VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

DATA SCIENCE STUDY PROGRAMME

Master's thesis

Measuring Interference Effects in Digital Experiments

Sąveikos efektų matavimas skaitmeniniuose eksperimentuose

Aina Kugelevičiūtė

Supervisor : Dr. Rasa Giniūnaitė
Scientific advisor : Dr. Agnė Reklaitė

Vilnius
2026

Measuring interference effects in digital experiments

Summary

A comprehensive simulation-based analysis of causal inference under interference is conducted for shared-inventory marketplace experiments, focusing on the estimation of average treatment effects (ATE) under varying interaction dynamics and randomization schemes. Using the FINN.no slate dataset, three outcome-generating models are examined: a baseline independent click model restricted to first clicks, an independent click model allowing repeated interactions, and a repeated clicks model. Together, these models capture increasing levels of user interaction intensity, competition, and spillover effects. Bernoulli and cluster randomization schemes are evaluated across multiple treatment probabilities, and exposure mapping is employed to separate treatment effects into direct and interference components at the user level. The results show that more aggressive interaction dynamics reduce causal efficiency, as intensified competition diminishes realized treatment effects. Bernoulli randomization systematically overestimates treatment effects at low treatment probabilities, while cluster randomization substantially reduces this inflation but results in downward bias at higher treatment probabilities. Exposure mapping further reveals strong negative competitive spillovers and demonstrates that total treatment effects are substantially smaller than direct effects.

This study contributes to the field by showing that ignoring interference in shared-inventory marketplaces can lead to treatment effect estimates that are highly precise yet systematically biased. Appropriate randomization strategies and explicit accounting for exposure are therefore essential for obtaining credible and actionable causal insights to support product development and user experience improvements in modern digital experimentation.

Keywords: causal inference, A/B testing, online marketplaces, average treatment effect, Bernoulli randomization, cluster randomization, exposure mapping.

Sąveikos efektų matavimas skaitmeniniuose eksperimentuose

Santrauka

Darbe atliekama išsami simuliacijomis pagrįsta priežastinio ryšio (angl. *causal inference*) analizė bendro inventoriaus turinčiose internetinėse prekyvietėse, nagrinėjant sąveikos (angl. *interference*) poveikį ir vertinant vidutinio poveikio (angl. *ATE*) įverčius pagal skirtingus sąveikos modelius bei randomizacijos schemas. Naudojant FINN.no slate duomenų rinkinį, nagrinėjami trys rezultatus generuojantys modeliai: bazinis nepriklausomų paspaudimų modelis, apribotas tik pirmaisiais paspaudimais, nepriklausomų paspaudimų modelis, leidžiantis pasikartojančias sąveikas, ir pasikartojančių paspaudimų modelis. Kartu šie modeliai atspindi didėjantį vartotojų sąveikos intensyvumą, konkurenciją ir poveikio išsiliejimo (angl. *spillover*) efektus. Bernulio ir klasterinė randomizacijos schemos vertinamos skirtingomis variacijos (angl. *treatment*) tikimybėmis, o poveikio žemėlapiai (angl. *exposure mapping*) naudojami efektų skaidymui į tiesioginius ir tarpusavio sąveikos komponentus vartotojo lygiu. Rezultatai rodo, kad agresyvesnė sąveikų dinamika mažina priežastinį efektyvumą, nes intensyvi konkurencija mažina realizuotus variacijos efektus. Bernulio randomizacija sistemingai pervertina efektus esant žemoms variacijos tikimybėms, o klasterinė randomizacija ženkliai sumažina šį pervertinimą, tačiau didesnėse tikimybėse lemia neigiamą nuokrypį (angl. *downward bias*). Poveikio žemėlapiai taip pat atskleidžia stiprų neigiamą konkurencinį išsiliejimą ir parodo, kad bendri efektai yra gerokai mažesni už tiesioginius efektus.

Šis tyrimas prisideda prie nagrinėjamos srities, parodydamas, kad ignoruojant tarpusavio sąveikas bendros inventoriaus turinčiose rinkose efektų įverčiai gali būti labai tikslūs, tačiau sistemingai šališki. Todėl tinkamos randomizacijos parinkimas ir aiškus poveikio įvertinimas yra būtini norint gauti patikimas ir praktiškai pritaikomas priežastinio ryšio įžvalgas, kurios padėtų paremti produktų kūrimą ir vartotojų patirties gerinimą šiuolaikiniuose skaitmeniniuose eksperimentuose.

Raktiniai žodžiai: priežastinė analizė, A/B testavimas, internetinės prekyvietės, vidutinis poveikis, Bernulio randomizacija, klasterinė randomizacija, ekspozicijos žemėlapis.

List of figures

List of tables

Contents

List of abbreviations

ATE	Average treatment effect
CI	Confidence interval
CTR	Click-through rate
EDA	Exploratory data analysis
ICM	Independent click model
RCM	Repeated clicks model
SUTVA	Stable unit treatment value assumption
TATE	Total average treatment effect

Introduction

Digital experiments, also known as A/B testing, have become an essential tool in the technology industry for guiding product development and supporting business decisions. These experiments enable organizations to evaluate the effects of product changes by randomly assigning users to treatment and control groups and measuring differences in outcomes. The validity of such experiments typically relies on the stable unit treatment value assumption (SUTVA), which states that the potential outcomes of a given unit of analysis are independent of the treatment assignments other units receive [13]. However, in many real-world digital platforms, this assumption is frequently violated due to interference effects: situations where the treatment allocated to one user influences the behavior or outcomes of others. Such interference is especially common in interconnected environments like social networks or two-sided marketplaces, where user interactions produce spillover effects. Violation of SUTVA leads to biased estimate of a new feature's effectiveness, namely treatment effect, due to interference between units. As a result, engineering teams depending on experiments to validate features may draw inaccurate conclusions, resulting in a less effective product development process [23].

This thesis analyses interference in online marketplaces using the FINN.no slate dataset, collected from Norway's leading online marketplace. The dataset reflects realistic recommendation scenarios where users are presented with multiple items and may click on one or none. It enables the study of sequential user behavior, competition over shared items, and multi-item decision processes under experimental interventions.

The purpose of this thesis is to investigate how interference shapes causal inference in marketplace experiments and to assess the performance of commonly used randomization schemes under controlled violations of SUTVA. Three outcome-generating models are considered: a baseline independent click model (ICM) restricted to first clicks only, an independent click model (ICM) retaining all interactions and a repeated clicks model (RCM) with diminishing returns. Additionally, exposure mapping is applied to decompose treatment effects into direct and interference components, providing insight into how competition and repeated interactions influence observed outcomes. The contributions are threefold: comparing interaction models to assess interference effects, evaluating randomization bias across treatment probabilities, and demonstrating exposure mapping as a diagnostic tool for separating direct and spillover effects.

The thesis is structured as follows: Section 1 reviews the related literature on causal inference under interference and experimental design in digital platforms. Section 2 presents the methodological framework, describing the outcome-generating models, simulation setup, randomization schemes, and exposure definitions. Section 3 reports the empirical results, including model comparisons, bias analyses across treatment probabilities, and exposure-based decompositions. Finally, the concluding section summarizes the main findings, discusses their implications for experimental practice in marketplaces, and outlines directions for future research.

1 Literature review

This section presents an overview of interference in digital experiments, with a particular focus on online platforms and outline the reasoning behind the methodological approaches chosen in this thesis.

SUTVA and foundational perspectives. The validity of a randomized experiment in estimating the total average treatment effect (TATE) relies on the stable unit treatment value assumption (SUTVA) [13]. This assumption requires that the potential outcomes for any unit do not vary with the treatments assigned to other units, and, for each unit, no multiple versions of a treatment level exist, that lead to different potential outcomes. These two components of the stability assumption allow researchers to use the presence of multiple units to estimate causal effects [15]. Early work by Cox [7] and Rubin [24] already recognized that violations of SUTVA undermine the interpretation of randomized experiments. Manski [21] formalized the identification challenges arising from social interactions, showing that causal effects are generally not point-identified without additional structure. Hudgens and Halloran [14] later introduced a potential-outcomes framework for causal inference under interference, defining direct, indirect, total, and overall effects. This framework has since become a foundational approach for analyzing spillover effects in networked and grouped experiments.

The problem of interference. Interference arises when the treatment assigned to one unit affects the outcomes of other units, a condition that naturally emerges in online marketplaces. These platforms are highly interconnected environments: multiple sellers may offer products that either complement or substitute for one another, and sellers often make strategic choices influenced by their competitors. As a result, SUTVA is unlikely to hold in these settings. Blake and Coey [4] illustrated how such interference can occur in real-world marketplace experiments. Drawing on examples from online platforms, they showed that interactions between buyers and sellers can spill across experimental groups, contaminating control conditions and leading to biased estimates of treatment effects. Beyond marketplaces, interference is also prevalent in content and recommendation platforms. A case study at Douyin by Ma et al. [20] demonstrated that algorithmic recommendation systems induce strong exposure spillovers, as treated users affect the content consumption and engagement of untreated users through shared recommendation pools and social feedback mechanisms. Their findings highlight that interference can arise even in the absence of explicit social networks, making it a fundamental challenge for experimentation in large-scale recommender systems.

Network and partial interference models. To address interference structurally, Eckles et al. [8] developed a formal framework for analyzing experiments conducted within networked environments. They proposed cluster-based randomization, where units are grouped according to their network connections and randomized at the cluster rather than the individual level. This method helps localize interference within clusters, thereby reducing its impact on estimated treatment effects. Closely related work formalizes this idea through the assumption of partial interference, under which units can be partitioned into groups that experience interference internally but not across groups [14, 27]. Hudgens and Halloran [14] introduced a potential-outcomes framework that distinguishes between direct, indirect, total, and overall effects, providing a principled way to interpret causal estimands in the presence of spillovers. Aronow and Samii [1] further generalized these ideas by introducing exposure mappings, allowing researchers to define treatment conditions as functions of neighbors' assignments. This approach enables flexible modeling of interference that depends on treatment intensity rather than binary exposure alone.

Interference in marketplaces and strategic settings. Interference is especially consequential in two-sided and competitive platforms, where experimental interventions can disrupt market equilibrium. Johari et al. [16] show that naïve randomization in two-sided platforms can induce bias due to endogenous responses on both sides of the market. Li et al. [19] quantify the resulting bias–variance trade-offs and provide guidance for experimental design under interference. From a broader economic perspective, Athey and Imbens [2] emphasize that experiments affecting prices, congestion, or strategic behavior may fail to recover policy-relevant effects if equilibrium adjustments are ignored. Related work on competition in networked and algorithmic markets [3, 6] further illustrates how strategic interactions can amplify or attenuate experimental treatment effects, reinforcing the need for interference-aware designs.

Detecting and modeling interference in practice. Several studies develop practical approaches to identify and limit interference in large-scale experimentation environments. Han et al. [12] propose increasing-allocation designs that detect interference by exploiting nonlinear responses as treatment saturation rises, without requiring explicit network information. Reklaitė and Gamper [23] introduce an offline diagnostic approach that analyzes historical A/B tests to uncover recurring interference patterns. Saveski et al. [25] develop a meta-experimental framework that randomizes over randomized experiments to identify network effects through outcome heterogeneity. Interference can also emerge across concurrent experiments: Fradkin et al. [11] show that parallel experiments on advertising platforms interact through shared auctions and budget constraints, biasing individual experimental estimates.

Dynamic and decision-theoretic perspectives. Recent work emphasizes that interference often evolves over time as treatments are rolled out. Boyarsky et al. [5] model interference induced by experiment roll-out dynamics, capturing temporal spillovers between early- and late-treated units. From a decision-theoretic standpoint, Johari et al. [17] study when interference is large enough to alter experimental conclusions, providing criteria for trading off statistical rigor against operational feasibility. Related contributions clarify how treatment effects should be interpreted and estimated when interference is unknown or only partially observed [18, 26, 28]. Together, these studies demonstrate that interference is more than a statistical challenge but a central feature of experimentation in modern digital platforms.

2 Methodology

The following subsections describe methodologies used in this work to model, identify and measure interference effects in digital A/B experiments.

2.1 Notation and causal setup

Consider a population of N experimental units (e.g., users), indexed by $i = 1, \ldots, N$. Each unit is assigned a treatment indicator $Z_i \in \{0,1\}$, where $Z_i = 1$ denotes treatment and $Z_i = 0$ denotes control. The classical stable unit treatment value assumption (SUTVA) posits that the potential outcome of unit i depends only on its own treatment assignment:

$$Y_i = Y_i(Z_i).$$

However, in the presence of interference, the outcome may depend on the entire treatment vector $\mathbf{Z} = (Z_1, \ldots, Z_N)$, i.e.,

$$Y_i = Y_i(\mathbf{Z}),$$

which violates SUTVA and complicates causal inference. This formulation follows the standard potential outcomes framework for interference [8, 15] and provides the foundation for the methods considered in this section.

2.2 Interference modeling

This chapter presents a simulation based framework for analyzing interference in a realistic marketplace environment. The framework explicitly models competition for purchases, repeated exposure to items, and spillover effects across users. By fully controlling the data generating process, the framework enables a systematic comparison of alternative experimental designs and allows for quantification of the bias arising from interference. The primary objective is to understand how standard randomized experiments behave when the SUTVA is violated, and to evaluate whether alternative randomization and analysis strategies can mitigate resulting biases.

2.2.1 Experimental setup

The experimental setting mimics a large scale online marketplace where users are exposed to multiple items over time and where items may receive multiple clicks from different users. Users can be in either a control or treatment condition, with treatment intended to represent an intervention such as improved ranking, enhanced presentation, or promotional highlighting of items.

Let $i = 1, \ldots, N$ index users, $t = 1, \ldots, T_i$ index interactions (slates) for user i, and $k = 1, \ldots, K_{it}$ index positions within slate t of user i. Then each row corresponds to (i,t,k) and is associated with:

- a displayed item j_{itk} (corresponding to *clicked_item_id*),
- a position-level contextual feature (category) x_{itk},
- and interaction-level features (slate length, interaction type) x_{it}.

To introduce stochasticity in purchase decisions while maintaining full experimental control, a latent random variable

$$X_{itk} \sim \text{Uniform}(0{,}1)$$

is generated independently for each row (i,t,k). This variable represents unobserved momentary purchase propensity and serves as the basis for determining potential and realized purchases.

Throughout the simulation, purchase probabilities are:

$$p_{\mathrm{C}} = 0.2 \quad \text{for control users, and} \quad p_{\mathrm{T}} = 0.3 \quad \text{for treated users.}$$

These probabilities will be mapped to each row (i,t,k) according to the user's treatment assignment.

2.2.2 Bernoulli randomization

Treatment is assigned at the user level using independent Bernoulli randomization. Let $Z_i \in \{0{,}1\}$ denote the treatment indicator for user i, where $Z_i = 1$ corresponds to treatment and $Z_i = 0$ to control. Each user is then independently assigned to the treatment group with a probability of 0.5:

$$Z_i \sim \text{Bernoulli}(0.5), \quad i = 1, \ldots, N.$$

All interactions associated with a given user inherit that user's treatment assignment. That is, for each row (i,t,k), the row-level treatment indicator is defined as

$$\text{is\_treatment}_{itk} := Z_i.$$

This design reflects common practice in online platforms, where each user is consistently exposed to the same experimental condition across all interactions.

2.2.3 Cluster-based randomization

An alternative approach for mitigating interference is cluster-based randomization at the interaction (slate) level. Each cluster corresponds to a user–interaction pair, defined by a specific slate shown to a given user. Formally, let clusters be indexed by $c = (i,t)$, where i denotes a user and t an interaction.

Let $Z_c \in \{0{,}1\}$ denote the treatment indicator for cluster $c = (i,t)$. All rows (i,t,k) belonging to the same interaction inherit this assignment:

$$\text{is\_treatment}_{itk} := Z_{it}.$$

Clusters are randomly assigned to treatment or control such that approximately half of the interactions are treated. This design ensures that all items within a given slate are consistently exposed to the same experimental condition, while allowing treatment status to vary across interactions for the same user.

