Tracking Control in the Wasserstein Space

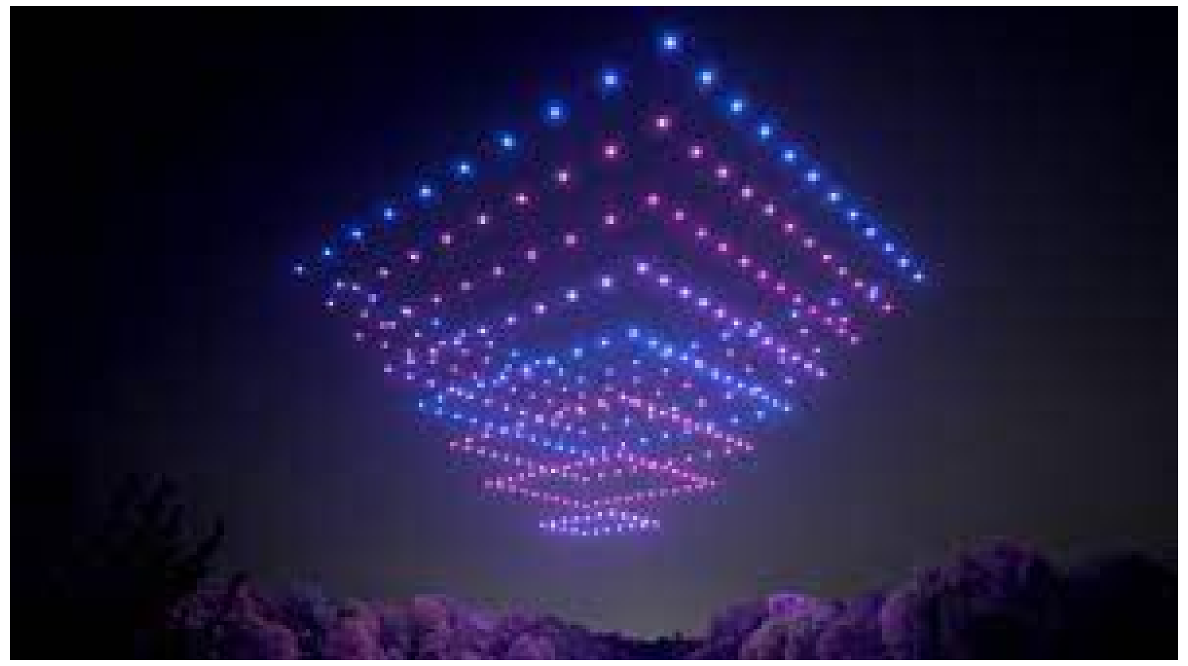
44th SoCal Control Workshop
November 1, 2024

Max Emerick (UC Santa Barbara, Advisor: Bassam Bamieh)

Many Applications for Autonomous Swarms

Emergency Response

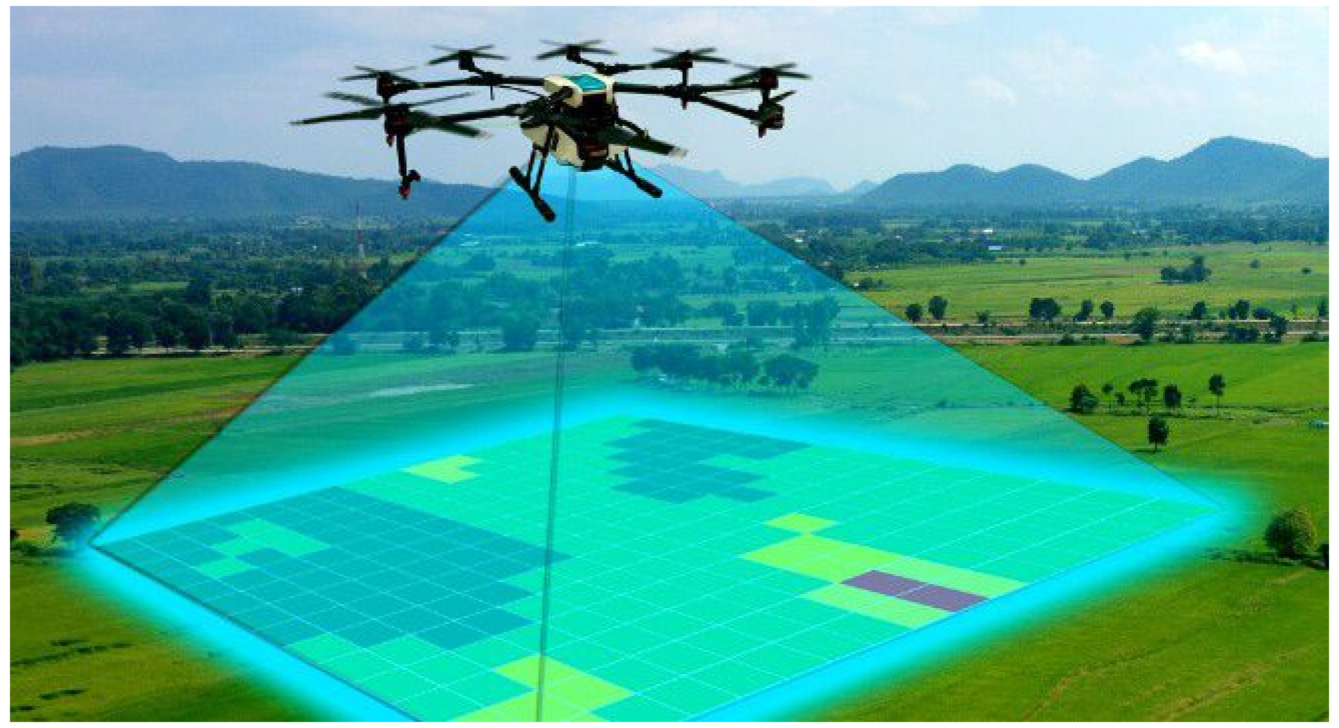
Logistics



Entertainment

Transportation

Defense



Data Collection

Motivation

The Problem in Focus:

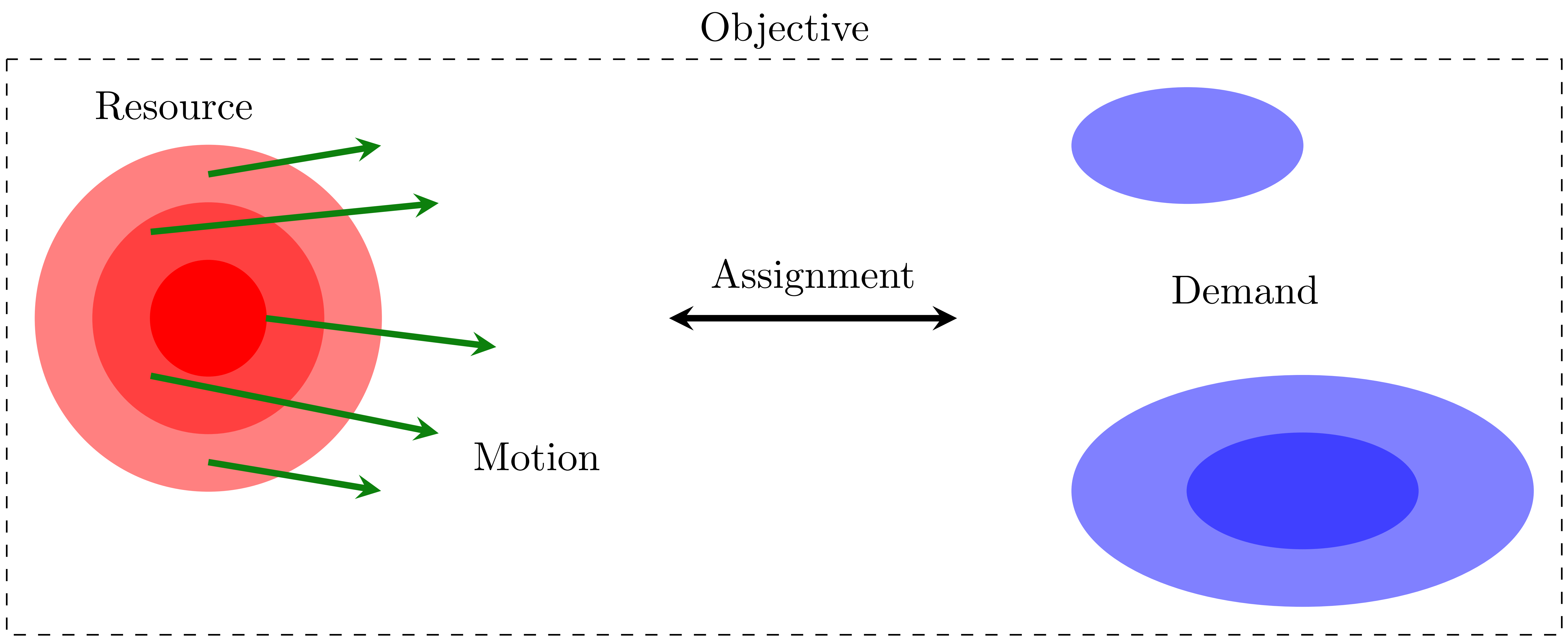
- Large swarms robust/efficient, but hard to model/control
- Want to develop theoretical foundations for design heuristics

Aim to Answer Questions:

- How should large swarms move and communicate?
- Which control architectures can achieve which behaviors?
- What are the attainable performance limits of these architectures?

