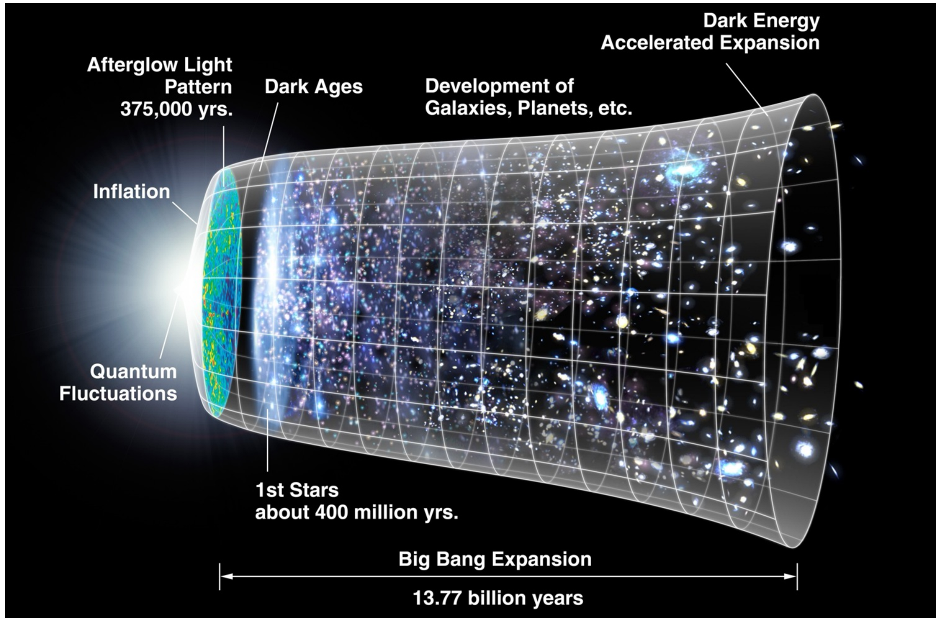
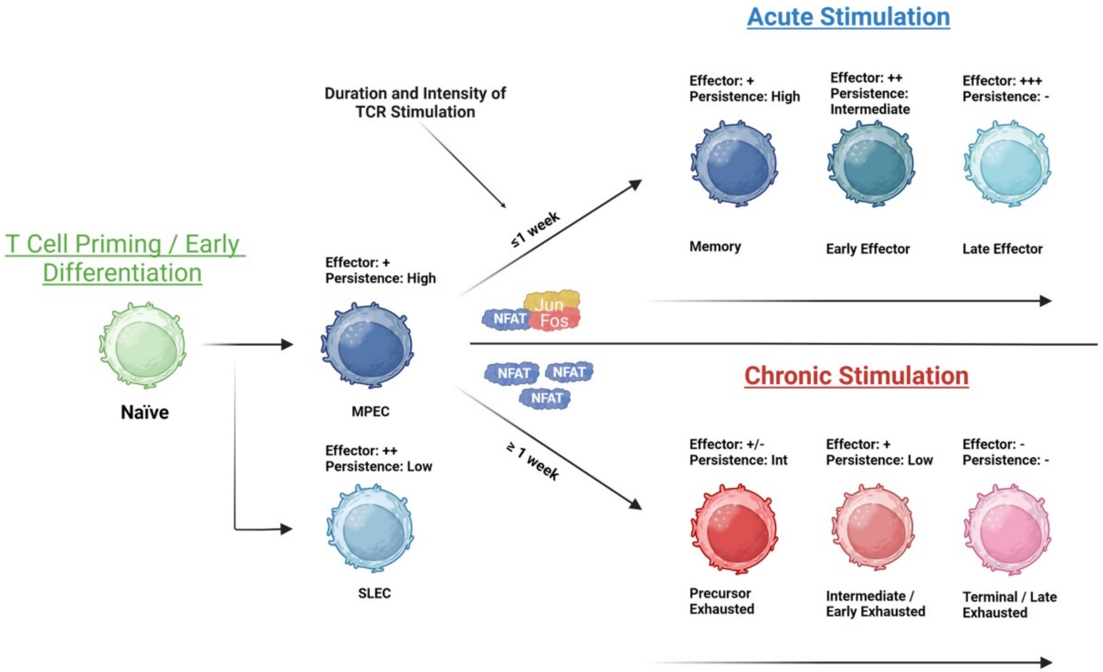
Learning the Branching

to understand evolution

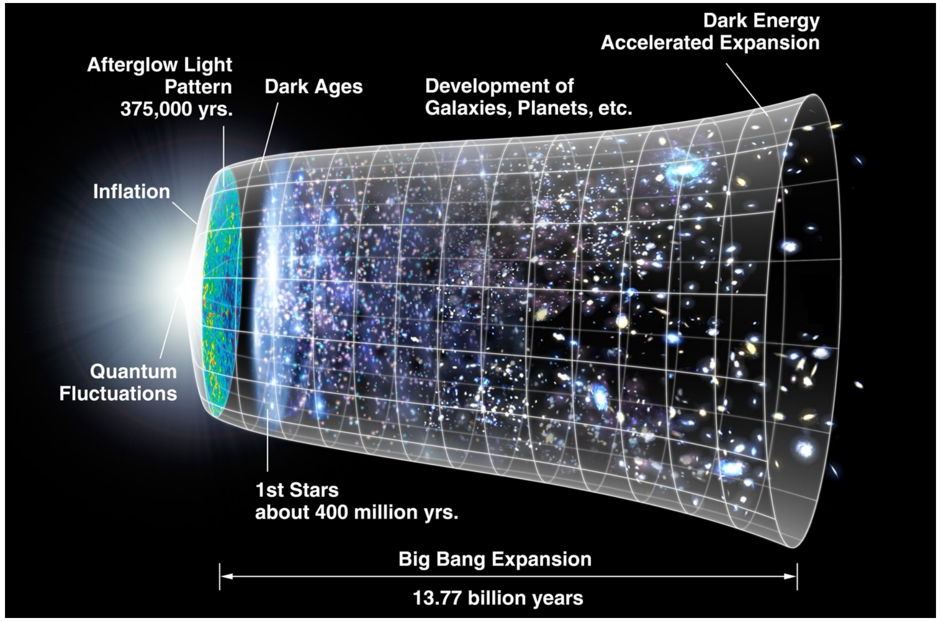
GM - Qichen Huang
Paul François Lab

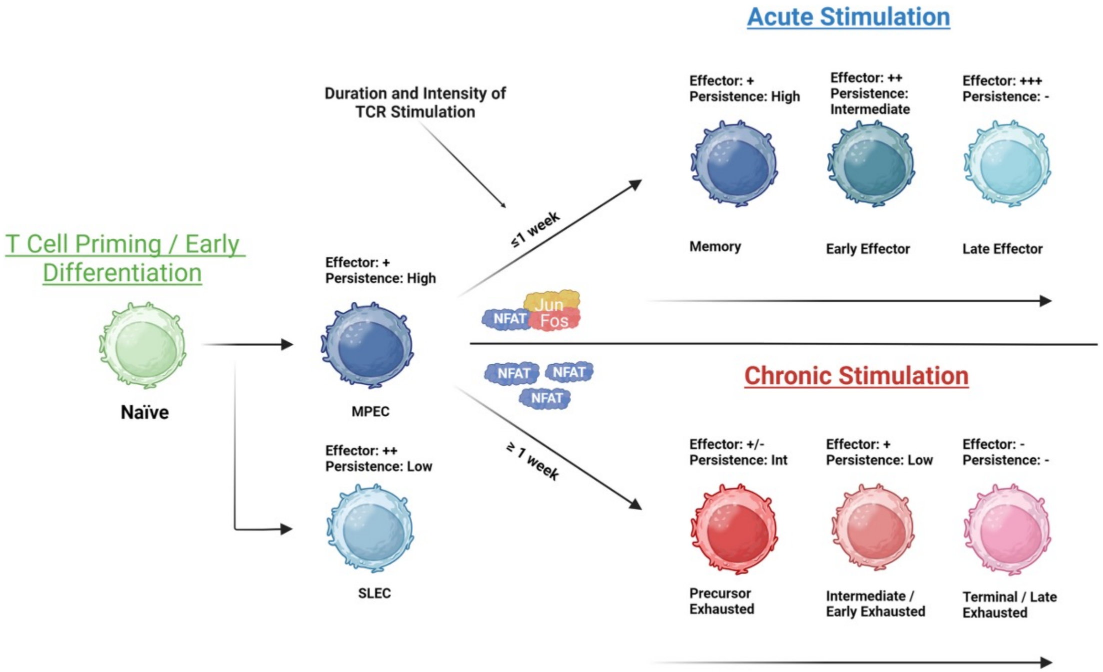
BUT how to Learn them ?

