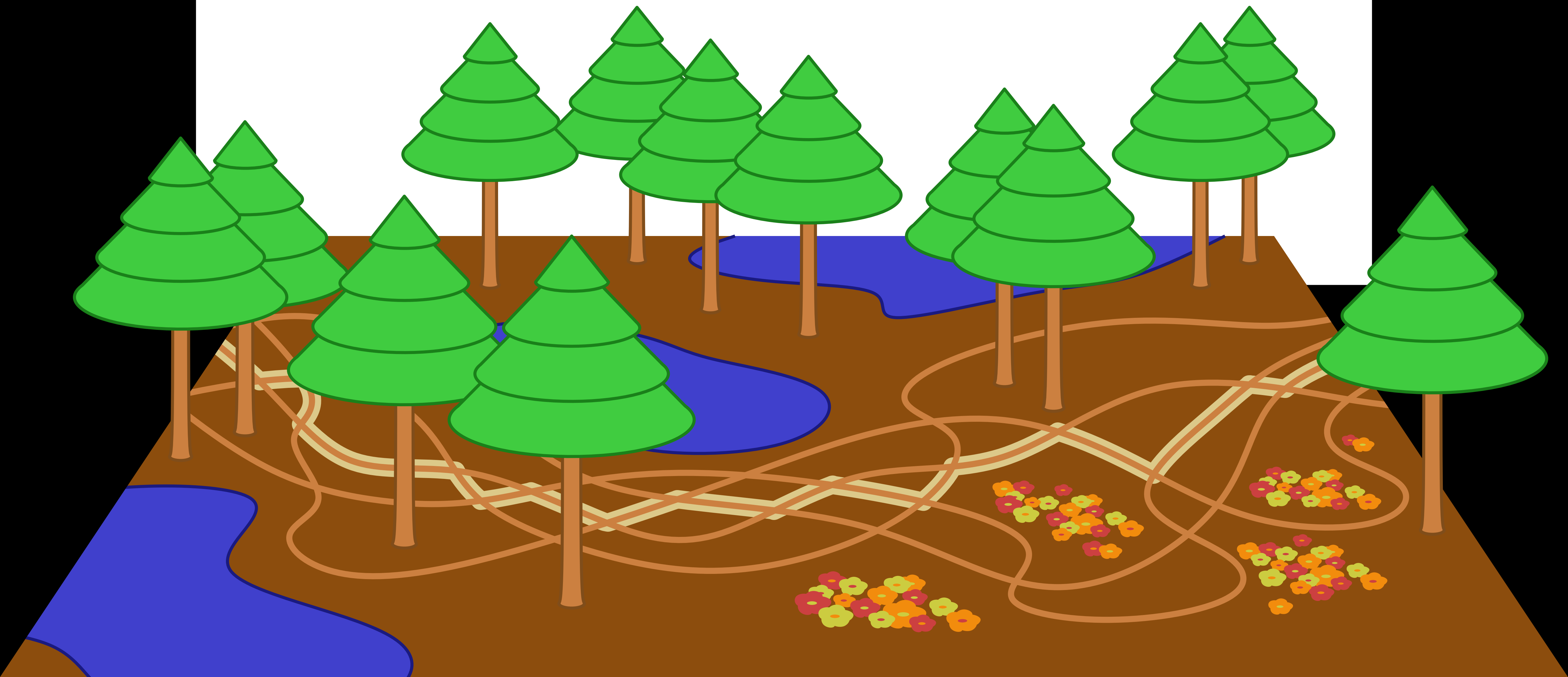
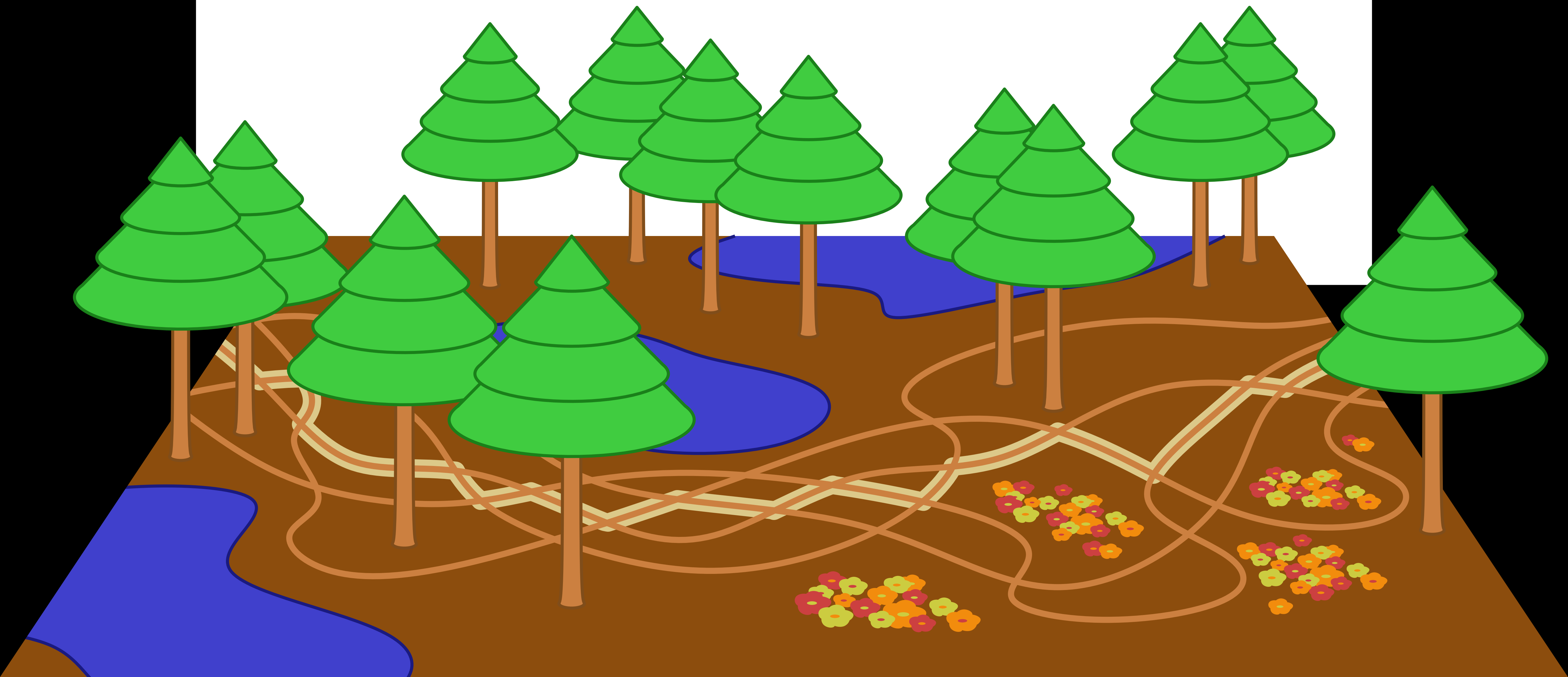
THANK YOU!



NEW APPROACHES FOR REPRESENTATIVE TRAJECTORIES

~~THANK YOU!~~



Homotopy Measures for Representative Trajectories

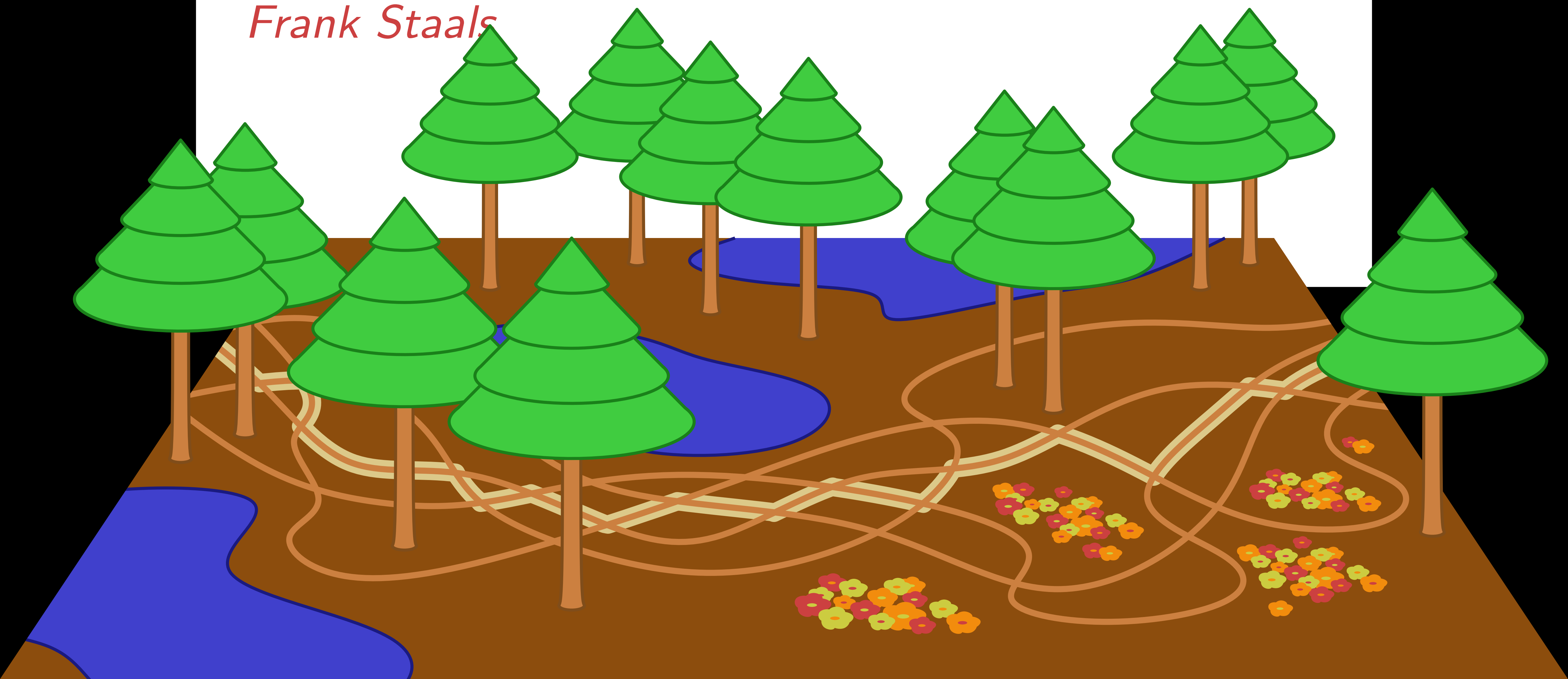
10:50 - 11:10

Erin Chambers
Irina Kostitsyna
Maarten Löffler
Frank Staals

Central Trajectories

11:10 - 11:30

Marc van Kreveld
Maarten Löffler
Frank Staals



TRAJECTORIES

TRAJECTORIES

- Let P be n points in the plane

TRAJECTORIES

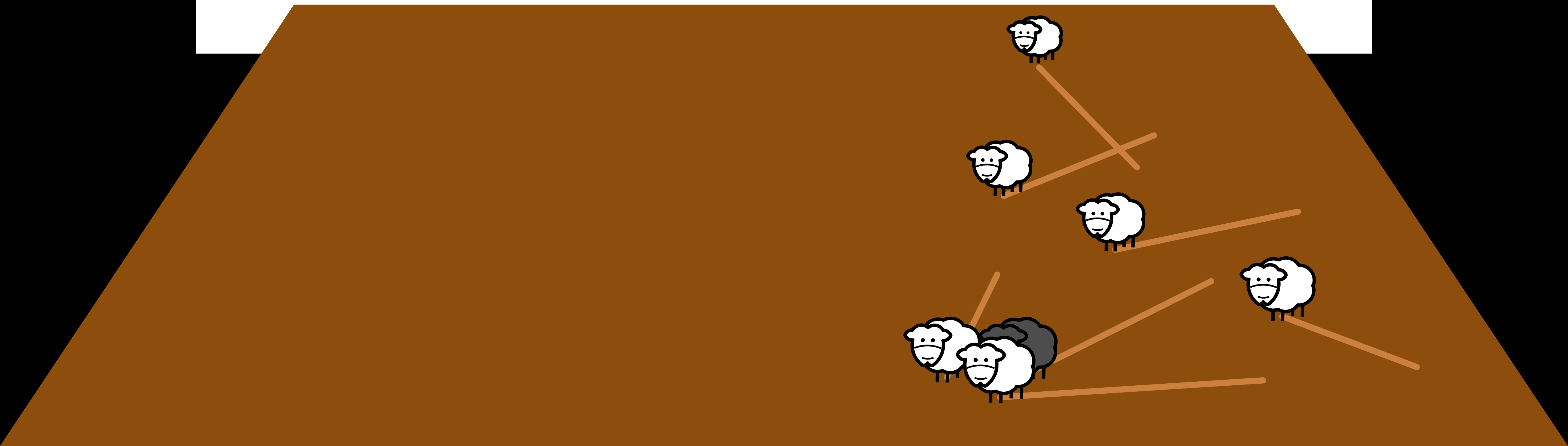
- Let P be n points in the plane

TRAJECTORIES

- Let P be n points in the plane
- Now suppose your points run away

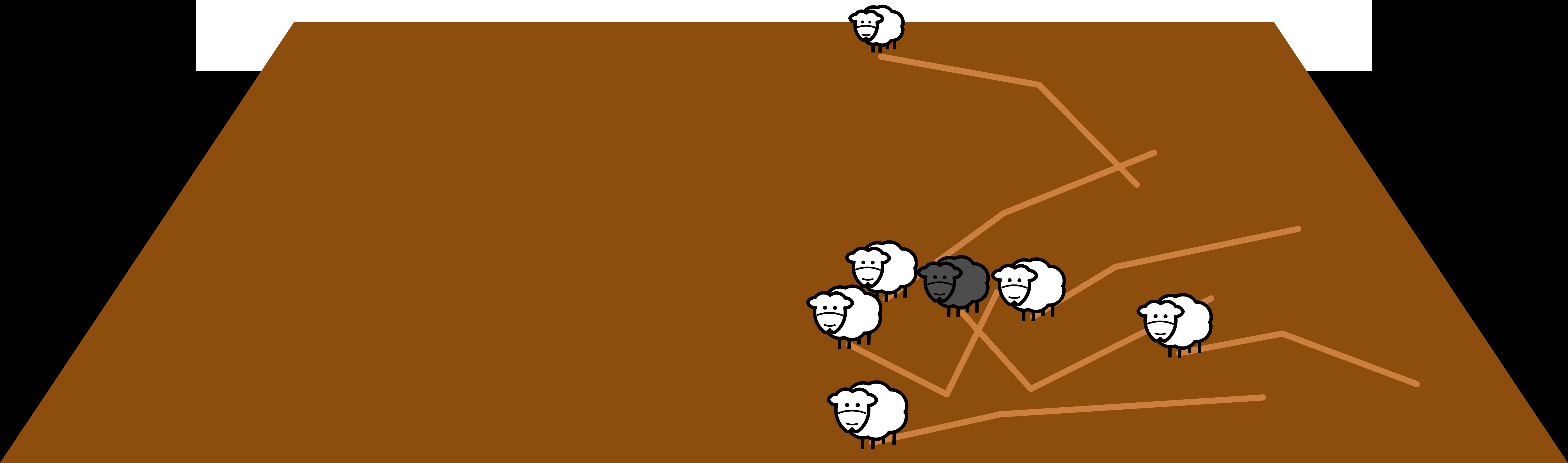
TRAJECTORIES

- Let P be n points in the plane
- Now suppose your points run away



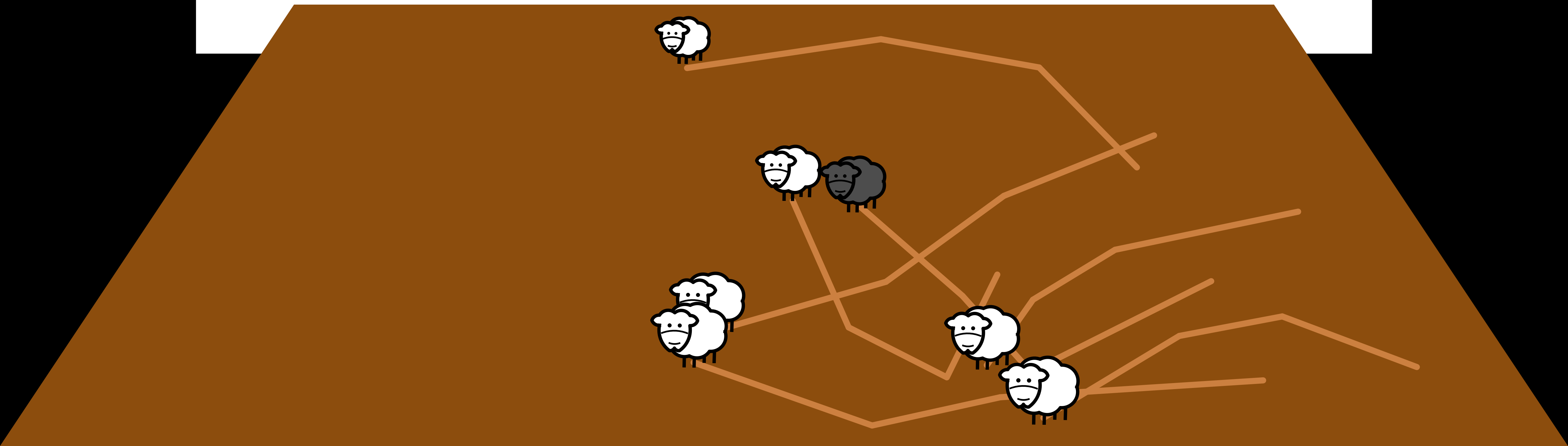
TRAJECTORIES

- Let P be n points in the plane
- Now suppose your points run away



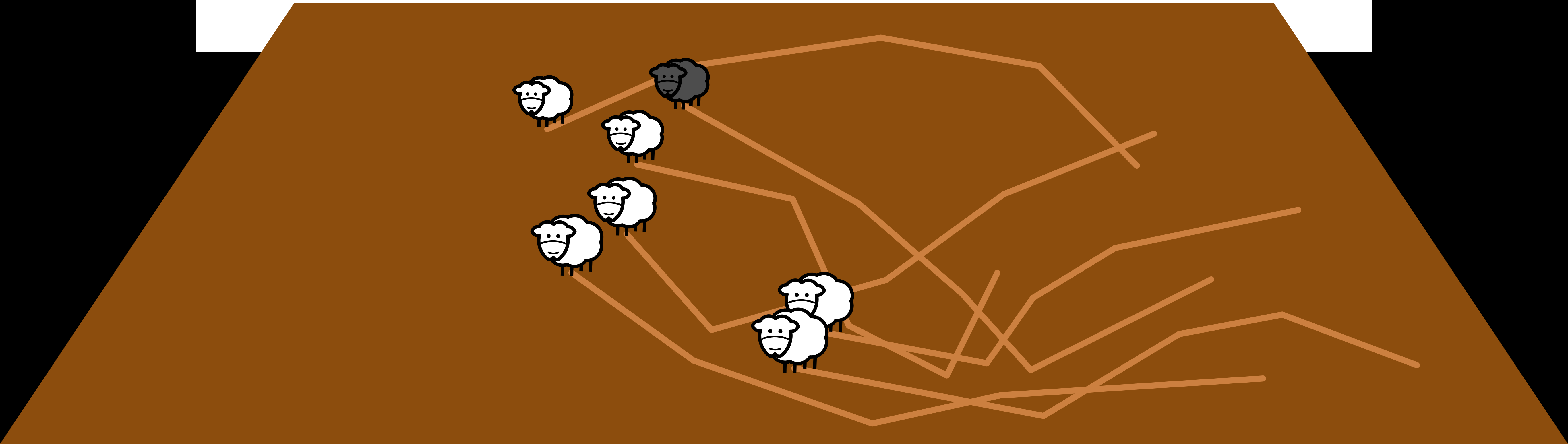
TRAJECTORIES

- Let P be n points in the plane
- Now suppose your points run away



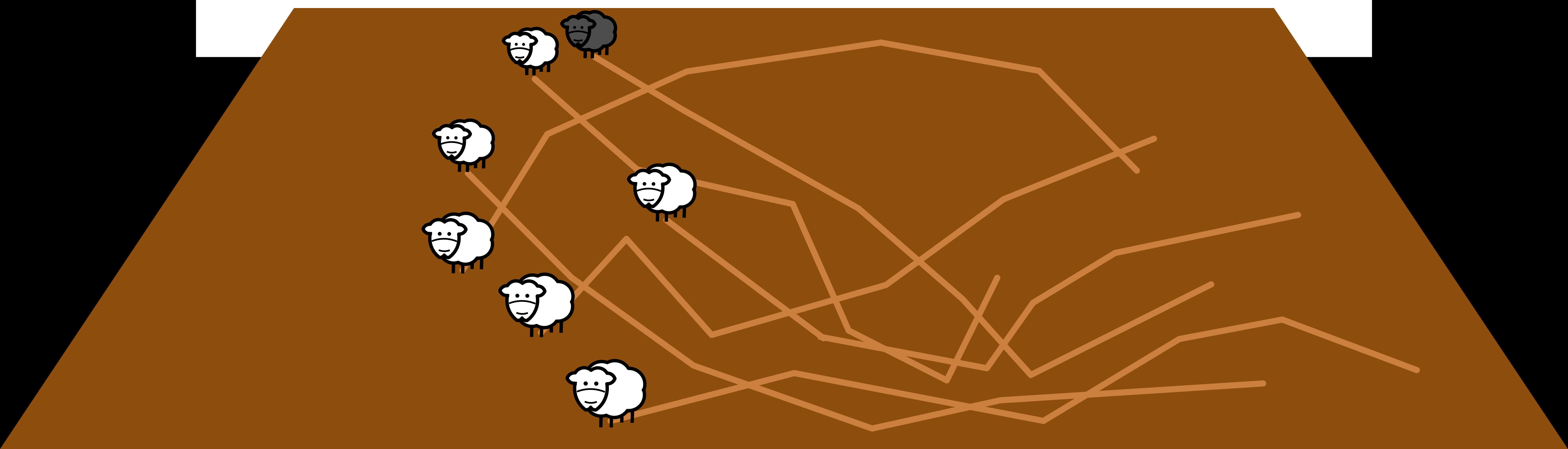
TRAJECTORIES

- Let P be n points in the plane
- Now suppose your points run away



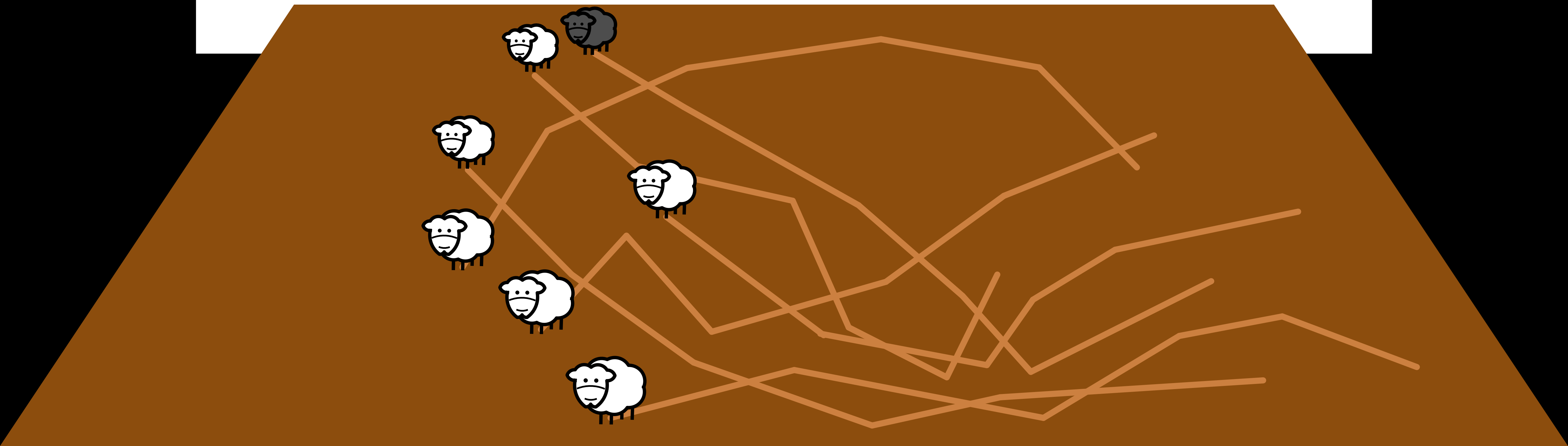
TRAJECTORIES

- Let P be n points in the plane
- Now suppose your points run away



TRAJECTORIES

- Let P be n points in the plane
- Now suppose your points run away
- P traces a set of n *trajectories*

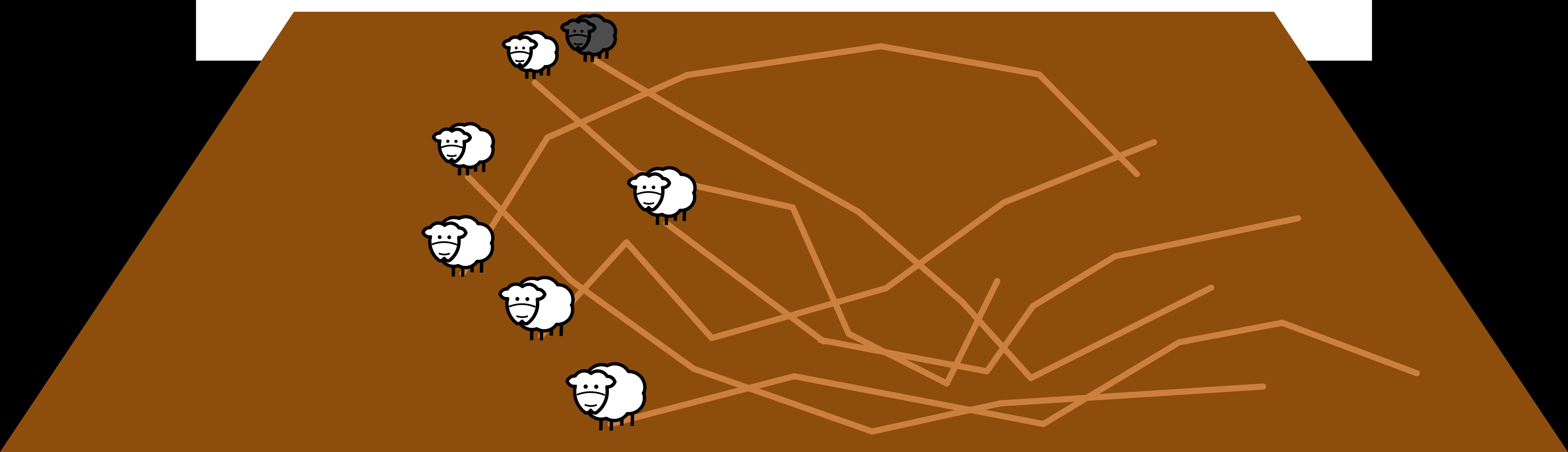


TRAJECTORIES

- Let P be n points in the plane
- Now suppose your points run away
- P traces a set of n *trajectories*

DEFINITION.

A *trajectory* is a polyline defined by a sequence of points, each with a location in $\mathbb{R}^d$ and a time stamp in $\mathbb{R}$.



TRAJECTORIES

TRAJECTORIES

- Trajectories are ubiquitous

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists

TRAJECTORIES

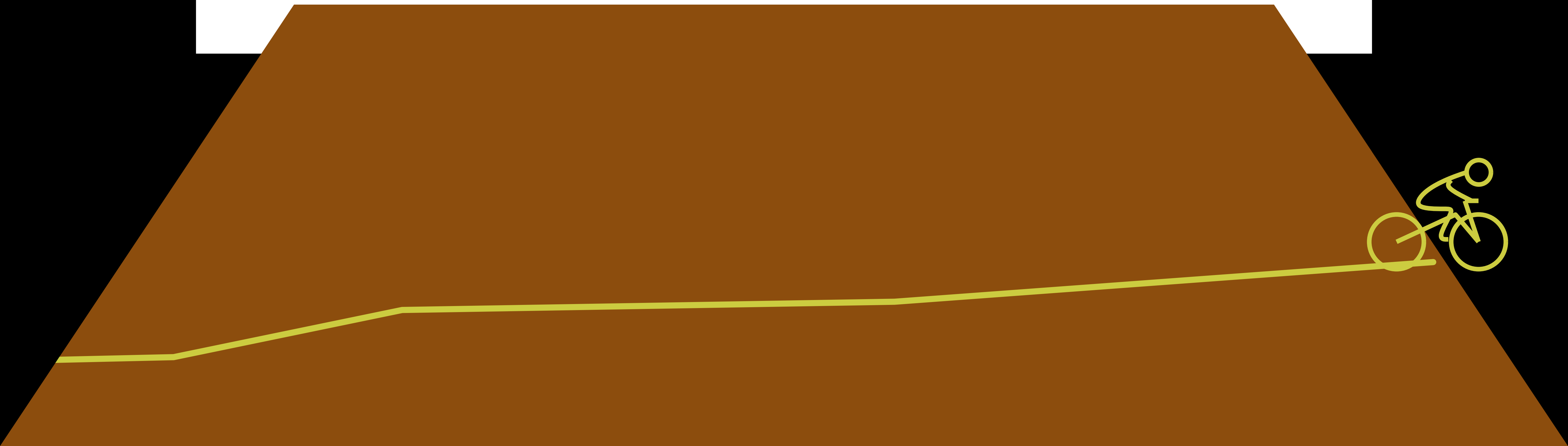
- Trajectories are ubiquitous
  - GPS technology
  - Cyclists

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists



TRAJECTORIES

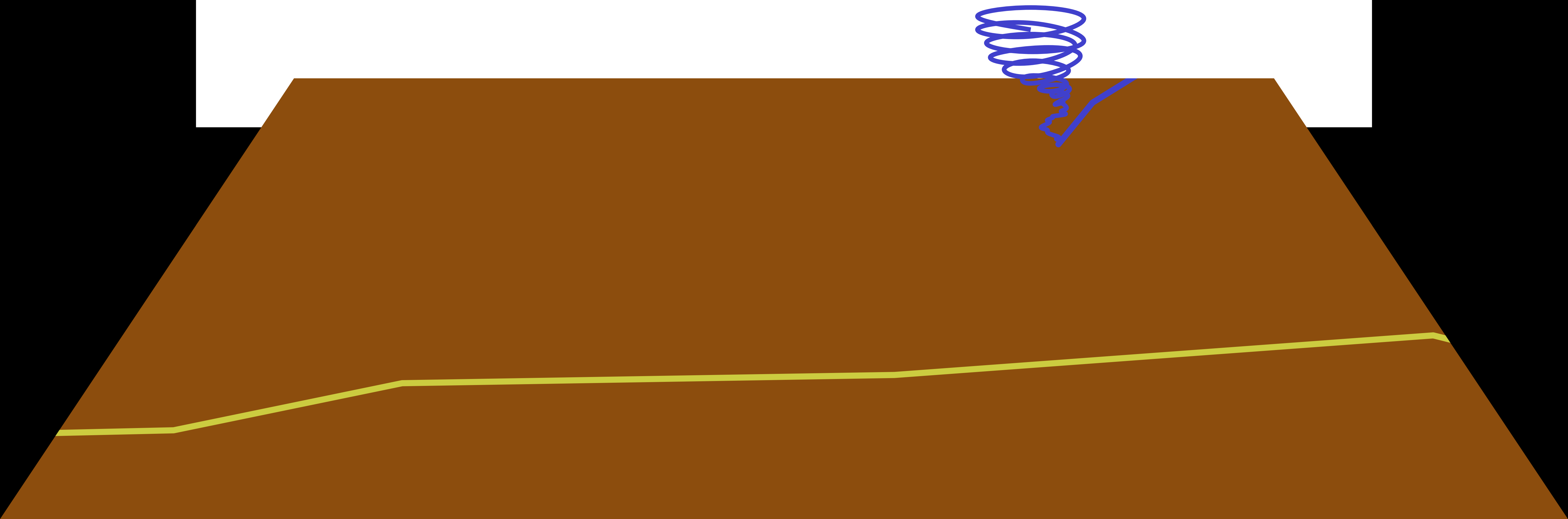
- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes

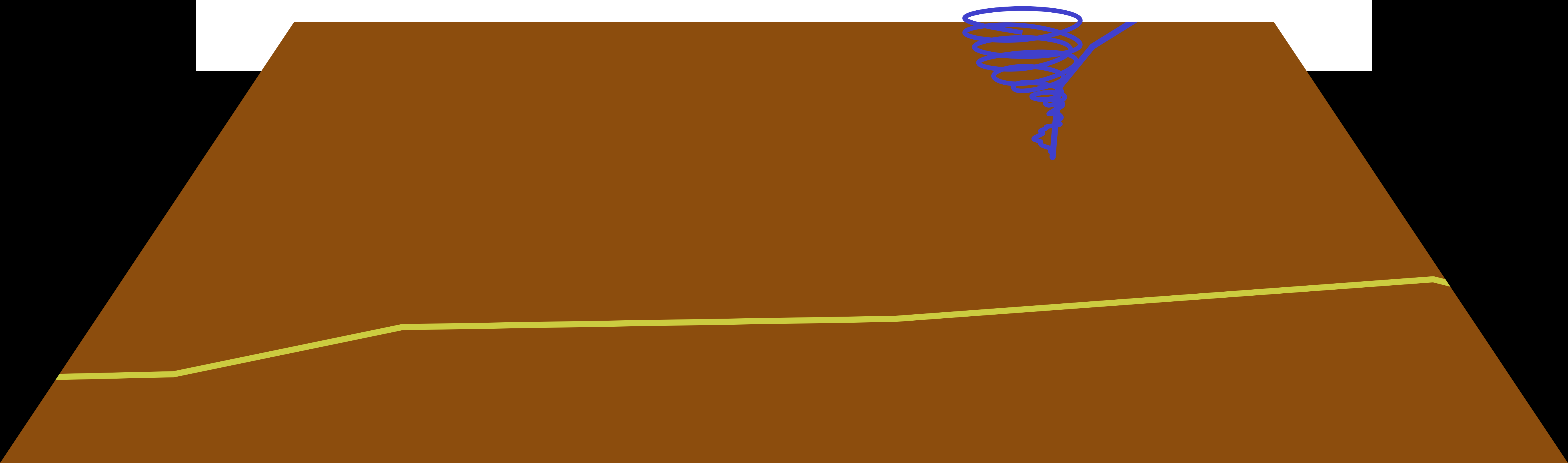
TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes



TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes



TRAJECTORIES

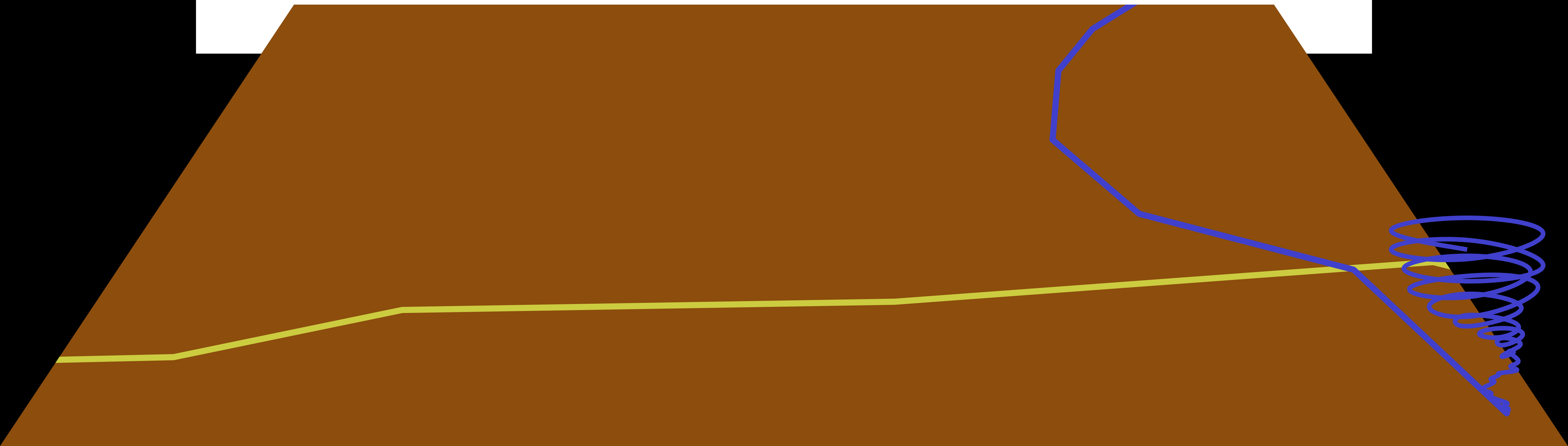
- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes

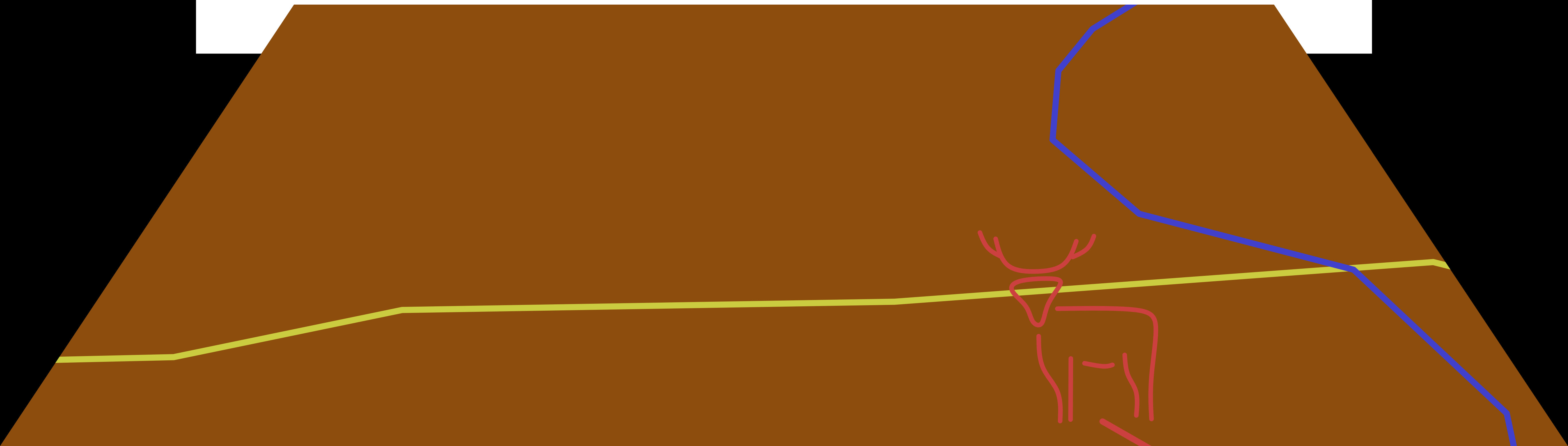


TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer



TRAJECTORIES

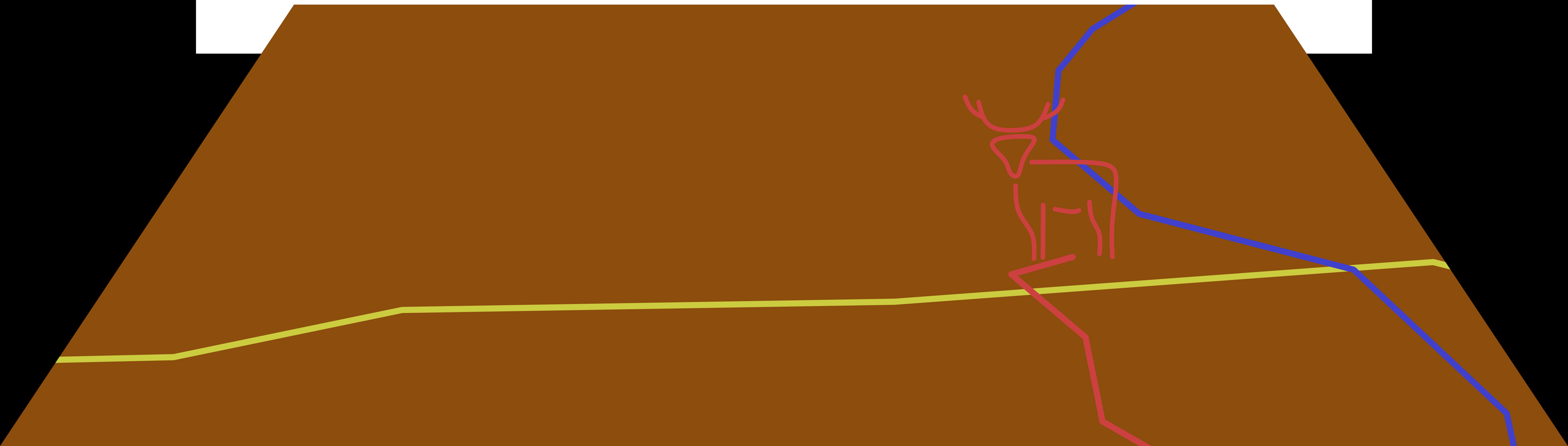
- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer

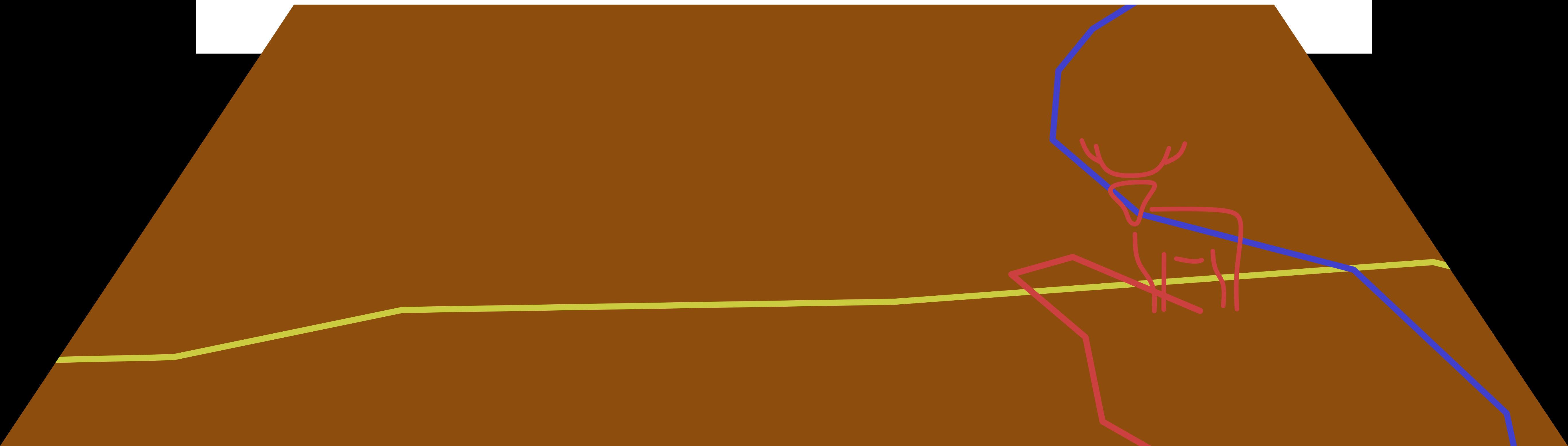
TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer



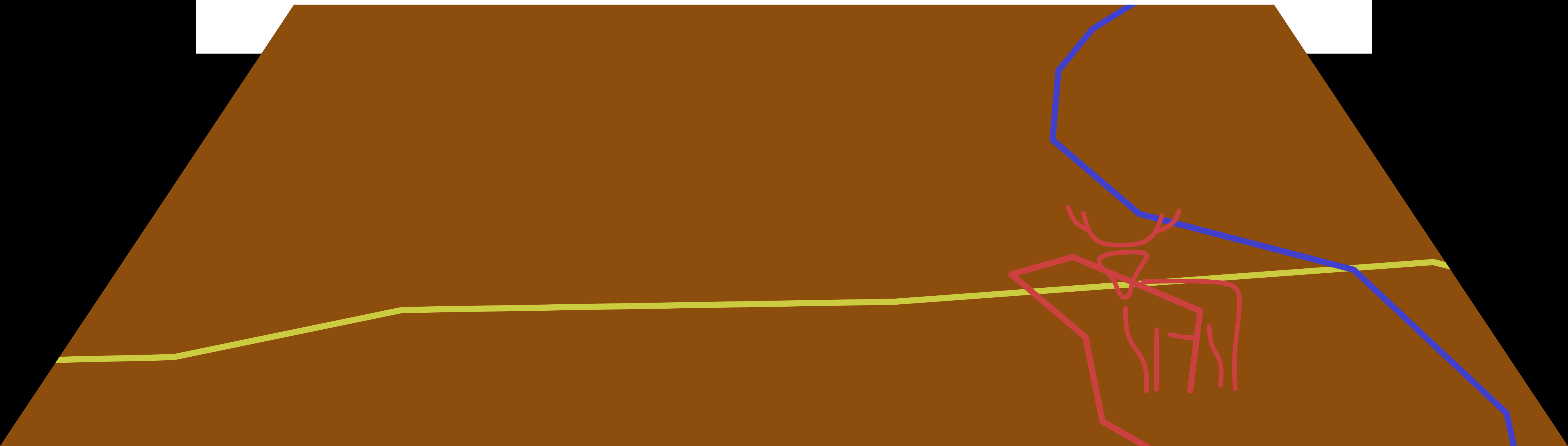
TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer



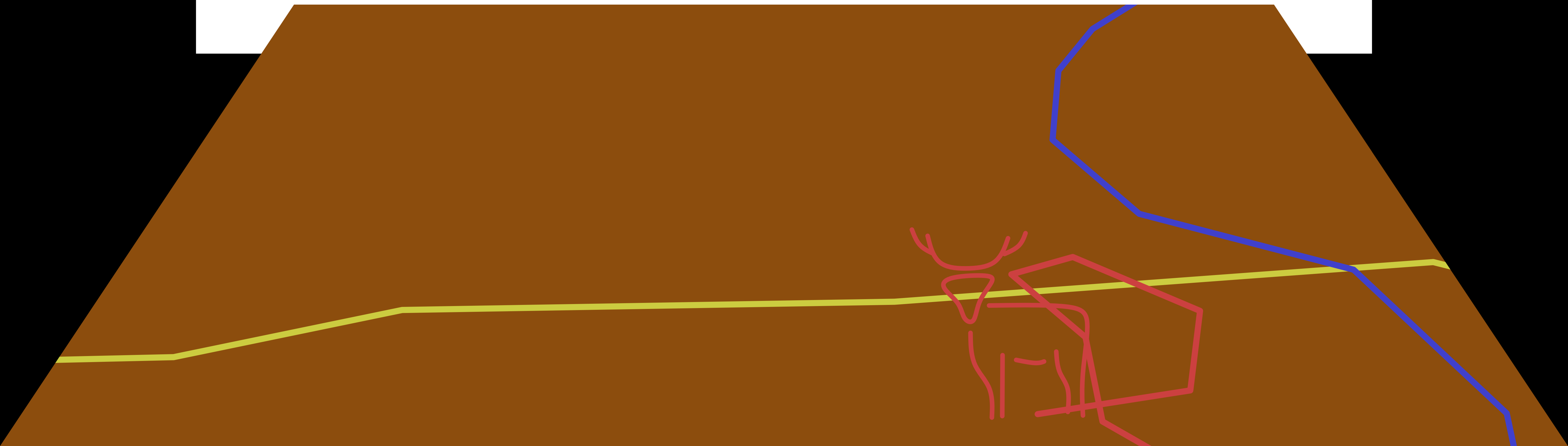
TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer



TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer



TRAJECTORIES

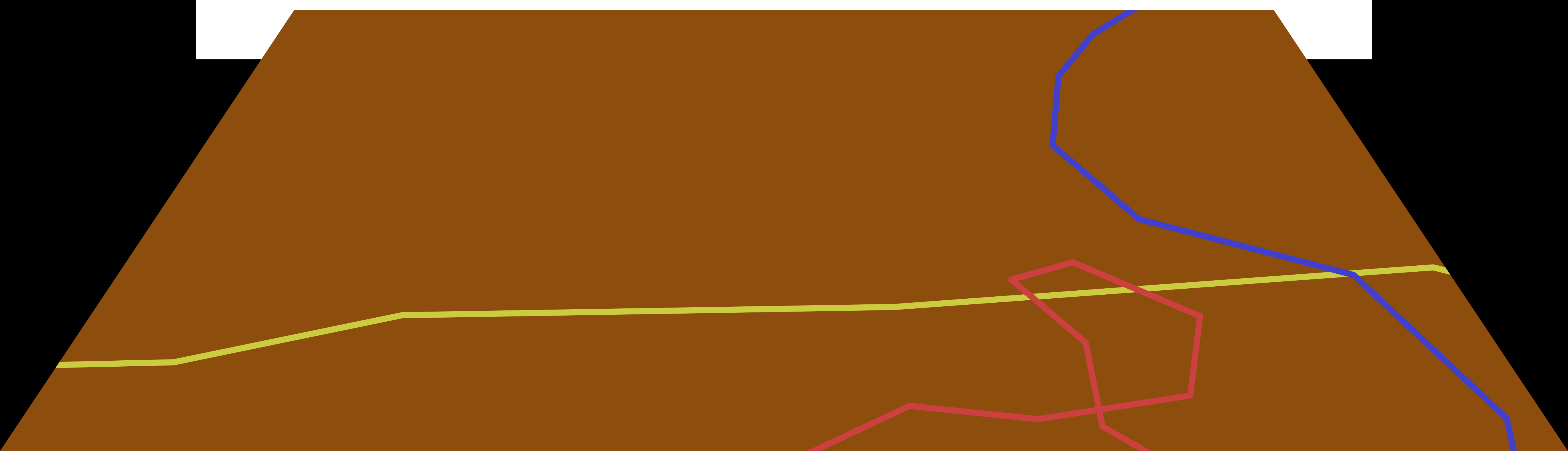
- Trajectories are ubiquitous
    - GPS technology
    - Cyclists
    - Hurricanes
    - Deer

TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer

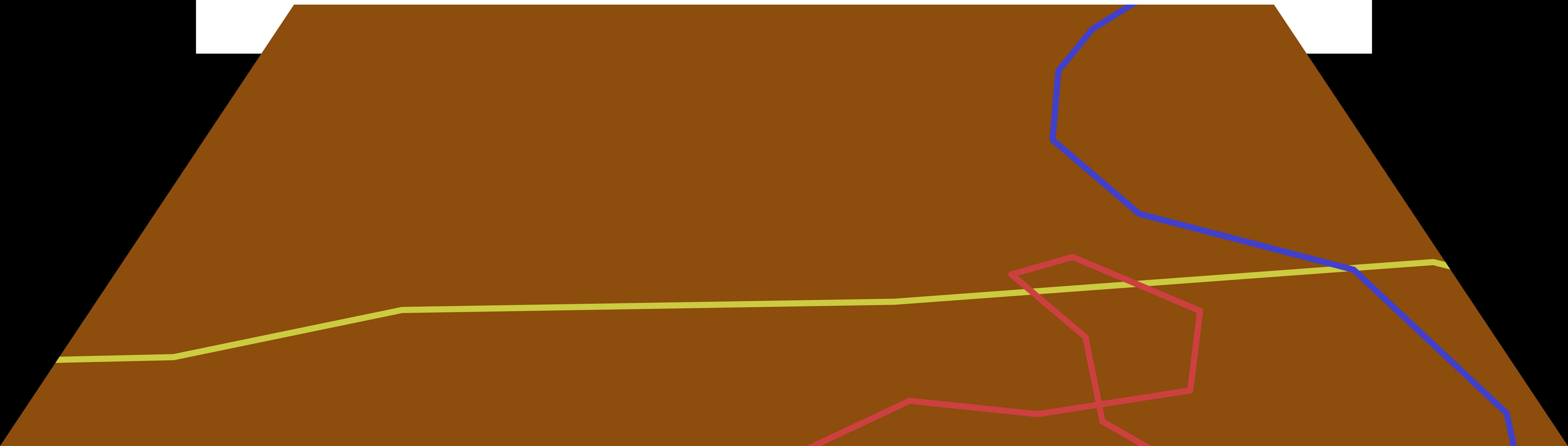
TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer
- Trajectories are interesting



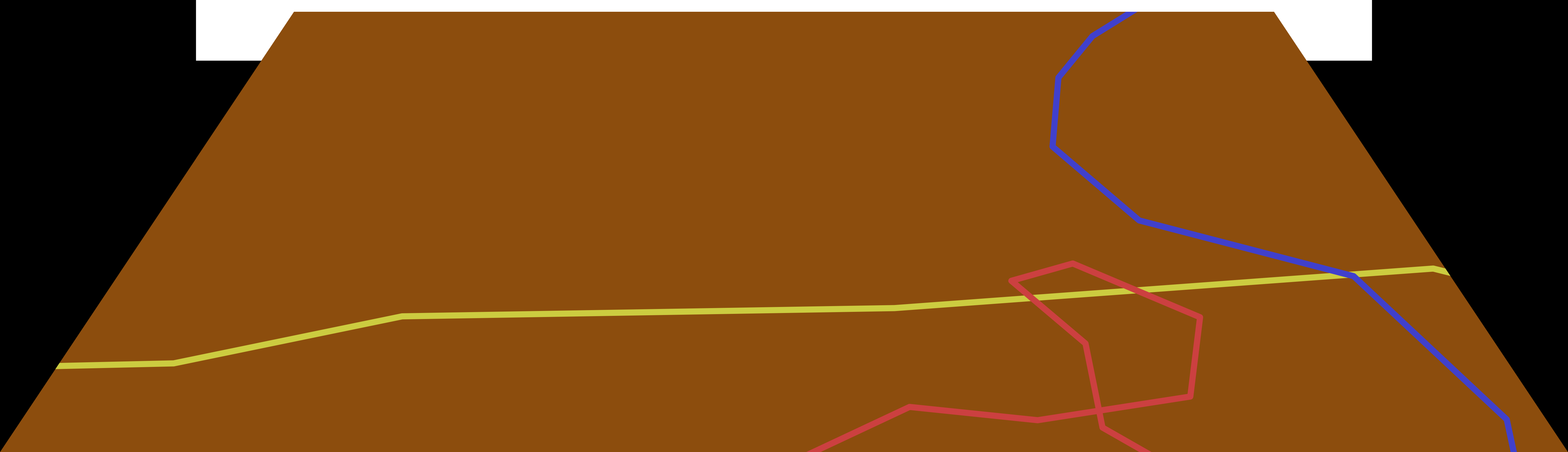
TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer
- Trajectories are interesting
  - Many different analysis tasks



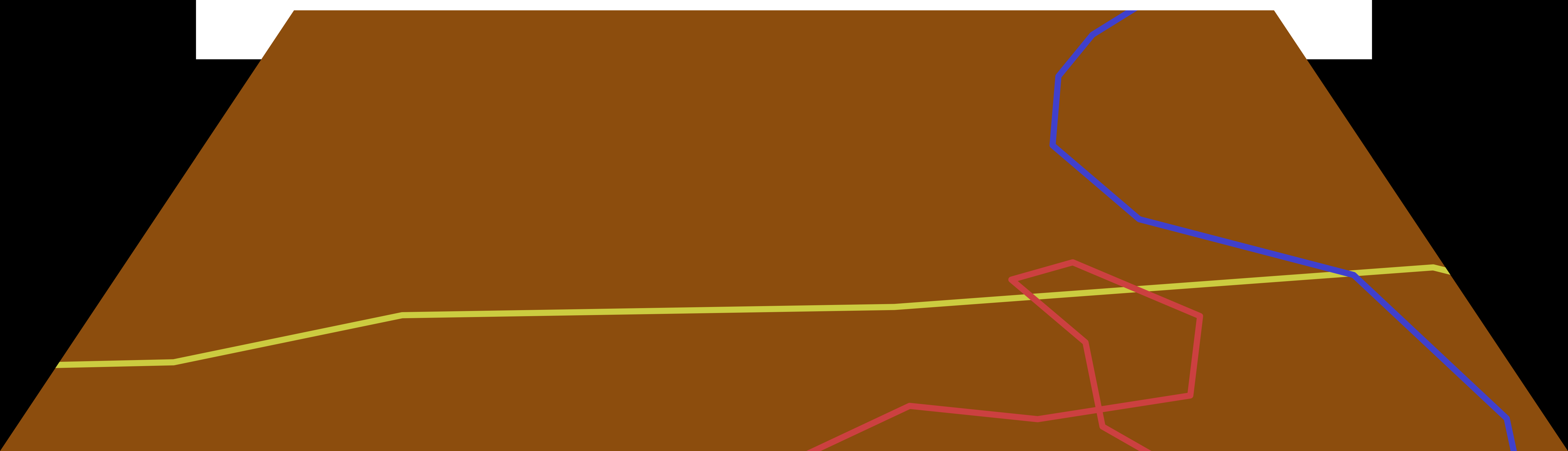
TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer
- Trajectories are interesting
  - Many different analysis tasks
  - Complex geometry



TRAJECTORIES

- Trajectories are ubiquitous
  - GPS technology
  - Cyclists
  - Hurricanes
  - Deer
- Trajectories are interesting
  - Many different analysis tasks
  - Complex geometry
  - Lots of fun!



REPRESENTATIVE TRAJECTORY

REPRESENTATIVE TRAJECTORY

- Problem

REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories

REPRESENTATIVE TRAJECTORY

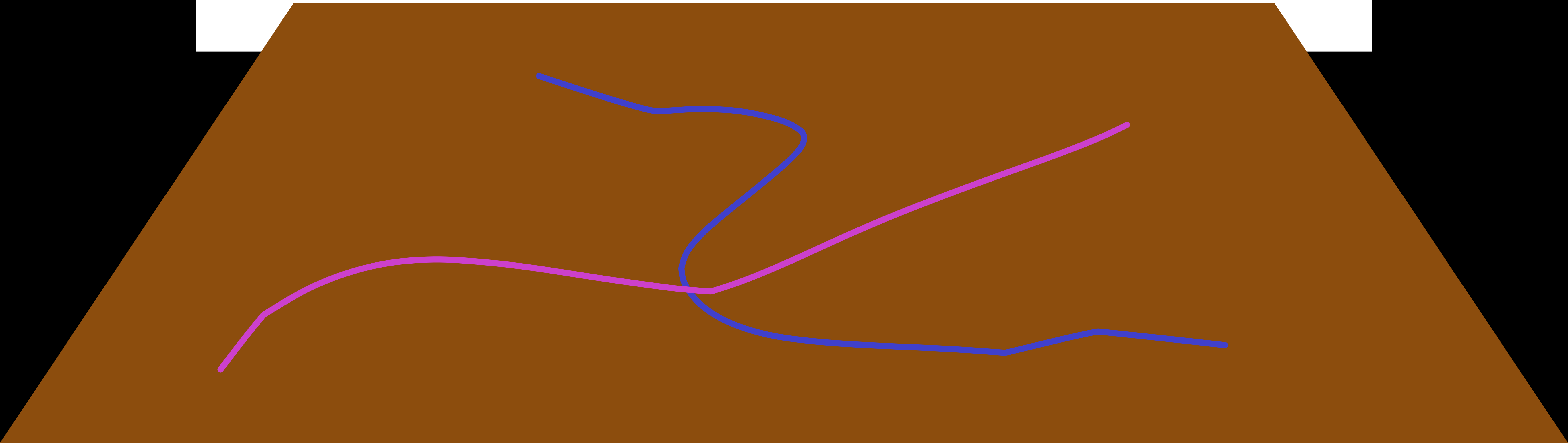
- Problem
  - Suppose we have lots of trajectories

REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns

REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns



REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns

REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns
- Solution

REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns
- Solution
  - Cluster the trajectories

REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns
- Solution
  - Cluster the trajectories

REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns
- Solution
  - Cluster the trajectories
  - Pick a good representative for each cluster

REPRESENTATIVE TRAJECTORY

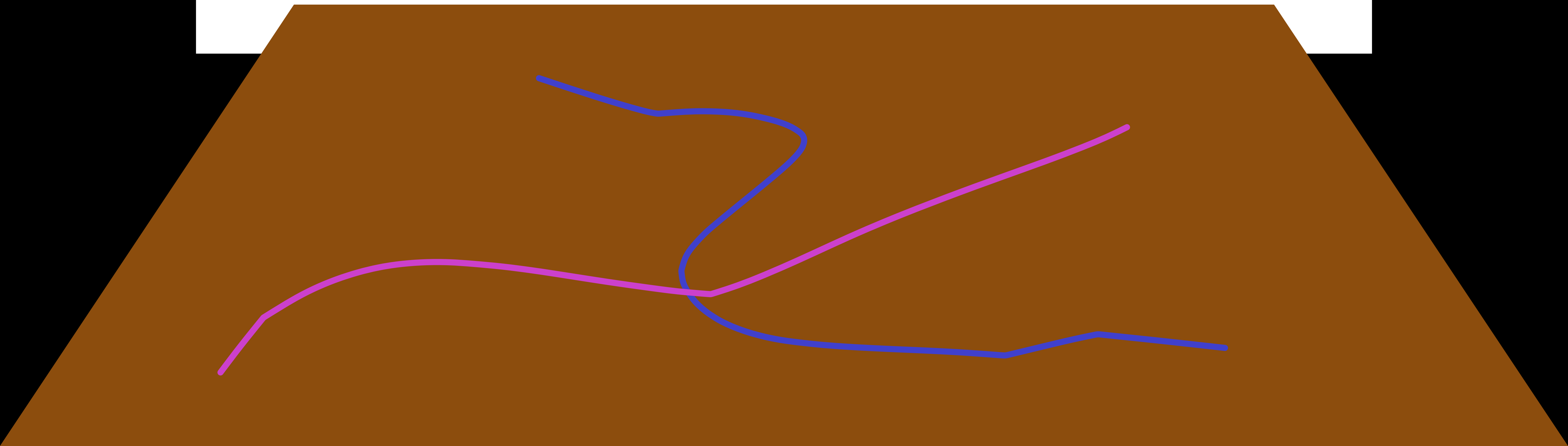
- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns
- Solution
  - Cluster the trajectories
  - Pick a good representative for each cluster

REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns
- Solution
  - Cluster the trajectories
  - Pick a good representative for each cluster
  - Keep only the representatives

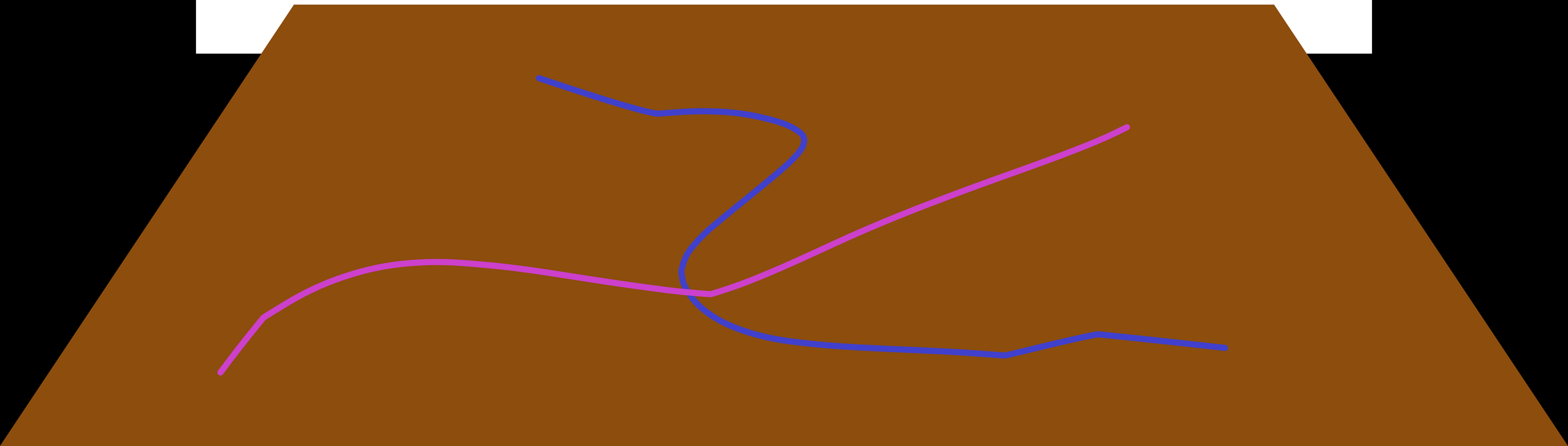
REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns
- Solution
  - Cluster the trajectories
  - Pick a good representative for each cluster
  - Keep only the representatives



REPRESENTATIVE TRAJECTORY

- Problem
  - Suppose we have lots of trajectories
  - Suppose we want to extract significant patterns
- Solution
  - Cluster the trajectories
  - Pick a good representative for each cluster
  - Keep only the representatives
- But what is a good representative?



REPRESENTATIVE TRAJECTORY

REPRESENTATIVE TRAJECTORY

- Input: a set of ‘similar’ trajectories

REPRESENTATIVE TRAJECTORY

- Input: a set of ‘similar’ trajectories
  - Sort of the same start and end points

REPRESENTATIVE TRAJECTORY

- Input: a set of 'similar' trajectories
  - Sort of the same start and end points
  - Sort of the same shape

REPRESENTATIVE TRAJECTORY

- Input: a set of 'similar' trajectories
  - Sort of the same start and end points
  - Sort of the same shape

REPRESENTATIVE TRAJECTORY

- Input: a set of ‘similar’ trajectories
  - Sort of the same start and end points
  - Sort of the same shape
- Output: a representative trajectory

REPRESENTATIVE TRAJECTORY

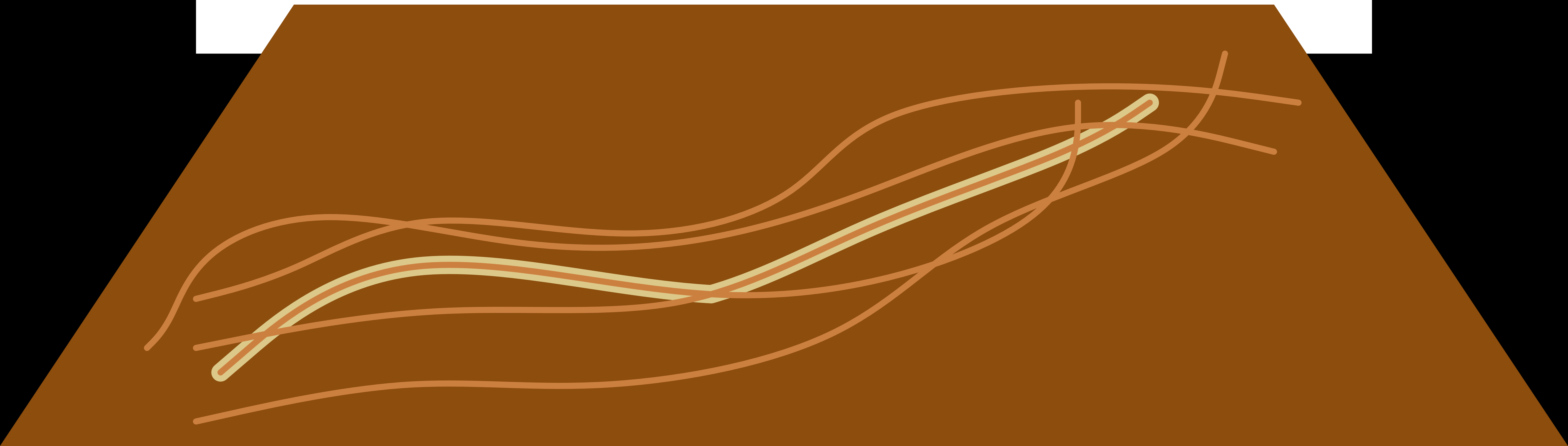
- Input: a set of 'similar' trajectories
  - Sort of the same start and end points
  - Sort of the same shape
- Output: a representative trajectory
  - Should also have sort of the same shape

REPRESENTATIVE TRAJECTORY

- Input: a set of 'similar' trajectories
  - Sort of the same start and end points
  - Sort of the same shape
- Output: a representative trajectory
  - Should also have sort of the same shape
  - Shape should represent the whole set of input trajectories

REPRESENTATIVE TRAJECTORY

- Input: a set of 'similar' trajectories
  - Sort of the same start and end points
  - Sort of the same shape
- Output: a representative trajectory
  - Should also have sort of the same shape
  - Shape should represent the whole set of input trajectories



USE INPUT TRAJECTORY?

USE INPUT TRAJECTORY?

USE INPUT TRAJECTORY?

- No?

USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory

USE INPUT TRAJECTORY?

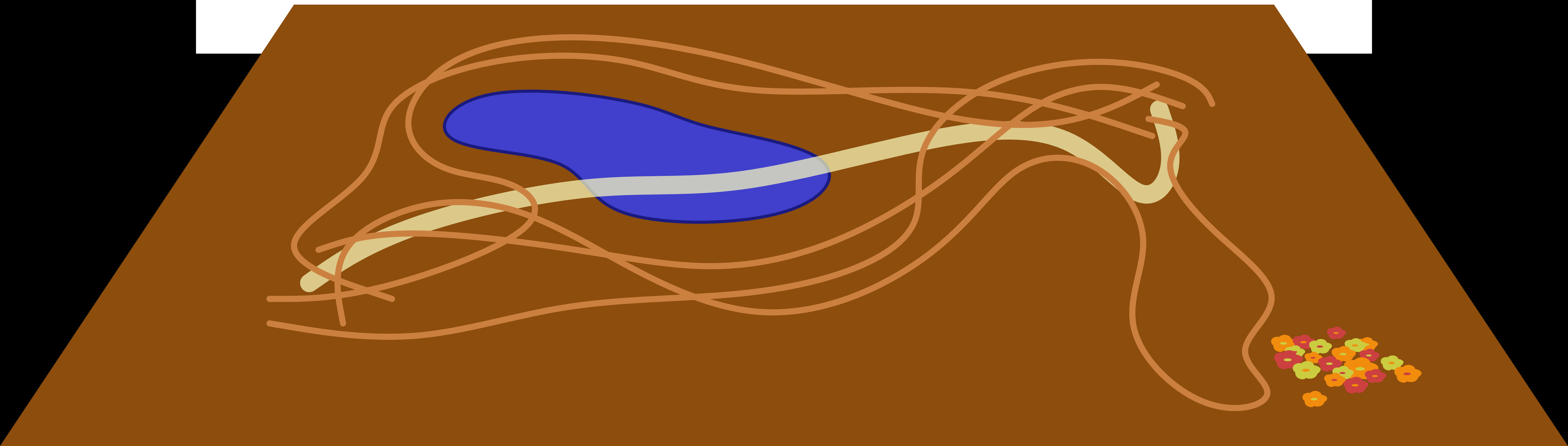
- No?
  - Parameterised mean trajectory

USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!

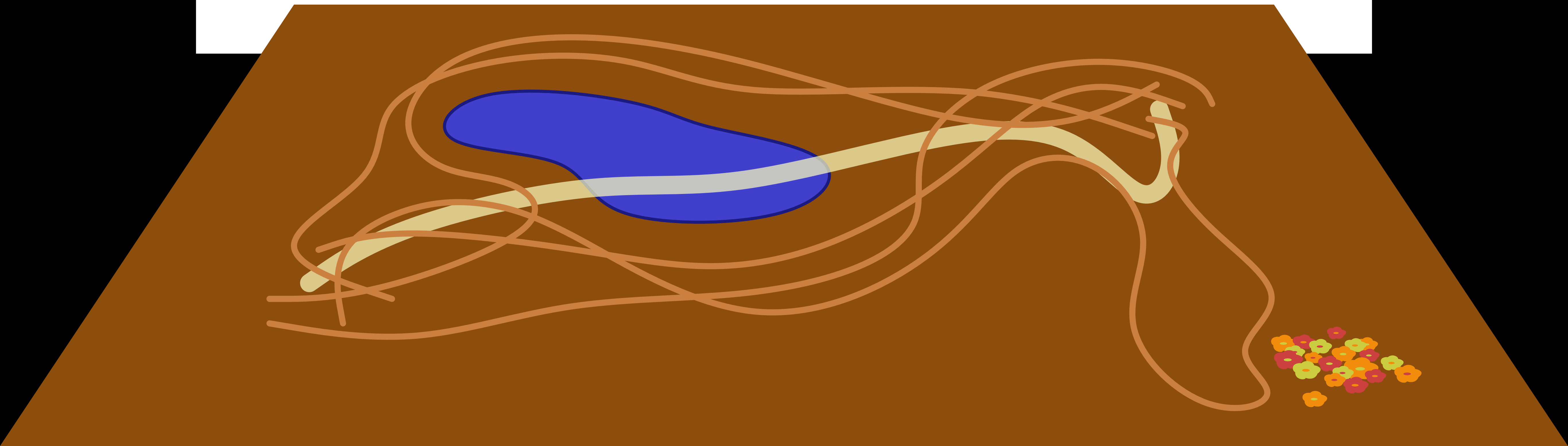
USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!



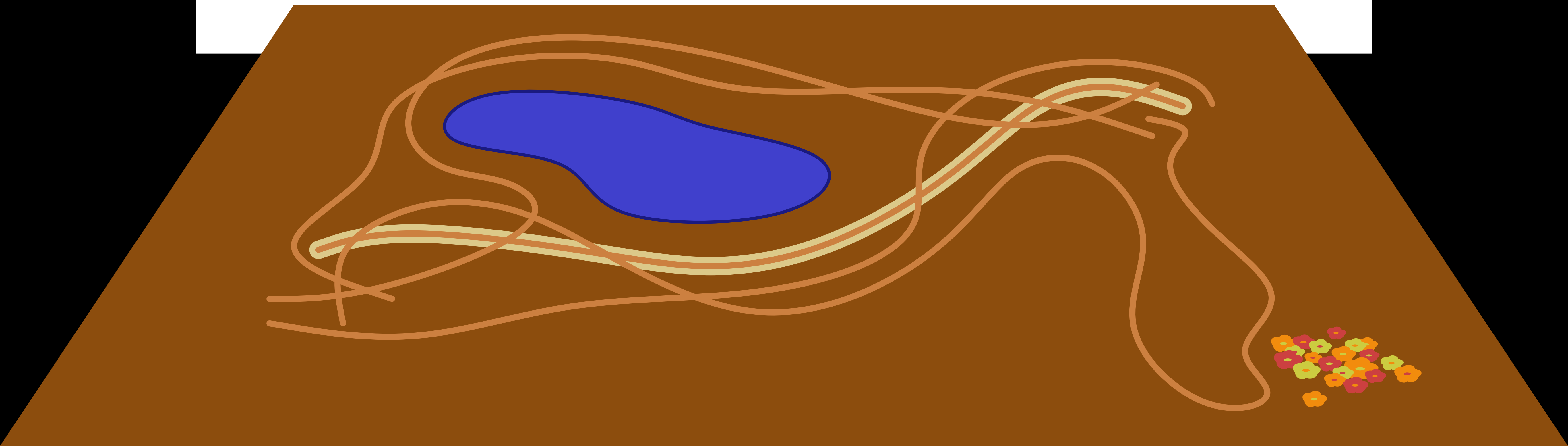
USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!
- Yes?



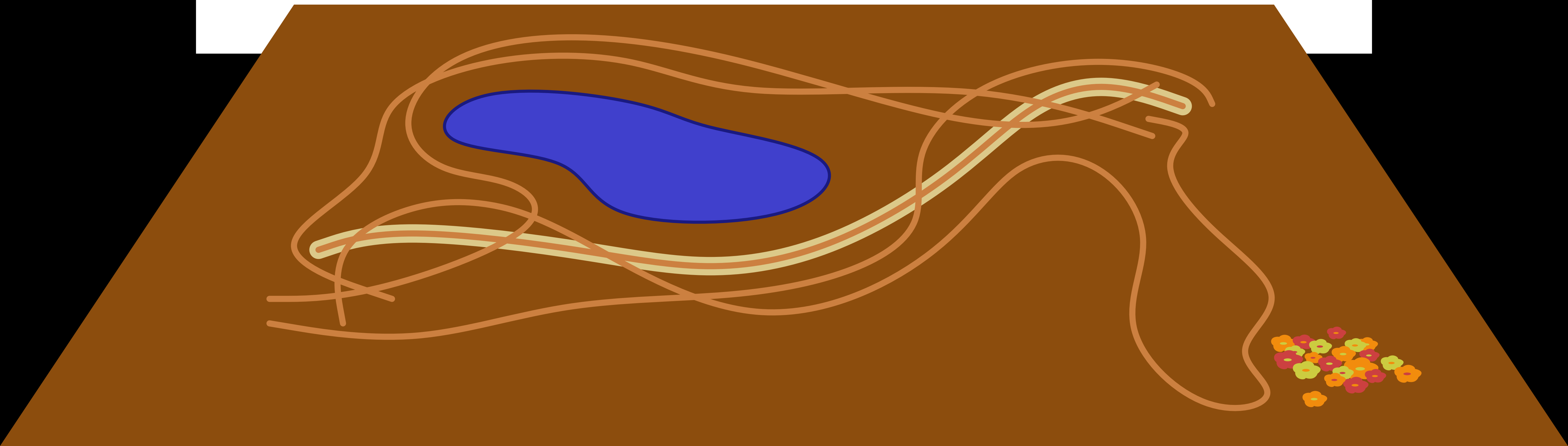
USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!
- Yes?



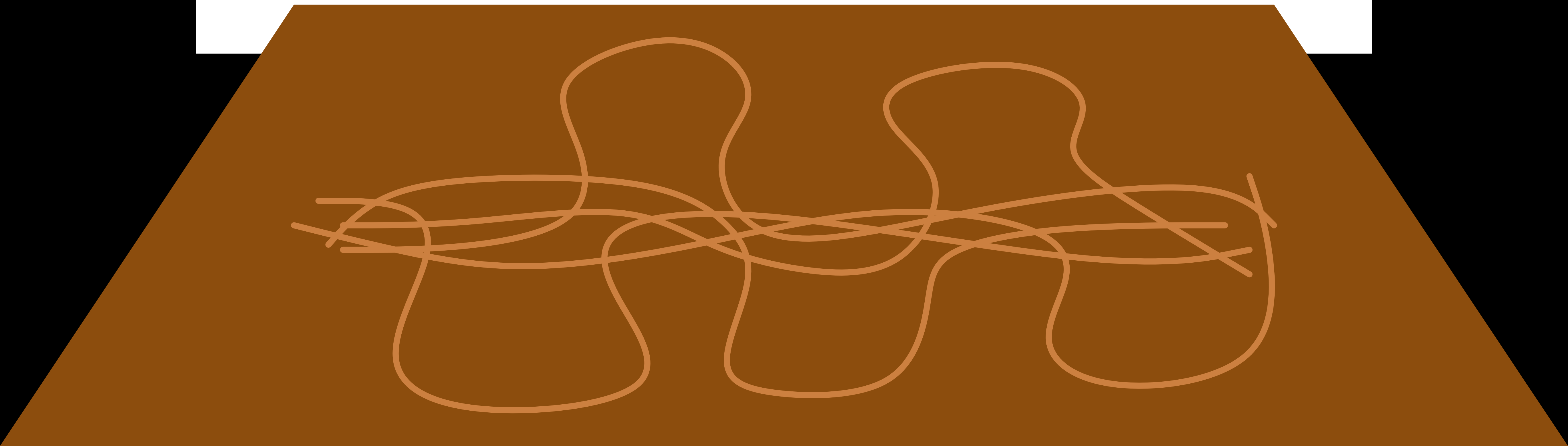
USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!
- Yes?
  - Which trajectory do we pick?



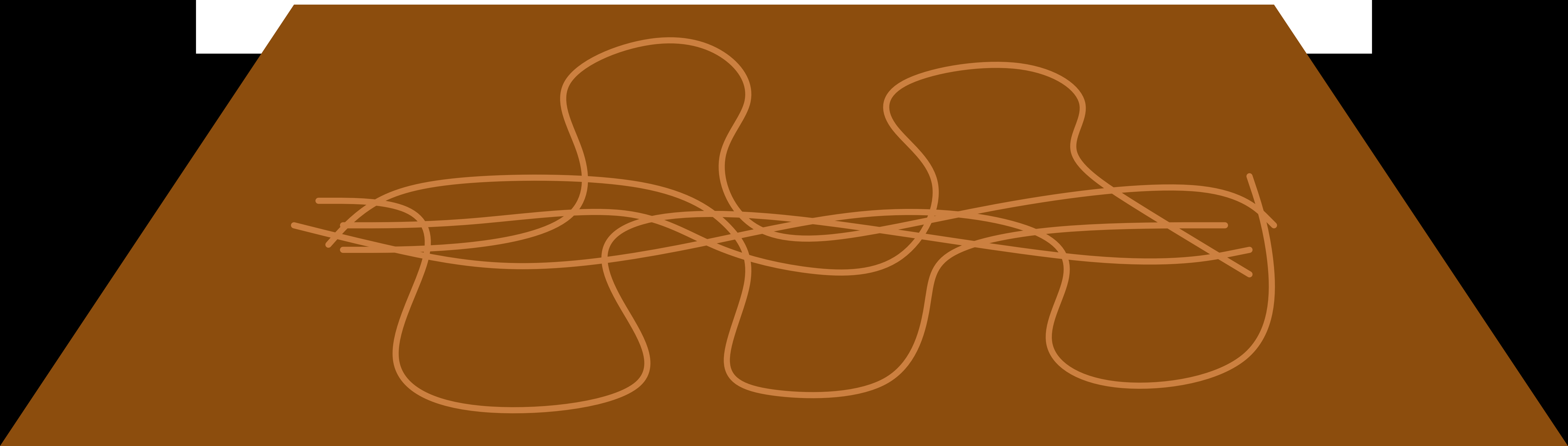
USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!
- Yes?
  - Which trajectory do we pick?