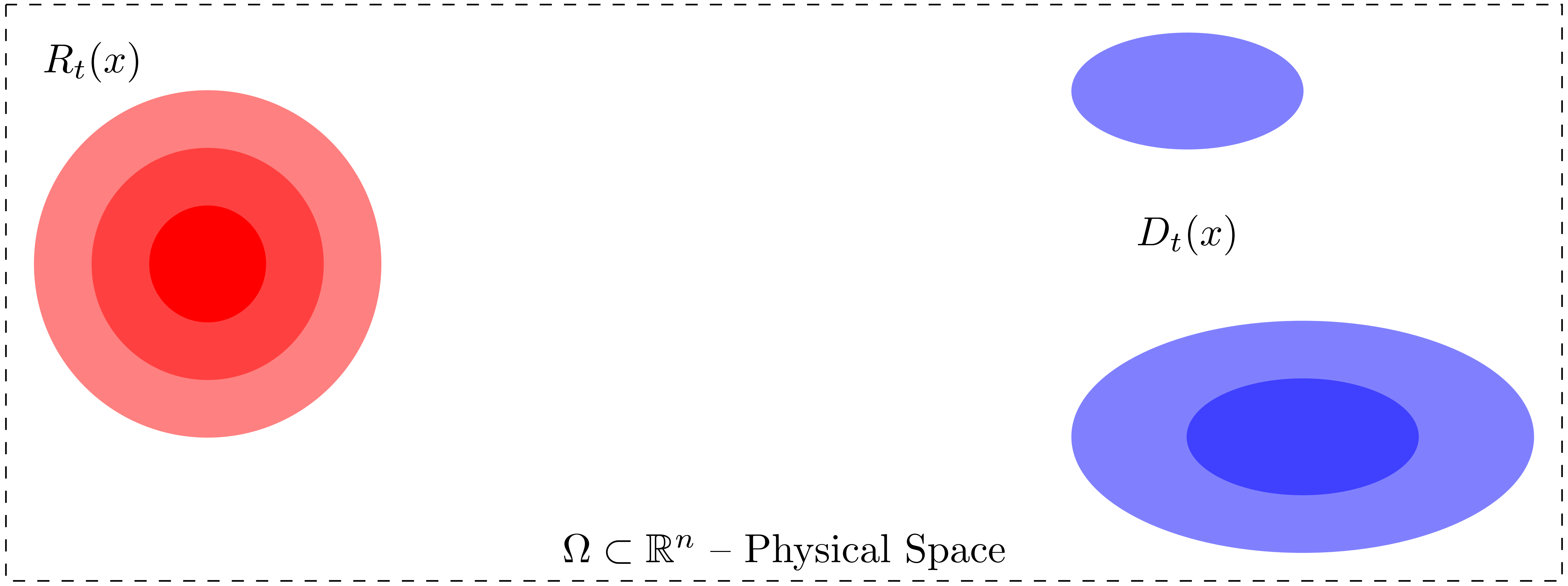
Approach

- This work: what sorts of motion patterns are optimal?
- Looking at motion planning and control for **tracking**
- Using **continuum models, optimal transport, optimal control**



Problem Formulation: Demand/Resource Distributions

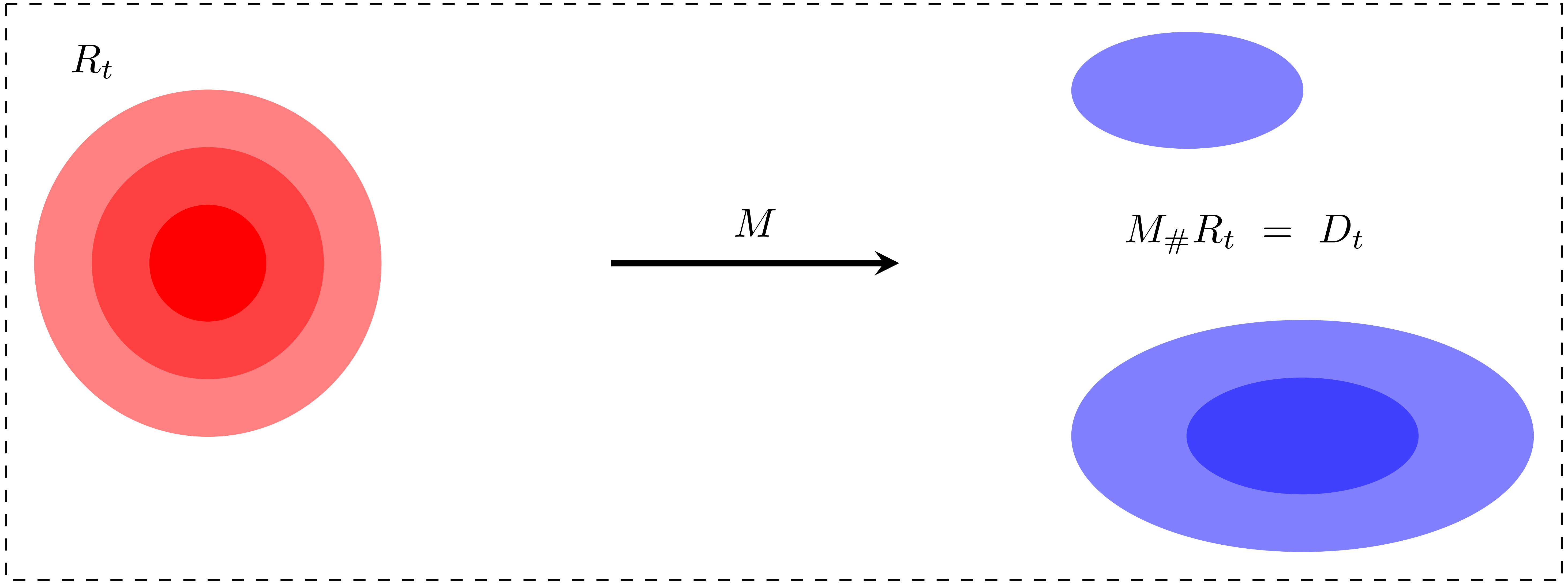
- Demand = known entity (requires services)
- Resource = controlled mobile agents (provides services)



Problem Formulation: Assignment

Monge Problem (Optimal Transport): $\inf \int_\Omega \|M(x) - x\|_2^2 \, R_t(x) \, dx$ s.t. $M_\# R_t = D_t$

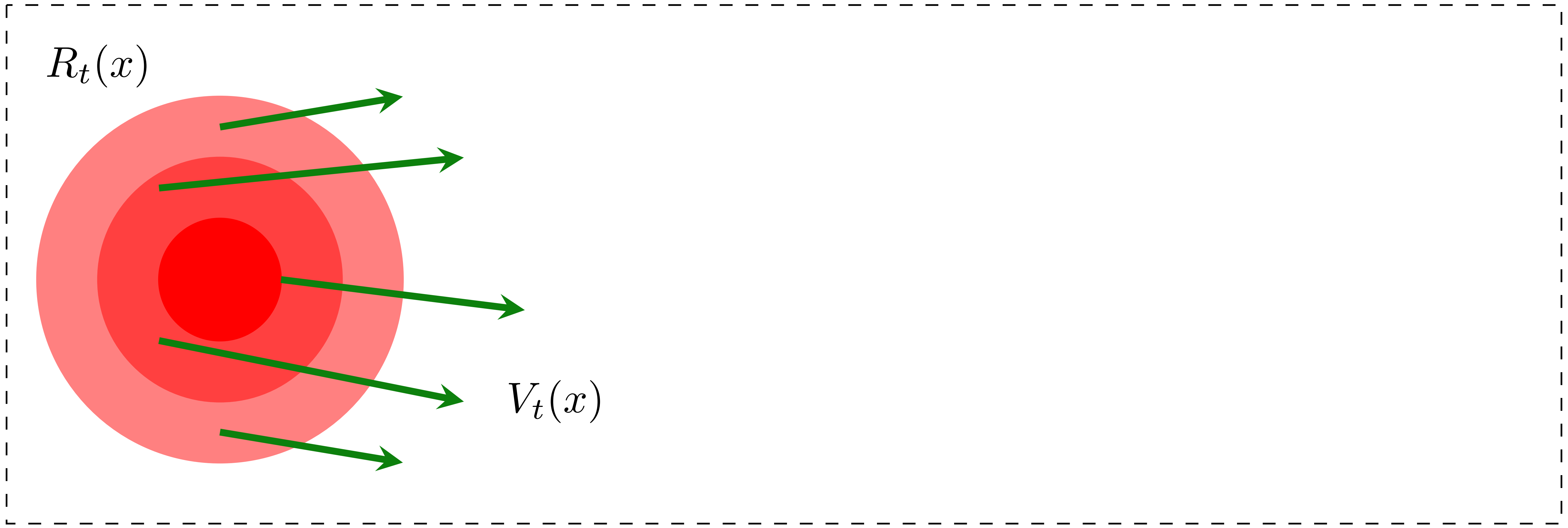
- # denotes **measure pushforward**
- Minimizer $\bar{M}_{R_t \to D_t}$ is **optimal assignment map**
- Minimum $W_2^2(R_t, D_t)$ is **Wasserstein distance**



Problem Formulation: Dynamic Model

- Tracking → want resources close to demand
- Control resource through **velocity field** V

Dynamics (Transport Equation): $\partial_t R_t(x) = -\nabla \cdot \big(R_t(x)\, V_t(x)\big)$



Motion Cost: $\|V_t\|^2_{L^2(R_t)} := \int_\Omega \|V_t(x)\|_2^2\, R_t(x)\, dx$

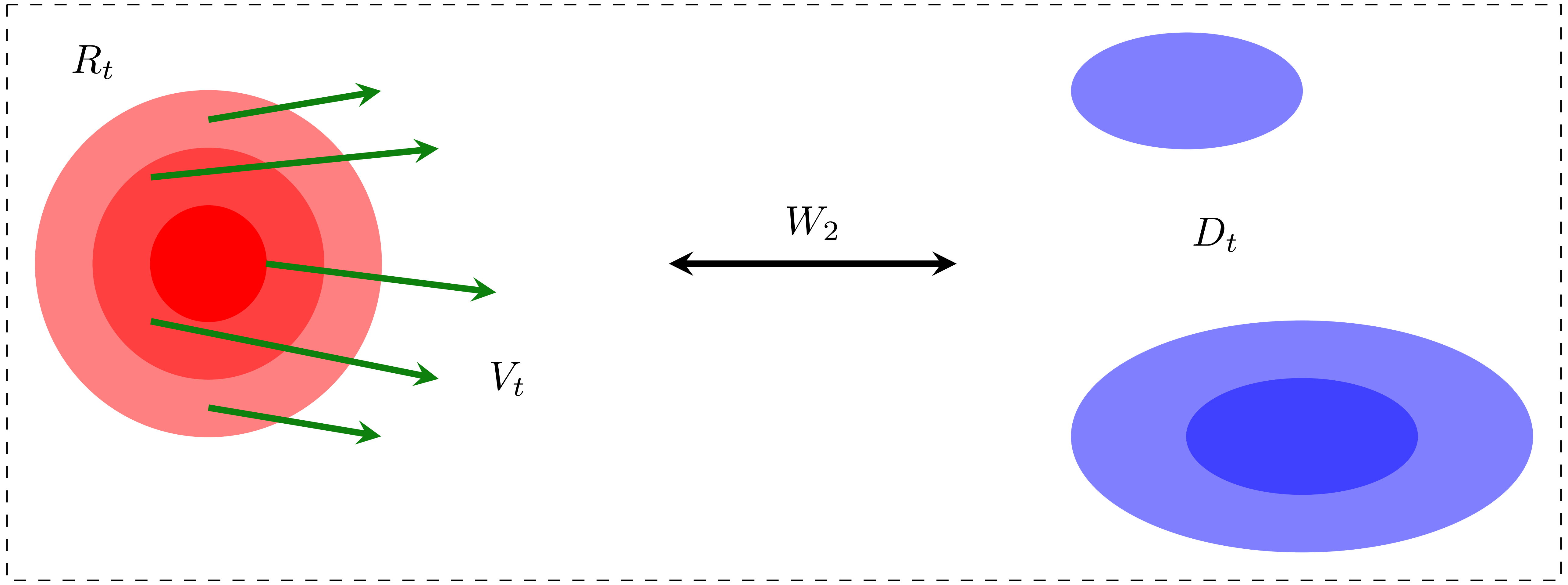
Formal Problem Statement

Given an initial resource distribution R_0 and demand trajectory D, solve

$$\inf_{R,V} \int_0^T \underbrace{W_2^2(R_t, D_t)}_{\text{Assignment Cost}} + \alpha \underbrace{\|V_t\|^2_{L^2(R_t)}}_{\text{Motion Cost}} \, dt \qquad \text{s.t.} \qquad \underbrace{\partial_t R_t = -\nabla \cdot (R_t V_t)}_{\text{Dynamic Constraint}}$$

