Enterprise Experimentation with Hierarchical Entities

Shan Ba, Shilpa Garg, Jitendra Agarwal, Hanyue Zhao

LinkedIn Corporation
700 E Middlefield Rd,
Mountain View, CA 94043 USA
{sba, shigarg, jagarwal, hazhao}@linkedin.com

Abstract

In this paper, we address the challenges of enterprise experimentation with hierarchical entities (e.g., for recruiter products) and present the methodologies behind the implementation of the Enterprise Experimentation Platform (EEP) at LinkedIn, enabling intelligent, scalable, and reliable experimentation to optimize performance across the company's enterprise offerings. We start with an introduction to the hierarchical entity relationships of the enterprise products and how such complex entity structure poses challenges to experimentation. We then delve into the details of our solutions for EEP including taxonomy based design setup with multiple entities, analysis methodologies in the presence of hierarchical entities, and advanced variance reduction techniques, etc. Recognizing the hierarchical ramping patterns inherent in enterprise experiments, we also propose a two-level Sample Size Ratio Mismatch (SSRM) detection methodology.

Introduction

The LinkedIn ecosystem propels member and customer value through a series of enterprise products, including talent solutions (for job seekers and recruiters), marketing solutions (for advertisers), sales solutions and learning solutions. The optimization of this value is achieved through the strategic utilization of data-informed decision-making and the integration of A/B testing (Kohavi, Tang, and Xu 2020) for more precise measurement of feature performance across LinkedIn's products. Enterprise products at LinkedIn used to suffer from inadequate experimentation capabilities due to several challenges associated with the intricate nature of its entity relationships.

(1) Different from individual consumers, the enterprise customers purchase LinkedIn's products (Recruiter, Sales, Learning) under contract or account entity, and under each contract or account, there are seats or profiles ranging from ten to ten thousands, therefore form the "Hierarchical entity relationships" (Figure 1). When we launch a new feature or deramp an existing feature with A/B testing to measure its impact, the enterprise customers often time are very sensitive to such change and require "same account same experience" to ensure all seats (i.e., recruiters, sales representatives, etc.) under the same contract or account get consistent

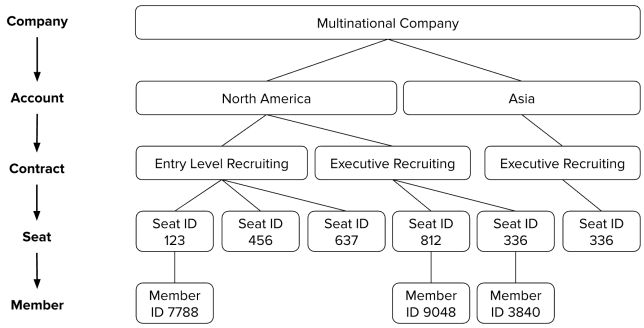

Figure 1: Hierarchical Entity Relationship in LinkedIn Talent Solutions (Recruiter Products).

experience during A/B testing. Therefore, in designing enterprise experiments, it is imperative to use "higher order entity" such as contract as the randomization entity, while the success metrics we are interested in are the "lower-order" seat entity metrics.

(2) Enterprise experimentation faces the small sample size and high variance problem. Because the experiment is randomized by contract / account entity and the total number of contracts / accounts is two or more order of magnitude smaller than the total number of seats / members, the enterprise experimentation can have two or more orders of magnitude larger variance than seat / member level experiments (i.e., a common consumer experimentation). Enterprise experimentation also tends to suffer from outliers due to high heterogeneity among accounts / contracts. For instance, a contract with a multinational company customer may encompass 10,000 or more seats, while a contract with a small business customer may include only 2 to 10 seats.

