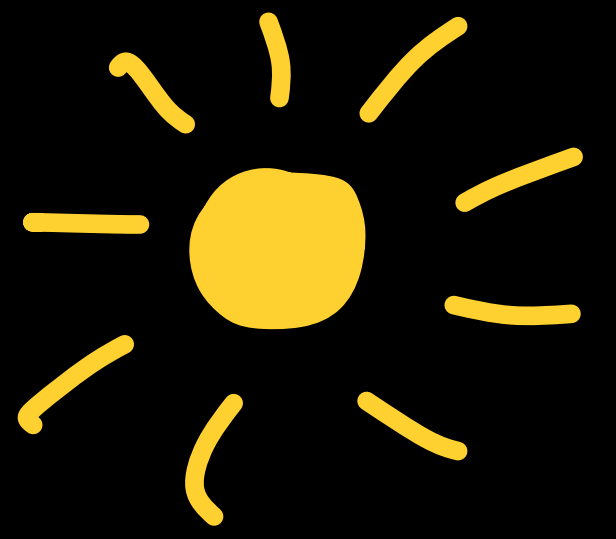


BEYOND SCHRÖDINGER BRIDGES for LEARNING TRAJECTORIES from SNAPSHOT SAMPLES



Renato Berlinghieri

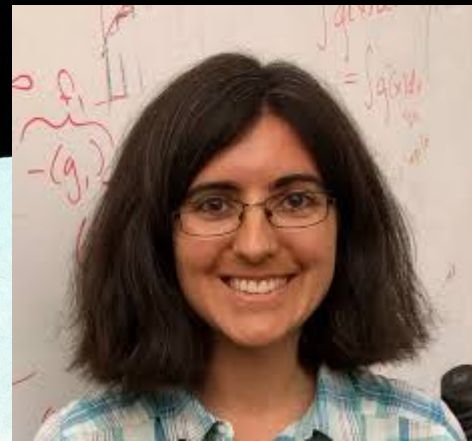
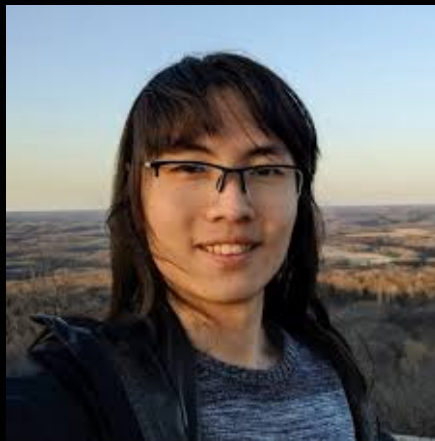
January 13th, 2026

ICERM



BEYOND SCHRÖDINGER BRIDGES for

LEARNING TRAJECTORIES from SNAPSHOT SAMPLES



Renato Berlinghieri

January 13th, 2026

ICERM

joint work with Yunyi
Shen & Tamara Broderick

OUR SCIENTIFIC PROBLEM: Learning Dynamics from Snapshots

- Motivation: model gene expression of a cell over its life cycle

OUR SCIENTIFIC PROBLEM: Learning Dynamics from Snapshots

- Motivation: model gene expression of a cell over its life cycle
- Problem: cells killed to measure gene expression → only observe snapshot samples (from many cells), never trajectories

OUR SCIENTIFIC PROBLEM: Learning Dynamics from Snapshots

- Motivation: model gene expression of a cell over its life cycle
- Problem: cells killed to measure gene expression $\rightarrow$ only observe snapshot samples (from many cells), never trajectories
- Idea: model gene expression using Stochastic Differential Equations (SDEs)
 $\rightarrow$ But cannot measure gene expression in continuous time...

OUR SCIENTIFIC PROBLEM: Learning Dynamics from Snapshots

- Motivation: model gene expression of a cell over its life cycle
- Problem: cells killed to measure gene expression $\rightarrow$ only observe snapshot samples (from many cells), never trajectories
- Idea: model gene expression using Stochastic Differential Equations (SDEs)
 $\rightarrow$ But cannot measure gene expression in continuous time...
- Better idea: Schrödinger Bridges (SBs). These either
 - (1) interpolate between just two time points, or
 - (2) require single fixed reference

OUR SCIENTIFIC PROBLEM: Learning Dynamics from Snapshots

- Motivation: model gene expression of a cell over its life cycle
- Problem: cells killed to measure gene expression $\rightarrow$ only observe snapshot samples (from many cells), never trajectories
- Idea: model gene expression using Stochastic Differential Equations (SDEs)
 $\rightarrow$ But cannot measure gene expression in continuous time...
- Better idea: Schrödinger Bridges (SBs). These either
 - (1) interpolate between just two time points, or
 - (2) require single fixed reference $\rightarrow$ But unrealistic for biologists to know exact reference

OUR SCIENTIFIC PROBLEM: Learning Dynamics from Snapshots

- Motivation: model gene expression of a cell over its life cycle
- Problem: cells killed to measure gene expression $\rightarrow$ only observe snapshot samples (from many cells), never trajectories
- Idea: model gene expression using Stochastic Differential Equations (SDEs)
 $\rightarrow$ But cannot measure gene expression in continuous time...
- Better idea: Schrödinger Bridges (SBs). These either
 - (1) interpolate between just two time points, or
 - (2) require single fixed reference $\rightarrow$ But unrealistic for biologists to know exact reference
- Our solution: Multimarginal Schrödinger Bridges + Reference Family Refinement

OUTLINE

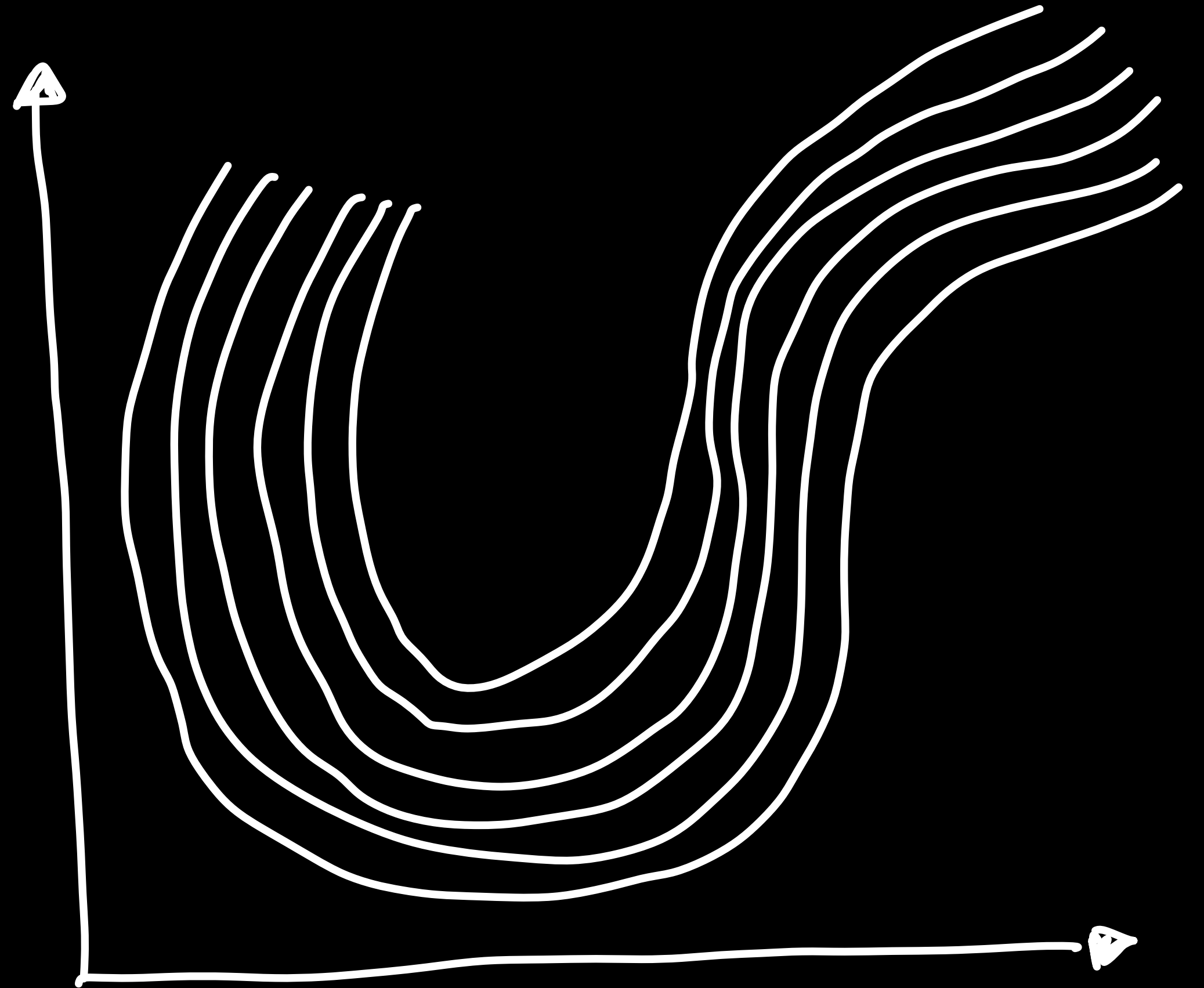
- Problem Setup using single cell example
- Background: SDEs & Schrödinger Bridges
- SBIRR : Multimarginal SB with iterative reference refinement (AISTATS 2025, oral)
- SnapMMD : Forecasting Stochastic Dynamics using MMD (Submitted)
- Summary & Future Work

OUTLINE

- Problem Setup using single cell example
- Background: SDEs & Schrödinger Bridges
- SBIRR : Multimarginal SB with iterative reference refinement (AISTATS 2025, oral)
- SnapMMD : Forecasting Stochastic Dynamics using MMD (Submitted)
- Summary & Future Work

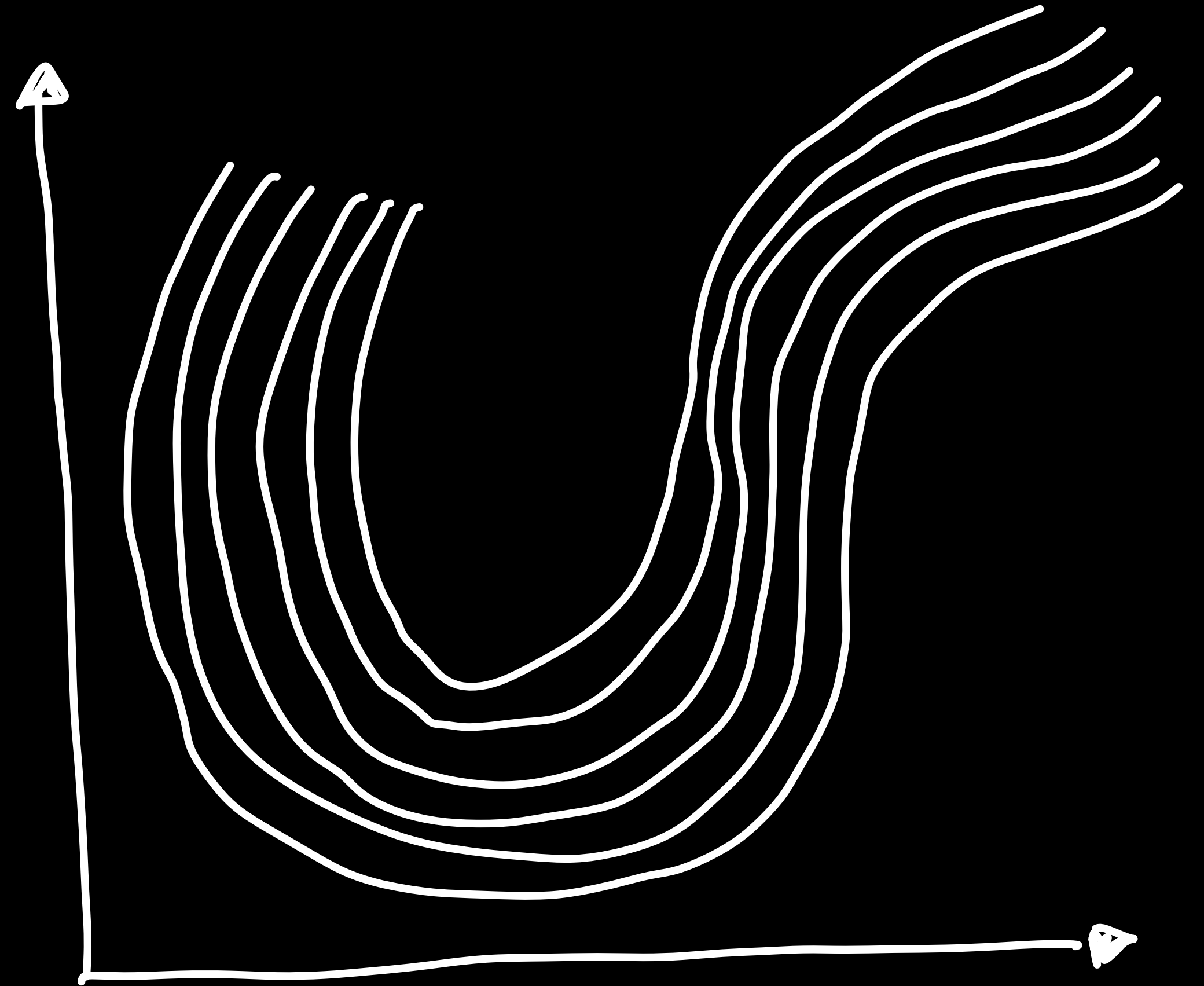
PROBLEM SETUP

- Each cell has a trajectory of gene expression



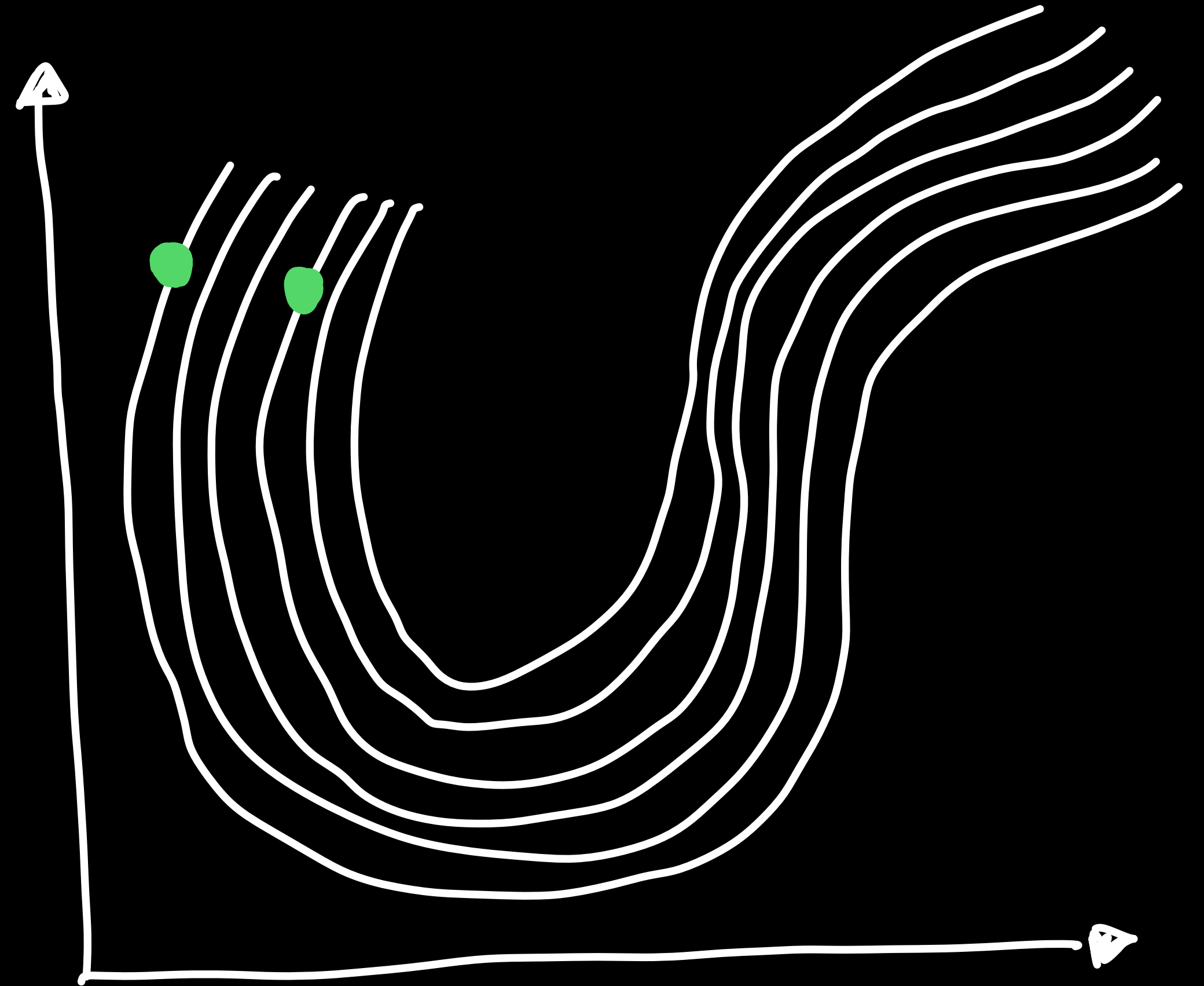
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot



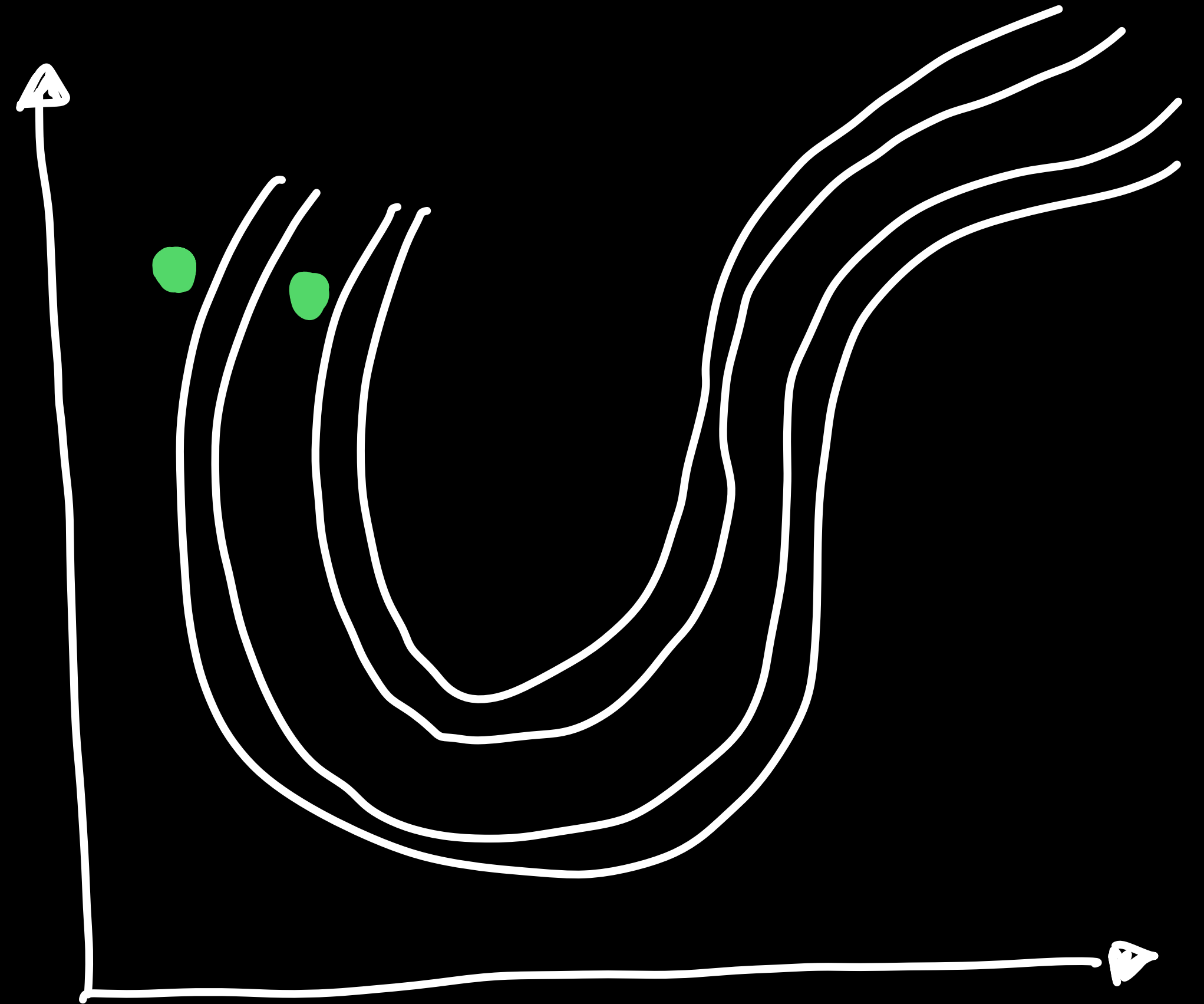
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot



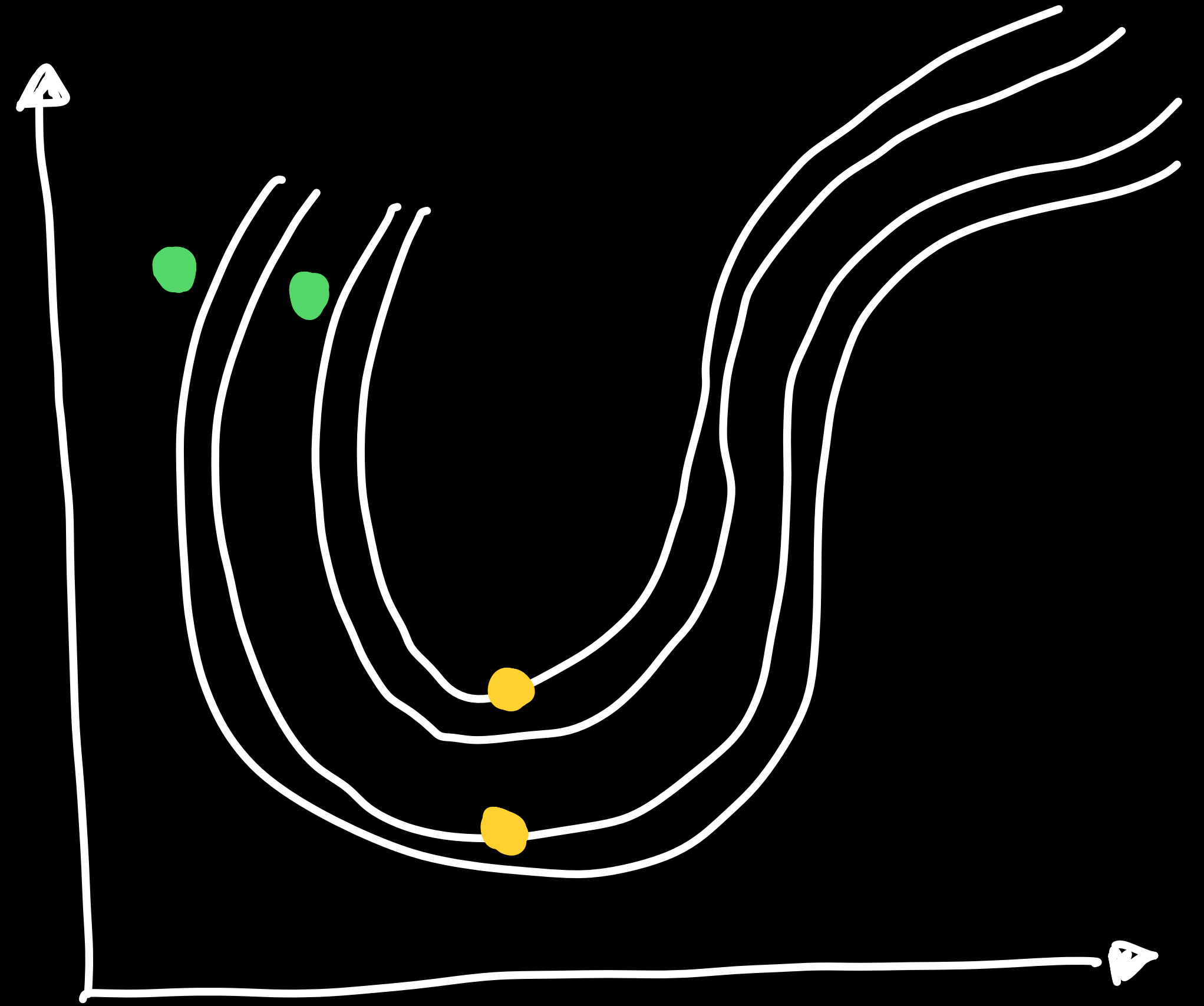
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot



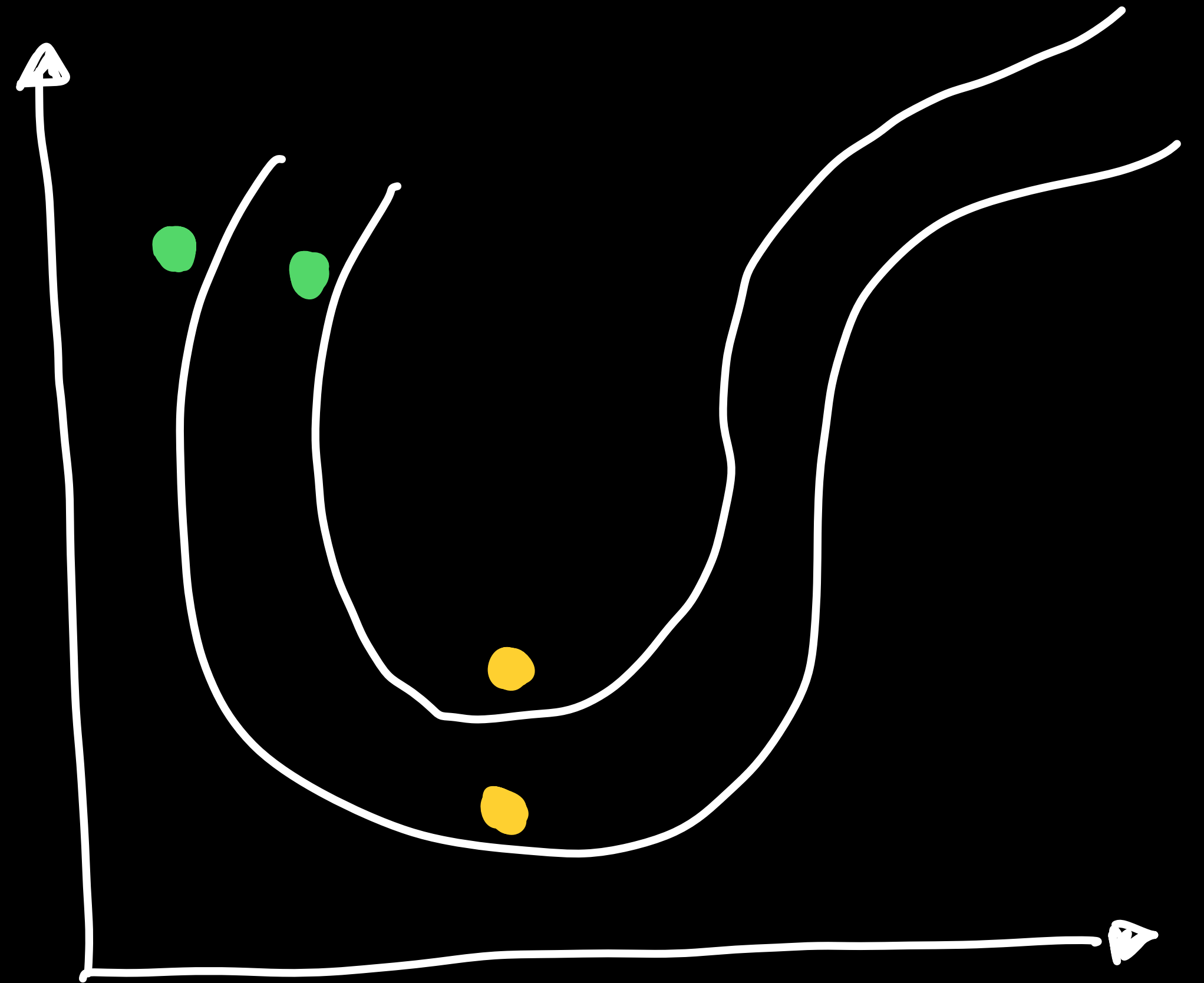
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot



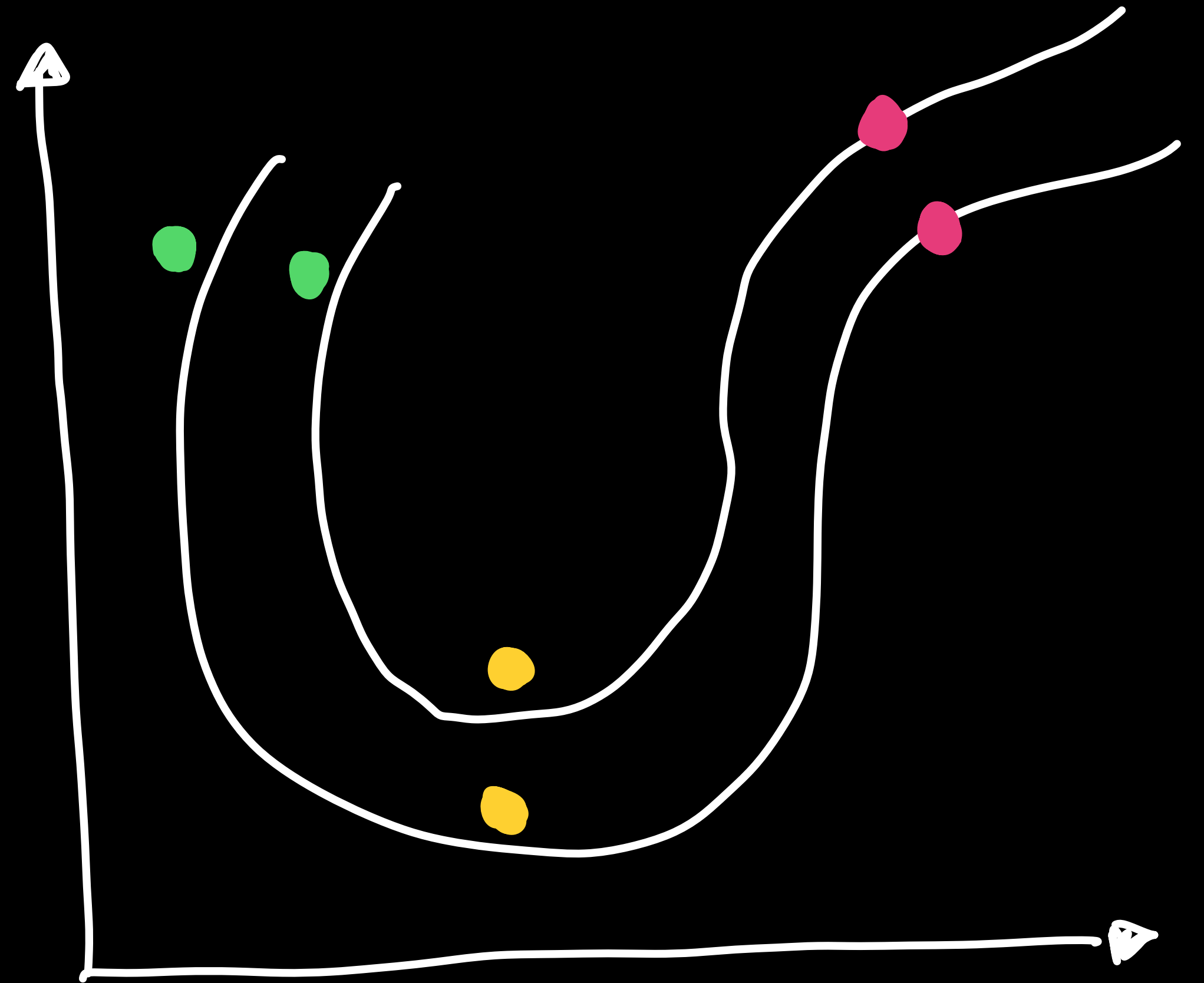
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot



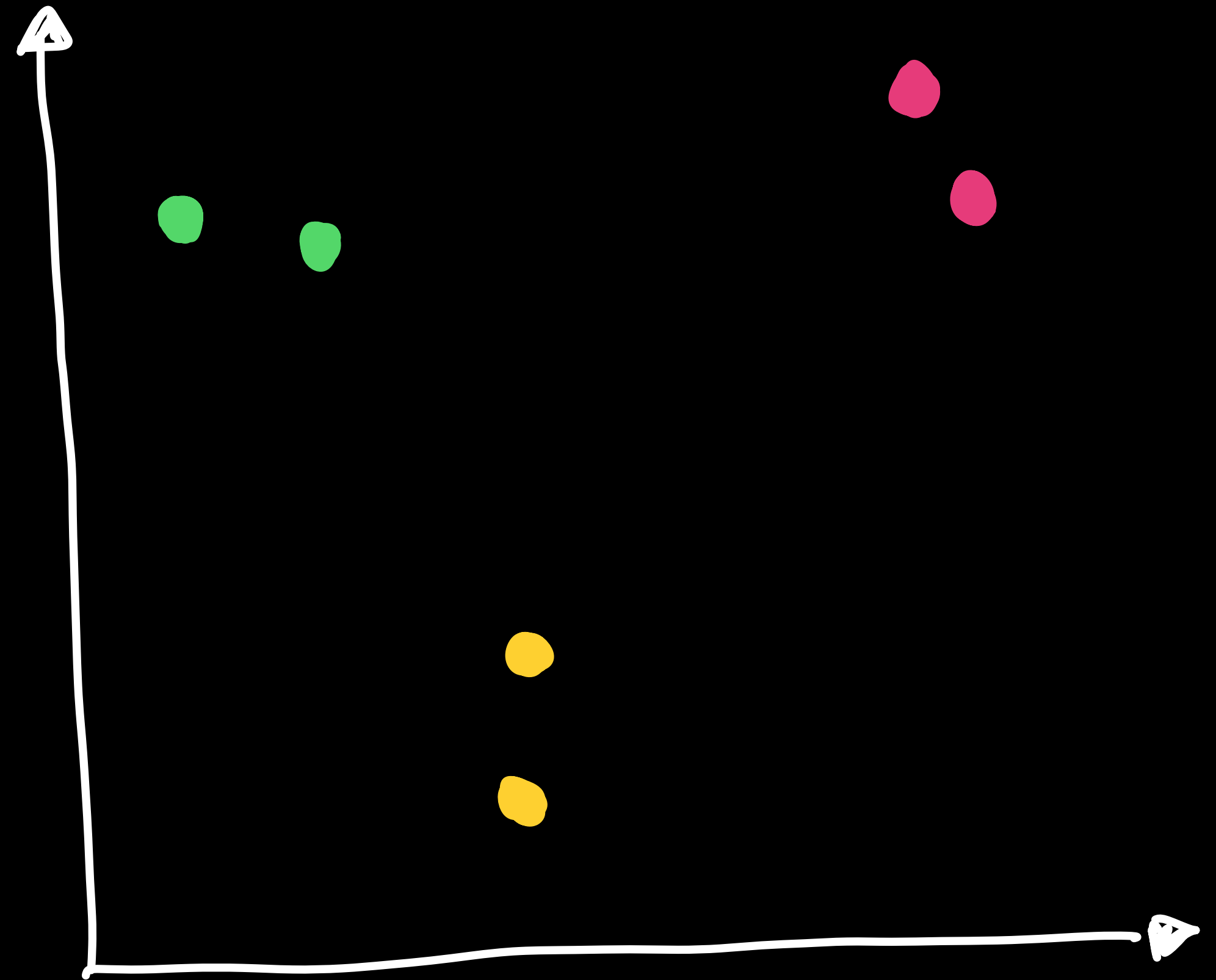
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot



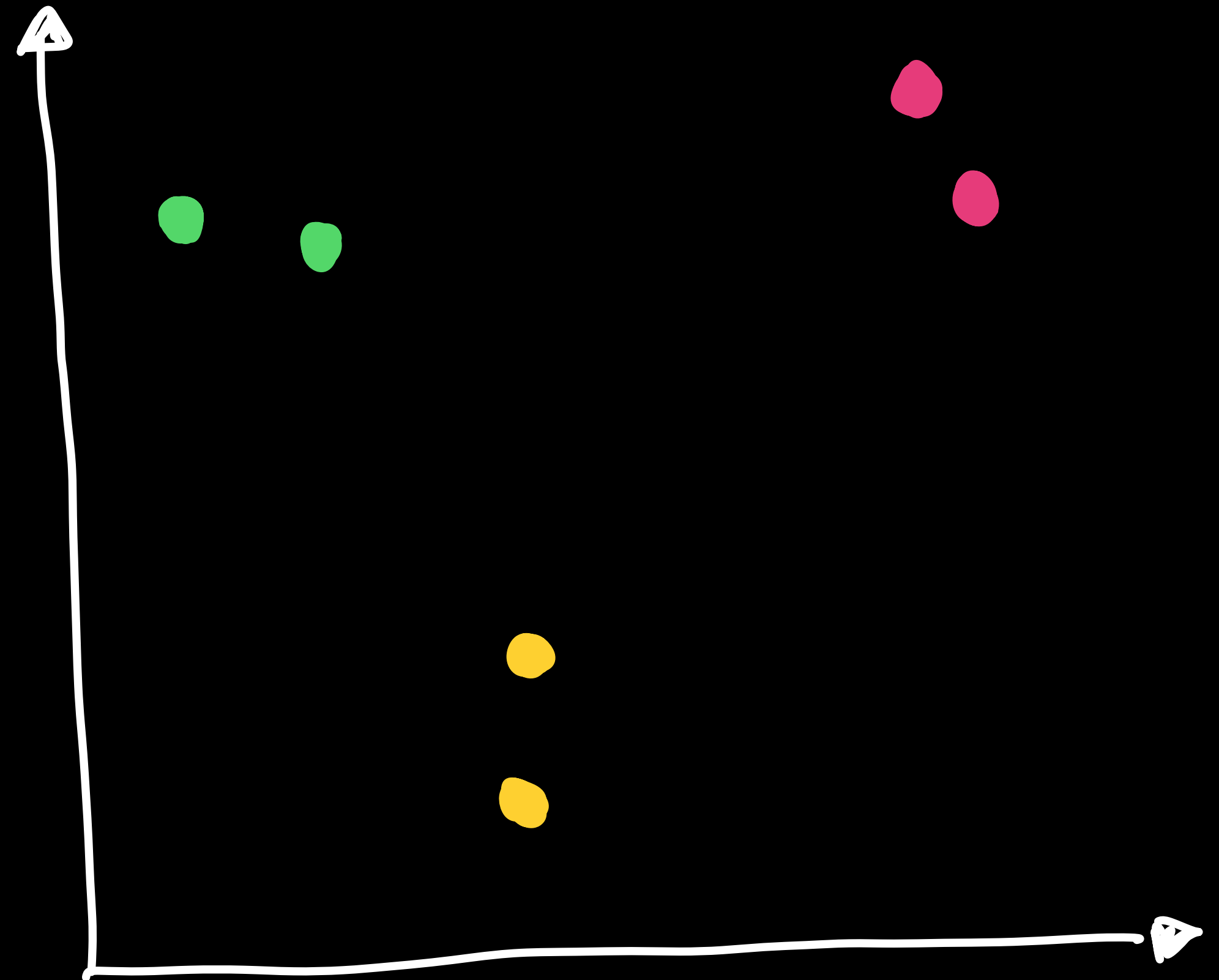
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot



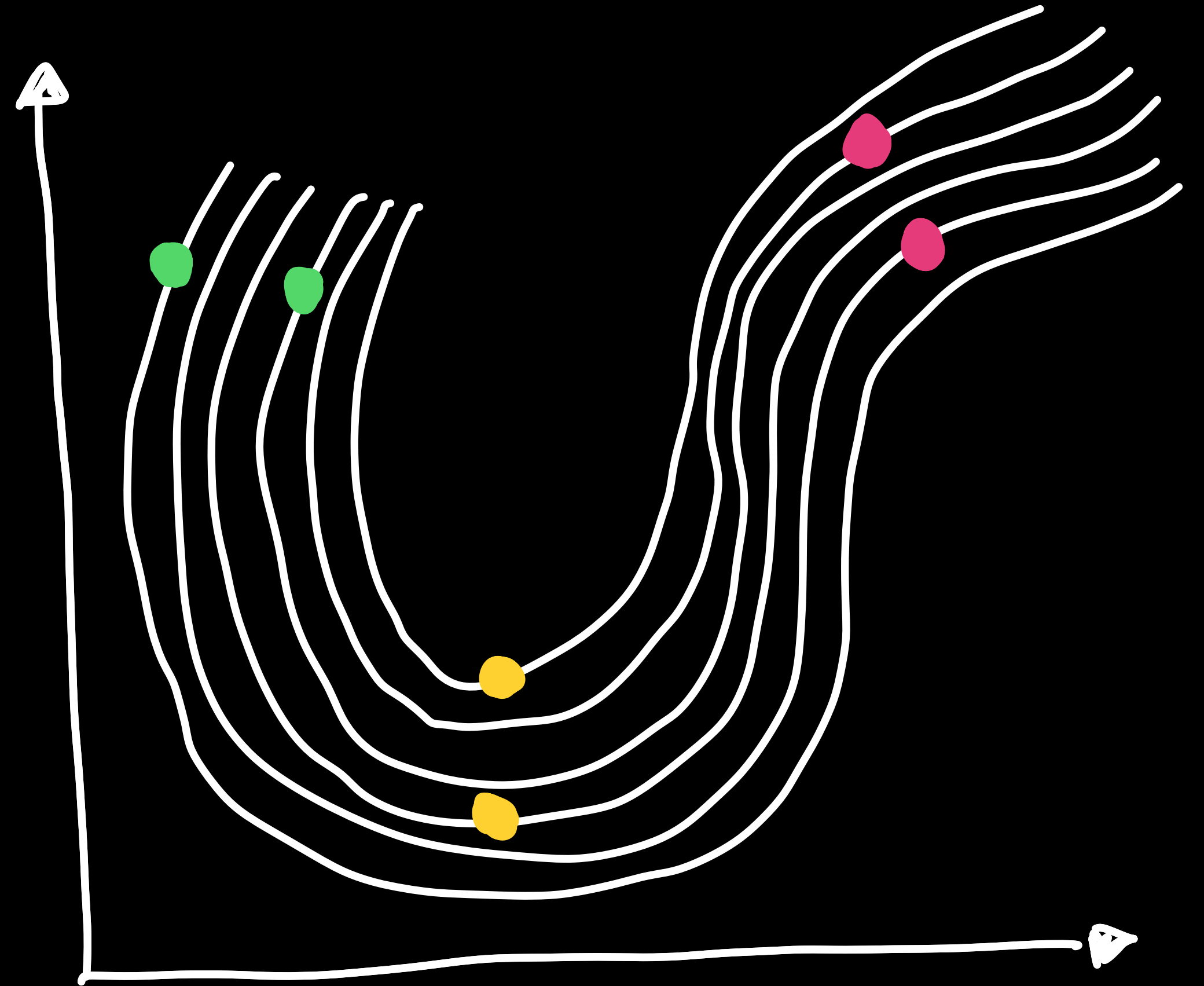
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot
- Goal: to infer/sample the unobserved trajectories from sample snapshots



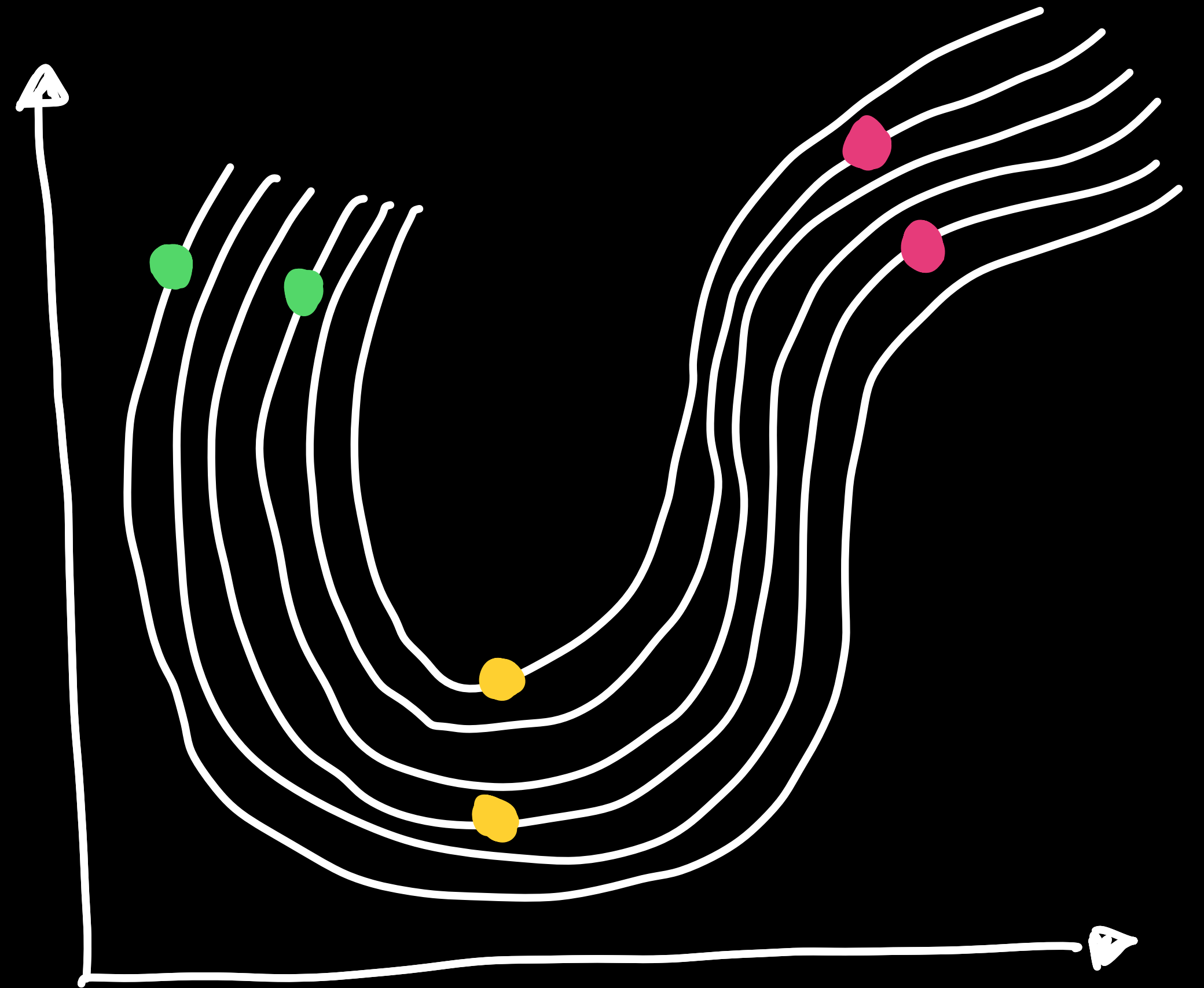
PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot
- Goal: to infer/sample the unobserved trajectories from sample snapshots



PROBLEM SETUP

- Each cell has a trajectory of gene expression
- We Kill the cell when we see the trajectory at a snapshot
- Goal: to infer/sample the unobserved trajectories from sample snapshots
- Note: this setup comes up in a variety of situations (not specific to gene expression)



OUTLINE

- Problem Setup using single cell example
- Background: SDEs & Schrödinger Bridges
- SBIRR : Multimarginal SB with iterative reference refinement (AISTATS 2025, oral)
- SnapMMD : Forecasting Stochastic Dynamics using MMD (Submitted)
- Summary & Future Work

STOCHASTIC DIFFERENTIAL EQUATIONS REVIEW

- We model evolution of dynamical systems (e.g. mRNA levels in a differentiating cell) using SDEs

STOCHASTIC DIFFERENTIAL EQUATIONS REVIEW

- We model evolution of dynamical systems (e.g. mRNA levels in a differentiating cell) using SDEs

$$dX_t = b(x_t, t)dt + g(x_t, t)dW_t$$

STOCHASTIC DIFFERENTIAL EQUATIONS REVIEW

- We model evolution of dynamical systems (e.g. mRNA levels in a differentiating cell) using SDEs

$$dX_t = b(X_t, t)dt + g(X_t, t)dW_t$$

mRNA levels
evolution 

STOCHASTIC DIFFERENTIAL EQUATIONS REVIEW

- We model evolution of dynamical systems (e.g. mRNA levels in a differentiating cell) using SDEs

$$dX_t = b(x_t, t) dt + g(x_t, t) dW_t$$

mRNA levels
evolution ↗

↑
drift: systematic
tendency of system
(e.g. average production/
decay forces changing
gene expression)

STOCHASTIC DIFFERENTIAL EQUATIONS REVIEW

- We model evolution of dynamical systems (e.g. mRNA levels in a differentiating cell) using SDEs

$$dX_t = b(x_t, t) dt + g(x_t, t) dW_t$$

mRNA levels
evolution

↑
drift: systematic
tendency of system
(e.g. average production/
decay forces changing
gene expression)

↑
volatility: variability
in how system evolves
(e.g. cell-to-cell
heterogeneity)

STOCHASTIC DIFFERENTIAL EQUATIONS REVIEW

- We model evolution of dynamical systems (e.g. mRNA levels in a differentiating cell) using SDEs

$$dX_t = b(X_t, t) dt + g(X_t, t) dW_t$$

mRNA levels
evolution

↑
drift: systematic
tendency of system
(e.g. average production/
decay forces changing
gene expression)

↑
volatility: variability
in how system evolves
(e.g. cell-to-cell
heterogeneity)

← Brownian motion:
source of stochasticity

STOCHASTIC DIFFERENTIAL EQUATIONS REVIEW

- We model evolution of dynamical systems (e.g. mRNA levels in a differentiating cell) using SDEs

$$dX_t = b(X_t, t) dt + g(X_t, t) dW_t$$

mRNA levels
evolution

drift: systematic
tendency of system
(e.g. average production/
decay forces changing
gene expression)

volatility: variability
in how system evolves
(e.g. cell-to-cell
heterogeneity)

Brownian motion:
source of stochasticity

- An SDE defines a probability distribution over trajectories ("path measure")

STOCHASTIC DIFFERENTIAL EQUATIONS REVIEW

- We model evolution of dynamical systems (e.g. mRNA levels in a differentiating cell) using SDEs

$$dX_t = b(X_t, t) dt + g(X_t, t) dW_t$$

mRNA levels
evolution

↑
drift: systematic
tendency of system
(e.g. average production/
decay forces changing
gene expression)

↑
volatility: variability
in how system evolves
(e.g. cell-to-cell
heterogeneity)

← Brownian motion:
source of stochasticity

- An SDE defines a probability distribution over trajectories ("path measure")
 - one way to compare trajectories generated by two SDEs is by computing a divergence between their induced path measures

SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots

SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
→ minimize KL divergence to a single reference dynamic

SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
→ minimize KL divergence to a single reference dynamic

●
single
reference

SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
→ minimize KL divergence to a single reference dynamic

●
single
reference

$$\min_Q \text{KL} (Q \parallel \text{Path}(\text{reference}))$$

SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
 - minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_Q \text{KL} (Q \parallel \text{Path}(\text{reference}))$$

●
single
reference

SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
→ minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_Q \text{KL} (Q \parallel \text{Path}(\text{reference}))$$

●
single
reference

Family
of dynamics
interpolating
snapshots

SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
→ minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_{Q: X_0 \sim \mu_0, X_1 \sim \mu_1} KL(Q \parallel \text{Path}(\text{reference}))$$

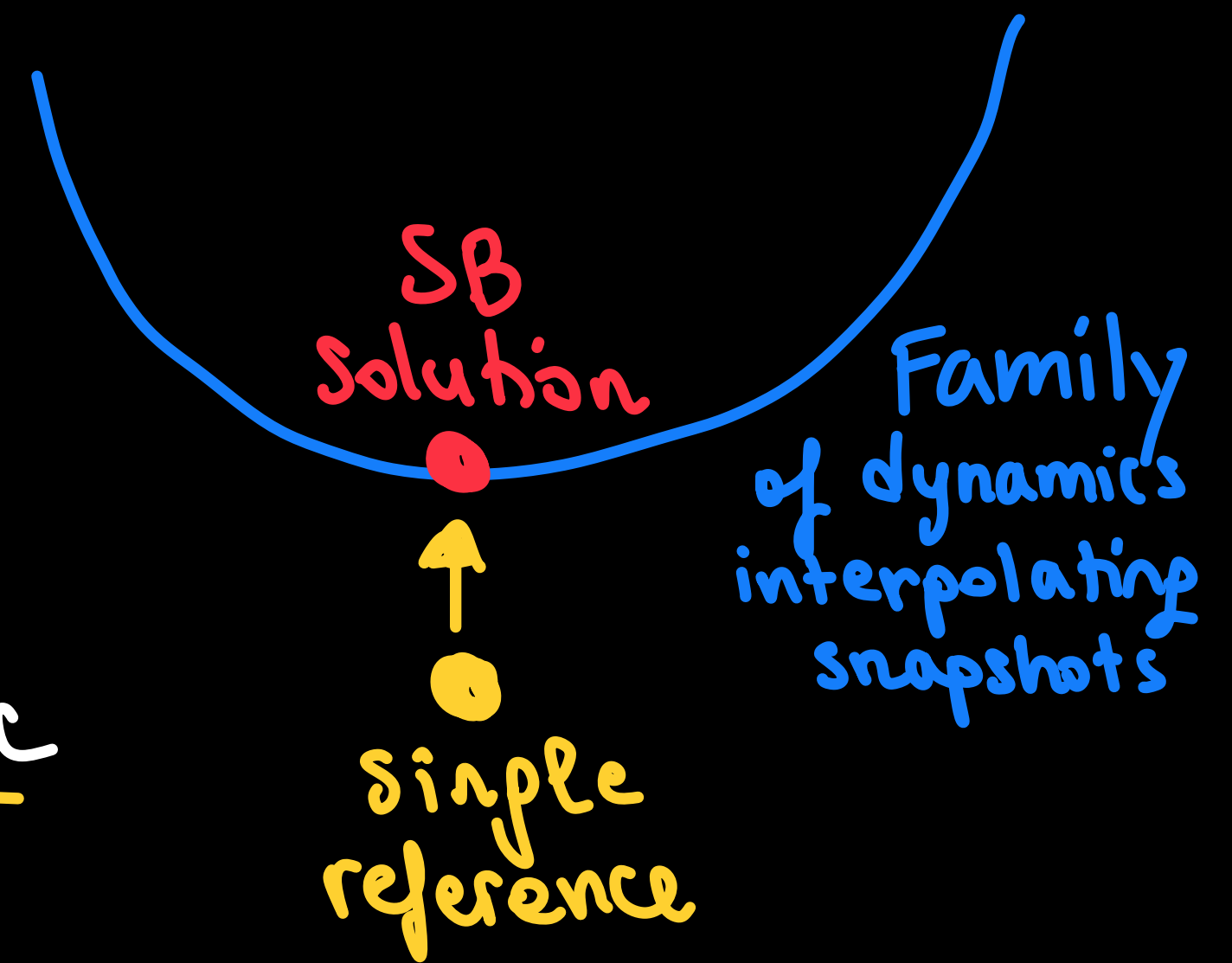
●
single
reference

Family
of dynamics
interpolating
snapshots

SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
 - minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_{Q: X_0 \sim \mu_0, X_1 \sim \mu_1} KL(Q \parallel \text{Path}(\text{reference}))$$

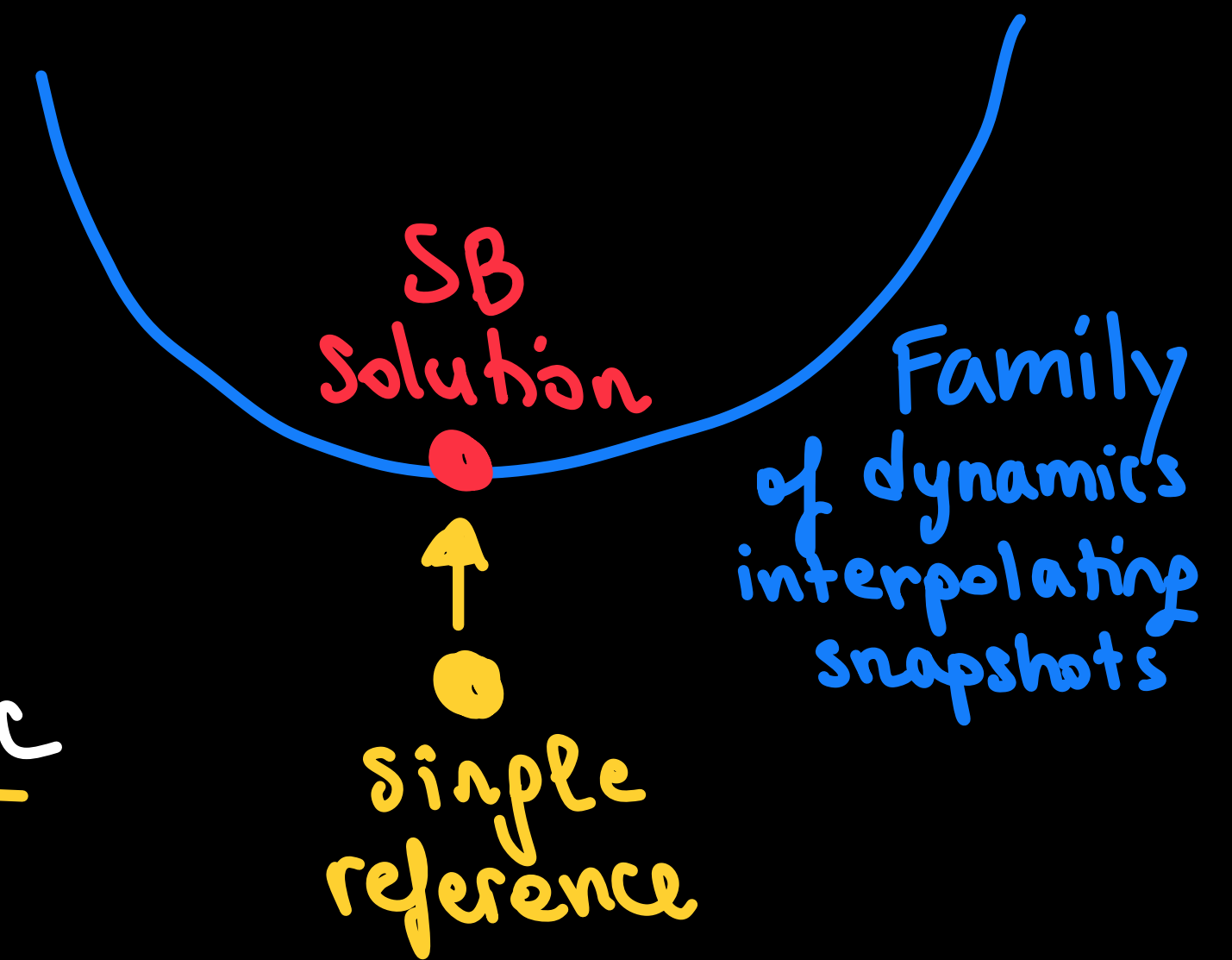


SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
→ minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_{Q: X_0 \sim \mu_0, X_1 \sim \mu_1} KL(Q \parallel \text{Path}(\text{reference}))$$

- Issues under this setup:

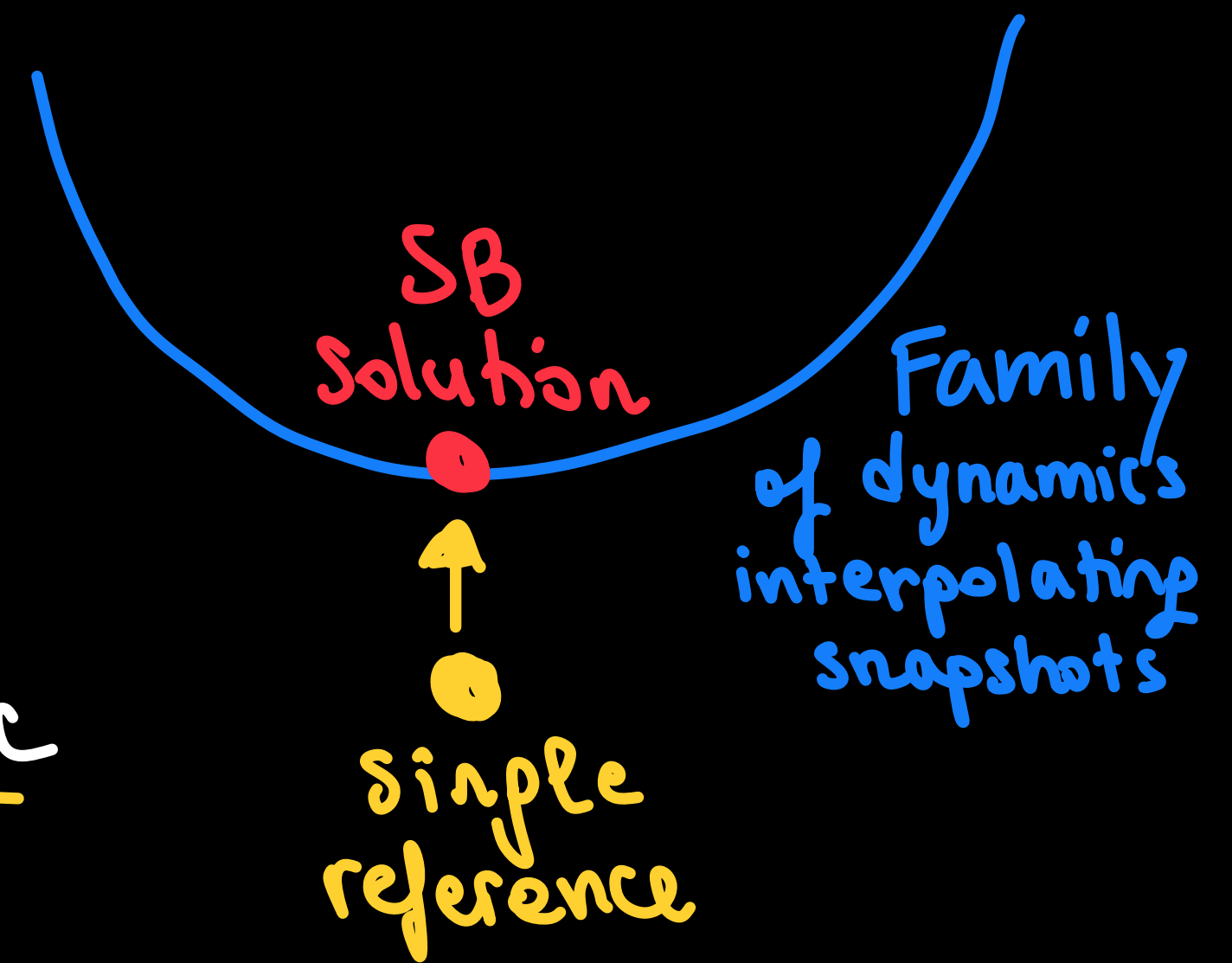


SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
 - minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_{Q: X_0 \sim \mu_0, X_1 \sim \mu_1} KL(Q \parallel \text{Path}(\text{reference}))$$

- Issues under this setup:
 - multimarginal SB does piecewise interpolation (Lavenant et al. 2024)

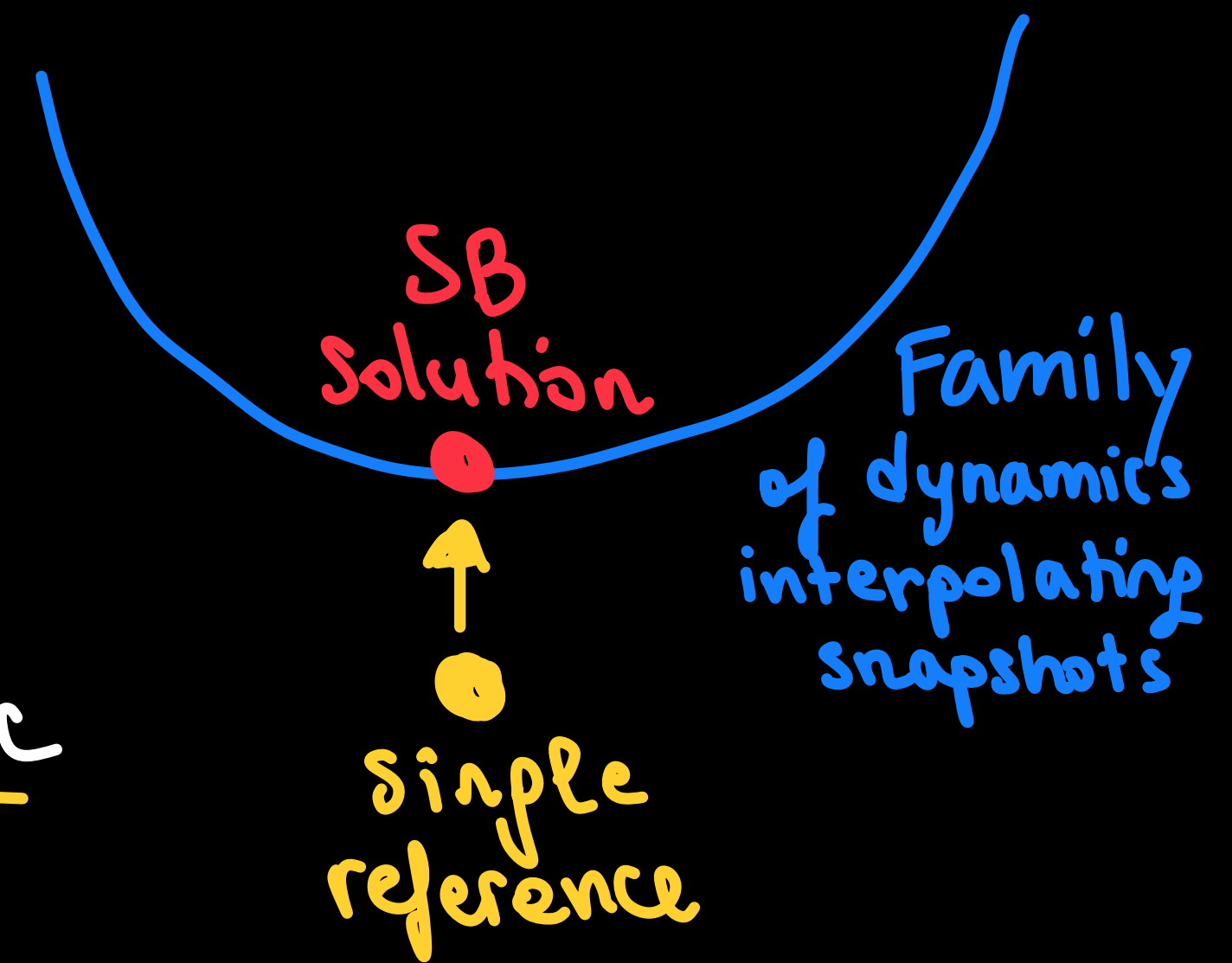


SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
 - minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_{Q: X_0 \sim \mu_0, X_1 \sim \mu_1} KL(Q \parallel \text{Path}(\text{reference}))$$

- Issues under this setup:
 - multimarginal SB does piecewise interpolation (Lavenant et al. 2024)
 - requires a single fixed reference dynamic with all parameters known, often Brownian motion

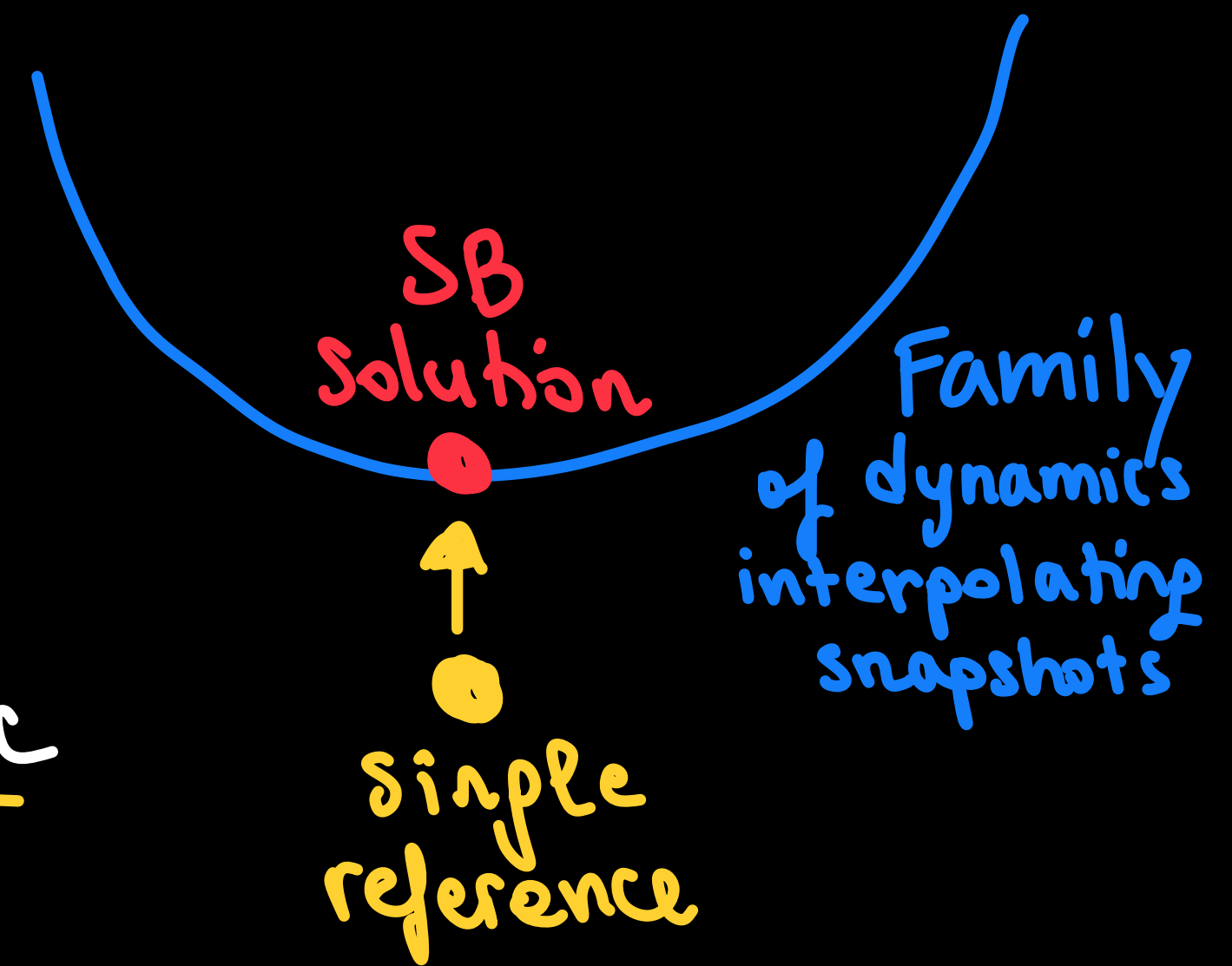


SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
 - minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_{Q: X_0 \sim \mu_0, X_1 \sim \mu_1} KL(Q \parallel \text{Path}(\text{reference}))$$

