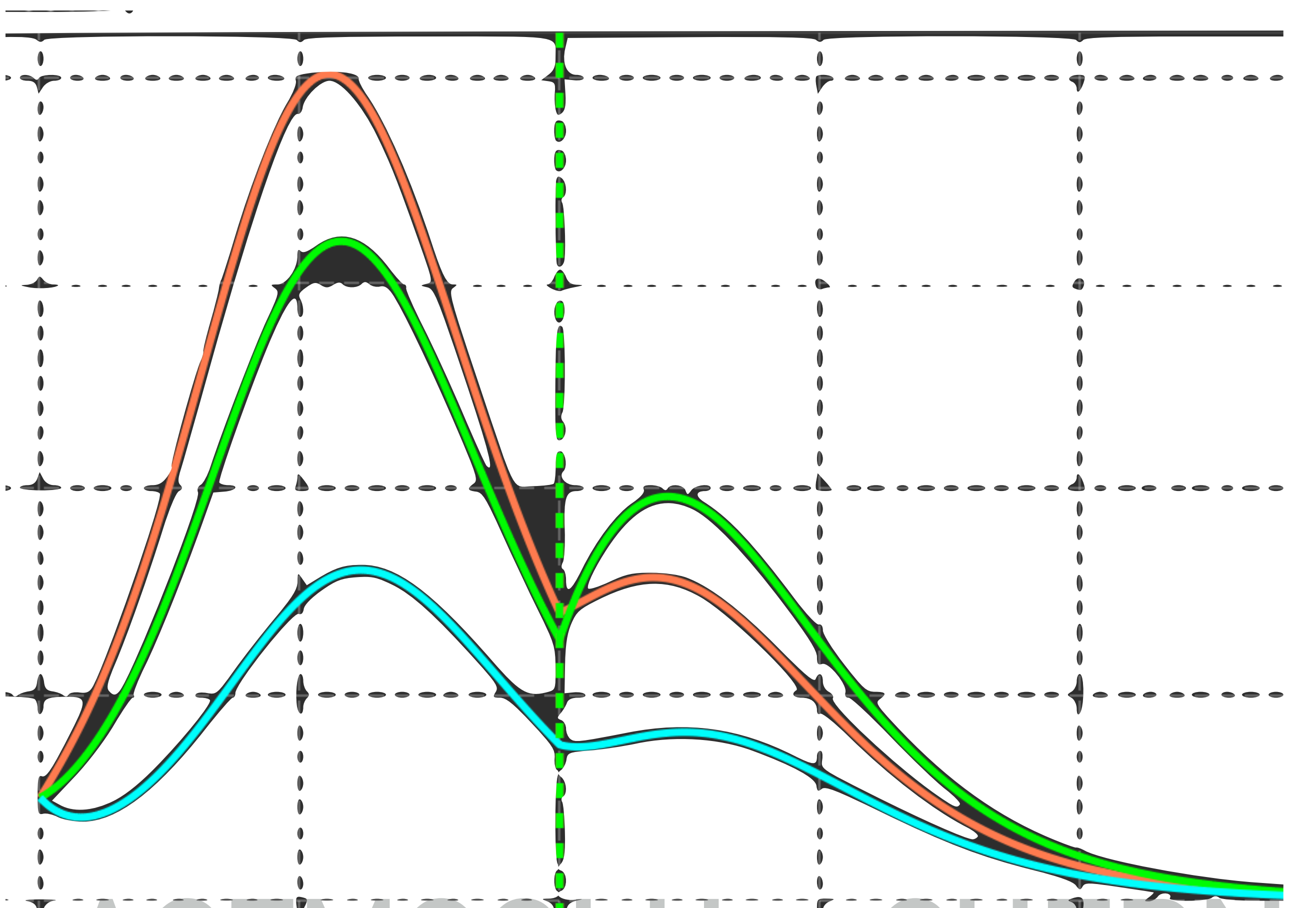


ACEMOGLU + CHERNOZHUKOV + WERNING + WHINSTON

(MIT) (MIT) (MIT) (MIT& SLOAN)

OPTIMAL TARGETED LOCKDOWNS IN A MULTI-GROUP SIR MODEL

THIS PAPER

- **Policy analysis for COVID-19…**
  - **Epi:** herd immunity, effect of mitigation, timing, etc.
  - **Econ:** costs of lockdowns, optimal policy, etc.
- **COVID-19:** very asymmetric effects

Age Group	Mortality rate
20-49	0.001
50-64	0.01
65+	0.06

(Ferguson, 2020)

