A Structural Model of Rewards Programs on Digital Platforms: The Case of Livestreaming

Sueyoul Kim*

Last Update: February 15, 2026

Abstract

Many digital platforms, including Uber, eBay, and YouTube, use rewards programs to motivate sellers. This paper presents an empirical framework for assessing the impact of such rewards programs on platform revenue. The context is a Korean livestreaming platform, where sellers called streamers broadcast content, earn revenue from viewer tips, and qualify for commission discount rewards through performance-based tournaments. I collect microdata on effort and revenue from streamer tracking websites and estimate a dynamic game model to capture the effect of rewards program design on streamers' behavior. Counterfactual simulations show that commission discount rewards motivate streamers—especially more profitable ones—to stream more and increase total revenue, but make the platform take a substantially smaller share of the generated revenue. Because the latter effect quantitatively dominates, I find that the current program is not optimal, and reducing the discount rate can improve platform revenue.

*Korea Development Institute. sueyoul.econ@gmail.com. I am deeply indebted to Ginger Jin and Andrew Sweeting for their constant guide and support. I thank Mario Leccese, Yongjoon Park, Michael Sullivan, Daniel Vincent and Chenyu Yang for helpful comments. Shanglyu Deng, Keaton Ellis, Rachel Nesbit, Ece Yegane, and participants in the Mannheim Centre for Competition and Innovation (MaCCI) 2022 Summer School on Platform Economics kindly provided valuable discussions and encouragement. All remaining errors are my own.

1 Introduction

To motivate their sellers, numerous digital platforms operate performance-based rewards programs. For example, Uber drivers can earn cash rewards and gas subsidies by accumulating points and reaching the Diamond tier within the Uber Pro program.[1] The eBay Top Rated Seller Program offers selling fee discounts and enhanced visibility for sellers based on their sales records and on-time shipment.[2]

What factors should a platform consider when designing such rewards programs? I study this question in the context of a Korean livestreaming platform. In this industry, sellers (called streamers) broadcast content such as video games and generate revenue by receiving tips from viewers.[3] Platform revenue is determined by three factors. First, total watch time, which is the amount of time viewers spend watching streamers. Second, average profitability, i.e., how much tipping revenue streamers extract from a given amount of watch time. Third, the platform's share of the collected total revenue.

The reward program in my context is a commission discount for highly performing streamers. This reward creates a trade-off among the three aforementioned factors. Offering the reward could raise platform revenue through two channels. First, streamers work more on average, leading to an increase in total watch time. Second, more profitable streamers, who earn higher tipping revenue per unit of watch time, could be disproportionately motivated by the commission discount reward. If such streamers occupy a larger portion of watch time, this increases the average tipping revenue per unit of watch time. However, the platform receives a smaller share of the generated revenue due to the commission discounts.

To empirically study this trade-off, I collect microdata from a leading livestreaming platform in South Korea. The environment is suitable for my study for two reasons. First, individual streamer-time data on effort and outcome measures are available, such as streaming hours, viewership, tipping revenue, and rewards program acceptance. Second, the benefits and requirements of the rewards program are straightforward. The key benefit is a commission discount, and acceptance is determined by monthly tournaments based on observable streamer performance metrics.

To assess how different reward program designs affect heterogeneous streamers'

[1]https://www.uber.com/us/en/drive/uber-pro/, accessed on May 23, 2023

[2]https://www.ebay.com/sellercenter/protections/top-rated-program, accessed on May 23, 2023

[3]This paper does not incorporate ad revenue due to the lack of data. In my empirical context, financial reports from the focal platform show that about 80% of its revenue comes from tips.

effort choices and competition, I estimate a dynamic game model of streamers. In the model, streamers choose their streaming hours each period, taking into account their current revenue and the potential to gain popularity and receive rewards. The model primitives are the effort cost of streaming and the distribution of outside option values.

Why use a dynamic game model? The model should be dynamic to account for the intertemporal trade-offs faced by streamers. They may increase their streaming hours today to enhance their chances of receiving the commission discount and starting with a larger fan base in the future. In addition, the game among streamers captures two relevant interactions. First, streamers with similar content compete for viewership, which means that a streamer's watch time may decrease if other streamers increase their streaming hours. Second, the reward program is determined through a performance-based tournament. Therefore, the likelihood that a streamer is accepted decreases when the number of other streamers who perform better increases.[4]

In my model, streamers primarily differ along two dimensions: their profitability, measured by tipping revenue per unit of watch time, and the category of content that they broadcast. These differences reflect a streamer's personal characteristics, such as their entertainment skills and talent for creating specific types of content. Both dimensions affect platform revenue in the model. Whether more profitable streamers capture a larger share of watch time affects total tipping revenue, and the market-stealing effect from within-category competition influences total watch time.

To estimate the dynamic game model, I solve the model for each parameter guess and search for a set of parameters that maximize the pseudo log-likelihood of the model predictions. Two computational difficulties arise at this step.

First, the curse of dimensionality: streamers in the model should account for all possible state and action profiles of their competitors, but there are thousands of streamers, which makes the state space excessively large. I address this challenge by imposing assumptions in the style of an oblivious equilibrium approach (Weintraub et al., 2008). Streamers are assumed to believe that the influence of competing streamers is captured by a summary statistic: the weighted sum of the streaming hours of other streamers broadcasting the same type of content.

Second, the typical logit-based approach for approximating watch time share creates unrealistic incentives for streamers. In logit models, watch time share increases

[4] The score is computed based on three observable factors: viewership, fan base, and cumulative streaming time. How the platform converts these factors to a score is publicly available information and is summarized in Table A.8. Detailed description on the tournament is in section 2.2.

exponentially with a streamer's streaming hours. As a result, marginal revenue from increasing streaming hours becomes convex and grows rapidly at higher effort levels. When combined with standard effort costs, this leads to corner solutions in which streamers optimally choose either zero or the maximum feasible streaming hours.

To address this issue, I instead approximate watch time as a flexible function of streamers' streaming hour choices using a tree-based model (XGBoost). This nonparametric approximation captures diminishing returns to streaming hours and yields interior optimal effort choices. It also provides better out-of-sample predictive performance for watch time.

Using the estimated model, I conduct three counterfactual simulations to evaluate the effects of different reward program design on platform revenue. First, I adjust the number of streamers who receive the reward in each period. Second, I change the magnitude of the commission discount provided as a reward. Third, I study a more granular program design, in which approval slots are allocated at the broadcasting category level instead of at the platform level.

Overall, the counterfactual results show that reducing the reward can improve platform revenue. As expected, offering commission discounts makes streamers stream more, which increases total watch time (*watch time effect*). Additionally, it disproportionately motivates more profitable streamers, who can extract more tips per unit of watch time, thereby improving profitability measured by revenue per watch time (*profitability effect*).

However, the revenue improvement from these factors is not sufficient to offset the decrease in the platform's share of generated revenue due to the commission discounts (*platform share effect*). This situation is analogous to a scenario in which a firm's revenue does not increase despite offering price discounts due to inelastic demand.

In the first counterfactual simulation, I double and halve the number of streamers receiving the reward each period. When the number is doubled, both the watch time effect and profitability effect increase total revenue by 0.22% and 0.94%, respectively. However, the platform's share of total revenue decreases by 3.66%, resulting in a 2.53% decrease in platform revenue when all factors are taken into account. On the other hand, halving the number of streamers receiving the reward leads to a 3.03% increase in platform revenue compared to the original platform revenue.

Similar results emerge when modifying another design factor: the amount of commission discount. When the discount benefit increases to 15% (from the current 10%), the effects on watch time and profitability raise total revenue by 0.07% and

0.34%, respectively. However, due to a 6.63% decrease in the platform's share, the platform's revenue decreases by 6.24% overall. Conversely, reducing the commission discount benefit to 5% raises platform revenue by 6.08%.

I also conduct a third counterfactual simulation in which the number of streamers who can receive the reward is allocated at the category level (Game, Social, or Other), instead of at the platform level. The purpose of this exercise is twofold. First, it reveals whether and to what extent the platform can gain from adopting a more granular program design. Second, while the results above imply that reducing or even removing program benefits may improve platform revenue, such changes may be difficult to implement in practice.[5] Category-level reallocation could be an alternative way to increase platform revenue without unilaterally reducing overall streamer benefits.

Specifically, I increase the number of streamers receiving the discount reward in the Game category by two, while simultaneously reducing this number by two in the Social category. The key observation for this analysis is that since the current program offers rewards without differentiation among streamers in different categories, the marginal revenue improvement from providing a reward to each category is not equalized across categories. Therefore, giving more rewards to the more responsive category should increase platform revenue. I found that this design change improved platform revenue by 1.19%.

The main takeaway from the counterfactual simulations is that providing rewards that reduce a firm's revenue must be done carefully, taking into account sellers' responsiveness to the reward. As demonstrated in this study, such benefits can accumulate and significantly reduce the firm's revenue in the long-run equilibrium.

While this paper focuses on a specific institutional context, this takeaway is relevant for a wide range of platforms and industries. Because livestreaming platforms have similar business models across countries, the model and findings are applicable to other livestreaming platforms such as Twitch (US) and Huya (China).

More broadly, this takeaway applies to platforms that rely on commission-based revenue and use performance-based rewards to motivate sellers, including gig economy platforms (e.g., Uber, DoorDash) and online marketplaces (e.g., eBay, Etsy). In these settings, increasing seller rewards may stimulate effort and output but can also

[5] The streamers I interviewed all expressed very negative reactions to the idea of removing existing benefits. Similarly, Bewley (1998) found that, in practice, pay cuts are rarely implemented because managers believe they significantly harm employee morale, despite their justification in economic models with rational agents.

reduce platform revenue if the increase in seller activity does not sufficiently offset the reduction in the platform's commission share.

Related Literature This paper contributes to the literature on incentive schemes in a firm. Two strands of literature connect to this research question. First, from the perspective of personnel economics, the rewards programs can be considered as performance-based job promotions on a digital platform.[6] Such promotion incentives have been studied by Eriksson (1999), DeVaro and Morita (2013), and Belzil and Bognanno (2008), among many others. Second, salesforce management literature in marketing investigates how compensation plans, like bonuses and quotas, shape worker efforts. Examples include Misra and Nair (2011), Chung, Steenburgh, and Sudhir (2014), Daljord, Misra, and Nair (2016), Chung, Kim, and Park (2021).

The primary contribution of this paper is to develop a structural model that incorporates worker competition and relative performance-based compensation. In this paper's model, workers compete with each other. As a result, one worker's effort may negatively the affect other workers' outcomes. Furthermore, the model indirectly captures a feature wherein a worker has fewer chances to receive rewards when other workers perform better.

This paper is also related to the dynamic contests/tournaments design literature. A number of papers study this topic, particularly regarding how a designer should provide information for participants. Examples include Lemus and Marshall (2021), Bimpikis, Ehsani, and Mostagir (2019) and Mostagir, Chen, and Yeckehzaare (2019). While such literature mainly studies dynamics *within* a single tournament, I focus on a dynamic tournament that repeats over time by using a steady state equilibrium framework. This approach is suitable to capture dynamics *across* tournaments. For example, a participant of an online prediction tournament (e.g. Kaggle) may improve their coding skills and perform better in future tournaments.

Additionally, this paper contributes to a small but growing body of literature on the livestreaming industry. This new industry has provided a good environment for studying topics like congestion externality and influencer marketing. Some examples are Tudón (2021), Simonov, Ursu, and Zheng (2021), Huang and Morozov (2022), and Lu et al. (2021). In this strand of literature, this paper has two marginal contributions. First, it places its focus on the understudied role of streamer rewards programs. Second, it directly incorporates micro-level revenue information, enabling a more

[6]See Table 1 in Lazear (2018) for the taxonomy.

accurate capture of the marginal benefit of streamers' efforts.[7]

Lastly, from a methodological perspective, this paper relies on the literature of dynamic games in Industrial Organization, building upon the seminal work by Ericson and Pakes (1995). Solving a dynamic game poses a computational challenge called the curse of dimensionality.[8] I follow an idea developed in the oblivious equilibrium literature to address this challenge.[9] I additionally incorporate tournament between players within this framework to show its empirical usefulness for evaluating the rewards program designs.

Roadmap Section 2 describes the general industry background, my specific empirical set up, and data sources. Section 3 presents my model. Section 4 describes the estimation procedure and model parameter estimates. Section 5 presents counterfactual simulations, and Section 6 concludes.

2 Empirical Context & Data

2.1 General Industry Background

A livestreaming platform is an online space where sellers, called streamers, broadcast various types of content, such as singing or playing video games. Each streamer functions like a television channel, as shown in Figure 1.

The livestreaming industry has become a flourishing business over the last decade, both in terms of time and money spent. Twitch, a dominant firm in the U.S. market, had 31 million average daily visitors and 8 million monthly active streamers in 2021.[10] This popularity has made streaming a lucrative venture for some. For example, a top streamer on Twitch is estimated to earn about $2 million per year from streaming

[7]Obtaining such micro-level revenue information is often challenging due to the involvement of third parties. For instance, on platforms like Twitch, many streamers receive tips through third-party applications such as Twip and Toonation. As a result, even the platform itself does not have precise information about its streamers' revenue. In my empirical context, advertising such third-party applications is not allowed.

[8]For example, when there are 100 players and 10 possible states, a rational player has to account for 10^{100} possible configurations to develop a contingency plan.

[9]The key simplification is that each player solves a single-agent dynamic programming, believing that other players' states and actions are unaffected by the player's actions. This methodology describes markets well when market shares are not too highly concentrated (Weintraub, Benkard, and Van Roy (2008)). In my empirical context, even the top 1% of streamer-month observations take up 2.3% of the monthly market share as measured by watch time.

[10]https://twitchadvertising.tv/audience/, accessed on May 1, 2022.

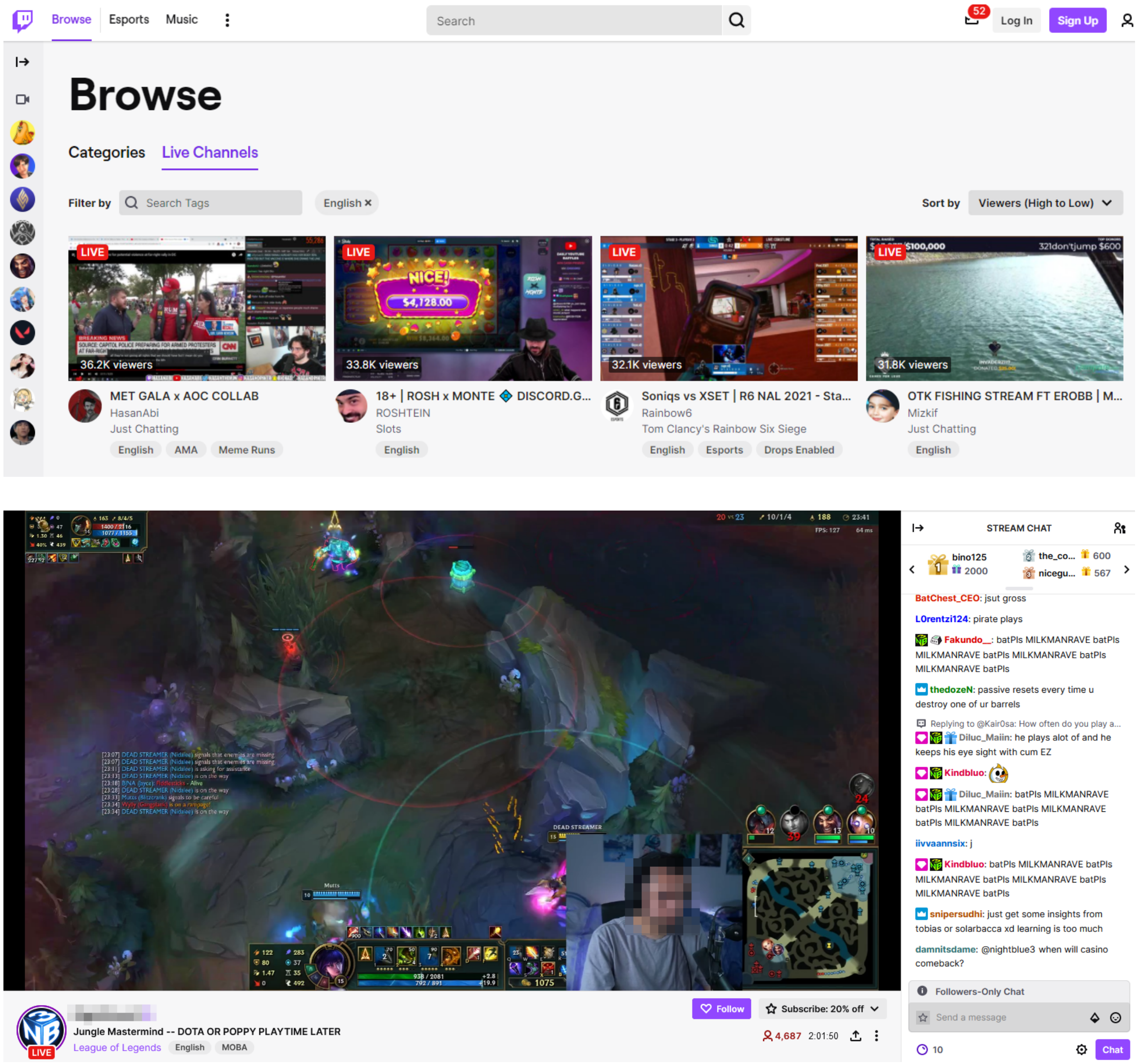


Figure 1. Twitch Homepage (Top) and Streamer Channel (Bottom)

Notes: Twitch is the dominant live-streaming platform in the U.S. Viewers interact with content through individual streamer channels, which feature a video player and an integrated chat room. The interface shown here is characteristic of live-streaming platforms worldwide.

on the platform.[11] The majority of users are young, making the industry appear promising for further growth. For example, as of 2022, 73% of Twitch viewers were under the age of 35.[12]

Livestreaming platforms are not only popular in the United States but also in other countries. In China, the leading platforms include Douyu, Huya, Douyin, and Bilibili, followed by Zhanqi and LiveQQ, all with massive user bases. Douyu alone had 163.6 million monthly active users in 2019.[13] In Japan, YouTube Live and Twitch are popular, competing with local rivals such as Niconico. In South Korea, during the data period of this paper, the top two platforms were AfreecaTV and Twitch, while YouTube Live also holds a significant market share.[14]

Business Model Streamers and livestreaming platforms primarily generate revenue from viewer tipping. Viewers donate money to their favorite streamers, similar to tipping street performers. The platform earns revenue by charging a commission on tipping income.

