ORF 418 Special Topic

The "Ad-click" Game

Apr 15, 2019

Donghun Lee

Ph.D. Candidate,
Dept. of Computer Science

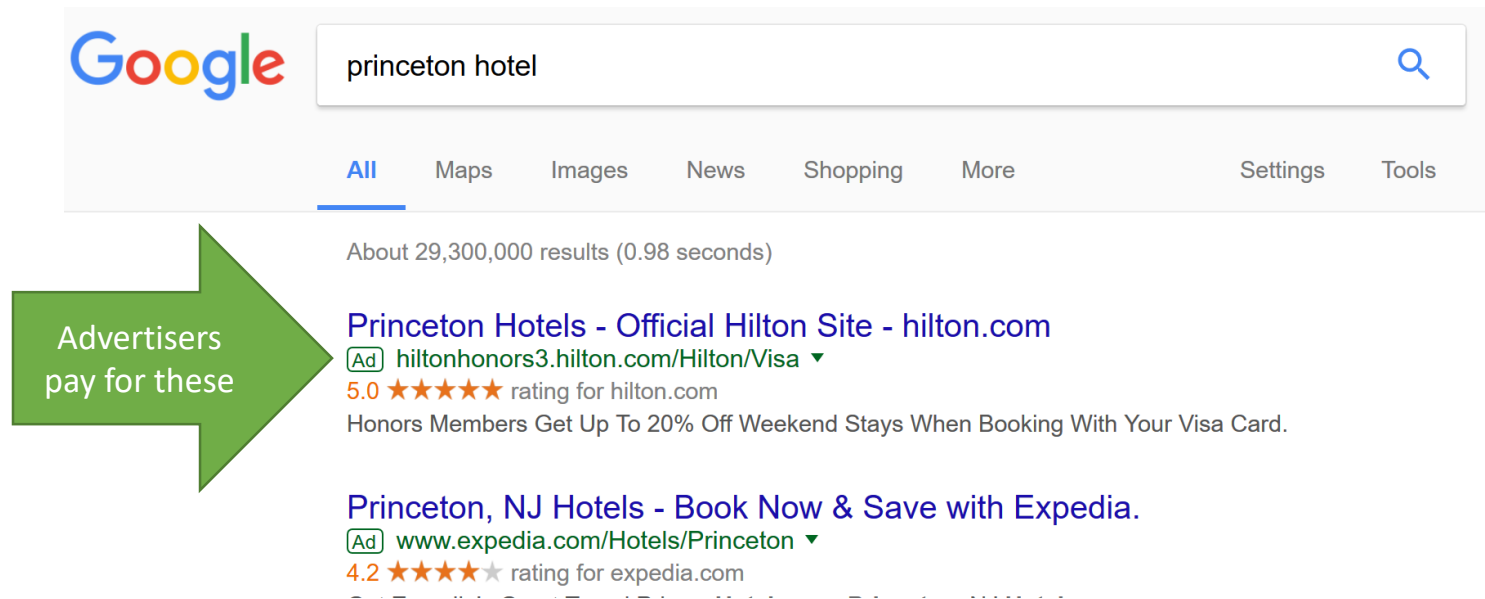PRINCETON
UNIVERSITY

Contents

- Intro to Ad-click Game (Behind the SSAs)

- Simulator Operation

- Attributes in SSA: Google's Twist

Intro to Ad-click Game

Behind the sponsored search auctions

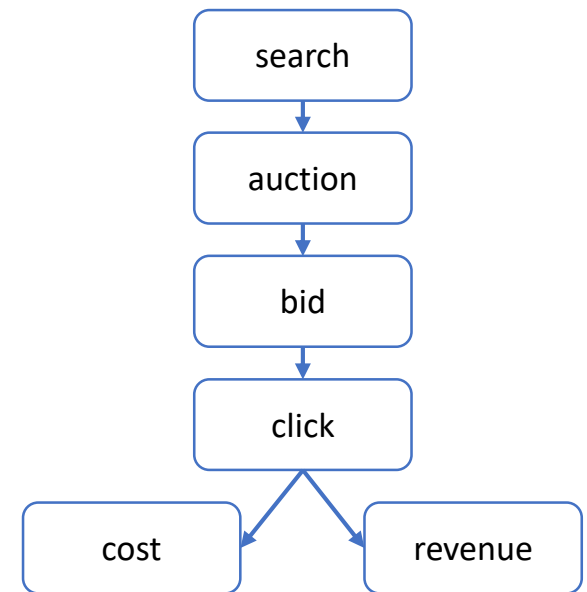# Sponsored Search Auction (SSA)

- Sponsored search auction result

Behind the SSA

```
search
  ↓
auction
  ↓
bid
  ↓
click
 ↙   ↘
cost   revenue
```

PRINCETON
UNIVERSITY

Behind the SSA

- When someone sends in a search query
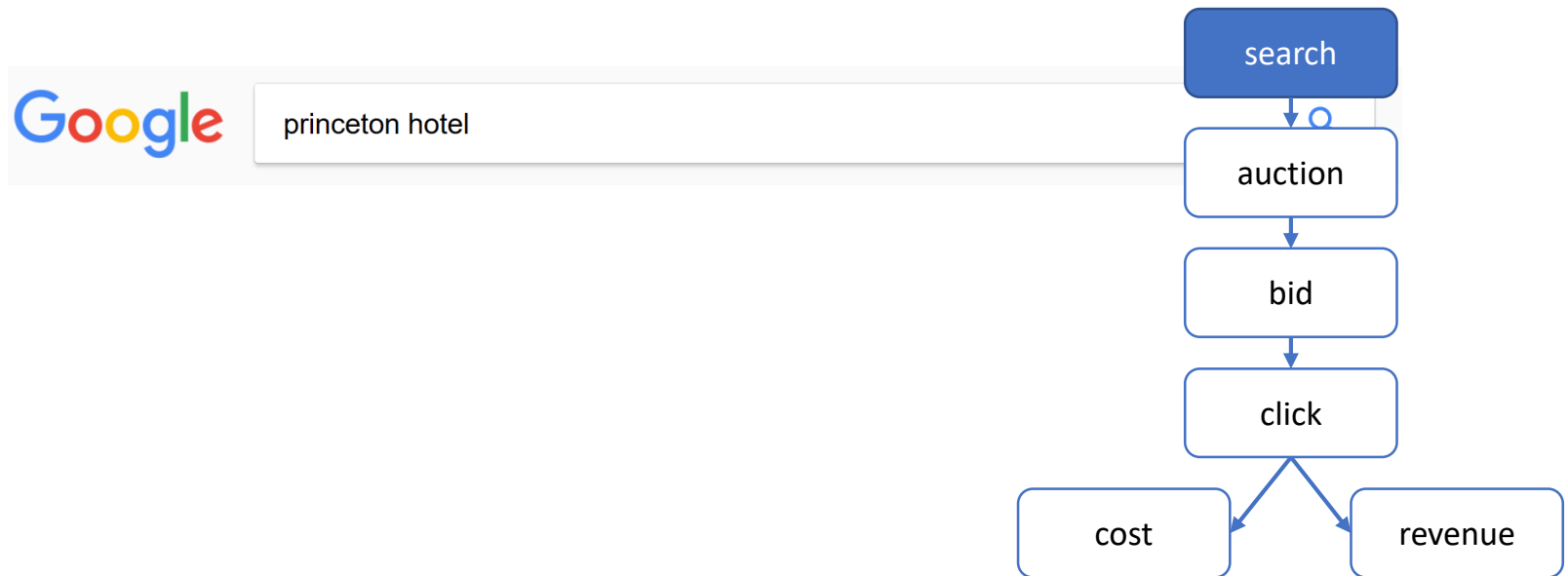
Behind the SSA