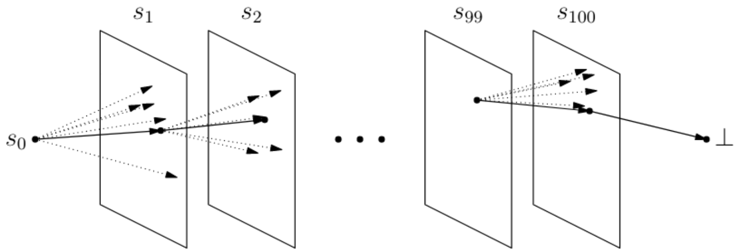




BUT how to Learn them ?

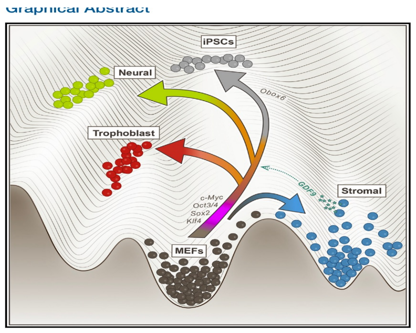






← need a Generative Model

need a landscape
as reward function →



There are 4 cases where Branching are Learnable

1: Markov Process w fixed Transition (time-invariant policy)

2: Tree structure in nature (only one way to reach the leave)

3: with Partial Reward signal over time.

4: If the trajectories are optimal transport (OT) in nature

There are 4 cases where Branching are Learnable

1: Markov Process w fixed Transition (time-invariant policy)

2: Tree structure in nature (only one way to reach the leave)

3: with Partial Reward signal over time.

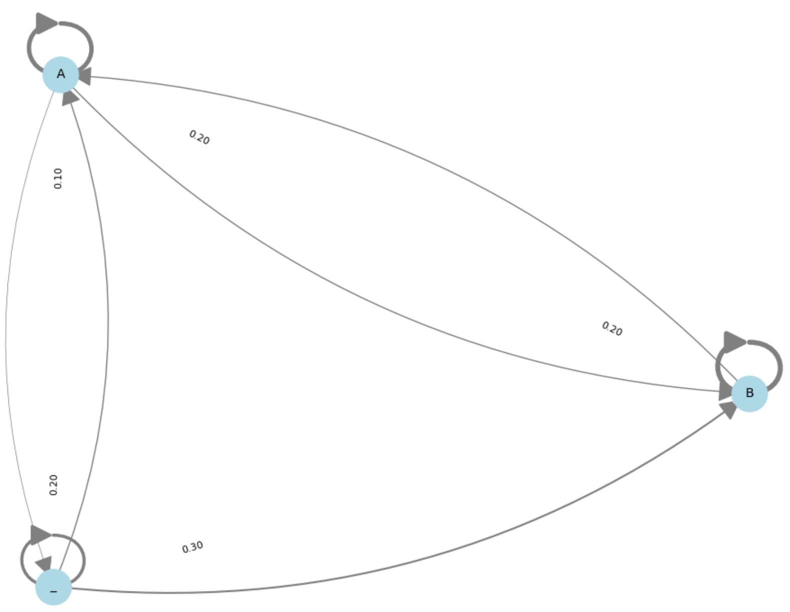
4: If the trajectories are optimal transport (OT) in nature

proposition: branch is learnable when only one optimal flow exist (easy to prove)

Case 1: Markov Process w Time-Invariant Policy (i.e. fixed transition matrix P over time)

toy-eg: 1 char DNA seq from {_, A, B}

Markov Chain Visualization



fixed P over time

(a) Markov Chain Evolution as Transport:

- The transition matrix P evolves a distribution $\pi(0)$ over time:

$$\pi(t) = \pi(0)P^t.$$

- This can be viewed as a transport problem, where P specifies how mass is "moved" between states at each step.

The Markov chain has a unique stationary distribution π^*, which satisfies:

$$\pi^* P = \pi^*$$

and the sum of probabilities in π^* is 1:

$$\sum_i \pi_i^* = 1.$$

Regardless of the initial distribution $\pi(0)$, the chain will converge to π^* as $t \to \infty$:

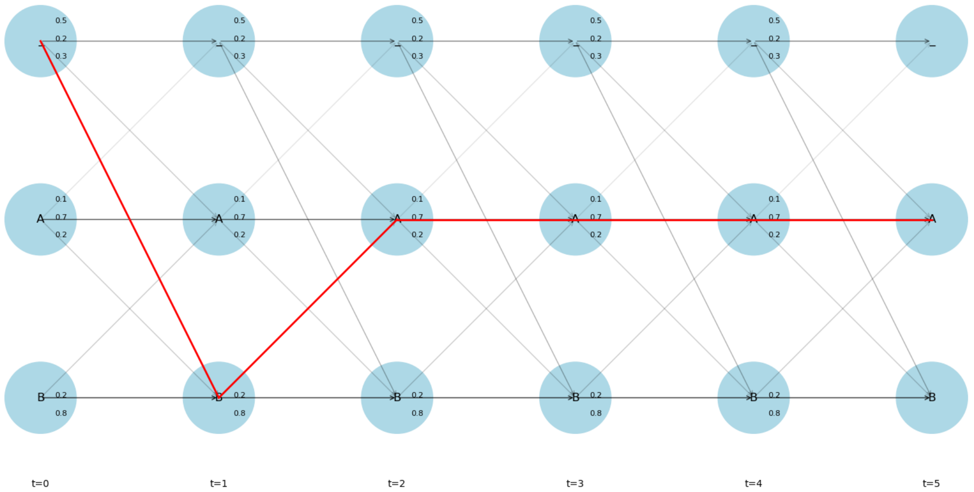
$$\pi(t) \to \pi^* \quad \text{as } t \to \infty.$$

In this case:

$$P \leftrightarrow \pi^*$$

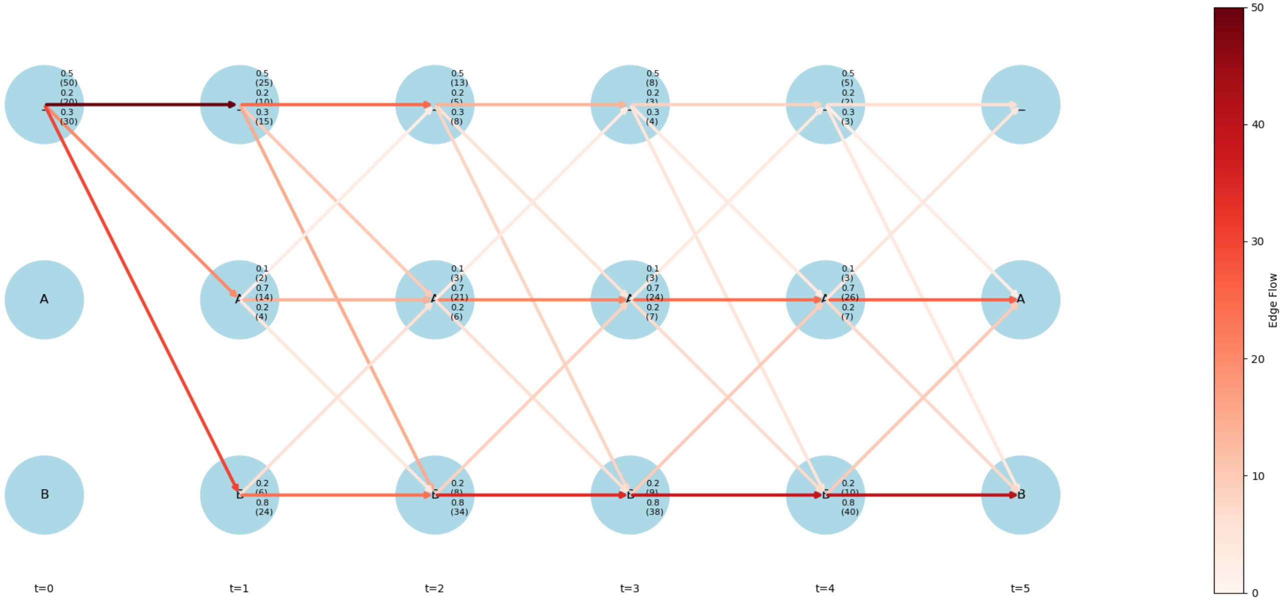
is a one-to-one relationship because:

- P uniquely determines π^* as the stationary distribution.
- π^*, if known, implies properties about P (e.g., it can help you infer the long-term behavior of the chain or properties like irreducibility).

