Online Appendix

A Design of the cloud-based automated system

Our cloud-based automated system integrates streaming sensor data, artificial intelligence (AI) and customer communications. It has three major modules (Figure A.1).

- **Remaining days inference module.** This module receives real time data, like temperature, humidity as well as images of products, to infer remaining days (i.e., food quality) of fresh produce. Environmental factors, like temperature and humidity, have long been identified as key contributors to the deterioration of fresh produce (Kim et al. 2015). Recent advances in computer vision have opened another opportunity to assess food quality based on images captured from cameras. The technique has been applied in grading fruits and vegetables (Blasco et al. 2007, Cubero et al. 2011, Wang and Nguang 2007, Zou et al. 2010). It is worth noting that the inference from the raw data (i.e., temperature, humidity, images, ect.) into remaining days (i.e., food quality) can happen either at the edge or in the cloud. The trade-off lies between network latency and computation resource. While cloud computing exploits centralized resources to process and analyze data, it puts great pressure on the network bandwidth and leads to high latency in data transmission in a wireless network (Satyanarayanan et al. 2009). Edge computing moves computing power from the server to the edge that is at the proximity of the data source (Shi et al. 2016). It speeds up data processing and relieves network pressure. However, it creates new challenges for hardware design as edges (devices/sensors) are inherently limited in resources of energy and power.[1]

- **AI module.** This module implements a deep reinforcement learning (DRL) algorithm to obtain optimal pricing and information strategies. It consists of two parts. One is an online inference model that provides real time decisions. It takes inputs of inventory level and remaining days (i.e., food quality) and outputs optimal price and whether to disclose remaining days in real time. The other is an offline training model. It utilizes historical data

[1] Recent advances in hardware design include field-programmable gate arrays (FPGAs) (Biookaghazadeh et al. 2018, Liao et al. 2013) and application-specific integrated circuits (ASICs).

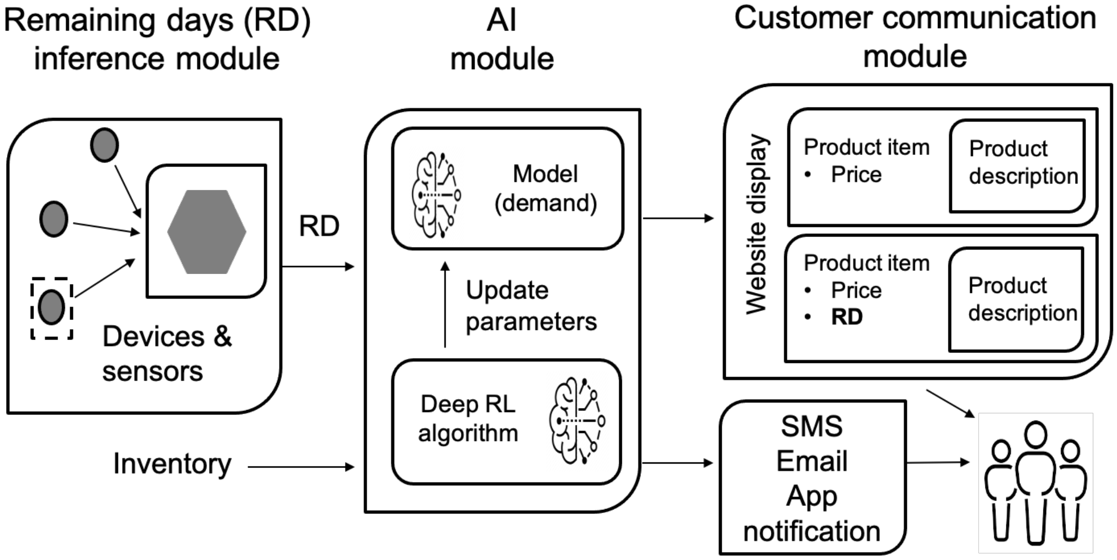

Figure A.1: Design of the cloud–based automated system

on inventory level and remaining days from cloud storage to update the model periodically. The online inference model then uses the updated offline training model to make real time decisions. The design could capture potential changes in the complex and evolving market demand.