(3) The complex entity relationships in enterprise experiments also greatly complicates the Sample Size Ratio Mismatch (SSRM) issue. SSRM represents the situation where the observed sample size ratio (treatment sample size/control sample size) in the experiment is different from the expected ratio (Fabijan et al. 2019). A prior analysis revealed that approximately 10% of triggered analyses at LinkedIn exhibited SSRM (Chen, Liu, and Xu 2019). In order to ensure the internal validity and trustworthiness of the analy-

sis results, SSRM analysis should be included for every experiment (Kohavi, Tang, and Xu 2020). When SSRM is detected, it signals experiment is bias, rendering metric analyses invalid, and the experiment owner needs to diagnose/-fix the issue before interpreting the experiment readout. While SSRM is a well-explored topic in regular member-randomized experiments (Fabijan et al. 2019), SSRM under the hierarchical entity relationships have not been previously studied in the literature. The absence of a mechanism to detect SSRMs in EEP poses the risk that an ineffective treatment may erroneously seem beneficial in the enterprise experiments and be deployed to users.

Taxonomy based Design Setup

In traditional consumer experiments, the randomization, targeting, and success metric measurement are typically conducted on the same entity (e.g., member ID, guest cookie browser ID, etc.). However, in enterprise experimentation, EEP offers a high degree of flexibility. Users have the capability to utilize multiple entities for setting up randomization, targeting conditions, and success metric entities, as long as the entities and relationship are compliant with the taxonomy of the business line.

Greater flexibility in the setup comes with a greater complexity. Users at all levels have different knowledge in the complex domain and unrestricted configuration can be error-prone. Therefore we have introduced a formal model of the LinkedIn Enterprise domain, called "taxonomies", which define entities and types of relationships (Figure 2). Taxonomies are used to limit users' selections to a corresponding subset of entities and restrict their selections for entities used in A/B metric attribution and randomization. For example, users' test setup can only use the entities included in their corresponding taxonomy (i.e., Talent Solution cannot use advertiser which is a Marketing Solution entity in setup) and the hierarchical ramp must follow a strict 1:N relationship to yield valid A/B testing result (i.e., Sales Solution can randomize at the higher order "contract" entity and measure at the lower order "seat" entity, but cannot do vice versa).

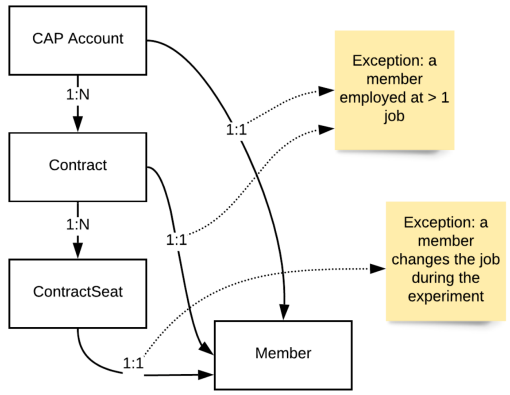

Figure 2: Taxonomy of LinkedIn Talent Solutions.

Variance Estimation with Hierarchical Entities

Enterprise experiments have *misaligned randomization unit and analysis unit*: the experiment needs to be randomized by accounts or contracts to ensure "same account same experience", but the success metric for analysis are the "lower-order" seat entity metrics. These metrics align with standard practices, facilitate comparisons across experiments, and avoid inconsistencies in the size of "higher-order" entities like accounts or contracts, which can vary significantly and lack stability over time.

Due to the misalignment of randomization unit and analysis unit, all metrics of interest in enterprise experiments need to be analyzed as ratio metrics. Mathematically, suppose that n contracts $i = 1, \ldots, n$ are randomly allocated to treated or control groups in an enterprise experiment. Let Y_i represent the revenue of contract i and N_i represent the number of seats in contract i, both of which are count metrics. Because contracts match the randomization units in the experiment, they can be assumed to be independent and we can directly calculate $Var(Y)$ and $Var(N)$ using the sample variance formula. When it comes to revenue per seat (our metric of interest in the enterprise experiment), however, the sample variance formula cannot be directly applied to calculate their variance because the seats under each contract are not independent. Instead, we need to view the seat-level metric (revenue per seat) as a "ratio" derived from two contract-level count metrics, which can be defined as Z below:

$$Z_{\text{treated}} = \frac{\sum_{i \text{ treated}} Y_i}{\sum_{i \text{ treated}} N_i} = \frac{\frac{1}{n_t}\sum_{i \text{ treated}} Y_i}{\frac{1}{n_t}\sum_{i \text{ treated}} N_i} \tag{1}$$

and

$$Z_{\text{control}} = \frac{\sum_{i \text{ control}} Y_i}{\sum_{i \text{ control}} N_i} = \frac{\frac{1}{n_c}\sum_{i \text{ control}} Y_i}{\frac{1}{n_c}\sum_{i \text{ control}} N_i}, \tag{2}$$

Note that Y_i and N_i are aggregated across all contracts (randomization units) in the treatment/control groups first before calculating the ratio. Because $\bar{Y}$ and $\bar{N}$ are jointly normal based on the central limit theorem, $Z = \bar{Y}/\bar{N}$ is also normally distributed, whose variance can be calculated by the delta method:

$$Var(Z) = \frac{1}{\bar{N}^2}Var(\bar{Y}) + \frac{\bar{Y}^2}{\bar{N}^4}Var(\bar{N}) - 2\frac{\bar{Y}}{\bar{N}^3}Cov(\bar{Y}, \bar{N}). \tag{3}$$

Before EEP has been made generally available at LinkedIn, a common mistake many analysts made in analyzing enterprise experiment was considering revenue per seat as a count metric $Z_i = Y_i/N_i$ and computing its variance by the sample variance formula $Var(Z) = \frac{1}{n-1}\sum_{i=1}^{n}(Z_i - \bar{Z})^2$. This would lead to incorrect variance estimate because Z_i $(i = 1, \ldots, n)$ are not independent in an enterprise experiment that was randomized by a higher-order entity such as the contract.

Variance Reduction

Because enterprise experiments have low sample size and highly heterogeneous experimental entities (i.e., account,

contract), it is important to apply variance reduction techniques to ensure that they could have enough statistical power in detecting treatment effects.

Basic Variance Reduction

In EEP, we reduce variance by leveraging covariates that are independent of the treatment but correlated with the experimental outcomes. The default analysis pipeline in EEP uses the CUPED methodology (Deng et al. 2013), which leverages pre-experiment metrics as the covariates. Suppose $T_i \in \{0, 1\}$ represents whether contract i has been randomly assigned into the control or treatment group, Y_i is a count metric at the contract level (e.g., revenue of contract i) during the experiment period, N_i is the number of seats triggered in contract i during the experiment period, and X_i and M_i are the corresponding measurements of Y_i and N_i during the pre-experiment period. Compared to the regular difference-in-mean estimator (without variance reduction):

$$\frac{\sum_{T_i=1} Y_i}{\sum_{T_i=1} N_i} - \frac{\sum_{T_i=0} Y_i}{\sum_{T_i=0} N_i}, \tag{4}$$

CUPED estimates the treatment effect by:

$$\frac{\sum_{T_i=1} Y_i}{\sum_{T_i=1} N_i} - \frac{\sum_{T_i=0} Y_i}{\sum_{T_i=0} N_i} - \theta \cdot \left(\frac{\sum_{T_i=1} X_i}{\sum_{T_i=1} M_i} - \frac{\sum_{T_i=0} X_i}{\sum_{T_i=0} M_i} \right), \tag{5}$$

where

$$\theta = cov(\frac{\bar{Y}}{\mu_N} - \frac{\mu_Y \bar{N}}{\mu_N^2}, \frac{\bar{X}}{\mu_M} - \frac{\mu_X \bar{M}}{\mu_M^2}) / var(\frac{\bar{X}}{\mu_M} - \frac{\mu_X \bar{M}}{\mu_M^2}). \tag{6}$$

Because the pre-experiment metrics X_i and M_i typically have a high correlation with the experimental outcomes Y_i and N_i, they can be used to largely remove the pre-existing difference among the experimental entities. The EEP variance reduction pipeline has implemented both the regression adjustment method and the stratification method including outlier capping capabilities.