This paper: simple multi-group model

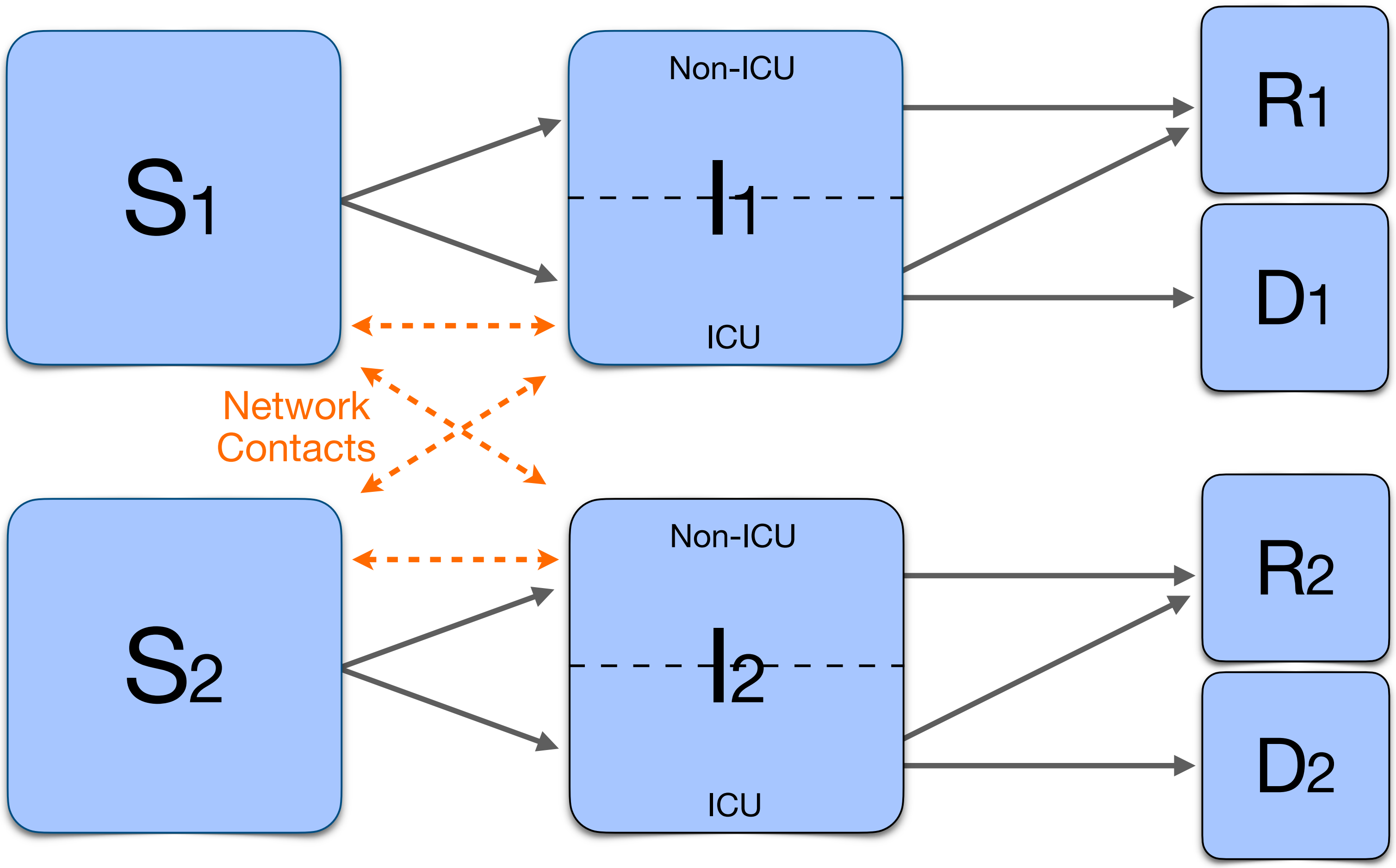
+

explore optimal policy implications

IMPORTANT CAVEATS

- We are obviously not epidemiologists, but we made our best effort to integrate Econ with Multi-Group SIR models as well as calibrate the model.
- Used Ferguson R0 =2.4 x .8 (to reflect mask use etc).
- Not a literal policy prescription: The goal is to find areas of policy focus where welfare gains are the largest.
- Private behavior can help ease some implementational aspects.

SIR MODEL + ECONOMIC COSTS



- ``Lockdowns"
  - benefit: reduce interactions, lower infections
  - costs: lost output
- Optimal Control timing of lockdown

TWO OBJECTIVES

- Trade-off between deaths and economic losses
- Costs:
  - Lives lost $= \sum_j D_j(T)$
  - Economic losses: $\int_0^T \sum_j \psi_j(t)dt$
- Goal: Choosing policies to minimize overall losses:

$$\int_0^T \sum_j \Psi_j(t)dt + \chi \sum_j D_j(T).$$

- Or, Safety-Focus: Min EL st LL < .001
- Or, Econ-Focus: Min LL st EL < .10

ECON LOSS AND DYNAMIC STRUCTURE:

The economic losses for group j are given by

$$\begin{aligned}\Psi_j(t) &= (1-\xi_j)w_jS_j(t)L_j(t) + \\ &+ (1-\xi_j)w_jE_j(t)(1-\eta_k^E(1-L_j(t))) + \\ &+ (1-\xi_j)w_jI_j(t)(1-\eta_k^E\eta_k^I(1-L_j(t))) + \\ &+ (1-\xi_j)w_j(1-\kappa_j)R_j(t)L_j(t) + \\ &+ w_j\Delta_j\iota_j\delta_j^d(t)I_j(t),\end{aligned}$$

$$\dot{E}_j = M_j(S,E,I,R,L;\alpha)\beta(1-\theta_jL_j)S_j\sum_k \rho_{jk}\eta_k^E\eta_k^I(1-\theta_kL_k)I_k - \gamma_j^E E_j$$

