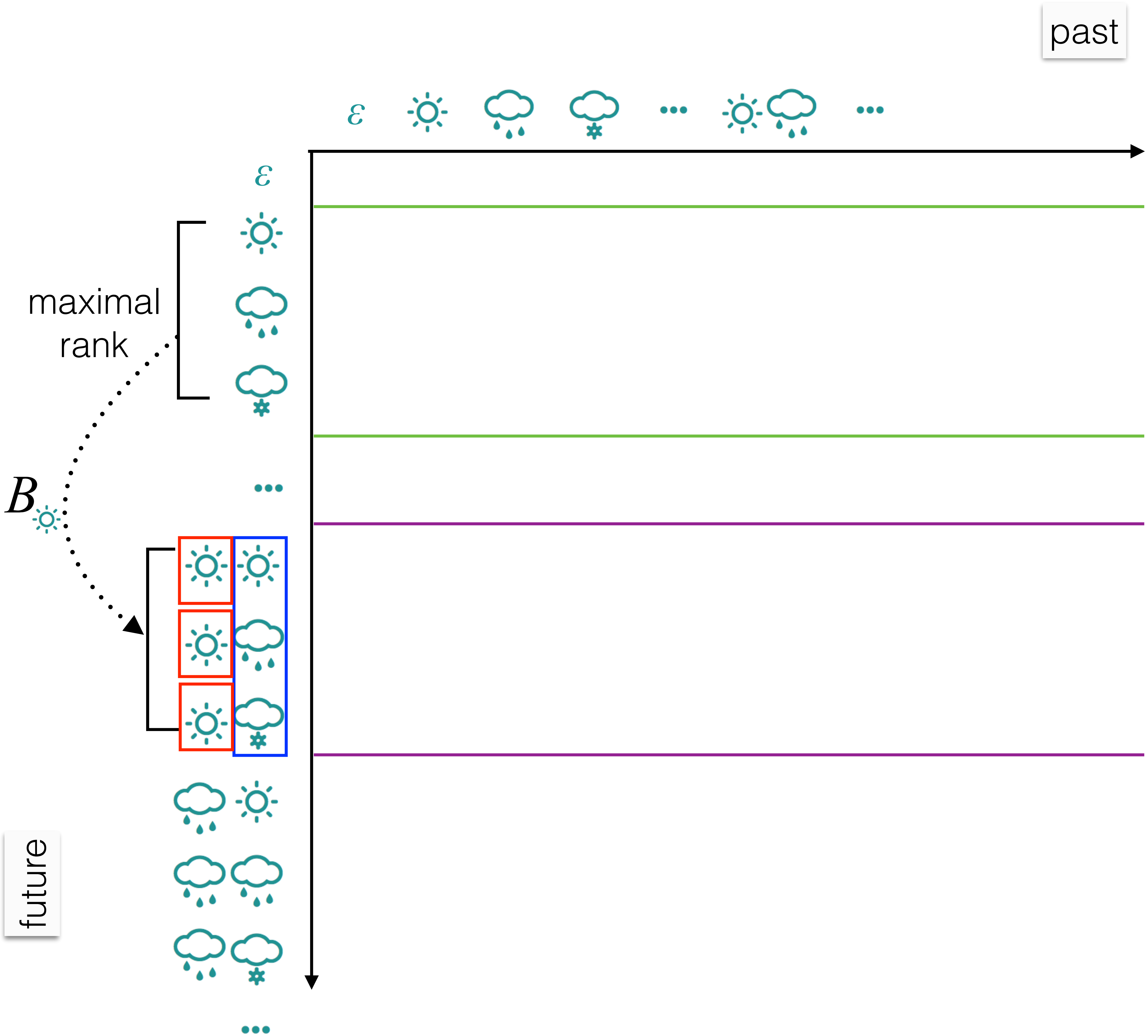

past
ε
future
Pr( )
Pr( )
Pr( )
Pr( )



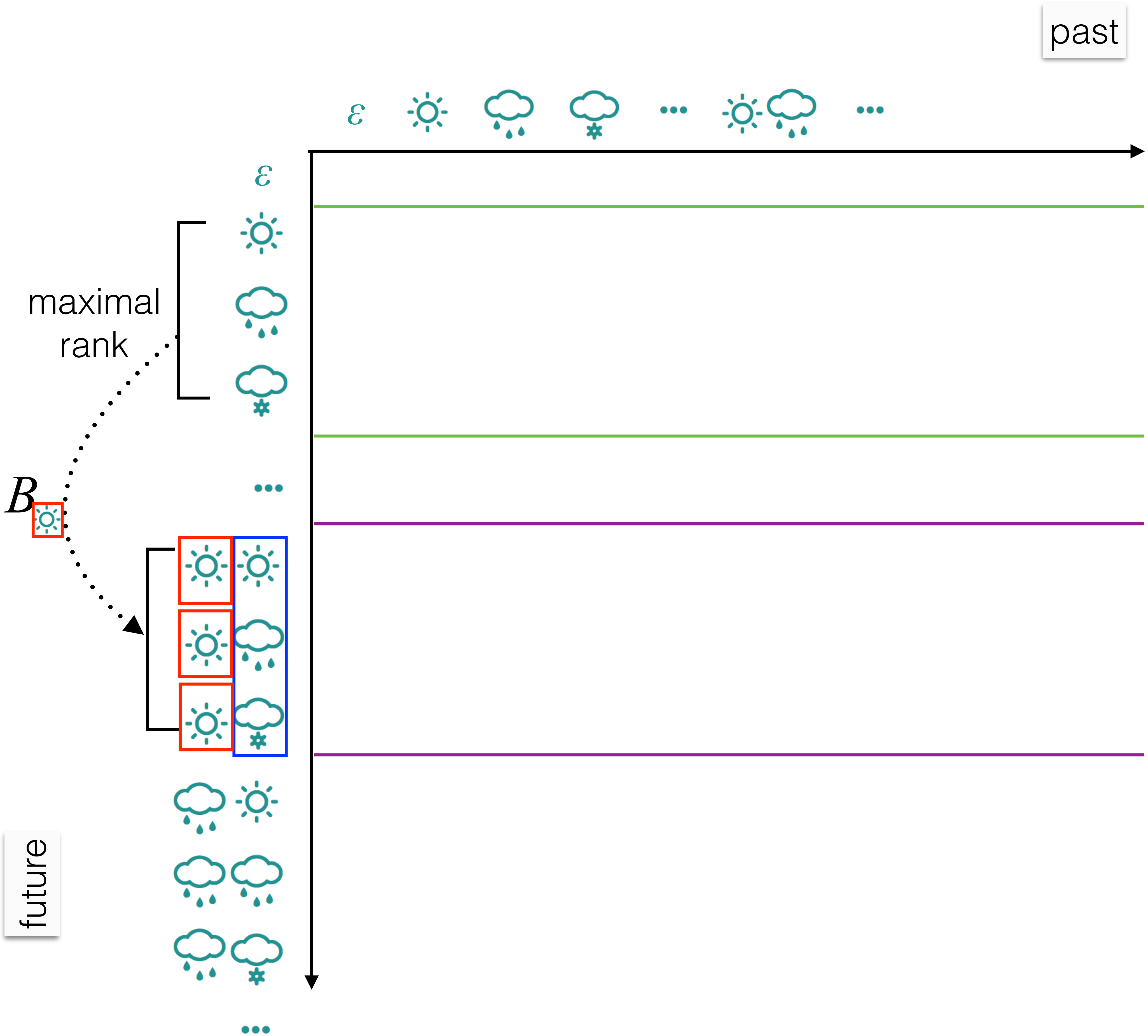
The SDM M is a Hankel matrix


past
ε
ε
future

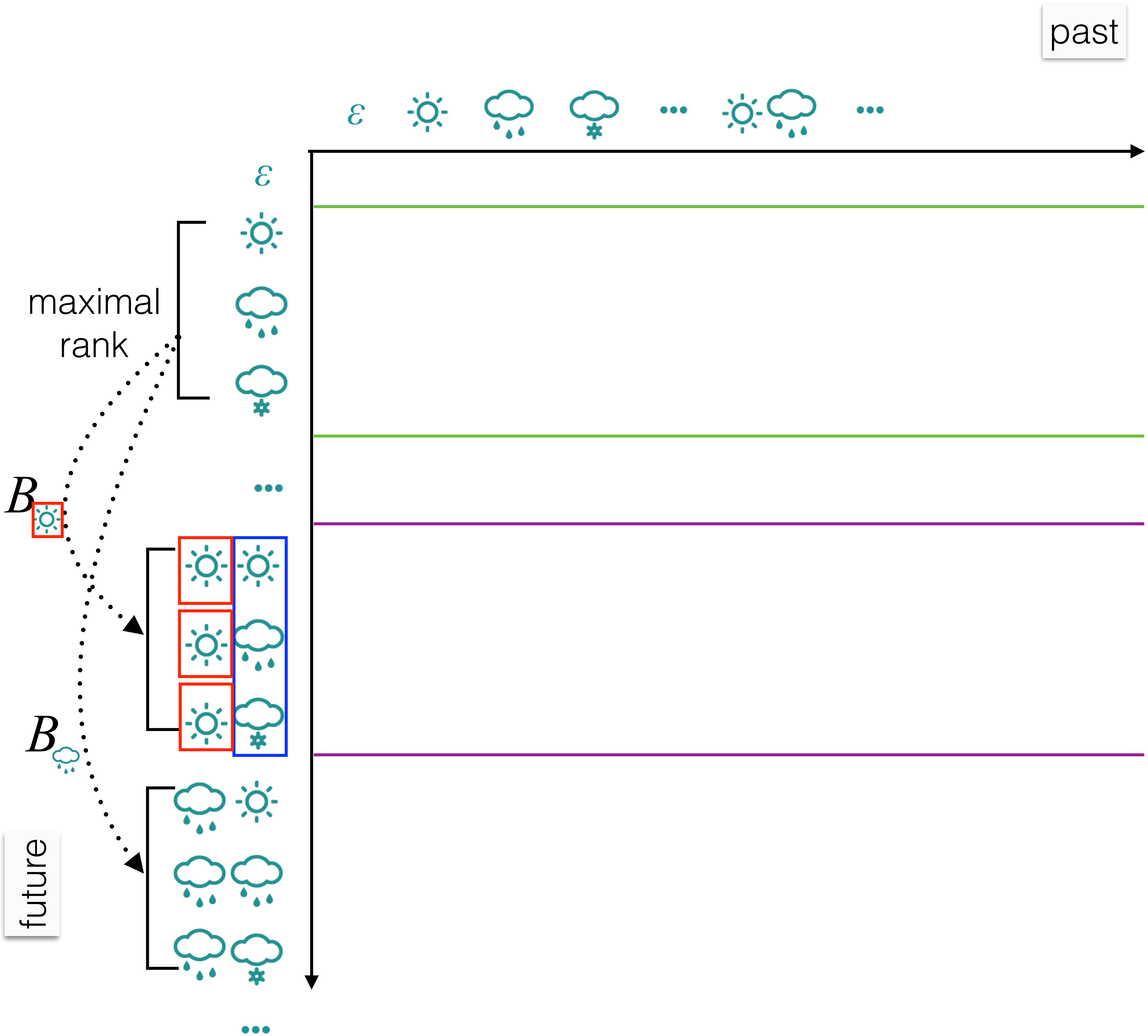

past
ε
...
...
ε
maximal rank
...
future
...