Let $\mathcal{R}_T$ and $\mathcal{R}_C$ denote the sets of treated and control rows, respectively. A natural estimator of the average treatment effect at the row level is

$$\hat{\tau} = \frac{1}{|\mathcal{R}_T|} \sum_{(i,t,k)\in\mathcal{R}_T} Y_{itk} - \frac{1}{|\mathcal{R}_C|} \sum_{(i,t,k)\in\mathcal{R}_C} Y_{itk}.$$

Under standard assumptions, this estimator is unbiased when interference is contained within interactions.

2.2.4 True average treatment effect

To establish a ground-truth benchmark, a no-interference outcome model is defined. Under this model, each row-level outcome depends only on its own treatment status and an independent latent variable, and is unaffected by the treatment assignments or actions of other users or interactions. Because users may click on the same item multiple times, the dataset used for this benchmark is restricted to the first observed click for each user–item pair. Formally, for each user i and item j, only the earliest row (i,t,k) such that $j_{itk} = j$ is retained. This ensures that each observation corresponds to a unique potential outcome and avoids overweighting repeated interactions, thereby aligning the data with the assumptions of the no-interference model.

For each retained row (i,t,k), the realized purchase outcome is generated as

$$Y_{itk} = \mathbf{1}\{X_{itk} < p_{itk}\},$$

where

$$p_{itk} = \begin{cases} p_{\mathrm{T}}, & \text{if } Z_i = 1, \\ p_{\mathrm{C}}, & \text{if } Z_i = 0. \end{cases}$$

This specification satisfies SUTVA, as outcomes are unaffected by the treatment assignments or actions of other users or interactions. The true average treatment effect (ATE) is then defined as

$$\mathrm{ATE} = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)]\,.$$

Since outcomes are generated by a Bernoulli process with fixed purchase probabilities, the true ATE admits the closed-form expression

$$\mathrm{ATE} = p_{\mathrm{T}} - p_{\mathrm{C}}.$$

To ensure consistency with the simulation framework and to verify this analytical result, the true ATE is also computed numerically by constructing two counterfactual datasets:

1. a treated world, in which all users are assigned $Z_i = 1$;
2. a control world, in which all users are assigned $Z_i = 0$.

For each world, outcomes are generated using the described no-interference model, yielding vectors of potential outcomes $Y(1)$ and $Y(0)$. The true ATE is recovered as the difference in their sample means, which coincides with $p_{\mathrm{T}} - p_{\mathrm{C}}$.

The resulting true ATE serves as a benchmark against which estimators obtained under realistic interference mechanisms are evaluated. Because the data-generating process is fully controlled, any discrepancy between estimated treatment effects and this benchmark can be attributed exclusively to interference and biases induced by the experimental design.

2.2.5 Modeling interference through competition for purchases

A key feature of real-world marketplaces is that multiple users may compete for the same item, while each item can only be purchased once. This creates interference across users, as one user's purchase prevents others from purchasing the same item. In the simulation framework, this is modeled by allowing multiple rows to generate potential purchases for the same item, but restricting the final outcome such that at most one purchase occurs per item. Among all rows associated with a given item that generate a potential purchase, the row with the highest latent affinity X is selected as the realized purchase.

Let D_{itk} denote a binary indicator for a potential purchase at row (i,t,k):

$$D_{itk} = \mathbf{1}\{X_{itk} < p_{itk}\},$$

where p_{itk} is the purchase probability determined by treatment assignment. For a given item j, let

$$\mathcal{S}(j) = \{(i,t,k) : j_{itk} = j\}$$

denote the set of rows corresponding to that item.
The realized purchase outcome is then

$$Y_{itk} = \begin{cases} 1, & \text{if } D_{itk} = 1 \text{ and } X_{itk} = \max\{X_{i't'k'} : (i',t',k') \in \mathcal{S}(j_{itk}),\, D_{i't'k'} = 1\}, \\ 0, & \text{otherwise.} \end{cases}$$

This construction ensures that treatment effects propagate across users: increasing the purchase probability for one group raises the likelihood that treated rows "win" the competition, thereby reducing observed purchases among control rows for the same item. Such negative spillovers violate SUTVA and induce bias in naïve estimators.

Independent click model (ICM). The first interference model assumes that each click represents an independent purchase opportunity. The probability of a potential purchase depends only on treatment status and not on previous interactions with the same item. Under this model,

$$p_{itk} = \begin{cases} p_{\mathrm{T}}, & \text{if is\_treatment}_{itk} = 1, \\ p_{\mathrm{C}}, & \text{if is\_treatment}_{itk} = 0. \end{cases}$$

While simplistic, this model isolates interference that arises purely from cross-user competition, without confounding effects from repeated exposure, and thus serves as a useful baseline for understanding how minimal forms of interference can bias experimental estimates.

Repeated clicks model (RCM) with diminishing returns. In practice, users may click on the same item multiple times before making a purchase decision. To capture this behavior, the repeated clicks model introduces within-user interference over time. For each row (i,t,k), let r_{itk} denote the number of prior clicks by user i on item j_{itk} before interaction (t,k). Then the purchase probability is modeled as a cumulative process with diminishing returns:

$$p_{itk} = 1 - (1-p)^{r_{itk}+1},$$

where p equals to p_{C} or p_{T} depending on the treatment assignment $\text{is\_treatment}_{itk}$.

This formulation implies that repeated exposure increases purchase likelihood, but with decreasing marginal impact. Repeated clicks from a single user therefore increase that user's probability of capturing the item, intensifying competition with other users. As before, realized purchases are restricted to at most one per item using the competition rule, generating interference both across users and across time.

2.3 Average treatment effect estimation

This subsection describes how treatment effects are estimated under different interference regimes and experimental designs, and how resulting estimates are compared against the true benchmark ATE defined in the 2.2.4 subsection.

2.3.1 User-level aggregation

While interference is modeled at the click and item level, the estimand of interest is the effect of treatment on the probability that a user makes at least one purchase. Consequently, outcomes are aggregated to the user level prior to estimation.

For each user i the binary outcome is defined:

$$\tilde{Y}_i = \mathbf{1}\{\exists\,(t,k) \text{ such that } Y_{itk} = 1\}\,.$$

which equals one if user i makes at least one realized purchase during the experiment. This aggregation ensures that each user contributes exactly one observation, preventing users with many clicks from dominating the estimates.

2.3.2 ATE estimator

For a given experimental design and interference model, the average treatment effect is estimated using a simple difference in means:

$$\hat{\tau} = \frac{1}{N_1} \sum_{i:Z_i=1} \tilde{Y}_i - \frac{1}{N_0} \sum_{i:Z_i=0} \tilde{Y}_i.$$

where N_1 and N_0 denote the number of treated and control users, respectively.

This estimator is unbiased for the true ATE under SUTVA and independent randomization. However, when interference is present, the estimator may capture not only the direct effect of treatment but also indirect spillover effects arising from competition for items. Comparing $\hat{\tau}$ to the true benchmark ATE therefore allows direct quantification of interference-induced bias.

2.3.3 Uncertainty quantification

To assess statistical uncertainty, standard errors are computed using the classical variance estimator for the difference in means:

$$\widehat{\mathrm{Var}}(\hat{\tau}) = \frac{\widehat{\mathrm{Var}}(\tilde{Y}_i \mid Z_i = 1)}{N_1} + \frac{\widehat{\mathrm{Var}}(\tilde{Y}_i \mid Z_i = 0)}{N_0}.$$

Approximate 95% confidence intervals are then constructed as

$$\hat{\tau} \pm 1.96\sqrt{\widehat{\mathrm{Var}}(\hat{\tau})}.$$

Although this variance estimator does not explicitly account for interference, it reflects standard practice in large-scale online experimentation. Deviations of the resulting confidence intervals from the true ATE therefore illustrate how interference can invalidate conventional inference procedures.

2.3.4 Bias decomposition

For each experimental configuration, the bias of the estimator is computed as

$$\mathrm{Bias} = \hat{\tau} - \mathrm{ATE},$$

where the true ATE is obtained from the no-interference benchmark. Since the data-generating process is fully controlled, any observed bias can be attributed solely to interference effects and the experimental design.

2.4 Exposure mapping

Cluster-based randomization can mitigate interference, though it may frequently be impractical in realistic marketplace settings. Exposure mapping provides a way to account for interference by explicitly modeling it rather than attempting to eliminate it. In the simulated marketplace, interference arises because multiple users compete for the same items, so a user's outcome depends not only on their own treatment status but also on the treatment assignments of other users targeting the same items. A binary treatment indicator alone cannot capture this structure. Exposure mapping augments treatment with a continuous measure of competitive exposure, enabling a decomposition of treatment effects into direct effects and spillovers.

2.2.4. Definition. For each item j, define the *item-level treatment exposure* as the fraction of clicks on that item coming from treated users:

$$E_j = \frac{1}{|\mathcal{S}(j)|} \sum_{(i,t,k)\in\mathcal{S}(j)} \text{is\_treatment}_{itk},$$

where $\mathcal{S}(j)$ denotes the set of all clicks on item j.
To avoid mechanical correlation between a user's own treatment and exposure, a leave-one-out version is constructed. For each click (i,t,k) on item j,

$$E_{j,-i} = \begin{cases} \dfrac{\sum_{(i',t',k')\in\mathcal{S}(j)} \text{is\_treatment}_{i't'k'} - \text{is\_treatment}_{itk}}{|\mathcal{S}(j)| - 1}, & |\mathcal{S}(j)| > 1, \\ 0, & \text{otherwise.} \end{cases}$$

This quantity represents the fraction of treated competitors faced by user i when attempting to purchase item j. The *user-level exposure* is then defined as the average leave-one-out exposure across all items clicked by user i:

$$\bar{E}_i = \frac{1}{|\mathcal{I}_i|} \sum_{j\in\mathcal{I}_i} E_{j,-i},$$

where $\mathcal{I}_i$ denotes the set of items clicked by user i.

User-level outcomes are aggregated into a binary indicator $\tilde{Y}_i$ equal to one if user i makes at least one purchase. Let Z_i denote treatment assignment and $\bar{E}_i$ the exposure measure. Then for the independent click model, the following linear probability model is estimated:

$$\tilde{Y}_i = \beta_0 + \beta_1 Z_i + \beta_2 \bar{E}_i + \beta_3 (Z_i \times \bar{E}_i) + \varepsilon_i.$$

For the repeated clicks model, user click intensity k_i is additionally controlled for:

$$\tilde{Y}_i = \beta_0 + \beta_1 Z_i + \beta_2 \bar{E}_i + \beta_3 (Z_i \times \bar{E}_i) + \beta_4 k_i + \varepsilon_i.$$

Here, β_1 captures the direct treatment effect at zero exposure, β_2 measures spillovers from treated competitors among controls, and β_3 describes how exposure moderates the treatment effect. In the repeated clicks model specification, β_4 captures the role of repeated engagement intensity.

The total average treatment effect evaluated at the mean exposure level is

$$\text{ATE}_{\text{total}} = \beta_1 + \beta_3 \cdot \mathbb{E}[\bar{E}_i].$$

Since exposure is defined through shared items, outcomes are correlated among users competing for the same items. Inference therefore relies on cluster-robust standard errors, clustering at the item level.

3 Analytical part

3.1 Data description

The analyzed FINN.no slate dataset was developed to support research on dynamic slate recommendation systems. The data was collected from FINN.no, which is the leading online marketplace in Norway, providing users with a platform to buy and sell general merchandise, vehicles, real estate, house rentals, and job listings. The dataset reflects realistic recommendation scenarios, where users are presented with multiple items on a slate and may click on one or none of them. This makes it suitable for evaluating models that account for both sequential user behavior and multi-item decision processes.

The dataset captures 30 days of user interactions, logging every interaction between users and displayed slates. For each user u and interaction step t, items in the visible slate $a_t^u(s_t^u)$ were recorded, along with the user's click response c_t^u. Each slate is labeled according to its origin: search query (approximately 80% of slates) or recommendation algorithm (approximately 20% of slates). This sequential structure enables the evaluation of models such as Gated Recurrent Units (GRUs) for capturing evolving user preferences and Thompson Sampling for adaptive slate recommendations. In total, the dataset contains approximately 37.4 million interactions, with $|\mathcal{U}| \approx 2.3$ million users and $|\mathcal{I}| \approx 1.3$ million items, that belong to one of $G = 290$ item groups, defined based on category and geographical location [9].

The FINN.no slate dataset, as described in [10], includes three key files:

- interaction data (`data.npz`), which stores all slate and click data. The dataset includes approximately 2.3 million unique users, each with up to 20 recorded interactions on the platform. At each interaction, users are exposed to slates containing up to 25 items. The two primary arrays in this file are *slate* and *click*. The array dimensions follow the convention: first dimension is per user, second is time and third dimension is the presented slate. The arrays include:
  - *slate* [userId, interaction num, slate pos]: items presented to users in each slate;
  - *click* [userId, interaction num]: items clicked by the users in each slate;
  - *interaction type* [userId, interaction num]: indicates type of interaction the user had with the platform (search or recommendation);
  - *click_idx* [userId, interaction num]: auxiliary data representing position of the clicked item within the slate (0–24);
  - *slate_lengths* [userId, interaction num]: auxiliary data of actual length of each slate, calculated as 25 minus the number of padding indices.
- index to item (`ind2val.json`), which provides mappings from indices to textual values for key attributes. The arrays include:

  - *category* [290]: maps each category index to a descriptive string that identifies the category and location of the group. The first word denotes the main category (e.g., “MOTOR”, “REAL_ESTATE”, or “BAP” for general merchandise). Subcategories are appended with commas when sufficient items exist. Two consecutive commas (e.g., “MOTOR, , ”) indicate missing or inapplicable subcategories. The third word specifies the geographical location;
  - *interaction_type* [3]: indicates whether the presented slate originated from a search query, a recommendation algorithm, or is undefined.

- item attributes (`itemattr.npz`), that contains the array *category* [itemId], encoding the group to which each item belongs to.

The dataset, along with its documentation, is publicly accessible through the FINN.no GitHub repository.

3.2 Data preprocessing

Due to its large size, the dataset was distributed using Git Large File Storage (Git LFS), which enables efficient handling of large binary files while ensuring full reproducibility of the data acquisition process. All data processing steps, including data download and transformation, were implemented in Python and can be accessed in a Jupyter notebook running on Google Colab (FINN.no Slate Dataset preprocessing.ipynb).

Given the scale of the dataset, all NumPy arrays were loaded using memory mapping, allowing the data to be accessed directly from disk without loading the full dataset into memory. This approach is essential for scalability and enables efficient processing of large interaction tensors. Metadata files were loaded once and reused across processing steps, while item attributes were stored using reduced-precision floating-point formats where possible to further minimize memory usage.

To facilitate analysis, the original tensor-based data representation was transformed into a flat, interaction-level dataset, where each row corresponds to a single user–interaction instance with a valid click. During this transformation, interactions with zero slate length were discarded, and invalid clicks, such as clicks referring to positions outside the displayed slate, were excluded. For valid interactions, the clicked item was identified based on the recorded click position, and item categories were mapped from integer indices too descriptive labels using the provided metadata. These transformations converted the high-dimensional raw data into a format suitable for exploratory analysis, causal modeling, and simulation.

To manage computational and memory constraints, preprocessing was performed in chunks of users. The dataset was divided into batches of fixed size, each processed independently. For each chunk, interaction-level rows were generated and written to disk as Apache Parquet files. Parquet was chosen for its efficient columnar storage, built-in compression, and compatibility with both Python and R analytics pipelines. This chunked approach ensures that the preprocessing pipeline scales to millions of interactions without exceeding memory limits.

3.3 Exploratory data analysis

The exploratory data analysis (EDA) of the FINN.no slate dataset, as well as all subsequent modeling, was implemented in R using RStudio. The purpose of the analysis is to understand the structure of the data, user behavior patterns, interaction characteristics, and potential biases that may influence downstream modeling and evaluation.