- Intuitively, "R should track D efficiently"
- Trade-off parameter α controls relative importance of costs

Minimize cost over (R, V)



Structural Features of Solution

Necessary Conditions for Optimality:

$$\partial_t R_t = -\nabla \cdot (R_t \nabla \Lambda_t) \qquad R_0 = R_0$$

$$\partial_t \Lambda_t = -\frac{1}{2}\|\nabla \Lambda_t\|_2^2 + \frac{1}{2\alpha}\frac{\delta}{\delta R_t} W_2^2(R_t, D_t) \qquad \Lambda_T = 0$$

Structural Features of Solution

Necessary Conditions for Optimality:

Optimal velocity field is irrotational!

$$\partial_t R_t = -\nabla \cdot (R_t \boxed{\nabla \Lambda_t}) \qquad R_0 = R_0$$

$$\partial_t \Lambda_t = -\frac{1}{2}\|\nabla \Lambda_t\|_2^2 + \frac{1}{2\alpha}\frac{\delta}{\delta R_t} W_2^2(R_t, D_t) \qquad \Lambda_T = 0$$

Structural Features of Solution

Necessary Conditions for Optimality:

$$\begin{aligned} \partial_t R_t &= -\nabla \cdot (R_t \nabla \Lambda_t) \\ \partial_t \Lambda_t &= -\frac{1}{2}\|\nabla \Lambda_t\|_2^2 + \frac{1}{2\alpha}\frac{\delta}{\delta R_t} W_2^2(R_t, D_t) \end{aligned}$$

$$\begin{aligned} R_0 &= R_0 \\ \Lambda_T &= 0 \end{aligned}$$

Nonlinear two-point boundary value PDE

Structural Features of Solution

Necessary Conditions for Optimality:

$$\partial_t R_t = -\nabla \cdot (R_t \nabla \Lambda_t) \qquad R_0 = R_0$$

$$\partial_t \Lambda_t = -\frac{1}{2}\|\nabla \Lambda_t\|_2^2 + \boxed{\frac{1}{2\alpha}\frac{\delta}{\delta R_t} W_2^2(R_t, D_t)} \qquad \Lambda_T = 0$$

Demand trajectory enters through forcing term

Structural Features of Solution

Necessary Conditions for Optimality:

$$\partial_t R_t = -\nabla \cdot (R_t \nabla \Lambda_t) \qquad R_0 = R_0$$

$$\partial_t \Lambda_t = -\frac{1}{2}\|\nabla \Lambda_t\|_2^2 + \frac{1}{2\alpha}\frac{\delta}{\delta R_t} W_2^2(R_t, D_t) \qquad \Lambda_T = 0$$

- Optimal solutions are **noncausal:** need to know D ahead of time

Structural Features of Solution

Necessary Conditions for Optimality:

$$\partial_t R_t = -\nabla \cdot (R_t \nabla \Lambda_t) \qquad R_0 = R_0$$

$$\partial_t \Lambda_t = -\frac{1}{2}\|\nabla \Lambda_t\|_2^2 + \frac{1}{2\alpha}\boxed{\frac{\delta}{\delta R_t} W_2^2(R_t, D_t)} \qquad \Lambda_T = 0$$

Requires solving an optimization problem

- Optimal solutions are **noncausal:** need to know D ahead of time

Structural Features of Solution

Necessary Conditions for Optimality:

$$\begin{aligned} \partial_t R_t &= -\nabla \cdot (R_t \nabla \Lambda_t) & R_0 = R_0 \\ \partial_t \Lambda_t &= -\frac{1}{2}\|\nabla \Lambda_t\|_2^2 + \frac{1}{2\alpha}\frac{\delta}{\delta R_t} W_2^2(R_t, D_t) & \Lambda_T = 0 \end{aligned}$$

- Optimal solutions are **noncausal:** need to know D ahead of time
- Computational nightmare

Structural Features of Solution

Necessary Conditions for Optimality:

$$\partial_t R_t = -\nabla \cdot (R_t \nabla \Lambda_t) \qquad R_0 = R_0$$

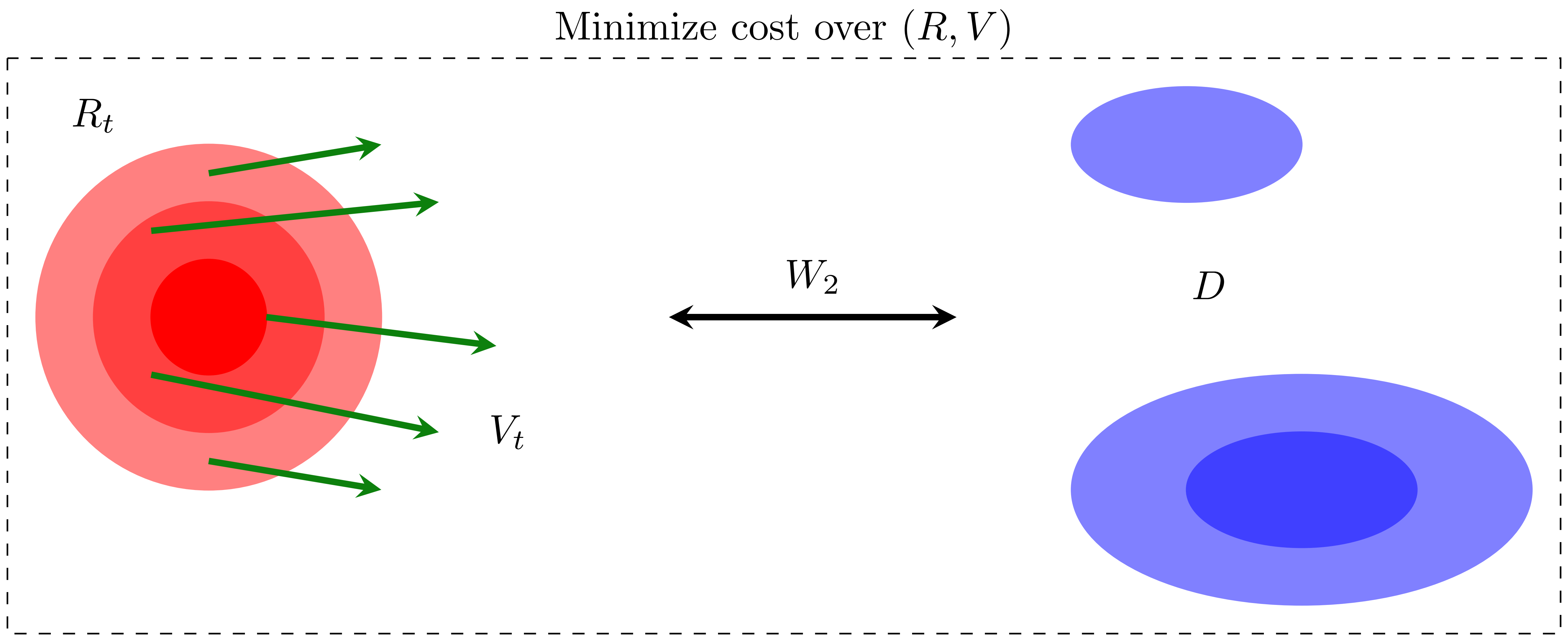
$$\partial_t \Lambda_t = -\frac{1}{2}\|\nabla \Lambda_t\|_2^2 + \frac{1}{2\alpha}\frac{\delta}{\delta R_t} W_2^2(R_t, D_t) \qquad \Lambda_T = 0$$

- Optimal solutions are **noncausal:** need to know D ahead of time
- Computational nightmare
- How to approach this?

Simple Case: Regulation Problem

- Consider same problem with D constant in time
- (In this case, know future trajectory of D!)