In general, viewers can watch streamers' content without paying, but many appear to derive utility from tipping because it facilitates communication with streamers they like. Tips are typically accompanied by messages of encouragement or questions, and streamers respond to these messages. Viewers who donate seem to enjoy this attention from streamers—similar to how fans feel happy when a celebrity they admire recognizes them—as well as recognition from other viewers.

Although other revenue sources, such as advertising, exist, their role is limited, especially in the empirical context of this paper. Fact sheets from AfreecaTV (the focal platform of this paper) show that in 2019, 72% of the company's revenue came from tipping, while advertising accounted for only 17%.[15] In this paper, I assume that all revenue comes from tipping.

Rewards programs are commonly used to motivate streamers in this industry. For example, Twitch operates two such programs: Affiliate and Partner. Facebook Gaming classifies its streamers into three tiers: general gaming creators (default), Level

[11] https://www.cashnetusa.com/blog/highest-paid-twitch-streamers-world/

[12] https://www.businessofapps.com/data/twitch-statistics/, accessed on August 29, 2022.

[13] https://old.capitalwatch.com/article-4697-1.html

[14] As of 2025, Twitch no longer operates in the Korean market, and AfreecaTV has been rebranded as SOOP. In place of Twitch, Naver (the dominant search engine in Korea) launched its own streaming service, Chzzk, thereby preserving the duopoly structure of the livestreaming market.

[15] Source: https://corp.sooplive.co.kr/ir.php?page=irbbs, 2020 earnings release, retrieved on February 12, 2026.

Up creators, and Partners. The requirements and benefits of these programs vary across platforms. In general, platforms evaluate streamers based on total streaming time, viewership, and the number of "fans" (i.e., viewers who have bookmarked the streamer).[16]

2.2 AfreecaTV and Its Rewards Program

The empirical context of this study is AfreecaTV (which stands for "Anybody can Freely Broadcast TV"), a dominant live streaming platform in South Korea that held approximately 40% of the watch time share in 2020.[17] Established in 2006, AfreecaTV has been a market leader in South Korea, both in terms of market share and innovation. For example, the platform introduced a tipping system for viewers to support streamers as early as 2007, even before Twitch, the dominant player in the US market, was launched in 2011. AfreecaTV has maintained a mutually beneficial relationship with the popular e-sports culture in South Korea, regularly broadcasting high-quality video game competitions and hosting numerous streamers focused on video game content.

One prominent goal for streamers on AfreecaTV is to achieve *Best Broadcaster* status, which is the rewards program that this paper explicitly focuses on.[18] The primary benefit to streamers of achieving this status is to earn a commission discount. Prior to participating in the rewards program, streamers on AfreecaTV receive only 60% of the revenue generated from the tips they receive from viewers, while the platform takes a 40% commission. However, the commission rate for a Best Broadcaster is reduced by 10 percentage points, allowing streamers to receive 70% of the revenue they generate. According to the collected data, this benefit results in an approximate monthly revenue increase of 988 USD from the perspective of an average new rewards program participant.[19]

[16] For example, the minimum requirements to qualify as a Twitch Affiliate are: $\geq$ 50 fans, $\geq$ 500 minutes of total streaming time, $\geq$ 7 broadcast days, and $\geq$ 3 average viewers over the past 30 days. Source: `https://help.twitch.tv/s/article/joining-the-affiliate-program`, retrieved on March 24, 2022.

[17] `http://rank.afreehp.kr/view`, retrieved on Jan 31, 2023.

[18] Strictly speaking, there is one additional rewards program called the Partner Broadcaster. However, this rewards program involves a private contract between the platform and a streamer, and I can only observe the timing. Furthermore, acceptance to this rewards program happened only 14 times during the data period. For the purposes of this paper, I assume that the benefits for partner-level streamers are the same as those for Best-level streamers.

[19] To be precise, the original unit of revenue is an item called a starballon. Throughout the paper, I use the approximation that 10 starballons are equivalent to 1,000 KRW or 1 USD.

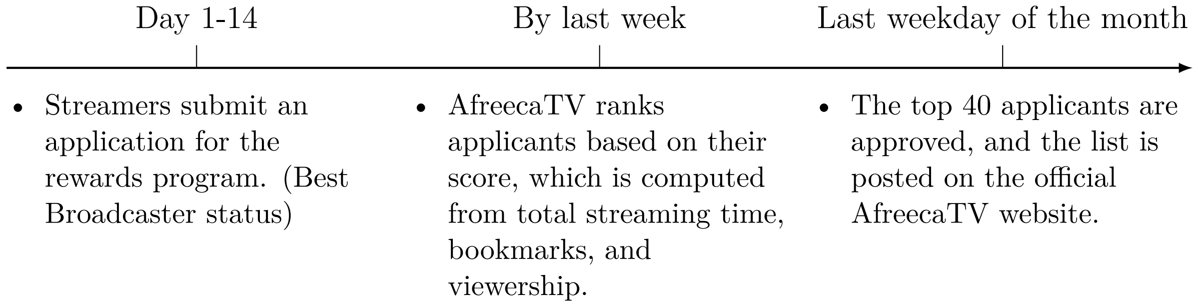


Figure 2. The monthly tournament timeline for the rewards program on AfreecaTV during the data period. (from Oct 2019 to April 2020) The score calculation table is available in Table A.8 in the appendix.

Importantly, the reward is effectively *permanent*. AfreecaTV approves new streamers into its rewards program on a monthly basis, as shown in Figure 2. Once approved, the only requirement to maintain eligibility is to stream for a minimum of five days, totaling 15 hours per month. This condition is easily achievable for streamers who have gained enough popularity to be approved for the rewards program. As a result, while only 40 out of around 6,000 streamers get the reward each month, streamers who received the reward account about 23% of the observation (See Table 2).

2.3 Data

I collected the main dataset for this study from two websites that track streamer information from AfreecaTV. Specifically, I gathered variables at the individual streamer-day level, which included streamers' revenue, streaming hours, bookmarks, average viewership, and watch time. Additionally, I assembled streaming hours by (broadcasting) category at the streamer-month level, which provides information about the content on which streamers focus. Lastly, information regarding the rewards program benefits and requirements, and monthly announcements of rewards recipients were obtained from the platform's official website. I provide definitions of key variables in Table 1.

To understand the coverage of the collected data set, I compute the platform revenue from the data and compare it with official financial reports posted on the platform's website. It turns out that my main data set covers 61.3% of the official tipping revenue as of the fourth quarter of 2019.[20] The gap is because streamer

[20]Specifically, for the fourth quarter of 2019, the platform revenue (from tipping) computed from the data was 18.5 billion KRW, while the financial report shows a platform revenue of 30.2 billion KRW.

Table 1. Key Variable Definition

Variable	Description
streaming hour	The number of hours a streamer streams each month.
watch time	The total duration for which viewers have watched a streamer. For example, if one viewer watches a streamer for 10 hours and another watches for 5 hours in a month (and there are no other viewers), the streamer's watch time for that month is 15 hours.
(total) revenue	The amount of "tip" that a streamer receives, which is split linearly between the streamer and the platform.
(broadcasting) category	A category for a streamer's broadcasting content (e.g. Game, Social)
bookmark	The number of viewers who have bookmarked a streamer, which serves as a measure of the streamer's fan base.
score	A performance metric used to determine eligibility for the rewards program. It is computed based on viewership, bookmarks, and cumulative streaming time.
$\mathbb{1}\{Best\ Broadcaster\}$	A dummy variable that indicates whether a streamer has been approved for the rewards program or not.
$\mathbb{1}\{exit\}$	A dummy variable that indicates whether a streamer appears in the next period or not.

tracking websites cannot track all tipping behaviors. This is because 1) the servers of streamer tracking websites can fail to record all tipping events when too many viewers are tipping simultaneously, and 2) some streamers are too unpopular to be tracked.

For this study, the data set was aggregated at the individual streamer-month level for two reasons. First, this approach greatly simplifies model building, given that the approval of the rewards program occurs on a monthly basis. Second, aggregation smooths out day-of-the-week effects.

Descriptive statistics at the monthly level are presented in Table 2. Two observations are worth noting. First, as expected, watch time and tipping revenue are highly skewed. Streamers at median only earn about 236 USD per month, while top 10% earn more than 6,000 USD per month. Second, streamers who have obtained the reward (Best broadcaster status) account for a substantial number of observations.

Table 2. Descriptive statistics of Key variables.

Variable	Mean	Std	P10	P50	P90
streaming hour	82.942	91.769	0.000	57.000	204.000
watch time (1,000)	10.629	118.024	0.001	0.768	10.222
(total) revenue (1,000$)	3.627	9.583	0.000	0.384	10.026
streamer's revenue (1,000$)	2.397	6.488	0.000	0.236	6.520
streamer score	44.355	27.332	12.000	38.400	81.600
$log(bookmark)$	6.968	2.106	4.454	6.760	9.930
$\mathbb{1}\{BestBroadcaster\}$	0.227	0.419	0.000	0.000	1.000
$\mathbb{1}\{exit\}$	0.080	0.271	0.000	0.000	0.000

Notes: The unit of observation is individual streamer-month. The streamer score and $log(bookmark)$ were measured at the beginning of each month. The score was computed using the platform's official score conversion table, based on observed total streaming hour, bookmarks, and viewership.

While only 40 streamers get the status each month, because this reward is effectively permanent, the reward receivers become common in the platform. The data period ranges from October 2019 to April 2020.[21] There are 8,282 unique streamers and a total of 41,221 observations.

2.4 Empirical Observations

2.4.1 Streamers focus on one broadcasting category

In theory, streamers have the flexibility to broadcast different types of content on different days. For instance, a streamer may stream a video game like League of Legends today and host a song contest tomorrow.

However, streaming hour data at the individual streamer-month-category level reveal that the majority of streamers adopt a "focus strategy". According to my category definition (either Game, Social, or Other), 90% of streamers have one category that accounts for more than two thirds of their total streaming hours.

Therefore, in my model, I assume that streamers have one broadcasting category that does not change over time. This category matters because streamers in different

[21] While there are some available data after April 2020, I focus on this period due to two events that significantly impacted the industry: COVID-19 and the increase in monthly program slots at the end of April. Starting from April 2020, the industry's state appears to be non-stationary. For instance, the fraction of streamers accepted into the rewards program has been consistently increasing over time, making it difficult to fit the data into a stationary model.

categories may have different marginal revenue/cost from streaming time, and streamers compete within a category. Streamers and their fans typically see streamers who broadcast similar content, such as the same video game, as their main competitors.

2.4.2 Revenue per watch time varies across individuals

There exists significant variation in profitability at the individual streamer-month level, as measured by revenue per watch time. It has a mean of 1.375 and a standard deviation of 14.224 (unit: USD per watch time). To understand the factors determining this profitability, I conducted a fixed-effects regression analysis.

According to the findings reported in Panel A in Table 3, individual (streamer) fixed effects explain a substantial portion of the variation. This result is intuitively comprehensible since tipping revenue is heavily influenced by personal traits such as communication skills and appearance.

My model abstracts from the possibility that the per watch time revenue can change within individual streamers due to approval to the reward program. To understand this issue, I regress individual streamer-month level per watch time revenue on the explanatory variables of streamer, month fixed effects, bookmarks, score, and Best Broadcaster status.

The first set of results in Panel B of Table 3 appears to suggest that per watch time revenue may increase after getting approved. However, the change seems to be driven by a short-run "congratulation effect", which means viewers temporarily tip more money to congratulate an approval. In the second set of regressions, I use a subsample that excludes periods from the month before and the month after an approval, and find the changes in per watch time revenue across approvals are not statistically significant.

Lastly, I additionally investigate how the watch time composition of streamers with heterogeneous profitability affects total tipping revenue, and whether this composition sharply changes within the data period. To do this, I divide streamers into nine groups. I first categorize them into three broadcasting categories (Game, Social, and Other). Then, within each category, I rank streamers based on their individual streamer level per watch time revenue and evenly split them into low, medium, or high profitability subgroups.

Figure 3 shows that watch time composition significantly affects tipping revenue, and there is no sharp change in composition during the data period. Within any broadcasting category, while the highly profitable streamers do not have a dominating

Table 3. Analysis of individual streamer-month level per watch time revenue

(Panel A: Explanatory Power of Fixed Effects)

$y = log(\text{per watch time revenue} + 1)$				
R^2	.852	.021	.057	.853
adjusted R^2	.816	.021	.057	.817
streamer FE	✓			✓
month FE		✓		✓
additional controls			✓	✓

(Panel B: Approval Effects on Per watch time revenue)

$y = log(\text{per watch time revenue} + 1)$	full sample			subsample		
$\mathbb{1}\{Best\ Broadcaster\}$ ($\hat{\beta}$)	.041	.051	.048	.020	.031	.025
$\mathbb{1}\{Best\ Broadcaster\}$ (s.e.)	(.015)	(.015)	(.015)	(.026)	(.026)	(.026)
streamer FE	✓	✓	✓	✓	✓	✓
month FE		✓	✓		✓	✓
additional controls			✓			✓

Notes: The unit of y variable is 1 USD per watch time approximately. In Panel B, the subsample refers to the sample after dropping observations from one month before to one month after an approval. Additional controls include bookmarks, score, and $\mathbb{1}\{BestBroadcaster\}$. To account for the skewness of the variable, I apply a log transformation and drop the top 2.5% of observations. The patterns remain similar when not applying the log transformation or using alternative 5 or 1% drop cutoffs.

share of watch time, their tipping revenue is much greater than the other two groups.

2.4.3 Streamers with lower profitability also receive rewards

Streamers approved for the rewards program are not necessarily those with the highest per watch time revenue. Within each broadcasting category, I divide streamers into three subgroups–low, medium, and high–based on their per watch time revenue. I then investigate which subgroup was approved. Among the approved streamers, the proportions of the low, medium, and high subgroups were 13.6%, 30.8%, and 55.6%, respectively.

While the program benefit (commission discount) is undoubtedly more appealing to streamers with high per watch time revenue, they may incur higher effort costs for streaming, such as producing higher quality broadcasting content. As a result, their response to the program benefit could be less pronounced than what one might expect when solely considering their revenue.

Interestingly, the platform does not seem to directly incentivize the approval of high per watch time revenue streamers. The score used for program acceptance is computed based on average viewership, bookmarks, and total streaming time. Per watch time revenue is not taken into consideration.

Two reasons could explain this. First, providing the commission discount benefit to high per watch time revenue streamers would result in greater program expenses. Second, if the platform were to directly incorporate per watch time revenue into the approval process for the rewards program, streamers might easily manipulate this by kicking out viewers who do not tip frequently.

2.4.4 Not all eligible streamers receive the reward

One fundamental data problem is that I can observe which streamers have newly received the reward, but I cannot observe which streamers have applied to the reward program. To formulate the streamer's problem, I need to determine two probabilities: the probability that streamers apply to the reward program ($\mathbb{P}\left(application\right)$), and the probability that streamers get approved to the reward program, conditional on their application and their score ($\mathbb{P}\left(approval|application\right)$).

To account this, I focus on streamers having $score \geq 66$, and use their observed probability of newly receiving the reward to determine $\mathbb{P}\left(application\right)$.[22] They have

[22] An implicit assumption is that the probability of application is the same across scores.

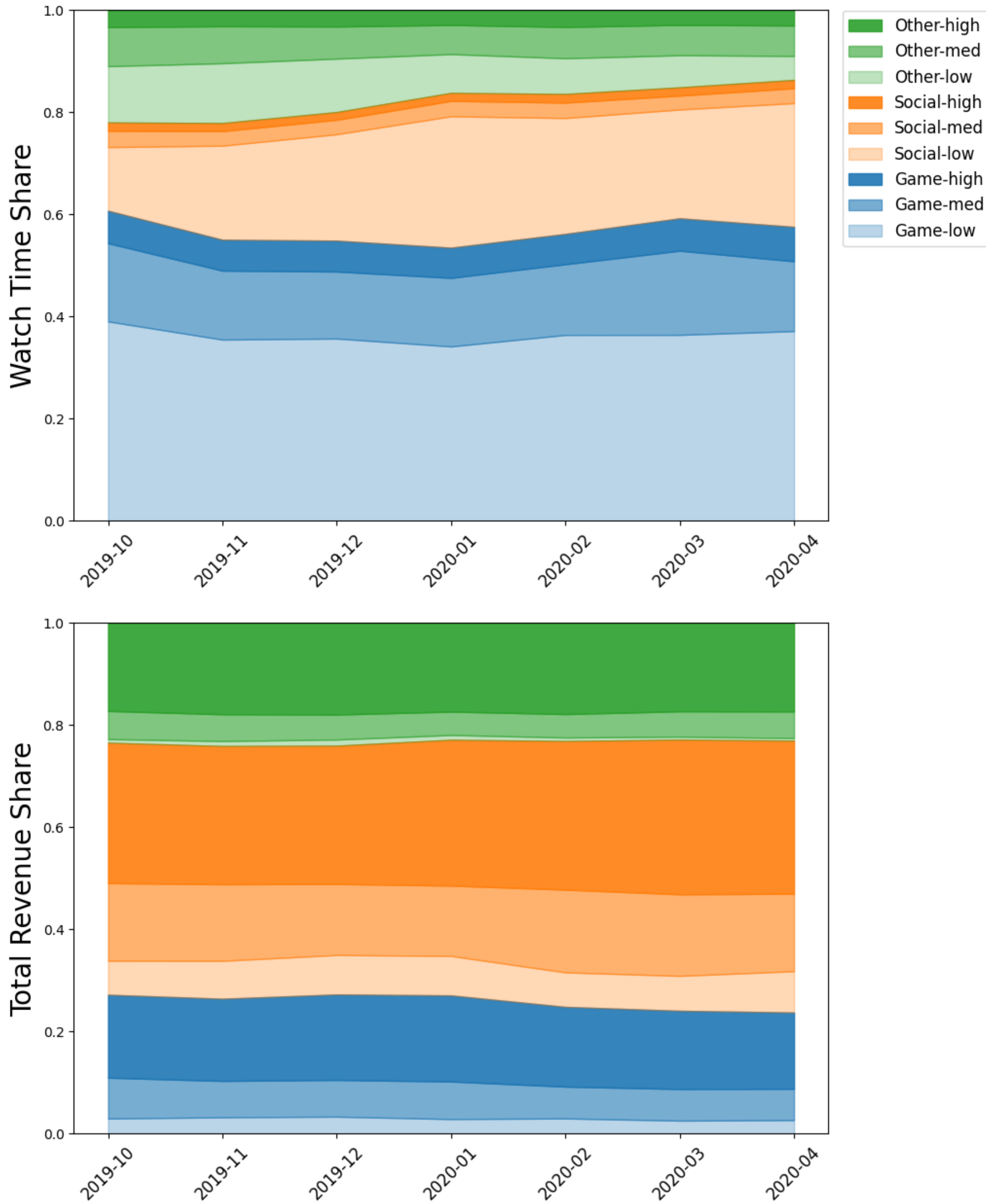


Figure 3. Watch time and revenue shares of streamers with varying profitabilities, measured by per watch time revenue. The first label represents broadcasting categories, while the second label indicates whether streamers have low, medium, or high per watch time revenue within each category.

