Online Appendix to

Selecting Cover Images for Restaurant Reviews: AI vs. Wisdom of the Crowd

A Performance of NIMA Model

In this appendix, we summarize the performance of the NIMA model for different choices of classifier, along with the performance of other image assessment models that are closely related to NIMA and report it in Table 1. The results in this table are based on testing the models using a large-scale database for aesthetic visual analysis (AVA) (Murray et al., 2012).

Table 1: Summary of notable classifiers tested with the NIMA model using the AVA database.

Approach	Input	Model	Output	Publicly available?	AVA accuracy
Murray et al. (2012)	Color and SIFT features	Support Vector Machines	Binary [High/Low]	No	66.7%
Kao et al. (2015)	GIST features	Support Vector Machines	Binary [High/Low]	No	65.2%
Kao et al. (2015)	Rescaled image	Deep CNNs	Average rating	No	71.4%
Lu et al. (2014)	Image patches (normalized)	Deep CNNs (RDCNN)	Binary [High/Low]	No	74.5%
NIMA(MobileNets)	Rescaled image	Transfer learning/ Deep CNNs	Rating distribution	Yes	80.4%
NIMA(VGG16)	Rescaled image	Transfer learning/ Deep CNNs	Rating distribution	Yes	80.6%
NIMA(Inception-v2)	Rescaled image	Transfer learning/ Deep CNNs	Rating distribution	Yes	81.5%
Ma et al. (2017)	Image patches (linked)	Deep CNNs (A-Lamb)	Binary [High/Low]	No	81.7%

The accuracy presented in this table is based on the common method used to measure the predictive performance for the AVA dataset through classification of photos into two classes (i.e., high and low aesthetic quality). Since the output of a model used for the AVA dataset comprises a quality score from 1 to 10, the photos with a predicted score (either as a probabilistic distribution or as a point estimate) closer to $(5 - \delta)$ are considered to be "low quality," while the photos with a predicted scores closer to $(5 + \delta)$ are considered to be "high quality." Finally,

the photos with aesthetic scores between (5 - δ) and (5 + δ) are also removed prior to running the experiments. The accuracy reported in this table is based on the standard case when δ = 0 and when the entire data is included in the evaluation.

B Deep-Learning Baseline Model Architecture

We denote by $\boldsymbol{y}$ a vector of target variables obtained from a predictive model $f(\boldsymbol{x})$ where $\boldsymbol{x}$ is an input vector (a set of features). In our case of the cover image scoring model, the output vector comprises the likelihood of image scores 1 to 10, i.e., $\boldsymbol{y} = (y_1, y_2, \ldots, y_{10})$ whereas an input $\boldsymbol{x}$ is an image of size 224×224 pixels. Note that, as in Talebi and Milanfar (2018), each original image is rescaled to 256×256 and then cropped to 224×224 pixels. The baseline image quality assessment model we leveraged in this work is a deep convolutional neural network model called MobileNets (Howard et al., 2017). The model consists of N neural network layers where each layer comprises multiple neural nodes. Let $f^{(n)}(\hat{\boldsymbol{x}}^{(n-1)})$ denote the approximate function of layer n of the deep-learning model which takes the input vector $\hat{\boldsymbol{x}}^{(n-1)})$ obtained from the output of the previous layer $n-1$. The target variables are obtained from the last layer of a multilayer function $\boldsymbol{y} = f^{(N)}(f^{(N-1)}(f^{(N-2)}(...(f^1(\boldsymbol{x})))))$ where the input of the first layer is the original input vector $\boldsymbol{x}$. At each layer, the function $f^{(n)}(\hat{\boldsymbol{x}}^{(n-1)})$ is described by a nonlinear activation function of an affine transformation function of the input vector, i.e., $f^{(n)}(\hat{\boldsymbol{x}}^{(n-1)}) = g^{(n)}(\boldsymbol{W}\hat{\boldsymbol{x}}^{(n-1)} + \boldsymbol{c})$ where $\boldsymbol{W}$ and $\boldsymbol{c}$ are the the learned weight and bias parameters of the affine function obtained from the training process of the model (Goodfellow et al., 2016).

In MobileNet, the model consists of a regular convolution as the first layer (denoted $f^n_{conv}(\hat{\boldsymbol{x}}^{(n-1)})$. This layer is followed by 13 sets of depthwise separable convolution blocks (denoted $f^n_{dwc}(\hat{\boldsymbol{x}}^{(n-1)})$ which comprises two types of convolutional layers, i.e., a deepwise convolution layer and a pointwise layer. These depthwise separable convolution blocks are then followed by a layer of average pooling, a fully connected layer and then the softmax function $\frac{e^{x_i^{(n-1)}}}{\sum_{i \in I} e^{x_i^{(n-1)}}}$ to transform the output vector into a vector of likelihood values.