Each observation in analyzed dataset corresponds to a single interaction and contains information about the user, clicked item, interaction type, and the position of the item within a slate. To focus on meaningful interactions, rows corresponding to "no-click" was removed. The resulting dataset contains only interactions where a user actively clicked on an item, which are interpreted as expressions of purchase intent. Moreover, to optimize computational performance by focusing on relevant features, the *slate_items* column, which contains the full list of items displayed in each slate, was entirely excluded from the analysis and further use. As the modeling approach relies on interaction level features, such as clicked item identifiers, this column was unnecessary. Its removal simplifies data processing and improves computational efficiency during aggregation and visualization, without loss of relevant information for the modeling tasks.

An initial examination of the dataset confirmed that there are no missing entries in the core variables used for analysis, including user identifiers, item identifiers, interaction types, and categories. This aligns with expectations given the preprocessing steps applied. The dataset contains a large number of interaction events across a substantial user base and a diverse set of items and categories. Table 1 provides an overview of the key dimensions of the dataset.

Rows	Users	Items	Categories	Interaction types
37,426,847	2,277,645	1,172,429	288	2

Table 1. Total number of main variables in the FINN.no slate dataset.

To understand user engagement, the distribution of interactions and clicks across the platform was analyzed. Each user is recorded with up to 20 interactions on the platform. This upper bound is reflected in the interaction distribution (Figure 10), where a sharp spike occurs at 20 interactions. Rather than indicating unusually high engagement, this spike arises because interaction counts do not increase beyond this fixed limit. Consequently, the distribution is shaped by the data collection process, and engagement levels beyond 20 interactions are not observable. Users with fewer interactions are progressively less frequent, with only a small fraction exhibiting very low activity (e.g., 1–3 interactions). However, the dominant mass at the maximum interaction count highlights that many users reached the recorded limit, making the distribution highly concentrated rather than smoothly decaying. A similar, though weaker, pattern is observed for click activity (Figure 11). Click counts are more smoothly distributed across users, with most users generating a moderate number of clicks and relatively few users exhibiting very low click activity. When aggregating interactions by type, clear differences in click-through rate (CTR) emerge between search based and recommendation based interactions

(Figure 12). Recommendation driven slates exhibit a CTR of approximately 63%, while search driven slates show a higher CTR of around 80%. This indicates that search slates result in user engagement more frequently than recommendation based slates, which may reflect the higher intent associated with user initiated search queries compared to automated recommendations. At the user level, CTR values vary substantially. The distribution of user CTRs (Figure 13) shows a small group of users with very low or near-zero CTRs, alongside a much larger concentration of users who click on a high fraction of presented items. For example, approximately 11,600 users have a CTR between 0.0 and 0.1, while about 1.8 million users fall within the 0.6–1.0 range. The mean user CTR, indicated by the dashed vertical line, illustrates how a minority of low CTR users pulls the average downward despite the majority of users exhibiting high engagement. An aggregate comparison between click and no-click events highlights a clear class imbalance (Figure 14). Approximately 75% of slates result in a click, while only about 25% do not.

The dataset was further analyzed by category, revealing an uneven distribution of interactions across them. A small number of categories account for a large fraction of total activity, with *REAL_ESTATE*, *MOTOR*, and *BAP*[1] together dominating the dataset (Figure 15). Due to its high volume of interactions, the *REAL_ESTATE* category was selected for a more detailed analysis. For these interactions, the category hierarchy was decomposed to extract location information. Then the ten most frequent locations were identified and geocoded to obtain latitude and longitude coordinates. Figure 16 illustrates these top locations, showing a strong concentration of interactions around major Norwegian cities, particularly Oslo, Akershus, and Hordaland. This spatial clustering reflects population density and housing market activity.

To examine user engagement with slates, click-through rate was analyzed as a function of slate length and item position (see Figures 17 and 18). CTR generally increases with slate length, reaching values above 0.9 for slates containing 12–25 items, though very long slates (e.g., 26 items) show a slight decrease, suggesting that excessively long lists may reduce user attention. At the same time, a clear positional bias is observed: items at the top of a slate are more likely to be clicked, with CTR decreasing monotonically for lower positions, from approximately 0.17 at the first position to around 0.005 at the 24th. These results highlight the trade-off between exposing more items and maintaining user engagement, as well as the impact of presentation order on click behavior.

Overall, the exploratory analysis reveals substantial user heterogeneity, uneven click activity, concentration of interactions across categories and geographic locations, and clear effects of slate length and click position. These findings provide important guidance for the modeling in the following sections.

---

[1] The general merchandise category (FINN.no/torget) is referred to as “bits and pieces”, a name originally given by a developer and retained ever since.

3.4 Model comparison under interference

This section compares the performance of three outcome-generating models under Bernoulli randomization. The models differ in how they translate user interactions into purchase opportunities and how strongly they induce interference through competition and repeated exposure. The goal of the comparison is twofold: first, to understand how different modeling assumptions affect observed treatment effects, and second, to assess the efficiency–aggressiveness trade-off induced by each model.

All models are applied to the same interaction-level dataset after removing "no-click" interactions. Potential purchases are generated according to either the independent click model (ICM) or the repeated clicks model (RCM) described in subsection 2.2.5. In all cases, realized purchases are restricted to at most one per item using a common competition rule, so that multiple potential buyers for the same item induce cross-user interference. The models differ only in which interactions are retained and in whether repeated clicks affect purchase probabilities.

The three models considered are:

- Model 0 (ICM with first-click restriction). This model applies the independent click model but retains only the first click for each user–item pair. Each retained interaction therefore represents a single, independent purchase opportunity whose probability depends only on treatment assignment.
- Model 1 (ICM with full interaction history). Like Model 0, this model follows the independent click model using the same treatment-dependent purchase probabilities, but it retains all click-level interactions. Each click constitutes an independent purchase opportunity, increasing competition for items when users click multiple times.
- Model 2 (RCM with diminishing returns). This model also operates on the full interaction history but replaces the ICM with the repeated clicks model. Repeated clicks by the same user on the same item increase that user's purchase probability over time, but with decreasing marginal impact.

All comparisons were conducted at the user level, aggregating outcomes across interactions to reflect standard experimental reporting metrics.

User-level aggregation and distributional properties. To ensure that results are not driven by a small number of highly active users, outcomes are aggregated at the user level. For each user, two quantities are computed:

1. the total number of potential purchase opportunities generated (*potential_buy*),
2. an indicator for whether the user makes at least one purchase (*final_buy*).

The distribution of potential purchase intensity at the user level is right-skewed but exhibits a limited upper tail (Figure 1). In the underlying data, the median user generates three potential purchase opportunities, and 90% of users generate no more than six. Only 1% of users exceed nine potential purchases, with an observed maximum of nineteen. This limited upper tail suggests that the empirical results are not driven by a small number of extreme users but instead reflect systematic patterns across the broader user population. Importantly, the overall shape of the distribution is highly similar across models and treatment conditions. This similarity indicates that differences in estimated treatment effects are unlikely to be driven by changes in user-level heterogeneity and instead stem from the underlying structural assumptions of the models. For visualization purposes, the distribution is plotted on a logarithmic scale, which preserves the relative differences across users while accounting for strong right skew.

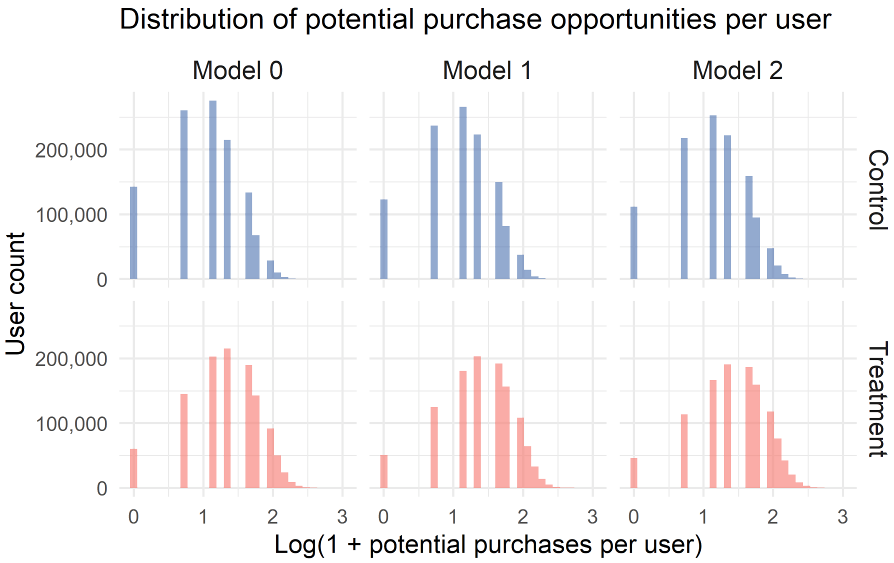


Figure 1. Distribution of the number of potential purchase opportunities per user (log scale). The distribution is right-skewed with a limited upper tail, indicating that results are not driven by a small number of extreme users.

Average treatment effects across models. Figure 2 demonstrates the estimated average treatment effect on the probability that a user makes at least one purchase. Model 0 and Model 1 yield very similar treatment effect estimates, with average treatment effects of approximately 0.21, indicating a comparable increase in purchase likelihood under treatment. In contrast, Model 2 produces a noticeably smaller average treatment effect, at approximately 0.17, despite inducing higher levels of user engagement. This reduction reflects the stronger competitive dynamics generated by repeated clicks under Model 2. Although repeated exposure raises the likelihood that an individual user purchases an item, it simultaneously intensifies competition among users for the same set of items, thereby limiting the number of purchases that can ultimately be realized as a result of treatment. As a consequence, a larger share of the induced purchase intent does not translate into additional completed transactions. Taken together, these findings demonstrate that interference mechanisms can substantially reduce observed treatment effects, even in settings where individual-level purchase tendencies increase in response to treatment.

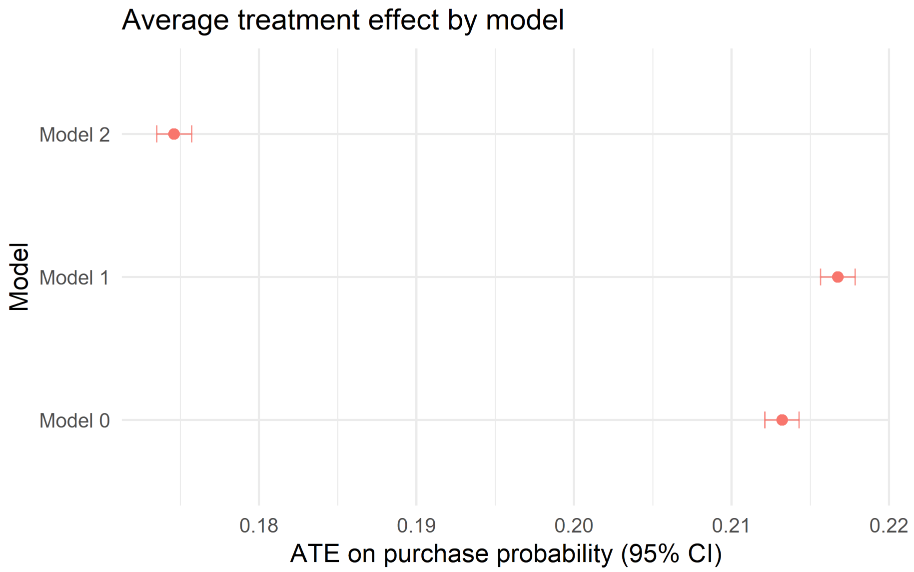


Figure 2. Estimated average treatment effect on the probability of at least one purchase across models. Model 2 exhibits a smaller ATE, reflecting stronger competitive interference under repeated clicks.

Potential buy intensity and value per unit of aggressiveness. To separate treatment impact from model aggressiveness, both the average number of potential purchase opportunities generated per user and the efficiency with which this increased activity translates into realized outcomes were examined. As shown in Figure 3, treated users generate more potential purchases than control users across all models, although the magnitude of this increase varies meaningfully. Model 2 is the most aggressive, producing nearly four potential purchase opportunities per treated user on average, compared with approximately 3.4 and 3.7 for Model 0 and Model 1, respectively.However, this higher level of induced activity does not translate proportionally into realized purchases, consistent with diminishing returns arising from interference effects. To capture this trade-off, the value per unit of potential buy uplift is computed, defined as the ratio of the average treatment effect to the treatment-induced increase in potential purchase intensity (Figure 4). By this metric, Model 0 delivers the highest value per unit of aggressiveness, followed by Model 1, while Model 2 performs substantially worse. Although Model 1 achieves the largest raw ATE, it does so by inducing a greater volume of activity, resulting in lower marginal returns than Model 0. Taken together, these findings highlight the importance of evaluating not only the magnitude of treatment effects, but also the efficiency with which increased model aggressiveness converts potential opportunities into realized outcomes.

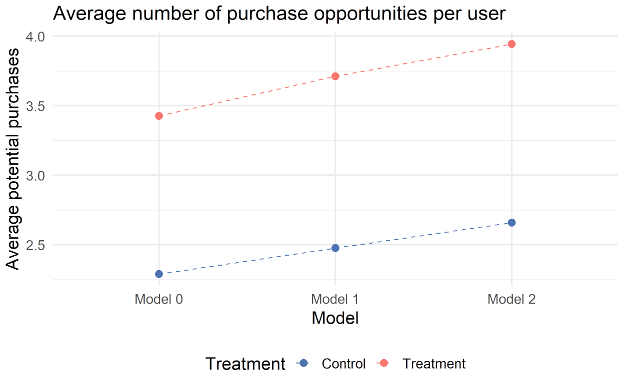


Figure 3. Potential purchase intensity by model and treatment status. Treated users generate more potential buys than control users, though the increase varies across models.

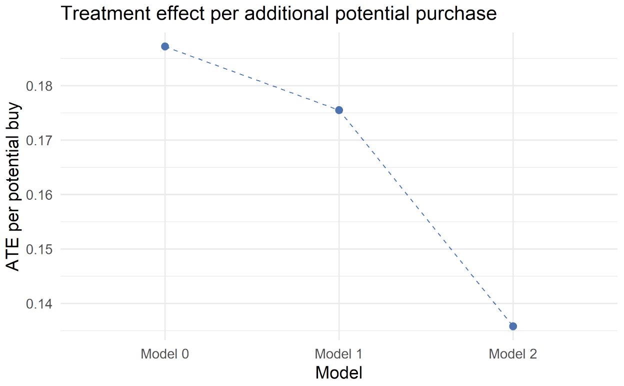


Figure 4. Value per unit of aggressiveness across models, calculated as the ratio of ATE to the treatment-induced increase in potential purchase intensity. Model 0 delivers the highest marginal return.

Conversion efficiency. To formalize differences in how models translate exposure into outcomes, conversion efficiency is defined as the ratio of the realized purchase probability to the average number of potential purchase opportunities. Higher values indicate more effective conversion of exposure into purchases. Figure 5 shows the implied efficiency frontiers for treatment and control users. In both settings, models nearer the upper-left boundary achieve higher purchase probabilities with fewer potential buy opportunities. Under treatment, Model 0 is the most efficient, followed closely by Model 1, while Model 2 is consistently less efficient, requiring substantially more exposure per purchase. This inefficiency is most evident under treatment: although Model 2 induces the largest increase in exposure, it yields disproportionately smaller outcome gains and lies strictly inside the efficiency frontier formed by Model 0 and Model 1, making it Pareto-dominated in outcome generation per unit of aggressiveness.