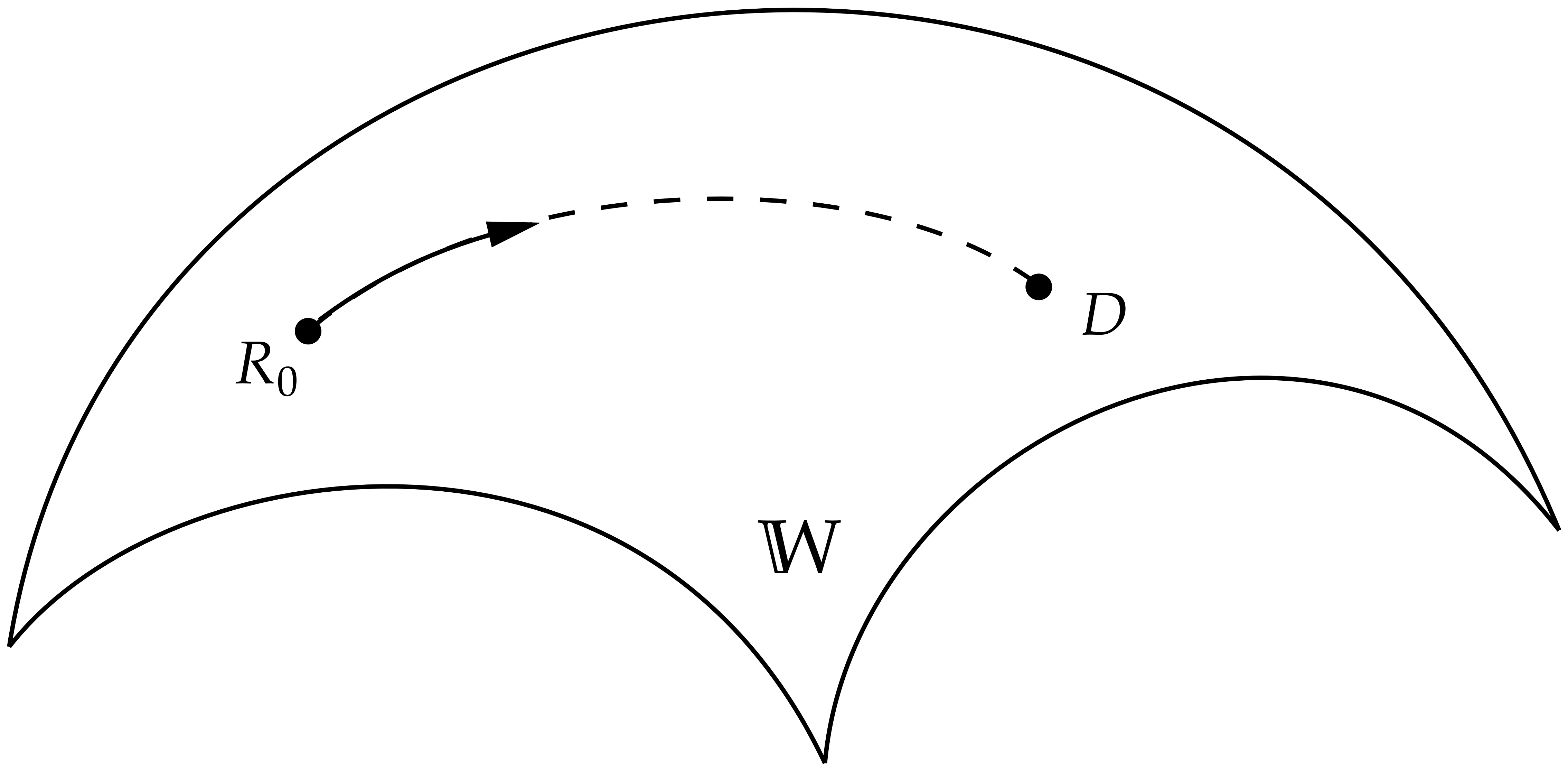
$$\inf_{R,V} \int_0^T W_2^2(R_t, D) \; + \; \alpha \, \|V_t\|^2_{L^2(R_t)} \, dt \qquad \text{s.t.} \qquad \partial_t R_t = -\nabla \cdot (R_t V_t)$$



Fact: Wasserstein Space = Riemannian Manifold

- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D:

$$R_t \;=\; \left[(1-\sigma(t))\; I \;+\; \sigma(t)\; \bar{M}_{R_0 \to D}\right]_{\#} R_0$$

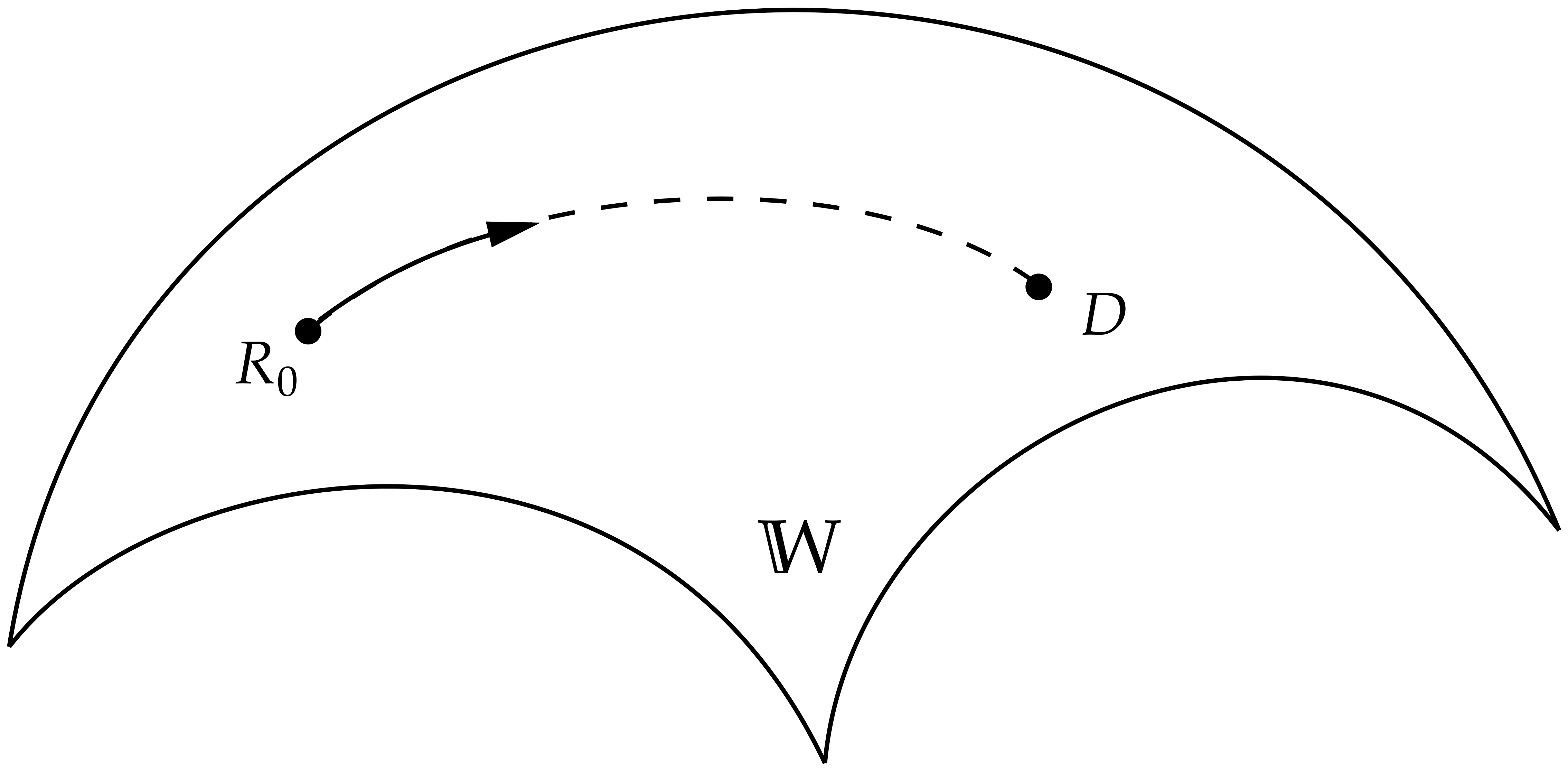


Fact: Wasserstein Space = Riemannian Manifold

- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D:

$$R_t = \Big[(1 - \sigma(t))\, I + \sigma(t)\, \bar{M}_{R_0 \to D} \Big]_{\#} R_0$$

Interpolation between identity and assignment map

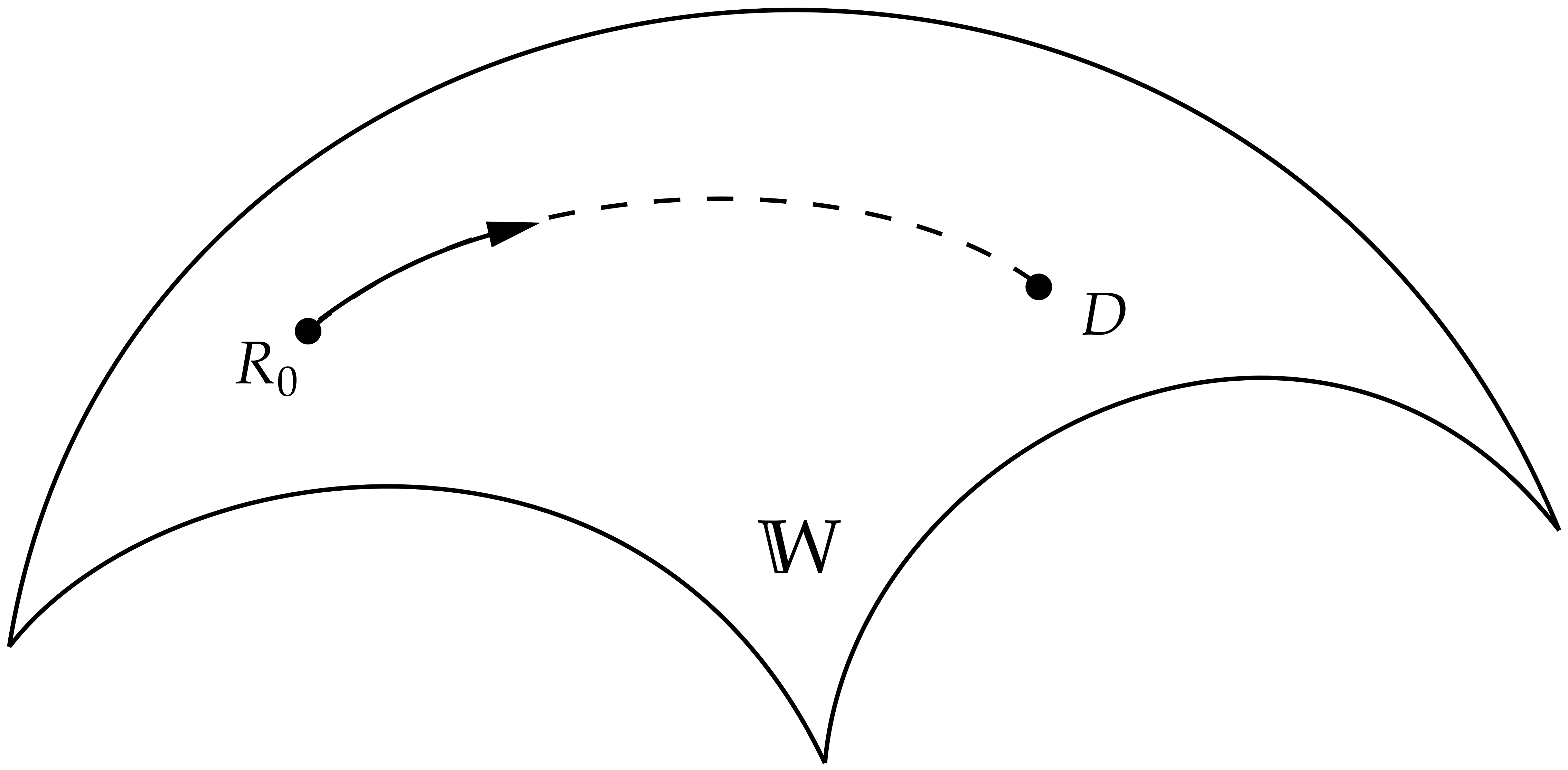


Fact: Wasserstein Space = Riemannian Manifold

- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D:

$$R_t \;=\; \left[(1-\sigma(t))\, I \;+\; \sigma(t)\, \bar{M}_{R_0\to D}\right]_{\#} R_0$$