Advanced Variance Reduction

In addition to the default pipeline, we have also developed an advanced nonlinear variance reduction solution for EEP which can leverage a large number of covariates (i.e., not just the pre-experiment metrics (Deng et al. 2013) and utilize nonlinear adjustment models (i.e., extending the linear adjustment method from CUPED to flexible machine learning methods (Guo et al. 2024; Jin and Ba 2023). Let X denote a rich class of covariates where the pre-experiment metric is included as a special case, and $\hat{\mu}^Y(.)$ and $\hat{\mu}^N(.)$ represent some machine learning predictors for Y and N based on X. In order to achieve unbiased variance reduction, it is important to eliminate two types of biases: (1) "*regressor bias*" from $\hat{\mu}(.)$ whose convergence rate could be slower than $n^{-1/2}$ without a well-posed parametric model; (2) "*double-dipping bias*" which occurs if the same dataset is used both for model-fitting and for prediction. Algorithm 1 describes our proposed variance reduction procedure which introduces de-biasing terms to correct the regressor bias and also employs the cross-fitting technique to remove the "double-dipping bias". The first step is the K-fold sample splitting for $\mathcal{D} = (Y_i, N_i, T_i, X_i)_{i=1}^n$ and then the second step is cross-fitting: for each $k \in [K]$, we use the data $\{(X_i, N_i, Y_i) \colon T_i = 1, i \in \mathcal{D}^{(-k)}\}$ to obtain estimators $\hat{\mu}_1^{Y,(k)}(x)$ for $\mathbb{E}[Y(1) \mid X = x]$ and $\hat{\mu}_1^{N,(k)}(x)$ for $\mathbb{E}[N(1) \mid X = x]$. Likewise, we use $\{(X_i, N_i, Y_i) \colon T_i = 0, i \in \mathcal{D}^{(-k)}\}$ to obtain $\hat{\mu}_0^{Y,(k)}(x)$ and $\hat{\mu}_0^{N,(k)}(x)$. Then, we calculate predictions $\hat{\mu}_w^Y(X_i) = \hat{\mu}_w^{Y,(k)}(X_i)$ and $\hat{\mu}_w^N(X_i) = \hat{\mu}_w^{N,(k)}(X_i)$ for all $i \in \mathcal{D}^{(k)}$, $w \in \{0, 1\}$. Finally, we estimate the treatment effect by

$$\frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n B_i} - \frac{\sum_{i=1}^n C_i}{\sum_{i=1}^n D_i}, \tag{7}$$

where $A_i = \hat{\mu}_1^Y(X_i) + \frac{T_i}{\hat{p}}\big(Y_i - \hat{\mu}_1^Y(X_i)\big)$, $B_i = \hat{\mu}_1^N(X_i) + \frac{T_i}{\hat{p}}\big(N_i - \hat{\mu}_1^N(X_i)\big)$, $C_i = \hat{\mu}_0^Y(X_i) + \frac{1-T_i}{1-\hat{p}}\big(Y_i - \hat{\mu}_0^Y(X_i)\big)$, and $D_i = \hat{\mu}_0^N(X_i) + \frac{1-T_i}{1-\hat{p}}\big(N_i - \hat{\mu}_0^N(X_i)\big)$ for $\hat{p} = n_t/n$. Compared to the difference-in-mean estimator in (4), the estimator in (7) can be viewed as substituting the sample means of the treated and control groups with averages of the fit-and-debias predictions for the potential outcomes of all n units (e.g., contracts). It can be proved that the variance reduction procedure in Algorithm 1 is finite sample unbiased and asymptotically optimal (in the sense of semi-parametric efficiency) among all regular estimators as long as the machine learning estimators are consistent, without any requirement for their convergence rates (Jin and Ba 2023). In practice, the proposed advanced nonlinear variance reduction methodology can further reduce up to 30% of variance compared to CUPED by going beyond linearity and incorporating a large number of extra covariates.