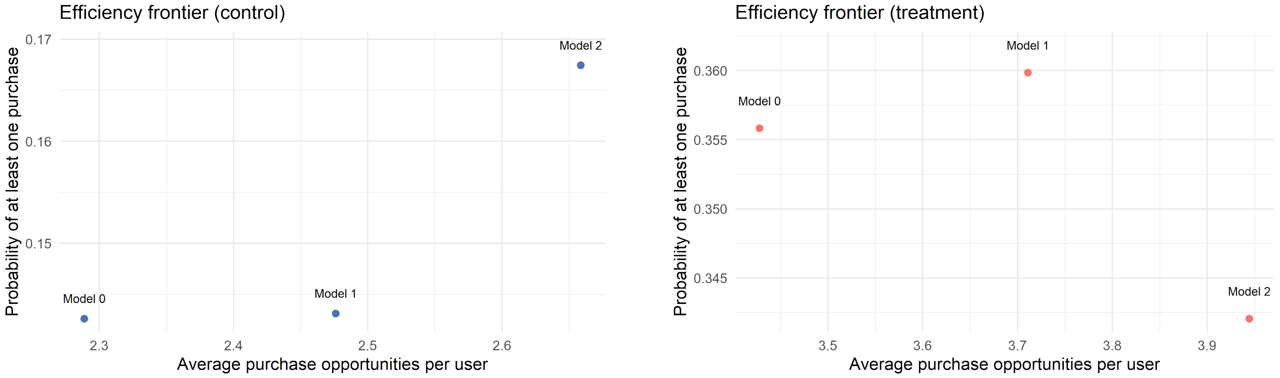


Figure 5. Conversion efficiency frontiers for control (left) and treatment (right) users. Each point corresponds to a model, plotting the final-buy rate against potential buy intensity. Models closer to the upper-left region convert purchase opportunities into realized outcomes more efficiently. Under treatment, Model 2 exhibits lower efficiency, lying inside the frontier formed by Model 0 and Model 1.

Statistical uncertainty and robustness. Statistical uncertainty is assessed by constructing user-level confidence intervals for ATE using normal approximations. Figure 6 shows the resulting ATE estimates and 95% confidence intervals against the corresponding increase in potential purchase opportunities induced by treatment. All models exhibit tight confidence intervals, indicating that the estimated treatment effects are precisely measured and that differences across models are statistically meaningful. When plotting treatment effects against exposure intensity, a clear negative relationship emerges: models that induce larger increases in potential buying opportunities tend to exhibit lower causal effectiveness per unit of exposure. Model 0 dominates this trade-off, achieving a large and precisely estimated treatment effect while inducing only a modest increase in potential purchase intensity. In contrast, Model 2 substantially increases exposure but delivers a markedly smaller treatment effect, reinforcing the conclusion that greater aggressiveness does not necessarily translate into higher causal impact.

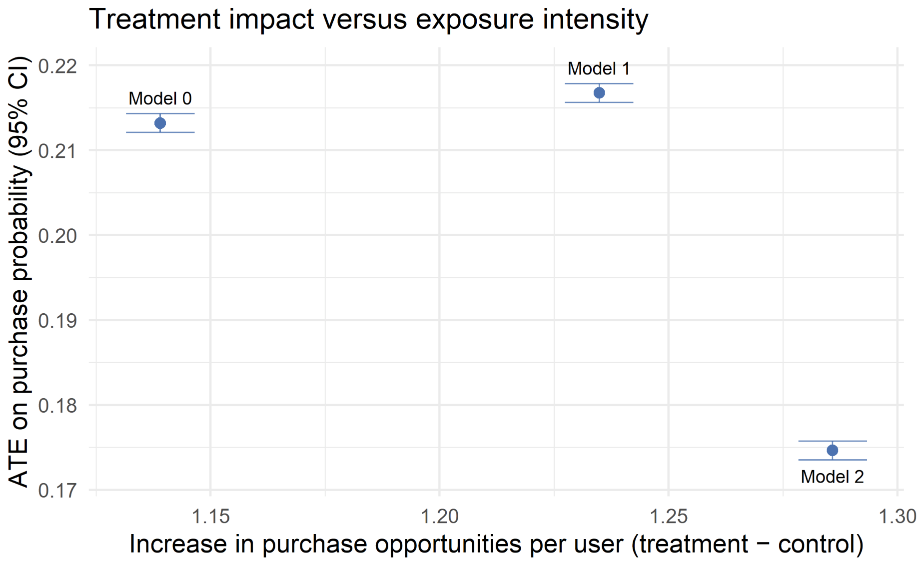


Figure 6. Average treatment effect with 95% confidence intervals plotted against the increase in potential buy intensity induced by treatment. Models with higher aggressiveness exhibit lower causal efficiency, with Model 0 dominating the trade-off.

Overall, the comparison reveals that more complex and aggressive interaction models do not necessarily yield better causal performance. While repeated clicks dynamics may appear behaviorally realistic, they can increase interference and reduce treatment effects at the user level. From an experimental design perspective, these findings highlight the importance of accounting for interference mechanisms when evaluating interventions in marketplace settings. Models that generate excessive interaction intensity may overstate engagement while understating true causal impact. Further analysis considers Model 1 as the independent click model (ICM) and Model 2 as the repeated clicks model (RCM), showing that distinct interference frameworks can create similar challenges for ATE estimation.

3.5 ATE estimates under varying treatment probabilities

This section analyzes how treatment probability, randomization scheme, and interference structure jointly affect ATE estimation. Using known ground-truth ATEs from the no-interference model, the results allow for a direct assessment of bias and estimator behavior under controlled violations of SUTVA. For clarity, results for a representative low treatment probability ($p_{\mathrm{T}} = 0.30$) are visualized in the main text and corresponding figures for higher treatment probabilities are reported in Appendix 2.

3.5.1 Bernoulli randomization

Tables 2 and 3 report Bernoulli-randomized ATE estimates for the independent click and repeated clicks models, respectively. Across all treatment probabilities, Bernoulli randomization consistently produces positively biased estimates relative to the true ATE. Under the independent click model, the bias is largest at lower treatment probabilities. When the true ATE is 0.10 ($p_{\mathrm{T}} = 0.30$), the estimated ATE exceeds 0.21, more than doubling the true effect. Figure 7 illustrates this behavior for the independent click model at $p_{\mathrm{T}} = 0.30$, highlighting the substantial upward bias induced by asymmetric exposure to treated competitors under Bernoulli assignment. As the treatment probability increases, the bias shrinks substantially, falling to 0.018 when $p_{\mathrm{T}} = 0.50$. This pattern reflects the fact that interference effects become increasingly symmetric as treatment becomes more prevalent, reducing the relative advantage of treatment.

The repeated clicks model exhibits a similar pattern, with bias slightly lower at low treatment probabilities. The diminishing-returns structure reduces the marginal effect of repeated exposures, which limits (but does not eliminate) the inflation caused by competitive interference. This behavior is visualized in Figure 8 for $p_{\mathrm{T}} = 0.30$, with corresponding figures for higher treatment probabilities reported in Appendix 2.

The narrow confidence intervals observed in all Bernoulli settings indicate that the bias is not driven by sampling variability but is instead systematic. This is particularly important from a practical standpoint: large-scale online experiments may yield highly precise yet substantially biased estimates if interference is ignored.

Treatment probability	True ATE	Estimated ATE	Bias	95% CI
0.30	0.10	0.217	0.117	0.216–0.218
0.45	0.25	0.302	0.052	0.301–0.303
0.50	0.30	0.318	0.018	0.317–0.319

Table 2. Bernoulli-randomized estimates of the treatment effect, bias, and 95% confidence intervals under the independent click model.

Treatment probability	True ATE	Estimated ATE	Bias	95% CI
0.30	0.10	0.175	0.075	0.174–0.176
0.45	0.25	0.286	0.036	0.285–0.287
0.50	0.30	0.306	0.006	0.305–0.307

Table 3. Bernoulli-randomized estimates of the treatment effect, bias, and 95% confidence intervals under the repeated clicks model.

3.5.2 Cluster randomization

Tables 4 and 5 summarize the ATE estimates obtained under cluster randomization for the independent click and repeated clicks models, respectively. Relative to Bernoulli randomization, cluster randomization markedly reduces bias in low-treatment regimes. At $p_{\mathrm{T}} = 0.30$, the independent click model bias drops from 0.117 under Bernoulli randomization to 0.038 under cluster randomization. This reduction is illustrated in Figure 7, where cluster randomization substantially narrows the gap between the estimated and true ATE at $p_{\mathrm{T}} = 0.30$. However, as treatment probability increases, cluster-randomized estimates become negatively biased, underestimating the true ATE by up to 0.09 at $p_{\mathrm{T}} = 0.50$. When a large fraction of clusters is treated, untreated users continue to experience substantial exposure to treated competitors, reducing experimental contrast and diminishing the observed effects. A similar pattern occurs under the repeated clicks model. Figure 8 shows that clustering substantially lowers bias at $p_{\mathrm{T}} = 0.30$, while additional results in Appendix 2 indicate that bias increases as the treatment probability rises.

Treatment probability	True ATE	Estimated ATE	Bias	95% CI
0.30	0.10	0.138	0.038	0.138–0.139
0.45	0.25	0.200	−0.050	0.199–0.200
0.50	0.30	0.212	−0.088	0.212–0.213

Table 4. Cluster-randomized estimates of the treatment effect, bias, and 95% confidence intervals under the independent click model.

Treatment probability	True ATE	Estimated ATE	Bias	95% CI
0.30	0.10	0.116	0.016	0.115–0.117
0.45	0.25	0.194	−0.056	0.193–0.194
0.50	0.30	0.208	−0.092	0.207–0.209

Table 5. Cluster-randomized estimates of the treatment effect, bias, and 95% confidence intervals under the repeated clicks model.

3.5.3 Comparison across treatment probabilities

Across both models and randomization schemes, the probability of treatment plays a central role in determining estimator behavior and the resulting bias patterns. At low treatment probabilities, upward bias is clearly noticeable under Bernoulli randomization. This occurs because treated users experience relatively little competition from other treated users, which amplifies their apparent treatment effect, while control users are minimally exposed to treated competitors and thus show little change. As the treatment probability increases, the interference becomes more symmetric, reducing this inflation.

In contrast, under cluster randomization, high treatment probabilities create the opposite challenge. When a large proportion of clusters are treated, spillover effects can overwhelm the system, reducing differences between treated and control clusters and thereby lowering observable treatment effects. The illustrative figures for $p_{\mathrm{T}} = 0.30$ (Figures 7 and 8) highlight these opposing dynamics: Bernoulli randomization tends to overstate treatment effects in low-exposure settings, whereas cluster randomization partially mitigates bias by structuring treatment assignment to align with the natural pathways of interference. Additional results for higher treatment probabilities, presented in Appendix 2, confirm that these dynamics persist and often intensify as treatment coverage expands, emphasizing the critical role of both randomization design and treatment intensity in shaping causal estimates in the presence of interference.

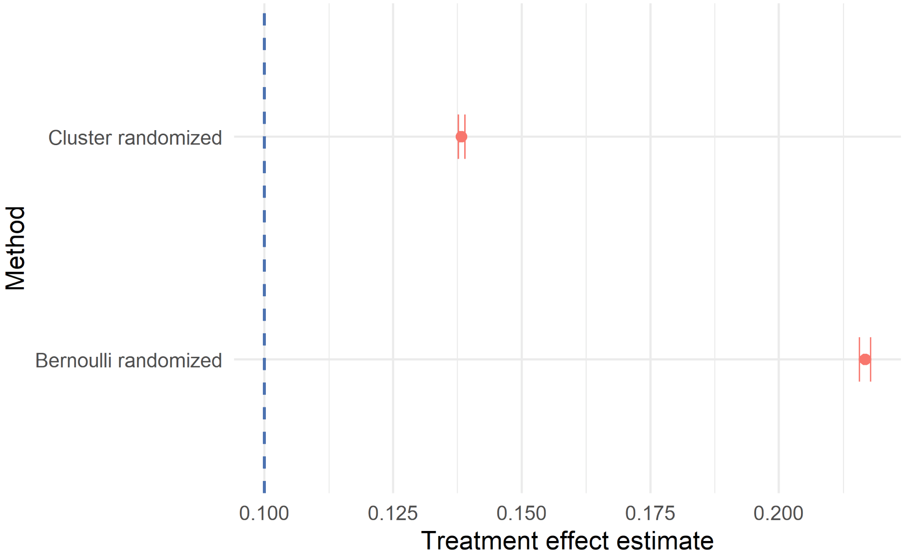


Figure 7. ATE estimates for the independent click model at $p_{\mathrm{T}} = 0.30$. Bernoulli randomization is upward biased, while cluster randomization reduces bias.

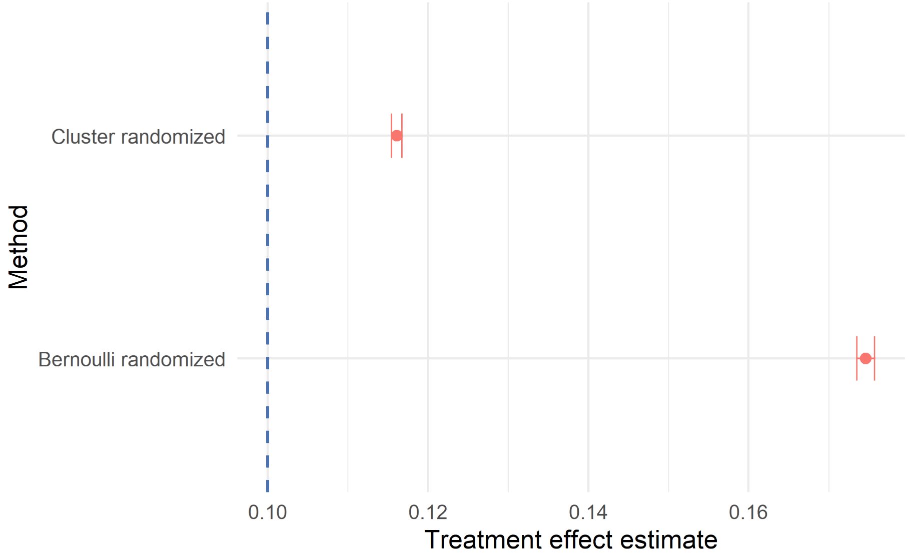


Figure 8. ATEs for the repeated clicks model at $p_{\mathrm{T}} = 0.30$, showing reduced bias under cluster randomization.

3.6 Exposure mapping and decomposition of interference effects

To move beyond design-based comparisons and explicitly quantify interference effects, an exposure-mapping approach was adopted. Exposure mapping was applied separately to the independent click model (ICM) and the repeated clicks model (RCM). For each user, exposure was defined as the leave-one-out fraction of treated competitors interacting with the same items. This measure captured the intensity of interference while avoiding mechanical correlation with own treatment status. In the RCM, user-level click intensity was additionally included to account for heterogeneity in repeated engagement.

At the user level, the following linear probability model was estimated:

$$\mathrm{final\_buy}_u = \alpha + \beta \cdot \mathrm{Treat}_u + \gamma \cdot \mathrm{Exposure}_u + \delta \cdot (\mathrm{Treat}_u \times \mathrm{Exposure}_u) + \eta \cdot k_u + \varepsilon_u,$$

where $\eta = 0$ for the ICM and is estimated for the RCM. The coefficient β captures the direct treatment effect at zero exposure, γ measures competitive spillovers, and δ describes how exposure moderates the treatment effect.

3.6.1 Regression results and treatment effect decomposition

Table 6 reports the results for the ICM. The intercept is estimated at $\hat{\alpha} = 0.291$, implying a baseline purchase probability of about 29.1% for control users facing no treated competitors. The direct treatment effect is large and positive, $\hat{\beta} = 0.159$, while exposure has a strong negative effect, $\hat{\gamma} = -0.423$. The interaction term is also negative, $\hat{\delta} = -0.041$, indicating that competitive exposure weakens the benefit of treatment. All coefficients are highly statistically significant with item-clustered standard errors.

Variable	Estimate	Std. Error (clustered)	p-value
Intercept	0.2910	0.00245	< 0.001
Treatment (Z_i)	0.1586	0.00373	< 0.001
Exposure ($\bar{E}_i$)	-0.4227	0.00487	< 0.001
Treatment × Exposure	-0.0414	0.00746	< 0.001

Table 6. Exposure-mapped regression for the independent click model (ICM). The dependent variable is a binary indicator for whether the user makes at least one purchase. Standard errors are clustered at the item level.