- Auction happens behind the curtain

Google   princeton hotel

- Advertisers bid to get their ads placed

search

auction

bid

click

cost   revenue

PRINCETON
UNIVERSITY
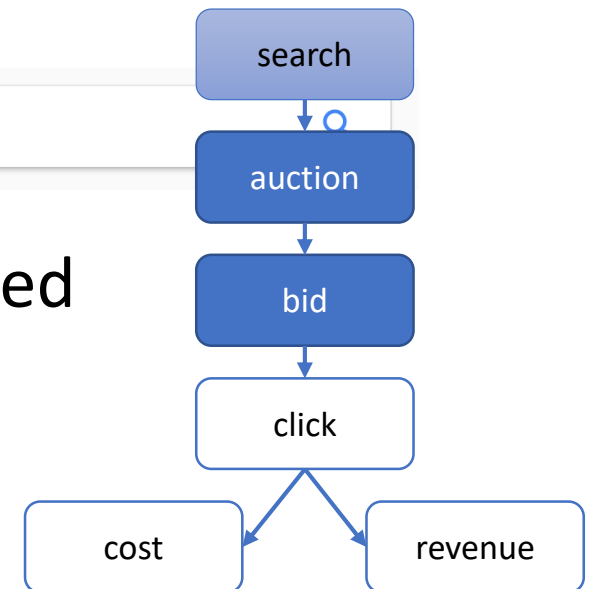
Behind the SSA

- The ads from the winners appear in the result
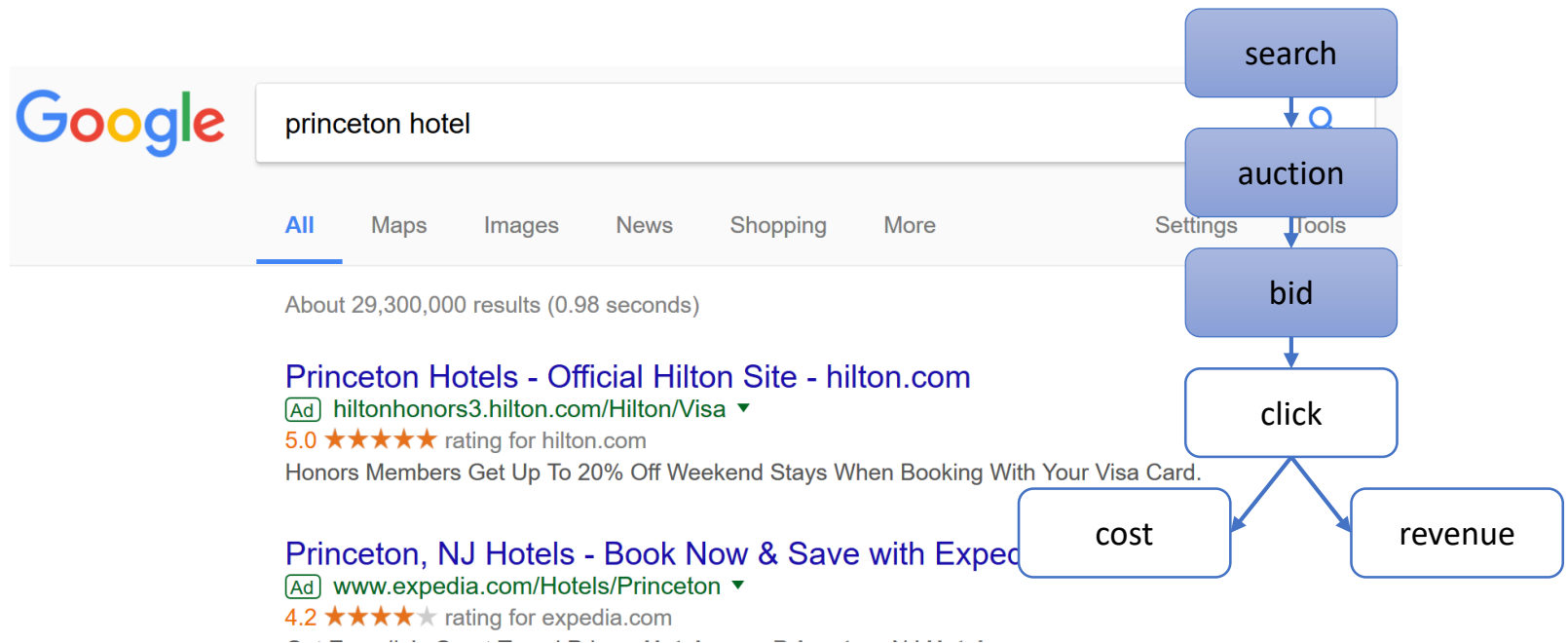
Behind the SSA

