From HeartSteps to HeartBeats: Personalized Decision-making

Raaz Dwivedi

HARVARD UNIVERSITY

Massachusetts Institute of Technology

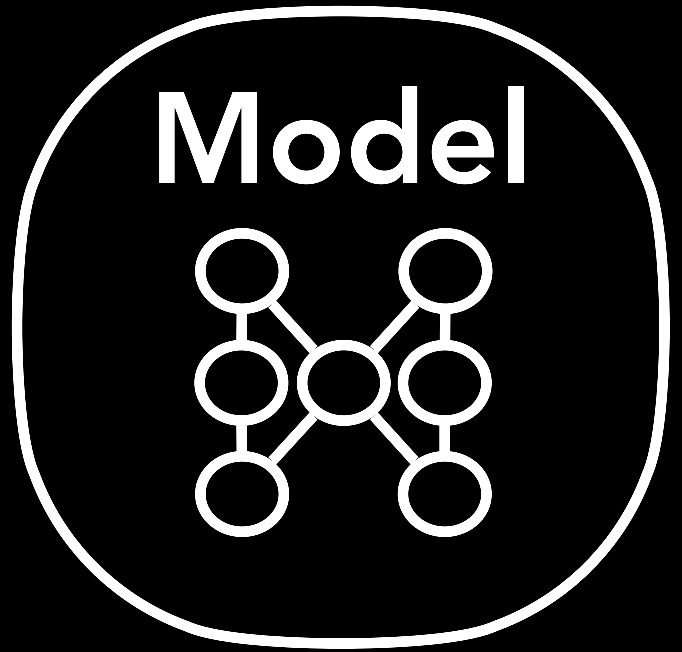
Stanford University, OIT Seminar, Jan 25

Personalized Decision-making

Personalized Decision-making

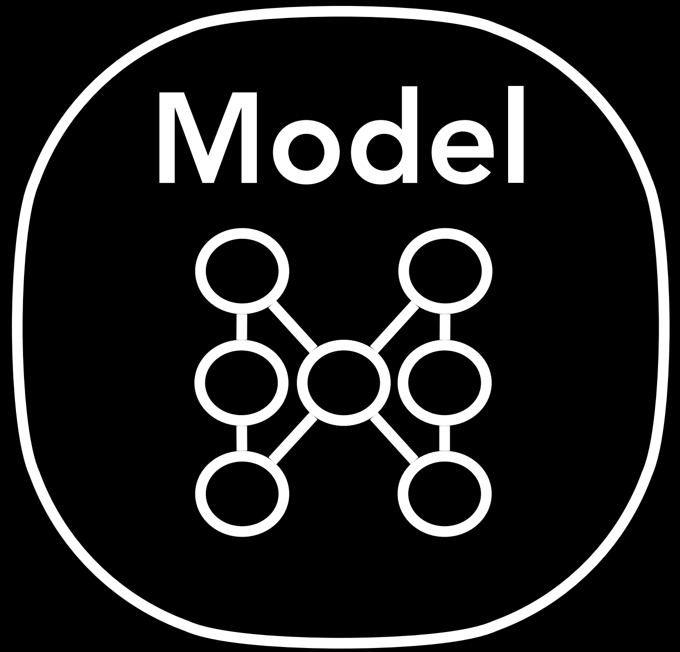
Driven by
extensive data collection,
decreasing cost of computation,
synergy between disciplines

Personalized Decision-making



Driven by
extensive data collection,
decreasing cost of computation,
synergy between disciplines

Personalized Decision-making

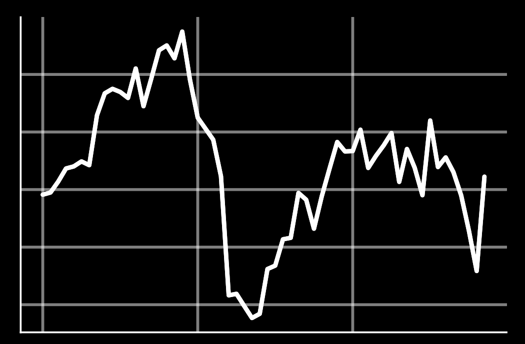


Driven by
extensive data collection,
decreasing cost of computation,
synergy between disciplines

Medical records

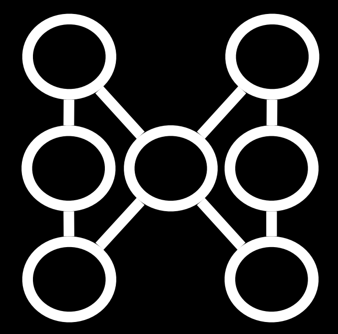
Observational studies

Data



Personalized Decision-making

Model



Driven by
extensive data collection,
decreasing cost of computation,
synergy between disciplines

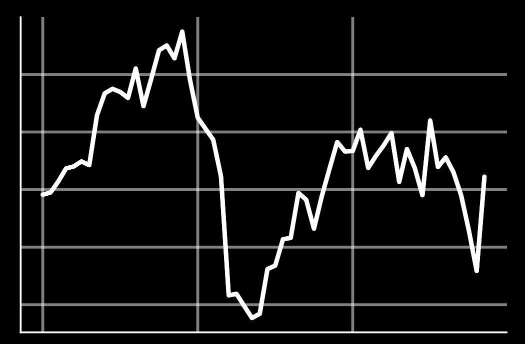
Drug trial

Randomized experiments

Observational studies

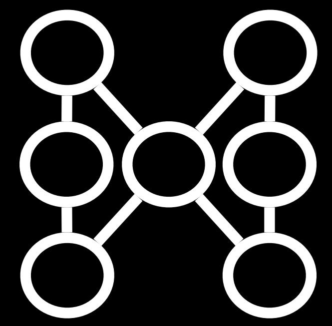
Medical records

Data



Personalized Decision-making

Model



Driven by
extensive data collection,
decreasing cost of computation,
synergy between disciplines

Drug trial

Randomized experiments

Medical records

Observational studies

Sequential experiments

Mobile health

Data

Personalized Decision-making

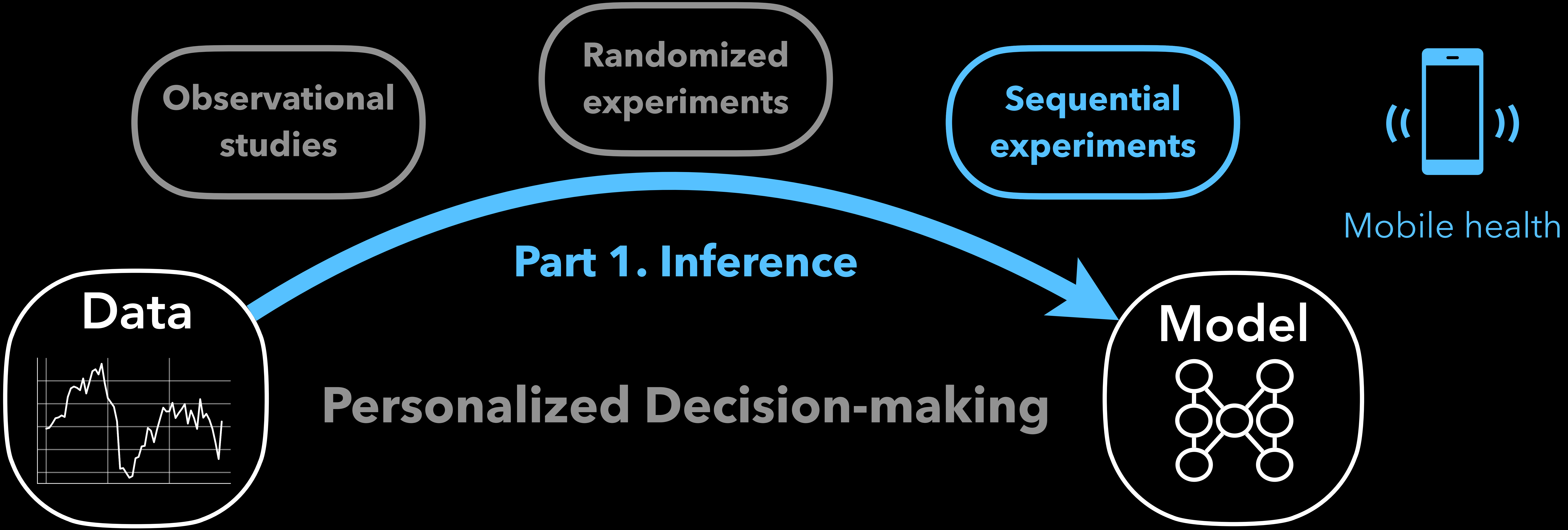
Model

Driven by
extensive data collection,
decreasing cost of computation,
synergy between disciplines

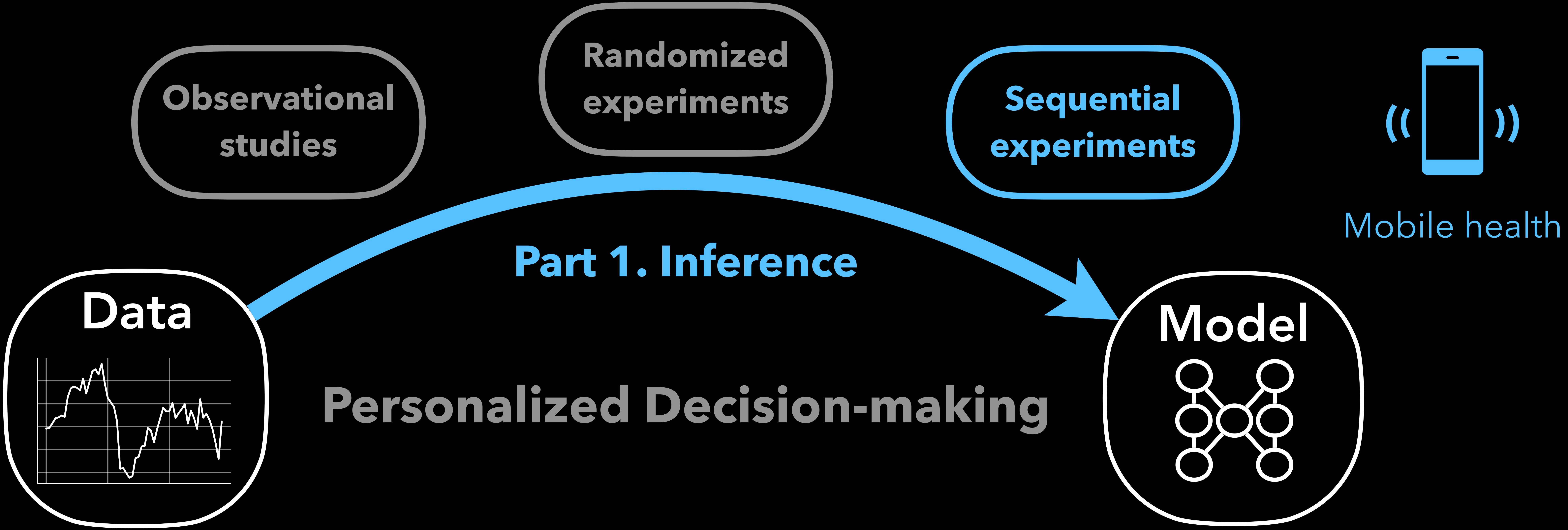
1. Use **real data** to infer decision's effect

Driven by
extensive data collection,
decreasing cost of computation,
synergy between disciplines

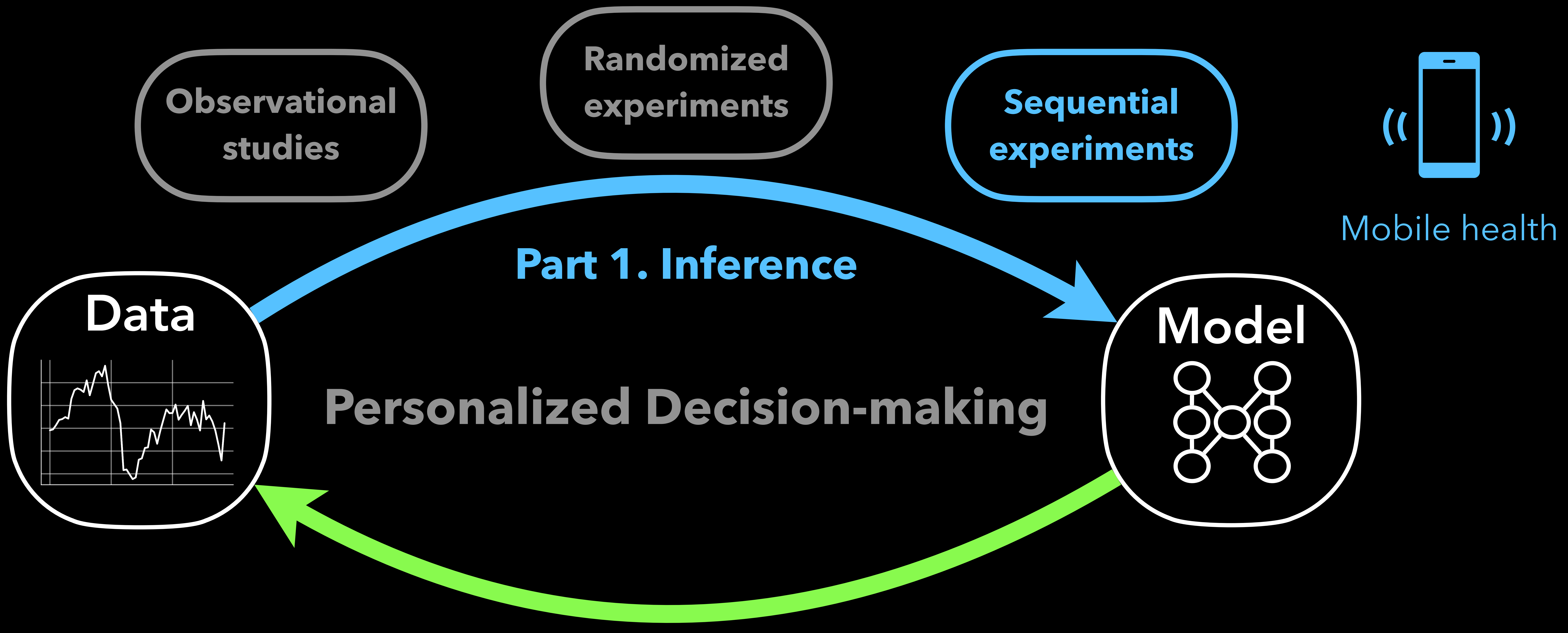
1. Use **real data** to infer decision's effect



1. Use **real data** to infer decision's effect

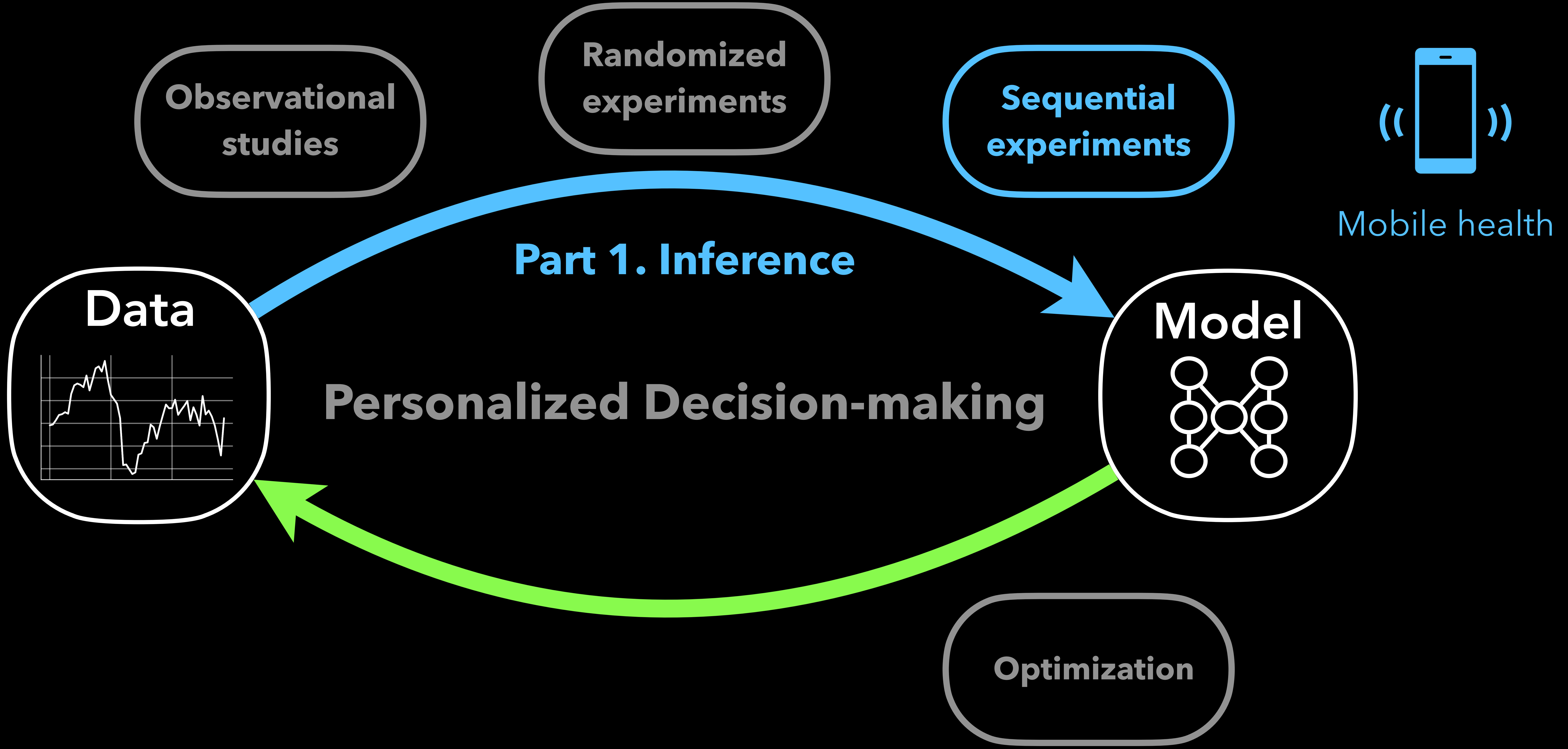


1. Use **real data** to infer decision's effect

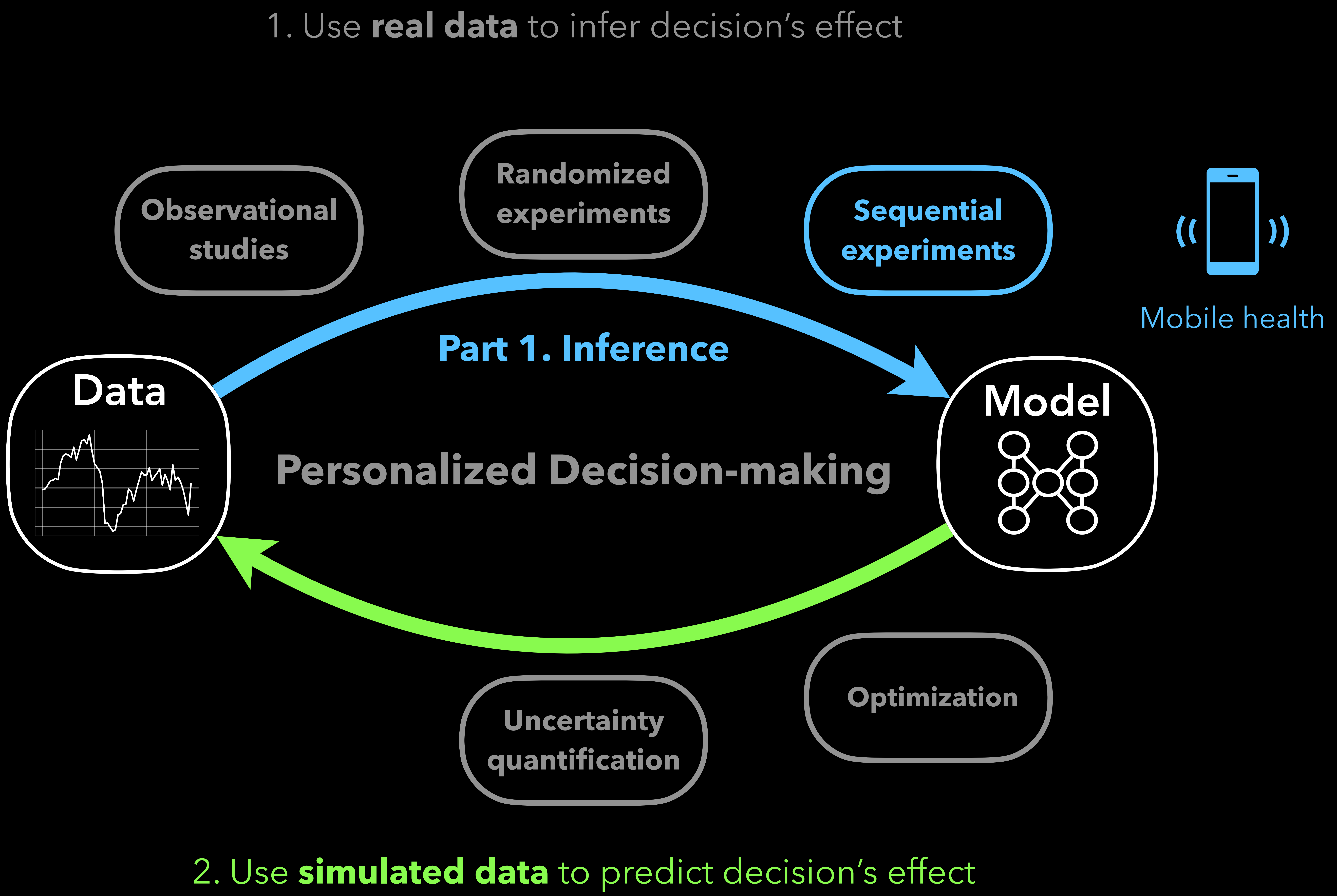


2. Use **simulated data** to predict decision's effect

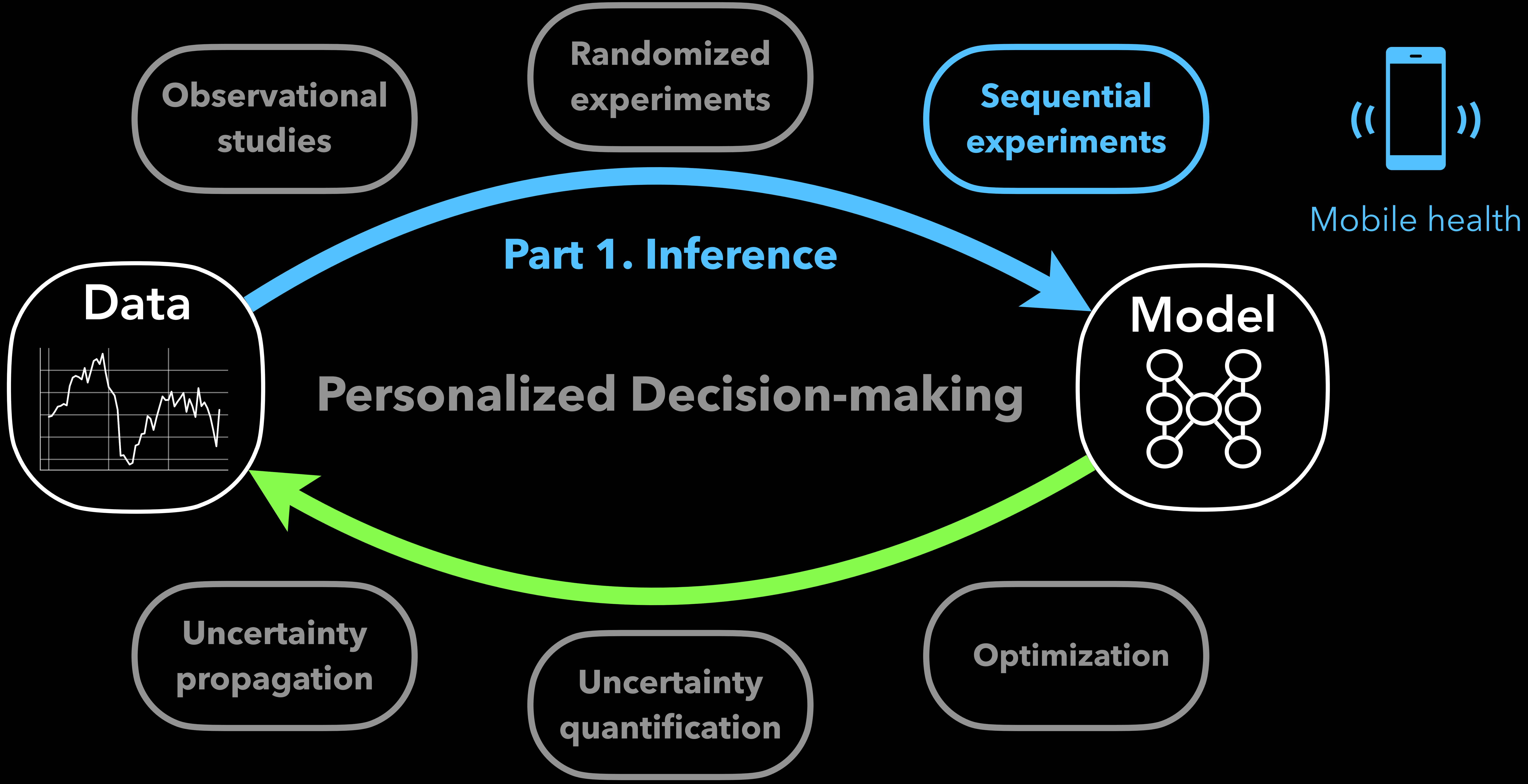
1. Use **real data** to infer decision's effect



2. Use **simulated data** to predict decision's effect


1. Use **real data** to infer decision's effect
Observational studies
Randomized experiments
Sequential experiments
Mobile health
Part 1. Inference
Data
Personalized Decision-making
Model
Uncertainty quantification
Optimization
2. Use **simulated data** to predict decision's effect

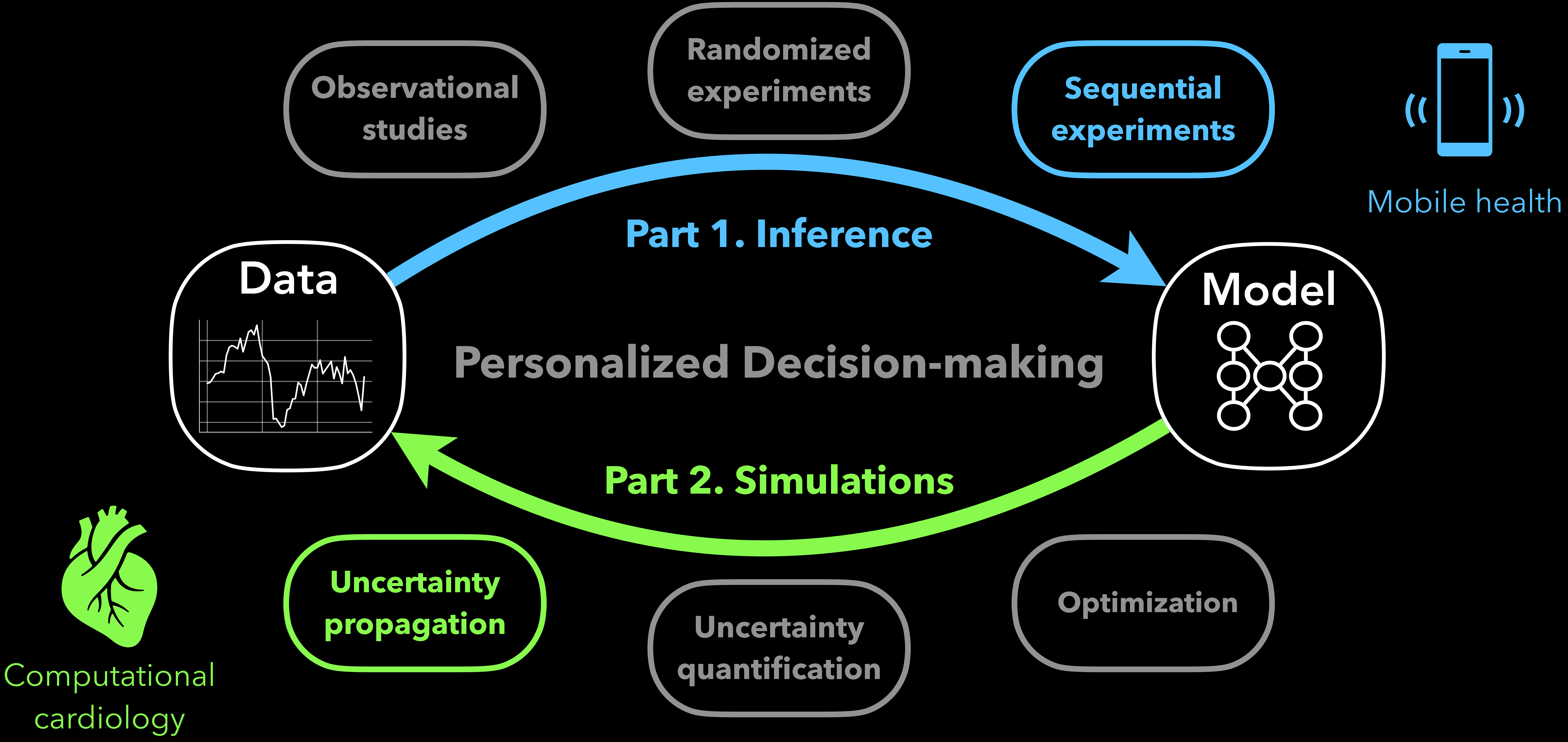
1. Use **real data** to infer decision's effect



Personalized Decision-making

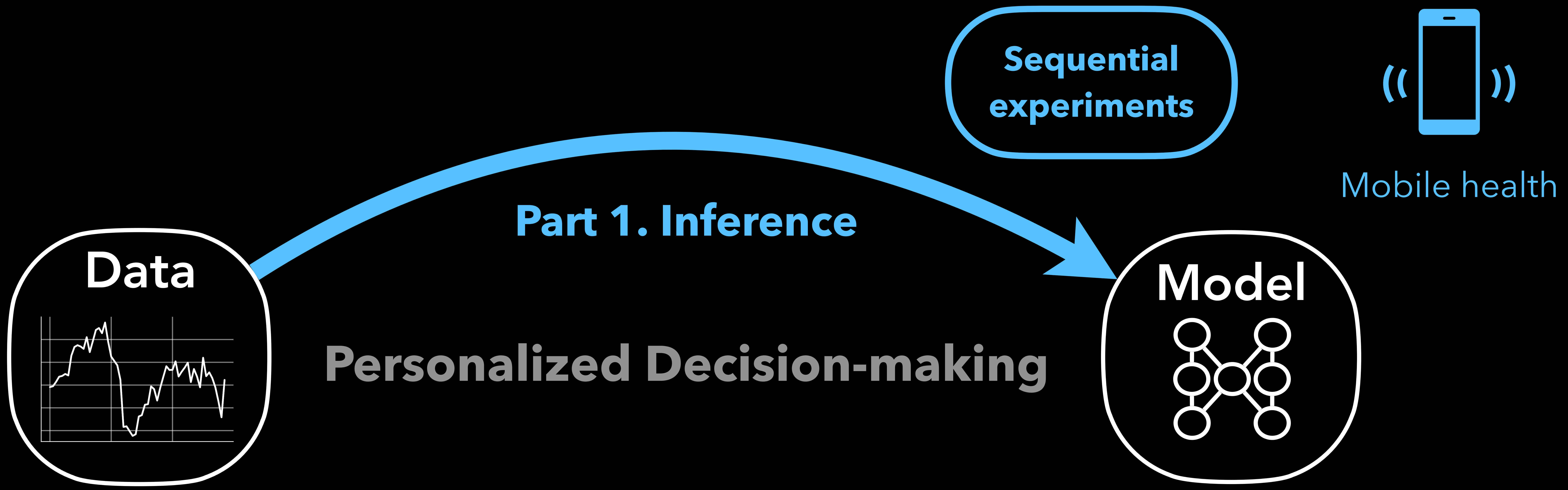
2. Use **simulated data** to predict decision's effect

1. Use **real data** to infer decision's effect

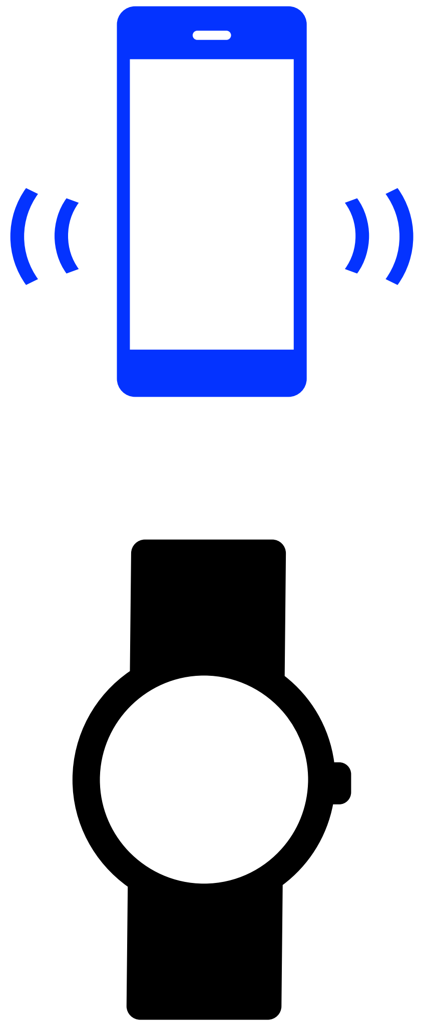


2. Use **simulated data** to predict decision's effect

1. Use **real data** to infer decision's effect

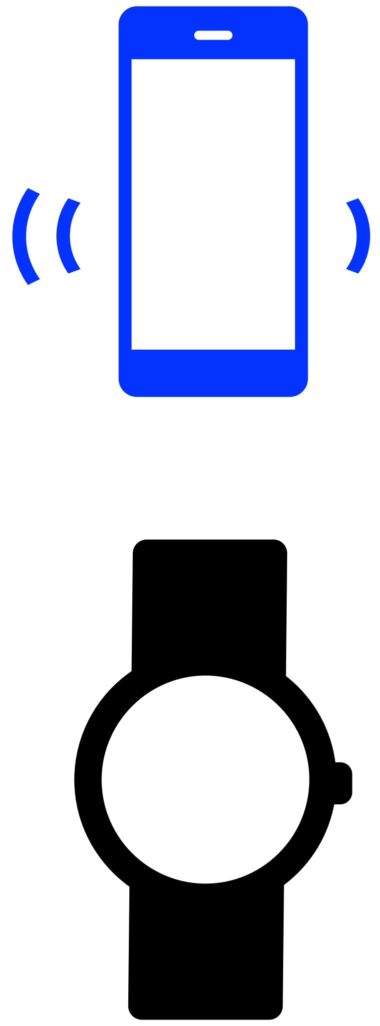


Building AI agents for personalized treatments



Building AI agents for personalized treatments

How to assign personalized digital treatments to help you?



Building AI agents for personalized treatments

How to assign personalized digital treatments to help you?

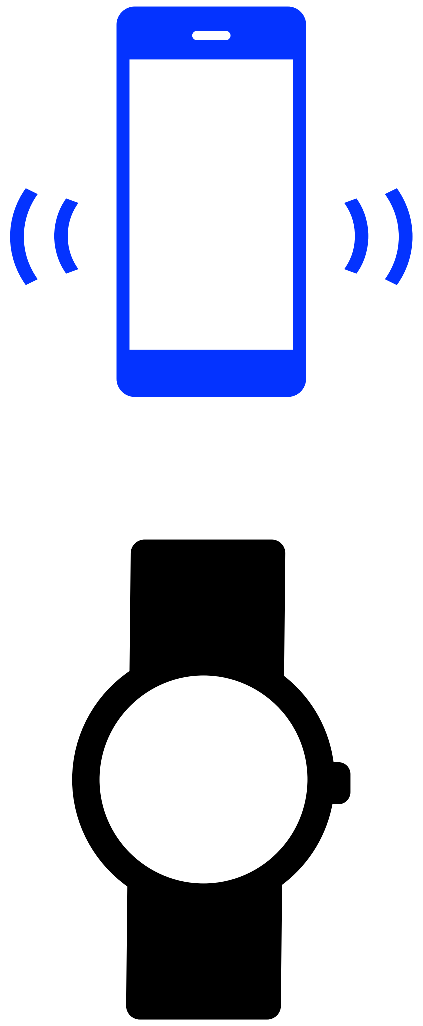
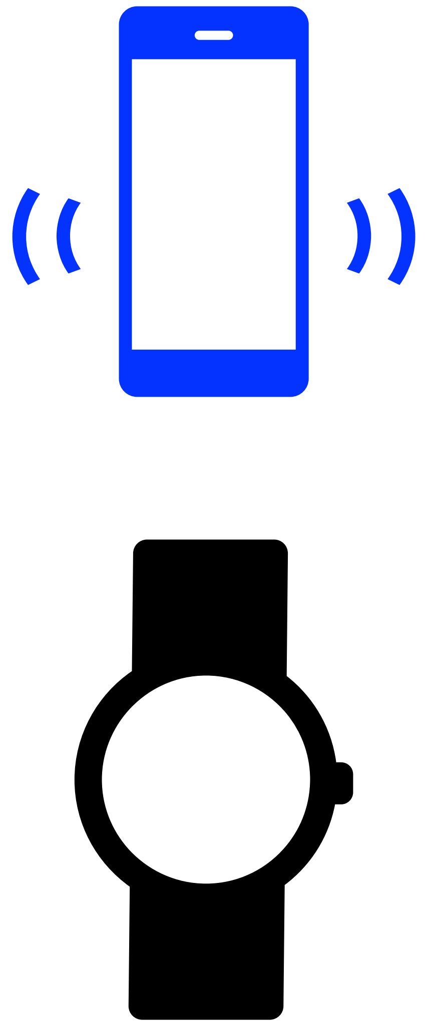


image credits: va.gov

Building AI agents for personalized treatments

How to assign personalized digital treatments to help you?