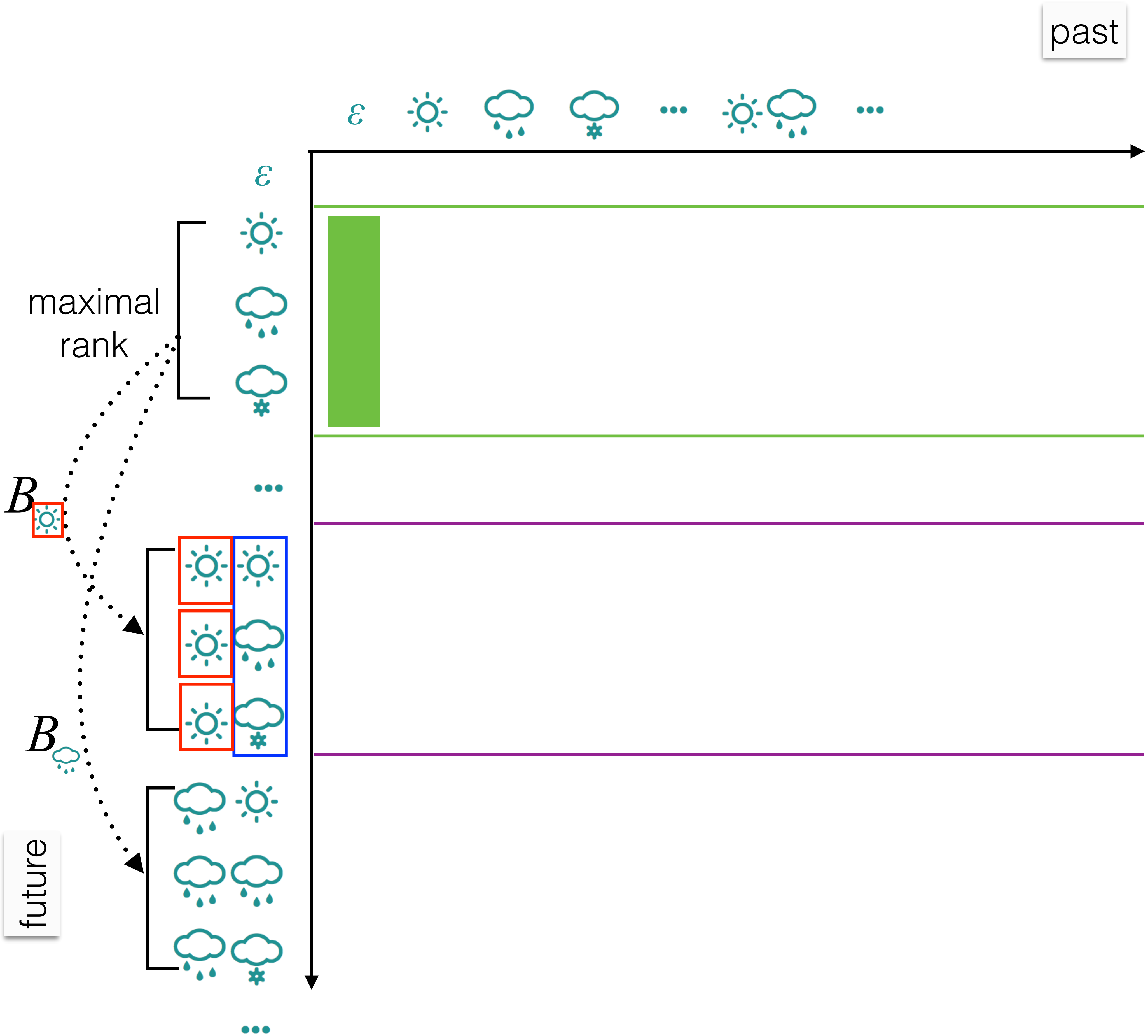


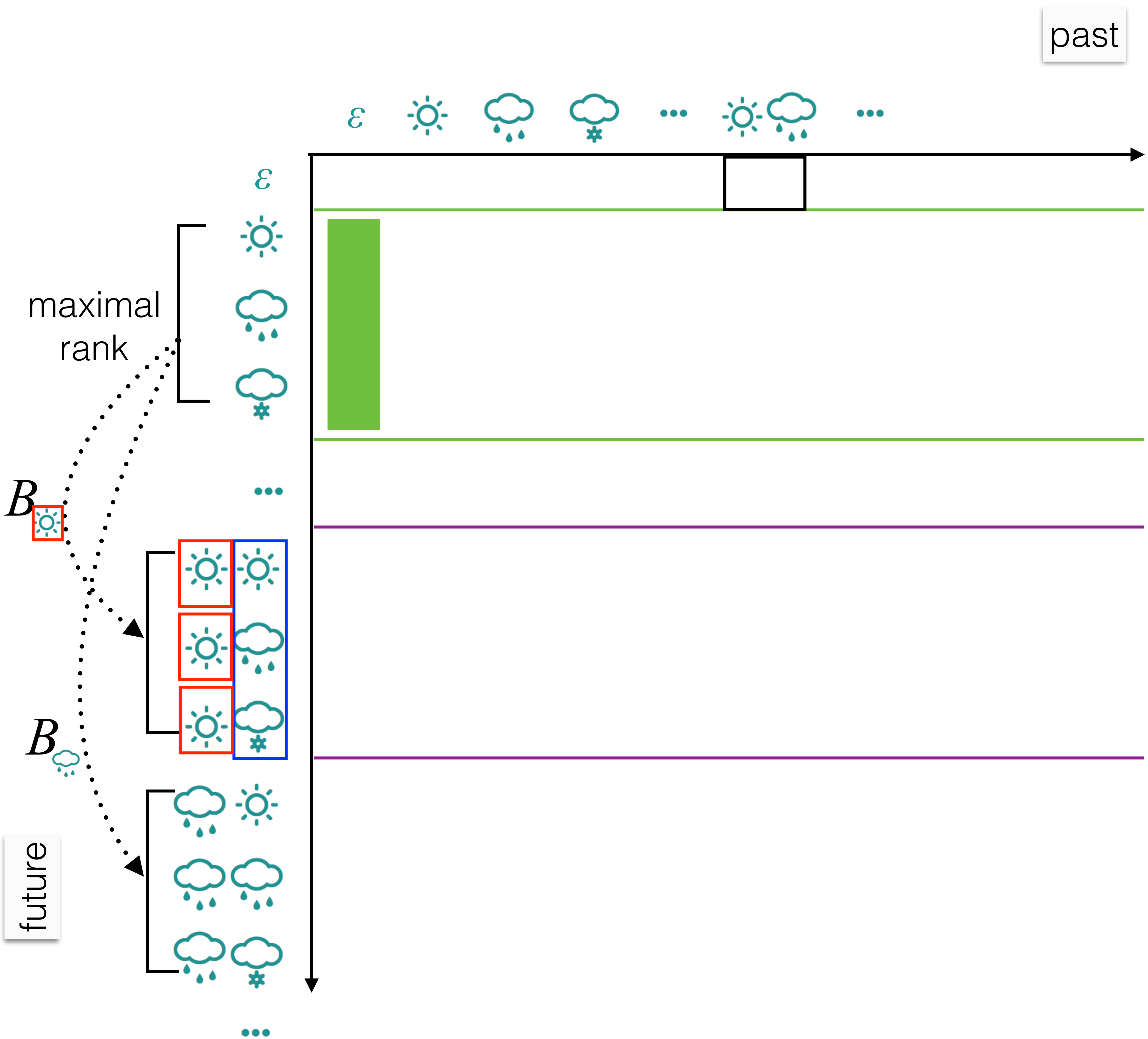
past
ε
maximal rank
$B_{☼}$
future



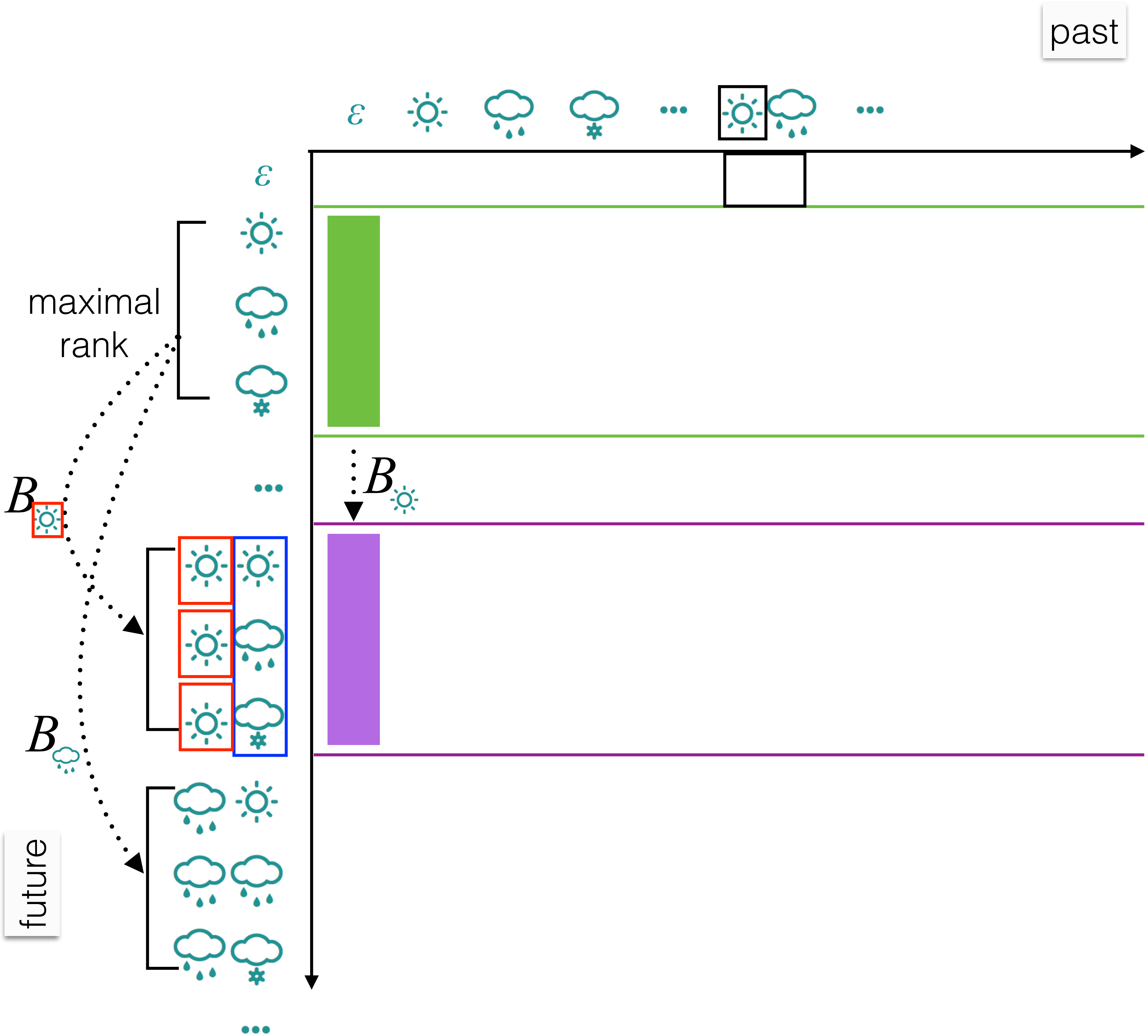
past
ε
ε
maximal
rank
B
future


past
ε
ε
maximal
rank
B
B
future

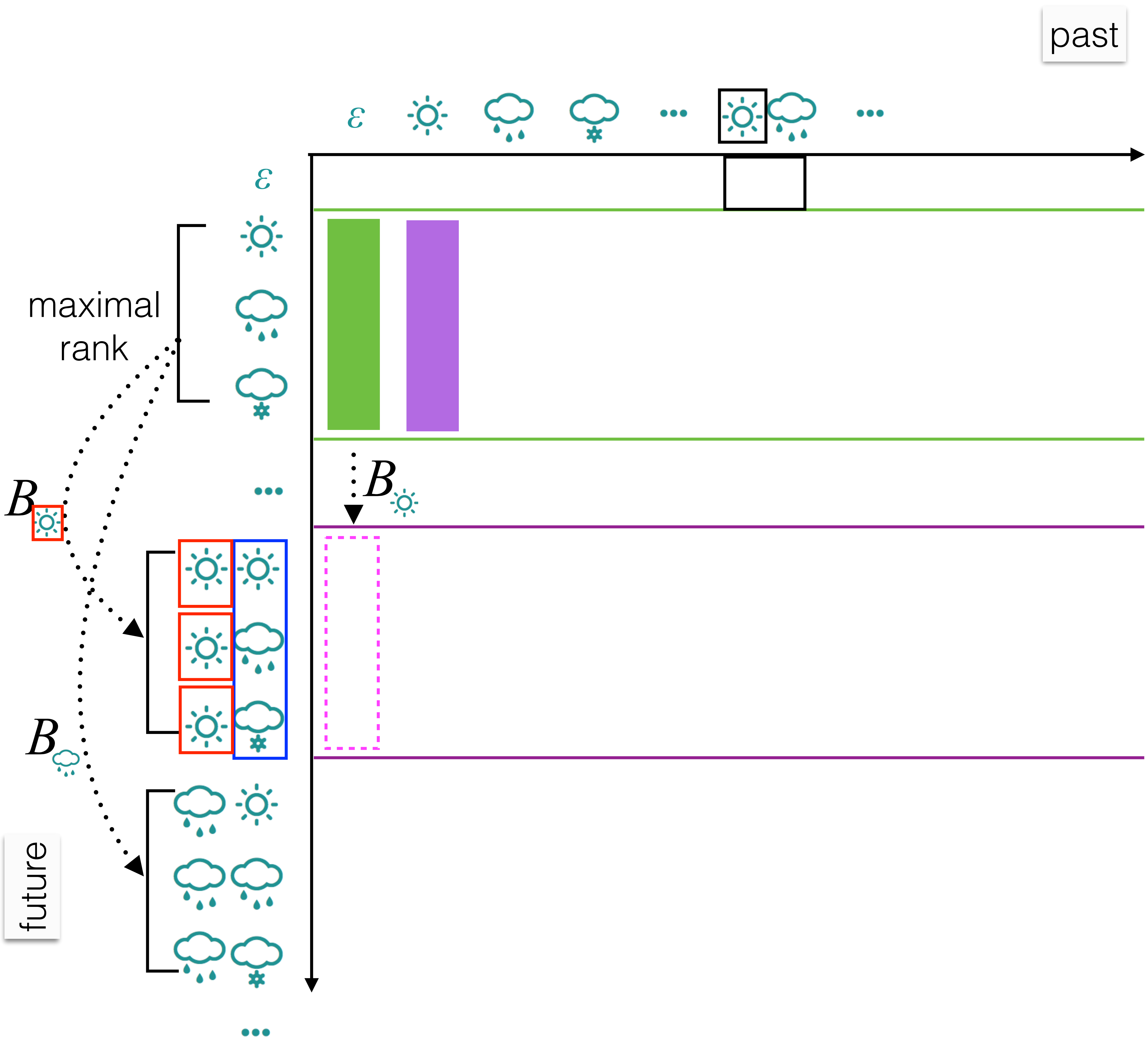

past
ε
ε
maximal
rank
B
B
future



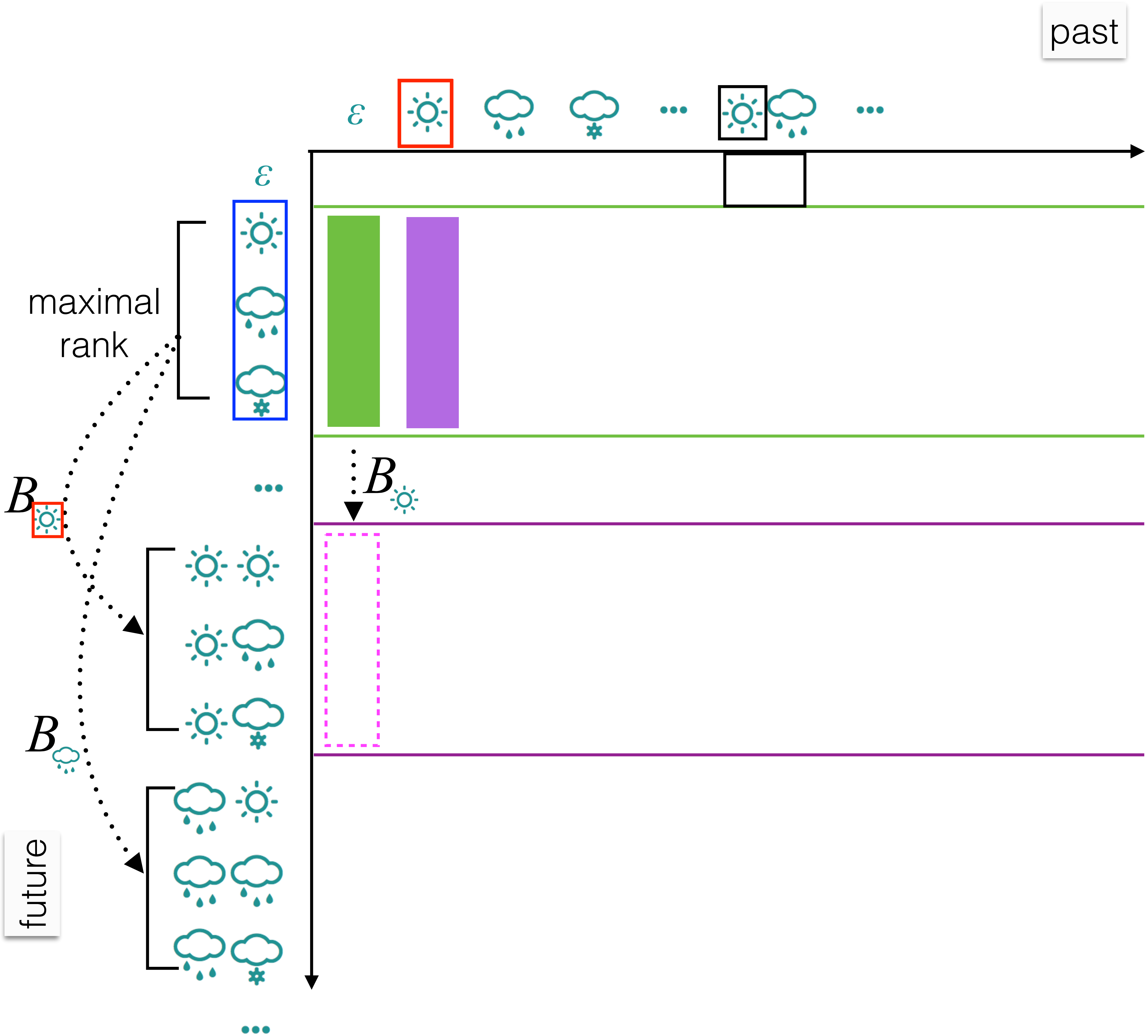
past
ε
ε
maximal rank
B
B
future



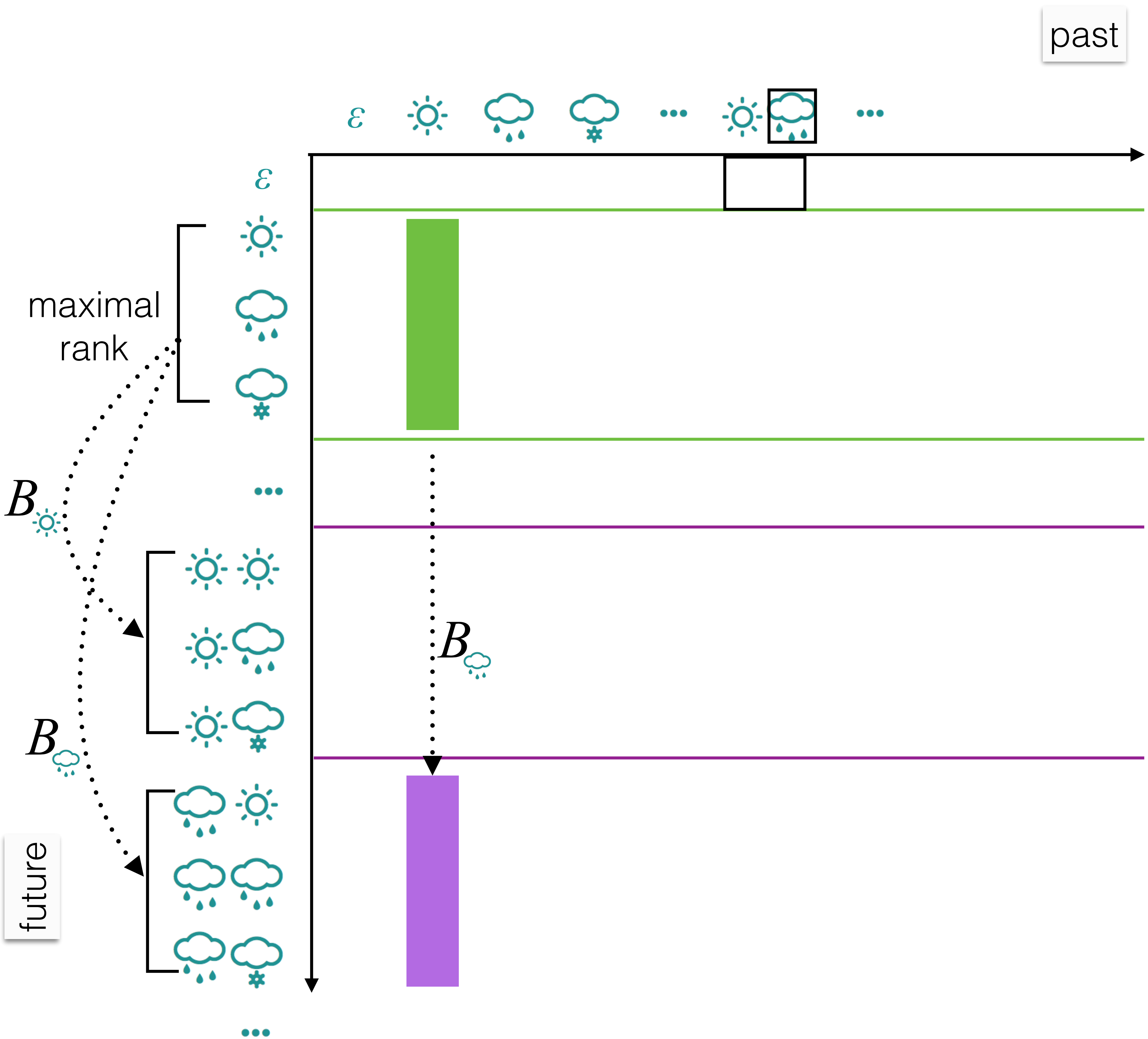
past
ε
ε
maximal rank
B
B
B
future

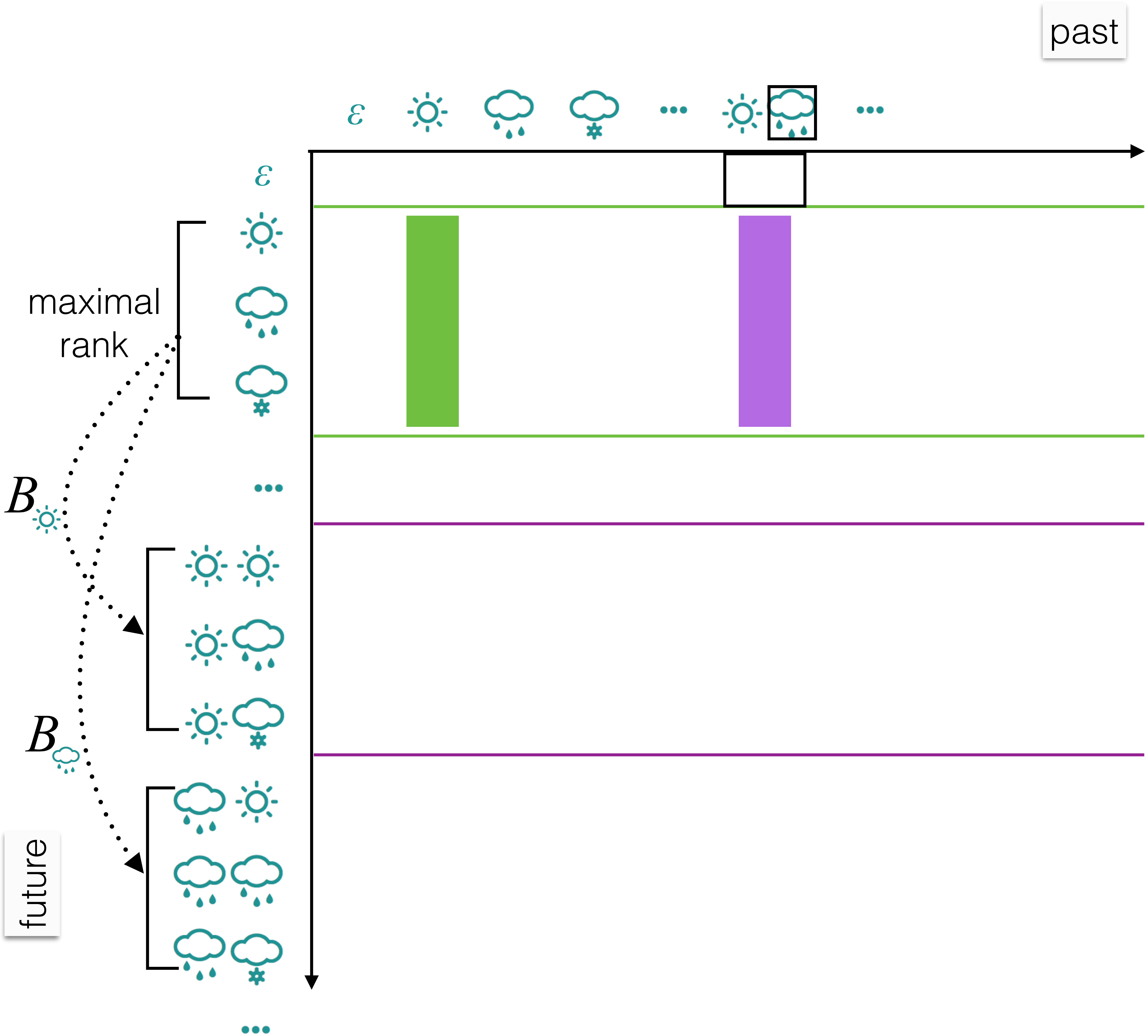


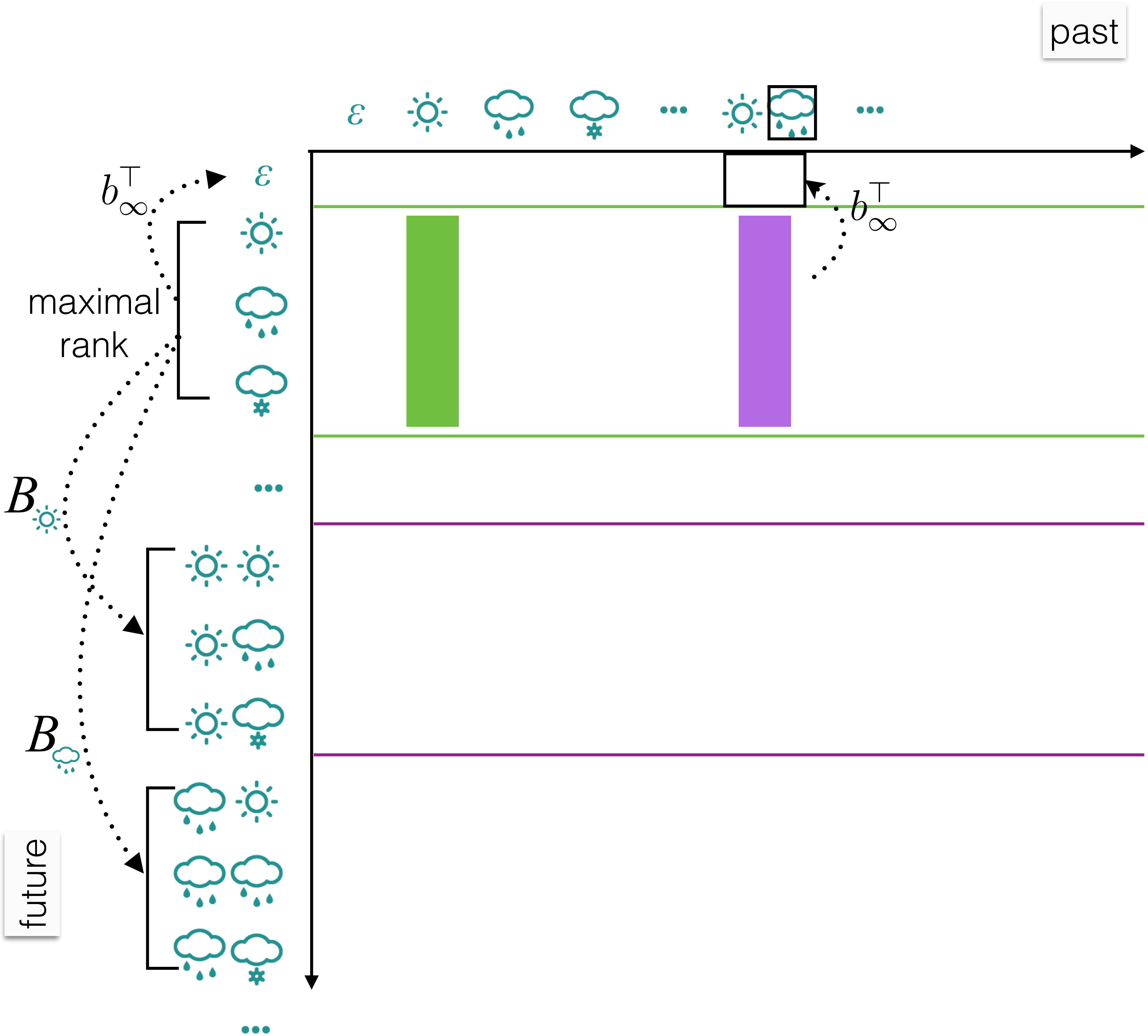
past
ε
ε
maximal rank
B
B
B
future



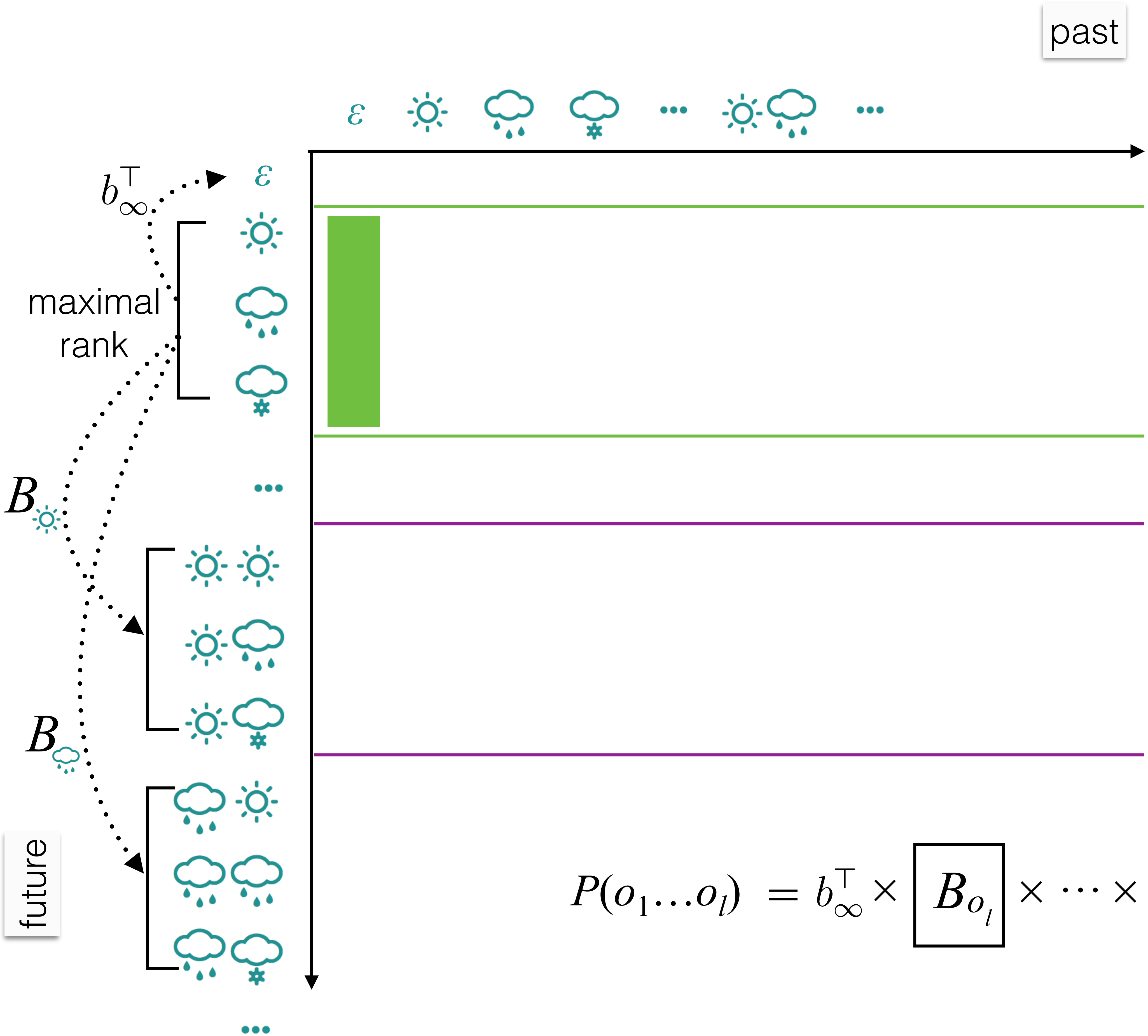
past
ε
ε
maximal rank
B
B
B
future


past
ε
ε
maximal rank
B
B
B
future

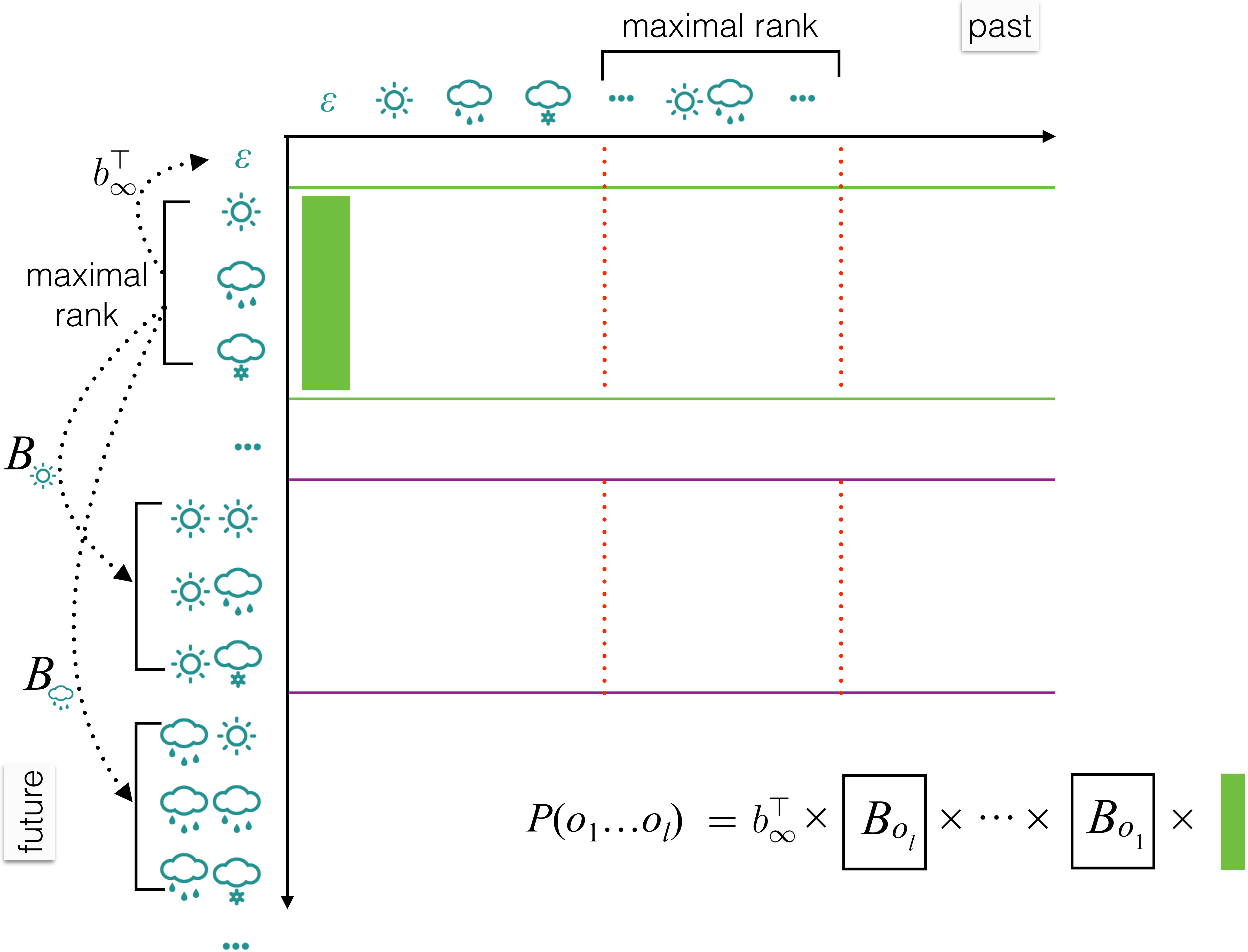

past
ε
ε
maximal rank
B
B
future

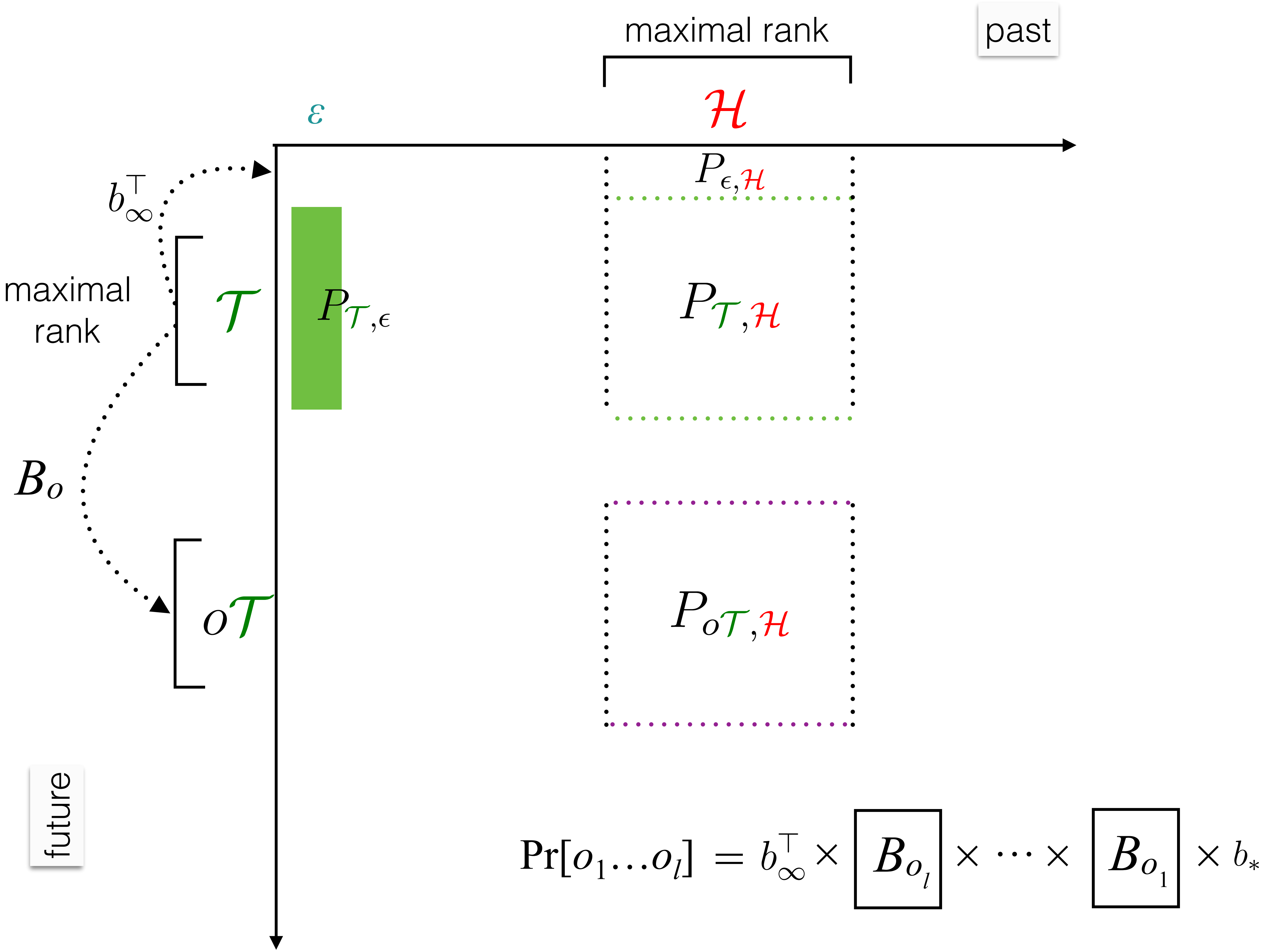


past
ε
$b_\infty^\top$
ε
$b_\infty^\top$
maximal rank
$B_{☼}$
$B_{☁}$
future

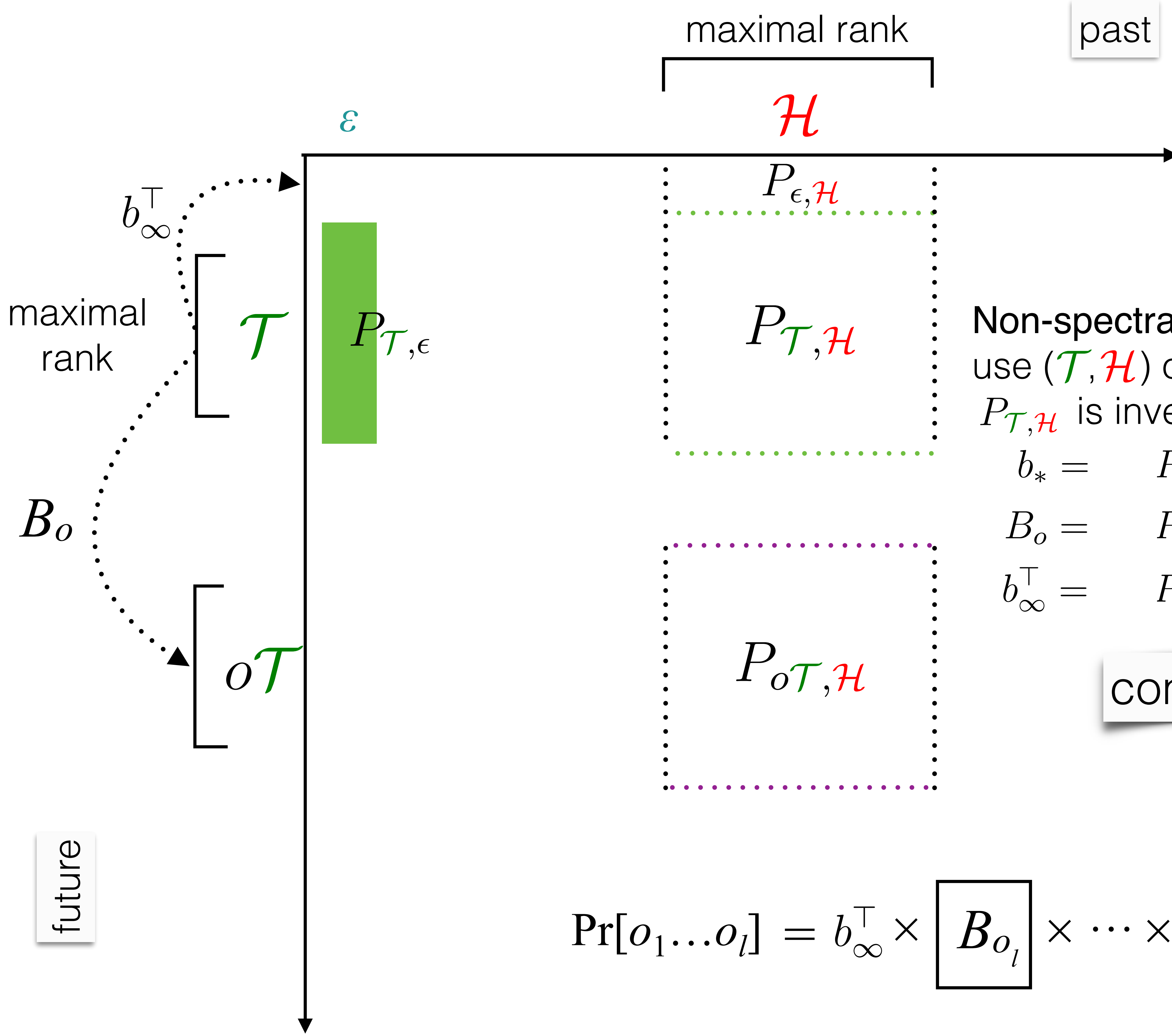


$$P(o_1 \ldots o_l) = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times$$


maximal rank
past
ε
$b_\infty^\top$
maximal rank
B
B
future
$P(o_1 \ldots o_l) = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times$



past
maximal rank
$\mathcal{H}$
ε
$P_{\epsilon,\mathcal{H}}$
$b_\infty^\top$
maximal rank
$\mathcal{T}$
$P_{\mathcal{T},\epsilon}$
$P_{\mathcal{T},\mathcal{H}}$
B_o
$o\mathcal{T}$