- You may click on an ad
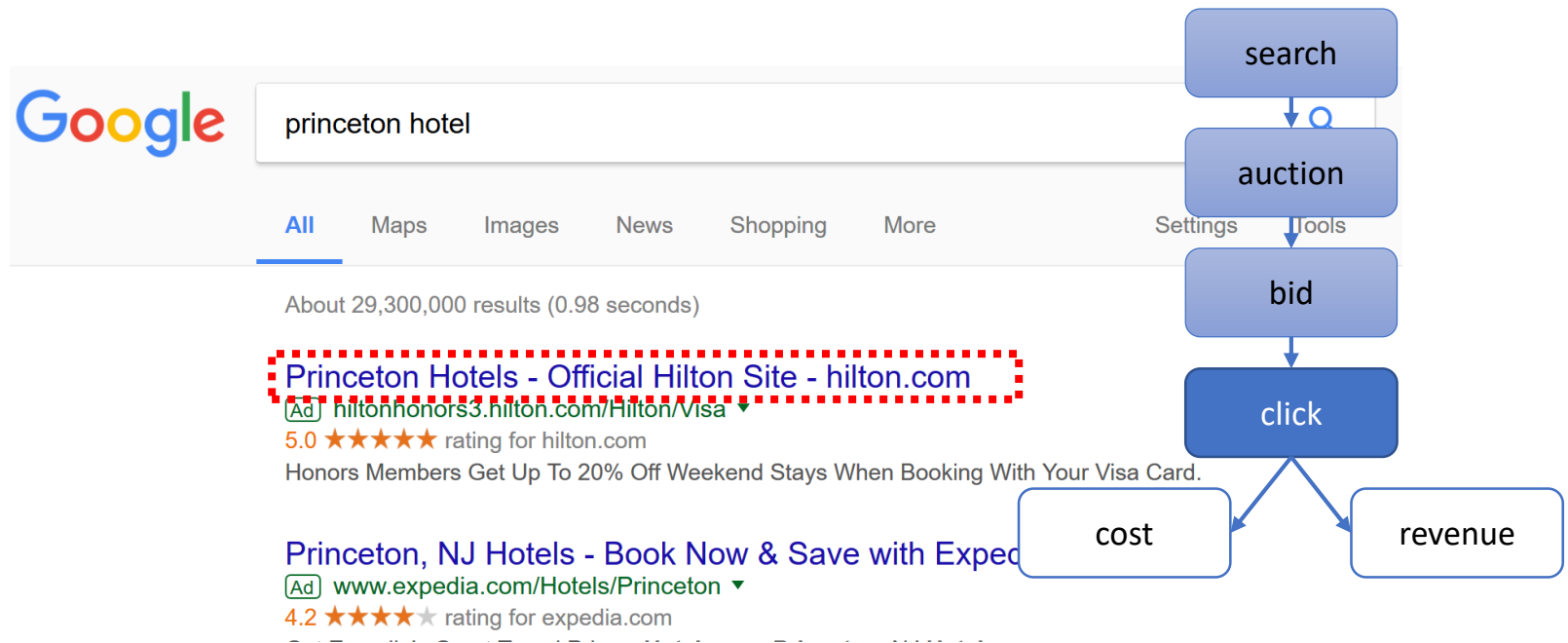
Behind the SSA

- Then that advertiser pay for your click

- Some revenue may go to that advertiser
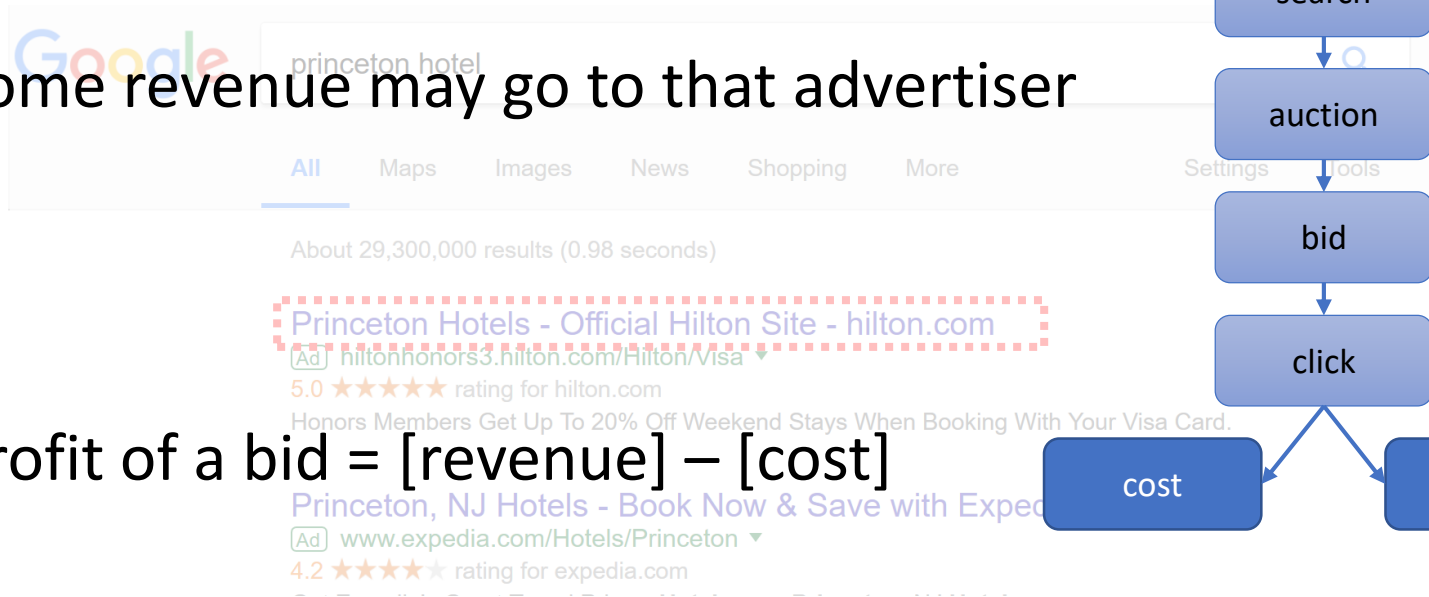
- Profit of a bid = [revenue] − [cost]

search

auction

bid

click

cost

revenue

Behind the SSA

- Goal: maximize profit (from advertising)
  - Profit = [revenue] – [cost]

- Find the policy
  - that maximizes profit
  - by bidding to advertise in the SSA
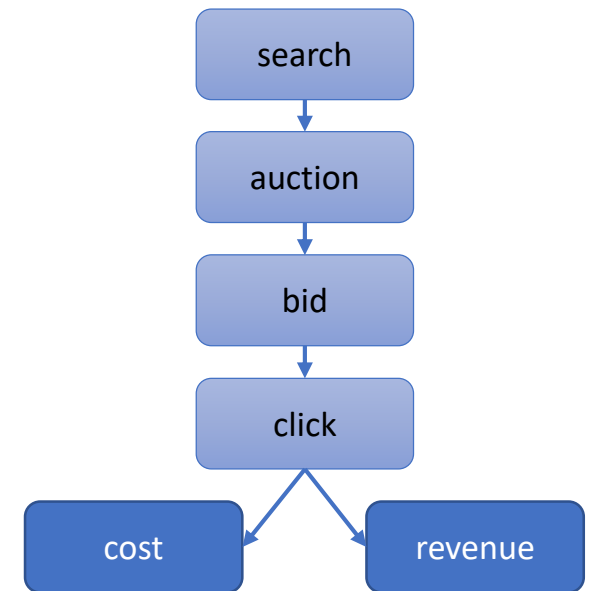
Behind the SSA

- Goal: maximize profit (from advertising)
  - Profit = [revenue] − [cost]

- Find the policy
  - that maximizes profit
  - by bidding to advertise in the SSA

- Simulated in the "Ad-click" game

```
search
  ↓
auction
  ↓
 bid
  ↓
click
 ↙  ↘
cost   revenue
```

PRINCETON UNIVERSITY
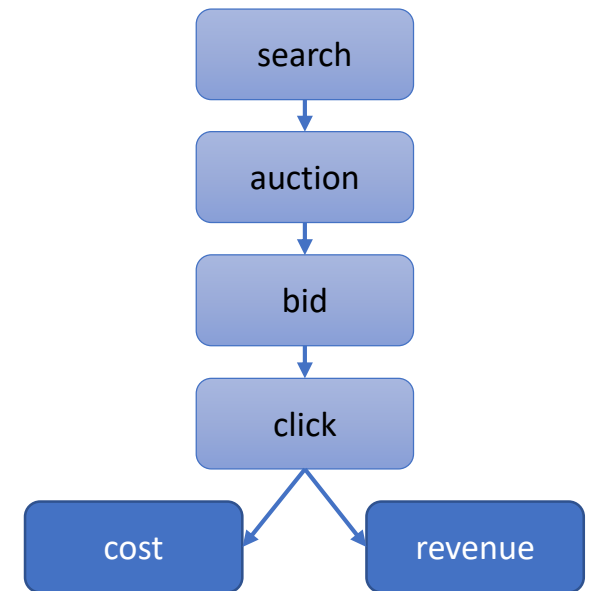
The Simulator Operation