The first convolution layer $f^n_{conv}(\hat{\boldsymbol{x}}^{(n-1)})$ can be described as $f^n_{conv}(\hat{\boldsymbol{x}}^{(n-1)}) = g^n_{ReLu}(g^n_{BN}(g^n_{conv}(\hat{\boldsymbol{x}}^{(n-1)})))$ where $g_{ReLu}(\boldsymbol{x})$, $g_{BN}(\boldsymbol{x})$ and $g_{conv}(\boldsymbol{x})$ are the retified linear unit (ReLu), batch normalization (BN) and standard convolution function, respectively. These functions are described by $g^n_{ReLu}(\boldsymbol{x}) = \max\{0, \boldsymbol{x}\}$, $g^n_{BN}(\boldsymbol{x}) = \boldsymbol{W}\frac{\boldsymbol{x} - E[\boldsymbol{x}]}{\sqrt{Var[\boldsymbol{x}] + \epsilon}} + \boldsymbol{c}$ and $g^n_{conv}(\boldsymbol{x}) = \boldsymbol{H}_{conv} \star \boldsymbol{x}$ where $E[\boldsymbol{x}]$ is the mean of input vector $\boldsymbol{x}$, $Var[\boldsymbol{x}]$ is the variance of $\boldsymbol{x}$, ϵ is a small constant (to ensure numerical stability), $\boldsymbol{H}_{conv}$ is the 3×3 convolution filter, and $\star$ is the discrete convolution operator (see Goodfellow et al. (2016)). We further define $g^n_{pw}(\boldsymbol{x}) = \boldsymbol{H}_{pw} \star \boldsymbol{x}$ and

by $g^n_{dw}(\boldsymbol{x}) = \boldsymbol{H}_{dw} \star \boldsymbol{x}$ where $\boldsymbol{H}_{pw}$ is the 1×1 convolution filter (Goodfellow et al., 2016) and $\boldsymbol{H}_{dw}$ is the depthwise convolution filer (Howard et al., 2017), respectively. A depthwise separable block f^n_{dwc} is defined as $f^n_{dwc}(\hat{\boldsymbol{x}}^{(n-1)}) = g^n_{ReLu}(g^n_{BN}(g^n_{pw}(g^n_{ReLu}(g^n_{BN}(g^n_{dw}(\boldsymbol{x}^{(n-1)})))))))$. Finally, the fully connect layer $f^{(n)}_{FC}(\hat{\boldsymbol{x}}^{(n-1)})$ is described by an affine function $f^{(n)}_{fc}(\hat{\boldsymbol{x}}^{(n-1)}) = \boldsymbol{W}\hat{\boldsymbol{x}}^{(n-1)} + \boldsymbol{c}$

The architecture of the MobileNets is provided in the Figure below.

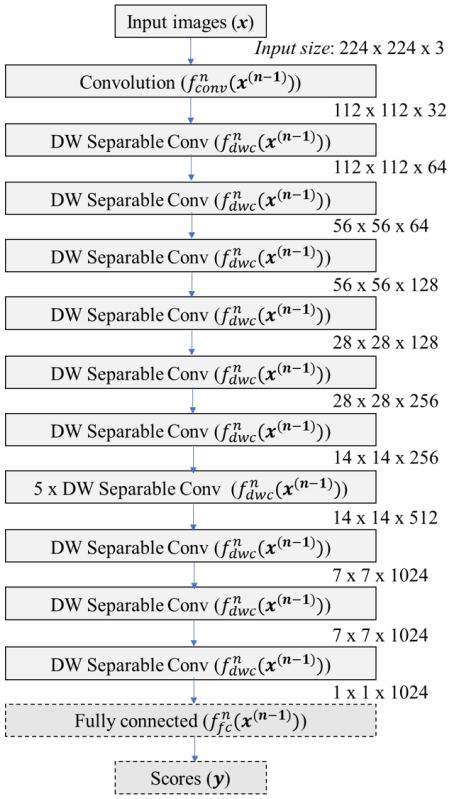


Figure 1: Illustration of baseline model, i.e., MobileNets (Howard et al., 2017) architecture. The last two blocks with dotted line are the layers which are replaced and retraining using transfer learning.

C Normalized Discounted Cumulative Gain Metric

Normalized discounted cumulative gain (nDCG) measures the ranking quality of the chosen cover images according to their relevance while taking into account the order in which each cover image appears. Denote by r_i, the graded relevance score (between 1 and 10) of cover image in position i determined by the human agent, and by p the number of maximum positions ($p = 10$ in our case). The discounted cumulative gain (DCG) is calculated as $DCG = \sum_{i=1}^{p} \frac{r_i}{\log_2(i+1)}$. We further define P' as the set of the selected cover images which are ordered by their relevance score in descending order (i.e., the case when the cover images are ordered perfectly), the *ideal* dis-

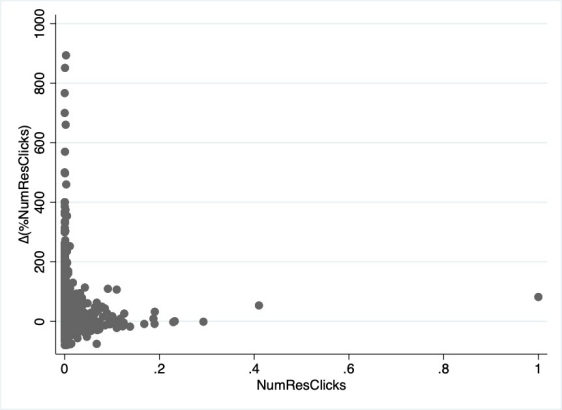
counted cumulative gain (IDCG) is calculated as $IDCG = \sum_{i \in P'} \frac{r_i}{\log_2(i+1)}$. The nCDG metric is computed as $nDCG = \frac{DCG}{IDCG}$. For example, if there are five positions ($p = 5$) and the cover image scores are as follows:

Position i	1	2	3	4	5
Score	5	10	7	0	1

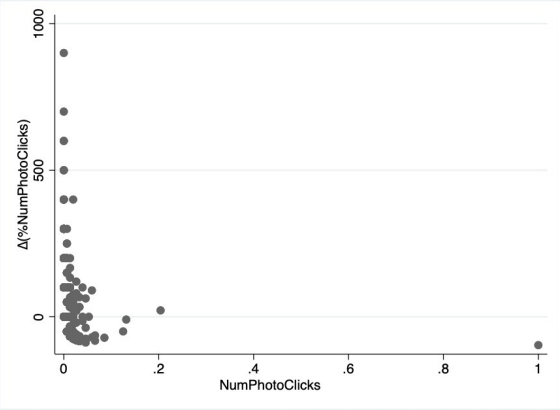
We can obtain $DCG = \frac{5}{\log_2(2)} + \frac{10}{\log_2(3)} + \frac{7}{\log_2(4)} + \frac{0}{\log_2(5)} + \frac{1}{\log_2(6)} = 15.196$, $IDCG = \frac{10}{\log_2(2)} + \frac{7}{\log_2(3)} + \frac{5}{\log_2(4)} + \frac{1}{\log_2(5)} + \frac{0}{\log_2(6)} = 17.347$ and $nDCG = \frac{15.196}{17.347} = 0.876$

D Additional Figures

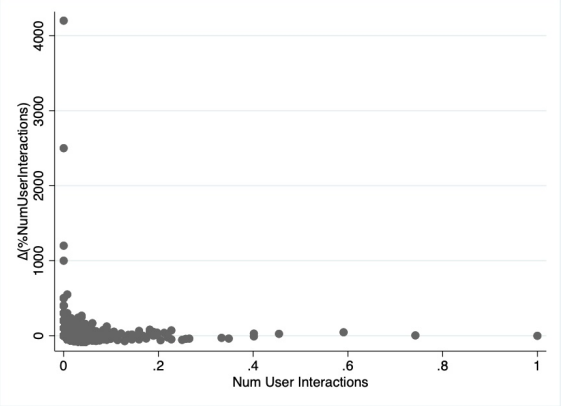
D.1 Distribution of Click Ratio Increases



(a) Restaurant clicks



(b) Photo clicks



(c) User interactions

Figure 2: The distribution of the increase in the click/interaction ratio

D.2 Distribution of Restaurants Characteristics

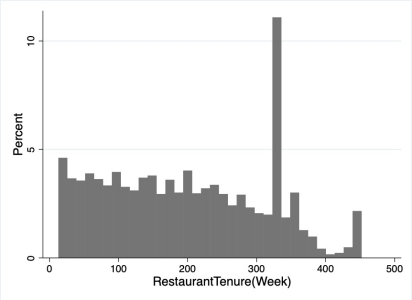


Figure 3: The distribution of restaurants with respect to their tenure.

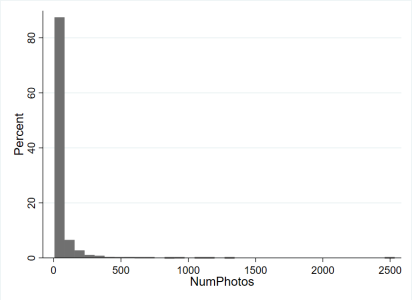


Figure 4: The distribution of restaurants with respect to the number of photos.

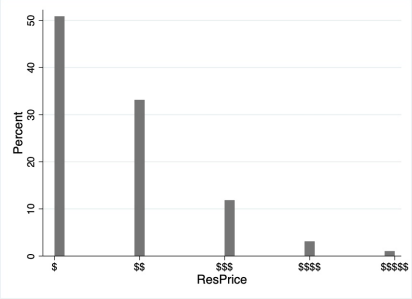


Figure 5: The distribution of restaurants with respect to their price.

