Online Appendix for "The Promotional Effects of Live Streams by Twitch Influencers"

Yufeng Huang (University of Rochester) and Ilya Morozov (Northwestern University)

A Industry and Data: Additional Details

A.1 Selection of streamers and games

During a preliminary one-week period, on April 22-28, 2021, we sent repeated high-frequency requests to Twitch API to record which video games are frequently streamed on Twitch and attract the largest audiences. To this end, every 10 minutes, we requested from Twitch API the list of 200 games that had the largest total number of viewers on Twitch at the time of the request. Each of these games had a corresponding game ID assigned to it by Twitch. We selected all game IDs that appeared in our top 200 list at least once during that week, which generated a list of 1,010 game IDs. We then manually removed game IDs of non-Steam games (e.g., "Fortnite" and "League of Legends") for which we had no way of tracking player counts, game IDs for non-gaming live streams (e.g., streamed labeled as "Just Chatting," "Music," "Sports"), and game IDs for which we could not access data on player counts from Steam API (23 games for which Steam does not report player counts). Only eight games from our sample had zero active players throughout the whole period, and we omitted these games from our analysis. After applying these filters, we were left with a list of 599 games that constitute our main sample throughout the paper.

In the same preliminary period, we sent hourly requests to Twitch API to collect the lists of the most popular streamers. Every hour, we requested the list of the 100,000 most-viewed streamers who were live on Twitch at the time of the request, and we recorded which game they were streaming and how many viewers they had at that time. By our calculations, sampling 100,000 most-viewed streamers in each request was sufficient to include the vast majority of all live channels with non-negligible viewership.[27] By the end of the preliminary data collection period, our sample included 912,754 unique streamers. Because of the daily quota of Twitch API requests, we knew we would not be able to simultaneously track more than 60,000 streamers during our main data collection period.[28] We therefore randomly sampled 60,000 streamers from the full set

[27] According to twitchtracker.com, in May 2021 Twitch hosted, on average, 110,367 live streams at any given time (source: https://twitchtracker.com/statistics/channels). In addition, many of these streams had an average viewership of no more than 1-2 people. For this reason, we are confident that the sample of streamers we built in this preliminary data collection period covers the vast majority of live channels with non-zero viewership.

[28] Although Twitch API allowed to retrieve the list of the most-viewed streamers using only one request, it required

of 912,754 streamers, with a sampling weight that is linearly increasing in the streamer's total number of viewers during this preliminary period. We then kept this list of 60,000 pre-selected streamers constant throughout the main data collection period between May 11, 2021, and December 31, 2021.

A.2 Re-weighting of viewership counts

As explained in Appendix A.1, we constructed our main sample of streamers by randomly sampling from the full list of streamers recovered during the preliminary period of April 22-28, 2021. We sampled streamers from this list while placing more weight on selecting those with higher average viewership. To account for this sampling procedure in our analysis, we re-weight the viewership and streamer counts so that they correctly reflect the expected values of these variables in the entire population of Twitch streamers.

Consider, for example, the variable viewers_{jt} in equations (1) and (2). Let $v_{jt}(p)$ denote the value of this viewer count for a streamer of popularity p, where our measure of "streamer popularity" is based on the total number of viewers that the streamer had during the preliminary period. Additionally, let $f(p)$ denote the number of streamers with popularity p in the sample of 912,754 streamers constructed during the preliminary period, and let $g(p)$ denote the number of streamers with popularity p within the sample of 60,000 streamers tracked during the main data collection period (call them S_{60K}). We calculate the total number of viewers_{jt} as follows:

$$\widehat{\text{viewers}}_{jt} = \sum_{s \in S_{60K}} v_{jts} \cdot \psi(p_s), \tag{6}$$

where $\psi(p) = f(p)/g(p)$ is an adjustment factor for streamers with popularity p. We use this re-weighting procedure to compute the total viewership as well as the number of sponsored and organic live streams at any given time.

This formula accounts for the fact that, while selecting 60,000 streamers via weighted sampling, we somewhat under-sampled high-viewership streamers and substantially under-sampled low-viewership streamers. For example, the factor $\psi(p)$ is around 1.5 for the top 5% most popular streamers, implying that in the expected terms, we are missing about one-third of these popular streamers in our sample. Similarly, this factor is 168.3 for streamers with viewership below the median, meaning that we only sampled 0.6% of those smaller streamers into our main sample. The weighted summation in (6) enables us to adjust for these imbalances by scaling up the viewership counts accordingly. An implicit assumption we make with this re-weighting is that streamers not included in our sample would on average stream the same games to the same number of viewers

making one request per streamer to track individual streamers. We could not send more than 60,000 requests in each 10-minute interval while still completing all requests within 10 minutes.

as the streamers included in our sample.

To gauge the accuracy of our viewership estimates in (6), we validated these estimates for 10 games selected from our sample. We chose five most viewed games and also randomly sampled five games that are outside the list top 100 most viewed games. For these 10 games, we manually collected the daily information on the peak numbers of viewers from Twitchtracker.com – the most accurate measure of viewership we could find – and we compared it to the peak viewership numbers implied by our main estimation sample.

Figures A.1 and A.2 present the results of this comparison. Although our estimates of daily peak viewers are somewhat lower than the viewership numbers reported by Twitchtracker, the discrepancy is relatively small. For most games, our viewership estimates are only about 70-90% of the total viewership reported by Twitchtracker, with the average coverage of 73.2% across the 10 games. Moreover, our estimates correctly capture the long-term viewership trends of all games, and they seem to accurately approximate most of the visible daily viewership fluctuations. The average correlation between the two time series is around 0.90 within this sample of 10 games. Overall, these results suggest that our viewership estimates provide a reasonable approximation to the total viewership of games.

A.3 Monetary incentives of Twitch streamers

One might wonder why streamers spend hours broadcasting video games on Twitch, considering that anyone can watch their broadcasts free of charge. As it turns out, live streaming video games can be a lucrative business for the top streamers and a steady source of extra income for many others. Streamers begin their careers as *Twitch Affiliates* who mostly earn income from the donations and subscriptions of viewers. Most donations are direct PayPal transfers from viewers (streamer gets 100% of income) or “virtual bits” that viewers donate through the chat window (streamer gets 75% of income). In addition, viewers can subscribe to a channel for a $4.99, $9.99, or $24.99 monthly fee or use one free subscription that comes with Amazon Prime. Subscribing to a streamer gives viewers access to exclusive content, custom emotes, and sometimes to ad-free viewing of the streamer’s content.

The most popular streamers can make *Twitch Partner*, which requires them to stream at least three times a week and consistently average at least 75 concurrent viewers. Twitch Partners get access to exclusive contracts for promoting games and non-gaming products on their channels. According to survey data, a typical Twitch Affiliate earns $100-$1,000 per month, and a full-time streaming Partner earns $3,000-$6,500 per month (Goodman, 2021). Although most streamers do not publicly disclose their incomes, the payout data leaked from Twitch in October 2021 suggest that the most popular streamers earn over $100,000 per month (Scullion, 2021).

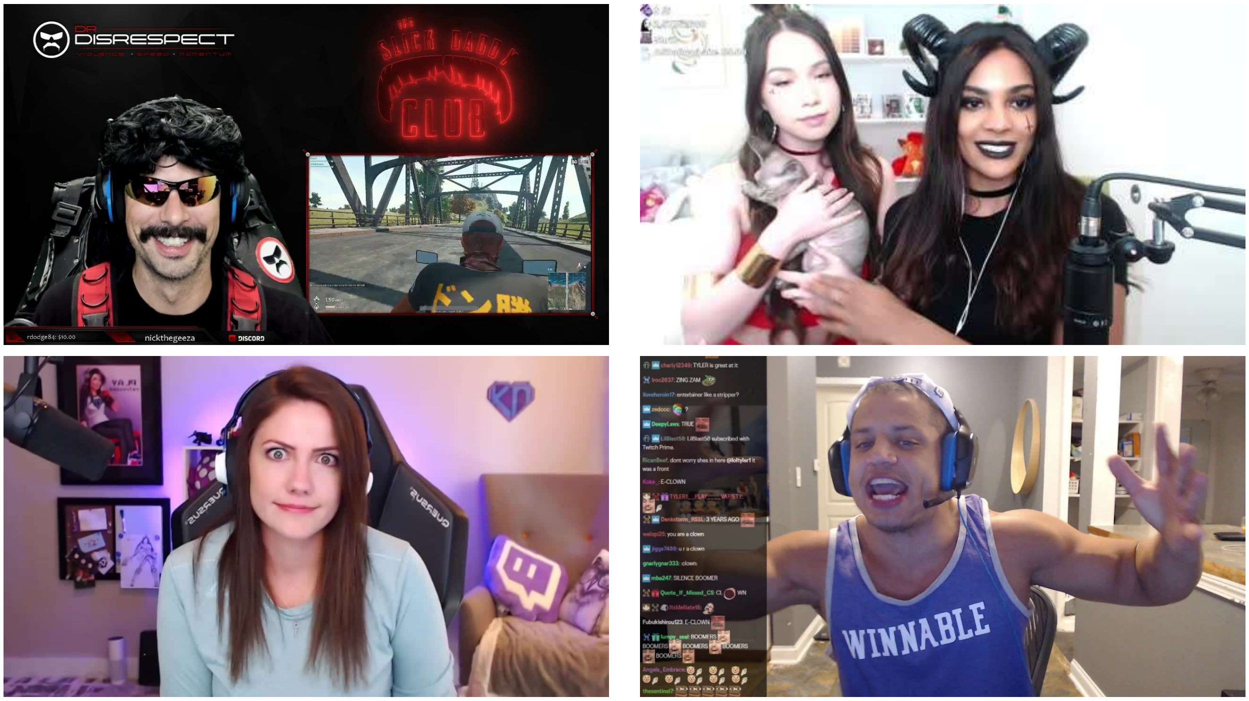
To estimate the cost of sponsoring live streams, we collected daily data on the number of active subscriptions from twitchtracker.com, a third-party website that reports the number of active subscriptions of the Twitch streamers who chose to publicly disclose this information. We tracked 10,000 most subscribed streamers reported on Twitchtracker daily and collected their current number of active subscriptions by subscription type (i.e., Tier 1, Tier 2, Tier 3, or Amazon Prime). Using these subscription counts, we estimate the monthly revenue of each streamer s on date d using the following formula:

$$\text{SubsRevenue}_{sd} = 50\% \times (\$4.99 \times \text{Tier 1 Subs}_d + \$4.99 \times \text{Prime Subs}_d + \$4.99 \times \text{Gifted Subs}_d + \\ + \$9.99 \times \text{Tier 2 Subs}_d + \$24.99 \times \text{Tier 3 Subs}_d), \tag{7}$$