$P_{o\mathcal{T},\mathcal{H}}$
future
$\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$



Non-spectral algorithm

use $(\mathcal{T},\mathcal{H})$ of just size so that $P_{\mathcal{T},\mathcal{H}}$ is invertible.

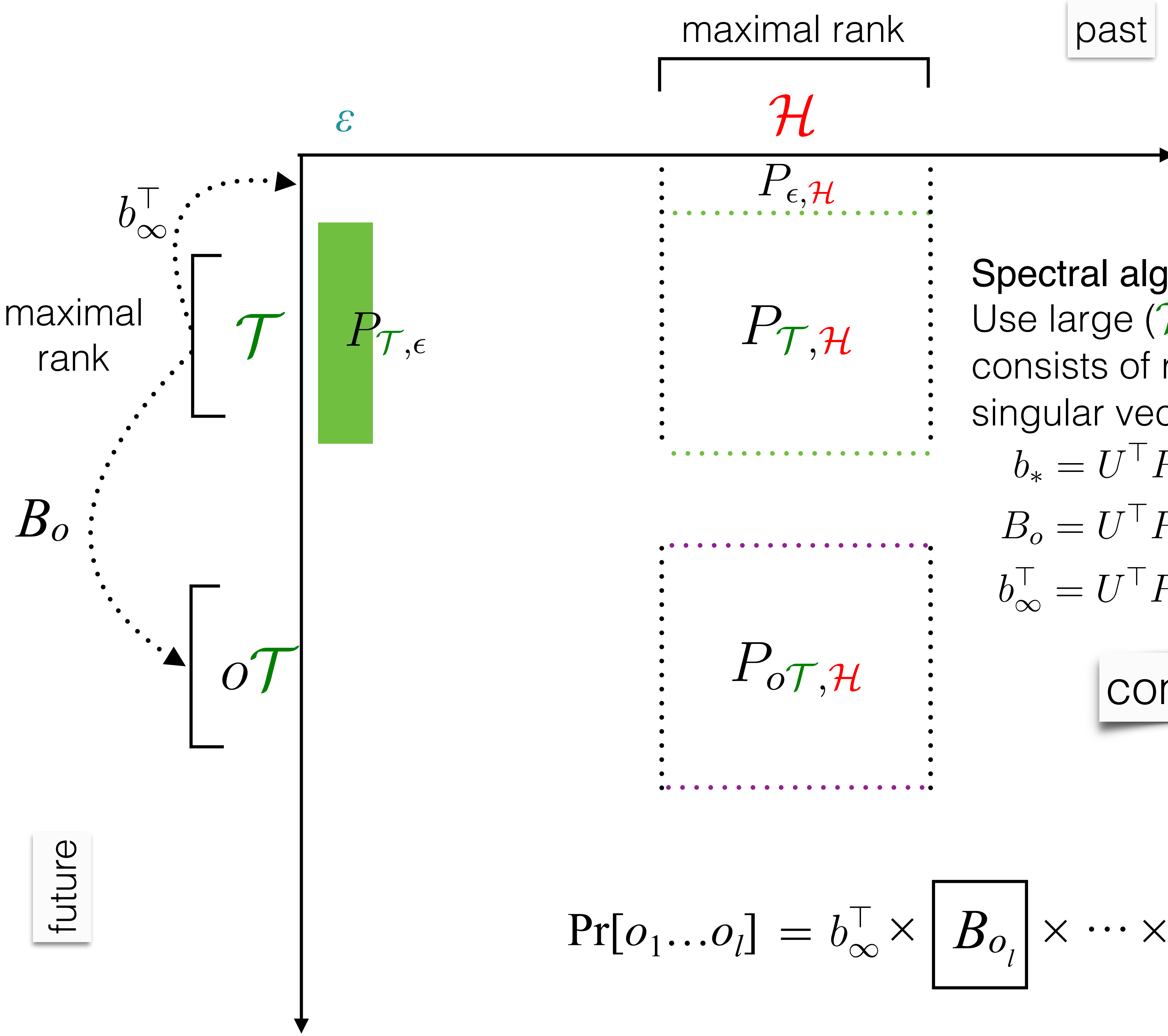
$$b_* = P_{\mathcal{T},\epsilon}$$

$$B_o = P_{o\mathcal{T},\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^{-1}$$

$$b_\infty^\top = P_{\epsilon,\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^{-1}$$

consistent

$$\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$$



Spectral algorithm

Use large $(\mathcal{T},\mathcal{H})$, and let U consists of rank(M) leading left singular vectors of $P_{\mathcal{T},\mathcal{H}}$

$$b_* = U^\top P_{\mathcal{T},\epsilon}$$

$$B_o = U^\top P_{o\mathcal{T},\mathcal{H}}(U^\top P_{\mathcal{T},\mathcal{H}})^+$$

$$b_\infty^\top = U^\top P_{\epsilon,\mathcal{H}}(U^\top P_{\mathcal{T},\mathcal{H}})^+$$

consistent

$$\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$$

The predictive interpretation

- The semantics of the state representation used in PSR: $P_{\mathcal{T}|h}$

The predictive interpretation

- The semantics of the state representation used in PSR: $P_{\mathcal{T}|h}$
  - Or its linear transformation $U^\intercal P_{\mathcal{T}|h}$

The predictive interpretation