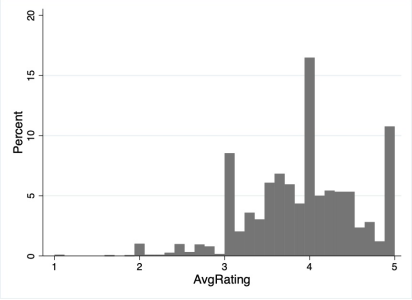
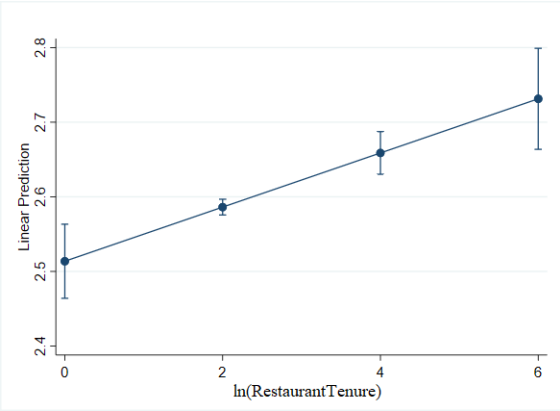
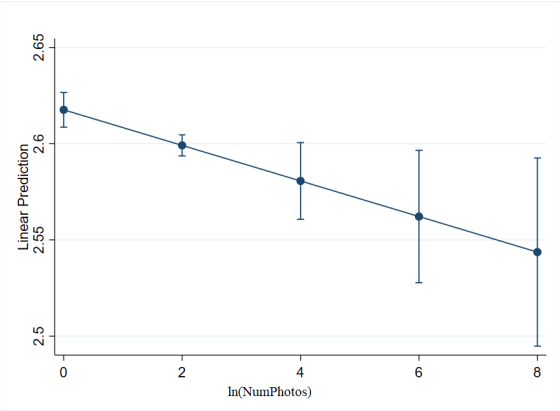


Figure 6: The distribution of restaurants with respect to their ratings.

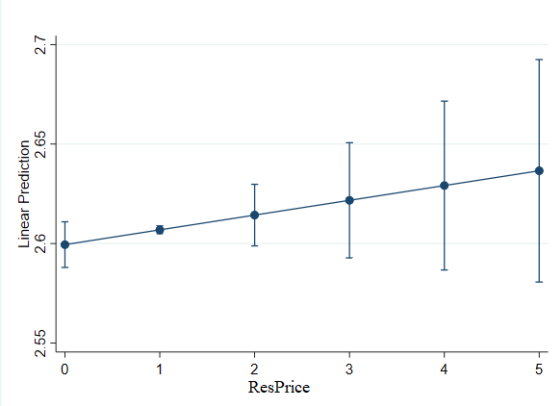
D.3 Marginal Effect of Interaction Terms



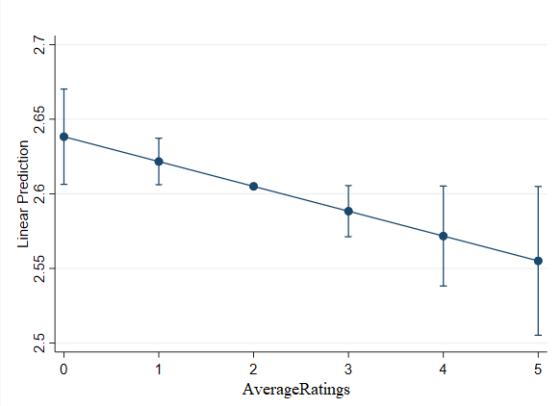
(a) Restaurant tenure



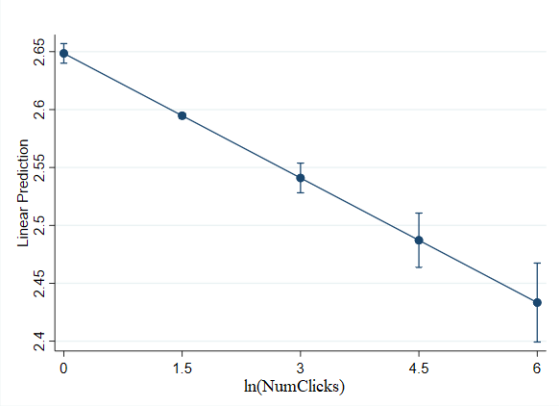
(b) Number of photos



(c) Restaurant price



(d) Restaurant ratings



(e) Number of clicks

Figure 7: Marginal effects of interaction terms

E Additional Results

E.1 Validating the Sample Representation

In this appendix, we validate that sample we randomly selected is representative of the entire population of restaurants on the platform. Here, we conduct a mean difference test (using the student t-test) and a distribution difference test (using the Kolmogorov–Smirnov test) between restaurants in our sample and all the restaurants on the platform based on the variables related to restaurant characteristics available to us. The results from the tests are presented in Table 2. Both tests (mean difference and distribution difference) readily confirm that none of the variables is statistically different at $p < 0.10$. Thus, these results confirm the representativeness of our random sample.

Table 2: Comparison tests for representativeness of our sample.

	Subsample (4,000 obs.)		Population (107,226 obs.)		Mean difference test (p-value)	Distribution difference test (p-value)
	Mean	Std. dev.	Mean	Std. dev.		
$RestaurantTenure$	201.018	114.963	203.570	114.321	0.166	0.666
$NumPhotos$	53.644	142.440	54.845	134.958	0.581	0.277
$ResRating$	3.916	0.641	3.919	0.635	0.734	0.807
$ResPrice$	1.722	0.876	1.735	0.888	0.379	0.973

E.2 Robustness Tests

In this appendix, we conduct three types of tests to showcase the robustness of our findings. First, we vary the time window used in our analysis. Second, we exploit two alternative identification strategies: (i) a regression discontinuity (RDiT) design and (ii) a combination of the propensity score matching (PSM) and difference-in-differences (DiD) design. Third, we perform placebo tests.