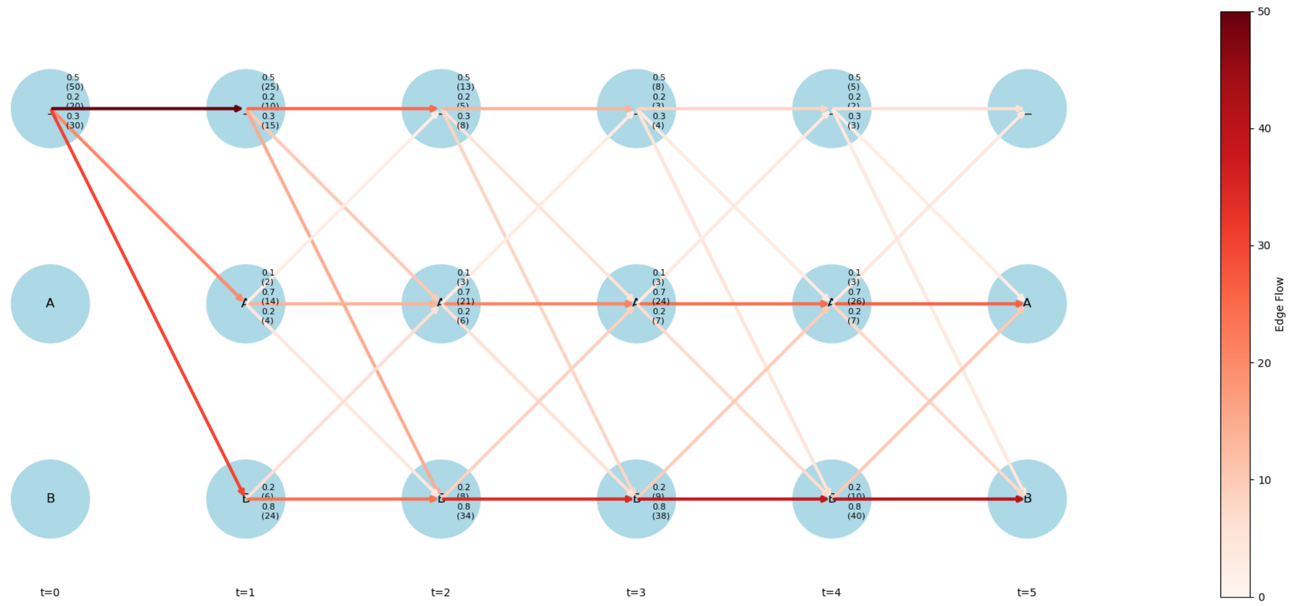


fixed policy

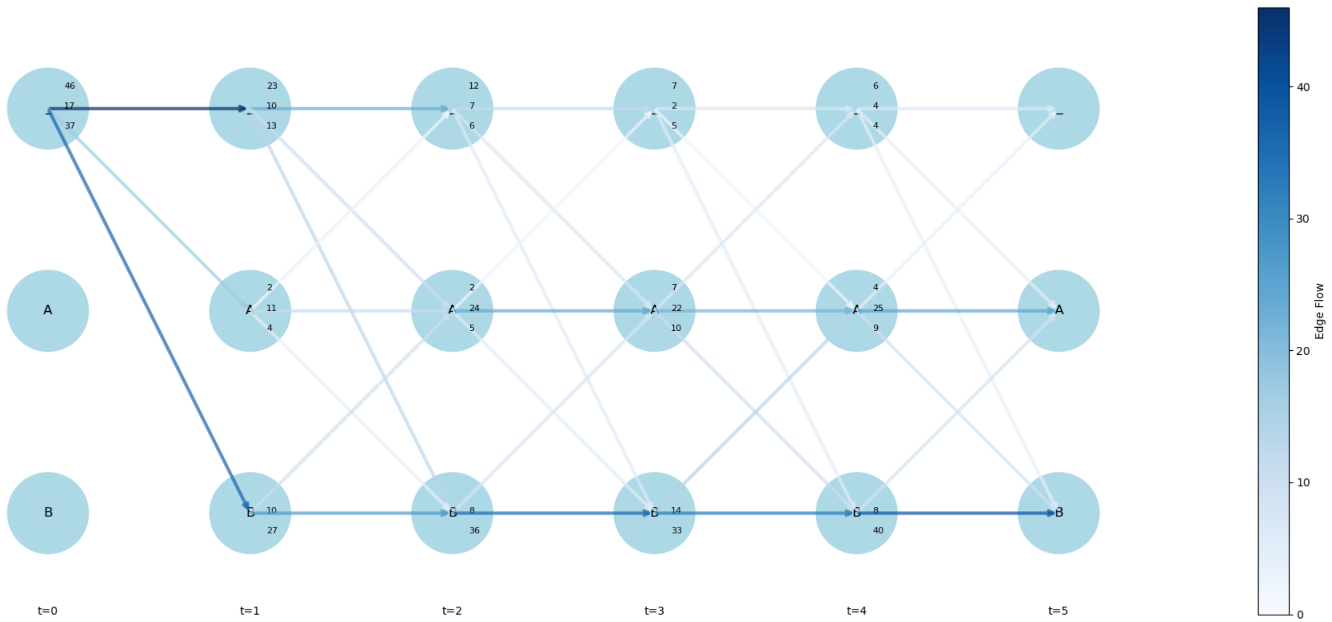
```python
# Define policy for each state
policies = {
    '_': {'A': 0.2, 'B': 0.3, '_': 0.5},  # Policy for initial state
    'A': {'A': 0.7, 'B': 0.2, '_': 0.1},  # Policy for state A
    'B': {'A': 0.2, 'B': 0.8, '_': 0.0}   # Policy for state B
}
```



theoretical Flow



theoretical Flow

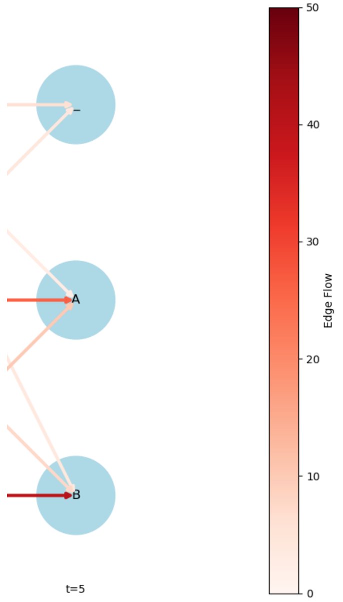


empirical Flow

(but what if policy is not fixed)

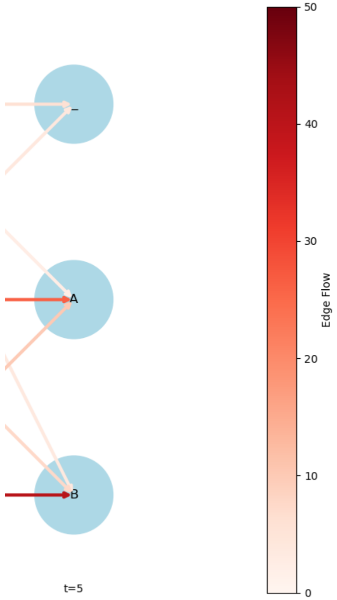
ie. Time-Variant Markov Process

what GFN learn?

