Supplementary Information for "A Simple Generative Model of Collective Online Behaviour"

James P. Gleeson, Davide Cellai, Jukka-Pekka Onnela, Mason A. Porter, and Felix Reed-Tsochas

SI1: Data Cleaning and Aggregate Installation Activity

The data was downloaded from Facebook for all existing 2720 applications ("apps") between 25 June 2007 (shortly after applications were introduced) and 14 August 2007 [1]. The data consists of time series $n_i(t)$, where $i \in \{1, 2, \ldots, 2720\}$, discrete time is indexed by the (real-time) hour $t \in \{0, 1, 2, \ldots, 1209\}$, and $n_i(t)$ corresponds to the aggregate number of users who have application i installed at time t. Data for 15 applications was corrupted, so we omitted these from our investigation and examined a total of $N = 2705$ applications. This data covers 100% of the population of 50 million potential app adopters and about 99% (2705 of 2720) of all applications that could be adopted. This thereby gives an almost complete view of system-wide adoptions during the time period of the data collection. We define the *launch time* t_i of app i as the smallest value of t for which $n_i(t) > 0$, and we define the *increment* in hour t for app i to be $f_i(t) = n_i(t) - n_i(t-1)$.

The data-cleaning process involves removing any undefined values within the data and imputing replacement values. For each app i, if $f_i(t)$ is undefined for $t > t_i$, then we copy the most recent well-defined increment value for app i into $f_i(t)$. A second cleaning step entails removing negative values of $f_i(t)$. Such values correspond to the (rare) cases in which deinstallations exceeded installations of an app in a given hour. We do this by setting any instances in the data with $f_i(t) < 0$ to $f_i(t) = 0$. The effects of the data cleaning are small in the context of the aggregate statistical characteristics of the data. In Fig. S1, the distribution function of the popularity at $t = t_{\max}$ for the cleaned data is shown in black. We show the corresponding function that uses the raw (pre-cleaning) $n_i(t)$ time series as red circles. The two distributions are almost indistinguishable, except for the smallest (i.e., least popular) apps, indicating that the cleaned data is very similar to the original data.

Figure S1 shows that the popularities $n_i(t)$ of the apps cover a range of scales from very small to extremely popular and that the distribution of n_i values is heavy-tailed. In Fig. S2, we show the total app installation activity $F(t)$, which is defined by Eq. (1) of the main text, of Facebook users during hour t. This function exhibits slow growth and 24-hour oscillations. We highlight these features by also plotting a linear growth function $A(t) = c_1 + c_2 t$ and (as a guide to the eye) a growing oscillation $A(t)\psi(t)$, where $\psi(t) = 1 + 0.5\cos(2\pi(t+8)/24)$ gives the oscillatory part of the function. Least-squares fitting gives $c_1 \approx 5.5 \times 10^4$ and $c_2 \approx 49$. Thus, by $t = t_{\max}$ (i.e., the end of the data-collection period), the mean hourly installation rate is approximately twice as large as it was at $t = 0$.

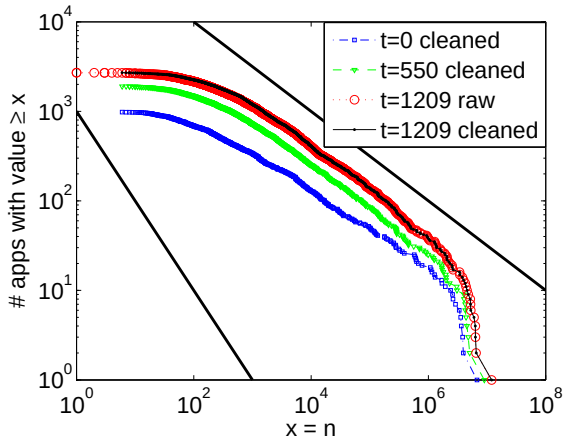

Figure S1: Distribution functions showing the number of applications ("apps") with popularity greater than or equal to n at times $t = 0$, $t = 550$, and $t = t_{\max}$. Red circles are from the raw data (i.e., prior to the cleaning process that we describe in Section SI1). The straight lines indicate the scalings corresponding to probability distribution functions with scalings $P(n) \sim n^{-\alpha}$ with $\alpha = 3/2$ (upper line) and $\alpha = 2$ (lower line).

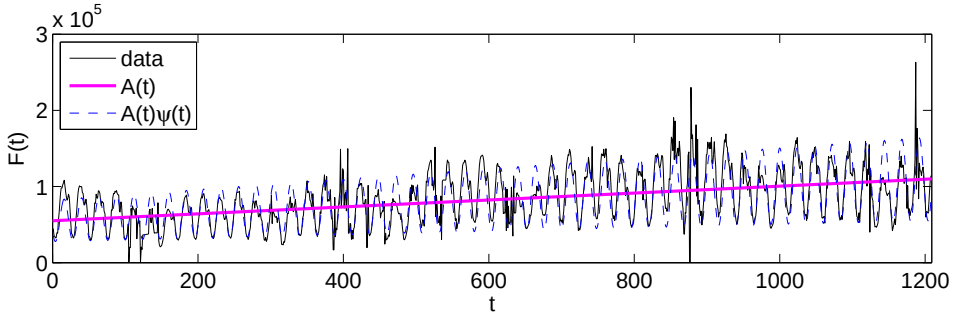

Figure S2: Total installation activity in hour t, as defined in Eq. (1) in the main text. We describe the linear growth function $A(t)$ and the 24-hour oscillation function $\psi(t)$ in Section SI1.

Rank	Name	C	D	E	θ
1	Likeness	1.59×10^5	1.48×10^{-2}	5.63×10^3	35
2	FunWall	4.25×10^7	5.26×10^{-5}	3.31×10^3	481
3	What's your stripper name?	4.10×10^4	1.12×10^{-2}	2.30×10^3	243
4	My Aquarium	4.44×10^3	2.85×10^{-2}	2.10×10^3	99
5	Vampires	3.14×10^4	2.58×10^{-2}	1.77×10^3	48
6	Harry Potter Magic Spells	3.10×10^4	8.30×10^{-3}	1.22×10^3	350
7	Pirates vs. Ninjas	7.65×10^3	1.23×10^{-2}	1.70×10^3	236
8	Booze Mail	7.22×10^3	9.95×10^{-3}	1.88×10^3	329
9	Superlatives	1.14×10^4	7.24×10^{-3}	1.50×10^3	402
10	Texas HoldEm Poker	1.10×10^4	8.24×10^{-3}	1.02×10^3	399

Table S1: Parameter values for the fitting functions $m(a)$ used in Fig. S3.

SI2: Top Ten Launched-Early-in-Study (LES) Apps

In Fig. S3, we show the ten most popular Launched-Early-in-Study (LES) apps. We order them by $\tilde{n}_i(t_{\rm LES})$, which denotes the number of installations by age $t_{\rm LES} \equiv 650$. To highlight common features of app growth, we use the heuristic fitting function

$$m(a) = \begin{cases} C(e^{Da} - 1)\,, & \text{if } a \leq \theta, \\ C(e^{D\theta} - 1) + E(a - \theta)\,, & \text{if } a > \theta\,, \end{cases} \tag{S1}$$

where the parameters C, D, E, and θ are determined by least-squares fitting of $m(a)$ to $\tilde{n}_i(a) - \tilde{n}_i(0)$ for each app i. We give the values of these parameters in Table S1. The parameter values that we obtain are sensitive to the initial guesses that are used in the fitting routine, but it is nevertheless clear that most apps exhibit exponential growth in a *novelty regime* (i.e., when age $a < \theta$) followed by linear growth at later ages (i.e., $a > \theta$).[1]

In Fig. S4, we show the scaled age-shifted growth rates $\tilde{f}_i(a)/\tilde{\mu}_i$ for the three most popular LES apps and the mean scaled age-shifted growth rate $r(a)$ (as defined in Eq. (2) of the main text) for the set of top-20 LES apps. At large values of a, the function $r(a)$ is qualitatively similar to that of the full LES set in Fig. 1a in the main text, as it exhibits a "quasi-stationary" (i.e., constant plus 24-hour oscillations) behaviour. However, the small-a novelty regime is different in the two cases; this reflects differences in early-stage growth patterns. In particular, the most popular apps exhibit steadily growing popularity during the novelty regime. This is consistent with the exponential growth in Fig. S3, but it contrasts with the decrease in novelty experienced by the majority of apps in their early stages (and reflected in the $r(a)$ curve in Fig. 1a of the main text).

[1] The notable exception among the top 10 in terms of fitting quality is Harry Potter Magic Spells (the 6th most popular app).