E.2.1 Excluding Restaurants that need Tie-Breaking

In our main analysis, 574 restaurants from 3,057 restaurants require a tie-breaking mechanism to choose cover images. That is because there are multiple photos that share the same amount of user votes but the cover image slots are limited. For example, consider an extreme case where none of the photos related to one restaurant receives any votes. In this case, under the crowd-based cover image system, the platform cannot solely rely on the crowd voting mechanism to select cover images and a tie-breaking mechanism is needed. In a more realistic scenario, consider a restaurant where the photo that receives the 5th highest number of votes and the photo that receives the 6th highest

number of votes has the same number of votes. Because there can only be five cover images, the platform needs a tie-breaking mechanism to pick one among the two to be the cover image of this restaurant.

During our study period and before the implementation of the AI-based system, the platform utilized a tie-breaking mechanism that relies on an aesthetic score of each photo generated by a third-party application programming interface (API). In this appendix, we exclude these 574 restaurants from the analysis to ensure clean identification. The results, reported in Table 3, are consistent with our main results. Namely, the coefficient of $PostImplementation$ is positive and statistically significant for all models, indicating the positive impact of the implementation of the AI-based cover image systems on the number of clicks that the platform receives.

Table 3: Impact of the AI-based system on user interactions (metric: number of clicks).

	(1)	(2)	(3)	(4)
Variables	$\ln(y_{it}+1)$	$\ln(y_{it}+1)$	$\ln(y_{it}+1)$	$\ln(y_{it}+1)$
$PostImplementation_t$	0.172*** (0.022)	0.172*** (0.009)	1.270*** (0.455)	1.260*** (0.190)
$ln(NumberofReviews)_{it}$	1.017*** (0.006)	0.361*** (0.067)	1.017*** (0.006)	0.361*** (0.066)
$AverageRatings_{it}$	0.028*** (0.009)	0.130 (0.087)	0.028*** (0.009)	0.134 (0.086)
$ln(RestaurantTenure)_{it}$	-0.381*** (0.008)	-0.429*** (0.092)	-0.381*** (0.008)	-0.407*** (0.092)
Linear time trend	-0.002 (0.002)	0.001 (0.001)	-0.0.87*** (0.031)	-0.084*** (0.013)
Constant	2.679*** (0.057)	3.707*** (0.577)	2.104*** (0.257)	3.009*** (0.583)
Time-fixed effects	No	No	Yes	Yes
Restaurant-fixed effects	No	Yes	No	Yes
Observations	39,728	39,728	39,728	39,728
R^2	0.396	0.035	0.401	0.086

Note: Standard errors in parentheses are robust and clustered by restaurant. *** p<0.01, ** p<0.05, * p<0.1.

E.2.2 Varying the Time Window

In our main specification, we use the data from eight weeks before and eight weeks after the implementation of the AI-based cover image system. We next consider different time windows to verify the robustness of our results under a different time horizon. Specifically, we consider time windows of four weeks and twelve weeks.

The results of Table 4 are consistent with our main result. The estimated coefficient of $PostImplementation_t$ continues to be positive and statistically significant across the alternative specifications. We highlight that the performance impact of the AI-based system grows over time (i.e., the magnitude of the estimated coefficient increases with the length of the time window). If we compare the average effects, we find a 8.81% performance increase when using a four-week time window. The performance increase becomes 16.05% (resp. 16.44%) when the comparison is based on a eight-week (resp. 12-week) time window.

Table 4: Impact of the AI-based system on user interactions (metric: number of clicks).

Variables	$\ln(y_{it}+1)$	
	4 weeks	12 weeks
$PostImplementation_t$	0.421***	3.474***
	(0.077)	(0.257)
$ln(NumberofReviews)_{it}$	-0.345***	-0.365***
	(0.066)	(0.040)
$AverageRatings_{it}$	0.514***	0.322***
	(0.230)	(0.090)
$ln(RestaurantTenure)_{it}$	-0.548***	-0.249***
	(0.145)	(0.0511)
Linear time trend	-0.042***	0.164***
	(0.012)	(0.012))
Constant	3.839***	0.949***
	(0.817)	(0.310)
Time-fixed effects	Yes	Yes
Restaurant-fixed effects	Yes	Yes
Observations	24,456	73,368
R^2	0.072	0.084

Note: Standard errors in parentheses are robust and clustered by restaurant.
*** p<0.01, ** p<0.05, * p<0.1.

E.2.3 Alternative Identification Strategy 1: RDiT

Our regression specification in Equation (1) is a time- and restaurant-fixed effects model that is extensively used in the literature (e.g., Cavusoglu et al., 2016). To ensure that our results are not solely driven by a specific identification strategy, we next consider an alternative strategy. In particular, we exploit the regression discontinuity in time (RDiT) framework (Hausman and Rapson, 2018) to estimate the difference in the performance of the AI-based system relative to the crowd-based system. The RDiT framework is commonly used to establish causality for natural experiments such as the one in our setting (see the list of studies that use RDiT for causal inference in Hausman and Rapson, 2018). Similar to the traditional regression discontinuity design, which estimates a treatment effect when an observed assignment variable exceeds a known cutoff value (Lee and Lemieux, 2010), in RDiT, the assignment variable is determined as a function of time. In our context, all the restaurants in the platform are suddenly "treated" at the time when the AI-based system is implemented. Thus, by leveraging the timing of the system implementation, we examine whether the AI-based system leads to a discontinuous jump in the outcome variable (i.e.,