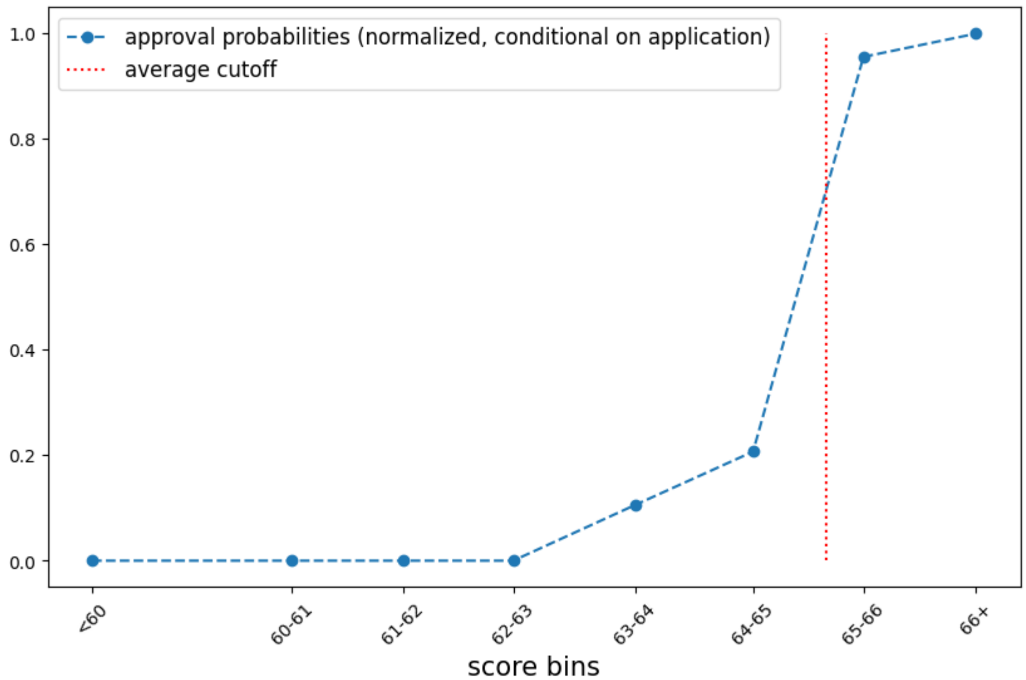


Figure 4. Probabilities of being approved by the reward program by score range.

Notes: The probabilities were calculated based on observed probabilities of newly receiving the reward. The probability for the $score \geq 66$ group was set to 1 because they will receive the reward if they apply, while the probabilities for other groups were normalized accordingly.

scores that are higher than a typical cutoff of 65 (during the data period), so if they do not receive the reward, it is because they had not applied. I assume that $score \geq 66$ group receive the reward with probability 1 if apply, and normalize $\mathbb{P}(approval|application)$ by score accordingly (Figure 4)[23]. These application probability and reward receiving probability will be used during estimation stage, and the latter will change during the counterfactual stage.

In my model, streamers draw an iid shock that determines whether a streamer applies to the program or not in each period. This shock determines probability of application, and conditional on application, streamers have reward receiving probabilities depending on their scores. I recap this set up in subsection 4.3.2. Limitations and future directions of this specification are discussed in section 3.4.2.

To empirically understand this observation better, I tracked streamers who were eligible for the reward program during the data period for three years, and found ap-

[23]For the sake of monotonicity, I have excluded 5 outlier cases (0.02%) where streamers with a $score \leq 62$ received the reward. These cases are likely due to measurement errors or exceptional circumstances.

proximately 80% of them receive the reward eventually.[24] Therefore, despite frequent delays, the majority of AfreecaTV streamers appear to eventually receive the reward in the long run.

To investigate the reasons behind this delay, I conducted interviews with several streamers in the platform.[25] The interviews revealed the presence of behavioral and strategic factors. Some streamers were initially unsure about their chances of getting approval despite having a score higher than a typical threshold, but they eventually applied and succeeded on their first attempt. Others mentioned the strategic consideration of multihoming as a reason for delaying their application. Once participating in the rewards program, streamers are not permitted to livestream simultaneously on other platforms.[26]

3 Model

3.1 Overview

The goal of my model is to capture how the rewards program affects heterogeneous streamers' decision-making and, consequently, the platform's revenue. Thus, the primary focus is on describing the problem faced by streamers in different states.

I build a model based on a dynamic game framework for two reasons. First, the rewards program operates dynamically, in that streamers work diligently today with the expectation of future period benefits. Second, the interaction between streamers significantly matters in two ways. From a perspective of a single period payoff, streamers earn less revenue if there are more competitors broadcasting similar content. From a dynamic perspective, since the rewards program approves a streamer based on a performance based tournament, and the scores of other streamers affect the likelihood of a focal streamer achieving approval.

Streamer heterogeneity is captured by their state variables. First, they have different broadcasting content and profitability, as measured by revenue per watch time. I assume that these two factors are time-invariant and individual-specific, as reflected

[24] Specifically, 981 streamers were eligible to receive the reward during my data period (Oct 2019 - Apr 2020). I tracked them until Feb 2023. 80% (506 out of 628) eventually received the reward, except for those who exited before then.

[25] I contacted streamers who received the reward by using the messaging function on the platform. I asked them 1) if they knew they would be accepted when applying, and 2) what reasons they believe some popular streamers do not apply for the reward program.

[26] Nevertheless, streamers are still allowed to post recorded videos on other platforms or broadcast on other platforms at different times.

by the empirical observations mentioned above. Second, they have varying numbers of bookmarks, streamer scores, and Best Broadcaster status, all of which evolve over time. The number of bookmarks represents the accumulated fan base. The streamer score is used for determining acceptance into the rewards program. The Best Broadcaster status is an indicator of whether they have already been admitted to the rewards program.

In the model, streamers draw an iid shock in each period that determines whether they apply to the rewards program or not.[27] Conditioning on application, they get approved stochastically based on their score. The design of the rewards program, the number of approval slots, and the commission discount all affect streamers' decisions by changing the approval likelihood by both score and any expected dynamic financial gain. For example, providing more reward program slots leads to a higher approval likelihood for the same score, and streamers respond by increasing their efforts. While this act alone can improve platform revenue, providing commission discounts to more streamers creates an additional expense that reduces platform revenue.

Streamers derive utility from the revenue they earn and solve dynamic programs (DP) to maximize the discounted sum of their single period payoffs. They first decide whether to exit the market or not. If they choose to stay, they must determine how many hours to stream. Streamers can increase their streaming efforts, which increases their watch time and revenue earned for the current period. Further, they also can expect an increase in the probability of acquiring both additional bookmarks and a higher streamer score at the beginning of the next period. However, working more to increase streaming efforts also entails higher effort costs.

The interaction between streamers within a streamer's DP is modelled as follows: I assume that streamers do not believe that their choices will affect the streaming choices of any other streamer, and I estimate the model under this behavioral assumption. However, in equilibrium there will be an effect as additional streaming will reduce the expected viewing and acceptance probabilities of other streamers, and this will be accounted for in my counterfactuals.

More specifically, during the counterfactual stage, I search for equilibria that are consistent with a tournament design (for instance, when approving the top 20 streamers in each period). This is achieved by adjusting approval probabilities across scores. Streamers do not consider complete information, such as the entire score distribution

[27]This is needed to explain some streamers who have enough score but not participate in the rewards program, as described in section 2.4.4

across all streamers. Nevertheless, if streamers believe that their individual actions will not affect the overall market-level score distribution, the approval probabilities become sufficient for calculating the expected dynamic gain from the potential reward.[28]

While solving their DP, streamers take actions based on their beliefs regarding the state-action profiles of other streamers. The model is designed to find a symmetric, stationary equilibrium, where a streamer believes that any aggregate moment that captures influence from other streamers is permanently fixed at a specific point. This belief is consistent at the equilibrium in the following sense: if all streamers decide on an action based on this belief, the resulting realization of the aggregate moment at a steady state aligns with the belief.[29]

3.2 Streamer's Problem

This section presents the formal setup of a streamer's problem.

3.2.1 Environment

Time The time is discrete and indexed by a subscript t. The time unit is month, and the common time discount factor is denoted by β.

Players The players of the game are streamers indexed by a subscript j.

States Streamer j in period (month) t is represented by a state vector x_{jt}. This vector has two time-invariant state variables: revenue per watch time α_j^{rev} and contents category c_j. I use a subscript c to denote a category.

In addition, the vector has three state variables that evolve over time: $log(bookmark)_{jt}$, $score_{jt}$, and $r_{jt} \equiv \mathbb{1}\{Best\ \ Broadcaster\}_{jt}$, which denotes whether streamer j in month t has been approved to the rewards program or not.

Shocks Four independently and identically distributed shocks affect a streamer's decision. The first shock is an exogenous exit shock, denoted as $\xi_{jt} \in \{0, 1\}$. A streamer exits the market exogenously if $\xi_{jt} = 1$, capturing exits not related to

[28] I discuss this point more extensively in section 3.4.1.

[29] Since my model incorporates shocks that are iid across individuals and over time, the realized moments may vary ex-post. Nonetheless, the expected value of these moments remains constant at an equilibrium, and this value is what streamers take into account.

profit, such as mandatory military service.[30] The second shock is a scrap value draw, represented as $\phi_{jt} > 0$, which captures a streamer's outside option.

The third shock is a program interest shock, denoted as $\mathcal{I}_{jt} \in \{0, 1\}$. A streamer applies for the rewards program if $\mathcal{I}_{jt} = 1$. Lastly, the fourth shock is a streaming cost shock, denoted as $\gamma_{jt} > 0$. This shock captures random events that make a streamer's streaming easier or harder, for example, health issues or updates of video games.[31]

Actions Each month, a streamer makes two decisions: exit and streaming time (in hours). First, a streamer may exit, either exogenously or endogenously. A streamer exits endogenously if a scrap value draw is greater than the expected net present value of continuation. If not exit, a streamer decides how many hours to stream.

State Dynamics Three states, $log(bookmark)_{jt}, score_{jt}$ and r_{jt}, each evolve in a Markov fashion. The first two states are discretized and assume that streamers can either go up/down one interval or can stay each period. All streamers have a higher probability of advancing to the higher interval when they stream more.

For r_{jt}, I assume streamers have new approval probabilities, i.e. $\mathbb{P}\left(r_{jt+1} = 1 | r_{jt} = 0\right)$ determined by their score, conditioning on $\mathcal{I}_{jt} = 1$ draw, i.e. they apply to the rewards program. If $\mathcal{I}_{jt} = 0$, approval probabilities are zero. If $r_{jt} = 1$, $r_{jt'} = 1$ for all $t' > t$ because an approval to the program is permanent.

Payoff Streamers obtain quasi-linear utility from their share of revenue. That revenue comes from the level of watch time that streamers gain each period.[32] Note that streamers have different abilities to extract revenue from the given amount of watch time, represented by their time-invariant per watch time revenue α_j^{rev}. Let $\mathcal{W}$ denote watch time, and c_1, c_2 denote quadratic streaming cost. For streamer j on period t who streams h_{jt} hours, the single period payoff is

$$u_{jt} = \underbrace{\left(0.6 + 0.1 r_{jt}\right) \alpha_j^{rev} \mathcal{W}\left(h_{jt}, x_{jt}, h_{-jt}, x_{-jt}\right)}_{\text{streamer's revenue}} - \underbrace{\gamma_{jt}(c_1 h_{jt} + c_2 h_{jt}^2)}_{\text{streaming cost}} \tag{1}$$

[30] This idea is similar to Chen and Xu (2022).

[31] Usually, when a new update of a video game is released, streamers broadcasting the video game find it easier to create content by exploring the new features.

[32] Some streamers may also derive non-financial returns, such as attention, from streaming. In my model, this could be accounted for by incorporating smaller streaming costs.

where h_{-jt} and x_{-jt} denote the streaming hour choices and states of all streamers other than streamer j during period t. The coefficient $(0.6 + 0.1r_{jt})$ is used to capture the empirical fact that streamers take 60% of generated revenue, and an additional 10% after they have been approved for the rewards program.

Timeline In each period, the sequence of events is as follows.

1. Streamers draw $\mathcal{I}_{jt}$ and an exogeneous exit shock ξ_{jt}. If $\xi_{jt} = 1$, they exit the industry exogenously.
2. Streamers draw a scrap value ϕ_{jt} and exit if it exceeds their exit cutoffs.
3. Streamers draw a cost shock γ_{jt} and determine their streaming hours by solving the following Bellman equation (2) conditioning on continuation. Because I focus on a steady state equilibrium, I drop t subscript and use $'$ to denote next period.

$$V\left(h_j|x_j, h_{-j}, x_{-j}\right) = \max_{h_j \in \ \{0,50,\cdots,300\}} \left[u + \beta \sum_{x'} \mathbb{P}\left(x'|x, h\right) \tilde{V}\left(x'\right)\right] \tag{2}$$

   where x, h without subscripts denote all players' state/hour, and $\tilde{V}\left(x'\right) = \mathbb{E}_{\xi,\phi,\gamma,\mathcal{I}}\left[V\left(x'\right)\right]$ represents the ex-ante value function.
4. Realization of transitions in $log(bookmark)_{jt}$ and $score_{jt}$.
5. Realization of transitions in r_{jt}, i.e. some streamers receive the Best Broadcaster status, leading to an increase in their r from 0 to 1 (permanently).
6. New entrants enter the platform, and the industry proceeds to a new state.[33]

3.2.2 Simplifying Assumptions

Without any simplification, any streamer must take into account all other streamers' states and actions. There are about 6,000 streamers appearing in the data each month. This implies that, even when there are only two states, a streamer would have to evaluate more than 2^{6000} possible configurations to develop a holistic contingency plan. To tackle this challenge, I impose two simplifying assumptions.

[33] In this paper, I assume that streamers' entry is exogenous and use the observed entry process, denoted as v^e, as described in Section 5.2. Solving streamers' DP determines the expected transition probabilities from one state to another, which in turn determines an industry-level transition matrix, denoted as M_t. Using these two variables, the industry state s_t evolves according to the equation $s_{t+1} = s_t M_t + v^e$.

Assumption 1. *A streamer's state and action do not directly affect other streamers' transition of state variables.*

More specifically, whether streamer j starts with a higher $log(bookmark)$ and $score$ in the next period depends solely on the value of j's own streaming hours, state, shock draws. Likewise, whether j gets newly approved to the rewards program this period, i.e. $r_{jt} = 0$ but $r_{jt+1} = 1$ depends only on j's own score (and the corresponding exogenously determined approval probability) and action.[34]

This assumption rules out a situation where other streamers' actions affect a focal streamer's bookmark (fan base) accumulation. One possible justification for this assumption is that the market is sufficiently large, on average, that other streamers' states and actions could all be averaged out at the steady state equilibrium.

Nevertheless, the competition effect between streamers still indirectly influences the progression of a streamer's own bookmarks and scores. This results from the streamer's individual choice of streaming hours effort, which affects both watch time and the progression of the streamer's bookmarks and scores. When other streamers within the same category stream more, each streamer may choose to stream less due to the reduced expected watch time for the day. Consequently, the growth of their bookmarks and scores becomes less likely.

From a tournament model perspective, this assumption implies streamers abstract from their exact ranking, and instead consider only exogenously determined approval probabilities across streamer scores. A more detailed discussion regarding this simplification is provided in section 3.4.1.

Next, I simplify the interaction between streamers through a single-period payoff. When watch time $\mathcal{W}$ is given, a streamer j's revenue is determined by j's own state variables: revenue per watch time α_j^{rev} and r_{jt}. Static interactions among streamers are reflected through a streamer's watch time $\mathcal{W}$ because the value of $\mathcal{W}$ can be affected by all other streamers' states and actions in theory. For example, when one popular streamer broadcasts the same video game as another streamer, the latter streamer's watch time is likely to decrease because some viewers might switch to the more popular streamer.

I impose a simplification on this watch time function in the model by assuming that an aggregate variable at the category level fully captures the influence of other streamers on watch time.

[34] Viewership affects the approval probability, but only through affecting the score, both in the real world and in my model.