From the Policy's Viewpoint, on the Simulator ver. 2

Simulator Operation

- In every iteration, the simulator
  - Gathers bids from all policies
  - Generates searches
    - And holds an auction for each search
  - For the winning policies
    - Randomly sample clicks and conversions
    - Compute costs and revenues
  - For all policies
    - Report individualized results

```
search
  ↓
auction
  ↓
bid
  ↓
click
 ↙  ↘
cost  revenue
```

Simulator Operation

- In every iteration, the simulator
  - Gathers bids from all policies
  - Generates searches
    - And holds an auction for each search
  - For the winning policies
    - Randomly sample clicks and conversions
    - Compute costs and revenues
  - For all policies
    - Report individualized results

```
search
  ↓
auction
  ↓
bid
  ↓
click
 ↙  ↘
cost  revenue
```

Simulator Operation

- In every iteration, the simulator
  - Gathers bids from all policies
  - Generates searches
    - And holds an auction for each search
  - For the winning policies
    - Randomly sample clicks and conversions
    - Compute costs and revenues
  - For all policies
    - Report individualized results

search

auction

bid

click
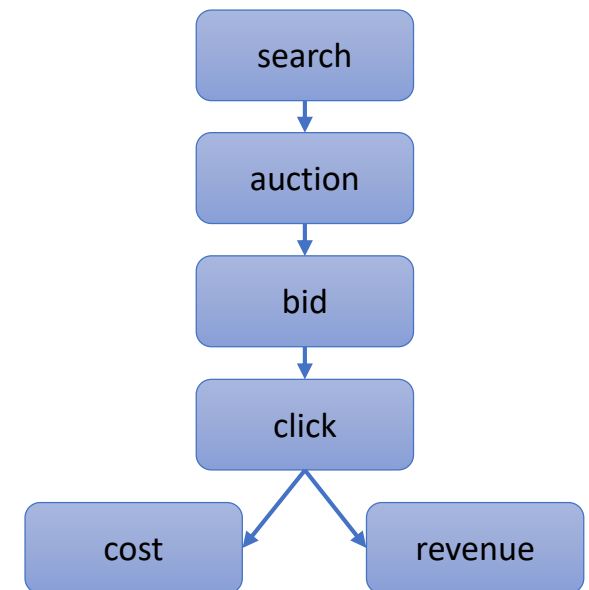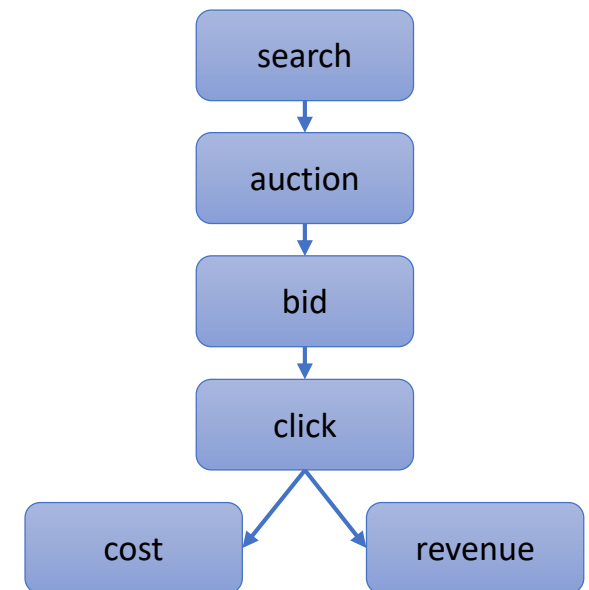
cost   revenue

Simulator Operation

- In every iteration, the simulator
  - Gathers bids from all policies
  - Generates searches
    - And holds an auction for each search
  - For the winning policies
    - Randomly sample clicks and conversions
    - Compute costs and revenues
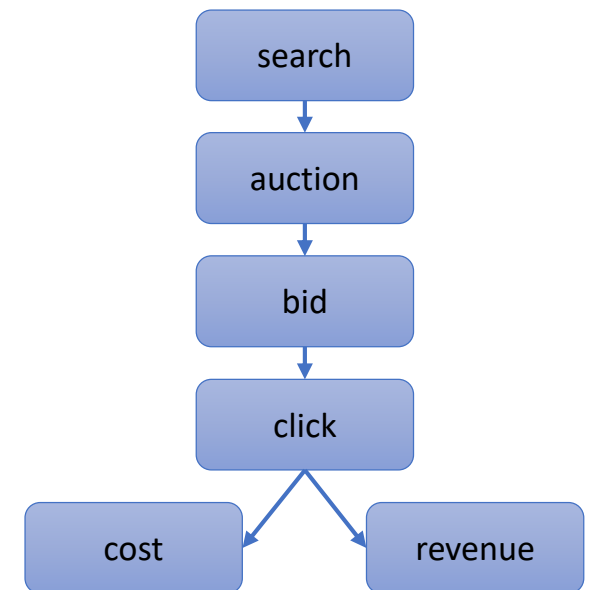 - For all policies
    - Report individualized results

- Replicate for each "attributes"

search → auction → bid → click → cost, revenue

PRINCETON UNIVERSITY

Simulator Operation: Policy's View

- In every iteration, each policy
    - Sends in the bids
    - Learns from the results

Search

Bid

Auction

Learn

next iter.

PRINCETON UNIVERSITY
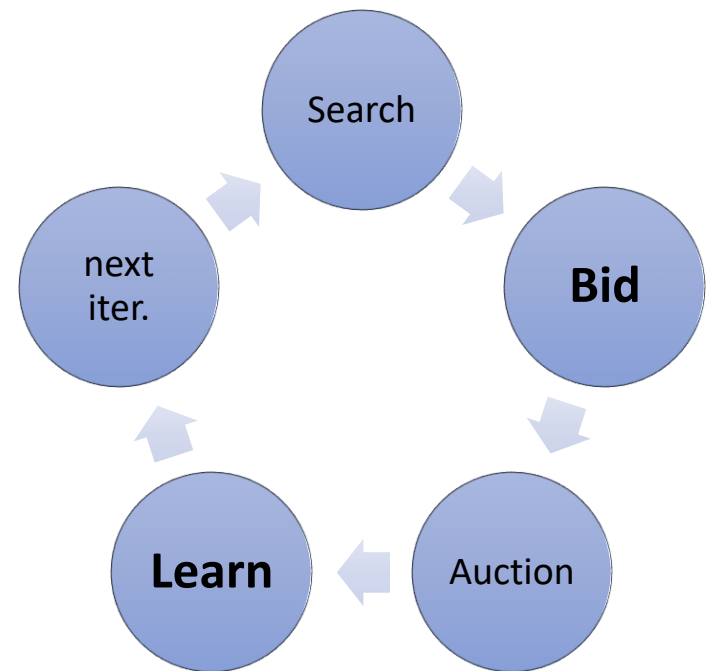
Policy.bid()

- In every iteration, each policy
  - **Sends in the bids**
  - Learns from the results

PRINCETON UNIVERSITY

Policy.learn()

- In every iteration, each policy
  - Sends in the bids
  - **Learns from the results**

Search

Bid

Auction

Learn

next iter.