Algorithm 1: Advanced variance reduction solution by leveraging flexible nonlinear models with a large number of covariates

1: Input: Dataset $\mathcal{D} = \{(Y_i, X_i, N_i, T_i\}_{i=1}^n$, number of folds K.
2: Randomly split $\mathcal{D}$ into K folds $\mathcal{D}^{(k)}$, $k = 1, \ldots, K$.
3: **for** $k = 1, \ldots, K$ **do**
4: Use all (X_i, N_i, Y_i) with $T_i = 1$ and $i \notin \mathcal{D}^{(k)}$ to obtain $\hat{\mu}_1^{Y,(k)}(x)$ and $\hat{\mu}_1^{N,(k)}(x)$;
5: Use all (X_i, N_i, Y_i) with $T_i = 0$ and $i \notin \mathcal{D}^{(k)}$ to obtain $\hat{\mu}_0^{Y,(k)}(x)$ and $\hat{\mu}_0^{N,(k)}(x)$;
6: Compute $\hat{\mu}_w^Y(X_i) = \hat{\mu}_w^{Y,(k)}(X_i)$ and $\hat{\mu}_w^N(X_i) = \hat{\mu}_w^{N,(k)}(X_i)$ for all $i \in \mathcal{D}^{(k)}$ and $w \in \{0, 1\}$.
7: **end for**
8: Compute the estimator according to (7).

Two-Level Sample Size Ratio Mismatch

By default, EEP is triggered by its analysis unit (individual seats), which is more granular than its randomization unit (the whole contracts). By only triggering a subset of active seats from a contract in each enterprise experiment, EEP effectively filters out noise generated by dormant seats

unaffected by the experiment treatment, enhancing sensitivity and experiment power.

Sample Size Ratio Mismatch (SSRM), a.k.a. Sample Ratio Mismatch (SRM), indicates a significant discrepancy between the observed ratio of triggered units across different experiment variants and the expected ratio as per the experiment's design. Previous studies in the literature have primarily addressed SSRM at the randomization unit level. Recognizing the hierarchical structure described above, we propose to examine two distinct types of SSRMs within EEP (or more generally, in a cluster-randomized experiment): one at the randomization unit level (contract-level SSRM) and the other at the analysis unit level (seat-level SSRM). Both of them are essential for safeguarding the trustworthiness of the experiment results (Figure 3).

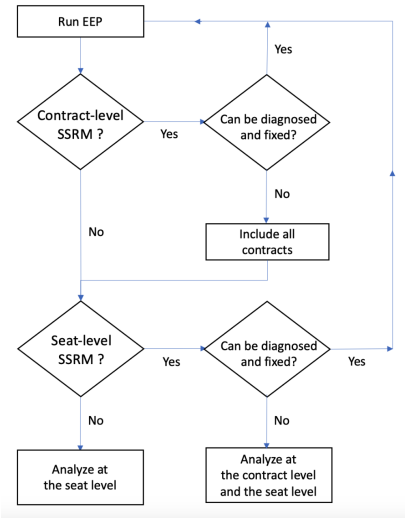

Figure 3: Flowchart for Sample Size Ratio Mismatch (SSRM) detection in EEP.

