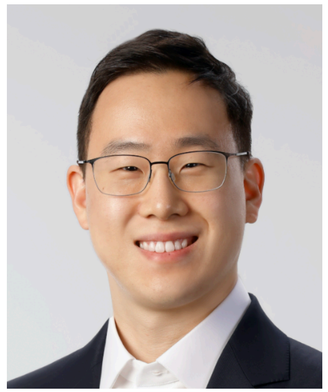
Optimization-Driven Adaptive Experimentation

Hongseok Namkoong

Columbia University



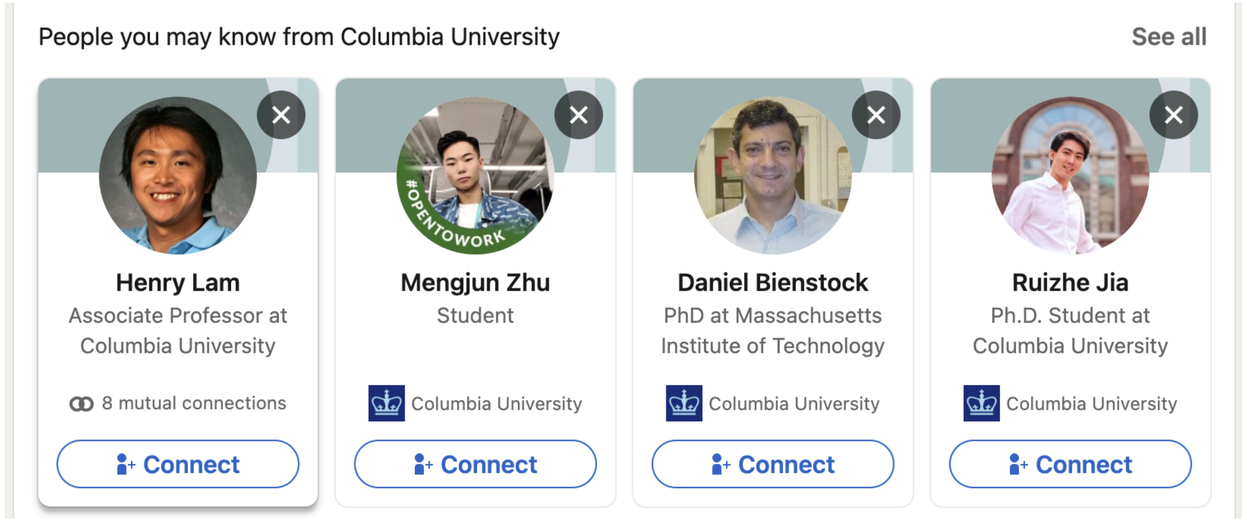
Ethan Che
Columbia

Daniel Jiang
Meta

Jimmy Wang
Columbia

Experimentation (prediction ⇒ decision)

- Imagine a ML engineer building a recommendation system



Configuration 1 2 … K

Goal: help users grow their professional network

- Underpowered: quality of service improvement < 2%
  - Business impact can nevertheless be big!

Adaptivity

- Adaptivity improves power => change how we do science!
  - Expand testable hypotheses beyond usual binary options
- Vast literature assuming unit-level continual reallocation
  - Thompson (’33), Chernoff (’59), Robbins & Lai (’52, ’85) + 1000s others
- Algo design guided by theory: regret as # reallocation $T \to \infty$

Batched Feedback

Challenges in adaptive experimentation

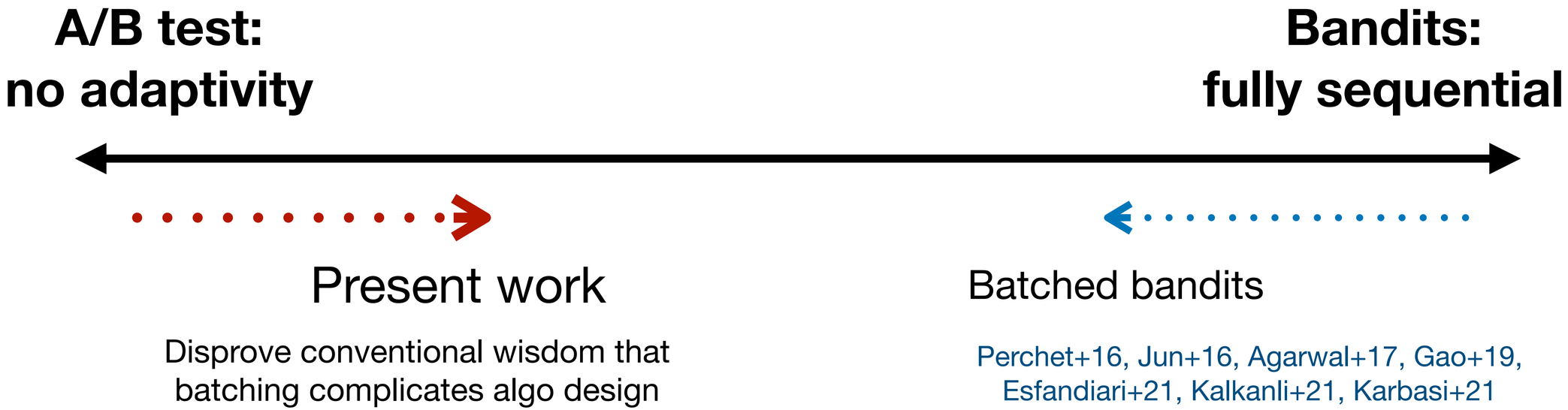
Practical setting: a **few, large batches**

(think $T = 7$ batches with $n = 100{,}000$ users per batch)

Due to delayed feedback or operational efficiency

Disclaimer for experts

- NOT about continual interaction nor sublinear regret (T=7)
  - It's all about constants! We want 30% gain in experiment efficiency.



Non-stationarity

Challenges in adaptive experimentation

- Treatment effects change over day-of-the-week

ASOS Dataset

Fashion retailer with > 26m active customers

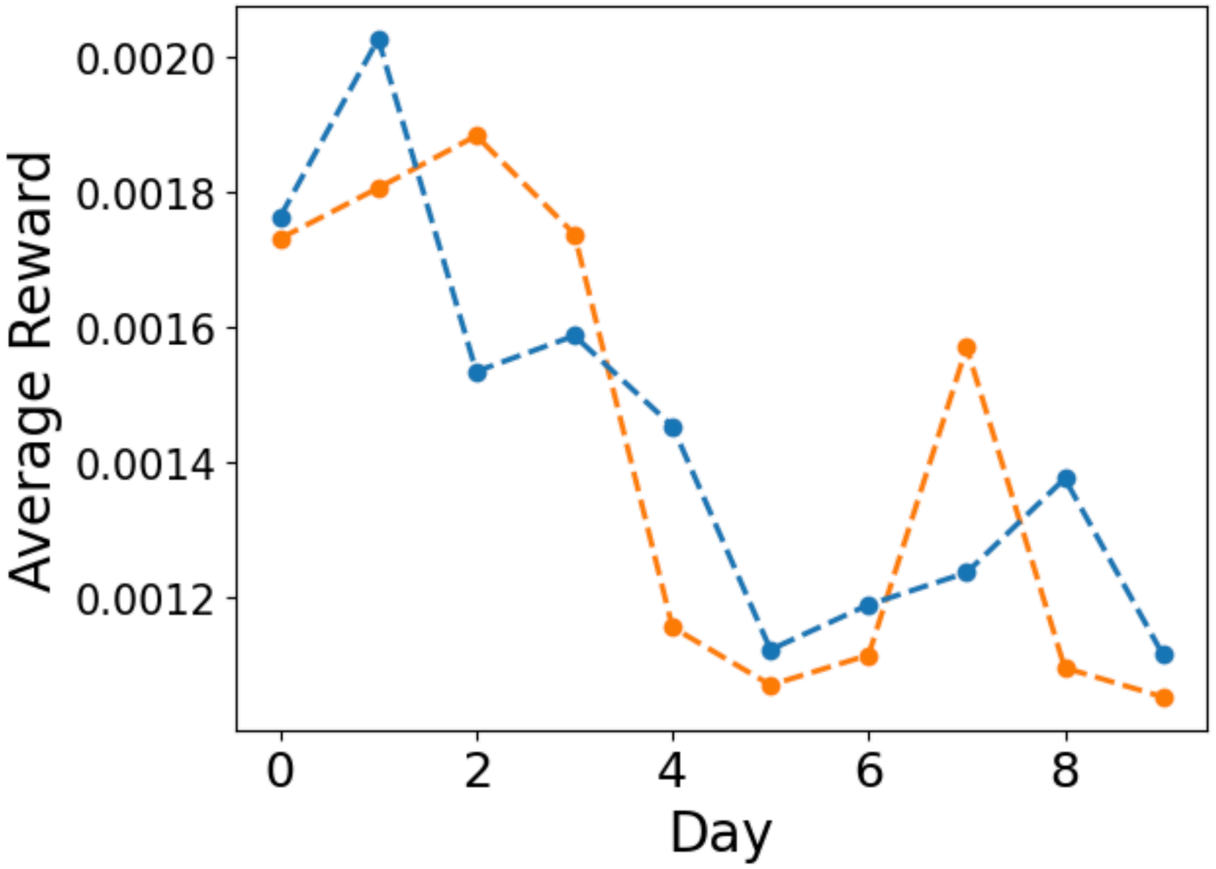
- 78 RCTs, two arms, four metrics
  - Mean, variances every 12 or 24-hours
  - 2~132 recorded intervals
- Generate 241 benchmark settings
  - By adding arms (total 10 arms) with similar gaps as real ones