- Issues under this setup:
 - multimarginal SB does piecewise interpolation (Lavenant et al. 2024)
 - requires a single fixed reference dynamic with all parameters known, often Brownian motion
 - Brownian motion reference makes sense for thermodynamics, not always for biology

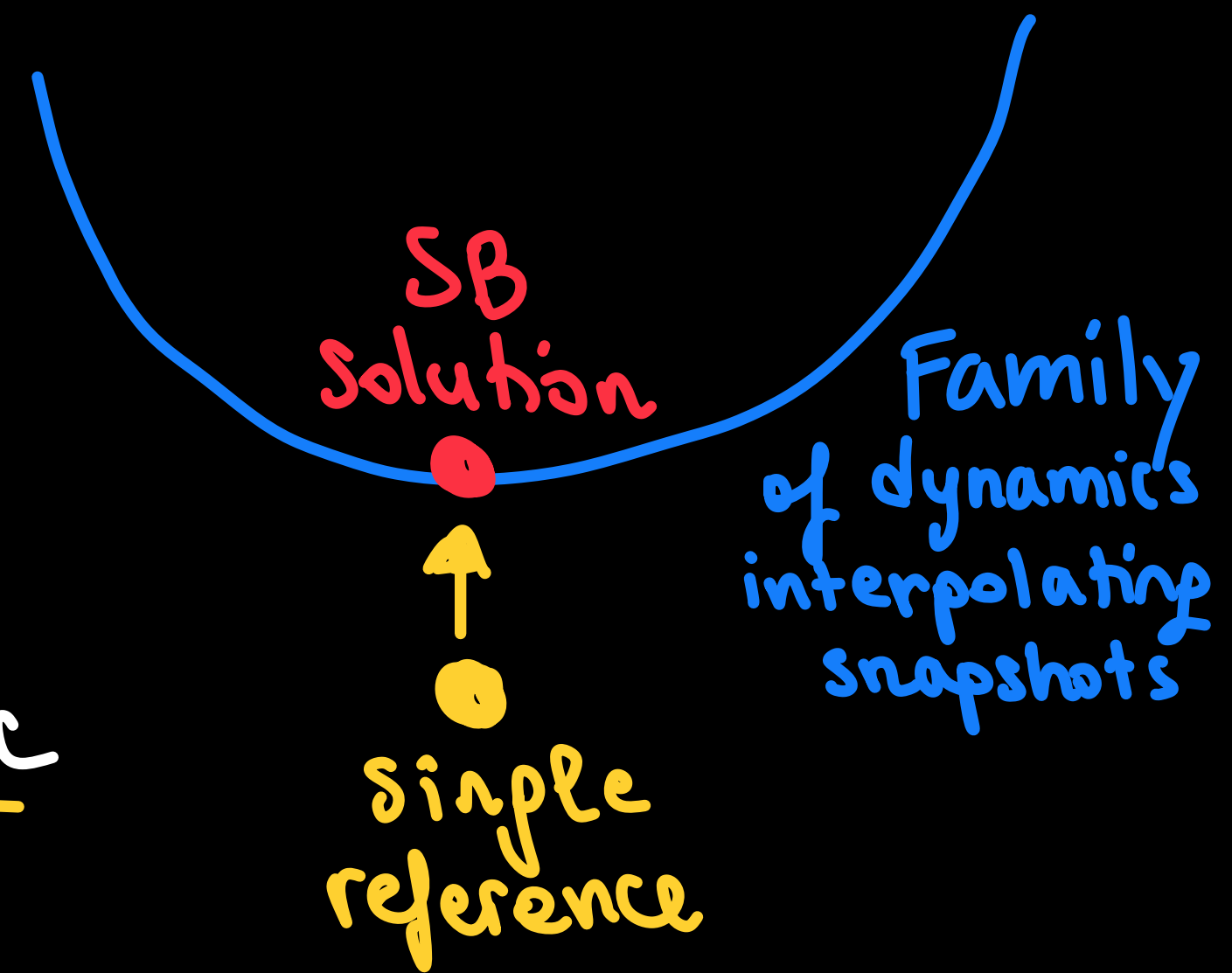


SCHRÖDINGER BRIDGES REVIEW

- Schrödinger Bridges (SBs) provide trajectory samples between two time snapshots
 - minimize KL divergence to a single reference dynamic & constraint to interpolate marginals

$$\min_{Q: X_0 \sim \mu_0, X_1 \sim \mu_1} KL(Q \parallel \text{Path}(\text{reference}))$$

- Issues under this setup:
 - multimarginal SB does piecewise interpolation (Lavenant et al. 2024)
 - requires a single fixed reference dynamic with all parameters known, often Brownian motion
 - Brownian motion reference makes sense for thermodynamics, not always for biology



"What organisms feed upon is negative entropy"

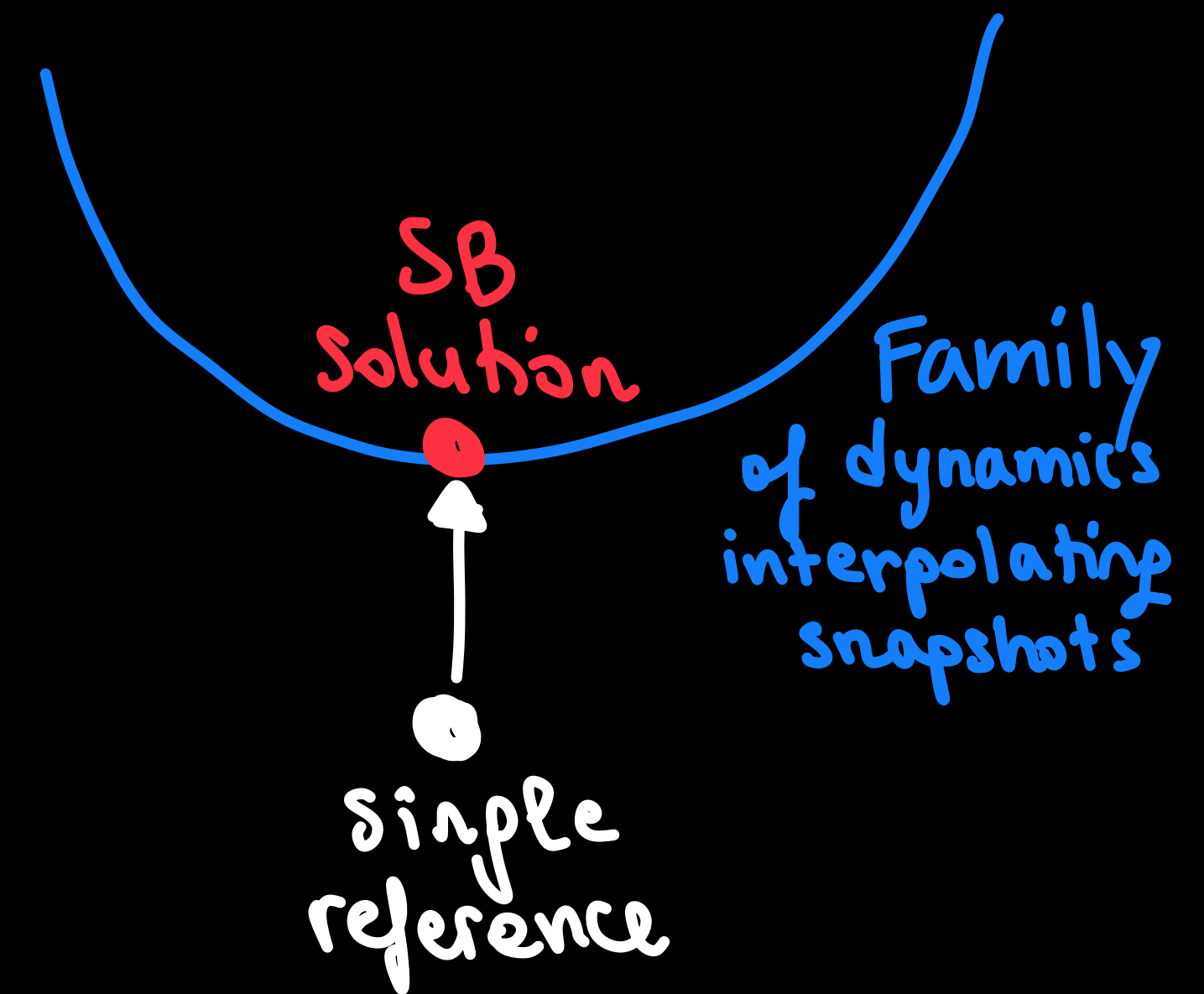


OUTLINE

- Problem Setup using single cell example
- Background: SDEs & Schrödinger Bridges
- SBIRR : Multimarginal SB with iterative reference refinement (AISTATS 2025, oral)
- SnapMMD : Forecasting Stochastic Dynamics using MMD (Submitted)
- Summary & Future Work

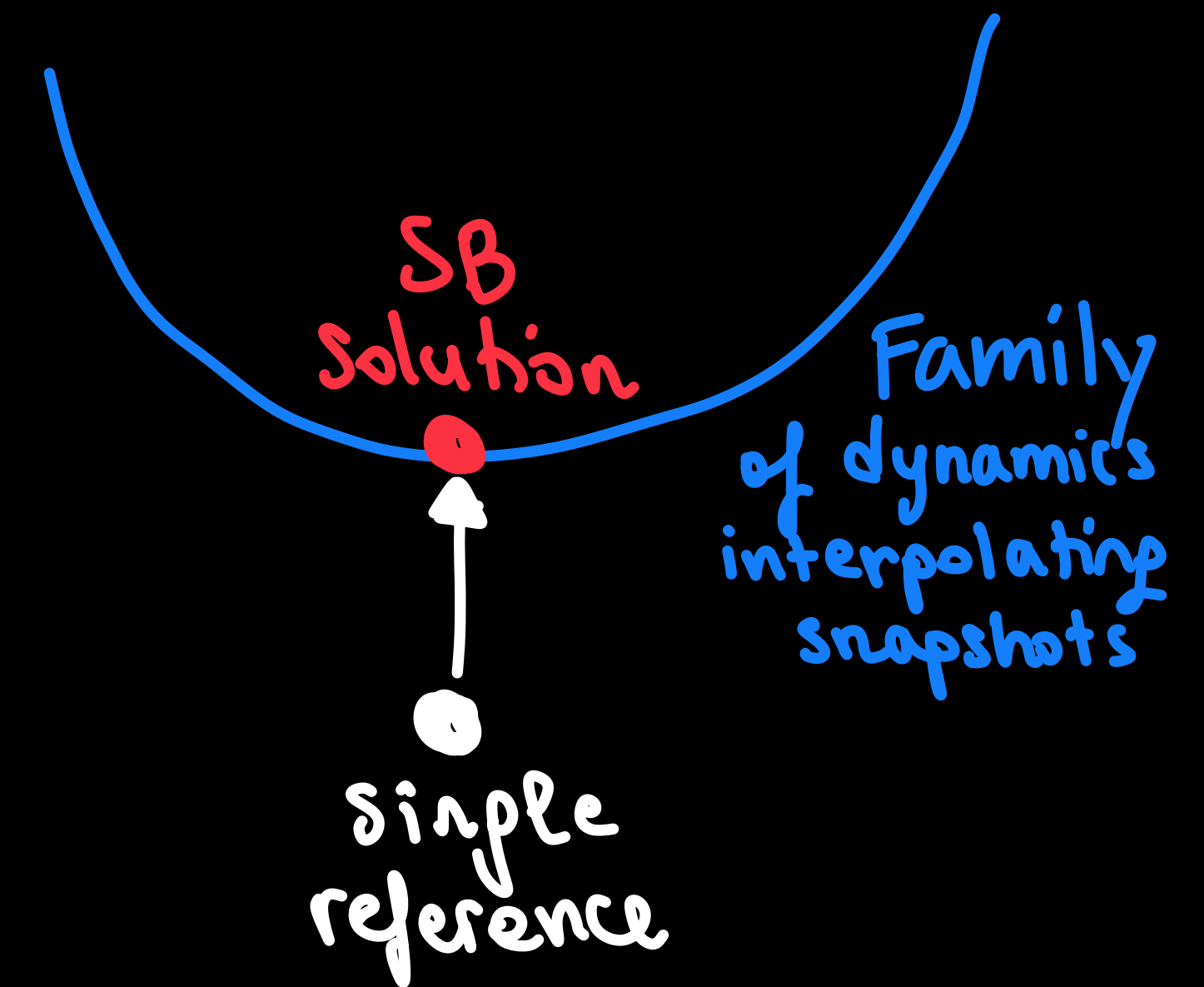
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference



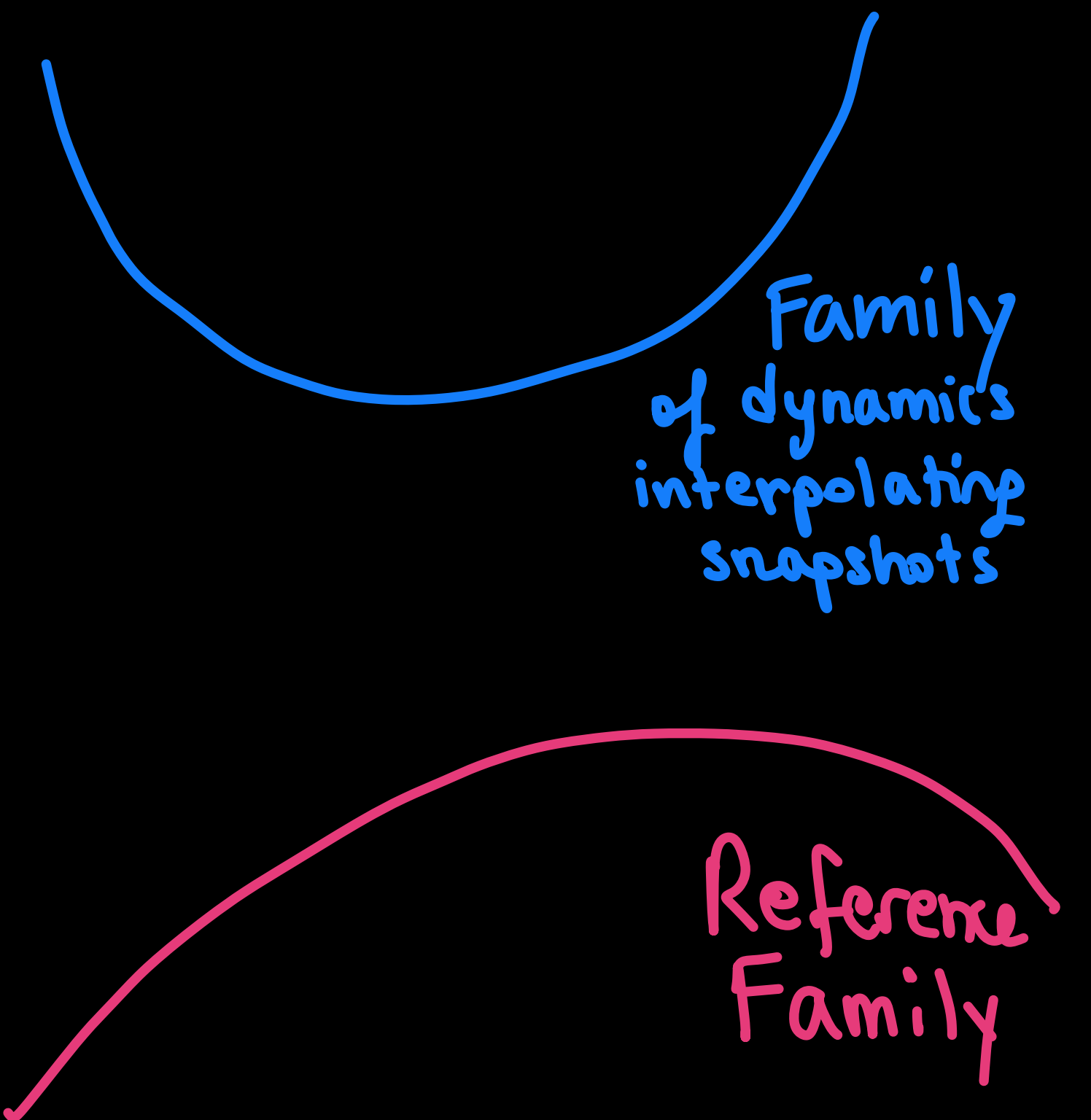
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
- When we have a family of references,



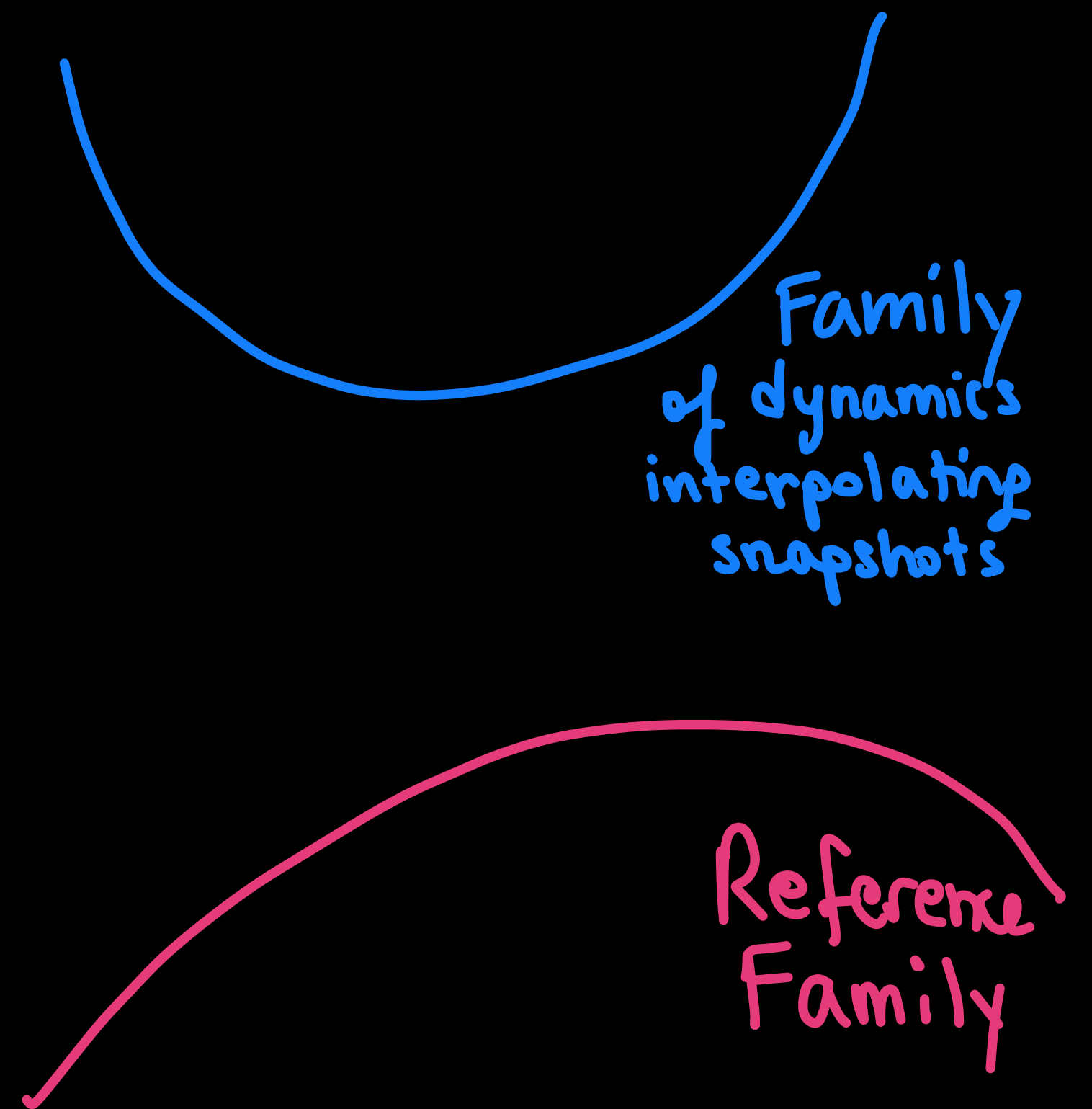
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
- When we have a family of references,



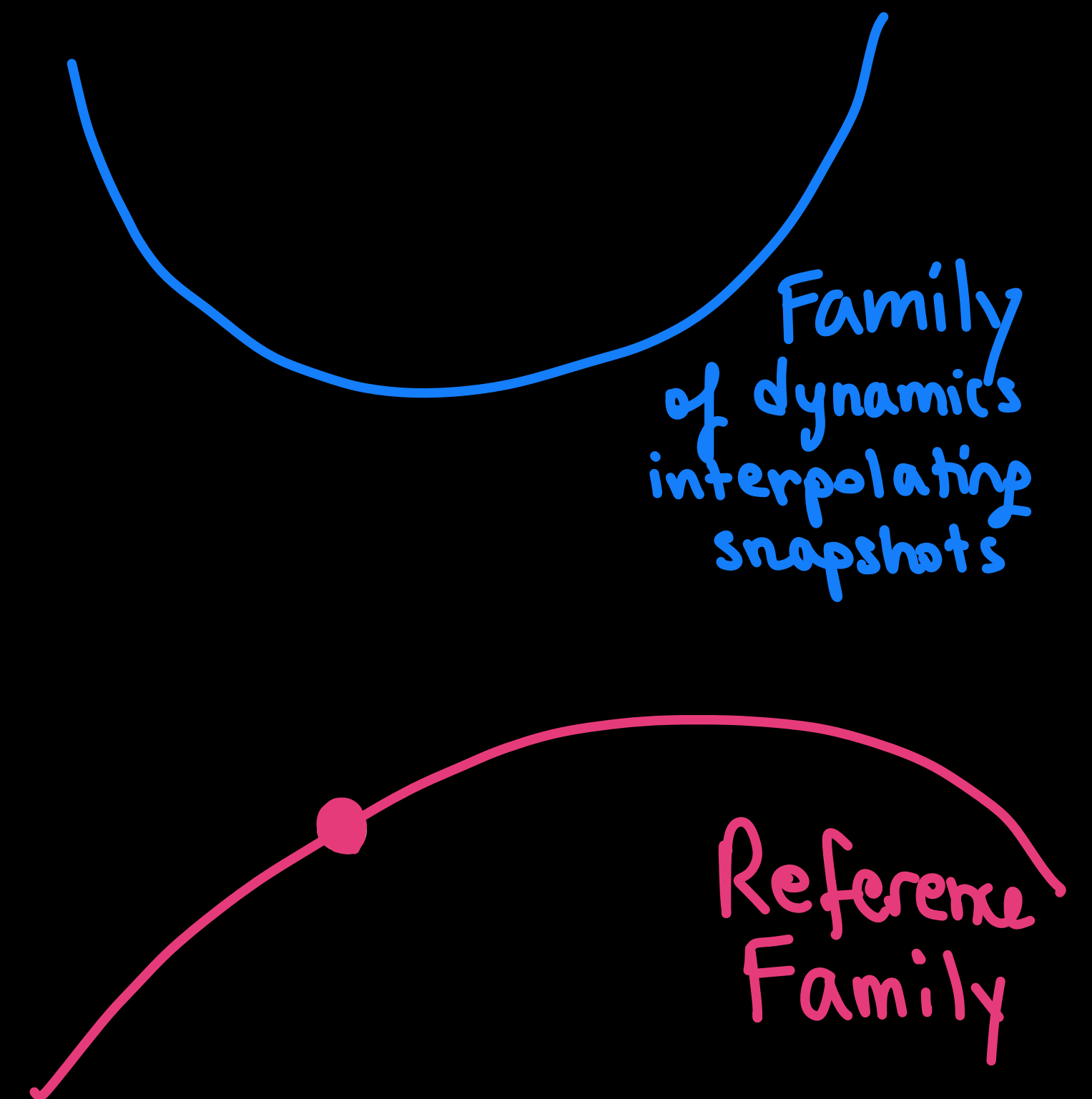
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
- When we have a family of references, we iteratively refine it!



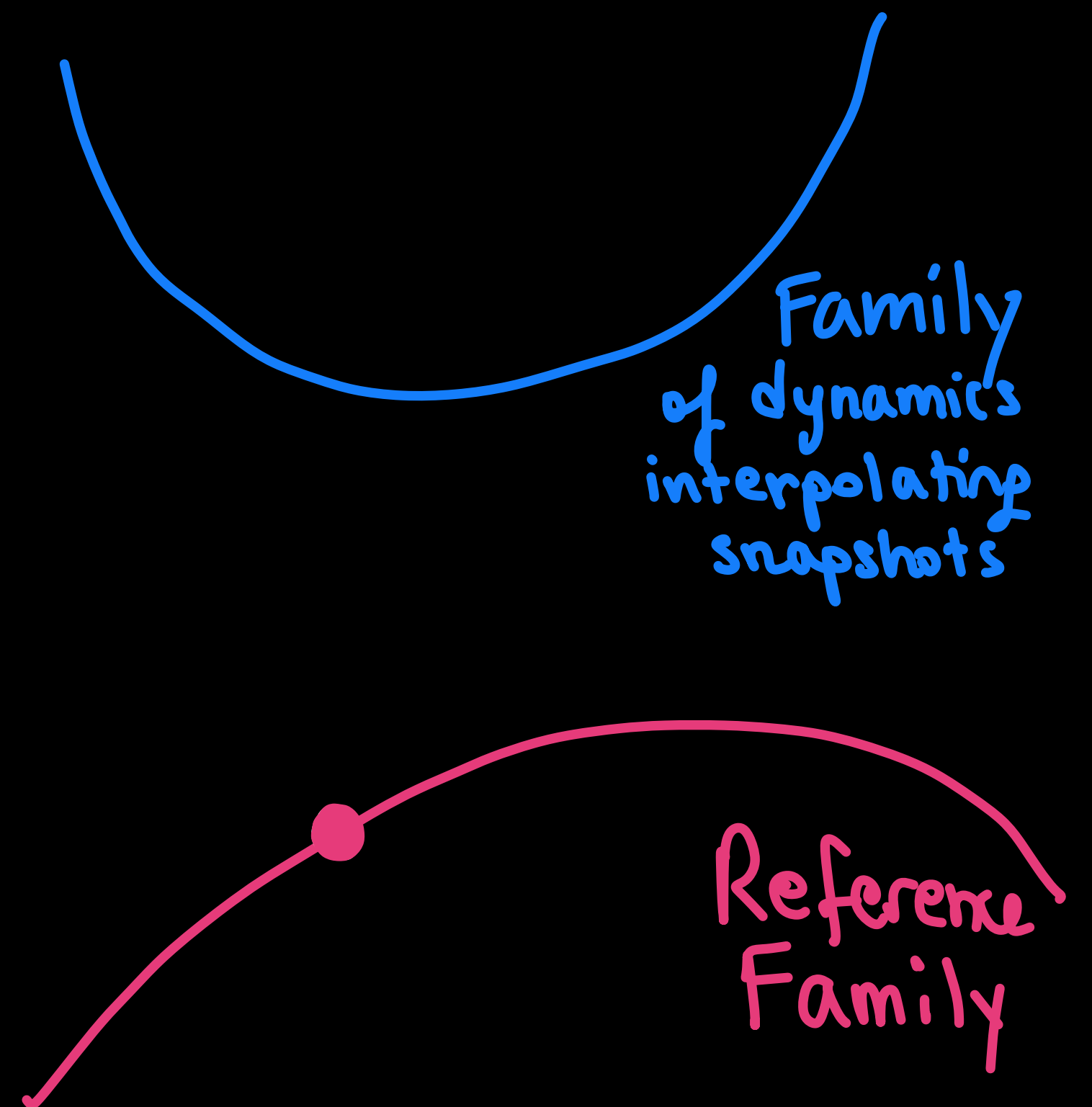
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
- When we have a family of references, we iteratively refine it!



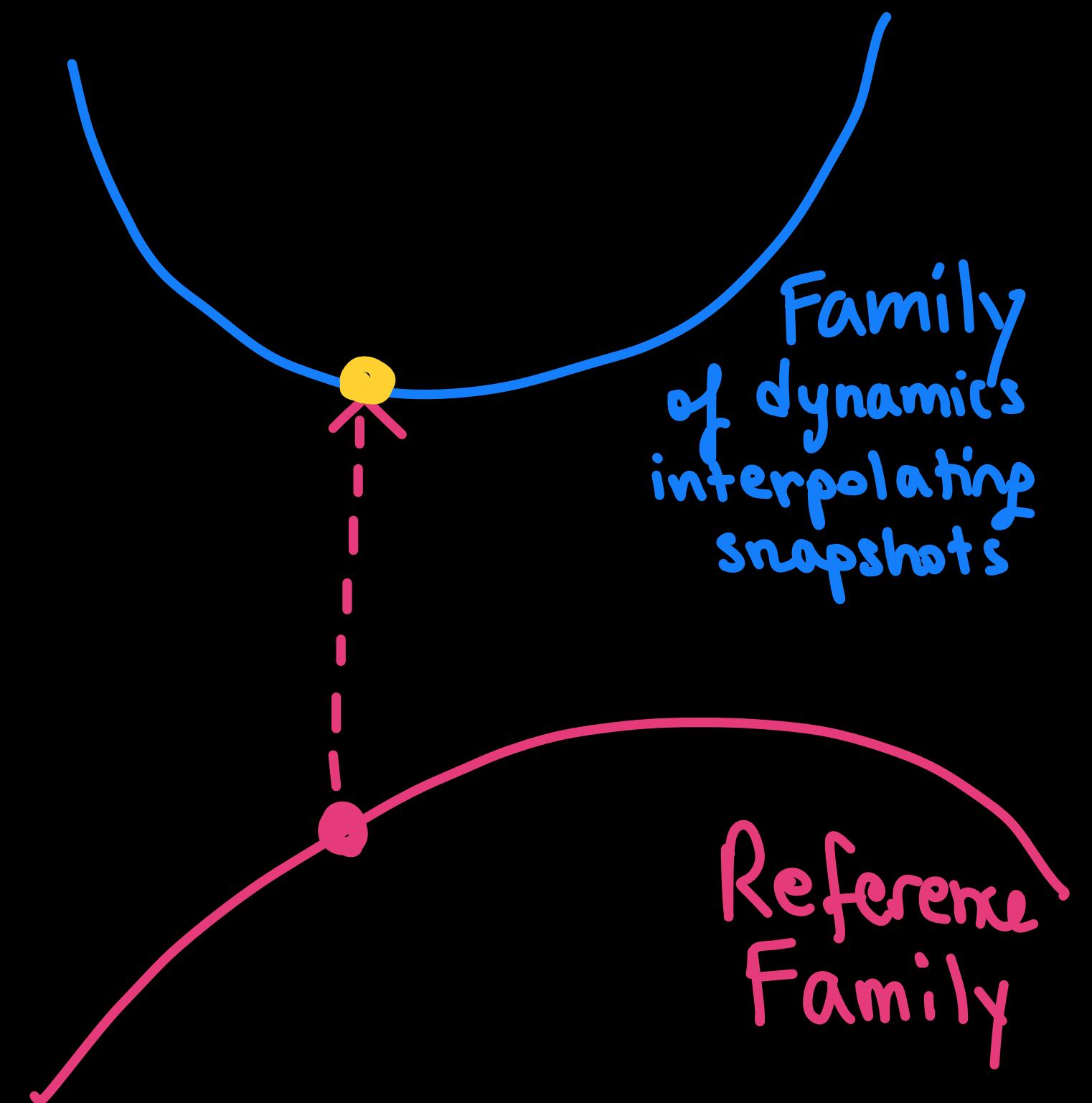
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
 - When we have a family of references, we iteratively refine it!
- STEP 1 : given the current reference dynamic, find solution to SB problem



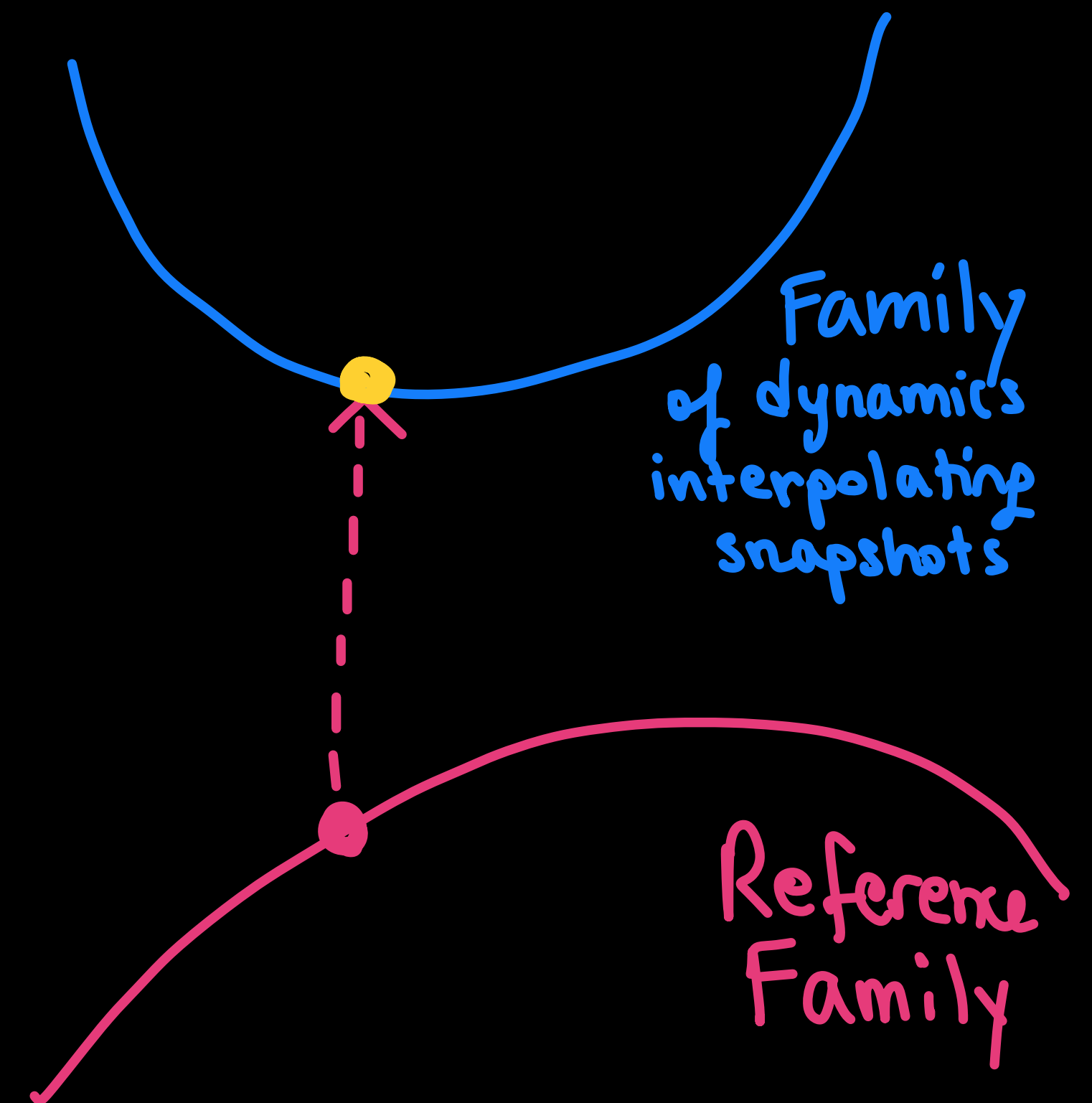
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
 - When we have a family of references, we iteratively refine it!
- STEP 1 : given the current reference dynamic, find solution to SB problem