Because a contract would get triggered as long as at least one of its seats is triggered into the experiment, the seat-level SSRM is more likely to occur than the contract-level SSRM. Consider a seat-level SSRM scenario where a bad treatment feature substantially reduces the number of active users (seats) within each contract. Failure to detect the seat-level SSRM could erroneously favor the ineffective treatment variant. This is because the treatment variant may exhibit a more favorable revenue-per-user ratio than the control variant, given that the remaining users within the contracts are likely the most active ones. Mathematically, the ratio metric Y/N becomes unsuitable for decision-making when the treatment alters N (seat-level SSRM), as it becomes challenging to interpret whether a change in the ratio value is beneficial or detrimental. Unfortunately, in this scenario, the traditional SSRM detection solution at the randomization unit level (contract-level SSRM) would not be effective. This is because a contract would be triggered into the experiment as long as at least one of its seats is active/triggered. Despite the considerable decline in active seats within the contracts, the overall number of contracts triggered in the experiment may not significantly change.

To the best of our knowledge, the detection of SSRM at the analysis unit level for a cluster-randomized experiment has not been presented in the literature. Given that the "expected sample size ratio" is unknown at the analysis unit level (seat level) and the analysis units (seats) within the hierarchical ramping pattern are not independent, its detection method must diverge from the existing approaches at the randomization unit level. Our proposed solution to detect analysis-unit-level (seat-level) SSRM is summarized in Algorithm 2, which ensures the stability of the denominator metric N, enabling meaningful conclusions based on the ratio metric Y/N. It involves using N_i^{pre} as a baseline to adjust for pre-existing size differences among contracts, which is crucial due to the high heterogeneity in contract size in EEP. The coefficient θ in step 3 minimizes $var(D)$ which is similar to the CUPED estimator. Under the traditional SSRM detection framework, $N^{triggered}$ can be viewed as the "observed sample size" in the experiment and N^{pre} acts as a surrogate for the unknown "expected sample size" at the seat level. Because the test for seat-level SSRM in step 4 is conducted at the randomization unit (contract) level, it no longer violates the independence assumption of the t test.

Algorithm 2: Detection of analysis-unit-level (seat-level) SSRM

1: Obtain the list of contracts that were triggered in the experiment (assuming no contract-level SSRM).
2: Compute the contract-level metrics $N^{triggered}$ and N^{pre} for each triggered contract. ($N^{triggered}$ represents the number of triggered seats per contract in the experiment period, and N^{pre} represents the number of active seats per contract in the pre-experiment period.)
3: For each triggered contract i, compute

$$D_i = N_i^{triggered} - \theta(N_i^{pre} - E(N^{pre})),$$

where $\theta = cov(N^{triggered}, N^{pre})/var(N^{pre})$.
4: Run a two-sample t test to compare whether the mean of D_i is significantly different between the treatment group and the control group. If the difference is significant, fire a seat-level SSRM alert.

Results

Enable and Scale Up Measurements

Before the implementation of EEP, enterprise-facing feature or infrastructure changes were rolled out to our customers in quarterly batches (known as Quarterly Product Releases) without A/B testing. While we diligently monitored metric changes before and after each release and collected qualitative feedback from our customers, our enterprise business line lacked robust measurement and data-driven insights necessary for optimal decision-making. With the introduction of EEP, we have transformed the concept of "A/B testing being impossible" into a feasible reality for enterprise products.

Since its launch, the number of enterprise experiments running on the EEP platform has rapidly scaled: starting with approximately 10 experiments during the pilot phase, the number escalated to over 100 per quarter during the beta phase, surpassed 500 per quarter post the General Availability of EEP, and currently maintains a pace of over 1000 experiments (testing around 300 unique new product features) per quarter. The typical run time of each experiment ranges from 2 to 4 weeks.

Bolster Trustworthiness through SSRM Guardrails

EEP has improved the quality of readouts by implementing SSRM guardrail monitoring and reducing manual analysis errors. At an aggregated level, we have found approximately 7% of tests suffer from SSRMs. Our observations include:

(1) The existing single-level SSRM detection method (at the contract level) failed to identify any issues within EEP. This is because a contract is triggered into the experiment if at least one of its seats is triggered. In most cases, contracts containing zero triggered seats were rare, and nearly all contracts were triggered regardless.