ASOS Dataset

Fashion retailer with > 26m active customers

- 78 RCTs, two arms, four metrics
  - (mean, var) every 12/24 hours
  - 2~132 recorded intervals
- Generate 241 benchmark settings
  - By adding arms (total 10 arms) with similar gaps as real ones

Best arm changes every day

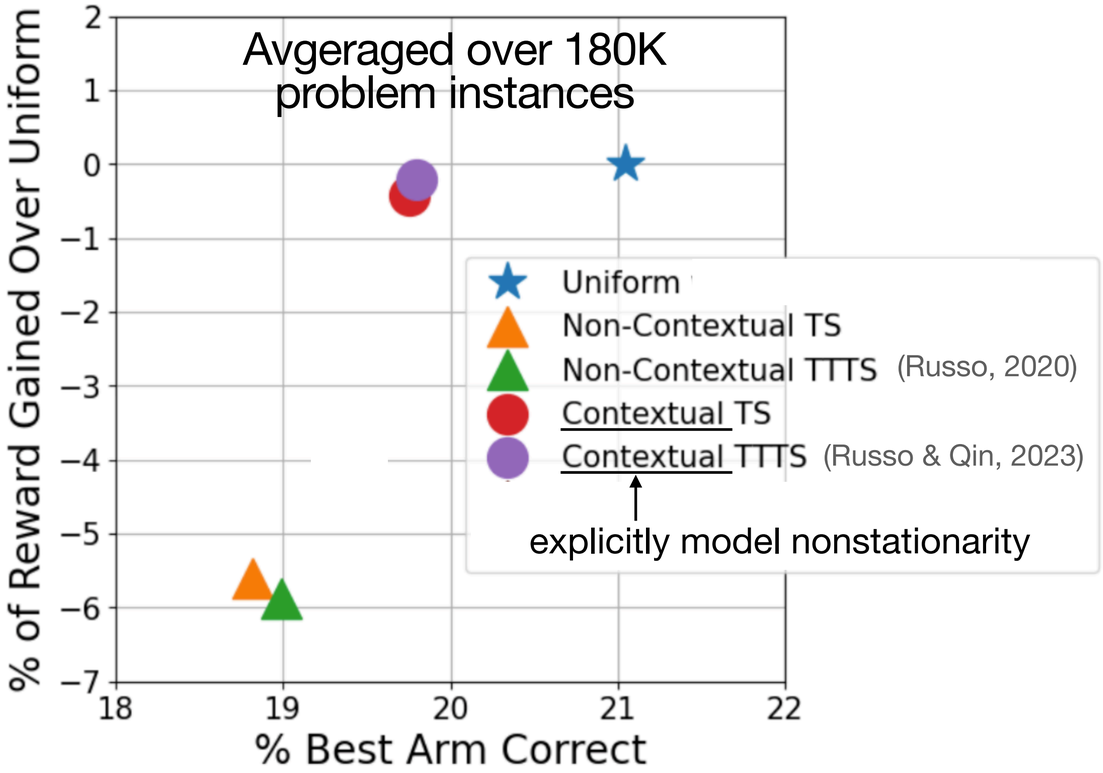


Vignette: Static RCT outperforms SoTA bandits

- TS: Select arms with Prob(arm optimal | History) [Thompsom, 1933]
- Top-two (TT): Same, but give second best arm a chance [Russo, 2020]

Vignette: Static RCT outperforms SoTA bandits

- TS: Select arms with Prob(arm optimal | History) [Thompsom, 1933]
- Top-two (TT): Same, but give second best arm a chance [Russo, 2020]



Overfits on initial, temporary performance when T = 10

batch size = 100K

People want different things

Challenges in adaptive experimentation

People want different things

Challenges in adaptive experimentation

- **Best Arm Identification:** I want the best treatment or max power

People want different things

Challenges in adaptive experimentation

- **Best Arm Identification:** I want the best treatment or max power
- **Top 5 Arm Identification:** actually, I want top-5 arms

People want different things

Challenges in adaptive experimentation

- **Best Arm Identification:** I want the best treatment or max power
- **Top 5 Arm Identification:** actually, I want top-5 arms
- **Personalization:** learn a *policy* that assigns treatments to users.

People want different things

Challenges in adaptive experimentation

- **Best Arm Identification:** I want the best treatment or max power
- **Top 5 Arm Identification:** actually, I want top-5 arms
- **Personalization:** learn a *policy* that assigns treatments to users.
- **Multiple Metrics:** find best arm in a primary metric that's not worse than control in another guardrail metric.

Constraints

Challenges in adaptive experimentation

- **Sample Coverage:** at least 10% of samples for all arms
- **Budget Constraint:** can’t give too many discounts
- **Quality of Service:** don’t want a regression in this metric
- **Pacing:** use budget efficiently over the experiment

Challenges in adaptive experimentation

What is a good algorithmic design principle for…

Top 5 arm identification +

Batched Feedback +

Non-stationarity +

Sample coverage constraints + …

…that will actually materialize into practical performance?

Current art

- Step 1: Hire top bandit researcher for two years
- Step 2: Develop a variant of Thomson sampling adapted to your particular objective & constraints
- Step 3: Prove a nice regret bound for said algorithm

- When infeasible, apply some algo not designed for your instance
  - Brittle performance: often even worse than uniform

Mathematical Programming

$$\begin{aligned} \text{minimize}_{\pi} \quad & \text{Objective}(\pi) \\ \text{subject to} \quad & \text{Constraint}(\pi) \leq B \end{aligned}$$

- Write down in a modeling language (e.g., CVX)
- Call a generic solver to get approximate solution (e.g., Gurobi)
- Good solvers should perform well across a wide set of problem instances, rather than focus only on a particular problem

Why do we design problem-specific algos?

