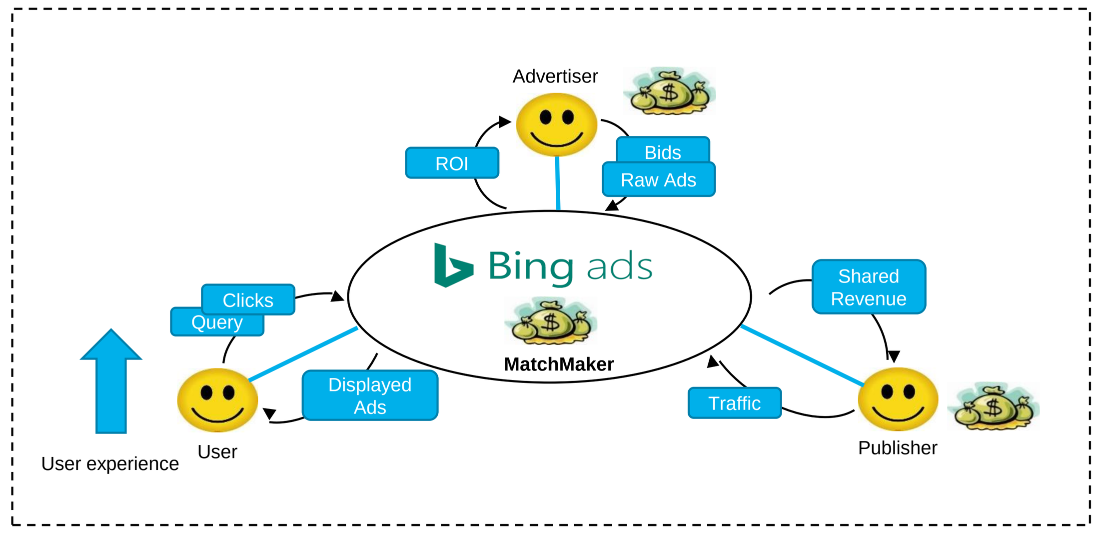
Genie: An Open Box Counterfactual Policy Estimator for Optimizing Sponsored Search Marketplace

Murat Ali Bayir, Mingsen Xu, Yaojia Zhu, Yifan Shi

{mbayir, mingx, yoazhu, yifanshi}@microsoft.com

Sponsored Search Optimization



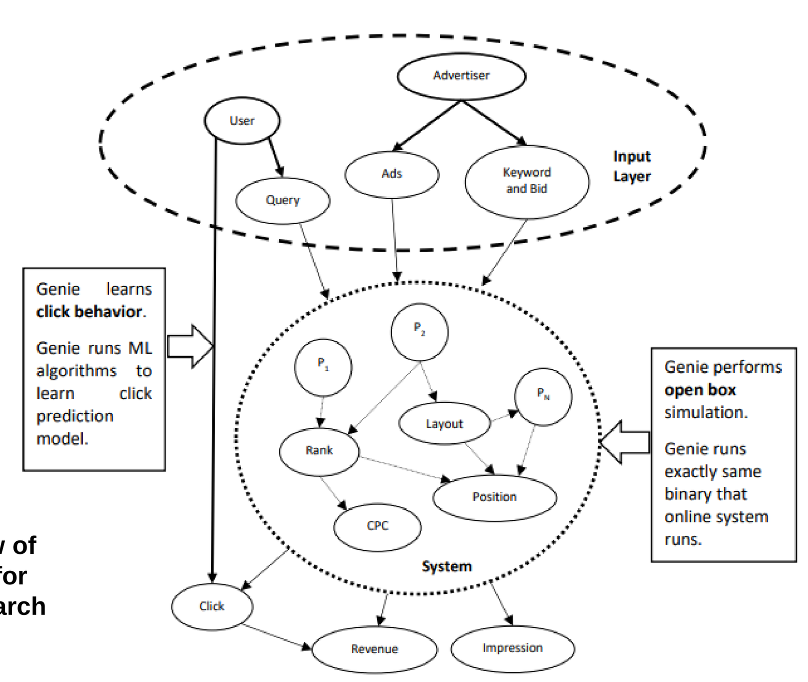
How do we optimize Marketplace?