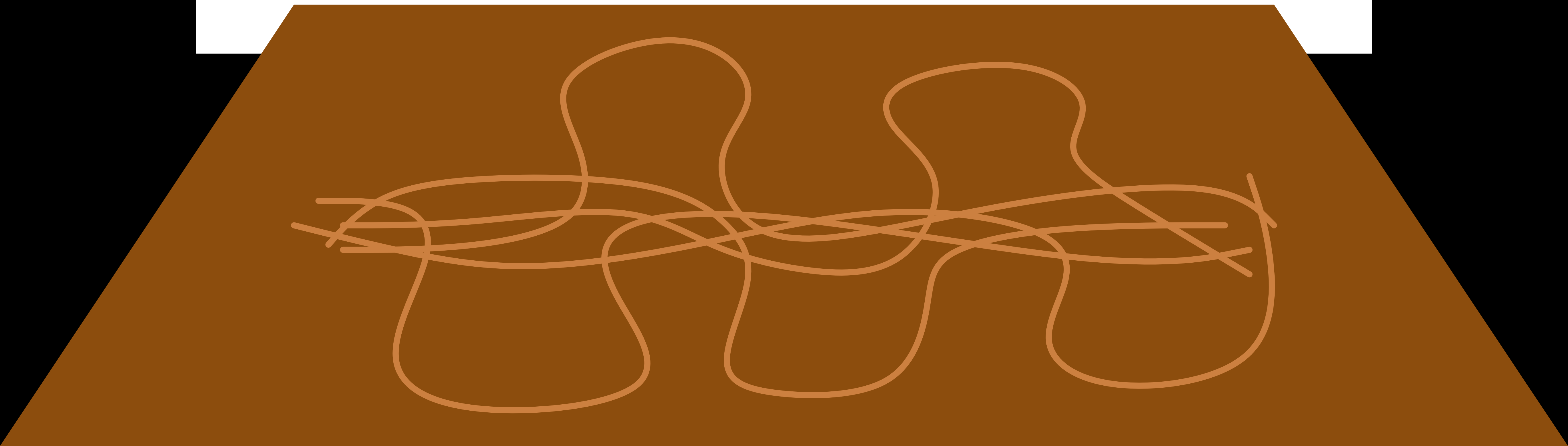
USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!
- Yes?
  - Which trajectory do we pick?
  - There may not be any single good representative!



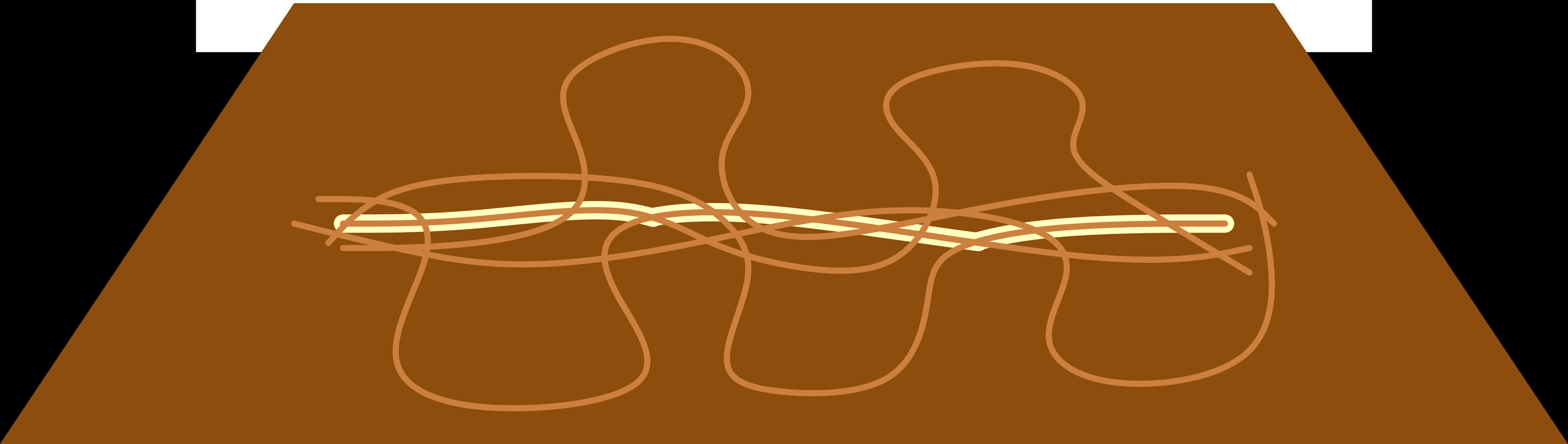
USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!
- Yes?
  - Which trajectory do we pick?
  - There may not be any single good representative!
- Use pieces of different trajectories?



USE INPUT TRAJECTORY?

- No?
  - Parameterised mean trajectory
  - May interfere with environment!
- Yes?
  - Which trajectory do we pick?
  - There may not be any single good representative!
- Use pieces of different trajectories?



USE INPUT TRAJECTORY?

- What if input trajectories do not cross?

USE INPUT TRAJECTORY?

- What if input trajectories do not cross?

USE INPUT TRAJECTORY?

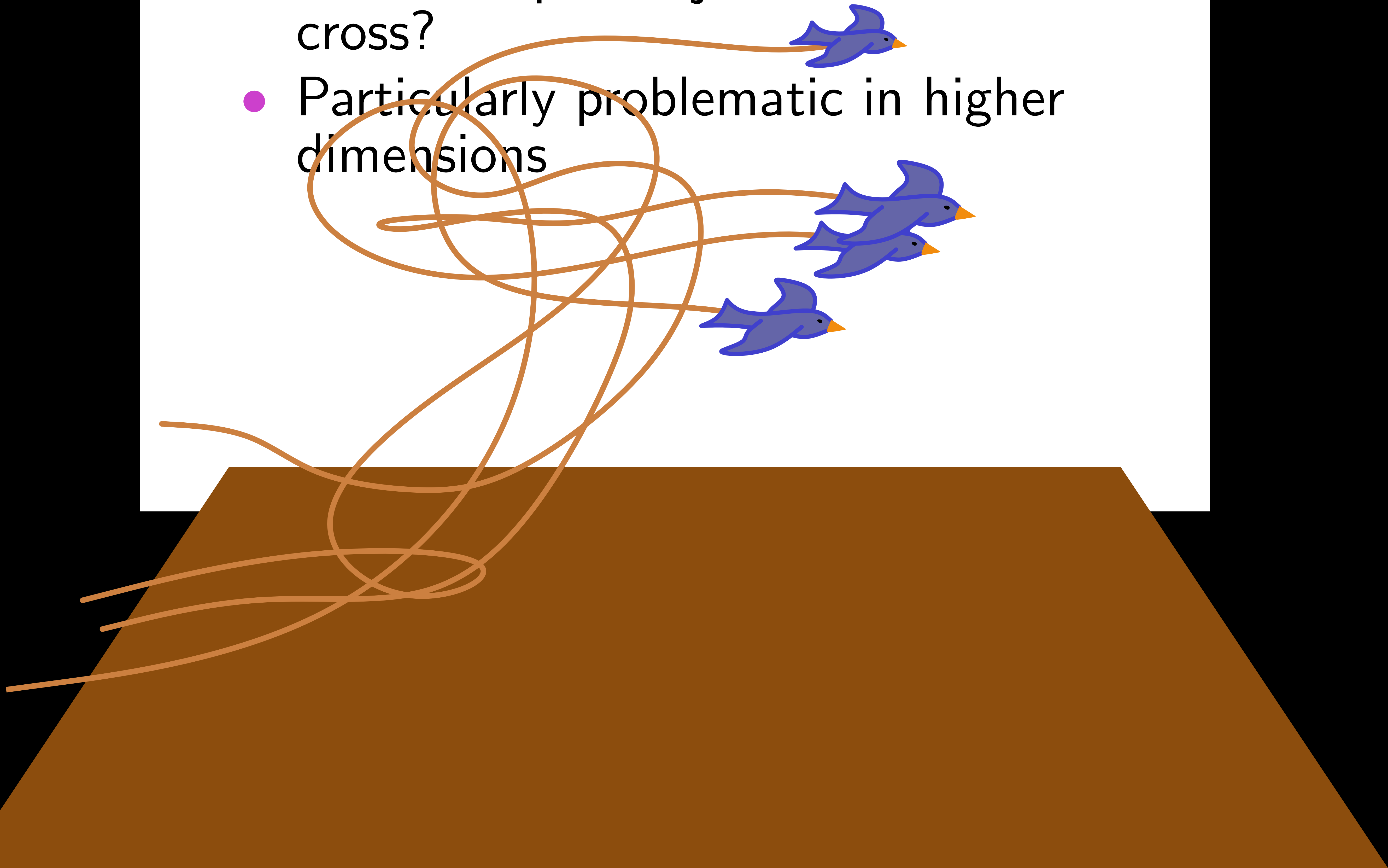
- What if input trajectories do not cross?

USE INPUT TRAJECTORY?

- What if input trajectories do not cross?
- Particularly problematic in higher dimensions

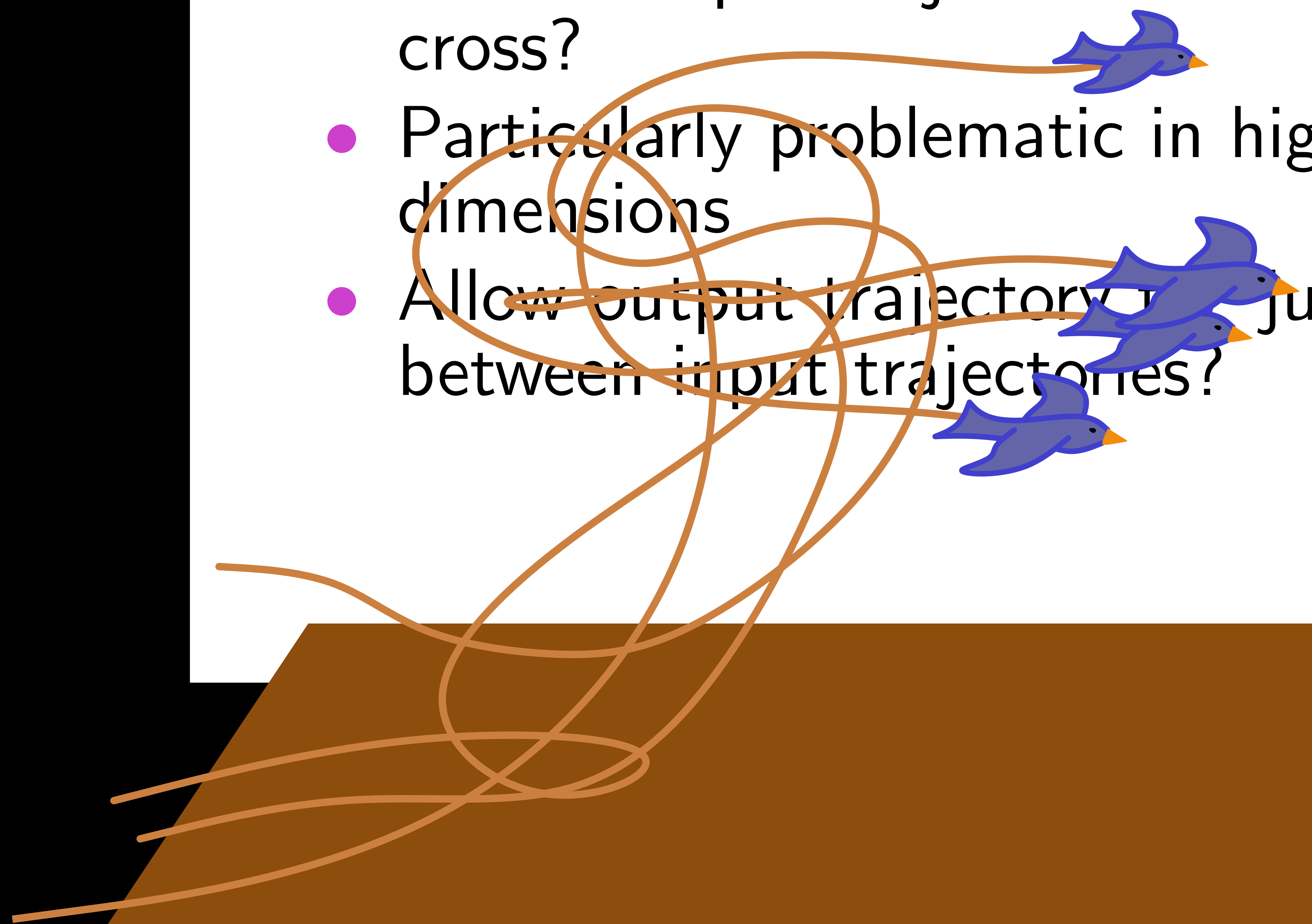
USE INPUT TRAJECTORY?

- What if input trajectories do not cross?
- Particularly problematic in higher dimensions



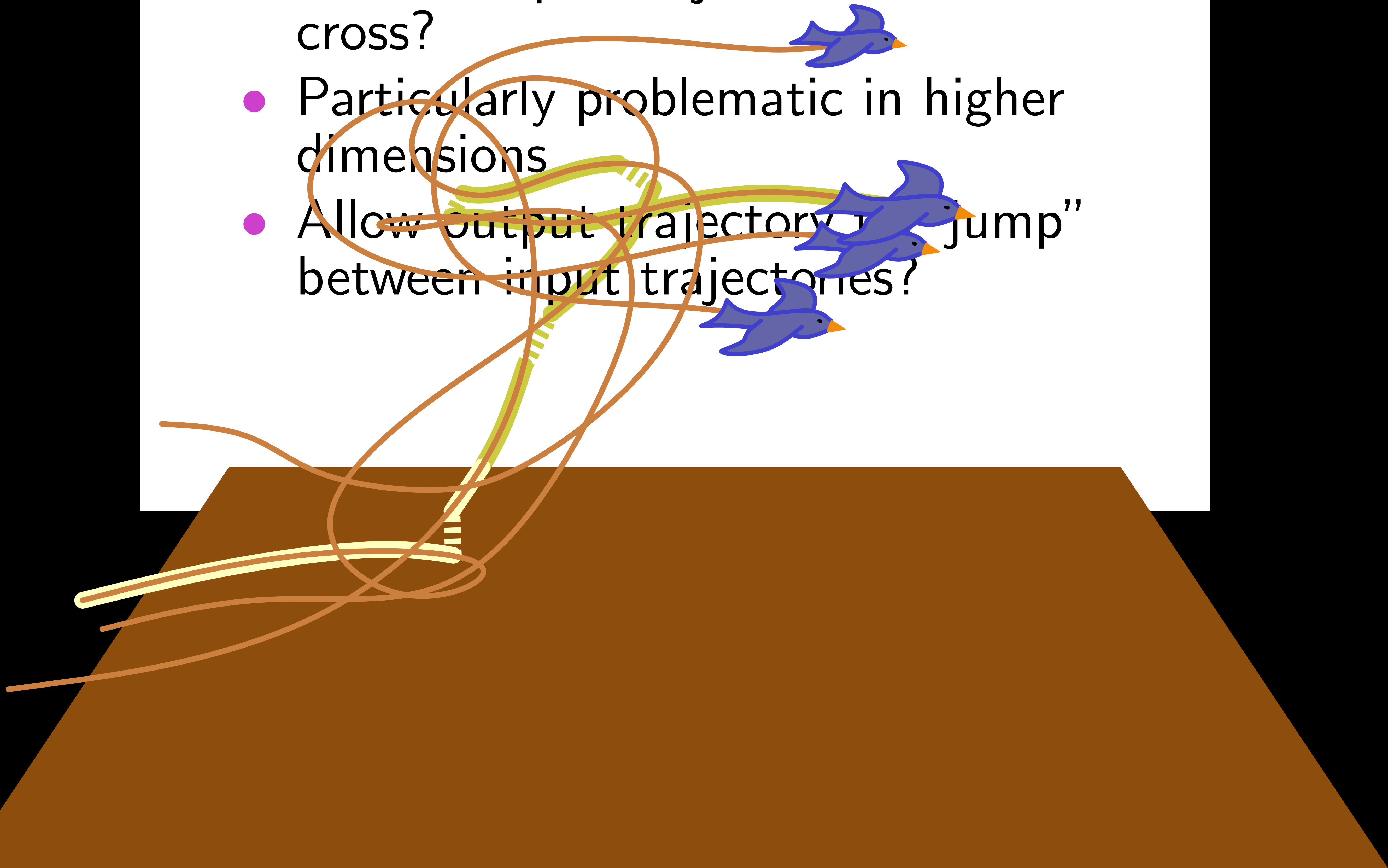
USE INPUT TRAJECTORY?

- What if input trajectories do not cross?
- Particularly problematic in higher dimensions
- Allow output trajectory to "jump" between input trajectories?



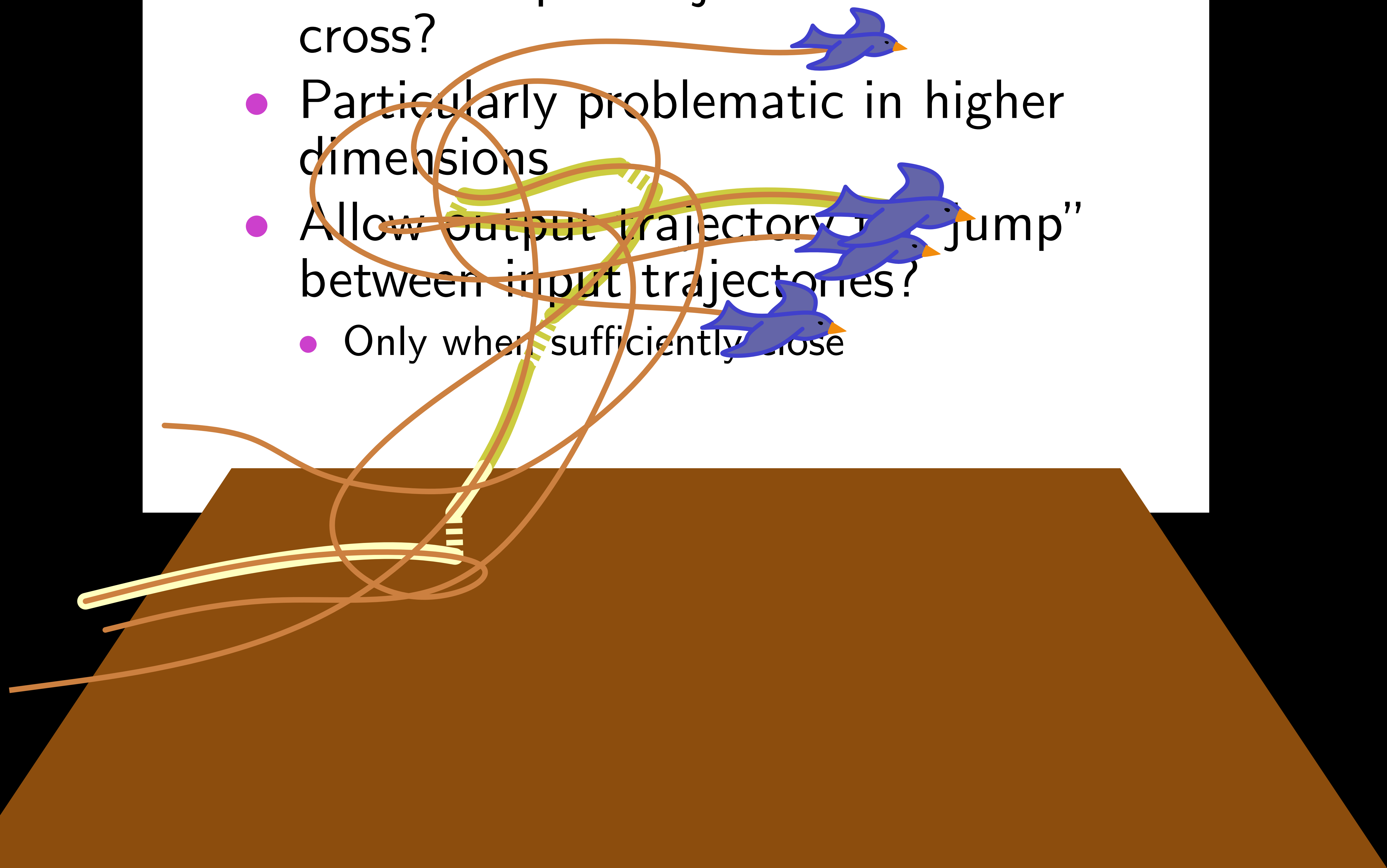
USE INPUT TRAJECTORY?

- What if input trajectories do not cross?
- Particularly problematic in higher dimensions
- Allow output trajectory to “jump” between input trajectories?



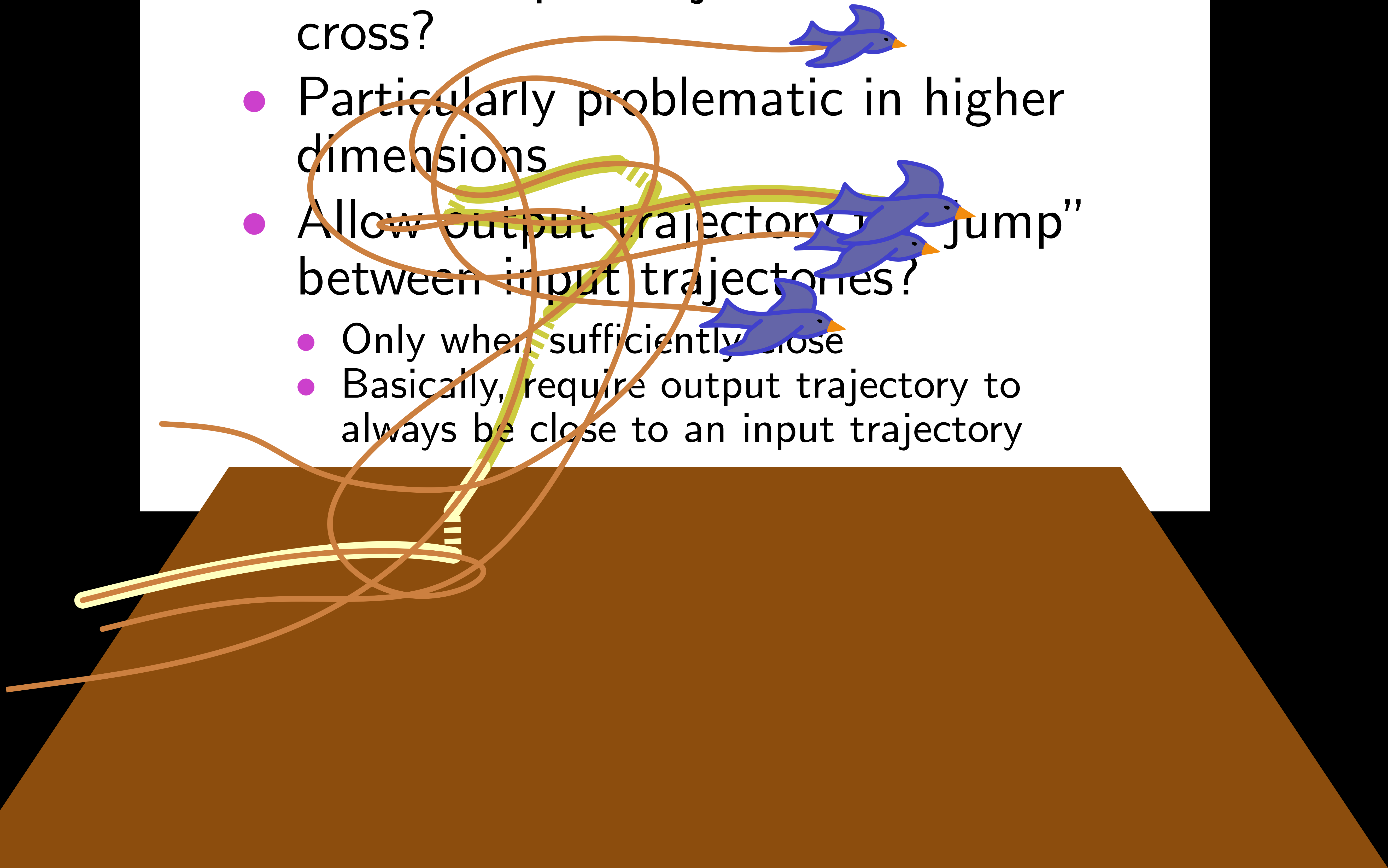
USE INPUT TRAJECTORY?

- What if input trajectories do not cross?
- Particularly problematic in higher dimensions
- Allow output trajectory to "jump" between input trajectories?
  - Only when sufficiently close



USE INPUT TRAJECTORY?

- What if input trajectories do not cross?
- Particularly problematic in higher dimensions
- Allow output trajectory to "jump" between input trajectories?
  - Only when sufficiently close
  - Basically, require output trajectory to always be close to an input trajectory



DO YOU HAVE TIME TO CARE?

DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations

DO YOU HAVE TIME TO CARE?

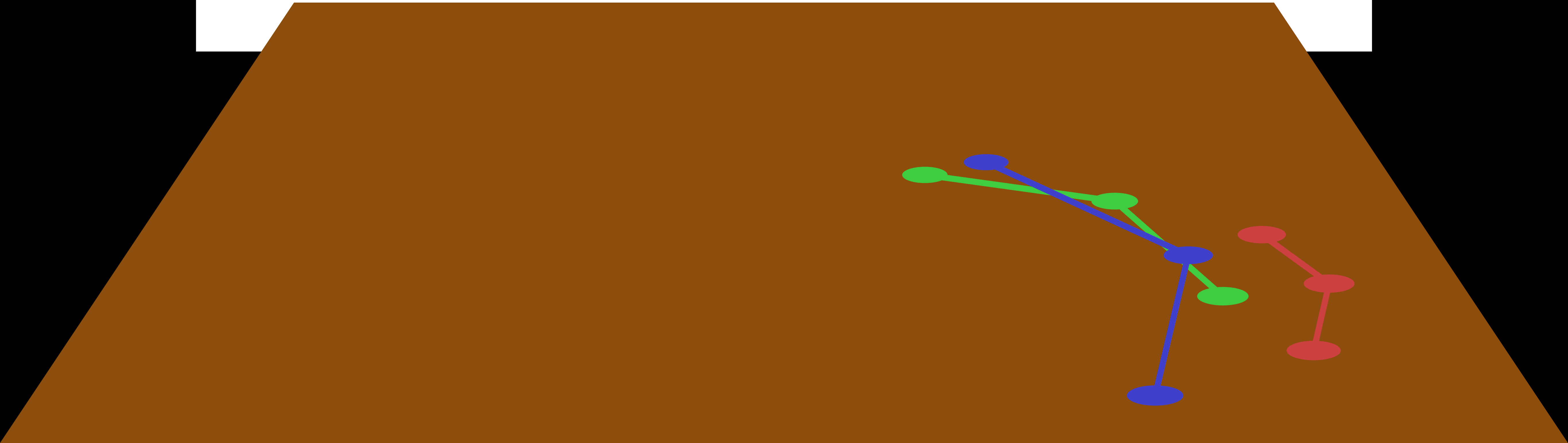
- Trajectories have *time-stamped* locations

DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations

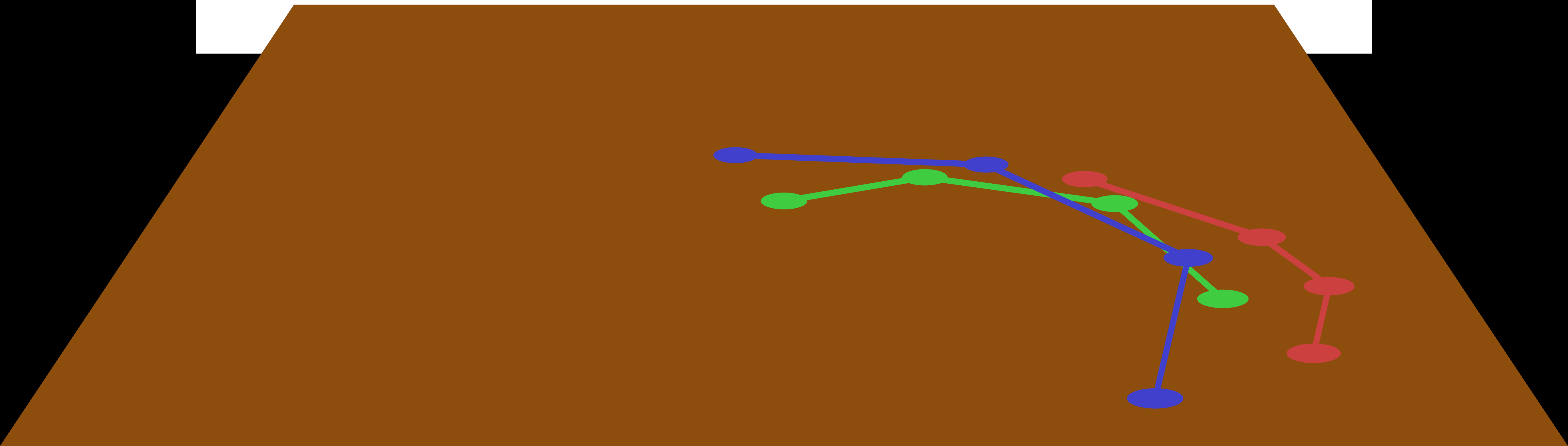
DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations



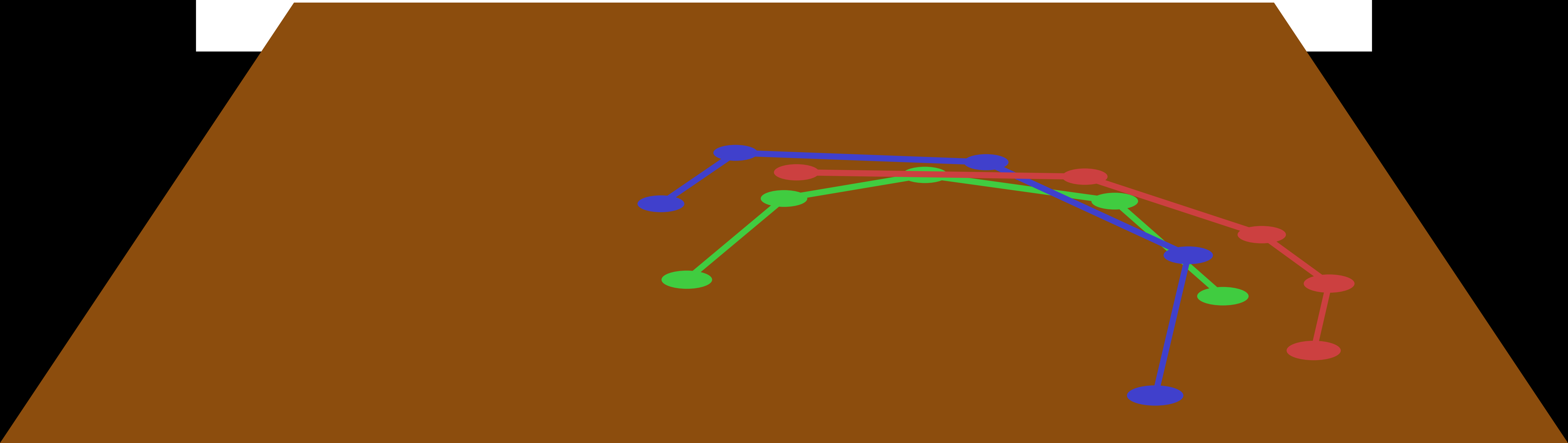
DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations



DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations



DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations

DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations

DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not

DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar

DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread

DO YOU HAVE TIME TO CARE?

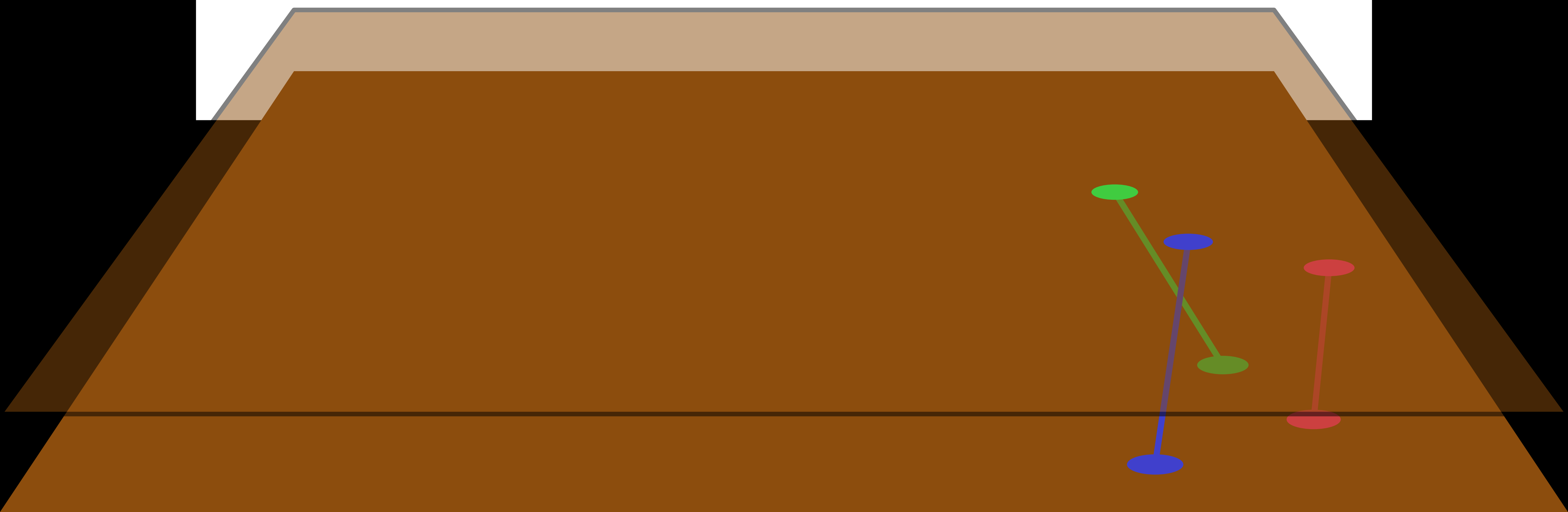
- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?

DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?

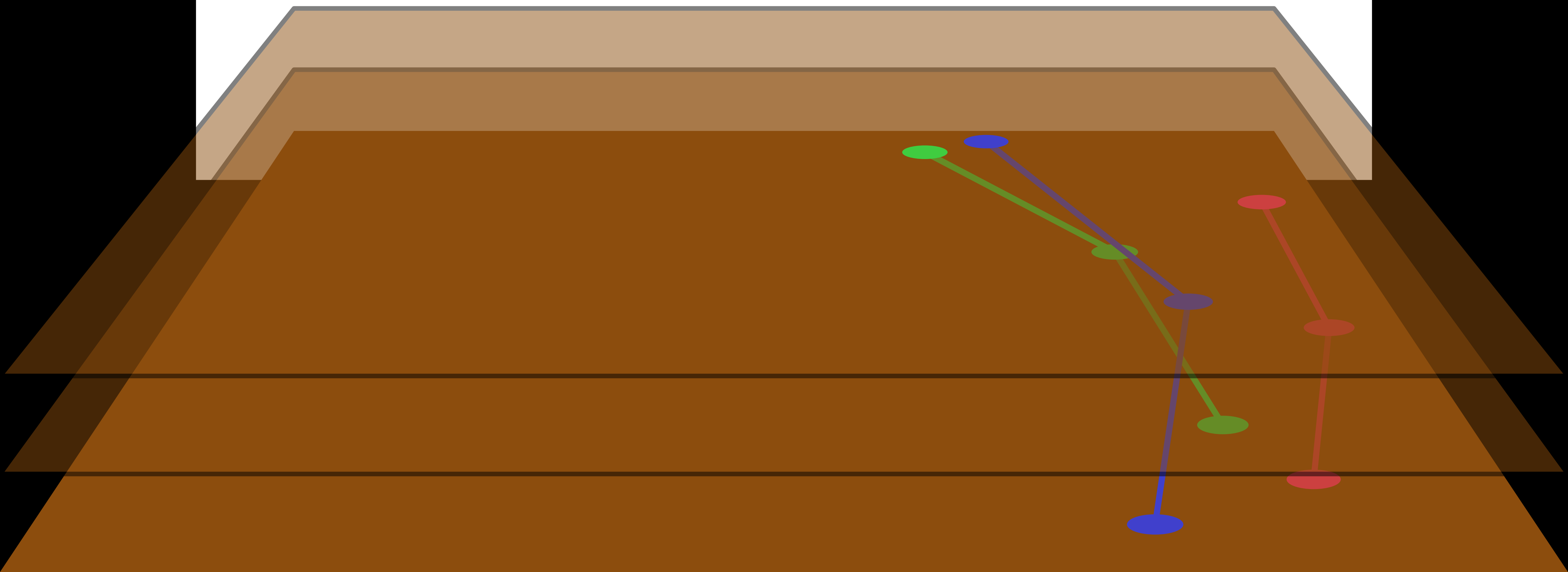
DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?



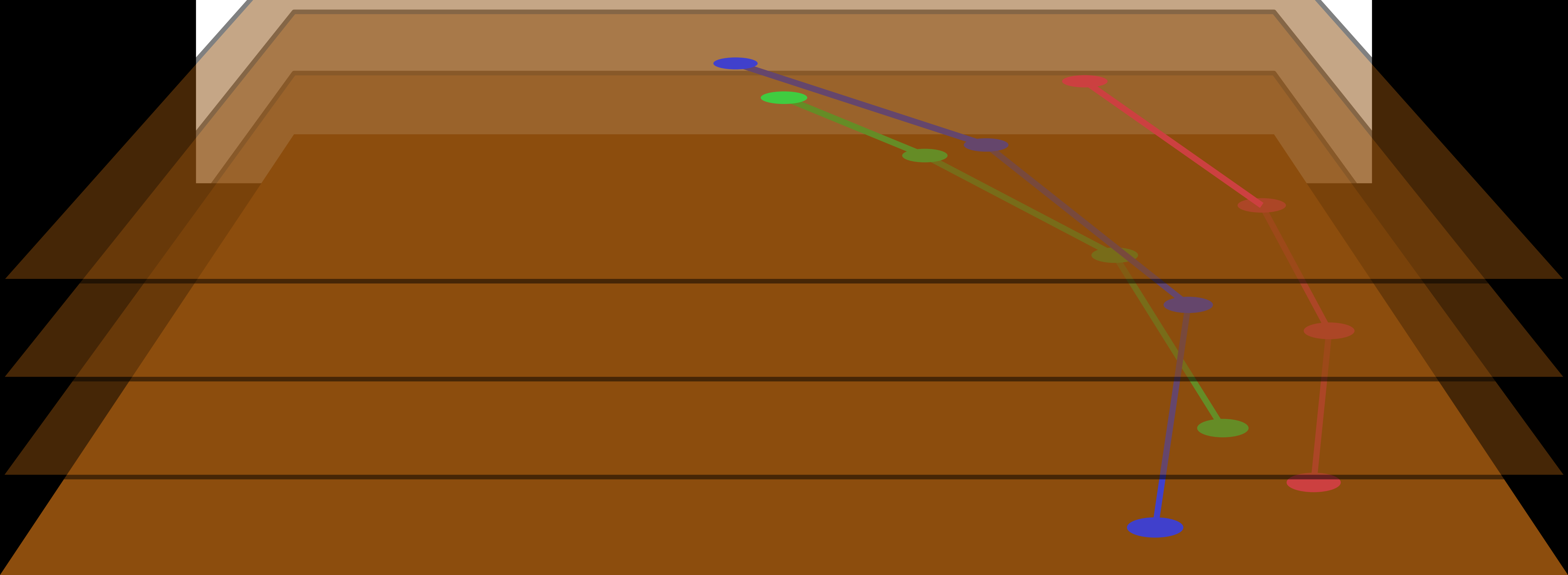
DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?



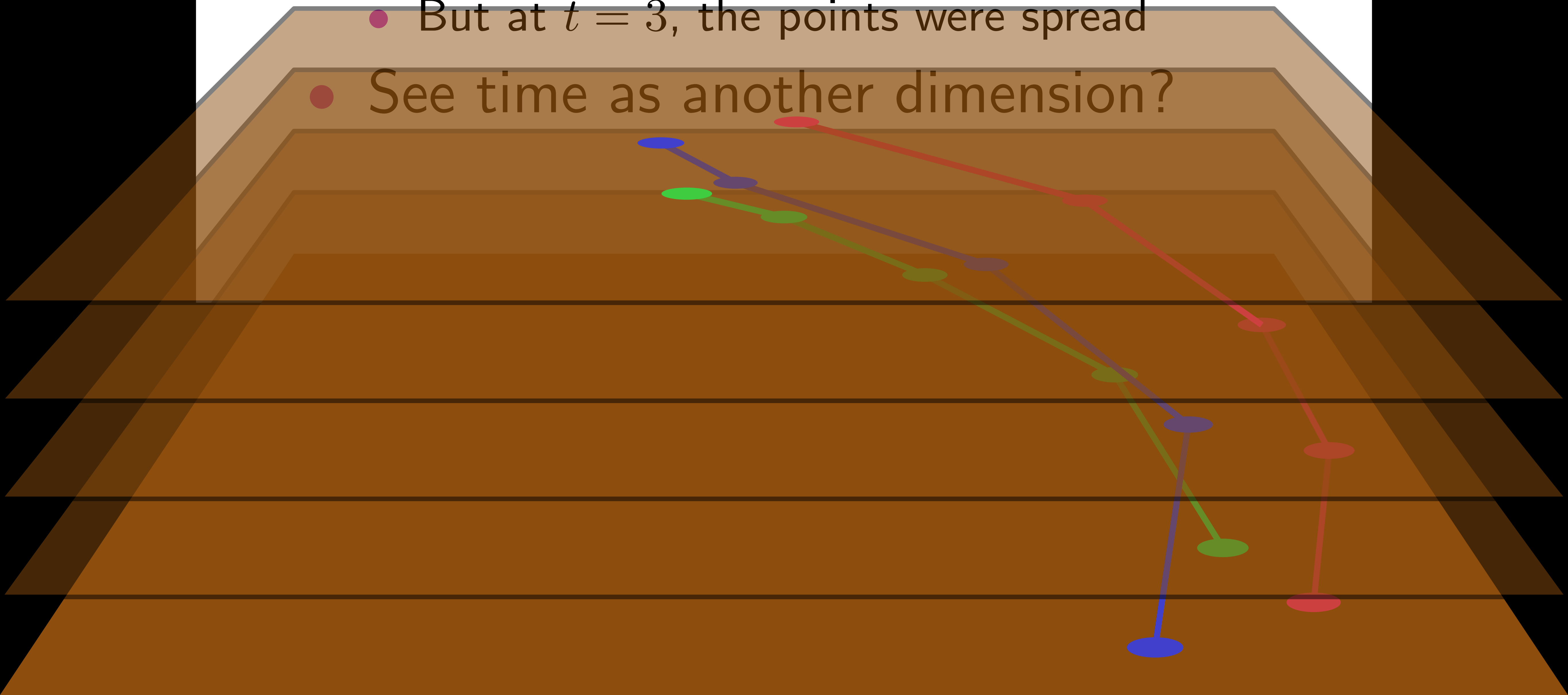
DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?