(2) The seat-level SSRM (under the proposed two-level SSRM solution) should be the focus for SSRM detection within EEP. Space limitations prevent us from discussing its root cause diagnosis (e.g., due to residual effects, etc.).

Expedite Experimentation Velocity and Productivity

With EEP, the end-to-end clock time required to leverage online testing for evaluating new enterprise features has significantly improved. Prior to EEP, the engineering team had to manually generate test / control contracts for each segment, implement workarounds to target contracts and seats correctly, manually check quality guardrails such as SSRM, and execute error-prone manual scripts for compute readouts with variance reduction procedures. With EEP, all these tasks are end-to-end automated by the platform including the advanced variance reduction solution, resulting in a 50% reduction in efforts and a 2-week reduction in clock time per experiment (Figure 4). Additionally, EEP has streamlined the post-review and quality check process, and introduced a user-friendly readout report UI that allows Product Managers to self-serve, thereby expediting the business decision-making process.

Conclusions

In conclusion, our work on the Enterprise Experimentation Platform (EEP) at LinkedIn has addressed critical challenges posed by the hierarchical entity relationships within enterprise products. The proposed two-level Sample Size Ratio Mismatch (SSRM) detection methodology, operating at both randomization unit and analysis unit levels, further enhances the platform's capability to ensure internal validity and the reliability of analysis results.

Acknowledgments

We would like to thank our current and former colleagues who have contributed to the realization EEP. Thanks to Alexander Ivaniuk, Weitao Duan, Min Liu, Justin Marsh, Juanyan Li, Ying Jin, Chunzhe Zhang, Wentao Su and special thanks to our leaders for their supports: Ya Xu, Kapil Surlaker, Vish Balasubramanian, Kuo-Ning Huang, Sofus Macskassy, Parvez Ahammad and Robert Kyle.

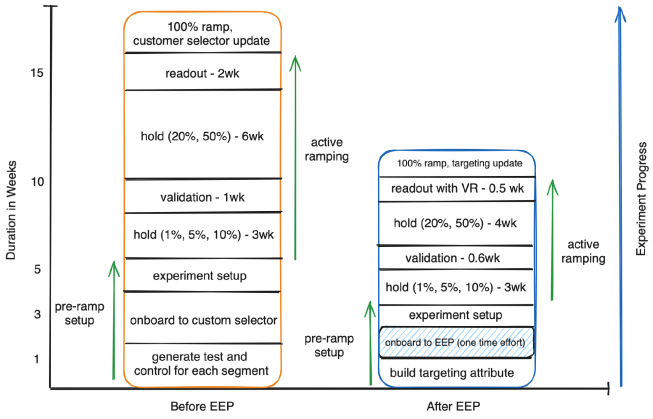

Figure 4: Experimentation speed and effort comparisons with/without EEP.

References

Chen, N.; Liu, M.; and Xu, Y. 2019. How A/B Tests Could Go Wrong: Automatic Diagnosis of Invalid Online Experiments. WSDM '19, 501–509. New York, NY, USA: Association for Computing Machinery. ISBN 9781450359405.

Deng, A.; Xu, Y.; Kohavi, R.; and Walker, T. 2013. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *Proceedings of the sixth ACM international conference on Web search and data mining*, 123–132.

Fabijan, A.; Gupchup, J.; Gupta, S.; Omhover, J.; Qin, W.; Vermeer, L.; and Dmitriev, P. 2019. Diagnosing Sample Ratio Mismatch in Online Controlled Experiments: A Taxonomy and Rules of Thumb for Practitioners. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 2156–2164. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016.

Guo, Y.; Coey, D.; Konutgan, M.; Li, W.; Schoener, C.; and Goldman, M. 2024. Machine learning for variance reduction in online experiments. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713845393.

Jin, Y.; and Ba, S. 2023. Toward Optimal Variance Reduction in Online Controlled Experiments. *Technometrics*, 65(2): 231–242.

Kohavi, R.; Tang, D.; and Xu, Y. 2020. *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*. Cambridge University Press.