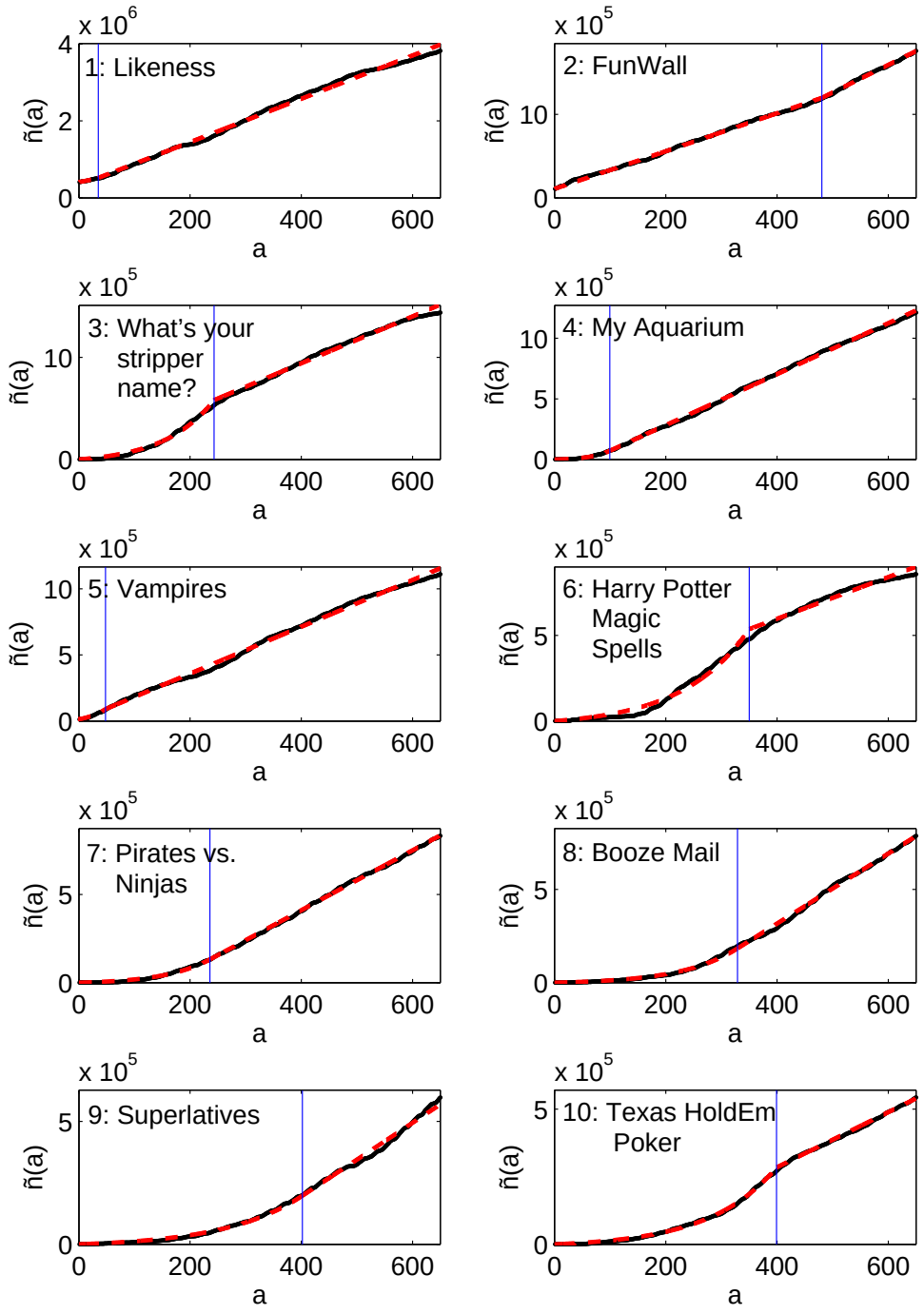

Figure S3: Growth trajectories of the 10 most popular LES apps, which we order according to their popularity when their age is t_{LES} hours. The data values are in black, and red dashed curves show the fitting function $m(a) + \tilde{n}_i(0)$ described in Eq. (S1) with parameter values from Table S1. The vertical lines mark the age θ at which the fitting function in Eq. (S1) changes from an exponential to a linear function.

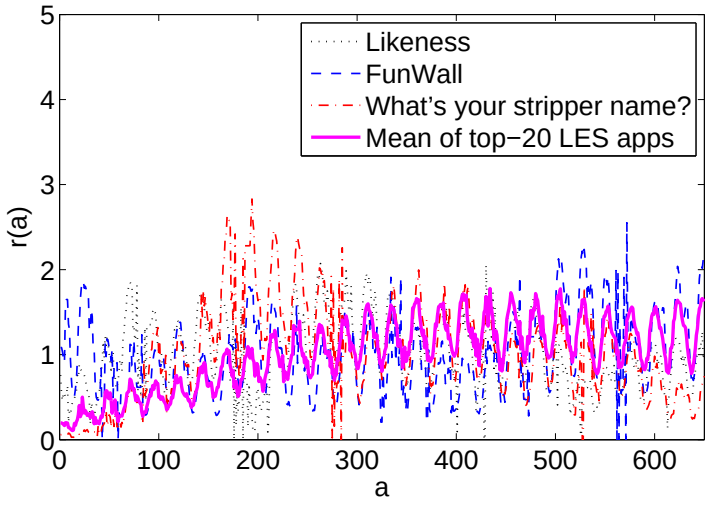

Figure S4: Scaled age-shifted growth rate functions $\tilde{f}_i(a)/\tilde{\mu}_i$ for the three most popular LES apps and the mean scaled growth rate of the 20 most popular LES apps.

SI3: Further Information on Figure 1 of the Main Text

SI3.1 Discussion of the L^2 Error in the Mean Scaled Age-Shifted Growth Rate

In Fig. 1a of the main text, we saw that the mean scaled age-shifted growth rate $r(a)$ for the entire LES set is similar to the corresponding $r(a)$ curves that we obtained by splitting the LES set into two disjoint subsets: the early-launch subset and the late-launch subset. To quantify the level of inherent diversity within the data, we calculate the L^2 norm of the difference between the $r(a)$ curves and call this the L^2 *error* of the partition:

$$E_{L^2} = \sqrt{\sum_{a=1}^{t_{\rm LES}} \left(r_{\rm LES}(a) - r_{\rm subset}(a)\right)^2}\,. \tag{S2}$$

For the aforementioned subsets, we find that the L^2 error is less than 3.11, and we take this value to represent a natural target for how well stochastic simulations can be fit to the data. In Fig. 3 of the main text, we showed all L^2 error values above 3.11 as dark red, and we concentrated on the light-coloured regions of the (H, T) parameter plane, as these constitute the locations where high-quality fits are possible.

We obtain similar results for any partition into two disjoint subsets of the same sizes as above. In Fig. S5, we again show the results of Figs. 1a,b of the main text, but we now also include curves for which we only use a subset (chosen uniformly at random and without replacement) that includes 460 of the LES apps. The randomly-chosen subset has very similar characteristics to the early-launch and late-launch subsets. Using 5000 realization of randomly drawn subsets of the same size, the mean L^2 error is 1.99 (with a standard deviation of 0.13.

In Fig. S6, we show the L^2 error E_{L^2} as a function of the memory time T for exponential memory function $W(\tau)$ for a fixed history window of length $H = 168$ and several values of the

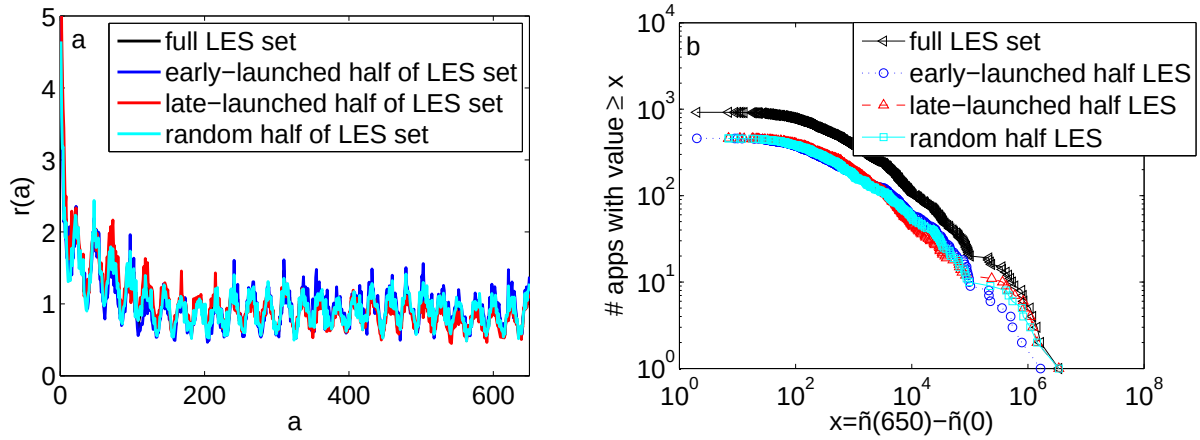

Figure S5: As in Figs. 1a,b of the main text, but including results for a randomly-chosen subset that includes 460 of the LES apps. (We chose the subset uniformly at random and without replacement.)

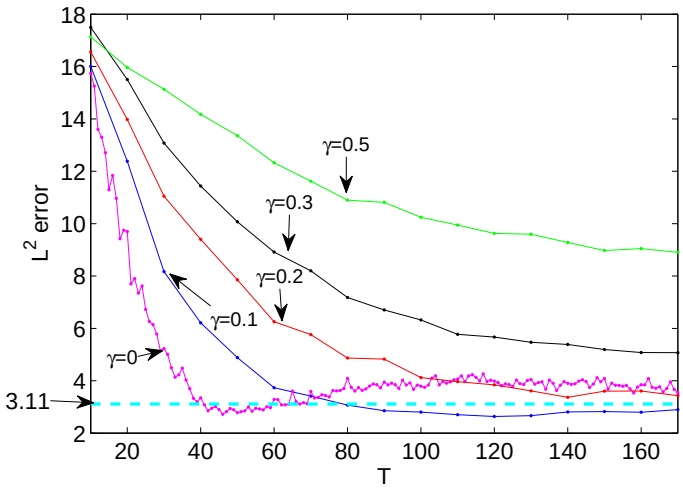

Figure S6: The L^2 error E_{L^2} in the mean scaled age-shifted growth rate $r(a)$ as a function of the memory time T for exponential memory function $W(\tau)$, history window $H = 168$, and several values of the parameter γ. Each point is the result of a single realization of a stochastic simulation of our model.

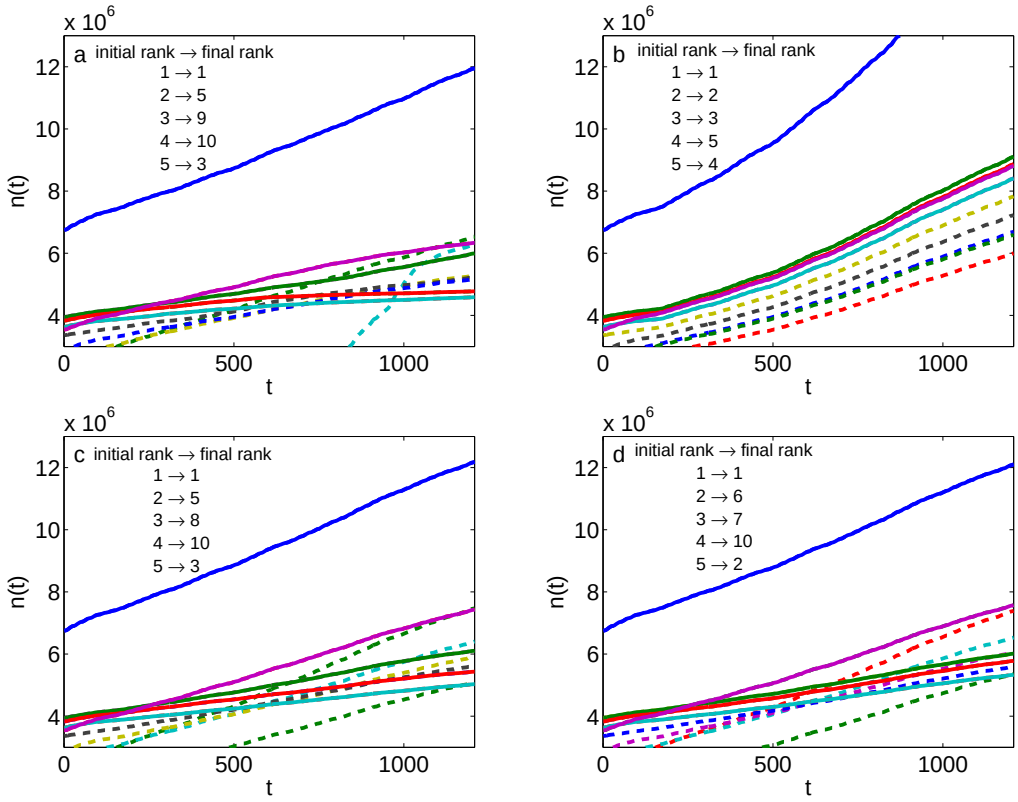

Figure S7: Growth trajectories of the 10 most popular apps in (a) data (as Fig. 1c), (b) simulation with cumulative-information model ($\gamma = 1$, as Fig. 1f), (c) simulation with recent-activity model with short memory (as Fig. 1i), (d) simulation with recent-activity model with long memory (as Fig. 1l). The solid curves show the popularities of the top-5 apps from $t = 0$; the dashed curves show the popularities of the remainder of the top-10 apps from $t = t_{\max}$.