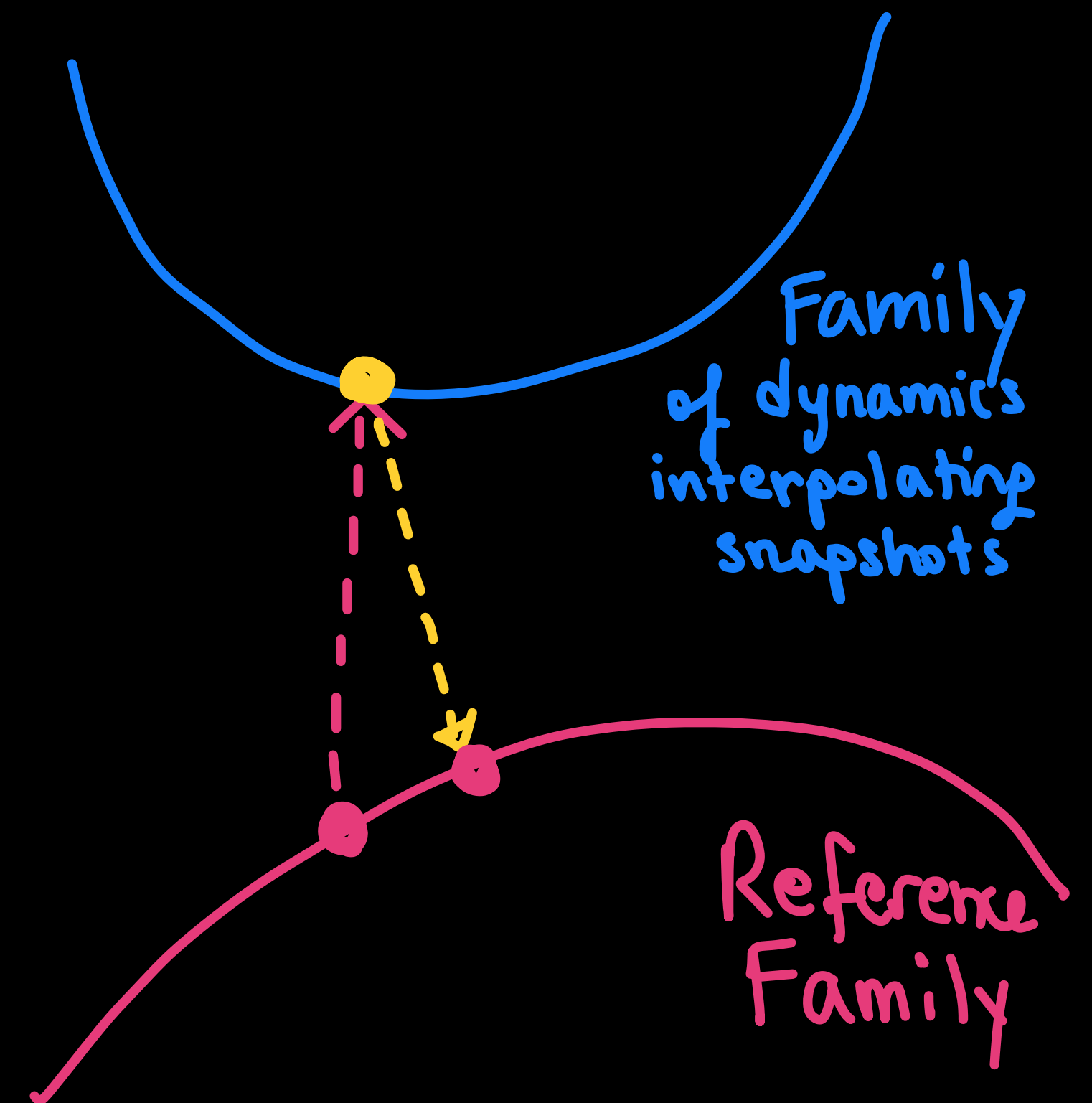
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
 - When we have a family of references, we iteratively refine it!
- STEP 1 : given the current reference dynamic, find solution to SB problem
- STEP 2 : use SB solution from step 1 to generate samples. Use these samples to solve a regression problem to find the new best reference in some parametric family



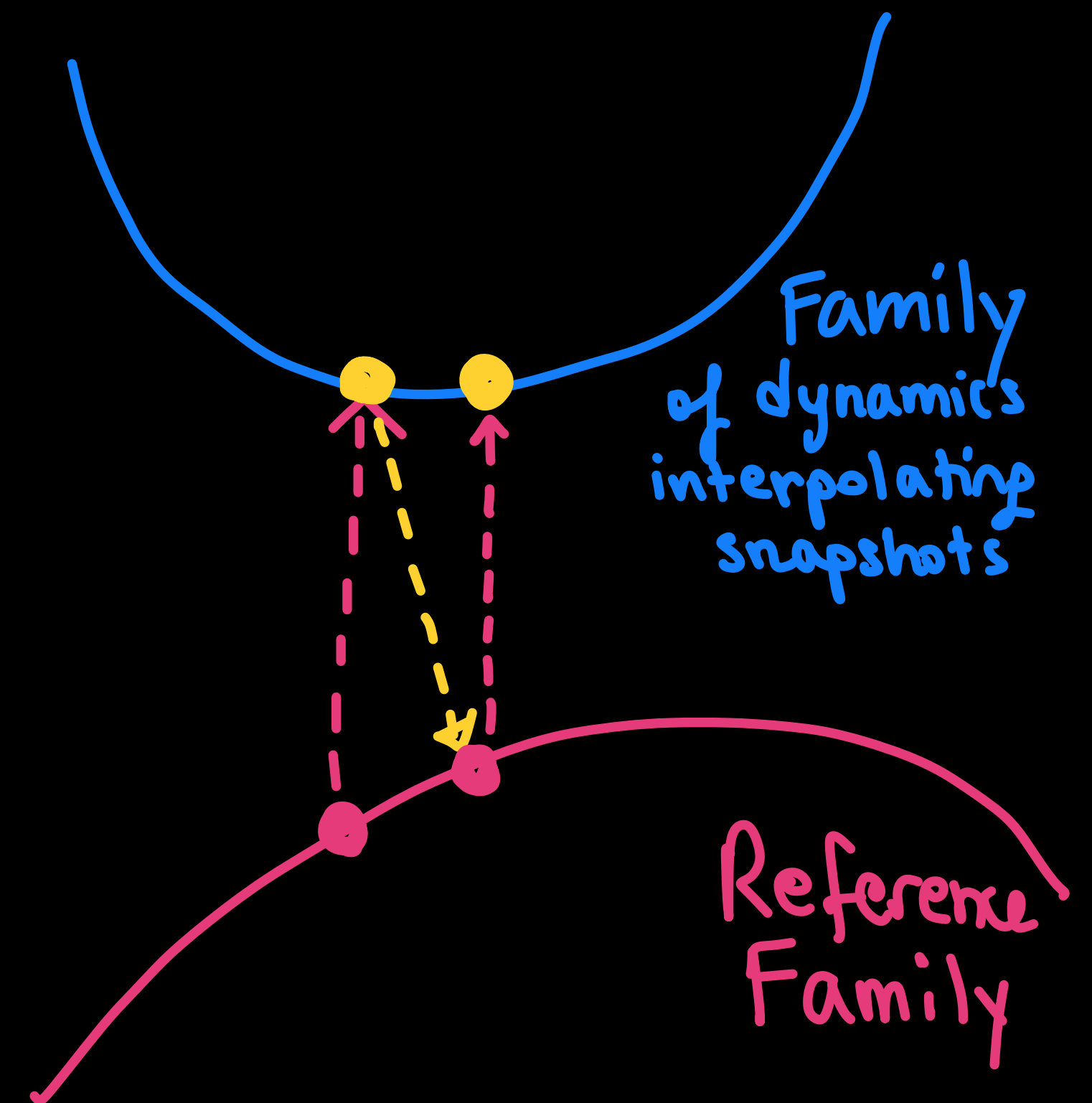
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
 - When we have a family of references, we iteratively refine it!
- STEP 1 : given the current reference dynamic, find solution to SB problem
- STEP 2 : use SB solution from step 1 to generate samples. Use these samples to solve a regression problem to find the new best reference in some parametric family



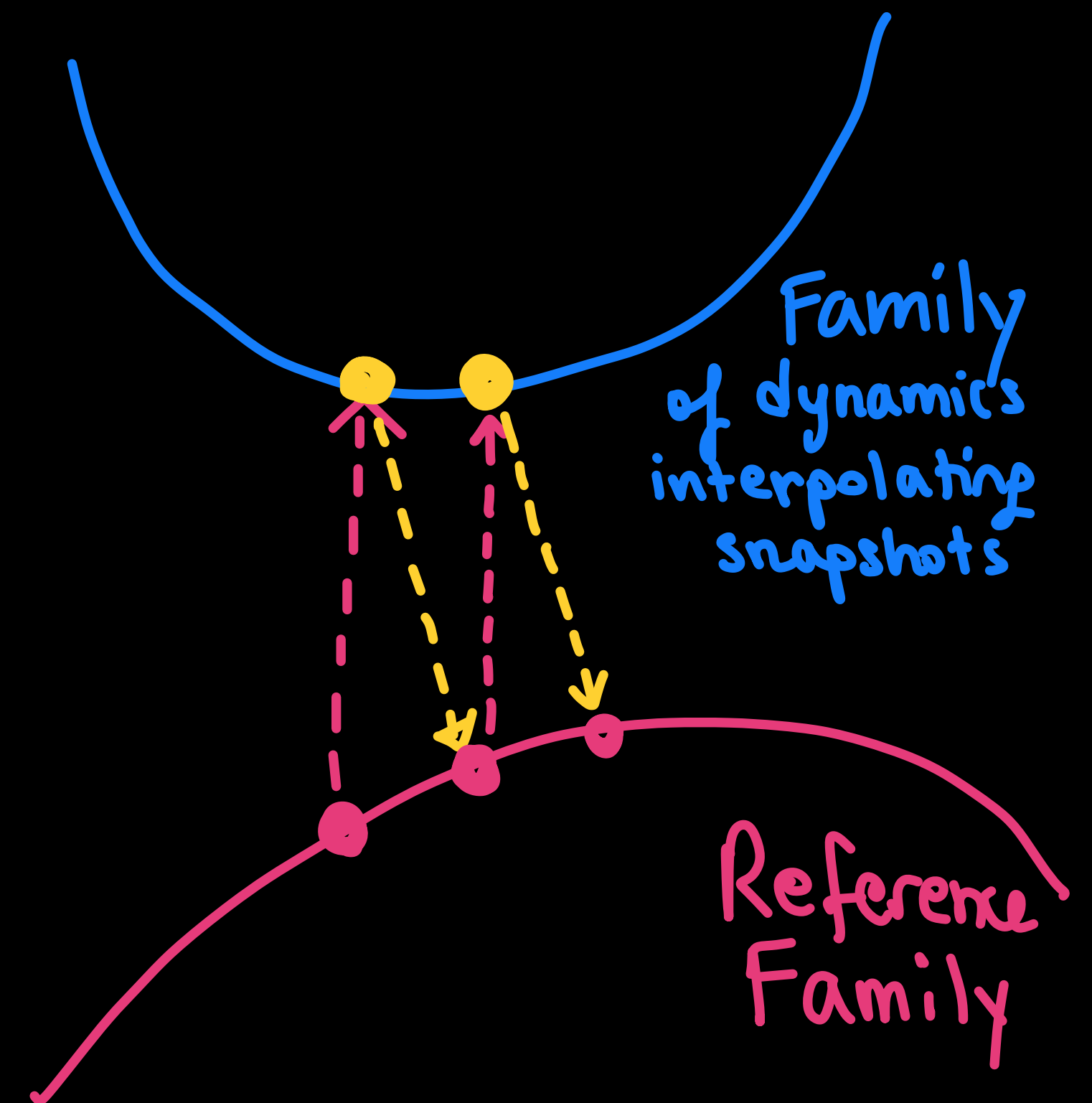
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
 - When we have a family of references, we iteratively refine it!
- STEP 1 : given the current reference dynamic, find solution to SB problem
- STEP 2 : use SB solution from step 1 to generate samples. Use these samples to solve a regression problem to find the new best reference in some parametric family



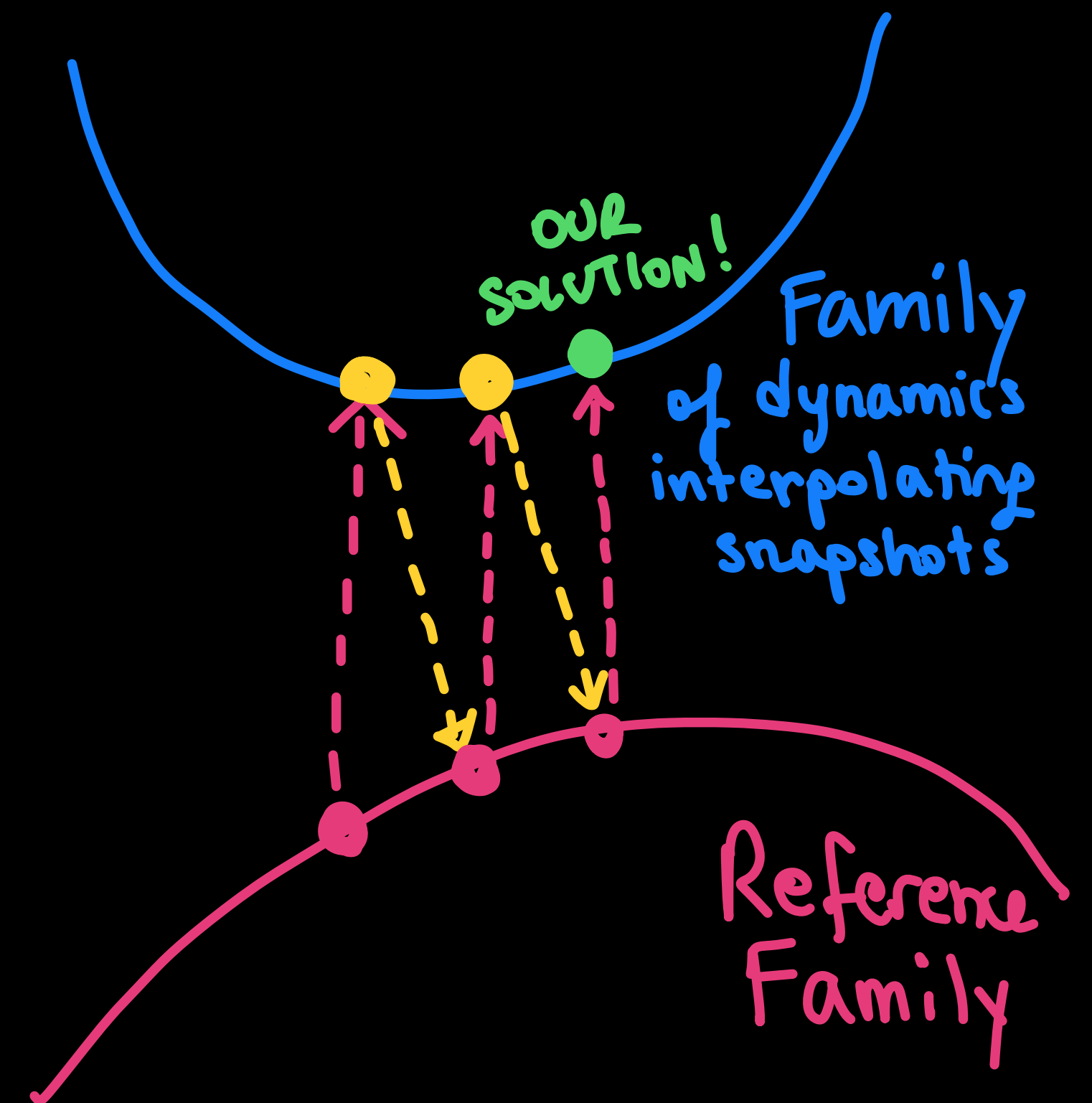
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
 - When we have a family of references, we iteratively refine it!
- STEP 1 : given the current reference dynamic, find solution to SB problem
- STEP 2 : use SB solution from step 1 to generate samples. Use these samples to solve a regression problem to find the new best reference in some parametric family



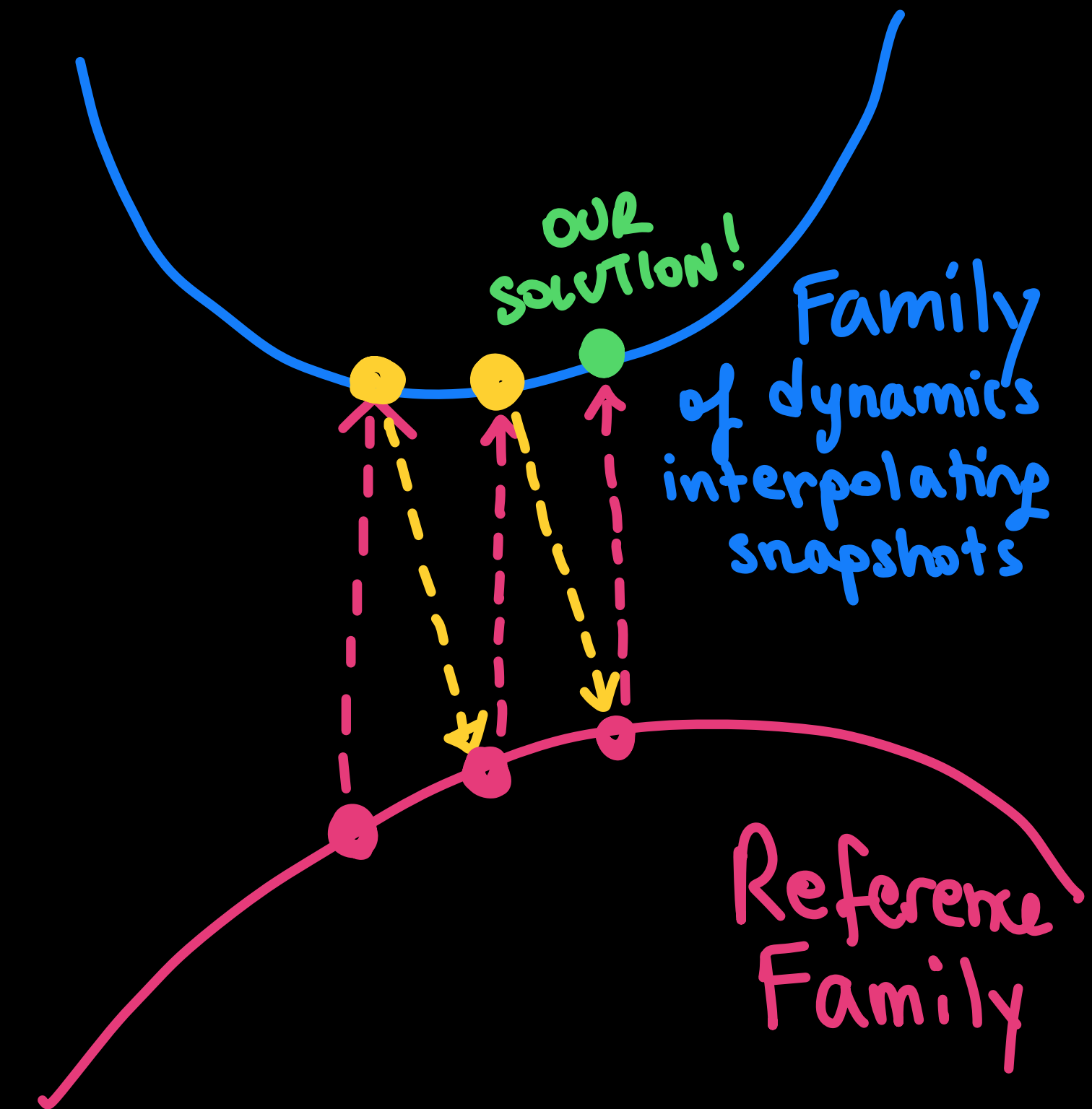
OUR METHOD : MULTIMARGINAL SB WITH ITERATIVE REFERENCE REFINEMENT

- It is more realistic for practitioners to specify a family, not a single reference
 - When we have a family of references, we iteratively refine it!
- STEP 1 : given the current reference dynamic, find solution to SB problem
- STEP 2 : use SB solution from step 1 to generate samples. Use these samples to solve a regression problem to find the new best reference in some parametric family



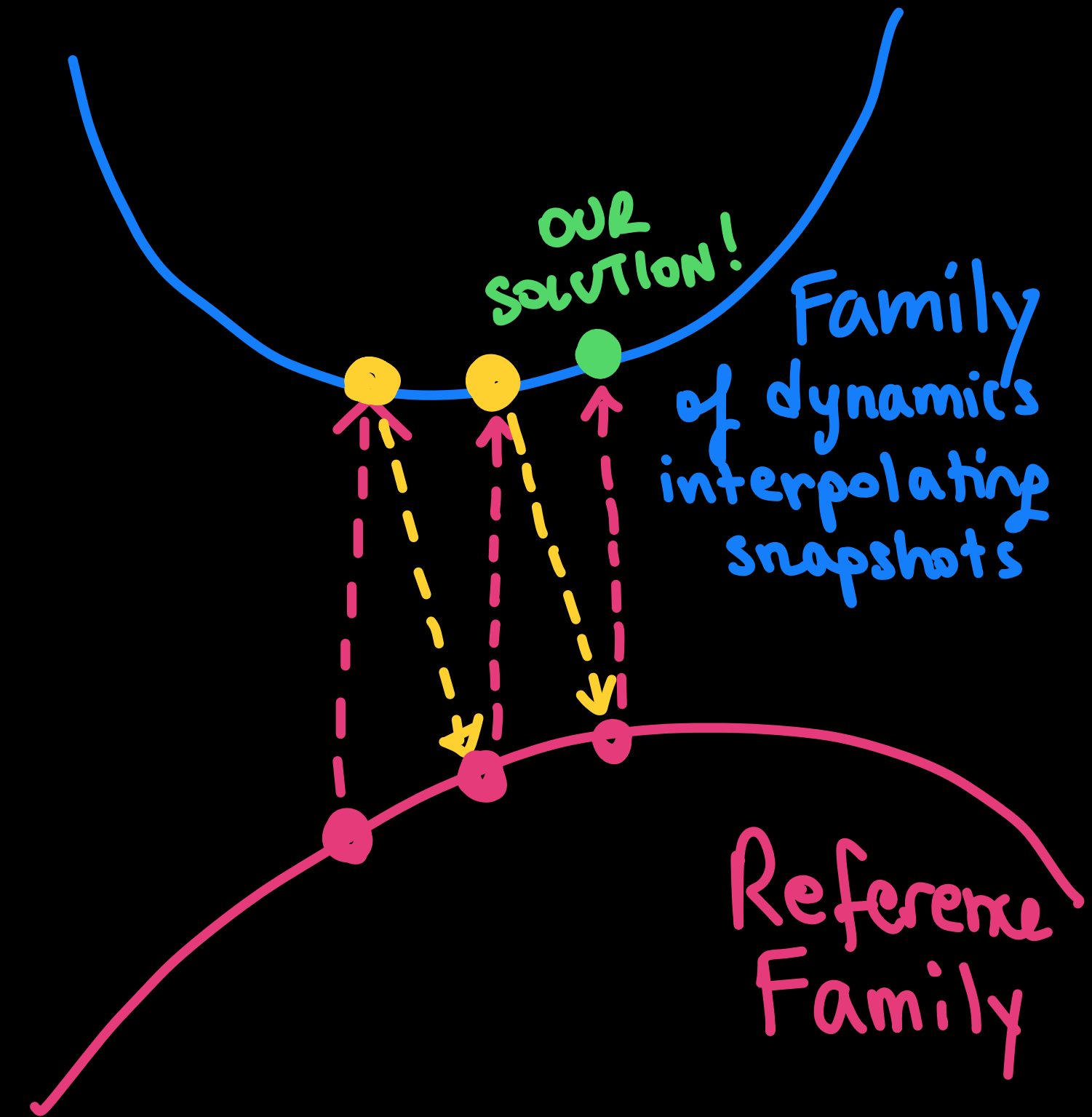
OUR METHOD IN PRACTICE

- Step 1: standard multi-marginal SB step



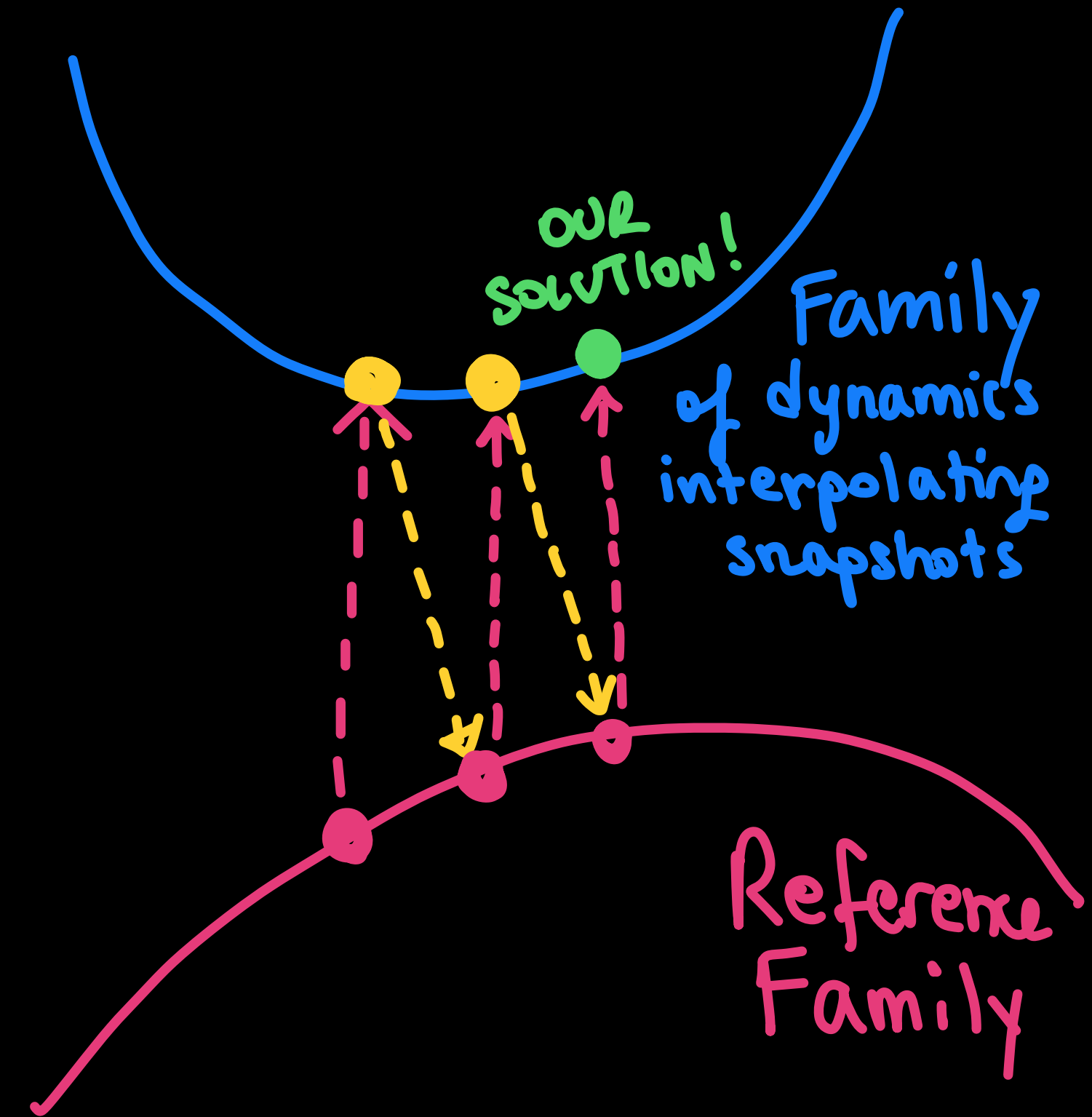
OUR METHOD IN PRACTICE

- Step 1: standard multi-marginal SB step
→ in principle*, any method solving SB problem can be used



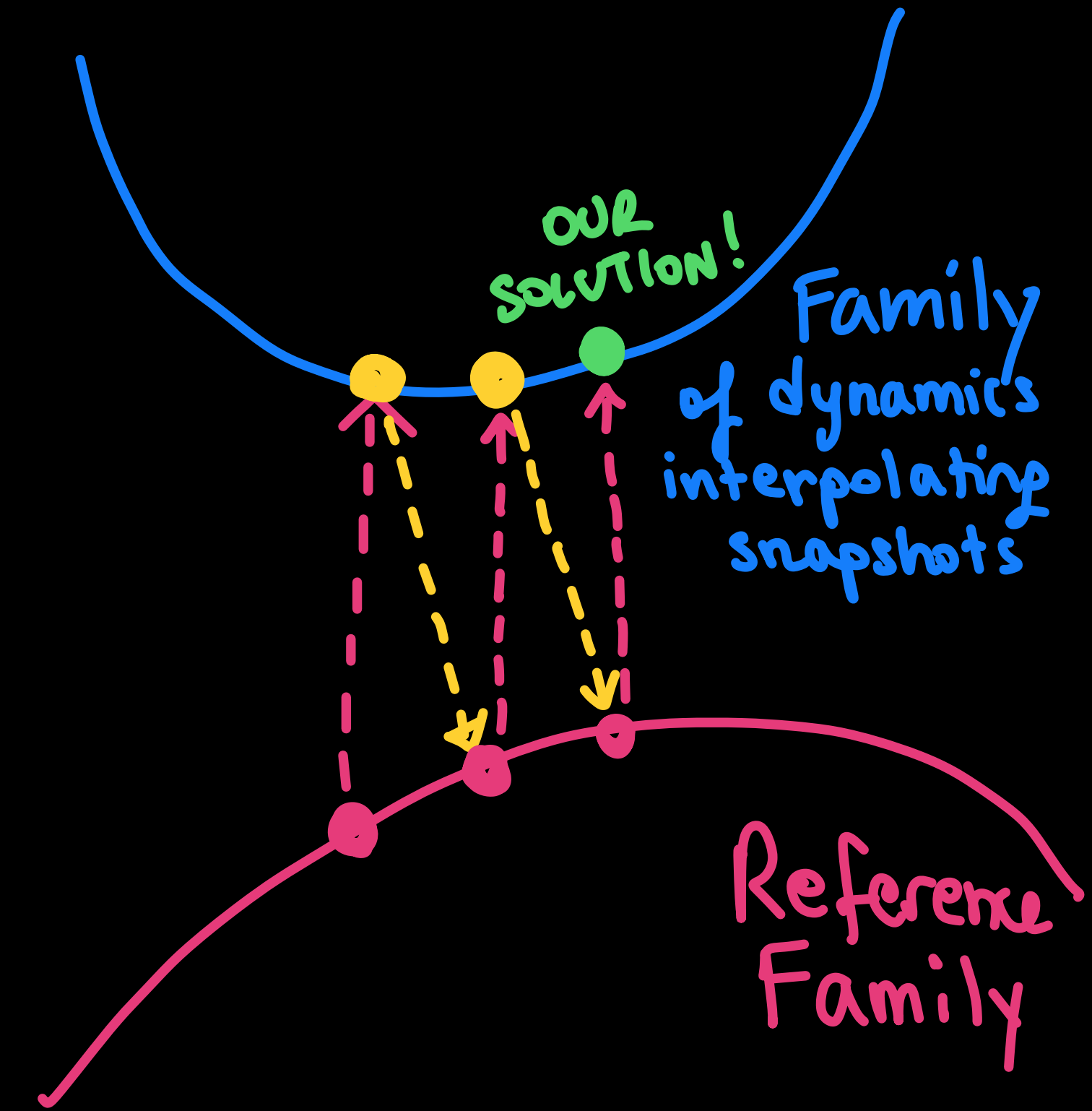
OUR METHOD IN PRACTICE

- Step 1: standard multi-marginal SB step
 - in principle*, any method solving SB problem can be used
 - We pick Vargas et al. 2021



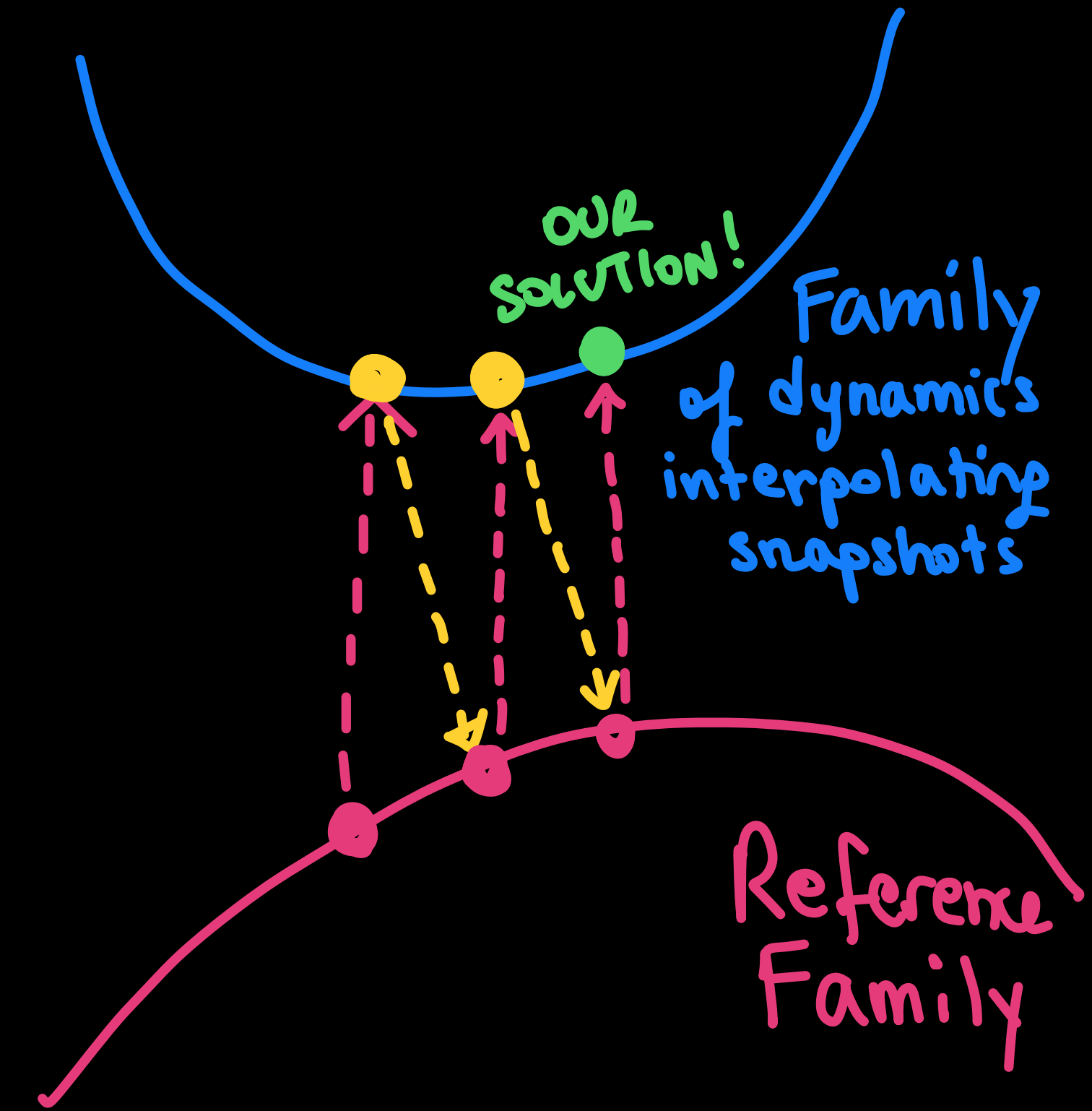
OUR METHOD IN PRACTICE

- Step 1: standard multi-marginal SB step
→ in principle*, any method solving SB problem can be used
→ We pick Vargas et al. 2021
- Step 2: ① Generate sample trajectories by starting at observations & simulating forward and backward (e.g. with Euler-Maruyama)