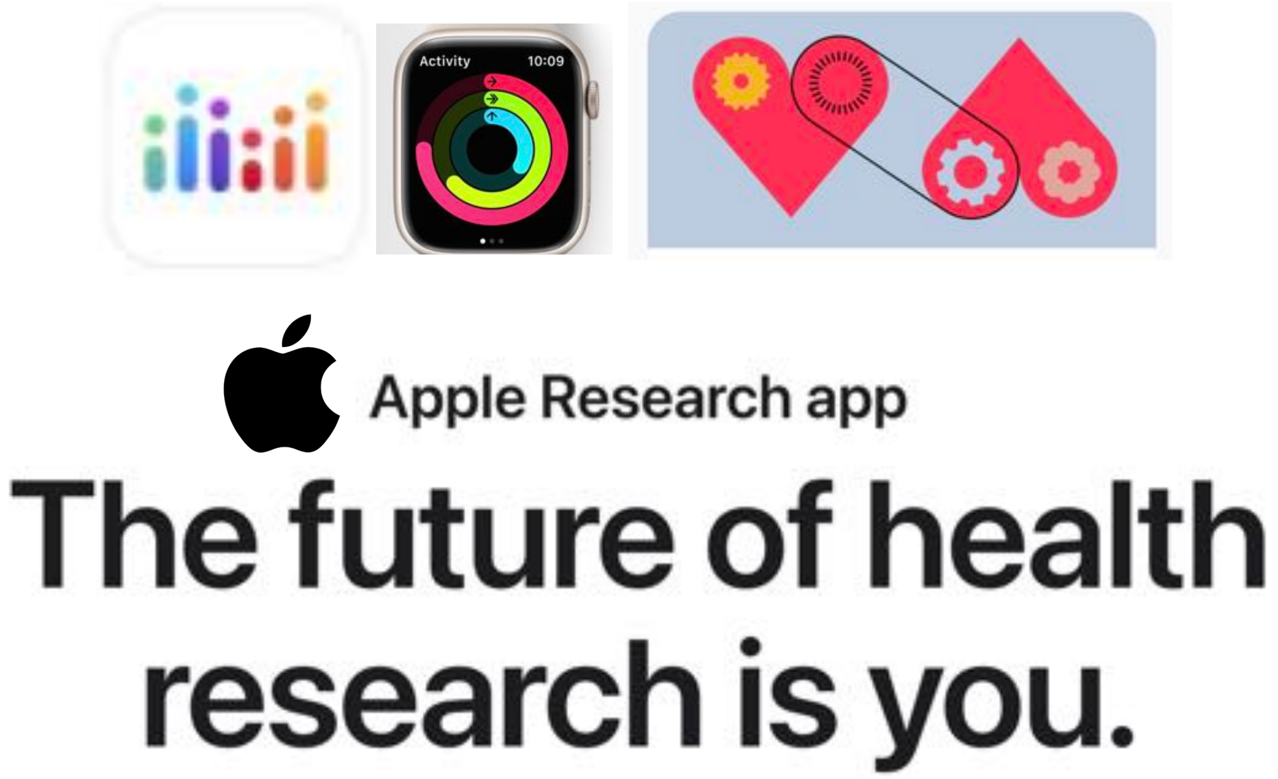
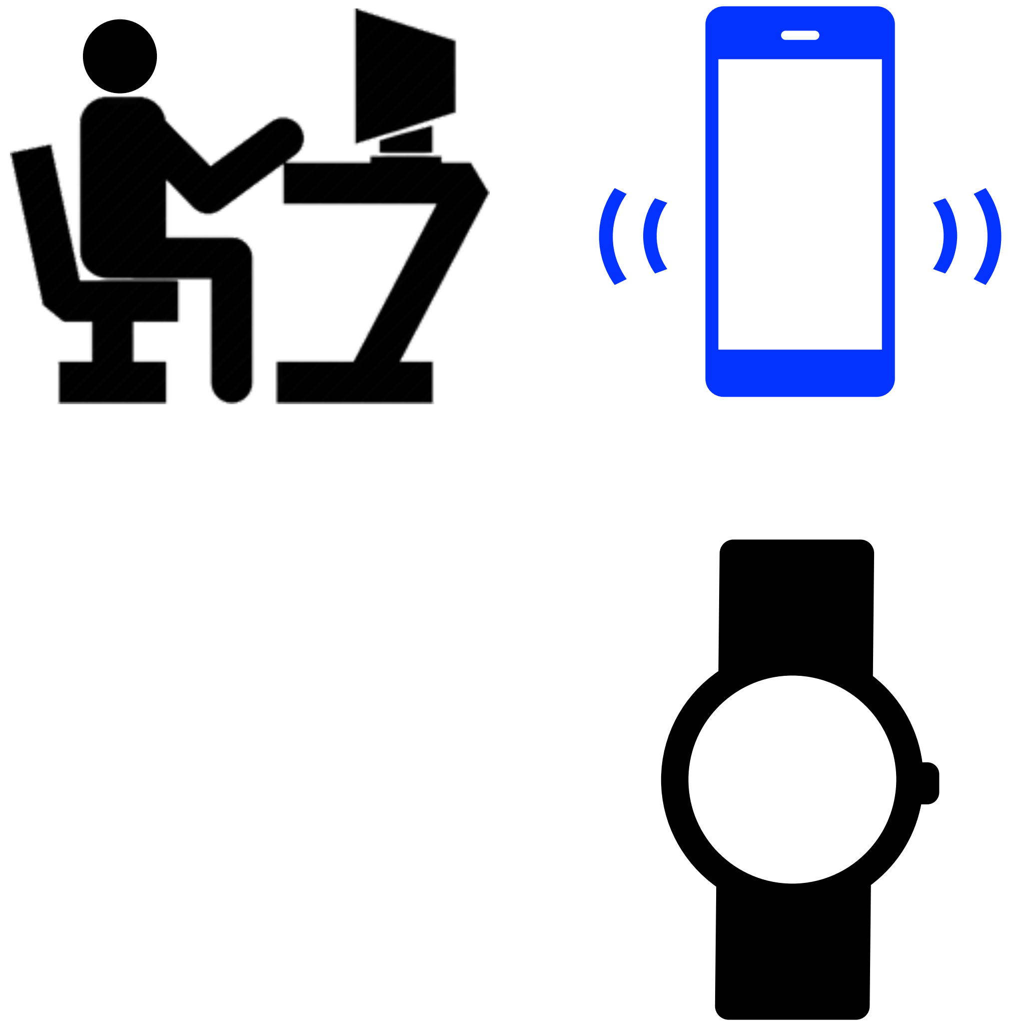


image credits
va.gov
apple.com

Building AI agents for personalized treatments

How to assign personalized digital treatments to help you?



Mobile health study:
Personalized HeartSteps [Liao+ '20]

- **Goal**: Promote physical activity via mobile app
- **Population:** 91 hypertension patients, 90 days

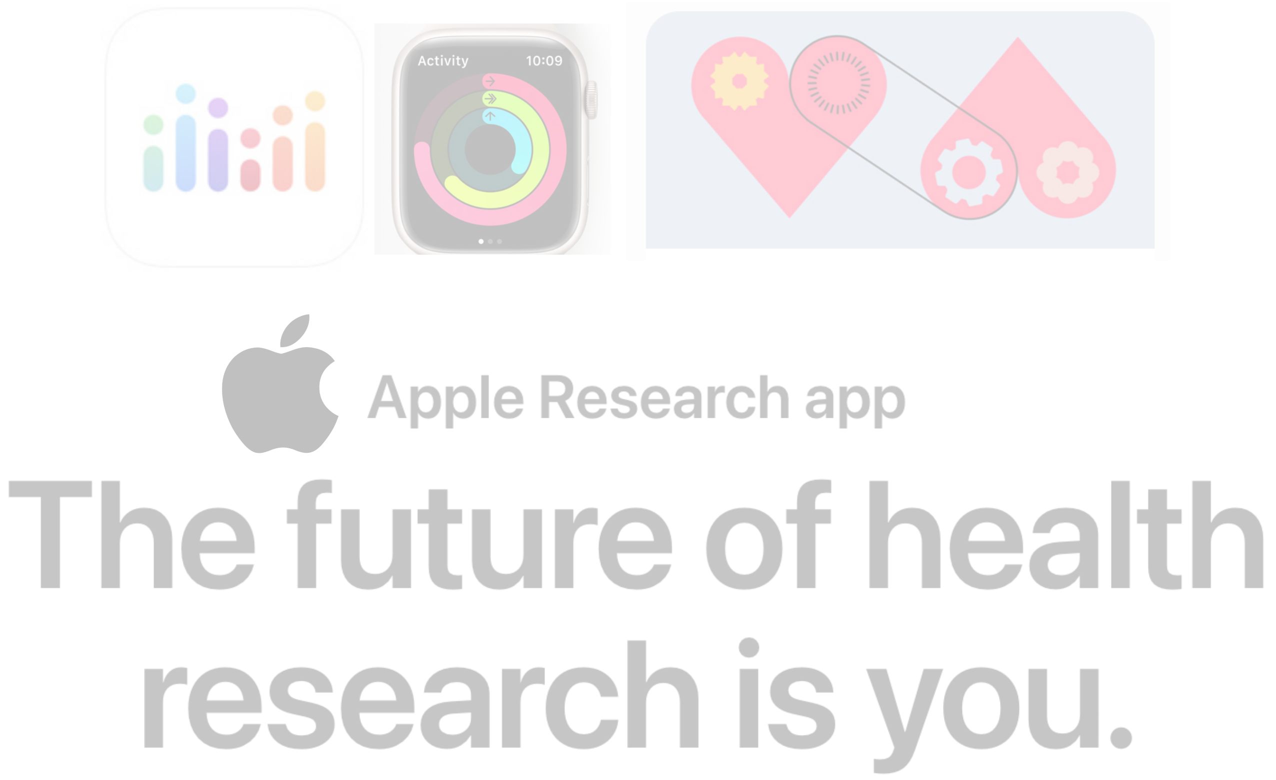
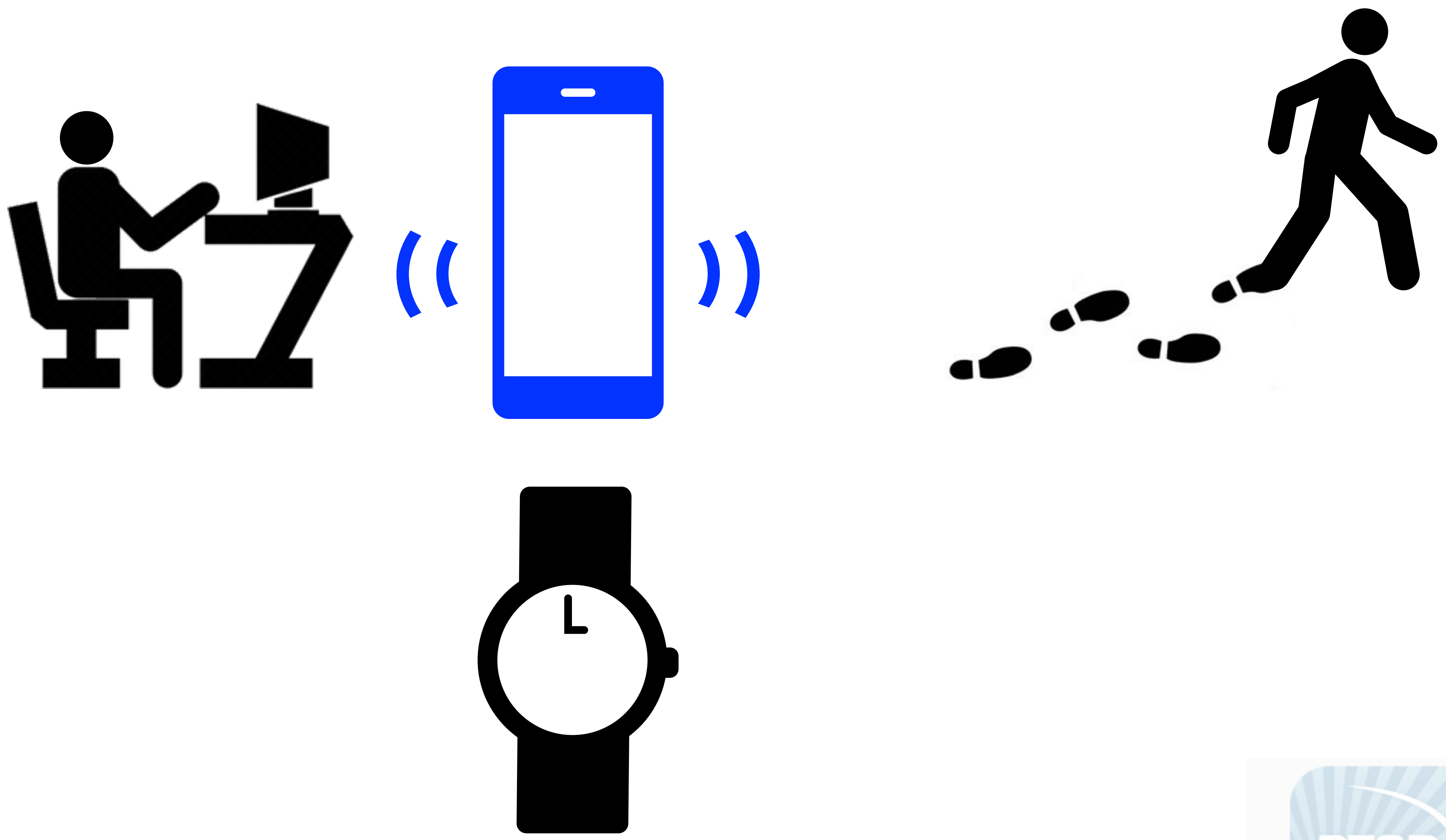


image credits:
va.gov
apple.com

Building AI agents for personalized treatments

How to assign personalized digital treatments to help you?



Mobile health study:
Personalized HeartSteps [Liao+ '20]

- **Goal**: Promote physical activity via mobile app
- **Population:** 91 hypertension patients, 90 days
- **Treatment:** Mobile notifications 5 times/day assigned by a bandit algorithm

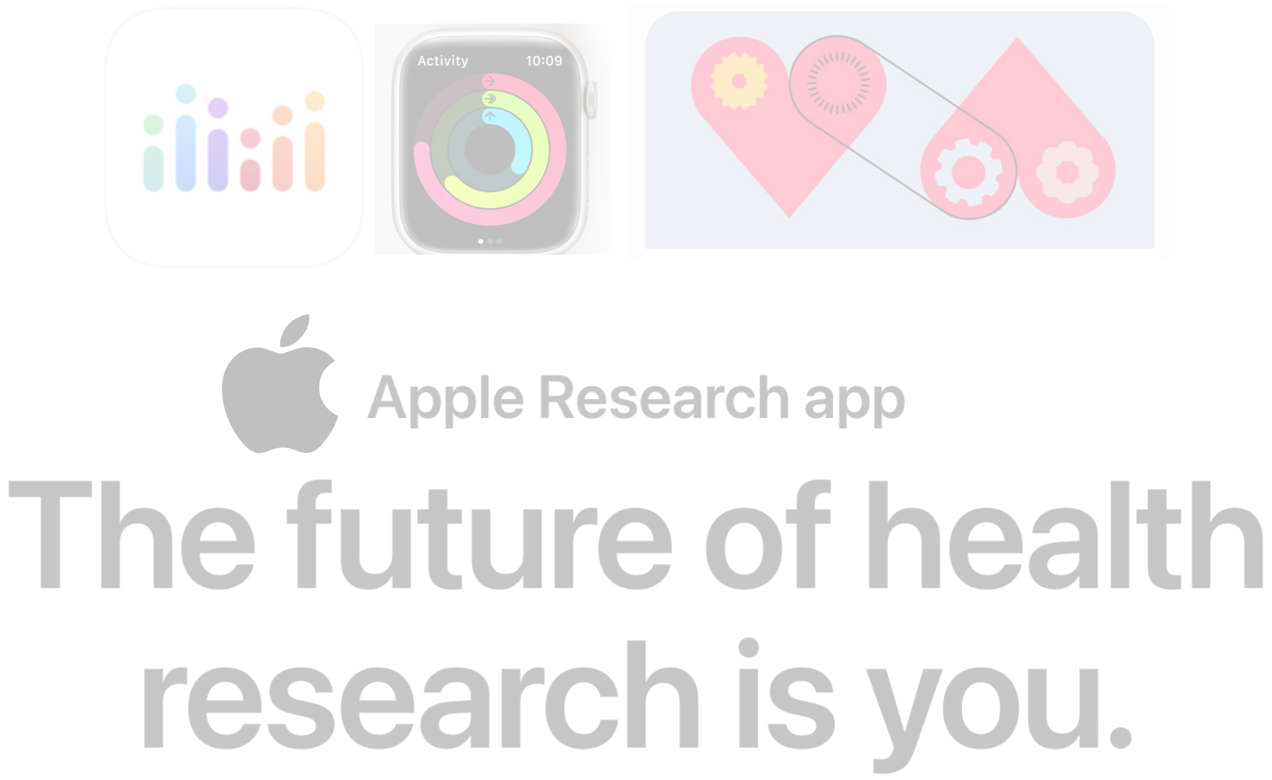
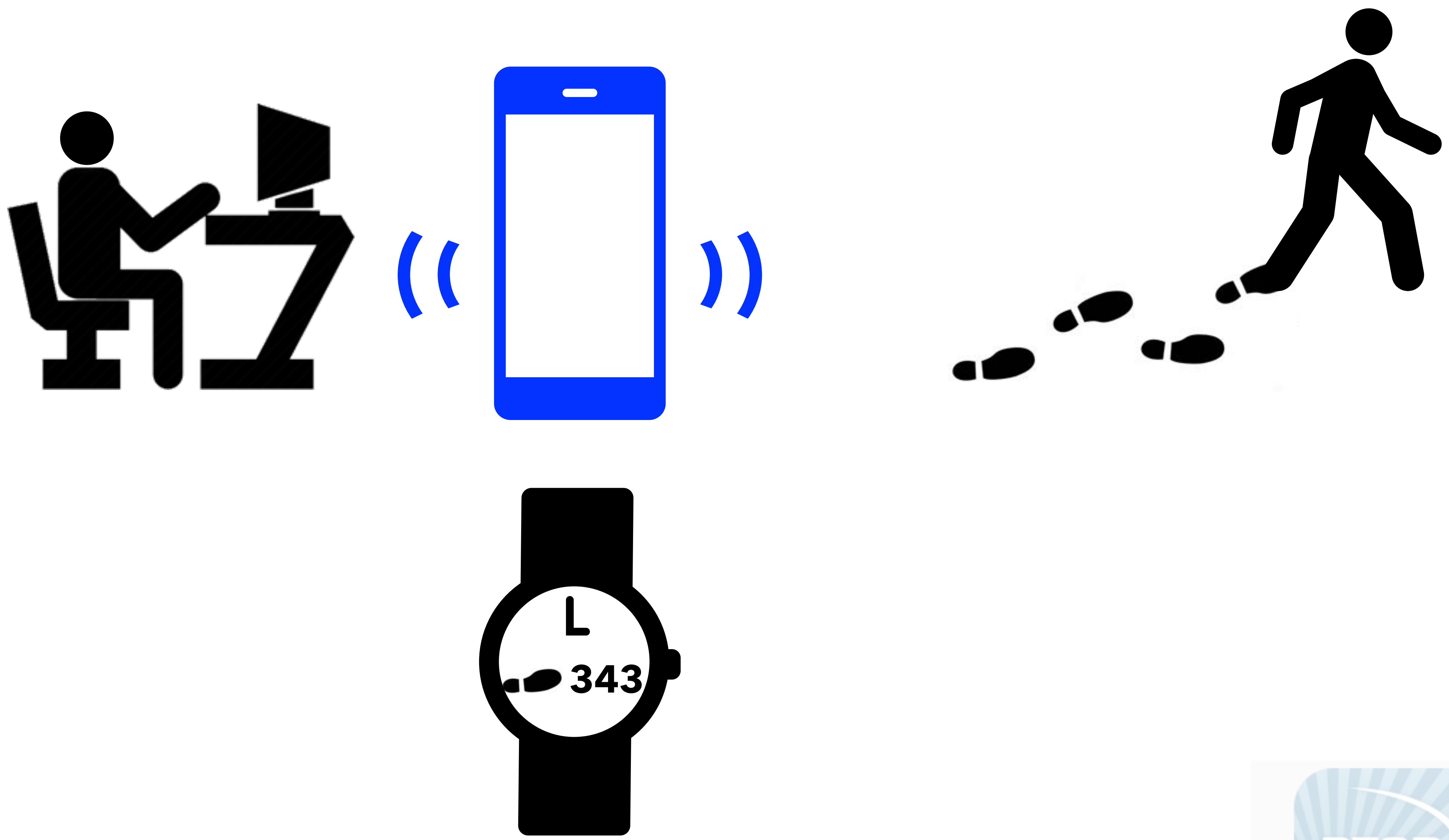


image credits:
va.gov
apple.com

Building AI agents for personalized treatments

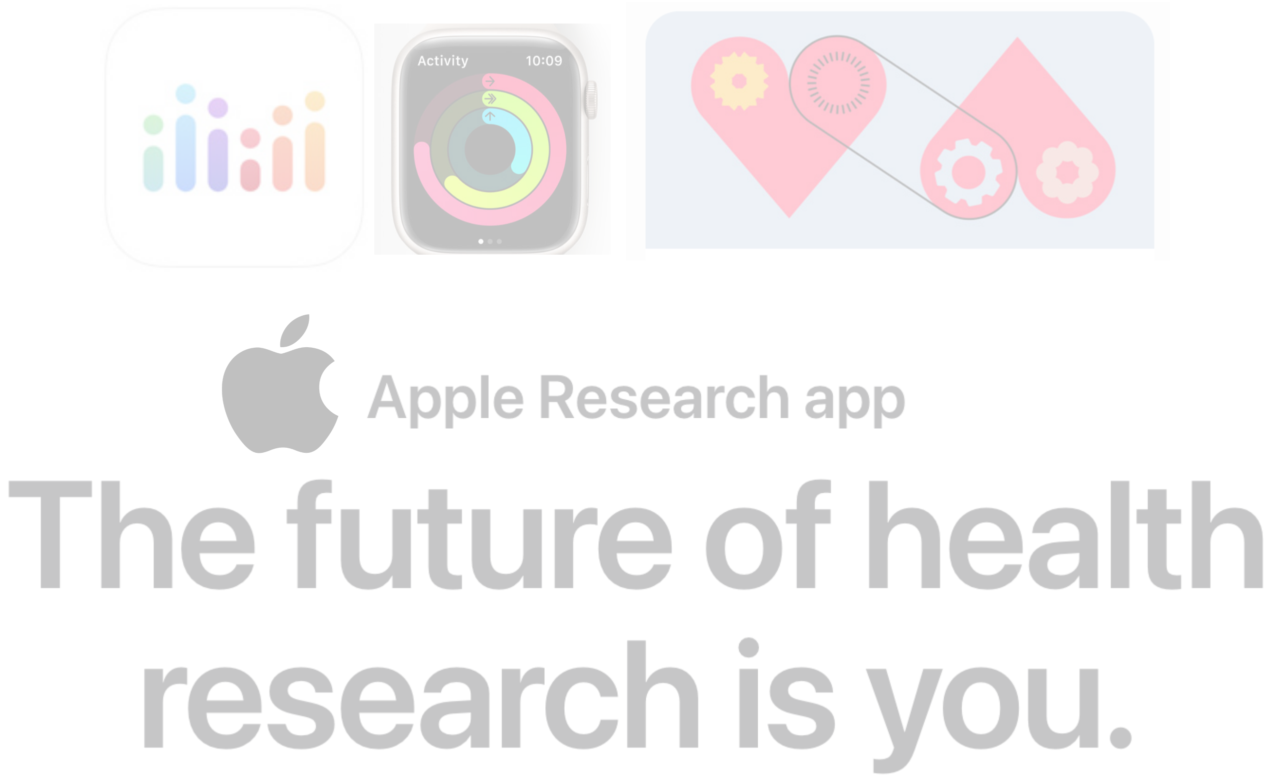
How to assign personalized digital treatments to help you?



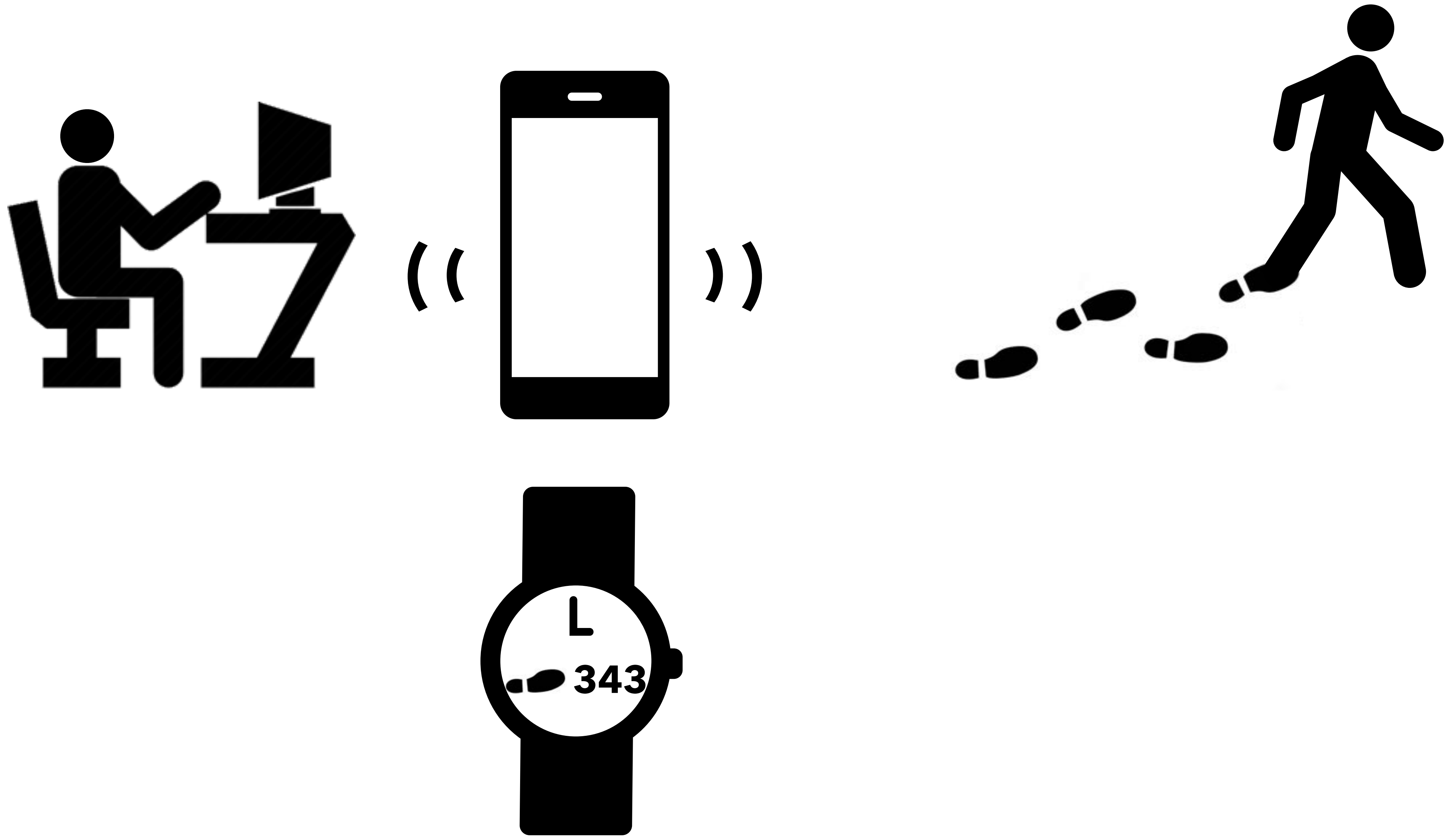
Mobile health study:
Personalized HeartSteps [Liao+ '20]

- **Goal**: Promote physical activity via mobile app
- **Population:** 91 hypertension patients, 90 days
- **Treatment:** Mobile notifications 5 times/day assigned by a bandit algorithm
- **Outcome**: 30-min step count after decision time

image credits: va.gov apple.com

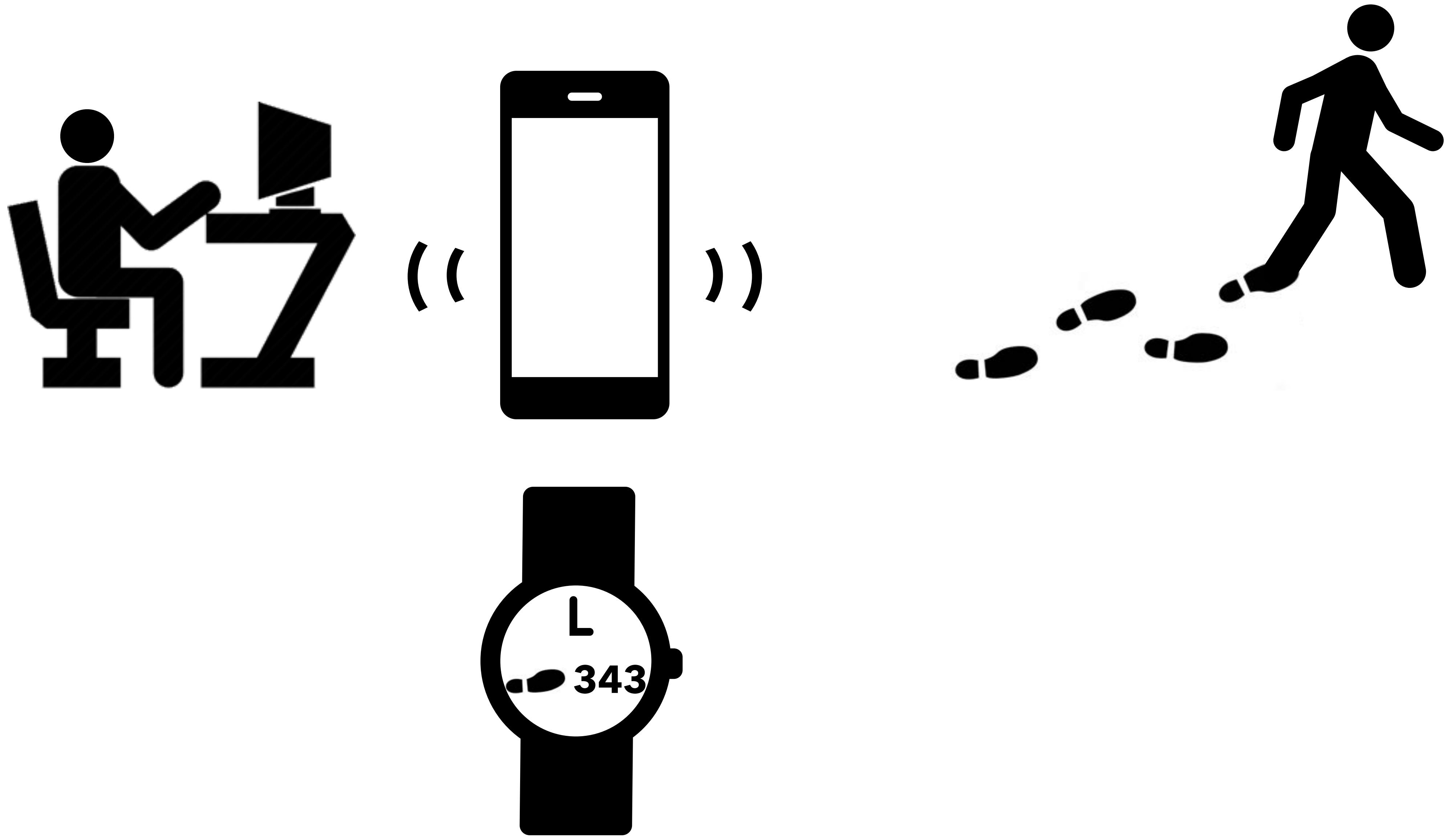


How to assign personalized digital treatments to help you?



After-study personalized inference questions

How to assign personalized digital treatments to help you?

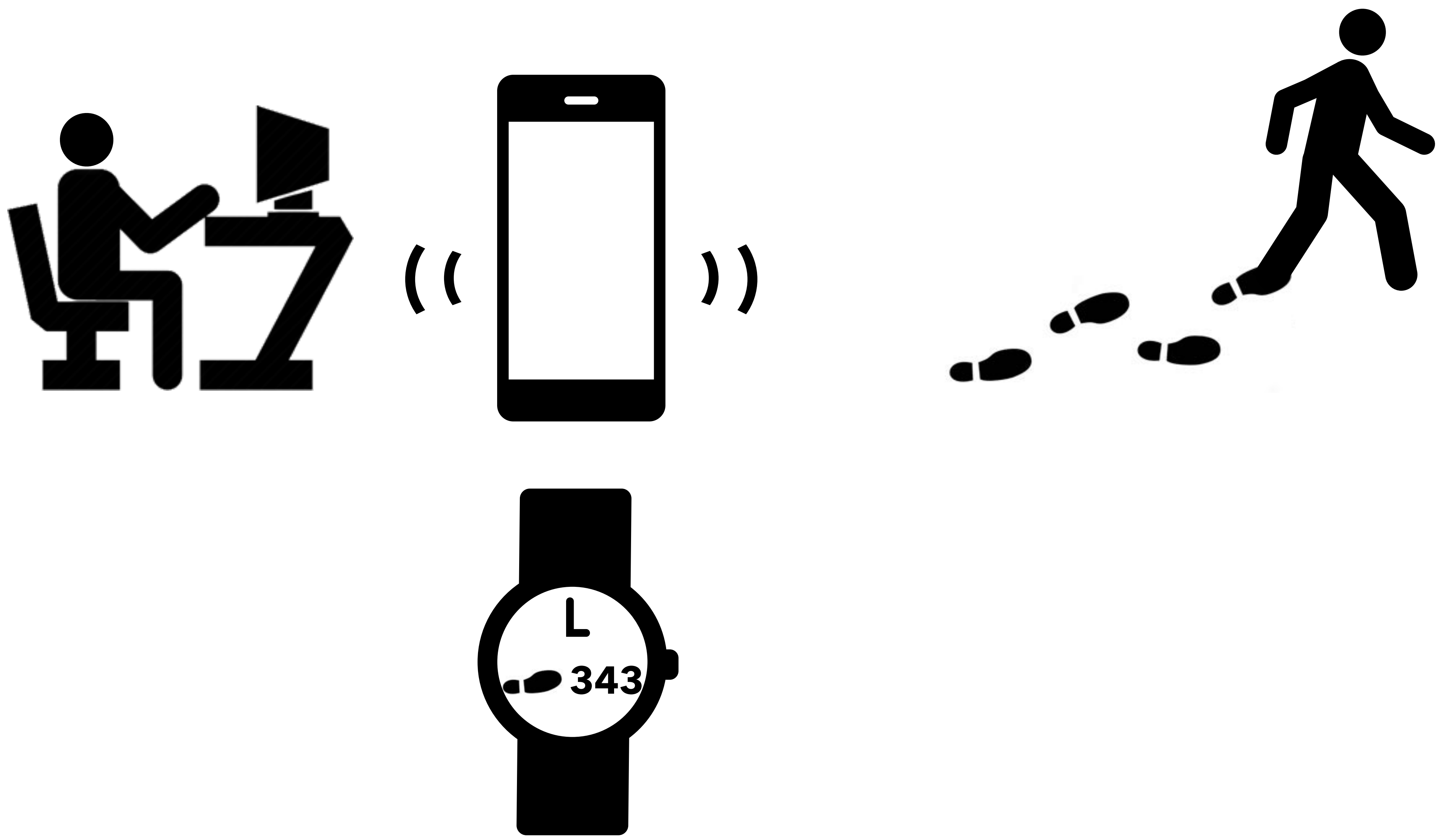


Did the app increase physical activity for a given user?

After-study personalized inference questions

How to assign personalized digital treatments to help you?

❓Did the app increase physical activity for a <u>given user</u>?



Was sending the notification effective?

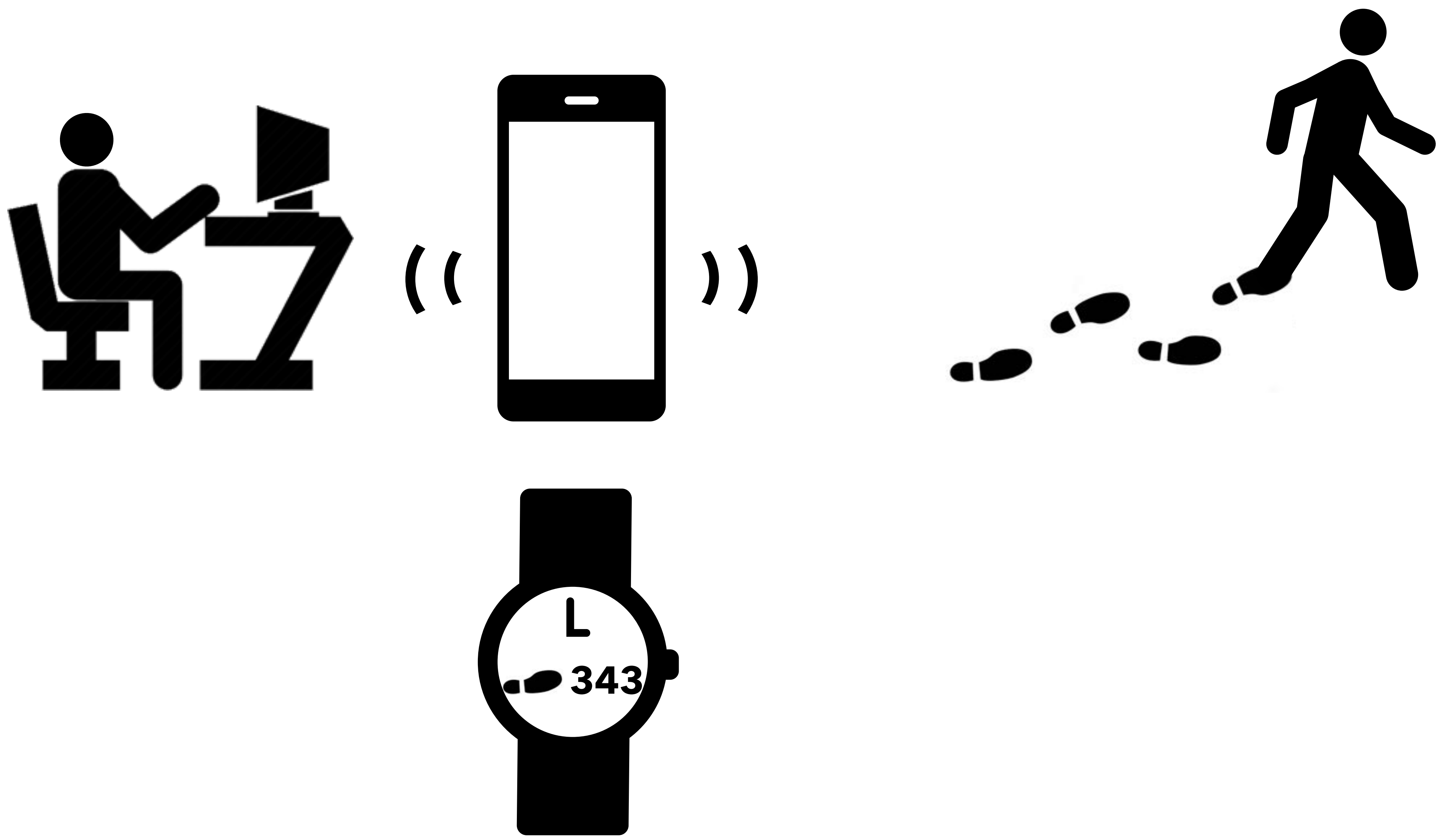
After-study personalized inference questions

How to assign personalized digital treatments to help you?

Did the app increase physical activity for a <u>given user</u>?

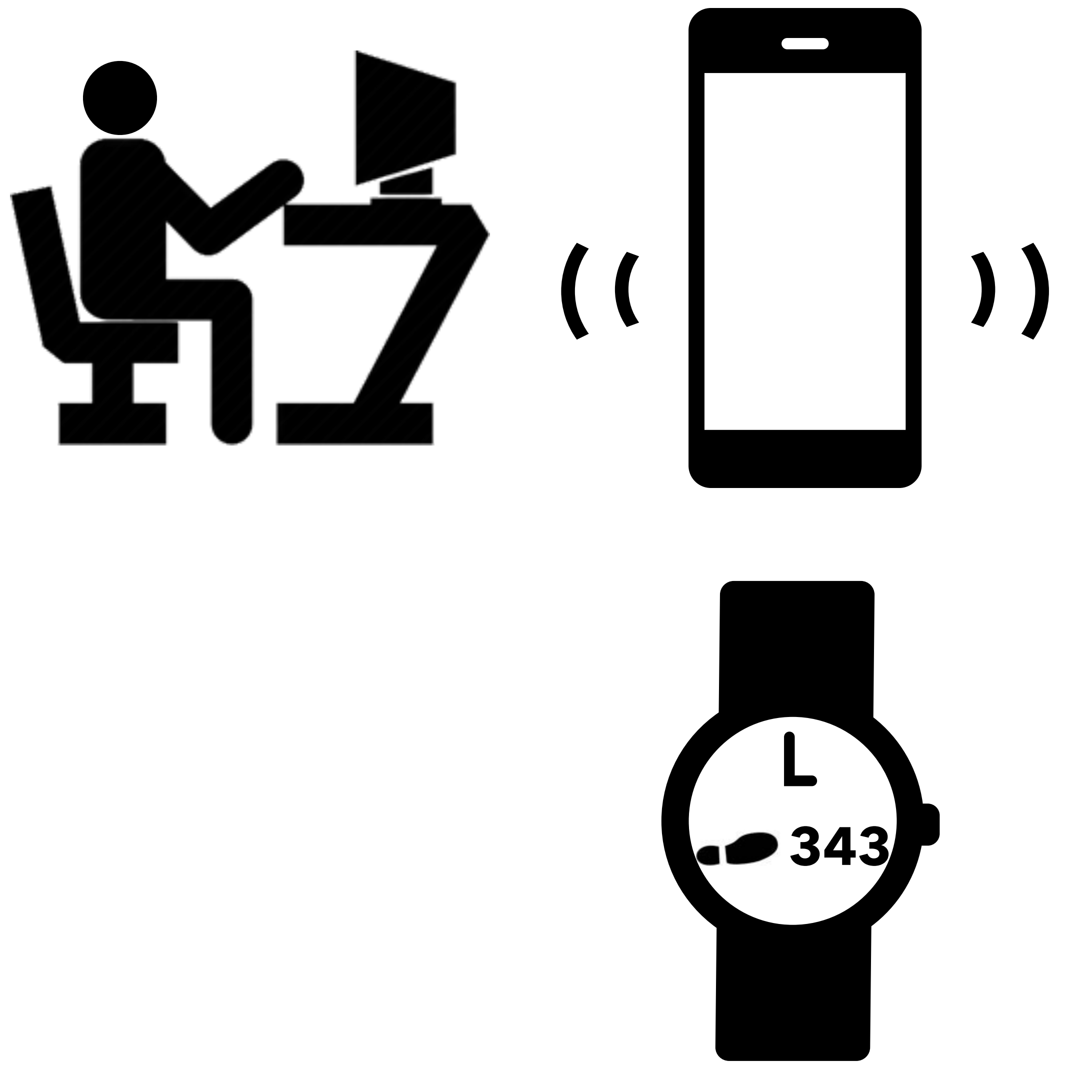
Was sending the notification effective?