parameter γ (see Fig. 3 of the main text). The dashed line indicates the threshold for the "good-fit" regime of $E_{L^2} \leq 3.11$. The error tends to increase with increasing γ and is unacceptably high for all values of T for $\gamma > 0.2$. It is interesting to note that the good-fit regime moves towards larger T values as γ increases. This seems to be a characteristic feature of the model—it appears also in Figs. S8 and S9—but we do not, as yet, have an explanation for it.

SI3.2: Turnover in the Top-10

The right column of Fig. 1 of the main text shows the popularity of those apps that are in the top-5 list at $t = 0$. Figure S7 shows more detail for each of the four cases (data plus three models) corresponding to Fig. 1c,f,i,l. In each panel of Fig. S7, the apps in the $t = 0$ top-5 are shown with solid lines, while dashed lines show the popularity of those apps that make up the remainder of the top-10 list at $t = t_{\max}$. As in Fig. 1, the change in rankings (turnover) is given in the legend of each panel, but here all apps in the top-10 are shown.

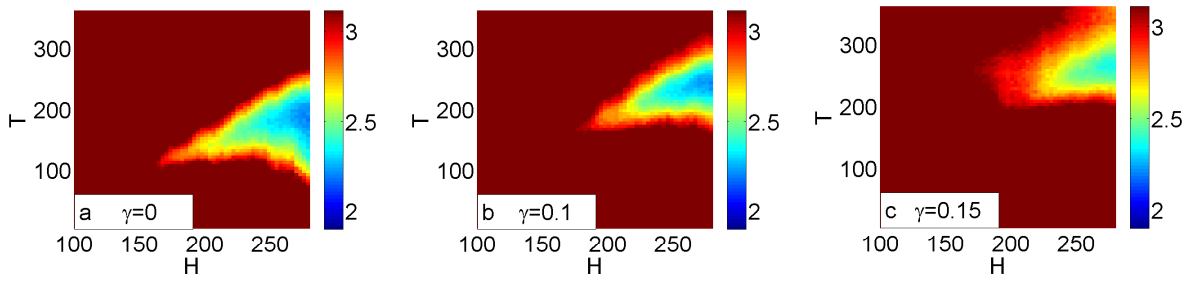

Figure S8: As in Fig. 3 of the main text, but now we use a memory function W that is generated from the uniform distribution (S3).

SI4: Response Functions Generating Memory Weighting

If one assumes that the total installation activity $F(t)$ is constant in time, then the memory function $W(\tau)$ introduced in Eq. (4) of the main text is proportional to the probability that an agent copies the installation choice of an agent from τ hours in the past (see SI5 for details). Consequently, we consider weighting functions that are related to previous empirical studies of the distribution of response times for e-mails [2–4]. Consider an update message that informs a Facebook user—which we model as a single computational agent—that a friend has installed a certain app, which can then lead to the user subsequently installing the app. Let τ' denote the time between receiving the update message and installing the app, and let $P(\tau')$ denote the probability distribution function (PDF) of these "response times" across the user population. We coarse-grain to the one-hour temporal resolution of the data by setting $W(\tau) = \int_{\tau-1}^{\tau} P(\tau')\, d\tau'$ (for $\tau \in \{1, 2, \ldots\}$), with an initial condition of $W(0) = 0$.

In the main text, we showed an example in which $P(t)$ is an exponential distribution. We now consider alternative assumptions on the underlying response-time distribution $P(t)$ and show results corresponding to Fig. 3 of the main text for the L^2 error in the mean scaled age-shifted growth rate. We find similar results for lognormal, gamma, and uniform distributions. In all of these cases, we obtain good results with a history window parameter of $H \approx 168$ hours (i.e., 1 week). Interestingly, when $H = 168$, the results for all distributions are very similar to those shown in Figs. 1j,k,l of the main text if the characteristic response-time $\langle\tau\rangle = \sum_{\tau=1}^{168} \tau\, W(\tau) / \sum_{\tau=1}^{168} W(\tau)$ is about 45 hours (i.e., approximately 2 days[2]).

In Fig. S8, we show results for the uniform distribution given by

$$P(t) = \begin{cases} \frac{1}{T}\,, & \text{if } t \le T\,, \\ 0\,, & \text{if } t > T\,, \end{cases} \tag{S3}$$

where T is the cutoff time. (The mean response time is $T/2$.) As with Fig. 3 in the main text, we show results in the (H, T) parameter plane to highlight the roles of both the history window H and the memory cutoff T. The three panels illustrate the effects of using increasing amounts

[2]The value of $\langle\tau\rangle$ is similar to the mean response time if most of the probability mass lies in the range $\tau < H$. The cutoff at $\tau = H$ reflects the fact that apps at early stages in their simulated growth possess a window of only approximately H hours of previous-installation history to drive their temporal evolution.

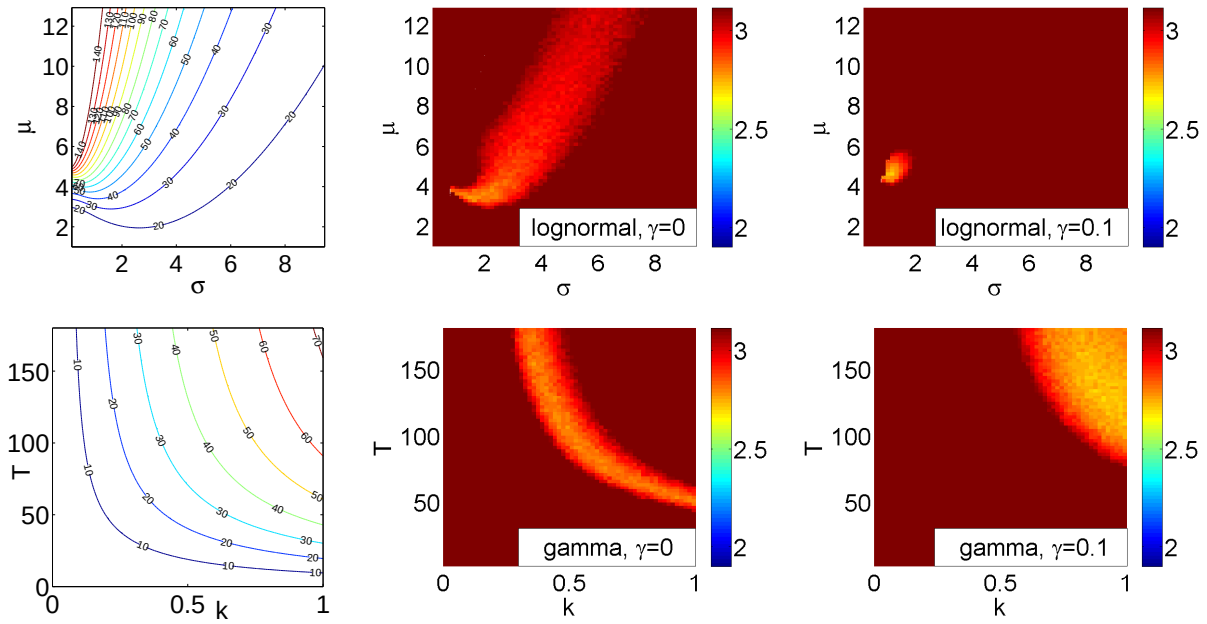

Figure S9: As in Fig. 3 of the main text, but now we use a (top row) lognormal distribution and (bottom row) gamma distribution of response times. The left panel in each row shows the contours of the cutoff mean response time $\langle\tau\rangle$, which we defined in Eq. (S4).

of cumulative information (i.e., progressively larger values of γ) in the installation probability p_i. Moving from left to right, the weighting of cumulative information increases from 0 to 0.1 and 0.15. As this weight increases, observe that the "good-fit" region of parameters moves to higher values of H and T. This supports our conclusion in the main text that the recent-activity case $\gamma = 0$ is "optimal" in the sense of requiring only a relatively small history window size H to fit the data. Similar conclusions were also reached in Ref. [5]. We have also confirmed that the other main results for the exponential distribution (e.g., the ones depicted in Fig. 1 of the main text) are closely reproduced using the uniform distribution (where we set $T \approx 100$ so that the mean response times are equal in the two cases).

In Fig. S9, we show results in which the response-time distribution $P(t)$ is given by (top row) lognormal and (bottom row) gamma distributions. These distributions both have two parameters, so we fix the history window size H to be 168 hours (i.e., 1 week) and consider the effect of the parameters that define the distributions. The lognormal distribution with parameters μ and σ is

$$P(t) = \frac{1}{t\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\ln t - \mu)^2}{2\sigma^2}\right\},$$

and the gamma distribution with parameters k and T is

$$P(t) = \frac{1}{\Gamma(k)T^k} t^{k-1} e^{-\frac{t}{T}}.$$

In the special case $k = 1$, the gamma distribution is an exponential distribution, while for $k < 1$

it limits to a power-law distribution as $T \to \infty$. The lognormal and gamma distributions were used in Refs. [2–4] to model distributions of e-mail response times.