- **Customer communications module.** This module can be realized through Twilio, a native cloud application. Powered by Amazon Web Services (AWS), Twilio provides the communications services to customers such as Uber, Netflix, and Airbnb for customer engagement.[2] Communication services include but are not limited to website display, text messages, emails, and push notifications on mobile phones. If it is optimal to disclose food quality to customers based on the online inference model in the AI module, a specialized cloud function is triggered to enable Twilio to communicate with customers about remaining days (i.e., food quality) along with the price of the product.[3]

Our work focuses on the AI module. We ask given that the grocery stores have set up the system, how could the retailer derive business decisions (i.e., pricing and information disclosure) from the

[2] AWS, "Twilio Case Study."

[3] Twilio provides API to top 3 cloud platforms in the United States, i.e., AWS, Google Cloud, and Microsoft Azure.

real time monitoring capability on the remaining days (i.e., food quality)? Thus, our work focuses on what *benefits* grocery retailers could get from the system. Though cost analysis of the technology is not the focus of our paper, it is essential for implementation in practice. Therefore, we outline relevant costs as well as benefits as below.

- **Costs.** The cost will include infrastructure investment, like operating IoT devices to collect data to infer remaining days. Such costs at retail stores can vary greatly depending on factors including the type of IoT devices, the type of internet connection it uses, and where it is deployed. Utilizing cloud resources will incur additional costs to keep the system running in real time and implement AI methods to derive optimal decisions. Finally, it will incur human capital as experts in IoT and/or AI are needed.

- **Benefits.** Benefits include both profit improvements and food waste reductions. Reducing food waste is important for companies who value sustainability since food waste brings negative social externalities, like environmental pollution. It is estimated that one-third of all food worldwide is wasted, totalling 2.6 trillion pounds and generating about 9% of worldwide annual greenhouse gas emission.[4] According to Economic Research Service, the social cost of 1 pound food waste ranges from $42 to $805.[5] Thus, it is critical for grocery stores to take sustainability into account when making business decisions.

[4] FAO, “Food Wastage Footprint” (2013); IPCC, “Mitigation of Climate Change” (2014).

[5] The Estimated Amount, Value, and Calories of Postharvest Food Losses at the Retail and Consumer Levels in the United States.

B PPO

We estimate the objective function $J(\omega)$ using an advantage function that subtracts a value function $V(s_t)$ from the expected future reward:

$$J(\omega) = \mathbb{E}_{\rho_\omega(\tau)}[A_t], \tag{B.1}$$

where $A_t = E[R_t] - V(s_t) = \sum_{t'=t} \gamma^{t'-t} r(s_{t'}, a_{t'}) - V(s_t)$ and γ is the discount rate. Note that the value function is not related to ω. This guarantees that the expectation of $V(s_t)$ with respect to ω is zero. Thus, the introduction of $V(s_t)$ will not change the bias of $J(\omega)$, but only reduce the variance of $J(\omega)$. In practice, the value function $V(s_t)$ is unknown, so it is typically approximated by a parameterized function $V_\phi(s_t)$ with parameters ϕ.

A trust region is normally a neighborhood centered around the current solution. It limits the amount by which any update is allowed to change the policy and is adjusted from iteration to iteration. We construct the trust region using the KL divergence, denoted as $\mathrm{KL}\left[\pi_{\omega_{\text{old}}}|\pi_\omega\right]$. It measures the difference between the policy (a probability distribution) and the old policy (a probability distribution). PPO attempts to optimize a *soft proximal surrogate* function:

$$J^{KL}_{PPO}(\omega) = \mathbb{E}_{\rho_{\omega_{\text{old}}}}[\frac{\pi_\omega(a_t|s_t)}{\pi_{\omega_{\text{old}}}(a_t|s_t)} A^{\omega_{\text{old}}}(a_t, s_t)] - \beta \mathrm{KL}\left[\pi_{\omega_{\text{old}}}|\pi_\omega\right], \tag{B.2}$$

where $A^\omega(s_t, a_t) = \mathbb{E}_\omega\left[R_t|s_t, a_t\right] - V_\phi(s_t)$. The ratio of importance sampling, $\frac{\pi_\omega(a_t|s_t)}{\pi_{\omega_{\text{old}}}(a_t|s_t)}$, measures the difference between the policy and the old policy. The second term resembles a penalty on the KL divergence constraint with the penalty coefficient β. The penalty coefficient β is adapted to achieve some target value of the KL divergence during each policy update. The first term approximates the advantage function locally at the old policy. However, this term becomes less accurate as the policy moves away from the old policy. The inaccuracy has an upper bound that is determined by the KL divergence $\mathrm{KL}[\pi_{\omega_{\text{old}}}|\pi_\omega]$ (for the proof, see Schulman et al. 2015). Given the upper bound of this inaccuracy, the optimal policy calculated within the trust region is guaranteed to always perform better than the old policy. Another heuristic to prevent the policy from moving too far away from the old policy is to constrain the ratio of importance sampling within a range. Thus, the objective