Was the bandit algorithm effective?



After-study personalized inference questions

How to assign personalized digital treatments to help you?



Did the app increase physical activity for a given user?

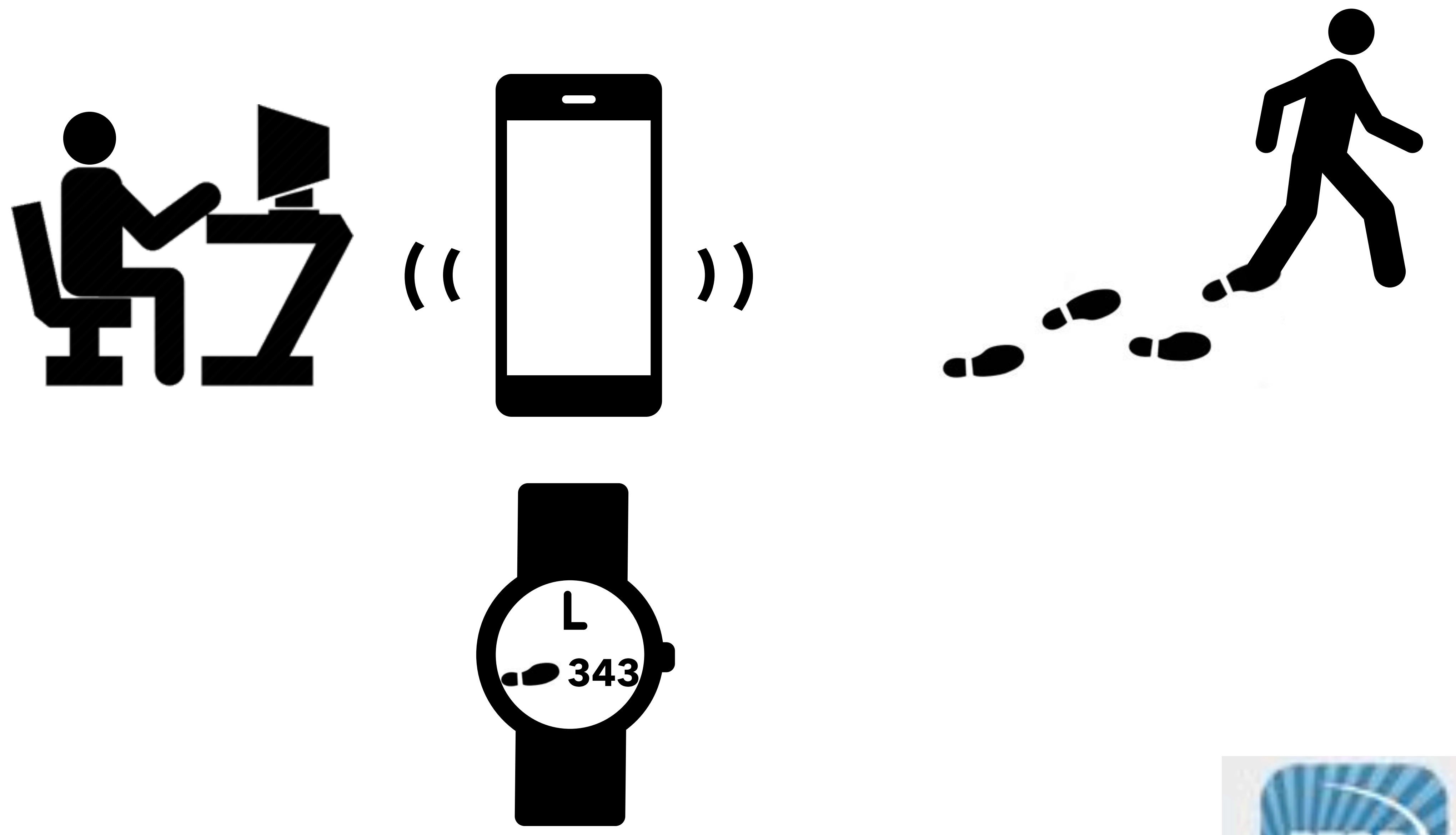
Was sending the notification effective?

Was the bandit algorithm effective?

Challenges: Lack of mechanistic models, adaptively collected data, expensive data collection

After-study personalized inference questions

How to assign personalized digital treatments to help you?



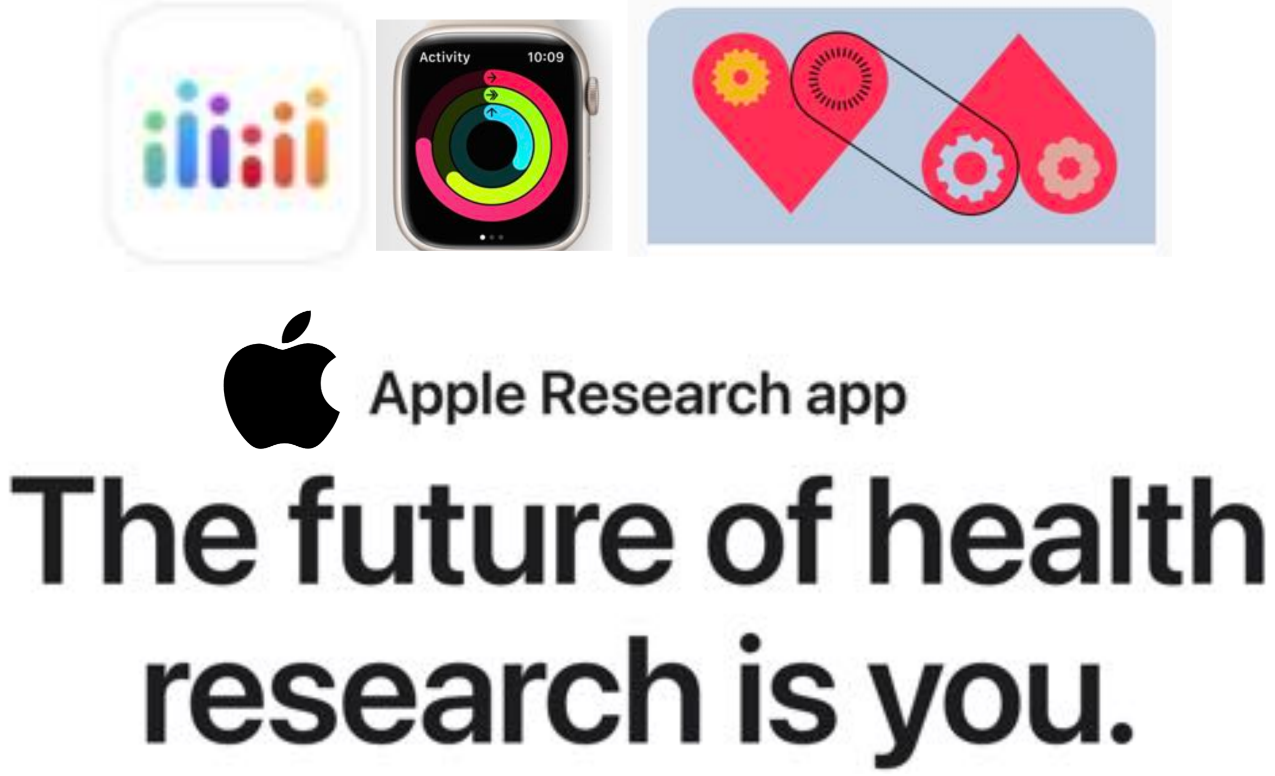
Did the app increase physical activity for a given user?

Was sending the notification effective?

Was the bandit algorithm effective?

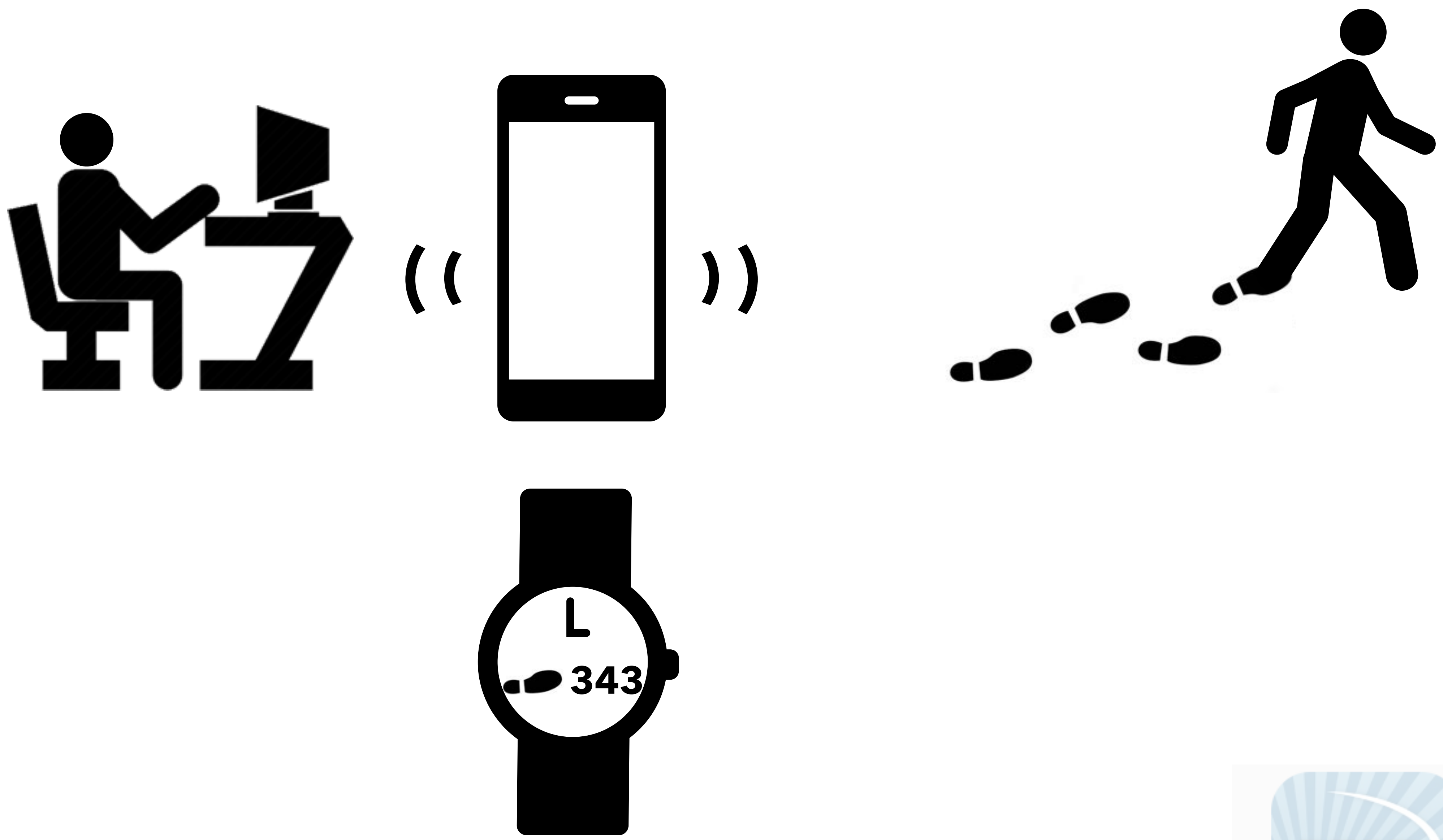
Challenges: Lack of mechanistic models, adaptively collected data, expensive data collection

image credits
va.gov
apple.com



Part 1 overview: Sample-efficient personalized inference in sequential experiments

How to assign personalized digital treatments to help you?



Did the app increase physical activity for a given user?

This talk

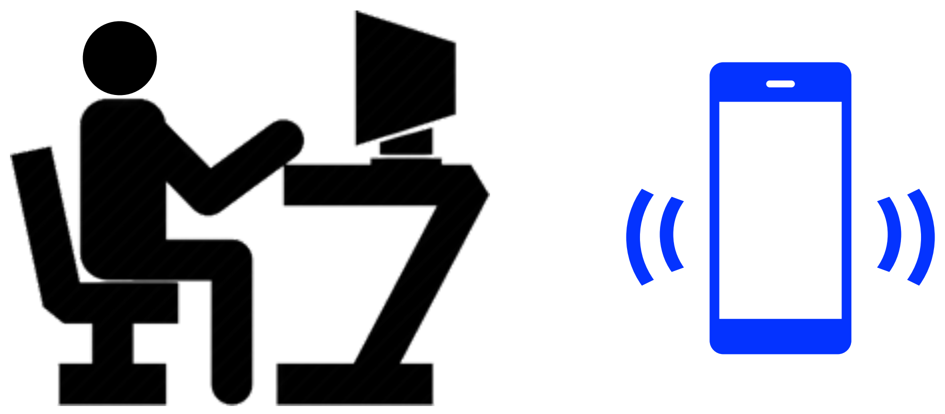
Was sending the notification effective?

Was the bandit algorithm effective?

Challenges: Lack of mechanistic models, adaptively collected data, expensive data collection

image credits
va.gov
apple.com

Problem set-up

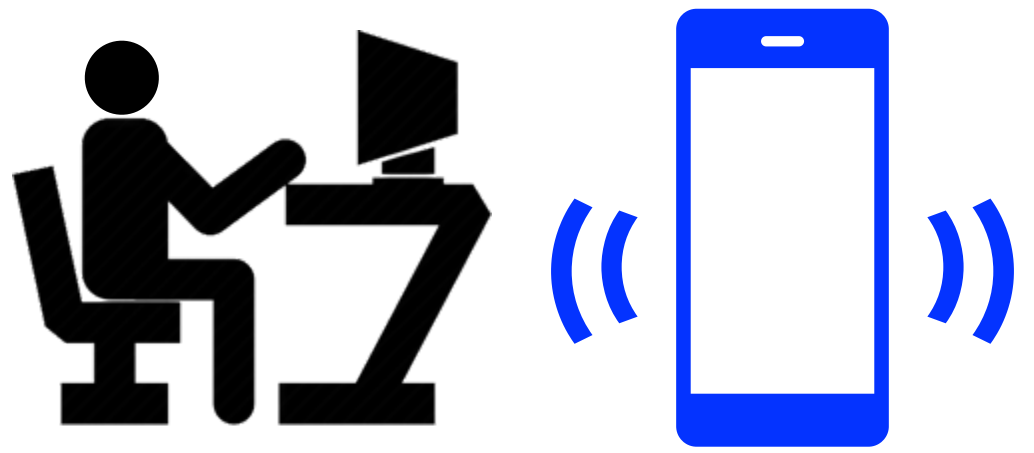


Problem set-up

For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using $\text{policy}_{i,t}$

Problem set-up



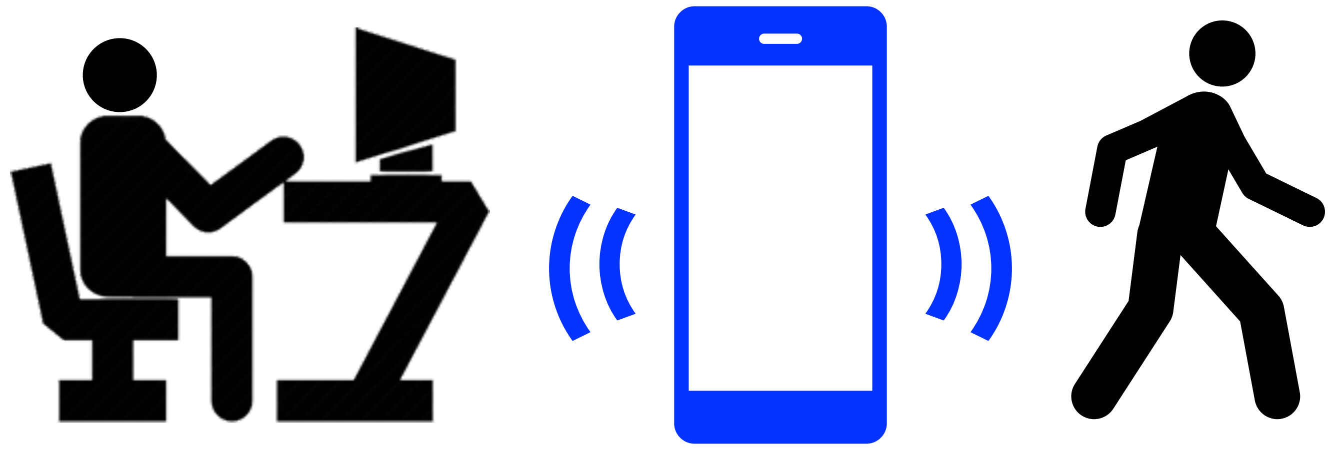
For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using policy$_{i,t}$

e.g., ε-greedy, Thompson sampling, softmax, multiplicative weights, pooled variants,…

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning

Problem set-up



For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using policy$_{i,t}$

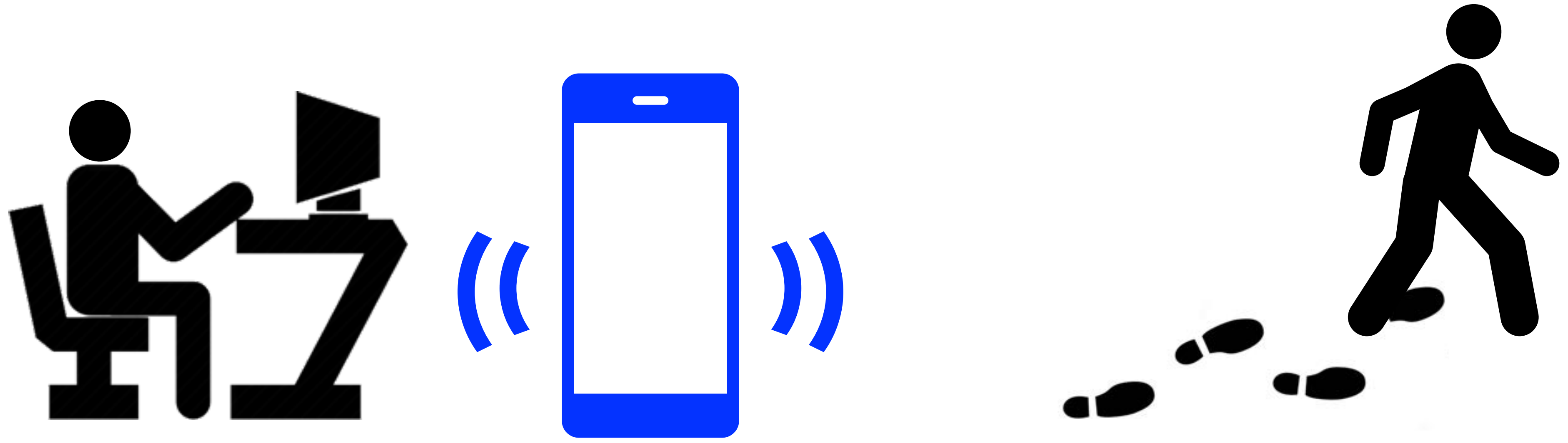
$\theta_{i,t}^{(a)}$: mean potential outcome/counterfactual for treatment $a \in \{0,1\}$ (mean step counts)

[Neyman-Rubin framework

e.g., ε-greedy, Thompson sampling, softmax, multiplicative weights, pooled variants,…

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning

Problem set-up



For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using policy$_{i,t}$

$\theta_{i,t}^{(a)}$: mean potential outcome/counterfactual for treatment $a \in \{0,1\}$ (mean step counts)

outcome observed:

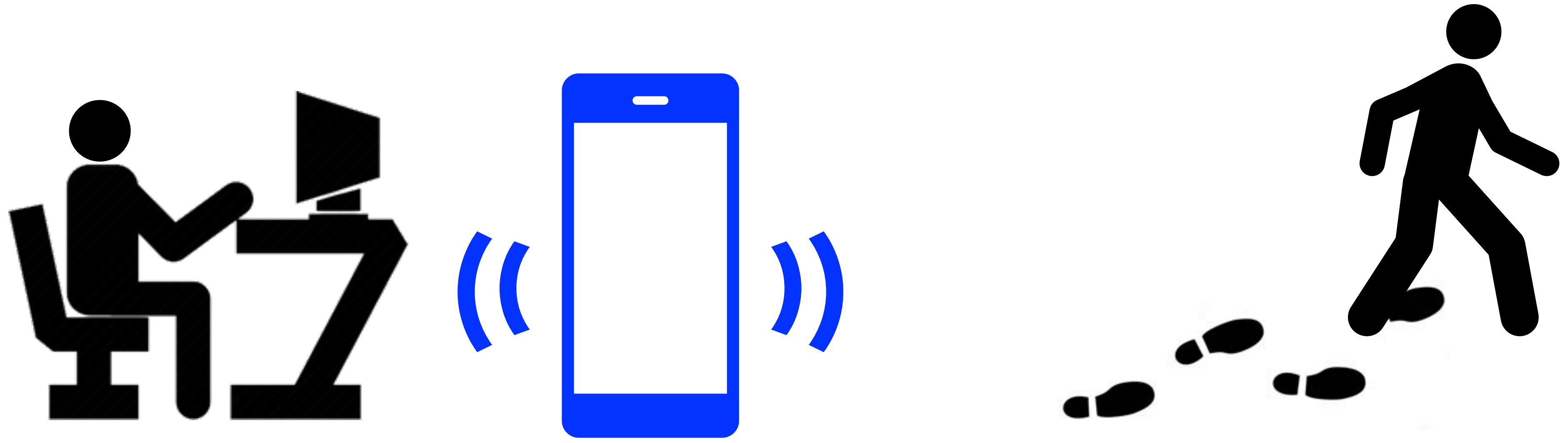
$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

[Neyman-Rubin framework + SUTVA]

e.g., ε-greedy, Thompson sampling, softmax, multiplicative weights, pooled variants,…

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning

Problem set-up



For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using policy$_{i,t}$

$\theta_{i,t}^{(a)}$: mean potential outcome/counterfactual for treatment $a \in \{0,1\}$ (mean step counts)

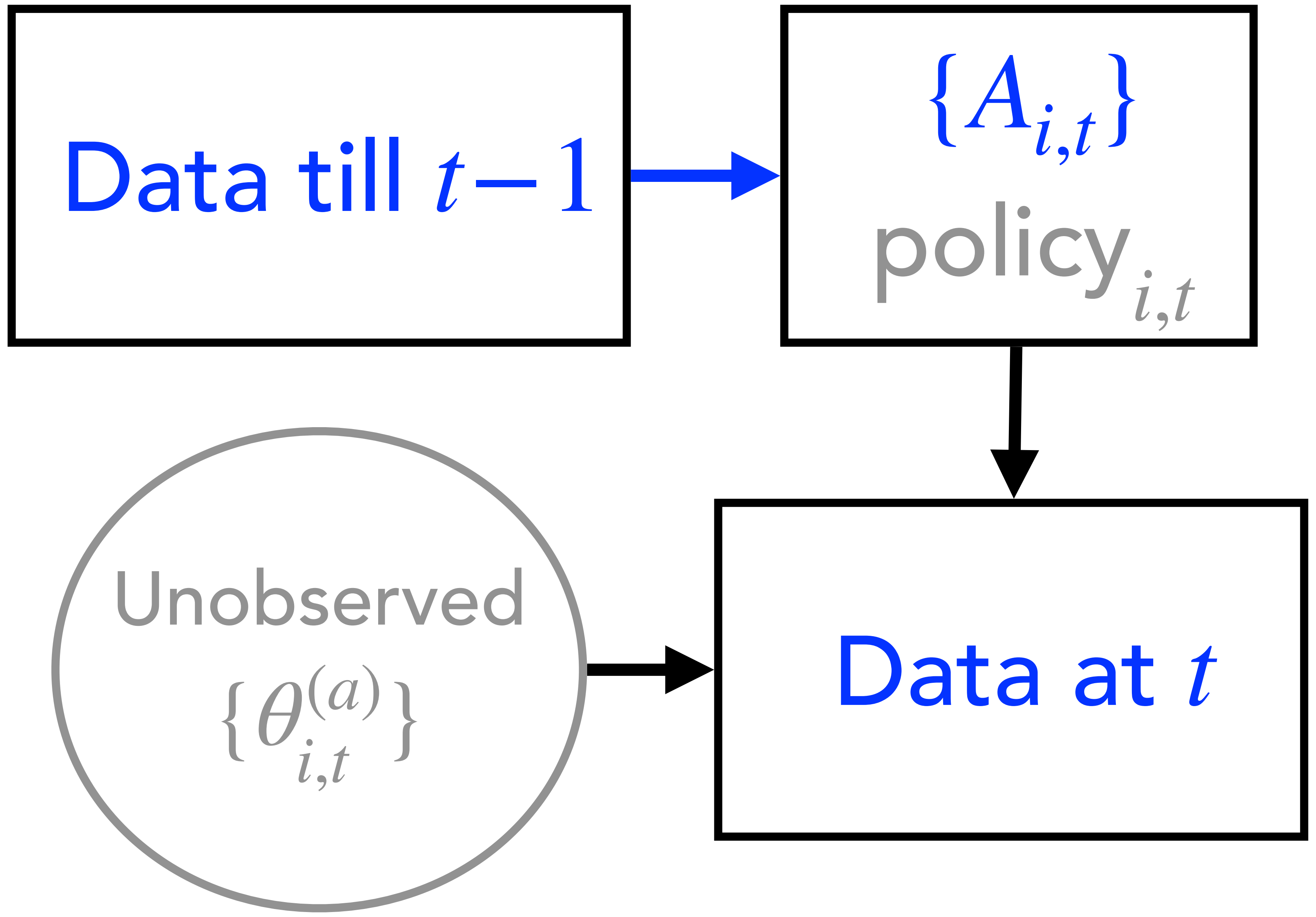
outcome observed:

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

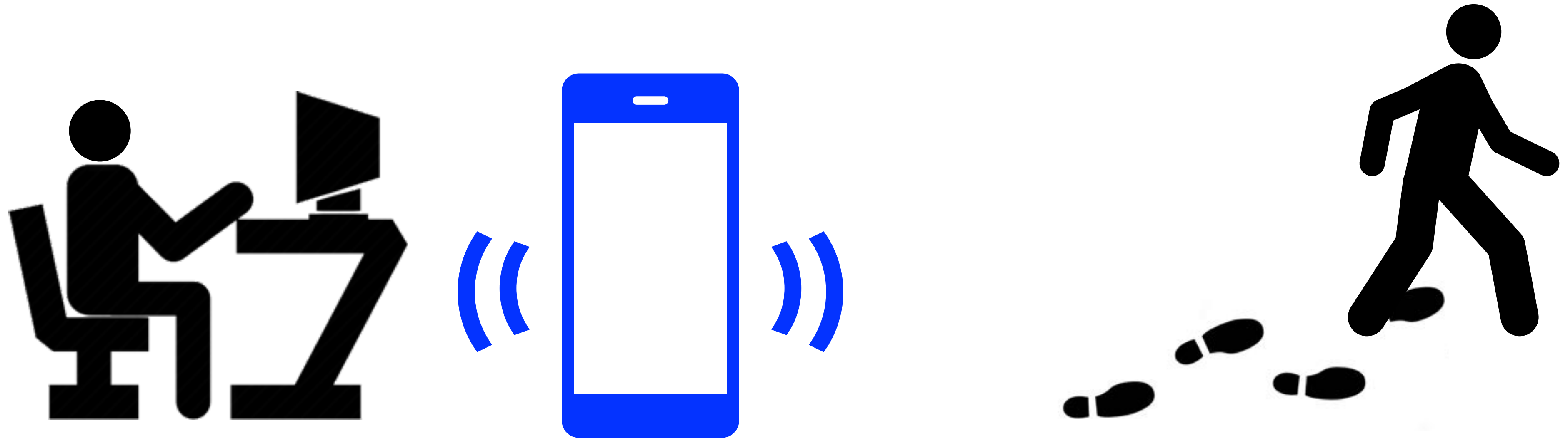
[Neyman-Rubin framework + SUTVA]

e.g., ε-greedy, Thompson sampling, softmax, multiplicative weights, pooled variants,…

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning



Problem set-up



For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using $\text{policy}_{i,t}$

$\theta_{i,t}^{(a)}$: mean potential outcome/counterfactual for treatment $a \in \{0,1\}$ (mean step counts)

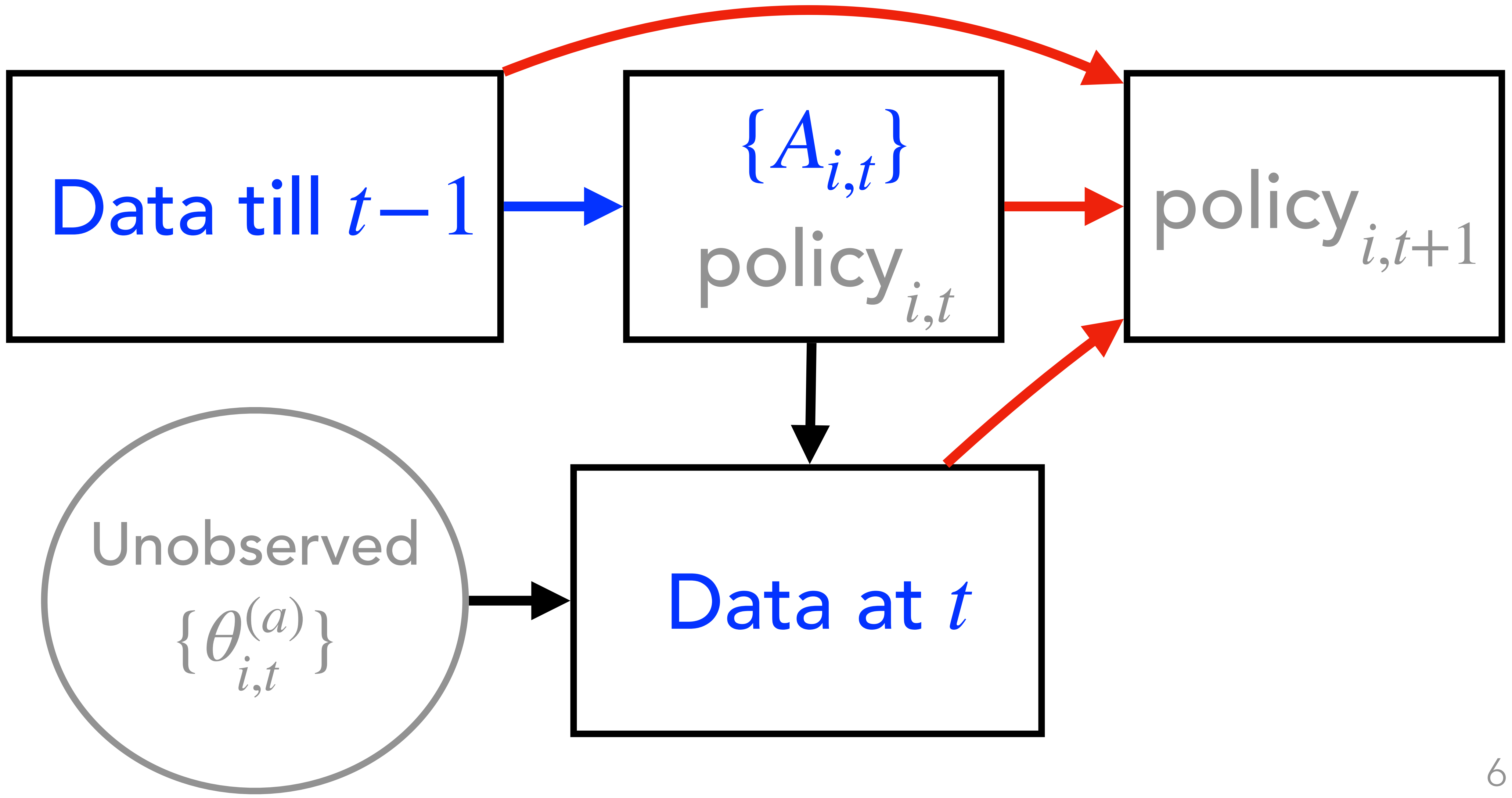
outcome observed:

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

[Neyman-Rubin framework + SUTVA]

e.g., ε-greedy, Thompson sampling, softmax, multiplicative weights, pooled variants,…

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning



Problem set-up and goal

For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using policy$_{i,t}$

$\theta_{i,t}^{(a)}$: mean potential outcome/counterfactual for treatment $a \in \{0,1\}$ (mean step counts)

outcome observed:

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

[Neyman-Rubin framework + SUTVA]

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning

Problem set-up and **goal**

For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using policy$_{i,t}$

$\theta_{i,t}^{(a)}$: mean potential outcome/counterfactual for treatment $a \in \{0,1\}$ (mean step counts)

outcome observed:

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

[Neyman-Rubin framework + SUTVA]

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, **all** N users & T times

Problem set-up and **goal**

For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using policy$_{i,t}$

$\theta_{i,t}^{(a)}$: mean potential outcome/counterfactual for treatment $a \in \{0,1\}$ (mean step counts)

outcome observed:

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

[Neyman-Rubin framework + SUTVA]

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$,

all N users & T times

- Enable generic after-study analyses and assist next study design

Problem set-up and **goal**

For user $i \in [N]$ at time $t \in [T]$

$A_{i,t}$: treatment $\in \{0,1\}$ (send a notification or not) assigned using policy$_{i,t}$

$\theta_{i,t}^{(a)}$: mean potential outcome/counterfactual for treatment $a \in \{0,1\}$ (mean step counts)

outcome observed:

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

[Neyman-Rubin framework + SUTVA]

Sequentially adaptive policy that **can pool** observed data **across users** to speed up learning

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, **all** N users & T times

- Enable generic after-study analyses and assist next study design
- E.g., how effective was the notification for user i at time t ($\theta_{i,t}^{(1)} - \theta_{i,t}^{(0)}$)?

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

➡ More unknowns than (noisy) observations

An impossible task without structural assumptions…

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available

An impossible task without structural assumptions…

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to

An impossible task without structural assumptions…

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- More unknowns than (noisy) observations
- No parametric model available
- Intricate dependencies due to
  - Heterogeneity across users and time

An impossible task without structural assumptions…

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - Heterogeneity across users and time
  - Sequentially adaptive policy

An impossible task without structural assumptions…

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - ◉ Heterogeneity across users and time
  - ◉ Sequentially adaptive policy
  - ◉ Pooling for policy design

An impossible task without structural assumptions…

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - ◉ Heterogeneity across users and time
  - ◉ Sequentially adaptive policy
  - ◉ Pooling for policy design

An impossible task without structural assumptions…

Hope:

★ N iid users

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - Heterogeneity across users and time
  - Sequentially adaptive policy
  - Pooling for policy design

An impossible task without structural assumptions…

Hope:

- ★ N iid users
- ★ T (dependent) observations per user

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - ◉ Heterogeneity across users and time
  - ◉ Sequentially adaptive policy
  - ◉ Pooling for policy design

An impossible task without structural assumptions…

Hope:

- ★ N iid users
- ★ T (dependent) observations per user
- ★ If users are not all too different & multiple observations can help find similarities

A possible task with some structural assumptions…

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - ◉ Heterogeneity across users and time
  - ◉ Sequentially adaptive policy
  - ◉ Pooling for policy design

An impossible task without structural assumptions...