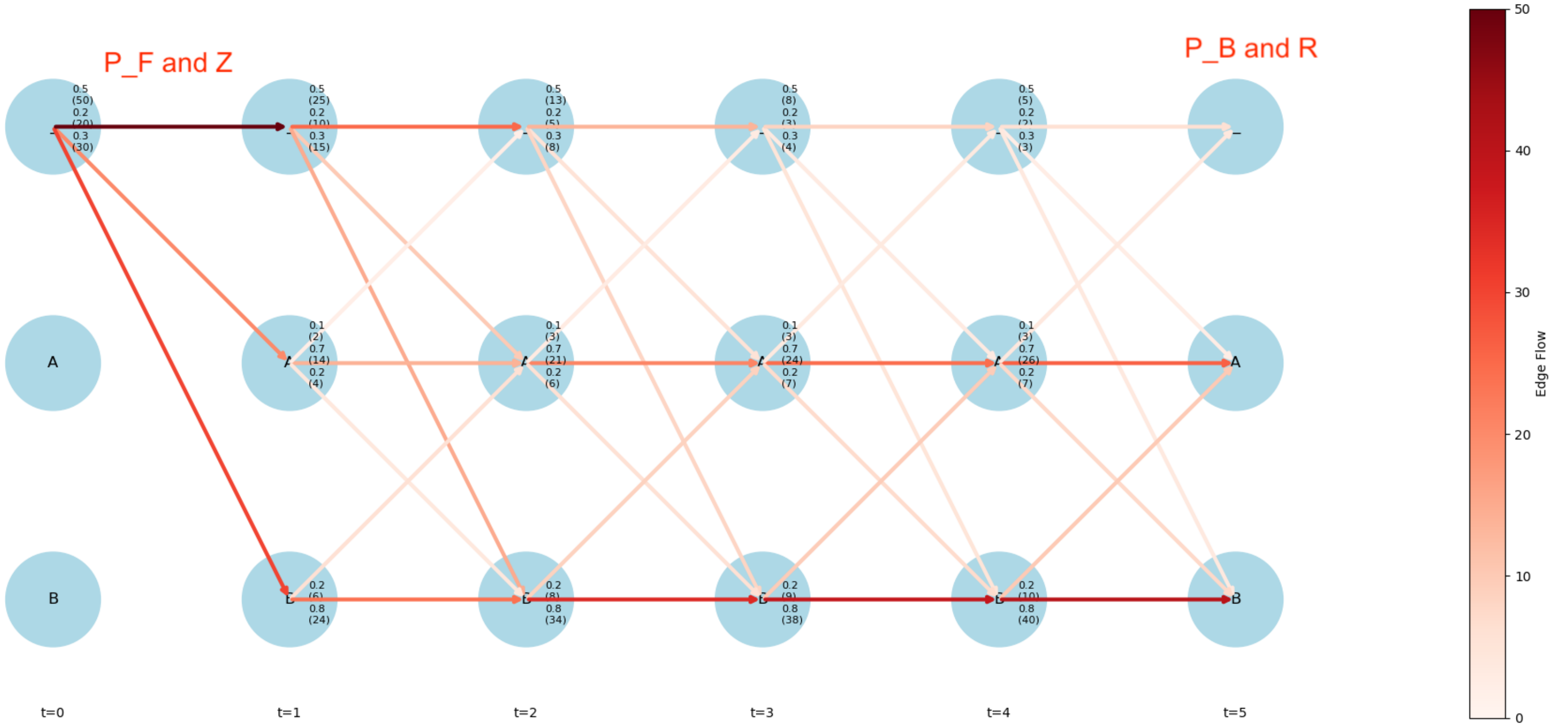


terminal Reward distribution

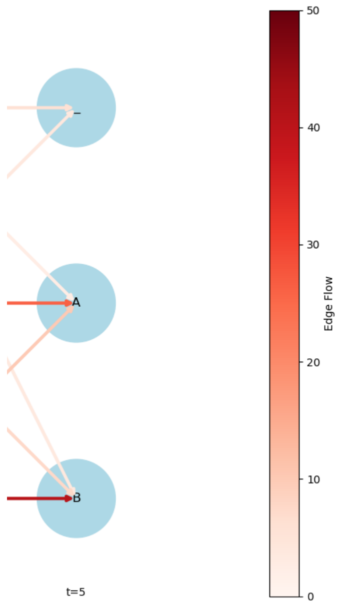
what GFN learn?



to infer →



what GFN learn?



terminal Reward distribution

policy (transition)

flow

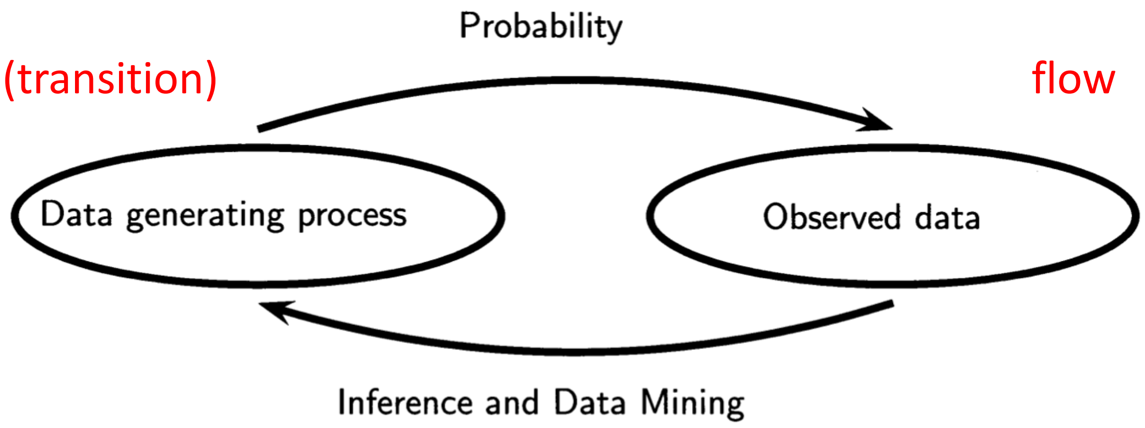


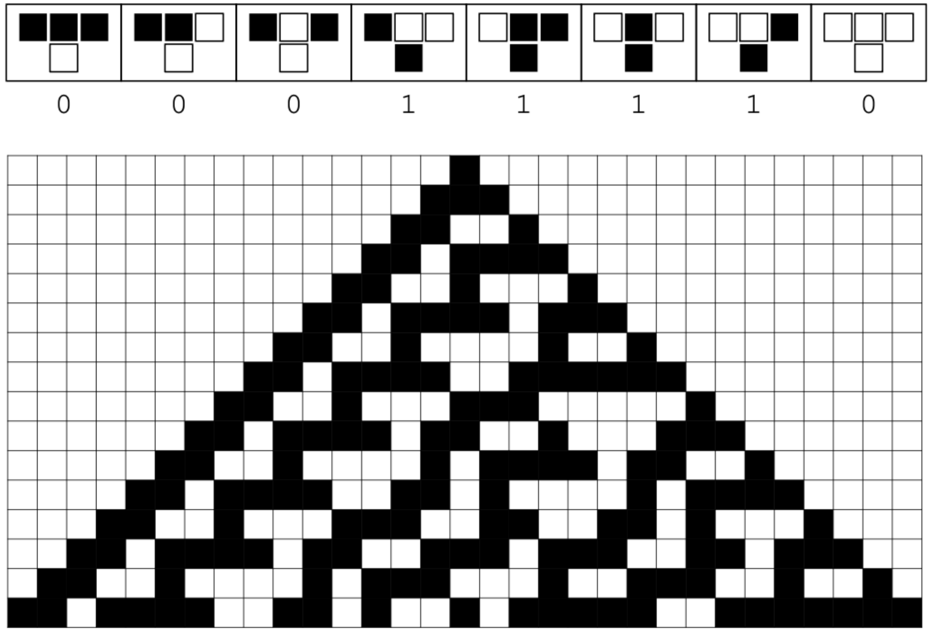
FIGURE 1. Probability and inference.

given
- State space
- Action space
- Reward

what GFN learn?