Assignment map comes from OT

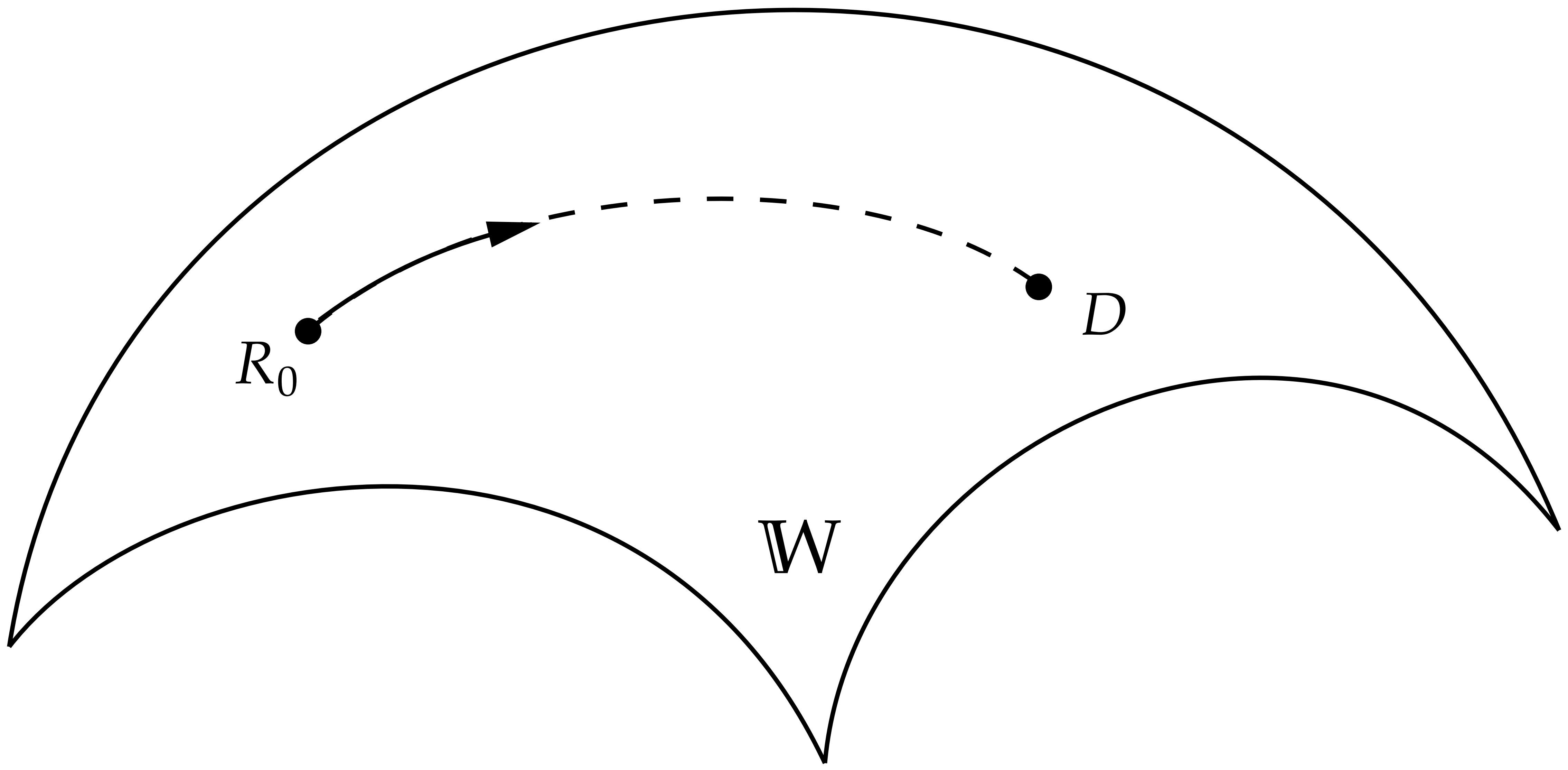


Fact: Wasserstein Space = Riemannian Manifold

- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D:

$$R_t = \left[(1-\sigma(t))\, I + \sigma(t)\, \bar{M}_{R_0 \to D} \right]_{\#} R_0$$

Time schedule σ comes from OC problem

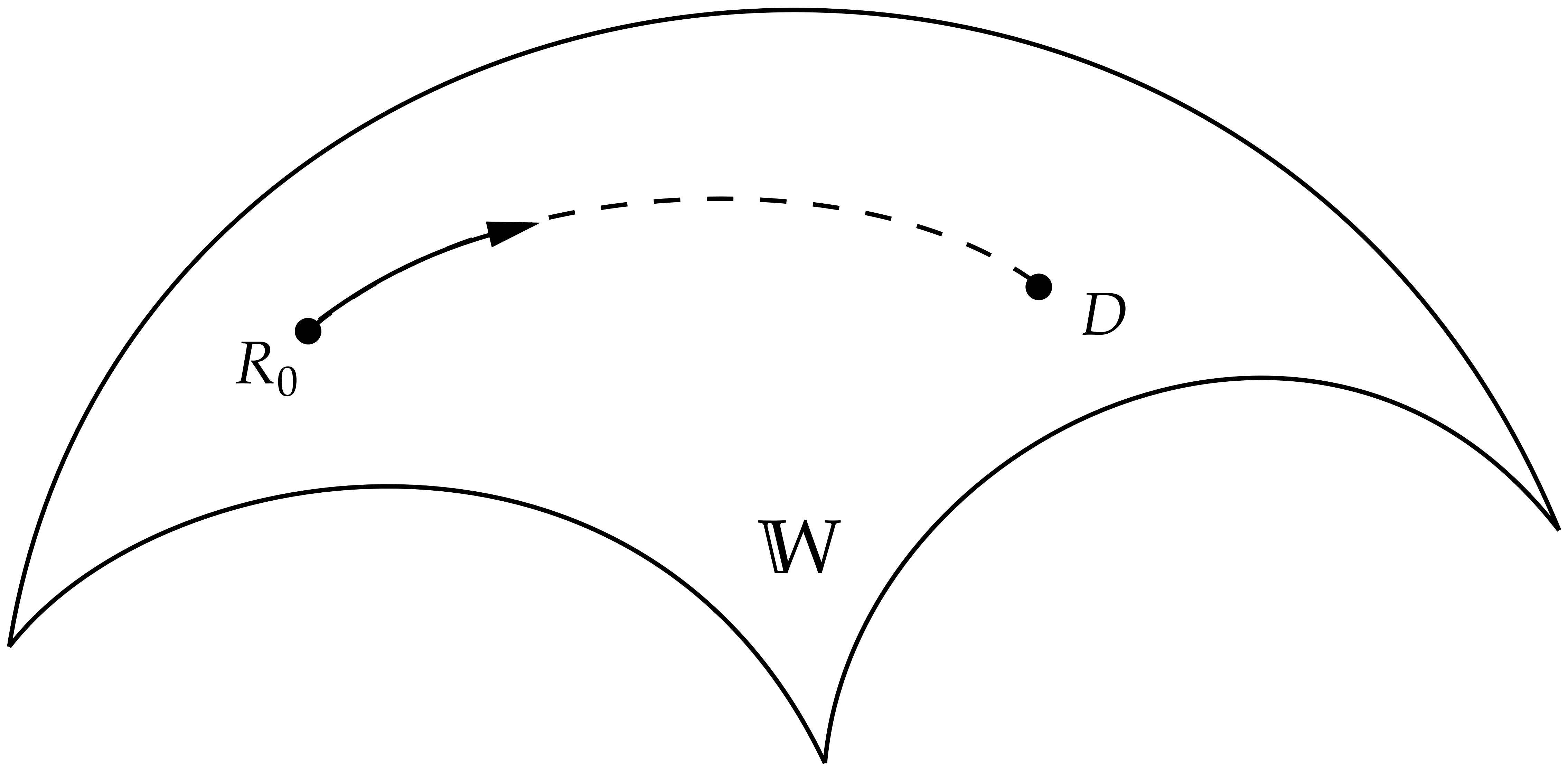


Fact: Wasserstein Space = Riemannian Manifold

- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D:

$$R_t \;=\; \Big[(1-\sigma(t))\;I \;+\; \sigma(t)\;\bar{M}_{R_0\to D}\Big]_{\#} R_0$$

- Can show assignments conserved → particle trajectories independent!