PRINCETON UNIVERSITY

Simulator Operation: Policy's View

- In every iteration, each policy
  - Sends in the bids
  - Learns from the results

- Bids:
  - All "joint" attributes
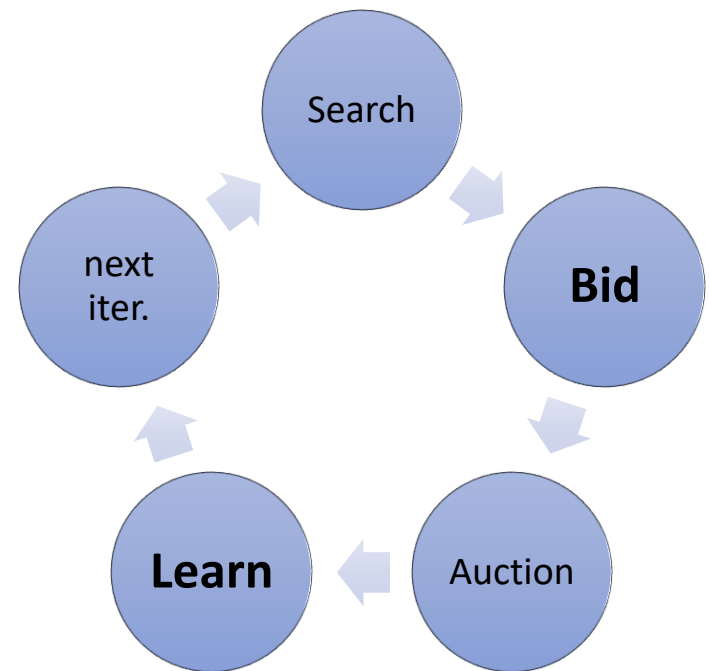
- Results:
  - Only "marginal" attributes

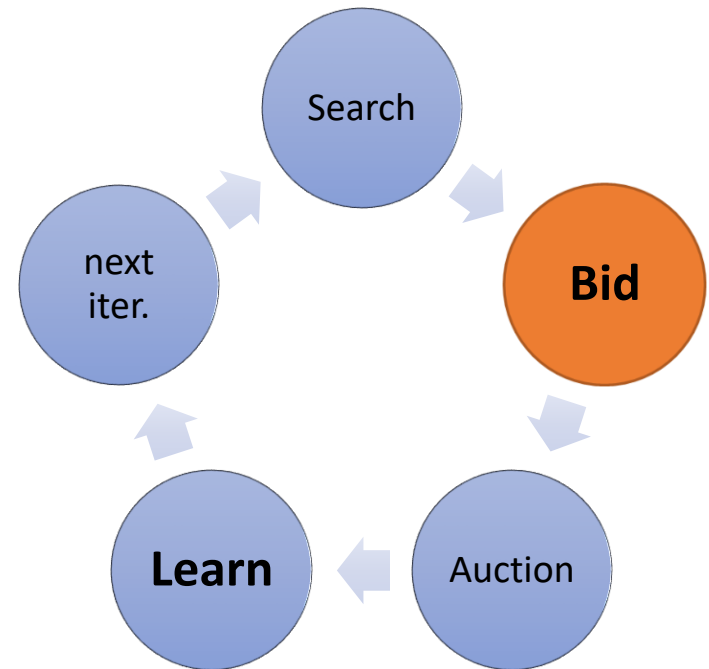Search

Bid

Auction

Learn

next iter.

PRINCETON UNIVERSITY

Simulator Operation: Policy's View

- In every iteration, each policy
    - Sends in the bids
    - Learns from the results

Google's Twist
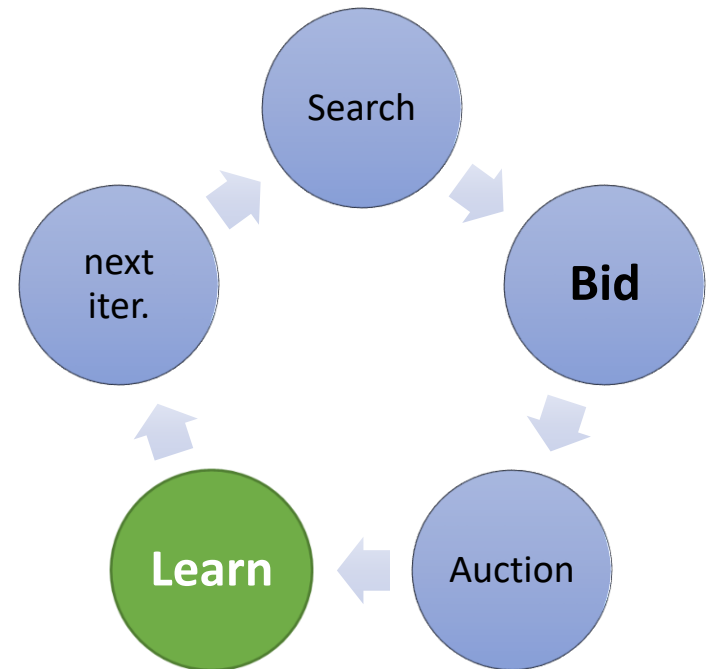- Bids:
    - All "joint" attributes
- Results:
    - Only "marginal" attributes
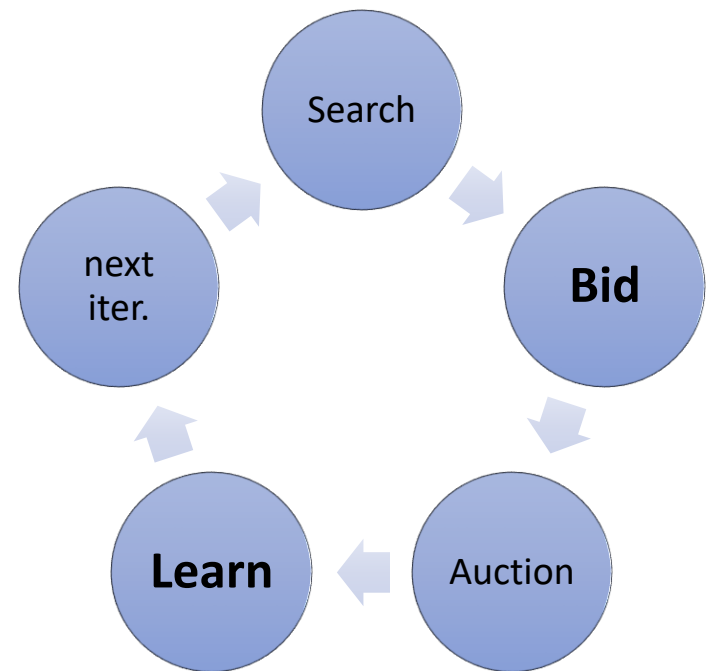
Attributes in SSA