number of clicks) at the cutoff point. Specifically, we consider the following RDiT specification:

$$\ln(y_{it}^{*}+1) = \beta_0 \quad + \quad \beta_1 PostImplementation_t + \sum_{k=1}^{K} r_k \big(Week_t - Week_c\big)^k \qquad (1)$$
$$+ \quad \sum_{k=1}^{K} \rho_k \big(Week_t - Week_c\big)^k \times PostImplementation_t + \varepsilon_{it},$$

where $ln(y_{it}^{*}+1) = ln(y_{it}+1) - ln(\overline{y}_t{+}1)$ is the adjusted outcome variable for restaurant i in week t. Namely, for each restaurant, we subtract the mean value over the time periods prior to the implementation date from the dependent variable. As before, $PostImplementation_t$ takes the value 1 after the implementation of the AI-based system and 0 otherwise. The difference $(Week_t - Week_c)$ refers to the duration (in weeks) with respect to the system implementation time. Similar to our main analysis, we consider a time window of eight weeks (we observe similar results for alternative time windows). We report the results from estimating the RDiT model from Equation (1) in Table 5.

Table 5: Regression discontinuity in time analysis (metric: number of clicks).

Variables	$\ln(y_{it}^{*}+1)$
$PostImplementation_t$	0.065***
	(0.010)
$Week$	-0.005***
	(0.001)
$Week \times PostImplementation_t$	0.007***
	(0.002)
Constant	-0.130***
	(0.007)
Observations	48,912
R^2	0.003

Note: Standard errors in parentheses. *** p<0.01, ** p<0.05, * p<0.1.

We highlight that RDiT is used primarily to establish causality around a short time period surrounding the cutoff time. Thus, the interpretation of the estimated coefficient of $PostImplementation_t$ is meant to be interpreted as the impact around the implementation time of the AI-based system. In other words, right after the implementation of the AI-based system, we observe a 6.5% jump in the number of clicks on the platform. This result further validates our previous findings.

E.2.4 Alternative Identification Strategy 2: PSM+DiD

Here, we consider an alternative identification strategy to further alleviate endogeneity concerns. Particularly, we utilize the propensity score matching (PSM) in combination with difference-in-differences (DiD), which is widely used in the literature to establish causal inferences, as an identification specification. Our control group consists of restaurants whose cover images continue to be exactly the same after the cover image system changes from the crowd-based system to the AI-based system. There are 108 restaurants that fit this definition. Then, for each restaurant in the control group, we find a restaurant that is similar in terms of observational characteristics but has different cover images after the cover image system changes, and treat it as a treated restaurant. To ensure the consistency with our main analyses, we consider restaurant characteristics that we use to conduct the heterogeneity analysis in Section 5.2 as matching variables for the PSM.

Table 6: Matching variables balance

	Control	Treatment	p-value
$RestaurantTenure$	4.822	4.7457	0.563
$NumPhotos$	0.017	0.021	0.814
$ResRating$	3.859	3.948	0.458
$ResPrice$	1.667	1.519	0.165

Table 6 reports the average difference on the matching variables between control and treated restaurants. Note that none of the variable is statistically significant, indicating that the matching process successfully finds pairs of restaurants that are similar in terms of these observable characteristics. We next use this matched dataset to estimate the following DiD specification:

$$\ln(y_{it}+1) = \gamma(Treatment_i \times PostImplementation_t) + \boldsymbol{\beta X_{it}} + \alpha_i + \delta_t + \epsilon_{it}. \tag{2}$$

In Equation (2), i denotes the restaurant and t denotes the time period (i.e., $t = 1$ for the first time period, $t = 2$ for the second time period, and so on). The variable y_{it} represents the dependent variable of interest, which is the number of clicks. Meanwhile, α_i captures restaurant fixed effects, which represent the restaurant's specific characteristics that impact our dependent variable and vary across a pair of matched restaurants but remain the same across time. The variable δ_t captures time fixed effects, which represent time-specific characteristics that affect our dependent variable. $Treatment_i$ is an indicator variable that represents if the restaurant is treated. $PostImplementation_t$ is an indicator to capture the post-treatment periods ($t > 8$). Lastly, $\boldsymbol{X_{it}}$ is a vector of control variables that consists of restaurant tenure and linear time trend.

Before proceeding with the DiD specification, we test the parallel trends assumption (i.e., the fact that the number of clicks received by treated restaurants is parallel to the number of clicks received by control restaurants during the pre-treatment period). For this test, we follow the approach from the literature based on the Augmented Dickey-Fuller (ADF) test of stationarity (Pamuru et al., 2021; Khern-am-nuai et al., 2018). Specifically, the definition of a stationary time series requires the mean, variance, and autocorrelation structure to not change over time. Hence, if the difference in our dependent variable between the treated and control groups satisfies the stationarity test (i.e., a test where the null hypothesis is that the variable contains a unit root and the alternative hypothesis is that the variable is generated by a stationary process), then this difference would indicate that the variables in the treated and control groups follow the same trend. The ADF test for the number of clicks yields a $Z(t)$ of -3.232 with a corresponding p-value of 0.0182. We thus can conclude that the parallel trends assumption holds.