The center panel of each row of Fig. S9 gives results for $\gamma = 0$, and the right panel of each row gives results for $\gamma = 0.1$. For $\gamma = 0.15$, the "good-fit" regions have almost disappeared from these plots, so we do not show them. The left panel of each row shows the contours of the quantity

$$\langle \tau \rangle = \frac{\sum_{\tau=1}^{168} \tau \, W(\tau)}{\sum_{\tau=1}^{168} W(\tau)} \, , \tag{S4}$$

which is related to the goodness-of-fit of the recent-activity ($\gamma = 0$) models. Observe that the light-coloured regions of the center panels align closely with the contours showing $\langle \tau \rangle$ values between 30 and 50 hours. Note that $\langle \tau \rangle$ is not identical to the mean response time of the distribution $P(t)$ because of the cutoff at 168 hours in the sums of Eq. (S4). This cutoff reflects the fact that the history window of 168 hours defines the τ range upon which the recent-activity model operates for an app that was launched recently. It seems that a memory weighting that corresponds to roughly 2 days (i.e., 48 hours) of recent activity is sufficient in all of these cases to fit the model to the data. A 2-day window was also identified as significant in the temporal clustering of adoption decisions among online friends in Ref. [6].

SI5: Recent-Activity Model as a Random-Copying-with-Memory Process

In this section, we show that one can interpret the recent-activity model ($\gamma = 0$) described in the main text as a random-copying process that is similar to those studied by Bentley et al. [7,8]. We also describe these models in terms of branching processes and discuss the circumstances under which one obtains critical branching processes. In this context, a critical branching process is one in which each parent has, on average, one child over its lifetime [9].

We consider a random-copying model in which each individual (an agent in our simulation) at time t copies the action (i.e., the choice of app to install) of an agent from a previous time step. In the schematic of Fig. S10a, we denote the copying action with an arrow from the earlier installation event to the later installation event (i.e., arrows point *from* the target of the copying *to* the copier). This generates a tree structure in time in which each node represents a single installation action and each arrow links a "parent" (target) node to some number of "child" (copier) nodes. Each child node has exactly one parent—this represents the installation action that was copied—but the number of children assigned to any given parent depends on the details of the random-copying process. As noted in the main text, we do not have any information on network topology, so we make the assumption that all agents can copy the action of any earlier agent, unrestricted by network connectivity.

There are $F(t)$ agents who install an app at time t, and they all act independently of each other. Consider one such agent Y, who must choose an earlier installation to copy. Let $\Phi(X, Y)$ denote the probability that Y copies the past action of a selected node X (see Fig. S10a). Normalization implies that $\sum_X \Phi(X, Y) = 1$, where the sum is over all possible target nodes X such that X takes an action before Y. We assume that the selection probability depends only on the time τ of the target node X and the time t of the action Y, so we write $\Phi(X, Y) = \Phi\left(X(\tau), Y(t)\right) \equiv \phi(\tau, t)$.

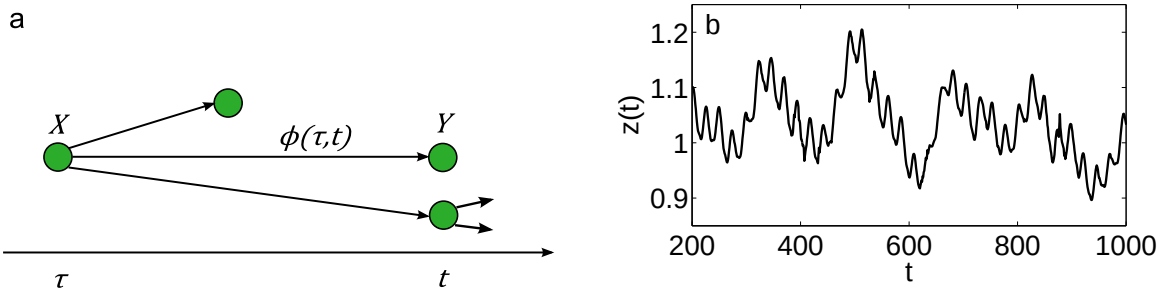

Figure S10: (a) Tree schematic for the model in Section SI5. Nodes indicate the installation of an app at the time indicated on the horizontal axis. (b) Effective branching number $z(t)$ for the data from Eq. (S8).

This implies that all installations at time τ are equally likely to be copied by Y. Moreover, we assume that the dependence on τ appears only through the time $t_e := t - \tau$ elapsed since the target event, so $\phi(\tau, t) \propto W(t-\tau)$, where W is the memory function (see SI4). Because there are $F(\tau)$ installing agents (i.e., nodes) at time τ, the correctly normalized copying probability must obey $\sum_{\tau<t} \phi(\tau, t)F(\tau) = 1$. This yields

$$\phi(\tau, t) = \frac{W(t-\tau)}{\sum_{\tau'=-\infty}^{t-1} W(t-\tau')F(\tau')}\,. \tag{S5}$$

Note we are allowing a potentially infinite history, which might be appropriate for very heavy-tailed memory-functions [10, 11].

Using this random-copying model, we want to compute the probability that user Y installs a given app i at time t. There are $f_i(\tau)$ agents who install app i at each time τ with $\tau < t$. (Installer X in Fig. S10a is just one example of many.) Agent Y can copy each of these agents with probability $\phi(\tau, t)$. Summing over all earlier times implies that the total probability that Y installs app i is

$$\sum_{\tau=-\infty}^{t-1} \phi(\tau, t) f_i(\tau)\,. \tag{S6}$$

Using the definition of $\phi(\tau, t)$, Eq. (S6) can be rewritten as

$$\frac{\sum_{\tau=-\infty}^{t-1} W(t-\tau) f_i(\tau)}{\sum_{\tau'=-\infty}^{t-1} W(t-\tau')F(\tau')}\,, \tag{S7}$$

which is precisely p_i^r in the main text, where we note that $f_i(\tau) = 0$ for $\tau < 0$ in Eq. (4) from the main text because data is available only from $t = 0$ onwards. The normalization constant L in

Eq. (4) can be written as

$$\begin{aligned} L &= \left(\sum_i \sum_{\tau'} W(t-\tau') f_i(\tau') \right)^{-1} \\ &= \left(\sum_{\tau'} W(t-\tau') \sum_i f_i(\tau') \right)^{-1} \\ &= \left(\sum_{\tau'} W(t-\tau') F(\tau') \right)^{-1} \end{aligned}$$

by reordering the summations and using Eq. (1) from the main text.

Returning to the branching-process interpretation of Fig. S10a, we calculate the expected number of children for each parent in the tree. Consider node X, which can be copied by any one of the $F(t)$ installing agents at time t. Each of these agents chooses to copy X with probability $\phi(\tau, t)$. Summing over t gives the expected number of children of node X (and indeed of any user at time τ) over all future times:

$$z(\tau) = \sum_{t=\tau+1}^{\infty} \phi(\tau, t) F(t) = \sum_{t=\tau+1}^{\infty} \frac{W(t-\tau)F(t)}{\sum_{\tau'=-\infty}^{t-1} W(t-\tau')F(\tau')} \,. \tag{S8}$$

This *effective branching number* $z(\tau)$ depends on the time label τ of the parent node (i.e., the time at which user X installed the app) because the interaction of the variable level of installation activity $F(t)$ with the memory function $W(\tau)$ implies that installations from some times are more likely to be copied in the future than installations from other times.

If $F(t)$ is constant, it follows that $z(\tau) = 1$ for all τ. (In this case, letting $s = t-\tau$ and $s' = t-\tau'$ gives $z(t) = \sum_{s=1}^{\infty} W(s) / \sum_{s'=1}^{\infty} W(s') = 1$.) Because each individual installation then has, on average, exactly one offspring, we obtain a critical branching process [9], for which one expects to obtain power-law distributions of popularity (with exponents $\alpha \in [3/2, 2)$) in the mean-field limit [12–14]. Consequently, the competition among apps for the finite number of installer slots leads to a critical branching process that is reminiscent of the self-organization mechanism in self-organized-criticality models [12, 15, 16]. Bentley et al. [8] used numerical simulations to examine this case of constant $F(t)$, though they did not give a branching-process interpretation. Note additionally that we concentrate on the accumulated popularity over time $n_i(t)$. In contrast, rewiring models such as those in Refs. [17, 18] focus instead on the distribution of short-time increments [similar to our $f_i(t)$].

As we show in Fig. S2, the total installation activity $F(t)$ exhibits substantial variation over time due to daily human activity patterns and to the aggregate growth in popularity of Facebook applications. In Fig. S10b, we show the effective branching number $z(t)$ calculated from Eq. (S8) using the $F(t)$ function taken from the data and the long-memory weighting function (i.e., an exponential response-time distribution with $T = 50$ hours) used in Figs. 1j,k,l of the main text. Despite the growth and fluctuations in $F(t)$ that are evident in Fig. S2, the values of $z(t)$ remain close to the critical value of 1 throughout the period of the study. This occurs because the memory of the weighting function W achieves a balance: it is sufficiently long so that it dampens the impact

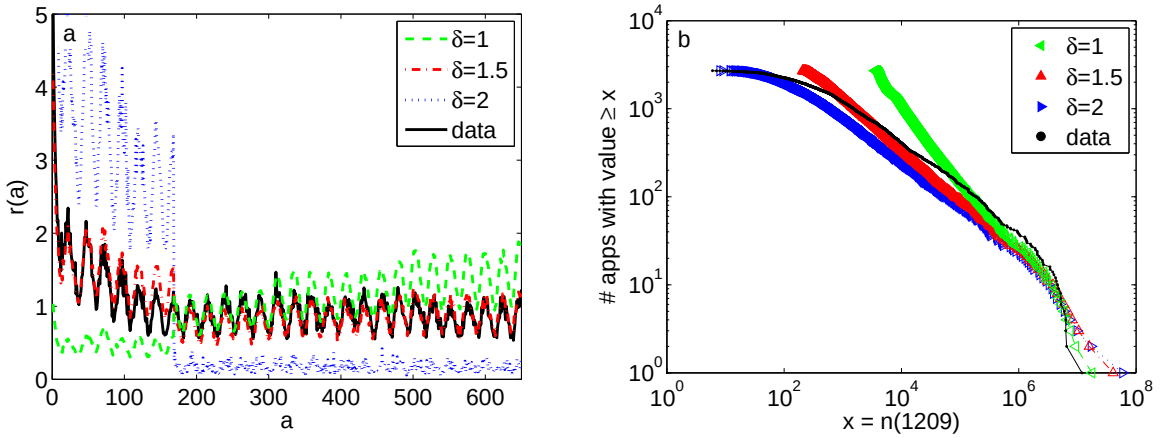

Figure S11: As in Fig. 1a,b of the main text, but now we use the ranking model described in Section SI6.

of daily oscillations on $z(t)$, but it is sufficiently short so that it also ameliorates the effect of the slow growth in $F(t)$ on $z(t)$. The resulting branching process is therefore near-critical [13], with an effective branching number between 0.9 and 1.2. This might help to explain the heavy-tailed popularity distributions that have been observed not only in this data set [1] but also in many other empirical data sets [19]. Recent models for the popularity of memes on Twitter have also been shown to be poised near criticality [16, 20].