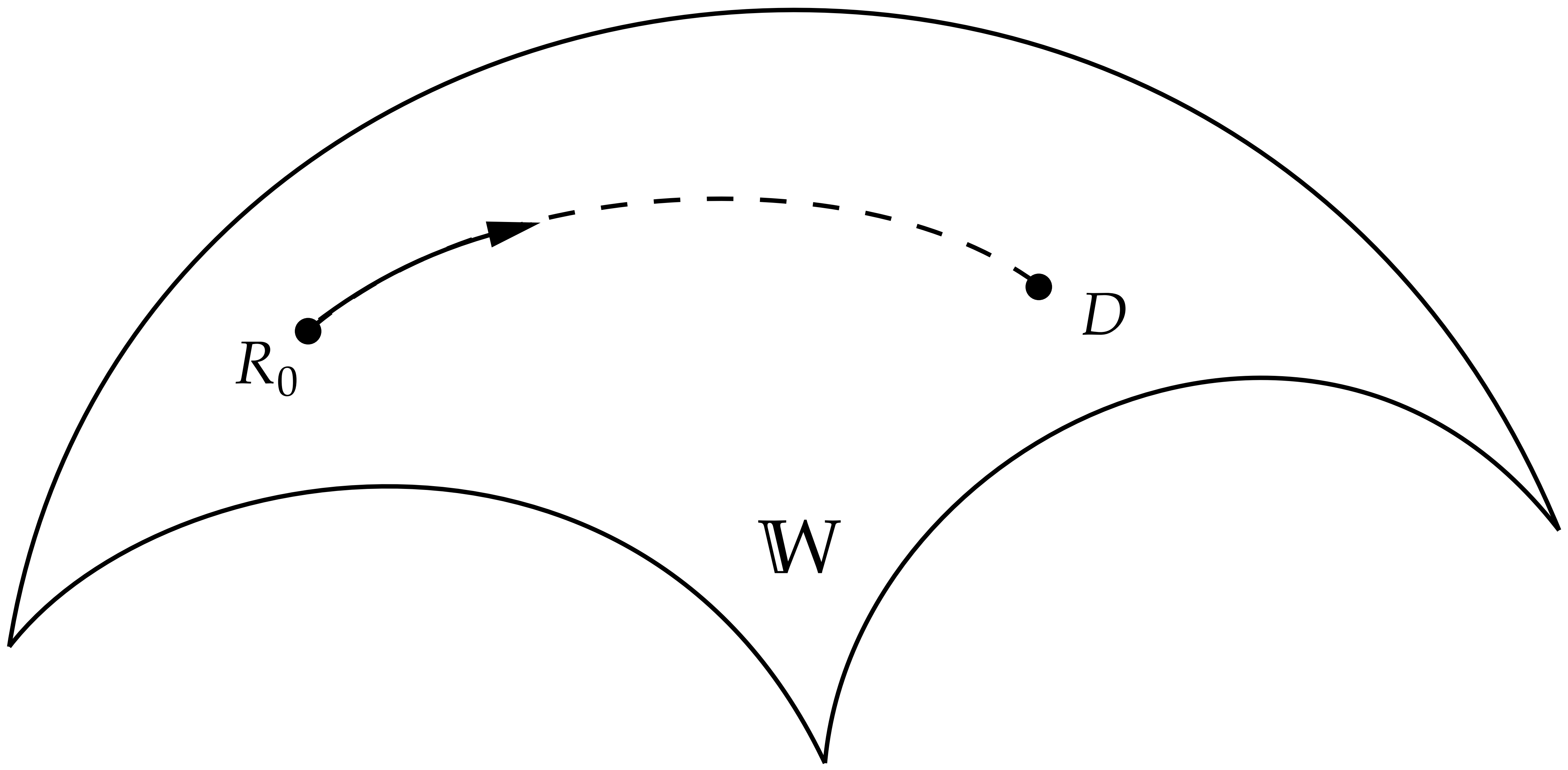
Fact: Wasserstein Space = Riemannian Manifold

- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D:

$$R_t \;=\; \left[(1-\sigma(t))\, I \;+\; \sigma(t)\, \bar{M}_{R_0 \to D}\right]_{\#} R_0$$

- Can show assignments conserved → particle trajectories independent!
- Gives form of optimal controller:

$$V_t \;=\; \dot{\sigma}(t)\left(\bar{M}_{R_t \to D} - I\right)$$



Fact: Wasserstein Space = Riemannian Manifold

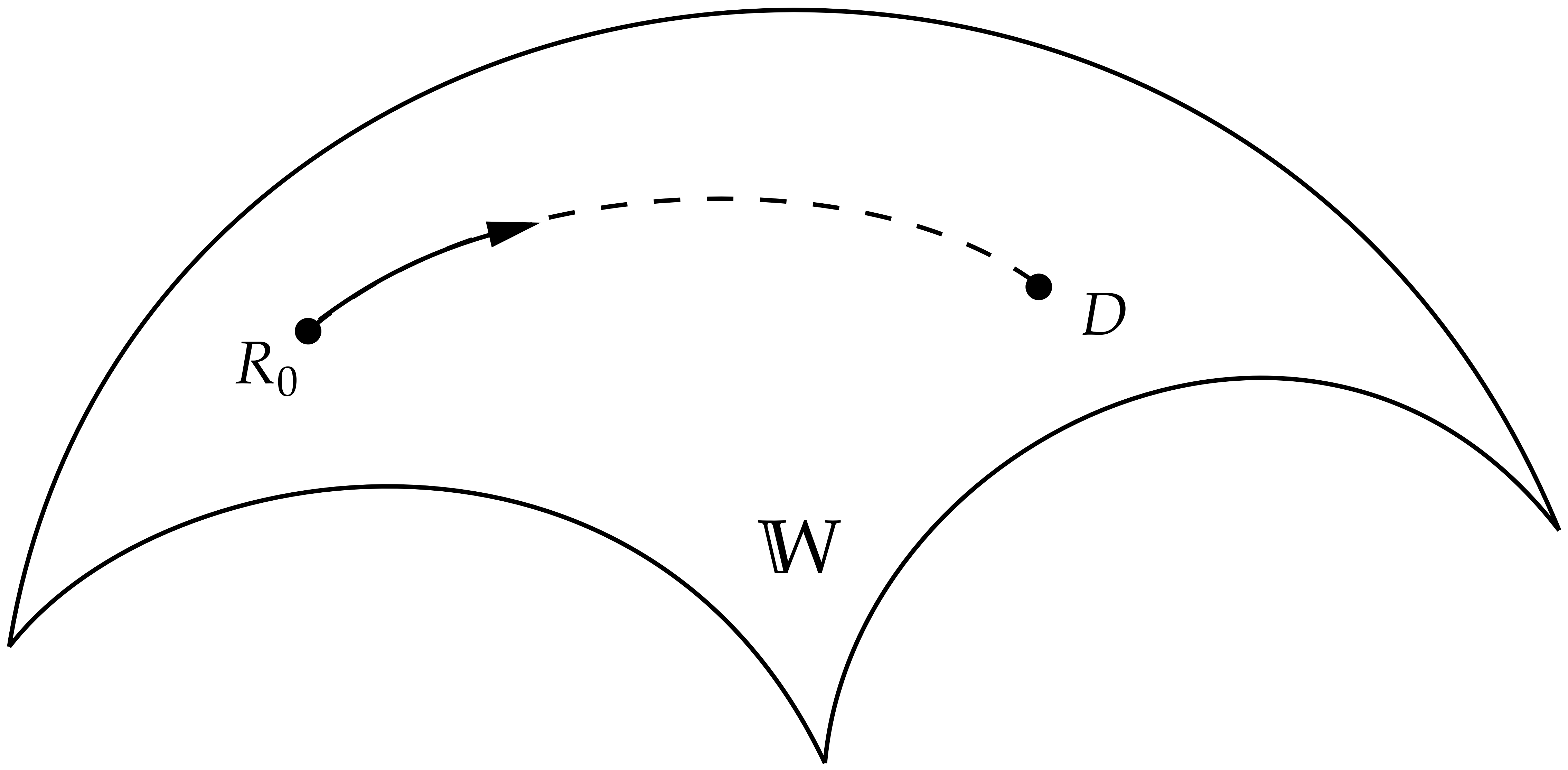
- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D:

$$R_t \;=\; \left[(1-\sigma(t))\; I \;+\; \sigma(t)\; \bar{M}_{R_0 \to D}\right]_{\#} R_0$$

- Can show assignments conserved → particle trajectories independent!
- Gives form of optimal controller:

$$V_t \;=\; \dot{\sigma}(t)\left(\bar{M}_{R_t \to D} - I\right)$$

Error vector



Fact: Wasserstein Space = Riemannian Manifold

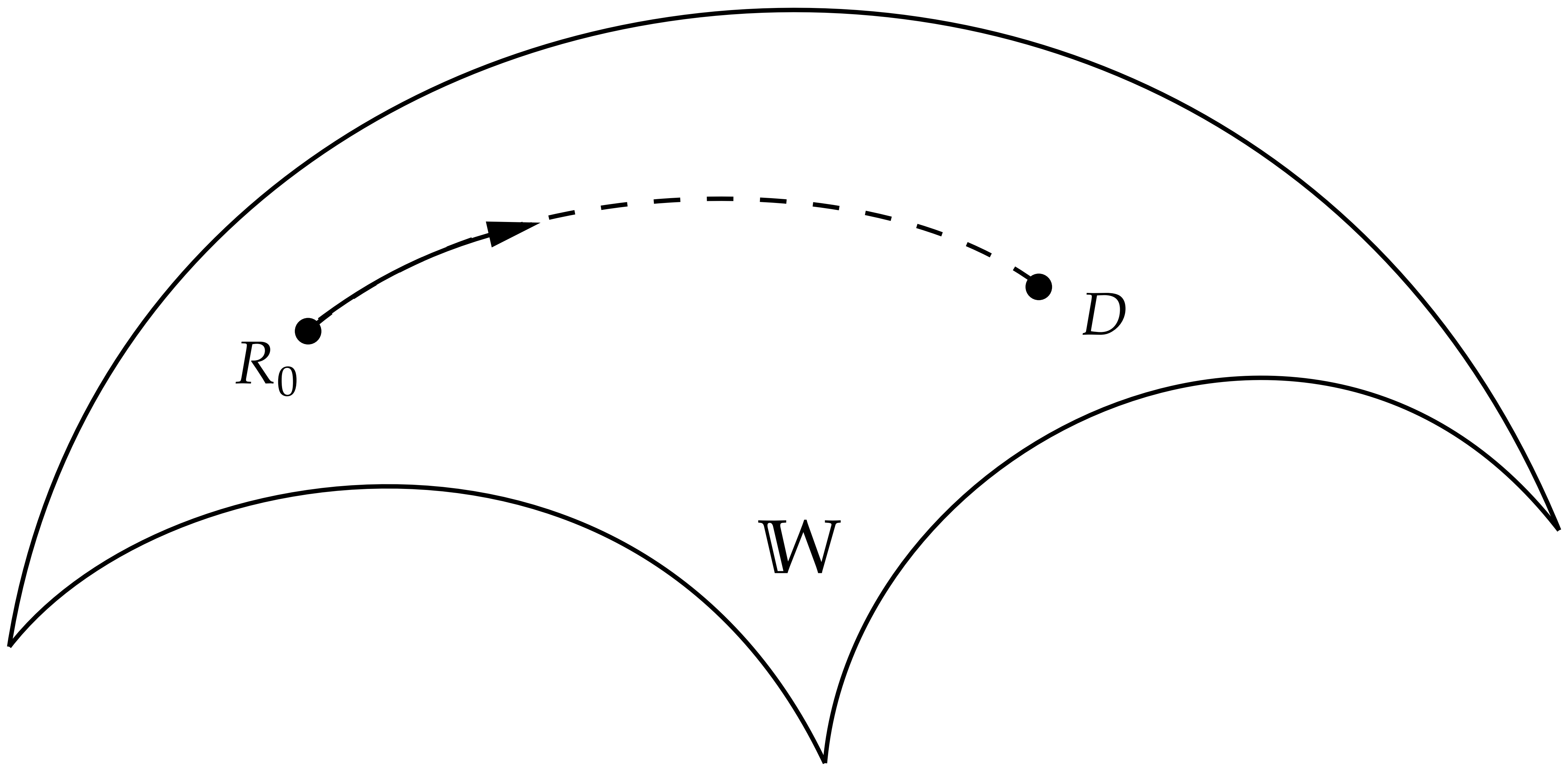
- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D:

$$R_t = \left[(1-\sigma(t))\, I + \sigma(t)\, \bar{M}_{R_0 \to D}\right]_{\#} R_0$$

