

Causal Tracking of Distributions in Wasserstein Space: A Model Predictive Control Scheme

63rd IEEE Conference on Decision and Control
December 19, 2024

Max Emerick, Jared Jonas, Bassam Bamieh

Many Applications for Autonomous Swarms



Emergency Response



Logistics



Entertainment



Transportation



Defense



Data Collection

Motivation

The Problem in Focus:

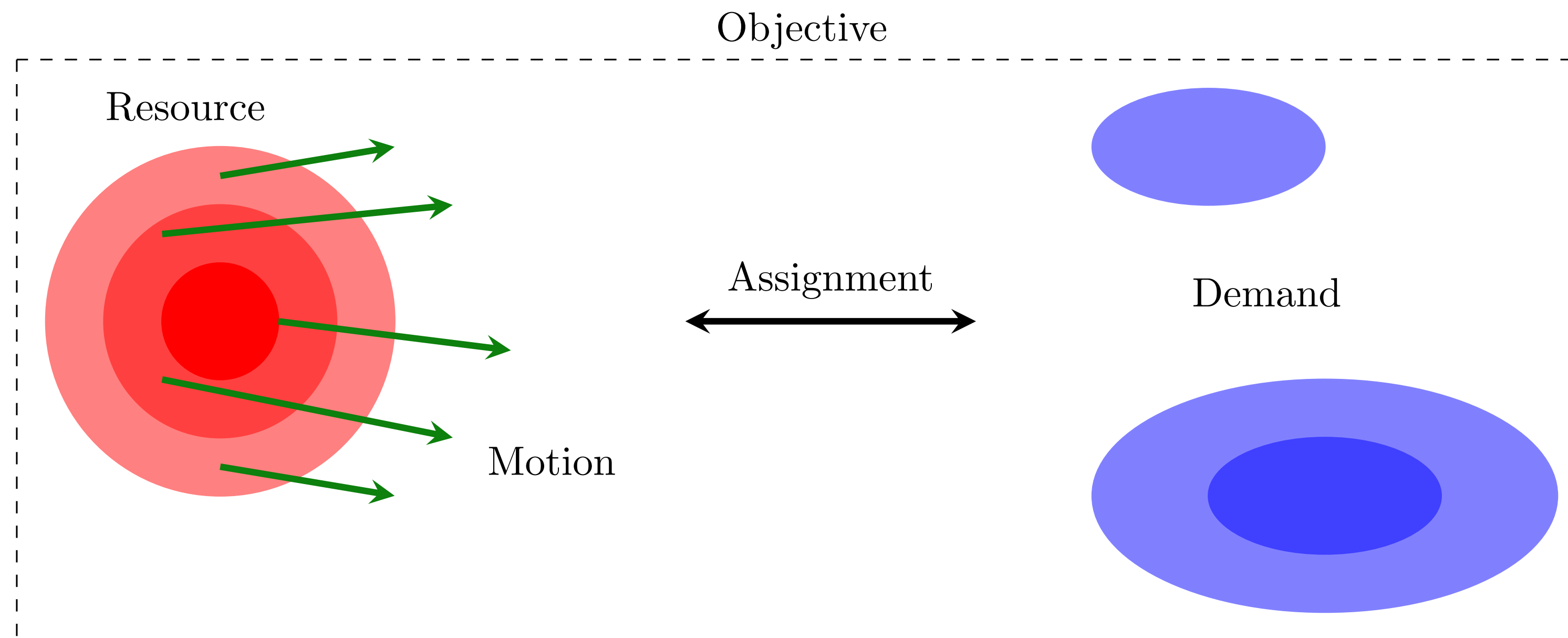
- Large swarms robust/efficient, but hard to model/control
- Want to develop theoretical foundations for design heuristics

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?

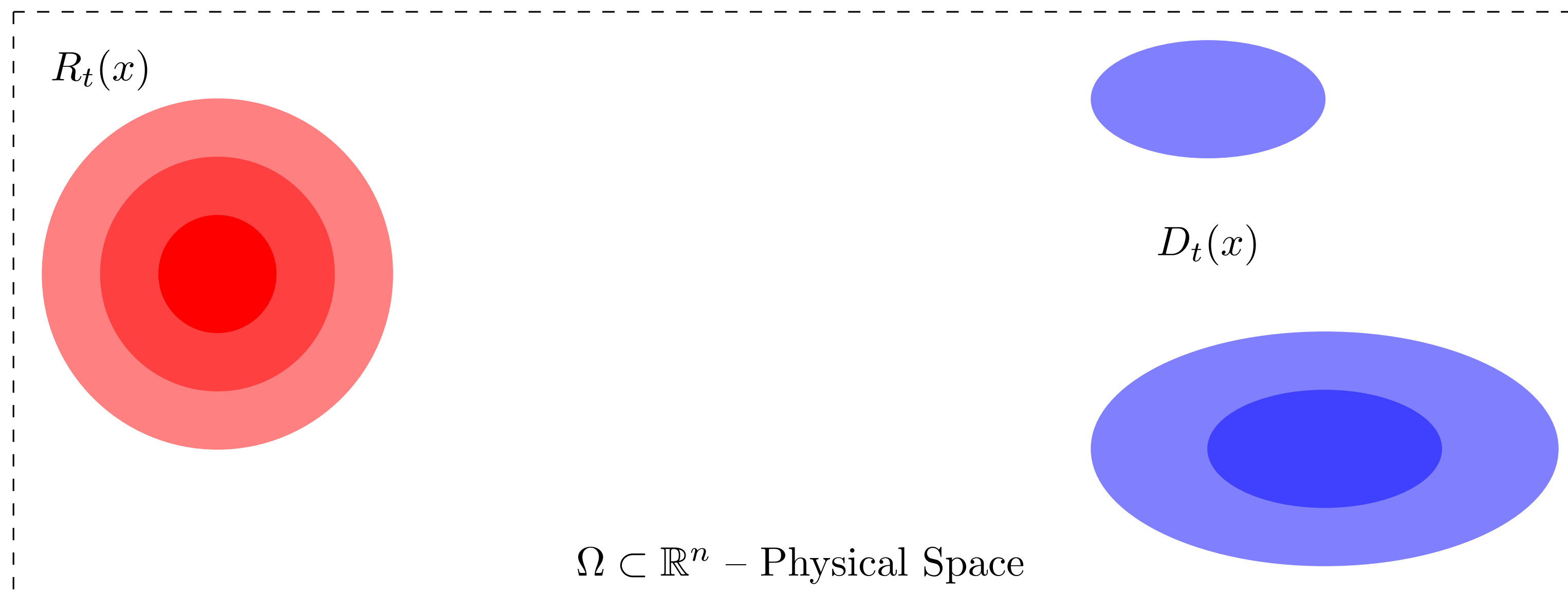
Approach

- 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

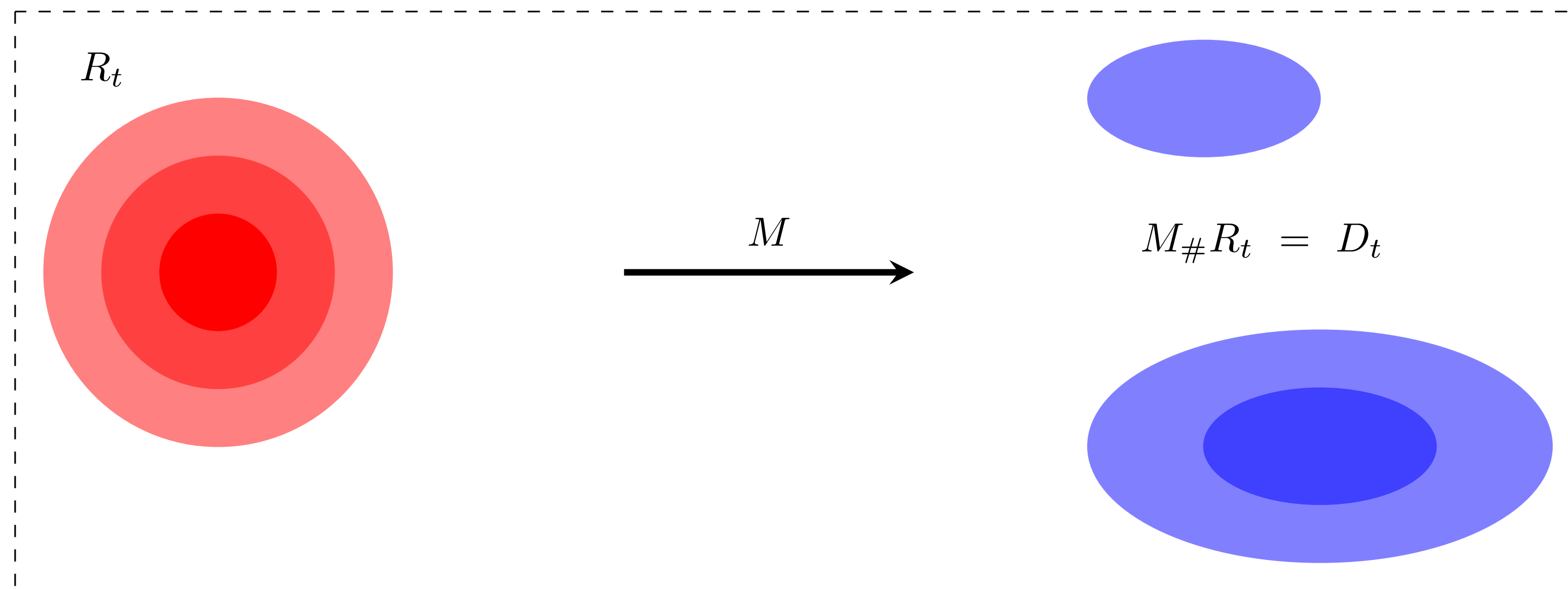
- **Demand** = known entity (requires services)
- **Resource** = controlled mobile agents (provides services)