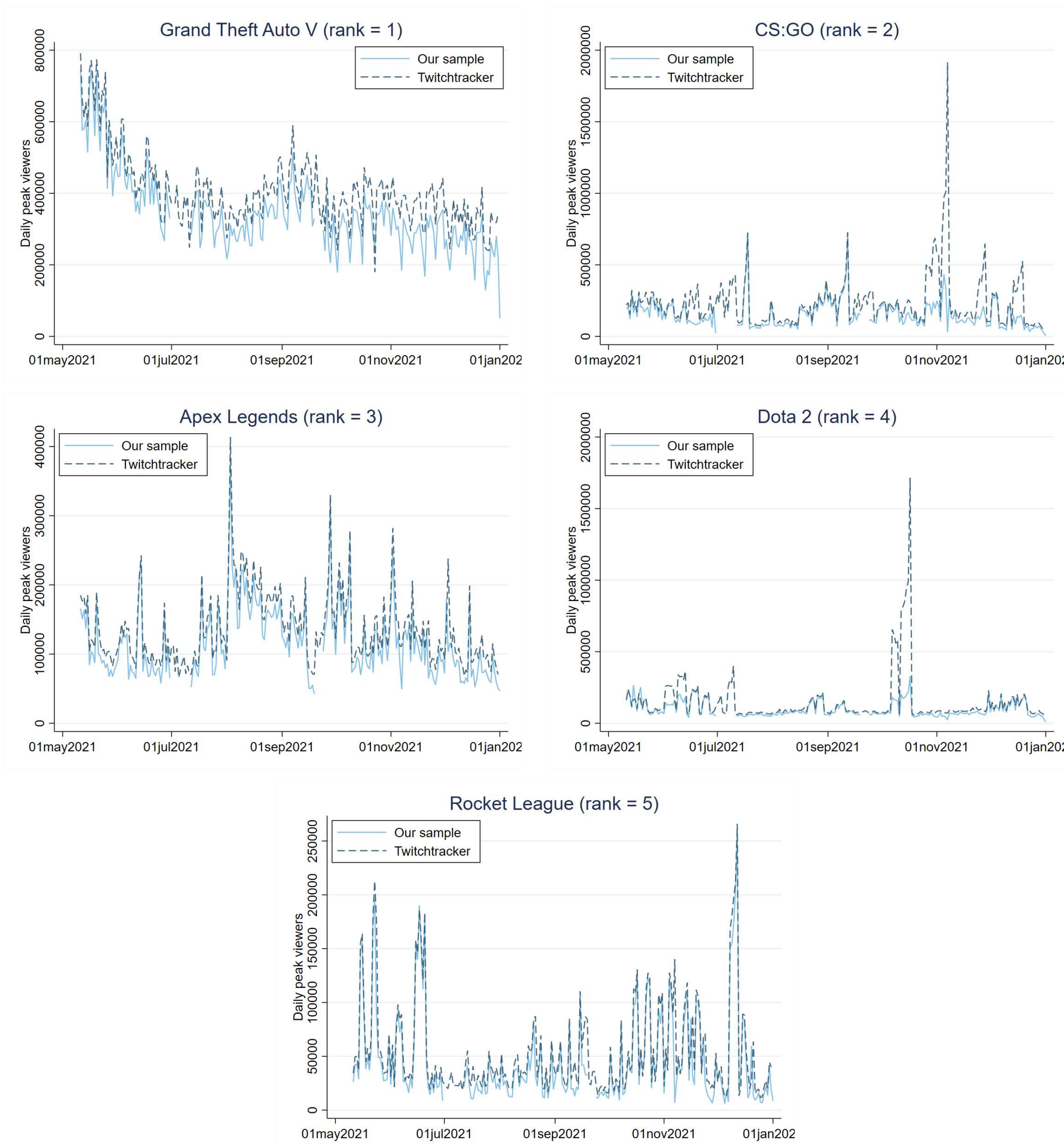
where Tier 1 Subs_d-Tier 3 Subs_d are the numbers of active Tier 1, 2, and 3 subscriptions; Prime Subs_d is the number of active Prime subscriptions; and Gifted Subs_d is the number of gifted subscriptions. The dollar values \$4.99, \$9.99, and \$24.99 reflect the contribution of each active subscription to the monthly revenue of this streamer. We multiply the total monthly revenue by 50% to reflect that Twitch retains 50% of subscription income. Having computed the expected monthly revenue of streamer s on date d, we average this value across all dates for which we have non-missing subscription data. This process yields an estimate of each streamer's monthly income after the streamer has paid the commission to Twitch.

Although we do not have systematic data on all income sources, anecdotal evidence suggests that subscription revenue is a major source of income for top streamers. For example, Mediakix estimates that streamer Ninja makes three-quarters of his income from subscriptions.[29] Similarly, the subscription income we observe for streamers xQcOW, Auronplay, and Asmongold is about 50% of their total monthly payouts of \$325,170, \$117,436, and \$98,139, which combine subscription and advertising revenues (Scullion, 2021).

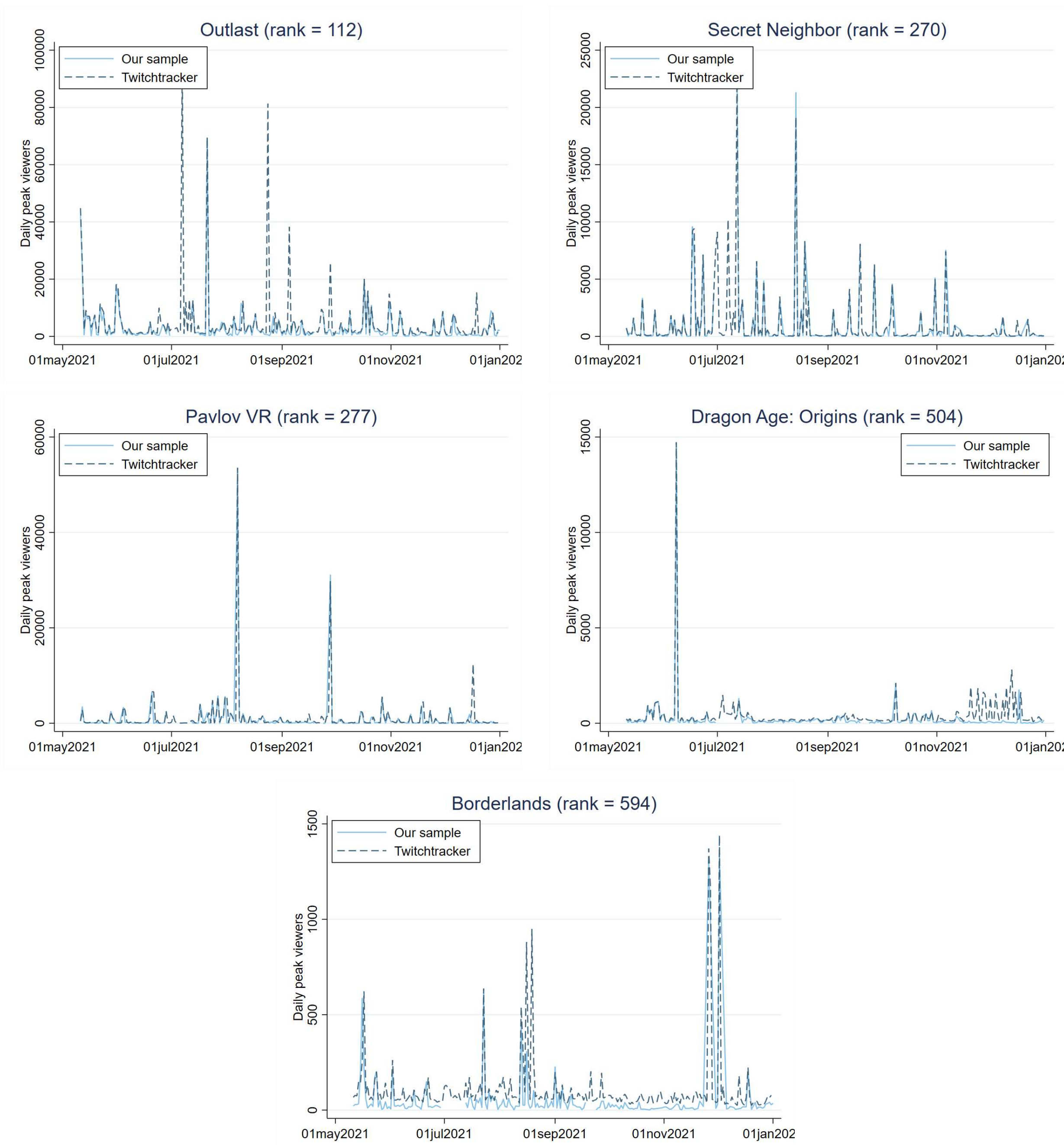
[29] Source: https://influencermarketinghub.com/twitch-money-calculator/.