- Requires **measuring KPI Impact of any modification to the System.**
- Need a counterfactual estimation system to answer "What If Questions?"

Existing Approaches

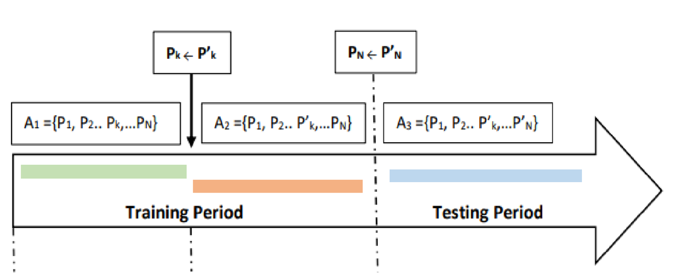
Method	Basic Idea	Pros	Cons
A/B Testing	- Run Two online experiment. - Deploy modification to real traffic run as treatment. - Compare with control traffic.	- Accurate - Can Measure many type of modification	- E2E Deployment - Risk in Real Traffic - Limited Parameter Space and policy Combinations.
Observational	- Run an online randomized experiment - Collect randomized data - Run Offline Training to generate new model or activate policy.	- Large Parameter Space. - Quick updates and efficiency.	- E2E Deployment. - Real Traffic for randomized experiment. - Cold Start problem.

Idea of Open Box Simulation



Simplified view of Causal Graph for Sponsored Search

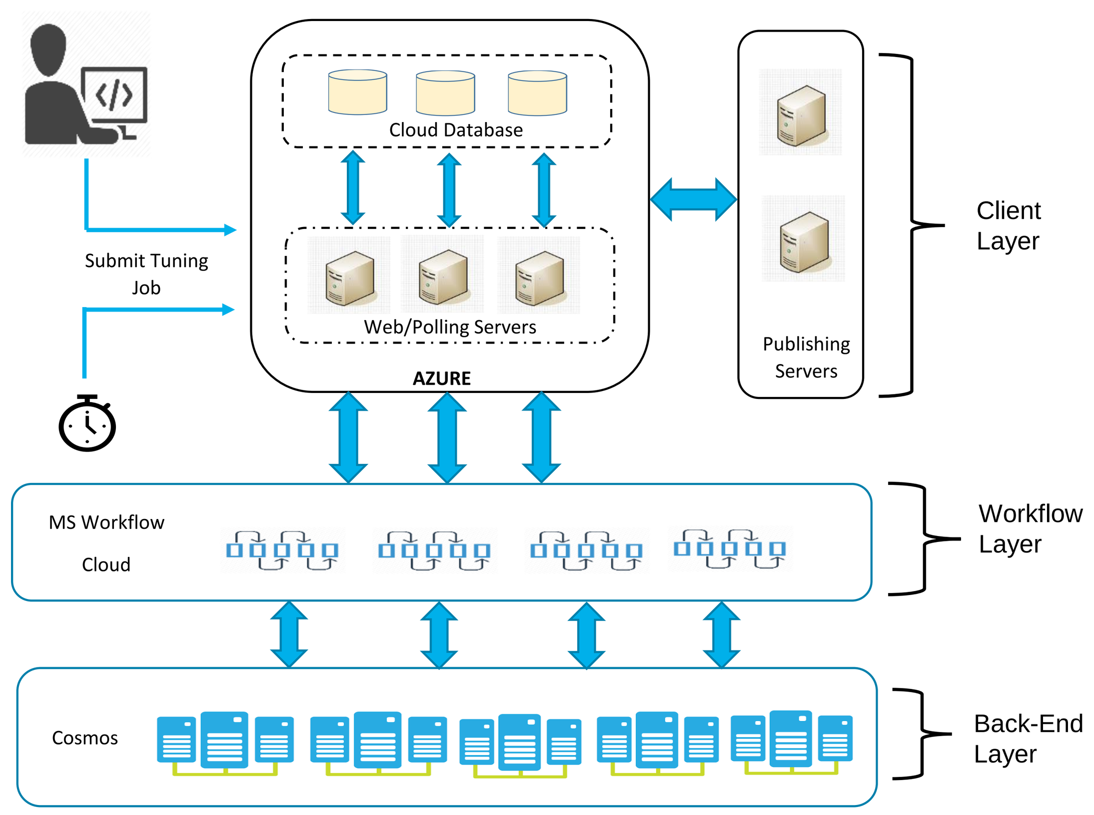
Why Open Box?