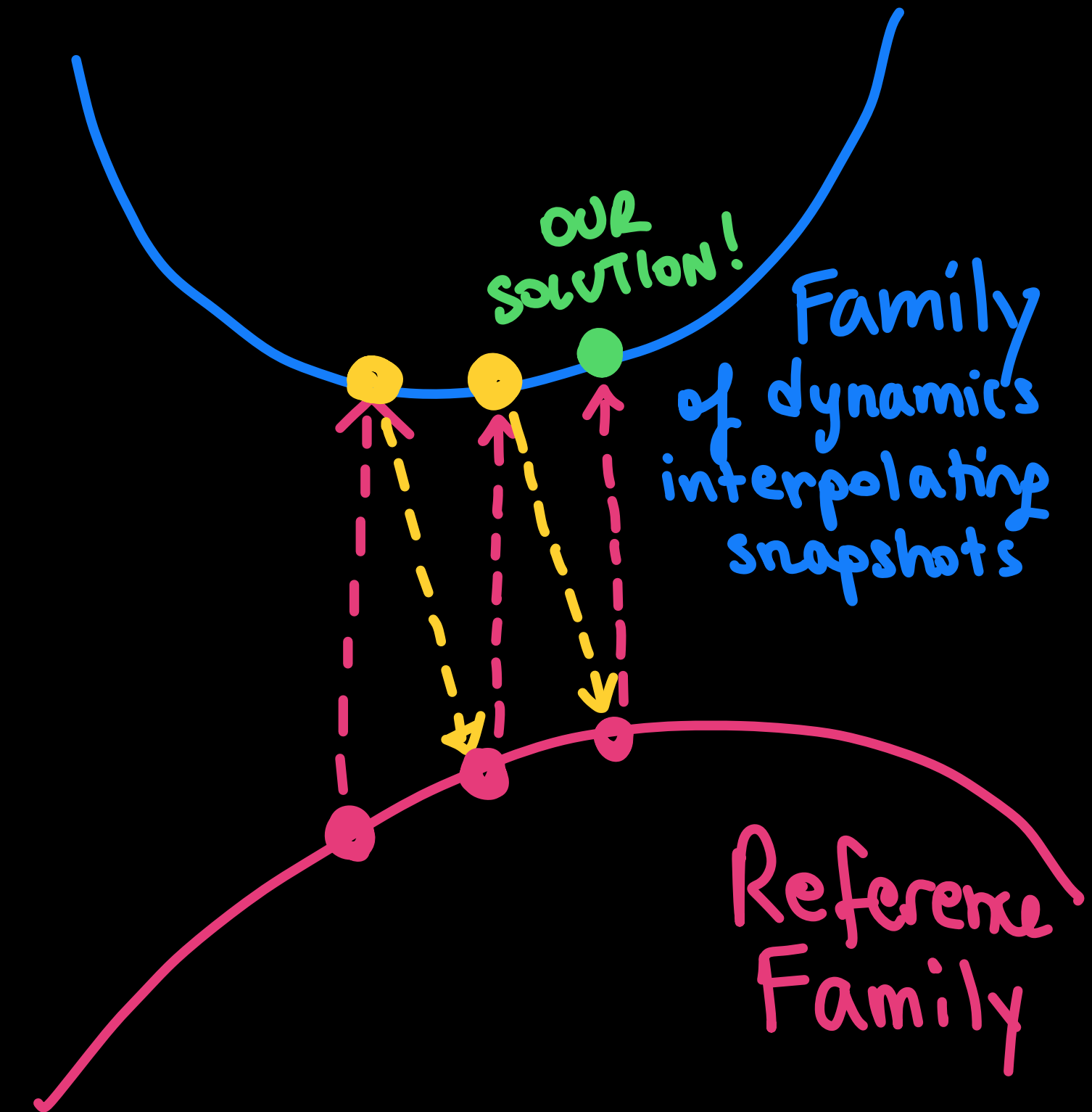


OUR METHOD IN PRACTICE

- Step 1: standard multi-marginal SB step
 - in principle*, any method solving SB problem can be used
 - We pick Vargas et al. 2021
- Step 2:
 - ① Generate sample trajectories by starting at observations & simulating forward and backward (e.g. with Euler-Maruyama)
 - ② Find reference in reference family that maximizes log-likelihood of the trajectories

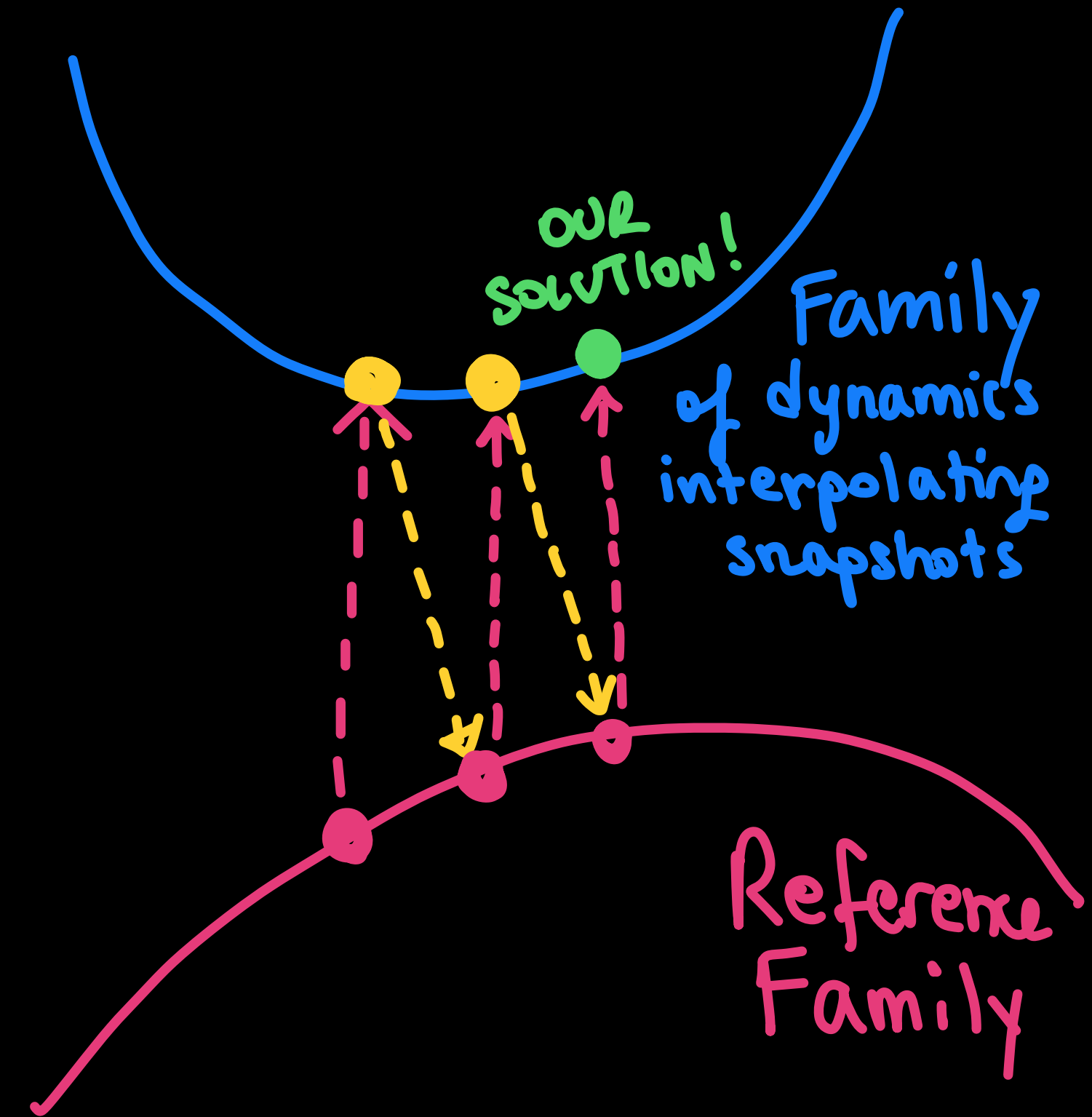


DOES THE ALGORITHM CONVERGE TO A SOLUTION?



DOES THE ALGORITHM CONVERGE TO A SOLUTION?

- Our Theorem: If all SDEs follow standard regularity conditions & reference family is convex, then our algorithm converges to the best divergence value given marginal constraints & family class

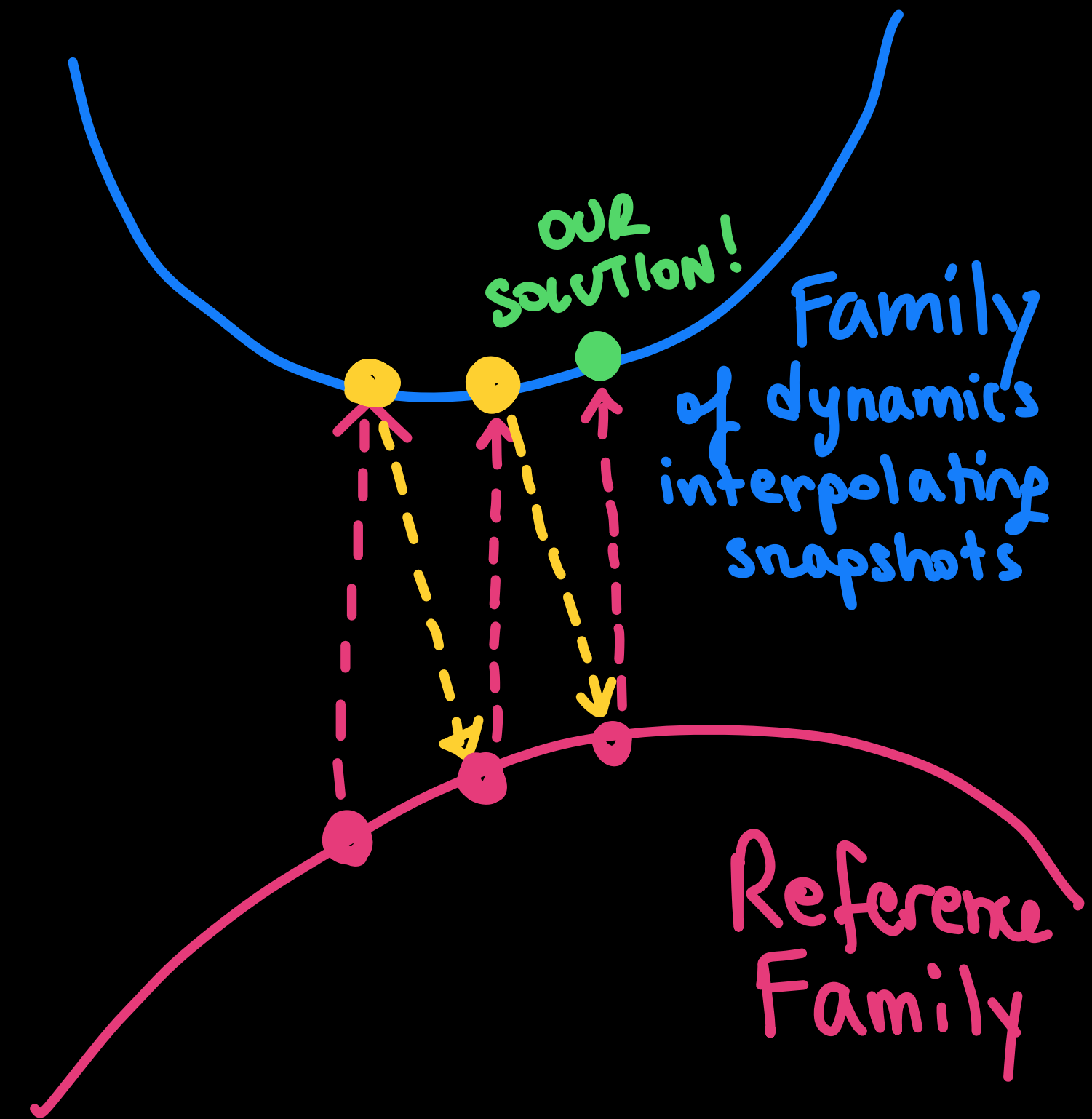


DOES THE ALGORITHM CONVERGE TO A SOLUTION?

- Our Theorem: If all SDEs follow standard regularity conditions & reference family is convex, then our algorithm converges to the best divergence value given marginal constraints & family class

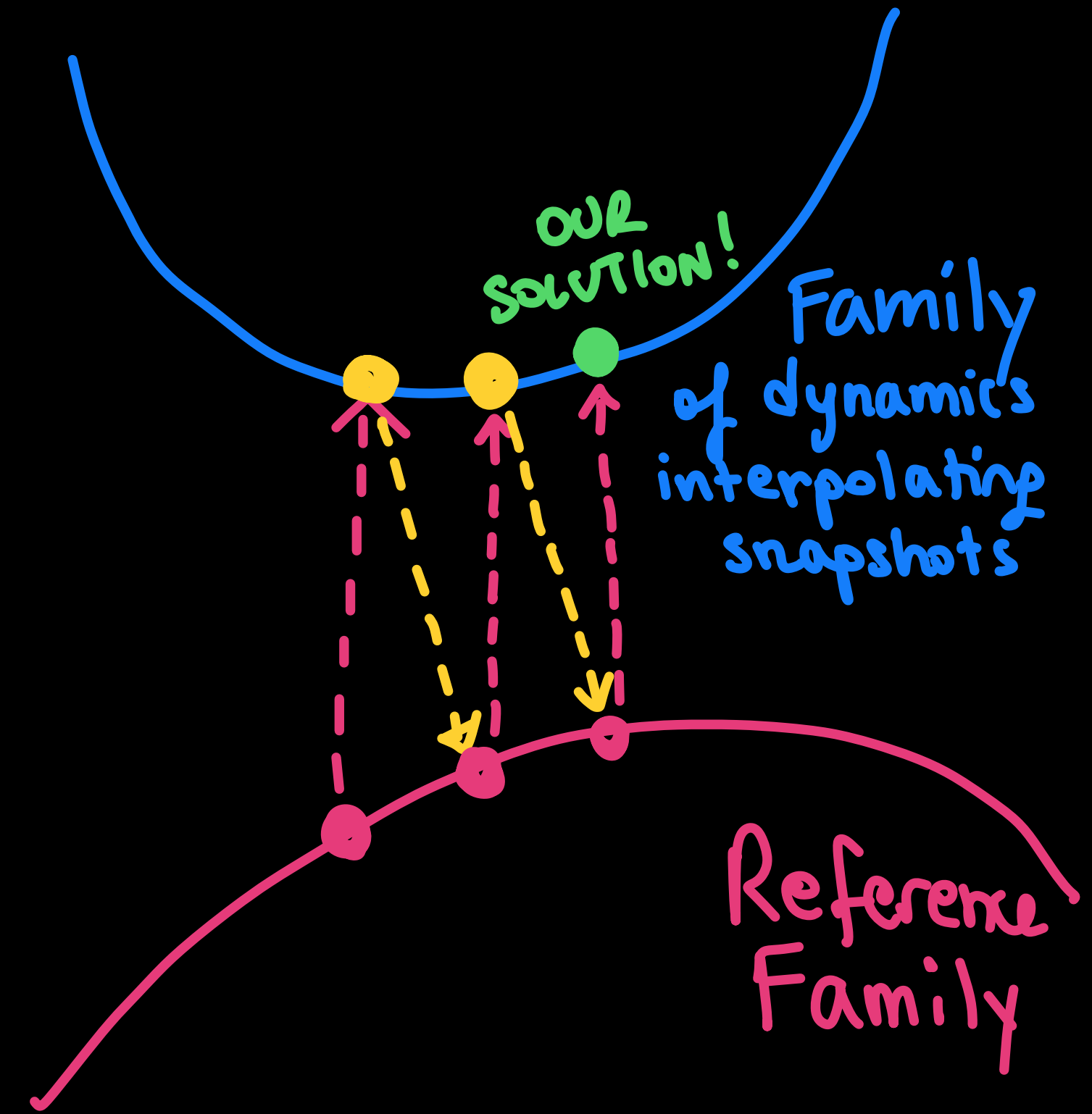
- Proof idea:

① Girsanov + SDE assumptions $\rightarrow$ finite KL + convex path-measure sets



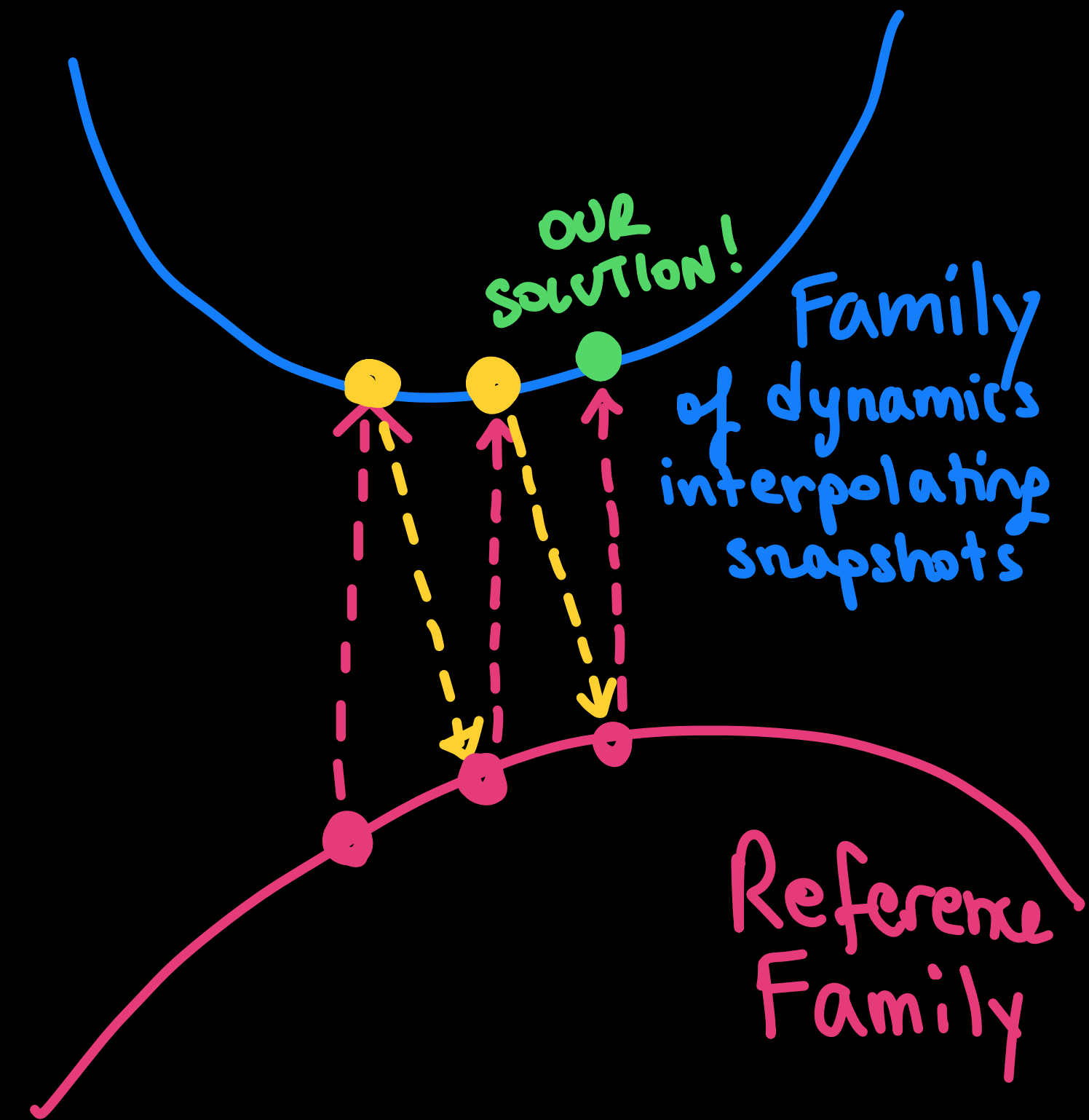
DOES THE ALGORITHM CONVERGE TO A SOLUTION?

- Our Theorem: If all SDEs follow standard regularity conditions & reference family is convex, then our algorithm converges to the best divergence value given marginal constraints & family class
- Proof idea:
 - ① Girsanov + SDE assumptions $\rightarrow$ finite KL + convex path-measure sets
 - ② Iterative I-projections (Csiszár and Shields, 2004) give monotone KL descent



DOES THE ALGORITHM CONVERGE TO A SOLUTION?

- Our Theorem: If all SDEs follow standard regularity conditions & reference family is convex, then our algorithm converges to the best divergence value given marginal constraints & family class
- Proof idea:
 - ① Girsanov + SDE assumptions $\rightarrow$ finite KL + convex path-measure sets
 - ② Iterative I-projections (Csiszár and Shields, 2004) give monotone KL descent
 - ③ KL value converges to optimal divergence

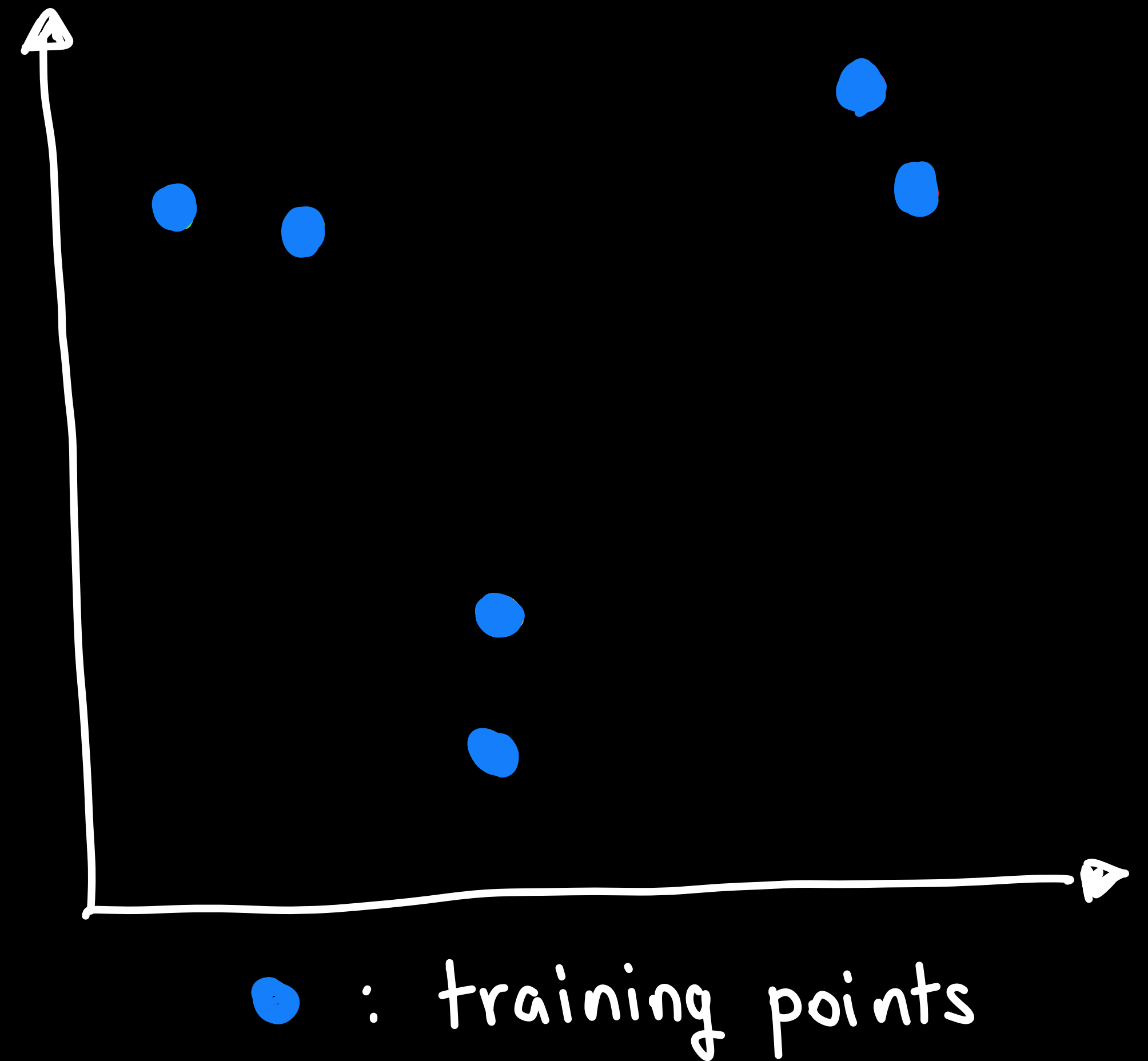


EXPERIMENTAL SETUP

- Synthetic and real data experiments: each explores a different application with a different natural reference family

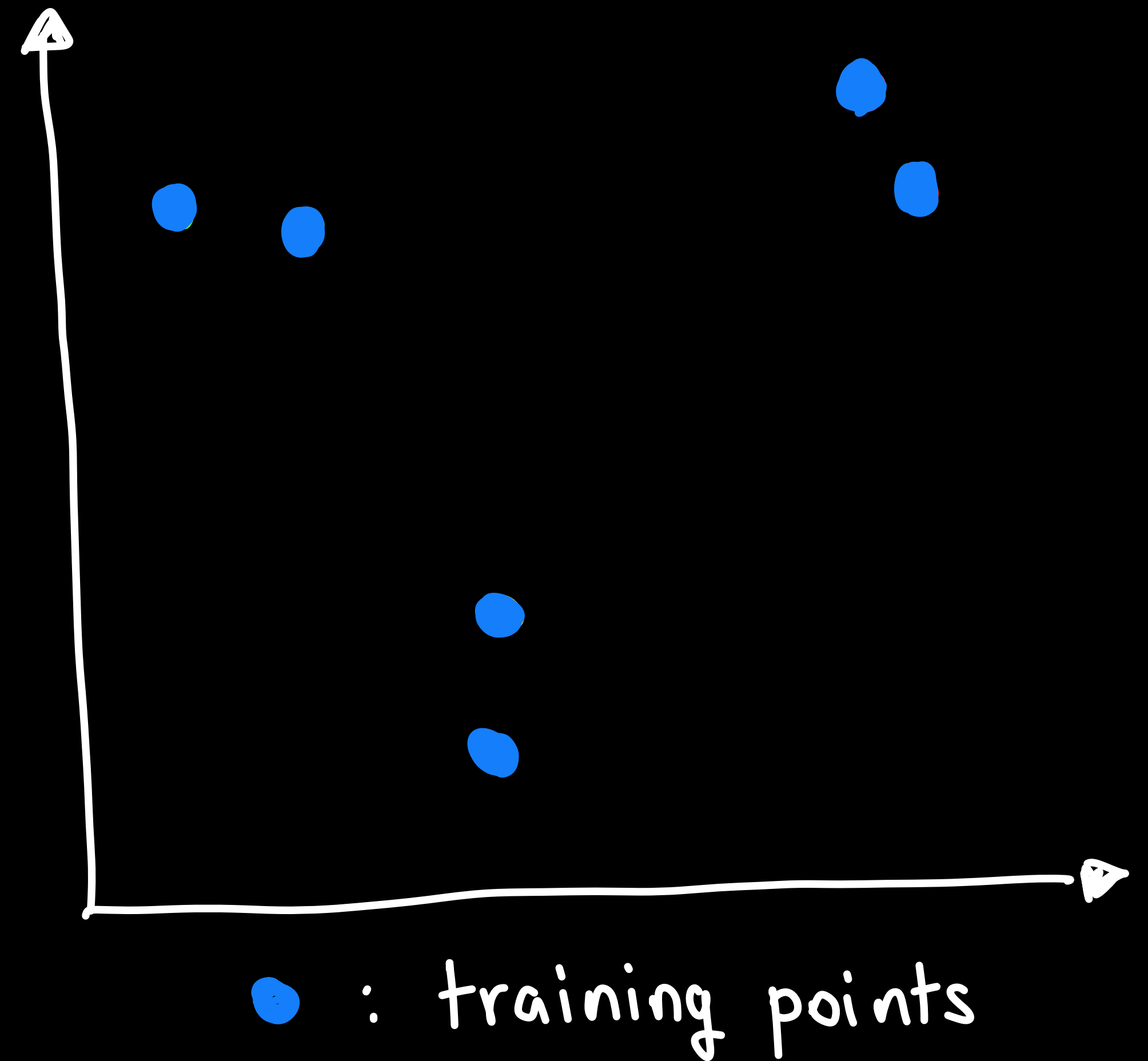
EXPERIMENTAL SETUP

- Synthetic and real data experiments: each explores a different application with a different natural reference family



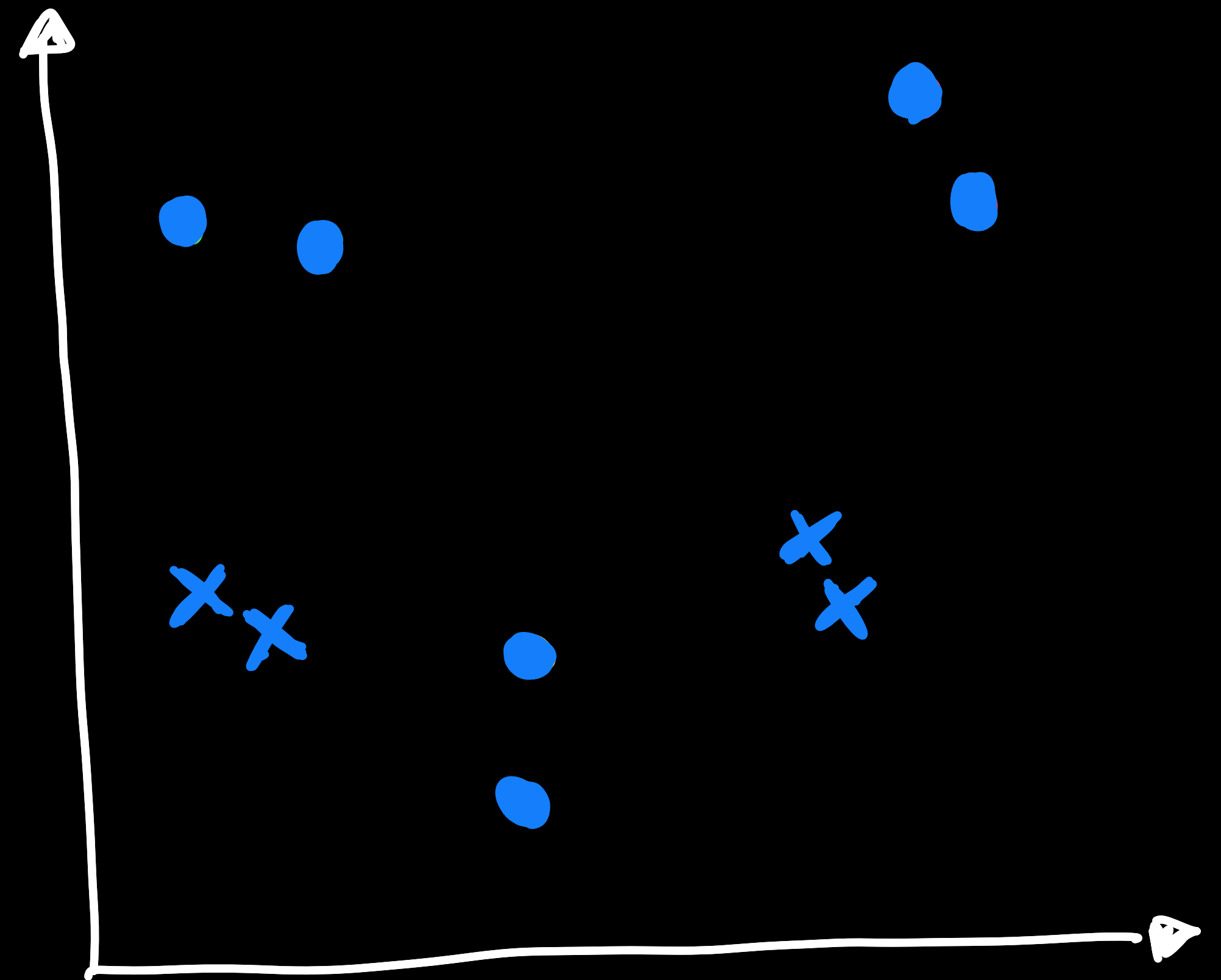
EXPERIMENTAL SETUP

- Synthetic and real data experiments: each explores a different application with a different natural reference family
- To test if what we learnt is good: retain some validation snapshots



EXPERIMENTAL SETUP

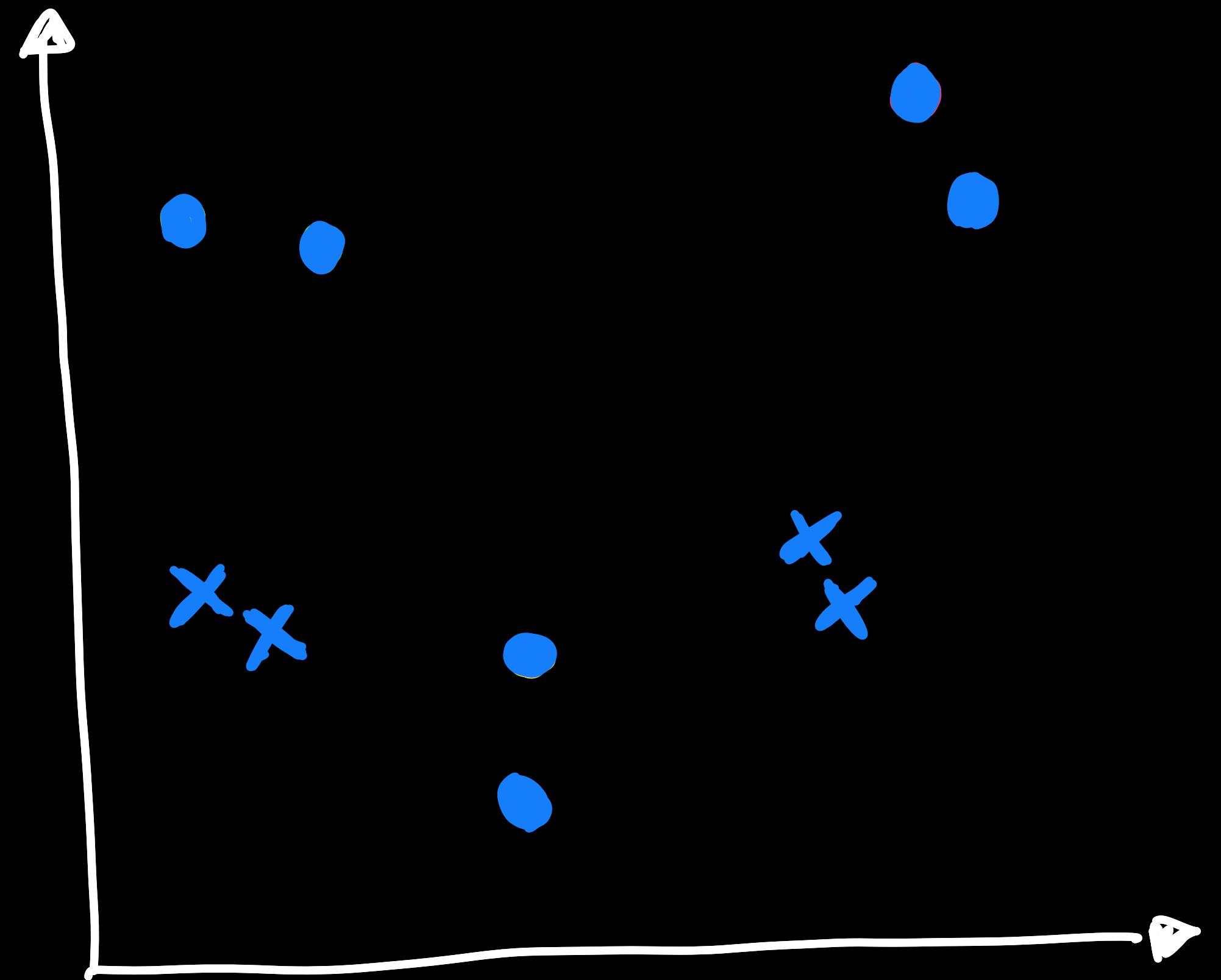
- Synthetic and real data experiments: each explores a different application with a different natural reference family
- To test if what we learnt is good: retain some validation snapshots