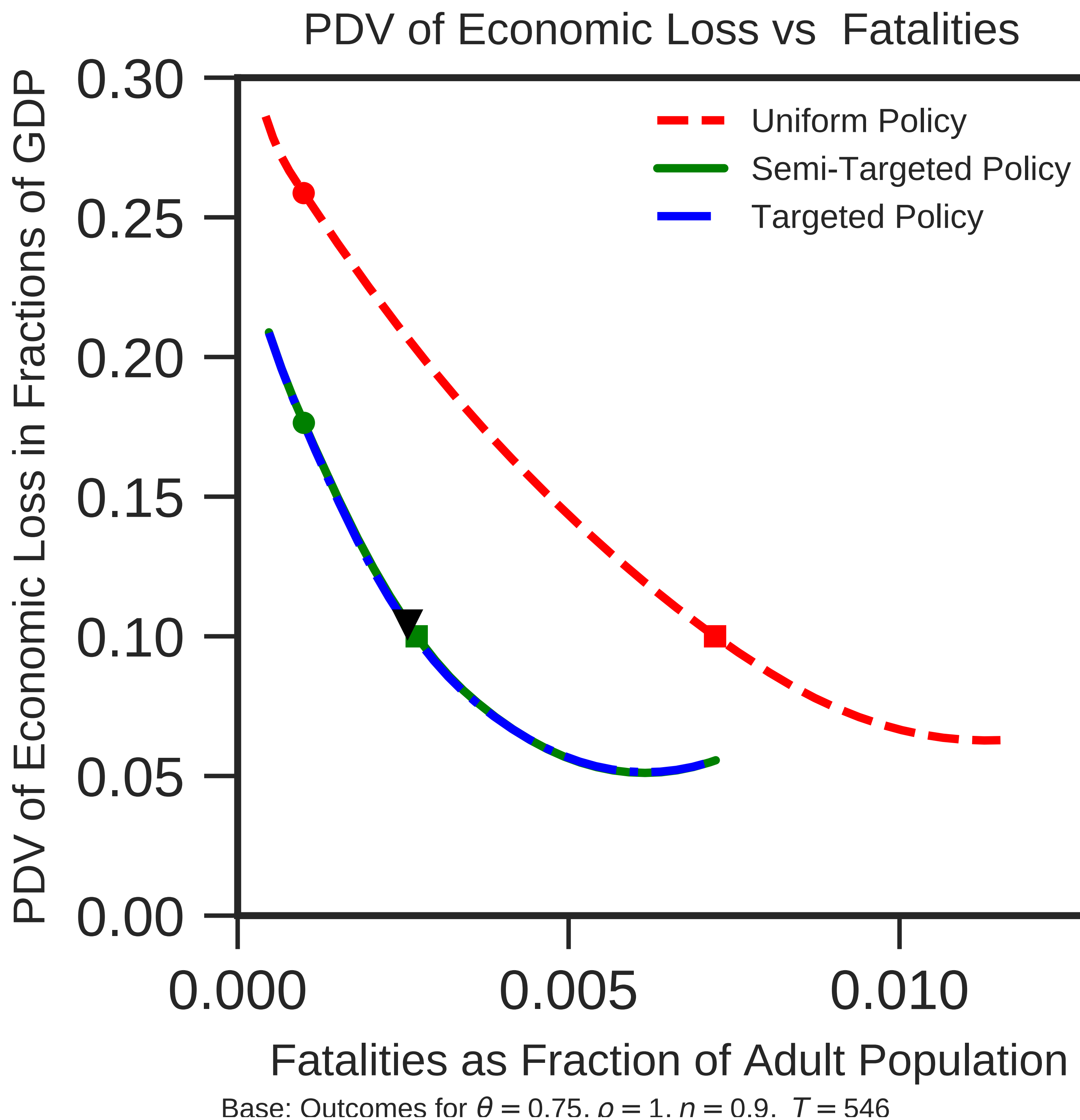
$$\begin{aligned}\dot{S}_j &= -\dot{E}_j - \gamma_j^E E_j, \\ \dot{I}_j &= \gamma_j^E E_k - \gamma_j^I I_j, \\ \dot{D}_j &= \delta_j^d H_j, \\ \dot{R}_j &= \delta_j^r H_j + \gamma_j^I(I_j - H_j),\end{aligned}$$