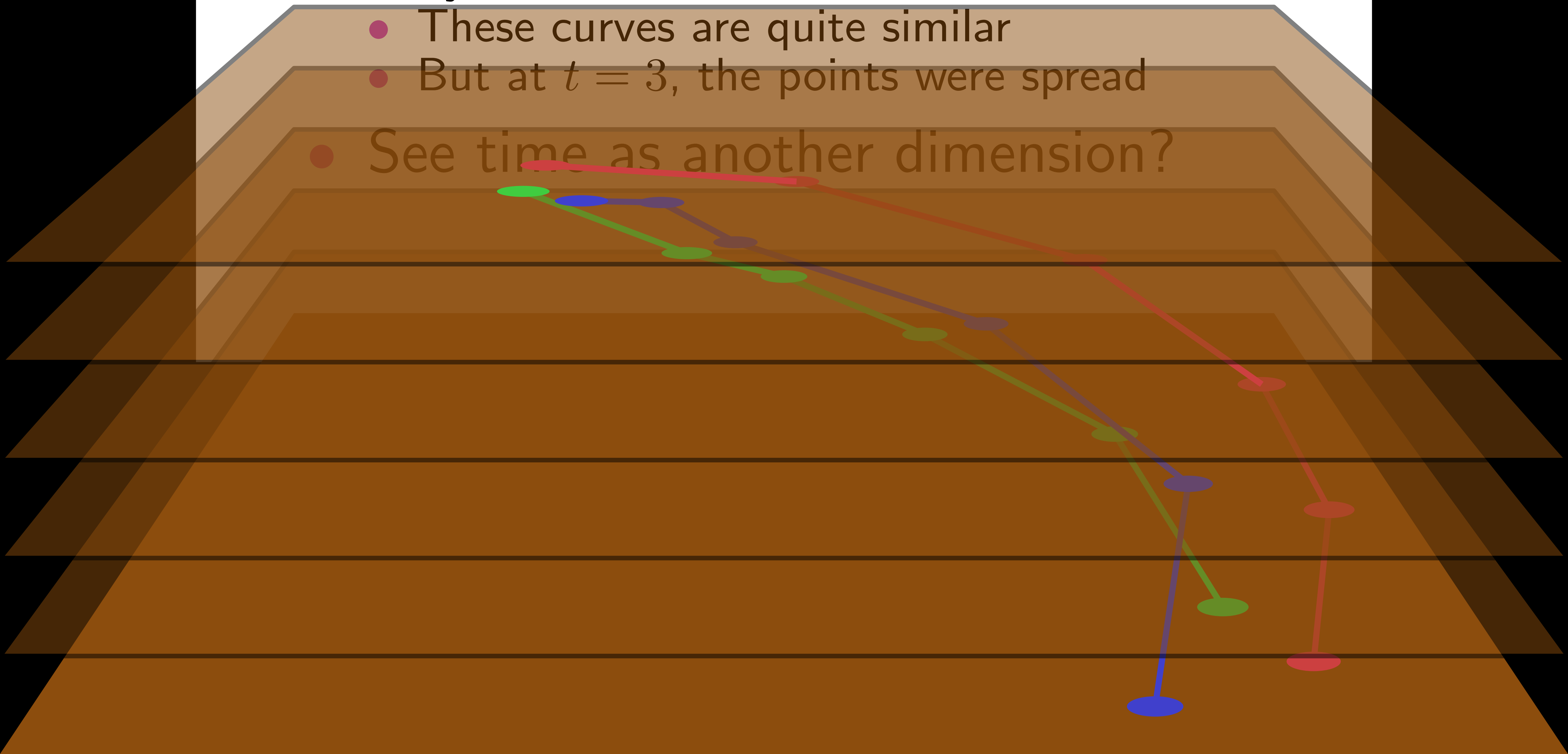
DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?



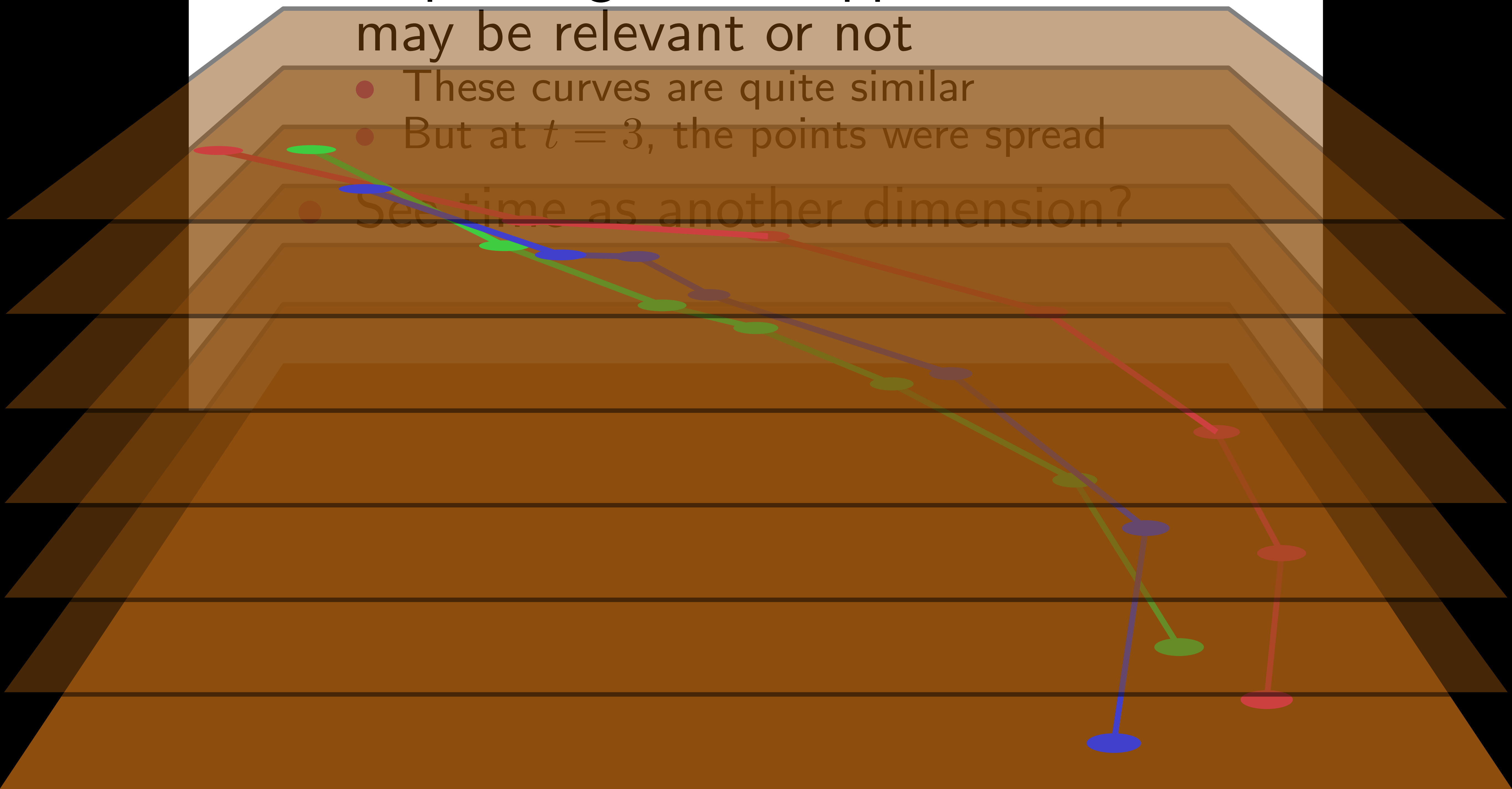
DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?



DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?



DO YOU HAVE TIME TO CARE?

- Trajectories have *time-stamped* locations
- Depending on the application, time may be relevant or not
  - These curves are quite similar
  - But at $t = 3$, the points were spread
- See time as another dimension?

WE DON'T CARE ABOUT TIME

WE DON'T CARE ABOUT TIME

- Trajectories are just curves

WE DON'T CARE ABOUT TIME

- Trajectories are just curves

WE DON'T CARE ABOUT TIME

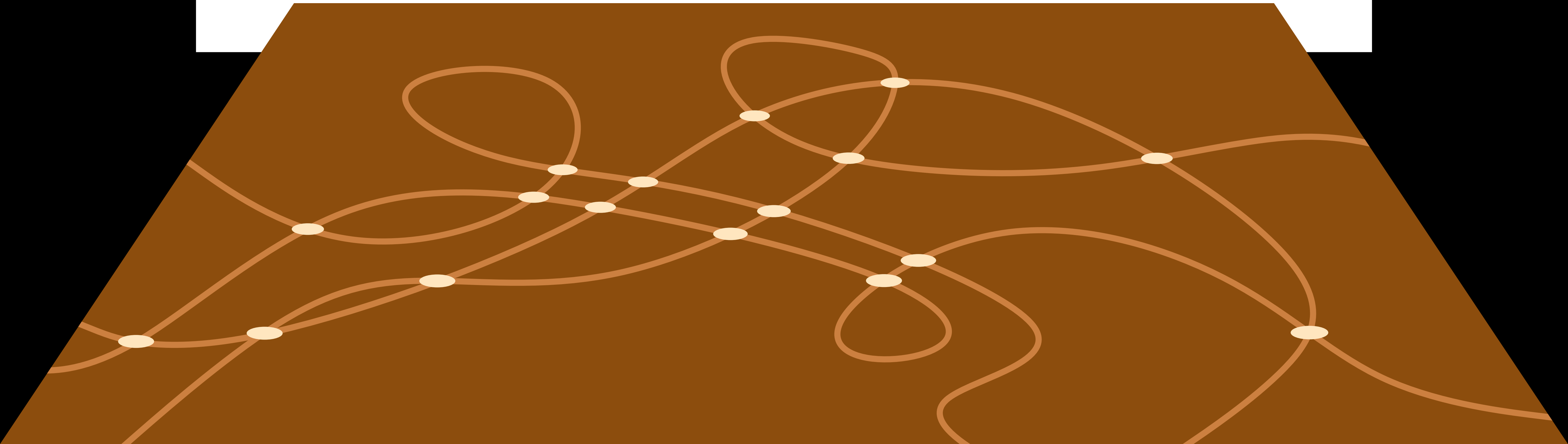
- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps

WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph

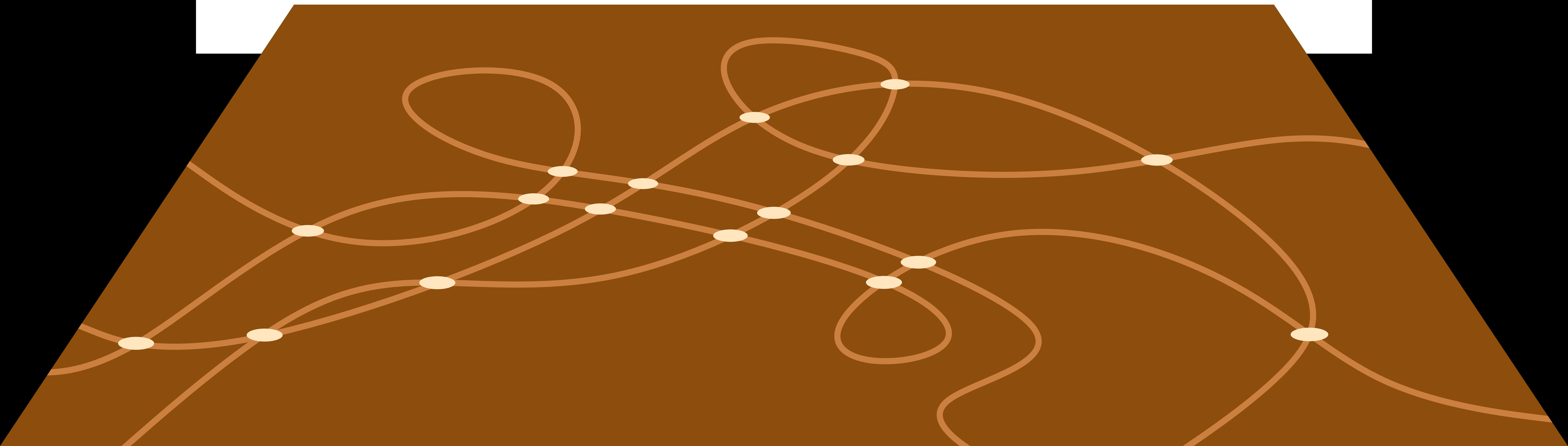
WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph



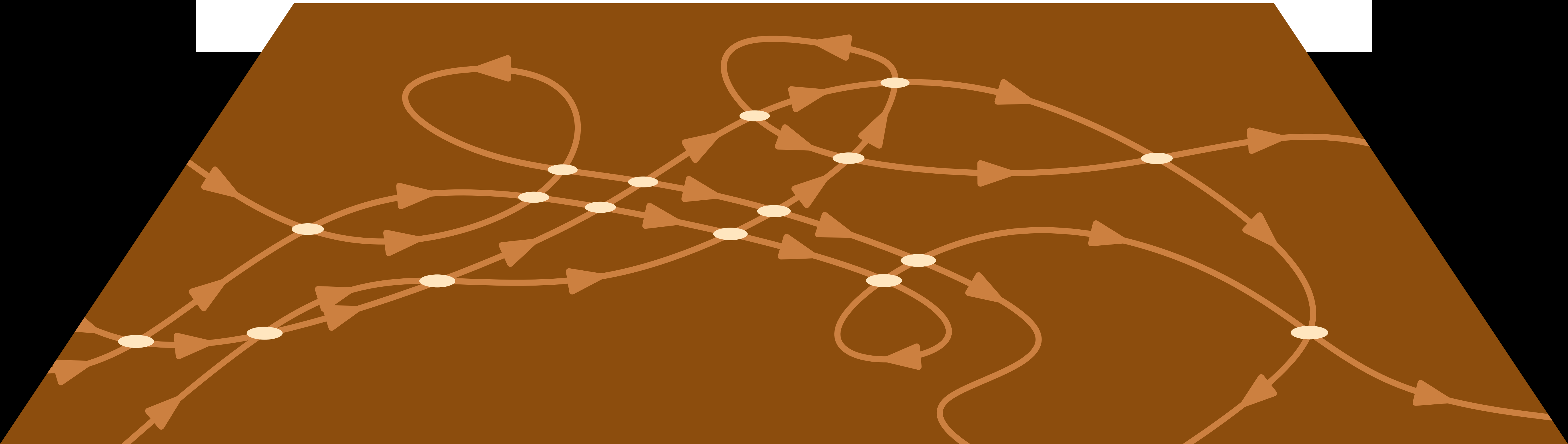
WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph
  - Edges are directed



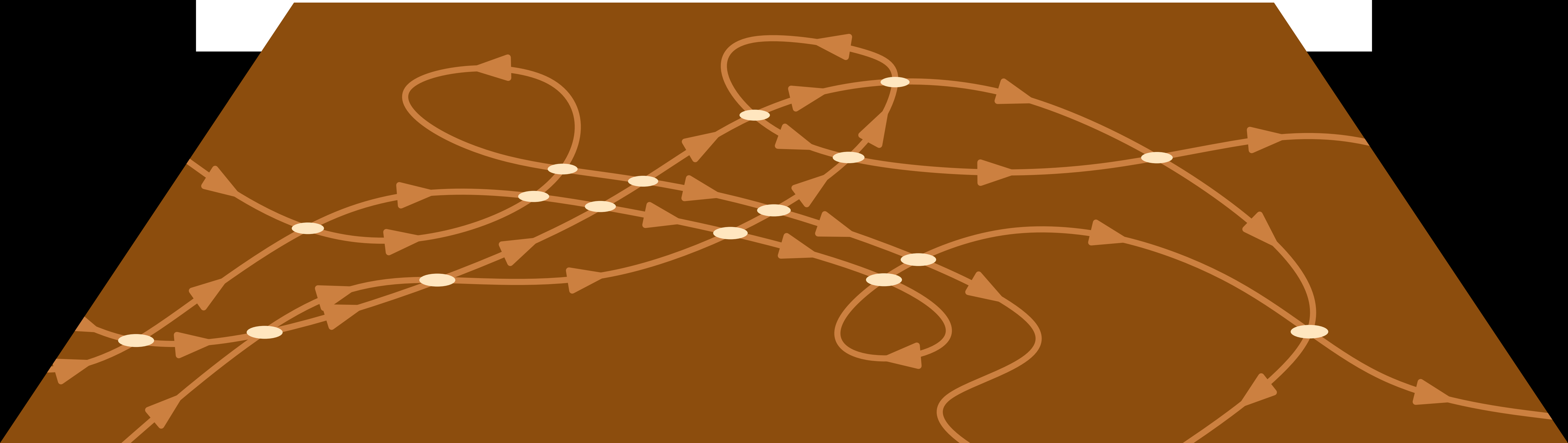
WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph
  - Edges are directed



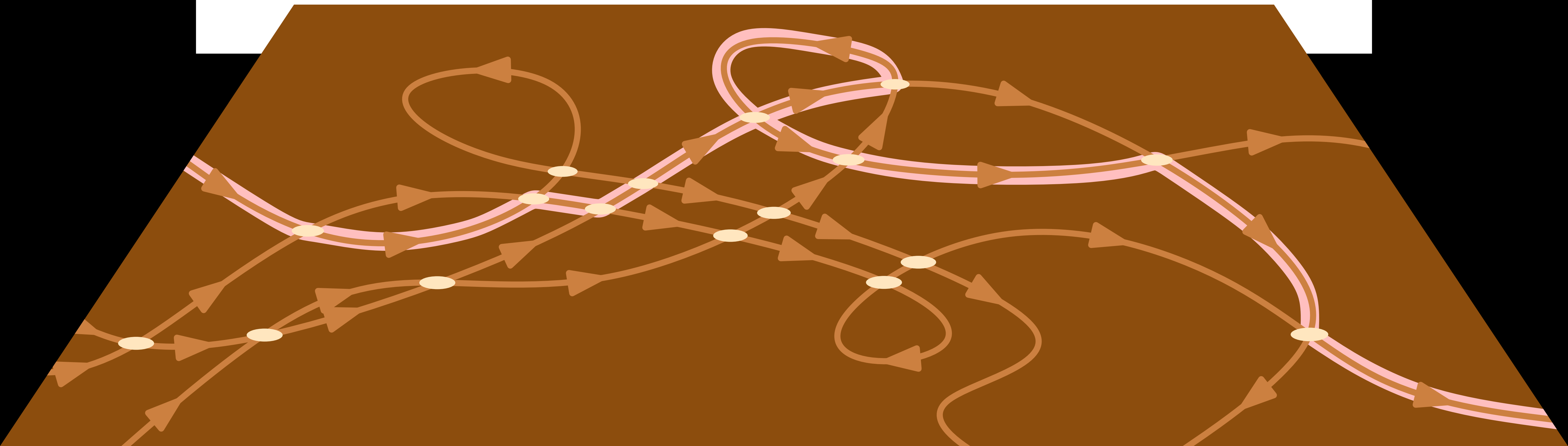
WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph
  - Edges are directed
- Output is a path in this graph



WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph
  - Edges are directed
- Output is a path in this graph

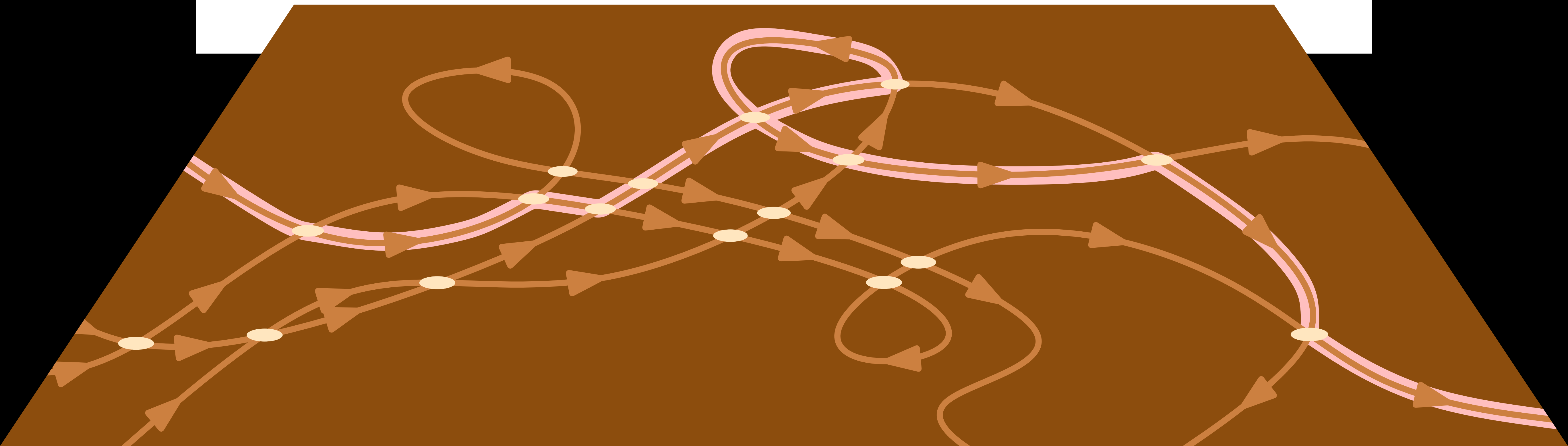


WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph
  - Edges are directed
- Output is a path in this graph
- What makes a good representative?

WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph
  - Edges are directed
- Output is a path in this graph
- What makes a good representative?
  - It should have sensible topology



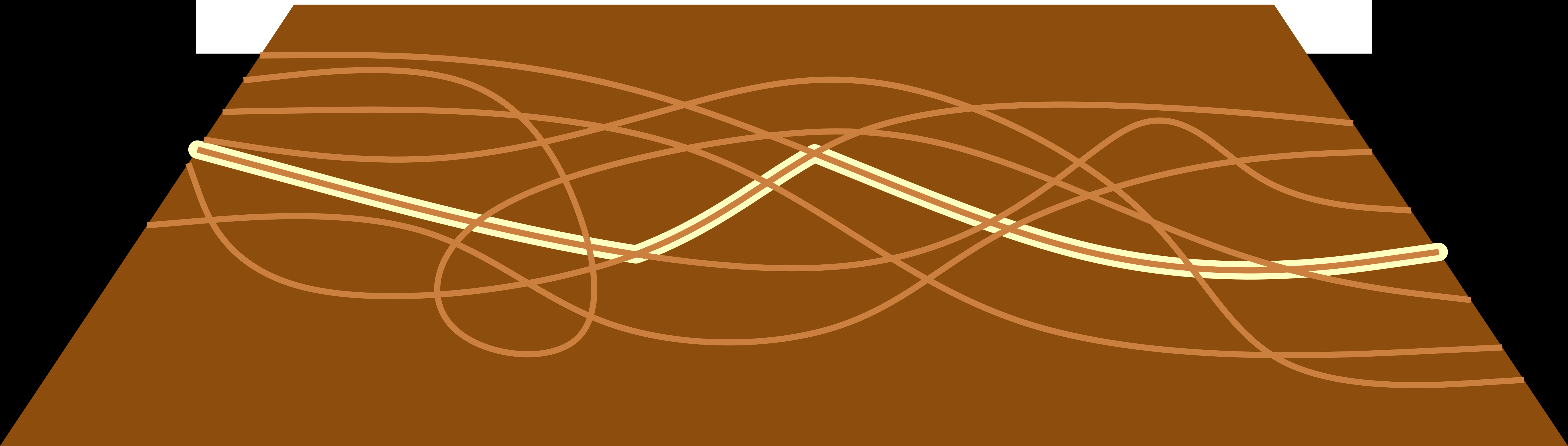
WE DON'T CARE ABOUT TIME

- Trajectories are just curves
  - In $\mathbb{R}^2$, we do not need jumps
  - Arrangement of curves forms a graph
  - Edges are directed
- Output is a path in this graph
- What makes a good representative?
  - It should have sensible topology
  - But we cannot ignore the geometry

PROBLEM STATEMENT

PROBLEM STATEMENT

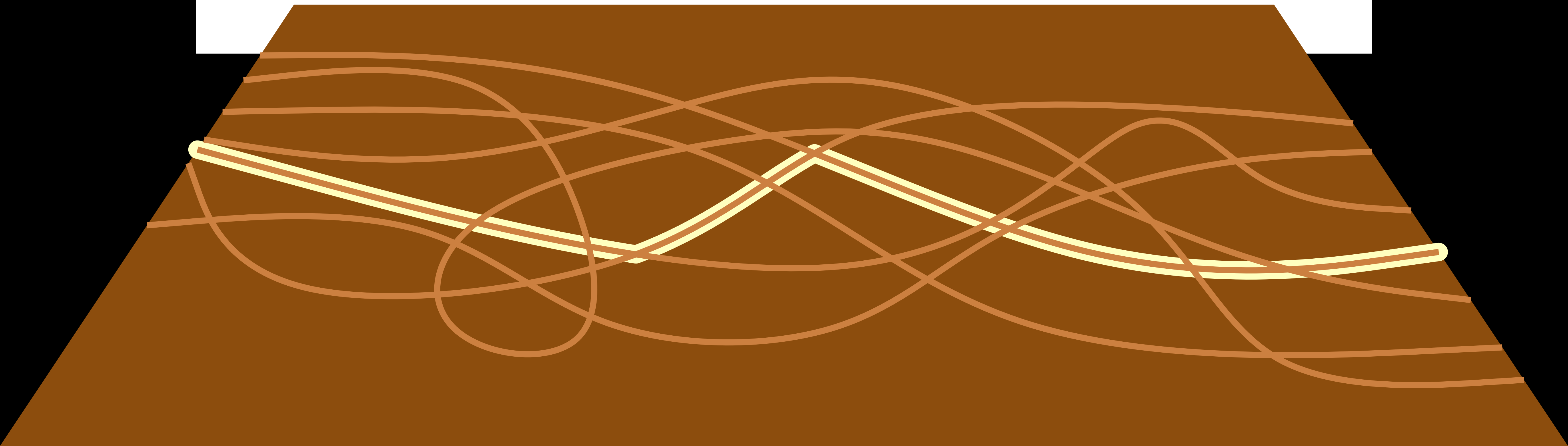
Given a set of ‘similar’ trajectories,
Find a representative trajectory:



PROBLEM STATEMENT

Given a set of ‘similar’ trajectories,
Find a representative trajectory:

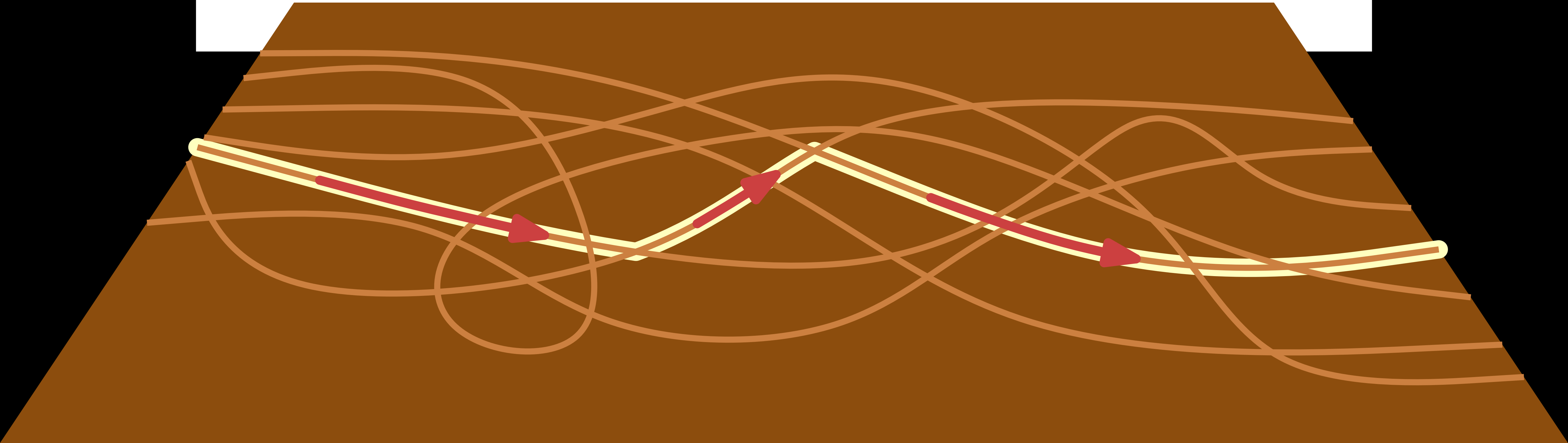
- consists of segments of input trajectories



PROBLEM STATEMENT

Given a set of ‘similar’ trajectories,
Find a representative trajectory:

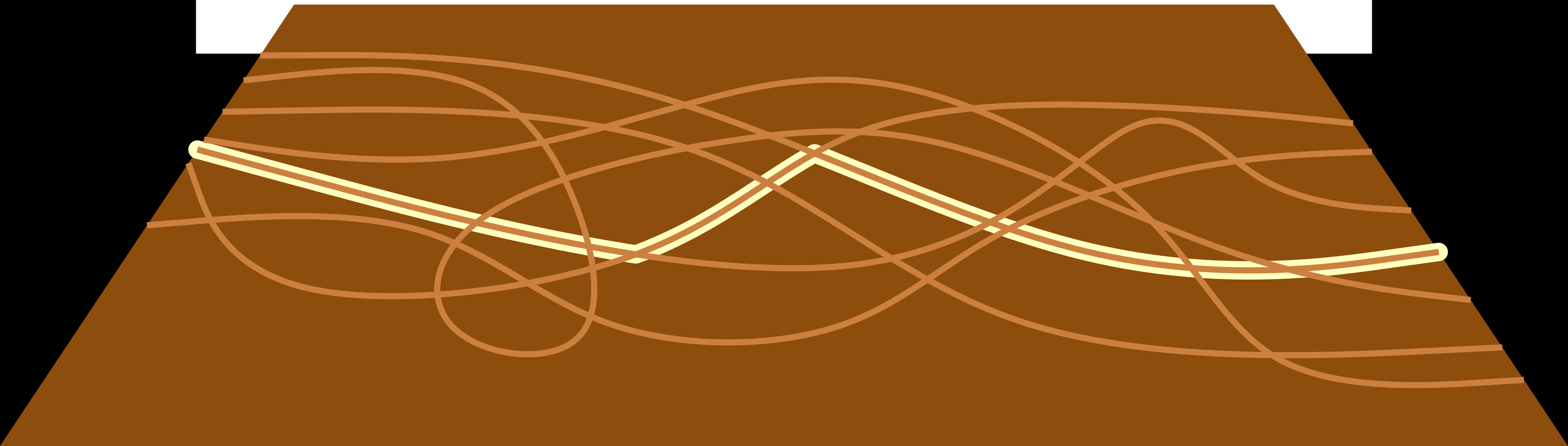
- consists of segments of input trajectories
- uses each segment in the correct direction



PROBLEM STATEMENT

Given a set of 'similar' trajectories,
Find a representative trajectory:

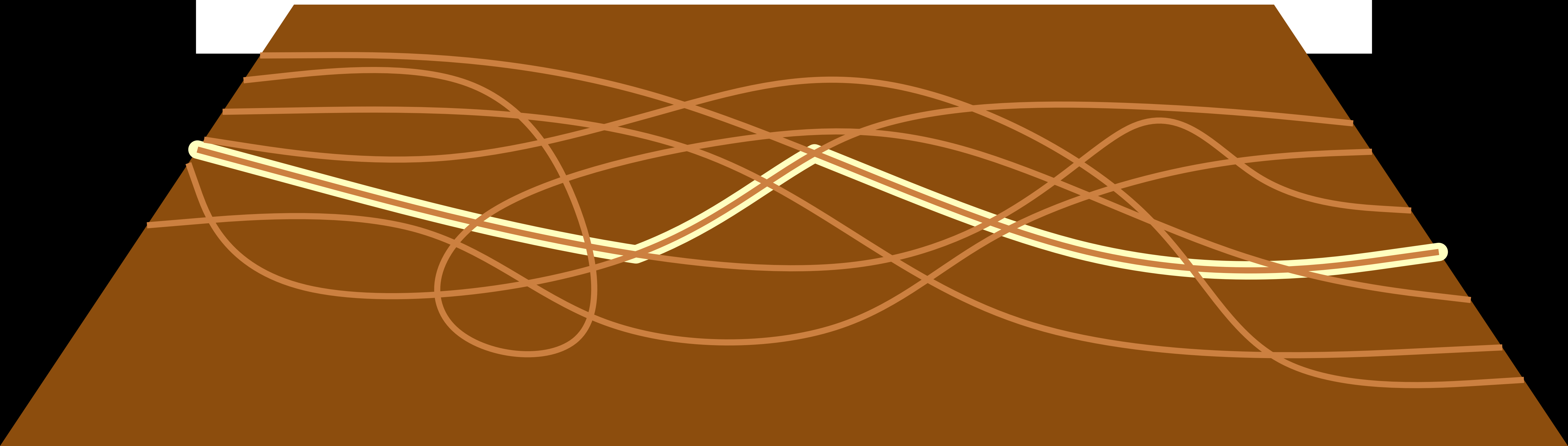
- consists of segments of input trajectories
- uses each segment in the correct direction
- all segments appear in the correct order



PROBLEM STATEMENT

Given a set of ‘similar’ trajectories,
Find a representative trajectory:

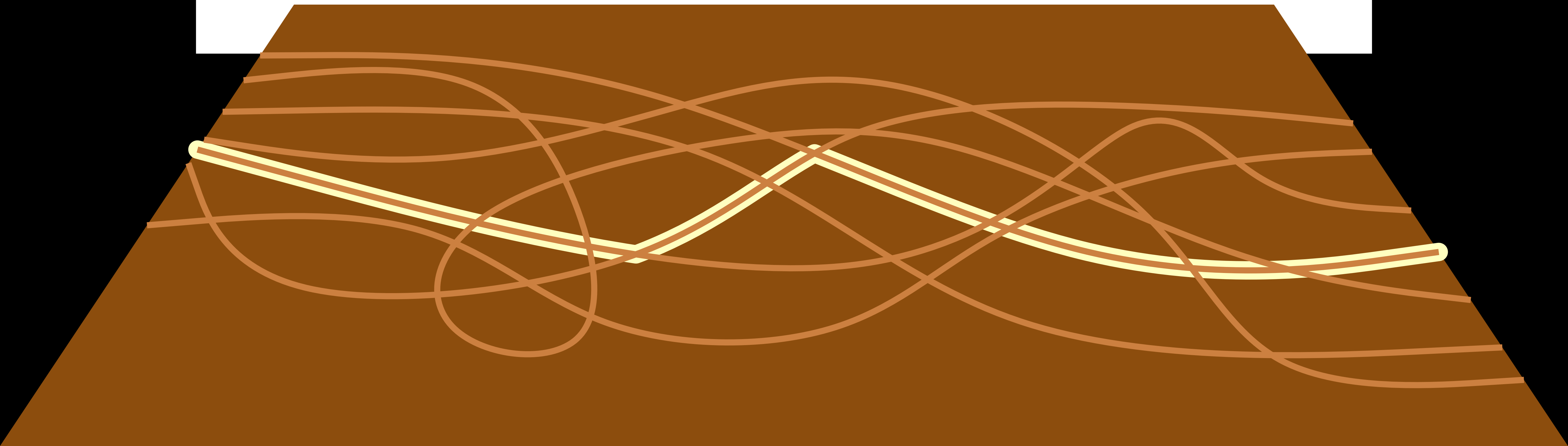
- consists of segments of input trajectories
- uses each segment in the correct direction
- all segments appear in the correct order
- represents trajectories ‘well’



PROBLEM STATEMENT

Given a set of 'similar' trajectories,
Find a representative trajectory:

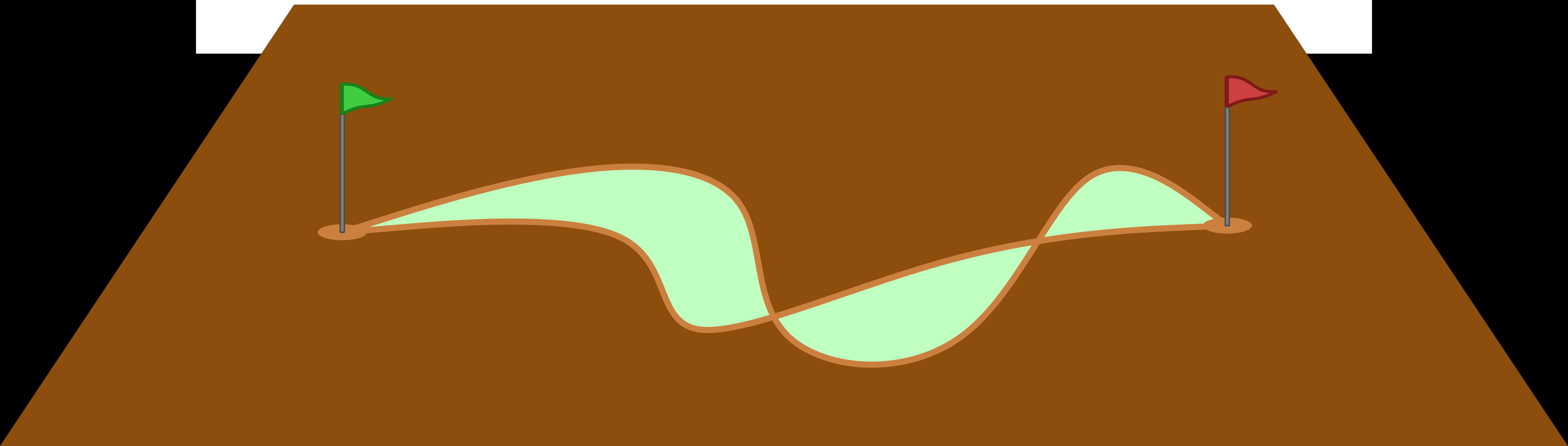
- consists of segments of input trajectories
- uses each segment in the correct direction
- all segments appear in the correct order
- minimizes homotopy area



HOMOTOPY AREA*

$$\int_{p \in \mathbb{R}^2} |\omega(p, \gamma)| \; \mathrm{d}s(p)$$

* As defined in [Chambers, Wang '13]

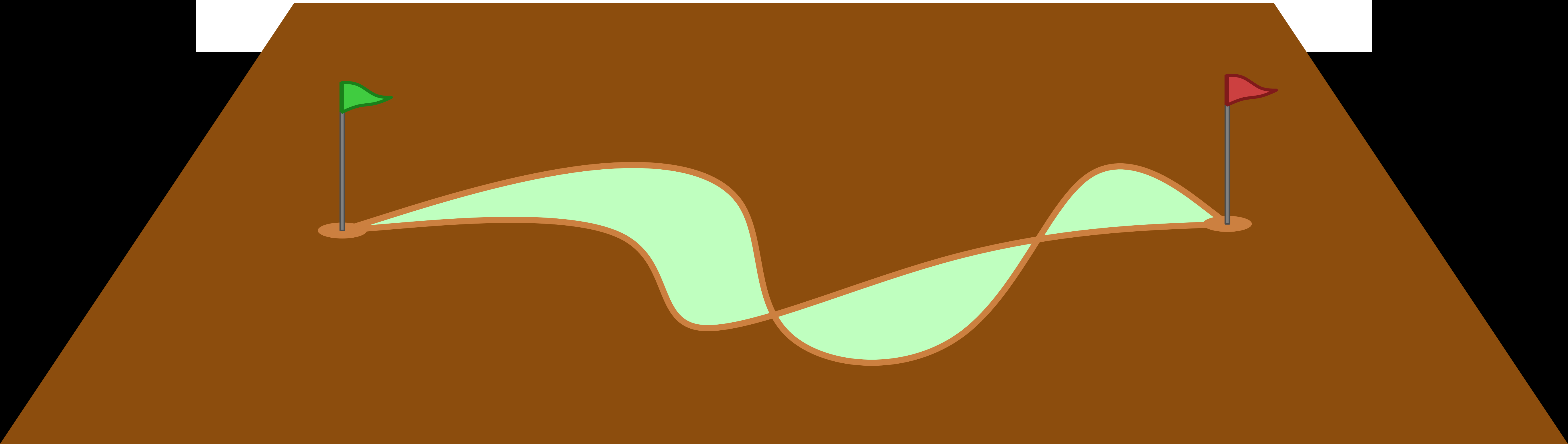


HOMOTOPY AREA*

$$\int_{p\in\mathbb{R}^2} |\omega(p,\gamma)| \; \mathrm{d}s(p)$$

Trajectories:

* As defined in [Chambers, Wang '13]