Assumption 2. *Category level log(bookmark) weighted sum of streaming hours* ψ_{ct} *fully captures other streamers' influence on streamer j's watch time. That is,*

$$\mathcal{W}\left(h_{jt}, x_{jt}, h_{-jt}, x_{-jt}\right) = \mathcal{W}\left(h_{jt}, x_{jt}, \psi_{c(j)t}\right) \tag{3}$$

where h_{jt} *is streamer j's streaming hour on period t,* $c(j)$ *is j's broadcasting category, and* $\psi_{c(j)t} := \sum_{k\in c(j)} log(bookmark)_{kt} \cdot h_{kt}$*. Moreover,* $\mathcal{W}$ *weakly decreases with respect to* ψ_{ct}*.*

While watch time can potentially be influenced by the states and actions of all other streamers, I assume that it is a function a focal streamer's state, action, and ψ_{ct} only. Therefore, when approximating the watch time in section 4.3.3, the independent variables are restricted to these three. When approximating $\mathcal{W}$, across month variation of ψ_{ct} is used to determine the degree to which a streamer's watch time decreases with increasing ψ_{ct}.

However, ψ_{ct} does not vary over time (but does vary across categories) at a steady-state equilibrium. Further, streamers believe that ψ_{ct} is fixed at a certain value both now and forever regardless of their actions. When computing the pseudo log-likelihood for each parameter guess, the model that all data observations are derived from a single equilibrium in which ψ_{ct} is fixed at the observed average, and streamers rationally take this as given. Subsequently, ψ_{ct} is endogenized, as described in section 5.2.

There are two implications for this model that arise from this assumption. First, from the perspective of a single-period payoff, there are no cross-category interactions, except for the number of approval slots that are allocated at the platform level. Second, the total market size in the model, as measured by watch time, may increase as streamers increase their streaming. However, the monotonicity assumption implies a weakly negative network effect exists between streamers within the same category. This restriction could be helpful for alleviating concerns about multiple equilibria, while not completely solving them.[35]

In this assumption, I abstract from across-category interactions and focus only on within-category interaction, as based on two observations. First, streamers stated that an important determinant of their viewership is if other popular streamers are broadcasting similar contents. Second, XSplit, a commonly used streaming aiding

[35]For example, if a streamer's watch time increases when other streamers have more streaming hours, one can easily think of two equilibria: either no one streams, or everyone streams extensively.

tool advises that it could be a good idea to avoid head-to-head competition with streamers broadcasting the same contents.[36] For these reasons, I rule out across category spillovers and capture within category interactions in a simplified fashion.

3.3 Equilibrium

I focus on a symmetric steady state equilibrium. Thus, the time subscript t is dropped in this section. This implies the market is assumed to be in a steady state, and that the streamers also believe this is true. Note that streamer interaction is fully captured by the category-level, weighted sum of streaming hours $\{\psi_c\}$. Given these values, streamers simply solve a single-agent dynamic programming (DP) problem.

In essence, the equilibrium of this model is reached when streamers' beliefs and realizations of $\{\psi_c\}$ in the steady state coincide. Streamers solve their DP and determine the optimal policy, believing that $\{\psi_c\}$ is fixed at $\{\psi_c^{old}\}$ now and forever. With this belief, streamer j's policy $\mu(x_j)$ becomes a function of their own state. It includes an exit cutoff, $\rho(x_j)$, and a streaming hour conditional on continuation, $h(x_j)$

From the resulting optimal policies, the corresponding realization (at steady state) of $\{\psi_c\}$, as denoted by $\{\psi_c^{new}\}$, can be computed. In doing so, two additional pieces of information are used: shock distributions (e.g., health issues that perturbate streaming cost) and the entry process. I impose parametric assumptions for shock distributions and determine their parameters during the estimation stage. Then, I use the empirically observed entry process, i.e., the number of new streamers appearing at the state level in the data each month.

Those information and policy μ determine an industry level transition matrix, $M_{\xi,\mathcal{I},\mu}$. Combined with an entry process vector, v^e, the corresponding long-run industry state can be determined by the following equation.

$$s_{\xi,\mathcal{I},\mu} = v^e + v^e M_{\xi,\mathcal{I},\mu} + v^e M^2_{\xi,\mathcal{I},\mu} + \cdots = v^e(I - M_{\xi,\mathcal{I},\mu})^{-1} \tag{4}$$

where I denotes an identity matrix. This steady state and the policy μ determines how many streamers are at each state, and how many hours they stream. Thus it is straightforward to compute corresponding the new $\psi_{ct} \equiv \mathbb{E}_{\xi,\mathcal{I},\gamma}\left[\sum_{k\in c} log(bookmark)_{kt} \cdot h_{kt}\right]$, where the expectation was taken with respect to iid shocks that streamers draw each period.

[36] https://www.xsplit.com/blog/best-time-to-stream, *"If you're a Tekken (a video game) streamer, maybe it might be good to pick a time that the most popular Tekken streamers aren't on."*, retrieved on June 30, 2023.

The equilibrium is where the realization of aggregate variable $\{\psi_c^{new}\}$ coincides with initial "old belief" $\{\psi_c^{old}\}$. I provide a formal definition of the equilibrium concept below.

Equilibrium Definition. *The equilibrium in this paper consists of a policy function $\mu^*(x_j) = \{\rho^*(x_j), h^*(x_j)\}$ and category level weighted sum of streaming hours $\{\psi_c^*\}$ that satisfy the following conditions.*

1. *(Optimal Policies) Under the belief that $\psi_{ct} = \psi_c^*$ presently and in the future, the policy μ^* satisfies the followings:*

   (a) h^* solves (2), a streamer's Bellman equation conditioning on continuation.

   (b) ρ^* solves a streamer's optimal exiting problem, in the sense that $\rho^*(x) = \mathbb{E}_\gamma [V(x)]$. In other words, streamers endogenously exit if and only if the scrap value exceeds the expected continuation value.[37]

2. *(Consistency) If all players follow the policy μ^*, $\{\psi_c^*\}$ is consistent with the corresponding expected steady state-action realization given by (4) and μ^*.*[38]

From a methodological perspective, this equilibrium concept is similar to the Oblivious Equilibrium proposed by Weintraub et al. (2008, 2010) in the sense that a streamer believes that other streamers' states are permanently fixed and not affected by their actions. In addition, I assume that streamers believe the distribution of what everyone else does is fixed. This industry state-action profile information is then collapsed into a single aggregate variable, and used to capture the influence from other streamers, as described in Assumption 2. Streamers believe this aggregate variable remains invariant over time ($\psi_{ct} = \psi_c^*$).[39]

3.4 Discussion

Before proceeding to estimation, the following section clarifies a number of features and limitations of my model.

---

[37] This endogenous exit is orthogonal to exogenous exits caused by ξ_{it} draw. The former is a streamer's decision based on expected discounted sum of profits and scrap value draw. The latter represents a "forced" exit unrelated to profits, such as a mandatory military service.

[38] The realization of this weighted sum of streaming hours is stochastic because of iid shocks that streamers draw within each period. What streamers consider is the expected value of this realization at the beginning of each period.

[39] One might also think ψ_{ct} like aggregate moments used in Moment-based Markov Equilibrium approach in Ifrach and Weintraub (2017), which is an extension of Oblivious Equilibrium. However, I abstract away from computing a perceived transition kernel of moments by assuming $\psi_{ct} = \psi_c^*$ permanently.

3.4.1 Tournament Feature

Regarding the tournament to decide who will be approved, my simplifying assumptions imply that streamers consider program approval probabilities as a function of their own score, without taking into account the entire score distribution and their exact ranking. In other words, streamers behave as if the approval probabilities are exogenously given.

If streamers believe that their individual actions do not significantly influence the overall score distribution and ranking, this simplification can still capture the tournament effect. For instance, consider a scenario where one person participates in a math contest involving 1,000 other participants, and the top 100 contestants receive a prize. Also assume that this person knows the score cutoff will be 90 out of 100, and the player's own actions alone cannot change it. In deciding how much effort to put in, knowing that the probability of winning the prize is 0 for scores below 90 and 1 for scores above 90, that would provide sufficient information to determine an optimal solution. Additional information about the complete score distribution across other players would not change that person's decision.

Another justification is that if the platform is large and the number of other streamers applying to the rewards program is unobservable, then it is challenging for streamers to know their true ranking in the real world. In such cases, considering the approval probabilities could be sufficient to describe a streamers' behavior.

Indeed, during my interviews with streamers who were recently accepted into the rewards program, I found that they only had a rough idea about their acceptance probability, let alone exact rankings. One streamer told me that he delayed his application for several months because he was not sure if his score was enough to get an approval. This was despite his score already was above the typical cutoff. Ultimately, he got accepted in the first trial.

At the counterfactual stage, I change approval probabilities at the outer loop until the resulting equilibrium aligns with the intended program design. (e.g. the number of approved streamers doubles). This approach enabled me to incorporate the dynamic influence of other streamers, at least to a certain extent. For example, if other streamers respond quickly to an increase in approval likelihood, the number of streamers getting newly approved at equilibrium will also increase quickly, making approval probability increase stop quickly at the outer loop. From a focal streamer's perspective, it could look like other streamers work harder, so the streamer would have a lower chance of getting approved in the tournament.

3.4.2 Application Decisions

I assume that streamers apply to the rewards program in an iid fashion, with $\mathcal{I}_{jt} \in \{0, 1\}$ draws. Each period, they apply to the rewards program if and only if $\mathcal{I}_{jt} = 1$. This shock is necessary to explain the fact that a substantial fraction of streamers do not apply to the rewards program despite being approved if they had applied (see subsection 2.4.4).

In the real world, some streamers may persistently show no interest in the rewards program. For the current version of the model, I assume that $\mathcal{I}_{jt}$ is drawn iid across streamers and across time. Because the length of the panel is short (seven months), it is difficult to identify the distribution of this persistent and unobserved heterogeneity.

3.4.3 Market Expansion, Stealing, and Network Effects

This model allows for both market expansion and market stealing, but it rules out any positive network effects among streamers within a given category. On the one hand, streamers may gain more watch time when they stream more, which can lead to an increase in their market size, as measured by total watch time. On the other hand, a streamer's watch time decreases when other streamers in the same category stream more (weighted sum of streaming hours ψ_c increases), capturing net market stealing/expansion effects across streamers. The monotonicity assumption with respect to ψ_c implies that, across streamers, the market stealing effect should weakly dominate.

My model abstracts from across-category interactions, except for the tournament approach towards the rewards program, which is conducted at the platform level. Therefore, there are no across-category network effects for watch time, and within-category network effects are assumed to be weakly negative. The shape restriction in assumption 2, where a streamer's watch time decreases as ψ_c increases, reflects this assumption.

3.4.4 Focus on Platform Revenue Instead of Profit

While a platform's ultimate goal would be to maximize its profit, this paper focuses on platform revenue. To clarify, this platform revenue accounts for share changes resulting from offering commission discounts, but does not include other cost changes, such as increased server maintenance costs due to the rewards program.

My reasoning is that compared to revenue change, such cost change is likely to be small. When the rewards program design changes, the main cost change associated with it is the dedicated internet circuit expense change due to changes in watch time. AfreecaTV's 2019 Q4 earnings fact sheet shows that the amount of internet circuit expense is about 7.7 percent of revenue generated from commissions collected on viewer tipping.[40]

4 Estimation

4.1 Overview

In the structural estimation, the watch time function is approximated first, as are some shock distributions and transition dynamics. Next, I estimate five model parameters: streaming costs c_1, c_2, scrap value distribution parameters K_0, K_1, and cost shocks dispersion parameter σ_γ.

These parameters are estimated for each category-α^{rev} subgroup separately. This allows for natural flexibility, such as streamers with higher revenue per watch time α^{rev} who may have better communication skills can enjoy a greater outside option on average.

I estimate these parameters through pseudo maximum likelihood estimation. I calculate a pseudo log-likelihood based on the probabilities of actions, such as exiting the platform or not, and the choices of streaming hours.[41] This log-likelihood is then maximized using a simplex method. To compute the log-likelihood for each parameter guess, I utilize a full solution method and solve a streamer's dynamic problem through policy iteration.

To implement this scheme, I first discretize the states and make additional simplifying assumptions. Next, I determine the probabilities of exogenous exits and their application to the rewards program. Lastly, the transition dynamics are approximated using a variant of a linear probability model. Further, watch time is approximated with a tree-based model that incorporates shape restrictions. These measures ensure that the model can replicate all observed actions and prevent the log-likelihood from

[40]The fact sheet is available from `https://corp.afreecatv.com/ir.php?page=earning`. Specifically, the circuit expense is about 2.3 billion KRW, while the revenue from tipping was about 30.2 billion KRW.

[41]This approach is pseudo because I fix a theoretically endogenous aggregate variable (weighted sums of streaming hours at the category level) at the observed average to reduce computational burdens. The justification for this is that streamers have rational expectations about the actual realization.

approaching negative infinity.

For the remaining part of the paper, I set the time discount factor $\beta = 0.99$. The time unit is short (a month), but having β too close to 1 may yield technical problems, such as making game solving very slow. Additionally, streamers' popularity is volatile, which may lead them to further discount future benefits.

4.2 Discretization

The platform provides an official code for broadcasting contents. To discretize broadcasting category, I collapse these formats into three broad categories: video game, social, and others. The first category includes streamers who play mobile/PC video games like League of Legends, Fortnite, and FIFA Online. The second category consists of streamers who engage in interactive conversations with their viewers, often incorporating dancing and singing in response to tipping. The last category encompasses streamers with diverse content, ranging from boat fishing and woodworking to stock price prediction.

Next, the continuous state variables of per watch time revenue α^{rev}, $log(bookmark)$, and *score* are discretized. In short, streamers are divided into each broadcasting category as classified by one of three groups (low, medium, high). This is based on their individual level α_j^{rev}, and discretize scores more granularly around the average approval cutoff. Details of this further discretization are summarized in Appendix B.1.

Lastly, I discretize streaming hour to one of $\{0, 50, \cdots, 300\}$ (hours).

4.3 Parametrization & Calibration

4.3.1 Shocks

I parameterize distributions of independently and identically distributed shocks as follows. First, scrap value $\phi_{jt} \overset{\text{iid}}{\sim} exp(K_0 + K_1 \cdot bookmark_{jt})$. Second, a multiplicative cost shock $\gamma_{jt} \overset{\text{iid}}{\sim} lognormal(0, \sigma_c^2)$.

Next, the parameters for binary shocks are determined directly from data. For the exogenous exit shock ξ_{jt}, $\mathbb{P}(\xi_{jt} = 1)$, i.e. probability of exogenous exit is 0.0128. While the observed exit probability decreases across popularity measured by $\log(bookmark)$, it becomes roughly constant for values in the range $\log(bookmark) \geq 12$ at 0.0128.

This possibility of exogenous exit is useful for two reasons. First, it captures exits not related to revenue (e.g., mandatory military service). Second, it prevents a situation where streamers in certain states almost never exit, causing the number of streamers in these states to become non-stationary.[42] Lastly, I discuss an application shock that determines whether streamers apply to the reward program or not below.

4.3.2 Application and Approval Probabilities

Streamers draw $\mathcal{I}_{jt} \in \{0, 1\}$ and apply to the reward program if $\mathcal{I}_{jt} = 1$. Each period, $\mathbb{P}\left(application\right) = \mathbb{P}\left(\mathcal{I}_{jt} = 1\right) = 0.067$.[43] This number represents the observed probability that streamers with a $score \geq 66$ receive the reward. Such streamers have scores higher than typical cutoffs, so their probability of receiving rewards would be $\mathbb{P}\left(application\right)$. Conditional on application, for the probabilities of getting approved by the program, I use the observed and normalized probabilities in Figure 4 (section 2.4.4). These probabilities will be considered exogenous and will be endogenized during the counterfactual stage.

4.3.3 Functional Form Specifications

I choose functional forms for watch time and transition dynamics involving $log(bookmark)$ and $score$. These functional forms are selected to make both types of gains weakly concave with respect to streaming hours. This aligns with a natural restriction that marginal gain decreases weakly, and this characteristic is helpful in ensuring that the model can generate all possible hour choices.[44]

Watch Time $\mathcal{W}$ is approximated using a tree based, machine learning model with shape restrictions. Note that $\mathcal{W}$ is a function of a streamer's state, specifically $log(bookmark)_{jt}$, r_{jt}, broadcasting category, α_j^{rev}, streaming hours, and $log(bookmark)$ weighted sum of category level streaming hours ψ_{ct}.

More specifically, I use a model called XGBoost (Extreme Gradient Boosting tree) and non-parametrically approximate its prediction to a function that is concave with

[42] In theory, because the scrap value distribution is assumed to follow an $exp()$ distribution with the support $[0, \infty]$, the endogenous exit probability is strictly positive for all states. However, in practice, I found that without exogenous exits, some states have endogenous exit probabilities as low as 10^{-27}, which makes streamers in the state almost never exit.

[43] While this number may seem too low, it does not imply that streamers do not care about rewards. The number appears small because streamers interested in rewards have already received them, so they are not considered in the new reward receiving probability. Additionally, as mentioned in section 2.4.4, 80% of eligible streamers eventually receive the reward.

[44] Further elaboration of the latter point is provided in Appendix B.2

respect to streaming hour.[45] Also, assumption 2 is directly incorporated into the approximation, resulting in a decrease in watch time as ψ_{ct} increases by imposing monotonicity constraint when fitting the model. Details for this XGBoost model, including a comparison with other models and fit assessment, can be found in appendix B.3.

This approach offers two advantages. First, shape restrictions that are desirable for model solving and computation can easily and explicitly be imposed. For example, marginal watch time gain decreases with respect to streaming hour. Second, this approach exhibits superior out-of-sample predictive accuracy, even after the shape restrictions were imposed. On average, compared to the nested logit model, I observed an improvement of approximately 24% in out-of-sample root mean squared error.