Batched Experiments

For t in range(T):

Two Treatment Arms: 

Sampling Allocation π_t

π_t

Batched Experiments

For t in range(T):

Two Treatment Arms:

Sampling Allocation π_t

π_t

Users x_t

Batched Experiments

For t in range(T):

Two Treatment Arms:

Sampling Allocation π_t

π_t

Users x_t

Batched Experiments

For t in range(T):

Two Treatment Arms:

Sampling Allocation π_t

π_t

Users x_t

Treatments a_t

Batched Experiments

For t in range(T):

Two Treatment Arms: 

Sampling Allocation π_t

π_t 

Users x_t 

Treatments a_t 

Features ϕ $\phi($  $)$ $\phi($  $)$ $\phi($  $)$ $\phi($  $)$ $\phi($  $)$

Batched Experiments

For t in range(T):

Two Treatment Arms: 

Sampling Allocation π_t

π_t 

Users x_t 

Treatments a_t 

Features ϕ $\phi($  $)$ $\phi($  $)$ $\phi($  $)$ $\phi($  $)$ $\phi($  $)$

Rewards R_t **1** **0** **0** **0** **1**

Batched Experiments

For t in range(T):

Two Treatment Arms: 

Sampling Allocation π_t

π_t

Users x_t

Treatments a_t

Features ϕ

$\phi($  $)$ $\phi($  $)$ $\phi($  $)$ $\phi($  $)$ $\phi($  $)$

Rewards R_t

1 **0** **0** **0** **1**

Adaptive experimentation as dynamic program

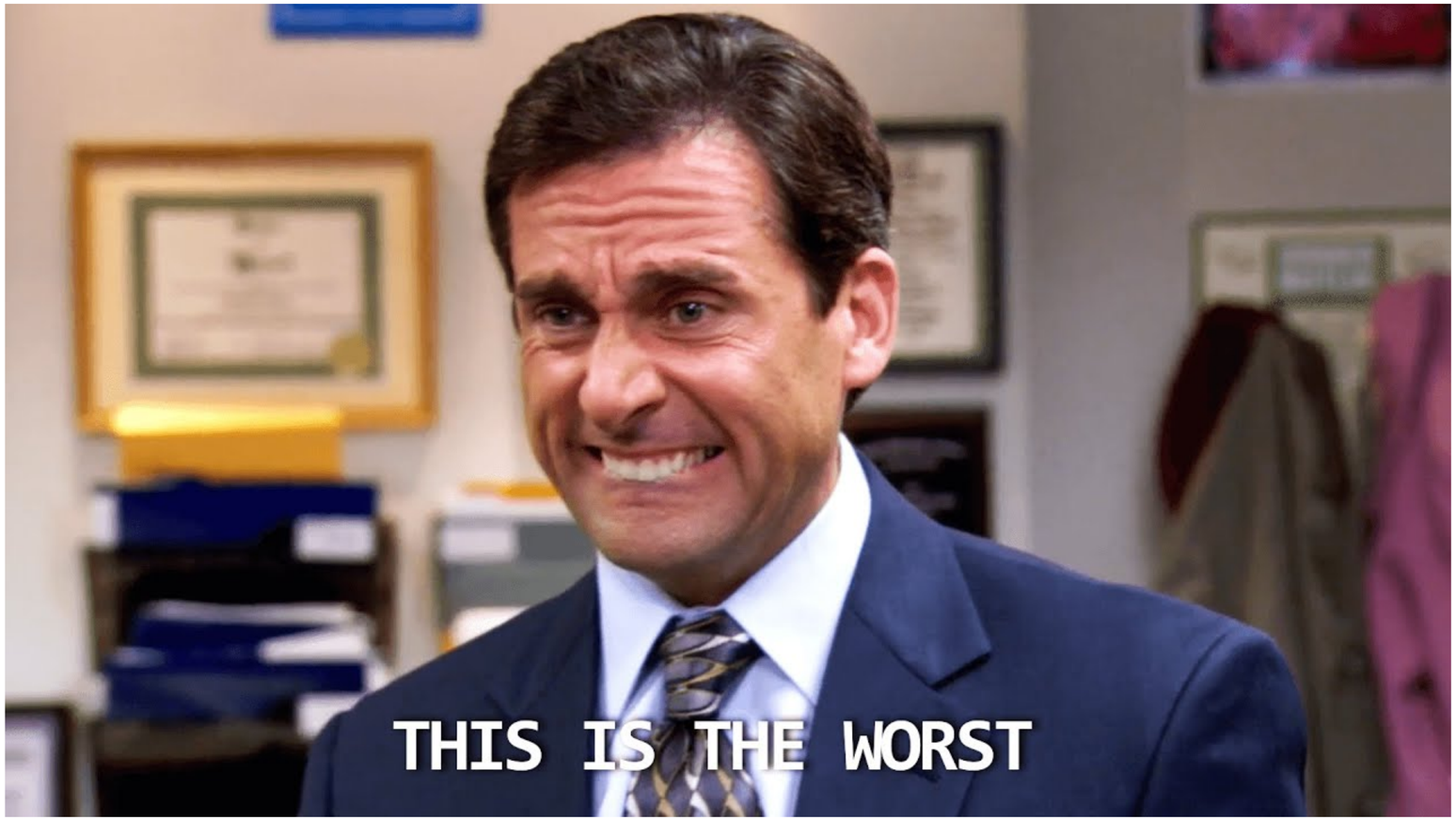
$$\text{minimize}_{\pi_t(H_t)} \quad \mathbb{E}\left[\sum_{t=0}^{T} \text{Objective}_t(\pi_t, H_t)\right]$$

$$\text{subject to} \quad \mathbb{E}\left[\sum_{t=0}^{T} \text{Cost}(\pi_t; H_t)\right] \leq c$$

H_t : history at epoch t

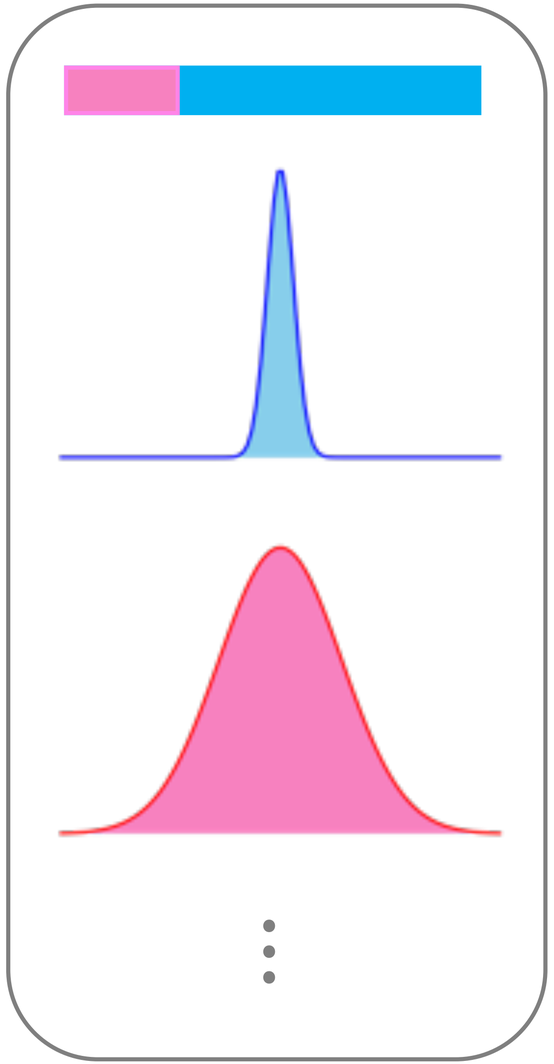
1. Unknown reward distribution
2. State space exponential in # units