Discrepancy between the Joint and the Marginals

Attributes in SSA

- Example
    - Gender: {M, F, U}
    - Age: {0-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-}

Attributes in SSA

- Example
  - Gender: {M, F, U}
  - Age: {0-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-}

- "Attributes" of a user
  - who does the web search
    - which initiates the sponsored search auction (SSA)

  - Me: {gender: M, age: 30-39}

PRINCETON UNIVERSITY

Simulator Operation, revisited

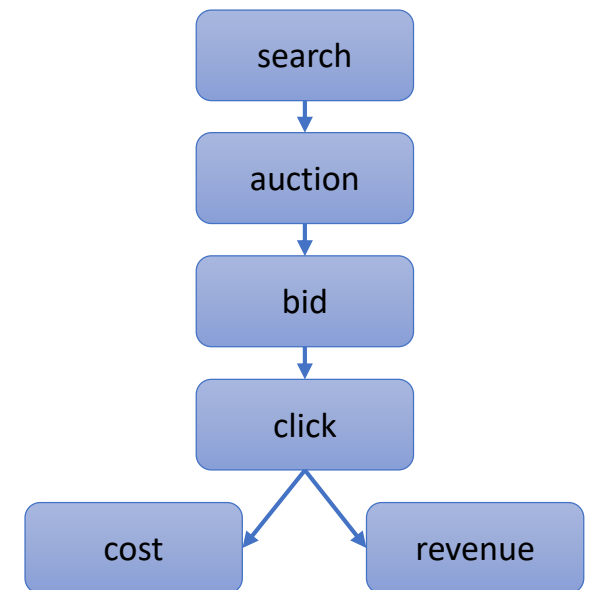- In every iteration, the simulator
  - Gathers bids from all policies
  - Generates searches
    - And holds an auction for each search
  - For the winning policies
    - Randomly sample clicks and conversions
    - Compute costs and revenues
  - For all policies
    - Report individualized results

```
search
  ↓
auction
  ↓
 bid
  ↓
click
 ↙   ↘
cost   revenue
```
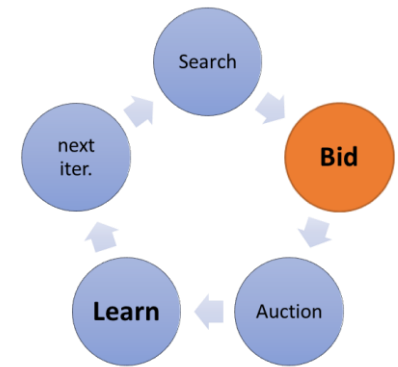
- Replicate for each "**attributes**"

Attributes in SSA

- Example
  - Gender: {M, F, U}
  - Age: {0-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-}

	0-19	20-29	30-39	40-49	50-59	60-69	70-
M							
F							
U							

Attributes in SSA

- Example
  - Gender: {M, F, U}
  - Age: {0-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-}

	0-19	20-29	30-39	40-49	50-59	60-69	70-
M							
F			**Joint**				
U							

- Your policy can specify bids
  - for each of these attribute combinations (the entire joint)

Attributes in SSA

- Example
  - Gender: {M, F, U}
  - Age: {0-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-}

		Marginal						
		0-19	**20-29**	**30-39**	**40-49**	**50-59**	**60-69**	**70-**
Marginal	M							
	F							
	U							

- Your policy will get results
  - only along the marginals

Attributes in SSA

- Example
  - Gender: {M, F, U}
  - Age: {0-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-}

		Marginal						
		0-19	20-29	30-39	40-49	50-59	60-69	70-
Marginal	M							
	F			**Joint**				
	U							

- Discrepancy
  - Bid (Act): can be specified for the entire joint
  - Learn: only gets data along the marginals

Attributes in SSA

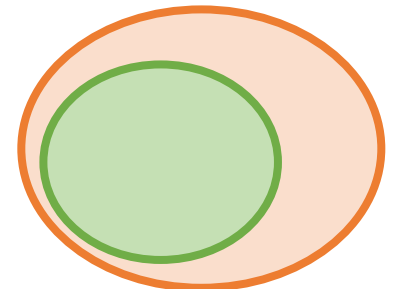- Example
  - Gender: {M, F, U}
  - Age: {0-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-}

Marginal							
	0-19	20-29	30-39	40-49	50-59	60-69	70-
M							
F			Joint				
U							

- Discrepancy
  - Bid (Act): can be specified for the entire joint
  - Learn: only gets data along the marginals

PRINCETON UNIVERSITY

Summary
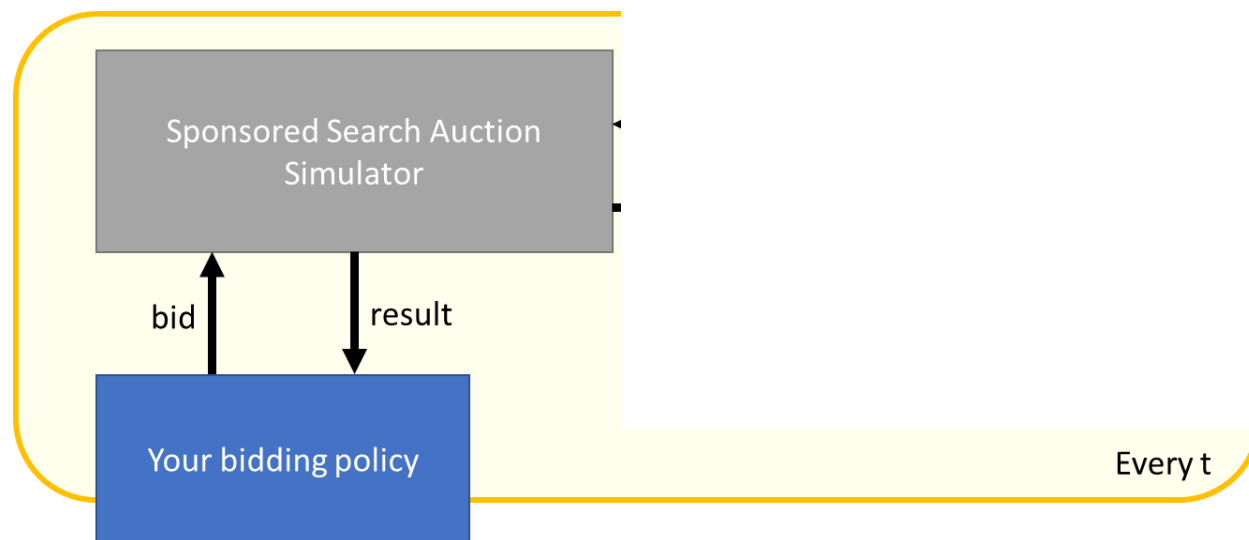
- The *new* ad-click simulator (2019)
  - Joint work with RoomSage Data Science team
  - Emulates Google's interface for SSA