Problem Formulation: Assignment

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

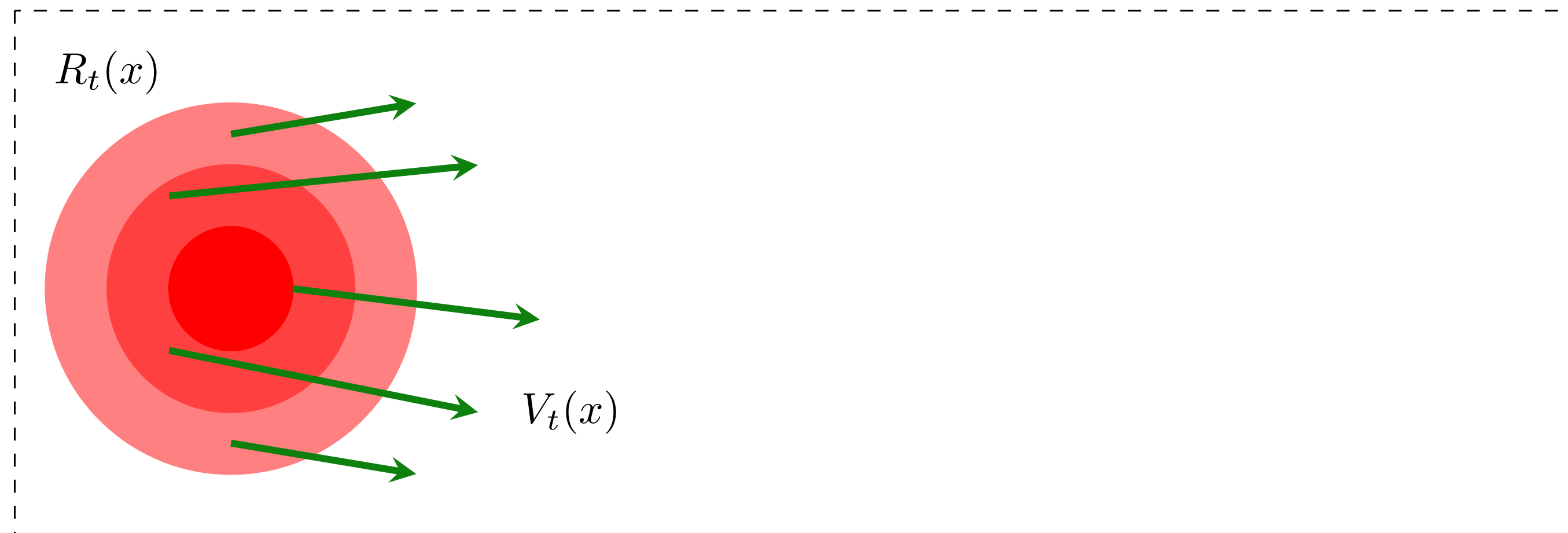
- # denotes **measure pushforward**
- Minimizer $\bar{M}_{R_t \rightarrow D_t}$ is **optimal assignment map**
- Minimum $W_2^2(R_t, D_t)$ is **Wasserstein distance**



Problem Formulation: Dynamic Model

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

Dynamics (Continuity Equation):
$$\partial_t R_t(x) = -\nabla \cdot (R_t(x) V_t(x))$$



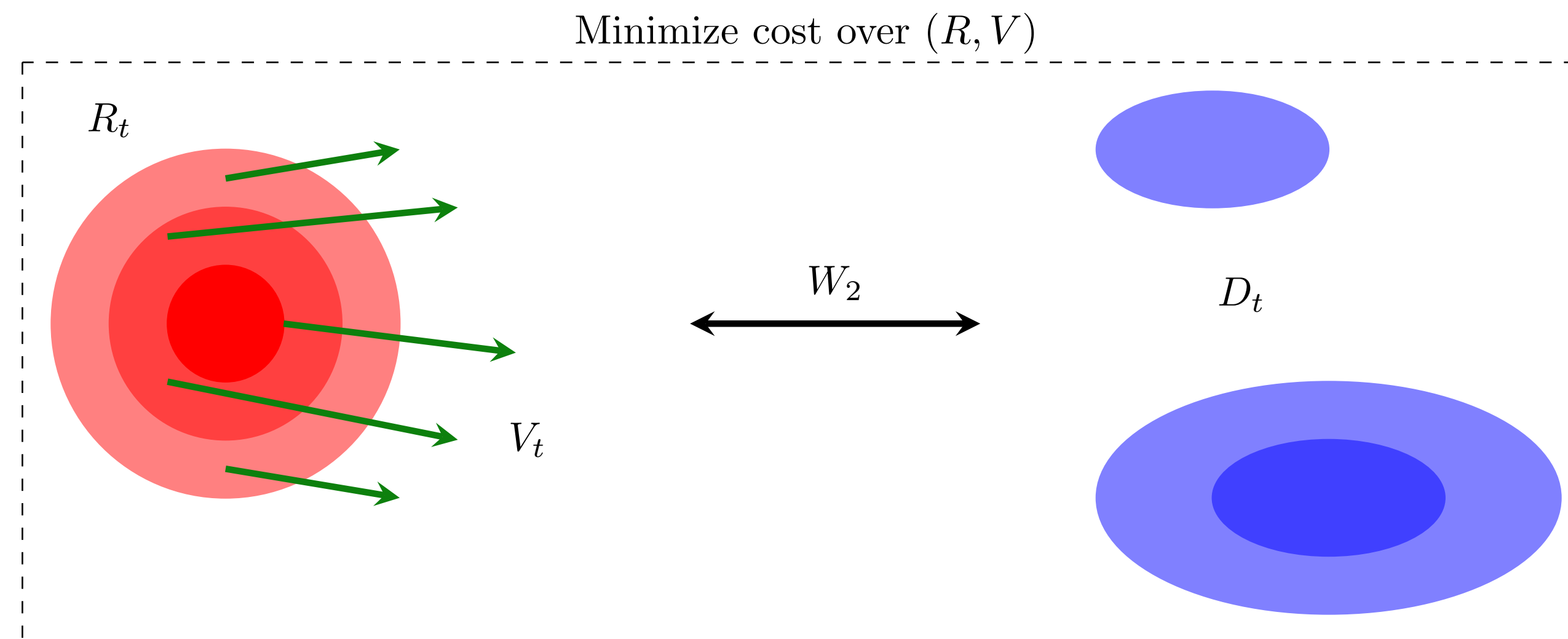
Motion Cost:
$$\|V_t\|_{L^2(R_t)}^2 := \int_{\Omega} \|V_t(x)\|_2^2 R_t(x) dx$$

Formal Problem Statement

Given an initial resource distribution R_0 and demand trajectory D over $[0, T]$, solve

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

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



Structural Features of Solution

Necessary Conditions for Optimality:

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

Structural Features of Solution

Necessary Conditions for Optimality:

Optimal velocity field is irrotational!