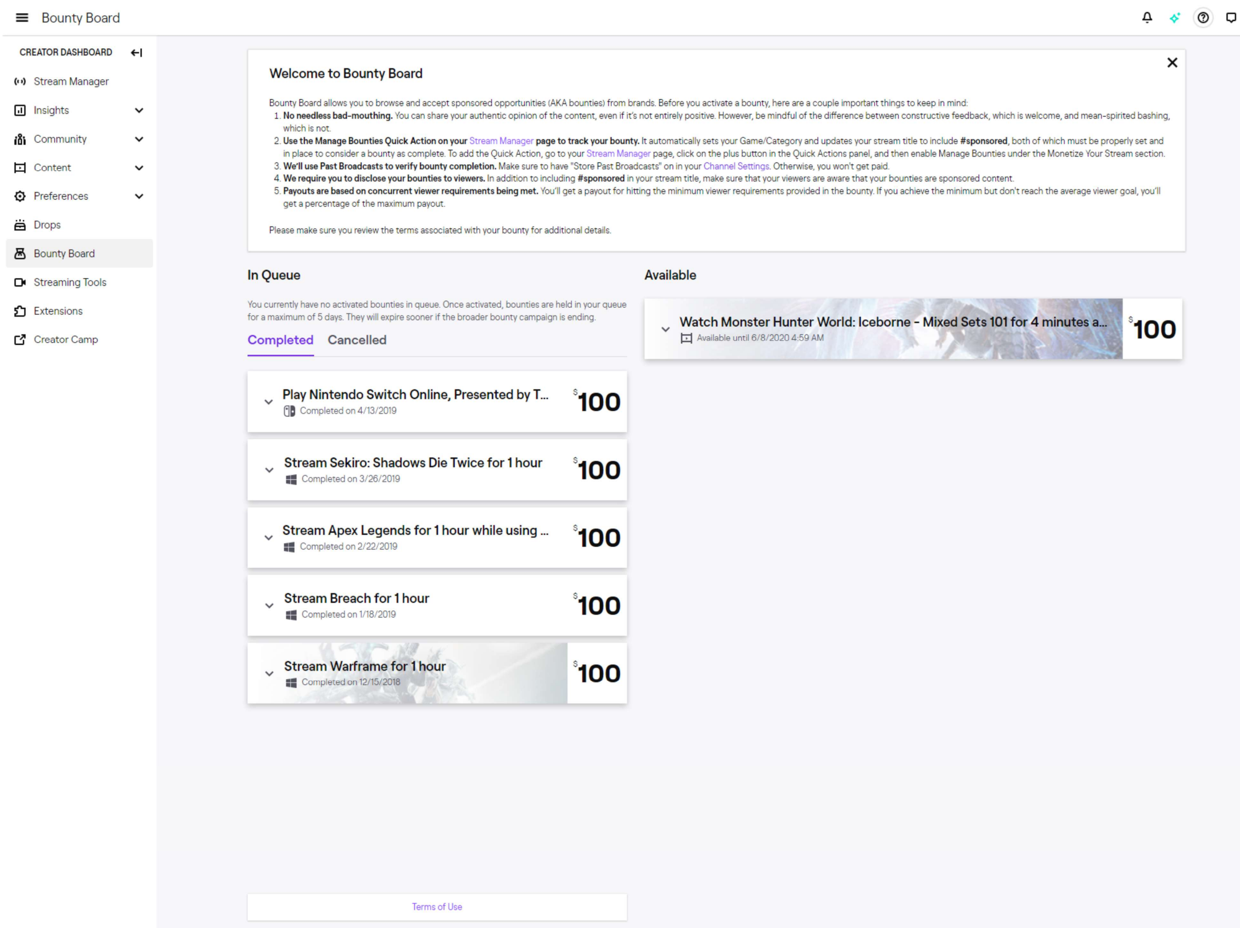
Appendix Figure A.3: **Many popular Twitch streamers have exuberant and memorable personalities, which makes their content funny and entertaining.** The figure shows screenshots from broadcasts of Twitch streamers Dr. Disrespekt (top left), Sydeon (top right), KayPea (bottom left), and Tyler1 (bottom right).



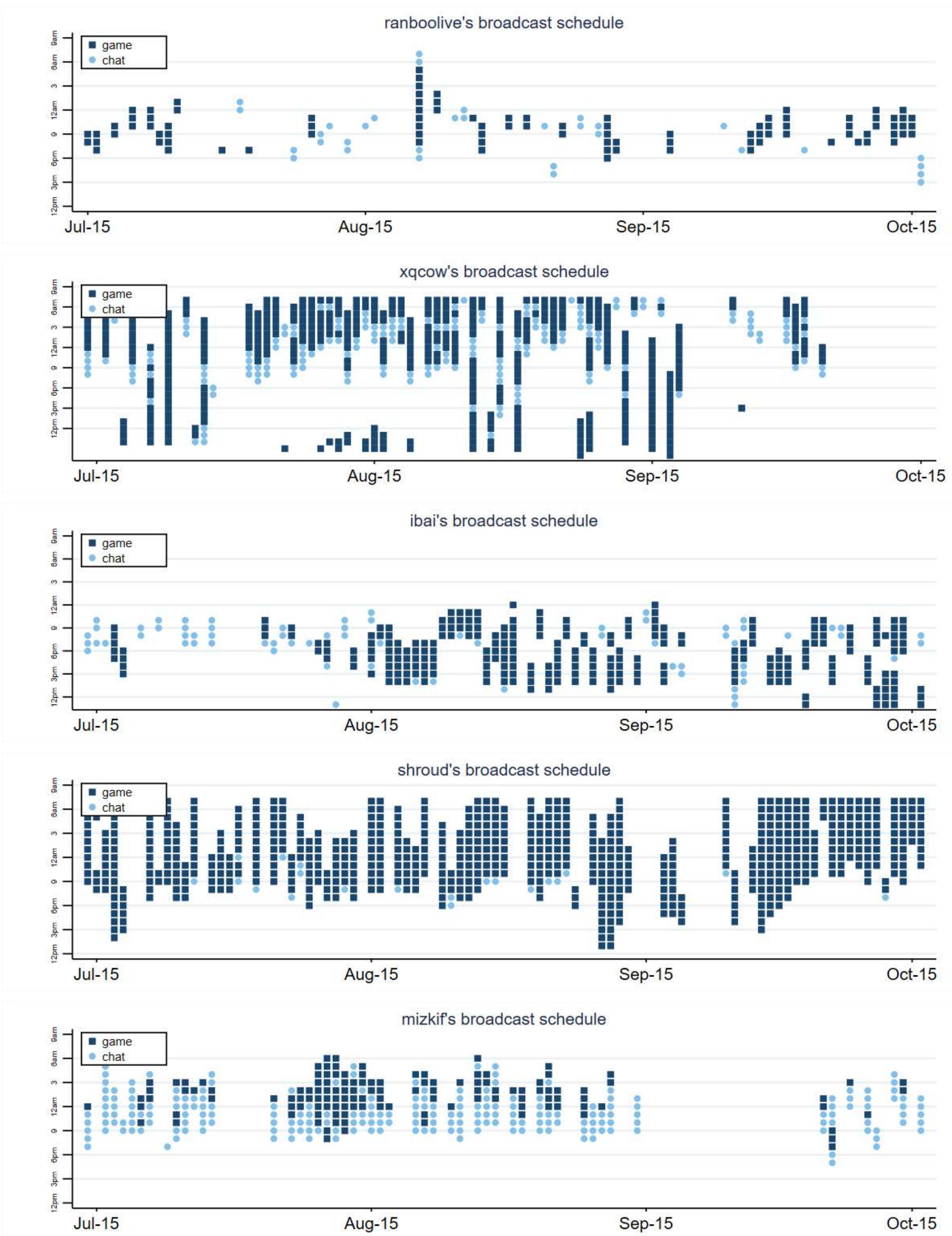
Appendix Figure A.1: **Comparison of daily peak viewership in our sample (solid line) and daily peak viewership reported on Twitchtracker.com (dashed line).** The graphs correspond to the five most viewed games in our sample.



Appendix Figure A.2: **Comparison of daily peak viewership in our sample (solid line) and daily peak viewership reported on Twitchtracker.com (dashed line).** The graphs correspond to the five games randomly sampled from the list of games outside the top 100 most viewed titles.



Appendix Figure A.5: **A screenshot of the Twitch Bounty Board.** Twitch Bounty Board is an internal application on Twitch that enables game publishers to find streamers willing to broadcast a game for a fixed sponsorship fee.



Appendix Figure A.4: **Daily broadcast times of top Twitch streamers (additional examples).** This figure extends Figure 2 in the main text by illustrating the daily work hours of five top streamers from Table 1. Each graph shows hours of the day in which a streamer was live on Twitch (squares), with the horizontal axis showing the date and the vertical axis showing the time of the day. Blue squares indicate when the streamer was broadcasting a game, and light grey squares indicate when the streamer was "just chatting" with the audience.

Appendix Table B.1: **Variance decomposition for start times, end times, and duration of streams**

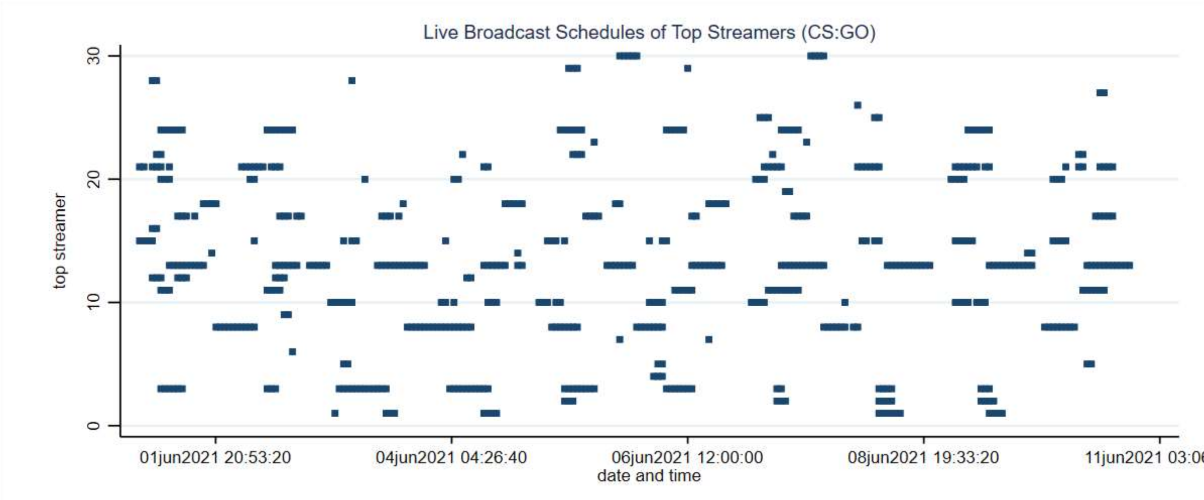
		% of total variance explained by fixed effects		
	Std.Dev	Game-date FE	Streamer-game FE	Residual variation
All streamers:				
Stream start time	6.10	4.3%	46.4%	49.2%
Stream end time	6.16	4.1%	42.4%	53.5%
Stream duration (hrs)	3.34	7.4%	35.4%	57.1%
Top streamers:				
Stream start time	6.07	10.4%	48.0%	41.6%
Stream end time	5.93	11.7%	43.0%	45.3%
Stream duration (hrs)	5.65	10.3%	35.1%	54.6%