- The semantics of the state representation used in PSR: $P_{\mathcal{T}|h}$
  - Or its linear transformation $U^{\intercal} P_{\mathcal{T}|h}$
  - Cond. prob. of a set of future events given the history h

The predictive interpretation

- The semantics of the state representation used in PSR: $P_{\mathcal{T}|h}$
  - Or its linear transformation $U^\intercal P_{\mathcal{T}|h}$
  - Cond. prob. of a set of future events given the history h
- Earlier question: what is the other trivial function that is always state???

The predictive interpretation

- The semantics of the state representation used in PSR: $P_{\mathcal{T}|h}$
  - Or its linear transformation $U^\intercal P_{\mathcal{T}|h}$
  - Cond. prob. of a set of future events given the history h
- Earlier question: what is the other trivial function that is always state???
- Answer: (exact) predictions of all future events is trivially state

The predictive interpretation

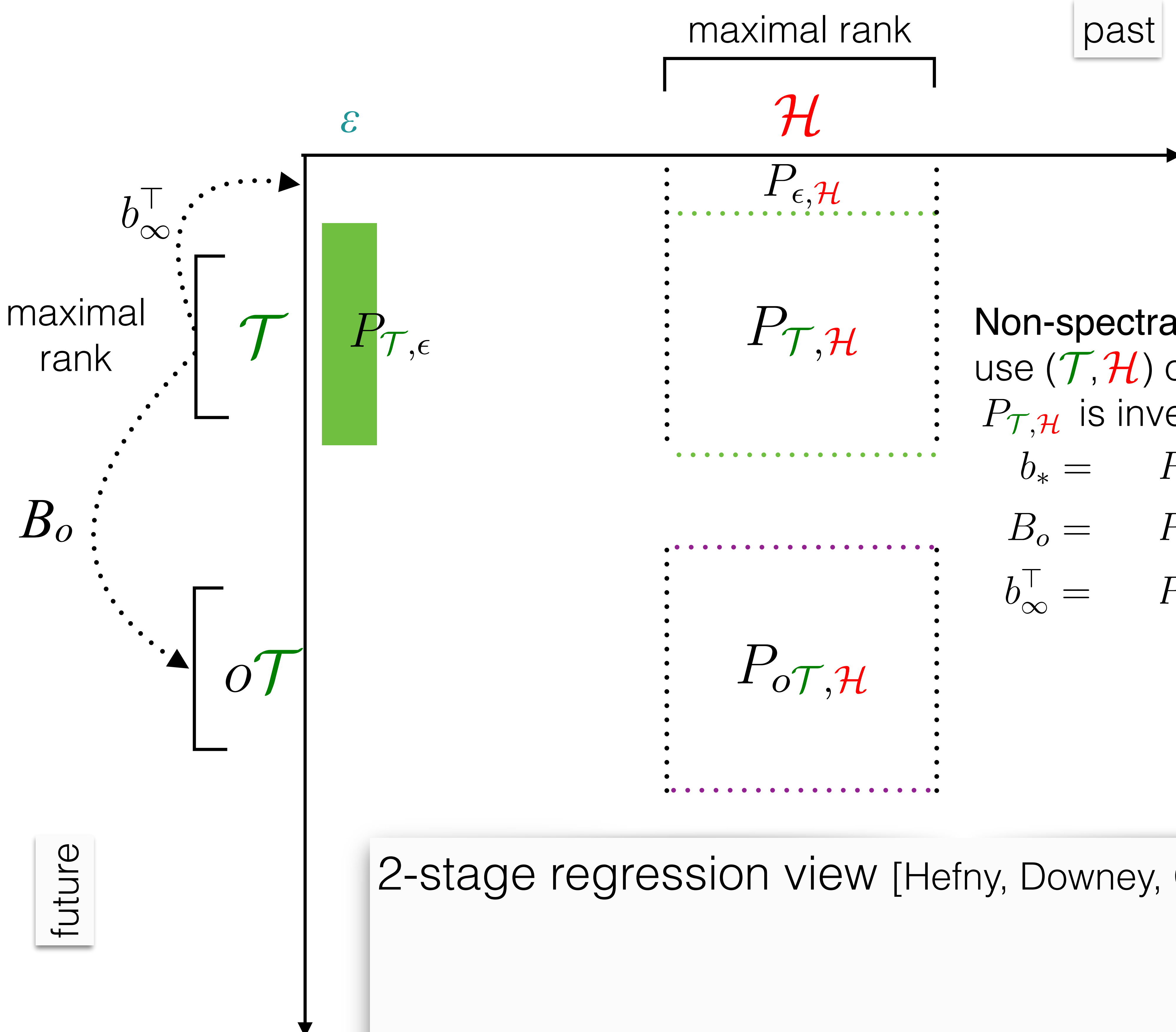
- The semantics of the state representation used in PSR: $P_{\mathcal{T}|h}$
  - Or its linear transformation $U^\intercal P_{\mathcal{T}|h}$
  - Cond. prob. of a set of future events given the history h
- Earlier question: what is the other trivial function that is always state???
- Answer: (exact) predictions of all future events is trivially state