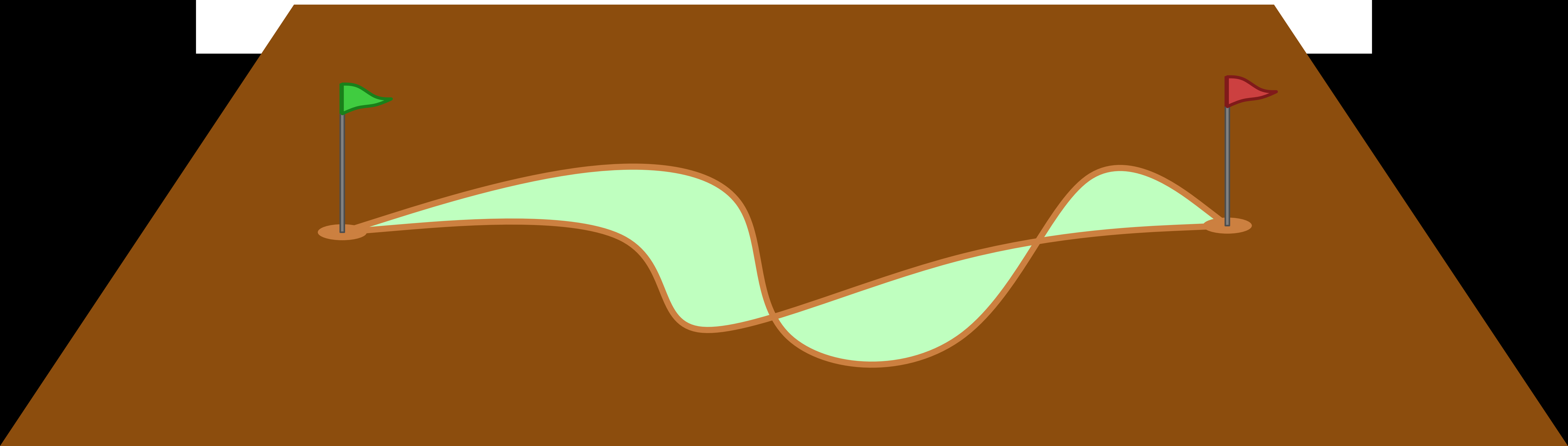
HOMOTOPY AREA*

$$\int_{p \in \mathbb{R}^2} |\omega(p, \gamma)| \, \mathrm{d}s(p)$$

Trajectories:

- start in s and end in t

* As defined in [Chambers, Wang '13]



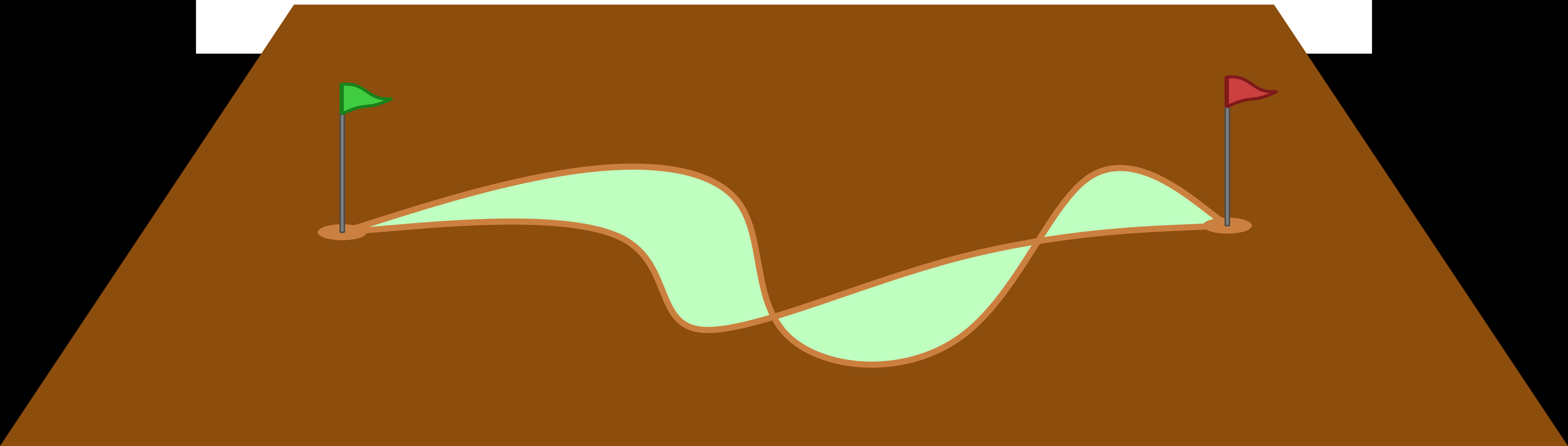
HOMOTOPY AREA*

$$\int_{p \in \mathbb{R}^2} |\omega(p, \gamma)| \, \mathrm{d}s(p)$$

Trajectories:

- start in s and end in t
- are simple

* As defined in [Chambers, Wang '13]



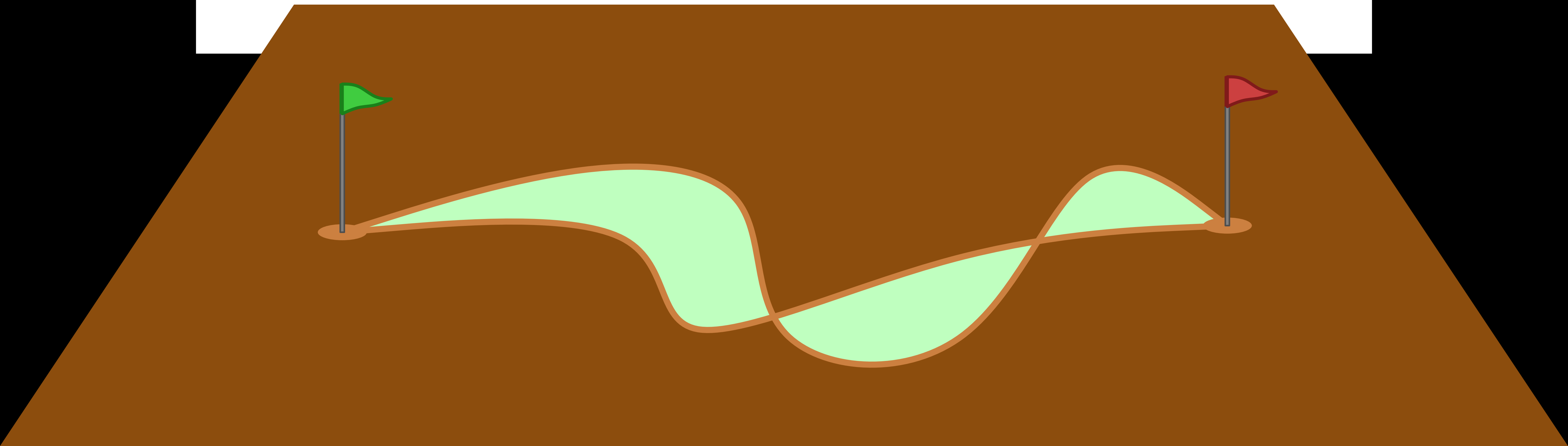
HOMOTOPY AREA*

$$\int_{p \in \mathbb{R}^2} |\omega(p, \gamma)| \; \mathrm{d}s(p)$$

Trajectories:

- start in s and end in t
- are simple
- are homotopic

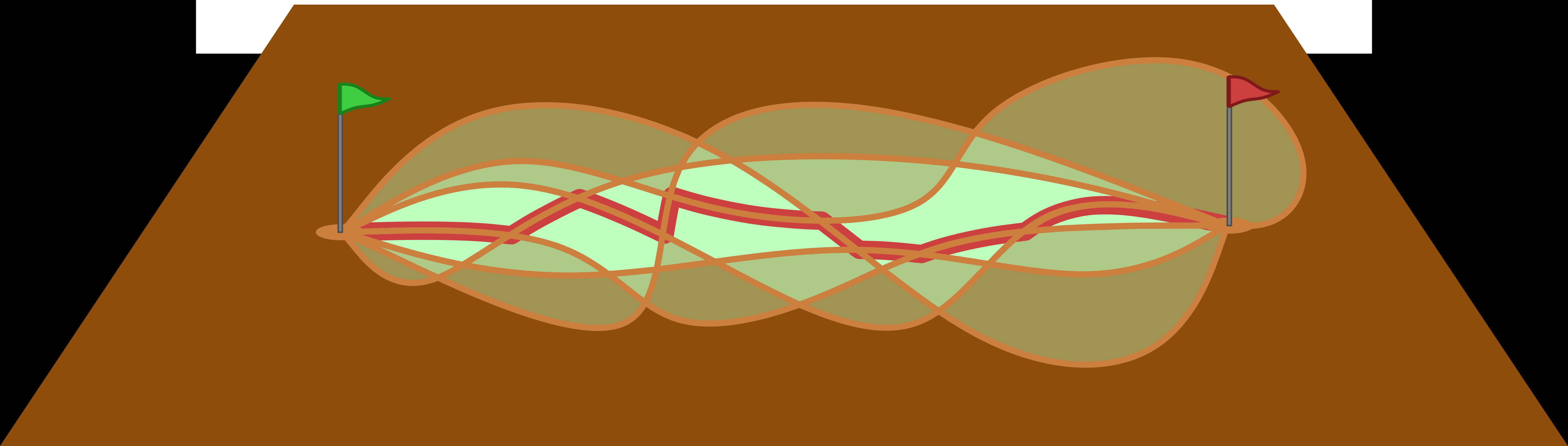
* As defined in [Chambers, Wang '13]



PROBLEM STATEMENT

Given a set of simple 'similar' trajectories, that all start in the same point and all end in the same point,
Find a representative trajectory:

- consists of segments of input trajectories
- uses each segment in the correct direction
- all segments appear in the correct order
- is simple
- minimizes max/avg homotopy area

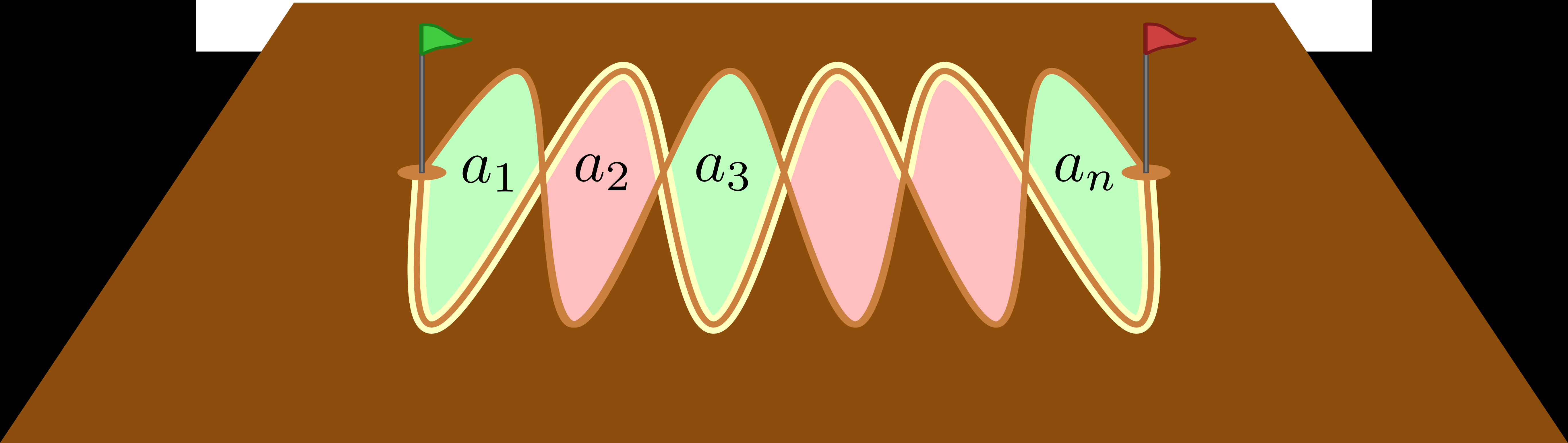


MIN MAX IS NP-HARD

Reduction from PARTITION:

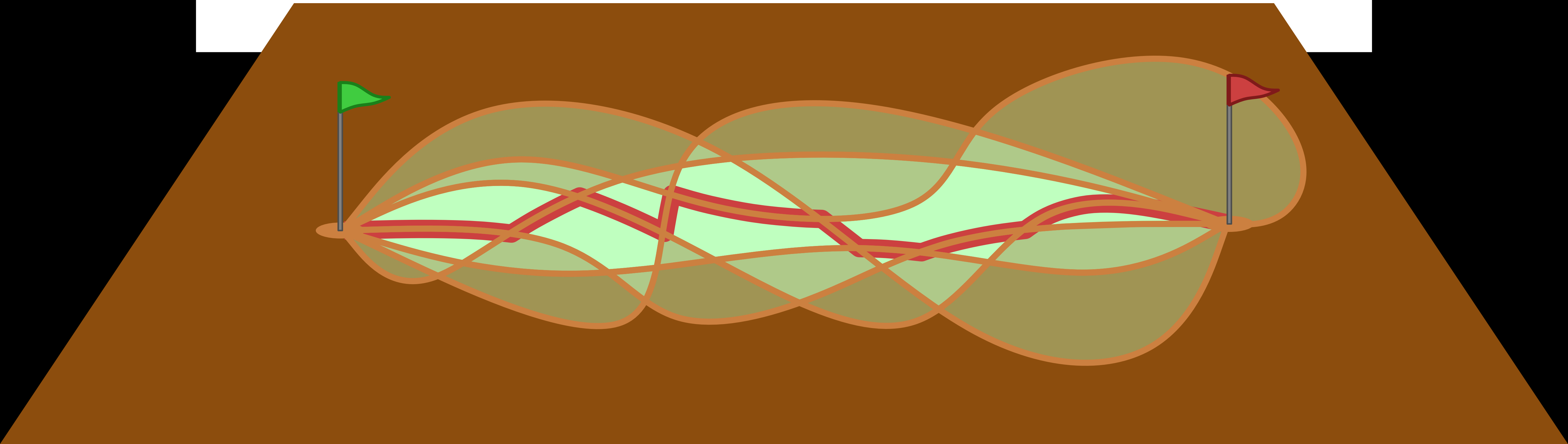
Partition a set of integers $S = \{a_1, a_2, \ldots, a_n\}$ into two subsets S_1 and S_2 with equal total sums:

$$\sum_{a \in S_1} a = \sum_{a \in S_2} a$$



MIN AVG IS EQUIVALENT TO MEDIAN*

* As defined in [Buchin et al. '13]

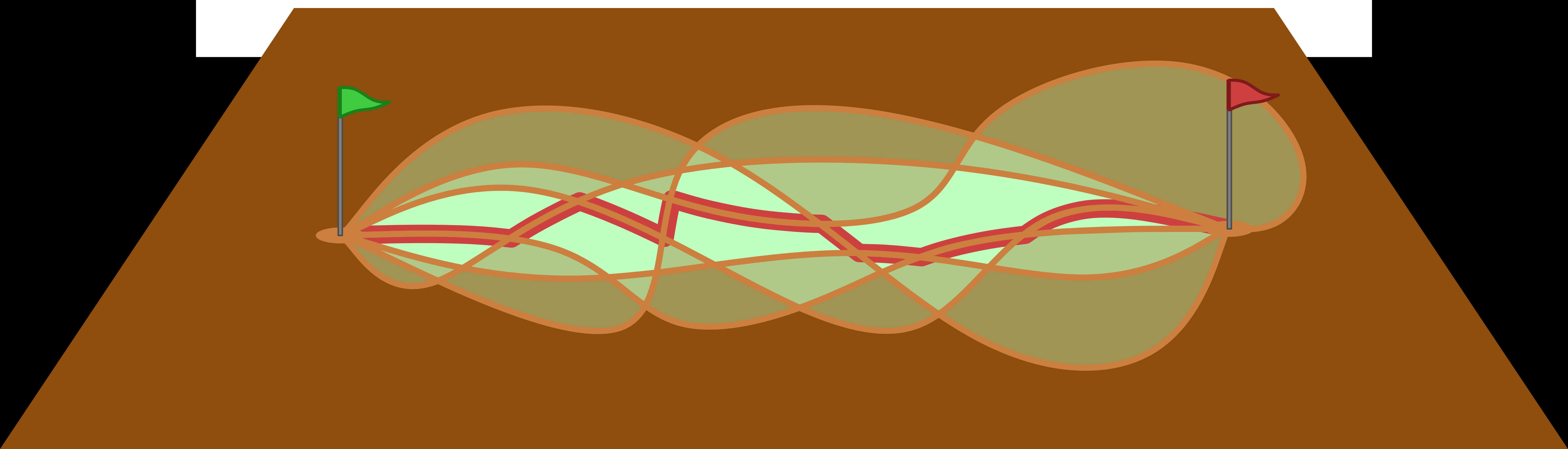


under certain conditions

MIN AVG IS EQUIVALENT TO MEDIAN*

- x-monotone case

* As defined in [Buchin et al. '13]

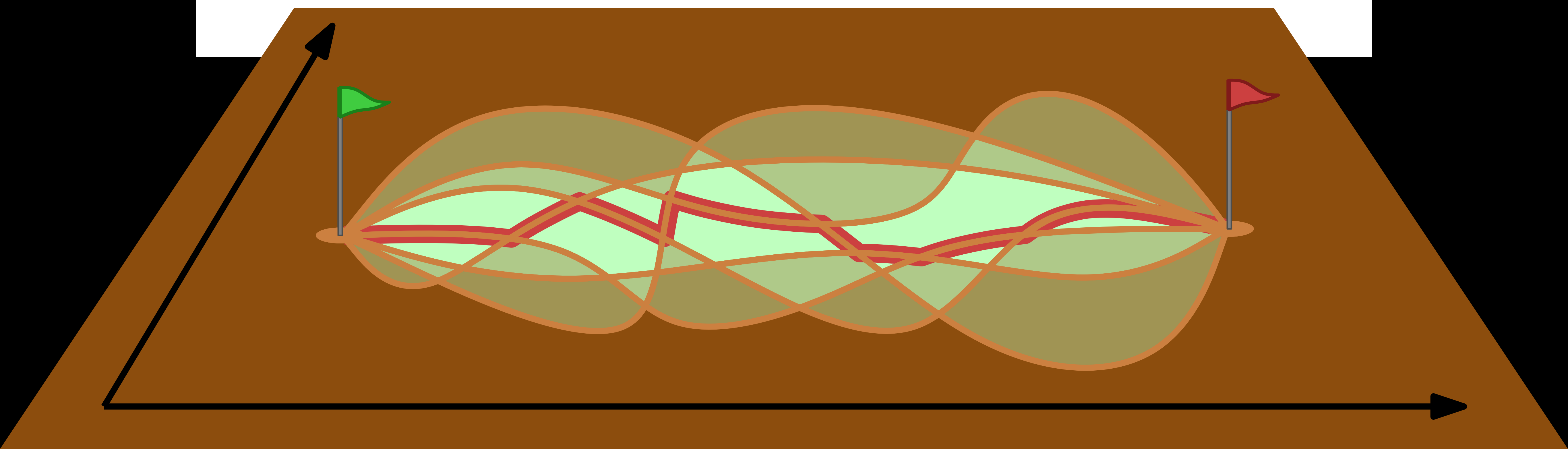


under certain conditions

MIN AVG IS EQUIVALENT TO MEDIAN*

- x-monotone case

* As defined in [Buchin et al. '13]

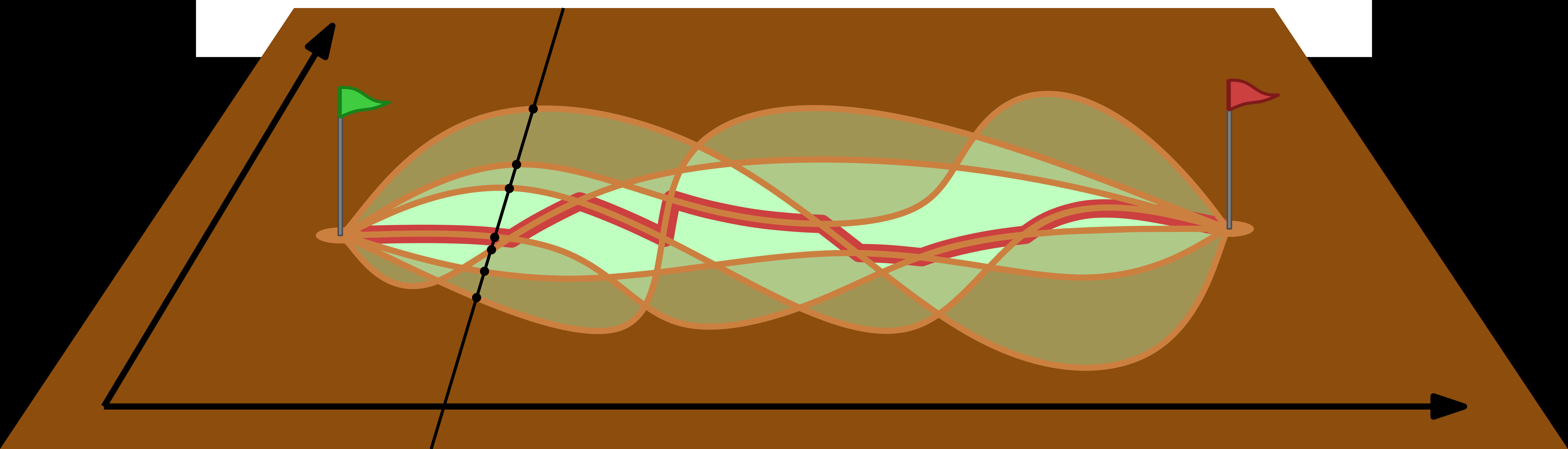


MIN AVG IS EQUIVALENT TO MEDIAN*

under certain conditions

- x-monotone case

* As defined in [Buchin et al. '13]

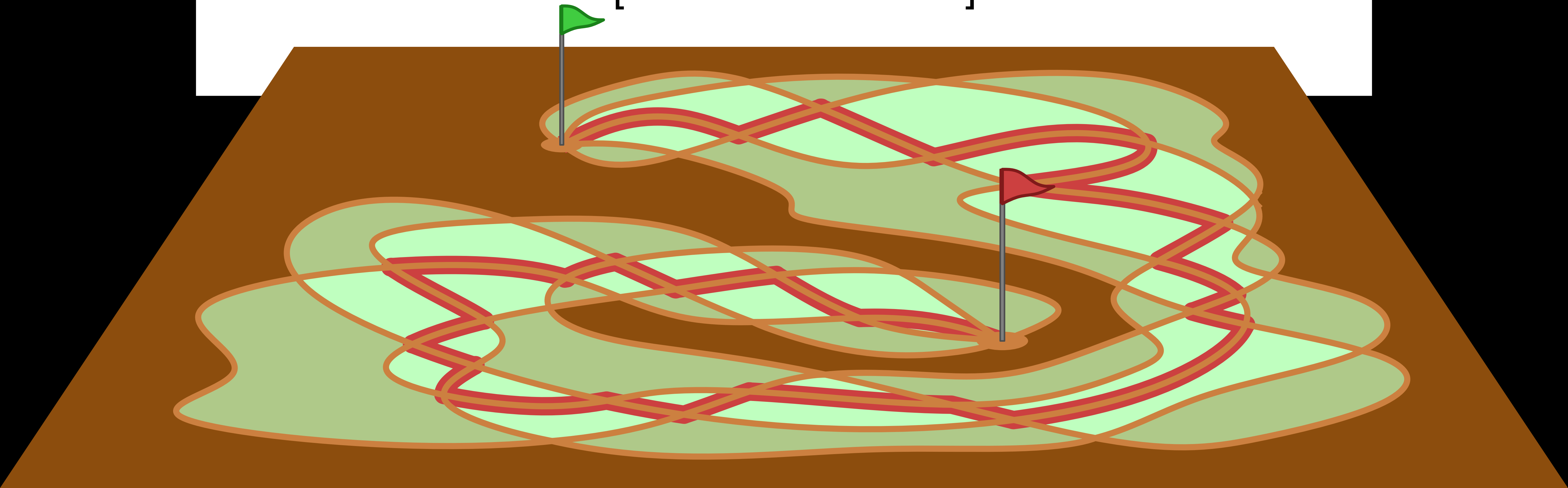


MIN AVG IS EQUIVALENT TO MEDIAN*

under certain conditions

- x-monotone case
- DAG

* As defined in [Buchin et al. '13]

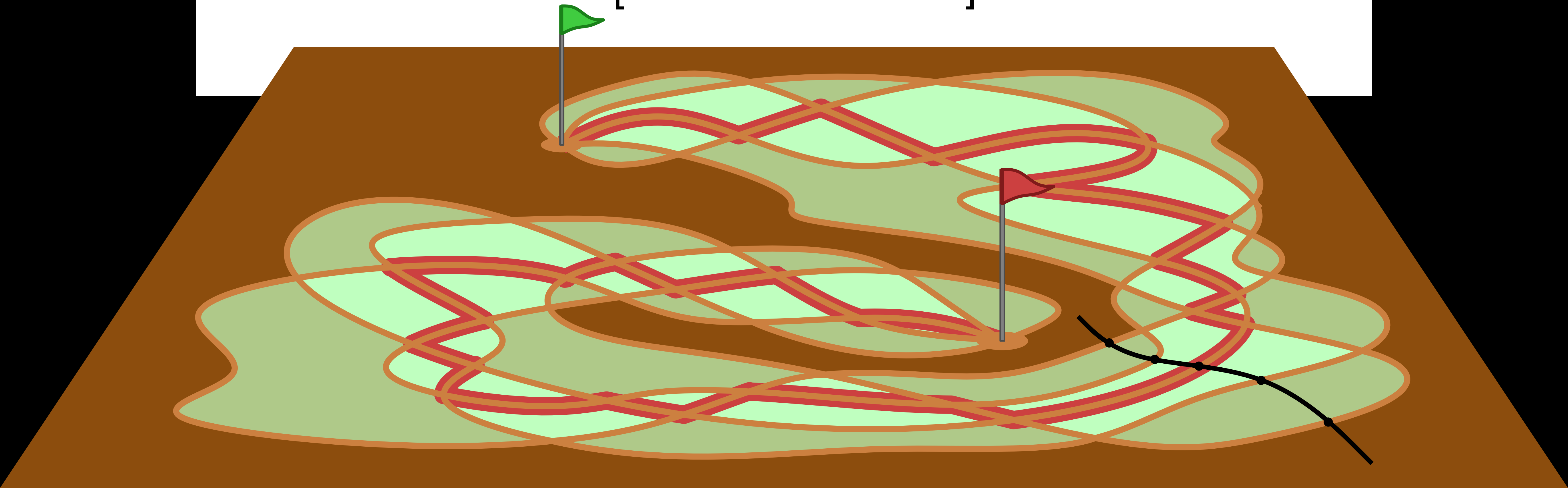


MIN AVG IS EQUIVALENT TO MEDIAN*

under certain conditions

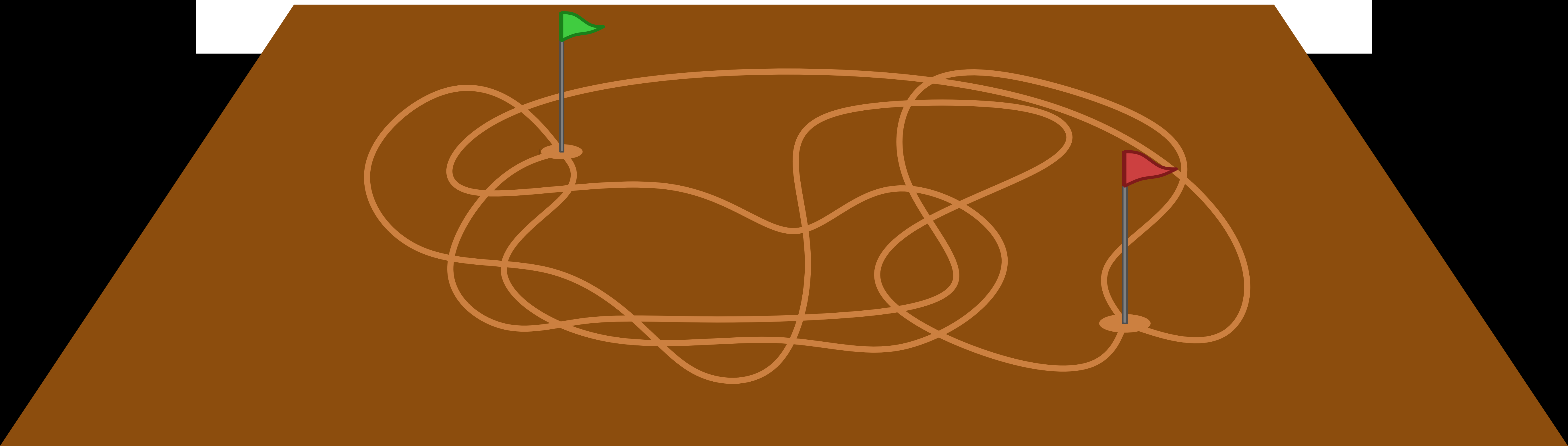
- x-monotone case
- DAG

* As defined in [Buchin et al. '13]



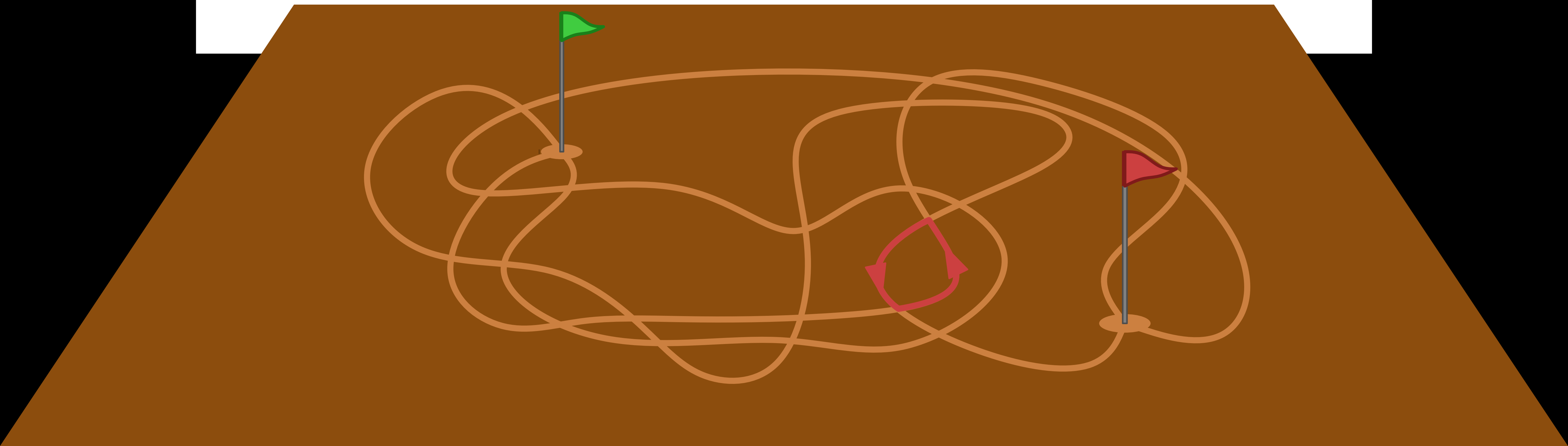
FUTURE WORK

Non-DAG:



FUTURE WORK

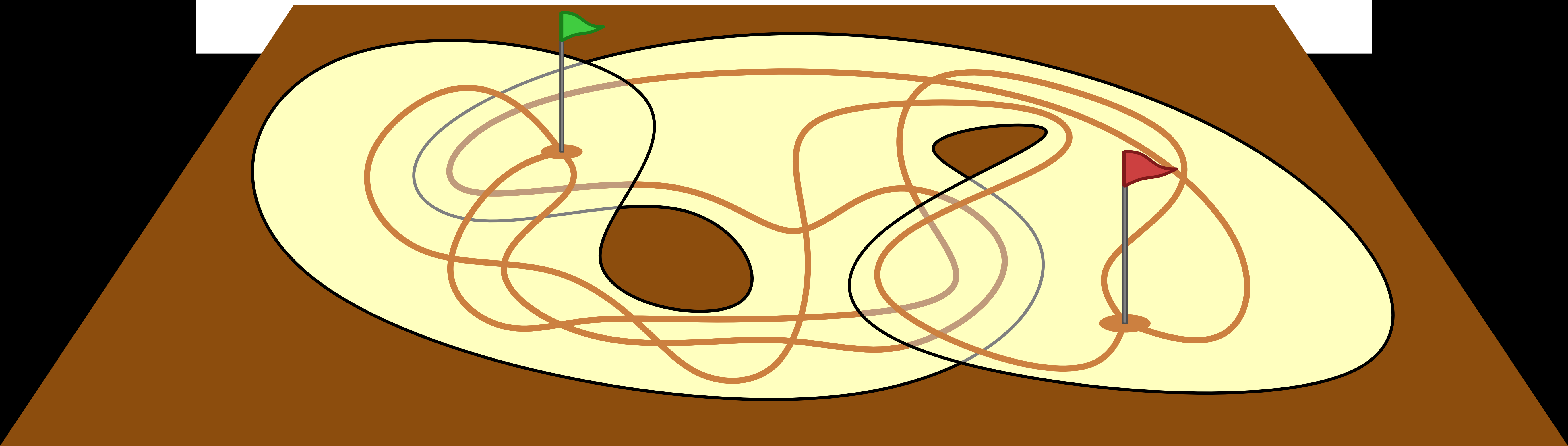
Non-DAG:



FUTURE WORK

Non-DAG:

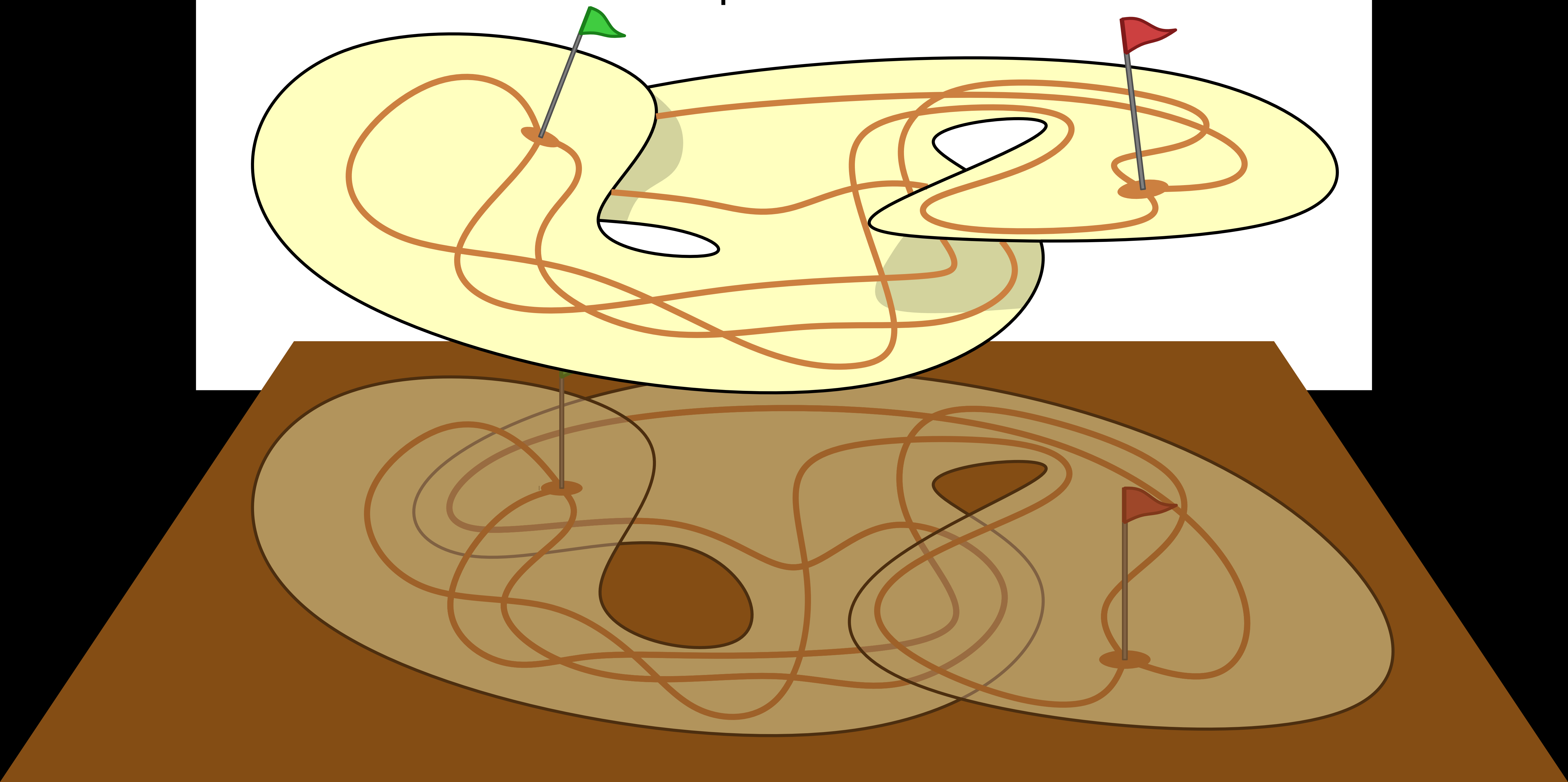
- define ‘corridor’



FUTURE WORK

Non-DAG:

- define ‘corridor’
- ‘lift’ to non-DAG space



CONCLUSIONS

CONCLUSIONS

- When time is not relevant

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - “simple median” optimises homotopy area

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - “simple median” optimises homotopy area
  - acyclic trajectory intersection graph

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - “simple median” optimises homotopy area
  - acyclic trajectory intersection graph
  - open problem: find appropriate space

INTERMISSION

WE DO CARE ABOUT TIME

WE DO CARE ABOUT TIME

- Simply increase the dimension?