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

Structural Features of Solution

Necessary Conditions for Optimality:

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

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

$$R_0 = R_0$$

$$\Lambda_T = 0$$

Nonlinear two-point
boundary value PDE

Structural Features of Solution

Necessary Conditions for Optimality:

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

Demand trajectory enters through forcing term

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

Structural Features of Solution

Necessary Conditions for Optimality:

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

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:

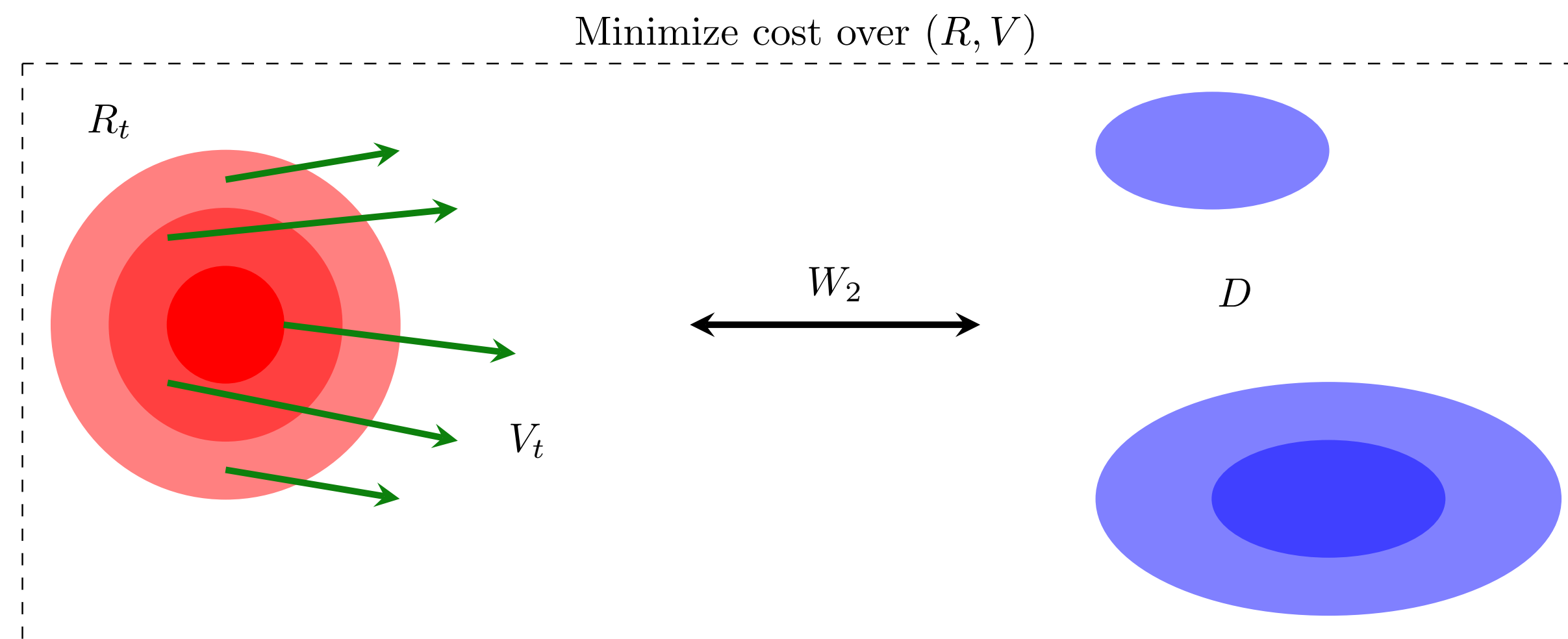
$$\begin{aligned}\partial_t R_t &= -\nabla \cdot (R_t \nabla \Lambda_t) & R_0 &= R_0 \\ \partial_t \Lambda_t &= -\frac{1}{2} \|\nabla \Lambda_t\|_2^2 + \frac{1}{2\alpha} \frac{\delta}{\delta R_t} W_2^2(R_t, D_t) & \Lambda_T &= 0\end{aligned}$$

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

Simple Case: Regulation Problem

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

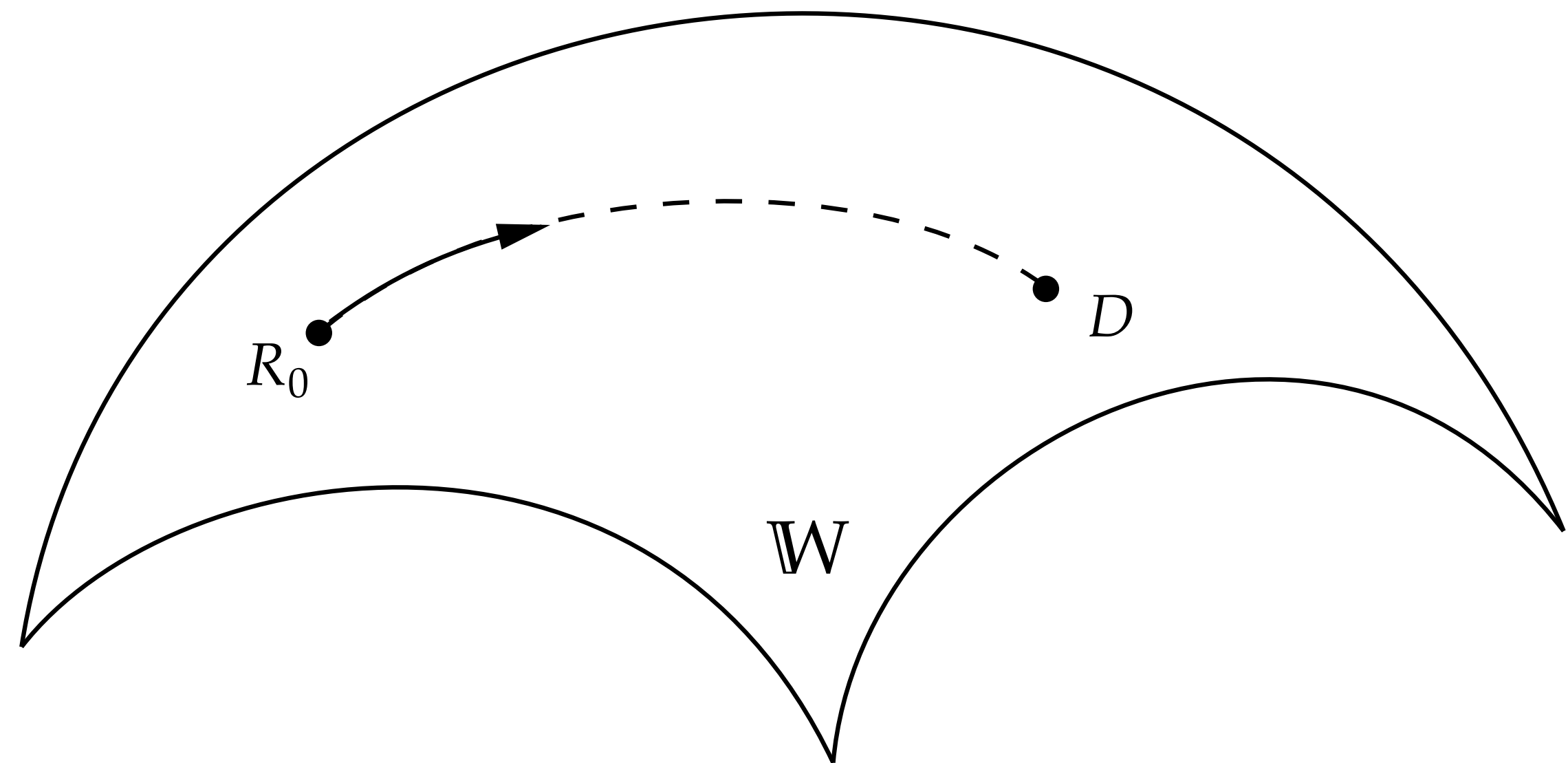
$$\inf_{R,V} \int_0^T W_2^2(R_t, D) + \alpha \|V_t\|_{L^2(R_t)}^2 dt \quad \text{s.t.} \quad \partial_t R_t = -\nabla \cdot (R_t V_t)$$