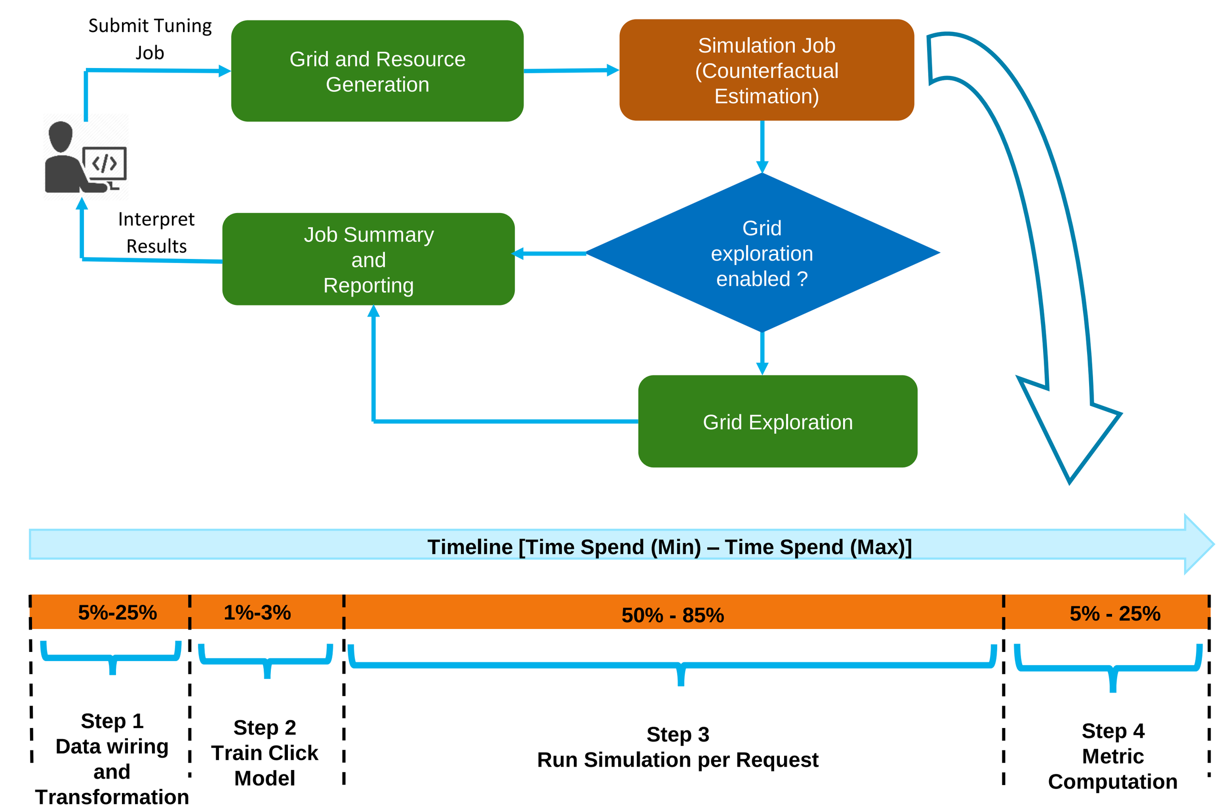
Actual Settings	[T1, T2) : $\{P_1, P_2, \dots P_k, \dots, P_N\}$ [T2, T3) : $\{P_1, P_2, \dots P_k', \dots, P_N\}$
Baseline For Open Box Estimator	[T1, T3) : $\{P_1, P_2, \dots P_k', \dots, P_N\}$
Counterfactual Setting	[T1, T3) : $\{P_1, P_2, \dots P_k', \dots, P_N'\}$