Algorithm 1 PPO

```
for $i \in \{1, \cdots, N\}$ do
    Run policy $\pi_\omega$ for $T$ time steps, collecting $\{s_t, a_t, r_t\}$
    Estimate advantages $\hat{A}_t = \sum_{t'>t} \gamma^{t'-t} r_{t'} - V_\phi(s_t)$
    $\pi_{\text{old}} \leftarrow \pi_\omega$
    for $j \in \{1, \cdots, M\}$ do
        $J_{PPO}(\omega) = \sum_{t=1}^{T} \min(\frac{\pi_\omega(a_t|s_t)}{\pi_{old}(a_t|s_t)} \hat{A}_t, \text{clip}(\frac{\pi_\omega(a_t|s_t)}{\pi_{old}(a_t|s_t)}, 1-\epsilon, 1+\epsilon)\hat{A}_t) - \beta \text{KL}[\pi_{old}|\pi_\omega]$
        Update $\omega$ by a gradient method with respect to $J_{PPO}(\omega)$
    end for
    for $j \in \{1, \cdots, B\}$ do
        $L_{BL}(\phi) = -\sum_{t=1}^{T}(\sum_{t'>t} \gamma^{t'-t} r_{t'} - V_\phi(s_t))^2$
        Update $\phi$ by a gradient method with respect to $L_{BL}(\phi)$
    end for
end for
```

function can be rewritten as

$$J_{PPO}^{clip}(\omega) = \mathbb{E}_{\rho_{\omega_{\text{old}}}}[\min(\delta(\omega)A^{\omega_{old}}(s_t, a_t), \text{clip}(\delta(\omega), 1-\epsilon, 1+\epsilon)A^{\omega_{old}}(s_t, a_t))], \tag{B.3}$$

where $\delta(\omega) = \frac{\pi_\omega(a_t|s_t)}{\pi_{\omega_{\text{old}}}(a_t|s_t)}$. The clipped function restricts the ratio $\delta(\omega)$ within the range $[1-\epsilon, 1+\epsilon]$ (for more details, see Schulman et al. 2017). The clipping helps reduce variance and stabilizes the training. Thus, we implement PPO using both an adaptive KL penalty and a clipped ratio of the importance sampling. See Algorithm 1 for the pseudo-code of the core PPO algorithm.

We parameterize policy π_ω and the value function V_ϕ using neural networks. We employ *stochastic gradient descent* to update the gradient. Given the dynamics of the environment, we repeatedly simulate a trajectory $\{(s_t, a_t, r_t)\}_{t=1}^{n}$ from the old policy, using Monte Carlo, and obtain an estimation $\hat{\nabla}_\omega J_{PPO}(\omega)$ based on that trajectory. Given $\hat{\nabla}_\omega J_{PPO}(\omega)$, we iteratively update ω by stochastic gradient descent to minimize the negative of the objective function, that is, $-J(\omega)$:

$$\omega_{t+1} := \omega_t + \beta_{step} \cdot \hat{\nabla}_\omega J(\omega_t), \tag{B.4}$$

where ω_t is the value of ω at iteration t and α_{step} is a proper step size.

C Estimates on customers' perceived deterioration rate: *maximum a posteriori* (MAP)

When an information strategy is incorporated (i.e., Model 2), at time t, the retailer decides with a probability m_t whether to disclose food quality. We assume that customers know that food quality deteriorates according to an exponential function, $\hat{\theta}_t = Qe^{-\hat{\gamma}t} + \epsilon_t,\ \epsilon_t \sim N(0, \sigma^2)$. If the retailer discloses information on remaining days (i.e., food quality), customers will observe the true food quality θ_t and will update their belief $\hat{\gamma}$ accordingly. The more information the customers receive, the more accurate their belief about food quality will be.