- Mind the gap:
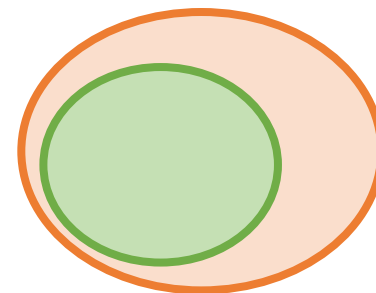 - result space (marginal) ⊆ bid space (joint)
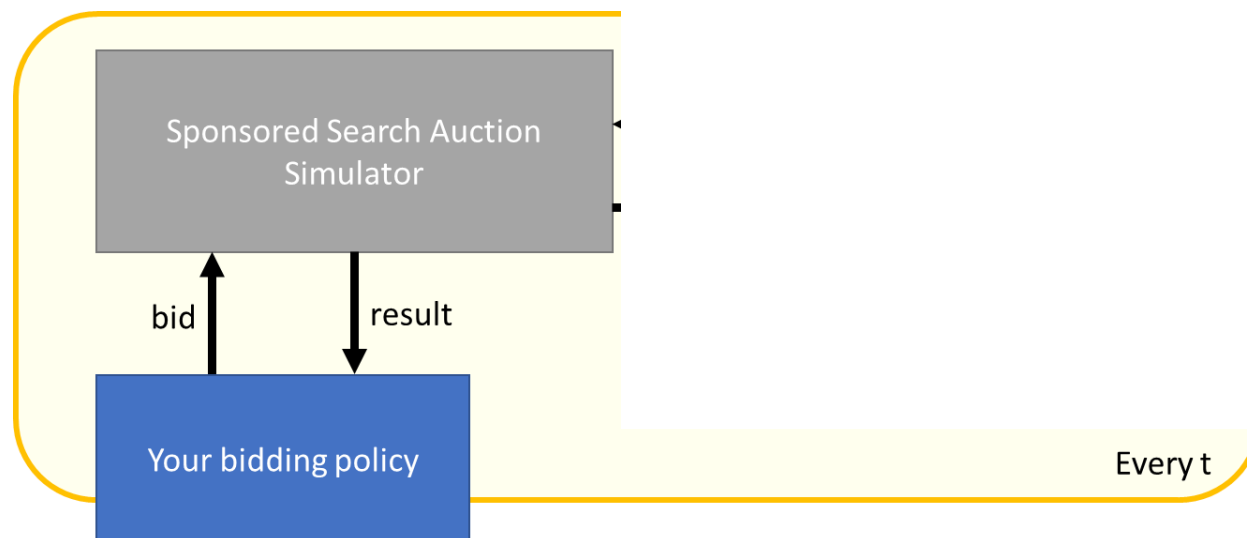
Summary

- Mind the gap:
  - result space (marginal) ⊆ bid space (joint)

Summary

- Mind the gap:
  - result space (marginal) ⊆ bid space (joint)

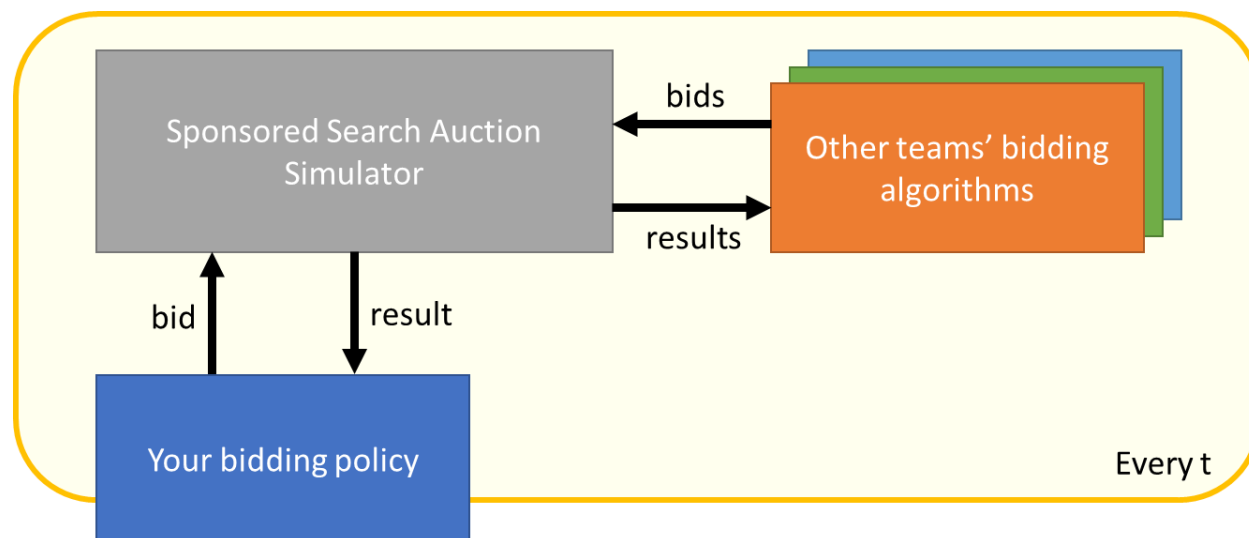# Summary

- The *new* ad-click simulator (2019)
  - Joint work with RoomSage Data Science team
  - Emulates Google's interface for SSA (Joint-Marginal gap)

  - Fully agent-based
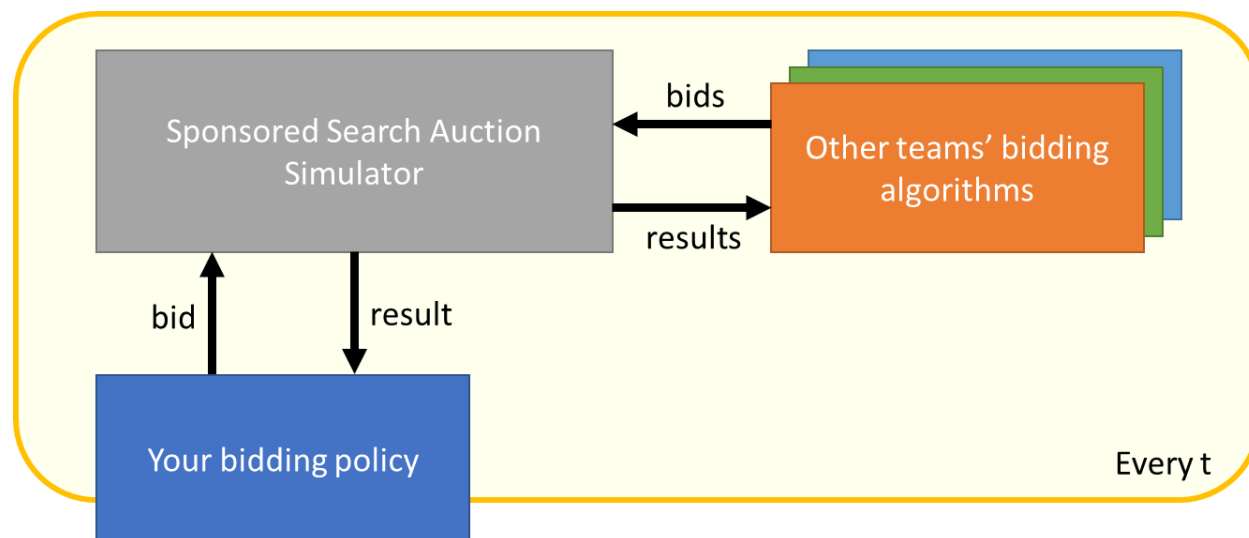 - Dynamic, transient, and potentially adversarial
    Inherited from the original ad-click simulator (2018)