The corresponding results for the RCM are shown in Table 7. The baseline intercept decreases to $\hat{\alpha} = 0.189$, and the direct treatment effect remains positive but smaller, at $\hat{\beta} = 0.141$. Exposure continues to exert a strong negative impact, $\hat{\gamma} = -0.431$, while the interaction term becomes more negative, $\hat{\delta} = -0.051$, pointing to a stronger reduction in the treatment effect under repeated competition. Click intensity enters positively and significantly, with $\hat{\eta} = 0.0173$, confirming that more active users are more likely to convert.

Variable	Estimate	Std. Error (clustered)	p-value
Intercept	0.1892	0.00255	< 0.001
Treatment (Z_i)	0.1413	0.00381	< 0.001
Exposure ($\bar{E}_i$)	-0.4312	0.00502	< 0.001
Click intensity (k_i)	0.0173	0.00008	< 0.001
Treatment × Exposure	-0.0514	0.00762	< 0.001

Table 7. Exposure-mapped regression for the repeated clicks model (RCM), controlling for user click intensity. The dependent variable is a binary indicator for whether the user makes at least one purchase. Standard errors are clustered at the item level.

These estimates allow a transparent separation of direct and total effects. For the ICM, the direct ATE equals 0.159, whereas the total ATE evaluated at mean exposure is approximately 0.138. For the RCM, the direct ATE is 0.141, and the exposure-adjusted total ATE declines further to about 0.116. The gap between direct and total effects quantifies the loss in effectiveness induced by competitive spillovers and is larger in the presence of repeated interactions.

3.6.2 Exposure–response patterns and implications

Figure 9 provides a visual representation of these patterns by plotting exposure–response curves for treated and control users under both models. In each case, purchase probability declines monotonically with exposure, reflecting intensified competition for shared items. Treated users achieve higher conversion rates at all exposure levels, consistent with a positive direct effect. However, the gap between treated and control users narrows as exposure increases, visually reinforcing the negative interaction effects reported in Tables 6 and 7. The curves for the RCM exhibit steeper declines and a smaller treatment–control gap at high exposure, consistent with a larger reduction in treatment effects implied by the regression estimates. This pattern highlights the role of repeated competition in intensifying interference effects.

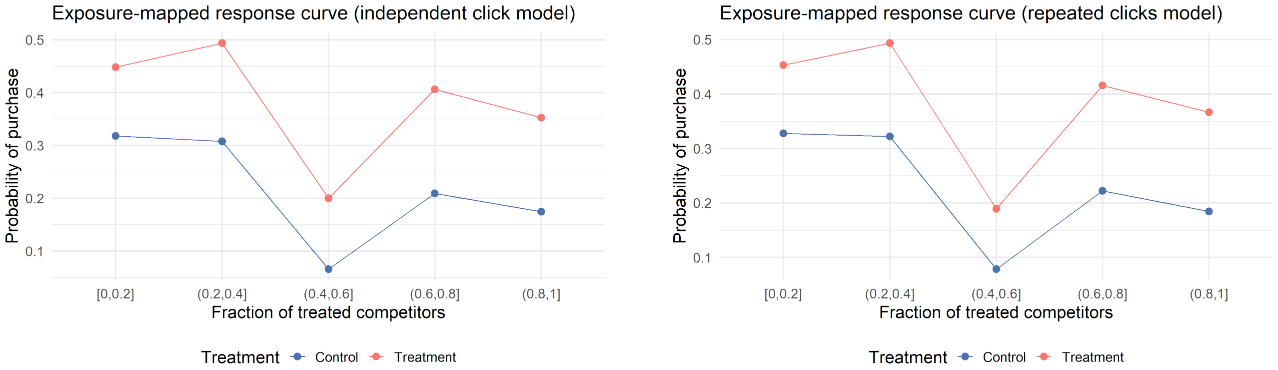


Figure 9. Exposure–response curves for the independent click model (left) and the repeated clicks model (right) at $p_{\mathrm{T}} = 0.30$. Purchase probability declines with exposure for both groups, and the treatment effect attenuates as competition increases, more strongly under the RCM.

Overall, exposure mapping shows that treatment effects in shared-inventory marketplaces are inherently conditional on the competitive environment. While direct gains from treatment are substantial at low exposure, spillovers from treated competitors progressively compress these gains as exposure rises. Explicitly modeling exposure therefore enables a clearer distinction of direct and interference effects and helps explain why naive estimators can misrepresent causal impacts under different randomization schemes.

Results and conclusions

An extensive simulation-based analysis of causal inference under interference in shared-inventory marketplaces is performed on the FINN.no slate dataset. Three outcome-generating models are considered: Model 0, a baseline independent click model restricted to first clicks only; Model 1, an independent click model retaining all interactions to capture repeated engagement; and Model 2, a repeated clicks model with diminishing returns, reflecting how repeated interactions increase purchase probability. These models capture progressively higher levels of interaction intensity and user competition. Treatment effects are evaluated under Bernoulli and cluster randomization schemes for multiple treatment probabilities, and exposure mapping is used to distinguish direct and spillover effects at the user level.

It is found that, in general, more aggressive interaction dynamics lead to lower causal efficiency. Although the independent click and baseline models yield similar average treatment effects of around 0.21, the repeated clicks model produces a smaller effect of approximately 0.17, despite generating the highest number of potential purchase opportunities. Conversion efficiency and value-per-aggressiveness measures consistently indicate that the baseline model dominates, while the repeated clicks model is strictly less efficient, lying inside the efficiency frontier formed by the simpler models. This suggests that repeated exposure intensifies competition and weakens realized outcomes, so that higher activity does not translate proportionally into higher causal impact.

Across all models, Bernoulli randomization is shown to induce substantial upward bias when treatment probabilities are low. For a treatment probability of 0.30, the estimated ATE in the independent click model exceeds the true effect by more than 0.11, more than doubling the ground-truth ATE. This inflation decreases as the treatment probability increases and exposure becomes more symmetric. In contrast, cluster randomization substantially reduces bias at low treatment probabilities, lowering the bias at $p_{\mathrm{T}} = 0.30$ to below 0.04, but produces considerable negative bias at higher treatment probabilities, reaching nearly -0.09 at $p_{\mathrm{T}} = 0.50$. In all cases, confidence intervals remain narrow, indicating that the observed distortions are systematic rather than driven by sampling variability.

Exposure mapping reveals a strong and robust pattern of competitive spillovers. In both the independent click and repeated clicks models, exposure to treated competitors has a large negative effect on purchase probability, and the interaction between treatment and exposure further reduces the treatment benefit. The direct treatment effect at zero exposure equals 0.159 in the independent click model and 0.141 in the repeated clicks model, while the corresponding total effects evaluated at mean exposure decline to approximately 0.138 and 0.116, respectively. Exposure–response curves confirm a monotonic decline in conversion as exposure increases, with steeper slopes under repeated interactions, indicating that diminishing returns amplify the loss from competitive interference.

Taken together, the results demonstrate that interference fundamentally reshapes both the magnitude and the interpretation of treatment effects in marketplace experiments. More complex and behaviorally realistic models do not necessarily yield better causal performance. Instead, they may intensify competition and reduce efficiency. Bernoulli randomization tends to overstate treatment effects when exposure is sparse, whereas cluster randomization mitigates this inflation but risks attenuation as treatment becomes widespread. Exposure-based analysis provides a clear explanation for these patterns and quantifies the gap between direct and realized effects.

Several directions for further research follow from these findings. Alternative randomization schemes that adapt to expected exposure patterns could be explored to balance inflation and attenuation across treatment probabilities. Richer interaction structures, such as endogenous supply, network effects, or dynamic user responses, may further illuminate how interference evolves over time. In addition, extending exposure mapping to nonlinear or machine learning based response models could capture heterogeneous spillovers more flexibly.

In conclusion, this thesis shows that ignoring interference in shared-inventory marketplaces can lead to treatment effect estimates that are highly precise yet systematically biased. Appropriate randomization, and explicit accounting for exposure are therefore essential for obtaining credible and actionable causal insights in modern digital experimentation.

References and sources

[1] P. M. Aronow, C. Samii. *Estimating average causal effects under general interference, with application to a social network experiment.* 2017. URL: `https://arxiv.org/pdf/1305.6156`.

[2] S. Athey, G. Imbens. *The State of Applied Econometrics - Causality and Policy Evaluation.* 2017. URL: `https://arxiv.org/pdf/1607.00699`.

[3] K. Bimpikis, S. Ehsani, R. Ilkiliç. "Cournot Competition in Networked Markets." In: *Management Science* 65.6 (2019), pages 2467–2487.

[4] T. Blake, D. Coey. "Why marketplace experimentation is harder than it seems: The role of test-control interference." In: *Proceedings of the Fifteenth ACM Conference on Economics and Computation.* 2014, pages 567–582.

[5] A. Boyarsky, H. Namkoong, J. Pouget-Abadie. *Modeling Interference Using Experiment Roll-out.* 2023. URL: `https://arxiv.org/pdf/2305.10728v2`.

[6] E. Calvano, G. Calzolari, V. Denicolò, S. Pastorello. "Algorithmic Pricing: What Implications for Competition Policy?" In: *Review of Industrial Organization* 56 (2020), pages 155–171.

[7] D. R. Cox. *Planning of Experiments.* pp. 70–80. New York, NY, USA: Wiley, 1958.

[8] D. Eckles, B. Karrer, J. Ugander. *Design and Analysis of Experiments in Networks: Reducing Bias from Interference.* 2014. URL: `https://arxiv.org/pdf/1404.7530`.

[9] S. Eide, D. S. Leslie, A. Frigessi. *Dynamic Slate Recommendation with Gated Recurrent Units and Thompson Sampling.* 2021. URL: `https://arxiv.org/pdf/2104.15046`.

[10] FINN.no. *FINN Slate Recommendation Dataset.* `https://github.com/finn-no/recsys-slates-dataset`. 2022. (Viewed 2025-12-14).

[11] A. Fradkin, G. Johnson, M. Schwarz. "Parallel Experimentation and Competitive Interference on Online Advertising Platforms." In: *Proceedings of the ACM Conference on Economics and Computation (EC).* 2022, pages 451–469.

[12] K. Han, S. Li, J. Mao, H. Wu. *Detecting Interference in A/B Testing with Increasing Allocation.* 2023. URL: `https://arxiv.org/pdf/2211.03262`.

[13] D. Holtz, R. Lobel, I. Liskovich, S. Aral. *Reducing Interference Bias in Online Marketplace Pricing Experiments.* 2020. URL: `https://arxiv.org/pdf/2004.12489`.

[14] M. G. Hudgens, M. E. Halloran. "Toward Causal Inference with Interference." In: *Journal of the American Statistical Association* 103.482 (2008), pages 832–842.

[15] G. W. Imbens, D. B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* Cambridge, UK: Cambridge University Press, 2015, pages 3–30.

[16] R. Johari, H. Li, I. Liskovich, G. Weintraub. *Experimental Design in Two-Sided Platforms: An Analysis of Bias.* 2021. URL: `https://arxiv.org/pdf/2002.05670`.

[17] R. Johari, H. Li, A. Murthy, G. Y. Weintraub. *When Does Interference Matter? Decision-Making in Platform Experiments.* 2025. URL: `https://arxiv.org/pdf/2410.06580`.

[18] M. P. Leung. "Causal Inference Under Approximate Neighborhood Interference." In: *Econometrica* 88.1 (2020), pages 267–293.

[19] H. Li, G. Zhao, R. Johari, G. Y. Weintraub. *Interference, Bias, and Variance in Two-Sided Marketplace Experimentation: Guidance for Platforms.* 2021. URL: `https://arxiv.org/pdf/2104.12222`.

[20] X. Ma, Y. Zhou, Y. Li, Y. Yu, P. Hui. "Correcting for Interference in Experiments: A Case Study at Douyin." In: *Proceedings of the ACM Web Conference.* 2023, pages 2405–2415.

[21] C. F. Manski. "Identification of Endogenous Social Effects: The Reflection Problem." In: *Review of Economic Studies* 60.3 (1993), pages 531–542.

[22] OpenAI. *ChatGPT.* 2025. URL: `https://www.openai.com/` (viewed 2025-12-19).

[23] A. Reklaitė, J. Gamper. "Offline assessment of interference effects in a series of AB tests." In: *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE '22).* 2022, pages 262–263.

[24] D. B. Rubin. "Randomization Analysis of Experimental Data: The Fisher Randomization Test." In: *Journal of the American Statistical Association* 75.371 (1980), pages 591–593.

[25] M. Saveski, J. Pouget-Abadie, G. Saint-Jacques, W. Duan, S. Ghosh, Y. Xu, E. M. Airoldi. "Detecting Network Effects: Randomizing Over Randomized Experiments." In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2017, pages 1027–1035.

[26] F. Sävje, P. M. Aronow, M. G. Hudgens. *Average treatment effects in the presence of unknown interference.* 2021. URL: `https://arxiv.org/pdf/1711.06399`.

[27] M. E. Sobel. "What Do Randomized Studies of Housing Mobility Demonstrate?" In: *Journal of the American Statistical Association* 101.476 (2006), pages 1398–1407.

[28] E. J. Tchetgen Tchetgen, T. J. VanderWeele. "On Causal Inference in the Presence of Interference." In: *Statistical Methods in Medical Research* 21.1 (2012), pages 55–75.