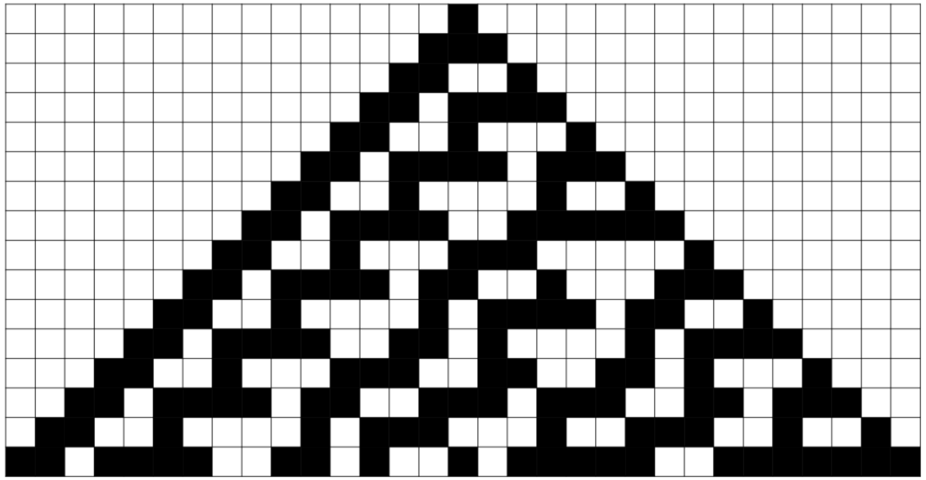
rule 30

Rule-Based GFN

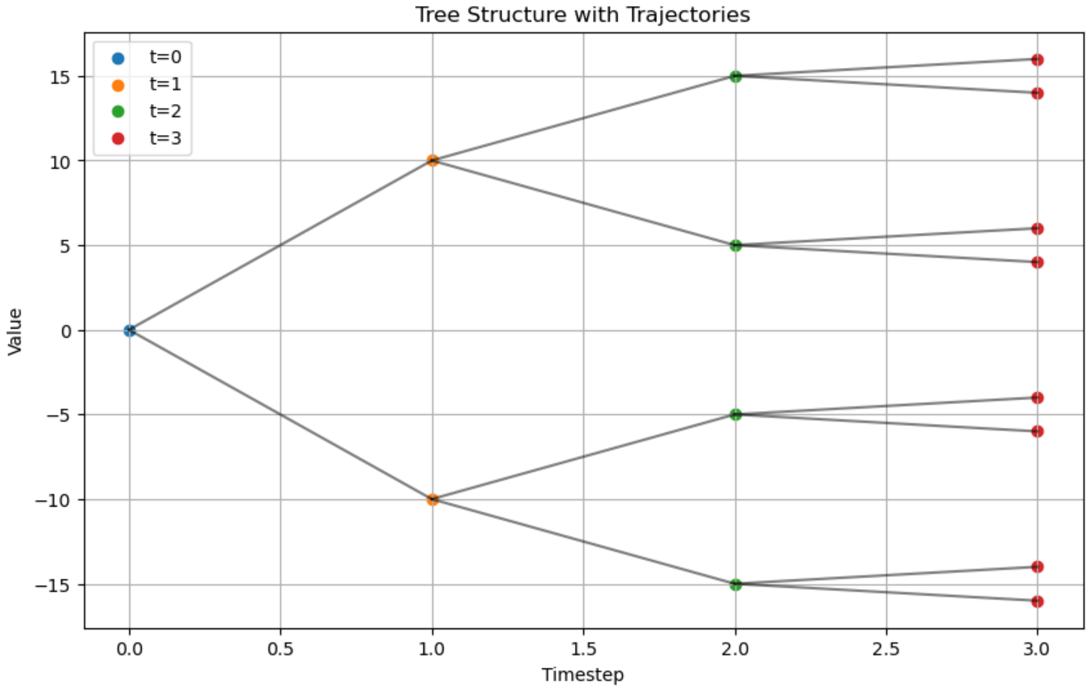


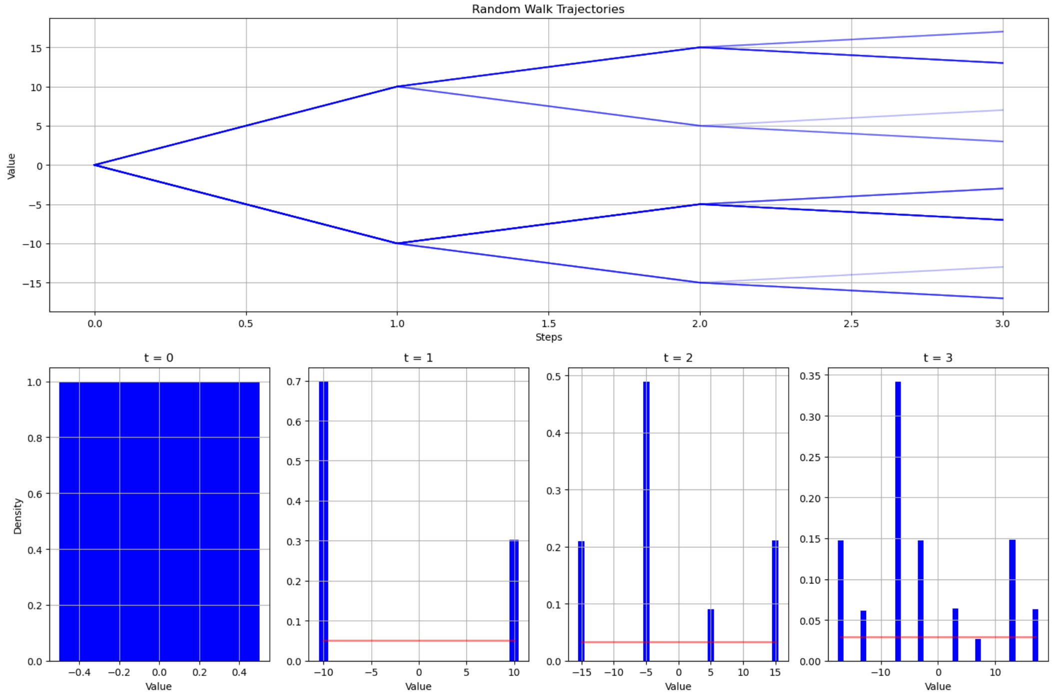
what GFN learn?

what’s the transition rule?



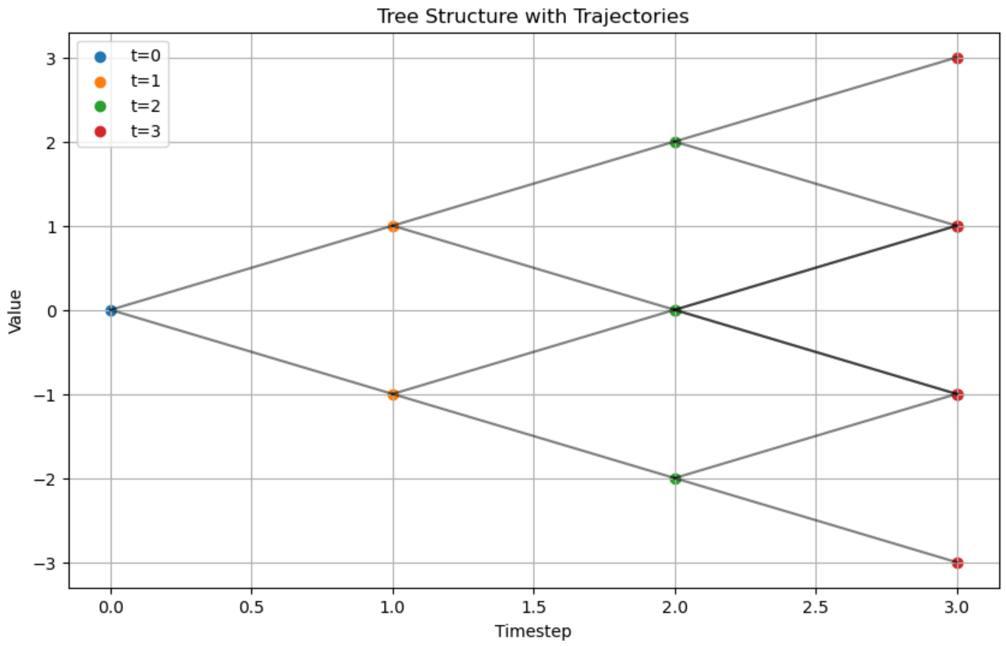
Case 2: Tree structure in nature (only one way to reach the leave)