Alternative models that I have considered were a nested logit model and complete nonparametric approach. These options were rejected for technical issues. First, logit models generate watch time predictions that are *convex* to streaming hours because of $exp(\cdot)$ function in the numerator. This feature causes the marginal benefit from streaming to increase. Combined with the quadratic cost of effort that I assume, this version of model makes streamers choose either not to stream at all or to stream for the maximum possible hours.[46] Therefore, the model cannot generate all streaming hour choices observed. I describe this problem more extensively in appendix B.2. Second, because there are 10 independent variables, a pure non-parametric approach that allows for flexible interaction between them would be difficult to implement.[47] Thus, I incorporate non-parametric approach only partially, to impose the desirable concavity restriction.

Additionally, I smoothly approximate the effect of ψ_{ct} at category-α^{rev}-bookmark-r level so that $\mathcal{W}$ changes smoothly as ψ_{ct} changes. In short, the baseline is the watch time predicted at the observed average (across months) of ψ_{ct}, denoted by $\bar{\psi}_c$. I assume that expected watch time increases (decreases) compared to this baseline when ψ_{ct} decreases (increases). This approximation helps the full equilibrium search algorithm

[45]To this end, I follow the methodology developed by Kuosmanen (2008) and Kuosmanen and Johnson (2010). For the approximation, the objective function is mean squared error, subject to the concavity.

[46]To tackle this issue, I also tried using an alternative, "hockey stick" (instead of quadratic) shaped cost function. However, then it is difficult to determine how cost shock γ_{jt} should perterbate costs. One should make those shocks affect hockey stick's threshold and slope, but then shock draw is not monotonic to action (hour choice). Without such monotonicity, one has to solve a large scale mixed interger programming (MIP) for each policy iteration loop, which makes policy iteration infeasible.

[47]To remind, 10 covariates are: bookmark, Best Broad caster status, dummy variables for three categories and three α^{rev} levels, streaming hour, and ψ_{ct}

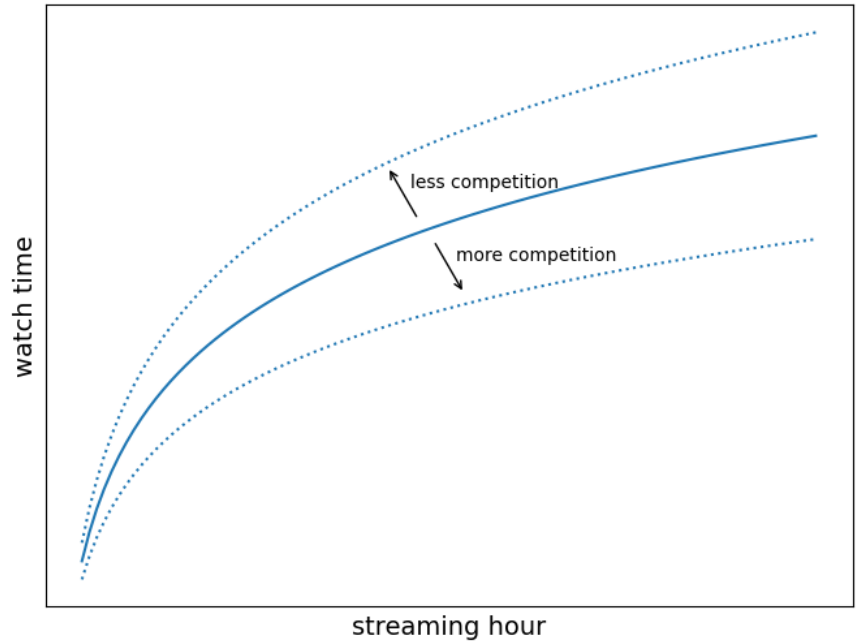


Figure 5. Visualization of watch time function $\mathcal{W}$ for one state.

Notes: Conditioning on a state x, watch time is increasing and concave with respect to streaming hours. Competition, measured by the category level weighted sum of streaming hours $\psi_{c(j)t}$, shifts $\mathcal{W}$.

converge well at the counterfactual stage.

The visualization of the fitted watch time $\mathcal{W}$ is shown in Figure 5. Conditional on the streamer's and market's state, a streamer's watch time is increasing and concave with respect to the streamer's streaming hours. When competition is more (less) intensive, meaning that the weighted sum of streamers' streaming hours at the broadcasting category level $\psi_{c(j)t}$ increases (decreases), the focal streamer's watch time decreases (increases). More technical details can be found in appendix B.3.

I utilize the across-month variation of ψ_{ct} to capture how an increase in ψ_{ct} decreases a streamer's watch time when fitting watch time $\mathcal{W}$. However, during the estimation stage, I assume that all observations are from a single steady-state equilibrium of the dynamic game, where all ψ_{ct} values are fixed at the observed average across the month. Therefore, this approximation does not have a significant impact at this step.

Transition Dynamics The transition dynamics of $r_{jt} \in \{0, 1\}$ was determined at subsection 4.3.2 in which approval probabilities by score were pinned down. For the remaining two states, $log(bookmark)_{jt}$ and $score_{jt}$, I use a variant of the linear probability model to capture progressions. As streamers stream more, they have a higher probability of starting with both a greater number of bookmarks and a higher score next period. To approximate the dynamics of either moving up or staying, a linear probability model is employed with one independent variable (streaming hours). Specifically, I assume the following linear probability model for moving up.

$$\mathbb{P}\left(log(bookmark)_{jt} \text{ or } score_{jt} \text{ goes up}\right) = \beta_0 + \beta_1 hour_{jt} + \varepsilon_{jt} \tag{5}$$

where $hour_{jt}$ represents the streaming hours of streamer j on month t. Conditional on remaining, there is a fixed probability of the streamer moving down one interval.[48] This fixed probability is determined by simply using the observed probability of moving down, conditioned on staying. Considering the heterogeneity across variable ranges, transition parameters are allowed to be different across subgroups of intervals. The results are summarized in Table 4.

This specification aims to make the marginal dynamic gain, resulting from a higher probability of starting at a better state tomorrow, concave with respect to streaming hours. An alternative like a logit model makes it convex.

4.4 Pseudo Maximum Likelihood Estimation

Given the previously approximated model components, I employ a full solution approach to estimate the structural parameters. For each parameter guess, I solve the dynamic programming problem of a streamer using policy iteration, which allows me to obtain the optimal policy function. Then I calculate the likelihood based on the predicted conditional choice probabilities of exit and streaming hours for streamer j in month t.

$$L_{jt} = \left[\widehat{Pr}(h_{jt}|x_{jt}) \cdot (1 - \hat{p}_{jt})\right]^{(1-1\{exit\}_{jt})} \cdot \hat{p}_{jt}^{1\{exit\}_{jt}}$$

Here, $\hat{p}$ represents the predicted exit probability, and $1\{exit\}_{jt}$ is a binary variable indicating the actual exit. By aggregating these likelihood values, I construct the log-likelihood and run the simplex algorithm to search for the set of parameters that maximizes the log-likelihood.

The identification comes from a number of variations. First, an individual streamer

[48] It is definitely possible to make the probability of moving down decrease as streamers stream more. However, because the probabilities are small, using the linear probability model again often yielded negative predicted probabilities. As a result, I decided to use a simple constant probability.

Table 4. $log(bookmark)_{jt}, score_{jt}$ transition dynamics estimates.

score range	$y = \mathbb{1}\{score\ up\}$		$\hat{\mathbb{P}}(down \mid stay)$
	$\hat{\beta}_0$	$\hat{\beta}_{100hours}$	
$0 - 60$	.0396	.0375	.0541
	(.0016)	(.0023)	
$60 - 66$	.1164	.1277	.1641
	(.0091)	(.0081)	
≥ 66	-	-	.0236

$log(bookmark)$ range	$y = \mathbb{1}\{bookmark\ up\}$		$\hat{\mathbb{P}}(down \mid stay)$
	$\hat{\beta}_0$	$\hat{\beta}_{100hours}$	
$0.0 - 5.0$	.0582	.1346	.0096
	(.0026)	(.0048)	
$5.0 - 8.5$	.0478	.0378	.0081
	(.0024)	(.0024)	
$8.5 - 11.0$	.0195	.0144	.0049
	(.0024)	(.0023)	
≥ 11	.0036	.0039	.0032
	(.0029)	(.0026)	

Notes: Streamers have the opportunity to either move up or remain in their current interval based on their streaming time. Conditioning on staying, there exists a probability of moving down. For readability, the unit of monthly streaming hours is set to 100. The numbers in parentheses are robust standard errors.

shows different actions around being approved to the rewards program and getting commission discount. This local variation helps identifying c_1, c_2, because the same individual faces time varying marginal gains from streaming. Second, streamers having different states exhibit different hour and exit decisions, so this across individual variation help identify all parameters $c_1, c_2, K_0, K_1, \sigma_c$.[49]

4.5 Estimates

The obtained estimates of $\{c_1, c_2, K_0, K_1, \sigma_c\}$ are presented in Table 5. There exists vast heterogeneity across and within broadcasting categories. First, within a category, streamers with high α_j^{rev} values appear to have higher streaming costs, which may result if they invest more effort into preparing their broadcasting content to impress viewers. Additionally, their superior ability would enable them to have higher outside options.

Second, the heterogeneity across categories could be explained by different psychological costs associated with running the broadcasting content. For example, while playing (video) games may be enjoyable, creating broadcasting content based on social interactions could be more nerve-wracking. This is because such streamers are usually involved in more emotional labor and care about maintaining a "good" image. When a Social streamer's relationship status is publicly revealed, it is often the case that some of their viewers leave.

While streamers with higher α^{rev} values may exhibit greater motivation due to the rewards program's 10% commission discount, their streaming costs could partially counterbalance this effect. In the counterfactual simulations, the cost parameters will determine which streamer types are more responsive to changes in rewards program design, crowding out other types. This shapes the composition change effects mentioned in the introduction.

4.6 Model Fit

This section investigates model fit through factual simulation and discusses the sources of discrepancies between model predictions and observed data. I first conduct a factual simulation using the full equilibrium search algorithm described in section 5.2.

[49]Streamer heterogeneity that is not captured by my model may disrupt this argument. However, I explicitly control observed state variables, bookmark and score, and in addition control per watch time revenue α_j^{rev} which could capture some unobserved characteristics like communication skill to mitigate this concern.

Table 5. Pseudo Maximum Likelihood Estimates.

category	α^{rev}	c_1	c_2	K_0	K_1	σ_c
Game	low	0.290	0.426	85.313	19.291	1.537
		(0.003)	(0.004)	(0.925)	(0.295)	(0.018)
	med	0.011	0.424	51.011	29.371	1.269
		(0.000)	(0.001)	(0.366)	(0.071)	(0.006)
	high	8.159	1.186	245.709	108.628	1.206
		(0.059)	(0.008)	(2.966)	(0.486)	(0.015)
Social	low	0.117	1.919	660.712	31.758	1.726
		(0.002)	(0.044)	(14.306)	(0.436)	(0.031)
	med	0.018	5.841	643.550	223.352	1.327
		(0.000)	(0.038)	(4.029)	(1.835)	(0.006)
	high	1.178	10.282	801.125	565.757	1.039
		(0.003)	(0.011)	(16.358)	(1.667)	(0.001)
Other	low	0.013	0.015	11.328	0.312	1.695
		(0.000)	(0.000)	(0.159)	(0.003)	(0.018)
	med	0.039	0.304	60.946	12.888	1.536
		(0.001)	(0.003)	(0.066)	(0.151)	(0.004)
	high	0.014	1.929	240.851	110.615	1.138
		(0.000)	(0.001)	(0.763)	(0.074)	(0.001)

Notes: The numbers in parentheses represent bootstrap standard errors ($B = 20$). The unit of model parameters is approximately 1 USD, except for K_0 (1,000 USD for readibility) and σ_c (unitless).

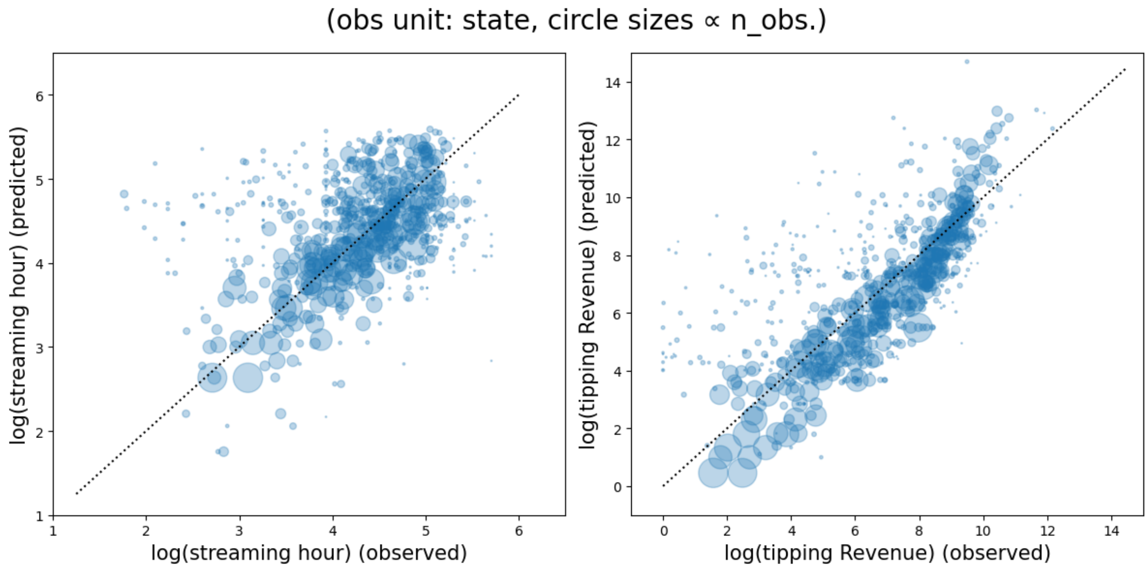


Figure 6. Predicted versus observed streaming hours and tipping revenue.

Notes: The unit of observation is streamer's state, i.e. combination of category, profitability, bookmark, score, and Best Broadcaster status. Each value is averaged at state level, and the size of each circle is proportional to the number of observations.

That is, the given environment including approval probabilities by a score remains unchanged, but only ψ_{ct} is endogenized. During the parameter estimation stage, ψ_{ct} was fixed at the observed average level to reduce the computational burden.

This factual simulation reveals how well my framework approximates the data. To evaluate the model fit, I compare predicted and observed counterparts in terms of streaming hour choices and generated tipping revenue. Figure 6 shows that the model predictions closely match the observed values, which confirms that the model does a reasonable job explaining streamers' behavior.

Further assessments of model fit are provided in Appendices C.1, C.2, and C.3, in addition to Figure 6. Specifically, I compare predicted and observed industry-level variables such as the distribution of streamers' bookmarks, evaluate whether the reward cutoff moved similarly to my model's prediction following AfreecaTV's design change implemented outside my data period, and discuss factors that may have contributed to the remaining gaps.

In the next section, I conduct counterfactual simulations to evaluate alternative rewards program design. When doing so, the counterfactual outcomes are always compared with factual simulation outcomes.

5 Counterfactual Simuations

5.1 Overview

I conduct counterfactual simulations to assess design changes of the rewards program. Such simulations use the estimated model parameters and full equilibrium search algorithm. First, the number of monthly approval slots is doubled and halved. Second, the existing 10% commission discount given to streamers who have been approved for the rewards program is now changed to a 5% and a 15% commission discount. Lastly, approval slots are reallocated at the category level, giving two more slots to the Social category and two fewer slots to the Game category.

The purpose of these counterfactual exercises is to study if there is room for platform net revenue improvement through changes in the rewards program design. The first two simulations explore how changes in the number of approval slots and commission discount benefits may raise platform revenue. The third simulation investigates if more granular program design, assigning rewards at category level instead of platform level, could raise platform revenue.[50]

From the perspective of platform revenue, the overall effect of the design changes can be broken down into three factors. Note that, platform revenue is a product of three terms: total watch time, average revenue per watch time, and the platform's share of the total revenue. Therefore, the overall effect on platform revenue can be decomposed into the product of changes in these three terms.

To summarize, I find that reducing the number of monthly approval slots, or the amount of commission discount seems to be a better option for raising the platform revenue. Providing more rewards motivates streamers—especially more profitable ones—to stream more. It thus increases overall watch time (watch time effect) and improves average profitability measures by per watch time revenue (profitability effect). However, the impact on revenue from a decrease in a platform's share of the tipping revenue due to more commission discount provision (platform share effect) generally outweigh the impact from other two effects.

To be more precise, when the platform provides more rewards, whether the watch

[50] Additionally, this third simulation is useful for identifying design changes that could improve platform revenue without negatively affecting overall streamers. While my simulation results support the idea of reducing the program, including rescinding benefits from already approved streamers, it could be challenging to implement in practice. Bewley (1998) shows that, while in theory, a pay cut could be a reasonable response for firms to changes in the economic environment, it is rarely used in practice because managers believe it may harm employee morale.

time effect on platform revenue is positive or negative is an empirical question. To illustrate this, consider a scenario where the platform offers higher commission discounts as a reward. On one hand, streamers' potential future benefits, conditional on strong performance, improve. This leads to increased streaming hours and consequently more watch time. On the other hand, streamers do not take into account the cannibalization effect they have on other streamers, potentially resulting in a decrease in total watch time.

Similarly, it is unclear whether the profitability effect resulting from the watch time composition of different streamers always raises platform revenue when the platform provides more rewards. On one hand, commission discounts are more appealing to highly profitable streamers, as they earn more tipping revenue. Thus, they could be motivated disproportionately and occupy a larger portion of watch time, improving the average profitability. However, the estimates indicate that highly profitable streamers have higher effort costs, so their share of watch time may not increase in response to more rewards.

That being said, I find that the watch time effect and profitability effect work as expected, in the sense that both effects contribute to increasing total revenue when the reward is provided to more streamers or when a greater amount of commission discount is offered. However, the aforementioned factors are still useful for understanding why revenue improvements from these two effects are not very strong.

When searching for a full equilibrium under different environments—the number of reward program slots and benefit (commission discount %)—the main operational challenge is that it is unclear how the tournament design can be directly mapped to a streamer's dynamic programming. My analysis takes an indirect approach. The approval probabilities by score, as described in section 4.3.2, are changed until I get a desired outcome at steady state. For example, to double the number of slots, approval probabilities are increased until the number of streamers getting the reward doubles at steady state, as compared to the number attained from factual simulation. I first describe the details of my full equilibrium search algorithm and then proceed to the results description.