Adaptive experimentation as dynamic program



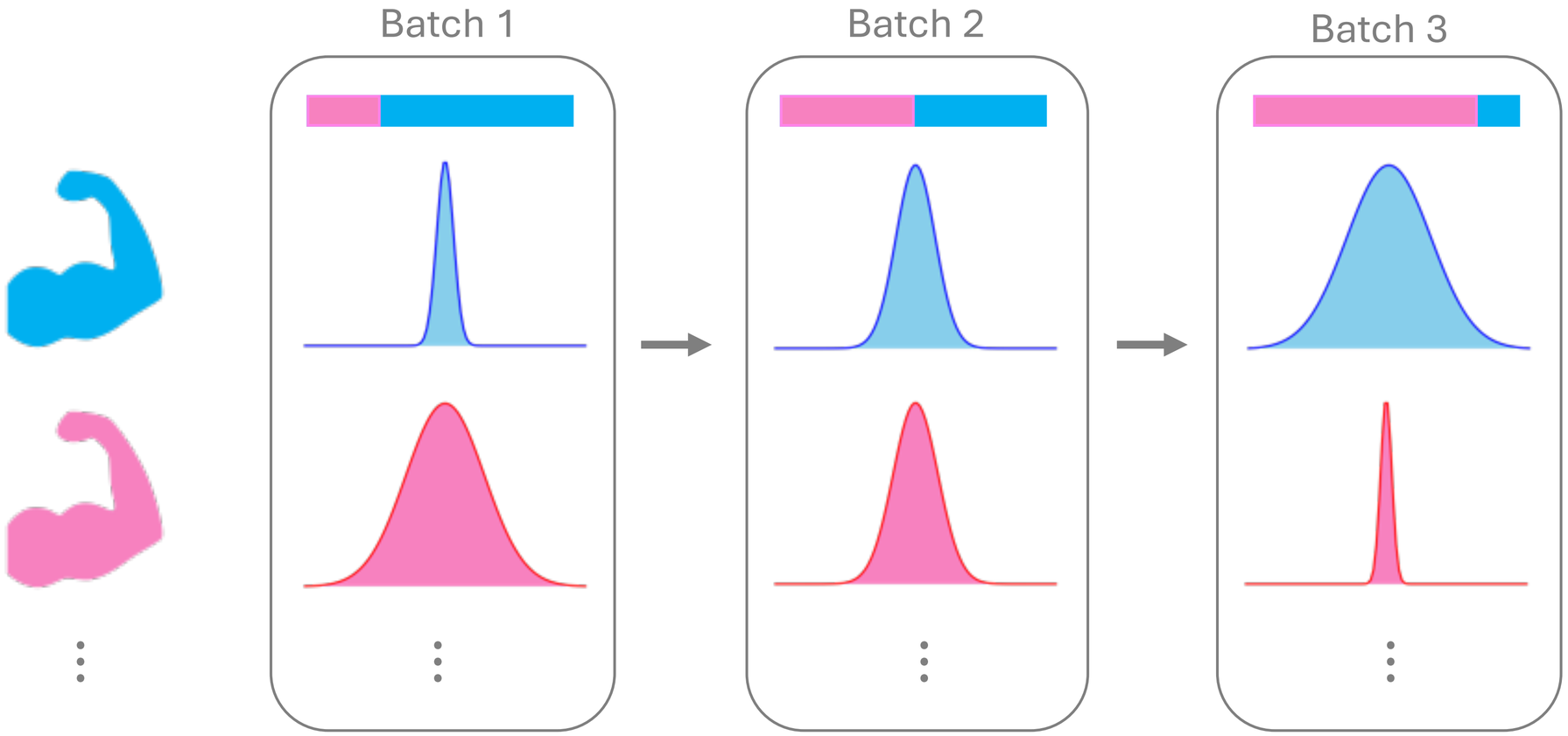
Gaussian approximations

Sample mean in a batch ~ Gaussian



- Allocation controls the effective sample size
  - Gaussian is skinny if the arm is sampled more
- Normal approximations, universal in inference, is also useful for design of adaptive algorithms

Gaussian sequential experiment



Sequence of Gaussian observations gives a tractable MDP

Modeling average behavior

- Parametric model for mean rewards
- Examples
  - Non-contextual: $\theta^\star$ = average reward across arms
  - Contextual model: for known feature map $\phi(X, A)$,
    - Linear/logistic: $\mathbb{E}[R \mid X, A] = \text{Link}(\phi(X, A)^\top \theta^\star)$
    - Confounders: Terms that don't depend on A (e.g., day-of-the-week)

Gaussian approximations

- Within each batch t, central limit theorem says

maximum likelihood estimator $\hat{\theta}_t \sim N\left(\theta^\star, \frac{\text{Var}(\pi_t)}{n}\right)$

- 99% of statistics; everyone uses this to calculate p-values
- CLT compress entire batch to sufficient statistic $\hat{\theta}_t$

Compress batch to sufficient statistic

Governed by **posterior mean and variance** (β_t, Σ_t)

Prior | Likelihood | Posterior

$$\theta^\star \sim N(\beta_0, \Sigma_0) \longrightarrow \widehat{\theta}_t \sim N(\theta^\star, n^{-1}\text{Var}(\pi_t)) \longrightarrow \theta^\star \sim N(\beta_1, \Sigma_1)$$

Compress batch to sufficient statistic

Governed by **posterior mean and variance** (β_t, Σ_t)

Prior | Likelihood | Posterior

$$\theta^\star \sim N(\beta_0, \Sigma_0) \longrightarrow \widehat{\theta}_t \sim N(\theta^\star, n^{-1}\text{Var}(\pi_t)) \longrightarrow \theta^\star \sim N(\beta_1, \Sigma_1)$$

- Known, closed-form posterior state transitions
  - Posterior update formula for Gaussian conjugate family
  - Differentiable dynamics

Batch Limit Dynamic Program

$$\text{minimize}_{\pi_t(\beta_t,\Sigma_t)} \quad \mathbb{E}\left[\sum_{t=0}^{T} \text{Objective}_t(\pi_t, \beta_t, \Sigma_t)\right]$$

Posterior beliefs as states!

$$\text{subject to} \quad \mathbb{E}\left[\sum_{t=1}^{T} \text{Cost}(\pi_t; \beta_t, \Sigma_t)\right] \leq c$$

$$\pi_t(\beta_t, \Sigma_t) \in \text{Simplex}$$

- State dimension = $O(\dim(\theta)^2)$

Batch Limit Dynamic Program

- Model any objective and constraint written with posterior states
  - Cumulative- and simple-regret, top-k regret
  - Budget constraints, minimum allocation constraints
  - Above applied to any number of rewards/outcomes/metrics

- Today: Simple solver to showcase our optimization approach

Formalization: local asymptotic normality

- For measurement noise s^2, define sequential Gaussian experiment

$$G_t \mid G_{0:t-1} \sim N\left(\pi_t \cdot \theta^\star, \text{diag}(\pi_t \cdot s^2)\right)$$