Suppose customers' perceived deterioration rate $\hat{\gamma}$ has a prior distribution $p_0(\hat{\gamma}) \sim N(\mu_0, \sigma_0^2)$. Since we consider three groups of customers, we set the prior mean at $\mu_0 = 0.15$ for customers who perceive quality lower than the actual level, i.e., a higher deterioration rate. We set the prior mean at $\mu_0 = 0.05$ for customers who perceive quality higher than the actual level, i.e., a lower deterioration rate. We set the prior mean at $\mu_0 = 0.1$ for customers who perceive quality as the same as the actual level. When the customers do not receive any message from the retailer, they have a belief $\hat{\gamma}$ drawn from $N(\mu_0, \sigma_0^2)$. After customers observe the true food quality θ_t disclosed by the retailer, the posterior distribution over $\hat{\gamma}$ can be calculated by Bayes rule:

$$p(\hat{\gamma}|\theta, t) = \frac{p(\theta|t, \hat{\gamma})p(\hat{\gamma})}{p(\theta, t)}, \tag{C.1}$$

where $p(\theta|t, \hat{\gamma})$ is the likelihood model for food quality θ. We employ *maximum a posteriori* (MAP) to obtain customers' perceived deterioration rate $\hat{\gamma}$ given the true quality θ disclosed by the retailer at time t. MAP gives estimates of the maximum value of the distribution. This method leads to fast inference, since it only requires one sample from the posterior distribution. Fast inference is important for real time decision making, as in our case. Thus, given a set of data points $\{\theta_i, t_i\}_{i=1}^{n}$, we want to get the optimal $\hat{\gamma}^\star$ that maximizes the posterior $p(\hat{\gamma}|\boldsymbol{\theta}, \boldsymbol{t})$:

$$\hat{\gamma}^\star = \arg\max_{\hat{\gamma}} p(\boldsymbol{\theta}|\hat{\gamma}, \boldsymbol{t})p(\hat{\gamma}). \tag{C.2}$$

If we take $z := \theta_t - \hat{\theta}_t = \theta_t - Qe^{-\hat{\gamma}t}$, it is easy to show that $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2)$. Suppose we observe n data points $\{\theta_i, t_i\}_{i=1}^n$, we can write the log-likelihood of θ as

$$\log p(\theta|t, \hat{\gamma}) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\bigg(\sum_{i=1}^{n}(\theta_i - \exp(-\hat{\gamma}t_i))^2\bigg). \tag{C.3}$$

The log-likelihood function of the prior distribution of $\hat{\gamma}$ can be calculated as

$$\log p_0(\hat{\gamma}) = -\frac{1}{2}\ln(2\pi\sigma_0^2) - \frac{1}{2\sigma_0^2}(\hat{\gamma} - \mu_0)^2. \tag{C.4}$$

Combining the above two equations, we can obtain the log posterior distribution $\log p(\hat{\gamma}|\theta, t)$ as

$$\begin{aligned}\log p(\hat{\gamma}|\theta, t) &\propto \log p(\theta|t, \hat{\gamma}) + \log p_0(\hat{\gamma}) \\ &\propto -\frac{1}{2\sigma^2}\bigg(\sum_{i=1}^{n}(\theta_i - \exp(-\hat{\gamma}t_i))^2\bigg) - \frac{1}{2\sigma_0^2}(\hat{\gamma} - \mu_0)^2 \\ &\propto -\bigg(\sum_{i=1}^{n}(\theta_i - \exp(-\hat{\gamma}t_i))^2 + \frac{\sigma^2}{\sigma_0^2}(\hat{\gamma} - \mu_0)^2\bigg). \end{aligned} \tag{C.5}$$

Since there is no close form solution to obtain optimal $\hat{\gamma}$, we perform gradient ascent iteratively to get $\hat{\gamma}$ that maximize the posterior $\log p(\hat{\gamma}|\theta, t)$:

$$\hat{\gamma}_{i+1} := \hat{\gamma}_i + \alpha_{step} \cdot \nabla_{\hat{\gamma}} \log p(\hat{\gamma}_i|\theta, t) \tag{C.6}$$

where α_{step} is a relatively small step size.

D Simulated environments

We limit the selling period to $T = 12$ days (approximately two weeks). We start with $C = 500$ units of inventory, and the unit cost of inventory is $q = 3$. We formulate our problem as a Markov Decision Process (MDP) which consists of $\{S, A, R, T\}$, where S is the state space, A is the action space, R is the reward signal, and T is the transition probability that determines the next state given the current states and actions, denoted by $T(s'|s, a)$.