Prior work:

- **Average treatment effect**

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - ◉ Heterogeneity across users and time
  - ◉ Sequentially adaptive policy
  - ◉ Pooling for policy design

An impossible task without structural assumptions…

Prior work:

- **Average treatment effect**
  - IID users & deterministic rules/policies

[… Robins '94, '97, '00, '08, Murphy '03, '05, Hernan+ '06, Moodie+ '07,

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - ◉ Heterogeneity across users and time
  - ◉ Sequentially adaptive policy
  - ◉ Pooling for policy design

An impossible task without structural assumptions…

Prior work:

- **Average treatment effect**
  - IID users & deterministic rules/policies
  - IID users at each time with stochastic policies

[… Robins '94, '97, '00, '08, Murphy '03, '05, Hernan+ '06, Moodie+ '07,
… Deshpande+ '18, Hadad+ '21, Bibaut+ '21, Khamaru+ '21, Zhang+ '21,

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - ◉ Heterogeneity across users and time
  - ◉ Sequentially adaptive policy
  - ◉ Pooling for policy design

An impossible task without structural assumptions…

Prior work:

- **Average treatment effect**
  - IID users & deterministic rules/policies
  - IID users at each time with stochastic policies
  - IID user trajectories (per user policy, no pooling)

[… Robins '94, '97, '00, '08, Murphy '03, '05, Hernan+ '06, Moodie+ '07, … Deshpande+ '18, Hadad+ '21, Bibaut+ '21, Khamaru+ '21, Zhang+ '21,

Estimate counterfactual means $\{\theta_{i,t}^{(a)}\}$ for $a \in \{0,1\}$, all N users & T times

Challenges:

- ➡ More unknowns than (noisy) observations
- ➡ No parametric model available
- ➡ Intricate dependencies due to
  - Heterogeneity across users and time
  - Sequentially adaptive policy
  - Pooling for policy design

An impossible task without structural assumptions…

Prior work:

- **Average treatment effect**
  - IID users & deterministic rules/policies
  - IID users at each time with stochastic policies
  - IID user trajectories (per user policy, no pooling)
- Observational studies (once treated forever treated; synthetic control, causal panel data)

[… Robins '94, '97, '00, '08, Murphy '03, '05, Hernan+ '06, Moodie+ '07, … Deshpande+ '18, Hadad+ '21, Bibaut+ '21, Khamaru+ '21, Zhang+ '21, … Abadie+ '03, '10, Athey+ '17, Arkhangelsky+ '18, Agarwal+ '20 …]

Structural assumption: Non-parametric factor model

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

Structural assumption: Non-parametric factor model

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

$$\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)})$$

user factor
(e.g., personal
traits)

time factor
(e.g., societal, weather
changes)

Structural assumption: Non-parametric factor model

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

$$\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)})$$

user factor (e.g., personal traits)

time factor (e.g., societal, weather changes)

No parametric assumptions on

- **unknown** non-linearity
- distributions of **unobserved** latent factors and noise

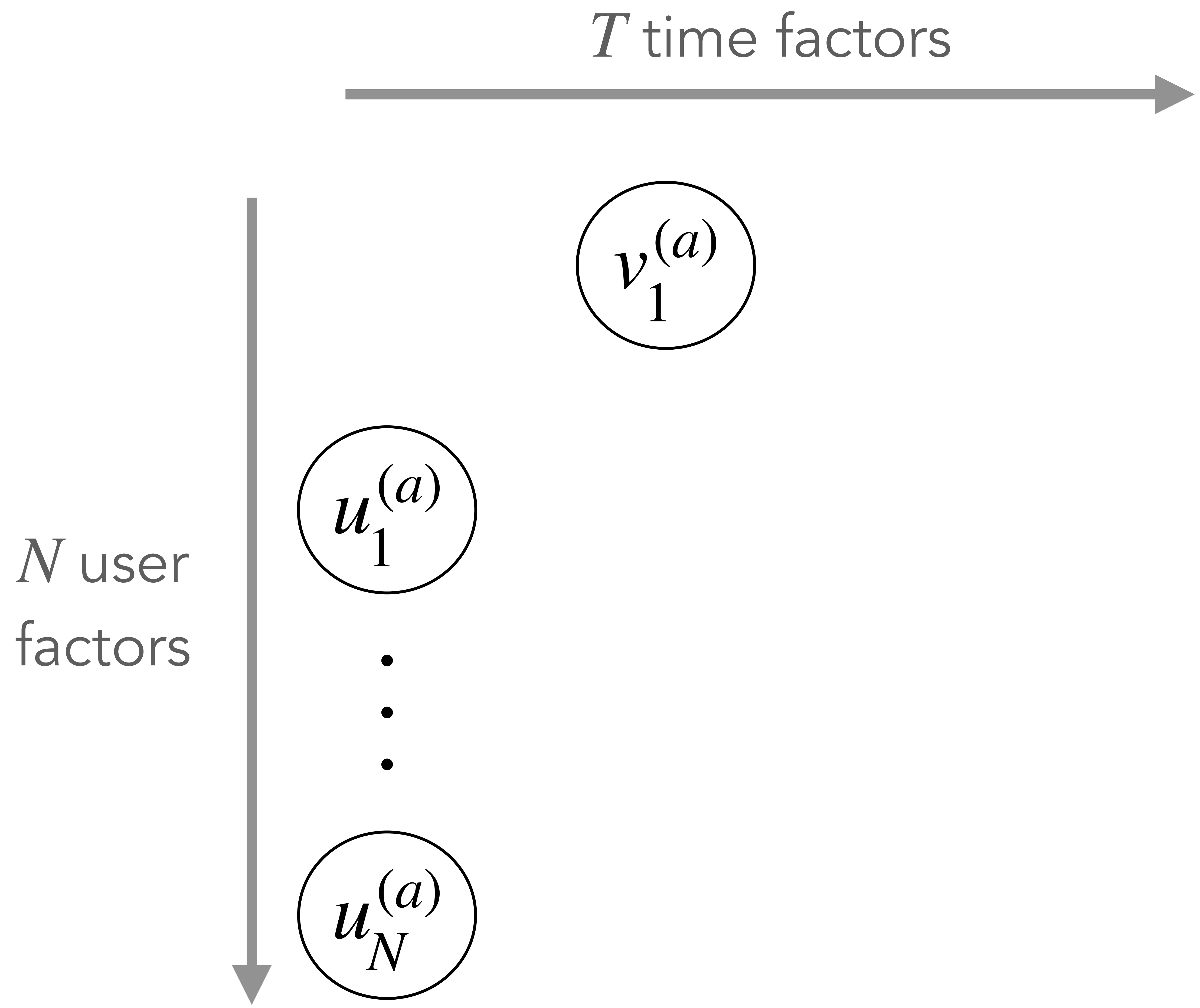
Structural assumption: Non-parametric factor model

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

$$\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)})$$

user factor (e.g., personal traits)

time factor (e.g., societal, weather changes)



No parametric assumptions on

- **unknown** non-linearity
- distributions of **unobserved** latent factors and noise

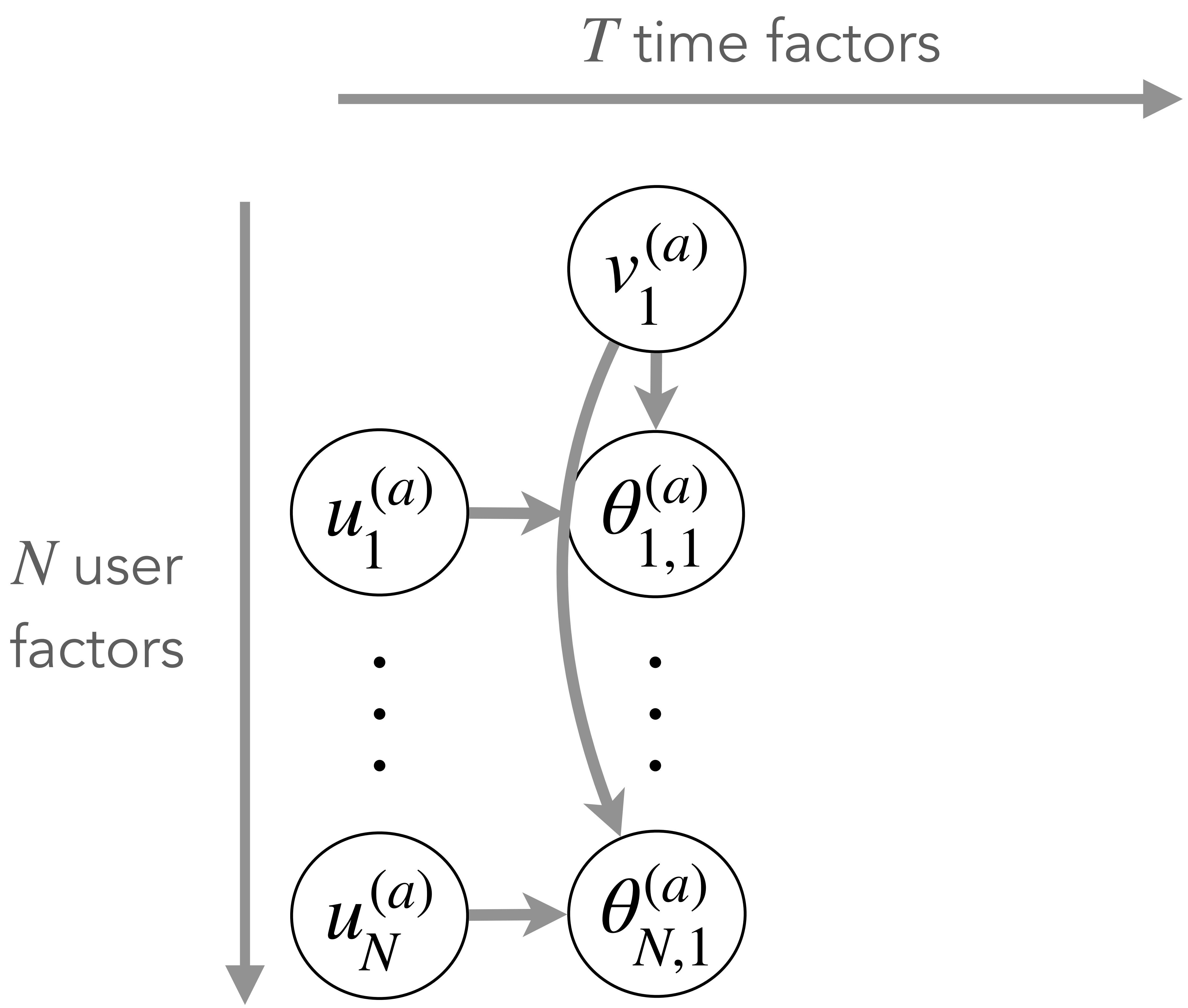
Structural assumption: Non-parametric factor model

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

$$\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)})$$

user factor (e.g., personal traits)

time factor (e.g., societal, weather changes)



No parametric assumptions on

- **unknown** non-linearity
- distributions of **unobserved** latent factors and noise

Structural assumption: Non-parametric factor model

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

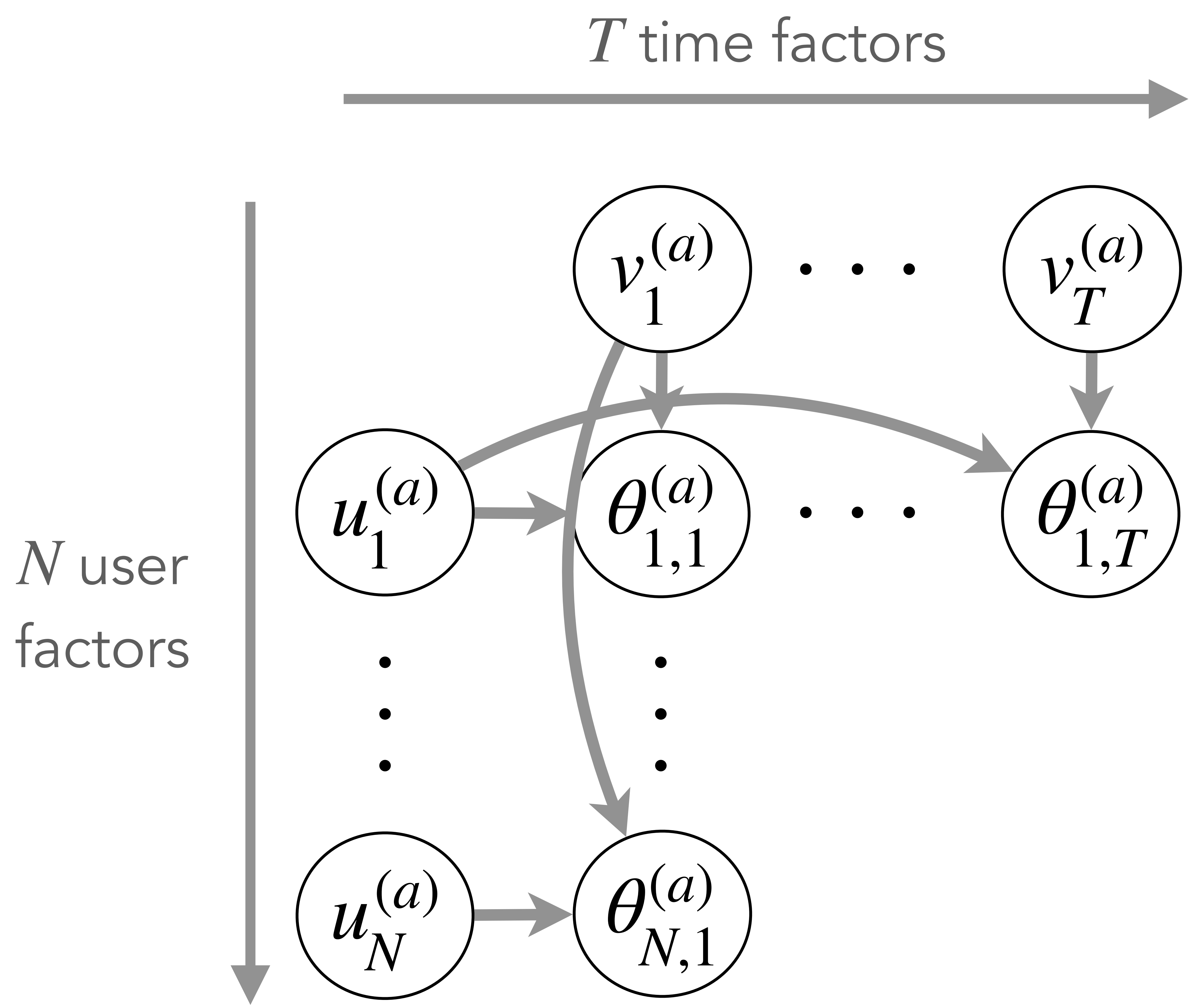
$$\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)})$$

user factor (e.g., personal traits)

time factor (e.g., societal, weather changes)

No parametric assumptions on

- **unknown** non-linearity
- distributions of **unobserved** latent factors and noise



Structural assumption: Non-parametric factor model

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

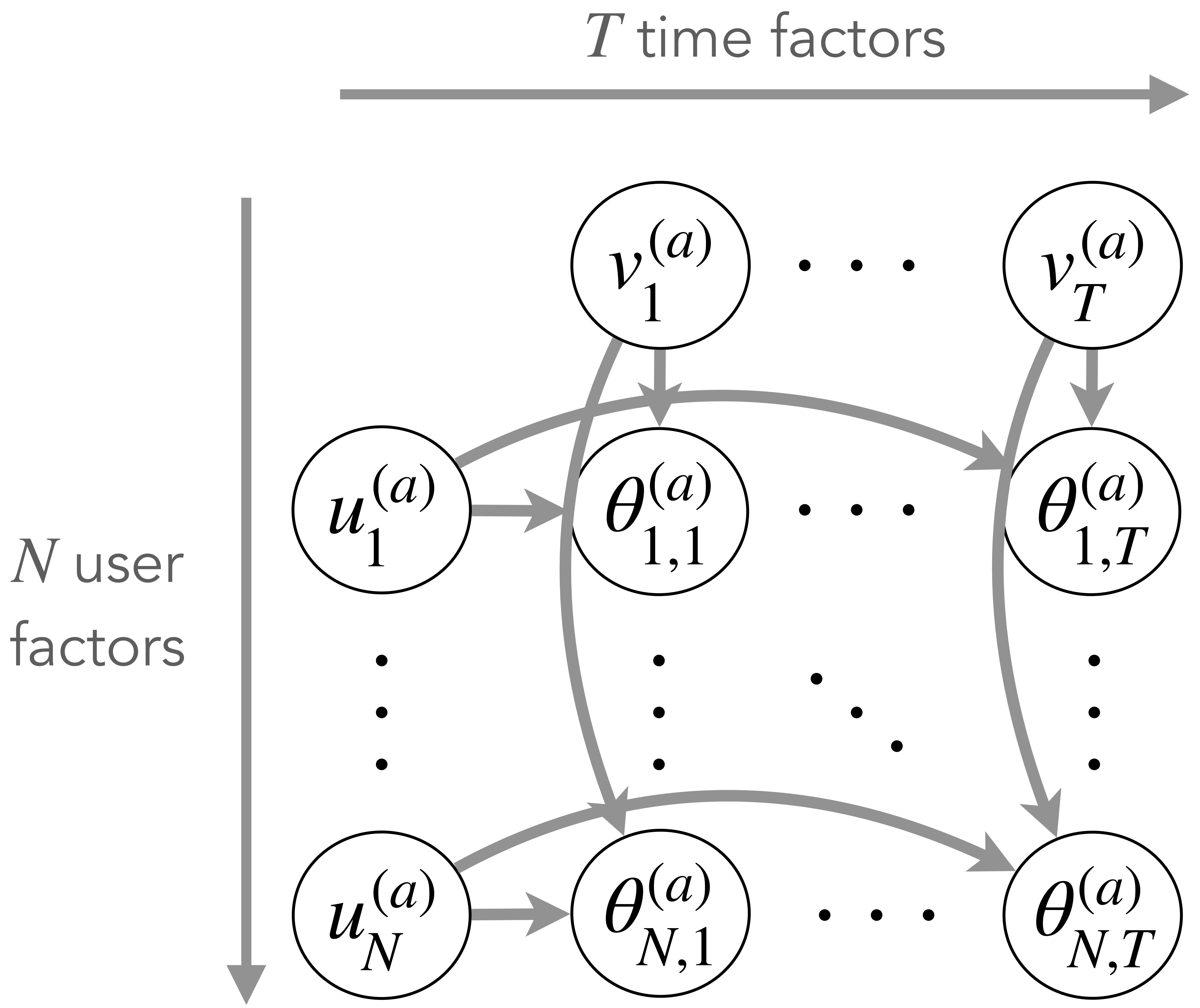
$$\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)})$$

user factor (e.g., personal traits)

time factor (e.g., societal, weather changes)

No parametric assumptions on

- **unknown** non-linearity
- distributions of **unobserved** latent factors and noise



User nearest neighbors estimator for $\theta_{i,t}^{(a)}$

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

User nearest neighbors estimator for $\theta_{i,t}^{(a)}$

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

1. Compute distance between two users i and j under treatment a

User nearest neighbors estimator for $\theta_{i,t}^{(a)}$

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

1. Compute distance between two users i and j under treatment a

$$\rho_{i,j}^{(a)} = \frac{\sum_{t'=1}^{T} (Y_{i,t'} - Y_{j,t'})^2 \cdot \mathbf{1}(A_{i,t'} = A_{j,t'} = a)}{\sum_{t'=1}^{T} \mathbf{1}(A_{i,t'} = A_{j,t'} = a)}$$

User nearest neighbors estimator for $\theta_{i,t}^{(a)}$

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

1. Compute distance between two users i and j under treatment a

$$\rho_{i,j}^{(a)} = \frac{\sum_{t'=1}^{T} (Y_{i,t'} - Y_{j,t'})^2 \cdot \mathbf{1}(A_{i,t'} = A_{j,t'} = a)}{\sum_{t'=1}^{T} \mathbf{1}(A_{i,t'} = A_{j,t'} = a)}$$

Squared distance between outcomes averaged over **<u>all times</u> when i and j are both treated with a**

User nearest neighbors estimator for $\theta_{i,t}^{(a)}$

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

1. Compute distance between two users i and j under treatment a

$$\rho_{i,j}^{(a)} = \frac{\sum_{t'=1}^{T} (Y_{i,t'} - Y_{j,t'})^2 \cdot \mathbf{1}(A_{i,t'} = A_{j,t'} = a)}{\sum_{t'=1}^{T} \mathbf{1}(A_{i,t'} = A_{j,t'} = a)}$$

Squared distance between outcomes averaged over **all times when i and j are both treated with a**

2. Average outcome across user neighbors treated with a at time t

$$\mathbf{1}(\rho_{i,j}^{(a)} \leq \eta, A_{j,t} = a)$$

User nearest neighbors estimator for $\theta_{i,t}^{(a)}$

$$Y_{i,t} = \theta_{i,t}^{(A_{i,t})} + \text{noise}_{i,t}$$

1. Compute distance between two users i and j under treatment a

$$\rho_{i,j}^{(a)} = \frac{\sum_{t'=1}^{T} (Y_{i,t'} - Y_{j,t'})^2 \cdot \mathbf{1}(A_{i,t'} = A_{j,t'} = a)}{\sum_{t'=1}^{T} \mathbf{1}(A_{i,t'} = A_{j,t'} = a)}$$

Squared distance between outcomes averaged over **all times when i and j are both treated with a**

2. Average outcome across user neighbors treated with a at time t

$$\widehat{\theta}_{i,t,\text{user-NN}}^{(a)} = \frac{\sum_{j=1}^{N} Y_{j,t} \cdot \mathbf{1}(\rho_{i,j}^{(a)} \le \eta, A_{j,t} = a)}{\sum_{j=1}^{N} \mathbf{1}(\rho_{i,j}^{(a)} \le \eta, A_{j,t} = a)}$$

Main result: A non-asymptotic guarantee for each (i, t, a)

Main result: A non-asymptotic guarantee for each (i, t, a)

Informal theorem: [**Dwivedi**-Tian-Tomkins-Klasnja-Murphy-Shah ’22a]

For suitably chosen η & under regularity conditions

Main result: A non-asymptotic guarantee for each (i, t, a)

Informal theorem: [**Dwivedi**-Tian-Tomkins-Klasnja-Murphy-Shah '22a]

For suitably chosen η & under regularity conditions

- Lipschitz non-linearity, iid latent factors, sub-Gaussian noise

Main result: A non-asymptotic guarantee for each (i, t, a)

Informal theorem: [**Dwivedi**-Tian-Tomkins-Klasnja-Murphy-Shah '22a]
For suitably chosen η & under regularity conditions

- Lipschitz non-linearity, iid latent factors, sub-Gaussian noise
- generic sequentially adaptive policies that assign treatments independently to users conditioned on observed history & choose a with probability $\geq p^{\dagger}$

(†Our general results allow p to decay as $\gtrsim T^{-1/2}$)

Main result: A non-asymptotic guarantee for each (i, t, a)

Informal theorem: [**Dwivedi**-Tian-Tomkins-Klasnja-Murphy-Shah '22a]
For suitably chosen η & under regularity conditions

- Lipschitz non-linearity, iid latent factors, sub-Gaussian noise
- generic sequentially adaptive policies that assign treatments independently to users conditioned on observed history & choose a with probability $\geq p^{\dagger}$

for each user i at each time t, with high probability

(†Our general results allow p to decay as $\gtrsim T^{-1/2}$)

Main result: A non-asymptotic guarantee for each (i, t, a)

Informal theorem: [**Dwivedi**-Tian-Tomkins-Klasnja-Murphy-Shah '22a]

For suitably chosen η & under regularity conditions

- Lipschitz non-linearity, iid latent factors, sub-Gaussian noise
- generic sequentially adaptive policies that assign treatments independently to users conditioned on observed history & choose a with probability $\geq p^{\dagger}$

for each user i at each time t, with high probability

$$|\,\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} - \theta^{(a)}_{i,t}\,| \lesssim \frac{1}{T^{1/4}} + \frac{1}{(N/M)^{1/2}}$$

User factor distribution

$\downarrow$

(Uniform on finite set of size M)

($\dagger$Our general results allow p to decay as $\gtrsim T^{-1/2}$)

Main result: A non-asymptotic guarantee for each (i, t, a)

Informal theorem: [**Dwivedi**-Tian-Tomkins-Klasnja-Murphy-Shah '22a]
For suitably chosen η & under regularity conditions

- Lipschitz non-linearity, iid latent factors, sub-Gaussian noise
- generic sequentially adaptive policies that assign treatments independently to users conditioned on observed history & choose a with probability $\geq p^{\dagger}$

for each user i at each time t, with high probability

User factor distribution

$$|\,\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} - \theta^{(a)}_{i,t}\,| \lesssim \frac{1}{T^{1/4}} + \frac{1}{(N/M)^{1/2}}$$

(Uniform on finite set of size M)

$$|\,\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} - \theta^{(a)}_{i,t}\,| \lesssim \frac{1}{T^{1/4}} + \frac{1}{N^{1/(d+2)}}$$

(Uniform over $[-1,1]^d$)

($\dagger$Our general results allow p to decay as $\gtrsim T^{-1/2}$)

User-NN guarantees: Advantages

User-NN guarantees: **Advantages**

- Asymptotic **confidence intervals** as $N, T \to \infty$:

$$\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} \pm \frac{1.96\widehat{\sigma}}{\sqrt{\#\text{neighbors}_{i,t,a}}}$$

User-NN guarantees: Advantages

- Asymptotic **confidence intervals** as $N, T \to \infty$:

$$\widehat{\theta}^{(a)}_{i,t,\text{user–NN}} \pm \frac{1.96\,\widehat{\sigma}}{\sqrt{\#\text{neighbors}_{i,t,a}}}$$

$$\downarrow$$

Confidence intervals for treatment effect $\theta^{(1)}_{i,t} - \theta^{(0)}_{i,t}$

User-NN guarantees: Advantages

- Asymptotic **confidence intervals** as $N, T \to \infty$:

$$\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} \pm \frac{1.96\,\widehat{\sigma}}{\sqrt{\#\text{neighbors}_{i,t,a}}}$$

$$\downarrow$$

Confidence intervals for treatment effect $\theta^{(1)}_{i,t} - \theta^{(0)}_{i,t}$

Challenges tackled: First guarantee for user-time-level counterfactuals

✓ More unknowns than observations

✓ Non-parametric model

✓ Heterogeneity across users & time

✓ Generic sequential policies

User-NN guarantees: Advantages

- Asymptotic **confidence intervals** as $N, T \to \infty$:

$$\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} \pm \frac{1.96\widehat{\sigma}}{\sqrt{\#\text{neighbors}_{i,t,a}}}$$

- $|\,\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} - \theta^{(a)}_{i,t}\,| = \tilde{O}\left(\frac{1}{T^{1/4}} + \frac{1}{\sqrt{N}}\right)$

Challenges tackled: First guarantee for user-time-level counterfactuals

✓ More unknowns than observations

✓ Non-parametric model

✓ Heterogeneity across users & time

✓ Generic sequential policies

User-NN guarantees: Advantages

- Asymptotic **confidence intervals** as $N, T \to \infty$:

$$\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} \pm \frac{1.96\widehat{\sigma}}{\sqrt{\#\text{neighbors}_{i,t,a}}}$$

- $|\,\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} - \theta^{(a)}_{i,t}\,| = \tilde{O}\left(\frac{1}{T^{1/4}} + \frac{1}{\sqrt{N}}\right)$

$$\Downarrow$$

$$|\,??\, - \theta^{(a)}_{i,t}\,| = \tilde{O}\left(\frac{1}{\sqrt{T}} + \frac{1}{\sqrt{N}}\right)$$

Can we improve the <u>slow rate</u> in T?

Challenges tackled: First guarantee for user-time-level counterfactuals

- ✓ More unknowns than observations
- ✓ Non-parametric model
- ✓ Heterogeneity across users & time
- ✓ Generic sequential policies

Yes, we can!

A near-quadratic improvement over user-NN

Yes, we can!

A near-quadratic improvement over user-NN

Informal theorem: [**Dwivedi**-Tian-Tomkins-Klasnja-Murphy-Shah '22b]

A suitable variant of nearest neighbors improves* upon the user-NN error

$$| \widehat{\theta}^{(a)}_{i,t,\text{user–NN}} - \theta^{(a)}_{i,t} | = \tilde{O}\left(\frac{1}{T^{1/4}} + \frac{1}{\sqrt{N}}\right)$$

$$\Big\downarrow$$

$$| \widehat{\theta}^{(a)}_{i,t,\text{DR–NN}} - \theta^{(a)}_{i,t} | = \tilde{O}\left(\frac{1}{\sqrt{T}} + \frac{1}{\sqrt{N}}\right)$$

Yes, we can!

A near-quadratic improvement over user-NN

Informal theorem: [**Dwivedi**-Tian-Tomkins-Klasnja-Murphy-Shah '22b]

A suitable variant of nearest neighbors improves* upon the user-NN error

$$|\,\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} - \theta^{(a)}_{i,t}\,| = \tilde{O}\left(\frac{1}{T^{1/4}} + \frac{1}{\sqrt{N}}\right)$$

$$\Big\downarrow$$

$$|\,\widehat{\theta}^{(a)}_{i,t,\text{DR-NN}} - \theta^{(a)}_{i,t}\,| = \tilde{O}\left(\frac{1}{\sqrt{T}} + \frac{1}{\sqrt{N}}\right)$$

*for Lipschitz non-linearity with Lipschitz gradients & non-adaptive policies

Proof intuition for user-NN

Proof intuition for user-NN

Simple case: Estimate $\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)}) \;= u_i v_t$

Proof intuition for user-NN

Simple case: Estimate $\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)}) \ = u_i v_t$

- $\widehat{\theta}_{i,t,\text{user-NN}}^{(a)} \quad = \dfrac{\sum_{j \in \text{user-nn}} Y_{j,t}}{\#\ \text{user-nn}} \ = \dfrac{\sum_{j \in \text{user-nn}} \theta_{j,t}^{(a)} + \text{noise}_{j,t}}{\#\ \text{user-nn}}$

Proof intuition for user-NN

Simple case: Estimate $\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)}) \;= u_i v_t$

- $\widehat{\theta}_{i,t,\text{user-NN}}^{(a)} \quad = \dfrac{\sum_{j\in\text{user-nn}} Y_{j,t}}{\#\ \text{user-nn}} = \dfrac{\sum_{j\in\text{user-nn}} \theta_{j,t}^{(a)} + \text{noise}_{j,t}}{\#\ \text{user-nn}}$

$$= \underbrace{\boxed{\frac{\sum_{j\in\text{user-nn}} u_j}{\#\ \text{user-nn}}}}_{\hat{u}_i} v_t + \text{avg. noise}_t$$

Proof intuition for user-NN

Simple case: Estimate $\theta^{(a)}_{i,t} \triangleq f^{(a)}(u^{(a)}_i, v^{(a)}_t) \;= u_i v_t$

- $\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} = \dfrac{\sum_{j\in\text{user-nn}} Y_{j,t}}{\#\text{ user-nn}} = \dfrac{\sum_{j\in\text{user-nn}} \theta^{(a)}_{j,t} + \text{noise}_{j,t}}{\#\text{ user-nn}}$

$$= \underbrace{\boxed{\dfrac{\sum_{j\in\text{user-nn}} u_j}{\#\text{ user-nn}}}}_{\hat{u}_i} v_t + \text{avg. noise}_t$$

- $|u_i v_t - \widehat{\theta}^{(a)}_{i,t,\text{user-NN}}| \leq |u_i v_t - \hat{u}_i v_t| + |\text{avg. noise}_t| = O(|u_i - \hat{u}_i|)$

Proof intuition for user-NN

Simple case: Estimate $\theta^{(a)}_{i,t} \triangleq f^{(a)}(u^{(a)}_i, v^{(a)}_t) \; = u_i v_t$

- $\widehat{\theta}^{(a)}_{i,t,\text{user-NN}} = \dfrac{\sum_{j \in \text{user-nn}} Y_{j,t}}{\text{\# user-nn}} = \dfrac{\sum_{j \in \text{user-nn}} \theta^{(a)}_{j,t} + \text{noise}_{j,t}}{\text{\# user-nn}}$

$$= \underbrace{\boxed{\dfrac{\sum_{j \in \text{user-nn}} u_j}{\text{\# user-nn}}}}_{\hat{u}_i} v_t + \text{avg. noise}_t$$

- $|u_i v_t - \widehat{\theta}^{(a)}_{i,t,\text{user-NN}}| \leq |u_i v_t - \hat{u}_i v_t| + |\text{avg. noise}_t| = O(|u_i - \hat{u}_i|)$

? noise at t correlated with user neighbors (sequential policy)

Proof intuition for user-NN

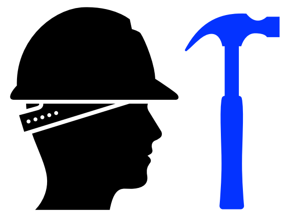
Simple case: Estimate $\theta^{(a)}_{i,t} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)}) \;= u_i v_t$

- $$\widehat{\theta}^{(a)}_{i,t,\text{user–NN}} = \frac{\sum_{j\in\text{user-nn}} Y_{j,t}}{\#\text{ user-nn}} = \frac{\sum_{j\in\text{user-nn}} \theta^{(a)}_{j,t} + \text{noise}_{j,t}}{\#\text{ user-nn}}$$

$$= \underbrace{\boxed{\frac{\sum_{j\in\text{user-nn}} u_j}{\#\text{ user-nn}}}}_{\hat{u}_i} v_t + \text{avg. noise}_t$$

- $|u_i v_t - \widehat{\theta}^{(a)}_{i,t,\text{user–NN}}| \leq |u_i v_t - \hat{u}_i v_t| + |\text{avg. noise}_t| = O(|u_i - \hat{u}_i|)$

? noise at t correlated with user neighbors (sequential policy)



Martingale concentration, **new sandwich argument** for nn

Proof intuition for user-NN

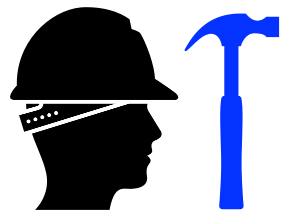
Simple case: Estimate $\theta_{i,t}^{(a)} \triangleq f^{(a)}(u_i^{(a)}, v_t^{(a)}) = u_i v_t$

- $\widehat{\theta}_{i,t,\text{user–NN}}^{(a)} = \dfrac{\sum_{j \in \text{user-nn}} Y_{j,t}}{\#\text{ user-nn}} = \dfrac{\sum_{j \in \text{user-nn}} \theta_{j,t}^{(a)} + \text{noise}_{j,t}}{\#\text{ user-nn}}$

$$= \underbrace{\boxed{\frac{\sum_{j \in \text{user-nn}} u_j}{\#\text{ user-nn}}}}_{\hat{u}_i} v_t + \text{avg. noise}_t$$

- $|u_i v_t - \widehat{\theta}_{i,t,\text{user–NN}}^{(a)}| \le |u_i v_t - \hat{u}_i v_t| + |\text{avg. noise}_t| = O(|u_i - \hat{u}_i|)$
- $|u_i v_t - \widehat{\theta}_{i,t,\text{time–NN}}^{(a)}| \le |u_i v_t - u_i \hat{v}_t| + |\text{avg. noise}_i| = O(|v_t - \hat{v}_t|)$

? noise at t correlated with user neighbors (sequential policy)



Martingale concentration, **new sandwich argument** for nn

Steps towards the improved estimator…

Steps towards the improved estimator…

- **Plug-in** principle: $|u_i v_t - \hat{u}_i \hat{v}_t| \leq |u_i v_t - \hat{u}_i v_t| + |\hat{u}_i v_t - \hat{u}_i \hat{v}_t|$

$$= O(|u_i - \hat{u}_i| + |v_t - \hat{v}_t|)$$

Steps towards the improved estimator…

- **Plug-in** principle: $|u_i v_t - \hat{u}_i \hat{v}_t| \leq |u_i v_t - \hat{u}_i v_t| + |\hat{u}_i v_t - \hat{u}_i \hat{v}_t|$

$$= O(|u_i - \hat{u}_i| + |v_t - \hat{v}_t|)$$

- **Convert $+$ to $\times$:** $|u_i v_t - \text{??}| = O(|u_i - \hat{u}_i| \times |v_t - \hat{v}_t|)$

Steps towards the improved estimator…

- **Plug-in** principle: $|u_i v_t - \hat{u}_i \hat{v}_t| \leq |u_i v_t - \hat{u}_i v_t| + |\hat{u}_i v_t - \hat{u}_i \hat{v}_t|$

$$= O(|u_i - \hat{u}_i| + |v_t - \hat{v}_t|)$$

$$\approx \max\{|\hat{u}_i - u_i|, |v_t - \hat{v}_t|\}$$

- **Convert + to ×:** $|u_i v_t - ??| = O(|u_i - \hat{u}_i| \times |v_t - \hat{v}_t|)$

$$\approx \min\{|\hat{u}_i - u_i|, |v_t - \hat{v}_t|\}$$

What should be our estimator? Let's expand the RHS…

What should be our estimator? Let's expand the RHS...

$$u_i v_t - \text{??} = (u_i - \hat{u}_i) \times (v_t - \hat{v}_t)$$

What should be our estimator? Let's expand the RHS...

$$\cancel{u_i v_t} - \text{??} = (u_i - \hat{u}_i) \times (v_t - \hat{v}_t)$$

$$= \cancel{u_i v_t} - \hat{u}_i v_t - u_i \hat{v}_t + \hat{u}_i \hat{v}_t$$

$$\Rightarrow \qquad \text{??} = \hat{u}_i v_t + u_i \hat{v}_t - \hat{u}_i \hat{v}_t$$

What should be our estimator? Let's expand the RHS...

$$\cancel{u_i v_t} - ?? = (u_i - \hat{u}_i) \times (v_t - \hat{v}_t)$$

$$= \cancel{u_i v_t} - \hat{u}_i v_t - u_i \hat{v}_t + \hat{u}_i \hat{v}_t$$

$$\Rightarrow \qquad ?? = \hat{u}_i v_t + u_i \hat{v}_t - \hat{u}_i \hat{v}_t$$

$$Y_{j,t} + Y_{i,t'} - Y_{j,t'}$$

$$\rho_{i,j}^{(a)} \leq \eta, \quad \rho_{t,t'}^{(a)} \leq \eta'$$

This is our improved nearest neighbors estimator!

$$\cancel{u_i v_t} - ?? = (u_i - \hat{u}_i) \times (v_t - \hat{v}_t)$$

$$= \cancel{u_i v_t} - \hat{u}_i v_t - u_i \hat{v}_t + \hat{u}_i \hat{v}_t$$

$$\Rightarrow \qquad ?? = \hat{u}_i v_t + u_i \hat{v}_t - \hat{u}_i \hat{v}_t$$

$$\widehat{\theta}^{(a)}_{i,t,\mathrm{DR\text{–}NN}} = \frac{\sum_{j,t'} (Y_{j,t} + Y_{i,t'} - Y_{j,t'}) \mathbf{1}_{i,t,j,t'}}{\sum_{j,t'} \mathbf{1}_{i,t,j,t'}}$$

$$\mathbf{1}_{i,t,j,t'} = \mathbf{1}(\rho^{(a)}_{i,j} \leq \eta,\ \ \rho^{(a)}_{t,t'} \leq \eta', A_{j,t} = A_{i,t'} = A_{j,t'} = a)$$

This is our improved nearest neighbors estimator!

$$\cancel{u_i v_t} - \text{??} = (u_i - \hat{u}_i) \times (v_t - \hat{v}_t)$$

$$= \cancel{u_i v_t} - \hat{u}_i v_t - u_i \hat{v}_t + \hat{u}_i \hat{v}_t$$

$$\Rightarrow \quad \text{??} = \hat{u}_i v_t + u_i \hat{v}_t - \hat{u}_i \hat{v}_t$$

DR-NN error $\approx$ **user-NN error** $\times$ **time-NN error**

$\lesssim$ **min{user-NN error, time-NN error}**

This is our improved nearest neighbors estimator!

$$\cancel{u_i v_t} - \text{??} = (u_i - \hat{u}_i) \times (v_t - \hat{v}_t)$$

$$= \cancel{u_i v_t} - \hat{u}_i v_t - u_i \hat{v}_t + \hat{u}_i \hat{v}_t$$

$$\Rightarrow \quad \text{??} = \hat{u}_i v_t + u_i \hat{v}_t - \hat{u}_i \hat{v}_t$$

DR-NN error $\approx$ **user-NN error** $\times$ **time-NN error**

$\lesssim$ **min{user-NN error, time-NN error}**

Doubly robust to heterogeneity in user factors & time factors

Double robustness, double machine learning…

[… Cassel+ '77, Robinson '88, Särndal+ '89, Robins+ '94, '95, '08, '09, Newey+ '94, '18, Bickel+ '98, van der Laan+ '03, Lunceford+ '04, Davidian+ '05, Li+ '11, Jiang+ '15, Chernozhukov+ '18, Hirshberg+ '18, Diaz '19, Arkhangelsky+ '21, Dorn+ '21 …]

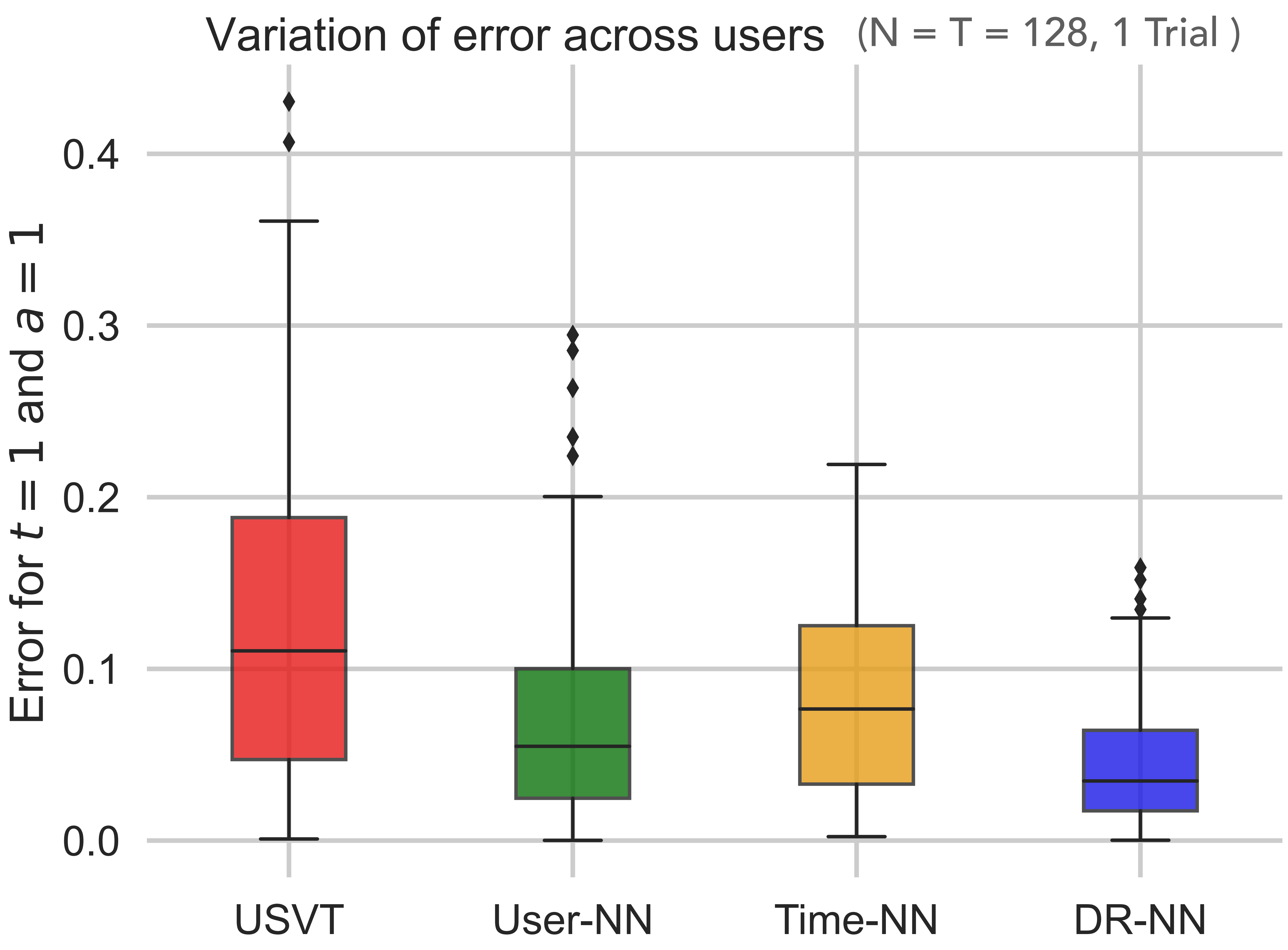
Simulation results

Simulation results

Uniform latent factors on $[-0.5,0.5]^4$, Gaussian noise, pooled ε-greedy policy ($\varepsilon = 0.5$)