SI6: Ranking Model

The ranking model introduced in Refs. [21, 22] suggests an alternative rule for how Facebook users might choose an app to install if they base their decisions only on a global listing of all apps according to their popularity. If an agent focuses only on the rank order of apps within the list and ignores the popularities (i.e., the numbers of installations) of the apps, then it is plausible that the probability of choosing app i at time t depends only on its ranking at time $t-1$. In the ranking model, this probability is

$$p_i^r(t) = \frac{r_i^{-\delta}}{\sum_j r_j^{-\delta}}\,, \tag{S9}$$

where r_i is the rank of app i at time $t-1$ and the quantity δ is a tunable parameter. For example, the second-ranked app ($r_i = 2$) is 2^δ times less likely to be chosen than the top-ranked app ($r_i = 1$). Such rich-get-richer dynamics is different to the linear preferential attachment mechanism of Eq. (3) of the main text, although it can also lead to power-law distributions of popularity [21, 22].

In Fig. S11, we show the results of replacing the cumulative rule of Eq. (3) from the main text with the ranking model rule (S9) while neglecting all recent information (i.e., putting $\gamma = 1$). For $\delta = 2$, the ranking model results are qualitatively similar to those of the linear preferential attachment case of Figs. 1d,e,f of the main text. Both models underpredict installations of LES

apps, so the $r(a)$ curve is too low at large ages. For $\delta = 1$, however, installations of (less-popular) LES apps are overpredicted by the ranking model, so the $r(a)$ curve in Fig. S11a is higher than the data curve at large a. In all cases—even $\delta = 1.5$, for which the fit to $r(a)$ is reasonably good—the distributions of app popularities differ dramatically from the data (see Fig. S11b). We conclude that the ranking model, like the cumulative-information model that we considered in the main text, cannot provide a good fit to the data on Facebook apps.

SI7: Cumulative-Rule Models Requiring Parameter-Fitting

In this section, we examine three extensions of the basic cumulative rule [see Eq. (3) of the main text]. Unlike the parsimonious models that we studied in the main text, each of the extensions that we now consider includes multiple parameters—one or more for each app—that need to be fitted from the available data. In order to make a fair comparison with the results that we presented in Fig. 1 of the main text, we use a history window of $H = 168$ hours to fit the parameters for each app, and we then implement a stochastic simulation using the appropriate version of the cumulative rule (with $\gamma = 1$ in all cases). In Fig. S13, we present our results for mean scaled age-shifted growth rates, distributions of app popularity, and turnover plots. They should be compared with Fig. 1 of the main text, as that figure shows the corresponding results for the models that we described in the main text.

In each of the three models that we describe below, we define the probability $p_i(t)$ that app i will be chosen by one of the $F(t)$ agents who install an app at time t. To allow the models to be fitted to the history-window data of each app, we need to make an important assumption. We assume that the actual number $f_i(t)$ of installers of app i at time t is equal to its expected value:

$$\text{Assumption 1:} \qquad f_i(t) = p_i(t)\, F(t)\,. \tag{S10}$$

This assumption is likely to be good when the mean number of installers $p_i(t)F(t)$ is large, but it can be inaccurate for unpopular apps that have small numbers of installations at time t.

To test the effect of Assumption 1, we calculate the exact installation probabilities $p_i(t) = f_i(t)/F(t)$ from the full data set and then insert these probabilities into a stochastic simulation (using a history window of $H = 168$). We show the results of this calculation in Fig. S12, from which it is clear that Assumption 1 does not cause the simulation results to differ appreciably from the data (see Fig. 1a,b,c). This test also provides an important check on our stochastic simulations: when the probabilities $p_i(t)$ are set correctly, it is evident that the data can indeed be accurately reproduced by our simulations.

We proceed to consider several models that are based on extensions of the cumulative rule. Each model is motivated by an example from the extensive literature on modelling heavy-tailed distributions [23–28]. In Sections SI7.1–SI7.3, we use $H = 168$ data points for each app to fit the parameters of the model. This gives a fair comparison with the situations that we considered in the main text. In Section SI7.4, we check whether we can obtain better results if we use *all* available data for model fitting. We conclude that the parsimonious recent-activity model of the main text gives superior performance to the alternatives that we consider in this section, as it can produce accurate results based on a history window of only 168 data points.

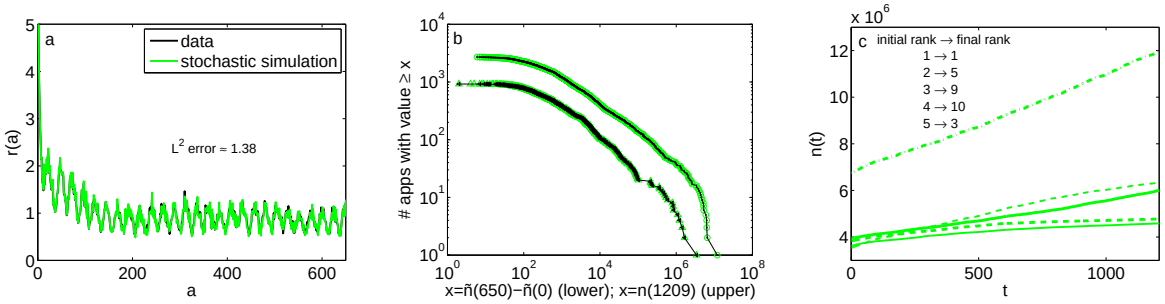

Figure S12: Results of using the exact installation probabilities $p_i(t) = f_i(t)/F(t)$ in stochastic simulators. The excellent match to data (compare these results to Figs. 1a,b,c of the main text) implies that any violations of Assumption 1 do not cause appreciable errors in the simulation results.

SI7.1: Cumulative Advantage with Heterogeneous Fitnesses

The first extension of the basic cumulative rule (see Eq. (3) of the main text) is based on the idea that a cumulative model supplemented with fitnesses can allow new entrants a head start [25]. To implement this idea, we replace the original cumulative rule by a refined version:

$$p_i^c(t) = \lambda_i \, K \, n_i(t-1) \,, \tag{S11}$$

where λ_i is the *fitness* of app i (cf. Section SI8) and the constant K is determined by the usual normalization: $\sum_i p_i^c(t) = 1$ [so $K = 1/\sum_i \lambda_i n_i(t-1)$]. As noted above, we use the history window (with $H = 168$) of data for each app to infer the values of the λ_i parameters and then run stochastic simulations based on the rule (S11).

To estimate the λ_i values for this model, we begin with the full data set (i.e., the exact values of $f_i(t)$ and $n_i(t)$ for all i and all t). If the rule (S11) were exact, then Assumption 1 would imply that

$$\begin{aligned} f_i(t) &= p_i^c(t) \, F(t) \\ &= \lambda_i \, K \, n_i(t-1) \, F(t) \\ &= \frac{\lambda_i \, n_i(t-1)}{\sum_j \lambda_j \, n_j(t-1)} F(t) \end{aligned} \tag{S12}$$

for all times t and all apps i. We can thus write the unknown λ_i values in terms of known quantities:

$$\frac{\lambda_i \, n_i(t-1)}{\sum_j \lambda_j \, n_j(t-1)} = \frac{f_i(t)}{F(t)} \,. \tag{S13}$$

Recalling from Eq. (1) in the main text that $F(t) = \sum_j f_j(t)$, we obtain a solution of Eq. (S13) by setting $\lambda_i \, n_i(t-1)$ equal to $f_i(t)/F(t)$ for each app i. Solving for the fitnesses then yields