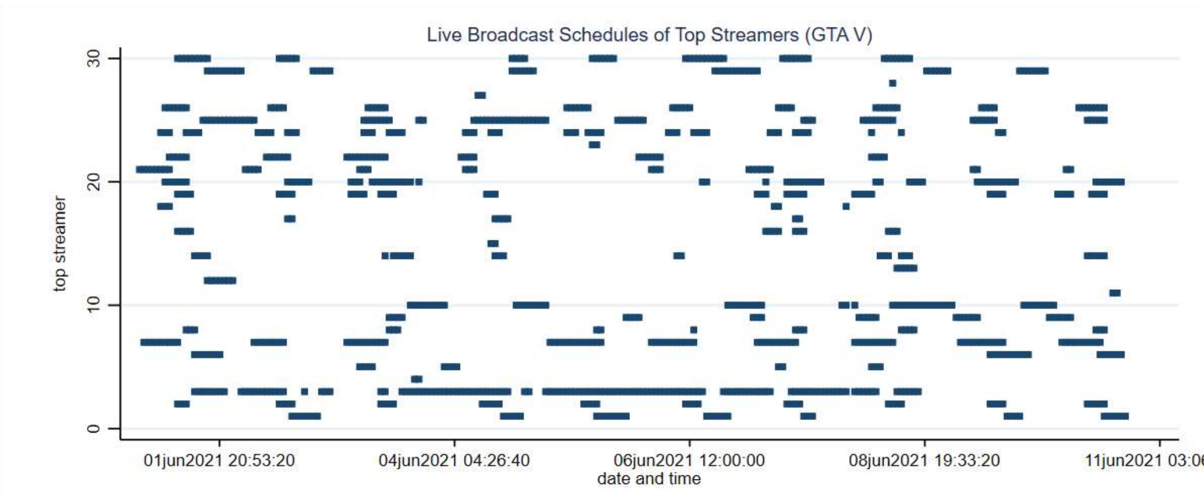
The table shows the standard deviations of start times, end times, and duration of streams (column 1) for 599 Steam games in our sample. We decompose this variance into three components: (a) variation explained by game-date fixed effects (column 2); (b) variation explained by streamer-game fixed effects (column 3); and (c) residual variation after controlling for both game-date and streamer-game fixed effects (column 4). The top panel reports statistics for all streamers, and the bottom panel reports the same statistics for the top 5% streamers.

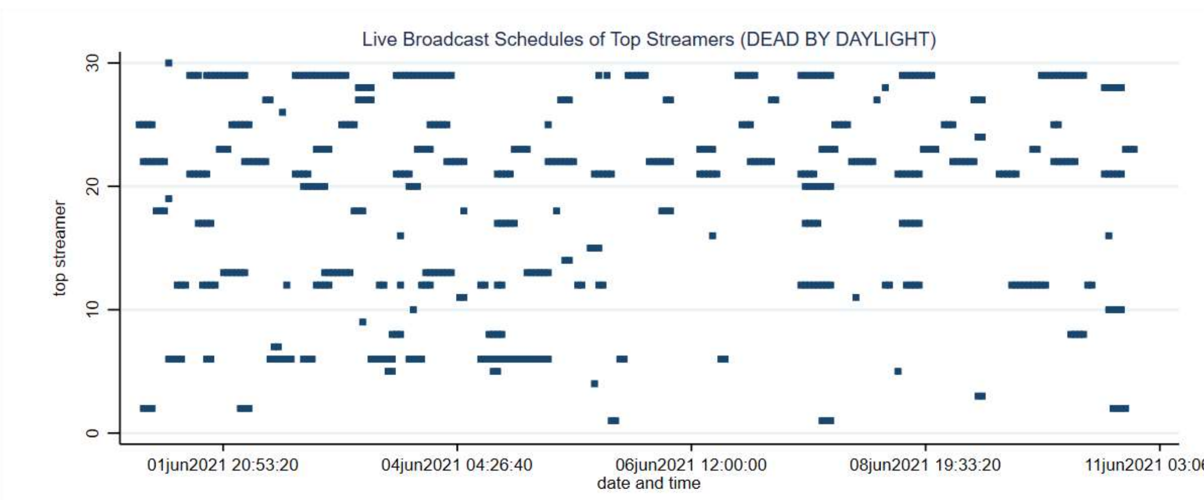
B Additional Estimation Details

B.1 Variation in streaming schedules: a decomposition

As explained in Section 3, our empirical strategy leverages this within-day variation in the broadcast times of top streamers as a plausibly exogenous shifter of the game's viewership on Twitch. To provide further support for this strategy, in Table B.1 we summarize this variation by reporting the standard deviations of start times, end times, and stream duration. We decompose this variation into three components: (a) variation explained by game-date fixed effects, (b) variation explained by streamer-game fixed effects, and (c) residual variation after accounting for game-date and streamer-game fixed effects. Among the top streamers, 42%-55% of the total variation of top streamers' broadcast hours cannot be explained by long-term trends in game popularity (game-date fixed effects) or by the fact that certain streamers prefer to broadcast specific games at fixed times (streamer-game fixed effects). This decomposition confirms that streamers follow irregular broadcast times for a given game and that the broadcast times are rarely synchronized across streamers who broadcast the same game on the same date.







Appendix Figure B.1: **Distribution of broadcast times across top Twitch streamers for individual games.** Each graph shows broadcast hours of 30 randomly sampled top streamers for the period between June 1 and June 10, 2021. Blue squares indicate when each streamer was broadcasting the game.

	Distribution of daily averages across games						
	Mean	S.E.	Q 5%	Q 25%	Q 50%	Q 75%	Q 95%
Avg num. of top streamers live	0.46	2.98	0.00	0.00	0.03	0.12	1.37
Avg num. of top streamers live (if non-zero)	1.51	3.14	1.00	1.00	1.04	1.21	2.39
Maximum number of top streamers live	1.00	5.05	0.00	0.03	0.16	0.57	4.08
Stream duration (hrs)	2.79	1.15	1.19	1.99	2.80	3.48	4.60

Appendix Table B.2: **Broadcasting activity of top streamers across different games.** These summary statistics illustrate the variation in our IVs $\tilde{z}_{jt}$, which capture how many top streamers are broadcasting game j in time period t. We first compute the daily averages across all days and then report the distribution of these averages across 599 games.

B.2 Variation captured by instruments z_{jt}

In this section, we describe variation isolated by instruments z_{jt} in Section 3.2. Table B.2 describes the variation in the number of active top 5% streamers captured by these instruments. In this table, we compute the average number of top streamers broadcasting a specific game at a given time, the maximum number of top streamers broadcasting a game simultaneously at a given point, and the average stream duration among top streamers. We report these statistics in Table B.2. As these statistics show, the average game is not always broadcasted by top streamers and is broadcasted by at most one top streamer on an average day. Additionally, when top streamers do broadcast a game, those broadcasts are on average almost three hours long. This long average stream time suggests that most top streamers spend several hours exploring a given game and showcasing its gameplay rather than picking it up for a few minutes during breaks in their main activity.

B.3 Details of GMM Estimation

We estimate model (1) using the following GMM estimator. First, to simplify notation, we re-write (1) as

$$y_{jt} = \beta \cdot g(v_{jt}, \delta) + \gamma' u_{jt} + \varepsilon_{jt}, \tag{8}$$

where $y_{jt} = \log(1 + \text{Players}_{jt})$; $g(v_{jt}, \delta) = \log(1 + \sum_{\tau=0}^{72} \delta^{\tau} \widetilde{v}_{j,t-\tau})$, where $v_{jt} = (\tilde{v}_{jt}, \ldots, \tilde{v}_{j,t-72})$ is a vector of current and lagged viewership counts; $\gamma' u_{jt} = \lambda_{j,d(t)} + \mu_{j,h(t)} + \eta_t$, where γ is a vector that stacks game-date, game-hour-of-the-day, and time fixed effects; and u_{jt} is a vector of corresponding indicator functions. Recall that we also define a vector of instruments $z_{jt} = (\tilde{z}_{j,t}, \tilde{z}_{j,t-1}, \ldots, \tilde{z}_{j,t-12})'$ where $\tilde{z}_{j,t}$ is the number of top streamers broadcasting game j in hour t.

To construct a GMM estimator, we use the moment conditions $E(\varepsilon_{jt} w_{jt}) = 0$, where $w_{jt} = (z'_{jt}, u'_{jt})$ and $\varepsilon_{jt} = y_{jt} - \beta \cdot g(v_{jt}, \delta) - \gamma' u_{jt}$. A straightforward way to obtain GMM estimates of parameters would be to minimize the GMM objective function with the identity weighting matrix:

$$G(\theta) = \hat{m}(\theta)'\hat{m}(\theta), \tag{9}$$

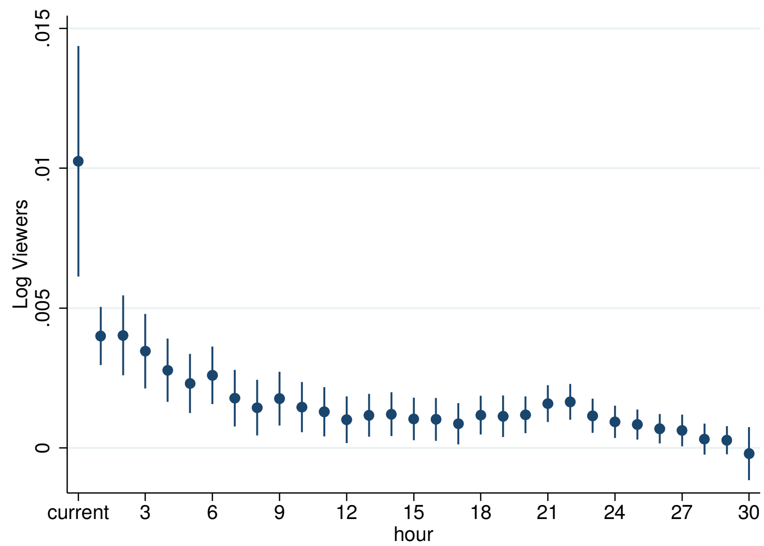
where $\hat{m}(\theta) = (1/JT)\sum_{j=1}^{J}\sum_{t=1}^{T}\varepsilon_{jt}(\theta)\cdot w_{jt}$ are consistent estimates of moments $m(\theta) = E(\varepsilon_{jt}w_{jt})$, and θ is a vector of all parameters to be estimated. However, the main challenge with minimizing the objective function (9) is the large number of fixed effects in w_{jt}. Given that our sample includes 599 games, we need to estimate 599×240 game-date fixed effects, 599×24 game–hour-of-the-day fixed effects, and $5,760$ hour fixed effects, which amounts to about 160,000 parameters in total. We also cannot apply the standard within-group 2SLS estimator because the function $g(v_{jt}, \delta)$ is non-linear in the unknown parameters.