- Get rid of the negative impact of historical policy interactions. (Bing Ads PC is running more than 200 experiment simultaneously)
- Provides using higher volume of historical data. One sampling point could represent multiple settings/policies
- No Randomization cost and minimize the risk for Real Experiment.
- Good solution for cold start problem
- Can be leveraged when randomized experiment or A/B Testing is not appropriate.
  - Bid vs Traffic/Click Estimation Recommendation.

Genie Open Box Counterfactual Policy Estimator

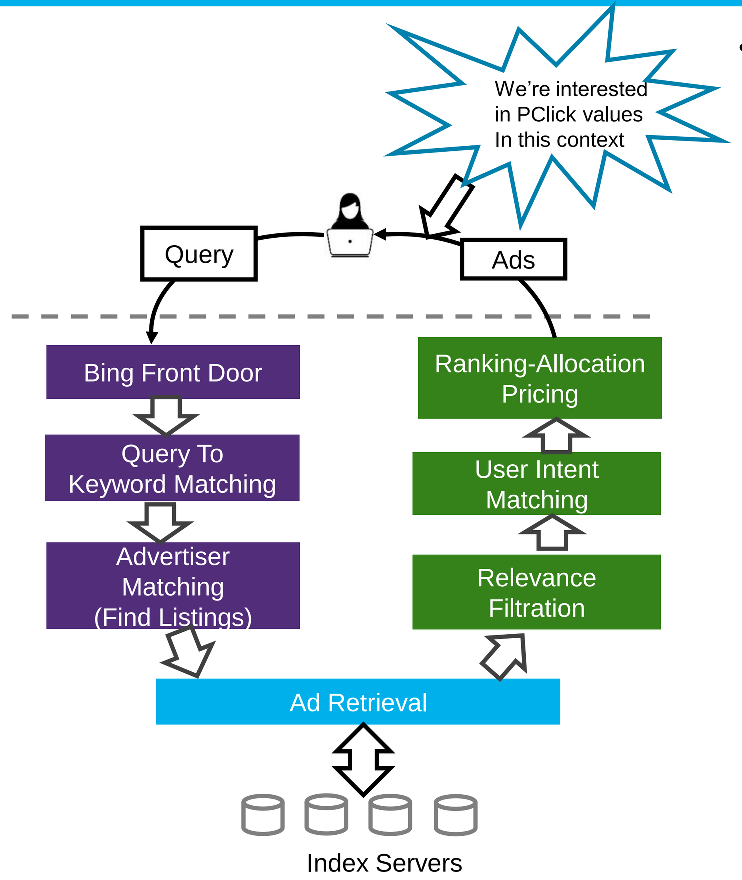


Genie Architecture



Genie Master Workflow

How Offline Click Prediction Training Works?