$$\lambda_i = \frac{f_i(t)}{F(t) \, n_i(t-1)} \,. \tag{S14}$$

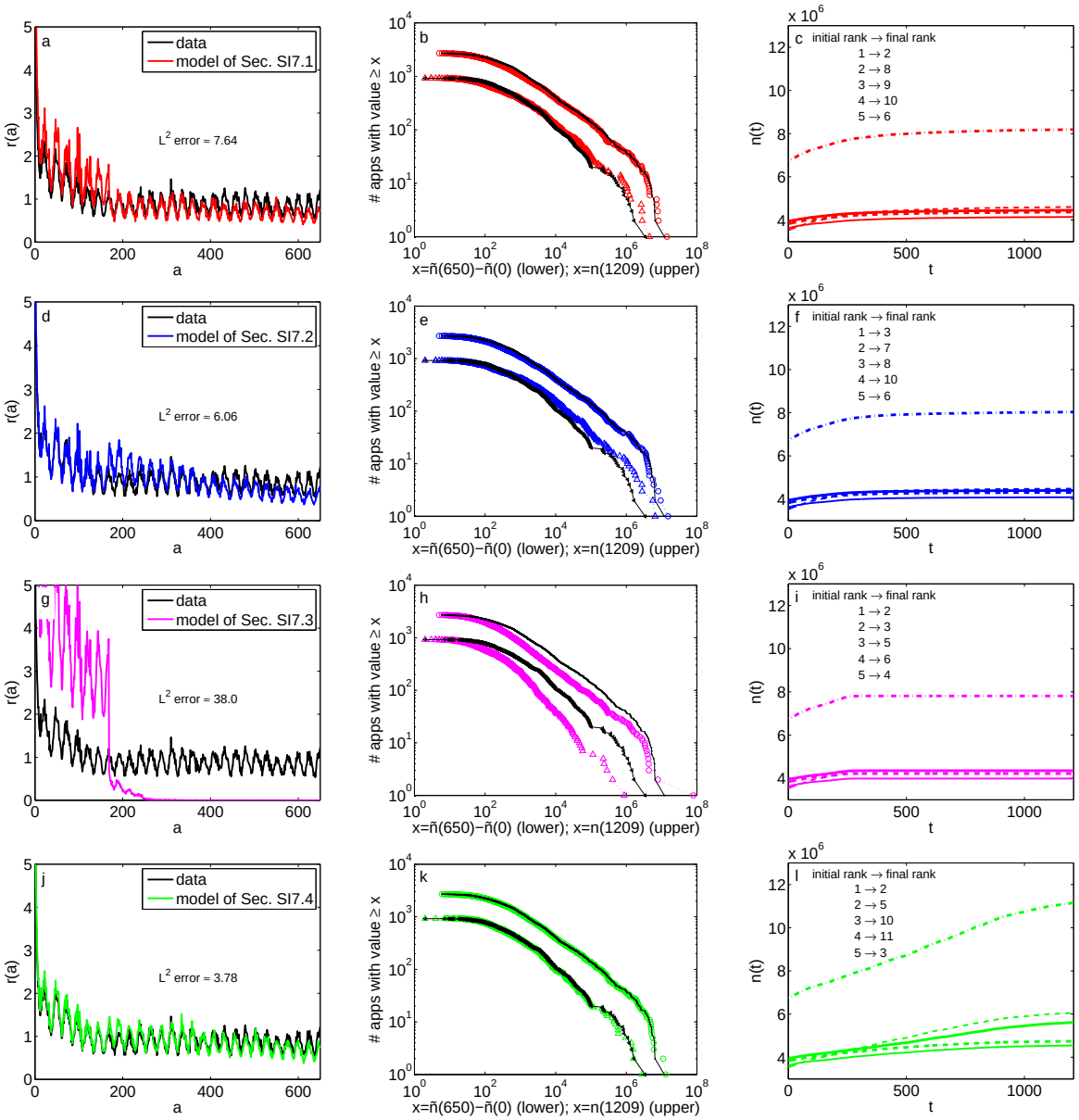

Figure S13: As in Fig. 1 of the main text (with $H = 168$), but now we show results for the extensions to the cumulative-information model that we examine in Sections SI7.1–SI7.4. (a,b,c) Results for the cumulative-information-with-fitness model of Section SI7.1. (d,e,f) Results for the cumulative rule with fitness and novelty decay (see Section SI7.2). (g,h,i) Results for the model with app-specific novelty decay (see Section SI7.3). Our parameter estimation for panels (a)–(i) uses 168 data points for each app. (j,k,l) Results for the model with app-specific novelty decay, but with parameters estimated using all available data (see Section SI7.4).

Because the model is not exact, the right-hand side of Eq. (S14) is not constant. To estimate the parameters in a manner consistent with the models that we study below, we sum both sides of Eq. (S14) over the history window of app i to obtain the relation

$$\lambda_i(T - t_i) = \sum_{t=t_i+1}^{T} \frac{f_i(t)}{F(t)n_i(t-1)} \quad \text{for } T \in \{t_i + 1, \ldots, t_i + H\}\,. \tag{S15}$$

We calculate the values of the right-hand side of this relation from the history-window data, and then estimate the parameter λ_i using least-squares fitting on the H data points.

In Figs. S13a,b,c, we show the results of using the rule (S11) with the fitness values inferred in the way that we just described. The turnover plot in Fig. S13c highlights the shortcoming of this model: the app that was initially most popular (and that continues to grow linearly in time in the real data, as illustrated in Fig. S12c and in Fig. 1c of the main text) has a sudden decrease in installation rate as soon as it exits its history window. By comparing with the benchmark case of Fig. S12c, we identify the reason for this loss of popularity: the inferred fitnesses of many other apps give installation probabilities of $p_i^c(t)$ at time $t = H + 1$ that substantially exceed their true probabilities from Eq. (S10). Because the total number of installing agents at each time step is restricted to be exactly $F(t)$, there is competition between the apps for the limited resource of agent attention. (Such competition has been examined in several data sets of online social networks [20, 29, 30].) Therefore, when many apps have installation probabilities that are too high, some other apps must suffer the consequence of fewer installations. In this case, the initially most-popular apps become victims of the intense competition. Apps that were initially less popular but have high fitnesses rise to take the top ranking by $t = t_{\max}$. It is clear that this model—despite having 2705 fitted parameters—does not do as well in reproducing the temporal behaviour of the data as the recent-activity model of the main text (see, e.g., Fig. 1l of the main text).

SI7.2: Cumulative Rule with Fitness and Novelty Decay

Wu and Huberman [26] examined data from the news web site `digg.com` and proposed a model that includes an age-dependent decay in the novelty value of stories. In our notation, their basic idea is a further refined version of the cumulative rule of Eq. (S11):

$$p_i^c(t) = \lambda_i\, K\, n_i(t-1)\, d(t - t_i)\,, \tag{S16}$$

where $d = d(a)$ is a decaying function of its argument (recall $a = t - t_i$ is the age of app i at time t, because t_i is its launch time) that models the loss of attractiveness due to novelty decay over time. As before, K is a normalization constant.

We begin by considering how to estimate the unknown parameters in Eq. (S16) using only the data for each app within its history window (with $H = 168$). Following the same steps as those leading from Eq. (S12) to Eq. (S14) yields the relation

$$\lambda_i\, d(t - t_i) = \frac{f_i(t)}{F(t)n_i(t-1)} \tag{S17}$$

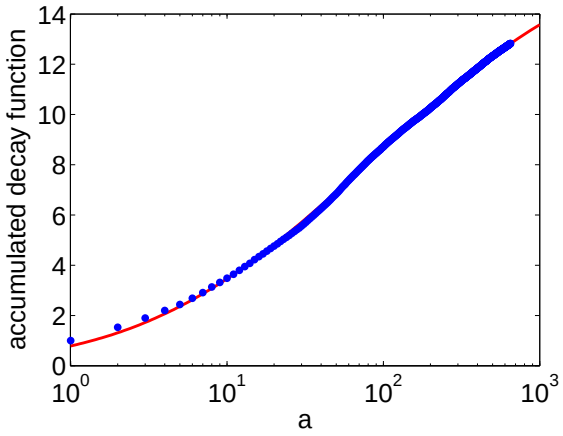

Figure S14: Accumulated novelty-decay function—i.e., the left-hand side of Eq. (S19)—as a function of age a fitted by $\lambda\Phi\left((\log(a)-\mu)/\sigma\right)$ with parameters $\lambda \approx 16.58$, $\mu \approx 4.47$, and $\sigma \approx 2.68$.

for each app i and for all times $t > t_i$. Because the novelty-decay function $d(a)$ is assumed to be the same for all apps (we will relax this assumption in Section SI7.3), it can be computed explicitly, up to a scaling factor, by averaging Eq. (S17) over all apps i in the LES subset $\mathcal{I}$ (see Eq. (2) of the main text). We thereby obtain

$$d(a) \propto \left\langle \frac{\tilde{f}_i(a)}{F(t_i+a)\tilde{n}_i(a-1)} \right\rangle_{\mathcal{I}} \qquad \text{for } a \in \{1,\ldots,H\}\,. \tag{S18}$$

We find that one can fit the novelty-decay function by a lognormal function of age. Specifically, Fig. S14 illustrates a successful fit to the accumulated decay function

$$\sum_{s=1}^{a} d(s) \propto \Phi\left(\frac{\log(a)-\mu}{\sigma}\right)\,, \tag{S19}$$

where $\Phi(x) = (2\pi)^{-1/2}\int_{-\infty}^{x} e^{-y^2/2}dy$ is the cumulative normal distribution and the parameters of the lognormal decay function are $\mu \approx 4.47$ and $\sigma \approx 2.68$. This form of novelty decay contrasts to the stretched exponential function fitted to data from `digg.com` in [26], but a lognormal decay function was successfully used in [27] to model the likelihood of a paper being cited at a time a after its publication (see Section SI7.3).

Now that we have estimated the novelty-decay function using the LES apps, we determine the fitness parameter λ_i for each LES app i by summing both sides of Eq. (S17) over the app's

history window and using Eq. (S19):

$$\lambda_i \sum_{t=t_i+1}^{T} d(t-t_i) = \sum_{t=t_i+1}^{T} \frac{f_i(t)}{F(t)n_i(t-1)} \quad \text{for } T \in \{t_i+1,\ldots,t_i+H\}\,,$$

$$\Rightarrow \lambda_i\,\Phi\left(\frac{\log(T-t_i)-\mu}{\sigma}\right) = \sum_{t=t_i+1}^{T} \frac{f_i(t)}{F(t)n_i(t-1)} \quad \text{for } T \in \{t_i+1,\ldots,t_i+H\}\,. \tag{S20}$$

Because we know the right-hand side of Eq. (S20) from the data, we can use least-squares fitting to the determine the best fit parameter λ_i for each app. Recall that for those apps that are not launched in the study window, we set $t_i = 0$; we examine the effects of this approximation in Section SI7.4.

Now that we have used the history window for each app to estimate the parameters for this model, we run stochastic simulations using rule (S16). We show our results in Figs. S13d,e,f. As we also saw for the model of Section SI7.1, we observe that the competition between the apps quickly causes the growth of the largest apps to deviate from their exact trajectories, leading to a turnover plot (see Fig. S13f) that is very different to that in the data (see Fig. S12c).

SI7.3: App-Specific Novelty Decay

Wang, Song, and Barabási recently proposed a cumulative-advantage model for the the number of citations that scientific papers garner over time [27]. In our notation, their model can be expressed in a manner similar to the Wu and Huberman model (see Section SI7.2), but with app-specific novelty-decay functions $d_i = d_i(a)$ replacing the universal decay function $d(a)$ of Eq. (S16):

$$p_i^c(t) = \lambda_i\, K\, n_i(t-1)\, d_i(t-t_i)\,. \tag{S21}$$

Wang et al. used a lognormal function to describe the novelty decay observed in their data, and our analysis of LES apps in Section SI7.2 supports a similar choice for our study. We therefore assume that the $d_i(a)$ are lognormal functions with app-specific parameters μ_i and σ_i. All of the derivations of Section SI7.2 also hold for this model, with the consequence that we estimate the values of λ_i, μ_i, and σ_i from the data by least-squares fitting of the relation [compare to Eq. (S20)]

$$\lambda_i\,\Phi\left(\frac{\log(T-t_i)-\mu_i}{\sigma_i}\right) = \sum_{t=t_i+1}^{T} \frac{f_i(t)}{F(t)n_i(t-1)} \quad \text{for } T \in \{t_i+1,\ldots,t_i+H\}\,. \tag{S22}$$

As in Section SI7.2, we set $t_i = 0$ for those apps that are not launched within the study window (see Section SI7.4).

As an aside, we note that one can make the connection to the model in Ref. [27] explicit by taking the continuous-time approximation

$$\tilde{f}_i(a) \approx \frac{d}{da}\tilde{n}_i(a)\,, \tag{S23}$$

setting $F(t)$ to be constant, and replacing sums by integrals. Equation (S22) then becomes

$$\lambda_i\,\Phi\left(\frac{\log(a)-\mu_i}{\sigma_i}\right) = \int_0^a \frac{1}{\tilde{n}_i(a')}\frac{d}{da'}\tilde{n}_i(a')\,da'\,, \tag{S24}$$

and its solution

$$\tilde{n}_i(a) \propto \exp\left[\lambda_i \, \Phi\left(\frac{\log(a) - \mu_i}{\sigma_i}\right)\right] \tag{S25}$$

gives the popularity of app i at age a. Equation (S25) reproduces, up to an additive constant, Eq. (3) of Ref. [27].