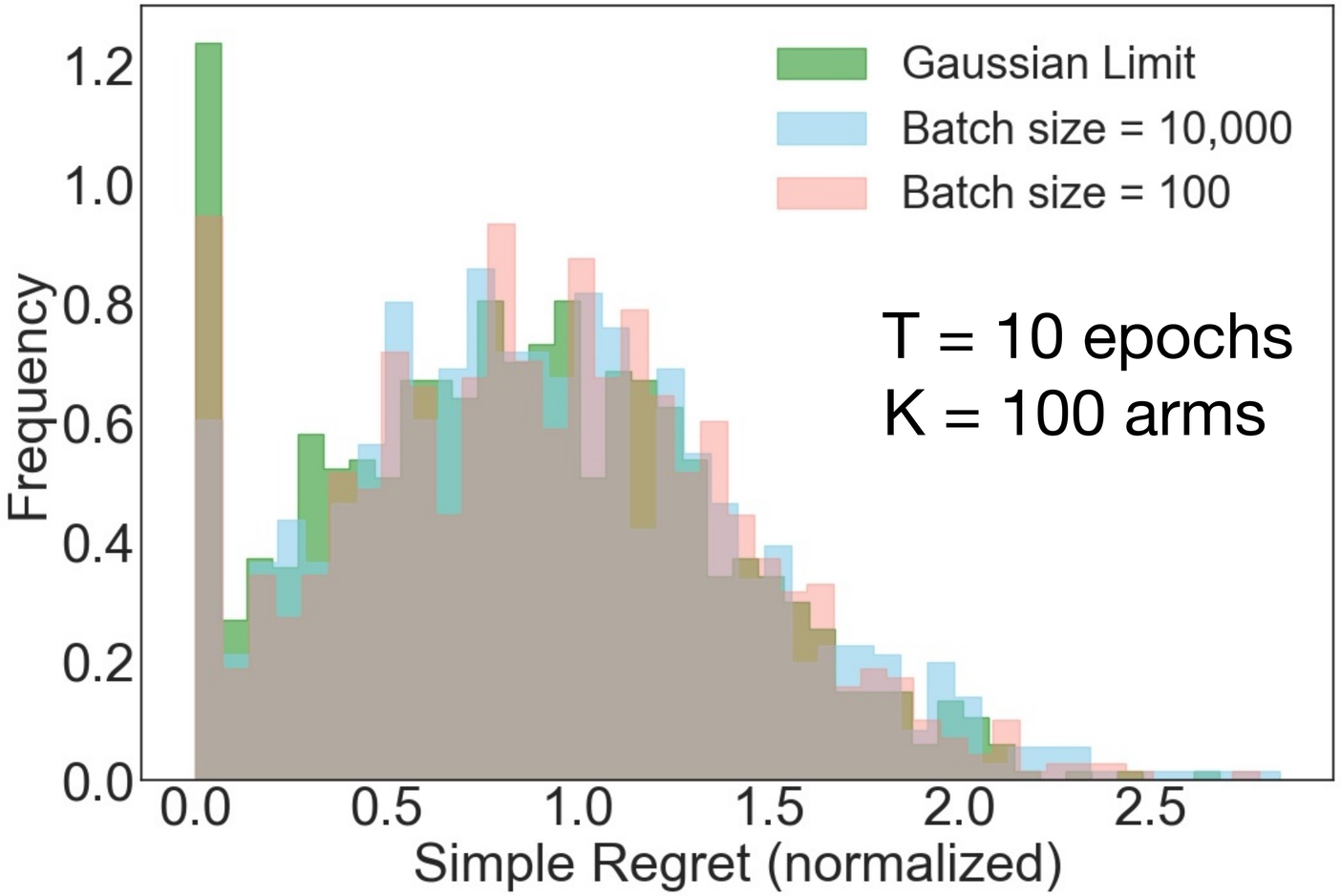
Theorem (Che & N. '23) If π's is continuous is batch statistics,

$$\left(\sqrt{n}\bar{R}_0, \ldots, \sqrt{n}\bar{R}_{T-1}\right) \Rightarrow (G_0, \ldots, G_{T-1})$$

We don't impose any assumption on the magnitude of π_t (big gap with best result in the literature).

This result significantly expands the scope of normal approximations adaptive settings.

Empirical Validity



Normal approximation reasonable even for small batch sizes!

Proof based on Stein's method

Corollary L: Lip. const. of policy π_t

Metrize weak convergence using bounded 1-Lipschitz functions. Then,

$$\mathrm{dist}\left(\sqrt{n}\bar{R}_{0:T-1}, G_{0:T-1}\right) \lesssim L^T n^{-1/6}$$

- No assumption on the magnitude of π_t
  - If π_t uniformly lower bounded, our proof gives standard $O(n^{-1/2})$-bound
- Despite empirics, conservative convergence rates
  - Nevertheless, usually $T \lll n$ in online platforms

Residual Horizon Optimization

- At every epoch, given posterior state (β, Σ), solve for the optimal **static** sampling allocations
- Resolve every batch, based on new information

$$\text{minimize}_{\pi_t(\beta_t,\Sigma_t)} \quad \mathbb{E}\left[\sum_{t=s}^{T} \text{Objective}_t(\pi_t, \beta_t, \Sigma_t) \mid \beta_s, \Sigma_s\right]$$

$$\text{subject to} \qquad \pi_t(\beta_t, \Sigma_t) \in \text{Simplex}$$

Residual Horizon Optimization

- At every epoch, given posterior state (β, Σ), solve for the optimal **static** sampling allocations
- Resolve every batch, based on new information

Constants

$$\text{minimize}_{\pi_t(\beta_t, \Sigma_t)} \quad \mathbb{E}\left[\sum_{t=s}^{T} \text{Objective}_t(\pi_t, \beta_t, \Sigma_t) \mid \beta_s, \Sigma_s\right]$$

$$\text{subject to} \qquad \pi_t(\beta_t, \Sigma_t) \in \text{Simplex}$$

Residual Horizon Optimization

- At every epoch, given posterior state (β, Σ), solve for the optimal **static** sampling allocations
- Resolve every batch, based on new information

$$\text{minimize}_{\pi_t} \quad \mathbb{E}\left[\sum_{t=s}^{T} \text{Objective}_t(\pi_t, \beta_t, \Sigma_t) \mid \beta_s, \Sigma_s\right]$$

$$\text{subject to} \qquad \pi_t \in \text{Simplex}$$

Residual Horizon Optimization

$$\text{minimize}_{\pi_t} \quad \mathbb{E}\left[\sum_{t=s}^{T} \text{Objective}_t(\pi_t, \beta_t, \Sigma_t) \mid \beta_s, \Sigma_s\right]$$

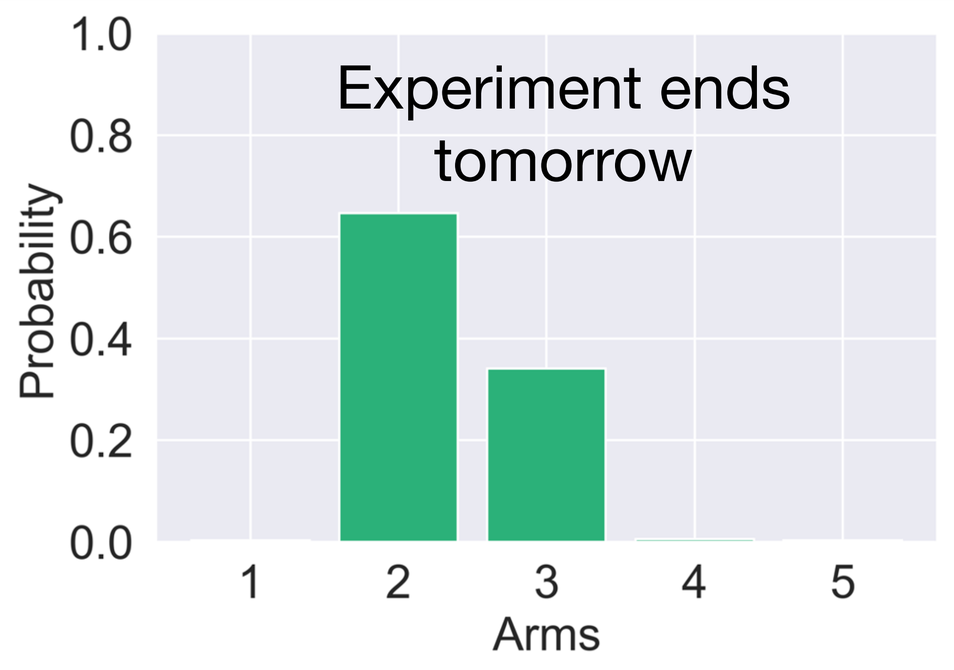
$$\text{subject to} \qquad \pi_t \in \text{Simplex}$$