Fact: Wasserstein Space = Riemannian Manifold

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

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

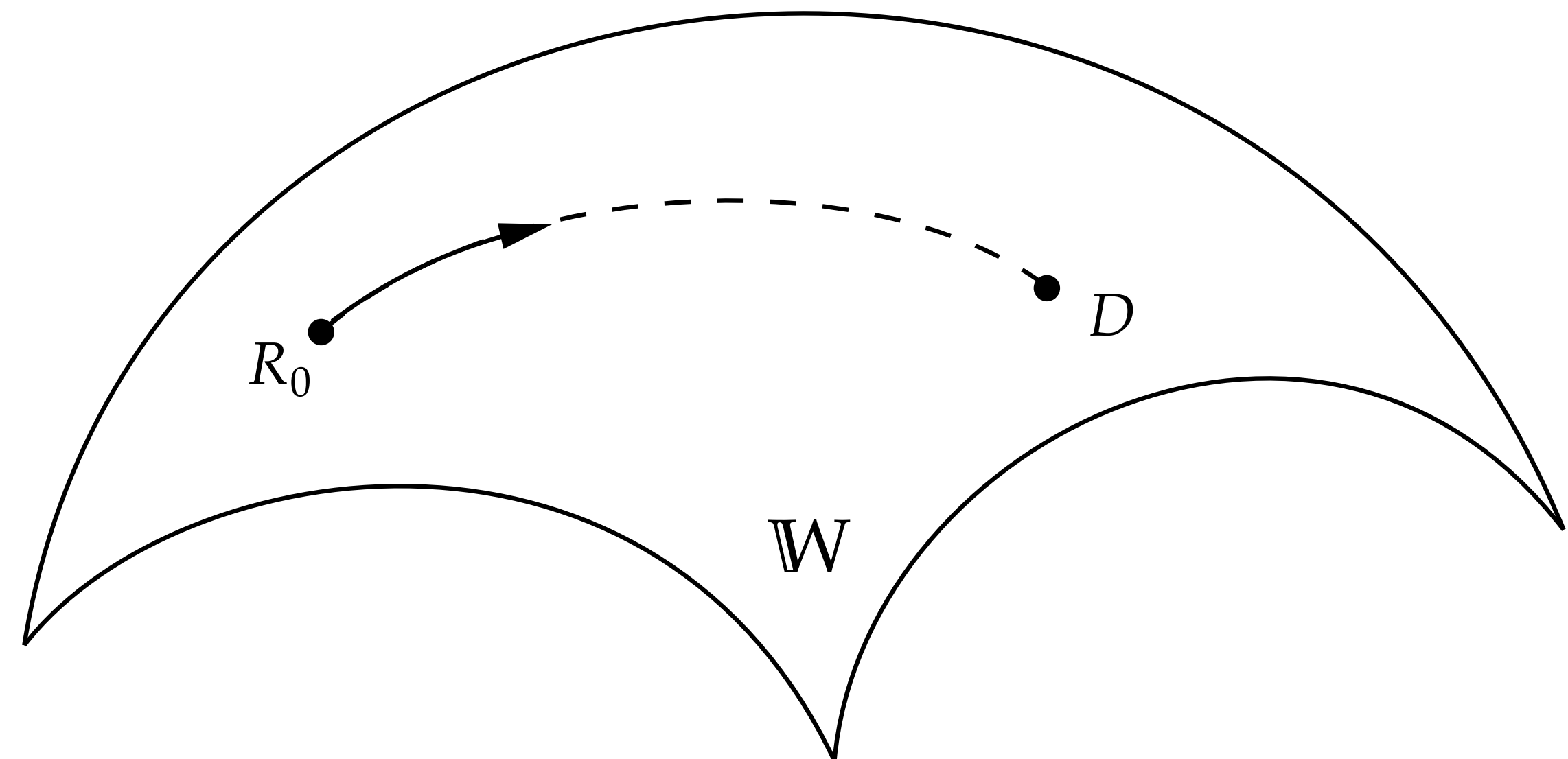


Fact: Wasserstein Space = Riemannian Manifold

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

$$R_t = \boxed{(1 - \sigma(t)) I + \sigma(t) \bar{M}_{R_0 \rightarrow D}}_{\#} R_0$$