● : training points
x : validation points

EXPERIMENTAL SETUP

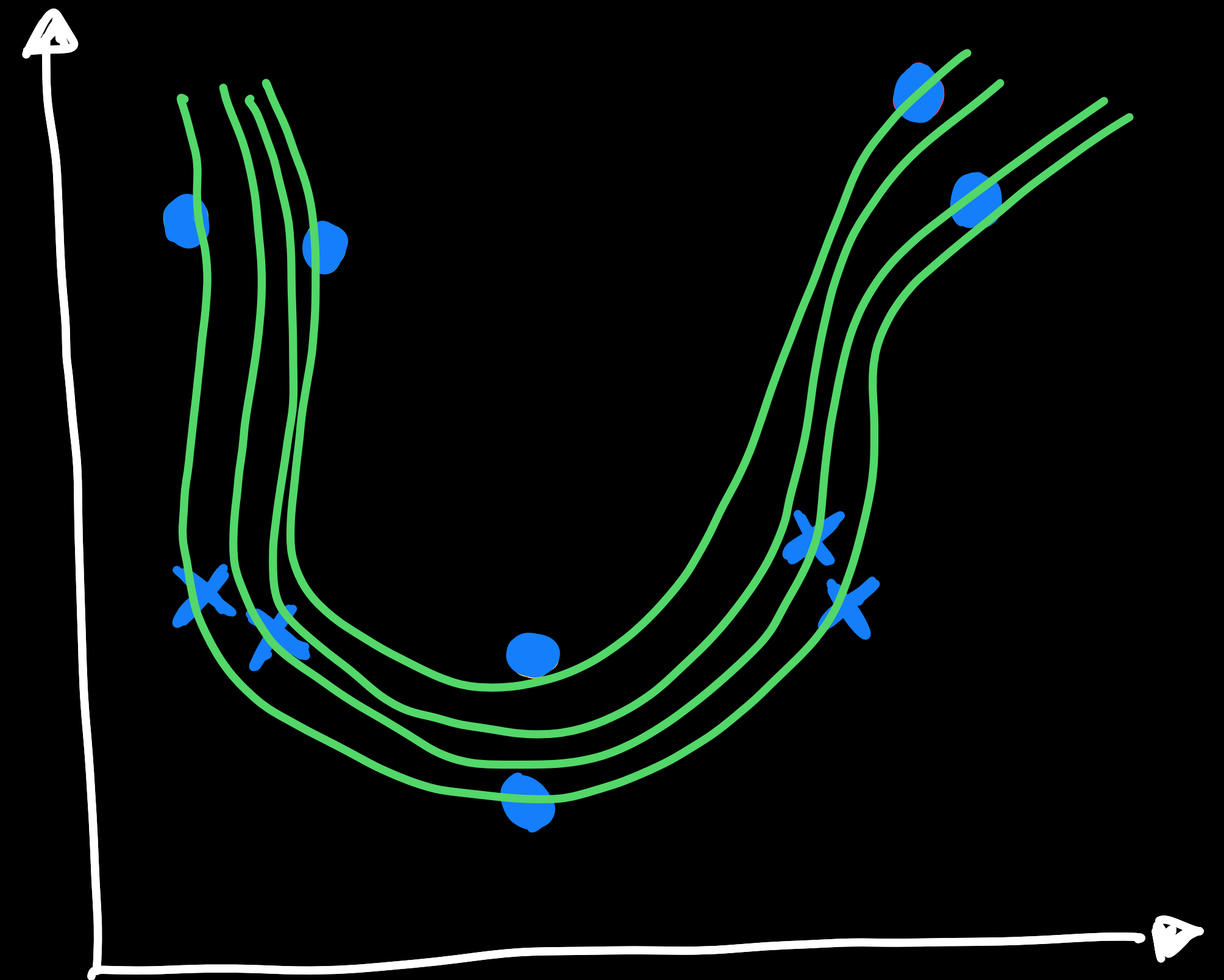
- Synthetic and real data experiments: each explores a different application with a different natural reference family
- To test if what we learnt is good: retain some validation snapshots
- ✓ **SUCCESS**: if sampled trajectories go through validation points



● : training points
x : validation points

EXPERIMENTAL SETUP

- Synthetic and real data experiments: each explores a different application with a different natural reference family
- To test if what we learnt is good: retain some validation snapshots
- ✓ **SUCCESS**: if sampled trajectories go through validation points



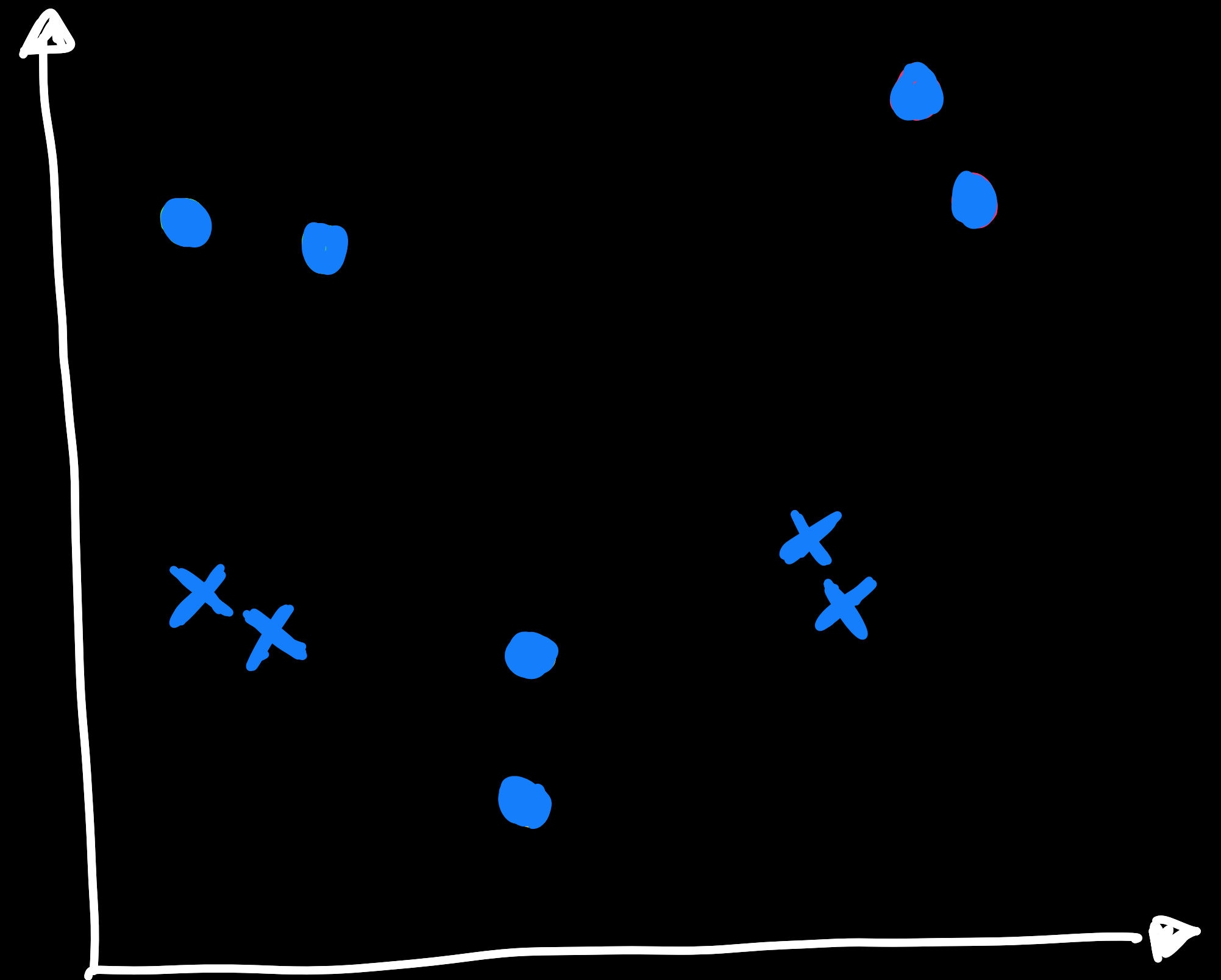
● : training points
x : validation points

EXPERIMENTAL SETUP

- Synthetic and real data experiments: each explores a different application with a different natural reference family
- To test if what we learnt is good: retain some validation snapshots

✓ **SUCCESS**: if sampled trajectories go through validation points

✗ **FAILURE**: if sampled trajectories miss the validation points



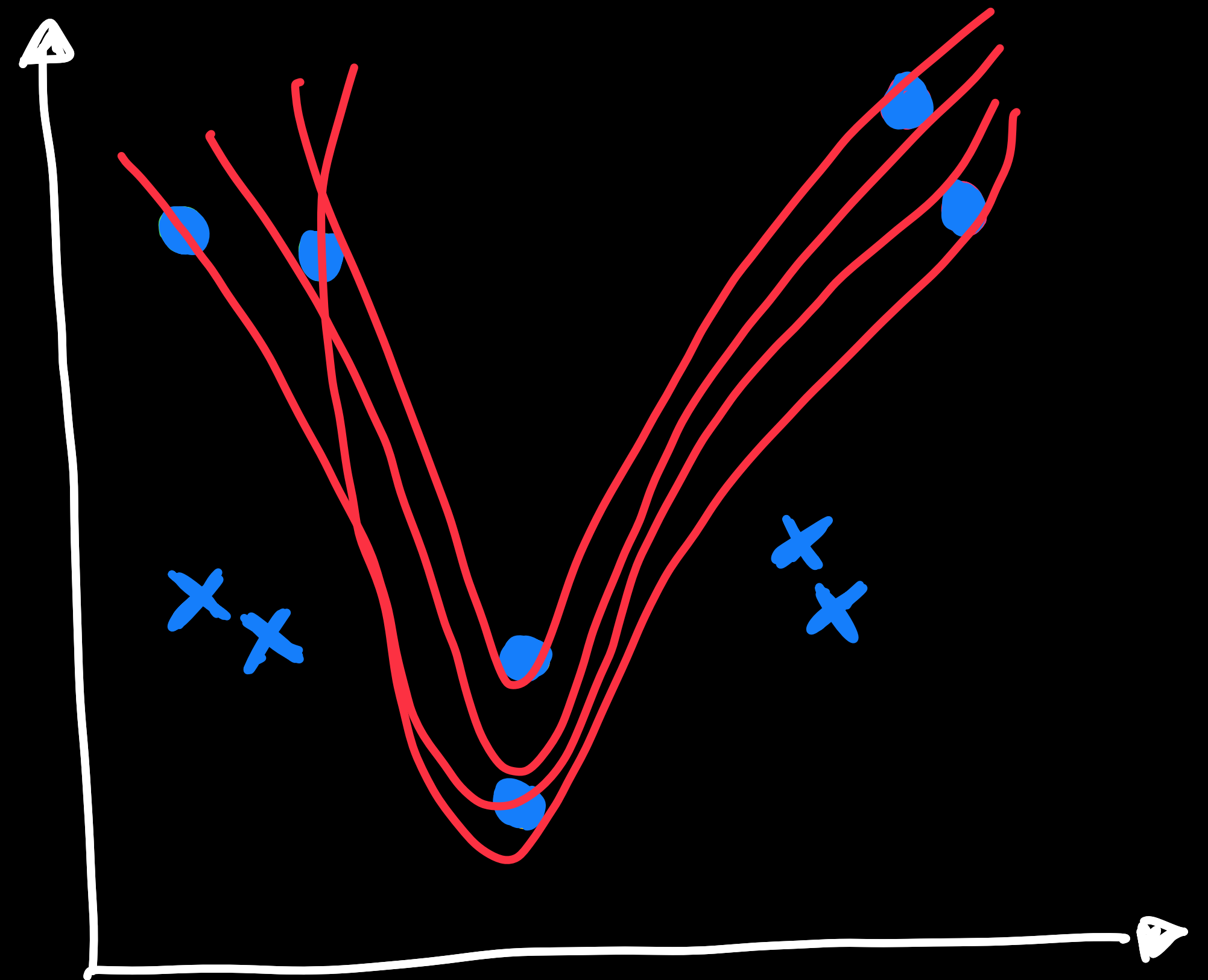
● : training points
x : validation points

EXPERIMENTAL SETUP

- Synthetic and real data experiments: each explores a different application with a different natural reference family
- To test if what we learnt is good: retain some validation snapshots

✓ **SUCCESS**: if sampled trajectories go through validation points

✗ **FAILURE**: if sampled trajectories miss the validation points



● : training points
x : validation points

EXPERIMENT 1: GULF of MEXICO DRIFTERS

- Use real ocean current data: simulate particles (e.g. ocean debris) that evolve according to this field around a vortex-like feature



EXPERIMENT 1: GULF of MEXICO DRIFTERS

- Use real ocean current data: simulate particles (e.g. ocean debris) that evolve according to this field around a vortex-like feature
- In our method, use a reference parametric family of a vortex with unknown center, direction, and shape



EXPERIMENT 1: GULF of MEXICO DRIFTERS

- Use real ocean current data: simulate particles (e.g. ocean debris) that evolve according to this field around a vortex-like feature
- In our method, use a reference parametric family of a vortex with unknown center, direction, and shape



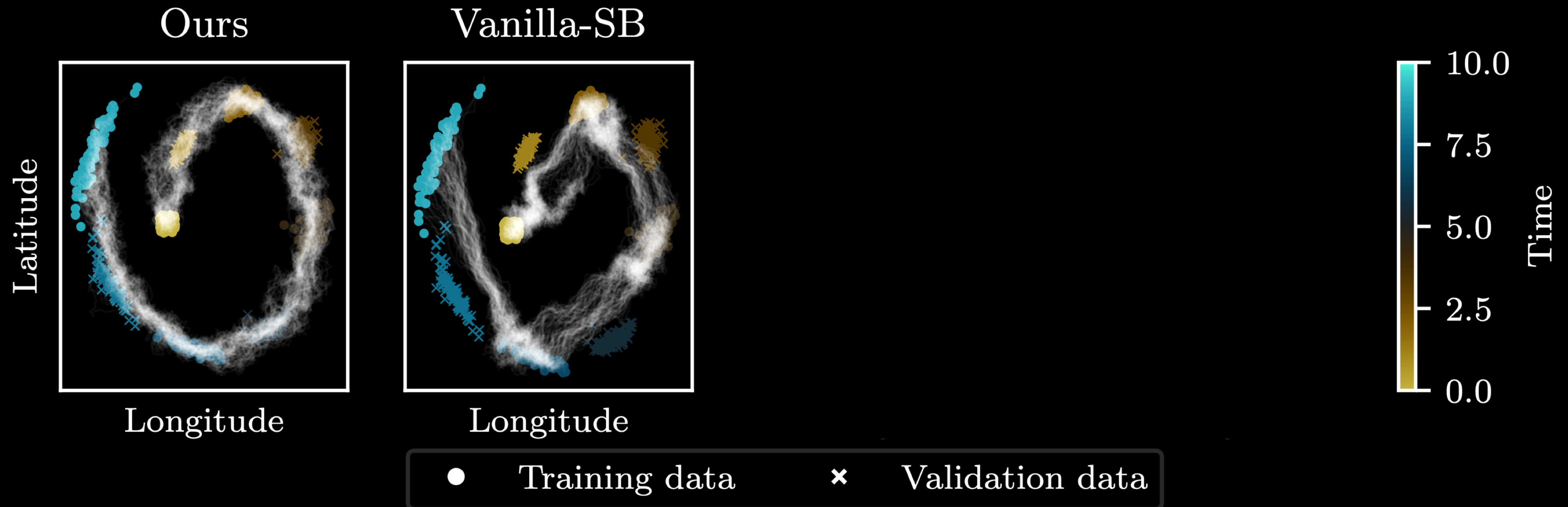
EXPERIMENT 1: GULF of MEXICO DRIFTERS

- Use real ocean current data: simulate particles (e.g. ocean debris) that evolve according to this field around a vortex-like feature
- In our method, use a reference parametric family of a vortex with unknown center, direction, and shape



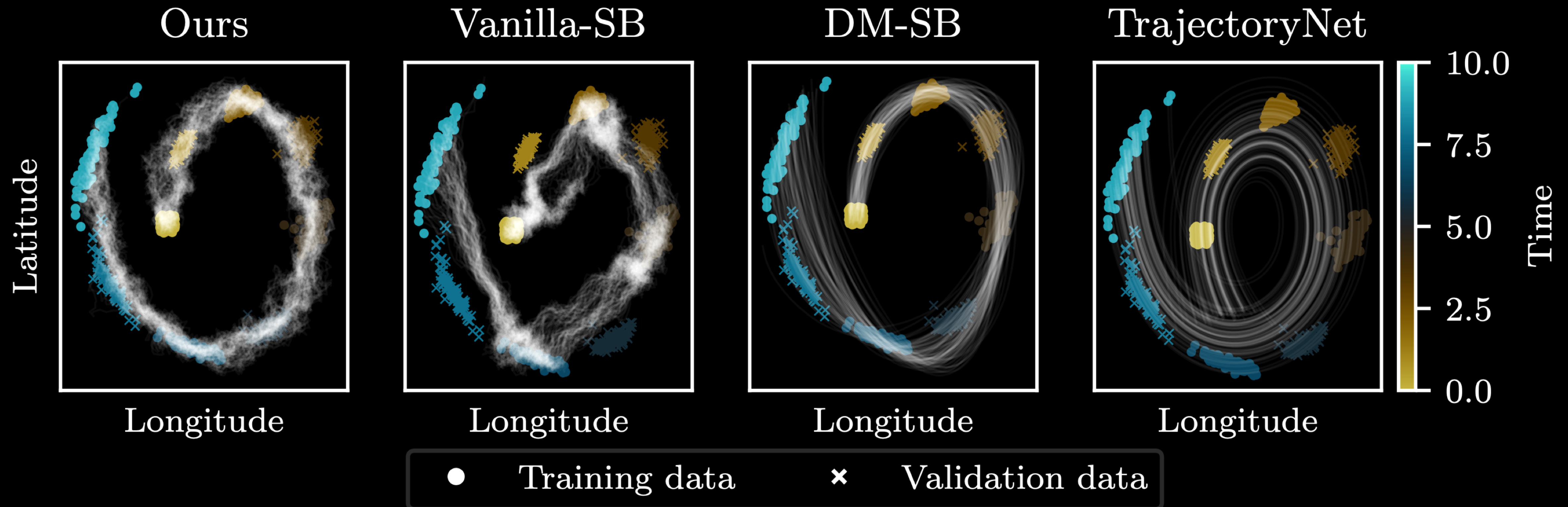
EXPERIMENT 1: GULF of MEXICO DRIFTERS

- Use real ocean current data: simulate particles (e.g. ocean debris) that evolve according to this field around a vortex-like feature
- In our method, use a reference parametric family of a vortex with unknown center, direction, and shape



EXPERIMENT 1: GULF of MEXICO DRIFTERS

- Use real ocean current data: simulate particles (e.g. ocean debris) that evolve according to this field around a vortex-like feature
- In our method, use a reference parametric family of a vortex with unknown center, direction, and shape



EXPERIMENT 2 : REPRESSILATOR

- The repressilator is a synthetic gene regulatory network that functions as a biological oscillator → three genes that inhibit each other in a cyclic manner



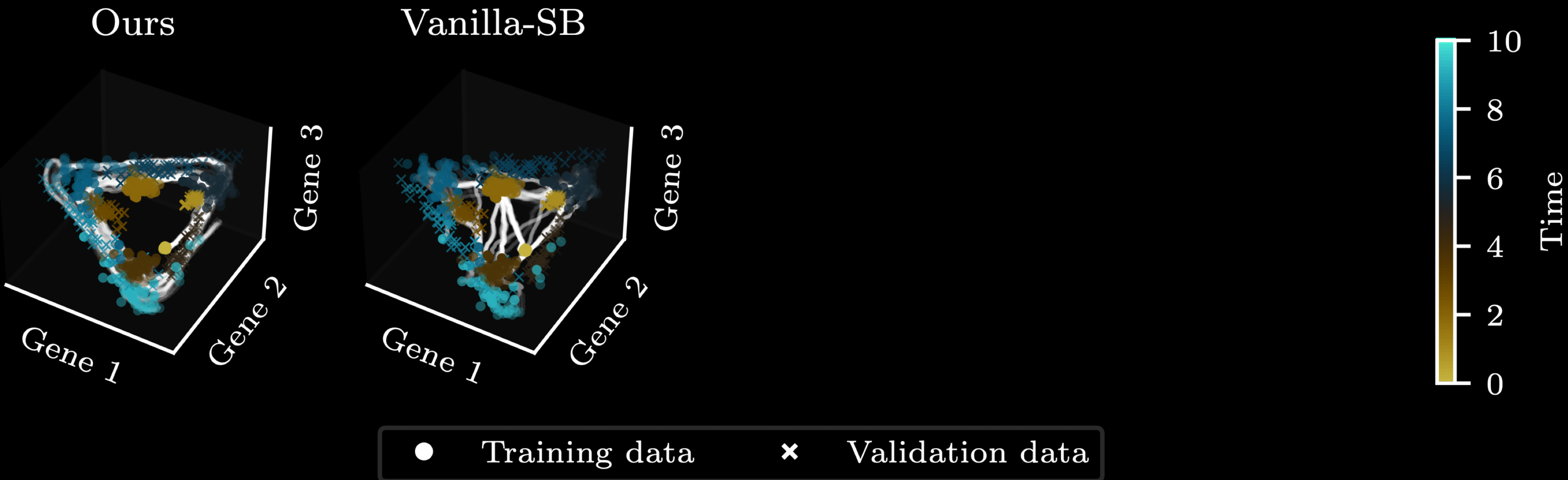
EXPERIMENT 2 : REPRESSILATOR

- The repressilator is a synthetic gene regulatory network that functions as a biological oscillator → three genes that inhibit each other in a cyclic manner



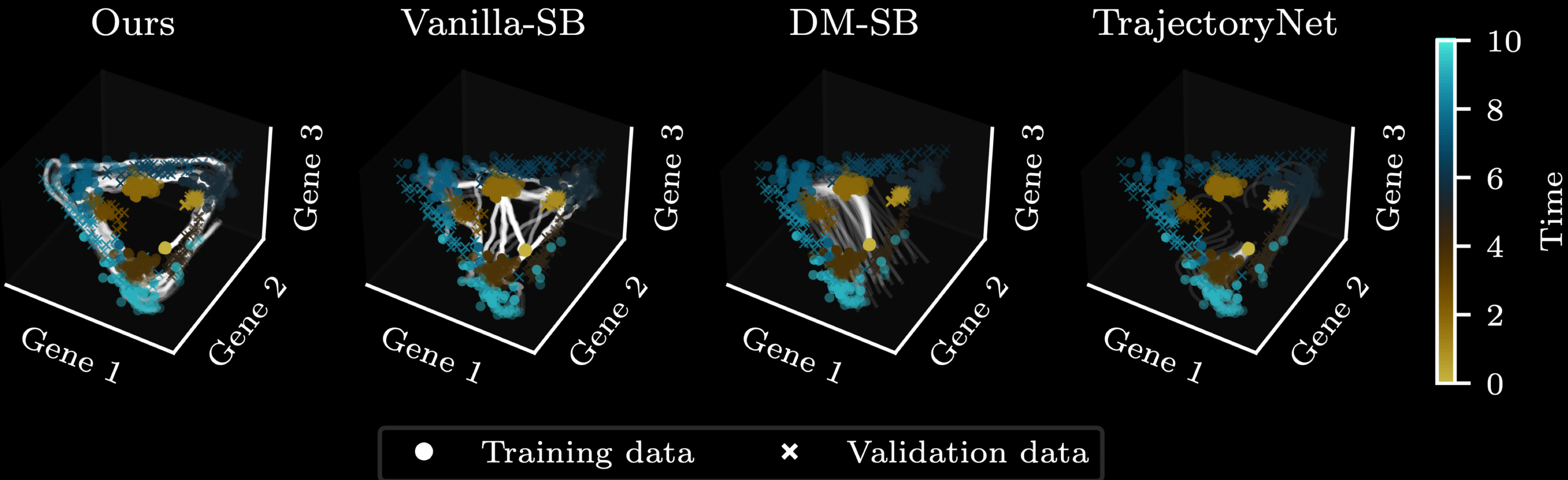
EXPERIMENT 2 : REPRESSILATOR

- The repressilator is a synthetic gene regulatory network that functions as a biological oscillator → three genes that inhibit each other in a cyclic manner



EXPERIMENT 2 : REPRESSILATOR

- The repressilator is a synthetic gene regulatory network that functions as a biological oscillator → three genes that inhibit each other in a cyclic manner



OUTLINE

- Problem Setup using single cell example
- Background: SDEs & Schrödinger Bridges
- SBIRR : Multimarginal SB with iterative reference refinement (AISTATS 2025, oral)
- SnapMMD : Forecasting Stochastic Dynamics using MMD (Submitted)
- Summary & Future Work

IS THIS ALL WE NEED?

- Problem: KL divergence well-defined if two underlying SDEs have

IS THIS ALL WE NEED?

- Problem: KL divergence well-defined if two underlying SDEs have
① same volatility

IS THIS ALL WE NEED?

- Problem : KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon

IS THIS ALL WE NEED?

- Problem: KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!

IS THIS ALL WE NEED?

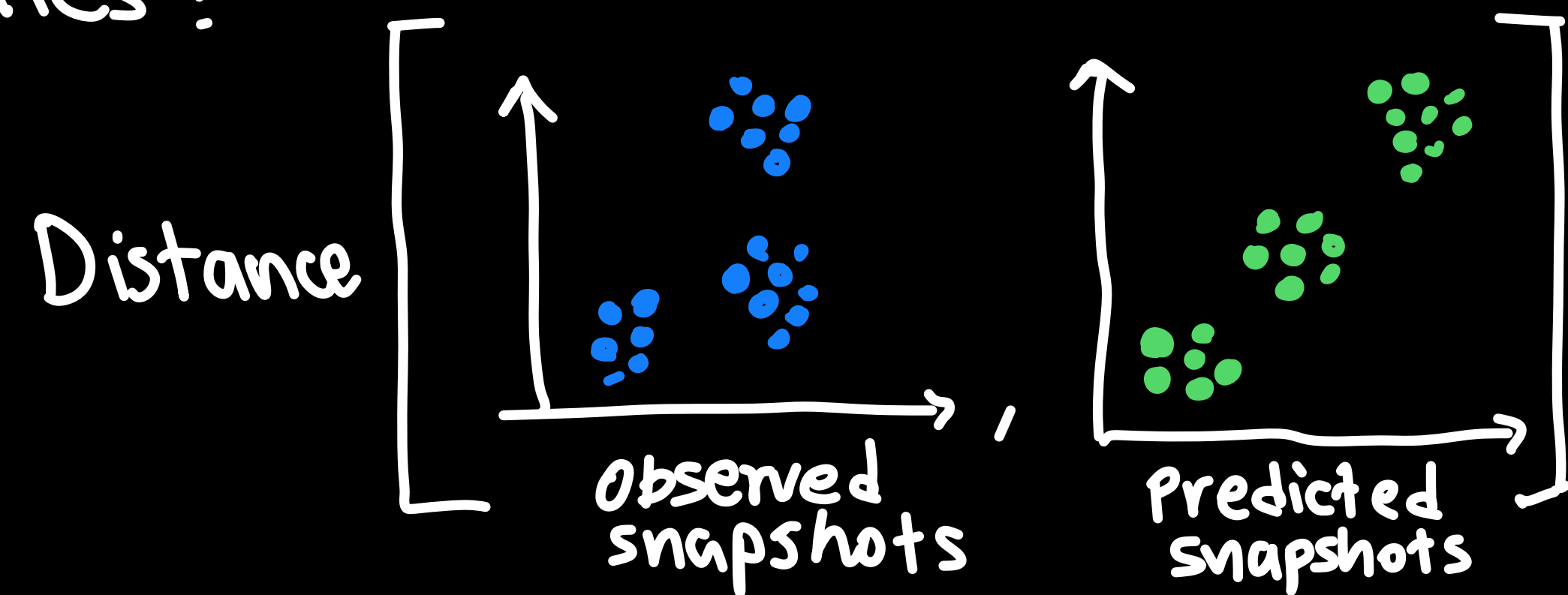
- Problem: KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?

IS THIS ALL WE NEED?

- Problem : KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?
- Idea : find model within a family
"closest" to observed snapshots

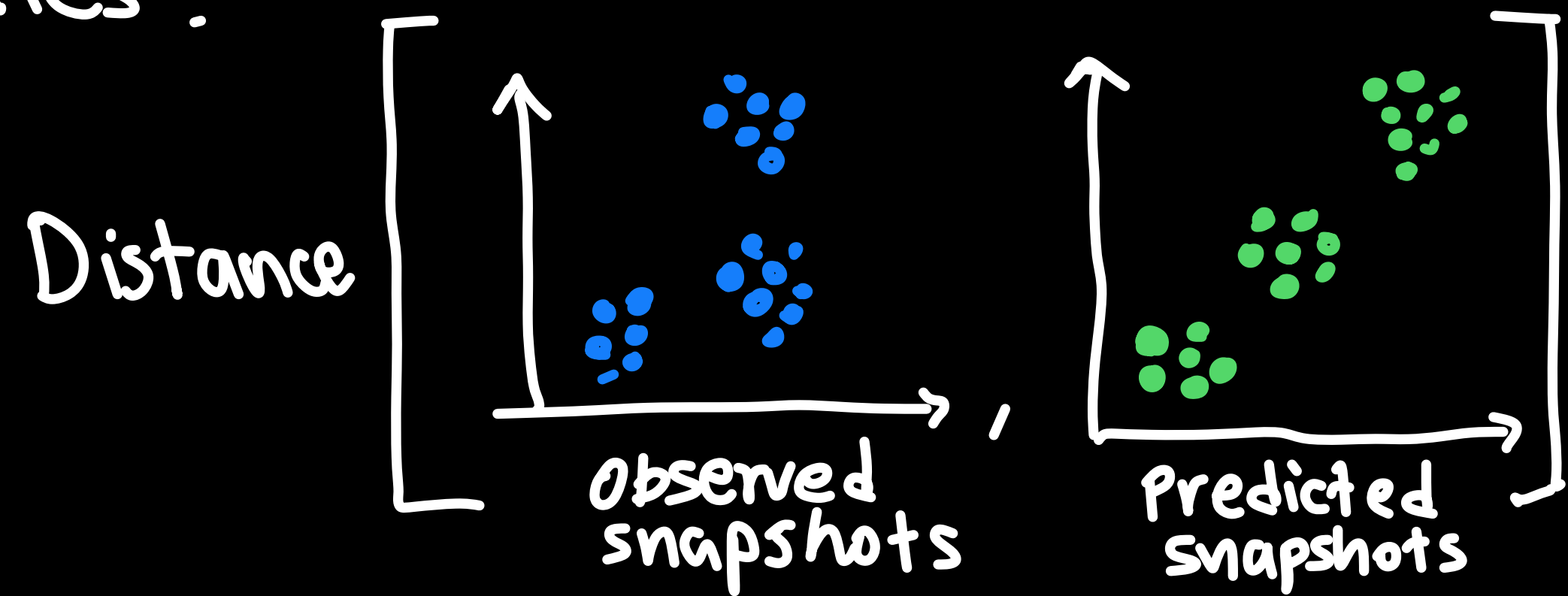
IS THIS ALL WE NEED?

- Problem: KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?
- Idea: find model within a family
"closest" to observed snapshots



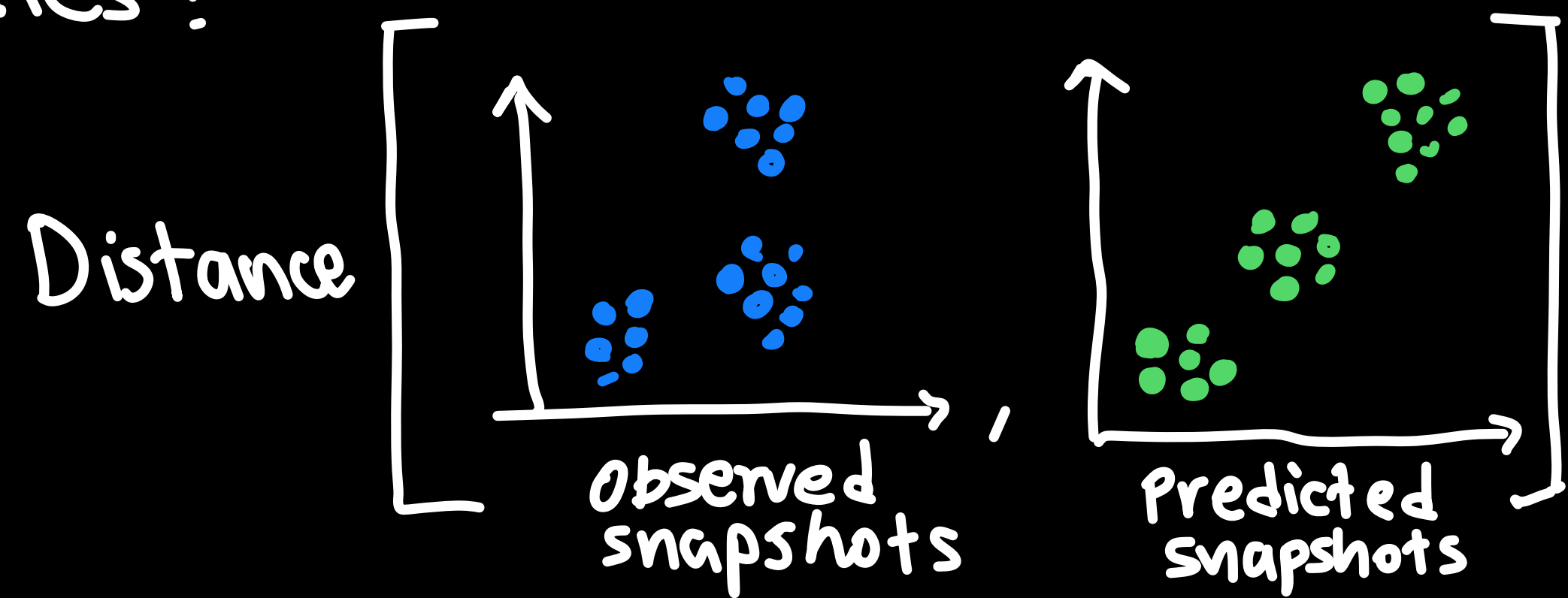
IS THIS ALL WE NEED?

- Problem : KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?
- Idea : find model within a family
"closest" to observed snapshots
- In practice : use Maximum Mean Discrepancy
→ computationally easy + "least-squares" interpretation



IS THIS ALL WE NEED?