EVOLUTION

OF (S, I, D, R) by Group

IN SEIR STRUCTURE

FRONTIER

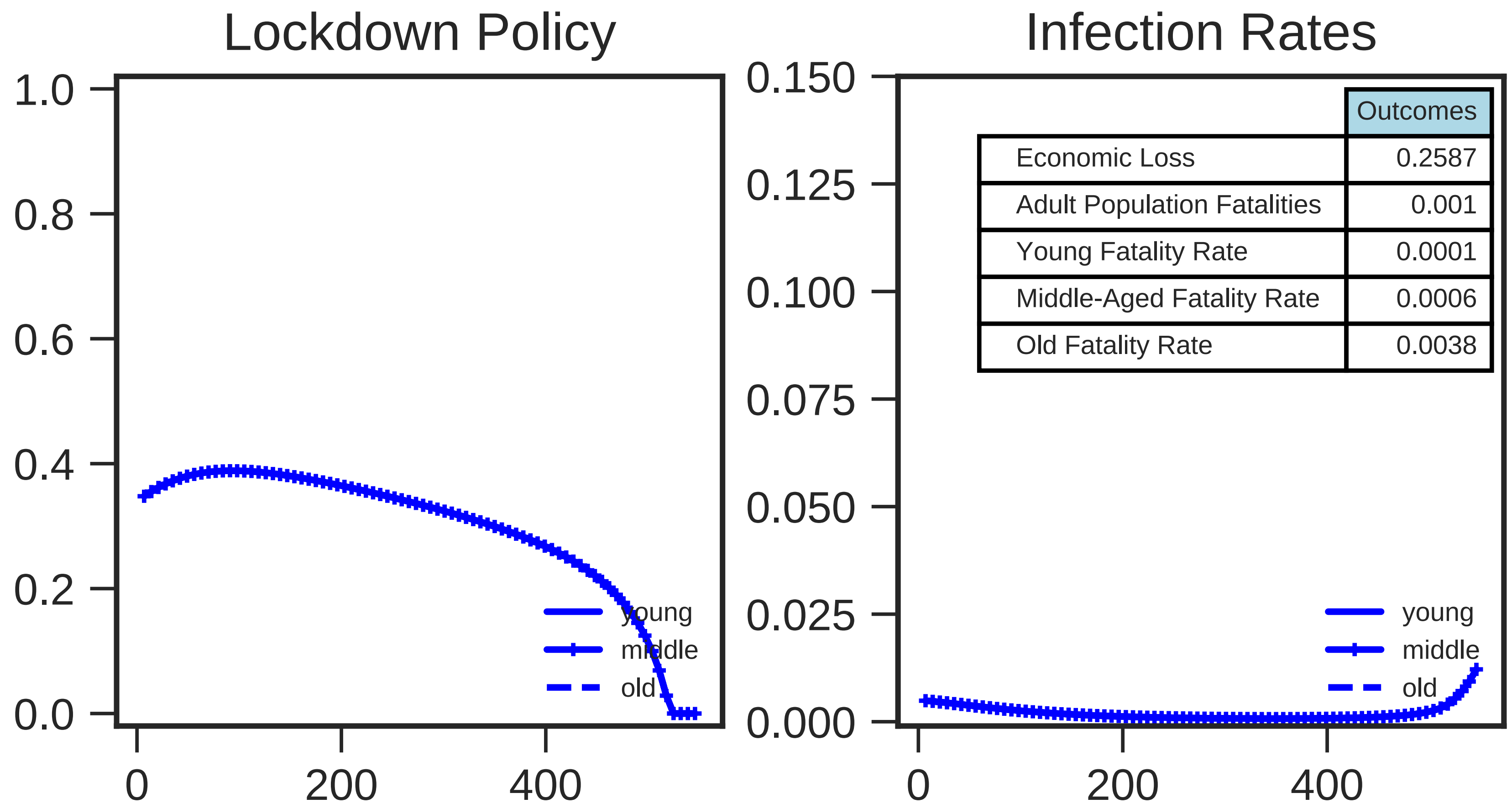


Large gains for Semi-Targeting

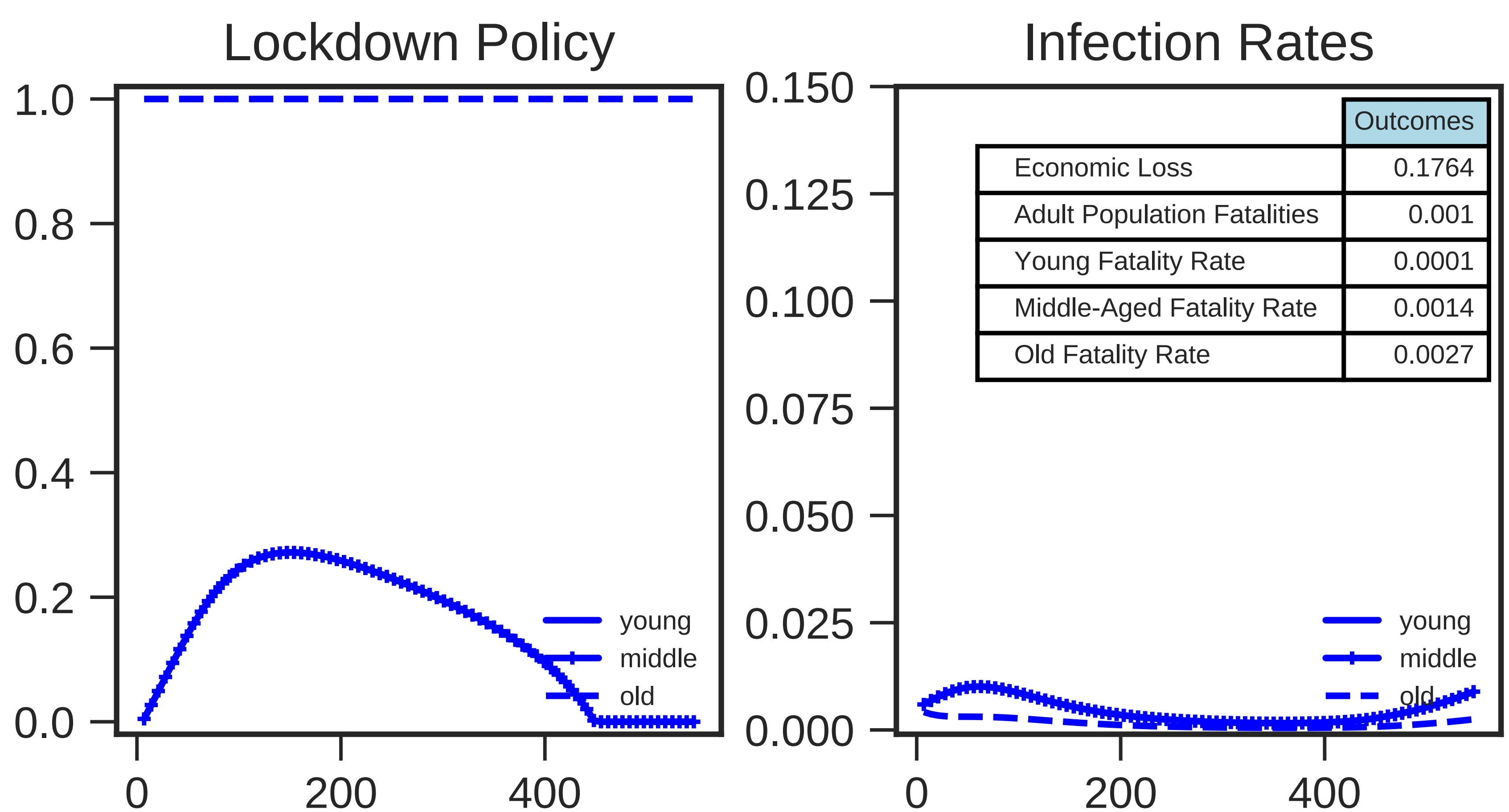