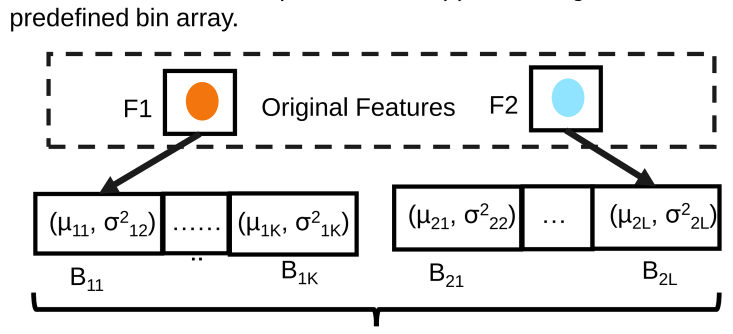


- **Goal:** Learn Click Model to *calibrate PClick scores* in replay/counterfactual page allocations.
- **Inputs:** (Impression Vector, click) pairs.
- **Outputs:** Trained model on Cosmos
- The PClick Score here is special one that is used for Offline metric Computation!

- **Genie currently supports two models:**
  - Bayesian Probit (Full Support). **Around %95 of scenarios.**
  - GTB: (Evaluation Only in Genie. Training is outside the Genie)

Bayesian Probit

Generalized linear model with standard CDF as a probit function [2]. Each feature from ad impression is mapped to weights in predefined bin array.



Gaussian Beliefs for Model Weights

Sampling Distribution $$p(y|\boldsymbol{x},\boldsymbol{w}) := \Phi\left(\frac{y\cdot \boldsymbol{w}^T\boldsymbol{x}}{\beta}\right)$$

Φ is standard cumulative distribution function and β is constant and controls steepness of the curve. Y can be either {-1 (no click), +1 (click)} in this context.

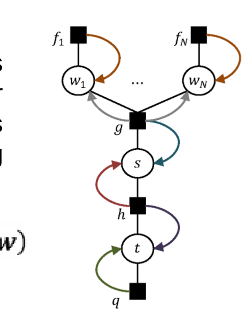
Prior for a given w: $$p(\boldsymbol{w}) = \prod_{i=1}^{N}\prod_{j=1}^{M_i}\mathcal{N}(w_{i,j}; \mu_{i,j}, \sigma^2_{i,j})$$

Bayesian Inference $$p(\boldsymbol{w}|\boldsymbol{x},y) \propto p(y|\boldsymbol{x},\boldsymbol{w})\cdot p(\boldsymbol{w})$$

Approximate P(w|x,y) with **factor graphs** as it does not have closed form solution [1].

Define a new latent variables s and t where s is a linear combination of weights and t is the sign of s after adding gaussian noise.

$$p(y \mid t)\cdot p(t \mid s)\cdot p(s \mid \boldsymbol{x},\boldsymbol{w})\cdot p(\boldsymbol{w})$$



- **Training Algorithm**
  - For each Impression data point.
    - Find matched bins for each feature
    - Compute the total variance and mean using gaussian of matched bins:

      (*) $\Sigma^2 := \beta^2 + \boldsymbol{x}^T\boldsymbol{\sigma}^2 \quad \mu = \boldsymbol{x}^T\boldsymbol{\mu}$
    - For each matched bin:
      - Update the mean and variance.
      - w and v are dynamic learning rate functions.

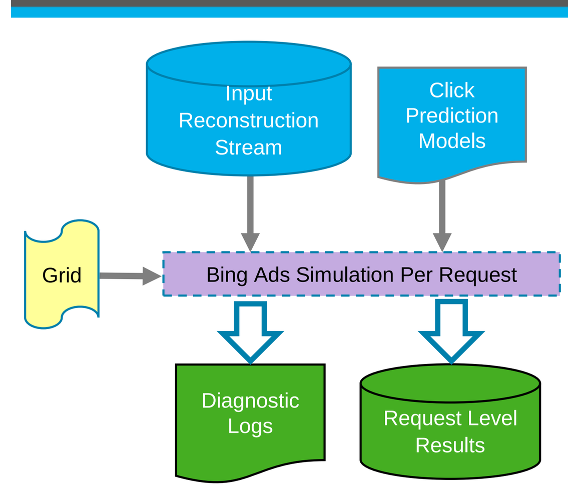
$$\tilde{\mu}_{i,j} = \mu_{i,j} + y x_{i,j}\cdot\frac{\sigma^2_{i,j}}{\Sigma}\cdot v\left(\frac{y\cdot \boldsymbol{x}^T\boldsymbol{\mu}}{\Sigma}\right)$$