5.2 Full Equilibrium Search

During the estimation step, I assumed the category-level weighted sums of hours, denoted as ψ_{ct}, which capture streamer interactions, were fixed at their observed averages to reduce computational burden. To conduct counterfactual simulations, it is essential to endogenize these sums and search for a full equilibrium consistent with the definition in section 3.3. To do so, I first determine the entry process, which specifies how many new streamers enter each period at the state level.

Entry Process I use the empirical entry process observed in the data. The data begin from October 2019, but one streamer tracking website started monitoring streamers in January 2019. Therefore, I can identify streamers who newly appear each month. For example, I can observe 1.29 streamers having the lowest level of bookmarks and score, non-Best Broadcaster status, Social category and low profitability enter the market each month, on average.

This entry process is assumed to remain unchanged throughout the counterfactual simulations. While the entry process itself can respond to reward program designs, it would have a limited impact on simulation outcomes, given that many entrants are not popular streamers with little chance of receiving the reward.

Next, I describe an algorithm to search for a full equilibrium when specific rewards program design is given, i.e., the number of approval slots and commission discount percentage. In the outer loop, I change approval probabilities by score. In the inner loop, I search for the corresponding equilibrium and industry state. I iterate the outer loop until the number of streamers newly approved at equilibrium coincides with the given number of slots provided. The commission discount percentage remains fixed throughout the algorithm, but it may be fixed at 5 or 15 percent instead of 10 percent (status quo) to evaluate alternative program designs.

Note that to find an equilibrium in the inner loop, it suffices to find category-level, weighted sums of streaming hours at equilibrium, which is denoted as ψ_c^*. Once the weighted sums are given, the only remaining step is solving a streamer's single-agent dynamic programming problem.

Lastly, I clarify some notations used in the algorithm described below. N_{slot} represents the desired number of slots, $\hat{N}_{slot}$ indicates the predicted number of streamers approved at each equilibrium, and N_{slot}^{gap} represents the difference between these two values. $\{\psi_c\}$ denotes the category-level weighted sum of hours.

In the context of the algorithm, the term "marginal" score group refers to the highest (lowest) score group for which the approval probability is less than 1 (greater than 0) when increasing (decreasing) approval probabilities. It is implicitly assumed that changes in the number of slots primarily affect the probability of the highest/lowest score group.

Algorithm Full Equilibrium Search

(Note: "marginal" score group means highest (lowest) score group for which the approval probability is less than 1 (greater than 0) when increasing (decreasing) approval probabilities.)

Specify commission discount percentage.
Specify the number of slots N_{slot}.
Initial guess of approval probabilities by score
$N_{slot}^{gap}, \psi_c^{gap} \leftarrow 1$
while $N_{slot}^{gap} > \epsilon^N$ **do**
    Increase or decrease "marginal" (see above) score group's approval probability
    Initial guess of $\{\psi_c^*\} = \{\psi_c^{old}\}$ (weighted sum of hours)
    $\lambda \leftarrow 1$
    while $\psi_c^{gap} > \epsilon^{\psi}$ for any c **do**
        Given $\{\psi_c^{old}\}$, solve a streamer's DP and obtain optimal polices.
        Compute the corresponding industry state and ψ_c^{new} (cf. equation (4))
        $\psi_c^{gap} \leftarrow \|\psi_c^{new} - \psi_c^{old}\|_1$
        $\psi_c^{old} \leftarrow \psi_c^{old} + \frac{1}{\sqrt{\lambda}} \cdot (\psi_c^{old} - \psi_c^{new})$; $\lambda \leftarrow \lambda + 1$
    end while
    From equilibrium industry state, compute $\hat{N}_{slot}$
    $N_{slot}^{gap} \leftarrow \|N_{slot}^{gap} - \hat{N}_{slot}\|_1$
end while

Before concluding this subsection, I clarify two limitations of the full equilibrium search algorithm and the implementation of counterfactuals. First, the algorithm indirectly assesses tournament design changes only through changes in approval probabilities that are based on score. This point was extensively discussed in section 3.4.

Second, the probability of applying for the rewards program ($\mathbb{P}(\mathcal{I}_{jt} = 1)$) is assumed to remain unchanged. While approval probabilities can increase in the outer loop, this assumption serves as a cap on the ex-ante approval probability.

5.3 Counterfactual 1: Changing the Number of Approval Slots

I first examine a counterfactual platform policy that involves doubling and halving the monthly approval slots. In other words, the number of streamers who receive the permanent commission discount reward doubles or halves. Overall, simulation results show that the former decreases platform revenue by 2.53%, while the latter increases it by 3.03%. Offering the reward to more streamers leads to an increase in both total watch time and the share of profitable streamers, but the resulting decline in the platform's share on total revenue is significant enough to result in an overall negative effect.

The detailed breakdown of these effects can be found in Table 6. Computing the total watch time effect is straightforward. I simply compare the amounts of watch time before and after the design change is implemented. To compute the profitability effect, I first calculate the total revenue based on the changed watch time, assuming that the share of each category-α^{rev} group remains the same as in the factual simulation. This value is then compared with the total revenue of the counterfactual simulation outcome.

Similarly, to isolate the platform share effect, the platform's revenue is computed based on the changed total revenue, assuming that the share of both the platform and streamers remains unchanged. This result is then compared with the outcome of the counterfactual simulation to compute this effect. Table 7 shows the specific composition changes and changes in the platform/streamers' share in each counterfactual simulation.

Table 6. Decomposition of the effect of an alternative design on platform revenue.

(Units: %, relative to the current program design)

Counterfactual	Watch time effect	Profitability effect	Platform share effect	Overall effect
1. The number of streamers getting the reward each period				
doubled	0.22	0.94	-3.66	-2.53
halved	-0.75	-0.08	3.90	3.03
2. Benefit change (current commission discount: 10%)				
increase to 15%	0.07	0.34	-6.63	-6.24
reduce to 5%	-0.10	-0.32	6.52	6.08
3. More granular (category level) slot reallocation				
Social to Game	0.02	-0.45	1.62	1.19

Notes: Watch time, profitability, and 'platform share effects refer to the platform's revenue gain resulting from changes in total watch time, average per watch time tipping revenue, and the platform's share among tipping revenue, respectively..

Table 7. Detailed streamer decomposition (above) and platform's share change (below).

(Panel A: Composition Effect Details)

category	α^{rev}	Watch Time Share Compositions (%)					
		Status Quo	Doubling Slots	Halving Slots	5%p more commission cut	5%p less commission cut	Slot Reallocation
Game	low (0.017)	23.27	23.07	23.18	23.35	23.16	23.23
	med (0.150)	20.83	21.20	20.48	20.95	20.72	20.92
	high (0.698)	16.87	17.01	16.84	16.95	16.79	16.91
Social	low (0.239)	1.46	1.44	1.48	1.45	1.47	1.46
	med (1.591)	3.94	3.98	3.96	3.94	3.95	3.91
	high (5.192)	7.66	7.72	7.67	7.68	7.64	7.60
Other	low (0.012)	11.00	10.45	11.48	10.66	11.34	11.00
	med (0.205)	6.98	7.01	6.99	6.98	6.99	6.98
	high (1.573)	7.99	8.11	7.93	8.04	7.94	7.99

(Panel B: Platform Share Change Details. All numbers are percentage)

Counterfactual	Status Quo	Doubling Slots	Halving slots	5%p more commission cut	5%p less commission cut	Slot Reallocation
Streamer's share	64.59	65.88	63.21	66.94	62.28	64.12
Platform's share	35.41	34.12	36.79	33.06	37.72	35.88
Total	100.00	100.00	100.00	100.00	100.00	100.00

Notes: The numbers in parentheses represent discretized values of α^{rev}. The unit of measurement is approximately 1 USD per watch time.

5.4 Counterfactual 2: Changing Rewards Program Benefit

Second, I examine a counterfactual platform policy that offers either a 5% or a 15% commission discount (instead of the existing 10%) whenever a streamer gets approved. Using the same decomposition procedure described above, I compute the overall effect and present the results accordingly.

Platform revenue increases by 6.08% when the commission benefit is reduced. Watch time and composition effects decrease platform revenue in response to the reduced approval benefit, but the platform share change effect dominates these two opposing effects. These factors go exactly opposite when the commission discount benefit increases to 15%, which results in a 6.24% decrease in platform revenue.

5.5 Counterfactual 3: Reallocation of slots at Category Level

Lastly, I investigate whether, even without unilaterally reducing the overall program, the platform revenue can be improved by adopting more granular program design. This perspective focused on the fact that the current program determines the number of slots available at the platform level, while the effect of giving one more slot can differ across broadcasting categories.

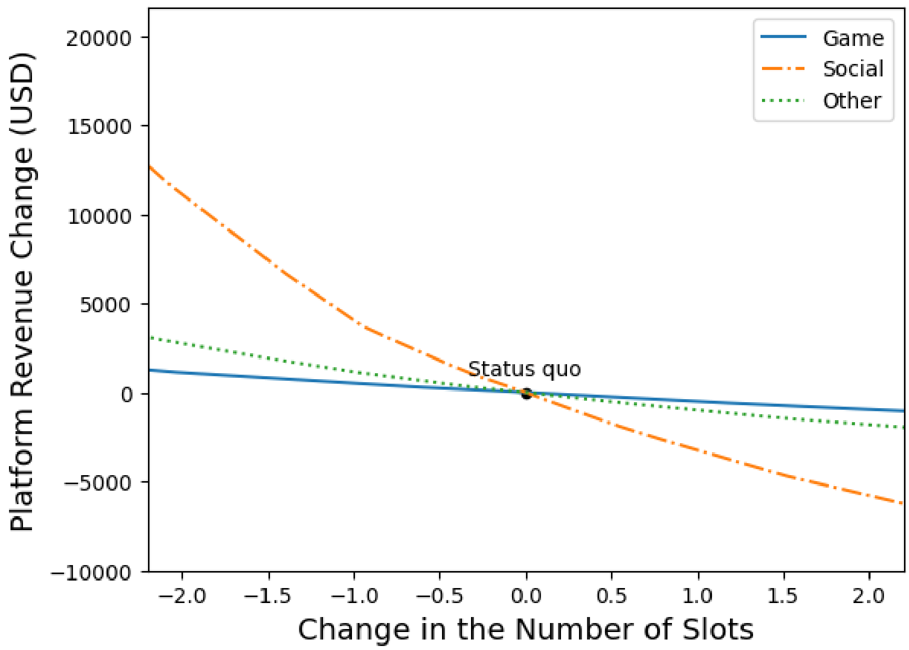
It is straightforward to apply the full equilibrium search algorithm and change the number of slots at the category level because there are no across-category interactions once the approval probability by score is given. I run the algorithm at the category level and plot how platform revenue changes when allocating one more or fewer slots to a category compared to the status quo.

Figure 7 shows these results. The main takeaway is that the marginal change in platform revenue is different across categories. While platform revenue decreases in all categories when offering more slots, the slopes of the all revenue changes are all different. Therefore, platform revenue can be improved by decreasing the number of slots in the Social category (where the loss from giving rewards program benefits is the greatest) and increasing slots for other categories.

I conduct this simulation by reducing two slots for the Social category and adding two more for the Game category. The number of slots in the Other category remains the same. In the status quo, there are 12.19, 4.58, and 4.59 streamers newly approved each period in the Game, Social, and Other categories, respectively. After the slot reallocation, the numbers change to 14.19, 2.58, and 4.59, respectively.

It turns out that this reallocation raises platform revenue by 1.19%. Note that

Figure 7. Platform revenue changes when allocating more or fewer slots to specific categories. The black dot in the middle represents the status quo.



streamers in the Social category tend to have higher per watch time revenue. Thus, providing them with a smaller commission discount results in substantially increasing the program expenses arising from commission discounts, which drives these results. Again, the results are summarized in Table 6 and Table 7.

Lastly, one potential concern is whether streamers can switch their categories in response to this slot reallocation. However, such switching behavior is not likely to happen frequently because the human capital needed for each category is difficult to acquire. For example, to be successful in the Social category, a streamer typically has to be physically attractive, have eloquent speaking skills, or both. In the Game category, viewers are usually fans of a specific video game, so a deep understanding of that video game is required. Moreover, switching categories could lead to substantial loss of existing viewers (who enjoy the current content) and cause them to leave. Thus, it is difficult for streamers to switch their categories freely.

6 Conclusion

I have developed an empirical dynamic model to assess the impact of seller rewards programs on the revenue of digital platforms, within the empirical context of livestreaming platforms. The design of rewards programs influences platform revenue through changes in three key factors. First, it affects the working hours of the average streamer (a seller in this industry), and consequently, the total market size (watch time). Second, it changes the watch time share of streamers who can extract more or less tipping revenue from the same amount of watch time. This change affects the average profitability measured by per watch time tipping revenue. Lastly, it impacts the platform's share among the generated tipping revenue, as the platform offers a permanent commission discount as the reward program benefit.

Counterfactual simulations suggest that the last platform share effect quantitatively dominates as an influence over net platform revenues. This implies that reducing the rewards program by offering the reward to the smaller number of streamers per month, or decreasing the amount of commission discount given as the reward, would raise net platform revenue. In case unilateral reduction is practically difficult, I additionally show that reallocating program approval slots at more granular level, i.e. at the broadcasting category level instead of the platform level, may still improve the net platform revenue.

Some limitations of this methodology may offer directions for future research. First, the current model assumes streamers draw iid shocks to decide whether to apply to the rewards program or not, but having unobserved preferences about the application correlated over time would be more realistic. Second, my model does not account for competition between platforms and streamer multihoming.[51] Lastly, my model lacks an endogenous entry process. Ideally, incorporating these features and extending the model could open up promising avenues for further research.

[51] One observation that could support this simplification is that popular contents in the focal platform (AfreecaTV) and its rival platform (Twitch) are quite different, so they could be considered two separate markets.

References

V. Aguirregabiria, A. Collard-Wexler, and S. P. Ryan. Dynamic games in empirical industrial organization. In *Handbook of Industrial Organization*, volume 4, pages 225–343. Elsevier, 2021.

C. Belzil and M. Bognanno. Promotions, demotions, halo effects, and the earnings dynamics of american executives. *Journal of labor Economics*, 26(2):287–310, 2008.

T. F. Bewley. Why not cut pay? *European Economic Review*, 42(3-5):459–490, 1998.

K. Bimpikis, S. Ehsani, and M. Mostagir. Designing dynamic contests. *Operations Research*, 67(2):339–356, 2019.

Y. Chen and D. Xu. A structural empirical model of r&d investment, firm heterogeneity, and industry evolution. *Journal of Industrial Economics*, 2022.

D. J. Chung, T. Steenburgh, and K. Sudhir. Do bonuses enhance sales productivity? a dynamic structural analysis of bonus-based compensation plans. *Marketing Science*, 33(2):165–187, 2014.

D. J. Chung, B. Kim, and B. G. Park. The comprehensive effects of sales force management: A dynamic structural analysis of selection, compensation, and training. *Management Science*, 67(11):7046–7074, 2021.

Ø. Daljord, S. Misra, and H. S. Nair. Homogeneous contracts for heterogeneous agents: Aligning sales force composition and compensation. *Journal of Marketing Research*, 53(2):161–182, 2016.

J. DeVaro and H. Morita. Internal promotion and external recruitment: A theoretical and empirical analysis. *Journal of Labor Economics*, 31(2):227–269, 2013.

R. Ericson and A. Pakes. Markov-perfect industry dynamics: A framework for empirical work. *The Review of economic studies*, 62(1):53–82, 1995.

T. Eriksson. Executive compensation and tournament theory: Empirical tests on danish data. *Journal of labor Economics*, 17(2):262–280, 1999.

Y. Huang and I. Morozov. Video advertising by twitch influencers. *Available at SSRN 4065064*, 2022.

B. Ifrach and G. Y. Weintraub. A framework for dynamic oligopoly in concentrated industries. *The Review of Economic Studies*, 84(3):1106–1150, 2017.

T. Kuosmanen. Representation theorem for convex nonparametric least squares. *The Econometrics Journal*, 11(2):308–325, 2008.

T. Kuosmanen and A. L. Johnson. Data envelopment analysis as nonparametric least-squares regression. *Operations Research*, 58(1):149–160, 2010.

E. P. Lazear. Compensation and incentives in the workplace. *Journal of Economic Perspectives*, 32(3):195–214, 2018.

J. Lemus and G. Marshall. Dynamic tournament design: Evidence from prediction contests. *Journal of Political Economy*, 129(2):383–420, 2021.

S. Lu, D. Yao, X. Chen, and R. Grewal. Do larger audiences generate greater revenues under pay what you want? evidence from a live streaming platform. *Marketing Science*, 40(5):964–984, 2021.

S. Misra and H. S. Nair. A structural model of sales-force compensation dynamics: Estimation and field implementation. *Quantitative Marketing and Economics*, 9: 211–257, 2011.

M. Mostagir, Y. Chen, and Y. Yeckehzaare. Information provision in dynamic contests: An experimental study. Technical report, Working paper, University of Michigan, Ann Arbor, 2019.

A. Simonov, R. Ursu, and C. Zheng. Do suspense and surprise drive entertainment demand? evidence from twitch.tv. *Columbia Business School Research Paper*, 2021.

J. Tudón. Prioritization vs. neutrality on platforms: Evidence from amazon's twitch. tv. *Rand Journal of Economics*, (forthcoming), 2021.

G. Y. Weintraub, C. L. Benkard, and B. Van Roy. Markov perfect industry dynamics with many firms. *Econometrica*, 76(6):1375–1411, 2008.

G. Y. Weintraub, C. L. Benkard, and B. Van Roy. Computational methods for oblivious equilibrium. *Operations research*, 58(4-part-2):1247–1265, 2010.