Returning to the least-squares fitting of Eq. (S22), we estimate the 3×2705 parameters for this model, and then use our stochastic-simulation framework to make predictions. We show our results in Figs. S13g,h,i. As we have seen for the other extensions of the cumulative-information model, competition between apps (which is not considered in Refs. [26, 27]) amplifies any error in the fitting functions of the model. We conclude that none of these adaptations of cumulative-advantage models provide a generative mechanism that describes the Facebook apps data as well as the recent-activity model that we described in the main text.

SI7.4: App-Specific Novelty Decay Using All Data

As we noted in Sections SI7.2 and SI7.3, there are 980 apps that were launched prior to the study window and thus have unknown launch times. Throughout our work, we assume that $t_i = 0$ for these apps. It is possible that this assumption might adversely affect the fitting of the models that rely on age-dependent novelty decay, as the ages of some apps will be misrepresented. To check the impact of this assumption, we therefore recalibrate the model of Wang et al. [27] that we described in Section SI7.3 by using *all* available data for every app to estimate parameters rather than just the 168 hours used in Section SI7.3 as a priori information. To do this, we replace the set of T values in Eq. (S22) by $T \in \{t_i + 1, \ldots, t_{\max}\}$. The extra data is helpful for the model, as it enables it to perform much better in stochastic simulations—see Figs. S13j,k,l—although it is still not quite as accurate as the recent-activity model of Figs. 1j,k,l of the main text. The improvement in accuracy from using extra data implies that the inaccurate launch times do not prevent this model from fitting reasonably well to the data. However, the quantity of data required to estimate the parameters is much larger than the history window of 168 hours that suffices to produce good results for the recent-activity model of Figs. 1j,k,l. The model of Wang et al. also has many more fitting parameters than the model that we presented in the main text.

SI8: Recent-Activity Model With Heterogeneous Fitnesses

We now consider replacing the recent-activity rule [see Eq. (4) of the main text] with an alternative that includes a fitness parameter λ_i for app i. The refined recent-activity rule is

$$p_i^r(t) = \lambda_i L \sum_{\tau=0}^{t-1} W(t-\tau) f_i(\tau)\,, \tag{S26}$$

which is normalized so that $\sum p_i^r(t) = 1$. All else being equal, apps with higher fitnesses are more likely to be selected for installation than apps with lower fitnesses. Thus far for the recent-activity model, we have focused on the so-called *neutral-model* [19, 31] scenario, in which all fitnesses are equal (with $\lambda_i = 1$ for all i). Noting from Fig. 1k of the main text that some of the largest LES app popularities are underpredicted by the otherwise successful recent-activity, long-memory model

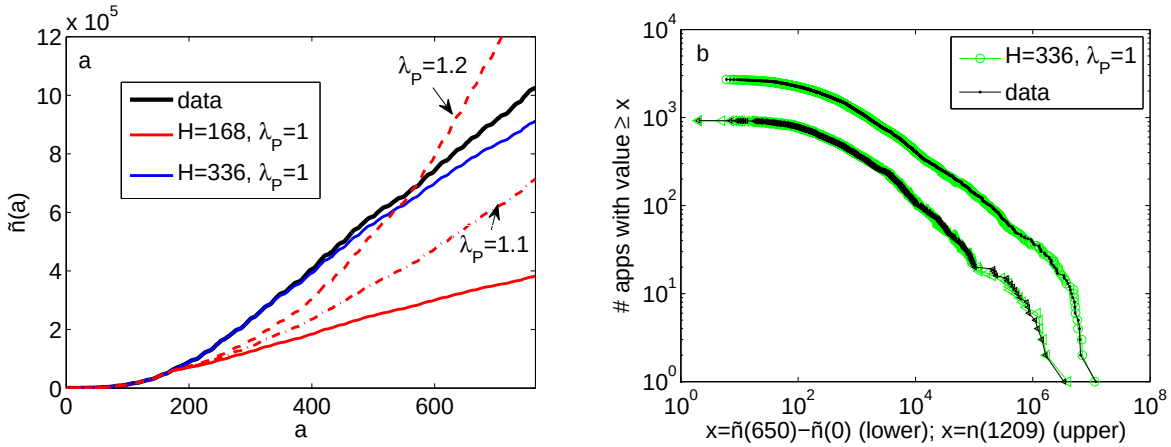

Figure S15: (a) Growth trajectory for "Pirates vs. Ninjas" from data (black) and from models (red and blue). (b) Popularity distributions as in Figs. 1e,h,k but for $H = 336$ hours (i.e., 2 weeks).

with homogeneous fitnesses (e.g., for $n \in [10^5, 10^6]$), it is natural to ask whether heterogeneous fitnesses might lead to a better fit to the data.

In Fig. S15, we show the growth of "Pirates vs. Ninjas", the 7th most popular (at age $t = t_{\rm LES}$) LES app (see panel 7 of Fig. S3). This is one of the apps in which the recent-activity, long-memory model of the main text with a 1-week history window gives an inaccurate prediction (solid red curve). This leads to notable differences between the popularity distributions of LES apps in Fig. 1h of the main text near $n = 10^6$. We thus consider changing the fitness of this particular app to a value $\lambda_P > 1$, while maintaining $\lambda_i = 1$ for all other apps. In Fig. S15a, we show the results of typical simulations using the dashed red curves. Although it is clearly possible to increase the popularity of this app by changing its fitness, we note that the $\lambda_P > 1$ trajectories exhibit increasing curvature, and the growth is super-linear in time rather than linear in time. For comparison, we also show results of an equal-fitness simulation in which we use a larger history window of 2 weeks (i.e., $H = 336$ hours) for all apps. In this case, the model's linear growth is much closer to the data, because the history window now includes the transition from novelty to post-novelty regimes (see Table S1 and the heuristic fit of Fig. S3) at about 236 hours (i.e., about 1.4 weeks). The plots in Fig. S15b confirm that using this longer history window leads to a much closer match between model and data.

We conclude that there does not appear to be strong evidence for heterogeneous fitnesses [as defined in our model through Eq. (S26)] among the apps, at least in the post-novelty regime. This conclusion is consistent with the findings of Bentley et al. regarding the applicability of the neutral model to other instances of choice among multiple alternatives [7] as well as with the experimental results of Salganik et al. [32], who showed that attractiveness of downloaded music is influenced more heavily by the actions of other downloaders than by the inherent quality of the music itself.

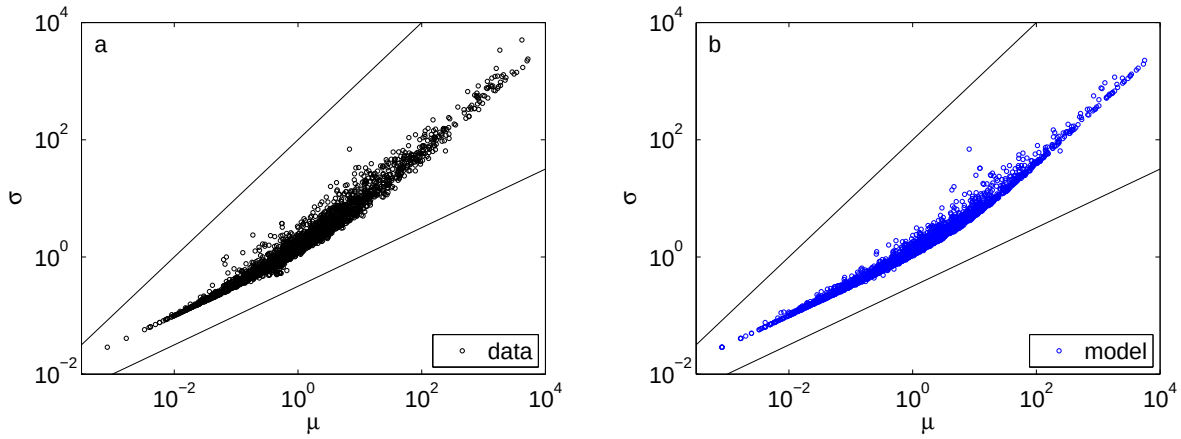

Figure S16: Fluctuation-scaling plots for (a) data and (b) the recent-activity, long-memory model described in the main text. The straight lines correspond to scaling exponents of $\beta = 1/2$ (lower line) and $\beta = 1$ (upper line).

SI9: Fluctuation-Scaling Relations

In Fig. S16a, we show a fluctuation-scaling (FS) plot of the Facebook apps data. As in Ref. [1], we calculate for each app i the mean μ_i and standard deviation σ_i of the increments $f_i(t)$ over times t from launch time t_i (recall we set $t_i = 0$ if the launch time is unknown) to the end of the data (i.e., $t = t_{\max}$). We then plot μ_i versus σ_i for all i to generate Fig. S16a. Reference [1] highlighted the existence of two FS regimes: the relation $\sigma_i \sim \mu_i^{\beta}$ with $\beta \approx 1/2$ is evident for small-μ_i apps, whereas a larger β value ($\beta \approx 0.85$) occurs for large-μ_i apps. In Fig. S16b, we show the corresponding FS plot for the simulated results from the recent-activity, long-memory model of Figs. 1j,k,l in the main text. Clearly, the plot is qualitatively similar to that of the data. In particular, it has scaling regimes with FS exponents of $\beta \approx 1/2$ for low-μ_i (i.e., low popularity) apps and $\beta \approx 1$ for high-μ_i (i.e., high popularity) apps. We now use our model to further analyze these two regimes. (The possible nature of the transition between these regimes is discussed in Ref. [1].)

As we discuss below, our model reveals that the $\beta \approx 1$ scaling of the large-μ_i apps is related intimately to the large diurnal oscillations in Facebook user activity. Recall that we represent such oscillations at the population level using the function $F(t)$. In simulations using non-oscillatory versions of $F(t)$, we find that the $\beta = 1/2$ regime extends to much larger values of μ_i, which suggests that the $\beta \approx 1$ regime in Fig. S16 appears because very popular apps exhibit coherent diurnal oscillations in their levels of installation activity. By contrast, small-μ_i apps receive a mean of fewer than 2 installations per hour, and their $f_i(t)$ time series appear similar to shot noise, for which one expects an FS exponent of $\beta = 1/2$.