- **State** S**.** We consider two states. One is the inventory level, i.e., how much inventory is left. The other is the remaining days (i.e., food quality) which refers to the remaining days before the product remains saleable. We normalize the remaining days at the beginning of the selling season as 1.

- **Action** A**.** We consider two actions. One is price p_t which ranges between zero and six.[6] The other is information disclosure m_t which ranges between zero and one. This gives us the probability that a retailer would like to disclose the information on remaining days.

- **Reward** R**.** We take profit as the reward and try to maximize the discounted total profit within the selling season $T = 12$. The discount rate is 0.9.

- **Transition probability** T**.** The transition probability, or the market demand, depends on customers' purchasing behavior. We model consumers' behavior in line with prior literature (Bitran and Mondschein 1997; Feng and Gallego 1995; Gallego and Ryzin 1994; Zhao and Zheng 2000). The arrival of customers is assumed to be a Poisson process with intensity λ. The probability that a customer will buy the product is affected by the customer's reservation price and reservation quality, which are given by $(1 - F(p_t))M(\hat{\theta}_t)$, where $F(\cdot)$ is the distribution of the reservation price and $M(\cdot)$ is the distribution of reservation quality. Note that $\hat{\theta}_t$ is customers' perceived quality of fresh produce. Customers' perceived quality depends on their perceived deterioration rate which is updated in a Bayesian way after observing the true quality disclosed by the retailer. A customer will buy the product if her reservation price is

[6] According to US Department of Agriculture, the retailing prices of most fresh produce fall between this range.

greater than the posted price p_t and the estimated quality $\hat{\theta}_t$ is greater than her reservation quality. Thus, we can consider the number of actual buyers as a nonhomogeneous Poisson process characterized by arrival rate $n_t(p_t)$ during the tth period, where

$$n_t(p_t) = \int_t^{t-1} \lambda(1 - F(p_t))M(\hat{\theta}_t)dt. \tag{D.1}$$

In line with prior literature (Bitran and Mondschein 1997; Gallego and Ryzin 1994), we assume a Weibull distribution for $F(\cdot)$ and $M(\cdot)$ functions. Therefore, the probability mass function for the number of actual buyers in period t is given by

$$Pr\{j_t(p_t) = j\} = exp(-n_t(p_t))n_t(p_t)^j/j!. \tag{D.2}$$

We compile our simulated environments in Open AI Gym, a toolkit for developing and comparing RL algorithms.[7] We train our PPO algorithm on these simulated environments. Table D.1 lists the specification of all parameters.

Table D.1: Model settings

Symbol	Description	Values
λ	Arrival rate of customers at each time period	70
C	Inventory level at the beginning	500
q	Ordering cost per unit of inventory	3
T	Number of time periods	12
Q	Initial quality level	1
γ	True value of deterioration rate	0.1
σ_0^2	Variance of customers' prior belief on deterioration rate	0.001
σ^2	Variance of noise in quality	0.004
ρ_F	Scale parameter in $F(\cdot)$	0.004
k_F	Shape parameter in $F(\cdot)$	4
ρ_M	Scale parameter in $G(\cdot)$	4
k_M	Shape parameter in $G(\cdot)$	1

[7] Various simulated environments for games can be found in Gym.

E Robustness check

E.1 More customers with high perceived quality

We consider two additional cases. In case 4, customers with *high* perceived quality take up 50%, and the other two groups take up 25% respectively. In case 5, customers with *high* perceived quality take up 75%, and the other two groups take up 12.5% respectively. We find that the superiority of a quality-based pricing strategy over a pricing strategy without quality still remains (Table E.1 and Table E.2).

Table E.1: Profit comparison for case 4 and case 5

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 4: 50% of high perceived quality	418.46	447.54	450.16	29.08***	2.62
Case 5: 75% of high perceived quality	466.68	492.95	490.62	26.27***	-2.33

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table E.2: Leftover inventory comparison for case 4 and case 5

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 4: 50% of high perceived quality	20.9	6.2	6.9	-14.7***	0.7
Case 5: 75% of high perceived quality	18.1	5.0	4.6	-13.1***	-0.4

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

But incorporating information disclosure no longer takes effect as the retailer has less incentives to disclose information (Figure E.1). Thus, Model 2 behaves in a similar way as Model 1 when most customers perceive quality as higher than the actual level. Additionally, average prices per unit sold are generally higher in case 4 and case 5 than those in case 1, case 2 and case 3 (Table E.3). This is intuitive as most customers perceive quality as high, they will still buy the product even though the prices are high.