Online Appendix for
"A Structural Model of Rewards Programs on Digital Platforms: The Case of Livestreaming"

Sueyoul Kim
Korea Development Institute (sueyoul.econ@gmail.com)
February 15, 2026

A Data Construction Details

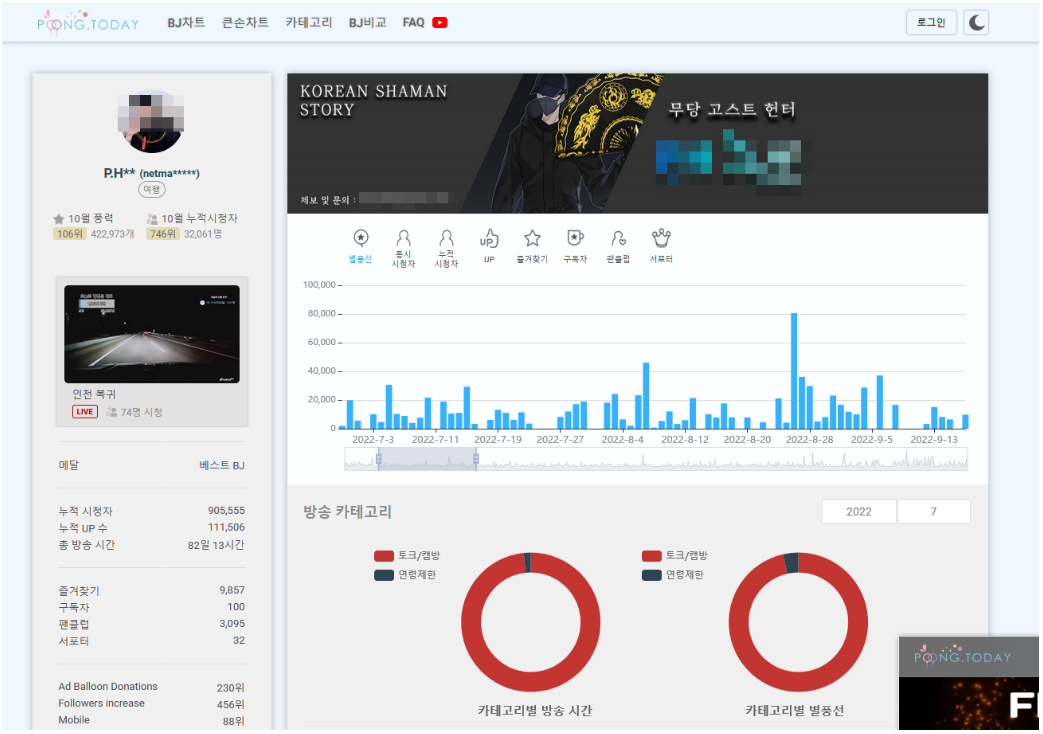
The main data set comes from three sources. From streamer tracking website 1, I collect individual streamer-day level revenue and streamer-month-category level streaming hours. Figure A.8 provides a snapshot of tracking website 1. From streamer tracking website 2, I collect streamer-day level streaming hours, bookmarks, and watch time. Lastly, from the platform's official website, I collect the list of streamers who got accepted to the rewards program each month, and conversion table to compute a streamer's score.

Data from the period October 2019 to April 2020 is used because the website 2 started recording revenue generation from October 2019, and the number of monthly approval slots doubled in April 2020. As a result, the fraction of approved streamers increased over time from May 2020, which makes it difficult to fit the observed pattern in the steady-state framework.

I merge two data sets from website 1 and 2 based on the streamer's account name and day. Some observations are dropped because the two websites do not track the exact same set of streamers. However, the merged data still covers 90.5 percent of generated revenue. The merged data was then collapsed into the individual streamer-month level.

Next, I add the streamer's score and a dummy variable that indicates if a streamer has already been approved, based on the information collected from the platform's official website. Streamers' scores are computed based on the platform's official conversion table (See Table A.8 below). The dummy variable can be constructed from the monthly approval list.

Figure A.8. A snapshot of tracking website 1 for a single streamer, where the blue bars depict the daily tipping received.



Subsequently, streamers who always broadcast more than 360 hours per month were dropped. These streamers account for 0.89 percent of observations and they are not individuals but firms like television news channels. In addition, observations at the streamer month level were dropped if a streamer always exhibits zero streaming hours after a certain period. This event is regarded as an exit. Plus, observations were if streaming hour, score, or bookmark data was not available. These latter two steps drop 13.52 and 14.02 percent of observations, respectively. Finally, I construct an exit flag dummy variable, which takes the value of 1 if a streamer does not appear in the next month.

Table A.8. The score conversion table on AfreecaTV.

score	average viewership (40%)	bookmarks (40%)	total broadcasting hours (20%)
100	(more than) 1000	50,000	10,000
98	750	40,000	9,500
96	500	30,000	9,000
94	450	20,000	8,500
92	400	10,000	8,000
90	350	9,000	7,500
88	300	8,000	7,000
86	250	7,000	6,500
84	200	6,000	6,000
82	160	5,600	5,500
80	130	5,200	5,000
78	100	4,800	4,500
76	90	4,400	4,000
74	80	4,000	3,500
72	70	3,600	3,000
70	60	3,200	2,600
68	50	2,800	2,200
66	40	2,400	1,800
64	30	2,000	1,400
62	20	1,500	1,000
60	10	1,000	500

Notes: The variable average viewership is computed based on a 3-month window. Bookmarks represent the number of viewers who have bookmarked the streamer. The score is a weighted sum of scores from three items. For example, a streamer with an average viewership of 320, 5,000 bookmarks, and a total streaming duration of 2,400 hours would have a score of $0.4 \cdot 88 + 0.4 \cdot 80 + 0.2 \cdot 68 = 80.8$. Below the bottom row, the score for each item is zero. The source of this table is https://afevent2.afreecatv.com/app/star_bj/bestbj/order_info.php, retrieved on July 15, 2023.

B Computation Details

B.1 Discretization of Continuous Variables

As mentioned earlier, I divide streamers in each broadcasting category into three groups (low, medium, high) based on their individual level α_j^{rev}. Figure B.9 displays the distribution of α_j^{rev} in each broadcasting category and the corresponding cutoffs. For readability, I apply a log transformation.

Next, I discretize the state variables that evolve over time. The binary variable $r_{jt} = \mathbb{1}\{Best\ Broadcaster\}_{jt}$ clearly needs not to be discretized further.I discretize the using the following cutoffs: $[4.0, 4.5, \cdots, 12.5, 13.0]$. Lastly, I discretize the streamer *score* ranging from 0 to 100 using the following cutoffs: $[20, 40, 60, 61, 62, 63, 64, 65, 66]$. This decision is based on approval cutoffs during the 7-month data period were observed to be around 63.5-65. Therefore, these scores are discretized more granularly around the cutoff value. Below 60, the approval probability is essentially zero, but they were discretized sparsely to capture the rate of streamer growth. Finally, for scores above 66, streamers are all highly likely to get approved if they apply.

B.2 Computational Usefulness of Concavity of Watch Time

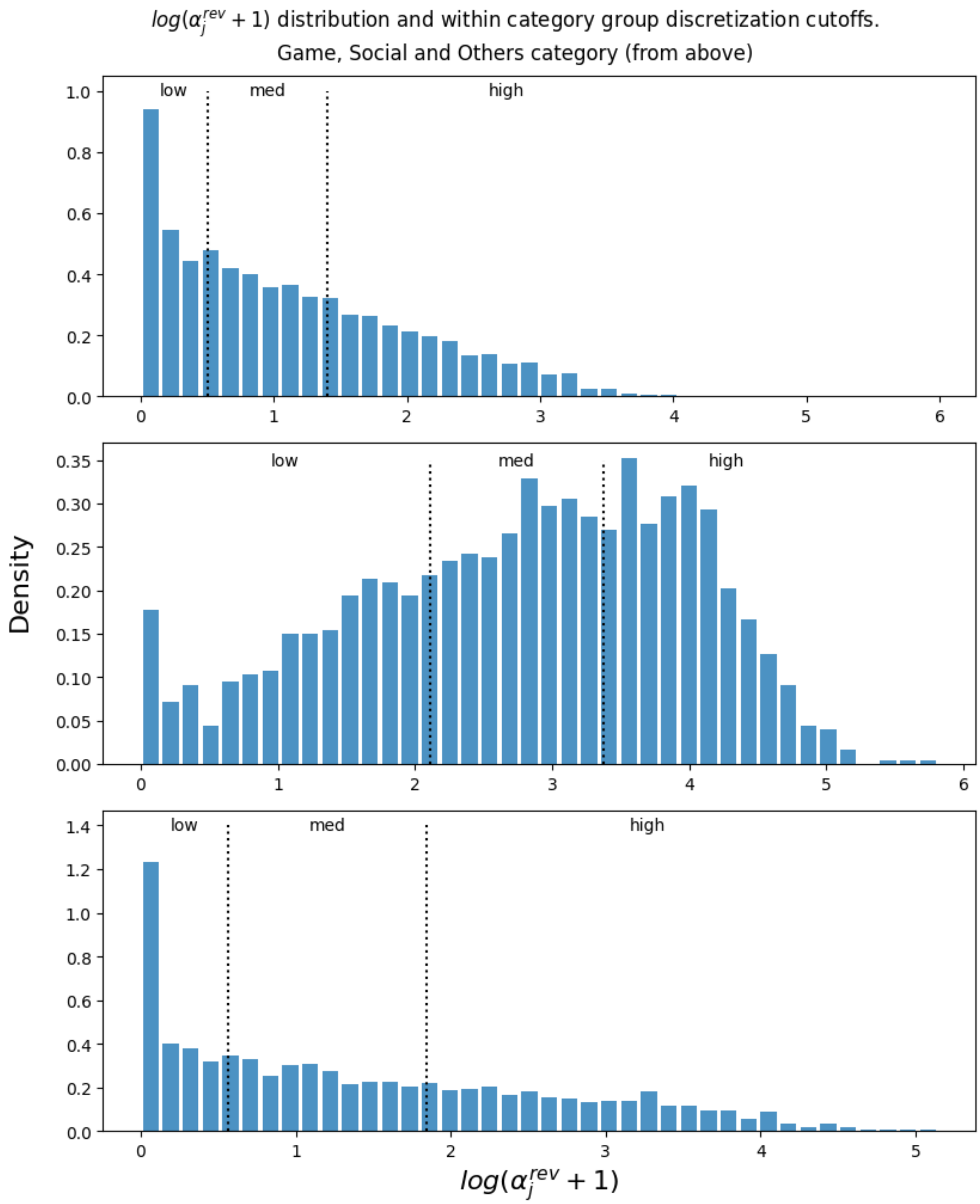
To ensure a finite log-likelihood (LL), it is essential to ensure that the model can generate all possible actions. Specifically, the cost shock draws γ_{jt} must result in all streaming hour choices being selected with positive probability. For this, it is very helpful to impose that the marginal revenue from increasing streaming hours is non-increasing.

If this condition is not met, as depicted in Figure B.10, certain hour choices will be strictly dominated by other options and will not be chosen in the model. Consequently, the model assigns zero probability to some observed hour choices, causing my log-likelihood to diverge to negative infinity.

B.3 Watch Time (XGBoost) Details

Model Selection I first investigate the best performing machine learning (ML) tool for making proper watch time predictions. To do so, I utilize `AutoGluon`, a library developed by Amazon that automates the application and comparison of multiple ML tools for a data set. With `AutoGluon`, I evaluate the out-of-sample predictive accuracy of various ML models, including tree-based models (Random Forest, Extra

Figure B.9. Per watch time revenue α_j^{rev} distributions within each category



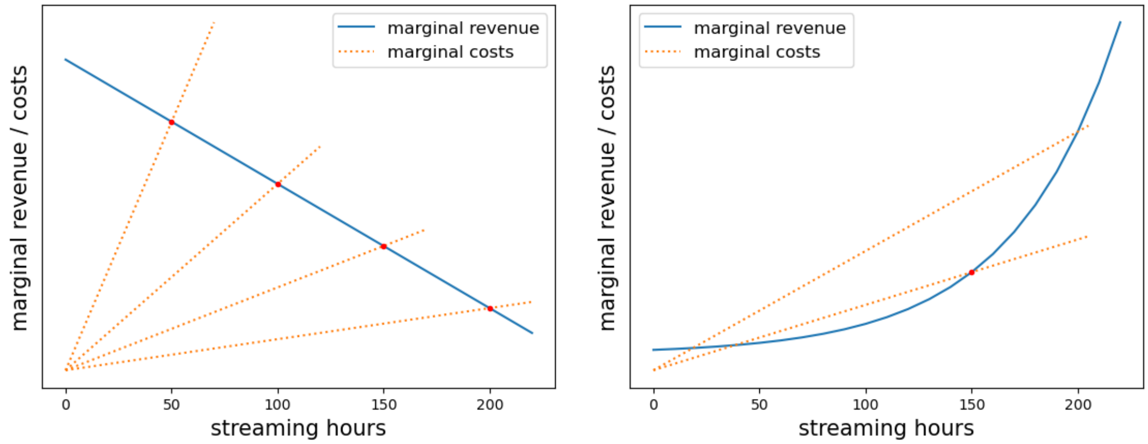


Figure B.10. When marginal revenue decrease with respect to streaming hours, multiplicative cost shocks can lead a streamer to choose any number of hours (left). However, when the marginal revenue is increasing, certain options, like 150 hours, are strictly dominated and are not chosen for any shock draws (right).

Trees, LightGBM, CatBoost, and XGBoost), neural network models, and K nearest neighbor. XGBoost was chosen after comparing their out-of-sample prediction performances.

Specifically, I compare the aforementioned models with four specifications: using the first or last month as a test set and using out-of-sample root mean squared error (RMSE) or mean absolute error (MAE) as a performance metric. For each specification, I compute the relative loss of each model. The relative loss is defined by how much worse a model's performance is compared to the model that showed the best out-of-sample performance. For example, if XGBoost shows the best performance of 10,000 out-of-sample MAE, and the NeuralNetTorch shows 11,000, the two models have a relative loss of 0.0 and 0.1, respectively, for the specification. These results appear in Table B.9.

Hyperparameters I determine hyperparameters with 7-fold cross validation as there are seven month data period. The chosen hyperparametes included: the number of estimators (trees) 20, max depth 15, learning rate (η) 0.1, minimum child weight 1, subsample ratio 0.8. $log(bookmark)$ turned out to be the most "important" determinant, from perspective of how many times a variable was used to split nodes, followed by streaming hour and ψ_c.

Table B.9. Comparison of machine learning models through the `AutoGluon` library.

Model	(1)	(2)	(3)	(4)	Mean Relative Loss
XGBoost	0.043	0.105	0.006	0.000	0.038
NeuralNetTorch	0.064	0.032	0.074	0.007	0.044
LightGBM	0.014	0.469	0.004	0.253	0.185
ExtraTreesMSE	0.017	0.481	0.007	0.283	0.197
CatBoost	0.000	0.507	0.009	0.308	0.206
LightGBMLarge	0.036	1.696	0.006	0.340	0.520
LightGBMXT	0.069	1.523	0.036	0.523	0.538
RandomForestMSE	0.125	1.569	0.012	0.526	0.558
NeuralNetFastAI	0.067	8.689	0.000	0.121	2.219
KNeighborsUnif	1.371	20.474	0.081	0.166	5.523
KNeighborsDist	1.371	20.474	0.081	0.166	5.523
Test set	First month		Last Month		
Performance Metric	RMSE	MAE	RMSE	MAE	

Notes: The numbers in the cells represent the relative loss, which is defined as the relative difference in out-of-sample RMSE/MAE compared to the best-performing model in each specification. The best model in each specification, therefore, has a relative loss of zero. In specification (2), there is no model with a loss of zero because the weighted ensemble model, not included in this table, performed the best.

Shape and Smoothness Restrictions I additionally address two issues. First, XGBoost's raw watch time predictions may not exhibit concavity (with respect to streaming hours), which is essential for the reasons outlined in Appendix B.2.

Second, because XGboost uses step functions as its basis, watch time often remains flat even when independent variables change. This characteristic isn't problematic for independent variables like bookmarks, but it poses a challenge for the category-level weighted sum of streaming hours, denoted as ψ_{ct}. The issue arises in the outer loop of the full equilibrium search algorithm in section 5.2. In this algorithm, I search for equilibrium by adjusting ψ_{ct}. If watch time remains flat concerning ψ_{ct}, the algorithm may fail to converge. [1] To resolve these issues, I proceed as follows.

1. Train XGBoost model. The observation level is individual streamer-month. The dependent variable is watch time, and independent variables are bookmarks,

[1] More specifically, without imposing some smoothness with respect to ψ_{ct}, I found that the algorithm often results in the following infinite loop: streamers initially believe that ψ_{ct} is small (indicating less competition), causing them to stream more, resulting in a realized ψ_{ct} that is large. Streamers then update their belief based on this realization, leading them to believe that ψ_{ct} is large (indicating more competition), and subsequently, they stream less, resulting in a realized ψ_{ct} that is small again. As a result, ψ_{ct} can oscillate between small and large values without converging.

streaming hours, broadcasting category, profitability (per watch time tipping revenue), and ψ_{ct}.[2] Note that, at this point, ψ_{ct} varies across months, and I impose watch time to decrease when ψ_{ct} increases.

2. Obtain watch time predictions at the state level while holding ψ_{ct} constant at the monthly average $\bar{\psi}_c$. Denote these watch time predictions as $\widehat{\mathcal{W}}(x, h, \bar{\psi}_c)$, where x and h represent a state, i.e., a combination of category, profitability, and bookmarks, and streaming hours, respectively.

3. To impose concavity with respect to streaming hours, begin by collecting watch time predictions at the $x, \bar{\psi}_c$ level, i.e., $\{\widehat{\mathcal{W}}(x, h=0, \bar{\psi}_c), \widehat{\mathcal{W}}(x, h=50, \bar{\psi}_c), \ldots, \widehat{\mathcal{W}}(x, h=300, \bar{\psi}_c)\}$. Next, approximate these predictions using non-parametric concave functions.[3] Denote this approximated concave watch time by $\widehat{\mathcal{W}}^{conc}(x, h, \bar{\psi}_c)$.