- If $\phi(h) = \{\Pr[t' \mid h]\}_{t' \in O^*}$, then $\Pr[t \mid h] = \Pr[t \mid \phi(h)]$, trivially

The predictive interpretation

- The semantics of the state representation used in PSR: $P_{\mathcal{T}|h}$
  - Or its linear transformation $U^{\intercal} P_{\mathcal{T}|h}$
  - Cond. prob. of a set of future events given the history h
- Earlier question: what is the other trivial function that is always state???
- Answer: (exact) predictions of all future events is trivially state
- If $\phi(h) = \{\Pr[t' \mid h]\}_{t' \in O^*}$, then $\Pr[t \mid h] = \Pr[t \mid \phi(h)]$, trivially
- But this ϕ is infinite-dimensional and difficult to work with

The predictive interpretation

- The semantics of the state representation used in PSR: $P_{\mathcal{T}|h}$
  - Or its linear transformation $U^\intercal P_{\mathcal{T}|h}$
  - Cond. prob. of a set of future events given the history h
- Earlier question: what is the other trivial function that is always state???
- Answer: (exact) predictions of all future events is trivially state
- If $\phi(h) = \{\Pr[t' \mid h]\}_{t' \in O^*}$, then $\Pr[t \mid h] = \Pr[t \mid \phi(h)]$, trivially
- But this ϕ is infinite-dimensional and difficult to work with
- PSR: when system has certain low-rank structure, the infinite-dimensional object is uniquely determined by a subset of its coordinates, which is tractable.