$$\tilde{\sigma}^2_{i,j} \leftarrow \sigma^2_{i,j}\cdot\left[1 - x_{i,j}\cdot\frac{\sigma^2_{i,j}}{\Sigma^2}\cdot w\left(\frac{y\cdot \boldsymbol{x}^T\boldsymbol{\mu}}{\Sigma}\right)\right].$$

- **Evaluation:**
  - Total mean and variance are computed for new data x (*)
  - The cumulative distribution on total mean over square root of variance:

P(y|x) = $\Phi(\mu/\sqrt{\sigma^2+\beta^2})$

How Offline Simulation Works?



Algorithm 1: Simulation Algorithm

```
Input: Auction Data: A, Grid: G, Click Model: C
Output: List of (Setting, KPI) pairs as KPIs
M ← G.GenerateModifiers(A)
KPIs ← { } // Initialize the output.
foreach M_i ∈ M do
    // Modify the input and get (restorer, setting)
    (R_i, S_i) ← M_i.Modify(A)
    P_i ← OnlineLibrary(A) // Create i^th page allocation
    C.Predict(P_i) // Adjust click probabilities
    KPI_i ← GetKPI(P_i)
    KPIs ← KPIs ∪ (S_i, KPI_i)
    R_i.Restore(A) // Restore the input to original value
end
```

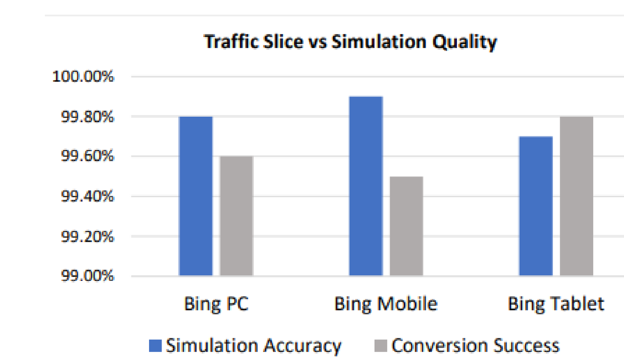
- Each counterfactual is converted into **Modifier and Restorers**.
- Modifiers modifies the simulation input in place and returns a restorer.
- Restorer restores the input to the original value.
- Each Inner loop do the following:
  - Modify Input
  - Call Online Library
  - Calibrate Page Assignment
  - Compute KPI
  - Add (request, setting id) level KPI to result pool.
  - Restore Inputs.

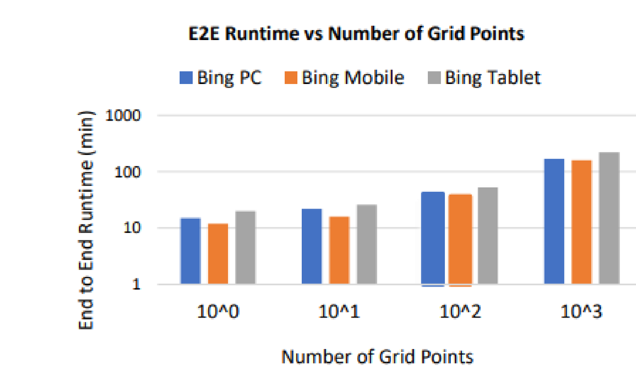
[1] Tom Minka. A family of algorithms for approximate Bayesian inference. PhD thesis, MIT. 2001