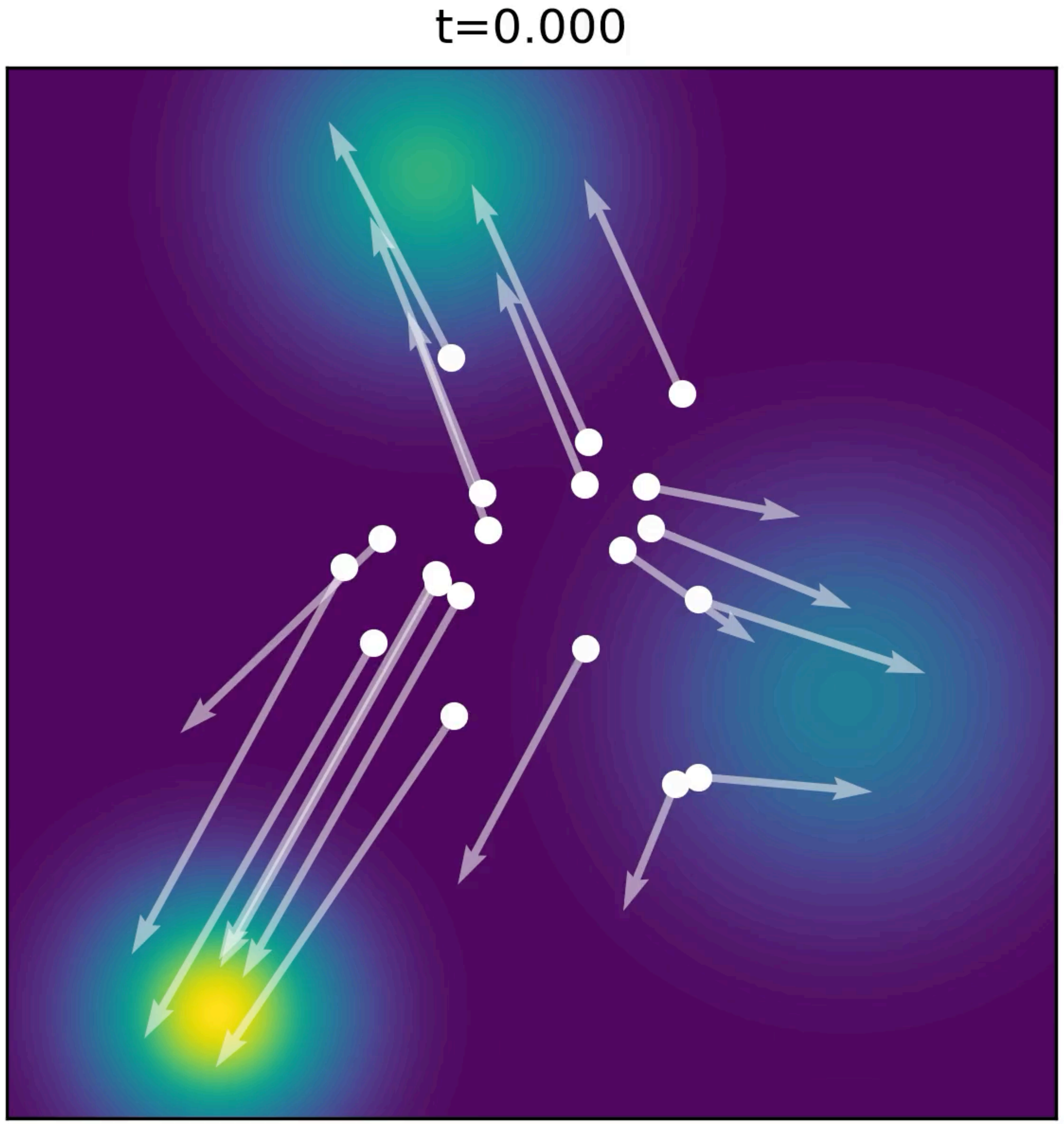
- Can show assignments conserved → particle trajectories independent!
- Gives form of optimal controller:

$$V_t = \boxed{\dot{\sigma}(t)}\left(\bar{M}_{R_t \to D} - I\right)$$

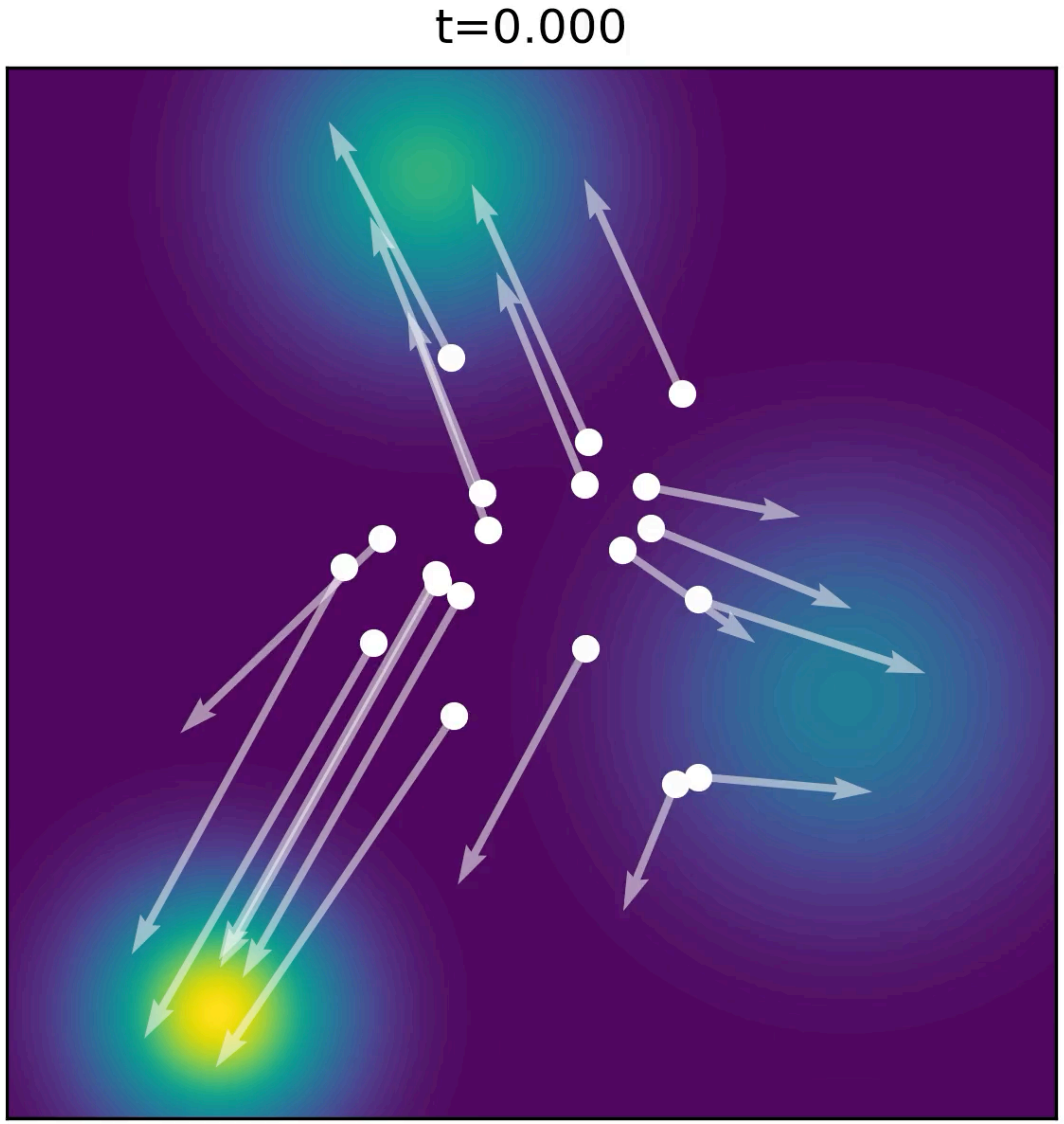
Rate of traversal



Regulation Problem: Simulation



Regulation Problem: Simulation



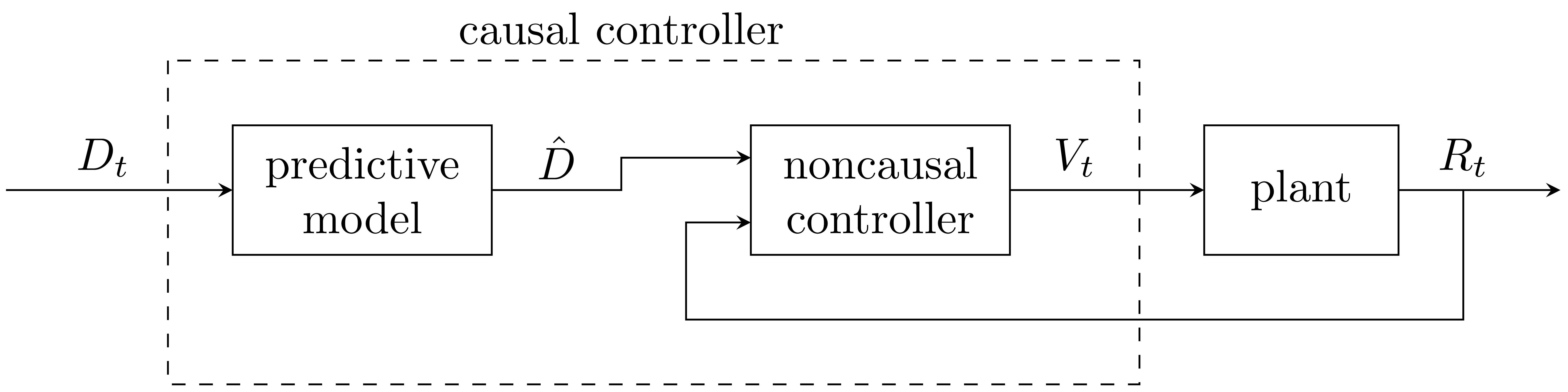
Back to Tracking Problem

Back to Tracking Problem

- **Problem #1:** Need to address noncausality

Back to Tracking Problem

- **Problem #1:** Need to address noncausality
- **Idea #1:** use model to predict demand trajectory, use predicted trajectory in necessary conditions in receding horizon scheme