Simulation results

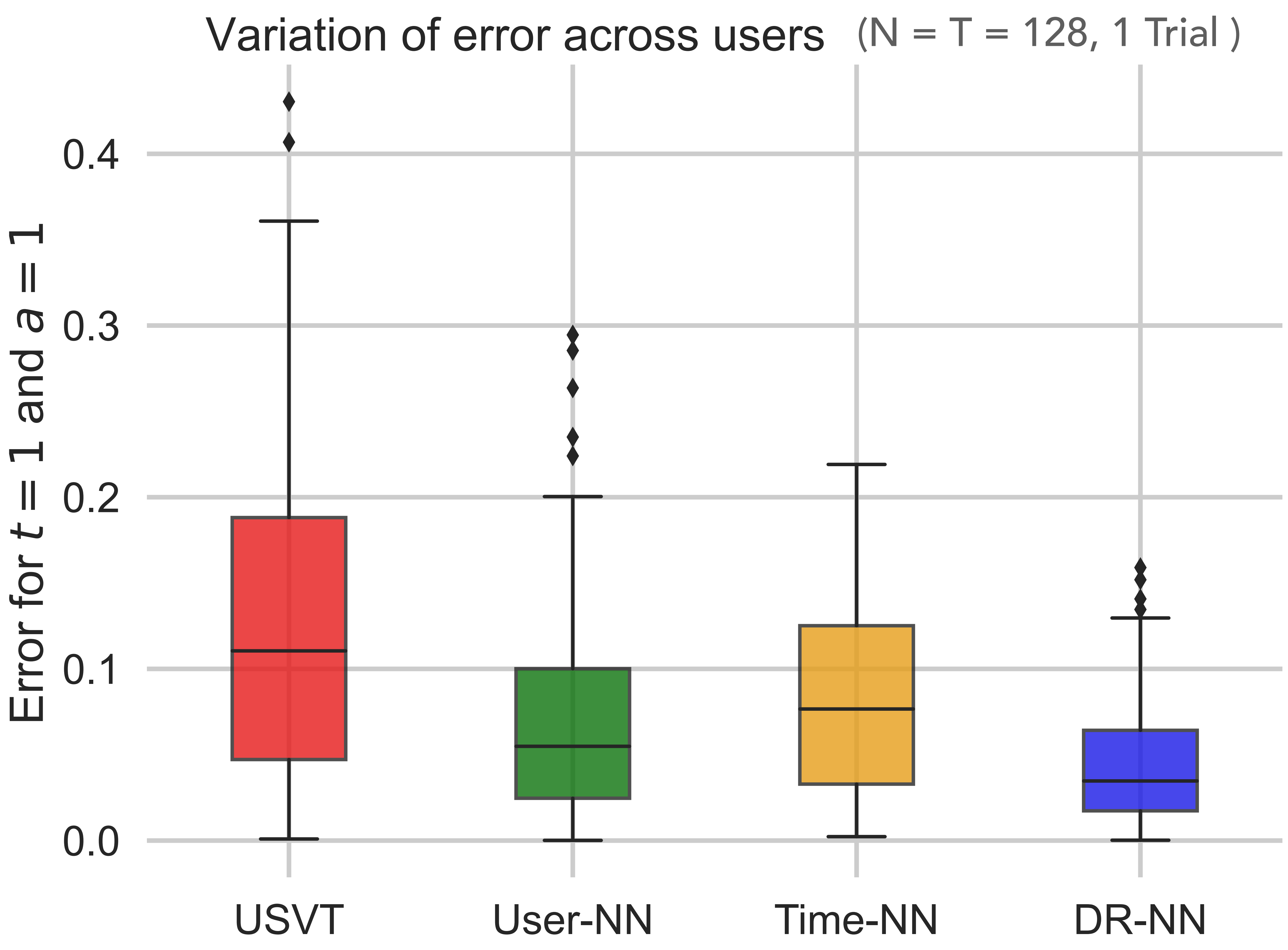
Uniform latent factors on $[-0.5,0.5]^4$, Gaussian noise, pooled ε-greedy policy ($\varepsilon = 0.5$)

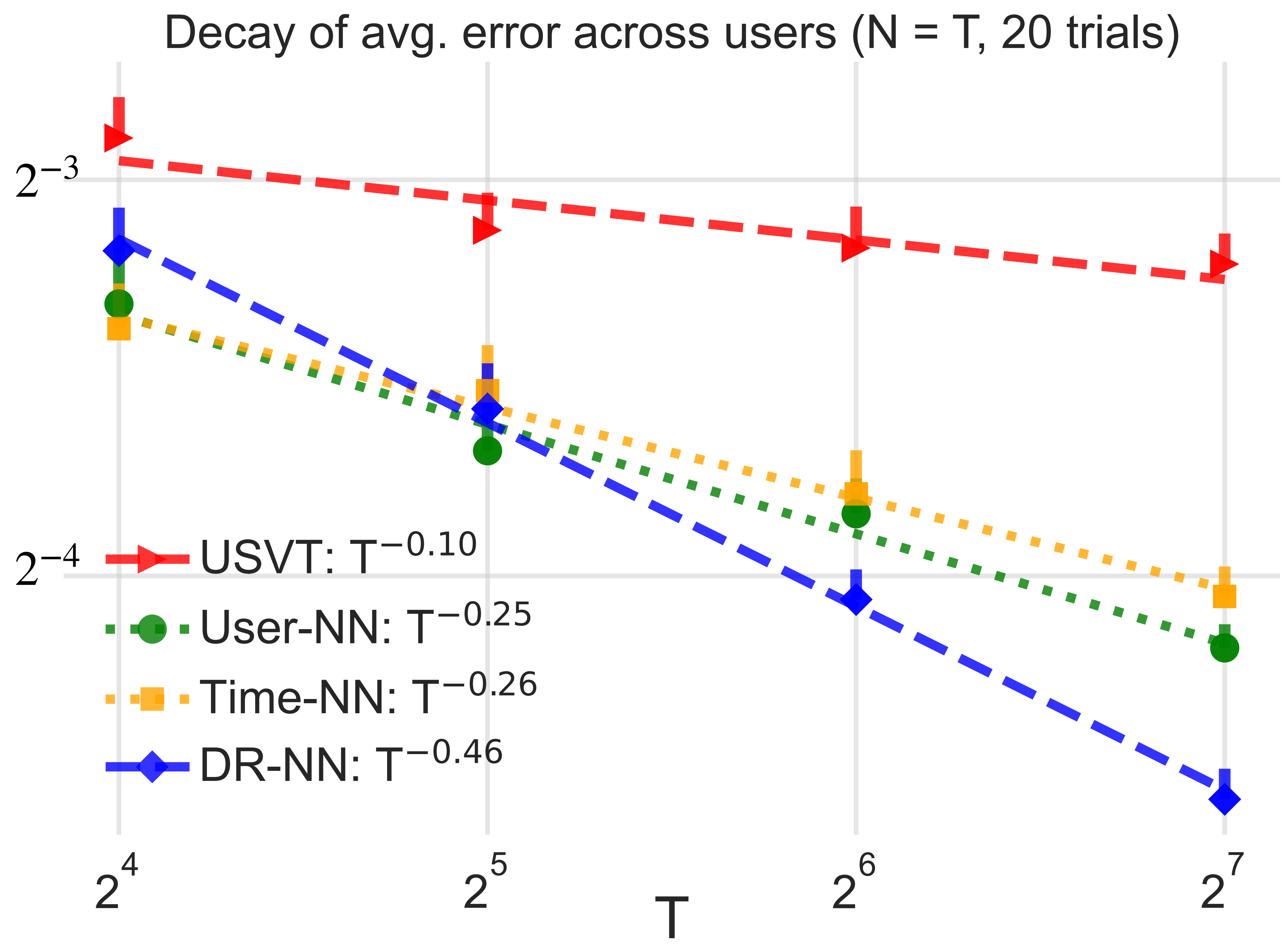


A baseline algorithm from [Chatterjee 2014]

Simulation results

Uniform latent factors on $[-0.5, 0.5]^4$, Gaussian noise, pooled ε-greedy policy ($\varepsilon = 0.5$)





A baseline algorithm from [Chatterjee 2014]

DR-NN error $\ll$ min { user-NN error, time-NN error }

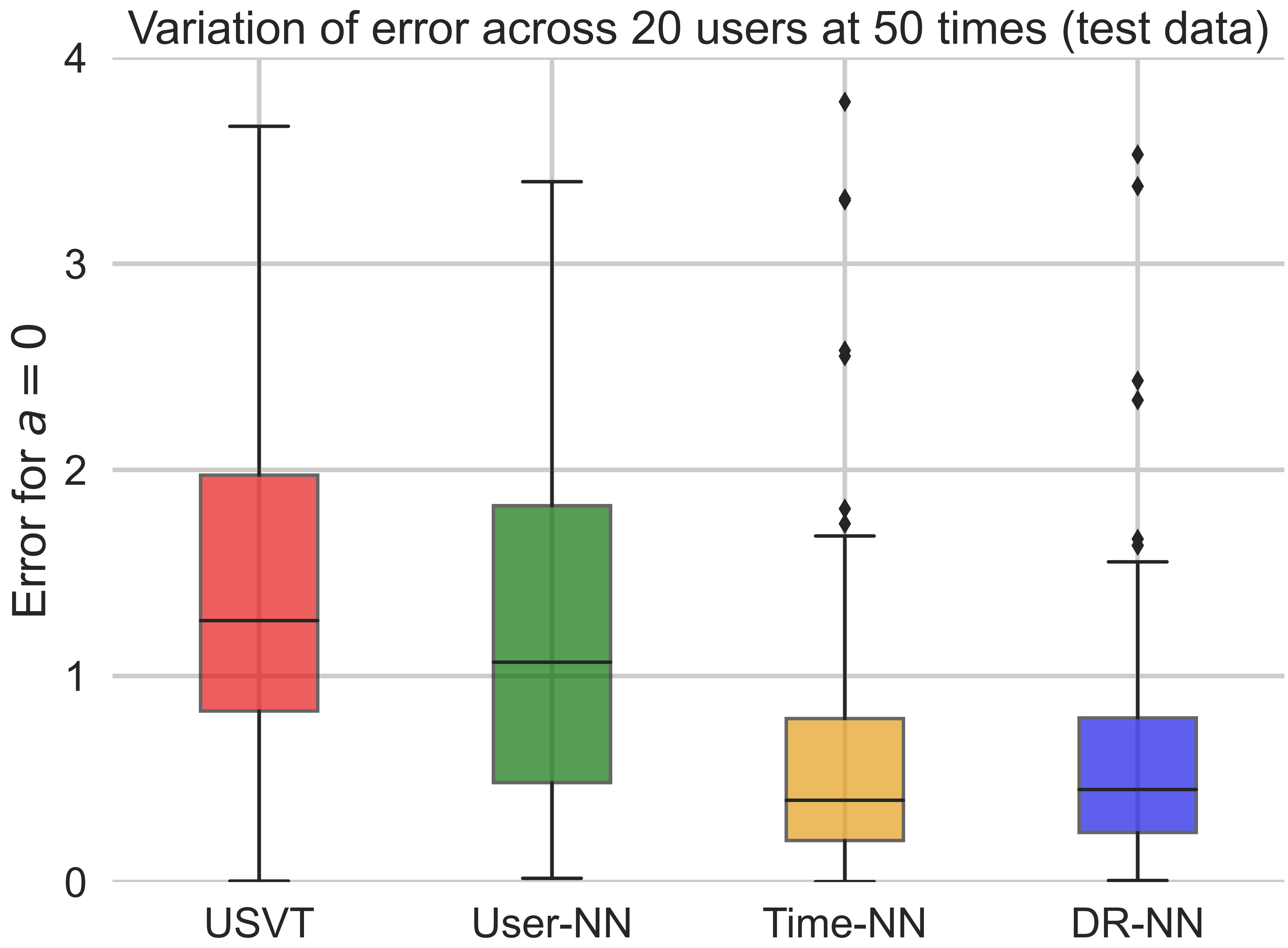
Personalized HeartSteps results

Personalized HeartSteps results 

Treatments assigned with Thompson sampling independently for 91 users for 90 days, 5 times a day

Personalized HeartSteps results

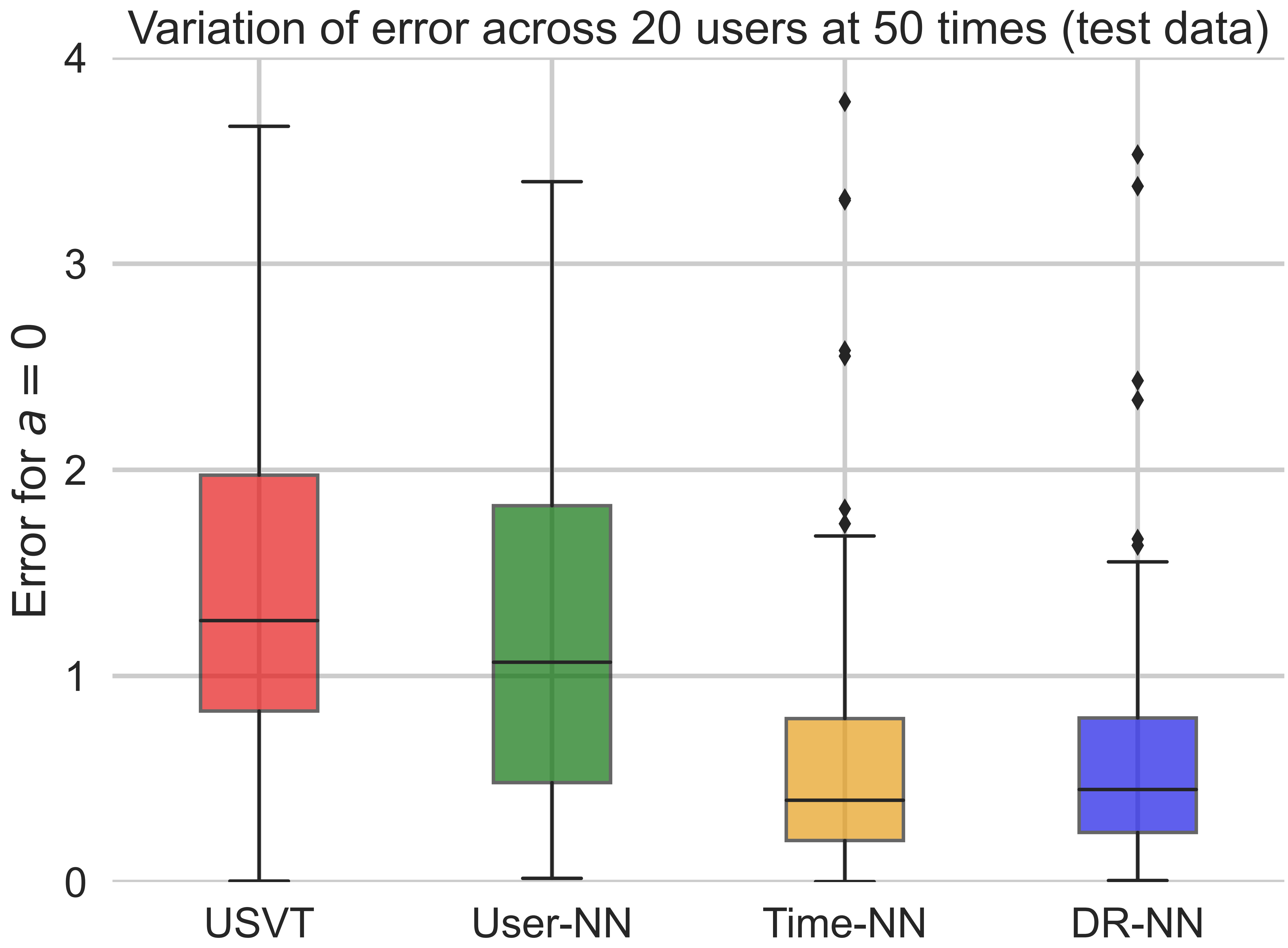
Treatments assigned with Thompson sampling independently for 91 users for 90 days, 5 times a day

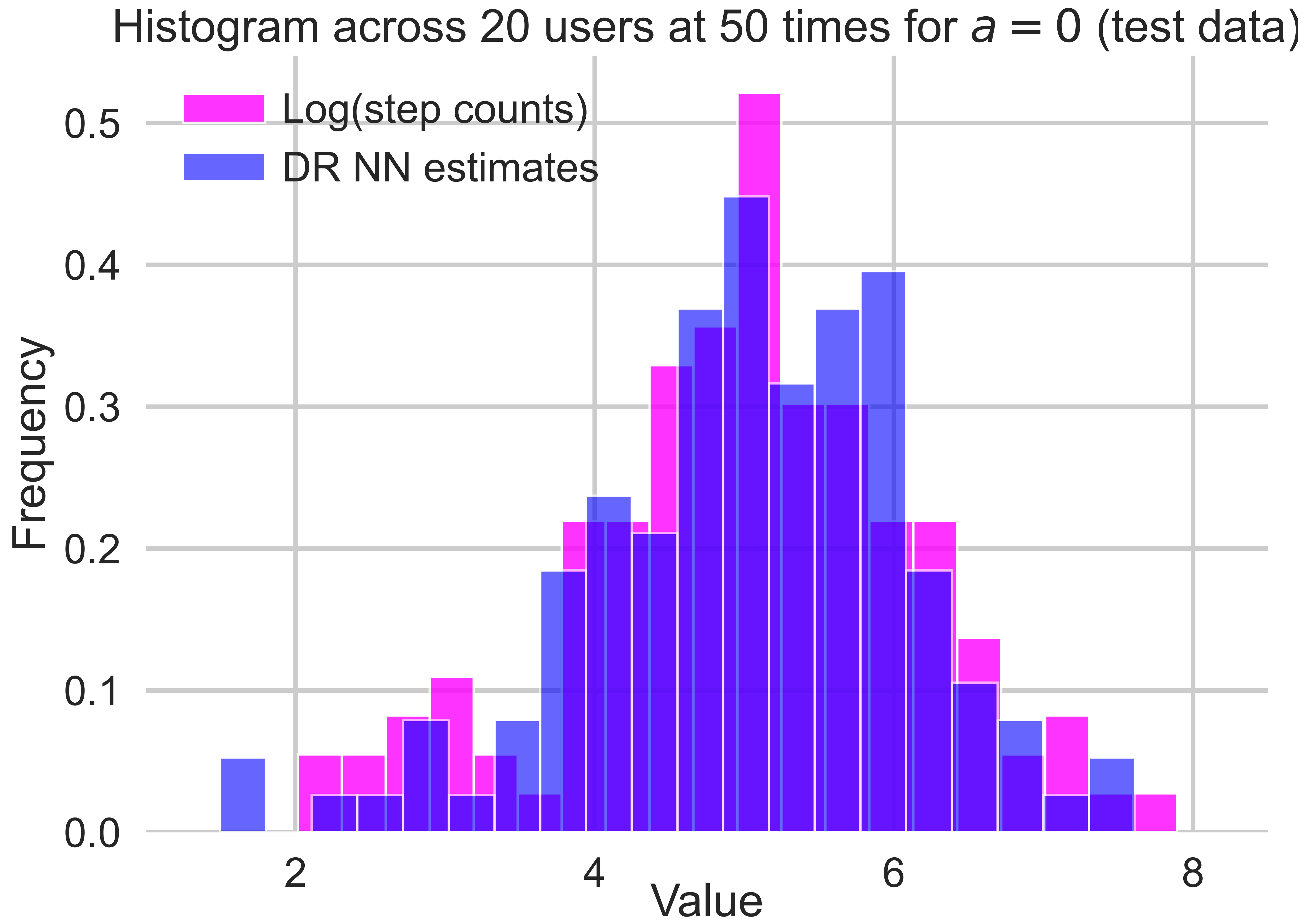


DR-NN error ≈ min { user-NN error, time-NN error }

Personalized HeartSteps results 

Treatments assigned with Thompson sampling independently for 91 users for 90 days, 5 times a day





DR-NN error ≈ min { user-NN error, time-NN error }

Part 1 summary:
Sample-efficient inference with non-parametric factor models

Part 1 summary:
Sample-efficient inference with non-parametric factor models

✓ Inference in sequential experiments: User-NN with $\tilde{O}(T^{-1/4})$ error

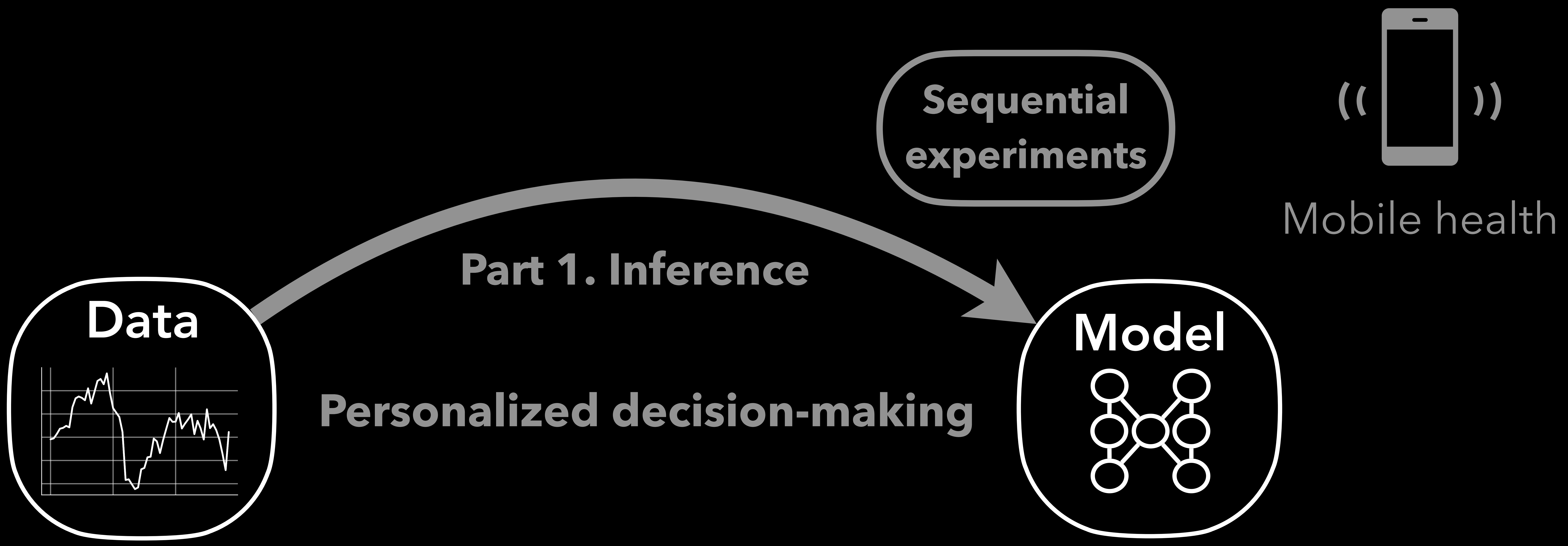
✓ Efficient estimators: Doubly robust-NN with $\tilde{O}(T^{-1/2})$ error

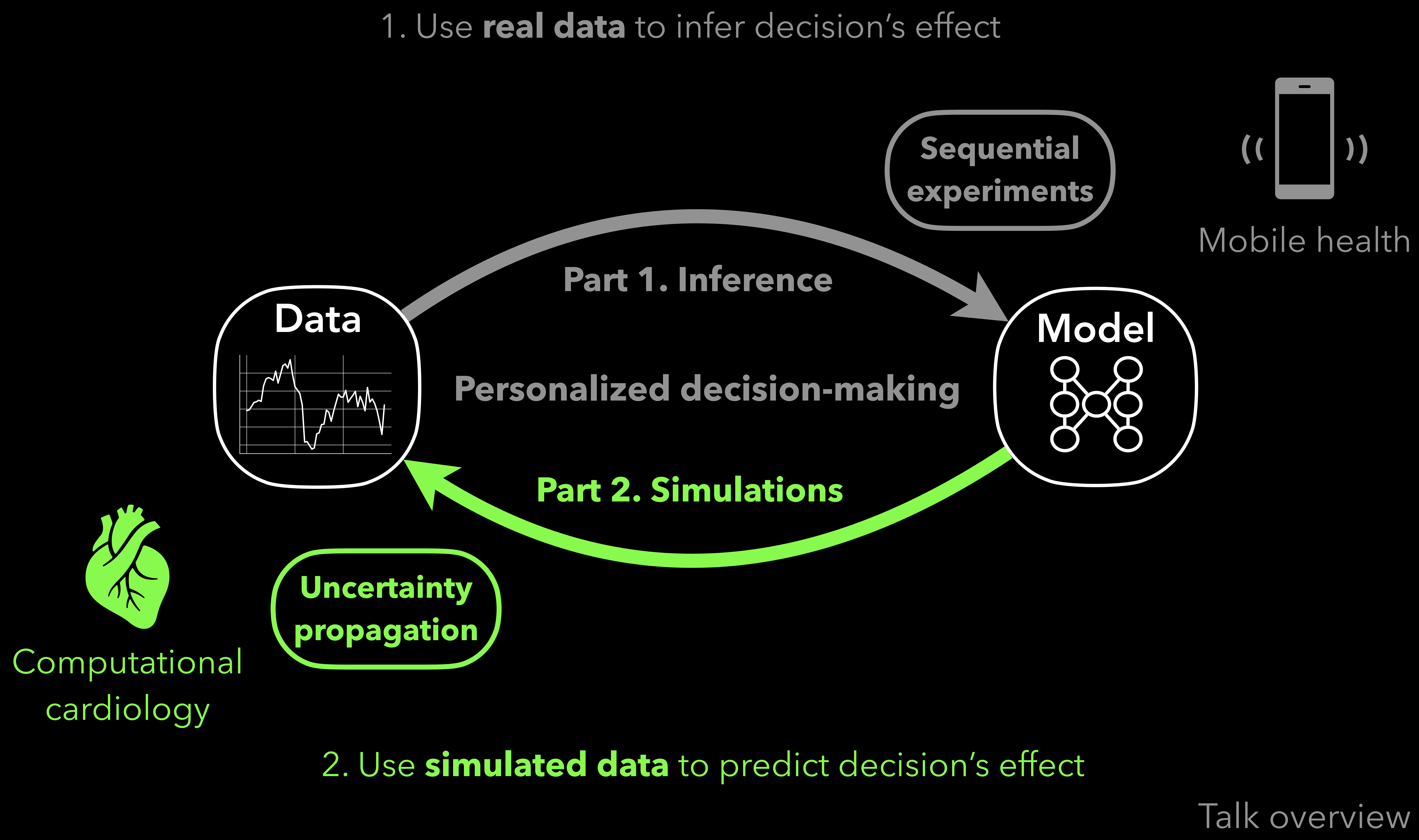
DR-NN error $\approx$ **user-NN error** $\times$ **time-NN error**

$\lesssim$ **min{user-NN error, time-NN error}**

✦ Future: Settings with contexts and covariates

1. Use **real data** to infer decision's effect



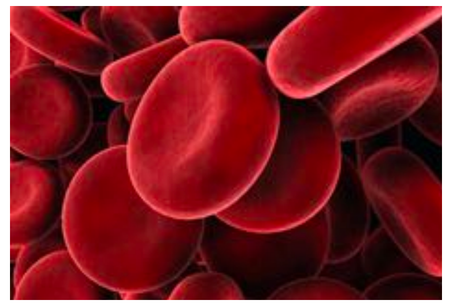

1. Use **real data** to infer decision's effect
Sequential experiments
Mobile health
Part 1. Inference
Data
Model
Personalized decision-making
Part 2. Simulations
Uncertainty propagation
Computational cardiology
2. Use **simulated data** to predict decision's effect
Talk overview

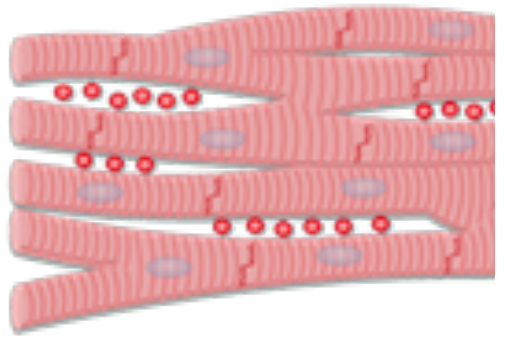
Complex multi-scale simulation systems

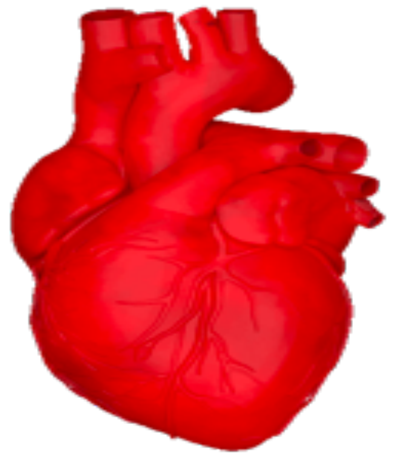
Complex multi-scale simulation systems

Basic unit | Sub-component | System

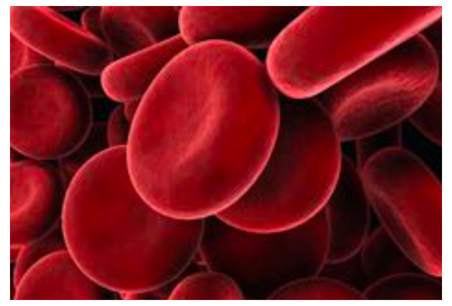
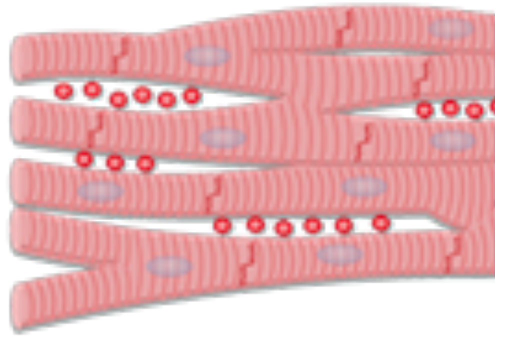
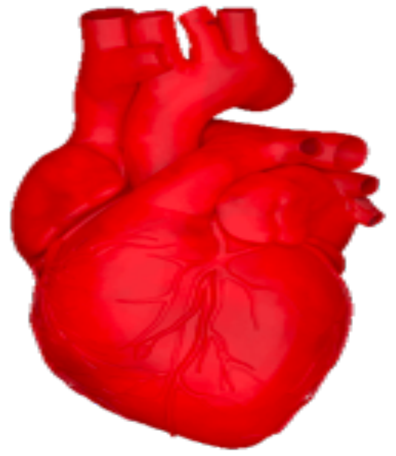
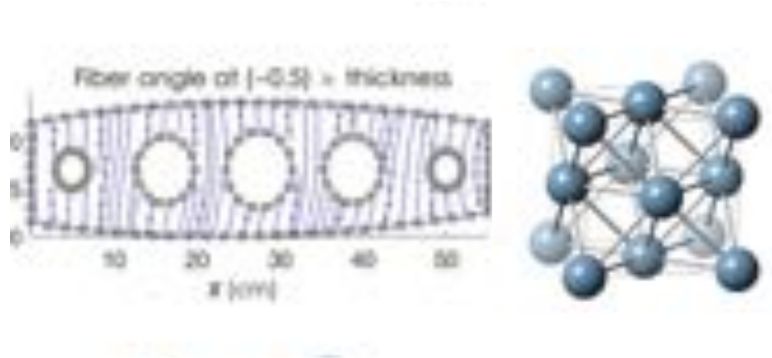
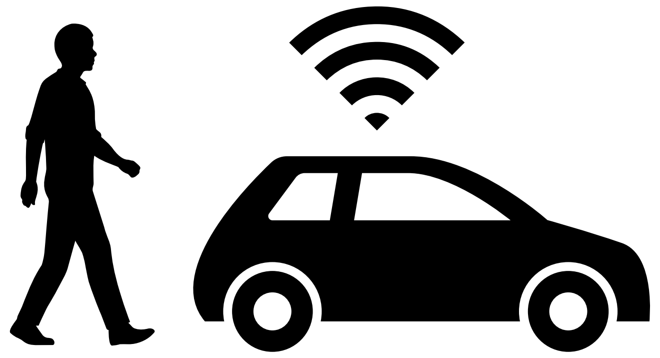
Cardiology



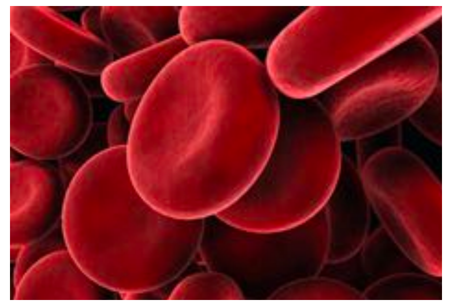
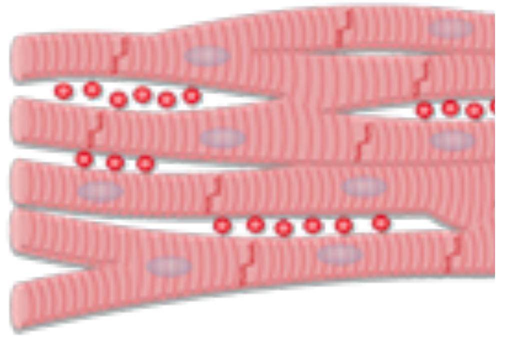
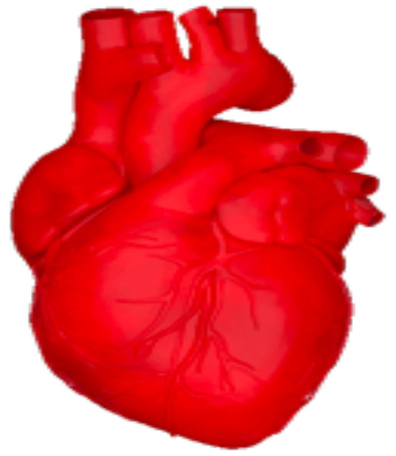
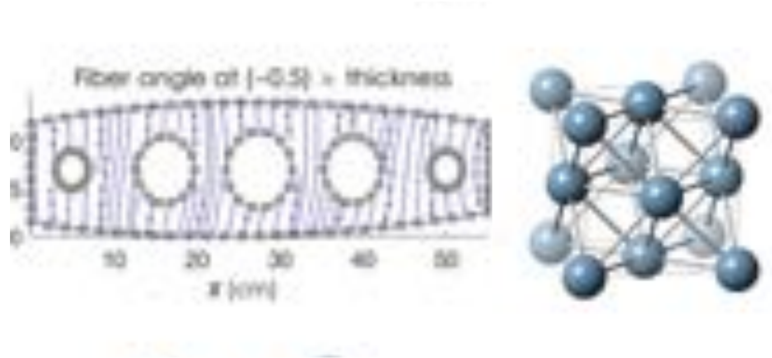
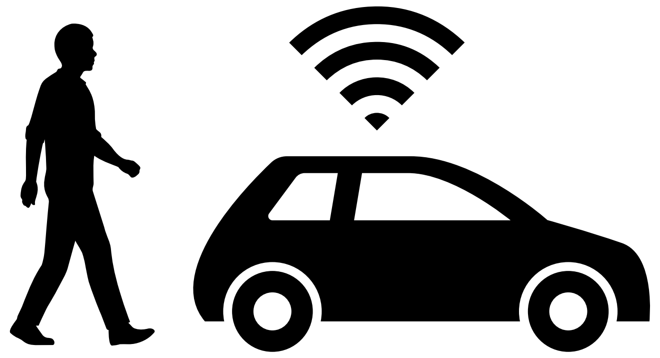




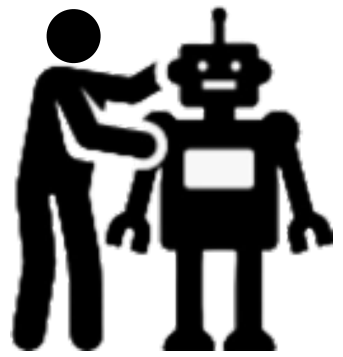
Complex multi-scale simulation systems

	Basic unit	Sub-component	System
Cardiology			
Aerospace			
Self-driving			

Complex multi-scale simulation systems

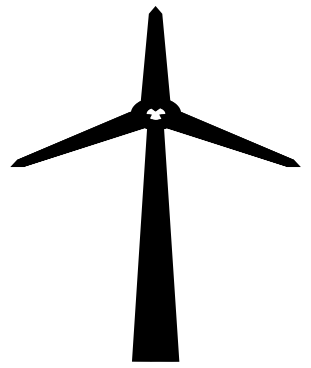
	Basic unit	Sub-component	System
Cardiology			
Aerospace			
Self-driving			

…



Human-robot interaction

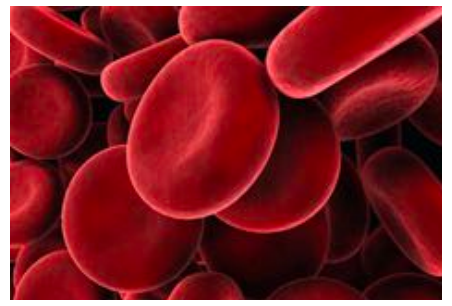
Built environment (city planning)

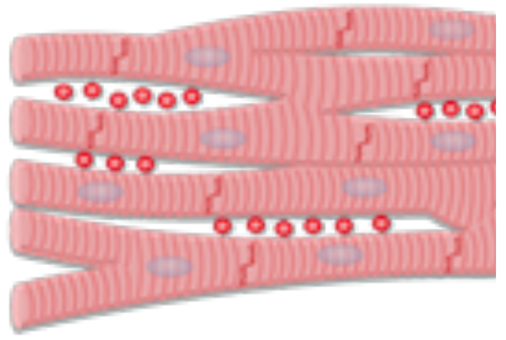


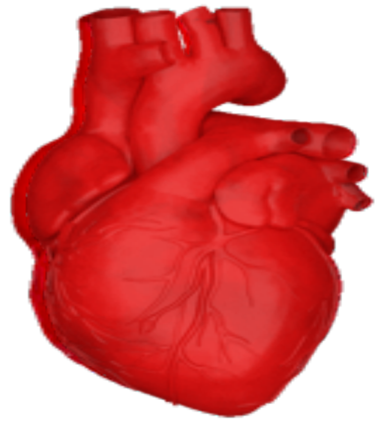
Plant design

Computational cardiology: Personalized HeartBeats

Cell model — Tissue — Heart model

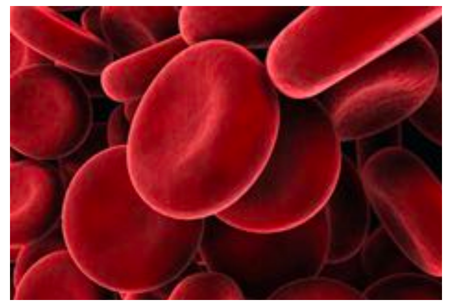




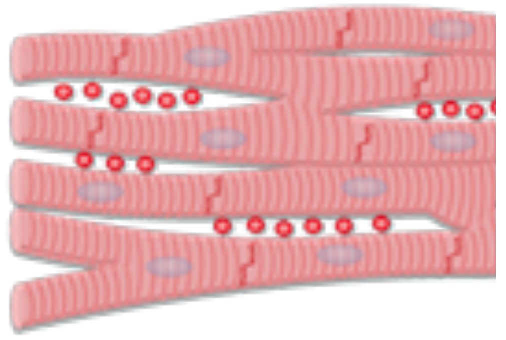


Computational cardiology: Personalized HeartBeats

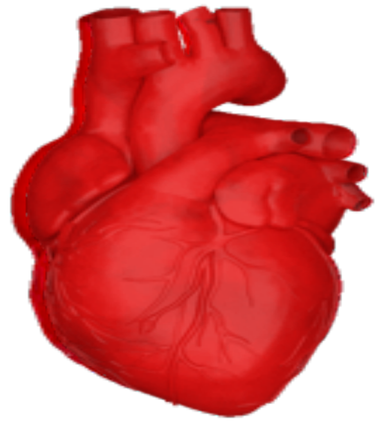
Cell model



Tissue

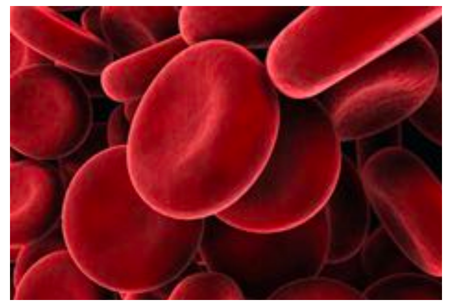


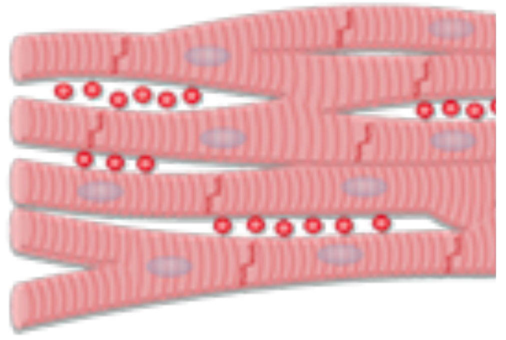
Heart model

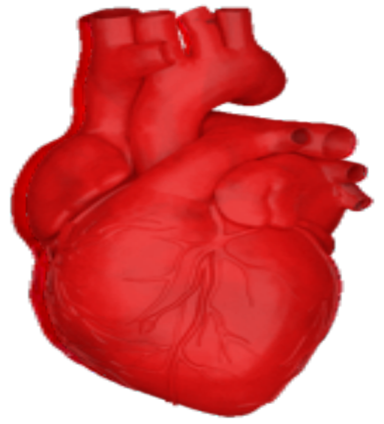


Computational cardiology: Personalized HeartBeats

Cell model | Tissue | Heart model







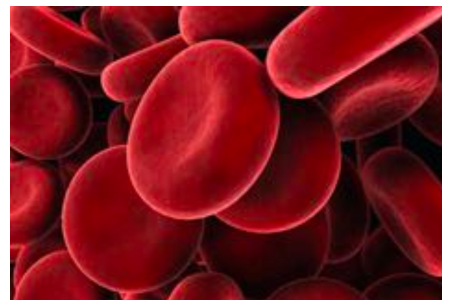
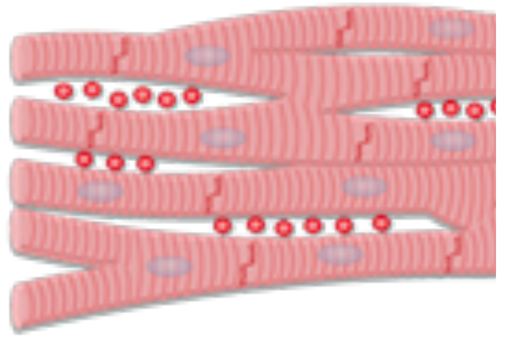
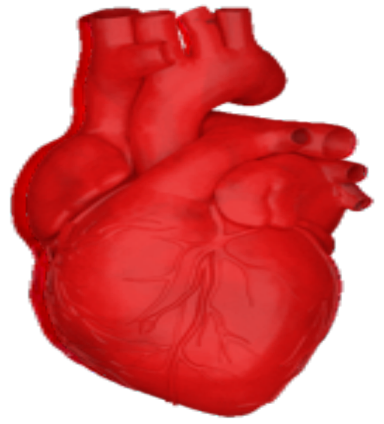
gif credits
alperdurmaz