Table 7: Difference-in-Differences results (metric: number of clicks).

Variables	$\ln(y_{it}+1)$
$Treatment_i \times PostImplementation_t$	0.065***
	(0.032)
$ln(RestaurantTenure)_{it}$	0.223*
	(0.130)
Linear time trend	-0.150***
	(0.045)
Constant	-0.009
	(0.732)
Observations	3,456
R^2	0.114

Note: Standard errors in parentheses are robust and clustered by restaurant. *** p<0.01, ** p<0.05, * p<0.1.

The results from the DiD specification are reported in Table 7. We find that the coefficient of $Treatment_i \times PostImplementation_t$ is positive and statistically significant, hence indicating the positive effect of adopting the AI-based cover image system on the number of clicks, which is consistent with our main result.

E.2.5 Alternative Model for The Selection of Cover Images

In Section5.3.3, we perform a regression model to identify factors that influence the selection of cover images by the crowd. Since the dependent variable of the analysis, the number of votes, is a count variable, we consider an alternative model that utilizes the negative binomial model in this robustness test. The results, reported in Table 8, are consistent with our main results.

Table 8: The selection of cover images by the crowd.

	The number of votes
$Brightness$	-1.249*** (0.279)
$Saturation$	0.530** (0.266)
$RuleofThirds$	-0.290 (0.244)
$DiagonalDistance$	0.109 (0.171)
$\ln(NumFollowers)$	0.268*** (0.010)
$\ln(PhotoTenure)$	-1.743*** (0.031)
$Feature$	0.475*** (0.092)
$\ln(NumReviews)$	0.520** (0.024)
$AvgRatings$	-0.001 (0.068)
Constant	0.506 (0.396)
Observations	132,865
Log-Likelihood	-6,849

Note: Standard errors in parentheses. *** p<0.01, ** p<0.05, * p<0.1.

E.2.6 Placebo Tests

To ensure that the effects we uncover are not artifacts of time-correlated confounders, we conduct a series of placebo tests. Specifically, we request the platform to access the data from the entire 2018 calendar year for this analysis (recall that the implementation of the AI-based system occurred in 2019). More precisely, we consider each week in 2018 as a potential placebo implementation date and re-estimate the coefficient of $PostImplementation_t$ from Equation (1) for each run. Accordingly, the first placebo test is such that the first eight weeks of 2018 are pre-treatment, the intervention date is the beginning of the ninth week, and the eight subsequent weeks are post-treatment. Similarly, the last placebo test is such that week 37 to week 44 are pre-treatment, the beginning of week 45 is the intervention date, and week 45 to week 52 are post-treatment. In total, we have 36 synthetic scenarios. Note that out of 3,057 restaurants in our dataset, 481 restaurants did not exist in 2018 so they are dropped from our placebo tests.

After estimating the value of β from our 36 placebo tests, we conduct two diagnostic statistical tests that are commonly used in the literature for placebo testings (e.g., Cheng et al., 2020; Park et al., 2020). First, we use an independent t-test to check whether the average placebo β estimate is statistically larger than zero. Second, we statistically compare the magnitude of the placebo β estimates relative to the actual estimated value from the implementation of the AI-based system.

Table 9 reports the mean and standard error of the placebo β estimates, along with the results from both diagnostic statistical tests. As we can see, we fail to reject the null hypothesis that the average placebo estimate is zero (p-value=0.351). In addition, we find that the the placebo estimates of β is statistically lower than the actual estimated β at $p < 0.01$. In conclusion, the results from our placebo tests demonstrate that it is very unlikely that our observed effects are occurring by chance.

Table 9: Results of placebo tests.

Estimations	$\ln(y_{it}+1)$
Mean of placebo β	0.178
Standard error of placebo β	0.459
Replication	36
Actual estimated β	1.428
H_0: placebo $\beta = 0$	
H_a: placebo $\beta > 0$	
t-score	0.387
p-value	p=0.351
H_0: placebo $\beta >$ Actual estimated β	
H_a: placebo $\beta \leq$ Actual estimated β	
t-score	-2.728
p-value	p<0.01

E.3 Additional Performance Heterogeneity

In this appendix, we examine additional heterogeneity effects of the implementation of the AI-based cover image system with respect to (i) $NumPhotoClicks$ (the number of photo clicks each restaurant attained during the crowd-based cover image system), and (ii) $Urban$ (whether the restaurant is located in an urban area or not). For the variable $Urban$, we follow the approach commonly used in the literature to divide restaurant locations into two types: urban (i.e., restaurants located in an area where people have relatively less limited access to the restaurants) and non-urban areas (i.e., restaurants located where people have relatively limited access to the restaurants) based on the median value of the number of restaurants in the area. The results are presented in Table 10. First, restaurants that perform well in the crowd-based system era (i.e., those that receive a higher number of photo clicks), benefit less from the implementation of the AI-based system. Second, restaurants located in urban areas benefit less from the implementation of the AI-based cover image system than restaurants located in rural areas. We also vary the threshold by considering the first quartile. The results are qualitatively similar.