WE DO CARE ABOUT TIME

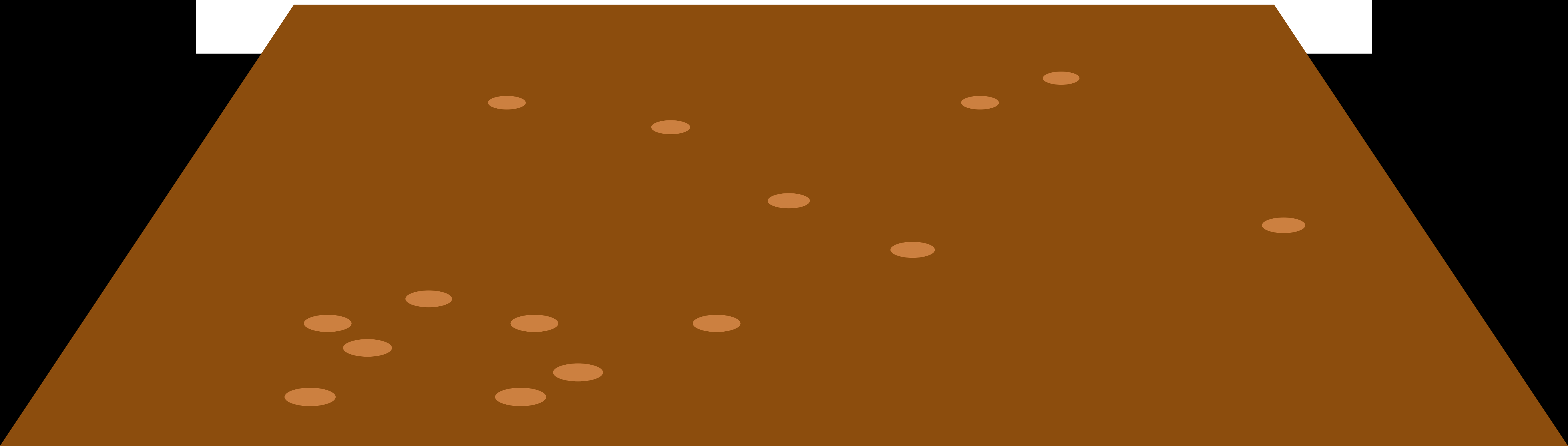
- Simply increase the dimension?
  - Distance in space is not comparable to distance in time

WE DO CARE ABOUT TIME

- Simply increase the dimension?
  - Distance in space is not comparable to distance in time
  - Representative trajectory should summarise P at every time

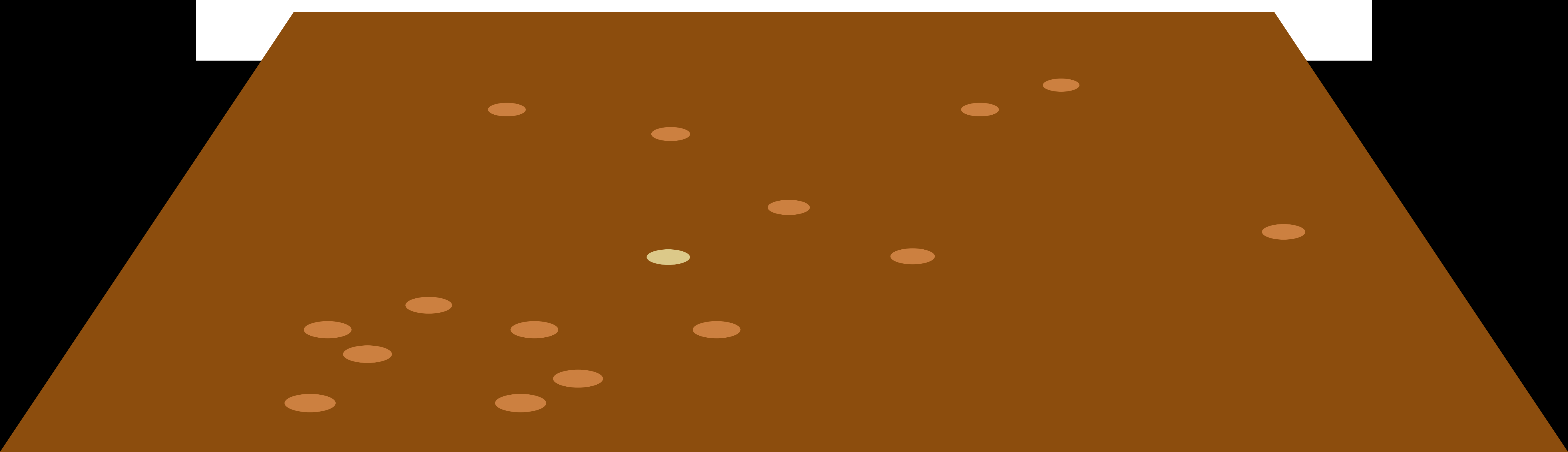
WE DO CARE ABOUT TIME

- Simply increase the dimension?
  - Distance in space is not comparable to distance in time
  - Representative trajectory should summarise P at every time
- Existing representative points



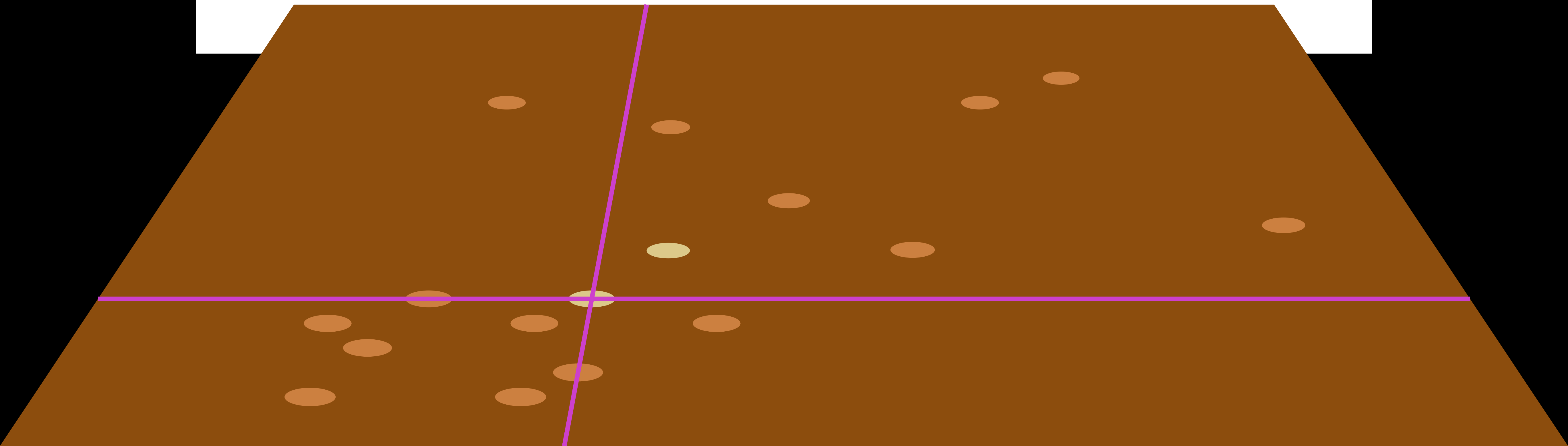
WE DO CARE ABOUT TIME

- Simply increase the dimension?
  - Distance in space is not comparable to distance in time
  - Representative trajectory should summarise P at every time
- Existing representative points
  - Average/mean of x- and y-coordinates



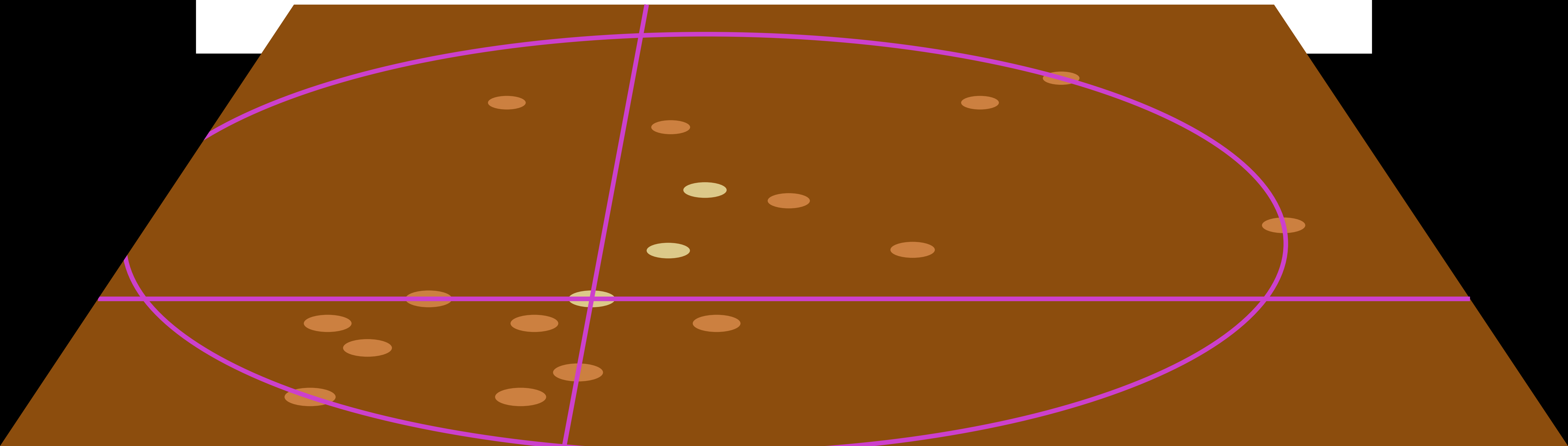
WE DO CARE ABOUT TIME

- Simply increase the dimension?
  - Distance in space is not comparable to distance in time
  - Representative trajectory should summarise P at every time
- Existing representative points
  - Average/mean of x- and y-coordinates
  - Median of x- and y-coordinates



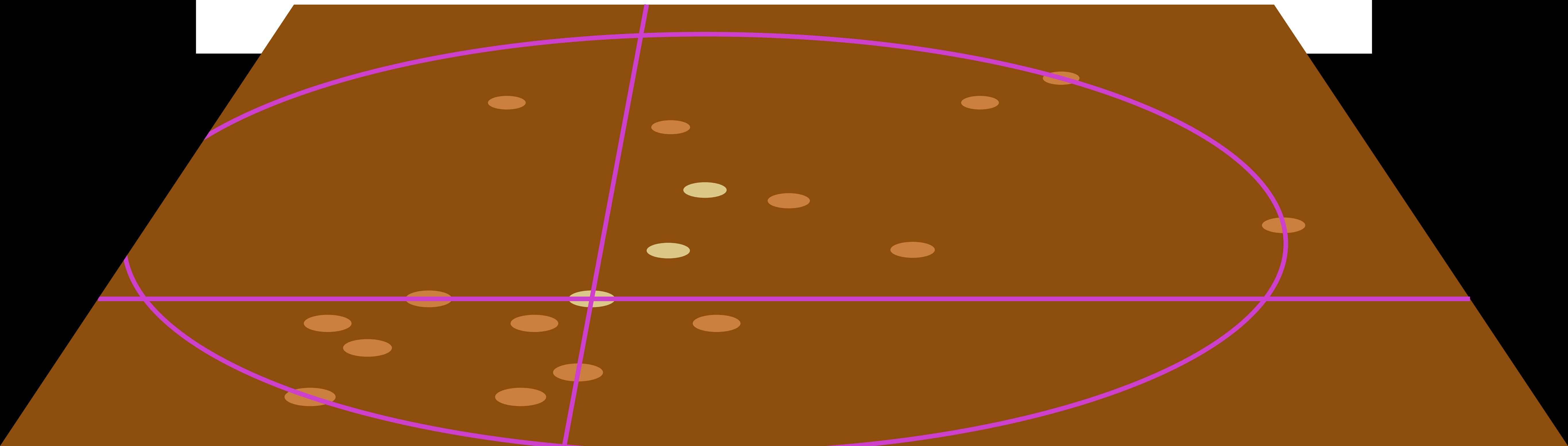
WE DO CARE ABOUT TIME

- Simply increase the dimension?
  - Distance in space is not comparable to distance in time
  - Representative trajectory should summarise P at every time
- Existing representative points
  - Average/mean of x- and y-coordinates
  - Median of x- and y-coordinates
  - Centre of smallest enclosing circle



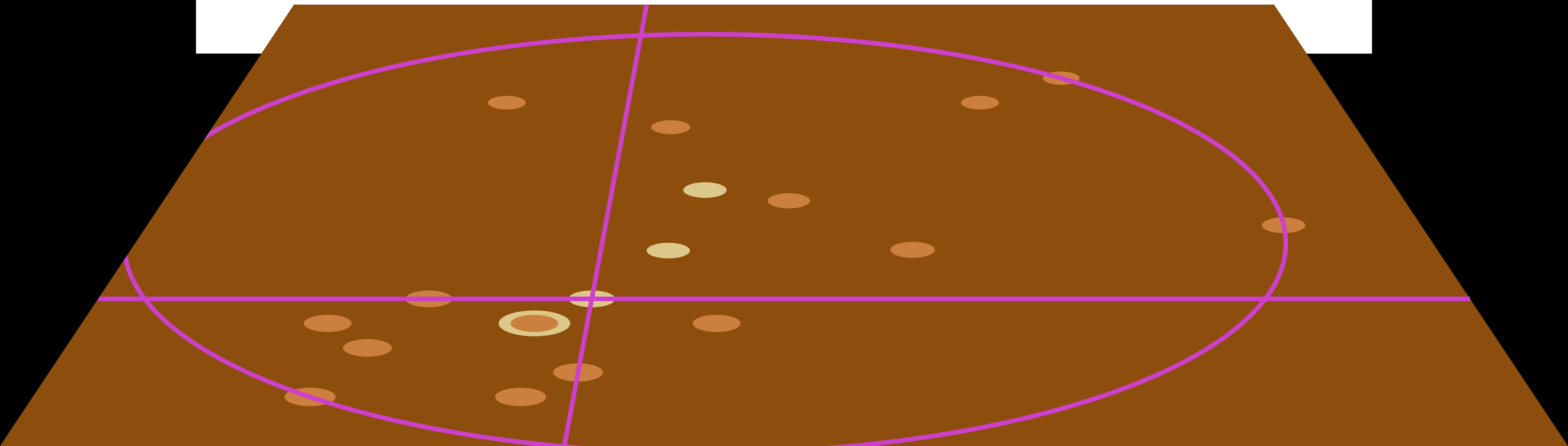
WE DO CARE ABOUT TIME

- Simply increase the dimension?
  - Distance in space is not comparable to distance in time
  - Representative trajectory should summarise P at every time
- Existing representative points
  - Average/mean of x- and y-coordinates
  - Median of x- and y-coordinates
  - Centre of smallest enclosing circle
- We want to be close all the time!

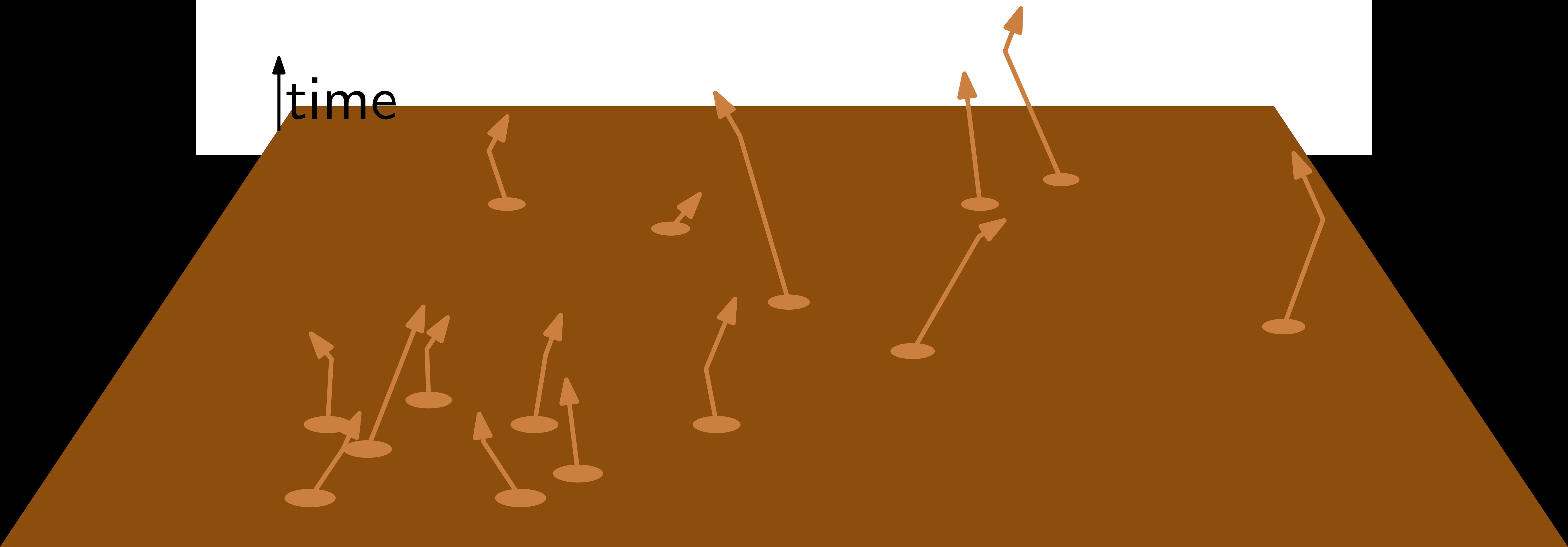


WE DO CARE ABOUT TIME

- Simply increase the dimension?
  - Distance in space is not comparable to distance in time
  - Representative trajectory should summarise P at every time
- Existing representative points
  - Average/mean of x- and y-coordinates
  - Median of x- and y-coordinates
  - Centre of smallest enclosing circle
- We want to be close all the time!

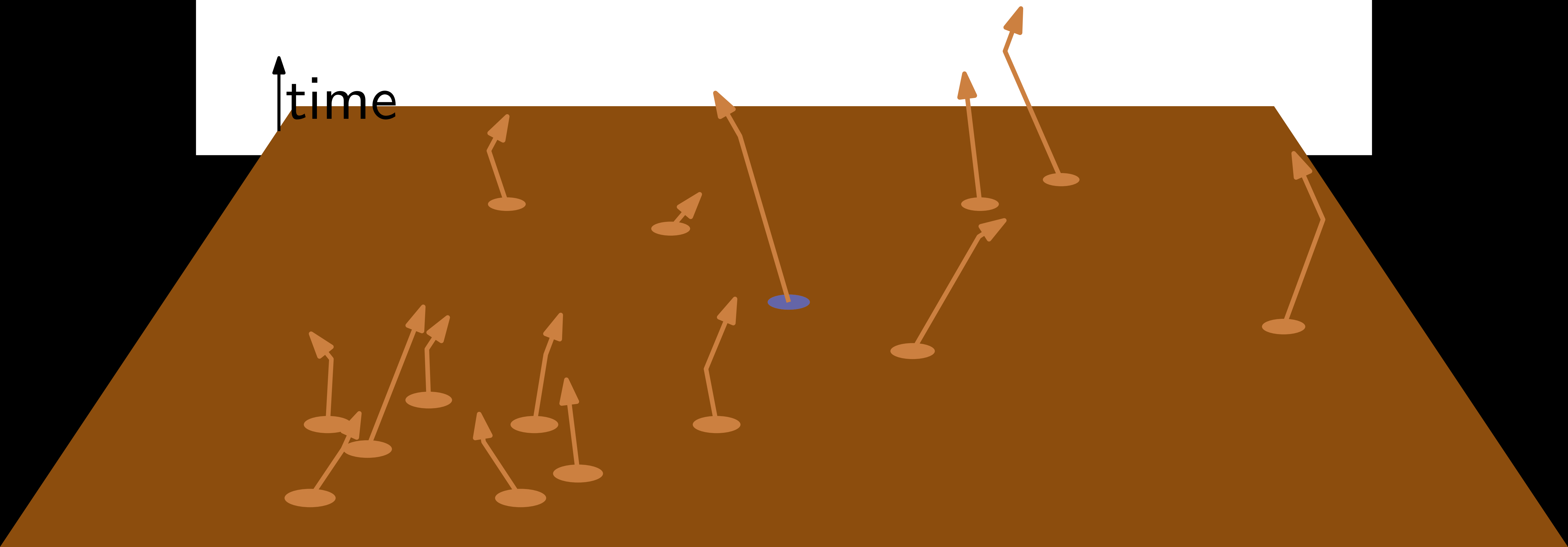


CENTRAL TRAJECTORIES



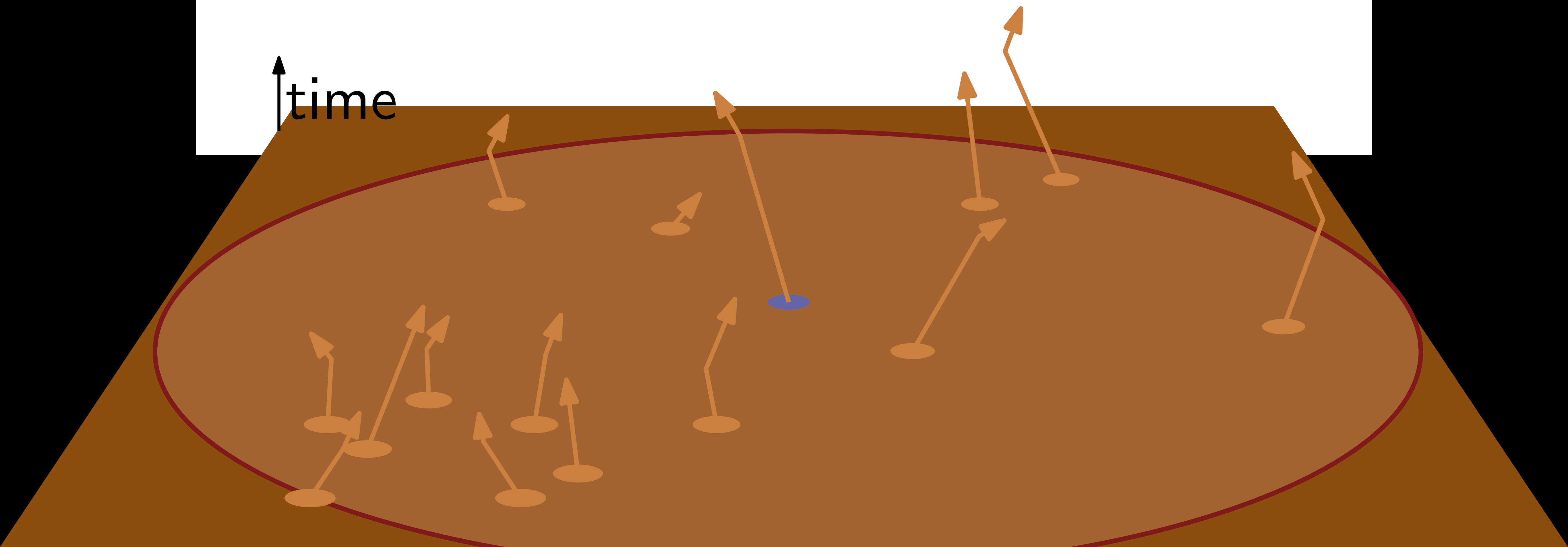
CENTRAL TRAJECTORIES

- At any time t, a central trajectory $\mathcal{C}$ should
  - be on an input trajectory, i.e. $\mathcal{C}(t) = \sigma(t)$ for some entity σ, and
  - be as central as possible.



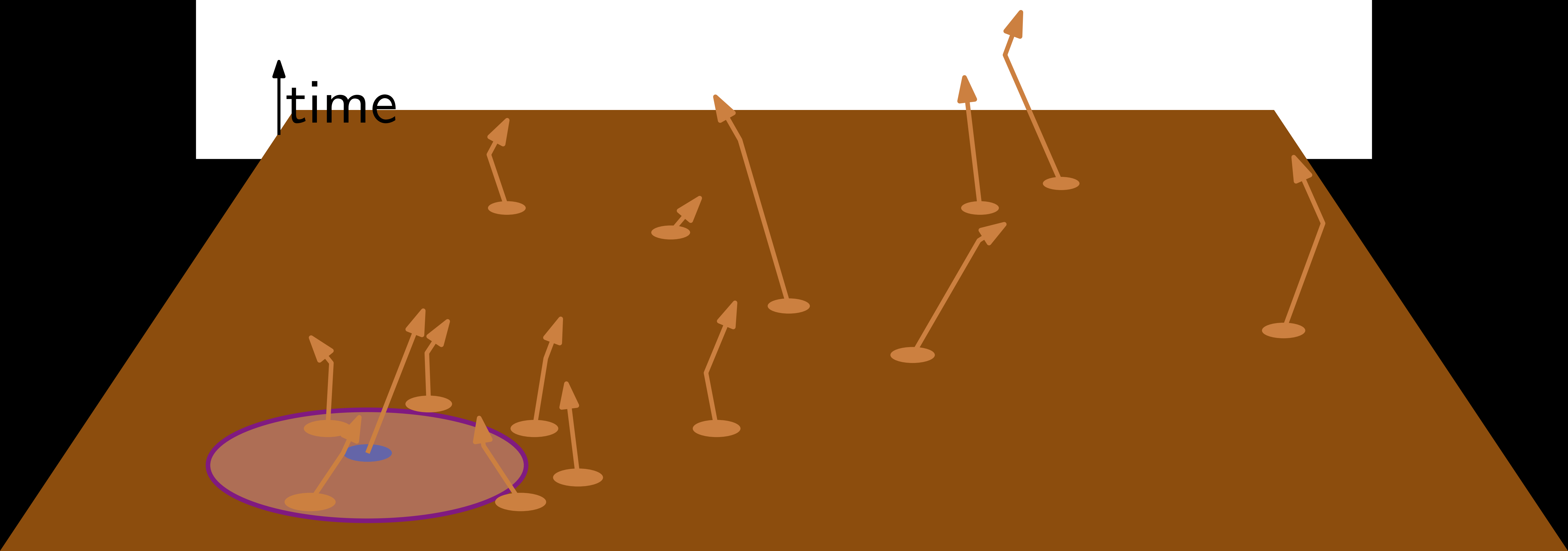
CENTRAL TRAJECTORIES

- At any time t, a central trajectory $\mathcal{C}$ should
  - be on an input trajectory, i.e. $\mathcal{C}(t) = \sigma(t)$ for some entity σ, and
  - be as central as possible i.e. minimize $D(\sigma, t) = \max_{\psi \in P} \|\sigma(t)\psi(t)\|$



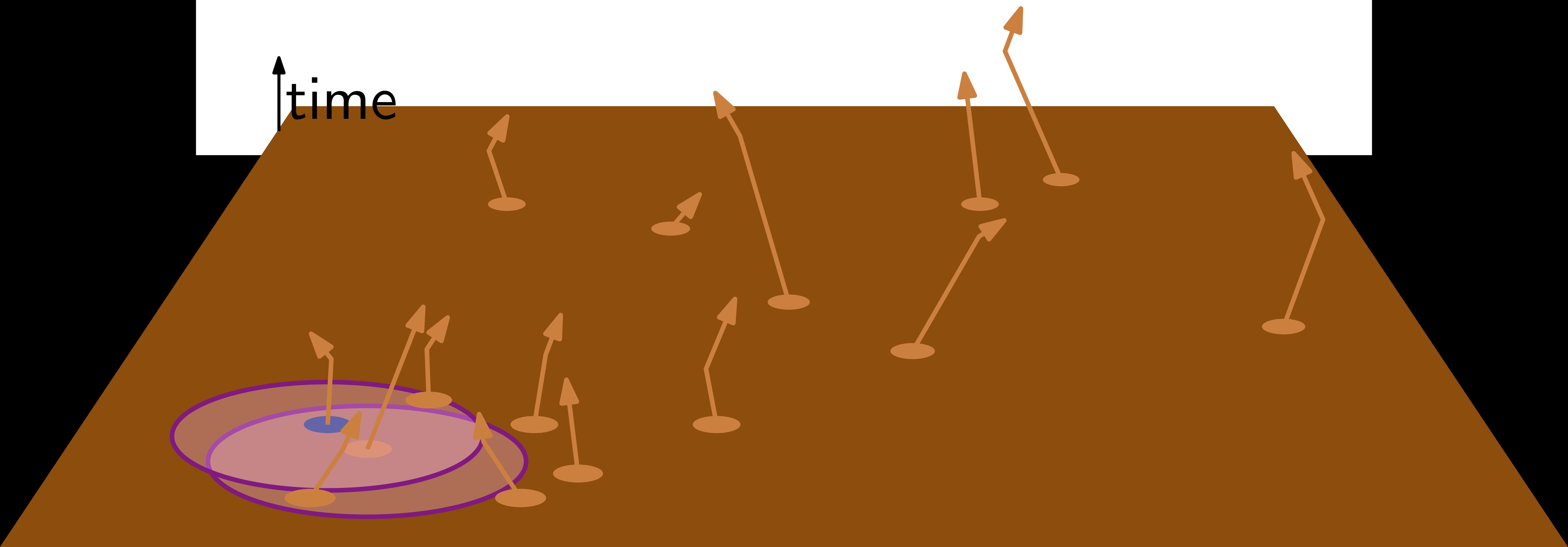
CENTRAL TRAJECTORIES

- At any time t, a central trajectory $\mathcal{C}$ should
  - be on an input trajectory, i.e. $\mathcal{C}(t) = \sigma(t)$ for some entity σ, and
  - be as central as possible.i.e. minimize
    $$D(\sigma, t) = \max_{\psi \in P} \|\sigma(t)\psi(t)\|$$
  - jump over a distance at most ε.



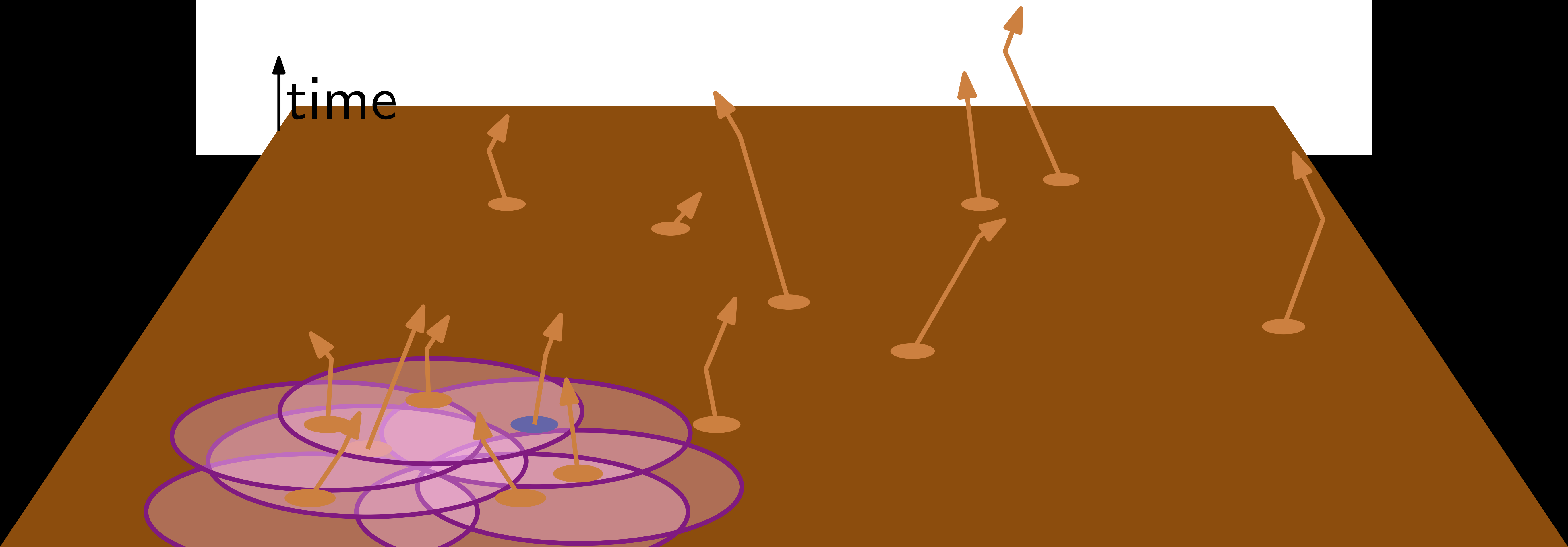
CENTRAL TRAJECTORIES

- At any time t, a central trajectory $\mathcal{C}$ should
  - be on an input trajectory, i.e. $\mathcal{C}(t) = \sigma(t)$ for some entity σ, and
  - be as central as possible.i.e. minimize
    $$D(\sigma, t) = \max_{\psi \in P} \|\sigma(t)\psi(t)\|$$
  - jump over a distance at most ε.
  - jumps may be chained.



CENTRAL TRAJECTORIES

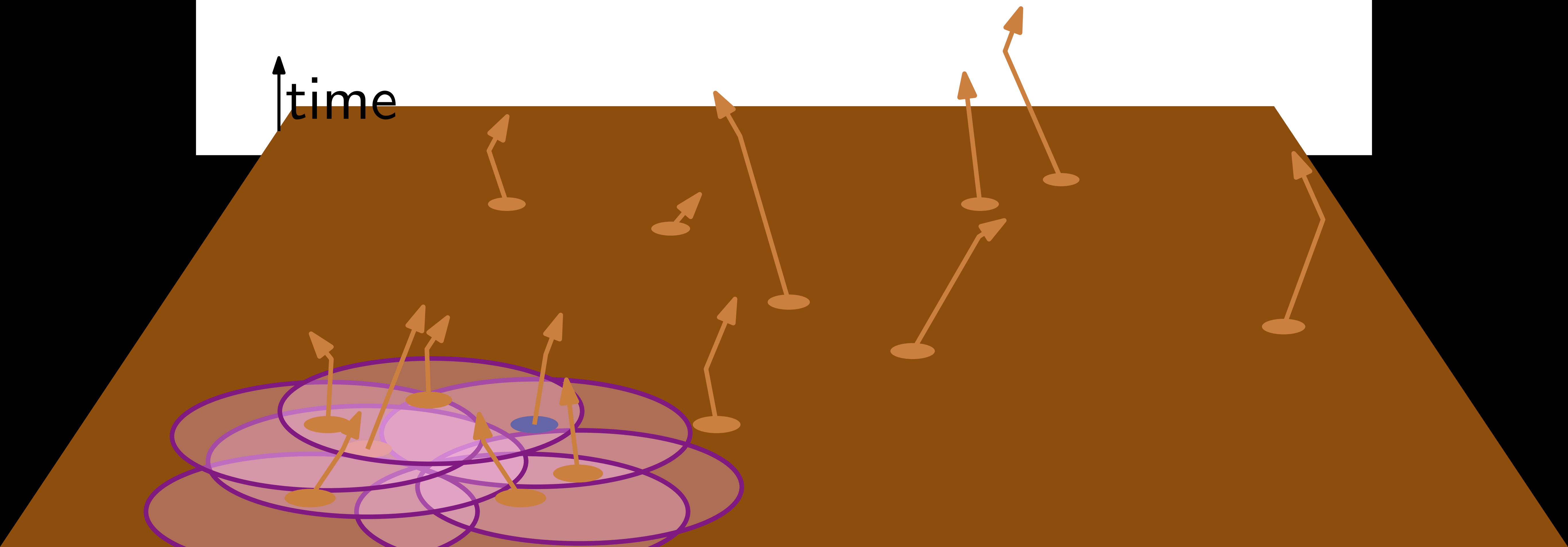
- At any time t, a central trajectory $\mathcal{C}$ should
  - be on an input trajectory, i.e. $\mathcal{C}(t) = \sigma(t)$ for some entity σ, and
  - be as central as possible.i.e. minimize
    $$D(\sigma, t) = \max_{\psi \in P} \|\sigma(t)\psi(t)\|$$
  - jump over a distance at most ε.
  - jumps may be chained.



CENTRAL TRAJECTORIES

- Globally, $\mathcal{C}$ should minimize

$$\mathcal{D}(\mathcal{T}) = \int_{t_0}^{t_\tau} D(\mathcal{T}, t)\,\mathrm{d}t.$$



RESULTS

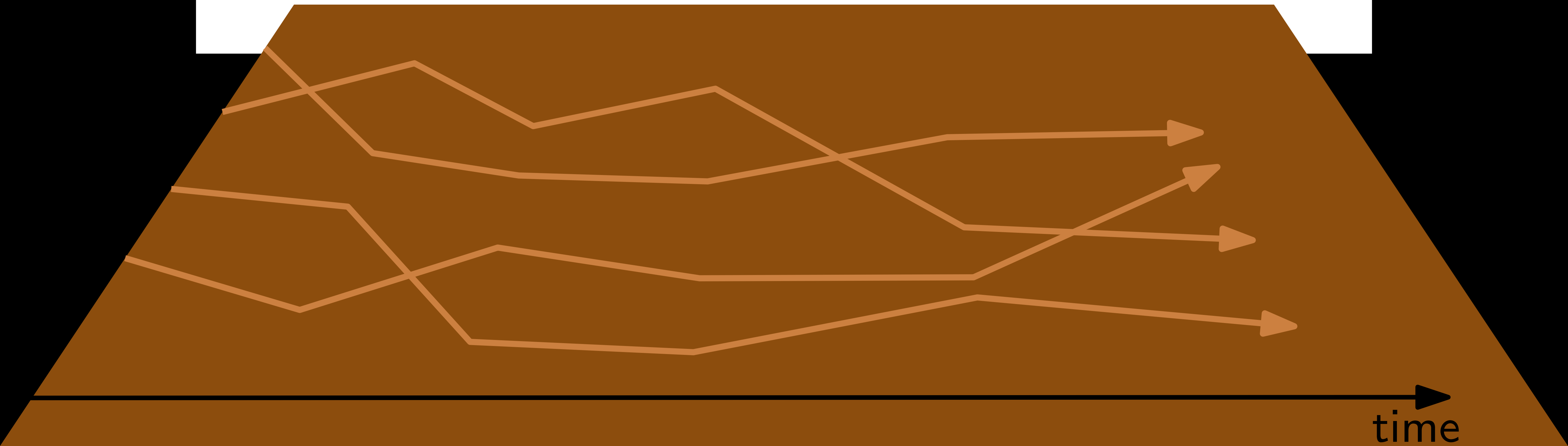
	Complexity $\mathcal{C}$		Algorithm
	Lower bound	Upper bound	
$\mathbb{R}^1$	$\Omega(\tau n^2)$	$O(\tau n^2)$	$O(\tau n^2 \log n)$
$\mathbb{R}^d$	$\Omega(\tau n^2)$	$O(\tau n^2 \sqrt{n})$	$O(\tau n^3)$

- n = #trajectories
- τ = #vertices in each trajectory

RESULTS

	Complexity $\mathcal{C}$		Algorithm
	Lower bound	Upper bound	
$\mathbb{R}^1$	$\Omega(\tau n^2)$	$O(\tau n^2)$	$O(\tau n^2 \log n)$
$\mathbb{R}^d$	$\Omega(\tau n^2)$	$O(\tau n^2 \sqrt{n})$	$O(\tau n^3)$

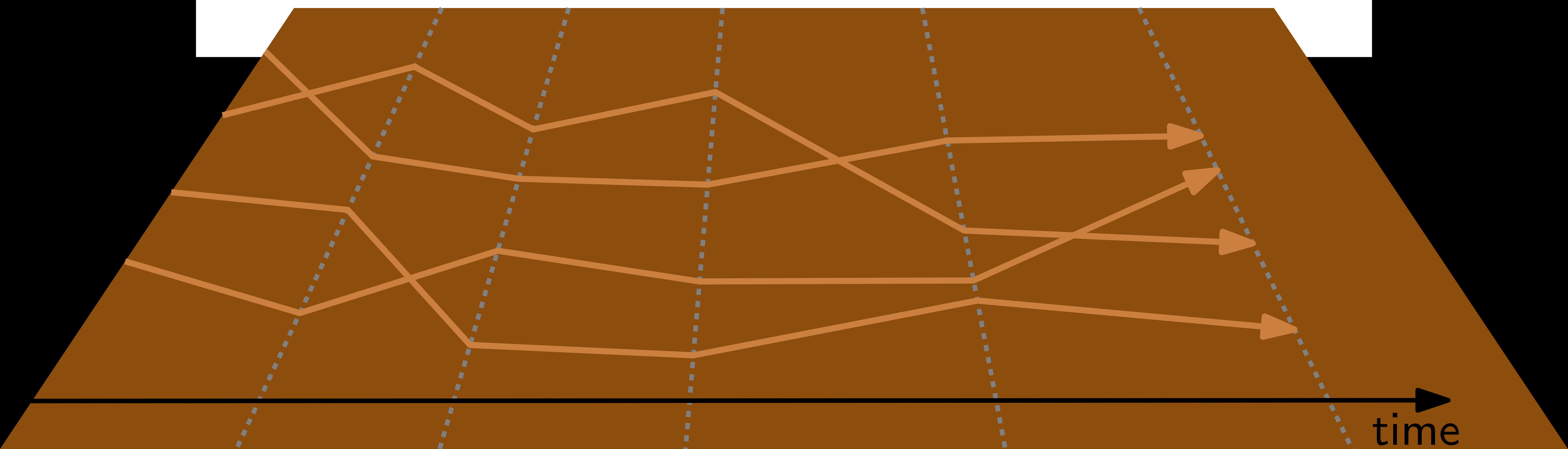
- $n = \#$trajectories
- $\tau = \#$vertices in each trajectory



RESULTS

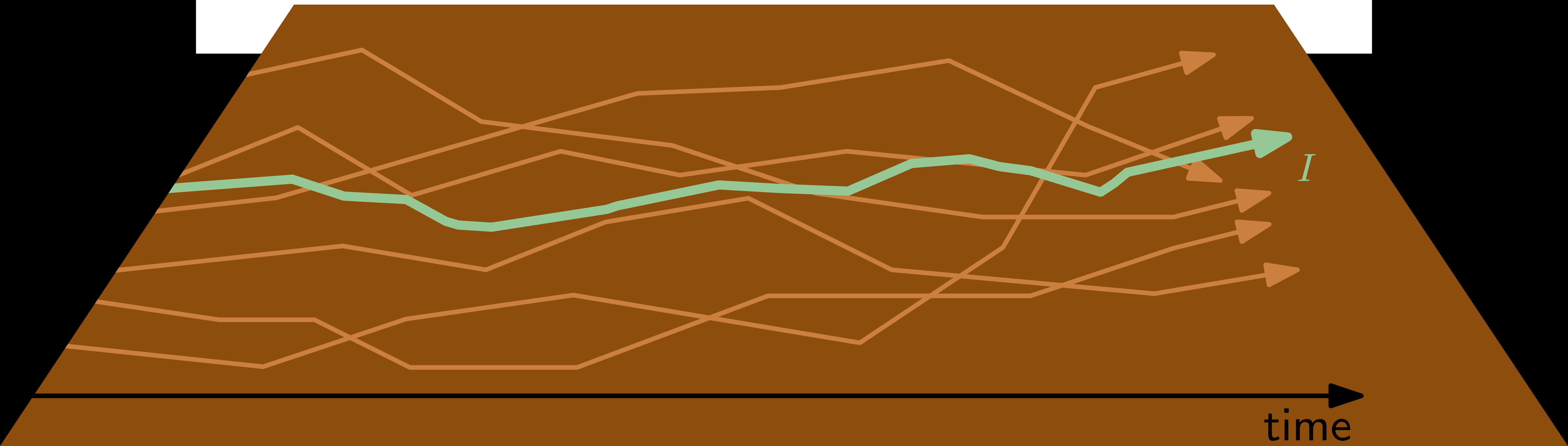
	Complexity $\mathcal{C}$ Lower bound	Complexity $\mathcal{C}$ Upper bound	Algorithm
$\mathbb{R}^1$	$\Omega(\tau n^2)$	$O(\tau n^2)$	$O(\tau n^2 \log n)$
$\mathbb{R}^d$	$\Omega(\tau n^2)$	$O(\tau n^2 \sqrt{n})$	$O(\tau n^3)$

- n = #trajectories
- τ = #vertices in each trajectory



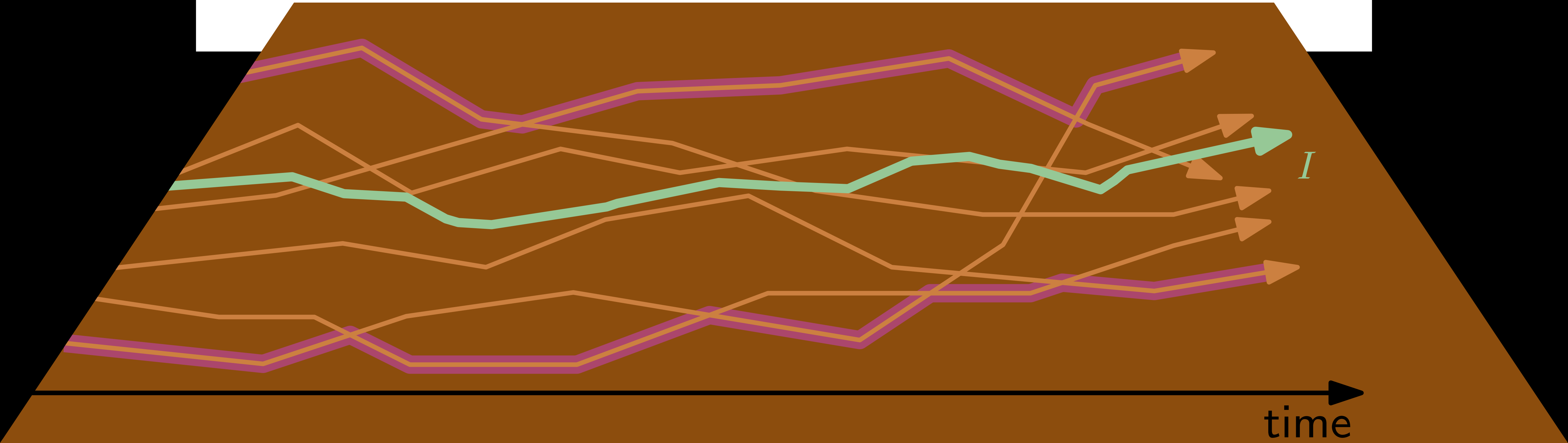
THE IDEAL TRAJECTORY

- Let I be the *ideal trajectory*: the trajectory that minimizes $\mathcal{D}$.