Non-spectral algorithm

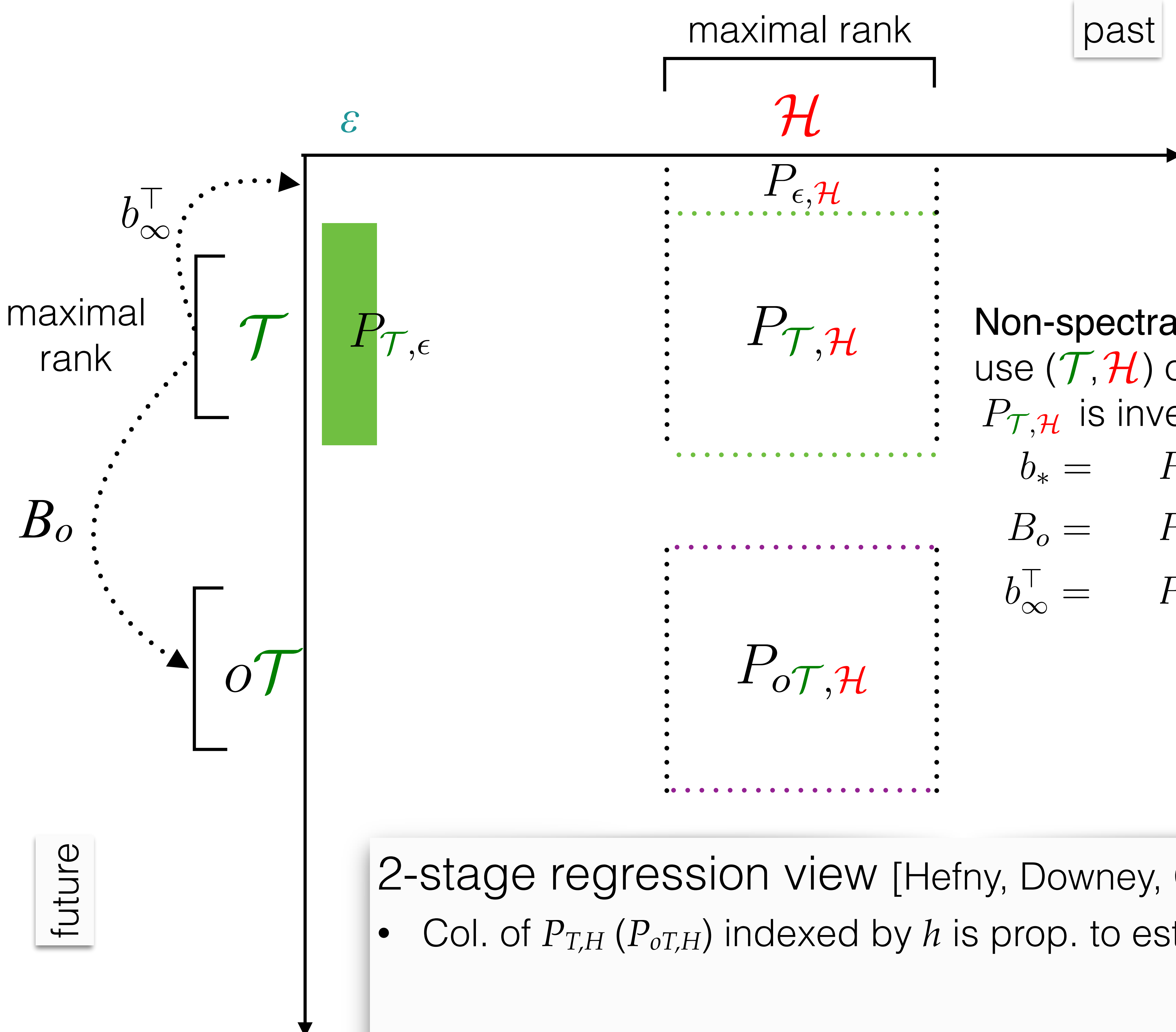
use $(\mathcal{T},\mathcal{H})$ of just size so that $P_{\mathcal{T},\mathcal{H}}$ is invertible.

$$b_* = P_{\mathcal{T},\epsilon}$$

$$B_o = P_{o\mathcal{T},\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^+$$

$$b_\infty^\top = P_{\epsilon,\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^+$$

2-stage regression view [Hefny, Downey, Gordon 2015]



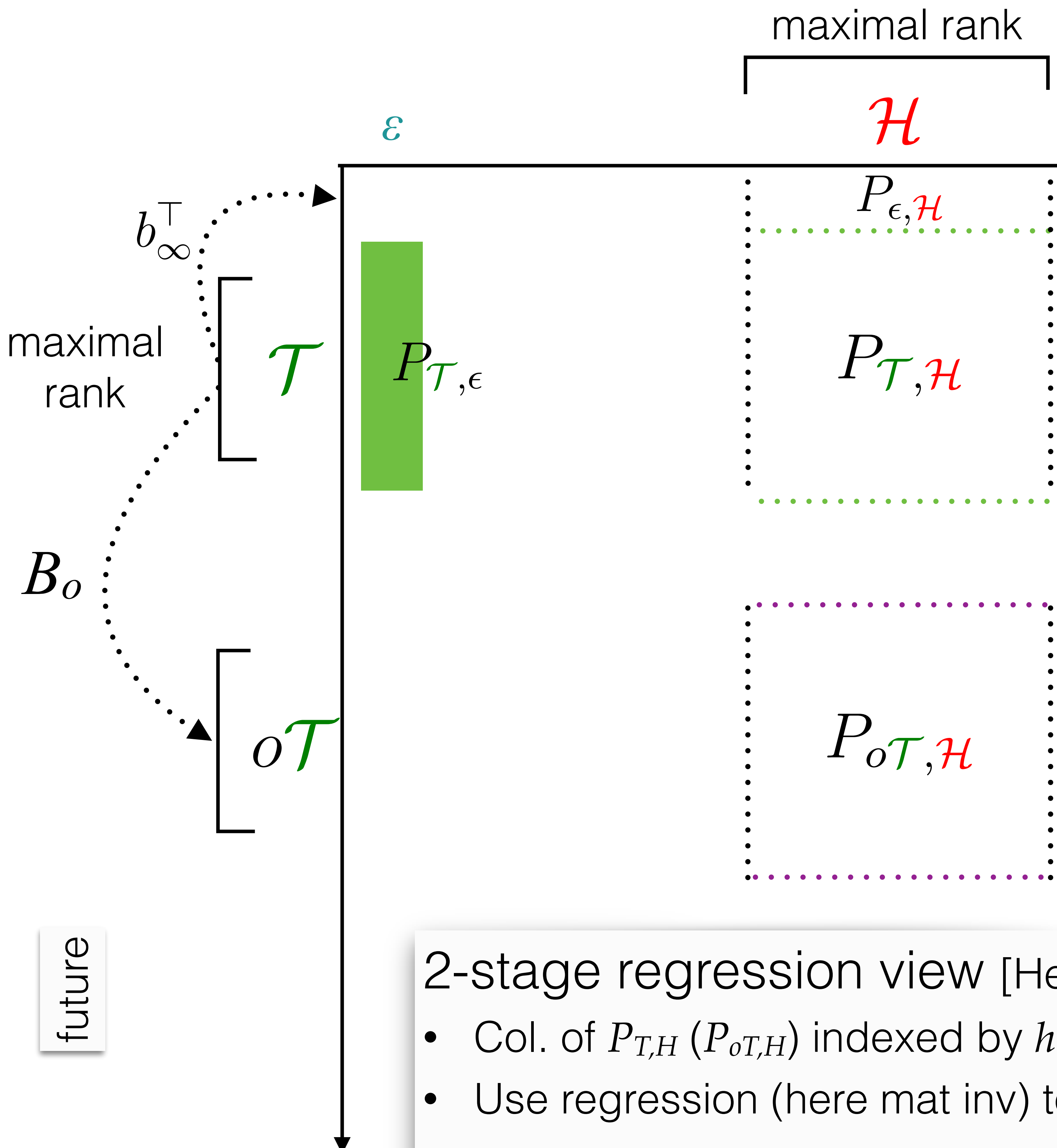
Non-spectral algorithm

use $(\mathcal{T},\mathcal{H})$ of just size so that $P_{\mathcal{T},\mathcal{H}}$ is invertible.

$$b_* = P_{\mathcal{T},\epsilon}$$
$$B_o = P_{o\mathcal{T},\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^+$$
$$b_\infty^\top = P_{\epsilon,\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^+$$

2-stage regression view [Hefny, Downey, Gordon 2015]

- Col. of $P_{T,H}$ ($P_{oT,H}$) indexed by h is prop. to estimated state of h (ho)



Non-spectral algorithm

use $(\mathcal{T},\mathcal{H})$ of just size so that $P_{\mathcal{T},\mathcal{H}}$ is invertible.

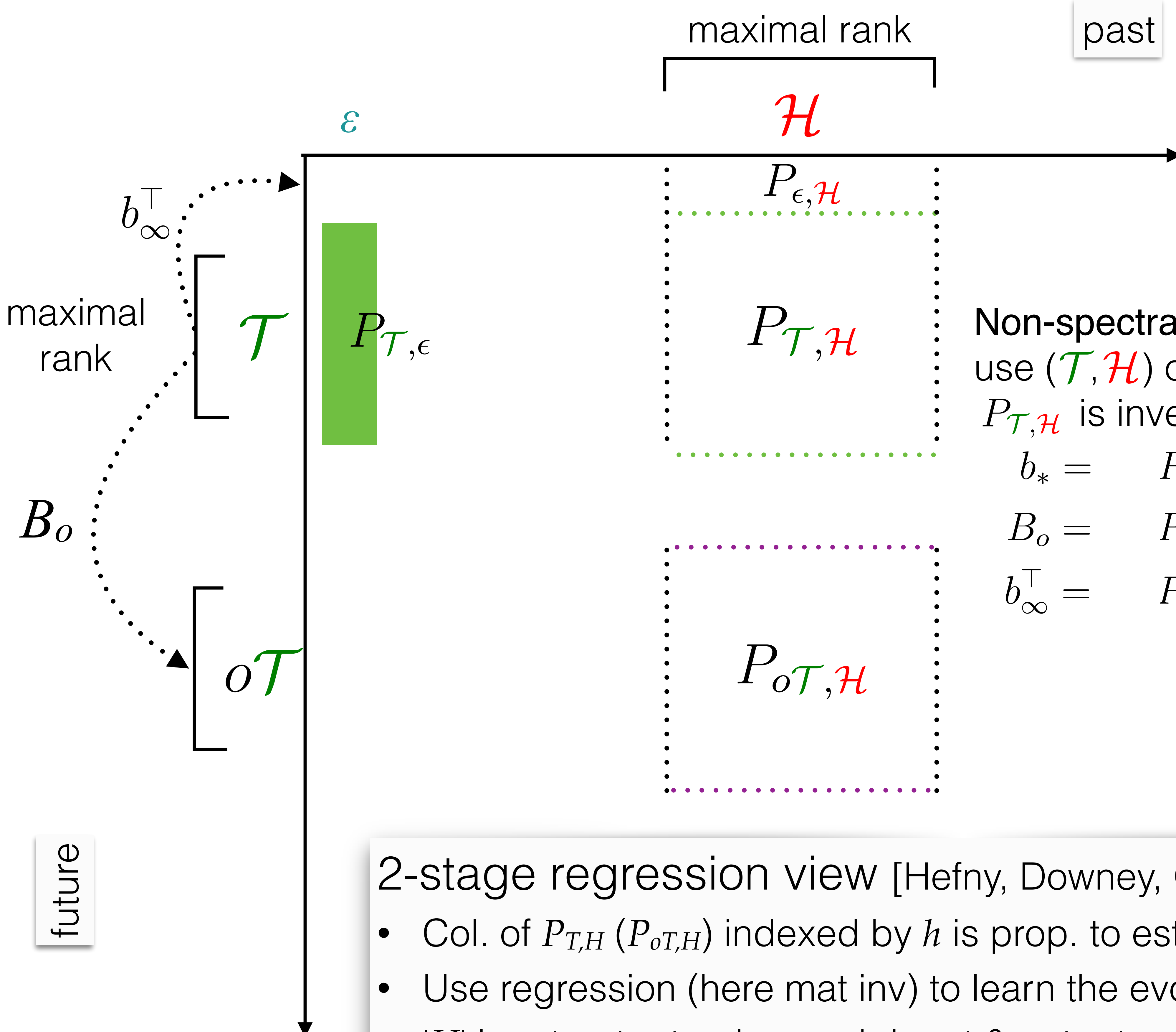
$$b_* = P_{\mathcal{T},\epsilon}$$

$$B_o = P_{o\mathcal{T},\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^+$$

$$b_\infty^\top = P_{\epsilon,\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^+$$