To reduce the computational burden, we adopt the following numerical procedure. We first fix a candidate value of the persistence parameter δ and estimate all other parameters of the model. Importantly, when δ is known, the model (8) becomes a linear regression model, which is much easier to estimate. To estimate parameters other than δ, we minimize the objective function $G(\theta)$ in (9) using a numerically efficient implementation of the 2SLS estimation algorithm.[30] We note that this estimator is equivalent to 2SLS applied to the transformed model where all variables are demeaned with respect to their means in each game-date group (Wooldridge, 2010, p.300). Assuming the number of game-date groups goes to infinity (i.e., $JT \to \infty$ while T is fixed), this within-group 2SLS estimator is consistent under the strict exogeneity (Arellano, 2003, p.133). For this reason, our estimator is consistent under the strict exogeneity assumption made in equation (3).

Since we can recover the GMM estimates of other parameters for a fixed value of δ, we can conduct a uni-dimensional search over possible values of δ in an outer loop, while re-computing the estimates of all other parameters using the procedure described above. In practice, we perform a golden-section search for δ by minimizing the GMM objective function $G(\theta)$ in (9). We start from a wide interval $(0.001, 0.999)$ and terminate the search algorithm when the length of the search interval falls below the tolerance level of 10^{-5}.

Using this nested numerical procedure, we obtain point estimates of parameters β and δ. To compute standard errors clustered at the game-date level, we perform a block bootstrap that samples game-date pairs from the original sample with replacement (Efron and Tibshirani, 1994, p.86). We draw 50 bootstrap samples. For specifications in Section 4 where we do not estimate the persistence parameter δ, we do not bootstrap the estimates and instead obtain standard errors using

[30] We use Stata packages *reghdfe* and *ivreghdfe* that rely on an iterative algorithm that was proposed by Guimaraes and Portugal (2010) and was further optimized by Correia (2016). This algorithm relies on a simple fixed-point iteration principle whereby all regression coefficients are partitioned into groups (e.g., by the type of fixed effects), and the algorithm iteratively solves group-specific first-order conditions while fixing the values of coefficients in all other groups. Guimaraes and Portugal (2010) show that this algorithm converges to the correct least-squares estimates but manages memory more efficiently than other existing numerical implementations.



Appendix Figure C.2: **Distributed lag regression results.** This figure shows the coefficient estimates from the regression model (10). Vertical bars around point estimates indicate the 95% confidence intervals.

the standard asymptotic theory of the 2SLS estimators.

C Robustness Analyses

C.1 Nonparametric estimation of lagged effects

Our geometric decay model in (1) imposes that the concurrent and carryover effect have the same sign. However, live streams might distract viewers from playing the game, thus potentially creating a negative concurrent effect and a positive carryover effect. To explore this possibility, we estimate a flexible distributed lag model that does not impose this assumption. Specifically, we estimate the following model:

$$\log(1+\text{Players}_{jt}) = \sum_{\tau=0}^{30} \beta_\tau \log(1+\text{Viewers}_{j,t-\tau}) + \lambda_{j,d(t)} + \mu_{j,h(t)} + \eta_t + \varepsilon_{jt}. \qquad (10)$$

We include the same fixed effects as our main specification but model the number of concurrent players as a flexible function of lagged viewership counts. Although the model (10) does not nest our main model in (1) as a special case, it allows us to estimate a more flexible response function. Figure C.2 visualizes the estimates. Consistent with our main specification, we find that both the concurrent effect and the carryover effects are positive. In addition, the estimated effect quickly declines and dissipates almost to zero within about 8-10 hours. Therefore, we conclude that our

Appendix Table C.1: **Robustness analyses for the main specification in Section 3**

Specification	Elasticity $\hat{\beta}$	Persistence $\hat{\delta}$
(1) Main specification	0.027	0.712
(2) Main + game-week fixed effects	0.035	0.775
(3) Main + 6 lags in z_{jt}	0.027	0.712
(4) Main + 18 lags in z_{jt}	0.029	0.725
(5) Main + 10-minute periods t	0.023	0.910
(6) Main + drop games without z_{jt} variation	0.027	0.717
(7) Main + only non-zero players$_{jt}$ and viewers$_{jt}$	0.029	0.707
(8) Main + instruments z_{jt} only use streamers with highly variable schedules	0.029	0.733

geometric decay in Section 3 does not impose overly rigid assumptions on the signs of concurrent and carryover streaming effects.

C.2 Alternative specifications

We explore the robustness of our main results from Section 3 with respect to the (a) included fixed effects, (b) definition of the instrument z_{jt}, (c) definition of the time period t, and (d) sample definition. Table C.1 presents the estimates $\hat{\beta}$ and $\hat{\delta}$ for different specifications. Row 2 shows that our results are robust to controlling for game-week fixed effects. Our initial motivation for including game-date fixed effects was that they allow us to better control for unobserved game-specific events. With game-week fixed effects, we find a somewhat larger streaming elasticity $\hat{\beta} = 0.035$ compared to 0.027 in our baseline specification with game-date fixed effects. This contrast reaffirms that we need to control for day-to-day changes in the popularity of specific games to obtain consistent elasticity estimates. Nevertheless, the two estimates are similar in magnitudes, suggesting the selection bias is mild.

Next, in rows 3-4 we show that constructing the instrument z_{jt} using a different number of lagged values $\tilde{z}_{jt}$ affects our estimates only marginally. Row 5 shows how the estimates change when we switch from one-hour to 10-minute time periods t. Although this switch makes estimation computationally burdensome, it does not change our qualitative conclusions. We obtain the estimated elasticity of 0.023 and an implied hourly persistence parameter of $(0.910)^6 = 0.568$, somewhat lower than the estimates in our main specification. Row 6 shows that dropping games that are never broadcasted by the top 5% streamers (i.e., games that have zero variation in the instruments z_{jt}) yields estimates $\hat{\beta} = 0.027$ and $\hat{\delta} = 0.717$, which are similar to those in the main specification.

In row 7, we explore the robustness of our results with respect to the functional form assumptions on the log expressions $\log(1+\text{Players}_{jt})$ and $\log(1+V_{jt})$. When we remove games that have on average less than 10 concurrent viewers or less than 10 concurrent players, which typically means that both viewer and player counts equal zero most of the time, we obtain results similar to our main specification. Thus, our estimates are unlikely to be driven by the functional form assumptions.

Finally, in row 8, we re-estimate our main specification by constructing instruments z_{jt} based only on top streamers with sufficiently variable work hours. If streamers have predictable broadcast schedules, and if viewers tune into their live broadcasts at regular times of day, our elasticity estimates $\hat{\beta}$ may mistakenly capture the temporal within-day changes in game popularity. To explore this alternative explanation, we compute the average variance of start and end hours for each streamer, and select those streamers whose average variability of broadcast times is above the median. The average standard deviation of start and end times among these selected streamers is around 5.6 hours, implying they vary their broadcast times substantially across days. As Table C.1 shows, we obtain a streaming elasticity of $\hat{\beta} = 0.029$, similar to 0.027 in the main specification. We therefore conclude that our main results are not driven by the inclusion of streamers with relatively stable broadcast schedules.

D Estimating Heterogeneous Streaming Effects

D.1 Estimates from median sample splits

We start our heterogeneity analysis by subsampling games and comparing the estimated streaming elasticities across subsamples. The first two columns of Table D.2 present the estimated streaming elasticities $\hat{\beta}$ from the main specification in (1)-(2) for different subsamples of games. Throughout this section, we simplify estimation by fixing the persistence parameter at the level estimated in Section 3 ($\hat{\delta} = 0.712$).

In rows 1-4 of Table D.2, we estimate streaming elasticities by game age and publisher size. To this end, we define new games as those released within 2.7 years before our sample period. We additionally define small publishers as those who sell at most two games from our sample of 599 titles. Both definitions correspond to the median splits of variables "game age" and "publisher size." We find that new games have a somewhat higher estimated elasticity, although the difference is small and not statistically significant (elasticities of 0.031 vs 0.025). Nevertheless, we find that the games produced by small publishers benefit from streaming 2.1 times more than the games of large publishers (elasticities of 0.036 vs 0.017). This result is consistent with our conjecture that small publishers have modest advertising budgets and do not get the same media coverage as big conglomerates such as EA Games and Ubisoft. Consumers might therefore be unaware of these publishers' games, and Twitch broadcasts might be helping them break this awareness barrier.

Appendix Table D.2: **Streaming elasticities by game characteristics**

		Median Splits 2SLS		Generalized Random Forests	
		Estimate $\hat{\beta}$	S.E.	Average $\hat{\beta}_j$	Average S.E.
By game age:	(1) New games (<2.7 years old)	0.031	(0.004)	0.025	(0.015)
	(2) Old games ($\geq$2.7 years old)	0.025	(0.008)	0.019	(0.008)
By publisher size:	(3) Small publisher (1-2 games)	0.036	(0.008)	0.024	(0.016)
	(4) Large publisher (3+ games)	0.017	(0.004)	0.020	(0.008)
By price:	(5) Inexpensive (<$20)	0.034	(0.010)	0.024	(0.017)
	(6) Expensive ($\geq$$20)	0.022	(0.003)	0.020	(0.007)
By quality:	(7) High quality (Metacritic rating >80)	0.042	(0.010)	0.026	(0.005)
	(8) Low quality (Metacritic rating $\leq$80)	0.017	(0.004)	0.019	(0.014)
By rating variance:	(9) Niche games (rating std. >2.4)	0.042	(0.010)	0.027	(0.006)
	(10) Mainstream games (rating std. $\leq$2.4)	0.006	(0.002)	0.010	(0.009)

This table presents the estimates of streaming elasticities $\hat{\beta}$ for games with different characteristics, holding the persistence parameter δ at the level estimated in Table 3. The first two columns report the results from our main specification, whereas the last two columns show the average estimates $\hat{\beta}_j$ from GRF estimation for the same subsamples of games. All specifications include game-date, game–hour-of-day, and time fixed effects. Standard errors are clustered at the game-date level.

We further study how live streaming effects differ across games with different attributes. Rows 5-8 of Table D.2 show that streaming effects are higher for inexpensive games (i.e., priced below the median $20) and high-quality games. We estimate an elasticity of 0.034 for inexpensive games, about 50% higher than the elasticity for expensive games. Similarly, we obtain a higher estimated elasticity for high-quality games, defined as games whose Metacritic ratings are above the median. The estimated elasticity is about twice as high for high-quality than low-quality games (0.042 vs 0.017). These results suggest that live streams reveal information about the game's quality to consumers.