THE IDEAL TRAJECTORY

- Let I be the *ideal trajectory*: the trajectory that minimizes $\mathcal{D}$.



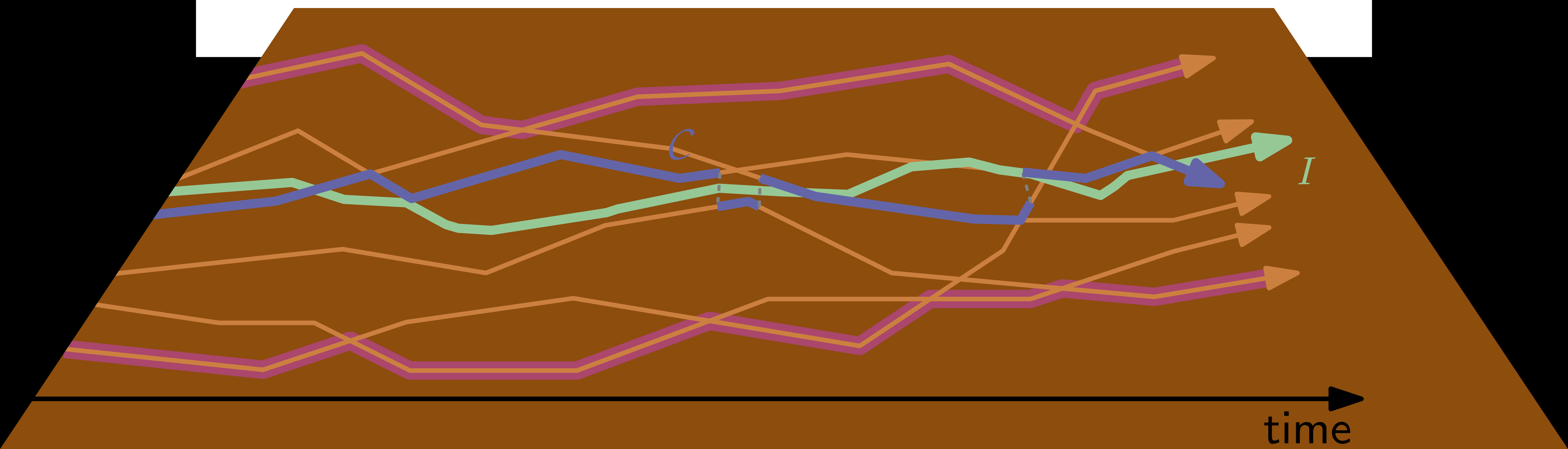
THE IDEAL TRAJECTORY

- Let $\mathcal{I}$ be the *ideal trajectory*: the trajectory that minimizes $\mathcal{D}$.

LEMMA.

A central trajectory $\mathcal{C}$ minimizes

$$\mathcal{D}'(\mathcal{T}) = \int |\mathcal{T}(t) - \mathcal{I}(t)| \, \mathrm{d}t$$



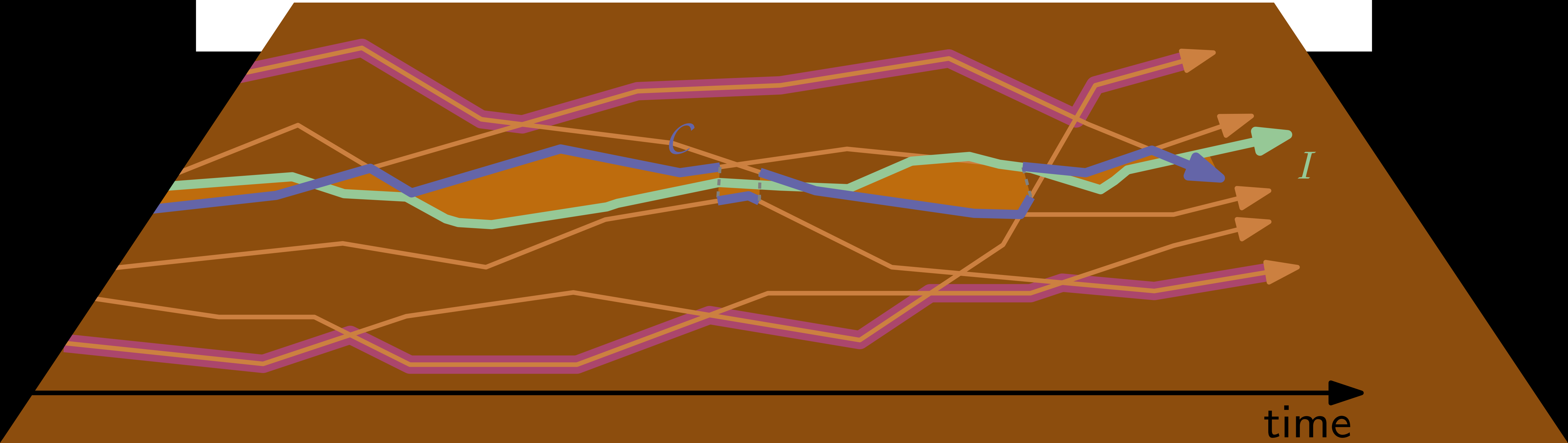
THE IDEAL TRAJECTORY

- Let $\mathcal{I}$ be the *ideal trajectory*: the trajectory that minimizes $\mathcal{D}$.

LEMMA.

A central trajectory $\mathcal{C}$ minimizes

$$\mathcal{D}'(\mathcal{T}) = \int |\mathcal{T}(t) - \mathcal{I}(t)| \, \mathrm{d}t$$



LOWER BOUND

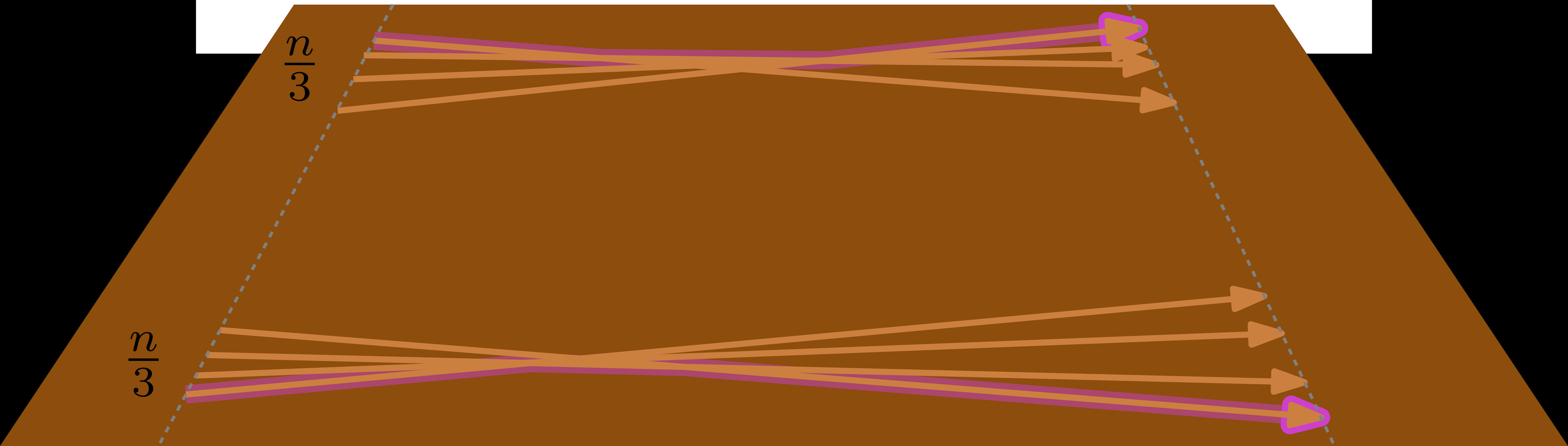
LEMMA.

A central trajectory $\mathcal{C}$ in $\mathbb{R}^1$ may have complexity $\Omega(\tau n^2)$.

LOWER BOUND

LEMMA.

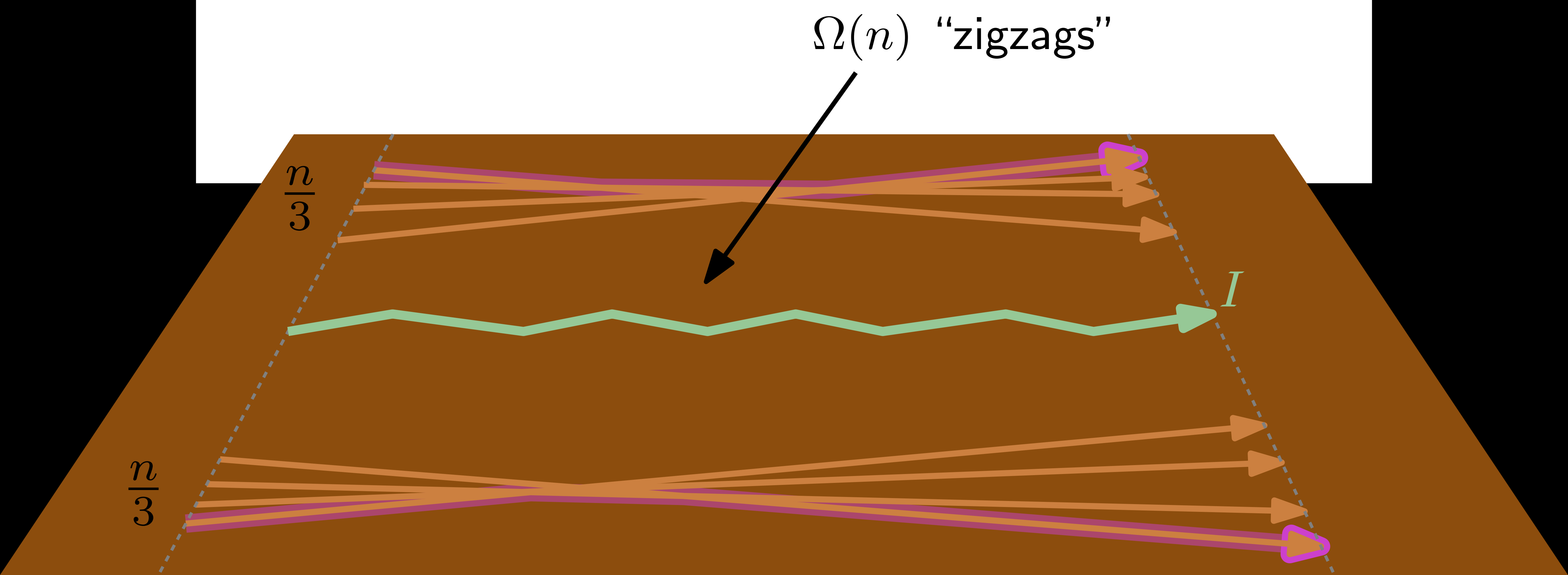
A central trajectory $\mathcal{C}$ in $\mathbb{R}^1$ may have complexity $\Omega(\tau n^2)$.



LOWER BOUND

LEMMA.

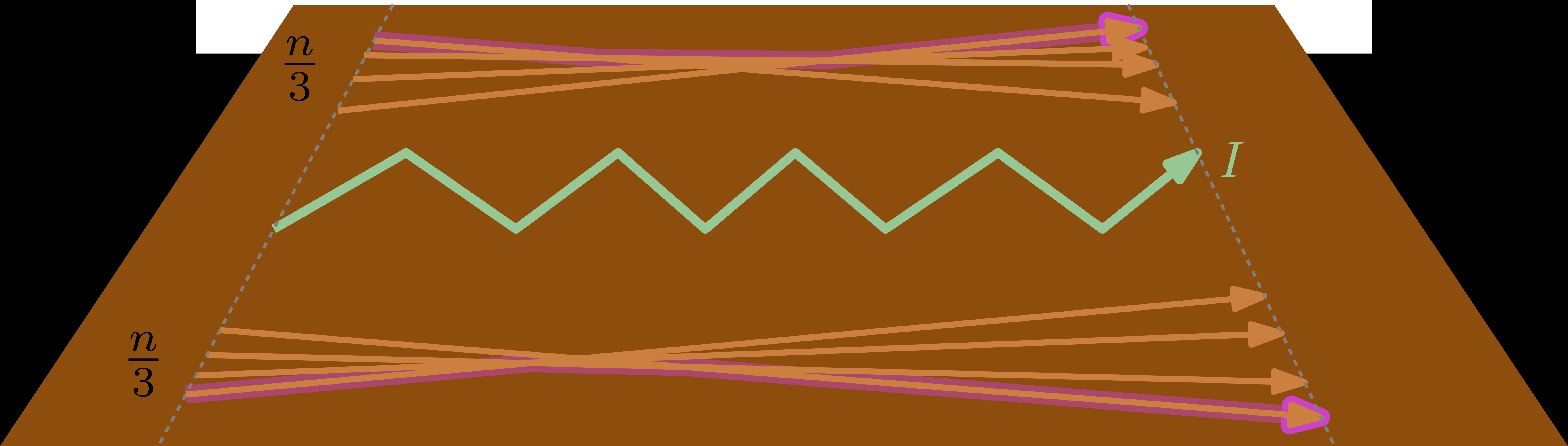
A central trajectory $\mathcal{C}$ in $\mathbb{R}^1$ may have complexity $\Omega(\tau n^2)$.



LOWER BOUND

LEMMA.

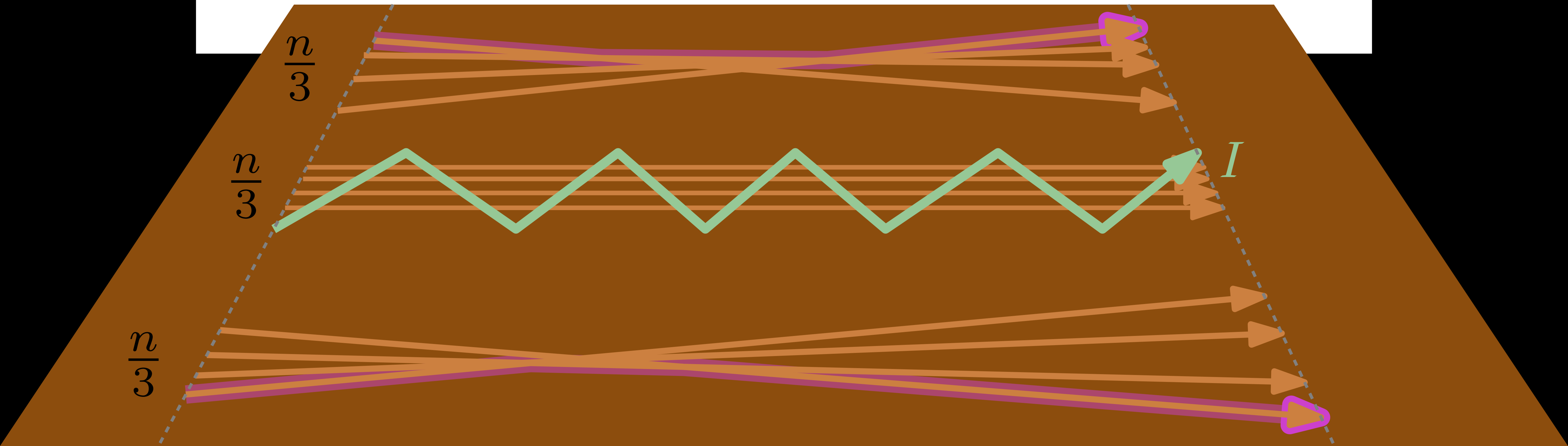
A central trajectory $\mathcal{C}$ in $\mathbb{R}^1$ may have complexity $\Omega(\tau n^2)$.



LOWER BOUND

LEMMA.

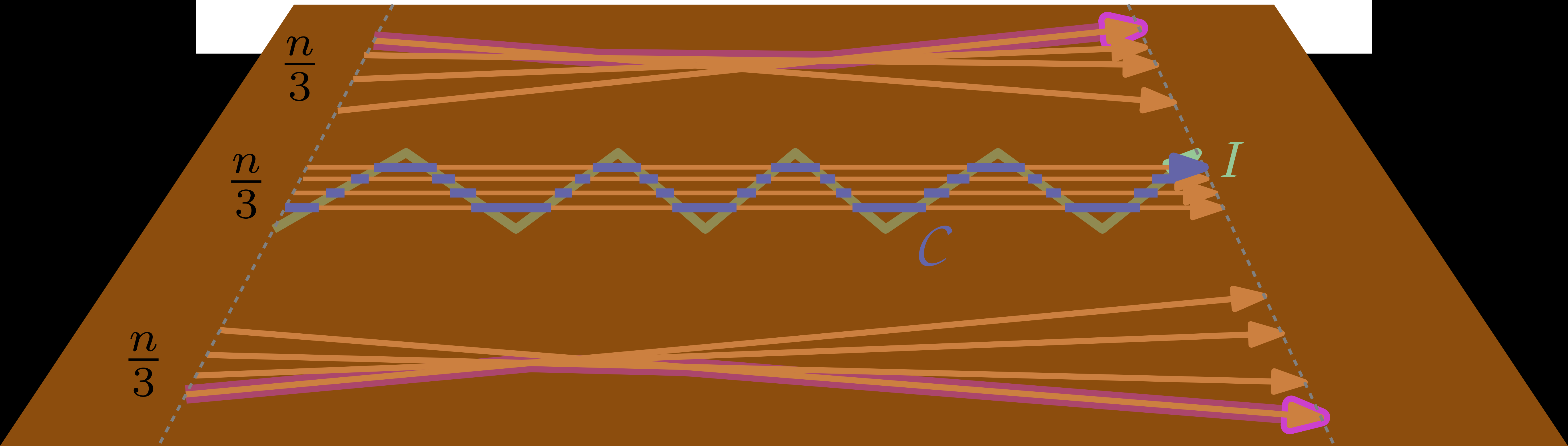
A central trajectory $\mathcal{C}$ in $\mathbb{R}^1$ may have complexity $\Omega(\tau n^2)$.



LOWER BOUND

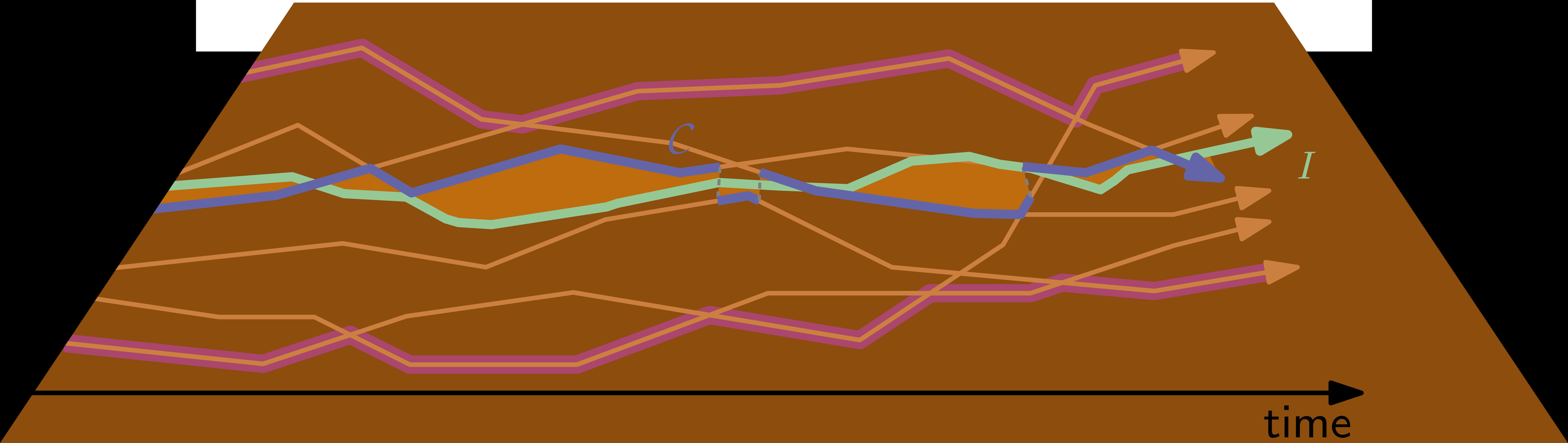
LEMMA.

A central trajectory $\mathcal{C}$ in $\mathbb{R}^1$ may have complexity $\Omega(\tau n^2)$.



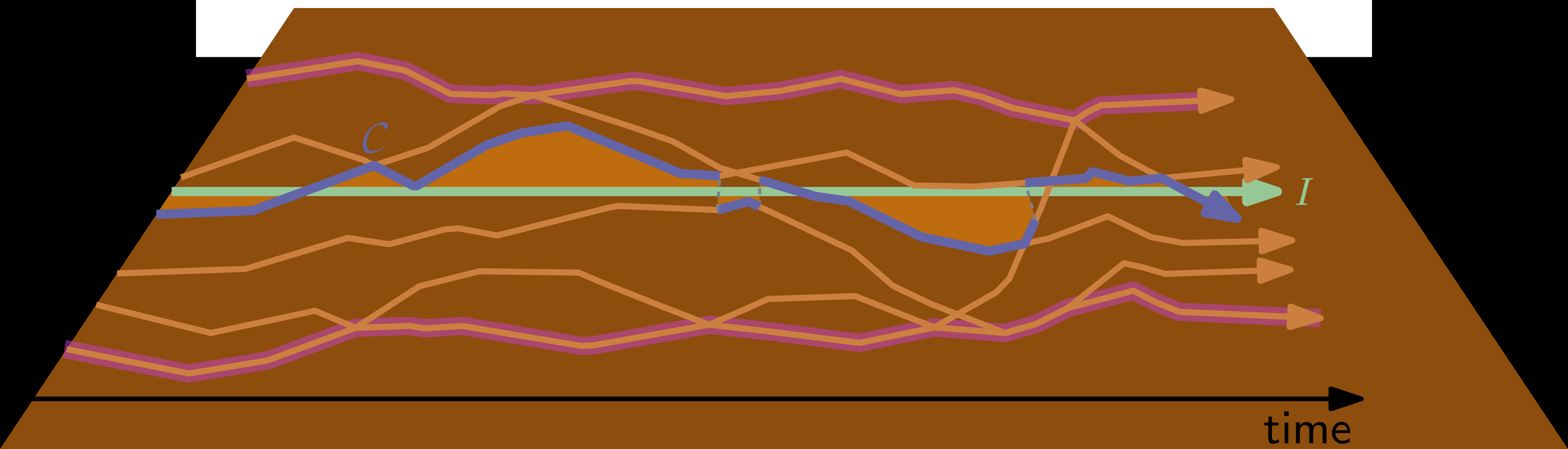
UPPER BOUND

- "Straighten" I



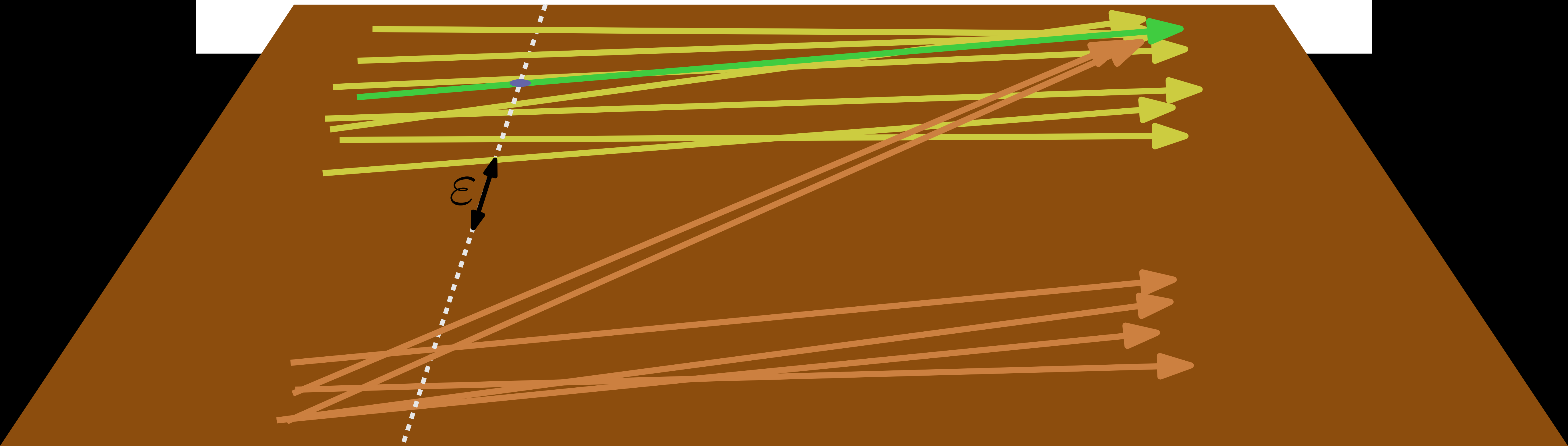
UPPER BOUND

- "Straighten" I
  - We now have n trajectories, with $O(\tau n)$ vertices each



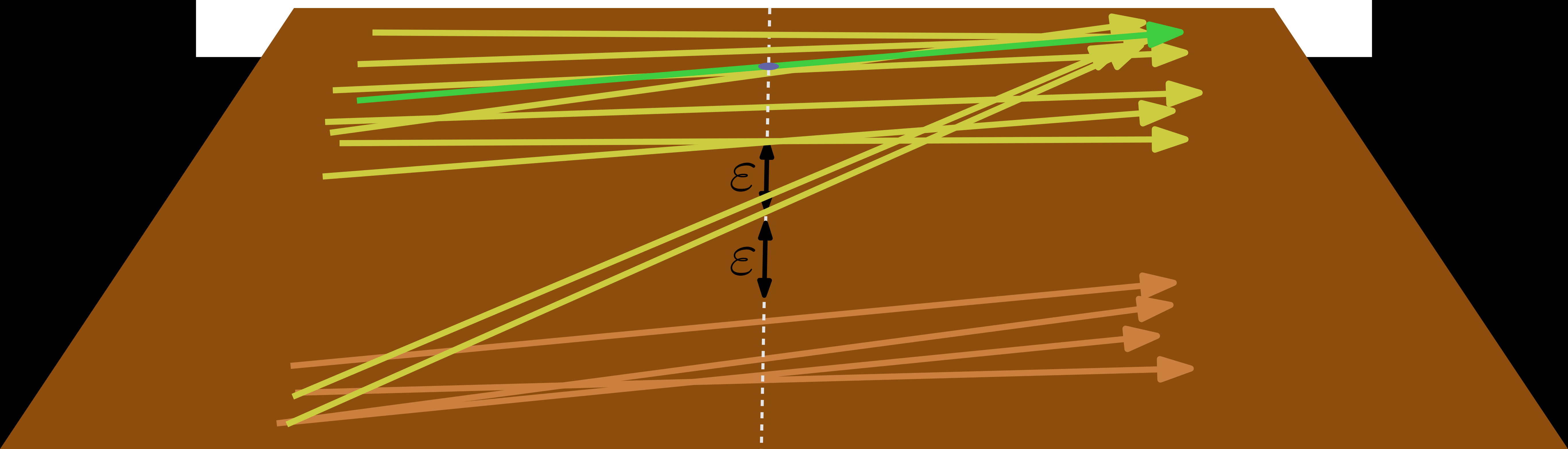
UPPER BOUND

OBSERVATION.
$\mathcal{C}$ can jump from σ to ψ at time t
$\Leftrightarrow$
σ and ψ are ε-connected at time t.



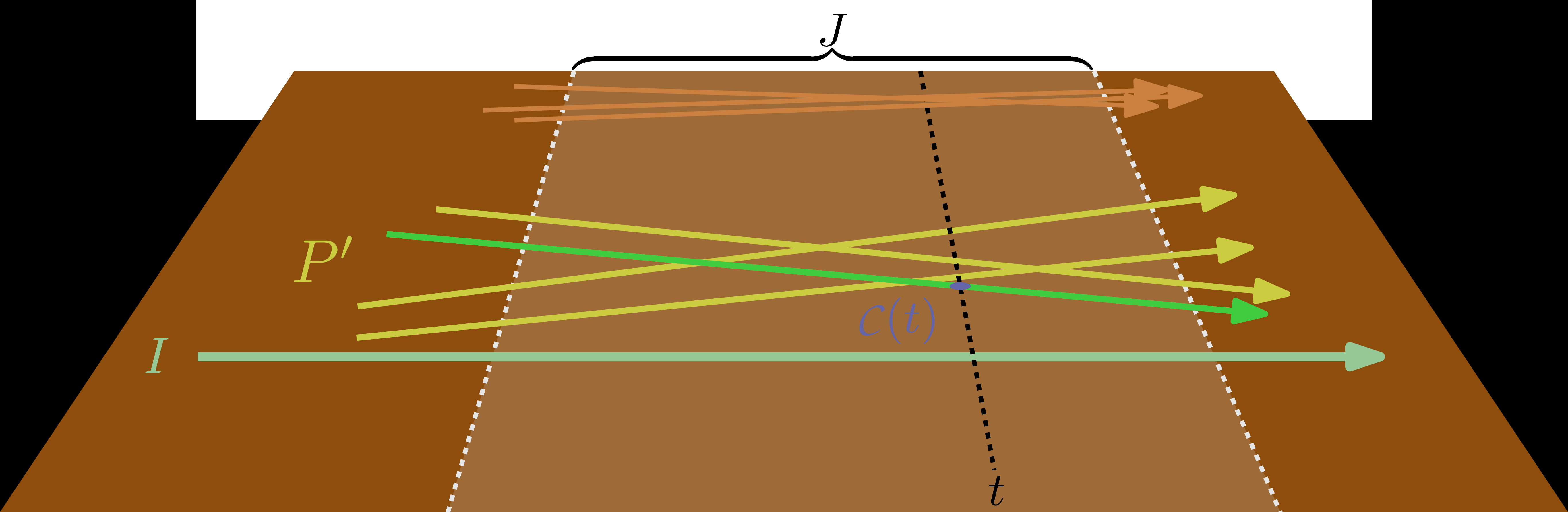
UPPER BOUND

OBSERVATION.
$\mathcal{C}$ can jump from σ to ψ at time t
$\Leftrightarrow$
σ and ψ are ε-connected at time t.



UPPER BOUND

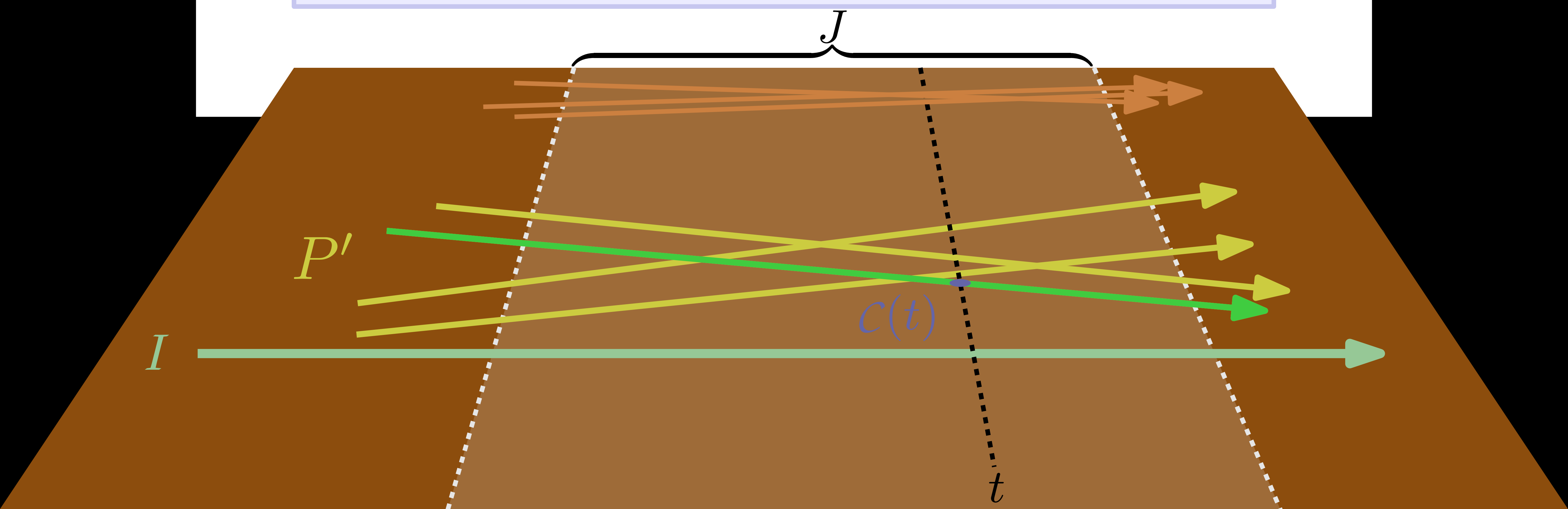
- Let $P' \ni \sigma$ be a maximal set of entities that is ε-connected during interval J, and s.t. $\mathcal{C} \in P'$ during J.



UPPER BOUND

- Let $P' \ni \sigma$ be a maximal set of entities that is ε-connected during interval J, and s.t. $\mathcal{C} \in P'$ during J.

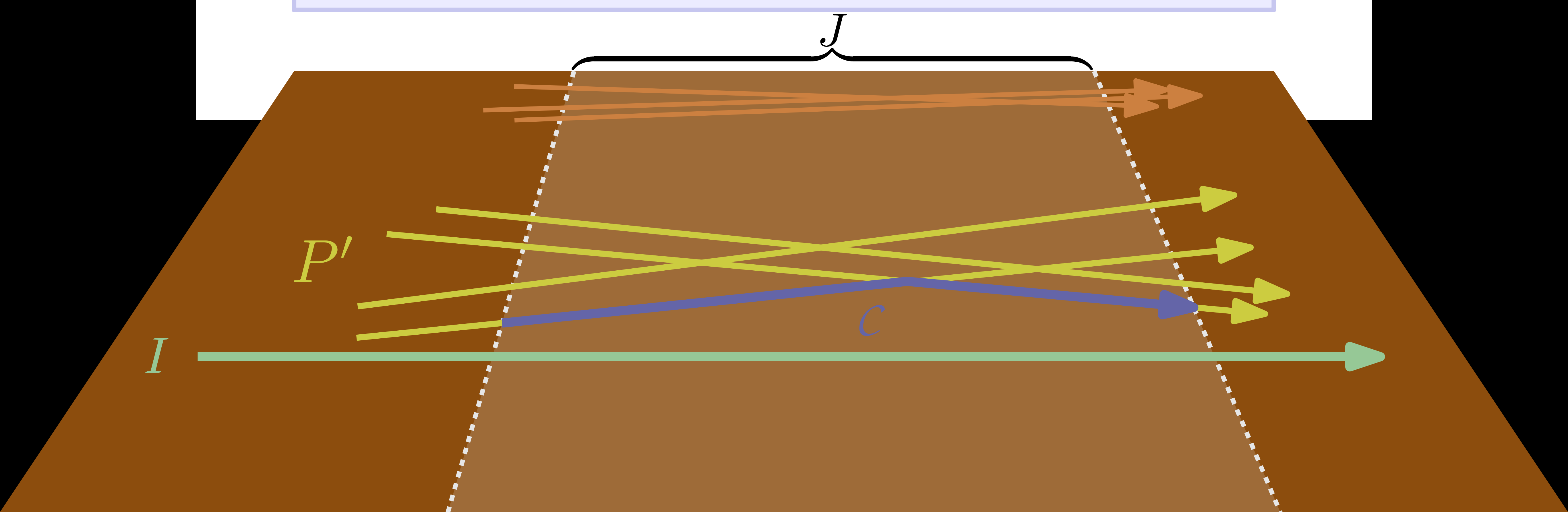
OBSERVATION.
For any time $t \in J$, $\mathcal{C}(t) = \sigma(t)$
$\Leftrightarrow$
f_σ is on the lower envelope of $\{f_\psi \mid \psi \in \mathcal{X}'\}$ at time t, where $f_\sigma(t) = |\sigma(t)|$.



UPPER BOUND

- Let $P' \ni \sigma$ be a maximal set of entities that is ε-connected during interval J, and s.t. $\mathcal{C} \in P'$ during J.

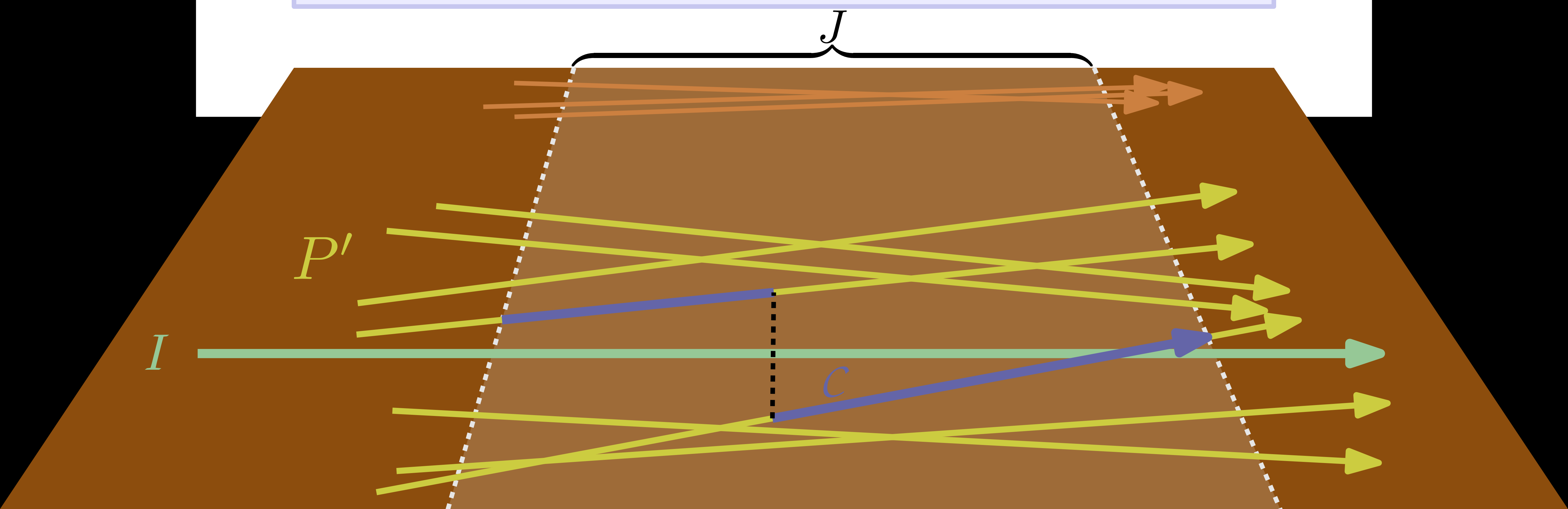
OBSERVATION.
For any time $t \in J$, $\mathcal{C}(t) = \sigma(t)$
$\Leftrightarrow$
f_σ is on the lower envelope of $\{f_\psi \mid \psi \in \mathcal{X}'\}$ at time t, where $f_\sigma(t) = |\sigma(t)|$.



UPPER BOUND

- Let $P' \ni \sigma$ be a maximal set of entities that is ε-connected during interval J, and s.t. $\mathcal{C} \in P'$ during J.