In the top row of Fig. S17, we show FS plots for the data and the model with a slightly different way of calculating μ_i and σ_i from the one that discussed above. (Recall that we calculated μ_i as the temporal mean of the increments $f_i(t)$ from $t = t_i$ to the final time $t = t_{\max}$; we calculated the standard deviation σ_i similarly.) In Fig. S17, however, we instead begin the temporal averaging

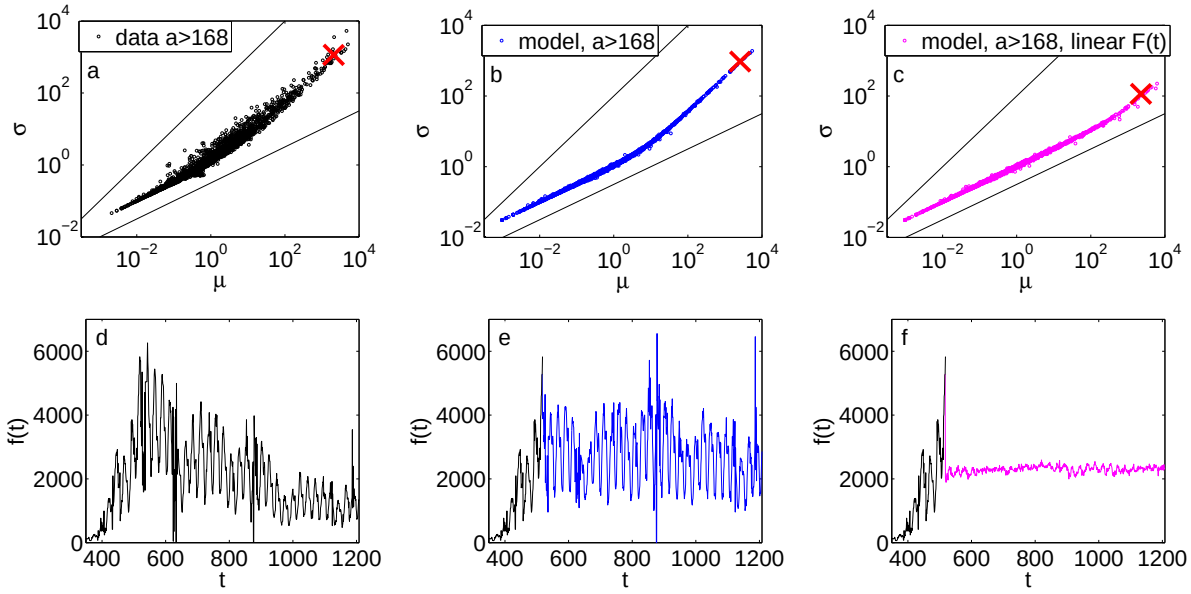

Figure S17: (Top row) Fluctuation-scaling plots using only values of a such that $a > 168$. The red crosses in the top row and the $f_i(t)$ time series in the bottom row correspond to the app "What's your stripper name?".

at $t = t_i + 168$. (If $t_i + 168 > t_{\max}$, then we drop this point from the plot.) This implies that we calculate the means and standard deviations only over ages from 1 week onwards, so we neglect the novelty regimes for most apps. Comparing Fig. S17a with Fig. S16a, we see that this change in definition of μ_i and σ_i does not strongly affect the FS plot of the data. However, as one can see by comparing Fig. S17b to Fig. S16b, the model results clearly are impacted by ignoring the novelty regime in calculating μ_i and σ_i. This arises from the relatively small fluctuations in the model for very popular apps. For example, the panels in the bottom row of Fig. S17 show the $f_i(t)$ time series for the app "What's your stripper name?" (see panel 3 of Fig. S3). In the data (Fig. S17d), the $f_i(t)$ time series decays slowly with the age of the app. However, the model does not reproduce this decay (see Fig. S17e), as it instead has a mostly unchanging envelope of $f_i(t)$ values in the post-novelty regime, and the fluctuations are due mainly to the aggregate activity level $F(t)$ that is input into the model. These fluctuations clearly give the main contribution to the standard deviation σ_i in our revised calculation. Indeed, the diurnal variations are inherited directly from the $F(t)$ function, and these fluctuations have the same order of magnitude as the mean. See, in particular, the function $\psi(t)$ in Fig. S2. This implies that $\sigma_i \sim \mu_i$ in this case, and it thereby yields an FS scaling exponent of $\beta = 1$.

We generate the third panel in each row of Fig. S17 using a further modification of our model: we replace the total activity function $F(t)$ that we input into the model with the linear growth function $A(t)$ from Fig. S2. This revised model has a total installation activity that grows linearly in time, but it does not experience the system-wide diurnal variations of the data. We still copy the $H = 168$ hour history window for newly-launched apps directly from the data (see the black curve in Fig. S17f). This introduces some residual 24-hour variation, but it is much less prominent than

in the model before modififcation. The resulting post-novelty standard deviations σ_i for popular apps are much smaller than in the other cases considered. Moreover, the scaling $\sigma_i \sim \mu_i^{1/2}$ holds for a much larger range of μ_i values. (Compare Fig. S17c to Fig. S17b.) We conclude that the high-μ_i scaling of $\beta \approx 1$ is connected intimately with diurnal variations in the activity levels of Facebook users.

References

[1] J.-P. Onnela and F. Reed-Tsochas. Spontaneous emergence of social influence in online systems. *Proceedings of the National Academy of Sciences of the United States of America*, 107(43):18375–18380, 2010.

[2] J. L. Iribarren and E. Moro. Impact of human activity patterns on the dynamics of information diffusion. *Physical Review Letters*, 103(3):38702, 2009.

[3] J. L. Iribarren and E. Moro. Branching dynamics of viral information spreading. *Physical Review E*, 84(4):046116, 2011.

[4] A. Vazquez, B. Racz, A. Lukacs, and A. L. Barabási. Impact of non-Poissonian activity patterns on spreading processes. *Physical Review Letters*, 98(15):158702, 2007.

[5] A. Zeng, S. Gualdi, M. Medo, and Y.-C. Zhang. Trend prediction in temporal bipartite networks: The case of MovieLens, Netflix, and Digg. *Advances in Complex Systems*, 16(4–5), 2013.

[6] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences of the United States of America*, 106(51):21544–21549, December 2009.

[7] R. A. Bentley, M. W. Hahn, and S. J. Shennan. Random drift and culture change. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 271(1547):1443–1450, 2004.

[8] R. A. Bentley, P. Ormerod, and M. Batty. Evolving social influence in large populations. *Behavioral Ecology and Sociobiology*, 65(3):537–546, 2011.

[9] T. E. Harris. *The Theory of Branching Processes*. Dover Publications, 2002.

[10] R. D. Malmgren, D. B. Stouffer, A. E. Motter, and L. A. N. Amaral. A Poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences of the United States of America*, 105(47):18153–18158, 2008.

[11] M. Karsai, K. Kaski, A. L. Barabási, and J. Kertész. Universal features of correlated bursty behaviour. *Scientific Reports*, 2(397), 2012.

[12] S. Zapperi, K. B. Lauritsen, and H. E. Stanley. Self-organized branching processes: Mean-field theory for avalanches. *Physical Review Letters*, 75(22):4071–4074, 1995.

[13] C. Adami and J. Chu. Critical and near-critical branching processes. *Physical Review E*, 66(1):011907, 2002.

[14] K. I. Goh, D. S. Lee, B. Kahng, and D. Kim. Sandpile on scale-free networks. *Physical Review Letters*, 91(14):148701, 2003.

[15] P. Bak. *How Nature Works: The Science of Self-Organized Criticality*. Springer, 1999.

[16] J. P. Gleeson, J. A. Ward, K. P. O'Sullivan, and W. T. Lee. Competition-induced criticality in a model of meme popularity. *Physical Review Letters*, 112:048701, 2014.

[17] T. S. Evans and A. D. K. Plato. Exact solution for the time evolution of network rewiring models. *Physical Review E*, 75(5):056101, 2007.

[18] M. Beguerisse-Díaz, M. A. Porter, and J.-P. Onnela. Competition for popularity in bipartite networks. *Chaos*, 20:043101, 2010.

[19] R. A. Bentley, M. Earls, and M. J. O'Brien. *I'll Have What She's Having: Mapping Social Behavior*. MIT Press, 2011.

[20] L. Weng, A. Flammini, A. Vespignani, and F. Menczer. Competition among memes in a world with limited attention. *Scientific Reports*, 2:335, 2012.

[21] S. Fortunato, A. Flammini, and F. Menczer. Scale-free network growth by ranking. *Physical Review Letters*, 96(21):218701, 2006.

[22] J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani. Characterizing and modeling the dynamics of online popularity. *Physical Review Letters*, 105(15):158701, 2010.

[23] M. V. Simkin and V. P. Roychowdhury. Re-inventing Willis. *Physics Reports*, 502(1):1–35, 2011.

[24] M. V. Simkin and V. P. Roychowdhury. A mathematical theory of citing. *Journal of the American Society for Information Science and Technology*, 58(11):1661–1673, 2007.

[25] C. Borgs, J. Chayes, C. Daskalakis, and S. Roch. First to market is not everything: An analysis of preferential attachment with fitness. In *Proceedings of the 39th annual ACM symposium on theory of computing*, pages 135–144. ACM, 2007.

[26] F. Wu and B. A. Huberman. Novelty and collective attention. *Proceedings of the National Academy of Sciences of the United States of America*, 104(45):17599–17601, 2007.

[27] D. Wang, C. Song, and A-L. Barabási. Quantifying long-term scientific impact. *Science*, 342:127, 2013.

[28] H.-W. Shen, D. Wang, C. Song, and A.-L. Barabási. Modeling and predicting popularity dynamics via reinforced Poisson processes. *arXiv:1401.0778*, 2014.

[29] K. Lerman, P. Jain, R. Ghosh, J.-H. Kang, and P. Kumaraguru. Limited attention and centrality in social networks. In *Proceedings of International Conference on Social Intelligence and Technology (SOCIETY2013)*. IEEE, 2013. arXiv:1303.4451.

[30] N. Hodas and K. Lerman. Attention and visibility in an information rich world. In *2nd International Workshop on Social Multimedia Research 2013, in conjunction with IEEE International Conference on Multimedia & Expo (ICME 2013)*. IEEE, 2013. in press (arXiv:1307.4798).

[31] O. A. Pinto and M. A. Muñoz. Quasi-neutral theory of epidemic outbreaks. *PLoS ONE*, 6(7):e21946, 2011.

[32] M. J. Salganik, P. S. Dodds, and D. J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856, 2006.