2-stage regression view [Hefny, Downey, Gordon 2015]

- Col. of $P_{T,H}$ ($P_{oT,H}$) indexed by h is prop. to estimated state of h (ho)
- Use regression (here mat inv) to learn the evolution of state given o



Non-spectral algorithm

use $(\mathcal{T},\mathcal{H})$ of just size so that $P_{\mathcal{T},\mathcal{H}}$ is invertible.

$$
\begin{aligned}
b_* &= P_{\mathcal{T},\epsilon} \\
B_o &= P_{o\mathcal{T},\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^+ \\
b_\infty^\top &= P_{\epsilon,\mathcal{H}}(P_{\mathcal{T},\mathcal{H}})^+
\end{aligned}
$$

2-stage regression view [Hefny, Downey, Gordon 2015]

- Col. of $P_{T,H}$ ($P_{oT,H}$) indexed by h is prop. to estimated state of h (ho)
- Use regression (here mat inv) to learn the evolution of state given o
- $|H|$ input-output pairs, each input & output are vectors in $R^{|T|}$

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$
- HMM can be converted into such a parametrization

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$
- HMM can be converted into such a parametrization
- For an HMM with transition T, emission E, initial dist. π,

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$
- HMM can be converted into such a parametrization
- For an HMM with transition T, emission E, initial dist. π,
  - $b_* = \pi$, $B_o = T\,\mathrm{diag}\{E[o \mid z^{(1)}], \ldots, E[o \mid z^{(|Z|)}]\}$, $b_\infty = \mathbf{1}$

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$
- HMM can be converted into such a parametrization
- For an HMM with transition T, emission E, initial dist. π,
  - $b_* = \pi$, $B_o = T\,\mathrm{diag}\{E[o \mid z^{(1)}], \ldots, E[o \mid z^{(|Z|)}]\}$, $b_\infty = \mathbf{1}$
- "Observable Operator Model (OOM)"

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$
- HMM can be converted into such a parametrization
- For an HMM with transition T, emission E, initial dist. π,
  - $b_* = \pi$, $B_o = T\,\mathrm{diag}\{E[o \mid z^{(1)}], \ldots, E[o \mid z^{(|Z|)}]\}$, $b_\infty = \mathbf{1}$
- "Observable Operator Model (OOM)"
  - informally, PSRs without a predictive semantics

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$
- HMM can be converted into such a parametrization
- For an HMM with transition T, emission E, initial dist. π,
  - $b_* = \pi$, $B_o = T\ \mathbf{diag}\{E[o \mid z^{(1)}], \ldots, E[o \mid z^{(|Z|)}]\}$, $b_\infty = \mathbf{1}$
- "Observable Operator Model (OOM)"
  - informally, PSRs without a predictive semantics
  - …Really?

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$
- HMM can be converted into such a parametrization
- For an HMM with transition T, emission E, initial dist. π,
  - $b_* = \pi$, $B_o = T\, \mathrm{diag}\{E[o \mid z^{(1)}], \ldots, E[o \mid z^{(|Z|)}]\}$, $b_\infty = \mathbf{1}$
- "Observable Operator Model (OOM)"
  - informally, PSRs without a predictive semantics
  - …Really?
  - (informally) any state representation that can predict $\Pr[o' \mid h]$ using a linear rule is a (transformed) PSR! (see appendix of my NeurIPS paper this year)

Connections to HMMs

- Recall $\Pr[o_1 \ldots o_l] = b_\infty^\top \times \boxed{B_{o_l}} \times \cdots \times \boxed{B_{o_1}} \times b_*$
- HMM can be converted into such a parametrization
- For an HMM with transition T, emission E, initial dist. π,
  - $b_* = \pi$, $B_o = T\, \mathrm{diag}\{E[o \mid z^{(1)}], \ldots, E[o \mid z^{(|Z|)}]\}$, $b_\infty = \mathbf{1}$
- "Observable Operator Model (OOM)"
  - informally, PSRs without a predictive semantics
  - …Really?
  - (informally) any state representation that can predict $\Pr[o' \mid h]$ using a linear rule is a (transformed) PSR! (see appendix of my NeurIPS paper this year)
  - Also known under the name Weighted Finite Automata (WFA)

Example: Markov Chain

Let f be the one-hot encoding of the last observation for an MC. Assume the transition matrix of the MC, T, is invertible. Define $\mathcal{T}$ as the set of length-1 sequences, then .

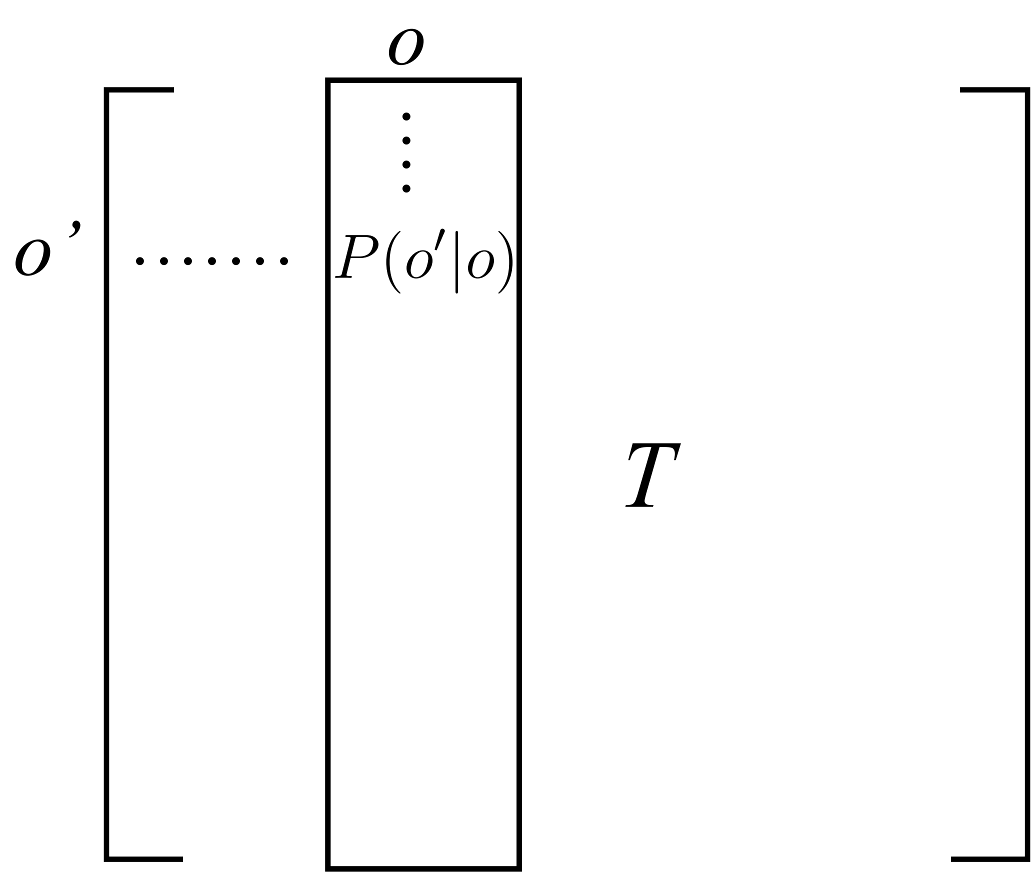
$$f(h) = T^{-1} P_{\mathcal{T}|h}$$

$$\begin{array}{cc} & o \\ o' & \begin{bmatrix} & \vdots & & \\ \cdots\cdots & P(o'|o) & & \\ & & T & \\ & & & \end{bmatrix} \end{array}$$

Example: Markov Chain

Let f be the one-hot encoding of the last observation for an MC. Assume the transition matrix of the MC, T, is invertible. Define $\color{green}{\mathcal{T}}$ as the set of length-1 sequences, then .

$$f(h) = T^{-1} P_{\color{green}{\mathcal{T}}|h}$$



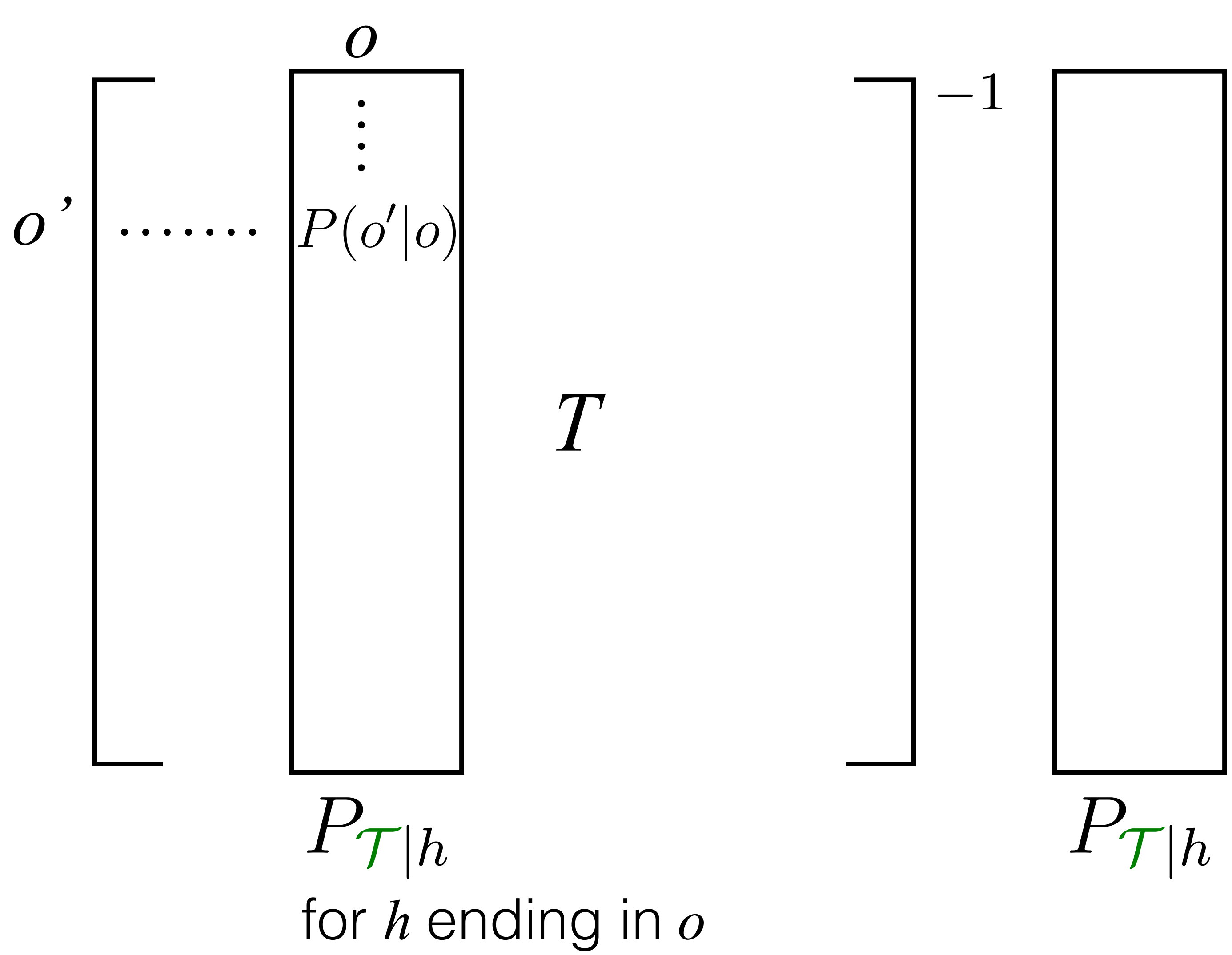
$P_{\color{green}{\mathcal{T}}|h}$

for h ending in o

Example: Markov Chain

Let f be the one-hot encoding of the last observation for an MC. Assume the transition matrix of the MC, T, is invertible. Define $\mathcal{T}$ as the set of length-1 sequences, then .