Small gains for Full-Targeting

BENEFIT OF SEMI-TARGETING

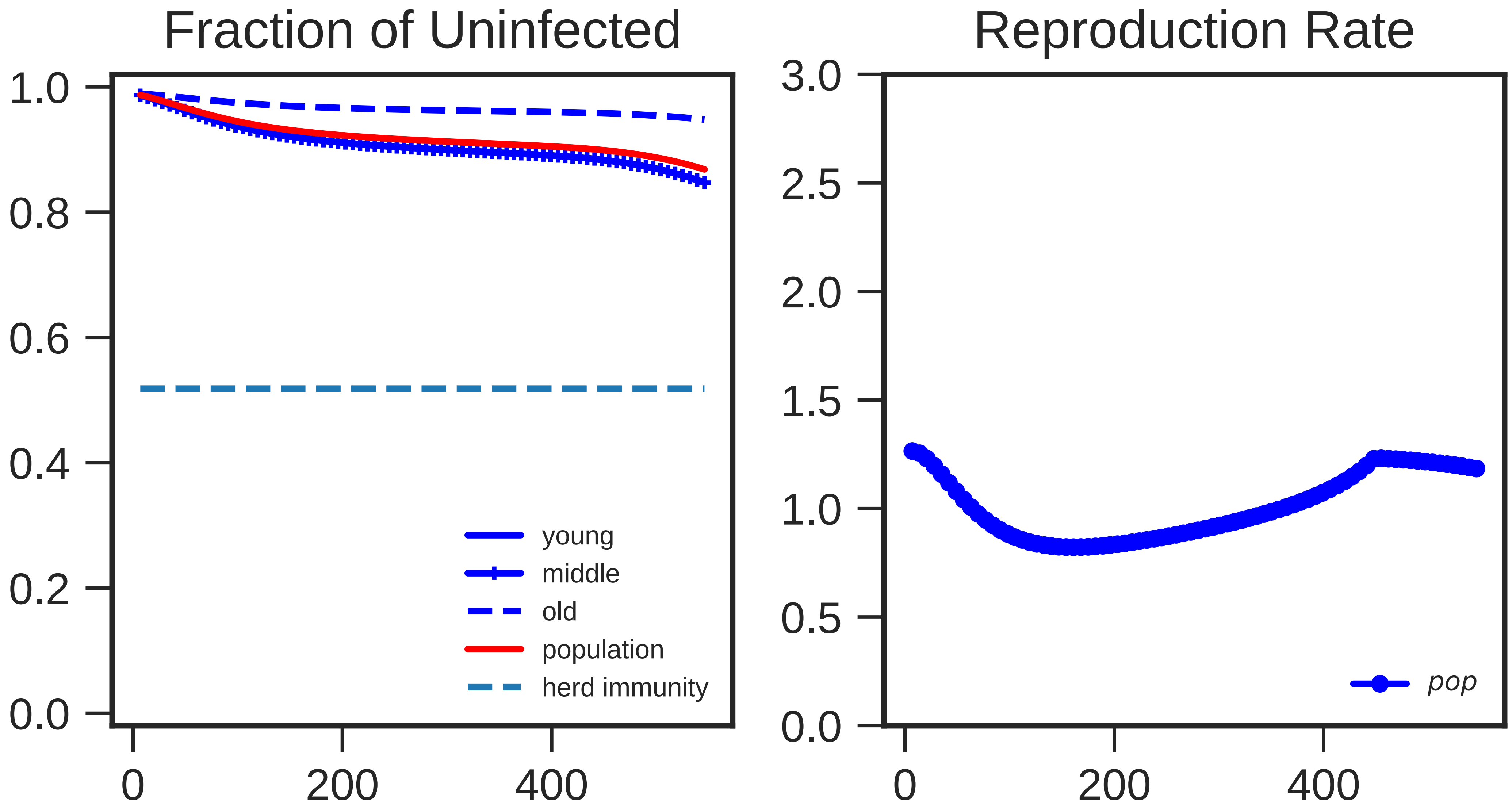
Uniform Policy



Semi-Targeted Policy