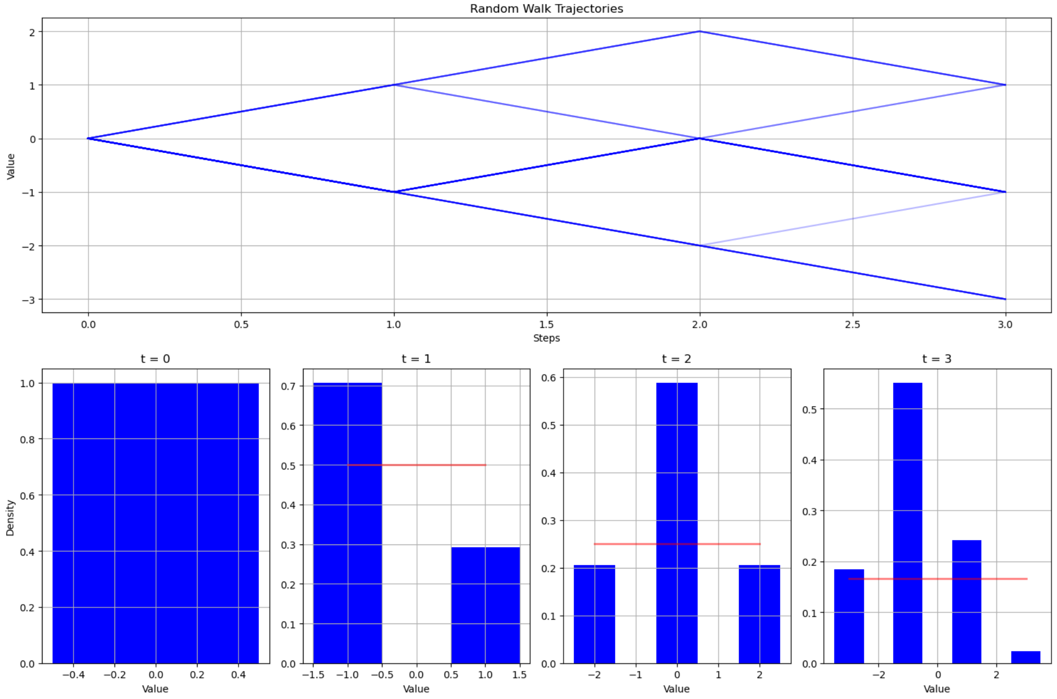





```python
# Parameters for random walk
n_samples = 100000
n_steps = 4  # Changed to 4 to match the tree structure
step_sizes = [10, 5, 2]  # Step sizes corresponding to each timestep

# Define biased probabilities for each timestep
probabilities = [
    [-1, -1],  # t = 0, not used
    [0.7, 0.3],  # t = 1, more likely to go up
    [0.3, 0.7],  # t = 2, more likely to go down
    [0.7, 0.3]  # t = 3, uniform
]
```

Case 3: Agent gets Partial Reward signal on the Trajectory over time.

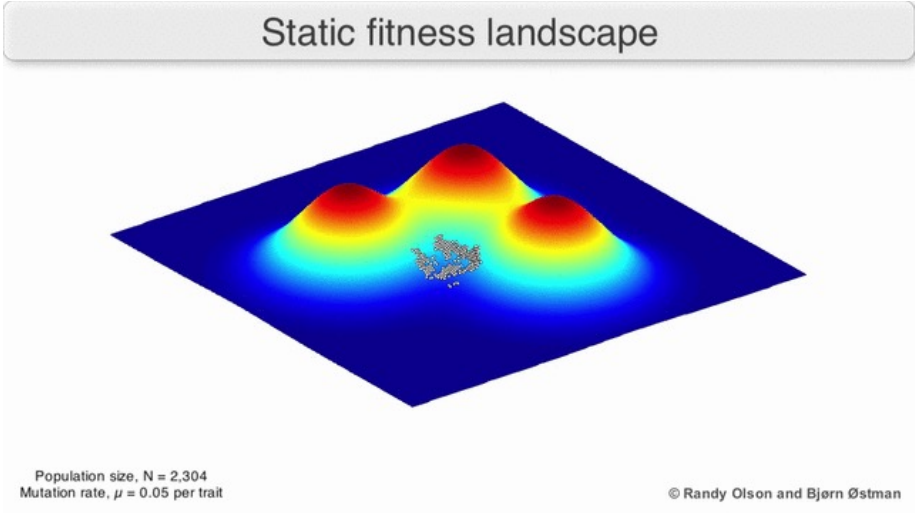




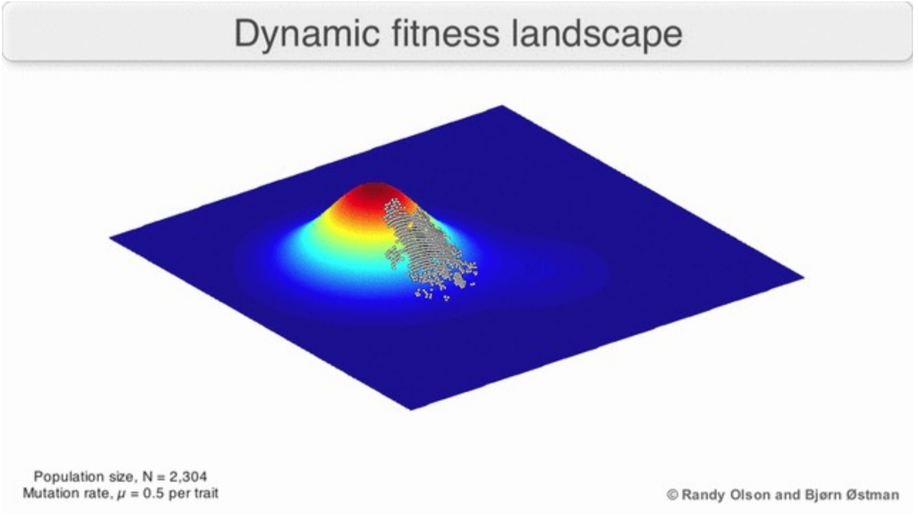

```
# Parameters for random walk
n_samples = 10000
n_steps = 4  # Changed to 4 to match the tree structure
step_sizes = [1, 1, 1]  # Step sizes corresponding to each timestep, adapted to actions

# Define biased probabilities for each timestep
probabilities = [
    [-1, -1],  # t = 0, not used
    [0.7, 0.3],  # t = 1, more likely to go up
    [0.3, 0.7],  # t = 2, more likely to go down
    [0.9, 0.1]   # t = 3, uniform
]
```

Reward signal over time
(any prob distribution is learnable)

Case 3: Agent gets Partial Reward signal on the Trajectory over time.