Appendix 1. EDA graphs

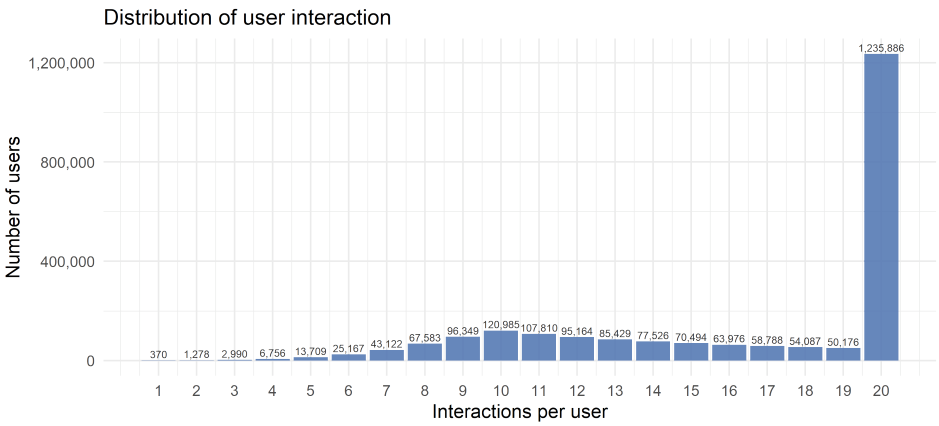


Figure 10. Distribution of the number of interactions per user

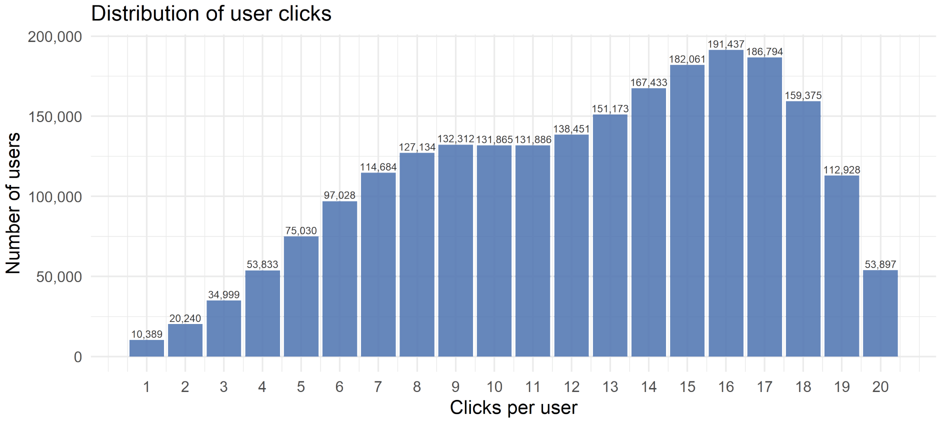


Figure 11. Distribution of the number of clicks per user

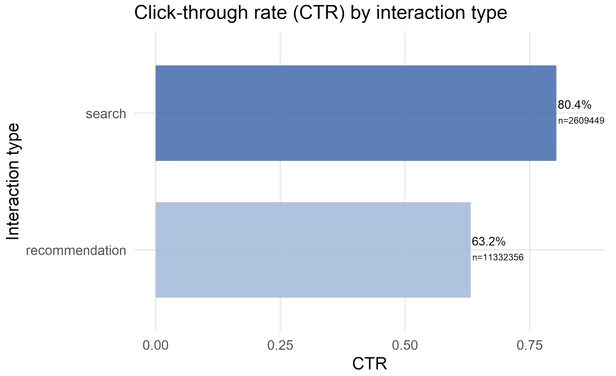


Figure 12. Summary statistics of interactions across users

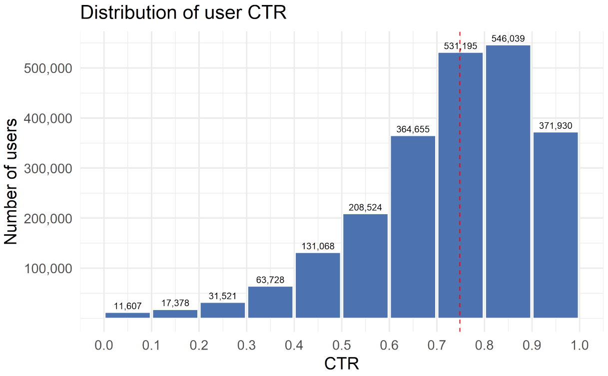


Figure 13. Distribution of user-level click-through rates

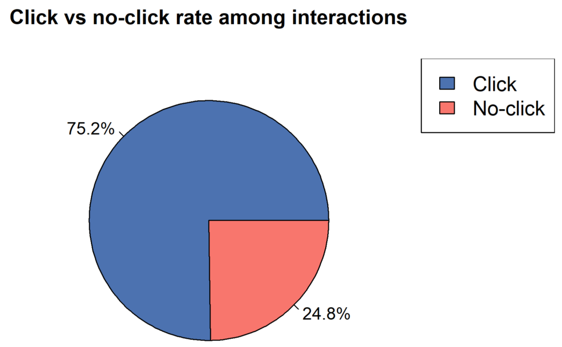


Figure 14. Proportion of click versus no-click events in the dataset

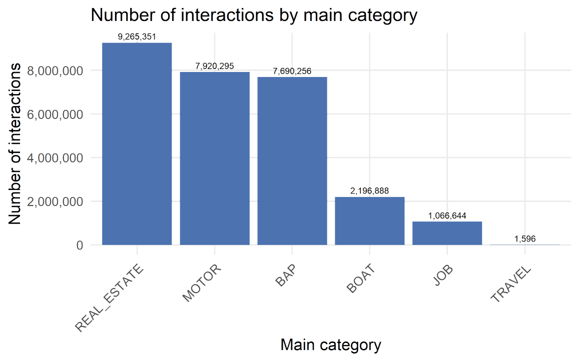


Figure 15. Top item categories by frequency of interactions in the dataset

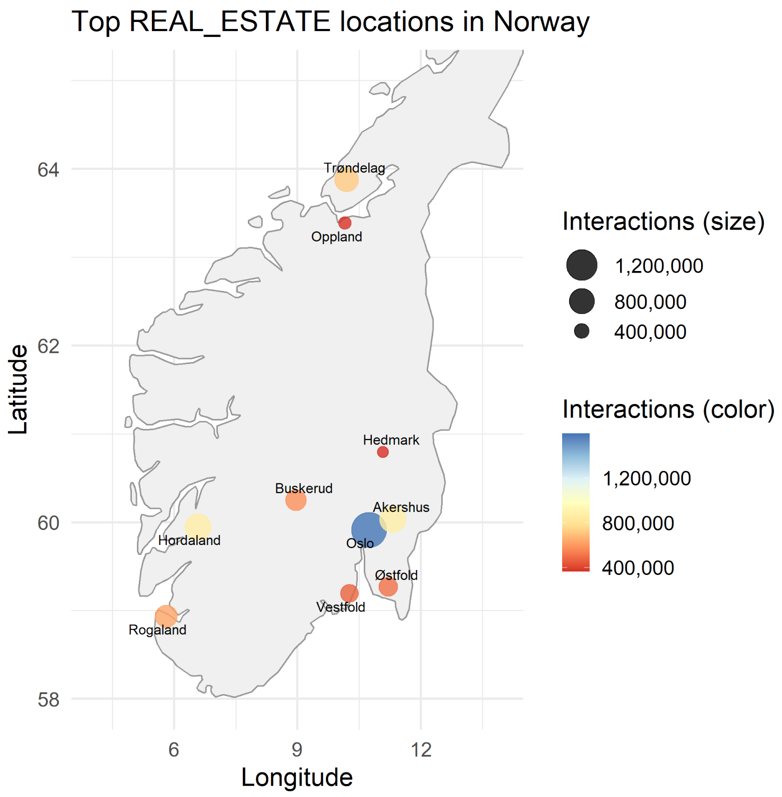
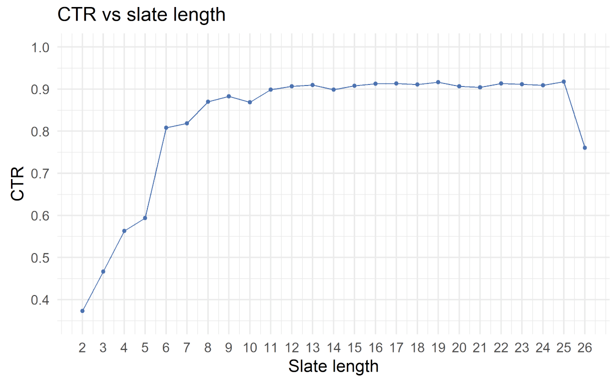


Figure 16. Geographical distribution of real estate listings based on item locations



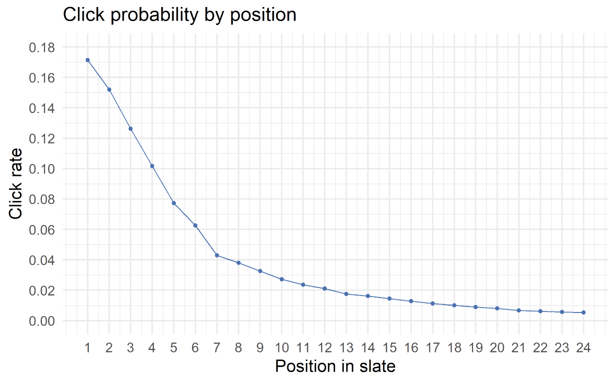


Figure 17. Click-through rate as a function of slate length

Figure 18. Distribution of clicks by position within the slate

Appendix 2. Additional ATE figures

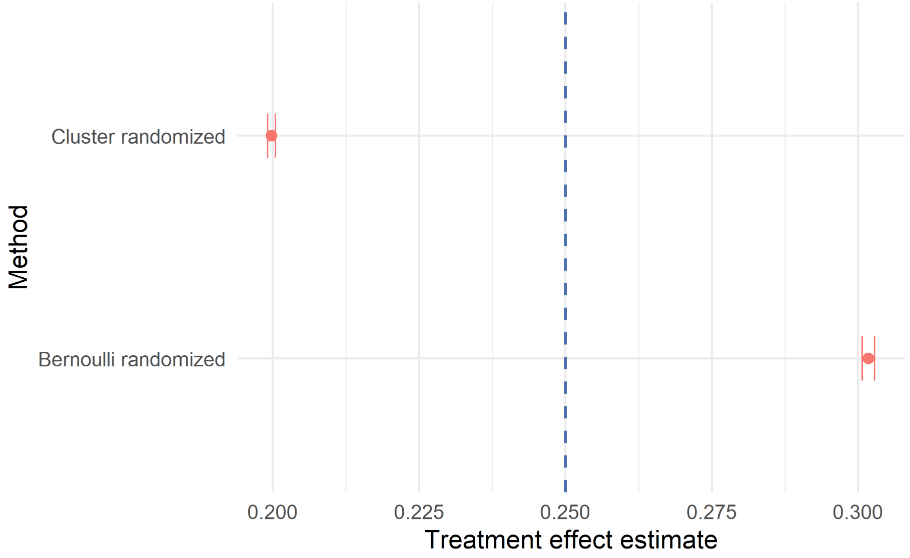


Figure 19. ATE estimates for the independent click model under Bernoulli and cluster randomization at $p_{\mathrm{T}} = 0.45$, with 95% confidence intervals

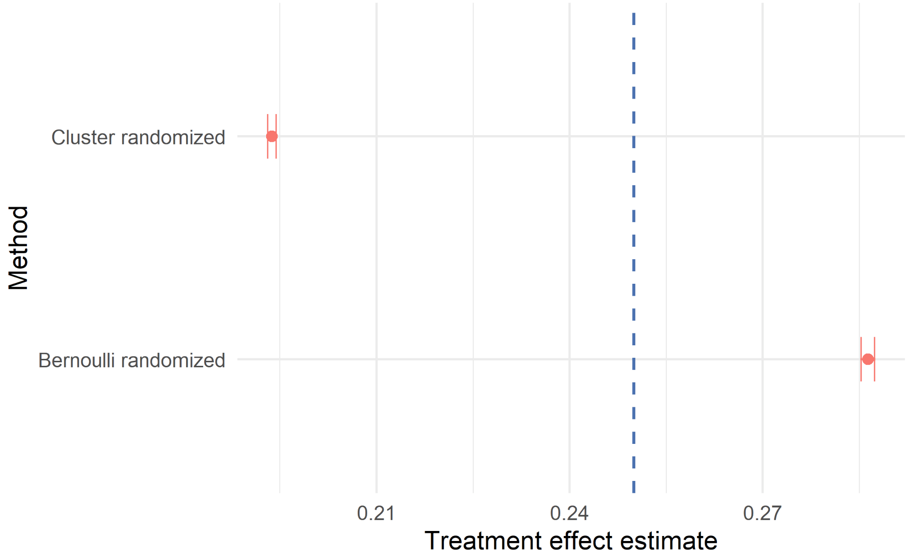


Figure 20. ATE estimates for the repeated clicks model under Bernoulli and cluster randomization at $p_{\mathrm{T}} = 0.45$, with 95% confidence intervals

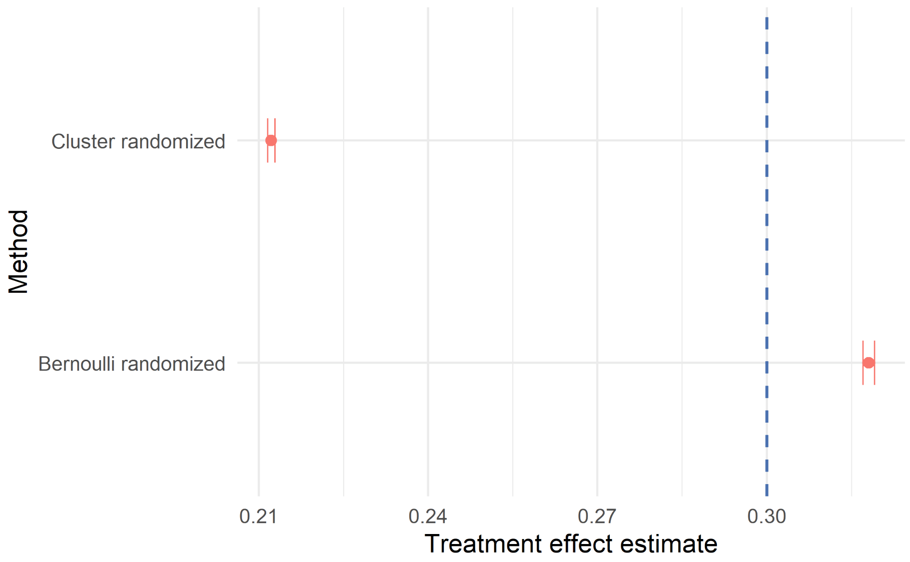


Figure 21. ATE estimates for the independent click model under Bernoulli and cluster randomization at $p_{\mathrm{T}} = 0.50$, with 95% confidence intervals

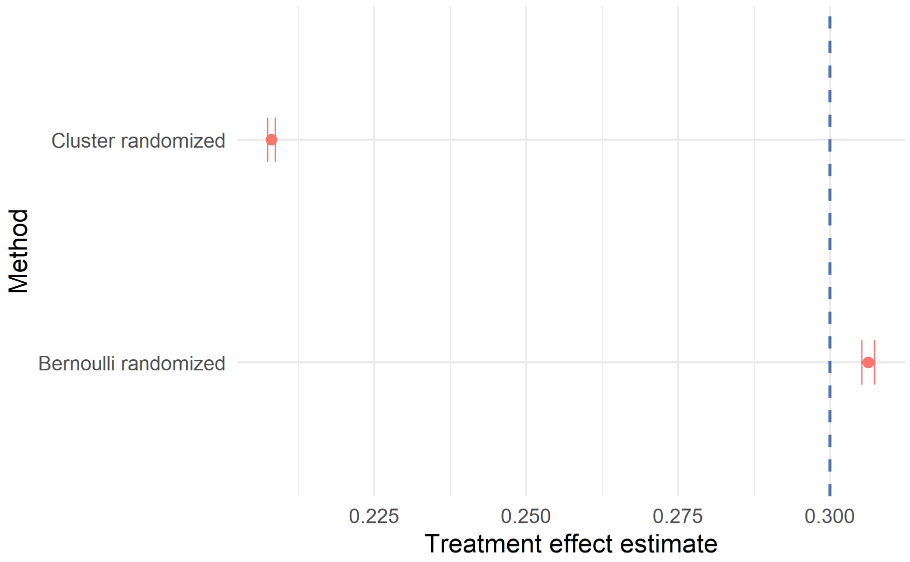


Figure 22. ATE estimates for the repeated clicks model under Bernoulli and cluster randomization at $p_{\mathrm{T}} = 0.50$, with 95% confidence intervals

Appendix 3. Declaration of tool usage

During the preparation of this thesis, the following tools were used to support the writing and editing process:

1. *Grammarly* was used to check grammar, spelling, and stylistic consistency to improve the clarity and readability of the text.
2. *ChatGPT* was used to support language improvement, identify potential textual errors, assist with idea generation and code optimization.

The content and findings presented in this thesis are the result of the author's original work, with the mentioned tools serving solely supplementary purposes.