Interpolation between identity and assignment map

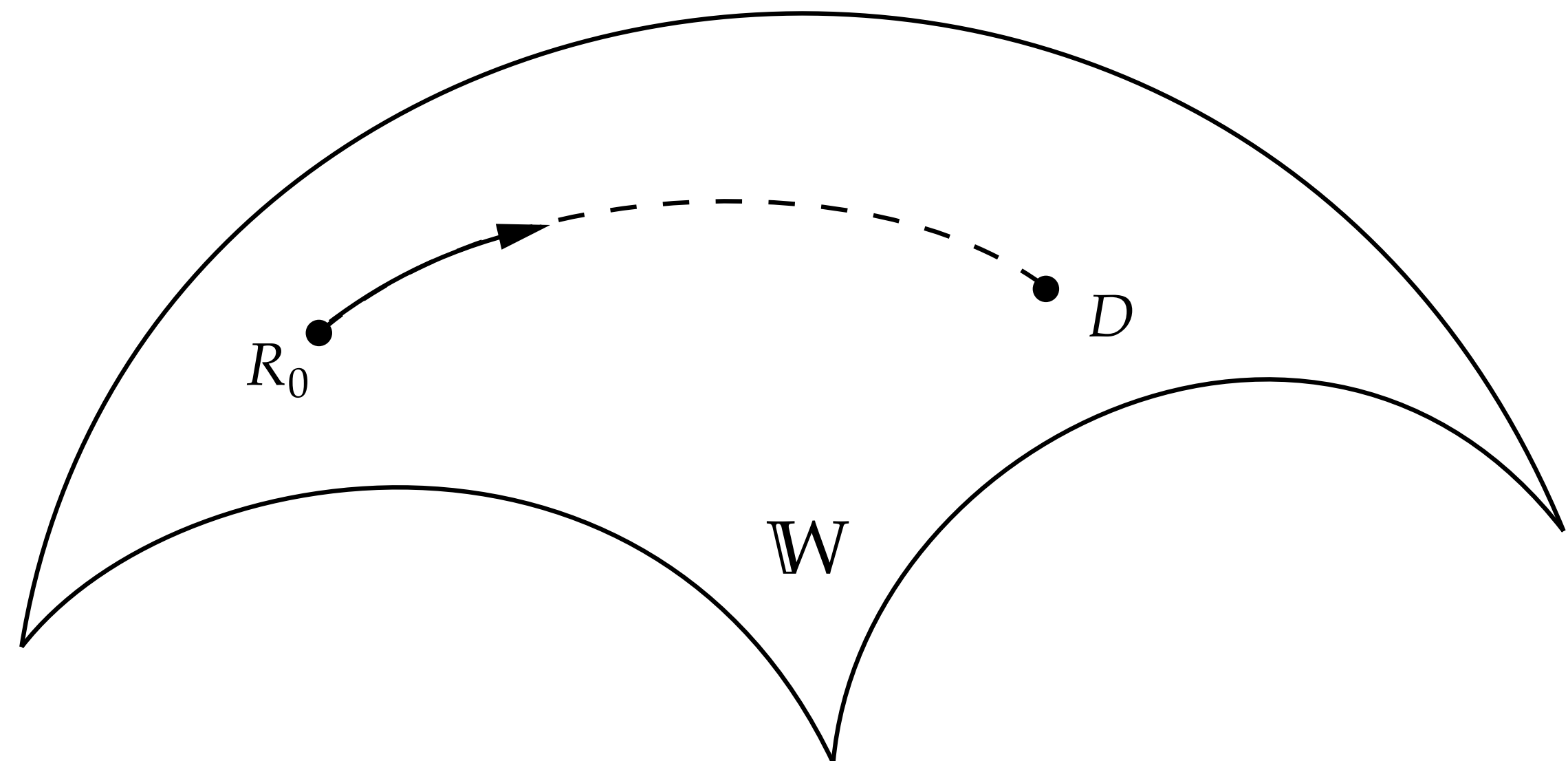


Fact: Wasserstein Space = Riemannian Manifold

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

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

Assignment map comes from OT

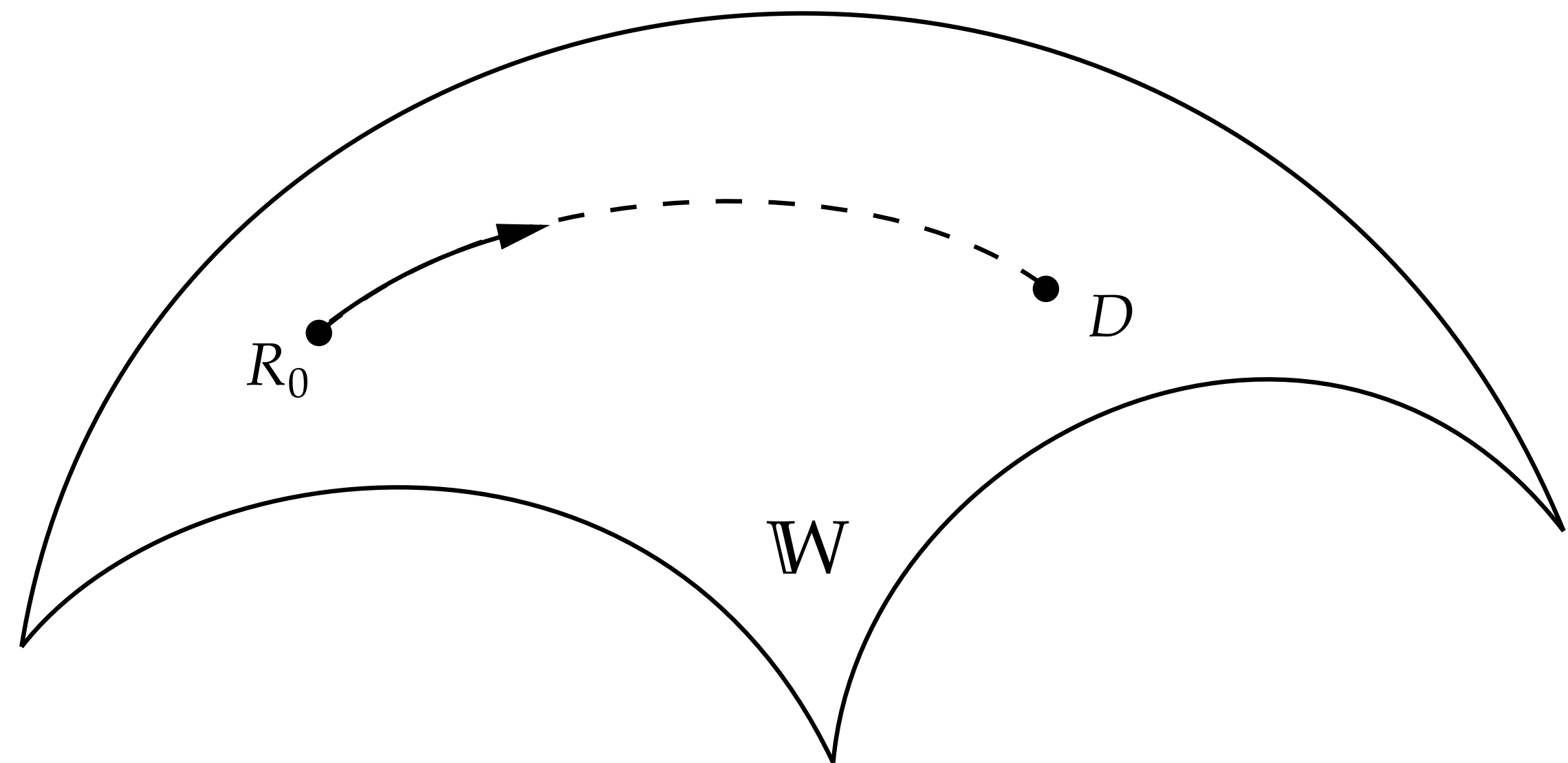


Fact: Wasserstein Space = Riemannian Manifold

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

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

Time schedule σ comes from OC problem

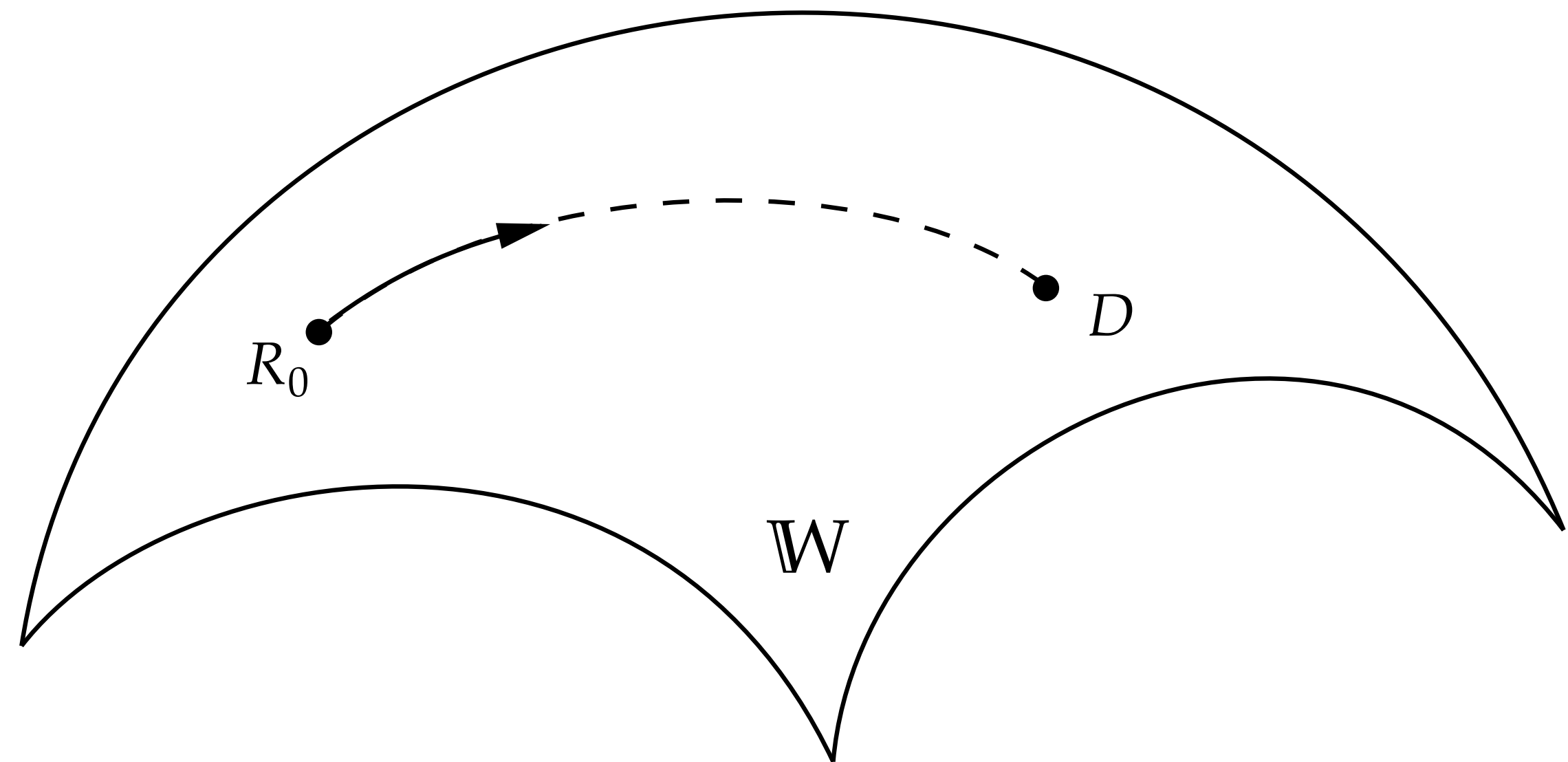


Fact: Wasserstein Space = Riemannian Manifold

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

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

- Can show assignments conserved \rightarrow particle trajectories independent!



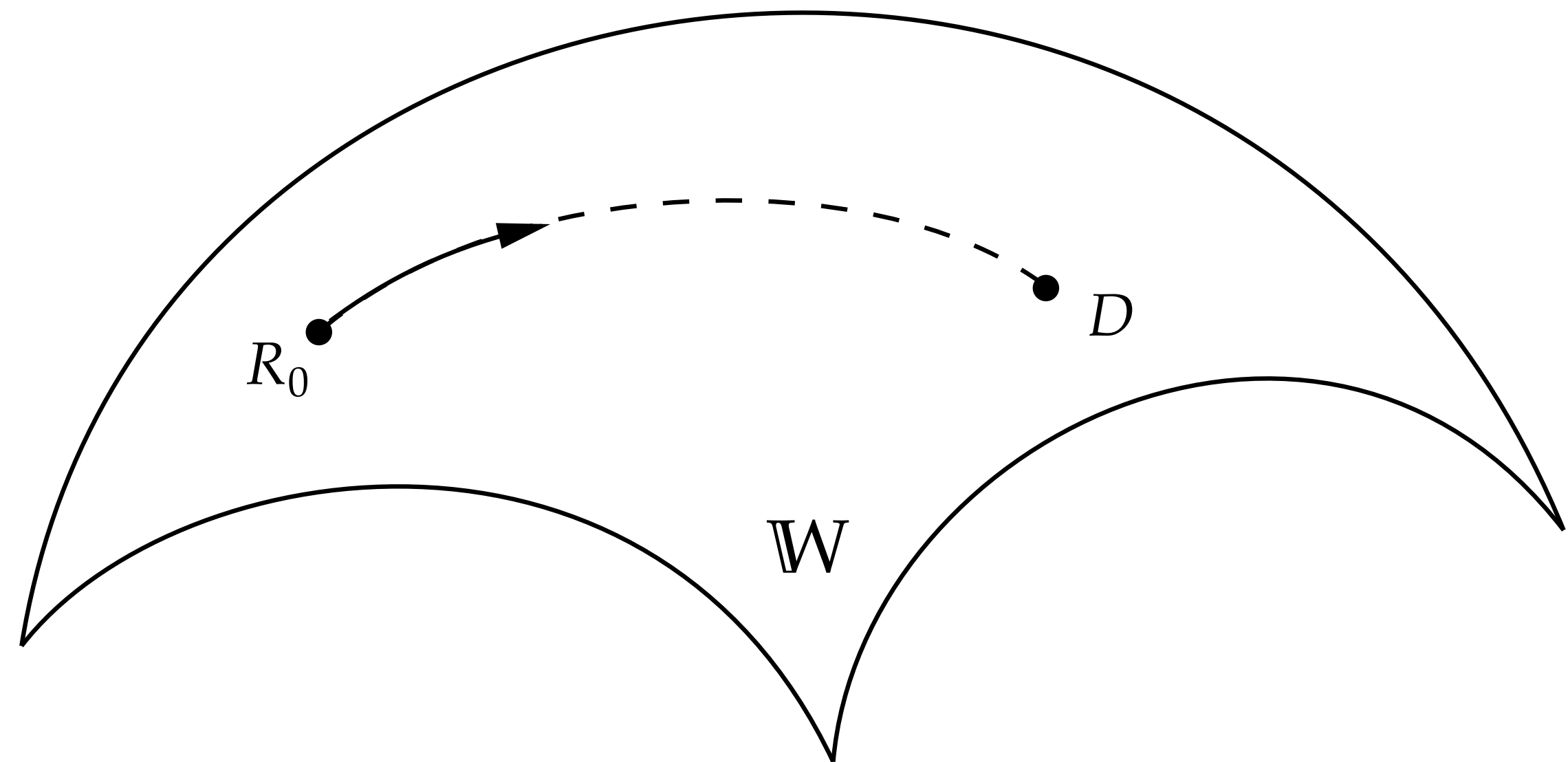
Fact: Wasserstein Space = Riemannian Manifold