Table 10: Additional Heterogeneity of the impact of the AI-based system on user interactions (metric: number of clicks).

	(1) $\ln(y_{it}+1)$	(2) $\ln(y_{it}+1)$
$PostImplementation_t$	1.443*** (0.175)	1.452*** (0.176)
$ln(NumberofReviews)_{it}$	0.386*** (0.058)	0.386*** (0.058)
$AverageRatings_{it}$	0.058 (0.051)	0.064 (0.051)
$ln(RestaurantTenure)_{it}$	-0.298*** (0.072)	-0.316*** (0.071)
Linear time trend	-0.098*** (0.012)	-0.098*** (0.012)
$ln(NumPhotoClicks)_{t=0} \times PostImplementation_t$	-0.037*** (0.008)	
$Urban_i \times PostImplementation_t$		-0.309*** (0.011)
Constant	2.486*** (0.413)	2.582*** (0.411)
Time-fixed effects	Yes	Yes
Restaurant-fixed effects	Yes	Yes
Observations	48,912	48,912
R^2	0.089	0.088

Note: Standard errors in parentheses are robust and clustered by restaurant. *** p<0.01, ** p<0.05, * p<0.1.

E.4 Investigating Potential Cannibalization Effects of the AI-based Cover Image System

In this appendix, we investigate the potential cannibalization effect of the AI-based cover image system. Particularly, we examine the impact of the AI-based cover image system on photo votes and photo uploads. Here, we use the specification in Equation (1) where the dependent variable y_{it} is the number of photo votes and the number of photo uploads.

Table 11: Impact of the AI-based system on user interactions (metrics: photo votes and photo uploads).

Variables	$ln(NumberofPhotoVotes_{it}+1)$	$ln(NumberofPhotoUploads_{it}+1)$
$PostImplementation_t$	-0.063 (0.094)	-0.107 (0.097)
$ln(NumberofReviews)_{it}$	0.461*** (0.065)	0.590*** (0.084)
$AverageRatings_{it}$	-0.317 (0.028)	-0.450** (0.022)
$ln(RestaurantTenure)_{it}$	-0.264*** (0.050)	-0.274*** (0.045)
Linear time trend	0.005 (0.006)	0.008 (0.007)
Constant	0.770*** (0.276)	0.683** (0.289)
Time-fixed effects	Yes	Yes
Restaurant-fixed effects	Yes	Yes
Observations	48,912	48,912
R^2	0.009	0.018

Note: Standard errors in parentheses are robust and clustered by restaurant. *** p<0.01, ** p<0.05, * p<0.1.

The results in Table 11 show that the coefficient of $PostImplementation_t$ is not statistically significant for both the number of photo votes and the number of photo uploads. Thus, although the number of clicks significantly increases after the adoption of the AI-based system, there is no statistically significant change in user partic-

ipation measured by the number of photo votes and photo uploads by platform users three months after adopting the new system. This finding showcases that the adoption of the AI-based cover image system does not cannibalize other user participation metrics.

References

Cavusoglu, H., T. Q. Phan, H. Cavusoglu, and E. M. Airoldi (2016). Assessing the impact of granular privacy controls on content sharing and disclosure on facebook. *Information Systems Research 27*(4), 848–879.

Cheng, Z., M.-S. Pang, and P. A. Pavlou (2020). Mitigating traffic congestion: The role of intelligent transportation systems. *Information Systems Research 31*(3), 653–674.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT press.

Hausman, C. and D. S. Rapson (2018). Regression discontinuity in time: Considerations for empirical applications. *Annual Review of Resource Economics 10*, 533–552.

Howard, A. G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.

Kao, Y., C. Wang, and K. Huang (2015). Visual aesthetic quality assessment with a regression model. In *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 1583–1587. IEEE.

Khern-am-nuai, W., K. Kannan, and H. Ghasemkhani (2018). Extrinsic versus intrinsic rewards for contributing reviews in an online platform. *Information Systems Research 29*(4), 871–892.

Lee, D. S. and T. Lemieux (2010). Regression discontinuity designs in economics. *Journal of economic literature 48*(2), 281–355.

Lu, X., Z. Lin, H. Jin, J. Yang, and J. Z. Wang (2014). Rapid: Rating pictorial aesthetics using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 457–466.

Ma, S., J. Liu, and C. Wen Chen (2017). A-lamp: Adaptive layout-aware multi-patch deep convolutional neural network for photo aesthetic assessment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4535–4544.

Murray, N., L. Marchesotti, and F. Perronnin (2012). Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2408–2415. IEEE.

Pamuru, V., W. Khern-am-nuai, and K. Kannan (2021). The impact of an augmented-reality game on local businesses: A study of pokémon go on restaurants. *Information Systems Research 32*(3), 950–966.

Park, Y., Y. Bang, and J.-H. Ahn (2020). How does the mobile channel reshape the sales distribution in e-commerce? *Information Systems Research 31*(4), 1164–1182.

Talebi, H. and P. Milanfar (2018). Nima: Neural image assessment. *IEEE Transactions on Image Processing 27*(8), 3998–4011.