Appendix 4. Software code

In this thesis, data preprocessing, including downloading and transforming the dataset, was implemented in Python and can be accessed via a Jupyter notebook on Google Colab: FINN.no Slate Dataset preprocessing.ipynb. The R programming language was then used to perform data analysis and subsequent modeling.

```
library(here)
library(arrow)
library(dplyr)
library(data.table)
library(ggplot2)
library(plotrix)
library(stringr)
library(tidygeocoder)
library(ggrepel)
library(igraph)
library(sandwich)
library(lmtest)

# ---------- Read data ----------
path <- here()
model_dir <- file.path(path, "models")
dir.create(model_dir, recursive = TRUE, showWarnings = FALSE)

data_files <- list.files(file.path(path, "data_chunks"),
                         pattern = "\\.parquet$", full.names = TRUE)

data_df <- lapply(data_files, read_parquet) %>% bind_rows()

# remove column slate_items
data_without_slate_items <- data_df %>% select(-slate_items)

# save the data for later use
saveRDS(data_without_slate_items,
        file = file.path(model_dir, "data_without_slate_items.rds"))

# load saved data
data_without_slate_items <- readRDS(
  file.path(model_dir, "data_without_slate_items.rds"))
```

```
# ---------- Exploratory data analysis (EDA) ----------
# convert to data.table
data <- setDT(data_without_slate_items)

## 1. Missing values ----
sapply(data, function(x) sum(is.na(x)))

## 2. Dataset size & core dimensions ----
data[, .(n_rows = .N, n_users = uniqueN(user_id),
         n_items = uniqueN(clicked_item_id),
         n_categories = uniqueN(category),
         n_interaction_types = uniqueN(interaction_type))]

## 3. User-level interaction distribution ----
# total interactions per user
user_interacts <- data[, .N, by = user_id]
interact_counts <- user_interacts[, .(user_count = .N),
                                  by = .(interactions = N)]

ggplot(interact_counts, aes(x = interactions, y = user_count)) +
  geom_col(fill = "#4C72B0", alpha = 0.85) +
  geom_text(aes(label = scales::comma(user_count)),
            vjust = -0.3, size = 3, alpha = 0.8) +
  scale_x_continuous(
    breaks = seq(min(interact_counts$interactions),
                 max(interact_counts$interactions), by = 1)) +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Distribution of user interaction",
       x = "Interactions per user", y = "Number of users") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18))

# create a 'clicked' column for easier filtering
data[, clicked := clicked_item_id != 1]

## 4. User-level click distribution ----
# clicks per user
user_clicks <- data[clicked == TRUE, .N, by = user_id]
click_counts <- user_clicks[, .(user_count = .N), by = .(clicks = N)]
```

```
ggplot(click_counts, aes(x = clicks, y = user_count)) +
  geom_col(fill = "#4C72B0", alpha = 0.85) +
  geom_text(aes(label = scales::comma(user_count)),
            vjust = -0.3, size = 3, alpha = 0.8) +
  scale_x_continuous(breaks = seq(min(click_counts$clicks),
                                  max(click_counts$clicks), by = 1)) +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Distribution of user clicks",
       x = "Clicks per user", y = "Number of users") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18))

## 5. Interaction type summary ----
unique(data$interaction_type)
interaction_map <- c("1" = "search", "2" = "recommendation")
data[, interaction_type_label := interaction_map[as.character(interaction_type)]]

interaction_summary <- data[, .(total_interactions = .N, CTR = mean(clicked)),
                            by = interaction_type_label]

ggplot(interaction_summary, aes(x = reorder(interaction_type_label, CTR),
                                y = CTR, fill = CTR)) +
  geom_col(width = 0.7, alpha = 0.9, show.legend = FALSE) +
  geom_text(aes(y = CTR + 0.037, label = sprintf("%.1f%%", CTR*100)),
            vjust = -0.5, size = 4) +
  geom_text(aes(y = CTR + 0.055, label = paste0("n=", total_interactions)),
            vjust = 1.5, size = 3) +
  scale_fill_gradient(low = "#a6bddb", high = "#4C72B0") +
  labs(title = "Click-through rate (CTR) by interaction type",
       x = "Interaction type", y = "CTR") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18),
        panel.grid.minor = element_blank()) +
  coord_flip()

## 6. User CTR distribution ----
user_ctr <- data[, .(interactions = .N, CTR = mean(clicked)), by = user_id]

hist_data <- hist(user_ctr$CTR, breaks = seq(0, 1, by = 0.1), plot = FALSE)
```

```r
hist_df <- data.frame(
  bin_start = hist_data$mids - diff(hist_data$breaks)[1]/2,
  bin_end = hist_data$mids + diff(hist_data$breaks)[1]/2,
  count = hist_data$counts,
  mid = hist_data$mids)

ggplot() +
  geom_col(data = hist_df, aes(x = mid, y = count),
           fill = "#4C72B0", color = "white", width = 0.1*0.95) +
  geom_text(data = hist_df, aes(x = mid, y = count,
                                 label = scales::comma(count)),
            vjust = -0.5, size = 3) +
  geom_vline(aes(xintercept = mean(user_ctr$CTR)),
             color = "red", linetype = "dashed", linewidth = 0.5) +
  scale_x_continuous(breaks = seq(0, 1, by = 0.1), limits = c(0, 1)) +
  scale_y_continuous(labels = scales::comma,
                     breaks = seq(100000, 600000, by = 100000)) +
  labs(title = "Distribution of user CTR",
       x = "CTR", y = "Number of users") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18))

## 7. Click and no-click fraction ----
noclick_rate <- data[, mean(clicked_item_id == 1)]

rates_df <- data.table(
  type = c("Click", "No-click"),
  rate = c(1 - noclick_rate, noclick_rate))

values <- rates_df$rate
labels <- rates_df$type
piepercent <- paste0(round(100 * values / sum(values), 1), "%")

# pie chart
pie(values, labels = piepercent,
    main = "Click vs no-click rate among interactions",
    col = c("#4C72B0", "#F8766D"))
legend("topright", labels, fill = c("#4C72B0", "#F8766D"))
```

```
## 8. Category analysis ----
# extract first word before comma
data[, category_main := sub(",.*", "", category)]

# count per category
category_counts <- data[category_main != "noClick", .N,
                        by = category_main][order(-N)]

ggplot(category_counts, aes(x = reorder(category_main, -N), y = N)) +
  geom_col(fill = "#4C72B0") +
  geom_text(aes(label = scales::comma(N)), vjust = -0.5, size = 3) +
  labs(title = "Number of interactions by main category",
       x = "Main category", y = "Number of interactions") +
  scale_y_continuous(labels = scales::comma,
                     breaks = seq(0, max(category_counts$N) + 2000000,
                                  by = 2000000)) +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text( size = 18),
        axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.minor = element_blank())

## 9. Top locations for REAL_ESTATE ----
# filter only REAL_ESTATE rows
real_estate_data <- data[category_main == "REAL_ESTATE"]

# split category column by "," directly into new columns
split_cols <- tstrsplit(real_estate_data$category, ",",
                        fixed = TRUE, fill = NA)

# trim whitespace
split_cols <- lapply(split_cols, str_trim)

# bind back into the data.table
real_estate_data[, paste0("category_part_",
                          seq_along(split_cols)) := split_cols]

# rename first three columns meaningfully
setnames(real_estate_data,
         old = c("category_part_1", "category_part_2", "category_part_3"),
         new = c("main_category", "subcategory", "location"))
```

```
location_counts <- real_estate_data[, .N, by = location][order(-N)]

top_locations <- head(location_counts, 10)
top_locations_df <- as.data.frame(top_locations) %>%
  rename(city = location, interactions = N) %>%
  mutate(query = paste(city, "Norway", sep = ", "))

# geocode locations
geo_df <- top_locations_df %>%
  geocode(address = query, method = "osm", lat = "lat", long = "lon",
          return_input = TRUE, limit = 1) %>%
  filter(!is.na(lat) & !is.na(lon))

# get Norway map from maps package
norway_map <- map_data("world", region = "Norway")

ggplot() +
  geom_polygon(data = norway_map, aes(x = long, y = lat, group = group),
               fill = "#f0f0f0", color = "grey60") +
  geom_point(data = geo_df, aes(x = lon, y = lat, size = interactions,
                                 color = interactions), alpha = 0.8) +
  geom_text_repel(data = geo_df, aes(x = lon, y = lat, label = city), size = 3) +
  scale_size_continuous(range = c(3, 10), labels = scales::label_comma(),
                        name = "Interactions (size)",
                        guide = guide_legend(reverse = TRUE)) +
  scale_color_distiller(palette = "RdYlBu", labels = scales::label_comma(),
                        direction = 1, name = "Interactions (color)",
                        guide = guide_colorbar(reverse = FALSE)) +
  labs(title = "Top REAL_ESTATE locations in Norway",
       x = "Longitude", y = "Latitude") +
  coord_cartesian(xlim = c(4, 14), ylim = c(58, 65)) +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18))

## 10. Slate-level analysis ----
# check how slate length affects CTR
slate_summary <- data[, .(impressions = .N, clicks = sum(clicked),
                          CTR = mean(clicked)),
                      by = slate_length][order(slate_length)]
```

```
ggplot(slate_summary, aes(slate_length, CTR)) +
  geom_line(color="#4C72B0") +
  geom_point(color="#4C72B0") +
  scale_x_continuous(
    breaks = seq(min(slate_summary$slate_length),
                 max(slate_summary$slate_length), by = 1)) +
  scale_y_continuous(breaks = seq(0.4, 1, by = 0.1), limits = c(0.35, 1)) +
  labs(title = "CTR vs slate length", x = "Slate length", y = "CTR") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18))

## 11. Click position analysis ----
# where do users tend to click in a slate?
click_pos_summary <- data[clicked == TRUE, .N,
                          by = clicked_pos][order(clicked_pos)]

click_pos_summary[, CTR := N / sum(N)]

ggplot(click_pos_summary, aes(clicked_pos, CTR)) +
  geom_line(color="#4C72B0") +
  geom_point(color="#4C72B0") +
  scale_x_continuous(
    breaks = seq(min(click_pos_summary$clicked_pos),
                 max(click_pos_summary$clicked_pos), by = 1)) +
  scale_y_continuous(breaks = seq(0, 0.2, by = 0.02), limits = c(0, 0.18)) +
  labs(title="Click probability by position",
       x="Position in slate", y="Click rate") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18))

# ---------- Modeling ----------
set.seed(991)

# base purchase probabilities
p_control <- 0.2
p_treat_values <- c(0.3, 0.45, 0.5)

# remove noClick rows (noClick item has index 1)
click_data <- setDT(data_without_slate_items) %>% filter(clicked_item_id != 1)
```

```r
# simulate X ~ Uniform(0,1)
click_data <- click_data %>% mutate(X = runif(n()))

## 1. Independent click model
independent_click_model <- function(dt, p_control, p_treat) {

  dt <- copy(dt)
  setDT(dt)

  # determine potential purchase per click
  dt[, potential_buy := X < fcase(is_treatment == 1, p_treat,
                                  is_treatment == 0, p_control)]

  # mark final buyer: only the click with the highest X
  # among potential buyers per item
  dt[, final_buy := 0]
  dt[potential_buy == TRUE,
     final_buy := fifelse(X == max(X), 1, 0), by = .(clicked_item_id)]
}

## 2. Repeated clicks model
repeated_clicks_model <- function(dt, p_control, p_treat) {

  dt <- copy(dt)
  setDT(dt)

  # compute click index per user-item
  setorder(dt, user_id, clicked_item_id, interaction_id)
  dt[, k := seq_len(.N), by = .(user_id, clicked_item_id)]

  # diminishing-returns probability per click
  dt[, p_buy := fcase(is_treatment == 1, 1 - (1 - p_treat)^k,
                      is_treatment == 0, 1 - (1 - p_control)^k)]

  # potential buys
  dt[, potential_buy := X < p_buy]

  # mark final buyer: only the click with the highest X
  # among potential buyers per item
  dt[, final_buy := 0]
```

```
  dt[potential_buy == TRUE,
     final_buy := fifelse(X == max(X), 1, 0), by = .(clicked_item_id)]
}

# ---------- Bernoulli randomization ----------
click_data_bernoulli <- copy(click_data)

# randomize users with Bernoulli(0.5)
user_assignments <- click_data %>%
  distinct(user_id) %>%
  mutate(is_treatment = rbinom(n(), 1, 0.5))

# attach user assignment to each row
click_data_bernoulli <- click_data %>%
  left_join(user_assignments, by = "user_id")

## No interference (true ATE) ----
no_interference_model <- function(dt, p_control, p_treat) {
  setDT(dt)

  # purchase per click
  dt[, final_buy := as.integer(
    X < fcase(is_treatment == 1, p_treat,
              is_treatment == 0, p_control))]
}

# keep only the first click per user-item pair
click_data_bernoulli_0 <- click_data_bernoulli[!duplicated(
  click_data_bernoulli, by = c("user_id", "clicked_item_id"))]

compute_true_ATE <- function(dt, p_control, p_treat) {
  setDT(dt)

  # all treated
  dt_t <- copy(dt)[, is_treatment := 1]
  dt_t <- no_interference_model(dt_t, p_control, p_treat)

  # all control
  dt_c <- copy(dt)[, is_treatment := 0]
  dt_c <- no_interference_model(dt_c, p_control, p_treat)
```

```r
  # outcomes
  y1 <- dt_t$final_buy
  y0 <- dt_c$final_buy

  # ATE
  true_ATE <- mean(y1) - mean(y0)

  return(round(true_ATE, 3))
}

# get true ATE
true_ATE_pTreat_0.3 <- compute_true_ATE(click_data_bernoulli_0,
                                       p_control, p_treat_values[1])

true_ATE_pTreat_0.45 <- compute_true_ATE(click_data_bernoulli_0,
                                        p_control, p_treat_values[2])

true_ATE_pTreat_0.5 <- compute_true_ATE(click_data_bernoulli_0,
                                       p_control, p_treat_values[3])

## Model comparison ----
# apply Bernoulli randomization with p_treat = 0.3

# Model 0: no interference (first click per user-item)
bernoulli_icm_0 <- independent_click_model(
  click_data_bernoulli_0, p_control, p_treat_values[1])

# Model 1: independent clicks with interference across users
bernoulli_icm <- independent_click_model(
  click_data_bernoulli, p_control, p_treat_values[1])

# Model 2: repeated clicks with diminishing returns
bernoulli_rcm <- repeated_clicks_model(
  click_data_bernoulli, p_control, p_treat_values[1])

# save fitted model outputs
saveRDS(bernoulli_icm_0, file = file.path(model_dir, "bernoulli_icm_0.rds"))
saveRDS(bernoulli_icm, file = file.path(model_dir, "bernoulli_icm.rds"))
saveRDS(bernoulli_rcm, file = file.path(model_dir, "bernoulli_rcm.rds"))
```

```r
# load saved model outputs
bernoulli_icm_0 <- readRDS(file.path(model_dir, "bernoulli_icm_0.rds"))
bernoulli_icm <- readRDS(file.path(model_dir, "bernoulli_icm.rds"))
bernoulli_rcm <- readRDS(file.path(model_dir, "bernoulli_rcm.rds"))

# combine all models
all_models_dt <- rbind(
  bernoulli_icm_0[, .(user_id, potential_buy, final_buy, is_treatment)
  ][, model := "Model 0"],
  bernoulli_icm[, .(user_id, potential_buy, final_buy, is_treatment)
  ][, model := "Model 1"],
  bernoulli_rcm[, .(user_id, potential_buy, final_buy, is_treatment)
  ][, model := "Model 2"])

# aggregate outcomes to the user level
user_level_dt <- all_models_dt[, .(
  final_buy = as.integer(any(final_buy == 1)),
  potential_buy = as.integer(sum(potential_buy == TRUE)),
  by = .(user_id, is_treatment, model)]

# 1. Are results driven by a few heavy users?
user_level_dt[, model := factor(
  model,levels = c("Model 0", "Model 1", "Model 2"))]