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

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

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

$$V_t = \dot{\sigma}(t) (\bar{M}_{R_t \rightarrow D} - I)$$



Fact: Wasserstein Space = Riemannian Manifold

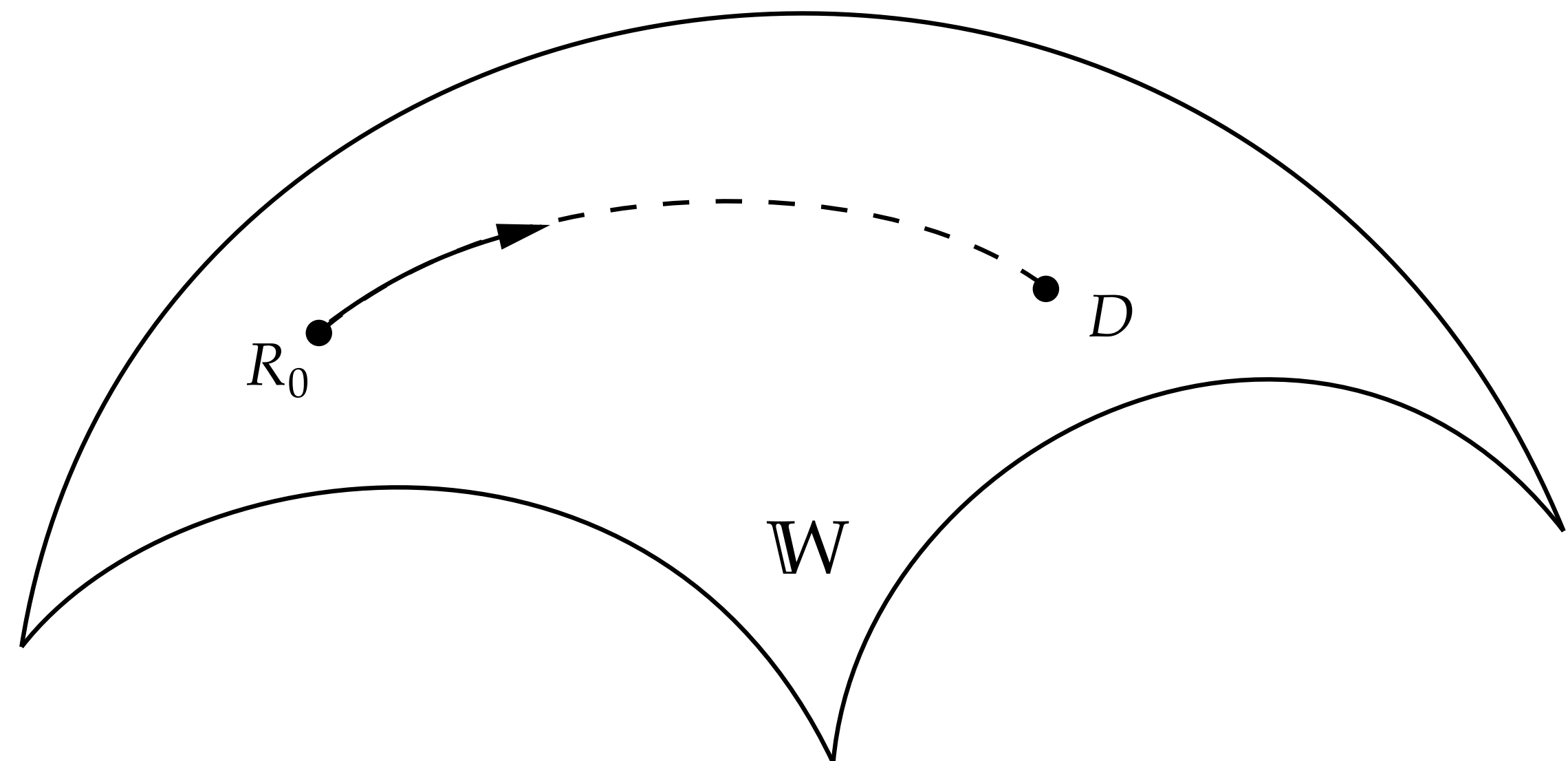
- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D :

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

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

$$V_t = \dot{\sigma}(t) (\bar{M}_{R_t \rightarrow D} - I)$$

Error vector



Fact: Wasserstein Space = Riemannian Manifold

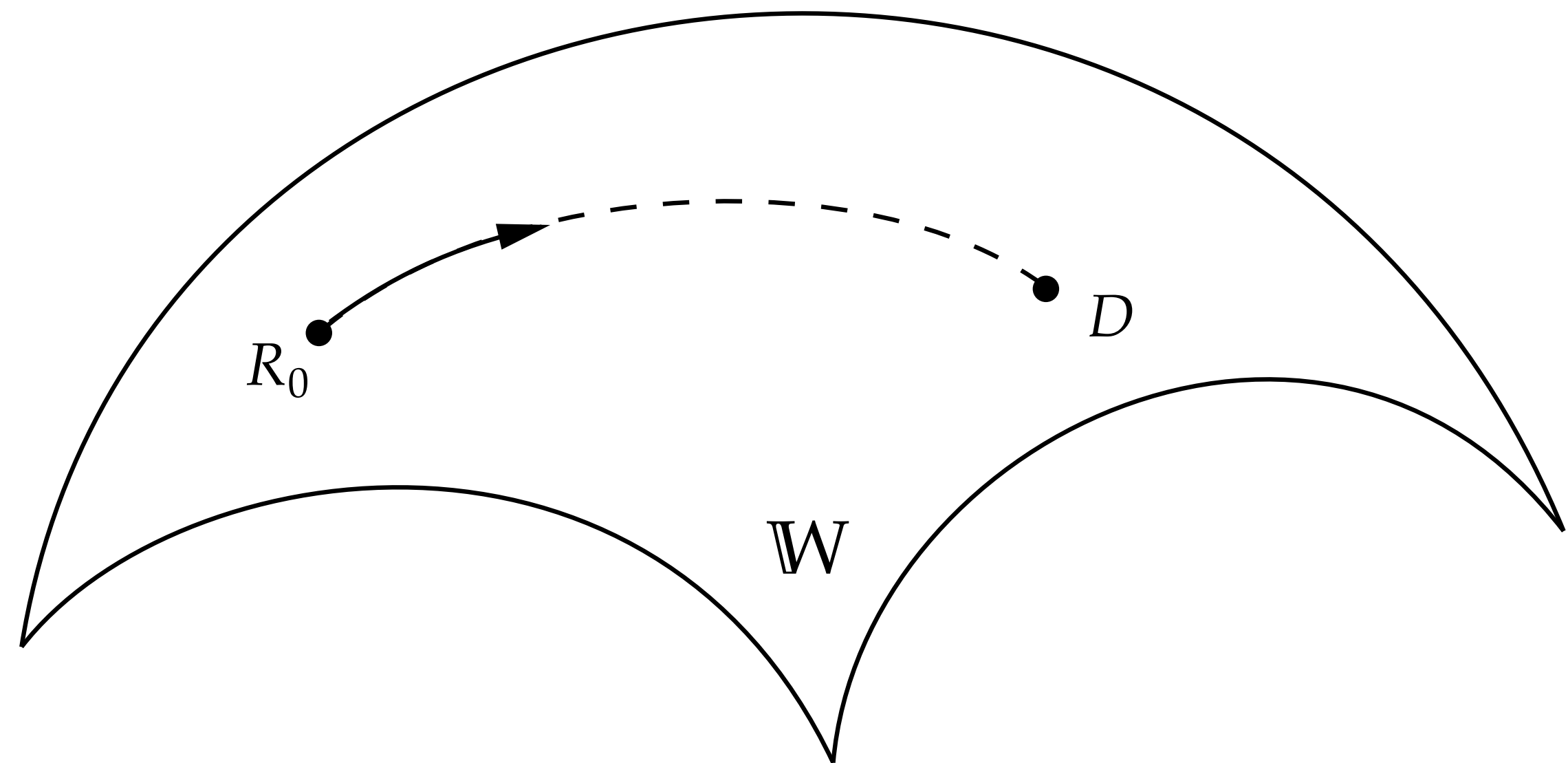
- Leverage geometric structure to solve problem
- Can show that R moves along Wasserstein geodesic towards D :