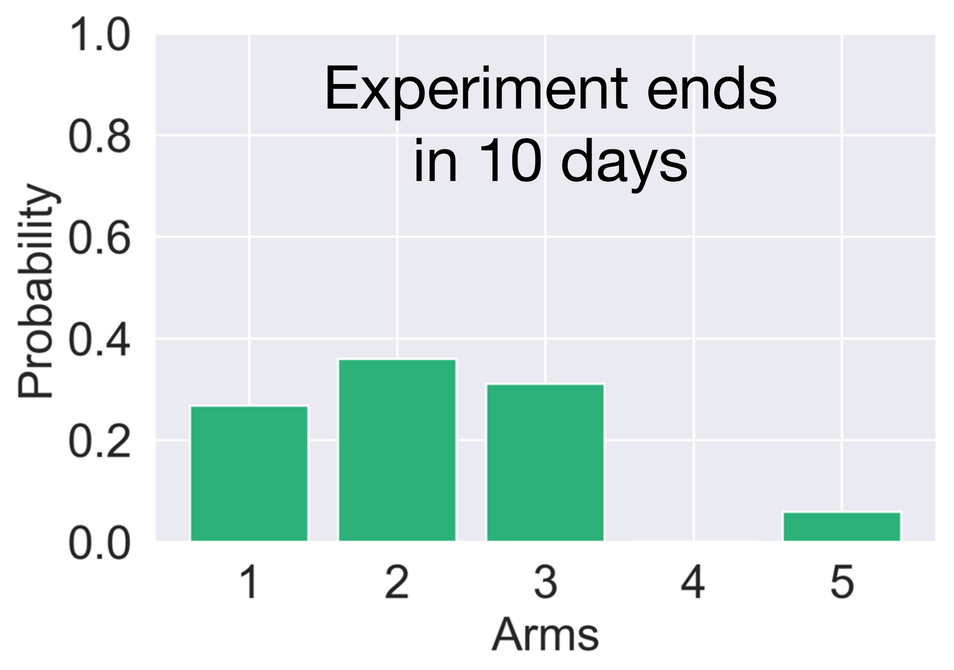
- Closed-form dynamics means (β_t, Σ_t) can be expressed explicitly
- Use stochastic gradients to optimize allocations!

PyTorch

Residual Horizon Optimization

Why planning? Calibrate exploration to horizon





Algo Design Principle

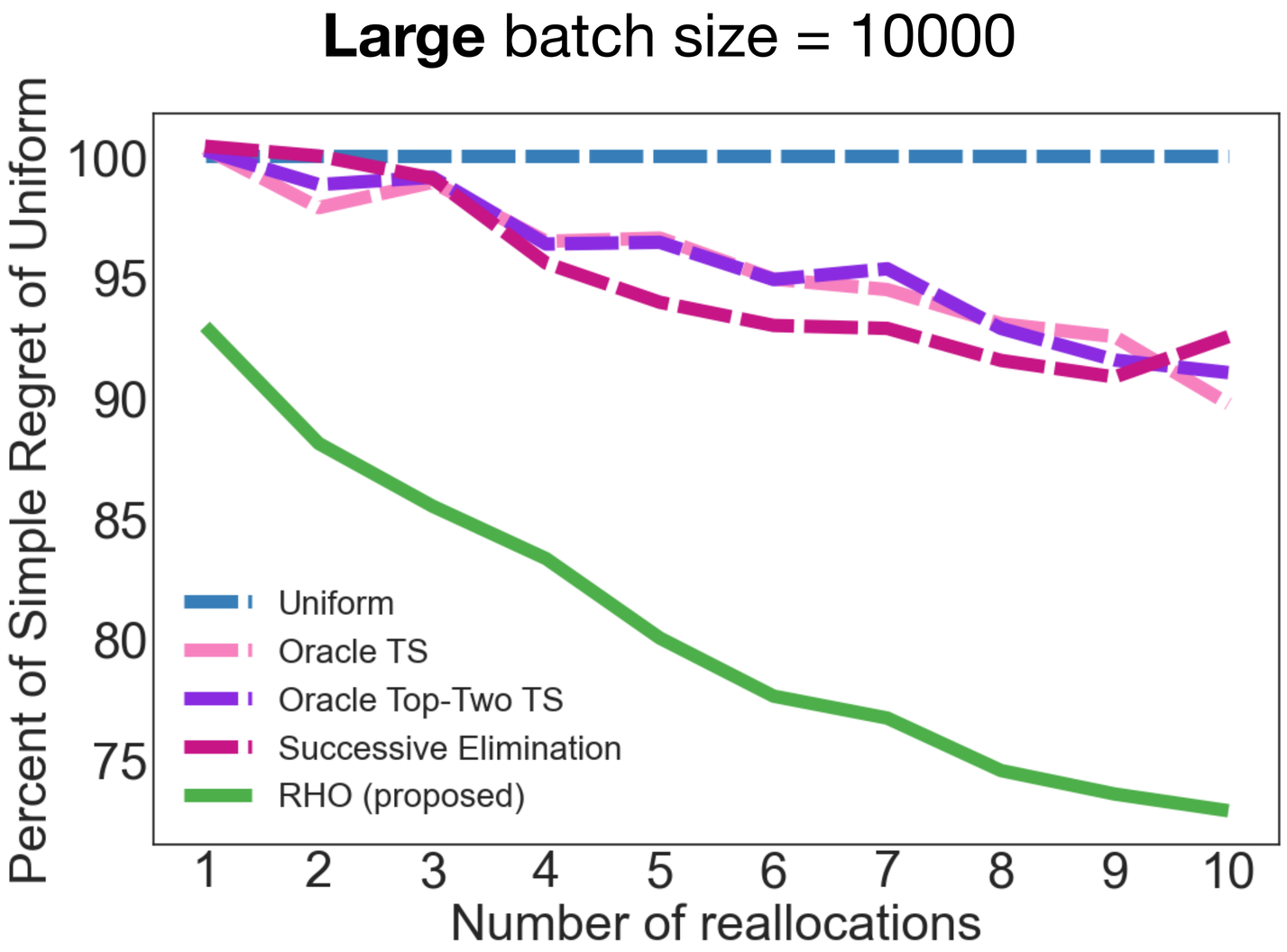
Theorem: RHO outperforms *any* static policy (including A/B tests)

- For any time horizon T
- For any constraints
- For any objective
- For any time non-stationarity