- Dysregulation of calcium signaling in heart cells can cause lethal arrhythmias

[Augustin+ '16, Colman '19, Riabiz+ '21, Niederer+ '21]

Computational cardiology: Personalized HeartBeats

Cell model | Tissue | Heart model

gif credits
alperdurmaz

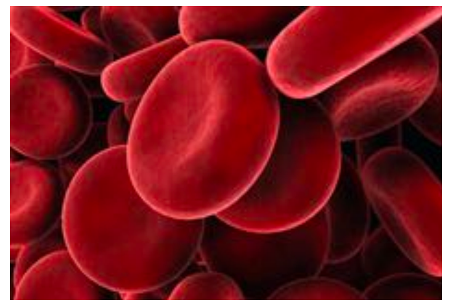
- Dysregulation of calcium signaling in heart cells can cause lethal arrhythmias

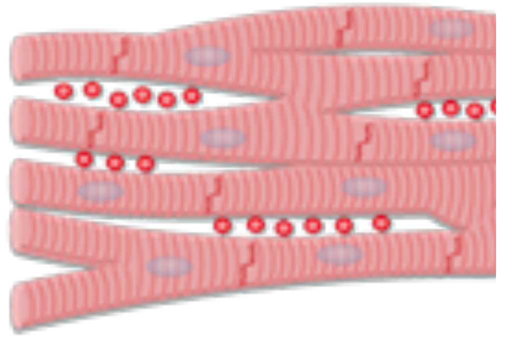
- Task: **Simulate** multi-scale **digital twin** models of heart for **personalized predictions** of dysregulation's effect on a patient's heartbeat

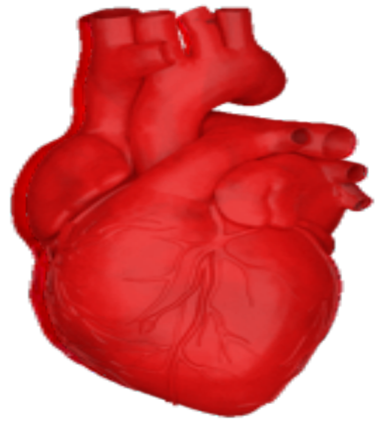
[Augustin+ '16, Colman '19, Riabiz+ '21, Niederer+ '21]

Computational cardiology: Personalized HeartBeats

Cell model — Tissue — Heart model







gif credits
alperdurmaz

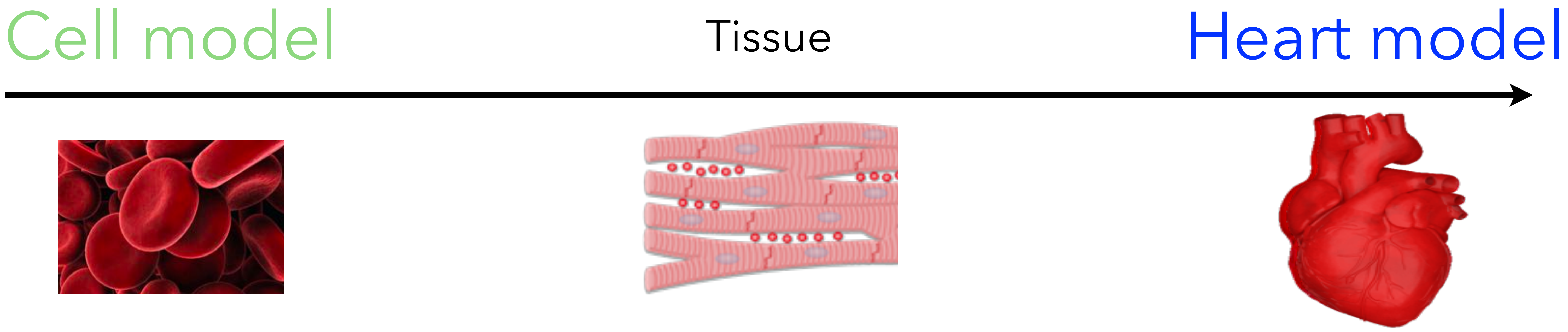
- Dysregulation of calcium signaling in heart cells can cause lethal arrhythmias

- Task: **Simulate** multi-scale **digital twin** models of heart for **personalized predictions** of dysregulation's effect on a patient's heartbeat

1. **Estimate** cell-model parameters with uncertainty quantification with single cell measurements via Bayesian inference and posterior **sampling**

[Augustin+ '16, Colman '19, Riabiz+ '21, Niederer+ '21]

Computational cardiology: Personalized HeartBeats



gif credits
alperdurmaz

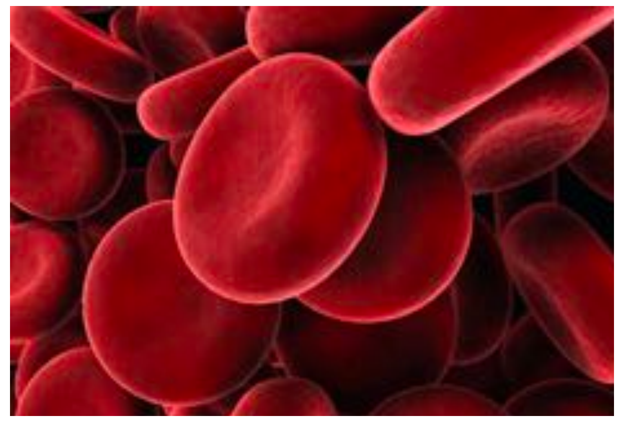
- Dysregulation of calcium signaling in heart cells can cause lethal arrhythmias
- Task: **Simulate** multi-scale **digital twin** models of heart for **personalized predictions** of dysregulation's effect on a patient's heartbeat

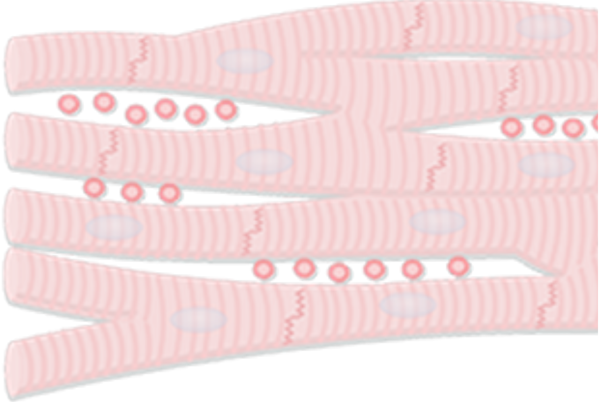
1. **Estimate** cell-model parameters with uncertainty quantification with single cell measurements via Bayesian inference and posterior **sampling**
2. **Propagate** cell-model **uncertainty** to whole-heart model via simulations and Monte Carlo **integration**

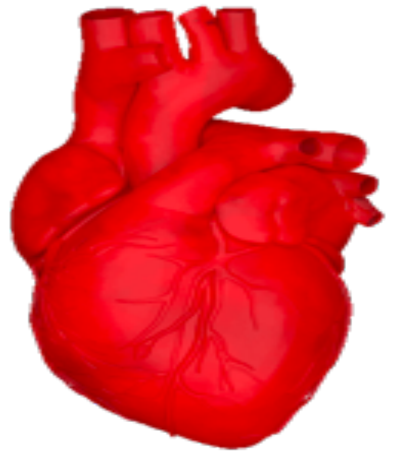
[Augustin+ '16, Colman '19, Riabiz+ '21, Niederer+ '21]

Impact of calcium signaling dysregulation on heartbeat—Two-stage inferential pipeline

Cell model X



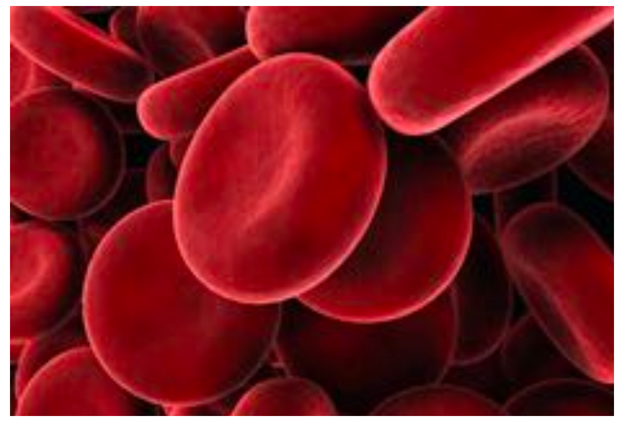


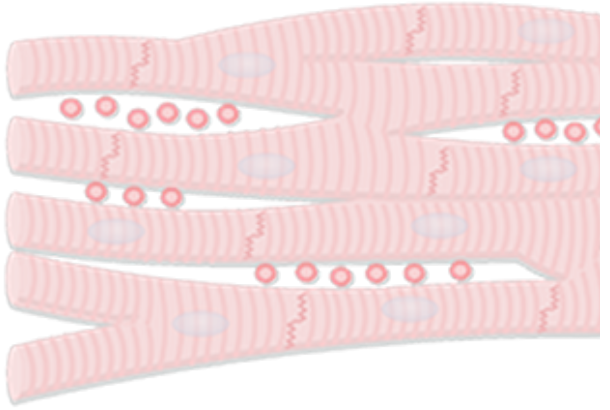


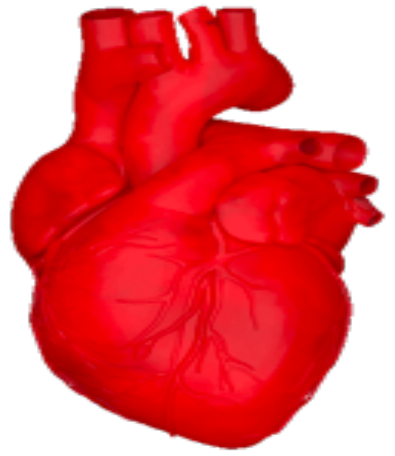
Heart model f

Impact of calcium signaling dysregulation on heartbeat—Two-stage inferential pipeline

Cell model X







Heart model f

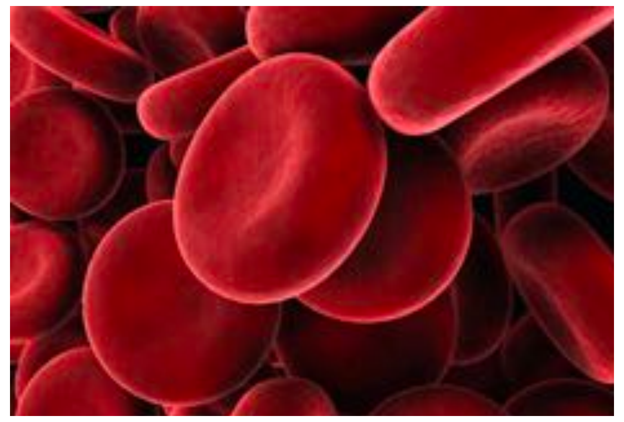
1. Random sampling via MCMC

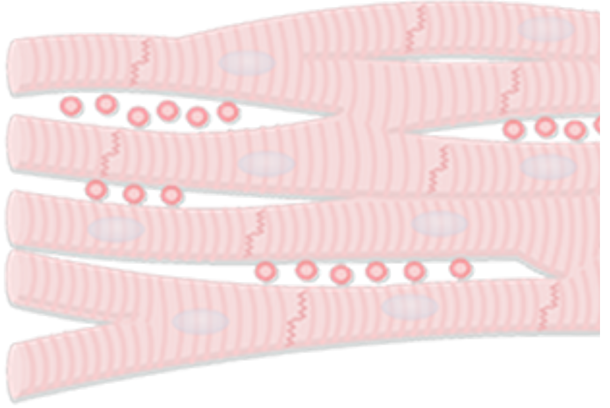
$$X_1, \ldots, X_T \sim \mathbb{P}^\star$$

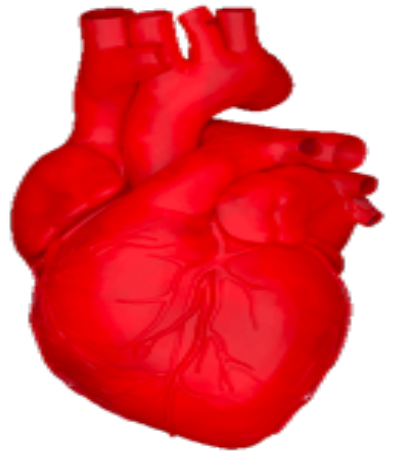
(posterior in $\mathbb{R}^{38}$)

Impact of calcium signaling dysregulation on heartbeat—Two-stage inferential pipeline

Cell model X







Heart model f

1. Random sampling via MCMC

$$X_1, \ldots, X_T \sim \mathbb{P}^\star$$

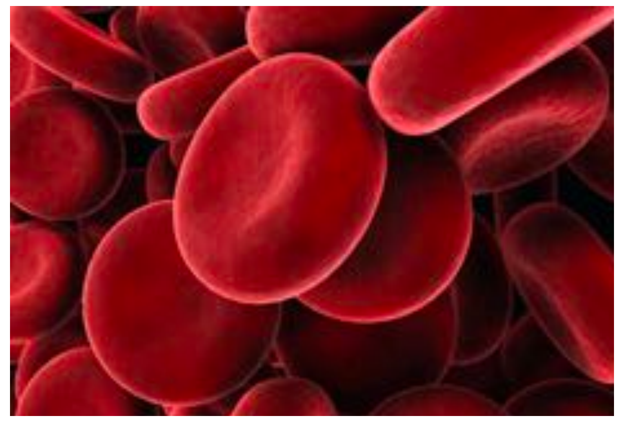
(posterior in $\mathbb{R}^{38}$)

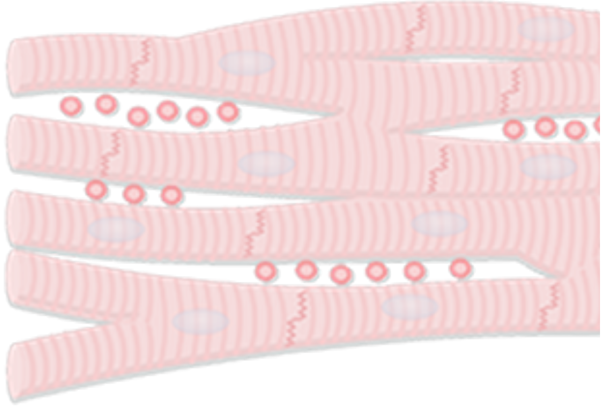
2. Uncertainty propagation via Monte Carlo integration (mean, variance,..)

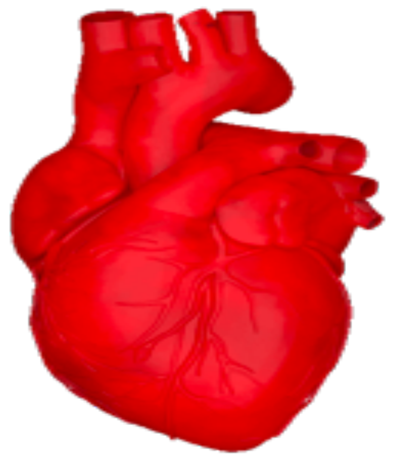
$$\mathbb{P}^\star f \triangleq \int f(X) d\mathbb{P}^\star(X) \approx \frac{1}{T} \sum_{i=1}^{T} f(X_i)$$

Standard tasks but computationally challenging…

Cell model X







Heart model f

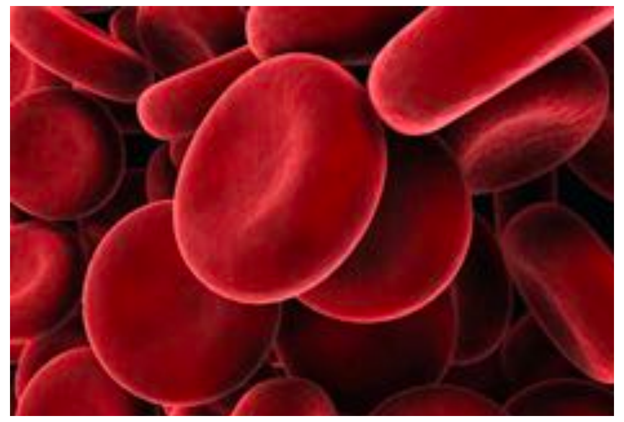
1. Random sampling via MCMC
   $X_1, \ldots, X_T \sim \mathbb{P}^\star$
   (posterior in $\mathbb{R}^{38}$)

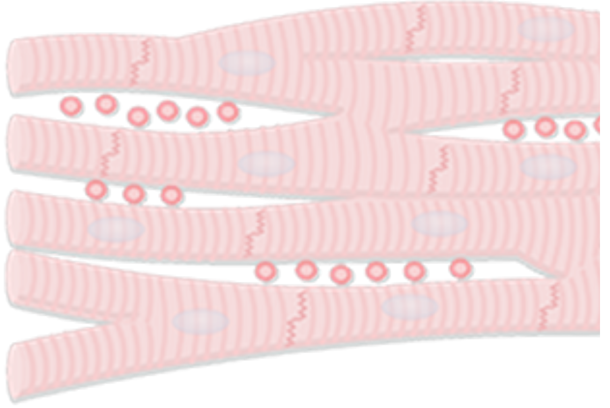
2. Uncertainty propagation via Monte Carlo integration (mean, variance,..)
   $$\mathbb{P}^\star f \triangleq \int f(X) d\mathbb{P}^\star(X) \approx \frac{1}{T}\sum_{i=1}^{T} f(X_i)$$

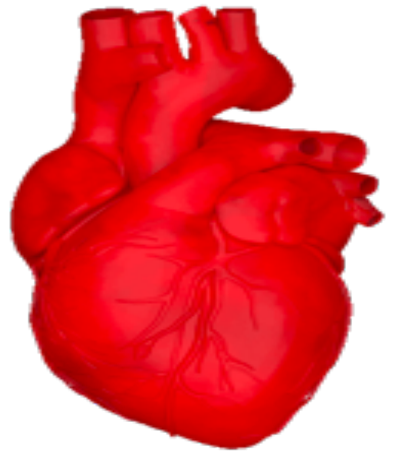
- $T = 10^6$ to explore $\mathbb{P}^\star$ well

Standard tasks but computationally challenging...

Cell model X







Heart model f

1. Random sampling via MCMC

$$X_1, \dots, X_T \sim \mathbb{P}^\star$$

(posterior in $\mathbb{R}^{38}$)

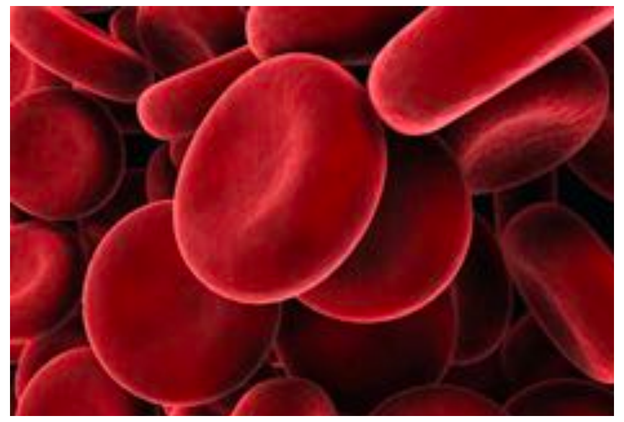
2. Uncertainty propagation via Monte Carlo integration (mean, variance,..)

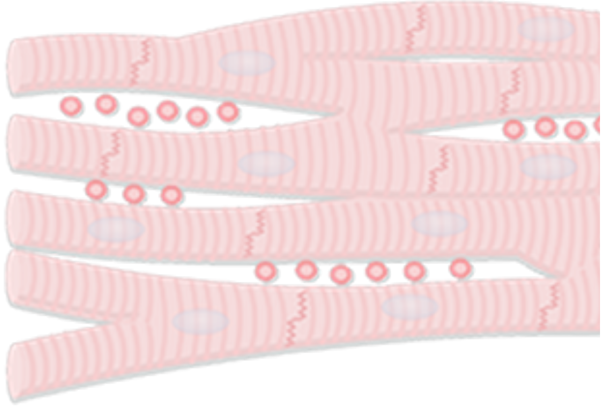
$$\mathbb{P}^\star f \triangleq \int f(X) d\mathbb{P}^\star(X) \approx \frac{1}{T}\sum_{i=1}^{T} f(X_i)$$

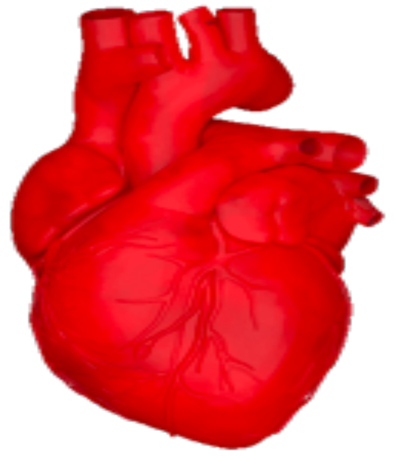
- $T = 10^6$ to explore $\mathbb{P}^\star$ well
- Time to run **MCMC** **~ 2 CPU weeks**

Standard tasks but computationally challenging...

Cell model X







Heart model f

1. Random sampling via MCMC

$$X_1, \ldots, X_T \sim \mathbb{P}^\star$$

(posterior in $\mathbb{R}^{38}$)

- $T = 10^6$ to explore $\mathbb{P}^\star$ well
- Time to run **MCMC**
  ~ 2 CPU weeks

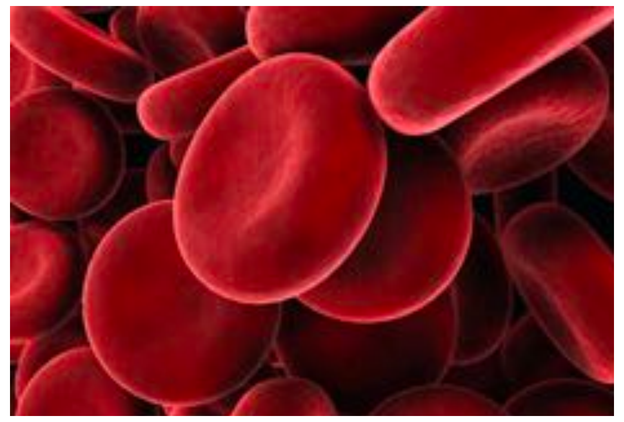
2. Uncertainty propagation via Monte Carlo integration (mean, variance,..)

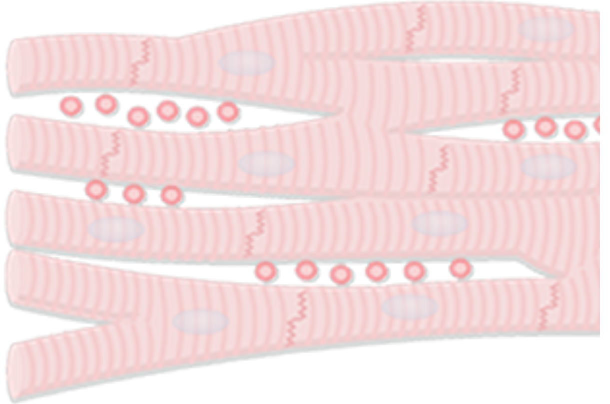
$$\mathbb{P}^\star f \triangleq \int f(X) d\mathbb{P}^\star(X) \approx \frac{1}{T} \sum_{i=1}^{T} f(X_i)$$

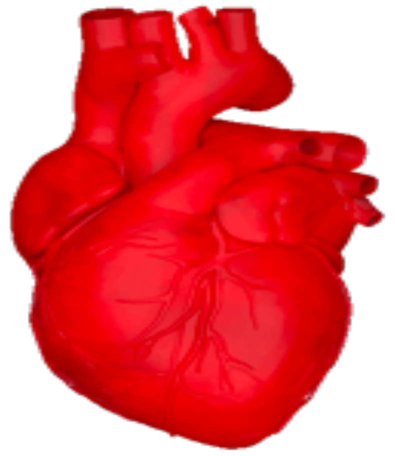
- Single f simulation~ 4 CPU weeks

Standard tasks but computationally challenging...

Cell model X







Heart model f

1. Random sampling via MCMC

$X_1, \ldots, X_T \sim \mathbb{P}^\star$

(posterior in $\mathbb{R}^{38}$)

2. Uncertainty propagation via Monte Carlo integration (mean, variance,..)

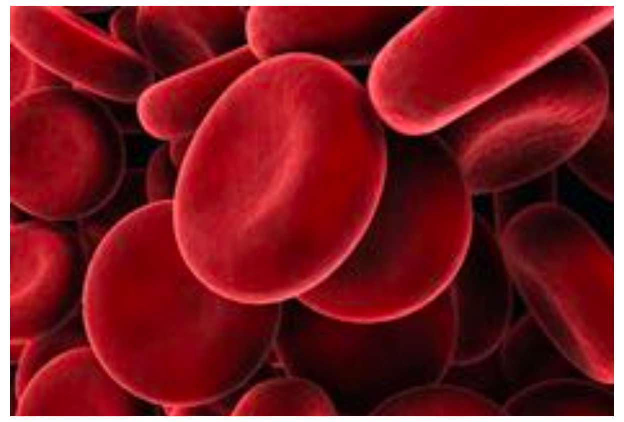
$$\mathbb{P}^\star f \triangleq \int f(X) d\mathbb{P}^\star(X) \approx \frac{1}{T}\sum_{i=1}^{T} f(X_i)$$

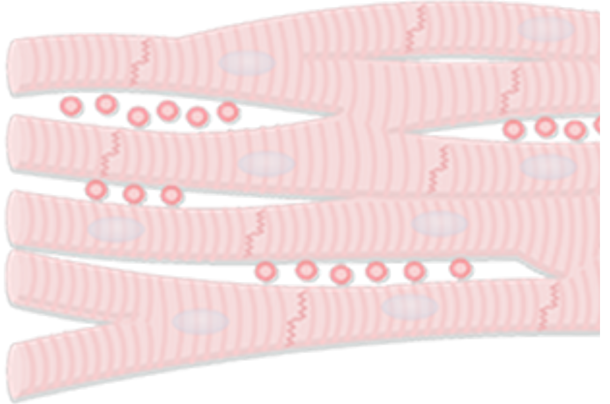
- $T = 10^6$ to explore $\mathbb{P}^\star$ well
- Time to run **MCMC** **~ 2 CPU weeks**

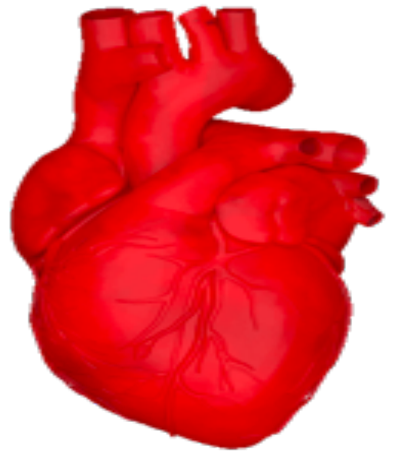
- Single f simulation ~ 4 CPU weeks
- Time to compute **sample mean** **~ 4 Million CPU weeks**

Standard tasks but computationally challenging...

Cell model X







Heart model f

1. Random sampling via MCMC

$$X_1, \ldots, X_T \sim \mathbb{P}^\star$$

(posterior in $\mathbb{R}^{38}$)

- $T = 10^6$ to explore $\mathbb{P}^\star$ well
- Time to run **MCMC ~ 2 CPU weeks**
- **How to make MCMC computationally faster?**

??

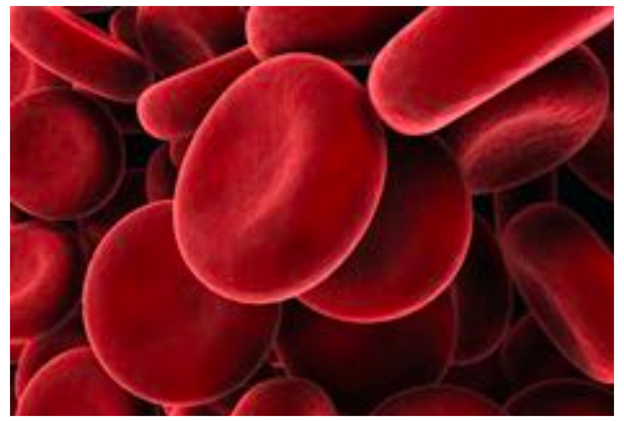
2. Uncertainty propagation via Monte Carlo integration (mean, variance,..)

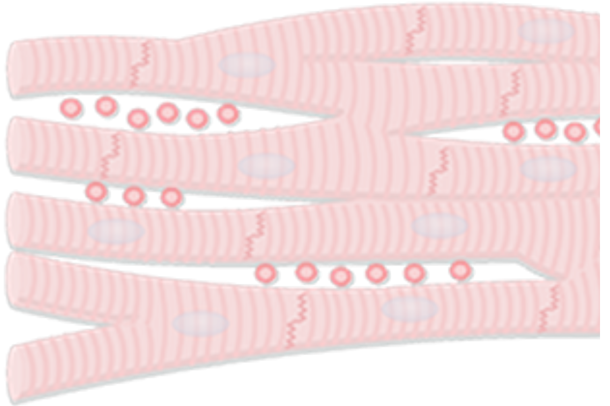
$$\mathbb{P}^\star f \triangleq \int f(X) d\mathbb{P}^\star(X) \approx \frac{1}{T}\sum_{i=1}^{T} f(X_i)$$

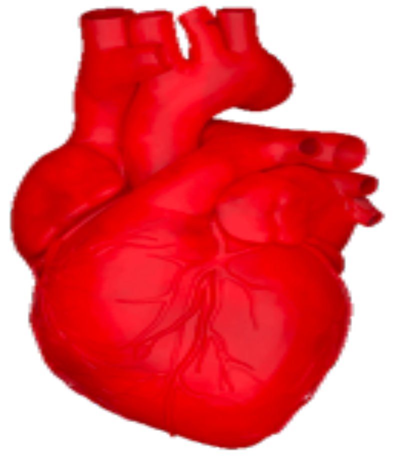
- Single f simulation ~ 4 CPU weeks
- Time to compute **sample mean ~ 4 Million CPU weeks**
- **How to make integration computationally feasible?**

Part 2 overview: Computationally-efficient integration for high-dimensional models

Cell model X







Heart model f

1. Random sampling via MCMC
$X_1, \ldots, X_T \sim \mathbb{P}^\star$
(posterior in $\mathbb{R}^{38}$)

2. Uncertainty propagation via Monte Carlo integration (mean, variance,..)

$$\mathbb{P}^\star f \triangleq \int f(X) d\mathbb{P}^\star(X) \approx \frac{1}{T}\sum_{i=1}^{T} f(X_i)$$

- $T = 10^6$ to explore $\mathbb{P}^\star$ well
- Time to run **MCMC** **~ 2 CPU weeks**
- **How to make MCMC computationally faster?**

??

This talk

- Single f simulation~ 4 CPU weeks
- Time to compute **sample mean** **~ 4 Million CPU weeks**
- **How to make integration computationally feasible?**

Efficient integration via distribution compression

Efficient integration via distribution compression

T IID or MCMC points

$$X_1, \ldots, X_T$$

$$\mathbb{P}_T f \triangleq \frac{\Sigma_{i=1}^T f(X_i)}{T}$$

Efficient integration via distribution compression

T IID or MCMC points

$$X_1, \ldots, X_T$$

$$\mathbb{P}_T f \triangleq \frac{\Sigma_{i=1}^{T} f(X_i)}{T}$$

s output points (coreset)

Compress $\longrightarrow$

$$X'_1, \ldots, X'_s$$

$$\mathbb{P}_{out} f \triangleq \frac{\Sigma_{i=1}^{s} f(X'_i)}{s}$$

s (fewer) function evaluations

Efficient integration via distribution compression

T IID or MCMC points

s output points (coreset)

$X_1, \dots, X_T$ $\xrightarrow{\text{Compress}}$ $X'_1, \dots, X'_s$

$$\mathbb{P}_T f \triangleq \frac{\Sigma_{i=1}^{T} f(X_i)}{T}$$

$$\mathbb{P}_{out} f \triangleq \frac{\Sigma_{i=1}^{s} f(X'_i)}{s}$$

s (fewer) function evaluations

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2})$$

Efficient integration via distribution compression

T IID or MCMC points

s output points (coreset)

$$X_1, \ldots, X_T \xrightarrow{\text{Compress}} X'_1, \ldots, X'_s$$

$$\mathbb{P}_T f \triangleq \frac{\Sigma_{i=1}^T f(X_i)}{T}$$

$$\mathbb{P}_{out} f \triangleq \frac{\Sigma_{i=1}^s f(X'_i)}{s}$$

s (fewer) function evaluations

Standard thinning

(take every T/s-th point)

or iid thinning/

uniform **sub-sampling**

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2})$$

$$|\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(s^{-1/2})$$

$$|\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

when $s = T^{1/2}$

a million → a thousand

T IID or MCMC points

a million → a thousand

$T^{1/2}$ output points

Standard thinning

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2}) \longrightarrow |\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

What is the best error we can hope for?

T IID or MCMC points   a million → a thousand $T^{1/2}$ output points

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2}) \xrightarrow{\text{Standard thinning}} |\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

What is the best error we can hope for?