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

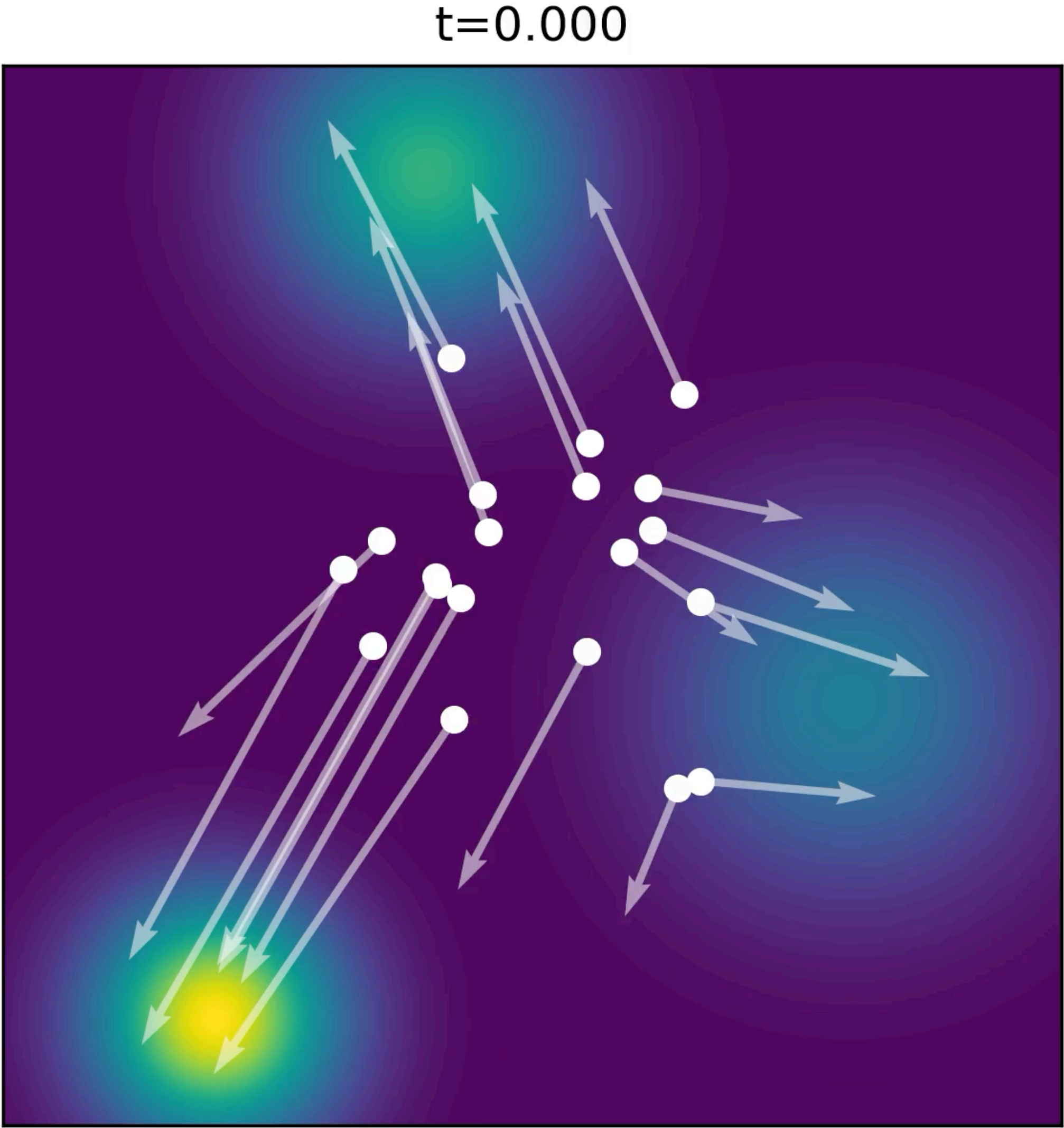
- Can show assignments conserved \rightarrow particle trajectories independent!
- Gives form of optimal controller:

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

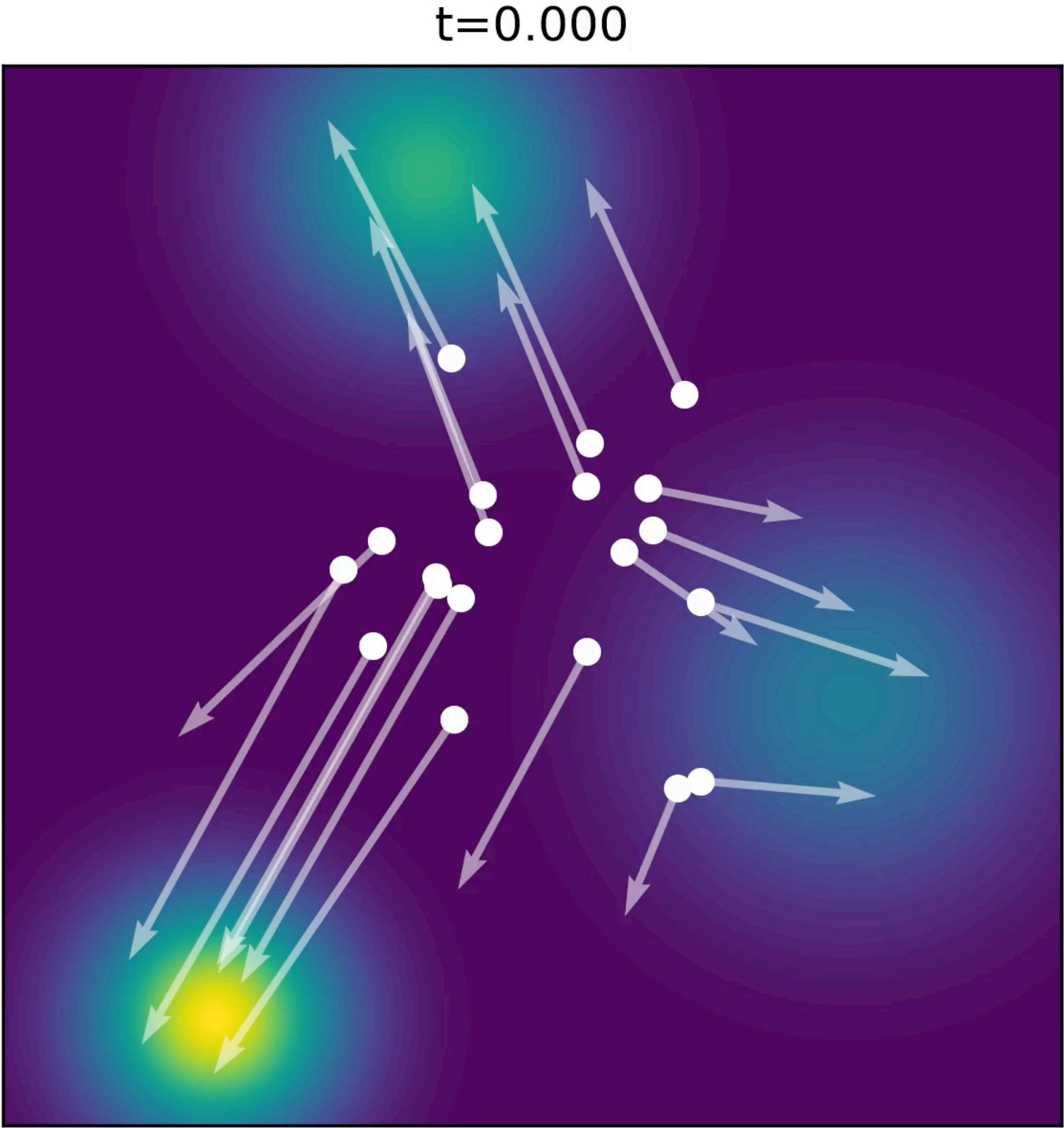
Rate of traversal



Regulation Problem: Simulation



Regulation Problem: Simulation



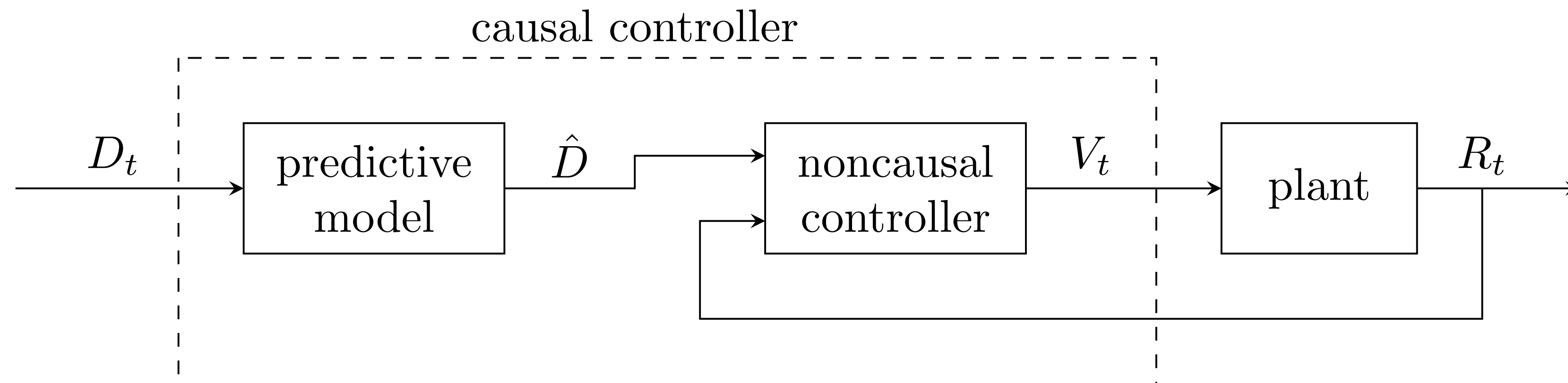
Back to Tracking Problem

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 forecast demand trajectory, use forecasted trajectory in necessary conditions in receding horizon scheme