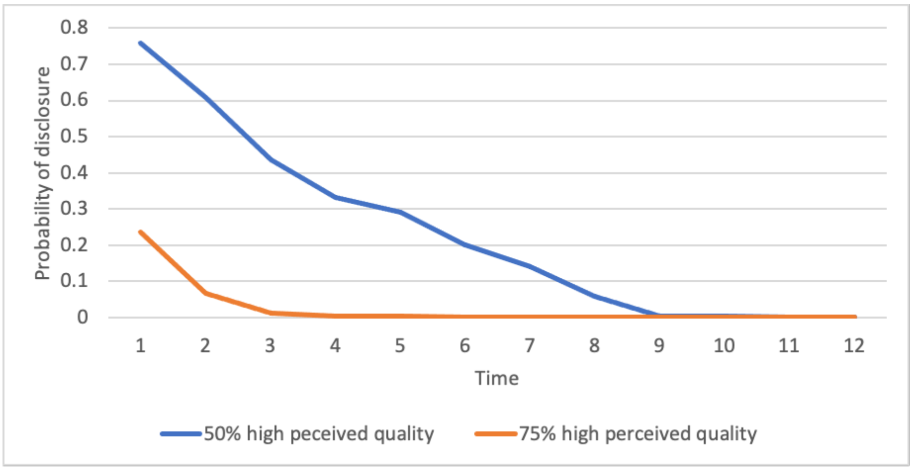

Figure E.1: Optimal information disclosure for case 4 and case 5

Table E.3: Average price per unit sold

	Benchmark	Model 1	Model 2
Case 5: 75% with high perceived quality	4.081	4.026	4.018
Case 4: 50% with high perceived quality	4.004	3.944	3.955
Case 1: 33% with low perceived quality	3.964	3.861	3.888
Case 2: 50% with low perceived quality	3.963	3.839	3.851
Case 3: 75% with low perceived quality	3.894	3.729	3.836

E.2 Different true deterioration rate and customers' perceived deterioration rate

We first test our results for a smaller bias between customers' biased perceptions about the quality and the true state of the quality. Keeping the true deterioration rate γ at 0.1, we set the perceived deterioration rate $\hat{\gamma}$ of customers with low perception quality at 0.12 and that of customers with high perception quality at 0.08. Main results remain the same qualitatively (Table E.4 and Table E.5).

Table E.4: Profit comparison when bias is small

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	403.30	425.15	430.44	21.85***	5.29*
Case 2: 50% of low perceived quality	387.27	410.94	421.02	23.67***	10.08***
Case 3: 75% of low perceived quality	363.73	386.88	416.78	23.15***	29.90***

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table E.5: Leftover inventory comparison when bias is small

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	22.1	8.4	4.8	-13.7***	-3.6**
Case 2: 50% of low perceived quality	22.4	6.7	7.0	-15.7***	0.3
Case 3: 75% of low perceived quality	28.5	8.9	8.8	-19.6***	-0.1

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

We also find that when the bias between customers' perceptions about food quality and true state of quality is small, the retailer has less incentive to disclose quality information (Figure E.2, compared to Figure 8). This corroborates that the retailer does not always have incentives to disclose the information on food quality. Some bias leaves room for the retailer to extract more value. Additionally, we find that the retailer is able to set higher average price per unit sold when the bias is smaller (Table E.6), especially for Model 1. This echoes the trade-off between pricing and information disclosure we find in Model 2. Greater alignment between customers' perceptions and the true state of the remaining days (i.e., food quality) enables the retailer to set higher prices.