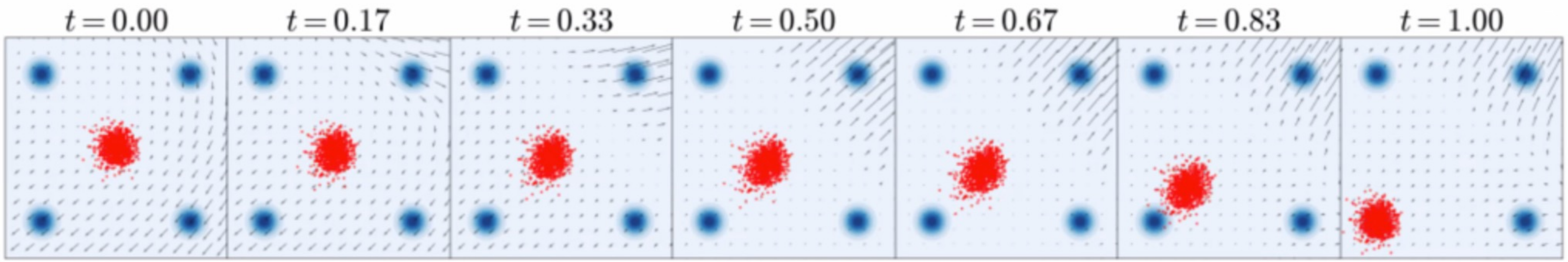


static Reward signal



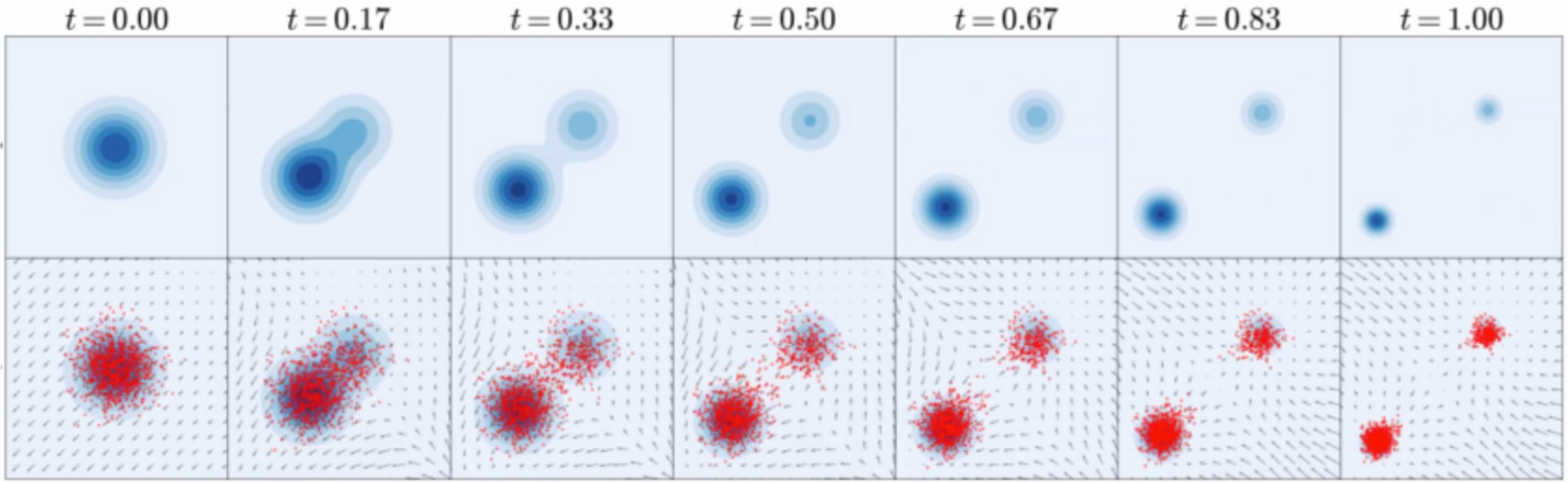
dynamic Reward signal

Case 4: If the trajectories are optimal transport (OT) in nature



not diverse

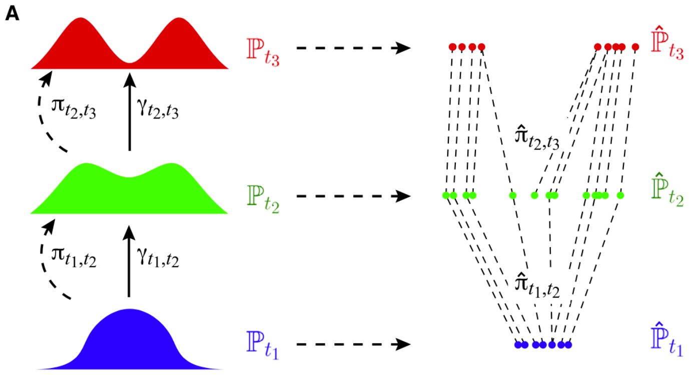
Learning Interpolations between Boltzmann Densities, Máté & Fleuret, 2023



diverse

Learning Interpolations between Boltzmann Densities, Máté & Fleuret, 2023

Case 4: If the trajectories are optimal transport (OT) in nature



???

what's this distribution?

is cell diff OT ?
if so GFN can simulate it !!!

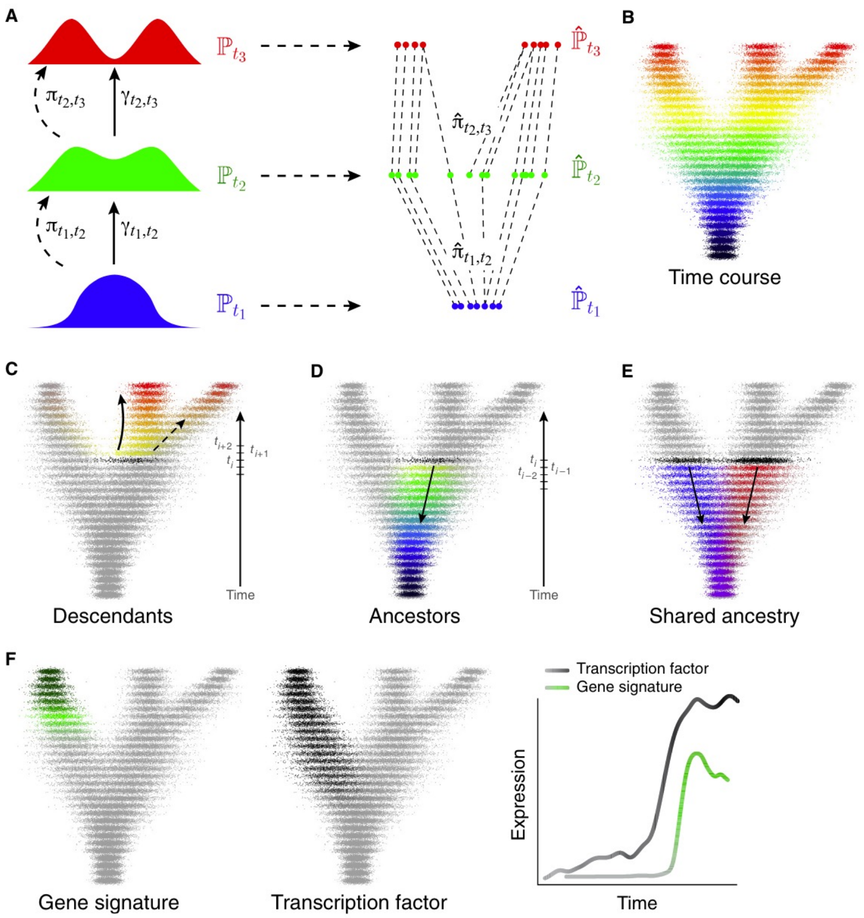


Figure 1. Modeling Developmental Processes with Optimal Transport

(A) A temporal progression of a time-varying distribution $\mathbb{P}_t$ (left) can be sampled to obtain finite empirical distributions of cells $\hat{\mathbb{P}}_{t_i}$ at various time points t_1, t_2, t_3 (right). Over short timescales, the unknown true coupling, γ_{t_1,t_2}, is assumed to be close to the optimal transport coupling, π_{t_1,t_2}, which can be approximated by $\hat{\pi}_{t_1,t_2}$ computed from the empirical distributions $\hat{\mathbb{P}}_{t_1}$ and $\hat{\mathbb{P}}_{t_2}$.

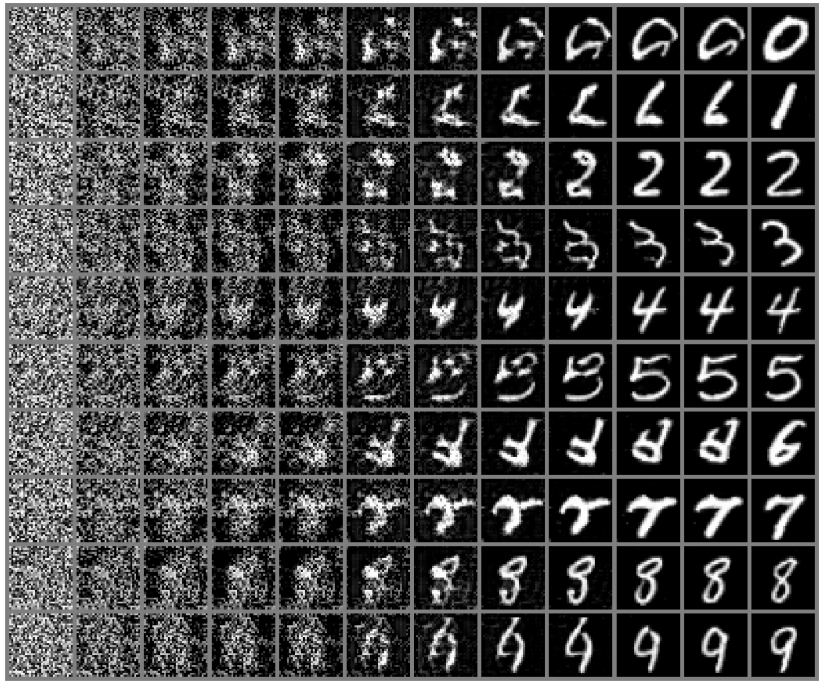
(B) Single-cell profiles (individual dots) are colored by the time of collection.

(C) Descendants of a cell set (black) at later times.

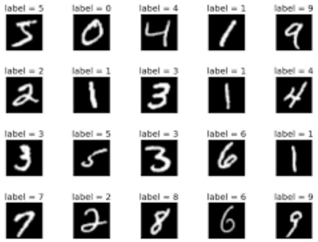
(D) Ancestors at earlier times.

(E) Shared ancestry of two cell sets (black). Ancestors of each population shown in red and blue, shared ancestors in purple.