Back to Tracking Problem

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

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

Back to Tracking Problem

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

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

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

MPC Algorithm for Time-Varying Demands

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

MPC Algorithm for Time-Varying Demands

- Forecast that future demand = current demand
- Compute optimal control, apply for Δ_t seconds
- Update demand and repeat
- Resulting controller has form:

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

MPC Algorithm for Time-Varying Demands

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

- Resulting controller has form:

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

- Heuristic, not optimal in general

MPC Algorithm for Time-Varying Demands

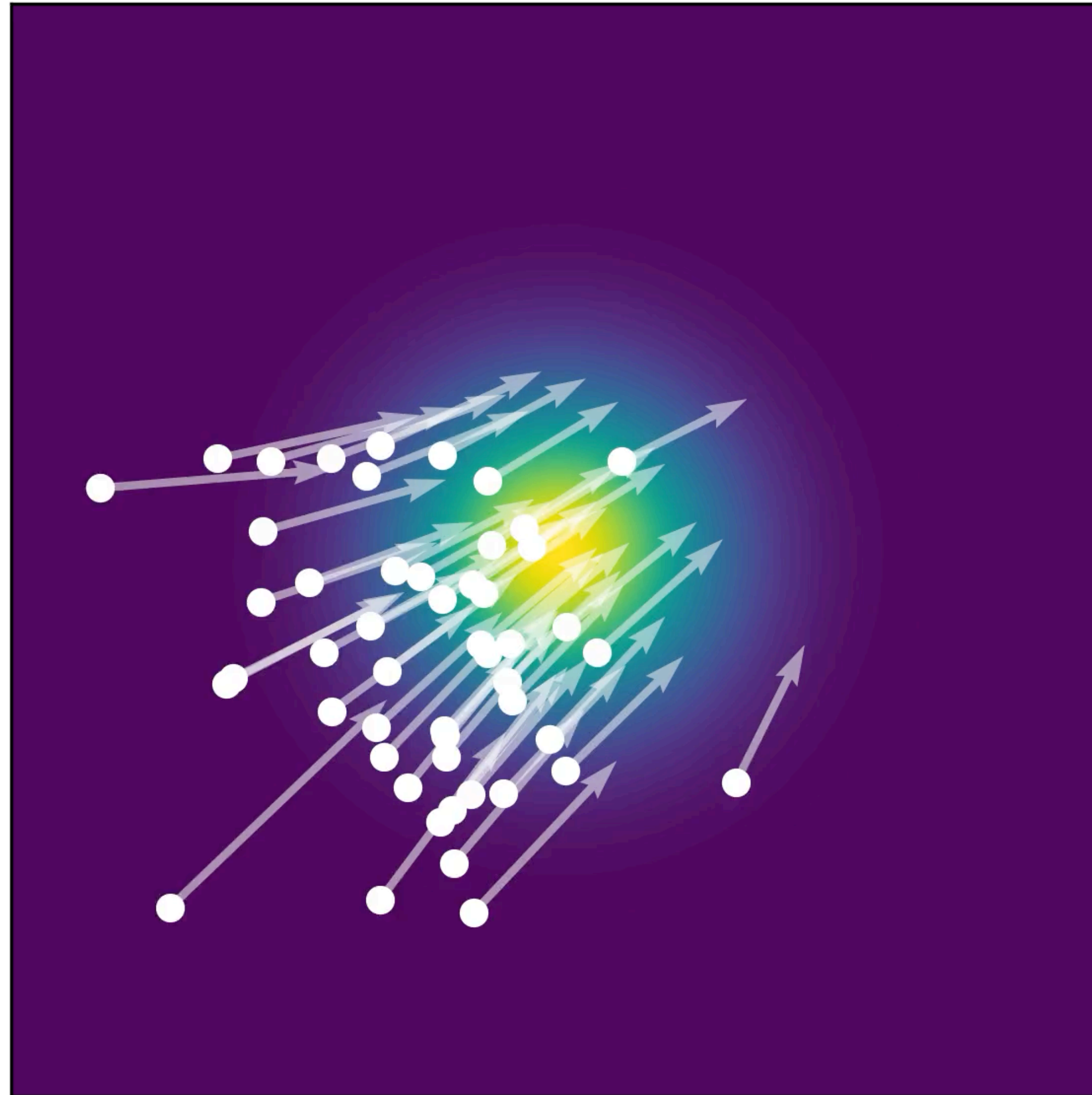
- Forecast that future demand = current demand
- Compute optimal control, apply for Δ_t seconds
- Update demand and repeat
- Resulting controller has form:

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

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

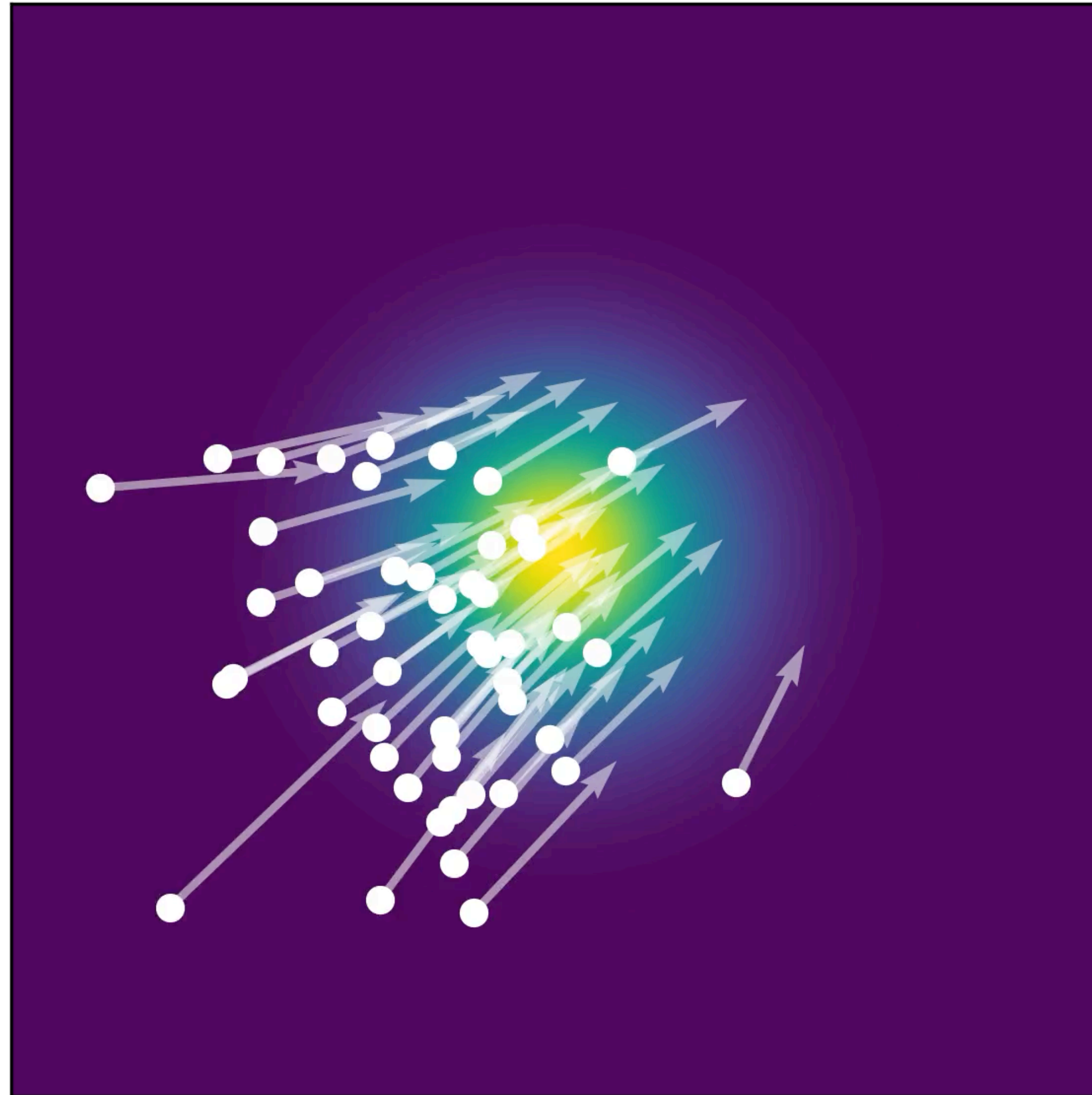
MPC Algorithm: Simulations

$t=0.000$



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?