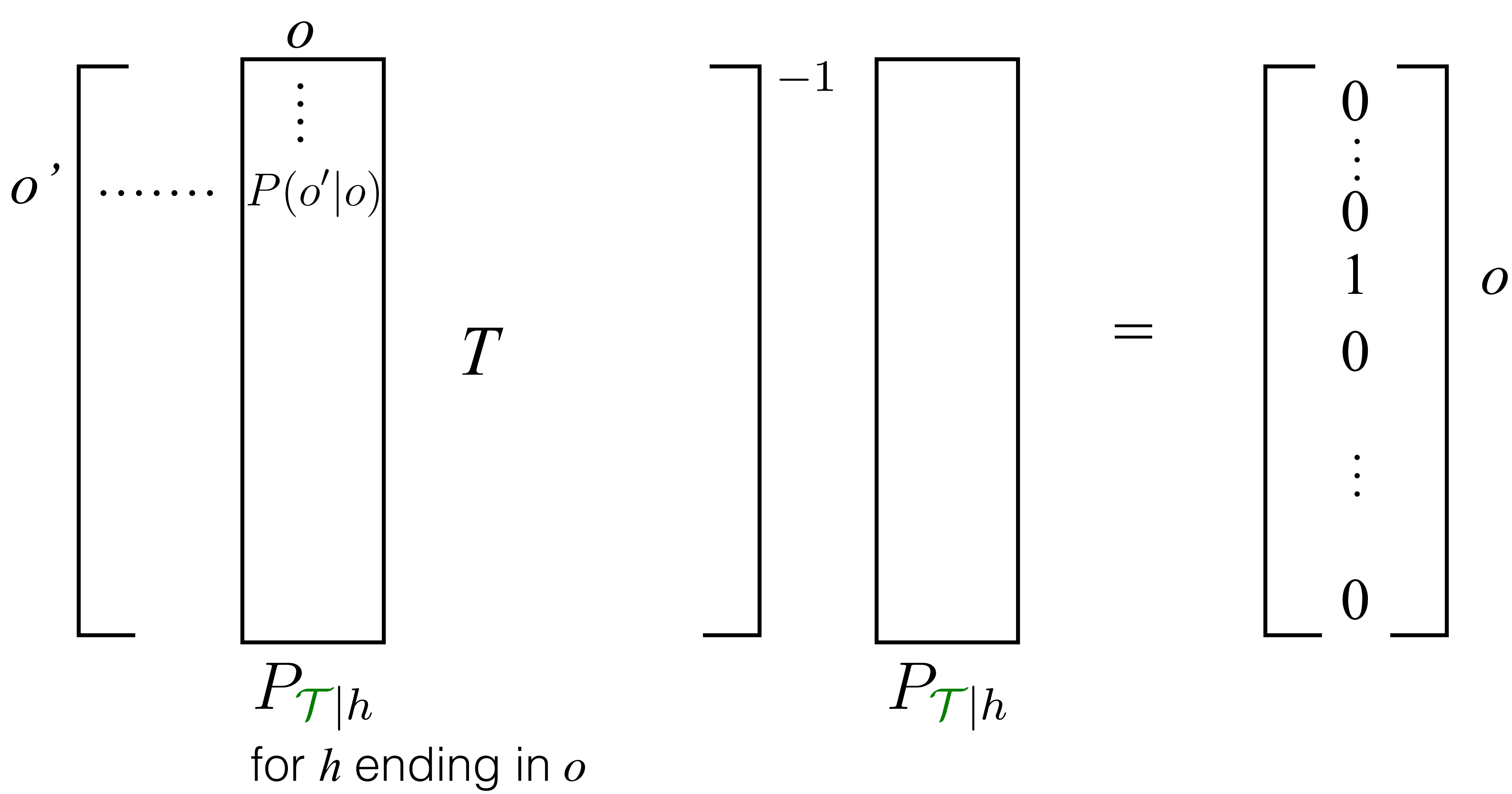
$$f(h) = T^{-1} P_{\mathcal{T}|h}$$



Example: Markov Chain

Let f be the one-hot encoding of the last observation for an MC. Assume the transition matrix of the MC, T, is invertible. Define $\mathcal{T}$ as the set of length-1 sequences, then .

$$f(h) = T^{-1} P_{\mathcal{T}|h}$$



What systems fall in PSRs \ HMMs?

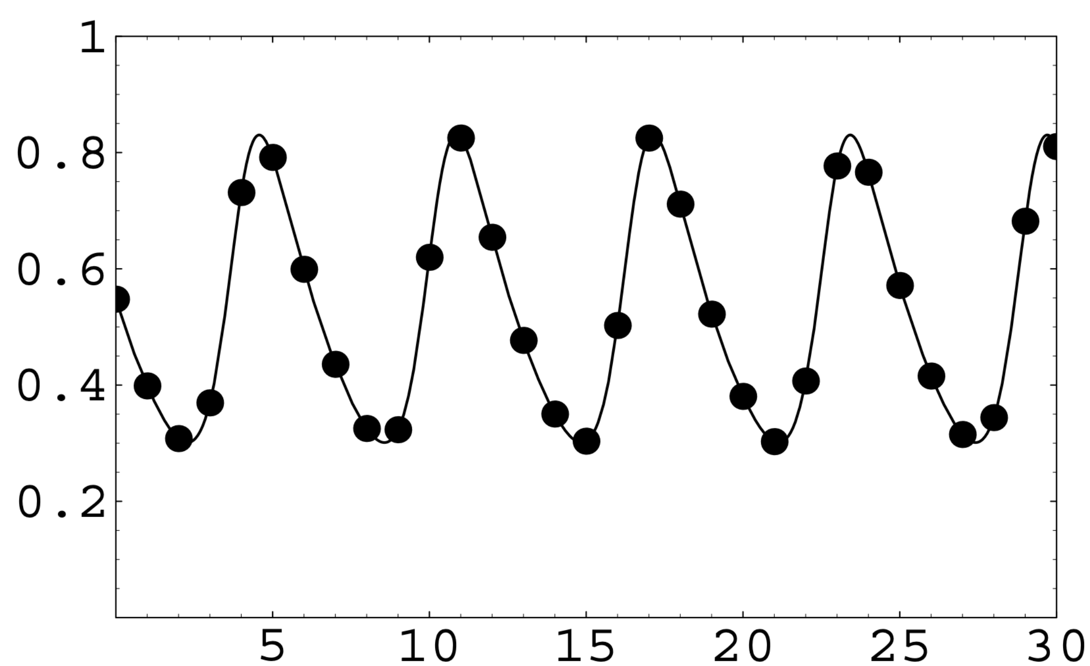
- Recall that HMMs with n states has an SDM with rank $\leq n$, hence can be represented by a PSR with rank $\leq n$

What systems fall in PSRs \ HMMs?

- Recall that HMMs with n states has an SDM with rank $\leq n$, hence can be represented by a PSR with rank $\leq n$
- Not vice versa: there exists PSR with constant size that **cannot** be represented by any HMM with **finitely many** hidden states

What systems fall in PSRs \ HMMs?

- Recall that HMMs with n states has an SDM with rank $\leq n$, hence can be represented by a PSR with rank $\leq n$
- Not vice versa: there exists PSR with constant size that **cannot** be represented by any HMM with **finitely many** hidden states
  - "Probability lock": 0-1 sequence where the probability of 1 appearing next goes like a sine wave sampled at an interval that is not a rational multiple of the wave's period; see Jaeger [2000] for details



Controlled systems

- Almost everything extend straightforwardly

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_{1 \ldots} o_l]$ specifies an uncontrolled system

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_{1}...o_{l}]$ specifies an uncontrolled system
  - $\Pr[o_{1}...o_{l} \,||\, a_{0}...a_{l\text{-}1}]$ specifies a controlled system

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_1 \ldots o_l]$ specifies an uncontrolled system
  - $\Pr[o_1 \ldots o_l \,||\, a_0 \ldots a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_{1\ldots}o_l]$ specifies an uncontrolled system
  - $\Pr[o_{1\ldots}o_l \,||\, a_{0\ldots}a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*
  - "If I were to take $a_{0\ldots}a_{l\text{-}1}$, what's the odds that I see $o_{1\ldots}o_l$?"

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_{1...}o_l]$ specifies an uncontrolled system
  - $\Pr[o_{1...}o_l \,||\, a_{0...}a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*
  - "If I were to take $a_{0...}a_{l\text{-}1}$, what's the odds that I see $o_{1...}o_l$?"
  - Does it restrict us to open-loop policies? Answer: no.

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_1 \ldots o_l]$ specifies an uncontrolled system
  - $\Pr[o_1 \ldots o_l \mid\mid a_0 \ldots a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*
  - "If I were to take $a_0 \ldots a_{l\text{-}1}$, what's the odds that I see $o_1 \ldots o_l$?"
  - Does it restrict us to open-loop policies? Answer: no.
- Conditional: $\Pr[\mathrm{obs}(t) \mid h \mid\mid \mathit{do}\ \mathrm{act}(t)]$

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_{1}...o_{l}]$ specifies an uncontrolled system
  - $\Pr[o_{1}...o_{l} \mid\mid a_{0}...a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*
  - "If I were to take $a_{0}...a_{l\text{-}1}$, what's the odds that I see $o_{1}...o_{l}$?"
  - Does it restrict us to open-loop policies? Answer: no.
- Conditional: $\Pr[\mathrm{obs}(t) \mid h \mid\mid \mathit{do}\ \mathrm{act}(t)]$
  - $\mathrm{obs}(.)$ and $\mathrm{act}(.)$ omit actions and obs., respectively

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_{1}...o_{l}]$ specifies an uncontrolled system
  - $\Pr[o_{1}...o_{l} \,||\, a_{0}...a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*
  - "If I were to take $a_{0}...a_{l\text{-}1}$, what's the odds that I see $o_{1}...o_{l}$?"
  - Does it restrict us to open-loop policies? Answer: no.
- Conditional: $\Pr[\mathrm{obs}(t) \,|\, h \,||\, do\ \mathrm{act}(t)]$
  - $\mathrm{obs}(.)$ and $\mathrm{act}(.)$ omit actions and obs., respectively
  - what you've done in the past are factual; *what you could do in the future are counterfactual*

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_1 \ldots o_l]$ specifies an uncontrolled system
  - $\Pr[o_1 \ldots o_l \mid\mid a_0 \ldots a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*
  - "If I were to take $a_0 \ldots a_{l\text{-}1}$, what's the odds that I see $o_1 \ldots o_l$?"
  - Does it restrict us to open-loop policies? Answer: no.
- Conditional: $\Pr[\mathrm{obs}(t) \mid h \mid\mid do\ \mathrm{act}(t)]$
  - $\mathrm{obs}(.)$ and $\mathrm{act}(.)$ omit actions and obs., respectively
  - what you've done in the past are factual; *what you could do in the future are counterfactual*
  - Hence t stands for "**test**": take actions to probe the response of the system

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_1 \ldots o_l]$ specifies an uncontrolled system
  - $\Pr[o_1 \ldots o_l \,||\, a_0 \ldots a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*
  - "If I were to take $a_0 \ldots a_{l\text{-}1}$, what's the odds that I see $o_1 \ldots o_l$?"
  - Does it restrict us to open-loop policies? Answer: no.
- Conditional: $\Pr[\mathrm{obs}(t) \mid h \,||\, do\ \mathrm{act}(t)]$
  - $\mathrm{obs}(.)$ and $\mathrm{act}(.)$ omit actions and obs., respectively
  - what you've done in the past are factual; *what you could do in the future are counterfactual*
  - Hence t stands for "**test**": take actions to probe the response of the system
  - Dynamical system research is inherently related to causality

Controlled systems

- Almost everything extend straightforwardly
  - … as long as you know how to define SDM
- $\Pr[o_1 ... o_l]$ specifies an uncontrolled system
  - $\Pr[o_1 ... o_l \,||\, a_0 ... a_{l\text{-}1}]$ specifies a controlled system
  - Actions are not r.v. (unless we fix a policy); they are *interventions*
  - "If I were to take $a_0 ... a_{l\text{-}1}$, what's the odds that I see $o_1 ... o_l$?"
  - Does it restrict us to open-loop policies? Answer: no.
- Conditional: $\Pr[\mathrm{obs}(t) \mid h \,||\, do\ \mathrm{act}(t)]$
  - $\mathrm{obs}(.)$ and $\mathrm{act}(.)$ omit actions and obs., respectively
  - what you've done in the past are factual; *what you could do in the future are counterfactual*
  - Hence t stands for "**test**": take actions to probe the response of the system

    e.g., off-policy eval with unknown behavior policy
  - Dynamical system research is inherently related to causality

Challenges in PSRs

Challenges in PSRs

- Moment matching algorithm; no optimization

Challenges in PSRs

- Moment matching algorithm; no optimization
  - sensitive to model mismatch

Challenges in PSRs

- Moment matching algorithm; no optimization
  - sensitive to model mismatch
- Rely on linearity

Challenges in PSRs

- Moment matching algorithm; no optimization
  - sensitive to model mismatch
- Rely on linearity
  - some ideas extend to nonlinear but little can be said theoretically

Challenges in PSRs

- Moment matching algorithm; no optimization
  - sensitive to model mismatch
- Rely on linearity
  - some ideas extend to nonlinear but little can be said theoretically
- Cannot handle rich/continuous observations well

Challenges in PSRs

- Moment matching algorithm; no optimization
  - sensitive to model mismatch
- Rely on linearity
  - some ideas extend to nonlinear but little can be said theoretically
- Cannot handle rich/continuous observations well
  - Aim to learn $\Pr[o_{1...}o_l]$

Challenges in PSRs

- Moment matching algorithm; no optimization
  - sensitive to model mismatch
- Rely on linearity
  - some ideas extend to nonlinear but little can be said theoretically
- Cannot handle rich/continuous observations well
  - Aim to learn $\Pr[o_{1 \ldots} o_l]$
  - Explicitly modeling density of rich obs is hard (c.f., GAN)

Challenges in PSRs

- Moment matching algorithm; no optimization
  - sensitive to model mismatch
- Rely on linearity
  - some ideas extend to nonlinear but little can be said theoretically
- Cannot handle rich/continuous observations well
  - Aim to learn $\Pr[o_{1...}o_l]$
  - Explicitly modeling density of rich obs is hard (c.f., GAN)
  - There are a lot of details that we don't care—need to factor that into PSR theory

Challenges in PSRs

- Moment matching algorithm; no optimization
  - sensitive to model mismatch
- Rely on linearity
  - some ideas extend to nonlinear but little can be said theoretically
- Cannot handle rich/continuous observations well
  - Aim to learn $\Pr[o_{1\ldots}o_l]$
  - Explicitly modeling density of rich obs is hard (c.f., GAN)
  - There are a lot of details that we don't care—need to factor that into PSR theory
- When combined with planning, the approach is model-based RL (which isn't working quite well yet in the era of deep RL)