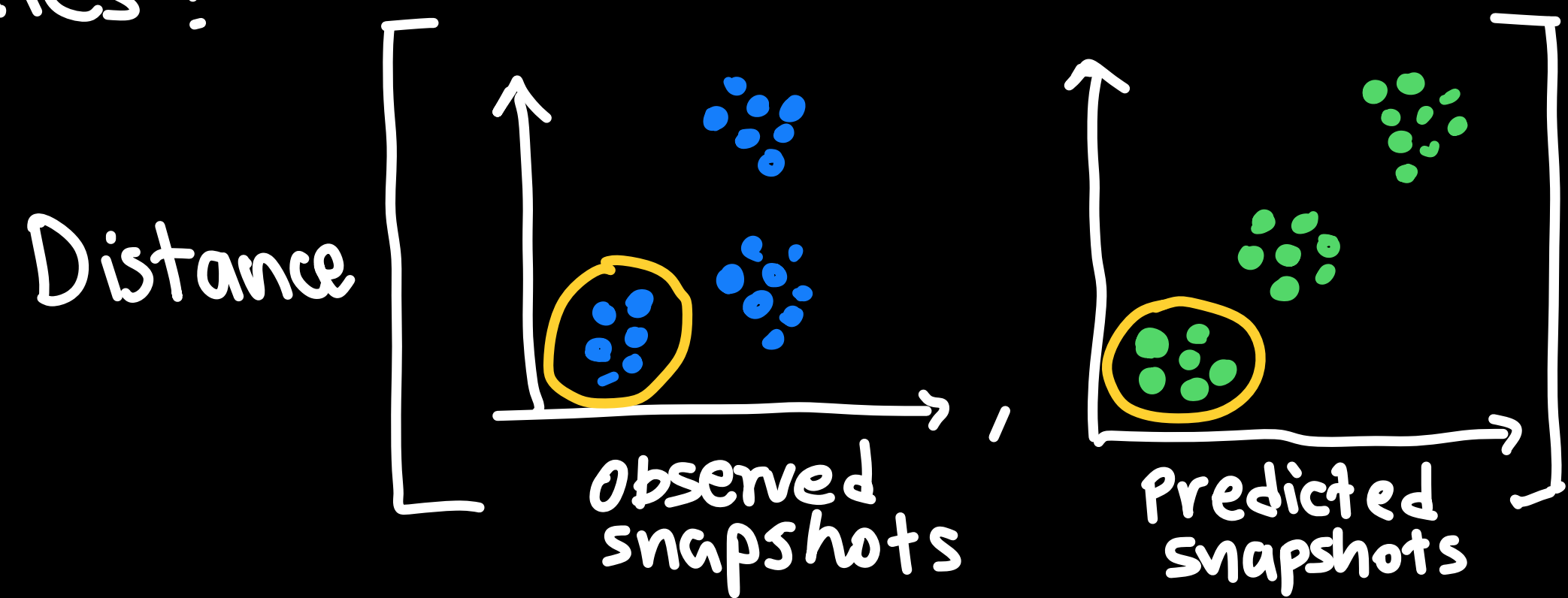
- Problem: KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?
- Idea: find model within a family
"closest" to observed snapshots
- In practice: use Maximum Mean Discrepancy
→ computationally easy + "least-squares" interpretation



$$\text{MMD}^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right]$$

IS THIS ALL WE NEED?

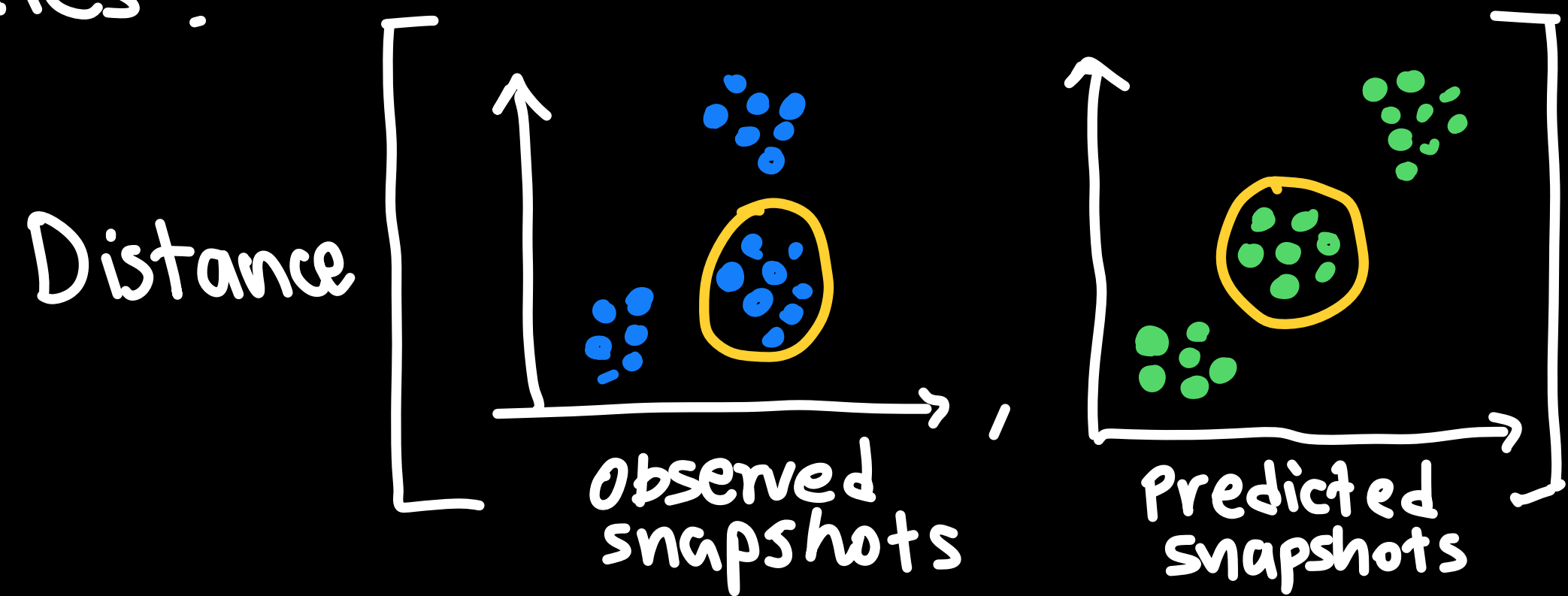
- Problem: KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?
- Idea: find model within a family "closest" to observed snapshots
- In practice: use Maximum Mean Discrepancy
→ computationally easy + "least-squares" interpretation



$$\Pi \Pi^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right] = w_1 \cdot \Pi \Pi^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right]$$

IS THIS ALL WE NEED?

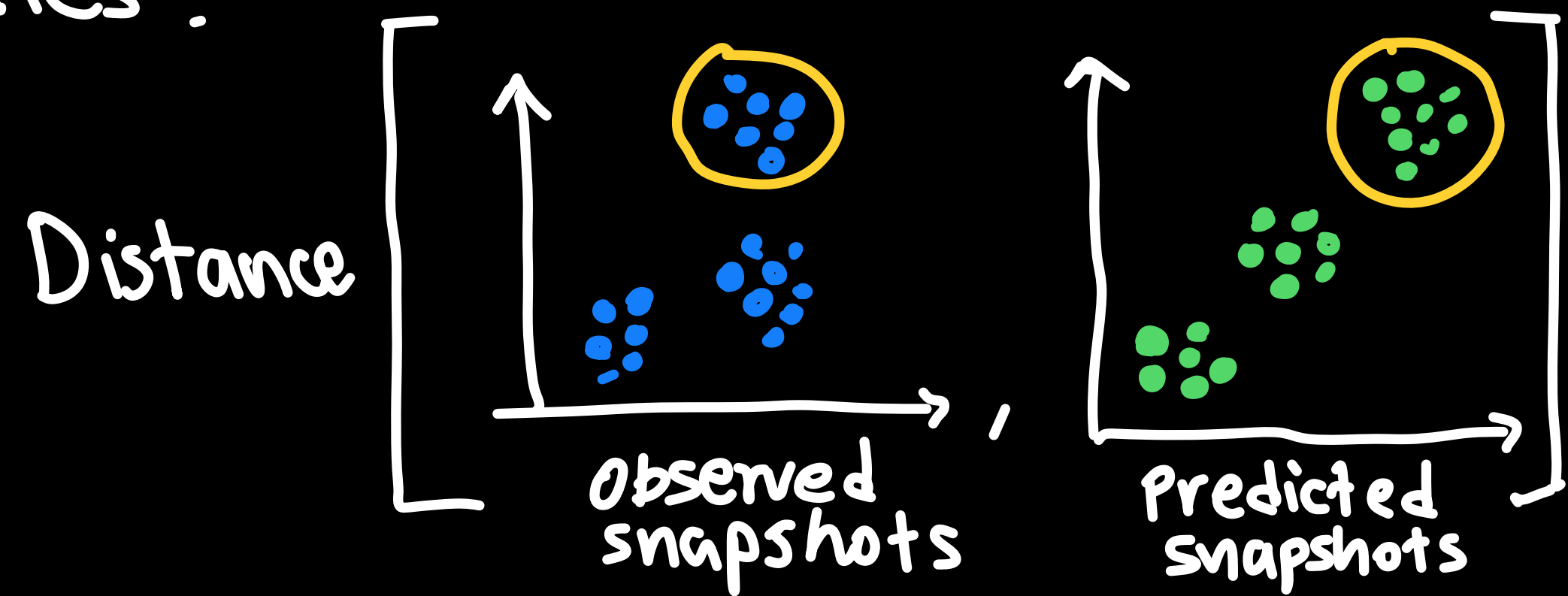
- Problem: KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?
- Idea: find model within a family "closest" to observed snapshots
- In practice: use Maximum Mean Discrepancy
→ computationally easy + "least-squares" interpretation



$$\pi\pi^2 \left[\begin{array}{c} \uparrow \\ \text{OBS} \end{array}, \begin{array}{c} \uparrow \\ \text{PRED} \end{array} \right] = w_1 \cdot \pi\pi^2 \left[\begin{array}{c} \uparrow \\ \text{OBS} \end{array}, \begin{array}{c} \uparrow \\ \text{PRED} \end{array} \right] + w_2 \cdot \pi\pi^2 \left[\begin{array}{c} \uparrow \\ \text{OBS} \end{array}, \begin{array}{c} \uparrow \\ \text{PRED} \end{array} \right]$$

IS THIS ALL WE NEED?

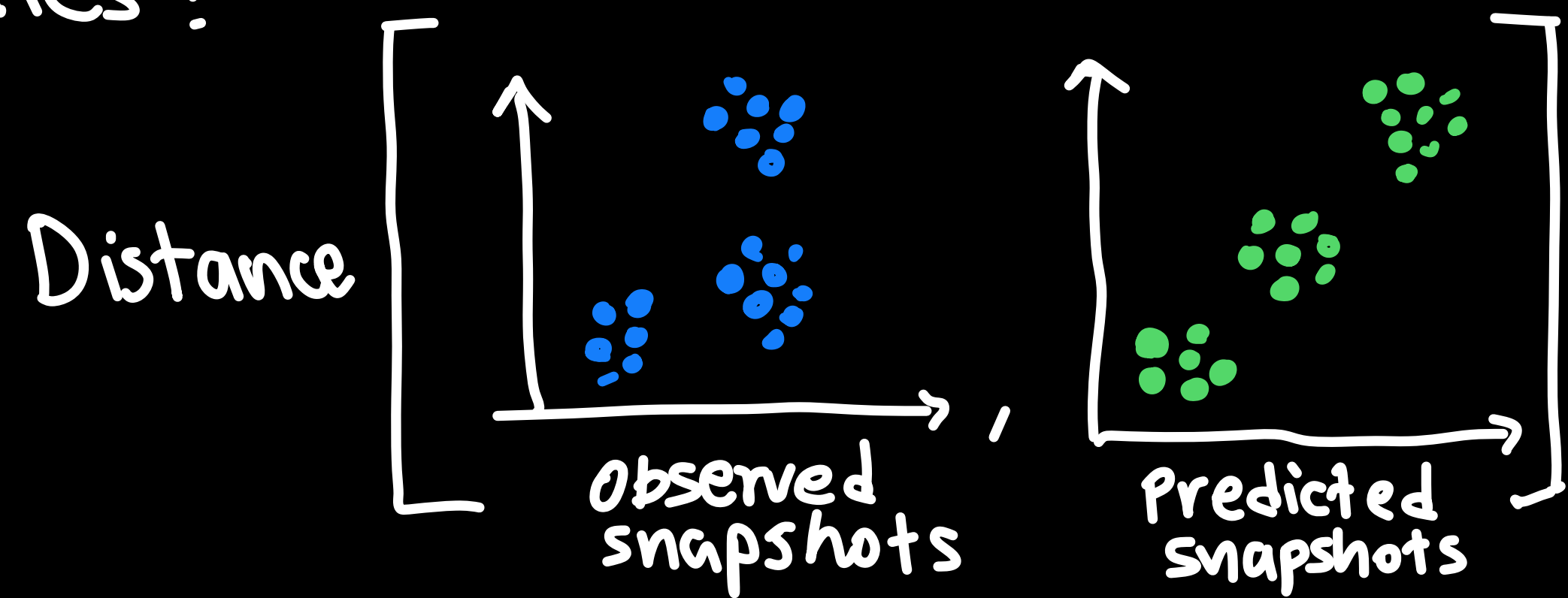
- Problem: KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?
- Idea: find model within a family "closest" to observed snapshots
- In practice: use Maximum Mean Discrepancy
→ computationally easy + "least-squares" interpretation



$$\pi \pi 0^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right] = w_1 \cdot \pi \pi 0^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right] + w_2 \cdot \pi \pi 0^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right] + w_3 \cdot \pi \pi 0^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right]$$

IS THIS ALL WE NEED?

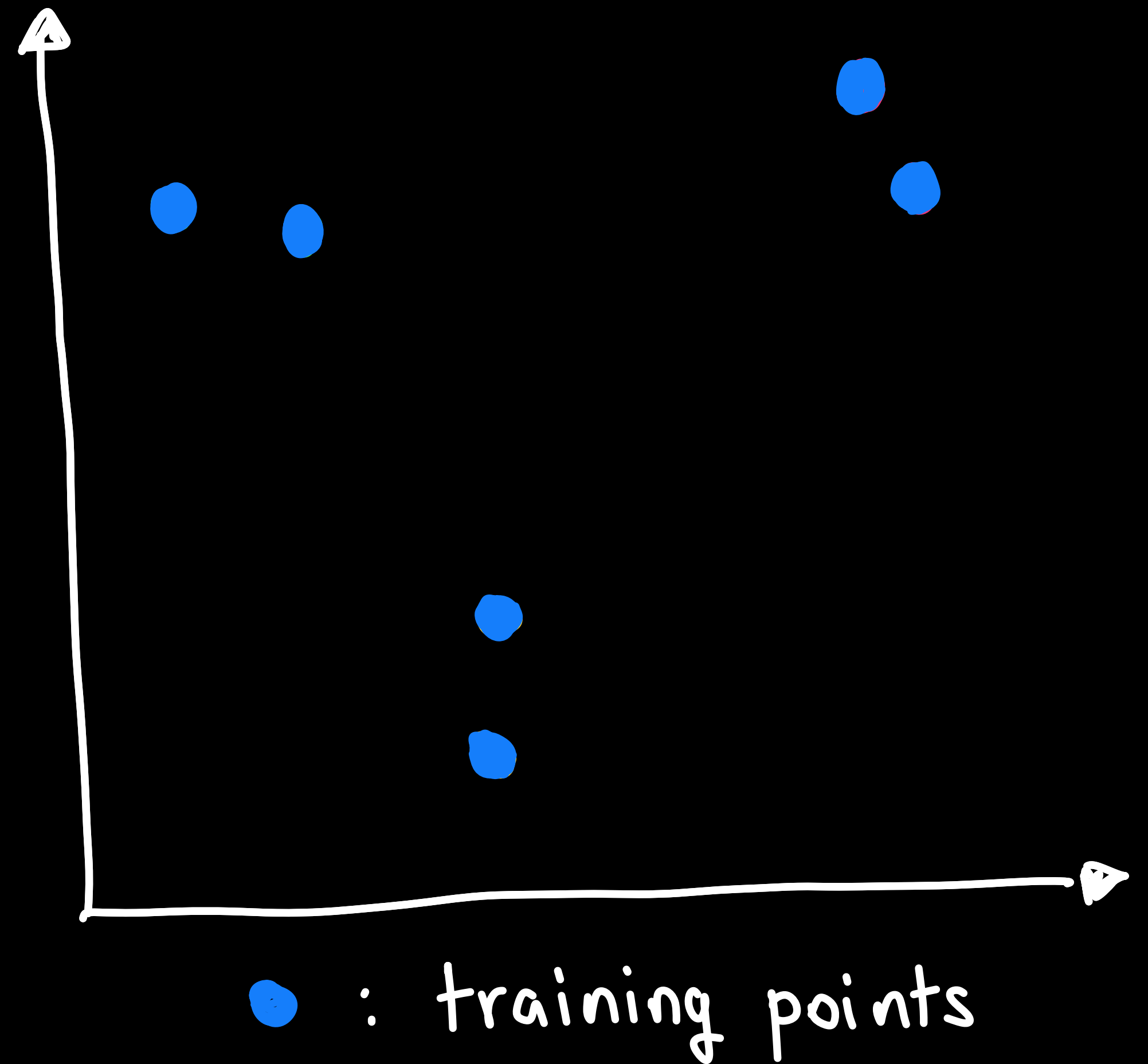
- Problem: KL divergence well-defined if two underlying SDEs have
 - ① same volatility
 - ② same time horizon→ unrealistic for some scientific applications!
- Since data are in "marginal" space, no need to compare full trajectories
→ can we just match marginal quantities?
- Idea: find model within a family
"closest" to observed snapshots
- In practice: use Maximum Mean Discrepancy
→ computationally easy + "least-squares" interpretation



$$\pi \pi 0^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right] = w_1 \cdot \pi \pi 0^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right] + w_2 \cdot \pi \pi 0^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right] + w_3 \cdot \pi \pi 0^2 \left[\begin{array}{c} \text{OBS} \\ \text{PRED} \end{array} \right]$$

EXPERIMENTAL SETUP (REVISITED)

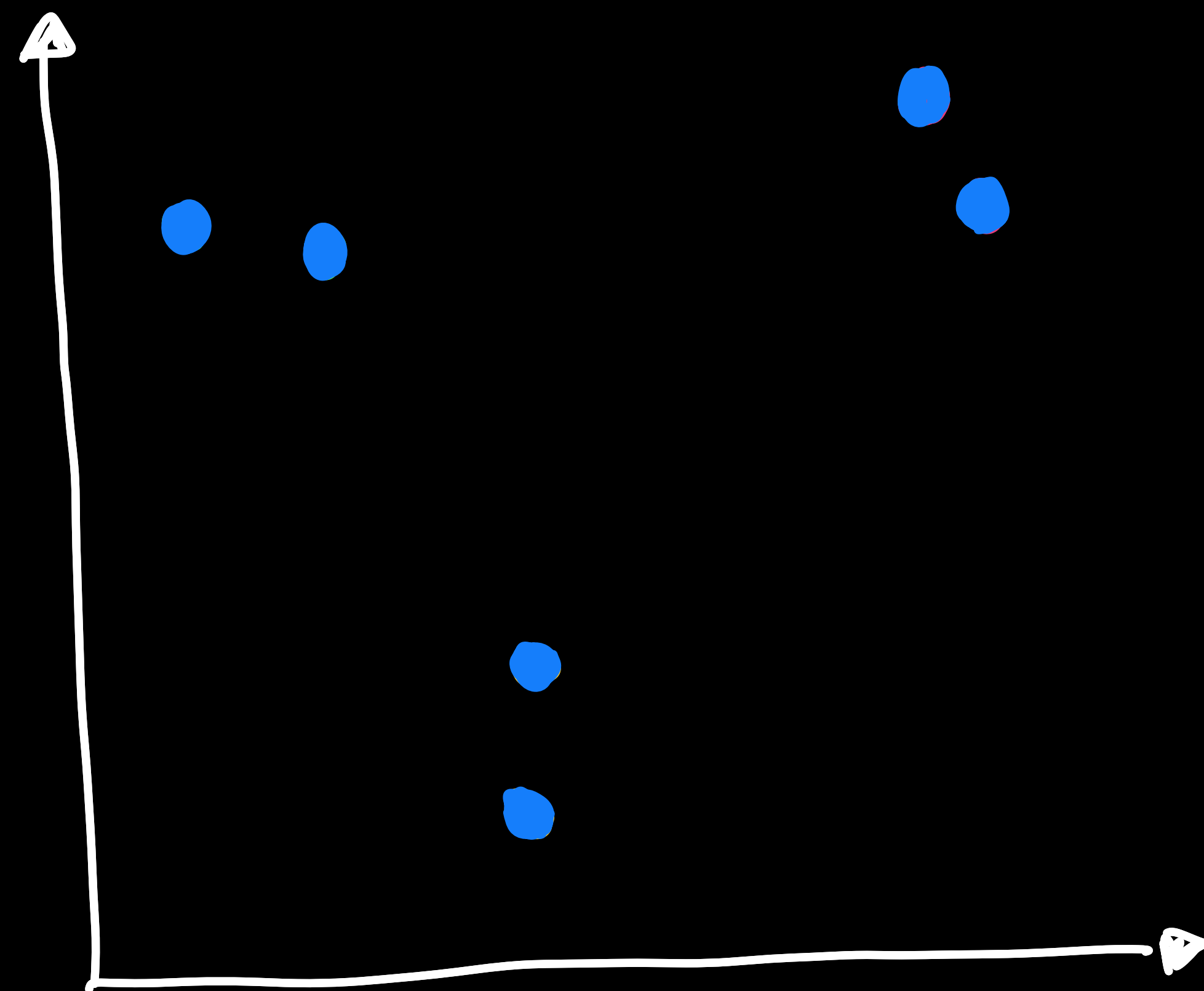
- Same synthetic and real data experiments with same natural reference families



EXPERIMENTAL SETUP (REVISITED)

xx

- Same synthetic and real data experiments with same natural reference families
- To test if what we learnt is good: retain one-step-ahead validation snapshot

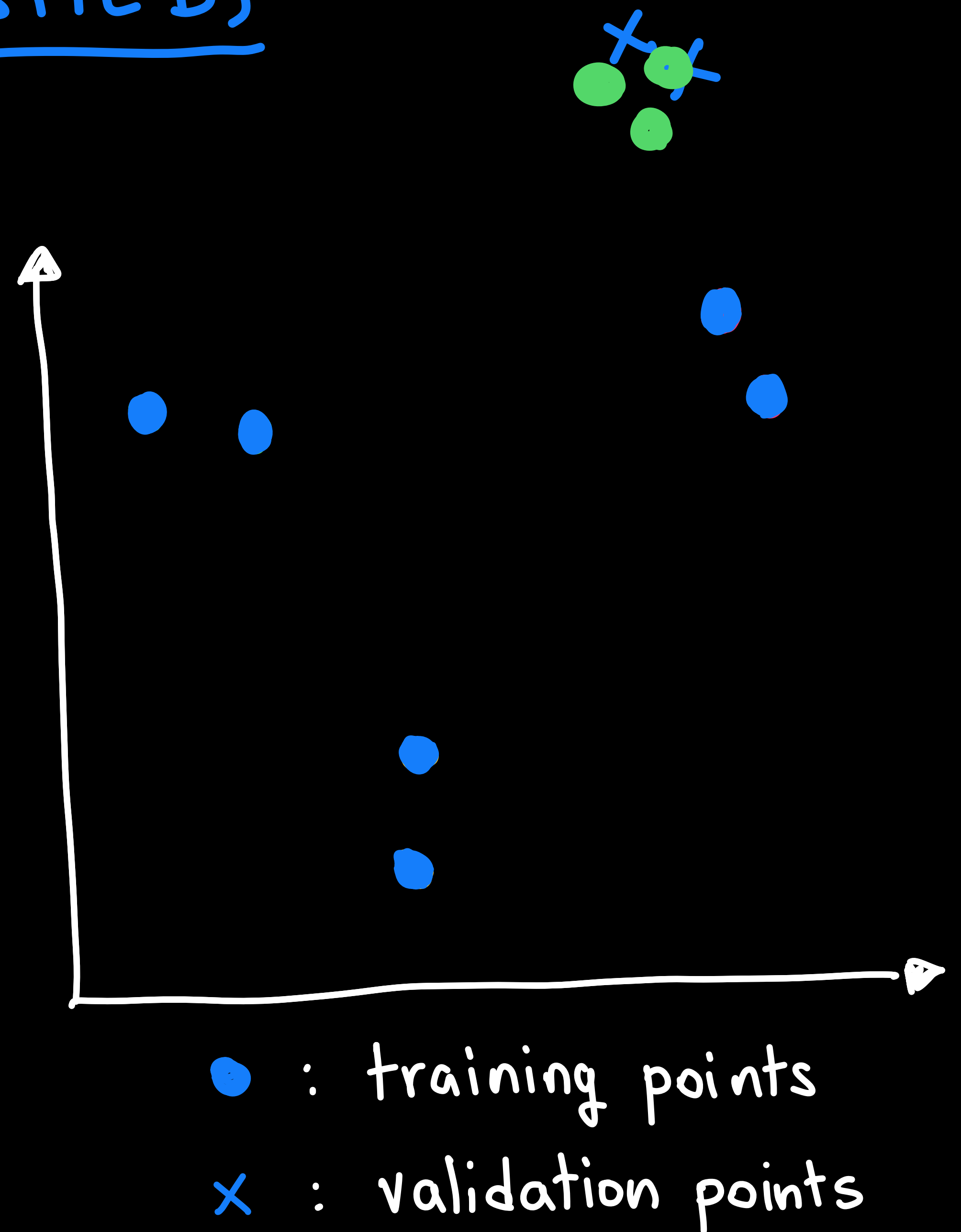


● : training points

x : validation points

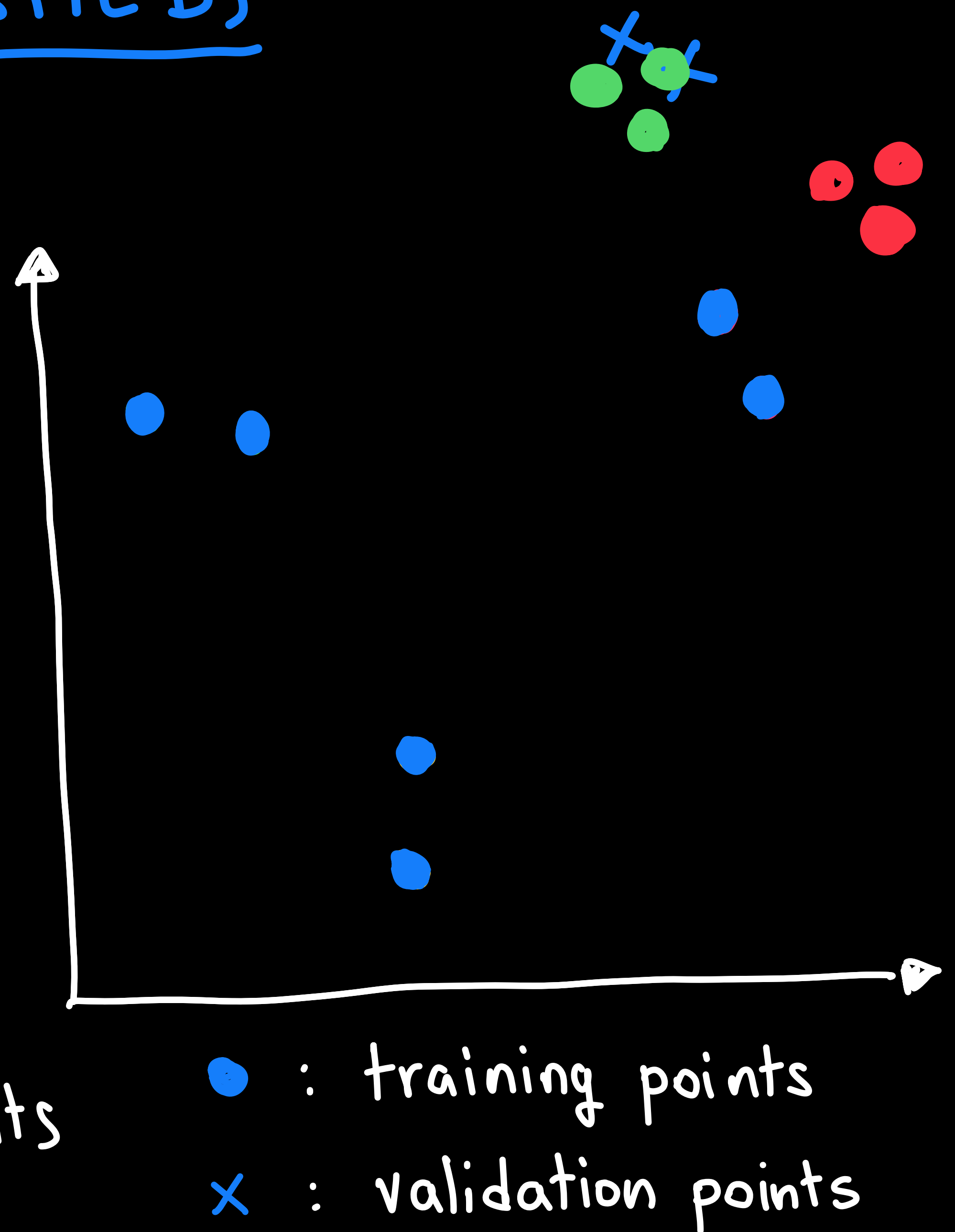
EXPERIMENTAL SETUP (REVISITED)

- Same synthetic and real data experiments with same natural reference families
 - To test if what we learnt is good: retain one-step-ahead validation snapshot
- ✓ **SUCCESS**: if one-step ahead forecast states are close to validation points



EXPERIMENTAL SETUP (REVISITED)

- Same synthetic and real data experiments with same natural reference families
 - To test if what we learnt is good: retain one-step-ahead validation snapshot
- ✓ **SUCCESS**: if one-step ahead forecast states are close to validation points
- ✗ **FAILURE**: if one-step ahead forecast states are not close to validation points



EXPERIMENT 1 : GULF of MEXICO DRIFTERS

- As before: ① simulate particles around a vortex-like feature

EXPERIMENT 1 : GULF of MEXICO DRIFTERS

- As before: ① simulate particles around a vortex-like feature
② Use reference parametric family of vortex with unknown center, direction, and shape

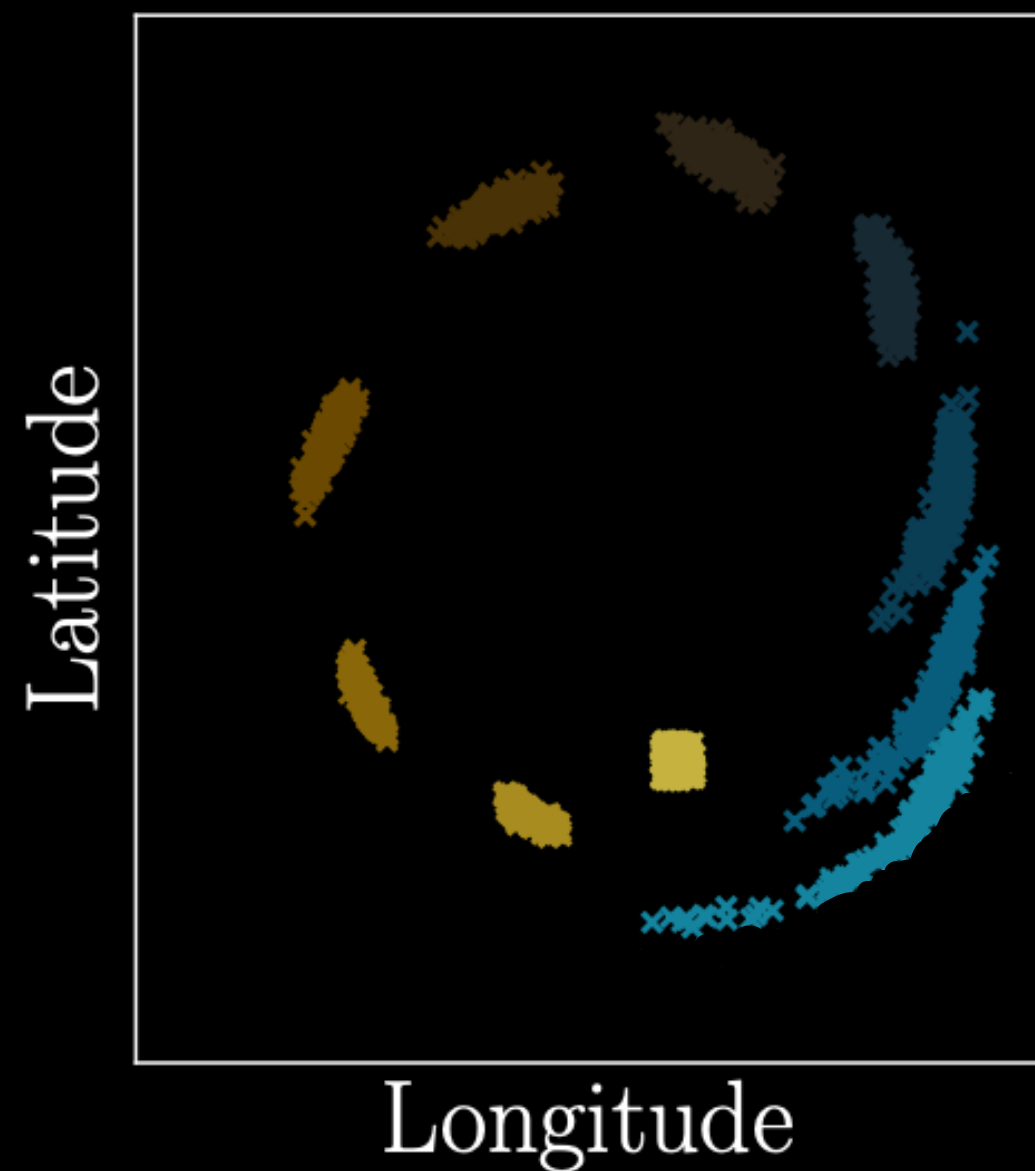
EXPERIMENT 1: GULF of MEXICO DRIFTERS

- As before: ① simulate particles around a vortex-like feature
② Use reference parametric family of vortex with unknown center, direction, and shape
- Here: volatility is an unknown parameter that we also learn

EXPERIMENT 1 : GULF of MEXICO DRIFTERS

- As before: ① simulate particles around a vortex-like feature
② Use reference parametric family of vortex with unknown center, direction, and shape
- Here: volatility is an unknown parameter that we also learn

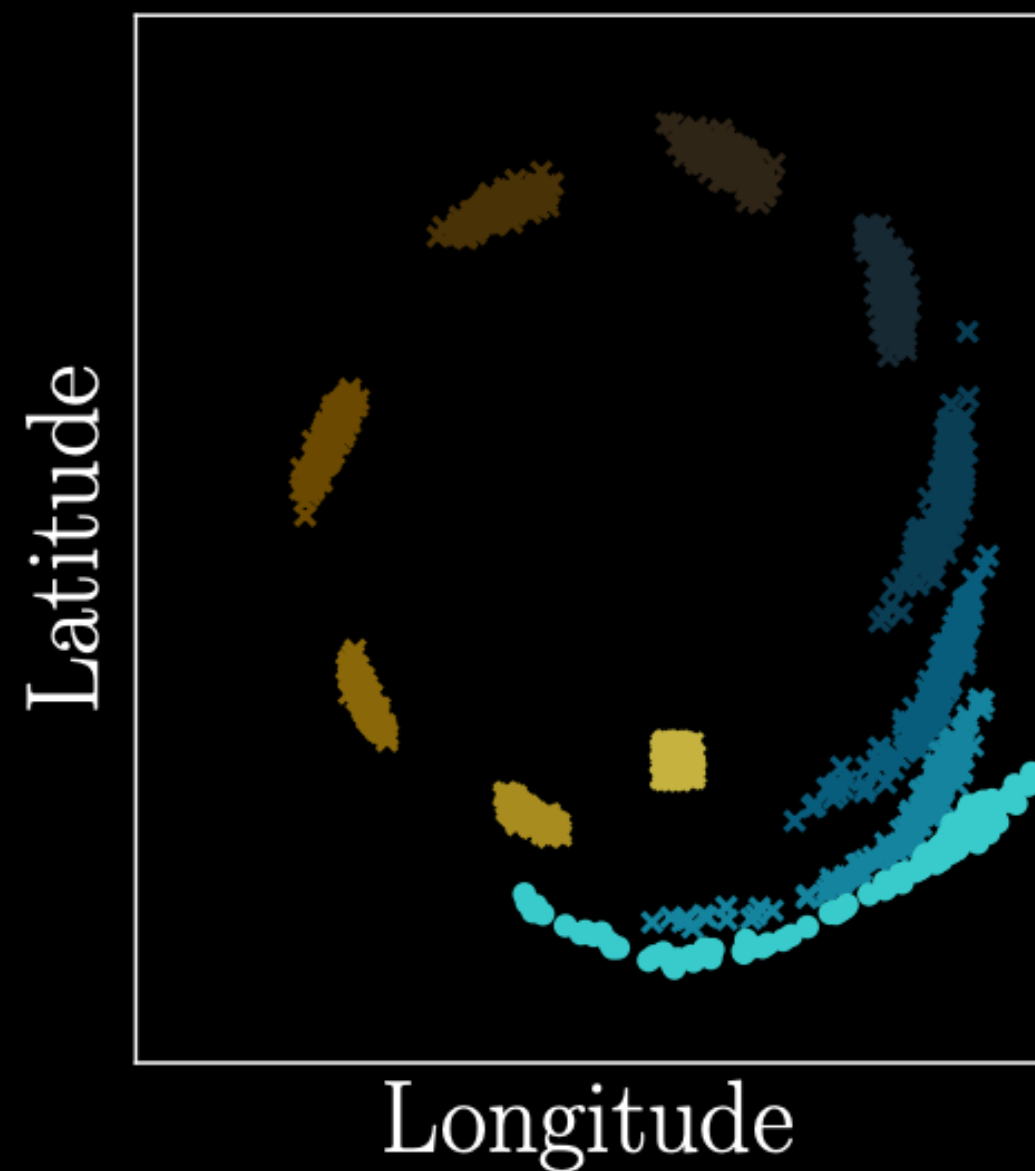
Ground Truth



EXPERIMENT 1 : GULF of MEXICO DRIFTERS

- As before: ① simulate particles around a vortex-like feature
② Use reference parametric family of vortex with unknown center, direction, and shape
- Here: volatility is an unknown parameter that we also learn