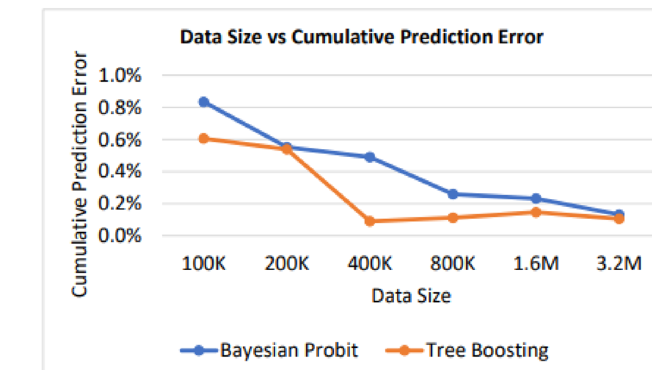
[2] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, Ralf Herbrich: Web-Scale Bayesian Click-Through rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine. ICML 2010: 13-20

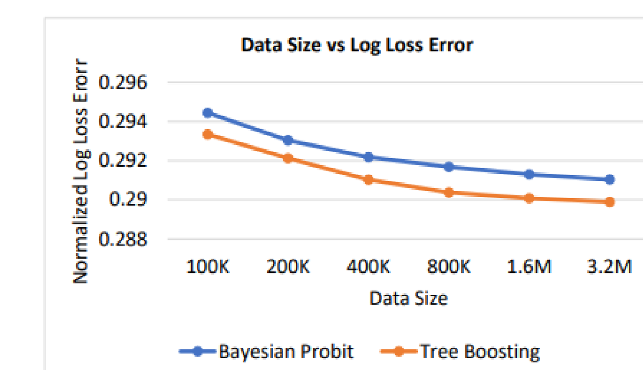
[3] Léon Bottou, Jonas Peters, Joaquin Quiñonero Candela, Denis Xavier Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, Ed Snelson: Counterfactual reasoning and learning systems: the example of computational advertising. Journal of Machine Learning Research 14(1): 3207-3260 (2013)