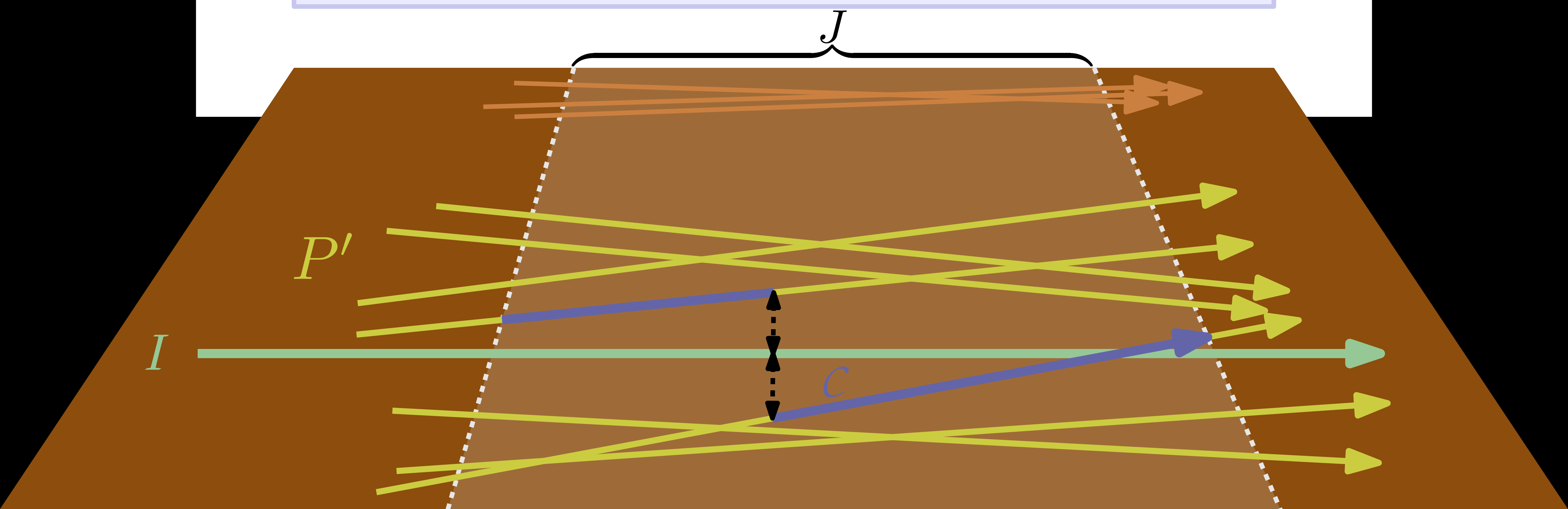
OBSERVATION.
For any time $t \in J$, $\mathcal{C}(t) = \sigma(t)$
$\Leftrightarrow$
f_σ is on the lower envelope of $\{f_\psi \mid \psi \in \mathcal{X}'\}$ at time t, where $f_\sigma(t) = |\sigma(t)|$.



UPPER BOUND

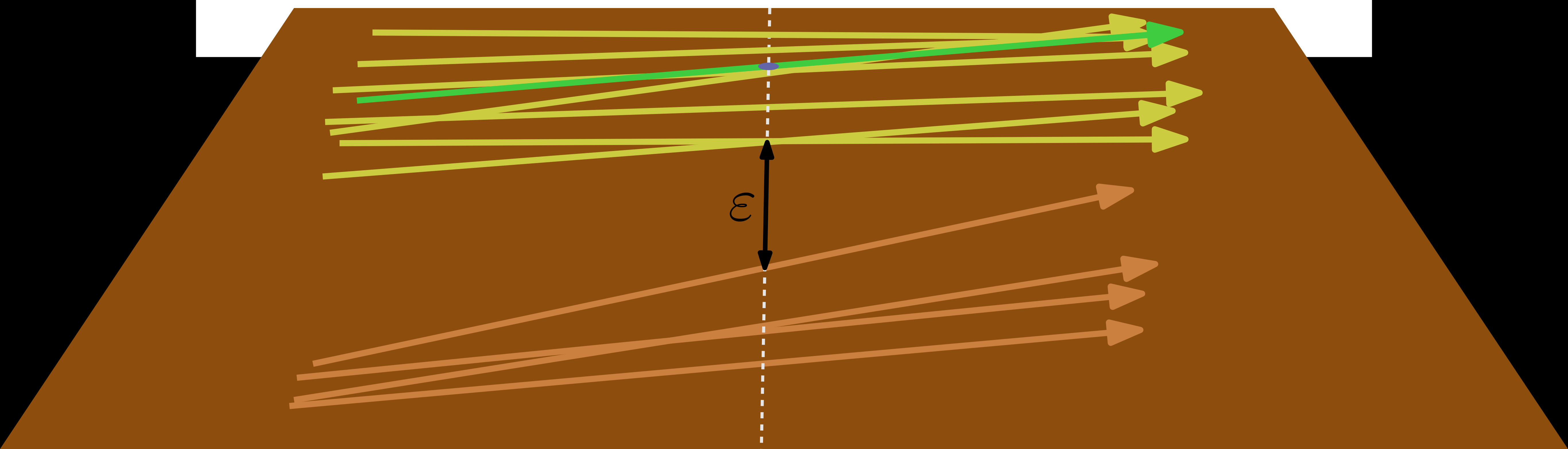
- Let $P' \ni \sigma$ be a maximal set of entities that is ε-connected during interval J, and s.t. $\mathcal{C} \in P'$ during J.

OBSERVATION.
For any time $t \in J$, $\mathcal{C}(t) = \sigma(t)$
$\Leftrightarrow$
f_σ is on the lower envelope of $\{f_\psi \mid \psi \in \mathcal{X}'\}$ at time t, where $f_\sigma(t) = |\sigma(t)|$.



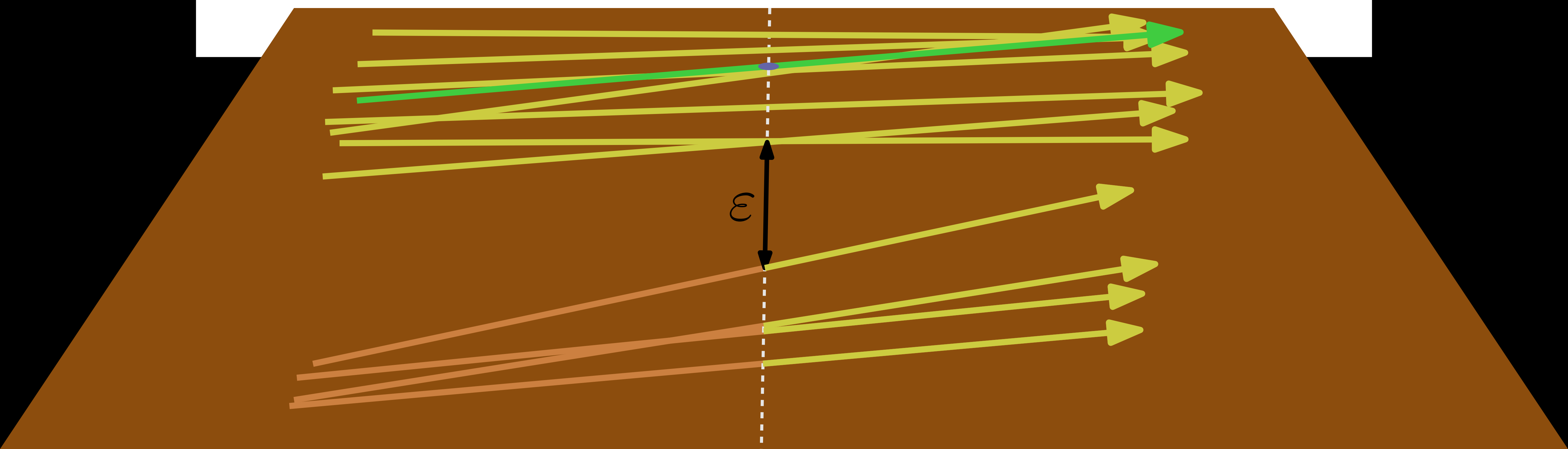
UPPER BOUND

- $\mathcal{C}$ may have a vertex at time t if:
  - Partition into ε-connected sets changes at time t, or



UPPER BOUND

- $\mathcal{C}$ may have a vertex at time t if:
  - Partition into ε-connected sets changes at time t, or

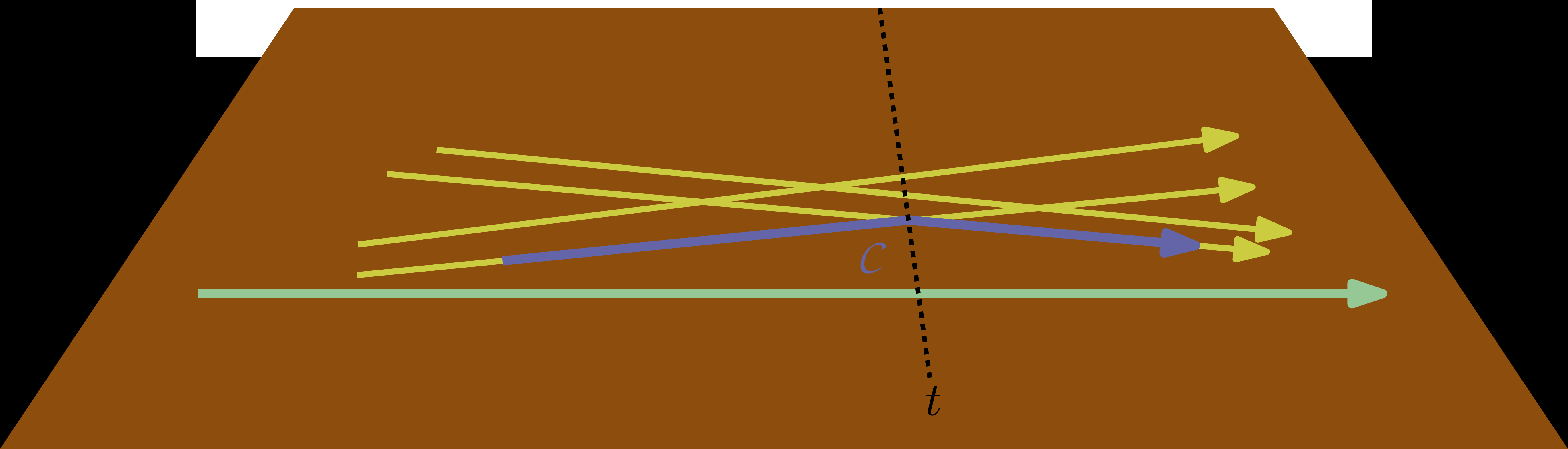


UPPER BOUND

- $\mathcal{C}$ may have a vertex at time t if:
  - Partition into ε-connected sets changes at time t, or
  - Lower envelope $\mathcal{L}'$ of a maximal ε-connected set P' has a break point at t.

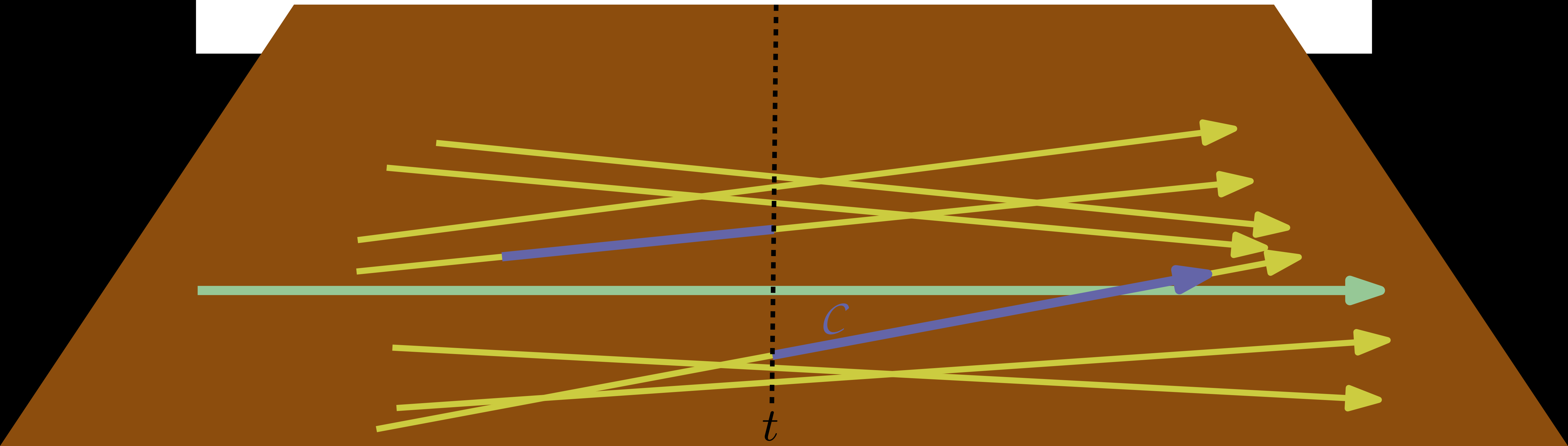
UPPER BOUND

- $\mathcal{C}$ may have a vertex at time t if:
  - Partition into ε-connected sets changes at time t, or
  - Lower envelope $\mathcal{L}'$ of a maximal ε-connected set P' has a break point at t.
    - Intersection of the trajectories



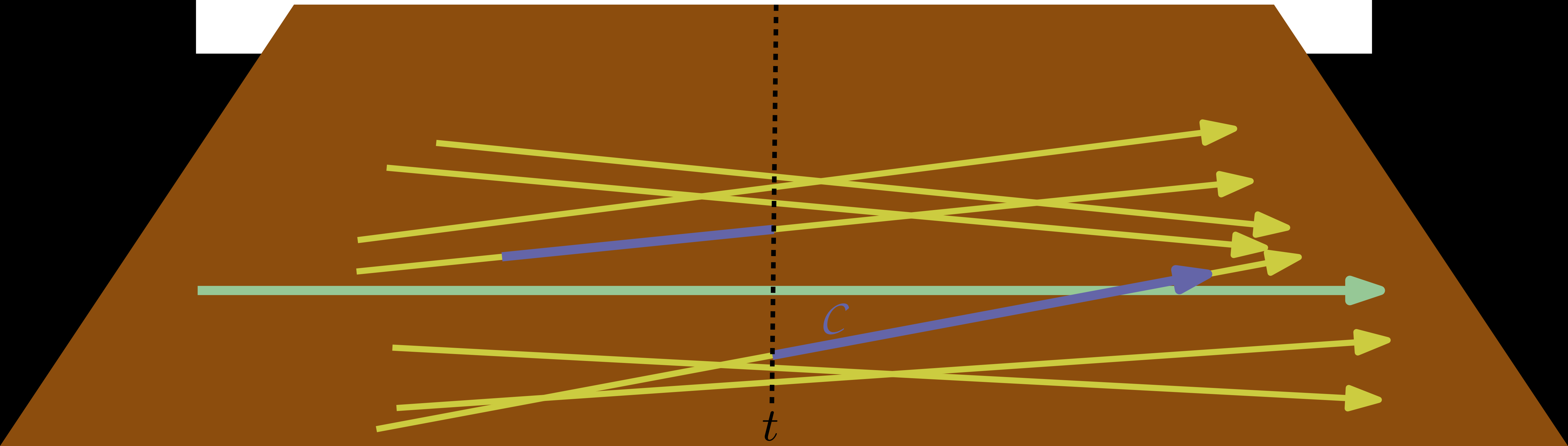
UPPER BOUND

- $\mathcal{C}$ may have a vertex at time t if:
  - Partition into ε-connected sets changes at time t, or
  - Lower envelope $\mathcal{L}'$ of a maximal ε-connected set P' has a break point at t.
    - Intersection of the trajectories
    - Jump crossing I



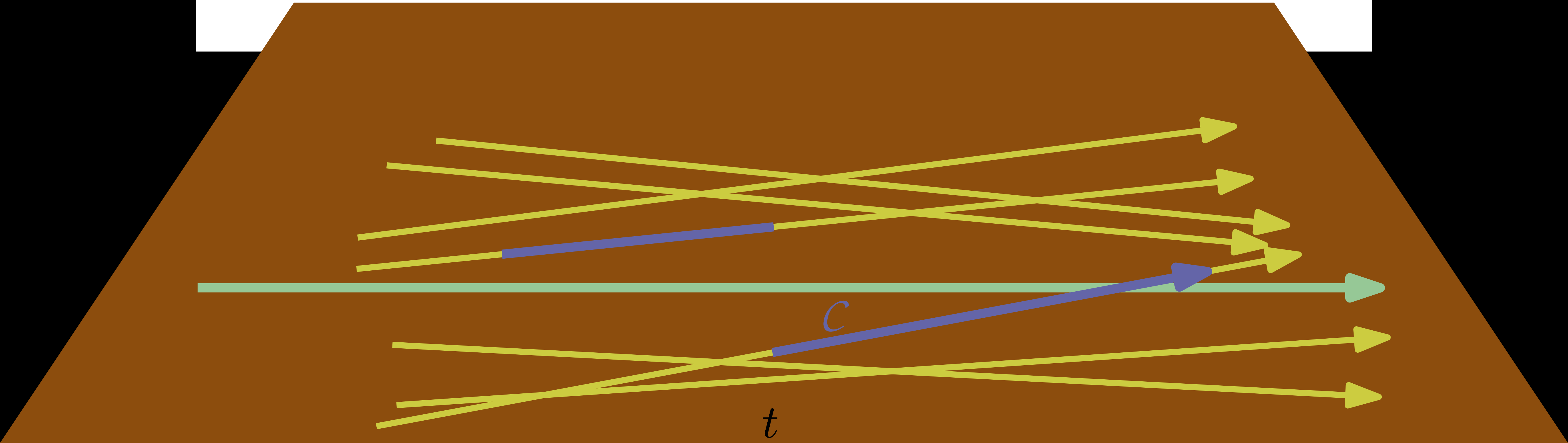
UPPER BOUND

- $\mathcal{C}$ may have a vertex at time t if:
  - Partition into ε-connected sets changes at time t, or $O(\tau n^2)$
  - Lower envelope $\mathcal{L}'$ of a maximal ε-connected set P' has a break point at t.
    - Intersection of the trajectories $O(\tau n^2)$
    - Jump crossing I $O(\tau n^2)$



UPPER BOUND

THEOREM.
Given a set of n trajectories in $\mathbb{R}^1$, each with vertices at times $t_0, .., t_\tau$, a central trajectory $\mathcal{C}$ has worst case complexity $O(\tau n^2)$.

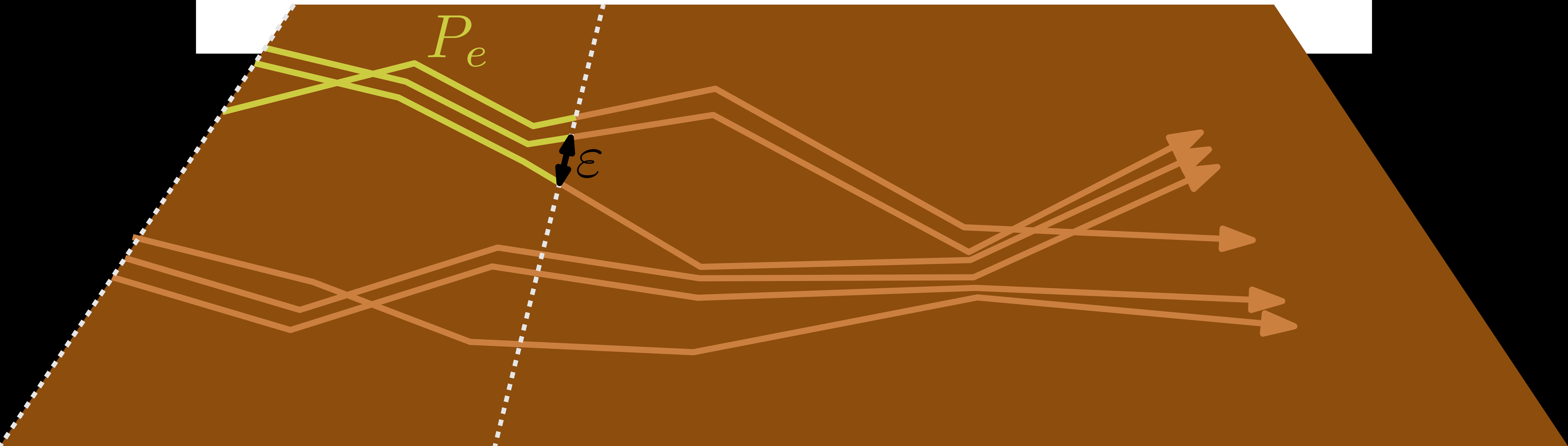


ALGORITHM

- Construct a weighted graph $\mathcal{R}$ s.t. $\mathcal{C}$ corresponds to a shortest path in $\mathcal{R}$.

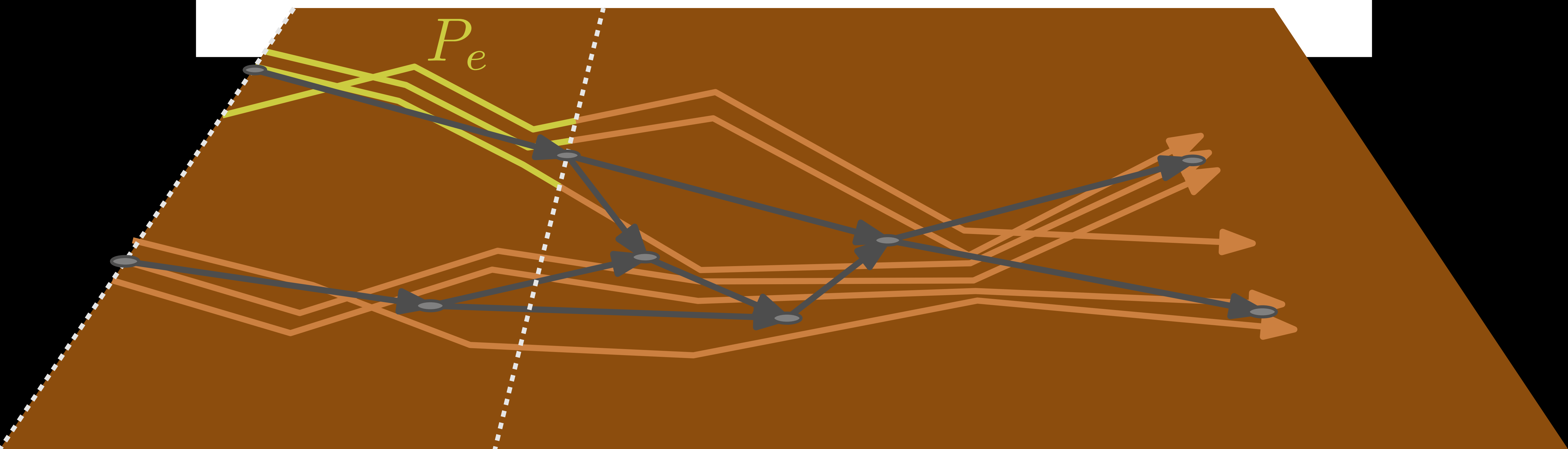
ALGORITHM

- Construct a weighted graph $\mathcal{R}$ s.t. $\mathcal{C}$ corresponds to a shortest path in $\mathcal{R}$.
  - The *Reeb graph* $\mathcal{R}$ captures which sets of entities are ε-connected.



ALGORITHM

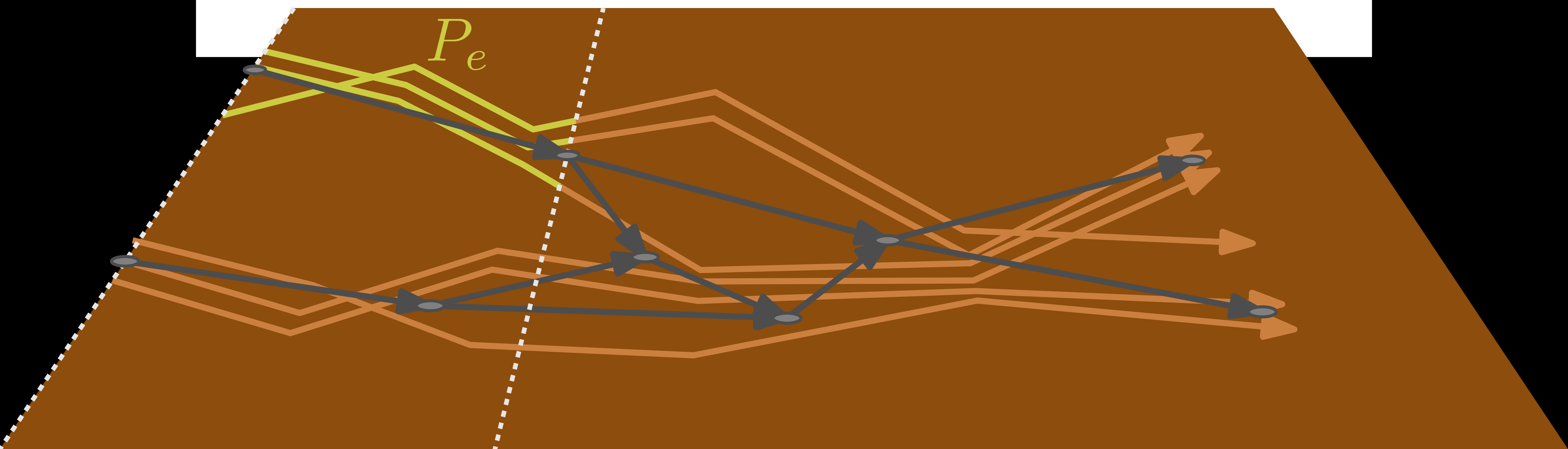
- Construct a weighted graph $\mathcal{R}$ s.t. $\mathcal{C}$ corresponds to a shortest path in $\mathcal{R}$.
  - The *Reeb graph* $\mathcal{R}$ captures which sets of entities are ε-connected.



ALGORITHM

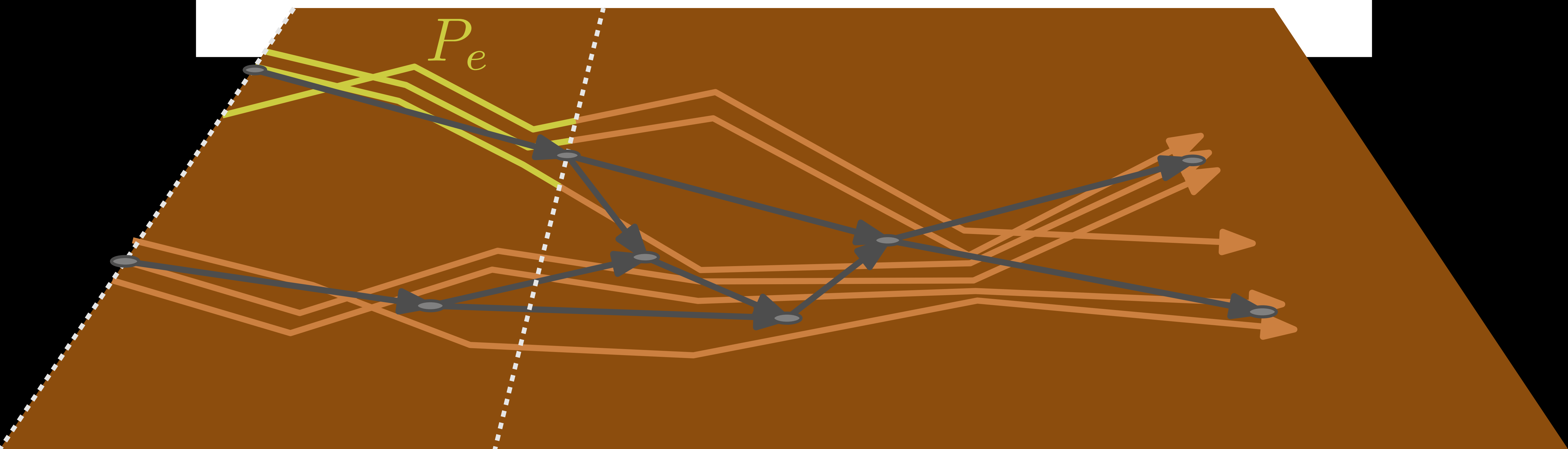
- Construct a weighted graph $\mathcal{R}$ s.t. $\mathcal{C}$ corresponds to a shortest path in $\mathcal{R}$.
  - The *Reeb graph* $\mathcal{R}$ captures which sets of entities are ε-connected.
  - Each edge e corresponds to a maximal ε-connected set P_e and has weight

$$\int \mathcal{L}(\{f_\sigma \mid \sigma \in P_e\})(t)\,\mathrm{d}t$$



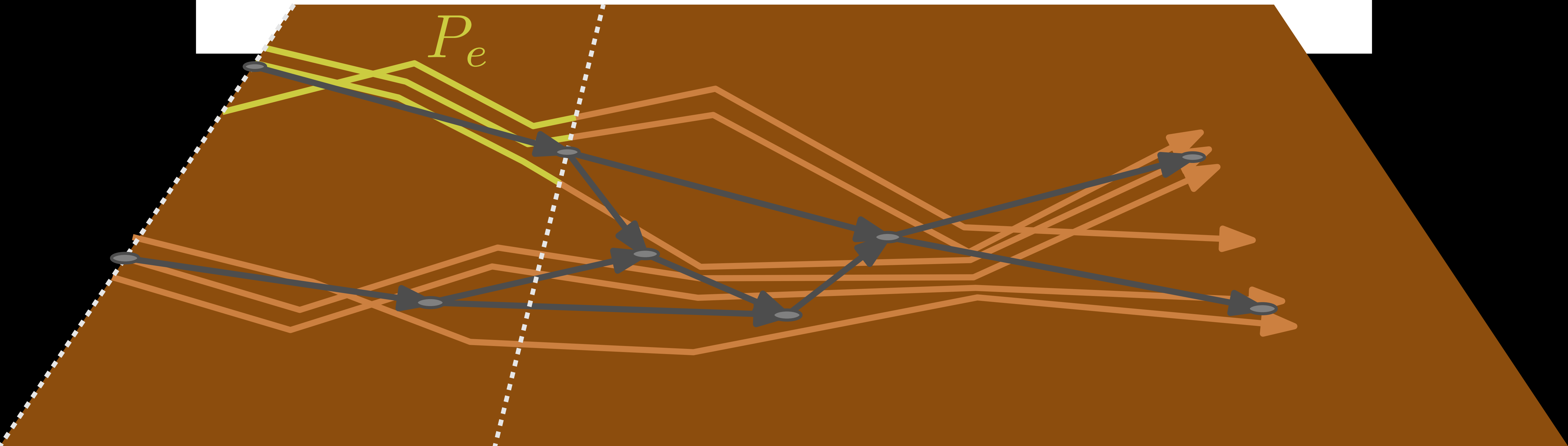
ALGORITHM

- Construct a weighted graph $\mathcal{R}$ s.t. $\mathcal{C}$ corresponds to a shortest path in $\mathcal{R}$.
  - The *Reeb graph* $\mathcal{R}$ captures which sets of entities are ε-connected.
  - Each edge e corresponds to a maximal ε-connected set P_e and has weight
    $$\int \mathcal{L}(\{f_\sigma \mid \sigma \in P_e\})(t)\,\mathrm{d}t$$
  - Constructing $\mathcal{R}$ takes $O(\tau n^2 \log n)$ time.



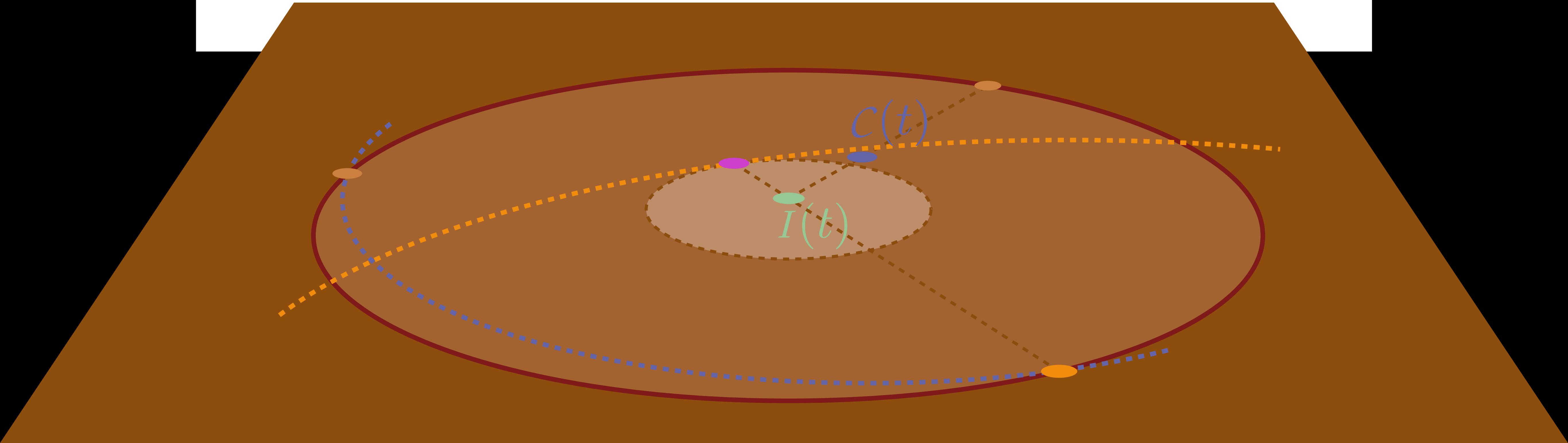
ALGORITHM

THEOREM.
Given a set of n trajectories in $\mathbb{R}^1$, each with vertices at times $t_0, .., t_\tau$, computing a central trajectory $\mathcal{C}$ takes $O(\tau n^2 \log n)$ time.



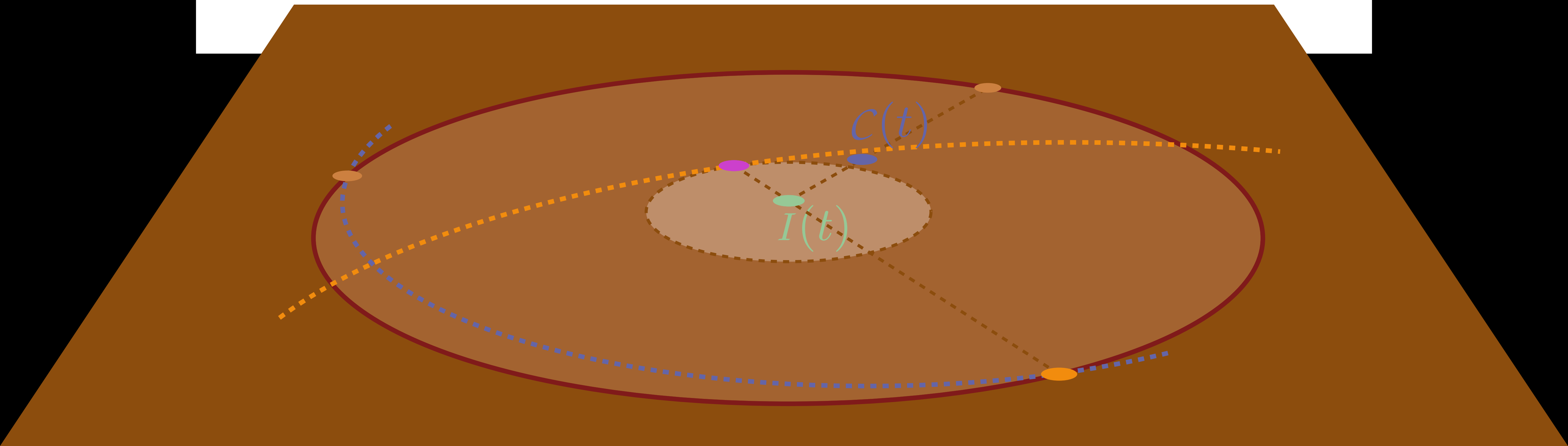
ENTITIES MOVING IN $\mathbb{R}^d$

- Minimizing $\mathcal{D}$ and $\mathcal{D}'$ not the same.



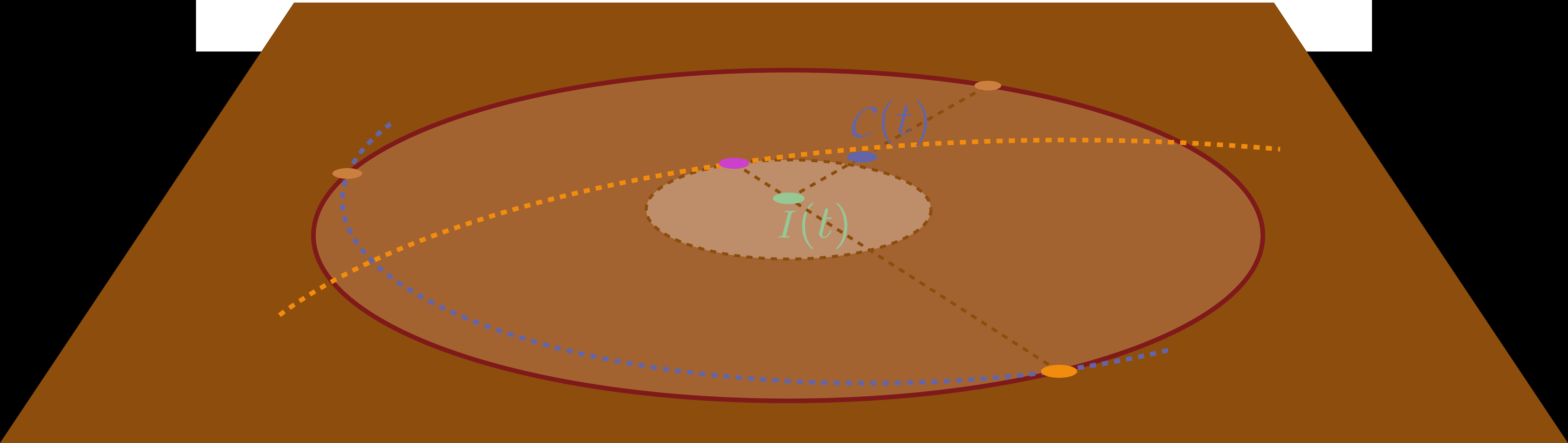
ENTITIES MOVING IN $\mathbb{R}^d$

- Minimizing $\mathcal{D}$ and $\mathcal{D}'$ not the same.
  - Use the distance $D(\sigma, \cdot)$ to the entity furthest from σ directly.



ENTITIES MOVING IN $\mathbb{R}^d$

- Minimizing $\mathcal{D}$ and $\mathcal{D}'$ not the same.
  - Use the distance $D(\sigma, \cdot)$ to the entity furthest from σ directly.
  - Same observations hold, and allow to show that $\mathcal{C}$ has complexity $O(\tau n^2 \sqrt{n})$.
  - Computing $\mathcal{C}$ takes $O(\tau n^3)$ time.

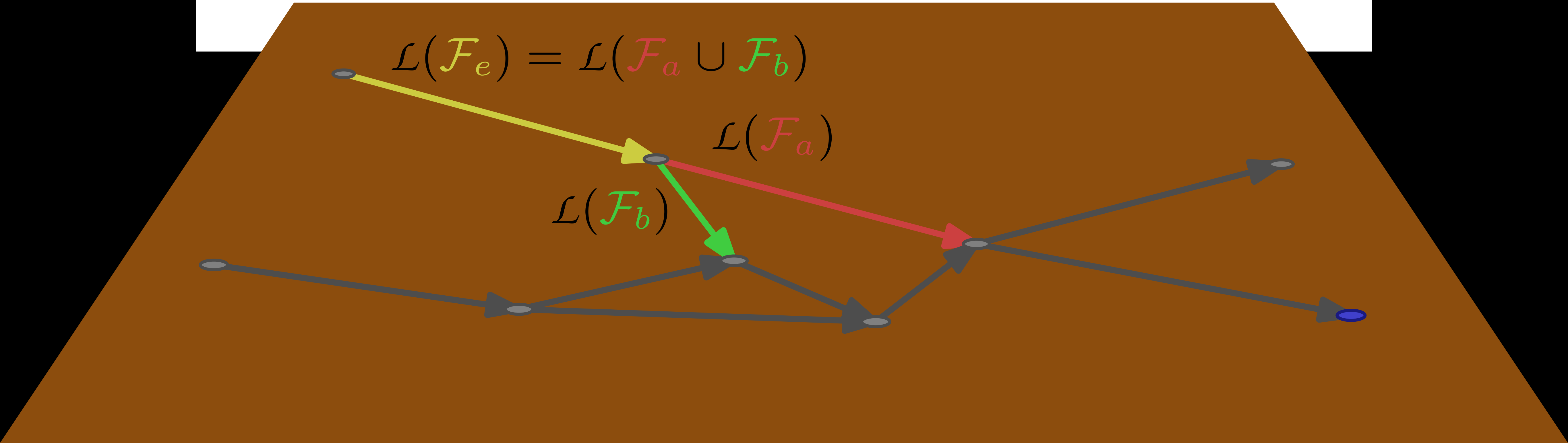


ENTITIES MOVING IN $\mathbb{R}^d$

OPEN PROBLEM.
How to compute lower envelopes for all edges of $\mathcal{R}$ efficiently?

- The weight of edge e is

$$\int \mathcal{L}(\{D(\sigma, \cdot) \mid \sigma \in \mathcal{X}_e\})(t)\, \mathrm{d}t$$



CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - “simple median” optimises homotopy area
  - acyclic trajectory intersection graph
  - open problem: find appropriate space

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - "simple median" optimises homotopy area
  - acyclic trajectory intersection graph
  - open problem: find appropriate space
- When time is relevant

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - "simple median" optimises homotopy area
  - acyclic trajectory intersection graph
  - open problem: find appropriate space
- When time is relevant
  - time dimension is special

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - “simple median” optimises homotopy area
  - acyclic trajectory intersection graph
  - open problem: find appropriate space
- When time is relevant
  - time dimension is special
  - based on any point set representative

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - “simple median” optimises homotopy area
  - acyclic trajectory intersection graph
  - open problem: find appropriate space
- When time is relevant
  - time dimension is special
  - based on any point set representative
  - polynomial time for many measures

CONCLUSIONS

- When time is not relevant
  - based on geometry and topology
  - "simple median" optimises homotopy area
  - acyclic trajectory intersection graph
  - open problem: find appropriate space
- When time is relevant
  - time dimension is special
  - based on any point set representative
  - polynomial time for many measures
  - open problem: improve running time

THANK YOU!