Why? The algorithm is **Policy Iteration on Static Designs**

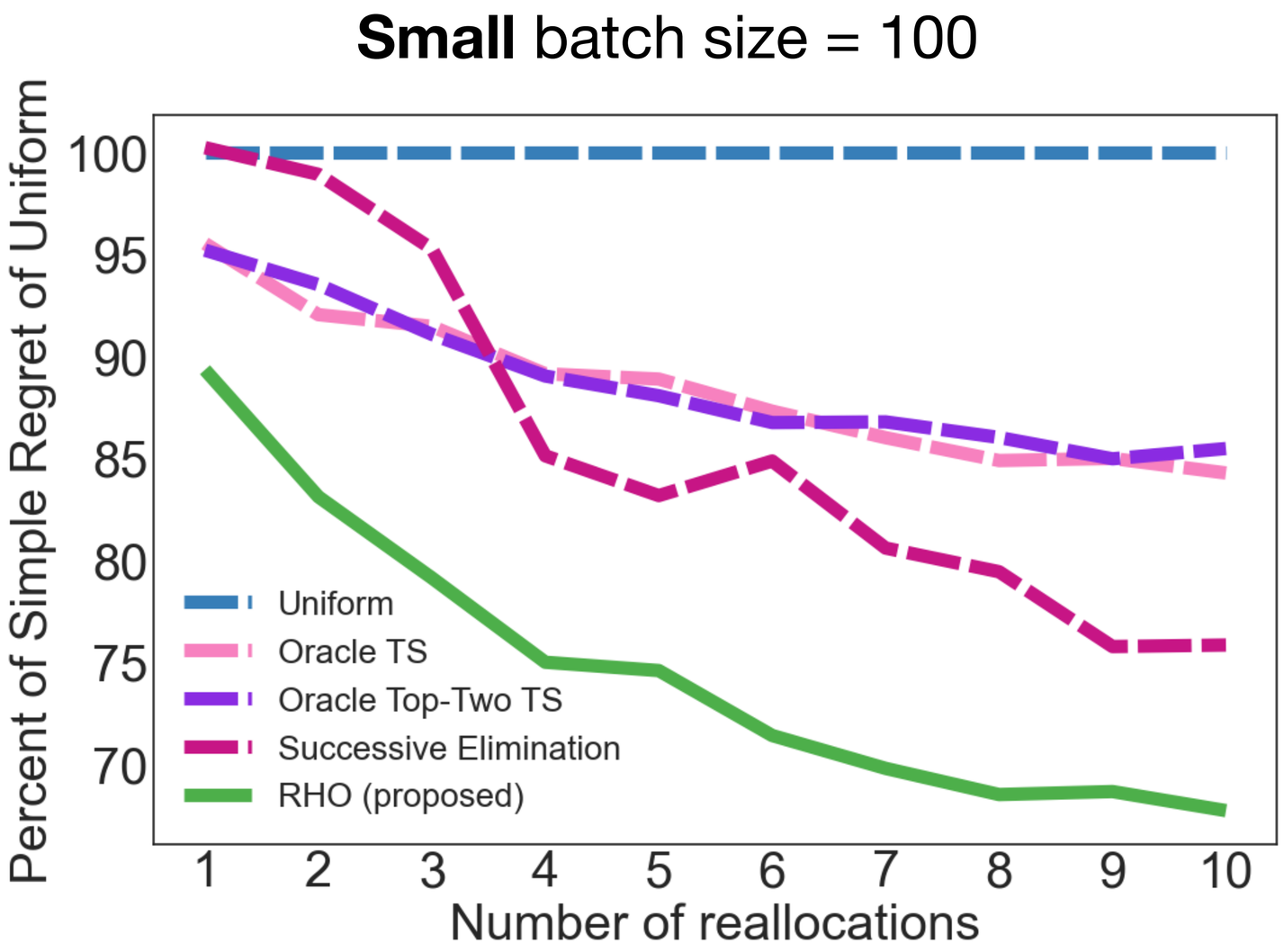
Simple non-contextual example

arms = 100



Simple non-contextual example

arms = 100



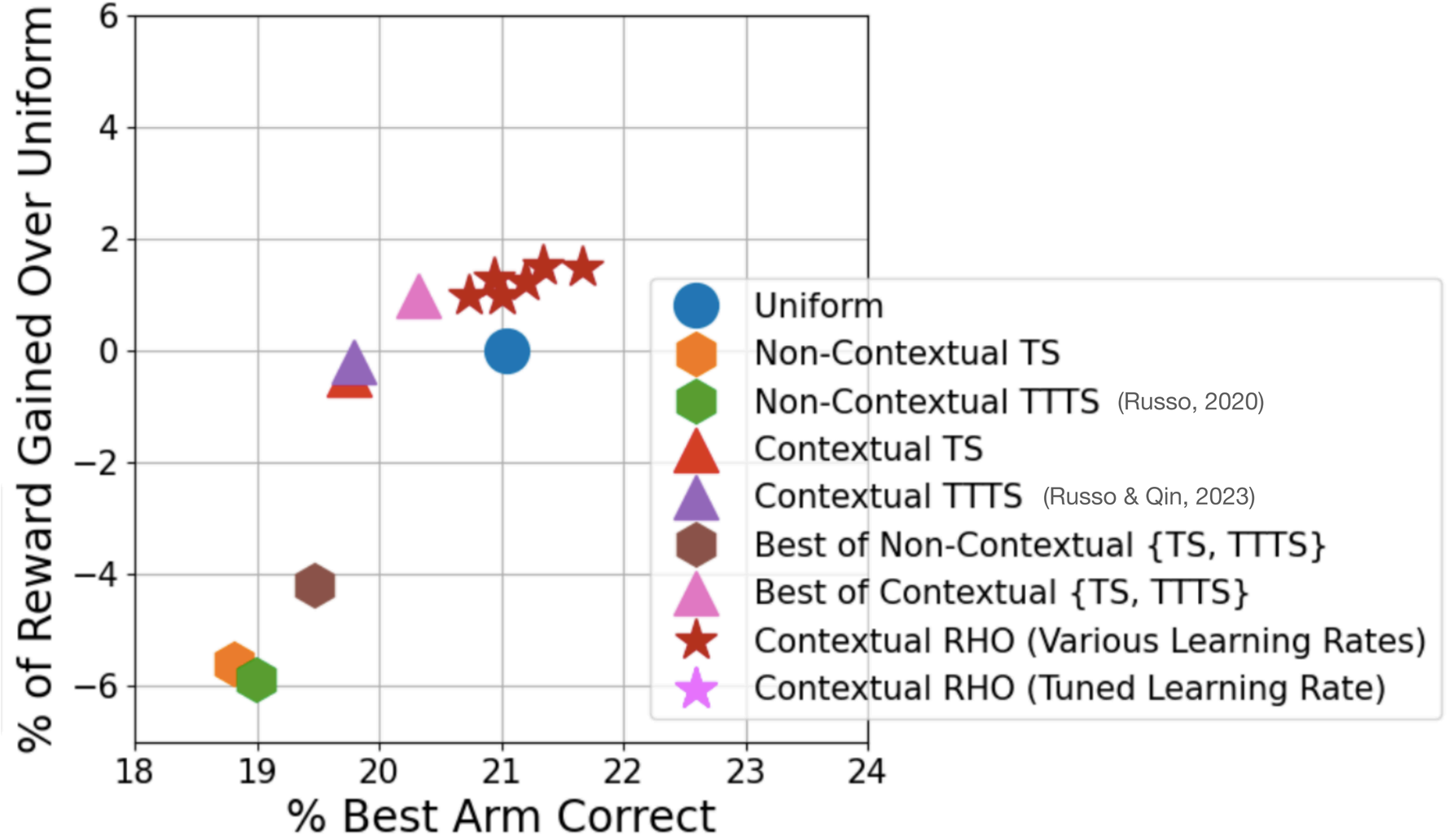
Back to non-stationarity

Benchmarking results over 180K different instances

Contextual = model time-varying trends

Batch size = 100K

Horizon T = 10



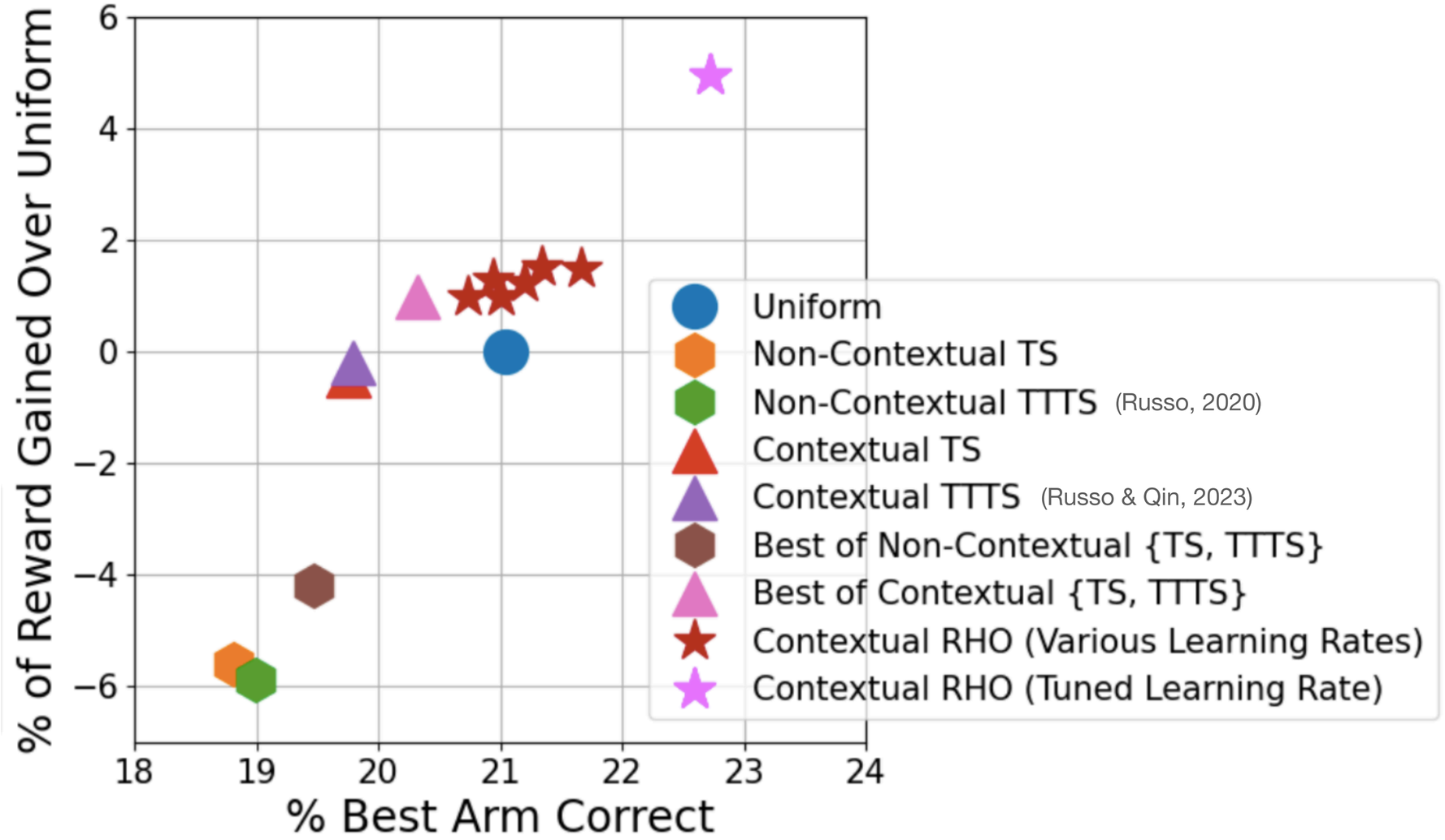
Back to non-stationarity

Benchmarking results over 180K different instances

Contextual = model time-varying trends

Batch size = 100K

Horizon T = 10



Encoding different objectives

- Imagine social platform tuning weights on clicks vs. likes vs. shares

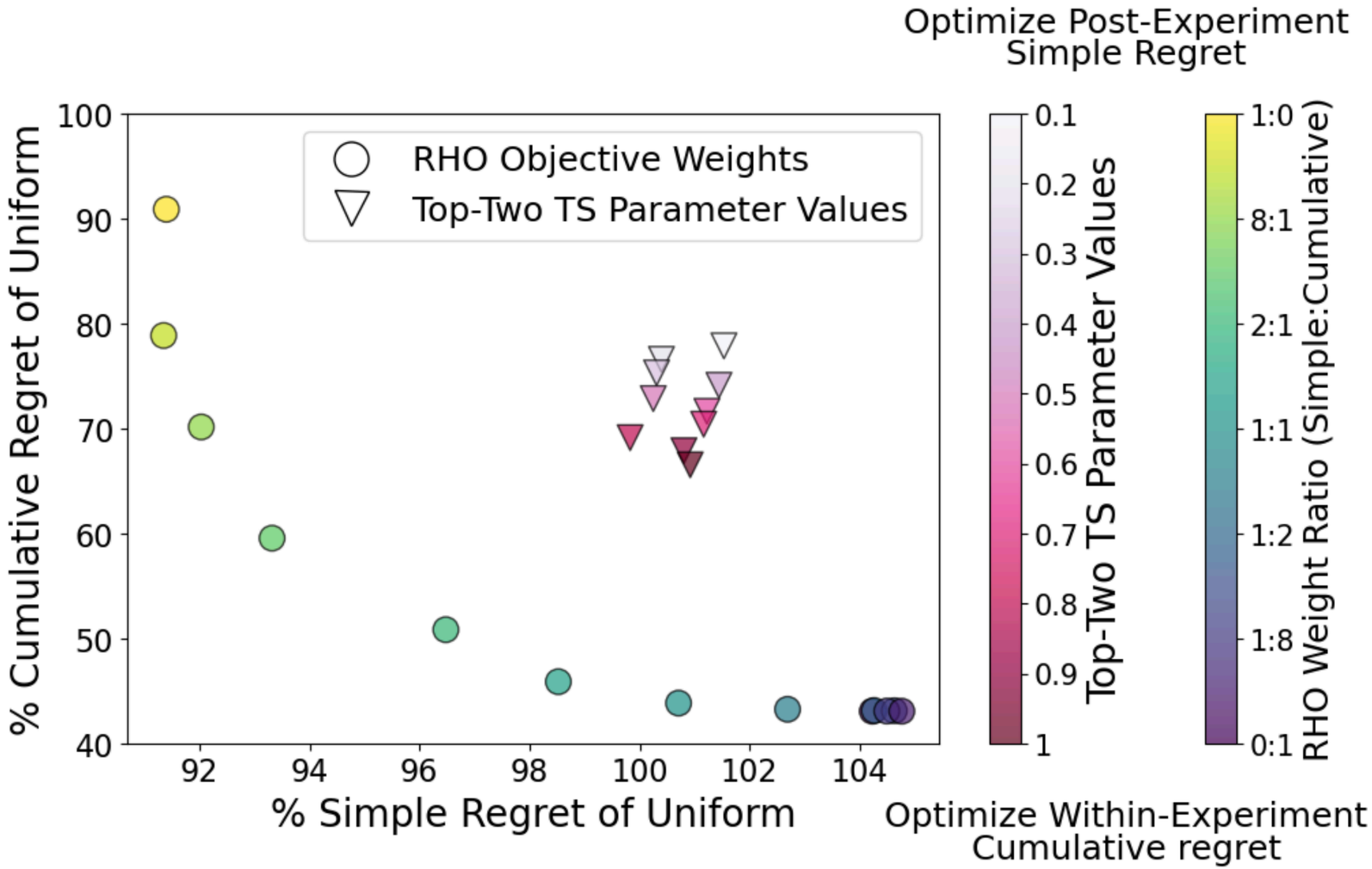
$$\text{minimize}_{\pi_t} \quad \mathbb{E}\left[\sum_{t=0}^{T-1} \text{Within-exp Rewards}_t(\pi_t, \beta_t, \Sigma_t) + \lambda \cdot \text{Post-exp Rewards}(\pi_T, \beta_T, \Sigma_T)\right]$$

- Natural candidate for λ: # in experiment / # affected by treatment

- Unlike TS-based policies, easy to balance within-experiment (simple) vs. post-experiment (cumulative) regret

Encoding different objectives

Batch size n = 100, Horizon T = 5



Applications at Netflix

by Ethan Che (I had nothing to do with it)

- Artwork personalized for each user
- New movies? Requires **exploration** to learn (ϵ-greedy).
- How should the **exploration rate** be calibrated across a limited horizon (think 7 days)?

NETFLIX

Applications at Allegheny County (PA)

Given limited budget, how do we allocate resources?

- 7K people exit county jail each year; re-entry ~30%
- Outcomes: re-entry, multiple ED visits, involuntary psychiatric commitment, involvement in violence, shelter usage
- Interventions: cash transfer, jobs program, CBT
- Status quo: risk score-based allocation

CLT for adaptive designs

- Normal approximations => tractable optimization formulation for AEx
- Flexibly handles batches, objectives, constraints, and non-stationarity
  - Unlike other heuristics (e.g., TS), reliably outperform A/B tests
- Empirical benchmarking can derive methodological progress!

aes-batch.streamlit.app github.com/namkoong-lab/aexgym

Optimization-Driven Adaptive Experimentation, with E. Che, D. Jiang, J. Wang

Adaptive Experimentation at Scale: A Computational Framework for Flexible Batches, with E. Che, Major Revision in Operations Research

AExGym: Benchmarks and Environments for Adaptive Experimentation, with J. Wang, E. Che, D. Jiang