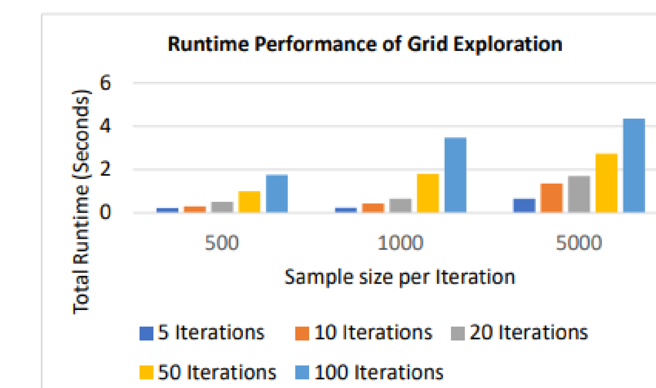
Experimental Results

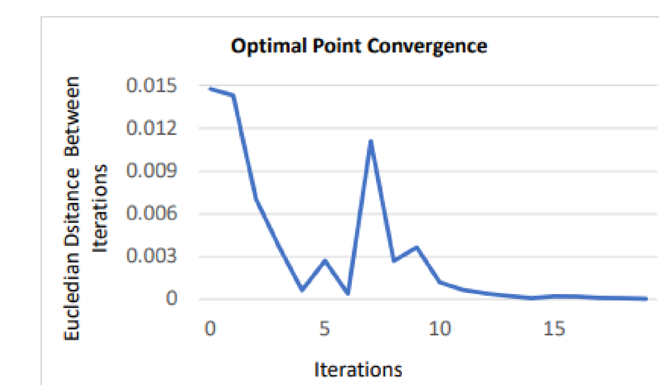










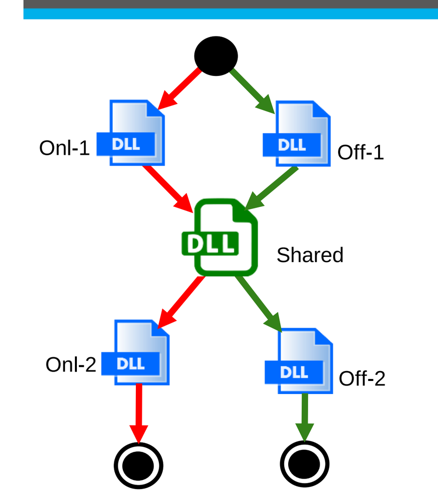


Method	RPM	MLIY	CY	CPC
IS (Historical)	1.27%	0.41%	0.39%	1.14%
Genie (Historical)	**1.16%**	**0.32%**	**0.37%**	**0.93%**
IS (Regression)	0.90%	0.36%	**0.24%**	0.98%
Genie (Regression)	**0.88%**	**0.25%**	0.27%	**0.66%**

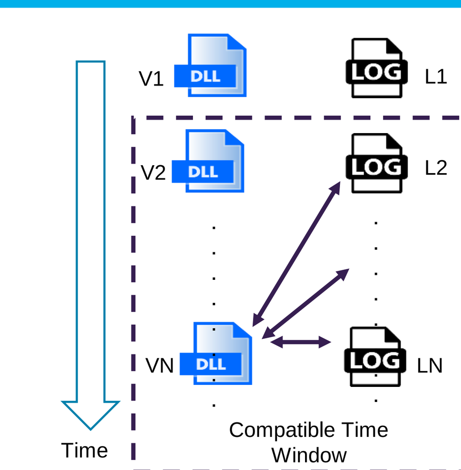
Comparison with Importance Sampling [3]

- Bing PC Experiment on 5 consecutive tuning time period during April to May 2018
- Each cell corresponds to KPI delta compared to A/B testing.
- Regression corresponds to metrics obtained from logs that has same date range with A/B testing.

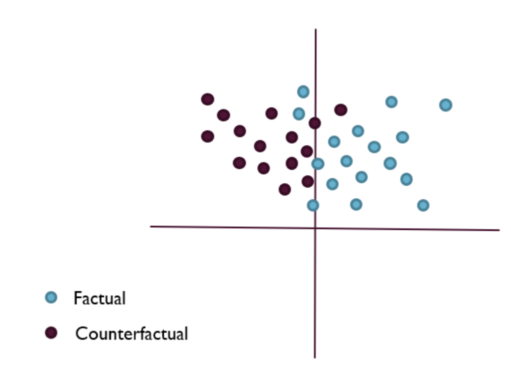
Challenges and Lessons Learned



Online vs Offline Parity



Backward Compatibility



Factual vs Counterfactual Feature Distribution

Future Work

- **Genie Explorer:** Running large number of candidate grid points in Genie is very costly. While completely data driven approach like Importance Sampling supports up to evaluation of 300K Grid points, Genie can only support up to 10K Grid points within 10-12 hours. Genie Explorer will focus on fixing this problem.
- Grid Exploration performance is poor for extrapolation, **Bayesian Optimization** could be used with single box simulator to explore points outside the bounding box of initial grid.

Acknowledgements

- The authors would like thank Emre Kiciman for feedbacks on improving the content of this paper, Tianji Wang and Qian Yao for their help on running Policy Estimators comparision experiments.
- Finally, authors would like to thank Rukmini Iyer, Eren Manavoglu, Patrick Jordan, Manish Agrawal, Patryk Zaryjewski, Anton Schwaighofer, Sendill Arunachalam, Pei Jiang, Suyesh Tiwari, Lohith Shesharam, Ranjita Naik, Marcin Machura, Urun Dogan, Elon Portugaly, Jitendra Ajmera, Aniruddha Gupta, Dilan Gorur, Debabrata Sengupta, Eugene Larchyk, Tommy Tan, Xiaoliang Ling, Thomas Borchert, many talented scientists and engineers in Microsoft Bing Ads Team for their help on Genie implementation and feedbacks for many features of Genie.