PRINCETON UNIVERSITY
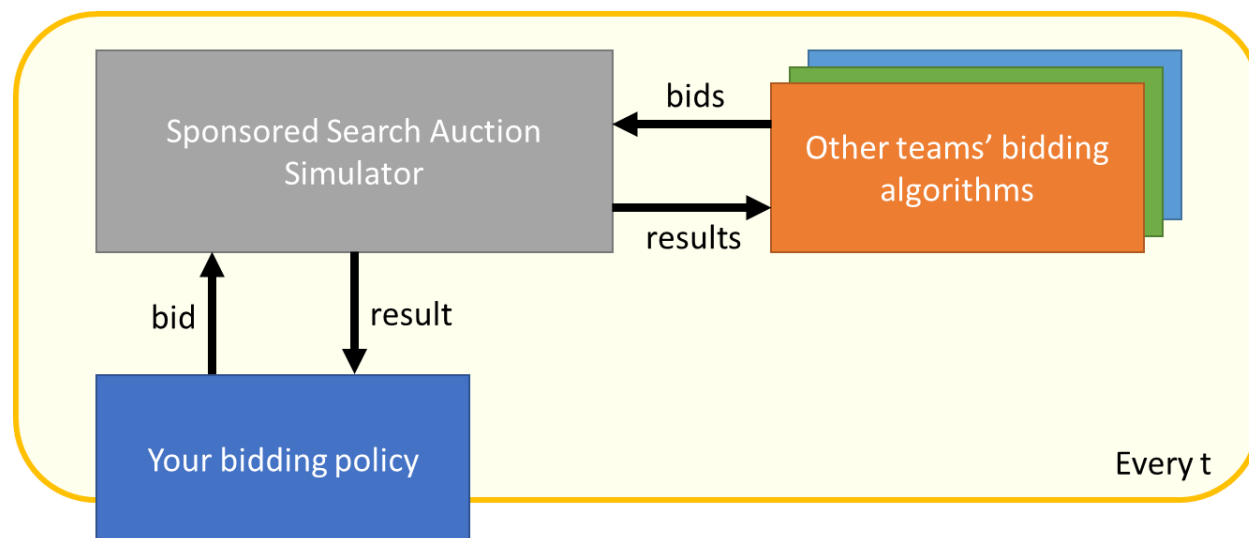
Summary

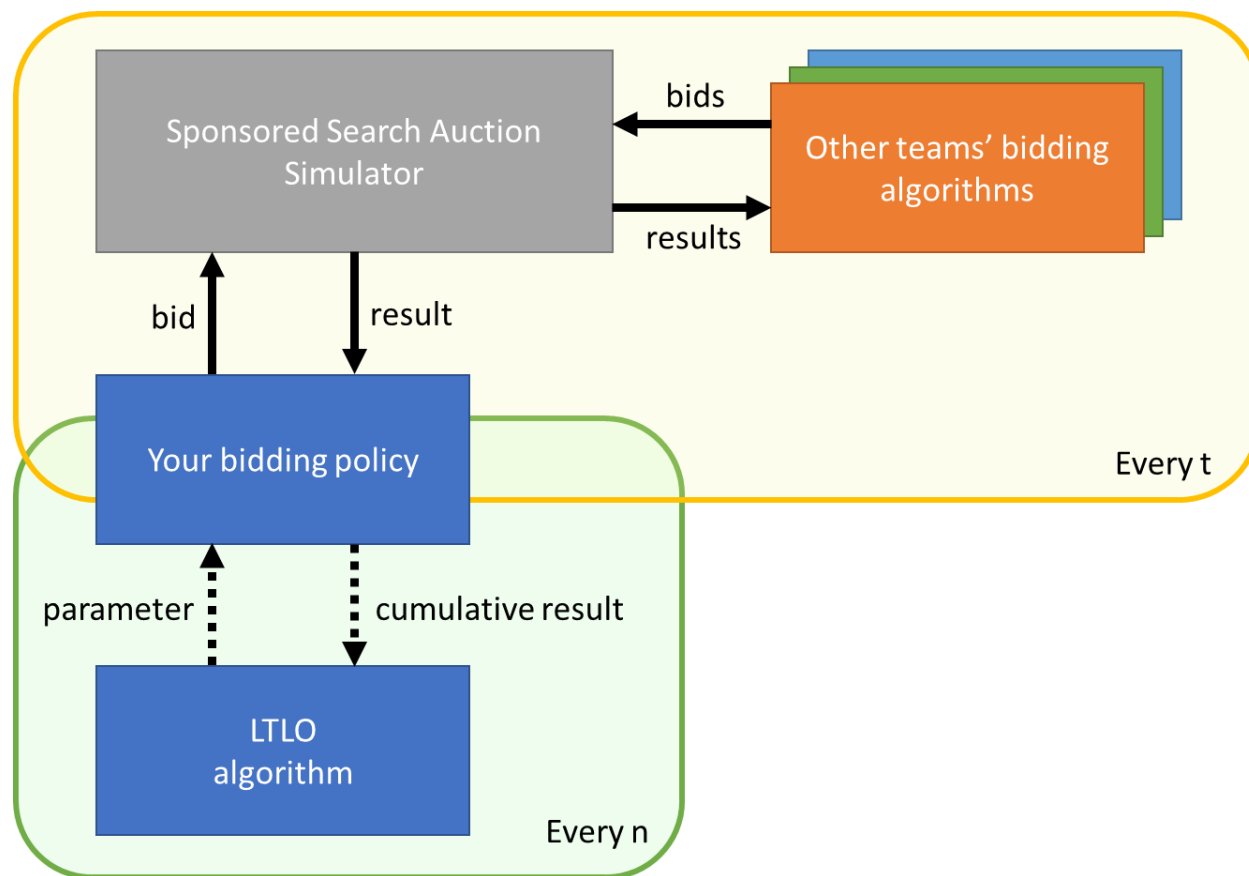- Your bidding policy competes

Summary

- Your bidding policy competes
  - against other policies

Summary

- Your bidding policy competes
  - against other policies
    - that may learn from the results as well

Summary

- This is how LTLO fits in the ad-click game

Summary

- The *new* ad-click simulator (2019)
    - Joint work with RoomSage Data Science team
    - Emulates Google's interface for SSA (Joint-Marginal gap)

    - Fully agent-based
    - Dynamic, transient, and potentially adversarial
      Inherited from the original ad-click simulator (2018)

    - Fully compatible with the LTLO framework

Questions?

PRINCETON UNIVERSITY