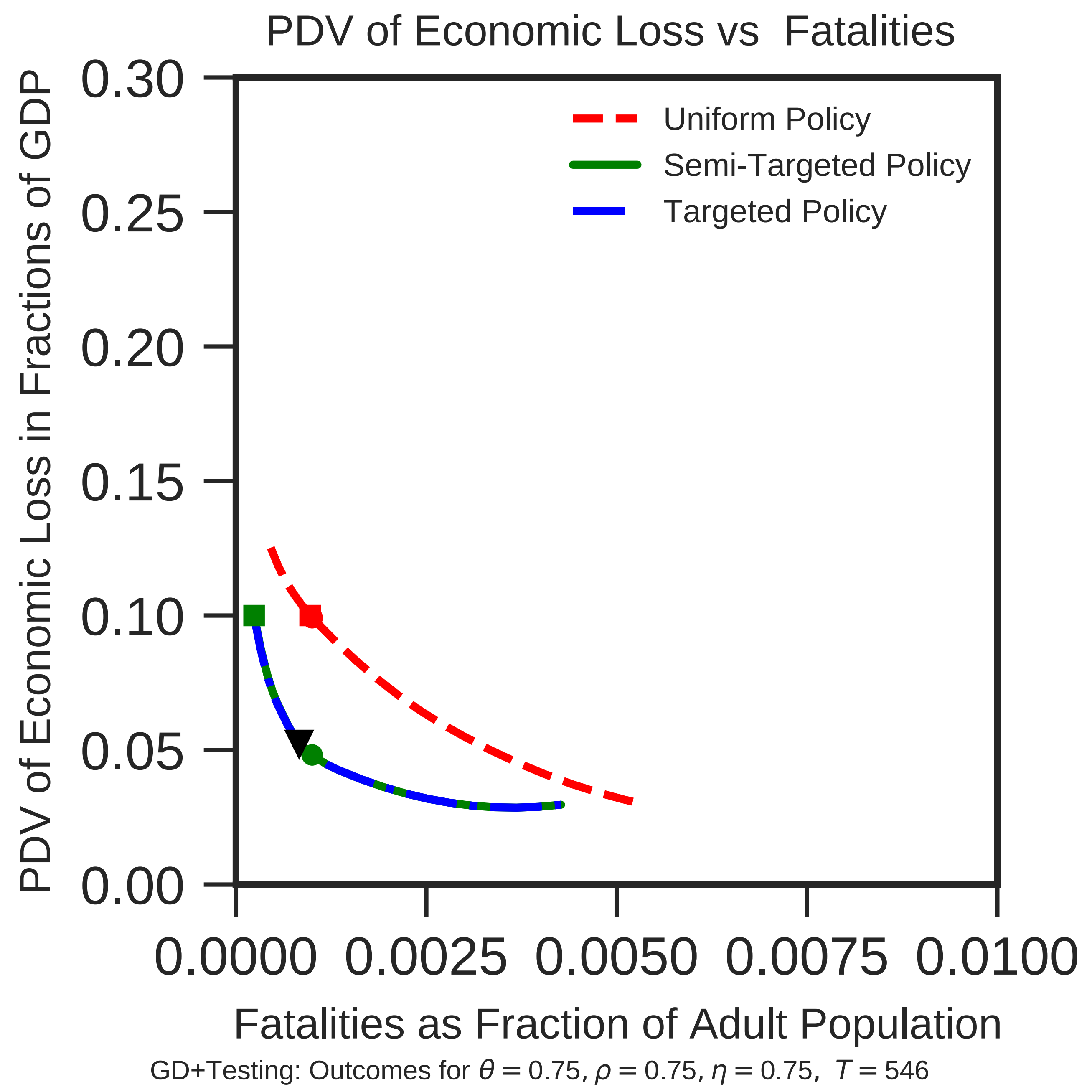
HERD IMMUNITY IS NOT THE GOAL



Base: Safety Focused Optimal SemiTargeted Policy for $\theta = 0.75$ $\eta = .9$ $\rho = 1.0$