Twitch streams might also reveal the match value of the game, helping consumers understand whether the game matches their idiosyncratic preferences. For this reason, streaming elasticities might be higher for "niche" games that strongly appeal to some consumers but leave others indifferent. Using the standard deviation of user ratings to proxy "niche" games, we compare the estimated elasticity for games above and below the median value of this variable. Rows 9-10 of Table D.2 show niche games have a streaming elasticity of 0.042, much higher than the elasticity 0.006 among other games. This stark contrast supports our hypothesis that Twitch broadcasts reveal match value information to consumers.

We note that Table D.2 shows results of univariate analyses for one game characteristic at a time that do not control for other game characteristics. This explains why the estimated elasticities β in this table differ from those reported in Section 4.2, where we condition on all other game characteristics being at their average values.

D.2 Generalized random forests (GRF): Implementation details

The median splits in Section D.1 divide games based on somewhat arbitrary criteria. Additionally, splitting on each variable does not rule out the possibility that the effect is driven by other game attributes omitted from each pair of regressions. To address these concerns, we estimate heterogeneous streaming effects using GRF, a nonparametric estimation method proposed by Athey et al. (2019). The method estimates heterogeneous treatment effects as a flexible function of observables using a set of local moment conditions. The local moment conditions are weighted across nearby observations, and the weights are adaptively computed using the random forest algorithm (Breiman, 2001). By using these local moment conditions, this approach effectively generalizes our IV strategy. The method also generates estimates with known asymptotic distributions, enabling us to construct confidence intervals for the estimates.

We now provide additional details for the estimation procedure we use in Section 4 to study the heterogeneity in streaming elasticities. We generalize our main model in (1) by making the streaming elasticity β a function of game characteristics x_j:

$$\log(1+\text{Players}_{jt}) = \beta(x_j)\cdot\log(1+V_{jt}) + \lambda_{j,d(t)} + \mu_{j,h(t)} + \eta_t + \varepsilon_{jt}. \tag{11}$$

The vector of game attributes, x_j, includes five variables: game age, publisher size, regular price, critic rating (from Metacritic), and the standard deviation of consumer ratings.

To estimate a causal forest, we use the `grf` R package developed by Athey et al. (2019). We fix the persistence at the level estimated in Section 3, setting $\hat{\delta} = 0.712$, because the `grf` package does not admit nonlinear specifications. We estimate streaming effects $\beta(x_j)$ in equation (11) using local moment conditions:

$$\mathbb{E}\left(w_{jt}\cdot\varepsilon_{jt}|x_j = x\right) = 0, \tag{12}$$

where the error term ε_{jt} is defined as $\varepsilon_{jt} = \log(1+\text{Players}_{jt}) - \beta(x)\cdot\log(1+V_{jt}) - \lambda_{j,d(t)} - \mu_{j,h(t)} - \eta_t$.

The model is additive and separable in game-date, game–hour-of-the-day, and time fixed effects. Using this fact, we first demean the log player counts $y_{jt} \equiv \log(1+\text{Players}_{jt})$, weighted viewership counts $w_{jt} \equiv \log(1+V_{jt})$, and instruments z_{jt} from these three sets of fixed effects. To demean these variables from three sets of high-dimensional fixed effects, we use the alternating projection method proposed by Gaure (2013) and implemented in the `lfe` R package. Applying the Frisch-Waugh-Lovell Theorem, we then estimate the function $\beta(x)$ by using the following local moment conditions:

$$\mathbb{E}\left(\check{z}_{jt} \cdot \left(\check{y}_{jt} - \beta(x) \times \check{w}_{jt}\right) | x_j = x\right) = 0, \tag{13}$$

where $\check{y}_{jt}$, $\check{w}_{jt}$ and $\check{z}_{jt}$ represent the demeaned values of variables y_{jt}, w_{jt}, and z_{jt}. Equation (13) then represents a standard set of local moment conditions in Athey et al. (2019), which allows us to estimate streaming effects $\beta(x_j)$ using their `grf` R package. The `grf` package only permits one endogenous variable and one instrument. For this reason, we take the sum $z_{jt} = \sum_{\tau=t-12}^{t} \hat{\delta}^{t-\tau} \tilde{z}_{j\tau}$, where $\tilde{z}_{j\tau}$ is the number of top streamers broadcasting game j at time interval τ. With these instruments, we obtain qualitatively similar elasticities to those in the main specification (1), which we estimated using the entire vector of lagged top streamer counts $z_{jt} = (\tilde{z}_{j,t}, \tilde{z}_{j,t-1}, \ldots, \tilde{z}_{j,t-12})'$ as instruments.

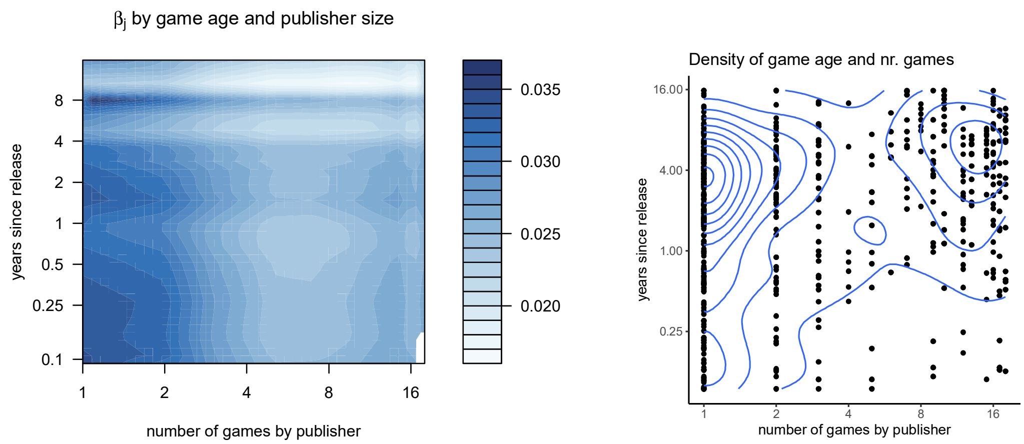
D.3 Tuning and Validation of GRF estimates

We validate GRF by comparing in-sample and out-of-sample estimates of streaming elasticities. We first obtain in-sample GRF estimates $\hat{\beta}_j$ under a given leaf size that determines the amount of regularization imposed by the model. We then also obtain out-of-sample analogs of these streaming elasticities. To this end, we split games into ten random blocks. We then predict elasticities β_j in each block by holding out this block, estimating GRF on the remaining data, and making an out-of-sample extrapolation to predict elasticities β_j for games in the held out block. Let $\hat{\beta}_{j,out}$ denote the resulting out-of-sample elasticity estimates. If GRF recover the true heterogeneity in streaming elasticities, then out-of-sample estimates $\hat{\beta}_{j,out}$ should strongly correlate with in-sample estimates $\hat{\beta}_j$. Conversely, if GRF overfit the data, in-sample estimates $\hat{\beta}_j$ will be overly affected by sampling noise, in which case the out-of-sample estimates $\hat{\beta}_{j,out}$ will be less correlated with in-sample estimates $\hat{\beta}_j$.

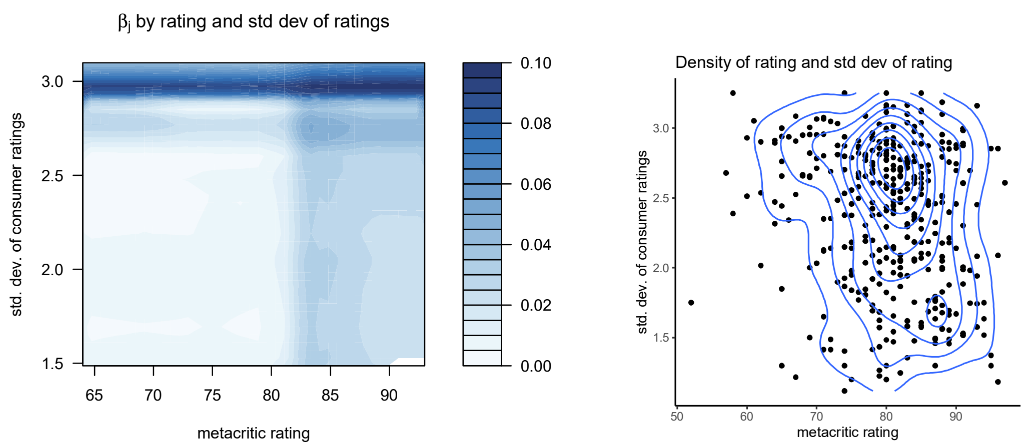
To visualize the two sets of estimates, in Figure D.1 we group games by quartiles of in-sample estimates $\hat{\beta}_j$ and plot the average estimated elasticities $\hat{\beta}_j$ and $\hat{\beta}_{j,out}$ within each quartile group. Within the range of tested leaf sizes ($n \times 5,472$ for $n = 1,2,4,8$), we find that out-of-sample estimates $\hat{\beta}_{j,out}$ increase across quartiles and that $\hat{\beta}_j$ and $\hat{\beta}_{j,out}$ are reasonably aligned.[31] Even in the most flexible model, we do not find any evidence of overfitting. We therefore fix the leaf size at $1 \times 5,472$ because it retains the highest degree of model flexibility without overfitting the data.

[31] We split trees using time-invariant game attributes x_j. We would ideally include all observations of each game in the same leaf, but due to the random splitting of trees in random forest algorithms, this ideal data split is not guaranteed. For model tuning, we consider leaf sizes $n \times 5,472$ with $n = 1,2,4,8$ where $5,472$ reflects the number of observations available for each game.