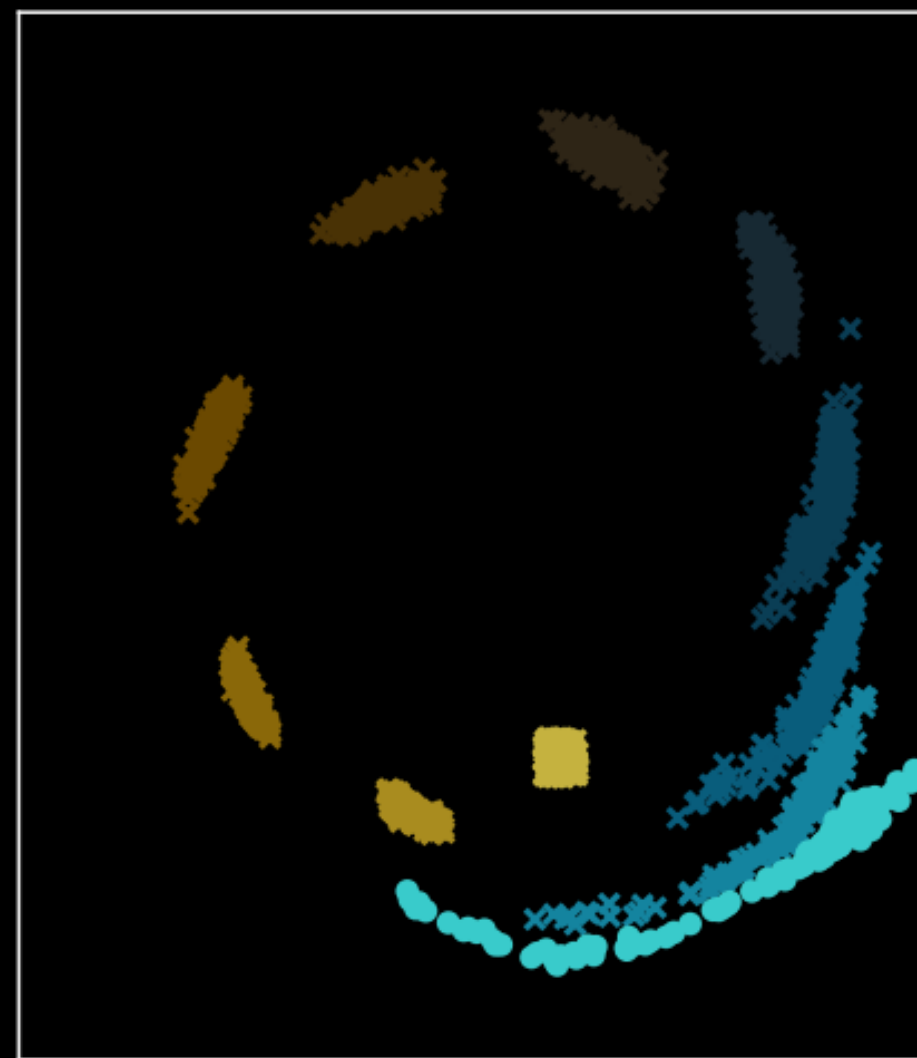
Ground Truth



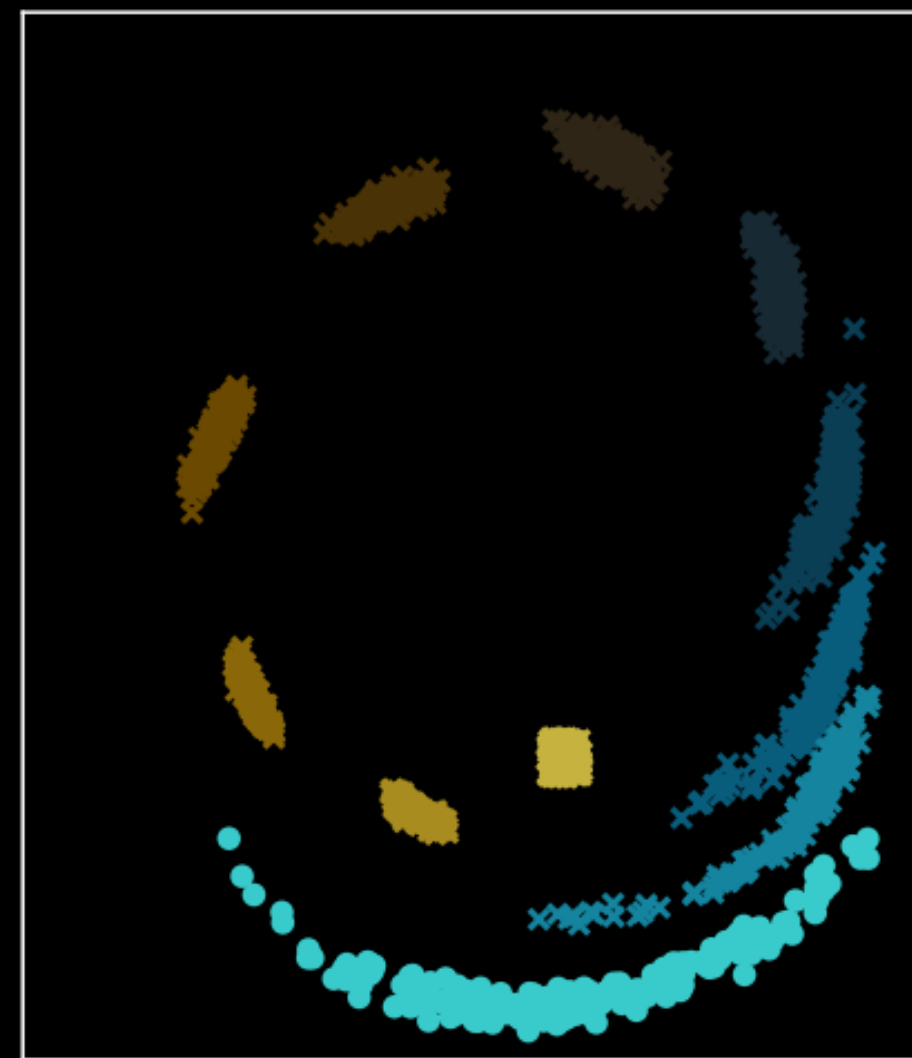
EXPERIMENT 1 : GULF of MEXICO DRIFTERS

- As before: ① simulate particles around a vortex-like feature
② Use reference parametric family of vortex with unknown center, direction, and shape
- Here: volatility is an unknown parameter that we also learn

Ground Truth

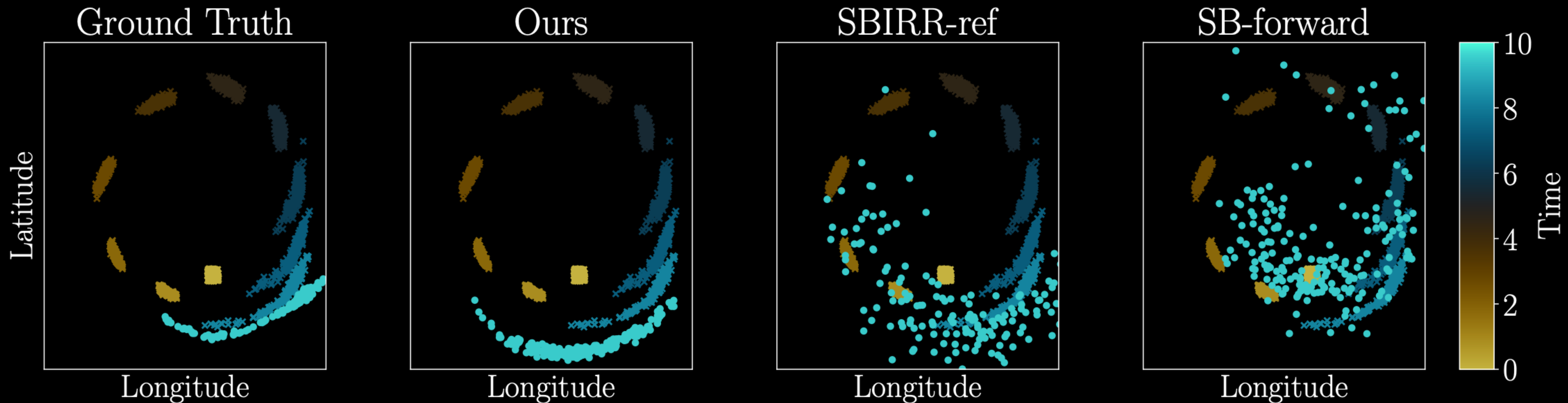


Ours



EXPERIMENT 1: GULF of MEXICO DRIFTERS

- As before: ① simulate particles around a vortex-like feature
② Use reference parametric family of vortex with unknown center, direction, and shape
- Here: volatility is an unknown parameter that we also learn



EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

$$dX_t = (\overset{\substack{\text{maximum} \\ \text{production rate}}}{M} \cdot f_{\theta}(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

maximum production rate $\nearrow$

unknown activation function, $f_\theta: \mathbb{R}_+^3 \rightarrow [0,1]^3$
(Multi-layer perceptron)

EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

maximum production rate $\nearrow$

unknown activation function, $f_\theta: \mathbb{R}_+^3 \rightarrow [0,1]^3$
(Multi-layer perceptron) $\nwarrow$

degradation rate $\nwarrow$

EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

maximum production rate $\nearrow$

unknown activation function, $f_\theta: \mathbb{R}_+^3 \rightarrow [0,1]^3$
(Multi-layer perceptron) $\nwarrow$

degradation rate $\nwarrow$

volatilities $\nwarrow$

EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

$$dX_t = (M \cdot f_{\theta}(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

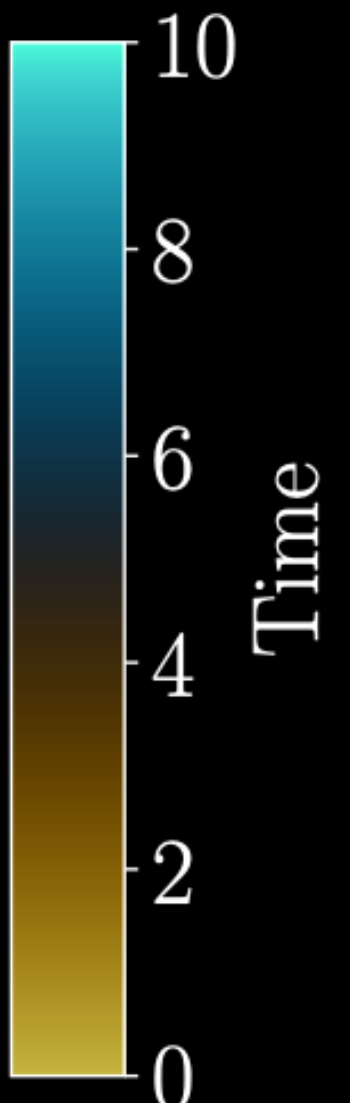
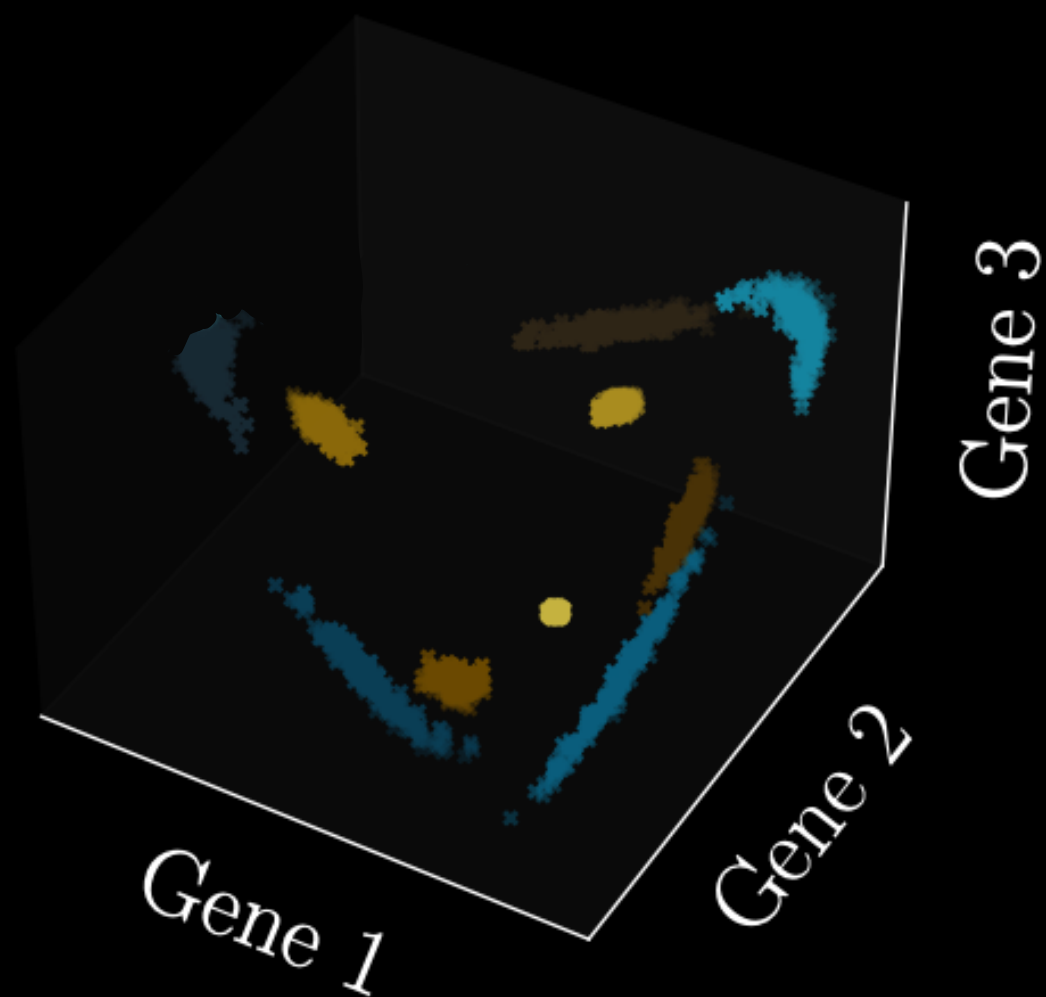
maximum production rate $\nearrow$

unknown activation function, $f_{\theta}: \mathbb{R}_+^3 \rightarrow [0,1]^3$ (Multi-layer perceptron) $\nwarrow$

degradation rate $\nwarrow$

volatilities $\nwarrow$

Ground Truth



EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

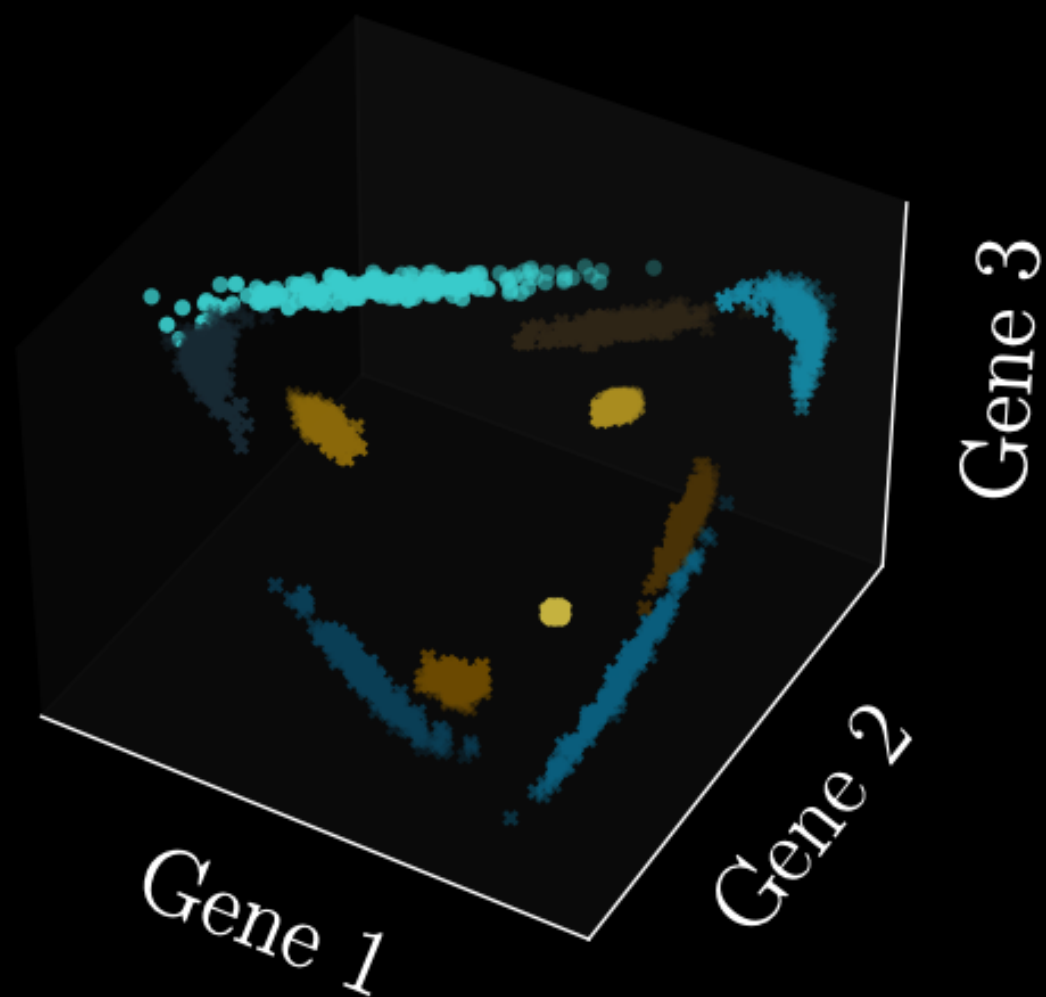
maximum production rate $\nearrow$

unknown activation function, $f_\theta: \mathbb{R}_+^3 \rightarrow [0,1]^3$ (Multi-layer perceptron) $\nwarrow$

degradation rate $\nwarrow$

volatilities $\nwarrow$

Ground Truth



EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

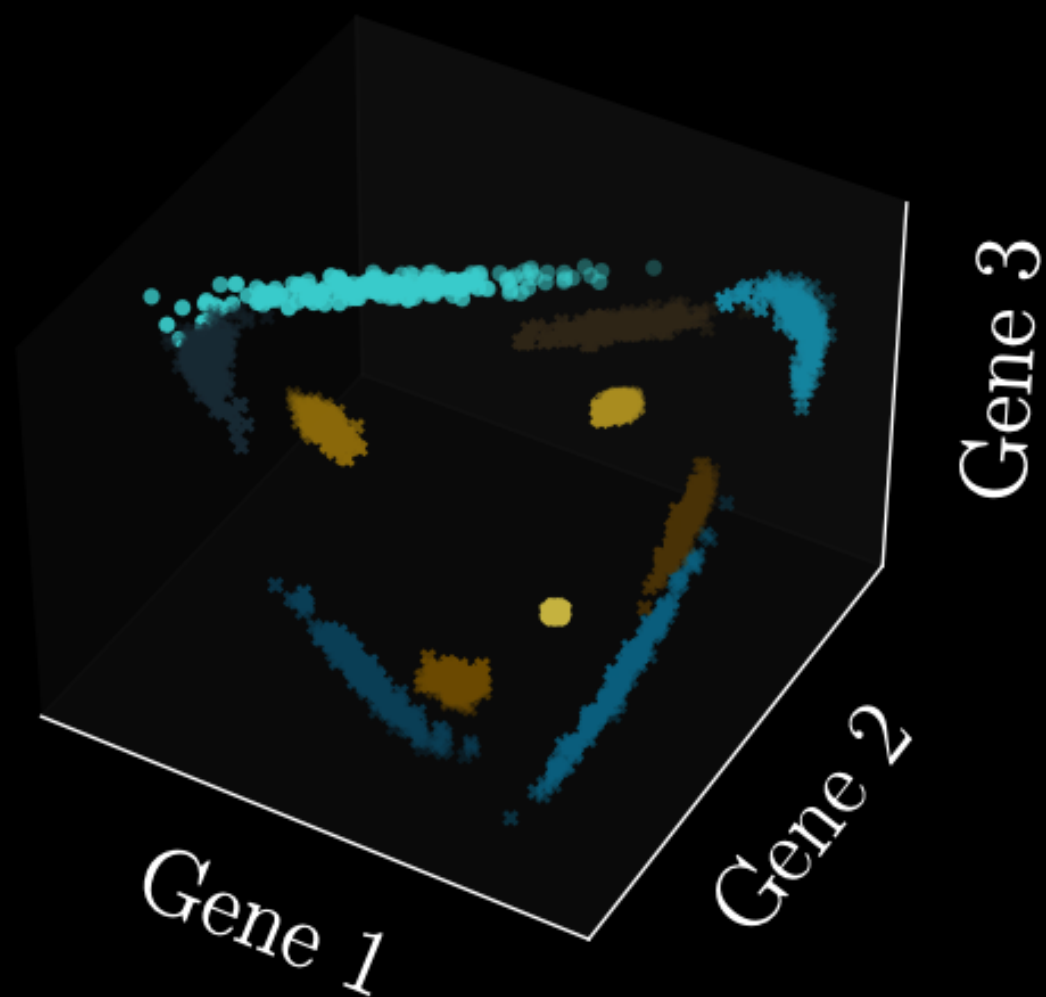
maximum production rate $\nearrow$ (points to M)

unknown activation function, $f_\theta: \mathbb{R}_+^3 \rightarrow [0,1]^3$ (Multi-layer perceptron) $\nwarrow$ (points to f_θ)

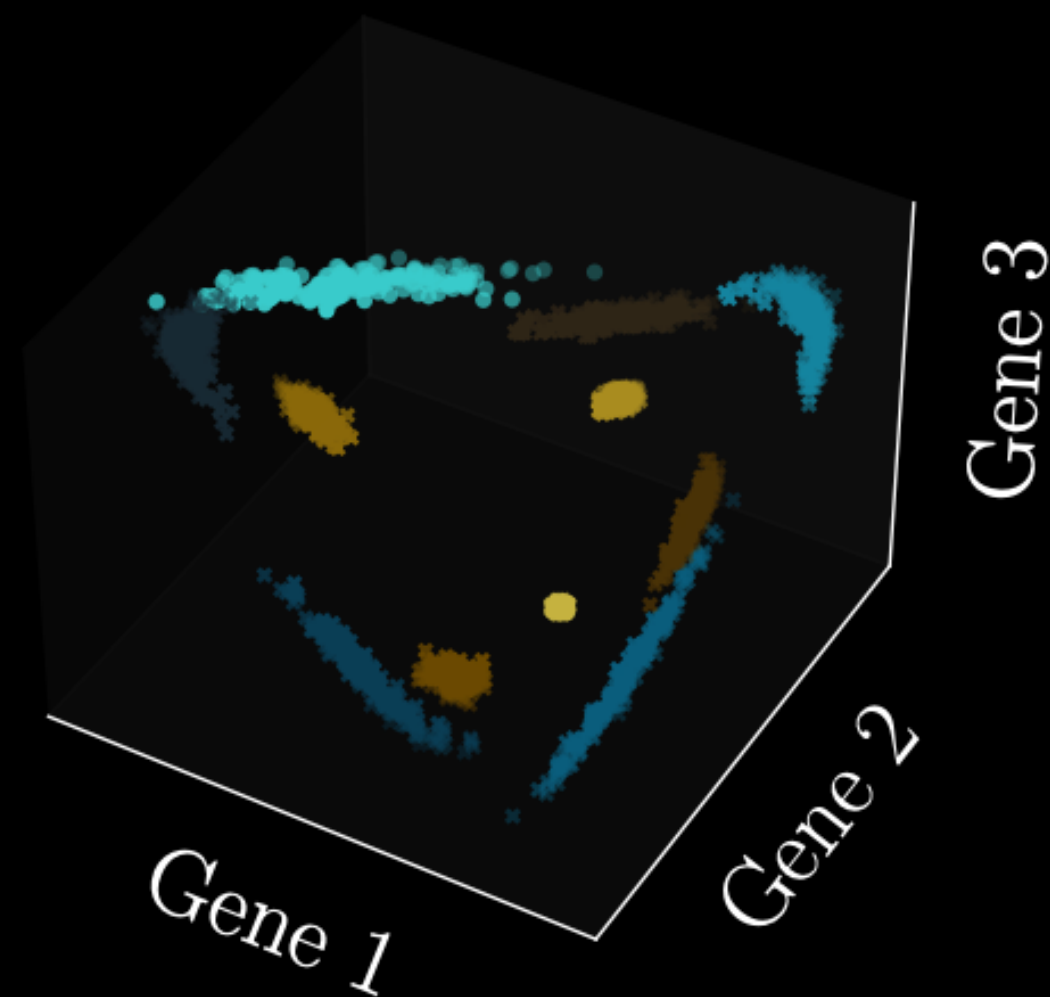
degradation rate $\nwarrow$ (points to L)

volatilities $\nwarrow$ (points to G)

Ground Truth



Ours



EXPERIMENT 2 : REPRESSILATOR

- Here, we have unknown volatility $\rightarrow$ we use a semiparametric model family

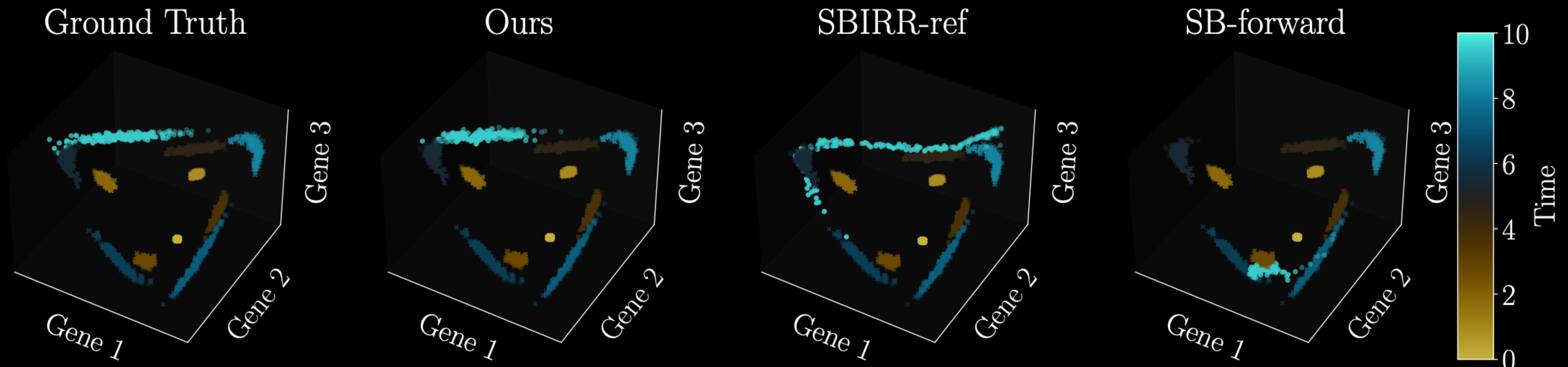
$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$

maximum production rate $\nearrow$ (points to M)

unknown activation function, $f_\theta: \mathbb{R}_+^3 \rightarrow [0,1]^3$ (Multi-layer perceptron) $\nwarrow$ (points to $f_\theta(X_t)$)

degradation rate $\nwarrow$ (points to $L \cdot X_t$)

volatilities $\nwarrow$ (points to $G \cdot \text{diag}(X_t) dW_t$)

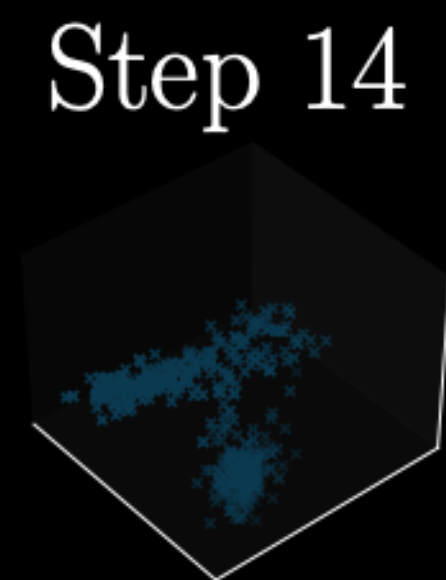
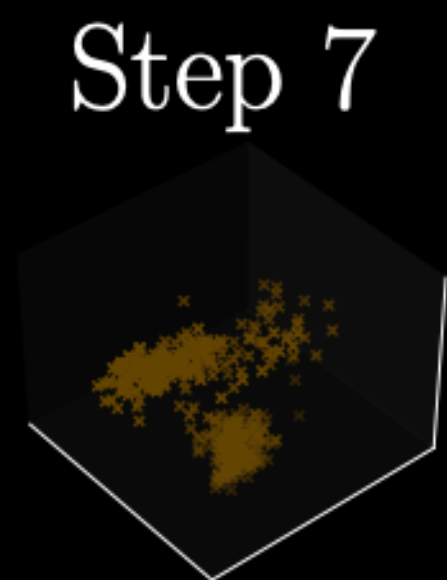
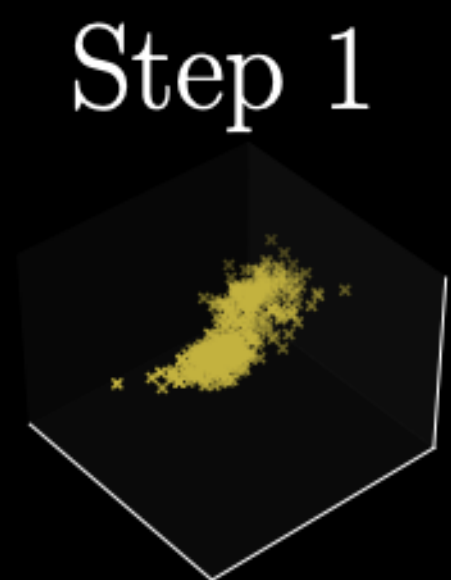


EXPERIMENT 3 : SINGLE-CELL DATASET

- Dataset: T cell-mediated immune activation in peripheral blood mononuclear cells (PBMCs) by Jiang et al. 2024

EXPERIMENT 3 : SINGLE-CELL DATASET

- Dataset: T cell-mediated immune activation in peripheral blood mononuclear cells (PBMCs) by Jiang et al. 2024
- 20 training points (recorded every hour), 30 dimensional projection



EXPERIMENT 3 : SINGLE-CELL DATASET

- Dataset: T cell-mediated immune activation in peripheral blood mononuclear cells (PBMCs) by Jiang et al. 2024
- 20 training points (recorded every hour), 30 dimensional projection
- Model family: same as the repressilator experiment

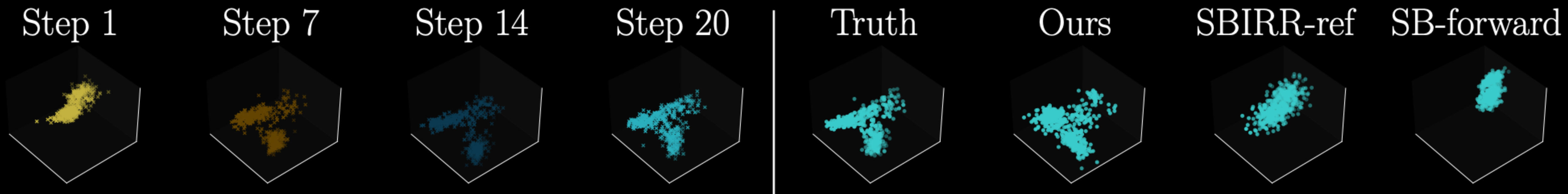
$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$



EXPERIMENT 3 : SINGLE-CELL DATASET

- Dataset: T cell-mediated immune activation in peripheral blood mononuclear cells (PBMCs) by Jiang et al. 2024
- 20 training points (recorded every hour), 30 dimensional projection
- Model family: same as the repressilator experiment

$$dX_t = (M \cdot f_\theta(X_t) - L \cdot X_t) dt + G \cdot \text{diag}(X_t) dW_t$$



OUTLINE

- Problem Setup using single cell example
- Background: SDEs & Schrödinger Bridges
- SBIRR : Multimarginal SB with iterative reference refinement (AISTATS 2025, oral)
- SnapMMD : Forecasting Stochastic Dynamics using MMD (Submitted)
- Summary & Future Work

IN SUMMARY

We propose two new methods that (1) learn the unobserved trajectories from sample snapshots across multiple time points, (2) require specification only of a class of reference dynamics, and (3) can be used for interpolation (SB or SnapMMD) or forecasting (SnapMMD).

IN SUMMARY

We propose two new methods that (1) learn the unobserved trajectories from sample snapshots across multiple time points, (2) require specification only of a class of reference dynamics, and (3) can be used for interpolation (SB or SnapMMD) or forecasting (SnapMMD).

Future work

- Extend to backtracking + satellite images, e.g. for studying wildfires and oil spills

IN SUMMARY

We propose two new methods that (1) learn the unobserved trajectories from sample snapshots across multiple time points, (2) require specification only of a class of reference dynamics, and (3) can be used for interpolation (SB or SnapMMD) or forecasting (SnapMMD).

Future work

- Extend to backtracking + satellite images, e.g. for studying wildfires and oil spills
- Can we make a hybrid model with best of both worlds?

IN SUMMARY

We propose two new methods that (1) learn the unobserved trajectories from sample snapshots across multiple time points, (2) require specification only of a class of reference dynamics, and (3) can be used for interpolation (SB or SnapMMD) or forecasting (SnapMMD).

Future work

- Extend to backtracking + satellite images, e.g. for studying wildfires and oil spills
- Can we make a hybrid model with best of both worlds?
- What about identifiability?

IN SUMMARY

We propose two new methods that (1) learn the unobserved trajectories from sample snapshots across multiple time points, (2) require specification only of a class of reference dynamics, and (3) can be used for interpolation (SB or SnapMMD) or forecasting (SnapMMD).



Thank you!

Email : renb@mit.edu

Website : renatoberlinghieri.github.io