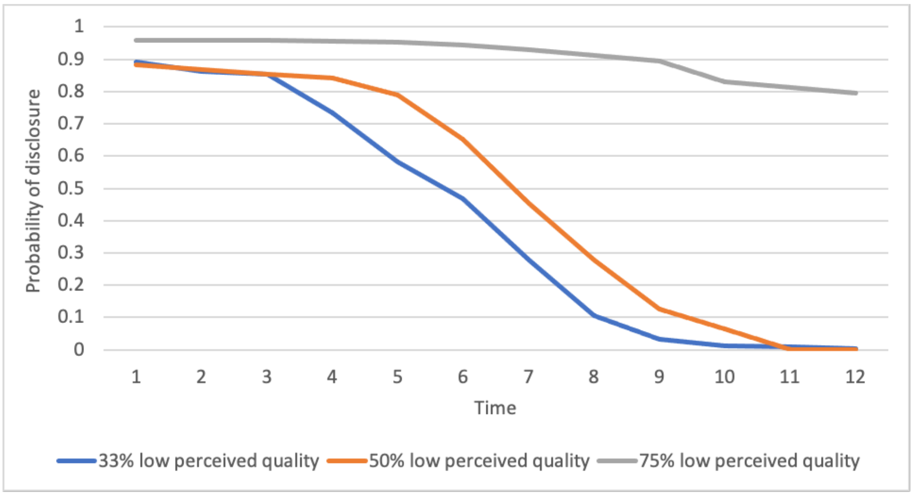

Figure E.2: Optimal information disclosure when bias is smaller

Table E.6: Average price per unit sold when bias is small

	Benchmark	Model 1	Model 2
Case 1: 33% with low perceived quality	3.983	3.916	3.898
Case 2: 50% with low perceived quality	3.952	3.874	3.897
Case 3: 75% with low perceived quality	3.953	3.842	3.902

Then, we change the true deterioration rate γ to 0.2, and set the perceived deterioration rate $\hat{\gamma}$ of customers with low perception quality at 0.25 and that of customers with high perception quality at 0.15. A higher deterioration rate indicates a lower perceived quality. Thus, the performances are generally worse than the case when $\gamma = 0.1$ (Table E.7 and Table E.8). Prices are again lower in Model 1 than those in Benchmark and Model 2, and the retailer has more incentive to disclose information when more customers perceive quality as low.

Table E.7: Profit comparison when deteriorate rate $\gamma = 0.2$

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	139.14	167.17	168.65	28.03***	1.48
Case 2: 50% of low perceived quality	82.35	111.58	119.79	29.23***	8.21**
Case 3: 75% of low perceived quality	−7.19	26.80	77.80	33.99***	51***

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table E.8: Leftover inventory comparison when deteriorate rate $\gamma = 0.2$

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	74.1	35.5	40.7	-38.6***	-5.2**
Case 2: 50% of low perceived quality	89.8	56.5	54.3	-33.3***	-2.2*
Case 3: 75% of low perceived quality	116.5	80.7	66.1	-35.8***	-14.6***

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

E.3 Demand rate

We consider two cases: (1) high demand rate where we set the arrival rate $\lambda = 80$, and (2) low demand rate where we set the arrival rate $\lambda = 60$. Our main results remain the same qualitatively. When demand is high, while the superiority of a quality-based dynamic pricing still remains, the benefit brought by information strategy diminishes. Table E.9 and Table E.10 show that in case 1 and case 2, the differences in profit and leftover inventory are statistically insignificant. This is intuitive as the retailer does not need to align customers' biased perceptions in order to drive the demand up. When the majority of customers perceives the quality as low, information disclosure still plays a role in aligning customers' biased perceptions with the true state. Thus, a combination of quality-based dynamic pricing and information disclosure (Model 2) helps increase profit and reduce food waste in case 3. When demand rate is low, information disclosure plays a bigger role. The improvement in terms of increased profit and reduced food waste is highest when most customers have lower perceptions about the quality than the actual quality (Table E.11 and Table E.12).

Table E.9: Profit comparison when demand is high

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	550.48	572.72	569.14	22.24***	-3.56
Case 2: 50% of low perceived quality	518.90	543.81	539.43	29.92***	-4.38
Case 3: 75% of low perceived quality	465.71	493.51	501.77	27.80***	11.26**

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table E.10: Leftover inventory comparison when demand is high

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	13.5	7.3	7.9	-6.2***	0.6
Case 2: 50% of low perceived quality	17.0	4.6	4.6	-12.4***	-1.3
Case 3: 75% of low perceived quality	21.3	6.4	4.6	-14.9***	-1.9*

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table E.11: Profit comparison when demand is low

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	159.62	185.33	196.19	25.71***	10.86***
Case 2: 50% of low perceived quality	110.21	141.22	187.34	31.01***	46.12***
Case 3: 75% of low perceived quality	51.82	71.19	170.46	19.37***	99.27***

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table E.12: Leftover inventory comparison when demand is low

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	68.5	36.7	33.2	-31.8***	-3.5**
Case 2: 50% of low perceived quality	82.9	44.6	30.5	-38.3***	-14.1***
Case 3: 75% of low perceived quality	90.4	61.3	39.6	-29.1***	-21.7***