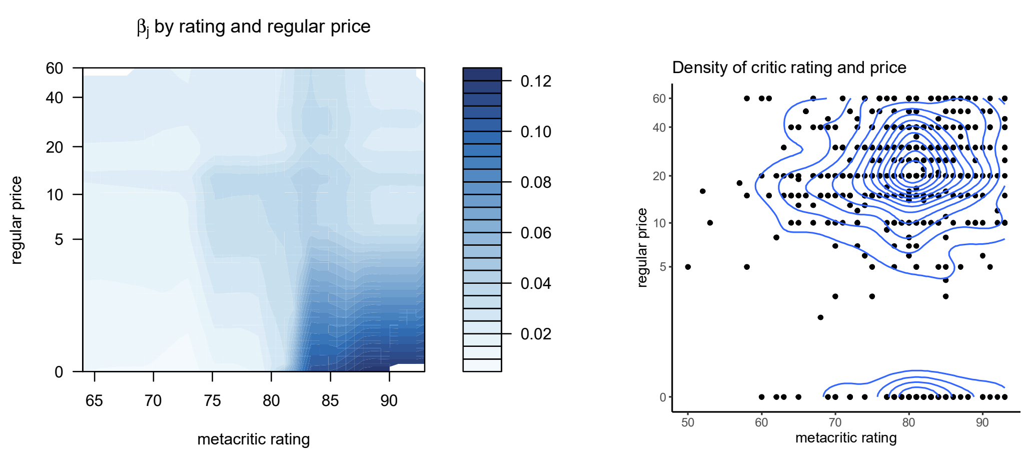
(A) By publisher size and game age



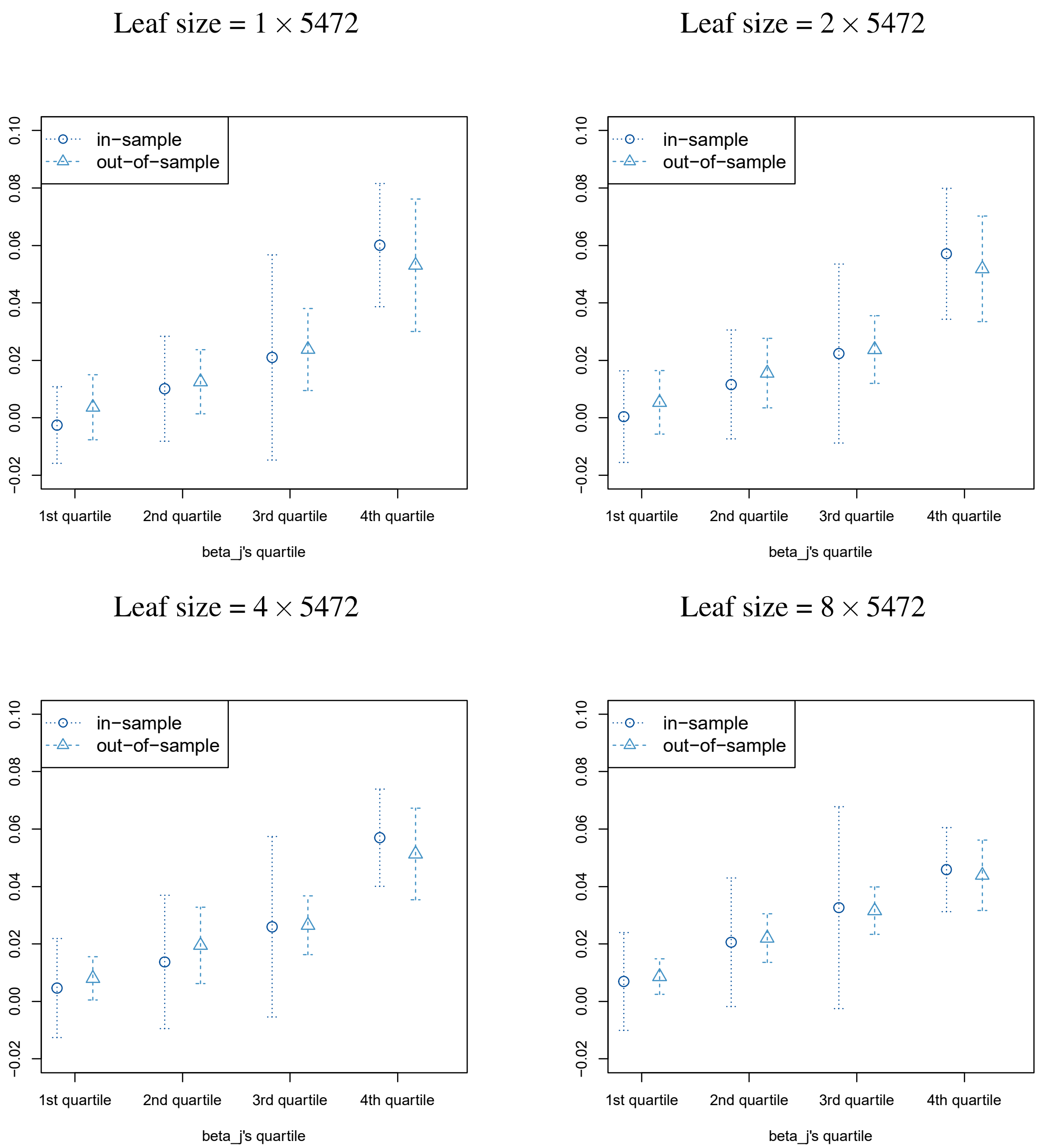
(B) By Metacritic rating and std.dev. of the consumer ratings



(C) By Metacritic rating and price



Appendix Figure D.2: **Estimated streaming elasticities from Generalized Random Forests.** The graphs on the left visualize the estimated streaming elasticities $\hat{\beta}(x_j)$ for two game attributes at a time, while holding all other attributes x_j fixed at their average levels. The figures on the right show the empirical distribution the relevant game attributes and display the contour lines of the estimated density functions.



Appendix Figure D.1: **Out-of-sample validation of GRF estimates.** Each graph splits games into four quartiles based on the distribution of in-sample estimates of streaming elasticities $\hat{\beta}_j$. In each quartile, we show the average in-sample estimates $\hat{\beta}_j$ and out-of-sample estimates $\hat{\beta}_{j,out}$ of streaming elasticities. To obtain out-of-sample estimates $\hat{\beta}_{j,out}$, we split games into ten random blocks, and we predict elasticities β_j in each block by holding out this block, estimating GRF on the remaining data, and making an out-of-sample extrapolation. These four graphs in this figure report results for leaf sizes $n \times 5,472$, where $n = 1,2,4,8$. In our main analysis, we fix the leaf size at 1×5472 for reasons detailed in Appendix D.3.

Appendix Table E.1: **Streaming elasticities of sponsored and partnered streams**

Parameter		Sponsored streams	Partner programs
		IV	IV
Organic stream elasticity	β^{org}	0.031	0.033
		(0.002)	(0.003)
Sponsored elasticity factor	ω	0.172	0.129
($\beta^{spons} = \omega \cdot \beta^{org}$)		(0.110)	(0.084)
Persistence parameter	δ	0.722	0.734
		(0.087)	(0.099)
Game-Date FE		Yes	Yes
Game–Hour-of-Day FE		Yes	Yes
Time FE		Yes	Yes
Observations		1,485,120	1,485,120

The table shows the estimates of β^{org}, ω, and δ from equation (5). To make the estimates of organic and sponsored elasticities comparable, we limit the estimation sample to 272 games that are sponsored on Twitch at least once. The first-stage F statistics are 62.9 and 67.1 for the log number of viewers in sponsored and non-sponsored streams, and 11.4 and 66.5 for the log number of viewers in partnered and non-partnered streams. Standard errors are clustered at the game-date level. All specifications include game-date, game–hour-of-day, and time fixed effects.

E ROI Computations: Additional Details

E.1 Estimating the effect of sponsored streams

In this Appendix, we provide the details of how we estimate the model (5) in Section 5.2. To simplify estimation, we assume a common persistence parameter δ for organic and sponsored streams, and we jointly estimate parameters δ, β^{org}, and ω by extending the GMM estimator in Appendix B.3. The vector of instruments z_{jt} includes (a) the numbers of top streamers broadcasting game j organically at $t, t-1, \ldots, t-12$ and (b) the numbers of sponsored top streamers broadcasting game j at $t, t-1, \ldots, t-12$.

One might worry that game publishers strategically choose which games to sponsor and when to sponsor them. To address this concern, in regression (5) we control for game-date fixed effects $\lambda_{j,d(t)}$. These fixed effects are allowed to freely correlate with the current viewership of organic and sponsored live streams, V_{jt}^{org} and V_{jt}^{spons}. For example, this specification allows publishers to only sponsor relatively unpopular games, and it allows for the possibility that publishers disproportionately sponsor games that are growing (or declining) in popularity. Both of these endogeneity concerns are captured by the model in (5) and do not introduce any omitted variable bias to the estimated parameters δ, β^{org}, and ω.

There is an important caveat to how we should interpret the estimated elasticities β^{org} and

β^{spons}. Some games never get sponsored, and therefore these observations do not influence our estimates of sponsored elasticities β^{spons}. To make organic and sponsored elasticities more comparable, we estimate equation (5) on the subsample of 272 games that are sponsored at least once in our data. Therefore, both estimated elasticities reflect the effect of live streaming on game usage for games that publishers decided to sponsor at some point. Publishers may also choose to sponsor games during periods when such sponsorships are most likely to be effective. Therefore, one should interpret the sponsored elasticity β^{spons} as reflecting the effect of sponsored live streams in the "best-case scenario" rather than for the average game in a typical time period. For this reason, we interpret β^{spons} as an upper bound on the average effect of sponsored streams on game usage.

This upper bound interpretation yields conservative estimates of sponsorship ROI, thus reinforcing our qualitative conclusions. Nevertheless, we emphasize that the reader should interpret our comparison of organic and sponsored elasticities as correlational rather than causal. We view the causal identification of these differences as a promising direction for future research.

E.2 Estimating the viewership increase from live streams

In Section 5, we estimate the ROI of each sponsored stream, which requires knowing how many viewers each sponsored stream attracts and how many of these additional viewers purchase the game. To understand how many viewers each sponsored and organic stream attracts, we estimate the following equation using OLS:

$$\text{Viewers}_{jt} = \theta_1 \cdot \text{NumSponsored}_{jt} + \theta_2 \cdot \text{NumOrganic}_{jt} + \lambda_{j,d(t)} + \mu_{j,h(t)} + \eta_t + \varepsilon_{jt}, \tag{14}$$

where fixed effects are the same as in equation (1). We obtain estimates $\hat{\theta}_1 = 2,559$ and $\hat{\theta}_2 = 2,288$, suggesting that a live stream by a top streamer increases the number of viewers by about the same amount regardless of whether it is sponsored or organic (see Table E.2). We also estimate an alternative specification where we restrict the effect of organic and sponsored streams to be the same, assuming $\theta_1 = \theta_2$ (see the second column of the same table).