Back to Tracking Problem

- **Problem #1:** Need to address noncausality
- **Idea #1:** use model to predict demand trajectory, use predicted trajectory in necessary conditions in receding horizon scheme
- **Problem #2:** Don't have model for demand

Back to Tracking Problem

- **Problem #1:** Need to address noncausality
- **Idea #1:** use model to predict demand trajectory, use predicted trajectory in necessary conditions in receding horizon scheme
- **Problem #2:** Don't have model for demand
- **Idea #2:** Use static predictive model:

$$\boxed{\partial_t \hat{D}_t = 0}$$

Back to Tracking Problem

- **Problem #1:** Need to address noncausality
- **Idea #1:** use model to predict demand trajectory, use predicted trajectory in necessary conditions in receding horizon scheme
- **Problem #2:** Don't have model for demand
- **Idea #2:** Use static predictive model:

$$\boxed{\partial_t \hat{D}_t = 0}$$

- (Also solves **Problem #3:** computational cost)

MPC Algorithm for Time-Varying Demands

- Predict that future demand = current demand
- Compute optimal control, apply for Δ_t seconds
- Update demand and repeat

MPC Algorithm for Time-Varying Demands

- Predict that future demand = current demand
- Compute optimal control, apply for Δ_t seconds
- Update demand and repeat

- Resulting controller has form:

$$V_t \;\approx\; \dot{\sigma}(t)\left(\bar{M}_{R_t \to D_t} - I\right)$$

MPC Algorithm for Time-Varying Demands

- Predict that future demand = current demand
- Compute optimal control, apply for Δ_t seconds
- Update demand and repeat

- Resulting controller has form:

$$V_t \;\approx\; \dot{\sigma}(t)\left(\bar{M}_{R_t \to D_t} - I\right)$$

- Heuristic, not optimal in general

MPC Algorithm for Time-Varying Demands

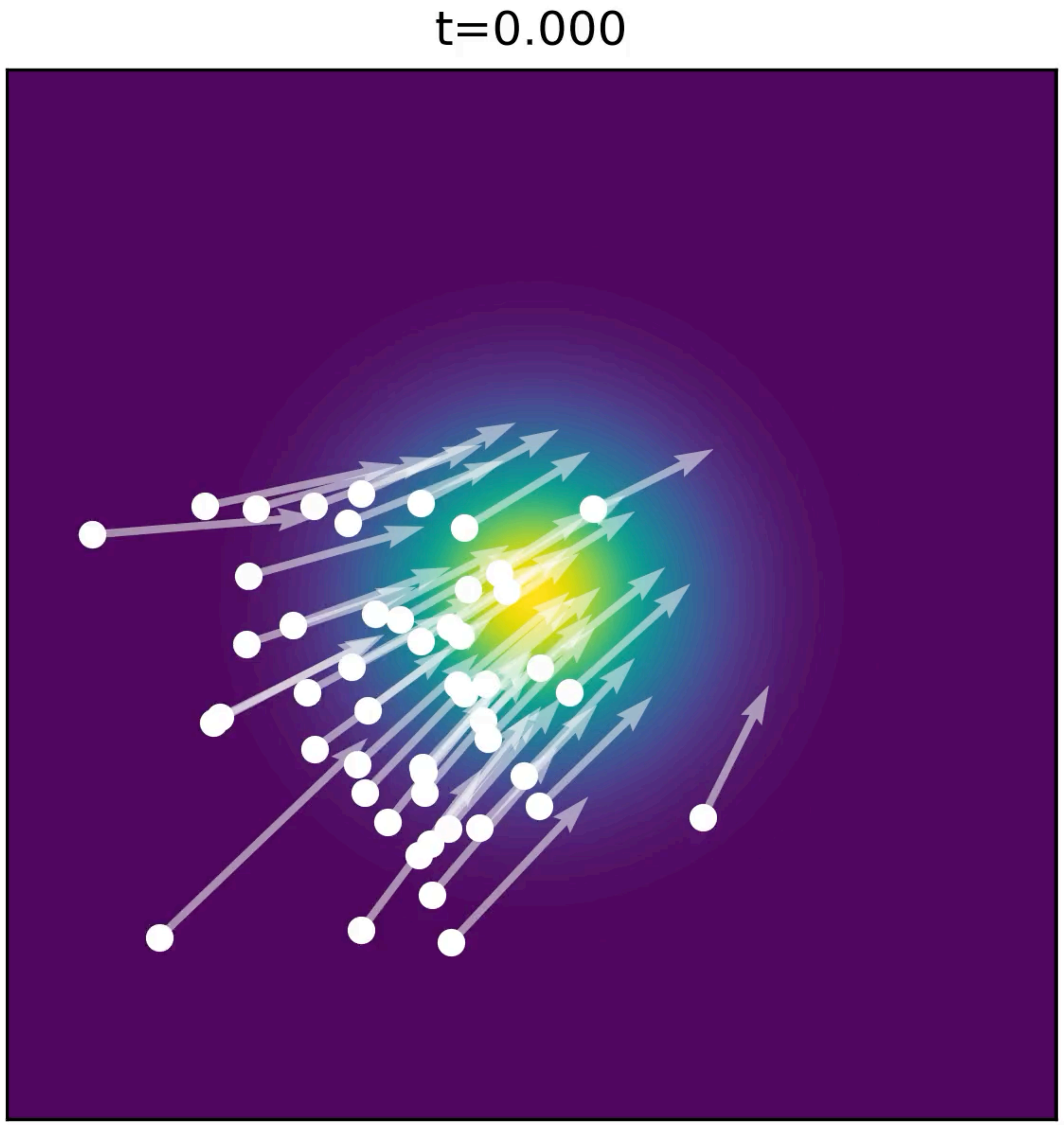
- Predict that future demand = current demand
- Compute optimal control, apply for Δ_t seconds
- Update demand and repeat

- Resulting controller has form:

$$V_t \approx \dot{\sigma}(t)\left(\bar{M}_{R_t \to D_t} - I\right)$$

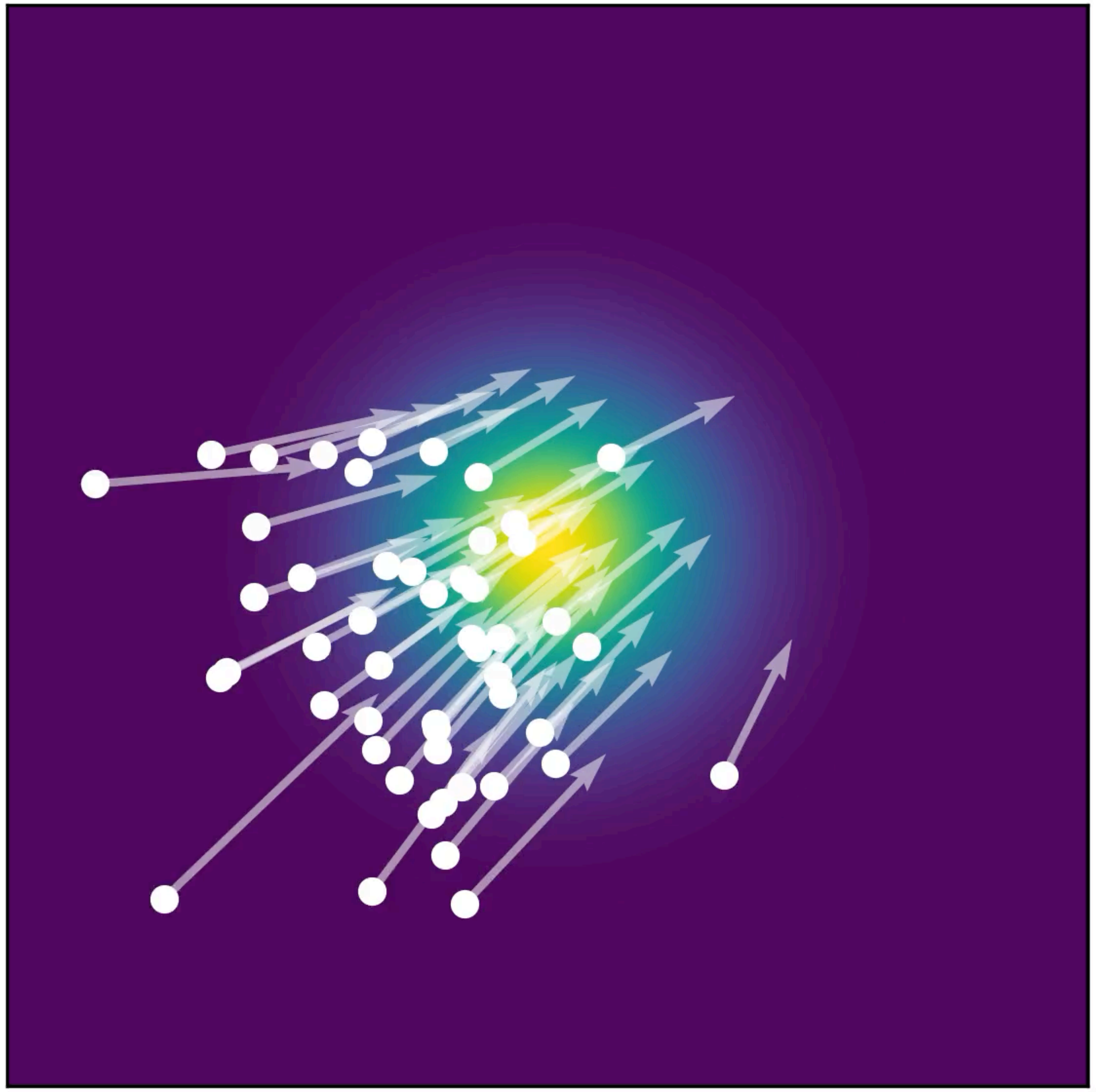
- Heuristic, not optimal in general
- But, can be applied in real-time

MPC Algorithm: Simulations



MPC Algorithm: Simulations

t=0.000



Conclusion

Takeaways:

- Simplified models can provide insight and design heuristics
- Leveraging geometric structure can be powerful

Future Work:

- Solving necessary conditions
- More sophisticated demand models
- Investigating resulting controllers

Thanks to My Collaborators

Bassam Bamieh

Jared Jonas

Thanks for watching!

Questions?