T IID or MCMC points   a million → a thousand $T^{1/2}$ output points

Standard thinning

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2}) \xrightarrow{\text{Standard thinning}} |\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

$\Omega(T^{-1/2})$ minimax **lower bound**

- If output = $T^{1/2}$ points
- If input = T IID points (any estimator)

[Tolstikhin+ '17, Philips+ '20]

Prior strategies for efficient integration

T IID or MCMC points

a million → a thousand

$T^{1/2}$ output points

Standard thinning

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2}) \xrightarrow{\text{Standard thinning}} |\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

$\Omega(T^{-1/2})$ minimax lower bound

Prior strategies for efficient integration

T IID or MCMC points   a million → a thousand $T^{1/2}$ output points

Standard thinning

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2}) \longrightarrow |\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

Special $\mathbb{P}^\star$
- Uniform on $[0,1]^d$
- Bounded support & special function class

- $o(T^{-1/4})$ error guarantee: Quasi Monte Carlo, Bayesian quadrature, determinantal point processes, Haar thinning
[O'Hagan '91, Hickernell '98, Novak+'10, Liu+ '18, Karvonen+'18, **Dwivedi+**'19, Belhadji+ '20]

$\Omega(T^{-1/2})$ minimax lower bound

Prior strategies for efficient integration

T IID or MCMC points   a million → a thousand $T^{1/2}$ output points

Standard thinning

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2}) \longrightarrow |\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

Special $\mathbb{P}^\star$
- Uniform on $[0,1]^d$
- Bounded support & special function class

$o(T^{-1/4})$ error guarantee:
Quasi Monte Carlo, Bayesian quadrature, determinantal point processes, Haar thinning
[O'Hagan '91, Hickernell '98, Novak+'10, Liu+ '18, Karvonen+'18, **Dwivedi+**'19, Belhadji+ '20]

Generic $\mathbb{P}^\star$ & rich function class

$\tilde{O}(T^{-1/4})$ error guarantee:
Kernel herding, greedy sign selection, Stein points MCMC, support points, supersampling
[Chen+ '10, Lacoste+ '15, Paige+ '16, Tolstikhin+ '17, Mak+ '18, Chen '19, Karnin '19]

$\Omega(T^{-1/2})$ minimax lower bound

T IID or MCMC points

a million → a thousand

$T^{1/2}$ output points

Standard thinning

$$|\mathbb{P}^{\star}f - \mathbb{P}_T f| = \Theta(T^{-1/2}) \longrightarrow |\mathbb{P}^{\star}f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

$\Omega(T^{-1/2})$ minimax lower bound

A new practical & provably near-optimal procedure

T IID or MCMC points   a million → a thousand $T^{1/2}$ output points

Standard thinning

$$|\mathbb{P}^\star f - \mathbb{P}_T f| = \Theta(T^{-1/2}) \longrightarrow |\mathbb{P}^\star f - \mathbb{P}_{out} f| = \Theta(T^{-1/4})$$

Kernel thinning

Dwivedi and Mackey '21, '22

$$|\mathbb{P}^\star f - \mathbb{P}_{out} f| = \tilde{O}(T^{-1/2})$$

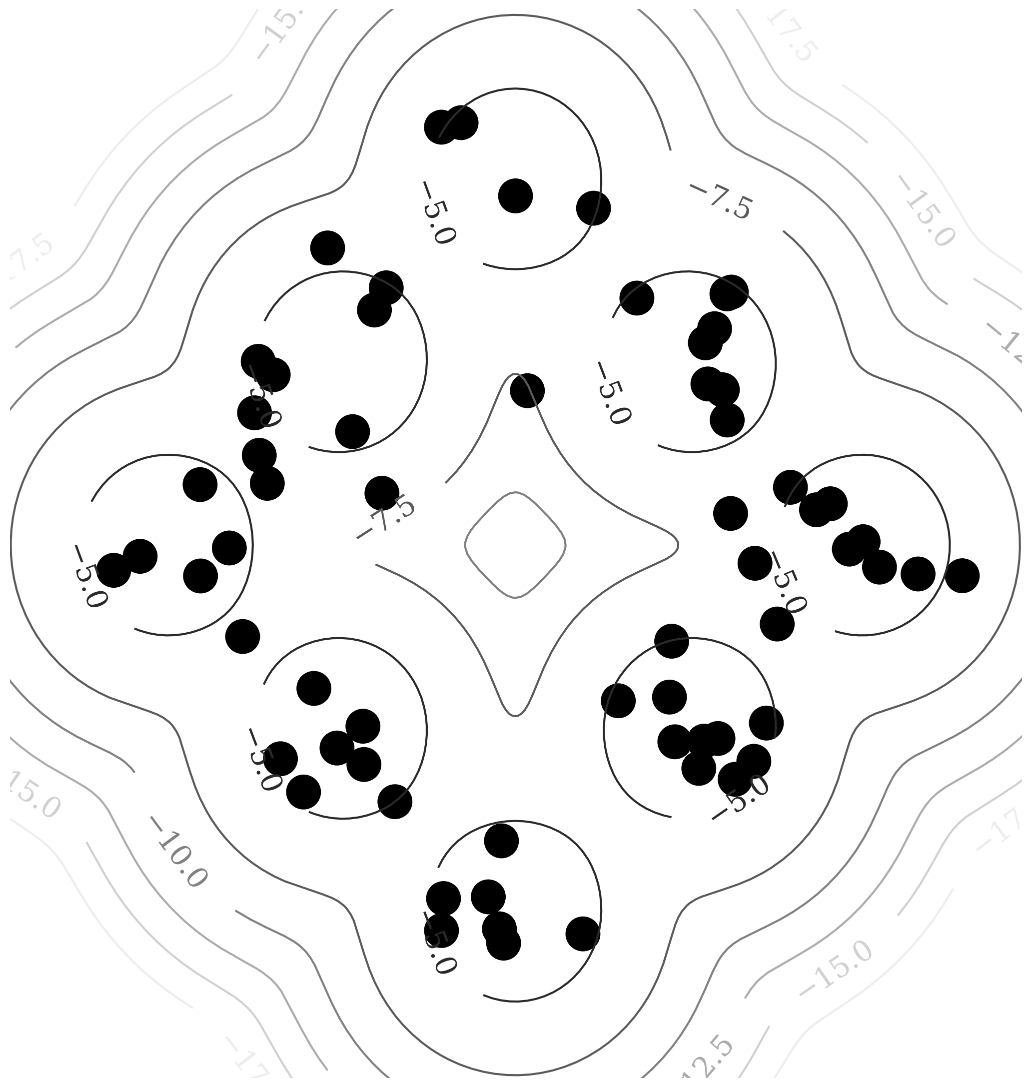
✓ for generic $\mathbb{P}^\star$ on generic domains
✓ for rich function classes

$\Omega(T^{-1/2})$ minimax lower bound

Visual comparison on $\mathbb{P}^\star$ = 8 mixture of Gaussian

64 iid input points

8 output points

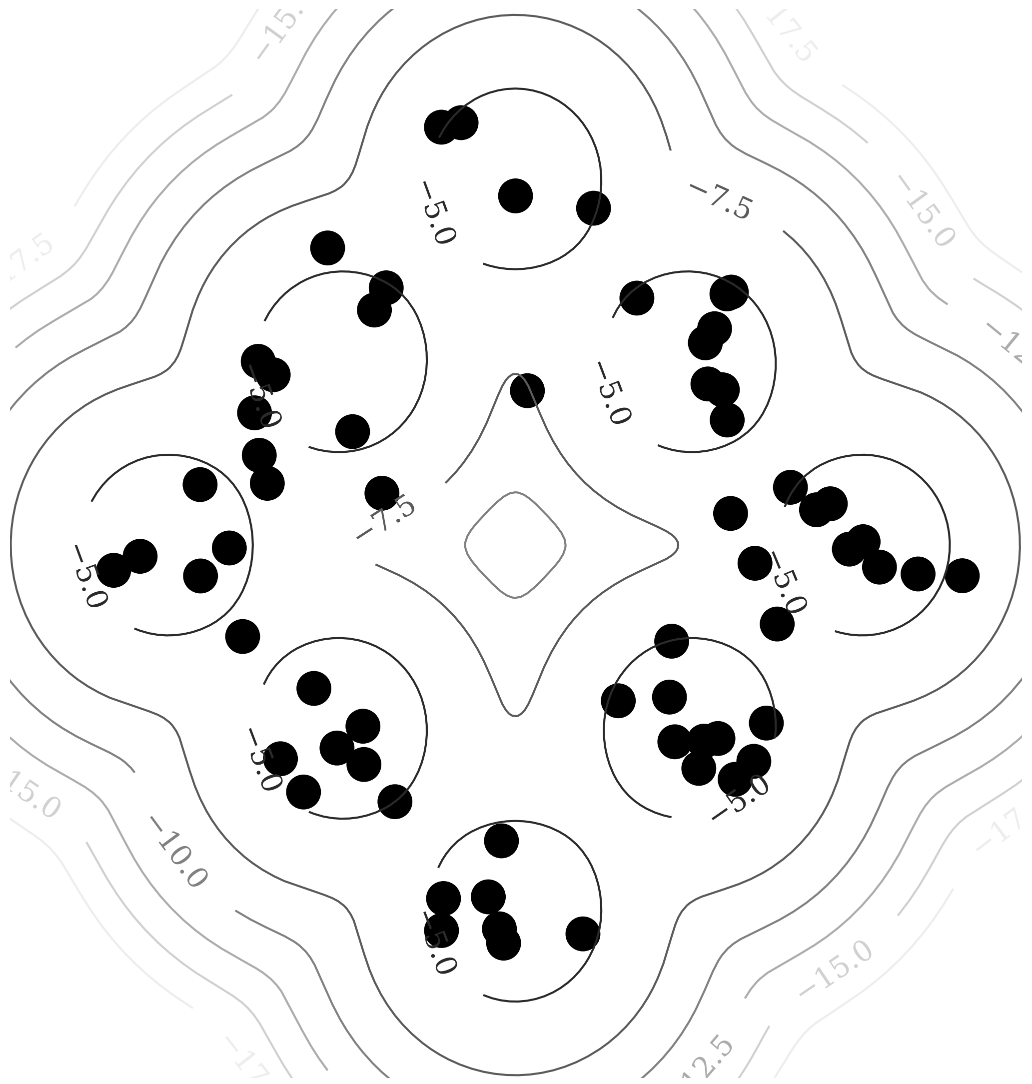


Standard thinning

Kernel thinning

Visual comparison on $\mathbb{P}^\star$ = 8 mixture of Gaussian

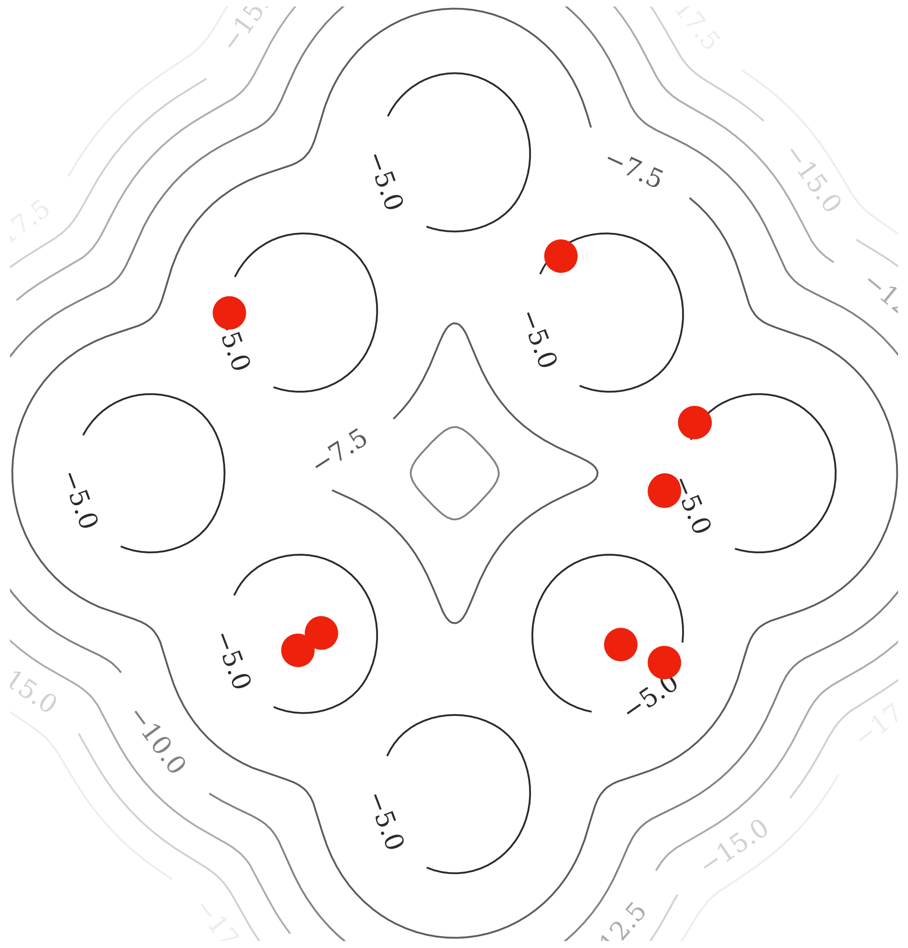
64 iid input points

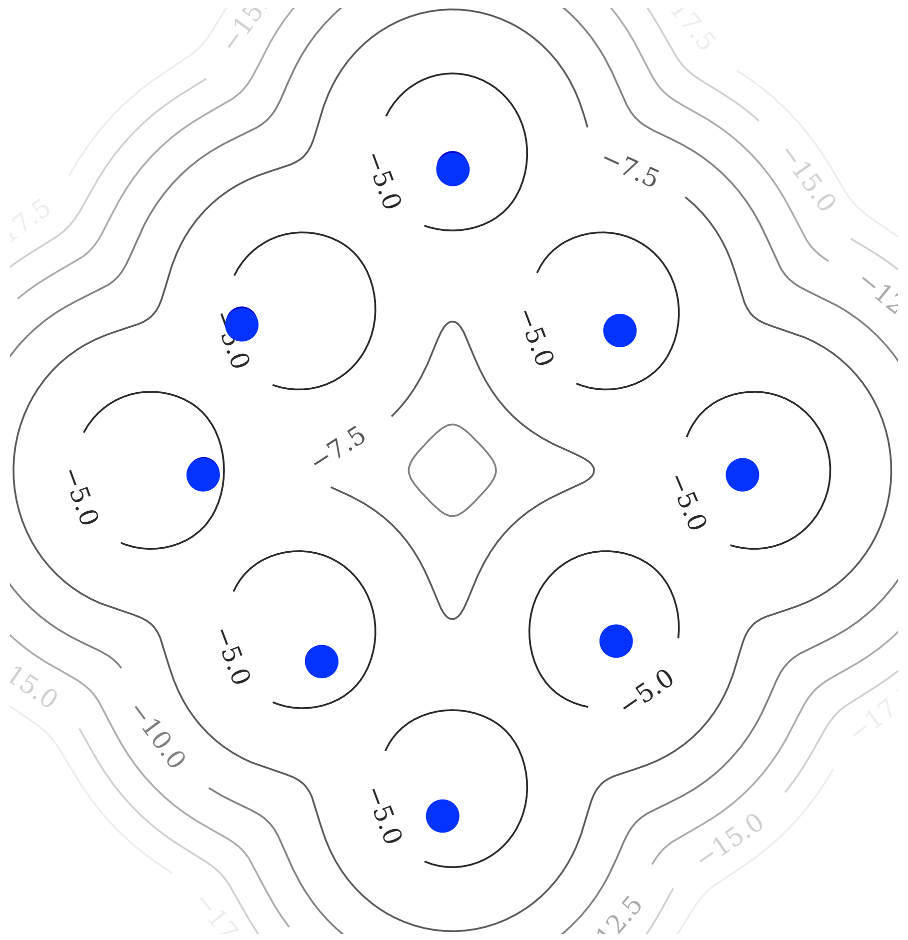


Standard thinning

Kernel thinning

8 output points





Quantitative measure: Worst-case error over a rich class

Quantitative measure: Worst-case error over a rich class

Namely, over the unit ball of a reproducing kernel Hilbert space (RKHS)

$$\sup_{\|f\|_{\mathbf{k}} \leq 1} |\mathbb{P}^\star f - \mathbb{P}_{out} f|$$

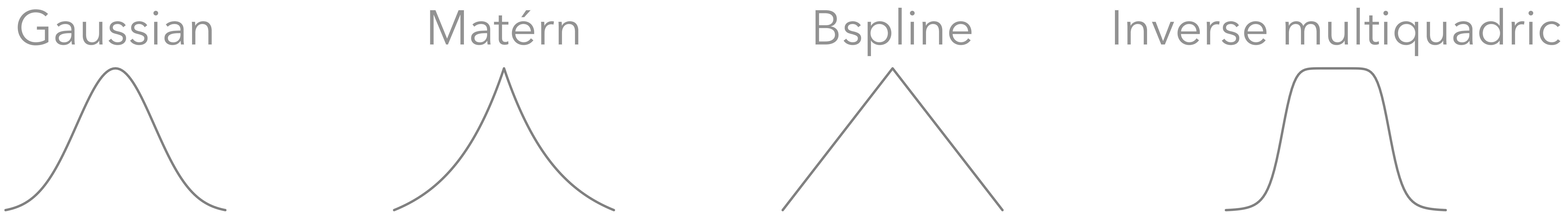
Quantitative measure: Worst-case error over a rich class

Namely, over the unit ball of a reproducing kernel Hilbert space (RKHS)

$$\sup_{\|f\|_{\mathbf{k}} \leq 1} |\mathbb{P}^{\star} f - \mathbb{P}_{out} f|$$

- Parameterized by a reproducing kernel $\mathbf{k}$

  any symmetric ($\mathbf{k}(x, y) = \mathbf{k}(y, x)$) and positive semidefinite function



Quantitative measure: Worst-case error over a rich class

Namely, over the unit ball of a reproducing kernel Hilbert space (RKHS)

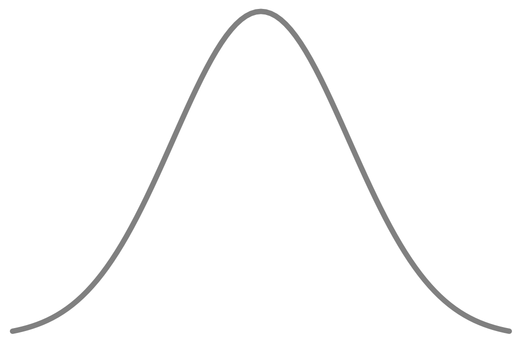
$$\sup_{\|f\|_{\mathbf{k}} \leq 1} |\mathbb{P}^{\star} f - \mathbb{P}_{out} f|$$

- Parameterized by a reproducing kernel $\mathbf{k}$

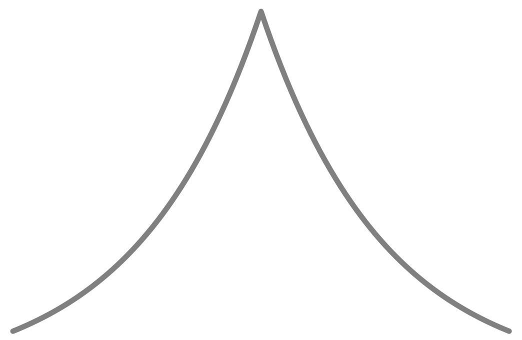
  any symmetric ($\mathbf{k}(x, y) = \mathbf{k}(y, x)$) and positive semidefinite function

- **Metrizes convergence in distribution** for popular infinite-dimensional $\mathbf{k}$

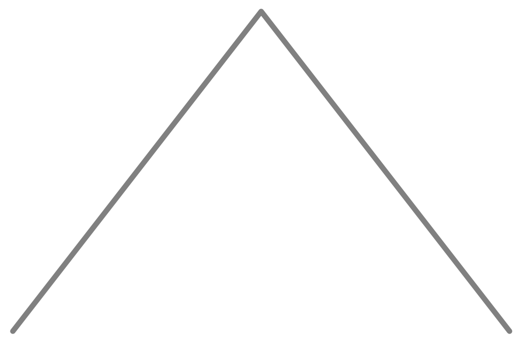
Gaussian



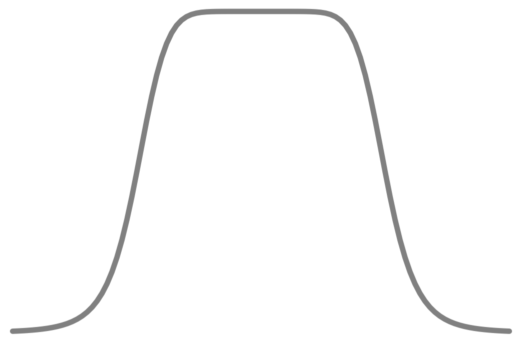
Matérn



Bspline



Inverse multiquadric



Main result: A high probability bound for generic $\mathbb{P}^\star$ and $\mathbf{k}$

Main result: A high probability bound for generic $\mathbb{P}^\star$ and $\mathbf{k}$

Informal theorem: [**Dwivedi** and Mackey'21, '22 and **Dwivedi**-Shetty-Mackey '22]
Kernel thinning uses $O(T \log^3 T)$ **kernel evaluations** to output $T^{1/2}$ points, that with high probability satisfy

Main result: A high probability bound for generic $\mathbb{P}^\star$ and $\mathbf{k}$

Informal theorem: [**Dwivedi** and Mackey'21, '22 and **Dwivedi**-Shetty-Mackey '22]
Kernel thinning uses $O(T\log^3 T)$ **kernel evaluations** to output $T^{1/2}$ points, that with high probability satisfy

- $|\mathbb{P}^\star f - \mathbb{P}_{out} f| \lesssim \sqrt{\frac{\log T}{T}} \cdot \|f\|_{\mathbf{k}} \sqrt{\|\mathbf{k}\|_\infty}$ for a fixed f in the RKHS of $\mathbf{k}$ (any kernel)

when $|\mathbb{P}^\star f - \mathbb{P}_T f| \lesssim T^{-1/2}$

- A near-quadratic gain over $T^{-1/4}$ standard thinning error

Main result: A high probability bound for generic $\mathbb{P}^\star$ and $\mathbf{k}$

Informal theorem: [**Dwivedi** and Mackey'21, '22 and **Dwivedi**-Shetty-Mackey '22]
Kernel thinning uses $O(T\log^3 T)$ **kernel evaluations** to output $T^{1/2}$ points, that with high probability satisfy

- $|\mathbb{P}^\star f - \mathbb{P}_{out} f| \lesssim \sqrt{\frac{\log T}{T}} \cdot \|f\|_{\mathbf{k}} \sqrt{\|\mathbf{k}\|_\infty}$ for a fixed f in the RKHS of $\mathbf{k}$ (any kernel)

  when $|\mathbb{P}^\star f - \mathbb{P}_T f| \lesssim T^{-1/2}$

- $\sup_{\|f\|_{\mathbf{k}} \leq 1} |\mathbb{P}^\star f - \mathbb{P}_{out} f| \lesssim \sqrt{\frac{\log^{d/2+1} T}{T}}$ Sub-gaussian $\mathbb{P}^\star$ and $\mathbf{k}$ on $\mathbb{R}^d$ (Gaussian)

  $\lesssim \sqrt{\frac{\log^{d+1} T}{T}}$ Sub-exponential $\mathbb{P}^\star$ and $\mathbf{k}$ on $\mathbb{R}^d$ (Matérn)

- A near-quadratic gain over $T^{-1/4}$ standard thinning error
- Matches minimax lower bounds $T^{-1/2}$ up to log factors

Kernel thinning

points $X_1, X_2, \ldots, X_T$ $\rightarrow$ **Kernel** $\rightarrow$ functions in RKHS $v_1, v_2, \ldots, v_T$

Kernel thinning ≡ Recursive halving via kernel evaluations

points $X_1, X_2, \ldots, X_T$ → **Kernel** → functions in RKHS $v_1, v_2, \ldots, v_T$

Input T points → Kernel halving → $T/2$ points → $O(\log T)$ Kernel halving rounds → **Output** $\sqrt{T}$ points + some refinement

Kernel halving

points $X_1, X_2, \ldots, X_T$ → Kernel → functions in RKHS $v_1, v_2, \ldots, v_T$

$v_1, v_2, \ldots, v_T$ → Kernel halving → $v'_1, v'_2, \ldots, v'_{T/2}$

$$\left| \frac{\Sigma_{i=1}^{T} v_i}{T} - \frac{\Sigma_{i=1}^{T/2} v'_i}{T/2} \right| = \text{small}$$

Kernel halving $\equiv$ Discrepancy minimization problem

points $X_1, X_2, \ldots, X_T$ → Kernel → functions in RKHS $v_1, v_2, \ldots, v_T$

$v_1, v_2, \ldots, v_T$ → Kernel halving → $v'_1, v'_2, \ldots, v'_{T/2}$

$$\left| \frac{\Sigma_{i=1}^{T} v_i}{T} - \frac{\Sigma_{i=1}^{T/2} v'_i}{T/2} \right| = \text{small}$$

$\equiv$

Assign $\varepsilon_i \in \{-1,1\}$ to v_i
such that $|\Sigma_{i=1}^{T} \varepsilon_i v_i|$ is small
& output points with $\varepsilon_i = -1$

Kernel halving ≡ Discrepancy minimization problem

points $X_1, X_2, \ldots, X_T$ → Kernel → functions in RKHS $v_1, v_2, \ldots, v_T$

$v_1, v_2, \ldots, v_T$ → Kernel halving → $v'_1, v'_2, \ldots, v'_{T/2}$

$$\left| \frac{\Sigma_{i=1}^{T} v_i}{T} - \frac{\Sigma_{i=1}^{T/2} v'_i}{T/2} \right| = \text{small}$$

≡

Assign $\varepsilon_i \in \{-1,1\}$ to v_i
such that $|\Sigma_{i=1}^{T} \varepsilon_i v_i|$ is small
& output points with $\varepsilon_i = -1$

$$\sum_{i=1}^{T} \varepsilon_i v_i = \sum_{\varepsilon_i=+1} v_i - \sum_{\varepsilon_i=-1} v_i$$

Kernel halving ≡ Discrepancy minimization problem

points $X_1, X_2, \ldots, X_T$ → Kernel → functions in RKHS $v_1, v_2, \ldots, v_T$

Assign $\varepsilon_i \in \{-1,1\}$ to v_i
such that $|\Sigma_{i=1}^{T} \varepsilon_i v_i|$ is small
& output points with $\varepsilon_i = -1$

$v_1, v_2, \ldots, v_T$ → Kernel halving → $v'_1, v'_2, \ldots, v'_{T/2}$

$$\left| \frac{\Sigma_{i=1}^{T} v_i}{T} - \frac{\Sigma_{i=1}^{T/2} v'_i}{T/2} \right| = \text{small}$$

$\equiv$

$$\sum_{i=1}^{T} \varepsilon_i v_i = \sum_{\varepsilon_i=+1} v_i - \sum_{\varepsilon_i=-1} v_i$$

$$= \sum_{i=1}^{T} v_i - 2 \sum_{\varepsilon_i=-1} v_i$$

Kernel halving ≡ Discrepancy minimization problem

points

$X_1, X_2, \ldots, X_T$ → Kernel → functions in RKHS

$v_1, v_2, \ldots, v_T$

$v_1, v_2, \ldots, v_T$ → Kernel halving → $v'_1, v'_2, \ldots, v'_{T/2}$

Assign $\varepsilon_i \in \{-1,1\}$ to v_i

such that $|\sum_{i=1}^T \varepsilon_i v_i|$ is small

& output points with $\varepsilon_i = -1$

$$\left| \frac{\sum_{i=1}^T v_i}{T} - \frac{\sum_{i=1}^{T/2} v'_i}{T/2} \right| = \text{small}$$

$$\equiv$$

$$\sum_{i=1}^{T} \varepsilon_i v_i = \sum_{\varepsilon_i=+1} v_i - \sum_{\varepsilon_i=-1} v_i$$

$$= \sum_{i=1}^{T} v_i - 2 \sum_{\varepsilon_i=-1} v_i$$

$$\frac{\sum_{i=1}^T \varepsilon_i v_i}{T} = \frac{\sum_{i=1}^T v_i}{T} - \frac{\sum_{\varepsilon_i=-1} v_i}{T/2}$$

Kernel halving ≡ Discrepancy minimization problem

points

$X_1, X_2, \ldots, X_T$ → Kernel → functions in RKHS $v_1, v_2, \ldots, v_T$

$v_1, v_2, \ldots, v_T$ → Kernel halving → $v'_1, v'_2, \ldots, v'_{T/2}$

$$\left| \frac{\sum_{i=1}^{T} v_i}{T} - \frac{\sum_{i=1}^{T/2} v'_i}{T/2} \right| = \text{small}$$

≡

Assign $\varepsilon_i \in \{-1, 1\}$ to v_i

such that $|\sum_{i=1}^{T} \varepsilon_i v_i|$ is small

& output points with $\varepsilon_i = -1$

$$\sum_{i=1}^{T} \varepsilon_i v_i = \sum_{\varepsilon_i = +1} v_i - \sum_{\varepsilon_i = -1} v_i$$

$$= \sum_{i=1}^{T} v_i - 2 \sum_{\varepsilon_i = -1} v_i$$

$$\frac{\sum_{i=1}^{T} \varepsilon_i v_i}{T} = \frac{\sum_{i=1}^{T} v_i}{T} - \frac{\sum_{\varepsilon_i = -1} v_i}{T/2}$$

Error — Input mean — Halve output mean

KT intuition: IID vs correlated signs

$|\sum_{i=1}^{T} \varepsilon_i v_i|$ is small

KT intuition: IID vs correlated signs

$|\sum_{i=1}^{T} \varepsilon_i v_i|$ is small

$\varepsilon_i = \pm 1$ with equal probability

KT intuition: IID vs correlated signs

$|\Sigma_{i=1}^{T} \varepsilon_i v_i|$ is small

$\varepsilon_i = \pm 1$ with equal probability

$\sigma_T^2 = \sigma_{T-1}^2 + v_T^2$

$|\Sigma_{i=1}^{T} \varepsilon_i v_i| = O(\sigma_T) = O(T^{1/2})$

Standard thinning

KT intuition: IID vs correlated signs

$|\Sigma_{i=1}^{T} \varepsilon_i v_i|$ is small

$\varepsilon_i = \pm 1$ with equal probability

$\sigma_T^2 = \sigma_{T-1}^2 + v_T^2$

$|\Sigma_{i=1}^{T} \varepsilon_i v_i| = O(\sigma_T) = O(T^{1/2})$

Standard thinning

ε_i negatively correlated with $\Sigma_{j=1}^{i-1} \varepsilon_j v_j$

KT intuition: IID vs correlated signs

$|\Sigma_{i=1}^{T}\varepsilon_i v_i|$ is small

$\varepsilon_i = \pm 1$ with equal probability

$\sigma_T^2 = \sigma_{T-1}^2 + v_T^2$

$|\Sigma_{i=1}^{T}\varepsilon_i v_i| = O(\sigma_T) = O(T^{1/2})$

Standard thinning

ε_i negatively correlated with $\Sigma_{j=1}^{i-1}\varepsilon_j v_j$

$\sigma_T^2 \leq \beta\, \sigma_{T-1}^2 + v_T^2$ for $\beta < 1$

KT intuition: IID vs correlated signs

$|\Sigma_{i=1}^{T} \varepsilon_i v_i|$ is small

$\varepsilon_i = \pm 1$ with equal probability

$\sigma_T^2 = \sigma_{T-1}^2 + v_T^2$

$|\Sigma_{i=1}^{T} \varepsilon_i v_i| = O(\sigma_T) = O(T^{1/2})$

Standard thinning

ε_i negatively correlated with $\Sigma_{j=1}^{i-1} \varepsilon_j v_j$

$\sigma_T^2 \leq \beta\, \sigma_{T-1}^2 + v_T^2$ for $\beta < 1$

$|\Sigma_{i=1}^{T} \varepsilon_i v_i| = O(\sigma_T) = O(\sqrt{\log T})$

KT intuition: IID vs correlated signs

$|\Sigma_{i=1}^{T} \varepsilon_i v_i|$ is small

$\varepsilon_i = \pm 1$ with equal probability

$\sigma_T^2 = \sigma_{T-1}^2 + v_T^2$

$|\Sigma_{i=1}^{T} \varepsilon_i v_i| = O(\sigma_T) = O(T^{1/2})$

Standard thinning

ε_i negatively correlated with $\Sigma_{j=1}^{i-1} \varepsilon_j v_j$

$\sigma_T^2 \leq \beta\, \sigma_{T-1}^2 + v_T^2$ for $\beta < 1$

$|\Sigma_{i=1}^{T} \varepsilon_i v_i| = O(\sigma_T) = O(\sqrt{\log T})$

Kernel thinning

KT intuition: IID vs correlated signs

$|\Sigma_{i=1}^{T}\varepsilon_i v_i|$ is small

$\varepsilon_i = \pm 1$ with equal probability

$\sigma_T^2 = \sigma_{T-1}^2 + v_T^2$

$|\Sigma_{i=1}^{T}\varepsilon_i v_i| = O(\sigma_T) = O(T^{1/2})$

Standard thinning

ε_i negatively correlated with $\Sigma_{j=1}^{i-1}\varepsilon_j v_j$

$\sigma_T^2 \leq \beta\, \sigma_{T-1}^2 + v_T^2$ for $\beta < 1$

$|\Sigma_{i=1}^{T}\varepsilon_i v_i| = O(\sigma_T) = O(\sqrt{\log T})$

Kernel thinning

Discrepancy minimization

[… Spencer '77, Banaszczyk '98, '12, Eldan+ '18, …
Bansal+ '16, '18, '19, '20, **Dwivedi**+ '19, Alweiss+ '21, …]

Is KT better practically? Gaussian $\mathbb{P}^\star$ in $\mathbb{R}^d$

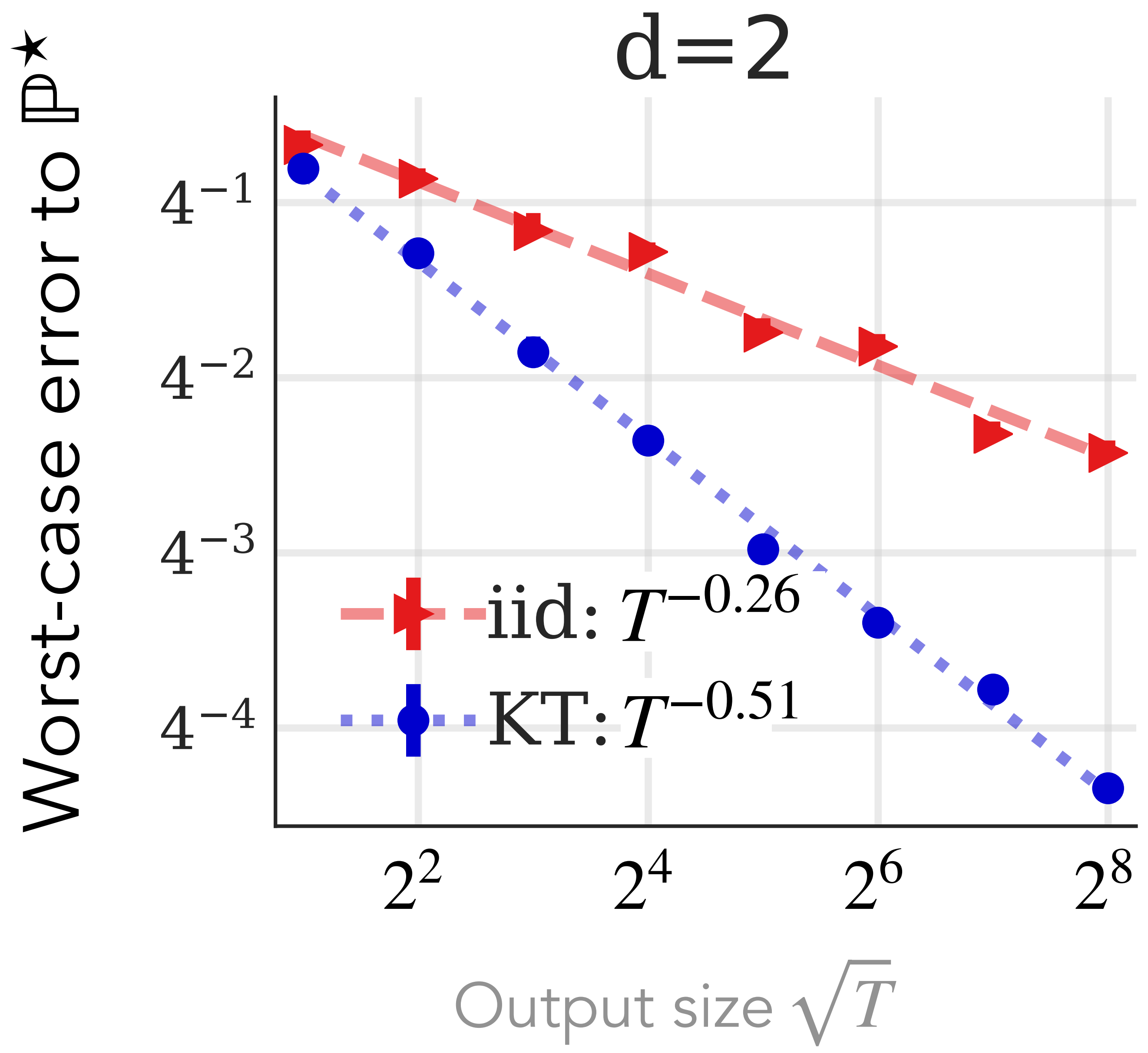
iid input, Gaussian kernel

Worst-case error to $\mathbb{P}^\star$

Output size $\sqrt{T}$

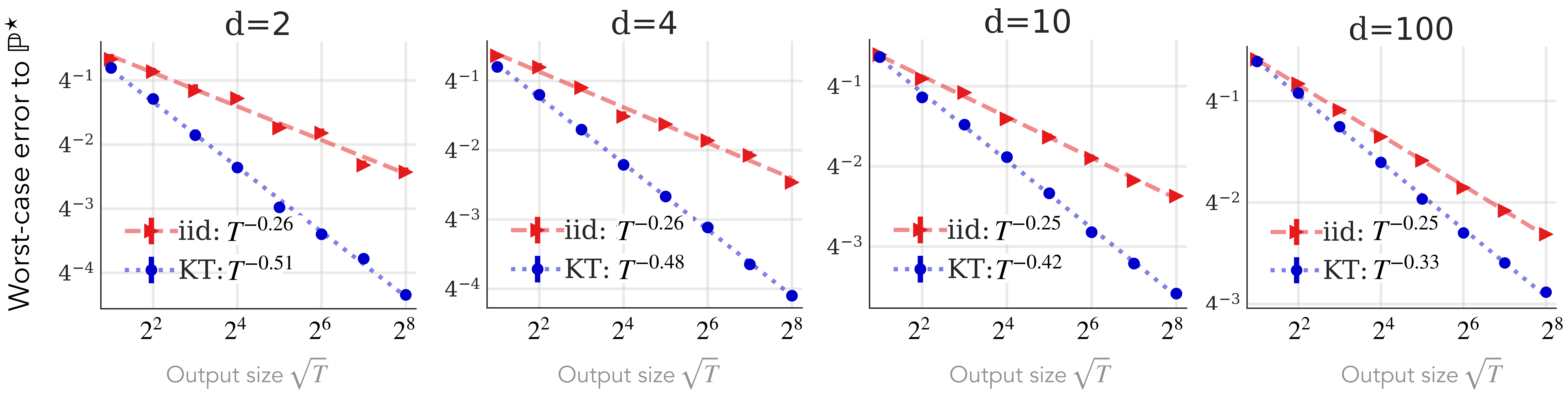
Is KT better practically? Gaussian $\mathbb{P}^\star$ in $\mathbb{R}^d$

iid input, Gaussian kernel



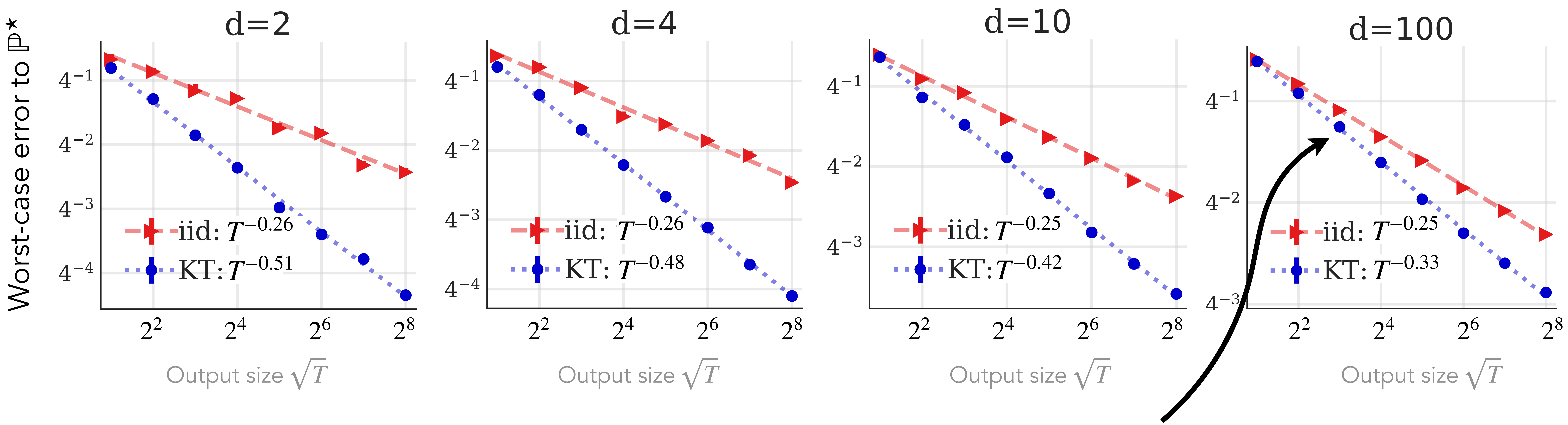
Is KT better practically? Gaussian $\mathbb{P}^\star$ in $\mathbb{R}^d$

iid input, Gaussian kernel



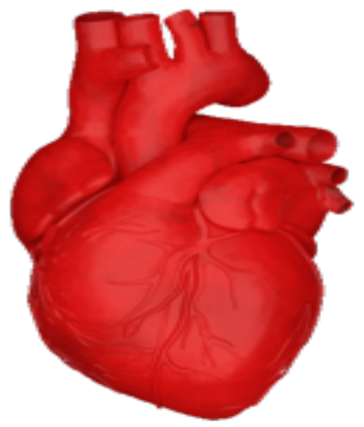
Is KT better practically? Gaussian $\mathbb{P}^\star$ in $\mathbb{R}^d$

iid input, Gaussian kernel



Significant gains in $d = 100$

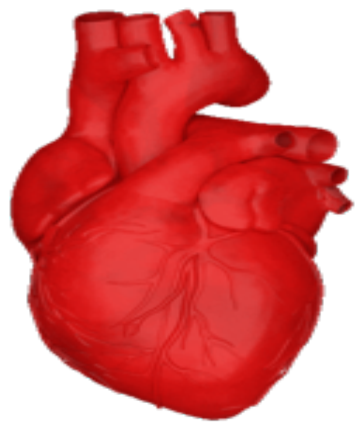
with just 8 output points

KT on MCMC points for $\mathbb{P}^\star$ in  experiments $(d = 38)$

†Input = 2 MCMC runs on 2 posteriors $\mathbb{P}^\star$, Gaussian kernel

Worst-case error to $\mathbb{P}_T$

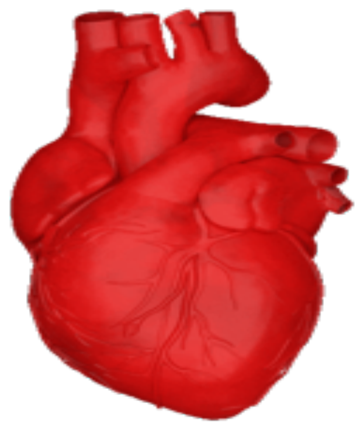
[†MCMC data from Riabiz-Chen-Cockayne-Swietach-Niederer-Mackey-Oates '21]

KT on MCMC points for $\mathbb{P}^\star$ in  experiments ($d = 38$)