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

E.4 Linear deterioration of fresh produce

We assume that the true quality of fresh produce deteriorates according to a linear form

$$\theta_t = 1 - \gamma t + \epsilon_t \tag{E.1}$$

where γ represents the true deterioration rate and ϵ_t is noise that is normally distributed according to $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. We assume that customers are aware of this linear deterioration form, so their perceived quality is

$$\hat{\theta}_t = 1 - \hat{\gamma} t + \epsilon_t \tag{E.2}$$

We set the true deterioration rate at $\gamma = 0.06$. We set perceived deterioration rate of customers who perceive quality as low at $\hat{\gamma} = 0.08$, and that of customers who perceive quality as high at $\hat{\gamma} = 0.04$. In model 2 when information disclosure is allowed, we assume customers' perceived deterioration rate has a normal prior with $p_0(\hat{\gamma}) \sim \mathcal{N}(\mu_0, \sigma_0^2)$. Customers will update their perceptions in a Bayesian way after they observe information on remaining days (i.e., food quality) disclosed by the retailer. Table E.13 and Table E.14 present results on profits and leftover inventory. A quality-based pricing strategy outperforms a pricing strategy without quality in all cases. The incorporation of information disclosure further improves profit and reduces food waste at the end of the selling season. On average, prices are lower in Model 1 than those in the Benchmark. In Model 2, the retailer is able to maintain prices at roughly the same level. Finally, the retailer has more incentives to disclose information when there are more customers with low perceived quality.

Table E.13: Profit comparison when quality deteriorates linearly

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	392.97	421.11	452.06	28.14***	30.95***
Case 2: 50% of low perceived quality	348.88	377.43	444.88	28.55***	67.45***
Case 3: 75% of low perceived quality	279.99	302.14	428.57	22.15***	126.43***

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table E.14: Leftover inventory comparison when quality deteriorates linearly

	Benchmark	Model 1	Model 2	Model 1 - Benchmark	Model 2 - Model 1
Case 1: 33% of low perceived quality	23.9	7.0	5.5	-16.9***	-1.5
Case 2: 50% of low perceived quality	31.1	8.3	7.0	-22.8***	-1.3
Case 3: 75% of low perceived quality	38.2	15.9	8.8	-22.3***	-7.1***

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

E.5 Robustness check of PPO

Finally, we test the sensitivity of our algorithm PPO to several key parameters that determine the performance and convergence of the algorithm.

- **Hidden size of the networks**. The hidden size of the neural networks specifies how many neurons each hidden layer has. A larger size of nerual network usually has better approximation ability, but it is more likely to lead to overfitting.

- **Step size in stochastic gradient descent (SGD)**. The step size of SGD determines the convergence speed of the training process. Usually using a larger step size allows the training process convergences faster, but the final solution may not be optimal. Here we use an adaptive stochastic gradient algorithm called Adaptive Moment Estimation (ADAM) to make the training more stable.

- **Batch size in SDG**. The batch size controls the variance of SGD. A large variance could lead to unstable learning process. We use a variant of batch size to verify the stability of the optimization process.

- **Clip ratio range**. The clip ratio range, denoted as $[1-\epsilon, 1+\epsilon]$ in equation B.3, controls the variance of the importance sampling. A smaller clip ratio range will give a more biased estimator with lower variance.

We examine the total reward and leftover inventory under different values of these parameters. To save space, we exhibit the results for Model 2 under Case 1. Results for other models and cases are qualitatively similar. Table E.15 reveals that PPO is quite robust to these key parameters. Total reward and leftover inventory stay the same across different settings of parameters. The differences are not statistically significant.

Table E.15: Performance of Model 2 Case 1 under different parameters

	Profit	**Leftover inventory**
Hidden size		
32	**422.16**	**7.95**
64	421.98	8.08
128	421.90	8.09
256	423.33	7.73
Batch size		
64	**422.16**	**7.95**
128	421.60	8.02
256	421.91	8.00
Step size		
0.0001	420.84	8.31
0.0003	**422.16**	**7.95**
0.001	421.92	8.17
0.003	423.83	7.49
Clip ratio		
0.1	421.27	8.01
0.2	**422.16**	**7.95**
0.3	420.55	8.01

Note: Parameters employed in main results are displayed in bold