4. Impose that watch time changes smoothly with respect to ψ_{ct}. To determine the degree of change, I utilize the across-month variation of predicted (raw) watch times, denoted as $\widehat{\mathcal{W}}(x, h, \psi_{ct})$. Specifically, for each state x, I estimate coefficients θ_0 and θ_1 in the following specification:

$$log\left(\widehat{\mathcal{W}}\left(x, h, \psi_{ct}\right)\right) - log\left(\widehat{\mathcal{W}}\left(x, h, \bar{\psi}_c\right)\right) = \theta_0 + \theta_1\left(\psi_{ct} - \bar{\psi}_c\right) \tag{6}$$

   θ_1 represents, at the state level, the extent to which watch time decreases (on average across streaming hours) when there is increased competition, denoted by ψ_{ct}. Because I imposed that predicted watch time should decrease with respect to ψ_{ct} during the training of XGBoost, θ_1 is negative for all states as expected. This is consistent with the assumption that a streamer's watch time decreases when other streamers in the same category collectively stream more.

5. Among the above outcomes, only $\widehat{\mathcal{W}}^{conc}(x, h, \bar{\psi}_c)$ is used during the estimation stage. This reflects the assumption that ψ_{ct} is fixed at observed average level and streamers have a rational expectation for it.[4]. When searching for the full

[2]To be precise, all independent variables were discretized, and broadcasting category and profitability are individual streamer level variables.

[3]For this approximation, I minimize the mean squared errors. The methodology was developed by Kuosmanen (2008) and Kuosmanen and Johnson (2010). I implement their method using python `cvxopt` package and `mosek` optimization tool.

[4]From this perspective, I am assuming that all data points come from one equilibrium, while the endogenous variable ψ_{ct}, which differs across months, is represented as $\bar{\psi}_c + e_{ct}$. I am disregarding e_{ct} during the estimation stage because ψ_{ct} does not vary significantly across months

equilibrium during the counterfactual stage, I use the adjusted version of watch time predictions, across ψ_{ct}, given by

$$\widehat{\mathcal{W}}^{adj}\left(x, h, \psi_{ct}\right) = exp\left(\hat{\theta}_0 + \hat{\theta}_1\left(\psi_{ct} - \bar{\psi_c}\right)\right) \cdot \widehat{\mathcal{W}}^{conc}\left(x, h, \bar{\psi_c}\right) \tag{7}$$

where $\hat{\theta}_0, \hat{\theta}$ are from equation 6. $\widehat{\mathcal{W}}^{adj}$ maintains concavity with respect to streaming hours, and smoothly changes with respect to ψ_{ct}, as desired.

Fit Assessment Watch time approximation errors may arise from three sources. First, there are prediction errors stemming from the discretization of continuous variables. Second, there are additional approximation errors resulting from the ML model and the imposition of concavity on watch time with respect to my streaming hour. Lastly, I assume that the weighted sum of hours remains fixed at the mean value over the month, whereas in the real world, it can vary across months.

To investigate how well this XGboost with shape restrictions approximates watch time, I plot observed vs. predicted watch time (specifically, $\widehat{\mathcal{W}}^{conc}(x, h, \bar{\psi_c})$, i.e. concavity imposed predictions) at the state-hour level (Figure B.11). One concerning aspect is the horizontal blue line at the bottom, which suggests that some non-zero watch time was systematically approximated to zero.

Two factors contributing to the emergence of this blue line can be identified. First, the imposition of concavity on watch time concerning hours, as well as the predicted watch times for zero hours, both tend to be zero. Second, since I approximated streaming hours as $0, 50, ..., 300$, some streamers who streamed for around 10-20 hours are treated as observations with zero hours, even though their observed watch time might not be as small.

However, further investigation reveals that this horizontal error line might not be a significant concern because the average observed watch time for observations with zero predicted watch time was only 312.23, whereas the average observed watch time is 11433.26. The log transformation that I used for readability visually amplifies the errors where values are small.

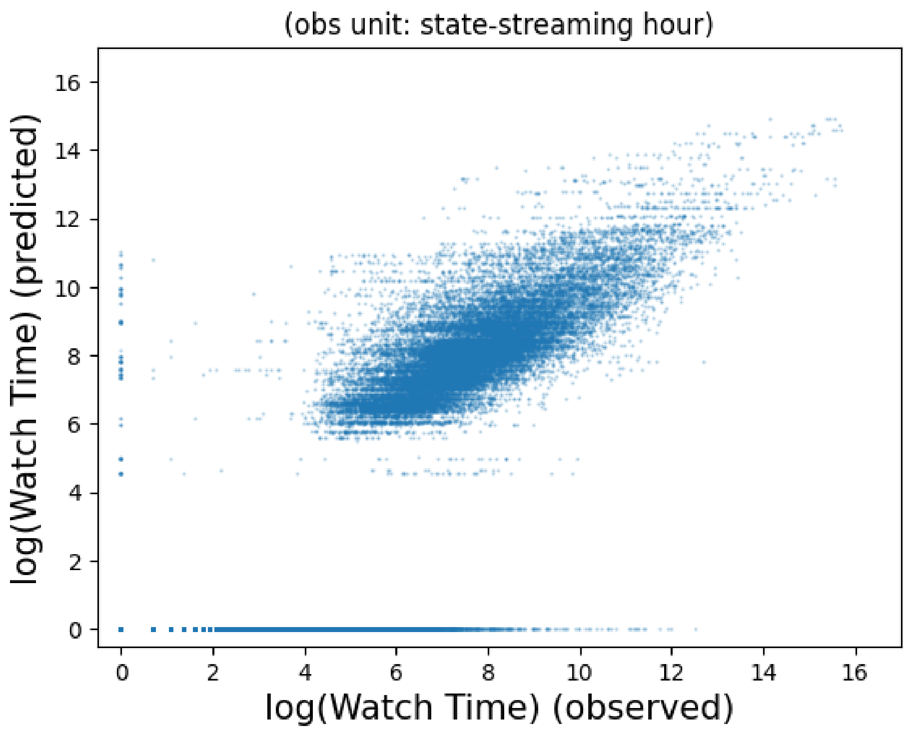


Figure B.11. Observed vs. Predicted watch time (specifically, $\widehat{\mathcal{W}}^{conc}(x, h, \bar{\psi}_c)$, i.e. concavity imposed predictions). The observation unit is state-action (streaming hour). Further investigation of the horizontal blue band at the bottom is in Fit Assessment paragraph below.

C Additional Model Fit Assessments

C.1 Predicted vs. Observed Industry state

Using the full equilibrium search algorithm, I additionally investigate if the distribution of score and bookmarks at industry level from factual simulation are close to observed counterpart. This counterpart was taken as given throughout the estimation stage. The assessment appears in Figure C.12.

Overall, the largest gaps come from the fraction of highly popular streamers, and it may have come from potential bias in entry process. A streamer tracking website started tracking on January 2019, and my data period starts from October 2019 since revenue information is available from then.

It could be the case that the website started to track (relatively) popular streamers first and less popular streamers later. In this case, in the entry process I am using, less popular streamers may have been overrepresented.

Additionally, I compare various industry-level state variables from factual simulations with the observed averages, as shown in Table C.10. The main takeaway is that the predictions from factual simulations, while not perfect, are not too far from the observed industry state. This finding is encouraging, as the estimation process using pseudo-MLE did not directly target matching these features.

Figure C.12. Observed vs. predicted marginal distributions of *score* and *log*(*bookmark*)

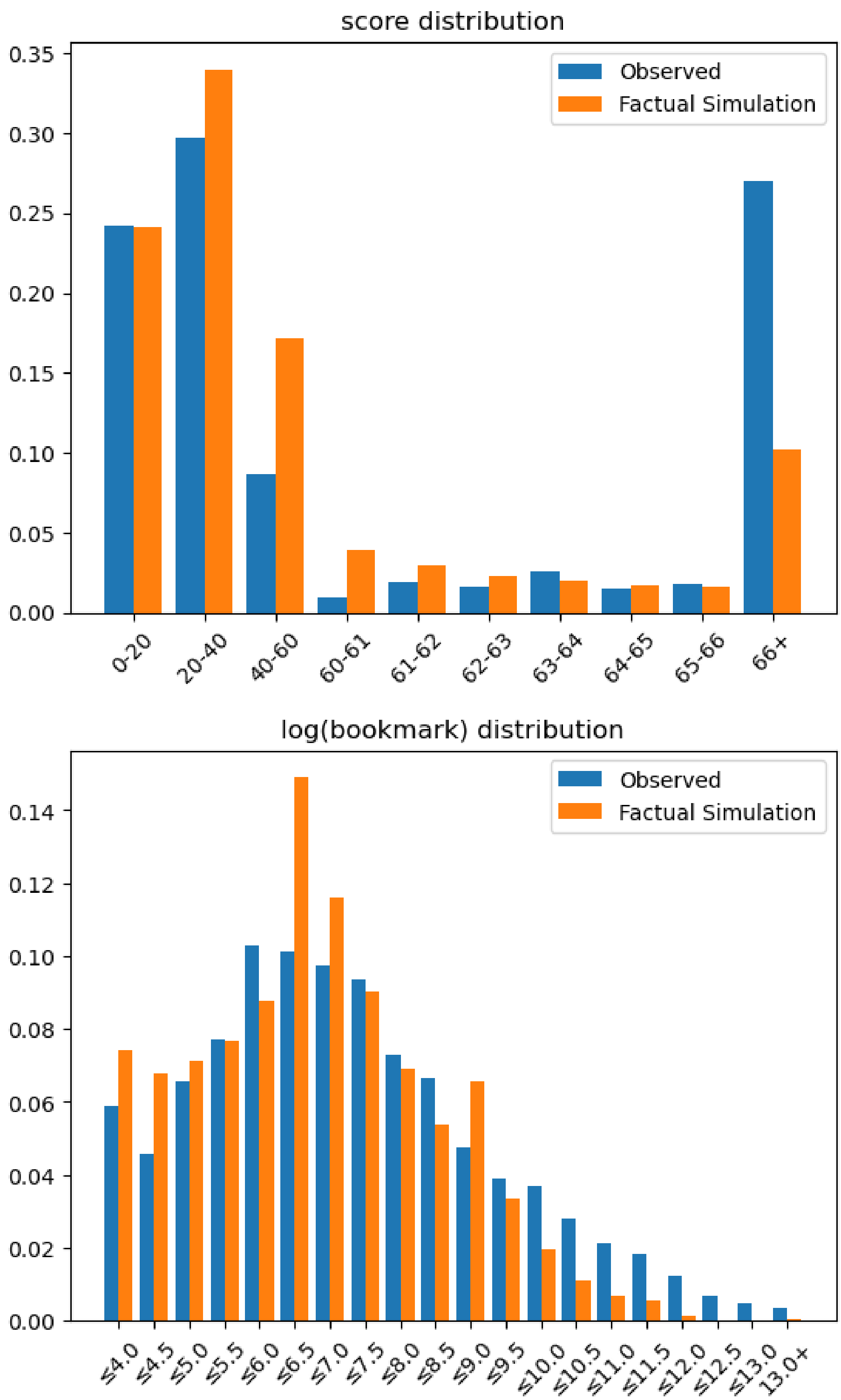


Table C.10. Predicted versus observed platform level outcomes.

	Observed	Predicted
Streamers newly getting the reward		
Total number	38.00	21.36
Game (category share)	0.39	0.57
Social	0.36	0.21
Other	0.25	0.21
Low (α^{rev} share)	0.14	0.15
Mid	0.31	0.33
High	0.56	0.51
Weighted sum of streaming hour ψ_{ct} (Unit: 1,000)		
Game (category)	1827.13	2397.42
Social	725.69	812.12
Other	962.43	949.58
The number of streamers	5888.71	8034.73
Total revenue (Unit: 1,000 USD)	188.91	229.47
Platform's revenue share	0.34	0.35

Notes: The numbers in the left column are from data averaged across seven months. The numbers in the right column are from factual simulation endogenizing ψ_{ct}. The shares may not add up to one due to rounding errors.

C.2 Out of sample assessment through doubling slot counterfactual

Counterfactual 1 (doubling slots) provides a prediction: the new approval probability by score when the number of slots was doubled. In the real world, this change was implemented, allowing me to compare the prediction with the realized data and conduct an out-of-sample evaluation of model fit.

First, empirical approval probabilities and cutoffs were plotted for two time periods: the main data period (October 2019 to April 2020) and the time period after the slot increased and the cutoffs were stabilized (August 2020 to October 2021). I used 2-point score bins, with the x-coordinates representing the average score of each bin. (The decrease in probability well above the cutoff may indicate the presence of popular streamers who are consistently uninterested in the rewards program, which I will discuss as a limitation below.)

The main takeaway is that the average score cutoff decreased from 65.52 to 60.68. This result is roughly consistent with my counterfactual prediction. As shown in Figure C.14, counterfactual simulation 1 predicted that if the number of slots doubled and reached a steady state, all streamers with scores greater than 61 would always be approved if they applied, while streamers with scores ranging from 60 to 61 would have some probability of being approved.

Lastly, one confounding factor remains. In the real world, there was a slight change in the calculation method for scores when the number of slots increased. Streamers with an average viewer count of less than 10 started to receive 0.4×50 points instead of a value of zero. To make the scores comparable across the two periods, I computed all scores in the old way. But because the change was minor, the two scores are highly correlated with similar levels.

C.3 Deviation Factors

There are multiple factors that contribute to the gap between the model's prediction and the real data, which is listed below. I do not repeat factors that were discussed in section 3.4. (e.g. presence of streamers who are persistently not interested in the rewards).

First, I discretize continuous variables. For example, I found those streamers who were accepted into the rewards program generally had a higher per watch time revenue parameter, α^{rev}, within a discretized group. For instance, in the Social-low

Figure C.13. Observed approval probabilities by score and average cutoff, before slot increase: October 2019 to April 2020, after slot increase: August 2020 to October 2021

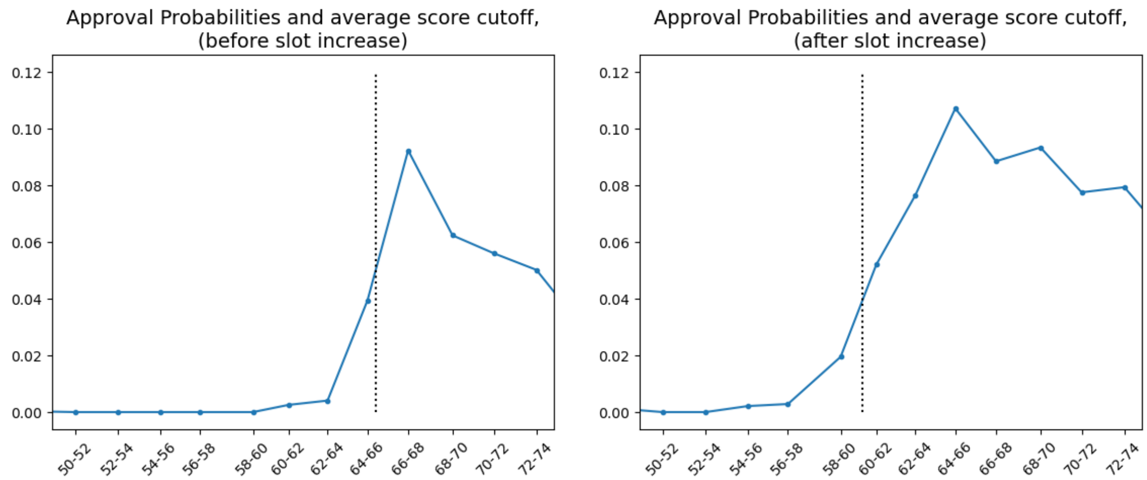
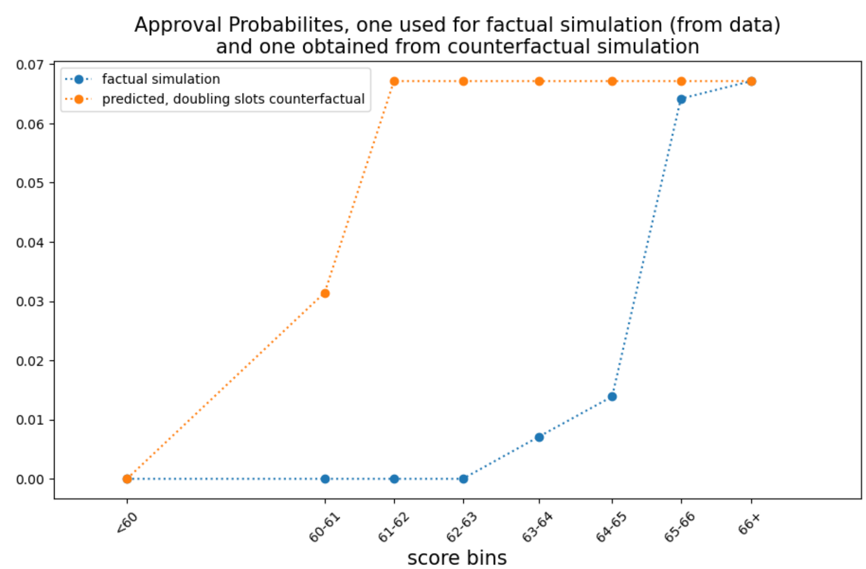


Figure C.14. Approval probabilities change from counterfactual simulation 1



α^{rev} group, the discretized value was 0.239. This contrasts with streamers who were newly approved in the group and had an average α^{rev} of 0.488.

Second, I use the entry process observed during the data period. However, a substantial fraction of streamers in the data would have entered the market long before the data period, possibly when the entry process was different.

Third, the equilibrium of the dynamic game described in section 3.3 is assumed to approximate the streamers' decisions. However, in Table C.10, there is a gap between the endogenized (predicted) and observed weighted sum of streaming hours. Therefore, the observed data may not align perfectly with this equilibrium concept.