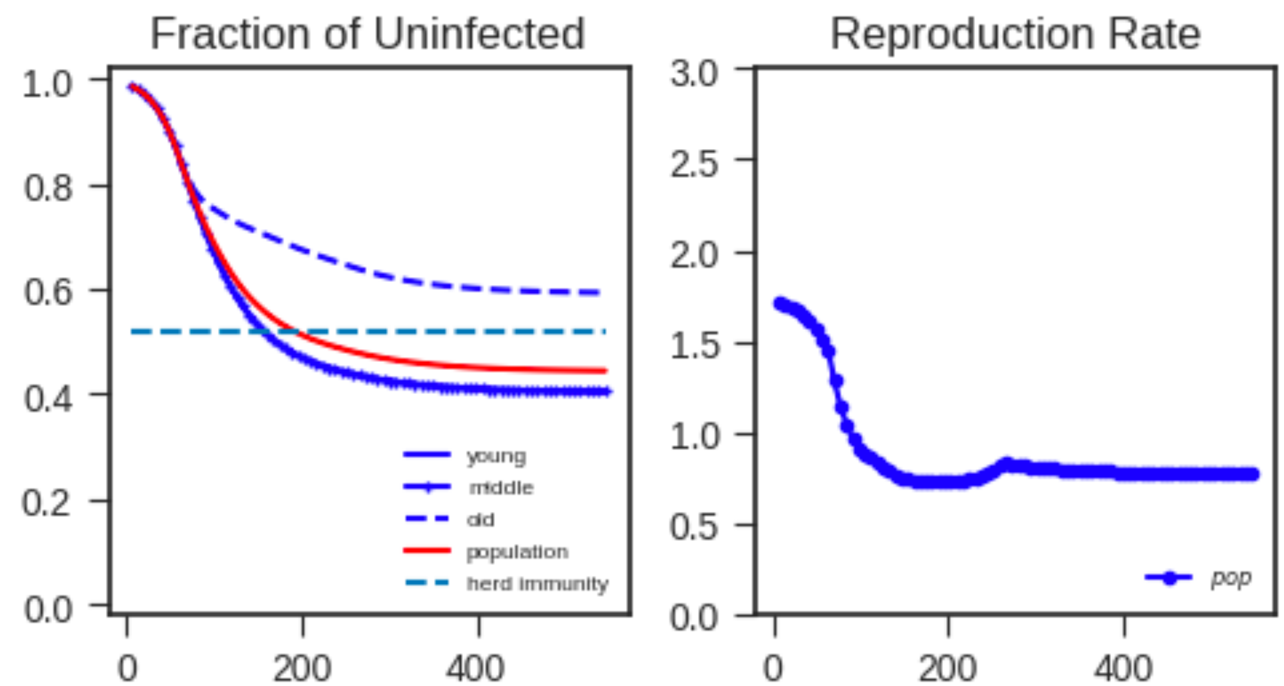
FINDINGS

- Large gains from targeted policy
- Most gains from simple semi-targeted policies: treat 65+ group differentially
- Non-zero lockdown for young
- Testing is important
- COMBO (test+dist+targ)

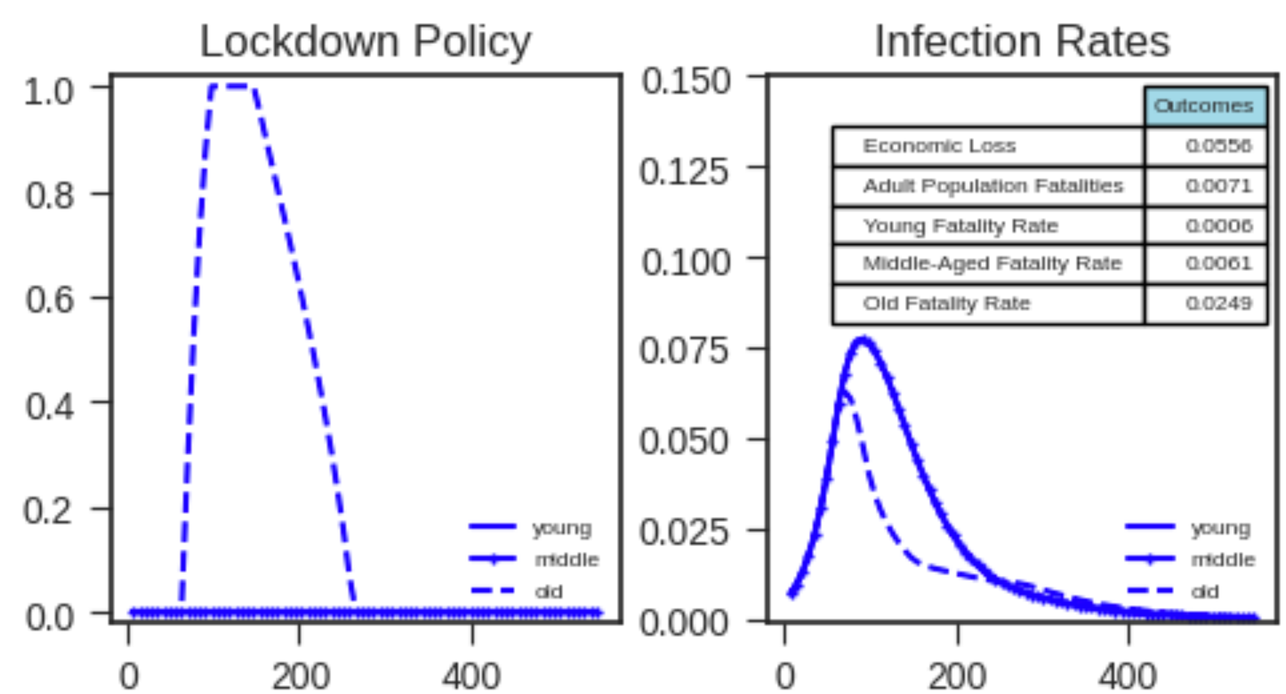


FINDINGS: “OBVIOUS” THINGS THAT DIDN’T WORK

- “Quasi-SWEDISH”– mild shield the old, let everyone go free and attain herd immunity.
  - Sub-optimal policy. Even herd immunity is attained sub optimally. (Total Infected > Herd Immunity Threshold).
- “OBVIOUS”: Treat <45 and 45-65 very differently. Wrong again. Small gains only.



Swedish: Fixed χ Optimal SemiTargeted Policy for $\theta = 0.75$ $\eta = .9$ $\rho = 1.0$



Swedish: Fixed χ Optimal SemiTargeted Policy for $\theta = 0.75$ $\eta = .9$ $\rho = 1.0$

COMPUTATION

Optimal Dynamic Control

$$\min J(L(\cdot), X(\cdot)) = \int_0^T \ell(L(t), X(t), t)dt$$

where $L(\cdot)$ is vector of control functions; $X(\cdot)$ is a vector of state functions; t is time from 0 up to $T \leq \infty$, $\ell(\cdot, \cdot, \cdot)$ is incremental loss, such that

$$\begin{array}{ll|l} \dot{X}(t) = f(X(t), L(t), t) & & \text{state evolution} \\ h(X(t), L(t), t) \leq 0 & & \text{path constraints} \\ e[X(0), L(0)] \leq 0 & & \text{initial constraints} \end{array}$$

COMPUTATION:

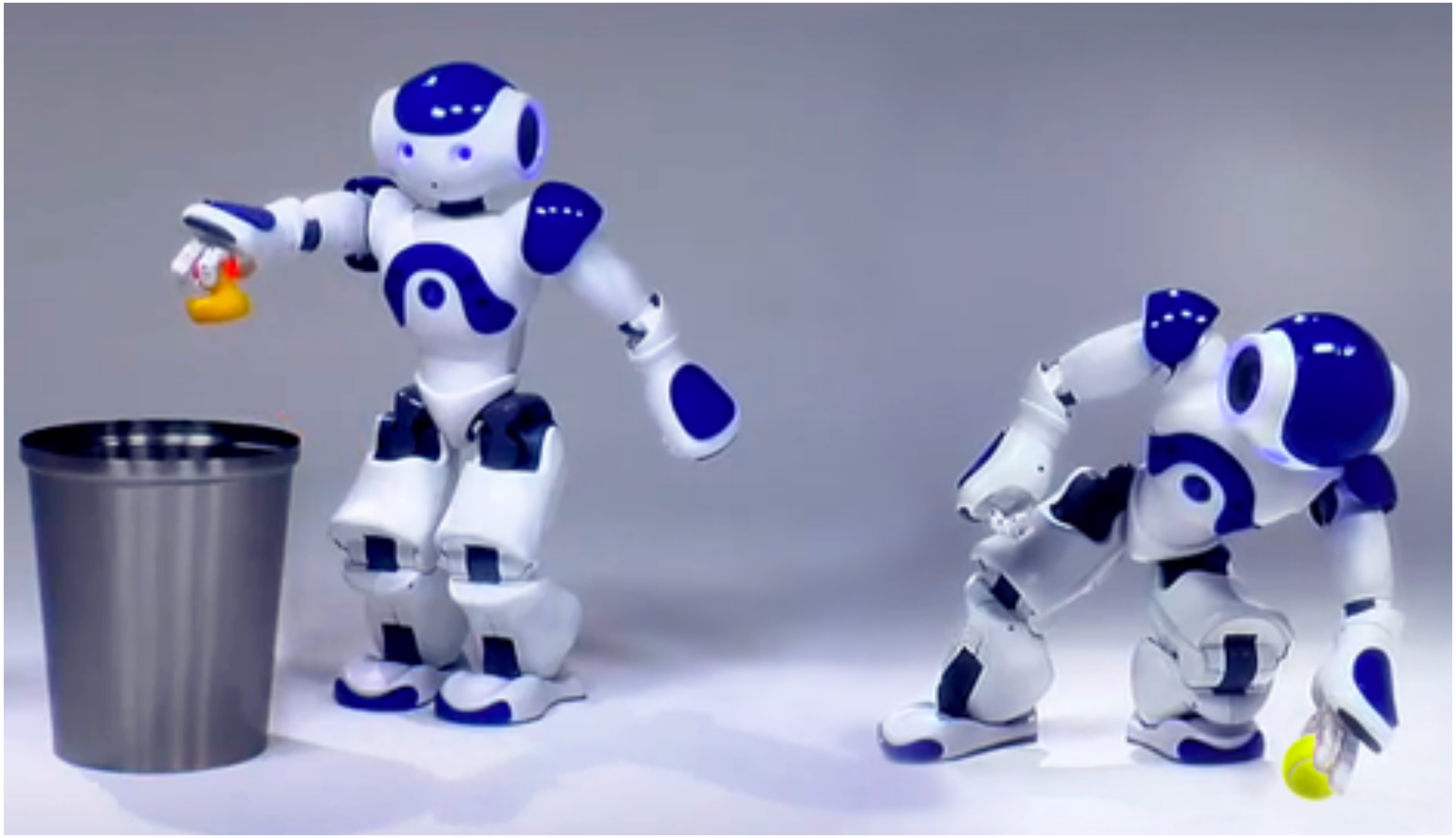
- "Indirect solution" methods:
  - Dynamic Programming: Value Function Iterations.
  - Indirect methods: solve a dynamic system – Hamiltonina– via first-order conditions subject to boundary conditions.
- "Direct solution" method: which is widely *used in large scale industrial engineering applications*. Write as a large Nonlinear Programming Problem with flexibly parameterized – nonparametric– policy and state functions and optimize.

See Wikipedia.

GEKKO
DYNAMIC OPTIMIZATION

COMPUTATION: DIRECT SOLUTION

USED IN MOTION ROBOTICS



COMPUTATION: DIRECT SOLUTION

$$\min_{Z} F(Z)$$

over discretized (over time) control and policy variables

$$Z = \{L_j(t_k), X_m(t_k) : 1 \leq j \leq J, 1 \leq m \leq M, 1 \leq k \leq K\} \in \mathbb{R}^{JKM}.$$

subject to equality and inequality constraints:

$$g(Z) \leq 0, h(Z) = 0 \quad (L \text{ constraints})$$

- Size of the problem: $L + JKM$ is several thousands.
  - IOPT: Interior Point Optimal Nonlinear Programming
  - GEKKO interface: takes Optimal Control Problem, reformulates, passes to IPOPT.

CONNECTIONS TO "AI"

"AI" is a collection of methods, developed mostly by engineering and CS community:

- nonparametric (extremely high-dimensional) parameterization of policy P;

  beautiful art in many applications

- "train" policy by minimizing objective function $F(P)$ using closed-form gradients;
- over-parameterization and adding noise to gradients helps optimization.
- gives good practical solutions, though "global" optima are not guaranteed.

THANK YOU

PYTHON CODE IN THE CLOUD (Google Colab):

https://colab.research.google.com/drive/16zhsso-NzNbxn9C_MMP4lqzdVA2N-xxt?usp=sharing

GEKKO Dynamic Optimization

https://apmonitor.com/wiki/index.php/Main/GekkoPythonOptimization