E.3 Computing the lift in the number of players

We now present additional details for how we compute the lift in the number of players from sponsored streams, $\Delta\text{Players}_j$, in Section 5. We assume that. absent the sponsored stream, the game j is in a "steady state" with the number of players, Players_j^0, the number of viewers, Viewers_j^0, and the baseline stock of viewership V_j^0. We assume the baseline values of the first two variables are equal to the average number of players and viewers of that game in our sample. Because the number of viewers remains constant in this steady state, the baseline stock of viewership V_j^0 in (2)

Appendix Table E.2: **Impact of sponsored and organic streams on the number of viewers**

	Number of viewers OLS	Number of viewers OLS
Number of sponsored streams	2,558.8	
by top streamers (θ_1)	(373.8)	
Number of organic streams	2,288.4	
by top streamers (θ_2)	(59.0)	
Number of streams		2,281.4
by top streamers		(60.8)
Fixed effects:	game-date, time, game-hour-of-day	game-date, time, game-hour-of-day
Observations	1,487,304	3,257,904

Column 1 shows the estimates of parameters θ_1 and θ_2 from equation (14). The outcome variable is the absolute number of viewers of game j on Twitch. Standard errors are clustered at the game-date level. Column 2 shows the effect of the total number of top streamer broadcasts on the game's number of viewers.

can be computed as a geometric sum, so that $V_j^0(\hat{\delta}) = \text{Viewers}_j^0 \cdot (1-\hat{\delta}^{T+1})/(1-\hat{\delta})$.

The sponsored stream increases the number of viewers by $\hat{\theta}_1 = 2,529$ (column 1 in Table E.2), which increases the number of viewers to $\text{Viewers}_j^1 = \text{Viewers}_j^0 + \hat{\theta}_1$ and lifts the viewership stock to $V_j^1(\hat{\delta}) = \text{Viewers}_j^1 \cdot (1-\hat{\delta}^{T+1})/(1-\hat{\delta})$. This increase in the viewership stock in turn lifts the number of players with the sponsored elasticity $\beta_j^{\text{spons}} = \hat{\omega} \cdot \hat{\beta}(x_j)$, where $\hat{\omega} = 0.173$ is the sponsored factor from Table E.1 and $\hat{\beta}(x_j)$ is the estimated game-specific streaming elasticity obtained from the GRF estimation in Section 4. Given the structure of the model in (1)-(2), we have $\log\left(\left(1+\text{Players}_j^1\right) / \left(1+\text{Players}_j^0\right)\right) = \beta_j \log\left(\left(1+V_j^1\right) / \left(1+V_j^0\right)\right)$. After exponentiating both sides of this equation, rearranging terms, and computing $\Delta\text{Players}_j = \text{Players}_j^1 - \text{Players}_j^0$, we obtain the expression:

$$\Delta\text{Players}_j = (1+\text{Players}_j^0) \cdot \left(\left(\frac{1+V_j^1(\hat{\delta})}{1+V_j^0(\hat{\delta})} \right)^{\hat{\omega}\cdot\hat{\beta}(x_j)} - 1 \right). \tag{15}$$

This expression reflects the number of additional players brought into the game by the sponsored stream. We evaluate $\Delta\text{Players}_j$ for each game based on our estimates and use it to compute the ROI in equation (4).

E.4 The effect of Twitch on game sales

Live streams can affect the number of players in two ways: by inducing consumers to buy the game and by encouraging consumers to return to playing a game they already own. Because game publishers mainly earn revenue by selling new game copies, we need to distinguish between the two channels for ROI calculations. We disentangle the two channels by using supplementary consumer-level data from the Comscore Web Behavior Panel, described in Section 2.6. The main advantage of this dataset is that we observe which games individual consumers watch on Twitch and which games they buy on Steam. We use these data to assess whether watching a given game on Twitch makes consumers more likely to buy this game online within the same period of time.

E.4.1 Specification

Our main goal is to estimate the effect of watching a game on Twitch in a given hour t on the probability of browsing or buying this game online in the same hour. We focus on hourly analysis because it is the most consistent with our hourly analysis of streaming elasticities in Section 3.2. As we explain below, using hourly data also enables us to include highly granular controls, which helps address potential endogeneity concerns.

We estimate a model where the decision Y_{ijt} consumer i makes with respect to game j in hour t is given by:

$$Y_{ijt} = \kappa \times \text{WatchTwitch}_{ijt} + \theta_{ij} + \lambda_{j,d(t)} + \mu_{j,h(t)} + \eta_t + \nu_{ijt}, \tag{16}$$

where WatchTwitch_{ijt} is the indicator that the consumer watched at least one stream of game j in hour t, θ_{ij} are consumer-game fixed effects, $\lambda_{j,d(t)}$ are game-date fixed effects, $\mu_{j,h(t)}$ are game–hour-of-day fixed effects, and η_t are hour fixed effects. Depending on the specification, we use three outcome variables Y_{ijt}: (1) the indicator that the consumer i browsed game j *on Steam* in hour t by opening this game's information page; (2) the indicator that consumer i bought game j *on Steam* in hour t; or (3) the indicator that consumer i bought game j from any online retailer in hour t.

To reduce the computational burden, we limit the sample to 8,611 Comscore panelists who purchased at least one video game online, not necessarily one of the Steam games from our dataset. Even for this limited panel, it is not feasible to estimate the hourly specification (16) on the entire dataset because we would need to estimate the regressions using over 30 billion observations (8,611 consumers $\times$ 203 games $\times$ 730 days $\times$ 24 hours in a day). For this reason, we limit our sample only to consumer-date combinations on which a consumer viewed at least one Twitch stream of any game. We then estimate the model (16) on a balanced panel in which we observe consumer i's decisions Y_{ijt} in each hour of these selected days.

The OLS estimate of κ in (16) suffers from the same simultaneity bias discussed in Section 3. To address this concern, we mimic our empirical strategy in Section 3 by leveraging the variation in streamers' broadcast schedules. We first define each consumer's "favorite streamer" as the streamer this consumer watched the most times throughout the entire data period. We then instrument Twitch viewership indicator WatchTwitch_{ijt} with an indicator that this consumer's favorite streamer broadcasted game j on Twitch at least once in hour t. The idea behind this instrument is that consumers are likely to watch any game their favorite streamer broadcasts, implying that by watching this streamer's broadcasts, they might get exposed to a live-stream of game j. We make an identifying assumption that conditional on fixed effects in (16), the game broadcasted by the favorite streamer in a given hour t is conditionally independent from the unobserved factors ν_{ijt} that affect the likelihood of browsing or purchasing game j online.

This IV specification addresses several concerns. First, a consumer might watch a game on Twitch after having already planned to buy this game, which raises a reverse causality concern. For example, a consumer who plans to purchase a game might watch it on Twitch to learn more about its gameplay. Alternatively, a consumer who just purchased the game might watch it on Twitch to learn how to play it well. Our "favorite streamer" IV addresses these concerns by exogenously shifting whether a consumer watches game j on Twitch, thus allowing us to estimate the causal effect of Twitch viewership on game purchases. Additionally, one might worry that some consumers strongly prefer games of a certain genre, which makes them more likely to buy such games and watch them on Twitch. We address this concern by including consumer-game fixed effects θ_{ij} in regression (16), which allows for arbitrary taste heterogeneity as long as the game preferences of individual consumers do not change over time. Finally, as in our main specification (1), game-date fixed effects $\lambda_{j,d(t)}$ allow for the possibility that games rise and fall in popularity over time, which may simultaneously affect their sales and live stream viewership.

We note that this IV strategy recovers the local average treatment effect (LATE) of Twitch viewership on game purchases, i.e., the average effect among "compliers" who watch a game on Twitch if their favorite streamer broadcasts this game. Because in Figure (3) we find a sharp and sizable increase in total viewership when a top streamer broadcasts the focal game, we conjecture that many consumers are willing to watch whichever games top streamers decide to broadcast. As such, the set of compliers who watch whatever games their favorite streamer broadcasts might not be highly selected. Nevertheless, if compliers happen to be more active gamers, one would expect Twitch broadcasts to have a larger effect on the compliers than the average viewer, which would make our conversion rate estimate optimistic.

E.4.2 Estimation results

Column 1 in Table E.3 shows the first-stage regression estimates. The fact that the favorite streamer broadcasts game j on Twitch substantially increases the probability that the consumer watches this game on Twitch in the same hour. We find that watching game j on Twitch increases the probability of browsing the game on Steam by 0.56 percentage points (column 2) and increases the probability of purchasing it online by 0.16 percentage points (column 3). This last estimate implies that each 1,000 additional Twitch viewers generate, on average, 1.6 game purchases. To compare this effect with the average estimated streaming elasticity from Section 4, we focus on the same set of 203 games used for the regressions in Table E.3. The average estimated elasticity $\hat{\beta}(x_j)$ for these games is 0.0206, implying that each 1,000 additional Twitch viewers increase the number of players by around 4.2. It follows that $1.6/4.2 \approx 36.7\%$ of additional players brought to the game by Twitch streams are new to this game and need to purchase it. We use this 36.7% conversion rate in our ROI calculations in Section 5.

Appendix Table E.3: **The effect of Twitch viewership on the browsing and purchase probabilities**

	Watched This Game On Twitch (First Stage)	Browsed This Game On Steam (IV)	Purchased This Game On Steam (IV)	Purchased This Game From Any Retailer (IV)
Watched game j on Twitch today		0.0056	0.0016	0.0014
		(0.0016)	(0.0008)	(0.0008)
Favorite streamer of this consumer	0.0392			
broadcasted game j today	(0.0001)			
Consumer-Game FE	Yes	Yes	Yes	Yes
Game-Date FE	Yes	Yes	Yes	Yes
Game-Hour of Day FE	Yes	Yes	Yes	Yes
Hour FE	Yes	Yes	Yes	Yes
Unique Consumers	8,611	8,611	8,611	8,611
Observations	764.1M	764.1M	764.1M	764.1M

Column 1 presents the estimates from the first-stage regression of the model (16), in which the outcome variable is the indicator that consumer i watched game j in hour t on Twitch. We regress this outcome on the indicator that this consumer's favorite streamer broadcasted game j in hour t. In columns 2-4 we regress different outcome variables on the indicator that consumer i watched or game j in hour t on Twitch while instrumenting Twitch viewership with a "favorite streamer" IV described in Appendix E.4. The outcome variables Y_{ijt} are the indicators that the consumer browsed game j on Steam (column 2), purchased this game on Steam (column 3), and purchased this game from any online retailer (column 4) in that hour. Robust standard errors are reported in parentheses.