# check the skew
summary(user_level_dt$potential_buy)
quantile(user_level_dt$potential_buy, probs = c(0.5, 0.9, 0.99, 1))

# log-transform to handle heavy tails and zeros (log1p(x) = log(1+x))
ggplot(user_level_dt, aes(x = log1p(potential_buy),
                          fill = factor(is_treatment))) +
  geom_histogram(bins = 30, alpha = 0.6, position = "identity") +
  facet_grid(factor(is_treatment,
                    labels = c("Control", "Treatment")) ~ model) +
  labs(title = "Distribution of potential purchase opportunities per user",
       x = "Log(1 + potential purchases per user)",
       y = "User count", fill = "Treatment") +
  scale_fill_manual(values = c("0" = "#4C72B0", "1" = "#F8766D")) +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal(base_size = 16) +
```

```r
  theme(plot.title = element_text(size = 18),
        strip.text = element_text(size = 16), legend.position = "none")

# 2. Potential buy intensity:
# how many buying opportunities does this model create per user?
potential_intensity <- user_level_dt[
  , .(mean_potential = mean(potential_buy)),
  by = .(model, is_treatment)]

# which model is more aggressive regardless of outcome?
ggplot(potential_intensity,
       aes(x = model, y = mean_potential, color = factor(is_treatment),
           group = factor(is_treatment))) +
  geom_point(size = 3) + geom_line(linetype = "dashed") +
  labs(title = "Average number of purchase opportunities per user",
       x = "Model", y = "Average potential purchases", color = "Treatment") +
  scale_color_manual(values = c("#4C72B0", "#F8766D"),
                     labels = c("Control", "Treatment")) +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18), legend.position = "bottom")

# 3. Efficiency metric:
# how many potential buys does it take to get one real buy?
efficiency <- user_level_dt[
  , .(final_rate = mean(final_buy),
      potential_rate = mean(potential_buy)),
  by = .(model, is_treatment)
][, efficiency := final_rate / pmax(potential_rate, 1e-6)]

# which model converts potential buys into real ones most efficiently?
ggplot(efficiency[is_treatment == 0],
       aes(x = potential_rate, y = final_rate, label = model)) +
  geom_point(size = 3, color = "#4C72B0") +
  geom_text(nudge_y = 0.002, size = 4) +
  labs(title = "Efficiency frontier (control)",
       x = "Average purchase opportunities per user",
       y = "Probability of at least one purchase") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18))
```

```
ggplot(efficiency[is_treatment == 1],
       aes(x = potential_rate, y = final_rate, label = model)) +
  geom_point(size = 3, color = "#F8766D") +
  geom_text(nudge_y = 0.002, size = 4) +
  labs(title = "Efficiency frontier (treatment)",
       x = "Average purchase opportunities per user",
       y = "Probability of at least one purchase") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18))

# 4. final-buy rates by model and treatment
summary_user <- user_level_dt[
  , .(final_rate = mean(final_buy),
      potential_rate = mean(potential_buy)),
  by = .(model, is_treatment)]

# which model truly moves final outcomes under treatment?
ggplot(summary_user,
       aes(x = model, y = final_rate, color = factor(is_treatment),
           group = factor(is_treatment))) +
  geom_point(size = 3) + geom_line(linetype = "dashed") +
  labs(title = "Final buy rate by model and treatment",
       x = "Model", y = "User purchase probability", color = "Treatment") +
  scale_color_manual(values = c("#4C72B0", "#F8766D"),
                     labels = c("Control", "Treatment")) +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18), legend.position = "bottom")

# 5. Value per unit of aggressiveness
ate_dt <- summary_user[
  , .(ATE = final_rate[is_treatment == 1] - final_rate[is_treatment == 0],
      potential_uplift = potential_rate[is_treatment == 1] -
        potential_rate[is_treatment == 0]),
  by = model]

ate_dt[, model := factor(model, levels = c("Model 0", "Model 1", "Model 2"))]

# which model gives the most lift per unit of aggressiveness?
ate_dt[, value_per_potential := ATE / pmax(potential_uplift, 1e-6)]
```

```r
ggplot(ate_dt, aes(x = model, y = value_per_potential, group = 1)) +
  geom_point(size = 3, color = "#4C72B0") +
  geom_line(linetype = "dashed", color = "#4C72B0") +
  labs(title = "Treatment effect per additional potential purchase",
       x = "Model", y = "ATE per potential buy") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18), legend.position = "none")

# 6. Causal impact vs aggressiveness trade-off
ate_ci_dt <- user_level_dt[, {
  y1 <- final_buy[is_treatment == 1]
  y0 <- final_buy[is_treatment == 0]
  ate <- mean(y1) - mean(y0)
  se <- sqrt(var(y1) / length(y1) + var(y0) / length(y0))
  .(ATE    = ate, ATE_lo = ate - 1.96 * se, ATE_hi = ate + 1.96 * se)
  }, by = model]

plot_dt <- merge(ate_ci_dt, ate_dt[, .(model, potential_uplift)], by = "model")

# which model causally increases user conversion the most
# and at what cost in aggressiveness?
ggplot(plot_dt, aes(x = potential_uplift, y = ATE, label = model)) +
  geom_point(size = 3, color = "#4C72B0") +
  geom_errorbar(aes(ymin = ATE_lo, ymax = ATE_hi),
                width = 0.015, alpha = 0.8, color = "#4C72B0") +
  geom_text(aes(label = model),
            nudge_y = c(0.003, 0.003, -0.003), size = 4) +
  labs(title = "Treatment impact versus exposure intensity",
       x = "Increase in purchase opportunities per user (treatment - control)",
       y = "ATE on purchase probability (95% CI)") +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18), legend.position = "none")

# 7. ATE on final_buy: which model causes more users to buy?
ggplot(ate_ci_dt, aes(x = model, y = ATE, group = 1)) +
  geom_point(size = 3, color = "#F8766D") +
  geom_errorbar(
    aes(ymin = ATE_lo, ymax = ATE_hi),
    width = 0.12, color = "#F8766D", alpha = 0.8) +
```

```
  labs(title = "Average treatment effect by model",
       x = "Model",
       y = "ATE on purchase probability (95% CI)") +
  coord_flip() +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18),
        plot.margin = margin(t = 5.5, r = 20, b = 5.5, l = 5.5))

## Run Bernoulli randomization ----
p_treat <- p_treat_values[1]

bernoulli_icm <- independent_click_model(
  dt = click_data_bernoulli,
  p_control = p_control,
  p_treat = p_treat)

bernoulli_rcm <- repeated_clicks_model(
  dt = click_data_bernoulli,
  p_control = p_control,
  p_treat = p_treat)

# save results
saveRDS(bernoulli_icm, file = file.path(
  model_dir, sprintf("bernoulli_icm_pTreat_%0.2f.rds", p_treat)))

saveRDS(bernoulli_rcm, file = file.path(
  model_dir, sprintf("bernoulli_rcm_pTreat_%0.2f.rds", p_treat)))

# ---------- Cluster randomization ----------
# user-slate clustering
click_data_cluster <- copy(click_data)

# group by interaction sequence per user
click_data_cluster[, cluster_id := .GRP, by = .(user_id, interaction_id)]

clusters <- unique(click_data_cluster$cluster_id)

treated_clusters <- sample(clusters, size = floor(length(clusters) * 0.5),
                           replace = FALSE)
```

```
click_data_cluster[, is_treatment :=
                     as.integer(cluster_id %in% treated_clusters)]

# run cluster randomization
p_treat <- p_treat_values[1]

cluster_icm <- independent_click_model(
  dt = click_data_cluster,
  p_control = p_control,
  p_treat = p_treat)

cluster_rcm <- repeated_clicks_model(
  dt = click_data_cluster,
  p_control = p_control,
  p_treat = p_treat)

# save resuts
saveRDS(cluster_icm, file = file.path(
  model_dir, sprintf("cluster_icm_pTreat_%0.2f.rds", p_treat)))

saveRDS(cluster_rcm, file = file.path(
  model_dir, sprintf("cluster_rcm_pTreat_%0.2f.rds", p_treat)))

# ---------- ATE estimation ----------
## Load models ----
load_models <- function(model_name, model_dir) {
  files <- list.files(model_dir,
                      pattern = paste0("^", model_name, "_pTreat_.*\\.rds$"),
                      full.names = TRUE)

  models <- lapply(files, readRDS)
  names(models) <- sub(".*pTreat_", "pTreat_", files)
  names(models) <- sub("\\.rds$", "", names(models))

  return(models)
}

# independent click models
bernoulli_icm <- load_models("bernoulli_icm", model_dir)
cluster_icm <- load_models("cluster_icm", model_dir)
```

```
# repeated clicks models
bernoulli_rcm <- load_models("bernoulli_rcm", model_dir)
cluster_rcm <- load_models("cluster_rcm", model_dir)

## Estimate ATE ----
aggregate_user_level <- function(dt) {
  dt[, .(final_buy = as.integer(any(final_buy == 1))),
     by = .(user_id, is_treatment)]
}

estimate_user_level <- function(final_buy, is_treatment, ATE_true) {

  y1 <- final_buy[is_treatment == 1]
  y0 <- final_buy[is_treatment == 0]

  ate <- mean(y1) - mean(y0)

  se <- sqrt(var(y1) / length(y1) + var(y0) / length(y0))

  list(ATE = ate,
       Bias = ate - ATE_true,
       CI_lower = ate - 1.96 * se,
       CI_upper = ate + 1.96 * se)
}

ATE_for_pTreat <- function(p_treat, ATE_true,
                           model_bernoulli, model_cluster) {

  p_name <- sprintf("pTreat_%.2f", p_treat)

  # Bernoulli randomized
  bernoulli_ul <- aggregate_user_level(model_bernoulli[[p_name]])

  res_bernoulli <- estimate_user_level(
    bernoulli_ul$final_buy, bernoulli_ul$is_treatment, ATE_true)
```

```
  # cluster randomized
  cluster_ul <- aggregate_user_level(model_cluster[[p_name]])

  res_cluster <- estimate_user_level(
    cluster_ul$final_buy, cluster_ul$is_treatment, ATE_true)

  # combine
  data.table(
    pTreat = p_treat,
    Method = factor(
      c("True", "Bernoulli randomized", "Cluster randomized"),
      levels = c("True", "Bernoulli randomized", "Cluster randomized")),
    ATE = c(ATE_true, res_bernoulli$ATE, res_cluster$ATE),
    Bias = c(0, res_bernoulli$Bias, res_cluster$Bias),
    CI_lower = c(NA, res_bernoulli$CI_lower, res_cluster$CI_lower),
    CI_upper = c(NA, res_bernoulli$CI_upper, res_cluster$CI_upper))
}

# independent click models
ATE_0.3_icm <- ATE_for_pTreat(p_treat_values[1], true_ATE_pTreat_0.3,
                              bernoulli_icm, cluster_icm)

ATE_0.45_icm <- ATE_for_pTreat(p_treat_values[2], true_ATE_pTreat_0.45,
                               bernoulli_icm, cluster_icm)

ATE_0.5_icm <- ATE_for_pTreat(p_treat_values[3], true_ATE_pTreat_0.5,
                              bernoulli_icm, cluster_icm)

# repeated clicks models
ATE_0.3_rcm <- ATE_for_pTreat(p_treat_values[1], true_ATE_pTreat_0.3,
                              bernoulli_rcm, cluster_rcm)

ATE_0.45_rcm <- ATE_for_pTreat(p_treat_values[2], true_ATE_pTreat_0.45,
                               bernoulli_rcm, cluster_rcm)

ATE_0.5_rcm <- ATE_for_pTreat(p_treat_values[3], true_ATE_pTreat_0.5,
                              bernoulli_rcm, cluster_rcm)
```

```
plot_ATE <- function(ATE) {

  # filter out the true values
  plot_data <- ATE %>% filter(Method != "True")

  # set true ATE for dashed line
  true_ATE <- ATE$ATE[ATE$Method == "True"]

  ggplot(plot_data, aes(y = Method, x = ATE, color = factor(pTreat))) +
    geom_errorbarh(aes(xmin = CI_lower, xmax = CI_upper), height = 0.2) +
    geom_point(size = 3) +
    geom_vline(xintercept = true_ATE, linetype = "dashed",
               color = "#4C72B0", linewidth = 1) +
    labs(x = "Treatment effect estimate", y = "Method") +
    theme_minimal(base_size = 16) +
    theme(plot.title = element_text(size = 18),
          legend.position = "none")
}

plot_ATE(ATE_0.3_icm)
plot_ATE(ATE_0.45_icm)
plot_ATE(ATE_0.5_icm)

plot_ATE(ATE_0.3_rcm)
plot_ATE(ATE_0.45_rcm)
plot_ATE(ATE_0.5_rcm)

# ---------- Exposure mapping ----------
p_treat <- p_treat_values[1]

dt0 <- copy(click_data_cluster)
setDT(dt0)

# fraction of treated clicks per item
dt0[, exposure_item := mean(is_treatment), by = clicked_item_id]

# leave-one-out exposure
dt0[, exposure_item_loo := if (.N > 1)
  (sum(is_treatment) - is_treatment) / (.N - 1)
  else 0, by = clicked_item_id]
```

```
# run both models
dt_icm <- independent_click_model(copy(dt0), p_control, p_treat)
dt_icm[, model := "ICM"]

dt_rcm <- repeated_clicks_model(copy(dt0), p_control, p_treat)
dt_rcm[, model := "RCM"]

# combine
dt_both <- rbind(dt_icm, dt_rcm, fill = TRUE)

# aggregate to user level
user_dt <- dt_both[
  , .(final_buy = as.integer(any(final_buy == 1)),
      exposure  = mean(exposure_item_loo),
      k_user    = sum(k, na.rm = TRUE)),   # <- add intensity
  by = .(user_id, is_treatment, model)]

# Exposure regressions ----
# ICM
fit_icm <- lm(
  final_buy ~ is_treatment * exposure,
  data = user_dt[model == "ICM"])

# RCM: adjusted for k_user
fit_rcm2 <- lm(
  final_buy ~ is_treatment * exposure + k_user,
  data = user_dt[model == "RCM"])

summary(fit_icm)
summary(fit_rcm2)

# Exposure curves ----
# ICM curve
exp_icm <- user_dt[model == "ICM"] %>%
  mutate(exposure_bin =
           cut(exposure, breaks = seq(0, 1, by = 0.2),
               include.lowest = TRUE)) %>%
  group_by(is_treatment, exposure_bin) %>%
  summarise(mean_buy = mean(final_buy), n = n(), .groups = "drop")
```

```
p_icm <- ggplot(exp_icm,
                aes(x = exposure_bin, y = mean_buy,
                    color = factor(is_treatment),
                    group = is_treatment)) +
  geom_point(size = 3) +
  geom_line() +
  labs(title = "Exposure-mapped response curve (independent click model)",
       x = "Fraction of treated competitors",
       y = "Probability of purchase",
       color = "Treatment") +
  scale_color_manual(values = c("#4C72B0", "#F8766D"),
                     labels = c("Control", "Treatment")) +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18),
        legend.position = "bottom")

# RCM curve
exp_rcm2 <- user_dt[model == "RCM"] %>%
  mutate(exposure_bin =
           cut(exposure, breaks = seq(0, 1, by = 0.2),
               include.lowest = TRUE)) %>%
  group_by(is_treatment, exposure_bin) %>%
  summarise(mean_buy = mean(final_buy), n = n(), .groups = "drop")

p_rcm2 <- ggplot(exp_rcm2,
                 aes(x = exposure_bin, y = mean_buy,
                     color = factor(is_treatment),
                     group = is_treatment)) +
  geom_point(size = 3) +
  geom_line() +
  labs(title = "Exposure-mapped response curve (repeated clicks model)",
       x = "Fraction of treated competitors",
       y = "Probability of purchase",
       color = "Treatment") +
  scale_color_manual(values = c("#4C72B0", "#F8766D"),
                     labels = c("Control", "Treatment")) +
  theme_minimal(base_size = 16) +
  theme(plot.title = element_text(size = 18),
        legend.position = "bottom")
```

```r
p_icm
p_rcm2

# Direct and total ATEs ----
get_ate_from_fit <- function(fit, user_sub) {
  direct <- coef(fit)["is_treatment"]

  mean_exp <- mean(user_sub$exposure)

  total <- coef(fit)["is_treatment"] +
    coef(fit)["is_treatment:exposure"] * mean_exp

  c(direct_ATE = direct, total_ATE = total)
}

ate_icm  <- get_ate_from_fit(fit_icm,  user_dt[model == "ICM"])
ate_rcm2 <- get_ate_from_fit(fit_rcm2, user_dt[model == "RCM"])

ate_icm
ate_rcm2

# Cluster-robust SEs ----
vc_icm <- vcovCL(
  fit_icm,
  cluster = dt_icm$clicked_item_id[
    match(user_dt[model == "ICM"]$user_id, dt_icm$user_id)])

vc_rcm2 <- vcovCL(
  fit_rcm2,
  cluster = dt_rcm$clicked_item_id[
    match(user_dt[model == "RCM"]$user_id, dt_rcm$user_id)])

coeftest(fit_icm,  vcov = vc_icm)
coeftest(fit_rcm2, vcov = vc_rcm2)
```