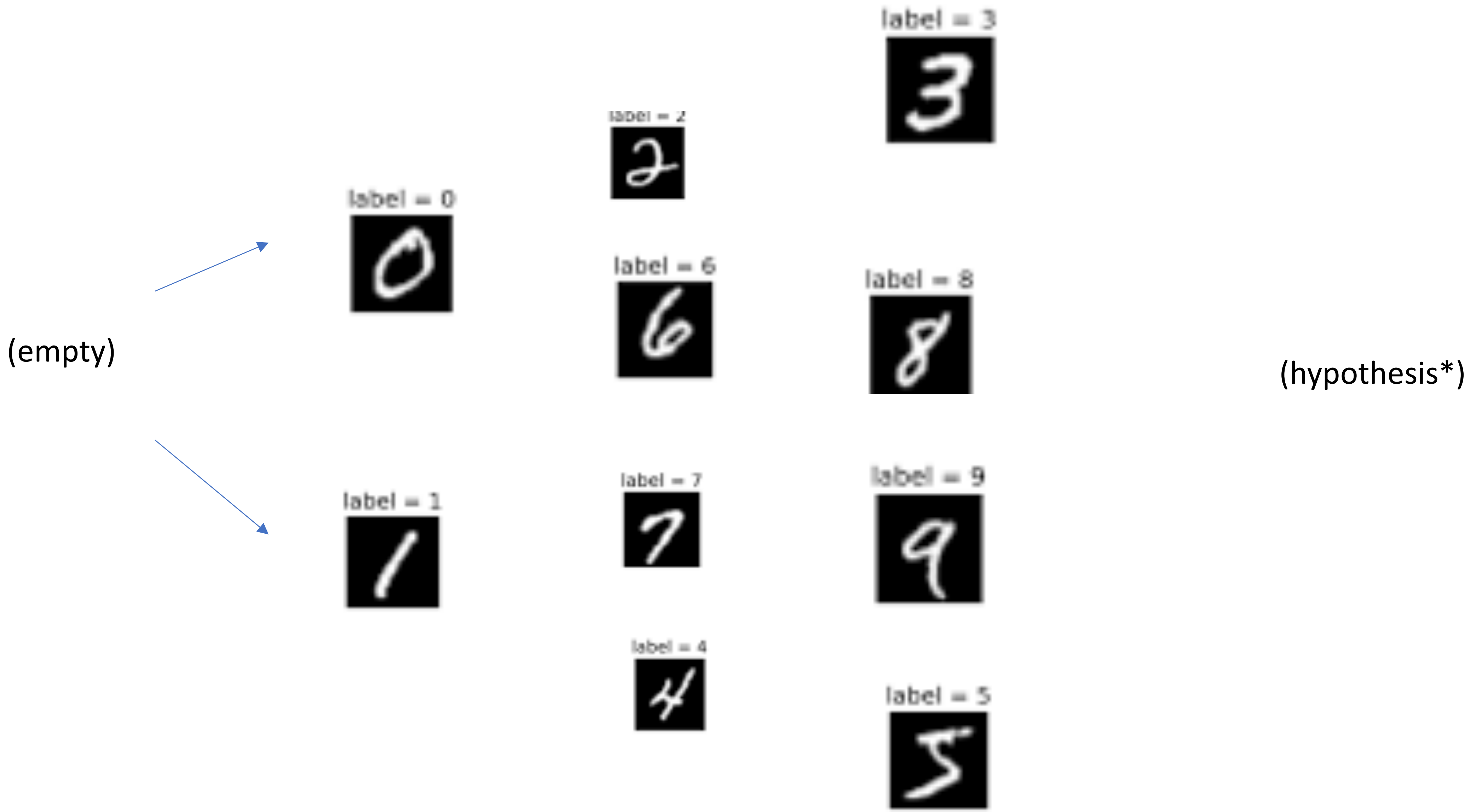
(F) Expression of gene signatures (left; green, high expression; gray, low expression) can be predicted from earlier expression of transcription factors (middle; black, high expression; gray, low expression) in a gene regulatory model by analyzing trends along ancestor trajectories (right).

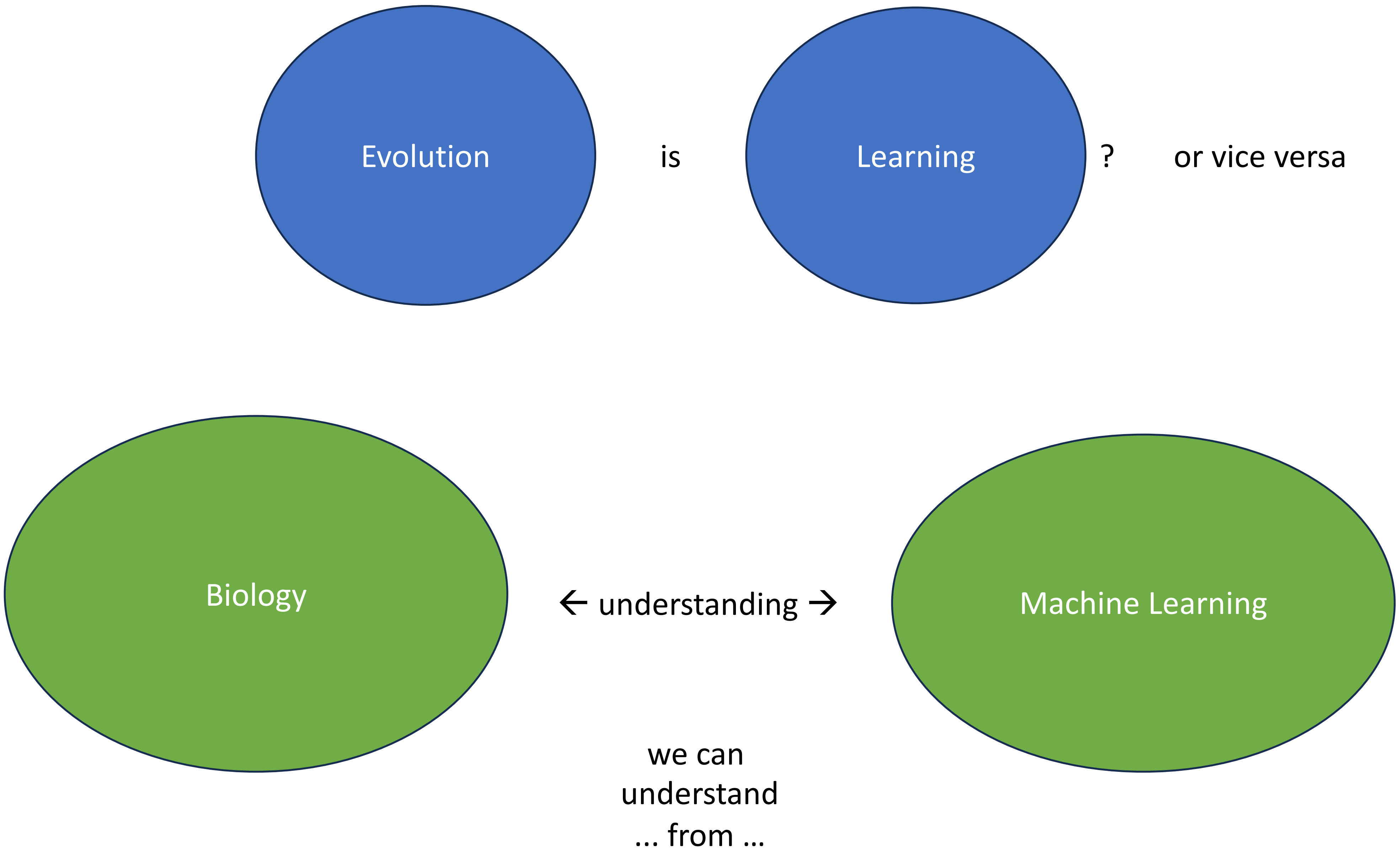


diffusion is not evolution !!



OT can be ?


(empty)
label = 0
label = 1
label = 2
label = 6
label = 7
label = 4
label = 3
label = 8
label = 9
label = 5
(hypothesis*)


Evolution
is
Learning
?
or vice versa
Biology
← understanding →
Machine Learning
we can
understand
... from ...

Learning the Branching

to understand evolution

with **branching** and **diversity**
we can

Understand the Evolution

to understand
Life, Universe, and Everything