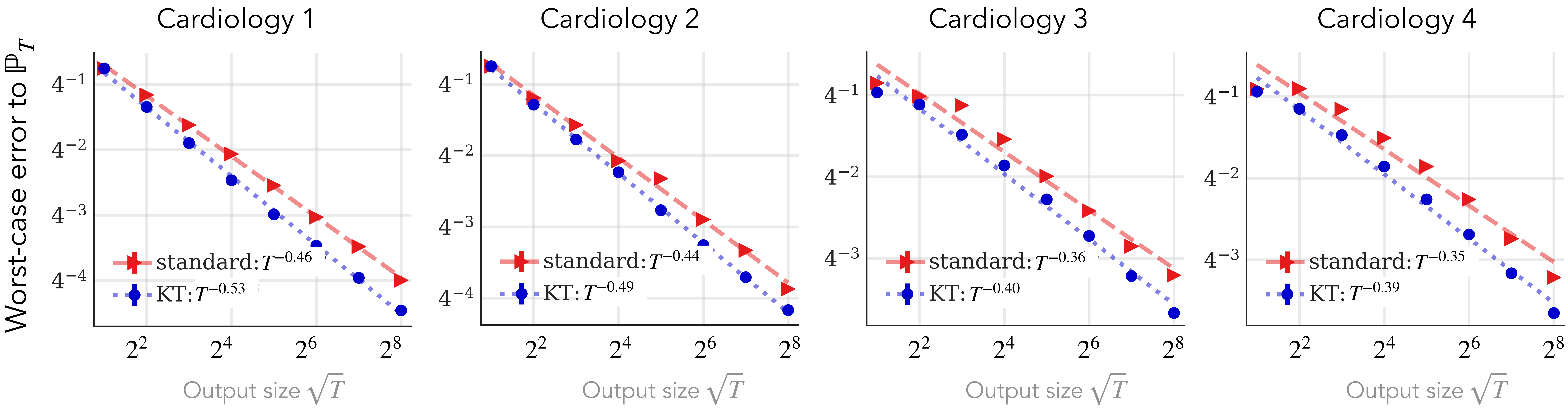
†Input = 2 MCMC runs on 2 posteriors $\mathbb{P}^\star$, Gaussian kernel

Worst-case error to $\mathbb{P}_T$

[†MCMC data from Riabiz-Chen-Cockayne-Swietach-Niederer-Mackey-Oates '21]

KT on MCMC points for $\mathbb{P}^\star$ in  experiments ($d = 38$)

†Input = 2 MCMC runs on 2 posteriors $\mathbb{P}^\star$, Gaussian kernel



Standard thinning does well but **KT provides further improvement**
& offers **50% computational savings** (each point ~ 4 CPU weeks)

[†MCMC data from Riabiz-Chen-Cockayne-Swietach-Niederer-Mackey-Oates ’21]

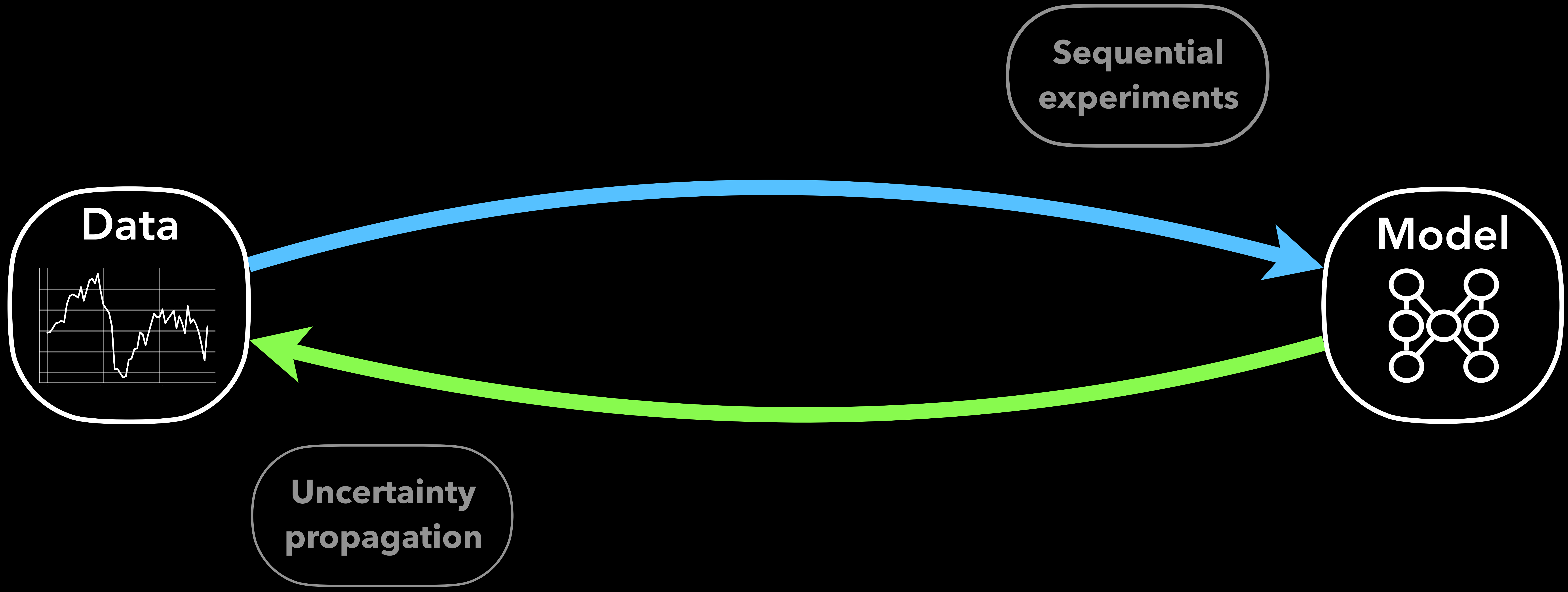
Kernel thinning: Near-optimal compression in near-linear time

Kernel thinning: Near-optimal compression in near-linear time

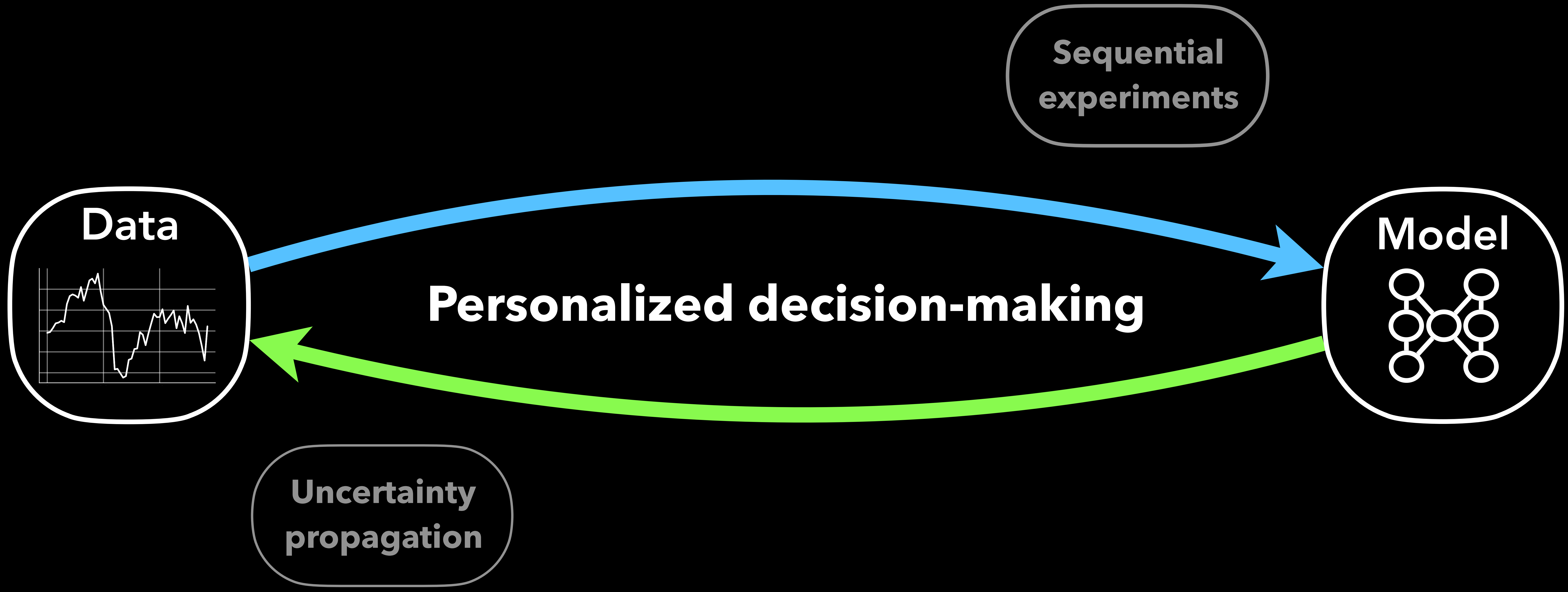
python™ **pip install goodpoints**

Thin 100k points in 100 dimensions in 10mins

talk summary



talk summary



From HeartSteps

Personalized inference by **averaging neighborhoods**

Sequential experiments

Uncertainty propagation

talk summary

From HeartSteps

Personalized inference by **averaging neighborhoods**

Sequential experiments

Uncertainty propagation

to HeartBeats

Personalized simulations by **thinning neighborhoods**

talk summary

From HeartSteps

Personalized inference by **averaging neighborhoods**

Quadratic gains via double robustness

Sequential experiments

Uncertainty propagation

to HeartBeats

Personalized simulations by **thinning neighborhoods**

talk summary

From HeartSteps

Personalized inference by **averaging neighborhoods**

Quadratic gains via double robustness

Sequential experiments

Uncertainty propagation

to HeartBeats

Personalized simulations by **thinning neighborhoods**

Quadratic gains via discrepancy minimization

talk summary

From HeartSteps

Personalized inference by **averaging neighborhoods**

Quadratic gains via double robustness

Sequential experiments

Uncertainty propagation

to HeartBeats

Personalized simulations by **thinning neighborhoods**

Quadratic gains via discrepancy minimization

talk summary

From HeartSteps

Personalized inference by **averaging neighborhoods**

Quadratic gains via double robustness

Sequential experiments

Uncertainty propagation

to HeartBeats

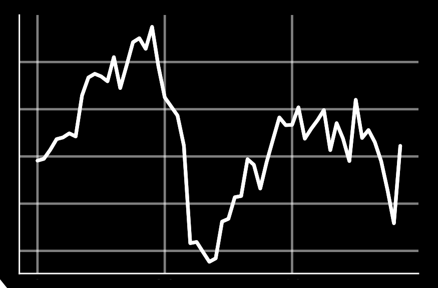
Personalized simulations by **thinning neighborhoods**

Quadratic gains via discrepancy minimization


Sequential experiments
Data
Model
Uncertainty propagation

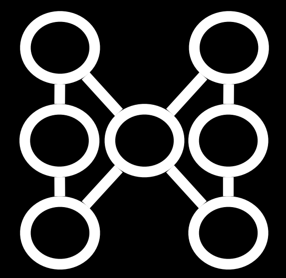
Sequential experiments

Data



Data and computation efficient methods
for personalized decision-making

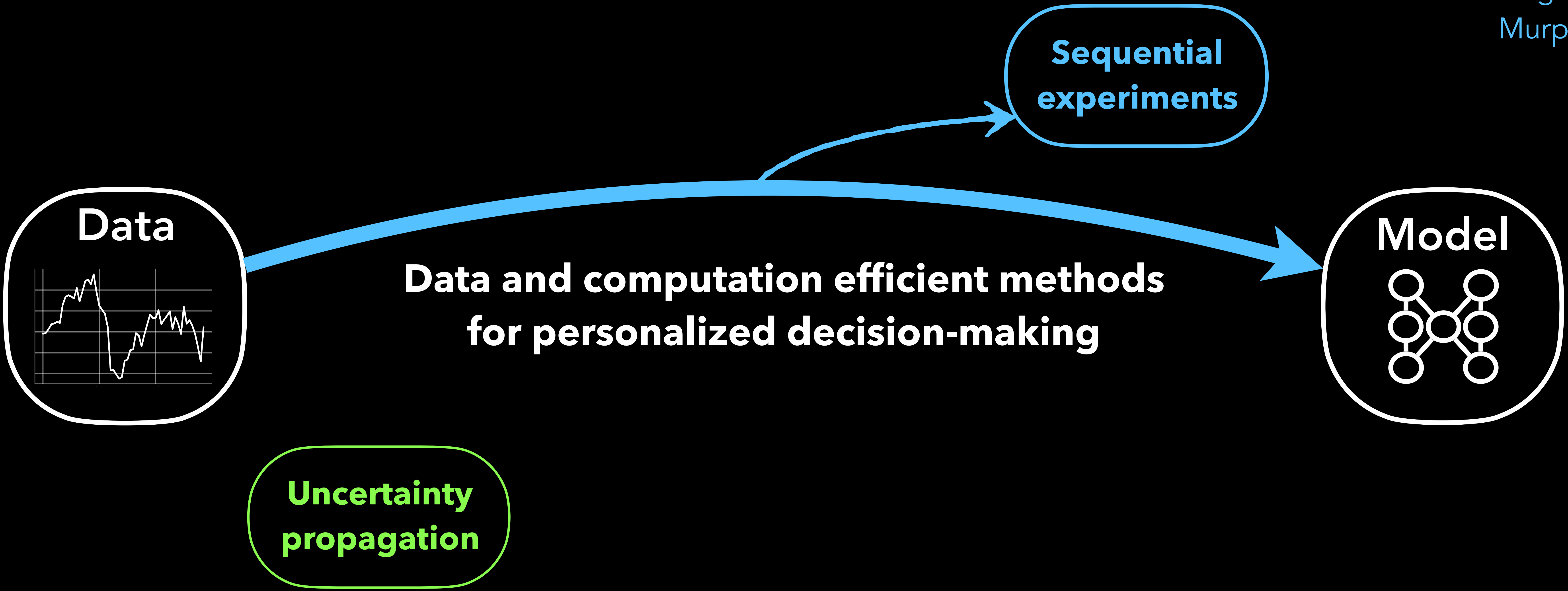
Model



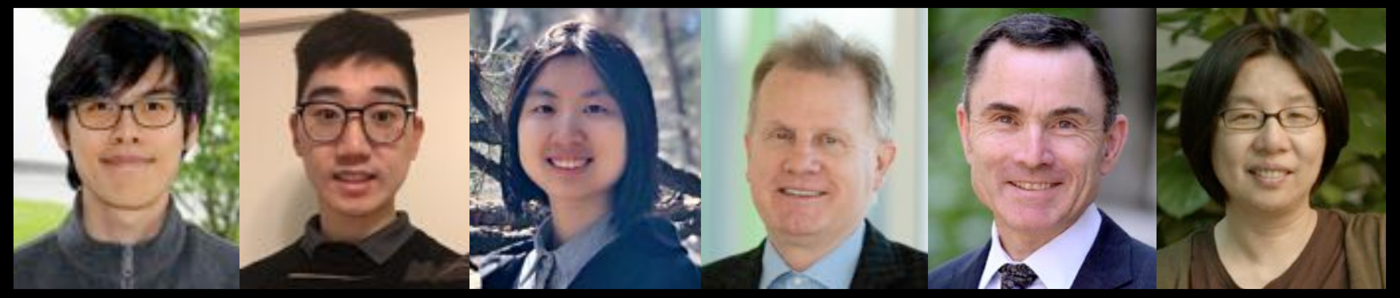
Uncertainty propagation

Deep dive into personalization by a reinforcement learning algorithm

Dwivedi*-Zhang*-Chhabria-Klasnja-Murphy '23



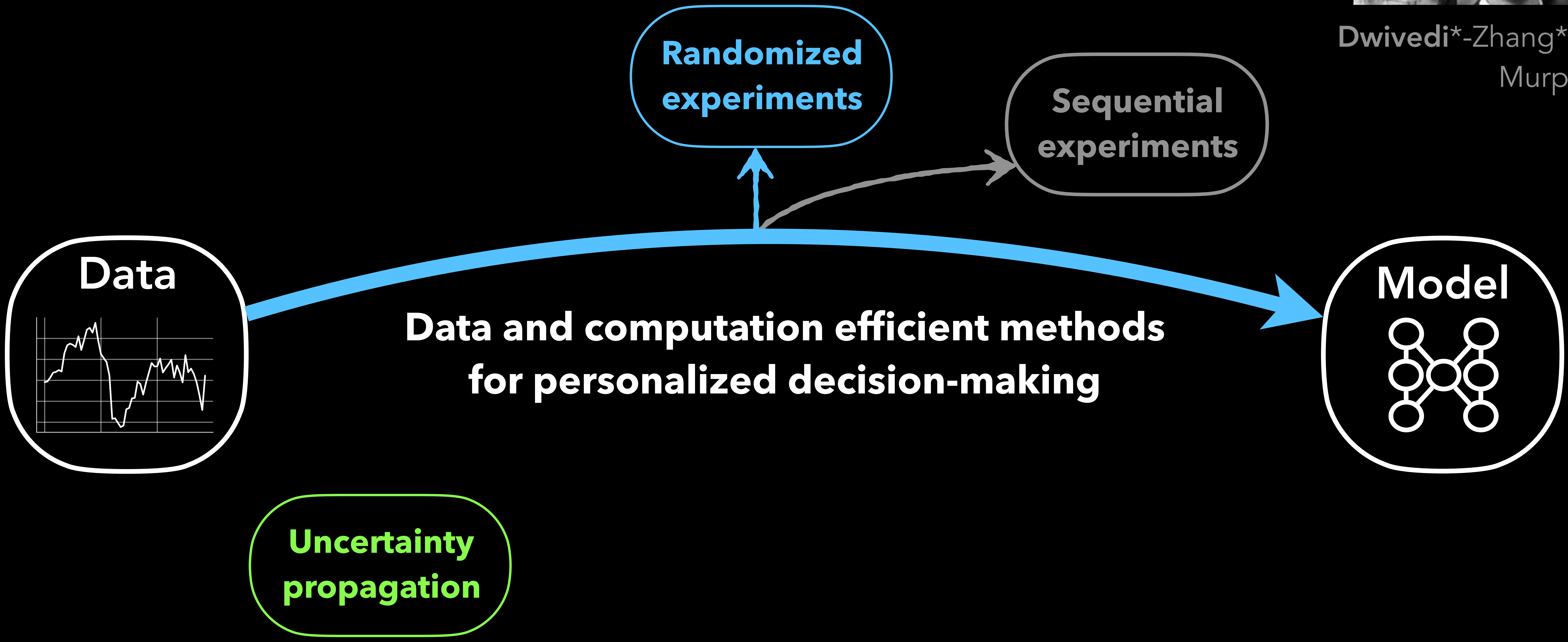
Stable discovery of interpretable subgroups in randomized studies via calibration



Dwivedi*-Tan*-Park-Wei-Horgan-Madigan-Yu '20

Deep dive into personalization by a reinforcement learning algorithm

Dwivedi*-Zhang*-Chhabria-Klasnja-Murphy '23



On counterfactual inference with unobserved confounding via exponential family

Shah-**Dwivedi**-Shah-Wornell '22

Stable discovery of interpretable subgroups in randomized studies via calibration

Dwivedi*-Tan*-Park-Wei-Horgan-Madigan-Yu '20

Deep dive into personalization by a reinforcement learning algorithm

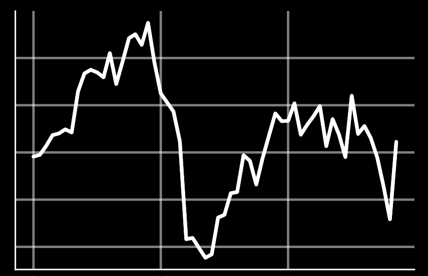
Dwivedi*-Zhang*-Chhabria-Klasnja-Murphy '23

Observational studies

Randomized experiments

Sequential experiments

Data



Data and computation efficient methods for personalized decision-making

Model

Uncertainty propagation

research overview

On counterfactual inference with unobserved confounding via exponential family

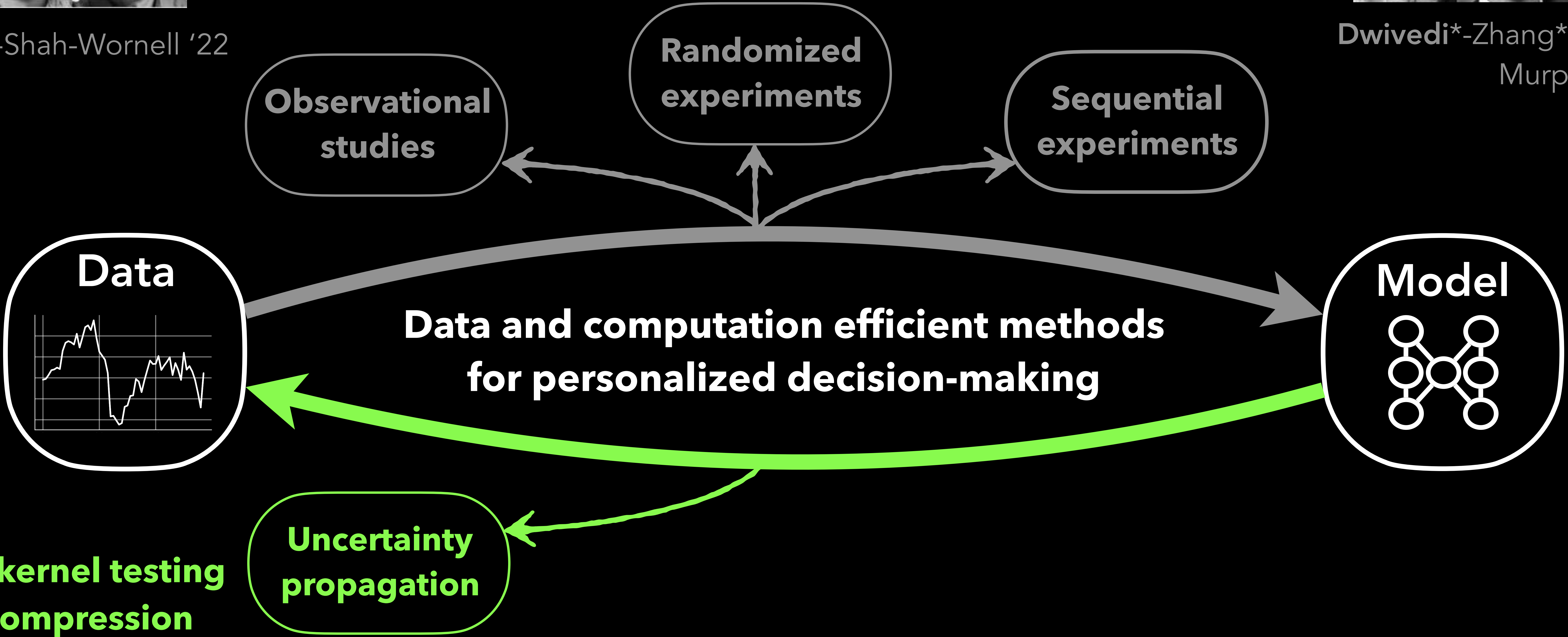
Shah-**Dwivedi**-Shah-Wornell '22

Stable discovery of interpretable subgroups in randomized studies via calibration

Dwivedi*-Tan*-Park-Wei-Horgan-Madigan-Yu '20

Deep dive into personalization by a reinforcement learning algorithm

Dwivedi*-Zhang*-Chhabria-Klasnja-Murphy '23



Fast and powerful kernel testing via distribution compression

Shetty-**Dwivedi**-Mackey '22,
Domingo Enrich-**Dwivedi**-Mackey '23

research overview

On counterfactual inference with unobserved confounding via exponential family

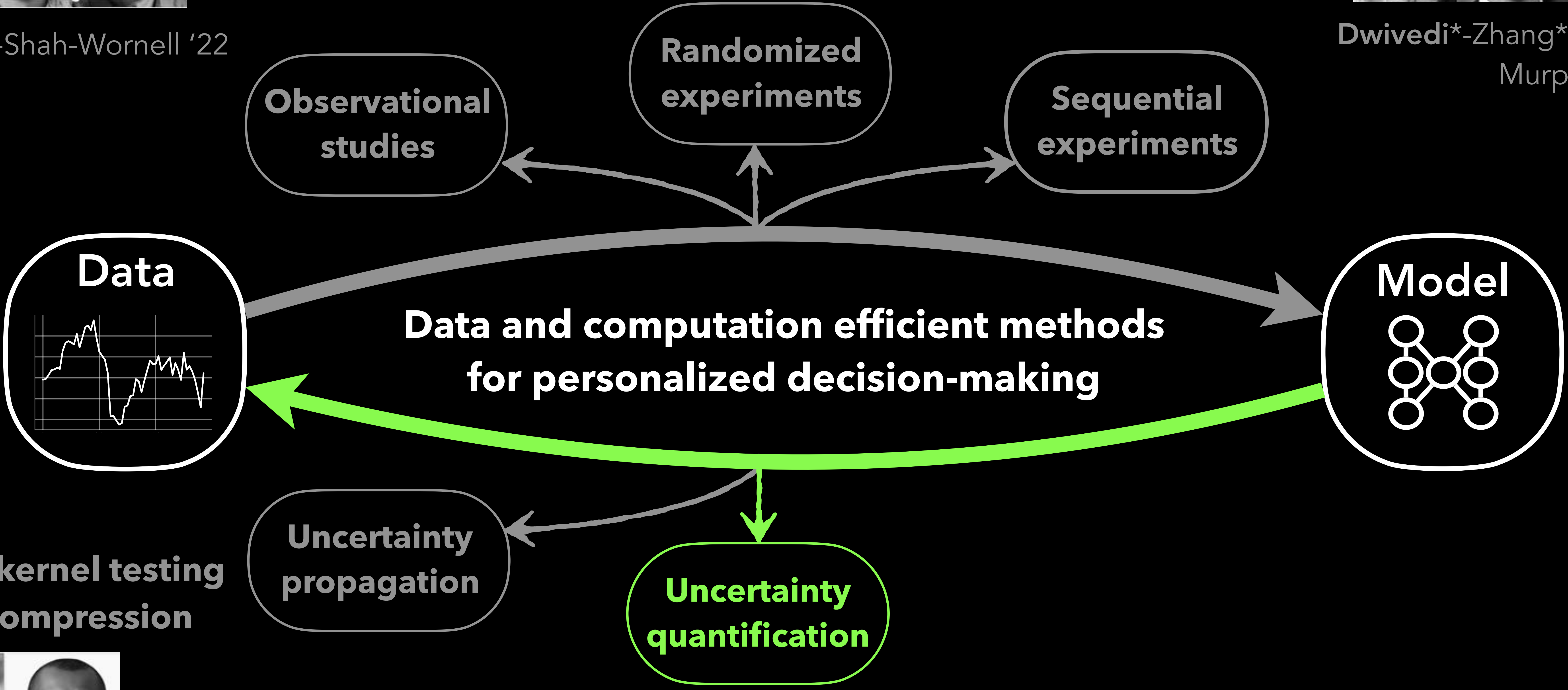
Shah-**Dwivedi**-Shah-Wornell '22

Stable discovery of interpretable subgroups in randomized studies via calibration

Dwivedi*-Tan*-Park-Wei-Horgan-Madigan-Yu '20

Deep dive into personalization by a reinforcement learning algorithm

Dwivedi*-Zhang*-Chhabria-Klasnja-Murphy '23



Fast and powerful kernel testing via distribution compression

Shetty-**Dwivedi**-Mackey '22,
Domingo Enrich-**Dwivedi**-Mackey '22

Mixing time guarantees for MCMC algorithms in high dimensions

Chen*-**Dwivedi***-Wainwright-Yu '18, '19, '20

Stable discovery of interpretable subgroups in randomized studies via calibration

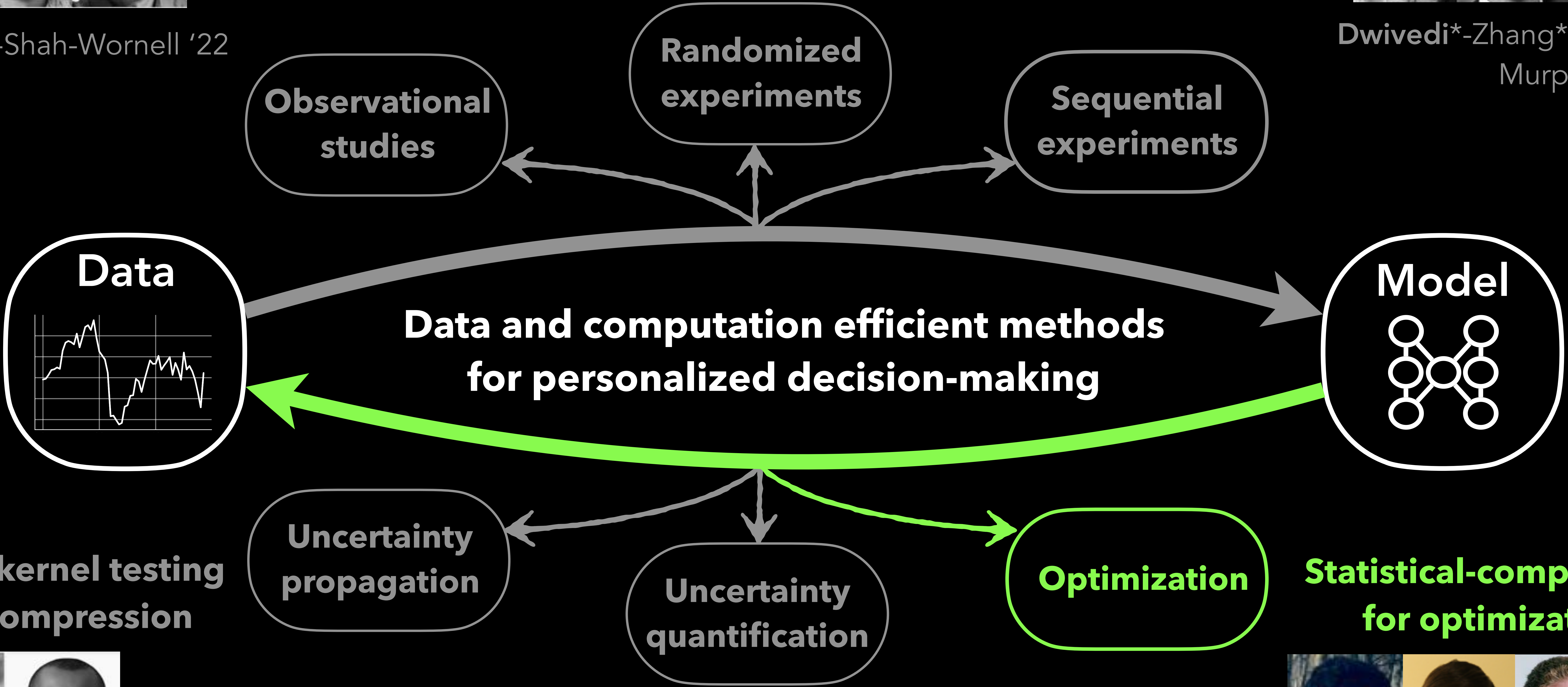
Dwivedi*-Tan*-Park-Wei-Horgan-Madigan-Yu ’20

On counterfactual inference with unobserved confounding via exponential family

Shah-**Dwivedi**-Shah-Wornell ’22

Deep dive into personalization by a reinforcement learning algorithm

Dwivedi*-Zhang*-Chhabria-Klasnja-Murphy ’23



Fast and powerful kernel testing via distribution compression

Shetty-**Dwivedi**-Mackey ’22,
Domingo Enrich-**Dwivedi**-Mackey ’22

Mixing time guarantees for MCMC algorithms in high dimensions

Chen*-**Dwivedi***-Wainwright-Yu ’18, ’19, ’20

Statistical-computational tradeoffs for optimization algorithms

Dwivedi*-Ho*-Khamaru*-Wainwright-Jordan-Yu ’19, ’20, ’21, ’22+

On counterfactual inference with unobserved confounding via exponential family

Shah-**Dwivedi**-Shah-Wornell '22

Stable discovery of interpretable subgroups in randomized studies via calibration

Dwivedi*-Tan*-Park-Wei-Horgan-Madigan-Yu '20

Deep dive into personalization by a reinforcement learning algorithm

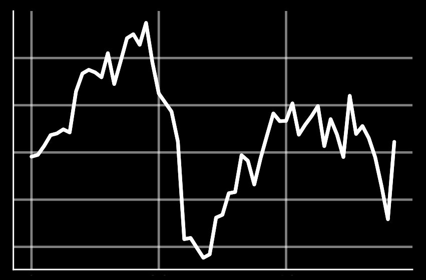
Dwivedi*-Zhang*-Chhabria-Klasnja-Murphy '23

Observational studies

Randomized experiments

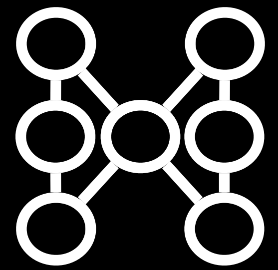
Sequential experiments

Data



Data and computation efficient methods for personalized decision-making

Model



Uncertainty propagation

Uncertainty quantification

Optimization

Fast and powerful kernel testing via distribution compression

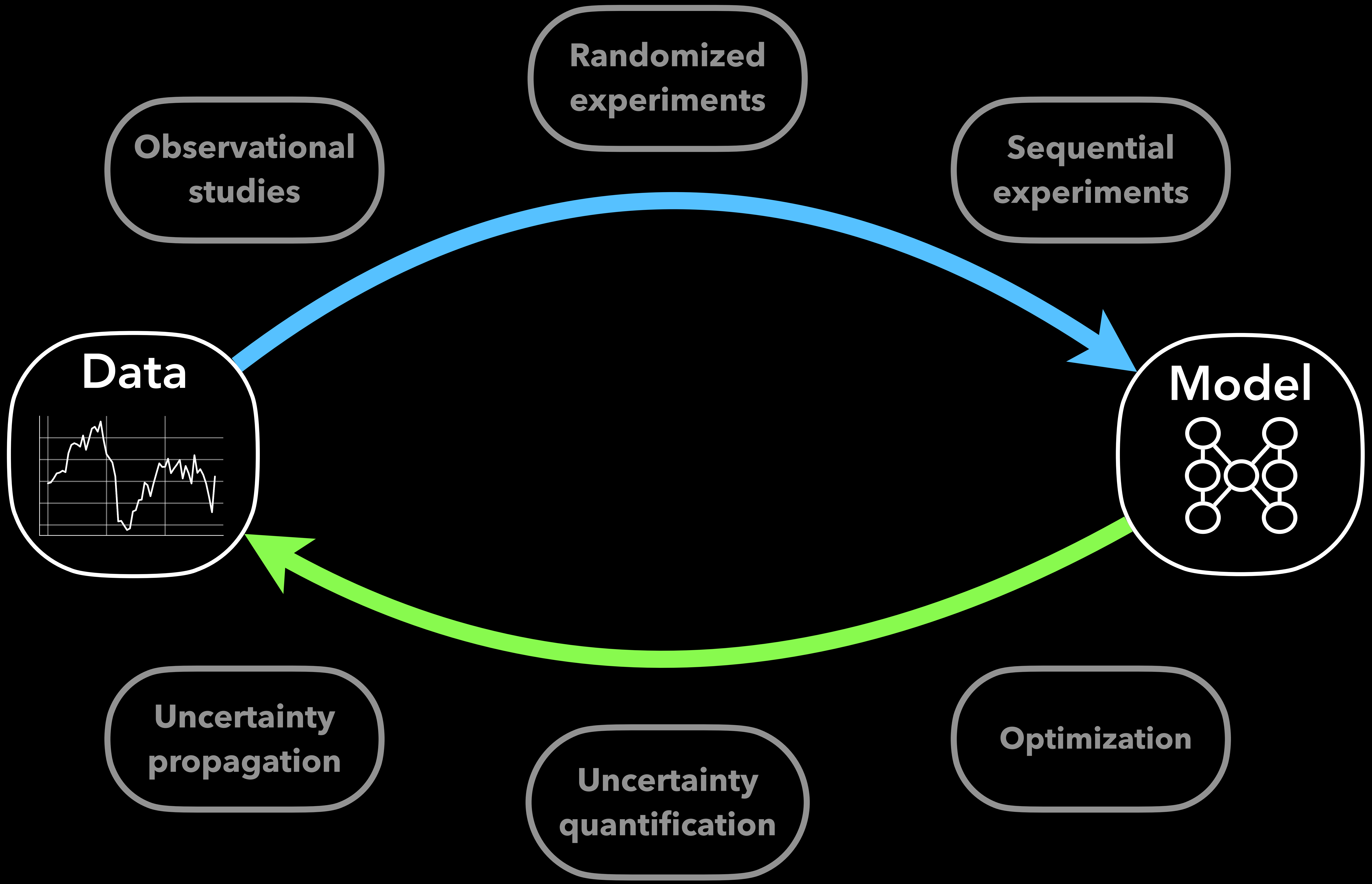
Shetty-**Dwivedi**-Mackey '22,
Domingo Enrich-**Dwivedi**-Mackey '22

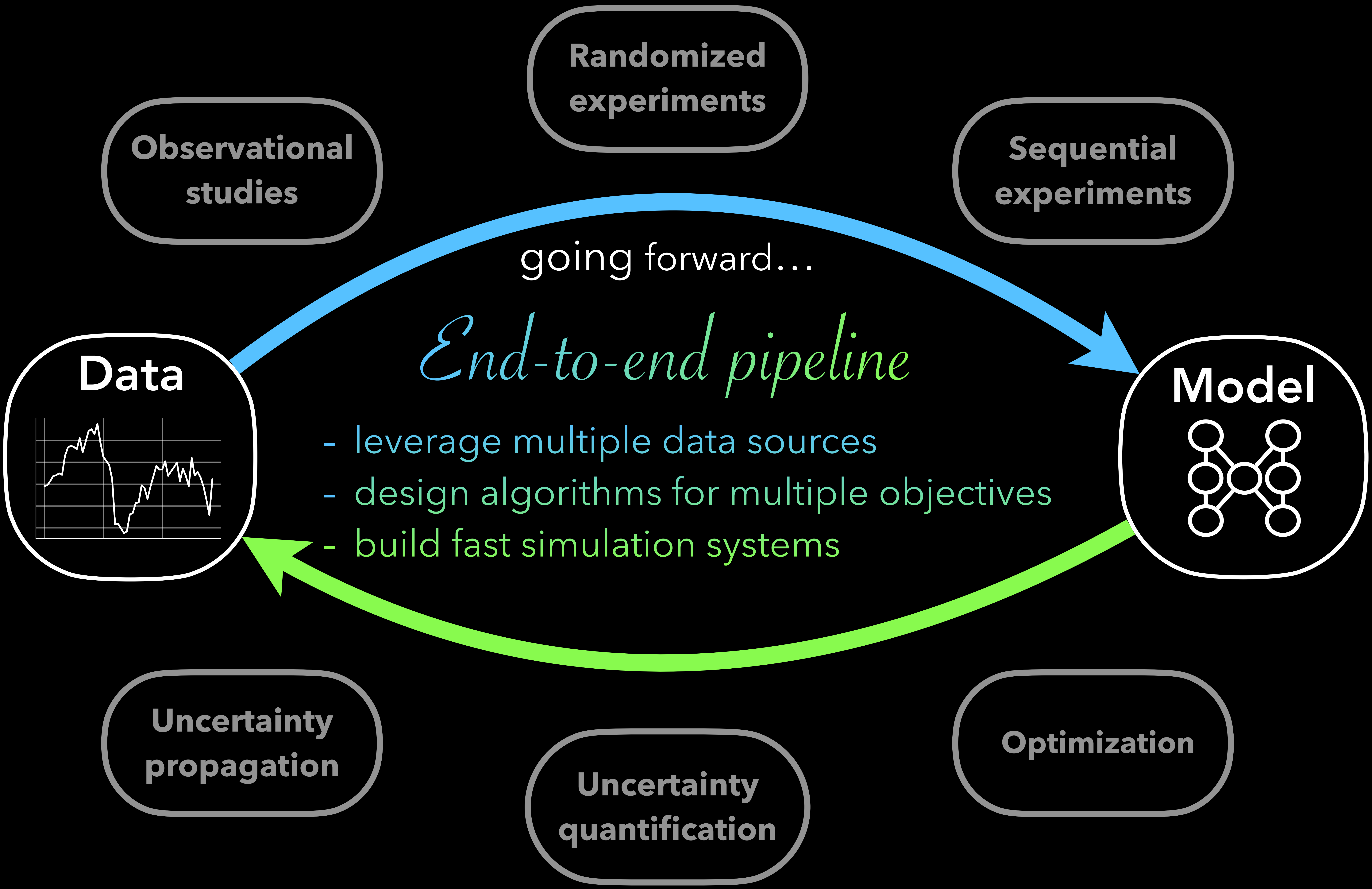
Mixing time guarantees for MCMC algorithms in high dimensions

Chen*-**Dwivedi***-Wainwright-Yu '18, '19, '20

Statistical-computational tradeoffs for optimization algorithms

Dwivedi*-Ho*-Khamaru*-Wainwright-Jordan-Yu '19, '20, '21, '22+


Randomized experiments
Observational studies
Sequential experiments
Data
Model
Uncertainty propagation
Uncertainty quantification
Optimization



Thank you!

raazdwivedi.github.io